Lab 9 - Linear Regression 1

Nick Sumpter (Edited by Eddie-Williams Owiredu & Guy Twa)

2023-09-18

Today’s Lab

During this lab, we will learn how to use the `lm` function in R to produce a linear regression model, initially with a single predictor variable, and subsequently with more than one predictor. We will also learn how to report the results of these models in a tabular format, as well as in text.

Loading Packages and Data

For this lab we will just be using the `tidyverse` package.

```
library(tidyverse)

theme_set(theme_bw())
```

We will also be using a new dataset. This dataset includes 50 individuals from the Framingham Heart Study and can be loaded from the file `framingham.RData` , available on Canvas.

```
setwd("/Users/eddie-williamsowiredu/Desktop/grd770_23/Lab9")

load("framingham.RData")

source("functions.R")
```

If we look at the `fhs` object in our Environment, we can see that it includes the following variables:

1. `ID` : individual identifier (character)

2. `gender` : factor with 2 levels (male, female)

3. `age` : age in years (numeric)

4. `sysBP` : systolic blood pressure in mmHg (numeric)

5. `diaBP` : diastolic blood pressure in mmHg (numeric)
6. `glucose` : blood glucose in mg/dL (numeric)

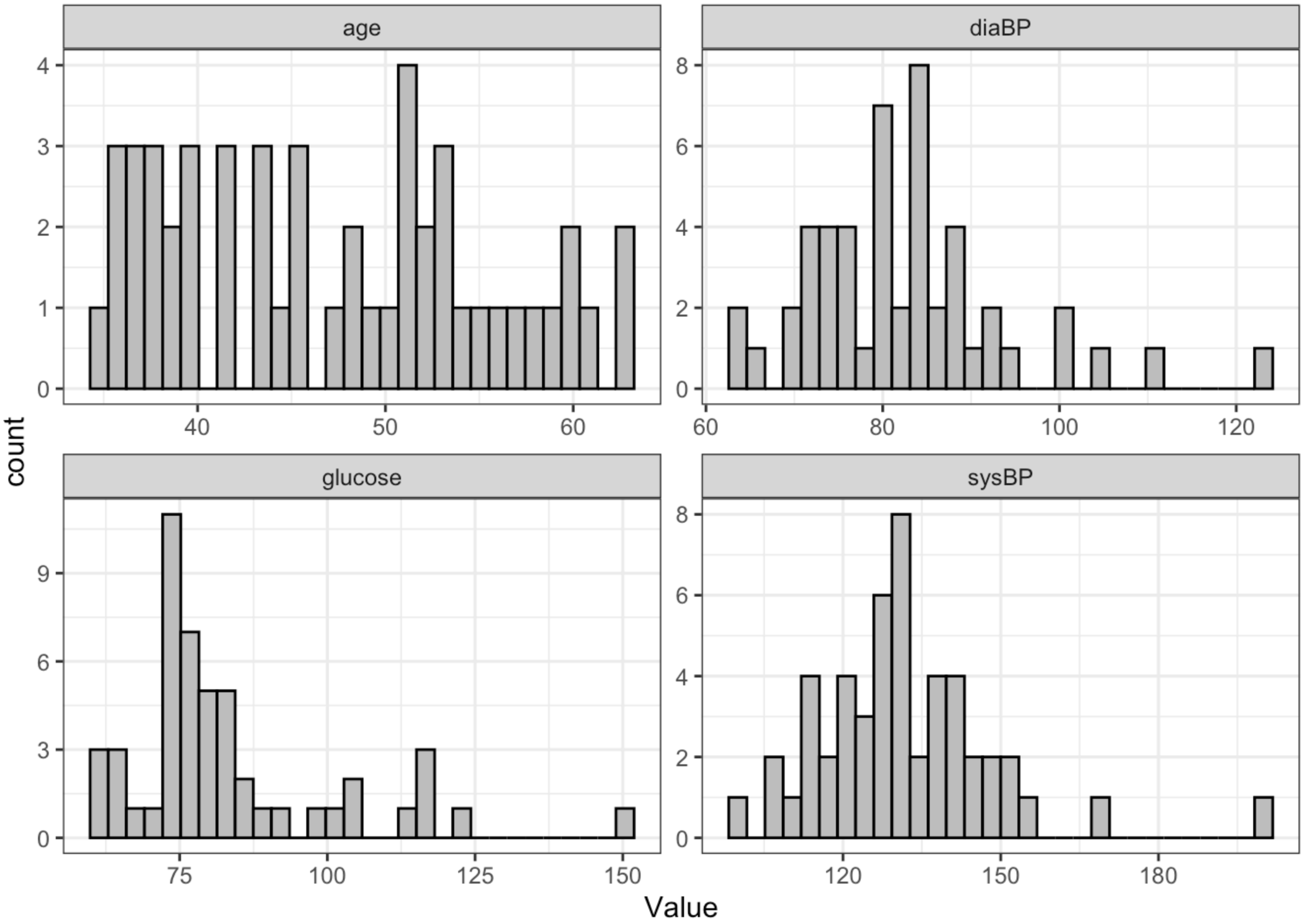
Getting to Know Your Data

We will be investigating the effect of both glucose and age on systolic blood pressure, testing each predictor individually, then together in a single model. First, let's get a summary of the whole dataset. Then we can plot the distribution of each of these variables. Finally, let's plot the relationship between these variables using a couple of scatter plots.

