R4Stats

Colin Madland

2023-03-04

Contents

About

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports; for example, a math equation $a^2 + b^2 = c^2$.

Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select "All formats" if you'd like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in "Preview book", or from the R console:

```
bookdown::serve_book()
```

Chapter 1

Getting Started with R

1.1 Problems

```r
1+2*(3+4)
```

```
## [1] 15
```

```r
(4^3)+(3^(2+1))
```

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

```r
sqrt(4+3)*(2+1)
```

```
## [1] 7.937254
```

```r
(1+(2*3^4))/(5/6)-7
```

```
## [1] 188.6
```

```r
(0.25 - 0.2) / (0.2 * (1 - 0.2)/100)^(1/2)
```

```
## [1] 1.25
```

```
x <- 2
y <- 3
z <- 4
w <- 5

x*y*z*w
```

```
## [1] 120
```

```
rivers
```

```
##   [1]  735  320  325  392  524  450 1459  135  465  600  330  336  280  315  870
##  [16]  906  202  329  290 1000  600  505 1450  840 1243  890  350  407  286  280
##  [31]  525  720  390  250  327  230  265  850  210  630  260  230  360  730  600
##  [46]  306  390  420  291  710  340  217  281  352  259  250  470  680  570  350
##  [61]  300  560  900  625  332 2348 1171 3710 2315 2533  780  280  410  460  260
##  [76]  255  431  350  760  618  338  981 1306  500  696  605  250  411 1054  735
##  [91]  233  435  490  310  460  383  375 1270  545  445 1885  380  300  380  377
## [106]  425  276  210  800  420  350  360  538 1100 1205  314  237  610  360  540
## [121] 1038  424  310  300  444  301  268  620  215  652  900  525  246  360  529
## [136]  500  720  270  430  671 1770
```

```
Orange
```

```
##    Tree  age circumference
## 1     1  118            30
## 2     1  484            58
## 3     1  664            87
## 4     1 1004           115
## 5     1 1231           120
## 6     1 1372           142
## 7     1 1582           145
## 8     2  118            33
## 9     2  484            69
## 10    2  664           111
## 11    2 1004           156
## 12    2 1231           172
## 13    2 1372           203
## 14    2 1582           203
## 15    3  118            30
## 16    3  484            51
## 17    3  664            75
## 18    3 1004           108
```

```
## 19    3 1231           115
## 20    3 1372           139
## 21    3 1582           140
## 22    4  118            32
## 23    4  484            62
## 24    4  664           112
## 25    4 1004           167
## 26    4 1231           179
## 27    4 1372           209
## 28    4 1582           214
## 29    5  118            30
## 30    5  484            49
## 31    5  664            81
## 32    5 1004           125
## 33    5 1231           142
## 34    5 1372           174
## 35    5 1582           177
```

```
mean(Orange$age)
```

```
## [1] 922.1429
```

```
max(Orange$circumference)
```

```
## [1] 214
```

Chapter 2

Univariate data

Levels of measurement

The view in most textbooks is from Stanley Smith Stevens (1964)

Definition 2.1. Nominal
Such data is qualitative or descriptive, but not numeric. An example might be the name of a person or the town they are from, or the number on a bib a runner wears in a race.

Definition 2.2. Ordinal
Ordinal data is data with some order, so that we can sort the data from largest to smallest. An example might be the place a runner takes in a race.

Definition 2.3. Interval
Interval data is ordinal data where the difference between two values has some interpretation. The clock time a person finishes might be an example. If we know runner a finishes at noon and runner B at 1PM then we know that runner B took longer. Since we haven't specified when they started, we don't know what percent longer though.

Definition 2.4. Ratio
Ration data has a meaningful 0. If we record not the time of finishing, but the time since starting, then 0 has a meaning and we can take a ration of the total time for runner A and B to compare the two.

