Exploratory Data Analysis

Here, we explore the trends in wages across various features and summarize our findings.

Data Loading

```
list.of.packages <- c("tidyverse", "maps", "ggplot2", "dplyr")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

library('tidyverse')
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(ggplot2)
library(dplyr)
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(viridis) # For color scales in maps
```

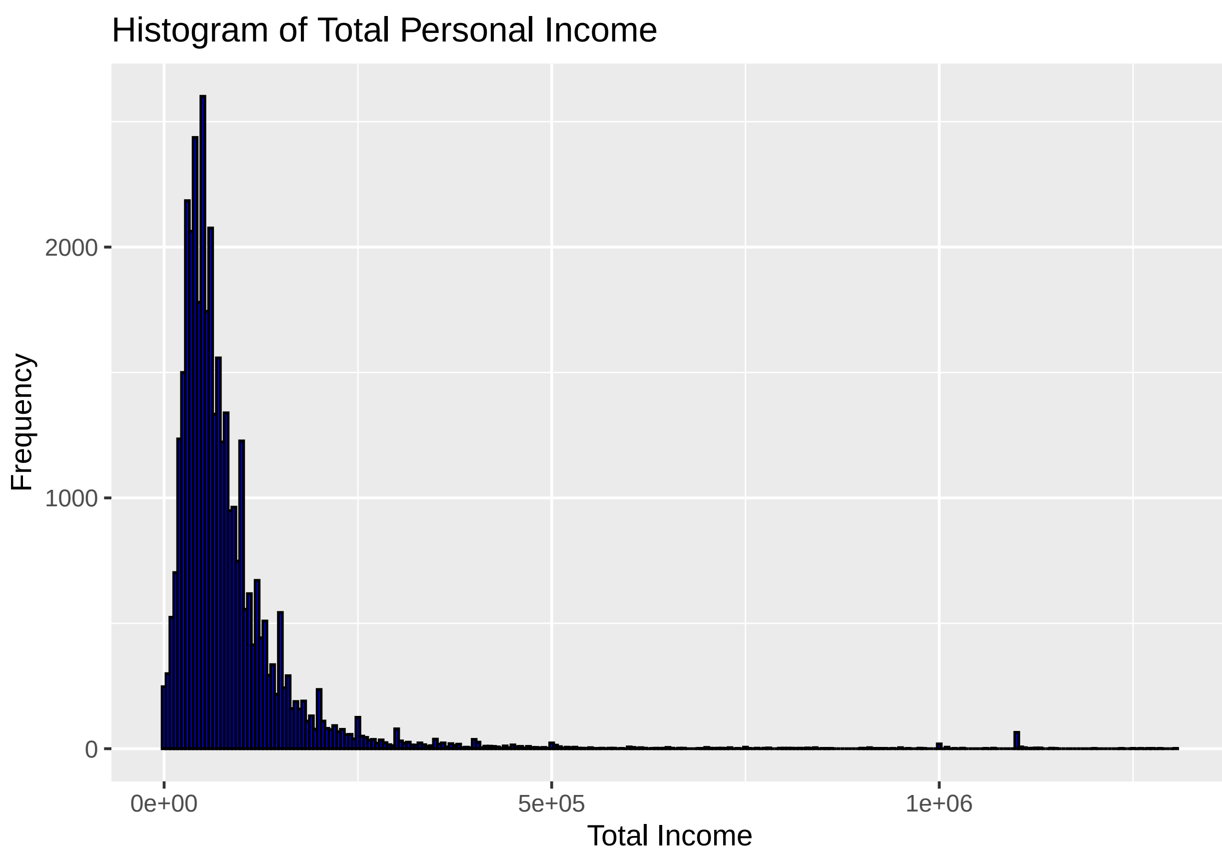
```
## Loading required package: viridisLite
##
## Attaching package: 'viridis'
##
## The following object is masked from 'package:maps':
##
##     unemp
```

```r
set.seed(123) # Setting a seed for reproducibility

# Loading the processed data
data <- read.csv("eda_data_clean.csv")
```

