VIETNAM NATIONAL UNIVERSITY, HANOI
INTERNATIONAL SCHOOL

FINAL EXAM

Subject: Statistical models for data analysis 1

Class: INS3079_01
Group: 06

Topic: Salary and Job Satisfaction Prediction of the Data Science Field

Member: Vũ Thị Quế Anh – 20070900
Bùi Ngọc Anh – 20070890
Nguyễn Thu Hoài – 20070931
Dương Diệu Linh – 20070944

Lecturer: Phạm Thị Việt Hương

Hanoi, 1/2024

TABLE OF CONTENTS

I. INTRODUCTION

These days, the most significant component of employee remuneration and a key component of organizational strategy is salary. Knowledge about pay ranges for potential careers can be helpful for job seekers. Labor can maximize the development of their talents by focusing on high-paying industries; besides job satisfaction that has been highly emphasized today for organizational commitment. However, there is a lack of research on the implementation of analytical models in predicting salary and job satisfaction of technology jobs for job seekers; taking into account the uncertainty of key features that contribute to salary increases for data science jobs. Therefore, this project aims to develop a model that successfully predicts multiple linear regression salaries while identifying key characteristics that help earn higher salaries in data scientist roles.

1. Dataset

The Salary prediction dataset used for this research has 742 entries and is derived from the Kaggle. It is a compilation of tech job positions scraped from glassdoor.com in 2017. In this dataset, there is one column labeled "job title" which refers to tech job positions. The remaining 31 features will represent the important information associated with the tech job. However, 12 features will only be selected as our study focus of this project.

No.	Variable	Variable Type	Description
1	Id	Numeric	The unique identifier for the job posting
2	Job Title	String	The tech job positions
3	Salary Estimate	String	The approximate range or value of compensation that an individual can expect to receive for a particular job or position.
4	Job Description	String	A document that outlines the specific tasks, responsibilities, qualifications, and expectations associated with a particular job or position within an organization.
5	Rating	Numeric	The evaluation or assessment of an individual's performance, skills, or qualities based on a predetermined set of criteria.
6	Company Name	String	The name of the company
7	Location	String	The company address
8	Headquarters	String	The company's headquarters
9	Size	String	The company size (employees)

10	Founded	Numeric	Year of establishment of the company
11	Type of ownership	String	The type of ownership
12	Industry	String	The industry of company
13	Sector	String	The sector of company
14	Revenue	String	The revenue of company
15	Competitors	String	The competitors of company
16	hourly	Numeric	A method of calculating and paying wages or compensation based on the number of hours worked.
17	employer_provided	Numeric	The company needs to be provided with employees
18	min_salary	Numeric	The lowest amount of salary an employee receives
19	max_salary	Numeric	The highest salary an employee receives
20	avg_salary	Numeric	Average salary received by employees
21	company_txt	String	Company partner
22	job_state	String	The state where the job is located
23	same_state	String	A binary indicator of whether the job is in the same state as the person looking at the job
24	age	Numeric	The age of the person looking at the job
25	python_yn	String	A binary indicator of whether the person looking at the job knows Python
26	R_yn	String	A binary indicator of whether the person looking at the job knows R
27	spark	String	A binary indicator of whether the person looking at the job knows Spark
28	aws	String	A binary indicator of whether the person looking at the job knows AWS
29	excel	String	A binary indicator of whether the person looking at the job knows Excel

30	job_simp	String	A simplified job title
31	seniority	String	The seniority of the job
32	desc_len	Numeric	The length of the job description
33	num-comp	Numeric	The number of competitors for the job

2. *Methodology*

- Data Cleaning: Apply appropriate method for data cleansing
- Exploratory Data Analysis: Create visualization of the cleansed data to provide insights and to answer the objective
- Build model: create multiple linear regression

II. DISCARD OUTLIERS AND VARIABLES

Because the purpose of the project is to predict job salary in the data science field. The remaining 31 features will represent the important information associated with the tech job. However, 12 features will only be selected as our study focus of this project.

1. *Data understanding*

a) Check missing value

```
> summary(is.na(employee))
   Rating          Size           Industry        Revenue        avg_salary        age
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:742       FALSE:742       FALSE:742       FALSE:742       FALSE:742       FALSE:742
 python_yn          R_yn           spark            aws            excel          job_simp
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:742       FALSE:742       FALSE:742       FALSE:742       FALSE:742       FALSE:742
```

There are no missing values in the data.

b) Structure of data

```
> str(employee)
tibble [742 × 12] (S3: tbl_df/tbl/data.frame)
 $ Rating    : num [1:742] 3.8 3.4 4.8 3.8 2.9 3.4 4.1 3.8 3.3 4.6 ...
 $ Size      : chr [1:742] "501 to 1000 employees" "10000+ employees" "501 to 1000 employees" "1001 to 5000 empl
oyees" ...
 $ Industry  : chr [1:742] "Aerospace & Defense" "Health Care Services & Hospitals" "Security Services" "Energy"
...
 $ Revenue   : chr [1:742] "$50 to $100 million (USD)" "$2 to $5 billion (USD)" "$100 to $500 million (USD)" "$5
00 million to $1 billion (USD)" ...
 $ avg_salary: num [1:742] 72 87.5 85 76.5 114.5 ...
 $ age       : num [1:742] 47 36 10 55 22 20 12 15 6 11 ...
 $ python_yn : num [1:742] 1 1 1 1 1 1 0 1 0 1 ...
 $ R_yn      : num [1:742] 0 0 0 0 0 0 0 0 0 0 ...
 $ spark     : num [1:742] 0 0 1 0 0 0 0 1 0 1 ...
 $ aws       : num [1:742] 0 0 0 0 0 1 0 1 0 0 ...
 $ excel     : num [1:742] 1 0 1 0 1 1 1 1 0 0 ...
 $ job_simp  : chr [1:742] "data scientist" "data scientist" "data scientist" "data scientist" ...
```

The data is mostly numeric, but there is some character data such as Size, Industry and job_simp.

