assignment_08-09_MunjewarSheetal-01

Sheetal M

2023-02-12

Install and Load required packages :

```
# Package names
# packages <- c("ggplot2","dplyr","tidyr","magrittr","tidyverse","purrr")
packages <- c("ggplot2","dplyr","magrittr","tidyverse","purrr","pander","pandoc")

# Install packages not yet installed
installed_packages <- packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(packages[!installed_packages])
}

# Packages loading
invisible(lapply(packages, library, character.only = TRUE))
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v purrr   1.0.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
## Set the working directory to the root of your DSC 520 directory
setwd("E:\\Data_Science_DSC510\\DSC520-Statistics\\dsc520")
```

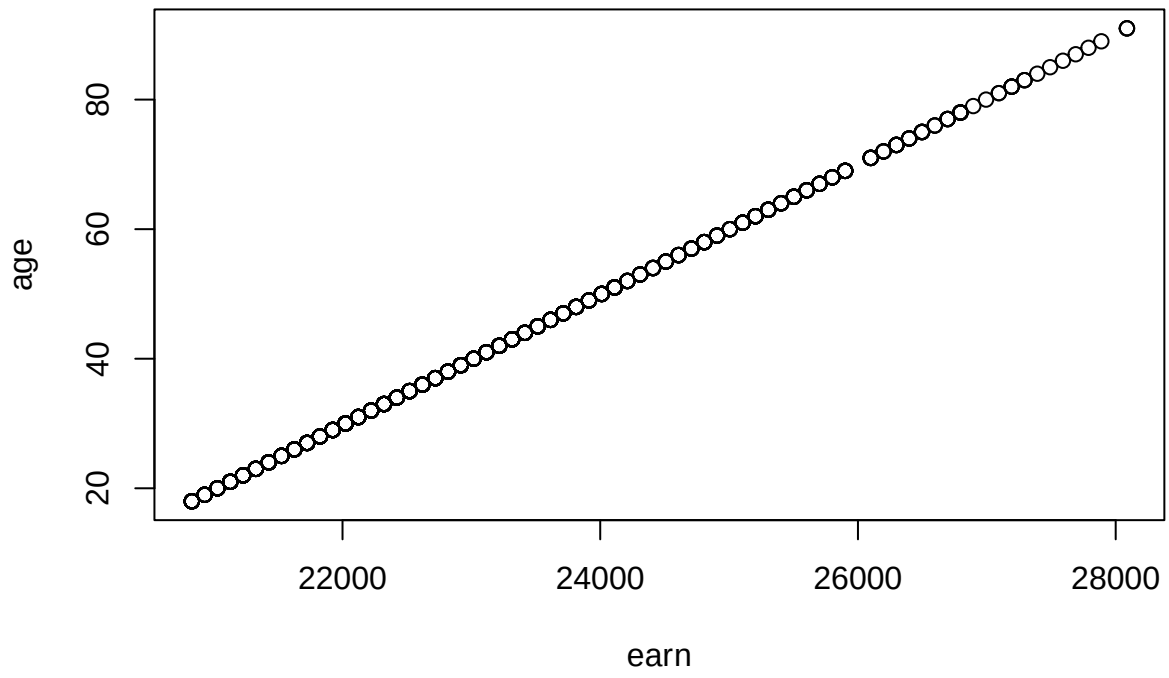
```r
## Load the `data/r4ds/heights.csv` to
heights_df <- read.csv("data/r4ds/heights.csv")
# nrow(heights_df)
## Load the ggplot2 library
library(ggplot2)

## 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, data = heights_df)

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

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

```r
# plot(age_lm)
```

Creating predictions using `predict()`

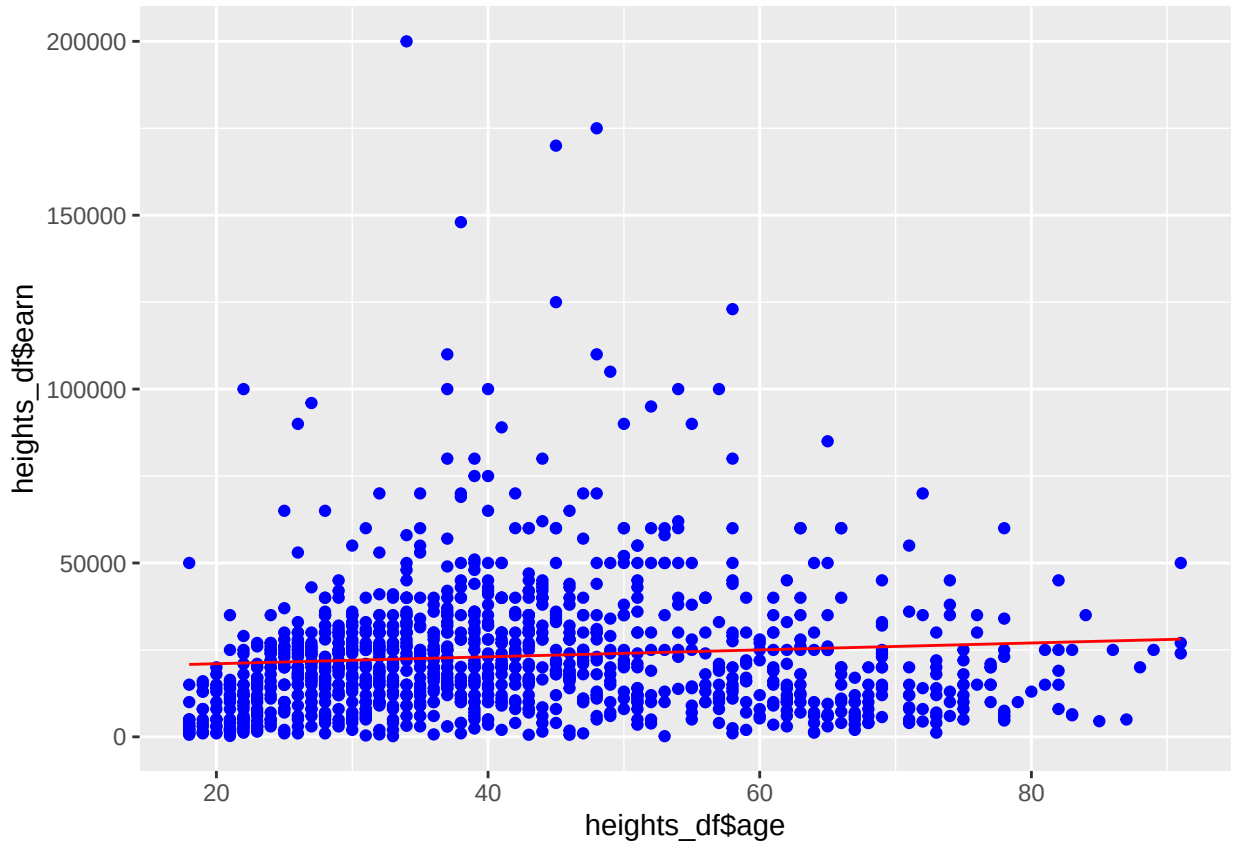
```r
#str(heights_df)
#mt_age <- data.frame(heights_df$age)
age_predict_df <- data.frame(earn = predict(age_lm, heights_df), age=heights_df$age)
plot(age_predict_df)
```




```
# - Reference https://www.youtube.com/watch?v=rfH7pCFvFT0 ( Linear Regression )
ggplot(data = heights_df, aes(x = heights_df$age, y = heights_df$earn)) +
  geom_point(color='blue') +
  geom_line(color='red',data = age_predict_df, aes(x=age_predict_df$age, y=age_predict_df$earn))
```

```
## Warning: Use of `heights_df$age` is discouraged.
## i Use `age` instead.
```

```
## Warning: Use of `heights_df$earn` is discouraged.
## i Use `earn` instead.
```




```
#ggplot(data = heights_df, aes(x = heights_df$age, y = heights_df$earn)) +
#  geom_point(color='blue') +
#  geom_smooth(method = "lm") +
#  geom_line(color='red',data = age_predict_df, aes(x=age_predict_df$age, y=age_predict_df$earn)) +
#  geom_smooth(method = "lm")
```

```
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
```

```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)
```