Final_Project_USAJOBs_ETL

Maxwell Miller-Golub

2024-12-13

Step 2: Clean

2a) Combine data tables and remove duplicate jobs

```
Data_Scientist_Data_Set <- read_csv("Individual_Scrape_CSVs/Data_Scientist_Data_Set.csv")
```

```
## Rows: 327 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (17): Keyword, Full URL, Title, Agency, Pay scale & grade, Remote job, ...
## dbl   (3): Job Code, salary_min, salary_max
## date  (1): Date Accessed
##
## 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.
```

```
Data_Analyst_Data_Set <- read_csv("Individual_Scrape_CSVs/Data_Analyst_Data_Set.csv")
```

```
## Rows: 1127 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (17): Keyword, Full URL, Title, Agency, Pay scale & grade, Remote job, ...
## dbl   (3): Job Code, salary_min, salary_max
## date  (1): Date Accessed
##
## 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.
```

```
Data_Engineer_Data_Set <- read_csv("Individual_Scrape_CSVs/Data_Engineer_Data_Set.csv")
```

```
## Rows: 1030 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (17): Keyword, Full URL, Title, Agency, Pay scale & grade, Remote job, ...
## dbl   (3): Job Code, salary_min, salary_max
## date  (1): Date Accessed
##
## 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.
```

```r
df2 <- rbind(Data_Analyst_Data_Set, Data_Engineer_Data_Set)
full_data_with_dups <- rbind(df2, Data_Scientist_Data_Set)

# Remove duplicates, keeping information from the unique column
df_cleaned <- full_data_with_dups %>%
  group_by(`Job Code`, `Date Accessed`, `Full URL`, Title, Agency, `Pay scale & grade`, `Remote job`, `
  summarize(
    Keyword = paste(unique(Keyword), collapse = ", "),  # Combine the unique_column values
    .groups = "drop"  # Remove grouping structure
  )

#write_csv(df_cleaned, "Full_Clean_Data_Jobs_Dataset.csv")
```

2b) Build stopwords out to clean "Qualifications" more

```r
# Edit the list of words to remove from Qualifications
numbers <- c("one", "two", "three", "four", "five", "six", "seven", "eight",
             "nine", "ten")
removal_words <- stopwords::stopwords("en")
removal_words <- append(removal_words, "qualification")
removal_words <- append(removal_words, "qualifications")
removal_words <- append(removal_words, "")
removal_words <- append(removal_words, letters)
removal_words <- append(removal_words, numbers)
```

2c) Telework, Travel, Schedule, Remote, Relocation Columns Cleaned and Combined

```r
Full_Clean_Data_Jobs_Dataset <- read_csv("various_output_files/Full_Clean_Data_Jobs_Dataset.csv")
```

```
## Rows: 2185 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (17): Full URL, Title, Agency, Pay scale & grade, Remote job, Telework ...
## dbl   (3): Job Code, salary_min, salary_max
## date  (1): Date Accessed
##
## 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.
```

```r
cleaned_five_columns <- Full_Clean_Data_Jobs_Dataset %>%
  mutate(`Telework eligible` = case_when(
    str_sub(`Telework eligible`, 1, 3) == "Yes" ~ "Yes",
    `Telework eligible` == "No" ~ "No",
    `Telework eligible` == "Not applicable, this is a remote position." ~ "N/A (Remote Position)",
    TRUE ~ "Unspecified"
  )) %>%
  # Removes 80 cases of the 2100 where the data scraped incorrectly
  filter(`Telework eligible` != "Unspecified") %>%
  mutate(`Travel Required` = case_when(
```

```
    str_starts(`Travel Required`, "25% or less") |
      str_starts(`Travel Required`, "Occasional travel") ~ "<= 25%",
    str_sub(`Travel Required`, 1, 11) == "50% or less" ~ "<= 50%",
    str_sub(`Travel Required`, 1, 11) == "75% or less" |
      str_sub(`Travel Required`, 1, 14) == "76% or greater" ~ "> 50%",
    `Travel Required` == "Not required" ~ "Not Required",
    TRUE ~ "Unspecified")) %>%
  filter(`Travel Required` != "Unspecified") %>%
   mutate(`Work schedule` = case_when(
    str_starts(`Work schedule`, "Full-Time") |
      str_starts(`Work schedule`, "Full-time") ~ "Full-time",
    str_starts(`Work schedule`, "Multiple Schedules") ~ "Multiple Schedules (Schedules may vary dependi
    str_starts(`Work schedule`, "Part-time") ~ "Part-time",
    `Work schedule` == "Intermittent" ~ "Intermittent",
    TRUE ~ "Unspecified")) %>%
  mutate(`Remote job` = case_when(
    str_sub(`Remote job`, 1, 3) == "Yes" ~ "Yes",
    `Remote job` == "No" ~ "No",
    TRUE ~ "Unspecified"
  )) %>%
  mutate(`Relocation expenses reimbursed` = case_when(
    str_sub(`Relocation expenses reimbursed`, 1, 3) == "Yes" ~ "Yes",
    `Relocation expenses reimbursed` == "No" ~ "No",
    TRUE ~ "Unspecified"
  ))
```