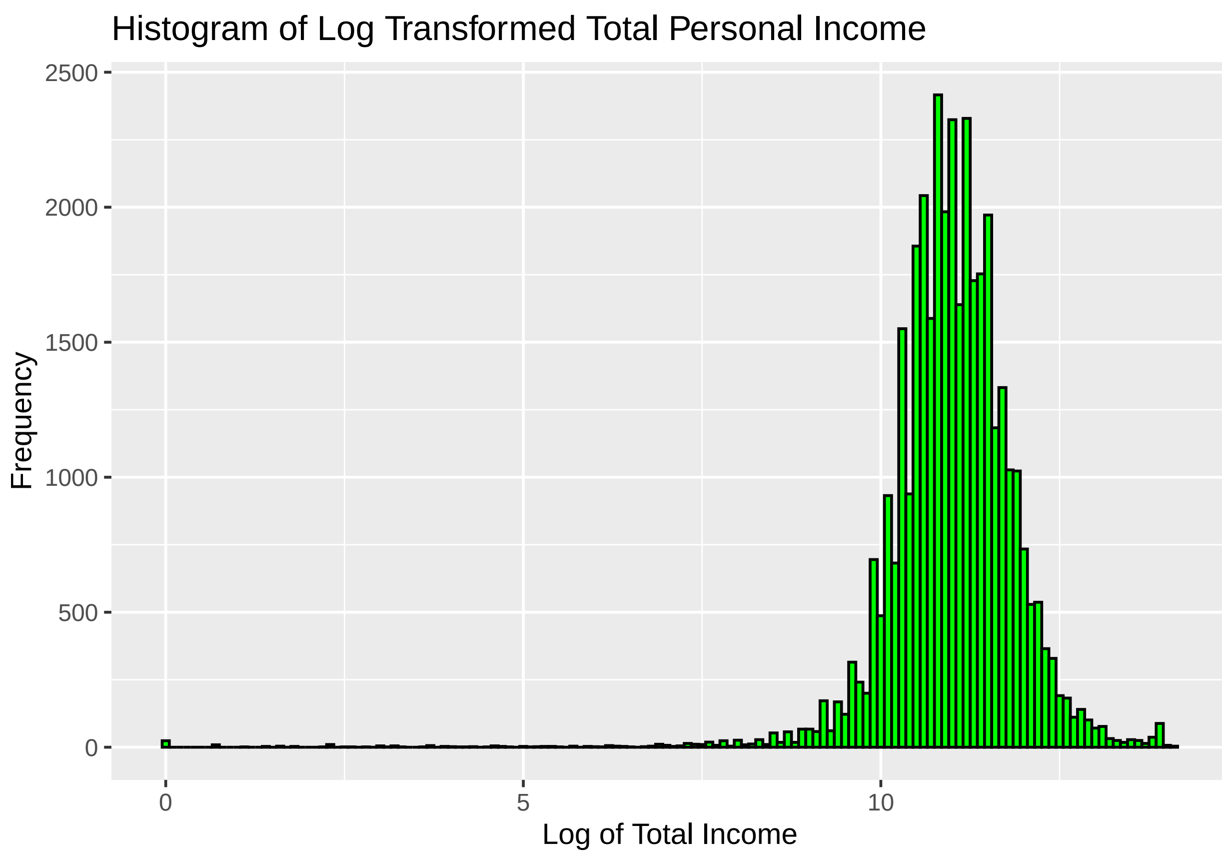
Target Variable Histograms

Here, we look at histograms for INCTOT and Log(INCTOT)

```r
# Histogram for INCTOT
ggplot(data, aes(x = INCTOT)) +
geom_histogram(binwidth = 5000, fill = "blue", color = "black") +
labs(title = "Histogram of Total Personal Income", x = "Total Income", y = "Frequency")
```



