MT5763_Lushingtons

MT5763 - Software for Data Analysis

Lushingtons Group

16 November 2020

Group members:

- Kinjal Patel - 200028225
- James Mackie - 170011168
- Anoni Michal Sieminski - 200024322
- Changwoo Lee - 200030189
- Ekaterina Tsvetkova - 200027567

Github link(click)

Github link for Shiny app(click)

Task 1

Introduction

The app examines the usage of the popular social media platform Twitter by political figures in Poland. The politicians chosen are members of the current Polish parliament, which has a total of 460 seats. Out of the 460 representatives, 383 actively use Twitter, an equivalent of 83% of the MPs. Twitter is becoming a tool for disseminating information worldwide and the prevalent usage of the social media platform among Polish MPs was a topic worth examining. It downloads the tweets via Twitter's API and translates them into English via Google Translate's API. The app updates every two hours.

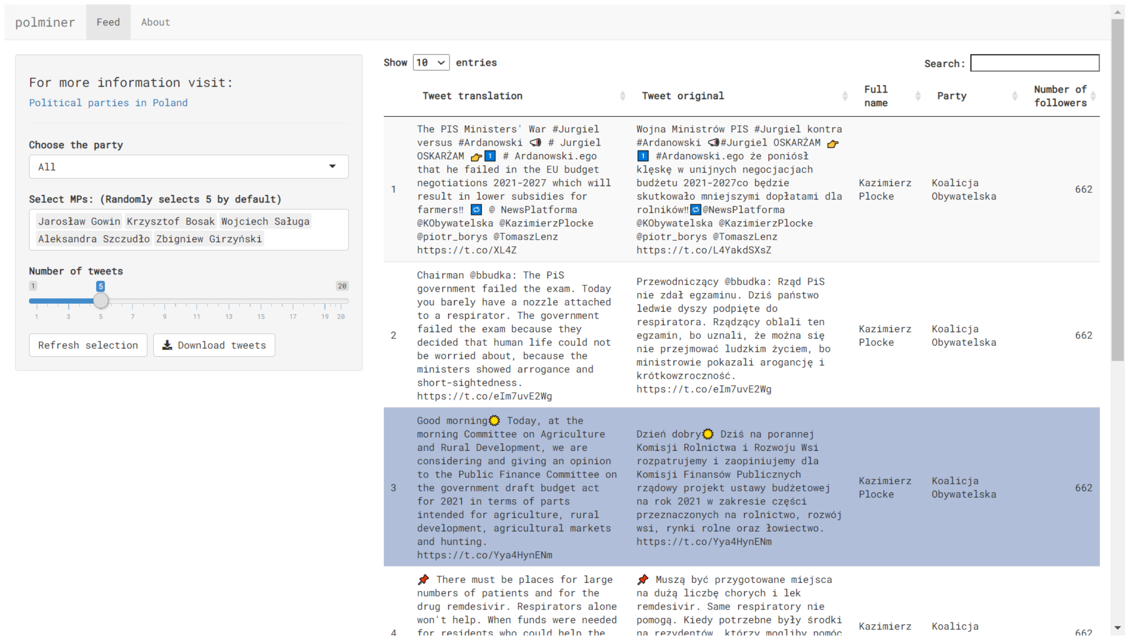
Method

The following methodology delineates the particular steps a user would like to adhere to in order to reproduce the same app in r

First the data stream for all sessions was set up. Afterwards, the app was given a name. The API key and secret as well as the token key and secret were presented as inputs, assigned to their respective objects. An object called **twitter_token**, which contains the aforementioned app name, API key and secret as well as token key and secret, was created using the R package dubbed **rtweet**. Following the token object creation, the Google Natural Machine Learning APIs were authorized totranslate the tweets from Polish to English through the **googleLanguageR** package. Thereafter, a timer, which automatically collects the data every two hours, is set up. The tweets are updated every time the timer is invalidated.

Shiny app description

Figure 1: Layout of Shiny App as seen by the user



For more information visit:
Political parties in Poland

Choose the party: All

Select MPs: (Randomly selects 5 by default): Jarosław Gowin, Krzysztof Bosak, Wojciech Saługa, Aleksandra Szczudło, Zbigniew Girzyński

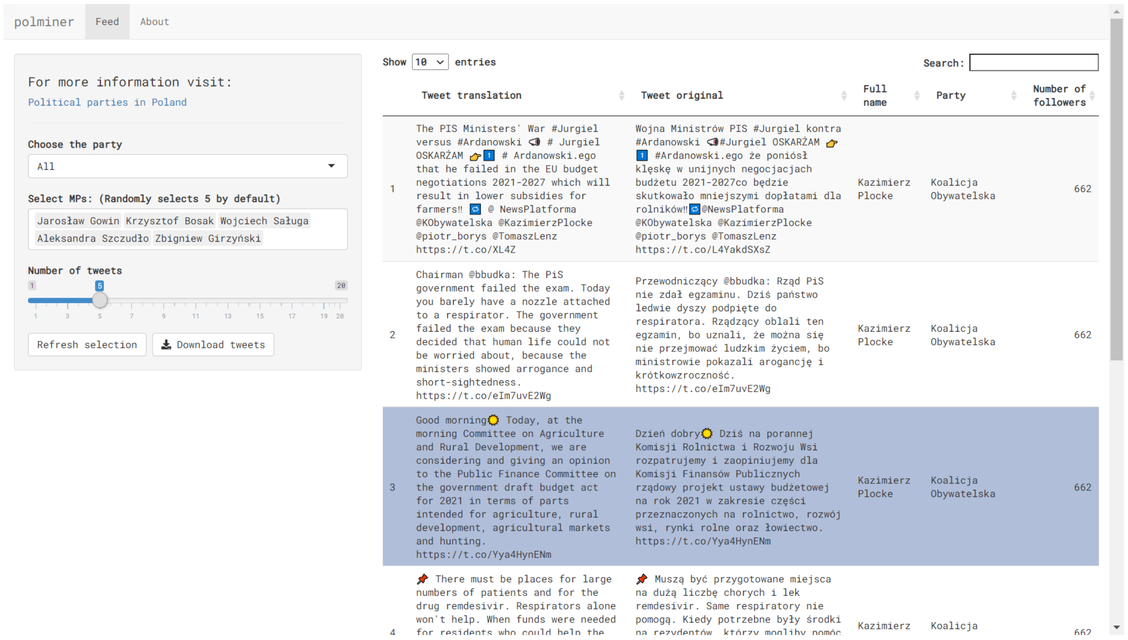
Number of tweets: 5 (range 1–20)

Show 10 entries   Search:

	Tweet translation	Tweet original	Full name	Party	Number of followers
1	The PIS Ministers' War #Jurgiel versus #Ardanowski 📢 # Jurgiel OSKARŻAM 👉1️⃣ # Ardanowski.ego that he failed in the EU budget negotiations 2021-2027 which will result in lower subsidies for farmers‼ 🔄 @ NewsPlatforma @KObywatelska @KazimierzPlocke @piotr_borys @TomaszLenz https://t.co/XL4Z	Wojna Ministrów PIS #Jurgiel kontra #Ardanowski 📢#Jurgiel OSKARŻAM 👉 1️⃣ #Ardanowski.ego że poniósł klęskę w unijnych negocjacjach budżetu 2021-2027co będzie skutkowało mniejszymi dopłatami dla rolników‼🔄@NewsPlatforma @KObywatelska @KazimierzPlocke @piotr_borys @TomaszLenz https://t.co/L4YakdSXsZ	Kazimierz Plocke	Koalicja Obywatelska	662
2	Chairman @bbudka: The PiS government failed the exam. Today you barely have a nozzle attached to a respirator. The government failed the exam because they decided that human life could not be worried about, because the ministers showed arrogance and short-sightedness. https://t.co/eIm7uvE2Wg	Przewodniczący @bbudka: Rząd PiS nie zdał egzaminu. Dziś państwo ledwie dyszy podpięte do respiratora. Rządzący oblali ten egzamin, bo uznali, że można się nie przejmować ludzkim życiem, bo ministrowie pokazali arogancję i krótkowzroczność. https://t.co/eIm7uvE2Wg	Kazimierz Plocke	Koalicja Obywatelska	662
3	Good morning🌞 Today, at the morning Committee on Agriculture and Rural Development, we are considering and giving an opinion to the Public Finance Committee on the government draft budget act for 2021 in terms of parts intended for agriculture, rural development, agricultural markets and hunting. https://t.co/Yya4HynENm	Dzień dobry🌞 Dziś na porannej Komisji Rolnictwa i Rozwoju Wsi rozpatrujemy i zaopiniujemy dla Komisji Finansów Publicznych rządowy projekt ustawy budżetowej na rok 2021 w zakresie części przeznaczonych na rolnictwo, rozwój wsi, rynki rolne oraz łowiectwo. https://t.co/Yya4HynENm	Kazimierz Plocke	Koalicja Obywatelska	662
4	📌 There must be places for large numbers of patients and for the drug remdesivir. Respirators alone won't help. When funds were needed for residents who could help the	📌 Muszą być przygotowane miejsca na dużą liczbę chorych i lek remdesivir. Same respiratory nie pomogą. Kiedy potrzebne były środki na rezydentów, którzy mogliby pomóc	Kazimierz	Koalicja	662

The side panel on the left side of the page displays the various inputs of the app that the user interact with.The very top of the panel displays a link of a page, which provides context for the political scene in Poland and the various political parties, which the user can choose from. The drop-down menu titled **Choose the party** provides the option to select the party of interest among **Prawo i Sprawiedliwosc**, **Koalijia Obywatelska**, **Lewica**, **Koalijica Polska**, and **Konfederacija**.

Figure: Example of generated user set preferences



For more information visit:
Political parties in Poland

Choose the party: All

Select MPs: (Randomly selects 5 by default)
Jarosław Gowin, Krzysztof Bosak, Wojciech Saługa, Aleksandra Szczudło, Zbigniew Girzyński

Number of tweets: 5 (1–20)

Refresh selection | Download tweets

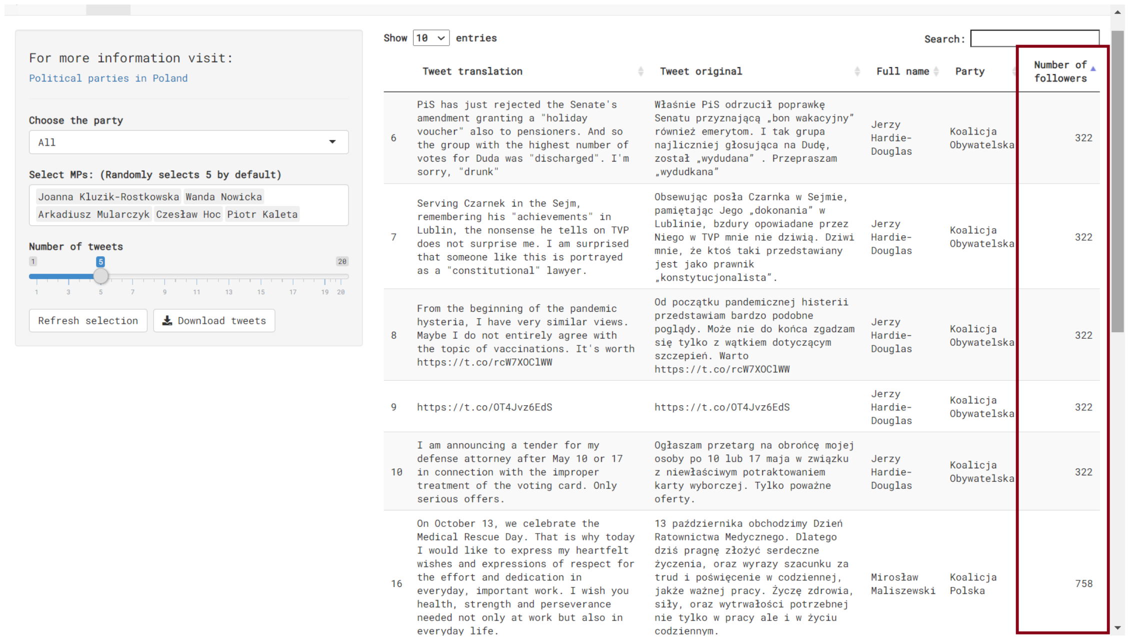
Show 10 entries

Search:

	Tweet translation	Tweet original	Full name	Party	Number of followers
1	The PIS Ministers' War #Jurgiel versus #Ardanowski 📢 # Jurgiel OSKARŻAM 👉1️⃣ # Ardanowski.ego that he failed in the EU budget negotiations 2021-2027 which will result in lower subsidies for farmers‼ 🔁 @ NewsPlatforma @KObywatelska @KazimierzPlocke @piotr_borys @TomaszLenz https://t.co/XL4Z	Wojna Ministrów PIS #Jurgiel kontra #Ardanowski 📢#Jurgiel OSKARŻAM 👉 1️⃣ #Ardanowski.ego że poniósł klęskę w unijnych negocjacjach budżetu 2021-2027co będzie skutkowało mniejszymi dopłatami dla rolników‼🔁@NewsPlatforma @KObywatelska @KazimierzPlocke @piotr_borys @TomaszLenz https://t.co/L4YakdSXsZ	Kazimierz Plocke	Koalicja Obywatelska	662
2	Chairman @bbudka: The PiS government failed the exam. Today you barely have a nozzle attached to a respirator. The government failed the exam because they decided that human life could not be worried about, because the ministers showed arrogance and short-sightedness. https://t.co/eIm7uvE2Wg	Przewodniczący @bbudka: Rząd PiS nie zdał egzaminu. Dziś państwo ledwie dyszy podpięte do respiratora. Rządzący oblali ten egzamin, bo uznali, że można się nie przejmować ludzkim życiem, bo ministrowie pokazali arogancję i krótkowzroczność. https://t.co/eIm7uvE2Wg	Kazimierz Plocke	Koalicja Obywatelska	662
3	Good morning🌞 Today, at the morning Committee on Agriculture and Rural Development, we are considering and giving an opinion to the Public Finance Committee on the government draft budget act for 2021 in terms of parts intended for agriculture, rural development, agricultural markets and hunting. https://t.co/Yya4HynENm	Dzień dobry🌞 Dziś na porannej Komisji Rolnictwa i Rozwoju Wsi rozpatrujemy i zaopiniujemy dla Komisji Finansów Publicznych rządowy projekt ustawy budżetowej na rok 2021 w zakresie części przeznaczonych na rolnictwo, rozwój wsi, rynki rolne oraz łowiectwo. https://t.co/Yya4HynENm	Kazimierz Plocke	Koalicja Obywatelska	662
4	📌 There must be places for large numbers of patients and for the drug remdesivir. Respirators alone won't help. When funds were needed for residents who could help the	📌 Muszą być przygotowane miejsca na dużą liczbę chorych i lek remdesivir. Same respiratory nie pomogą. Kiedy potrzebne były środki na rezydentów, którzy mogliby pomóc	Kazimierz	Koalicja	662

