Is Wage Gap Real?

MGSC 310, Fall 2019

Group 5

```r
###### GRAPHS #####
rm(list = ls())
jobs_gender <- readr::read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
```

```
## Parsed with column specification:
## cols(
##   year = col_double(),
##   occupation = col_character(),
##   major_category = col_character(),
##   minor_category = col_character(),
##   total_workers = col_double(),
##   workers_male = col_double(),
##   workers_female = col_double(),
##   percent_female = col_double(),
##   total_earnings = col_double(),
##   total_earnings_male = col_double(),
##   total_earnings_female = col_double(),
##   wage_percent_of_male = col_double()
## )
```

```r
jobs_gender <- jobs_gender[complete.cases(jobs_gender), ]
numerical_jg <- jobs_gender[c(1,5:12)]

# Describing the Data
# year: (integer)   Year
# occupation:   (character) Specific job/career
# major_category:   (character) Broad category of occupation
# minor_category:   (character) Fine category of occupation
# total_workers:    (double)    Total estimated full-time workers > 16 years old
# workers_male: (double)    Estimated MALE full-time workers > 16 years old
# workers_female:   (double)    Estimated FEMALE full-time workers > 16 years old
# percent_female:   (double)    The percent of females for specific occupation
# total_earnings:   (double)    Total estimated median earnings for full-time workers > 16 years old
# total_earnings_male:  (double)    Estimated MALE median earnings for full-time workers > 16 years old
# total_earnings_female:    (double)    Estimated FEMALE median earnings for full-time workers > 16 yea
# wage_percent_of_male: (double)    Female wages as percent of male wages

# Summary tables of means, max, mins, and standard deviations
summary(numerical_jg)
```

```
##       year      total_workers      workers_male     workers_female
##  Min.   :2013   Min.   :  11383   Min.   :   5360   Min.   :   1333
##  1st Qu.:2014   1st Qu.:  61748   1st Qu.:  25674   1st Qu.:  20994
##  Median :2014   Median : 131104   Median :  63438   Median :  49108
##  Mean   :2015   Mean   : 309739   Mean   : 170211   Mean   : 139528
```

```
##  3rd Qu.:2016   3rd Qu.: 371588   3rd Qu.: 174450   3rd Qu.: 136992
##  Max.   :2016   Max.   :3758629   Max.   :2570385   Max.   :2290818
##  percent_female  total_earnings   total_earnings_male
##  Min.   : 1.20   Min.   : 17266   Min.   : 17302
##  1st Qu.:25.90   1st Qu.: 32318   1st Qu.: 36217
##  Median :46.91   Median : 46460   Median : 50250
##  Mean   :45.81   Mean   : 50968   Mean   : 55457
##  3rd Qu.:63.80   3rd Qu.: 62246   3rd Qu.: 69851
##  Max.   :98.01   Max.   :201542   Max.   :231420
##  total_earnings_female wage_percent_of_male
##  Min.   : 16771        Min.   : 50.87
##  1st Qu.: 30075        1st Qu.: 77.56
##  Median : 41753        Median : 85.16
##  Mean   : 46103        Mean   : 84.03
##  3rd Qu.: 56739        3rd Qu.: 90.62
##  Max.   :166388        Max.   :117.40
```

```r
sapply(numerical_jg, sd, na.rm = TRUE)
```

```
##                  year         total_workers          workers_male
##          1.119203e+00          4.511729e+05          2.854461e+05
##        workers_female        percent_female        total_earnings
##          2.618260e+05          2.455227e+01          2.456764e+04
##   total_earnings_male total_earnings_female  wage_percent_of_male
##          2.672805e+04          2.162012e+04          9.380084e+00
```

```r
# find all of categories
categories_major <- unique(jobs_gender[3])
categories_minor <- unique(jobs_gender[4])
print(categories_major)
```

```
## # A tibble: 8 x 1
##   major_category
##   <chr>
## 1 Management, Business, and Financial
## 2 Computer, Engineering, and Science
## 3 Education, Legal, Community Service, Arts, and Media
## 4 Healthcare Practitioners and Technical
## 5 Service
## 6 Sales and Office
## 7 Natural Resources, Construction, and Maintenance
## 8 Production, Transportation, and Material Moving
```

```r
print(categories_minor)
```

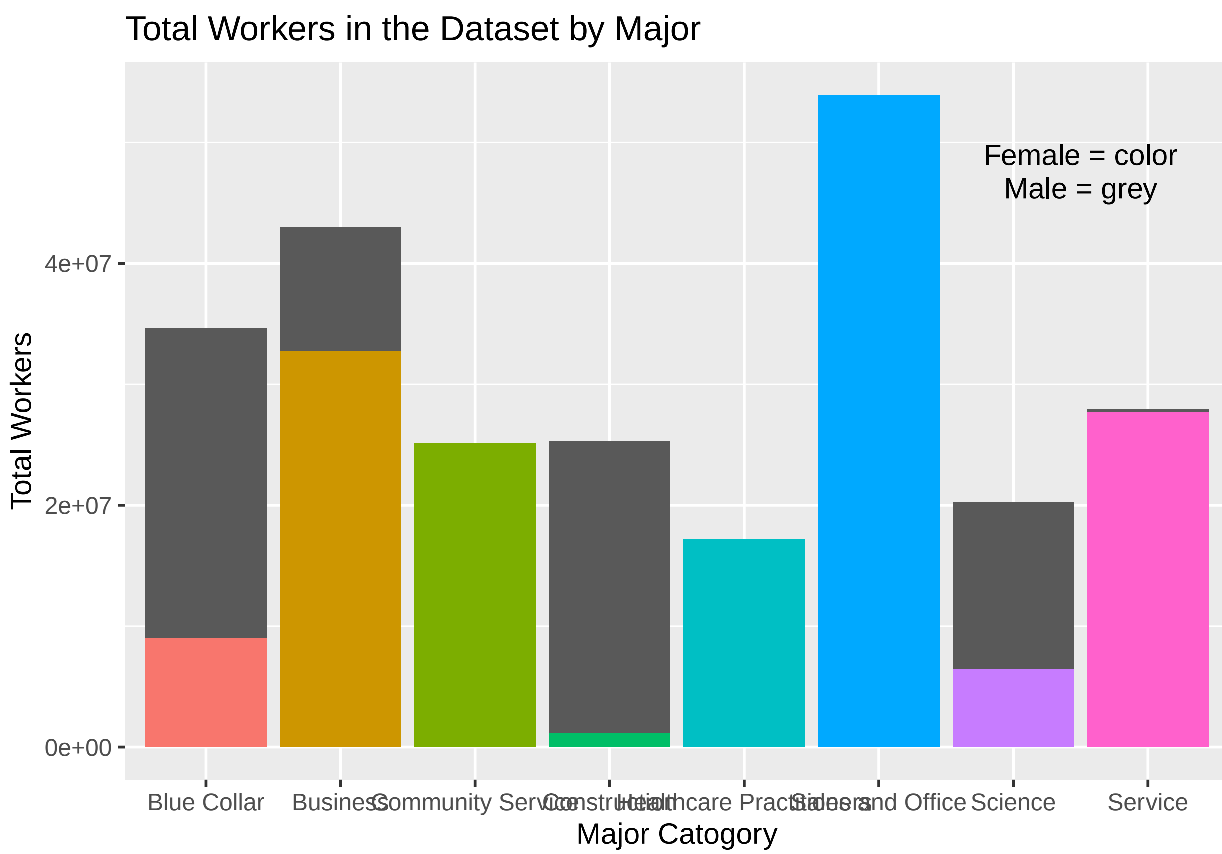
```
## # A tibble: 23 x 1
##    minor_category
##    <chr>
##  1 Management
##  2 Business and Financial Operations
##  3 Computer and mathematical
##  4 Architecture and Engineering
```

```
##   5 Life, Physical, and Social Science
##   6 Community and Social Service
##   7 Legal
##   8 Education, Training, and Library
##   9 Arts, Design, Entertainment, Sports, and Media
##  10 Healthcare Practitioners and Technical
## # ... with 13 more rows
```

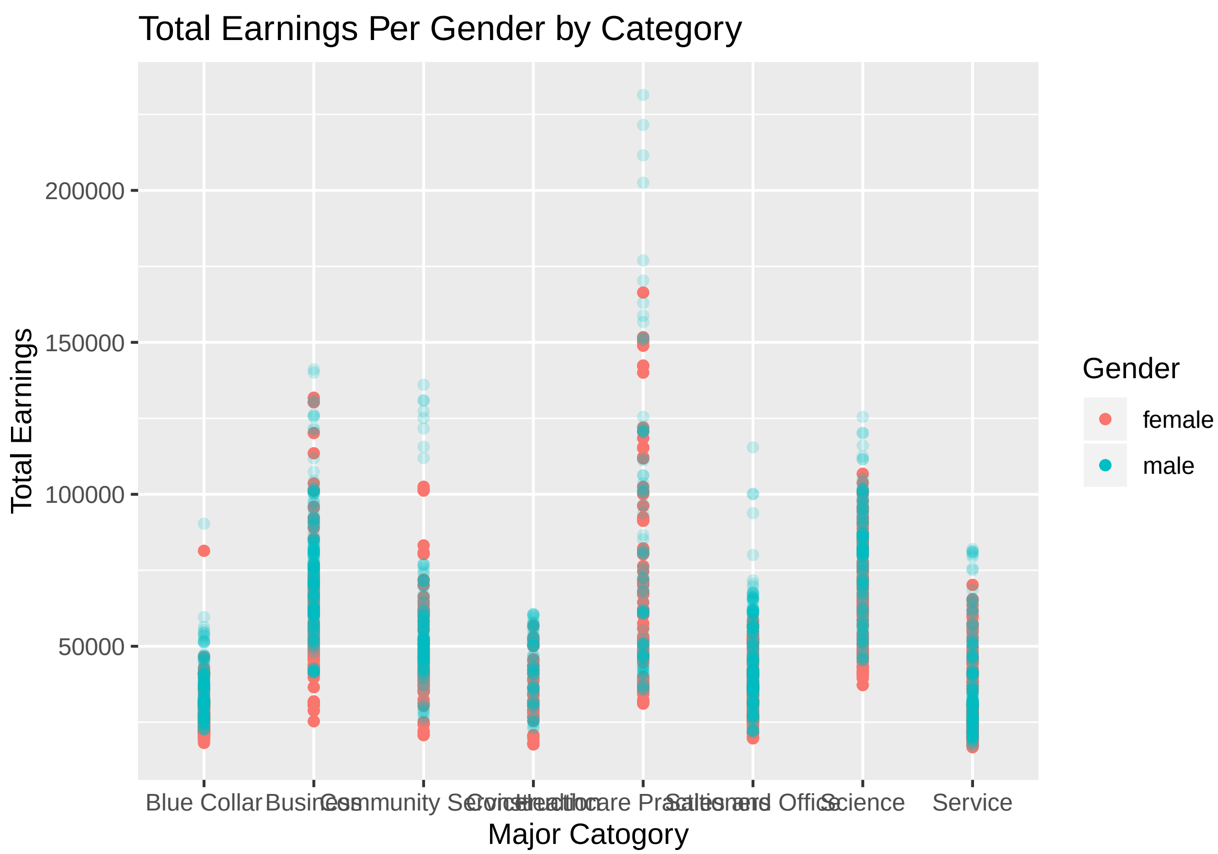
```
library(stringr)
clean_jg <- jobs_gender
clean_jg$year <- as.factor(clean_jg$year)
# changing the name of majors
clean_jg$major_category <- (gsub(",","",clean_jg$major_category))
clean_jg$major_category <- gsub('Management Business and Financial', 'Business', clean_jg$major_categor
clean_jg$major_category <- gsub('Computer Engineering and Science', 'Science', clean_jg$major_category)
clean_jg$major_category <- gsub('Education Legal Community Service Arts and Media', 'Community Service'
clean_jg$major_category <- gsub('Healthcare Practitioners and Technical', 'Healthcare Practitioners', c
clean_jg$major_category <- gsub('Natural Resources Construction and Maintenance', 'Construction', clean_
clean_jg$major_category <- gsub('Production Transportation and Material Moving', 'Blue Collar', clean_j
categories <- unique(clean_jg[3])
print(categories)
```

```
## # A tibble: 8 x 1
##   major_category
##   <chr>
## 1 Business
## 2 Science
## 3 Community Service
## 4 Healthcare Practitioners
## 5 Service
## 6 Sales and Office
## 7 Construction
## 8 Blue Collar
```

```
# adding columun of factor: who earns more in the job -> female or male?
library(formattable)
clean_jg$earns_more_female <- (clean_jg$total_earnings_female / clean_jg$total_earnings)
clean_jg$earns_more_female <-percent(clean_jg$earns_more_female)
clean_jg$earns_more_male <- (clean_jg$total_earnings_male / clean_jg$total_earnings)
clean_jg$earns_more_male <-percent(clean_jg$earns_more_male)


library(ggplot2)
# plot 1 - Total Workers in the Dataset by Major
ggplot(clean_jg)  +
  geom_bar(aes(major_category, workers_male),stat = "identity") +
  geom_bar(aes(major_category, workers_female, fill = major_category),stat = "identity") +
  labs(title ="Total Workers in the Dataset by Major", x = "Major Catogory", y = "Total Workers") +
  guides(fill=FALSE) + annotate("text", x = 7.5, y = 7000^2,label = "Female = color") +
  annotate("text", x = 7.5, y = (6800^2),label = "Male = grey")
```



