412TW-PA-15215

# The Last Word on TLE

4
1
2
T
W


**James Brownlow**

**AIR FORCE TEST CENTER**
**EDWARDS AFB, CA**



**12-14 May, 2015**





**AIR FORCE TEST CENTER**
**EDWARDS AIR FORCE BASE, CALIFORNIA**
**AIR FORCE MATERIEL COMMAND**
**UNITED STATES AIR FORCE**

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| | | |
|---|---|---|
| **1. REPORT DATE** *21/04/2015* | **2. REPORT TYPE** ITEA presentation | **3. DATES COVERED** *(From - To)* |
| **4. TITLE AND SUBTITLE**<br>The Last Word on TLE | | **5a. CONTRACT NUMBER**<br>**5b. GRANT NUMBER**<br>**5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)**<br>James Brownlow | | **5d. PROJECT NUMBER**<br>**5e. TASK NUMBER**<br>**5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)**<br>412 TSS/ENT | | **8. PERFORMING ORGANIZATION REPORT NUMBER**<br>412TW-PA-15215 |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** | | **10. SPONSOR/MONITOR'S ACRONYM(S)**<br>N/A<br>**11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release A: distribution is unlimited.

**13. SUPPLEMENTARY NOTES**
CA: Air Force Test Center Edwards AFB CA CC: 012100

**14. ABSTRACT**
Use a probability density function developed by Herschel Weil in 1953 to characterize TLE. The method provided for independent normally distributed errors in both the along- and cross track axes.

**15. SUBJECT TERMS**

| **16. SECURITY CLASSIFICATION OF:** Unclassified | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | 412 TENG/EN (Tech Pubs) |
| Unclassified | Unclassified | Unclassified | None | 30 | **19b. TELEPHONE NUMBER** *(include area code)*<br>661-277-8615 |

**Standard Form 298 (Rev. 8-98)**
**Prescribed by ANSI Std. Z39.18**

*War-Winning Capabilities … On Time, On Cost*

# The Last Word on TLE

## 12-14 May, 2015

**James Brownlow**
**James.Brownlow.1@us.af.mil**
**812 TSS/ENT**
**661 277-4843**

U.S. AIR FORCE

**Approved for public release; distribution is unlimited.**
**412TW-PA No.: 412 TW-PA-15215**

# Overview

- Purpose
- Background
- Estimating CEP, CE90
  - Current Methods
  - Weil:  derivation of p(r) (r=radial error)
- Example
- Conclusions

# Purpose of the presentation

- **First- demonstrate Python code for estimation of CEP and/or CE90 values from a "small" data set**
- **Secondly- present aspects of Python that are different from standard programming languages- python can be characterized as a cross between R and C++ with methods**

# Title of the presentation

- **Original thought: Snakes in the Cubicle**

**Python runs in both Windows and Linux**

# background

- **A typical MOP for applications such as navigation, target locater systems, ballistics is CEP or CE90 *aka* the $50^{th}$ and $90^{th}$ percentile (radial) error values**
- **A variety of techniques exist for estimating these quantities from a data sample- basically the procedures fall into one of two categories:**

# Estimating CEP, CE90

- First category: collect the radial errors, use maximum likelihood to fit right-tailed distributions (Rayleigh, Weibull, gamma, lognormal, log-logistic) to the data
- Use the "best fit" distribution to estimate the 50$^{th}$ and/or 90$^{th}$ percentile values, or
  - Use a combination (weighted sum) of the fit distributions to estimate the 50$^{th}$ or 90$^{th}$ percentile – weighting based on AICc

# Estimating CEP, CE90

- Second category: use a bivariate Rayleigh distribution predicated on the assumptions:
  - different standard deviations in each axis, and
  - target location is the mean point of impact
- then use numerical estimation techniques, to estimate the CEP and CE90 values (‘Distribution of Radial Error in the Bivariate Elliptical Normal Distribution,’ Victor Chew et al. , *Operations Research Branch, U.S. Naval Weapons Laboratory*, Dahlgren, VA, 1962)