When the user selects a party, five representatives from that party are generated by default. However, the user could select particular MPs from that party or all of the MPs from the party, whose tweets would be displayed in the main panel of the app. Following the drop down menu is a slider bar, which allows the user to select the number of tweets shown. The maximum number of tweets being able to be presented is limited to 20. There are two action buttons below the slider bar. The *Refresh selection* button allow the user to initialize their preferences and the *Download tweets* action button allows the user to save the main panel output to their machine. The main panel displays the tweets from the people and parties that the user has wished to see. The top left corner of the panel hosts a **Show_entries** button, which allows the user to select how many tweets to be displayed from the selection of 10, 25, 50 or 100 tweets. The top right corner hosts a **Search** button which allows the user to filter by a specific MP or political party. The table in the main panel displays the data generated by the user's selection. The first column displays the particular MP's tweet translated to English. The second column displays the original tweet in Polish followed by name of the MP and the party that they belong to. The last column shows the number of followers that the person has on the platform. All columns can be sorted by descending or ascending order.

Figure: Tweets sorted by number of followers for respective MP in ascending order



For more information visit:
Political parties in Poland

Choose the party: All

Select MPs: (Randomly selects 5 by default)
Joanna Kluzik-Rostkowska, Wanda Nowicka, Arkadiusz Mularczyk, Czesław Hoc, Piotr Kaleta

Number of tweets: 5

	Tweet translation	Tweet original	Full name	Party	Number of followers
6	PiS has just rejected the Senate's amendment granting a "holiday voucher" also to pensioners. And so the group with the highest number of votes for Duda was "discharged". I'm sorry, "drunk"	Właśnie PiS odrzucił poprawkę Senatu przyznającą „bon wakacyjny" również emerytom. I tak grupa najliczniej głosująca na Dudę, został „wydudana" . Przepraszam „wydudkana"	Jerzy Hardie-Douglas	Koalicja Obywatelska	322
7	Serving Czarnek in the Sejm, remembering his "achievements" in Lublin, the nonsense he tells on TVP does not surprise me. I am surprised that someone like this is portrayed as a "constitutional" lawyer.	Obsewując posła Czarnka w Sejmie, pamiętając Jego „dokonania" w Lublinie, bzdury opowiadane przez Niego w TVP mnie nie dziwią. Dziwi mnie, że ktoś taki przedstawiany jest jako prawnik „konstytucjonalista".	Jerzy Hardie-Douglas	Koalicja Obywatelska	322
8	From the beginning of the pandemic hysteria, I have very similar views. Maybe I do not entirely agree with the topic of vaccinations. It's worth https://t.co/rcW7XOClWW	Od początku pandemicznej histerii przedstawiam bardzo podobne poglądy. Może nie do końca zgadzam się tylko z wątkiem dotyczącym szczepień. Warto https://t.co/rcW7XOClWW	Jerzy Hardie-Douglas	Koalicja Obywatelska	322
9	https://t.co/OT4Jvz6EdS	https://t.co/OT4Jvz6EdS	Jerzy Hardie-Douglas	Koalicja Obywatelska	322
10	I am announcing a tender for my defense attorney after May 10 or 17 in connection with the improper treatment of the voting card. Only serious offers.	Ogłaszam przetarg na obrońcę mojej osoby po 10 lub 17 maja w związku z niewłaściwym potraktowaniem karty wyborczej. Tylko poważne oferty.	Jerzy Hardie-Douglas	Koalicja Obywatelska	322
16	On October 13, we celebrate the Medical Rescue Day. That is why today I would like to express my heartfelt wishes and expressions of respect for the effort and dedication in everyday, important work. I wish you health, strength and perseverance needed not only at work but also in everyday life.	13 października obchodzimy Dzień Ratownictwa Medycznego. Dlatego dziś pragnę złożyć serdeczne życzenia, oraz wyrazy szacunku za trud i poświęcenie w codziennej, jakże ważnej pracy. Życzę zdrowia, siły, oraz wytrwałości potrzebnej nie tylko w pracy ale i w życiu codziennym.	Mirosław Maliszewski	Koalicja Polska	758

Task 2

Introduction

The task at hand shows the usage of a simple linear regression of seal data using a non-parametric bootstrapped sample in SAS. The linear regression aims to determine what the relationship between seal length, treated as the explanatory variable, and testosterone levels, the response variable, is. The first part of the task shows the generation of an optimized bootstrap sample, which is significantly more time-efficient than the already existing bootstrapping code, available in the *RegBoot.sas* file. The optimized bootstrapping approach significantly speeds up the computational process of the program by utilizing SAS macros.

Method

Method1_Built in procedure using PROC REG

To compare the results of bootstrapping about confidence interval values and parameter estimates, PROC REG built-in function is used in the analysis. The model used for testosterone levels and lengths.

Method2_Longwinded loopy model time(rootBoot.SAS)

Initially the RegBoot.sas file was examined as a baseline for the rendering process' performance. First a loop with the number of randomization was initialized. Then sampling with replacement was performed. Afterwards a regression was run for the seal lengths and the testosterone level covariate using the bootstrapped sample. The parameter estimates for the slope and the intercept of the regression were derived and stored in the *Temp* dataset. Then, a new dataset called *ResultHolder* was created to store the to store the slope and the intercept of the regression output . The intercept and slope were renamed to "RandomIntercept" and "RandomSlope" for readability. At the end, the rendering time for the code was computed to serve as a baseline to compare the efficiency of the modified version.

Method3_Modified Model(Avoid a loop)

After the bootstrapping and regression was done using the *RegBoot* code, the same procedure was repeated using the optimized code. The bootstrapped samples were generated and a regression with the bootstrapped sample was performed in order to obtain the parameter estimates for the intercept and the slope, which were later extracted into the *ResulsHolder* dataset. The parameters were renamed to match the ones from the *RegBoot* code for consistency.

The procedure from the aforementioned steps was repeated, mainly, the bootstrapped samples were generated, the regression was run and parameter estimates for slope and intercept were derived. The output of the regression was summarized and stored in a dataset. The expected values of the means as well as 95 percent confidence intervals for the regression intercept and explanatory variable were computed. Afterwards, the distributions of the response and the covariate were visualized. Two graphs of density plots stacked on histograms were added.

