FinalProjectR

Shreeya Chugh, Abhinav Khanna, Tyler Griffith

30/03/2022

Name of the Dataset: Data Scientist Salary Link to the Dataset: https://www.kaggle.com/nikhilbhathi/data-scientist-salary-us-glassdoor

In this analysis, we are studying how the different attributes of an employee impacts salary. This Data Scientist Salary data set gives us the estimated salary of an employee and many other variables about that employee, their qualifications and the company they work for as well as their positions in their respective companies.

The unit of observation is the position of the data scientist at a particular company.

The following code loads all the necessary packages used in the data analysis.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

```
library(readr)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4

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

```
library(ggridges)
data <- read_csv("/Users/ruchhichughgmail.com/Downloads/data_cleaned_2021 3.csv")
```

```
## Rows: 742 Columns: 42
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): Job Title, Salary Estimate, Job Description, Company Name, Locatio...
## dbl (25): index, Rating, Founded, Hourly, Employer provided, Lower Salary, U...
##
## 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.
```

```
glimpse(data)
```

```
## Rows: 742
## Columns: 42
## $ index                <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
## $ `Job Title`          <chr> "Data Scientist", "Healthcare Data Scientist", "Da~
## $ `Salary Estimate`    <chr> "$53K-$91K (Glassdoor est.)", "$63K-$112K (Glassdo~
## $ `Job Description`    <chr> "Data Scientist\nLocation: Albuquerque, NM\nEducat~
## $ Rating               <dbl> 3.8, 3.4, 4.8, 3.8, 2.9, 3.4, 4.1, 3.8, 3.3, 4.6, ~
## $ `Company Name`       <chr> "Tecolote Research\n3.8", "University of Maryland ~
## $ Location             <chr> "Albuquerque, NM", "Linthicum, MD", "Clearwater, F~
## $ Headquarters         <chr> "Goleta, CA", "Baltimore, MD", "Clearwater, FL", "~
## $ Size                 <chr> "501 - 1000", "10000+", "501 - 1000", "1001 - 5000~
## $ Founded              <dbl> 1973, 1984, 2010, 1965, 1998, 2000, 2008, 2005, 20~
## $ `Type of ownership`  <chr> "Company - Private", "Other Organization", "Compan~
## $ Industry             <chr> "Aerospace & Defense", "Health Care Services & Hos~
## $ Sector               <chr> "Aerospace & Defense", "Health Care", "Business Se~
## $ Revenue              <chr> "$50 to $100 million (USD)", "$2 to $5 billion (US~
## $ Competitors          <chr> "-1", "-1", "-1", "Oak Ridge National Laboratory, ~
## $ Hourly               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ `Employer provided`  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ `Lower Salary`       <dbl> 53, 63, 80, 56, 86, 71, 54, 86, 38, 120, 126, 64, ~
## $ `Upper Salary`       <dbl> 91, 112, 90, 97, 143, 119, 93, 142, 84, 160, 201, ~
## $ `Avg Salary(K)`      <dbl> 72.0, 87.5, 85.0, 76.5, 114.5, 95.0, 73.5, 114.0, ~
## $ company_txt          <chr> "Tecolote Research", "University of Maryland Medic~
## $ `Job Location`       <chr> "NM", "MD", "FL", "WA", "NY", "TX", "MD", "CA", "N~
## $ Age                  <dbl> 48, 37, 11, 56, 23, 21, 13, 16, 7, 12, 10, 53, 59,~
## $ Python               <dbl> 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,~
## $ spark                <dbl> 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0,~
## $ aws                  <dbl> 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,~
## $ excel                <dbl> 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1,~
## $ sql                  <dbl> 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0,~
## $ sas                  <dbl> 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ keras                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ pytorch              <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ scikit               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ tensor               <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,~
## $ hadoop               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,~
## $ tableau              <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,~
```

```
## $ bi                  <dbl> 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,~
## $ flink               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ mongo               <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ google_an           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ job_title_sim       <chr> "data scientist", "data scientist", "data scientis~
## $ seniority_by_title  <chr> "na", "na", "na", "na", "na", "na", "na", "na", "n~
## $ Degree              <chr> "M", "M", "M", "na", "na", "na", "na", "M", "P", "~
```

The following code cleans the names by removing the spaces between them and lowers their cases. For this, we first installed the janitor package and then loaded it.

```
# installing the package
if (!require("janitor")) install.packages("janitor")
```

```
## Loading required package: janitor
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
# loading
library(janitor)
data <- data%>%
  # cleaning the names
  janitor:: clean_names()
# finally viewing it
View(data)
```

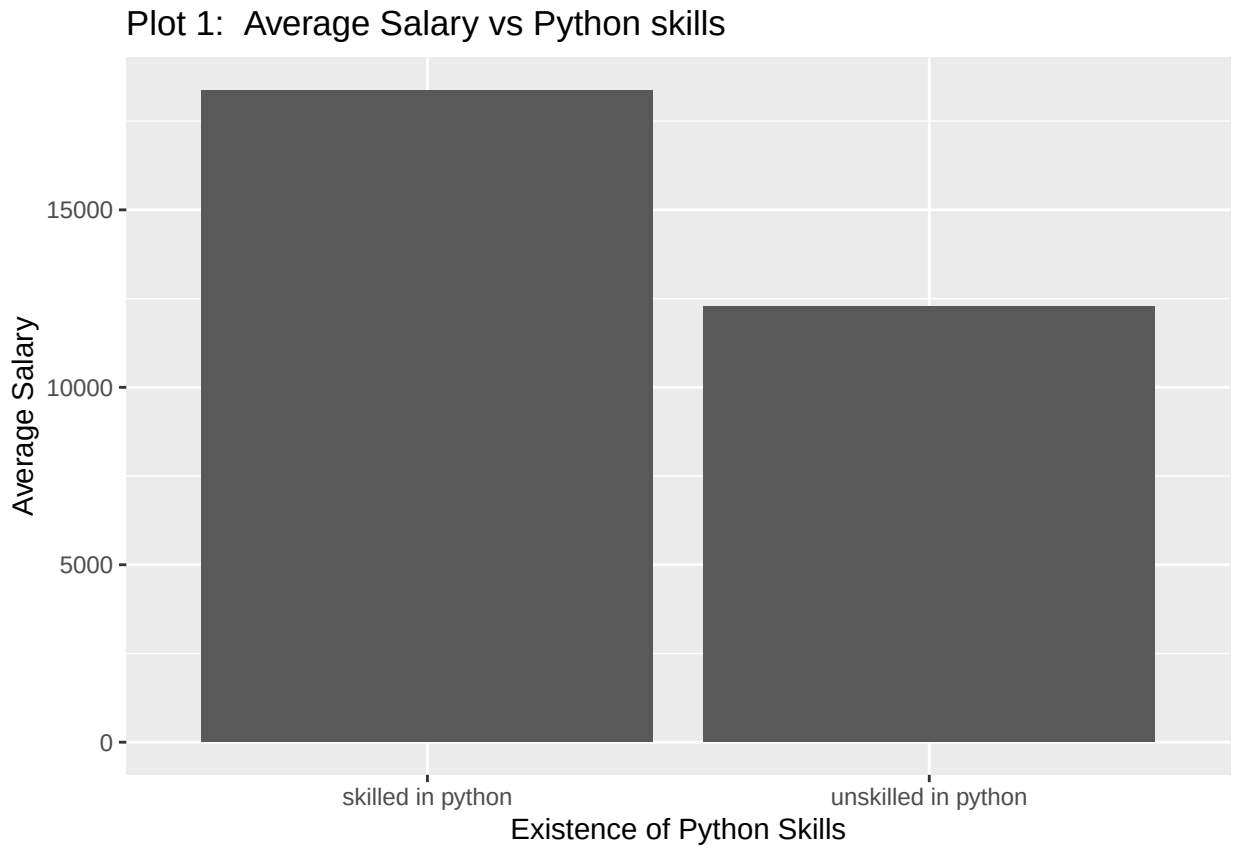
In the following code, we are diving the age of the company into ranges for ease in visualization. We are using mutate because we are creating a new variable called age_company_range storing company age ranges.