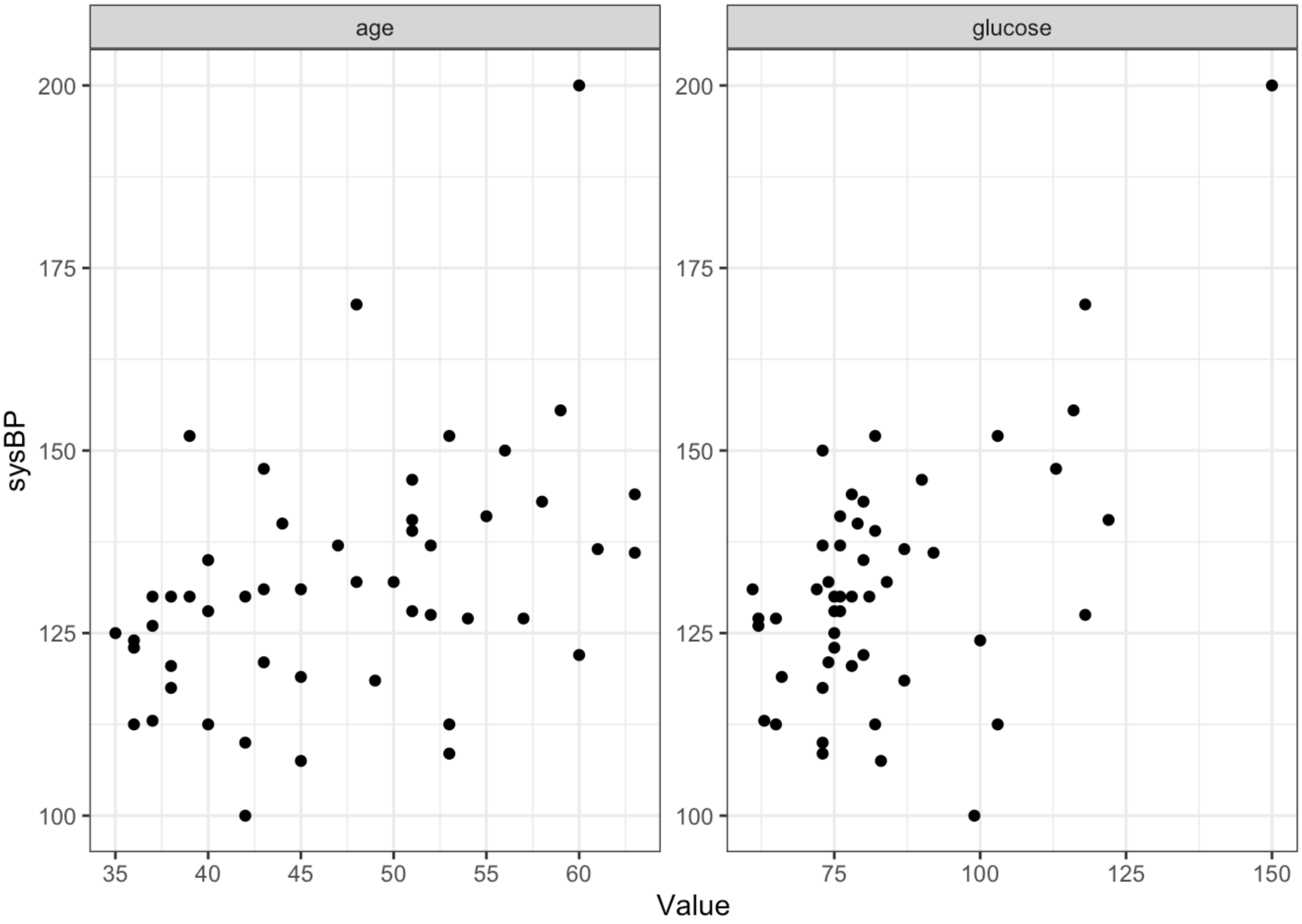
```
summary(fhs)
```

```
##       ID               gender        age            sysBP           diaBP       
##  Length:50          female:31   Min.   :35.0   Min.   :100.0   Min.   : 63.00  
##  Class :character   male  :19   1st Qu.:40.0   1st Qu.:121.2   1st Qu.: 74.25  
##  Mode  :character               Median :46.0   Median :130.0   Median : 81.50  
##                                 Mean   :47.1   Mean   :131.6   Mean   : 82.49  
##                                 3rd Qu.:53.0   3rd Qu.:139.8   3rd Qu.: 87.75  
##                                 Max.   :63.0   Max.   :200.0   Max.   :122.50  
##     glucose      
##  Min.   : 61.00  
##  1st Qu.: 73.25  
##  Median : 78.00  
##  Mean   : 83.56  
##  3rd Qu.: 87.00  
##  Max.   :150.00
```

```
# Transforming the data to make it easy to plot with facets
fhs2 <- fhs %>%
  select(-gender) %>%
  pivot_longer(cols = age:glucose,
               names_to = "Metric",
               values_to = "Value")

ggplot(data = fhs2, mapping = aes(x = Value)) +
  geom_histogram(bins = 30, fill = 'gray', color = 'black') +
  facet_wrap(~ Metric, scales = "free")
```



```
# Relationship between the variables
fhs3 <- fhs %>%
  select(-gender) %>%
  pivot_longer(cols = c(age, glucose),
               names_to = "Metric",
               values_to = "Value")

ggplot(data = fhs3, mapping = aes(x = Value, y = sysBP)) +
  geom_point() +
  facet_wrap(~ Metric, scales = "free")
```