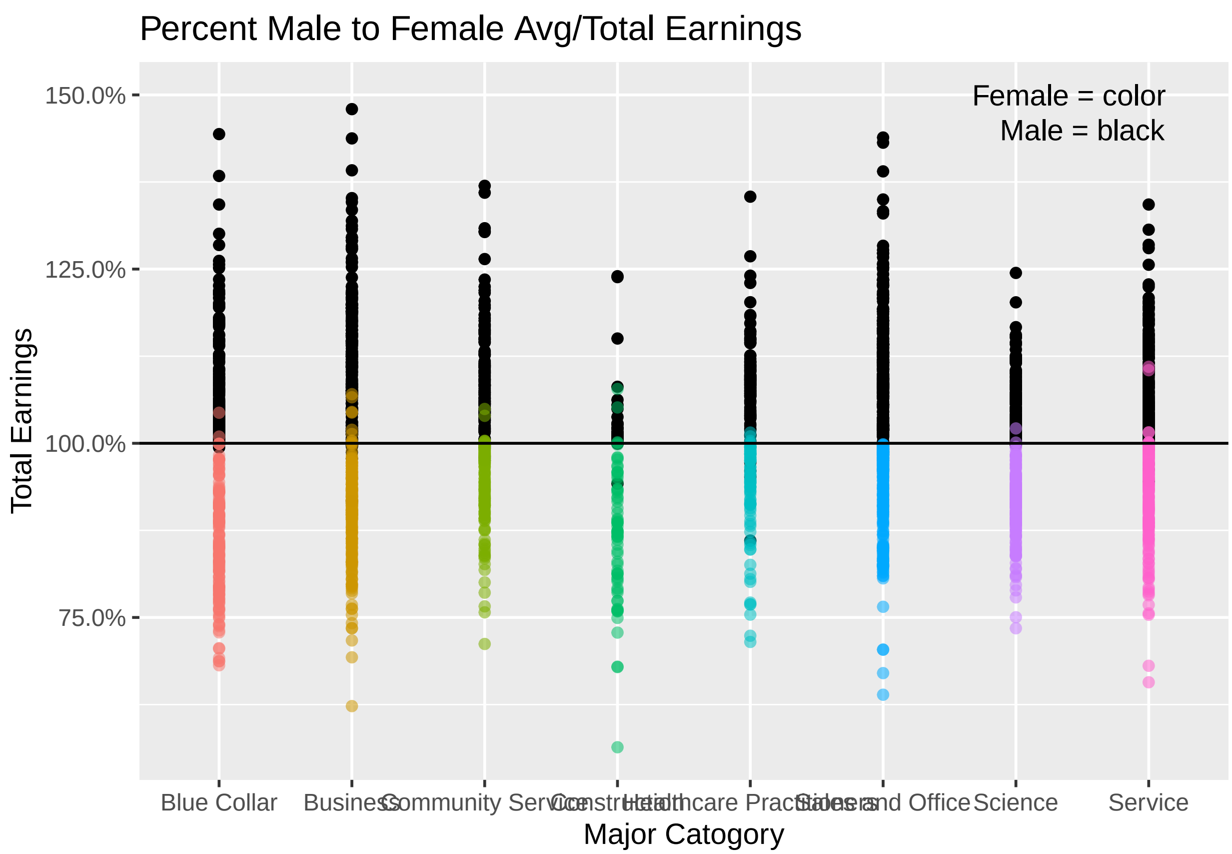

```r
# plot 2 - Total Earnings Per Gender
ggplot(clean_jg) +
  geom_point(aes(major_category, total_earnings_female, col = "female")) +
  geom_point(aes(major_category, total_earnings_male, col = "male"), alpha = .2) +
  labs(title ="Total Earnings Per Gender by Category", x = "Major Catogory", y = "Total Earnings") +
  labs(color = "Gender")
```



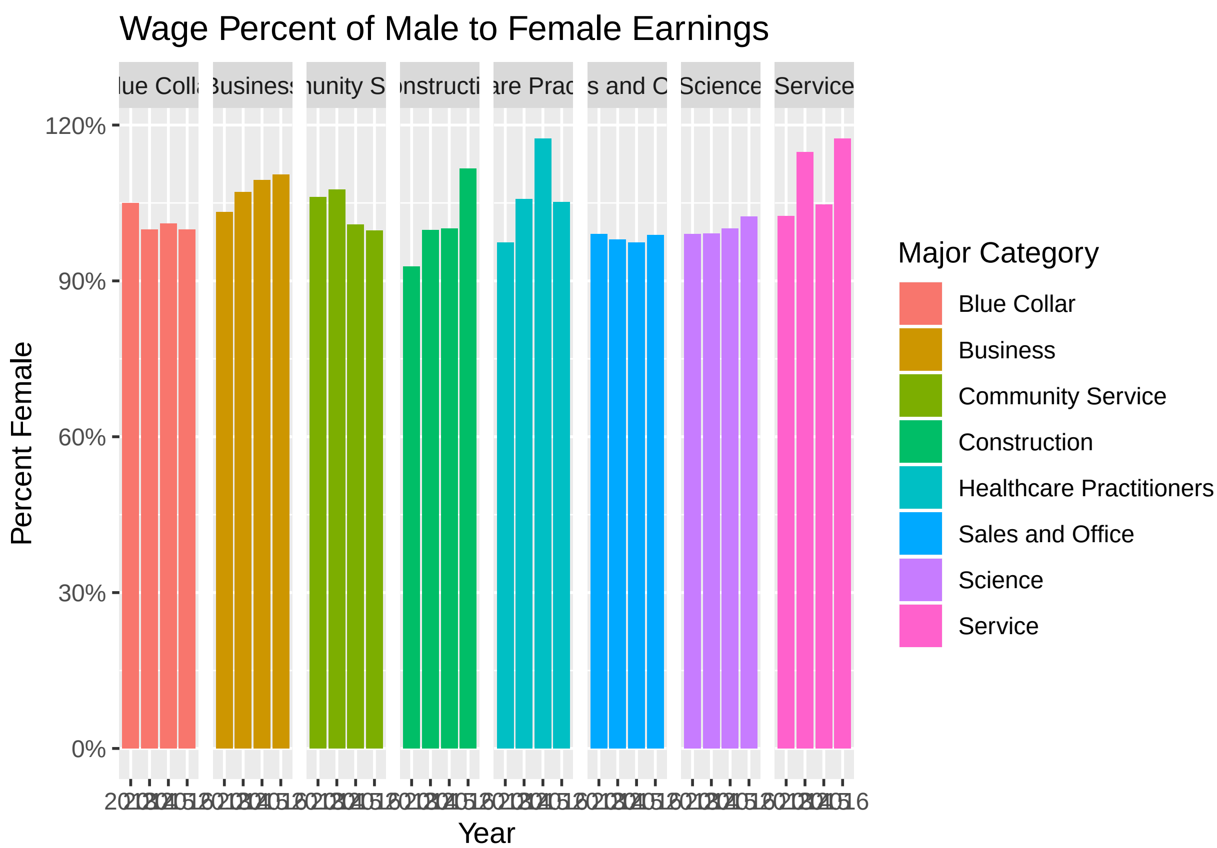

```r
# plot 3 - Who Earns More by Major
data(clean_jg, package = "reshape2")
```

```
## Warning in data(clean_jg, package = "reshape2"): data set 'clean_jg' not
## found
```

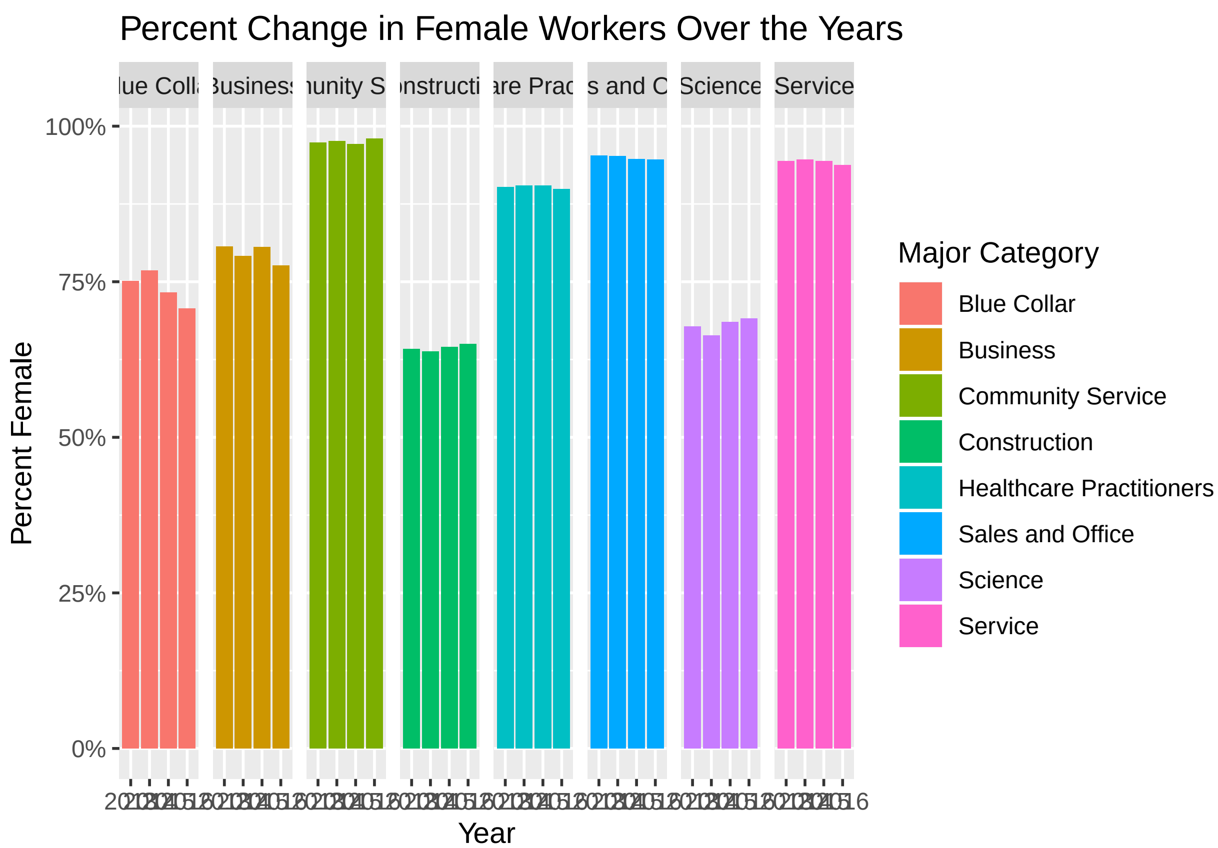
```r
ggplot(clean_jg, aes(x = major_category)) +
  geom_point(aes(y = (earns_more_male),fill = major_category), stat = "identity") +
  geom_point(aes(y = (earns_more_female),fill = major_category, color = major_category, alpha = .1),
             stat = "identity") + scale_y_continuous(labels=scales::percent) + geom_hline(yintercept = 1
  theme(legend.position = "none") + labs(title ="Percent Male to Female Avg/Total Earnings",
                                        x = "Major Catogory", y = "Total Earnings")  + annotate("text"
  annotate("text", x = 7.5, y = 1.45,label = "Male = black")
```




```r
# plot 4 - Wage Percent of Male to Female Earnings
ggplot(clean_jg) +
  geom_bar(aes(x = year, y = wage_percent_of_male,
               fill = major_category), stat = "identity", position = "dodge") +
  facet_grid(~ major_category) +
  labs(title ="Wage Percent of Male to Female Earnings",
       x = "Year", y = "Percent Female") +
  labs(fill = "Major Category") +
  scale_y_continuous(labels = function(x) paste0(x, "%"))
```




```r
# plot 5 - Percent Change in Female Workers Over the Years
ggplot(clean_jg) +
  geom_bar(aes(x = year, y = percent_female,
               fill = major_category), stat = "identity", position = "dodge") +
  facet_grid(~ major_category) +
  labs(title ="Percent Change in Female Workers Over the Years",
       x = "Year", y = "Percent Female") +
  labs(fill = "Major Category") +
  scale_y_continuous(labels = function(x) paste0(x, "%"))
```




```
#### CLEANING DATA ####
# Data transformation performed for feature engineering
jobs_gender <- jobs_gender[complete.cases(jobs_gender), ]
numerical_jg <- jobs_gender[c(1,5:12)]

library(stringr)
clean_jg <- jobs_gender
clean_jg$year <- as.factor(clean_jg$year)
# changing the name of majors
clean_jg$major_category <- (gsub(",","",clean_jg$major_category))
clean_jg$major_category <- gsub('Management Business and Financial', 'Business', clean_jg$major_categor
clean_jg$major_category <- gsub('Computer Engineering and Science', 'Science', clean_jg$major_category)
clean_jg$major_category <- gsub('Education Legal Community Service Arts and Media', 'Community Service'
clean_jg$major_category <- gsub('Healthcare Practitioners and Technical', 'Healthcare Practitioners', c
clean_jg$major_category <- gsub('Natural Resources Construction and Maintenance', 'Construction', clean_
clean_jg$major_category <- gsub('Production Transportation and Material Moving', 'Blue Collar', clean_j
categories <- unique(clean_jg[3])
print(categories)
```

```
## # A tibble: 8 x 1
##   major_category
##   <chr>
## 1 Business
## 2 Science
## 3 Community Service
## 4 Healthcare Practitioners
```

```
## 5 Service
## 6 Sales and Office
## 7 Construction
## 8 Blue Collar
```

```r
# adding columun of factor: who earns more in the job -> female or male?
library(formattable)
clean_jg$earns_more_female <- (clean_jg$total_earnings_female / clean_jg$total_earnings)

clean_jg$wage_gap <- clean_jg$total_earnings_male-clean_jg$total_earnings_female
clean_jg$gap_ratio <- clean_jg$wage_gap / clean_jg$total_earnings_female

clean_jg$woman_earn_more <- ifelse(clean_jg$wage_gap<0,1,0)

clean_jg$major_category <-  as.factor(clean_jg$major_category)
clean_jg$year <-  as.factor(clean_jg$year)
clean_jg <- clean_jg[c(-2,-4)]


########## PREDICTIONS ################
MSE <- function(p,t){mean((t-p)^2)} #predicted and true are input

  #### Percent Female Regression ####
# split into train and test
set.seed(2019)
trainSize <- 0.75
train_idx <- sample(1:nrow(clean_jg), size = floor(nrow(clean_jg) * trainSize))
train <- as.data.frame(clean_jg[train_idx,])
test <- as.data.frame(clean_jg[-train_idx,])

# look at the stats by major
library(doBy)
summaryBy(. ~ major_category, data = train)
```

```
##             major_category total_workers.mean workers_male.mean
## 1              Blue Collar           269047.4         207315.36
## 2                 Business           318558.2         187248.01
## 3        Community Service           268175.5         104432.31
## 4             Construction           395213.6         378329.78
## 5 Healthcare Practitioners           261133.6          71326.56
## 6          Sales and Office           416119.3         173602.90
## 7                  Science           170318.7         128797.14
## 8                  Service           273219.8         145121.23
##   workers_female.mean percent_female.mean total_earnings.mean
## 1            61732.05           33.152356            32828.81
## 2           131310.23           45.200420            65482.53
## 3           163743.17           59.484254            51361.27
## 4            16883.80            8.063453            42478.63
## 5           189807.00           64.118356            73050.90
## 6           242516.38           60.104476            40215.70
## 7            41521.55           29.466404            74618.50
## 8           128098.57           48.277778            33054.62
##   total_earnings_male.mean total_earnings_female.mean
## 1                 35620.56                   27805.59
```

```
## 2                 73091.06                     58748.16
## 3                 56564.40                     47472.75
## 4                 42931.45                     36714.12
## 5                 79828.79                     67863.99
## 6                 45037.34                     37083.56
## 7                 78354.11                     67585.01
## 8                 35394.17                     30240.98
##   wage_percent_of_male.mean earns_more_female.mean wage_gap.mean
## 1                  78.50042              0.8554324      7814.964
## 2                  80.97296              0.8988031     14342.904
## 3                  85.30965              0.9364460      9091.650
## 4                  85.49981              0.8665301      6217.333
## 5                  86.44119              0.9418571     11964.803
## 6                  84.06956              0.9320388      7953.778
## 7                  86.45350              0.9065947     10769.105
## 8                  86.22558              0.9250395      5153.188
##   gap_ratio.mean woman_earn_more.mean
## 1      0.2859813          0.000000000
## 2      0.2540438          0.061643836
## 3      0.1858470          0.048543689
## 4      0.1861473          0.058823529
## 5      0.1669863          0.014084507
## 6      0.2066492          0.000000000
## 7      0.1646816          0.008064516
## 8      0.1722855          0.020134228
```

