Final Project: Data Science Culmination Project

Josh Miller & Elijah Hill

Project Proposal: Week of November 13-17th

Project Due: Our Final Exam Time.

Assignment Description

This is it! This is the culmination of all your work in both of the data science courses! The parameters of this project are very general because I want to give you the chance to explore your interests and be creative. Generally, your project will go through the entire data science process. The project will involve a formal written document as well as a presentation. The written document will be worth 2/3rd's of the final grade and the presentation will be worth the other 1/3rd.

Here are the sections on how the project will be evaluated (A rubric will be released later with more specific parameters).

- Questions and Goals: The questions you wish to answer and the goals of the project. You should have multiple questions that you answer in the project. Not all of them need to be questions that require modeling to answer, but some of them need to be.
- Data Acquisition: The project describes how the data was obtained and gives substantial backround on the data.
- Data Preprocessing: Throughout the project, the proper preprocessing techniques (variable transformations, reshaping data, etc.) are utilized.
- Exploratory Data Analysis: Proper exploratory plots and summarizes are utilized to describe the data and showcase certain interesting aspects of the data that you will explore later in the project.
- Modeling and Analysis: This is a large portion of the project! You work toward answer the questions/goals you stated at the beginning. Your project needs to include at least 3 modeling techniques we discussed in class. You will fit, tune, and compare the methods. You will discuss the results and why they make sense in context. This section can include

a wide range of modeling techniques. Your proposal should be focused on describing what you want to do in this section.

- Data Product: You present your data, models, and conclusions in a professional manner. This could include an interactive data product.

Place Work Below!!

```r
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.4      v purrr   1.0.2
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.3.0      v stringr 1.5.1
v readr   2.1.3      v forcats 0.5.2

Warning: package 'ggplot2' was built under R version 4.2.3

Warning: package 'tidyr' was built under R version 4.2.3

Warning: package 'purrr' was built under R version 4.2.3

Warning: package 'stringr' was built under R version 4.2.3

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
styled <-
  theme_bw() +
  theme(
    plot.title = element_text(face = "bold", size = 12),
    legend.background = element_rect(
      fill = "white",
      linewidth = 4,
      colour = "white"
    ),
    axis.ticks = element_line(colour = "grey70", linewidth = 0.2),
    panel.grid.major = element_line(colour = "grey70", linewidth = 0.2),
    panel.grid.minor = element_blank()
  )
```

```r
library("tidymodels") ; theme_set(styled)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.0.0 --

v broom        1.0.1     v rsample      1.1.0
v dials        1.0.0     v tune         1.0.1
v infer        1.0.3     v workflows    1.1.0
v modeldata    1.0.1     v workflowsets 1.0.0
v parsnip      1.0.2     v yardstick    1.1.0
v recipes      1.0.2

-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Use tidymodels_prefer() to resolve common conflicts.
```

```r
library("janitor")
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library("olsrr")
```

```
Attaching package: 'olsrr'

The following object is masked from 'package:datasets':

    rivers
```

```r
library("doParallel")
```

```
Loading required package: foreach


Attaching package: 'foreach'

The following objects are masked from 'package:purrr':

    accumulate, when

Loading required package: iterators

Loading required package: parallel
```

```r
library("dplyr")
library("kernlab")
```

```
Warning: package 'kernlab' was built under R version 4.2.2


Attaching package: 'kernlab'

The following object is masked from 'package:scales':

    alpha

The following object is masked from 'package:purrr':

    cross

The following object is masked from 'package:ggplot2':

    alpha
```

```r
library("rpart.plot")
```

```
Warning: package 'rpart.plot' was built under R version 4.2.3
```

```
Loading required package: rpart
```

```
Attaching package: 'rpart'
```

```
The following object is masked from 'package:dials':

    prune
```

```r
library("glmnet")
```

```
Loading required package: Matrix
```

```
Attaching package: 'Matrix'
```

```
The following objects are masked from 'package:tidyr':

    expand, pack, unpack
```

```
Loaded glmnet 4.1-4
```

```r
library("GGally")
```

```
Warning: package 'GGally' was built under R version 4.2.3
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```r
library("cowplot")
```

```
Warning: package 'cowplot' was built under R version 4.2.3
```

```r
library("jtools")
```

```
Warning: package 'jtools' was built under R version 4.2.3


Attaching package: 'jtools'

The following object is masked from 'package:yardstick':

    get_weights
```

```r
library("caret")
```

```
Warning: package 'caret' was built under R version 4.2.3

Loading required package: lattice


Attaching package: 'caret'

The following objects are masked from 'package:yardstick':

    precision, recall, sensitivity, specificity

The following object is masked from 'package:purrr':

    lift
```

```r
all_cores <- parallel::detectCores(logical = FALSE)
cl <- makePSOCKcluster(all_cores)
registerDoParallel(cl)
```

Introduction:

For our Final Project, the dataset we decided to use was titled Salary by Job Title and Country. We found the dataset from Kaggle.com.

https://www.kaggle.com/datasets/amirmahdiabbootalebi/salary-by-job-title-and-country/data

The dataset creator sourced this data from reputable employment websites and surveys, leaving out names and companies to ensure privacy for both parties.

```
Salary <- read_csv("Salary.csv")
```

```
Rows: 6684 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (4): Gender, Job Title, Country, Race
dbl (5): Age, Education Level, Years of Experience, Salary, Senior

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

There are 9 variables in the data, with 6684 observations. The variables are as follows: Age, Gender, Education Level, Job Title, Years of Experience, Salary, Country, Race, and Senior. Education level is encoded from 0-3, 0 meaning the employee has a high school diploma as their highest level of education, 1 meaning that they have a Bachelor's degree, 2 meaning they have a Master's, and 3 meaning they have a Doctorate's. The senior variable is a binary value indicating whether or not they have a senior-level position. Salary has been converted into USD for all countries for the sake of being on the same scale.

```
head(Salary)
```

```
# A tibble: 6 x 9
    Age Gender `Education Level` `Job Title` Years~1 Salary Country Race  Senior
  <dbl> <chr>              <dbl> <chr>         <dbl>  <dbl> <chr>   <chr>  <dbl>
1    32 Male                   1 Software E~       5  90000 UK      White      0
2    28 Female                 2 Data Analy~       3  65000 USA     Hisp~      0
3    45 Male                   3 Manager          15 150000 Canada  White      1
4    36 Female                 1 Sales Asso~       7  60000 USA     Hisp~      0
5    52 Male                   2 Director         20 200000 USA     Asian      0
6    29 Male                   1 Marketing ~       2  55000 USA     Hisp~      0
# ... with abbreviated variable name 1: `Years of Experience`
```

Questions and Goals:

Our main question we wanted to answer was "Can we accurately predict the salary of a job given the predictors in this data set", those being Age, Senior, Country, Race, Job Title, Gender, and Education Level. We also wanted to explore the roles each of the predictors play in determining Salary. Some secondary questions we asked to determine this during our EDA were: "Is one gender more often lower-paid than another?", "Does an increase in age usually lead to an increase in salary?", "How big a difference does a job being a senior position make on average to Salary?", and more to go along with that: "Are older people more likely to be the ones occupying senior positions?". Whether or not the Education Level or Country of the job seems to give access to a higher salary were also questions we asked and found answers to.

Preprocessing:

For preprocessing, we quickly found 2 issues: First, we realized that certain job titles only appear once in the entire data set, one of the most notable being CEO. While this had one of the largest values for salary in the entire dataset, we realized that this would not only skew our EDA but would also cause problems for our testing and training splits later on. Therefore, we decided to drop them.

We then found an issue with values that were likely misreported within the dataset. Upon analyzing the bottom-most values for annual salary in the dataset, we found multiple employees reported only making 3 figures with jobs that in every other case paid well above that, such as Software Engineer Manager. We could be making a large assumption here that this was a full-time position being paid a yearly salary, but even if these values were correctly recorded, it would still be inconsistent with the rest of the dataset and cause a skew in the lowest-paying jobs.

```r
##PREPROCESSING

#removing any job only included once
Salary_cleaned <- Salary %>%
  group_by(`Job Title`) %>%
  mutate(
    count = n()
  ) %>%
  filter(count > 1) %>%
  ungroup() %>%
  select(-count)
#dropping probable mistaken entries (reported less than 1k salaries)
```

```
Salary_cleaned <- Salary_cleaned %>%
  arrange(Salary) %>%
  filter(!row_number() %in% c(1,2,3,4))
```

EDA:

Exploring Gender:

We wanted to explore if Females still earned less than men on average, as they have historically, so we first looked at a general average of all salaries of men versus those of women.

```
#EDA (Gender)

##GENDER DIFFERENCES
Salary_cleaned_by_gender <- Salary_cleaned %>%
  group_by(Gender) %>%
    summarize(Mean = mean(Salary, na.rm = TRUE))
#On average, women earn less than men

Salary_cleaned_by_gender
```

```
# A tibble: 2 x 2
  Gender    Mean
  <chr>    <dbl>
1 Female 107981.
2 Male   121503.
```

This table shows a sizable difference (about 19000) in the average salary of a male over one of a female, supporting our initial theory. We then split up the data to more deeply delve into the differences in pay between the two Genders.

```
#Splitting salary into male and female
salary_male <- Salary_cleaned %>%
  group_by(Gender) %>%
    filter(Gender == "Male")

salary_female <- Salary_cleaned %>%
  group_by(Gender) %>%
    filter(Gender == "Female")
```

After splitting the data, we tried making four plots showing the top 15 highest-salary jobs and the bottom 15 lowest-salary jobs for comparison.

```
# #comparing highest/lowest earning male/female jobs
#  top_male_salaries <- salary_male %>%
#  arrange(desc(Salary)) %>%
#    slice(1:15)
#
#  #ignoring mistaken entries (1 and 2 row)
#  bottom_male_salaries <- salary_male %>%
#  arrange(Salary) %>%
#    slice(3:17)
#
#  top_female_salaries <- salary_female %>%
#  arrange(desc(Salary)) %>%
#  slice(1:15)
#
# #ignoring mistaken entries (1 and 2 row)
#  bottom_female_salaries <- salary_female %>%
#  arrange(Salary) %>%
#    slice(3:17)

#plots
#tms_plot <- ggplot(top_male_salaries, aes(x = Salary)) +       #geom_bar(fill = "blue") +
#theme_light()

#bms_plot <- ggplot(bottom_male_salaries, aes(x = Salary)) +
# geom_bar(fill = "turquoise2") +
# theme_dark()

#the above plots don't look good...
#they are mostly the same jobs

#trying again but with...
```

the above plots don't look good… they are mostly the same jobs trying again but with averaging jobs with the same title together.

Upon making the first few plots, we realized that the above plots did not look good as they were mostly showing the same job titles' salaries repeated multiple times. We remade the graphs but this time combined the job titles to eliminate repeated Job Titles. First, we made new tables to use with a new Average_Salary column for each job title, then eliminated other columns and rows besides unique Average_Salaries and Job Titles since those were what we were focusing on.

