NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

A PRODUCTION EARLY WARNING SYSTEM (PEWS) MODEL WHICH PREDICTS FUTURE USAREC MISSION ACCOMPLISHMENT

by

Scott G. Roesler

September 1994

Thesis Advisor: Lyn R. Whitaker

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1994	Master's Thesis

4. TITLE AND SUBTITLE
A PRODUCTION EARLY WARNING SYSTEM (PEWS) MODEL WHICH PREDICTS FUTURE USAREC MISSION ACCOMPLISHMENT (U)

5. FUNDING NUMBERS

6. AUTHOR(S)
Roesler, Scott G.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release, distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This thesis develops a framework for a statistical Production Early Warning System (PEWS) model which predicts the United States Army Recruiting Command's contract production. Model predictions are based on the initial Armed Forces Qualification Test (AFQT) taken by applicants over the past two years, the number of applicants expected to take the AFQT throughout the projection period, the historical probability that an applicant will sign a contract, and the distribution of time from when applicants take the AFQT until they sign a contract. Model parameters are based on the last five years of historical testing and contracting data. Yearly, seasonal, and monthly trends are incorporated by analyzing historical data using semi-monthly segments split on the 15th of the month. The model predicts contract production overall and for seven separate mission box categories. Performance of the model is measured by subtracting the number of actual contracts from the number of predicted contracts, and dividing by the number of actual contracts for FY 1993 time periods. The model's accuracy is greatly reduced because the testing data base does not include applicants who took the AFQT as part of a batch test group.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Production Early Warning System, Contract Production, Initial AFQT, Batch AFQT, Time to Contract, Semi-Monthly Period, Prediction Period	156
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

Approved for public release; distribution is unlimited.

A PRODUCTION EARLY WARNING SYSTEM (PEWS) MODEL WHICH PREDICTS FUTURE USAREC MISSION ACCOMPLISHMENT

Scott G. Roesler
Captain, United States Army
B.S., United States Military Academy, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1994

Author: _____
Scott G. Roesler

Approved by: _____
Lyn Whitaker, Thesis Advisor

_____
Harold Larson, Second Reader

_____
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

This thesis develops a framework for a statistical Production Early Warning System (PEWS) model which predicts the United States Army Recruiting Command's contract production. Model predictions are based on the initial Armed Forces Qualification Test (AFQT) taken by applicants over the past two years, the number of applicants expected to take the AFQT throughout the projection period, the historical probability that an applicant will sign a contract, and the distribution of time from when applicants take the AFQT until they sign a contract. Model parameters are based on the last five years of historical testing and contracting data. Yearly, seasonal, and monthly trends are incorporated by analyzing historical data using semi-monthly segments split on the 15th of the month. The model predicts contract production overall and for seven separate mission box categories. Performance of the model is measured by subtracting the number of actual contracts from the number of predicted contracts, and dividing by the number of actual contracts for FY 1993 time periods. The model's accuracy is greatly reduced because the testing data base does not include applicants who took the AFQT as part of a batch test group.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this thesis may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

Additionally, a portion of the analysis conducted for this thesis was performed using *APL2/PC* and *AGSS*. Naval Postgraduate School uses this program under a test agreement with IBM Research.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Each year, the Department of the Army (DA) stipulates the number and quality of recruits required to access into the Army (enter into military service). The United States Army Recruiting Command (USAREC) is responsible for ensuring that this accession mission is met, both in volume and in the quality of recruits required. To meet the volume requirement, USAREC specifies a contract mission which identifies the number and type of recruits needed to sign a contract. To ensure that the overall quality requirements of the Army are met, USAREC first divides the contract mission into 23 separate "mission box" categories which are based on the gender, education level, AFQT score, and prior service status of each prospective recruit. USAREC then divides this annual mission into separate monthly missions to ensure an even flow of recruits throughout the fiscal year.

One of the most important concerns at USAREC is predicting how many contracts will be achieved in the future for each of the critical mission box categories. An accurate and timely prediction would allow decision makers at USAREC the necessary time to change policy, adjust mission quotas, or exert command influence on the subordinate recruiting elements to ensure that the yearly accession mission is met. The current production projection system has three basic limitations. First, it is based primarily on subjective and qualitative analysis, so that the quality of the prediction relies heavily on the experience of the analyst and his or her knowledge of the prior years' production figures, trends, and idiosyncrasies. Second, the system only gives two weeks advance notice that the contract mission is in jeopardy of not being met. Thus, the command may not have sufficient time to adjust policy or give guidance to the recruiters in the field to attempt to manage the situation. Finally, the system does not specifically account for those individuals who took the Armed Forces Qualification Test (AFQT), were qualified for enlistment, but did not sign a contract immediately. Prospective recruits who have taken the AFQT but not yet signed a contract constitute a large pool of potential future contracts, since an AFQT score remains valid for two years after the test date. The

current system is based solely on persons who took the AFQT in the current month, and does not consider this reserve of possible recruits from the past two years. These limitations suggest that a more quantitative and comprehensive projection system would be useful in predicting USAREC contract mission accomplishment.

The Production Early Warning System (PEWS) model framework is designed to predict USAREC's contract production up to six weeks into the future, based upon quantitative analysis of the last five years of historical data. The main inputs to the model include the number of prospective recruits that have taken the AFQT over the past two years, the number of individuals that are expected to take the AFQT in the near future, the probability that a prospective recruit will actually sign a contract once he or she has taken the AFQT, and the amount of time that is expected to elapse between the day that an individual takes the AFQT until he or she signs a contract, (time to contract, or TTC). Yearly, seasonal and monthly time trends are incorporated into the model by analyzing historical data using semi-monthly segments of the beginning versus the end of the month. The model predicts mission accomplishment for four separate and four composite mission box categories:

GMA and GFA (high school graduates, male or female, AFQT score >= 50),
GMB and GFB (high school graduates, male or female, 30 < AFQT score < 50)
SMT and SFT (high school seniors, male or female, AFQT score >= 30)
OTH (all other mission box categories), and
TOT (all mission box categories combined).

The PEWS model provides prediction capability for each of these eight mission box categories for any period of time up to six weeks in the future.

The testing data was drawn from the Testing data base maintained by the Military Entrance Processing Command (MEPCOM), and includes information on all individuals who have taken the AFQT since 1988. Two problems with the Testing data base used for this thesis are that subsequent AFQTs and individuals who took the AFQT as part of a group (batch AFQT) are not included. The contract data was drawn from the Contract data base maintained by USAREC. These two data bases were matched by SSN to

determine time to contract distribution. Approximately 40% of all contract records did not match a testing record. These are assumed to be from individuals who took the batch AFQT. These individuals are stored on a auxiliary data base due to the large number of records and relatively small percentage of contracts achieved from them.

Validation of the model was performed by running two and four week projections into FY 1993, the most current year included in the historical data. FY 1994 testing data was not available. The monthly validation runs show that the model predicts on the average ten percent fewer contracts than are actually realized , with extensive variation between months and a range from (-17% to 3%). The overall underestimation stems from the exclusion of the batch AFQT records, while the month to month variation is explained by seasonal and school year changes which effect graduate and senior contracts significantly. The semi-monthly validation also shows that the underestimation occurs almost entirely during the second half of the month. This is due to individuals who took the AFQT more than once and have a different probability of contracting and a different time to contract distribution. This was not considered in the model because only the initial AFQT test dates were available.

Conclusions and Recommendations. To increase model accuracy, the model should incorporate a "Retest" category that includes all individuals that have taken the AFQT more than once, and a batch AFQT category that includes all individuals that took the AFQT as a part of a group test, both with separately derived parameters to describe their different behaviors. Data is currently available in the MEPCOM data base to do this. Also, time series analyses should be performed to further clarify the impact of seasonal, school year, contracting incentives, and other historical trends on contract production.. Overall, The PEWS model as it is currently defined provides an excellent platform from which to launch a truly effective production prediction model. and is sufficient to be used to augment the current qualitative production projection system.

I. INTRODUCTION

The United States Army Recruiting Command (USAREC) is the Army agency responsible for recruiting civilians to enlist in the Army. The number and quality of recruits required to enlist (report to basic training) annually is specified by the Department of the Army (DA) in the DA accession mission. USAREC uses the DA accession mission to specify a contract mission, which stipulates the number of recruits required to sign an enlistment contract in that year. USAREC then partitions the yearly contract mission into quarterly and monthly contract mission quotas, to ensure a steady and regulated flow of enlistments throughout the year. Ultimately, the monthly contract mission quotas determine the number of signed enlistment contracts that each individual recruiter will be required to attain monthly.

In addition to the total number of accessions required, the DA also stipulates specific standards for the yearly enlistment quotas, to ensure that the overall quality of Army's soldiers remains high. To meet these quality control standards, USAREC divides the contract mission into 23 separate "mission box" categories, based on the gender, education level, Armed Forces Qualification Test (AFQT) score, and prior service status of each recruit (Ref. 8). The monthly contract quotas are then allocated by mission box category to each individual recruiter.

One of the most important concerns of USAREC is to be able to predict as far in advance as possible whether or not the monthly contract mission will be met. This advance warning provides the decision makers time to change policy, adjust mission quotas, or exert command influence as necessary to ensure that the yearly accession mission is met. The current system used to predict contract production is based on a qualitative analysis of historical data. In this thesis, a statistical model is developed to predict whether the monthly mission quota will be met up to six weeks in advance, based on quantitative analysis of five years of historical recruiting data (Ref. 8).

A. DEFINITIONS

Several terms are used throughout this thesis which have very specific meanings. Several of these terms are defined here.

Applicant: Any person who has been contacted by a recruiter and has expressed an interest in joining the Army.

The Armed Forces Qualification Test (AFQT): A general aptitude test used to classify applicants. The AFQT score is based on the percentile score that the applicant achieves among all other persons who have taken the test.

Military Entrance Processing Station (MEPS): The MEPS is responsible for administering the AFQT to applicants, conducting physicals to pre-qualify applicants for military service, providing career counseling on jobs available and incentives offered for prospective candidates for enlistment, and managing the paperwork for prospects who sign a contract.

Prospect: Any applicant who has taken the AFQT.

Contract: The official document that a prospect signs when he or she agrees to enter into the Armed Services.

TestDate: The date on which an applicant took the AFQT for the first time.

ContDate: The date that a prospect signed a contract.

Time to Contract (TTC): The number of days from taking the AFQT to actual contract signing.

Testing Mission Box Category: A classification based on the education level, sex, prior service status, and AFQT score of an applicant at the time they take the AFQT.

Contract Mission Box Category: A classification based on the education level, sex, prior service status, and AFQT score of a prospect at the time they signed a contract.

Production Projection System: Any system used to predict the number of contracts that are expected to be signed by a given date in the future.

Projection Period: The period of time between the current date and the projection date.

B. BACKGROUND

The current production projection system used by USAREC is based on a qualitative analysis of historical data. Each month, the total number of applicants who arrive daily at the MEPS are graphically compared against the (base case) number of applicants who arrived at the MEPS on the same specific days during the preceding three years. This system has been a fairly good indicator when factors known to have existed in the behavioral year are considered, but it relies heavily on the analyst's knowledge of the base case data. Projections from the current system provide a general picture from about two weeks into the month of whether the end-of-month contract mission will be above, at, or below the base case (Ref. 11).

The current USAREC production projection system has three basic limitations. First, it is based primarily on subjective and qualitative analysis. The quality of the prediction relies heavily on the experience of the analyst, and his or her knowledge of the prior years' production figures, trends, and idiosyncrasies. Second, the system gives only two weeks advance notice that the contract mission is in jeopardy of not being met. Thus, the command may not have sufficient time to adjust policy or give guidance to the field recruiting stations to attempt to remedy the situation. Finally, the system does not specifically account for those applicants who took the AFQT at the MEPS, were qualified for enlistment, but did not sign a contract immediately. Applicants who have taken the AFQT but not yet signed a contract constitute a large pool of potential future contracts, since a prospect's AFQT score remains valid for two years after the test date (Ref. 5). The current system is based solely on the applicants who took the AFQT in the current month, and does not consider this reserve of possible contracts from the past two years. These limitations suggest that a more quantitative and comprehensive projection system would be useful in predicting USAREC contract mission accomplishment (Ref. 1, 8).

C. PROBLEM STATEMENT

The goal of this thesis is to develop a framework for a statistical Production Early Warning System (PEWS) model that can predict USAREC's contract production up to six weeks into the future. The main prediction criteria are: 1) the number of applicants that have taken the AFQT over the two years prior to the current date and the number of prospects that have signed a contract in the current year; 2) the number of applicants that are expected to take the AFQT during the projection period; 3) the probability that an applicant will sign a contract given that he or she has taken the AFQT; and 4) the distribution of time that a prospect takes to sign a contract, given that he or she will sign a contract.

D. THESIS STRUCTURE

Chapter II of this thesis outlines the recruiting process within USAREC from the headquarters level down through the individual recruiters in the recruiting stations. Chapter III describes how the PEWS model works. The PEWS model is designed to predict the number of contracts that will be signed up to six weeks into the future, using information and parameters drawn from analysis of the last five years of historical MEPS testing and USAREC contracting data. This chapter first explains the assumptions made in developing the model and defines the model parameters. Then it describes the iterative structure and the mathematical formulation of the model. Chapter IV describes the data sources used to develop the parameters of the PEWS model. The first data source is the Testing data base, which stores data on all applicants beginning when they are first administered the AFQT until they report for basic training or are no longer considered as a viable applicant (more than two years have passed since they took the AFQT). The second source is the Contract data base, which contains the mission box classification of all applicants who have signed an enlistment contract. Both of these data bases span five years of historical recruiting data. Chapter V explains the procedure used to validate the

model, summarizes the results of this validation, identifies weaknesses in the model, and provides recommendations for further refinement and study based on these weaknesses.

II. RECRUITING PROCESS

The following sections explain the recruiting process at USAREC. The first section describes USAREC's organizational structure and how it interacts with the Deputy Chief of Staff for Personnel (DSCPER). The second section explains the steps that are used in breaking out the contract mission and discusses the mission box categories. The final section describes how recruiting is accomplished at USAREC's lowest organizational levels. This chapter is paraphrased from Chapter II of Reference 2.

A. USAREC

1. USAREC Organization

The highest level of the Army recruiting organization is the USAREC Headquarters (HQ), which provides leadership and coordination for all echelons of the recruiting organization (see Figure 1).

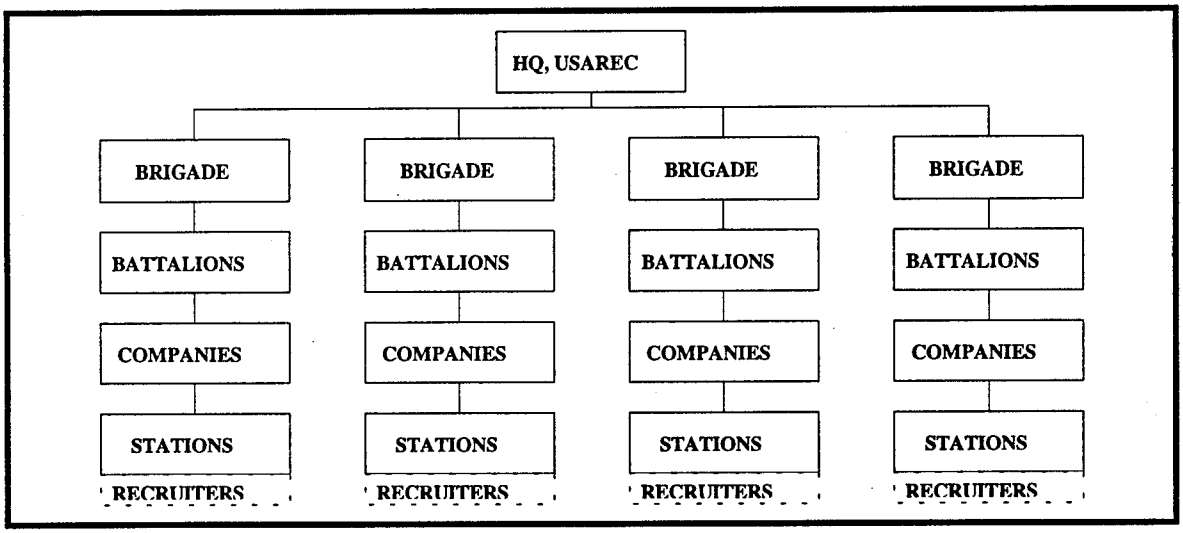


Figure 1 USAREC Organizational Structure

The Headquarters element of USAREC consists of ten directorates which form the Commanding General's staff. These ten directorates coordinate, plan and support the recruiting effort for the entire U.S. Army. The next subordinate level in the

USAREC chain of command consists of four recruiting brigades, each of which is responsible for one of four separate geographical regions of the continental United States (CONUS). Each brigade commands a number of recruiting battalions responsible for a specific area within the brigade's region (usually one or two states). Each of these battalions consists of recruiting companies, which are similarly comprised of recruiting stations, each with a small piece of the recruiting pie. The recruiting stations are the smallest organizational level within USAREC. Each station consists of two to twenty recruiters, who are the individuals that actually produce enlistment contracts. The recruiters form the sales force for the Regular Army.

Currently, USAREC consists of 4 brigades, 42 battalions, 222 companies, 1495 stations, and approximately 4200 recruiters. These numbers are constantly in a state of flux as the Army downsizes and USAREC realigns its sales force to keep up with ongoing changes in structure and mission requirements.

2. Interactions with DSCPER

The Directorate of Personnel Management at DCSPER is the Army agency that establishes the future personnel needs of the Army and stipulates the DA accession mission for USAREC. As stated before, the DA accession mission specifies the number of recruits who must access into the Army (report for basic training), and the proportions of those recruits required to meet the quality standards for the Army. These proportions are determined by six descriptive categories that place bounds on the composition mix of all recruits entering the Army. These six categories include recruits (i) with prior military service (PS), (ii) with no prior military service (NPS), (iii) with a high school diploma (HSDG), (iv) with no prior service and an AFQT score in the top 50th percentile (NPS-A), (v) with no prior service and an AFQT score between the 26th and 30th percentiles (TSC-4), and (vi) female (FEM) recruits. The DA stipulates a maximum allowable percentage of total recruits with prior military service and with an AFQT score of less than the 30th percentile. A minimum allowable percentage NPS-A and female recruits is also

mandated. Table I below summarizes the 1994 accession mission and tabulates the six descriptive categories that define the Department of the Army's quality control standards.

Total Accessions (Volume) = 75,000
Service Mix = 70,000 Non-Prior Service (NPS)
= 5,000 Prior Service (PS)
Quality Mix for NPS accessions
>= 95% must be high school graduates (HSDG)
>= 67% must score in the top 50th percentile on the AFQT (NPS-A)
<= 2% can score between the 21st and 30th percentile on the AFQT (TSC-4) *
Gender Mix for NPS accessions
>= 14.8 % must be female
* Current policy restricts TSC-4 to scores between the 26th and 30th percentile.

Table I. 1994 DA Accession Mission.

B. THE MISSIONING PROCESS

The Mission Division of the Program and Evaluation (PAE) Directorate is responsible for ensuring that USAREC is able to fulfill the DA accession mission. To do this, the analysts of the Mission Division set monthly recruiting goals on a quarterly basis. In these goals, they specify the number of individuals to be recruited (contracts to be signed) during each month of the quarter. This goal is referred to as the USAREC contract mission, and it incorporates a 15% increase above the DA accession mission to compensate for the corresponding percentage of individuals who renege on their contracts prior to reporting to basic training.

To ensure that the required proportions of the DSCPER quality standards are met, the USAREC analysts group the contract mission requirements into 23 separate mission box categories, which are based on the service history, education level, AFQT score and gender of the prospect. A recruit's service history defines whether he has prior service (PS) in the army or no prior service (NPS). There is also a special code for those prior

service prospects that were in the special forces (SPF). The education level identifies the prospect as a high school graduate with a diploma or an approved substitute (G), a senior in high school (S), a high school student who is not yet a senior (H), or a person who does not have a high school diploma (N). The AFQT score is based on the percentile score of the prospect's AFQT test. The three levels are: Greater than 50% (A), 31% to 50% (B), and 26% to 30% (4). The gender of a prospect is identified as male (M) or female (F). These codes are then combined to form the three letter designator codes for a prospect's mission box category, which are listed in Table II below.

MISSION BOXES

GMA	GFA	SMA	SFA	HMA	HFA	NMA	NFA	MPS	SPF
GMB	GFB	SMB	SFB	HMB	HFB	NMB	NFB	FPS	
GM4	GF4	SM4	SF4						

LEGEND

HIGH SCHOOL EDUCATION:

G	= Graduate	N	= Non-Graduate
S	= Senior	H	= In high school, not a senior

SERVICE HISTORY;

PS = Prior Service NPS = Non-Prior Service SPF = Special Forces

GENDER:

M = Male F = Female

AFQT SCORE:

A = 50th Percentile or better
B = between 31st and 49th Percentile
C = between 26th and 30th Percentile

Table II. USAREC Mission Box Categories.

Historically, some mission box categories are easier to fill than others, due either to a small mission requirement or a large population with a high propensity to enlist into a specific category. These mission box categories are deemed to be non-critical, and are not specifically addressed in this thesis. They include all prospects who have had prior service, prospects without a high school diploma, high school students who are not yet seniors, and all persons who score less than the 30th percentile on the AFQT. Only the

mission box categories considered to be critical (difficult to fill) are addressed in this thesis (Ref. 2). Of these difficult to fill mission boxes, the senior male and senior female AFQT level A and B categories were combined, because of the small number of actual cases involved in the SMB and SFB categories. Table III tabulates how the mission box categories were consolidated for analysis in this thesis.

Consolidated Mission Boxes	Mission Boxes included
GMA	GMA
GMB	GMB
GFA	GFA
GFB	GFB
SMT	SMA, SMB
SFT	SFA, SFB
OTH	GM4, GF4, SM4, SF4, HMA, HMB, HFA, HFB, NMA, NMB, NFA, NFB, MPS, FPS, SPF
TOT	All mission box categories combined

Table III. Combination of Critical Mission Box Categories.

Ultimately, the monthly recruiting mission must produce the required number of accessions with the desired mix of overall quality attributes. This requires analysts to closely monitor the recruiting process and modify the recruiting mission quotas as needed. Monthly recruiting quotas are assigned quarterly by mission box to each of the four recruiting brigades. The recruiting quarters are defined in Table IV below.

QUARTER	1	2	3	4
SEASON	Fall	Winter	Spring	Summer
MONTHS	Oct/Nov/Dec	Jan/Feb/Mar	Apr/May/Jun	Jul/Aug/Sep

Table IV. Definition of Recruiting Quarters.

Given the time required to distribute the mission quota throughout the recruiting organization, USAREC's mission branch typically determines and disseminates the quarterly quotas at least one quarter in advance. This allows the brigades to analyze their missions and distribute them to their battalions over the following three to five months. In a similar manner, the mission filters down through the companies and recruiting stations,

until it is broken down to the individual recruiter basis. Historically, a recruiter is usually expected to produce between one and two contracts each month (Ref. 2).

C. THE RECRUITING PROCESS

To accomplish their contract mission, recruiters begin by making contact with youths, or prospects, at high schools, recruiting stations, or through informal introductions. If a prospect is interested and eligible to join the Army, he or she becomes an applicant and is processed for enlistment at a Military Entrance Processing Station (MEPS). At the MEPS, each applicant takes the AFQT, and is given a thorough physical examination to ensure that he or she is fit for military service. Those applicants who score sufficiently high on the AFQT (26th percentile and up) and pass the physical become prospects and are allowed to talk to a Guidance Counselor, with whom they may select a career and sign a contract agreeing to join the Army. This contract is credited to the recruiter's mission box requirement for the month that the contract is signed, and the applicant enters the Delayed Entry Program (DEP). All applicants who sign a contract will remain in the DEP for at least one month. The purpose of the DEP is to allow the eligibility of each enlistee to be reviewed, and for a background check to be conducted on them to ensure that they have not falsified any information on their application.

Not all applicants who eventually sign a contract do so on their first visit to the MEPS station. There are three major reasons that an individual who will eventually sign a contract may not do so immediately. First, an otherwise fully qualified applicant may fail the physical for a minor, easily correctable reason. Some of the more common reasons range from having too much ear wax to reporting for the physical without wearing underwear. Due to the large number of prospects screened daily, even minor problems are not allowed to be corrected on the spot, and it is necessary to schedule the applicant to retake the physical at a later date, after the problem has been fixed. The second reason is that an applicant may wish to, or need to, retake the AFQT test. An applicant is allowed to take the AFQT as many times as he or she wishes, but a waiting period of at least six

months is required before a retest is allowed (in special cases, a month delay is allowed with the approval of the recruiting battalion commander). Improving one's AFQT test score is desirable because it can result in the applicant being eligible for a wider range of more technical, or more appealing, jobs in the Army. Finally, some applicants are simply not ready to sign a contract immediately. They may need more time to decide if they really want to join the Army, or they may wish to wait for a month (or more) to see if a more appealing job is offered at a later date. Of these three reasons for not signing a contract immediately, waiting for a better job option is the most common (Ref. 1, 8).

In the interim between signing the contract and beginning basic training, each recruit is enrolled in the DEP. During this time, the recruiter and the recruit maintain close communication to ensure that the recruit remains eligible for service. It is the recruiter's responsibility to ensure that each of his or her recruits eventually enters basic training, or the recruiter will lose the credit for that recruit on his or her mission quota. When this happens, the recruiter is required to replace the lost contract with a new contract of the same mission box classification.

III. PEWS MODEL

The Production Early Warning System model is designed to allow USAREC analysts to predict contract production up to six weeks into the future using statistical parameters derived from empirical analysis of the last five years of historical testing and contracting data. Written in PASCAL code, the model receives the current date and the prediction date from the analyst. It then calculates the actual number of contracts signed to date from historical contracting data. The model projects the number of contracts expected to be achieved during the projection period, based on the number of prospects who have taken the AFQT over the past five years, their historical propensity to sign a contract conditioned on the fact that they have taken the AFQT, and the estimated Time To Contract (TTC) distribution. Finally, the model predicts the number of applicants expected to take the AFQT during the projection period, and uses this prediction to project the number of contracts expected to be achieved from these future prospects, again using the propensity to sign a contract and the TTC distribution. Adding up the actual number of contracts, the projected number of contracts expected from those who have taken the AFQT from historical data, and the predicted number of contracts expected from those expected to take the AFQT during the projection period, the model provides an estimate of the number of total contracts expected to be achieved as of the projection date.

The parameters of the PEWS model incorporates yearly, seasonal, monthly and beginning vs. end of month time trends by breaking out the past five years of historical data into approximately 15-day increments. Model parameters are read from a text file which is designed to be easily updated from a Microsoft EXCEL worksheet. This allows the user to update the actual number of applicants who have taken the AFQT and the actual number of contracts that have been signed to date on a daily and a semi-monthly basis. This improves model accuracy by decreasing the accumulation error that grows as the current date progresses further away from the historical data. Also, other model

parameters can be updated yearly, to incorporate each previous year's information into the raw data source, which improves model accuracy from year to year.

Mission box classification of both applicants at the time of testing and prospects at the time they sign a contract are also considered in the formulation of the parameters. The model is designed to provide predictions for four separate mission box categories (GMA, GMB, GFA, GFB), three composite mission box categories (SMT, SFT, OTH), and the overall mission accomplishment (TOT). The TOT mission box category is simply an aggregation of all of the other mission box categories. The remainder of this chapter describes the structure of the model, discusses the assumptions made in building the PEWS model, and defines and explains each of the model's parameters.

A. ASSUMPTIONS

1. Consolidation of Mission Box Categories

USAREC uses 23 separate mission box categories in determining the mission requirements for each month. However, some of these categories are considered more critical to accomplishing the overall mission than others. To scale down the scope of this thesis, the most critical mission box categories were identified by the USAREC mission branch analysts (Ref. 1). These included GMA, GMB, GFA, GFB, SMA, SMB, SFA, and SFB. All other categories were consolidated into the OTH category, and considered as a group for continuity purposes only. Therefore, the OTH category is not expected to be of significant analytical use. Also, despite the fact that the Senior categories are considered important separately, the total number of contracts signed by the SMB and SFB categories are too small to obtain accurate parameters when considered separately. Therefore, they were consolidated into the SMT and the SFT categories accordingly.

2. Semi-Monthly Time Periods

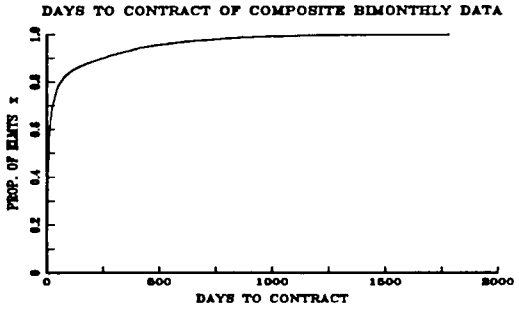
One of the most pronounced trends observed over the five years of historical data is that more contracts are signed at the end of the month than at the beginning of the month. The primary reason for this trend is that the recruiters are required to meet their monthly quota by the end of the month, but do not get additional credit for achieving their mission earlier than this. Therefore, the tendency to procrastinate results in a larger proportion of the monthly contracts signed during the last two weeks of the month rather than the first two weeks of the month (Ref. 8). To capture this result, the historical parameters in the model are classified into 24 semi-monthly periods, two for each month of the year. These periods were broken down by the 1st through the 15th of each month, and the 16th through the end of each month. In addition to capturing the trend of increased contracts at the end of the month, these semi-monthly classes also help to capture seasonal effects, major holiday effects, and the end of the Fiscal Year procrastination effects.

Figures 2 and 3 on the following page illustrate the differences between the time to contract distribution for the first half and the last half of a given month. Each figure depicts the cumulative distribution function and a table of quantile statistics for the TTC distribution of a random sample of 25 percent of all matched records. Figure 2 describes the days to contract for the first half of a given month, and figure 3 describes the last half. Note that the mean time to contract for the last half is four days longer than that of the first half. Also, as early as the 25th quantile, a shorter TTC distribution is observed for the second half of the month than for the first half. Similar analyses of the beginning vs. the end of the month was conducted for each of the separate mission box categories with identical results, which verifies that this phenomenon is consistent across all mission box categories.

Within the semi-monthly periods, it is assumed that contracts are signed according to a uniform distribution throughout the period. Each MEPS closes on weekends and national holidays, so no contracts are signed on these days. This results in a

lower daily average for contracts predicted within the projection period. This lower average is offset in the model by the assumption of uniformity.

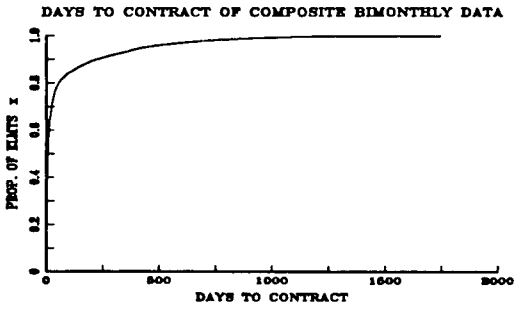
CDF OF 25 P/C R.S. OF FIRST 15 DAYS OF MONTH



CUMULATIVE DISTRIBUTION TABLE	
X	:TOTBIMO1
SELECTION	:ALL
X LABEL	:DAYS TO CONTRACT
NO. OF ELEMENTS	:13117
X MEAN	:77.781
STD. DEVIATION	:188.36
X MIN.	:0.1 0.1 0.1
X MAX.	:1784 1764 1707
QUANTILE OF X:	
0.05	: 1.0000
0.1	: 1.0000
0.2	: 1.0000
0.25	: 2.0000
0.3	: 3.0000
0.4	: 6.0000
0.5	: 8.0000
0.6	: 14.0000
0.7	: 27.0000
0.75	: 41.0000
0.8	: 64.0000
0.9	: 254.0000
0.95	: 458.6500

Figure 2. CDF of Time to Contract Distribution for Beginning of the Month.

CDF OF 25 P/C R.S. OF LAST 15 DAYS OF MONTH



CUMULATIVE DISTRIBUTION TABLE	
X	:TOTBIMO2
SELECTION	:ALL
X LABEL	:DAYS TO CONTRACT
NO. OF ELEMENTS	:14226
X MEAN	:73.643
STD. DEVIATION	:183.6
X MIN.	:0.1 0.1 0.1
X MAX.	:1744 1740 1696
QUANTILE OF X:	
0.05	: 1.0000
0.1	: 1.0000
0.2	: 1.0000
0.25	: 1.0000
0.3	: 2.0000
0.4	: 5.0000
0.5	: 7.0000
0.6	: 13.0000
0.7	: 24.0000
0.75	: 34.0000
0.8	: 56.0000
0.9	: 230.0000
0.95	: 440.2000

Figure 3. CDF of Time to Contract Distribution for End of the Month.

3. Time To Contract Distribution

The TTC distribution is based solely on the records in the Composite data base. Therefore, it only describes the days to contract for those applicants that took the AFQT after talking to a recruiter directly, and does not include those who took the AFQT as part of the batch tests. These batch testing records are not available for analysis, but since almost 40 percent of the overall contracts are realized from these batch tested applicants, they could not be completely ignored in building the model parameters. Therefore, for the purposes of this thesis, it was assumed that these batch AFQT records followed the same TTC distribution as did the regularly tested applicants. It is expected that the actual TTC distribution for these batch records is longer than the current distribution, which could cause a possible underestimation of the actual number of contracts signed in a given month.

The TTC distribution assumes a uniform distribution within each TTC window. For example, if six percent of the contracts are expected to fall within the 8-9 day TTC window, then it is assumed that three percent will sign a contract on the eighth day, and three percent will sign a contract on the ninth day. The TTC windows were chosen to contain approximately four to six percent of the total contracts signed, resulting in 20 total bins. It is considered impractical to break these categories into smaller bins of percentages of the overall contracts signed in a time period.

Each Time to Contract distribution is based on the difference between the date that a prospect signed a contract and the date that he or she took the AFQT, given that the prospect eventually signed a contract. Approximately 0.2 percent of these differences result in a negative number of days to contract, which suggests that these individuals actually signed a contract before they took the AFQT. Since this is theoretically impossible, it is assumed that these cases are due to erroneous entries of either the AFQT test date in the Testing data base of the contract date in the Contract data base. Therefore, these cases are ignored in determining the TTC distributions.

4. AFQT Retests

The AFQT testing records used to derive the parameters in this model are based solely on an applicant's initial AFQT test. This is because subsequent AFQT test dates were not available when the Testing data base was requested from the MEPCOM. Therefore, for the purposes of this thesis, it is assumed that model parameters can be accurately estimated using only the initial test, and that all distributions based on this initial test are the same as those for an applicant's most recent test. This is probably not entirely true, and it is suspected that a prospect who takes more than one AFQT has a higher probability of signing a contract than a prospect who has only taken an initial AFQT. In addition, a shorter TTC distribution is expected for a prospect taking a subsequent AFQT. This deficiency is addressed in more detail in Chapter V.

B. MODEL PARAMETERS

All parameters used in the PEWS model are based on the last five years of testing and contracting data because the automated system used to collect this data into a computerized data base was not implemented until the end of FY 1988. Therefore, only data from FY 1989 forward is complete, accurate, and accessible for analysis. The parameters are built compartmentally for each of the critical mission box categories, so that a separate and distinct projection can be run for each category. The TOT mission box category allows projection of the overall mission, regardless of mission box category. The semi-monthly categories allow the analyst to run a projection during any time of the year. Appendix E contains more complete information about each parameter, including parameter derivations, how the parameters are accessed and stored within the model, and schematics of the data structures. Model parameters are defined below.

1. Number Tested (FirstTest)

Number Tested represents the total number of applicants who have taken the AFQT, broken out for each testing semi-monthly period by year for the past five years,

and by the applicant's testing mission box classification. This parameter is used to calculate the actual number of applicants who have historically taken the AFQT. Number Tested should be updated every two weeks, and is denoted by

$$FirstTest_{i,n} \tag{1}$$

where i is the testdate semi-monthly period and n is the contracted mission box category.

2. Number Contracted (FirstCont)

Number Contracted represents the total number of prospects who signed a contract, broken out for each contracting semi-monthly period by year for the past five years, and by the applicant's contracting mission box classification. This parameter is used to calculate the actual number of prospects that have historically signed a contract. Number Contracted should be updated every two weeks, and is denoted by

$$FirstCont_{i,m} \tag{2}$$

where i is the contdate semi-monthly period and m is the projection mission box category.

3. Expected Number Tested (NumTest)

Expected Number Tested represents the number of applicants expected to take the AFQT in each future semi-monthly period. This parameter is used to predict the number of testers who will take the AFQT during each day of the projection period. Expected Number Tested should be updated yearly, and is denoted by

$$NumTest_{i} \tag{3}$$

where i is the testdate semi-monthly period.