c) Summary data

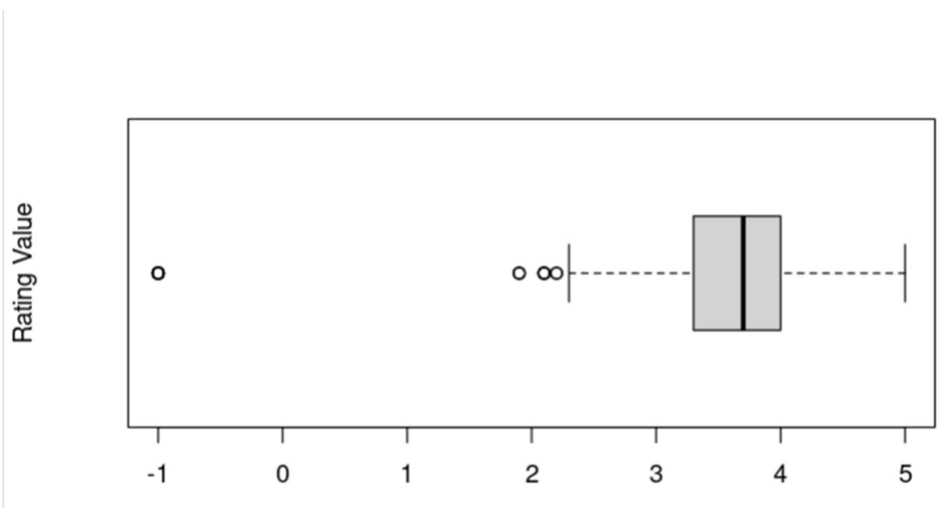
```
> summary(employee)
     Rating            Size             Industry           Revenue            avg_salary          age
 Min.   :-1.000   Length:742         Length:742         Length:742         Min.   : 13.5   Min.   : -1.00
 1st Qu.: 3.300   Class :character   Class :character   Class :character   1st Qu.: 73.5   1st Qu.: 11.00
 Median : 3.700   Mode  :character   Mode  :character   Mode  :character   Median : 97.5   Median : 24.00
 Mean   : 3.619                                                            Mean   :100.6   Mean   : 46.59
 3rd Qu.: 4.000                                                            3rd Qu.:122.5   3rd Qu.: 59.00
 Max.   : 5.000                                                            Max.   :254.0   Max.   :276.00
   python_yn          R_yn               spark             aws              excel          job_simp
 Min.   :0.0000   Min.   :0.000000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Length:742
 1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
 Median :1.0000   Median :0.000000   Median :0.0000   Median :0.0000   Median :1.0000   Mode  :character
 Mean   :0.5283   Mean   :0.002695   Mean   :0.2251   Mean   :0.2372   Mean   :0.5229
 3rd Qu.:1.0000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.000000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

This table gives us an overview of the employee data set:

- The average employee salary is $100.60K.
- The average age of employees is 46.59 years old.
- About 53% of employees use Python.
- About 2.7% of employees use R.
- About 23.72% of employees use Spark.
- About 23.72% of employees use AWS.
- About 52.3% of employees know how to use Excel.

2. Data cleaning

a) Rating



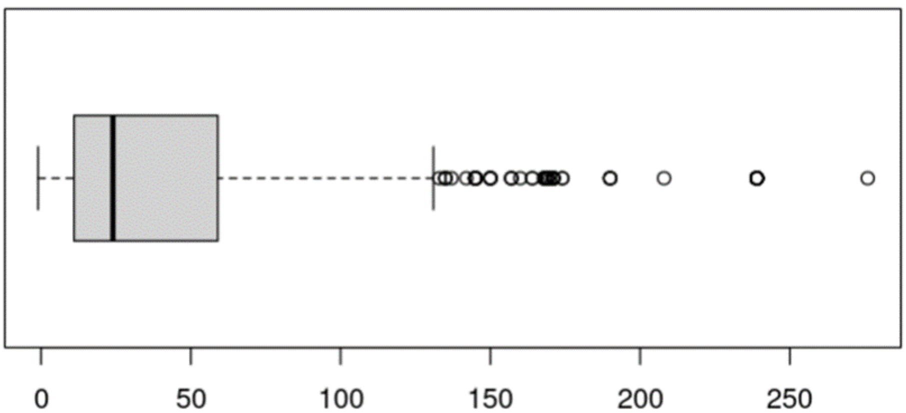
We remove 21 outliers, instances 742 to 721, and bin Ratings to categorical:

- 0.0 - 0.9 Very Dissatisfied
- 1.0 - 1.9 Dissatisfied
- 2.0 - 2.9 Neutral
- 3.0 - 3.9 Satisfied
- 4.0 - 5.0 Very Satisfied

```
                Rating
Very Dissatisfied:  0
Dissatisfied     :  0
Neutral          : 79
Satisfied        :470
Very Satisfied   :172
```

Most employees are satisfied with their work at the company, 79 people feel normal and 172 people feel very satisfied.

b) Age



We assume that the age 18-70 are the working age group. There are 290 instances of min_age < 18 & 156 instances of max_age > 70 which are outliers. We replace <18 outliers with IQR Q1 (30) & replace >70 outliers with IQR Q3(52).

c) Average Salary

Because the data unit is $K, we will have to convert $.

```
employee$avg_salary = employee$avg_salary*1000
```

d) Job Title Simplified

Replace NA with other tech jobs.

