FINAL ASSIGNMENT
ANALYSIS OF A “FREE DATASET”

Author : Manel Carrillo Maíllo
NIU: 1633426
Date: 24/05/2023

Abstract:

This project explores data job salaries using a comprehensive dataset. The analysis includes multiple linear regression, variable selection, and outlier detection techniques. The predictive model is evaluated based on coefficients, p-values, and residuals. Parametric and non-parametric bootstrap methods refine the predictions. The study provides insights into the accuracy of salary predictions and the impact of outliers and logarithmic transformations. Overall, this project contributes to understanding and predicting data job salaries.

INDEX:

Introduction

Have you ever found yourself sitting in a classroom, contemplating the future, and pondering the question: "How much can I earn once I complete my studies?" It is a natural curiosity that transcends academic limits, and in my case as a data science student, it led me to analyze the realm of data job salaries.

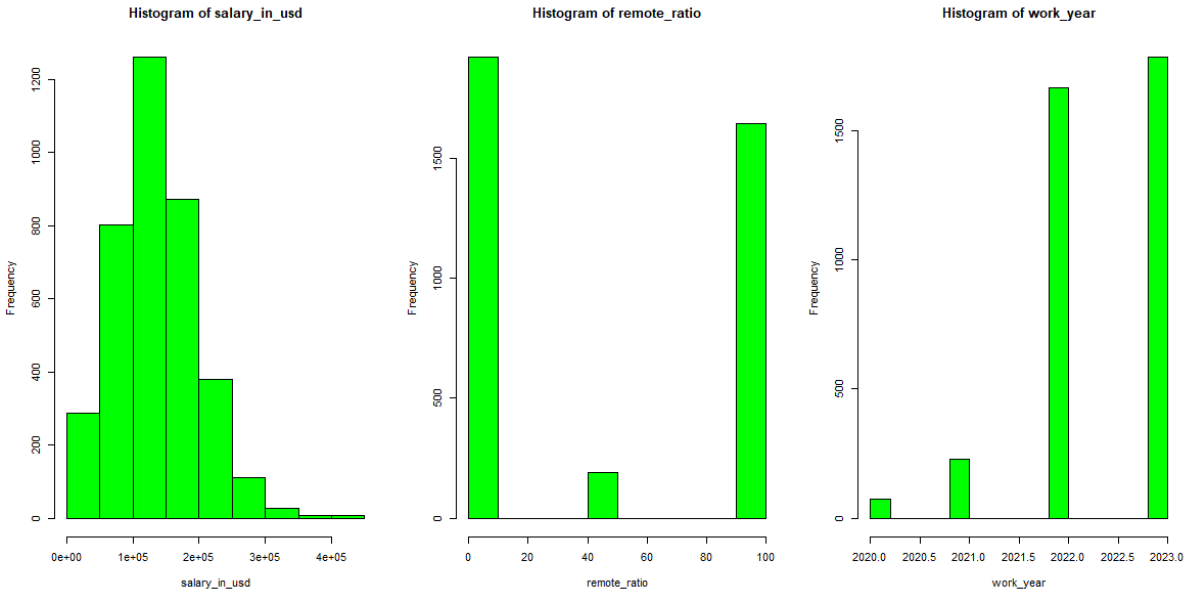
As a passionate about data, I am constantly intrigued by the financial rewards that await in this evolving field. The temptingness of data jobs extends beyond curiosity, as professionals are persuaded by the promises of lucrative salaries and remarkable growth opportunities.

In this study, my objective is to delve into the world of data job salaries and make predictions based on a comprehensive dataset.

Description of the Dataset

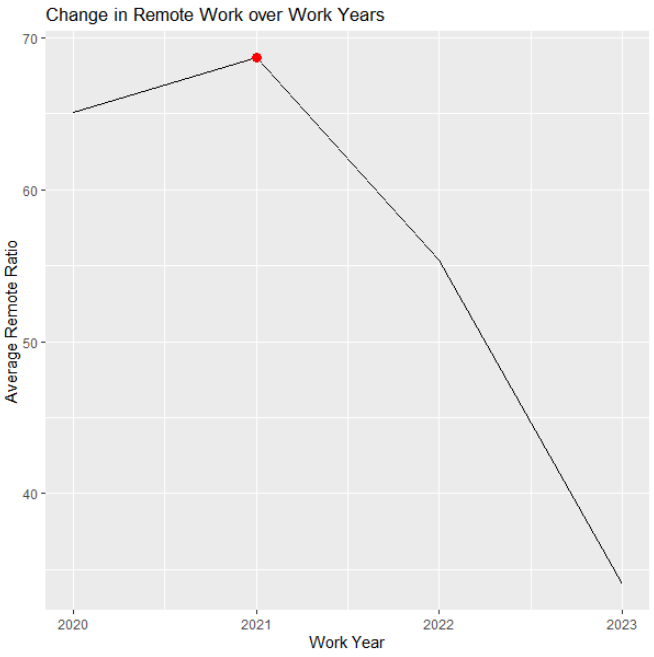
In this section, we will take a closer look at the dataset named "Data Science Salaries 2023." from kaggle and explore its variables. (Link)
This dataset has 11 variables which are:
work_year: Year when the salary was paid.
experience_level: Experience level in the job.
employment_type: The type of employment for the role.
job_title: The role worked during the year.
salary: The total gross salary amount paid.
salary_currency: The currency of the salary paid .
salary_in_usd: The salary in USD.
employee_residence: Employee's primary country of residence during the work.
work_remote_ratio: The overall amount of work done remotely.
company_location: The country of the employer's main office.
company_size: The median number of people that worked for the company during the year.

To know how these variables are distributed and to gain a better understanding of the data's characteristics, it is necessary to do histograms and bar charts or box plots.

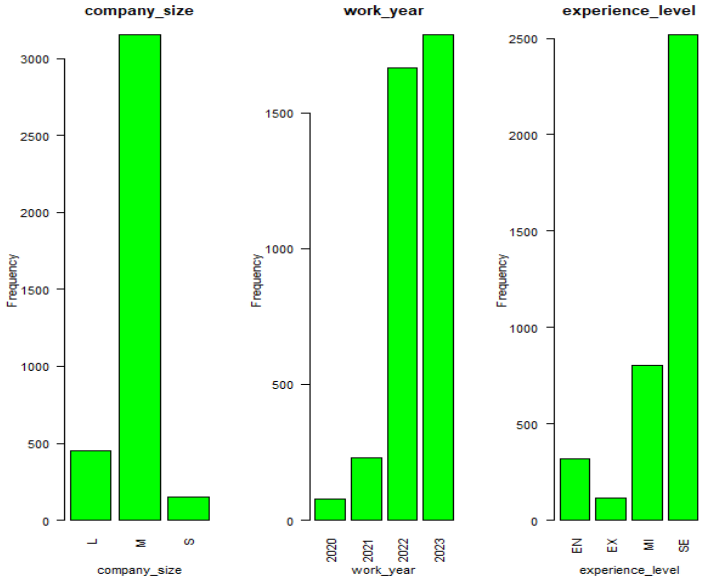


As we can see from the salary_in_usd histogram, it looks like this variable follows a normal distribution, this can be useful to do parametric stadistical methods as linear regressions, predictions or t-test.
Additionally, the median of the salary (135000) and the mean (137770) suggests relatively symmetrical distribution

From the other two histograms, we can say that non-remote work dominates the dataset, followed by remote work.
We can also see that 50% of remote work is relatively low.
This observation prompted me to consider the impact of COVID-19 on remote work practices, so I made a plot that shows the change in Remote Work over the years.
As we can see, there is a significant rise in remote work during the pandemic driven by lockdown measures raising its peak in 2021.
However, the subsequent decrease in the remote work ratio suggests a shift back to non-remote work settings post-pandemic.



Now, let's take a look at the categorical variables. In this part, not all variables are represented as some have a lot of different values and can not be well visualized.



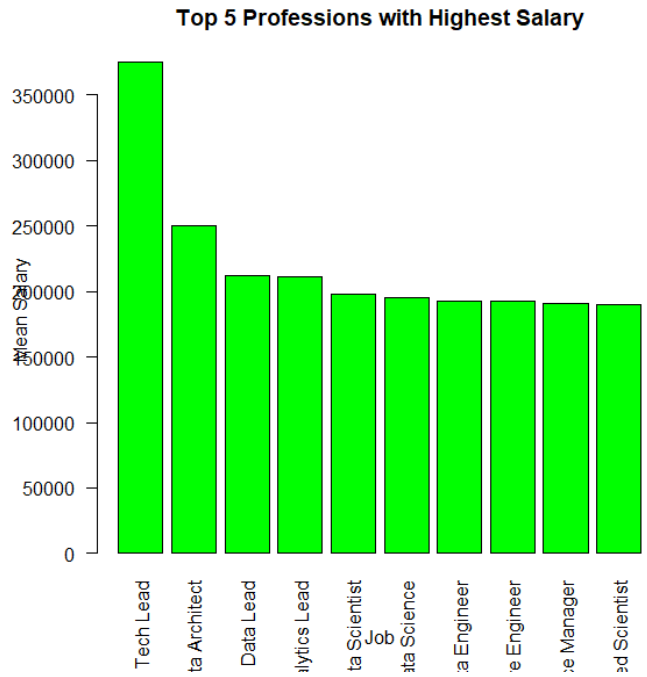
These boxplots focus on three categorical variables: company size, work year, and experience level.
The company size histogram shows that the frequency of the medium-sized company category is notably higher than small or large companies, indicating a predominance of medium-sized companies in the dataset.

The work year histogram reveals that the majority of salary data in the dataset corresponds to the years 2022 and 2023, suggesting a temporal focus on recent years.

Finally, analyzing the experience level, we conclude that senior-level positions are most prevalent, while junior and expert levels are less represented. (EN = Junior, MI = Mid-level, SE = Senior-level, and EX = Executive-level).

As it was informed there are variables of the dataset that I have not represented and that's because I made some special histograms ranking them to have more information.

1. **Top 5 professions with the highest salary:**



The first histogram shows the 5 professions with the highest salary, which was calculated doing the mean.

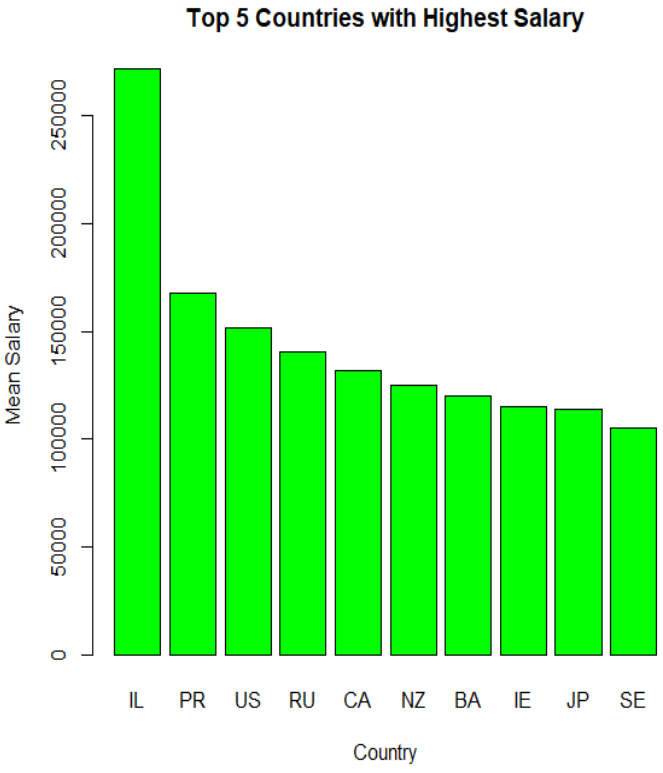
The highest-paid professions include Data Science Tech Leader, Cloud Data Architect and Data Leader, among others.

These findings indicate the value placed on specialized roles within the data field.

