Line Plot Basics

your name

2024-10-22

Math 2265 Chapter 8. Linear Regression

- Work as a group!
- You will need to replace `"ans"` or `your_answer` in the source code
- Update your name in L3
- Add your group members' name below; students may lose one point if Question 0 is unanswered
- Make sure you save and `knit` your work (to html or pdf) before submitting it to Canvas

---

Goal

- Review linear (line) equations
- Learn how to define a `function` in R
- Review how to define a `data frame` (data set) in R
- Review vector calculations in R
- Learn how to plot lines from a (linear) data set

Question 0. Who are your group members? (List their first names)

Answer:

1. `<name_1>`
2. `<name_2>`

If you need more time to get used to `Markdown`, use the `Visual` mode.

The icon is located in the upper-left corner next to `source`.

Functions

Roughly speaking, a function f is an assignment that for each value of x, there is exactly one output value y by the assignment $f(x)$. Often they are denoted by $f(x) =$ expression in x or $y =$ expression in x.

Here are some examples of functions.

Linear (line):

1. $f(x) = x$
2. $f(x) = -3x + 1$

Quadratic:

1. $f(x) = x^2$
2. $f(x) = -(x-3)^2 + 5$.

Example 1: Compute the y values of a given function

We will use $f(x) = -3x + 1$ as an example, but it applies to any function. Below, we compute $f(1), f(2), f(3)$.

```r
-3*(1) + 1
```

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

```r
-3*(2) + 1
```

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

```r
-3*(3) + 1
```

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

Indeed, it is often desired and convenient to use a vector.

```r
inputs <- c(1,2,3) # define a vector consisting of inputs
-3*inputs + 1       # compute all of them
```

```
## [1] -2 -5 -8
```

Task 1: Do the same for $f(x) = -(x-3)^2 + 5$

Compute $f(1), f(2), f(3)$, where $f(x) = -(x-3)^2 + 5$:

```r
-(1-3)^2 + 5
```

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

```r
-(2-3)^2 + 5
```

```
## [1] 4
```

```r
-(3-3)^2 + 5
```

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

Use the vector method to compute the same.

```r
inputs <- c(1,2,3)
-(inputs - 3)^2 + 5
```

```
## [1] 1 4 5
```

Example 2: Define a function

Let's revisit Example 1, where $f(x) = -3x + 1$. This time we will define the function $f(x)$ first, and use it to compute the rest.

```r
#define f(x) = -3x + 1
my_func <- function(x){
  -3*x + 1 # be sure to put * for multiplication
}

my_func(1); my_func(2); my_func(3) # f(1); f(2); f(3)
```

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

```
## [1] -5
## [1] -8
```

```
inputs <- c(1,2,3)
my_func(inputs)
```

```
## [1] -2 -5 -8
```

Task 2: Do the same for $f(x) = -(x-3)^2 + 5$

```
my_func2 <- function(x) {
  # write your answer below
  # your_answer
  -(x - 3)^2 + 5
}

my_func2(1); my_func2(2); my_func2(3)
```

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

```
## [1] 4
```

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

Using a vector:

```
inputs <- c(1, 2, 3)
my_func2(inputs)
```

```
## [1] 1 4 5
```