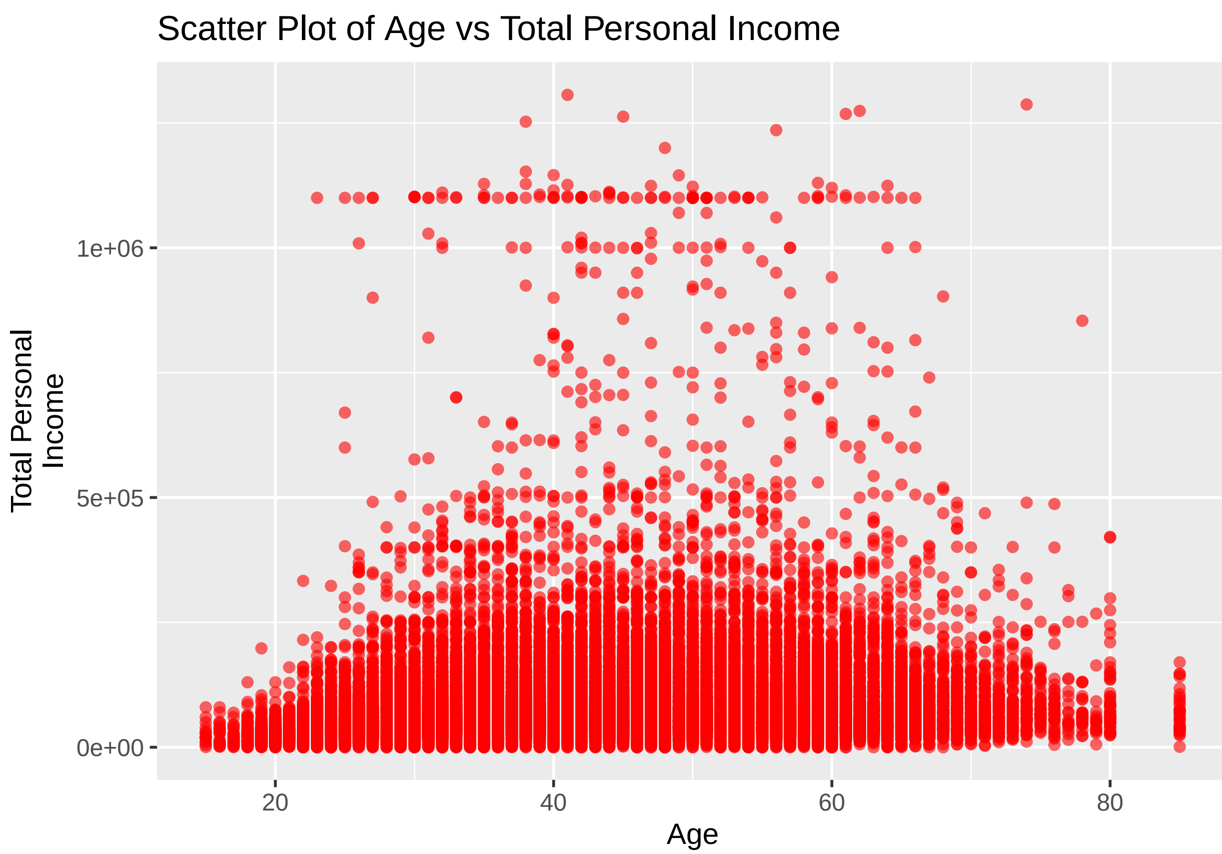

```r
# Histogram for Log(INCTOT)
ggplot(data, aes(x = log_INCTOT)) + # Assuming the log column is named as Log.INCTOT
geom_histogram(binwidth = 0.1, fill = "green", color = "black") +
labs(title = "Histogram of Log Transformed Total Personal Income", x = "Log of Total Income", y
= "Frequency")
```



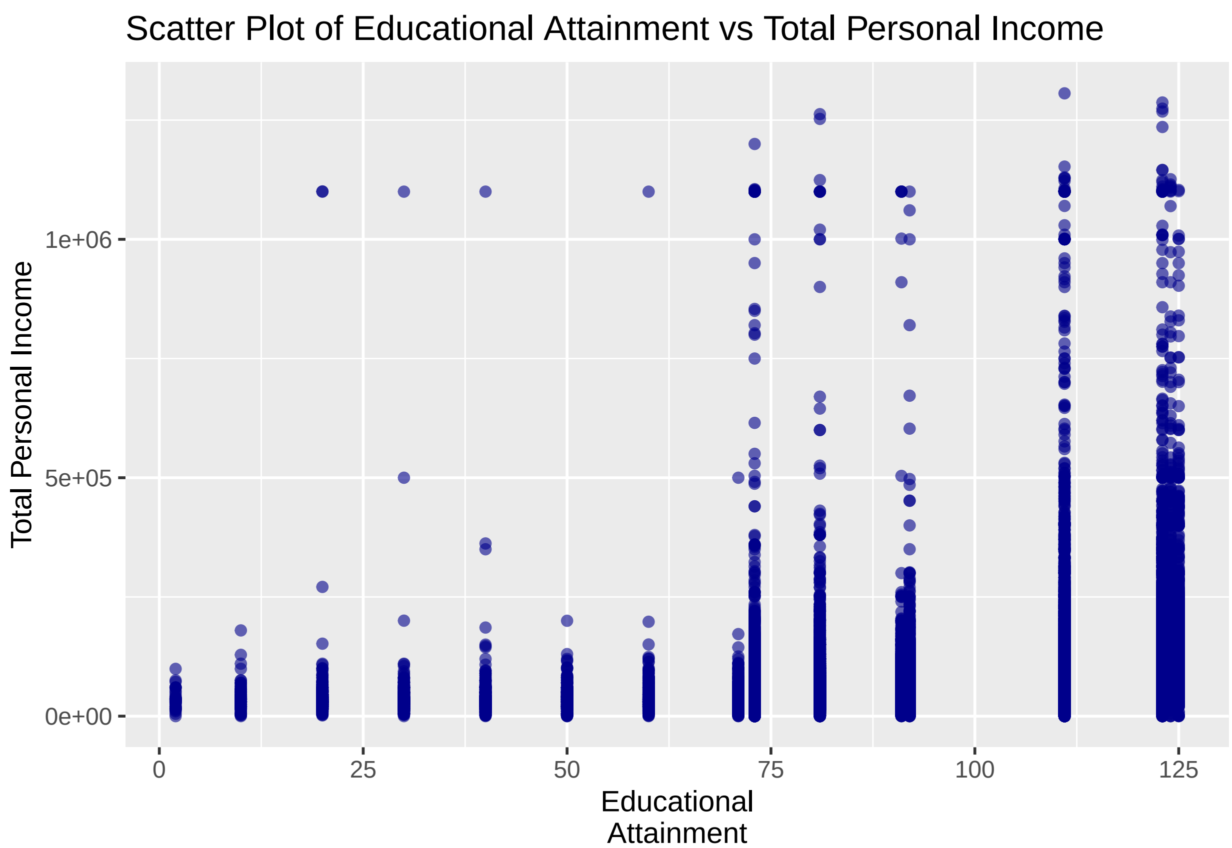
We observe that by log transformation, we are able to shift the concentration from the left side of the original plot to a distribution which is more central and suitable for our analysis.

Plotting AGE and Education against Total Income

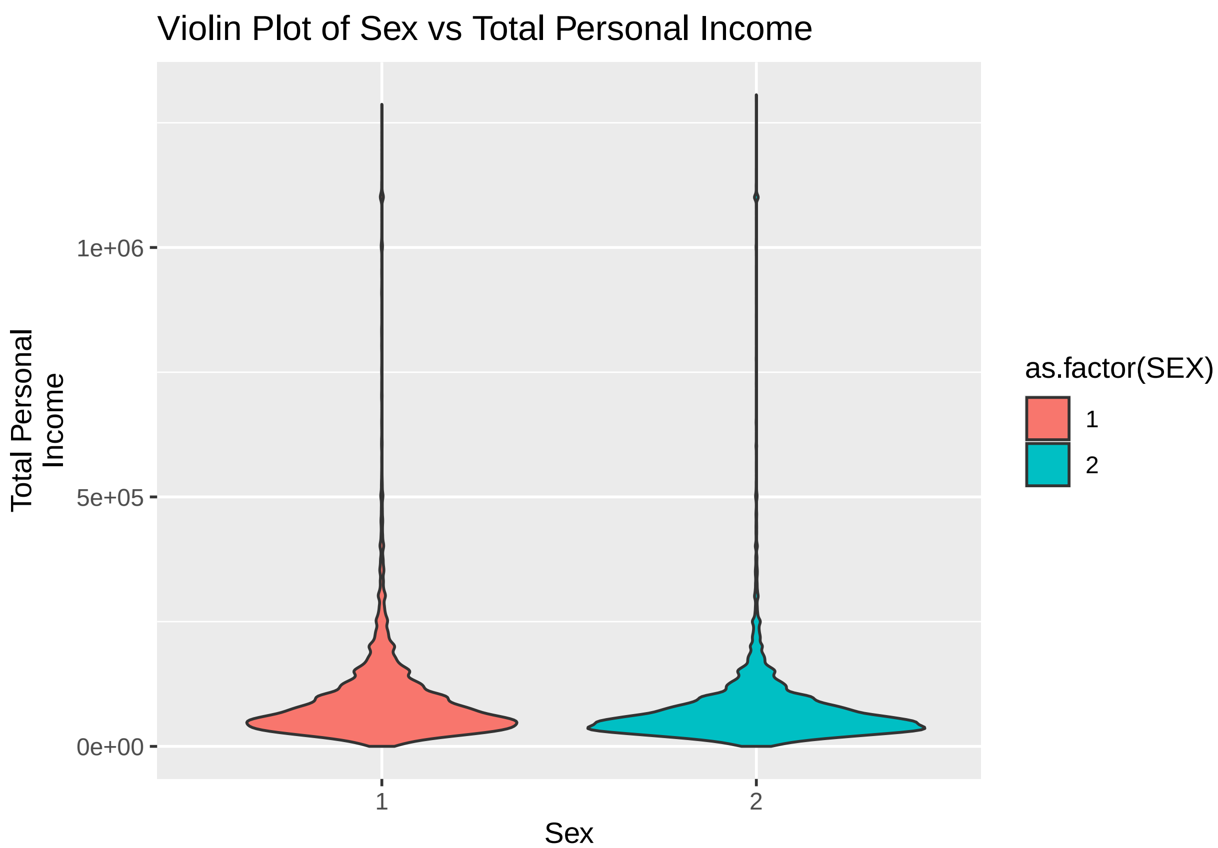
```
# Scatter Plot for Age vs INCTOT
ggplot(data, aes(x = AGE, y = INCTOT)) +
geom_point(alpha = 0.6, color = "red") +
labs(title = "Scatter Plot of Age vs Total Personal Income", x = "Age", y = "Total Personal
Income")
```




```r
# Scatter Plot for EDUC vs INCTOT
ggplot(data, aes(x = EDUC, y = INCTOT)) +
geom_point(alpha = 0.6, color = "darkblue") +
labs(title = "Scatter Plot of Educational Attainment vs Total Personal Income", x = "Educational
Attainment", y = "Total Personal Income")
```




```
# Violin Plot for SEX vs INCTOT
ggplot(data, aes(x = as.factor(SEX), y = INCTOT, fill = as.factor(SEX))) +
geom_violin() +
labs(title = "Violin Plot of Sex vs Total Personal Income", x = "Sex", y = "Total Personal
Income")
```



We are able to observe trends in Age and Education, where in we see a rise in income as Age increases and then after a certain point (retirment), the income goes down. In case of Education, we see as the degree attained is better, the income is also generally better.

There are no stark differences for the gender plot. However, we can see that Males(1) have a slightly higher concentration on higher personal income as compared to Female(2).

State-wise Trends

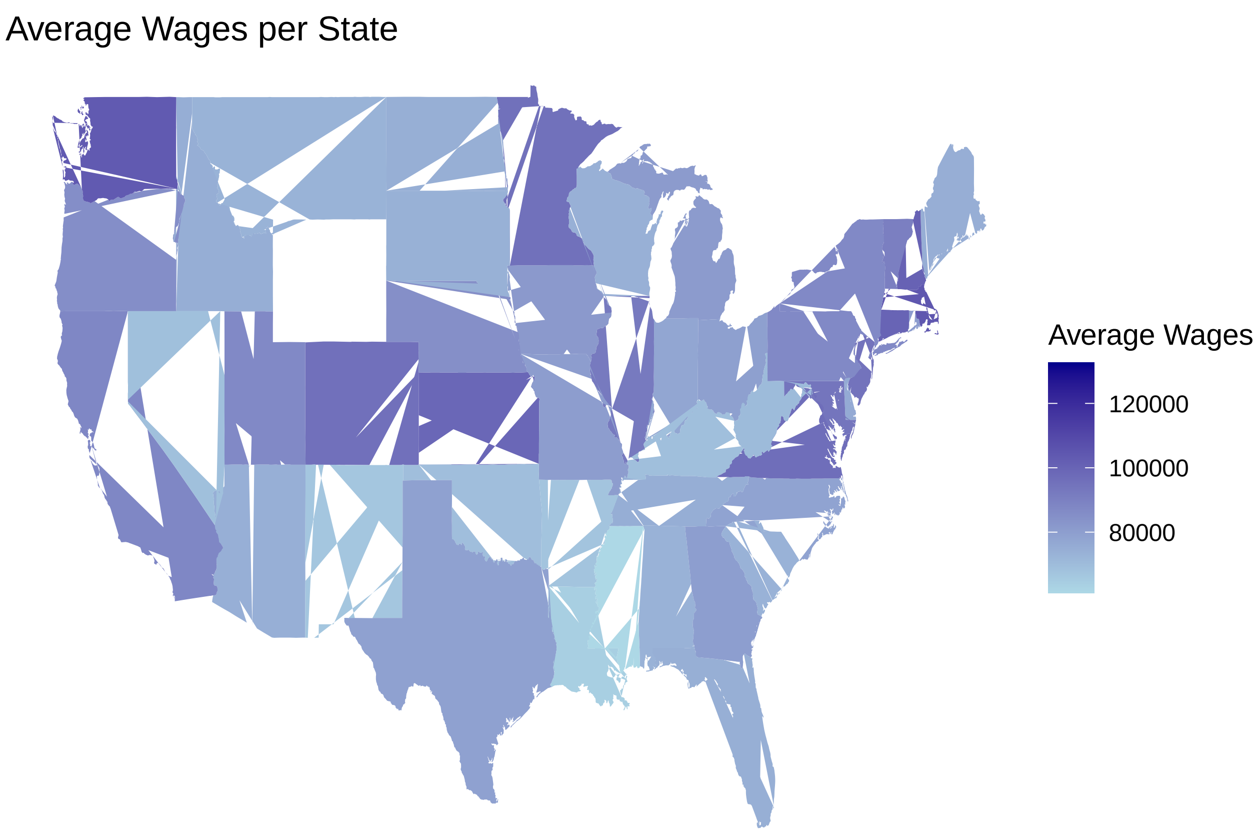
We plot the average wages across various states on the US map.

```
us_map <- map_data("state")

# Aggregate data to get average INCTOT by STATEFIP
state_income <- data %>%
group_by(STATEFIP, STATE) %>%
summarise(Avg_INCTOT = mean(INCTOT, na.rm = TRUE))
```

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

```
state_income$STATE <- toupper(state_income$STATE)
us_map$region <- toupper(us_map$region)
```

```r
map_data_merged <- merge(us_map, state_income, by.x = "region", by.y = "STATE", all.x = TRUE)
map_data_merged <- map_data_merged %>% select(-subregion)
map_data_merged_clean <- na.omit(map_data_merged)


# Plotting the map
ggplot(data = map_data_merged_clean) +
  geom_polygon(aes(x = long, y = lat, group = group, fill = Avg_INCTOT)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "Average Wages") +
  labs(title = "Average Wages per State") +
  theme_void()
```



We observe the typical states where we would see large incomes, on eastern coast, New York and on the Western Coast, Seattle and California

Top Metropolitans, Occupations, and Industry

Here, we aggregate the data to get the top 10 areas, occupations and indsutry based on average wages.

```r
# Top 10 Metropolitan Areas by Average Salary
data %>%
group_by(MET) %>%
summarise(Avg_Salary = mean(INCTOT, na.rm = TRUE)) %>%
```

```r
arrange(desc(Avg_Salary)) %>%
head(10)
```

```
## # A tibble: 10 x 2
##    MET                                           Avg_Salary
##    <chr>                                              <dbl>
##  1 Boulder, CO                                      146960.
##  2 San Jose-Sunnyvale-Santa Clara, CA               139548.
##  3 San Francisco-Oakland-Fremont, CA                128529.
##  4 Santa Cruz-Watsonville, CA                       127287.
##  5 Bridgeport-Stamford-Norwalk, CT                  126619.
##  6 South Bend-Mishawaka, IN-MI                      125892
##  7 Washington-Arlington-Alexandria, DC-VA-MD-WV    121899.
##  8 Medford, OR                                      115758.
##  9 Sherman-Dennison, TX                             115681.
## 10 Durham-Chapel Hill, NC                           115178.
```

```r
data %>%
group_by(OCC_VAL) %>%
summarise(Avg_Salary = mean(INCTOT, na.rm = TRUE)) %>%
arrange(desc(Avg_Salary)) %>%
head(10)
```

```
## # A tibble: 10 x 2
##    OCC_VAL                                      Avg_Salary
##    <chr>                                             <dbl>
##  1 Fire inspectors                                 361253
##  2 Podiatrists                                     352205
##  3 Surgeons                                        344075.
##  4 Prepress technicians and workers                268707.
##  5 Other physicians                                255147.
##  6 Chief executives                                225412.
##  7 Lawyers                                         220460.
##  8 Dentists                                        194531.
##  9 Broadcast announcers and radio disc jockeys     194410.
## 10 Petroleum engineers                             191392.
```

```r
data %>%
group_by(OCC_CATEG) %>%
summarise(Avg_Salary = mean(INCTOT, na.rm = TRUE)) %>%
arrange(desc(Avg_Salary)) %>%
head(10)
```

```
## # A tibble: 10 x 2
##    OCC_CATEG                                                Avg_Salary
##    <chr>                                                         <dbl>
##  1 Legal occupations                                           171316.
##  2 Computer and mathematical science occupations               127758.
##  3 Management occupations                                      124286.
##  4 Architecture and engineering occupations                    122267.
##  5 Healthcare practitioner and technical occupations           109344.
##  6 Business and financial operations occupations               104068.
```

```
##  7 Life, physical, and social science occupations                99804.
##  8 Arts, design, entertainment, sports, and media occupations     90943.
##  9 Protective service occupations                                83385.
## 10 Sales and related occupations                                 83073.
```

```r
data %>%
group_by(IND_VAL) %>%
summarise(Avg_Salary = mean(INCTOT, na.rm = TRUE)) %>%
arrange(desc(Avg_Salary)) %>%
head(10)
```

```
## # A tibble: 10 x 2
##    IND_VAL                                                        Avg_Salary
##    <chr>                                                               <dbl>
##  1 Internet publishing and broadcasting and web search portals       219765.
##  2 Oil and gas extraction                                            203224.
##  3 Wholesale electronics markets, agents and brokers                 179979.
##  4 Securities, commodities, funds, trusts, and other financial inves~ 167255.
##  5 Legal services                                                    157467.
##  6 Software publishers                                               151709.
##  7 Cutlery and hand tool manufacturing                               150291.
##  8 Computer systems design and related services                      144991.
##  9 Aerospace product and parts manufacturing                         138477.
## 10 Textile and fabric finishing and coating mills                    135777
```

```r
data %>%
group_by(IND_CATEG) %>%
summarise(Avg_Salary = mean(INCTOT, na.rm = TRUE)) %>%
arrange(desc(Avg_Salary)) %>%
head(10)
```

```
## # A tibble: 10 x 2
##    IND_CATEG                                              Avg_Salary
##    <chr>                                                       <dbl>
##  1 Broadcasting (except internet)                            157811.
##  2 Publishing industries (except internet)                   131513.
##  3 Management of companies and enterprises                   130441.
##  4 Professional and technical services                       129379.
##  5 Finance                                                   127787.
##  6 Internet service providers and data processing services   125266.
##  7 Computer and electronic products                          110703.
##  8 Chemical manufacturing                                    110112.
##  9 Mining                                                    109368.
## 10 Telecommunications                                        102464
```

Saving Data for modelling

Finally, once we have utilized the required columns for EDA, we save a data with only the required features for modelling

```
data1 <- data %>% select(-METFIPS, -MET, -STATE,
                         -OCC, -OCC_VAL, -OCC_CATEG,
                         -IND, -IND_VAL, -IND_CATEG)

write.csv(data1, "model_data_clean.csv", row.names = FALSE)
```