Data Science Tech Lead	375000.0
Cloud Data Architect	250000.0
Data Lead	212500.0
Data Analytics Lead	211254.5
Principal Data Scientist	198171.1
Director of Data Science	195140.7
Principal Data Engineer	192500.0
Machine Learning Software Engineer	192420.0
Data Science Manager	191278.8
Applied Scientist	190264.5

To know which countries were the ones that have higher salaries for data-jobs, I made a histogram where it shows the mean salary and the country (in descending order) being the top five countries with the highest salary:

2. **Top 5 Countries with highest Salary**



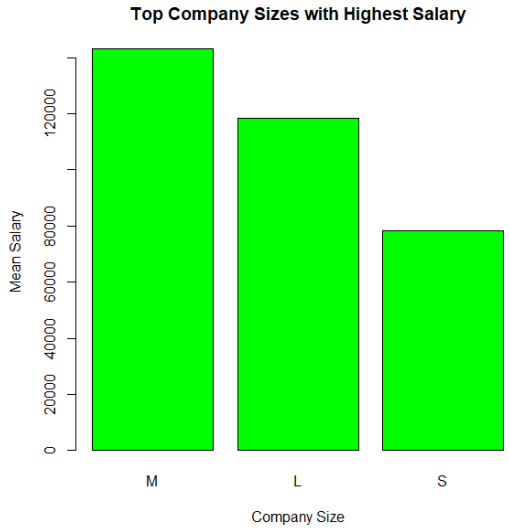
company_location	salary_in_usd
IL	271446.5
PR	167500.0
US	151822.0
RU	140333.3
CA	131917.7
NZ	125000.0
BA	120000.0
IE	114943.4
JP	114127.3
SE	105000.0

IL - Illinois, United States
PR - Puerto Rico
US - United States
RU - Russia
CA - Canada
NZ - New Zealand
BA - Bosnia and Herzegovina
IE - Ireland
JP - Japan
SE - Sweden

These findings highlight the global distribution of high-paying data jobs and may provide insights for individuals seeking lucrative opportunities in specific countries.

Another interesting piece of information to know about the categorical values, is the company size that has the highest salary, in other words, in what kind of company, data is more paid.

3. Company sizes with highest salary



company_size	salary_in_usd
M	143130.55
L	118300.98
S	78226.68

It is interesting to note that medium-sized companies tend to offer higher salaries compared to large companies.

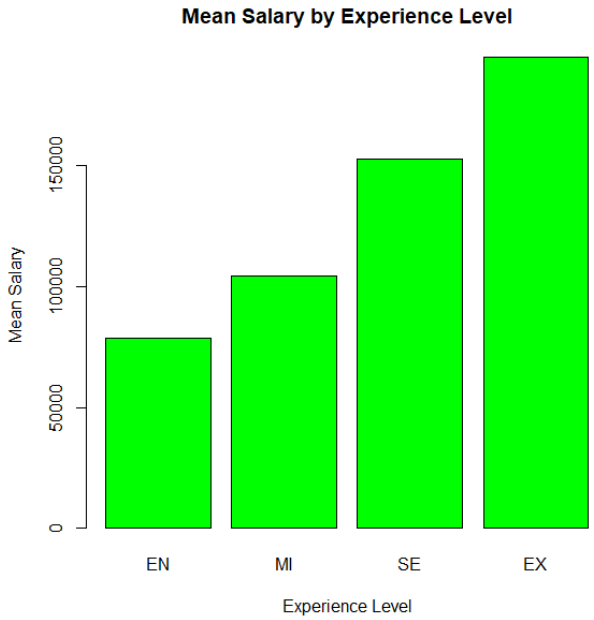
This finding suggests that the size of the company may play a role in determining salary levels.

One possible explanation for this observation could be the availability of more data for medium-sized companies in the dataset. The larger sample size of medium-sized companies might provide a more accurate representation of their salary distribution, whereas smaller or larger companies have fewer data points, leading to potential variability in salaries.

4 . Salaries distribution by Experience

It is also important to know for students or juniors how their salary will change over time, so I made this bar plot to know this information

Looking at the salaries distribution by employment type, we can see that, as expected, the highest salaries are related to more experienced workers or jobs even though there is less data for it.



experience_level	salary_in_usd
EN	78546.28
MI	104525.94
SE	153051.07
EX	194930.93

Data Analysis and Multiple Linear Regression

In this section, we focus on predicting data job salaries using multiple linear regression techniques. After eliminating currency-related variables and establishing the correlation between original salaries and their USD equivalents, we delve into the process of building predictive models. By identifying significant predictors and analyzing their impact on salary predictions, we aim to provide valuable insights into the factors influencing data on job salaries. This analysis serves as a valuable resource for data professionals looking to understand salary trends and make informed decisions regarding their careers.

Elimination of Currency Variables

As my goal was to predict salaries with the same currency unit to compare all of them, salary (that is the salary in a unit of currency), and the unit of currency itself were eliminated from the dataset.

Doing that, I was able to simplify the model and focus on the variables that have a stronger impact on the target variable.

Before eliminating them and to be sure that they weren't needed, I computed the correlation between salary in USD and salary and it gave -0.0236, a weak negative correlation.

Predictive Modeling And Variable Selection

To predict data job salaries, I employed multiple linear regression techniques and used the salary_in_usd variable as the response variable.
In this analysis, I engage various variable selection methods, like backward selection, forward selection, and stepwise selection, to identify the most influential predictors for predicting data job salaries in USD.

Backward Selection

Backward selection starts with a model that includes all potential predictors and sequentially removes the least significant predictors until an optimal subset is obtained.

Step 1:
The initial model included all predictor variables: work_year, experience_level, employment_type, job_title, employee_residence, remote_ratio, company_location, and company_size. Among these variables, company_location was found to be the least significant and was removed from the model. The resulting model had an AIC of 80948.

```
Start:  AIC=80975.58
salary_in_usd ~ work_year + experience_level + employment_type +
    job_title + employee_residence + remote_ratio + company_location +
    company_size

                    Df  Sum of Sq        RSS   AIC
- company_location   40 1.0809e+11 7.8557e+12 80948
- employment_type     3 7.3285e+09 7.7549e+12 80973
- remote_ratio        1 3.1925e+07 7.7476e+12 80974
<none>                              7.7476e+12 80976
- work_year           1 2.2465e+10 7.7700e+12 80984
- company_size        2 3.1254e+10 7.7788e+12 80987
- employee_residence 46 2.5301e+11 8.0006e+12 81004
- experience_level    3 7.2699e+11 8.4746e+12 81306
- job_title          92 1.4572e+12 9.2047e+12 81439

Step:  AIC=80947.6
```

Step 2:

In the second step, the employment_type predictor was identified as the least significant and was subsequently eliminated. This led to a slight reduction in the model's AIC to 80944.

```
salary_in_usd ~ work_year + experience_level + employment_type +
    job_title + employee_residence + remote_ratio + company_size

                     Df  Sum of Sq        RSS   AIC
- employment_type     3 4.3599e+09 7.8600e+12 80944
- remote_ratio        1 3.5383e+07 7.8557e+12 80946
<none>                             7.8557e+12 80948
- company_size        2 2.5824e+10 7.8815e+12 80956
- work_year           1 2.3091e+10 7.8788e+12 80957
- experience_level    3 7.4473e+11 8.6004e+12 81282
- job_title          92 1.4496e+12 9.3053e+12 81399
- employee_residence 77 2.1852e+12 1.0041e+13 81715

Step:  AIC=80943.69
```

Step 3:

Next, the remote_ratio predictor was determined to be the least significant and was removed from the model. This resulted in an AIC of 80943.69.

```
salary_in_usd ~ work_year + experience_level + job_title + employee_residence +
    remote_ratio + company_size

                     Df  Sum of Sq        RSS   AIC
- remote_ratio        1 4.1020e+07 7.8601e+12 80942
<none>                             7.8600e+12 80944
- company_size        2 2.7521e+10 7.8875e+12 80953
- work_year           1 2.3495e+10 7.8835e+12 80953
- experience_level    3 7.4853e+11 8.6085e+12 81279
- job_title          92 1.4547e+12 9.3147e+12 81397
- employee_residence 77 2.2137e+12 1.0074e+13 81721

Step:  AIC=80941.7
```

Step 4:

In the final step of backward selection, no additional predictors were found to be statistically insignificant. Therefore, the final model included the following variables: work_year, experience_level, job_title, employee_residence, and company_size. The resulting AIC for this model was 80941.7.

```
                     Df  Sum of Sq        RSS   AIC
<none>                             7.8601e+12 80942
- company_size        2 2.7480e+10 7.8875e+12 80951
- work_year           1 2.3898e+10 7.8840e+12 80951
- experience_level    3 7.5009e+11 8.6102e+12 81278
- job_title          92 1.4548e+12 9.3149e+12 81395
- employee_residence 77 2.2145e+12 1.0075e+13 81720
```

The summary of the final model is:

```
Response: salary_in_usd
                     Df     Sum Sq    Mean Sq  F value    Pr(>F)
work_year             1 7.7789e+11 7.7789e+11 354.2024 < 2.2e-16 ***
experience_level      3 2.5030e+12 8.3433e+11 379.9035 < 2.2e-16 ***
job_title            92 1.4143e+12 1.5372e+10   6.9996 < 2.2e-16 ***
employee_residence   77 2.3433e+12 3.0432e+10  13.8570 < 2.2e-16 ***
company_size          2 2.7480e+10 1.3740e+10   6.2564  0.001939 **
Residuals          3579 7.8601e+12 2.1962e+09
```

Forward Selection

Forward selection begins with a null model and progressively adds predictors that improve the model's prediction accuracy.

Step 1:

The initial model includes only the intercept term (no predictors). The first predictor added is employee_residence, resulting in a significant improvement in the model's fit, as indicated by a decrease in AIC from 82999.81 to 81866.43.

```
Start:  AIC=82999.81
salary_in_usd ~ 1

                     Df  Sum of Sq        RSS   AIC
+ employee_residence 77 4.3323e+12 1.0594e+13 81866
+ company_location   71 4.0279e+12 1.0898e+13 81961
+ experience_level    3 2.9719e+12 1.1954e+13 82172
+ job_title          92 2.2309e+12 1.2695e+13 82576
+ company_size        2 7.8726e+11 1.4139e+13 82800
+ work_year           1 7.7789e+11 1.4148e+13 82801
+ employment_type     3 2.4482e+11 1.4681e+13 82944
+ remote_ratio        1 6.1464e+10 1.4864e+13 82986
<none>                             1.4926e+13 83000

Step:  AIC=81866.43
```

Step 2:

```
salary_in_usd ~ employee_residence

                   Df  Sum of Sq        RSS   AIC
+ job_title        92 1.8679e+12 8.7258e+12 81322
+ experience_level  3 1.1943e+12 9.3994e+12 81423
+ work_year         1 6.8831e+10 1.0525e+13 81844
+ company_size      2 4.6540e+10 1.0547e+13 81854
+ employment_type   3 3.2203e+10 1.0561e+13 81861
<none>                           1.0594e+13 81866
+ remote_ratio      1 2.0436e+09 1.0592e+13 81868
+ company_location 40 1.0826e+11 1.0485e+13 81908

Step:  AIC=81322.07
salary_in_usd ~ employee_residence + job_title
```

In the second step, the predictor job_title is added to the model, leading to a further decrease in AIC to 81322.07. This addition indicates that job_title is a significant predictor of salary_in_usd.

