**Part 1:**

**Dataset Source:** <https://www.kaggle.com/mylesoneill/game-of-thrones>

**Dataset:** Battles\_File.csv Part 1 Tools: Tableau

**Dataset:** Character-Deaths.csv; Part 2 Tools: R Studio

1. **Who has led the most attacks?**

**Ans:** Joffrey/Tommen Baratheon is the most attacking king with highest no. of attacks, 14.

**Process:** To understand the king who lead most attacks, first the **Name** column that represents the name of the battle has to be considered as rows in Tableau Sheet. Then adding **Attacker King** as column to the sheet gives the information about which king has led which battle. Then changing the measure of **Name** to represent the count gives a bar graph visualization about each king and the no. of battles fought.

1. **Who has been attacked the most?**

**Ans:** Robb Stark is the king who has been attacked the most with 14 attacks on him.

**Process:** Consider the **Name** column that represents the name of the battle, has to be considered as rows in Tableau Sheet. Then adding **Defender King** as column to the sheet gives the information about which king has been attacked in which battle. Then changing the measure of **Name** to represent the count gives a bar graph visualization about each king and the count of the battles he was attacked in.

1. **What is the region with second most battles?**

**Ans:** The North region with a total of 10 battles fought.

**Process:** Consider the **Name** column that represents the name of the battle, has to be considered as rows in Tableau Sheet. Then adding **Region** as column to the sheet gives the information about which battle has been bought in which region. Then changing the measure of **Name** to represent the count gives a bar graph visualization about each battle fought in each region.

1. **There are null values in attacker\_size and defender\_size columns. How would you clean these variables and explain your reasoning?**

**Ans:** Variablesattacker\_size has14null values anddefender\_size has 19 null values. Upon checking the variables, it is difficult to interpret the values of the variables. In this case, we can deal with missing values by imputing the values statistically. An approach could be imputing the missing values using an algorithm like decision tree or XGBoost.

**Part 2:**

**Dataset:** Character-Deaths.csv;

**Tools:** R Studio.

1. **Clean up the dataset & walk through the method that you choose for each variable.**

**Exploratory Data Analysis:** The .csv file has 917 observations and 13 variables that was imported into the data frame using read.csv().

**Missing Values:**

Death.Year: 612 values

Book.of.Death: 610 values

Death.Chapter:618 values

Book.Intro.Chapter:12 values

No duplicates found.

**Handling Missing Values:**

**Death Year:** If a value exists in variable Death.Year, that means the character is dead. If there is no value that exists, then there are 2 possibilities:

1. Characters are alive

or

1. Missing data

Hence, if all 3 columns Death.Year, Book of Death and Death Chapter have missing values for the same record, then the character is considered alive. If data exists in some columns and missing in other columns, then the data is considered missing.

It is difficult to predict the missing data given the other values in the dataset.

Hence removed the missing values from the dataset as they constitute 2.5% of the entire dataset.

After removing missing data, the rest of the NAs are converted to 0 as they give more information that the character is alive. This results in increased correlation among the variables and also increases the variance of the data.

**Handling Data Types:** Converted Names and Allegiances non-numeric variables into factors.

**Handling Anomalies:**

**Name:** The variable name has the value Myles that appeared 3 times in the dataset. Given that all 3 of them were introduced in different books and in different chapters of the books, we have considered that them as 3 different people in this context.

1. **Run a Logistic Regression Model to predict which characters would live or die.**

We have added a new column name **Dead** to the dataset which will be our dependent variable to be predicted using the algorithm.

If Dead.Year has a value, then variable Dead is 1(True) and if not, it is 0 (False).

A logistic regression model was fit using glm() to predict if the character would Live or Die. Removing the non-numeric factor variables name and Allegiance.

Also, from the heatmap we see that Dead Year(-1), Book of Death(-0.87) and Death Chapter(-0.85) have high correlation values that result into overfitting and inaccurate prediction. Hence removed those variables from Logistic Regression model.

The Logistic model resulted in a prediction with **accuracy of 69.8%**.

1. **Interpret: How Accurate & What Do Coefficients Mean**

**Coefficients of Logistic Regression model:**

Log(odds) is the ratio of dead vs alive in the Dead variable.

**Positive Coefficients:**

**Intercept:** 0.305124700

When all the independent variables in the Regression are at 0, then the log (odds of Dead) will increase by 0.305124700

**FfC:** On a log scale with 1.721564106 units of increase in the FfC variable, the odds of being Dead also increases accordingly. The standard error for this estimate is 0.225121283. This implies that the variable has a positive impact on the prediction of Dead variable. This is confirmed by the p value (2.052848e-14), which is less than 0.05 significant at 95% confidence interval.

**DwD:** On a log scale with 1.154961997 units of increase in the DwD variable, the odds of being Dead also increases accordingly. The standard error for this estimate is 0.194830772. This is confirmed by the p value (3.065968e-09)which is less than 0.05 significant at 95% confidence interval.

**Nobility:** On a log scale with 0.465043678 units of increase in the Nobility variable, the odds of being Dead also increases accordingly. The standard error for this estimate is 0.160864900. This is confirmed by the p value (3.841454e-03),which is less than 0.05 significant at 95% confidence interval.

FfC 1.721564106 0.225121283 7.64727389 2.052848e-14

DwD 1.154961997 0.194830772 5.92802660 3.065968e-09

Nobility 0.465043678 0.160864900 2.89089588 3.841454e-03

**Negative coefficients:**

**Gender:** On a log scale with -0.5828040 units of decrease in the Gender variable, the odds of being Dead also decreases accordingly. This means, if the value decreases, (is close to 0, in other words if gender is a woman) then the possibility of the character being dead also decreases and vice versa. The standard error for this estimate is 0.2272526. This is confirmed by the p value (0.01033052), which is less than 0.05 significant at 95% confidence interval.

**GOT:** On a log scale with -0.3965539 units of decrease in the GOT variable, the odds of being Dead also decreases accordingly. The standard error for this estimate is 0.1737805. This is confirmed by the p value (0.02249381), which is less than 0.05 significant at 95% confidence interval.

Estimate Std. Error z value Pr(>|z|)

Gender -0.5828040 0.2272526 -2.564565 0.01033052

GoT -0.3965539 0.1737805 -2.281924 0.02249381

1. **Run Another Model Using An Optimization Technique From Last Week (Forward, Backward, PCA), Explain Why You Choose That Method & Compare This Model & The One In Step 2**

**Decision Tree Model:**

Built a Decision Tree model removing the non-numeric variables name, Allegiance and also the variables indicative of multicollinearity Death Year, Death Chapter and Book of Death to avoid overfitting issue.

The Decision Tree thus built resulted in a prediction with accuracy of 75.5%.

**Optimization:**

**Stepwise Regression (forward):**

Using stepwise regression model, we have fit the decision tree model and tried to optimize the model. For this purpose, we started with a null dataset initially. The model thus built resulted in a prediction with accuracy of 69.8% which is same as the Logistic Regression model.

After analyzing the predictions of the models, we proceed with Decision Tree model which has highest accuracy of 75.5%.

> cbind(Logisticaccuracy, DTree\_accuracy, Stepaccuracy)

Logisticaccuracy DTree\_accuracy Stepaccuracy

[1,] 0.6987682 0.7558791 0.6987682