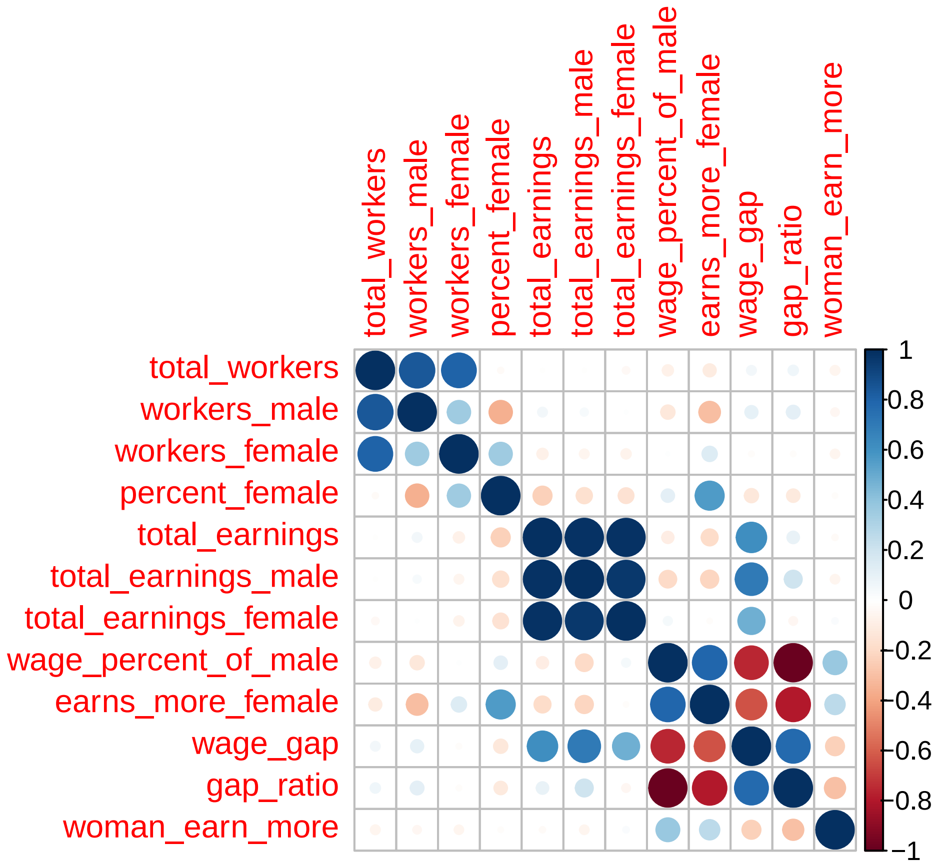
```r
# Correlation
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
sapply(clean_jg, class)
```

```
##                  year        major_category         total_workers
##              "factor"              "factor"             "numeric"
##          workers_male        workers_female        percent_female
##             "numeric"             "numeric"             "numeric"
##        total_earnings   total_earnings_male total_earnings_female
##             "numeric"             "numeric"             "numeric"
##  wage_percent_of_male     earns_more_female              wage_gap
##             "numeric"             "numeric"             "numeric"
##             gap_ratio       woman_earn_more
##             "numeric"             "numeric"
```

```r
cor_dataframe <- clean_jg[,c(-1,-2)]
cor <- cor(cor_dataframe)
corrplot(cor)
```




```r
x <- cor[,4] # percent_female correlation
abs_x <- abs(x)
tail(sort(abs_x),8)
```

```
##              wage_gap total_earnings_female   total_earnings_male
##             0.1273893             0.1575816             0.1674997
##        total_earnings        workers_female          workers_male
##             0.2332207             0.3525289             0.3571848
##     earns_more_female        percent_female
##             0.5603202             1.0000000
```

```r
# top variables: earns_more_female, total_earnings, total_earnings_male, total_earnings_female, wage_ga
# variables of workers cant be used

# lasso model
library(glmnet)
```

```
## Loading required package: Matrix
```

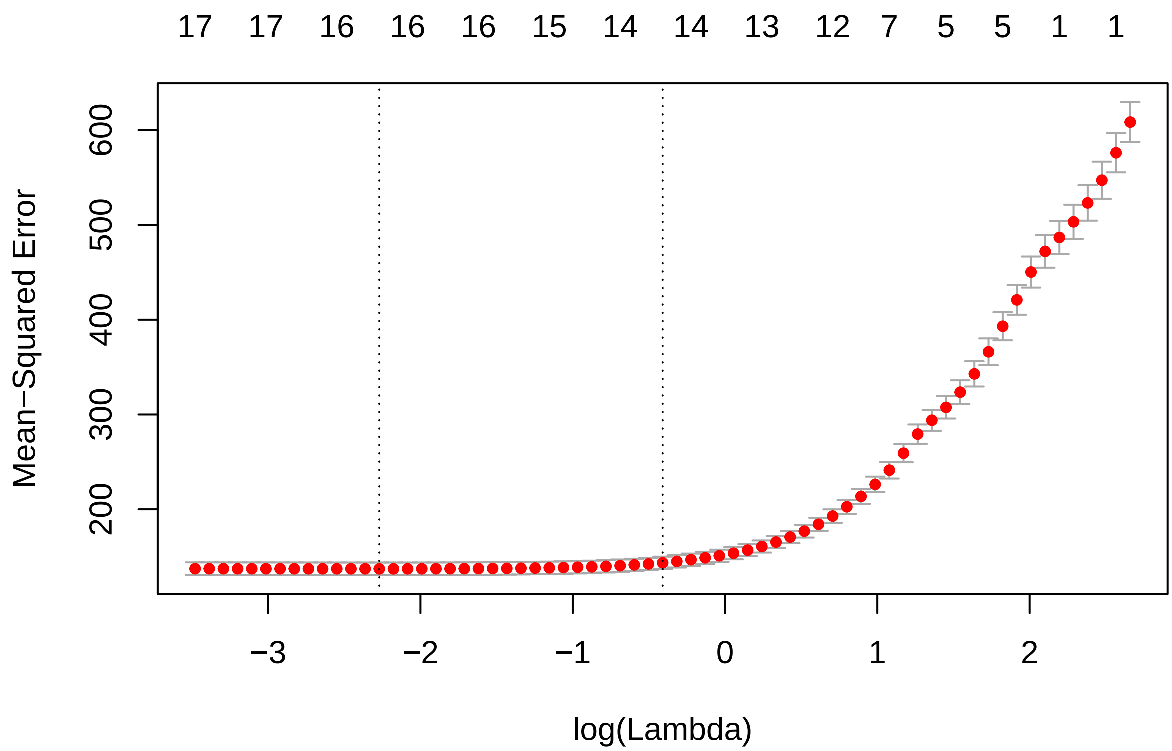
```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-18
```

```r
library(glmnetUtils)
```

```
##
## Attaching package: 'glmnetUtils'

## The following objects are masked from 'package:glmnet':
##
##     cv.glmnet, glmnet
```

```r
lasso_mod <- cv.glmnet(percent_female ~ .,
                       data = train, alpha = 1)
plot(lasso_mod)
```




```r
coefs <- data.frame(
  lasso_lambda_min = as.matrix(round(coef(lasso_mod, s = "lambda.min"),3)),
  lasso_lambda_1se = as.matrix(round(coef(lasso_mod, s = "lambda.1se"),3)))

colnames(coefs) <- c("Lasso Min","Lasso 1se")
print(coefs)
```

```
##                                          Lasso Min Lasso 1se
## (Intercept)                                -183.423  -154.253
## year2013                                      0.000     0.000
```

```
## year2014                                    0.218     0.000
## year2015                                    0.000     0.000
## year2016                                    0.000     0.000
## major_categoryBlue Collar                  -4.724    -2.581
## major_categoryBusiness                      0.000     0.000
## major_categoryCommunity Service             5.434     4.752
## major_categoryConstruction                -16.653   -15.164
## major_categoryHealthcare Practitioners     11.949     9.712
## major_categorySales and Office              2.798     3.425
## major_categoryScience                      -6.472    -6.711
## major_categoryService                      -1.869     0.000
## total_workers                               0.000     0.000
## workers_male                                0.000     0.000
## workers_female                              0.000     0.000
## total_earnings                              0.000     0.000
## total_earnings_male                         0.000     0.000
## total_earnings_female                       0.000     0.000
## wage_percent_of_male                       -0.582    -0.638
## earns_more_female                         304.335   275.253
## wage_gap                                    0.000     0.000
## gap_ratio                                  44.960    40.686
## woman_earn_more                           -11.637    -8.667
```

```r
# which variables are selected:
# Lasso Min: year, major_category, wage_percent_of_male, earns_more_female, gap_ratio, woman_earn_more

# more managable set of variables no need for lasso 1se
# Lasso 1se: year, major_category, wage_percent_of_male, earns_more_female, gap_ratio, woman_earn_more

# lambda min values
lasso_mod$lambda.min
```

```
## [1] 0.1032929
```

```r
# lambda 1se values
lasso_mod$lambda.1se
```

```
## [1] 0.6639746
```

```r
# MSE of lasso
indx <- which(lasso_mod$lambda == lasso_mod$lambda.min)
lasso_mod$cvm[indx]
```

```
## [1] 137.2418
```

```r
# has the lowest MSE

mod_cor <- lm(percent_female ~ earns_more_female+total_earnings+total_earnings_male+total_earnings_femal
              data=train)
summary(mod_cor)
```

```
##
## Call:
## lm(formula = percent_female ~ earns_more_female + total_earnings +
##     total_earnings_male + total_earnings_female + wage_gap, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.133  -7.696   0.632   9.327  58.465
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.934e+02  1.415e+01 -13.672  < 2e-16 ***
## earns_more_female      2.687e+02  1.552e+01  17.315  < 2e-16 ***
## total_earnings        -2.540e-03  3.159e-04  -8.041  2.7e-15 ***
## total_earnings_male    3.084e-03  1.283e-04  24.043  < 2e-16 ***
## total_earnings_female -1.015e-03  2.824e-04  -3.594 0.000343 ***
## wage_gap                     NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.13 on 926 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.6247
## F-statistic: 388.1 on 4 and 926 DF,  p-value: < 2.2e-16
```

```
mod_lasso <- lm(percent_female ~ as.factor(year)+as.factor(major_category)+wage_percent_of_male+earns_mo
                  gap_ratio+woman_earn_more,data=train)
summary(mod_lasso)
```

```
##
## Call:
## lm(formula = percent_female ~ as.factor(year) + as.factor(major_category) +
##     wage_percent_of_male + earns_more_female + gap_ratio + woman_earn_more,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.083  -7.732   0.025   8.082  64.376
##
## Coefficients:
##                                                     Estimate Std. Error
## (Intercept)                                        -249.7644    32.3156
## as.factor(year)2014                                   0.8468     1.1983
## as.factor(year)2015                                   0.1431     1.2194
## as.factor(year)2016                                   0.1209     1.2014
## as.factor(major_category)Business                     1.0247     1.6782
## as.factor(major_category)Community Service            9.6131     1.8622
## as.factor(major_category)Construction               -15.1758     2.2530
## as.factor(major_category)Healthcare Practitioners   14.0042     2.0610
## as.factor(major_category)Sales and Office            8.5499     1.6808
## as.factor(major_category)Science                    -7.2153     1.7478
## as.factor(major_category)Service                     4.0324     1.6976
## wage_percent_of_male                                -0.7929     0.3243
## earns_more_female                                  377.4066    10.9530
## gap_ratio                                           77.0493    20.5965
```

```
## woman_earn_more                                    -14.9029     3.4827
##                                                   t value Pr(>|t|)
## (Intercept)                                        -7.729 2.85e-14 ***
## as.factor(year)2014                                 0.707 0.479925
## as.factor(year)2015                                 0.117 0.906629
## as.factor(year)2016                                 0.101 0.919861
## as.factor(major_category)Business                   0.611 0.541598
## as.factor(major_category)Community Service          5.162 2.99e-07 ***
## as.factor(major_category)Construction              -6.736 2.88e-11 ***
## as.factor(major_category)Healthcare Practitioners   6.795 1.95e-11 ***
## as.factor(major_category)Sales and Office           5.087 4.42e-07 ***
## as.factor(major_category)Science                   -4.128 3.99e-05 ***
## as.factor(major_category)Service                    2.375 0.017738 *
## wage_percent_of_male                               -2.445 0.014671 *
## earns_more_female                                  34.457  < 2e-16 ***
## gap_ratio                                           3.741 0.000195 ***
## woman_earn_more                                    -4.279 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13 on 916 degrees of freedom
## Multiple R-squared:  0.7274, Adjusted R-squared:  0.7233
## F-statistic: 174.6 on 14 and 916 DF,  p-value: < 2.2e-16
```

```r
# add prediction into the dataframe
# mod_cor
scores_train <- predict(mod_cor)
scores_test <- predict(mod_cor,newdata=test)
```

```
## Warning in predict.lm(mod_cor, newdata = test): prediction from a rank-
## deficient fit may be misleading
```

```r
train$scores_train_cor <- scores_train
test$scores_test_cor <- scores_test
# mod_lasso
scores_train <- predict(mod_lasso)
scores_test <- predict(mod_lasso,newdata=test)
train$scores_train_lasso <- scores_train
test$scores_test_lasso <- scores_test

# mod_cor
MSE(train$scores_train_cor,train$percent_female)
```

```
## [1] 227.7927
```

```r
MSE(test$scores_test_cor,test$percent_female)
```

```
## [1] 291.909
```

```r
# mod_lasso
MSE(train$scores_train_lasso,train$percent_female)
```

```
## [1] 166.1629
```