Step 3:

Next, the predictor experience_level is included in the model, resulting in a decrease in AIC to 80963.5. This addition indicates that experience_level contributes significantly to the prediction of salary_in_usd.

```
salary_in_usd ~ employee_residence + job_title

                   Df  Sum of Sq        RSS   AIC
+ experience_level  3 8.0735e+11 7.9185e+12 80964
+ work_year         1 7.1171e+10 8.6546e+12 81293
+ company_size      2 7.4074e+10 8.6517e+12 81294
<none>                           8.7258e+12 81322
+ employment_type   3 1.3251e+10 8.7126e+12 81322
+ remote_ratio      1 6.0846e+08 8.7252e+12 81324
+ company_location 40 1.0761e+11 8.6182e+12 81355

Step:  AIC=80963.5
```

Step 4:

In the fourth step, the predictor work_year is added to the model, resulting in a decrease in AIC to 80950.81. This addition indicates that work_year is also a significant predictor of salary_in_usd.

```
salary_in_usd ~ employee_residence + job_title + experience_level

                   Df  Sum of Sq        RSS   AIC
+ work_year         1 3.0924e+10 7.8875e+12 80951
+ company_size      2 3.4506e+10 7.8840e+12 80951
<none>                           7.9185e+12 80964
+ remote_ratio      1 1.2711e+09 7.9172e+12 80965
+ employment_type   3 6.8715e+09 7.9116e+12 80966
+ company_location 40 9.7329e+10 7.8211e+12 80997

Step:  AIC=80950.81
```

Step 5:
Finally, the predictor company_size is added to the model, resulting in the lowest AIC of 80941.7. This addition indicates that company_size is a significant predictor of salary_in_usd.

```
salary_in_usd ~ employee_residence + job_title + experience_level +
    work_year

                   Df  Sum of Sq        RSS   AIC
+ company_size      2 2.7480e+10 7.8601e+12 80942
<none>                           7.8875e+12 80951
+ remote_ratio      1 2.8278e+05 7.8875e+12 80953
+ employment_type   3 6.0571e+09 7.8815e+12 80954
+ company_location 40 9.9771e+10 7.7878e+12 80983

Step:  AIC=80941.7
```

After these steps, the final model includes the following predictors: employee_residence, job_title, experience_level, work_year, and company_size. The same result as doing the backward selection

The summary of the final model

```
Analysis of Variance Table

Response: salary_in_usd
                     Df     Sum Sq    Mean Sq  F value    Pr(>F)
employee_residence   77 4.3323e+12 5.6263e+10  25.6189 < 2.2e-16 ***
job_title            92 1.8679e+12 2.0303e+10   9.2447 < 2.2e-16 ***
experience_level      3 8.0735e+11 2.6912e+11 122.5393 < 2.2e-16 ***
work_year             1 3.0924e+10 3.0924e+10  14.0809 0.0001779 ***
company_size          2 2.7480e+10 1.3740e+10   6.2564 0.0019392 **
Residuals          3579 7.8601e+12 2.1962e+09
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Stepwise Selection

Stepwise selection combines both backward and forward selection processes by iteratively adding and removing predictors based on their statistical significance and impact on the model's performance

Step 1:
The initial model includes only the intercept term (no predictors). The first predictor added is employee_residence, resulting in a significant improvement in the model's fit, as indicated by a decrease in AIC from 82999.81 to 81866.43.

```
Start:  AIC=82999.81
salary_in_usd ~ 1

                    Df  Sum of Sq        RSS   AIC
+ employee_residence 77 4.3323e+12 1.0594e+13 81866
+ company_location   71 4.0279e+12 1.0898e+13 81961
+ experience_level    3 2.9719e+12 1.1954e+13 82172
+ job_title          92 2.2309e+12 1.2695e+13 82576
+ company_size        2 7.8726e+11 1.4139e+13 82800
+ work_year           1 7.7789e+11 1.4148e+13 82801
+ employment_type     3 2.4482e+11 1.4681e+13 82944
+ remote_ratio        1 6.1464e+10 1.4864e+13 82986
<none>                             1.4926e+13 83000

Step:  AIC=81866.43
```

Step 2:
In the second step, the predictor job_title is added to the model, leading to a further decrease in AIC to 81322.07. This addition indicates that job_title is a significant predictor of salary_in_usd.

```
salary_in_usd ~ employee_residence

                     Df  Sum of Sq        RSS   AIC
+ job_title          92 1.8679e+12 8.7258e+12 81322
+ experience_level    3 1.1943e+12 9.3994e+12 81423
+ work_year           1 6.8831e+10 1.0525e+13 81844
+ company_size        2 4.6540e+10 1.0547e+13 81854
+ employment_type     3 3.2203e+10 1.0561e+13 81861
<none>                             1.0594e+13 81866
+ remote_ratio        1 2.0436e+09 1.0592e+13 81868
+ company_location   40 1.0826e+11 1.0485e+13 81908
- employee_residence 77 4.3323e+12 1.4926e+13 83000

Step:  AIC=81322.07
```

Step 3:
Next, the predictor experience_level is included in the model, resulting in a decrease in AIC to 80963.5. This addition indicates that experience_level contributes significantly to the prediction of salary_in_usd.

```
salary_in_usd ~ employee_residence + job_title

                     Df  Sum of Sq        RSS   AIC
+ experience_level    3 8.0735e+11 7.9185e+12 80964
+ work_year           1 7.1171e+10 8.6546e+12 81293
+ company_size        2 7.4074e+10 8.6517e+12 81294
<none>                             8.7258e+12 81322
+ employment_type     3 1.3251e+10 8.7126e+12 81322
+ remote_ratio        1 6.0846e+08 8.7252e+12 81324
+ company_location   40 1.0761e+11 8.6182e+12 81355
- job_title          92 1.8679e+12 1.0594e+13 81866
- employee_residence 77 3.9692e+12 1.2695e+13 82576

Step:  AIC=80963.5
```

Step 4:
In the fourth step, the predictor work_year is added to the model, resulting in a decrease in AIC to 80950.81. This addition indicates that work_year is also a significant predictor of salary_in_usd.

```
salary_in_usd ~ employee_residence + job_title + experience_level

                     Df  Sum of Sq        RSS   AIC
+ work_year           1 3.0924e+10 7.8875e+12 80951
+ company_size        2 3.4506e+10 7.8840e+12 80951
<none>                             7.9185e+12 80964
+ remote_ratio        1 1.2711e+09 7.9172e+12 80965
+ employment_type     3 6.8715e+09 7.9116e+12 80966
+ company_location   40 9.7329e+10 7.8211e+12 80997
- experience_level    3 8.0735e+11 8.7258e+12 81322
- job_title          92 1.4809e+12 9.3994e+12 81423
- employee_residence 77 2.5743e+12 1.0493e+13 81867

Step:  AIC=80950.81
```

Step 5:

```
salary_in_usd ~ employee_residence + job_title + experience_level +
    work_year

                     Df  Sum of Sq        RSS   AIC
+ company_size        2 2.7480e+10 7.8601e+12 80942
<none>                             7.8875e+12 80951
+ remote_ratio        1 2.8278e+05 7.8875e+12 80953
+ employment_type     3 6.0571e+09 7.8815e+12 80954
- work_year           1 3.0924e+10 7.9185e+12 80964
+ company_location   40 9.9771e+10 7.7878e+12 80983
- experience_level    3 7.6710e+11 8.6546e+12 81293
- job_title          92 1.4810e+12 9.3685e+12 81413
- employee_residence 77 2.3433e+12 1.0231e+13 81774

Step:  AIC=80941.7
```

Finally, the predictor company_size is added to the model, resulting in the lowest AIC of 80941.7. This addition indicates that company_size is a significant predictor of salary_in_usd.

```
salary_in_usd ~ employee_residence + job_title + experience_level +
    work_year + company_size

                     Df  Sum of Sq        RSS   AIC
<none>                             7.8601e+12 80942
+ remote_ratio        1 4.1020e+07 7.8600e+12 80944
+ employment_type     3 4.3655e+09 7.8557e+12 80946
- company_size        2 2.7480e+10 7.8875e+12 80951
- work_year           1 2.3898e+10 7.8840e+12 80951
+ company_location   40 1.0512e+11 7.7549e+12 80971
- experience_level    3 7.5009e+11 8.6102e+12 81278
- job_title          92 1.4548e+12 9.3149e+12 81395
- employee_residence 77 2.2145e+12 1.0075e+13 81720
```

After these steps, the final model includes the following predictors: employee_residence, job_title, experience_level, work_year, and company_size.

```
Response: salary_in_usd
                    Df     Sum Sq    Mean Sq  F value    Pr(>F)
employee_residence   77 4.3323e+12 5.6263e+10  25.6189 < 2.2e-16 ***
job_title            92 1.8679e+12 2.0303e+10   9.2447 < 2.2e-16 ***
experience_level      3 8.0735e+11 2.6912e+11 122.5393 < 2.2e-16 ***
work_year             1 3.0924e+10 3.0924e+10  14.0809 0.0001779 ***
company_size          2 2.7480e+10 1.3740e+10   6.2564 0.0019392 **
Residuals          3579 7.8601e+12 2.1962e+09
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Based on the results obtained from the backward selection, forward selection, and stepwise selection, it appears that the final selected model is consistent across all three methods.

Study Of the Resultant Model

After doing the variable selection our multivariable linear model is: lm(salary_in_usd~ job_title + company_size + work_year + experience_level employee_residence,data=ds_salaries)

Coefficient and P-Value study

In this part of the analysis, I focused on studying the coefficients, confidence intervals, and p-values obtained from the model. This analysis is performed to understand the impact and significance of different predictor variables on the response variable.

The coefficients represent the estimated effect of each predictor variable on the response variable.
The confidence intervals provide a range of values within which we can be reasonably confident that the true coefficient lies.
Lastly, the p-values indicate the statistical significance of each predictor, helping us determine if there is strong evidence to reject the null hypothesis of no effect.

	Coefficient	Lower_CI	Upper_CI	p_value
(Intercept)	-8899565.5	-14190947.65	-3608183.4	9.847628e-04
employee_residenceIL	287580.6	177548.28	397612.9	3.144425e-07
job_titleData Science Tech Lead	271830.4	166306.06	377354.8	4.624942e-07
job_titleData Analytics Lead	166185.5	83192.58	249178.5	8.800702e-05
job_titleCloud Data Architect	151234.3	45685.78	256782.7	4.992376e-03
job_titleAI Developer	149195.9	86788.60	211603.3	2.872474e-06

In our case, employee_residenceIL shows a significant positive effect on salary, with an estimated increase ranging from $177,548.28 to $397,612.9. The p-value suggests strong evidence to reject the null hypothesis of no effect.

When a coefficient in the model is positive, it indicates a positive relationship between the corresponding predictor variable and the response variable. This means that as the predictor variable increases, the response variable is also expected to increase
On the other hand, when a coefficient is negative, it signifies a negative relationship between the predictor variable and the response variable. In this case, as the predictor variable increases, the response variable is expected to decrease.

If we want to know which variable is more significant and has more influence on the response variable, we have to order them by p-value, being the most significant, the ones with less p-value.

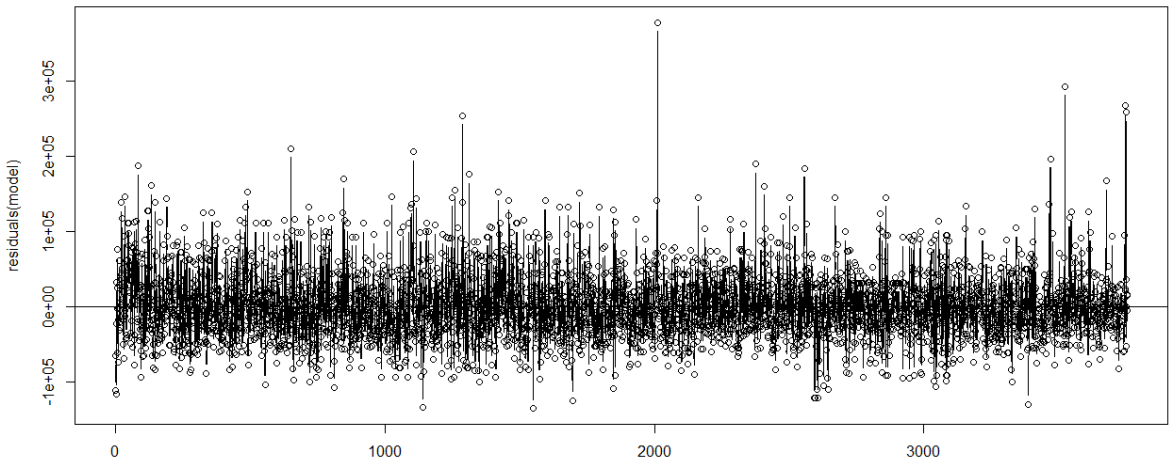
	Coefficient	p_value
experience_levelEX	83576.6509	1.841982e-46
experience_levelSE	44474.7923	8.474337e-43
experience_levelMI	20576.5077	1.526254e-09
employee_residenceIL	287580.6005	3.144425e-07
job_titleData Science Tech Lead	271830.4152	4.624942e-07

As we can see, experience level (Expert,Senior) have a positive coefficient and a very low p-value, this indicates that they are highly significant and have a positive impact on the response variable.

