UNITED STATES MILITARY ACADEMY

MODELING PROJECT 2: TANK GUNNERY RANGE

MA391: MATHEMATICAL MODELING

SECTION B1-1

COL JAKE LAPORTE

By

CADET SAMANTHA CHINN '22, CO F1

CADET ANNA VINNEDGE '22, CO E2

WEST POINT, NEW YORK

01 MAY 2020

SC AV OUR DOCUMENTATION IDENTIFIES ALL SOURCES USED AND ASSISTANCE RECEIVED IN COMPLETING THIS ASSIGNMENT.

_______ WE DID NOT USE ANY SOURCES OR ASSISTANCE REQUIRING DOCUMENTATION IN COMPLETING THIS ASSIGNMENT.

SIGNATURES: Samantha Chinn Anna Vinnedge

MA391 PROJECT: MODELING TRAINING AT A TANK GUNNERY RANGE

INTRODUCTION

As the platoon leader of a training company at a tank gunnery range we are responsible for safely and effectively training 60 soldiers in a 12-hour time block. We will be training all soldiers simultaneously in the morning. Later in the day we will be running crews, consisting of a gunner and a driver, through individually. The length of the training block has an affect on the time it takes a crew to navigate the live fire as well as the probability that they will pass.

DATA

The data we use in this project was collected over the last couple of years at the tank range and consists of 1000 observations across 4 variables. A crew indicator variable, time spent training (hours), probability of success, and the time spent navigating the live fire course (minutes). We used this data to create models and run simulations in order to answer various questions which helped us to produce a schedule to be efficient in running the tank gunnery range.

PROBLEM #1: Predicting Navigation Time and Probability of Success

Assumptions:

- The training time educates every soldier on both positions (gunner and driver).
- All crews receive the same amount of training time.
    - There is no additional or secondary training if a team fails to qualify on their first run.
- The time it takes to navigate the live fire includes each soldier and both roles.
- We have 4 operational tanks available and working, but only 1 tank can be training at a time.
- The operational day is 12-hours, to include training and qualifying time.

In order to calculate the average time through the live fire and the probability of success, we created two linear models based on our data set. The first linear model predicts the amount of time it will take to navigate the live fire using the amount of time they spent training beforehand.

$$\widehat{Navigation}\ Time = 9.96 - 0.69 * (Training\ Time)$$

This linear model is useful in ensuring we can be most effective in balancing training and navigating time during our limited 12-hour day qualifying soldiers.

The second linear model predicts the probability that a crew is successful in qualifying on the live fire range using the amount of time spent training.

$$Probabil\widehat{ity\ of}\ Success = 0.65 + 0.05 * (Training\ Time)$$

This model is essential because in order to be most effective during the day we need to balance time spent training with the probability that the soldiers are successful. If we don't spend enough time training because we think we are saving more time to let them qualify, we actually could be decreasing our efficiency; as the training time decreases, so does the probability that they are successful and with that the number of iterations it will take in order to reach 95% of our soldier being qualified will increase, subsequently adding to our total time.

Using R Studio (see Appendix A), we tried various values for hours of training eventually narrowing to between 2 and 3 hours. We wanted to have a high probability of success and be able to train and qualify all soldiers in a 12-hour period. Of the values we tried, 2.5 hours of training allowed us to meet both those goals. With 2.5 hours of training in the morning, the predicted time through the live fire was 8.24 minutes with a 0.776 probability of qualifying.

PROBLEM #2: Random Variates from a Triangular Distribution

Given our predicted average time of 8.24 minutes to navigate the live fire from problem 1, we used the R command *rtri()* to get a distribution of 10,000 random variates. We then plotted them in R (see Appendix B) as a histogram and verified that they came from a triangular distribution.

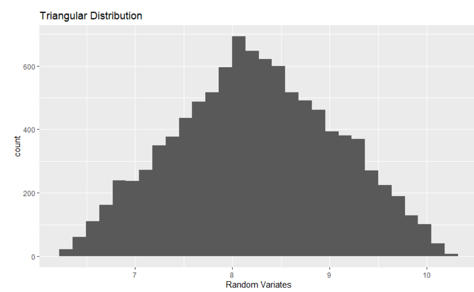


Figure 1: Triangular Distribution of 10,000 Random Variates

PROBLEM #3: 95% Qualification Analysis

Using the 2.5 hours of training and corresponding probability of success, 0.776, we were able to run a simulation in R (see appendix C) to determine how many iterations it would take for 95% of our crews to qualify. After running the simulation multiple times, on average, it took 69 iterations to qualify 57 out of our 60 crews.

