IDS 702 Final Project: Attacks in WWI/WWII

Derek Wales

10DEC19

Summary:

For this report I was analyzing a data set that contained hundreds of battles dating from the 14th century all the way to the 1990s. My goal was to determine the key factors that would make an attack successful in both World War I and World War II. To do this I used a number of different statistical analysis techniques that included Exploratory Data Analysis (EDA), Critical Thinking, Hierarchical Modeling (for both conflict and location) and ultimately logistic regression. Optimizing on Bayesian Information Criterion or BIC. I chose this because it rewards less having fewer terms while maintaining accuracy. This means its easier for Commanders to implement the model. These steps ultimately led to the model below.

Model:

$$\log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \beta_1 Surprise + \beta_2 Air.Power + \beta_3 Morale + \beta_4 Initiative \ + \beta_5 Leadership$$

where $y_i|x_i \sim$ Bernouilli(π_i)

Introduction:

The United States Army strongly attempts to minimize casualties, because it preserves combat power for future operations and boosts morale. Because of this, the Army commissioned the Concepts and Analysis Agency starting in 1992 began to look at more than 600 battles across several hundred years to see if there were empirical relationships between many of the factors in battle.

The research team looked at more than 200 different factors from tangible things like how many troops each side had at the start of the battle to intangible things like the organization's morale.

The original document compared all these factors together for a period spanning more than 300 years.

For my project, I took this data and focused on attacks in World War I and World War II to determine what key factors increased the chances of winning in battle.

Data and Exploratory Data Analysis:

The first task for this project was understanding the dataset from the study and then putting it into a manageable format that could be modeled. To do this I used several different tools and techniques prior to putting the data into R. The first step was getting a scope of the data in the original Army study. This was done by first reading the data into Python where I discovered that there were more than 200 different predictor values. I then pared my data down by selecting only World War I and II battles and began to select meaningful variables.

For variable selection I used my domain knowledge/statistic techniques to eliminate variables that were either highly correlated or the same (such as the width of the attacking army and the width of the defending army's defense because they were usually the same in both World War I and II) and combining things like the defending terrain and attacking terrain. I also removed non-applicable values for the modern era, such as the amount of cavalry that both the attackers and the defenders had available.

Additionally, the study considered not only the army that was attacking or defending, but also who the commanders were. But there were not enough occurrences in the dataset for them to be meaningful predictors (most of them appeared only once or twice throughout the dataset). The data was also grouped across dozens

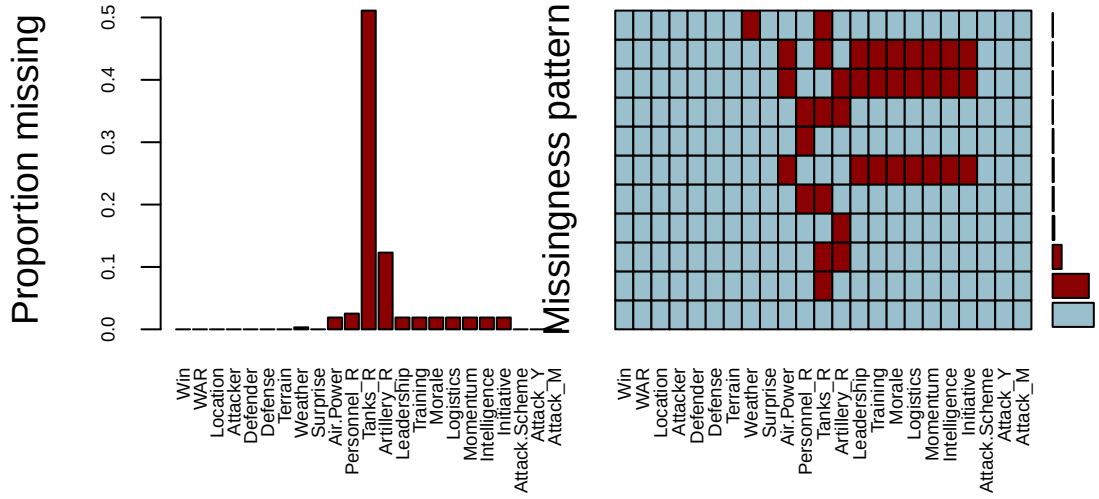
of countries that were often bordering or in the same region. So, I grouped them into the following regions: Europe, North Africa, Middle East, Russian Empire/Soviet Union (WW1/WW2), and the Pacific.

The original data was also organized by unit (aka 2nd Panzer Division, 1st Marine Division, etc.). These were also grouped into their corresponding nation's armies of British, French, American, Japanese, Soviet Union, and German. I aligned the data based purely upon the nation of origin, since this is an excellent proxy for that nation's training program, amount of training and soldier experience, weapon types and sophistication, officer quality, experience and tactical acumen, and overall logistic support. The military also considered the weather and terrain when conducting the study when analyzing the results of the battles. So, I renamed these variables to be more descriptive, and grouped the multiple columns into combined discriptors (things like sunny and hot or overcast and raining).

Next, I went into feature engineering. For this data the Army not only looked at how many troops were on the attacking side and defending side at the start of the battle, but also how many troops they had in reserves (not committed to any particular aspect of the battle) and how much each side received in reinforcements during the battle. Looking at this I realized what was important was not necessarily the quantity of troops but the ratio of troops at the start of the battle (having 10,000 troops doesn't matter if the other guy also has 10,000 troops, but it does really matter if the other guy only has 1,000 troops). I did the same thing for the equipment each side had as well (I converted the number of tanks and artillery into ratios indexed off the attacker instead of raw numbers).

With all of this complete I had a much more manageable dataset going from 208 predictor variables to 22. These variables were all indexed off of the Attacker and on a scale -3 to 3 (negative numbers were advantages to the defenders and positive to the attacker with neutral being 0) ratios attacker/defender, or named factor variables. The final variable list was: War, Location, Attacker, Defender, Defense, Terrain, Weather, Surprise (-3 to 3), Air_Power (-3 to 3), Personnel_Ratio, Tanks_Ratio, Artillery_Ratio, Leadership (-3 to 3), Training (-3 to 3), Morale (-3 to 3), Logistics (-3 to 3), Momentum (-3 to 3, your success in recent operations), Intelligence (-3 to 3, how much you know about the enemy in terms of strength and location), Initiative (-3 to 3, wheter you or the enemy is determining the events going on in the battle), Attack_Scheme (frontal attack, flank attack, etc), Attack_Year, Attack_Month.

