ASSIGNMENT 6

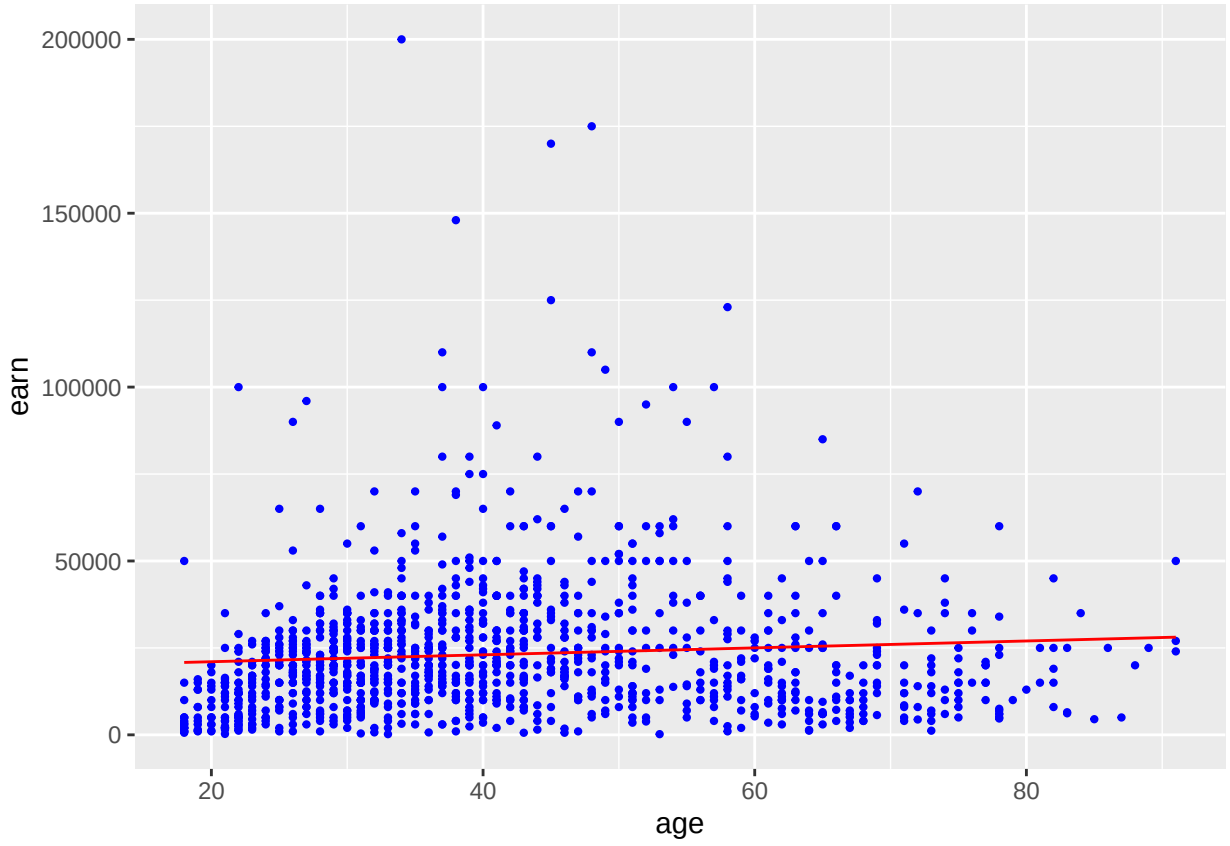
Simek, Kathryn

2010-10-11

```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/Users/katie/OneDrive/Documents/GitHub/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
age_lm <- lm(heights_df$earn ~ heights_df$age)

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

```
##
## Call:
## lm(formula = heights_df$earn ~ heights_df$age)
##
## 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 ***
## heights_df$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
```

```
## 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
ggplot(heights_df, aes(x = age, y = earn)) +
  geom_point(color='blue', size=.8) +
  geom_line(color='red',data = age_predict_df, aes(y=earn, x=age))
```




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

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

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

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

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

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

```
## Number of observations
n <- 1192

## Number of regression parameters
p <- 2

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

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

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

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

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

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

## F Statistic F = MSM/MSE
f_score <- msm/mse
```

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

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