2d) Jobs with hourly wages are converted to salaries (based on 40hrs/52wks)

```
cleaned_seven_columns <- cleaned_five_columns %>%
  mutate(salary_min = case_when(
    salary_min < 54 & salary_min > 2 ~ salary_min*40*52,
    TRUE ~ salary_min
    )) %>%
  mutate(salary_max = case_when(
    salary_max < 54 ~ salary_max*40*52,
    TRUE ~ salary_max
    ))

head(cleaned_seven_columns)
```

```
## # A tibble: 6 x 21
##   `Job Code` `Date Accessed` `Full URL`         Title Agency `Pay scale & grade`
##        <dbl> <date>          <chr>              <chr> <chr>  <chr>
## 1  642735900 2024-12-08      https://www.usajo~ INTE~ Depar~ GS 13
## 2  669932000 2024-12-08      https://www.usajo~ FIRE~ Depar~ GS 11 - 12
## 3  673224000 2024-12-08      https://www.usajo~ SUPE~ Depar~ GS 13
## 4  675452900 2024-12-08      https://www.usajo~ ENGI~ Depar~ GS 11
## 5  692407800 2024-12-08      https://www.usajo~ Civi~ Depar~ GS 13
## 6  693894500 2024-12-08      https://www.usajo~ Elec~ Depar~ GS 11 - 12
## # i 15 more variables: `Remote job` <chr>, `Telework eligible` <chr>,
## #   `Travel Required` <chr>, `Relocation expenses reimbursed` <chr>,
```

```
## #   `Appointment type` <chr>, `Work schedule` <chr>, `Hiring Process` <chr>,
## #   `Promotion Potential` <chr>, `Supervisory Status` <chr>,
## #   `Security Clearance` <chr>, `Drug Test` <chr>, salary_min <dbl>,
## #   salary_max <dbl>, Qualifications <chr>, Keyword <chr>
```

2e) Function to clean qualifications -> create lists of words

```
shrink_qualifications <- function(sample_qualification){
  sample_qualification <- str_replace_all(sample_qualification, "\n", " ")
  sample_qualification <- strsplit(sample_qualification, " ")
  sample_qualification <- lapply(sample_qualification, function(s) s[nchar(s) <= 25])
  sample_qualification <- lapply(sample_qualification, function(s) gsub("[^[:alnum:]]", "", s))
  sample_qualification <- lapply(sample_qualification, tolower)
  sample_qualification <- sample_qualification[[1]]
  sample_qualification <- base::setdiff(sample_qualification, removal_words)
  sample_qualification <- lapply(sample_qualification, function(x) if (!grepl("\\d", x)) x else NULL)
sample_qualification <- base::Filter(base::Negate(is.null), sample_qualification)
sample_qualification <- unlist(sample_qualification)
sample_qualification <- list(sample_qualification)
sample_qualification <- sapply(sample_qualification, function(x) paste(x, collapse = " "))
sample_qualification <- as.character(sample_qualification)
return(sample_qualification)
}
```