I then was able to read these variables into R. Upon initial inspection I had a wide spread in terms of the amount of missing data. Some categories such tanks ratio had 50% of values missing/unknown (because even though they were used in WWI it is not well documented).



With the missing data imputed I could then move into the Exploratory Data Analysis.

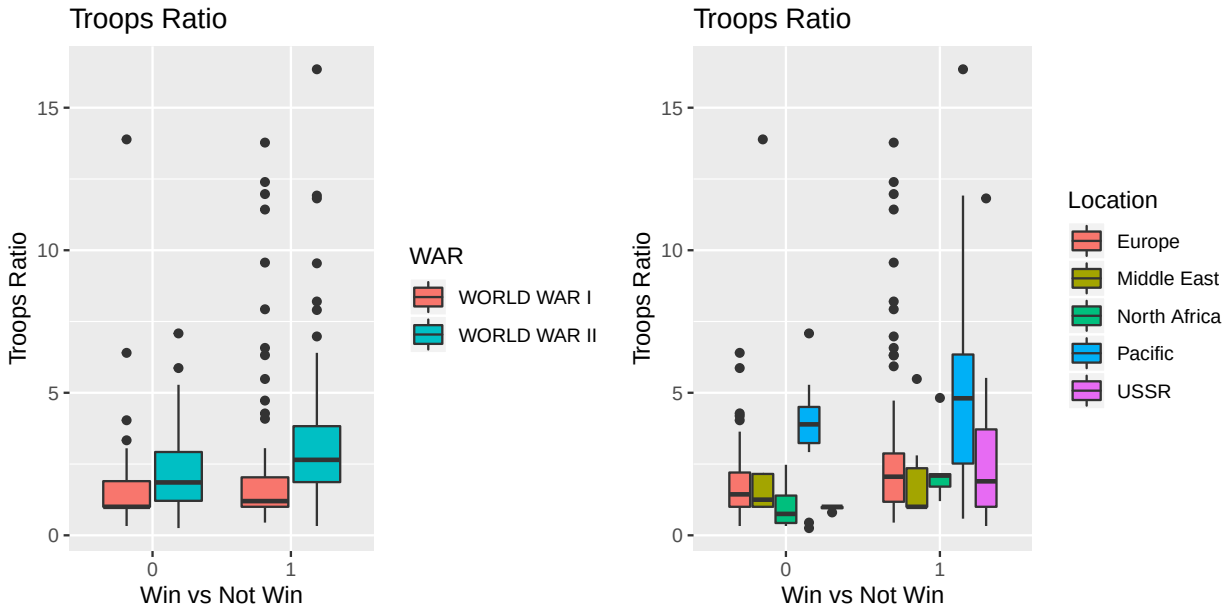
I started with the assumption that this would make a hierarchal model with differing intercepts based upon WW1 and WW2.

I started my analysis by doing the Chi Squared test on many of the predictor variables. Although many of the obvious things did initially seem to be significant. The p values were lower for many of the intangible characteristics.

As I continued going through the data some of the results were not what I was expecting. There was not very much difference between the two conflicts. In military history circles, the trench warfare of WW1 was considered radically different from the Blitzkrieg or maneuver warfare of WW2, but this proved not to be the case from the standpoint of which variables determined success. This was validated when I built the initial model and there was no difference in the variance of the two models. With that in mind I then decided to look at a varying intercepts model based upon the location where the battle took place instead of the conflict itself.

Although slightly more informative than the conflict, including random intercepts by location proved to have little effect on the final model.

Examples of EDA from the Two Initial Modeling Approaches



But what was insightful is that many of the intangibles such as surprise, leadership, and initiative were even stronger predictors than how many troops each unit had. In military terms, they are significant force multipliers. They have an outsized impact compared to their raw economic cost.

This is significant because it means that enhanced planning to achieve initiative and surprise can be more important than how much is spent on weapons systems or soldiers. As a corollary, there may be technologies that enhance surprise and initiative that are relatively low in cost that represent an excellent investment (such as recon planes in WWII or night vision in the current era that allow you to detect the enemy before they detect you).

Model Selection Process and Assessment:

As mentioned earlier, I was trying to build a model based upon BIC, and I had assumed a hierarchal model would give me better results. First starting with varying intercepts by conflict, then with varying intercepts by location. Before ultimately discovering that standard logistic regression preformed the best.

My first attempt used the following model to start and then I progressively removed terms to reach a better BIC:

model_0 <- glmer(Win ~ (1|WAR) + Location + Attacker + Defender + Defense + Terrain + Weather + Surprise + Air.Power + Personnel_R + Tanks_R + Artillery_R + Leadership + Training + Morale + Logistics + Momentum + Intelligence + Initiative + Attack_Scheme, data = war_df, family = binomial)

My initial indicator that the hierarchal model was not necessary was the variance and standard deviation were both zero. After some additional analysis I repeated some EDA on the various variables and decided to determine if a varying intercepts model based upon location.

For the second model I ran the code below:

model_1 <- glmer(Win ~ (1|Location) + WAR + Attacker + Defender + Defense + Terrain + Weather + Surprise + Air.Power + Personnel_R + Tanks_R + Artillery_R + Leadership + Training + Morale + Logistics + Momentum + Intelligence + Initiative + Attack_Scheme, data = war_df, family = binomial)

During this I discovered that only these predictors were significant in determining whether an attack would be successful:

DefenderBritish Army, DefenderFrench Army, DefenseFortified Defense, Surprise, Air.Power, Artillery_R, Leadership, Training, Initiative, Attack_Schemefrontal_attack, and Attack_Schemeriver_crossing

I then began to use various combinations and interaction terms to get as low of a BIC as possible. But I noticed the standard deviation and variance between the five locations were 4e-14 and 2e-07. Because of this I decided to explore whether or not a standard logistic regression would perform better.

Again repeating EDA I ran stepwise on a standard logistic regression with all 22 variables, in addition to several other tests to find the model listed below with the lowest BIC of 353.97, lower than both hierarchal models.

Final Model

$$\log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \beta_1 Surprise + \beta_2 Air.Power + \beta_3 Morale + \beta_4 Initiative \ + \beta_5 Leadership$$

where $y_i|x_i \sim$ Bernouilli(π_i)

Interpretation/Coefficients (Exponentiated)

Intercept	Surprise	Air.Power	Morale	Initiative	Leadership
0.3408395	1.8745726	1.9926527	2.2902150	3.0185522	2.7752378

What this means that you start with a 34% chance of winning if Surprise, Air.Power, Morale, Initiative, and Leadership are at parity with the defender (recall it was -3 to 3 indexed off of the attacker, with zero being both sides are equal). Meaning that you can more than triple your odds of winning by having the initiative.

