Linear Regression Basics

your name

2024-10-24

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 the basic concepts in linear regression
  - Meaning of line fitting
    - Residuals
  - The R-value
  - When not to use linear regression
    - Linearity
    - Nearly normal residuals
    - Constant variability
    - Independent observations
- Next time we will learn how to use an `R`-function to find the least squares line

---

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`.

Line fitting

We will use the possums data set:

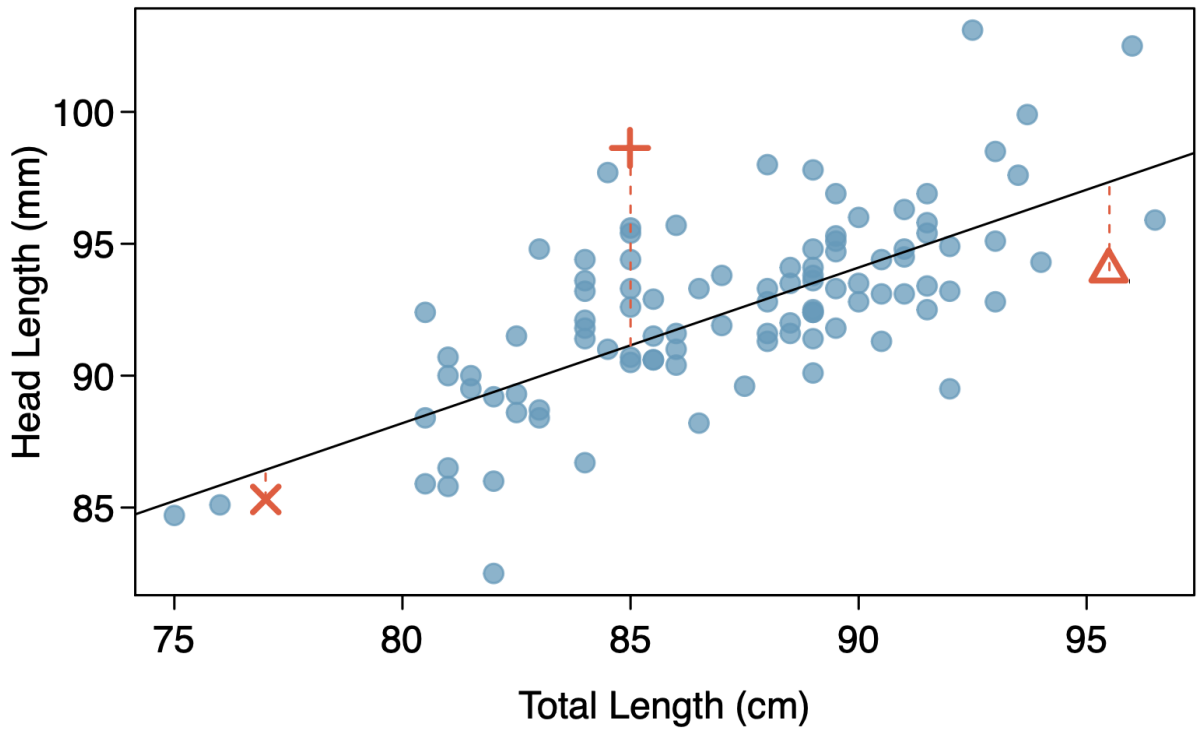


Figure 1: example

Question:

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

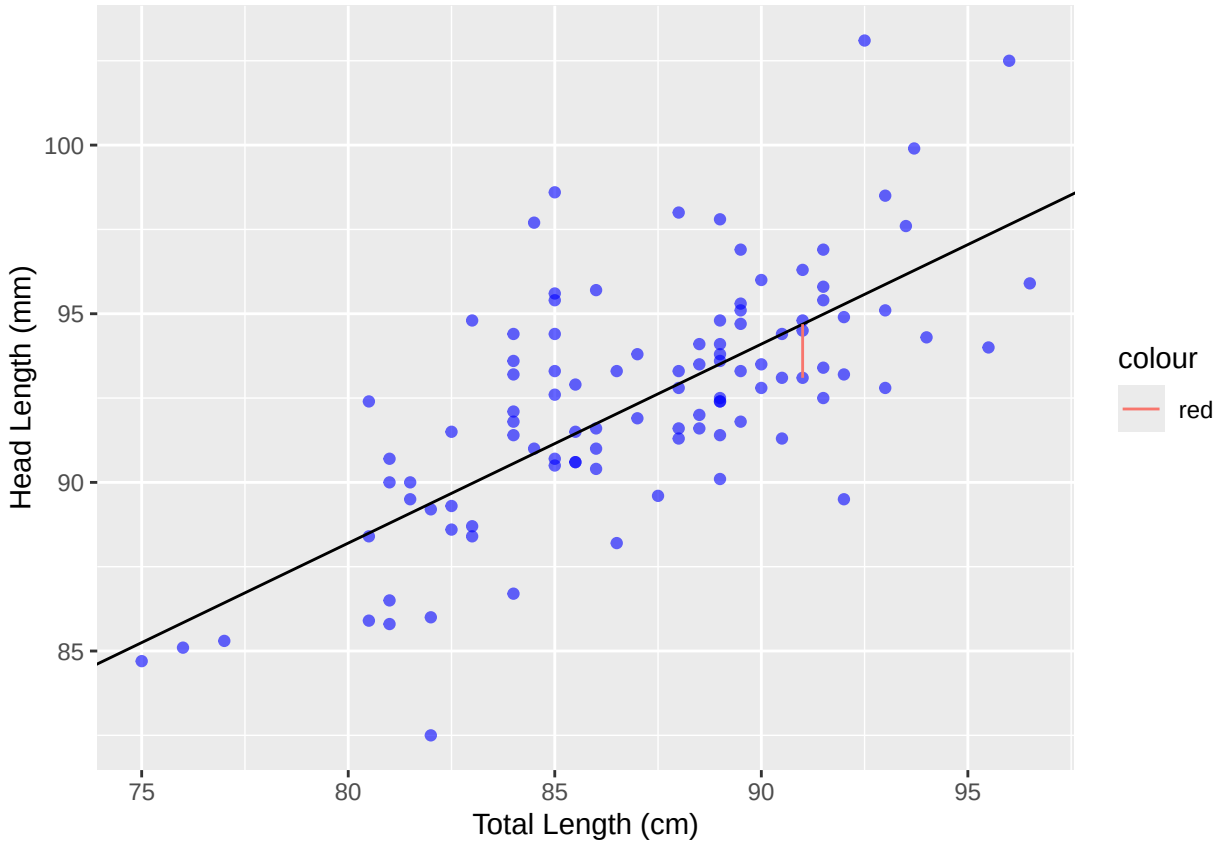
Here is the scatter plot of the data.

```
# We define the function of the line and use `_hat` since it is an estimate
y_hat <- function(x) {
  41 + 0.59 * x
}

# We choose one sample point to demonstrate the residual
some_number = 53 # try different numbers between 1 and 104
possum$total_l[some_number]
```

```
## [1] 91
```

```
sample_x <- possum$total_l[some_number]
sample_y <- possum$head_l[some_number]

ggplot(data = possum, mapping = aes(x=total_l, y=head_l)) +
  geom_point(color='blue', alpha = 0.6) +
  geom_abline(slope = 0.59, intercept = 41) +
  # the following line draws the residual in red
  geom_line(data = data.frame(x = c(sample_x, sample_x), y = c(sample_y, y_hat(sample_x))), aes(x = x, y
  xlab("Total Length (cm)") +
```

```
  ylab("Head Length (mm)")
```



Computing residuals with respect to this line $y = 41 + 0.59x$ is fairly simple. Recall that the residual is the true value of y minus the estimation $\hat{y}$.