4. Percent Contracted (PctCont)

Percent Contracted represents the probability that a prospect will sign a contract, given that he or she took the AFQT during a specific semi-monthly period and

was classified into a specific testing mission box category at that time. This parameter is used to predict the percentage of applicants who took the AFQT that are expected to sign a contract at some time during the next two years. This parameter should be updated yearly, and is denoted by

$$PctCont_{i,n} \tag{4}$$

where i is the testdate semi-monthly period and n is the tested mission box category.

5. Unmatched Contracted (PctUnm)

Unmatched Contracted represents the percentage of additional contracts who are expected to be obtained from the batch AFQT tests during a specific semi-monthly period for each critical mission box category. This parameter is used to adjust for the additional contracts that cannot be predicted from the data contained in the Testing data base. This parameter should be updated yearly. Unmatched Contracted is denoted by

$$PctUnm_{j,m} \tag{5}$$

where j is the contdate semi-monthly period and m is the contract mission box.

6. Time to Contract (TTC) Distribution

Time To Contract Distribution tabulates the probability that a prospect will sign a contract after a specific length of time, given that the prospect took the AFQT in a specific semi-monthly period and that the prospect was classified into a specific testing mission box category at the time he or she took the AFQT. This parameter is used to project unrealized contracts from two years prior to the current date forward into the projection period. This parameter should be updated yearly. The TTC Distribution is denoted by

$$TTC_{(j-i),n} \tag{6}$$

where i is the date that the prospect took the AFQT, j is the date that the prospect signed a contract, and n is the prospect's tested mission box category.

7. Daily Averages (DailyTest, DailyCont)

Daily Averages represents the number of applicants expected to take the AFQT on a single day in a specific testing semi-month period, and the number of prospects expected to sign a contract on a single day in a specific contract semi-monthly period. This parameter is used to predict the number of AFQT tests taken and the number of contracts signed on a given day of the projection period. This parameter should be updated daily if possible. Daily Averages are denoted by

$$DailyTest_i \tag{7}$$

and

$$DailyCont_j \tag{8}$$

where i is the test date and j is the contract date.

8. Percent Mission Box (PctNBox)

Percent Mission Box represents the probability that a prospect will sign a contract for a specific mission box category, given that a contract will be signed. This parameter predicts which mission box category a projected daily contract is expected to fall in to. This parameter should be updated yearly. Percent Mission Box is denoted by

$$PctNBox_m \tag{9}$$

where m is the projection mission box category.

C. MODEL STRUCTURE

The PEWS model is an interactive, menu driven program written in PASCAL code, designed for use on a personal computer. Parameter input is read from a pre-formatted, tab-delimited text file that can be updated in Microsoft EXCEL as required. The program is initiated by executing the Pascal program PEWSMAIN.pas from the TurboPascal 6.0 environment. The model consists of five major elements: user inputs, summation of contracts already realized, calculation of contracts predicted to be realized

from historical testing data, calculation of contracts projected to be realized from AFQT tests expected to be taken during the projection period, and model output.

1. User Inputs

From the main program menu, the user chooses the "run projection" option. The program then prompts the user to enter the current date, the projection date, and the mission box category for which a projection run is desired. The model assumes that the number of contracts signed on the current date is known.

2. Calculating Currently Realized Contracts

The first step the model takes is to calculate the number of contracts already known to be signed through the current date. This is accomplished by adding up the number of prospects that have signed a contract in the specified mission box category during each semi-monthly period through the end of the month prior to the current month. Then model then computes the total number of contracts that are signed on each day of the current month, up to and including the current date, and multiplies this daily total by the proportion of all contracts expected to fall within the specified mission box category. These two totals are then added together to get the total number of contracts currently signed at the start of the projection period:

$$\sum_{i=1}^{a} (FirstCont_{i,m}) + \sum_{j=1}^{c} (DailyCont_j)(PctNBox_m) \qquad (10)$$

where i = contdate semi-monthly periods
j = day of the current month
a = last semi-monthly period prior to the beginning of the current month
c = current calendar day
m = projection mission box category

3. Calculating Historically Predicted Contracts

The next step is to calculate the number of contracts predicted to be signed during each day of the projection period, based on unrealized contracts from the prior two

years of historical data. This is done by projecting from each day of the two years prior to the current date into each day of the projection period, using a set of nested loops. The outside loop steps through the projection period one day at a time (ProjDay). The inside loops step through the two years prior to the current date one day at a time (TestDay) until the day before the current day. For each possible set of TestDay and ProjDay, the model calculates: 1) the number of applicants that took the AFQT on the TestDay; 2) multiplies by the probability that an applicant will sign a contract given that the prospect took the AFQT during a specific semi-monthly period and the prospect's testing mission box category; 3) multiplies by the probability that a prospect will sign a contract on a specific ProjDay given that the prospect will sign a contract, that the AFQT was taken in a given semi-monthly period, and the prospect's testing mission box category; 4) adds the number of additional contracts expected to be realized from batch testing records for the prospect's contract mission box category and the ProjDay's semi-monthly period; and 5) multiplies by the proportion of contracts expected to fall into the projection mission box category. This sum estimates the number of contracts that are predicted to be obtained from historical AFQTs during each day of the projection period for the projection mission box category:

$$\sum_{i=(c-730)}^{c-1}\sum_{j=c}^{p}\left[\left\{\sum_{n=GMA}^{OTH}(FirstTest_{i,n})(TTC_{(j-i),n})(PctCont_{i,n})\right\}(PctUnm_{j,m})(PctNBox_{m})\right] \tag{11}$$

$$+\sum_{i=b}^{c-1}\sum_{j=c}^{p}\left[\left\{(DailyTest_{i})(TTC_{(j-i),n=TOT})(PctCont_{i,n=TOT})\right\}(PctUnm_{j,m})(PctNBox_{m})\right]$$

where i = julian test date
j = julian contract date
n = tested mission box category
m = projection mission box category
b = first day of current month
c = contract date
d = projection date

4. Calculating Predicted Contracts During the Projection Period

The next step is to calculate the number of contracts projected to be signed during each day of the projection period based on the number of applicants expected to

take the AFQT during this period. This is accomplished using the same steps as in (3) above, substituting the expected number of applicants predicted to take the AFQT on each TestDay in lieu of the actual number tested from historical data. It is basically the same procedure that was used in determining the historically predicted contracts, except this set of nested loops uses probabilistic data rather than known historical data as input. This sum represents the number of contracts expected to be achieved from applicants who are predicted to take the AFQT during the projection period, and is calculated as

$$\sum_{i=c}^{p-1} \sum_{j=c+1}^{p} \left[(DailyTest_i)(TTC_{(j-i),TOT})(PctCont_{i,TOT})(PctUnm_{j,m})(PctNBox_m) \right] \quad (12)$$

where i = julian test days
j = julian contract days
c = current date
p = projection date
m = projection mission box category

5. Model Outputs

Model output from the projection runs is sent both to the screen and to a text file called PEWSOUT.txt. This outfile serves as a log of all projections run during a single session. Output includes the number of contracts actually signed to date, the number of contracts predicted to be signed during the projection period (based on historical tested), and the number of contracts projected to be to be signed during the projection period (based on predicted tested).

IV. DESCRIPTION OF DATA SOURCES

The first hard data recorded at an organizational level in the recruiting process is taken when the applicant arrives at the Military Entrance Processing Station (MEPS) and takes the AFQT for the first time. This information is entered into the Military Entrance Processing Command (MEPCOM) Testing data base as an initial record, and is updated or appended each time the applicant retakes the AFQT or meets with a Guidance Counselor at the MEPS. MEPCOM is the source of raw data for all data bases used for analysis at USAREC, including the USAREC Contract data base. The main difference between the Testing data base and the Contract data base lies in the method that the USAREC Contract data base uses to affiliate records with a mission box category, a process which uses fields from the MEPCOM Testing data base. This chapter describes the two data bases, the methodology and underlying assumptions with which they are "cleaned" to remove incorrect and unneeded records, how they are manipulated to reformat the data into the desired configuration for analysis, and how they are merged to form a combined data base. For the remainder of this thesis, the MEPCOM data base will be referred to as the Testing data base, the USAREC data base will be referred to as the Contract data base, and the data base built from these two merged data bases will be referred to as the Composite data base. A detailed schedule of data transformations and the SPSS code that was used to transform each of the data bases into the desired final form is included in Appendix B.

A. TESTING DATA BASE

As stated above, the Testing data base is the primary source of all recruiting data. The data which contributes to the Testing data base is initially collected at each individual MEPS. MEPS are located throughout the country, and process thousands of applicants for possible enlistment into the military on a daily basis. The data on each applicant is entered automatically when his or her AFQT is electronically scored and processed at the

MEPS, and the applicant's record is then updated manually by the Guidance Counselor as needed. The data collected at the MEPS is transmitted electronically to the MEPCOM on a daily basis, where it automatically updates the Testing data base. Overall, the Testing data base contains a record of descriptive information for every individual who has taken the AFQT at a MEPS station, as well as the date that the individual took the AFQT. This thesis uses a subset of data fields drawn from the Testing data base. Table V below shows an excerpt from the Testing data base as it appeared in its original MEPCOM format.

SSN	SEX	EDYRS	AFQT	TESTDATE	PS
527757993	M	12L	26	880901	YY
267113812	M	12L	79	880901	NY
639051884	F	12L	39	880901	NN
363623841	F	12L	41	880901	NN
266695248	M	11S	27	880901	NN
560295853	M	12L	66	880901	NN
599043447	F	13L	23	880901	NN

Table V. Initial Form of the Testing Data Base.

The SSN, SEX, and AFQT fields are self explanatory. The first two characters of the EDYRS field identify the total number of years of school the applicant had completed when he took the AFQT. The third character indicates the highest academic level obtained by the applicant, e.g.: L = HS Grad, S = HS Senior, etc. The TESTDATE field represents the date (yymmdd) that the applicant took the AFQT. The PS field identifies the applicant's prior service status. The first character represents the DOD definition of prior service, and the second represents the DA definition. The DA definition is the indicator USAREC uses to determine an applicant's prior service status. The SEX, AFQT, EDYRS and PS fields are used to determine the testing mission box category of an applicant at the time that he or she took the AFQT.

As with any large data base, there are input errors in the Testing data base. In order to properly define the distributions and parameters for the model, these erroneous entries were removed from the data set. There are three main reasons which require that a record be removed from the Testing data. First, if a record is missing the individual's

Social Security Number (SSN) then the record is useless, since without a SSN it is impossible to identify a record as a unique data element. Second, records missing the test date are also inutile, since without a test date it is impossible to determine the distribution of time it takes for an applicant to sign a contract from when he first took the AFQT. Finally, if the SSN of an applicant is repeated on two or more records, then it is assumed that the most current record (with the latest test date) is inaccurate, and should be deleted. All subsequent transactions for an applicant should only update his or her existing initial record. Therefore, it is presumed that duplicate SSN records were mistakenly regenerated by a software problem (Ref. 8). Since the distributions in the model describe the time to contract for individuals who have taken the AFQT for the first time, then only the first AFQT test for each individual is considered to be accurate. Table VI below summarizes the number of erroneous records removed from the original Testing data base.

Reason	Number of Records in	Number
Missing SSN	880,404	1
Missing Test Date	880,403	64
Duplicate SSN	883,339	5
Outside 5-yr Test Date	883,334	34,471
TOTALS	845,863	34,541

Table VI. Summary of Testing Data Base Records Removed.

The next step in the manipulation of the Testing data base involves breaking variables into data elements, which aids in analysis and in building the distributions required for the model. This consists of two major changes in the format of the data. First, the TESTDATE field is divided into separate day, month and year fields. These fields are converted into their fiscal year equivalents to allow for easier analysis of time trends. Second, the mission box category for each of the remaining records in the Testing data base is derived. To determine the mission box, first the EDYRS category is used to determine the level of schooling attained by each applicant, which represents their high school graduate level, which is transformed into the GRAD field. Then the GRAD, SEX,

and AFQT fields are used to determine the mission box category for each record, which is put in the testing mission box (TNBOX) field. These changes are incorporated into the revised Testing data base as illustrated in Table VII below.

SSN	SEX	EDYRS	AFQT	TDATE	PS	TYR	TMO	TDAY	GRAD	TNBOX
527757993	M	12L	33	880901	YN	88	9	1	grad	GMB
267113812	M	12M	79	880901	YN	88	9	1	non	NMA
639051884	F	12L	39	880901	NY	88	9	1	grad	FPS
363623841	F	12L	41	880901	NN	88	9	1	grad	GFB
266695248	M	11S	55	880901	NN	88	9	1	senior	SMA
226046603	M	12L	84	880901	NN	88	9	1	grad	GMA
563372464	F	12L	43	880901	NN	88	9	1	senior	SFB
599043447	F	13L	23	880901	NN	88	9	1	grad	GF4

Table VII. Revised Form of the Testing Data Base.

After determining the mission box categories, the EDYRS, PS, SEX, AFQT and GRAD fields are deleted. Then the TESTYR and TESTMO variables are converted to their fiscal year equivalents, and then concatenated to form a separte TYRMO category. Finally, the testing mission box categories are consolidated into the mission box classifications identified as critical to the model, and saved into the TNBOX field. At this point, the Testing data base is in its final form, and is ready to be merged with the records in the Contract data base. In its final form, the only fields that are maintained are those which are used to derive the model parameters. An example of the Testing data base in its final form is shown in Table VIII below.

SSN	TDATE	FTYR	FTMO	FTYRMO	TDAY	TNBOX
267113812	880901	89	1	8901	1	GMA
363623841	880901	89	1	8901	1	GFB
266695248	880901	89	1	8901	1	OTH
563372464	880901	89	1	8901	1	SMT
560295853	880901	89	1	8901	1	GMA

Table VIII. Final Form of the Testing Data Base.

It is important to note that the TNBOX field is only an estimation of the prospect's mission box categorization at the time they first took the AFQT. In many cases, a prospect will retake the AFQT, achieve a high school diploma (or its equivalent), or

become a high school senior by the time that they actually sign a contract. Therefore, the contract mission box category, determined by USAREC for each individual at the time that they sign a contract, may be different than the Testing mission box category.

B. CONTRACT DATA BASE

The missioning branch of the PAE directorate uses the Contract data base as its primary analytical data source. The Contract data base is updated monthly from a DOD data base, which contains recruiting information for all branches of the military. For this thesis, an excerpt of the Contract data base consisting of the records of all prospects who signed a contract for the Army during FY 1989-1993 was used. The variables are similar to those of the Testing data base. The CDATE variable indicates the date a prospect signed a contract. An excerpt of the original Contract data base is seen in Table IX.

SSN	CNBOX	AFQT	CDATE
01584424	SMB	35	890620
02543537	SFA	62	920110
02643501	GMA	66	890824
02704151	NMA	71	901001
02705568	GFA	79	910115

Table IX. Initial Contract Data Base.

An example of the Contract data base in its final form is shown in Table X below.Similar steps were taken to remove incorrect or incomplete records from the Contract data base as were done for the Testing data base. There were no duplicate records found, and only one record had an incorrectly entered contract date. This record was removed, leaving 503,768 records for analysis. Finally, as with the Testing data base, the CDATE field was broken out into fields based on the fiscal calendar.

SSN	CDATE	FCYR	FCMO	FCYRMO	CDAY	NBOX
01584424	890620	89	9	8909	20	SMB
02543537	920110	92	4	9204	10	SFA
02643501	890824	89	11	8911	24	GMA
02704151	901001	91	1	9101	1	NMA
02705568	910115	91	4	9104	15	GFA

Table X. Final Form of the Contract Data Base.

C. COMPOSITE DATA BASE

The Composite data base, also called the Match data base, is formed by merging the Testing data base with the Contract data base. The main purpose for merging these data bases is to match the SSN's between the two, so that the date that a prospect signed a contract can be compared to the date that he or she took the AFQT for the first time. From this, the distribution of time from AFQT to contract can be determined. Not all of the records in the two data bases match up, and the reasons for these unmatched files from the Testing and Contract data are discussed below.

Unmatched records from the Testing data base represent those applicants who took the AFQT, but did not sign a contract. Therefore, these unmatched records are used to determine the number of applicants that will not sign a contract. These records are removed from the Composite data base and saved to a separate Unmatched data base.

Unmatched records from the Contracting data base represent those individuals that took the AFQT as part of a group test outside of the normal MEPS processing cycle. These group tests are administered by MEPS personnel at "field" sites (Ref. 5). The most common examples of these field sites include administering the test to an entire class of high school students at their high school, or setting up the test in a van, specially designed for this purpose, away from the MEPS and at the request of a local recruiter, and administering the test to any who wish to stop off the street and take it. For the purpose of this thesis, tests administered in this manner are referred to as "batch" AFQT tests. The results of these batch tests are not stored in the Testing data base, due to the large volume of testing records collected and the comparatively small expected percentage of contracts achieved through these tests. These unmatched Contract data base records are used to determine the number of contracts signed by applicants who took their initial AFQT as part of a batch AFQT test. Therefore, the records are removed from the Composite data base and saved to a separate Unmatched Contract data base (Ref. 8).

The final form of the Composite data base is designed to capture the distribution of time it takes from when an applicant first takes the AFQT to when he or she signs a

contract, given that the applicant signs a contract. The test year/month (TYRMO) and the contract year/month (CYRMO) fields both represent fiscal year values, and the new variable (WAIT) represents the number of days between the test date and the contract date. An example of the Composite data base is shown in Table XI below.

SSN	TNBOX	CNBOX	TDATE	TYRMO	TDAY	CDATE	CYRMO	CDAY	WAIT
1425441	GMA	GMA	900722	8910	22	910124	9004	24	186
1429865	GMA	GMA	910409	9007	9	910610	9009	10	62
1464622	GFB	GFA	910618	9009	18	910702	9010	2	14
1480390	OTH	GMB	911007	9101	7	911008	9101	8	1
1480587	SFT	SFT	930517	9208	17	930628	9209	28	42
1481384	GMA	GMA	930514	9208	14	930525	9208	25	11
1485435	SMT	SMT	910716	9010	16	910724	9010	24	8
1486830	GMB	GMA	881118	8802	18	881123	8802	23	5
1487748	SMT	SMT	881205	8803	5	881215	8803	15	10
1500015	GFA	GFA	920804	9111	4	920828	9111	28	24
1500065	OTH	OTH	901025	9001	25	920928	9112	28	704

Table XI. Final Form of the Composite Data Base.

In Table IX, it is important to note the occasional changes in the testing mission box category and the contracting mission box category, which was explained earlier in this chapter. The parameters for the model are derived from the five data bases described in this chapter: 1) Testing data base, 2) Contract data base, 3) Composite data base, 4) Unmatched Testing data base, and 5) Unmatched Contract data base.

V. RESULTS AND RECOMMENDATIONS

Validation of the PEWS model was performed by running two and four week projections into Fiscal Year 1993, the most current year included in the historical data. Though it would have been preferable to use Fiscal Year 1994 data to verify the model's accuracy outside of the historical data set, this data is not yet available from MEPCOM for analysis. In the validation, the model's prediction of the number of contracts expected to be achieved is compared with the actual contracts signed for each month and each bi-weekly period during the base year. This method is chosen to measure the accuracy of the model in predicting contract accomplishment for each time period of the year.

A. VALIDATION RESULTS

1. Monthly Validation Results

Overall, the model tends to under-predict the number of contracts expected to be signed monthly. Figure 4 below shows the percentage difference between the model's estimation and the actual number of contracts signed monthly during FY 1993.

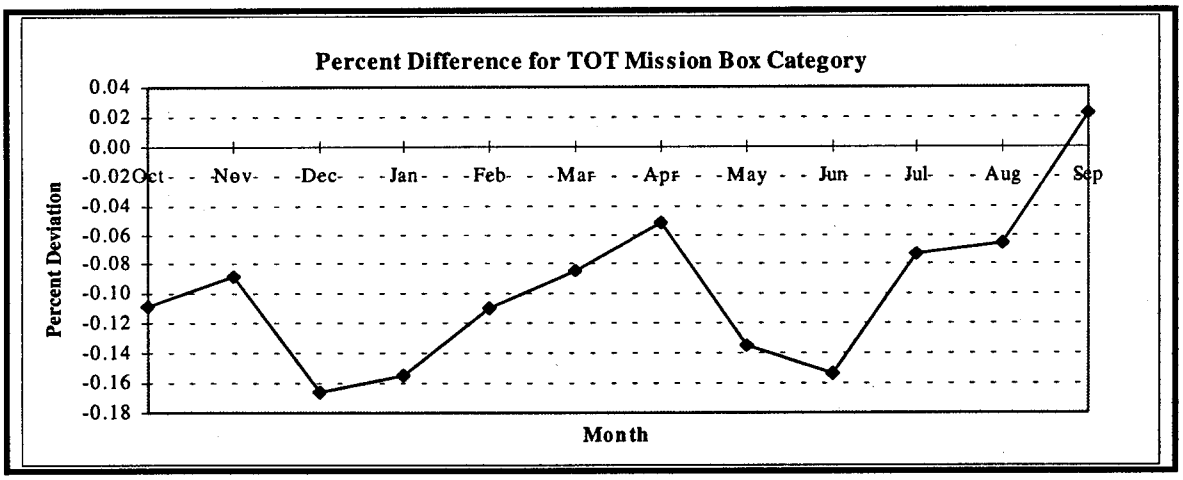


Figure 4. Monthly Percent Deviation of Model Outputs.

The model predicts on the average ten percent fewer contracts than are actuallly realized during any given month, with significant variation (-17% to +3%)

between months. The main reason presumed for the overall underestimation stems from the inability of the model to incorporate the actual distributions of the batch AFQT records. Contracts realized from batch AFQT prospects constitute approximately 40 percent of all contracts. Since their time to contract is expected to be longer that that of regular AFQT prospects, they form a large pool of possible contracts that are not accurately described by the model. An accurate estimation of the time to contract and the probability of signing a contract for these batch AFQT prospects is expected to improve model accuracy.

The variation between months is due mainly to fluctuations in the number of contracts signed from month to month as the seasons and the school year change. A large proportion of contracts is achieved from seniors in high school, and when they sign a contract they generally enter the DEP for longer periods of time than do High School Graduates. Therefore, there is a large influx of seniors who will sign a contract as the school year draws to a close, which is difficult to model effectively given only the prospect's first AFQT test. Other factors which affect monthly contract production but are not included in the model are the "quick fix" changes enacted by USAREC. If production is low for a given month, or if USAREC perceives that they are in jeopardy of not achieving the yearly mission, they often either increase mission quotas or offer additional incentives to prospects earlier in the year to attempt to avoid failing the yearly mission. This type of variation is almost impossible to predict in a systematic manner, since it is largely based on current circumstances and currently available resources which cannot be modeled by the probability distributions in the PEWS model. It is recommended that these trends be further studied using time series analysis, with possible independent variables including seasonal factors, times of the school year, and the historical effects of incentives offered in earlier years.

The same trends are evident for each of the other mission box categories, which are not shown in this chapter. A complete tabulation of the monthly validation results is located in Appendix F.

2. Semi-Monthly Validation Results

The model predicts contract production for the beginning of each month fairly accurately, but grossly underestimates contract production for the last two weeks of each month. Figure 5 below shows the percentage difference between the model's estimation and the actual number of contracts signed monthly during FY 1993.

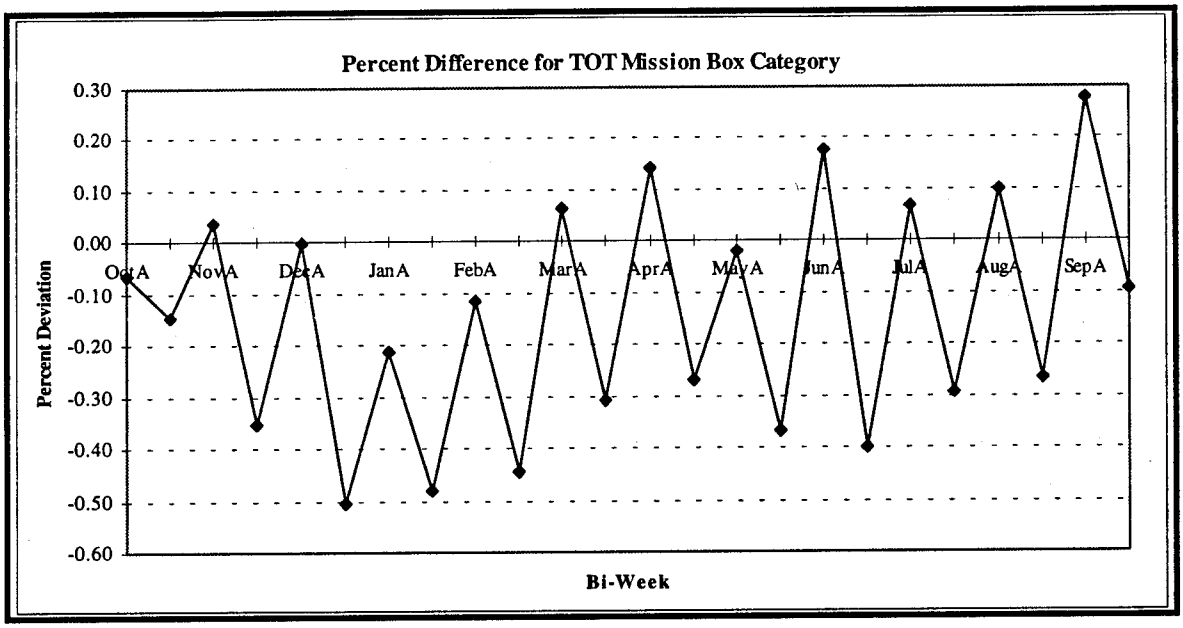


Figure 5. Semi-Monthly Validation Results.

The tendency of the model to underestimate the last half of each month is seen clearly in the spikes between each semi-monthly period. The reasons for this understimation are thought to be related to the model's inability to properly incorporate the distribution of individuals who have taken the AFQT more than once. This was impossible to capture because the MEPCOM data only included the initial AFQT for each applicant. It is expected that an individual who takes an AFQT more than once has a higher probability of signing a contract, and most likely a different time to contract distribution based on his or her most current AFQT. This is correlated to the end of the month increase in contracts because recruiters who have not yet made their monthly mission quota are likely to rely heavily upon encouraging an individual who has already

taken an AFQT in the past to retake the AFQT and hopefully, with a better score, qualify for a better job and sign a contract. Inclusion of this AFQT retest category into the model should greatly increase model accuracy, especially during the last half of the month. Another possible contributor to this discrepancy is that the bi-weekly category is insufficient to accurately describe the increase in contract production at the end of the month, and further division of time periods is required. Time series analysis may help to minimize this trend.

As with the monthly validation results, semi-monthly validation results are similar for all mission box categories. A complete tabulation of these results is located in Appendix F.

B. RECOMMENDATIONS

Three recommendations are suggested for areas of further analysis and model improvement. First, the model should incorporate a "Retest" category that includes all prospects that have taken the AFQT more than once and applies separately derived parameters for their probability to sign a contract and their time to contract distribution. This retest data is stored in the MEPCOM Testing data base, but needs to be specifically requested. Second, the batch testing records should be included as a separate category as well, with their own separately derived parameters for probability to sign a contract and time to contract distribution. These records are also maintained by MEPCOM in a data base that is separate from the regular Testing data base used for this model. Finally, additional time series analysis on seasonal, school year, contracting incentives, and other historical trends should be initiated to determine their impact on contract production. The results of this analysis could be incorporated into the model, or used to change model parameters such as the bi-weekly time periods presently used. Any or all of these areas of further study should increase the effectiveness of the model and improve model performance.

C. CONCLUSIONS

The PEWS model as it is currently defined provides an excellent platform from which to launch a truly effective production prediction model. Deficiencies in the available data severely limit the accuracy of the current model, but USAREC analysts should be able to follow the framework outlined in this thesis to improve on the model with little difficulty. Verification of the model can be accomplished by USAREC analysts upon receipt of MEPCOM Testing data for Fiscal Year 1994, and the model in its current form is sufficient to be used to augment the current production projection system, which is based on qualitative analysis of historical data. As a future analyst for USAREC, the author of this thesis looks forward to the challenge of transforming the current model into an effective and accurate planning tool.

APPENDIX A. PEWS MODEL PASCAL PROGRAM CODE

The code on the following pages is written in TurboPascal. The PEWS Model program is designed to be used on a personal computer. Due to the large number of iterations required for each projection run, suggested computer requirements include a 486/33 or better system with a math co-processor and at least eight megabytes of RAM. To run the program, complete the following steps:

1) Create a directory called C:\PEWS.

2) Save the following files into the C:\PEWS directory:

a. PEWSMAIN.pas	d. CURRDATA.xls	g. CURRDATA.txt
b. PEWSCURR.pas	e. PARADATA.xls	h. PARADATA.txt
c. PEWSPARA.pas	f. AVGDATA.xls	i. AVGDATA.txt

3) Open TurboPascal, and from the *File* menu, *Change Directory* to C:\PEWS.

4) From the *File* menu, *Open* PEWSMAIN.pas, PEWSCURR.pas, and PEWSPARA.pas.

5) From the *Compile* menu, *Make* PEWSCURR.pas and PEWSPARA.pas.

6) From the *Compile* menu, *Build* PEWSMAIN.pas.

7) From the *Run* menu, *Run* PEWSMAIN.pas

8) Follow menu prompts from the program.

