NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CAPTURING THE FULL POTENTIAL OF THE SYNTHETIC THEATER OPERATIONS RESEARCH MODEL (STORM)

by

Christian N. Seymour

September 2014

Thesis Co-Advisors:	Thomas W. Lucas
	Dashi I. Singham
Second Reader:	Rachel Silvestrini

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
CAPTURING THE FULL POTENTIAL OF THE SYNTHETIC THEATER OPERATIONS RESEARCH MODEL (STORM)

5. FUNDING NUMBERS

6. AUTHOR(S) Christian N. Seymour

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)
N/A

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. IRB protocol number ____N/A____.

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

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)
The Synthetic Theater Operations Research Model (STORM) is the primary campaign analysis tool used by the Office of the Chief of Naval Operations, Assessment Division (OPNAV N81) and other Department of Defense organizations to aid in providing analysis to top-level officials on force structures, operational concepts, and military capabilities. This thesis describes how STORM works, analyzes the variability associated with many replications, and evaluates the trade-off between the expected number of replications and the precision and probability of coverage of confidence intervals. The results of this research provide OPNAV 81 with the ability to capitalize on STORM's full potential on a time-line conducive to its high-paced environment.

The distribution of outcomes is examined via standard statistical techniques for multiple metrics. All metrics appear to have sufficient variability, which is critical in modeling the combat environment. The trade-off for confidence intervals between the expected number of replications, precision, and the probability of coverage is very important. If a more precise solution and a higher probability of coverage are required, more replications are generally needed. This relationship is explored and a framework is provided to conduct this analysis on simulation output data.

14. SUBJECT TERMS: Synthetic Theater Operations Research Model (STORM), N81, campaign-level simulation, stopping rules, confidence interval procedures, StormMiner, simulation

15. NUMBER OF PAGES
97

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	UU

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CAPTURING THE FULL POTENTIAL OF THE SYNTHETIC THEATER OPERATIONS RESEARCH MODEL (STORM)

Christian N. Seymour
Lieutenant, United States Navy
B.A., Radford University, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2014

Author: Christian N. Seymour

Approved by: Thomas W. Lucas
Thesis Co-Advisor

Dashi I. Singham
Thesis Co-Advisor

Rachel Silvestrini
Second Reader

Robert F. Dell
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Synthetic Theater Operations Research Model (STORM) is the primary campaign analysis tool used by the Office of the Chief of Naval Operations, Assessment Division (OPNAV N81) and other Department of Defense organizations to aid in providing analysis to top-level officials on force structures, operational concepts, and military capabilities. This thesis describes how STORM works, analyzes the variability associated with many replications, and evaluates the trade-off between the expected number of replications and the precision and probability of coverage of confidence intervals. The results of this research provide OPNAV 81 with the ability to capitalize on STORM's full potential on a time-line conducive to its high-paced environment.

The distribution of outcomes is examined via standard statistical techniques for multiple metrics. All metrics appear to have sufficient variability, which is critical in modeling the combat environment. The trade-off for confidence intervals between the expected number of replications, precision, and the probability of coverage is very important. If a more precise solution and a higher probability of coverage are required, more replications are generally needed. This relationship is explored and a framework is provided to conduct this analysis on simulation output data.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AEW	airborne early warning
C2	command and control
CASA	Common Analytical Simulation Architecture
CG	cruiser
CLF	combat logistics force ship
COCOM	combatant commander
CONOPS	concepts of operations
CSG	Carrier Strike Group
CVN	nuclear powered aircraft carrier
DDG	destroyer
DOD	Department of Defense
EW	early warning
GAO	Government Accountability Office
HQ, USAF/A9	Headquarters, United States Air Force/Studies and Analyses, Assessments, and Lessons Learned
IADS	Integrated Air Defense System
ISR	intelligence, surveillance, and reconnaissance
ITEM	integrated theater engagement model
JSF	Joint Strike Fighter
LCAC	landing craft air cushion
LHD	landing helicopter dock
M	Marines
MIW	mine warfare ship
MRF	multirole fighter
MSQL	Mini Structured Query Language
N	Navy
NPS	Naval Postgraduate School
OPNAV N81	The Office of the Chief of Naval Operations, Assessment Division
QDR	Quadrennial Defense Review
R.V.	random variable

SAG	surface action group
SEED	Simulations, Experiments, and Efficient Designs
SS	submarine (non-nuclear)
SSGN	guided missile submarine, nuclear powered
SSN	submarine (nuclear)
STORM	Synthetic Theater Operations Research Model
SWEMP	Swiss Empire
UAV	unmanned aerial vehicle
WITTW	what it takes to win

EXECUTIVE SUMMARY

The Synthetic Theater Research Operations Model (STORM) is a pillar of campaign analysis conducted by various Department of Defense (DOD) organizations. The Office of the Chief of Naval Operations, Assessments Division (OPNAV N81) is one of the key users of STORM and it has a requirement to perform quick, turn-around analysis in the fast-paced and budget-constrained environment in which it operates. N81's analysis helps decision makers with force structure decisions, assists in developing operational plans, and helps to assess military capabilities. This thesis explains how STORM works, describes the variability inherent in STORM, and examines the trade-off between the number of replications and their associated precision with confidence intervals. The results of this research provide OPNAV 81 with the ability to capitalize on STORM's full potential on a time-line conducive to its high-paced environment.

STORM is a complex, stochastic, constructive, theater-level campaign simulation. Although there are hundreds of variables that could be analyzed, this thesis focuses on four metrics: The number of blue ships remaining at simulation termination, the number of red ships remaining at simulation termination, the day in which blue forces achieve air supremacy, and the number of blue multirole fighter missions flown. The unclassified scenario known as Punic21 is used to demonstrate how STORM works, to determine the variability in STORM output, and to examine the trade-off relationship in determining the number of replications to perform. Due to STORM's inherent stochasticity, no input variables were changed; enough variability was found by changing the random seed in different replications.

The complexity of STORM can be realized in the approximately one thousand pages of instructions found in the *STORM User's Manual* (Group W., 2012c), *STORM Analyst's Manual* (Group W., 2012a), and *STORM Programmer's Manual* (Group W., 2012b). This thesis focuses on the highlights from these resources by explaining how STORM works and the built-in analytical capabilities internal to it. After gaining an understanding of how STORM works and how N81 uses it, the next challenge was analyzing the inherent stochasticity. This is accomplished by examining the distribution

of outcomes of the four metrics mentioned above. Summary statistics and histograms of the metrics show the variability of STORM, as the outcomes are dispersed around the mean without changing any input variables. In addition, normality testing is conducted through hypothesis testing, using formal normality testing, and reviewing figures that can help determine normality.

The largest portion of the analysis examines the trade-off between the number of replications and the associated precision and probability of coverage. Precision is the length of the confidence interval surrounding the estimated mean (e.g., the difference between [16.95, 17.05] and [15, 19] may be critical to a decision maker). The probability of coverage is the probability that the confidence interval contains the true, unknown mean. Replications have a cost of time and memory. As a result, the cost must be minimized to a level where the analyst is content with the precision and coverage obtained. The trade-off relationship is analyzed using previous stopping rule research conducted by Singham (Singham, 2010), which laid a framework for resampling original simulation output and running the simulation until a calculated half-width is less than the specified precision (delta). The half-width is the value that determines the precision around the mean in developing a confidence interval. Once the half-width is less than the required precision, we evaluate whether the confidence interval covers the true mean. Only after many replications can we estimate the probability of coverage. The relationship between the expected number of replications, precision, and probability of coverage is plotted, which helps the analyst visualize the trade-offs. As a result of this research, this methodology is being used in software, known as StormMiner, which is being built by the Naval Postgraduate School's (NPS's) Simulations, Experiments, and Efficient Designs (SEED) Center to develop postprocessing tools to increase the efficiency in analyzing the output data. See http://harvest.nps.edu for more information on the SEED Center.

The trade-off relationship can be seen in Figure 1. The different Kstart values are broken out; Kstart is the minimum number of replications taken prior to examining the relationship between delta and the half-width. For an extremely small value of delta, the expected number of replications gets very large. Likewise, for a large value of delta, the

sleep expected number of replications goes down. This thesis explores this relationship in depth to include the coverage obtained, which increases with a higher expected number of replications.

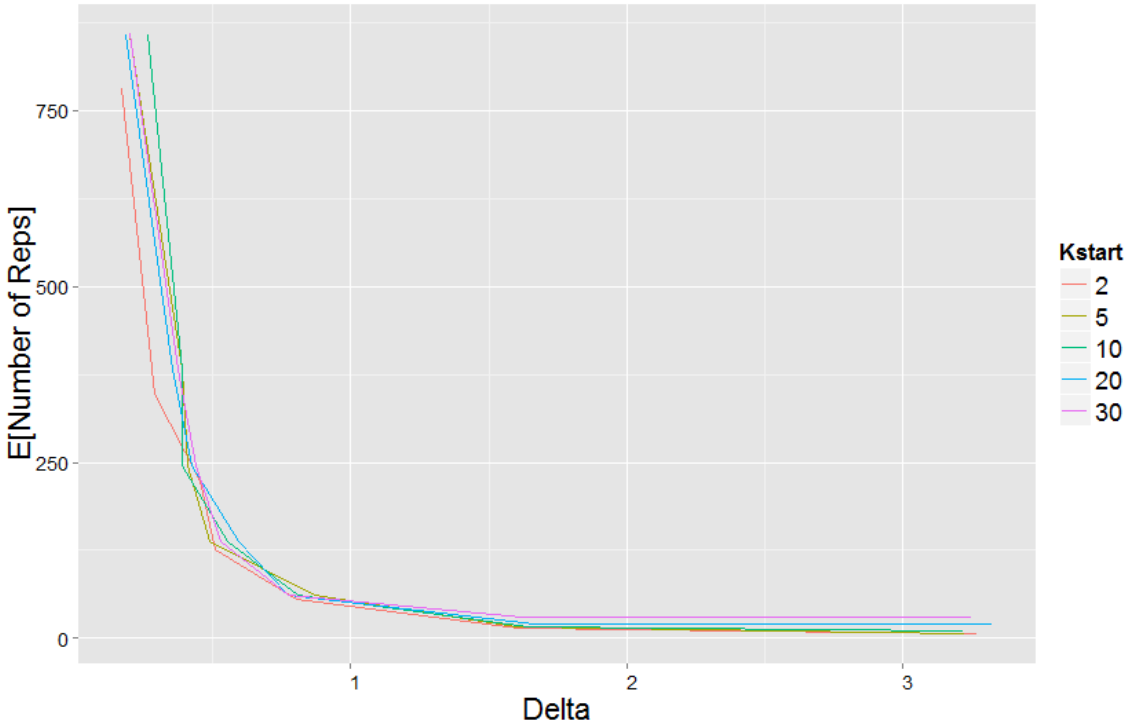

Figure 1. Expected number of replications versus delta for red ships remaining. Desired confidence equal to 95%.

LIST OF REFERENCES

Group W. (2012a). *STORM: Analyst's Manual Version 2.3*. Fairfax, VA: Group W.

Group W. (2012b). *STORM: Programmer's Manual Version 2.3*. Fairfax, VA: Group W.

Group W. (2012c*). STORM: User's Manual Version 2.3*. Fairfax, VA: Group W.

Singham, D. I. (2010). *Analysis of sequential stopping rules in simulation experiments* (Doctoral dissertation). Retrieved from http://escholarship.org/uc/item/3hb6p7bg

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Completing the coursework and this thesis at NPS has been the biggest intellectual challenge I have ever experienced. I would like to give a very special thanks to Captain Jim Kilby, USN, for encouraging me to pursue this academic endeavor and working to get me detailed here. In addition, the professors in the Operations Research and Math Departments did an incredible job of transforming me into someone who can identify, explore through analysis, and draw conclusions on a wide variety of problems. I am most thankful for the patience, guidance, and advice that my co-advisors, Dr. Tom Lucas and Dr. Dashi Singham, gave me throughout this process. Their expert knowledge and experience in converting military officers into intellectual achievers, is a direct reflection of their desire to help us succeed.

In addition, other members of the SEED team at NPS were huge resources. Steve Upton enabled me to run STORM locally at NPS and is the best resource—even better than Google—for all the roadblocks I encountered in R. Mary McDonald was instrumental in helping me understand the output of STORM and answered many questions that I came across on approaches to analyzing data. Drs. Paul and Susan Sanchez were also critical in answering simulation questions that I encountered and were a great resource for all aspects of this project.

I would also like to thank Professor Wayne Hughes and Professor Jeff Applegate for encouraging me to succeed, constantly providing me with new ideas, and having an open door that I always could knock on when I was frustrated. Dr. Rachel Silvestrini, my second reader, and the first person to give me an A at NPS, was instrumental as an outside set of eyes and a resource for helping me develop many of the plots presented in my thesis.

My wife, Laura, and my three children, Riley, Levi, and Adelyn, deserve as much credit for my success as I do. They supported me, although I spent many hours away, in working to complete this academic endeavor. I am also thankful to God, as it is only through His grace and mercy that I have been able to succeed.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The U.S. Navy and other Department of Defense (DOD) organizations use the Synthetic Theater Operations Research Model (STORM) as a campaign analysis tool to provide decision makers with information regarding force structures, operational concepts, and military capabilities. Force structure decisions revolve around the acquisition of military assets worth many billions of dollars. For example, the Joint Strike Fighter (JSF) program has an estimated total acquisition cost of almost 400 billion dollars (United States Government Accountability Office [GAO], 2012). Campaign simulations like STORM help decide whether a particular asset and its associated capabilities are worth the cost. This is especially the case when DOD-wide events, such as the *Quadrennial Defense Review* (QDR), are executed. Powerful simulations like STORM provide information to aid decision makers involved in critical activities such as the QDR.

STORM was originally sponsored by Headquarters, United States Air Force/Studies and Analyses, Assessments, and Lessons Learned (HQ, USAF/A9). The U.S. Navy wanted to capitalize on the potential of STORM, specifically because it is a stochastic modeling environment. STORM's predecessor is the Integrated Theater Engagement Model (ITEM), which is deterministic. As a result, the Office of the Chief of Naval Operations, Assessment Division (OPNAV N81) commissioned a study to determine whether adding a maritime capability to STORM was feasible, which it was, and a new project called STORM+ was created (Sweeney, Hamman, & Biemer, 2011).

OPNAV N81 uses STORM to provide warfighting analysis to senior Navy and DOD leadership to inform operational planning and acquisition decisions. It currently has challenges using STORM due to the time frame in which it operates. A STORM model can take anywhere from 4 minutes to 12 hours to run a set of replications, depending on the complexity. In addition to the run time, analysts must also process output that can be millions of lines of data. This postprocessing can take up to three days for a single set of replications. Moreover, building a new scenario in STORM may take upwards of a year.

A major encumbrance to responsive analysis for OPNAV N81 is the time required in postprocessing the output data, which can take up to a few days. As a result, a project was started with the Simulation, Experiments, and Efficient Designs (SEED) Center at the Naval Postgraduate School (NPS) to develop postprocessing tools to increase the efficiency in analyzing the output data (see http://harvest.nps.edu for more information on the SEED Center). The main effort of the project was to develop a program that harvested output data from STORM, performed statistical analysis, and generated figures and plots that can help analysts in providing quick, turn-around analysis. This thesis supports that effort by focusing on developing a process for dynamically determining the minimum number of replications required to provide a desired level of precision.

STORM is a stochastic simulation and, therefore, requires multiple replications to be made for a set of inputs. Replication allows analysts to better understand output measures (e.g., blue systems lost), evaluate the variance of responses, and determine the distributions of outcomes (Lucas, 2000). In addition, more replications add precision to these mean performance measures and also help to identify unique events or outliers.

Due to the cost in time and computer memory, minimizing the number of replications required to capture most of the information becomes an important factor. Currently, STORM normally runs 30–50 replications unless there is not enough time available. Creating a sequential, dynamic method to determine the appropriate number of replications will ensure that analysts make enough runs to provide good, statistical information in a timely fashion. The quality of the simulation output data is analyzed by conducting trade-off analysis between the number of runs and quality of the confidence intervals.

A. LITERATURE REVIEW

Processing power and computer memory have increased substantially in the past few decades. Military researchers have been able to capitalize on these improvements since stochastic simulations require multiple replications, thereby requiring more processing power and memory than a single run of a deterministic simulation. The advantages of stochastically simulating a combat model versus strictly modeling deterministically are numerous, but the bottom line is that combat is inherently

uncertain—and stochastic models are often the only way to capture the variability in outcomes that may potentially occur (Lucas, 2000). OPNAV N81 decided to investigate simulating in the same stochastic simulation environment that the Air Force was using (known as STORM) as early as 2006 (Sweeney et al., 2011, pp. 327–328).

The DOD divides simulations into four broad levels: campaign, mission, engagement, and engineering. Each level models a different level of detail. Understanding the different levels, and how much detail is required, is fundamental to proper modeling. An engineering simulation models an entity with the most detail, such as a missile's navigation component. On the other end of the spectrum, a campaign model is less detailed and is used to study force–on–force engagements over an extended time horizon (e.g., two weeks to three months), based on forces, orders of battle, and probabilities of kill (Hawley & Blauwkamp, 2010).

The developers of STORM publish three manuals that serve as reference documents for organizations using STORM. The *User's Manual* is written as a resource for analysts operating STORM as software (Group W, 2012c). The *Programmer's Manual* is designed for individuals who develop, maintain, and modify source code in STORM (Group W, 2012b). Analysts employing STORM as a campaign-level tool, reporting credible results to decision makers, use the *Analyst's Manual* as their primary source of information (Group W, 2012a). All of these resources were used in for the composition of this thesis.

The analytical work done for this thesis focuses on applying techniques proven in Dr. Dashi Singham's research for selecting appropriate sequential sampling rules with stochastic simulations. Her dissertation describes how to optimally obtain nominal coverage with the minimum expected number of replications (Singham, 2010). It proved that these optimal stopping rules can be applied to independently and identically distributed data, and that the performance of these rules can be quantified (Singham & Schruben, 2012). It also proved that optimal stopping rules can be found for data that is not normally distributed by modifying various input parameters (Singham, 2014).

B. RESEARCH QUESTIONS

It is not the intent of this thesis to analyze an actual existing campaign model, but rather to develop and implement a postprocessing tool on an unclassified scenario (Punic21) that comes with the STORM installation kit. As a result, this thesis is guided by the following questions:

- Does the inherent stochasticity in STORM provide enough variability to adequately model combat? Can the output be considered approximately normally distributed?
- What methods can identify the number of replications needed for a given scenario to achieve desired statistical significance in the mean output? What is the trade-off between the number of replications completed and the precision and confidence achieved by the procedure?
- What techniques can help analysts more efficiently process and analyze the simulation output from STORM?

C. BENEFITS OF THIS THESIS

This thesis provides OPNAV N81 with the ability to capitalize on STORM's full potential on a time line conducive to the high-paced environment that their analysts work in. A foundation for future research is also laid by describing STORM in detail, thereby enabling future researchers to gain a high level of understanding without attending a course on STORM. The main focus of this research is to ensure that analysts at OPNAV N81 have a methodology to determine the appropriate number of replications required to build a high level of confidence, based on their desired precision.

D. METHODOLOGY

This thesis begins with an in-depth review and provides an understanding of STORM. The variability of STORM due to its stochastic nature is analyzed; this analysis builds the foundation for stopping rule criteria. The stopping rule is applied to four metrics to obtain the performance (in terms of coverage and expected replications) for various stopping rules (with inputs of desired confidence and precision). This information is used in developing a tool that analysts at OPNAV N81 can use to assess the accuracy

of their results. In addition, members of the SEED Center are developing a broader postprocessing tool that incorporates the research on stopping rules completed in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

II. OVERVIEW OF STORM

To facilitate a basic understanding of STORM and a baseline for future research in the SEED Center, this chapter provides a detailed description of how STORM works, which includes its characteristics and framework, input and output, and how the U.S. Navy uses it. STORM is a stochastic, closed-form analytical simulation of air, space, ground, and maritime planning and execution. It is a campaign-level simulation designed to help decision makers evaluate military strategy and capabilities in a theater of operations.