However, working with data on a computer is different, requiring different categories…

Definition 2.5. Factor
When we look at many variables, some may simply record categories used to group the data. In R we will use *factors* to store these variables. An example might be the browser a user has used to view a website, as gleaned from a log.

Definition 2.6. character
Some categorical data are factors, but others are really just identifiers, and are not used for grouping. An example might be a user's IP address. Difference can be thought of as distinguishing between *categorizing* a case or *characterizing* a case. While both factor and categorical data are *nominal*, we keep the distinction as we will interact with the data differently.

Definition 2.7. discrete
Discrete data comes from measurements where there are essentially only distinct and separate possible values that can be counted. For example, the number of visits a person makes to a website will always be integer data, as will other counting data.

Definition 2.8. continuous
Data which could conceivably come from a continuum of variables. The recording of time in milliseconds of a visit to a website might be such data. A useful distinction is that for discrete data we expect that cases will share values, whereas for continuous data this will be impossible, or at least very unlikely. We can also turn continuous data into discrete data by truncating (record the minute instead of the millisecond) or by binning.Rather than draw distinctions between ordinal, interval, and ratio, it is more important for statistical theory - in finding a model for the recorded data - to know if the data is discrete or continuous.

Definition 2.9. time and date
Though we just saw that time and date can be considered continuous or discrete, for computers there are often separate ways to handle date and time data. Issues that complicate matters are leap days and time zones, but also scale (some people want millisecond data)

Definition 2.10. hierarchical
while much data is several measurements for several cases and fits nicely onto a rectangular spreadsheet, data on networks does not fit this

2.1 Data Vectors

Suppose the number of whale beachings in Texas during the 1990s was

`74 122 235 111 292 111 211 133 156 79`

We can combine these into a data set through

```
whale <- c(74, 122, 235, 111, 292, 111, 211, 133, 156, 79)
```

The `whale` object is a *data vector*.

the size of the data set is retreived with the `length` function

```
length(whale)
```

```
## [1] 10
```

```
sum(whale)
```

```
## [1] 1524
```

Average can be found with combining the two…

```
sum(whale)/length(whale)
```

```
## [1] 152.4
```

or

```
mean(whale)
```

```
## [1] 152.4
```

2.1.1 Vectorization

The arithmetic operations and the mathematical functions are vectorized, in that they will be called for each element in a data vector.

```
whale - mean(whale)
```

```
##  [1] -78.4 -30.4  82.6 -41.4 139.6 -41.4  58.6 -19.4   3.6 -73.4
```

```
whale^2 / length(whale)
```

```
##  [1]  547.6 1488.4 5522.5 1232.1 8526.4 1232.1 4452.1 1768.9 2433.6  624.1
```

```
sqrt(whale)
```

```
##  [1]  8.602325 11.045361 15.329710 10.535654 17.088007 10.535654 14.525839
##  [8] 11.532563 12.489996  8.888194
```

2.1.2 Missing values

```
hipcost <- c(10500, 45000, 74100, NA, 83500, 86000, 38200, NA, 44300, 12500, 55700, 439
```

NA is interpreted as a missing value, but which may have meaning, so it is not 0

```
sum(hipcost)
```

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

- leads to NA
- solution is to use na.rm

```
sum(hipcost, na.rm = TRUE)
```

```
## [1] 627600
```

```
mean(hipcost, na.rm = TRUE)
```

```
## [1] 52300
```

- multivariate datasets have more options related to NA values

2.1.3 Attributes: names

```
head(precip)
```

```
##      Mobile      Juneau     Phoenix Little Rock Los Angeles  Sacramento
##        67.0        54.7         7.0        48.5        14.0        17.2
```

```
head(sort(precip, decreasing=TRUE))
```

```
##       Mobile        Miami     San Juan  New Orleans       Juneau Jacksonville
##         67.0         59.8         59.2         56.8         54.7         54.5
```

```
head(names(precip))
```

```
## [1] "Mobile"      "Juneau"      "Phoenix"     "Little Rock" "Los Angeles"
## [6] "Sacramento"
```