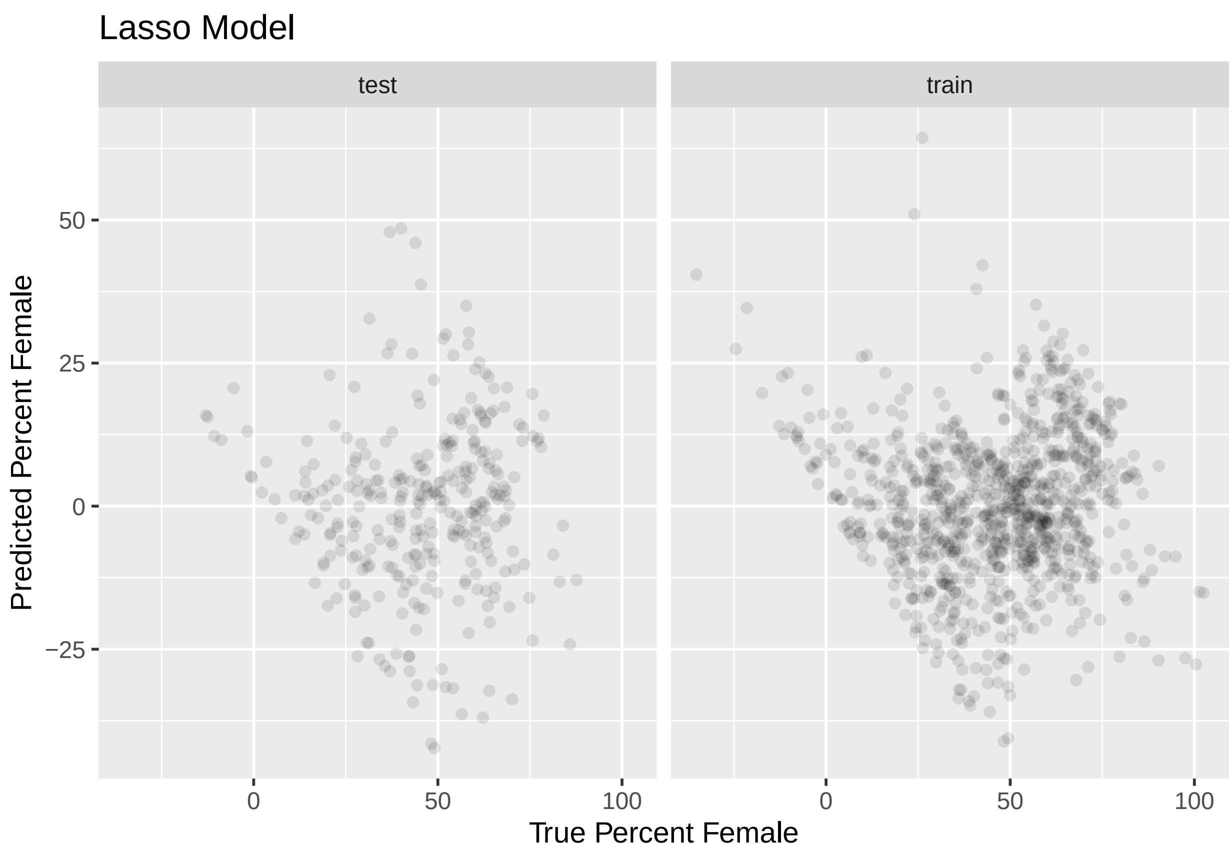
```r
MSE(test$scores_test_lasso,test$percent_female)
```

```
## [1] 219.6469
```

```r
#plot correlation predicted vs. true for train & test
library(ggplot2)
resids_train_cor <- train$percent_female - train$scores_train_cor
resids_test_cor <- test$percent_female - test$scores_test_cor

preds_df_cor <- data.frame(preds = c(train$scores_train_cor,test$scores_test_cor),
                          resids = c(resids_train_cor,resids_test_cor),
                          type = c(rep("train",nrow(train)),rep("test",nrow(test))))

ggplot(preds_df_cor, aes(x = preds, y = resids)) + geom_point(alpha = 1/10) +
  facet_wrap(~type) + labs(x = "True Percent Female", y = "Predicted Percent Female") +
  labs(title="Correlation Model")
```


```r
#plot lasso predicted vs. true for train & test
resids_train_lasso <- train$percent_female - train$scores_train_lasso
resids_test_lasso <- test$percent_female - test$scores_test_lasso

preds_df_lasso <- data.frame(preds = c(train$scores_train_lasso,test$scores_test_lasso),
                            resids = c(resids_train_lasso,resids_test_lasso),
```

```r
                                type = c(rep("train",nrow(train)),rep("test",nrow(test))))

ggplot(preds_df_lasso, aes(x = preds, y = resids)) + geom_point(alpha = 1/10) +
  facet_wrap(~type) + labs(x = "True Percent Female", y = "Predicted Percent Female") +
  labs(title="Lasso Model")
```




```r
  #### Gap Ratio Regression ####
wage_data <- data.frame(clean_jg$major_category,clean_jg$gap_ratio)
library('scales')
```

```
##
## Attaching package: 'scales'

## The following objects are masked from 'package:formattable':
##
##     comma, percent, scientific
```

```r
sort_gap_ratio_DF <- wage_data[order((wage_data$clean_jg.gap_ratio)),]
sort_gap_ratio_DF$clean_jg.gap_ratio <- percent(sort_gap_ratio_DF$clean_jg.gap_ratio)
dim(sort_gap_ratio_DF)
```

```
## [1] 1242    2
```

```r
print(sort_gap_ratio_DF[1:10,]) #the lowest wage gap
```

```
##       clean_jg.major_category clean_jg.gap_ratio
## 1091                  Service               -15%
## 750  Healthcare Practitioners               -15%
## 505                   Service               -13%
## 1189             Construction               -10%
## 969                  Business               -10%
## 650                  Business                -9%
## 1195             Construction                -7%
## 429         Community Service                -7%
## 324                  Business                -7%
## 118         Community Service                -6%
```

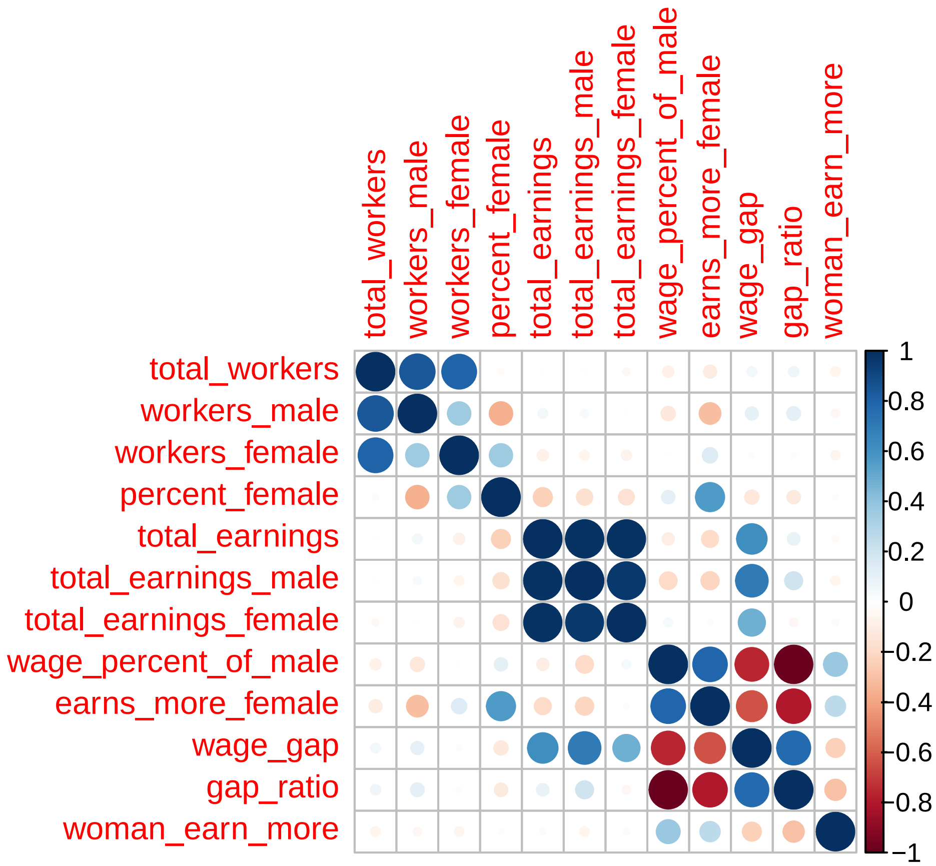
```r
print(sort_gap_ratio_DF[1232:1242,]) #the biggest wage gap
```

```
##      clean_jg.major_category clean_jg.gap_ratio
## 610              Blue Collar                71%
## 1099                 Service                72%
## 606              Blue Collar                74%
## 519         Sales and Office                75%
## 830         Sales and Office                75%
## 1155        Sales and Office                78%
## 579             Construction                79%
## 208         Sales and Office                83%
## 674                 Business                83%
## 363                 Business                86%
## 1140        Sales and Office                97%
```

```r
# split into train and test
set.seed(2019)
trainSize <- 0.75
train_idx <- sample(1:nrow(clean_jg), size = floor(nrow(clean_jg) * trainSize))
train <- as.data.frame(clean_jg[train_idx,])
test <- as.data.frame(clean_jg[-train_idx,])

# Correlation
library(corrplot)
sapply(clean_jg, class)
```

```
##                  year        major_category         total_workers
##              "factor"              "factor"             "numeric"
##          workers_male        workers_female        percent_female
##             "numeric"             "numeric"             "numeric"
##        total_earnings   total_earnings_male total_earnings_female
##             "numeric"             "numeric"             "numeric"
##   wage_percent_of_male     earns_more_female              wage_gap
##             "numeric"             "numeric"             "numeric"
##             gap_ratio       woman_earn_more
##             "numeric"             "numeric"
```