```
employee$job_simp = str_replace_all(employee$job_simp, "\\bna\\b", "Other tech jobs")
```

e) Number of Employees categorized in company size

Bin Size to Categorical:

- 1 to 200 employees - “Small”
- 201 to 1000 employees - ‘Medium’
- 1001 and above - ‘Large’

f) Revenue categorized by business size

Bin Revenue to categorical:

- Less than $1 million (USD) - ‘micro-bus’
- $1 million to 10 million (USD) - ‘small-bus’
- $10 million to 50 million (USD) - ‘medium-bus’
- $50 million and above (USD) - ‘large-bus’

Impute Missing Value Revenue (195) and Size (2) by MICE. polyreg method was chosen because there are more than 2 factor variables.

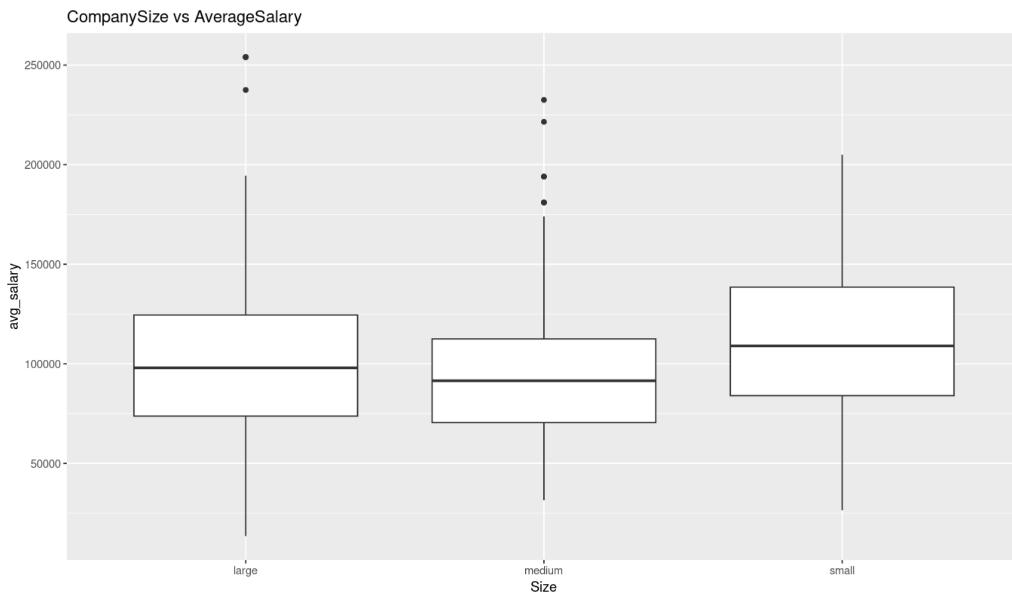
```
iter imp variable
 1    1   Size   Revenue
 1    2   Size   Revenue
 1    3   Size   Revenue
 1    4   Size   Revenue
 1    5   Size   Revenue
 2    1   Size   Revenue
 2    2   Size   Revenue
 2    3   Size   Revenue
 2    4   Size   Revenue
 2    5   Size   Revenue
 3    1   Size   Revenue
 3    2   Size   Revenue
 3    3   Size   Revenue
 3    4   Size   Revenue
 3    5   Size   Revenue
 4    1   Size   Revenue
 4    2   Size   Revenue
 4    3   Size   Revenue
```

After cleaning data, we save the new data into “employee_clean.csv”.

Rating	Size	Industry	Revenue	avg_salary	age	python_yn	R_yn	spark	aws	excel	job_simp
Satisfied	medium	Aerospace & Defense	medium-bus	72000	47	1	0	0	0	1	data scientist
Satisfied	large	Health Care Services & Hospitals	large-bus	87500	36	1	0	0	0	0	data scientist
Very Satisfied	medium	Security Services	large-bus	85000	30	1	0	1	0	1	data scientist
Satisfied	large	Energy	large-bus	76500	55	1	0	0	0	0	data scientist
Neutral	small	Advertising & Marketing	medium-bus	114500	22	1	0	0	0	1	data scientist
Satisfied	medium	Real Estate	large-bus	95000	20	1	0	0	1	1	data scientist
Very Satisfied	medium	Banks & Credit Unions	large-bus	73500	30	0	0	0	0	1	data scientist
Satisfied	medium	Consulting	medium-bus	114000	30	1	0	1	1	1	data scientist
Satisfied	large	Health Care Services & Hospitals	large-bus	61000	30	0	0	0	0	0	Other tech jobs
Very Satisfied	small	Internet	large-bus	140000	30	1	0	1	0	0	data scientist
Satisfied	medium	Other Retail Stores	large-bus	163500	30	1	0	0	0	0	data scientist

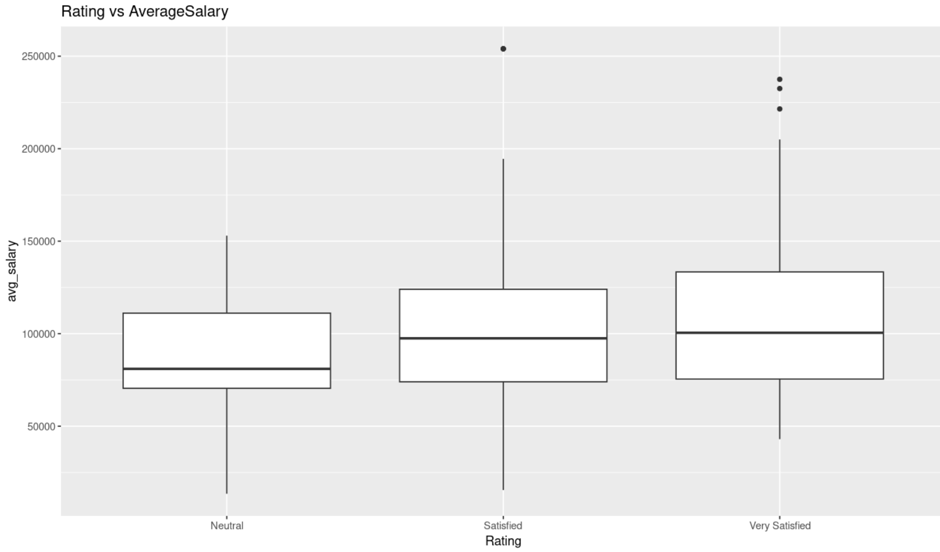
III. EXPLORATORY DATA ANALYSIS

1. Compare the average salary distribution for each company size



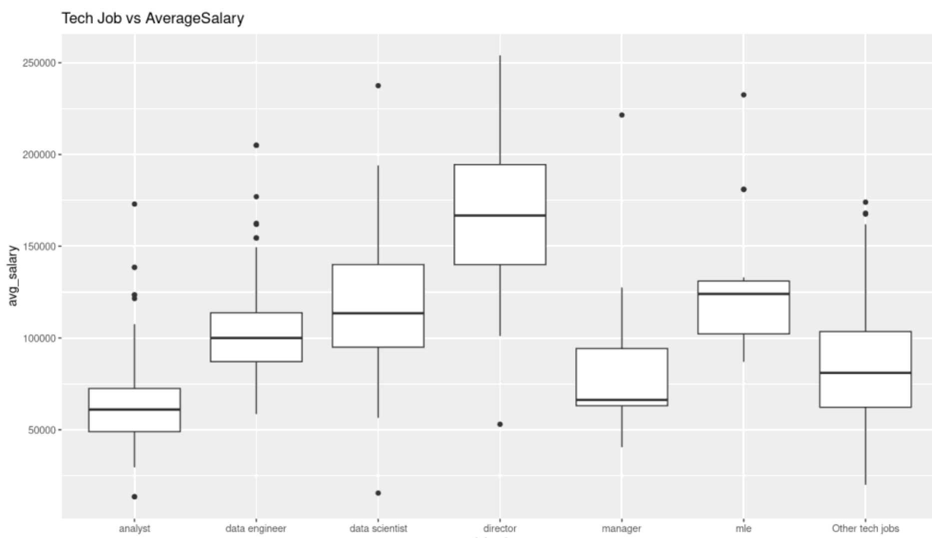
Overall, the 3 batches of data look as if they were generally distributed similarly. The interquartile ranges are reasonably similar (as shown by the lengths of the boxes). The median average salary of small-sized companies is greater than large-sized companies and medium-sized companies. There are outliers observed across large and medium-sized companies which indicate that these data values are far away from other data values which needs further evaluation.

2. *Compare average salary distribution for rating (job satisfaction)*



Because Dissatisfied and Very Dissatisfied both have the value 0, they will not appear in the box plot. Overall, individuals who rated very satisfied scored higher median average salary compared to individuals who rated neutral and satisfied.

3. *Compare salary distribution for each tech job*



Overall, the director position has a higher median average salary compared to mle, data scientist, data engineer, other tech jobs, manager, and analyst. As shown in the analyst, data engineer, manager, and mle column, the box plots are comparatively short which indicates that individuals that are in these mentioned positions have around the same average salary as each other. In addition, the 4 sections of the box plot for manager and mle are relatively uneven in size. This indicates that individuals in that particular position respectively have similar average salary at certain parts of the scale, but in other parts of the scale, individuals in that position are more variable in their average salary. The short upper whisker in the mle boxplot indicates that their average salary is varied among the least positive quartile group, and very similar for the most positive quartile group. There are outliers observed across all the positions which indicate that these data values are far away from other data values which need further evaluation.

IV. MULTIPLE LINEAR REGRESSION

1. *Create a Multiple Regression Model*