A. STOCHASTIC SIMULATION

> In short, so-called mathematical, factors never find a firm basis in military calculations. From the very start there is an interplay of possibilities, probabilities, good luck and bad that weaves its way throughout the length and breadth of the tapestry. (Clausewitz, 1832, pp. 86)

STORM is a complex, very high-dimensional, stochastic, campaign-level simulation that replaced its deterministic predecessor (a model known as ITEM) at OPNAV N81. As Clausewitz indicates to in the quote above, the nature of combat, along with fundamental mathematical principles, implies that most combat simulations should be stochastic because of combat's inherent randomness (Lucas, 2000). For instance, a basic, deterministic Lanchester equation has one set of inputs and, therefore, only provides one output. In stochastic modeling, the same set of inputs provides a range of outcomes when the random seed(s) are changed for each replication.

1. Arguments for a Deterministic Combat Model

A common argument for using a deterministic model is that "A good point estimate is sufficient for my purposes" (Lucas, 2000, pp. 10). The point estimate, however, is frequently biased and does not provide information about variability (Lucas, 2000). Many times, a decision maker might know the outcome is in his favor; on average, however, there is a significant chance that the outcome will be unfavorable. For example, if the probability of failure is one percent on a five million dollar project, the decision

maker might be willing to accept that risk. However, what if there is an eight percent chance of failure: Is the decision going to be different? Other arguments for deterministic models can be similarly contested (Lucas, 2000). In a meeting with the developers of STORM, they alluded to the fact that a deterministic model is a just an intuition confirmation device with knobs that can be changed to get any result the analyst wants. In other words, since there is no variability, one can adjust the input levels to get any result one likes. This is not always the case with a stochastic simulation, since there may exist chances for an unlikely result—even with the inputs set at generally favorable levels.

2. Arguments for a Stochastic Model

As stated previously, combat is inherently stochastic. Many uncertainties arise in combat, such as outcome, process, future, and decision uncertainty (Lucas, 2000). These factors are impossible to determine exactly. No battle fought will ever be exactly the same as another because of changing technologies, terrain, strategy, and human variability. As a result, the factors must be varied over a probability distribution to provide robust results of statistical significance.

The stochastic nature of STORM comes from the numerous data input parameters specified from 12 common probability distributions—including, but not limited to, the normal, binomial, and uniform. For example, if a ship has a damage and repair capability, the amount of time it takes for the ship to be repaired may be pulled from a uniform distribution, with a mean of three hours and a standard deviation of six hours. Random variability exists throughout STORM, including in other areas such as the probability of a hit or the probability of an intercept.

B. STORM—A CONSTRUCTIVE SIMULATION

STORM is a constructive simulation. A constructive model involves simulated entities, operating simulated systems, making decisions and interacting. Real people prescribe the decision-making logic to such simulations, but are not involved in determining the outcomes once the inputs are provided (Department of Defense, 2010). In STORM's case, at OPNAV N81, analysts select input data, build a scenario—including the development of concepts of operations (CONOPS)—and execute the

simulation. The simulation runs without a human in the loop and the analyst receives the output data at the conclusion of the run (or set of runs, consisting of many replications).

C. STORM—A CAMPAIGN SIMULATION

STORM is a campaign-level simulation. A campaign is a series of related major operations aimed at achieving strategic and operational objectives within a given time and space (Department of Defense, 2014). DOD simulations are normally classified into the following four categories: engineering, engagement, mission, and campaign. The spectrum of differences is wide. A campaign model is one which is used to determine, for example, the best mix of "blue" forces to battle "red" forces by focusing on order of battle and broad probabilities of kill (Hawley & Blauwkamp, 2010). An engineering simulation, which is at the other end of the spectrum in terms of detail, might only model a certain weapon system's components and interactions. An example of an engineering model would be exploring the relationship of the weight of a bomb and the range of an aircraft. As the weight of the bomb increases, the range of the aircraft decreases. It might be ideal to include the level of detail in an engineering simulation in all simulations; however, at the campaign level, it is virtually impossible to represent that much information for every entity. The result would be an extremely long run time and large amounts of memory required in order to get a single run. This, of course, is not conducive to the timeline in which current staff at OPNAV N81 operates.

A theater in the context of military applications is the geographical area for which a commander of a geographic combatant command has been assigned responsibility (Department of Defense, 2014). STORM is typically utilized to simulate a single theater or combatant commander (COCOM), in a time frame of weeks to months, with a goal of completing operational objectives.

D. STORM CONCEPTUAL MODEL

STORM was originally developed as a campaign simulation by the Air Force. The interesting approach in design that the developers took, however, was to ensure that the model adapted an approach inclusive to definition, design, and development. The end goal was a simulation that can be an interservice tool. Such a tool might reduce the

"modeling wars" throughout the DOD. In addition, STORM was not hard-wired to include only current day issues, but, instead, has the capability to include the evolving environment of doctrine and operational concepts at low cost. This will maximize STORM's use over an operational life of perhaps 20 or more years.

1. COMMON ANALYTICAL SIMULATION ARCHITECTURE

To maintain the flexibility desired, STORM employs the common analytical simulation architecture (CASA) decoupling components and applications, which endows modularity and segmentation, and insulates the system from local changes within individual segments. This enables STORM to operate with three different relational databases: Mini Structured Query Language (MSQL), Microsoft Office, and Oracle. In addition, this architecture enabled a database switch to be completed in less than two human days of programming effort (Group W, 2012a).

2. STORM'S LOGICAL DESIGN

STORM models military operations from the real world with five classes: command and control (C2) manager, asset, intelligence manager, environment, and interaction manager. The flow concept of these classes can be seen in Figure 1.

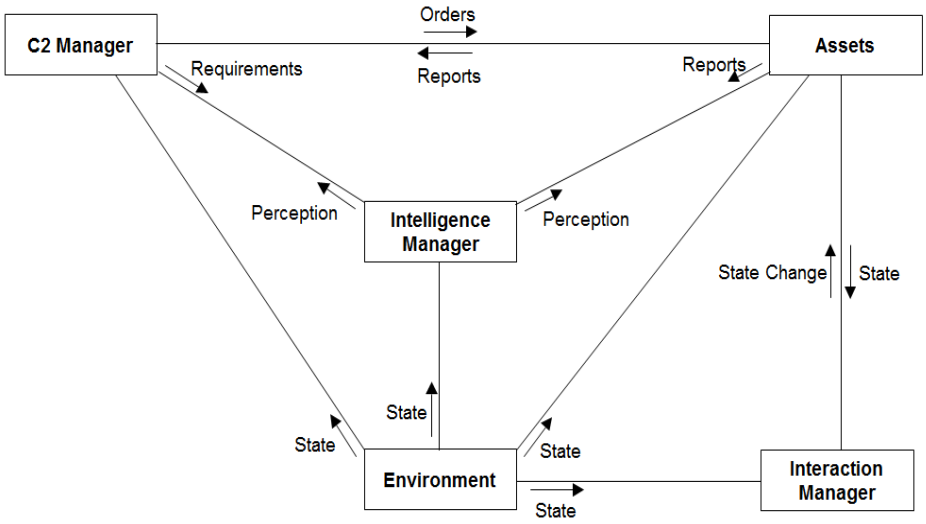

Figure 1. STORM's conceptual model (from Group W, 2012a).

3. ASSETS

Assets in STORM are entities that act or are acted upon. Therefore, they have the ability to complete activities like move, attack, conduct surveillance, or be killed. Examples of three different types of assets can be seen in Table 1.

Surface Asset	Air Asset	Orbital Asset
Armored Units	Aircraft	Satellites
Ships	Squadrons	Space-based platforms
Airbases	Munitions	
Logistics Nodes	Unmanned Aerial Vehicle (UAV)	

Table 1. Asset examples from STORM.

Assets are tasked by the C2 manager and receive perceptions from the intelligence manager, weather changes from the environment manager, and state changes from the interaction manager. They also communicate status reports and state changes to the intelligence manager and interaction manager, respectively. Each type of asset has explicit information embedded in it such as mobility, location, and intelligence, surveillance, and reconnaissance (ISR) characteristics.

4. ENVIRONMENT

The purpose of the environment class is to model real-world environmental conditions such as time of day, weather, and terrain. The environment in which an asset is operating affects its capabilities. For example, environmental factors, such as high sea state, darkness, and dense fog, may affect the ISR capabilities of numerous assets.

5. INTERACTIONS

Interactions take place when two or more assets have the opportunity to affect one another. The three types of interaction managers are the motion managers, adjudication managers, and support managers. The motion manager is responsible for the movement of assets in response to their tasking and battle space dynamics, subject to resource and environmental constraints. The adjudication manager provides the result of engagements between two or more assets in combat, sensing, or communication missions. The support

manager enables the movement and interaction of assets, subject to resource and environmental constraints, such as airbase operations (Group W, 2012a).

6. INTELLIGENCE

The intelligence manager provides perceptions to the C2 manager and assets. Information is gathered by different ISR platforms and analyzed to provide information like targeting data. Intelligence is not always correct—and, therefore, can lead to bad targeting data.

7. COMMAND AND CONTROL

The command and control (C2) manager tasks and receives reports from assets. In addition, requests for intelligence are sent to the intelligence manager, and perceptions are sent from the intelligence manager back to the C2 manager. The objective of the C2 manager is to coordinate asset behaviors to meet operational and strategic goals. The decision-making process is modeled using optimization techniques and other algorithms.

E. INPUT FILES

The information that populates the above-mentioned classes are contained in input files. The PUNIC21 scenario utilized in this thesis has over 100 input files. The files are available to view through STORM Front, which comes with the standard STORM installation. It is a tedious job to understand what each file contains. An example of an input file can be seen in Figure 2. The file designates the naval commanders for Allied and Red forces.

```
/*****************************************************************************

    $Id: navalcommand.dat,v 1.1 2012/01/18 23:48:05 rlennox Exp $

    File:   data/baseline_test/punic21/navalcommand.dat
    Rev:    1-1
    Date:   01/18/2012

    Developed for the U.S. Government under contract(s):
        N00178-04-D-4119

    Classification: UNCLASSIFIED

*****************************************************************************

    Define naval commands.

*****************************************************************************

    Modification Log:

    1-1:    01/18/2012  lennox/lburdette
        : Initial implementation of Punic21 scenario
        : for Detached Units Task 23DI0608.

*****************************************************************************/


Begin Naval_Command_File

Begin Naval_Command_List

ID: "Carthage Naval Command" {
    Side:       'Allied Coalition"
    ID: 'Carthage Naval Fleet" {
    }
    ID: 'Anglo Republic Naval Fleet" {
    }
    ID: 'Blue Submarine Fleet" {
    }
}

ID: "SWEMP Naval Command" {
    Side:       'Red"
    ID: 'Western SWEMP Naval Fleet" {
    }
    ID: 'Eastern SWEMP Naval Fleet" {
    }
    ID: 'Red Submarine Fleet" {
    }
}
```

Figure 2. Input file example for naval command.

F. OUTPUT DATA IN STORM

A majority of the information from a replication can be found in the dbase.out and debug.out files. The dbase.out files contain raw data and must be processed through a relational database before any analysis can be done. The processing of data takes place in the data warehouse that is built into STORM. Once the data is loaded into the data warehouse, STORM contains three analyst tools (the Map Tool, the Graph Tool, and the Report Tool) designed to analyze and view the data. Although these tools are very easy to

use and provide a quick way to look at different metrics, they are lacking in some desired capabilities. For example, not all output data is available to view in these tools. A programmer has the ability to write scripts in order to gain access to the information not included, but the typical analyst will usually require some additional training to accomplish this task.

1. Map Tool

The purpose of the Map Tool is to visually explore interactions taking place between different assets over time in a geographic region. The user is able to choose the time frame, how quickly the visualization of the simulation appears, and the geography and assets that they would like to be displayed. A screen shot of the Map Tool can be seen in Figure 3.

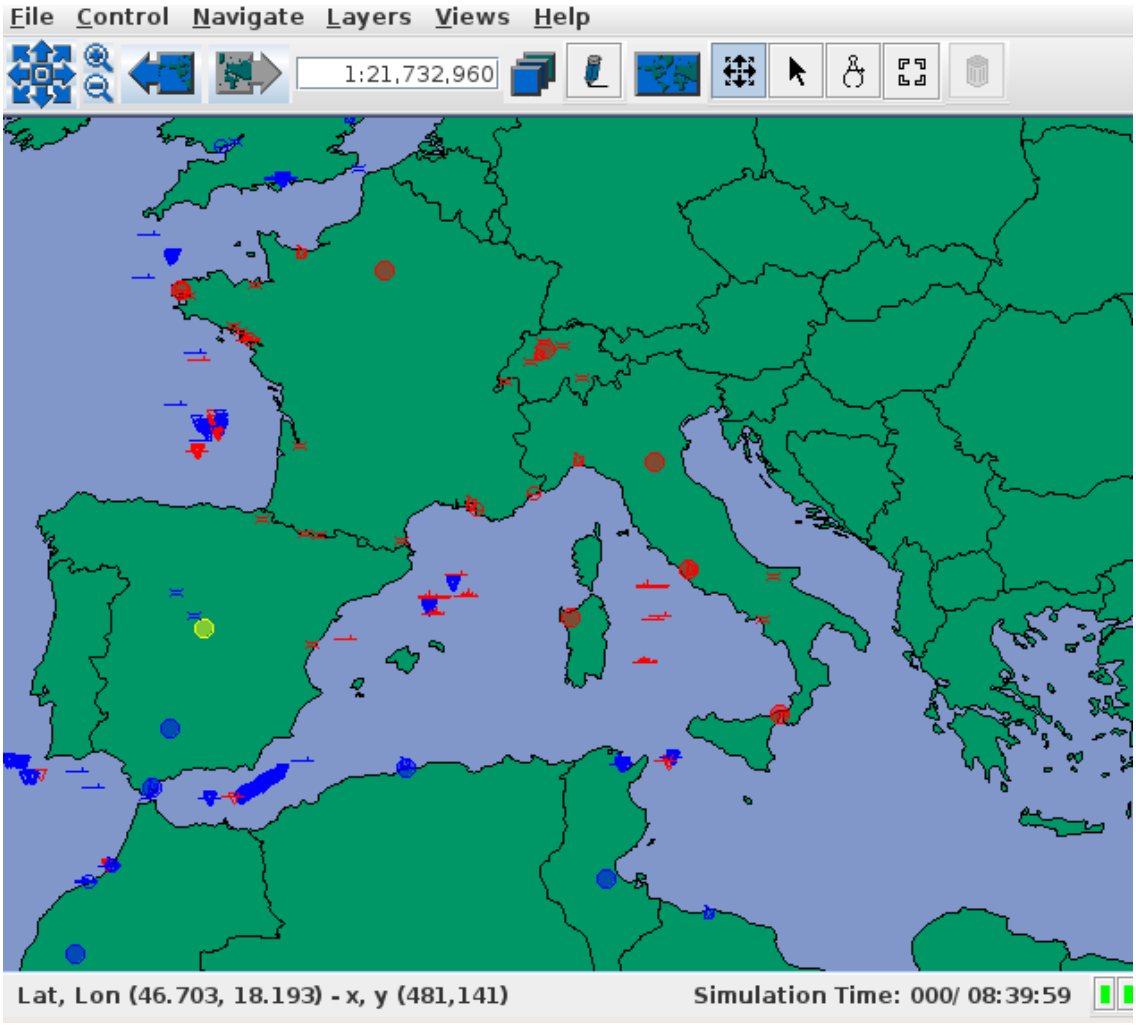

Figure 3. Screenshot of the Map Tool in STORM.

2. Graph Tool

The Graph Tool is designed to quickly and easily pull and graph user selected data from the data warehouse. Not all data is available in the Graph Tool, but there is a sufficient amount to gain insight rapidly on key output metrics of the simulation. An example of a Graph Tool output can be seen in Figure 4, which reveals the number of ships remaining for blue and red forces at the end of each day in the 20-day simulation.

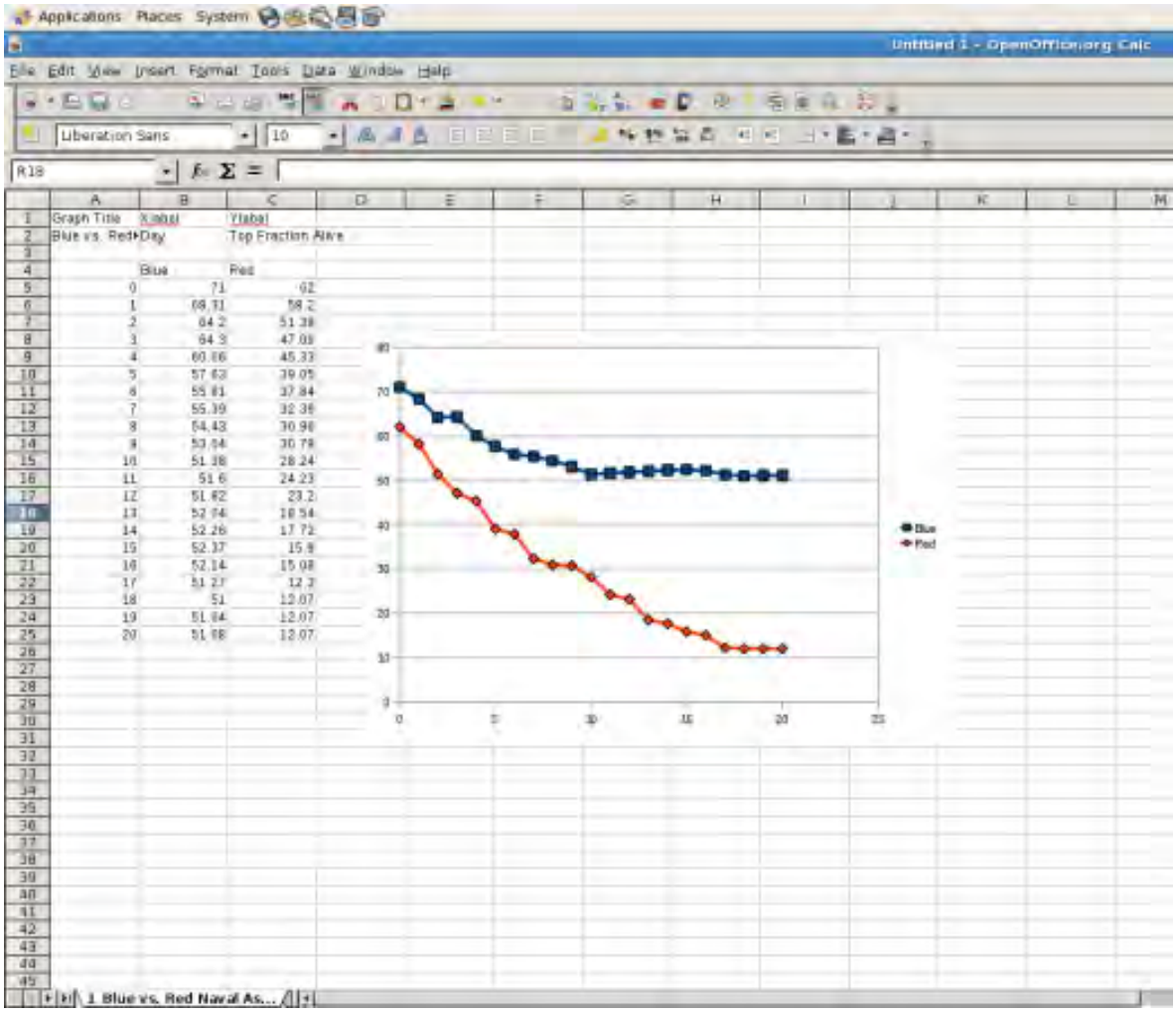

Figure 4. Screenshot of the Graph Tool in STORM.

3. Report Tool

The Report Tool function in the Study Tools of STORM allows the user to collect and organize specific data that can be further exported to conduct additional analysis. There is a wide selection of data available to collect in the Report Tool; however, like the Graph Tool, it is not inclusive of all output data from a simulation run. A nice feature of the Report Tool is the ability to export some of the data directly into a .csv file for analysis. An example of the output of the Report Tool can be seen in Table 2. This reflects a user–selected criterion to see which ships in the Blue Atlantic Surface Action Group (SAG) killed Red Cruisers on Day 2.