PROBLEM #4: Recommended Schedule

The best way to achieve a 95% qualification rate would be to train all soldiers for 2.5 hours. This schedule assumes the accompanied success rate of 77.6% and predicted 8.24 minutes to navigate the live fire course. After training is complete, we have 9.5 hours (570 minutes) to complete 69 iterations of the live fire course and reach a 95% qualification rate. At 8.24 minutes each, the total time for qualifications is approximately 568.56 minutes. In total, it will take 11.98 hours (718.56 minutes) to train and qualify 95% of our soldiers. By using our recommended schedule, we will be able to train the required number of soldiers within our 12-hour time limit for the day.

In order to verify our schedule, we created a simulation in R (See appendix D) to run through the live fire qualification. Each time a group went through we subtracted off the average navigation time from the original 9.5 hours slotted for qualification iterations.

PROBLEM #5: Controlling Tank Supply

With old tanks that breakdown one per hour on average but can be fixed at a rate of 1.5 tanks per hour, in order to ensure that we will have 4 tanks operational per hour 95% of the time, we will need to transport 3 additional tanks to the training site. This will cost us an additional $6,000.

We determined that 7 total tanks are needed to ensure the 95% operation rate by using steady state probabilities and a Markov chain. We plugged the rates of transition from being in a state of operability versus a state of maintenance into a Markov transition matrix and calculated the null space in order to determine the probability of a tank being in either state at any given time.

In order to use this modeling method, we worked under the following assumptions:

- The rates of transition are constant
- Any number of tanks can be repaired at any given time
- Tanks break and are repaired independently
- We only ever use 4 tanks to train the soldiers
- Every tank is fixable
- Only 1 tank will break at a time
- State 1 is the tank is in use
- State 2 is the tank is in repair

We determined our steady state function to be

$$S_1'(t) = -S_1 + (1.5 * S_2)$$

$$S_2'(t) = S_1 - (1.5 * S_2)$$

The matrix operations were done in R (see Appendix E) to give us probabilities for the movement of a single tank. Using these results, we were able to test probabilities and find that between 6 and 7 tanks results in the 95% operability parameter. To safely ensure this parameter is met, we will bring up 3 additional tanks for a total of 7 tanks on site.

CONCLUSION

After an in-depth analysis of our data, our key final recommendations to the training site are as follows:

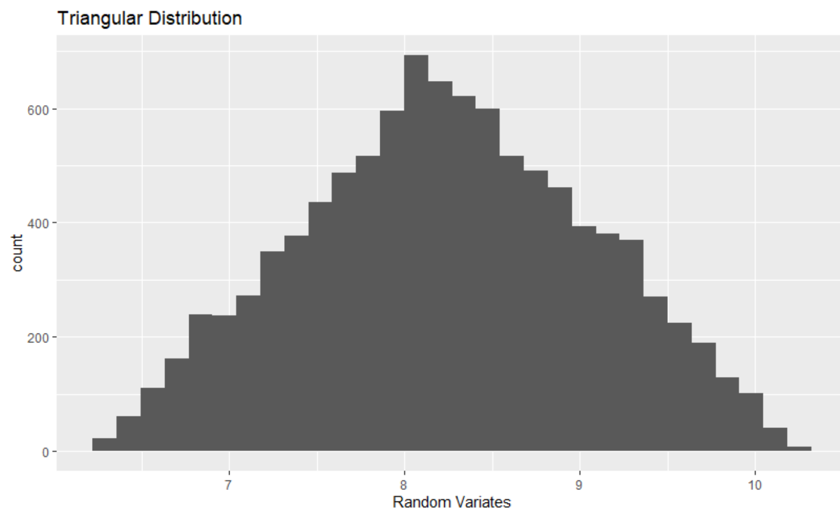
1. Train all soldiers for 2.5 hours
2. To qualify 95% of 60 soldiers on average, expect to run 69 iterations of testing to account for retraining.
3. Allot an additional $6,000 in order to pay for 3 extra tanks to allow for constant training

APPENDIX A

```
########## Problem 1 ######################################################################
navigate.model = lm(navigate.time ~ training.time, data = data);summary(navigate.model)
pass.model = lm(prob.success ~ training.time, data = data);summary(pass.model)

# We messed around with some numbers (720 minutes in a training day):
# 2.3 training hours = 716.22 total minutes
# 2.5 training hours = 715.8 total minutes  ################
# 2.6 training hours = 719.04 total minutes
# 2.8 training hours = 722.07 total minutes
# 2.5 looks the best

# We want to train for 2.5 hours.
# Calculate navigate time off of 2.5 hours of training.
ggplot(data=data) + geom_point(aes(x=training.time,y=navigate.time))
nav.time = 9.95773 - 0.68789*(2.5)
print(nav.time) # 8.238005 minutes per group

# Calculate predicted probability of success off of 2.5 hours training time:
prob.1 = 0.6511726 + 0.0500404*(2.5)
print(prob.1) # 0.776
```