2f) Clean Qualifications and Export to CSV and JSON

```
jobs_tibble <- as_tibble(cleaned_seven_columns)

# Splitting data here into different outputs. Cleaning qualifications for the 80 jobs that match on all

all_3 <- jobs_tibble %>%
  filter(Keyword == "Data Analyst, Data Engineer, Data Scientist") %>%
  mutate(Reduced_Qualifications = map(Qualifications, shrink_qualifications))

head(all_3)
```

```
## # A tibble: 6 x 22
##   `Job Code` `Date Accessed` `Full URL`         Title Agency `Pay scale & grade`
##        <dbl> <date>          <chr>              <chr> <chr>  <chr>
## 1  757695300 2024-12-08      https://www.usajo~ DST ~ Centr~ GS 8 - 15
## 2  757698900 2024-12-08      https://www.usajo~ DST ~ Centr~ GS 8 - 15
## 3  759328900 2024-12-08      https://www.usajo~ Data~ Centr~ GS 9 - 13
## 4  778678800 2024-12-08      https://www.usajo~ Math~ Depar~ GS 11 - 14
## 5  780005100 2024-12-08      https://www.usajo~ IT S~ Depar~ GS 9
## 6  780196200 2024-12-08      https://www.usajo~ Subj~ Depar~ GS 13 - 14
## # i 16 more variables: `Remote job` <chr>, `Telework eligible` <chr>,
## #   `Travel Required` <chr>, `Relocation expenses reimbursed` <chr>,
## #   `Appointment type` <chr>, `Work schedule` <chr>, `Hiring Process` <chr>,
## #   `Promotion Potential` <chr>, `Supervisory Status` <chr>,
## #   `Security Clearance` <chr>, `Drug Test` <chr>, salary_min <dbl>,
```

```
## #   salary_max <dbl>, Qualifications <chr>, Keyword <chr>,
## #   Reduced_Qualifications <list>
```

```
all_data_for_json <- jobs_tibble %>%
  mutate(Reduced_Qualifications = map(Qualifications, shrink_qualifications))

head(all_data_for_json)
```

```
## # A tibble: 6 x 22
##   `Job Code` `Date Accessed` `Full URL`         Title Agency `Pay scale & grade`
##        <dbl> <date>          <chr>              <chr> <chr>  <chr>
## 1  642735900 2024-12-08      https://www.usajo~ INTE~ Depar~ GS 13
## 2  669932000 2024-12-08      https://www.usajo~ FIRE~ Depar~ GS 11 - 12
## 3  673224000 2024-12-08      https://www.usajo~ SUPE~ Depar~ GS 13
## 4  675452900 2024-12-08      https://www.usajo~ ENGI~ Depar~ GS 11
## 5  692407800 2024-12-08      https://www.usajo~ Civi~ Depar~ GS 13
## 6  693894500 2024-12-08      https://www.usajo~ Elec~ Depar~ GS 11 - 12
## # i 16 more variables: `Remote job` <chr>, `Telework eligible` <chr>,
## #   `Travel Required` <chr>, `Relocation expenses reimbursed` <chr>,
## #   `Appointment type` <chr>, `Work schedule` <chr>, `Hiring Process` <chr>,
## #   `Promotion Potential` <chr>, `Supervisory Status` <chr>,
## #   `Security Clearance` <chr>, `Drug Test` <chr>, salary_min <dbl>,
## #   salary_max <dbl>, Qualifications <chr>, Keyword <chr>,
## #   Reduced_Qualifications <list>
```

```
clear_qualifications <- jobs_tibble %>%
  select(-Qualifications)

head(clear_qualifications)
```

```
## # A tibble: 6 x 20
##   `Job Code` `Date Accessed` `Full URL`         Title Agency `Pay scale & grade`
##        <dbl> <date>          <chr>              <chr> <chr>  <chr>
## 1  642735900 2024-12-08      https://www.usajo~ INTE~ Depar~ GS 13
## 2  669932000 2024-12-08      https://www.usajo~ FIRE~ Depar~ GS 11 - 12
## 3  673224000 2024-12-08      https://www.usajo~ SUPE~ Depar~ GS 13
## 4  675452900 2024-12-08      https://www.usajo~ ENGI~ Depar~ GS 11
## 5  692407800 2024-12-08      https://www.usajo~ Civi~ Depar~ GS 13
## 6  693894500 2024-12-08      https://www.usajo~ Elec~ Depar~ GS 11 - 12
## # i 14 more variables: `Remote job` <chr>, `Telework eligible` <chr>,
## #   `Travel Required` <chr>, `Relocation expenses reimbursed` <chr>,
## #   `Appointment type` <chr>, `Work schedule` <chr>, `Hiring Process` <chr>,
## #   `Promotion Potential` <chr>, `Supervisory Status` <chr>,
## #   `Security Clearance` <chr>, `Drug Test` <chr>, salary_min <dbl>,
## #   salary_max <dbl>, Keyword <chr>
```

```
#write_json(all_3, "Final_Data_Science_Only_Jobs_Dataset.json")
#write_json(all_data_for_json, "Final_Full_Dataset.json")
#write_csv(all_3, "Only_Definite_Data_Science_Jobs_Dataset.csv")
#write_csv(clear_qualifications, "Data_Jobs_Dataset_Without_Qualifications.csv")
```