```r
#averaging jobs with the same title together
salary_male_unique <- salary_male %>%
  group_by(`Job Title`) %>%
  mutate(Average_Salary = mean(Salary)) %>%
  distinct(Average_Salary)

salary_female_unique <- salary_female %>%
  group_by(`Job Title`) %>%
  mutate(Average_Salary = mean(Salary)) %>%
  distinct(Average_Salary)

salary_male_unique
```

```
# A tibble: 61 x 2
# Groups:   Job Title [61]
   `Job Title`                      Average_Salary
   <chr>                                     <dbl>
 1 Sales Associate                          33515.
 2 Delivery Driver                          28000
 3 Sales Representative                     46444.
 4 Digital Marketing Manager                75968.
 5 HR Generalist                            72776.
 6 HR Coordinator                           34667.
 7 Accountant                               53750
 8 Software Developer                       68011.
 9 Business Development Associate           38333.
10 Operations Analyst                       69167.
# ... with 51 more rows
```

```r
salary_female_unique
```

```
# A tibble: 67 x 2
# Groups:   Job Title [67]
   `Job Title`                        Average_Salary
   <chr>                                       <dbl>
 1 Sales Associate                            28207.
 2 Sales Representative                       35833.
 3 Receptionist                               25000
```

```
 4 HR Coordinator                         41062.
 5 Customer Service Representative        33333.
 6 HR Generalist                          48855.
 7 Juniour HR Coordinator                 32000
 8 Marketing Analyst                      63083.
 9 Business Development Associate         42500
10 Operations Manager                     95200
# ... with 57 more rows
```

```r
#comparing highest/lowest earning male/female jobs
top_male_salaries_unique <-
  salary_male_unique %>%
  ungroup() %>% arrange(desc(Average_Salary)) %>% slice(1:10)

bottom_male_salaries_unique <-
  salary_male_unique %>%
  ungroup() %>%
  arrange(Average_Salary) %>%
  slice(1:10)

top_female_salaries_unique <-
  salary_female_unique %>%
  ungroup() %>%
  arrange(desc(Average_Salary)) %>% slice(1:10)

bottom_female_salaries_unique <-
  salary_female_unique %>%
  ungroup %>%
  arrange(Average_Salary) %>%
    slice(1:10)

top_male_salaries_unique
```

```
# A tibble: 10 x 2
   `Job Title`               Average_Salary
   <chr>                              <dbl>
 1 Director of Data Science         207742.
 2 Marketing Director               189900
 3 Director of Engineering          180000
 4 Software Engineer Manager        173385.
 5 Project Engineer                 173344.
 6 Director of Operations           171667.
```

```
 7 Director of Finance              170000
 8 Research Director                165870.
 9 Data Scientist                   165062.
10 Director of Marketing            160641.
```

```
bottom_male_salaries_unique
```

```
# A tibble: 10 x 2
   `Job Title`                   Average_Salary
   <chr>                                  <dbl>
 1 Delivery Driver                        28000
 2 Sales Associate                        33515.
 3 HR Coordinator                         34667.
 4 Business Operations Analyst            35000
 5 Business Development Associate         38333.
 6 Juniour HR Generalist                  43000
 7 Sales Representative                   46444.
 8 Sales Executive                        47083.
 9 Graphic Designer                       51667.
10 Accountant                             53750
```

```
top_female_salaries_unique
```

```
# A tibble: 10 x 2
   `Job Title`                Average_Salary
   <chr>                               <dbl>
 1 Director of Data Science          200769.
 2 Director of Human Resources       187500
 3 Director of Finance               180000
 4 Director of Operations            174000
 5 Product Manager                   172476.
 6 Software Engineer Manager         171793.
 7 Data Scientist                    162667.
 8 Marketing Director                162667.
 9 Data Engineer                     160000
10 Research Director                 159310.
```

```
bottom_female_salaries_unique
```

```
# A tibble: 10 x 2
```

```
   `Job Title`                     Average_Salary
   <chr>                                    <dbl>
 1 Receptionist                           25000
 2 Sales Associate                        28207.
 3 Juniour HR Coordinator                 32000
 4 Customer Service Representative        33333.
 5 Sales Representative                   35833.
 6 HR Coordinator                         41062.
 7 Sales Executive                        41154.
 8 Business Development Associate         42500
 9 Copywriter                             42500
10 Juniour HR Generalist                  43000
```

We made the plots again, making sure to standardize the x-axis values to more clearly show any differences in pay. We made male plots blue, and female red, top salary plots have a light theme, and bottom salary plots use the dark theme to differentiate and help show the comparisons we were looking for.

```r
tms_plot <- ggplot(top_male_salaries_unique, aes(x = Average_Salary)) +
    geom_histogram(fill = "blue") +
    labs(y = "Job Count", x = "Average Salary (Male)") +
    xlim(150000, 250000)
```

```r
bms_plot <- ggplot(bottom_male_salaries_unique, aes(x = Average_Salary)) +
    geom_histogram(fill = "turquoise2", bins = 40) +
    labs(y = "Job Count", x = "Average Salary (Male)") +
    xlim(24000, 58000)+ ylim(0, 3) + theme_dark()
```

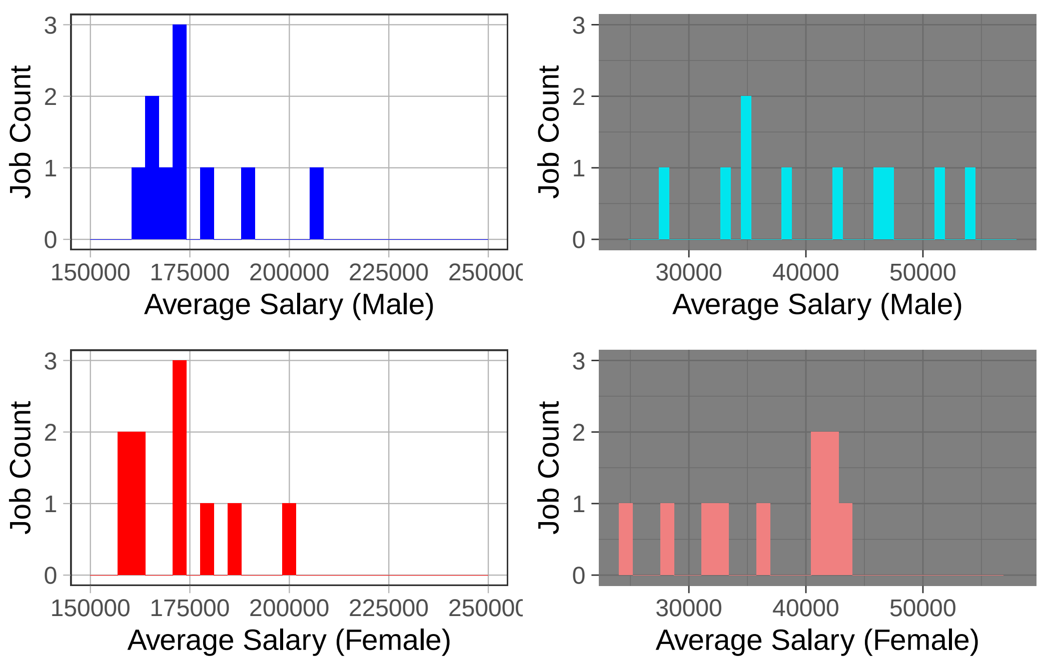
```r
tfs_plot <- ggplot(top_female_salaries_unique, aes(x = Average_Salary)) +
    geom_histogram(fill = "red") +
    labs(y = "Job Count", x = "Average Salary (Female)") +
    xlim(150000, 250000)
```

```r
bfs_plot <- ggplot(bottom_female_salaries_unique, aes(x = Average_Salary)) +
    geom_histogram(fill = "lightcoral") +
    labs(y = "Job Count", x = "Average Salary (Female)") +
    xlim(24000, 58000) + ylim(0, 3) + theme_dark()
```

We used the “cowplot” package to easily combine all four plots into one graphic for a more complete visual comparison of gender salary differences on the poles of the data.

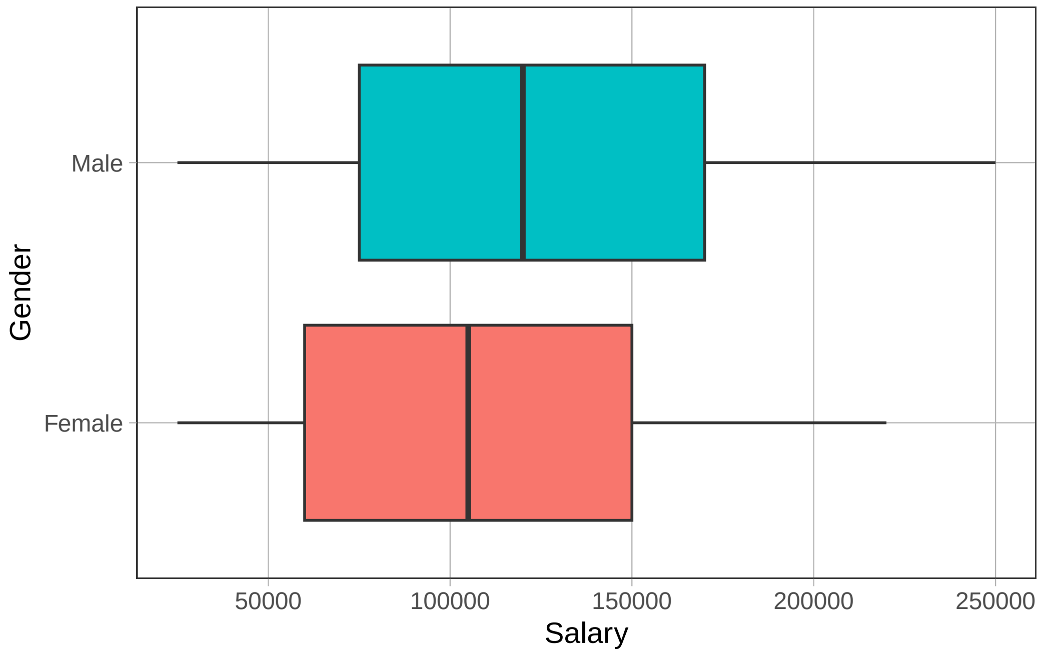
```r
plot_grid(tms_plot, bms_plot, tfs_plot, bfs_plot, nrow = 2, ncol = 2)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 1 rows containing missing values (`geom_bar()`).

Warning: Removed 2 rows containing missing values (`geom_bar()`).

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 1 rows containing missing values (`geom_bar()`).