```
sal_df<-
  dplyr::select(employee_clean,-c("Industry", "Rating"))
norm_minmax <- function(x){
  (x- min(x)) /(max(x)-min(x))
}
sal_df[sapply(sal_df, is.numeric)] <- lapply(sal_df[sapply(sal_df, is.numeric)],norm_minmax)
sal_df[sapply(sal_df, is.character)] <- lapply(sal_df[sapply(sal_df, is.character)],as.factor)
str(sal_df)
tibble [730 × 10] (S3: tbl_df/tbl/data.frame)
 $ Size      : Factor w/ 3 levels "large","medium",..: 2 1 2 1 3 2 2 2 1 3 ...
 $ Revenue   : Factor w/ 4 levels "large-bus","medium-bus",..: 2 1 1 1 2 1 1 2 1 1 ...
 $ avg_salary: num [1:730] 0.243 0.308 0.297 0.262 0.42 ...
 $ age       : num [1:730] 0.5686 0.3529 0.2353 0.7255 0.0784 ...
 $ python_yn : num [1:730] 1 1 1 1 1 1 0 1 0 1 ...
 $ R_yn      : num [1:730] 0 0 0 0 0 0 0 0 0 0 ...
 $ spark     : num [1:730] 0 0 1 0 0 0 0 1 0 1 ...
 $ aws       : num [1:730] 0 0 0 0 0 1 0 1 0 0 ...
 $ excel     : num [1:730] 1 0 1 0 1 1 1 1 0 0 ...
 $ job_simp  : Factor w/ 7 levels "analyst","data engineer",..: 3 3 3 3 3 3 3 3 7 3
```

Normalizing the numeric columns will help to ensure that all features are on the same scale, which can improve the performance of some machine learning algorithms. Converting the character columns to factors will make it possible to use them in categorical feature encoding.

- Size is a factor with 3 levels: large, medium, small
- Revenue is a factor with 4 levels: large-bus, medium-bus, small-bus, micro-bus
- job_simp is a factor with 7 levels: analyst, data engineer, data scientist, director, manager, mle and Other tech jobs.
- Columns like avg_salary, age, python_yn, R_yn, spark, aws, excel are numeric, where python_yn, R_yn, spark, aws, excel are dummy values.

```
slmModel=lm(avg_salary~.,data=sal_df)
summary(slmModel)
```

```
> summary(slmModel)

Call:
lm(formula = avg_salary ~ ., data = sal_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.44954 -0.08291 -0.01758  0.06421  0.55686 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)             0.2223706  0.0197874  11.238  < 2e-16 ***
Sizemedium             -0.0258015  0.0138656  -1.861 0.063179 .  
Sizesmall              -0.0063300  0.0228514  -0.277 0.781856    
Revenuemedium-bus       0.0001656  0.0159345   0.010 0.991710    
Revenuemicro-bus       -0.0291032  0.0508808  -0.572 0.567510    
Revenuesmall-bus        0.0609059  0.0304530   2.000 0.045879 *  
age                    -0.0463317  0.0233898  -1.981 0.047991 *  
python_yn               0.0398789  0.0119644   3.333 0.000903 ***
R_yn                   -0.0086880  0.0956384  -0.091 0.927644    
spark                   0.0094428  0.0138482   0.682 0.495535    
aws                     0.0172365  0.0128893   1.337 0.181559    
excel                   0.0077495  0.0101699   0.762 0.446310    
job_simpdata engineer   0.1388583  0.0200899   6.912 1.06e-11 ***
job_simpdata scientist  0.1918565  0.0170514  11.252  < 2e-16 ***
job_simpdirector        0.4222977  0.0388002  10.884  < 2e-16 ***
job_simpmanager         0.0747120  0.0315057   2.371 0.017986 *  
job_simpmle             0.2431506  0.0326986   7.436 2.98e-13 ***
job_simpOther tech jobs 0.0806056  0.0172279   4.679 3.45e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.133 on 714 degrees of freedom
Multiple R-squared:  0.3453,	Adjusted R-squared:  0.3297 
F-statistic: 22.15 on 17 and 714 DF,  p-value: < 2.2e-16
```

This model uses linear regression analysis to predict the average salary of an employee. The results of the model show that the variables that are statistically significant for the average employee salary include:

- Size: Employees working at large companies have higher average salaries than employees working at small or medium-sized companies.
- Revenue: Employees working at companies with high revenue have higher average salaries than employees working at companies with low revenue.
- python_yn: Employees with Python skills have a higher average salary than employees without this skill.
- R_yn: Employees with skill R have a higher average salary than employees without this skill.

- job_simp: Employees working in positions such as data engineer, data scientist, director, or manager have higher average salaries than employees working in other positions.
- Additionally, the model also shows that employee age is negatively related to average salary. This means that average employee salaries tend to decrease as they get older.

The results of this model are consistent with what we expected. Large companies with high turnover often have higher salaries than small companies with low turnover. Employees with technical skills and work experience are worth more and can therefore earn higher salaries.
Variables like Revenue, R_yn, spark, aws, excel have also high p-value indicating that it needs to be handled differently. However, we still have to retain the age variable to solve the following problems. Other variables like python_yn and job_simp are more significant and explain variability in salary. Multiple R-squared = 0.3453, meaning this model explains 34.53% of the variation in the dependent variable (avg_salary) with the independent variables (Size, Revenue, age, python_yn, R_yn, spark, aws, excel, job_simp).

2. *Check collinearity*