# BUT there's a third way!

- In 1954 Herschel Weil published, "The Distribution of Radial Error," *The Annals of Mathematical Statistics*, Vol. 25 No. 1 (Mar 1954) pp 168-170
- Assumes radial errors are based on errors in x- and y-axes assumed normally distributed with means ($\mu_x$ , $\mu_y$ ) and standard deviations ($\sigma_x$ , $\sigma_y$ )
- This is most appropriate to the TLE problem, because…

# Weil- generally best because…

- **Central limit theorem supports the notion that errors along- and cross track are normally distributed**
- **No reason to assume that the mean error is zero, nor that the standard deviations are the same in each axis**
- **This is exactly the situation we face in flight test analysis**

# Weil estimation

- **So why hasn't this been in use since 1954??**
- **-> the difficulty in integrating, and using infinite sums of Bessel functions!** ***Viz.*****, the radial error density function is:**

$$p(r) = A * r * \exp\{-r^2 * \frac{(\sigma_x^2 + \sigma_y^2)}{4 * (\sigma_x^2 * \sigma_y^2)}\} *$$

$$\{I_0(a * r^2) * I_0(d * r) + 2 * \sum_{j=1}^{\infty} I_j(a * r^2) * I_{2*j}(dr) * \cos(2 * j * \psi)\}$$

- **For details, see Weil**

# What's all this mean?

- Weil derived the probability density function for radial errors based on normally distributed errors in x- and y- axes- with no restriction on the two normal distributions
- Without loss of generality, Weil's formulation assumes no correlation between the x- and y- axis error. Flight test data can be rotated to attain 0 correlation

# So what’s the procedure?

- Use along- and cross-track error means and standard deviations as initial values for maximum likelihood estimates of the parameters of the radial error distribution
- If along- and cross-track data are not available, only the radial errors, use “vague” initial values for the initial along- and cross-track parameters in maximum likelihood estimation
- In either case we get a pdf for radial errors that is statistically defensible

# Aside: Python

- **Edwards introduced the use of python as an alternative to MATLAB**

- **Enthought Canopy Python is now available to engineers and analysts**
- **Enthought did a one-week Python for Engineers class at Edwards in November 2014.**

# Python, continued

- Like R, Python has packages for statistics; it has a vast collection of packages for numerical methods, data processing, engineering and scientific problems and graphics packages
- Also, like R, Python as a somewhat steep learning curve.  It is an application language, not a "canned program."
- I implemented Weil's p(r) in Python

# Estimation of CEP, CE90

- I wrote a short (three page, ~ 180 lines of code) python program to estimate the parameters of the Rayleigh density function in Weil's paper
- I found a number of python attributes that are different from languages like C++, Matlab ® scripts or R scripts

# Getting packages in python

```
from scipy import arctan2, median
import numpy as np
from math import sqrt, cos, isnan, log, atan2, sin
from scipy.optimize import minimize
from matplotlib.pyplot import hist, show, plot, figure
import pandas as pd
from os import chdir
```

