Workbook 2

Emma

03/03/2021

```r
# 7. Mathematical Equations

250+567
```

```
## [1] 817
```

```r
2+3
```

```
## [1] 5
```

```r
1234+5678
```

```
## [1] 6912
```

```r
345-78
```

```
## [1] 267
```

```r
65-10
```

```
## [1] 55
```

```r
5678-1234
```

```
## [1] 4444
```

```r
24*56
```

```
## [1] 1344
```

```r
2*4
```

```
## [1] 8
```

```r
6/3
```

```
## [1] 2
```

```r
54/8
```

```
## [1] 6.75
```

```r
# 8. Square Root of 324
sqrt(324)
```

```
## [1] 18
```

```r
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
# 9-11 Iris Dataset

# Flower with longest sepal length

max(iris$Sepal.Length)
```

```
## [1] 7.9
```

```r
# Flower with shortest sepal length

min(iris$Sepal.Length)
```

```
## [1] 4.3
```

```r
# Difference between longest and shortest length

(max(iris$Sepal.Length) - min(iris$Sepal.Length))
```

```
## [1] 3.6
```

```r
# 12. Making a new dataframe

df<-data.frame( a = c(1:100),
                b = c(1:100),
                c = c(1:100),
                d = c(1:100),
                e = c(1:100),
                f = c(1:100),
                g = c(1:100) )
```

```r
# 13. Renaming the dataframe

names(df)[]<- c("Grumpy", "Dopey", "Doc", "Happy", "Bashful", "Sneezy", "Sleepy")

str(df)
```

```
## 'data.frame':    100 obs. of  7 variables:
##  $ Grumpy : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Dopey  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Doc    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Happy  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Bashful: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Sneezy : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Sleepy : int  1 2 3 4 5 6 7 8 9 10 ...
```

```r
# 14. women dataset, Linear model.

w1<- lm(height ~ weight, data = women)
summary(women)
```

```
##      height         weight     
##  Min.   :58.0   Min.   :115.0  
##  1st Qu.:61.5   1st Qu.:124.5  
##  Median :65.0   Median :135.0  
##  Mean   :65.0   Mean   :136.7  
##  3rd Qu.:68.5   3rd Qu.:148.0  
##  Max.   :72.0   Max.   :164.0
```

```r
# 15. table
sjPlot::tab_model(w1)
```

height

Predictors

Estimates

CI

p

(Intercept)

25.72

23.47 – 27.98

<0.001

weight

0.29

0.27 – 0.30

<0.001

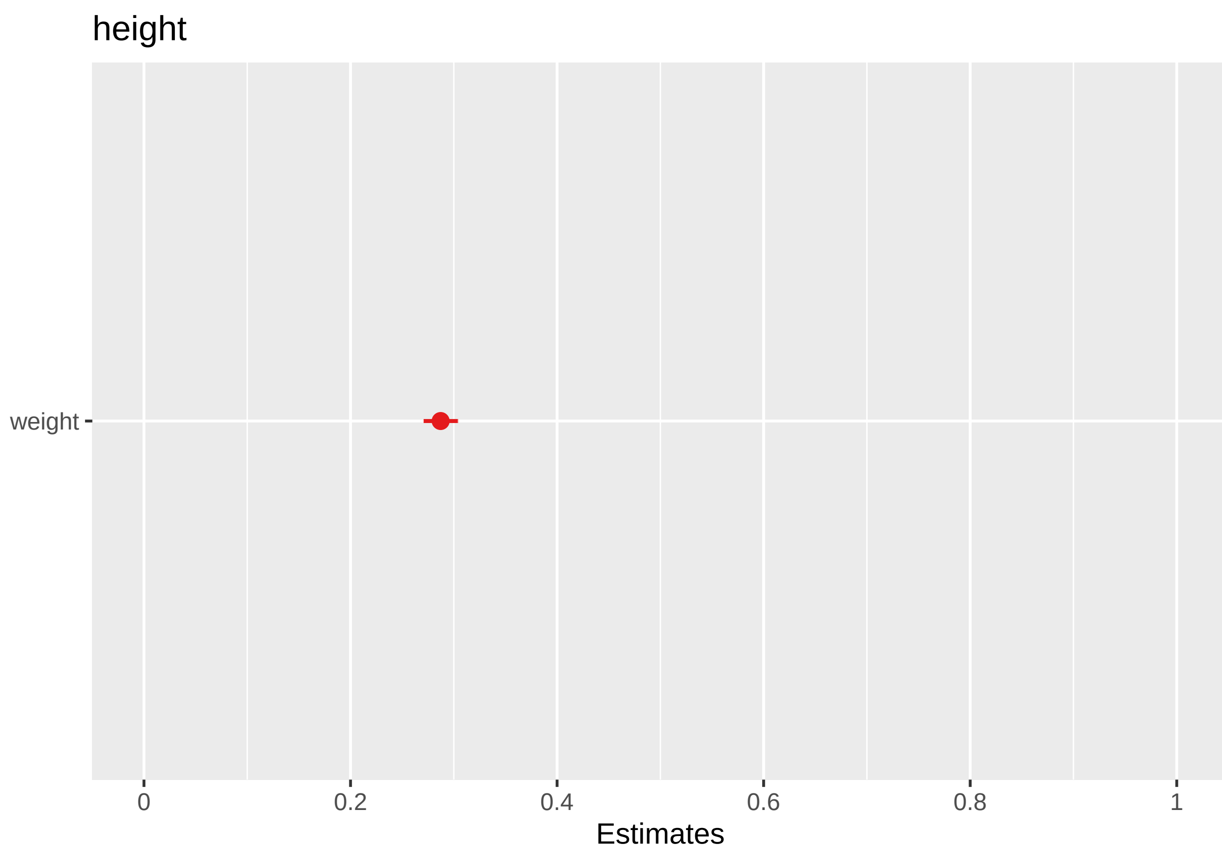
Observations

15

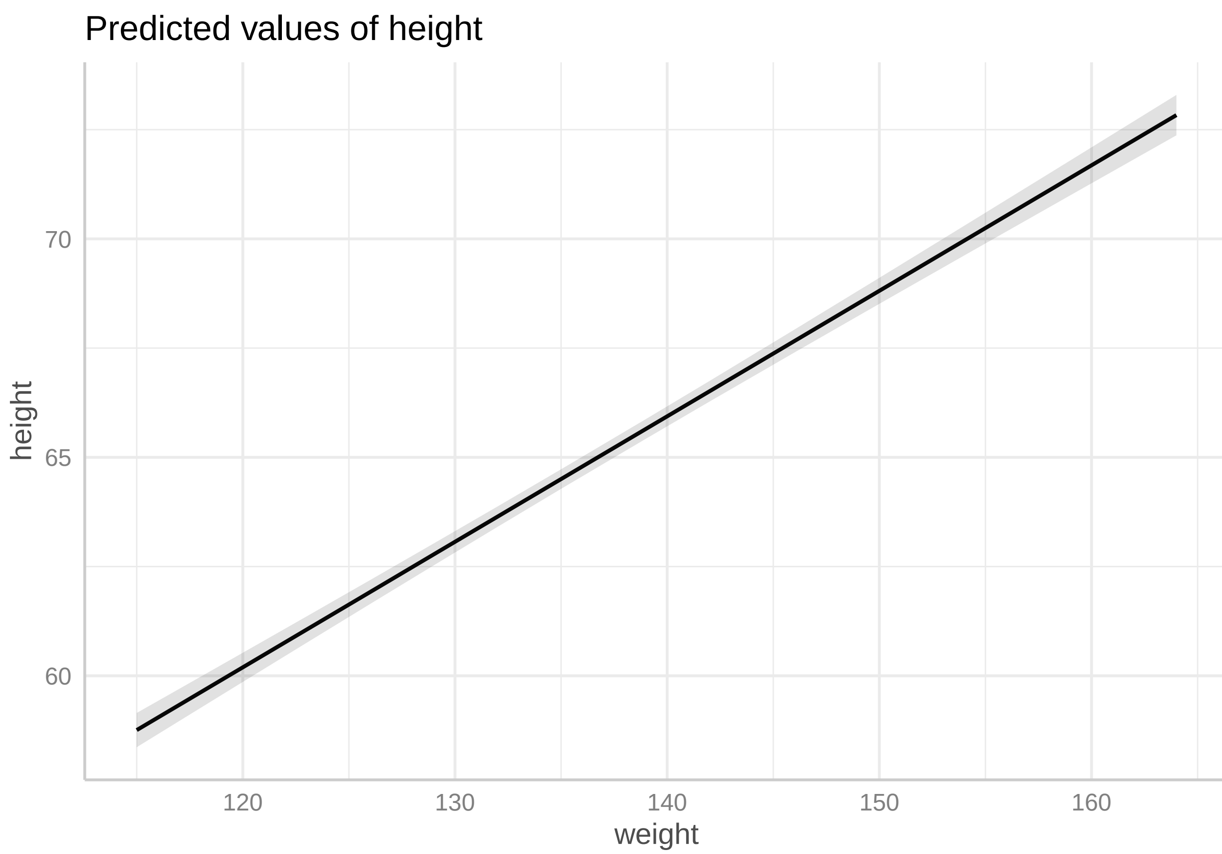
R2 / R2 adjusted

0.991 / 0.990

```r
# plot the coefficients
sjPlot::plot_model(w1)
```




```r
# 16. Using ggeffects, create a prediction plot for this model.
p1<- ggeffects::ggpredict(w1, terms = "weight")
plot(p1)
```




```r
# 17. this calculation gives you the proportion of women that weigh over 140llb.

sum(women$weight > 140) / length(women$weight)
```

```
## [1] 0.4
```

```r
# 18. calculate the mean and then calculate the proportion of women over that mean weight

sum(women$weight) / length(women$weight)
```

```
## [1] 136.7333
```

```r
mean(women$weight)
```

```
## [1] 136.7333
```

```r
sum(women$weight > 136.7333) / length(women$weight)
```

```
## [1] 0.4666667
```

```r
# 19. The advantages of having more breaks in the Petal Length indicator are that you can see more deta
```

```r
# 20. error in the below code
mh <- mean(women$height)
sum(women$weight > mh) / length(women$height)
```

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

```r
# the weight should be height. It calculates the proportion of womens heights under the mean height for

sum(women$height > mh) / length(women$height)
```

```
## [1] 0.4666667
```

```r
# 21. reorder columns of woman dataset so weight comes before height.
# Then rename columns w and h.

women2 <- women

women2[c("weight", "height")]
```

```
##    weight height
## 1     115     58
## 2     117     59
## 3     120     60
## 4     123     61
## 5     126     62
## 6     129     63
## 7     132     64
## 8     135     65
## 9     139     66
## 10    142     67
## 11    146     68
## 12    150     69
## 13    154     70
## 14    159     71
## 15    164     72
```

```r
names(women2)[] <- c("w", "h")

str(women2)
```

```
## 'data.frame':    15 obs. of  2 variables:
##  $ w: num  58 59 60 61 62 63 64 65 66 67 ...
##  $ h: num  115 117 120 123 126 129 132 135 139 142 ...
```

```r
#22. Read data into R using the following method:

library(readr)
testdata <- readr::read_csv(url("https://raw.githubusercontent.com/go-bayes/psych-447/main/data/testdata
```

```
##
```

```
## -- Column specification --------------------------------------------------------
## cols(
##   id = col_double(),
##   weight = col_double(),
##   height = col_double()
## )
```

```
str(testdata)
```

```
## spec_tbl_df [100 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id    : num [1:100] 1 2 3 4 5 6 7 8 9 10 ...
##  $ weight: num [1:100] 80.5 87.8 95.7 74.6 68.3 ...
##  $ height: num [1:100] 153 182 188 142 137 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   weight = col_double(),
##   ..   height = col_double()
##   .. )
```

```
# Save data into data folder

library(here)
```

```
## here() starts at C:/Users/emzfz/OneDrive/Documents/R Project 447/447-Workbook
```

```
saveRDS(testdata, here::here("data", "td1.RDS"))

# read data back into R

td <- readRDS(here::here("data", "td1.RDS"))
str(td)
```

```
## spec_tbl_df [100 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id    : num [1:100] 1 2 3 4 5 6 7 8 9 10 ...
##  $ weight: num [1:100] 80.5 87.8 95.7 74.6 68.3 ...
##  $ height: num [1:100] 153 182 188 142 137 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   weight = col_double(),
##   ..   height = col_double()
##   .. )
```

```
# 23. using td dataset, write linear model for height ~ weight as above

t1<- lm(height ~ weight, data = td)
summary(td)
```

```
##        id               weight            height
##  Min.   :  1.00   Min.   : 34.40   Min.   : 70.46
```

```
##  1st Qu.: 25.75   1st Qu.: 63.56   1st Qu.:132.89
##  Median : 50.50   Median : 76.36   Median :157.99
##  Mean   : 50.50   Mean   : 79.30   Mean   :161.06
##  3rd Qu.: 75.25   3rd Qu.: 92.75   3rd Qu.:186.89
##  Max.   :100.00   Max.   :138.55   Max.   :273.12
```

```
# 24. table
sjPlot::tab_model(t1)
```

height

Predictors

Estimates

CI

p

(Intercept)

12.74

5.51 – 19.98

0.001

weight

1.87

1.78 – 1.96

<0.001

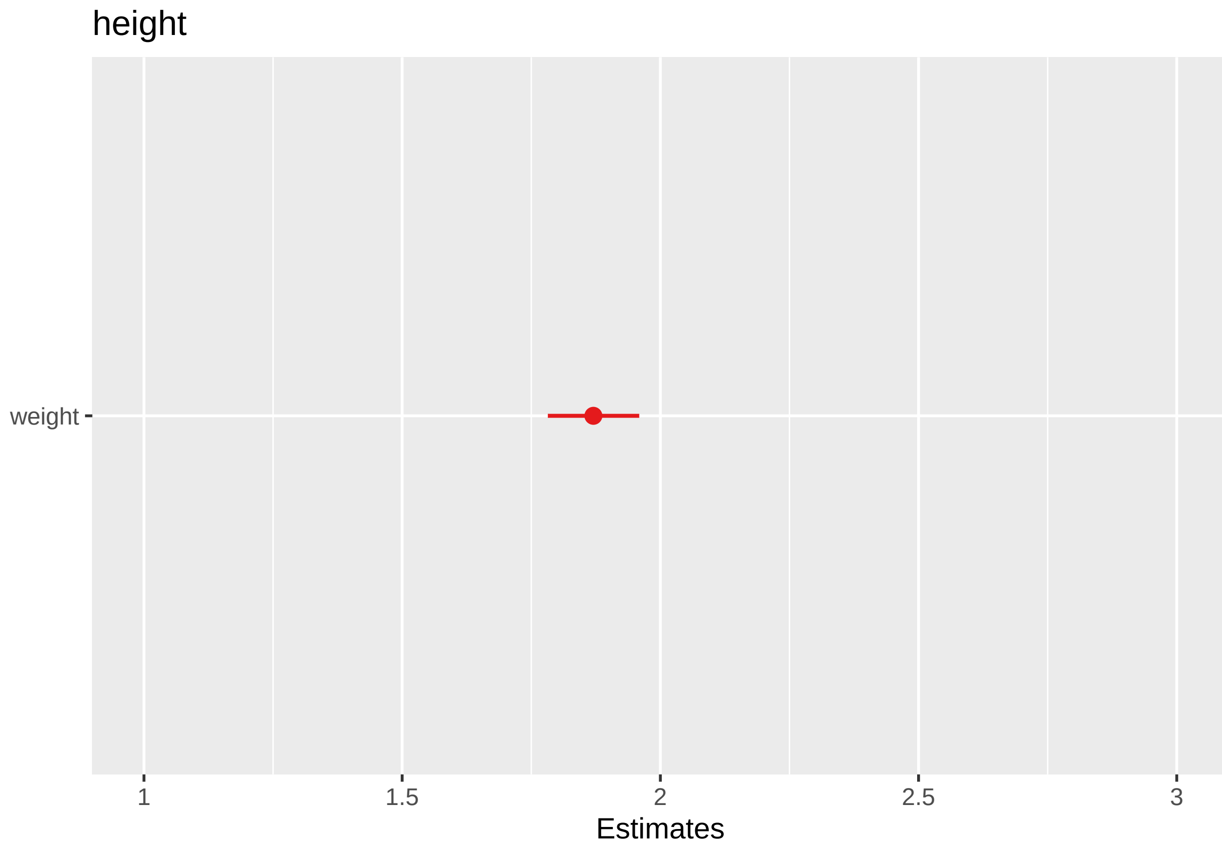
Observations

100

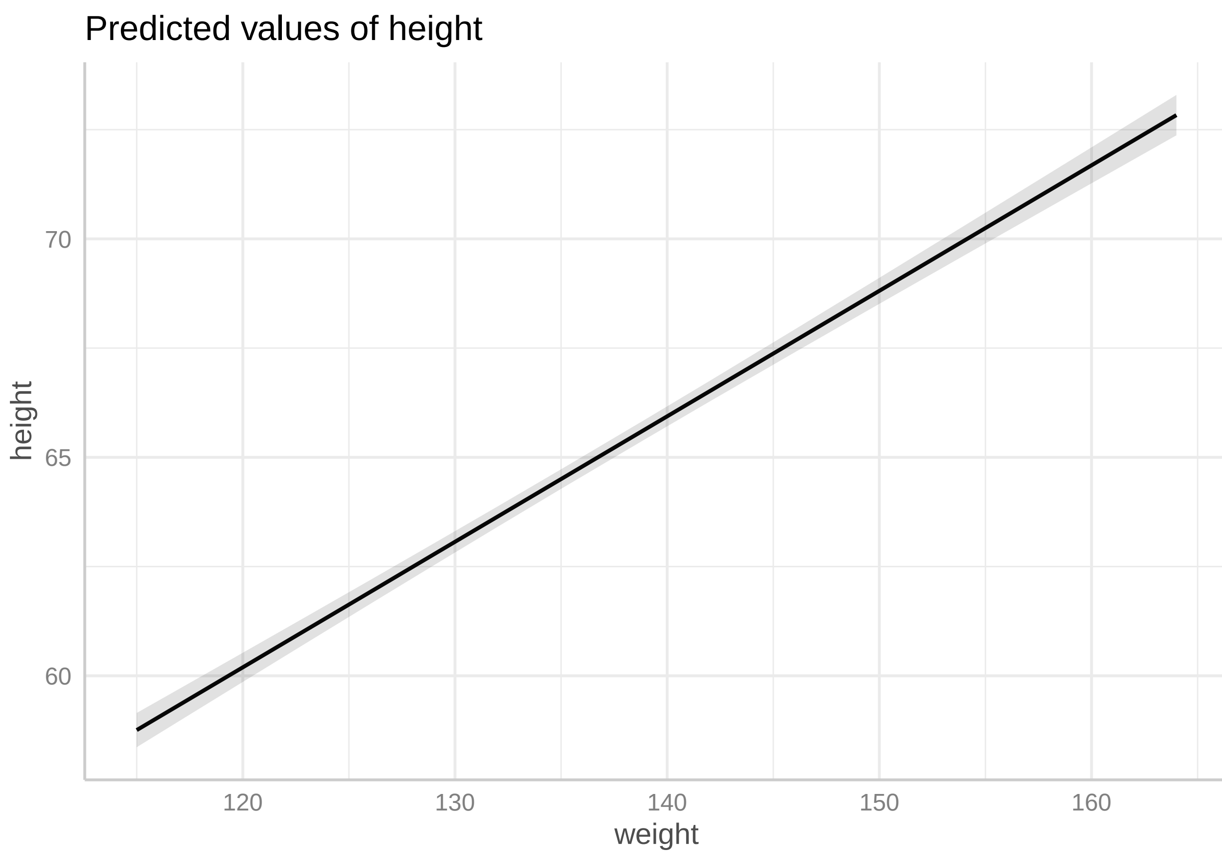
R2 / R2 adjusted

0.948 / 0.947

```
#coefficient plot
sjPlot::plot_model(t1)
```




```
# 25. Prediction plot for td dataset
pt1<- ggeffects::ggpredict(w1, terms = "weight")
plot(pt1)
```


Predicted values of height
70
65
60
height
120
130
140
150
160
weight