Model Performance

With the odds listed above the model performed well, especially considering the model was on something as chaotic as a battlefield. Maintaining a 75.4% accuracy rating (guessing a true positive or a true negative) an 0.823 ROC, and a 0.823 sensitivity (meaning it had high chance of predicting a true positive vs a false negative). The model did not perform as well with true negatives, maintaining 0.648 for specificity. What this means in application is Commanders on the Battlefield can use this model to reliably determine if their attack will be successful, but it will not be as reliable in determining if their attack will be unsuccessful.

An additional note about the data is that since Troops and Equipment levels were generally similar, this means effects of having a significant numbers advantage over the enemy could not captured in the model.

Conclusion/Remarks:

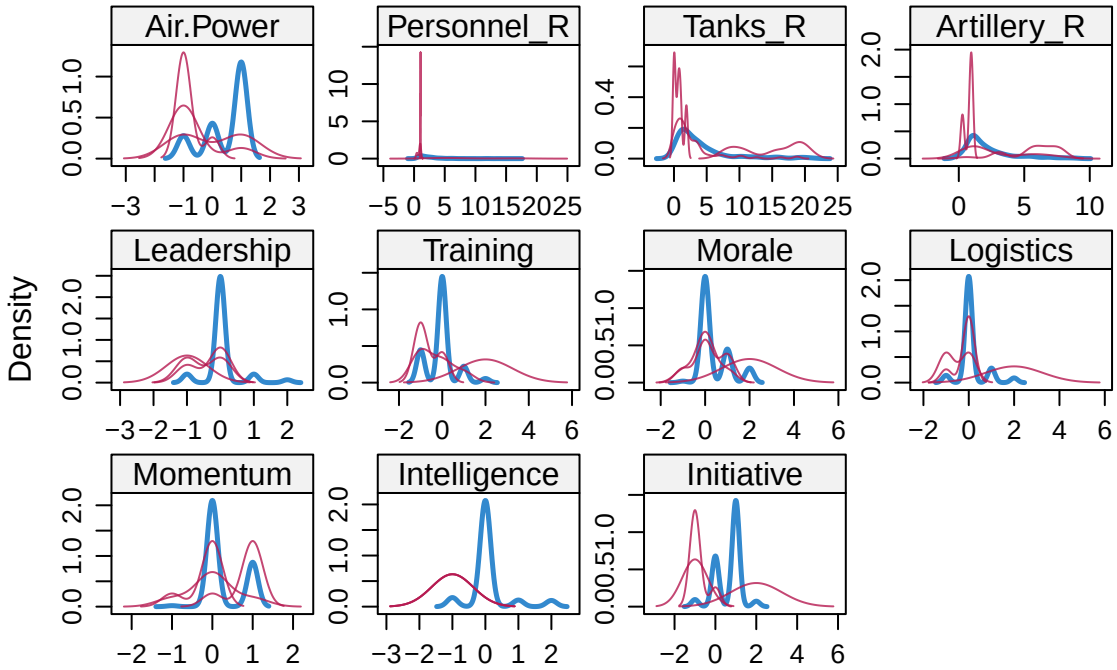
In completing this project the model shows that intangible factors are just as important as tangible ones. Specifically, surprise, leadership, morale, and initiative had an outsized effect on victory. This holds true for both the first and second world wars.

Future Research

For future research it would be interesting to see how accurate the model is on other conflicts later in the 20th century to test its applicability to the modern battlefield.

Appendix: R Code

```
densityplot(war_df_imp)
```




```
chisq.test(table(war_df[,c("Attacker","Win")]))          #p-value = 0.009699
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(war_df[, c("Attacker", "Win")])
## X-squared = 21.752, df = 9, p-value = 0.009699
```

```
chisq.test(table(war_df[,c("Defender","Win")]))          #p-value = 0.006987
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(war_df[, c("Defender", "Win")])
## X-squared = 13.123, df = 10, p-value = 0.2169
```

```
chisq.test(table(war_df[,c("Defense","Win")]))           #p-value = 0.06264
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(war_df[, c("Defense", "Win")])
## X-squared = 10.484, df = 5, p-value = 0.06264
```

```
chisq.test(table(war_df[,c("Terrain","Win")]))           #p-value = 0.08479
```

```
##
##  Pearson's Chi-squared test
```

```
##
## data:  table(war_df[, c("Terrain", "Win")])
## X-squared = 16.556, df = 10, p-value = 0.08479
```

```
chisq.test(table(war_df[,c("Weather","Win")]))          #p-value = 0.2219
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(war_df[, c("Weather", "Win")])
## X-squared = 13.031, df = 10, p-value = 0.2219
```

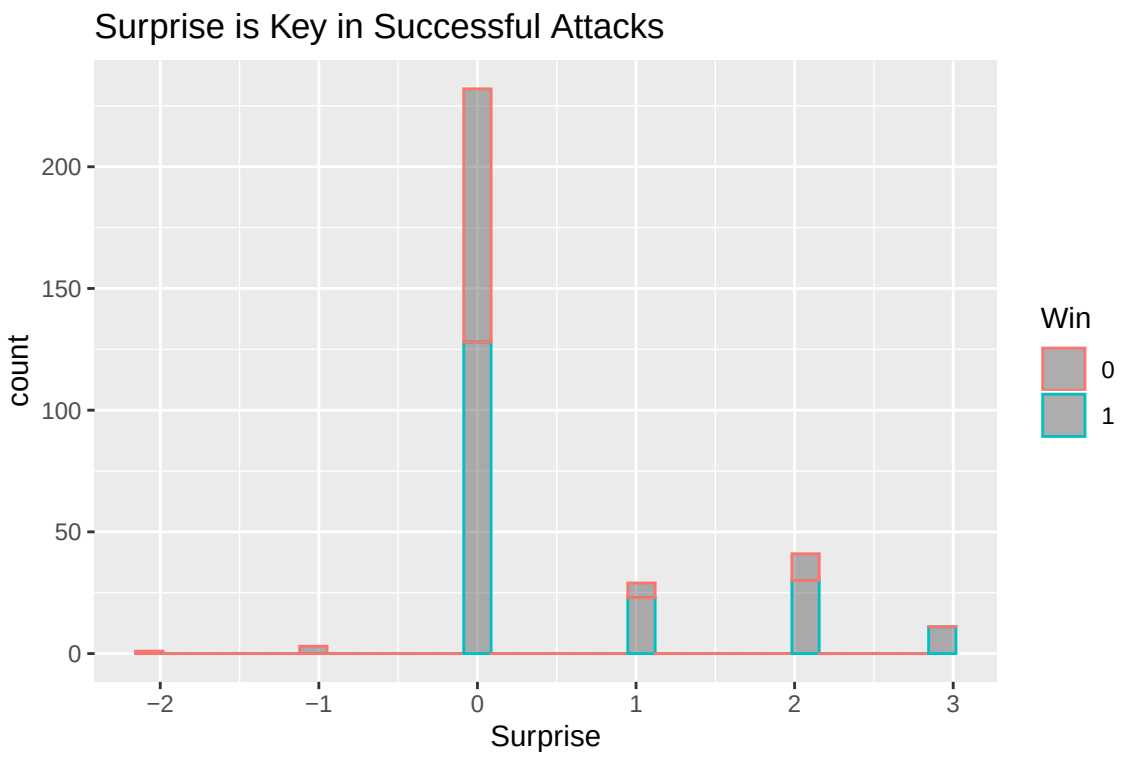
```
chisq.test(table(war_df[,c("Attack_Scheme","Win")]))    #p-value = 0.4603
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(war_df[, c("Attack_Scheme", "Win")])
## X-squared = 3.6171, df = 4, p-value = 0.4603
```