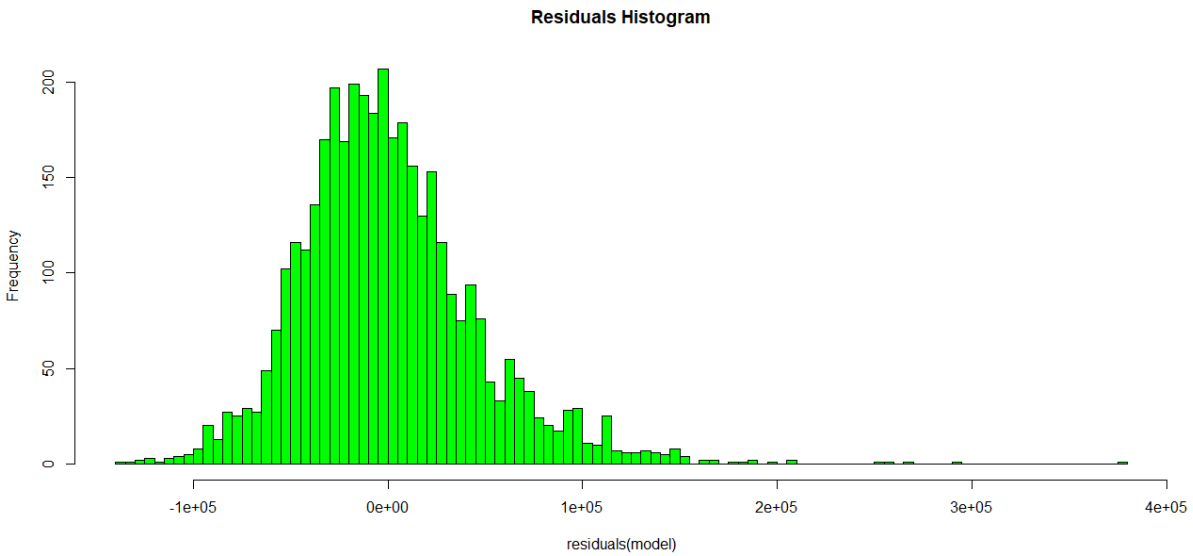
In summary, sorting by coefficient shows the magnitude of the variables' influence on salary prediction, while sorting by p-value helps identify variables that have a statistically significant effect.

Residuals Study

When analyzing a predictive model, it is important to assess its goodness-of-fit. One useful tool for evaluating model performance is the plot of residuals. A good model is expected to have residuals that exhibit a random scatter of points around the horizontal line at 0. Any discernible trends in the residuals indicate a poor fit of the model to the data.



In our case, the residual plot does not show any noticeable trend, which is a positive indicator. Additionally, the scatter of points is centered around the horizontal line at 0, further suggesting a good fit.

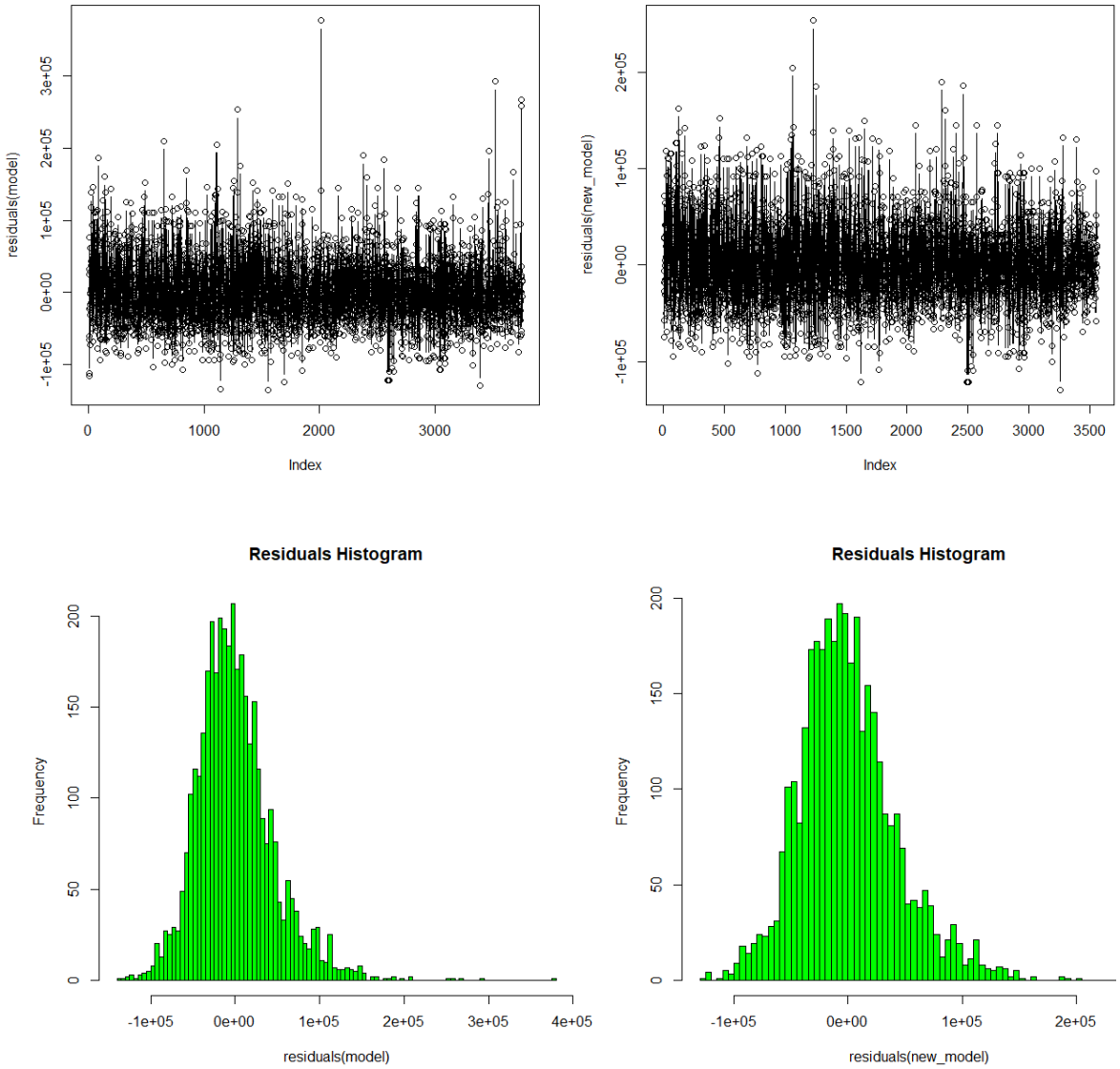


Furthermore, examining the residuals histogram, we observe that it approximates a normal distribution with a slight left skew. However, it is worth noting the presence of some outliers, which can be addressed by calculating Cook's distance.

Finding Outliers With Cook's Distance

By examining Cook's distance, we can identify observations that have a significant impact on the model's parameters and predictions. Outliers, influential points, or leverage points can have a substantial effect on the regression results, and detecting them is crucial for ensuring the validity of our model.

By evaluating Cook's distance, we can identify observations that may be excluded from the analysis to improve our regression model.



Upon analyzing the residual plot and histograms,before(left) and after (right) considering Cook's distance, we observe that the range of values for the residuals has decreased. This reduction indicates that the outliers, which were identified and excluded using Cook's distance, had a significant influence on the model's predictions. By removing these influential observations, the model's residuals have become more concentrated, indicating an improved fit to the data.

Predictions:

In this new phase of our analysis, we will focus on predictions using two models: the original model with outliers and the modified model without outliers and notice how the outliers influenced the model.

Furthermore, it is important to address the issue of negative predictions when dealing with salary data as happens with this model.
To overcome this challenge, one common technique is to apply logarithms to the salaries.
By taking the logarithm of the salaries, we can transform the data and ensure that all predictions are positive.

After applying the logarithmic transformation, we can make predictions using the transformed data. However, it's important to note that the predictions will be in logarithmic form. To obtain the actual predicted salary values, we can apply the exponential function (exp) to the predictions. This will convert the logarithmic predictions back to their original scale, providing us positive salary predictions.
In order to calculate the log salary model, it was used the data without outliers. The reason is because detected influential outliers at the log salary model, can not be influential or an outlier in the original model.

In this study, I wanted to compare the means of the predicted values in order to see the influence of the outliers in the model and then compare these means with the log model.

As we can see from the output, the scaled_log_pred model has a mean predicted salary of $131,847.3. This value is lower than the other models. The reason could be that logarithmic transformations have affected the overall scale of the predictions.

```
              Model     Mean
1 scaled_log_pred 131847.3
2      pred_model 137570.4
3  pred_new_model 137848.2
```

The pred_model model, representing the original data without transformations or outlier removal, has a mean predicted salary of $137,570.4. This indicates that, on average, the model predicts slightly higher salaries compared to the scaled_log_pred model.

The pred_new_model model, which excludes outliers while utilizing the original data, has a mean predicted salary of $137,848.2. This improvement suggests that removing outliers and working with the original data leads to slightly higher average predicted salaries compared to the pred_model model.

In order to assess the performance of our predictive models, we will now proceed with making predictions using each of the three models: scaled_log_pred, pred_model, and pred_new_model.

Prediction 1
This prediction can provide insights into the expected salary for a senior data analyst working in a large company. It helps individuals in this role to understand the potential salary range based on the given attributes.
Model prediction: 205808.7
New model prediction: 201536.8
Model with log :191852.3

Prediction 2
This prediction can shed light on the salary expectations for an executive data scientist working in a medium-sized company.
Model prediction: 203729.9
New model prediction: 202111.3
Model with log : 194114

Parametric and Non-Parametric Bootstrap

To explore the accuracy and reliability of the predictive models we employed two different bootstrap techniques: non-parametric and parametric bootstrap. The objective is to estimate the mean salaries using two models: the new model without outliers and the model with logarithmic transformations.

The non-parametric bootstrap method allows us to resample from the predicted salaries without making any assumptions about the underlying distribution

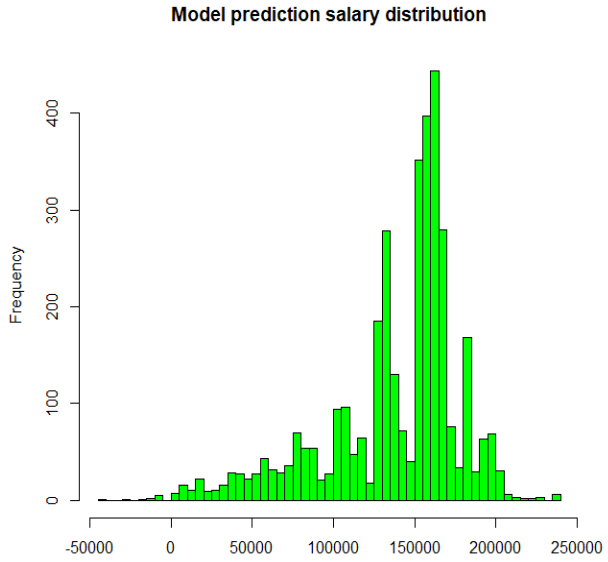
On the other hand, the parametric bootstrap leverages the assumption of a specific distribution.