```
> vif(slmModel)
            Sizemedium                  Sizesmall      Revenuemedium-bus
              1.793465                   3.061425               2.034107
      Revenuemicro-bus           Revenuesmall-bus                    age
              1.158607                   1.983476               1.307547
             python_yn                       R_yn                  spark
              1.475275                   1.031851               1.398144
                   aws                      excel   job_simpdata engineer
              1.256162                   1.068468               2.259474
 job_simpdata scientist          job_simpdirector        job_simpmanager
              2.835437                   1.169285               1.198005
           job_simpmle job_simpOther tech jobs
              1.290443                   2.244142
```

As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity. There is no collinearity: all variables have a value of VIF well below 5.

V. INTERACTION MODEL AND ADDITION MODEL

1. *Interaction model*

a) Interaction model 1

avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark + aws + excel + job_simp + age:job_simp

```
> summary(slmModel1_interaction)

Call:
lm(formula = avg_salary ~ . + age:job_simp, data = sal_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.48292 -0.08166 -0.01699  0.06482  0.55990 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  0.1951334  0.0272576   7.159 2.04e-12 ***
Sizemedium                  -0.0228802  0.0140081  -1.633 0.102838    
Sizesmall                   -0.0030692  0.0228790  -0.134 0.893323    
Revenuemedium-bus           -0.0009724  0.0159621  -0.061 0.951439    
Revenuemicro-bus            -0.0296268  0.0506295  -0.585 0.558621    
Revenuesmall-bus             0.0566188  0.0303148   1.868 0.062217 .  
age                          0.0249948  0.0623070   0.401 0.688426    
python_yn                    0.0409830  0.0119106   3.441 0.000614 ***
R_yn                        -0.0030043  0.0950918  -0.032 0.974805    
spark                        0.0076200  0.0138606   0.550 0.582660    
aws                          0.0177528  0.0129383   1.372 0.170463    
excel                        0.0112239  0.0102667   1.093 0.274661    
job_simpdata engineer        0.1503970  0.0345847   4.349 1.57e-05 ***
job_simpdata scientist       0.2149909  0.0298947   7.192 1.63e-12 ***
job_simpdirector             0.2853313  0.1103915   2.585 0.009945 ** 
job_simpmanager              0.0802874  0.0596228   1.347 0.178543    
job_simpmle                  0.4384980  0.0651187   6.734 3.42e-11 ***
job_simpOther tech jobs      0.1182113  0.0320214   3.692 0.000240 ***
age:job_simpdata engineer   -0.0273146  0.0866756  -0.315 0.752751    
age:job_simpdata scientist  -0.0689305  0.0709916  -0.971 0.331896    
age:job_simpdirector         0.2250479  0.1970560   1.142 0.253819    
age:job_simpmanager         -0.0240863  0.1392109  -0.173 0.862685    
age:job_simpmle             -0.5005824  0.1452565  -3.446 0.000602 ***
age:job_simpOther tech jobs -0.1004335  0.0743527  -1.351 0.177200    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1321 on 708 degrees of freedom
Multiple R-squared:  0.3596,	Adjusted R-squared:  0.3388 
F-statistic: 17.28 on 23 and 708 DF,  p-value: < 2.2e-16
```

b) Interaction model 2

avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark + aws + excel + job_simp + age:python

```
> summary(slmModel2_interaction)

Call:
lm(formula = avg_salary ~ . + age:python, data = sal_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.44122 -0.08244 -0.01647  0.06803  0.56077 

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              0.2315399  0.0212088  10.917  < 2e-16 ***
Sizemedium              -0.0246933  0.0138921  -1.778   0.0759 .  
Sizesmall               -0.0047601  0.0228819  -0.208   0.8353    
Revenuemedium-bus       -0.0001606  0.0159320  -0.010   0.9920    
Revenuemicro-bus        -0.0281272  0.0508718  -0.553   0.5805    
Revenuesmall-bus         0.0591622  0.0304785   1.941   0.0526 .  
age                     -0.0719749  0.0316919  -2.271   0.0234 *  
python_yn                0.0201483  0.0203466   0.990   0.3224    
R_yn                    -0.0039232  0.0956917  -0.041   0.9673    
spark                    0.0107920  0.0138897   0.777   0.4374    
aws                      0.0185221  0.0129299   1.432   0.1524    
excel                    0.0076473  0.0101671   0.752   0.4522    
job_simpdata engineer    0.1388907  0.0200838   6.916 1.04e-11 ***
job_simpdata scientist   0.1920535  0.0170470  11.266  < 2e-16 ***
job_simpdirector         0.4219073  0.0387897  10.877  < 2e-16 ***
job_simpmanager          0.0746065  0.0314962   2.369   0.0181 *  
job_simpmle              0.2420818  0.0327008   7.403 3.76e-13 ***
job_simpOther tech jobs  0.0819339  0.0172582   4.748 2.49e-06 ***
pythonYes                       NA         NA      NA       NA    
age:pythonYes            0.0500685  0.0417686   1.199   0.2310    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1329 on 713 degrees of freedom
Multiple R-squared:  0.3466,	Adjusted R-squared:  0.3301 
F-statistic: 21.01 on 18 and 713 DF,  p-value: < 2.2e-16
```

2. Addition model

avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark + aws + excel + job_simp