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



After doing this, we realized that we could do almost the same thing, but in a broader sense (as well as faster), by just using a box plot.

```r
#comparing all salaries
ggplot(Salary_cleaned, aes(x = Salary, y = Gender)) +
    geom_boxplot(aes(
        fill = as.factor(`Gender`))) +
    scale_color_manual(values = c("blue", "red")) +
    theme(legend.position = "none")
```



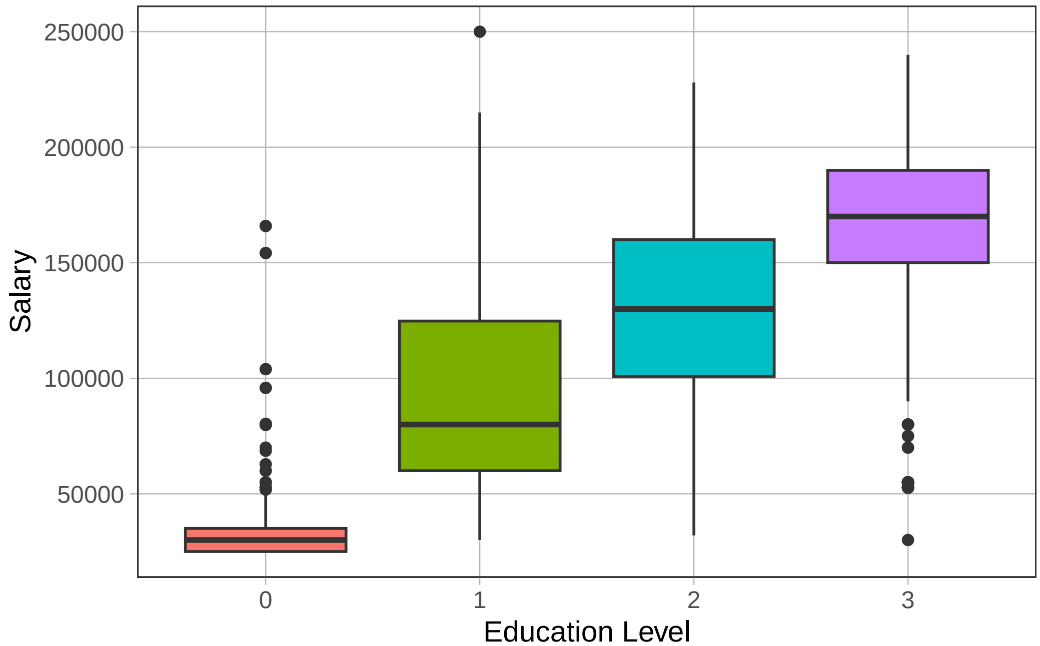
The box plots, as well as the four plots prior, all point to what we had guessed which was that males do indeed earn higher salaries than females on average.

Exploring Education Level:

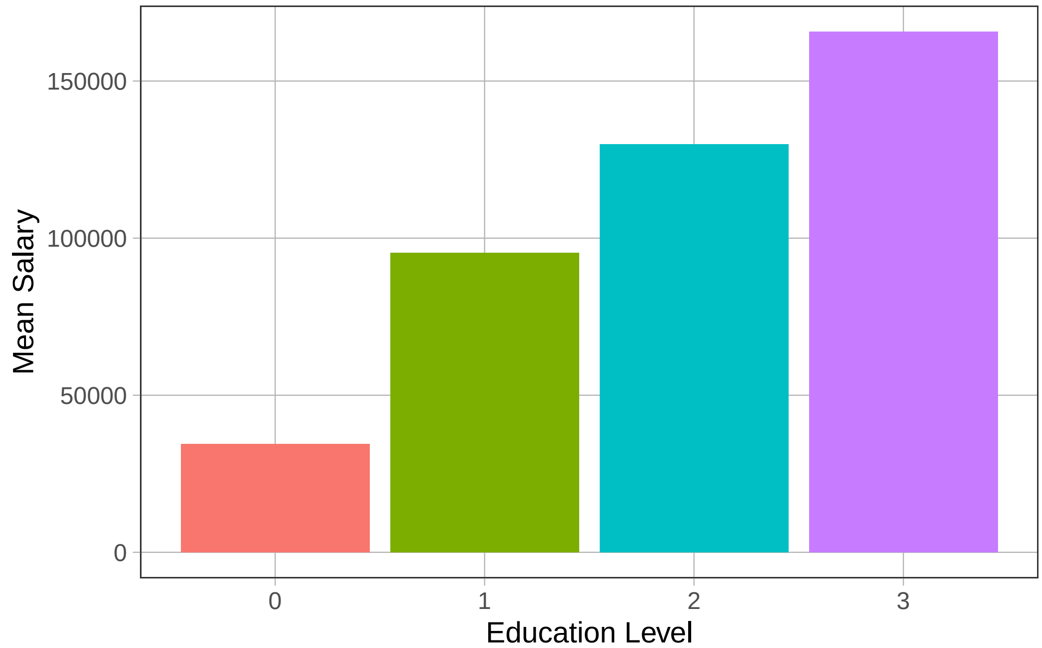
Next, we examined the role education level played on salary amount. This time, we started with the general plot comparing all salaries grouped by Education Level, then moved on to showing the average salary of each Education Level after that.

```r
#EDA (Education Level)

ggplot(Salary_cleaned, aes(y = Salary,
                           x = as.factor(`Education Level`),
                           fill = as.factor(`Education Level`
                                            )))+
```

```
    scale_color_manual(values = c("red", "green", "yellow", "darkorchid3")) +
    labs(x = "Education Level") +
    geom_boxplot() +
    theme(legend.position = "none")
```




```
salary_by_ed_lvl <- Salary_cleaned %>%
  group_by(`Education Level`) %>%
  summarize(Mean = mean(Salary))

ggplot(salary_by_ed_lvl, aes(x = `Education Level`,
                             y = Mean,
                             fill = as.factor(
                                 `Education Level`
                                 ))) +
    scale_color_manual(values = c("red", "green", "yellow", "darkorchid3" )) +
    labs(y = "Mean Salary") +
    geom_col() +
    theme(legend.position = "none")
```

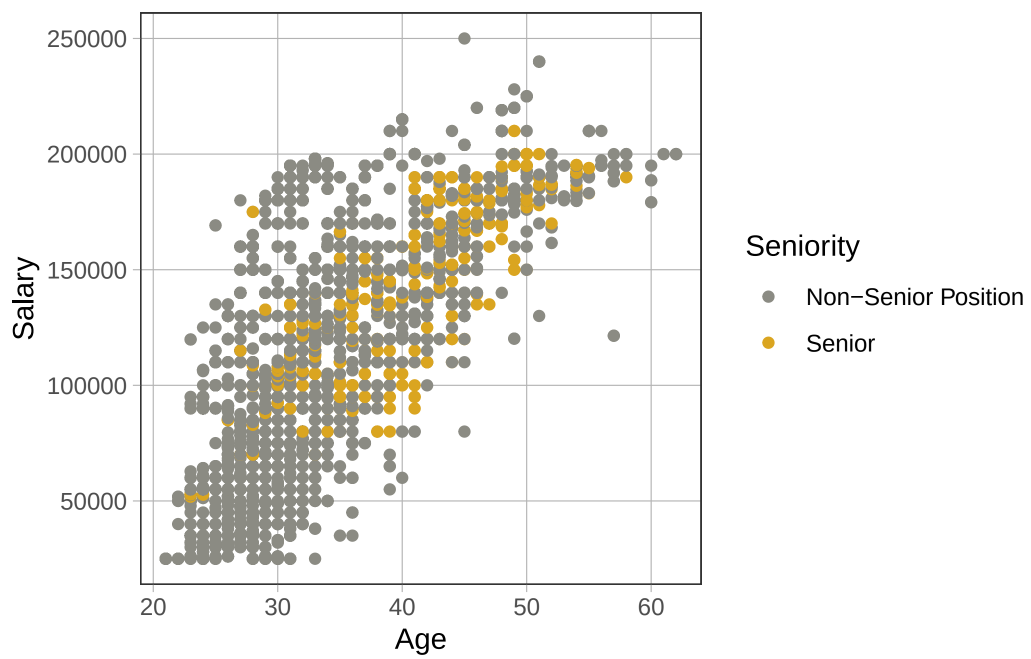


We were pleased to see not only that the data seemed to indicate that going to college is indeed still worth it, but that the data was nice and linear as well for both the raw and the average salary by education level comparisons.

Exploring Age and Seniority:

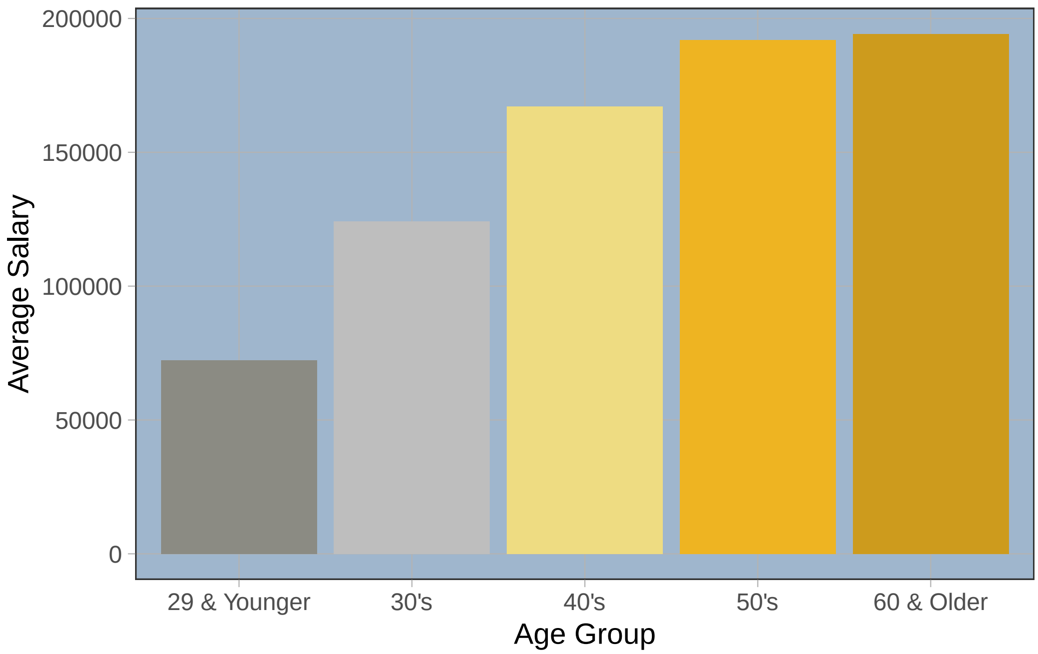
Age and Seniority were two predictors we were especially excited to look at, and we had high expectations on the strength of the correlation between them and the salaries those of high age and in senior positions would hold. Once again, we showed a general plot using the raw salary data when compared with Age, this time using whether or not the job holder had the Senior status to determine the color of the plot point. After getting a nice-looking scatter plot from that (and being very happy with the color palette), we could see that there was some positive correlation between the age of a person, whether or not they would be in a senior position, and their salary. To get a slightly different perspective, we grouped the ages by decade and compared each Age Group's average salary to each other, and were once again satisfied to see a seemingly linear relationship between Age and Salary.

```
#EDA (Age/Seniority)
ggplot(Salary_cleaned, aes(y = Salary, x = Age, color = as.factor(Senior))) +
    scale_color_manual(values = c("ivory4", "goldenrod"),
                       labels = c("Non-Senior Position",
                                  "Senior")) +
    labs(color = "Seniority") + geom_point()
```




```r
salary_by_age <- Salary_cleaned %>%
  mutate(Age_Group = case_when(
    Age < 30 ~ "29 & Younger",
    Age < 40 & Age >= 30 ~ "30's",
    Age < 50 & Age >= 40 ~ "40's",
    Age < 60 & Age >= 50 ~ "50's",
    Age >= 60 ~ "60 & Older"
  )) %>%
  group_by(Age_Group) %>%
  summarize(Average_Salary = mean(Salary))

ggplot(salary_by_age, aes(x = Age_Group,
                          y = Average_Salary,
                          fill = as.factor(Age_Group)),) +
    scale_fill_manual(values = c("ivory4","grey",
                                 "lightgoldenrod2",
                                 "goldenrod2",
                                 "goldenrod3")) +
    geom_col() +
    theme(legend.position = "none",
```

```
                        panel.background = element_rect(fill = "slategray3")) +
    labs(x = "Age Group", y = "Average Salary")
```