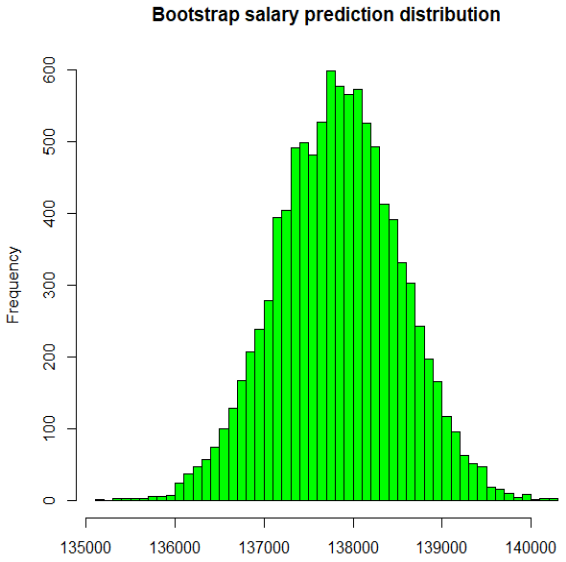
Mean of predictions

By comparing the results from the non-parametric and parametric bootstrap methods, we aim to evaluate the consistency and accuracy of the mean salary predictions.
Additionally, we will examine the distribution of predicted salaries and assess the width of confidence intervals to gain insights into the precision of our estimation.

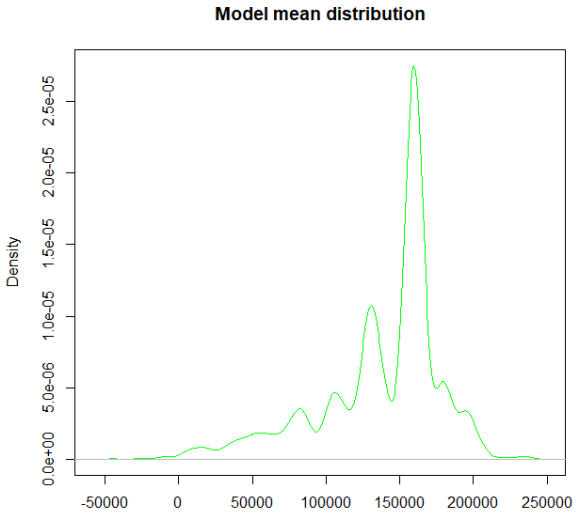
Non parametric Bootstrap (Model with no outliers)

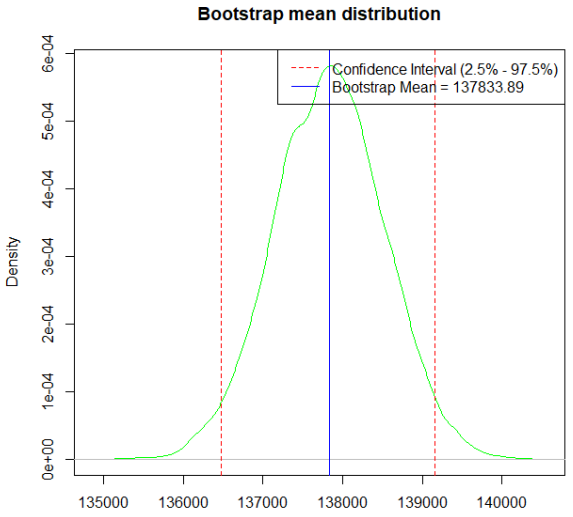
To compare the bootstrap distribution against the model with no outliers, we plotted histograms and density plots. The histogram of the model's predicted salaries showed the presence of negative predictions and a deviation from a normal distribution around the mean. In contrast, the histogram of the bootstrap results exhibited no negative predictions and followed a more normal distribution.
The mean of the bootstrap results was calculated as 137,847.6, which was close to the mean salary of 137,848.2 from the original model. This indicates that the non-parametric bootstrap estimates provided a reliable approximation of the mean salary.





Overall, the non-parametric bootstrap analysis allowed us to assess the distribution of predicted salaries and obtain a more robust estimate of the mean salary.



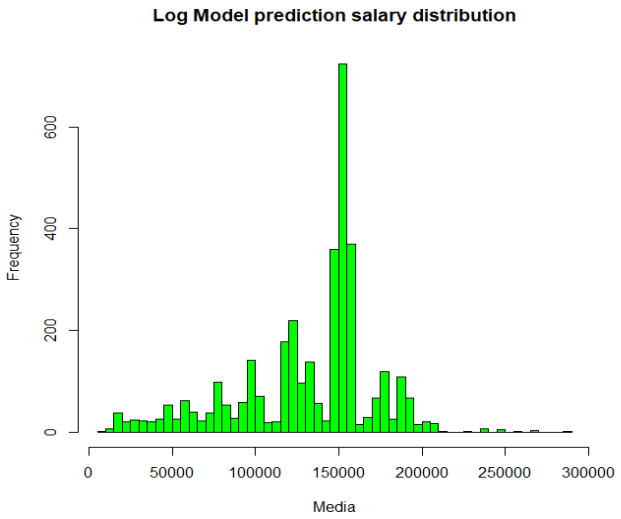


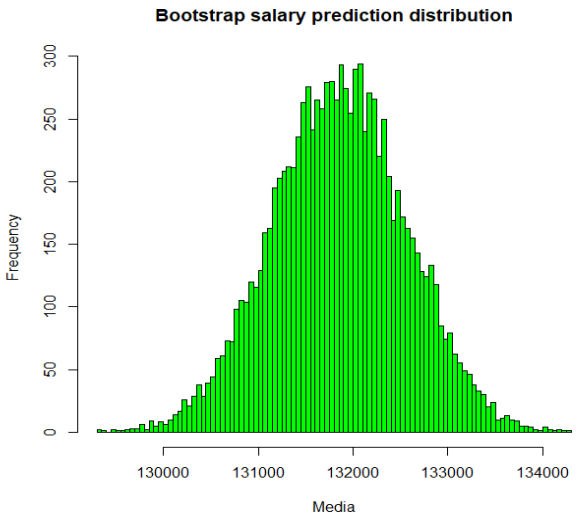
The resulting confidence interval for the bootstrap distribution for the mean salary was found to be 136,471.8 to 139,155.6. This interval represents the range within which we can be confident that the true mean salary lies.

Comparing this with the confidence interval obtained from the original data, which was 136,483.6 to 139,212.8, we can observe that the bootstrap confidence interval is slightly narrower. This suggests that the bootstrap method provides a more precise estimate of the population parameter, taking into account the variability in the data and the model's predictions.

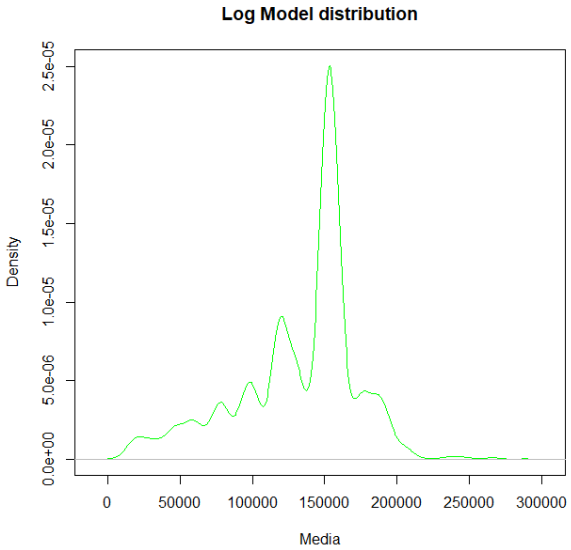
Non parametric Bootstrap (Log Transformation Model)

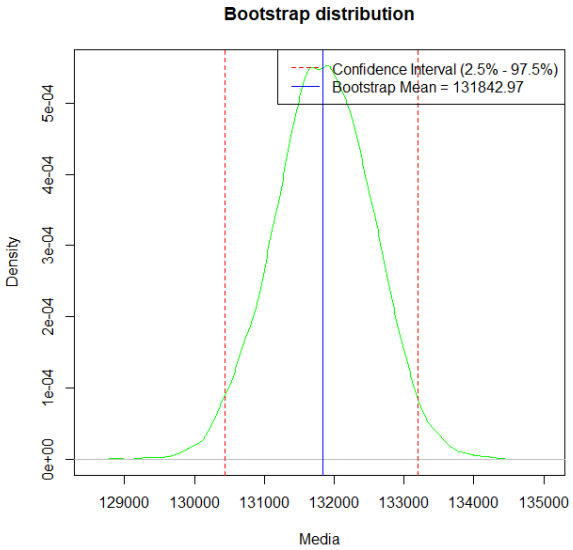
In a similar manner, I performed the same analysis using the logarithmic transformation model. I transformed the predicted salaries using the exponential function to obtain the actual salary predictions. The mean salary obtained from this model was found to be 131,837.9.
Comparing the mean salary obtained from the original data, which was 131,847.3, with the bootstrap mean, we observed that they were very close. This suggests that the non-parametric bootstrap method accurately captures the mean salary, even with the log transformation.





The histogram plots of the predicted salaries and the bootstrap results demonstrate that there are no negative predictions in any of the model distributions, thanks to the log transformation.



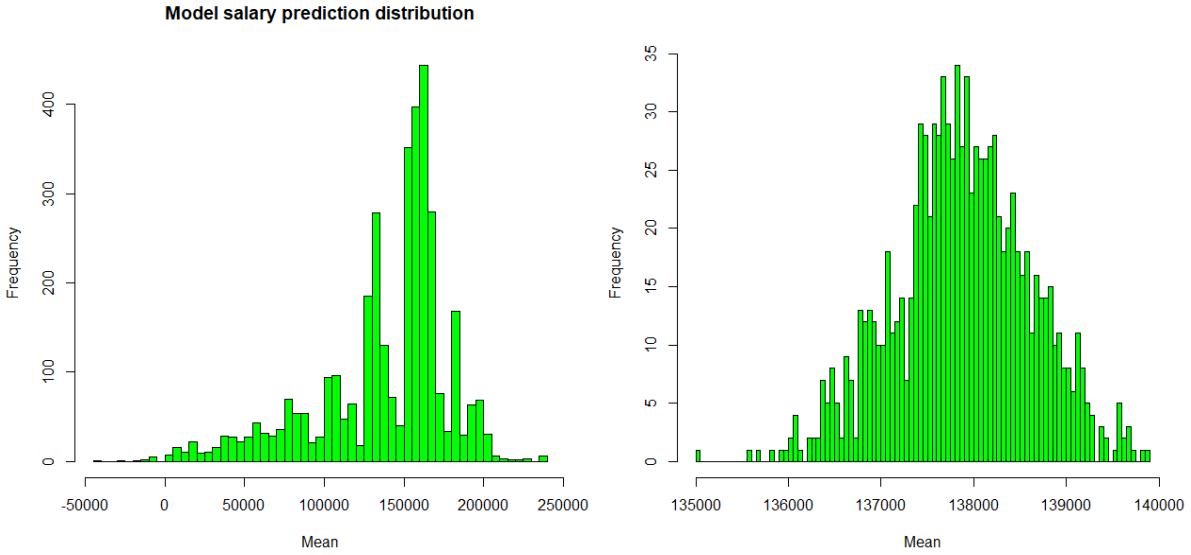


The bootstrap confidence interval (130468.7 - 133217.7) is narrower than the confidence interval of the log model (130474.6 - 133220), indicating that the bootstrap method provides a more precise estimate of the population mean.