```
data <- data%>%

  mutate(age_company_range = case_when(
    age %in% seq(0,10)~ "0-10",
    age %in% seq(11,20)~ "11-20",
    age %in% seq(21,50)~ "21-50",
    age %in% seq(51,70)~ "51-70",
    age %in% seq(71,100)~ "71-100",
    age %in% seq(101,200)~ "101-200",
    age %in% seq(201,1000)~ "201-1000"
  ))
```

```
plotdata1 <- data%>%
  group_by(python, avg_salary_k)%>%
  summarize(avg_python = mean(python))%>%
  mutate(avg_python = case_when(
    avg_python == "1" ~ "skilled in python",
    avg_python == "0" ~ "unskilled in python"
  ))
```

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

```
ggplot(plotdata1, aes(x = avg_python, y = avg_salary_k)) +
  geom_col() +
  labs(
    x = "Existence of Python Skills",
    y = "Average Salary",
    title = "Plot 1:  Average Salary vs Python skills"
  )
```



In order to generate plot 2, we first grouped it by python and average_salary_k(which is the average salary) because only these variables are required, summarized it, and brought out the meaning out of 0 and 1, 0 meaning unskilled and 1 meaning skilled. Plot 2 shows the average salary by python skills. Those skilled in python have a higher average salary as compared to those unskilled.

```
plotdata2 <- data%>%
  group_by(sql, avg_salary_k)%>%
  summarize(avg_sql = mean(sql))%>%
  mutate(avg_sql = case_when(
    sql == "1" ~ "skilled in sql",
    sql == "0" ~ "unskilled in sql"
  ))
```

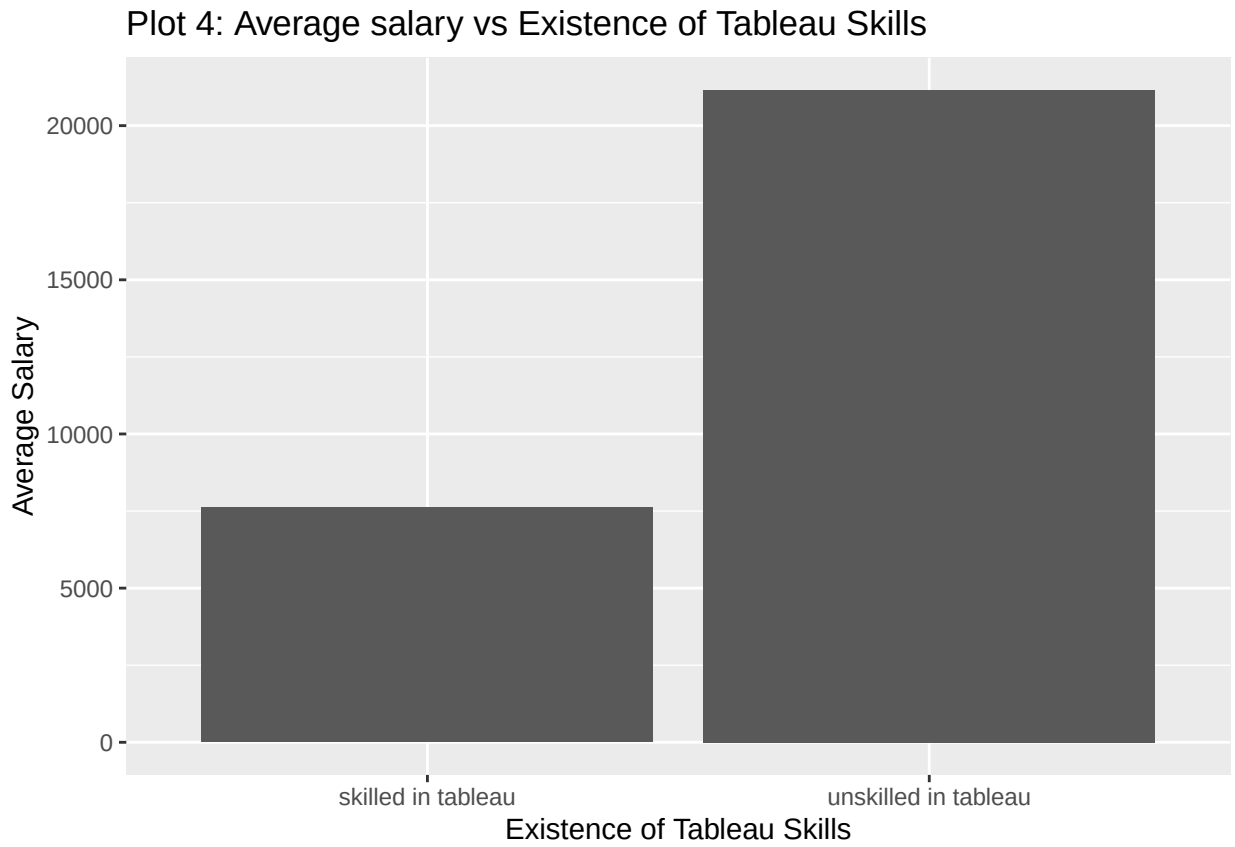
```
## `summarise()` has grouped output by 'sql'. You can override using the `.groups`
## argument.
```

```
ggplot(plotdata2, aes(x = avg_sql, y = avg_salary_k))+
  geom_col() +
  labs(
    x = "Existence of skills in sql",
    y = "Average Salary",
    title = "Plot 3: Average Salary vs Skills in SQL"
  )
```


```
plotdata3 <- data%>%
  group_by(tableau, avg_salary_k)%>%
  summarize(avg_tableau = mean(tableau))%>%
  mutate(avg_tableau = case_when(
    tableau == "1" ~ "skilled in tableau",
    tableau == "0" ~ "unskilled in tableau"
  ))
```

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

```
ggplot(plotdata3, aes(x = avg_tableau, y = avg_salary_k))+
  geom_col() +
  labs(
    x = "Existence of Tableau Skills",
    y = "Average Salary",
    title = "Plot 4: Average salary vs Existence of Tableau Skills")
```

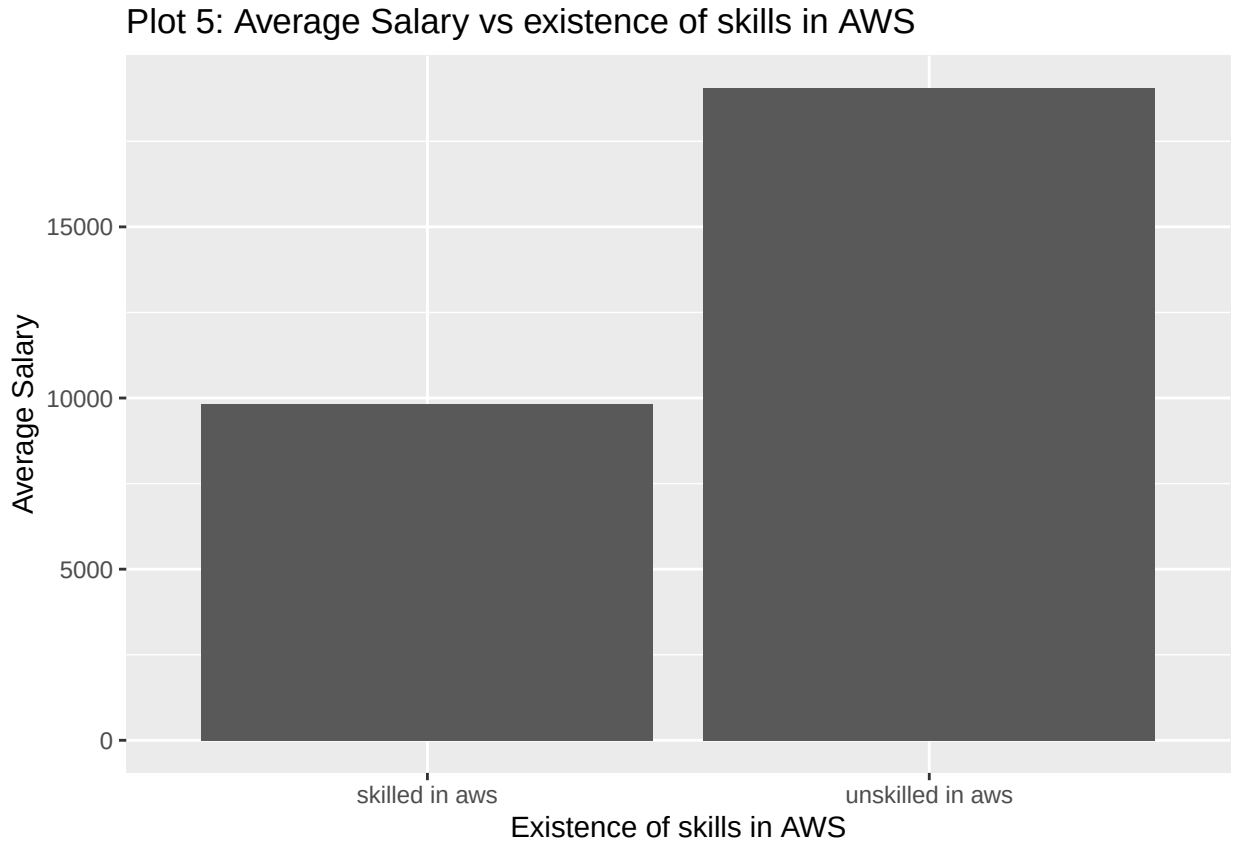


Here, we grouped sql and average_salary_k and then summarized sql. We used case_when() to derive meaning out of 0(unskilled) and 1(skilled). Plot 3 shows how the average salary varies with existence of tableau skills. Suprisingly, it shows that those who are unskilled in tableau have a higher average salary.

