Assignment: ASSIGNMENT 7

Name: Shekhar , Manish

Date: 2021-05-05

Set the working directory to the root of your DSC 520 directory

Load the `data/r4ds/heights.csv` to

```
heights_df <- read.csv("./heights.csv")
str(heights_df)
```

```
## 'data.frame':    1192 obs. of  6 variables:
##  $ earn  : num  50000 60000 30000 50000 51000 9000 29000 32000 2000 27000 ...
##  $ height: num  74.4 65.5 63.6 63.1 63.4 ...
##  $ sex   : chr  "male" "female" "female" "female" ...
##  $ ed    : int  16 16 16 16 17 15 12 17 15 12 ...
##  $ age   : int  45 58 29 91 39 26 49 46 21 26 ...
##  $ race  : chr  "white" "white" "white" "other" ...
```

```
# sex and race are two categorical variables in the data
# to be able to use them we will have to change them to factor
# and assign numeric value to each category
# checking unique categories in each categorical variable
unique(heights_df$sex)
```

```
## [1] "male"   "female"
```

```
unique(heights_df$race)
```

```
## [1] "white"    "other"    "hispanic" "black"
```

```
# changing categorical variables to factor
heights_df$sex <- factor(heights_df$sex,
                         levels = c('male','female'),
                         labels = c(1,2))
heights_df$race <- factor(heights_df$race,
                         levels = c('white','other','hispanic','black'),
                         labels = c(0,1,2,3))
# check data structure again, categorical variables should be factors now
# with numerical values for each categories
str(heights_df)
```

```
## 'data.frame':    1192 obs. of  6 variables:
##  $ earn  : num  50000 60000 30000 50000 51000 9000 29000 32000 2000 27000 ...
##  $ height: num  74.4 65.5 63.6 63.1 63.4 ...
##  $ sex   : Factor w/ 2 levels "1","2": 1 2 2 2 2 2 2 1 1 1 ...
##  $ ed    : int  16 16 16 16 17 15 12 17 15 12 ...
##  $ age   : int  45 58 29 91 39 26 49 46 21 26 ...
##  $ race  : Factor w/ 4 levels "0","1","2","3": 1 1 1 2 1 1 1 1 3 1 ...
```