Parametric Bootstrap (Model with no outliers)

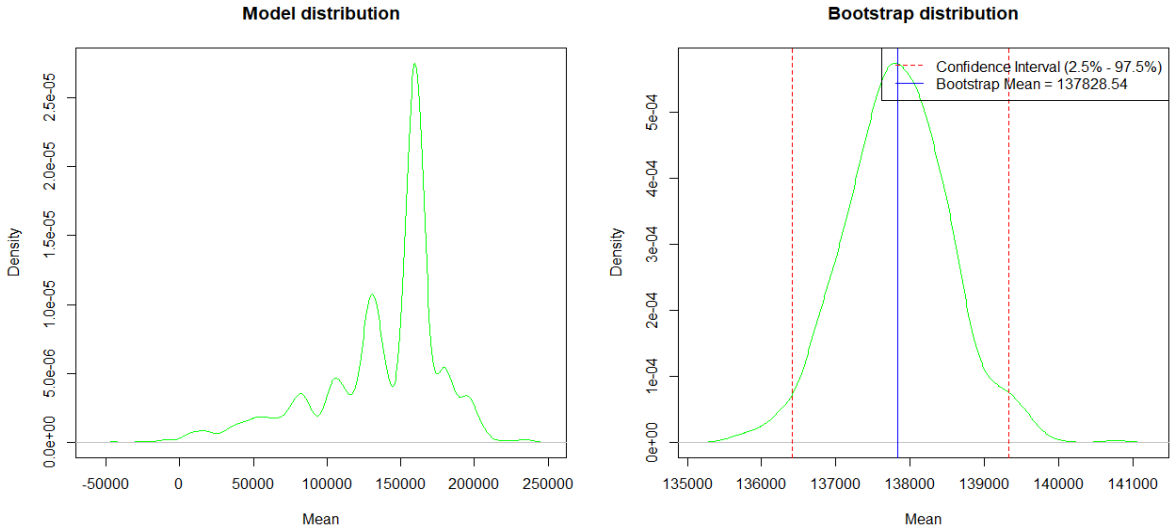
In the parametric bootstrap analysis using the no outliers model, we estimated the mean salary prediction for the new model to be approximately 137,848.2.

To perform the parametric bootstrap, we assumed a normal distribution for the predicted salaries and generated 1000 bootstrap samples based on this assumption.



Comparing the distributions, we found that the predicted salary distribution from the model did not follow a normal distribution. However, the distribution of the bootstrap results appeared to follow a more or less normal distribution.

By examining the distributions and comparing the mean estimates, we can assess the accuracy and precision of the parametric bootstrap method in predicting the population mean of salaries.



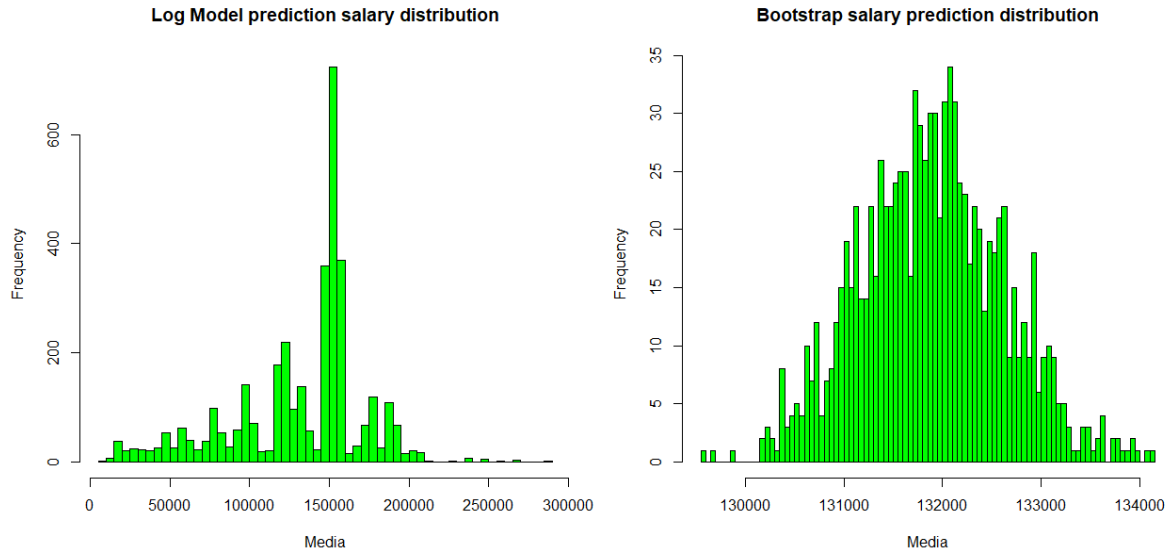
The parametric bootstrap analysis using the no outliers model has a confidence interval of 36,471.8 - 139,155.6 while the confidence interval of the Original Data is 136,483.6 - 139,212.8

Comparing these intervals, we can observe that the bootstrap confidence interval is narrower than the original confidence interval. This indicates that the parametric bootstrap method has provided a more precise estimate of the population parameter

Parametric Bootstrap (Log Model)

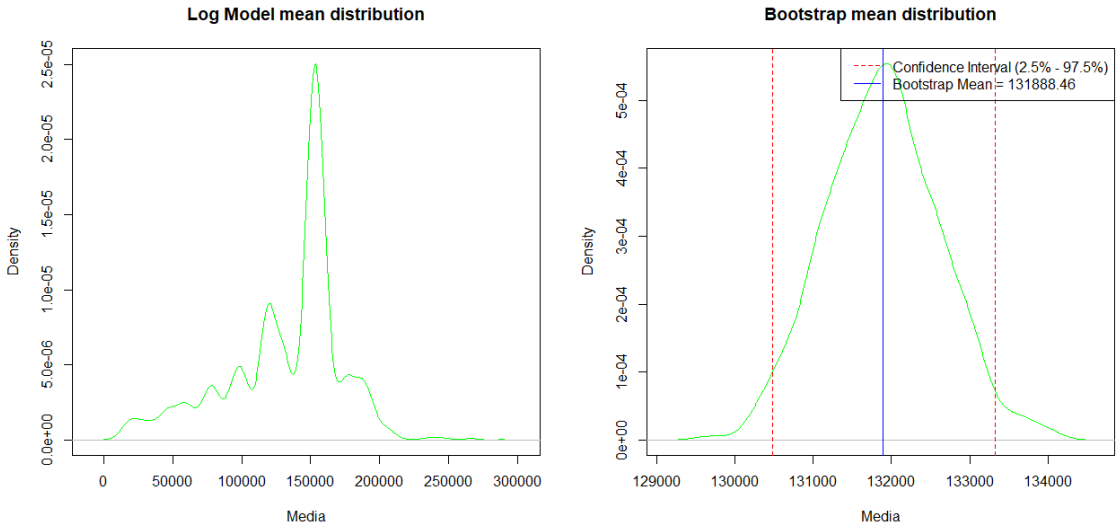
The non-parametric bootstrap analysis using the log model resulted in the following comparisons, the mean salary prediction (Bootstrap parametric) is 131,837.9 and the mean salary (Log Model): 131,847.3

The means of the log model and the original data are quite similar. However, it is important to examine their respective distributions.



In the histogram of predicted salaries from the log model, we can observe that there are no negative predictions due to the log transformation. However, the distribution does not follow a normal distribution. On the other hand, the histogram of the bootstrap results demonstrates a distribution that approximates a normal distribution.

This comparison suggests that the log model's predicted salaries do not follow a normal distribution, while the bootstrap results exhibit a more normal distribution.



General Conclusion

In conclusion, the model demonstrated a good fit to the data, with residuals exhibiting a random scatter around the horizontal line at 0. This indicates that the model accurately captures the relationship between the predictor variables and the target variable.

Moreover, outliers were identified using Cook's distance and subsequently removed from the analysis. By excluding these influential observations, the model's predictions improved, as indicated by the reduced range of residual values.

To address the issue of negative predictions, a logarithmic transformation was applied to the salary data. This transformation ensured that all predictions were positive, providing more meaningful and interpretable results avoiding negative predictions.

Comparing the mean predicted salaries from different models, we observed that the model without outliers and logarithmic transformation resulted in slightly lower average predicted salaries compared to the original model. This suggests that the outliers and logarithmic transformation had an impact on the overall scale of the predictions.

To assess the accuracy and reliability of the predictive models, non-parametric and parametric bootstrap techniques were employed. The bootstrap results provided robust estimates of the mean salaries, taking into account the variability in the data and model predictions.

Overall, the analysis and evaluation of the predictive model provided valuable insights into its performance, the influence of outliers, the impact of logarithmic transformations, and the accuracy of mean salary predictions. These findings contribute to a better understanding of the model's effectiveness in predicting salaries based on the given attributes.

Bibliography and Webgraphy

1. Smith, J. (2021). "A Practical Guide to Bootstrap with R: Examples." Towards Data Science. Retrieved from: Link
2. DataCamp. (n.d.). "Tutorial: Linear Regression in R." DataCamp. Retrieved from: Link
3. DragonflyStats. (n.d.). "Cooks Distance." RPubs. Retrieved from: Link
4. Chaki, A. (2023). "Data Science Salaries 2023." Kaggle. Retrieved from: Link
5. Puig, P. (2023). "Multiple Regression." UAB e-Aules. Retrieved from: Link
6. Puig, P. (2023). "Residuals and Departures." UAB e-Aules. Retrieved from: Link
7. Puig, P. (2023). "Parametric Bootstrap." UAB e-Aules. Retrieved from: Link
8. Puig, P. (2023). "Non-Parametric Bootstrap." UAB e-Aules. Retrieved from: Link