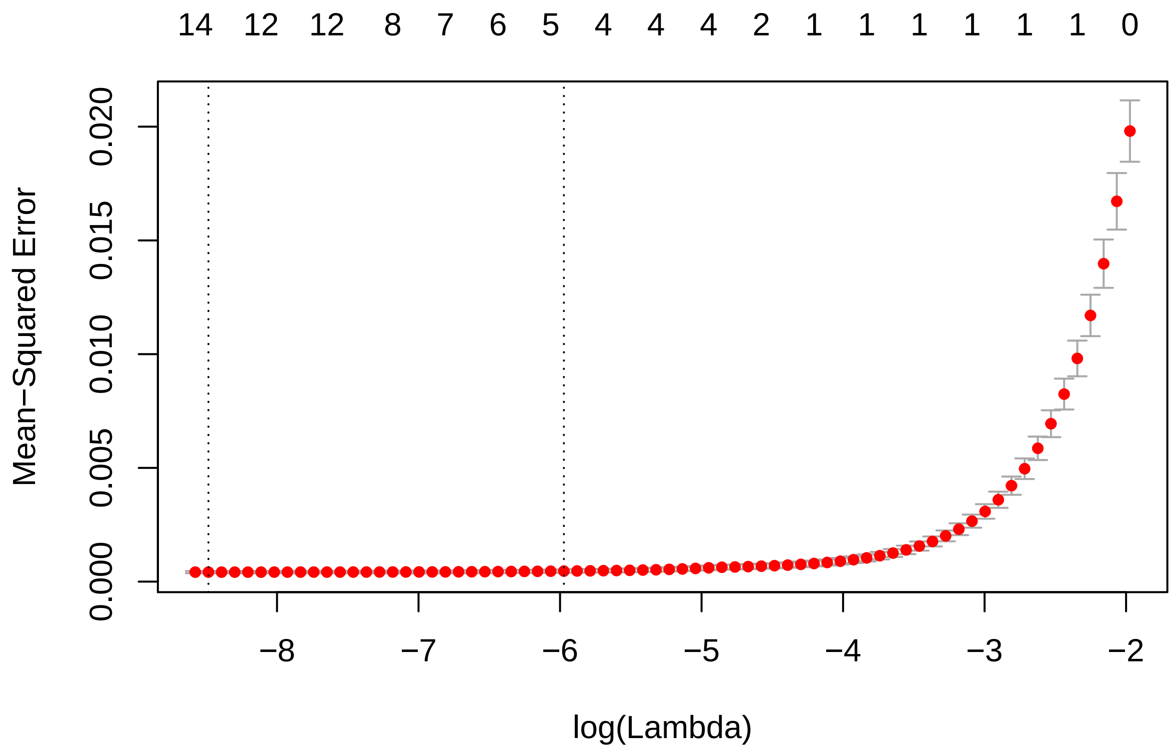
```r
cor_dataframe <- clean_jg[,c(-1,-2)]
cor <- cor(cor_dataframe)
corrplot(cor)
```




```r
x <- cor[,11] # gap_ratio correlation
abs_x <- abs(x)
tail(sort(abs_x),8)
```

```
##       percent_female         workers_male  total_earnings_male
##            0.1129286            0.1182477            0.2057956
##       woman_earn_more             wage_gap    earns_more_female
##            0.2912414            0.7701468            0.7936507
## wage_percent_of_male            gap_ratio
##            0.9841549            1.0000000
```

```r
# top variables: earns_more_female, wage_gap, woman_earn_more, total_earnings_male, workers_male
# cant use wage_percent_of_male as it is the same as gap_ratio but for males

# lasso model
library(glmnet)
library(glmnetUtils)
lasso_mod <- cv.glmnet(gap_ratio ~ .,
                       data = train, alpha = 1)
plot(lasso_mod)
```




```r
coefs <- data.frame(
  lasso_lambda_min = as.matrix(round(coef(lasso_mod, s = "lambda.min"),3)),
  lasso_lambda_1se = as.matrix(round(coef(lasso_mod, s = "lambda.1se"),3)))

colnames(coefs) <- c("Lasso Min","Lasso 1se")
print(coefs)
```

```
##                                        Lasso Min Lasso 1se
## (Intercept)                                1.454     1.456
## year2013                                   0.000     0.000
## year2014                                   0.000     0.000
## year2015                                   0.000     0.000
## year2016                                   0.000     0.000
## major_categoryBlue Collar                  0.000     0.000
## major_categoryBusiness                    -0.001     0.000
## major_categoryCommunity Service            0.000     0.000
## major_categoryConstruction                 0.001     0.000
## major_categoryHealthcare Practitioners     0.000     0.000
## major_categorySales and Office             0.005     0.000
## major_categoryScience                      0.000     0.000
## major_categoryService                      0.001     0.000
## total_workers                              0.000     0.000
## workers_male                               0.000     0.000
## workers_female                             0.000     0.000
## percent_female                             0.000     0.000
```

```
## total_earnings                            0.000     0.000
## total_earnings_male                       0.000     0.000
## total_earnings_female                     0.000     0.000
## wage_percent_of_male                     -0.014    -0.015
## earns_more_female                        -0.117    -0.029
## wage_gap                                  0.000     0.000
## woman_earn_more                           0.074     0.051
```

```r
# which variables are selected:
# Lasso Min: major_category, wage_percent_of_male, earns_more_female, woman_earn_more

# more managable set of variables no need for lasso 1se
# Lasso 1se: wage_percent_of_male, earns_more_female, woman_earn_more

# lambda min values
lasso_mod$lambda.min
```

```
## [1] 0.0002065795
```

```r
# lambda 1se values
lasso_mod$lambda.1se
```

```
## [1] 0.002546808
```

```r
# MSE of lasso
indx <- which(lasso_mod$lambda == lasso_mod$lambda.min)
lasso_mod$cvm[indx]
```

```
## [1] 0.0004172239
```

```r
# has the lowest MSE

#### Gap_ratio Regression Model
mod_cor <- lm(gap_ratio ~ earns_more_female+wage_gap+woman_earn_more+total_earnings_male+workers_male,
              data=train)
summary(mod_cor)
```

```
##
## Call:
## lm(formula = gap_ratio ~ earns_more_female + wage_gap + woman_earn_more +
##     total_earnings_male + workers_male, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.295774 -0.028941 -0.004706  0.021379  0.287102
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.115e-01  3.239e-02  25.057  < 2e-16 ***
## earns_more_female   -6.519e-01  3.468e-02 -18.800  < 2e-16 ***
## wage_gap             1.645e-05  4.165e-07  39.492  < 2e-16 ***
```

```
## woman_earn_more     -3.784e-04  1.153e-02  -0.033    0.974
## total_earnings_male -2.860e-06  9.990e-08 -28.624  < 2e-16 ***
## workers_male        -3.472e-08  6.706e-09  -5.177 2.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05101 on 925 degrees of freedom
## Multiple R-squared:  0.8702, Adjusted R-squared:  0.8695
## F-statistic:  1241 on 5 and 925 DF,  p-value: < 2.2e-16
```

```r
mod_lasso <- lm(gap_ratio ~ as.factor(major_category)+wage_percent_of_male+earns_more_female+woman_earn
                data=train)
summary(mod_lasso)
```

```
##
## Call:
## lm(formula = gap_ratio ~ as.factor(major_category) + wage_percent_of_male +
##     earns_more_female + woman_earn_more, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.034400 -0.012550 -0.006067  0.007375  0.156339
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                       1.5420699  0.0093532
## as.factor(major_category)Business                 0.0039535  0.0026837
## as.factor(major_category)Community Service        0.0054705  0.0029741
## as.factor(major_category)Construction             0.0026097  0.0036049
## as.factor(major_category)Healthcare Practitioners  0.0069376  0.0032902
## as.factor(major_category)Sales and Office         0.0109709  0.0026657
## as.factor(major_category)Science                  0.0026601  0.0027965
## as.factor(major_category)Service                  0.0072233  0.0027072
## wage_percent_of_male                             -0.0151857  0.0001372
## earns_more_female                                -0.0748241  0.0173556
## woman_earn_more                                   0.0795080  0.0049217
##                                                   t value Pr(>|t|)
## (Intercept)                                       164.871  < 2e-16 ***
## as.factor(major_category)Business                   1.473  0.14105
## as.factor(major_category)Community Service          1.839  0.06618 .
## as.factor(major_category)Construction               0.724  0.46929
## as.factor(major_category)Healthcare Practitioners    2.109  0.03525 *
## as.factor(major_category)Sales and Office           4.116 4.21e-05 ***
## as.factor(major_category)Science                    0.951  0.34174
## as.factor(major_category)Service                    2.668  0.00776 **
## wage_percent_of_male                             -110.705  < 2e-16 ***
## earns_more_female                                  -4.311 1.80e-05 ***
## woman_earn_more                                    16.154  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02081 on 920 degrees of freedom
## Multiple R-squared:  0.9785, Adjusted R-squared:  0.9783
## F-statistic:  4191 on 10 and 920 DF,  p-value: < 2.2e-16
```

```r
# add prediction into the dataframe
# mod_cor
scores_train <- predict(mod_cor)
scores_test <- predict(mod_cor,newdata=test)
train$scores_train_cor <- scores_train
test$scores_test_cor <- scores_test
# mod_lasso
scores_train <- predict(mod_lasso)
scores_test <- predict(mod_lasso,newdata=test)
train$scores_train_lasso <- scores_train
test$scores_test_lasso <- scores_test

# mod_cor
MSE(train$scores_train_cor,train$percent_female)
```

```
## [1] 2705.148
```

```r
MSE(test$scores_test_cor,test$percent_female)
```

```
## [1] 2618.349
```

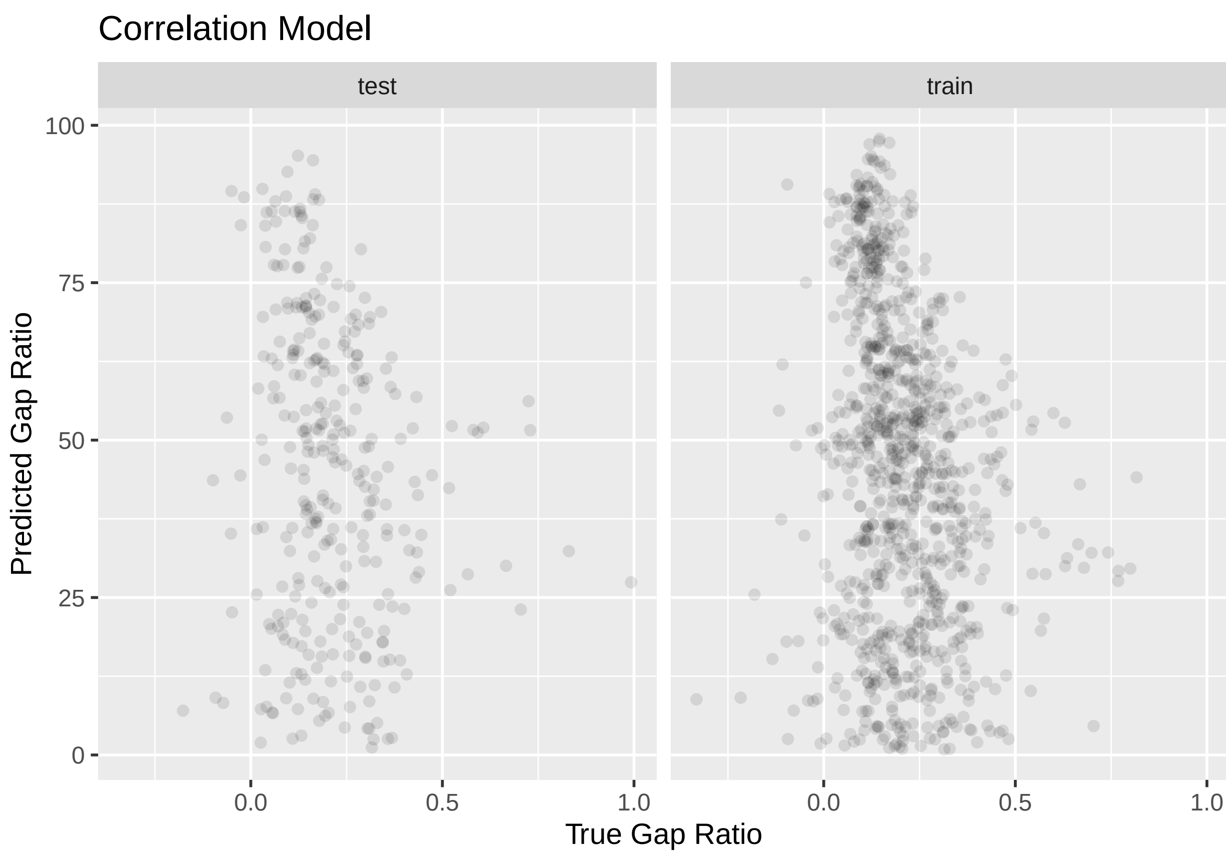
```r
# mod_lasso
MSE(train$scores_train_lasso,train$percent_female)
```

```
## [1] 2704.669
```

```r
MSE(test$scores_test_lasso,test$percent_female)
```

```
## [1] 2618.369
```

```r
#plot correlation predicted vs. true for train & test
library(ggplot2)
resids_train_cor <- train$percent_female - train$scores_train_cor
resids_test_cor <- test$percent_female - test$scores_test_cor

preds_df_cor <- data.frame(preds = c(train$scores_train_cor,test$scores_test_cor),
                           resids = c(resids_train_cor,resids_test_cor),
                           type = c(rep("train",nrow(train)),rep("test",nrow(test))))

ggplot(preds_df_cor, aes(x = preds, y = resids)) + geom_point(alpha = 1/10) +
  facet_wrap(~type) + labs(x = "True Gap Ratio", y = "Predicted Gap Ratio") +
  labs(title="Correlation Model")
```




```
#plot lasso predicted vs. true for train & test
resids_train_lasso <- train$percent_female - train$scores_train_lasso
resids_test_lasso <- test$percent_female - test$scores_test_lasso

preds_df_lasso <- data.frame(preds = c(train$scores_train_lasso,test$scores_test_lasso),
                             resids = c(resids_train_lasso,resids_test_lasso),
                             type = c(rep("train",nrow(train)),rep("test",nrow(test))))

ggplot(preds_df_lasso, aes(x = preds, y = resids)) + geom_point(alpha = 1/10) +
  facet_wrap(~type) + labs(x = "True Gap Ratio", y = "Predicted Gap Ratio") +
  labs(title="Lasso Model")
```