```
plotdata4 <- data%>%
  group_by(aws, avg_salary_k)%>%
  summarize(avg_aws = mean(aws))%>%
  mutate(avg_aws = case_when(
    aws == 1 ~ "skilled in aws",
    aws == 0 ~ "unskilled in aws"
  ))
```

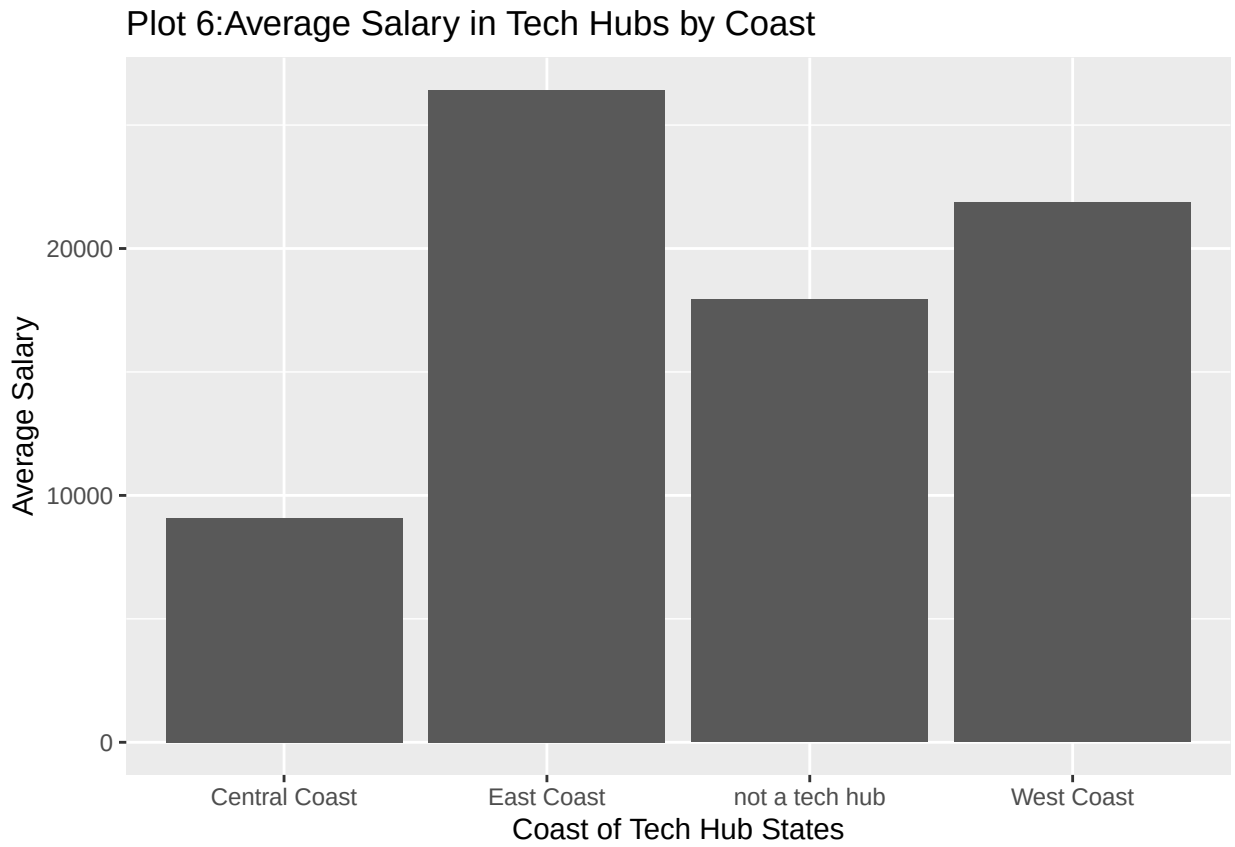
```
## `summarise()` has grouped output by 'aws'. You can override using the `.groups`
## argument.
```

```
ggplot(plotdata4, aes(x = avg_aws, y = avg_salary_k))+
  geom_col() +
  labs(
    x = "Existence of skills in AWS",
    y = "Average Salary",
    title = "Plot 5: Average Salary vs existence of skills in AWS"
  )
```



Here we first grouped by aws and average_salary_k, and then summarized aws. We used case_when() to derive meaning out of 0(unskilled) and 1(skilled).Plot 4 shows the average salary vs the existence of skills in Amazon Web Services.Suprisingly, it seems that those unskilled in AWS have a higher average salary.

```
plotdata5 <- data%>%
  group_by(avg_salary_k, job_location)%>%
  mutate(techhubs_coast = case_when(
    job_location == "NY"| job_location == "MA"| job_location == "PA"| job_location == "GA"| job_locatio
    job_location == "CA"| job_location == "WA"| job_location == "AZ"| job_location == "OR"| job_locatio
    job_location == "IL"| job_location == "MI"| job_location == "OH"| job_location == "TX"~ "Central Co

  ))%>%
  mutate(techhubs_coast = ifelse(is.na(techhubs_coast), "not a tech hub",techhubs_coast))
ggplot(plotdata5, aes(x = techhubs_coast, y = avg_salary_k))+
  geom_col() +
  labs(
    x = "Coast of Tech Hub States",
    y = "Average Salary",
    title = "Plot 6:Average Salary in Tech Hubs by Coast"
  )
```



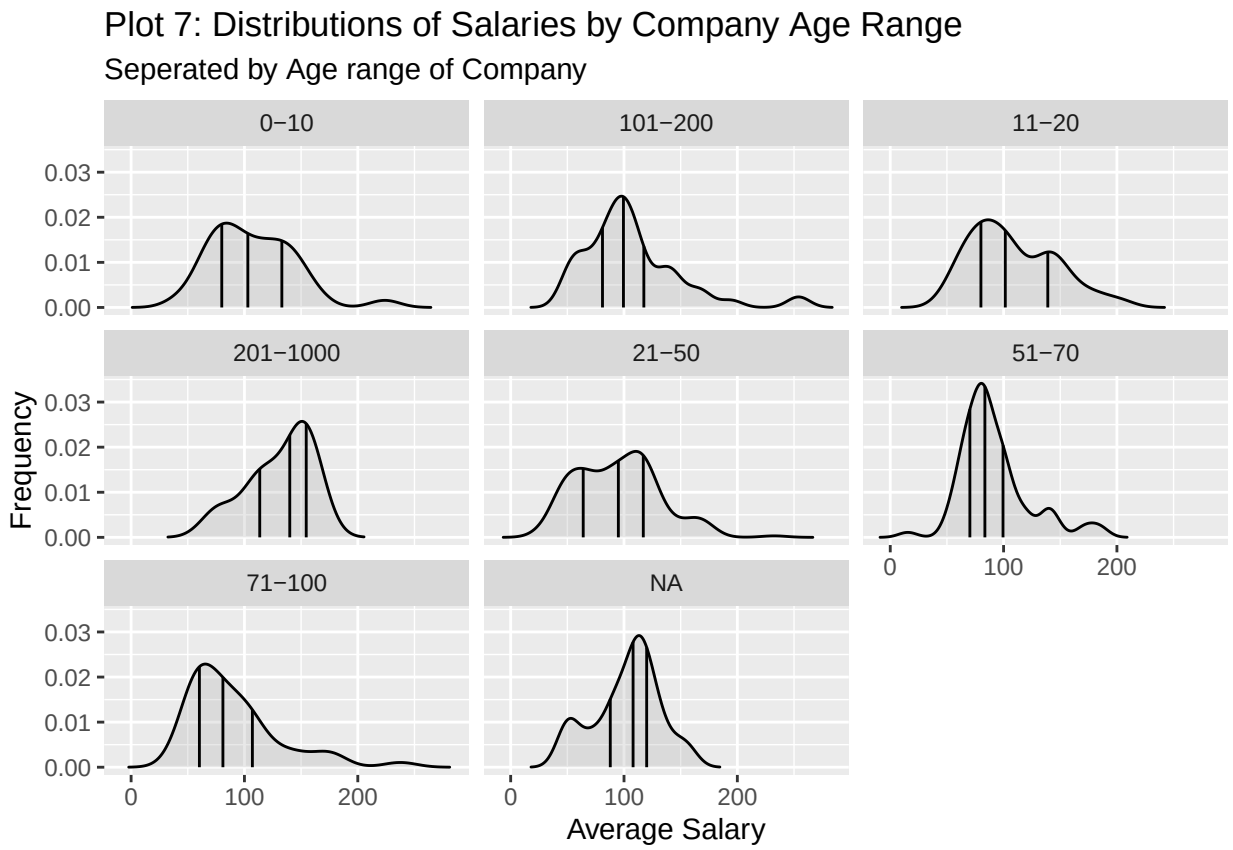
We grouped by job_location and average_salary_k and then used case_when to create the techhubs_coast variable which was used in the plot. Plot 5 shows the average salary by location of company. We restricted this to the tech hubs. It seems that those in east coast have a higher average salary, followed by west coast and central coast.