Appendix :R-Scrip

```
#----------------REALM DATA SALARY PREDICTION -------------
library(ggplot2)
#----------------------1 Read dataset---------------------
ds_salaries <- read.csv("ds_salaries.csv", header = TRUE, sep = ",")
#---------------------------------------------------
#----------------------2 Explore dataset-----------
summary(ds_salaries)#We can see the descriptive stadistics for each variable
in the dataset
str(ds_salaries)
cat("The dataset contains", ncol(ds_salaries), "variables.")
#--------------VARIABLES INFORMATION-------------------------
#-------Unique values of the categorical dataset variables--------
variables <- c("experience_level", "employment_type", "job_title",
"salary_currency",
               "employee_residence", "remote_ratio", "company_location",
"company_size")

for (variable in variables) {
  unique_values <- unique(ds_salaries[[variable]])
```

```
  cat("Unique values of", variable, ":", "\n")
  print(unique_values)
  cat("\n")
}
#--------------------MIN MAX MEAN numerical variables--------- """
variables <- c("work_year", "salary_in_usd", "remote_ratio")
for (variable in variables) {
  max_value <- max(ds_salaries[[variable]])
  min_value <- min(ds_salaries[[variable]])
  mean_value <- mean(ds_salaries[[variable]])
  median_value <- median(ds_salaries[[variable]])

  cat("Summary statistics for", variable, ":", "\n")
  cat("Maximum:", max_value, "\n")
  cat("Minimum:", min_value, "\n")
  cat("Mean:", mean_value, "\n")
  cat("Median:", median_value, "\n\n")
}
#We can see that the mean and the median of salary in usd are more or less
equal, this can mean that this variable follows a normal distribution.

#----How are our variables distribution?(Histogram and boxplots)------
# Variables for histograms (numerical variables)
variables <- c("salary_in_usd", "remote_ratio", "work_year")
par(mfrow = c(1, 3))
# Loop through variables and create histograms
for (variable in variables) {
  # Create histogram
  hist(ds_salaries[[variable]], col = "green",
       main = paste("Histogram of", variable),
       xlab = variable, ylab = "Frequency")
}
for (variable in variables) {
  # Create histogram
  boxplot(ds_salaries[[variable]], col = "green",
       main = paste("Histogram of", variable),
       xlab = variable, ylab = "Frequency")
}
par(mfrow = c(1, 1)) #Reset the layout
# Barplots of Categorical Variables (Here there is only some of them as
job_title, employee residence and company location has a lot of variables in
them)
variables <- c("company_size", "work_year", "experience_level")
# Create a layout for displaying multiple plots
par(mfrow = c(1, 3))
# Loop through each variable and create the corresponding barplot
for (variable in variables) {
  barplot(table(ds_salaries[[variable]]),
          main = variable,
          xlab = variable,
          ylab = "Frequency",
          col = "green",
          las = 2)
}
par(mfrow = c(1, 1)) # Reset de layout
```

```
#-----HOW DID COVID19 AFFECT REMOTE RATIO?-----------
# Create a subset of the data with the variables to study
data_subset <- ds_salaries[, c("work_year", "remote_ratio")]
# Calculate the average remote ratio for each work year
avg_remote_ratio <- aggregate(remote_ratio ~ work_year, data = data_subset,
FUN = mean)
ggplot(avg_remote_ratio, aes(x = work_year, y = remote_ratio)) +
  geom_line() +
  geom_point(data = subset(avg_remote_ratio, remote_ratio ==
max(remote_ratio)), color = "red", size = 3) +
  labs(x = "Work Year", y = "Average Remote Ratio", title = "Change in Remote
Work over Work Years")
#-------------------------------------------------------------------
#---------------Interesting Plots and other information------------
# Top 5 professions with highest salary
mean_salary <- aggregate(salary_in_usd ~ job_title, data = ds_salaries, FUN =
mean)
sorted_jobs <- mean_salary[order(mean_salary$salary_in_usd, decreasing =
TRUE),]
top_10_salary <- head(sorted_jobs, n = 10)
top_10_salary
barplot(top_10_salary$salary_in_usd, names.arg = top_10_salary$job_title,
        main = "Top 5 Professions with Highest Salary",
        xlab = "Job", ylab = "Mean Salary", las = 2,col="green")

# Top 5 Cities with highest salary
mean_salary_city <- aggregate(salary_in_usd ~ company_location, data =
ds_salaries, FUN = mean)
sorted_city <- mean_salary_city[order(mean_salary_city$salary, decreasing =
TRUE),]
top_10_city <- head(sorted_city, n = 10)
top_10_city
barplot(top_10_city$salary, names.arg = top_10_city$company_location,
        main = "Top 5 Countries with Highest Salary",
        xlab = "Country", ylab = "Mean Salary",col="green")
# Company sizes with highest salary
mean_salary_company <- aggregate(salary_in_usd ~ company_size, data =
ds_salaries, FUN = mean)
sorted_company <-
mean_salary_company[order(mean_salary_company$salary_in_usd, decreasing =
TRUE),]
top_5_company <- head(sorted_company, n = 5)
top_5_company
barplot(top_5_company$salary_in_usd, names.arg = top_5_company$company_size,
        main = "Top Company Sizes with Highest Salary",
        xlab = "Company Size", ylab = "Mean Salary",col="green")

# Relation between experience and salary
mean_salary_experience <- aggregate(salary_in_usd ~ experience_level, data =
ds_salaries, FUN = mean)
sorted_experience <-
mean_salary_experience[order(mean_salary_experience$salary),]
barplot(sorted_experience$salary, names.arg =
sorted_experience$experience_level,
        main = "Mean Salary by Experience Level",
        xlab = "Experience Level", ylab = "Mean Salary", col = "green")
```

```
sorted_experience
#-------------------------------------------------------------------
#------3.Multiple linear regression with salary in usd as the response
#Is salary needed? (salary in currency units)
correlation <- cor(ds_salaries$salary, ds_salaries$salary_in_usd)
correlation
#In this dataset we have three variables that are related with the salary and
I want to predict it with the same currency units
#So in order to simplify my model and focus in those variables that have
stronger impact on the target variable I removed them
ds_salaries <- subset(ds_salaries, select = -c(salary,
salary_currency))#subset without salary ans salary_currency
head(ds_salaries)
#---------VARIABLE SELECTION ( BACKWARD, FORWARD, STEPWISE)-----
#---------------------backward selection-----------
library(MASS)
lm_initial <- lm(salary_in_usd~ ., data = ds_salaries)
lm_backward <- stepAIC(lm_initial, direction = "backward")
anova(lm_backward)
#----------------------forward selection------------------
null_model <- lm(salary_in_usd ~ 1, data = ds_salaries)
full_model <-lm(salary_in_usd~ ., data = ds_salaries)

# Running forward stepwise selection with AIC to choose the best model
library("MASS")
model_forward <- stepAIC(null_model, direction="forward",
scope=list(lower=null_model, upper=full_model), trace=TRUE)
model_forward
anova(model_forward)

#----------------------Stepwise selection-----------------------
null_model <- lm(salary_in_usd ~ 1, data = ds_salaries)
full_model <- lm(salary_in_usd ~ ., data = ds_salaries)
# Running stepwise selection with AIC to choose the best model
library("MASS")
model_stepwise <- stepAIC(null_model, direction="both",
scope=list(lower=null_model, upper=full_model), trace=TRUE)
model_stepwise
anova(model_stepwise)
#-------------------------------------------------------------
#Same model in each selection.
#---------------------------------------------

#---------------------------4 MODEL STUDY ---------------------
model=lm(salary_in_usd~ job_title + company_size + work_year +
experience_level + employee_residence,data=ds_salaries)
summary(model)
#----------------------COEFFICIENT EXAMINATION-----------------
# Get the coefficients
coefficients <- coef(model)
# Get the 95% confidence intervals for the coefficients
conf_intervals <- confint(model)
p_values <- summary(model)$coefficients[, 4]
# Create a data frame with coefficients, confidence intervals, and p-values
coef_summary <- data.frame(
  Coefficient = coefficients,
```

```
  Lower_CI = conf_intervals[, 1],
  Upper_CI = conf_intervals[, 2],
  p_value = p_values
)
# Order the coefficients by absolute value
coef_summary <- coef_summary[order(-abs(coef_summary$Coefficient)), ]
print(coef_summary)
#Possitive coefficients = If predictor variable increases, the response
variable
                         #is also expected to increase.
#Negative coefficients = If predictor variable decrease, the response
variable
                         #is also expected to decrease.
#-------------------------------------------------------------------
#-----------Significant Predictor Variables (sort by p-value)----
# Get the p-values associated with the coefficients
p_values <- summary(model)$coefficients[, 4]
sorted_indices <- order(p_values)
sorted_coefficients <- coefficients[sorted_indices]
sorted_p_values <- p_values[sorted_indices]
sorted_coef_summary <- data.frame(Coefficient = sorted_coefficients,
                                  p_value = sorted_p_values)
print(sorted_coef_summary)

#-------------------------------------------------------------------
#-------------------------RESIDUALS----------------------------------
#In order to have more information about our first model, we did a plot of
the residuals
#If the model is good we would expect the plot to show a random scatter of
points around the horizontal line at 0.
#We also know that any trends in the residuals indicates a bad model to fit
the data.
plot(residuals(model),type = "b",ylim = range(residuals(model, type =
"pearson")))#It does not follow any tendence
#but it has outliers
abline(h = 0)
#To have evidence that the model does not fit good the data, is important to
do a residual histogram and see if the resiudals are normally distributed or
not.
#If they are, that means that our model is a good fit.
#As the histogram is slightly skewed to the left, which means that the model
might be overestimating the number of affected individuals for some time
points.
#However, the skewness is not too extreme and that is a good sign.

hist(residuals(model), breaks = 100, col = "green",main="Residuals
Histogram")

#It seems that there are outliers, so I study them applying cook's distance.
#Cook's distance is a mesure used to identify influencial observations in a
regression analysis.
#If the outliers are influential, it will be excluded from the model.
#----------------------Cook's distance---------------------------

model <- lm(salary_in_usd ~ job_title + company_size + work_year +
experience_level + employee_residence, data = ds_salaries)
```

```
# Calculate Cook's distance
cooksd <- cooks.distance(model)
# Identify outliers
outliers <- cooksd > 4 / length(cooksd)
# Exclude the outliers
ds_salaries_no_outliers <- ds_salaries[!outliers, ]
# Refit the model without the outliers
new_model <- lm(salary_in_usd ~ job_title + company_size + work_year +
experience_level + employee_residence, data = ds_salaries_no_outliers)
plot(residuals(new_model),type="b")
hist(residuals(new_model), breaks = 100, col = "green",main="Residuals
Histogram")

#-------------------------------------------------------------------
#----------------------PREDICTIONS----------------------------------
#We can make a comparison between the predictions with outliers and the new
model.
pred_model<-predict(model)

#First of all, check there are negative salary predictions
all_positive <- all(pred_model > 0)
if (all_positive) {
  print("All predictions are positive.")
} else {
  print("Not all predictions are positive.")
}

#Problem: Salaries can't be negative
#Solution:Compute logarithms to the predictions.(They will be in a different
scale
#         but we can reverse it with exp())

#---------------LOG SALARY MODEL---------------------------------
ds_salaries_no_outliers$log_salary <-
log(ds_salaries_no_outliers$salary_in_usd)#new variable
model_log <- lm(log_salary ~ job_title + company_size + work_year +
experience_level + employee_residence, data =  ds_salaries_no_outliers)
#No outliers dataset is used.
log_predictions <- predict(model_log)
hist(log_predictions)

#Check if all log predictions are possitive
all_positive <- all(log_predictions > 0)
if (all_positive) {
  print("All predictions are positive.")
} else {
  print("Not all predictions are positive.")
}
#All predictions are possitive
# Reverse the logarithmic transformation to obtain predictions in the
original scale
scaled_log_pred <- exp(log_predictions)
#Check if all log predictions are possitive
all_positive <- all(scaled_log_pred > 0)
if (all_positive) {
  print("All predictions are positive.")
```

```
} else {
  print("Not all predictions are positive.")
}
#Brief comparison all models mean
#Mean of the predicted values of both models
mean_pr_model <- mean(pred_model)
mean_pr_nmodel <- mean(pred_new_model)
mean_log_model <- mean(scaled_log_pred)
mean_models <- c(mean_pr_model, mean_pr_nmodel, mean_log_model)
sorted_means <- sort(mean_models)
model_names <- c("pred_model", "pred_new_model", "scaled_log_pred")
sorted_models <- model_names[order(mean_models)]