```
> summary(slmModel_addition)

Call:
lm(formula = avg_salary ~ ., data = sal_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.44954 -0.08291 -0.01758  0.06421  0.55686 

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              0.2223706  0.0197874  11.238  < 2e-16 ***
Sizemedium              -0.0258015  0.0138656  -1.861 0.063179 .  
Sizesmall               -0.0063300  0.0228514  -0.277 0.781856    
Revenuemedium-bus        0.0001656  0.0159345   0.010 0.991710    
Revenuemicro-bus        -0.0291032  0.0508808  -0.572 0.567510    
Revenuesmall-bus         0.0609059  0.0304530   2.000 0.045879 *  
age                     -0.0463317  0.0233898  -1.981 0.047991 *  
python_yn                0.0398789  0.0119644   3.333 0.000903 ***
R_yn                    -0.0086880  0.0956384  -0.091 0.927644    
spark                    0.0094428  0.0138482   0.682 0.495535    
aws                      0.0172365  0.0128893   1.337 0.181559    
excel                    0.0077495  0.0101699   0.762 0.446310    
job_simpdata engineer    0.1388583  0.0200899   6.912 1.06e-11 ***
job_simpdata scientist   0.1918565  0.0170514  11.252  < 2e-16 ***
job_simpdirector         0.4222977  0.0388002  10.884  < 2e-16 ***
job_simpmanager          0.0747120  0.0315057   2.371 0.017986 *  
job_simpmle              0.2431506  0.0326986   7.436 2.98e-13 ***
job_simpOther tech jobs  0.0806056  0.0172279   4.679 3.45e-06 ***
pythonYes                       NA         NA      NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.133 on 714 degrees of freedom
Multiple R-squared:  0.3453,	Adjusted R-squared:  0.3297 
F-statistic: 22.15 on 17 and 714 DF,  p-value: < 2.2e-16
```

3. Anova

a) Addition model versus Interaction model 1

```
> anova(slmModel_addition, slmModel1_interaction)
Analysis of Variance Table

Model 1: avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark +
    aws + excel + job_simp + python
Model 2: avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark +
    aws + excel + job_simp + age:job_simp
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1    714 12.624
2    708 12.349  6   0.27508 2.6285 0.01581 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is low (0.01581 < 0.05), so between the two, we choose the interaction model 1.

b) Addition model versus Interaction model 2

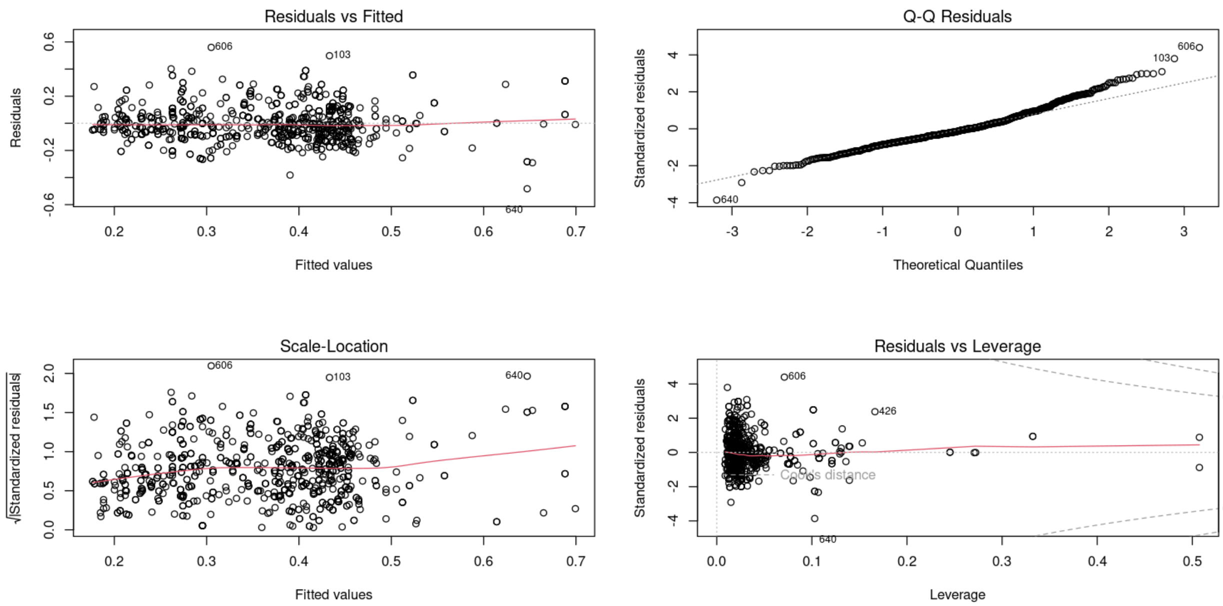
```
> anova(slmModel_addition, slmModel2_interaction)
Analysis of Variance Table

Model 1: avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark +
    aws + excel + job_simp + python
Model 2: avg_salary ~ Size + Revenue + age + python_yn + R_yn + spark +
    aws + excel + job_simp + python + age:python
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    714 12.624
2    713 12.598  1  0.025389 1.4369  0.231
```

The p-value is high (0.231 > 0.05), so between the two, we choose the addition model.

VI. CHECK ASSUMPTIONS

1. Check assumptions

```
par(mfrow = c(2, 2))
plot(slmModel1_interaction)
```



The diagnostic plots show residuals in four different ways:

- Residuals vs Fitted: is used to check the assumptions of linearity. If the residuals are spread equally around a horizontal line without distinct patterns (the red line is approximately horizontal at zero), that is a good indication of having a linear relationship.

- Normal Q-Q: is used to check the normality of the residuals assumption. If the majority of the residuals follow the straight dashed line, then the assumption is fulfilled.

- Scale-Location: is used to check the homoscedasticity of residuals (equal variance of residuals). If the residuals are spread randomly and they see a horizontal line with equally (randomly) spread points, then the assumption is fulfilled.