Daily Maritime Kills (Rep Matrix)

Day	Killer Side	Killer Asset	Killer Naval Unit	Victim Side	Victim Target Type	1
2	Allied Coalition	Anglo Republic CSG S Cruiser 1	Blue Atlantic SAG	Red	Red Cruiser	1
2	Allied Coalition	Anglo Republic CSG S Cruiser 2	Blue Atlantic SAG	Red	Red Cruiser	1
2	Allied Coalition	Anglo Republic CSG S Destroyer 1	Blue Atlantic SAG	Red	Red Cruiser	1
2	Allied Coalition	Anglo Republic CSG S Destroyer 2	Blue Atlantic SAG	Red	Red Cruiser	1

Table 2. Screenshot of the Report Tool in STORM.

G. PUNIC21 SCENARIO IN STORM

For STORM to execute a run, a scenario is needed as input. The STORM installation comes with two unclassified baseline scenarios. This thesis focuses on one of these scenarios, known as Punic21, in particular, due to its strong maritime focus. This section describes the order of battle, geography, and phases of the war to provide the reader with a basic understanding of how the scenario plays out.

The blue forces consist of two allied nations known as the Anglo Republic and Carthage. The red forces are made up of the Swiss Empire (SWEMP). Tensions have recently increased between Carthage and the SWEMP due to the SWEMPs goal of expansion. The SWEMP secretly lays mines in the vicinity of Gibraltar to slow Carthage's attempt to resupply the Anglo forces arriving in Spain. The SWEMP forces initiate attacks against Anglo naval forces and Integrated Air Defense Systems (IADS).

1. Order of Battle

Based on the premise that this scenario is largely a naval campaign, the order of battle includes the blue and red naval and air forces. For a campaign that centered on land operations, the order of battle would include land forces such as Army divisions, tanks, and artillery.

a. Naval Assets

The force structure of the naval assets can be seen in Table 3. The blue forces have an additional carrier, mine warfare ships, and an amphibious capability. The red forces have a few more destroyers and cruisers.

	BLUE FORCES	RED FORCES
Cruiser (CG)	8	14
Destroyer (DDG)	24	29
Nuclear Powered Aircraft Carrier (CVN)	3	2
Submarine Nuclear (SSN)	10	11
Guided Missile Submarine, Nuclear Powered (SSGN)	1	0
Mine Warfare Ship (MIW)	2	0
Landing Craft Air Cushion (LCAC)	3	0
Combat Logistics Force Ship (CLF)	11	3
OILER	6	3
Landing Helicopter Dock (LHD)	3	0
TOTAL	71	62

Table 3. Naval order of battle.

b. Air Assets

The blue forces have a slightly larger air capability, with additional multirole fighters (MRFs), fighters, and helicopters. The breakdown of the air forces can be seen in Table 4.

	Blue Naval	Blue Air Force	Red Naval	Red Air Force
MRF—N	120	0	100	0
MRF—M	40	0	0	0
MRF—Tanker	15	0	0	0
MRF—EW	15	12	0	10
MRF	0	138	0	144
Fighter	0	70	0	64
Vertical Assault	40	0	0	0
AEW	9	12	3	10
MPA	12	0	8	0
Bomber	0	32	0	32
Tanker	0	36	0	0
UAV (ISR)	0	16	0	16
Airlift	0	24	0	24

Table 4. Air order of battle. Multirole fighter (MRF), Navy (N), Marines (M), early warning (EW), airborne early warning (AEW), Intelligence-surveillance-reconnaissance (ISR), unmanned aerial vehicle (UAV).

2. GEOGRAPHY

The area of conflict is located in the Mediterranean Sea, the Bay of Biscay, and the English Channel. The land geography in the scenario is Northwest Africa and Western Europe. Figure 5 reflects a geographical outline with the location of current military forces.

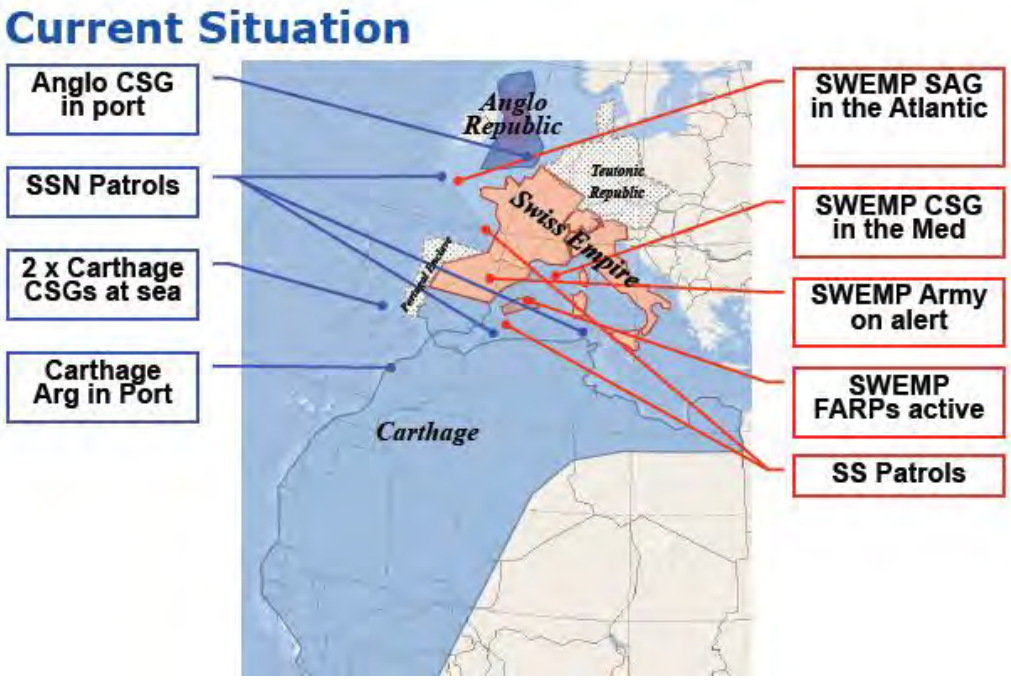

Figure 5. Current blue and red force layout in geographical perspective (STORM).

3. PHASES OF THE CAMPAIGN

The scenario is separated into four phases: The Battle of the Atlantic, the Battle of the Mediterranean, the fight for Spain, and the fight for Italy. These phases overlap, but generally take place in the order listed above. Although no input variables are changed, metrics that relate to these events are analyzed through the rest of this thesis and take place at different times, due to the stochasticity of STORM.

III. VARIABILITY IN STORM OUTPUT

Given STORM's inherent stochasticity, the variability of outcomes must be investigated to fully understand the power of this simulation. Chapter III focuses on analyzing the distribution of outcomes. This provides a basis for understanding how many replications are required, which will be explained in Chapter IV. In addition, key metrics are analyzed to determine whether the distributions of outcomes are approximately normal. Meeting the normality assumption allows the use of a wide variety of statistical techniques, such as developing valid confidence intervals. This exploration looks at the variability of STORM's output without changing any of the input variables. That is, we are taking many replications, while varying only the random number seed.

Millions of lines of output appear in the debug.out file from a single large campaign-level set of replications from STORM. Recall from Chapter II that this file is harvested for specific data, which is loaded into the data warehouse and made available in the Study Tools provided by STORM. The metrics reviewed in this chapter bypass the data warehouse load and the Study Tool. The metrics are directly pulled from the debug.out file, via specially developed scripts, and exported to an Excel file to be analyzed. The software programs used to conduct the analysis were Excel, R, and JMP.

The following categories of metrics are analyzed in this chapter:

- Blue and red force levels.
- A what-it-takes-to-win (WITTW) metric; specifically, the time at which blue achieves air supremacy.
- A high-variance metric; the number of blue, multirole fighter missions flown.

A. FORCE LEVELS AT SIMULATION TERMINATION

One of the most important metrics of any campaign simulation is to look at force levels at the beginning and end of the simulation. This information can provide breakpoints on how many and which types of forces are required to build the desired

confidence in how, and whether, the enemy should be engaged. The blue and red force levels analyzed in this section include blue and red ships remaining at simulation termination.

Figures 6 (blue) and 7 (red) are histograms for force levels at the end of the 20-day terminating simulation for 110 replications from the Punic21 scenario. The data appears to be distributed around the mean, displaying STORM's inherent stochasticity, since no input variables other than the random number seed were ever changed. Summary statistics can also be seen in Table 5. The number of red ships remaining has a slightly smaller standard deviation than the number of blue ships remaining and, therefore, a slightly smaller 95 percent confidence interval. The ranges of the number of remaining ships are similar for both red and blue. In both cases, the median and mean are virtually equal. In addition, both distributions are slightly skewed to the left; the blue force remaining (skewness = –0.198) is more skewed than the red force (skewness = –0.072).

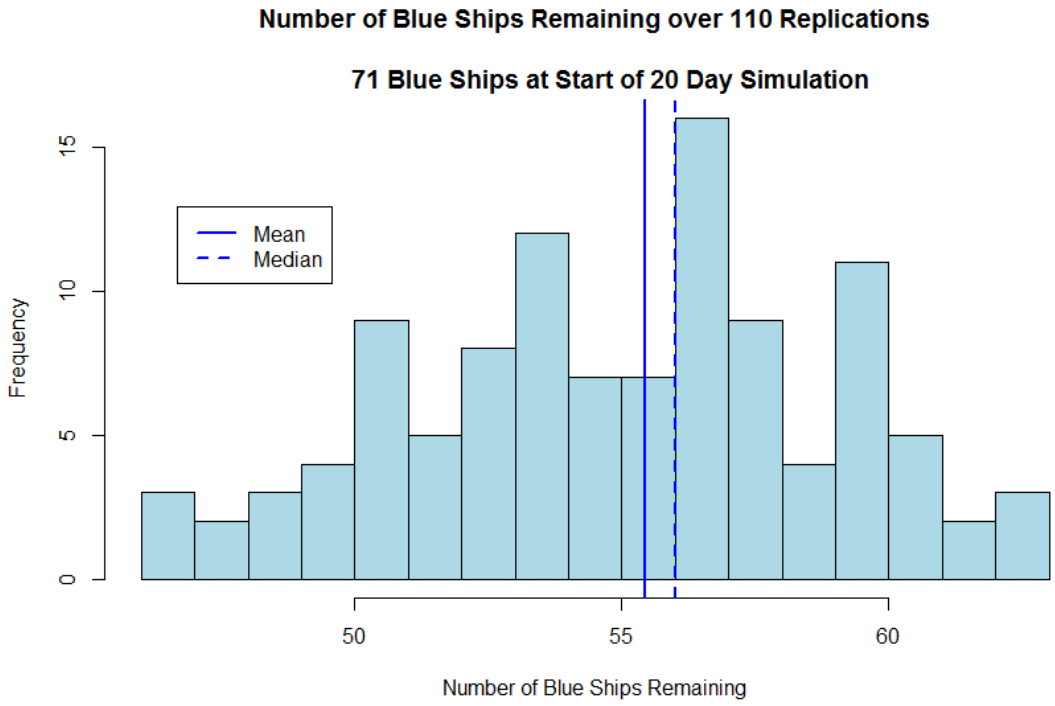

Figure 6. Blue ships remaining at simulation termination.

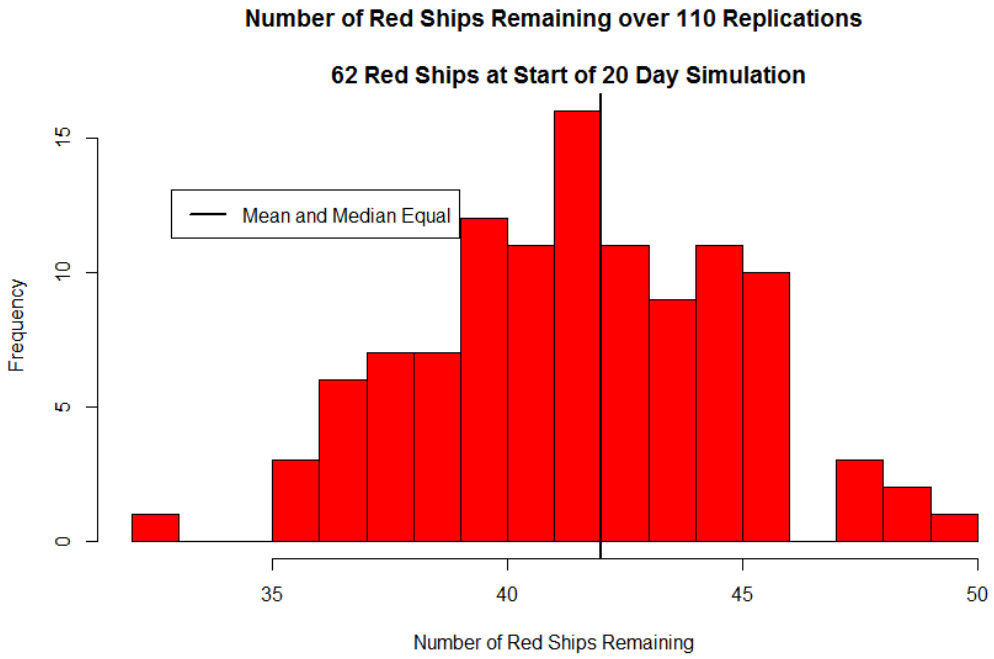

Figure 7. Red ships remaining at simulation termination.

	Blue	**Red**
Mean	55.45	42
Sample Standard Deviation	3.97	3.30
95% Confidence Interval on Mean	[54.69,56.19]	[41.36, 45.62]
Min	46	32
Max	63	50
Median	56	42
Skewness	–0.198	–0.072

Table 5. Summary statistics for blue and red ships remaining at simulation termination.

The remaining blue and red ships at simulation termination appear to be approximately normally distributed. This can be shown through a variety of analytical tools, such as histograms, QQ plots, and formal normality tests.

To explicitly show that not all data is approximately normal and that the raw data in this case is closely related to a normal distribution, Figures 8 and 9 compare the raw output of the number of remaining ships to the exponential, uniform, and normal distribution. The sample mean, minimum, and maximum of the remaining ships for the blue and red forces were used to determine the parameter(s) for the random distributions.

For the exponential distribution, a sample of size 110 was drawn with a rate of 1/sample mean. The uniform distribution draws were generated using the minimum and maximum of the raw data. The normal distribution was generated using the sample mean and variance. For both red and blue forces, the histograms of the raw data are most similar to the randomly generated normal distribution (top left and bottom right). The exponential and uniform distributions are clearly not normal (top right and bottom left).

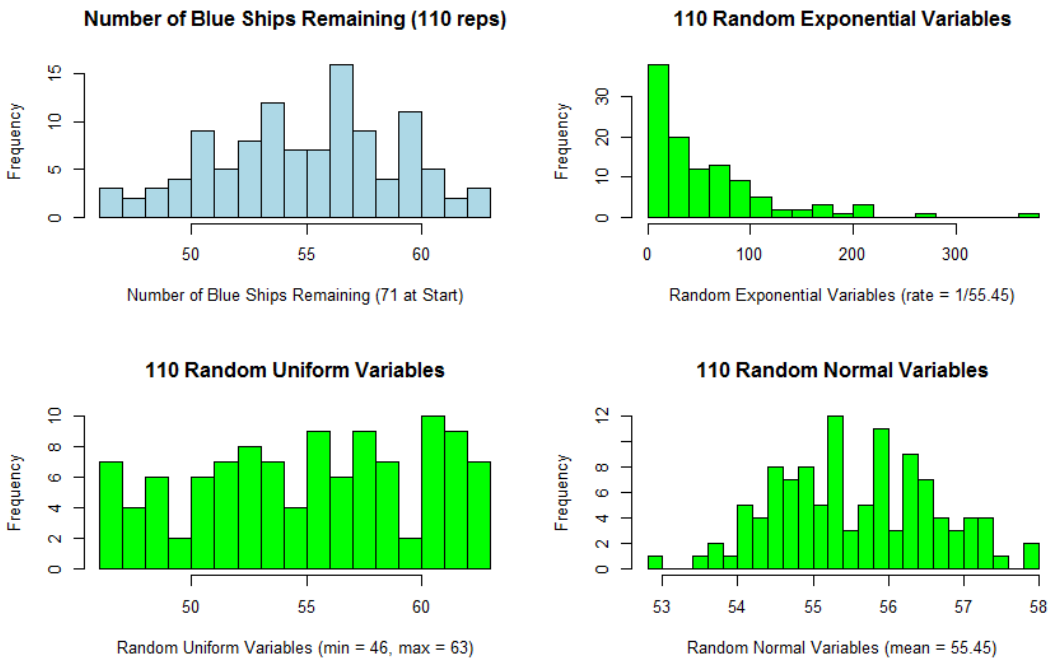

Figure 8. Comparison of "blue force remaining ships" to distributions of ships remaining via histograms.

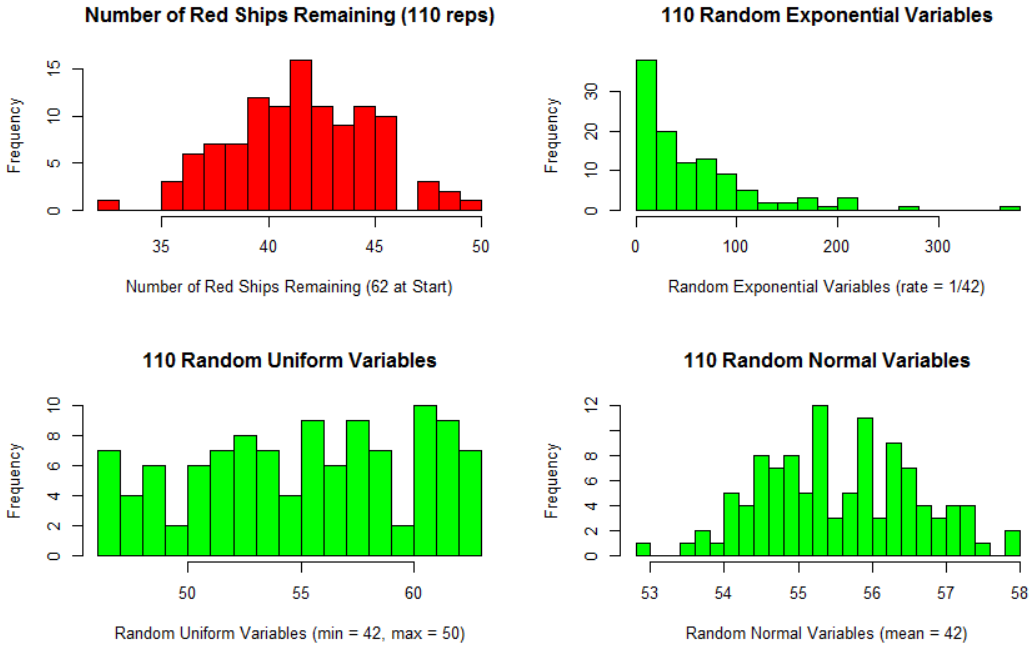

Figure 9. Comparison of "red force remaining ships" to distributions of ships remaining via histograms.

Quantile-Quantile (QQ) plots are also a good tool for checking for normality (Law, 2007). Figures 10 and 11 are QQ plots of the same data from the histograms above. QQ plots compare the two distributions. In each plot, the data is compared to the normal distribution. For a perfect fit, the plot would reveal a straight line. The QQ plots confirm that the raw data for the blue and red ships remaining is most similar to the normal distribution (top left and bottom right).

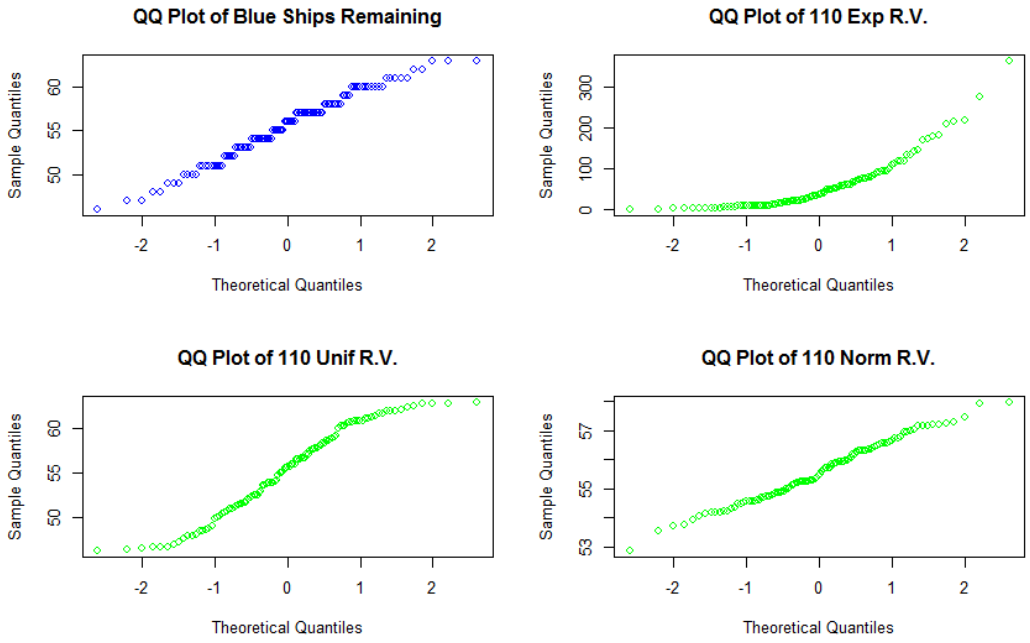

Figure 10. Comparison of "blue force remaining ships" to distributions of ships remaining via QQ plot.