```
ggplot(data, aes(x = avg_salary_k, y = 0)) +
   geom_density_ridges(alpha = 0.3,
                       quantile_lines = TRUE,
                       quantiles = c(0.25, 0.5,0.75)) +
  facet_wrap(vars(age_company_range)) +
  labs(title = "Plot 7: Distributions of Salaries by Company Age Range",
       subtitle = "Seperated by Age range of Company",
       x = "Average Salary",
       y = "Frequency")
```

```
## Picking joint bandwidth of 13.2

## Picking joint bandwidth of 9.9

## Picking joint bandwidth of 12.3

## Picking joint bandwidth of 14.7

## Picking joint bandwidth of 11.4

## Picking joint bandwidth of 8.18
```

```
## Picking joint bandwidth of 14.5
```

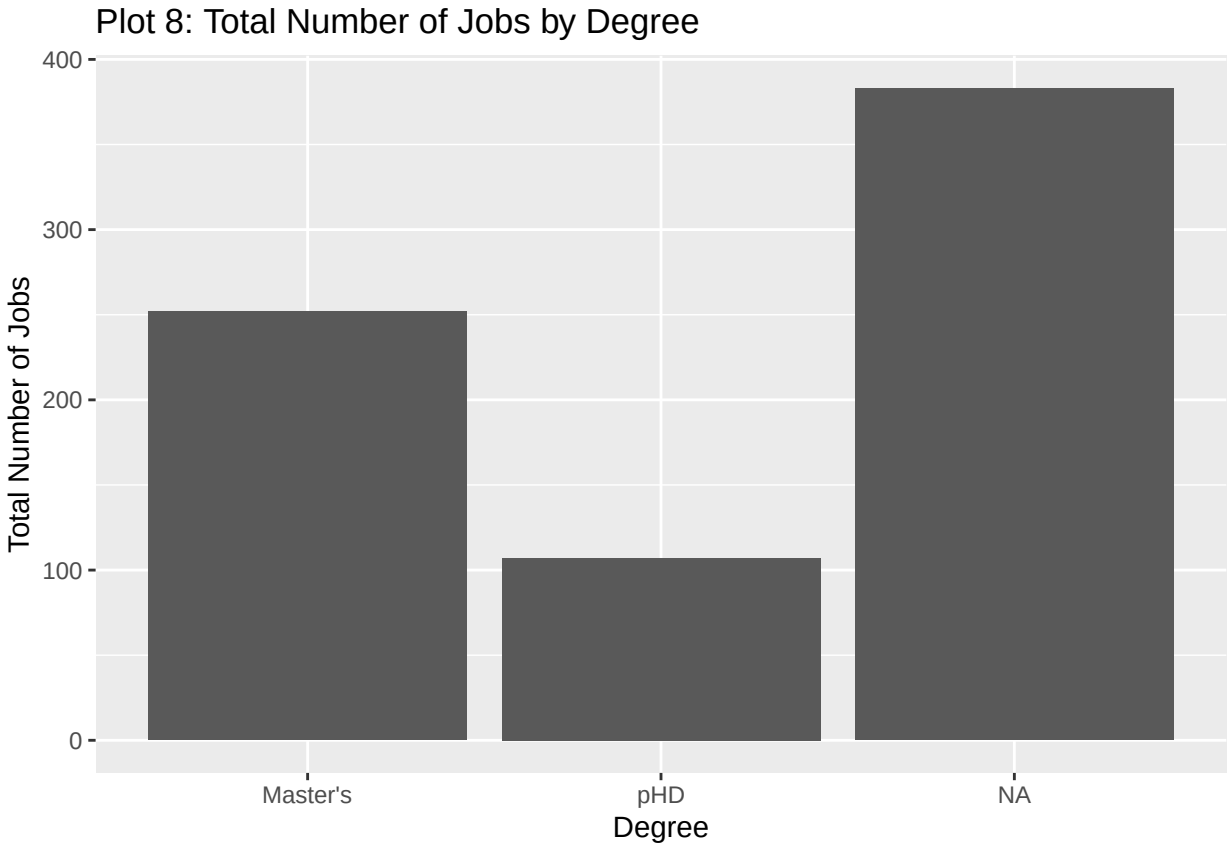
```
## Picking joint bandwidth of 9.87
```



We created quantile lines. Plot 6 shows the distribution of average salary by the range of company age. Generally, younger companies(11-20 years and 51-70 years) give more average salary. The others have too many variations to conclude anything.