- Residuals vs Leverage: is used to identify any influential value in our dataset. Influential values are extreme values that might influence the regression results when included or excluded from the analysis. Look for cases outside of a dashed line.

a) Assumption 1: Linearity

```
#Linearity: the response can be written as a linear combination of the predictors.
#(With noise about this true linear relationship.)
plot(fitted(slmModel1_interaction), resid(slmModel1_interaction), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",
     main = "salary: Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```

For any fitted value, the residuals seem roughly centered at 0. The linearity assumption is not violated. We also use the Rainbow Test to check linearity.

```
> raintest(slmModel1_interaction)

        Rainbow test

data:  slmModel1_interaction
Rain = 0.97873, df1 = 366, df2 = 342, p-value = 0.5805
```

We see that p-value = 0.5805, which is greater than 0.05, so it does not violate the assumption of Linearity.

b) Assumption 2: Homoscedasticity

The Breusch-Pagan test is a statistical test used to diagnose heteroscedasticity, which is a violation of the assumption of homoscedasticity in a regression model.
$H0$: Homoscedasticity. The errors have constant variance about the true model.
$H1$: Heteroscedasticity. The errors have non-constant variance about the true model.

```
> bptest(slmModel1_interaction)

        studentized Breusch-Pagan test

data:  slmModel1_interaction
BP = 55.129, df = 16, p-value = 3.387e-06
```

P-value = 3.387x10^-6 < 0,05, so we reject the null of homoscedasticity (H0).
The constant variance assumption is violated.

c) Assumption 3: Independence of Errors

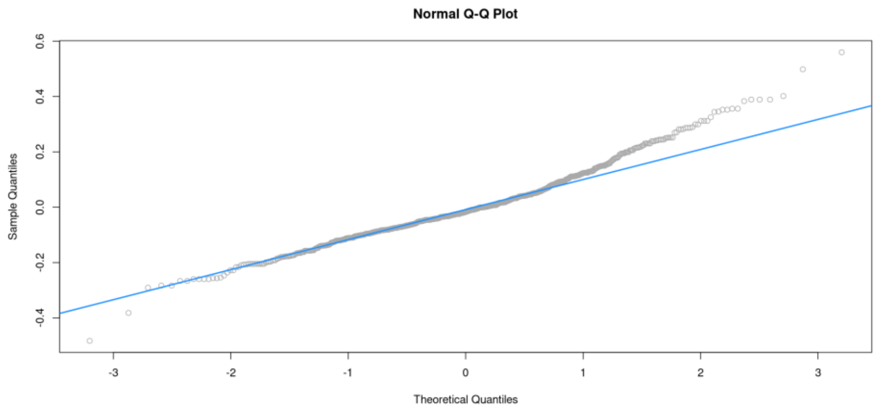
```
> durbinWatsonTest(slmModel1_interaction)
 lag Autocorrelation D-W Statistic p-value
   1     -0.03625889      2.069979   0.344
 Alternative hypothesis: rho != 0
```

A Durbin-Watson test is used to detect the presence of autocorrelation in the residuals of a regression.
H0 (null hypothesis): There is no correlation among the residuals.
H1 (alternative hypothesis): The residuals are autocorrelated.
Since this p-value = 0.344 is greater than 0.05, we can not reject the null hypothesis and conclude that the residuals in this regression model are not autocorrelated.
This assumption is valid.

d) Assumption 4: Normality: the distribution of the errors should follow a normal distribution

```
> qqnorm(resid(slmModel1_interaction), col = "darkgrey")
> qqline(resid(slmModel1_interaction), col = "dodgerblue", lwd = 2)
> shapiro.test(resid(slmModel1_interaction))

	Shapiro-Wilk normality test

data:  resid(slmModel1_interaction)
W = 0.97358, p-value = 3.144e-10
```

Shapiro-Wilk test
H0: data were sampled from a normal distribution
H1: data were not sampled from a normal distribution
Since p-value = 3.144x10^-10 < 0.05, so we reject H0
The normality assumption is violated



Also, the points that are far from the lines so the normality assumption is violated

e) Assumption 5: Multicollinearity

```
> vif(slmModel1_interaction)
                 Sizemedium                      Sizesmall
                   1.855575                       3.110810
           Revenuemedium-bus               Revenuemicro-bus
                   2.069085                       1.162889
            Revenuesmall-bus                            age
                   1.992401                       9.405520
                   python_yn                           R_yn
                   1.482048                       1.034049
                       spark                            aws
                   1.419815                       1.283045
                       excel          job_simpdata engineer
                   1.103808                       6.787658
      job_simpdata scientist               job_simpdirector
                   8.834683                       9.594594
             job_simpmanager                    job_simpmle
                   4.349186                       5.187946
     job_simpOther tech jobs      age:job_simpdata engineer
                   7.859053                       5.891956
  age:job_simpdata scientist           age:job_simpdirector
                  12.866088                      10.252854
         age:job_simpmanager                age:job_simpmle
                   4.584296                       5.393911
 age:job_simpOther tech jobs
                  11.741077
```

As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity.
Variables age, job_simp, age:job_simp have a value of VIF well above 5.
This assumption is violated.

2. *Transform variable to correct model*

Fitting a linear regression model using the *lm* function. The dependent variable *log(avg_salary + 1)* is the natural logarithm of the average salary plus 1. The addition of 1 is often used to handle cases where the salary includes zero values. This model predicts the natural logarithm of the average salary based on all other variables, and it includes an interaction term between age and job_simp.

Comparing the RMSE using the original and transformed response, we see that the log-transformed model simply does not fit better, with a larger average squared error (1.002475 > 0.1298836).

