College Data

Introduction —-

This data contain earnings and information of college graduate based on their fields of study, major. In this project, I will try to do some exploratory analysis.I will try do a prediction on the salary if the time allow me

let's start by loading this below packages and our data —-

```
library(tidyverse)
library(ggplot2)

college_grad <- read.csv("C:/Users/Amara Diallo/Desktop/college_data.txt")
```

Let's convert our column in capital letter —

```
college_grad <- college_grad %>%
  set_names(names(.) %>%
              str_to_title())


glimpse(college_grad)
```

```
## Rows: 173
## Columns: 21
## $ Rank                 <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,...
## $ Major_code           <int> 2419, 2416, 2415, 2417, 2405, 2418, 6202, 5001...
## $ Major                <chr> "PETROLEUM ENGINEERING", "MINING AND MINERAL E...
## $ Total                <int> 2339, 756, 856, 1258, 32260, 2573, 3777, 1792,...
## $ Men                  <int> 2057, 679, 725, 1123, 21239, 2200, 2110, 832, ...
## $ Women                <int> 282, 77, 131, 135, 11021, 373, 1667, 960, 1090...
## $ Major_category       <chr> "Engineering", "Engineering", "Engineering", "...
## $ Sharewomen           <dbl> 0.1205643, 0.1018519, 0.1530374, 0.1073132, 0....
## $ Sample_size          <int> 36, 7, 3, 16, 289, 17, 51, 10, 1029, 631, 399,...
## $ Employed             <int> 1976, 640, 648, 758, 25694, 1857, 2912, 1526, ...
## $ Full_time            <int> 1849, 556, 558, 1069, 23170, 2038, 2924, 1085,...
## $ Part_time            <int> 270, 170, 133, 150, 5180, 264, 296, 553, 13101...
## $ Full_time_year_round <int> 1207, 388, 340, 692, 16697, 1449, 2482, 827, 5...
## $ Unemployed           <int> 37, 85, 16, 40, 1672, 400, 308, 33, 4650, 3895...
## $ Unemployment_rate    <dbl> 0.018380527, 0.117241379, 0.024096386, 0.05012...
## $ Median               <int> 110000, 75000, 73000, 70000, 65000, 65000, 620...
## $ P25th                <int> 95000, 55000, 50000, 43000, 50000, 50000, 5300...
```

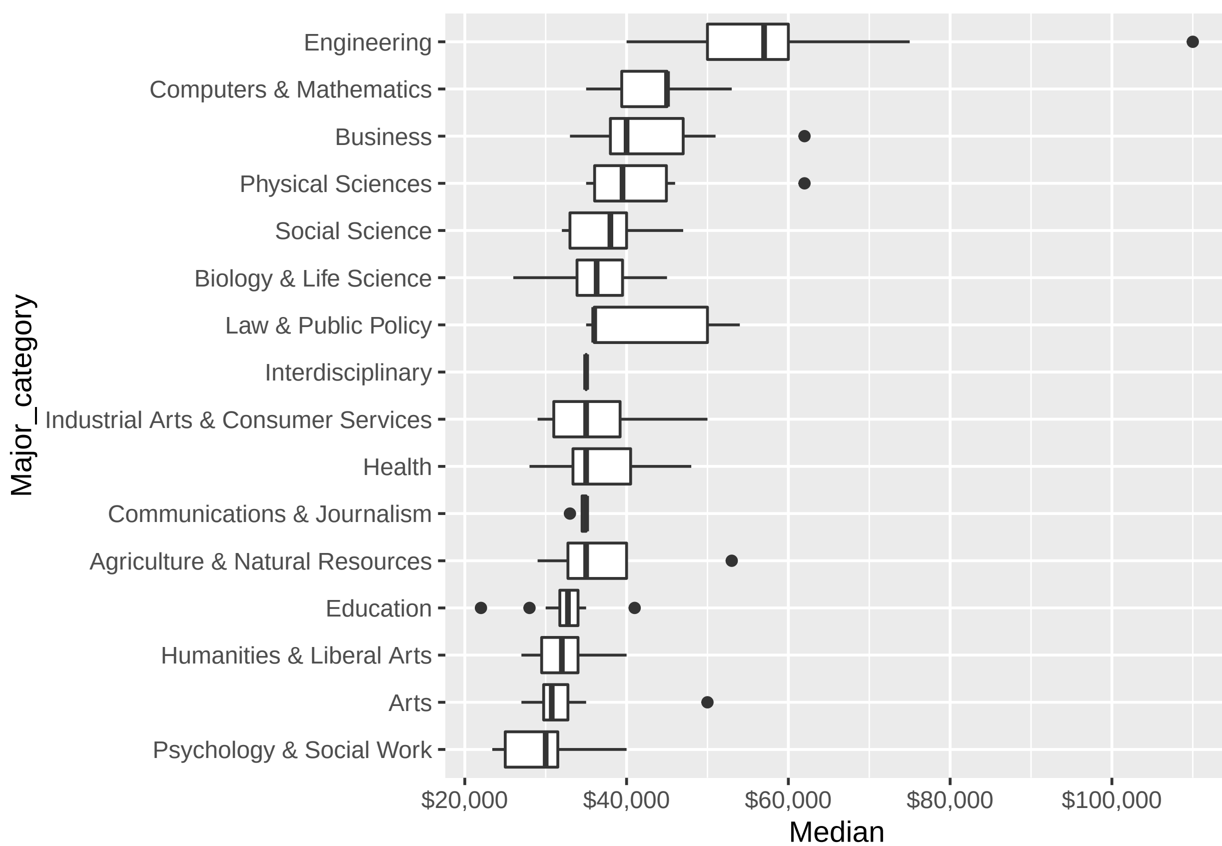
```
## $ P75th             <int> 125000, 90000, 105000, 80000, 75000, 102000, 7...
## $ College_jobs      <int> 1534, 350, 456, 529, 18314, 1142, 1768, 972, 5...
## $ Non_college_jobs  <int> 364, 257, 176, 102, 4440, 657, 314, 500, 16384...
## $ Low_wage_jobs     <int> 193, 50, 0, 0, 972, 244, 259, 220, 3253, 3170,...
```

```
#view(college_grad)
```

Here we will look for the Major categories that make more money upon graduation —-

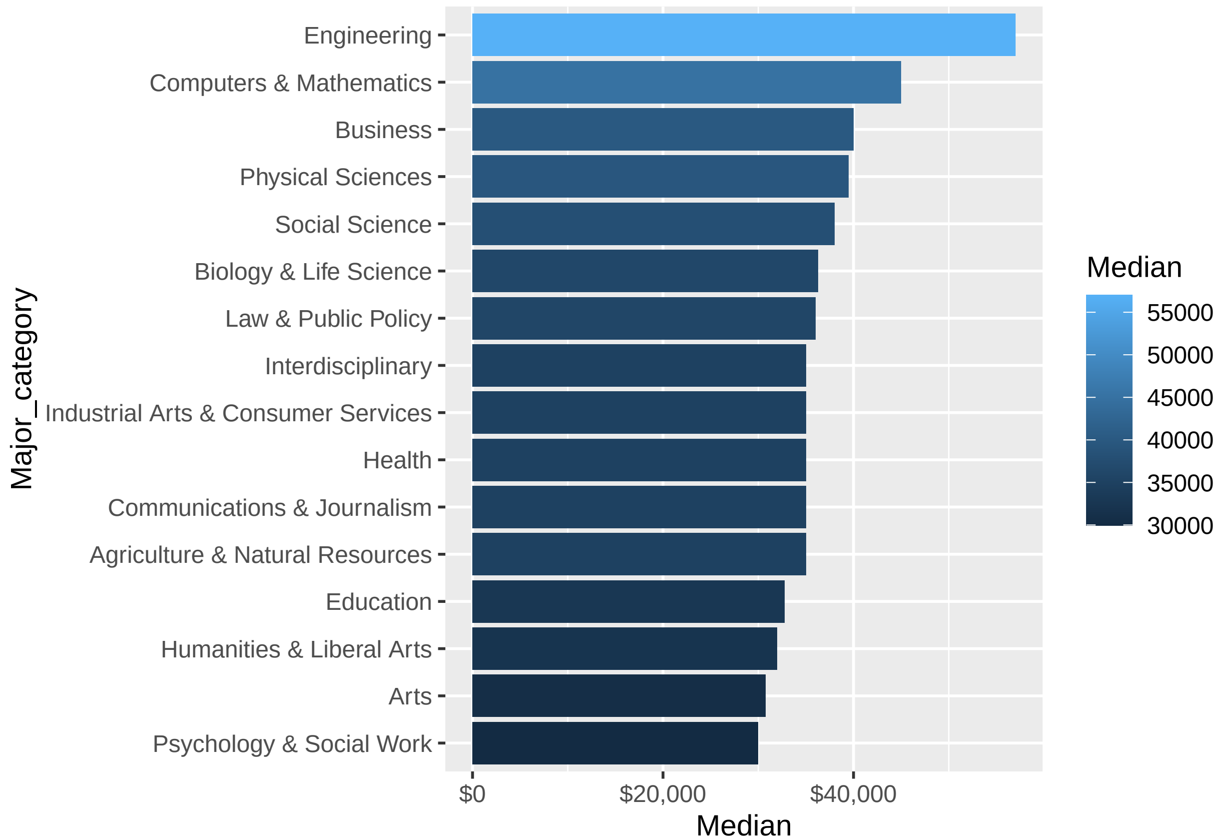
In this section I will do a visualizations of the data in order to find out what major category is leading in term of salary in the job market.

```
college_grad %>%
    mutate(Major_category = fct_reorder(Major_category, Median)) %>%
    ggplot(aes(Major_category, Median)) +
    geom_boxplot() +
    scale_y_continuous(labels = scales::dollar_format()) +
    coord_flip()
```




```
college_grad %>%
    group_by(Major_category) %>%
    summarize(Median = median(Median)) %>%
    mutate(Major_category = fct_reorder(Major_category, Median)) %>%
    ggplot(aes(Major_category, Median, fill =Median )) +
```

```
    geom_bar(stat="identity") +
    scale_y_continuous(labels = scales::dollar_format()) +
    coord_flip()
```