Step 3: Analysis

3a) Prepping both job groups

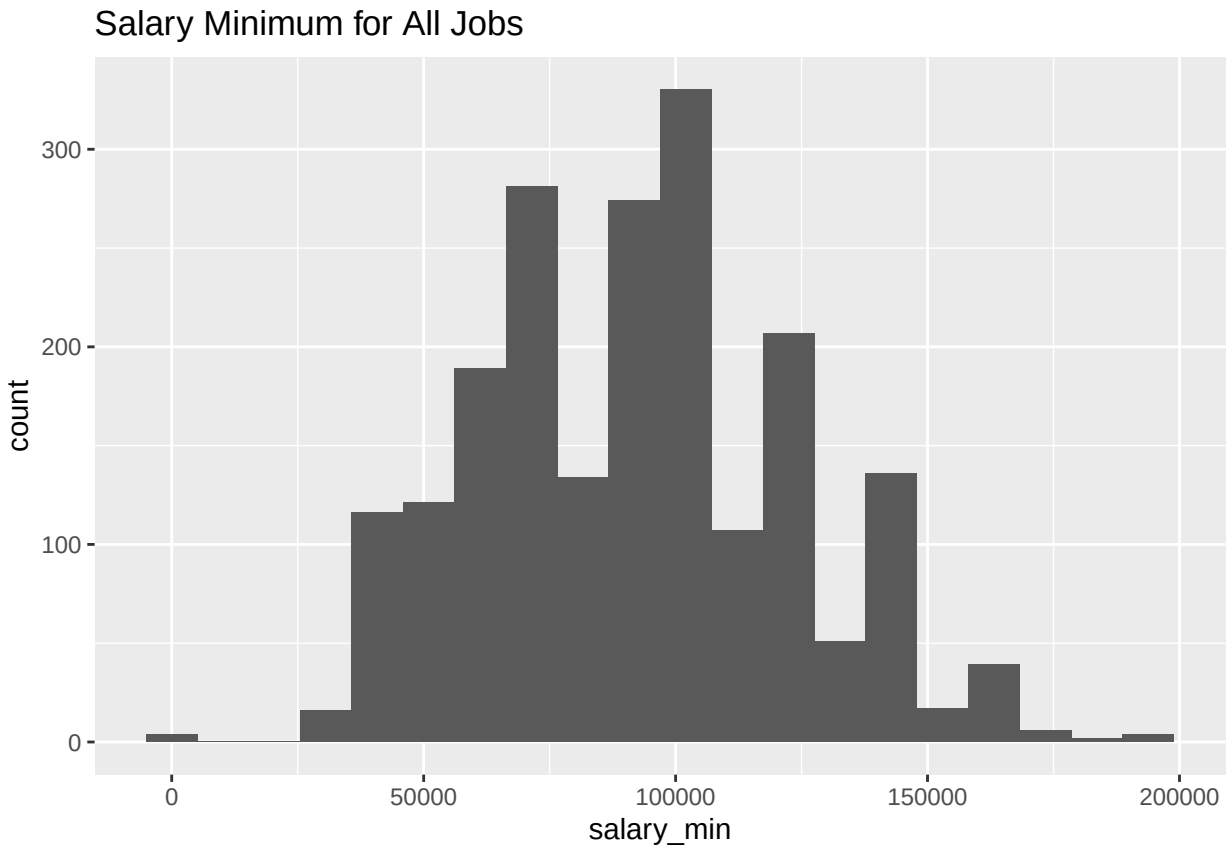
```
main_3 <- jobs_tibble %>%
  filter(Keyword == "Data Analyst, Data Engineer, Data Scientist") %>%
  mutate(Reduced_Qualifications = map(Qualifications, shrink_qualifications))

the_rest <- jobs_tibble %>%
  filter(Keyword != "Data Analyst, Data Engineer, Data Scientist") %>%
  mutate(Reduced_Qualifications = map(Qualifications, shrink_qualifications))
```

3b) Salary Expectations

```
#RQ2: What salaries can be expected in the field?

#Salary Minimum for All Jobs
summary(the_rest$salary_min)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0   69596   88520   91406  112064  193819
```

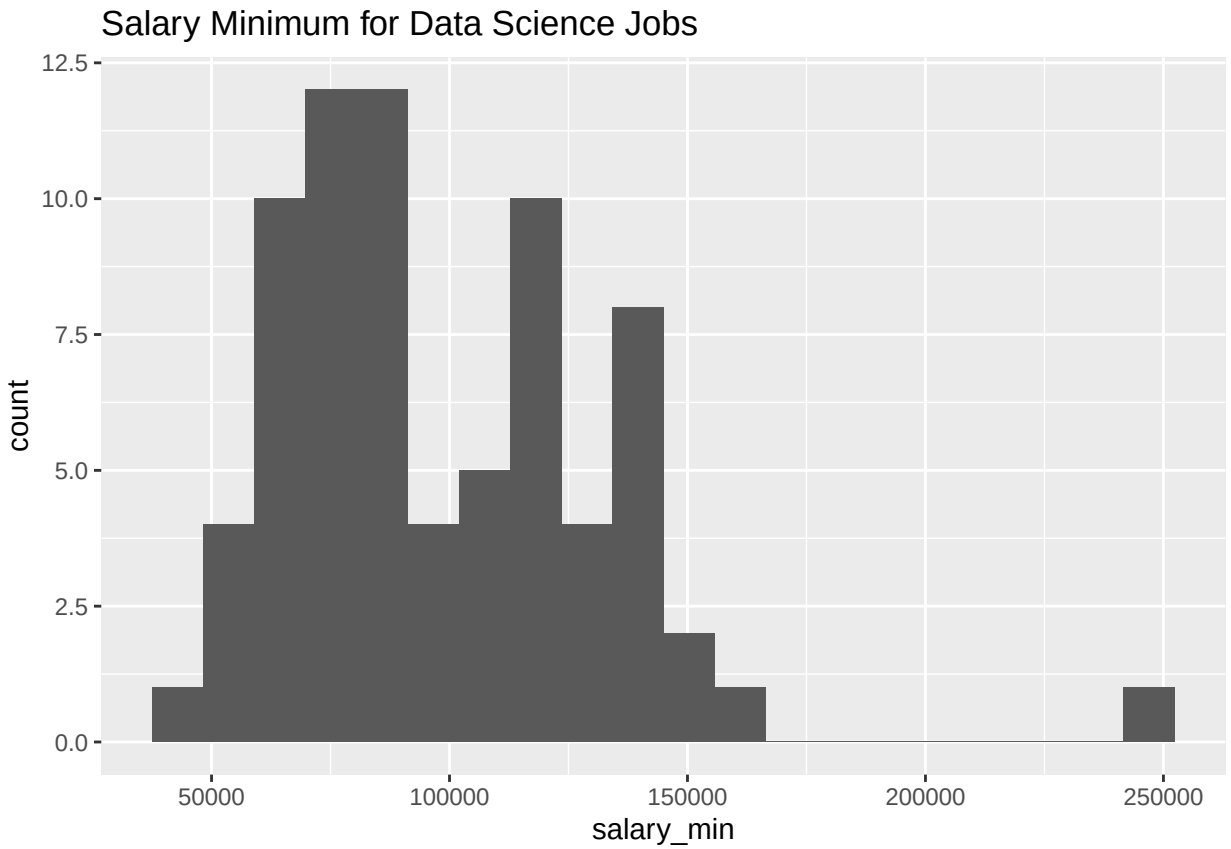
```
ggplot(data = the_rest) +
  geom_histogram(aes(x = salary_min), bins = 20)+
  ggtitle("Salary Minimum for All Jobs")
```




```
#Salary Minimum for Just Data Science Jobs
summary(all_3$salary_min)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46020   72553   90310   99110  122198  250000
```