Results

speed test (between bootstrapping method 2 and 3)

The method 2(regBoot.SAS) performs relatively slow and inefficient method, which takes 3.09 seconds to execute. In comparison, the optimized version(method3) reduces that time to 0.78 seconds, expediting the process by almost four times. In short, the table bellow indicates that the method 3 is more efficient than the second one.

Table: Rendering times of standard model versus modified model per sample size

Sample size	Method 2	Method 3
100	3.09	0.78
1000	21.05	0.85
10000	257.68	1.47

Compare the parameter estimates and 95% confidence intervals

All of the methods suggest that there is a significant positive effect between the explanatory variable seal length, and the response variable, testosterone levels, because 95% the confidence intervals for the coefficients of the covariates do not contain zero. The parameter estimates shown signify only a small difference in each method, meaning the bootstrapping test result might be similar to the variable of the true population. However, the third method has a much wider interval compared to methods 1 and 2; this means that the third method is less accurate in to estimating the parameter. In addition, method 3 has a larger variation than methods 1 and 2. As for the regression, the histogram below illustrates the bootstrap distributions of the X variable, which represents seal length. The distribution of Method 2 and 3 for the X appear close to normal distributions. Also, in method 2 and 3, if we increase the sampling test number, this distribution matches normality better. However, those methods utilize bootstrapping, which is non-parametric test, so it is not related to the assumption of normality. Yet, as the histogram for method 1 looks right-skewed, the Shapiro-Wilk test is needed to check the assumption of normality. In conclusion, we are 95% confident that the testosterone level(µg/l) on length(cm) is between 74.83 to 119.43 by method 3.

Table: Results for Confidence Intervals(Sample size: 1000 for bootstrapping)

Model type	Method 1		Method 2		Method 3	
CI	lower	upper	lower	upper	lower	upper
Intercept	76.20	116.10	79.57	112.17	74.83	119.43
Length	1.86	2.22	1.89	2.18	1.83	2.22

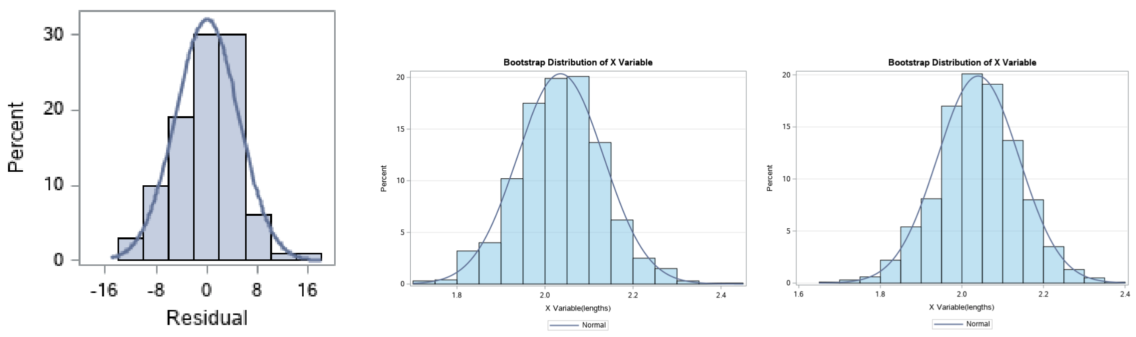
Table 95% Confidence Interval for Bootstrapped RandomIntercept

Model type	Method 1	Method 2	Method 3
Mean	96.15	96.12	96.61
Standard Deviation	10.05	8.53	11.04
Variation	101.01	72.83	122.02

Table 95% Confidence Interval for Bootstrapped RandomSlope

Model type	Method 1	Method 2	Method 3
Mean	2.036	2.036	2.031
Standard Deviation	0.091	0.076	0.099
Variation	0.008	0.006	0.009

Figure: Distribution of Seal Length(method1, method2, method3)



Task 3

The third task utilizes the Jackknife method, an alternative to the bootstrap method, to conduct non-parametric sampling of the *seal* dataset, which contains seals' testosterone levels as well as their lengths. Whereas the bootstrapping method uses sampling with replacement, the Jackknife method constructs sampling with omission.More specifically, the Jackknife method sequentially omits the ith observation when performing the sampling procedure.

Method

Initially, the *seals* dataset was replicated and a separate dataset, which contains only the *length* variable, was created, as that is the variable of interest for sampling. The relevant summary statistics for *length* were later generated. Those would be later used to be compared against the estimates obtained by the Jackknife approach. The length column of the new dataset was converted to a matrix. The rows of the matrix were counted, as they constitute the sample size that would later be used by the Jackknife method. A separate matrix used for storing the means of the Jackknife samples was created. Thereafter, a loop, which computes the Jackknife samples, was initialized. The loop ran throw 100 times, generating a total of 100 samples of size 99, due to the sequential omission of the ith term performed by the Jackknife method. Following the sample generation, the relevant estimators for each sample were computed. The mean of each sample was calculated. Afterwards, the mean of the sample means was computed and used for comparison against the mean obtained by the original sample of the length observations. The standard error and the bias associated with the Jackknife mean were also computed.

Results

The SAS generated *Statistical Summary* table displays the results of the Jackknife method regarding the estimation of the mean as well as the standard error of the sample. The results below show that the estimate of the original, 110.71628, exactly matches that derived using the Jackknife method, which is a strong indicator that the sampling methodology utilizes is in fact robust.

Table: Statistical summary of parameters obtained using the Jackknife method

Mean (original sample)	Standard Error (original sample)	Mean Jackknife Estimate	Jackknife Bias	Standard Error
110.71628	0.5537712	110.71682	-1.41E-12	0.5537712

The bias derived from the method is a small negative number -1.41E-12. However, any deviation from zero indicates that the parameter is biased, which in this case suggests that the mean obtained from the Jackknife Method is a biased estimator of the true population mean. The biased estimate shows that the correlation coefficient is small; therefore, a conclusion can be drawn that the generated samples have identical or lower probability as the population. The standard error of the mean obtained from the Jackknife method is 0.5537712, which implies that the data is not spread out and concludes that the the Jackknife mean estimate is a plausible representation of the population parameter. Moreover, the Jackknife standard error is the same as that obtained from the original sample.The exact match of the sample mean and standard errors obtained from the original sample and the Jackkinfe method sub-samples indicates that for this dataset, the Jackknife method is a valid sampling approach.

Task 4a

Intro

Given two random variables X, Y (described by the following distributions: $X \sim \mathsf{Norm}(\mu = 4, \sigma^2 = 10)$, $Y \sim \mathsf{Unif}(a = 2, b = 8)$), we estimated, using Monte Carlo simulations, the probability $Pr(X > Y)$. We then derived a sampling distribution for this estimate using bootstrapping techniques, before finally investigating how the variance of this sampling distribution changes as we vary the number of Monte Carlo simulations.

Method

Our first hurdle, estimating the value of $Pr(X > Y)$, can be completed by using Monte Carlo simulations. We set the number of Monte Carlo simulations we wished to run, in our case 10000. We then fed that number in to the built-in R functions `rnorm(), runif()` to generate two vectors which store 10000 observations from one of our distributions. We then compared each pair of values and created a vector which keeps track of whether that specific observation of X was greater than that of Y. To finish off, we simply counted how many $TRUE$ values appeared in the aformentioned vector and divided it by the number of Monte Carlo simulations we did to end up with out p value.