2.1.4 Structured Data

```
1:length(whale)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
0:length(whale)-1
```

```
##  [1] -1  0  1  2  3  4  5  6  7  8  9
```

```
0:(length(whale)-1)
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

…

2.2 Numeric summaries

2.2.1 Center

- most common - mean, median, mode

```
wts <- kid.weights
```

```
sort(wts$weight, decreasing = TRUE)
```

```
##   [1] 150 150 144 131 125 108 105 100  98  94  93  90  89  87  86  85  85  80
##  [19]  80  80  78  76  74  72  70  70  69  69  65  65  65  64  61  60  60  60
##  [37]  59  58  55  55  55  54  53  52  52  52  52  52  50  50  50  50  50  50
##  [55]  49  48  47  47  47  47  46  46  45  45  45  45  45  45  45  45  44  43
##  [73]  43  43  42  42  42  42  42  41  41  41  40  40  40  40  40  40  40  40
##  [91]  40  40  40  38  38  38  38  38  38  37  37  36  36  35  35  35  35  35
## [109]  35  34  34  34  34  34  33  33  32  32  32  32  32  32  32  32  32  32
## [127]  31  31  31  30  30  30  30  30  30  30  30  30  30  30  30  30  30  29
## [145]  29  29  29  29  28  28  28  28  28  28  28  28  27  27  27  27  27  27
## [163]  26  26  26  26  26  26  26  26  25  25  25  25  25  25  24  24  24  23
## [181]  23  23  23  23  23  22  22  22  22  22  22  21  21  21  20  20  20  20
## [199]  20  20  19  19  19  19  19  19  18  18  18  18  18  18  18  17  17  17
## [217]  17  17  16  16  16  16  16  15  15  15  14  14  14  14  14  14  14  14
## [235]  14  13  13  13  13  13  13  13  13  12  12  12  11  11  11  10
```

- sample mean known as `x bar` or $\bar{x}$

```
mean(wts$weight)
```

```
## [1] 38.384
```

2.2.2 Spread

- variability of the data

2.2.2.1 `sample variance`

$$s^2 = \frac{1}{n-1}\sum_i (x_i - \bar{x})^2$$

```
var(wts$weight)
```

```
## [1] 615.3781
```

2.2.2.2 Sample `standard deviation`

$$\sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_i (x_i - \bar{x})^2}$$

```
sd(hipcost, na.rm=TRUE)
```

```
## [1] 24848.85
```

2.2.2.3 `z-score`

- deviations, $d_i = x_i - \bar{x}$, express data relative to its centre, rather than absolute
  $$z - score = \frac{x_i - \bar{x}}{s}$$
  gives the size of the data point in terms of its relative position to centre on a scale of standard deviations, so z-score of 3 means the data point is 3sd larger than mean

2.2.2.4 defining function