# File processing

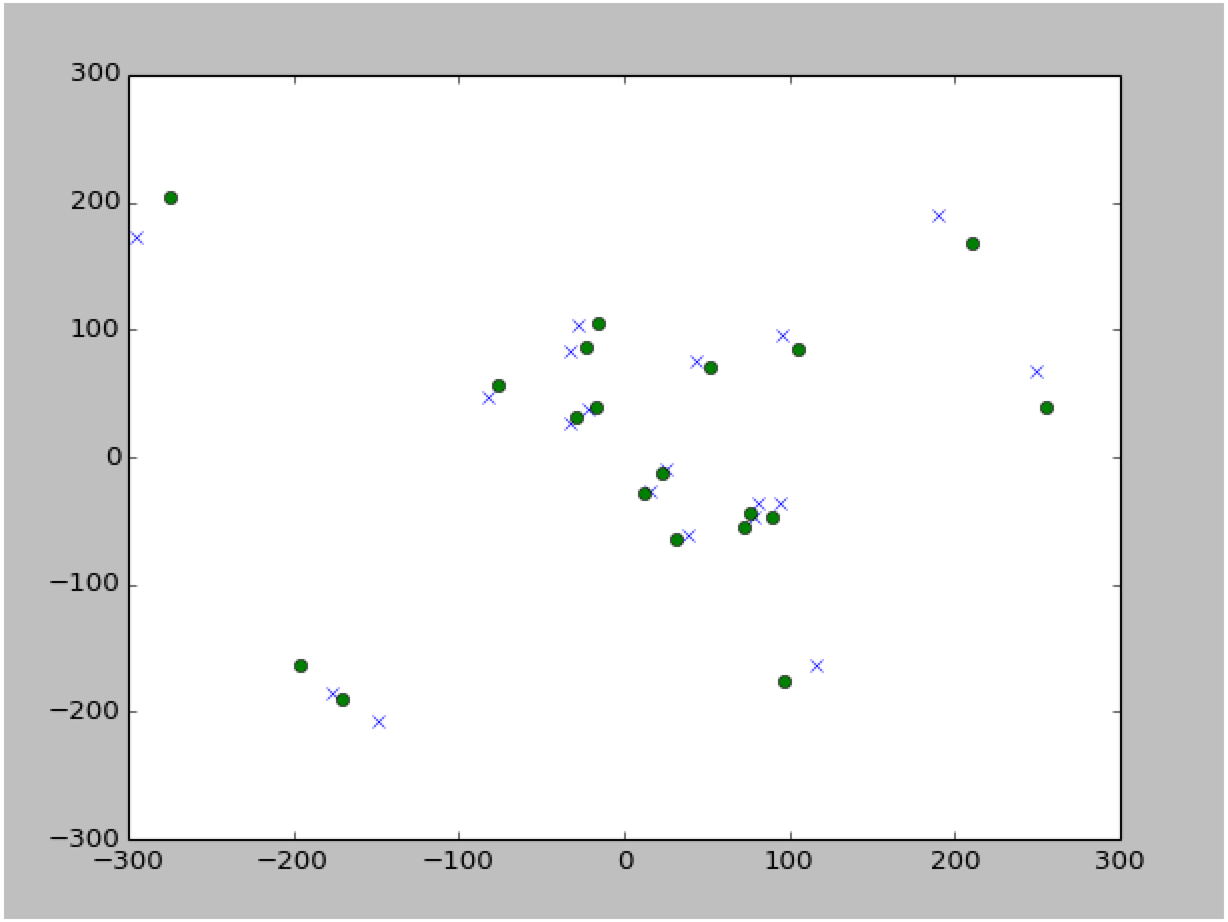
- filename="TPSrun.csv"

```
arr = pd.read_csv(filename)
r=(arr.x*arr.x + arr.y*arr.y)**0.5
np.mean(arr.x) # notice reference to np
np.mean(arr.y)
np.std(arr.x)
np.std(arr.y)
```