Fit a linear model

```r
# lm function takes care of scaling the numeric variables
# Also, lm function takes care of dummy variable trap
# meaning it creates n-1 variables for each categorical variable, where n = distinct number of categori
# In the example below lm function breaks sex into sex1 and sex2 and uses only one of them to create th
# It also breaks race into race1, race2, race3, and race4 and uses only three of them to create the mod
earn_lm <-  lm(earn ~ height + sex + ed + age + race, data=heights_df)
```

```r
# View the summary of your model
summary(earn_lm)
```

```
##
## Call:
## lm(formula = earn ~ height + sex + ed + age + race, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39423  -9827  -2208   6157 158723
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28720.4    13360.4  -2.150   0.0318 *
## height         202.5      185.6   1.091   0.2754
## sex2        -10325.6     1424.5  -7.249 7.57e-13 ***
## ed            2768.4      209.9  13.190  < 2e-16 ***
## age            178.3       32.2   5.537 3.78e-08 ***
## race1        -2061.4     3515.5  -0.586   0.5577
## race2        -3846.7     2212.0  -1.739   0.0823 .
## race3        -2432.5     1723.9  -1.411   0.1585
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1184 degrees of freedom
## Multiple R-squared:  0.2199, Adjusted R-squared:  0.2153
## F-statistic: 47.68 on 7 and 1184 DF,  p-value: < 2.2e-16
```

```r
# Applying backward elimination predictor selection and model building technique
# Execution 1:  with all the variables
# -------------------------------------
# Adjusted R^2 = 0.2153
# F-statistic = 47.68 and p-value = 2.2e-16
# identify least relevant variable by picking one by highest p-value and removing it from model
# Even though race1 is showing highest p-value, because race2 is little significant we can keep race an
# Recreate model without height variable and check model stats
earn_lm_2 <-  lm(earn ~ sex + ed + age + race, data=heights_df)
# check stats
summary(earn_lm_2)
```

```
##
## Call:
## lm(formula = earn ~ sex + ed + age + race, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -38935  -9913  -2150   6184 158499
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -14617.27    3384.28  -4.319 1.70e-05 ***
## sex2        -11417.48    1013.88 -11.261  < 2e-16 ***
## ed            2788.94     209.04  13.341  < 2e-16 ***
## age            174.04      31.97   5.445 6.31e-08 ***
## race1        -2459.26    3496.87  -0.703   0.4820
## race2        -4089.08    2201.03  -1.858   0.0634 .
## race3        -2486.37    1723.30  -1.443   0.1493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1185 degrees of freedom
## Multiple R-squared:  0.2191, Adjusted R-squared:  0.2152
## F-statistic: 55.42 on 6 and 1185 DF,  p-value: < 2.2e-16
```

```r
# Execution 2:  without height variables
# -------------------------------------
# Adjusted R^2 = 0.2152
# F-statistic = 55.42 and p-value = 2.2e-16
# We can see that adjusted R^2 has not changed much while F-statistic has improved keeping p-value same
# We can also see that relevance on intercept has improved from one start to three starts
# If we be strict with the rules we can try another run and compare stats without race variable.
# race2 is little significant but p-value is still over 0.05 critical value
# Recreate model without height and race variable and check model stats
earn_lm_3 <-  lm(earn ~ sex + ed + age, data=heights_df)
# check stats
summary(earn_lm_3)
```

```
##
## Call:
## lm(formula = earn ~ sex + ed + age, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -38461  -9836  -2406   6172 158926
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15679.21    3350.28  -4.680 3.20e-06 ***
## sex2        -11429.88    1014.64 -11.265  < 2e-16 ***
## ed            2814.53     208.64  13.490  < 2e-16 ***
## age            179.16      31.87   5.621 2.36e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17270 on 1188 degrees of freedom
## Multiple R-squared:  0.2155, Adjusted R-squared:  0.2135
## F-statistic: 108.8 on 3 and 1188 DF,  p-value: < 2.2e-16
```

```r
# Execution 2:  without height variables
# -------------------------------------
# Adjusted R^2 = 0.2135
# F-statistic = 108.8 and p-value = 2.2e-16
# As we compare stats, we can see that adjusted R^2 went down a little bit but we can see significant i
```

```r
# Clearly this is huge improvement in the model
# Also, all variables are now highly significant in predicting the earn (predicted variable).
```

```r
# create a dummy data frame as test data to predict
test_predict_df <- data.frame(ed = c(14,13,10,11),
                              race = factor(c(2,2,3,1)),
                              height = c(45.5,40.4,55,67.1),
                              age = c(42,40,45,67),
                              sex = factor(c(1,1,2,2)))
# predicting earn using multiple regression model earn_lm
predicted_df <- data.frame(
  earn = predict(earn_lm, newdata = test_predict_df),
  ed=test_predict_df$ed, race=test_predict_df$race, height=test_predict_df$height,
  age=test_predict_df$age, sex=test_predict_df$sex
  )
# print predicted data frame
predicted_df
```

```
##        earn ed race height age sex
## 1 22891.269 14    2   45.5  42   1
## 2 18733.680 13    2   40.4  40   1
## 3  5364.917 10    3   55.0  45   2
## 4 14876.922 11    1   67.1  67   2
```

```r
## Compute deviation (i.e. residuals)
mean_earn <- mean(heights_df$earn)
mean_earn
```

```
## [1] 23154.77
```

```r
## Corrected Sum of Squares Total
sst <- sum((heights_df$earn - mean_earn)^2)
sst
```

```
## [1] 451591883937
```

```r
## Corrected Sum of Squares for Model
## To be able to show the same model evaluation stats will let model predict using
## training data -> heights_df
## recreating predicted_df by predicting on heights_df
predicted_df <- data.frame(
  earn = predict(earn_lm, newdata = heights_df),
  ed=heights_df$ed, race=heights_df$race, height=heights_df$height,
  age=heights_df$age, sex=heights_df$sex
  )
ssm <- sum((predicted_df$earn - mean_earn)^2)
ssm
```

```
## [1] 99302918657
```

```r
## Residuals
residuals <- (heights_df$earn - predicted_df$earn)
## Sum of Squares for Error
sse <- sum(residuals^2)
sse
```

```
## [1] 3.52289e+11
```

```r
## R Squared
r_squared <- ssm/sst
r_squared
```

```
## [1] 0.2198953
```

```r
## Number of observations
n <- nrow(heights_df)
n
```

```
## [1] 1192
```

```r
## Number of regression parameters
p <- 8
p
```

```
## [1] 8
```

```r
## Corrected Degrees of Freedom for Model
dfm <- p-1
dfm
```

```
## [1] 7
```

```r
## Degrees of Freedom for Error
dfe <- n-p
dfe
```

```
## [1] 1184
```

```r
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n-1
dft
```

```
## [1] 1191
```

```r
## Mean of Squares for Model:   MSM = SSM / DFM
msm <- ssm/dfm
msm
```

```
## [1] 14186131237
```

```r
## Mean of Squares for Error:   MSE = SSE / DFE
mse <- sse/dfe
mse
```

```
## [1] 297541356
```

```r
## Mean of Squares Total:   MST = SST / DFT
mst <- sst/dft
mst
```

```
## [1] 379170348
```

```r
## F Statistic
f_score <- msm/mse
f_score
```

```
## [1] 47.67785
```

```r
## Adjusted R Squared R2 = 1 - (1 - R2)(n - 1) / (n - p)
adjusted_r_squared <- 1-(1-r_squared)*(n-1)/(n-p)
adjusted_r_squared
```

```
## [1] 0.2152832
```