To find the sampling distribution of our p value, we resampled our X and Y observations by bootstrapping 10000 times. For each bootstrap we, again, worked out the p value for our bootstrapped X and Y observations in the method described above. With each iteration of our loop we stored this p value in a vector. Ultimately we end up with 10000 bootstrapped p values. When we set the number of bootstrapped samples we want to be high, this code can take a while to run. To speed up the process, a parallel computing technique was implemented, which drastically sped up the run time to only a few seconds.

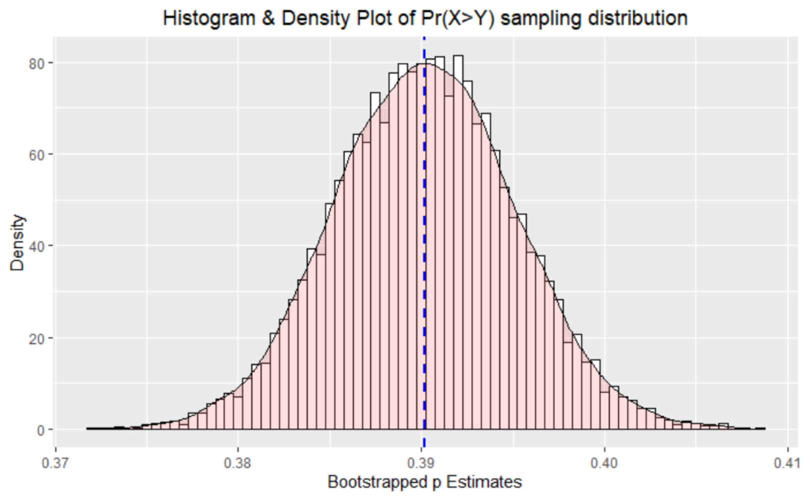
Lastly, to investigate how the variance of this sampling distribution depends on the initial number of Monte Carlo simulations we ran, we created a pair of nested for loops. We loop over every number of possible Monte Carlo simulations from 1 until we decide to end the loop.Firstly, we obtained observed data from our initial distributions. The number of observations we collect is equal to the number of Monte Carlo simulations we wanted in that iteration. We then bootstrap from these obserations in our second loop to end up with a vector of p values as above, from which we can pull and store the variance for each iteration of the outside loop. As this section of code takes even longer to run than the previous section, as we are looping over two variables not just one, parallel computational techniques were used again to speed up this process. However, even having implemented these techniques, the code is still relatively slow, so we restricted the maximum number of Monte Carlo simulations and number of bootstraps to 1000 and 100 respectively.

The number of Monte Carlo simulations to run, stored in the variable dubbed *MCsim*, was chosen to be 1000; the same number as the sample for the nonparametric bootstrapping method employed in order to derive the sampling distribution of the probability of **X** being greater than **Y**. A for loop was used to perform bootstrapping with resampling. The output, stored in the *p_store* variable, was then plotted to a histogram. The graph illustrates the sampling distribution of the probability that the variable **X** is greater than the variable **Y**. The data is normally distributed. In order to determine how the sample variance changes with the number of Monte Carlo simulations, new bootstrapping samples of 100 observations were constructed. The data was plotted with bootstrap increments of 100 observations for the total 1000 Monte Carlo simulations on the X-axis and the variance of the Pr(X>Y) distribution on the Y-axis.

Results

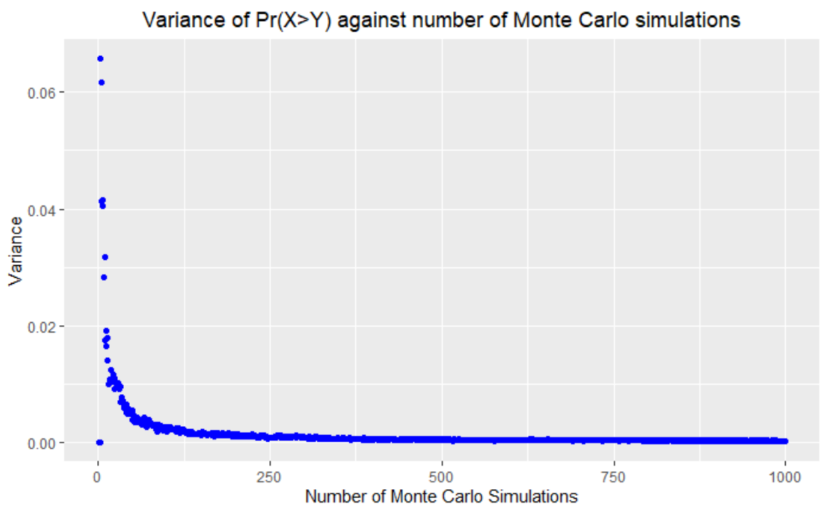
From our observed data, we find our estimate for $Pr(X > Y)$ to be 0.3922. After boostrapping, we can visually display the sampling distribution using the histogram below. From this, we can see that the sampling distribution appears relatively normal. The mean of the distribution is approximately 0.397 and the variance of the distribution is approximately 0.000024. Therefore, we describe the sampling distribution as $\hat{p} \approx \mathsf{Norm}(\mu = 0.397, \sigma^2 = 0.000024)$.

Fig 4a1: Sampling distribution of $Pr(X > Y)$



Having varied the number of Monte Carlo simulations as described in the Method section, we produced the following plot. For only 1 Monte Carlo simulation, we see that the variance is 0, which it is by definition as you need more than one observation to generate a variance. Beyond this, the variance jumps up to around 0.06 before exponentially decaying towards 0 as we increase the number of Monte Carlo simulations.

Fig 4a2: Variance of sampling distribution of $Pr(X > Y)$ as number of Monte Carlo simulations vary



Task 4b

Intro

A football tournament is structured so that a team plays games until they reach either 7 wins or 3 losses (draws are not allowed).We assumed a fixed win rate for a team throughout the tournament (i.e. if $p = 0.5$ then that team has a probability of winning a match is 0.5 for every match). We are interested in both how the average number of total matches played by a team varies with their win rate p and how the average observed win rate for a simulation of the tournamenet differs from their assumed win rate p.

Method

We used Monte Carlo simulations to retreive some observed data about the tournament; each simulation refers to simulating a whole tournament for a team. We set the number of Monte Carlo simulations to 10000. As the team can either win or lose a specific game, the random variable W which describes whether the team won that game is distributed binomially: $W \sim \mathsf{Binom}(n = 1, p)$.

