Assignment_08_01_PothineniKalyan

PothineniKalyan

2023-05-07

Assignment_08_01

```r
heights_df <- read.csv("data/r4ds/heights.csv")
head(heights_df, n=5)
```

```
##    earn   height    sex ed age  race
## 1 50000 74.42444   male 16  45 white
## 2 60000 65.53754 female 16  58 white
## 3 30000 63.62920 female 16  29 white
## 4 50000 63.10856 female 16  91 other
## 5 51000 63.40248 female 17  39 white
```

```r
## Load the ggplot2 library
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

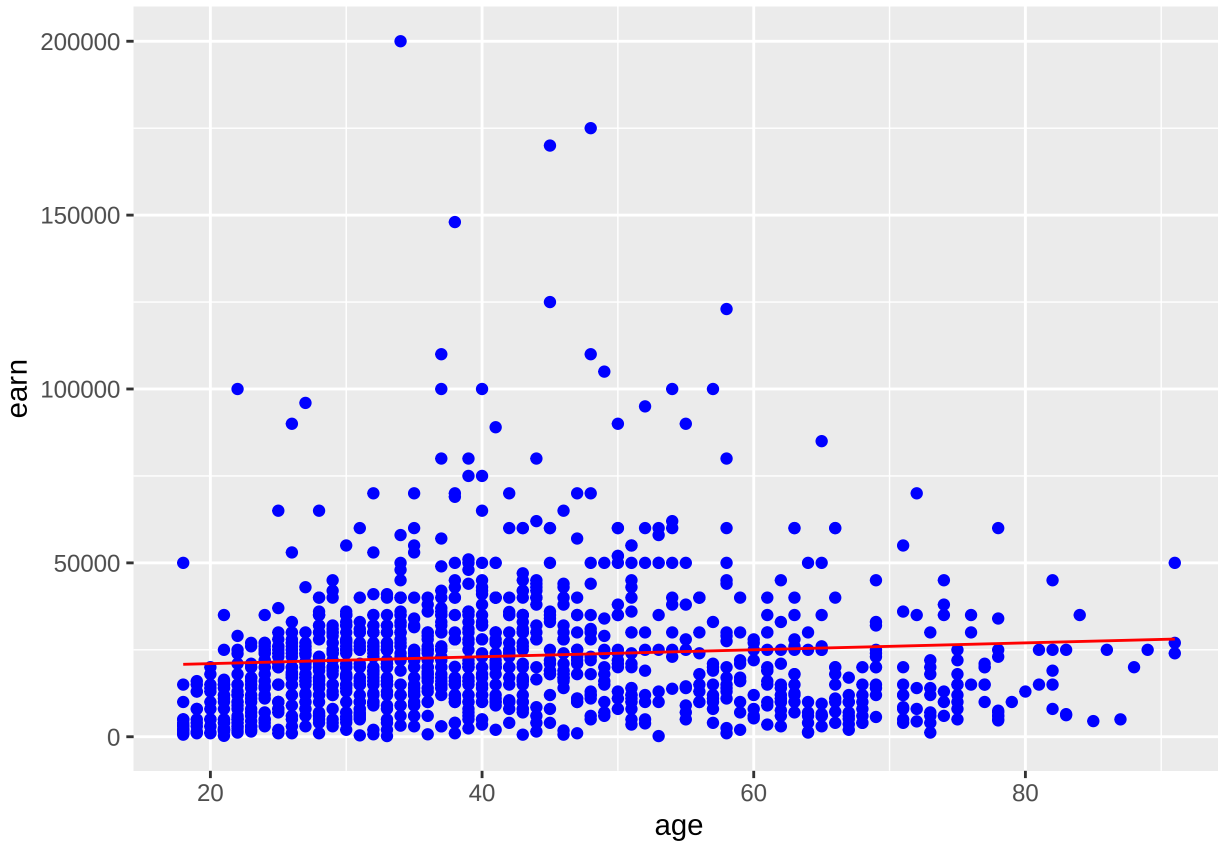
```r
## Fit a linear model using the `age` variable as the predictor and `earn` as the outcome
age_lm <-  lm(earn ~ age, data = heights_df)

## 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
```

```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
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$earn)
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$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 <- nrow(heights_df)
## Number of regression parameters
p <- 2
## Corrected Degrees of Freedom for Model (p-1)
```

```r
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
```