```
education <- data %>%
  group_by(degree) %>%
  summarise(total_jobs = n())
```

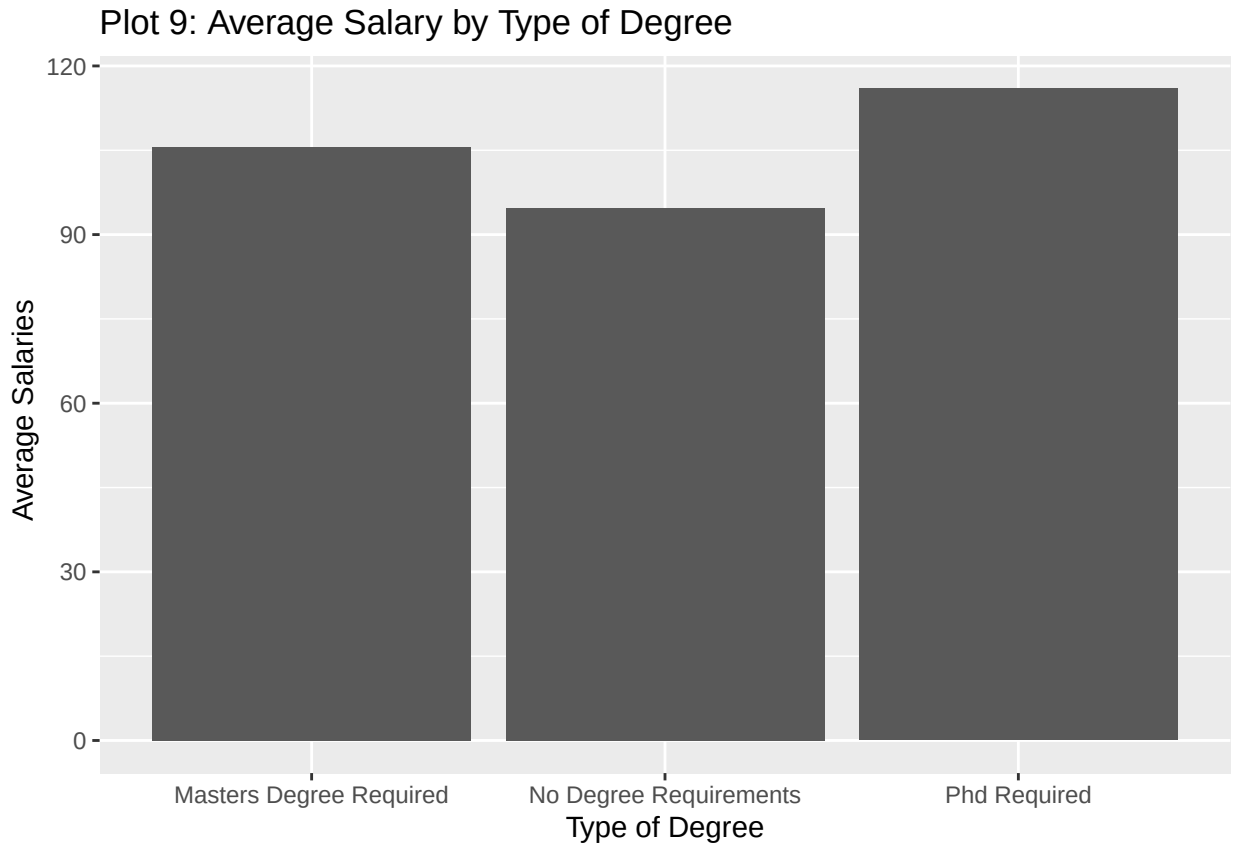
```
education <- education%>%
  mutate( degree = case_when(
    degree == "M" ~ "Master's",
    degree == "P" ~ "pHD",
    is.na(degree) ~ "na"
  ))
 ggplot(education, aes(x = degree, y= total_jobs)) +
geom_col() +
  labs(
    x = "Degree",
    y = "Total Number of Jobs",
    title = "Plot 8: Total Number of Jobs by Degree"
  )
```



In Plot 7, we wanted to find the relationship between the degree requirements of a job listed in the data and the number of jobs available. The degree column has three types of values, 'M' for Masters requirements, 'P' for Phd requirements and 'na' for no specific requirements. We found the number of jobs per each degree requirement and visualized it using a bar chart. We found that the number of jobs decreases as the education level requirement increases i.e there are higher numbers of jobs without any specific requirements, and lower number of jobs with Phd requirements.

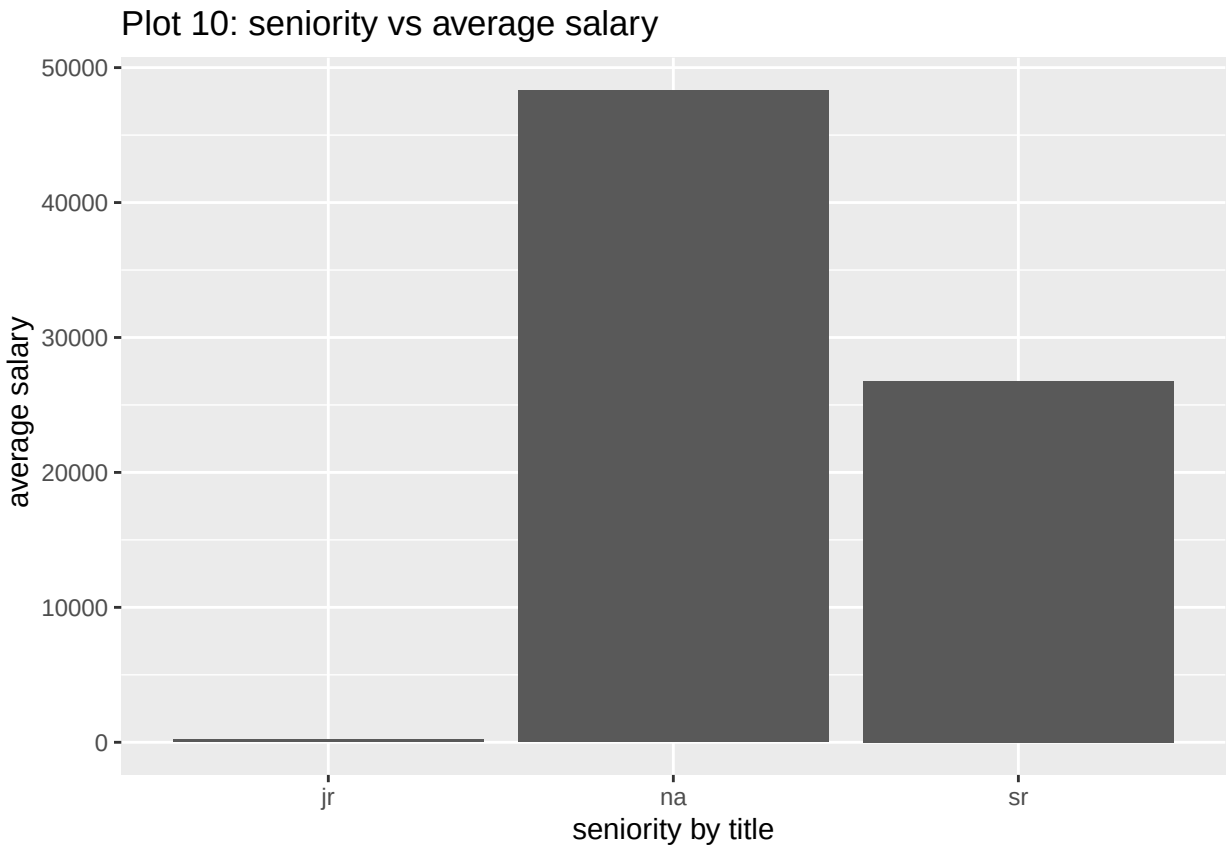
```
masters_salary <- data %>%
filter(degree == 'M')

phd_salary <- data %>%
filter(degree =='P')

Masters_average <- mean(masters_salary$avg_salary_k)

phd_average <- mean(phd_salary$avg_salary_k)

noDegree <- data %>%
filter(degree == 'na')

noDegree_average <- mean(noDegree$avg_salary_k)

average_salaries <- data.frame(Degree = c("No Degree Requirements", "Masters Degree Required", "Phd Requ

ggplot(average_salaries, aes(x = Degree, y= avg_salary))+
geom_col()+
  labs(
```