result <- data.frame(Model = sorted_models, Mean = sorted_means)
result


#OBSERVATION:
#The new model without outliers produces higher mean predictions compared to
the model with outliers.
#This suggests that the outliers were influencing the average predicted
salaries downward.
#The log mean is lower than the others (Scale problems?)

#----------- Comparing Model's Predictions-------------------------
#PREDICTION 1
values1 <- data.frame(job_title = 'Data Scientist', company_size = 'L',
work_year = 2023, experience_level = 'EX',employee_residence='US')
# Predict using the original model
prediction <- predict(model, newdata = values1)
cat("Model: Original Model\n")
cat("Prediction:", prediction, "\n\n")

# Predict using the new model without outliers
predictionnew <- predict(new_model, newdata = values1)
cat("Model: New Model without Outliers\n")
cat("Prediction:", predictionnew, "\n\n")

# Predict using the model with logarithm transformation
log_pred <- exp(predict(model_log, newdata = values1))
cat("Model: Model with Logarithm Transformation\n")
cat("Prediction:", log_pred, "\n\n")

#PREDICTION 2
values2 <- data.frame(job_title = 'Data Scientist', company_size = 'M',
work_year = 2023, experience_level = 'EX', employee_residence = 'US')
# Predict using the original model
prediction <- predict(model, newdata = values2)
cat("Model: Original Model\n")
cat("Prediction:", prediction, "\n\n")

# Predict using the new model without outliers
predictionnew <- predict(new_model, newdata = values2)
cat("Model: New Model without Outliers\n")
cat("Prediction:", predictionnew, "\n\n")

# Predict using the model with logarithm transformation
```

```
log_pred <- exp(predict(model_log, newdata = values2))
cat("Model: Model with Logarithm Transformation\n")
cat("Prediction:", log_pred, "\n\n")

#PREDICTION 3
values3 <- data.frame(job_title = 'Data Engineer', company_size = 'S',
work_year = 2023, experience_level = 'SE', employee_residence = 'US')
# Predict using the original model
prediction <- predict(model, newdata = values3)
cat("Model: Original Model\n")
cat("Prediction:", prediction, "\n\n")

# Predict using the new model without outliers
predictionnew <- predict(new_model, newdata = values3)
cat("Model: New Model without Outliers\n")
cat("Prediction:", predictionnew, "\n\n")

# Predict using the model with logarithm transformation
log_pred <- exp(predict(model_log, newdata = values3))
cat("Model: Model with Logarithm Transformation\n")
cat("Prediction:", log_pred, "\n\n")

#--------------BOOTRSAP STUDY FOR MEAN ACCURACY------------------

#---NON PARAMETRIC BOOOSTRAP (MODEL NO OUTLIERS)-----------
predicted_salaries<-predict(new_model)
mean_salary=mean(predicted_salaries)#137848.2
# Create a vector to store the bootstrap results
bootstrap_results <- vector("numeric", 10000)
# Loop through the bootstrap iterations
for(i in 1:10000){
  # Sample with replacement from the predicted salaries
  bootstrap_sample <- sample(predicted_salaries, size =
length(predicted_salaries), replace = TRUE)
  # Calculate the mean of the bootstrap sample
  bootstrap_mean <- mean(bootstrap_sample)
  # Store the bootstrap mean in the bootstrap_results vector
  bootstrap_results[i] <- bootstrap_mean
}

# Comparison of distributions NON PARAMETRIC (MODEL NO OUTLIERS)
bootstrap_mean <- mean(bootstrap_results)
bootstrap_mean#137847.6 (In my execution)
mean_salary#137848.2
#The means are mostly equal, lets see how they are distributed.
# Histogram and density plots (data with no outliers vs nonparametric
Bootrsap)
par(mfrow = c(1, 2))
hist((predicted_salaries), main = "Model prediction salary distribution",
xlab = "Mean", col = "green",breaks=50)
hist((bootstrap_results), main = "Bootstrap salary prediction distribution",
xlab = "Mean", col = "green",breaks=50)
#We can see that there are negative predictions on the models distribution
and
#that it doesn't follow a normal distribution around the mean value
```

```
#The histogram of bootrsap results, we can see that there are no negative
predictions,
#and that it follows a normal distribution.
par(mfrow = c(1, 1))
par(mfrow = c(1, 2))


#Condidence Interval NON PARAMETRIC BOOTSTRAP (NO OUTLIERS MODEL)
#----Bootstrap IC-------------
alpha <- 0.05
sorted_results <- sort(bootstrap_results)
lower_limit <- sorted_results[round(alpha/2 * length(sorted_results))]
upper_limit <- sorted_results[round((1 - alpha/2) * length(sorted_results))]
cat("Confidence Interval:", lower_limit, "-", upper_limit, "\n")#136471.8 -
139155.6

# ------Confidence interval of THE ORGINAL DATA-------------
se <- sd(predicted_salaries) / sqrt(length(predicted_salaries))
t <- qt(0.975, df = length(predicted_salaries) - 1)
# Confidence Interval
ci_lower <- mean(predicted_salaries) - t * se
ci_upper <- mean(predicted_salaries) + t * se
cat("Confidence Interval:", ci_lower, "-", ci_upper, "\n")#136483.6 -
139212.8


#-------------------------DENSITY PLOTS-----------------------
plot(density(predicted_salaries), main = "Model mean distribution", xlab =
"Mean", col = "green")
plot(density(bootstrap_results), main = "Bootstrap mean distribution", xlab =
"Mean", col = "green")
abline(v=upper_limit,col="red",lty=2)
abline(v=lower_limit,col="red",lty=2)
abline(v=bootstrap_mean,col="blue",lty=1)
legend_text <- c("Confidence Interval (2.5% - 97.5%)", paste("Bootstrap Mean
=", round(bootstrap_mean, 2)))
legend("topright", legend = legend_text, col = c("red", "blue"), lty = c(2,
1))
par(mfrow = c(1, 1))

#-----------------------------------------------------------------------
#The bootstrap confidence interval is narrower than the original confidence
interval,
#It suggests that the bootstrap method has provided a more precise estimate
of the population parameter.

#-----------------------THE SAME STUDY FOR THE LOG MODEL--------
predicted_salaries<-exp(predict(model_log))
mean_salary=mean(predicted_salaries)
bootstrap_results <- vector("numeric", 10000)
for(i in 1:10000){
  bootstrap_sample <- sample(predicted_salaries, size =
length(predicted_salaries), replace = TRUE)
  bootstrap_mean <- mean(bootstrap_sample)
  # Store the bootstrap mean in the bootstrap_results vector
  bootstrap_results[i] <- bootstrap_mean
```

```
}

#------ Comparison of distributions NON PARAMETRIC (LOG MODEL)-----
bootstrap_mean <- mean(bootstrap_results)
bootstrap_mean#131837.9
mean_salary#131847.3
#The means are mostly equal, lets see how the are distributed.
# Histograms (data with no outliers vs nonparametric Bootrsap)
par(mfrow = c(1, 2))
hist((predicted_salaries), main = "Log Model prediction salary distribution",
xlab = "Mean", col = "green",breaks=80)
hist((bootstrap_results), main = "Bootstrap salary prediction distribution",
xlab = "Mean", col = "green",breaks=80)
#We can see that there are NO negative predictions in any model distribution
due to the log transformation
#The predicred_salaries does NOT FOLLOW a normal distribution
#The bootstrap results follows a normal distribution.
par(mfrow = c(1, 1))
par(mfrow = c(1, 2))

#  ---Condidence Interval NON PARAMETRIC BOOTSTRAP (LOG MODEL)--------
#---------------------------Bootrsap IC-------------
alpha <- 0.05
sorted_results <- sort(bootstrap_results)
lower_limit <- sorted_results[round(alpha/2 * length(sorted_results))]
upper_limit <- sorted_results[round((1 - alpha/2) * length(sorted_results))]
cat("Confidence Interval:", lower_limit, "-", upper_limit, "\n") #130468.7 -
133217.7

# ---------Confidence interval of THE ORGINAL DATA---------
se <- sd(predicted_salaries) / sqrt(length(predicted_salaries))
t <- qt(0.975, df = length(predicted_salaries) - 1)
# Confidence Interval
ci_lower <- mean(predicted_salaries) - t * se
ci_upper <- mean(predicted_salaries) + t * se
cat("Confidence Interval:", ci_lower, "-", ci_upper, "\n") #130474.6 - 133220

#-------------------------DENSITY PLOTS-------------------------
par(mfrow = c(1, 2))
plot(density(predicted_salaries), main = "Log Model distribution", xlab =
"Mean", col = "green")
plot(density(bootstrap_results), main = "Bootstrap distribution", xlab =
"Mean", col = "green")

abline(v=upper_limit,col="red",lty=2)
abline(v=lower_limit,col="red",lty=2)
abline(v=bootstrap_mean,col="blue",lty=1)

legend_text <- c("Confidence Interval (2.5% - 97.5%)", paste("Bootstrap Mean
=", round(bootstrap_mean, 2)))
legend("topright", legend = legend_text, col = c("red", "blue"), lty = c(2,
1))
par(mfrow = c(1, 1))

#-------------------------PARAMETRIC BOOTSTRAP----------------
```

```
#-----------PARAMETRIC BOOSTRAP NORMAL DISTRIBUTION (NO OUTLIERS
MODEL)----------------------------------------
#mean salary prediction for newmodel 137848.2

predicted_salaries<-predict(new_model)
model <- lm(salary_in_usd ~ job_title + company_size + work_year +
experience_level + employee_residence, data = ds_salaries_no_outliers)
coefficients <- coef(model)
sqrt_residual_variance <- sigma(model)
bootstrap_results <- vector("numeric", 1000)

for (i in 1:1000) {
  bootstrap_sample <- rnorm(length(predicted_salaries), mean =
predicted_salaries, sd =sqrt_residual_variance)
  bootstrap_mean <- mean(bootstrap_sample)
  bootstrap_results[i] <- bootstrap_mean
}
bootstrap_mean <- mean(bootstrap_results)# 137893.5

# ----Comparison of distributions (NO OUTLIERS MODEL)----------
par(mfrow = c(1, 2))
hist((predicted_salaries), main = "Model salary prediction distribution",
xlab = "Mean", col = "green",breaks=80)
#The predicted values does not follow a normal distribution.
hist((bootstrap_results), main = "", xlab = "Mean", col = "green",breaks=80)
#Bootstrap Results follow more or less a normal distribution
par(mfrow = c(1, 1))
par(mfrow = c(1, 2))
#  ---Condidence Interval PARAMETRIC BOOTSTRAP (NO OUTLIERS MODEL)----

alpha <- 0.05
sorted_results <- sort(bootstrap_results)
lower_limit <- sorted_results[round(alpha/2 * length(sorted_results))]
upper_limit <- sorted_results[round((1 - alpha/2) * length(sorted_results))]
cat("Confidence Interval:", lower_limit, "-", upper_limit, "\n") #136459.3 -
139340.7
plot(density(predicted_salaries), main = "Model distribution", xlab = "Mean",
col = "green")
plot(density(bootstrap_results), main = "Bootstrap distribution", xlab =
"Mean", col = "green")
abline(v=upper_limit,col="red",lty=2)
abline(v=lower_limit,col="red",lty=2)
abline(v=bootstrap_mean,col="blue",lty=1)
legend_text <- c("Confidence Interval (2.5% - 97.5%)", paste("Bootstrap Mean
=", round(bootstrap_mean, 2)))
legend("topright", legend = legend_text, col = c("red", "blue"), lty = c(2,
1))
par(mfrow = c(1, 1))


#----------SAME STUDY FOR LOG MODEL--------------------
mean_salary=mean(predicted_salaries)
coefficients <- coef(model_log)
sqrt_residual_variance <- sigma(model)
bootstrap_results <- vector("numeric", 1000)
```

```
for (i in 1:1000) {
  bootstrap_sample <- rnorm(length(predicted_salaries), mean =
predicted_salaries, sd =sqrt_residual_variance)
  bootstrap_mean <- mean(bootstrap_sample)
  bootstrap_results[i] <- bootstrap_mean
}

#---- Comparison of distributions NON PARAMETRIC (LOG MODEL)-----
bootstrap_mean <- mean(bootstrap_results)
bootstrap_mean#131866
mean_salary#131847.3
#The means are mostly equal, lets see how they are distributed.
# Histograms (data with no outliers vs nonparametric Bootrsap)
par(mfrow = c(1, 2))
hist((predicted_salaries), main = "Log Model prediction salary distribution",
xlab = "Mean", col = "green",breaks=80)
hist((bootstrap_results), main = "Bootstrap salary prediction distribution",
xlab = "Mean", col = "green",breaks=80)
#We can see that there are NO negative predictions in any model distribution
due to the log transformation
#The predicred_salaries does NOT FOLLOW a normal distribution
#The bootstrap results follows a normal distribution.
par(mfrow = c(1, 1))
par(mfrow = c(1, 2))

#  ----Condidence Interval NON PARAMETRIC BOOTSTRAP (LOG MODEL)------
alpha <- 0.05
sorted_results <- sort(bootstrap_results)
lower_limit <- sorted_results[round(alpha/2 * length(sorted_results))]
upper_limit <- sorted_results[round((1 - alpha/2) * length(sorted_results))]
cat("Confidence Interval:", lower_limit, "-", upper_limit, "\n")

# ---Confidence interval of THE ORGINAL DATA-------
se <- sd(predicted_salaries) / sqrt(length(predicted_salaries))
t <- qt(0.975, df = length(predicted_salaries) - 1)
# Confidence Interval
ci_lower <- mean(predicted_salaries) - t * se
ci_upper <- mean(predicted_salaries) + t * se
cat("Confidence Interval:", ci_lower, "-", ci_upper, "\n") #130474.6 - 133220
#-------------------------------------------------------------------------

plot(density(predicted_salaries), main = "Log Model mean distribution", xlab
= "Mean", col = "green")
plot(density(bootstrap_results), main = "Bootstrap mean distribution", xlab =
"Mean", col = "green")

abline(v=upper_limit,col="red",lty=2)
abline(v=lower_limit,col="red",lty=2)
abline(v=bootstrap_mean,col="blue",lty=1)

legend_text <- c("Confidence Interval (2.5% - 97.5%)", paste("Bootstrap Mean
=", round(bootstrap_mean, 2)))
legend("topright", legend = legend_text, col = c("red", "blue"), lty = c(2,
1))
par(mfrow = c(1, 1))
```