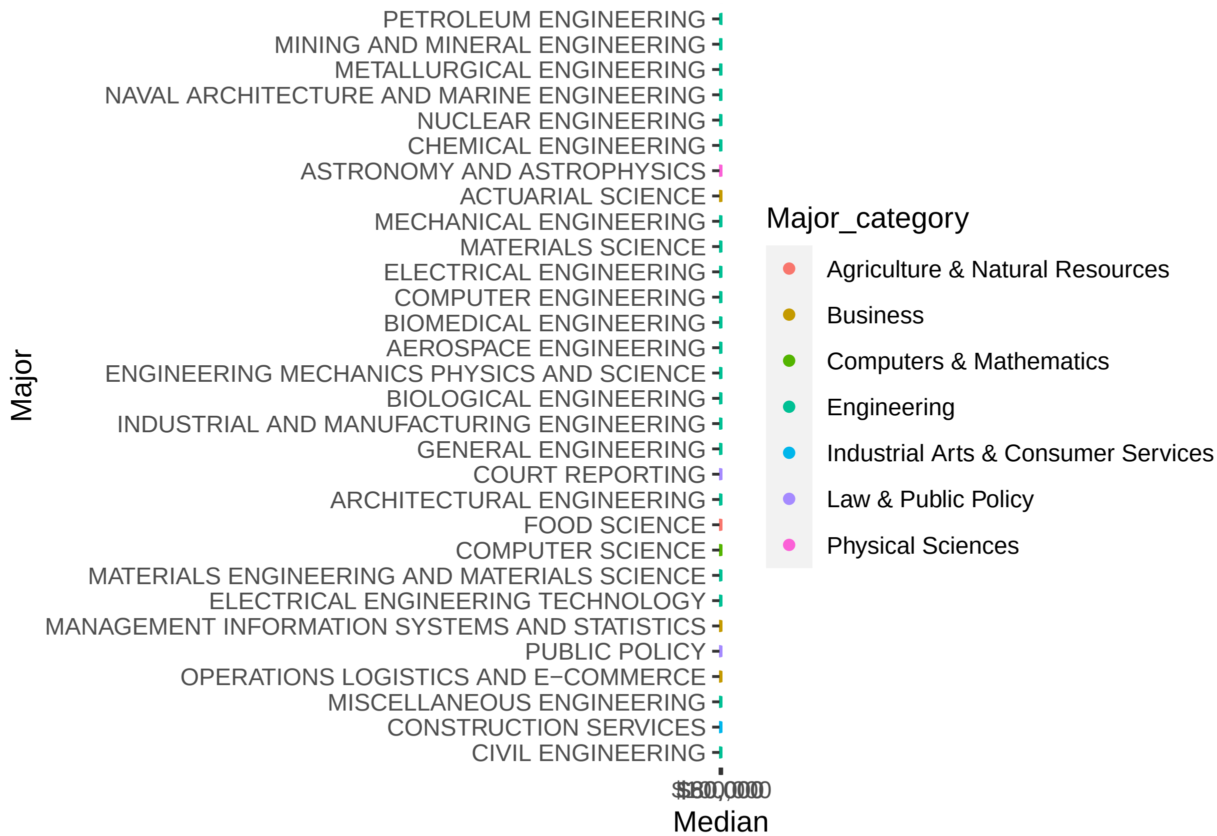
As we can see above, Engineering student are the one who get more money after graduation and follow by computer&Math students. With Arts & Journalism as the lowest paying job. This finding is based on the current data we have, I am also assuming that this is probably for junior position. But, on the first graph we could point out an outlier, which mean that there is a field in the **Engineering Major** that makes a lot of money than the other field in the ***Engineering***

In this section, we will find the highest top earning majors in all major category.—-

This will help us extrapolate what is the highest major amont all major category, but it will also allow us to find out what the *OUTLIER IN ENGINEERING* we found in our early graph: The field that makes more money than all other **ENGINEERING** fields.

```
Majors <- college_grad %>%
    arrange(desc(Median)) %>%
    select(Major, Major_category, Median, P25th, P75th, Sample_size)  %>%
    mutate(Major = fct_reorder(Major, Median))

Majors %>% head(30) %>%
    ggplot(aes(Major, Median, color = Major_category)) +
    geom_point() +
```

```
    scale_y_continuous(labels = scales::dollar_format()) +
    coord_flip()
```

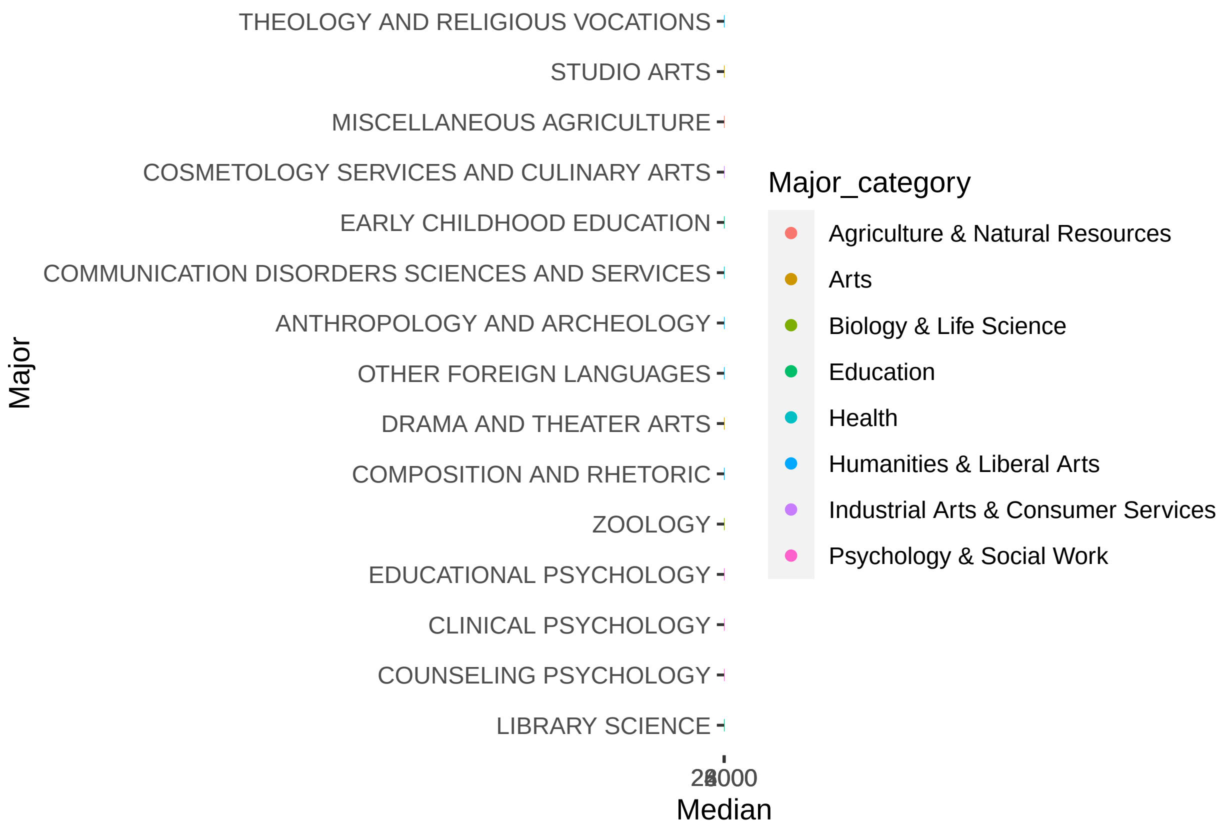


Most the highest earning majors are from **ENGINEERING** field. We also realized that *PETROLEUM ENGINEERING* is not only the highest paying position in **ENGINEERING** field, but it is also the highest paying job in all major category…according to this dataset.