Assumptions

We will be covering the assumptions of linear regression and how to test them in the next lab, for now just know that all variables in the model need to be continuous.

Simple Linear Regression

For any sort of basic linear model, the function you will use is `lm`. If you look it up in the Help panel you will see that `lm` has the following form:

```
lm(formula, data)
```

- `formula`: a formula expressing the outcome (dependent) variable as a function of the predictor (independent) variables. For example, if I wanted to create a model of y as a function of x, my formula would be `y ~ x`.
- `data`: the dataframe the variables can be found in

The output of `lm` is called a model object, which is essentially a list of attributes for the model. When running a linear model, it is important to assign the output of `lm` to an object in our Environment. You can then use the `summary` function on this object to create a table with the test statistics and some other information from that model.

In our dataset we have a few variables we could use as our dependent variable. For this first example, we can model systolic blood pressure as a function of age.

```
sysBPvsAge <- lm(formula = sysBP ~ age, data = fhs)

summary(sysBPvsAge)
```

```
##
## Call:
## lm(formula = sysBP ~ age, data = fhs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.293  -9.533   1.273   5.580  57.010
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  89.8743    12.9313   6.950 8.77e-09 ***
## age           0.8853     0.2705   3.272  0.00198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.57 on 48 degrees of freedom
## Multiple R-squared:  0.1824, Adjusted R-squared:  0.1653
## F-statistic: 10.71 on 1 and 48 DF,  p-value: 0.001981
```

The output of the `summary` function, when used on a model object, includes the function call, a summary of the residual errors, a summary of the coefficients for each model term, and some measures of error and variance explained. For understanding the output of the `sysBPvsAge` model, we will focus on the coefficients and error calculations. We can focus in on the Coefficients table by subsetting the model summary like so:

```
summary(sysBPvsAge)$coefficients
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 89.874300 12.9312652 6.950155 8.767720e-09
## age          0.885259  0.2705416 3.272174 1.981259e-03
```

the outcome var   predictor = 0

For the (Intercept) row, the Estimate column provides the value of sysBP when age = 0 (the model y-intercept). You can see that, according to the model, the baseline sysBP is 89.874 when age is 0. The standard error for this intercept term is then shown in the Std. Error column, followed by the t value and corresponding p value, both of which can be ignored for the intercept row.