Example 3: Plot the line $f(x) = 3x - 1$

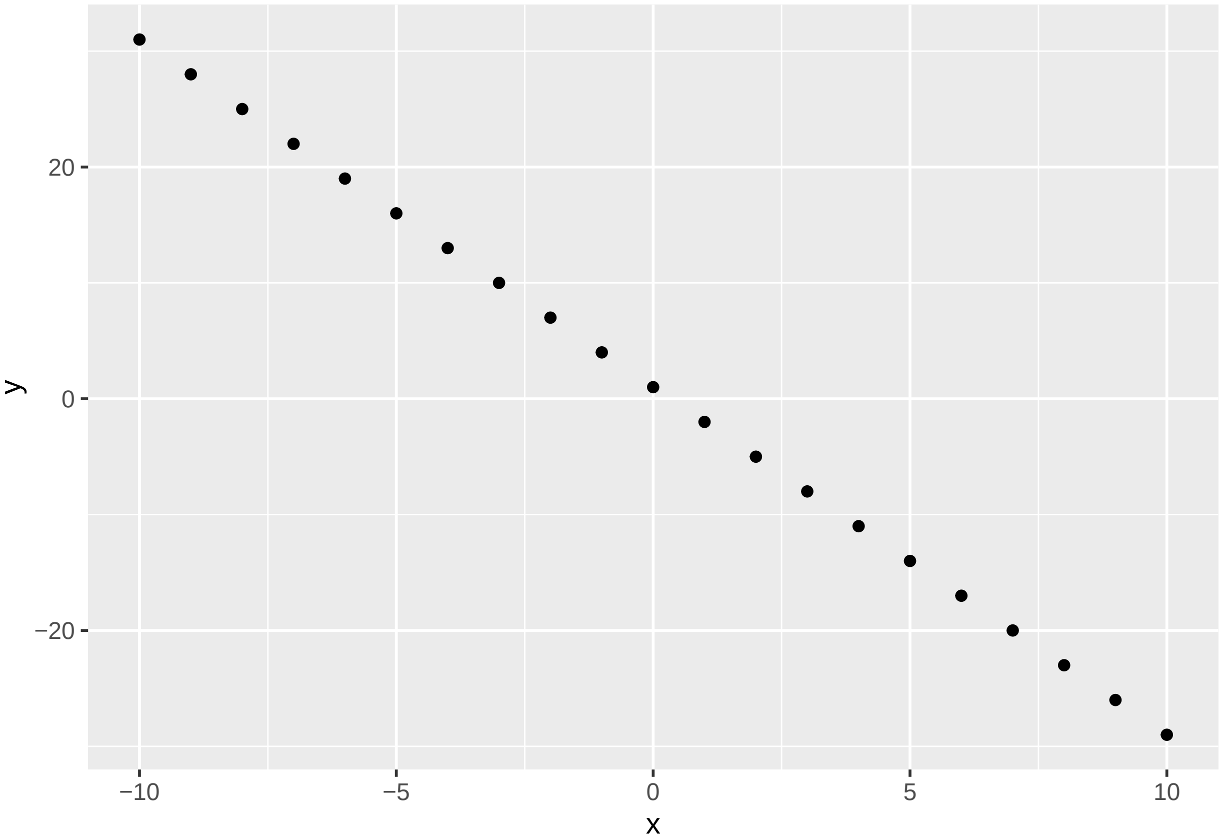
We use the function `my_func` defined above to plot the line $f(x) = 3x - 1$. First we make a scatter plot, and then we add a line. To use `ggplot2`, we need to make a data frame consisting of two variables x and y.

```
x <- c(-10:10)                  # c(-10:10) creates a vector consisting of -10,-9,..,10
y <- my_func(x)                 # compute y values from x values
my_dataframe <- data.frame(
  x,
  y
)
str(my_dataframe)
```

```
## 'data.frame':    21 obs. of  2 variables:
##  $ x: int  -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 ...
##  $ y: num  31 28 25 22 19 16 13 10 7 4 ...
```