```r
z_score <- function(x)(x-mean(x))/sd(x)
```

```r
z_score(wts$weight)
```

```
##   [1] -0.01547962  1.95978406  0.46825843  2.40321060  0.34732391 -0.33797165
##   [7] -0.57984067  0.26670091  4.25753976 -0.57984067  0.06514339 -0.94264420
##  [13] -0.74108668 -1.14420172 -0.78139819 -0.37828315  1.67760353 -0.74108668
##  [19] -0.41859465 -0.57984067 -0.13641413 -0.33797165  0.54888143  1.35511150
##  [25] -0.41859465 -0.98295570 -0.82170969 -1.10389021 -0.29766014 -0.41859465
##  [31] -0.29766014 -0.98295570 -0.01547962 -0.94264420 -0.45890616  0.87137346
##  [37]  0.06514339  0.06514339 -0.70077518  0.34732391  0.30701241 -0.33797165
##  [43]  1.23417699  1.51635752 -0.90233270 -1.10389021 -0.01547962 -0.53952916
##  [49] -0.37828315 -0.70077518 -0.09610262 -0.41859465  0.10545489 -1.02326721
##  [55] -0.70077518 -0.21703714  0.06514339 -0.25734864 -0.45890616 -0.74108668
##  [61] -0.33797165  0.26670091 -0.17672563 -0.78139819  0.10545489  0.06514339
##  [67] -0.78139819 -0.98295570  0.18607790  2.08071857  0.26670091 -0.37828315
##  [73]  2.68539112  2.24196458 -0.66046367  1.23417699 -0.86202119 -0.45890616
##  [79] -1.06357871  0.06514339 -0.53952916 -1.02326721 -0.74108668 -0.90233270
##  [85]  0.26670091 -0.45890616 -0.74108668  0.62950444  0.87137346 -0.33797165
##  [91] -0.82170969  0.66981594 -0.98295570 -0.17672563  0.26670091  0.38763542
##  [97] -0.33797165 -0.53952916 -0.01547962 -1.02326721 -0.66046367 -0.78139819
## [103]  1.43573451 -0.33797165  2.04040706 -0.13641413 -0.21703714  1.59698053
## [109]  0.18607790  0.06514339 -0.25734864 -0.29766014 -0.94264420 -0.78139819
## [115]  0.66981594  0.06514339 -0.05579112 -0.17672563  0.26670091  0.22638940
## [121]  0.46825843  0.46825843 -0.41859465 -0.25734864 -0.66046367  0.54888143
## [127]  0.06514339  0.14576640 -0.66046367 -0.62015217 -0.90233270 -0.86202119
## [133] -0.33797165  0.06514339 -0.25734864 -0.49921766 -0.33797165  0.91168496
## [139]  1.07293098  0.54888143 -0.62015217 -0.37828315 -0.82170969 -0.25734864
## [145]  0.46825843 -1.02326721 -0.09610262  0.34732391  1.67760353 -0.62015217
## [151]  1.03261948  0.26670091 -0.53952916  1.07293098 -0.49921766 -0.01547962
## [157] -0.74108668  4.49940878  0.14576640  0.58919294 -0.13641413 -0.90233270
## [163]  4.49940878 -0.53952916 -0.33797165  0.54888143 -0.45890616 -0.45890616
## [169] -0.66046367 -0.33797165 -0.13641413  0.14576640 -1.06357871  0.34732391
## [175] -0.98295570 -0.25734864 -0.53952916  2.20165308 -0.33797165  0.83106196
## [181]  2.48383360 -0.17672563  0.54888143  0.18607790 -0.25734864 -1.02326721
## [187] -0.82170969 -0.01547962 -0.86202119  0.30701241 -0.37828315  0.46825843
## [193] -0.98295570 -1.02326721 -0.49921766 -1.10389021 -0.25734864  1.27448850
## [199] -0.90233270  0.46825843 -0.49921766 -0.62015217 -0.05579112  0.87137346
## [205]  0.26670091 -0.49921766  3.73349022 -0.66046367 -1.06357871  1.27448850
## [211]  1.91947255 -0.13641413 -0.98295570  1.87916105  0.79075045 -0.41859465
## [217] -0.86202119 -0.17672563 -0.78139819 -1.02326721 -0.82170969 -0.62015217
## [223] -0.86202119 -0.33797165  0.10545489 -0.33797165 -1.02326721  3.49162119
## [229] -0.25734864 -0.82170969 -0.13641413 -0.25734864  2.80632563  0.06514339
## [235]  0.14576640 -0.41859465 -0.49921766 -0.49921766 -0.62015217  0.14576640
```

```
## [241] -0.41859465  0.66981594  1.07293098  1.67760353  0.42794692 -0.82170969
## [247] -0.98295570 -0.49921766  1.87916105 -0.98295570
```

Example

Prof scales on z-scores and those who have z-score value of greater than 1.28, get an A

```
x <- c(54, 50, 79, 79, 51, 69, 55, 62, 100, 80)
z <- (x-mean(x))/sd(x)
x[z >= 1.28]
```

```
## [1] 100
```

what score is just good enough for an A?