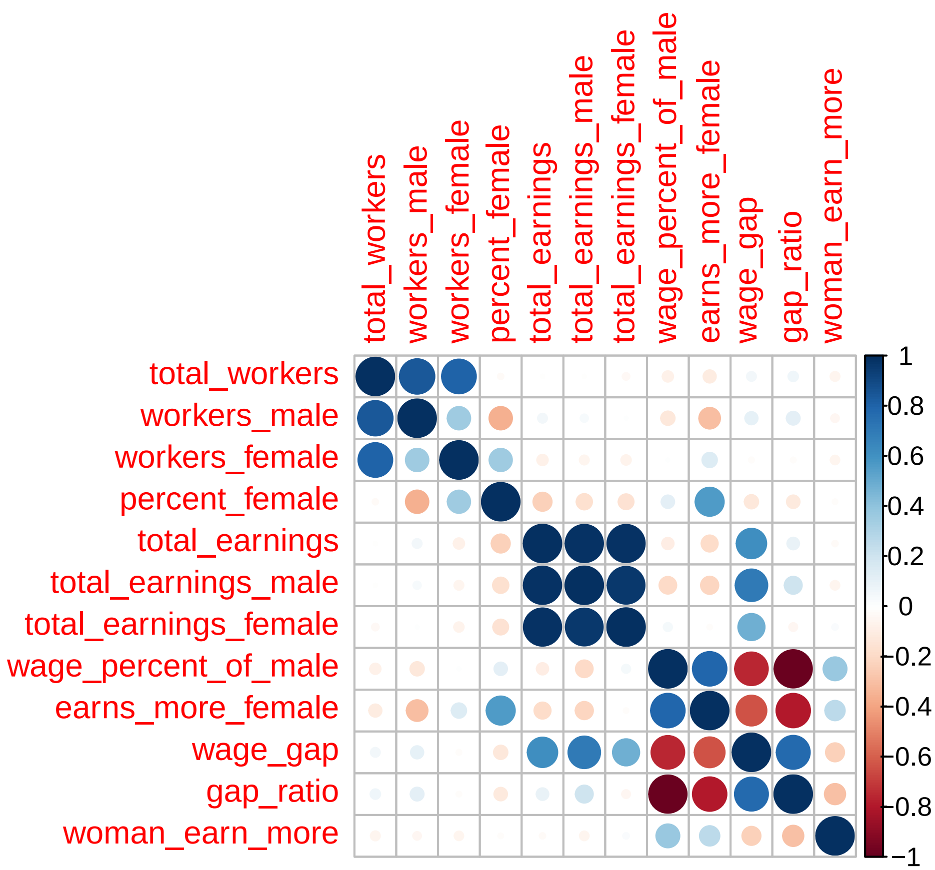

```
  #### Woman Earn More Classification ####
DF_percent <- as.data.frame(summaryBy(woman_earn_more ~ major_category, data = train))
DF_percent$woman_earn_more.mean <- percent(DF_percent$woman_earn_more.mean)
print(DF_percent)
```

```
##              major_category woman_earn_more.mean
## 1               Blue Collar                0.00%
## 2                  Business                6.16%
## 3         Community Service                4.85%
## 4              Construction                5.88%
## 5 Healthcare Practitioners                 1.41%
## 6           Sales and Office               0.00%
## 7                   Science                0.81%
## 8                   Service                2.01%
```

```
set.seed(2019)
trainSize <- 0.75
train_idx <- sample(1:nrow(clean_jg), size = floor(nrow(clean_jg) * trainSize))
train <- as.data.frame(clean_jg[train_idx,])
test <- as.data.frame(clean_jg[-train_idx,])

# Correlation
library(corrplot)
sapply(clean_jg, class)
```

```
##                 year       major_category        total_workers
##             "factor"             "factor"            "numeric"
##         workers_male       workers_female       percent_female
##            "numeric"            "numeric"            "numeric"
##       total_earnings   total_earnings_male total_earnings_female
##            "numeric"            "numeric"            "numeric"
##   wage_percent_of_male    earns_more_female             wage_gap
##            "numeric"            "numeric"            "numeric"
##            gap_ratio      woman_earn_more
##            "numeric"            "numeric"
```

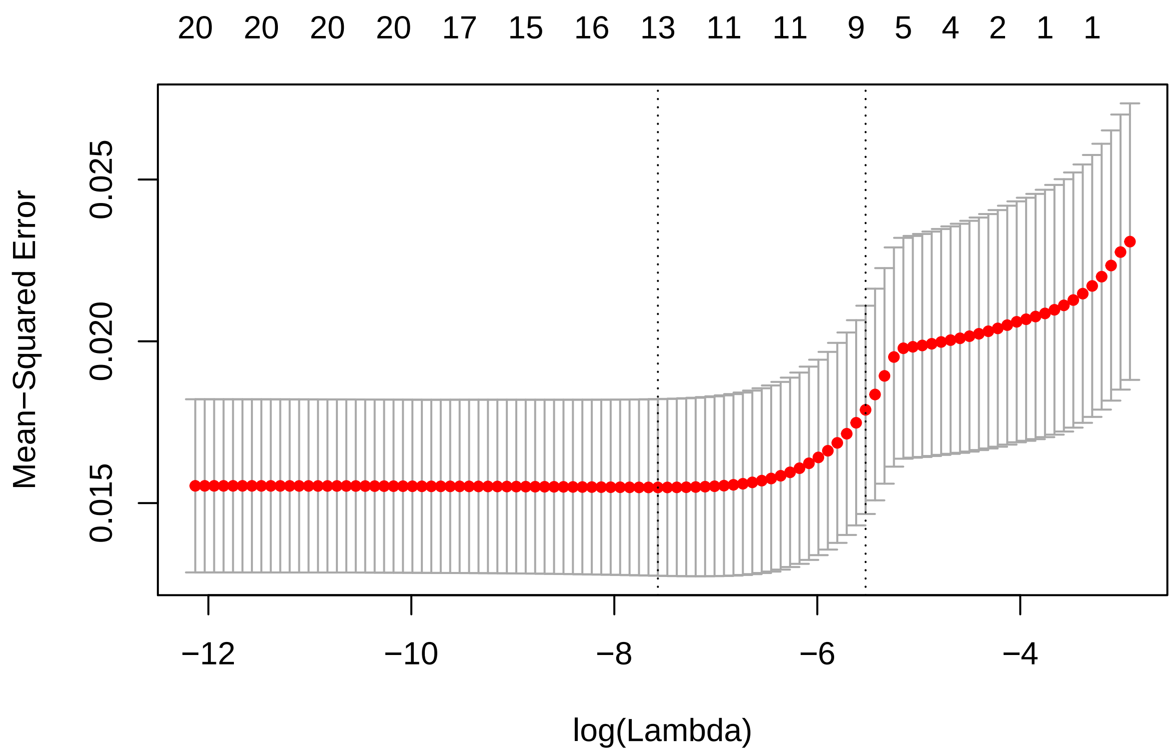
```r
cor_dataframe <- clean_jg[,c(-1,-2)]
cor <- cor(cor_dataframe)
corrplot(cor)
```




```r
x <- cor[,12] # woman_earn_more correlation
abs_x <- abs(x)
tail(sort(abs_x),8)
```

```
##       workers_female  total_earnings_male        total_workers
##           0.05285937           0.05429422           0.05760929
##             wage_gap    earns_more_female            gap_ratio
##           0.23210551           0.26737343           0.29124136
## wage_percent_of_male      woman_earn_more
##           0.37311863           1.00000000
```

```r
# top variables: wage_percent_of_male, gap_ratio, earns_more_female, wage_gap, total_workers,
# total_earnings_male, workers_female

    #### Choose Variables With Lasso ####
lasso_mod <- cv.glmnet(woman_earn_more ~ .,data = train, alpha = 1)
plot(lasso_mod)
```




```r
coefs <- data.frame(
  lasso_lambda_min = as.matrix(round(coef(lasso_mod, s = "lambda.min"),3)),
  lasso_lambda_1se = as.matrix(round(coef(lasso_mod, s = "lambda.1se"),3)))

colnames(coefs) <- c("Lasso Min","Lasso 1se")
print(coefs)
```

```
##                                          Lasso Min Lasso 1se
## (Intercept)                                 -4.429    -1.556
## year2013                                     0.000     0.000
## year2014                                     0.000     0.000
## year2015                                     0.000     0.000
## year2016                                     0.000     0.000
## major_categoryBlue Collar                    0.011     0.002
## major_categoryBusiness                       0.054     0.052
## major_categoryCommunity Service              0.027     0.019
## major_categoryConstruction                   0.000     0.000
## major_categoryHealthcare Practitioners       0.000     0.000
```

```
## major_categorySales and Office           -0.020    -0.003
## major_categoryScience                    -0.023    -0.017
## major_categoryService                    -0.015     0.000
## total_workers                             0.000     0.000
## workers_male                              0.000     0.000
## workers_female                            0.000     0.000
## percent_female                           -0.001     0.000
## total_earnings                            0.000     0.000
## total_earnings_male                       0.000     0.000
## total_earnings_female                     0.000     0.000
## wage_percent_of_male                      0.042     0.017
## earns_more_female                         0.485     0.000
## wage_gap                                  0.000     0.000
## gap_ratio                                 2.599     0.742
```

```
# which variables are selected:
# Lasso Min: major_category,percent_female, wage_percent_of_male,earns_more_female, gap_ratio

# more managable set of variables no need for lasso 1se
# Lasso 1se: major_category, wage_percent_of_male, gap_ratio

# lambda min values
lasso_mod$lambda.min
```

```
## [1] 0.0005153922
```

```
# lambda 1se values
lasso_mod$lambda.1se
```

```
## [1] 0.003990495
```

```
# MSE of lasso
indx <- which(lasso_mod$lambda == lasso_mod$lambda.min)
lasso_mod$cvm[indx]
```

```
## [1] 0.01548383
```

```
    #### ####
    # Lasso Min: major_category,percent_female, wage_percent_of_male,earns_more_female, gap_ratio

    #### Logistic Regression ####
# predicting why woman_earn_more in some work places
logit_fit_cor <- glm(woman_earn_more ~ wage_percent_of_male+gap_ratio+earns_more_female+wage_gap+
                       total_workers+total_earnings_male+workers_female, data = train, family = binomial
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logit_fit_cor)
```

```
##
```

```
## Call:
## glm(formula = woman_earn_more ~ wage_percent_of_male + gap_ratio +
##     earns_more_female + wage_gap + total_workers + total_earnings_male +
##     workers_female, family = binomial, data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
##
## Coefficients:
##                       Estimate Std. Error    z value Pr(>|z|)
## (Intercept)          -5.072e+16  1.443e+08 -351379191   <2e-16 ***
## wage_percent_of_male  3.135e+14  1.434e+06  218548381   <2e-16 ***
## gap_ratio             2.187e+16  9.774e+07  223769533   <2e-16 ***
## earns_more_female     1.976e+16  5.871e+07  336507334   <2e-16 ***
## wage_gap              3.571e+09  8.973e+02    3979385   <2e-16 ***
## total_workers        -1.763e+08  1.011e+01  -17448653   <2e-16 ***
## total_earnings_male  -4.665e+09  1.806e+02  -25826487   <2e-16 ***
## workers_female       -8.792e+08  1.781e+01  -49358418   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 208.27  on 930  degrees of freedom
## Residual deviance: 648.79  on 923  degrees of freedom
## AIC: 664.79
##
## Number of Fisher Scoring iterations: 17
```

```r
exp(logit_fit_cor$coefficients)
```

```
##          (Intercept) wage_percent_of_male            gap_ratio
##                    0                  Inf                  Inf
##    earns_more_female             wage_gap        total_workers
##                  Inf                  Inf                    0
##  total_earnings_male       workers_female
##                    0                    0
```

```r
logit_fit_lasso <- glm(woman_earn_more ~ as.factor(major_category)+percent_female+wage_percent_of_male+e
                       data = train, family = binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(logit_fit_lasso)
```

```
##
## Call:
## glm(formula = woman_earn_more ~ as.factor(major_category) + percent_female +
##     wage_percent_of_male + earns_more_female + gap_ratio, family = binomial,
##     data = train)
##
```