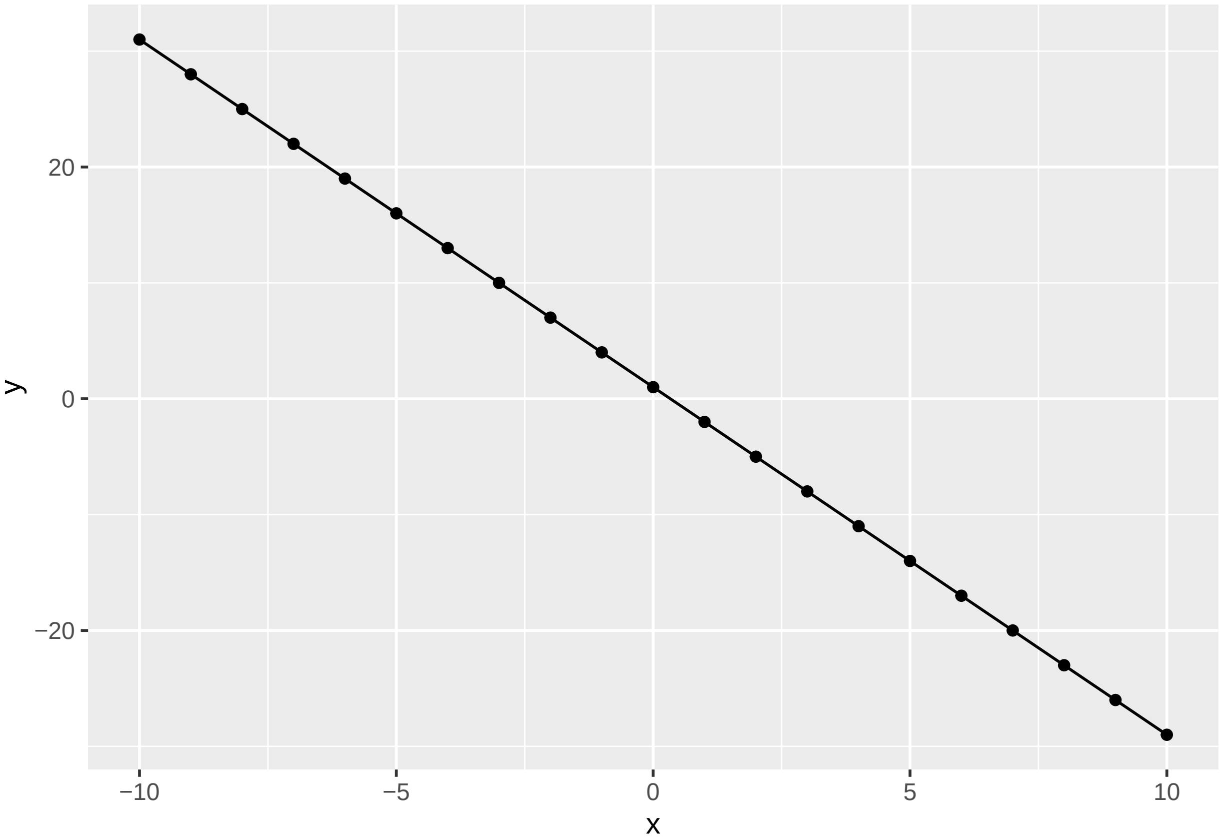
First we make a scatter plot:

```
ggplot(data=my_dataframe, mapping = aes(x=x, y=y)) +
  geom_point()
```



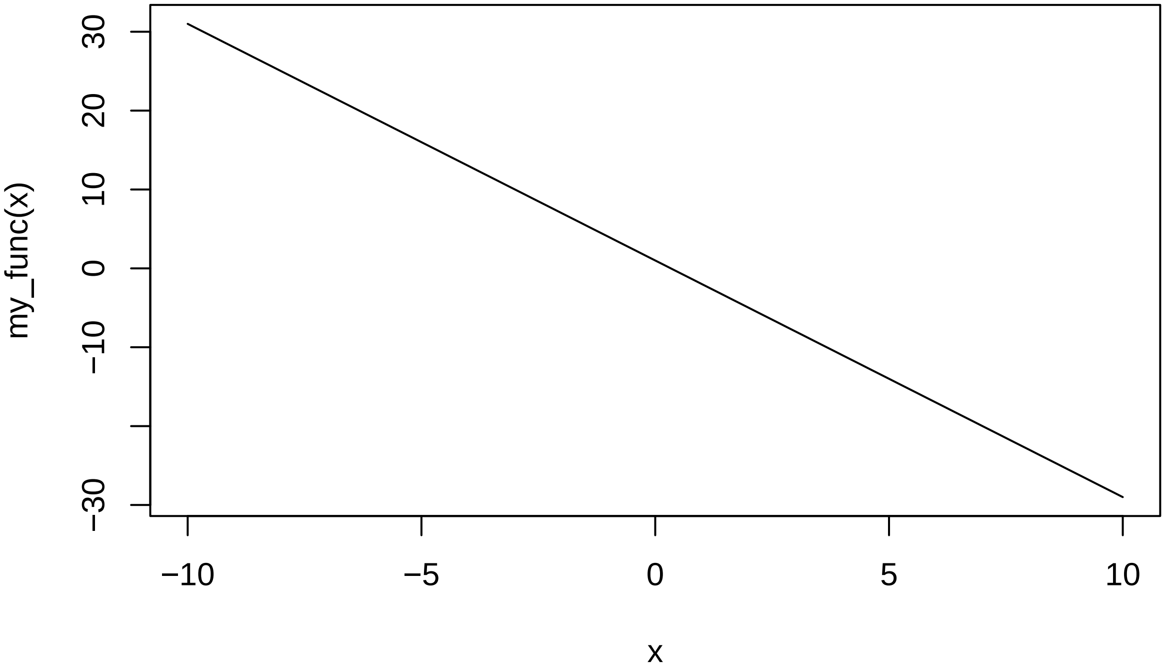
We connect the dots by adding `geom_line()`:

```
ggplot(data=my_dataframe, mapping = aes(x=x, y=y)) +
  geom_point() +
  geom_line()
```