ASTRONOMY & ASTROPHYSICS is the second highest paying job, coming from the *COMPUTER & MATH* Major category; **ACTUARIAL SCIENCE** is the third and the first in the Business department;

The lowest earning Majors —-

```
college_grad %>%
    select(Major, Major_category, Median, P25th, P75th)  %>%
    tail(15) %>%

    mutate(Major = fct_reorder(Major, Median)) %>%
    ggplot(aes(Major, Median,color = Major_category)) +
    geom_point() +
    coord_flip()
```



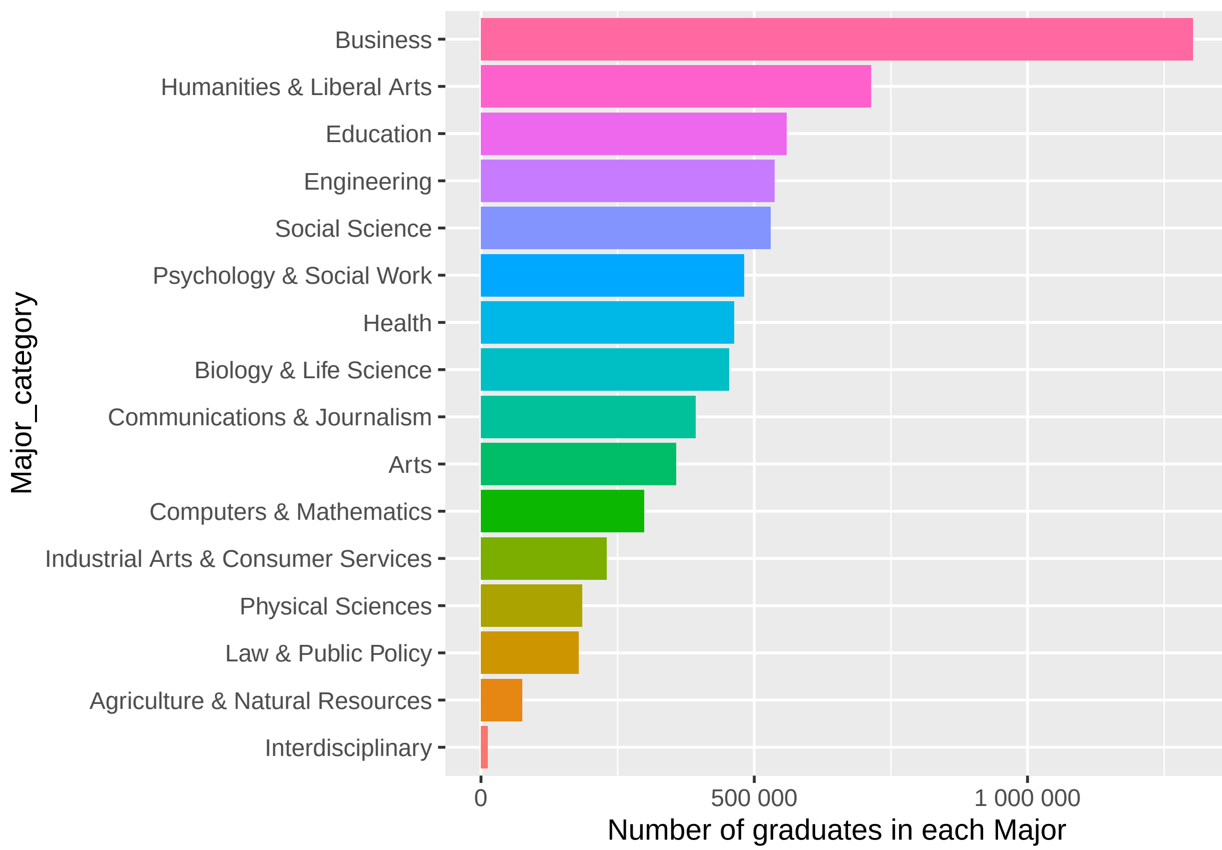
The above graph shows the top 15 lowest paying job ** according to this dataset**. *Library Science* is the lowest paying job. As we notice, no field in the** ENGINEERING** is present in this list.

```
# Majors  %>%
#     ggplot(aes(Sample_size, Median)) +
#     geom_point() +
#     scale_x_log10()
# install.packages("tm")           # for text mining
# install.packages("SnowballC")    # for text stemming
# install.packages("wordcloud")    # word-cloud generator
# install.packages("RColorBrewer") # color palettes

# Load the packages
# library("tm")
# library("SnowballC")
# library("wordcloud")
# library("RColorBrewer")
# wordcloud(words = Majors$Major_category,
#           freq = Majors$Median,
#           min.freq = 1,
#           max.words =200,
#           random.order = TRUE,
#           rot.per = 0.35,
#           colors = brewer.pal(8, "Dark2"))
```

Most common majors —-

This part will tell us what is the major that attact most of students. We are not surprise to see that **BUSINESS** is by far the common major for college students. It is twice attractive than the rest of the major… Specially Engineering.

```
college_grad %>%
    count(Major_category, wt = Total, sort = TRUE) %>%
    mutate(Major_category = fct_reorder(Major_category,n)) %>%
    ggplot(aes(Major_category, n, fill = Major_category)) +
    theme(legend.position = "none") +
    geom_col() +
    coord_flip() +
    scale_y_continuous(labels = scales::number_format()) +
    labs(y = "Number of graduates in each Major")
```

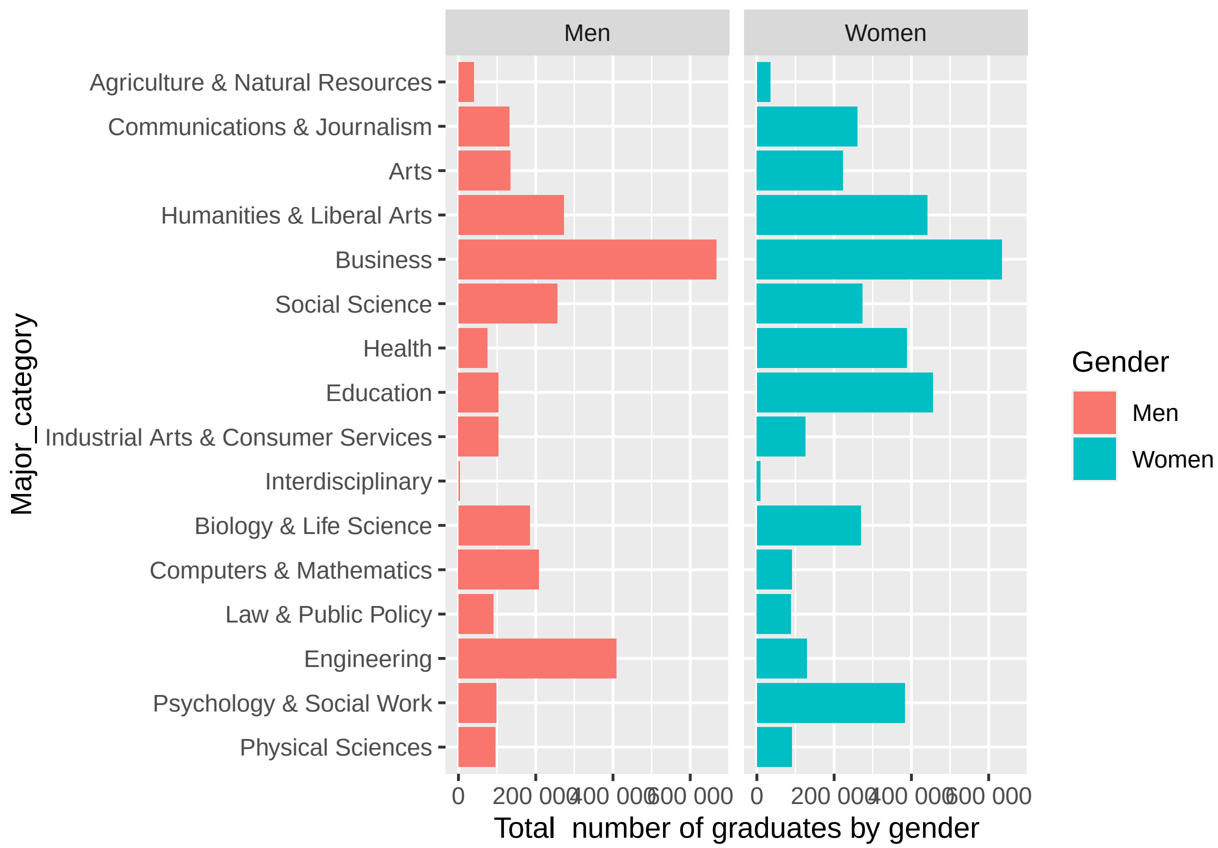


Let's see if we can find the number of graduate in each major category, but based on gender. —-

```
college_grad %>%
    mutate(Major_category = fct_reorder(Major_category, Total)) %>%
    gather(Gender,Total, Men, Women) %>%
    group_by(Major_category, Gender) %>%
    #summarize(Median = median(Median)) %>%
```

```
ggplot(aes(Major_category,Total, fill = Gender)) +
geom_col() +
facet_grid(~Gender) +
coord_flip() +
scale_y_continuous(labels = scales::number_format()) +
labs(y = "Total  number of graduates by gender")
```

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

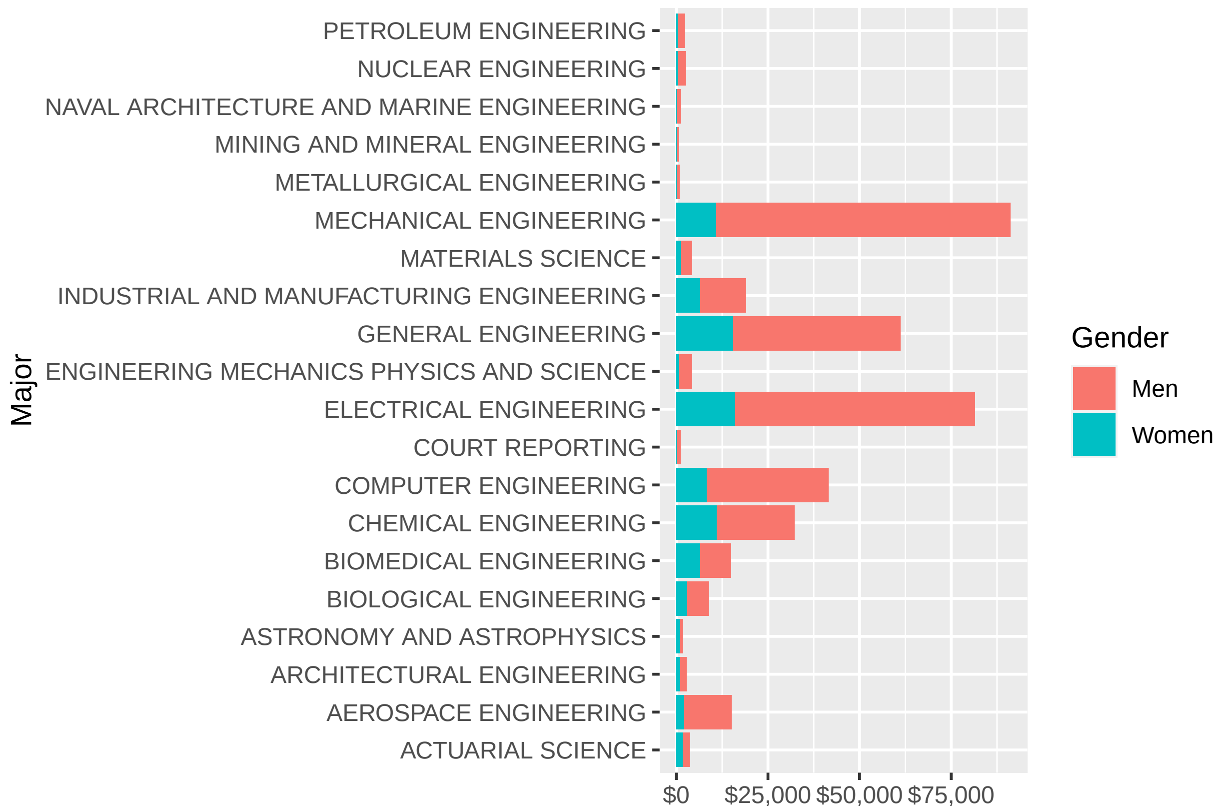


It looks like *MEN* are more STERM oriented than *WOMEN*. As we can, most of the graduate students in Engineering, Computer & MATH are *MEN*. However, it is undeniable that women are ahead in health and social science related majors. Business Major is dominated by both gender.

which gender earn more money based on top 15 Major —-

```
college_grad %>%
mutate(Major_category = fct_reorder(Major_category, Median)) %>%
  top_n(20, Median) %>%

gather(Gender,Median, Men, Women) %>%
group_by(Major_category, Gender) %>%

  ggplot(aes(Major, Median, fill = Gender)) +
```

```
    geom_col() +
    scale_y_continuous(labels = scales::dollar_format()) +
    labs(y = " ") +

    coord_flip()
```

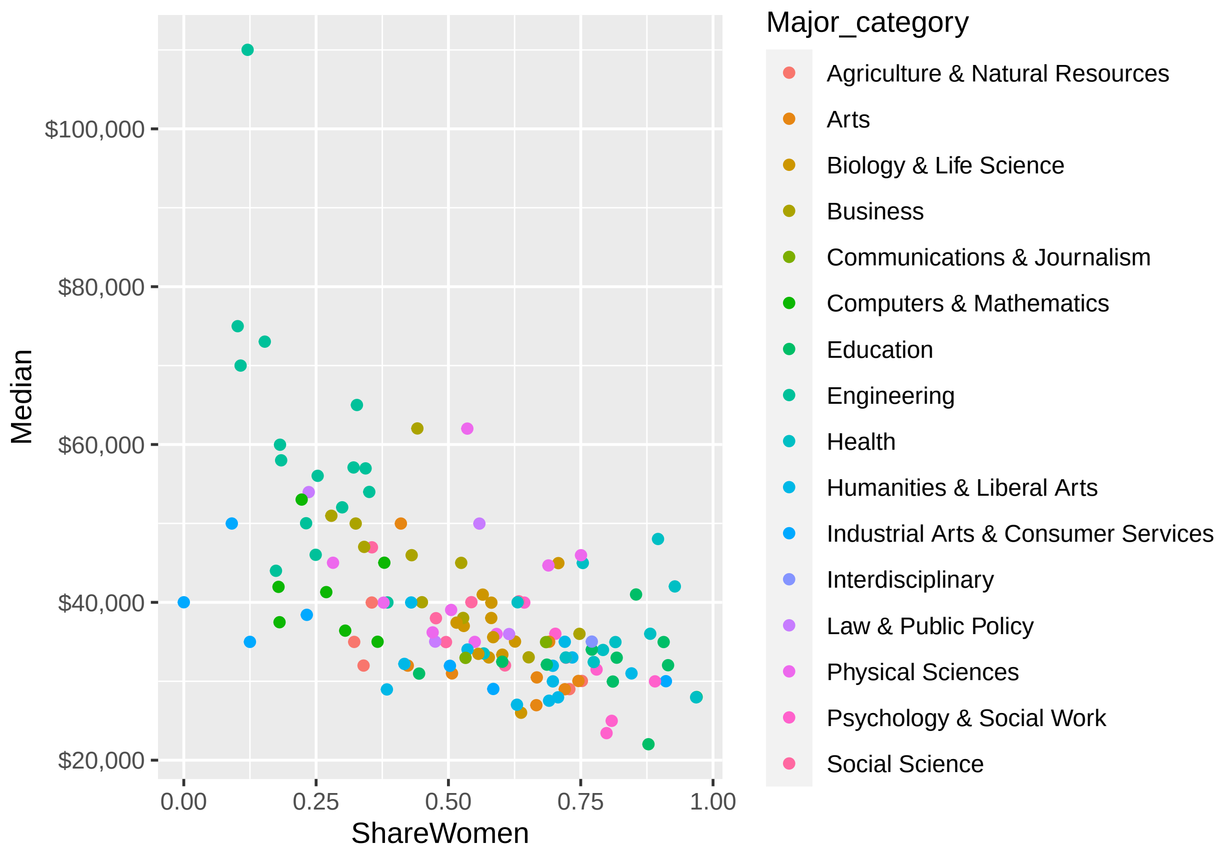


This above graph tells us that Men are pay more than women even though they are in the same major. this is just a dataset about recent graduate, so we can not totally rely on it. In order to really know the ins and out of this above question, we will need more dataset in order to get a better insight.

Share of Women in each Major —-

```
college_grad %>%
    group_by(Major_category, Median) %>%
    summarize_at(vars(Total, Men, Women), sum, na.rm=TRUE) %>%
    mutate(ShareWomen = Women/Total) %>%
    arrange(desc(ShareWomen)) %>%

    ggplot(aes(ShareWomen, Median, color = Major_category)) +
    geom_jitter() +
    scale_y_continuous(labels = scales::dollar_format())
```

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



The previous plot shows that less .25% of women make more than 50k,that is due to the fact that most of those women that are in the .25% are in the high paying major like STEM. More than .75% are below 45K, that can be explain in their major choice… like **SOCIAL SCIENCE, PSYCHOLOGY, EDUCATION, JOURNALISM ect**

```
college_grad %>%
    select(Major, Major_category, Total, Sharewomen, Sample_size, Median) %>%
    add_count(Major_category) %>%
    filter(n >= 10) %>%
    count(Major_category) %>%
    arrange(desc(n))
```

Total Major in each major's category —-

```
##                     Major_category  n
## 1                      Engineering 29
## 2                        Education 16
## 3         Humanities & Liberal Arts 15
## 4            Biology & Life Science 14
## 5                         Business 13
## 6                           Health 12
## 7           Computers & Mathematics 11
## 8  Agriculture & Natural Resources 10
## 9                Physical Sciences 10
```

Let's see how much money Women get in health care profession —-

```
college_grad %>% filter(Major_category == 'Health' & Sharewomen >0.7) %>%
  select(Major, Median)
```

```
##                                        Major Median
## 1                                    NURSING  48000
## 2             MEDICAL TECHNOLOGIES TECHNICIANS  45000
## 3                 MEDICAL ASSISTING SERVICES  42000
## 4       MISCELLANEOUS HEALTH MEDICAL PROFESSIONS  36000
## 5                         NUTRITION SCIENCES  35000
## 6     HEALTH AND MEDICAL ADMINISTRATIVE SERVICES  35000
## 7                 COMMUNITY AND PUBLIC HEALTH  34000
## 8               TREATMENT THERAPY PROFESSIONS  33000
## 9            GENERAL MEDICAL AND HEALTH SERVICES  32400
## 10 COMMUNICATION DISORDERS SCIENCES AND SERVICES  28000
```

```
#View(fresh_grad_health)
```

Majors with the lowest unemployment rate that might be interesting for students —-

```
college_grad %>%
  select(Major, Unemployment_rate) %>%
  filter(Unemployment_rate<0.02)
```

```
##                                      Major Unemployment_rate
## 1                      PETROLEUM ENGINEERING       0.018380527
## 2   ENGINEERING MECHANICS PHYSICS AND SCIENCE       0.006334343
## 3                           COURT REPORTING       0.011689692
## 4           MATHEMATICS AND COMPUTER SCIENCE       0.000000000
## 5                        GENERAL AGRICULTURE       0.019642463
## 6                      MILITARY TECHNOLOGIES       0.000000000
## 7                                    BOTANY       0.000000000
## 8                              SOIL SCIENCE       0.000000000
## 9              MATHEMATICS TEACHER EDUCATION       0.016202835
## 10 EDUCATIONAL ADMINISTRATION AND SUPERVISION       0.000000000
```

```
#View(fresh_grads_science)
```

Recent graduates with median salary > 40,000 USD where Women are represented by More than 50 percent. —-

```
college_grad %>% filter(Median >=40000 & Sharewomen >.5) %>%
  select(Major, Median)
```

```
##                                                       Major Median
## 1                                   ASTRONOMY AND ASTROPHYSICS  62000
```

```
## 2                                                     PUBLIC POLICY  50000
## 3                                                           NURSING  48000
## 4  NUCLEAR, INDUSTRIAL RADIOLOGY, AND BIOLOGICAL TECHNOLOGIES  46000
## 5                                                        ACCOUNTING  45000
## 6                                 MEDICAL TECHNOLOGIES TECHNICIANS  45000
## 7                                  STATISTICS AND DECISION SCIENCE  45000
## 8                                                      PHARMACOLOGY  45000
## 9                                                      OCEANOGRAPHY  44700
## 10                                     MEDICAL ASSISTING SERVICES  42000
## 11                              COGNITIVE SCIENCE AND BIOPSYCHOLOGY  41000
## 12                                       SCHOOL STUDENT COUNSELING  41000
## 13                                         INTERNATIONAL RELATIONS  40100
## 14                                          INTERNATIONAL BUSINESS  40000
## 15        PHARMACY PHARMACEUTICAL SCIENCES AND ADMINISTRATION  40000
## 16                                               MOLECULAR BIOLOGY  40000
## 17                                                        GENETICS  40000
## 18                                   MISCELLANEOUS SOCIAL SCIENCES  40000
## 19                        INDUSTRIAL AND ORGANIZATIONAL PSYCHOLOGY  40000
```

CONCLUSION —-

The above analysis hels us to understand how our major's choice can have an impact on our financial status. this analysis also touch upon the choice of majors based on gender. For instance, we saw that *Men* are more attracted to STEM option, while *WOMEN* dominate health science profession.

Predict the ShareWomen in each major —-

Let's start creating our data partitioning. 80% of our data will be used for the training set

```r
# Preprocessing & Sampling
library(recipes)
```

```
##
## Attaching package: 'recipes'

## The following object is masked from 'package:stringr':
##
##     fixed

## The following object is masked from 'package:stats':
##
##     step
```

```r
library(rsample)

# Standard
library(readxl)
library(tidyverse)
library(tidyquant)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'

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

## Loading required package: PerformanceAnalytics

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

## Loading required package: quantmod

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## == Need to Learn tidyquant? ====================================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimiza
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```
# Modeling
library(parsnip)

# Plotting Decision Trees
library(rpart.plot)
```

```
## Loading required package: rpart
```

```r
college_grad <- na.omit(college_grad)
set.seed(1113) # make the function reproduceable

dat_split <- rsample::initial_split(college_grad, prop = 0.80, strata = NULL)

#dat_split %>% training()
#dat_split%>% testing()

train_data <- training(dat_split)
test_data <- testing(dat_split)
```

LINEAR METHODS —-

LINEAR REGRESSION - NO ENGINEERED FEATURES —-

.1.1 Model —-

```r
?linear_reg
```

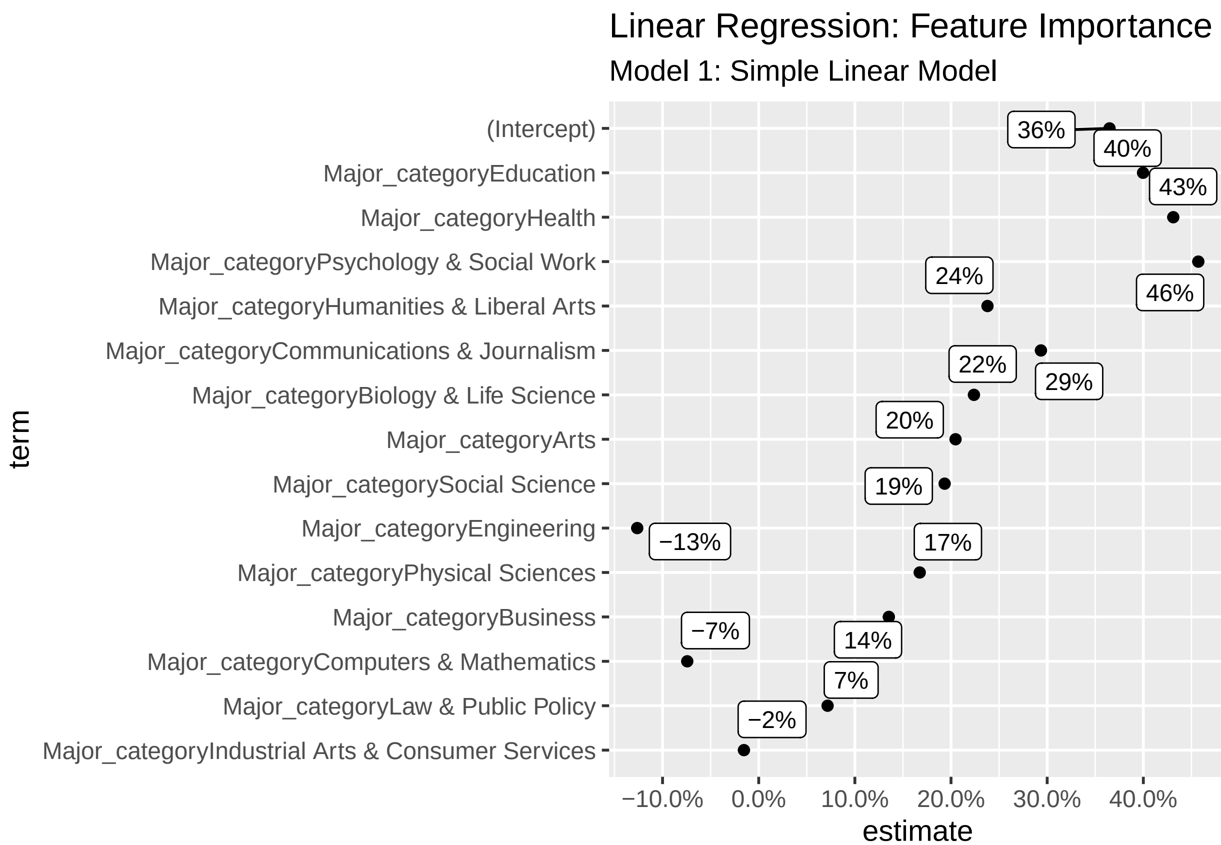
```
## starting httpd help server ... done
```

```r
test_data <- train_data %>%
  bind_rows(train_data %>% filter(Major_category %>% str_detect("Interdisciplinary")))

model_01_lm <- linear_reg(mode = "regression") %>%
  set_engine("lm") %>%
  fit(Sharewomen ~  Major_category, data = train_data)

model_01_lm %>%
  predict(new_data = test_data) %>%

  bind_cols(Sharewomen=test_data$Sharewomen) %>%
  mutate(residuals = Sharewomen - .pred)  %>%
  yardstick::metrics(truth = Sharewomen, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.137
## 2 rsq     standard       0.671
## 3 mae     standard       0.108
```

```r
#model_01_lm$fit

model_01_lm$fit %>%
    broom::tidy() %>%
    arrange(p.value) %>%
    mutate(term = as_factor(term) %>% fct_rev()) %>%

    ggplot(aes(x = estimate, term)) +
    geom_point() +
```

```
    ggrepel::geom_label_repel(aes(label = scales::percent(estimate, accuracy = 1)),
                              size = 3) +
    scale_x_continuous(labels = scales::percent_format()) +
    labs(
        title = "Linear Regression: Feature Importance",
        subtitle = "Model 1: Simple Linear Model"
    )
```



The intercept shows that without any features added, WomenShare is 36% in all major-category. When we start adding "Education & Health" our model changes from 36% to predicting up to 43%. However, when we added "Engineering & Math", the model abstracted WomenShare Percentages

The plot also shows that, each predictor has a coefficient that is in terms of the final output. Major_categoryArt, Major_categoryHealth, Major_categoryEducation, ect, the linear equation becomes:

y_pred = Intercept + c1 x Major_categoryArt + c2 x Major_categoryHealth + c3 x Major_categoryEducation + c4 x etc

Everything else in the model that do not have coeficent is zero because the features are not present.

```
calc_metrics <- function(model, new_data = test_data){

    model %>%
        predict(new_data = new_data) %>%

        bind_cols(new_data %>% select(Sharewomen)) %>%
        mutate(residuals = Sharewomen - .pred) %>%
```

```
    yardstick::metrics(truth = Sharewomen, estimate = .pred)

}

model_01_lm %>%
    calc_metrics(test_data)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.137
## 2 rsq     standard       0.671
## 3 mae     standard       0.108
```

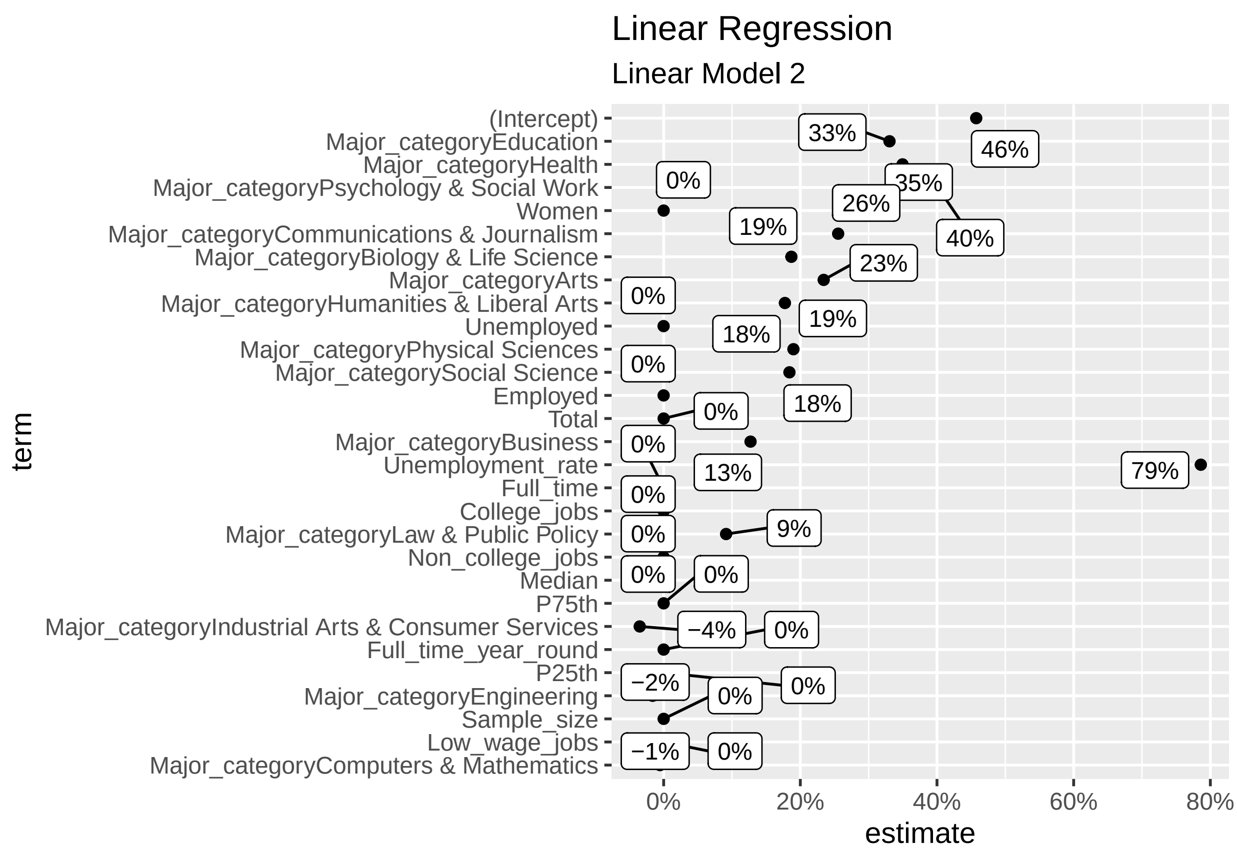
LINEAR REGRESSION - WITH ENGINEERED FEATURES —-

```
Model_2_lm <- linear_reg("regression") %>%
  set_engine("lm") %>%
  fit(Sharewomen~., data = train_data %>% select(-Rank, -Major,-Major_code, -Men, -Part_time ))

Model_2_lm %>%
  calc_metrics(new_data = test_data)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      0.111
## 2 rsq     standard      0.784
## 3 mae     standard      0.0869
```

This second Linear Model has a Lower RMSE & Lower MAE (Mean Absolute Error), which indicates better fit. RSQ (R square): Our model explain 78% of variation within our data- The larger the R2, the better the regression model fits your observations.

```
Model_2_lm$fit %>%
  broom::tidy() %>%
  arrange(p.value) %>%
  mutate(term = as_factor(term) %>% fct_rev()) %>%

  ggplot(aes(x = estimate, term)) +
  geom_point() +
  ggrepel::geom_label_repel(aes(label = scales::percent(estimate, accuracy = 1)),
                            size = 3) +
  scale_x_continuous(labels = scales::percent_format())+
  labs(
    title = "Linear Regression",
    subtitle = "Linear Model 2"
  )
```



TREE-BASED METHODS —-

DECISION TREES —-

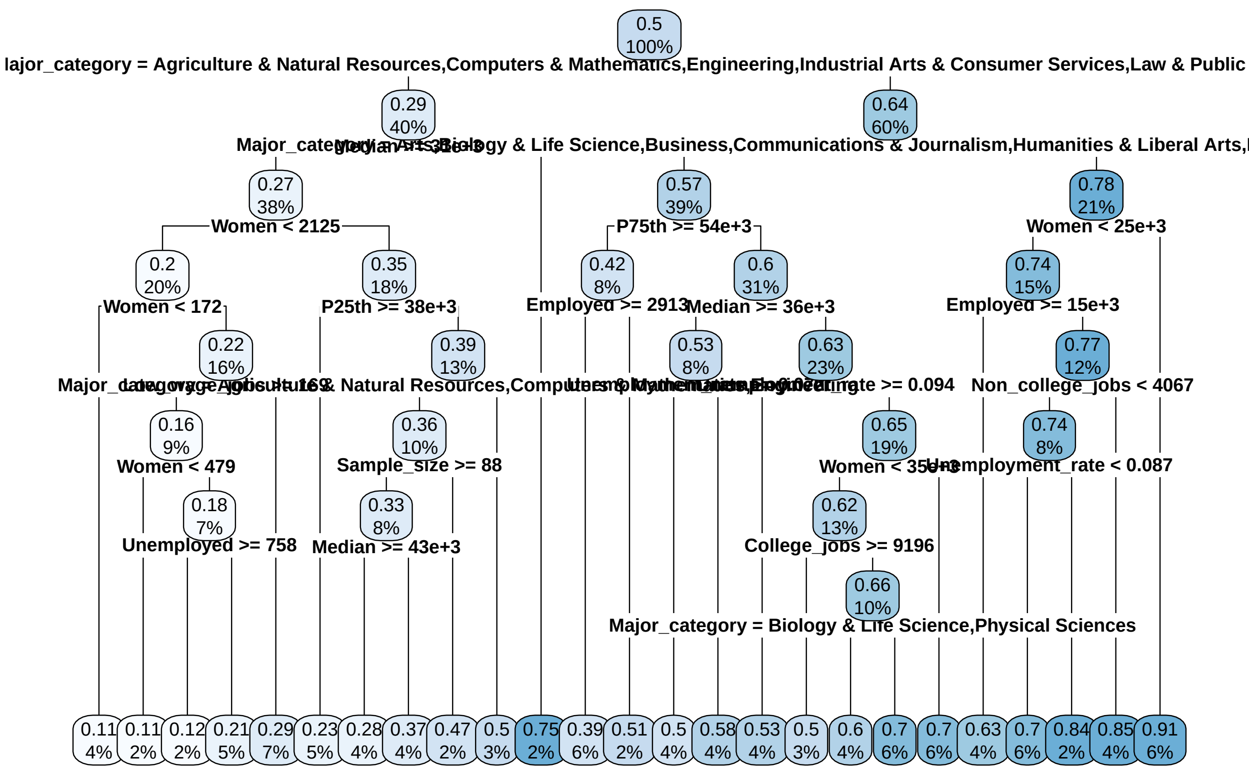
```
#?linear_reg
#?

model_03_D_Tree <- decision_tree( mode = "regression",
                                  cost_complexity = 0.001,
                                  tree_depth = 8,
                                  min_n = 10) %>%
  set_engine("rpart") %>%
  fit(Sharewomen~., data = train_data %>% select(-Rank, -Major,-Major_code, -Men, -Part_time ))

model_03_D_Tree %>%
  calc_metrics(new_data = test_data)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      0.0665
## 2 rsq     standard      0.922
## 3 mae     standard      0.0507
```

this above decision tree has a RSQ that is equal to 92% : Our model explain 92% of variation within our data- In addition, both of our RMSE & MAE are low, which might be a good sign so far

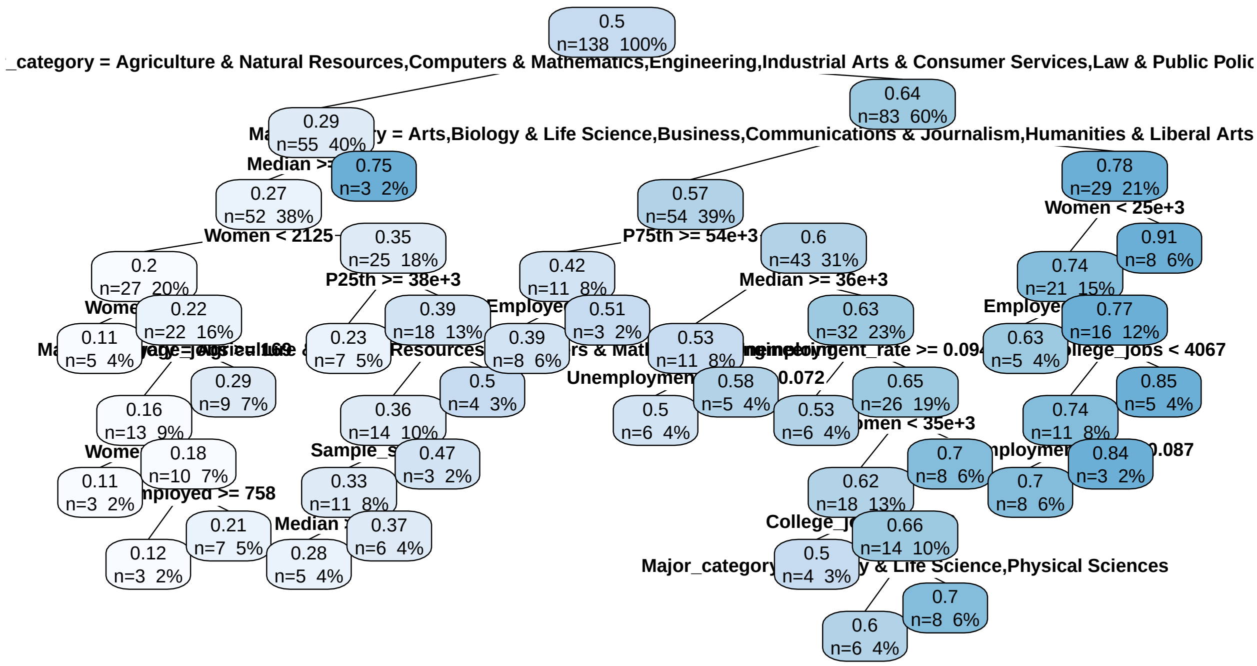
These plots are not easy to explain at all

```r
model_03_D_Tree$fit %>%
  rpart.plot(roundint = FALSE, cex = 0.6)
```




```r
model_03_D_Tree$fit %>%
    rpart.plot(roundint = FALSE,
               type = 2,
               extra = 101,
               fallen.leaves = FALSE,
               cex = 0.6,
               main = 'Model 04: Decision Tree')
```

Model 04: Decision Tree



RANDOM FOREST —-

Model: ranger —-

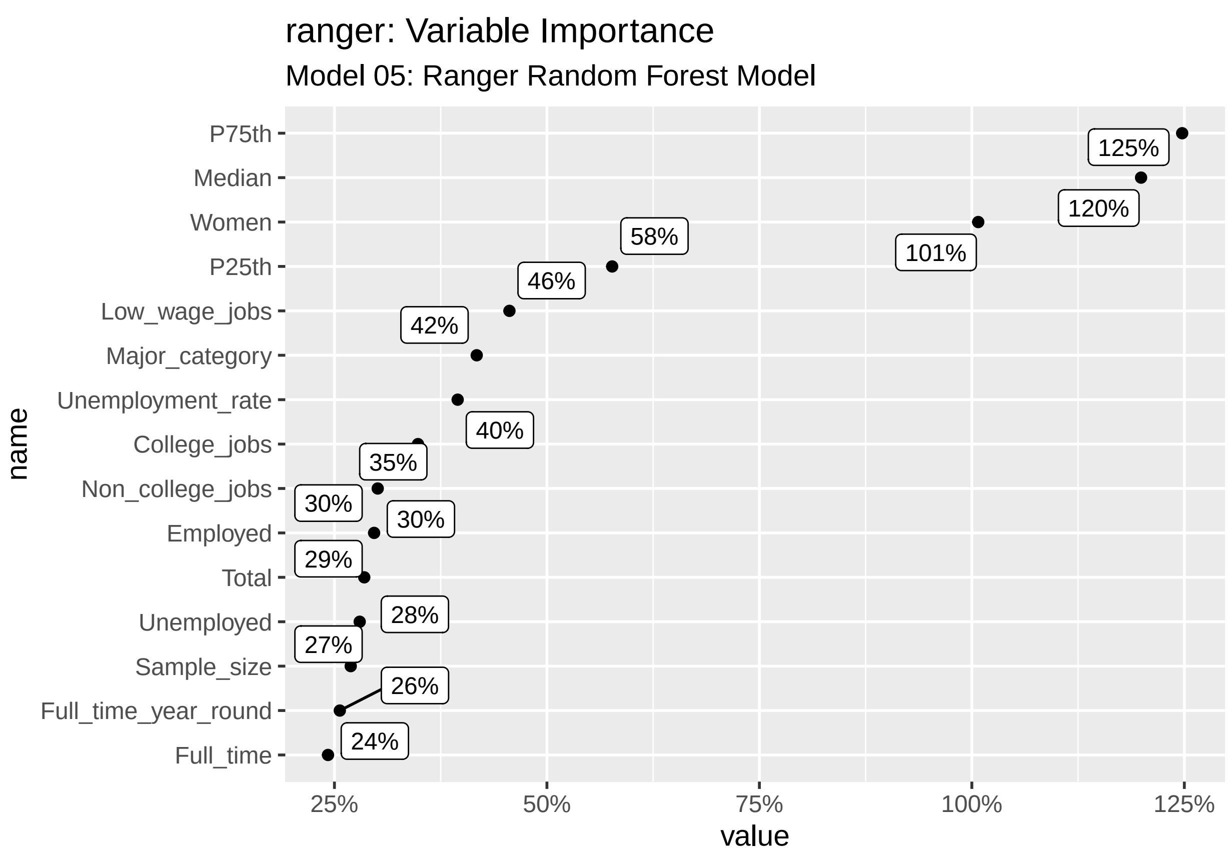
```
library(ranger)
#?rand_forest()
#?ranger::ranger

set.seed(1234)

model_04_rf_ranger <-rand_forest(mode = "regression", trees = 6000, min_n = 4) %>%
  set_engine("ranger", importance = "impurity") %>%

  fit(Sharewomen~., data = train_data %>% select(-Rank, -Major,-Major_code, -Men, -Part_time ))

model_04_rf_ranger %>%
  calc_metrics(new_data = test_data)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      0.0658
## 2 rsq     standard      0.954
## 3 mae     standard      0.0530
```

As we can see, our RSQ in the above model is really good too even though the MAE is a lit bit higher than the previous model by 0.002; If if have to choose between this model and the Tree based model, I would probably go for the Tree based model.

ranger: Feature Importance —-

```
model_04_rf_ranger$fit %>%
  ranger::importance() %>%
  enframe() %>%

  arrange(desc(value)) %>%
  mutate(name = as_factor(name) %>% fct_rev()) %>%

  ggplot(aes(value, name)) +
    geom_point() +
    ggrepel::geom_label_repel(aes(label = scales::percent(value, accuracy = 1)),
                              size = 3) +
    labs(title = "ranger: Variable Importance",
         subtitle = "Model 05: Ranger Random Forest Model") +
    scale_x_continuous(labels = scales::percent_format())
```



Model XGBOOST —-

```
library(xgboost)
```

```
##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```
?boost_tree
?xgboost::xgboost

set.seed(1234)

model_07_boost_xgboost <- boost_tree("regression") %>%
    set_engine("xgboost",
               mtry= 30,
               learn_rate = 0.25,
               tree_depth=7, objective = 'reg:squarederror') %>%
  fit(Sharewomen~., data = train_data %>% select(-Rank, -Major,-Major_code, -Men, -Part_time))
```

```
## [22:19:00] WARNING: amalgamation/../src/learner.cc:541:
## Parameters: { learn_rate, mtry, tree_depth } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindings but
##   passed down to XGBoost core.  Or some parameters are not used but slip through this
##   verification. Please open an issue if you find above cases.
```

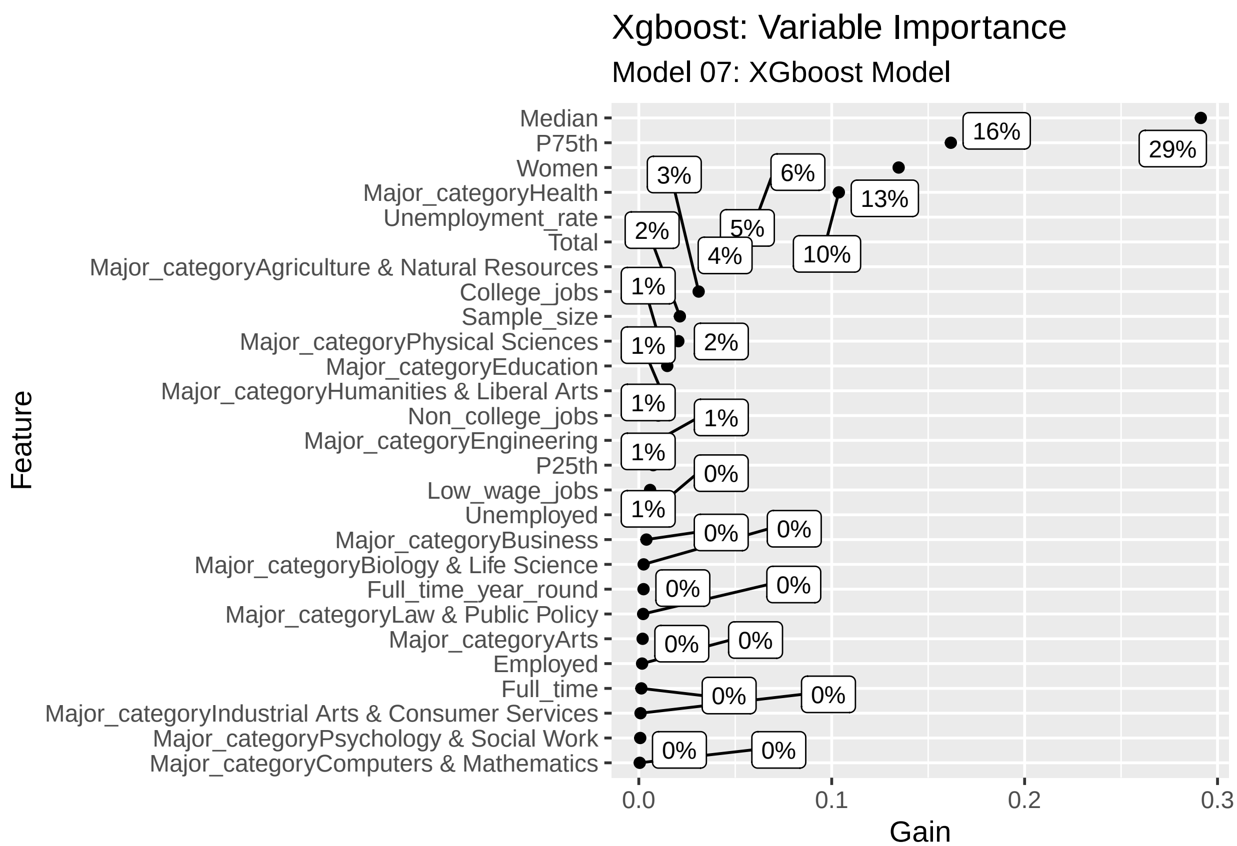
```
model_07_boost_xgboost %>% calc_metrics(test_data)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      0.0166
## 2 rsq     standard      0.996
## 3 mae     standard      0.0125
```

4.3.2 Feature Importance —-

```
model_07_boost_xgboost$fit %>%
  xgboost::xgb.importance(model = .) %>%
  as_tibble() %>%

  arrange(desc(Gain)) %>%
  mutate(Feature = as_factor(Feature) %>% fct_rev()) %>%

  ggplot(aes(Gain, Feature)) +
```

```r
geom_point() +
ggrepel::geom_label_repel(aes(label = scales::percent(Gain, accuracy = 1)),
                          size = 3) +

labs(
      title = "Xgboost: Variable Importance",
      subtitle = "Model 07: XGboost Model"
  )
```



OUr XGBOOST metrics outperformed all the above models. This model explained 99% of the data and as we can see, the Mean Absolute Error & Root Mean Square Error are considerably smaller than the others.