# plotting

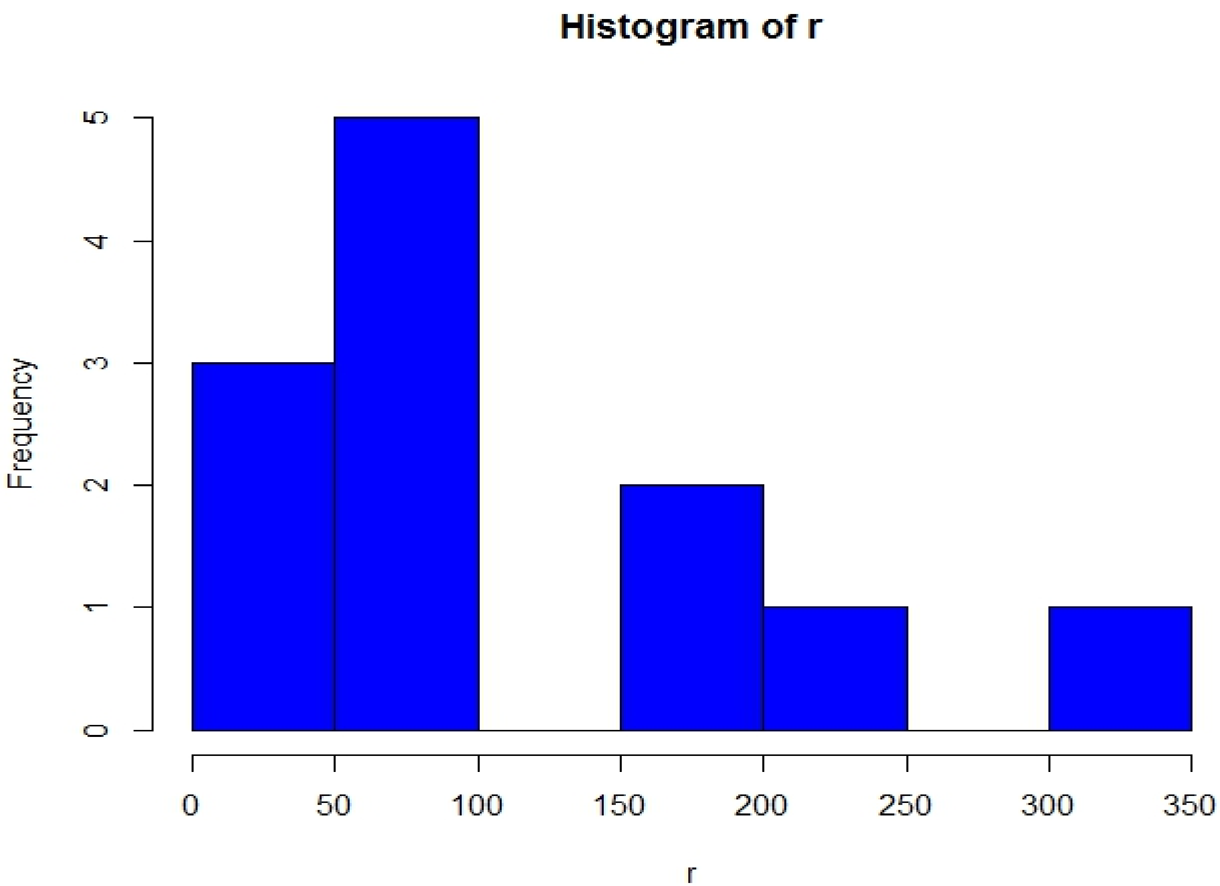
```
figure()   # open new plot figure
plot(arr.x, arr.y, 'x')  # plot the x and y data, ‘x’ plot symbol
show()
model = pd.ols(y=arr.y, x=arr.x) # linear model pandas
theta = atan2(model.beta.x, 1.0)
tArrX=[ ] ; tArrY=[ ]  # define arrays
for i in range(len(arr.y)):
# how to put stuff into an array!  append
tArrX.append(arr.x[i]*cos(theta) + arr.y[i]*sin(theta))
tArrY.append(-arr.x[i]*sin(theta) + arr.y[i]*cos(theta))

plot(tArrX, tArrY,'o')
show()
```

# Example: original and rotated data



# Radial error

- **Histogram of radial errors:**



# Arrays and loops

tArrx[ ], tArry[ ] are rotated data points

Loops in python

for i in list:   # no “{   }” to contain loop;