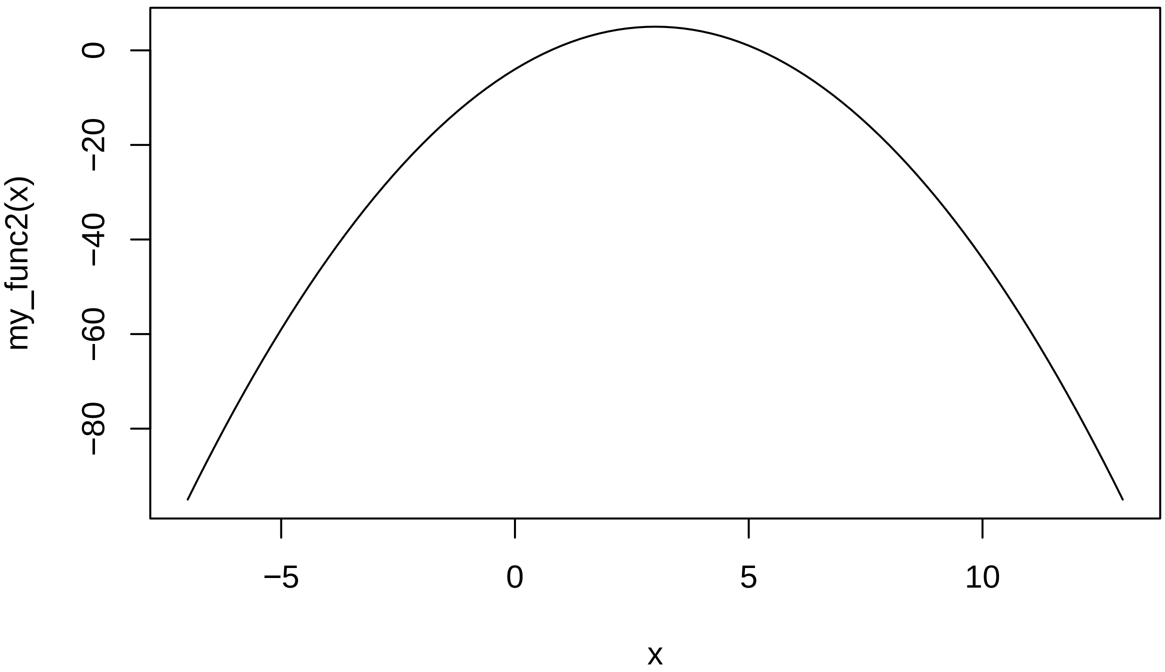


Indeed, there is a function `curve` which sketches the function `f(x)` without needing to define a data frame.

```r
curve(my_func, from=-10,to=10)
```