```
    x = "Type of Degree",
    y = " Average Salaries",
    title = "Plot 9: Average Salary by Type of Degree"
  )
```



In Plot 8, we wanted to find the relationship between the education level of a person and the salary earned by them. Our hypothesis was that for jobs with a higher education level, the salary offered is more than jobs without any specific requirements. For this, we used the degree and the average_salary column. The plot represents the average salaries as per the education level. We found the average salaries for every education level i.e. Masters, Phd, and no specific requirement and then visualized the data to find that as the education level requirement increases, the average salary also increases. This can be correlated to the expertise that a person with higher education level brings to the company and hence they are compensated more than jobs that do not require a higher education level.

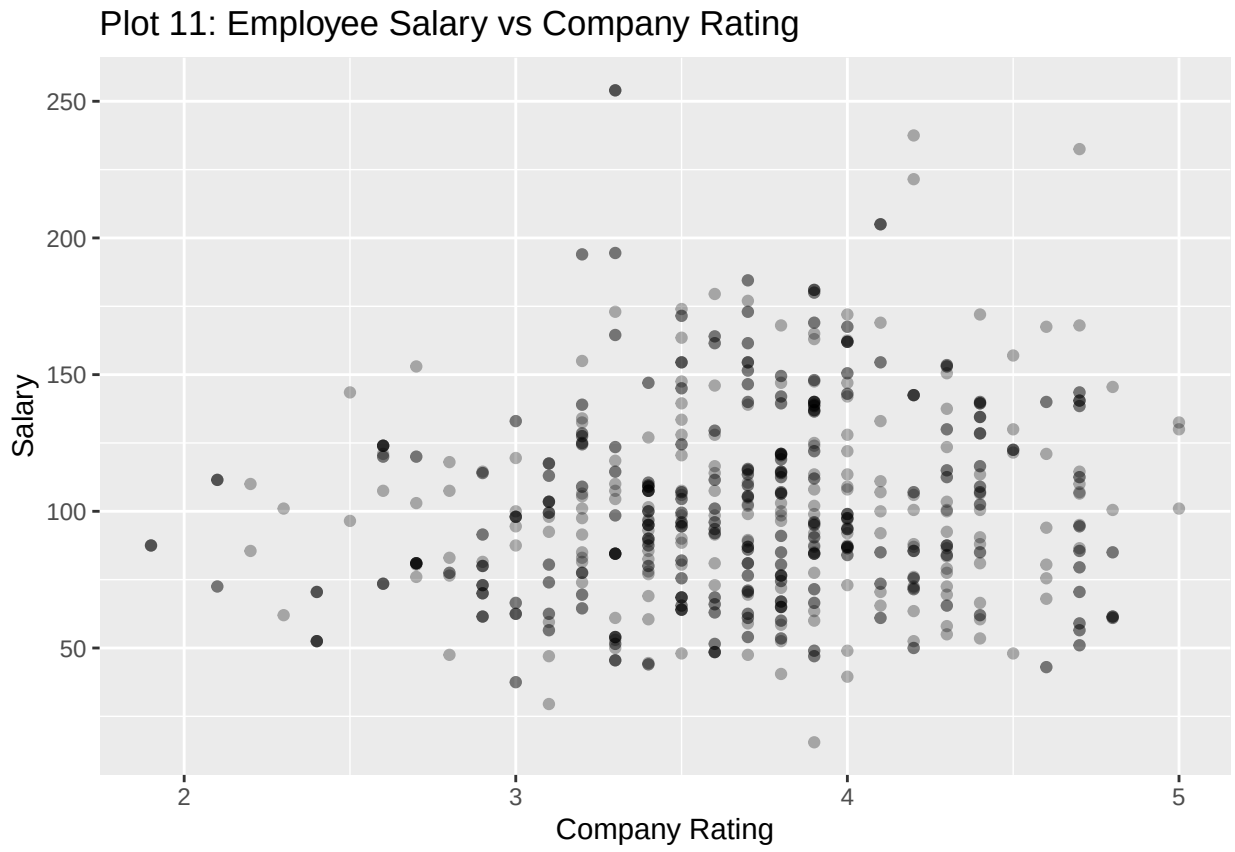
```
ggplot(data, aes(x = seniority_by_title, y = avg_salary_k)) +
  geom_col() +
  labs(
    x = "seniority by title",
    y = " average salary",
    title = "Plot 10: seniority vs average salary"
  )
```



Plot 10 shows the average salary by seniority. As we can see, seniors seem to have a higher average salary than juniors.

The following plot data is made from the original data. We keep only the relevant variables by using the select function. Then we filter the data to get rid of any undesirable observations.

```
plotdataT1 <- data %>%
  select(avg_salary_k, rating, hourly, job_location, size, revenue) %>%
  filter(hourly == 0, rating != -1, size != "unknown") %>%
  group_by(job_location)


ggplot(data = plotdataT1,
       aes(x = rating,
           y = avg_salary_k)) +
  geom_point(alpha = 0.3) +
  labs(title = "Plot 11: Employee Salary vs Company Rating",
       x = "Company Rating",
       y = "Salary")
```

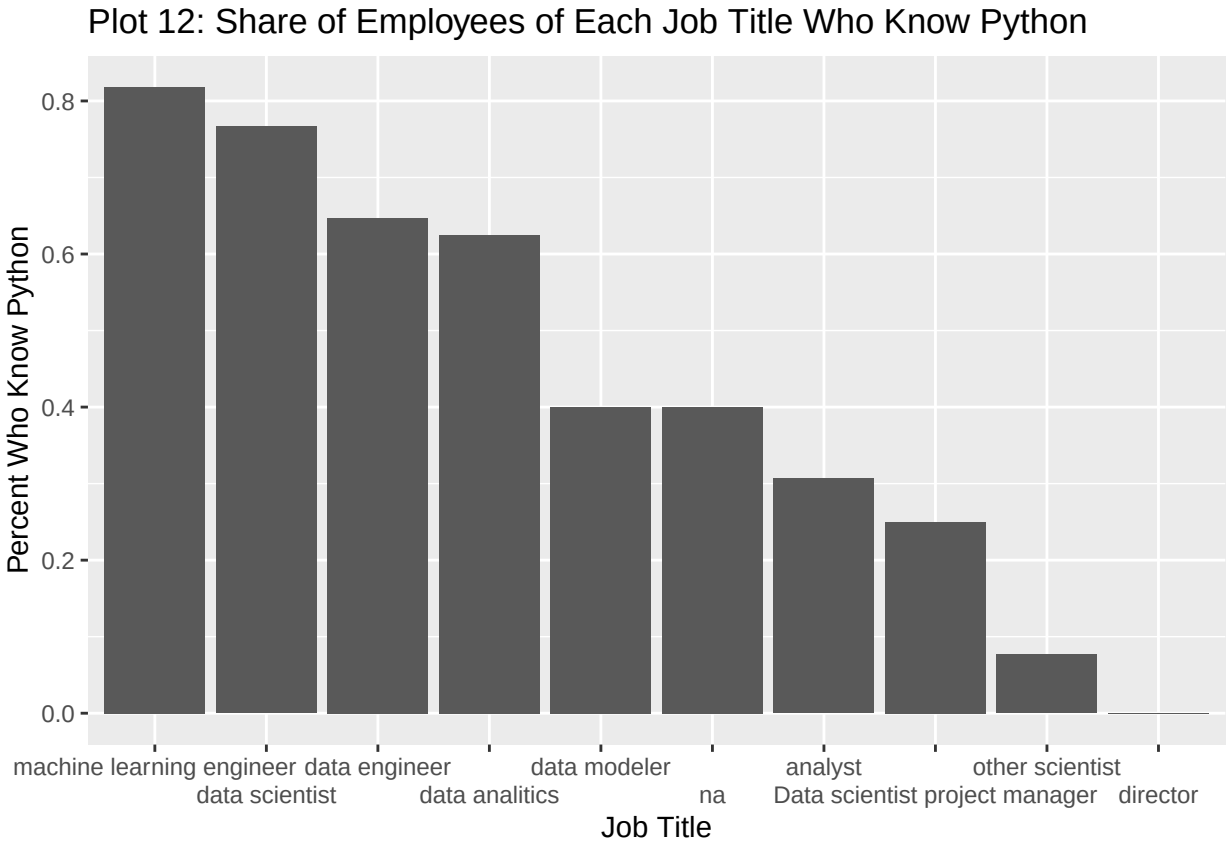


This plot is rather inconclusive as is as there doesn't seem to be much of a correlation simply between Company rating and employee salary.

We made these next four plots in particular because we thought that together, they might be useful for discerning which types of data scientist jobs are most coding intensive which may have implications on salary.

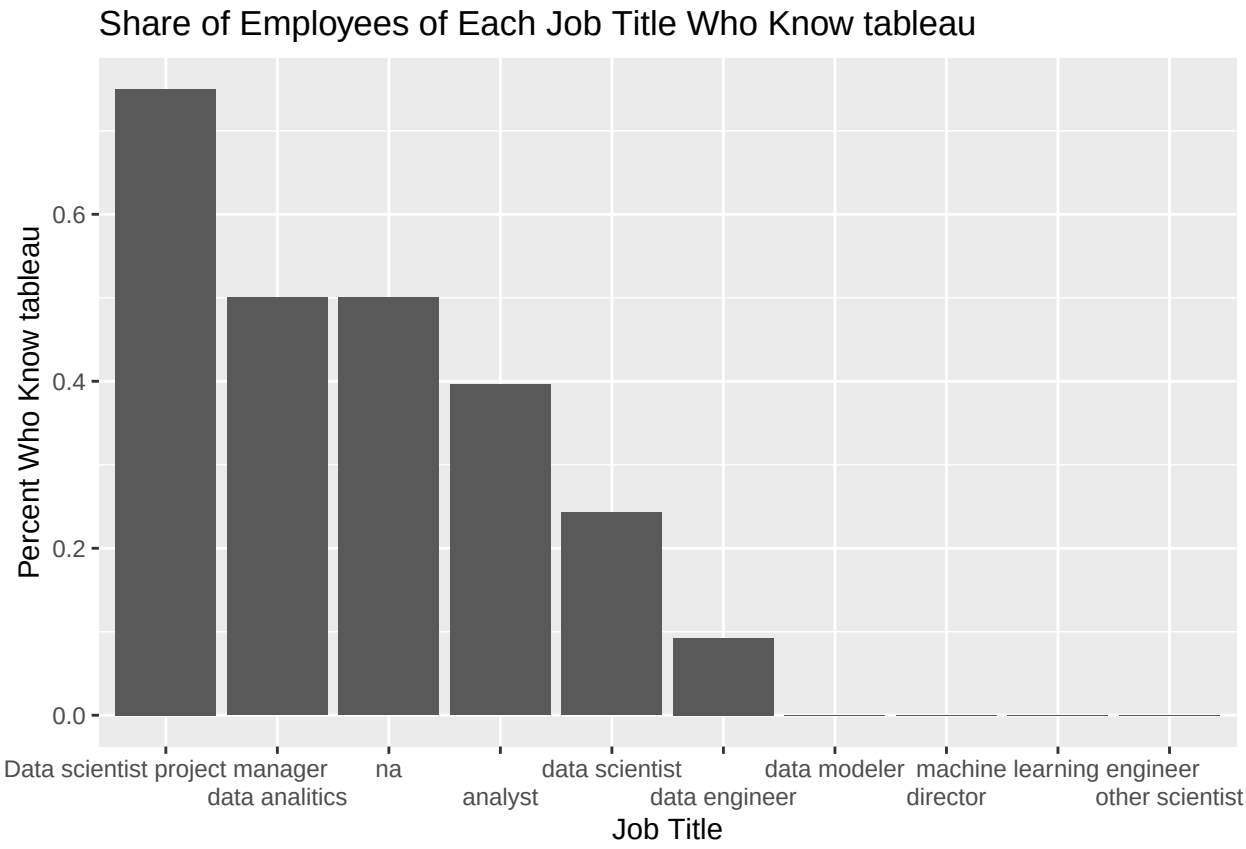
This plot data takes ony the variables for job location and python proficiency by piping in the select function. We summarize the data to show the percent of employees of each job type that know python. Then we reorder the job title column in terms of descending portion of those data scientists who know python so that the plot will be easier to read.