```
mean(x) + 1.28 * sd(x)
```

```
## [1] 88.91046
```

- formula reverses the z-score formula is read as the score which is 1.28 SD above the mean
- z-score allow datasets with different scales to be compared

2.2.3 Shape (distribution)

- normal

2.3 categorical data

```
x <- babies$smoke
x <- factor(x, labels=c("never", "now", "until current", "once, quit", "unknown" ))
table(x)
```

```
## x
##         never           now until current    once, quit       unknown
##           544           484            95           103            10
```

```
out <- table(x)
prop <- 100 * out / sum(out)
round(prop, digits = 2)
```

```
## x
##         never           now until current    once, quit       unknown
##         44.01         39.16          7.69          8.33          0.81
```

Chapter 3

Bivariate Data

```
names(fat)
```

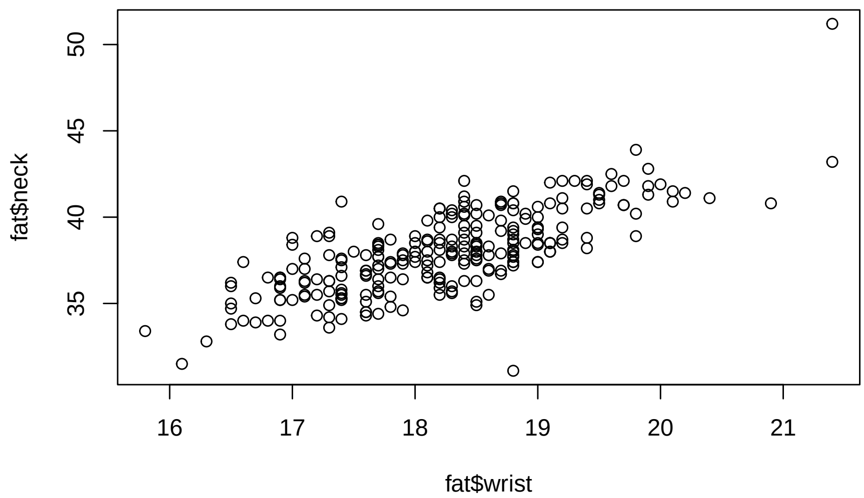
```
##  [1] "case"          "body.fat"      "body.fat.siri" "density"
##  [5] "age"           "weight"        "height"        "BMI"
##  [9] "ffweight"      "neck"          "chest"         "abdomen"
## [13] "hip"           "thigh"         "knee"          "ankle"
## [17] "bicep"         "forearm"       "wrist"
```

3.1 Correlation

p.106

- `correlation` is a numeric summary of how closely related are the measures of two numeric variables when they are in a linear relationship.
- perfect correlation would mean data on a straight line;
- no correlation means values are scattered