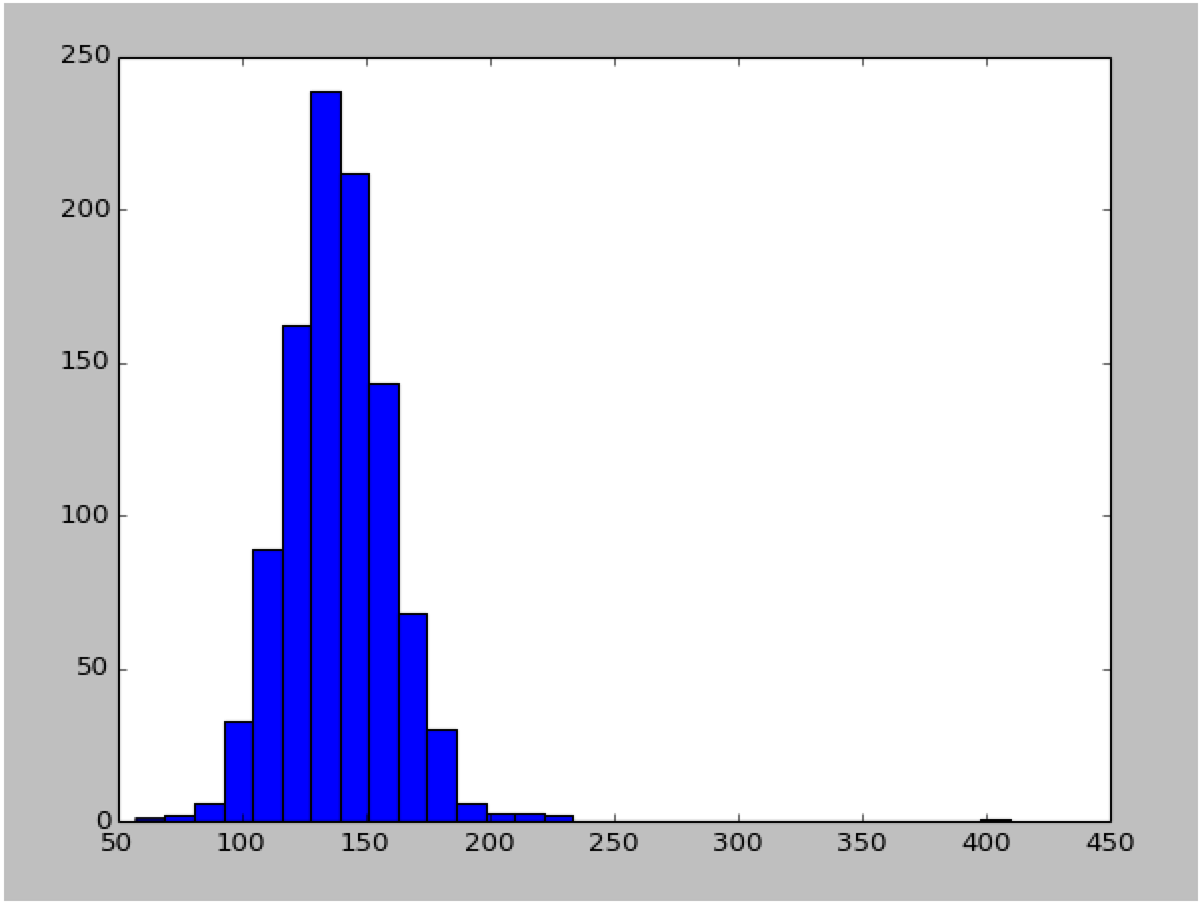
    print i

Loops defined by indentation

# And show the code…

- **Cut to Python code**

# CEP



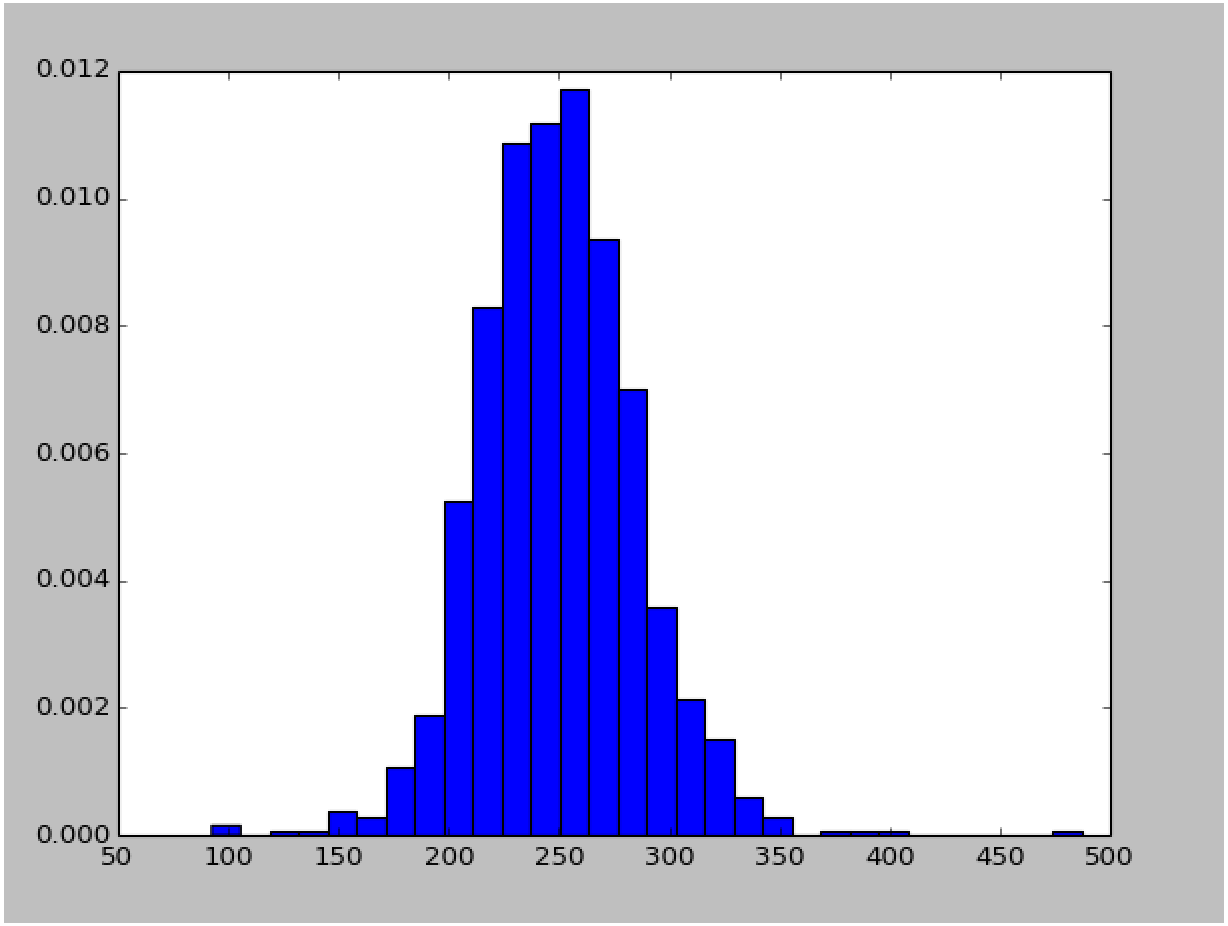
# Comparison of results- CEP

- **Summary**

| Estimation procedure | Estimated CEP | 95% CI lower bound | 95% CI upper bound |
|---|---|---|---|
| Single Distribution fit | | | |
| Multiple distribution fit | 117 | 75 | 160 |
| Bayes* | 149 | 104 | 230 |
| Weil- bivariate Rayleigh | 140 | 100 | 180 |

- ***Bayes based on two-dimensional bivariate normal, mean in each axis=0, offset applied**

# CE90



0.012
0.010
0.008
0.006
0.004
0.002
0.000
50
100
150
200
250
300
350
400
450
500

# Comparison of results- CE90

- **Summary**

| Estimation procedure | Estimated CE90 | 95% CI lower bound | 95% CI upper bound |
| --- | --- | --- | --- |
| Single Distribution fit | | | |
| Multiple distribution fit | 265 | 167 | 382 |
| Bayes | 240 | 160 | 390 |
| Weil- bivariate Rayleigh | 250 | 185 | 330 |

# Bottom line-

- **Small sample result shows about a 10% difference between current methods in use and Weil-radial distribution**
- **Likely due to differences in tail behavior of the Weil-radial distribution and the right-tailed distributions used to approximate radial error distribution**

# Questions?

- **Ready to try Python?  Data files and Python code available through O/A website.**
- **Edwards firewall will not permit sending/receiving .py files.  They will be text files you can read as text and copy into Enthought editor**