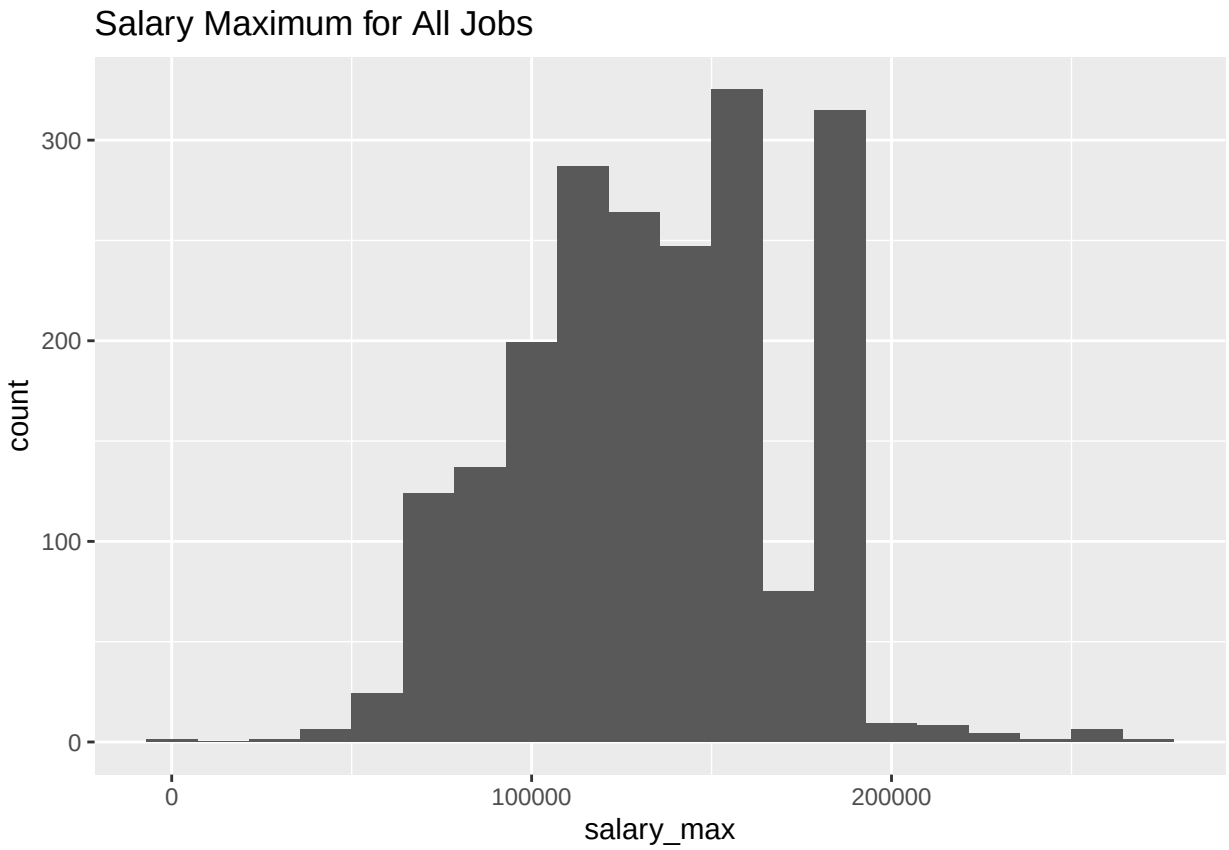
```
ggplot(data = all_3) +
  geom_histogram(aes(x = salary_min), bins = 20)+
  ggtitle("Salary Minimum for Data Science Jobs")
```




```
#Salary Maximum for All Jobs
summary(the_rest$salary_max)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0  107590  134435  133547  153442  271360
```