```
pythondata <- data %>%
  select(job_title_sim, python) %>%
  group_by(job_title_sim) %>%
  summarize(share_python = mean(python)) %>%
  mutate(job_title_sim = fct_reorder(job_title_sim, desc(share_python)))

ggplot(data = pythondata,
       aes(x = job_title_sim,
           y = share_python)) +
  geom_col() +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))+
  labs(title = "Plot 12: Share of Employees of Each Job Title Who Know Python",
       x = "Job Title",
       y = "Percent Who Know Python")
```



This plot data is very similar to the one made for python proficiency except it measures proficiency in the tableau language instead.

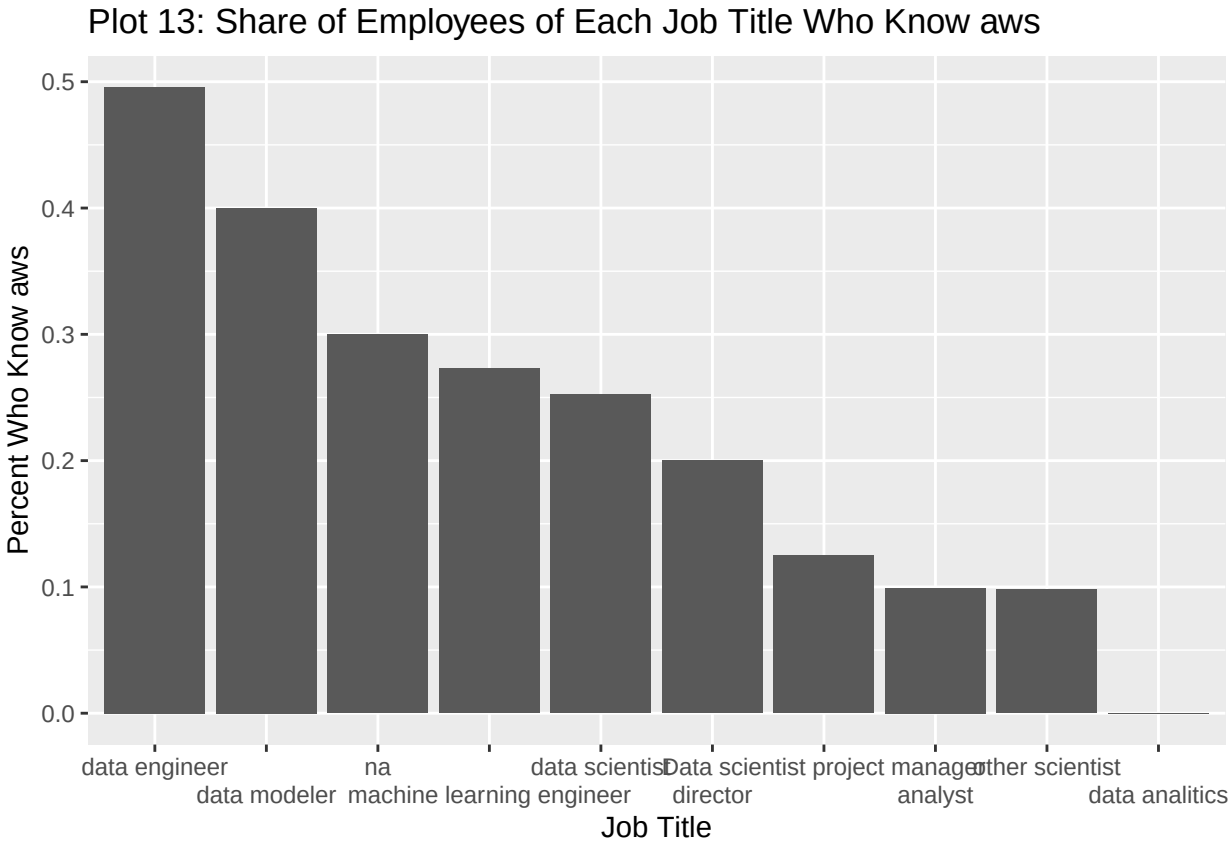
```
tableaudata <- data %>%
  select(job_title_sim, tableau) %>%
  group_by(job_title_sim) %>%
  summarize(share_tableau = mean(tableau)) %>%
  mutate(job_title_sim = fct_reorder(job_title_sim, desc(share_tableau)))