```
plot(fat$wrist, fat$neck)
```

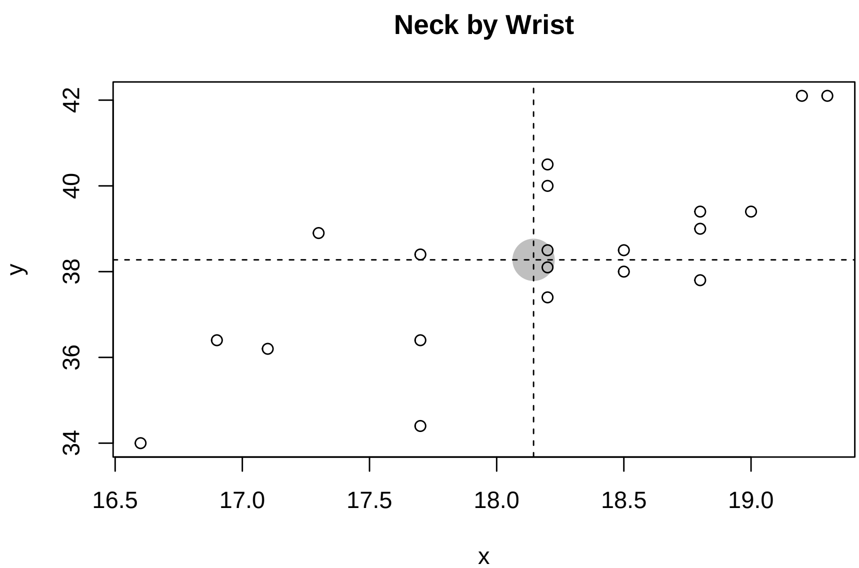


or

```
plot(neck ~ wrist, data = fat)
```

To investigate `correlation`, shift centre of scatter plot to the mean of `x` variable and the mean of `y` variable and add `abline` and `point` functions - plot below separates the data into 4 quadrants as determined by points that are above and below the mean of all `x` values and those above and below the mean of all `y` values

```
x <- fat$wrist[1:20]; y <- fat$neck[1:20]                    # data
plot(x, y, main = "Neck by Wrist")
abline(v = mean(x), lty=2)                                   # dashed vertical line
abline(h = mean(y), lty=2)                                   # dashed horizontal lin
points(mean(x), mean(y), pch=16, cex=4, col=rgb(0,0,0, .25))
```



- correlated data shows in opposite regions (bottom left and top right)

3.1.1 covariance

$$cov(x, y) = \frac{1}{n-1} \sum (x_i - \bar{x})(y_i - \bar{y})$$

3.1.2 Pearson correlation coefficient

- uses `z-score` instead of deviations…

$$cor(x, y) = \frac{1}{n-1} \sum \frac{(x_i - \bar{x})}{s_x} \frac{(y_i - \bar{y})}{s_y} = cov(x, y)/(s_x s_y)$$

- formula is symmetric, so `x` and `y` can have order switched
- using `z-score` removes the effect of spread and centre by standardizing

3.1.2.1 `cov` and `cor` functions

```
cor(fat$wrist, fat$neck)    # strongly correlated
```

```
## [1] 0.7448264
```

```
cor(fat$wrist, fat$height)  # mildly correlated
```

```
## [1] 0.3220653
```

```
cor(fat$age, fat$ankle)     # basically uncorrelated
```

```
## [1] -0.1050581
```

3.1.3 Spearman correlation coefficient

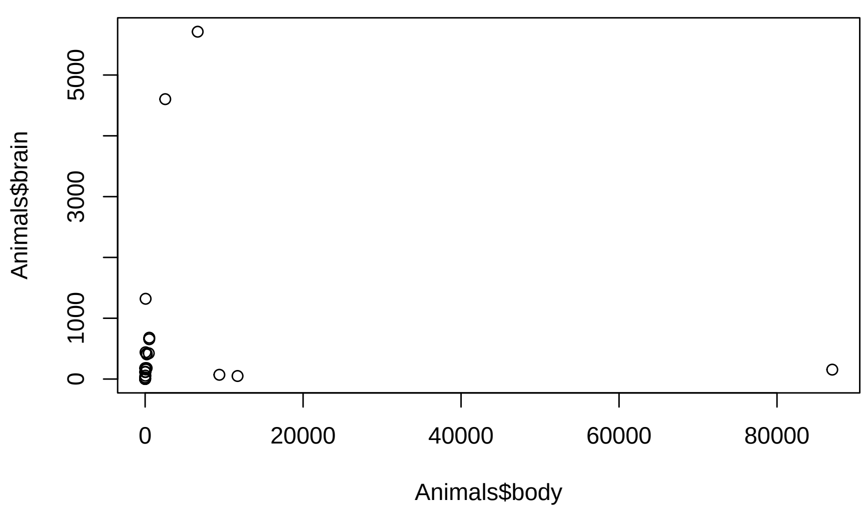
- `Pearson` measures the strength of linear ralationship