Exploring Country and Race:

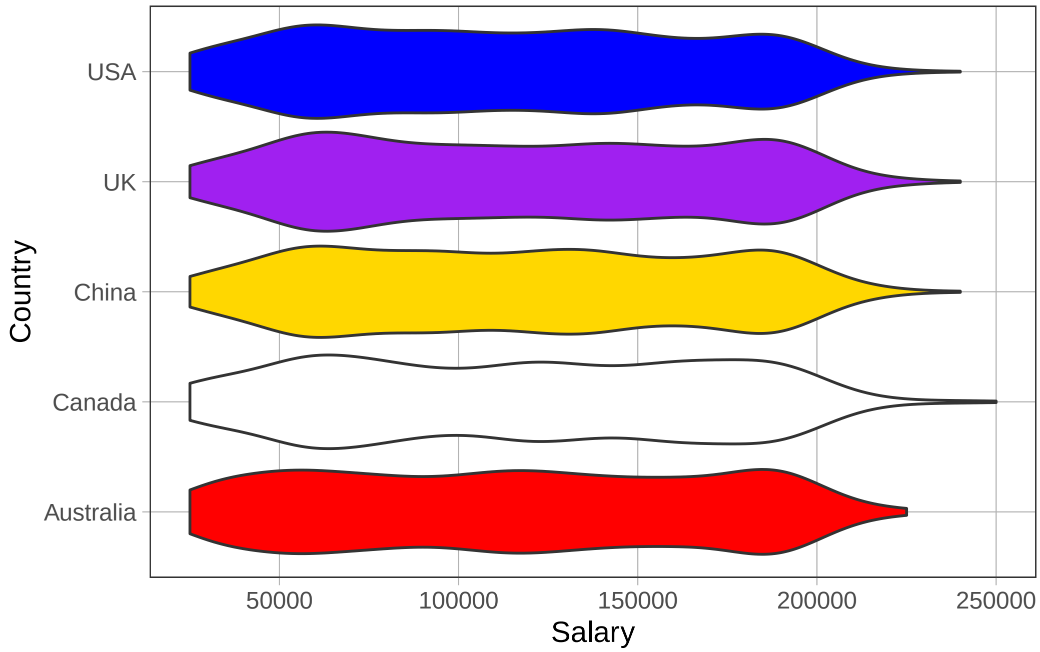
The first thing we did to look at Country and Race was to use multilevel grouping to get a better understanding of the demographics of the data. After noting the variety of Races in each Country, we proceeded to make a violin plot comparing the Salaries of those living in different Countries. That plot did not look great, so we reverted to using box plots for comparing Races' Salary earnings. The main takeaway we received from these two plots was that the Country and Race of a person do not seem to be significant factors in determining one's salary.

```
#EDA (Country/Race)
Salary_cleaned %>%
  group_by(Country, Race) %>%
  summarize(count = n())
```

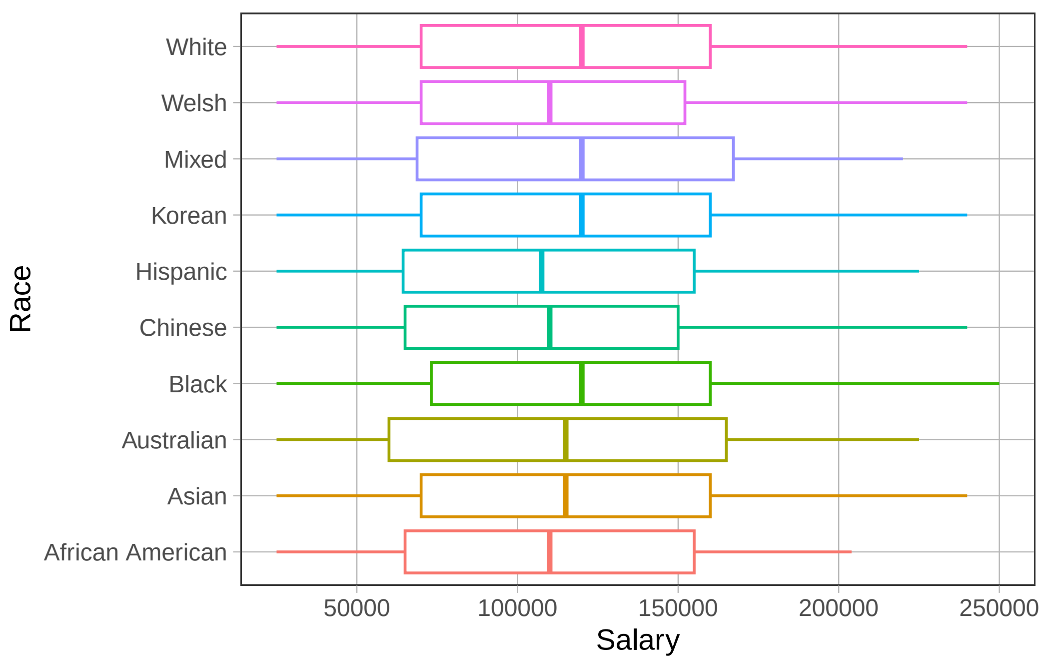
```
`summarise()` has grouped output by 'Country'. You can override using the
`.groups` argument.
```

```
# A tibble: 17 x 3
# Groups:   Country [5]
   Country   Race             count
   <chr>     <chr>            <int>
 1 Australia Asian              470
 2 Australia Australian         449
 3 Australia White              407
 4 Canada    Asian              452
 5 Canada    Black              428
 6 Canada    White              431
 7 China     Chinese            441
 8 China     Korean             454
 9 China     White              438
10 UK        Asian              328
11 UK        Mixed              329
12 UK        Welsh              330
13 UK        White              327
14 USA       African American   349
15 USA       Asian              330
16 USA       Hispanic           318
17 USA       White              346
```

```r
ggplot(Salary_cleaned, aes(x = Salary, y = Country, fill = as.factor(Country))) +
    scale_fill_manual(values = c("red", "white", "gold", "purple", "blue")) +
    geom_violin() +
    theme(legend.position = "none")
```




```r
ggplot(Salary_cleaned, aes(x = Salary, y = Race, color =  as.factor(Race))) +
    geom_boxplot() +
    theme(legend.position = "none")
```



Exploring Job Title:

When thinking of what to explore with Job Titles, we were at first a little unsure of what to compare, since there were so many unique Job Titles in the data. We ended up simply making a table of the top 10 highest-salary jobs and the “top 10” lowest-salary jobs.

```
#EDA (Job Title)
#hrm...
salary_by_job_title <- Salary_cleaned %>%
  group_by(`Job Title`) %>%
  mutate(Average_Salary = mean(Salary)) %>%
  distinct(Average_Salary)

top_jobs <- salary_by_job_title %>%
  ungroup() %>%
  arrange(desc(Average_Salary)) %>%
  slice(1:10)

worst_jobs <- salary_by_job_title %>%
  ungroup() %>%
  arrange(Average_Salary) %>%
  slice(1:10)

top_jobs
```

```
# A tibble: 10 x 2
   `Job Title`                   Average_Salary
   <chr>                                  <dbl>
 1 Director of Data Science             204561.
 2 Director of Human Resources          187500
 3 Marketing Director                   183615.
 4 Director of Engineering              180000
 5 Director of Finance                  175000
 6 Software Engineer Manager            172961.
 7 Director of Operations               172727.
 8 Project Engineer                     166064.
 9 Data Scientist                       164099.
10 Research Director                    163333.
```

```
worst_jobs
```

```
# A tibble: 10 x 2
```

```
   `Job Title`                        Average_Salary
   <chr>                                       <dbl>
 1 Receptionist                               25000
 2 Delivery Driver                            28000
 3 Sales Associate                            30736.
 4 Juniour HR Coordinator                     32000
 5 Customer Service Representative            33333.
 6 Business Operations Analyst                35000
 7 HR Coordinator                             38321.
 8 Business Development Associate             40714.
 9 Sales Representative                       41728.
10 Copywriter                                 42500
```

One interesting thing that we could see from these tables is that Job Titles with "Director" and "Engineer" are featured frequently in the higher end of the Salary data. This could either be an insight into the types of jobs that give high Salaries, or the types of jobs that the data was scraped from. Either way, the wide range of names meant that job titles were most likely going to be largely ineffective as a predictor for our models.

Modeling:

Now that we have gathered some insights about this data as well as having answered our minor questions from our exploratory analysis, we will use modeling to answer our main question.

Before we get into creating the models, we will split the salary dataset into a training and testing data frame, using a 90/10 proportion respectively.

```r
salary_split <- initial_split(Salary_cleaned, prop = 0.90)
training <-training(salary_split)
testing <- testing(salary_split)
```

To create a ridge regression model, we need to turn all of our datasets into numeric factors. We will get to the ridge regression model later. This is simply up here for rendering reasons.

```r
ridge_salary <- Salary_cleaned %>%
  transform(.,
                 Race = as.numeric(as.factor(Race)),
                 Country = as.numeric(as.factor(Country)),
                 `Job Title` = as.numeric(as.factor(`Job Title`)),
                 Gender = as.numeric(as.factor(Gender)))
ridge_split <- initial_split(ridge_salary, prop = .90)
```

```r
train <- training(ridge_split)
test <- testing(ridge_split)
```

Multiple Linear Regression:

Now that we split the data into training and testing, we will create our first model: a multiple linear regression model. A multiple linear regression model is simple, yet it can still give a good benchmark for comparisons to our other models.

```r
fit <- lm(Salary ~ ., data = training)
summary(fit)
```

```
Call:
lm(formula = Salary ~ ., data = training)

Residuals:
    Min      1Q  Median      3Q     Max 
-130031  -11632     -39   11627   64819 

Coefficients:
                                           Estimate Std. Error t value
(Intercept)                               29928.289  11802.594   2.536
Age                                         -14.080    129.725  -0.109
GenderMale                                  408.407    649.929   0.628
`Education Level`                          6555.230    605.413  10.828
`Job Title`Accountant                     -1241.538  14509.808  -0.086
`Job Title`Administrative Assistant      -37338.367  19474.323  -1.917
`Job Title`Back end Developer             29009.116  11346.786   2.557
`Job Title`Business Analyst               13137.112  12428.202   1.057
`Job Title`Business Development Associate -9708.276  14508.993  -0.669
`Job Title`Business Development Manager   26873.222  15093.180   1.780
`Job Title`Business Operations Analyst   -12172.564  25133.411  -0.484
`Job Title`Content Marketing Manager      21133.828  11597.082   1.822
`Job Title`Copywriter                     -9266.246  19474.657  -0.476
`Job Title`Customer Service Manager      -29261.380  25166.053  -1.163
`Job Title`Customer Service Representative -9908.242  15090.651  -0.657
`Job Title`Data Analyst                   55116.572  11309.428   4.874
`Job Title`Data Engineer                  27869.128  15929.931   1.749
`Job Title`Data Scientist                 54843.132  11337.019   4.838
`Job Title`Delivery Driver                -4430.745  15895.226  -0.279
```

```
`Job Title`Digital Marketing Manager        11720.800  11702.851   1.002
`Job Title`Digital Marketing Specialist      5068.344  12854.389   0.394
`Job Title`Director of Data Science         60171.042  11755.169   5.119
`Job Title`Director of Engineering          25574.474  19511.560   1.311
`Job Title`Director of Finance              19884.510  19499.300   1.020
`Job Title`Director of HR                    8781.104  11647.376   0.754
`Job Title`Director of Human Resources      20154.762  19522.319   1.032
`Job Title`Director of Marketing            24180.932  11560.182   2.092
`Job Title`Director of Operations           12476.124  13556.848   0.920
`Job Title`Engineer                          9539.483  25176.469   0.379
`Job Title`Event Coordinator               -32340.228  19461.607  -1.662
`Job Title`Financial Advisor                16927.053  15086.755   1.122
`Job Title`Financial Analyst                18141.979  11692.480   1.552
`Job Title`Financial Manager                44225.634  11424.797   3.871
`Job Title`Front end Developer              22475.597  11346.619   1.981
`Job Title`Front End Developer              18796.445  11997.842   1.567
`Job Title`Full Stack Engineer              35102.420  11331.680   3.098
`Job Title`Graphic Designer                  1020.743  12367.284   0.083
`Job Title`HR Coordinator                   -9158.079  12072.138  -0.759
`Job Title`HR Generalist                      633.249  11480.731   0.055
`Job Title`HR Manager                        3326.472  17178.843   0.194
`Job Title`Human Resources Coordinator      -9744.386  11736.003  -0.830
`Job Title`Human Resources Manager          15267.205  11410.124   1.338
`Job Title`IT Consultant                    27279.850  19485.582   1.400
`Job Title`IT Support Specialist            -3721.373  19474.892  -0.191
`Job Title`Juniour HR Coordinator           -5105.120  17178.421  -0.297
`Job Title`Juniour HR Generalist             2533.346  19479.390   0.130
`Job Title`Manager                          24358.106  19508.886   1.249
`Job Title`Marketing Analyst                 7856.637  11414.485   0.688
`Job Title`Marketing Coordinator             9518.554  11393.315   0.835
`Job Title`Marketing Director               64181.876  11660.830   5.504
`Job Title`Marketing Manager                18040.873  11333.788   1.592
`Job Title`Marketing Specialist              7083.117  13503.677   0.525
`Job Title`Operations Analyst               -7442.419  14084.533  -0.528
`Job Title`Operations Coordinator           21705.957  15896.798   1.365
`Job Title`Operations Manager               14198.842  11438.440   1.241
`Job Title`Product Designer                  6432.888  11554.360   0.557
`Job Title`Product Manager                  57184.910  11323.601   5.050
`Job Title`Product Marketing Manager        27833.851  11639.769   2.391
`Job Title`Project Coordinator               6257.842  15916.323   0.393
`Job Title`Project Engineer                 52603.498  11359.423   4.631
`Job Title`Project Manager                  24496.634  11936.545   2.052
`Job Title`Receptionist                     -6410.276  11686.008  -0.549
```

```
`Job Title`Recruiter                      -21951.130  17166.964  -1.279
`Job Title`Research Director               50314.961  11658.754   4.316
`Job Title`Research Scientist              43471.626  11507.443   3.778
`Job Title`Sales Associate                 -7900.317  11370.077  -0.695
`Job Title`Sales Director                  32680.199  11632.546   2.809
`Job Title`Sales Executive                 -1755.312  11971.399  -0.147
`Job Title`Sales Manager                   19714.850  11657.916   1.691
`Job Title`Sales Representative            -6732.375  11551.173  -0.583
`Job Title`Scientist                       16044.362  17202.574   0.933
`Job Title`Social Media Manager              758.593  12863.367   0.059
`Job Title`Social Media Specialist        -10199.809  25122.532  -0.406
`Job Title`Software Developer               3055.415  11371.820   0.269
`Job Title`Software Engineer               45585.683  11269.879   4.045
`Job Title`Software Engineer Manager       35912.978  11365.923   3.160
`Job Title`Training Specialist            -17175.178  19476.932  -0.882
`Job Title`UX Designer                     17972.537  15111.272   1.189
`Job Title`Web Developer                      -2.519  11433.755   0.000
`Years of Experience`                       5503.175    159.973  34.401
CountryCanada                                400.249   1133.572   0.353
CountryChina                                 -27.303   1461.210  -0.019
CountryUK                                    448.525   1240.060   0.362
CountryUSA                                  -356.912   1207.575  -0.296
RaceAsian                                   1985.676   1618.122   1.227
RaceAustralian                              2491.922   2091.627   1.191
RaceBlack                                    874.105   2092.007   0.418
RaceChinese                                  -73.100   2272.955  -0.032
RaceHispanic                                1811.532   1847.325   0.981
RaceKorean                                  2453.398   2265.309   1.083
RaceMixed                                   1573.856   2251.454   0.699
RaceWelsh                                   -904.009   2261.541  -0.400
RaceWhite                                   2200.761   1612.457   1.365
Senior                                    -12053.225   1286.107  -9.372
                                          Pr(>|t|)
(Intercept)                                0.01125 *
Age                                        0.91357
GenderMale                                 0.52977
`Education Level`                          < 2e-16 ***
`Job Title`Accountant                      0.93181
`Job Title`Administrative Assistant        0.05525 .
`Job Title`Back end Developer              0.01060 *
`Job Title`Business Analyst                0.29054
`Job Title`Business Development Associate  0.50344
`Job Title`Business Development Manager    0.07505 .
```

```
`Job Title`Business Operations Analyst      0.62818
`Job Title`Content Marketing Manager        0.06845 .
`Job Title`Copywriter                       0.63423
`Job Title`Customer Service Manager         0.24499
`Job Title`Customer Service Representative  0.51148
`Job Title`Data Analyst                    1.13e-06 ***
`Job Title`Data Engineer                    0.08026 .
`Job Title`Data Scientist                  1.35e-06 ***
`Job Title`Delivery Driver                  0.78045
`Job Title`Digital Marketing Manager        0.31661
`Job Title`Digital Marketing Specialist     0.69338
`Job Title`Director of Data Science        3.17e-07 ***
`Job Title`Director of Engineering          0.19000
`Job Title`Director of Finance              0.30789
`Job Title`Director of HR                   0.45093
`Job Title`Director of Human Resources      0.30193
`Job Title`Director of Marketing            0.03650 *
`Job Title`Director of Operations           0.35746
`Job Title`Engineer                         0.70477
`Job Title`Event Coordinator                0.09662 .
`Job Title`Financial Advisor                0.26192
`Job Title`Financial Analyst                0.12081
`Job Title`Financial Manager                0.00011 ***
`Job Title`Front end Developer              0.04766 *
`Job Title`Front End Developer              0.11725
`Job Title`Full Stack Engineer              0.00196 **
`Job Title`Graphic Designer                 0.93422
`Job Title`HR Coordinator                   0.44811
`Job Title`HR Generalist                    0.95601
`Job Title`HR Manager                       0.84647
`Job Title`Human Resources Coordinator      0.40640
`Job Title`Human Resources Manager          0.18094
`Job Title`IT Consultant                    0.16157
`Job Title`IT Support Specialist            0.84847
`Job Title`Juniour HR Coordinator           0.76634
`Job Title`Juniour HR Generalist            0.89653
`Job Title`Manager                          0.21187
`Job Title`Marketing Analyst                0.49129
`Job Title`Marketing Coordinator            0.40350
`Job Title`Marketing Director               3.87e-08 ***
`Job Title`Marketing Manager                0.11149
`Job Title`Marketing Specialist             0.59993
`Job Title`Operations Analyst               0.59723
```

```
`Job Title`Operations Coordinator          0.17217
`Job Title`Operations Manager              0.21453
`Job Title`Product Designer                0.57772
`Job Title`Product Manager                4.55e-07 ***
`Job Title`Product Marketing Manager       0.01682 *
`Job Title`Project Coordinator             0.69421
`Job Title`Project Engineer               3.72e-06 ***
`Job Title`Project Manager                 0.04019 *
`Job Title`Receptionist                    0.58334
`Job Title`Recruiter                       0.20106
`Job Title`Research Director              1.62e-05 ***
`Job Title`Research Scientist              0.00016 ***
`Job Title`Sales Associate                 0.48719
`Job Title`Sales Director                  0.00498 **
`Job Title`Sales Executive                 0.88343
`Job Title`Sales Manager                   0.09087 .
`Job Title`Sales Representative            0.56003
`Job Title`Scientist                       0.35103
`Job Title`Social Media Manager            0.95298
`Job Title`Social Media Specialist         0.68476
`Job Title`Software Developer              0.78818
`Job Title`Software Engineer              5.30e-05 ***
`Job Title`Software Engineer Manager       0.00159 **
`Job Title`Training Specialist             0.37791
`Job Title`UX Designer                     0.23435
`Job Title`Web Developer                   0.99982
`Years of Experience`                      < 2e-16 ***
CountryCanada                              0.72404
CountryChina                               0.98509
CountryUK                                  0.71759
CountryUSA                                 0.76758
RaceAsian                                  0.21982
RaceAustralian                             0.23355
RaceBlack                                  0.67609
RaceChinese                                0.97434
RaceHispanic                               0.32682
RaceKorean                                 0.27884
RaceMixed                                  0.48455
RaceWelsh                                  0.68937
RaceWhite                                  0.17235
Senior                                     < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

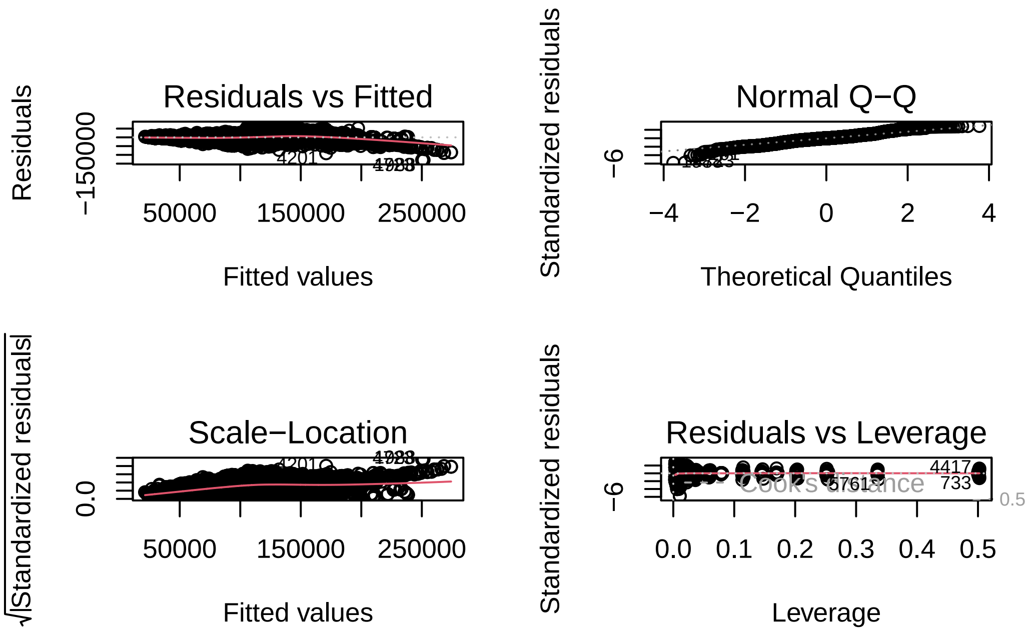
```
Residual standard error: 22450 on 5870 degrees of freedom
Multiple R-squared:  0.8203,    Adjusted R-squared:  0.8175
F-statistic: 288.2 on 93 and 5870 DF,  p-value: < 2.2e-16
```

As we can see, the most significant predictors are education level, years of experience, and senior, all of which make sense as it is logical that the more years of experience you have in a profession and the higher level of education you have, the more likely you are going to earn more money than someone who has less experience and a lesser degree. Seniority also makes sense as a senior-level position undoubtedly has more responsibilities than someone who isn't. However, we can see that several job title codes are good indicators. Jobs such as software engineer, research scientist, research director, product manager, and data scientist/analyst all appear to be very good predictors for our model. This may well be because there are simply more observations of these job titles in the data set, but all of these fields are certainly very highly-paying positions. Now, to look at the results. We can see that the model generated an R-squared value of .82, on an F-stat of 296.3, and a p-value of <2.2e-16, so needless to say, this is a respectable model; it is not perfect, but there is a strong positive correlation between the predictors and Salary.

Before we go any further, we should check the assumptions of our model to see if this dataset even can be fitted into a linear model.