APPENDIX B

```
########## Problem 2 ##########################################################################
# Probability distribution based off of our nav.time
library(EnvStats)
set.seed(10)
random.dist = rtri(10000, nav.time-2, nav.time+2, nav.time)
random.distro = data.frame(random.dist)
# Histogram of 10,000 random variates
ggplot(data = random.distro) + geom_histogram(aes(x=random.dist)) +
  labs(x="Random Variates",title="Triangular Distribution")
```



APPENDIX C

```
########## Problem 3 ###################################################################
# With average time 8.24 minutes, probability is 0.776
# For loop, runif() for 0 to 1. Count if it is between 0 and 0.776.
# Initialize 0 crews qualified at start of day
qualified = 0
# Initialize 0 iterations have gone through at start of day
iterations = 0
# Probability of failure/additional repetition
p=1-prob.1
# Initialize data frame to record data
record = data.frame(iterations=0,probability=0,qualified=qualified)
# While the number of qualified crews is less than 7 continue to randomize a
# probability that they qualify assuming they have the same amount of train time
while (qualified<57){
  probability = runif(1)
  if (probability<p){qualified=qualified}else{qualified=qualified+1}
  iterations=iterations+1
  record = rbind(record,data.frame(iterations=iterations,
                                   probability=probability,
                                   qualified=qualified))
  }
print(record)
```

```
> print(record)
   iterations probability qualified
1           0 0.000000000         0
2           1 0.844122630         1
3           2 0.502823933         2
4           3 0.391525650         3
5           4 0.670863153         4
6           5 0.554064189         5
7           6 0.307477539         6
8           7 0.834640258         7
9           8 0.635306600         8
10          9 0.573431367         9
11         10 0.883152452        10
12         11 0.346233611        11
13         12 0.079647981        11
14         13 0.446145688        12
15         14 0.422817030        13
16         15 0.160340504        13
17         16 0.896748103        14
18         17 0.252497838        15

60         59 0.322634557        51
61         60 0.073471148        51
62         61 0.546092081        52
63         62 0.929471402        53
64         63 0.192337859        53
65         64 0.175140500        53
66         65 0.949621053        54
67         66 0.762678753        55
68         67 0.130509660        55
69         68 0.351947702        56
70         69 0.356179848        57
```

APPENDIX D

```
# Problem 4 #################################################################
summary(navigate.model)
# Training time is 2.5 hours
# Average navigating time is 8.24 minutes
# 69 repetitions are predicted to qualify 95% of soldiers
# Time to reach 95% qualified
nav.time*69 # 568.4223 minutes

# Checking the Schedule
# Available minutes in the day
(12*60) # 720
# Minutes remaining after training
720-(60*2.5) # 570
# Minutes remaining after training everyone
570-568.4223 # 1.5777

# Verify with a simulation
# Initialize 0 crews qualified at start of day
qualified = 0
# Initialize 0 iterations have gone through at start of day
iterations = 0
# Probability of failure/additional repetition
p=1-prob.1
# Time Left
time.left = (12-2.5)*60
# Initialize data frame to record data
record = data.frame(iterations=0,probability=0,qualified=qualified,time.left=time.left)
# While the number of qualified crews is less than 7 continue to randomize a
# probability that they qualify assuming they have the same amount of train time
while (qualified<57){
  probability = runif(1)
  if (probability<p){qualified=qualified}else{qualified=qualified+1}
  iterations=iterations+1
  time.left = time.left - nav.time
  record = rbind(record,data.frame(iterations=iterations,
                                   probability=probability,
                                   qualified=qualified,
                                   time.left=time.left))
}
print(record)
```

APPENDIX E

```
a=matrix(c(-1,1.5,1,-1.5,1,1),nrow=3,byrow = T)
print(a)
b=matrix(c(1,1,1))
print(b)
solve(t(a)%*%a,t(a)%*%b)
```

```
> print(a)
     [,1] [,2]
[1,]   -1  1.5
[2,]    1 -1.5
[3,]    1  1.0
> b=matrix(c(1,1,1))
> print(b)
     [,1]
[1,]    1
[2,]    1
[3,]    1
> solve(t(a)%*%a,t(a)%*%b)
     [,1]
[1,]  0.6
[2,]  0.4
```

WORK CITED

CDT Kvasnak, Margot ‘22 CO F-2. Assistance to the authors, oral and written. CDT Kvasnak suggested using a Markov transition matrix to simulate part 5. 30MAR2020 (via teams comms).

“EnvStats.” *Function | R Documentation*, www.rdocumentation.org/packages/EnvStats/versions/2.3.1/topics/Triangular.

Millard SP (2013). _EnvStats: An R Package for Environmental Statistics_. Springer, New York. ISBN 978-1-4614-8455-4, <URL: http://www.springer.com>.