```
cor(Animals$body, Animals$brain)
```

```
## [1] -0.005341163
```

- very low correlation…

```
plot(Animals$body, Animals$brain)
```



```
body <- Animals$body; brain <- Animals$brain
cross_prods <- (body - mean(body)) * (brain - mean(brain))
Animals[cross_prods < 0, ]
```

```
##                  body  brain
## Dipliodocus     11700   50.0
## Asian elephant   2547 4603.0
## Horse             521  655.0
## Giraffe           529  680.0
## Human              62 1320.0
## Triceratops      9400   70.0
## Brachiosaurus   87000  154.5
```

- brachiosaurus is a huge outlier…
- so, rank the data

```
cor(rank(body), rank(brain))
```

```
## [1] 0.7162994
```

or

```
cor(body, brain, method="spearman")
```

```
## [1] 0.7162994
```

3.1.3.1 correlation with replication

```
cor(ToothGrowth$dose, ToothGrowth$len)
```

```
## [1] 0.8026913
```

- positive correlation
- for each dosage there are several experimental units -split data into 3 groups and then compute the correlation for these dosage values and the group averages

```
l <- split(ToothGrowth$len, ToothGrowth$dose)
group_means <- c(mean(l[[1]]), mean(l[[2]]), mean(l[[3]]))
cor(c(0.5, 1, 2), group_means)
```

```
## [1] 0.9574428
```

- 0.95 for aggregated data is higher than 0.8 for individual data. In general, correlations formed from averages are typically closer to 1 or -1.

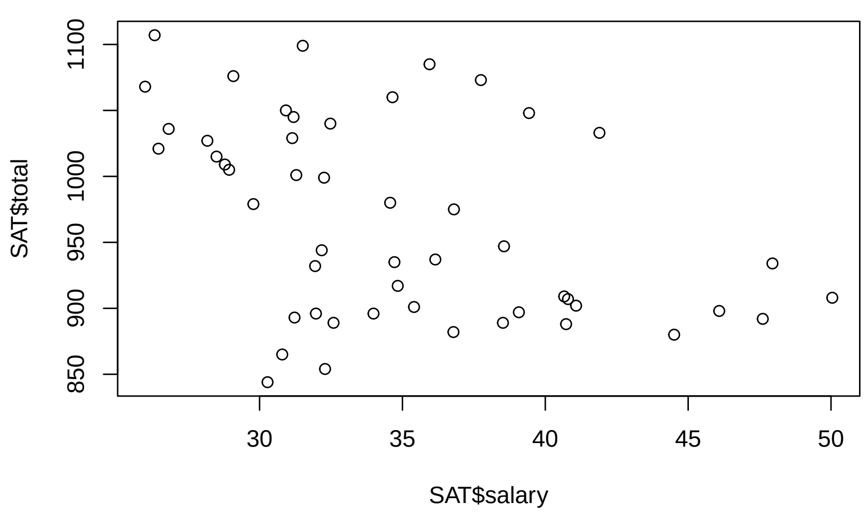
3.1.3.2 correlation v causation

- what is the relationship between average teacher pay and SAT scores?

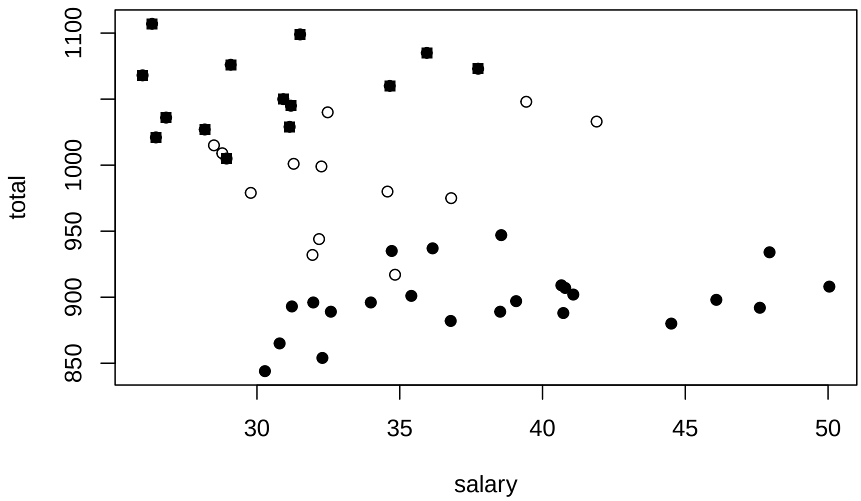
```
cor(SAT$salary, SAT$total)
```