ggplot(data = tableaudata,
       aes(x = job_title_sim,
           y = share_tableau)) +
  geom_col() +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))+
  labs(title = "Share of Employees of Each Job Title Who Know tableau",
       x = "Job Title",
       y = "Percent Who Know tableau")
```



Interestingly, project managers seem to be the most likely to use tableau. The fact that these workers don't use python as much but use tableau much more suggests a fundamental difference in the jobs of project managers and other data scientists.

This plot data is very similar to the one made for python proficiency except it measures proficiency in the aws language instead.

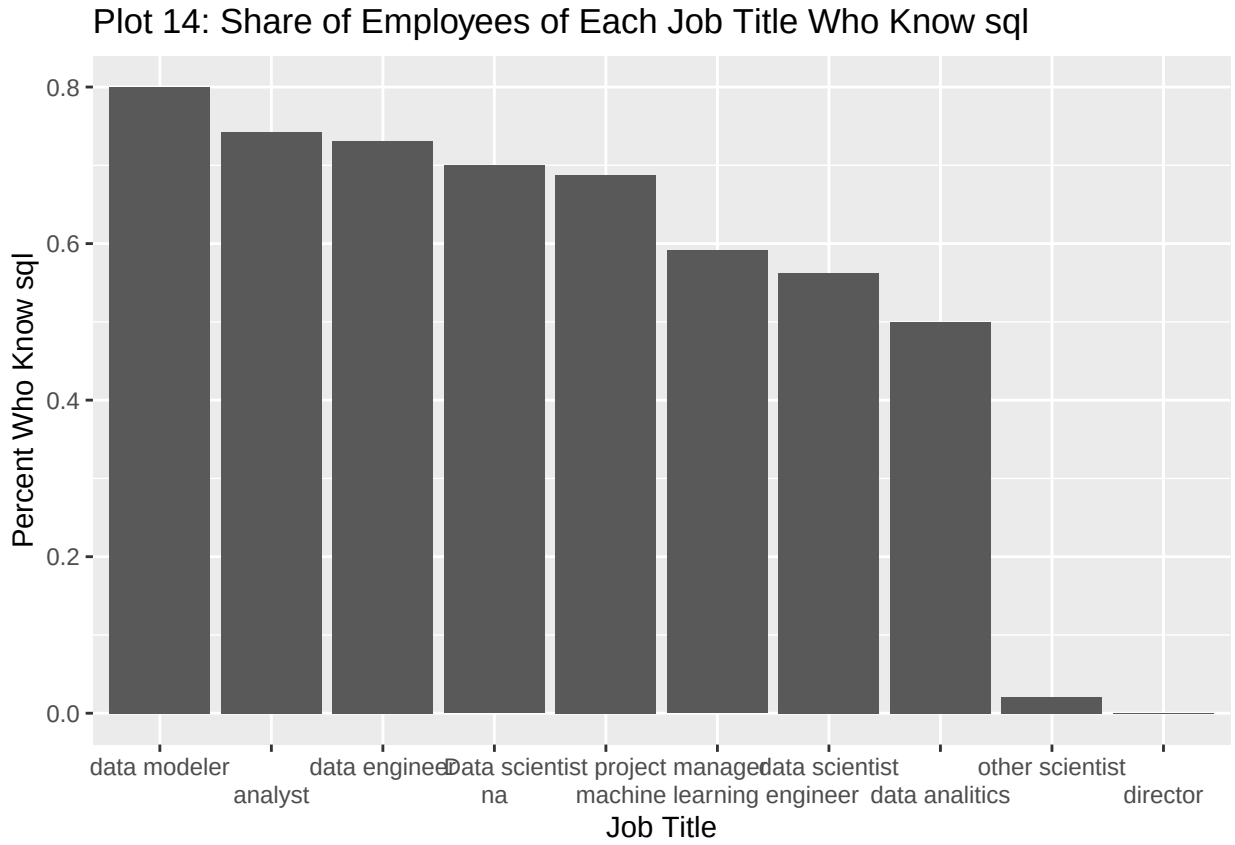
```
awsdata <- data %>%
  select(job_title_sim, aws) %>%
  group_by(job_title_sim) %>%
  summarize(share_aws = mean(aws)) %>%
  mutate(job_title_sim = fct_reorder(job_title_sim, desc(share_aws)))

ggplot(data = awsdata,
       aes(x = job_title_sim,
           y = share_aws)) +
  geom_col() +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))+
  labs(title = "Plot 13: Share of Employees of Each Job Title Who Know aws",
       x = "Job Title",
       y = "Percent Who Know aws")
```



Here we see data engineers and data modelers being the data scientists who more typically use aws. It is interesting that so far we have not seen the same job titles on the top or bottoms of these rankings. This suggests that different data scientist job types tend to use different languages more frequently but all job types typically use at least some language to a high frequency.

This plot data is very similar to the one made for python proficiency except it measures proficiency in the sql language instead.

```
sqldata <- data %>%
  select(job_title_sim, sql) %>%
  group_by(job_title_sim) %>%
  summarize(share_sql = mean(sql)) %>%
  mutate(job_title_sim = fct_reorder(job_title_sim, desc(share_sql)))

ggplot(data = sqldata,
       aes(x = job_title_sim,
           y = share_sql)) +
  geom_col() +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))+
  labs(title = "Plot 14: Share of Employees of Each Job Title Who Know sql",
       x = "Job Title",
       y = "Percent Who Know sql")
```



As we would expect sql is used relatively commonly throuout almost all data scientist job types. This reaffirms that all data scientist jobs have high rates of proficiency in at least some languages.

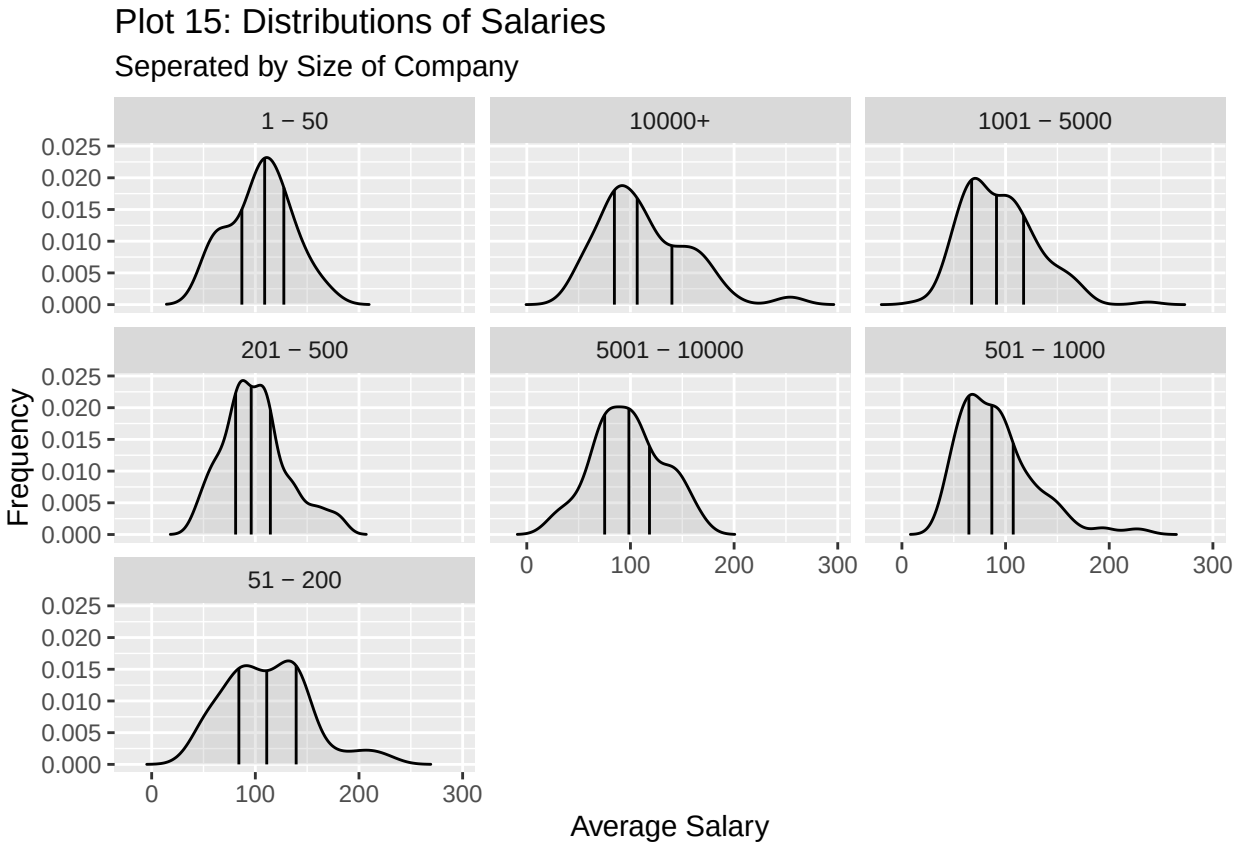
This plotdata selects only the variables that measure salary and job title from the original data. Then we filter the data so it will not include any observations for which the size of the company that the data scientist works for is unknown.

```
plotdataT2 <- data %>%
  select(avg_salary_k, size, job_title_sim) %>%
  filter(size != "unknown")

ggplot(data = plotdataT2,
       aes(x = avg_salary_k,
           y = 0)) +
  geom_density_ridges(alpha = 0.3,
                      quantile_lines = TRUE,
                      quantiles = c(0.25, 0.5,0.75)) +
  facet_wrap(vars(size)) +
  labs(title = "Plot 15: Distributions of Salaries",
       subtitle = "Seperated by Size of Company",
       x = "Average Salary",
       y = "Frequency")
```

```
## Picking joint bandwidth of 13.6
```

```
## Picking joint bandwidth of 14.1
```

```
## Picking joint bandwidth of 11.8

## Picking joint bandwidth of 8.68

## Picking joint bandwidth of 12.2

## Picking joint bandwidth of 10.8

## Picking joint bandwidth of 14.8
```



From these plots, we see that for any size of company, the distribution of the salaries of their employees seems to be roughly the same, with the median being very close to $100,000 in all cases.