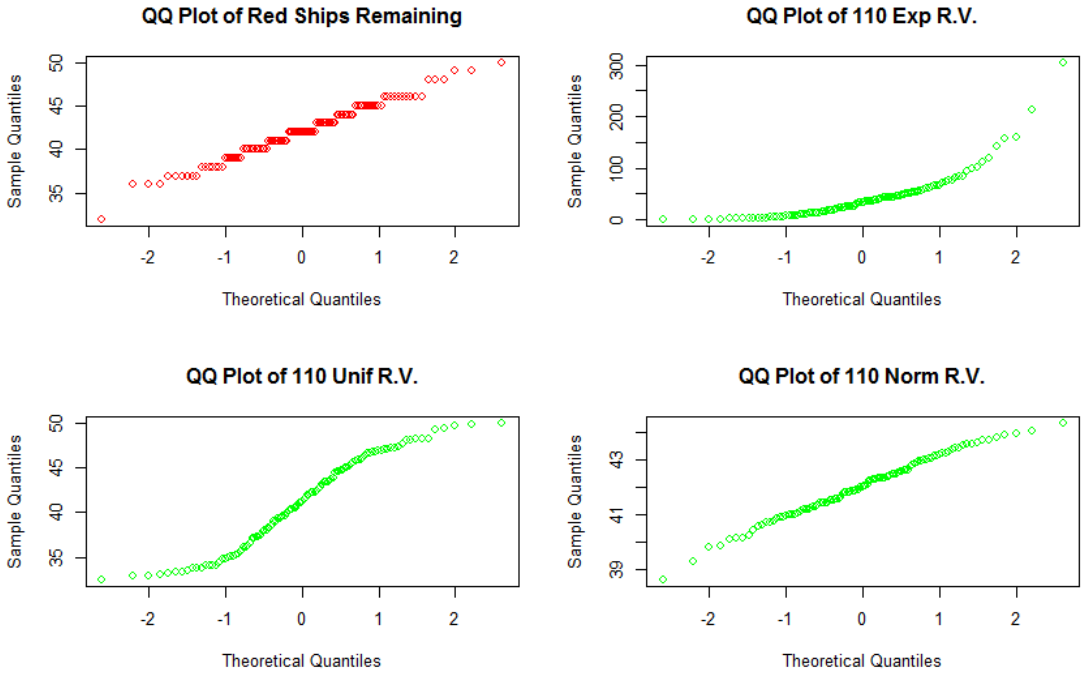

Figure 11. Comparison of "red force remaining ships" to distributions of ships remaining via QQ plot.

In terms of determining whether the raw data is approximately normally distributed in an analytic fashion, the Shapiro-Wilk, Anderson-Darling, Kolmogorov-Smirnov, and Cramer-von Mises tests were used with the following hypotheses (Currie & Cheng, 2013):

$H_0 = The\ underlying\ distribution\ is\ normal.$
$H_a = The\ underlying\ distrition\ is\ not\ normal.$

The test was conducted by taking 1,000 random draws of size 30 from the original 110 replications. A sample size of 30 was chosen since OPNAV N81 typically completes 30 replications. The formal normality tests were then applied to the 1,000 draws of sample size 30. The *p-values* from the normality tests are plotted in Figures 12 and 13 for the blue and red forces, respectively. A *p-value* is the smallest level of significance for which the observed data indicates that the null hypothesis should be rejected (Wackerly, Mendenhall III, & Scheaffer, 2008. The vast majority of the *p-values* for the blue and red data are above the cutoff of 0.05, thereby not rejecting the null hypothesis that the underlying distribution is approximately normal. In this case, for samples of size 30 drawn from the actual data, most of the time we will retain the null hypothesis; however, with a mean *p-value* of 0.2, there is evidence that the output is not quite normal. Indeed, since it is discrete, it cannot be. Most tests assuming normality, however, will be pretty accurate.

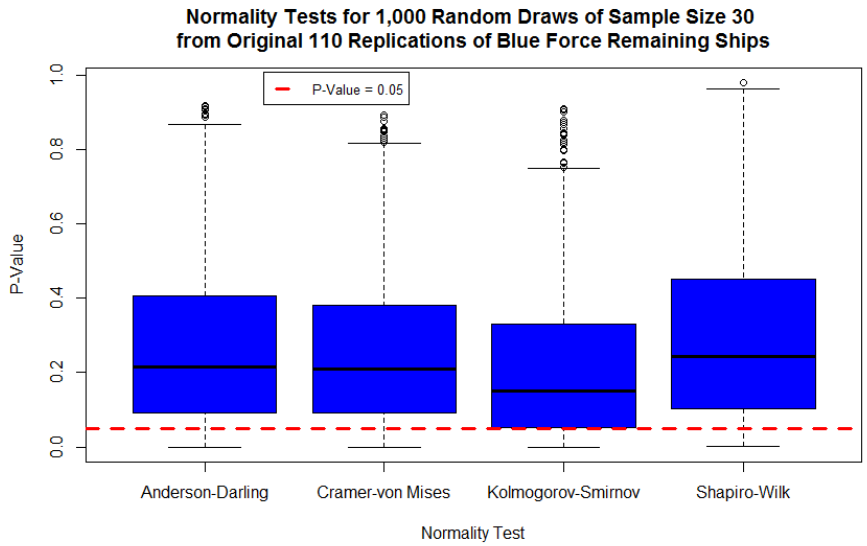

Figure 12. Box plots of normality tests for the number of "blue ships remaining" at simulation termination.

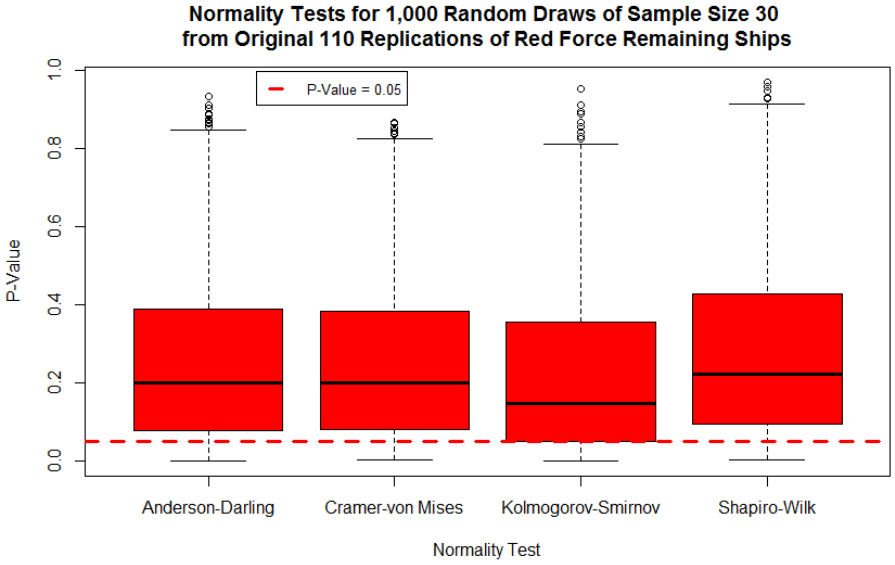

Figure 13. Box plots of normality tests for the number of "red ships remaining" at simulation termination.

Further results of the investigation of the raw data for blue and red ships remaining are displayed in Table 6. These results look at the original 110 replications and reveal the *p-values* for the four normality tests. Assuming a *p-value* cutoff of 0.05, there is a single normality test failure; specifically, the Kolmogorov-Smirnov test for blue ships

remaining. Being that only one out of the four tests failed, however, we can still assume normality with a high level of confidence that our statistical procedures will be robust to the mild nonnormality that may exist.

	Blue Ships Remaining	Red Ships Remaining
Anderson-Darling	0.060	0.13
Cramer-von Mises	0.060	0.13
Kolmogorov-Smirnov	0.003	0.16
Shapiro-Wilk	0.090	0.30

Table 6. *P-Values* from the four formal normality tests for 110 replications of the number of "red and blue ships remaining."

B. WHAT-IT-TAKES-TO-WIN METRIC

OPNAV N81 is very interested in metrics they call "what it takes to win" (WITTW). That is, they want to identify key variables and thresholds that enable blue to win a given replication. The next metric analyzed is the time at which the blue forces achieve air supremacy in the simulation. An important distinguishing characteristic of some metrics, such as achieving a goal, is the fact that the event may not take place in all of the replications. This is common in a terminating simulation like the Punic21 scenario used for this research. Even if the event does not always occur, it is still informative to analyze what takes place when the event does happen. Alternative analysis would involve determining which variables in the scenario either support or prohibit the blue forces from achieving air supremacy. This would involve investigating relationships among different variables to determine whether there is causation of one event to another through correlation and or dependency relationships.

The blue forces achieve air supremacy 82 times out of the 110 reps, which is 74.5 percent of the time. The confidence in this point estimate can be expanded upon by taking the normal approximation to the binomial distribution to develop a confidence interval by

$$\hat{p} \pm z_{0.975}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \qquad \text{(Law, 2007), (3.1)}$$

where p is the probability of success and n is the number of trials (i.e., 110). Confidence intervals constructed this way will include the true proportion approximately 95 percent of the time. The confidence interval on the estimate for the day that blue forces achieve air supremacy is [0.66, 0.83], by Equation 3.1. The remaining portion of this chapter focuses on the time at which air supremacy is achieved (using only 82 data points of 110 replications), since it was achieved at some point. This is interesting to the decision maker because timing can be everything in a military campaign. For example, if a land attack cannot take place until air supremacy is achieved, the time at which it is achieved, and the precision of that estimate, become very important.

The distribution of outcomes for the time at which blue forces achieve air supremacy, given they achieved it, can be seen in Figure 14 and Table 7. The outcomes are spread relatively evenly between the minimum of 11.75 and the maximum of 19.75 days, with the exception of the majority of events taking place around the median. Although the median and mean are relatively close, this data does not appear to be normally distributed with such a long left tail.

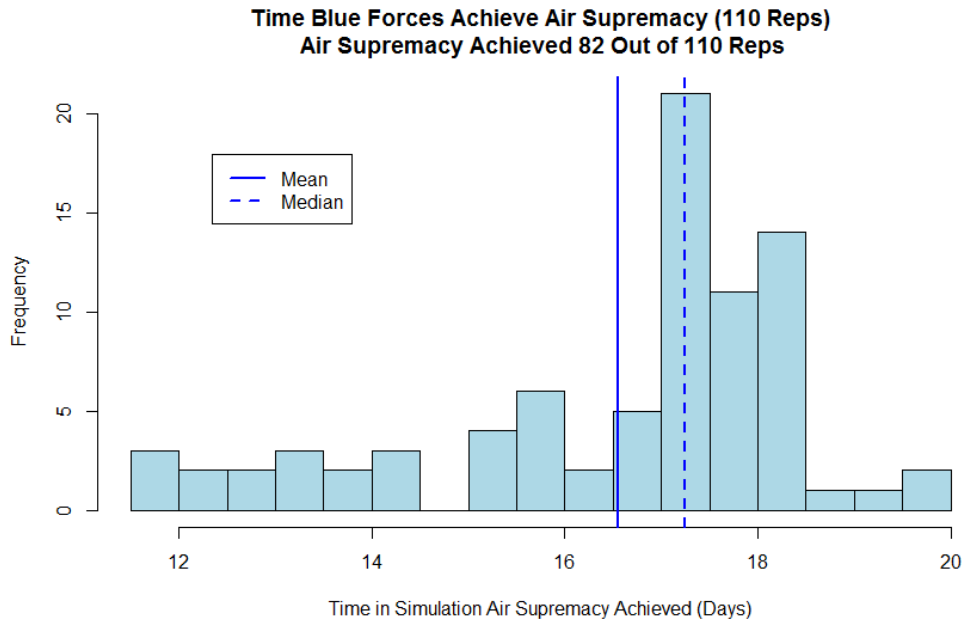

Figure 14. Time in simulation at which blue forces achieve air supremacy.

	Time At Which Blue Achieves Air Supremacy
Mean	16.55
Sample Standard Deviation	1.98
95% Confidence Interval	[16.12,16.99]
Min	11.75
Max	19.75
Median	17.25
Skewness	–1.00

Table 7. Summary statistics on the time at which air supremacy (blue) is achieved.

Although the numbers of remaining ships for blue and red forces are approximately normally distributed, the case is made in this section that the time at which blue achieves air supremacy does not reflect an approximately normal distribution. We can still use the methods in Chapter IV to estimate the appropriate number of replications. The advantage of normally distributed data is that it exhibits predictability and probability, which results in easier computations in performing analysis, which many statistical tests assume.

To visually represent how the air supremacy data is not normally distributed, Figure 15 represents the raw data compared to the exponential, uniform, and normal distributions, with parameters (mean, min, max) calibrated for the raw data. The raw data in the top left histogram does not appear to be normally distributed, especially when compared to the normally distributed data with the same mean on the bottom right.

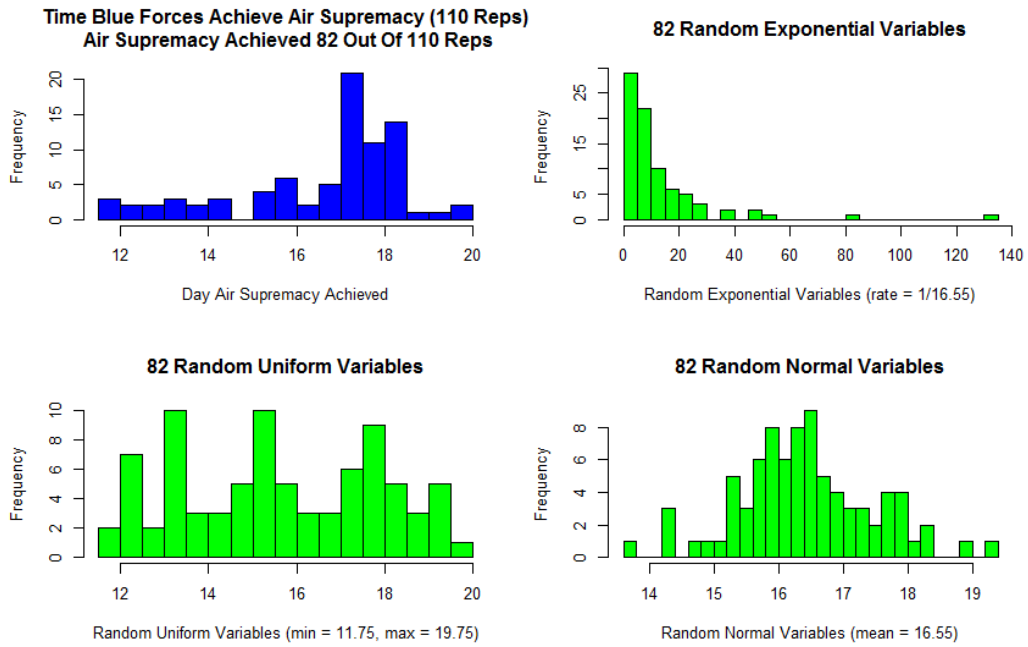

Figure 15. Histogram comparison of raw air supremacy data versus the exponential, uniform, and normal distributions, with parameters derived from the raw data.

In addition to the histograms, QQ plots are good methods for comparing distributions. Since we are looking at whether the data is normally distributed, each data set from the histograms above is compared to a theoretical normal distribution. Figure 16 reveals that the only distribution that approximately matches the normal distribution is the bottom right plot, which was randomly created with normally distributed data. The top left plot is the raw air supremacy data, which reveals nonnormality.

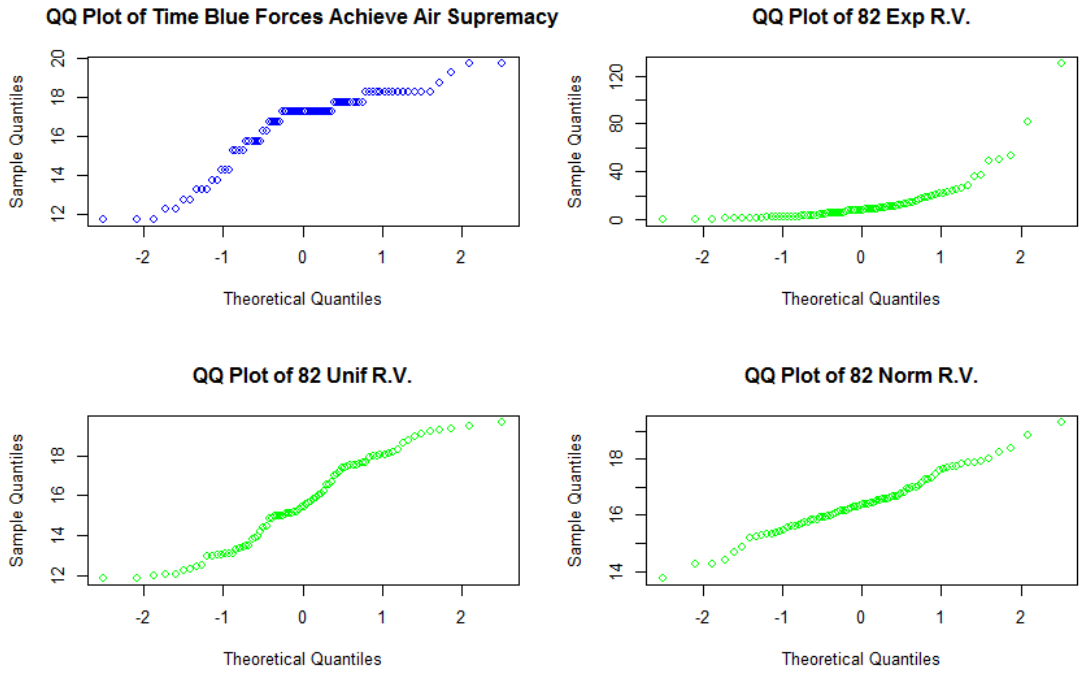

Figure 16. QQ plot comparison of raw air supremacy data versus the exponential, uniform, and normal distributions, with parameters derived from the raw data.

To analytically show that the air supremacy data is not normally distributed, the same hypothesis testing that was conducted on the ship remaining data. Figure 17 reveals that a large portion of the data is below the *p-value* of 0.05, which causes us to reject the null hypothesis of the underlying distribution being normally distributed.

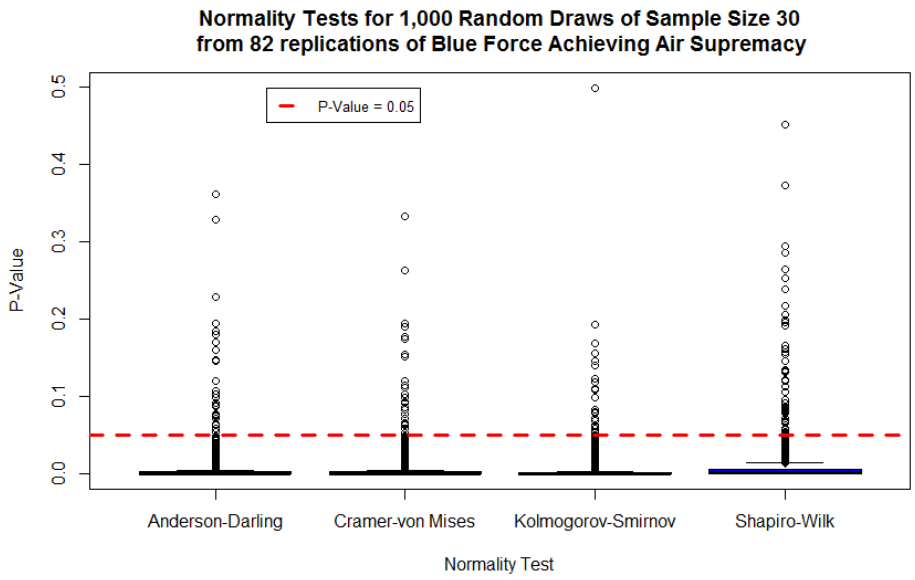

Figure 17. Normality testing of the air supremacy data.