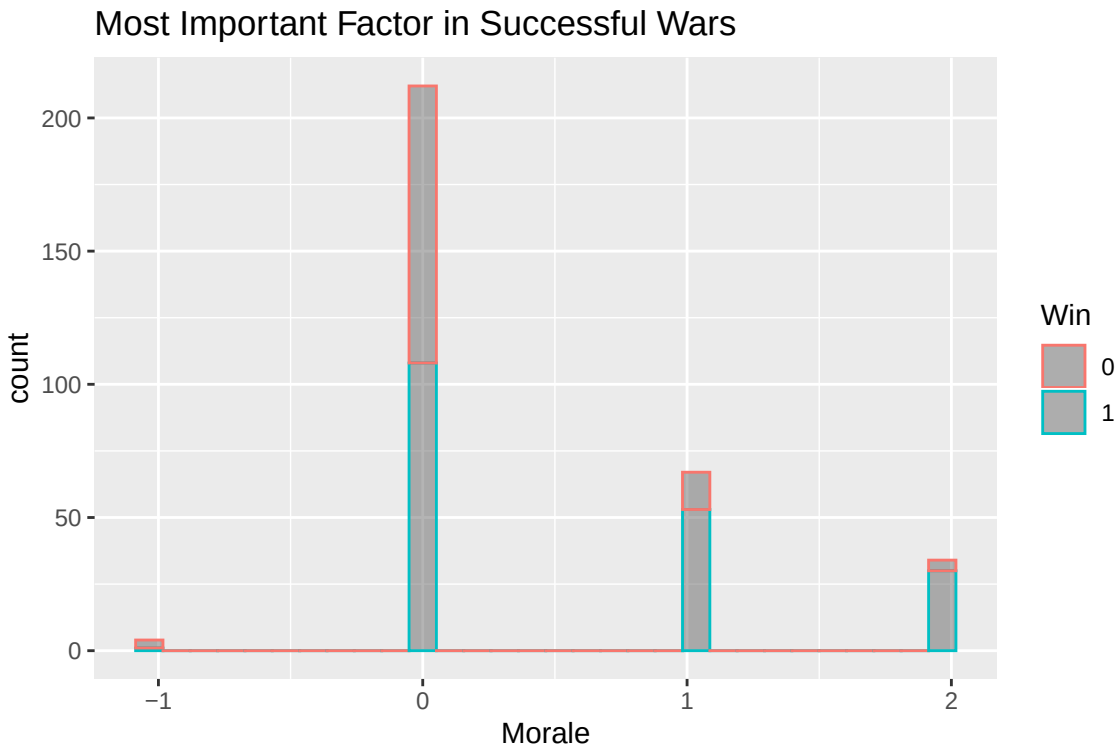
```
chisq.test(table(war_df[,c("Surprise","Win")]))         #p-value = 0.0003194 ### Mention something
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(war_df[, c("Surprise", "Win")])
## X-squared = 23.125, df = 5, p-value = 0.0003194
```

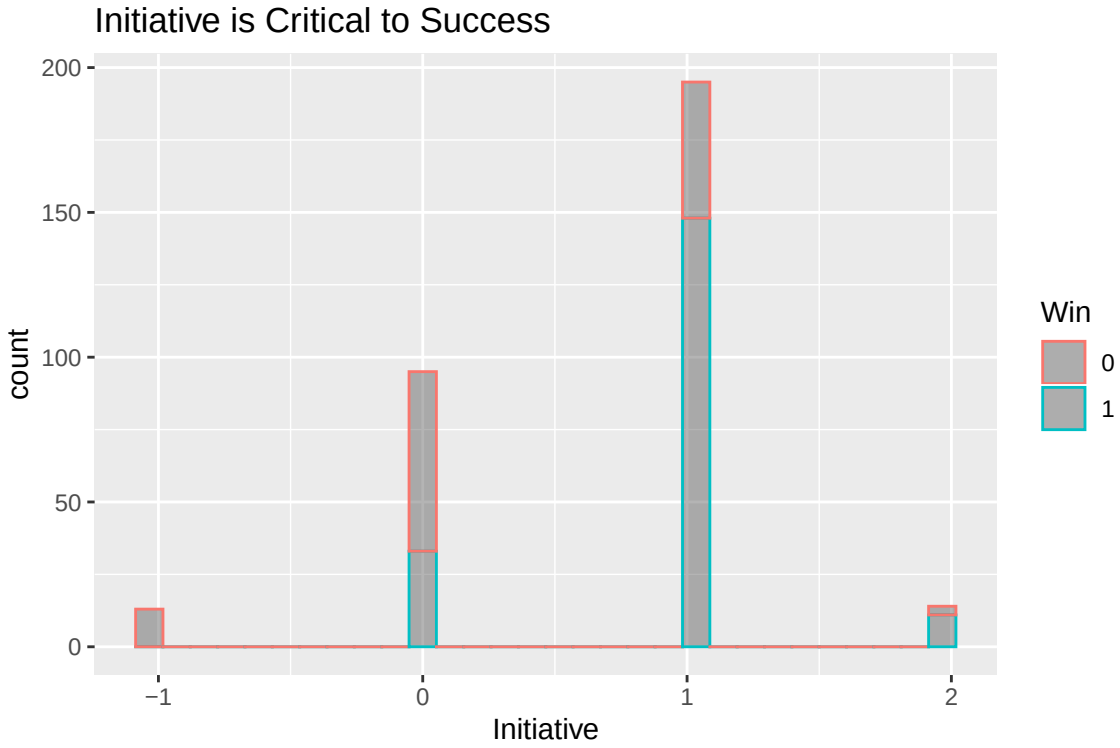
```
ggplot(war_df, aes(x= Surprise, color =  Win)) +  geom_histogram(alpha =.45) + ggtitle('Surprise is Key
```



