assignment_06_BrownLincoln.R

x

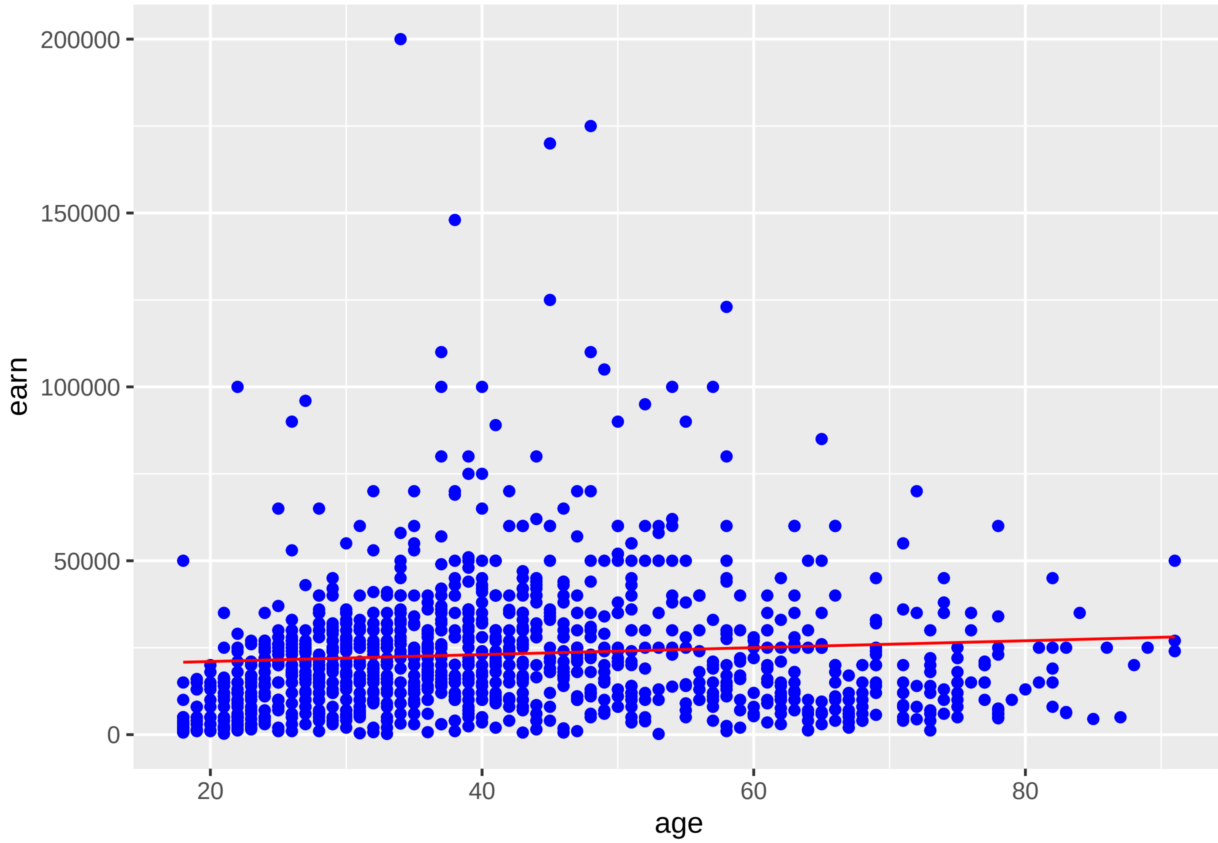
2021-08-01

```r
# Assignment: ASSIGNMENT 6
# Name: Brown, Lincoln
# Date: 2021-07-28

## Set the working directory to the root of your DSC 520 directory
setwd("/media/x/disk/School/DSC520/git/dsc520/")

## Load the `data/r4ds/heights.csv` to
heights_df <- read.csv("data/r4ds/heights.csv")

## Load the ggplot2 library
library(ggplot2)

## Fit a linear model using the `age` variable as the predictor and `earn` as the outcome
## lm([target variable] ~ [predictor variables], data = [data source])
age_lm <-  lm(formula = "earn ~ age", data = heights_df)

## View the summary of your model using `summary()`
summary(age_lm)
```

```
##
## Call:
## lm(formula = "earn ~ age", data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25098 -12622  -3667   6883 177579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19041.53    1571.26  12.119  < 2e-16 ***
## age            99.41      35.46   2.804  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19420 on 1190 degrees of freedom
## Multiple R-squared:  0.006561,   Adjusted R-squared:  0.005727
## F-statistic:  7.86 on 1 and 1190 DF,  p-value: 0.005137
```

```r
## Creating predictions using `predict()`
age_predict_df <- data.frame(earn = predict(age_lm, heights_df), age=heights_df$age)

## Plot the predictions against the original data
```

```r
ggplot(data = heights_df, aes(y = earn, x = age)) +
  geom_point(color='blue') +
  geom_line(color='red',data = age_predict_df, aes(y=earn, x=age))
```




```r
mean_earn <- mean(heights_df$earn)
mean_earn
```

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

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

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

```r
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - age_predict_df$earn)^2)
ssm
```

```
## [1] 2963111900
```

```r
## Residuals
residuals <- heights_df$earn - age_predict_df$earn

## Sum of Squares for Error
sse <- sum(residuals^2)
sse
```

```
## [1] 448628772037
```

```r
## R Squared R^2 = SSM\SST
r_squared <- ssm/sst
r_squared
```

```
## [1] 0.006561482
```

```r
## Number of observations
n <- nrow(heights_df)
## Number of regression parameters
p <- 2
## Corrected Degrees of Freedom for Model (p-1)
dfm <- p-1
print(dfm)
```

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

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

```
## [1] 1190
```

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

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

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

```
## [1] 2963111900
```

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

```
## [1] 376998968
```

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

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

```r
## F Statistic F = MSM/MSE
f_score <- msm/mse
print(f_score)
```

```
## [1] 7.859735
```

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

```
## [1] 0.005726659
```

```r
## Calculate the p-value from the F distribution
p_value <- pf(f_score, dfm, dft, lower.tail=F)
```

```r
print(p_value)
```

```
## [1] 0.005136826
```