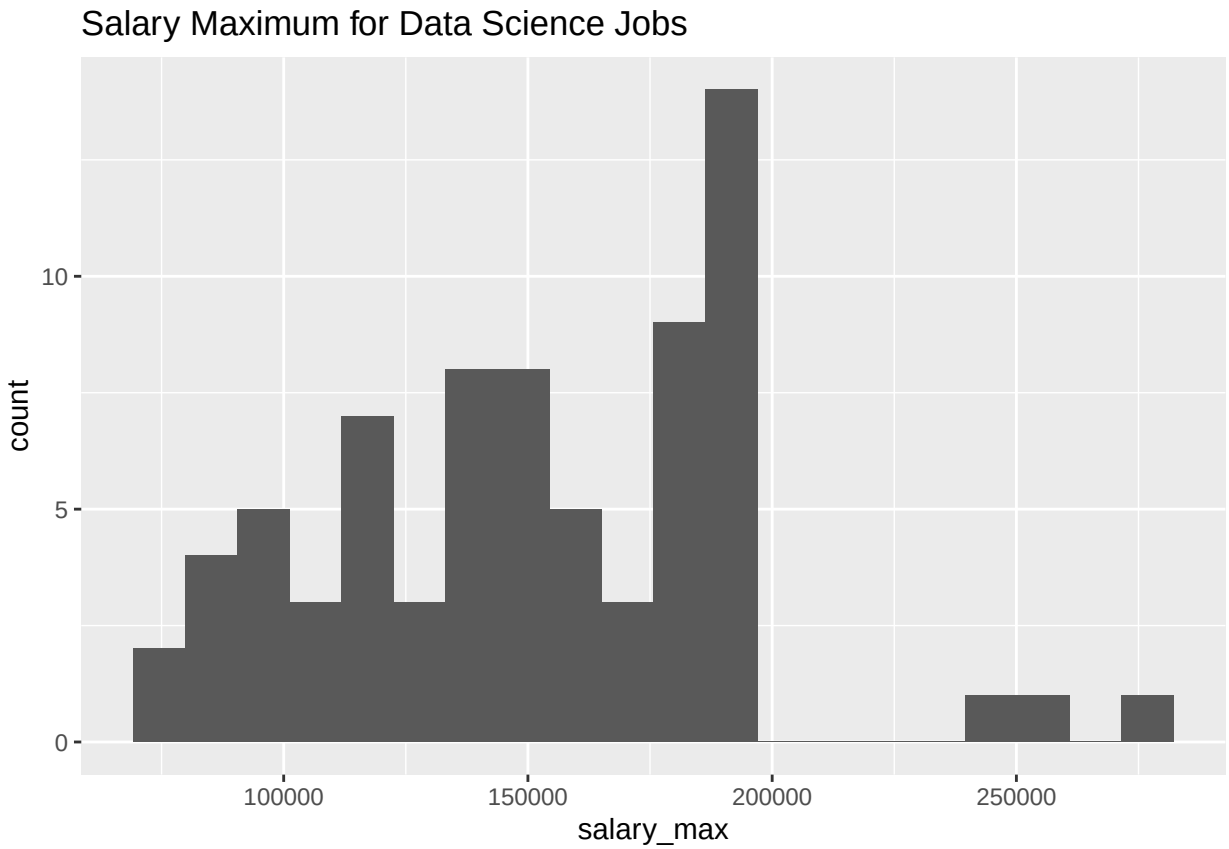
```
ggplot(data = the_rest) +
  geom_histogram(aes(x = salary_max), bins = 20)+
  ggtitle("Salary Maximum for All Jobs")
```




```
#Salary Maximum for Just Data Science Jobs
summary(all_3$salary_max)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   72703  117786  153354  151139  181216  275000
```

```
ggplot(data = all_3) +
  geom_histogram(aes(x = salary_max), bins = 20)+
  ggtitle("Salary Maximum for Data Science Jobs")
```




```
#mean(all_3$salary_min)
#sd(all_3$salary_min)
# n = 74

#mean(the_rest$salary_min)
#sd(the_rest$salary_min)
# n = 2034

#mean(all_3$salary_max)
#sd(all_3$salary_max)
# n = 74

#mean(the_rest$salary_max)
#sd(the_rest$salary_max)
# n = 2034
```

3c) Remote/Telework

```
#RQ4: Are jobs/careers still operating remotely (since COVID)?
the_rest %>%
  group_by(`Telework eligible`) %>%
  count(`Telework eligible`)
```

```
## # A tibble: 3 x 2
```

```
## # Groups:   Telework eligible [3]
##   `Telework eligible`       n
##   <chr>                 <int>
## 1 N/A (Remote Position)    73
## 2 No                      355
## 3 Yes                    1606
```

```
all_3 %>%
  group_by(`Telework eligible`) %>%
  count(`Telework eligible`)
```

```
## # A tibble: 3 x 2
## # Groups:   Telework eligible [3]
##   `Telework eligible`       n
##   <chr>                 <int>
## 1 N/A (Remote Position)     7
## 2 No                       13
## 3 Yes                      54
```

```
#(73/2034)*100
#(355/2034)*100
#(1606/2034)*100

#(7/74)*100
#(13/74)*100
#(54/74)*100
```