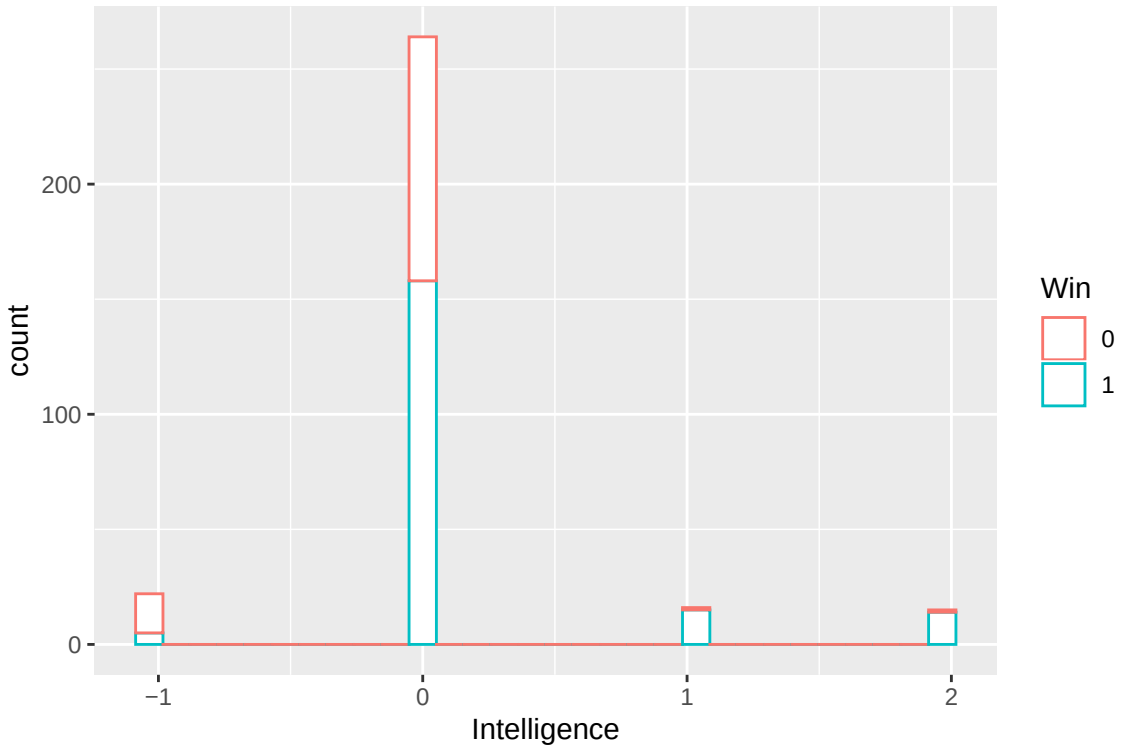

```
# Morale - Morale is the greatest single factor in successful war.
ggplot(war_df, aes(x= Morale, color = Win)) +  geom_histogram(alpha = .45) + ggtitle('Most Important Fa
```




```
# Initiative Army Doctrinal Publication 3-0 -> Seize, retain, and exploit the initiative
# The power of making our adversary's movements conform. to our own.
ggplot(war_df, aes(x= Initiative, color = Win)) +  geom_histogram(alpha = .45) + ggtitle('Initiative is
```



```
# Intelligence
ggplot(war_df, aes(x= Intelligence, color = Win)) +  geom_histogram(fill = "white")
```