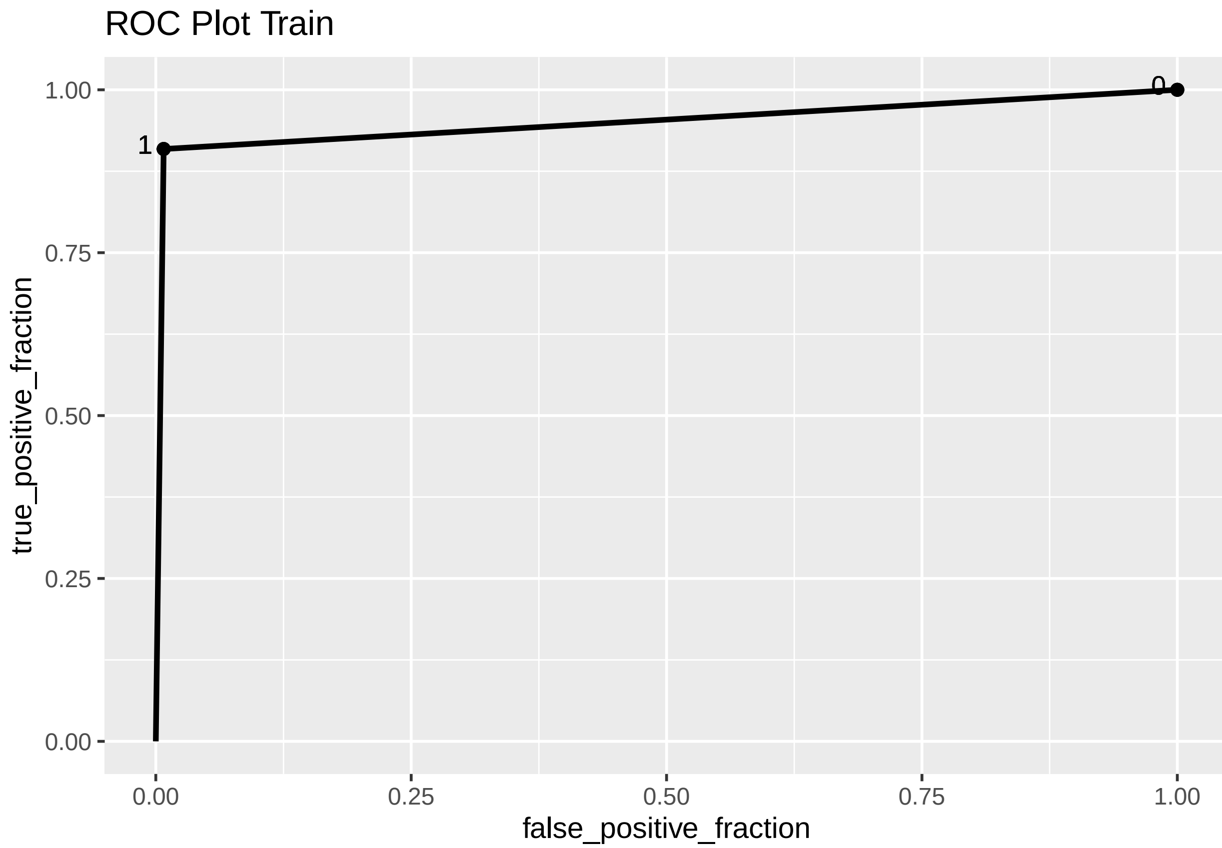
```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
##
## Coefficients:
##                                                   Estimate Std. Error
## (Intercept)                                     -2.967e+16  1.476e+08
## as.factor(major_category)Business                2.152e+14  8.651e+06
## as.factor(major_category)Community Service       3.553e+14  9.748e+06
## as.factor(major_category)Construction           -3.592e+14  1.191e+07
## as.factor(major_category)Healthcare Practitioners  9.024e+14  1.089e+07
## as.factor(major_category)Sales and Office       -3.590e+14  8.768e+06
## as.factor(major_category)Science                 5.538e+12  9.062e+06
## as.factor(major_category)Service                -4.695e+13  8.770e+06
## percent_female                                  -1.909e+13  1.689e+05
## wage_percent_of_male                             4.695e+14  1.456e+06
## earns_more_female                               -1.683e+16  8.502e+07
## gap_ratio                                        1.830e+16  9.404e+07
##                                                    z value Pr(>|z|)
## (Intercept)                                     -201044774   <2e-16 ***
## as.factor(major_category)Business                 24879744   <2e-16 ***
## as.factor(major_category)Community Service        36446124   <2e-16 ***
## as.factor(major_category)Construction            -30168934   <2e-16 ***
## as.factor(major_category)Healthcare Practitioners   82825552   <2e-16 ***
## as.factor(major_category)Sales and Office        -40944228   <2e-16 ***
## as.factor(major_category)Science                    611118   <2e-16 ***
## as.factor(major_category)Service                  -5353185   <2e-16 ***
## percent_female                                  -113020358   <2e-16 ***
## wage_percent_of_male                             322386743   <2e-16 ***
## earns_more_female                               -197977712   <2e-16 ***
## gap_ratio                                        194631192   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 208.27  on 930  degrees of freedom
## Residual deviance: 792.96  on 919  degrees of freedom
## AIC: 816.96
##
## Number of Fisher Scoring iterations: 22
```

```r
exp(logit_fit_lasso$coefficients)
```

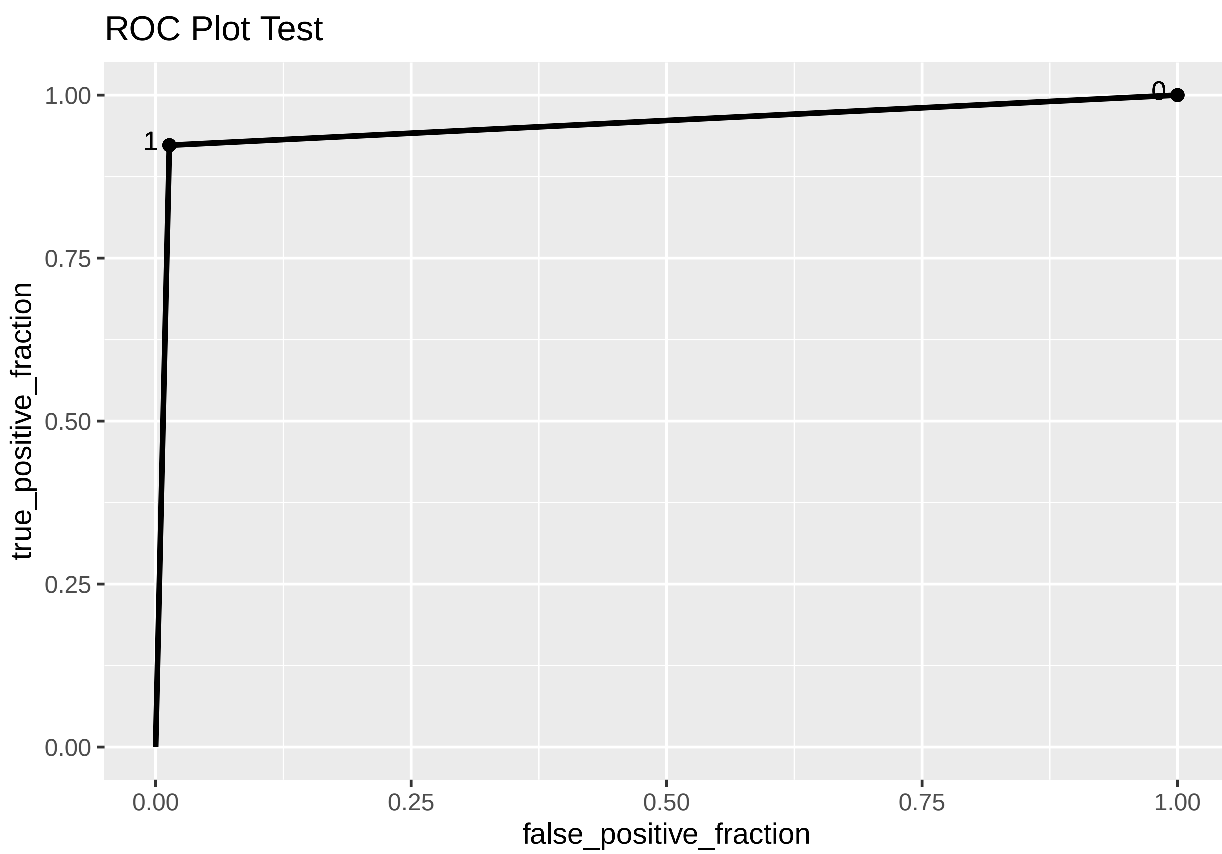
```
##                                       (Intercept)
##                                                 0
##                 as.factor(major_category)Business
##                                               Inf
##        as.factor(major_category)Community Service
##                                               Inf
##            as.factor(major_category)Construction
##                                                 0
## as.factor(major_category)Healthcare Practitioners
##                                               Inf
```

```
##         as.factor(major_category)Sales and Office
##                                                 0
##                  as.factor(major_category)Science
##                                               Inf
##                  as.factor(major_category)Service
##                                                 0
##                                    percent_female
##                                                 0
##                               wage_percent_of_male
##                                               Inf
##                                 earns_more_female
##                                                 0
##                                         gap_ratio
##                                               Inf
```

```
# predict probability for the train and test
# correlation
preds_train_cor <- data.frame(scores = predict(logit_fit_cor, newdata = train, type = "response"),train)
preds_test_cor <- data.frame(scores = predict(logit_fit_cor, newdata = test, type = "response"),test)
# lasso
preds_train_lasso <- data.frame(scores = predict(logit_fit_lasso, newdata = train, type = "response"),tr
preds_test_lasso <- data.frame(scores = predict(logit_fit_lasso, newdata = test, type = "response"),tes


# ROC Curve
library(plotROC)
# Correlation
# train
ROC_train <- ggplot(preds_train_cor, aes(m = scores, d = woman_earn_more)) +
  geom_roc(labelsize = 3.5, cutoffs.at = c(.99,.9,.7,.5,.3,.1,0)) +
  labs(title = "ROC Plot Train")
plot(ROC_train)
```




```r
# test
ROC_test <- ggplot(preds_test_cor, aes(m = scores, d = woman_earn_more)) +
  geom_roc(labelsize = 3.5, cutoffs.at = c(.99,.9,.7,.5,.3,.1,0)) +
  labs(title = "ROC Plot Test")
plot(ROC_test)
```



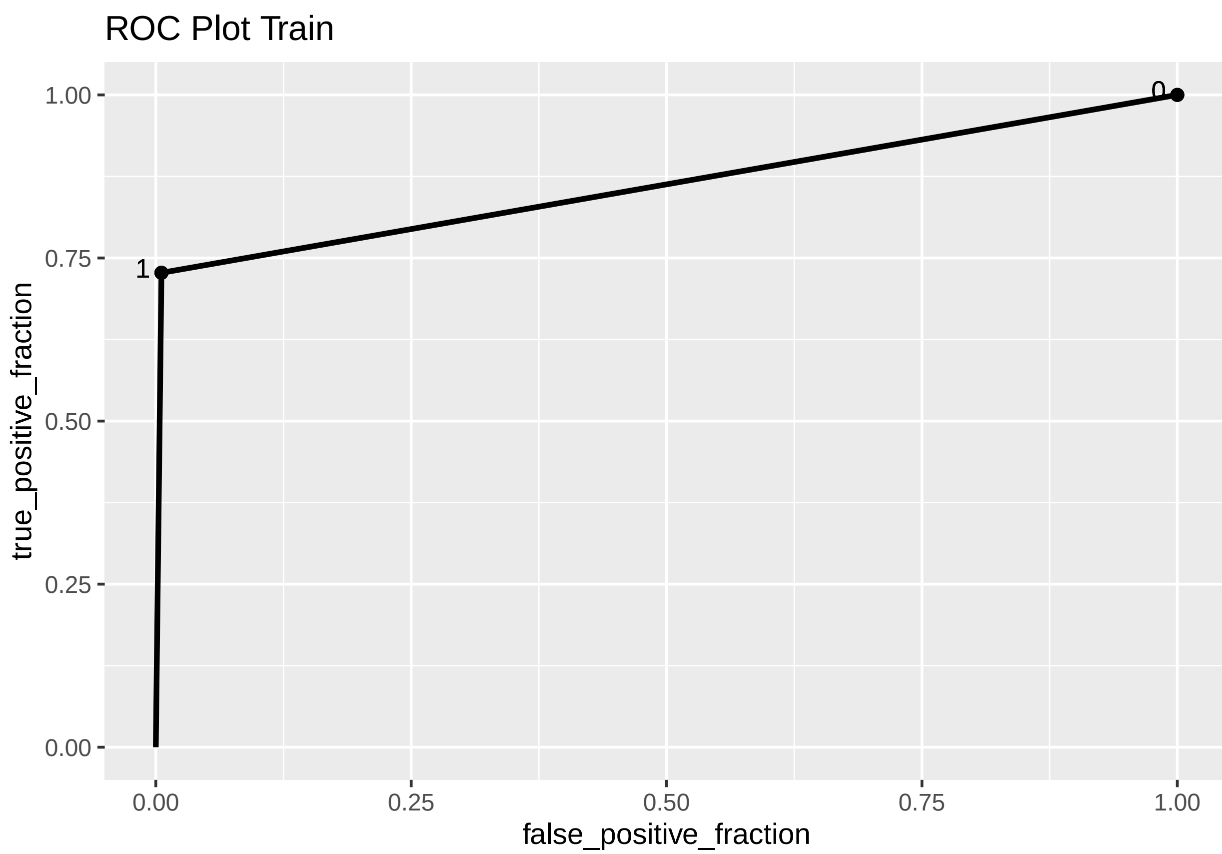

```r
calc_auc(ROC_train)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9506951
```