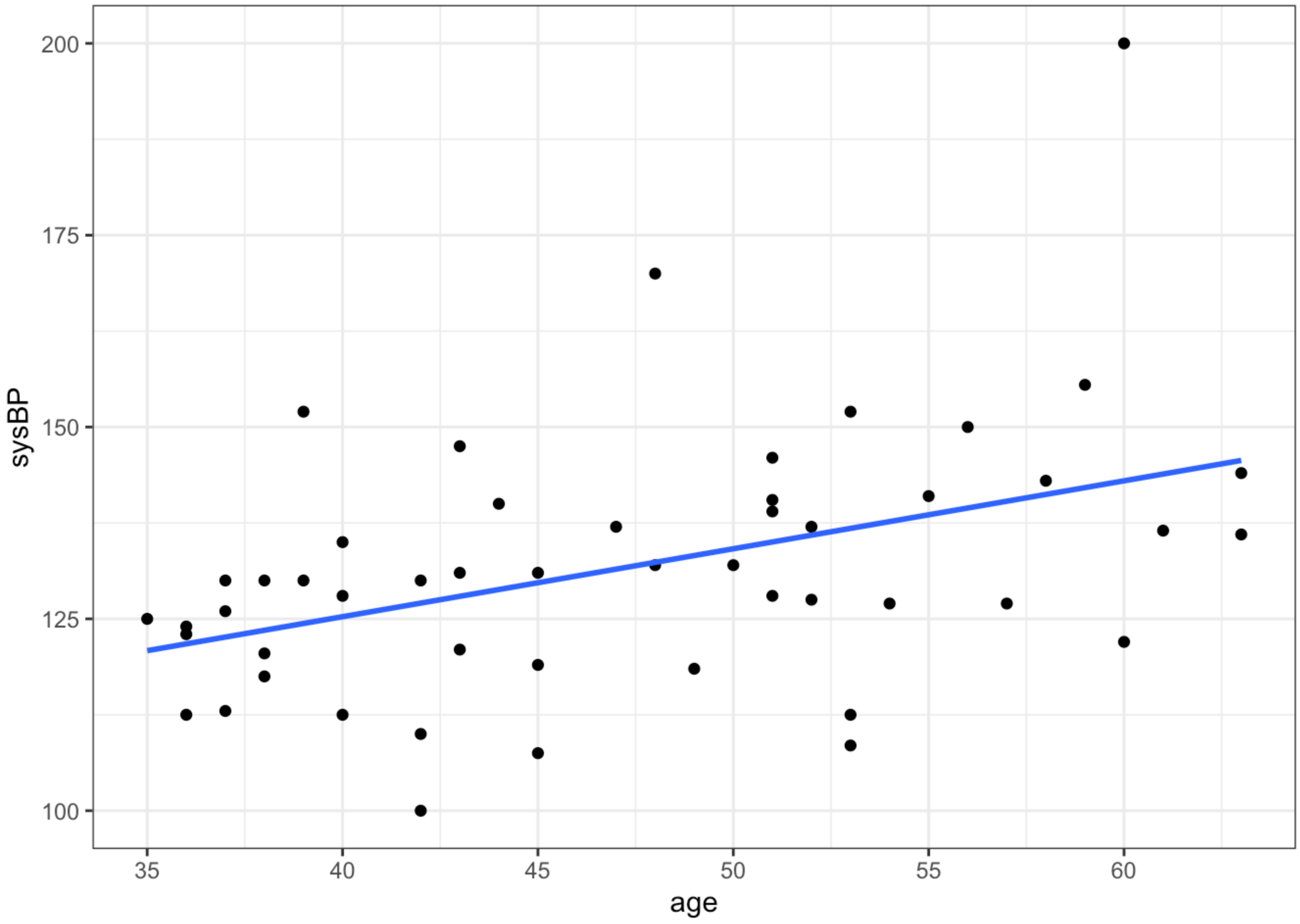
For the age row, the Estimate column provides the change in sysBP per 1 year increase in age. You can see that, according to the model, for each year a person ages, sysBP will change by 0.885. The standard error for the age term is then shown in the Std. Error column, followed by the t value and corresponding p value. For our purposes, we will focus on the p value (Pr(>|t|)) for interpreting whether the association of age with sysBP is significant, noting that this is derived from the t value, which is derived from the previous two columns. Based on the p value of 0.002, we can say that age is significantly associated with systolic blood pressure, confirmed by the ** significance code (indicating a p value between 0.001 and 0.01).

