assignment_06_ChattapadhyayKausik.R

kausik

2022-10-19

```r
# Assignment: ASSIGNMENT 6
# Name: Chattapadhyay, Kausik
# Date: 2022-10-20

## Set the working directory to the root of your DSC 520 directory
setwd("/Users/kausik/desktop/MS Data Science/DSC 520/dsc520-stats-r-assignments")

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

```
##       earn   height    sex ed age  race
## 1187 10000 70.05628 female 16  36 white
## 1188 19000 72.16573   male 12  29 white
## 1189 15000 61.13580 female 18  82 white
## 1190  8000 63.66416 female 12  33 white
## 1191 60000 71.92584   male 12  50 white
## 1192  6000 68.36849   male 12  27 white
```

```r
## 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(earn ~ age, data=heights_df)

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

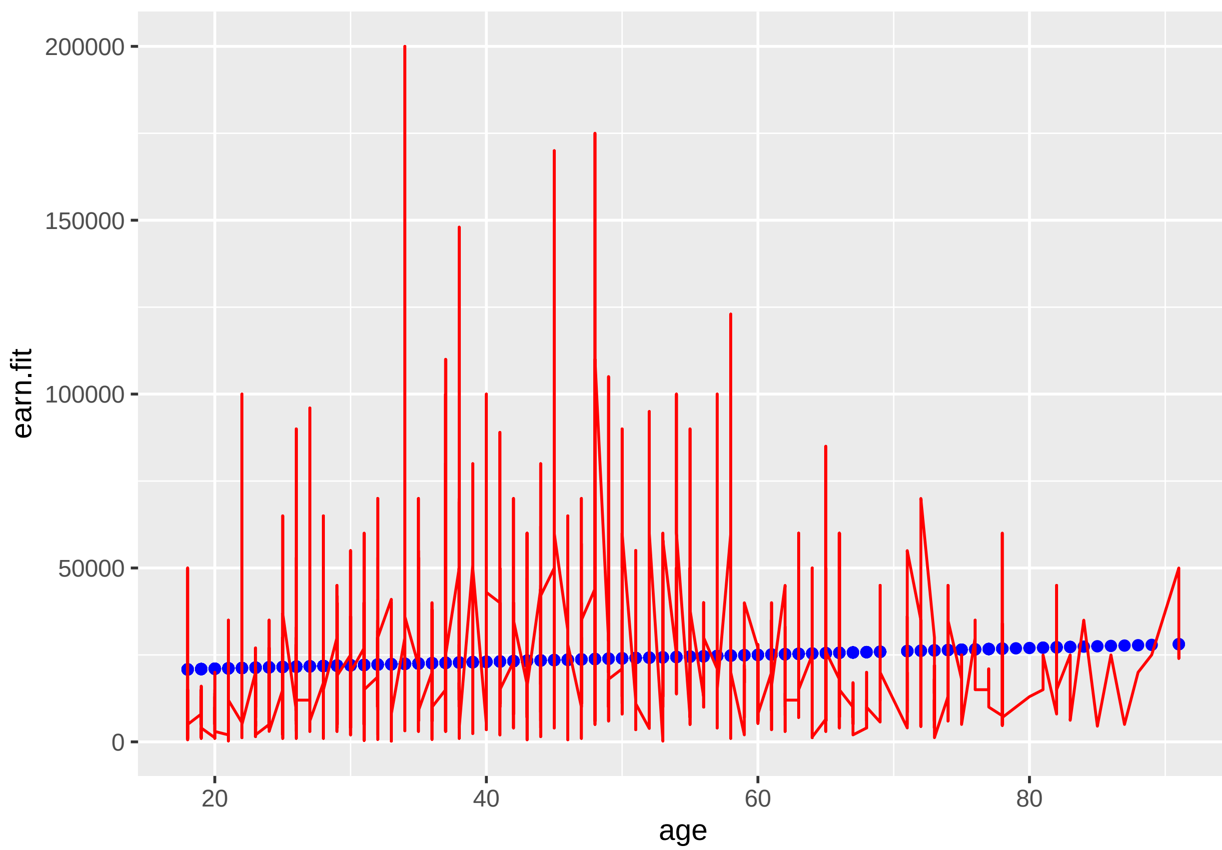
```
##
## Call:
## lm(formula = earn ~ age, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25098 -12622  -3667   6883 177579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19041.53    1571.26  12.119  < 2e-16 ***
## age            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
```

```
## Creating predictions using `predict()`
age_predict_df <- data.frame(earn = predict(age_lm, data.frame(age=heights_df$age),
                             interval = "prediction"),
                             earn.actual = heights_df$earn,
                             age=heights_df$age)
head(age_predict_df)
```

```
##   earn.fit  earn.lwr earn.upr earn.actual age
## 1 23514.79 -14596.33 61625.90       50000  45
## 2 24807.06 -13320.76 62934.88       60000  58
## 3 21924.29 -16195.72 60044.31       30000  29
## 4 28087.45 -10178.85 66353.76       50000  91
## 5 22918.35 -15192.29 61029.00       51000  39
## 6 21626.08 -16499.22 59751.37        9000  26
```

```
## Plot the predictions against the original data
ggplot(data = age_predict_df, aes(y = earn.fit, x = age)) +
  geom_point(color='blue') +
  geom_line(color='red',data = heights_df, aes(y=earn, x=age))
```



```r
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$earn.fit)^2)
## Residuals
residuals <- heights_df$earn - age_predict_df$earn.fit
## Sum of Squares for Error
sse <- sum(residuals^2)
## R Squared R^2 = SSM\SST
r_squared <- ssm/sst

## Number of observations
n <- sum(complete.cases(heights_df))
n
```

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

```r
## Number of regression parameters
p <- 2
## Corrected Degrees of Freedom for Model (p-1)
dfm <- p - 1
## Degrees of Freedom for Error (n-p)
dfe <- n - p
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n -1

## 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

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

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

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