Formal normality tests on the original 82 replications of raw data reveal that the data is not normal. The results can be seen in Table 8.

	Day in Which Blue Achieves Air Supremacy
Anderson-Darling	8.2e-11
Cramer-von Mises	1.27e-08
Kolmogorov-Smirnov	1.02e-13
Shapiro-Wilk	7.44e-07

Table 8. *P-Values* from the four formal normality tests for 82 replications of the day in which blue achieves air supremacy.

C. HIGH-VARIANCE METRIC

The final metric selected to analyze is the number of missions flown by blue force multirole fighter planes. This metric was selected due to its relatively larger variance when compared to the other metrics. It can also be categorized as a performance measure in that it reveals the frequency of an event in each replication. A metric such as this could be useful to an analyst and, ultimately, to a decision maker in a simple scenario. The number of missions that an aircraft can fly can be constrained by many things, such as the

sortie rate, which is the rate at which a force can deploy aircraft. Knowing the distribution of outcomes for the number of missions flown for a particular aircraft could help estimate the sortie rate required to meet the demand. Higher sortie rates have monetary costs, such as the need for an additional aircraft carrier or a more capable air base. As a result, looking at the statistics for the number of missions flown could help plan and save resources.

The distribution of outcomes for the number of missions flown by future multirole fighters can be seen in Figure 18 and Table 9. From a qualitative perspective, the outcomes appear to be distributed fairly well and seem to represent an approximately normal distribution.

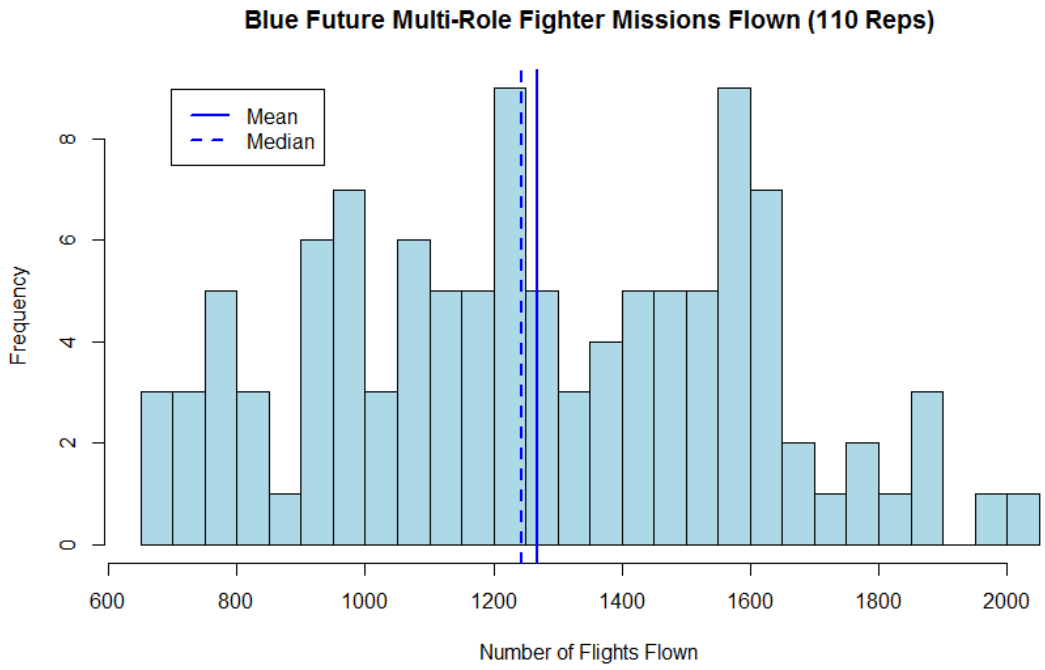

Figure 18. Number of missions flown by blue future multirole fighters (110 Reps).

Summary Statistic	Missions Flown by Blue Future Multirole Fighters
Mean	1,267.09
Sample Standard Deviation	332.73
95% Confidence Interval on Mean	[1204.21,1329.97]
Min	652
Max	2,024
Median	1,243.5
Skewness	0.06

Table 9. Summary statistics for the number of missions flown by blue future multirole fighters.

So far we have seen two metrics, the number of blue and red ships remaining, which were approximately normal, and the time in which blue forces achieve air supremacy, which was not approximately normal. The number of missions flown for blue future multirole fighters appears to be normally distributed, although it has a much larger standard deviation.

A visual representation of the actual outcomes to random variables from the exponential, uniform, and normal distributions can be seen in Figure 19. These distributions were derived from parameters developed from the raw data. Although the top left histogram does not pass the visual test for normality with confidence compared to the bottom right, which is a normal distribution, further testing reveals that it is approximately normal.

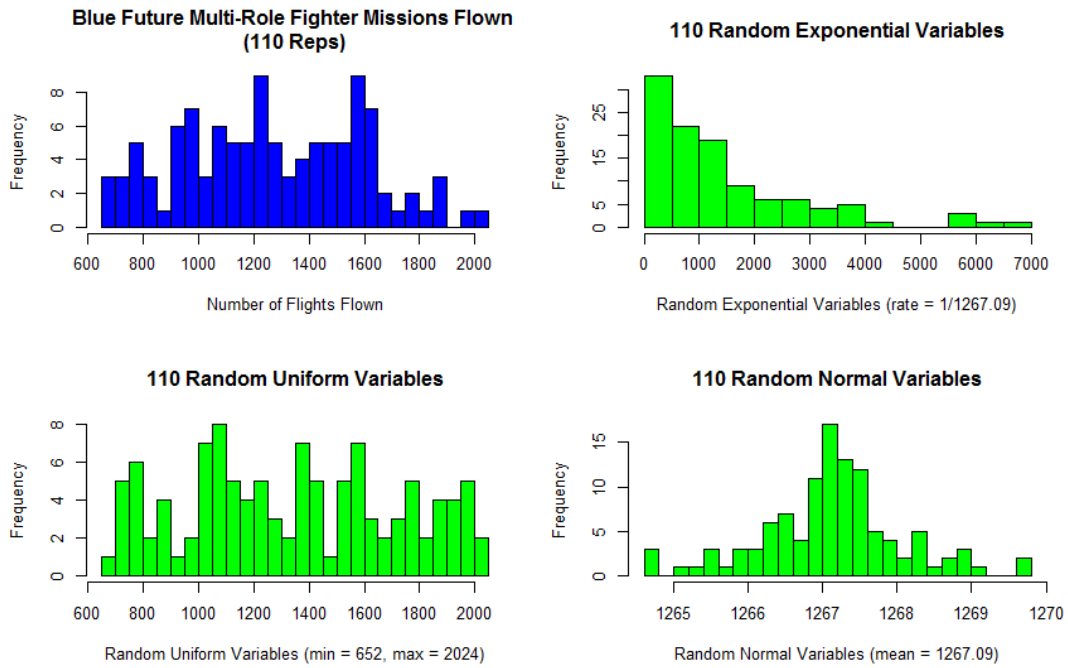

Figure 19. Histogram comparison of raw blue future multirole fighter missions flown compared to the exponential, uniform, and normal distributions.

The QQ plot in the top left of Figure 20 builds confidence that the raw data is approximately normally distributed because it follows an approximately straight line, which represents the theoretical values for the normal distribution.

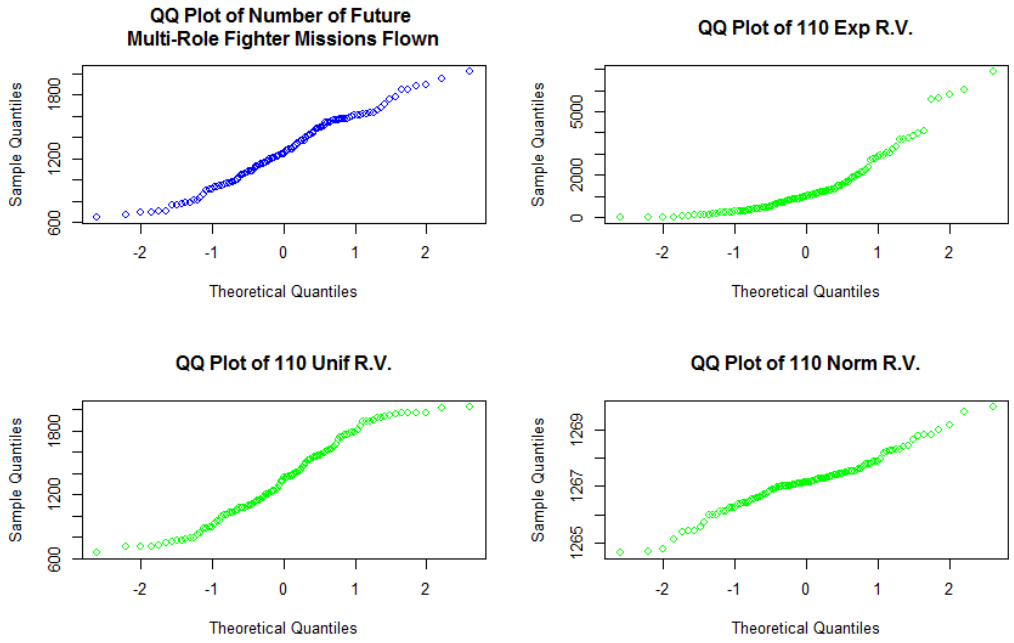

Figure 20. QQ plot comparison of raw blue future multirole fighter missions flown versus the exponential, uniform, and normal distributions.

The same hypothesis from the previously analyzed data in this chapter applies here. The null hypothesis is that the distribution of outcomes is approximately normal. Referring to the box plots in Figure 21, we see that in almost all cases the data passes the stated normality tests. As a result, we can assume that the data in this case is distributed approximately normally.

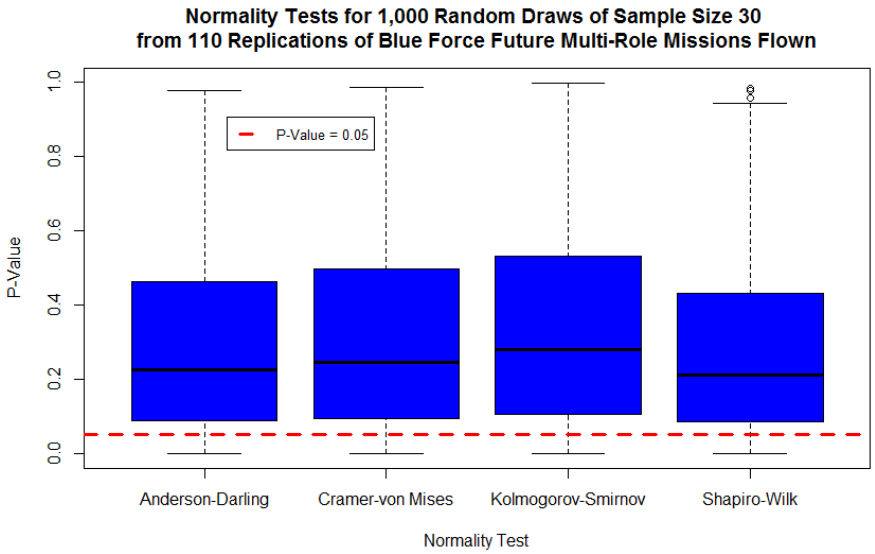

Figure 21. Normality testing of the number of blue future multirole fighter missions flown.

Further investigation of the raw data for the number of blue force multirole fighter missions flown can be seen in Table 10. All the *p-values* from the four normality tests on the raw data are greater than 0.05; therefore, this data can be considered roughly normally distributed for statistical procedures.

	Number of Blue Future Multirole Fighter Missions Flown
Anderson-Darling	0.104
Cramer-von Mises	0.132
Kolmogorov-Smirnov	0.121
Shapiro-Wilk	0.074

Table 10. *P-Values* from the four formal normality tests for 110 replications of the number of blue future multirole fighter missions flown.

D. STEPS TO PROCESSING SIMULATION OUTPUT DATA

This section is designed to be an aid to the analyst repeating the same steps that this chapter presents to gain insight into the distribution of outcomes and normality testing for any metric, for any campaign simulation run in STORM. The R code to accomplish the steps can be found in the Appendix A.

Selecting the metrics that require postprocessing is user defined and will vary, depending on the scenario and study objectives. Analysts may have many reasons to dive deeper into particular metrics. For example, they may know the historical impact of specific metrics from previous scenarios or need to be confident in the time in the campaign in which an event will take place.

An initial look at the summary statistics and histograms provides the analyst with a high-level view of the data. This information can automatically be generated using the R code in Appendix A. It is insightful to compare the data to other distributions, such as the exponential, uniform, and normal distributions via histograms and QQ plots. Finally, formal normality tests can be conducted to determine whether the data is normally distributed. The significance of having normally distributed data is that it enables the analyst to do many statistical tests with confidence and allows for better stopping rule and confidence interval results. The formal normality tests used in the R code are Anderson-Darling, Kolmogorov-Smirnoff, Cramer-von Mises, and Shapiro-Wilks. These normality tests are considered to be the most powerful, with the Shapiro-Wilks tests performing exceptionally well, even when there are a small number of data points (Currie, 2013). As a result, it is recommended for small sample-sized testing to utilize the Shapiro-Wilks normality test.

It is also important to understand how many observations must be in a data set to determine normality. This is another reason for looking at the histograms and QQ plots, in addition to the formal normality tests. To demonstrate this, 1,000 normal, exponential, uniform, and gamma random variables were generated. The Shapiro-Wilks test was conducted on the random draws from the 1,000 variables in sample sizes 10, 30, 60, and 100. A sample size of 10 usually causes a false positive by retaining the gamma, exponential, and uniform distributions as normal. As the sample size increases to 30, the exponential and uniform test results begin to fail the Shapiro-Wilks test. In this example, it takes a sample size of 100 for the Shapiro-Wilks test to reveal that the underlying distribution is not normal. Figure 22 depicts these results. These results are important to understand because there is risk in small sample-sized, normality tests.

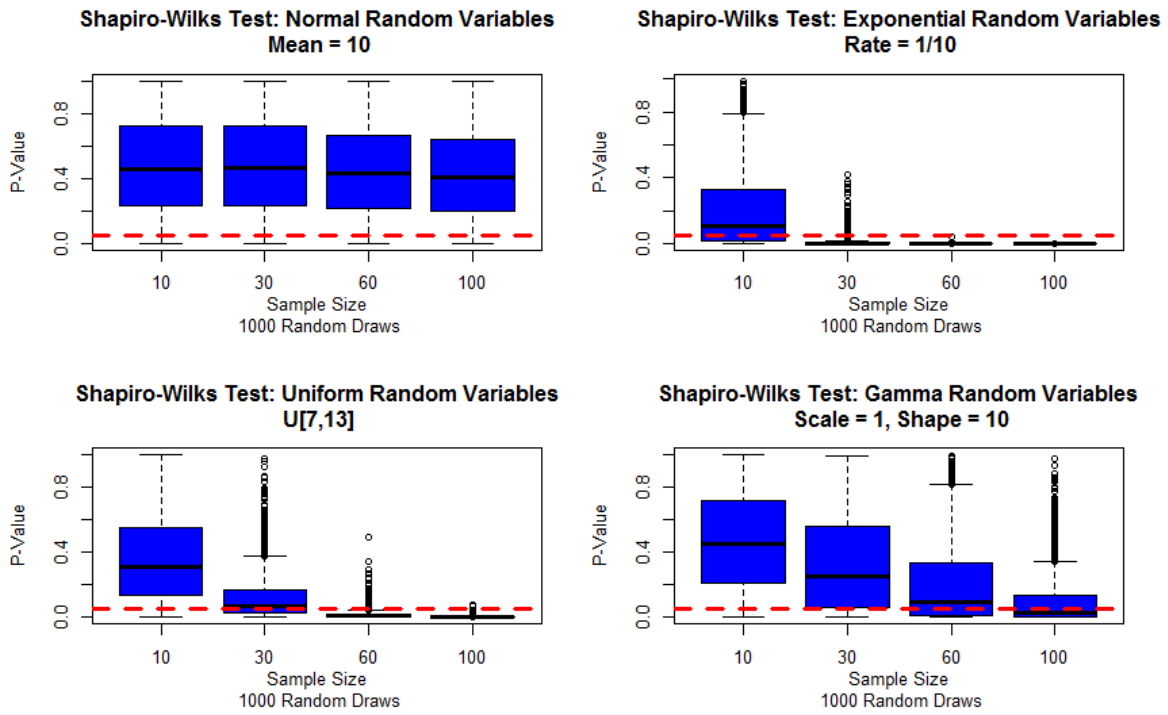

Figure 22. Formal normality testing of 1,000 random draws of sample sizes 10, 30, 60, and 100 from the normal, exponential, uniform, and gamma distributions.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. STOPPING RULES ANALYZED FOR STORM OUTPUT DATA

This chapter focuses on analyzing the trade-off between the expected number of replications and desired performance measures, such as precision and confidence. The unclassified scenario Punic21 is used as a baseline test to discuss the validity of the simulation results utilizing previously established simulation stopping rules. The goal is to provide OPNAV N81 with insight and tools by describing a methodology to analyze the trade-off between the appropriate number of replications and the precision and confidence for a given scenario.

The same metrics from Chapter III are tested, using stopping rules to help in understanding the relationship between variability and the normality assumption versus the expected number of replications required. The number of remaining ships for blue and red forces at simulation termination is examined first, followed by the day in which blue forces obtain air supremacy. The final metric analyzed is the high-variance metric, the number of missions flown by blue future multirole fighters. A high variance metric was selected because they tend to be the most sensitive to the choice of stopping rule parameters.

This chapter is meant to demonstrate the process in determining the appropriate number of replications. A script was generated in R and appears in Appendix B to aid future users of STORM in applying this methodology.

A. CONFIDENCE INTERVALS

A confidence interval is designed to help the analyst and/or decision maker have an understanding of how precise an estimate is for an unknown population parameter. An easy way to define a confidence interval is to consider constructing a very large number of independent $\eta = 1 - \alpha$ percent confidence intervals; the proportion of the confidence intervals that contain the unknown parameter would be $1 - \alpha$, where α is the probability that the interval does not include the true population parameter (Law, 2007). The existing study tool in STORM does not output confidence intervals or variance. The output in STORM is simply given as a mean over the replications. The problem with a sample

mean is that its variance (related to risk) is not considered. For example, consider two stock portfolios, A and B, that are available as investments. Portfolio A has an average return of 10 percent and a variance of 3 percent. Portfolio B has an average return of 11 percent and a variance of 20 percent. Although portfolio B has a slightly higher average return, knowing it is almost six times as risky in terms of variance may persuade the decision maker to choose portfolio A. Confidence intervals, which provide an estimate of the variability around the mean are defined next with examples from the Punic21 data set. It is important to understand confidence intervals prior to delving into stopping rules, since confidence interval procedures are often used with stopping rules.

Each output metric STORM provides for a given variable of interest is different; therefore, it would be difficult to prove that the underlying distribution is normal for every metric. As a result, we must assume the central limit theorem applies and after a sufficiently large number of replications; the distribution of the mean becomes approximately normal (Law, 2007). In addition, most of the time we do not know the population variance (σ^2), therefore, we estimate σ^2 by S^2, the sample variance, as the sample size (n) gets large.

Let $X_1, X_2, ..., X_n$ be independent, identically distributed random variables, with a finite mean and variance and, with the assumptions above, we can form a confidence interval. Equation 4.1 defines the two─sided confidence interval assuming a normal distribution by

$$\bar{X}(k) \pm z_{1-\alpha/2}\sqrt{\frac{S^2(k)}{k}} \qquad \text{(Law, 2007), (4.1)}$$

where k is the number of replications, $S^2(k)$ is the sample variance of the first k samples, and $z_{1-\alpha/2}$ is the z-score (or $(1-\alpha/2)*100$ *percentile*) from the normal distribution.