1. PASCAL CODE FOR PEWS MODEL MAIN PROGRAM

```
PROGRAM PEWSModel; {PEWSMAIN}

{$M 65520,0,655360}
Uses PEWSPARA, PEWSCURR, CRT;

{-- Type Definitions and Global Variable Declarations ----------}

type MonthTYPE = (OCT,NOV,DEC,JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP);

var Quit : Boolean;
var Dummy : String;
var CurrFY, CurrMo, CurrDay, ProjFY, ProjMo, ProjDay : integer;
var TypeRun, NumDaysProj : integer;
var NBox : MissionBoxTYPE;
var outfile : Text;

{--Procedures and Functions----------------------------------------------}

Function GetTypeRun : Integer;

  { Gets the type of run the user wants to do, dummy proofs it, and returns
    the integer value that corresponds to that type of run. }

  var Dummy : String;
  var Value, INumber, IError : Integer;
  var AnswerOK : Boolean;

  Begin
   repeat
     writeln;writeln;
     writeln('Enter the type of Run to perform.');
     writeln;
     CRT.TextColor(0);
     write(' 1');
     CRT.TextColor(1);
     writeln(' : perform a projection run.');
     CRT.TextColor(0);
     write(' 2');
     CRT.TextColor(1);
     writeln(' : update the number of prospects tested to date and');
     writeln('      the number of contracts signed to date.');
     CRT.TextColor(0);
     write(' 3');
     CRT.TextColor(1);
     writeln(' : print the current number of applicants tested and');
     writeln('      the number of contracts signed to date (by week).');
     CRT.TextColor(0);
```

```
    write(' 4');
    CRT.TextColor(1);
    writeln(' : quit this program.');
    writeln;
    write('Enter your choice: => ');
    readln (Dummy);
    Value := CheckInteger(Dummy);
    If ((Value > 0) AND (Value < 5)) then GetTypeRun := Value
    else begin
         AnswerOK := TRUE;
         CRT.ClrScr;
         writeln;writeln;
         writeln('You must enter one of the choices offered.');
        end; {else stmt}
   until AnswerOK;  {Repeat entire process if input out of range}
  end; {Function GetTypeRun}

{-----------------------------}
Function GetInteger: integer;

 {***Ensure that an input is of integer type***}

var IStr : String;
var IntegerOK : Boolean;
var INumber, IError : Integer;

begin
 readln(IStr);
 repeat
 IntegerOK := True;
  val(IStr,INumber,IError);
  if (IError <> 0) then begin
    write('Please enter a valid integer :> ');
    readln(IStr);
    IntegerOK := False;
  end; {if stmt}
 Until IntegerOK;
 GetInteger := INumber;
end; {Function CheckInteger}

{---------------------------------}
Function GetJulian(Day, Mo : integer):integer;

  {*** Calculates the Julian Date, given the Day, Month and Year ***}

var i,JulianDay : integer;

begin
 JulianDay := 0;
 {i signifies number of months prior to current month}
 For i := 0 to (Mo - 1) do begin
```

```
   if (i=0) then JulianDay := JulianDay + Day;
   if ((i=1) or (i=3) or (i=4) or (i=6) or (i=8) or (i=10) or (i=11)) then
    JulianDay := JulianDay + 31;
   if ((i=2) or (i=7) or (i=9)) then
    JulianDay := JulianDay + 30;
   if (i=5) then JulianDay := JulianDay + 28;
  end; {For loop}
  GetJulian := JulianDay;
end; {Function GetJulian}

{------------------------}
Function GetNBox: MissionBoxTYPE;

var NBOX : String[3];
var i : integer;
GoodNBox : Boolean;

begin
  writeln('Enter the Mission Box you want to use for this projection.');
  repeat
   GoodNBox := True;
   writeln;
   write('(GMA, GMB, GFA, GFB, SMT, SFT, OTH, or TOT) :> ');
   readln(NBOX);
   For i := 1 to 3 do NBOX[i] := UpCase(NBOX[i]);
   If NBOX = 'GMA' then GetNBox := GMA
    else if NBOX = 'GMB' then GetNBox := GMB
     else if NBOX = 'GFA' then GetNBox := GFA
      else if NBOX = 'GFB' then GetNBox := GFB
       else if NBOX = 'SMT' then GetNBox := SMT
        else if NBOX = 'SFT' then GetNBox := SFT
         else if NBOX = 'OTH' then GetNBox := OTH
          else if NBOX = 'TOT' then GetNBox := TOT
           else begin
            writeln;
            writeln('You have entered an invalid Mission Box.');
            GoodNBox := False;
            writeln('Please enter one of the Choices.');
           end; {else stmt}
  until GoodNBox;
end; {Function GetNBox}

{------------------------}
Procedure GetUserInput;

 {*** Gets the Current Date, Projection Cut Off Date, and NBox to use ***}

var DateOK, DateInRange : Boolean;

begin
  DateOK := False;
```

```
 CRT.ClrScr;
 repeat
   DateInRange := True;
{Get Current Date}
   repeat
     writeln;
     writeln('Please enter the current date: ');
     repeat
       write('  Current Day (1, 2, 3, ... , 31) >> ');
       CurrDay := GetInteger;
     until ((CurrDay >= 1) and (CurrDay <= 31));
     repeat
       write('  Current Month (1 = Oct, 2 = Nov, ... , 12 = Sept) >> ');
       CurrMo := GetInteger;
     until ((CurrMo >= 1) and (CurrMo <= 12));
     repeat
       write('  Current Fiscal Year (89, 90. 91, 92, 93, 94 or 95) >> ');
       CurrFY := GetInteger;
     until ((CurrFY >= 88) and (CurrFY <= 95));
     writeln('You have entered ',CurrDay:2,' / ',CurrMo:2,' / ',CurrFY:2,'.');
     write('Is this the correct current date? (Y or N) :> ');
     DateOK := GetAnswer;
     If (CurrMo=2) or (CurrMo=7) or (CurrMo=9) or (CurrMo=12) then begin
       If CurrDay > 30 then begin
         CRT.ClrScr;
         writeln;
         writeln('This month only has 28 days.');
         writeln('Re-Enter the Current Date.');
         DateOK := False;
       end; {inner If stmt}
     end; {outer If stmt}
     If (CurrMo = 5) then begin
       If CurrDay > 28 then begin
         CRT.ClrScr;
         writeln;
         writeln('February only has 30 days.');
         writeln('Re-Enter the Current Date.');
         DateOK := False;
       end; {inner If stmt}
     end; {outer If stmt}
   until DateOK;
{Get Projection Date }
   DateOK := False;
   repeat
     writeln;
     writeln('Please enter the projection cut-off date : ');
     repeat
       write('  Projection Day (1, 2, 3, ... , 31) >> ');
       ProjDay := GetInteger;
     until ((ProjDay >= 1) and (ProjDay <= 31));
     repeat
```

```
    write('  Projection Month (1 = Oct, 2 = Nov, ... , 12 = Sept) >> ');
    ProjMo := GetInteger;
   until ((ProjMo >= 1) and (ProjMo <= 12));
   repeat
    write('  Projection Fiscal Year (89, 90. 91, 92, 93, 94 or 95) >> ');
    ProjFY := GetInteger;
   until ((ProjFY >= 88) and (ProjFY <= 95));
   writeln('You have entered ',ProjDay:2,' / ',ProjMo:2,' / ',ProjFY:2,'.');
   write('Is this the correct projection cut-off date? (Y or N) :> ');
   DateOK := GetAnswer;
   If (ProjMo=2) or (ProjMo=7) or (ProjMo=9) or (ProjMo=12) then begin
    If ProjDay > 30 then begin
     CRT.ClrScr;
     writeln;
     writeln('This month only has 30 days.');
     writeln('Re-Enter the Projection Cut-off Date.');
     DateOK := False;
    end; {inner If stmt}
   end; {outer If stmt}
   If (ProjMo = 5) then begin
    If ProjDay > 28 then begin
     CRT.ClrScr;
     writeln;
     writeln('February only has 30 days.');
     writeln('Re-Enter the Projection Cut-off Date.');
     DateOK := False;
    end; {inner If stmt}
   end; {outer If stmt}
  until DateOK;

 If ProjFY > CurrFY then begin
   CRT.ClrScr;
   writeln;
   writeln('Cannot run projection past the end of the current FY.');
   writeln('Re-enter the current date and the projection date.');
   DateInRange := False;
 end {If stmt}
 Else If ((ProjFY < CurrFY) or (ProjMo < CurrMo)) then begin
   CRT.ClrScr;
   writeln;
   writeln('Projection date must be later than the current date.');
   writeln('Re-Enter the current date and the projection date.');
   DateInRange := False;
 end {Else If stmt}
 Else If ((ProjDay <= CurrDay) and (ProjMo = CurrMo) and (ProjFY = CurrFY))
  then begin
    CRT.ClrScr;
    writeln;
    writeln('Projection date must be later than the current date.');
    writeln('Re-Enter the current date and the projection date.');
    DateInRange := False;
```

```
    end {Else Else If stmt}
    Else If ((ProjMo - CurrMo) > 3) then begin
      CRT.ClrScr;
      writeln;
      write('Projections of more than 3 months are beyond the scope ');
      writeln('of this model.');
      writeln('Re-Enter the current date and a projection date less than ');
      writeln('three months in the future.');
      DateInRange := False;
    end; {Else Else Else If stmt}
  until DateInRange;
  writeln;
  NBox := GetNBox;
  CRT.ClrScr;
  writeln;writeln;writeln;
  writeln('              ...Running Projection ');
end; {Procedure GetUserInput}

{-------------------------------------------------------}
Function GetBiMonth(JulianDay : integer) : BiMonthTYPE;

begin
  If JulianDay <= 15 then GetBiMonth := OCTA
    Else If JulianDay <= 31 then GetBiMonth := OCTB
    Else If JulianDay <= 46 then GetBiMonth := NOVA
    Else If JulianDay <= 61 then GetBiMonth := NOVB
    Else If JulianDay <= 76 then GetBiMonth := DECA
    Else If JulianDay <= 92 then GetBiMonth := DECB
    Else If JulianDay <= 107 then GetBiMonth := JANA
    Else If JulianDay <= 123 then GetBiMonth := JANB
    Else If JulianDay <= 138 then GetBiMonth := FEBA
    Else If JulianDay <= 151 then GetBiMonth := FEBB
    Else If JulianDay <= 166 then GetBiMonth := MARA
    Else If JulianDay <= 182 then GetBiMonth := MARB
    Else If JulianDay <= 197 then GetBiMonth := APRA
    Else If JulianDay <= 212 then GetBiMonth := APRB
    Else If JulianDay <= 227 then GetBiMonth := MAYA
    Else If JulianDay <= 243 then GetBiMonth := MAYB
    Else If JulianDay <= 258 then GetBiMonth := JUNA
    Else If JulianDay <= 273 then GetBiMonth := JUNB
    Else If JulianDay <= 288 then GetBiMonth := JULA
    Else If JulianDay <= 304 then GetBiMonth := JULB
    Else If JulianDay <= 319 then GetBiMonth := AUGA
    Else If JulianDay <= 335 then GetBiMonth := AUGB
    Else If JulianDay <= 350 then GetBiMonth := SEPA
    Else If JulianDay <= 365 then GetBiMonth := SEPB;
end; {Function GetBiMonth}

{-------------------------------------------------------------------}
Function GetTTCDist(DBiMo : BiMonthTYPE; DDays : integer) : real;
```

```
begin
 TTCPtr1 := TTC[ord(NBox)+1].FirstPtr;
 If (ord(TTCPtr1^.BiMo) <> ord(DBiMo)) then begin
  repeat
   TTCPtr1 := TTCPtr1^.Next;
  until ord(TTCPtr1^.BiMo) = ord(DBiMo);
 end; {If stmt}
 If Ddays = 0 then GetTTCDist = TTCPtr1^.day1*0.04
 Else If  DDays = 1 then GetTTCDist := TTCPtr1^.day1*0.96
 Else If  DDays = 2 then GetTTCDist := TTCPtr1^.day2
 Else If  DDays = 3 then GetTTCDist := TTCPtr1^.day3
 Else If  DDays = 4 then GetTTCDist := TTCPtr1^.day4
 Else If  DDays = 5 then GetTTCDist := TTCPtr1^.day5
 Else If  DDays = 6 then GetTTCDist := TTCPtr1^.day6
 Else If  DDays = 7 then GetTTCDist := TTCPtr1^.day7
 Else If  DDays <= 9 then GetTTCDist := (TTCPtr1^.day8to9/2)
 Else If  DDays <= 12 then GetTTCDist := (TTCPtr1^.day10to12/3)
 Else If  DDays <= 14 then GetTTCDist := (TTCPtr1^.day13to14/2)
 Else If  DDays <= 18 then GetTTCDist := (TTCPtr1^.day15to18/4)
 Else If  DDays <= 22 then GetTTCDist := (TTCPtr1^.day19to22/4)
 Else If  DDays <= 30 then GetTTCDist := (TTCPtr1^.day23to30/7)
 Else If  DDays <= 45 then GetTTCDist := (TTCPtr1^.day31to45/15)
 Else If  DDays <= 60 then GetTTCDist := (TTCPtr1^.day46to60/15)
 Else If  DDays <= 120 then GetTTCDist := (TTCPtr1^.day61to120/60)
 Else If  DDays <= 180 then GetTTCDist := (TTCPtr1^.day121to180/60)
 Else If  DDays <= 365 then GetTTCDist := (TTCPtr1^.day181to365/185)
 Else If  DDays <= 730 then GetTTCDist := (TTCPtr1^.day366to730/365)
 Else GetTTCDist := (TTCPtr1^.dayGT731/365);
end; {Function GetTTCDist}

{-------------------------------------------------------------------------}
Function GetHistTested(DBiMo:BiMonthTYPE;FiscalYear,TNBox:integer):real;

var b, NDays : integer;

begin
 b := ord(DBiMo);
 TestContPtr1 := FirstTest[TNBox+1].FirstPtr;
 If TestContPtr1^.BiMo <> DBiMo then begin
  repeat                          {Go to proper BiMonth record}
   TestContPtr1 := TestContPtr1^.Next;
  until TestContPtr1^.BiMo = DBiMo;
 end; {If stmt}
 If (b = 9) then NDays := 13    {Number of days in BiMonth period}
 Else If ((b=1) or (b=5) or (b=7) or (b=11) or (b=15) or (b=19) or (b=21))
    then NDays := 16
 Else NDays := 15;
 If FiscalYear <= 89 then GetHistTested := TestContPtr1^.Num89/NDays;
 If FiscalYear = 90 then GetHistTested := TestContPtr1^.Num90/NDays;
 If FiscalYear = 91 then GetHistTested := TestContPtr1^.Num91/NDays;
 If FiscalYear = 92 then GetHistTested := TestContPtr1^.Num92/NDays;
```

```
 If FiscalYear = 93 then GetHistTested := TestContPtr1^.Num93/NDays;
 If FiscalYear = 94 then GetHistTested := TestContPtr1^.Num94/NDays;
 If FiscalYear >= 95 then GetHistTested := TestContPtr1^.Num95/NDays;
end; {Function GetHistTested}

{-----------------------------------------------}
Function GetPctCont(DBiMo : BiMonthTYPE; TNBox : integer) : Real;

var i : integer;

begin
 i := ord(DBiMo) + 1;
 If TNBox = 0 then GetPctCont := PercentCont[i].GMACont;
 If TNBox = 1 then GetPctCont := PercentCont[i].GMBCont;
 If TNBox = 2 then GetPctCont := PercentCont[i].GFACont;
 If TNBox = 3 then GetPctCont := PercentCont[i].GFBCont;
 If TNBox = 4 then GetPctCont := PercentCont[i].SMTCont;
 If TNBox = 5 then GetPctCont := PercentCont[i].SFTCont;
 If TNBox = 6 then GetPctCont := PercentCont[i].OTHCont;
 If TNBox = 7 then GetPctCont := PercentCont[i].TOTCont;
end; {Function GetPctCont}

{-----------------------------------------------}
Function GetPctUnm(DBiMo : BiMonthTYPE) : Real;

var i : integer;

begin
 i := ord(DBiMo) + 1;
 If NBox = GMA then GetPctUnm := PercentUnm[i].GMAUnm;
 If NBox = GMB then GetPctUnm := PercentUnm[i].GMBUnm;
 If NBox = GFA then GetPctUnm := PercentUnm[i].GFAUnm;
 If NBox = GFB then GetPctUnm := PercentUnm[i].GFBUnm;
 If NBox = SMT then GetPctUnm := PercentUnm[i].SMTUnm;
 If NBox = SFT then GetPctUnm := PercentUnm[i].SFTUnm;
 If NBox = OTH then GetPctUnm := PercentUnm[i].OTHUnm;
 If NBox = TOT then GetPctUnm := PercentUnm[i].TOTUnm;
end; {Function GetPctUnm}

{-----------------------------------------------}
Function GetDaysInMo(Mo : integer) : integer;

begin
 If (Mo=1) or (Mo=3) or (Mo=4) or (Mo=6) or (Mo=8) or (Mo=10) or (Mo=11)
   then GetDaysInMo := 31;
 If (Mo=2) or (Mo=7) or (Mo=9) or (Mo=12)
   then GetDaysInMo := 30;
 If (Mo=5) then GetDaysInMo := 28;
end; {Function GetDaysInMo}

{-----------------------------------------------}
```

```
Function GetCurrMo(Julian : integer) : integer;

{ Determine what month a given julian day falls into. }

begin
 If Julian <= 31 then GetCurrMo := 1
  Else If Julian <= 61 then GetCurrMo := 2
   Else If Julian <= 92 then GetCurrMo := 3
    Else If Julian <= 123 then GetCurrMo := 4
     Else If Julian <= 151 then GetCurrMo := 5
      Else If Julian <= 182 then GetCurrMo := 6
       Else If Julian <= 212 then GetCurrMo := 7
        Else If Julian <= 243 then GetCurrMo := 8
         Else If Julian <= 273 then GetCurrMo := 9
          Else If Julian <= 304 then GetCurrMo := 10
           Else If Julian <= 335 then GetCurrMo := 11
            Else GetCurrMo := 12;
end; {Function GetCurrMo}


{-----------------------------------------------}
Function GetDayOfMo(Julian : integer) : integer;

{Given a Julian Day, computes the corresponding day of the month.}

begin
 If Julian <= 31 then GetDayOfMo := Julian
 Else If Julian <= 61 then GetDayOfMo := (Julian - 31)
 Else If Julian <= 92 then GetDayOfMo := (Julian - 61)
 Else If Julian <= 123 then GetDayOfMo := (Julian - 92)
 Else If Julian <= 151 then GetDayOfMo := (Julian - 123)
 Else If Julian <= 182 then GetDayOfMo := (Julian - 151)
 Else If Julian <= 212 then GetDayOfMo := (Julian - 182)
 Else If Julian <= 243 then GetDayOfMo := (Julian - 212)
 Else If Julian <= 273 then GetDayOfMo := (Julian - 243)
 Else If Julian <= 304 then GetDayOfMo := (Julian - 273)
 Else If Julian <= 335 then GetDayOfMo := (Julian - 304)
 Else If Julian <= 365 then GetDayOfMo := (Julian - 335);
end; {Function GetDayOfMo}

{------------------------}
Function GetDailyTested(Julian, Mo : integer) : integer;

{Given a julian day and a month, retrieves the actual number of prospects
 tested from the DailyAvg data base.}

var DayOfMonth : integer;

begin
 DayOfMonth := GetDayOfMo(Julian);
 MoDayPtr1 := DailyAvg[Mo].FirstPtr;
```

```
 If (DayOfMonth > 1) then begin
    repeat
      MoDayPtr1 := MoDayPtr1^.Next;
    until (MoDayPtr1^.Day = DayOfMonth);
 end; {if stmt}
 GetDailyTested := MoDayPtr1^.NumTest;
end; {Function GetDailyTested}

{------------------------}
Procedure RunProjection;

{*** Projects the number of contracts expected to be accomplished by
   a specified date.                                    ***}

var ProjJulian, CurrJulian, TDay, PDay, i, DailyTested : integer;
var DayOfWeek, DistDays, AdjFY, StopDay, TBox : integer;
var SumContracts, PctDist, PctCont, PctUnm, NumContracts : real;
var NumHistTest, PctNBox, CurrCont, ProjCont, PredCont : real;
var DistBiMo : BiMonthTYPE;

begin
 GetUserInput;
 CurrJulian := GetJulian(CurrDay, CurrMo);
 ProjJulian := GetJulian(ProjDay, ProjMo);
 NumDaysProj := ProjJulian - CurrJulian;

{Sum up current # of cont's signed in CurrFY up to beg of CurrMonth}

 TBox := 0;
 SumContracts := 0.0;
 NumContracts := 0.0;
 TestContPtr1 := FirstCont[ord(NBox)+1].FirstPtr;
 For i := 1 to ((CurrMo-1)*2) do begin
  if (CurrFY <= 89) then NumContracts := TestContPtr1^.Num89;
  if (CurrFY = 90) then NumContracts := TestContPtr1^.Num90;
  if (CurrFY = 91) then NumContracts := TestContPtr1^.Num91;
  if (CurrFY = 92) then NumContracts := TestContPtr1^.Num92;
  if (CurrFY = 93) then NumContracts := TestContPtr1^.Num93;
  if (CurrFY = 94) then NumContracts := TestContPtr1^.Num94;
  if (CurrFY >= 95) then NumContracts := TestContPtr1^.Num95;
  SumContracts := SumContracts + NumContracts;
  TestContPtr1 := TestContPtr1^.Next;
 end; {For stmt}

    {Now add the # of cont's signed daily in the CurrMonth
          (up to and including the Current Day)}

 MoDayPtr1 := DailyAvg[CurrMo].FirstPtr; {ptr to 1st of CurrMonth}
 For i := 1 to CurrDay do begin   {add up # of contracts achieved daily}
  NumContracts := MoDayPtr1^.NumCont*PerCentNBox[ord(NBox)+1];
  SumContracts := SumContracts + NumContracts;
```

```
  MoDayPtr1 := MoDayPtr1^.Next;
 end; {For loop}
 CurrCont := SumContracts;

  {Now estimate the number of contracts expected to be signed during
   the projection period (from day 2 years prior thru current day) }

 For PDay := 1 to NumDaysProj do begin
 {loop thru each day of proj to calc # of cont's exp for each of these days}

  For TDay := CurrJulian to 365 do begin
  {from day 2 years prior to current day thru end of that FY}
   DistDays := (730 + CurrJulian - TDay) + PDay;
   DistBiMo := GetBiMonth(TDay);
   PctDist := GetTTCDist(DistBiMo, DistDays);
   AdjFY := CurrFY - 2;
   PctUnm := GetPctUnm(DistBiMo);
   PctNBox := PerCentNBox[ord(NBox)+1];
   If NBox = TOT then begin
    TBox := 7;
    NumHistTest := GetHistTested(DistBiMo, AdjFY, TBox);
    PctCont := GetPctCont(DistBiMo, TBox);
NumContracts := (NumHistTest)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
    SumContracts := SumContracts + NumContracts;
   end {If Stmt}
   Else begin
    For TBox := 0 to 6 do begin     {Loop thru 7 Mission Box categories}
     NumHistTest := GetHistTested(DistBiMo, AdjFY, TBox);
     PctCont := GetPctCont(DistBiMo, TBox);
NumContracts := (NumHistTest)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
     SumContracts := SumContracts + NumContracts;
    end; {For TBox loop}
   end; {Else}
  end; {For TDay loop 1}

  For TDay := 1 to 365 do begin
  {entire FY prior to current FY}
   DistDays := (365 + CurrJulian - TDay) + PDay;
   DistBiMo := GetBiMonth(TDay);
   PctDist := GetTTCDist(DistBiMo, DistDays);
   AdjFY := CurrFY - 1;
   PctUnm := GetPctUnm(DistBiMo);
   PctNBox := PerCentNBox[ord(NBox)+1];
   If NBox = TOT then begin
    TBox := 7;
    NumHistTest := GetHistTested(DistBiMo, AdjFY, TBox);
    PctCont := GetPctCont(DistBiMo, TBox);
NumContracts := (NumHistTest)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
    SumContracts := SumContracts + NumContracts;
   end {If Stmt}
   Else begin
```

```
      For TBox := 0 to 6 do begin     {Loop thru 7 Mission Box categories}
       NumHistTest := GetHistTested(DistBiMo, AdjFY, TBox);
       PctCont := GetPctCont(DistBiMo, TBox);
NumContracts := (NumHistTest)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
       SumContracts := SumContracts + NumContracts;
      end; {For TBox loop}
     end; {Else}
   end; {For TDay loop 2}

   StopDay := (CurrJulian - CurrDay);
   {StopDay is last day of month prior to current mo}

   For TDay := 1 to StopDay do begin
   {from beginning of current fiscal year thru StopDay}
    DistDays := (CurrJulian - TDay) + PDay;
    DistBiMo := GetBiMonth(TDay);
    PctDist := GetTTCDist(DistBiMo, DistDays);
    AdjFY := CurrFY;
    PctUnm := GetPctUnm(DistBiMo);
    PctNBox := PerCentNBox[ord(NBox)+1];
    If NBox = TOT then begin
     TBox := 7;
     NumHistTest := GetHistTested(DistBiMo, AdjFY, TBox);
     PctCont := GetPctCont(DistBiMo, TBox);
NumContracts := (NumHistTest)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
     SumContracts := SumContracts + NumContracts;
    end {If Stmt}
    Else begin  {Need # test days from 7 Msn cats, each w/ diff PctCont}
     For TBox := 0 to 6 do begin     {Loop thru 7 Mission Box categories}
      NumHistTest := GetHistTested(DistBiMo, AdjFY, TBox);
      PctCont := GetPctCont(DistBiMo, TBox);
NumContracts := (NumHistTest)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
       SumContracts := SumContracts + NumContracts;
      end; {For TBox loop}
     end; {Else}
   end; {For TDay loop 3}

   For TDay := (StopDay + 1) to CurrDay do begin
   {from first day of current month thru day before current date}
    DistDays := (CurrJulian - TDay) + PDay;
    DistBiMo := GetBiMonth(TDay);
    PctDist := GetTTCDist(DistBiMo, DistDays);
    DailyTested := GetDailyTested(TDay, CurrMo);
    TBox := 7; {Must sample only from total column, due to daily data}
    PctCont := GetPctCont(DistBiMo, TBox);
    PctUnm := GetPctUnm(DistBiMo);
    PctNBox := PerCentNBox[ord(NBox)+1];
NumContracts := (DailyTested)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
    SumContracts := SumContracts + NumContracts;
   end; {For TDay loop 4}
  end; {For PDay loop 1}
```

```
ProjCont := SumContracts - CurrCont;

For PDay := (CurrJulian + 1) to ProjJulian do begin
{Project into day after current day to stop projection day}
  For TDay := CurrJulian to PDay do begin
  {From current day to day prior to PDay}
    DistDays := PDay - TDay;
    DistBiMo := GetBiMonth(TDay);
    PctDist := GetTTCDist(DistBiMo, DistDays);
    DailyTested := GetDailyTested(TDay, GetCurrMo(TDay));
    TBox := 7;    {Because Daily Tested is all NBox combined}
    PctCont := GetPctCont(DistBiMo, TBox);
    PctUnm := GetPctUnm(DistBiMo);
    PctNBox := PerCentNBox[ord(NBox)+1];
NumContracts := (DailyTested)*(PctCont)*(PctDist)*(PctUnm)*(PctNBox);
    SumContracts := SumContracts + NumContracts;
  end; {For TDay loop 5}
end; {For PDay loop 2}
PredCont := SumContracts - CurrCont - ProjCont;

writeln;writeln;
writeln('Output from your Projection Run is as follows:');
writeln;
writeln('Current Date  : ',CurrDay:2,' / ',CurrMo:2,' / ',CurrFY:2);
writeln('Projected Date: ',ProjDay:2,' / ',ProjMo:2,' / ',ProjFY:2);
writeln;
writeln('NBox category tested: ',(Ord(NBox)+1):2);
write('Number of days in projection (ProjDay - CurrDay) : ');
writeln(NumDaysProj:5);
write('Number of contracts actually signed as of current date : ');
writeln(CurrCont:8:2);
write('Number of contracts projected during projection period : ');
writeln(ProjCont:8:2);
write('Number of contracts predicted during projection period : ');
writeln(PredCont:8:2);
writeln;
write('Total number of contracts expected as of Projected Date: ');
writeln(SumContracts:8:2);
writeln;
writeln('Results of projection runs are printed to "C:\PEWS\PEWSOut.txt"');
writeln;writeln;
write('Press Enter to continue :> ');
Readln(Dummy);

write(outfile,' ',(Ord(NBox)+1):2);
write(outfile,'  ',CurrDay:2,'/',CurrMo:2,'/',CurrFY:2);
write(outfile,'  ',ProjDay:2,'/',ProjMo:2,'/',ProjFY:2);
write(outfile,' ',NumDaysProj:6);
write(outfile,'    ',CurrCont:8:2);
write(outfile,'   ',ProjCont:8:2);
write(outfile,'   ',PredCont:8:2);
```

```
 writeln(outfile,'    ',SumContracts:8:2);

 CRT.ClrScr;
end; {Procedure RunProjection}

{--Main Program-----------------------------------------------------------}

Begin
 assign(outfile,'C:\PEWS\PEWSOut.txt');
 rewrite(outfile);
 write(outfile,'PEWSOut.txt logs all projections performed during a ');
 writeln(outfile,'session of projection runs.');
 writeln(outfile);
 write(outfile,'NBox  CurrDate   ProjDate  #DaysProj  CurrCont');
 writeln(outfile,'   ProjCont    PredCont    SumCont');
 write(outfile,'-----------------------------------------------------');
 writeln(outfile,'------------------------------');
 writeln(outfile);

 CRT.ClrScr;
 CRT.TextColor(1);
 CRT.TextBackground(7);
 writeln;
 GetParameters;
 GetCurrentData;
 GetDailyAvgData;
 Quit := False;
 Repeat

  TypeRun := GetTypeRun;
  CASE TypeRun of
     1 : RunProjection;
     2 : UpdateCurrentData;
     3 : begin
          PrintCurrentData;
          write('Press return to continue :> ');
          readln(Dummy);
          CRT.ClrScr;
         end; {Case 3 stmt}
     4 : Quit := True;
  end;   {CASE Stmt}

 Until Quit;   {End of REPEAT stmt}
 CRT.ClrScr;
 writeln;writeln;writeln;
 PrintParameters;
 PrintCurrentData;
 PrintDailyAvgData;
 writeln;writeln;
 writeln(' Thank you for using the PEWS Model.');
 writeln;
```

```
  write('       ...Press RETURN to return to DOS :> ');
  readln(Dummy);

  close(outfile);
END.  {PROGRAM PEWSMODEL}
```

2. PASCAL CODE FOR PEWS MODEL CURRENT DATA UNIT

```
unit PEWSCURR;

{-------------------------------------------------------------------------}
interface

uses CRT;

type MissionBoxTYPE = (GMA,GMB,GFA,GFB,SMT,SFT,OTH,TOT);

type MonthTYPE = (OCT,NOV,DEC,JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP);

type BiMonthTYPE = (OCTA,OCTB,NOVA,NOVB,DECA,DECB,JANA,JANB,
            FEBA,FEBB,MARA,MARB,APRA,APRB,MAYA,MAYB,
            JUNA,JUNB,JULA,JULB,AUGA,AUGB,SEPA,SEPB);

type TestContPOINT = ^NumberTestContTYPE;
   NumberTestContTYPE = Record
                  Next : TestContPOINT;
                  BiMo : BiMonthTYPE;
                  Num89 : integer;
                  Num90 : integer;
                  Num91 : integer;
                  Num92 : integer;
                  Num93 : integer;
                  Num94 : integer;
                  Num95 : integer;
                 end;

type FirstNodeTYPE = Record
                FirstPtr : TestContPOINT;
                NBox : MissionBoxTYPE;
               end;

type DailyAvgPOINT = ^DailyAverageTYPE;
   DailyAverageTYPE = Record
                 Next : DailyAvgPOINT;
                 Day : integer;
                 NumTest : integer;
                 NumCont : integer;
                end;

type FirstDailyAvgTYPE = Record
                  FirstPtr : DailyAvgPOINT;
                  Month : MonthTYPE;
                  DaysPerMo : integer;
                 end;

VAR FirstTest : Array[1..8] of FirstNodeTYPE;
```

```
    FirstCont : Array[1..8] of FirstNodeTYPE;
    TestContPtr1,TestContPtr2 : TestContPOINT;
    DailyAvg : Array[1..12] of FirstDailyAvgTYPE;
    MoDayPtr1, MoDayPtr2 : DailyAvgPOINT;
    PerCentNBox : Array[1..8] of Real;
    HalfMonth : BiMonthTYPE;

Function CheckInteger(InStr : string):Integer;

Function CheckBiMo:BiMonthTYPE;

Function CheckFiscalYear:Integer;

Function GetAnswer:Boolean;

Procedure GetCurrentData;

Procedure GetDailyAvgData;

Procedure UpdateCurrentData;

Procedure PrintCurrentData;

Procedure PrintDailyAvgData;
{-------------------------------------------------------------------------}
implementation

{-------------------------}
Procedure GetCurrentData;

  {*** Reads current number of AFQT tests taken and current number of
     Contracts signed by BiMonth from FY89 through FY95.  Reads from a
     flat file and saves into a linked list.                    ***}

 var ParamFile, DummyStr : String;
 var DummyStr5 : String[5];
 var i,j,DummyInt : integer;
 var MissionBox : MissionBoxTYPE;

 begin
  write('Current Contract and Testing data is read from textfile ');
  writeln('"C:\PEWS\CurrData.txt."');
  ParamFile := 'C:\PEWS\CurrData.txt';
  Assign(Input,ParamFile);
  Reset(Input);

        {First fill out the Testing Linked List Array}
  MissionBox := GMA;
  For i := 1 to 8 do begin    {Loop through 8 Mission Box categories}
   Readln(DummyStr);
   Readln(DummyStr);  {Read past 2 lines of explanatory text headings}
```

```
 FirstTest[i].NBox := MissionBox;
 new(FirstTest[i].FirstPtr);  {Initialize first pointer}
 FirstTest[i].FirstPtr^.Next := nil;
 TestContPtr1 := FirstTest[i].FirstPtr;
 HalfMonth := OCTA;
 For j := 1 to 24 do begin   {Loop through BiMos of the year}
  read(DummyStr5);
  TestContPtr1^.BiMo := HalfMonth;
  read(TestContPtr1^.Num89);
  read(TestContPtr1^.Num90);
  read(TestContPtr1^.Num91);
  read(TestContPtr1^.Num92);    {Read in number tested for BiMo j}
  read(TestContPtr1^.Num93);
  read(TestContPtr1^.Num94);
  readln(TestContPtr1^.Num95);
  If (j < 24) then begin     {Don't add bogus record at end of list}
   new(TestContPtr1^.Next);
   TestContPtr1 := TestContPtr1^.Next;
  end; {If stmt}
  TestContPtr1^.Next := nil;
  HalfMonth := succ(HalfMonth);
 end; {For j loop}
 MissionBox := succ(MissionBox);
 readln(DummyStr);     {Read past blank space separating NBOX Cat}
end; {For i loop}

      {Now fill out the Contracted Linked List Array}
MissionBox := GMA;
For i := 1 to 8 do begin   {Loop through 8 Mission Box categories}
 Readln(DummyStr);
 Readln(DummyStr);  {Read past 2 lines of explanatory text headings}
 FirstCont[i].NBox := MissionBox;
 new(FirstCont[i].FirstPtr);  {Initialize first pointer}
 FirstCont[i].FirstPtr^.Next := nil;
 TestContPtr1 := FirstCont[i].FirstPtr;
 HalfMonth := OCTA;
 For j := 1 to 24 do begin   {Loop through BiMos of the year}
  read(DummyStr5);
  TestContPtr1^.BiMo := HalfMonth;
  read(TestContPtr1^.Num89);
  read(TestContPtr1^.Num90);
  read(TestContPtr1^.Num91);
  read(TestContPtr1^.Num92);  {Read in number of contracted for BiMo j}
  read(TestContPtr1^.Num93);
  read(TestContPtr1^.Num94);
  readln(TestContPtr1^.Num95);
  If (j < 24) then begin     {Don't add bogus record at end of list}
   new(TestContPtr1^.Next);
   TestContPtr1 := TestContPtr1^.Next;
  end; {If stmt}
  TestContPtr1^.Next := nil;
```

```
    HalfMonth := succ(HalfMonth);
   end; {For j loop}
   MissionBox := succ(MissionBox);  {Increment Mission Box}
   readln(DummyStr);     {Read past blank space separating NBOX Cat}
  end; {For i loop}

 close(input);
 assign(input,'con');
 reset(input);
 end;  {Procedure GetCurrentData}

{-------------------------}
Procedure GetDailyAvgData;

  {*** Reads avg daily number of AFQT tests taken and avg daily
     number of contracts signed.  Reads from a flat file
     and saves into a linked list.                    ***}

 var ParamFile, DummyStr : String;
 var DummyStr4 : String[4];
 var i,j,DaysInMo : integer;
 var Month : MonthTYPE;


 begin
  write('Daily Avg Contract and Testing data is read from textfile ');
  writeln('"C:\PEWS\AvgData.txt."');
  ParamFile := 'C:\PEWS\AvgData.txt';
  Assign(Input,ParamFile);
  Reset(Input);

      {First read in the number of prospects tested daily}
  Month := Oct;
  readln(DummyStr);      {two lines of explanatory text}
  readln(DummyStr);
  For i := 1 to 12 do begin   {For each month of the year}
   DailyAvg[i].Month := Month;
   new(DailyAvg[i].FirstPtr);
   DailyAvg[i].FirstPtr^.Next := nil;
   MoDayPtr1 := DailyAvg[i].FirstPtr;
   read(DummyStr4);         {Month text}
   read(DaysInMo);
   DailyAvg[i].DaysPerMo := DaysInMo;
  For j := 1 to 8 do begin    {read first 8 days of the month}
   MoDayPtr1^.Day := j;
   read(MoDayPtr1^.NumTest);
   new(MoDayPtr1^.Next);
   MoDayPtr1 := MoDayPtr1^.Next;
   MoDayPtr1^.Next := nil;
  end; {For j = 1 thru 8 loop}
  readln(DummyStr);    {read to end of line and return}
```

```
  For j := 9 to 16 do begin   {read 9th thru 16th days of the month}
    MoDayPtr1^.Day := j;
    read(MoDayPtr1^.NumTest);
    new(MoDayPtr1^.Next);
    MoDayPtr1 := MoDayPtr1^.Next;
    MoDayPtr1^.Next := nil;
  end; {For j = 9 thru 16 loop}
  readln(DummyStr);   {read to end of line and return}
  For j := 17 to 24 do begin   {read 17th thru 24th days of the month}
    MoDayPtr1^.Day := j;
    read(MoDayPtr1^.NumTest);
    new(MoDayPtr1^.Next);
    MoDayPtr1 := MoDayPtr1^.Next;
    MoDayPtr1^.Next := nil;
  end; {For j = 17 thru 24 loop}
  readln(DummyStr);   {read to end of line and return}
  For j := 25 to DaysInMo do begin  {read 25th thru last day of the month}
    MoDayPtr1^.Day := j;
    read(MoDayPtr1^.NumTest);
    If (j < DaysInMo) then begin   {Don't add bogus record at end of list}
      new(MoDayPtr1^.Next);
      MoDayPtr1 := MoDayPtr1^.Next;
    end; {If stmt}
    MoDayPtr1^.Next := nil;
  end; {For j = 25 thru 24 loop}
  readln(DummyStr);   {read to end of line and return}
  Month := succ(Month);
end; {For i loop}

      {Now read in the number of contracts signed daily}
  Month := Oct;
  readln(DummyStr); {Blank line of text}
  readln(DummyStr);      {two lines of explanatory text}
  readln(DummyStr);
  For i := 1 to 12 do begin   {For each month of the year}
    MoDayPtr1 := DailyAvg[i].FirstPtr;
    read(DummyStr4);         {Month text}
    read(DaysInMo);
    DailyAvg[i].DaysPerMo := DaysInMo;
  For j := 1 to 8 do begin    {read first 8 days of the month}
    read(MoDayPtr1^.NumCont);
    MoDayPtr1 := MoDayPtr1^.Next;
  end; {For j = 1 thru 8 loop}
  readln(DummyStr);   {read to end of line and return}
  For j := 9 to 16 do begin   {read 9th thru 16th days of the month}
    read(MoDayPtr1^.NumCont);
    MoDayPtr1 := MoDayPtr1^.Next;
  end; {For j = 9 thru 16 loop}
  readln(DummyStr);   {read to end of line and return}
  For j := 17 to 24 do begin   {read 17th thru 24th days of the month}
    read(MoDayPtr1^.NumCont);
```

```
    MoDayPtr1 := MoDayPtr1^.Next;
   end; {For j = 17 thru 24 loop}
   readln(DummyStr);   {read to end of line and return}
   For j := 25 to DaysInMo do begin {read 25th thru last day of the month}
    read(MoDayPtr1^.NumCont);
    If (j < DaysInMo) then begin
     MoDayPtr1 := MoDayPtr1^.Next;
    end; {If stmt}
   end; {For j = 25 thru 24 loop}
   readln(DummyStr);   {read to end of line and return}
   Month := succ(Month);
  end; {For i loop}

 {Finally, read in the Percentage of daily Contracts that fall in each NBox}
  readln(DummyStr);
  readln(DummyStr);
  readln(DummyStr);
  For i := 1 to 8 do begin
   read(DummyStr4);
   readln(PercentNBox[i]);
  end; {For i loop}

  close(input);
  assign(input,'con');
  reset(input);
  end;  {Procedure GetDailyAvgData}

{-----------------------------------------}
Function CheckInteger(InStr : string):integer;

  {***Ensure that an input is of integer type***}

var IStr : String;
var IntegerOK : Boolean;
var INumber, IError : Integer;

begin
  IStr := InStr;
  repeat
  IntegerOK := True;
   val(IStr,INumber,IError);
   if (IError <> 0) then begin
    write('Please enter a valid integer :> ');
    readln(IStr);
    IntegerOK := False;
   end; {if stmt}
  Until IntegerOK;
  CheckInteger := INumber;
end; {Function CheckInteger}

{-----------------------------}
```

```
Function CheckBiMo:BiMonthTYPE;

  {*** Ensure that the week input is a valid week of the year. ***}

var BiMoOK : Boolean;
var Dummy4 : String[4];
var i : integer;

begin
 repeat
  BiMoOK := True;
  writeln('Choices: OctA, OctB, NovA, NovB, DecA, DecB, JanA, JanB,');
  writeln('         FebA, FebB, MarA, MarB, AprA, AprB, MayA, MayB,');
  writeln('         JunA, JunB, JulA, JulB, AugA, AugB, SepA, SepB.');
  write('Your Choice :> ');
  readln(Dummy4);
  For i := 1 to 4 do Dummy4[i] := UpCase(Dummy4[i]);
  If Dummy4 = 'OCTA' then CheckBiMo := OCTA
   else If Dummy4 = 'OCTB' then CheckBiMo := OCTB
    else If Dummy4 = 'NOVA' then CheckBiMo := NOVA
    else If Dummy4 = 'NOVB' then CheckBiMo := NOVB
     else If Dummy4 = 'DECA' then CheckBiMo := DECA
     else If Dummy4 = 'DECB' then CheckBiMo := DECB
      else If Dummy4 = 'JANA' then CheckBiMo := JANA
      else If Dummy4 = 'JANB' then CheckBiMo := JANB
       else If Dummy4 = 'FEBA' then CheckBiMo := FEBA
       else If Dummy4 = 'FEBB' then CheckBiMo := FEBB
        else If Dummy4 = 'MARA' then CheckBiMo := MARA
        else If Dummy4 = 'MARB' then CheckBiMo := MARB
         else If Dummy4 = 'APRA' then CheckBiMo := APRA
         else If Dummy4 = 'APRB' then CheckBiMo := APRB
          else If Dummy4 = 'MAYA' then CheckBiMo := MAYA
          else If Dummy4 = 'MAYB' then CheckBiMo := MAYB
           else If Dummy4 = 'JUNA' then CheckBiMo := JUNA
           else If Dummy4 = 'JUNB' then CheckBiMo := JUNB
            else If Dummy4 = 'JULA' then CheckBiMo := JULA
            else If Dummy4 = 'JULB' then CheckBiMo := JULB
             else If Dummy4 = 'AUGA' then CheckBiMo := AUGA
             else If Dummy4 = 'AUGB' then CheckBiMo := AUGB
              else If Dummy4 = 'SEPA' then CheckBiMo := SEPA
              else If Dummy4 = 'SEPB' then CheckBiMo := SEPB
               else begin
                 writeln('You have entered an invalid BiMonth Category.');
                 BiMoOK := False;
                 writeln('Please enter one of the choices.');
               end; {else stmt}
 until BiMoOK;
end; {Function CheckBiMo}

{--------------------------}
Function CheckFiscalYear:integer;
```

```
  {*** Ensure that the input value is a valid Fiscal Year. ***}

var FYOK : Boolean;
var Dummy : String;
var FYValue : integer;

begin
 FYOK := False;
 repeat
  readln(Dummy);
  FYValue := CheckInteger(Dummy);
  If ((FYValue=89) or (FYValue=90) or (FYValue=91) or (FYValue=92) or
    (FYValue=93) or (FYValue=94) or (FYValue=95)) then FYOK := True
  else
   begin
    writeln('You have entered a value outside the accepted range.');
    write('Enter new value (89, 90, 91, 92, 93, 94, or 95) :> ');
   end {else part of stmt}
 until FYOK;
 CheckFiscalYear := FYValue;
end; {Function CheckFiscalYear}

{------------------------------}
Function GetAnswer : Boolean;

var Answer : Char;
var Again : Boolean;

begin
 repeat
  Again := False;
  readln (Answer);
  case Answer of
   'Y', 'y' : GetAnswer := True;
   'N', 'n' : GetAnswer := False;
   Else begin
        Again := True;
        write ('I need a Yes or a No answer. :> ');
   end; {Else Stmt}
  end; {Case Stmt}
 until Again = False;
end; {Function Get Answer}



{-------------------------------------------}
Procedure MakeChange(MBox : MissionBoxType);

 {***Changes the Number Tested and the Number Contracted for a given
    week and Fiscal Year.                          ***}
```

```
var i, ChgBiMo, ChgFY, NewTested, NewContracted : integer;
var Dummy : String;
var TestPtr, ContPtr : TestContPOINT;

begin
 writeln('Enter the BiMonth category (OCTA, OCTB ... SEPB) to update.');
 ChgBiMo := ord(CheckBiMo);
 write('Enter the FY (89, 90, 91, 92, 93, 94, 95) to update :> ');
 ChgFY := CheckFiscalYear;
 write('Enter the number Tested in this BiMonth/year :> ');
 readln(Dummy);
 NewTested := CheckInteger(Dummy);
 write('Enter the number Contracted in this BiMonth/year : ');
 readln(Dummy);
 NewContracted := CheckInteger(Dummy);

 i := ord(MBox) + 1;              {Because ord(MBox) starts with 0}
 TestPtr := FirstTest[i].FirstPtr;    {Initialize pointer to 1st record}
 While ord(TestPtr^.BiMo) < ChgBiMo do    {Move pointer}
  begin
   TestPtr := TestPtr^.Next;
  end; {While test loop}
 ContPtr := FirstCont[i].FirstPtr;
 While ord(ContPtr^.BiMo) < ChgBiMo do
  begin
   ContPtr := ContPtr^.Next;
  end; {While Cont loop}

 CASE ChgFY of                {Update number Tested and Contracted}
  89 : begin                  {for that Fiscal Year and Week}
       TestPtr^.Num89 := NewTested;
       ContPtr^.Num89 := NewContracted;
      end; {Case 89}
  90 : begin
       TestPtr^.Num90 := NewTested;
       ContPtr^.Num90 := NewContracted;
      end; {Case 90}
  91 : begin
       TestPtr^.Num91 := NewTested;
       ContPtr^.Num91 := NewContracted;
      end; {Case 91}
  92 : begin
       TestPtr^.Num92 := NewTested;
       ContPtr^.Num92 := NewContracted;
      end; {Case 92}
  93 : begin
       TestPtr^.Num93 := NewTested;
       ContPtr^.Num93 := NewContracted;
      end; {Case 93}
  94 : begin
```

```
        TestPtr^.Num94 := NewTested;
        ContPtr^.Num94 := NewContracted;
       end; {Case 94}
   95 : begin
        TestPtr^.Num95 := NewTested;
        ContPtr^.Num95 := NewContracted;
       end  {Case 95}
   Else writeln('Fiscal Year was out of expected range.  No change made.');
  end; {CASE FY stmt}
  write('Changes have been made to BiMonth ',(ChgBiMo+1):2,' of ');
  writeln('Fiscal Year ',ChgFY:2);
end; {Procedure MakeChange}

{--------------------------}
Procedure UpdateCurrentData;

 {***Allows the user to update the number of applicants who took the
    AFQT and the number of prospects who signed contracts weekly.  ***}

var Dummy  : String;
var ChgNBox : String[3];
var NewBox : MissionBoxTYPE;
var StopWeek, StopUpdate, GoodNBox : Boolean;
var i : integer;

begin
 CRT.ClrScr;
 writeln('This procedure allows you to update the number of prospects');
 writeln(' who took the AFQT in a given week, and the actual number of');
 writeln(' contracts signed in a given week.  These changes are only ');
 writeln(' local within this run.  To make permanent changes to the ');
 writeln(' parameters, it is necessary to make them in the EXCEL source');
 writeln(' file and resave it as a tab-delimited text file.');
 writeln;
 StopUpdate := False;
 repeat
  writeln('Enter the mission box category that you wish to update:');
  writeln('Choices: GMA, GMB, GFA, GFB, SMT, SFT, OTH, TOT.');
  write('Entering STP will quit this operation.  Your choice? :> ');
  readln(ChgNBox);
  repeat
   For i := 1 to 3 do     {Make ChgNBox all uppercase}
    ChgNBox[i] := UpCase(ChgNBox[i]);
   GoodNBox := True;      {Assume NBox is good until proven otherwise}
   if ChgNBox = 'STP' then StopUpdate := True
     else if ChgNBox = 'GMA' then NewBox := GMA
      else if ChgNBox = 'GMB' then NewBox := GMB
       else if ChgNBox = 'GFA' then NewBox := GFA
        else if ChgNBox = 'GFB' then NewBox := GFB
         else if ChgNBox = 'SMT' then NewBox := SMT
          else if ChgNBox = 'SFT' then NewBox := SFT
```

```
          else if ChgNBox = 'OTH' then NewBox := OTH
           else if ChgNBox = 'TOT' then NewBox := TOT
            else begin
              writeln;
              writeln('You have entered an invalid Mission Box.');
              GoodNBox := False;
              write('Please enter one of the choices :> ');
              readln(ChgNBox);
            end; {Else stmt}
  Until GoodNBox;
  If (NOT StopUpdate) then MakeChange(NewBox);
  CRT.ClrScr;
 until StopUpdate;
end; {Procedure UpdateCurrentData}

{---------------------------------}
Procedure PrintCurrentData;

 {*** Outputs the Current Data File to a text file called ChkCurr.txt
    so that the data can be checked against the original file.    ***}

var CurrOutFile : String;
var i,j : integer;

begin
 write('Current Tested and Contracted data is output to ');
 writeln('C:\PEWS\ChkCurr.txt.');
 CurrOutFile := 'C:\PEWS\ChkCurr.txt';
 assign(output,CurrOutFile);
 rewrite(output);

{  For i := 1 to 8 do begin   }
   writeln('For Tested Data');
   writeln('BiMo   FY89  FY90  FY91  FY92  FY93  FY94  FY95');
   TestContPtr1 := FirstTest[i].FirstPtr;
   For j := 1 to 24 do begin
    write(' ',ord(TestContPtr1^.BiMo)+1:2,'   ');
    write(TestContPtr1^.Num89:6,' ');
    write(TestContPtr1^.Num90:6,' ');
    write(TestContPtr1^.Num91:6,' ');
    write(TestContPtr1^.Num92:6,' ');
    write(TestContPtr1^.Num93:6,' ');
    write(TestContPtr1^.Num94:6,' ');
    writeln(TestContPtr1^.Num95:6);
    TestContPtr1 := TestContPtr1^.Next;
   end; {For j loop}
   writeln;
{  end; {For i loop}

  For i := 1 to 8 do begin
   writeln('For Contracted Data');
```

```
   writeln('week   FY89  FY90  FY91  FY92  FY93  FY94  FY95');
   TestContPtr1 := FirstCont[i].FirstPtr;
   For j := 1 to 24 do begin
    write(' ',ord(TestContPtr1^.BiMo)+1:2,'   ');
    write(TestContPtr1^.Num89:6,' ');
    write(TestContPtr1^.Num90:6,' ');
    write(TestContPtr1^.Num91:6,' ');
    write(TestContPtr1^.Num92:6,' ');
    write(TestContPtr1^.Num93:6,' ');
    write(TestContPtr1^.Num94:6,' ');
    writeln(TestContPtr1^.Num95:6);
    TestContPtr1 := TestContPtr1^.Next;
   end; {For j loop}
   writeln;
 end; {For i loop}

 close(output);
 assign(output,'con');
 rewrite(output);
end; {Procedure PrintCurrentData}

{--------------------------}
Procedure PrintDailyAvgData;

 {*** Outputs the Daily Avg Data File to a text file called ChkCurr.txt
    so that the data can be checked against the original file.      ***}

 var DailyAvgOutFile : String;
 var i,j : integer;

 begin
  write('Daily Average Tested and Contracted data is output to ');
  writeln('C:\PEWS\ChkDaily.txt.');
  DailyAvgOutFile := 'C:\PEWS\ChkDaily.txt';
  assign(output,DailyAvgOutFile);
  rewrite(output);

  writeln('Average Number of Prospects Tested Daily.');
  writeln(' Mo DaysInMo   Number of prospects tested daily ');
  For i := 1 to 12 do begin
   MoDayPtr1 := DailyAvg[i].FirstPtr;
   write(i:3,DailyAvg[i].DaysPerMo:5,'    ');
   For j := 1 to 16 do begin
    write(MoDayPtr1^.NumTest,', ');
    MoDayPtr1 := MoDayPtr1^.Next;
   end; {For j=1 thru 16}
   writeln;write('           ');
   For j := 17 to DailyAvg[i].DaysPerMo do begin
    write(MoDayPtr1^.NumTest,', ');
    MoDayPtr1 := MoDayPtr1^.Next;
   end; {For j=17 thru rest of days}
```

```
    writeln;
  end; {For i loop}

  writeln;
  writeln('Average Number of Contracts Signed Daily.');
  writeln(' Mo DaysInMo   Number of contracts signed daily ');
  For i := 1 to 12 do begin
    MoDayPtr1 := DailyAvg[i].FirstPtr;
    write(i:3,DailyAvg[i].DaysPerMo:5,'     ');
    For j := 1 to 16 do begin
      write(MoDayPtr1^.NumCont,', ');
      MoDayPtr1 := MoDayPtr1^.Next;
    end; {For j=1 thru 16}
    writeln;write('              ');
    For j := 17 to DailyAvg[i].DaysPerMo do begin
      write(MoDayPtr1^.NumCont,', ');
      MoDayPtr1 := MoDayPtr1^.Next;
    end; {For j=17 thru rest of days}
    writeln;
  end; {For i loop}

 close(output);
 assign(output,'con');
 rewrite(output);
 end;

{--Unit Main Program-----------------------------------------------------}

begin
END. {UNIT PEWSCURR}
```