3d) Agencies hiring Data Scientists

```
#RQ5: Who's hiring Data Scientists?
the_rest %>%
  group_by(Agency) %>%
  count(Agency)
```

```
## # A tibble: 49 x 2
## # Groups:   Agency [49]
##    Agency                                                  n
##    <chr>                                               <int>
##  1 AmeriCorps                                              1
##  2 Central Intelligence Agency                            23
##  3 Chemical Safety and Hazard Investigation Board          2
##  4 Court Services and Offender Supervision Agency for DC   2
##  5 Defense Nuclear Facilities Safety Board                 2
##  6 Department of Agriculture                              86
##  7 Department of Commerce                                 47
##  8 Department of Defense                                 112
##  9 Department of Education                                 2
## 10 Department of Energy                                   47
## # i 39 more rows
```

```r
all_3 %>%
  group_by(Agency) %>%
  count(Agency)
```

```
## # A tibble: 22 x 2
## # Groups:   Agency [22]
##    Agency                                    n
##    <chr>                                 <int>
##  1 Central Intelligence Agency               3
##  2 Department of Agriculture                 1
##  3 Department of Commerce                    2
##  4 Department of Defense                     3
##  5 Department of Energy                      1
##  6 Department of Health and Human Services     9
##  7 Department of Homeland Security           5
##  8 Department of Justice                     2
##  9 Department of Transportation              4
## 10 Department of Veterans Affairs            6
## # i 12 more rows
```

4: Extra

4a) Glossary of Terms from USAJOBs

```r
#Glossary:
# pay scale and grade: A grade refers to the pay scale which sets the pay level and qualifications for

# Telework elligible: Determines if you will be able to work from home on some days.
# travel required: The amount of travel the job requires.

# relocation expenses reimbursed: Whether or not you will be reimbursed for relocation expenses.

# appointment type: The way that the Federal Government classifies the duration of certain jobs.

# work schedule: Determines the number of hours that you will work during the week.

#hiring process: The Federal Government has three services that determine how you are hired: Competitiv

#"Promotion Potential": Determines if you can move up to the next grade within your pay scale.

# supervisory status: Determines if you will be a supervisor.

# security clearance: The level of security clearance required to hold this position.

# Drug Test: Whether or not you will be tested for illegal drug use.
```

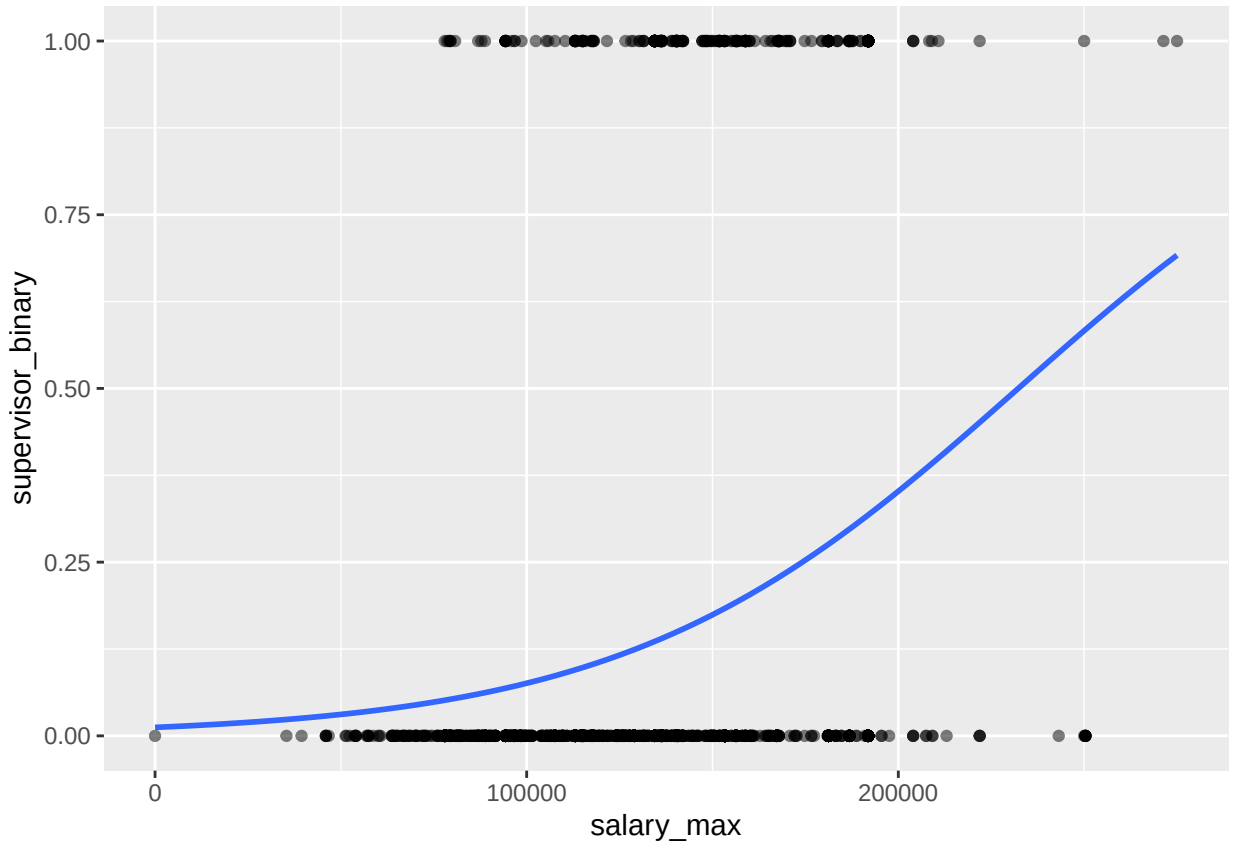
4b) Logistic Regression of Supervisors and Salary

```r
#RQ: Is there a relationship between salary and supervisor positions?

supervisor_data <- cleaned_seven_columns %>%
  select(`Supervisory Status`, salary_min, salary_max) %>%
  mutate(supervisor_binary = case_when(
    `Supervisory Status` == "No" ~ 0,
    `Supervisory Status` == "Yes" ~ 1
  ))

glm(supervisor_binary ~ salary_min + salary_max, family = "binomial", data = supervisor_data) -> supervi
summary(supervisor_model)
```

```
##
## Call:
## glm(formula = supervisor_binary ~ salary_min + salary_max, family = "binomial",
##     data = supervisor_data)
##
## Coefficients:
##                 Estimate   Std. Error z value             Pr(>|z|)
## (Intercept) -4.501972257  0.293495269 -15.339 < 0.0000000000000002 ***
## salary_min   0.000055861  0.000005425  10.298 < 0.0000000000000002 ***
## salary_max  -0.000020325  0.000004429  -4.589           0.00000446 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1819.2  on 2107  degrees of freedom
## Residual deviance: 1523.1  on 2105  degrees of freedom
## AIC: 1529.1
##
## Number of Fisher Scoring iterations: 5
```