An even more conservative approach to confidence interval formulation is using the t-distribution rather than the normal distribution. This is because $t_{k-1,1-\alpha/2} > z_{1-\alpha/2}$, for small values of alpha, which is a direct reflection of the t-distribution's lower peak and larger tails (Law, 2007). Equation 4.2 defines the confidence interval using the t-distribution by

$$\bar{X}(\mathrm{k}) \pm t_{k-1,1-\alpha/2}\sqrt{\frac{S^2(\mathrm{k})}{k}}\,. \qquad \text{(Law, 2007). (4.2)}$$

The only difference between Equations 4.1 and 4.2 is that the *t*-distribution is used.

The last part of introducing confidence intervals is the realization of the half width. The half width is the value that we subtract or add from $\bar{X}$, to form a confidence interval, and is a measure of precision. Decision makers like precise results (i.e., narrow intervals). The width of a confidence interval depends on the population variance, our desired coverage probability, and the number of replications. The smaller the half width, the more precise the results are. The half width is written as

$$HW_{\eta,k} = t_{\eta,k-1}\sqrt{\frac{S^2(k)}{k}} \qquad \text{(Singham, 2012), (4.3)}$$

where η is the desired confidence and *k* is the number of replications.

In examining the Punic21 data for STORM, and utilizing Equations 4.2 and 4.3, Table 11 represents raw data transformed into half-widths and their associated confidence intervals.

	Mean	**Variance**	**Half-Width**	**90% Confidence Interval**
Blue Ships Remaining	55.45	15.48	0.62	[54.82, 56.07]
Red Ships Remaining	42	10.72	0.52	[41.48, 42.52]
Day Blue Achieved Air Supremacy	16.55	3.82	0.36	[16.20, 16.91]
Number of Missions Flown by Blue Future Multirole Fighters	1267.09	108706.30	52.15	[1214.94, 1319.24]

Table 11. Summary statistics for metrics. Blue, red, and multirole fighters are from 110 replications. Day Blue Achieved Air Supremacy is from 82 replications. Air supremacy data is not normal; therefore, summary statistics are for comparison only.

B. BACKGROUND ON STOPPING RULE

This research utilizes the methodology of determining a desired stopping condition based on the research of Singham (Singham, 2010). The research within her dissertation provides a framework to gain nominal coverage with a minimal expected number of replications. As a result, we will be able to quantify the confidence in the mean estimate for output data for a given number of STORM replications and a given precision level using a sequential stopping rule.

Normally, stopping rules fall into two categories: fixed and sequential. Both fixed and sequential stopping rules are used in confidence interval procedures to generate confidence intervals. Fixed rules are simple; the user determines the number of replications to run and executes the simulation. This is the method that OPNAV N81 currently uses on STORM, typically running between 30 and 50 replications due to data storage requirements and the need to provide quick, turn-around analysis. For sequential stopping methods, a baseline number of replications is completed and testing is conducted to determine whether the desired precision is obtained. If the test fails, more replications are completed one by one, or in batches, until the stopping rule criterion is met.

1. Summary of Stopping Rules

To compare how good or bad a given number of replications is in terms of its respective half-width, we compare it to a parameter delta (δ). Delta is defined as a desired level of precision. For a very precise solution, a small value for delta would be required. Therefore, our goal is to complete the minimum number of replications that ensures our half-width is less than δ, as the following inequality suggests:

$$k^* = \arg\min_{k \geq 2} HW_{\eta,k} \leq \delta, \tag{4.4}$$

where k^* is the minimum number of replications meeting the inequality.

Intuitively, it can be seen that as δ gets smaller and smaller, the expected number of replications required would go to infinity. Now, we must complete testing at different

levels of η and δ. In this design, δ will be a function of the sample standard deviation. To get δ smaller, representing more precision, the sample standard deviation is be divided by a factor resulting in the following:

$$\delta_i = \frac{S^2(k)}{i}, index\ delta\ by\ i \in \{1,2,4,6,8,10,15\}. \quad (4.5)$$

The minimum number of replications for any experiment of this type is two, based on the premise that you must have two observations to calculate the sample variance, which is an input to determining the half-width. As a result, another factor, *Kstart* is used. *Kstart* is defined as the minimum number of replications observed prior to calculating the half-width. *Kstart* will be varied at levels of {2, 5, 10, 20, 30, 40, 50}. *Kstart* turns out to be relevant if you imagine the following set of output: {1,1,1,1,1,0,0,1}. If *Kstart* was equal to two, then the sample variance is zero until *Kstart* is greater than five. Once *Kstart* is equal to six, variance would be non-zero for the first time. This implies there is a danger associated with conducting a small number of replications because values that may be less probable may not yet be observed. The example presented is a binary case, but holds true for a nonbinary case in the same manner.

C. METHODOLOGY TO IDENTIFY THE RELATIONSHIP BETWEEN THE EXPECTED NUMBER OF REPLICATIONS, PROBABILITY OF COVERAGE, AND PRECISION

This analysis is intended to outline the process and provide OPNAV N81 with a structured tool that can be applied to any metric for which they want to concretely define the confidence associated with the number of replications completed. The data used in this chapter is output from 110 replications from the Punic21 scenario in STORM.

The R-Script employed reads in a metric file and calculates the summary statistics used for half-width calculations. The user defines the desired level of confidence and the values of δ as inputs to the script. A half-width is calculated with *Kstart* number of replications. The data is resampled until the half-width is less than or equal to δ, as the inequality in Equation 4.4 suggests. Solving the inequality is completed 10,000 times for each level of δ. The result allows the analysts to see the trade-off between the average

number of replications, the probability of coverage, and the precision. Three plots are generated, consisting of lines extrapolated though points: the expected number of replications versus δ, δ versus the probability of coverage, and the expected number of replications versus the probability of coverage. From the graphs, the analyst can determine how many replications (approximately) should be completed for each δ and the associated probability of coverage. In addition, a file is generated with the data that the script calculates for any follow-on analysis that the automatically generated plots do not reveal.

Figure 23 summarizes the methodology of the script used to determine the recommended number of replications, based on desired precision and probability of coverage.

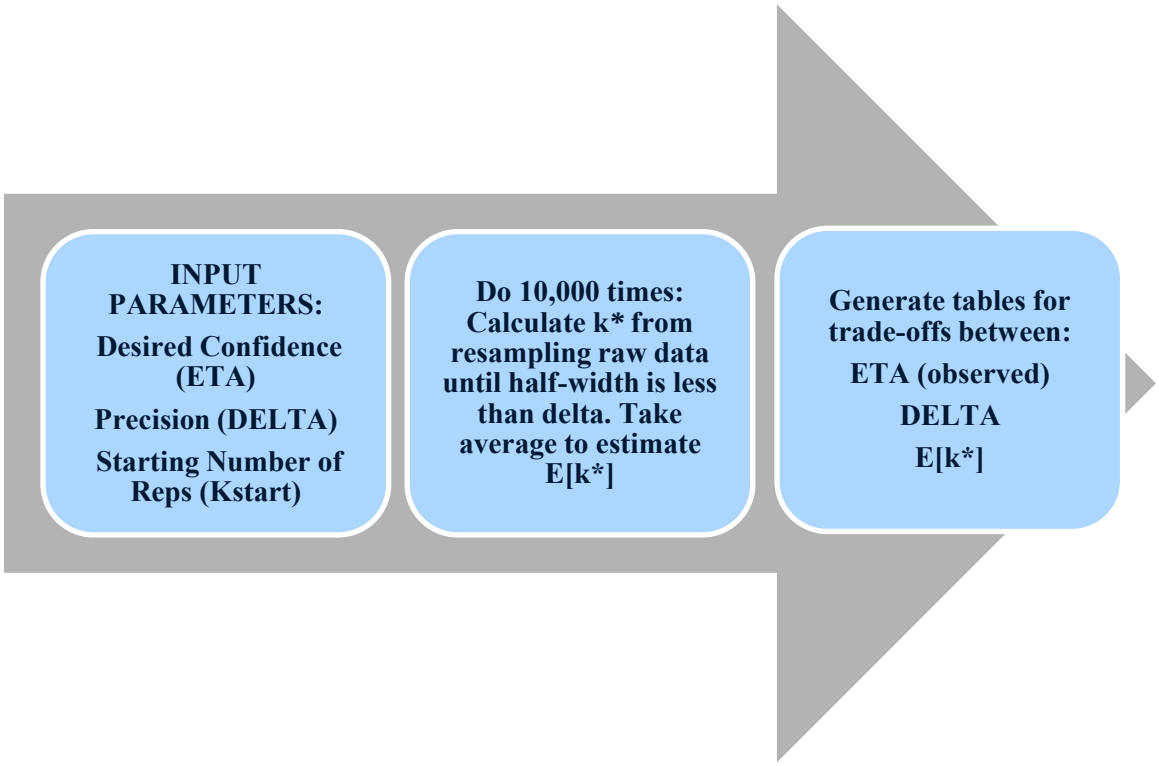

Figure 23. Flow chart of the process in which the R-Script determines the trade-off between the expected number of replications, precision, and the probability of coverage.

D. RELATIONSHIP BETWEEN DELTA AND THE EXPECTED NUMBER OF REPLICATIONS

The parameters for satisfying Equation 4.4 are the δ chosen and the desired probability of coverage. As δ gets smaller, or the desired probability of coverage gets higher, the number of replications required increases. The experiment defines δ as a factor of the standard deviation in order to provide robust analysis at multiple levels of precision. In addition, it is hard to choose an absolute precision level for δ, without knowledge of the scale of the data. In a real-world campaign simulation, analysts and decision makers would ultimately decide on the appropriate δ. This comes down to the desired level of precision. The importance of an event occurring by a certain time can be related to this. For example, if an event needed to take place by day 17 of a campaign, the precision would also be important. There is a noticeable difference in reporting a 95 percent confidence interval for the time of occurrence as [15, 19] versus [16.95, 17.05]. This smaller precision in the second interval gives the decision maker much more information in the average result.

The four metrics chosen from the Punic21 simulation selected are the number of blue and red ships remaining, the day in which blue achieved air supremacy, and the number of blue future multirole missions flown. The relationship between the expected number of replications and the delta of these metrics can be seen in Figures 24-27. Figures 24 and 25 represent the number of blue and red ships (respectively) remaining at simulation termination. The figures are similar in showing the relationship between the expected number of replications and the level of precision. For example, if the desired level of precision was 0.5, both graphs reveal that the expected number of replications would be approximately equal to 250. Figure 26, which uses a time-based metric, reveals that a lower number of replications would be required for a precision value of 0.5. In fact, this level of precision could be achieved with approximately 60-70 replications. Recall that in the final metric, the number of blue multirole fighter missions flown had a standard deviation of 332. Therefore, the scale on the precision (δ) axis is much larger, as seen in Figure 27. Although, the scale of this figure is much different than the other three, the same approximately exponential curve can be seen reflecting the trade-off

between the expected number of replications and the precision. For example, Figure 27 reveals that if you wanted a δ of 50, one would need to complete approximately 180 replications. In all of these figures, there is a sharp "knee in the curve" in which increased precision requires dramatically more expected runs. As the summary statistics revealed, the number of missions flown by the blue multirole fighter had a large variance of over 108,000, whereas the day that blue achieved air supremacy had a variance of only four. A final observation is the *Kstart* value in this relationship. Although it appears *Kstart* does not have an effect on the relationship, the next section reveals its effect.

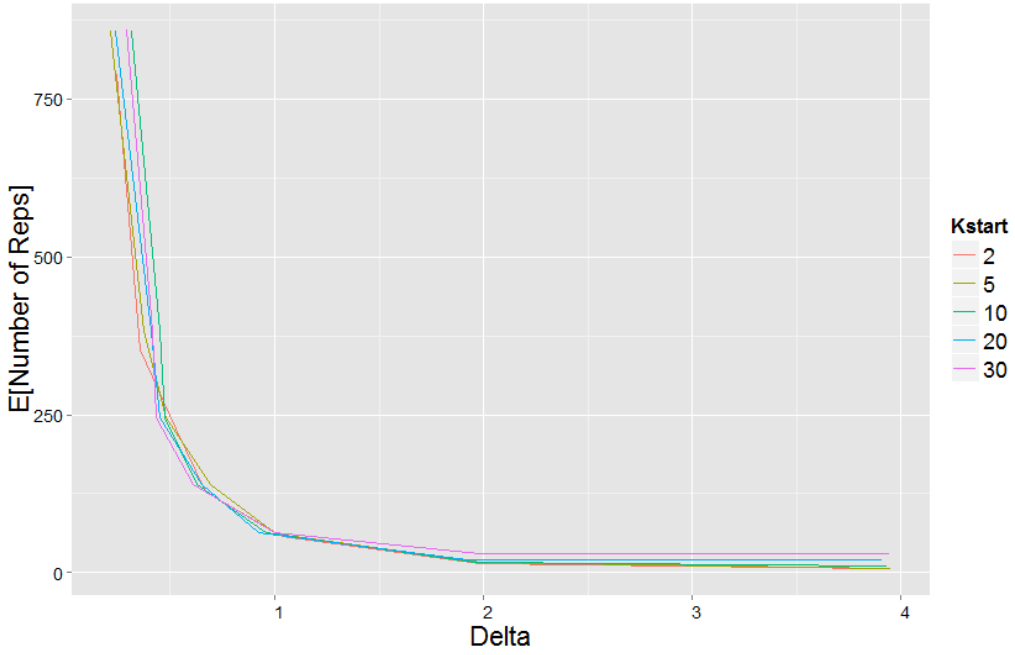

Figure 24. Expected number of replications versus delta for blue ships remaining. Desired confidence is equal to 95%.

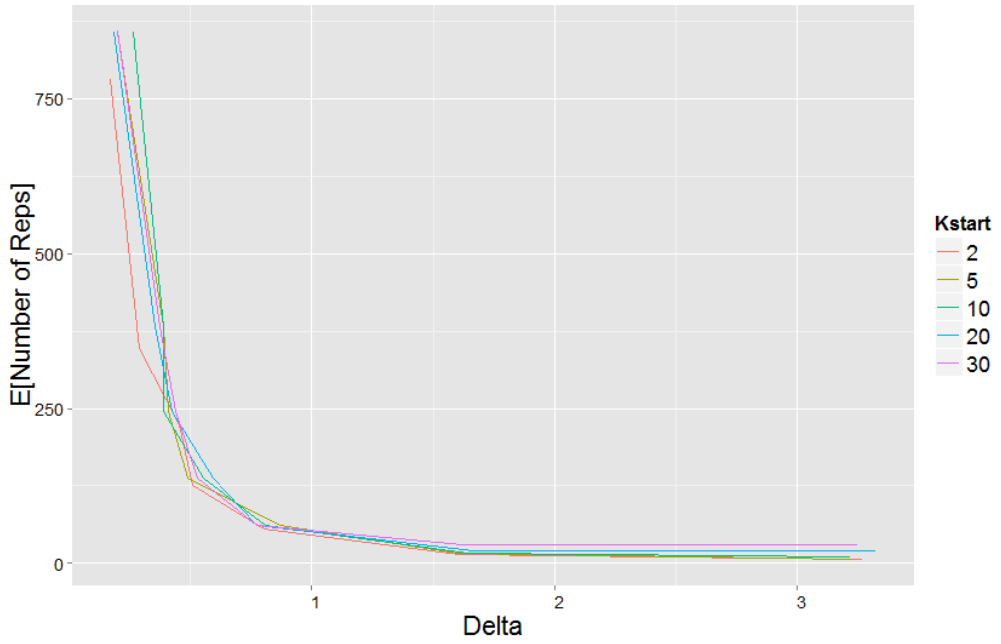

Figure 25. Expected number of replications versus delta for red ships remaining. Desired confidence is equal to 95%.

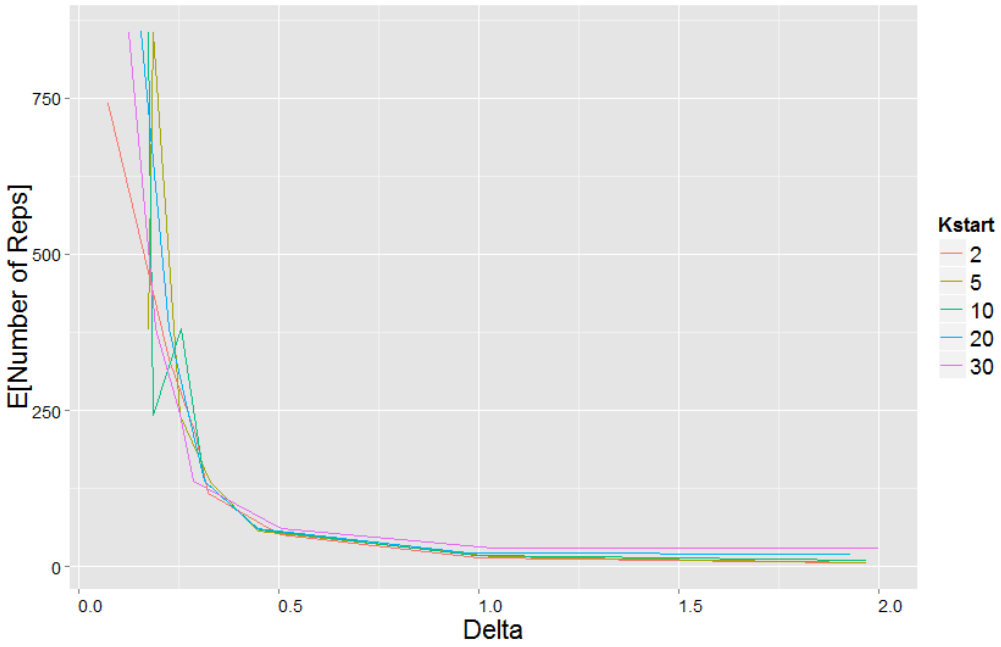

Figure 26. Expected number of replications versus delta for the number of days for blue to achieve air supremacy. Desired confidence is equal to 95%.

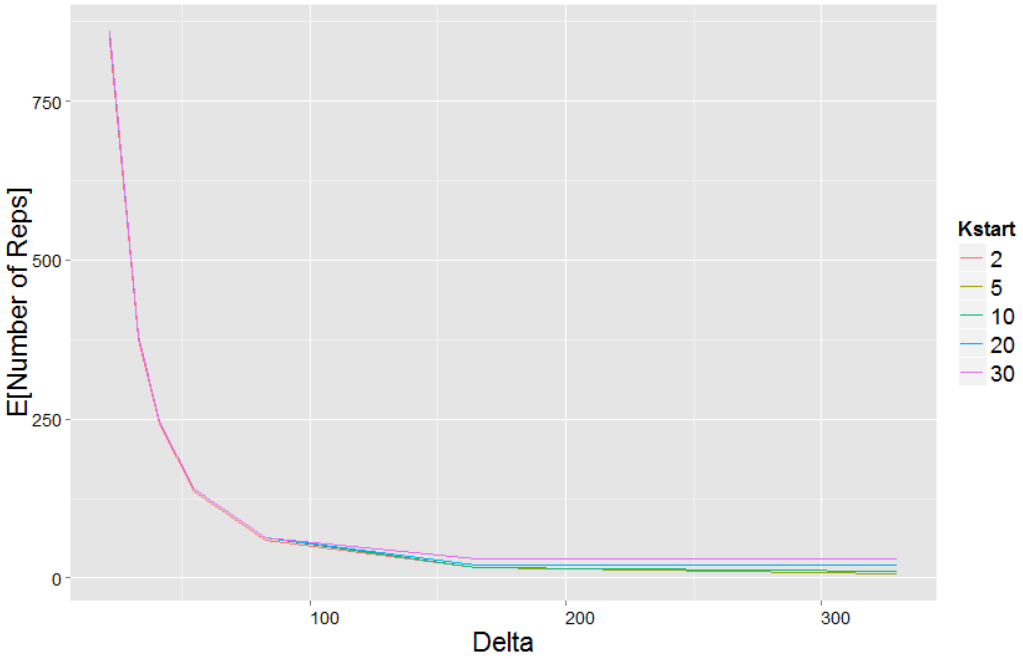

Figure 27. Expected number of replications versus delta for the number of multirole fighter missions flown for blue forces. Desired confidence is equal to 95%.

E. PROBABILITY OF COVERAGE VERSUS DELTA IN STORM

An analyst or decision maker typically desires a level of confidence in the area of 90 to 95 percent. The relationship between δ and the probability of coverage for all metrics can be seen in Figures 28 through 31. Recall that we assumed the desired coverage in this experiment was 95 percent. The figures reveal that for small δ values, the probability of coverage (approximately) meets our desired coverage of 95 percent for all levels of *Kstart*, with the exception of *Kstart* = 2. Next, the probability of coverage decreases as δ increases until a break point at which coverage exceeds the desired 95 percent. Considering the fact that small δ requires a large number of replications and a large δ requires fewer replications, the figures portray the trade-off. In many cases, experiments with a large δ result in intervals that are not meaningful; therefore, it is preferred to have a sufficiently small δ.

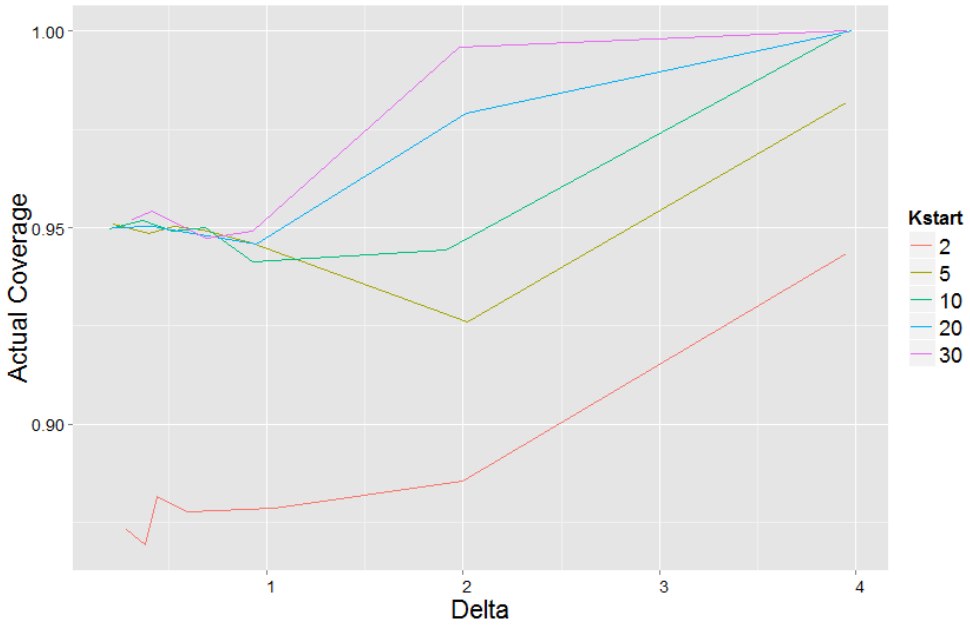

Figure 28. Delta versus probability of coverage for blue ships remaining. Desired confidence is equal to 95%.

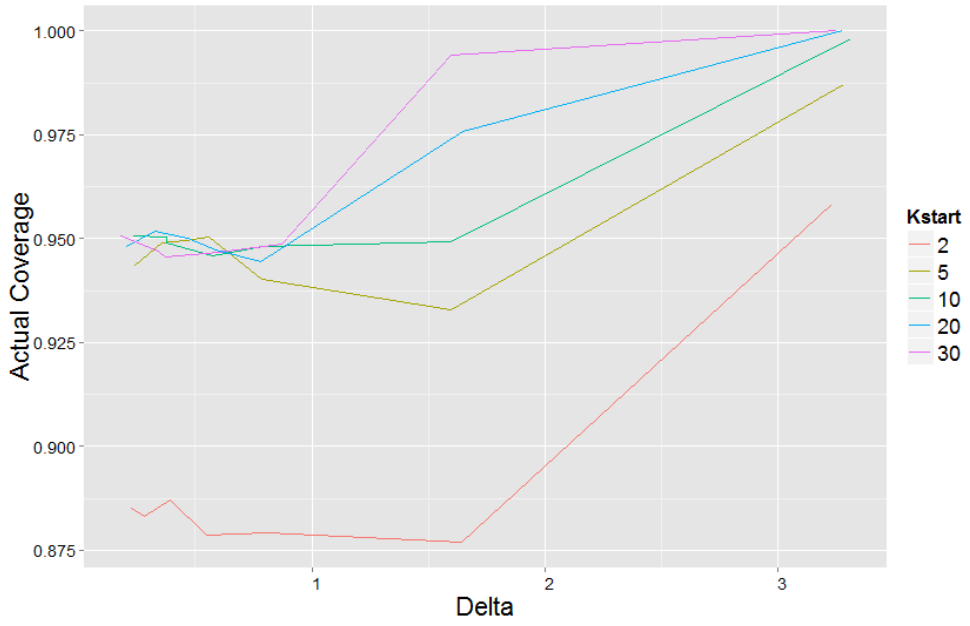

Figure 29. Delta versus probability of coverage for red ships remaining. Desired confidence is equal to 95%.

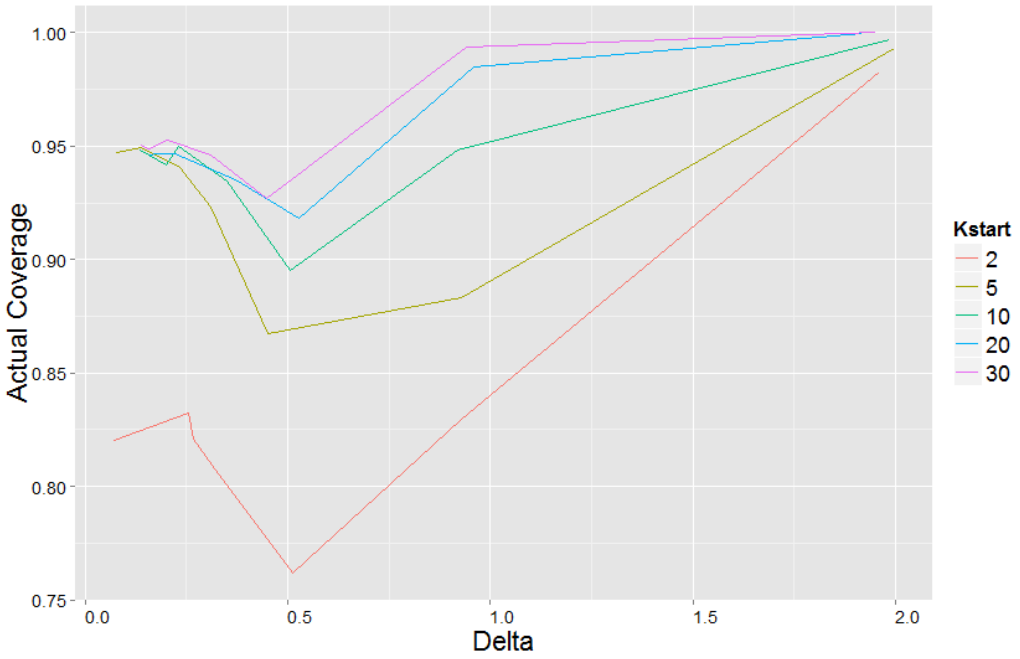

Figure 30. Delta versus probability of coverage for the day that blue forces achieve air supremacy. Desired confidence is equal to 95%.

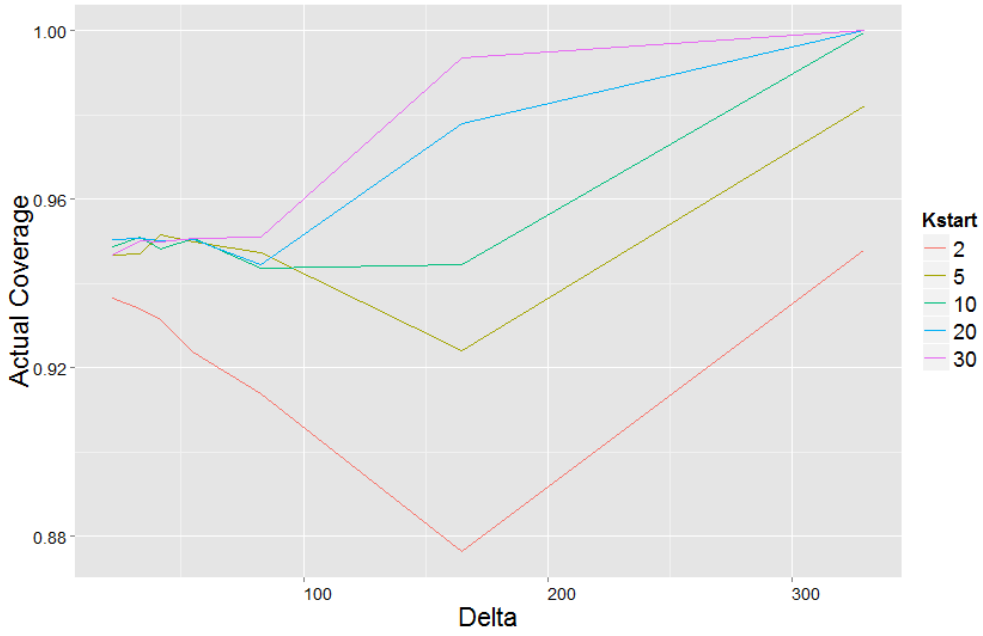

Figure 31. Delta versus probability of coverage for the number of blue multirole fighter missions flown. Desired confidence is equal to 95%.

The relationship between δ and the coverage shown in these figures, reveals that the coverage is approximately stable for *Kstart* values 10 or greater. This is portrayed in Figure 28, where, for all *Kstart* values that are greater than 10, they remain at or close to the desired probability of coverage (95 percent) for all values of δ. It is recommended that analysts explore this relationship to gain insight into the minimum number of replications to obtain when applying a sequential stopping rule.

F. EXPECTED NUMBER OF REPLICATIONS VERSUS PROBABILITY OF COVERAGE IN STORM

In a perfect world, an infinite number of replications would be performed to get perfectly precise estimates of the mean and variance of simulation output. As the number of replications increases, the probability of coverage converges to the desired level of coverage. There is, however, a cost to the number of replications performed. This cost is usually time or available memory. As a result, the goal is to find the expected minimum number of replications required to achieve a desired level of precision. Recall that the goal in this experiment was to determine the expected number of replications to get a desired coverage of 95 percent using a sequential stopping rule. Figures 32 through 36 reveal that the convergence to the desired level of coverage of 95 percent takes place at approximately 100-200 replications for all *Kstart* values $\geq$ 5. In addition, the figures show that a high probability of coverage can be achieved with an expected low number of replications; however, this coverage reflects a larger δ, which implies a larger half-width, resulting in a wider confidence interval. Figure 35 also shows that for a lower number of replications, approximately 100, the desired probability of coverage can be achieved. This is a result of a larger δ associated with the extremely high variance for this metric.

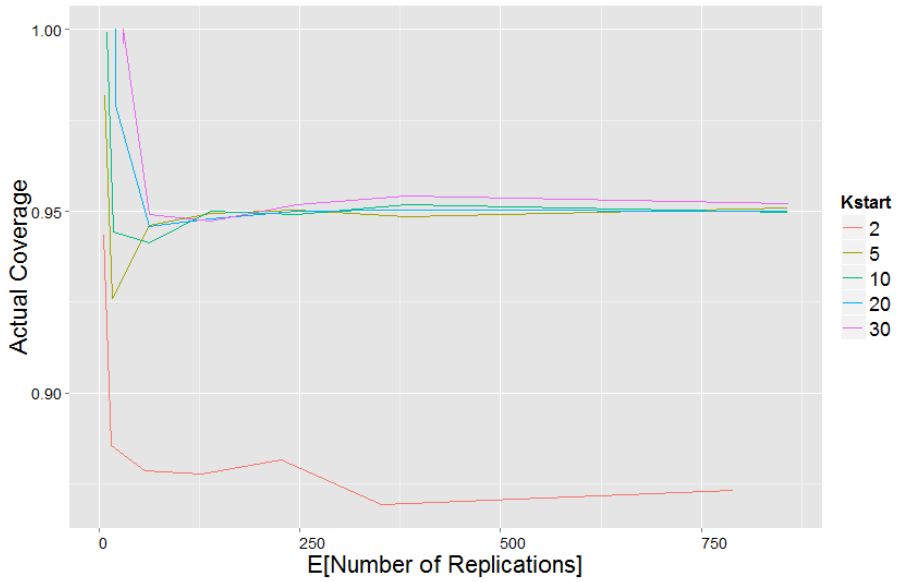

Figure 32. Expected number of replications versus probability of coverage for the number of blue ships remaining. Desired confidence is equal to 95%.

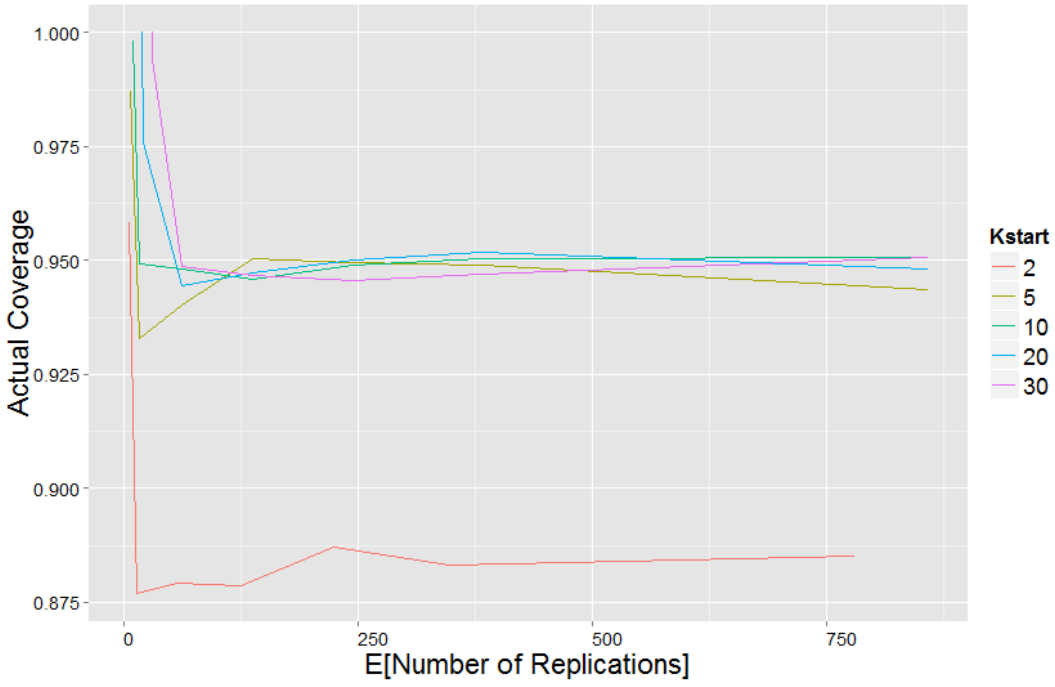

Figure 33. Expected number of replications versus probability of coverage for the number of red ships remaining. Desired confidence is equal to 95%.

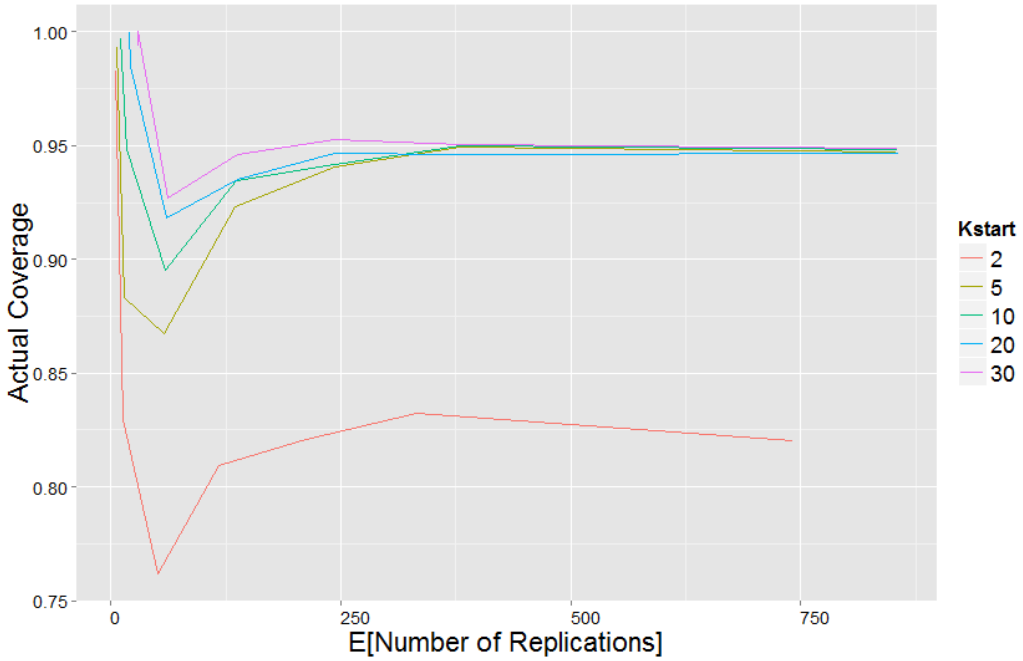

Figure 34. Expected number of replications versus probability of coverage for the day that blue forces achieve air supremacy. Desired confidence is equal to 95%.

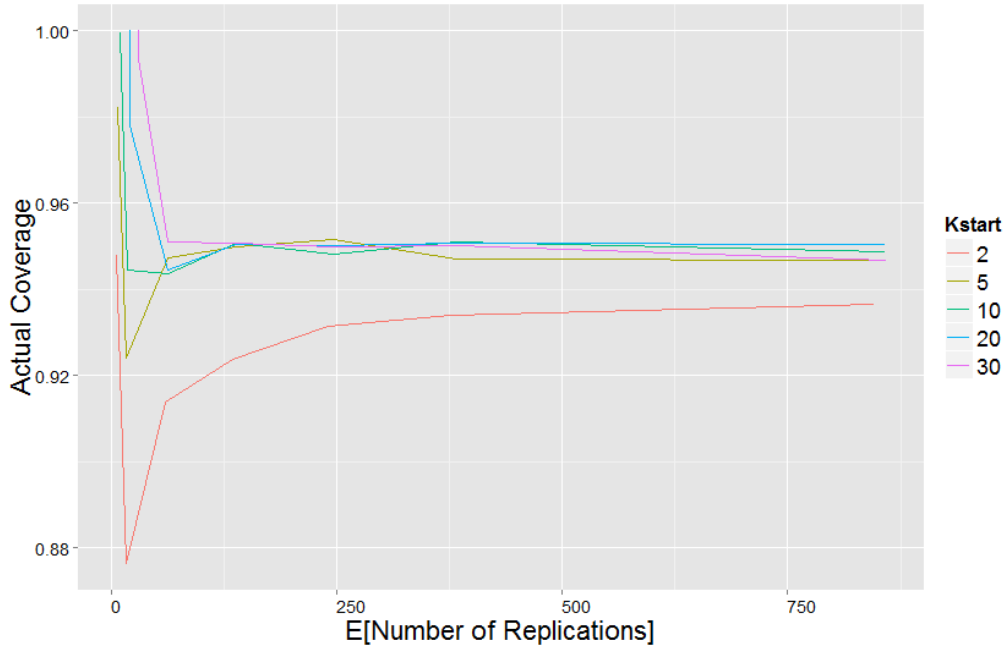

Figure 35. Expected number of replications versus probability of coverage for the number of blue multirole fighter missions flown. Desired confidence is equal to 95%.

In summary, coverage can be obtained with fewer than 100 replications. In fact, as seen in Figure 33, for *Kstart* values greater than 10, coverage can be obtained with approximately 50 replications. Analysts should explore this relationship on metrics of interest when determining the number of replications required in obtaining their desired coverage keeping in mind lower coverage may be acceptable.

G. APPLYING STOPPING RULES TO MAKE DECISIONS

Determining the number of replications required from trade-off analysis between precision and probability of coverage can be achieved with the methodology described in this chapter. Recall that there are two types of methods for sampling: fixed and sequential. Fixed sampling is determining the expected number of replications based on a desired probability of coverage and precision. Once the trade-offs between these factors are studied, the analyst will have an idea of how many replications to do. In contrast, sequential sampling involves calculations after each replication until the desired precision and probability of coverage is achieved. Both methods use the same idea as presented in this thesis, but differ slightly in the way in which the data is sampled.

The following example illustrates how the three figures can be used to inform the analyst. Table 12 is summary data extrapolated from Figures 25, 29, and 33, which are the figures for the number of red ships remaining. Recall that the desired coverage is equal to 95 percent and the mean number of red ships remaining is 42. Assume that *Kstart*, or the minimum number of replications, is equal to 30 to simplify the explanation.

Delta (δ)	E[Number of Replications]	Actual Coverage
0.5	125	0.9450
1.0	60	0.9550
1.5	40	0.9875
2.0	30	0.9900

Table 12. Data extrapolated from figures for the number of red ships remaining, which explains the trade-off between the expected number of replications, precision, and the probability of coverage.

From the information in Table 12, the following confidence intervals can be developed on the unknown population mean of the number of red ships remaining:

- [41.5, 42.5] with 94.50 percent confidence and a cost of 125 replications
- [41, 43] with 95.50 percent confidence and a cost of 60 replications
- [40.5, 43.5] with 98.75 percent confidence and a cost of 40 replications
- [40, 44] with 99.00 percent confidence and a cost of 30 replications

From these approximate confidence intervals, the analyst can determine the precision that suits the problem best. If a very precise confidence interval is required, then 125 replications would be recommended. If, however, a less precise interval is satisfactory, 30 replications may suffice.

Although this information is informative, it is important to understand that the results are approximate. If only one metric is used to determine the number of replications, then one stopping rule is being used for all metrics. In order to account for this, it is recommended that multiple metrics are tested with the stopping rule, as this research presents. High variance metrics are important because they can give worse results. In addition, since STORM is stochastic, it is possible that the output from a separate set of runs (e.g., different output metrics and scenario) has a completely different distribution of outcomes. If this is the case, stopping rule testing needs to be completed again to ensure that enough replications are completed. In addition, resampling the same data, as conducted in this research, provides approximate coverage.

To account for these approximations, it is recommended that multiple metrics are tested at different levels of precision. For a conservative approach, it is also important to ensure that a portion of the metrics have a relatively large variance, as experimented with in this thesis. In addition, stopping rules should be tested on both normal and nonnormal data, as conducted in this thesis. Including high-variance metrics, and both normal and non-normal output data, will allow the analyst to take a conservative approach.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

This research will enhance the ability of OPNAV N81 analysts who utilize STORM to quickly and accurately inform senior leaders. The results from Chapters III and IV are being used in software, known as STORMMiner, developed by research associates from the SEED Center at NPS. The purpose of STORMMiner is to harvest output data from STORM and automatically generate figures and plots that will decrease the amount of time it takes analysts to gain insight into a particular campaign. The research in this thesis has been included in STORMMiner, so that analysts can have confidence in the number of replications executed. In addition, Chapter II explains the fundamentals of how STORM works, laying a foundation for future research in the SEED Center.

A. DISTRIBUTION OF OUTCOMES

Chapter III focused on the showing the robustness of outputs that STORM provides without changing any input variables. This reveals the inherent stochasticity embedded in STORM, which is critical for modeling the unpredictability of combat. It was also shown that some of the key metrics of STORM are approximately normal; however, some are not. The importance of normality does not affect the research in Chapter IV; however, many statistical tests and algorithms require the normality assumption to be met.

B. THE VALUE OF REPLICATIONS

Replications can become increasingly expensive in time and memory for large-scale simulations such as STORM. The number of runs needed depends on the variability of the output, the precision required, and the coverage desired. Sometimes very few runs are needed (e.g., if the precision is large relative to the variance). Otherwise, many runs may be required to meet a narrow desired precision level. Organizations sometimes only do a certain number of replications due to the constrained environment in which they operate. Chapter IV quantifies the trade-off between the expected number of replications

versus precision and the probability of coverage. This trade-off is important to understand because if the precision and probability of coverage that the organization requires are not met, then the analysis is less valuable.

The exploration of the value of replications for sequential stopping rules in this thesis is being applied directly in STORMMiner. Less exploration of the trade-off is required if a user-defined level of precision is established and the algorithm executes at a single level. The algorithm will reveal the number of replications and the probability of coverage required to meet the desired level of precision. Analysts would then know, at a minimum, they needed to complete the expected number of replications in order to be confident in their analysis.

C. RECOMMENDATIONS

It is recommended that normality testing be conducted on metrics that require the assumption of normality in the statistical analysis chosen. This will prevent analysts from presenting results that violate fundamental statistical assumptions. In addition, it is recommended that, for all key metrics, testing be completed to determine the number of replications required to meet a desired precision and probability of coverage. STORMMiner provides a great deal of analysis quickly. This analysis uses many statistical techniques, including those covered in this thesis. It is recommended that all analysts acquire the basic training to utilize STORMMiner and begin to take advantage of the "free" analysis that it can provide.

There is room for follow-on research in two areas. The first is to apply the research from Chapters III and IV on a classified scenario at OPNAV N81 to determine if there is a significant difference. The second is to continue this analysis through a small design of experiments, which would not only the capture the stochasticity designed in STORM reviewed in this thesis, but the effect of changing key input variables. If key input variables are changed at different levels, more experiments have to be completed, thus reducing the time for experimentation at each level. As a result, it would be critical to know if an analyst could complete fewer replications for a desired level of precision.

APPENDIX A. CHAPTER III SOURCE CODE

This appendix contains code used to conduct statistical analysis and normality testing on a STORM output metric. Once the file is read into R, summary statistics and histograms are easily developed. Next, visualizations for normality are created utilizing histograms and QQ plots. Finally, formality normality testing is conducted and the results can easily be interpreted through the box plots that are generated.

```
##Calculates summary statistics and performs normality testing
##LT Christian Seymour
##May 2014

metric<-read.csv(file.choose()) #Read in metric file of choice

#calculate Summary Statistics
avg<-mean(metric$time)
min<-min(metric$time)
max<-max(metric$time)
med<-median(metric$time)
n<-length(metric$time)
sd<-sd(metric$time)*(n/(n-1))
error<-qt(0.975,df=n-1)*sd/sqrt(n)
left<-avg-error
right<-avg+error

####Generate random variables based on actual data parameters
exp.var.b.metric<-rexp(82, rate = 1/avg)
unif.var.b.metric<-runif(82,min =min, max=max)
norm.var.b.metric<-rnorm(82,mean = avg)

##Install R Package e1071
skewness(metric$time)

###Build histogram, must click on plot where you would like legend
hist(metric$time, breaks = 20, col = "light blue", main = "Title", xlab = "X axis Label")
abline(v=avg, col = "blue", lwd = 2)
abline(v=med, col = "blue", lty =2, lwd =2)
legend(locator(1),lty=c(1,2),lwd=c(2,2), col = c("blue", "blue"), legend
=c("Mean","Median") )


##Set up 4 by 4 plotting for comparing histograms
```

```
par(mfrow = c(2,2))
hist(metric$time, breaks = 20, col = "blue", main = "Title", xlab = "X axis Label")
hist(exp.var.b.metric, breaks = 20, col = "Green", xlab = "Random Exponential Variables
(rate = 1/16.55)", main = "82 Random Exponential Variables")
hist(unif.var.b.metric, breaks = 20, col = "Green", xlab = "Random Uniform Variables
(min = 11.75, max = 19.75)", main = "82 Random Uniform Variables")
hist(norm.var.b.metric, breaks = 20, col = "Green", xlab = "Random Normal Variables
(mean = 16.55)", main = "82 Random Normal Variables")

##Build QQ Plots
par(mfrow = c(2,2))
qqnorm(metric$time, col = "blue", main = "Title")
qqnorm(exp.var.b.metric, col = "green", main = "QQ Plot of 82 Exp R.V.")
qqnorm(unif.var.b.metric, col = "green", main = "QQ Plot of 82 Unif R.V.")
qqnorm(norm.var.b.metric, col = "green", main = "QQ Plot of 82 Norm R.V.")


#####NORMALITY TESTING
par(mfrow=c(1,1))
val.ad<-c()
vec.ad<-c()
val.cvm<-c()
vec.cvm<-c()
val.ks<-c()
vec.ks<-c()
val.sp<-c()
vec.sp<-c()

n<-1
while (n<1000){
  rnd <- sample(metric$time,30,TRUE)
  val.ad<-ad.test(rnd)$p.value
  vec.ad<-c(vec.ad,val.ad)
  val.cvm<-cvm.test(rnd)$p.value
  vec.cvm<-c(vec.cvm,val.cvm)
  val.ks<-lillie.test(rnd)$p.value
  vec.ks<-c(vec.ks,val.ks)
  val.sp<-shapiro.test(rnd)$p.value
  vec.sp<-c(vec.sp,val.sp)

  n<-n+1

}

### NORMALITY PLOTS
```

```
boxplot(vec.ad,vec.cvm,vec.ks,vec.sp, col = "blue",names = c("Anderson-
Darling","Cramer-von Mises","Kolmogrov-Smirnov","Shapiro-Wilk"),xlab = "Normality
test",ylab = "P-Value", main = "Title")
abline(h=.05, lty = 2, col = "red", lwd = 3)
legend(locator(1),lwd=3,lty=2,col="red",legend="P-Value = .05", cex = .8)
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. CHAPTER IV SOURCE CODE

This appendix contains the code for exploring the relationship between the expected number of runs, the desired cover, and the required precision. The metric of interest is read into R and the code will reveal the relationship through the generated output plots, as seen in Chapter IV. A csv file with the results is also outputted for any further analysis to be conducted.

```
# Calculates needed sample sizes for a simulation run for a set of metrics.
# see Singham and Schruben: Finite-Sample Performance of Absolute Precision Stopping Rules
# 2 INFORMS Journal on Computing, Articles in Advance, pp. 1–12, for more details.
# also see LT Christian Seymour's thesis
# LT Christian Seymour
# modifications by Stephen C. Upton
# SEED Center for Data Farming
# 7/23/2014

# NOTE: NA's are removed from the metric values for this computation; you'll need to clean/pre-process your input to account for any possible NAs


m.test <- 10000


# del <- c(15)
# k.start <- c(30,40,50)
# eta <- c(0.95)

del <- c(1,2,4,6,8,10,15)
k.start <- c(2,5,10,20,30)
eta <- c(0.8,0.85,0.9,0.95)

in.file <- "blue_ships_110.csv"
out.file.name <- "Blue_ships_110.stoprules-2.csv"

#dat.storm<-
read.csv("/Users/Seymour/Desktop/STORM_WORKING_DOCUMENTS/ACTUAL_THESIS/Chapter_4_N/blue.ships.remaining-offdailykillsreport.csv")

dat.storm <- read.csv(in.file)
start.seed <- 3141592
```

```
compute.needed.sample.size <- function(x,m=10000,nmax=1000,eta=0.90,kstart=2,delta=1,seed=693147) {
 # Computes the needed sample size for a simulation output metric, with a specific confidence level.
 #
 # Args:
 #   x: a vector of output metric data.
 #    m: The number of bootstrap experiments to run to estimate probability of coverage and sample size needed. Default is 10000.
 #   nmax: The number of resamples. Default is 1000.
 #   eta: The number of resamples. Default is 0.90.
 #   kstart: The number of resamples. Default is 2.
 #   delta: The number of resamples. Default is 1.
 #   seed: a starting seed to use for the bootstrap sampling
 #
 # Returns:
 #   A dataframe with the starting input values for kstart, delta, and eta, and
 # the output values of the approximate coverage, and needed sample size.

 cat(sprintf("using m value of %d, kstart value of %d, delta value of %f, eta value of %f, seed is %d\n",m,kstart,delta,eta,seed))

 if (delta <= 0 ) {
  cat("ignoring this input: delta value ", delta," is less than 0\n")
  return(list(eta.star=0, k.avg=0,delta.star=0))
 }

 mu <- mean(x)
 ek.cov <- compute.Ek.cov(x,delta,mu,m,nmax,eta,kstart,seed)

 df <- data.frame(kstart=kstart,delta.star=delta,eta=eta,eta.star=ek.cov$cov/m, k.avg=ek.cov$meanEk,runtime=ek.cov$runtime,seed=seed)
 cat("run time: ",df$runtime," secs\n")
 df
}

compute.Ek.cov <- function(x,delta,mu, m=10000,nmax=1000,eta=0.90, kstart=2,seed=693147) {
 # set the seed for this particular experiment, so we can easily reproduce results for this specific input set
 set.seed(seed)
 tval <- sapply(1:nmax,function(z) qt((1+eta)/2,z))
 t1 <- system.time({
  tt <- lapply(1:m,function(xx) {
```

```
    rnd <- sample(x,nmax,TRUE)
    k <- kstart
    repeat{
      hw <- tval[k-1]*sqrt(var(rnd[1:k])/k)
      if(hw <= delta || k >= nmax){
        break
      }
      k <- k + 1
    }
    xbar <- mean(rnd[1:k])
    data.frame(k=k,cov=abs(xbar - mu) <= delta,xbar=xbar)
  })
  df <- do.call(rbind,tt)
 })
 # uncomment the next 2 lines if you want an output file listing intermediate results
 # ofile <- paste0("ek_cov.",paste(kstart,eta,delta,sep="-"),".csv")
 # write.csv(df,ofile,quote=FALSE,row.names=FALSE)
 list(cov=sum(df$cov),meanEk=mean(df$k),runtime=t1[3])
}

t0 <- system.time({
 for (p in names(dat.storm)){
  x <- dat.storm[,p]
  x <- x[!is.na(x)]
  if ( length(x) == 0 ) {
    cat("WARNING: all of your input data is NA for this metric: ",p," - skipping
computation\n")
    break
  }
  sd<-sd(x)
  del.vec<-c(sd/del)
  o <- expand.grid(eta=eta,del.vec=del.vec,k.start=k.start)
  set.seed(start.seed)
  o$seed <- floor(runif(nrow(o))*1000000)
  df                                    <-                                    Map(function(e,d,k,s)
compute.needed.sample.size(x=x,m=m.test,eta=e,delta=d,kstart=k,seed=s),o$eta,o$del.ve
c,o$k.start,o$seed)
  dynamic.df <- do.call(rbind,df)
 }
})



t <- c(t0[3],t0[3]/60,t0[3]/3600)
m <- c("secs","mins","hrs")
```

```
cat("time to execute: ",paste(format(t,digits=2),m,collapse=":"),"\n")
#out.file <- file.path(getwd(),"Blue_ships.stoprules.csv")
#write.csv(dynamic.df, out.file,quote=FALSE,row.names=FALSE)
out.file <- file.path(getwd(),out.file.name)
write.csv(dynamic.df, out.file,quote=FALSE,row.names=FALSE)
df.B <- read.csv("/Users/Seymour/Desktop/Thesis_Formatted/Blue_ships.stoprules.csv")
df.B.95<-df.B[df.B$eta == .95,]

####PLOT Delta versus Expected Number of Replications#########################################################################
p<-ggplot(df.B.95, aes(x=jitter(amount=0.06, df.B.95$delta.star), y = df.B.95$k.avg, fill=as.factor(kstart), color = as.factor(kstart)))+geom_line()+
  xlab("Delta")+ylab("E[Number of Reps]") #+labs(title = "Delta versus E[Number of Reps]\n Blue Ships Remaining\nEta = .95")
p + guides(color=guide_legend(title="Kstart"))+theme(axis.text.x = element_text(angle = 0, hjust = 0, size=14,color="black")) + theme(axis.title.x = element_text(size = rel(1.8), angle = 00))+ theme(axis.title.y = element_text(size = rel(1.8), angle = 90))+theme(axis.text.y = element_text(angle = 0, hjust = 1, size=14,color="black"))+
  theme(legend.title=element_text(size=16)) +theme(legend.text=element_text(size=18))
p <- p + scale_color_manual(values=c("Red"))

####DELTA versus Probability of Coverage#########################################################################
p<-ggplot(df.B.95, aes(x=jitter(amount=0.06, df.B.95$delta.star), y = df.B.95$eta.star, fill=as.factor(kstart), color = as.factor(kstart)))+geom_line()+
  xlab("Delta")+ylab("Actual Coverage") #labs(title = "Delta versus P(Coverage)\n Blue Ships Remaining\nEta = .95")
p + guides(color=guide_legend(title="Kstart"))+theme(axis.text.x = element_text(angle = 0, hjust = 0, size=14,color="black")) + theme(axis.title.x = element_text(size = rel(1.8), angle = 00))+ theme(axis.title.y = element_text(size = rel(1.8), angle = 90))+theme(axis.text.y = element_text(angle = 0, hjust = 1, size=14,color="black"))+
  theme(legend.title=element_text(size=16)) +theme(legend.text=element_text(size=18))
p <- p + scale_color_manual(values=c("Red"))

#######Desired Coverage versus Expected Reps###########################################
p<-ggplot(df.B.95, aes(x=jitter(amount=0.06, df.B.95$k.avg), y = df.B.95$eta.star, fill=as.factor(kstart), color = as.factor(kstart)))+geom_line()+
  ylab("Actual Coverage")+xlab("E[Number of Replications]")
p + guides(color=guide_legend(title="Kstart"))+theme(axis.text.x = element_text(angle = 0, hjust = 0, size=14,color="black")) + theme(axis.title.x = element_text(size = rel(1.8),
```

```
angle = 00))+ theme(axis.title.y = element_text(size = rel(1.8), angle = 90))+theme(axis.text.y = element_text(angle = 0, hjust = 0, size=14,color="black"))+
  theme(legend.title=element_text(size=16)) +theme(legend.text=element_text(size=18))
p <- p + scale_color_manual(values=c("Red"))
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Clausewitz, Carl Von. (1852). What is war, Howard, Michael and Paret, Peter (Eds. and Trans.). *On War.* Princeton NJ: Princeton University Press.

Currie, C., & Cheng, R. (2013). A practical introduction to analysis of simulation output data. *Proceedings of 2013 Winter Simulation Conference*.

Department of Defense. (2010, December). *DOD modeling and simulation glossary*. Modeling and Simulation Coordination Office. Retrieved from http://www.msco.mil/documents/_4_Final_Glossary.pdf

Department of Defense. (2014, 15 June). *Department of Defense dictionary of military and associated terms* (JP 1-02). Retrieved from http://www.dtic.mil/doctrine/new_pubs/jp1_02.pdf

Group W. (2012a). *STORM: Analyst's Manual Version 2.3*. Fairfax, VA: Group W.

Group W. (2012b). *STORM: Programmer's Manual Version 2.3*. Fairfax, VA: Group W.

Group W. (2012c*). STORM: User's Manual Version 2.3*. Fairfax, VA: Group W.

Hawley, P., & Blauwkamp, R. (2010). Six-degree-of-freedom digital simulations for missile guidance, navigation, and control. *John Hopkins APL Technical Digest*, *29*(1), 71–84.

Law, A. (2007). *Simulation modeling & analysis.* New York, NY: McGraw-Hill.

Lucas, T. (2000). The stochastic versus deterministic argument for combat simulations: Tales of when the average won't do. *Military Operations Research*, *5*(3), 9–28.

Singham, D. I. (2010). *Analysis of sequential stopping rules in simulation experiments* (Doctoral dissertation). Retrieved from http://escholarship.org/uc/item/3hb6p7bg

Singham, D. I. (2014). Selecting stopping rules for confidence interval procedures. *ACM Transactions on Modeling and Computer Simulation*, *24*(3), Article 18.

Singham, D. I., & L. W. Schruben. (2012). Finite-sample performance of absolute precision stopping rules. *INFORMS Journal on Computing*, *24*(4), 624–635.

Sweeney, R., Hamman, J., & Biemer, S. (2011). The application of systems engineering to software development: A case study. *John Hopkins APL Technical Digest*, *29*(4), 327–337.

United States Government Accountability Office. (2012). *Joint Strike Fighter: DOD actions needed to further enhance restructuring and address affordability risks* (GAO-12-437). Washington, DC: Government Accountability Office. Retrieved from http://www.gao.gov/assets/600/591610.pdf

Wackerly, D., Mendenhall III, W., & Scheaffer, R. (2008). *Mathematical statistics with applications.* Belmont, CA: Brooks/Cole.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California