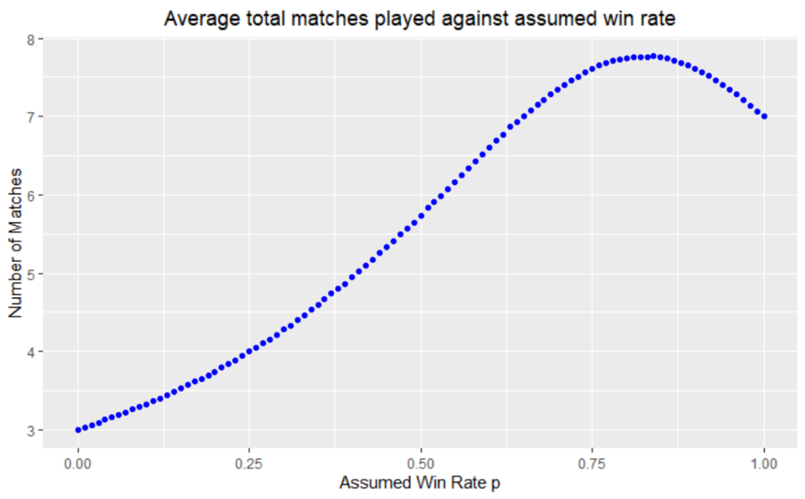
To simulate a tournament, we used a while loop which runs one binomial trial at a time as long as neither 7 wins nor 3 loses have been achieved. If it's a win, the win counter increments by one and if it's a loss, the lose counter increments by one. We then nest this inside another loop to run Monte Carlo simulations the prescribed 10000 times. At the end of each simulation, the total number of games played is calculated and stored, along with the observed win rate for that team in that tournament. We again nest these loops into another loop which loops over different p values in increments of 0.01. To speed up the code, parallel computational techniques were implemented.

Results

The below plot shows the relationship between the average number of games for differing p values. You can see that the relationship looks vaguely cubic, with the number of games increasing at an increasing rate as p

gets further away from 0. At around $p = 0.5$, the number of games continues to increase but at a decreasing rate before reaching a maximum at approximately $p = 0.8$ of between 7 and 8 games. After this point the number of games decreases towards 7 and $p = 1$. This is because as it gets increasinly likely to win a single game, we are less likely to see a single loss within the maximum number of games possible of 9 beyond $p = 0.8$. Therefore, we tend to win 7 games in a row, bringing the total number of games down from 8ish to 7.

Fig 4b1: Average number of matches played as assumed win rate p varies



The below figures show the observed win rate relative to assumed win rate and the residuals away from the slope 1 line (to aid in visualising the difference better). The intersection point represents where the probability of the tournament ending because of 7 wins is equal to the probability of the tournament ending because of 3 losses. To the right of this, we are more likely to see the game end because of wins. This means we see a disproportionate amount of games ending on a win, which drags up the observed win rate above the corresponding p value. To the left of this intersection point, we see a disproportionate amount of games ending in a loss which drags down the observed win rate below the corresponding p value. The deviation of the observed win rate away from the assumed win rate is so big because the final game will have a large affect on the observed win rate ($\pm 1/9$ or greater if less games are played). If the tournament instead ended with either 70 wins or 30 losses,, the final game would have a much small affect on the observed win rate ($\pm 1/90$) instead. Need to explain still why the deviation maxes at p=0.5.

Describing the graph, we can see that the observed win rate is less than the assumed win rate as p increases from 0. It gets further away as p increases, until it reaches it maximum deviation at approximately $p = 0.5$. From then on, the observed win rate increases up towards the assumed win rate. It intersects at around $p = 0.8$, after which, the observed win rate exceeds the assumed win rate, before coming down to meet it again at $p = 1$. From this, we can say that unless a team has an assumed win rate of above 0.8, (i.e. $p > 0.8$), they will win less games than they should due to the format of the tournament. This affect is maximised at $p = 0.5$, so average teams in terms of strength are the most hurt by the format. Exceptional teams ($p > 0.8$) can be said to have an unfair advantage as their observed win rate will be greater than their assumed win rate.

Fig 4b2: Observed win rate against assumed win rate

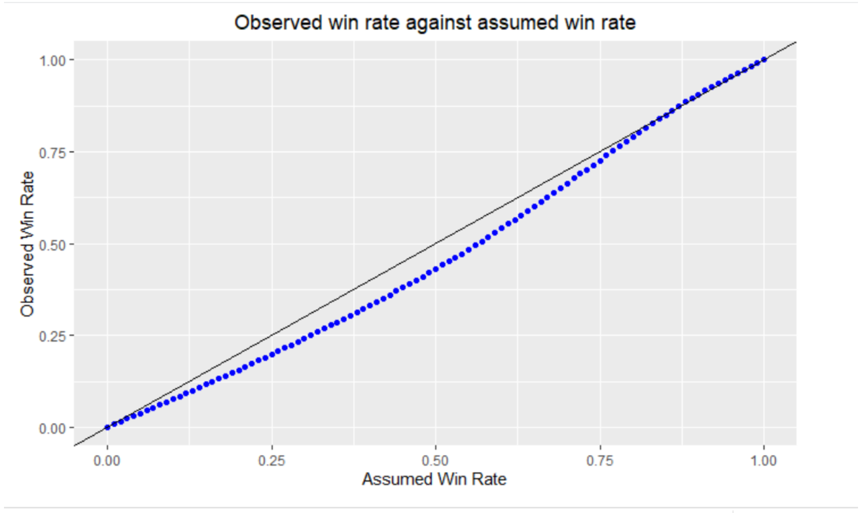
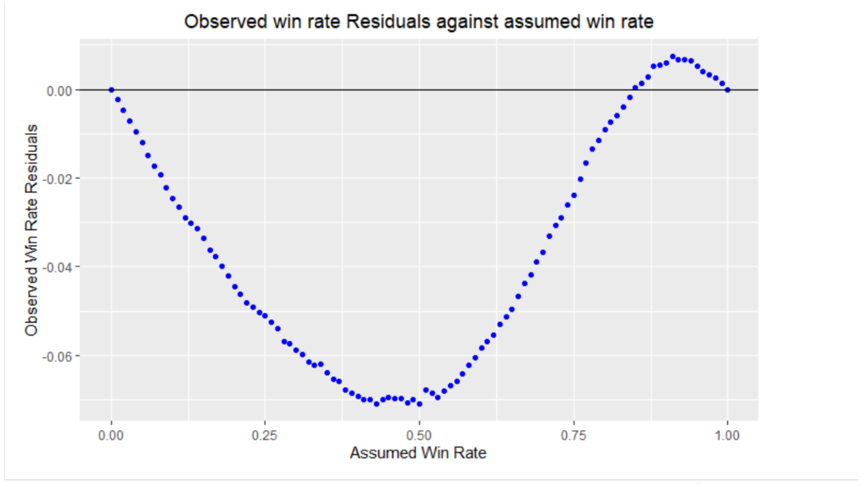


Fig 4b3: Residual plot of observed win rate against assumed win rate



Appendix

Code

Task 1

URL for Shiny App

```
knitr::include_app("https://asieminski.shinyapps.io/polminer/?fbclid=IwAR09rB2Hu4d5DY3f08W9xV4ATRs92IMs
```

Task 2

Method 1 code

```
/* Method1_Built in procedure using PROC REG */
%web_drop_table(WORK.IMPORT);
FILENAME REFFILE '/folders/myfolders/seals.csv';

PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=WORK.IMPORT;
    GETNAMES=YES;
RUN;

PROC CONTENTS DATA=WORK.IMPORT; RUN;
%web_open_table(WORK.IMPORT);

  PROC reg DATA = Work.import;
  MODEL testosterone = lengths / clb;
  RUN;
```