3. PASCAL CODE FOR PEWS MODEL PARAMETER DATA UNIT

```
unit PEWSPARA;

  {Contains Data Types and Procedures for Parameters for PEWS Model}

{-------------------------------------------------------------------}
interface

uses CRT;

type MissionBoxTYPE = (GMA,GMB,GFA,GFB,SMT,SFT,OTH,TOT);

type BiMonthTYPE = (OCTA,OCTB,NOVA,NOVB,DECA,DECB,JANA,JANB,
            FEBA,FEBB,MARA,MARB,APRA,APRB,MAYA,MAYB,
            JUNA,JUNB,JULA,JULB,AUGA,AUGB,SEPA,SEPB);

type TestedTYPE = record
             BiMo : BiMonthTYPE;
             TotTested : real;
            end;

type ContractedTYPE = record
              BiMo : BiMonthTYPE;
              GMACont : real;
              GMBCont : real;
              GFACont : real;
              GFBCont : real;
              SMTCont : real;
              SFTCont : real;
              OTHCont : real;
              TOTCont : real;
             end;

type UnmatchedTYPE = record
              BiMo : BiMonthTYPE;
              GMAUnm : real;
              GMBUnm : real;
              GFAUnm : real;
              GFBUnm : real;
              SMTUnm : real;
              SFTUnm : real;
              OTHUnm : real;
              TOTUnm : real;
             end;


type TimeToContractPOINT = ^TimeToContractTYPE;
  TimeToContractTYPE = Record
              Next : TimeToContractPOINT;
```

```
                BiMo : BiMonthTYPE;
                day1 : Real;
                day2 : Real;
                day3 : Real;
                day4 : Real;
                day5 : Real;
                day6 : Real;
                day7 : Real;
                day8to9 : Real;
                day10to12 : Real;
                day13to14 : Real;
                day15to18 : Real;
                day19to22 : Real;
                day23to30 : Real;
                day31to45 : Real;
                day46to60 : Real;
                day61to120 : Real;
                day121to180 : Real;
                day181to365 : Real;
                day366to730 : Real;
                dayGT731 : Real;
               end;

type FirstNodeTYPE = Record
                FirstPtr : TimeToContractPOINT;
                NBox : MissionBoxType;
               end;

VAR Infile,Outfile : Text;
   NumTested : Array[1..24] of TestedTYPE;
   PercentCont : Array[1..24] of ContractedTYPE;
   PercentUnm : Array[1..24] of UnmatchedTYPE;
   TTC : Array[1..8] of FirstNodeTYPE;
   TTCPtr1, TTCPtr2 : TimeToContractPOINT;

  Procedure GetParameters;

  Procedure PrintParameters;

{-------------------------------------------------------------------}
implementation

Procedure GetParameters;    {Reads model parameters from a flat text file}

var ParamFile, DummyStr : String;
var DummyStr5 : String[5];
var i,j,k : integer;
var MissionBox : MissionBoxTYPE;
var TwoWeeks : BiMonthTYPE;
```

```
Begin
  write('Model parameters are read from textfile ');
  writeln('"C:\PEWS\ParaData.txt".');
  ParamFile := 'C:\PEWS\ParaData.txt';
  Assign(Input,ParamFile);
  Reset(Input);

{** Read in the number of applicants expected to take the AFQT weekly. **}
  readln(DummyStr);
  readln(DummyStr); {Read past 2 lines of explanatory text headings.}
  TwoWeeks := OCTA;
  FOR i := 1 to 24 do
    begin
      read(DummyStr5);
      NumTested[i].BiMo := TwoWeeks;
      read(NumTested[i].TotTested);
      readln(DummyStr);   {Read to end of line and reset to next line.}
      TwoWeeks := succ(TwoWeeks);
    end; {FOR loop}

{** Read in the percentage of AFQT test takers expected to sign a contract.}
  readln(DummyStr); {Read past blank spacing line.}
  readln(DummyStr);
  readln(DummyStr); {Read past 2 lines of explanatory text headings.}
  TwoWeeks := OCTA;
  FOR i := 1 to 24 do
    begin
      read(DummyStr5);
      PercentCont[i].BiMo := TwoWeeks;
      read(PercentCont[i].GMACont);
      read(PercentCont[i].GMBCont);
      read(PercentCont[i].GFACont);
      read(PercentCont[i].GFBCont);
      read(PercentCont[i].SMTCont);
      read(PercentCont[i].SFTCont);
      read(PercentCont[i].OTHCont);
      read(PercentCont[i].TOTCont);
      readln(DummyStr);   {Read to end of line and reset to next line.}
      TwoWeeks := succ(TwoWeeks);
    end;  {FOR loop}

{** Read in the percentage of additional contracts expected due to
   unmatched records (from group and field administered AFQT's.  **}
  readln(DummyStr); {Read past blank spacing line.}
  readln(DummyStr);
  readln(DummyStr); {Read past 2 lines of explanatory text headings.}
  TwoWeeks := OCTA;
  FOR i := 1 to 24 do
    begin
      read(DummyStr5);
      PercentUnm[i].BiMo := TwoWeeks;
```

```
      read(PercentUnm[i].GMAUnm);
      read(PercentUnm[i].GMBUnm);
      read(PercentUnm[i].GFAUnm);
      read(PercentUnm[i].GFBUnm);
      read(PercentUnm[i].SMTUnm);
      read(PercentUnm[i].SFTUnm);
      read(PercentUnm[i].OTHUnm);
      read(PercentUnm[i].TOTUnm);
      readln(DummyStr);  {Read to end of line and reset to next line.}
      TwoWeeks := succ(TwoWeeks);
    end;  {FOR loop}

{**** Read in the expected distribution of time to contract (in days)
    for the GMA mission box category.                    *****}
 MissionBox := GMA;
 For i := 1 to 8 do begin  {Cycle thru 10 mission box categories}
  TTC[i].NBox := MissionBox;
  new(TTC[i].FirstPtr);
  TTC[i].FirstPtr^.Next := nil;
  For j := 1 to 48 do begin  {Cycle thru 24 biweekly breakouts two times}
   If ((j = 1) or (j = 25)) then begin
    readln(DummyStr);
    readln(DummyStr);        {read past 3 blank/heading lines}
    readln(DummyStr);
    TwoWeeks := OCTA;          {initialize TwoWeeks category}
    TTCPtr1 := TTC[i].FirstPtr;  {Go to the first record in the list}
   end; {If stmt}
   read(DummyStr5);         {read past TwoWeeks heading}
   TTCPtr1^.BiMo := TwoWeeks;
   If j <= 24 then begin
    read(TTCPtr1^.day1);     {read in first 10 Day to Contract Dist'ns}
    read(TTCPtr1^.day2);
    read(TTCPtr1^.day3);
    read(TTCPtr1^.day4);
    read(TTCPtr1^.day5);
    read(TTCPtr1^.day6);
    read(TTCPtr1^.day7);
    read(TTCPtr1^.day8to9);
    read(TTCPtr1^.day10to12);
    read(TTCPtr1^.day13to14);
    new(TTCPtr1^.Next);
    If (j < 24) then begin     {Don't add bogus record at end of list}
     TTCPtr1 := TTCPtr1^.Next;    {Make a new record for each BiWeek}
     TTCPtr1^.Next := nil;
    end; {inner If stmt}
   end; {If j<=24 stmt}
   If j >= 25 then begin
    read(TTCPtr1^.day15to18);  {read in last 10 Day to Contract Dist'ns}
    read(TTCPtr1^.day19to22);
    read(TTCPtr1^.day23to30);
    read(TTCPtr1^.day31to45);
```

```
      read(TTCPtr1^.day46to60);
      read(TTCPtr1^.day61to120);
      read(TTCPtr1^.day121to180);
      read(TTCPtr1^.day181to365);
      read(TTCPtr1^.day366to730);
      read(TTCPtr1^.dayGT731);
      TTCPtr1 := TTCPtr1^.Next;    {increment Current Pointer}
    end; {If j>=25 stmt}
    readln(DummyStr);            {Read to end of line and reset}
    TwoWeeks := succ(TwoWeeks);   {increment TwoWeeks}
  end; {For j loop}
  MissionBox := succ(MissionBox); {increment MissionBox}
 end; {For i loop}

 close(input);
 Assign(input,'con');        {Reset default input to keyboard}
 reset(input);
end; {Procedure GetParameters}

{-------------------------------}
Procedure PrintParameters;

 {***Prints out the current model parameters as read into the program
   from the text file. Output printed out to C:\PEWS\ChkPara.txt. ***}

var ParaOutFile : string;
var i,j,n : integer;

begin
 write('PEWS Model Parameters are output to ');
 writeln('C:\PEWS\ChkPara.txt.');
 ParaOutFile := 'C:\PEWS\ChkPara.txt';
 assign(output,ParaOutFile);
 rewrite(output);

 writeln('Number of applicants projected to take the AFQT weekly.');
 writeln('BiMo  #Testers');
 For i := 1 to 24 do begin
  writeln(ord(NumTested[i].BiMo)+1:3,'    ',NumTested[i].TotTested:5:0);
 end; {For i loop}
 writeln;

 writeln('Percentage of AFQT Test Takers expected to sign a contract.');
 writeln('BiMo  %GMA  %GMB  %GFA  %GFB  %SMT  %SFT  %OTH  %TOT');
 For i := 1 to 24 do begin
  write(ord(PercentCont[i].BiMo)+1:3,' ');
  write(PercentCont[i].GMACont:7:2);
  write(PercentCont[i].GMBCont:7:2);
  write(PercentCont[i].GFACont:7:2);
  write(PercentCont[i].GFBCont:7:2);
  write(PercentCont[i].SMTCont:7:2);
```

```
  write(PercentCont[i].SFTCont:7:2);
  write(PercentCont[i].OTHCont:7:2);
  writeln(PercentCont[i].TOTCont:7:2);
 end; {For i loop}
 writeln;

 writeln('Percentage of addtl contracts expected due to unmatched recds.');
 writeln('BiMo  %GMA  %GMB  %GFA  %GFB  %SMT  %SFT  %OTH  %TOT');
 For i := 1 to 24 do begin
  write(ord(PercentUnm[i].BiMo)+1:3,'  ');
  write(PercentUnm[i].GMAUnm:7:2);
  write(PercentUnm[i].GMBUnm:7:2);
  write(PercentUnm[i].GFAUnm:7:2);
  write(PercentUnm[i].GFBUnm:7:2);
  write(PercentUnm[i].SMTUnm:7:2);
  write(PercentUnm[i].SFTUnm:7:2);
  write(PercentUnm[i].OTHUnm:7:2);
  writeln(PercentUnm[i].TOTUnm:7:2);
 end; {For i loop}

{Print out the Days to Contract Distribution from the linked list}
 For n := 1 to 8 do begin
  For i := 1 to 48 do begin
   If ((i = 1) OR (i = 25)) then begin  {Heading -- Line1}
    writeln;
    If n = 1 then write('GMA ')
     Else If n = 2 then write('GMB ')
      Else If n = 3 then write('GFA ')
       Else If n = 4 then write('GFB ')
        Else If n = 5 then write('SMT ')
         Else If n = 6 then write('SFT ')
          Else If n = 7 then write('OTH ')
           Else If n = 8 then write('TOT ');
    writeln('Expected Pctg distribution of Time (in days) to contract.');
   end; {If stmt}
   If i = 1 then begin   {Heading -- First 10 Distribution Categories}
    write(' 1day   2day   3day   4day   5day   6day   7day');
    writeln('   8-9    10-12  13-14');
    TTCPtr1 := TTC[n].FirstPtr;   {Initialize pointer to first record}
   end; {If stmt}
   If i = 25 then begin  {Heading -- Last 10 Distribution Categories}
    write(' 15-18  19-22  23-30  31-45  46-60  61-120  121-180 ');
    writeln(' 181-360  361-720  GT721');
    TTCPtr1 := TTC[n].FirstPtr;   {Initialize pointer to first record}
   end; {If stmt}
   If i <= 24 then begin
    write(TTCPtr1^.day1:6:2);    {write first 10 distribution cat's}
    write(TTCPtr1^.day2:7:2);
    write(TTCPtr1^.day3:7:2);
    write(TTCPtr1^.day4:8:2);
    write(TTCPtr1^.day5:8:2);
```

```
      write(TTCPtr1^.day6:8:2);
      write(TTCPtr1^.day7:8:2);
      write(TTCPtr1^.day8to9:8:2);
      write(TTCPtr1^.day10to12:8:2);
      writeln(TTCPtr1^.day13to14:8:2);
      TTCPtr1 := TTCPtr1^.Next;
    end; {If i <= 24}
    If i >= 25 then begin
      write(TTCPtr1^.day15to18:5:2); {write last 10 distribution cat's}
      write(TTCPtr1^.day19to22:8:2);
      write(TTCPtr1^.day23to30:7:2);
      write(TTCPtr1^.day31to45:7:2);
      write(TTCPtr1^.day46to60:7:2);
      write(TTCPtr1^.day61to120:9:2);
      write(TTCPtr1^.day121to180:8:2);
      write(TTCPtr1^.day181to365:9:2);
      write(TTCPtr1^.day366to730:9:2);
      writeln(TTCPtr1^.dayGT731:9:2);
      TTCPtr1 := TTCPtr1^.Next;
    end; {If i >= 25}
  end; {For i loop}
 end; {For n loop}
 close(output);
 assign(output,'con');
 rewrite(output);
end; {Procedure PrintParameters}


{--Unit Main Program-----------------------------------------------------}

begin
end. {Unit PEWSPARA}
```

APPENDIX B. MODEL PARAMETER FILES

The following pages portray the EXCEL files in which the model parameters are stored. To change or update a parameter used in the model, perform the following steps:

1) Enter Microsoft EXCEL and open the file in which the parameter is located.

CURRDATA.xls	PARADATA.xls	AVGDATA.xls
FirstTest	- NumTested	- DailyTest
FirstCont	- PctCont	- DailyCont
	- PctUnm	- PctNBox
	- TTC	

2) While making corrections, *do not change the format of the parameters!* This specific format is required to enable the PEWS program to read in the parameters from a flat file.

3) After making changes, resave the EXCEL file as a tab-delimited text file to the appropriate .txt file name. This updates the text files that the program reads from to update the parameters in the model.

4) Ensure that all files are stored in the **C:\PEWS** directory, as this is the directory that will look for when reading the parameters.

The following pages of this appendix depict the parameters as they appear in their EXCEL source files.

1. CURRDATA.XLS

FIRSTTEST [1] **Current number of GMA prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	1150	1283	1041	1281	980	0	0
OctB	1597	1773	1485	1597	1063	0	0
NovA	1463	1457	1200	1240	934	0	0
NovB	1514	1567	1210	1113	1123	0	0
DecA	1698	1759	1442	1304	1307	0	0
DecB	1263	1204	945	1047	952	0	0
JanA	1765	2045	1649	1657	1506	0	0
JanB	2143	2375	3406	1575	1316	0	0
FebA	1559	1818	2051	1278	1147	0	0
FebB	1362	1388	1391	1115	1001	0	0
MarA	1447	1627	1612	1025	1235	0	0
MarB	1565	1476	1504	1223	1271	0	0
AprA	1323	1248	1473	873	1126	0	0
AprB	1215	1259	1510	874	965	0	0
MayA	1179	1230	1325	894	999	0	0
MayB	1576	1402	1473	814	1112	0	0
JunA	1690	1651	1721	1358	1288	0	0
JunB	1756	1675	1729	1478	1327	0	0
JulA	1397	1441	1591	1442	1178	0	0
JulB	2057	2172	2212	1636	1299	0	0
AugA	1686	1828	1767	1314	1180	0	0
AugB	1978	2031	1549	1427	1348	0	0
SepA	1352	1210	1391	969	542	0	0
SepB	1485	1337	1547	1112	807	0	0

FIRSTTEST [2] **Current number of GMB prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	718	735	475	537	465	0	0
OctB	941	888	627	642	494	0	0
NovA	685	680	488	463	432	0	0
NovB	803	711	467	401	424	0	0
DecA	848	749	583	446	558	0	0
DecB	666	515	372	384	383	0	0
JanA	874	972	684	552	629	0	0
JanB	1091	1171	1368	579	500	0	0

FebA	836	849	731	456	453	0	0
FebB	716	595	538	394	387	0	0
MarA	730	725	677	398	402	0	0
MarB	745	655	570	479	482	0	0
AprA	572	445	611	316	377	0	0
AprB	630	463	615	321	383	0	0
MayA	595	472	568	329	340	0	0
MayB	690	564	544	337	433	0	0
JunA	888	737	648	616	571	0	0
JunB	964	778	749	714	627	0	0
JulA	762	727	740	645	549	0	0
JulB	1057	1052	943	744	587	0	0
AugA	908	759	792	649	547	0	0
AugB	1111	940	716	661	623	0	0
SepA	763	570	643	458	249	0	0
SepB	808	577	678	508	446	0	0

FIRSTTEST [3] **Current number of GFA prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	384	375	316	363	301	0	0
OctB	508	542	473	437	315	0	0
NovA	365	429	349	405	300	0	0
NovB	402	485	263	314	327	0	0
DecA	405	498	329	374	353	0	0
DecB	362	358	207	316	256	0	0
JanA	474	568	339	388	428	0	0
JanB	543	682	531	419	411	0	0
FebA	391	462	387	373	319	0	0
FebB	332	369	293	340	320	0	0
MarA	401	463	389	300	362	0	0
MarB	487	453	367	390	413	0	0
AprA	389	330	426	290	333	0	0
AprB	422	409	418	314	346	0	0
MayA	377	380	393	307	341	0	0
MayB	462	487	395	311	377	0	0
JunA	577	551	518	449	448	0	0
JunB	569	530	596	540	428	0	0
JulA	421	452	467	502	396	0	0
JulB	623	598	674	523	442	0	0
AugA	543	493	518	402	408	0	0

AugB	631	473	455	428	449	0	0
SepA	404	322	367	321	140	0	0
SepB	445	384	505	398	206	0	0

FIRSTTEST [4] **Current number of GFB prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	334	309	187	223	187	0	0
OctB	350	342	296	249	208	0	0
NovA	295	281	189	195	170	0	0
NovB	288	297	178	153	179	0	0
DecA	298	352	175	171	188	0	0
DecB	262	192	112	157	151	0	0
JanA	314	351	186	235	248	0	0
JanB	393	447	260	217	192	0	0
FebA	294	280	194	174	186	0	0
FebB	258	203	157	148	165	0	0
MarA	275	250	221	172	197	0	0
MarB	298	237	203	232	186	0	0
AprA	264	205	224	171	154	0	0
AprB	286	197	260	154	170	0	0
MayA	264	202	198	166	161	0	0
MayB	323	252	200	173	204	0	0
JunA	396	282	313	313	239	0	0
JunB	425	349	315	281	262	0	0
JulA	377	286	269	285	274	0	0
JulB	481	354	374	322	265	0	0
AugA	416	306	280	294	256	0	0
AugB	471	268	290	258	237	0	0
SepA	323	202	235	179	95	0	0
SepB	327	227	270	213	145	0	0

FIRSTTEST [5] **Current number of SMT prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	2000	1804	1592	1278	1343	0	0
OctB	2663	2311	1974	1705	1300	0	0
NovA	2313	1931	1532	1440	1154	0	0
NovB	2063	1643	1134	1194	1110	0	0
DecA	2170	1884	1189	1122	1088	0	0
DecB	1537	1103	733	804	807	0	0
JanA	1452	1340	934	1084	918	0	0

JanB	2156	1910	942	1227	926	0	0
FebA	1847	1755	845	1227	974	0	0
FebB	1489	1321	802	959	870	0	0
MarA	1632	1592	1077	1017	1051	0	0
MarB	1697	1458	947	1166	1163	0	0
AprA	1267	1123	1031	792	1046	0	0
AprB	1218	1134	1077	757	834	0	0
MayA	1084	987	993	639	725	0	0
MayB	1078	932	848	419	684	0	0
JunA	1553	1242	1183	433	933	0	0
JunB	1995	1554	1339	500	1428	0	0
JulA	1486	1456	1197	490	1320	0	0
JulB	1082	2074	1650	639	1431	0	0
AugA	1897	1836	1499	881	1401	0	0
AugB	2080	1451	1319	1286	1522	0	0
SepA	1182	1106	946	1277	510	0	0
SepB	1865	1518	1565	1522	1058	0	0

FIRSTTEST [6] **Current number of SFT prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	202	270	218	104	227	0	0
OctB	289	400	279	124	231	0	0
NovA	319	349	122	119	255	0	0
NovB	291	291	115	127	237	0	0
DecA	298	353	113	148	207	0	0
DecB	267	249	99	145	160	0	0
JanA	257	326	138	170	171	0	0
JanB	390	531	161	216	199	0	0
FebA	312	444	157	242	245	0	0
FebB	239	317	139	217	203	0	0
MarA	345	357	185	240	254	0	0
MarB	391	367	181	289	291	0	0
AprA	320	365	243	236	249	0	0
AprB	313	383	244	210	197	0	0
MayA	276	325	246	183	214	0	0
MayB	275	271	173	167	194	0	0
JunA	185	166	103	68	145	0	0
JunB	194	152	77	67	182	0	0
JulA	129	85	58	63	155	0	0
JulB	195	141	90	76	214	0	0

AugA	207	124	76	106	189	0	0
AugB	227	108	64	158	251	0	0
SepA	130	82	45	172	32	0	0
SepB	217	149	96	246	63	0	0

FIRSTTEST [7] **Current number of OTH prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	4311	4340	2707	1448	2166	0	0
OctB	5216	5592	3112	1721	2203	0	0
NovA	4513	3982	2600	1228	1868	0	0
NovB	4732	3132	2997	1014	1860	0	0
DecA	5328	3534	3813	1214	2307	0	0
DecB	4031	2270	2254	920	2176	0	0
JanA	5391	3428	3852	1335	3719	0	0
JanB	6788	4242	6148	1329	3956	0	0
FebA	5383	2698	3399	1137	2952	0	0
FebB	4575	2002	2282	1181	1751	0	0
MarA	5392	2556	2552	1140	1932	0	0
MarB	6115	2162	2277	1472	1789	0	0
AprA	5826	1565	2145	1228	1528	0	0
AprB	6109	1648	2262	1251	1599	0	0
MayA	4717	1697	2124	1281	1705	0	0
MayB	5309	1769	1731	1127	1819	0	0
JunA	5571	2029	1709	1620	1954	0	0
JunB	5555	2062	1833	1601	2013	0	0
JulA	4641	1838	1522	1627	1986	0	0
JulB	6296	2467	2233	2101	2023	0	0
AugA	5976	2126	1785	2213	2028	0	0
AugB	6347	3644	1682	2233	2559	0	0
SepA	5043	2510	1483	2013	777	0	0
SepB	4436	2648	1809	2392	1478	0	0

FIRSTTEST [8] **Current number of TOT prospects AFQT tested to date**

BiMonth	FY89T	FY90T	FY91T	FY92T	FY93T	FY94T	FY95T
OctA	9099	9116	6536	5234	5669	0	0
OctB	11564	11848	8246	6475	5814	0	0
NovA	9953	9109	6480	5090	5113	0	0
NovB	10093	8126	6364	4316	5260	0	0
DecA	11045	9129	7644	4779	6008	0	0
DecB	8388	5891	4722	3773	4885	0	0

JanA	10527	9030	7782	5421	7619	0	0
JanB	13504	11358	12816	5562	7500	0	0
FebA	10622	8306	7764	4887	6276	0	0
FebB	8971	6195	5602	4354	4697	0	0
MarA	10222	7570	6713	4292	5433	0	0
MarB	11298	6808	6049	5251	5595	0	0
AprA	9961	5281	6153	3906	4813	0	0
AprB	10193	5493	6386	3881	4494	0	0
MayA	8492	5293	5847	3799	4485	0	0
MayB	9713	5677	5364	3348	4823	0	0
JunA	10860	6658	6195	4857	5578	0	0
JunB	11458	7100	6638	5181	6267	0	0
JulA	9213	6285	5844	5054	5858	0	0
JulB	11791	8858	8176	6041	6261	0	0
AugA	11633	7472	6717	5859	6009	0	0
AugB	12845	8915	6075	6451	6989	0	0
SepA	9197	6002	5110	5389	2345	0	0
SepB	9583	6840	6470	6391	4203	0	0

FIRSTCONT [1] **Current number of GMA contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	979	1153	1121	1243	1091	0	0
OctB	1508	1842	1553	1844	1320	0	0
NovA	962	1249	1042	1268	896	0	0
NovB	1358	1574	1266	1317	1269	0	0
DecA	1136	1560	1116	1143	1011	0	0
DecB	1269	1272	923	1141	1298	0	0
JanA	1074	1419	930	1288	1167	0	0
JanB	1855	2537	2156	1895	1286	0	0
FebA	1202	1551	1603	1266	1089	0	0
FebB	1298	1552	1224	1392	1110	0	0
MarA	1089	1299	1489	1016	1021	0	0
MarB	1479	1718	1484	1580	1521	0	0
AprA	905	1051	1319	769	924	0	0
AprB	1029	1242	1650	946	1214	0	0
MayA	994	1137	1233	825	944	0	0
MayB	1357	1405	1633	949	1307	0	0
JunA	1284	1462	1630	1181	1048	0	0
JunB	1794	1769	1914	1631	1600	0	0
JulA	1116	1286	1613	1333	1022	0	0