In the final three lines of the output, you can find the residual standard error calculation and the corresponding degrees of freedom. There are also two different versions of R-squared beneath that. For this model the Multiple R-squared value tells us that 18.238% of the variance in sysBP was explained by age in this model. The F-statistic indicates how well your model fits overall, and whether it is significantly different from a null model (i.e. using the mean age as a predictor of sysBP). For simple linear regression, the p-value of the full model is identical to the p-value of the predictor itself, and the F-value is equal to the squared t-value, highlighting their relationship.

Plotting the Regression

Plotting the regression for two variables is essentially the same as for plotting a correlation from last lab. We will plot using `geom_point` and add a regression line using `geom_smooth` with method being `lm`.

```
ggplot(fhs, aes(x = age, y = sysBP)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



Let's see the relationship between glucose and sysBP.

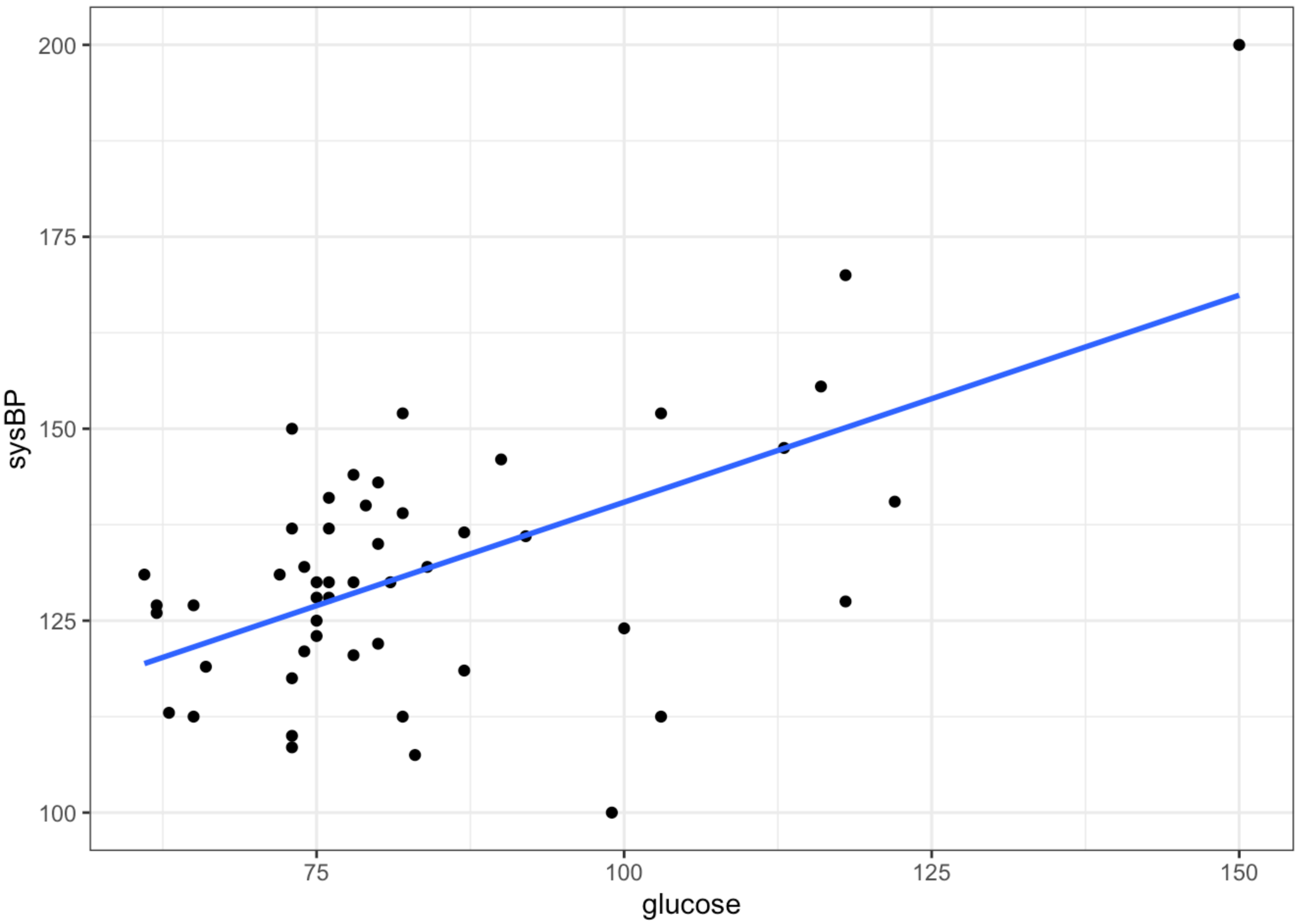
```
sysBPvsGlu <- lm(formula = sysBP ~ glucose, data = fhs)