```
> sqrt(mean((sal_df$avg_salary - fitted(slmModel1_interaction)) ^ 2))
[1] 0.1298836
> sqrt(mean((sal_df$avg_salary - exp(fitted(slmModel1_interaction_log))) ^ 2))
[1] 1.002475
```

VII. USING AIC AND BIC TO BUILD THE BEST PROPOSED MODEL

1. *BIC*

a) Backward search

```
#backward using BIC
n = length(resid(slmModel))
slmModel_back_bic = step(slmModel, direction = "backward" , k = log(n))
#the procedure is the same, at each step, we look to improve BIC
#which R still labels AIC
coef(slmModel_back_bic)
```

```
Step:  AIC=-2894.62
avg_salary ~ python_yn + job_simp

            Df Sum of Sq    RSS     AIC
<none>                   13.057 -2894.6
- python_yn  1    0.2817 13.338 -2885.6
- job_simp   6    4.0459 17.103 -2736.6
```

b) Forward search

```
#forward search using BIC
slmModel_start = lm(avg_salary ~ 1, data = sal_df)
slmModel_forw_bic = step(
  slmModel_start,
  scope = avg_salary ~ Size + Revenue + age + python_yn + R_yn + aws + excel + job_simp + spark,
  direction = "forward", k = log(n))
```

```
Step:  AIC=-2894.62
avg_salary ~ job_simp + python_yn

          Df Sum of Sq    RSS     AIC
<none>                 13.057 -2894.6
+ age      1  0.090546 12.966 -2893.1
+ aws      1  0.080721 12.976 -2892.6
+ spark    1  0.028257 13.028 -2889.6
+ excel    1  0.015075 13.042 -2888.9
+ Size     2  0.130835 12.926 -2888.8
+ Revenue  3  0.237341 12.819 -2888.3
+ R_yn     1  0.000244 13.056 -2888.0
```

c) Stepwise search

```
#stepwise search using BIC
slmModel_both_bic = step(
  slmModel_start,
  scope = avg_salary ~ Size + Revenue + age + python_yn + R_yn + aws + excel + job_simp + spark,
  direction = "both", k = log(n)
)
```

```
Step:  AIC=-2894.62
avg_salary ~ job_simp + python_yn

            Df Sum of Sq    RSS     AIC
<none>                   13.057 -2894.6
+ age        1    0.0905 12.966 -2893.1
+ aws        1    0.0807 12.976 -2892.6
+ spark      1    0.0283 13.028 -2889.6
+ excel      1    0.0151 13.042 -2888.9
+ Size       2    0.1308 12.926 -2888.8
+ Revenue    3    0.2373 12.819 -2888.3
+ R_yn       1    0.0002 13.056 -2888.0
- python_yn  1    0.2817 13.338 -2885.6
- job_simp   6    4.0459 17.103 -2736.6
```

We see that, in all 3 backward searche, forward search and stepwise search, the results stop at AIC = -2894.62 with the model including job_simp and python_yn.

2. AIC

a) Backward search

```
## AIC
#backward using AIC

slmModel = lm(avg_salary ~ ., data = sal_df)
slmModel_back_aic = step(slmModel, direction = "backward")
#results:
#at the 1st step, the model has AIC = -2941.81
#each row: shows the effect of deleting each of the current predictors
#"-" signs at the beginning of each row: consider removing a predictor
#<none> keeping the current model
#the process continues until we reach the model avg_salary ~ Size + age + python_yn + aws + job_simp
#as removing any additional variable will not improve the AIC
#this model is stored in slmModel_back_aic
```

```
Step:  AIC=-2941.01
avg_salary ~ Size + Revenue + age + python_yn + aws + job_simp

            Df Sum of Sq    RSS     AIC
<none>                   12.642 -2941.0
- Revenue    3    0.1080 12.750 -2940.8
- aws        1    0.0392 12.681 -2940.7
- Size       2    0.0844 12.726 -2940.1
- age        1    0.0797 12.722 -2938.4
- python_yn  1    0.2305 12.873 -2929.8
- job_simp   6    3.9000 16.542 -2756.2
```

b) Forward search

```
#forward search using AIC
slmModel_start = lm(avg_salary ~ 1, data = sal_df)
slmModel_forw_aic = step(
  slmModel_start,
  scope = avg_salary ~ Size + Revenue + age + python_yn + R_yn + aws + excel + job_simp + spark,
  direction = "forward")
```

```
Step:  AIC=-2941.01
avg_salary ~ job_simp + python_yn + Revenue + age + Size + aws

        Df Sum of Sq    RSS     AIC
<none>                12.642 -2941.0
+ excel  1 0.0098752 12.632 -2939.6
+ spark  1 0.0079418 12.634 -2939.5
+ R_yn   1 0.0001595 12.642 -2939.0
```

c) Stepwise search

```
#stepwise search using AIC
#Start with avg_salary ~ 1 and search up to full model
slmModel_both_aic = step(
  slmModel_start,
  scope = avg_salary ~ Size + Revenue + age + python_yn + R_yn + aws + excel + job_simp + spark,
  direction = "both"
)
```

```
Step:  AIC=-2941.01
avg_salary ~ job_simp + python_yn + Revenue + age + Size + aws

            Df Sum of Sq    RSS     AIC
<none>                   12.642 -2941.0
- Revenue    3    0.1080 12.750 -2940.8
- aws        1    0.0392 12.681 -2940.7
- Size       2    0.0844 12.726 -2940.1
+ excel      1    0.0099 12.632 -2939.6
+ spark      1    0.0079 12.634 -2939.5
+ R_yn       1    0.0002 12.642 -2939.0
- age        1    0.0797 12.722 -2938.4
- python_yn  1    0.2305 12.873 -2929.8
- job_simp   6    3.9000 16.542 -2756.2
```

We see that, in all 3 backward searche, forward search and stepwise search, the results stop at AIC = -2941.01 with the model including job_simp and python_yn, Revenue, age, Size, aws.

3. Compare models

```
summary(slmModel)$adj.r.squared
summary(slmModel_back_aic)$adj.r.squared
summary(slmModel_back_bic)$adj.r.squared
```

Model	**Adjusted R-squared**
slmModel	0.32972
slmModel_back_aic	0.3315564
slmModel_back_bic	0.3163078

slmModel_back_aic is the best model compared to the other two models.

IX. CONCLUSION

In this project, we aimed to address the critical aspect of employee average salary, specifically within data science and technology jobs. The focus was developing an analytical model to predict salaries for data scientist roles and identifying key features contributing to salary variations.

In conclusion, this project successfully addressed the gap in research related to predictive modeling for data science salaries. The developed model serves as a valuable tool for job seekers and offers strategic insights for organizations. By understanding the key features influencing salaries, stakeholders in the technology job market can make informed decisions that contribute to career growth, organizational success, and overall job satisfaction. The findings from this research provide a foundation for future explorations into the dynamic landscape of salaries in the technology sector.

X. CONTRIBUTION

No.	Name	Student ID	Contribution
1	Vũ Thị Quế Anh	20070900	25%
2	Bùi Ngọc Anh	20070890	25%
3	Nguyễn Thu Hoài	20070931	25%
4	Dương Diệu Linh	20070944	25%