```r
par(mfrow = c(2,2))
plot(fit)
```

```
Warning: not plotting observations with leverage one:
  1029, 1328, 1526, 3977
```



Checking the normal assumptions of linear regression, we can see that the data appears to fit to an acceptable level. The residuals vs. Fitted values graph is distributed mostly evenly from end to end, and the Q-Q Residuals plot, while both tails do slightly veer off the mean, they do at least mirror each other.

Now let us fit this model into our testing data. As you can see, we bound the predicted outcomes onto the testing dataset so we can compare the predicted value to the employee's actual salary.

```r
lm_preds <- predict(fit, testing) %>%
  bind_cols(testing)
```

```
New names:
* `` -> `...1`
```

```r
lm_preds
```

```
# A tibble: 663 x 10
     ...1   Age Gender Education L~1 Job T~2 Years~3 Salary Country Race   Senior
    <dbl> <dbl> <chr>          <dbl> <chr>     <dbl>  <dbl> <chr>   <chr>   <dbl>
 1 26766.    29 Female             0 Sales ~       1  25000 USA     Afri~       0
 2 30036.    24 Male               0 Sales ~       1  25000 UK      Asian       0
```

```
 3 29495.    30 Female             0 Sales ~       1  25000 Canada  Asian      0
 4 29282.    30 Female             0 Sales ~       1  25000 China   White      0
 5 25101.    21 Female             0 Sales ~       0  25000 Austra~ White      0
 6 25101.    21 Female             0 Sales ~       0  25000 Austra~ White      0
 7 25326.    21 Female             0 Sales ~       0  25000 China   Kore~      0
 8 25368.    23 Female             0 Recept~       0  25000 China   White      0
 9 22950.    25 Female             0 Sales ~       0  25000 Canada  Black      0
10 24124.    24 Female             0 Sales ~       0  25000 UK      Asian      0
# ... with 653 more rows, and abbreviated variable names 1: `Education Level`,
#   2: `Job Title`, 3: `Years of Experience`
```

While the predictions are not perfect, the model does get rather close to predicting the salary of some employees, with some predictions getting even within 1000 dollars of the actual value. However, it is not perfect, so let's tune the model to see if we can improve the accuracy.

Let us try to optimize the model by running a step-forward selection model to see what variables it would choose to use.

```
ols_step_forward_p(fit)
```

```
                                       Selection Summary
-----------------------------------------------------------------------------------------------
        Variable                                     Adj.
Step         Entered               R-Square     R-Square        C(p)            AIC           RMSE
-----------------------------------------------------------------------------------------------
   1     Age                           0.8173       0.8149     14.8766    136597.4169     22609.92
   2     `Education Level`             0.8200       0.8175    -70.5327    136511.4333     22445.66
   3     `Job Title`                       NA           NA          NA             NA            
   4     `Years of Experience`             NA           NA          NA             NA            
   5     Senior                            NA           NA          NA             NA            
-----------------------------------------------------------------------------------------------
```

Unsurprisingly, the summary has chosen the variables that I had highlighted in the original model. Interestingly, this model scraps Gender, Race, and Country; it does not consider them strong enough to influence the model.

Now that we've figured out the ideal variables for the model, let's create a new model to see if we can improve the accuracy by removing unnecessary predictors:

```r
step_fit <-
    lm(Salary ~ Age + `Education Level` +
           `Years of Experience` + Senior +
           `Job Title`, data = training)
summary(step_fit)
```

```
Call:
lm(formula = Salary ~ Age + `Education Level` + `Years of Experience` +
    Senior + `Job Title`, data = training)

Residuals:
    Min      1Q  Median      3Q     Max
-129513  -11475       2   11344   64806

Coefficients:
                                             Estimate Std. Error t value
(Intercept)                                 32183.287  11679.385   2.756
Age                                            -6.117    128.658  -0.048
`Education Level`                            6517.358    602.918  10.810
`Years of Experience`                        5500.394    159.398  34.507
Senior                                     -12011.461   1284.469  -9.351
`Job Title`Accountant                       -1840.322  14490.033  -0.127
`Job Title`Administrative Assistant        -37974.794  19444.595  -1.953
`Job Title`Back end Developer               28679.014  11333.089   2.531
`Job Title`Business Analyst                 12301.558  12414.596   0.991
`Job Title`Business Development Associate   -9901.037  14494.816  -0.683
`Job Title`Business Development Manager     26115.888  15071.593   1.733
`Job Title`Business Operations Analyst     -11773.834  25099.699  -0.469
`Job Title`Content Marketing Manager        20473.959  11584.626   1.767
`Job Title`Copywriter                       -9774.228  19442.392  -0.503
`Job Title`Customer Service Manager        -29397.335  25125.735  -1.170
`Job Title`Customer Service Representative -10227.194  15065.214  -0.679
`Job Title`Data Analyst                     54709.254  11296.302   4.843
`Job Title`Data Engineer                    27542.013  15921.582   1.730
`Job Title`Data Scientist                   54418.231  11325.278   4.805
`Job Title`Delivery Driver                  -4036.471  15883.396  -0.254
`Job Title`Digital Marketing Manager        11403.506  11691.097   0.975
`Job Title`Digital Marketing Specialist      4593.009  12838.143   0.358
`Job Title`Director of Data Science         59751.930  11743.207   5.088
`Job Title`Director of Engineering          24547.027  19480.411   1.260
`Job Title`Director of Finance              20070.802  19471.801   1.031
```

```
`Job Title`Director of HR                   8335.919  11635.185   0.716
`Job Title`Director of Human Resources     18326.628  19486.263   0.940
`Job Title`Director of Marketing           23711.257  11549.211   2.053
`Job Title`Director of Operations          11566.978  13544.414   0.854
`Job Title`Engineer                         9509.945  25127.956   0.378
`Job Title`Event Coordinator              -32736.831  19442.455  -1.684
`Job Title`Financial Advisor               16295.364  15068.647   1.081
`Job Title`Financial Analyst               17763.123  11678.115   1.521
`Job Title`Financial Manager               43635.738  11412.454   3.824
`Job Title`Front end Developer             22016.991  11334.787   1.942
`Job Title`Front End Developer             18041.921  11977.121   1.506
`Job Title`Full Stack Engineer             34564.765  11318.901   3.054
`Job Title`Graphic Designer                  176.921  12353.030   0.014
`Job Title`HR Coordinator                  -9324.122  12062.302  -0.773
`Job Title`HR Generalist                       7.556  11468.637   0.001
`Job Title`HR Manager                       3221.066  17160.489   0.188
`Job Title`Human Resources Coordinator    -10359.193  11722.287  -0.884
`Job Title`Human Resources Manager         14610.811  11396.271   1.282
`Job Title`IT Consultant                   27522.368  19463.689   1.414
`Job Title`IT Support Specialist           -5204.361  19445.257  -0.268
`Job Title`Juniour HR Coordinator          -5534.826  17154.220  -0.323
`Job Title`Juniour HR Generalist             -12.790  19447.475  -0.001
`Job Title`Manager                         24297.539  19480.880   1.247
`Job Title`Marketing Analyst                7350.420  11400.657   0.645
`Job Title`Marketing Coordinator            8692.080  11375.633   0.764
`Job Title`Marketing Director              63822.900  11644.619   5.481
`Job Title`Marketing Manager               17517.708  11320.783   1.547
`Job Title`Marketing Specialist             6535.768  13491.332   0.484
`Job Title`Operations Analyst              -8245.329  14069.841  -0.586
`Job Title`Operations Coordinator          21138.475  15885.506   1.331
`Job Title`Operations Manager              13629.780  11422.183   1.193
`Job Title`Product Designer                 6072.689  11535.889   0.526
`Job Title`Product Manager                 56684.433  11310.769   5.012
`Job Title`Product Marketing Manager       27152.087  11624.325   2.336
`Job Title`Project Coordinator              5406.736  15897.205   0.340
`Job Title`Project Engineer                52135.964  11348.870   4.594
`Job Title`Project Manager                 24122.171  11926.693   2.023
`Job Title`Receptionist                    -7038.750  11671.316  -0.603
`Job Title`Recruiter                      -22324.934  17147.014  -1.302
`Job Title`Research Director               49901.967  11645.350   4.285
`Job Title`Research Scientist              43083.422  11495.758   3.748
`Job Title`Sales Associate                 -8388.192  11359.752  -0.738
`Job Title`Sales Director                  32067.004  11613.304   2.761
```

```
`Job Title`Sales Executive                -2859.382  11955.783  -0.239
`Job Title`Sales Manager                  19244.112  11643.177   1.653
`Job Title`Sales Representative           -7445.820  11539.282  -0.645
`Job Title`Scientist                      16015.931  17190.268   0.932
`Job Title`Social Media Manager            -462.120  12841.878  -0.036
`Job Title`Social Media Specialist       -10030.542  25097.733  -0.400
`Job Title`Software Developer              2433.100  11361.740   0.214
`Job Title`Software Engineer              45173.197  11258.280   4.012
`Job Title`Software Engineer Manager      35376.316  11353.487   3.116
`Job Title`Training Specialist           -19216.690  19443.214  -0.988
`Job Title`UX Designer                    17462.816  15083.762   1.158
`Job Title`Web Developer                   -371.344  11422.028  -0.033
                                         Pr(>|t|)
(Intercept)                              0.005877 **
Age                                      0.962079
`Education Level`                         < 2e-16 ***
`Years of Experience`                     < 2e-16 ***
Senior                                    < 2e-16 ***
`Job Title`Accountant                    0.898940
`Job Title`Administrative Assistant      0.050870 .
`Job Title`Back end Developer            0.011414 *
`Job Title`Business Analyst              0.321778
`Job Title`Business Development Associate  0.494587
`Job Title`Business Development Manager  0.083186 .
`Job Title`Business Operations Analyst   0.639028
`Job Title`Content Marketing Manager     0.077223 .
`Job Title`Copywriter                    0.615175
`Job Title`Customer Service Manager      0.242045
`Job Title`Customer Service Representative 0.497252
`Job Title`Data Analyst                  1.31e-06 ***
`Job Title`Data Engineer                 0.083709 .
`Job Title`Data Scientist                1.59e-06 ***
`Job Title`Delivery Driver               0.799403
`Job Title`Digital Marketing Manager     0.329402
`Job Title`Digital Marketing Specialist  0.720534
`Job Title`Director of Data Science      3.73e-07 ***
`Job Title`Director of Engineering       0.207688
`Job Title`Director of Finance           0.302695
`Job Title`Director of HR                0.473748
`Job Title`Director of Human Resources   0.347005
`Job Title`Director of Marketing         0.040111 *
`Job Title`Director of Operations        0.393138
`Job Title`Engineer                      0.705102
```