summary(sysBPvsGlu)
```

```
##
## Call:
## lm(formula = sysBP ~ glucose, data = fhs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.893  -7.967   1.236   9.197  32.614
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  86.5248     9.5398   9.070 5.60e-12 ***
## glucose       0.5391     0.1116   4.829 1.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.12 on 48 degrees of freedom
## Multiple R-squared:  0.327,  Adjusted R-squared:  0.3129
## F-statistic: 23.32 on 1 and 48 DF,  p-value: 1.441e-05
```

```
summary(sysBPvsGlu)$coefficients
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 86.5248345  9.5398500 9.069832 5.598980e-12
## glucose      0.5390757  0.1116375 4.828807 1.440554e-05
```

```
ggplot(fhs, aes(x = glucose, y = sysBP)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



Multiple Linear Regression

Although making a model using only one predictor variable is valid, it doesn’t always tell the whole story. To add a second term to the model, we can use the generic formula

`y ~ x + z`

So here, we can model sysBP as a function of both age and glucose simultaneously:

```
sysBPvsAgeandGluc <- lm(formula = sysBP ~ age + glucose, data = fhs)

summary(sysBPvsAgeandGluc)
```

```
## 
## Call:
## lm(formula = sysBP ~ age + glucose, data = fhs)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -35.667  -4.662   3.054   7.687  30.106 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  65.0347    12.6703   5.133 5.36e-06 ***
## age           0.5939     0.2440   2.433 0.018815 *  
## glucose       0.4615     0.1110   4.158 0.000135 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 13.45 on 47 degrees of freedom
## Multiple R-squared:  0.4023, Adjusted R-squared:  0.3768 
## F-statistic: 15.81 on 2 and 47 DF,  p-value: 5.598e-06
```

You can see in the model summary that age and glucose are both significantly associated with sysBP. Also notice that the Estimates for both the Intercept and age have changed from previously. They adjust when new terms are added to the model.

Variance explained has increased by 21.988%. Generally, we can see if one model is better than another by looking at R-squared.

Reporting a Linear Regression

Reporting our results:

Multiple regression analysis was used to investigate whether glucose and age are significantly associated with systolic blood pressure in 50 individuals from the Framingham Heart Study. Together, age and glucose accounted for 40.23% of the variance (R-squared = 0.4023, p = 5.6e-6, N = 50). The partial regression coefficients for age and glucose were both significant [age: Estimate = 0.5939 ± 0.2440, t = 2.433, p = 0.019; glucose: Estimate = 0.4615 ± 0.1110, t = 4.158, p = 0.0001].

Explanation of the inserted variables:

- R-squared: Multiple R-squared term (not adjusted)
- p: found at the very bottom next to the F-Statistic
- N: Sample size

For each independent variable:

- Estimate: Estimate for that variable ± Std. Error for that variable
- t: t value from the table for that variable
- p: p value from the table for that variable

Independent Practice

Perform a multiple linear regression analysis to estimate the effect of glucose and age on diastolic blood pressure (diaBP).

1. Get to know your data: describe your variables and their distribution
2. Run the multiple linear regression test
3. Report your findings as you would describe them in a results section