The explanatory variable `total_l` plays the role of x, and the response variable `head_l` plays the role of y. Let's check these two variables first.

```
library(dplyr)
```

```
##
## 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
```

```
possum %>% select(total_l, head_l)
```

```
## # A tibble: 104 x 2
##    total_l head_l
##      <dbl>  <dbl>
##  1    89     94.1
##  2    91.5   92.5
##  3    95.5   94
##  4    92     93.2
```

```
##  5    85.5   91.5
##  6    90.5   93.1
##  7    89.5   95.3
##  8    91     94.8
##  9    91.5   93.4
## 10    89.5   91.8
## # i 94 more rows
```

We will add another column `head_l_hat` containing the estimations by using the line.

```
possum$head_l_hat <- y_hat(possum$total_l)
head(possum$head_l_hat)
```

```
## [1] 93.510 94.985 97.345 95.280 91.445 94.395
```

Now we display the three columns.

```
# if you see an error, run library(dplyr) first
# library(dplyr)
possum %>% select(total_l, head_l, head_l_hat)
```

```
## # A tibble: 104 x 3
##    total_l head_l head_l_hat
##      <dbl>  <dbl>      <dbl>
##  1    89     94.1       93.5
##  2    91.5   92.5       95.0
##  3    95.5   94         97.3
##  4    92     93.2       95.3
##  5    85.5   91.5       91.4
##  6    90.5   93.1       94.4
##  7    89.5   95.3       93.8
##  8    91     94.8       94.7
##  9    91.5   93.4       95.0
## 10    89.5   91.8       93.8
## # i 94 more rows
```

The residual for each point is the value of head_l minus head_l_hat. The vector calculation takes care of it. We will compute it, attach it to the data frame, and display the four columns.

```
# if you see an error, run the previous cell first
possum$residuals <- possum$head_l - possum$head_l_hat
possum %>% select(total_l, head_l, head_l_hat, residuals)
```

```
## # A tibble: 104 x 4
##    total_l head_l head_l_hat residuals
##      <dbl>  <dbl>      <dbl>     <dbl>
##  1    89     94.1       93.5    0.590
##  2    91.5   92.5       95.0   -2.48
##  3    95.5   94         97.3   -3.34
##  4    92     93.2       95.3   -2.08
##  5    85.5   91.5       91.4    0.0550
##  6    90.5   93.1       94.4   -1.30
##  7    89.5   95.3       93.8    1.49
##  8    91     94.8       94.7    0.110
##  9    91.5   93.4       95.0   -1.58
## 10    89.5   91.8       93.8   -2.01
## # i 94 more rows
```

The positive residuals mean that the points are lying above the line, and the negative residuals mean the points are below the line. The `least squares` line means that the sum of squares of the residuals is the least among all possible line fittings. In this example, the value is

```
sum_squares <- sum( (possum$residuals)^2 )
sum_squares
```

```
## [1] 692.6659
```

The calculation below demonstrates that this actually is not the least squares line.

Lastly, we can compute the R-value (the correlation factor) using the function `cor`.

```
r <- cor(possum$total_l, possum$head_l, method = "pearson")
r; r^2 # R and R^2
```

```
## [1] 0.6910937
```

```
## [1] 0.4776105
```

Next time

```
model <- lm(head_l ~ total_l, data = possum)
summary(model)
```

```
##
## Call:
## lm(formula = head_l ~ total_l, data = possum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1877 -1.5340 -0.3345  1.2788  7.3968
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.70979    5.17281   8.257 5.66e-13 ***
## total_l      0.57290    0.05933   9.657 4.68e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.595 on 102 degrees of freedom
## Multiple R-squared:  0.4776, Adjusted R-squared:  0.4725
## F-statistic: 93.26 on 1 and 102 DF,  p-value: 4.681e-16
```

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