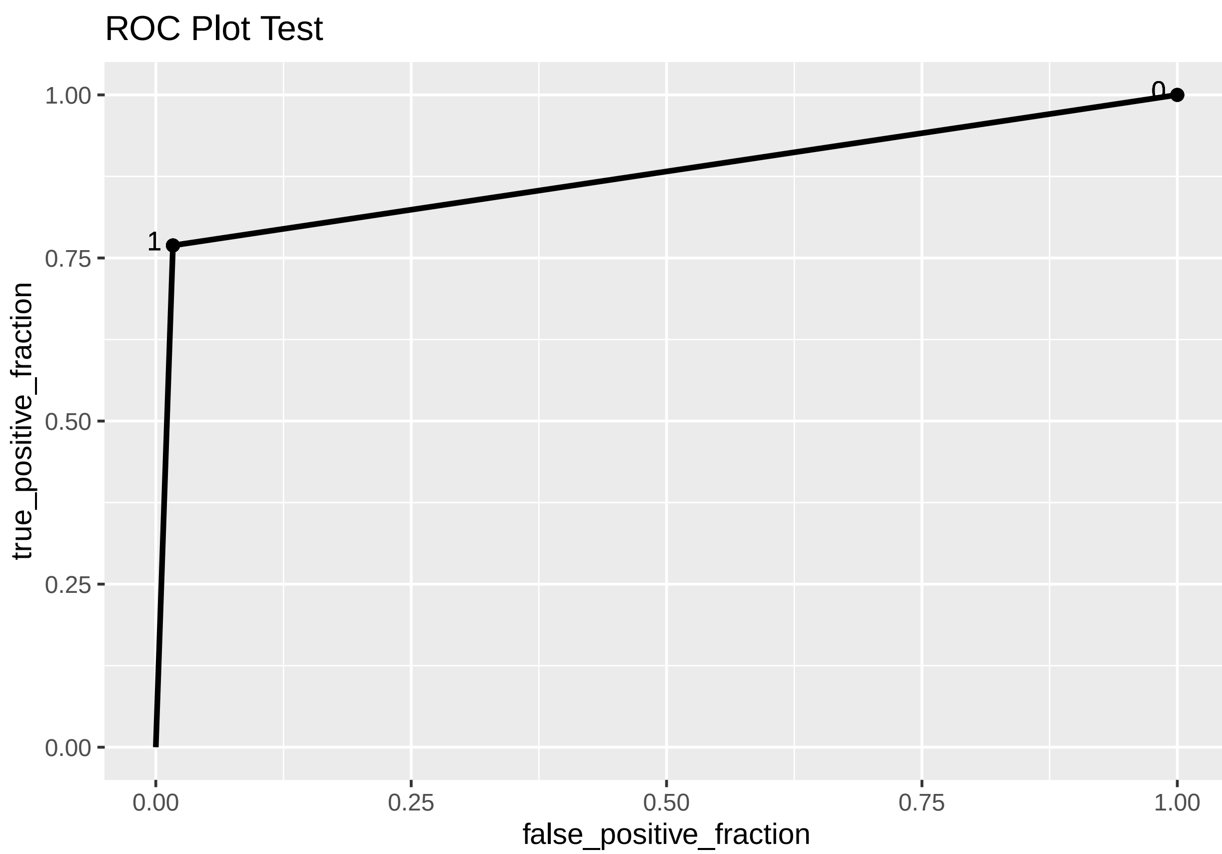
```r
calc_auc(ROC_test)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9548271
```

```r
# Lasso
# train
ROC_train <- ggplot(preds_train_lasso, aes(m = scores, d = woman_earn_more)) +
  geom_roc(labelsize = 3.5, cutoffs.at = c(.99,.9,.7,.5,.3,.1,0)) +
  labs(title = "ROC Plot Train")
plot(ROC_train)
```




```r
# test
ROC_test <- ggplot(preds_test_lasso, aes(m = scores, d = woman_earn_more)) +
  geom_roc(labelsize = 3.5, cutoffs.at = c(.99,.9,.7,.5,.3,.1,0)) +
  labs(title = "ROC Plot Test")
plot(ROC_test)
```




```r
calc_auc(ROC_train)
```

```
##   PANEL group       AUC
## 1     1    -1 0.8608861
```

```r
calc_auc(ROC_test)
```

```
##   PANEL group       AUC
## 1     1    -1 0.8762261
```

```r
# demonstrates that our ROC curve is great at identifying woman_earn_more with accuracy of 98.4%
# the thresholds do not affect our results
# false positive rate is lower on train cs test
# we know that the variables in this model greatly affect gow much woman earn

    #### Random Forest Tree ####
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(2019)
random_forest_m3 <- randomForest(woman_earn_more ~ .,data = train, mtry = 7,
                                 ntree = 500, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
# predict (train + test)
preds_train_bg <- predict(random_forest_m3)
preds_test_bg <- predict(random_forest_m3, newdata = test)
# MSE
MSE(preds_train_bg, train$woman_earn_more)
```

```
## [1] 0.0002859051
```

```
MSE(preds_test_bg, test$woman_earn_more)
```

```
## [1] 6.610932e-06
```

```
    #### Evenning Out Data ####
library("ROSE")
```

```
## Loaded ROSE 0.0-3
```

```
table(train$woman_earn_more)
```

```
##
##   0   1
## 909  22
```

```
data_balanced_over <- ovun.sample(woman_earn_more ~ ., data = train, method = "over",N = 1818)$data
table(data_balanced_over$woman_earn_more)
```

```
##
##   0   1
## 909 909
```

```
data_balanced_both <- ovun.sample(woman_earn_more ~ ., data = train, method = "both", p=0.5,
table(data_balanced_both$woman_earn_more)
```

```
##
##   0   1
## 520 480
```

```r
# two new datasets: data_balanced_over and data_balanced_both
set.seed(2019)
random_forest_m3 <- randomForest(woman_earn_more ~ .,data = data_balanced_over, mtry = 7,
                                 ntree = 500, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```r
# predict (train + test)
preds_train_bg <- predict(random_forest_m3)
preds_test_bg <- predict(random_forest_m3, newdata = test)
# MSE
MSE(preds_train_bg, train$woman_earn_more)
```

```
## Warning in t - p: longer object length is not a multiple of shorter object
## length
```

```
## [1] 0.5001351
```

```r
MSE(preds_test_bg, test$woman_earn_more)
```

```
## [1] 9.980707e-06
```

```r
set.seed(2019)
random_forest_m3 <- randomForest(woman_earn_more ~ .,data = data_balanced_both, mtry = 7,
                                 ntree = 500, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```r
# predict (train + test)
preds_train_bg <- predict(random_forest_m3)
preds_test_bg <- predict(random_forest_m3, newdata = test)
# MSE
MSE(preds_train_bg, train$woman_earn_more)
```

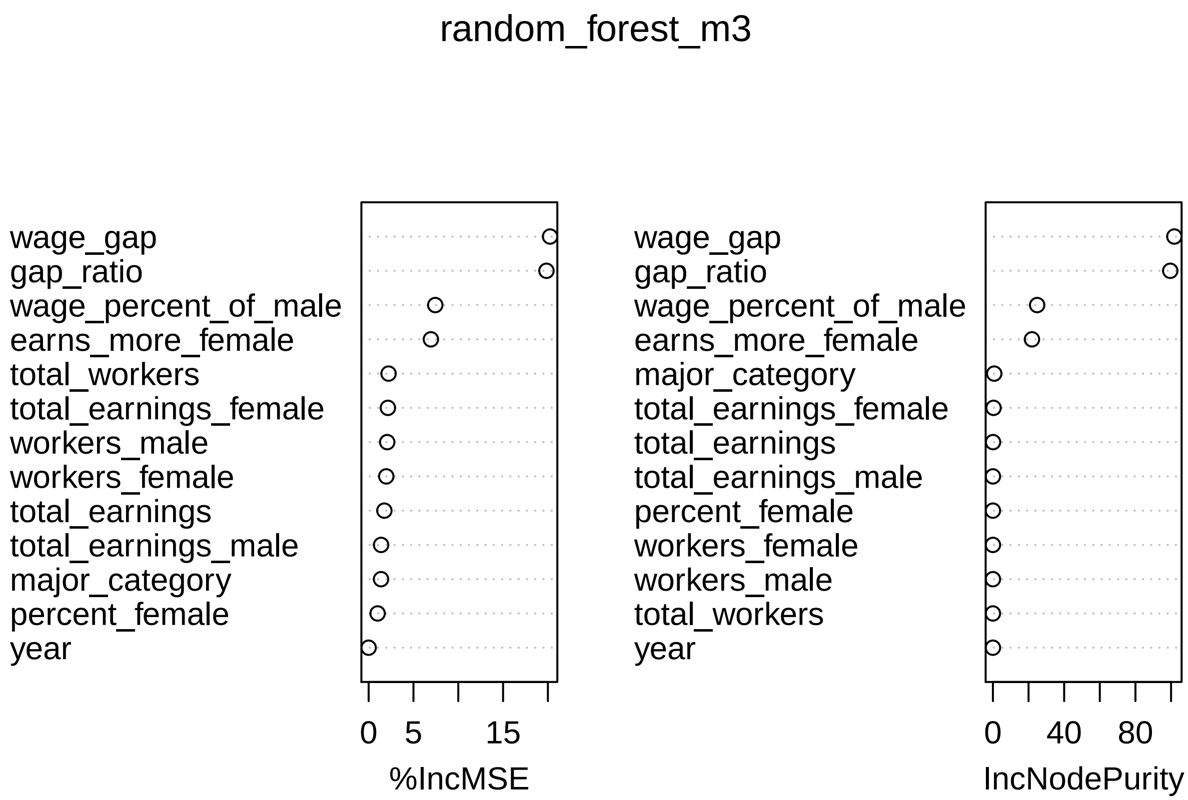
```
## Warning in t - p: longer object length is not a multiple of shorter object
## length
```

```
## [1] 0.4850884
```

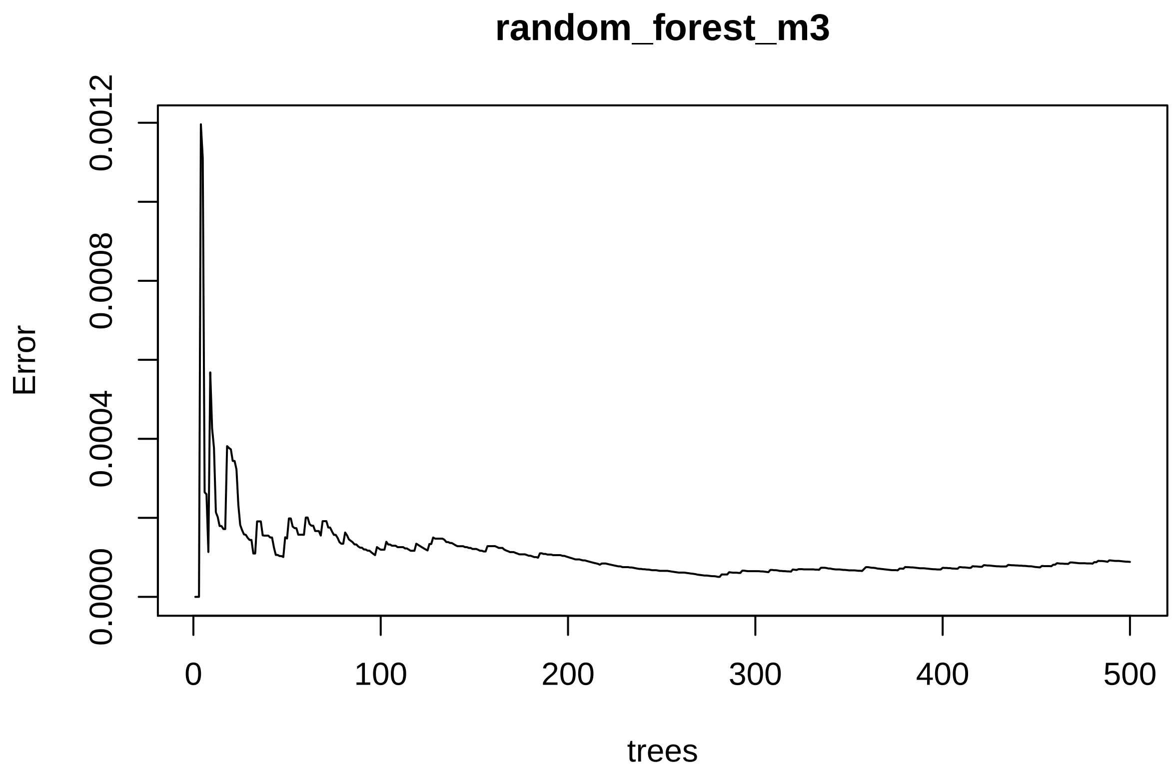
```r
MSE(preds_test_bg, test$woman_earn_more)
```

```
## [1] 6.965916e-05
```

```r
varImpPlot(random_forest_m3)
```




```r
plot(random_forest_m3)
```




```r
#### INSIGHTS ####
# percentage of difference in wage gap by majpr
DF_percent_gap <- as.data.frame(summaryBy(gap_ratio ~ major_category, data = train))
DF_percent_gap$gap_ratio.mean <- percent(DF_percent_gap$gap_ratio.mean)
print(DF_percent_gap)
```

```
##            major_category gap_ratio.mean
## 1             Blue Collar          28.6%
## 2                Business          25.4%
## 3       Community Service          18.6%
## 4            Construction          18.6%
## 5 Healthcare Practitioners          16.7%
## 6          Sales and Office          20.7%
## 7                 Science          16.5%
## 8                 Service          17.2%
```