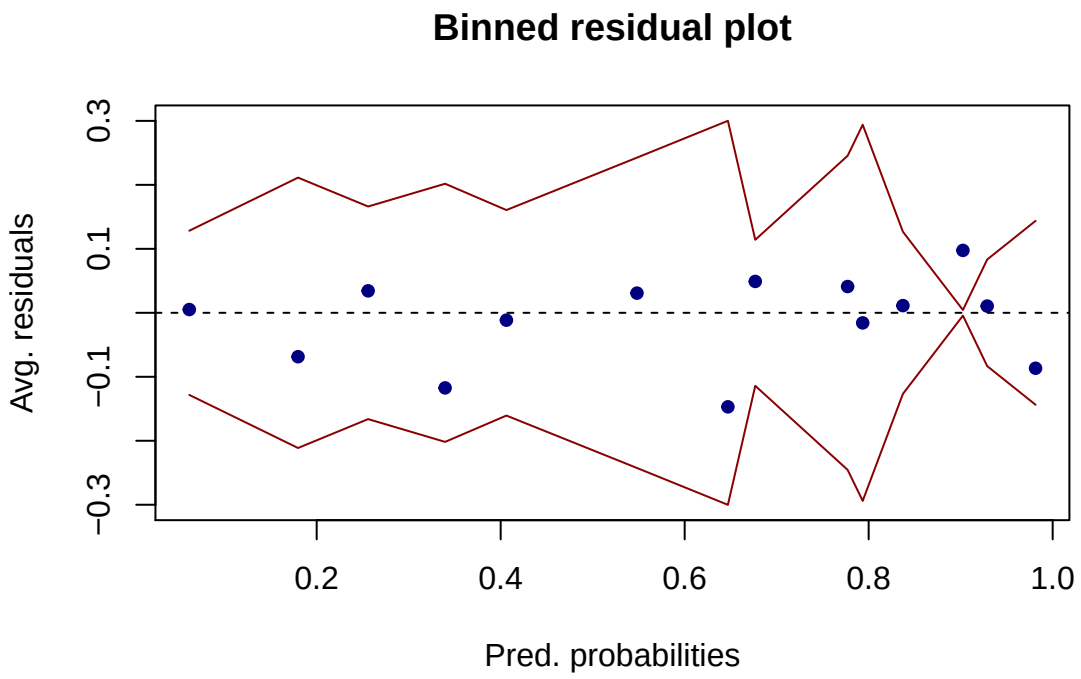



```
# Doing Stepwise on the main model
full_model <- glm(Win  ~ Location + WAR + Attacker + Defender + Defense + Terrain + Weather + Surprise +

#BIC_both <- step(full_model, direction = "backward", trace = 0)
#BIC(BIC_both)

#BIC_backward$call

# ## Building a Model with War as the intercept.
# model_1 <- glmer(Win  ~ (1|Location) + WAR + Attacker + Defender + Defense + Terrain + Weather
#                  + Surprise + Air.Power + Personnel_R + Tanks_R + Artillery_R + Leadership
#                  + Training + Morale + Logistics + Momentum + Intelligence + Initiative
#                  + Attack_Scheme, data = war_df, family = binomial)
# summary(model_1)

#Let's add a random slope
# mercreglmerintslope <- lmer(mercury ~ length_c + ( 1 + length_c  | station), data = bass)
# summary(mercreglmerintslope)

# # exp(model_final)
# # BIC(model_final)
#
# # model_final2 <- glm(Win ~  Surprise + Air.Power + Morale + Initiative + Leadership,
# #                  family = binomial,  data = war_df)
# # BIC(model_final2)
# #Accuracy/Sensitivity/Specificity
```

```r
# Conf_mat <- confusionMatrix(as.factor(ifelse(fitted(model_final2) >= .5, "1","0")),
#                             war_df$Win,positive = "1")
# Conf_mat$table
# Conf_mat$overall["Accuracy"];
# Conf_mat$byClass[c("Sensitivity","Specificity")]
# # ROC
# roc(war_df$Win,fitted(model_final2),plot=T,print.thres=.5,
#     legacy.axes=T, print.auc = T, print.auc.y = .4, col="red3")


#### Here you did stuff ###

model_final <- glm(Win ~  Surprise + Air.Power + Morale + Initiative + Leadership,
                family = binomial,  data = war_df)

### Model Validation/ROC Curve ###
exp(confint(model_final, level = 0.95))
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2101379 0.5364017
## Surprise    1.2805890 2.8492328
## Air.Power   1.3672018 2.9529506
## Morale      1.3967619 3.9518092
## Initiative  1.7803188 5.2120045
## Leadership  1.1329421 7.9315341
```

```r
# Model Validation Code
# Testing to see if we have any residuals
rawresid1 <- residuals(model_final,"resp")

#binned residual plot
binnedplot(x=fitted(model_final),y=rawresid1,xlab="Pred. probabilities",
           col.int="red4",ylab="Avg. residuals",main="Binned residual plot",col.pts="navy")
```




```
#binned residual plot
# binnedplot(x=fitted(model_final),y=rawresid1,xlab="Pred. probabilities",
#            col.int="red4",ylab="Avg. residuals",main="Binned residual plot",col.pts="navy")

#Accuracy/Sensitivity/Specificity
Conf_mat <- confusionMatrix(as.factor(ifelse(fitted(model_final) >= .5, "1","0")),
                            war_df$Win,positive = "1")
Conf_mat$table
```

```
##           Reference
## Prediction   0   1
##          0  81  34
##          1  44 158
```

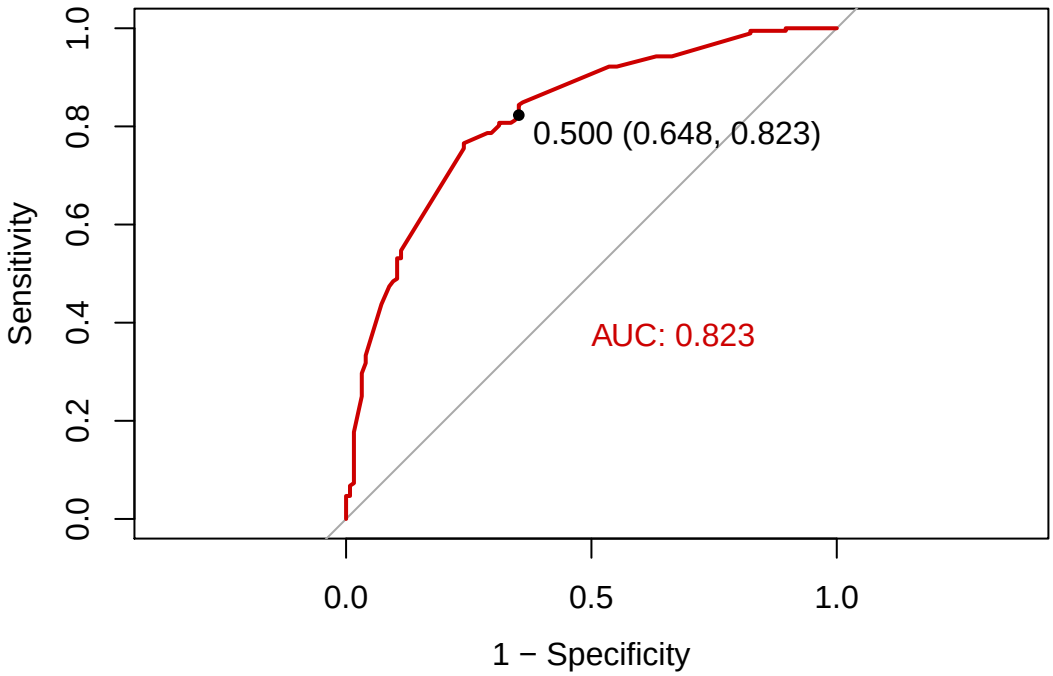
```
Conf_mat$overall["Accuracy"];
```

```
##  Accuracy
## 0.7539432
```

```
Conf_mat$byClass[c("Sensitivity","Specificity")]
```

```
## Sensitivity Specificity
##   0.8229167   0.6480000
```

```
# ROC
roc(war_df$Win,fitted(model_final),plot=T,print.thres=.5,
    legacy.axes=T, print.auc = T, print.auc.y = .4, col="red3")
```




```
##
## Call:
## roc.default(response = war_df$Win, predictor = fitted(model_final),     plot = T, print.thres = 0.5,
##
## Data: fitted(model_final) in 125 controls (war_df$Win 0) < 192 cases (war_df$Win 1).
## Area under the curve: 0.8227
```

In [7]:
```
import pandas as pd
import numpy as np
```

In [8]:
```
working_df = pd.read_csv('CDB90-patched.csv')
```

In [9]:
```
working_df.describe()
```

Out[9]:

	ISQNO	WOFA1	WOFD1	YR1	MO1	DA1	HR1
count	660.000000	660.000000	660.000000	660.000000	660.000000	660.000000	660.000000
mean	330.500000	24.218970	23.461212	9568.301515	94.096970	94.500000	9837.730303
std	190.669872	72.742631	72.714726	1821.495814	20.747115	19.129659	1186.861809
min	1.000000	-1.000000	-1.000000	1632.000000	1.000000	1.000000	530.000000
25%	165.750000	2.400000	2.000000	9999.000000	99.000000	99.000000	9999.000000
50%	330.500000	6.000000	5.600000	9999.000000	99.000000	99.000000	9999.000000
75%	495.250000	15.000000	14.000000	9999.000000	99.000000	99.000000	9999.000000
max	660.000000	1060.000000	1060.000000	9999.000000	99.000000	99.000000	9999.000000

8 rows × 175 columns

In [10]:
```
len(working_df.columns.to_list())
```

Out[10]: 208

Selecting WWI and WWII

In [11]:
```
war_df = working_df.loc[(working_df['WAR'] == 'WORLD WAR II')|
                        (working_df['WAR'] == 'WORLD WAR II (EASTERN FRONT)')
|
                        (working_df['WAR'] == 'WORLD WAR II (NORTH AFRICA 1942
-1943)')|
                        (working_df['WAR'] == 'WORLD WAR II (ITALY 1943-1944)'
)|
                        (working_df['WAR'] == 'WORLD WAR II (ITALY 1944)')|
                        (working_df['WAR'] == 'WORLD WAR II (EUROPEAN THEATE
R)')|
                        (working_df['WAR'] == 'WORLD WAR II (OKINAWA)')|
                        (working_df['WAR'] == 'WORLD WAR I (WESTERN FRONT 191
4)')|
                        (working_df['WAR'] == 'WORLD WAR I (EASTERN FRONT 191
4)')|
                        (working_df['WAR'] == 'WORLD WAR I (SERBIAN FRONT 191
4)')|
                        (working_df['WAR'] == 'WORLD WAR I')|
                        (working_df['WAR'] == 'WORLD WAR I (ITALIAN FRONT 191
5)')|
                        (working_df['WAR'] == 'WORLD WAR I (TURKISH FRONTS 191
5)')|
                        (working_df['WAR'] == 'WORLD WAR I (TURKISH FRONTS 191
7)')|
                        (working_df['WAR'] == 'WORLD WAR I (WESTERN FRONT 191
8)')]
```

In [12]:
```
war_df2 = war_df.copy()
```

In [13]:
```
war_df3 = war_df2.loc[:, ['ISEQNO', 'WINA', 'WAR','NAME', 'LOCN', 'CAMPAGN',
'NAMA', 'COA','NAMD','COD', 'WOFA1', 'WOFD1','POST1','TERRA1', 'WX1', 'SURPA',
'AEROA', 'INTSTA', 'RERPA', 'CASA',
                          'INTSTD', 'RERPD','CASD', 'TANKA', 'ARTYA', 'TANKD',
'ARTYD', 'LEADA', 'TRNGA','MORALA', 'LOGSA','MOMNTA', 'INTELA', 'INITA', 'ACH
A', 'PRIA1', 'ATPBYR1', 'ATPBMN1', 'ATPEYR1', 'ATPEMN1']]
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py:1418: Futu
reWarning:
Passing list-likes to .loc or [] with any missing label will raise
KeyError in the future, you can use .reindex() as an alternative.

See the documentation here:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#depreca
te-loc-reindex-listlike
  return self._getitem_tuple(key)
```

Reset Index

In [14]:
```
war_df4 = war_df3
war_df4.reset_index(inplace=True)
# del war_df4['ISEQNO']
# del war_df['level_0']
war_df4.head(2)
```

Out[14]:

	index	ISEQNO	WINA	WAR	NAME	LOCN	CAMPAGN	NAMA	COA	
0	261	NaN	-1	WORLD WAR I (WESTERN FRONT 1914)	ALSACE-LORRAINE I	GERMANY	NaN	FR 1ST & 2ND ARMIES, ARMY OF ALSACE	GEN JOFFRE	AI
1	262	NaN	1	WORLD WAR I (WESTERN FRONT 1914)	ALSACE-LORRAINE II	GERMANY AND FRANCE	NaN	GER 6TH & 7TH ARMIES	GEN MOLTKE	F AI & AL

2 rows × 41 columns

In [15]:
```
war_df4['POST1'] = war_df4['POST1'].replace('HD', "Hasty Defense")
war_df4['POST1'] = war_df4['POST1'].replace('PD', "Prepared Defense")
war_df4['POST1'] = war_df4['POST1'].replace('FD', "Fortified Defense")
war_df4['POST1'] = war_df4['POST1'].replace('DL', "Delay")
war_df4['POST1'] = war_df4['POST1'].replace('WD', "Withdrawal")
war_df4['POST1'] = war_df4['POST1'].replace('OO', "Unknown")
war_df4.head(2)
```

Out[15]:

	index	ISEQNO	WINA	WAR	NAME	LOCN	CAMPAGN	NAMA	COA	
0	261	NaN	-1	WORLD WAR I (WESTERN FRONT 1914)	ALSACE-LORRAINE I	GERMANY	NaN	FR 1ST & 2ND ARMIES, ARMY OF ALSACE	GEN JOFFRE	AI
1	262	NaN	1	WORLD WAR I (WESTERN FRONT 1914)	ALSACE-LORRAINE II	GERMANY AND FRANCE	NaN	GER 6TH & 7TH ARMIES	GEN MOLTKE	F AI & AL

2 rows × 41 columns

In [16]:
```
war_df5 = war_df4.copy()
```

In [17]:
```
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II', 'WORLD WAR II')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II (EASTERN FRONT)', 'WORLD
WAR II')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II (NORTH AFRICA 1942-194
3)', 'WORLD WAR II')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II (ITALY 1943-1944)', 'WOR
LD WAR II')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II (ITALY 1944)', 'WORLD WA
R II')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II (EUROPEAN THEATER)', 'WO
RLD WAR II')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR II (OKINAWA)', 'WORLD WAR I
I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (WESTERN FRONT 1914)', 'W
ORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (EASTERN FRONT 1914)', 'W
ORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (SERBIAN FRONT 1914)', 'W
ORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I', 'WORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (ITALIAN FRONT 1915)', 'W
ORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (TURKISH FRONTS 1915)',
'WORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (TURKISH FRONTS 1917)',
'WORLD WAR I')
war_df5['WAR'] = war_df5['WAR'].replace('WORLD WAR I (WESTERN FRONT 1918)', 'W
ORLD WAR I')
war_df5.head(2)
```