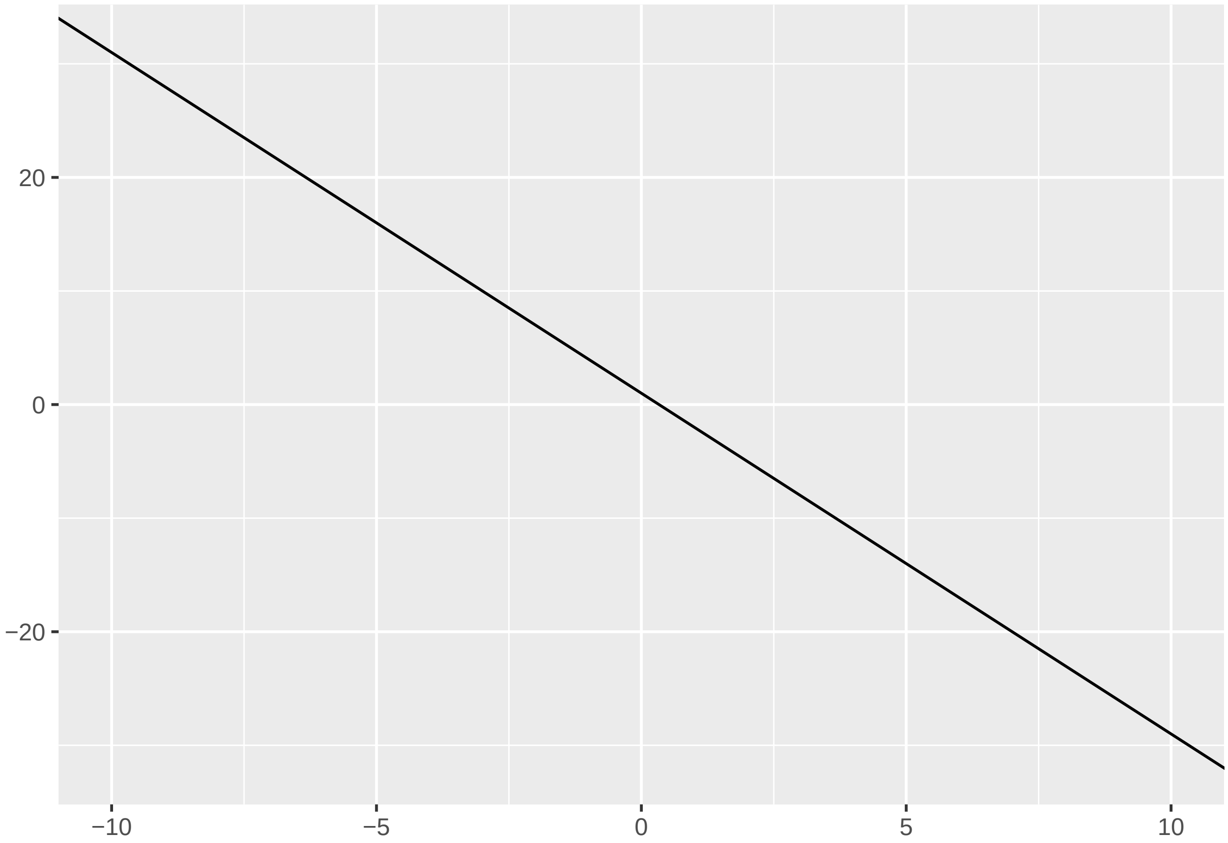


```
curve(my_func2, from=-7,to=13)
```



However, for the purpose of linear regression, we will stick to the `ggplot2` package. For lines, the `ggplot2` package has the `abline` function which takes the slope and intercept of a line as parameters. Here is an example for $f(x) = -3x + 1$ whose slope is -3 and y-intercept is 1.

```
ggplot() +
  geom_abline(slope = -3, intercept = 1) +
  xlim(-10,10) +
  ylim(-32,32)
```



Question:

We will create the possum example in the textbook link:

- **data set**: `possum`
- **explanatory variable**: `total_l`
- **response variable**: `head_l`

```
ggplot(data = possum, mapping = aes(x=total_l, y=head_l)) +
  geom_point()
```