JulB	2024	2179	2283	1984	1540	0	0
AugA	1378	1582	1708	1274	1017	0	0
AugB	2014	2430	2179	1760	1601	0	0
SepA	1386	1326	1499	1014	898	0	0
SepB	1674	1599	1907	1426	1407	0	0

FIRSTCONT [2] **Current number of GMB contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	600	707	576	801	560	0	0
OctB	796	1088	882	476	655	0	0
NovA	557	695	533	101	405	0	0
NovB	725	847	572	56	590	0	0
DecA	686	779	566	47	461	0	0
DecB	705	662	468	131	581	0	0
JanA	530	690	444	282	533	0	0
JanB	982	1322	931	283	620	0	0
FebA	638	845	645	114	519	0	0
FebB	709	819	526	741	584	0	0
MarA	563	653	677	323	425	0	0
MarB	752	831	652	565	661	0	0
AprA	490	362	575	433	363	0	0
AprB	639	458	840	426	537	0	0
MayA	493	145	586	363	367	0	0
MayB	743	576	430	452	636	0	0
JunA	739	843	113	686	488	0	0
JunB	1168	844	740	951	812	0	0
JulA	749	478	583	741	484	0	0
JulB	1303	888	768	1099	792	0	0
AugA	888	744	698	732	565	0	0
AugB	1407	1117	819	824	841	0	0
SepA	947	720	583	570	471	0	0
SepB	1019	770	502	672	690	0	0

FIRSTCONT [3] **Current number of GFA contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	320	300	325	410	318	0	0
OctB	398	517	481	490	368	0	0
NovA	275	305	298	327	250	0	0
NovB	366	460	293	358	353	0	0
DecA	312	389	259	311	256	0	0

DecB	310	377	213	342	334	0	0
JanA	292	329	233	294	247	0	0
JanB	449	654	378	499	318	0	0
FebA	252	384	283	314	294	0	0
FebB	310	410	231	358	298	0	0
MarA	257	322	308	265	256	0	0
MarB	431	528	316	409	451	0	0
AprA	280	286	329	236	265	0	0
AprB	300	423	364	341	357	0	0
MayA	275	326	300	246	252	0	0
MayB	374	442	432	322	373	0	0
JunA	339	434	407	319	324	0	0
JunB	518	575	489	491	467	0	0
JulA	289	320	391	408	314	0	0
JulB	599	512	586	548	454	0	0
AugA	329	406	431	336	302	0	0
AugB	629	579	542	459	470	0	0
SepA	343	351	300	333	294	0	0
SepB	468	469	485	422	446	0	0

FIRSTCONT [4] **Current number of GFB contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	199	213	184	144	158	0	0
OctB	155	300	332	122	216	0	0
NovA	258	251	209	98	120	0	0
NovB	204	222	218	90	165	0	0
DecA	211	217	199	64	126	0	0
DecB	189	202	125	80	139	0	0
JanA	175	184	121	117	178	0	0
JanB	243	355	223	121	155	0	0
FebA	196	6	166	78	119	0	0
FebB	210	6	113	188	95	0	0
MarA	204	6	130	109	57	0	0
MarB	363	6	153	231	146	0	0
AprA	315	0	102	157	46	0	0
AprB	238	2	85	170	94	0	0
MayA	292	6	51	166	103	0	0
MayB	331	31	77	191	150	0	0
JunA	351	103	39	257	120	0	0
JunB	432	25	177	272	198	0	0

JulA	303	1	87	231	161	0	0
JulB	383	2	128	307	247	0	0
AugA	416	2	110	199	142	0	0
AugB	467	2	166	247	158	0	0
SepA	302	1	93	148	78	0	0
SepB	314	168	102	217	166	0	0

FIRSTCONT [5] **Current number of SMT contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	982	1063	1039	624	1038	0	0
OctB	1648	1838	1114	1101	1297	0	0
NovA	1218	1244	739	815	852	0	0
NovB	1765	1731	848	925	937	0	0
DecA	1466	1607	641	803	875	0	0
DecB	1724	1541	1025	729	981	0	0
JanA	1020	983	721	583	698	0	0
JanB	1897	2153	772	1075	837	0	0
FebA	1313	1539	549	689	778	0	0
FebB	1476	1406	631	796	830	0	0
MarA	1145	1325	870	602	824	0	0
MarB	1638	1793	996	1140	1168	0	0
AprA	1087	779	766	756	777	0	0
AprB	1186	855	1138	833	1016	0	0
MayA	1001	700	808	572	745	0	0
MayB	949	612	819	672	742	0	0
JunA	688	500	148	199	164	0	0
JunB	2047	1378	1382	199	1546	0	0
JulA	999	827	762	64	711	0	0
JulB	1550	1203	1012	251	1071	0	0
AugA	1147	895	789	142	689	0	0
AugB	1616	975	987	205	1061	0	0
SepA	967	494	497	1806	528	0	0
SepB	1430	1187	962	1243	1027	0	0

FIRSTCONT [6] **Current number of SFT contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	58	72	124	20	87	0	0
OctB	115	153	127	41	135	0	0
NovA	100	102	38	28	108	0	0
NovB	148	182	65	111	166	0	0

DecA	105	149	13	95	97	0	0
DecB	139	191	110	93	131	0	0
JanA	81	116	89	96	93	0	0
JanB	182	310	87	171	126	0	0
FebA	132	196	58	107	118	0	0
FebB	158	205	62	127	136	0	0
MarA	104	142	98	94	123	0	0
MarB	219	221	96	189	194	0	0
AprA	154	201	101	130	119	0	0
AprB	167	266	139	165	148	0	0
MayA	164	147	102	92	102	0	0
MayB	153	157	107	118	116	0	0
JunA	48	20	2	30	29	0	0
JunB	121	9	13	18	148	0	0
JulA	48	9	7	4	87	0	0
JulB	96	30	22	19	131	0	0
AugA	55	9	18	8	83	0	0
AugB	115	31	22	7	140	0	0
SepA	43	11	11	155	86	0	0
SepB	88	126	46	151	172	0	0

FIRSTCONT [7] **Current number of OTH contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	1155	1053	387	94	567	0	0
OctB	1303	1505	475	95	678	0	0
NovA	1374	897	445	61	384	0	0
NovB	1385	445	1115	84	499	0	0
DecA	1451	408	707	66	500	0	0
DecB	1397	415	613	77	936	0	0
JanA	1258	299	306	19	1285	0	0
JanB	1967	710	624	68	1744	0	0
FebA	1576	46	550	24	1046	0	0
FebB	1415	39	252	47	390	0	0
MarA	1419	63	339	23	244	0	0
MarB	2212	69	250	78	210	0	0
AprA	2322	34	158	54	82	0	0
AprB	2296	34	134	116	411	0	0
MayA	1739	27	72	74	267	0	0
MayB	1481	43	77	86	322	0	0
JunA	955	17	56	96	144	0	0

JunB	1215	107	58	120	266	0	0
JulA	766	59	52	92	266	0	0
JulB	1318	71	72	732	431	0	0
AugA	1039	68	54	529	260	0	0
AugB	1822	415	76	715	423	0	0
SepA	790	287	58	417	202	0	0
SepB	1095	420	79	782	378	0	0

FIRSTCONT [8] **Current number of TOT contracts signed to date**

BiMonth	FY89C	FY90C	FY91C	FY92C	FY93C	FY94C	FY95C
OctA	4293	4561	3756	3336	3819	0	0
OctB	5923	7243	4964	4169	4669	0	0
NovA	4744	4743	3304	2698	3015	0	0
NovB	5951	5461	4377	2941	3979	0	0
DecA	5367	5109	3501	2529	3326	0	0
DecB	5733	4660	3477	2593	4400	0	0
JanA	4430	4020	2844	2679	4201	0	0
JanB	7575	8041	5171	4112	5086	0	0
FebA	5309	4567	3854	2592	3963	0	0
FebB	5576	4437	3039	3649	3443	0	0
MarA	4781	3810	3911	2432	2950	0	0
MarB	7094	5166	3947	4192	4351	0	0
AprA	5553	2713	3350	2535	2576	0	0
AprB	5855	3280	4350	2997	3777	0	0
MayA	4958	2488	3152	2338	2780	0	0
MayB	5388	3266	3575	2790	3646	0	0
JunA	4404	3379	2395	2768	2317	0	0
JunB	7295	4707	4773	3682	5037	0	0
JulA	4270	2980	3495	2873	3045	0	0
JulB	7273	4885	4871	4940	4666	0	0
AugA	5252	3706	3808	3220	3058	0	0
AugB	8070	5549	4791	4217	4694	0	0
SepA	4778	3190	3041	4443	2557	0	0
SepB	6088	4739	4083	4913	4286	0	0

2. MICROSOFT EXCEL FILE PARAMETER FILE PARADATA.XLS

NUMTEST Number of applicants projected to take the AFQT Bi-Weekly

BiMo	# testers
OctA	5813
OctB	6845
NovA	5561
NovB	5313
DecA	6144
DecB	4460
JanA	6941
JanB	8626
FebA	6309
FebB	4884
MarA	5479
MarB	5632
AprA	4957
AprB	4920
MayA	4710
MayB	4512
JunA	5543
JunB	6029
JulA	5585
JulB	6826
AugA	6195
AugB	6505
SepA	5399
SepB	6517

PCTCONT Percentage of AFQT Test Takers expected to sign a contract.

BiMo	% GMA	% GMB	% GFA	% GFB	% SMT	% SFT	% OTH	% TOT
OctA	.56582	.49863	.50086	.34194	.46676	.33888	.21507	.37328
OctB	.57791	.49276	.5222	.36747	.47132	.33862	.21682	.38319
NovA	.54703	.44141	.46861	.32832	.4546	.32818	.20309	.36262
NovB	.58373	.49073	.512	.32968	.46011	.33082	.23174	.38895
DecA	.57044	.49874	.54058	.33953	.4464	.34942	.22184	.37933
DecB	.57402	.50302	.52902	.36041	.4703	.36957	.23577	.3909
JanA	.55672	.49879	.50569	.34408	.4485	.33333	.20869	.36759
JanB	.5608	.49331	.53944	.35255	.48136	.35872	.2103	.37491
FebA	.54056	.47128	.49741	.29078	.46631	.34643	.19841	.36389
FebB	.57615	.51597	.5266	.30397	.48024	.38834	.20634	.38885
MarA	.54593	.47578	.47258	.29327	.46821	.36278	.19651	.36719
MarB	.57693	.48482	.51848	.31228	.48328	.35221	.20572	.38347
AprA	.55734	.47178	.50622	.29666	.46511	.3666	.20298	.36923
AprB	.58372	.50539	.51912	.30928	.47928	.37045	.20973	.37912
MayA	.56673	.46484	.49444	.30373	.46793	.34646	.2001	.3675
MayB	.58382	.48481	.50591	.30382	.4928	.36019	.20604	.38406
JunA	.56513	.51069	.51789	.33377	.46183	.31184	.17682	.37803
JunB	.58732	.52453	.50732	.3315	.45012	.3125	.17935	.38754
JulA	.58774	.50336	.48615	.31724	.43671	.3449	.17625	.37952

JulB	.57956	.50787	.50699	.32405	.42826	.28352	.1836	.38085
AugA	.5564	.50096	.47758	.31186	.41323	.26781	.17964	.36086
AugB	.58178	.5137	.52709	.34908	.42439	.29084	.19824	.37538
SepA	.55269	.51834	.50125	.33836	.46656	.33682	.21495	.3833
SepB	.58238	.50746	.5485	.36887	.48499	.36316	.22824	.40422

PCTUN M **Percentage of add'l contracts expected due to unmatched records**

BiMo	% GMA	% GMB	% GFA	% GFB	% SMT	% SFT	% OTH	% TOT
OctA	1.60454	1.66019	1.61486	1.86696	1.67112	2.13611	1.69759	1.66373
OctB	1.48262	1.61301	1.54914	1.71233	1.75874	2.02482	1.53114	1.59772
NovA	1.56696	1.79685	1.57126	1.86085	1.82939	2.13634	1.66281	1.69776
NovB	1.44494	1.60529	1.60808	1.8729	1.9178	2.19606	1.49936	1.62835
DecA	1.37244	1.51674	1.42978	1.70207	2.02708	2.39063	1.51745	1.58846
DecB	1.39287	1.5521	1.4812	1.68577	2.03116	2.41453	1.48317	1.61402
JanA	1.41707	1.54744	1.44559	1.67386	2.0816	2.16896	1.46688	1.58311
JanB	1.39324	1.48529	1.42114	1.62038	1.99053	2.15234	1.42982	1.54331
FebA	1.37718	1.51621	1.41521	1.6716	1.88829	1.97734	1.43641	1.53036
FebB	1.35643	1.50646	1.42464	1.74856	1.84128	1.84945	1.46381	1.52653
MarA	1.38631	1.53636	1.42511	1.69799	1.85591	2.13306	1.52856	1.55948
MarB	1.36575	1.50609	1.44942	1.73551	1.79456	1.8832	1.47745	1.53384
AprA	1.36558	1.52365	1.42159	1.72703	1.82196	1.95833	1.58208	1.5547
AprB	1.36468	1.4796	1.42458	1.62707	1.79189	1.92812	1.49178	1.52311
MayA	1.37502	1.49504	1.37291	1.72144	1.87091	2.03009	1.46242	1.53297
MayB	1.38708	1.60102	1.41309	1.68831	1.83996	1.96086	1.43295	1.52992
JunA	1.45102	1.56605	1.53322	1.73307	1.99414	1.74325	1.3934	1.54
JunB	1.46402	1.54307	1.47247	1.725	2.28212	2.51218	1.39936	1.64531
JulA	1.44346	1.55323	1.45195	1.71709	2.10186	2.09459	1.36616	1.57391
JulB	1.42755	1.56301	1.50029	1.66459	1.90454	2.1439	1.42763	1.54828
AugA	1.45892	1.48648	1.48355	1.64585	1.75636	2.21793	1.37227	1.5189
AugB	1.44823	1.52777	1.48834	1.67743	1.73683	2.1	1.40972	1.5196
SepA	1.57282	1.57766	1.56769	1.64549	1.82482	2.06757	1.3618	1.61095
SepB	1.62009	1.66803	1.65703	1.7646	1.8687	2.37959	1.47037	1.6843

TTC [1] **GMA Expected (percentage) distribution of Time (in days) to Contract**

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.22329	0.04725	0.04015	0.01729	0.03212	0.05188	0.04972	0.0664	0.04632	0.04077
OctB	0.26349	0.05602	0.0597	0.04772	0.04426	0.04288	0.0385	0.05533	0.04288	0.03089
NovA	0.22475	0.04862	0.04541	0.03348	0.03435	0.046	0.04687	0.06841	0.04891	0.04017
NovB	0.28275	0.04892	0.04971	0.03367	0.0384	0.04813	0.04103	0.04918	0.0455	0.03498
DecA	0.25942	0.05076	0.05731	0.04912	0.04327	0.0407	0.04795	0.06082	0.04351	0.02971
DecB	0.28239	0.05299	0.04297	0.02908	0.0252	0.03037	0.04136	0.05622	0.04426	0.03716
JanA	0.22723	0.04545	0.05215	0.03183	0.04209	0.04398	0.05424	0.07162	0.0511	0.03435
JanB	0.25529	0.05605	0.05605	0.04415	0.03737	0.05126	0.04514	0.05969	0.04398	0.03307
FebA	0.23299	0.05246	0.04773	0.03379	0.03663	0.04395	0.04962	0.08034	0.05198	0.03284
FebB	0.27528	0.05278	0.06389	0.04389	0.03778	0.04389	0.04028	0.04972	0.03917	0.03361
MarA	0.2461	0.04287	0.05239	0.03996	0.03731	0.0471	0.05398	0.07065	0.05954	0.03096
MarB	0.28239	0.05143	0.06182	0.0502	0.0366	0.03932	0.04674	0.04847	0.03561	0.0277
AprA	0.25992	0.04894	0.04446	0.03671	0.03551	0.04655	0.04596	0.06446	0.04685	0.02925
AprB	0.28145	0.05641	0.06084	0.04253	0.04223	0.03839	0.04371	0.05139	0.04341	0.03101
MayA	0.26904	0.0516	0.04751	0.03996	0.03744	0.04594	0.04594	0.06262	0.05349	0.03084

MayB	0.29271	0.07318	0.04923	0.02717	0.03175	0.0382	0.04439	0.06053	0.03847	0.03175
JunA	0.25541	0.04929	0.04929	0.04353	0.04099	0.05297	0.04675	0.06863	0.04629	0.03961
JunB	0.28541	0.05957	0.0585	0.04093	0.03578	0.04864	0.04328	0.05828	0.03428	0.02957
JulA	0.26331	0.04598	0.04864	0.03267	0.04187	0.05203	0.05905	0.06752	0.04695	0.04235
JulB	0.28927	0.05317	0.0515	0.0456	0.03969	0.04338	0.0491	0.06	0.04264	0.03415
AugA	0.25428	0.05002	0.04609	0.0359	0.04169	0.05002	0.05442	0.07202	0.05025	0.03868
AugB	0.30981	0.05422	0.05815	0.04367	0.04594	0.04263	0.04367	0.05339	0.04284	0.03187
SepA	0.24232	0.04774	0.0399	0.03401	0.04186	0.05134	0.05298	0.06409	0.04251	0.03041
SepB	0.25446	0.05594	0.05347	0.04223	0.03208	0.0414	0.04497	0.05539	0.03482	0.03839

TTC [1] GMA Expected (percentage) distn. of Time (in days) to Contract (cont.)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.0559	0.04077	0.04756	0.05003	0.0315	0.05219	0.03212	0.0383	0.02409	0.01235
OctB	0.03504	0.03066	0.04495	0.03942	0.02628	0.04864	0.02674	0.03642	0.02075	0.00945
NovA	0.04309	0.03697	0.04279	0.04949	0.023	0.06055	0.02358	0.04396	0.02678	0.01281
NovB	0.05208	0.0292	0.0434	0.0334	0.02946	0.0505	0.02052	0.03235	0.02104	0.01578
DecA	0.03556	0.01895	0.03532	0.05661	0.02807	0.05357	0.02199	0.03485	0.02035	0.01216
DecB	0.0504	0.03586	0.04943	0.05428	0.02553	0.04943	0.02229	0.03554	0.01809	0.01712
JanA	0.05613	0.03581	0.0377	0.04377	0.02806	0.05131	0.02262	0.03518	0.02178	0.01361
JanB	0.04315	0.03406	0.04712	0.04514	0.02414	0.04315	0.02199	0.03026	0.01885	0.01009
FebA	0.04206	0.03308	0.04182	0.04608	0.02623	0.05317	0.02174	0.03828	0.02387	0.01134
FebB	0.03389	0.03333	0.05722	0.03778	0.02139	0.04583	0.02694	0.03278	0.01889	0.01167
MarA	0.04684	0.03546	0.03599	0.04234	0.02487	0.04975	0.02064	0.03334	0.01879	0.01111
MarB	0.03759	0.02918	0.04278	0.05218	0.0225	0.04797	0.02448	0.03314	0.02102	0.0089
AprA	0.03969	0.04118	0.04118	0.04984	0.02626	0.0564	0.02417	0.03133	0.02238	0.00895
AprB	0.03869	0.03721	0.04784	0.03898	0.02599	0.04637	0.02097	0.02599	0.01861	0.00797
MayA	0.04311	0.02926	0.03713	0.04405	0.02108	0.04877	0.02643	0.03524	0.01951	0.01101
MayB	0.03901	0.03901	0.04412	0.0417	0.02314	0.04412	0.01668	0.03336	0.02233	0.00915
JunA	0.04146	0.02718	0.03593	0.04744	0.02672	0.05136	0.01658	0.03201	0.01935	0.00921
JunB	0.03214	0.02678	0.04693	0.04371	0.02507	0.04821	0.02271	0.03193	0.01864	0.00964
JulA	0.0438	0.03533	0.03727	0.04259	0.02638	0.0455	0.01621	0.02735	0.01864	0.00653
JulB	0.03821	0.03231	0.03932	0.03914	0.02012	0.04615	0.01772	0.03194	0.01717	0.00941
AugA	0.05164	0.03312	0.04284	0.04099	0.02409	0.04076	0.01783	0.02849	0.01853	0.00834
AugB	0.03725	0.03084	0.04305	0.03974	0.01759	0.03415	0.018	0.02897	0.01469	0.00952
SepA	0.04284	0.04088	0.03499	0.04644	0.03303	0.05461	0.02649	0.03336	0.02551	0.01472
SepB	0.03784	0.03647	0.04716	0.04661	0.02468	0.04826	0.03153	0.04113	0.02194	0.01124

TTC [2] GMB Expected (percentage) distribution of Time (in days) to Contract

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.16244	0.04455	0.05963	0.02193	0.04318	0.06717	0.05209	0.06511	0.0377	0.04044
OctB	0.2025	0.05218	0.07033	0.04878	0.03573	0.04821	0.0329	0.04878	0.04311	0.0346
NovA	0.17618	0.03309	0.04797	0.02647	0.04301	0.04715	0.04218	0.06865	0.04301	0.02647
NovB	0.22505	0.03642	0.0488	0.03059	0.03714	0.05317	0.03787	0.06336	0.04661	0.03059
DecA	0.18924	0.05127	0.05316	0.04937	0.02975	0.04241	0.04177	0.06203	0.04873	0.02278
DecB	0.22251	0.05756	0.03866	0.01976	0.01546	0.03522	0.0378	0.0567	0.04038	0.02577
JanA	0.17389	0.04225	0.05417	0.02329	0.03196	0.05309	0.052	0.06392	0.04225	0.03954
JanB	0.2	0.05086	0.05991	0.03405	0.04009	0.04828	0.0444	0.05647	0.04483	0.02371
FebA	0.18095	0.03581	0.04731	0.03836	0.02749	0.04604	0.0454	0.07417	0.05115	0.03645
FebB	0.21713	0.05391	0.06499	0.04136	0.04136	0.0421	0.04136	0.04874	0.03914	0.03102
MarA	0.17672	0.03664	0.04741	0.03664	0.04023	0.04454	0.04813	0.06466	0.05603	0.03376

MarB	0.22686	0.05583	0.05018	0.0523	0.0424	0.04311	0.04311	0.04099	0.03322	0.02968
AprA	0.18641	0.04132	0.05418	0.045	0.03949	0.05601	0.0551	0.05601	0.04867	0.03857
AprB	0.2235	0.0378	0.06656	0.04848	0.03862	0.05259	0.04848	0.04191	0.05259	0.03451
MayA	0.17306	0.04584	0.06174	0.03742	0.03742	0.03648	0.05706	0.06642	0.03461	0.02339
MayB	0.19243	0.06361	0.04911	0.02818	0.03382	0.04589	0.03865	0.05153	0.03301	0.03543
JunA	0.14188	0.04711	0.05619	0.02951	0.03575	0.05108	0.0454	0.0681	0.05562	0.04313
JunB	0.19741	0.05833	0.05334	0.04237	0.04686	0.05085	0.05384	0.06082	0.03589	0.02493
JulA	0.17356	0.04077	0.0332	0.04193	0.03844	0.06174	0.05416	0.07338	0.05183	0.03262
JulB	0.18063	0.05315	0.0509	0.04144	0.0482	0.05045	0.06532	0.06486	0.03829	0.03739
AugA	0.20252	0.0461	0.06312	0.02854	0.0472	0.05873	0.05708	0.06751	0.04061	0.03293
AugB	0.23144	0.04918	0.05159	0.03568	0.04195	0.05207	0.04629	0.07136	0.03568	0.03616
SepA	0.1753	0.04826	0.05039	0.03265	0.04329	0.04613	0.0511	0.06884	0.05039	0.03123
SepB	0.16089	0.0484	0.05298	0.03859	0.03793	0.05101	0.05232	0.05298	0.0412	0.03859

TTC [2] GMB Expected (percentage) distn. of Time (in days) to Contract (cont)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.05894	0.04044	0.04181	0.05894	0.03701	0.05346	0.03153	0.03907	0.0329	0.01165
OctB	0.03971	0.02779	0.03517	0.06069	0.03914	0.05615	0.03347	0.04538	0.0329	0.01248
NovA	0.03722	0.02812	0.04301	0.07775	0.03639	0.08437	0.03391	0.05707	0.03474	0.01323
NovB	0.0386	0.02695	0.04006	0.06336	0.03933	0.06992	0.02768	0.04443	0.0284	0.01165
DecA	0.03228	0.02089	0.03354	0.08038	0.04051	0.07722	0.03544	0.04684	0.03038	0.01203
DecB	0.03952	0.04038	0.05498	0.09794	0.03866	0.06787	0.02491	0.04725	0.0232	0.01546
JanA	0.05688	0.03684	0.03521	0.08017	0.04171	0.07042	0.02438	0.04063	0.02384	0.01354
JanB	0.04009	0.03491	0.05043	0.07888	0.03707	0.06595	0.01509	0.03233	0.02802	0.01466
FebA	0.05563	0.04476	0.039	0.07928	0.03133	0.06394	0.02494	0.0422	0.02238	0.01343
FebB	0.03545	0.0325	0.05613	0.06204	0.03471	0.05835	0.01994	0.03545	0.0288	0.01551
MarA	0.05101	0.04598	0.03807	0.05891	0.03664	0.06681	0.03233	0.04741	0.02658	0.01149
MarB	0.04735	0.02615	0.04311	0.06714	0.03604	0.05159	0.03746	0.03534	0.02686	0.01131
AprA	0.04867	0.03949	0.03214	0.07163	0.03857	0.05142	0.03306	0.0303	0.02571	0.00826
AprB	0.02794	0.02301	0.05177	0.06738	0.02712	0.06081	0.02629	0.03287	0.02547	0.01233
MayA	0.04116	0.04864	0.03929	0.07203	0.04303	0.06361	0.02713	0.03648	0.03835	0.01684
MayB	0.03221	0.03784	0.04589	0.09259	0.02496	0.07649	0.02738	0.04428	0.0306	0.0161
JunA	0.04824	0.04881	0.03405	0.08229	0.04427	0.06697	0.01986	0.0403	0.03348	0.00795
JunB	0.04088	0.03888	0.04387	0.07926	0.03589	0.05533	0.01446	0.0334	0.02094	0.01246
JulA	0.0431	0.04659	0.0396	0.0827	0.03611	0.06174	0.01165	0.03727	0.02621	0.0134
JulB	0.04099	0.02928	0.0455	0.08378	0.03784	0.05225	0.01396	0.03243	0.02477	0.00856
AugA	0.0472	0.03238	0.04281	0.06476	0.03458	0.04501	0.01207	0.04226	0.02086	0.01372
AugB	0.04243	0.03423	0.04966	0.06847	0.03134	0.04581	0.02218	0.02845	0.01832	0.00771
SepA	0.03833	0.03265	0.06033	0.05749	0.0362	0.05323	0.03691	0.04258	0.03336	0.01136
SepB	0.05101	0.04709	0.05625	0.06017	0.02878	0.05298	0.03074	0.0484	0.03663	0.01308

TTC [3] GFA Expected (percentage) distribution of Time (in days) to Contract

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.16782	0.05287	0.04713	0.01954	0.03908	0.05287	0.05862	0.07356	0.04253	0.03678
OctB	0.18312	0.0557	0.06582	0.04557	0.03882	0.04641	0.05232	0.06245	0.05232	0.03544
NovA	0.16918	0.04403	0.04287	0.03013	0.03592	0.04171	0.05214	0.06952	0.073	0.03824
NovB	0.20437	0.06011	0.04809	0.03934	0.04153	0.04809	0.05246	0.06011	0.04153	0.03825
DecA	0.17408	0.05866	0.04447	0.04447	0.04352	0.05203	0.05109	0.07001	0.04068	0.03406
DecB	0.2249	0.04956	0.0432	0.02668	0.01906	0.02668	0.03939	0.05083	0.04701	0.03558
JanA	0.15851	0.04801	0.0308	0.03351	0.04076	0.06159	0.04891	0.07156	0.05254	0.03714

JanB	0.20273	0.05679	0.05607	0.04026	0.04313	0.05464	0.04242	0.06039	0.05104	0.03235
FebA	0.14599	0.03858	0.04171	0.0365	0.03858	0.05109	0.04797	0.08759	0.04275	0.02711
FebB	0.24539	0.03687	0.04839	0.03341	0.04378	0.05415	0.03687	0.05991	0.03571	0.03571
MarA	0.1639	0.03987	0.04319	0.03987	0.03876	0.03876	0.06091	0.07198	0.06312	0.03212
MarB	0.24404	0.04312	0.04771	0.0422	0.03853	0.0422	0.04037	0.05963	0.0422	0.03211
AprA	0.18141	0.03583	0.03471	0.04367	0.03359	0.04815	0.05039	0.06831	0.04703	0.03695
AprB	0.21479	0.05268	0.05876	0.0385	0.04255	0.04559	0.04154	0.05066	0.05066	0.04053
MayA	0.17894	0.03398	0.04757	0.03511	0.03737	0.04304	0.05549	0.07361	0.05549	0.03964
MayB	0.22843	0.07647	0.05196	0.03824	0.03235	0.0451	0.04804	0.05294	0.04314	0.03725
JunA	0.14992	0.04338	0.04947	0.04338	0.03881	0.05784	0.04718	0.06545	0.05099	0.0449
JunB	0.21694	0.05349	0.05423	0.03863	0.03863	0.05201	0.03492	0.05572	0.04755	0.04383
JulA	0.19503	0.04048	0.046	0.02852	0.03772	0.05612	0.0506	0.0736	0.04692	0.03956
JulB	0.213	0.04495	0.03665	0.04357	0.04495	0.05602	0.05602	0.06501	0.0574	0.03804
AugA	0.16904	0.04715	0.04626	0.03648	0.02491	0.05249	0.06139	0.08007	0.0516	0.03648
AugB	0.24472	0.0602	0.05473	0.03597	0.03831	0.04066	0.06411	0.06489	0.04378	0.02189
SepA	0.14375	0.04	0.055	0.01375	0.0425	0.05625	0.045	0.07875	0.045	0.0375
SepB	0.18379	0.04995	0.06221	0.03487	0.03676	0.04713	0.0443	0.06315	0.04053	0.02828

TTC [3] GFA Expected (percentage) distn. of Time (in days) to Contract (cont)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.04828	0.05057	0.04138	0.05632	0.03563	0.07011	0.04368	0.04138	0.01264	0.0092
OctB	0.03713	0.03207	0.05823	0.04895	0.03038	0.05992	0.02532	0.03797	0.0211	0.01097
NovA	0.03708	0.03592	0.0394	0.0533	0.02897	0.09038	0.03013	0.04635	0.02781	0.0139
NovB	0.04918	0.03934	0.05574	0.04372	0.02842	0.06011	0.02077	0.03825	0.02186	0.00874
DecA	0.04163	0.02649	0.04068	0.06339	0.04825	0.05771	0.03217	0.03974	0.02176	0.01514
DecB	0.05337	0.05591	0.06353	0.07116	0.02414	0.06607	0.03685	0.0305	0.02287	0.01271
JanA	0.06703	0.0462	0.04801	0.0625	0.02899	0.0625	0.03261	0.03986	0.01812	0.01087
JanB	0.03738	0.03235	0.05176	0.0532	0.04026	0.06326	0.02229	0.03019	0.01797	0.0115
FebA	0.05839	0.04171	0.05214	0.08133	0.05005	0.07091	0.02607	0.0365	0.02086	0.00417
FebB	0.0553	0.02535	0.0576	0.04147	0.03571	0.0576	0.02419	0.03802	0.02765	0.00691
MarA	0.05316	0.03987	0.05759	0.05648	0.03765	0.06645	0.03544	0.03322	0.01772	0.00997
MarB	0.03853	0.03028	0.06147	0.05596	0.03486	0.05872	0.02569	0.03303	0.02018	0.00917
AprA	0.06271	0.04255	0.05151	0.06831	0.03471	0.07839	0.02464	0.028	0.01344	0.01568
AprB	0.03445	0.03951	0.05572	0.05978	0.03445	0.05066	0.0304	0.02837	0.01824	0.01216
MayA	0.05549	0.04077	0.04417	0.05436	0.03964	0.06795	0.03624	0.02492	0.02831	0.00793
MayB	0.04314	0.03824	0.05098	0.04706	0.02745	0.07255	0.01961	0.01765	0.02059	0.00882
JunA	0.05632	0.03881	0.03577	0.05632	0.04871	0.07154	0.03196	0.03272	0.02207	0.01446
JunB	0.04383	0.04532	0.05721	0.05795	0.02675	0.05944	0.0208	0.02675	0.02155	0.00446
JulA	0.0414	0.04048	0.0598	0.05244	0.03864	0.069	0.0276	0.02944	0.01656	0.01012
JulB	0.03734	0.03942	0.04219	0.06155	0.03458	0.04979	0.02006	0.02559	0.02697	0.00692
AugA	0.04181	0.03381	0.06495	0.05249	0.04093	0.0605	0.0258	0.04804	0.01868	0.00712
AugB	0.04457	0.04144	0.05942	0.05942	0.01798	0.0516	0.01095	0.02267	0.01564	0.00704
SepA	0.0575	0.0425	0.05125	0.065	0.045	0.05125	0.03875	0.0475	0.03125	0.0125
SepB	0.05938	0.03676	0.04713	0.05655	0.02639	0.06598	0.03676	0.04336	0.02733	0.00943

TTC [4] GFB Expected (percentage) distribution of Time (in days) to Contract

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.08019	0.03302	0.02358	0.01651	0.00943	0.03066	0.01179	0.04717	0.03774	0.01887
OctB	0.11864	0.0339	0.03202	0.02072	0.0339	0.03013	0.03578	0.05838	0.06026	0.04331
NovA	0.1186	0.03235	0.03235	0.00809	0.02695	0.02156	0.02695	0.03235	0.02426	0.02965

NovB	0.11634	0.03324	0.03324	0.01662	0.04155	0.04709	0.03878	0.06648	0.03324	0.03324
DecA	0.12469	0.02494	0.02993	0.02494	0.01995	0.03242	0.0399	0.04489	0.04988	0.01496
DecB	0.13016	0.04444	0.01905	0.0127	0.02222	0.04762	0.03175	0.03492	0.04762	0.0381
JanA	0.10941	0.03282	0.03282	0.03063	0.03063	0.03939	0.03939	0.05689	0.0372	0.03282
JanB	0.1297	0.04511	0.03383	0.0282	0.03571	0.04323	0.04511	0.06203	0.03947	0.03195
FebA	0.08589	0.00307	0.02761	0.03067	0.01227	0.03374	0.03681	0.05215	0.03988	0.02147
FebB	0.13929	0.01071	0.025	0.02143	0.03214	0.03929	0.03571	0.04643	0.03214	0.02857
MarA	0.08869	0.02752	0.01529	0.02446	0.02141	0.03364	0.03364	0.07951	0.0367	0.04587
MarB	0.11142	0.0195	0.02786	0.039	0.02228	0.03621	0.04178	0.02507	0.03064	0.03343
AprA	0.06333	0.02333	0.04333	0.03	0.02333	0.02333	0.03	0.04667	0.03	0.03667
AprB	0.09756	0.02439	0.02134	0.02744	0.02744	0.02134	0.02744	0.03354	0.03049	0.02744
MayA	0.08667	0.02667	0.02	0.01667	0.03	0.02667	0.03	0.05333	0.03667	0.03667
MayB	0.14655	0.01149	0.02586	0.00575	0.01437	0.02299	0.03161	0.06034	0.02586	0.0431
JunA	0.10938	0.02734	0.0293	0.01563	0.01172	0.03516	0.02734	0.05664	0.03906	0.02344
JunB	0.10906	0.04067	0.02957	0.02403	0.01664	0.02773	0.03512	0.04621	0.03697	0.03512
JulA	0.09574	0.02979	0.01915	0.02979	0.04255	0.04681	0.04255	0.04468	0.02128	0.01915
JulB	0.11917	0.03972	0.038	0.02418	0.02591	0.04318	0.038	0.04318	0.04836	0.03282
AugA	0.09917	0.03512	0.02273	0.02479	0.0124	0.04339	0.05579	0.04339	0.06818	0.03306
AugB	0.12218	0.0282	0.03008	0.01316	0.03195	0.04135	0.03947	0.05451	0.05075	0.02068
SepA	0.09471	0.02228	0.0195	0.01393	0.0195	0.02507	0.01671	0.02786	0.03064	0.0585
SepB	0.06682	0.02304	0.00922	0.02304	0.02535	0.02074	0.05991	0.05991	0.03226	0.05069

TTC [4] GFB Expected (percentage) distn. of Time (in days) to Contract (cont)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.04009	0.0684	0.0566	0.10142	0.07075	0.12264	0.04481	0.10849	0.04717	0.03066
OctB	0.05461	0.02448	0.03202	0.13183	0.04896	0.11676	0.02637	0.06403	0.02072	0.01318
NovA	0.04043	0.02965	0.08356	0.12938	0.09164	0.10512	0.03235	0.07278	0.03774	0.02426
NovB	0.03601	0.04709	0.0554	0.12188	0.04986	0.09418	0.03878	0.04155	0.03878	0.01662
DecA	0.03741	0.03242	0.06234	0.14464	0.05237	0.11721	0.01247	0.06733	0.03741	0.02993
DecB	0.06349	0.05397	0.06032	0.11429	0.0381	0.08571	0.03492	0.06667	0.04444	0.00952
JanA	0.05689	0.04814	0.03282	0.08972	0.07659	0.08753	0.02845	0.07659	0.03501	0.02626
JanB	0.03383	0.02068	0.03947	0.10902	0.04511	0.08271	0.03008	0.07519	0.04699	0.02256
FebA	0.04601	0.03988	0.03374	0.14417	0.06748	0.11043	0.03374	0.11043	0.04908	0.02147
FebB	0.03214	0.03929	0.03214	0.12857	0.03929	0.12857	0.04643	0.08214	0.03571	0.025
MarA	0.03976	0.0367	0.04587	0.11621	0.06422	0.09174	0.04587	0.08869	0.04281	0.02141
MarB	0.03343	0.03343	0.04178	0.1337	0.06685	0.08635	0.07242	0.06407	0.04735	0.03343
AprA	0.03333	0.03333	0.04	0.14667	0.09333	0.09333	0.07	0.08333	0.04	0.01667
AprB	0.03354	0.03963	0.04268	0.15854	0.06402	0.12805	0.04878	0.07012	0.04878	0.02744
MayA	0.03667	0.03	0.07	0.13667	0.07	0.12	0.04667	0.07	0.03333	0.02333
MayB	0.02299	0.04023	0.06034	0.14655	0.05747	0.11207	0.05747	0.06034	0.03736	0.01724
JunA	0.04102	0.06055	0.03906	0.125	0.0625	0.12109	0.04492	0.0625	0.04492	0.02344
JunB	0.04991	0.05176	0.03882	0.13124	0.06285	0.1146	0.04251	0.0536	0.03142	0.02218
JulA	0.0617	0.04468	0.0383	0.10426	0.04255	0.14681	0.05745	0.05532	0.04043	0.01702
JulB	0.038	0.02763	0.03972	0.10708	0.06045	0.12953	0.03972	0.05354	0.03282	0.019
AugA	0.03719	0.04752	0.03926	0.1281	0.08884	0.08471	0.02479	0.04752	0.04545	0.0186
AugB	0.03947	0.03383	0.04323	0.1391	0.05451	0.1109	0.03195	0.04887	0.04323	0.02256
SepA	0.03064	0.05292	0.06685	0.11978	0.10585	0.13092	0.03621	0.06964	0.02786	0.03064
SepB	0.05069	0.03456	0.06452	0.13364	0.07604	0.10369	0.0553	0.05069	0.04147	0.01843

TTC [5]	SMT Expected (percentage) distribution of Time (in days) to Contract									
week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.16684	0.02406	0.03021	0.02353	0.02433	0.03663	0.04706	0.06497	0.05374	0.02888
OctB	0.18789	0.02751	0.04649	0.03817	0.03498	0.03519	0.03242	0.05204	0.04201	0.0305
NovA	0.16224	0.02971	0.03339	0.02814	0.02787	0.03734	0.04049	0.06258	0.05312	0.0305
NovB	0.19026	0.0274	0.04018	0.02861	0.03166	0.03957	0.03318	0.05145	0.04201	0.02861
DecA	0.16386	0.02435	0.04179	0.04179	0.03909	0.03578	0.03909	0.06164	0.05292	0.02435
DecB	0.19292	0.03286	0.03244	0.02177	0.01579	0.02219	0.02988	0.05591	0.04012	0.03073
JanA	0.14965	0.02689	0.02884	0.02221	0.01949	0.0378	0.04326	0.06625	0.06235	0.04053
JanB	0.18491	0.02583	0.0508	0.03222	0.03019	0.03774	0.03687	0.05486	0.04673	0.03396
FebA	0.15032	0.03129	0.04097	0.02871	0.02968	0.03742	0.03742	0.06548	0.05516	0.03839
FebB	0.20192	0.03065	0.05134	0.03985	0.03103	0.04943	0.03372	0.05517	0.041	0.03793
MarA	0.17041	0.02382	0.03724	0.03287	0.03724	0.04797	0.04562	0.07313	0.05971	0.03355
MarB	0.20541	0.03574	0.05248	0.05023	0.03799	0.04057	0.03896	0.05505	0.0425	0.02511
AprA	0.17642	0.03234	0.04544	0.03029	0.03357	0.04052	0.05076	0.0659	0.04789	0.0352
AprB	0.20832	0.03867	0.05239	0.03784	0.03368	0.04324	0.04283	0.05364	0.03742	0.03035
MayA	0.19884	0.03234	0.04199	0.03571	0.03185	0.03137	0.03764	0.05647	0.05743	0.02703
MayB	0.20575	0.03797	0.03899	0.02873	0.01642	0.02258	0.02514	0.03438	0.0195	0.02463
JunA	0.05148	0.01135	0.01946	0.01378	0.01743	0.03202	0.04054	0.07702	0.06323	0.0681
JunB	0.15977	0.02967	0.03652	0.02576	0.03587	0.03521	0.03195	0.04076	0.03326	0.02478
JulA	0.14099	0.02042	0.02851	0.02388	0.02619	0.03274	0.03621	0.06279	0.04006	0.03775
JulB	0.17171	0.0255	0.03707	0.03114	0.02372	0.03974	0.0344	0.05338	0.03529	0.02906
AugA	0.13881	0.02963	0.03768	0.02383	0.02383	0.04348	0.04412	0.06119	0.0467	0.03092
AugB	0.18196	0.03417	0.04126	0.02986	0.02217	0.03818	0.03571	0.04895	0.03849	0.03664
SepA	0.15153	0.02386	0.0293	0.02093	0.02009	0.0293	0.04521	0.05776	0.04228	0.036
SepB	0.17813	0.02549	0.03206	0.0359	0.03343	0.03809	0.03727	0.05426	0.04111	0.0296

TTC [5]	SMT Expected (percentage) distn. of Time (in days) to Contract (cont)									
week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.05107	0.03636	0.04305	0.05401	0.03583	0.07807	0.03529	0.08396	0.05428	0.02781
OctB	0.03391	0.02986	0.04457	0.05097	0.03626	0.06867	0.03604	0.08573	0.05865	0.02815
NovA	0.04418	0.03234	0.03629	0.061	0.03182	0.07626	0.03918	0.08888	0.05548	0.02919
NovB	0.04718	0.03379	0.04627	0.05632	0.02435	0.07945	0.03866	0.07915	0.04962	0.03227
DecA	0.03428	0.02495	0.03548	0.06404	0.046	0.06915	0.03939	0.0887	0.0451	0.02826
DecB	0.03414	0.04183	0.05378	0.09006	0.03073	0.07042	0.03713	0.08963	0.05676	0.02091
JanA	0.06352	0.04209	0.03975	0.05612	0.03936	0.07366	0.04014	0.06937	0.053	0.02572
JanB	0.05138	0.02729	0.05341	0.05951	0.04122	0.06125	0.03657	0.06792	0.0447	0.02264
FebA	0.05097	0.04419	0.04065	0.06484	0.03742	0.06258	0.04516	0.07387	0.04194	0.02355
FebB	0.03793	0.03448	0.04789	0.059	0.03257	0.05785	0.03487	0.05747	0.04674	0.01916
MarA	0.05502	0.02986	0.03623	0.05904	0.03925	0.07581	0.0426	0.04864	0.0322	0.01979
MarB	0.0454	0.02769	0.04797	0.05666	0.02737	0.07888	0.03477	0.05087	0.03026	0.0161
AprA	0.04666	0.03889	0.03643	0.05444	0.03725	0.09374	0.03316	0.04298	0.04216	0.01596
AprB	0.03576	0.02994	0.0474	0.0499	0.03534	0.08067	0.042	0.04657	0.03576	0.0183
MayA	0.03668	0.01641	0.02606	0.09797	0.04681	0.07867	0.03137	0.04585	0.0473	0.0222
MayB	0.0313	0.0549	0.07953	0.08825	0.03899	0.09389	0.0354	0.04566	0.04926	0.02873
JunA	0.06486	0.04864	0.04256	0.06891	0.04175	0.11958	0.03486	0.06648	0.08634	0.03162
JunB	0.03097	0.03456	0.04891	0.06847	0.03978	0.10694	0.02674	0.07532	0.09032	0.02445
JulA	0.04468	0.0416	0.04468	0.06664	0.05855	0.09091	0.03005	0.06703	0.08128	0.02504
JulB	0.03381	0.03529	0.0433	0.07533	0.03707	0.08066	0.03203	0.07058	0.08155	0.02936
AugA	0.04444	0.04767	0.05346	0.0686	0.04509	0.06087	0.03671	0.06763	0.06634	0.02899

AugB	0.04711	0.0311	0.05665	0.07482	0.03695	0.05788	0.03264	0.07235	0.05634	0.02679
SepA	0.05274	0.03893	0.04772	0.0833	0.03516	0.07158	0.03809	0.0879	0.06153	0.02679
SepB	0.04111	0.03343	0.05207	0.06632	0.037	0.06577	0.03617	0.08221	0.05591	0.02466

TTC [6] SFT Expected (percentage) distribution of Time (in days) to Contract

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.10116	0.03179	0.0289	0.01734	0.03179	0.02601	0.02601	0.06647	0.04046	0.02023
OctB	0.12277	0.01786	0.02679	0.02679	0.01786	0.02902	0.03795	0.04241	0.03125	0.02679
NovA	0.04974	0.02356	0.02356	0.01832	0.02618	0.02094	0.04188	0.06806	0.05236	0.05236
NovB	0.10826	0.02279	0.02849	0.02849	0.05128	0.02849	0.02279	0.0114	0.02849	0.03419
DecA	0.06905	0.0179	0.03836	0.02302	0.02046	0.04348	0.04092	0.03069	0.03836	0.02302
DecB	0.09118	0.02941	0.03824	0.01176	0.01471	0.01471	0.03235	0.06471	0.03529	0.01471
JanA	0.05085	0.01977	0.0226	0.01695	0.0339	0.03672	0.03107	0.07627	0.05932	0.0452
JanB	0.13035	0.02235	0.04097	0.0298	0.02235	0.03538	0.03724	0.05214	0.04469	0.03724
FebA	0.08041	0.01649	0.03093	0.01443	0.02474	0.02887	0.03093	0.06598	0.06392	0.02887
FebB	0.14319	0.02079	0.03695	0.03233	0.04157	0.04157	0.02079	0.04157	0.03464	0.03464
MarA	0.09182	0.02595	0.03393	0.02794	0.03194	0.04591	0.02794	0.05788	0.04591	0.03593
MarB	0.12523	0.03178	0.04673	0.04299	0.01495	0.02991	0.04299	0.05234	0.04673	0.03178
AprA	0.11776	0.03282	0.03089	0.02317	0.04633	0.04247	0.04826	0.05792	0.05598	0.02703
AprB	0.1523	0.03206	0.03206	0.04008	0.04208	0.04008	0.03607	0.05611	0.04008	0.02004
MayA	0.12761	0.03016	0.0348	0.04408	0.0348	0.04176	0.03016	0.06961	0.04176	0.03016
MayB	0.1928	0.03342	0.03342	0.03085	0.02314	0.02057	0.02057	0.03599	0.01542	0.02571
JunA	0.0625	0	0.00481	0.00481	0.00962	0.03846	0.01923	0.03846	0.04808	0.05288
JunB	0.08571	0.02381	0.00476	0.02381	0.01429	0.03333	0.01905	0.04286	0.0381	0.00952
JulA	0.07692	0.02959	0.01775	0.01775	0.02959	0.02959	0.07692	0.0355	0.05917	0.02367
JulB	0.0936	0.03941	0.0197	0.01478	0.0197	0.02956	0.01478	0.04926	0.02463	0.00985
AugA	0.05319	0.01064	0.02128	0.01596	0.0266	0.01596	0.03723	0.04787	0.03723	0.01064
AugB	0.15745	0.02128	0.03404	0.01277	0.00851	0.03404	0.04255	0.05957	0.04681	0.02979
SepA	0.08075	0.01863	0.04348	0.00621	0.01242	0.01242	0.01863	0.0559	0.0559	0.0559
SepB	0.10714	0.01786	0.025	0.025	0.01786	0.03214	0.02857	0.05714	0.03571	0.02857

TTC [6] SFT Expected (percentage) distn. of Time (in days) to Contract (cont)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.04046	0.03757	0.04335	0.04913	0.04624	0.07225	0.06069	0.13006	0.08382	0.04624
OctB	0.02679	0.03348	0.08036	0.07366	0.03348	0.08036	0.06473	0.09598	0.08482	0.04688
NovA	0.04188	0.03141	0.05759	0.08901	0.0445	0.09424	0.03927	0.13089	0.06283	0.03141
NovB	0.03704	0.01994	0.04558	0.05983	0.07407	0.11681	0.06553	0.12821	0.05128	0.03704
DecA	0.02813	0.0179	0.03581	0.09207	0.07161	0.12532	0.06905	0.11509	0.06905	0.03069
DecB	0.05	0.03824	0.07353	0.08824	0.04412	0.11471	0.06765	0.08824	0.05588	0.03235
JanA	0.04802	0.03107	0.0565	0.06215	0.05085	0.15254	0.03955	0.0791	0.05932	0.02825
JanB	0.04469	0.02421	0.05959	0.04842	0.04842	0.10801	0.06518	0.07635	0.04655	0.02607
FebA	0.05979	0.02887	0.03093	0.07835	0.05567	0.11546	0.09278	0.0701	0.06186	0.02062
FebB	0.02771	0.03695	0.06697	0.09007	0.05543	0.10393	0.06005	0.06005	0.03464	0.01617
MarA	0.05589	0.03792	0.06188	0.08184	0.03393	0.1018	0.05788	0.07585	0.0499	0.01796
MarB	0.05794	0.03364	0.06729	0.05421	0.03551	0.11028	0.04673	0.0729	0.03364	0.02243
AprA	0.06371	0.03475	0.04054	0.0888	0.03089	0.09266	0.03668	0.05019	0.04633	0.03282
AprB	0.04609	0.03407	0.04008	0.07415	0.04609	0.11222	0.03607	0.06212	0.03006	0.02806
MayA	0.02784	0.01624	0.01392	0.10905	0.04872	0.10209	0.05336	0.058	0.06497	0.02088
MayB	0.0437	0.04884	0.06427	0.09769	0.04627	0.1054	0.04884	0.05913	0.03342	0.02057
JunA	0.08173	0.02404	0.02404	0.07692	0.0625	0.19231	0.05288	0.09135	0.08654	0.02885

JunB	0.03333	0.01905	0.0381	0.05714	0.05714	0.18095	0.04762	0.11429	0.10476	0.05238
JulA	0.02959	0.0355	0.0355	0.04734	0.08876	0.13609	0.01775	0.08876	0.08876	0.0355
JulB	0.03941	0.02956	0.05419	0.06897	0.07389	0.1133	0.03941	0.0936	0.10837	0.06404
AugA	0.03723	0.03723	0.03723	0.14894	0.07447	0.12234	0.04255	0.12234	0.07979	0.02128
AugB	0.05957	0.02979	0.07234	0.11489	0.03404	0.07234	0.0383	0.07234	0.04255	0.01702
SepA	0.04969	0.04969	0.0559	0.09938	0.0559	0.12422	0.0559	0.06211	0.07453	0.01242
SepB	0.04286	0.025	0.03571	0.07143	0.03929	0.10714	0.03214	0.14286	0.06786	0.06071

TTC [7] OTH Expected (percentage) distribution of Time (in days) to Contract

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.09628	0.02376	0.02094	0.015	0.01938	0.0297	0.03063	0.05002	0.03751	0.03063
OctB	0.13745	0.03756	0.03156	0.03026	0.02347	0.02895	0.02608	0.04356	0.03443	0.03312
NovA	0.11057	0.02204	0.02204	0.0161	0.01819	0.02694	0.02939	0.04129	0.03744	0.02519
NovB	0.14746	0.03094	0.03221	0.01516	0.02526	0.03063	0.02937	0.04326	0.03189	0.02652
DecA	0.13225	0.02802	0.0367	0.02578	0.02998	0.02774	0.03054	0.05015	0.03586	0.02578
DecB	0.17721	0.02721	0.02757	0.01581	0.01618	0.01765	0.02831	0.03529	0.03051	0.02279
JanA	0.11765	0.03557	0.03228	0.01997	0.02244	0.02818	0.03064	0.06731	0.04268	0.03447
JanB	0.17705	0.03896	0.03704	0.03361	0.02933	0.02997	0.02997	0.04539	0.03789	0.0289
FebA	0.14926	0.02669	0.02834	0.01746	0.01549	0.0168	0.03229	0.03987	0.03624	0.0257
FebB	0.1462	0.02924	0.033	0.02506	0.02464	0.03551	0.02548	0.03718	0.02172	0.02632
MarA	0.10186	0.02623	0.02471	0.01976	0.02357	0.02775	0.01938	0.03839	0.03915	0.02547
MarB	0.12531	0.02662	0.03372	0.02662	0.02378	0.0323	0.02556	0.03585	0.0323	0.0252
AprA	0.11252	0.03261	0.03261	0.02731	0.02568	0.02854	0.03057	0.04974	0.03914	0.03017
AprB	0.16045	0.02649	0.03993	0.03843	0.02276	0.03433	0.02985	0.03657	0.03433	0.02873
MayA	0.13327	0.01898	0.02471	0.01721	0.0203	0.02118	0.01677	0.03575	0.03222	0.01898
MayB	0.16256	0.028	0.02842	0.01337	0.01295	0.0188	0.02382	0.03218	0.02089	0.02089
JunA	0.10616	0.02453	0.02052	0.01829	0.02275	0.02186	0.0223	0.03613	0.02632	0.02498
JunB	0.125	0.02292	0.02682	0.02898	0.02249	0.016	0.0186	0.03157	0.02811	0.01817
JulA	0.11433	0.02097	0.01997	0.01548	0.02147	0.02896	0.03195	0.04593	0.03595	0.02197
JulB	0.13568	0.02787	0.0352	0.0319	0.022	0.0253	0.0264	0.0341	0.02897	0.02457
AugA	0.1279	0.02758	0.03237	0.01479	0.01559	0.03477	0.03157	0.04357	0.03437	0.03557
AugB	0.16443	0.03463	0.03058	0.02933	0.02465	0.0234	0.03557	0.04555	0.04056	0.02465
SepA	0.08612	0.02828	0.015	0.01757	0.01243	0.02442	0.02614	0.03385	0.03042	0.02656
SepB	0.12828	0.0242	0.02178	0.02248	0.02213	0.03285	0.0287	0.04426	0.03458	0.02663

TTC [7] OTH Expected (percentage) distn. of Time (in days) to Contract (cont)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.03314	0.04033	0.04877	0.09097	0.05846	0.11816	0.04533	0.09909	0.06252	0.04939
OctB	0.03156	0.02869	0.03704	0.07407	0.04982	0.10433	0.04956	0.09755	0.05921	0.04173
NovA	0.03464	0.03814	0.04024	0.08782	0.04899	0.12456	0.05108	0.10812	0.06648	0.05073
NovB	0.03663	0.03473	0.04294	0.07925	0.0562	0.11178	0.04578	0.08967	0.0581	0.03221
DecA	0.02886	0.02382	0.04511	0.08714	0.06444	0.09611	0.04847	0.08742	0.05043	0.04539
DecB	0.03676	0.04449	0.05588	0.10588	0.04449	0.10404	0.04007	0.08162	0.04779	0.04044
JanA	0.04378	0.04378	0.04733	0.08372	0.04514	0.09549	0.03338	0.09275	0.04596	0.03748
JanB	0.03896	0.02355	0.03982	0.07236	0.05117	0.08478	0.03276	0.07836	0.04731	0.04282
FebA	0.03262	0.02438	0.03987	0.09588	0.06656	0.10939	0.057	0.0972	0.04811	0.04086
FebB	0.02757	0.02506	0.04887	0.11947	0.05681	0.08605	0.05597	0.08855	0.04219	0.04511
MarA	0.04029	0.03421	0.04485	0.11403	0.0688	0.10034	0.06195	0.08058	0.06461	0.04409
MarB	0.02911	0.03088	0.05112	0.10827	0.04437	0.10224	0.06106	0.0891	0.06354	0.03301
AprA	0.04117	0.03995	0.04036	0.07868	0.03587	0.12352	0.0636	0.06523	0.06237	0.04036

AprB	0.02948	0.01754	0.03694	0.06642	0.05448	0.12425	0.06306	0.05821	0.06269	0.03507
MayA	0.03486	0.02207	0.03354	0.10282	0.07855	0.13989	0.06178	0.07855	0.06708	0.04148
MayB	0.02925	0.02507	0.05307	0.10071	0.05934	0.12704	0.06059	0.07438	0.06812	0.04053
JunA	0.03211	0.03167	0.03078	0.10839	0.07583	0.14719	0.04906	0.06869	0.08742	0.04505
JunB	0.03503	0.03114	0.0359	0.10943	0.06228	0.13668	0.05104	0.07958	0.0731	0.04715
JulA	0.03944	0.03195	0.04044	0.11383	0.07389	0.12681	0.04244	0.0684	0.0659	0.03994
JulB	0.03007	0.0253	0.0374	0.11771	0.05574	0.11551	0.04474	0.06967	0.07407	0.03777
AugA	0.03717	0.03677	0.04117	0.09392	0.07954	0.09193	0.05116	0.06515	0.06715	0.03797
AugB	0.03401	0.02652	0.03931	0.10328	0.05148	0.0936	0.05179	0.05928	0.05023	0.03713
SepA	0.03856	0.04327	0.05741	0.10283	0.0587	0.12982	0.07498	0.09126	0.06641	0.03599
SepB	0.03769	0.03458	0.04391	0.10719	0.05394	0.1065	0.05083	0.08402	0.06086	0.03458

TTC [8] TOT Expected (percentage) distribution of Time (in days) to Contract

week	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8_-9	10_-12	13-14
OctA	0.15871	0.03427	0.0345	0.01913	0.02779	0.04263	0.04339	0.06177	0.04467	0.03344
OctB	0.19317	0.0417	0.04938	0.03961	0.03461	0.03759	0.03425	0.05129	0.04164	0.03228
NovA	0.16462	0.03421	0.03576	0.026	0.02933	0.03761	0.04032	0.05975	0.04806	0.03266
NovB	0.20696	0.03765	0.04195	0.02746	0.0341	0.0418	0.03622	0.05002	0.04044	0.03108
DecA	0.18399	0.03822	0.0461	0.04007	0.03637	0.03726	0.04082	0.05795	0.04418	0.02651
DecB	0.21516	0.04134	0.03549	0.02192	0.01895	0.02564	0.03447	0.05017	0.03948	0.0301
JanA	0.1669	0.03856	0.04052	0.02589	0.03137	0.04113	0.04479	0.06831	0.04974	0.03653
JanB	0.2055	0.0445	0.04983	0.037	0.03436	0.04276	0.03938	0.05479	0.0435	0.03109
FebA	0.1748	0.03664	0.04036	0.02898	0.02847	0.03642	0.04146	0.06664	0.04861	0.03211
FebB	0.21458	0.0396	0.05165	0.037	0.03441	0.04376	0.03475	0.04862	0.03536	0.03285
MarA	0.17381	0.03284	0.04004	0.03276	0.0342	0.04195	0.04299	0.06361	0.05402	0.0314
MarB	0.21151	0.0407	0.04983	0.04422	0.03374	0.03831	0.03928	0.04698	0.03711	0.02746
AprA	0.18313	0.03838	0.04155	0.03376	0.03331	0.04146	0.04454	0.0602	0.04553	0.03241
AprB	0.21527	0.04147	0.05225	0.04034	0.03512	0.04017	0.03982	0.04704	0.04112	0.03095
MayA	0.19541	0.03667	0.04147	0.03285	0.03216	0.03549	0.03863	0.05677	0.04677	0.02755
MayB	0.22391	0.05308	0.04187	0.02505	0.02442	0.03165	0.035	0.04721	0.02993	0.02921
JunA	0.1549	0.03511	0.03799	0.02973	0.03067	0.04281	0.03993	0.06321	0.04756	0.04312
JunB	0.20327	0.04509	0.04552	0.03477	0.03442	0.04	0.03682	0.04954	0.0347	0.02728
JulA	0.1849	0.03434	0.03565	0.02859	0.03417	0.04534	0.04789	0.063	0.04354	0.03524
JulB	0.20753	0.04159	0.04335	0.03827	0.0342	0.04147	0.04416	0.05456	0.03946	0.03188
AugA	0.18209	0.03936	0.04275	0.02762	0.03086	0.04637	0.04859	0.06298	0.04578	0.03478
AugB	0.22785	0.04434	0.04635	0.03453	0.03427	0.03862	0.04219	0.05433	0.0409	0.0307
SepA	0.16299	0.03616	0.03387	0.02455	0.0294	0.03853	0.04263	0.05633	0.04082	0.03283
SepB	0.18151	0.03773	0.0395	0.03432	0.03083	0.03936	0.04032	0.05322	0.03758	0.03291

TTC [8] TOT Expected (percentage) distn. of Time (in days) to Contract (cont)

week	15-18	19-22	23-30	31-45	46-60	61-120	121-180	181-365	366-730	> 731
OctA	0.04798	0.04083	0.04572	0.06403	0.04173	0.07947	0.03804	0.07073	0.04437	0.02682
OctB	0.03497	0.02966	0.04348	0.0573	0.03699	0.07154	0.03616	0.06767	0.04313	0.02359
NovA	0.04048	0.03459	0.04172	0.06772	0.0356	0.08583	0.03637	0.07616	0.0462	0.02701
NovB	0.04474	0.03237	0.04489	0.05696	0.03727	0.07794	0.0335	0.06209	0.03908	0.02346
DecA	0.03356	0.0226	0.0387	0.0722	0.04569	0.07405	0.03562	0.06466	0.0363	0.02514
DecB	0.04283	0.0419	0.05472	0.08389	0.03372	0.07414	0.03317	0.06243	0.03688	0.0236
JanA	0.05509	0.04005	0.0412	0.06363	0.03808	0.07291	0.02988	0.05875	0.03449	0.02216
JanB	0.0426	0.02956	0.04735	0.06107	0.03795	0.06393	0.02792	0.05178	0.0332	0.02191
FebA	0.0454	0.03562	0.04095	0.07109	0.04197	0.07379	0.03832	0.06255	0.03489	0.02095

FebB	0.03501	0.03146	0.05304	0.06682	0.03562	0.06344	0.03545	0.05286	0.03285	0.02089
MarA	0.04851	0.03548	0.04099	0.06776	0.04115	0.07288	0.03908	0.05162	0.03428	0.02062
MarB	0.03943	0.02925	0.04826	0.06921	0.0324	0.07138	0.03823	0.0517	0.03389	0.01713
AprA	0.04526	0.03983	0.03983	0.06536	0.03476	0.08355	0.03766	0.04336	0.03684	0.01928
AprB	0.0346	0.02973	0.04582	0.05738	0.03738	0.07877	0.03799	0.04156	0.03452	0.01869
MayA	0.04	0.02755	0.03491	0.07736	0.04559	0.08266	0.03794	0.04824	0.04079	0.02118
MayB	0.03482	0.03898	0.05435	0.07415	0.03626	0.08139	0.03337	0.04594	0.03861	0.0208
JunA	0.0474	0.03752	0.03596	0.07146	0.04484	0.09045	0.0295	0.04842	0.04842	0.02102
JunB	0.03541	0.0335	0.04566	0.06976	0.03795	0.08198	0.02799	0.05089	0.04544	0.02
JulA	0.04345	0.03852	0.04173	0.06842	0.04501	0.07812	0.02546	0.04608	0.04247	0.01807
JulB	0.03621	0.03176	0.04116	0.07129	0.03571	0.07035	0.02612	0.04761	0.04416	0.01917
AugA	0.04519	0.03766	0.04659	0.06586	0.04497	0.05974	0.0288	0.04969	0.04061	0.01971
AugB	0.04038	0.03142	0.04784	0.07043	0.03213	0.05777	0.02889	0.04518	0.03285	0.01902
SepA	0.04434	0.04053	0.04891	0.07355	0.04339	0.0784	0.04301	0.06261	0.0451	0.02207
SepB	0.0424	0.03617	0.04914	0.07056	0.03684	0.07041	0.03802	0.06485	0.04314	0.0212

3. MICROSOFT EXCEL PARAMETER FILE AVGDATA.XLS

DAILYTEST **Average Number of Prospects Tested Daily (Avg 91-93)**

Mo	DaysInMo	Avg Number of prospects tested each day of the month							
Oct	31	388	388	388	388	388	388	388	388
		388	388	388	388	388	388	388	428
		428	428	428	428	428	428	428	428
		428	428	428	428	428	428	428	
Nov	30	371	371	371	371	371	371	371	371
		371	371	371	371	371	371	371	354
		354	354	354	354	354	354	354	354
		354	354	354	354	354	354		
Dec	31	410	410	410	410	410	410	410	410
		410	410	410	410	410	410	410	279
		279	279	279	279	279	279	279	279
		279	279	279	279	279	279	279	
Jan	31	463	463	463	463	463	463	463	463
		463	463	463	463	463	463	463	539
		539	539	539	539	539	539	539	539
		539	539	539	539	539	539	539	
Feb	28	421	421	421	421	421	421	421	421
		421	421	421	421	421	421	421	376
		376	376	376	376	376	376	376	376
		376	376	376	376				
Mar	31	365	365	365	365	365	365	365	365
		365	365	365	365	365	365	365	352
		352	352	352	352	352	352	352	352
		352	352	352	352	352	352	352	
Apr	30	330	330	330	330	330	330	330	330
		330	330	330	330	330	330	330	328
		328	328	328	328	328	328	328	328
		328	328	328	328	328	328		
May	31	314	314	314	314	314	314	314	314
		314	314	314	314	314	314	314	282
		282	282	282	282	282	282	282	282
		282	282	282	282	282	282	282	
Jun	30	370	370	370	370	370	370	370	370
		370	370	370	370	370	370	370	402
		402	402	402	402	402	402	402	402
		402	402	402	402	402	402		

Jul	31	372	372	372	372	372	372	372	372
		372	372	372	372	372	372	372	427
		427	427	427	427	427	427	427	427
		427	427	427	427	427	427	427	
Aug	31	413	413	413	413	413	413	413	413
		413	413	413	413	413	413	413	407
		407	407	407	407	407	407	407	407
		407	407	407	407	407	407	407	
Sep	30	360	360	360	360	360	360	360	360
		360	360	360	360	360	360	360	434
		434	434	434	434	434	434	434	434
		434	434	434	434	434	434		

DAILYCONT **Average Number of Contracts Signed Daily (Avg 91-93)**

Mo	DaysInMo	Number of contracts signed each day of the month							
Oct	31	242	242	242	242	242	242	242	242
		242	242	242	242	242	242	242	288
		288	288	288	288	288	288	288	288
		288	288	288	288	288	288	288	
Nov	30	200	200	200	200	200	200	200	200
		200	200	200	200	200	200	200	251
		251	251	251	251	251	251	251	251
		251	251	251	251	251	251		
Dec	31	208	208	208	208	208	208	208	208
		208	208	208	208	208	208	208	218
		218	218	218	218	218	218	218	218
		218	218	218	218	218	218	218	
Jan	31	216	216	216	216	216	216	216	216
		216	216	216	216	216	216	216	299
		299	299	299	299	299	299	299	299
		299	299	299	299	299	299	299	
Feb	28	231	231	231	231	231	231	231	231
		231	231	231	231	231	231	231	260
		260	260	260	260	260	260	260	260
		260	260	260	260				
Mar	31	207	207	207	207	207	207	207	207
		207	207	207	207	207	207	207	260
		260	260	260	260	260	260	260	260
		260	260	260	260	260	260	260	
Apr	30	188	188	188	188	188	188	188	188

		188	188	188	188	188	188	188	247
		247	247	247	247	247	247	247	247
		247	247	247	247	247	247		
May	31	184	184	184	184	184	184	184	184
		184	184	184	184	184	184	184	209
		209	209	209	209	209	209	209	209
		209	209	209	209	209	209	209	
Jun	30	166	166	166	166	166	166	166	166
		166	166	166	166	166	166	166	300
		300	300	300	300	300	300	300	300
		300	300	300	300	300	300		
Jul	31	209	209	209	209	209	209	209	209
		209	209	209	209	209	209	209	302
		302	302	302	302	302	302	302	302
		302	302	302	302	302	302	302	
Aug	31	224	224	224	224	224	224	224	224
		224	224	224	224	224	224	224	285
		285	285	285	285	285	285	285	285
		285	285	285	285	285	285	285	
Sep	30	223	223	223	223	223	223	223	223
		223	223	223	223	223	223	223	295
		295	295	295	295	295	295	295	295
		295	295	295	295	295	295		

PCTNBOX **Percentage of Contracts signed that fall within each NBox. (89-93 data)**

NBox	Percent
GMA	0.3294
GMB	0.1527
GFA	0.0881
GFB	0.0393
SMT	0.2355
SFT	0.0245
OTH	0.1306
TOT	1.0000

APPENDIX C. SPSS CODE

The following SPSS commands were used to manipulate the Testing, Contract, and Composite data bases from their initial formats into the formats required to derive the parameters used in the model.

1. COMMANDS USED TO MANIPULATE THE TESTING DATA BASE

a. Read in the initial data fields from the PRO8993 text file (flat textfile containing the MEPCOM data).

```
SET
  BLANKS=SYSMIS
 UNDEFINED=WARN.
DATA LIST
 FILE='C:\THESIS\BASEDATA\PRO8993.DAT' FIXED RECORDS=1 TABLE /1 ssn 1-9 sex
 10-10(A) edyrs 13-13(A) mafqt 23-24 tdate 25-30 ps 31-32(A) .
EXECUTE.

RECODE
 ssn mafqt tdate  (SYSMIS=0)  .
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

b. Manually remove the 64 records that were missing the tdate field (all are at beginning of file)..

c. Remove any records missing SSN.

```
RECODE
 ssn  (SYSMIS=0)  .
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.

FILTER OFF.
USE ALL.
SELECT IF((ssn = 0)).
EXECUTE .


SAVE OUTFILE='C:\THESIS\BASEDATA\BADSSN.sav'
  /COMPRESSED.

GET
 FILE='C:\THESIS\BASEDATA\TDATA1.SAV'.
EXECUTE .

FILTER OFF.
```

```
USE ALL.
SELECT IF((ssn ~= 0)).
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

(One bad SSN found)

d. Recode EDYRS category into Grad, Sen or NonGrad values and then calculate mnbox using edyrs, sex and afqt.

```
RECODE
  edyrs ('1'='non') ('2'='non') ('3'='non') ('7'='non') ('8'='grd')
  ('9'='non') ('A'='non') ('B'='non') ('C'='non') ('D'='grd')
  ('E'='non') ('F'='non') ('G'='grd') ('H'='non') ('J'='non')
  ('K'='grd') ('L'='grd') ('M'='non') ('N'='grd') ('R'='grd')
  ('S'='sen') ('U'='grd') ('W'='grd') .
EXECUTE .

IF (((edyrs = 'grd') & (sex = 'M') & (mafqt  >= 50))) mnbox = 'gma' .
EXECUTE .

IF (((edyrs = 'grd') & (sex = 'F') & (mafqt  >= 50))) mnbox = 'gfa' .
EXECUTE .

IF (((edyrs = 'grd') & (sex = 'M') & ((mafqt <= 49) & (mafqt >= 31))) mnbox = 'gmb' .
EXECUTE .

IF (((edyrs = 'grd') & (sex = 'F') & ((mafqt <= 49) & (mafqt >= 31))) mnbox = 'gfb' .
EXECUTE .

IF (((edyrs = 'grd') & (sex = 'M') & ((mafqt >=21) & (mafqt <= 30))) mnbox = 'gm4' .
EXECUTE .

IF (((edyrs = 'grd') & (sex = 'F') & ((mafqt >=21) & (mafqt <= 30))) mnbox = 'gf4' .
EXECUTE .

IF  (((edyrs = 'sen') & (sex = 'M') & (mafqt  >= 50))) mnbox = 'sma' .
EXECUTE .

IF  (((edyrs = 'sen') & (sex = 'F') & (mafqt  >= 50))) mnbox = 'sfa' .
EXECUTE .

IF  (((edyrs = 'sen') & (sex = 'M') & ((mafqt <= 49) & (mafqt >= 31))) mnbox = 'smb' .
EXECUTE .

IF  (((edyrs = 'sen') & (sex = 'F') & ((mafqt <= 49) & (mafqt >= 31))) mnbox = 'sfb' .
EXECUTE .
```

```
IF  (((edyrs = 'sen') & (AFQT <= 30))) mnbox = 'oth' .
EXECUTE .

IF  ((mafqt <= 20))) mnbox = 'oth' .
EXECUTE .

IF  ((edlvl = 'non')) mnbox = 'non' .
EXECUTE .

IF ((ps = 'YY') | (ps = 'NY')) mnbox = 'ps' .
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

e. Split testdate into day, month, and year categories, and then recode month and year categories into their corresponding fiscal month and fiscal year (tfmo and tfy) categories.

```
COMPUTE tday = mod(tdate,100) .
EXECUTE .

COMPUTE tfmo = trunc(mod(tdate,1000)/100) .
EXECUTE .

COMPUTE tfy = trunc(tdate/10000) .
EXECUTE .

DO IF (tfmo >= 10) .
RECODE
 tfy  (88=89)  (89=90)  (90=91)  (91=92)  (92=93)  (93=94)  (94=95)  .
END IF .
EXECUTE .

RECODE
 tfmo  (10=1)  (11=2)  (12=3)  (1=4)  (2=5)  (3=6)  (4=7)  (5=8)  (6=9)
 (7=10)  (8=11)  (9=12)  .
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

f. Remove all records that fall outside of the testdate window.

```
FILTER OFF.
USE ALL.
SELECT IF(testdate < 880901 | testdate >= 931001).
EXECUTE .
```

```
SAVE OUTFILE='C:\THESIS\BASEDATA\BADTDATE.SAV'
  /COMPRESSED.

GET
  FILE='C:\THESIS\BASEDATA\TDATA1.SAV'.
EXECUTE .

FILTER OFF.
USE ALL.
SELECT IF(testdate >= 880901 & testdate < 931001).
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

g. Calculate bimo category, using fiscal month and fiscal year categories.

```
IF (tday <= 15) tbimo = (tfmo * 2) -  1 .
EXECUTE .

IF (tday > 15) tbimo = (tfmo * 2) .
EXECUTE .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

h. Sort by ssn in preparation for match files command.

```
SORT CASES BY
  ssn (A) .

SAVE OUTFILE='C:\THESIS\BASEDATA\TDATA1.SAV'
  /COMPRESSED.
```

i. Make table of total tested for total bimonthly, then for each mnbox category bimonthly by tfy. Save to NUMTEST.txt and FSTTEST.txt respectively.

```
CROSSTABS
  /TABLES=mnbox  BY tbimo
  /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
  /CELLS= COUNT .

CROSSTABS
  /TABLES=tbimo  BY tfy  BY mnbox
  /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
  /CELLS= COUNT .
```

j. Make tables of total tested each day of the month by ftmo, over all fiscal years. Save to DAILYTST.txt.

```
CROSSTABS
 /TABLES=tday  BY tfy  BY tfmo
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .
```

k. Delete all unwanted categories (sex, edyrs, ps) and resave to final data base form into TDATA1.sav.

2. COMMANDS USED TO MANIPULATE THE CONTRACT DATA BASE

a. Read in contract data from CON8993.dat and save as CDATA.sav.

```
SET
  BLANKS=SYSMIS BLANKS=SYSMIS
 UNDEFINED=WARN.
DATA LIST
 FILE='C:\THESIS\CONTDATA\CON8993.DAT' FIXED RECORDS=1 TABLE /1 ssn 1-9
 nbox 10-11 afqt 12-13 cdate 14-19 .
EXECUTE.

SAVE OUTFILE='C:\THESIS\CONTDATA\CDATA.SAV'
  /COMPRESSED.
```

b. Split contdate into day, month, and year categories, and then recode month and year categories into fiscal month and fiscal year (cfmo and cfy) categories.

```
COMPUTE cday = mod(cdate,100) .
EXECUTE .

COMPUTE cfmo = trunc(mod(cdate,10000)/100) .
EXECUTE .

COMPUTE cfy = trunc(cdate/10000) .
EXECUTE .

DO IF (cfmo >= 10) .
RECODE
 cfy  (88=89)  (89=90)  (90=91)  (91=92)  (92=93)  (93=94)  (94=95)  .
END IF .
EXECUTE .

RECODE
 cfmo  (10=1)  (11=2)  (12=3)  (1=4)  (2=5)  (3=6)  (4=7)  (5=8)  (6=9)
 (7=10)  (8=11)  (9=12)  .
EXECUTE .

SAVE OUTFILE='C:\THESIS\CONTDATA\CDATA.SAV'
  /COMPRESSED.
```

c. Recode the number values of nbox into corresponding 3-letter designations. Then combine as follows: a) mps, fps => 'ps'
b) nma, nmb, nfa, nfb => 'non'
c) hma, hmb, hfa, hfb, sfs, sm4, sf4 => 'oth'

```
RECODE
 nbox (' 1'='sma') (' 2'='gma') (' 3'='smb') (' 4'='gmb') (' 5'='sm4')
 (' 6'='gm4') (' 7'='nma') (' 8'='nmb') (' 9'='sfa') ('10'='gfa')
 ('11'='sfb') ('12'='gfb') ('13'='sf4') ('14'='gf4') ('15'='nfa')
 ('16'='nfb') ('17'='mps') ('18'='fps') ('19'='hma') ('20'='hmb')
 ('21'='hfa') ('22'='hfb') ('23'='sfs') .
EXECUTE .

FREQUENCIES
 VARIABLES=nbox .

RECODE
 nbox ('nma'='non') ('nfa'='non') ('nmb'='non') ('nfb'='non') ('mps'='"+
' ps') ('fps'=' ps') ('hma'='oth') ('hmb'='oth') ('hfa'='oth')
 ('hfb'='oth') ('sfs'='oth') ('sm4'='oth') ('sf4'='oth') ('  '='oth')
 .
EXECUTE .

FREQUENCIES
 VARIABLES=nbox .

SAVE OUTFILE='C:\THESIS\CONTDATA\CDATA.SAV'
 /COMPRESSED.
```

d. Calculate bimo category, using fiscal month and fiscal day categories.

```
IF (cday <= 15) cbimo = (cfmo * 2) - 1 .
EXECUTE .


IF (cday > 15) cbimo = (cfmo * 2) .
EXECUTE .

SAVE OUTFILE='C:\THESIS\CONTDATA\CDATA.SAV'
 /COMPRESSED.
```

e. Sort by SSN in preparation for match files command.

```
SORT CASES BY
 ssn (A) .

SAVE OUTFILE='C:\THESIS\CONTDATA\CDATA.SAV'
 /COMPRESSED.
```

f. Make table of total contracted bimonthly, then for each mnbox category bimonthly by tfy. Save to NUMCONT.txt and FSTCONT.txt respectively.

```
CROSSTABS
 /TABLES=nbox  BY cbimo
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .

CROSSTABS
 /TABLES=cbimo  BY cfy  BY nbox
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .
```

g. Make tables of total tested each day of the month by ftmo,, over all fiscal years. Save to DAILYCNT.txt.

```
CROSSTABS
 /TABLES=cday  BY cfy  BY cfmo
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .
```

3. COMMANDS USED TO MANIPULATE THE COMPOSITE DATA BASE

a. First count if there are any repeated SSN's within TDATA.sav or CDATA.sav. Output counts of doubles into DBLSSN.lst.

```
GET
  FILE='c:\thesis\basedata\TDATA.sav'.
EXECUTE .

COUNT
  tdblssn = ssn  (0 thru Highest)  .
EXECUTE .

FREQUENCIES
  VARIABLES=tdblssn  .

GET
  FILE='c:\thesis\contdata\CDATA.sav'.
EXECUTE .

COUNT
  cdblssn = ssn  (0 thru Highest)  .
EXECUTE .

FREQUENCIES
  VARIABLES=cdblssn  .
```

b. Match TDATA1.sav with CDATA.sav by SSN. Start with TDATA1.sav as base file.

```
GET
  FILE='c:\thesis\basedata\TDATA.sav'.
EXECUTE .

SAVE OUTFILE='c:\thesis\match\MATCH.SAV'
   /COMPRESSED.

MATCH FILES /FILE=*
 /RENAME (mafqt = d0)
 /FILE='C:\THESIS\CONTDATA\CDATA.SAV'
 /RENAME (afqt = d1)
 /IN=source01
 /BY ssn
 /DROP= d0 d1.
VARIABLE LABELS source01
 'Case source is C:\THESIS\CONTDATA\CDATA.SAV'.
EXECUTE.
```

```
SAVE OUTFILE='c:\thesis\match\MATCH.SAV'
  /COMPRESSED.
```

c. Remove all unmatched files from MATCH.sav. Save unmatched CDATA records to UNMCONT.sav and save unmatched TDATA records to UNMTEST.sav. Frequency listings are saved in MATCH.lst.

```
RECODE
 tdate (MISSING=0) .
EXECUTE .

RECODE
 cdate (MISSING=0) .
EXECUTE .

SAVE OUTFILE='c:\thesis\match\MATCH.SAV'
  /COMPRESSED.

FREQUENCIES
 VARIABLES=nbox mnbox .

FILTER OFF.
USE ALL.
SELECT IF(tdate = 0).
EXECUTE .

SAVE OUTFILE='c:\thesis\match\UNMCONT.SAV'
  /COMPRESSED.

CROSSTABS
 /TABLES=cbimo BY nbox
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .

GET
 FILE='c:\thesis\match\MATCH.sav'.
EXECUTE .

FILTER OFF.
USE ALL.
SELECT IF(tdate > 0).
EXECUTE .

FREQUENCIES
 VARIABLES=nbox mnbox .

SAVE OUTFILE='c:\thesis\match\MATCH.SAV'
  /COMPRESSED.

FILTER OFF.
```

```
USE ALL.
SELECT IF(cdate = 0).
EXECUTE .

SAVE OUTFILE='c:\thesis\match\UNMTEST.SAV'
  /COMPRESSED.

CROSSTABS
 /TABLES=tbimo  BY mnbox
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .

GET
 FILE='c:\thesis\match\MATCH.sav'.
EXECUTE .

FILTER OFF.
USE ALL.
SELECT IF(cdate > 0).
EXECUTE .

SAVE OUTFILE='c:\thesis\match\MATCH.SAV'
  /COMPRESSED.

FREQUENCIES
 VARIABLES=nbox mnbox  .
```

d. Calculate the number of days between tdate and cdate, using the current MATCH.sav file. Frequency listing is saved to TTC.txt.

```
COMPUTE contnday = YRMODA(trunc(cdate/10000),trunc(mod(cdate,10000)
/100),cday) .
EXECUTE .

COMPUTE testnday = YRMODA(trunc(tdate/10000),trunc(mod(tdate,10000)
/100),tday) .
EXECUTE .

COMPUTE waitdays = contnday - testnday .
EXECUTE .

SAVE OUTFILE='c:\thesis\match\MATCH.SAV'
  /COMPRESSED.

FREQUENCIES
 VARIABLES=waitdays
 /FORMAT=ONEPAGE
 /NTILES= 4
 /NTILES= 10
```

```
 /STATISTICS=STDDEV VARIANCE RANGE MINIMUM MAXIMUM SEMEAN MEAN MEDIAN
MODE
 SKEWNESS SESKW KURTOSIS SEKUR .
```

e. Remove any records with waitdays values less than 1.

```
FILTER OFF.
USE ALL.
SELECT IF(waitdays < 1).
EXECUTE .

SAVE OUTFILE='C:\thesis\match\negwait.sav'
  /COMPRESSED.

FREQUENCIES
 VARIABLES=mnbox nbox .

FREQUENCIES
 VARIABLES=waitdays
 /FORMAT=ONEPAGE
 /STATISTICS=STDDEV VARIANCE RANGE MINIMUM MAXIMUM SEMEAN MEAN MEDIAN
MODE
 SUM .

GET
 FILE='C:\thesis\match\MATCH.sav'.
EXECUTE .

FILTER OFF.
USE ALL.
SELECT IF(waitdays >= 1).

SAVE OUTFILE='C:\thesis\match\MATCH.sav'
  /COMPRESSED.

FREQUENCIES
 VARIABLES=mnbox nbox .

FREQUENCIES
 VARIABLES=waitdays
 /FORMAT=ONEPAGE
 /STATISTICS=STDDEV VARIANCE RANGE MINIMUM MAXIMUM SEMEAN MEAN MEDIAN
MODE
 SUM .
```

f. Create CatTTC variable that consolidates waitdays into Time To Contract categories.

```
STRING cattc (A12).
RECODE
 waitdays
  (0 thru 1='day1') (2='day2') (3='day3') (4='day4') (5='day5') (6='day6')
 (7='day7') (8 thru 9='day8to9') (10 thru 12='day10to12') (13 thru
 14='day13to14') (15 thru 18='day15to18') (19 thru 22='day19to22') (23
 thru 30='day23to30') (31 thru 45='day31to45') (46 thru 60='day46to60')
 (61 thru 120='day61to120') (121 thru 180='day121to180') (181 thru
 365='day181to365') (366 thru 730='day366to730') (731 thru
 Highest='day_gt731') INTO cattc .
EXECUTE .

SAVE OUTFILE='C:\thesis\match\MATCH.sav'
  /COMPRESSED.

CROSSTABS
 /TABLES=tbimo BY cattc BY mnbox
 /FORMAT= AVALUE NOINDEX NOBOX LABELS TABLES
 /CELLS= COUNT .
```

APPENDIX D. DETAILED DESCRIPTION OF MODEL PARAMETERS

1. FIRSTTEST

Definition: The total number of applicants who took the AFQT in each bi-monthly period over the last five fiscal years, for each of the testing mission box categories.

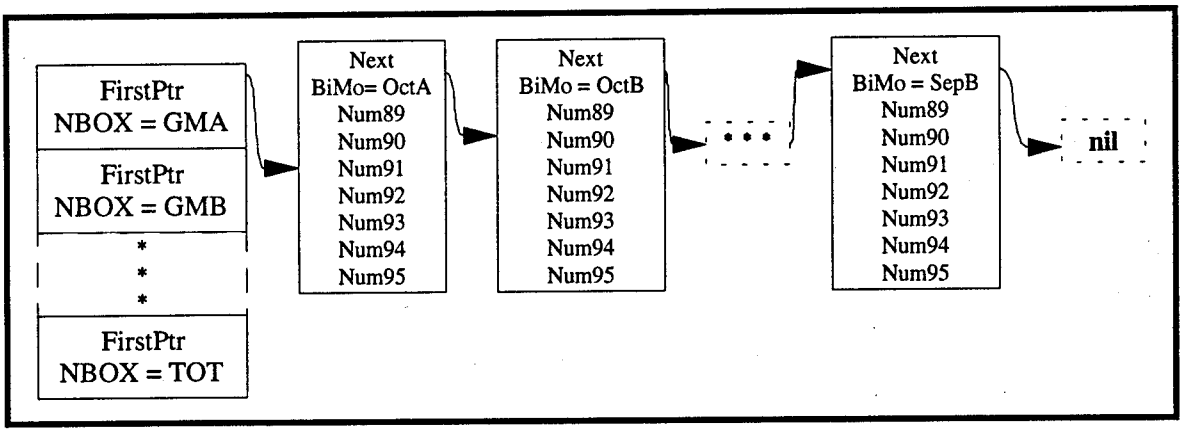
Pascal source code: Unit PEWSCURR.pas.

EXCEL location: CURRDATA.xls.

Derivation: From the Testing data base, break out the number of applicants that took the AFQT during each bi-monthly period and by their testing mission box category.

Data Structure within Model: FIRSTTEST is stored as an array of the eight critical mission boxes, with each cell pointing to a linked list of 24 bi-monthly records, each of which contains a field for the bi-month and the number tested for FY 1989 through 1995. The FY 1994 and 1995 fields are included so that current data can be continuously updated to the data base.

Schematic of FIRSTTEST data structure



2. FIRSTCONT

Definition: The total number of prospects who signed a contract in each bi-monthly period over the last five fiscal years, for each of the contract mission box. categories.

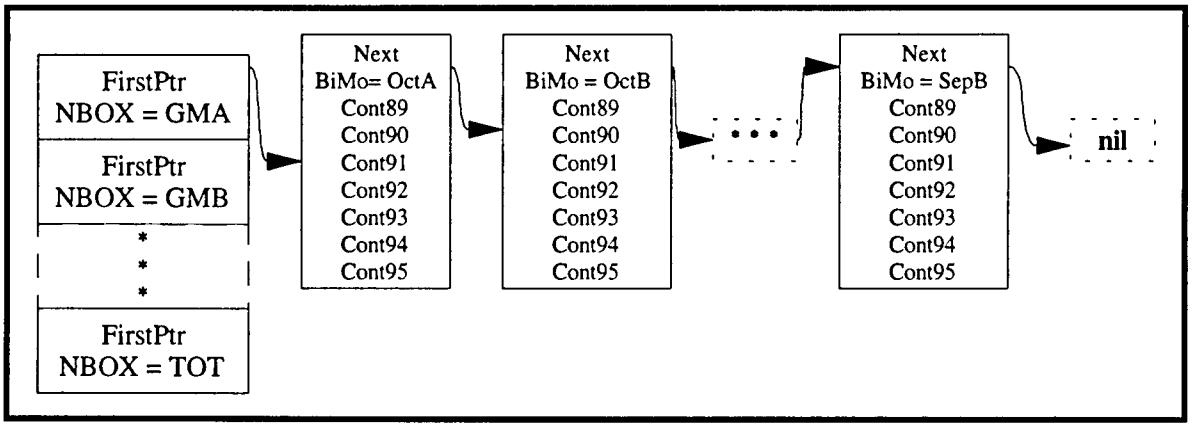
Pascal source code: Unit PEWSCURR.pas.

EXCEL location: CURRDATA.xls.

Derivation: From the Contract data base, break out the number of prospects that signed a contract during each bi-monthly period and by their contract mission box category.

Data Structure within Model: FIRSTCONT is stored as an array of the eight critical mission boxes, with each cell pointing to a linked list of 24 bi-monthly records, each of which contains a field for the bi-month and the number contracted for FY 1989 through 1995. The FY 1994 and 1995 fields are included so that current data can be continuously updated to the data base.

Schematic of FIRSTCONT data structure



3. NUMTESTED

Definition: Average number of applicants expected to take the AFQT in a given bi-monthly period.

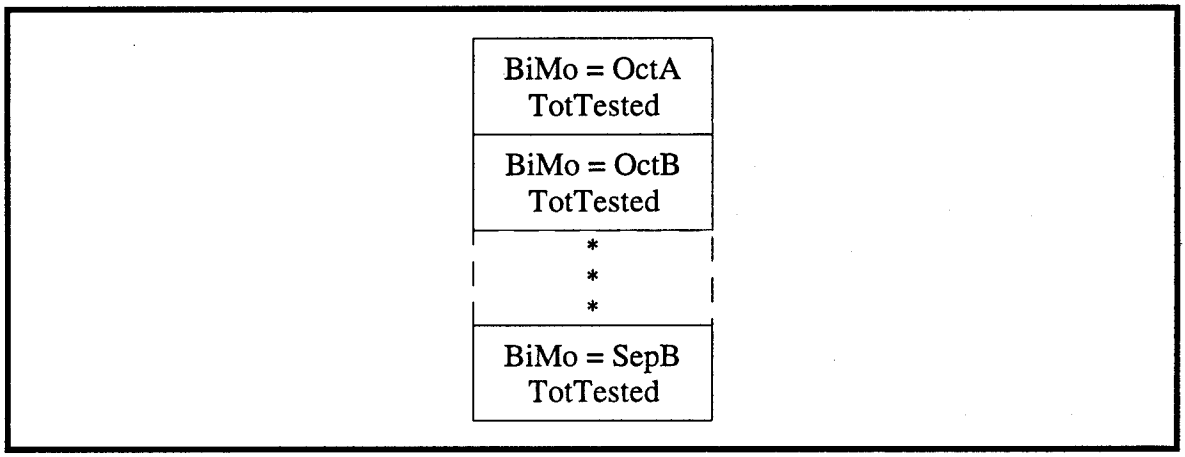
Pascal source code: Unit PEWSPARA.pas.

EXCEL location: PARADATA.xls.

Derivation: From the Testing data base, calculate the arithmetic average of the total number of applicants that took the AFQT for each bi-monthly category over the last three years.

Data Structure within Model: NUMTESTED is stored as an array of 24 records, each containing a bi-month field and eight fields pertaining to the percentage of prospects that are expected to sign a contract by their testing mission box classification.

Schematic of NUMTESTED data structure



4. PERCENT CONTRACTED (PCTCONT)

Definition: The probability that a prospect will sign a contract, given that he or she took the AFQT during a specific bi-monthly period and was classified into a specific testing mission box category at that time.

Pascal source code: Unit PEWSPARA.pas.

EXCEL location: PARADATA.xls.

Derivation: For each testing bi-monthly period and each testing mission box:

1) Calculate the total number of testing records from the Testing data base
   (~ total number of applicants that took the AFQT ~ NumTested)

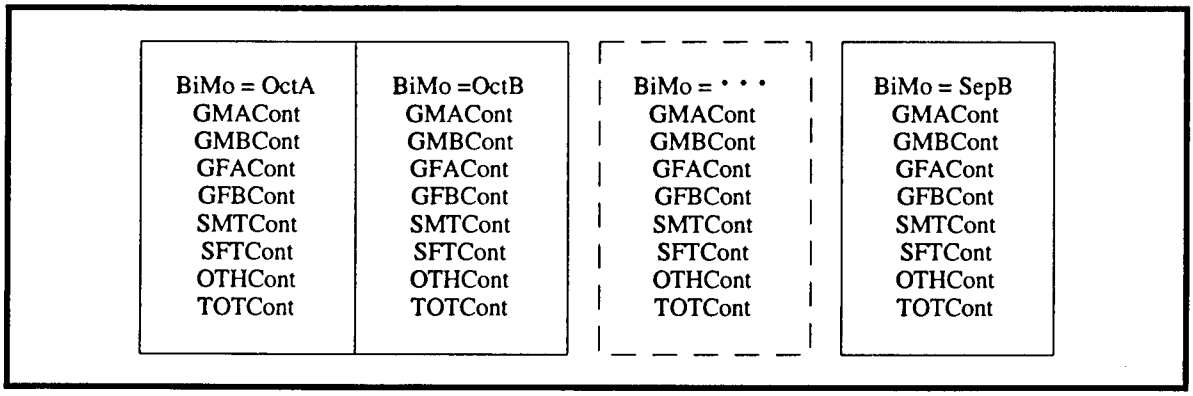
2) Calculate the total number of unmatched testing records from the Composite data base
   (~ number of applicants that did not sign a contract ~ NumNotContracted)

3) Calculate the total number of applicants that signed a contract
   (NumContracted = NumTested - NumNotContracted)

4) Calculate Percent Contracted
   (PCTCONT = NumContracted / NumTested)

Data Structure within Model: PCTCONT is stored as an array of 24 records, each containing a bi-month field and eight fields pertaining to the percentage of additional contracts that are expected to be realized based on the contract mission box classification.

Schematic of PCTCONT data structure



5. UNMATCHED CONTRACTED (PCTUNM)

Definition: The percentage of additional contracts that are expected to be obtained from the batch AFQT tests during a specific contracting bi-monthly period for each contracting mission box category.

Pascal source code: Unit PEWSPARA.pas.

EXCEL location: PARADATA.xls.

Derivation: For each contracting bi-monthly period and each contracted mission box:

1) Calculate the number of unmatched contract records from the Composite data base
(~ number of batch AFQT prospects that signed a contract ~ NumBatch)

2) Calculate the total number of contract records from the Contract data base
(~ total number of prospects that signed a contract ~ TotCont)

3) Calculate the percentage of all contracts that come from batch AFQT
(PctBatch = BatchCont / TotCont)

4) Rewrite equation to solve for the number of batch contracts signed
(BatchCont = TotCont*PctBatch)

5) Define: regular contracts = number of contracts from non-batch AFQT (~ RegCont)
(these are the only type of contracts that can be predicted by the model, since the batch AFQT records are unavailable for analysis)

6) Write equation for number of total records as a function of batch and regular contracts
(TotCont = BatchCont + RegCont)

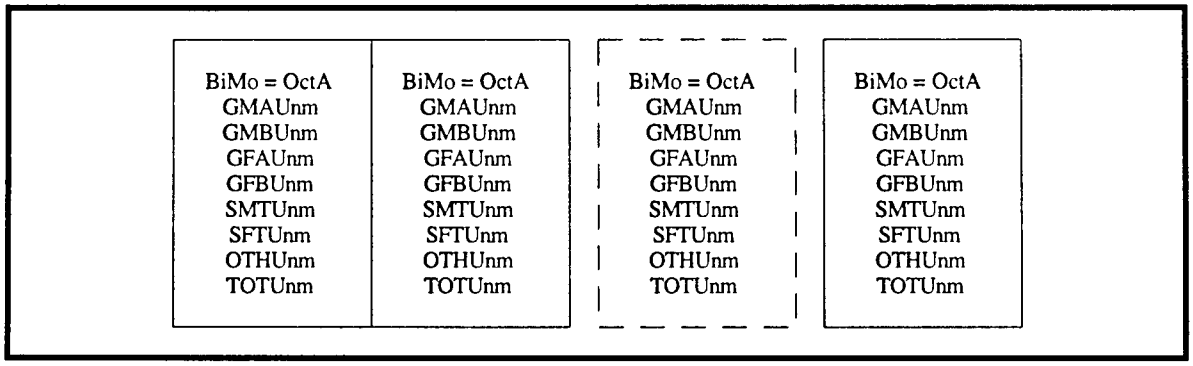
7) Substitute (4) into equation to find total contracts as a function of batch contracts and regular contracts
(TotCont = TotCont*PctBatch + RegCont)

8) Finally, solve for total contracts in terms of regular contracts and the percentage of contracts that come from batch AFQT
(TotCont = [RegCont / (1-PctBatch)]

9) PCTUNM is the value [1 / (1 - PctBatch)], and is used multiplicitavely to adjust the number of regular contracts predicted to be signed in the model into the total number of contracts expected to be signed.

Data Structure within Model: PCTUNM is stored as an array of 24 records, each containing a bi-month field and eight fields pertaining to the percentage of additional contracts that are expected to be realized based on the contract mission box classification.

Schematic of PCTUNM data structure



6. TIME TO CONTRACT (TTC)

Definition: The probability that a prospect will sign a contract within a specific time period, given that the prospect took the AFQT during a specific bi-month period and that the prospect will sign a contract in a specific mission box category.

Pascal source code: Unit PEWSPARA.pas.

EXCEL location: PARADATA.xls.

Derivation: For each testing bi-monthly period and each contracting mission box:

1) Calculate the julian test date for each record in the Composite data base
(~ date on which each prospect that signed a contract took the AFQT)

2) Calculate the julian contract date for each record in the Composite data base
(~ date on which each prospect that signed a contract signed the contract)

3) Calculate the number of days it took each prospect to sign a contract, given a contract was signed
(WaitDays = ContractDate - TestDate)

4) Tabulate results over all records, and choose 'days to contract' category bins so that four to six percent of the total records fall into each bin

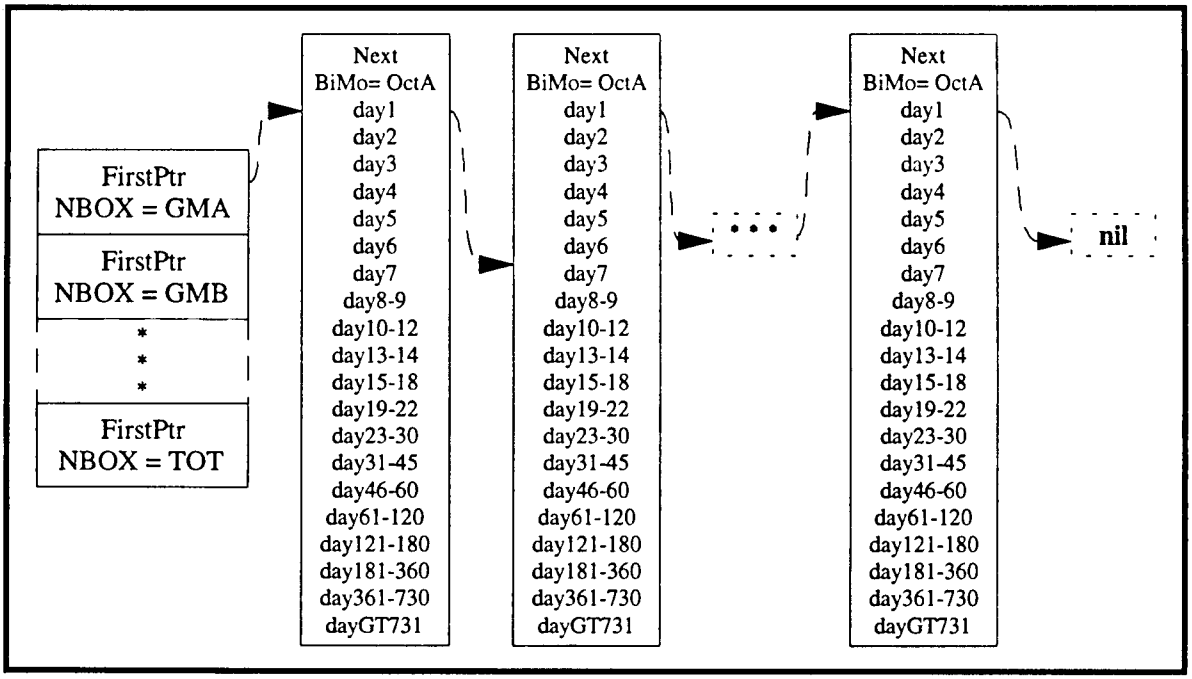
5) Tabulate the actual number of contracts signed that fall into each WaitDay window for each testing bi-monthly period and each testing mission box (~ NumTestContWait)

6) Tabulate the total number of prospects that took the AFQT in each bi-month period and signed a contract (~ TotTestCont)

7) Calculate the percentage of contracts that fall into each window
(%TTC = NumTestContWait / TotTestCont)

Data Structure within Model: TTC is stored as an array of eight testing mission box category records, with each cell pointing to a linked list of 24 records, each of which contains a field for the bi-month category and each of the 20 'days to contract' categories.

Schematic of TTC data structure



7. DAILY AVERAGES (DAILYTEST, DAILYCONT)

Definition: The number of applicants expected to take the AFQT on a single day in a specific testing bi-monthly period, and the number of prospects expected to sign a contract on a single day in a specific contracting bi-monthly period.

Pascal source code: Unit PEWSCURR.pas.

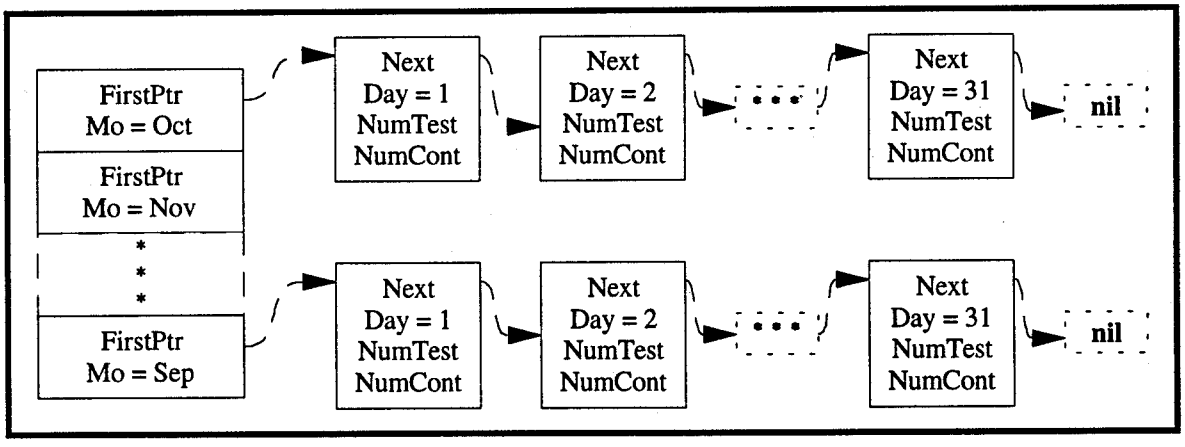
EXCEL location: AVGDATA.xls.

Derivation: 1) DAILYTEST: Take the simple arithmetic average of the total number of applicants that took the AFQT over the last three years for each testing bi-month period from the Testing data base.

2) DAILYCONT: Take the arithmetic average of the total number of prospects that signed a contract over the last three years for each contracting bi-month period from the Contract data base.

Data Structure within Model: DAILY AVERAGES is stored as an array of 12 monthly records, with each cell pointing to a linked list of 28, 30 or 31 records, depending on the day of the month. Each of these records contains fields to store the day of the month, the number expected to take an AFQT test on that day, and the number expected to sign a contract on that day.

Schematic of DAILY AVERAGES data structure



8. PERCENT MISSION BOX (PCTNBOX)

Definition: The probability that a prospect will sign a contract for a specific contract mission box category, given that the prospect will sign a contract.

Pascal source code: Unit PEWSCURR.pas.

EXCEL location: AVGDATA.xls.

Derivation: For each mission box category:

1) Calculate the number of contracts signed for each mission box category from the Contract data base (~ NboxCont)

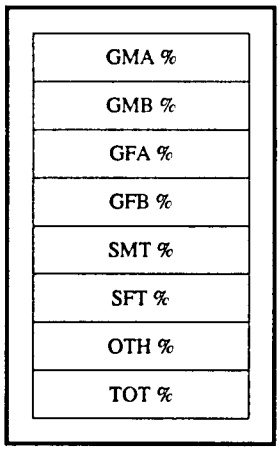
2) Calculate the total number of contracts signed over all mission box categories from the Contract data base (~ TotCont)

3) Calculate the percentage of contracts in each mission box category

(PCTNBOX = NBoxCont / TotCont)

Data Structure within Model: PCTNBOX is stored as an array of the eight critical mission box categories.

Schematic of PCTNBOX data structure



APPENDIX E. EXPLANATION OF TESTING DATA BASE EDLVL CODE

Definition/Explanation: Education status or level attained by an individual. The first two characters correspond to an applicant's number of years of education. The last character corresponds to the Grad Code, which represents the highest level of education that an individual has attained. This Grad Code corresponds directly to the USAREC Grad Certification (Grad Cert).

MEPCOM Grad Code	USAREC Grad Cert	Description
9	H	Status of an individual; who is currently in High School and is not yet a High School Senior.
S	S	Status of an individual who is currently attending High School and is a High School Senior.
1	N	Status of an individual who is not currently attending High School and who is neither a High School Graduate or an alternate High School credential holder.
M	N	Status of an individual who is currently pursuing completion of one of the credentials listed below.
E	N	A diploma or certificate of General Education Development (GED) or other test-based H. School equivalency diploma. This includes state-wide testing programs such as the California High School Proficiency Examination (CHSPE), whereby examines may earn a certificate or competency or proficiency. A state or locally issued secondary school diploma obtained solely on the basis of such equivalency testing is not to be considered a High School diploma. This is considered an alternate High School credential.

MEPCOM Grad Code	USAREC Grad Cert	Description
C	N	A certificate or diploma awarded for attending a non-correspondence vocational technical, or proprietary school for at least 6 months. An individual so coded must also have completed 11 years of regular day school. This is considered an alternate High School credential.
7	N	A secondary school diploma or certificate awarded upon completion of correspondence school coursework, regardless whether the diploma was issued by a correspondence school, state, or a secondary or post-secondary educational institution. This is considered an alternate High School credential.
H	N	A secondary school diploma or certificate ty[pically awarded by a state, based upon certification by a parent or guardian that an individual completed his/her secondary education at home. This is considered an alternate High School education.
B	N	A secondary school diploma awarded on the basis of attending and completing an adult education or "external" diploma program, regardless of whether the diploma was issued by state or by a secondary or postsecondary educational institution. This is considered an alternate High School credential.
J	N	An attendance-based High School certificate or diploma. These are sometimes called certificates of competency or completion but are based on course completion rather than test such as the GED or CHSPE. A state or locally issued secondary school diploma obtained solely on the basis of attendance. Attendance credential is not a High School diploma. This is an alternate High School credential.

MEPCOM Grad Code	USAREC Grad Cert	Description
8	G	The status of an individual who is a non High School graduate or alternate High School credential holder, attended a college or university, and completed at least 15 semester 20 quarter hours of college-level credit. Credit earned through testing, for pursuit of adult education, or for High School equivalency prpeparation is not applicable. For military enlistment purposes, an individual with this status is considered a High School graduate.
L	G	A diploma issued to an individual who has attended and completed a 12 -year or grade day program of classroom instruction; the diploma must be issued from the school where the individual completed all the program requirements.
D	G	A certificate conferred upon completion of a 2-year program at a junior college, university, or technical institutes.
G	G	A certificate conferred upon completion of a 3-year hospital school of nursing program.
K	G	A certificate conferred upon completion of a 4-year college program, other than a first professional degree.
W	G	A certificate conferred upon completion of the academic requirement for the first degrees awarded in selected professions: architecture, certified public accountant, chiropody (D.S.C. or POD.D.), Dentitry (D.D.S. or D.M.D.), Medicine (M.D.) Optometry (O.D.), Osteopathy (D.O.), Pharmacy, Veterinary Medicine, Law (L.L.B.) or J.D.) and Theology (B.D., Rabbi, or other first professional degree).

MEPCOM Grad Code	USAREC Grad Cert	Description
N	G	A certificate conferred upon completion of additional academic requirements beyond the baccalaureate or first professional degree but below the doctorate level.
R	G	A certificate conferred upon completion of extra academic requirements beyond the master's degree but below the doctorate level.
U	G	A certificate conferred upon completion of the highest academic achievement within an academic field, excluding honorary degrees and first professional degrees.

APPENDIX F. RESULTS FROM VALIDATION PROJECTION RUN

The results of the monthly and semi-monthly validation projection runs are tabulated on the following pages of this appendix. Validation runs are projected into Fiscal Year 1993, which is the most current year of the historical data.

1. TABULATION OF RESULTS FROM MONTHLY PROJECTION RUNS

SUMMARY STATISTICS FROM PROJECTION RUNS BY MONTH (FY 93) (SumProj - SumAct)

NBox	Month	CurrDate	ProjDate	#Days	Curr	Pro	Pred	SumProj	Actual	SumAct	Actual
GMA	Oct	1/1/93	31/ 1/93	30	80	867	1465	2412	2411	2411	0.00
GMA	Nov	1/2/93	30/ 2/93	29	2477	730	1216	4423	2165	4576	-0.07
GMA	Dec	1/3/93	31/ 3/93	30	4645	716	1162	6522	2309	6885	-0.16
GMA	Jan	1/4/93	31/ 4/93	30	6956	686	1627	9269	2453	9338	-0.03
GMA	Feb	1/5/93	28/ 5/93	27	9414	756	1143	11313	2199	11537	-0.10
GMA	Mar	1/6/93	31/ 6/93	30	11605	723	1198	13526	2542	14079	-0.22
GMA	Apr	1/7/93	30/ 7/93	29	14141	653	1023	15817	2138	16217	-0.19
GMA	May	1/8/93	31/ 8/93	30	16278	644	1003	17924	2251	18468	-0.24
GMA	Jun	1/9/93	30/ 9/93	29	18523	598	1332	20452	2648	21116	-0.25
GMA	Jul	1/10/93	31/10/93	30	21185	720	1425	23330	2562	23678	-0.14
GMA	Aug	1/11/93	31/11/93	30	23752	720	1474	25945	2618	26296	-0.13
GMA	Sep	1/12/93	30/12/93	29	26369	736	1429	28535	2305	28601	-0.03
GMB	Oct	1/1/93	31/ 1/93	30	37	512	669	1219	1215	1215	0.00
GMB	Nov	1/2/93	30/ 2/93	29	1246	391	556	2192	995	2210	-0.02
GMB	Dec	1/3/93	31/ 3/93	30	2242	408	516	3166	1042	3252	-0.08
GMB	Jan	1/4/93	31/ 4/93	30	3285	408	721	4414	1153	4405	0.01
GMB	Feb	1/5/93	28/ 5/93	27	4440	453	529	5422	1103	5508	-0.08
GMB	Mar	1/6/93	31/ 6/93	30	5540	454	547	6540	1086	6594	-0.05
GMB	Apr	1/7/93	30/ 7/93	29	6623	404	486	7513	900	7494	0.02
GMB	May	1/8/93	31/ 8/93	30	7522	386	435	8344	1003	8497	-0.15
GMB	Jun	1/9/93	30/ 9/93	29	8522	363	571	9456	1300	9797	-0.26
GMB	Jul	1/10/93	31/10/93	30	9829	448	618	10896	1276	11073	-0.14
GMB	Aug	1/11/93	31/11/93	30	11107	471	635	12214	1406	12479	-0.19
GMB	Sep	1/12/93	30/12/93	29	12513	458	591	13563	1161	13640	-0.07
GFA	Oct	1/1/93	31/ 1/93	30	21	276	371	668	686	686	-0.03
GFA	Nov	1/2/93	30/ 2/93	29	704	230	313	1246	603	1289	-0.07
GFA	Dec	1/3/93	31/ 3/93	30	1307	229	294	1830	590	1879	-0.08
GFA	Jan	1/4/93	31/ 4/93	30	1898	223	410	2531	565	2444	0.15
GFA	Feb	1/5/93	28/ 5/93	27	2464	233	280	2978	592	3036	-0.10
GFA	Mar	1/6/93	31/ 6/93	30	3054	239	301	3594	707	3743	-0.21
GFA	Apr	1/7/93	30/ 7/93	29	3760	221	258	4239	622	4365	-0.20
GFA	May	1/8/93	31/ 8/93	30	4381	214	248	4844	625	4990	-0.23
GFA	Jun	1/9/93	30/ 9/93	29	5005	193	324	5522	791	5781	-0.33
GFA	Jul	1/10/93	31/10/93	30	5799	240	346	6386	768	6549	-0.21
GFA	Aug	1/11/93	31/11/93	30	6569	241	359	7169	772	7321	-0.20
GFA	Sep	1/12/93	30/12/93	29	7341	252	342	7935	740	8061	-0.17
GFB	Oct	1/1/93	31/ 1/93	30	10	207	119	336	374	374	-0.10
GFB	Nov	1/2/93	30/ 2/93	29	382	184	108	674	285	659	0.05
GFB	Dec	1/3/93	31/ 3/93	30	667	189	109	966	265	924	0.16
GFB	Jan	1/4/93	31/ 4/93	30	932	178	160	1270	333	1257	0.04
GFB	Feb	1/5/93	28/ 5/93	27	1266	151	94	1511	214	1471	0.19
GFB	Mar	1/6/93	31/ 6/93	30	1479	164	104	1747	203	1674	0.36
GFB	Apr	1/7/93	30/ 7/93	29	1681	151	77	1910	140	1814	0.69
GFB	May	1/8/93	31/ 8/93	30	1821	156	82	2059	253	2067	-0.03
GFB	Jun	1/9/93	30/ 9/93	29	2074	150	107	2330	318	2385	-0.17
GFB	Jul	1/10/93	31/10/93	30	2393	174	122	2689	408	2793	-0.25
GFB	Aug	1/11/93	31/11/93	30	2802	172	122	3096	300	3093	0.01
GFB	Sep	1/12/93	30/12/93	29	3102	184	92	3378	244	3337	0.17

SUMMARY STATISTICS FROM PROJECTION RUNS BY MONTH (FY 93)

NBox	Month	CurrDate	ProjDate	#Days	Curr	Proj	Pred	SumProj	Actual	SumAct	(SumProj - SumAct) Actual
SMT	Oct	1/1/93	31/ 1/93	30	57	1076	910	2042	2335	2335	-0.13
SMT	Nov	1/2/93	30/ 2/93	29	2382	837	830	4048	1789	4124	-0.04
SMT	Dec	1/3/93	31/ 3/93	30	4173	861	936	5970	1856	5980	-0.01
SMT	Jan	1/4/93	31/ 4/93	30	6031	884	1317	8232	1535	7515	0.47
SMT	Feb	1/5/93	28/ 5/93	27	7569	960	895	9424	1608	9123	0.19
SMT	Mar	1/6/93	31/ 6/93	30	9172	961	953	11086	1992	11115	-0.01
SMT	Apr	1/7/93	30/ 7/93	29	11159	882	817	12858	1793	12908	-0.03
SMT	May	1/8/93	31/ 8/93	30	12951	839	721	14512	1487	14395	0.08
SMT	Jun	1/9/93	30/ 9/93	29	14434	878	882	16195	1710	16105	0.05
SMT	Jul	1/10/93	31/10/93	30	16154	1053	957	18165	1782	17887	0.16
SMT	Aug	1/11/93	31/11/93	30	17940	1052	888	19879	1750	19637	0.14
SMT	Sep	1/12/93	30/12/93	29	19690	1053	878	21621	1555	21192	0.28
SFT	Oct	1/1/93	31/ 1/93	30	6	151	87	244	222	222	0.10
SFT	Nov	1/2/93	30/ 2/93	29	227	121	75	423	274	496	-0.27
SFT	Dec	1/3/93	31/ 3/93	30	501	116	78	695	228	724	-0.13
SFT	Jan	1/4/93	31/ 4/93	30	729	132	116	978	219	943	0.16
SFT	Feb	1/5/93	28/ 5/93	27	949	129	74	1151	254	1197	-0.18
SFT	Mar	1/6/93	31/ 6/93	30	1202	132	87	1421	317	1514	-0.29
SFT	Apr	1/7/93	30/ 7/93	29	1519	138	80	1736	267	1781	-0.17
SFT	May	1/8/93	31/ 8/93	30	1786	124	73	1982	218	1999	-0.08
SFT	Jun	1/9/93	30/ 9/93	29	2003	120	61	2184	177	2176	0.05
SFT	Jul	1/10/93	31/10/93	30	2181	128	81	2389	218	2394	-0.02
SFT	Aug	1/11/93	31/11/93	30	2399	133	81	2613	223	2617	-0.02
SFT	Sep	1/12/93	30/12/93	29	2622	167	84	2873	258	2875	-0.01
OTH	Oct	1/1/93	31/ 1/93	30	32	542	363	936	1245	1245	-0.25
OTH	Nov	1/2/93	30/ 2/93	29	1271	481	296	2049	883	2128	-0.09
OTH	Dec	1/3/93	31/ 3/93	30	2155	475	322	2952	1436	3564	-0.43
OTH	Jan	1/4/93	31/ 4/93	30	3592	484	465	4542	3029	6593	-0.68
OTH	Feb	1/5/93	28/ 5/93	27	6623	460	282	7365	1436	8029	-0.46
OTH	Mar	1/6/93	31/ 6/93	30	8056	465	283	8804	454	8483	0.71
OTH	Apr	1/7/93	30/ 7/93	29	8508	477	296	9280	493	8976	0.62
OTH	May	1/8/93	31/ 8/93	30	9000	425	219	9644	589	9565	0.13
OTH	Jun	1/9/93	30/ 9/93	29	9587	407	247	10240	410	9975	0.65
OTH	Jul	1/10/93	31/10/93	30	10002	486	298	10786	697	10672	0.16
OTH	Aug	1/11/93	31/11/93	30	10701	521	334	11557	683	11355	0.30
OTH	Sep	33981	30/12/93	29	11384	520	268	12173	580	11935	0.41
TOT	Oct	1/1/93	31/ 1/93	30	242	3528	3796	7566	8488	8488	-0.11
TOT	Nov	1/2/93	30/ 2/93	29	8688	2913	3265	14866	6994	15482	-0.09
TOT	Dec	1/3/93	31/ 3/93	30	15690	2918	3319	21927	7726	23208	-0.17
TOT	Jan	1/4/93	31/ 4/93	30	23424	2971	4662	31056	9287	32495	-0.15
TOT	Feb	1/5/93	28/ 5/93	27	32726	3174	3184	39085	7406	39901	-0.11
TOT	Mar	1/6/93	31/ 6/93	30	40108	3121	3359	46589	7301	47202	-0.08
TOT	Apr	1/7/93	30/ 7/93	29	47390	2874	2962	53225	6353	53555	-0.05
TOT	May	1/8/93	31/ 8/93	30	53739	2681	2693	59112	6426	59981	-0.14
TOT	Jun	1/9/93	30/ 9/93	29	60147	2609	3445	66201	7354	67335	-0.15
TOT	Jul	1/10/93	31/10/93	30	67544	3156	3780	74480	7711	75046	-0.07
TOT	Aug	1/11/93	31/11/93	30	75270	3218	3797	82285	7752	82798	-0.07
TOT	Sep	1/12/93	30/12/93	29	83021	3248	3523	89793	6843	89641	0.02

2. TABULATION OF RESULTS FROM BI-WEEKLY PROJECTION RUNS

SUMMARY STATISTICS FROM PROJECTION RUNS BY MONTH (FY 93)

NBox	Month	CurrDate	ProjDate	#Days	Curr	Proj	Pred	SumProj	Actual	SumAct	(SumProj - SumAct) Actual
GMA	OctA	1/1/93	15/ 1/93	14	80	560	480	1120	1091	1091	0.03
GMA	OctB	16/ 1/93	31/ 1/93	15	1291	617	646	2553	1320	2411	0.11
GMA	NovA	1/2/93	15/ 2/93	14	2477	470	453	3400	896	3307	0.10
GMA	NovB	16/ 2/93	30/ 2/93	14	3482	238	476	4196	1269	4576	-0.30
GMA	DecA	1/3/93	15/ 3/93	14	4645	454	514	5613	1011	5587	0.03
GMA	DecB	16/ 3/93	31/ 3/93	15	5676	240	380	6295	1298	6885	-0.45
GMA	JanA	1/4/93	15/ 4/93	14	6956	407	534	7897	1167	8052	-0.13
GMA	JanB	16/ 4/93	31/ 4/93	15	8051	256	739	9045	1286	9338	-0.23
GMA	FebA	1/5/93	15/ 5/93	14	9414	503	474	10392	1089	10427	-0.03
GMA	FebB	16/ 5/93	28/ 5/93	12	10565	229	397	11190	1110	11537	-0.31
GMA	MarA	1/6/93	15/ 6/93	14	11605	444	431	12480	1021	12558	-0.08
GMA	MarB	16/ 6/93	31/ 6/93	15	12645	257	498	13400	1521	14079	-0.45
GMA	AprA	1/7/93	15/ 7/93	14	14141	407	384	14932	924	15003	-0.08
GMA	AprB	16/ 7/93	30/ 7/93	14	15089	226	422	15737	1214	16217	-0.40
GMA	MayA	1/8/93	15/ 8/93	14	16278	396	380	17054	944	17161	-0.11
GMA	MayB	16/ 8/93	31/ 8/93	15	17195	228	408	17831	1307	18468	-0.49
GMA	JunA	1/9/93	15/ 9/93	14	18523	372	484	19379	1048	19516	-0.13
GMA	JunB	16/ 9/93	30/ 9/93	14	19387	207	573	20167	1600	21116	-0.59
GMA	JulA	1/10/93	15/10/93	14	21185	447	488	22120	1022	22138	-0.02
GMA	JulB	16/10/93	31/10/93	15	22248	252	640	23140	1540	23678	-0.35
GMA	AugA	1/11/93	15/11/93	14	23752	458	515	24725	1017	24695	0.03
GMA	AugB	16/11/93	31/11/93	15	24879	241	635	25754	1601	26296	-0.34
GMA	SepA	1/12/93	15/12/93	14	26369	478	486	27334	898	27194	0.16
GMA	SepB	16/12/93	30/12/93	14	27495	236	656	28387	1407	28601	-0.15
GMB	OctA	1/1/93	15/ 1/93	14	37	328	217	581	560	560	0.04
GMB	OctB	16/ 1/93	31/ 1/93	15	598	332	287	1218	655	1215	0.00
GMB	NovA	1/2/93	15/ 2/93	14	1246	248	207	1700	405	1620	0.20
GMB	NovB	16/ 2/93	30/ 2/93	14	1711	132	214	2057	590	2210	-0.26
GMB	DecA	1/3/93	15/ 3/93	14	2242	247	221	2710	461	2671	0.09
GMB	DecB	16/ 3/93	31/ 3/93	15	2720	148	167	3034	581	3252	-0.38
GMB	JanA	1/4/93	15/ 4/93	14	3285	226	231	3743	533	3785	-0.08
GMB	JanB	16/ 4/93	31/ 4/93	15	3792	168	318	4278	620	4405	-0.20
GMB	FebA	1/5/93	15/ 5/93	14	4440	285	205	4930	519	4924	0.01
GMB	FebB	16/ 5/93	28/ 5/93	12	4974	154	181	5309	584	5508	-0.34
GMB	MarA	1/6/93	15/ 6/93	14	4958	260	183	5400	425	5933	-1.25
GMB	MarB	16/ 6/93	31/ 6/93	15	6022	170	227	6419	661	6594	-0.27
GMB	AprA	1/7/93	15/ 7/93	14	6041	242	177	6460	363	6957	-1.37
GMB	AprB	16/ 7/93	30/ 7/93	14	7062	147	189	7399	537	7494	-0.18
GMB	MayA	1/8/93	15/ 8/93	14	6940	226	156	7323	367	7861	-1.47
GMB	MayB	16/ 8/93	31/ 8/93	15	7947	148	174	8269	636	8497	-0.36
GMB	JunA	1/9/93	15/ 9/93	14	7940	214	184	8339	488	8985	-1.32
GMB	JunB	16/ 9/93	30/ 9/93	14	8923	137	241	9301	812	9797	-0.61
GMB	JulA	1/10/93	15/10/93	14	9247	269	198	9715	484	10281	-1.17
GMB	JulB	16/10/93	31/10/93	15	10322	166	272	10760	792	11073	-0.40
GMB	AugA	1/11/93	15/11/93	14	10525	285	224	11034	565	11638	-1.07
GMB	AugB	16/11/93	31/11/93	15	11630	172	266	12068	841	12479	-0.49
GMB	SepA	1/12/93	15/12/93	14	11931	287	199	12417	471	12950	-1.13
GMB	SepB	16/12/93	30/12/93	14	13035	157	258	13450	690	13640	-0.28

SUMMARY STATISTICS FROM PROJECTION RUNS BY MONTH (Cont) (FY 93)

NBox	Month	CurrDate	ProjDate	#Days	Curr	Proj	Pred	SumProj	Actual	SumAct	(SumProj - SumAct) Actual
GFA	OctA	1/1/93	15/ 1/93	14	21	177	121	319	318	318	0.00
GFA	OctB	16/ 1/93	31/ 1/93	15	345	181	159	685	368	686	0.00
GFA	NovA	1/2/93	15/ 2/93	14	704	146	107	957	250	936	0.08
GFA	NovB	16/ 2/93	30/ 2/93	14	972	77	127	1177	353	1289	-0.32
GFA	DecA	1/3/93	15/ 3/93	14	1307	142	122	1571	256	1545	0.10
GFA	DecB	16/ 3/93	31/ 3/93	15	1583	79	92	1755	334	1879	-0.37
GFA	JanA	1/4/93	15/ 4/93	14	1898	128	123	2149	247	2126	0.09
GFA	JanB	16/ 4/93	31/ 4/93	15	2191	88	183	2462	318	2444	0.06
GFA	FebA	1/5/93	15/ 5/93	14	2464	154	104	2723	294	2738	-0.05
GFA	FebB	16/ 5/93	28/ 5/93	12	2772	72	100	2944	298	3036	-0.31
GFA	MarA	1/6/93	15/ 6/93	14	3054	138	97	3289	256	3292	-0.01
GFA	MarB	16/ 6/93	31/ 6/93	15	3332	86	127	3545	451	3743	-0.44
GFA	AprA	1/7/93	15/ 7/93	14	3760	135	88	3982	265	4008	-0.10
GFA	AprB	16/ 7/93	30/ 7/93	14	4013	78	103	4194	357	4365	-0.48
GFA	MayA	1/8/93	15/ 8/93	14	4381	128	82	4591	252	4617	-0.10
GFA	MayB	16/ 8/93	31/ 8/93	15	4627	80	102	4809	373	4990	-0.49
GFA	JunA	1/9/93	15/ 9/93	14	5005	118	108	5231	324	5314	-0.26
GFA	JunB	16/ 9/93	30/ 9/93	14	5236	68	133	5436	467	5781	-0.74
GFA	JulA	1/10/93	15/10/93	14	5799	145	111	6056	314	6095	-0.13
GFA	JulB	16/10/93	31/10/93	15	6084	87	156	6327	454	6549	-0.49
GFA	AugA	1/11/93	15/11/93	14	6569	152	116	6836	302	6851	-0.05
GFA	AugB	16/11/93	31/11/93	15	6870	82	157	7109	470	7321	-0.45
GFA	SepA	1/12/93	15/12/93	14	7341	158	103	7601	294	7615	-0.05
GFA	SepB	16/12/93	30/12/93	14	7642	86	157	7884	446	8061	-0.40
GFB	OctA	1/1/93	15/ 1/93	14	10	117	32	158	158	158	0.00
GFB	OctB	16/ 1/93	31/ 1/93	15	154	119	53	325	216	374	-0.22
GFB	NovA	1/2/93	15/ 2/93	14	382	109	35	526	120	494	0.27
GFB	NovB	16/ 2/93	30/ 2/93	14	502	69	45	616	165	659	-0.26
GFB	DecA	1/3/93	15/ 3/93	14	667	106	42	815	126	785	0.24
GFB	DecB	16/ 3/93	31/ 3/93	15	790	77	33	901	139	924	-0.17
GFB	JanA	1/4/93	15/ 4/93	14	932	101	47	1081	178	1102	-0.12
GFB	JanB	16/ 4/93	31/ 4/93	15	1063	71	69	1202	155	1257	-0.35
GFB	FebA	1/5/93	15/ 5/93	14	1266	91	32	1389	119	1376	0.11
GFB	FebB	16/ 5/93	28/ 5/93	12	1403	55	33	1491	95	1471	0.21
GFB	MarA	1/6/93	15/ 6/93	14	1479	89	32	1600	57	1528	1.26
GFB	MarB	16/ 6/93	31/ 6/93	15	1603	68	39	1709	146	1674	0.24
GFB	AprA	1/7/93	15/ 7/93	14	1681	84	26	1792	46	1720	1.57
GFB	AprB	16/ 7/93	30/ 7/93	14	1795	61	26	1882	94	1814	0.72
GFB	MayA	1/8/93	15/ 8/93	14	1821	82	26	1929	103	1917	0.12
GFB	MayB	16/ 8/93	31/ 8/93	15	1931	68	30	2028	150	2067	-0.26
GFB	JunA	1/9/93	15/ 9/93	14	2074	83	35	2191	120	2187	0.04
GFB	JunB	16/ 9/93	30/ 9/93	14	2177	61	41	2278	198	2385	-0.54
GFB	JulA	1/10/93	15/10/93	14	2393	96	37	2526	161	2546	-0.12
GFB	JulB	16/10/93	31/10/93	15	2520	72	52	2643	247	2793	-0.61
GFB	AugA	1/11/93	15/11/93	14	2802	97	38	2937	142	2935	0.01
GFB	AugB	16/11/93	31/11/93	15	2936	70	47	3053	158	3093	-0.26
GFB	SepA	1/12/93	15/12/93	14	3102	101	26	3229	78	3171	0.74
GFB	SepB	16/12/93	30/12/93	14	3236	77	38	3351	166	3337	0.08

SUMMARY STATISTICS FROM PROJECTION RUNS BY MONTH (Cont) (FY 93)

NBox	Month	CurrDate	ProjDate	#Days	Curr	Proj	Pred	SumProj	Actual	SumAct	(SumProj - SumAct) Actual
SMT	OctA	1/1/93	15/ 1/93	14	57	637	279	973	1038	1038	-0.06
SMT	OctB	16/ 1/93	31/ 1/93	15	923	628	406	1957	1297	2335	-0.29
SMT	NovA	1/2/93	15/ 2/93	14	2382	497	288	3167	852	3187	-0.02
SMT	NovB	16/ 2/93	30/ 2/93	14	3101	312	330	3742	937	4124	-0.41
SMT	DecA	1/3/93	15/ 3/93	14	4173	494	390	5056	875	4999	0.07
SMT	DecB	16/ 3/93	31/ 3/93	15	4910	339	282	5531	981	5980	-0.46
SMT	JanA	1/4/93	15/ 4/93	14	6031	480	389	6899	698	6678	0.32
SMT	JanB	16/ 4/93	31/ 4/93	15	6813	375	565	7754	837	7515	0.28
SMT	FebA	1/5/93	15/ 5/93	14	7569	609	338	8516	778	8293	0.29
SMT	FebB	16/ 5/93	28/ 5/93	12	8392	319	302	9013	830	9123	-0.13
SMT	MarA	1/6/93	15/ 6/93	14	9172	537	319	10027	824	9947	0.10
SMT	MarB	16/ 6/93	31/ 6/93	15	9915	374	385	10674	1168	11115	-0.38
SMT	AprA	1/7/93	15/ 7/93	14	11159	512	290	11962	777	11892	0.09
SMT	AprB	16/ 7/93	30/ 7/93	14	11837	336	319	12492	1016	12908	-0.41
SMT	MayA	1/8/93	15/ 8/93	14	12951	472	287	13710	745	13653	0.08
SMT	MayB	16/ 8/93	31/ 8/93	15	13607	338	259	14205	742	14395	-0.26
SMT	JunA	1/9/93	15/ 9/93	14	14434	460	190	15084	164	14559	3.20
SMT	JunB	16/ 9/93	30/ 9/93	14	15052	383	394	15830	1546	16105	-0.18
SMT	JulA	1/10/93	15/10/93	14	16154	598	302	17053	711	16816	0.33
SMT	JulB	16/10/93	31/10/93	15	16914	421	395	17730	1071	17887	-0.15
SMT	AugA	1/11/93	15/11/93	14	17940	583	287	18810	689	18576	0.34
SMT	AugB	16/11/93	31/11/93	15	18745	434	357	19537	1061	19637	-0.09
SMT	SepA	1/12/93	15/12/93	14	19690	614	263	20566	528	20165	0.76
SMT	SepB	16/12/93	30/12/93	14	20494	405	397	21296	1027	21192	0.10
SFT	OctA	1/1/93	15/ 1/93	14	6	89	28	123	87	87	0.41
SFT	OctB	16/ 1/93	31/ 1/93	15	96	82	34	212	135	222	-0.08
SFT	NovA	1/2/93	15/ 2/93	14	227	67	21	315	108	330	-0.14
SFT	NovB	16/ 2/93	30/ 2/93	14	302	50	27	379	166	496	-0.71
SFT	DecA	1/3/93	15/ 3/93	14	501	65	29	595	97	593	0.02
SFT	DecB	16/ 3/93	31/ 3/93	15	578	48	24	650	131	724	-0.57
SFT	JanA	1/4/93	15/ 4/93	14	729	69	28	827	93	817	0.11
SFT	JanB	16/ 4/93	31/ 4/93	15	811	58	51	920	126	943	-0.18
SFT	FebA	1/5/93	15/ 5/93	14	949	79	25	1053	118	1061	-0.07
SFT	FebB	16/ 5/93	28/ 5/93	12	1034	45	24	1104	136	1197	-0.69
SFT	MarA	1/6/93	15/ 6/93	14	1202	67	27	1296	123	1320	-0.19
SFT	MarB	16/ 6/93	31/ 6/93	15	1279	60	31	1370	194	1514	-0.74
SFT	AprA	1/7/93	15/ 7/93	14	1519	79	27	1624	119	1633	-0.08
SFT	AprB	16/ 7/93	30/ 7/93	14	1589	54	29	1672	148	1781	-0.73
SFT	MayA	1/8/93	15/ 8/93	14	1786	67	27	1879	102	1883	-0.04
SFT	MayB	16/ 8/93	31/ 8/93	15	1854	52	27	1933	116	1999	-0.57
SFT	JunA	1/9/93	15/ 9/93	14	2003	63	13	2078	29	2028	1.73
SFT	JunB	16/ 9/93	30/ 9/93	14	2067	53	28	2148	148	2176	-0.19
SFT	JulA	1/10/93	15/10/93	14	2181	69	26	2276	87	2263	0.14
SFT	JulB	16/10/93	31/10/93	15	2260	54	30	2345	131	2394	-0.37
SFT	AugA	1/11/93	15/11/93	14	2399	68	20	2488	83	2477	0.13
SFT	AugB	16/11/93	31/11/93	15	2483	60	38	2581	140	2617	-0.25
SFT	SepA	1/12/93	15/12/93	14	2622	91	21	2735	86	2703	0.37
SFT	SepB	16/12/93	30/12/93	14	2706	70	36	2812	172	2875	-0.37

SUMMARY STATISTICS FROM PROJECTION RUNS BY MONTH (Cont) (FY 93)

NBox	Month	CurrDate	ProjDate	#Days	Curr	Proj	Pred	SumProj	Actual	SumAct	(SumProj - SumAct) Actual
OTH	OctA	1/1/93	15/ 1/93	14	32	304	106	442	567	567	-0.22
OTH	OctB	16/ 1/93	31/ 1/93	15	512	320	155	987	678	1245	-0.38
OTH	NovA	1/2/93	15/ 2/93	14	1271	276	99	1646	384	1629	0.04
OTH	NovB	16/ 2/93	30/ 2/93	14	1670	189	114	1973	499	2128	-0.31
OTH	DecA	1/3/93	15/ 3/93	14	2155	258	131	2544	500	2628	-0.17
OTH	DecB	16/ 3/93	31/ 3/93	15	2564	201	99	2864	936	3564	-0.75
OTH	JanA	1/4/93	15/ 4/93	14	3592	252	134	3978	1285	4849	-0.68
OTH	JanB	16/ 4/93	31/ 4/93	15	4026	215	211	4452	1744	6593	-1.23
OTH	FebA	1/5/93	15/ 5/93	14	6623	282	114	7019	1046	7639	-0.59
OTH	FebB	16/ 5/93	28/ 5/93	12	7079	163	96	7338	390	8029	-1.77
OTH	MarA	1/6/93	15/ 6/93	14	8056	238	90	8384	244	8273	0.46
OTH	MarB	16/ 6/93	31/ 6/93	15	8468	205	113	8786	210	8483	1.44
OTH	AprA	1/7/93	15/ 7/93	14	8508	259	100	8867	82	8565	3.68
OTH	AprB	16/ 7/93	30/ 7/93	14	8884	197	114	9194	411	8976	0.53
OTH	MayA	1/8/93	15/ 8/93	14	9000	238	77	9315	267	9243	0.27
OTH	MayB	16/ 8/93	31/ 8/93	15	9364	172	87	9623	322	9565	0.18
OTH	JunA	1/9/93	15/ 9/93	14	9587	208	82	9877	144	9709	1.17
OTH	JunB	16/ 9/93	30/ 9/93	14	9929	182	101	10213	266	9975	0.89
OTH	JulA	1/10/93	15/10/93	14	10002	250	88	10340	266	10241	0.37
OTH	JulB	16/10/93	31/10/93	15	10424	218	135	10778	431	10672	0.25
OTH	AugA	1/11/93	15/11/93	14	10701	273	104	11078	260	10932	0.56
OTH	AugB	16/11/93	31/11/93	15	11148	230	145	11523	423	11355	0.40
OTH	SepA	1/12/93	15/12/93	14	11384	288	71	11743	202	11557	0.92
OTH	SepB	16/12/93	30/12/93	14	11830	214	129	12174	378	11935	0.63
TOT	OctA	1/1/93	15/ 1/93	14	242	2136	1188	3567	3819	3819	-0.07
TOT	OctB	16/ 1/93	31/ 1/93	15	3918	2218	1663	7799	4669	8488	-0.15
TOT	NovA	1/2/93	15/ 2/93	14	8688	1769	1153	11610	3015	11503	0.04
TOT	NovB	16/ 2/93	30/ 2/93	14	11739	1051	1285	14075	3979	15482	-0.35
TOT	DecA	1/3/93	15/ 3/93	14	15690	1708	1405	18803	3326	18808	0.00
TOT	DecB	16/ 3/93	31/ 3/93	15	18820	1114	1050	20985	4400	23208	-0.51
TOT	JanA	1/4/93	15/ 4/93	14	23424	1643	1441	26509	4201	27409	-0.21
TOT	JanB	16/ 4/93	31/ 4/93	15	26747	1227	2081	30056	5086	32495	-0.48
TOT	FebA	1/5/93	15/ 5/93	14	32726	2029	1253	36009	3963	36458	-0.11
TOT	FebB	16/ 5/93	28/ 5/93	12	36220	1041	1102	38364	3443	39901	-0.45
TOT	MarA	1/6/93	15/ 6/93	14	40108	1795	1137	43040	2950	42851	0.06
TOT	MarB	16/ 6/93	31/ 6/93	15	43266	1226	1376	45868	4351	47202	-0.31
TOT	AprA	1/7/93	15/ 7/93	14	47390	1686	1064	50140	2576	49778	0.14
TOT	AprB	16/ 7/93	30/ 7/93	14	50269	1095	1169	52533	3777	53555	-0.27
TOT	MayA	1/8/93	15/ 8/93	14	53739	1547	994	56280	2780	56335	-0.02
TOT	MayB	16/ 8/93	31/ 8/93	15	56524	1049	1066	58638	3646	59981	-0.37
TOT	JunA	1/9/93	15/ 9/93	14	60147	1463	1095	62705	2317	62298	0.18
TOT	JunB	16/ 9/93	30/ 9/93	14	62771	1057	1502	65330	5037	67335	-0.40
TOT	JulA	1/10/93	15/10/93	14	67544	1813	1233	70589	3045	70380	0.07
TOT	JulB	16/10/93	31/10/93	15	70772	1244	1662	73678	4666	75046	-0.29
TOT	AugA	1/11/93	15/11/93	14	75270	1866	1276	78412	3058	78104	0.10
TOT	AugB	16/11/93	31/11/93	15	78691	1253	1601	81545	4694	82798	-0.27
TOT	SepA	1/12/93	15/12/93	14	83021	1939	1109	86069	2557	85355	0.28
TOT	SepB	16/12/93	30/12/93	14	86438	1207	1600	89244	4286	89641	-0.09

LIST OF REFERENCES

1. Buning, Daniel, MAJ U.S. Army, (Asst. Chief of Mission Division, PAE, USAREC), Information provided to the author through personal communications, 1993/1994.

2. Burris, Bryan D., *Missioning with Minimal Delayed Entry Program (DEP) Loss*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1993.

3. Headquarters, Department of the Army, Army Regulation 601-210, *Regular Army and Reserve Enlistment Program*, Washington DC, 1 Aug 91.

4. Headquarters, United States Army Military Entrance Processing Command, USMEPCOM PAM 25-5, *MEPCOM Testing Data Base*, Chicago, Illinois, 1988.

5. Headquarters, United States Army Recruiting Command, USAREC Manual 100-5, *Recruiting*, Fort Sheridan, Illinois, 1 Sep 89.

6. Headquarters, United States Army Recruiting Command, USAREC Manual 600-22, *Assignment of Enlistment Processing Responsibility,* Fort Sheridan, Illinois ,31 May 89.

7. Headquarters, United States Army Recruiting Command, USAREC Manual 601-73, *Personnel Procurement Missioning Procedures*, Fort Knox, Kentucky, 15 Feb 93.

8. Hershberger, John, MAJ U.S. Army, (Chief of Mission Division, PAE, USAREC), Information provided to the author through personal communications, 1993/1994.

9. Ross, S. M., *Introduction to Probability Models*, 4th ed., Academic Press, Inc., San Diego, California, 1989.

10. Szoka, John, LTC U.S. Army, (Chief of Recruiting Operations Center, USAREC), Information provided to the author through personal communication, 1993.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. CPT Scott G. Roesler 5
   204 Mabel Court
   Vine Grove, Kentucky 40175

4. Dr. Lyn Whitaker, Code OR/Wh 2
   Naval Postgraduate School
   Monterey, California 93943-5002

5. Dr. Harold Larson, Code OR/La 1
   Naval Postgraduate School
   Monterey, California 93943-5002

6. Director, PA&E 3
   Headquarters, United States Recruiting Command
   Fort Knox, Kentucky 40175-2171

7. Deputy Undersecretary of the Army 1
   for Operations Research
   Room 2E261, The Pentagon
   Washington D.C. 20310

8. Commander 1
   U.S. Army TRADOC Analysis Command
   ATTN: ATRC (Mr. Bauman)
   Fort Leavenworth, KS 66027-5200