```
`Job Title`Event Coordinator              0.092277 .
`Job Title`Financial Advisor              0.279560
`Job Title`Financial Analyst              0.128298
`Job Title`Financial Manager              0.000133 ***
`Job Title`Front end Developer            0.052133 .
`Job Title`Front End Developer            0.132027
`Job Title`Full Stack Engineer            0.002270 **
`Job Title`Graphic Designer               0.988574
`Job Title`HR Coordinator                 0.439555
`Job Title`HR Generalist                  0.999474
`Job Title`HR Manager                     0.851116
`Job Title`Human Resources Coordinator    0.376885
`Job Title`Human Resources Manager        0.199869
`Job Title`IT Consultant                  0.157404
`Job Title`IT Support Specialist          0.788985
`Job Title`Juniour HR Coordinator         0.746971
`Job Title`Juniour HR Generalist          0.999475
`Job Title`Manager                        0.212355
`Job Title`Marketing Analyst              0.519123
`Job Title`Marketing Coordinator          0.444840
`Job Title`Marketing Director             4.41e-08 ***
`Job Title`Marketing Manager              0.121822
`Job Title`Marketing Specialist           0.628090
`Job Title`Operations Analyst             0.557879
`Job Title`Operations Coordinator         0.183347
`Job Title`Operations Manager             0.232811
`Job Title`Product Designer               0.598618
`Job Title`Product Manager                5.56e-07 ***
`Job Title`Product Marketing Manager      0.019535 *
`Job Title`Project Coordinator            0.733789
`Job Title`Project Engineer               4.44e-06 ***
`Job Title`Project Manager                0.043166 *
`Job Title`Receptionist                   0.546478
`Job Title`Recruiter                      0.192977
`Job Title`Research Director              1.86e-05 ***
`Job Title`Research Scientist             0.000180 ***
`Job Title`Sales Associate                0.460293
`Job Title`Sales Director                 0.005776 **
`Job Title`Sales Executive                0.810987
`Job Title`Sales Manager                  0.098420 .
`Job Title`Sales Representative           0.518785
`Job Title`Scientist                      0.351537
`Job Title`Social Media Manager           0.971295
```

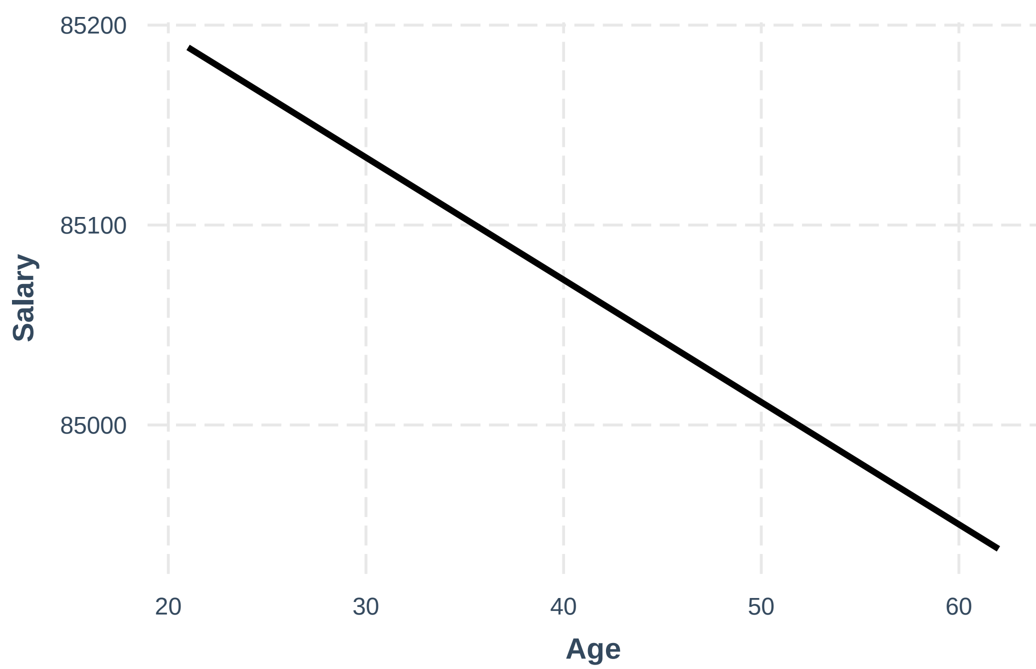
```
`Job Title`Social Media Specialist         0.689422
`Job Title`Software Developer              0.830439
`Job Title`Software Engineer               6.08e-05 ***
`Job Title`Software Engineer Manager       0.001843 **
`Job Title`Training Specialist             0.323022
`Job Title`UX Designer                     0.247024
`Job Title`Web Developer                   0.974065
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22450 on 5884 degrees of freedom
Multiple R-squared:   0.82, Adjusted R-squared:  0.8175
F-statistic: 339.2 on 79 and 5884 DF,  p-value: < 2.2e-16
```

Unfortunately, Rsq remained nearly the same. However, one saving grace of the tune is that we were able to slightly reduce residual standard error and increase our F-statistic, so it may not look like it at first glance, but the model is still stronger than our initial attempt, even if only slightly.

Now that we've tuned our model, let us visualize the predictions.

This first plot depicts the average predicted value of Salary at each age in the data.

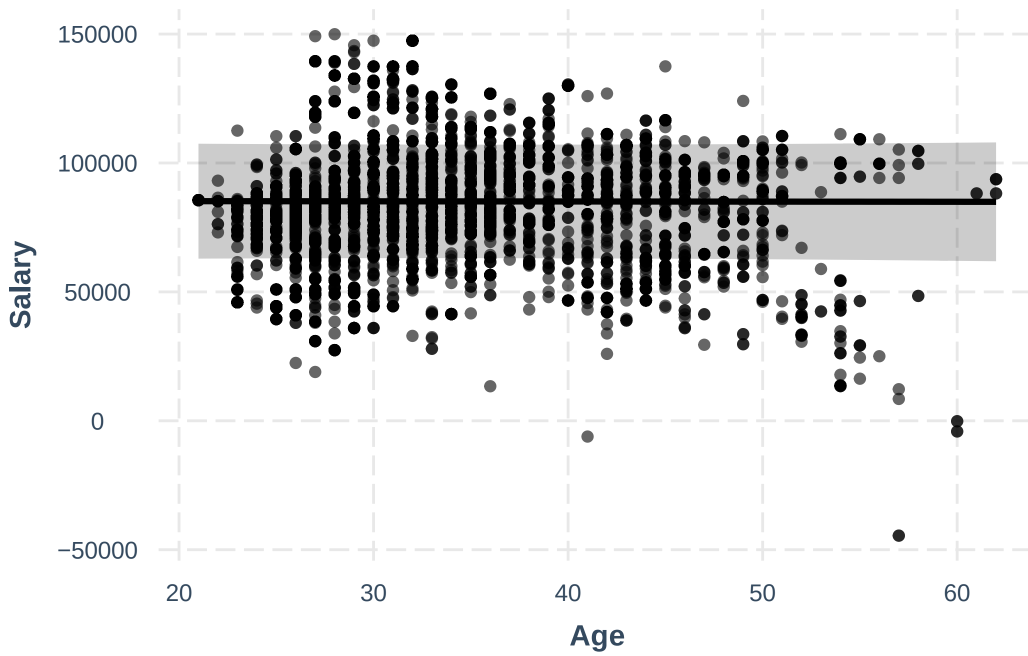
```r
effect_plot(step_fit, pred = Age)
```



As we can see, this shows a very strong correlation between salary and age.

This second plot once again depicts salary vs age, but this time plots the residual values along with showing the confidence interval of which the model operates. We can see most of our residuals lie within the interval, although there are a few outliers at both ends.

```
effect_plot(step_fit, pred = Age, interval = TRUE, partial.residuals = TRUE)
```



Tree Methods:

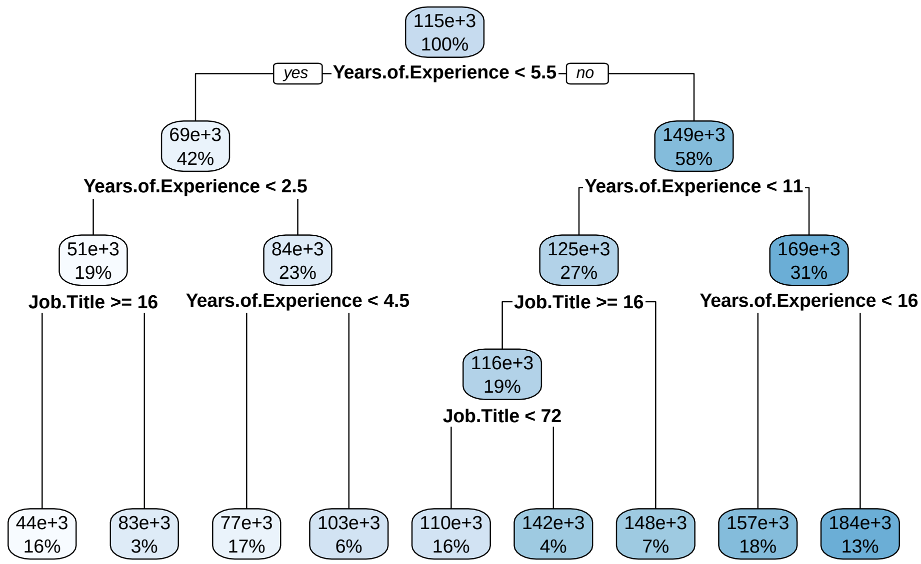
For our second model, we want to use the power of Tree methods to see if it could give us a better answer to our main question than multiple linear regression. We will be mainly focusing on the Decision Tree method, but we will also create a Random Forest tree for comparison.

Decision Tree:

As we learned from class, we know that decision trees can mirror human decision-making more than other methods. We want to try to put this to the test to create a decision tree model based on our salary data to see if it can accurately predict an employee's salary using binary decision-making.

To begin, we will make an untidy decision tree to visualize the decision-making process the model will take to determine salary.

```r
# Non-tidy way (for visualization purposes)
tree_fit <- rpart(Salary ~., data = train)

rpart.plot(tree_fit)
```



We can see from this output that Years of Experience and job titles are very influential in decision-making. To be able to print this tree without having tens of job names crowd out the actual Boolean expression, we coded the job title to be a numeric value and factorized it, so while it's a bit harder to understand what is happening, the lower a job title's value is, the less money the position makes. Back to the tree, we can see that the longer someone works, the more money they will earn, and there are no questions about what position they will hold; they will still earn more money due to their experience. However, when we go down the tree in the opposite direction (meaning an employee has less experience), their position starts to play a more pivotal role.

Below we can see the decision tree model fitted onto the testing data. We can also see the predicted values compared to the actual salary values.

```r
tree_fit_2 <- rpart(Salary ~., data = training)
tree_preds <- predict(tree_fit_2, newdata = testing) %>%
  bind_cols(testing)
```

```
New names:
```

```
* `` -> `...1`
```

```r
tree_preds
```

```
# A tibble: 663 x 10
     ...1   Age Gender Education L~1 Job T~2 Years~3 Salary Country Race  Senior
    <dbl> <dbl> <chr>          <dbl> <chr>     <dbl>  <dbl> <chr>   <chr>  <dbl>
 1 37692.    29 Female             0 Sales ~       1  25000 USA     Afri~      0
 2 37692.    24 Male               0 Sales ~       1  25000 UK      Asian      0
 3 37692.    30 Female             0 Sales ~       1  25000 Canada  Asian      0
 4 37692.    30 Female             0 Sales ~       1  25000 China   White      0
 5 37692.    21 Female             0 Sales ~       0  25000 Austra~ White      0
 6 37692.    21 Female             0 Sales ~       0  25000 Austra~ White      0
 7 37692.    21 Female             0 Sales ~       0  25000 China   Kore~      0
 8 37692.    23 Female             0 Recept~       0  25000 China   White      0
 9 37692.    25 Female             0 Sales ~       0  25000 Canada  Black      0
10 37692.    24 Female             0 Sales ~       0  25000 UK      Asian      0
# ... with 653 more rows, and abbreviated variable names 1: `Education Level`,
#   2: `Job Title`, 3: `Years of Experience`
```

While the predictions appear to be fairly accurate to the actual values, we can see that the model is not good at predicting small changes within similar records. Therefore, we need to tune for it to factor in these smaller changes into the data.

```r
# Tidy way + tuning
tree_model <- decision_tree(mode = "regression",
                            cost_complexity = tune(),
                            tree_depth = tune()) %>%
  set_engine("rpart")


data_recipe <- recipe(Salary ~., training)

wf <- workflow() %>%
  add_recipe(data_recipe) %>%
  add_model(tree_model)

tree_grid <- grid_regular(cost_complexity(),
                          tree_depth(),
                          levels = 5)
```

```r
cv_samples <- vfold_cv(training)

tree_tune <- wf %>%
  tune_grid(
    resamples = cv_samples,
    grid = tree_grid
  )

best_tree <- tree_tune %>%
  select_best(metric = "rmse")

final_wf <- wf %>%
  finalize_workflow(best_tree)


final_wf %>%
  last_fit(salary_split) %>%
  collect_metrics()
```

```
# A tibble: 2 x 4
  .metric .estimator .estimate .config
  <chr>   <chr>          <dbl> <chr>
1 rmse    standard    9448.    Preprocessor1_Model1
2 rsq     standard       0.969 Preprocessor1_Model1
```

```r
tuned_tree_preds <- final_wf %>%
  last_fit(salary_split) %>%
  collect_predictions() %>%
  bind_cols(testing)
```

```
New names:
* `Salary` -> `Salary...4`
* `Salary` -> `Salary...11`
```

As we can see from the output of the tuned decision tree above, we get an r-squared value of .958, which is an incredible accuracy considering decision trees often suffer from low predictive power. However, our RMSE value is at a staggering 10491.67, so our outliers are heavily impacting the model in a negative way, which is usually the case for Decision Trees.

```
tuned_tree_preds
```

```
# A tibble: 663 x 14
   id           .pred  .row Salar~1 .config   Age Gender Educa~2 Job T~3 Years~4
   <chr>        <dbl> <int>   <dbl> <chr>   <dbl> <chr>    <dbl> <chr>     <dbl>
 1 train/test~ 26491.     8   25000 Prepro~    29 Female       0 Sales ~       1
 2 train/test~ 25733.    13   25000 Prepro~    24 Male         0 Sales ~       1
 3 train/test~ 26491.    34   25000 Prepro~    30 Female       0 Sales ~       1
 4 train/test~ 26491.    38   25000 Prepro~    30 Female       0 Sales ~       1
 5 train/test~ 25143.    49   25000 Prepro~    21 Female       0 Sales ~       0
 6 train/test~ 25143.    53   25000 Prepro~    21 Female       0 Sales ~       0
 7 train/test~ 25143.    57   25000 Prepro~    21 Female       0 Sales ~       0
 8 train/test~ 25143.    76   25000 Prepro~    23 Female       0 Recept~       0
 9 train/test~ 25143.    77   25000 Prepro~    25 Female       0 Sales ~       0
10 train/test~ 25143.    82   25000 Prepro~    24 Female       0 Sales ~       0
# ... with 653 more rows, 4 more variables: Salary...11 <dbl>, Country <chr>,
#   Race <chr>, Senior <dbl>, and abbreviated variable names 1: Salary...4,
#   2: `Education Level`, 3: `Job Title`, 4: `Years of Experience`
```

Looking at our predicted values now, we can see that the model is way more accurate at factoring in slight differences between similar employees. Overall, this tuned regression decision tree does a really good job of making accurate predictions.

Random Forest Tree:

For comparison, let us look at this Random Forest Tree

```
rf_model <- rand_forest() %>%
    set_engine("ranger") %>%
    set_mode("regression")

# workflow
rf_wf <- workflow() %>%
    add_model(rf_model) %>%
    add_recipe(data_recipe)

# fit the regression tree
rf_fit <- rf_wf %>% fit(training)

# predict
```

```r
testing$pred <- predict(rf_fit, testing)$.pred

# metrics
testing %>% metrics(Salary, pred)
```

```
# A tibble: 3 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard   11495.
2 rsq     standard       0.958
3 mae     standard    7746.
```

The Random Forest tree did ever so slightly worse than the tuned decision tree model, but it still is very accurate at predicting salary.

Ridge Regression:

We chose ridge regression as our final model in the hopes that we could reduce the high amount of variance in our data to create an even more accurate model than our tuned Decision Tree.

Let us start with a ridge model that we manually assign the penalties for. Let us use a manual penalty of 4 for the estimate. We must also center and scale all of our predictors to standardize them before we fit the model.

```r
ridge_recipe <- recipe(Salary ~ ., data = train) %>%
  step_center(all_nominal_predictors()) %>%
  step_scale(all_nominal_predictors())


ridge_model <- linear_reg(mixture = 0, penalty = .1) %>%
  set_engine("glmnet")

ridge_wf <- workflow() %>%
  add_recipe(ridge_recipe) %>%
  add_model(ridge_model) %>%
  fit(train)
extract_fit_parsnip(ridge_wf) %>% tidy(penalty = 4)
```

```
# A tibble: 9 x 3
  term                 estimate penalty
```

```
  <chr>                  <dbl>   <dbl>
1 (Intercept)           37921.       4
2 Age                     284.       4
3 Gender                 5999.       4
4 Education.Level       14904.       4
5 Job.Title              -117.       4
6 Years.of.Experience    5046.       4
7 Country                -294.       4
8 Race                     67.8      4
9 Senior                -4887.       4
```

From the output of the model, we can tell that it is not very accurate at all. The estimated values are extremely far away from zero.

Now, let us try tuning the model to see if we can improve the accuracy of the ridge regression.

```r
## TUNING
folds <-vfold_cv(train)

model <- linear_reg(mixture = 0, penalty = tune()) %>%
  set_engine("glmnet")

tuned_wf <- workflow() %>%
  add_recipe(ridge_recipe) %>%
  add_model(ridge_model)

ridge_grid <- grid_regular(mixture(), penalty(), levels = 10)

tuned_grid <- tune_grid(tuned_wf, resamples = folds, grid = ridge_grid)
```

```
Warning: No tuning parameters have been detected, performance will be evaluated
using the resamples with no tuning. Did you want to [tune()] parameters?
```

```r
tuned_grid %>% collect_metrics() %>% filter(.metric == "rmse") %>% arrange(mean)
```

```
# A tibble: 1 x 6
  .metric .estimator   mean     n std_err .config
  <chr>   <chr>       <dbl> <int>   <dbl> <chr>
1 rmse    standard   28983.    10    236. Preprocessor1_Model1
```

The RMSE is almost three times larger than our decision tree model. It appears this model is not accurate at all at predicting salary.

Before we make any assumptions, let us take a look at the predictions

```r
tuned_grid %>%
    select_best() %>%
    finalize_workflow(tuned_wf, .) %>%
    last_fit(ridge_split) %>%
    collect_predictions()
```

```
Warning: No value of `metric` was given; metric 'rmse' will be used.
```

```
# A tibble: 663 x 5
   id                .pred  .row Salary .config
   <chr>             <dbl> <int>  <dbl> <chr>
 1 train/test split 49701.     2  25000 Preprocessor1_Model1
 2 train/test split 49371.    12  25000 Preprocessor1_Model1
 3 train/test split 50812.    35  25000 Preprocessor1_Model1
 4 train/test split 49510.    36  25000 Preprocessor1_Model1
 5 train/test split 49646.    40  25000 Preprocessor1_Model1
 6 train/test split 41280.    47  25000 Preprocessor1_Model1
 7 train/test split 40670.    58  25000 Preprocessor1_Model1
 8 train/test split 42894.    62  25000 Preprocessor1_Model1
 9 train/test split 42792.    65  25000 Preprocessor1_Model1
10 train/test split 43316.   112  25000 Preprocessor1_Model1
# ... with 653 more rows
```

Our tuned ridge regression model overestimates salary for every employee. It is now safe to say that this model is the least accurate out of the three that we have created today.

Comparison:

To compare our models, our decision tree by far did the best, as we have previously stated, but our multiple linear regression model was still respectable, being able to predict accurately within 82% of the data. Now for the ridge regression model. Our ridge regression model was not accurate even after being scaled, centered, and tuned. We are led to believe that this may have been due to the extremely large variance within the dataset.

Conclusion:

In conclusion, we were able to answer all of our questions after analyzing and modeling the data.

Starting with our minor questions:

- Women do, in fact, get paid less than men; while men do have lower-paying jobs than women, on average their jobs are likely to pay less than a man's.
- Age does play a large role in how much an employee earns. experience and age go hand in hand with one another, as you are going to gain experience as you age (unless you are unemployed or start work later than the average person). Still, being older in your field almost certainly leads to better pay. We did find, however, that 60-year-olds make about the same as 50-year-olds do on average. So do not anticipate a pay raise heading into your pre-retirement years
- Having a senior-level position does indeed lead to a pay increase on average, and while we found a handful of outliers under 30, most employees in a senior-level position were older than this mark.
- Having a higher level of education does lead to a higher salary, and quite significantly so. We would hope this would be the case considering the amount of time and resources it takes to get each higher level of education.
- No, you do not need to move to another country to get a better wage. While there may be other reasons (such as benefits) to entice you to move abroad, salary should not be one of them.

To finish off this project, let us answer our main question: Can we accurately predict the salary of an employee given the predictors from the dataset?

The answer to this question is yes. Using a tuned decision tree model we were able to achieve an accuracy of 95% on our testing data. The model is not entirely perfect, but it is certainly good for the fact that it is predicting using regression, which is extremely hard to achieve good accuracy for.

To say the accuracy of our decision tree was a surprise would be an understatement. Considering the relatively small amount of variables within the data set we thought we would not be able to accurately predict salary, so to create such an accurate model was a pleasant surprise for us.