Method 2 code

```
/* Method2_Longwinded loopy model time(rootBoot.SAS) */
%web_drop_table(WORK.IMPORT);
FILENAME REFFILE '/folders/myfolders/seals.csv';

PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=WORK.IMPORT;
    GETNAMES=YES;
RUN;

PROC CONTENTS DATA=WORK.IMPORT; RUN;
%web_open_table(WORK.IMPORT);


%macro regBoot(NumberOfLoops, DataSet, XVariable, YVariable);

/*Number of rows in my dataset*/
    data _null_;
    set &DataSet NOBS=size;
    call symput("NROW",size);
    stop;
    run;

/*loop over the number of randomisations required*/
%do i=1 %to &NumberOfLoops;


/*Sample my data with replacement*/
    proc surveyselect data=&DataSet out=bootData seed=-23434 method=urs noprint sampsize=&NROW;
    run;

/*Conduct a regression on this randomised dataset and get parameter estimates*/
    proc reg data=bootData outest=ParameterEstimates  noprint;
    Model &YVariable=&XVariable;
    run;
    quit;

/*Extract just the columns for slope and intercept for storage*/
    data Temp;
    set ParameterEstimates;
    keep Intercept &XVariable;
    run;

/*Create a new results dataset if the first iteration, append for following iterations*/
    data ResultHolder;
        %if &i=1 %then %do;
            set Temp;
        %end;
        %else %do;
            set ResultHolder Temp;
        %end;
    run;
    %end;
/*Rename the results something nice*/
data ResultHolder1;
set ResultHolder;
rename Intercept=RandomIntercept1 &XVariable=RandomSlope1;
run;

/*generate rtf file with required output2*/
/*mean estimates & 95% CI*/
ods rtf file="/folders/myfolders/output2.rtf";
/*95% Confidence interval on the Intercept*/
    PROC UNIVARIATE DATA=ResultHolder1;
    VAR RandomIntercept1;
    OUTPUT OUT=bootRegCI_beta1 PCTLPTS=2.5 97.5 PCTLPRE=CI;
    RUN;

/*95% Confidence interval on the Slope*/
    PROC UNIVARIATE DATA=ResultHolder1;
    VAR RandomSlope1;
    OUTPUT OUT=bootRegCI_beta2 PCTLPTS=2.5 97.5 PCTLPRE=CI;
    RUN;

/*Merge and print of confidence intervals bounds for Intercept and Slope*/
    DATA bootRegCI_parameters;
    SET bootRegCI_beta1 bootRegCI_beta2;
    RUN;
    PROC PRINT DATA = bootRegCI_parameters;
    TITLE 'Confidence intervals';
    VAR CI2_5 CI97_5;
    RUN;
/*sgplot distribution*/
```

```
ods graphics;
title "Bootstrap Distribution of X Variable";
proc sgplot data=WORK.RESULTHOLDER1;
    histogram RandomSlope1 / fillattrs=(color=skyblue transparency=0.5);
    density RandomSlope1;
    xaxis label= "X Variable(lengths)";
    yaxis grid;
run;
ods graphics;
ods rtf close;

%mend;

options nonotes;


/*Run the macro & Run time check */
/* Start timer */
%let _timer_start = %sysfunc(datetime());

%regBoot(NumberOfLoops=1000, DataSet=Work.import, XVariable=lengths, YVariable=testosterone);

/* Stop timer */
data _null_;
  dur = datetime() - &_timer_start;
  put 30*'-' / ' TOTAL DURATION:' dur time13.2 / 30*'-';
run;
```

Method 3 code

```
/* Method 3_Modified Model(Avoid a loop)*/
%macro regBoot(NumberOfLoops, DataSet, XVariable, YVariable);

/* Start timer */
%let _timer_start = %sysfunc(datetime());
/*Generate all bootstrap samples*/
    proc surveyselect data=&DataSet
    out=bootData(rename=(Replicate=SampleID))
    seed=-23434
    method=urs                                  /* resample with replacement */
    noprint samprate=1                          /* each bootstrap sample has N observations */
    reps=&NumberOfLoops;
    run;
/*Conduct a regression on each randomised dataset and get parameter estimates*/
    proc reg data=bootData outest=ParameterEstimates  noprint;
    by SampleID;
    freq NumberHits;
    Model &YVariable=&XVariable;
    run;
    quit;
/*Extract just the columns for slope and intercept for storage*/
    data ResultHolder2;
    set ParameterEstimates;
    keep Intercept &XVariable;
    rename Intercept=RandomIntercept2 &XVariable=RandomSlope2;
    run;

/*generate rtf file with required output2*/
/*mean estimates & 95% CI*/
ods rtf file="/folders/myfolders/output2.rtf";
/*95% Confidence interval on the Intercept*/
    PROC UNIVARIATE DATA=ResultHolder2;
    VAR RandomIntercept2;
    OUTPUT OUT=bootRegCI_beta1 PCTLPTS=2.5 97.5 PCTLPRE=CI;
    RUN;

/*95% Confidence interval on the Slope*/
    PROC UNIVARIATE DATA=ResultHolder2;
    VAR RandomSlope2;
    OUTPUT OUT=bootRegCI_beta2 PCTLPTS=2.5 97.5 PCTLPRE=CI;
    RUN;

/*Merge and print of confidence intervals bounds for Intercept and Slope*/
    DATA bootRegCI_parameters;
    SET bootRegCI_beta1 bootRegCI_beta2;
    RUN;
    PROC PRINT DATA = bootRegCI_parameters;
    TITLE 'Confidence intervals';
    VAR CI2_5 CI97_5;
    RUN;
/*sgplot distribution*/

ods graphics;
title "Bootstrap Distribution of X Variable";
proc sgplot data=WORK.RESULTHOLDER2;
    histogram RandomSlope2 / fillattrs=(color=skyblue transparency=0.5);
    density RandomSlope2;
    xaxis label= "X Variable(lengths)";
    yaxis grid;
run;
ods graphics;
ods rtf close;


/* Stop timer */
data _null_;
  dur = datetime() - &_timer_start;
  put 30*'-' / ' TOTAL DURATION:' dur time13.2 / 30*'-';
run;
%mend;
options nonotes;
/*Run the macro*/
%regBoot(NumberOfLoops=1000, DataSet=Work.import, XVariable=lengths, YVariable=testosterone);

# Task 3
/* ---------------------------------------------------------------------------- */
/* Task3-jackknide.SAS file */
/* This file has implementation of jackknife simpulation of seal body lengths.  */
/* As results the programme will display the mean of jackknife estimat, biased  */
/* and standard error of the mean.   */
/* ---------------------------------------------------------------------------------- */

%web_drop_table(JK.DATA);

FILENAME REFFILE '/folders/myfolders/seals.csv';

PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=JK.DATA;
    GETNAMES=YES;
RUN;

PROC CONTENTS DATA=JK.DATA;
RUN;

%web_open_table(JK.DATA);

/*Creating the copy of the dataset*/
DATA JK.COPY_DATA;
SET JK.DATA;
RUN;

/*Creating the dataset with only 'lengths' variable*/
DATA JK.DATA_LENGTHS;
SET JK.COPY_DATA;
KEEP Lengths
RUN;

/*Numeric summary of the length variable*/
PROC UNIVARIATE DATA=JK.DATA_LENGTHS;
VAR Lengths;
RUN;

/* Convert length column into matrix */
proc iml;
use JK.DATA_LENGTHS;
read all var {"Lengths"} into lengthMatrix;
close;

/* Compute a statistic for original sample */
originalstatisticMean = mean(lengthMatrix);

/* calculating the number of rows */
numberOfRows = nrow(lengthMatrix);

/* insilizatation of the matrix in which Jacknife samples can be stored */
outputMatrixOfJK = j(numberOfRows - 1, numberOfRows, 0);

/* to store the mean of each Jacknife sample */
outputMatrixOfJK_mean = j(numberOfRows, 1, 0);
```

```
/* loop to creating a Jacknife sample,
and remove the i th observation from i th jackknife sample and
compute a statistic mean for each and store it to a matrix */
do i = 1 to numberOfRows;
    outputMatrixOfJK[,i] = remove(lengthMatrix, i)`;
    outputMatrixOfJK_mean[i] = mean(outputMatrixOfJK[,i]);
end;

/* calculating the mean of each mean matrix of each jackknife sample */
meanJKestimate =  mean(outputMatrixOfJK_mean);

/* calculate jackknife bias */
jkBias = (numberOfRows-1)*(meanJKestimate -  originalstatisticMean);

/* calculate the standard error of the mean */
sampleSizeCalculation = (numberOfRows-1)/numberOfRows;
squareOfmeans = ssq(outputMatrixOfJK_mean  - meanJKestimate);
stdErrJack = sqrt(sampleSizeCalculation * squareOfmeans);

/* Printing the statistical summary of the results */
Statistical_summary = originalstatisticMean || meanJKestimate || jkBias || stdErrJack;
print Statistical_summary[c={"Estimate - original Sample " "Mean Jackknife Estimate" "Jackknife Bias" "S
```