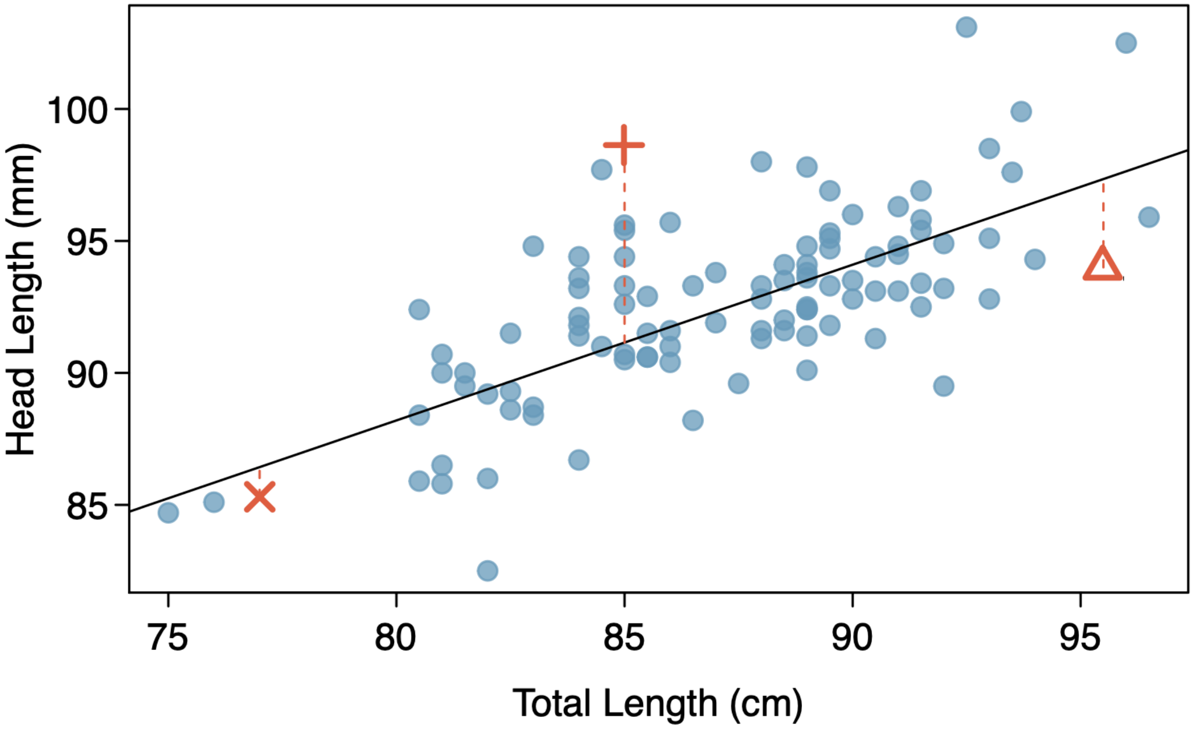
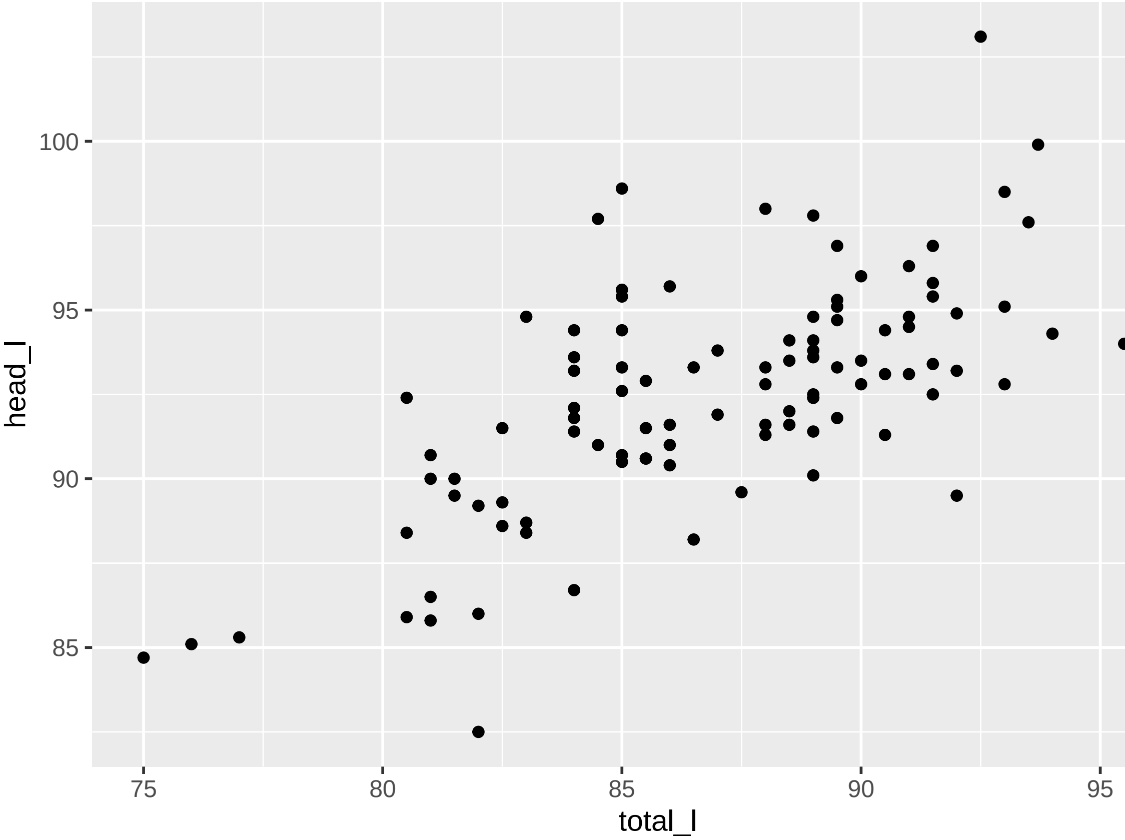