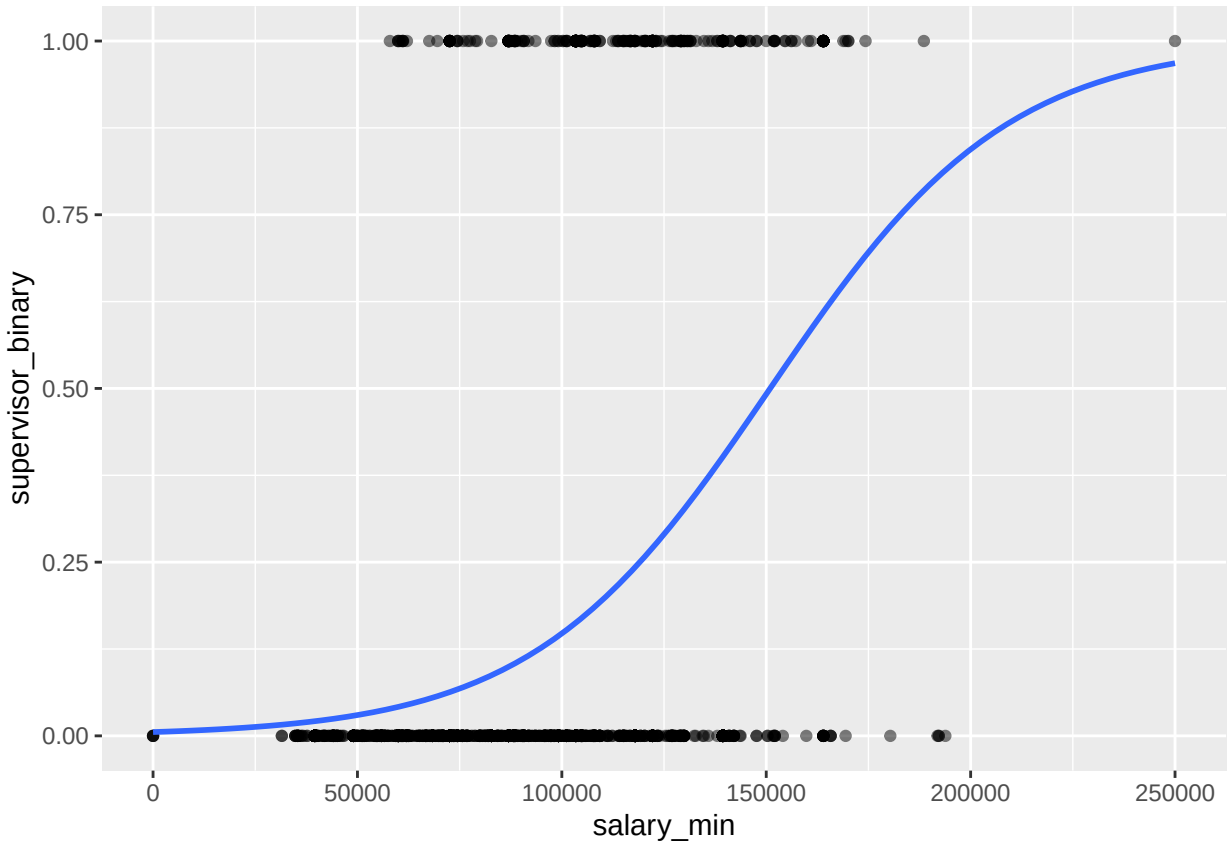
```r
#plot(supervisor_model)
ggplot(supervisor_data, aes(x=salary_max, y=supervisor_binary)) +
        geom_point(alpha=.5) +
        stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(supervisor_data, aes(x=salary_min, y=supervisor_binary)) +
        geom_point(alpha=.5) +
        stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


1.00
0.75
0.50
0.25
0.00
supervisor_binary
0
50000
100000
150000
200000
250000
salary_min