#Task 4

```r
#Import Libraries
library(tidyverse)
library(foreach)
library(parallel)
library(doParallel)

#Problem 4A

##Monte Carlo Simulation to compute Pr(X > Y)##
start=proc.time() # Used throughout code chunks to time how long computation takes

MCsim <- 10000 # No. of Monte Carlo simulations to run

set.seed(314159) # Set seed for reproducibility
x <- rnorm(MCsim, mean = 4, sd = sqrt(10))
y <- runif(MCsim, min = 2, max = 8)
a <- ifelse(x > y, TRUE, FALSE)
p <- sum(a %in% TRUE) / MCsim

time <- proc.time() - start
time


##Bootstrap to derive sampling distribution of Pr(X > Y)##
start <- proc.time()

N <- 10000 # No. of boostrapped observations
p_store = rep(NA, N)
```

```r
cl <- parallel::makeCluster(spec = 4, type = "PSOCK")
doParallel::registerDoParallel(cl)

p_store <- foreach(i = 1:N, .combine = "c") %dopar% {
  set.seed(314159+i)
  xBoot <- x[sample(x = MCsim, size = MCsim, replace = T)]
  yBoot <- y[sample(x = MCsim, size = MCsim, replace = T)]
  aBoot <- ifelse(xBoot > yBoot, TRUE, FALSE)
  sum(aBoot %in% TRUE) / MCsim
}

parallel::stopCluster(cl)

p_store_df <- data.frame(p_estimate = p_store)

ggplot(p_store_df, aes(x = p_estimate)) +
  geom_histogram(binwidth = 0.0005,aes(y=..density..), colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666") +
  geom_vline(aes(xintercept=mean(p_estimate)),
             color="blue", linetype="dashed", size=1) +
  labs(title = "Histogram & Density Plot of Pr(X>Y) sampling distribution", y = "Density", x = "Bootstra
  theme(plot.title = element_text(hjust=0.5))

time <- proc.time() - start
time


##How sample variance varies with number of MC Sims##
start=proc.time()

MCsim = 1000 # number of MC sims
N = 100 #number of bootstraps
p_store <- rep(NA, N)

cl <- parallel::makeCluster(spec = 4, type = "PSOCK")
doParallel::registerDoParallel(cl)

var_store <- foreach(j = 1:MCsim, .combine = "c") %dopar% {
  set.seed(314159)
  x <- rnorm(j, mean = 4, sd = sqrt(10))
  y <- runif(j, min = 2, max = 8)
  for (i in 1:N){
    xBoot <<- x[sample(x = j, size = j, replace = T)]
    yBoot <<- y[sample(x = j, size = j, replace = T)]
    aBoot <- ifelse(xBoot > yBoot, TRUE, FALSE)
    p_store[i] <- sum(aBoot %in% TRUE) / j
  }
  var(p_store)
}

parallel::stopCluster(cl)

var_df <- data.frame(MCsim = seq(1,MCsim,1),
                      Variance = var_store)

ggplot(var_df, aes(x = MCsim, y = Variance)) +
  geom_point(color = "blue") +
  labs(title = "Variance of Pr(X>Y) against number of Monte Carlo simulations", y = "Variance", x = "Nu
  theme(plot.title = element_text(hjust=0.5))

time <- proc.time() - start
time

#Problem 4B
start <- proc.time()

MCsim <- 10000
p <- seq(0,1,0.01)

cl <- parallel::makeCluster(spec = 4, type = "PSOCK")
doParallel::registerDoParallel(cl)

c <- foreach(j = 1:length(p), .combine = "cbind") %dopar% {
  set.seed(314159)
  matchCount <- rep(NA, MCsim)
  obsWinRate <- rep(NA, MCsim)
  for (i in seq(MCsim)){
    winCount = 0
    loseCount = 0
    while ((winCount < 7)&(loseCount < 3)){
      if (rbinom(1,1,p[j]) == 1){
        winCount = winCount + 1
      }
      else{
        loseCount = loseCount + 1
      }
    }
    matchCount[i] = winCount + loseCount
    obsWinRate[i] = winCount / matchCount[i]
  }
  c("meanCount" = mean(matchCount), "obsWinRateAvg" = mean(obsWinRate))
}

parallel::stopCluster(cl)

tournament_df <- data.frame(p = p, Count = c["meanCount",], ObsWinRate = c["obsWinRateAvg",])

ggplot(tournament_df, aes(x = p, y = Count)) +
  geom_point(color = "blue") +
  labs(title = "Average total matches played against assumed win rate", y = "Number of Matches", x = "A
  theme(plot.title = element_text(hjust=0.5))

ggplot(tournament_df, aes(x = p, y = ObsWinRate)) +
  geom_point(color = "blue") +
  geom_abline(mapping = NULL, data = NULL, slope=1) +
  labs(title = "Observed win rate against assumed win rate", y = "Observed Win Rate", x = "Assumed Win
```

```r
  theme(plot.title = element_text(hjust=0.5))

ggplot(tournament_df, aes(x = p, y = ObsWinRate - p)) +
  geom_point(color = "blue") +
  geom_abline(mapping = NULL, data = NULL, slope=0) +
  labs(title = "Observed win rate Residuals against assumed win rate", y = "Observed Win Rate Residuals
  theme(plot.title = element_text(hjust=0.5))

time <- proc.time() - start
time
```