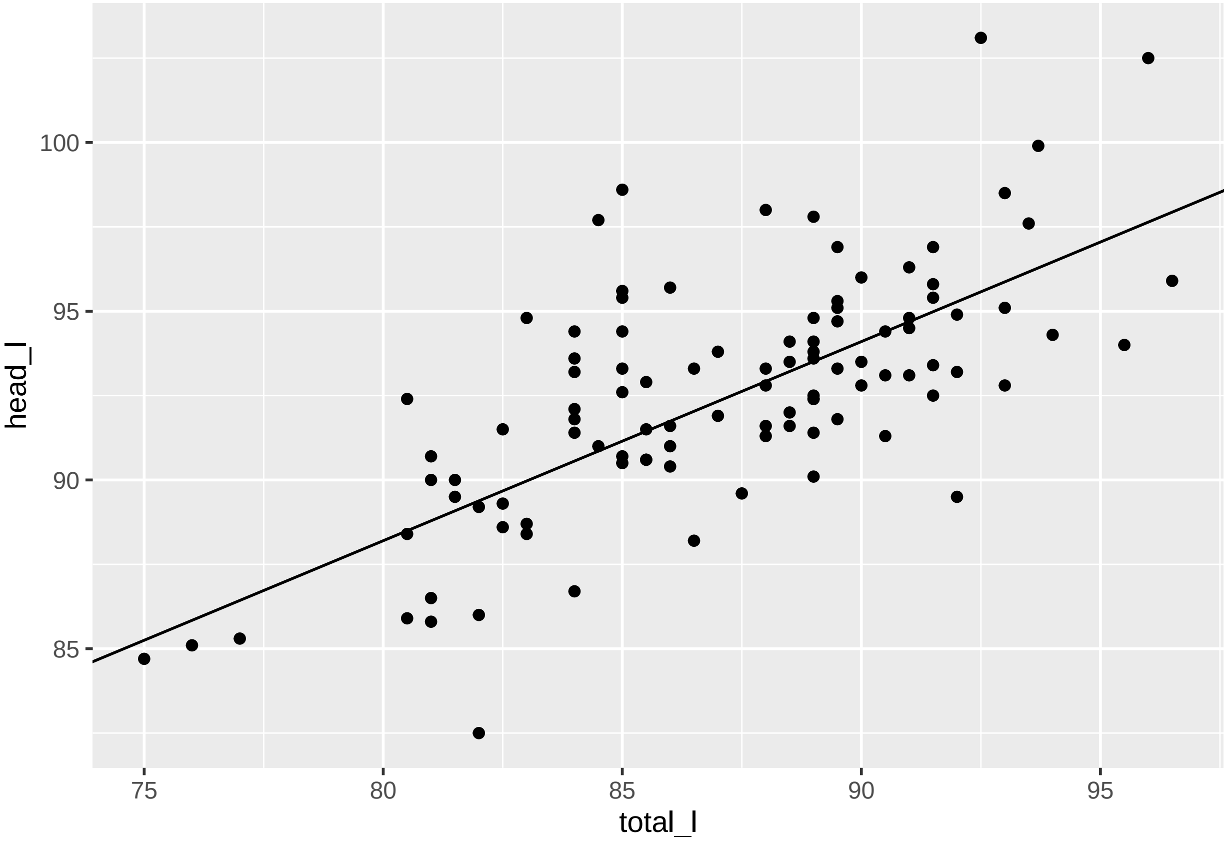
Figure 1: example



(a) Make a scatter plot

(b) Add the line with slope 0.59 and intercept 41

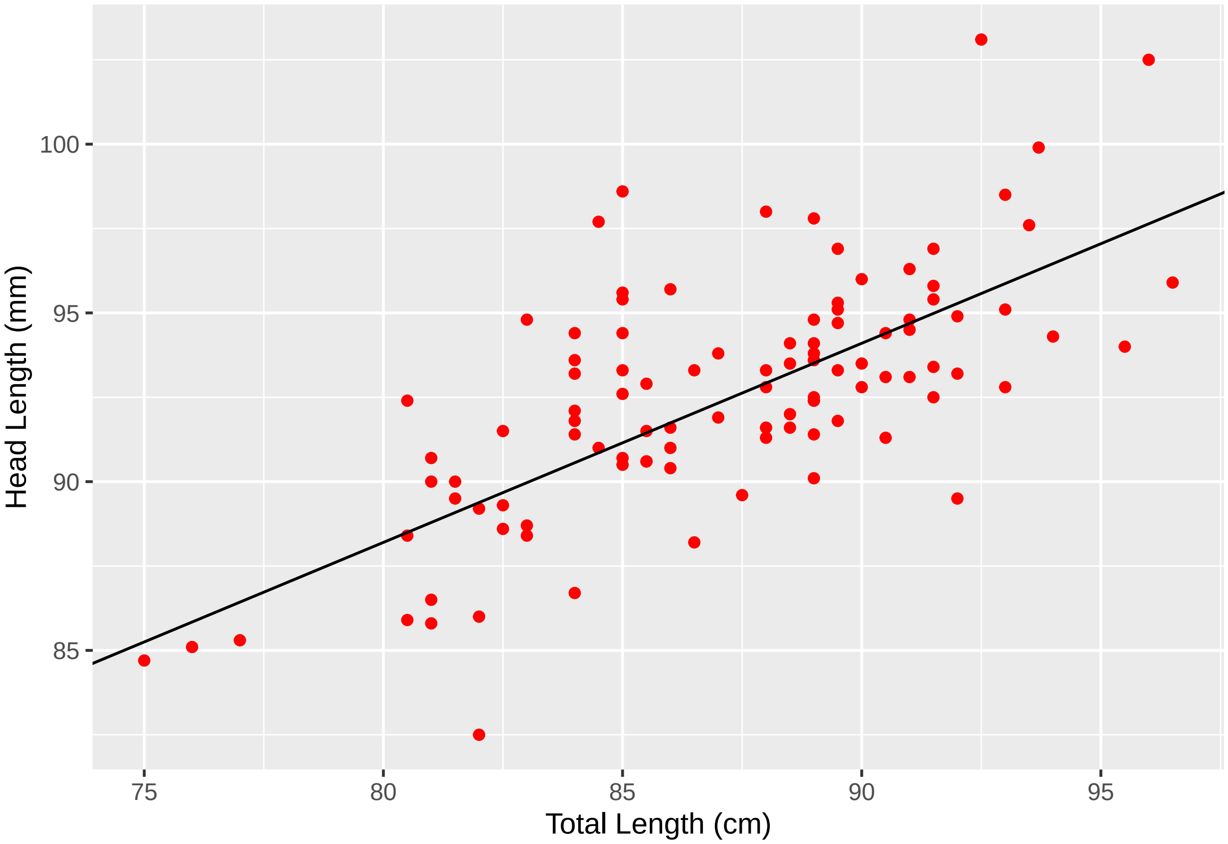
```
ggplot(data = possum, mapping = aes(x=total_l, y=head_l)) +
  geom_point() +
  geom_abline(slope = 0.59, intercept = 41)
```



(c) Decorations

We have not been concerned aesthetics so far, but with the a few commands we can make it look similar to the one in the book. For instance, change `red` to `blue` and lower the alpha value. Also change "x_title" and "y_title" by "Total Length (cm)" and "Head Length (mm)", respectively.

```
ggplot(data = possum, mapping = aes(x=total_l, y=head_l)) +
  geom_point(color='red', alpha = 1) +
  geom_abline(slope = 0.59, intercept = 41) +
  xlab("Total Length (cm)") +
  ylab("Head Length (mm)")
```



That's all for today! We will learn how to fit lines and find R next time.

Share your work and help your group members before uploading your work to Canvas