Out[17]:

	index	ISEQNO	WINA	WAR	NAME	LOCN	CAMPAGN	NAMA	COA	NAI
0	261	NaN	-1	WORLD WAR I	ALSACE-LORRAINE I	GERMANY	NaN	FR 1ST & 2ND ARMIES, ARMY OF ALSACE	GEN JOFFRE	G 6 &7 ARMI
1	262	NaN	1	WORLD WAR I	ALSACE-LORRAINE II	GERMANY AND FRANCE	NaN	GER 6TH & 7TH ARMIES	GEN MOLTKE	FR 1 & 2 ARMI & ARI ALSA

2 rows × 41 columns

In [18]:
```
replace_values = {'EGYPT':'North Africa', 'TUNISIA':'North Africa', 'FRANCE':
'Europe', 'GERMANY': 'Europe', 'LUXEMBOURG':'Europe', 'POLAND': 'Europe',
                  'NORTHWEST EUROPE':'Europe', 'BELGIUM':'Europe', 'Soviet Unio
n':'USSR', 'MALAYA':'Pacific', 'JAPAN':'Pacific', 'OKINAWA': 'Pacific'}
```

In [19]:
```
# Grouping Regions ()
war_df6 = war_df5.copy()
war_df6 = war_df6.replace({'LOCN':replace_values})
war_df6['LOCN'] = war_df6['LOCN'].replace('SOVIET UNION', "USSR")
war_df6['LOCN'] = war_df6['LOCN'].replace('MANCHURIA', "Pacific")
war_df6['LOCN'].value_counts()
```

Out[19]:
```
Europe                      142
ITALY                        72
Pacific                      34
USSR                         23
North Africa                 11
AUSTRIA                       9
PALESTINE                     5
AUSTRIAN GALICIA              4
RUSSIAN POLAND                4
AUSTRIA AND ITALY             3
TURKEY                        3
MESOPOTAMIA                   3
SERBIA                        2
GERMANY AND FRANCE            1
GERMANY AND RUSSIAN POLAND    1
Name: LOCN, dtype: int64
```

In [20]:
```
replace_values2 = {'AUSTRIA ':'Europe', 'ITALY':'Europe', 'AUSTRIAN GALICIA':
'Europe', 'RUSSIAN POLAND ': 'USSR', 'AUSTRIA AND ITALY':'Europe',
                   'SERBIA':'Europe','GERMANY AND FRANCE':'Europe', 'GERMANY A
ND RUSSIAN POLAND':'Europe', 'PALESTINE':'Middle East',
                   'MESOPOTAMIA ': 'Middle East', 'TURKEY':'Middle East'}
war_df6 = war_df6.replace({'LOCN':replace_values2})
# war_df6['LOCN'].value_counts()
```

In [21]:
```
del war_df6['COA']
del war_df6['COD']
del war_df6['CAMPAGN']
```

Putting file to excel for editing.

In [22]:
```
len(war_df6.columns.to_list())
```

Out[22]: 38

In [23]:
```
#war_df6.to_excel('War DF1.xlsx')
```

File Back from Excel, converted to ratios and simplified the attackers and defenders

In [24]:
```
df_post_excel = pd.read_excel('War DF1_editing_v4.xlsx')
```

In [25]:
```
df_post_excel.head(2)
```

Out[25]:

	Unnamed: 0	ISEQNO	Win	WAR	NAME	Location	Attacker	Defender	Defense	Terra
0	49	310	1	WORLD WAR I	1916 BRUSILOV OFFENSIVE	Europe	Russian Army	Austrian Army	Prepared Defense	RM
1	52	313	1	WORLD WAR I	TRENTINO COUNTER-OFFENSIVE	Europe	Italian Army	Austrian Army	Hasty Defense	GE

2 rows × 26 columns

Fixing Terrain/Weather Values

In [26]:
```
t_replace_values = {'RM0':'Rolling/Mixed', 'GM0':'Rugged/Mixed', 'FM0':'Flat/Mixed', 'FB0':'Flat/Bare', "GW0":'Rugged/Wooded','GB0':'Rugged/Bare','FD0':'Flat/Desert',
                    'RB0':'Rolling/Bare', 'RW0':'Rolling/Wooded', 'FW0': 'Flat/Wooded','RD0':'Rolling/Desert','R00':'Rugged/Wooded','FWM':'Flat/Wooded', 'Flat/Wooed':'Flat/Wooded'}
df_post_excel_2 = df_post_excel.replace({'Terrain':t_replace_values})
df_post_excel_2['Terrain'].value_counts()
```

Out[26]:
```
Rolling/Mixed     167
Rugged/Mixed       43
Flat/Mixed         42
Rugged/Wooded      17
Rugged/Bare        13
Flat/Bare          12
Flat/Wooded         6
Flat/Desert         5
Rolling/Bare        5
Rolling/Wooded      5
Rolling/Desert      2
Name: Terrain, dtype: int64
```

In [27]:
```
df_post_excel_2.copy()
w_replace_values = {'DST':'Dry/Sunny/Temperate', 'WLT':'Wet/L_Rain/Temperate', 'WLC':'Wet/L_Rain/Cold','DSC':'Dry/Sunny/Cold','WHT':'Wet/H_Rain/Temperate','DOT':'Dry/Overcast/Temperate',
                    'D0C':'Dry/Sunny/Cold', 'WHC':'Wet/H_Rain/Cold','W0C':'Wet/L_Rain/Cold','DOC':'Dry/Overcast/Cold', 'D0T':'Dry/Overcast/Temperate','WHH':'Wet/H_Rain/Hot'}
df_post_excel_2 = df_post_excel_2.replace({'Weather':w_replace_values})
```

In [28]:
```
df_post_excel_2['Weather'].replace('WOT', 'Wet/Overcast/Temperate')
df_post_excel_2['Weather'].replace('000 ', 'Unknown')
df_post_excel_2['Weather'].value_counts()
```

Out[28]:
```
Dry/Sunny/Temperate       169
Wet/L_Rain/Temperate       43
Dry/Overcast/Temperate     33
Wet/L_Rain/Cold            21
Dry/Sunny/Cold             19
Wet/H_Rain/Temperate       17
Wet/H_Rain/Cold             8
Dry/Overcast/Cold           3
WOT                         2
Wet/H_Rain/Hot              1
000                         1
Name: Weather, dtype: int64
```

In [29]:
```
#df_post_excel_2.to_excel('war_df_excel_2.xlsx')
```

In [30]:
```
df_final_v1 = pd.read_excel('war_df_excel_2.xlsx')
```

In [31]:
```
df_final_v1.to_csv('R_war_df.csv')
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: