Stat 408 Final Project

Stone Cai

2024-03-23

Repository link: https://github.com/scai1/Stat-408-Final-Project . Please refer to repository for version controls for this Rmarkdown, raw data set, final knitted PDF, and country letter code.txt file.

Introduction

As someone pursuing a masters in applied statistics with the goal of a career change, it is an intriguing to question the career opportunities available after finishing this degree. What kind of salary and compensation can one expect if pursuing a career in data science/data analyst,etc? For this final project, I am considering a ds_salaries dataset which tracks the salaries of n= 3755 data science related careers from the years 2020-2023. I hope to get a general sense of the salaries and compensation expectations of data science related careers. Additionally, I hope to answer questions such as which factors/attributes are significant contributors to overall salary? What is the best model from among the dataset's variables that best explains salary expectations? On a surface level, are there are any predictors in the initial dataset that are redundant or collinear?

To approach this problem, I am dividing this report into these sections:

1. Data Exploration and Preparation.

2. Model Building+ Selection
3. Interpretation of Results
4. Model refinement (Identifying outliers and checking influenceplots)
5. Evaluation and discussion.

Section 1: Data Exploration and Preparation Here, I am going to recode, clean, and collapse variables appropriately in a way that will allow me to do analysis as I see fit.

```
salary<- read.csv("C:/Users/stone/Documents/Stat 408/Final Project/ds_salaries.csv")

View(salary)

summary(salary)
```

```
##    work_year    experience_level   employment_type     job_title
##  Min.   :2020   Length:3755        Length:3755        Length:3755
##  1st Qu.:2022   Class :character   Class :character   Class :character
##  Median :2022   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2022
##  3rd Qu.:2023
##  Max.   :2023
##      salary         salary_currency    salary_in_usd    employee_residence
##  Min.   :    6000   Length:3755        Min.   :  5132   Length:3755
```

```
##  1st Qu.:  100000   Class :character   1st Qu.: 95000   Class :character
##  Median :  138000   Mode  :character   Median :135000   Mode  :character
##  Mean   :  190696                      Mean   :137570
##  3rd Qu.:  180000                      3rd Qu.:175000
##  Max.   :30400000                      Max.   :450000
##   remote_ratio    company_location   company_size
##  Min.   :  0.00   Length:3755        Length:3755
##  1st Qu.:  0.00   Class :character   Class :character
##  Median :  0.00   Mode  :character   Mode  :character
##  Mean   : 46.27
##  3rd Qu.:100.00
##  Max.   :100.00
```

```
##print(salary[order(salary$salary_in_usd, decreasing = TRUE), ]   )
```

I am using the data set ds_salaries from Kaggle.com,which tracked data science salaries from 2020-2023. Here, I have entered the data science salaries data set. From the initial summary of the dataset, I will determine how I need to clean data and re-code certain columns that I may want to use as predictors in a linear model. Here, I summarize the variables from the data card for the data set.

1.work_year: The year the salary was paid.

2.experience_level: The experience level in the job during the year with the following possible values: (EN) Entry-level / Junior (MI) Mid-level / Intermediate (SE) Senior-level / Expert (EX) Executive-level / Director.

3.employment_type: The type of employment for the role: PT: Part-time FT: Full-time CT: Contract FL: Freelance.

4.job_title: The role worked in during the year.

5.salary: The total gross salary amount paid.

6.salary_currency: The currency of the salary paid as an ISO 4217 currency code.

7.salary_in_usd: The salary in USD (FX rate divided by avg. USD rate for the respective year via fx-data.foorilla.com).

8.employee_residence: Employee's primary country of residence in during the work year as an ISO 3166 country code.

9.Remote_ratio: The overall amount of work done remotely, possible values are as follows: 0= No remote work (less than 20%) 50= Partially remote 100= Fully remote (more than 80%).

10.company_location The country of the employer's main office or contracting branch as an ISO 3166 country code.

11.company_size The average number of people that worked for the company during the year: S= less than 50 employees (small) M= 50 to 250 employees (medium) L= more than 250 employees (large)

From this initial summary I notice several issues.There are several potential redundant or irrelevant columns. As I've discussed in my introduction, I'm interested in data scientists' salary as the response variable and utilizing various predictors to determine what are the key factors that determines a data scientist's salary. Although it is nice to understand the currency that all employees in this data set are paid in, the variable salary_in_usd is the only relevant column because I can compare all the salaries in the same currency. Thus, I will remove both salary_currency and salary variables from the dataset, as the information is all contained within salary_in_USD.

```r
salary1 <- subset( salary, select = -c(salary,salary_currency) )
##View(salary1)

table(salary1$work_year)
```

```
##
## 2020 2021 2022 2023
##   76  230 1664 1785
```

```r
table(salary1$experience_level)
```

```
##
##   EN   EX   MI   SE
##  320  114  805 2516
```

```r
##table(salary1$job_title)

table(salary1$employment_type)
```

```
##
##   CT   FL   FT   PT
##   10   10 3718   17
```

```r
table(salary1$employee_residence)
```

```
##
##   AE   AM   AR   AS   AT   AU   BA   BE   BG   BO   BR   CA   CF   CH   CL   CN
##    3    1    6    2    6   11    1    5    1    3   18   85    2    4    2    1
##   CO   CR   CY   CZ   DE   DK   DO   DZ   EE   EG   ES   FI   FR   GB   GH   GR
##    4    1    1    2   48    3    1    1    1    1   80    2   38  167    2   16
##   HK   HN   HR   HU   ID   IE   IL   IN   IQ   IR   IT   JE   JP   KE   KW   LT
##    2    1    3    3    1    7    1   71    1    1    8    1    7    2    1    2
##   LU   LV   MA   MD   MK   MT   MX   MY   NG   NL   NZ   PH   PK   PL   PR   PT
##    1    4    1    1    1    1   10    1    7   15    1    2    8    6    5   18
##   RO   RS   RU   SE   SG   SI   SK   TH   TN   TR   UA   US   UZ   VN
##    3    1    4    2    5    4    1    3    1    5    4 3004    2    3
```

```r
table(salary1$remote_ratio)
```

```
##
##    0   50  100
## 1923  189 1643
```

```r
table(salary1$company_location)
```

```
##
##   AE   AL   AM   AR   AS   AT   AU   BA   BE   BO   BR   BS   CA   CF   CH   CL
##    3    1    1    3    3    6   14    1    4    1   15    1   87    2    5    1
##   CN   CO   CR   CZ   DE   DK   DZ   EE   EG   ES   FI   FR   GB   GH   GR   HK
```

```
##    1    4    1    3   56    4    1    2    1   77    3   34  172    2   14    1
##   HN   HR   HU   ID   IE   IL   IN   IQ   IR   IT   JP   KE   LT   LU   LV   MA
##    1    3    2    2    7    2   58    1    1    4    6    2    2    3    4    1
##   MD   MK   MT   MX   MY   NG   NL   NZ   PH   PK   PL   PR   PT   RO   RU   SE
##    1    1    1   10    1    5   13    1    1    4    5    4   14    2    3    2
##   SG   SI   SK   TH   TR   UA   US   VN
##    6    4    1    3    5    4 3040    1
```

```r
table(salary1$company_size)
```

```
##
##    L    M    S
##  454 3153  148
```

Here, I'm identifying the counts of every variable in the data set so I can get an intial sense of what the data looks like and see how I can appropriately code factors for these variables. Here, I notice a few things. Most of the datasets' observations for work year fall into 2022 and 2023. Here, I have the option to to code as either a continuous or categorical variable. While over a larger a range of years, I would intuitively recommend coding year as continuous, I know that inflation in this 4 year span was extremely anomalous. According to the federal reserve, inflation in 2020 was only 1.4% due to covid, while 2021 and 2022 were 7% and 6.5% respectively. Here, the swings in inflation are non-standard so I think it makes sense to code work_year as a categorical variable with 4 factors so as to compare salaries from each year as a category. Below, I go ahead and code all existing variables with their respective factors.

```r
salary1$experience_level <- factor(salary1$experience_level)
levels(salary1$experience_level) <- c("EN", "Ex", "MI", "SE")

salary1$work_year <- factor(salary1$work_year)
levels(salary1$work_year) <- c("2020", "2021", "2022", "2023")

salary1$employment_type <- factor(salary1$employment_type)
levels(salary1$employment_type) <- c("CT", "FL", "FT", "PT")

salary1$remote_ratio <- factor(salary1$remote_ratio)
levels(salary1$remote_ratio)<- c("0", "50", "100" )

salary1$company_size <- factor(salary1$company_size)
levels(salary1$company_size)<- c("L", "M", "S" )

summary(salary1)
```

```
##  work_year   experience_level employment_type  job_title
##  2020:  76   EN: 320          CT:   10         Length:3755
##  2021: 230   Ex: 114          FL:   10         Class :character
##  2022:1664   MI: 805          FT:3718          Mode  :character
##  2023:1785   SE:2516          PT:   17
##
##
##  salary_in_usd    employee_residence remote_ratio company_location
##  Min.   :  5132   Length:3755        0  :1923     Length:3755
##  1st Qu.: 95000   Class :character   50 : 189     Class :character
##  Median :135000   Mode  :character   100:1643     Mode  :character
```

```
##  Mean   :137570
##  3rd Qu.:175000
##  Max.   :450000
##  company_size
##  L: 454
##  M:3153
##  S: 148
##
##
##
```

The last 3 variables I need to clean/ recode are job_title, company_location and employee_residence. For company_location and employee residence I'm looking to collapse and consolidate the variable into fewer categories. Given that most observations are from the "US", I believe it makes sense to collapse these variables into US, EU, and other so as to have enough observations for each category and not have standard errors that are too large.

```r
salary1[ , 'employee_residence1'] <- NA
salary1[ , 'company_location1'] <- NA



library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
for(i in 1:length(salary1$employee_residence))
{
  if(salary1$employee_residence[i]=="CA" || salary1$employee_residence[i]=="US" )
  {
    salary1$employee_residence1[i] <- "US"

  }


 else if(salary1$employee_residence[i]== "ES" || salary1$employee_residence[i]== "DE" || salary1$employo

  {
    salary1$employee_residence1[i] <- "EU"

  }
```

```r
 else
  {
    salary1$employee_residence1[i] <- "OT"

  }


}

######################### cleaning company_location data

for(i in 1:length(salary1$company_location))
{
  if(salary1$company_location[i]=="CA" || salary1$company_location[i]=="US" )
  {
    salary1$company_location1[i] <- "US"

  }


 else if(salary1$company_location[i]== "ES" || salary1$company_location[i]== "DE" || salary1$company_lo

  {
    salary1$company_location1[i] <- "EU"

  }

 else
  {
    salary1$company_location1[i] <- "OT"

  }


}

##View(salary1)
table(salary1$employee_residence1)
```

```
##
##   EU   OT   US
##  459  207 3089
```

```r
table(salary1$company_location1)
```

```
##
##   EU   OT   US
##  454  174 3127
```

```r
salary2<- subset( salary1, select = -c(employee_residence,company_location) )
View(salary2)
```

First, I created a text document to match full country names to their 2 letter code names. (See attached .txt doucment in github). After, I match country names to their 2 letter code names, I determined which countries were part of Europe, US/Canada, and not either of the first 2.

Here, I collapsed the variable of employee residence and company location. First, I create 2 new columns called employee_residence1 and company_location1 as I will append these re-coded values to these new columns. I combined Canada counts with US counts under "US". I believe fundamentally, Canadian tech companies function very similarly to the US, operate under the same language,etc. So even though I don't necessarily believe Canadian data science salaries are homogeneous to US data science salaries, I believe this makes more sense then coding Canada as "Other". I also collapsed all European countries into one EU factor and the remaining countries will fall under "OT" or other. From the frequency count after I re-coded both variables I believe this is a reasonable way to re-code the observations. Here, I believe I have enough observations for my standard errors to be reasonable for reach factor. I could further break down the remaining countries geographically, but I'm afraid I have too few observations for these new factors and thus have standard errors that are extremely large. I'm choosing to collapse the variables like this because the vast majority of the ~3800 observations are from the US/Canada in both cases. Finally, I drop my original data frame columns from the re-coded dataframe.Note: One additional issue I believe I have here is multi-collinearity. The vast majority of people in this data set typically live in the same country as the Company that they work for. Obviously, these variables are not mirrored, but there's very likely some redundancy here. I will verify using a VIF calculation in a later section in this report, but I will likely have to drop either (employee_residence1 or company_location1) in my final model.

```
salary2$employee_residence1 <- factor(salary2$employee_residence1)
levels(salary2$employee_residence1)<- c("EU", "OT", "US" )

salary2$company_location1 <- factor(salary2$company_location1)
levels(salary2$company_location1)<- c("EU", "OT", "US" )

##summary(salary2)
```

In this step, I assign the appropriate factors for EU,OT. and US.

```
##install.packages("tidytext")
library(tidytext)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(dplyr)


salary2 %>% count(job_title, sort = TRUE)
```

```
##                              job_title    n
## 1                        Data Engineer 1040
## 2                       Data Scientist  840
## 3                         Data Analyst  612
## 4            Machine Learning Engineer  289
## 5                   Analytics Engineer  103
## 6                       Data Architect  101
## 7                   Research Scientist   82
## 8                    Applied Scientist   58
## 9                 Data Science Manager   58
```

```
## 10                          Research Engineer   37
## 11                                ML Engineer   34
## 12                               Data Manager   29
## 13                  Machine Learning Scientist   26
## 14                     Data Science Consultant   24
## 15                      Data Analytics Manager   22
## 16                    Computer Vision Engineer   18
## 17                               AI Scientist   16
## 18                            BI Data Analyst   15
## 19                      Business Data Analyst   15
## 20                            Data Specialist   14
## 21                               BI Developer   13
## 22          Applied Machine Learning Scientist   12
## 23                               AI Developer   11
## 24                          Big Data Engineer   11
## 25                    Director of Data Science   11
## 26 Machine Learning Infrastructure Engineer   11
## 27                     Applied Data Scientist   10
## 28                    Data Operations Engineer   10
## 29                              ETL Developer   10
## 30                               Head of Data   10
## 31         Machine Learning Software Engineer   10
## 32                                 BI Analyst    9
## 33                       Head of Data Science    9
## 34                        Lead Data Scientist    9
## 35                          Data Science Lead    8
## 36                   Principal Data Scientist    8
## 37                      Data Quality Analyst    7
## 38                  Machine Learning Developer    7
## 39                               NLP Engineer    7
## 40                     Data Analytics Engineer    6
## 41               Data Infrastructure Engineer    6
## 42                      Deep Learning Engineer    6
## 43                         Lead Data Engineer    6
## 44                 Machine Learning Researcher    6
## 45                     Cloud Database Engineer    5
## 46          Computer Vision Software Engineer    5
## 47                      Data Science Engineer    5
## 48                          Lead Data Analyst    5
## 49                       Product Data Analyst    5
## 50              3D Computer Vision Researcher    4
## 51            Business Intelligence Engineer    4
## 52                     Data Operations Analyst    4
## 53                             MLOps Engineer    4
## 54         Machine Learning Research Engineer    4
## 55                        Cloud Data Engineer    3
## 56                      Financial Data Analyst    3
## 57            Lead Machine Learning Engineer    3
## 58                   Machine Learning Manager    3
## 59                              AI Programmer    2
## 60          Applied Machine Learning Engineer    2
## 61              Autonomous Vehicle Technician    2
## 62                         Big Data Architect    2
## 63                  Data Analytics Consultant    2
```

```
## 64                    Data Analytics Lead    2
## 65              Data Analytics Specialist    2
## 66                              Data Lead    2
## 67                           Data Modeler    2
## 68                    Data Scientist Lead    2
## 69                        Data Strategist    2
## 70                           ETL Engineer    2
## 71                        Insight Analyst    2
## 72                 Marketing Data Analyst    2
## 73                 Principal Data Analyst    2
## 74                Principal Data Engineer    2
## 75                 Software Data Engineer    2
## 76                    Azure Data Engineer    1
## 77                       BI Data Engineer    1
## 78                   Cloud Data Architect    1
## 79                Compliance Data Analyst    1
## 80                   Data DevOps Engineer    1
## 81             Data Management Specialist    1
## 82                 Data Science Tech Lead    1
## 83               Deep Learning Researcher    1
## 84                   Finance Data Analyst    1
## 85               Head of Machine Learning    1
## 86                Manager Data Management    1
## 87                Marketing Data Engineer    1
## 88                     Power BI Developer    1
## 89               Principal Data Architect    1
## 90     Principal Machine Learning Engineer    1
## 91                 Product Data Scientist    1
## 92                     Staff Data Analyst    1
## 93                   Staff Data Scientist    1
```

```r
job_frequency<-data.frame(table(unlist(strsplit(tolower(salary2$job_title), " "))))
##View(job_frequency)

sorted <- job_frequency[order(-job_frequency$Freq),]
View(sorted)
print (sorted)
```

```
##              Var1 Freq
## 16           data 2944
## 22       engineer 1640
## 50      scientist 1065
## 3         analyst  684
## 31       learning  382
## 32        machine  375
## 4       analytics  137
## 47       research  123
## 49        science  116
## 34        manager  113
## 6       architect  105
## 5         applied   82
## 19      developer   42
## 9              bi   39
## 30           lead   38
```

```
## 36             ml   34
## 40             of   31
## 2              ai   29
## 14       computer   27
## 58         vision   27
## 15     consultant   26
## 26           head   20
## 11       business   19
## 27 infrastructure   17
## 51       software   17
## 52     specialist   17
## 41     operations   14
## 43      principal   14
## 10            big   13
## 23            etl   12
## 21       director   11
## 48     researcher   11
## 12          cloud    9
## 18           deep    7
## 39            nlp    7
## 46        quality    7
## 44        product    6
## 17       database    5
## 1              3d    4
## 29   intelligence    4
## 37          mlops    4
## 25      financial    3
## 35      marketing    3
## 7      autonomous    2
## 28        insight    2
## 33     management    2
## 38        modeler    2
## 45     programmer    2
## 53          staff    2
## 54     strategist    2
## 56     technician    2
## 57        vehicle    2
## 8           azure    1
## 13     compliance    1
## 20         devops    1
## 24        finance    1
## 42          power    1
## 55           tech    1
```

Here, I am doing two separate tasks. First, I am tabulating most frequent job titles. Based on the table, we see Data Engineer, Data Scientist, Data Analyst, Machine Learning Engineer, Analytics Engineer, and Data Architect are the 5 most frequent job titles. I also took the job_title column a step further and found the highest frequency of a singular word in a job title. I also find that the words: data, engineer, scientist, analyst, and learning are most frequent word in a job title for this dataset. Based on these results, I believe I will end up coding top 5-7 words as indicator variables to see if having a specific word or "phrase" in the case of Machine Learning impacts one's salary.

```r
salary2[ , 'employment_type1'] <- NA


for(i in 1:length(salary2$employment_type) )
{
  if(salary2$employment_type[i]=="FT")
  {
    salary2$employment_type1[i] <- "FT"
  }

  else
  {
    salary2$employment_type1[i] <- "PT"
  }



  }
##View(salary2)
##table( salary2$employment_type1)

salary3<- subset( salary2, select = -c(employment_type ))
View(salary3)

##table( salary3$employment_type1)

salary3$employment_type1 <- factor(salary3$employment_type1)
levels(salary3$employment_type1)<- c("FT", "PT" )

summary (salary3)
```

```
##   work_year   experience_level  job_title          salary_in_usd    remote_ratio
##   2020:  76   EN: 320          Length:3755        Min.   :  5132   0  :1923
##   2021: 230   Ex: 114          Class :character   1st Qu.: 95000   50 : 189
##   2022:1664   MI: 805          Mode  :character   Median :135000   100:1643
##   2023:1785   SE:2516                             Mean   :137570
##                                                   3rd Qu.:175000
##                                                   Max.   :450000
##   company_size employee_residence1 company_location1 employment_type1
##   L: 454       EU: 459             EU: 454           FT:3718
##   M:3153       OT: 207             OT: 174           PT:  37
##   S: 148       US:3089             US:3127
##
##
##
```

Here is another cleaning step I would like to conduct. Given that only 37 observations are not "full time", I want to collapse all other categories that are not "FT" into one a singular "PT" category thus effectively making this variable a binary categorical variable.

Finally, I am going to create a variety of indicator variables that will represent whether a person's job title has a specific word or not. From the previous section where I broke down frequency of job titles and words in job titles, I found Data Engineer, Data Scientist, Data Analyst, Machine Learning Engineer, and Analytics Engineer were 5 most frequent job titles. This means I want to code data,engineer, scientist, analyst, machine learning, analytics, and manager as indicator variables to include in my model. Note for the 5 most common job titles, I would have to code as interaction terms. For example, data:engineer would only return 1 if both data and engineer are in title aka Data Engineer.

```
salary3[ , 'data_ind'] <- NA
salary3[ , 'engineer_ind'] <- NA
salary3[ , 'scientist_ind'] <- NA
salary3[ , 'analyst_ind'] <- NA
salary3[ , 'ML_ind'] <- NA
salary3[ , 'analytics_ind'] <- NA
salary3[ , 'manager_ind'] <- NA

##View(salary3)


##DATA INDICATOR VARIABLE

for(i in 1:length(salary3$job_title) )
{

  if(grepl("DATA", salary3$job_title[i] , ignore.case = TRUE ) )
  {
    salary3$data_ind[i] <- 1
  }

  else
  {
    salary3$data_ind[i] <- 0
  }

}
 ## View( salary3)


##Engineer Indicator Variable

  for(i in 1:length(salary3$job_title) )
{

  if(grepl("Engineer", salary3$job_title[i] , ignore.case = TRUE ) )
  {
    salary3$engineer_ind[i] <- 1
  }

  else
  {
    salary3$engineer_ind[i] <- 0
  }

}



##Scientist Indicator Variable
```

```r
  for(i in 1:length(salary3$job_title) )
{

  if(grepl("Scientist", salary3$job_title[i] , ignore.case = TRUE ) )
  {
    salary3$scientist_ind[i] <- 1
  }

  else
  {
    salary3$scientist_ind[i] <- 0
  }

}



##Analyst Indicator

  for(i in 1:length(salary3$job_title) )
{

  if(grepl("Analyst", salary3$job_title[i] , ignore.case = TRUE ) )
  {
    salary3$analyst_ind[i] <- 1
  }

  else
  {
    salary3$analyst_ind[i] <- 0
  }

}

##Machine Learning Indicator

  for(i in 1:length(salary3$job_title) )
{

  if(grepl("Machine Learning", salary3$job_title[i] , ignore.case = TRUE )  || grepl("ml", salary3$job_
  {
    salary3$ML_ind[i] <- 1
  }

  else
  {
    salary3$ML_ind[i] <- 0
  }

}

table( salary3$ML_ind)
```

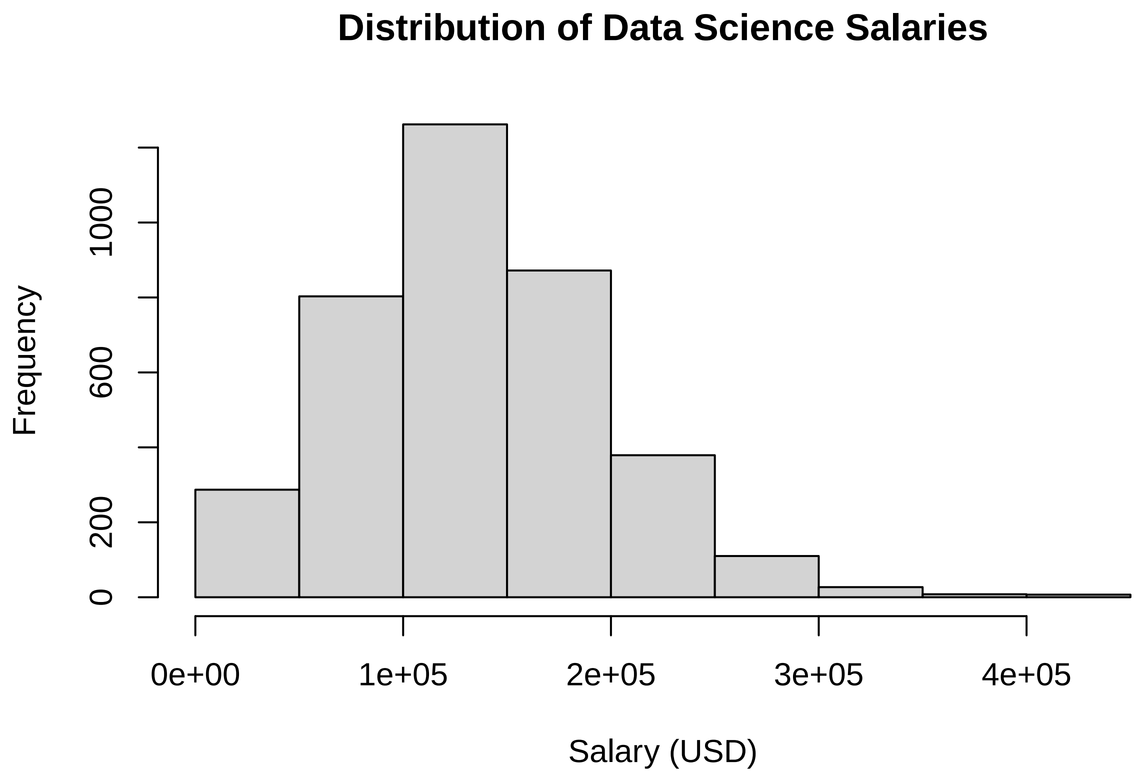
```
##
##    0    1
## 3342  413
```

```
##Analytics Indicator

  for(i in 1:length(salary3$job_title) )
{

  if(grepl("Analytics", salary3$job_title[i] , ignore.case = TRUE ) )
  {
    salary3$analytics_ind[i] <- 1
  }

  else
  {
    salary3$analytics_ind[i] <- 0
  }

}

##Manager Indicator

  for(i in 1:length(salary3$job_title) )
{

  if(grepl("Manager", salary3$job_title[i] , ignore.case = TRUE ) )
  {
    salary3$manager_ind[i] <- 1
  }

  else
  {
    salary3$manager_ind[i] <- 0
  }

}


View(salary3)
```

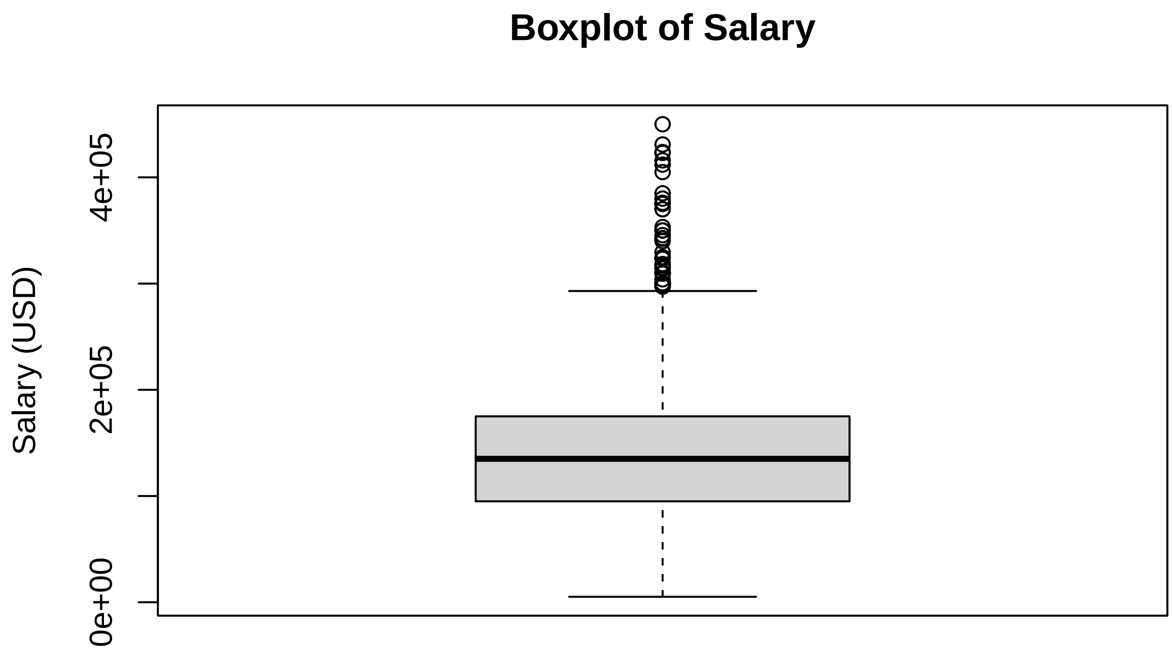
I believe I have cleaned my data as I have desired and will now move onto exploratory data analysis.

Section 1b) Exploratory Data Analysis.
Here, I will dive into the response variable (Salary) and every potential predictor from my cleaned dataset Salary3.I am looking to get a visual representation of salary based on the groups within each predictor variable.

```
library(dplyr)
hist (salary3$salary_in_usd,  xlab = 'Salary (USD)', ylab = 'Frequency', main = 'Distribution of Data S
```




```r
sort (boxplot(salary3$salary_in_usd, ylab= "Salary (USD)", main= "Boxplot of Salary" )$out )
```



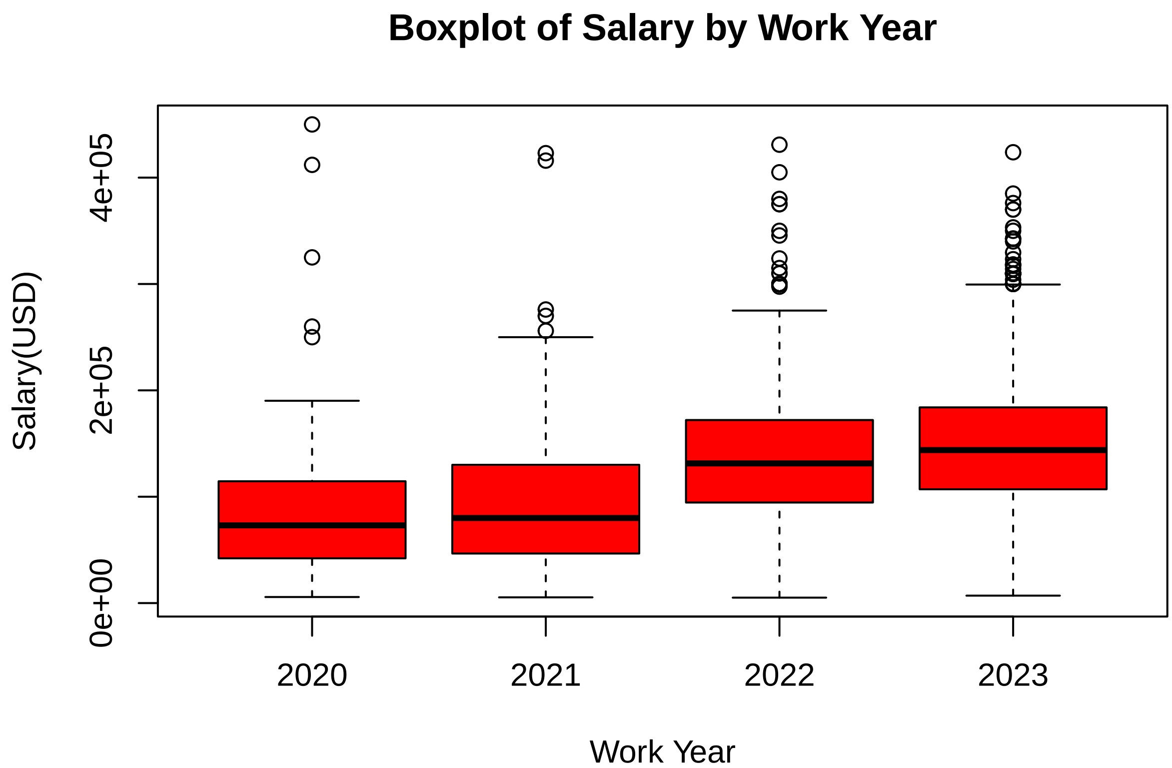

```
##  [1] 297300 297300 297300 297300 297500 299500 299500 299500 299500 299500
## [11] 300000 300000 300000 300000 300000 300000 300000 300000 300000 300000
## [21] 300000 300240 300240 304000 309400 309400 310000 310000 310000 310000
## [31] 310000 310000 310000 314100 315000 317070 318300 318300 323300 324000
## [41] 325000 329500 340000 342300 342810 345600 350000 350000 353200 370000
## [51] 370000 375000 375000 376080 380000 385000 405000 412000 416000 423000
## [61] 423834 430967 450000
```

```r
summary( salary3$salary_in_usd)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5132   95000  135000  137570  175000  450000
```

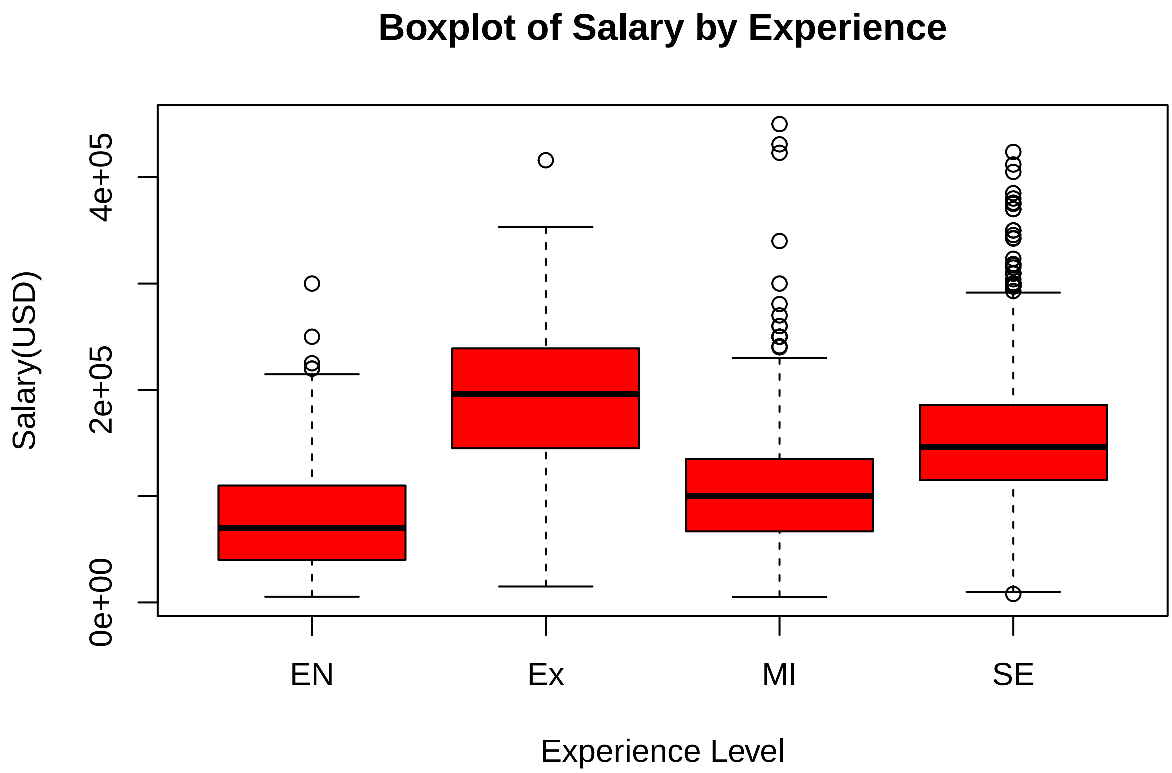
Here we have histogram and boxplot of overall Salary. Our 5 point summary statistic shows mean and median of 135,000 USD and 137,570 USD respectively.The histogram shows overall right skew suggesting that there are few instances in which salaries are very high, but the majority of observations are below 200,000. I performed a sort of boxplot outliers and found those ranged from 297,300-450,000 USD. Additionally, the boxplot confirms a right skew. The boxplot suggests that there are up to 64 outliers based on salary alone. For now, I suspect that these large salaries reflect positions of senior executive positions. In my initial model selection, I will include all outliers. In section 4, I will delve deeper into identifying outliers and seeing if keeping or removing outliers produces a better model.

```r
##boxplots of individual
boxplot(salary_in_usd ~ work_year, data = salary3, col = "red", xlab = 'Work Year', ylab = 'Salary(USD)
```



This boxplot of work_year confirms many of my intuitions about salaries with each passing year. Here, we see the first quartile, median value, and third quartile salaries increasing for each year. The maximum value of "non-outlier" observations for each year are also increasing. In general, there does seem to be some kind of inflation effect happening. Each year continues to have outliers in roughly the same range. In section 4, I will try to quantify to see if these outliers are mostly attributable to those in executive engineer positions.

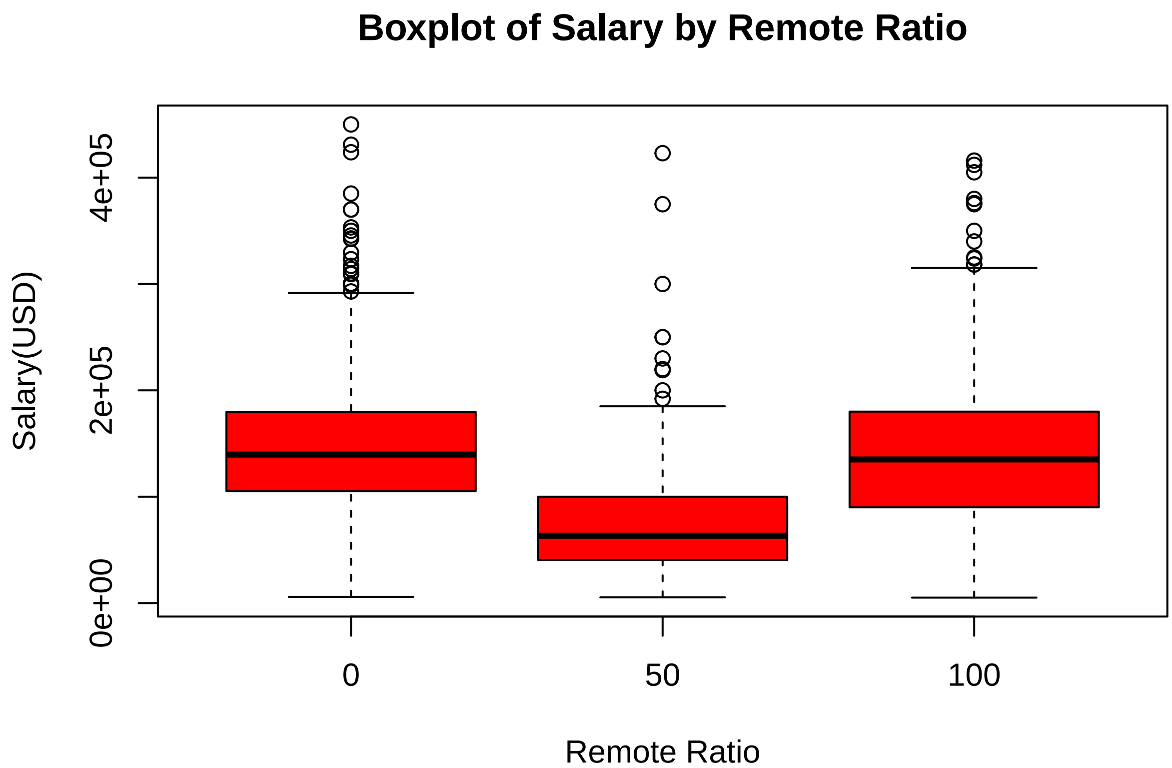
```
##boxplots of individual predictors
boxplot(salary_in_usd ~experience_level, data = salary3, col = "red", xlab = 'Experience Level', ylab =
```



Here, we have a boxplot of salaries grouped by experience level. We see that the interquartile ranges for each experience level does follow as one expects. Entry level has the lowest IQ range, followed by Mid Level, Senior Engineer, and then executive. One would expect that more experience correlates to a higher salary and this trend seems true on a surface level. However, we also see that there are instances of Mid level and Senior Engineer positions that compensation in the same range as very high executive. This suggests that research field and job title may actually contribute to salary more than I expected as there may be certain fields/ job titles with very lucrative compensation even if it is not a executive level position.

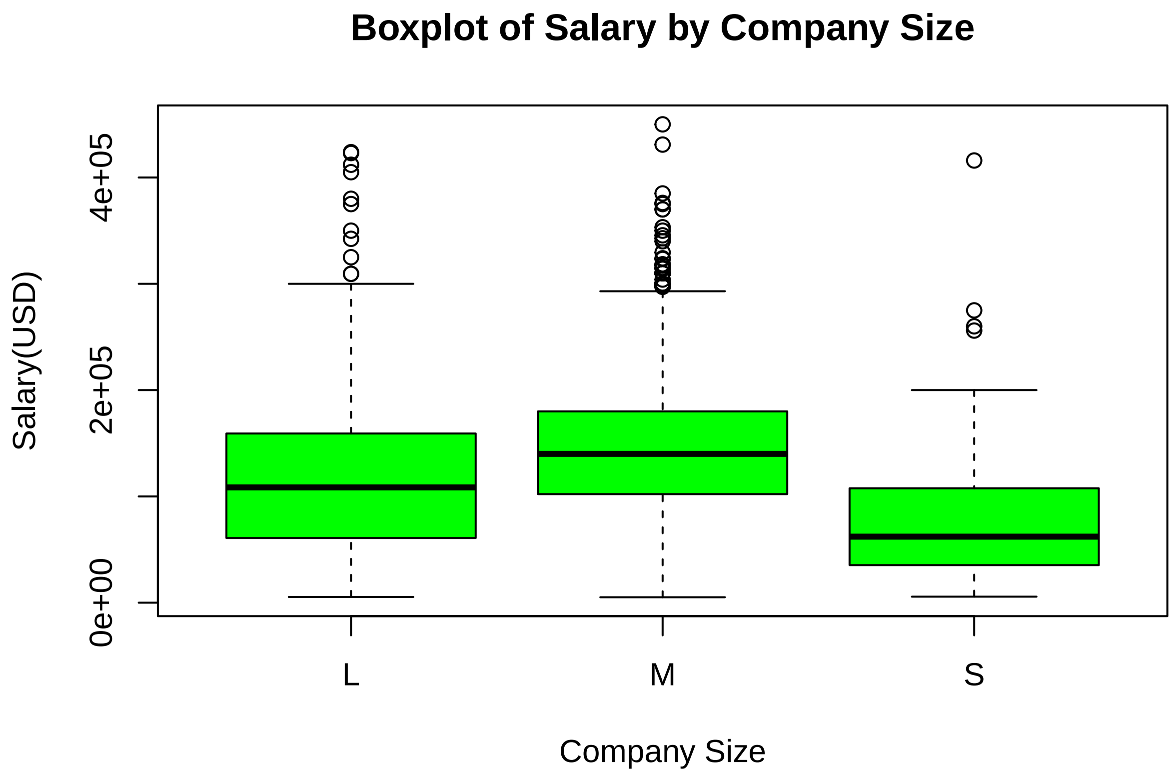
```
##exploratory data analysis for Job Title
```

```
##boxplot of remote_ratio
boxplot(salary_in_usd ~remote_ratio, data = salary3, col = "red", xlab = 'Remote Ratio', ylab = 'Salary
```



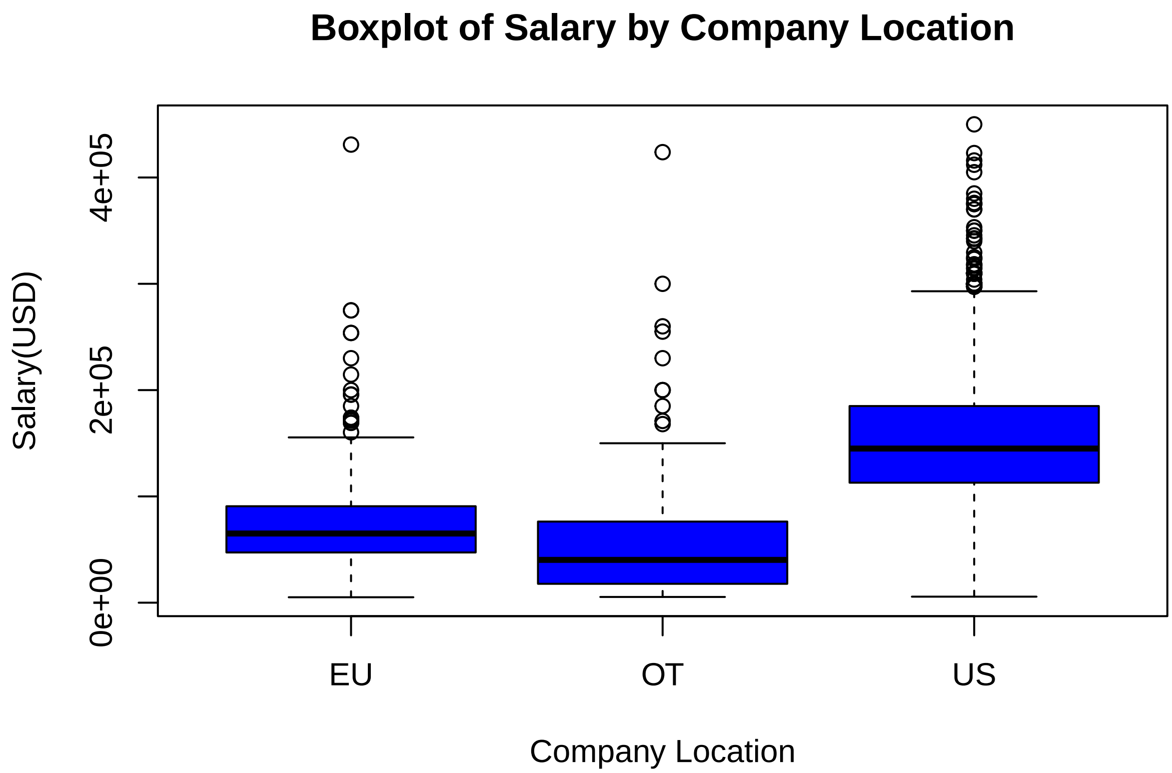
Recall 0= no remote work whereas 100= Fully remote. On a surface level, it seems those who work remotely and in office have a fairly similar interquartile distribution and overall similar number of outliers on the higher end of salary ranges. However, it does seems that the no remote group has a narrower interquartile region. Those who work at a hybrid remote Company seem to have lower salaries based on median and interquartile values.

```
##boxplot of Company size
boxplot(salary_in_usd ~company_size, data = salary3, col = "green", xlab = 'Company Size', ylab = 'Sala
```



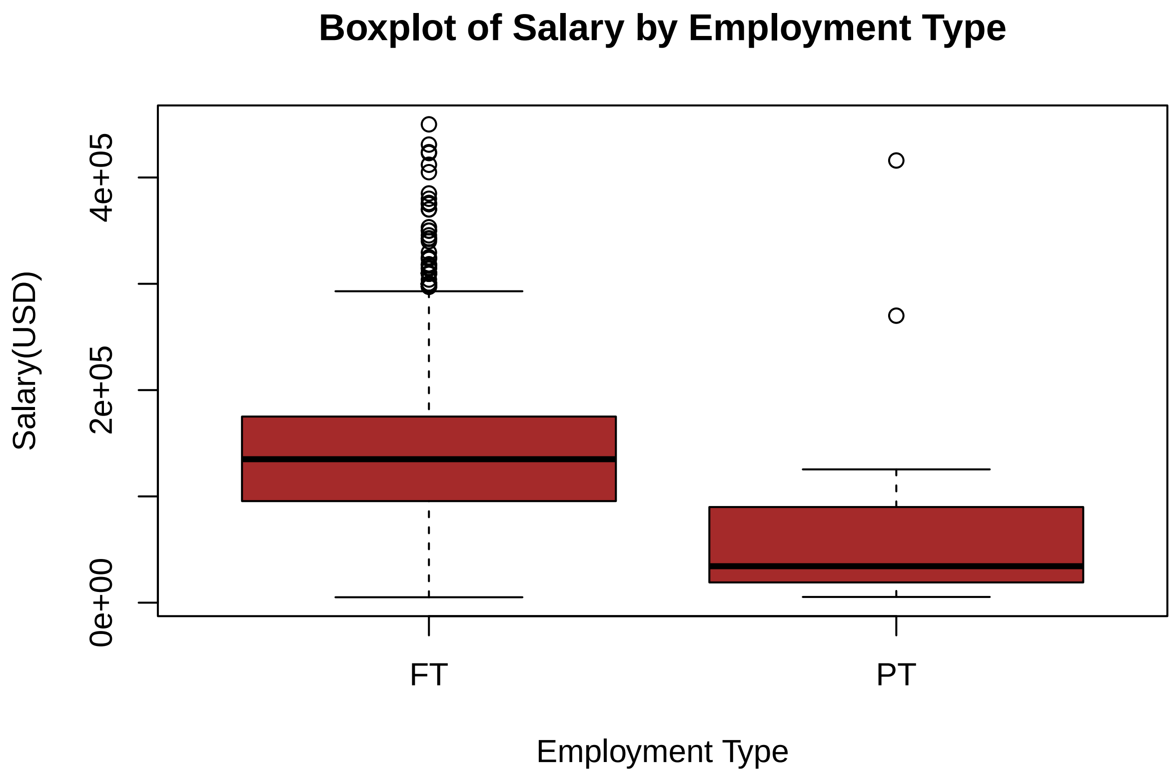
Here, on a surface level, seems that the distributions Company size seem to differ from each other. Those work at small companies have the lowest median+ interquartile range salaries. Additionally, even the outliers for this group have smaller salaries than the outliers of other groups. Next, the "Large" company group has the second highest median and interquartile range salaries. Additionally, the outliers in this group have higher salaries than those in the small group.Finally, it does seem the Medium Company group has the largest median and interquartile salaries among the 3 groups. The maximum "non-outlier" observation for the Large and Medium group are similar as well. Recall that we did determine that most observations in this data set (3153/3755) fall in the "Medium" category.

```
##boxplot of Company size
boxplot(salary_in_usd ~company_location1, data = salary3, col = "blue", xlab = 'Company Location', ylab
```



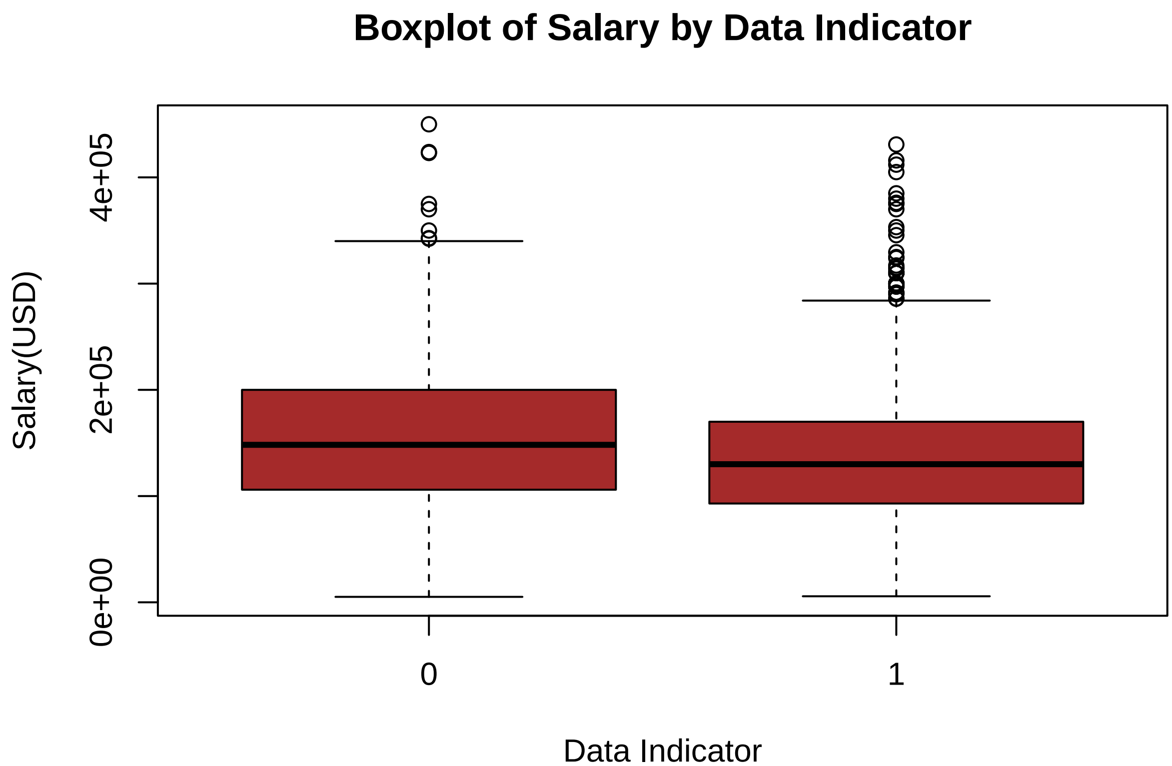
Here, a boxplot based on Company Location shows us that salries for the US+Canada are much higher than in Europe and the rest of the world. The median and interquartile salaries for this group is higher than the 2 other groups. Additionally, the maximum "non-outlier" salary in this group is higher as well. It does also seem that salaries for Companys in Europe are still higher than the rest of the world (excluding US+Canada). This boxplot suggests there may be a meaningful relationship between salary and Company location. I also notice that most high "Outlier" salaries for EU and OT fall well within the non-outlier maximum range for the US group.

```
##boxplot of Company size
boxplot(salary_in_usd ~employment_type1, data = salary3, col = "brown", xlab = 'Employment Type', ylab =
```



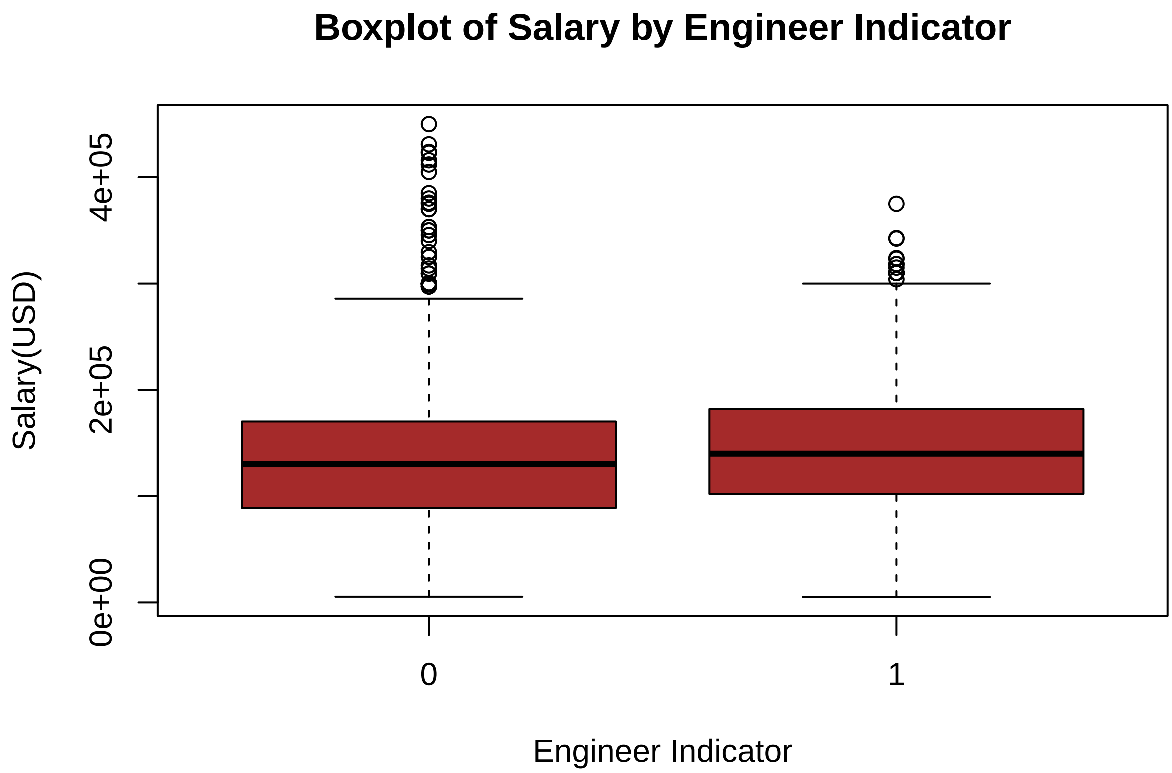
This boxplot doesn't necessarily offer us much insight. The part time group only has 37 observations and intuitively, those who work part time are paid less than full time employees because they are working less. The distribution of salaries for these groups confirms this fact.

```
##boxplot of data indicator
boxplot(salary_in_usd ~data_ind, data = salary3, col = "brown", xlab = 'Data Indicator', ylab = 'Salary
```



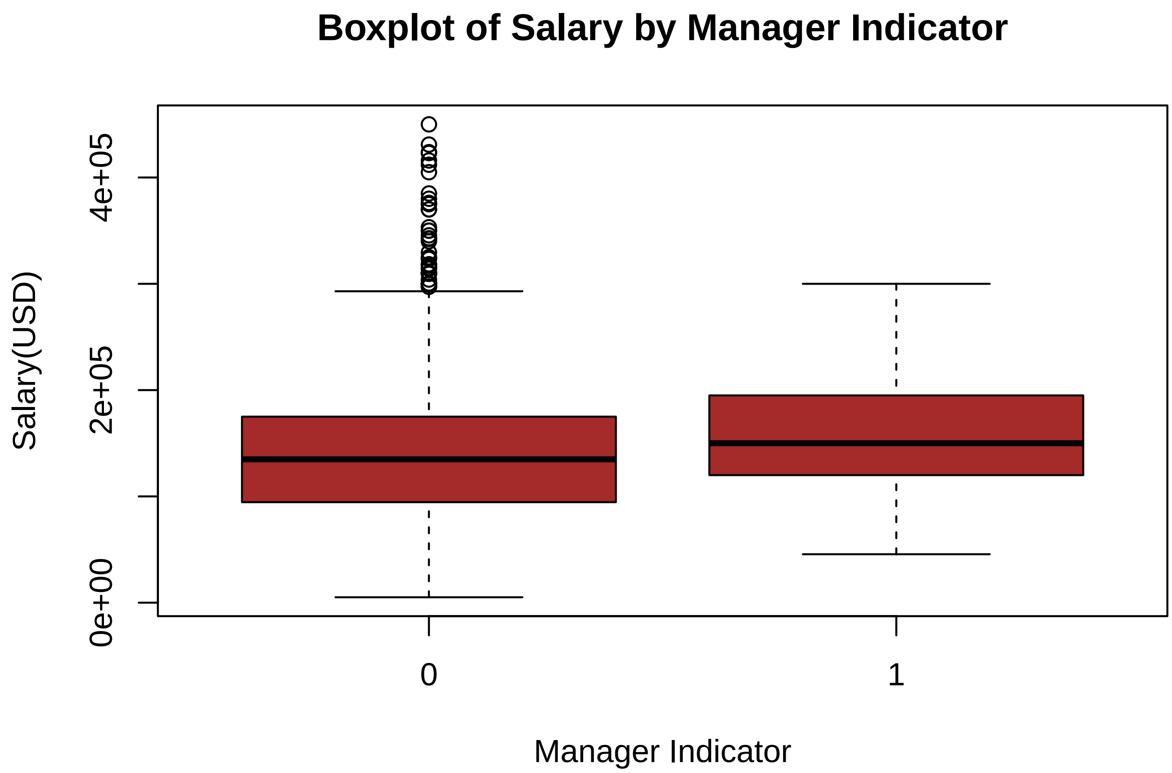
Here, the boxplot of salaries by Data Indicator variables suggests that the interquartile range and median value of those without data in their job title have higher salaries than those with data in their salaries. However, the group with data in job title has much larger sample size and contains way more of the higher "outlier" salaries.

```
##boxplot of Engineer indicator
boxplot(salary_in_usd ~engineer_ind, data = salary3, col = "brown", xlab = 'Engineer Indicator', ylab =
```



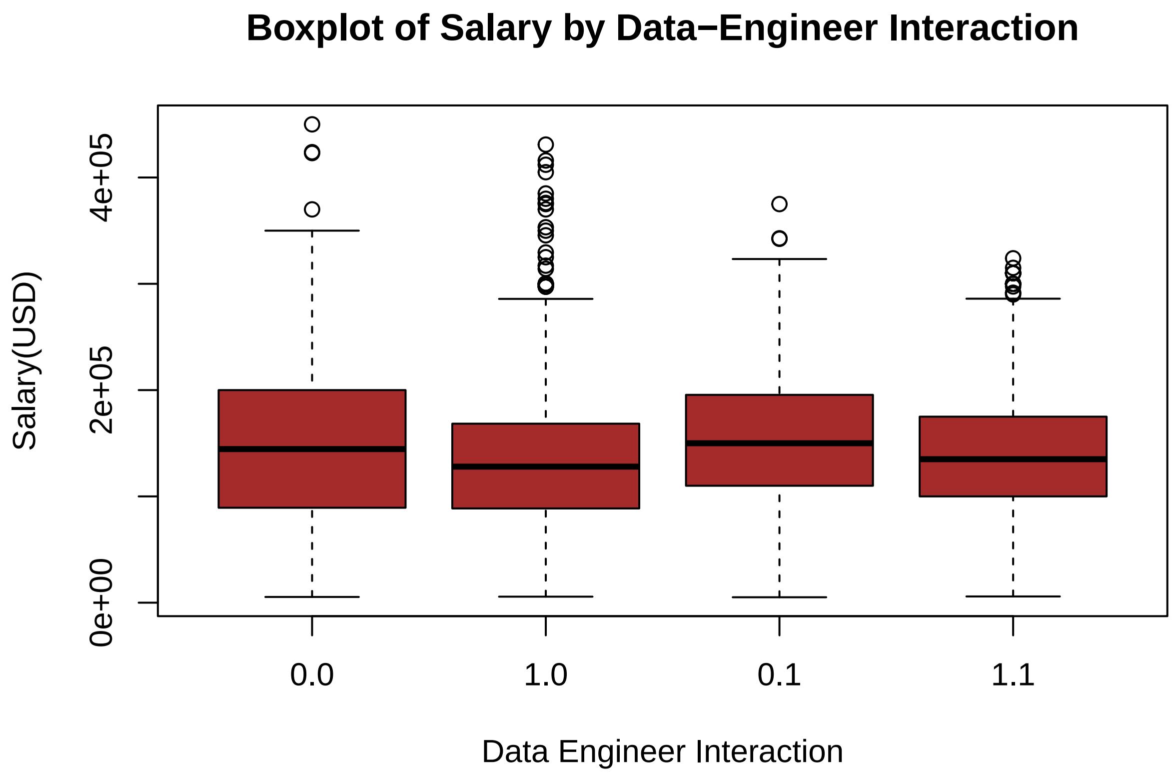
Here, the boxplot of the engineer indicator variables shows that the interquartile ranges and median values for engineer vs non-engineer are very similar. However, the non-engineers seem to have the outlier large salaries. It is possible that engineers are by nature a medium to senior level position, and precludes the group from having manager/executive level positions which may explain why the outlier high salaries are much less frequent in the engineer group.

```
##boxplot of Manager indicator
boxplot(salary_in_usd ~manager_ind, data = salary3, col = "brown", xlab = 'Manager Indicator', ylab = 'S
```



This boxplot is somewhat suprising. I expected those with manager in their title to have more executive roles since they would be directly in charge of others. I thought this would correspond to some of the higher salaries, but evidently the manager group didn't have any observations among the highest earning salaries in the dataset.This might manifest itself when I model the variable.

```
##boxplot of Data+Engineer interaction
boxplot(salary_in_usd ~data_ind:engineer_ind, data = salary3, col = "brown", xlab = 'Data Engineer Inte
```



This is a boxplot of the interaction between Data indicator and Engineer indicator variable. The 1.1 group is the "Data Engineer" group. I notice the interquartile range is very small for this group as if to suggest that salaries for this group are very clustered. There are very few outliers in this group and even then, their salaries aren't necessarily large compared to the entire dataset.It is possible that data engineer is very much a "mid-level" position and the boxplot distribution suggests that.

I will not present boxplot for every indicator variable/ interaction combination. I just wanted to explore the indicators with most frequent occurrences.

Section 2: Model Building+ Selection. Here, I am going to build the best possible model based on the predictors within this data set. I will still include a full model for reference and compare that to the best model I determine from stepwise selection.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2

## Loading required package: carData
```

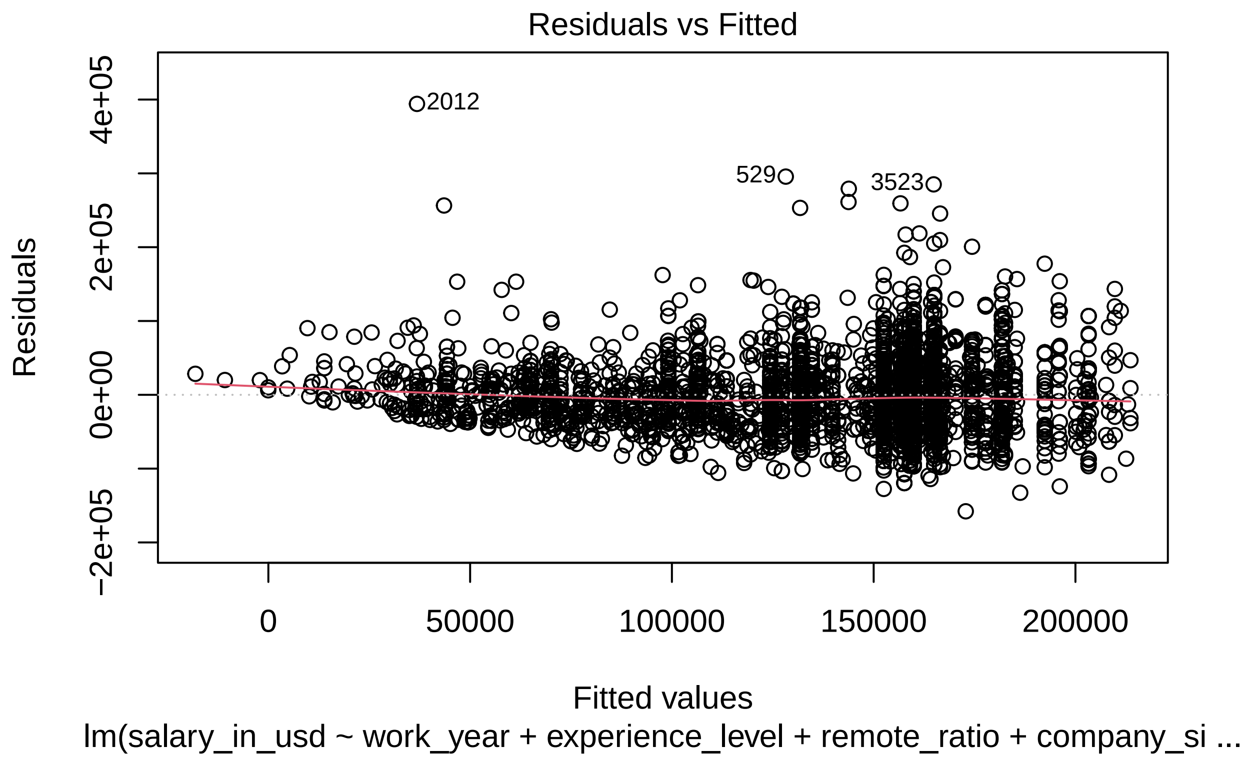
```
## Warning: package 'carData' was built under R version 4.3.2

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

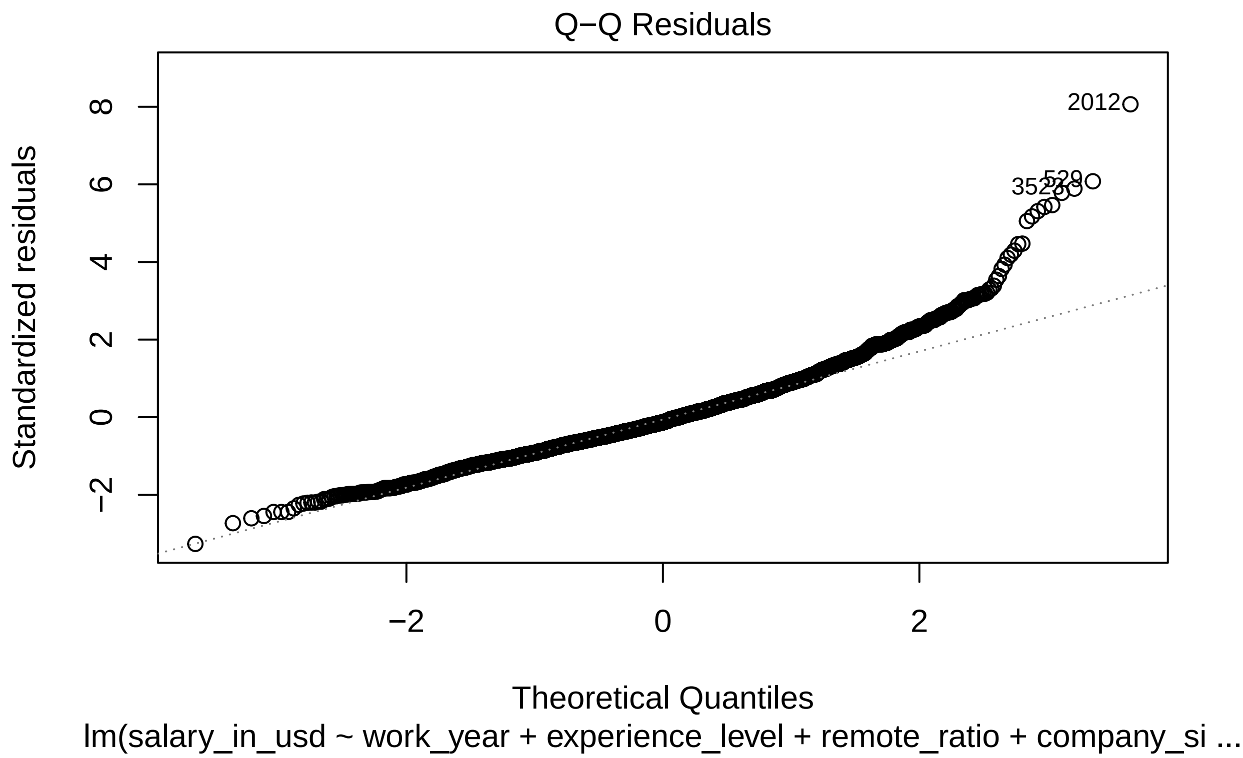
```
model_full<- lm (salary_in_usd~work_year+experience_level+remote_ratio+company_size+employee_residence1+
                   scientist_ind+ analyst_ind+ML_ind+analytics_ind+ manager_ind, data= salary3)

vif(model_full)
```

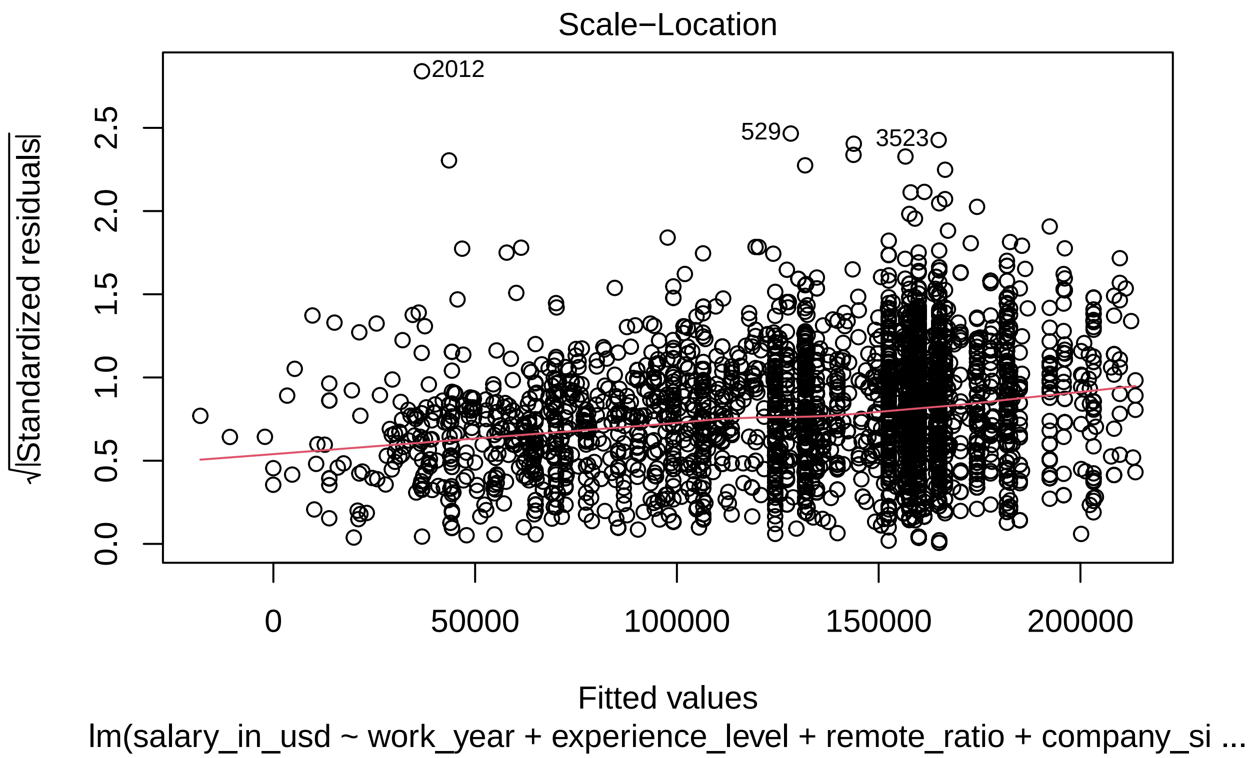
```
##                          GVIF Df GVIF^(1/(2*Df))
## work_year            1.598122  3        1.081272
## experience_level     1.346193  3        1.050795
## remote_ratio         1.447338  2        1.096838
## company_size         1.659587  2        1.135011
## employee_residence1 71.290412  2        2.905747
## company_location1   67.638199  2        2.867794
## employment_type1     1.101993  1        1.049759
## data_ind             2.389581  1        1.545827
## engineer_ind         4.434853  1        2.105909
## scientist_ind        3.812811  1        1.952642
## analyst_ind          3.185869  1        1.784900
## ML_ind               2.259729  1        1.503240
## analytics_ind        1.323541  1        1.150452
## manager_ind          1.478046  1        1.215749
```

Here, one of my first biggest suspicion was that employment_type1 and Company_location1 were too similar and would cause issue of collinearity. A vif test shows that both are above 10 so i need to remove one of them from my model. I will remove employee location as Company location is slightly more important since the Company is paying their workers.

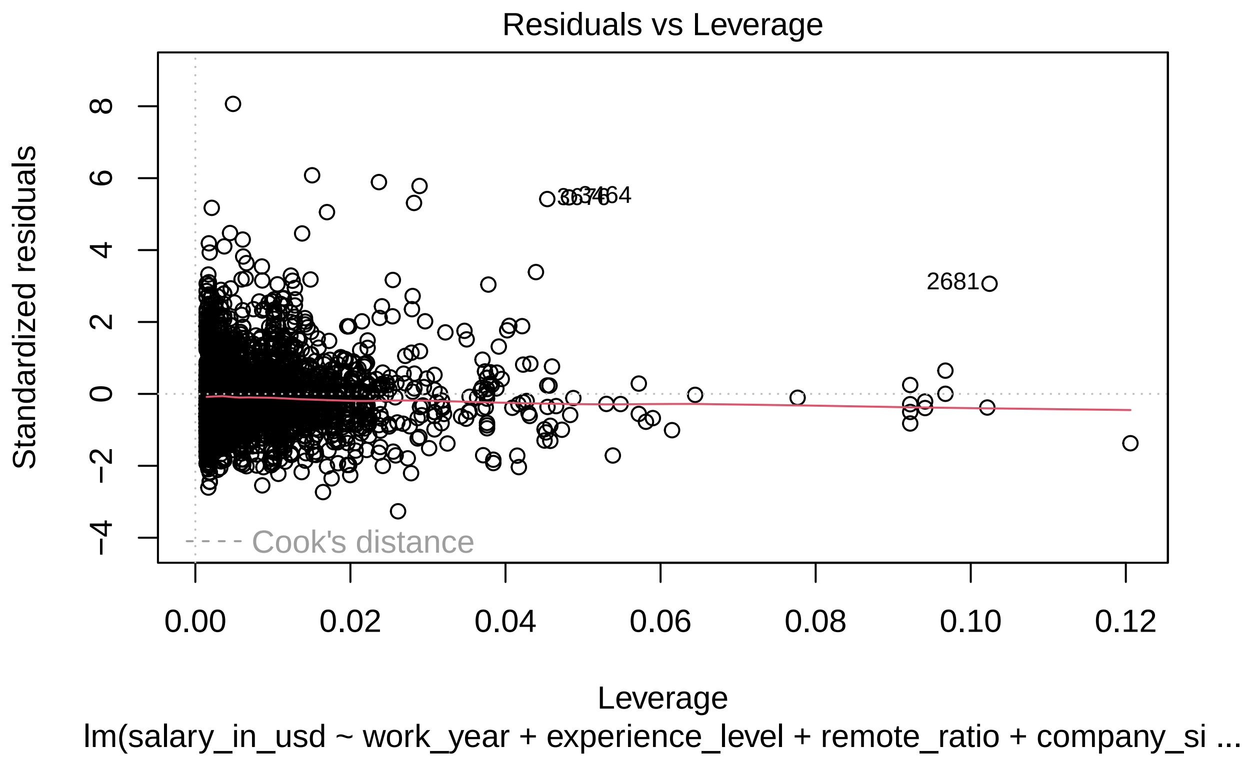
```
model_full2<- lm (salary_in_usd~work_year+experience_level+remote_ratio+company_size+company_location1+
                    analytics_ind:engineer_ind, data= salary3)






plot(model_full2)
```


Residuals vs Fitted
Residuals
4e+05
2e+05
0e+00
−2e+05
2012
529
3523
0
50000
100000
150000
200000
Fitted values
lm(salary_in_usd ~ work_year + experience_level + remote_ratio + company_si ...


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(salary_in_usd ~ work_year + experience_level + remote_ratio + company_si ...
2012
529
3523
−2
0
2
4
6
8



Scale−Location
√|Standardized residuals|
Fitted values
lm(salary_in_usd ~ work_year + experience_level + remote_ratio + company_si ...
2012
529
3523
0.0
0.5
1.0
1.5
2.0
2.5
0
50000
100000
150000
200000




```r
summary(model_full2)
```

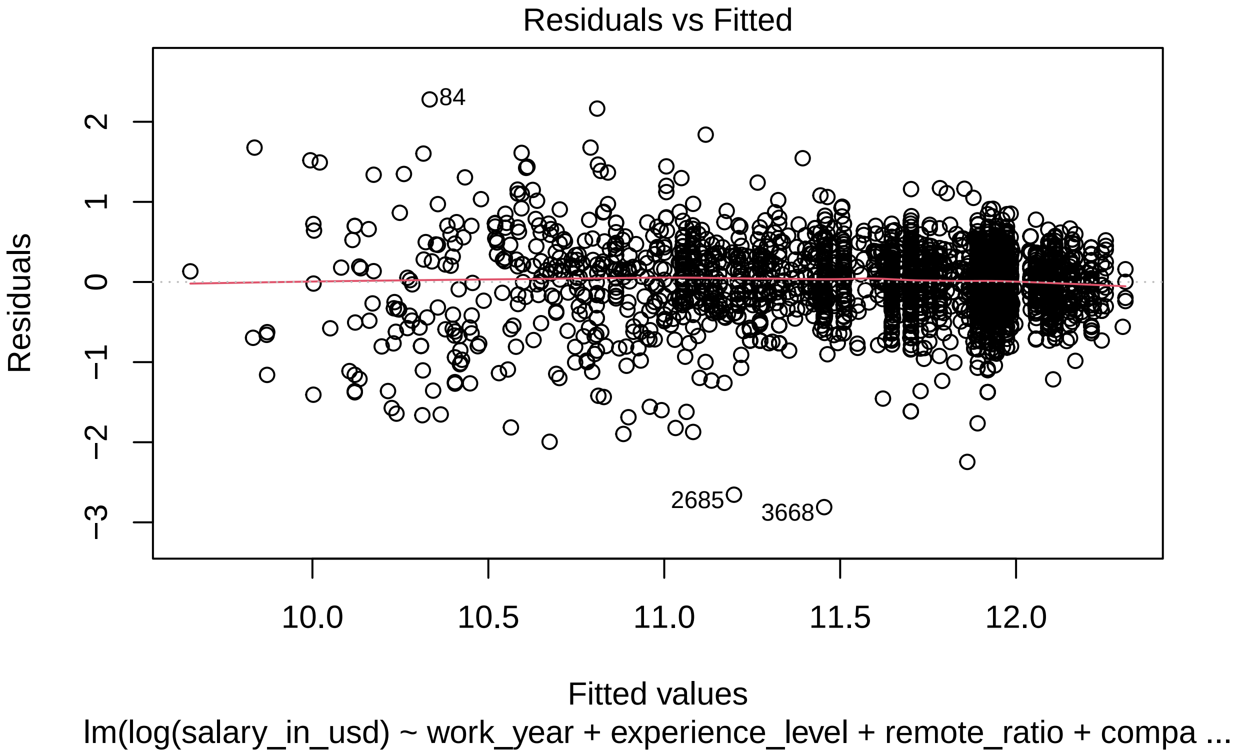
```
##
## Call:
## lm(formula = salary_in_usd ~ work_year + experience_level + remote_ratio +
##     company_size + company_location1 + employment_type1 + data_ind +
##     engineer_ind + scientist_ind + analyst_ind + ML_ind + analytics_ind +
##     manager_ind + data_ind:engineer_ind + data_ind:scientist_ind +
##     data_ind:analyst_ind + ML_ind:engineer_ind + analytics_ind:engineer_ind,
##     data = salary3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -157819  -31788   -6482   26034  394142
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 59076.46    9569.94   6.173 7.41e-10 ***
## work_year2021              -10754.64    6525.65  -1.648 0.099425 .
## work_year2022               -5134.37    6085.18  -0.844 0.398863
## work_year2023                2293.57    6173.72   0.372 0.710282
## experience_levelEx          89671.07    5538.32  16.191  < 2e-16 ***
## experience_levelMI          21105.15    3345.67   6.308 3.15e-10 ***
## experience_levelSE          46332.19    3165.81  14.635  < 2e-16 ***
## remote_ratio50              -4806.54    4368.30  -1.100 0.271263
```

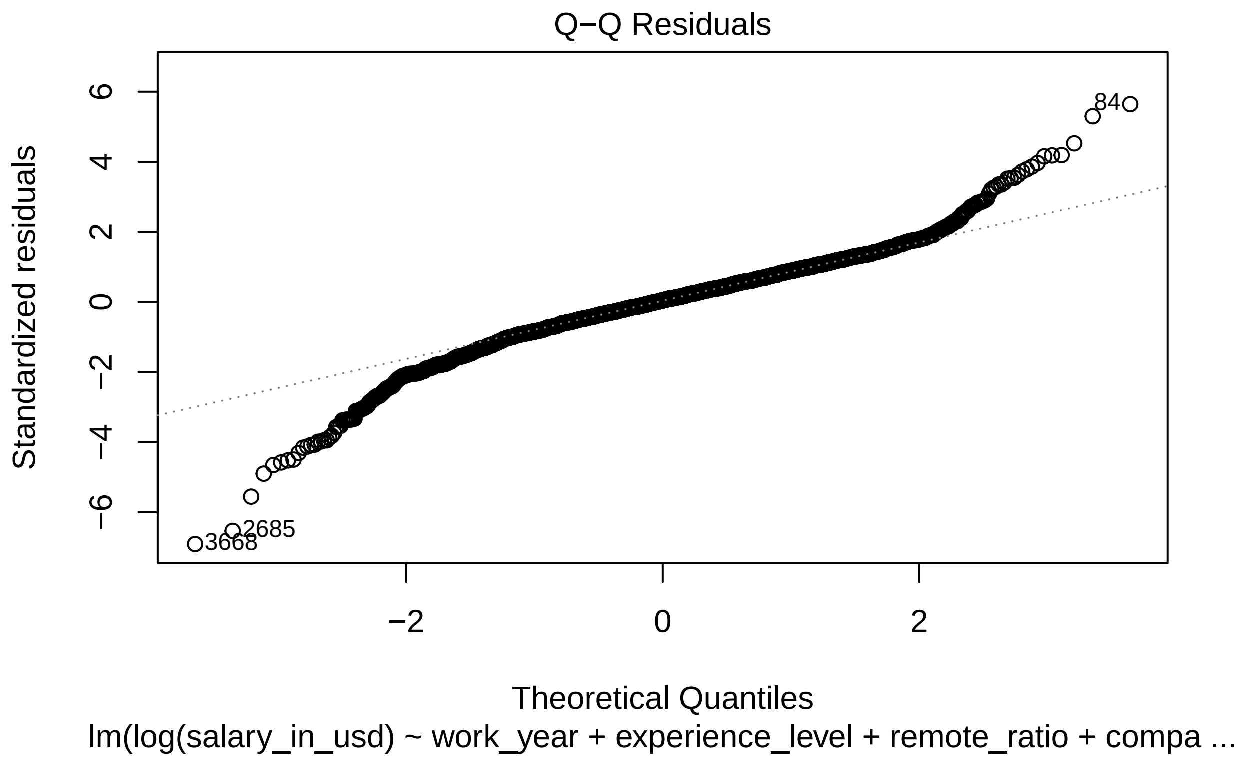
```
## remote_ratio100               21.15    1719.01   0.012 0.990186
## company_sizeM              -3733.81    2900.89  -1.287 0.198131
## company_sizeS             -19017.54    4782.49  -3.976 7.13e-05 ***
## company_location1OT        -5606.64    4485.46  -1.250 0.211393
## company_location1US        62293.98    2713.35  22.958  < 2e-16 ***
## employment_type1PT        -23389.43    8376.43  -2.792 0.005260 **
## data_ind                     164.28    7524.13   0.022 0.982582
## engineer_ind               16299.23    8551.87   1.906 0.056737 .
## scientist_ind              26126.24    7302.86   3.578 0.000351 ***
## analyst_ind               -17853.16   16214.88  -1.101 0.270952
## ML_ind                     -9206.55    7574.70  -1.215 0.224277
## analytics_ind             -18983.38   10130.50  -1.874 0.061025 .
## manager_ind                11298.11    5983.30   1.888 0.059067 .
## data_ind:engineer_ind     -22815.94    9292.89  -2.455 0.014126 *
## data_ind:scientist_ind    -27550.64    8234.28  -3.346 0.000828 ***
## data_ind:analyst_ind      -16799.69   16676.10  -1.007 0.313802
## engineer_ind:ML_ind         8432.44    9628.15   0.876 0.381189
## engineer_ind:analytics_ind   663.93   12312.44   0.054 0.956999
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48990 on 3729 degrees of freedom
## Multiple R-squared:  0.4004, Adjusted R-squared:  0.3964
## F-statistic: 99.61 on 25 and 3729 DF,  p-value: < 2.2e-16
```

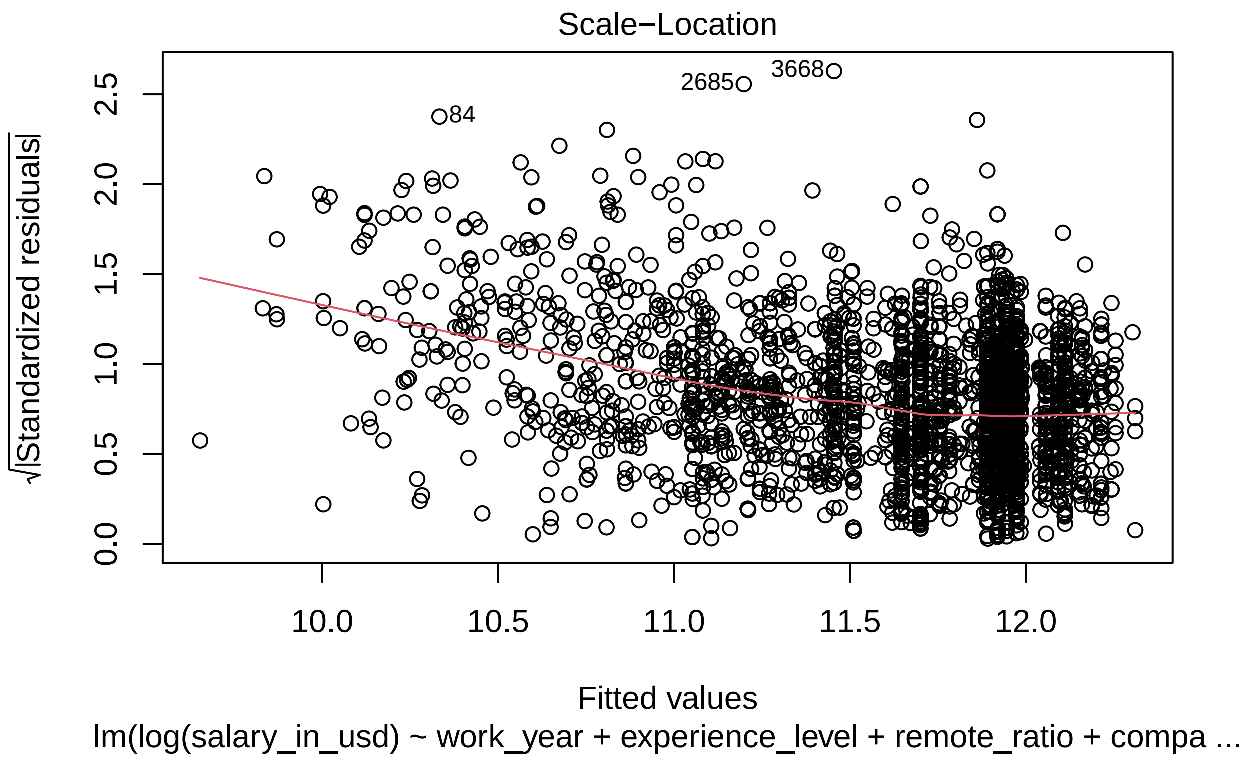
The full model I fit here includes every predictor, every job_title word indicator that I created in section 1, and then the interaction terms of job titles for the 5 most popular job titles in the data set. (i.e, data engineer, data scientist,etc). Here before I move onto model building through AIC stepwise selection, I need to check my regression assumptions and see if the response variable needs any transformation. Looking at the regression diagnostic plot, I notice issues in every diagnostic plot. The residuals vs fitted plot is exhibiting classic fanning where there residuals grow larger as the fitted salaries get larger. This violates our assumption of homoskedacity. For the QQplot I notice there is heavy right tail in the normality plot (which we already suspect from our histogram plot). This does makes sense as in any industry, those with the highest paying salaries have disproportionately higher salaries than others. Finally, the scale vs location plot is nowhere near horizontal slope. It starts at 0.5 studentized residual distance and ends at 0.8. Finally, if I look at the full modeled I constructed, I have every single indicator variable and then 5 interaction variables between the indicators that represent the 5 most frequently occurring job (i.e Data Engineer or Data Analyst). Here, I notice that all 3 factors of experience level seem to be significant, small company size, company being in the US, employment type, scientist_ind, and the interaction terms data engineer and data scientist seem to be significant predictors in the full model. R^2 is fairly low at 0.4004 and adjusted R^2 is also low due to the penalty of having so many predictors. In class, for very heavy right skew models we utilized a log transformation on our response variable so as to “scale” and reduce the weight that very high salary outliers would have on the model.

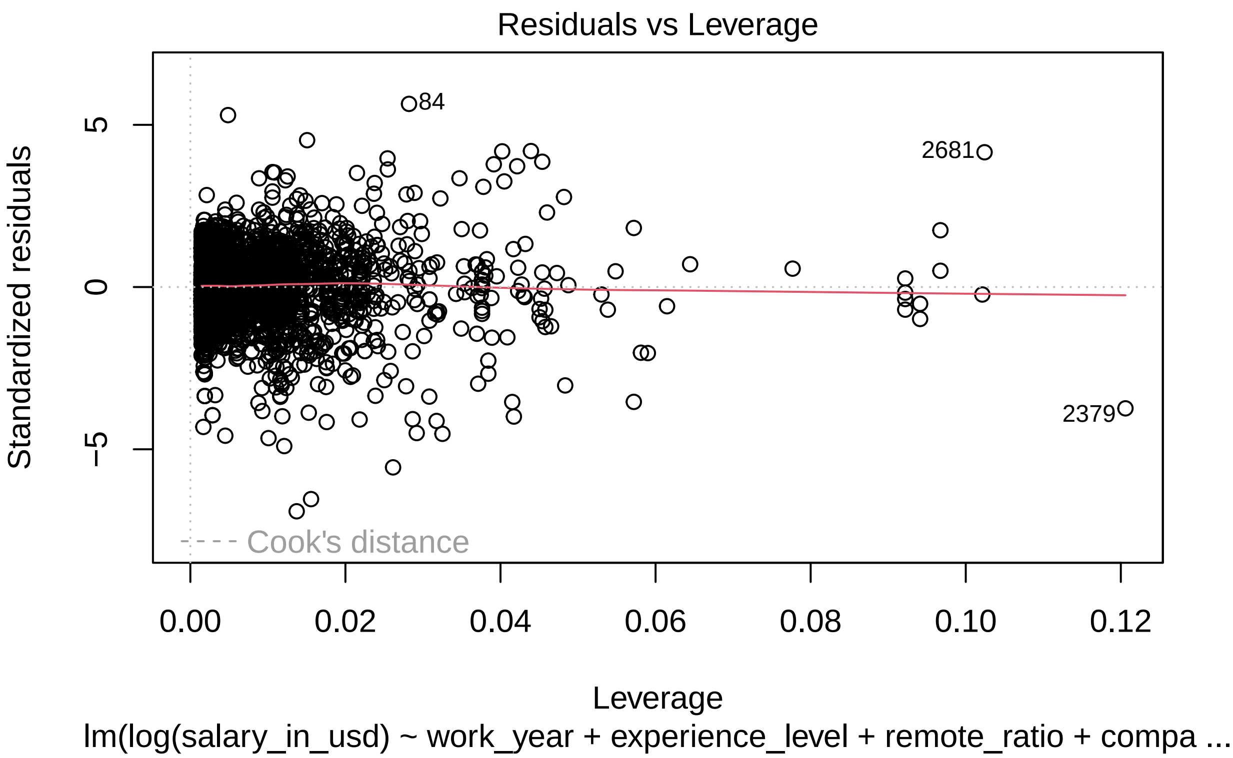
```r
log_fullmodel<- lm ( log(salary_in_usd)~work_year+experience_level+remote_ratio+company_size+company_lo
                   scientist_ind+ analyst_ind+ML_ind+analytics_ind+ manager_ind+ data_ind:engineer_ind+
                    analytics_ind:engineer_ind, data= salary3)
```

```r
plot(log_fullmodel)
```


Residuals vs Fitted
84
2685
3668
Residuals
Fitted values
lm(log(salary_in_usd) ~ work_year + experience_level + remote_ratio + compa ...
−3
−2
−1
0
1
2
10.0
10.5
11.0
11.5
12.0


Q−Q Residuals
Standardized residuals
6
4
2
0
−2
−4
−6
84
2685
3668
−2
0
2
Theoretical Quantiles
lm(log(salary_in_usd) ~ work_year + experience_level + remote_ratio + compa ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
10.0
10.5
11.0
11.5
12.0
84
2685
3668
Fitted values
lm(log(salary_in_usd) ~ work_year + experience_level + remote_ratio + compa ...




```r
summary(log_fullmodel)
```

```
##
## Call:
## lm(formula = log(salary_in_usd) ~ work_year + experience_level +
##     remote_ratio + company_size + company_location1 + employment_type1 +
##     data_ind + engineer_ind + scientist_ind + analyst_ind + ML_ind +
##     analytics_ind + manager_ind + data_ind:engineer_ind + data_ind:scientist_ind +
##     data_ind:analyst_ind + ML_ind:engineer_ind + analytics_ind:engineer_ind,
##     data = salary3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.81008 -0.21431  0.01673  0.24306  2.27833
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               10.703298   0.079980 133.825  < 2e-16 ***
## work_year2021             -0.038645   0.054537  -0.709 0.478619
## work_year2022              0.055630   0.050856   1.094 0.274087
## work_year2023              0.109336   0.051596   2.119 0.034150 *
## experience_levelEx         0.759946   0.046286  16.419  < 2e-16 ***
## experience_levelMI         0.299908   0.027961  10.726  < 2e-16 ***
## experience_levelSE         0.490757   0.026458  18.549  < 2e-16 ***
## remote_ratio50            -0.022688   0.036508  -0.621 0.534341
```

```
## remote_ratio100           -0.001167   0.014366  -0.081 0.935243
## company_sizeM             -0.015060   0.024244  -0.621 0.534525
## company_sizeS             -0.194781   0.039969  -4.873 1.14e-06 ***
## company_location1OT       -0.403035   0.037487 -10.751  < 2e-16 ***
## company_location1US        0.647586   0.022676  28.558  < 2e-16 ***
## employment_type1PT        -0.521091   0.070005  -7.444 1.21e-13 ***
## data_ind                   0.049920   0.062882   0.794 0.427324
## engineer_ind               0.155590   0.071471   2.177 0.029546 *
## scientist_ind              0.217512   0.061033   3.564 0.000370 ***
## analyst_ind               -0.056116   0.135514  -0.414 0.678824
## ML_ind                    -0.023274   0.063305  -0.368 0.713155
## analytics_ind             -0.142989   0.084664  -1.689 0.091324 .
## manager_ind                0.109600   0.050005   2.192 0.028455 *
## data_ind:engineer_ind     -0.195987   0.077664  -2.524 0.011660 *
## data_ind:scientist_ind    -0.229024   0.068817  -3.328 0.000883 ***
## data_ind:analyst_ind      -0.228180   0.139368  -1.637 0.101664
## engineer_ind:ML_ind        0.043233   0.080466   0.537 0.591101
## engineer_ind:analytics_ind  0.024230   0.102900   0.235 0.813859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4094 on 3729 degrees of freedom
## Multiple R-squared:  0.5293, Adjusted R-squared:  0.5261
## F-statistic: 167.7 on 25 and 3729 DF,  p-value: < 2.2e-16
```

```
salary3[ , 'log_salary'] <- NA
##View(salary3)

salary4 <- transform(salary3, log_salary = log(salary_in_usd))
View(salary4)
```

Here,I create another column in my data frame so as to obtain log values for salary_in_USD and then regress the full model of the transformed response variable. I look at the regression plot once more and I see improvement in the residuals vs fitted graph. The issues of fanning out is resolved and I would say the assumption of homoskedacity is met here. The QQ plot seems to have made things only slightly better as now there are skews on both tails, but is at least symmetric. Scale vs location plot doesn't seem to have improved at all. However, a vast majority of observations are within the 11.5-12.0 fitted values range, the red line in that range doesn't really jump studentized residual range (0.7-0.6) and most of the issues of this diagnostic plot lies within the range with much fewer observations. One can consider this graph marginally better. These plots may improve once I've found a simpler and final model. Finally, the adjusted R^2 of the transformed model is meaningfully better at 0.52 vs 0.40. So I believe the log transformation of my response variable is correct.

```
backward<- lm ( log_salary~work_year+experience_level+remote_ratio+company_size+company_location1+emplo
                   scientist_ind+ analyst_ind+ML_ind+analytics_ind+ manager_ind+ data_ind:engineer_ind+
                    analytics_ind:engineer_ind, data= salary4)


buildBackward<-stepAIC(backward,direction="backward")
```

```
## Start:  AIC=-6680.6
## log_salary ~ work_year + experience_level + remote_ratio + company_size +
##     company_location1 + employment_type1 + data_ind + engineer_ind +
```

```
##     scientist_ind + analyst_ind + ML_ind + analytics_ind + manager_ind +
##     data_ind:engineer_ind + data_ind:scientist_ind + data_ind:analyst_ind +
##     ML_ind:engineer_ind + analytics_ind:engineer_ind
##
##                             Df Sum of Sq    RSS     AIC
## - remote_ratio               2     0.066 625.14 -6684.2
## - engineer_ind:analytics_ind  1     0.009 625.09 -6682.5
## - engineer_ind:ML_ind        1     0.048 625.12 -6682.3
## <none>                                  625.08 -6680.6
## - data_ind:analyst_ind       1     0.449 625.53 -6679.9
## - manager_ind                1     0.805 625.88 -6677.8
## - data_ind:engineer_ind      1     1.067 626.14 -6676.2
## - data_ind:scientist_ind     1     1.857 626.93 -6671.5
## - work_year                  3     4.406 629.48 -6660.2
## - company_size               2     4.190 629.27 -6659.5
## - employment_type1           1     9.288 634.36 -6627.2
## - experience_level           3    76.702 701.78 -6252.0
## - company_location1          2   234.728 859.80 -5487.4
##
## Step:  AIC=-6684.2
## log_salary ~ work_year + experience_level + company_size + company_location1 +
##     employment_type1 + data_ind + engineer_ind + scientist_ind +
##     analyst_ind + ML_ind + analytics_ind + manager_ind + data_ind:engineer_ind +
##     data_ind:scientist_ind + data_ind:analyst_ind + engineer_ind:ML_ind +
##     engineer_ind:analytics_ind
##
##                             Df Sum of Sq    RSS     AIC
## - engineer_ind:analytics_ind  1     0.009 625.15 -6686.2
## - engineer_ind:ML_ind        1     0.050 625.19 -6685.9
## <none>                                  625.14 -6684.2
## - data_ind:analyst_ind       1     0.454 625.60 -6683.5
## - manager_ind                1     0.816 625.96 -6681.3
## - data_ind:engineer_ind      1     1.081 626.22 -6679.7
## - data_ind:scientist_ind     1     1.876 627.02 -6675.0
## - company_size               2     4.165 629.31 -6663.3
## - work_year                  3     4.828 629.97 -6661.3
## - employment_type1           1     9.464 634.61 -6629.8
## - experience_level           3    77.488 702.63 -6251.4
## - company_location1          2   246.287 871.43 -5441.0
##
## Step:  AIC=-6686.15
## log_salary ~ work_year + experience_level + company_size + company_location1 +
##     employment_type1 + data_ind + engineer_ind + scientist_ind +
##     analyst_ind + ML_ind + analytics_ind + manager_ind + data_ind:engineer_ind +
##     data_ind:scientist_ind + data_ind:analyst_ind + engineer_ind:ML_ind
##
##                         Df Sum of Sq    RSS     AIC
## - engineer_ind:ML_ind     1     0.043 625.19 -6687.9
## <none>                              625.15 -6686.2
## - data_ind:analyst_ind    1     0.451 625.60 -6685.4
## - manager_ind             1     0.822 625.97 -6683.2
## - data_ind:engineer_ind   1     1.167 626.32 -6681.2
## - analytics_ind           1     1.182 626.33 -6681.1
## - data_ind:scientist_ind  1     1.869 627.02 -6676.9
```

```
## - company_size            2     4.184 629.33 -6665.1
## - work_year               3     4.820 629.97 -6663.3
## - employment_type1        1     9.508 634.66 -6631.5
## - experience_level        3    77.953 703.10 -6250.9
## - company_location1       2   246.397 871.55 -5442.5
##
## Step:  AIC=-6687.89
## log_salary ~ work_year + experience_level + company_size + company_location1 +
##     employment_type1 + data_ind + engineer_ind + scientist_ind +
##     analyst_ind + ML_ind + analytics_ind + manager_ind + data_ind:engineer_ind +
##     data_ind:scientist_ind + data_ind:analyst_ind
##
##                          Df Sum of Sq    RSS     AIC
## - ML_ind                  1     0.000 625.19 -6689.9
## <none>                                625.19 -6687.9
## - data_ind:analyst_ind    1     0.488 625.68 -6687.0
## - manager_ind             1     0.830 626.02 -6684.9
## - analytics_ind           1     1.512 626.71 -6680.8
## - data_ind:engineer_ind   1     1.790 626.98 -6679.2
## - data_ind:scientist_ind  1     1.921 627.12 -6678.4
## - company_size            2     4.245 629.44 -6666.5
## - work_year               3     4.851 630.05 -6664.9
## - employment_type1        1     9.598 634.79 -6632.7
## - experience_level        3    78.070 703.26 -6252.0
## - company_location1       2   246.390 871.58 -5444.3
##
## Step:  AIC=-6689.89
## log_salary ~ work_year + experience_level + company_size + company_location1 +
##     employment_type1 + data_ind + engineer_ind + scientist_ind +
##     analyst_ind + analytics_ind + manager_ind + data_ind:engineer_ind +
##     data_ind:scientist_ind + data_ind:analyst_ind
##
##                          Df Sum of Sq    RSS     AIC
## <none>                                625.19 -6689.9
## - data_ind:analyst_ind    1     0.490 625.68 -6689.0
## - manager_ind             1     0.840 626.03 -6686.8
## - data_ind:scientist_ind  1     1.923 627.12 -6680.4
## - data_ind:engineer_ind   1     1.960 627.15 -6680.1
## - analytics_ind           1     2.060 627.25 -6679.5
## - company_size            2     4.245 629.44 -6668.5
## - work_year               3     4.862 630.06 -6666.8
## - employment_type1        1     9.598 634.79 -6634.7
## - experience_level        3    78.143 703.34 -6253.7
## - company_location1       2   246.473 871.67 -5445.9
```

```r
summary(buildBackward)
```

```
##
## Call:
## lm(formula = log_salary ~ work_year + experience_level + company_size +
##     company_location1 + employment_type1 + data_ind + engineer_ind +
##     scientist_ind + analyst_ind + analytics_ind + manager_ind +
##     data_ind:engineer_ind + data_ind:scientist_ind + data_ind:analyst_ind,
##     data = salary4)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8044 -0.2147  0.0167  0.2424  2.2710
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              10.68576    0.07578 141.019  < 2e-16 ***
## work_year2021            -0.03998    0.05446  -0.734 0.462902
## work_year2022             0.05802    0.05066   1.145 0.252088
## work_year2023             0.11203    0.05128   2.185 0.028966 *
## experience_levelEx        0.76131    0.04612  16.506  < 2e-16 ***
## experience_levelMI        0.30142    0.02776  10.859  < 2e-16 ***
## experience_levelSE        0.49300    0.02625  18.782  < 2e-16 ***
## company_sizeM            -0.01274    0.02374  -0.537 0.591464
## company_sizeS            -0.19434    0.03970  -4.896 1.02e-06 ***
## company_location1OT      -0.40320    0.03740 -10.780  < 2e-16 ***
## company_location1US       0.64986    0.02222  29.250  < 2e-16 ***
## employment_type1PT       -0.52681    0.06958  -7.571 4.64e-14 ***
## data_ind                  0.05739    0.06011   0.955 0.339799
## engineer_ind              0.18002    0.05634   3.195 0.001410 **
## scientist_ind             0.22072    0.06073   3.635 0.000282 ***
## analyst_ind              -0.04759    0.13425  -0.355 0.722974
## analytics_ind            -0.13571    0.03869  -3.508 0.000457 ***
## manager_ind               0.10837    0.04837   2.240 0.025130 *
## data_ind:engineer_ind    -0.21956    0.06418  -3.421 0.000630 ***
## data_ind:scientist_ind   -0.23192    0.06844  -3.389 0.000710 ***
## data_ind:analyst_ind     -0.23610    0.13804  -1.710 0.087283 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4092 on 3734 degrees of freedom
## Multiple R-squared:  0.5292, Adjusted R-squared:  0.5267
## F-statistic: 209.8 on 20 and 3734 DF,  p-value: < 2.2e-16
```

```r
vif(buildBackward)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##                             GVIF Df GVIF^(1/(2*Df))
## work_year               1.458791  3        1.064957
## experience_level        1.329467  3        1.048607
## company_size            1.585373  2        1.122103
## company_location1       1.352110  2        1.078333
## employment_type1        1.059377  1        1.029260
## data_ind               13.660967  1        3.696075
## engineer_ind           17.514613  1        4.185046
## scientist_ind          16.803940  1        4.099261
## analyst_ind            60.217066  1        7.759966
## analytics_ind           1.179996  1        1.086276
## manager_ind             1.531781  1        1.237651
## data_ind:engineer_ind  19.132957  1        4.374124
```

```
## data_ind:scientist_ind 18.714785  1        4.326059
## data_ind:analyst_ind   62.864926  1        7.928740
```

Here, I run a first initial backward selection using the StepAIC method. Here, at each step I'm trying to remove a predictor so as to decrease overall AIC of the model, until I reach the smallest AIC value. Here, I then summarize the "best model" that the backward function finds. The backward AIC model has adjusted R^2 essentially unchanged at 0.52.Some of the interaction terms are significant such as Data:Engineer and Data:scientist.Additionally, some of the indicators such as scientist and engineer are significant. However, when I run another VIF test of this model, I realize there are several issues with VIF and they all seem to relate to using both the indicator variables and their interaction terms together in the same model. Every VIF above 10 indicates issues of collinearity. First, I'm going to remove data_ind and analyst_ind as they were non-significant predictors and see if the VIF of that particular model has fewer issues.

```r
VIF_model1<-lm(formula = log_salary ~ work_year + experience_level + company_size +
    company_location1 + employment_type1  + engineer_ind +
    scientist_ind  + analytics_ind + manager_ind +
    data_ind:engineer_ind + data_ind:scientist_ind + data_ind:analyst_ind,
    data = salary4)

summary(VIF_model1)
```

```
##
## Call:
## lm(formula = log_salary ~ work_year + experience_level + company_size +
##     company_location1 + employment_type1 + engineer_ind + scientist_ind +
##     analytics_ind + manager_ind + data_ind:engineer_ind + data_ind:scientist_ind +
##     data_ind:analyst_ind, data = salary4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80521 -0.21360  0.01776  0.24208  2.23585
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            10.72564    0.06052 177.217  < 2e-16 ***
## work_year2021          -0.03968    0.05446  -0.729 0.466224
## work_year2022           0.05602    0.05062   1.107 0.268527
## work_year2023           0.11007    0.05125   2.148 0.031794 *
## experience_levelEx      0.76650    0.04588  16.706  < 2e-16 ***
## experience_levelMI      0.30305    0.02769  10.944  < 2e-16 ***
## experience_levelSE      0.49517    0.02616  18.930  < 2e-16 ***
## company_sizeM          -0.01290    0.02374  -0.543 0.586875
## company_sizeS          -0.19576    0.03966  -4.936 8.34e-07 ***
## company_location1OT    -0.40597    0.03733 -10.876  < 2e-16 ***
## company_location1US     0.65070    0.02220  29.307  < 2e-16 ***
## employment_type1PT     -0.53076    0.06949  -7.638 2.79e-14 ***
## engineer_ind            0.13967    0.03169   4.407 1.08e-05 ***
## scientist_ind           0.18050    0.03910   4.617 4.02e-06 ***
## analytics_ind          -0.13486    0.03868  -3.487 0.000494 ***
## manager_ind             0.12326    0.04664   2.643 0.008260 **
## engineer_ind:data_ind  -0.16241    0.02292  -7.086 1.64e-12 ***
## scientist_ind:data_ind -0.17470    0.03294  -5.303 1.20e-07 ***
## data_ind:analyst_ind   -0.26648    0.03018  -8.830  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4092 on 3736 degrees of freedom
## Multiple R-squared:  0.529,  Adjusted R-squared:  0.5267
## F-statistic: 233.1 on 18 and 3736 DF,  p-value: < 2.2e-16
```

```
vif (VIF_model1)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##                             GVIF Df GVIF^(1/(2*Df))
## work_year               1.452173  3        1.064150
## experience_level        1.313077  3        1.046442
## company_size            1.582990  2        1.121682
## company_location1       1.341388  2        1.076189
## employment_type1        1.056785  1        1.028001
## engineer_ind            5.541663  1        2.354074
## scientist_ind           6.965643  1        2.639251
## analytics_ind           1.179531  1        1.086062
## manager_ind             1.424386  1        1.193477
## engineer_ind:data_ind   2.440316  1        1.562151
## scientist_ind:data_ind  4.336649  1        2.082462
## data_ind:analyst_ind    3.005444  1        1.733622
```

Here I think I've found best model. The backward AIC model gave us a reduced model, but it still had collinearity issues with some of the predictors used. After removing "Data and analyst" indicators from the model, the remaining predictors all have VIF under 10. What this indicates is that outside of job titles of "Data Analyst", "Data Engineer" or "Data scientist", the words "Data" or "Analyst" appears infrequently. So by including the interacted terms, I almost completely encapsulate all instances of both "data" and "analyst" so there is no need to include both of those indicator word variables as they are redundant.

This final best model still has the same Adjusted R^2 as the backward AIC model, but with two fewer predictors as the model now captures the same amount of information without redundant predictors. There is consideration to reduce the model even further, but given that all the predictors in this model have significant p-values for at least of one the indicators, I am inclined to not reduce model any further.

Section 3 INTERPRETATION:

-Recall that we used a log transformation for our response variable. So taking e^{β} will get us the transformed effects of the regression coefficients from our best model. The intercept represents the base level salary. Essentially, every categorical variable we used here leaves out one category as a comparison in the regression. So the intercept salary of e^10.7526= 45,506 represents the expected salary of someone working in 2020, with an entry level job, working at a large company, whose company is based in EU, who works a full time job, and whose job title DOES NOT INCLUDE engineer, scientist, analytics, manager and IS NOT a data engineer, or data scientist, or data analyst. Essentially, the base salary reflects the comparison group salary across every predictor utilized.

-Recall our model here is given by: $ln(y) = \beta_0 + \beta_{workyear2021} * x_1 + \beta_{workyear2022} * x_2 + \beta_{workyear2023} * x_3 + \beta_{experiencelevelEX} * x_4 + \beta_{experiencelevelMI} * x_5 + \beta_{experiencelevelSE} * x_6 + \beta_{companysizeM} * x_7 + \beta_{companysizeS} * x_8 + \beta_{companylocationOT} * x_9 + \beta_{companylocationUS} * x_{10} + \beta_{employmenttypePT} * x_1 1 + \beta_{engineer} * x_{12} + \beta_{scientist} * x_{13} + \beta_{analytics} * x_{14} + \beta_{manager} * x_{15} + \beta_{dataengineer} * x_{16} + \beta_{datascientist} * x_{17} + \beta_{dataanalyst} * x_{18}$

Where x1, x2,.. x18 are all indicator variables taking the value of 1 if the observation has that particular attribute or zero if not.

If we exponentiate both sides we get y = exp ˆ (beta_0+beta_{workyear2021}*x_1+. . . ..)

Which is equivalent to y = expˆ(beta_0) * expˆ(beta_{workyear2021}* x_1) *. . . ..

This essentially means the coefficients in the R output for the model represents a MULTIPLIER for each β. Thus, expected salary can be found by multiplying these regression coefficients against β_0.

-For work year, since the base year (or category that is not included in output) is 2020. The coefficients work_year 2021, 2022, 2023 can be interpreted as the expected MULTIPLIER in salary when COMPARED to 2020.

work_year2021: exp (-0.03968) = 0.961. Here, when compared to 2020, a salary in 2021 is expected to be 0.961 times that of 2020. This indicator variable is not significant.

work_year2022: exp (0.05602) = 1.058. Compared to 2020, a salary in 2022 is expected to be 1.058 times or 5.8% in crease that of 2020. This indicator variable is not significant.

work_year2023: exp(0.1107) = 1.117. Compared to 2020, a salary in 2023 is expected to have 11.7% increase. There is a significant difference for a salary in 2023, compared to that in 2020.

-For experience level, the comparison category is entry level. The coefficients experience_levelEX, experience_levelMI, experience_levelSE represents the expected MULTIPLIER in salary when compared to an entry level position. Note: The R output shows that the pvalue for each of these coefficents is nearly zero and are significannt.

experience_levelEX: exp (0.7665) = 2.152. Compared to an entry-level position, an Executive is expected to make 2.152x or 115% more.

experience_levelMI: exp (0.30305) = 1.354. Compared to an entry level position, a mid level position is expected to make 35.4% more.

experience_levelSE: exp (0.49517) = 1.6408. Compared to an entry level position, a senior level engineer is expected to make 64.1% more.

-For company size, the comparison category is Large company. The coefficients company_sizeM and company_sizeS represents the expected MULTIPLIER in salary when compared to a

company_sizeM: exp (-0.012) = 0.988. Compared to a large Company, a job from medium Company is expected to make 1.2% less. There is not a signficant difference here.

company_sizeS: exp (-0.19576) = 0.822. Compared to a large Company, a job from a small company is expected to make 17.8% less. There is a strong significant difference in salary for this regression coefficient.

-For Company location, the comparison category is EU. The coefficients Company_location US and Company_locationOT represents the expected MULTIPLIER in salary when compared to Company located in EU. Both of these coefficients are significant with nearly 0 pvalue.

company_locationOT: exp (-0.040597) = 0.96. Compared to a Company from the EU, a country not from US or EU is expected to make 4% less.

company_locationUS: exp (0.6507) = 1.917. Compared to a Company from the EU, a Company is the US is expected to make 91.7% more.

employment_typePT:exp (-0.53076) = 0.588. Compared to a full time job, a part time job is expected to make 41.2% less. This is a significant predictor with nearly zero pvalue.

-For job title word indicators, these are all binary indicator variables, so the coefficients represent the expected MULTIPLIER in having the key word indicator vs not having the keyword in one's job title.

engineer_ind: exp (0.13967) = 1.150. Compared to a non-engineer, an engineer is expected to make 15% more. There is a significant difference here.

scientist_ind: exp (0.1805) = 1.198. Compared to a non-scientist, a job title with scientist is expcted to make 19.8% more. There is a significant difference here.

analytics_ind: exp (-0.013486) = 0.987. Compared to a non-analytics job, a job title with analytics is expected to make 1.3% less. There is a strong significant difference.

manager_ind: exp (0.12326) = 1.131. Compared to a non-manager job, a job title with manager is expected to make 13.1% more. There is a strong significant difference.

-For interacted job title indicators, these can be interpreted as the expected MULTIPLIER in salary in having both key words compared to not having both keywords in one's job title. All of these interacted terms have nearly zero pvalue and have strong significance.

data:engineer- exp(-0.16241) = 0.85. Compared to a non-data engineer job, a data engineer job is expected to make 15% less.

data:scientist- exp(-0.17470) = 0.8397. Compared to a non-data scientist job, a data scientist role is expected to make 16% less.

data:analyst- exp (-0.26648) = 0.766. Compared to a non-data analyst role, a data analyst role is expected to make 23.4 % less.

For practical purposes, I can use this model to predict expectations for various data science roles. Let say I wanted to predict the salary of a future job that would be an entry level role, at a medium size company, in the US, working full time, and for role title "Data Analyst". This could be calculated as $45,506 * (1.117)* (0.988)* (1.917)*(0.766) = 73,744 USD. (I use work_year 2023 as a best proxy for current year salary expectations).

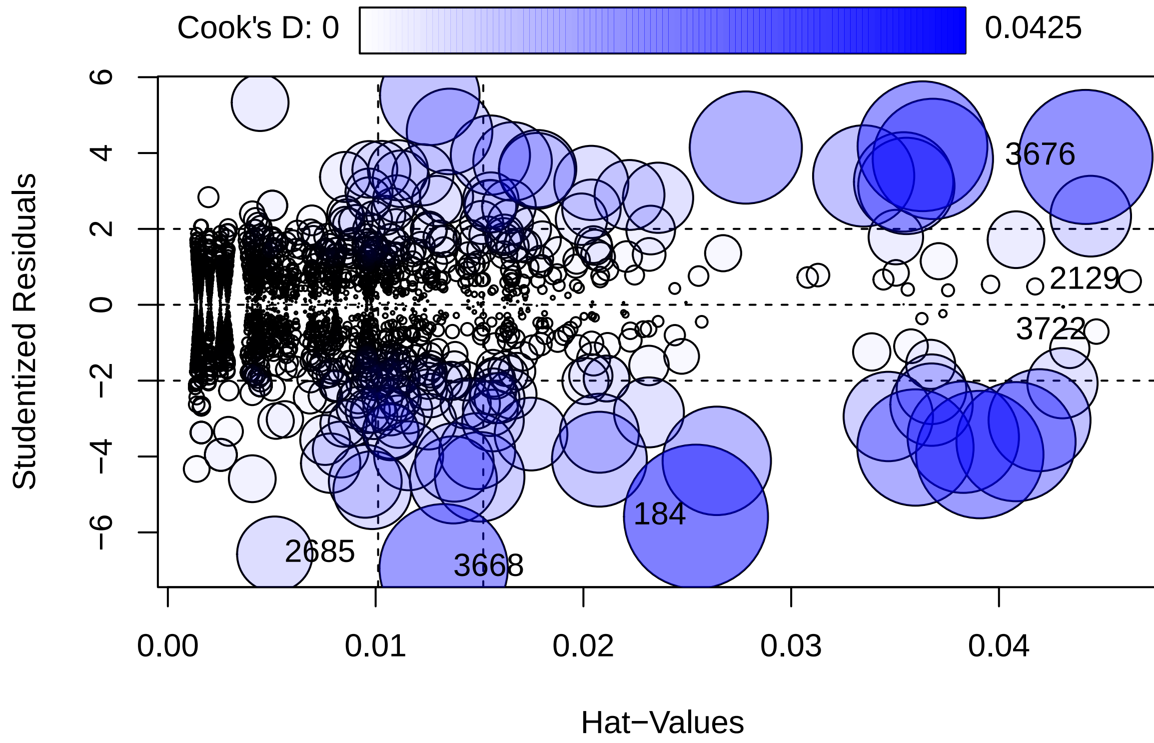
Take another example,an Executive role, at a large Company, in the US, working full time, with manager title can expect to make $45,506* (1.117)* (2.152) * (1.917) * 1.131 = 237,164 USD.

Based on these regression outputs, I think anyone applying to a data science role could use these outputs and same interpretation methodology to see what salary expectations are.

Overall, this model is fairly strong, but isn't quite perfect either. An adjusted R^2 of 0.5267 is fairly strong, but not every indicator within this model is significant. Only work_year 2023 indicator from the work_year predictor is significant, but I still want to include this predictor since a large number of observations are from this year. The only other non-significant indicator is large_companyM, but otherwise all other predictors I included seem to be significant. The predictive power of this model is fairly strong as I can predict salary based on job title/ role attributes. Note, that the best proxy for predicting current salaries for work_year is to use 2023 as a proxy especially since I coded work_year as a categorical variable. Obviously, there are some limitations in my model. For example, I could imagine an extremely niche job title that is extremely lucrative and pays near executive level salary, might be under predicted in this model to have a lower salary than it should. However, if there is a common data science role with a common title and standard attributes (full time, in the US,etc) than I believe my model should have a very strong prediction/ predictive power.

Section 4: Model Refinement

```r
influencePlot(VIF_model1)
```




```
##         StudRes         Hat        CookD
## 184  -5.5870436 0.025417852 0.0425042809
## 2129  0.6226226 0.046303934 0.0009907756
## 2685 -6.5750096 0.005145798 0.0116372635
## 3668 -6.9455776 0.013272072 0.0337246397
## 3676  3.8972227 0.044166141 0.0367974452
## 3722 -0.7065172 0.044686872 0.0012290919
```

```
print(salary4[c(84,2379,2681,2685,3668,184,3722,2129,3676),])
```

```
##      work_year experience_level                job_title salary_in_usd
## 84        2022               EN             AI Developer        300000
## 2379      2022               EN               BI Analyst         12000
## 2681      2022               SE               BI Analyst        200000
## 2685      2022               MI             NLP Engineer          5132
## 3668      2021               MI           Data Scientist          5679
## 184       2020               Ex       Staff Data Analyst         15000
## 3722      2020               SE  Computer Vision Engineer         60000
## 2129      2022               EN Data Analytics Consultant         50000
## 3676      2021               Ex  Principal Data Scientist        416000
##      remote_ratio company_size employee_residence1 company_location1
## 84             50            L                  OT                OT
## 2379          100            L                  OT                US
## 2681          100            S                  OT                OT
## 2685          100            M                  EU                EU
```

```
## 3668         100           S                  OT               US
## 184            0           M                  OT               US
## 3722         100           S                  OT               US
## 2129         100           S                  EU               US
## 3676         100           S                  US               US
##      employment_type1 data_ind engineer_ind scientist_ind analyst_ind ML_ind
## 84                 FT        0            0             0           0      0
## 2379               PT        0            0             0           1      0
## 2681               FT        0            0             0           1      0
## 2685               FT        0            1             0           0      0
## 3668               FT        1            0             1           0      0
## 184                FT        1            0             0           1      0
## 3722               PT        0            1             0           0      0
## 2129               PT        1            0             0           0      0
## 3676               PT        1            0             1           0      0
##      analytics_ind manager_ind log_salary
## 84               0           0  12.611538
## 2379             0           0   9.392662
## 2681             0           0  12.206073
## 2685             0           0   8.543251
## 3668             0           0   8.644530
## 184              0           0   9.615805
## 3722             0           0  11.002100
## 2129             1           0  10.819778
## 3676             0           0  12.938441
```

In this step, after I have determined my best model and interpreted its results, it is helpful to refine the model by looking at potential outliers. I produce an influence plot of my best model to identify large influence points. Recall that influence is a measure of both leverage and how much of an outlier a point is. Points that are very far right on the x-axis have large leverage and points with large absolute value studentized residuals can be considered outliers from the model.The influence plot has identified 6 points worth exploring and the regression assumptions plots I conducted earlier in section 2 also identified 3 additional points that might be outliers. I will individually explore each point and comment on whether I would keep the data observation.

Point 3676: This was likely flagged as a high influence point as this is one of the highest earning salaries in the entire data set. It has high leverage based on sheer salary amount and the salary is likely an outlier based on the fact it was coded as a Part time job. It is likely an error that this observation was coded as part time job, but since I cannot confirm, I would remove this observation.

Point 2129: This is a large leverage, but not an outlier point. This is a part-time, entry level position so $50k salary doesn't seem contextually abnormal. It is a large leverage point as this is one of the lowest salaries in the dataset, but I wouldn't remove from dataset since it isn't an outlier.

Point 3733: A senior engineer making $60k working part time. Again, this is a large leverage observation not an outlier. 60k is one of the lower salaries in the entire data set, but its feasible that a senior engineer would only make this much given a part time status. I wouldn't remove this observation.

Point 184: I would remove this observation. Given that the company is in the US, it is not possible that a Full time employee would only be making $15k salary. This observation was likely coded incorrectly, so it'd be better to drop this data point.

Point 3668: Again, a FT employee for a Company in the US could not possibly be making a $6k salary. I would drop this observation.

Point 2685: A FT time employee working for an EU company could not contextually make $5,132 salary. I would drop this observation.

These below point were identified by the regression assumption plots as possible outliers.

Point 2681: This might just be an outlier because this is one of the highest salaries for a Company location "OT". I would keep this observation as contextually, a full time senior level engineer could make $200k.

Point 2379: $12k salary for a Part time entry level position seems reasonable.There is nothing obvious about this observation that would make it an outlier. I would keep this observation.

Point 84: The only thing that makes this an outlier is that this is coded as entry level position with salary of $300k. This is not a US company salary. This seems pretty outside the compensation range of the lowest experience level. I would also drop this point from the dataset.

```r
#remove 84th, 184, 3668, and 3676 rows.
salary5 <- salary4[-c(84, 184, 3668, 3676), ]

View(salary5)

VIF_model1<-lm(formula = log_salary ~ work_year + experience_level + company_size +
    company_location1 + employment_type1  + engineer_ind +
    scientist_ind  + analytics_ind + manager_ind +
    data_ind:engineer_ind + data_ind:scientist_ind + data_ind:analyst_ind,
    data = salary5)

summary(VIF_model1)
```

```
##
## Call:
## lm(formula = log_salary ~ work_year + experience_level + company_size +
##     company_location1 + employment_type1 + engineer_ind + scientist_ind +
##     analytics_ind + manager_ind + data_ind:engineer_ind + data_ind:scientist_ind +
##     data_ind:analyst_ind, data = salary5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.66850 -0.21327  0.01728  0.24223  2.16546
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           10.738611   0.059868 179.373  < 2e-16 ***
## work_year2021         -0.062229   0.053938  -1.154 0.248689
## work_year2022          0.018210   0.050186   0.363 0.716733
## work_year2023          0.072336   0.050811   1.424 0.154634
## experience_levelEx     0.778885   0.045527  17.108  < 2e-16 ***
## experience_levelMI     0.311763   0.027280  11.428  < 2e-16 ***
## experience_levelSE     0.500163   0.025768  19.410  < 2e-16 ***
## company_sizeM         -0.004788   0.023374  -0.205 0.837701
## company_sizeS         -0.180652   0.039223  -4.606 4.25e-06 ***
## company_location1OT   -0.417969   0.036790 -11.361  < 2e-16 ***
## company_location1US    0.654938   0.021863  29.957  < 2e-16 ***
## employment_type1PT    -0.577643   0.069229  -8.344  < 2e-16 ***
## engineer_ind           0.147951   0.031216   4.740 2.22e-06 ***
## scientist_ind          0.189729   0.038499   4.928 8.66e-07 ***
## analytics_ind         -0.134620   0.038055  -3.538 0.000409 ***
## manager_ind            0.129936   0.045916   2.830 0.004681 **
## engineer_ind:data_ind -0.163853   0.022552  -7.266 4.50e-13 ***
## scientist_ind:data_ind -0.174950  0.032430  -5.395 7.29e-08 ***
## data_ind:analyst_ind  -0.255469   0.029747  -8.588  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4026 on 3732 degrees of freedom
## Multiple R-squared:  0.539,  Adjusted R-squared:  0.5368
## F-statistic: 242.4 on 18 and 3732 DF,  p-value: < 2.2e-16
```

In all, there are 4 points I would drop from above. Here, I remove the outlier observations I want to drop and rerun the same best regression model that I used earlier. The interpretation of coefficients is the same as above so I will not go through that again, but the adjusted R^2 of the model did improve to 0.5368. Obviously, dropping 4 points isn't enough to change the significance of the predictors I utilized, but the fit of the model definitely had meaningful improvement.

Section 5: Evaluation and discussions

Overall, there are many constraints that limit the extent we can draw conclusions and model build. 1) I'm only able to include predictors that were available to me within the data set. Even more, I showed that several of these predictors/ response variable were ultimately redundant or had issues of multi-collinearity. 2) My entire model only consists of categorical and indicator variables. Ideally, I would have preferred to have some continuous predictors within my model. A couple that come to mind are years of experience, number of software certifications,etc. Maybe another important categorical variable is level of education. Although there is no telling if these predictors would necessarily improve the model, they are definitely worth exploring and would alter the strength of my best regression model. 3) A final limitation of the data set were that despite having a large sample size, a lot of a observations were still heavily concentrated around certain categories. I.e a vast majority of observations for Company location were clustered around the US or there were ~1% of the sample that was part time. There may be underlying limitations to which the data was sampled and collected.

The model I ultimately determined, is a fairly strong model in that I tried to eliminate collinaerity between predictors and the predictors I included in final model were all significant.The final model did have a reasonable strong adjusted R^2. However, some of my regression plots even with a transformed response variables still demonstrated some issues. Obviously, in a real life data set, regression assumptions cannot be perfect and small issues may still exist no matter how hard statisticians try to address them. However, I still believe there may be some limitations as I was not able to fully correct them within the scope of this paper. I don't believe other approaches could have been taken given the scope of the variables available to me. I believe salary as the response variables with categorical predictors was the only feasible approach. I transformed the response variable based on regression assumption plots. I believe the only alternative approach is just to dive deeper in my usage of job title indicator variables or collapsing variables differently (see below).

If I had more time, I would have attempted to delve deeper into the job title indicators that I created. I only was able create indicators for the most frequently occurring words which often represent mid-level positions. I suspect there may be certain words in job titles that reflect more lucrative salaries.Additionally, I assumed that the number of observations may have been too small to get reasonable standard errors for Company_location had I broken out the location of counties even more than just US, EU, and other. Although, I don't think it would drastically alter my model, these are a couple examples of deeper analysis in the future.Also, if I could collect more data and observations across a large range of years, I think that would lend work_year to being coded as a a continuous variable, and the regression coefficient of that variable could then be interpreted as " average inflation effect" for each unit increase in year.

With these findings, I have shown that there are a number of variables that are significant in determining salary. The year in which one applies for a position in itself matters. While in this data set, I coded year as categorical and can only draw concrete conclusions for the years included in this model, there is certainly an inflation adjustment that needs to be considered. Experience level, Company size, Company location, and employment type were all shown to be significant predictors in my model. Finally, having certain key word or combination of words in job title was also shown to being a significant predictor. The implications of these results could help those applying for data science positions. Ultimately, one would just have to ask themselves what attributes they fit under to see what their expected salary would look like. For example, I personally might be applying to a large company, full time, at an entry level position, in the US. If I figure out what my job position would be, I can interpret the regression model above accordingly and figure out my expected salary. Someone else with different qualifications working somewhere else could use the same steps to figure out their expected salary,etc. Overall, models like these help give transparency for salary expectations and can give people expectations before even applying to data science roles.