```
## [1] -0.4398834
```

```
plot(SAT$salary, SAT$total)
```




```
plot(total~salary, SAT)
points(total~salary, SAT, subset = perc < 10, pch=15)  # square
points(total ~ salary, SAT, subset = perc > 40, pch=16) #solid
```



- shows correlation for each subgroup is positive

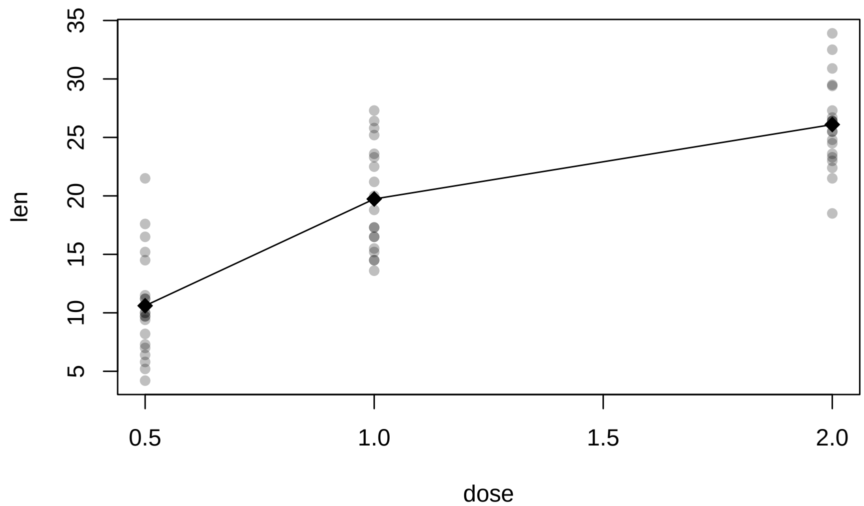
```
total <- SAT$total; salary <- SAT$salary; perc <- SAT$perc
less_10 <- perc < 10
more_40 <- perc > 40
between <- !less_10 & !more_40
c(less = cor(total[less_10], salary[less_10]),
  between = cor(total[between], salary[between]),
  more = cor(total[more_40], salary[more_40]))
```

```
##      less   between      more
## 0.2588199 0.2224926 0.3673461
```

- all correlations are positive for subgroups, yet overall correlation is negative, called `Simpson's paradox` where some trend that exists for subgroups disappears when data are aggregated

3.2 Trends

```
plot(len ~ dose, data=ToothGrowth, pch=16, col=rgb(0,0,0, .25))
points(c(0.5, 1, 2), group_means, cex=1.5, pch=18)
lines(c(0.5, 1, 2), group_means)
```



- summarize a relationship between two numeric variables
- model for a linear trend can be specified as follows: > The mean response value depends linearly on the predictor value.

$$\mu_{y|x} = \beta_0 + \beta_1 x$$

where $\mu_{y|x}$ means the mean of the response variable for a specified value of the predictor x

- for individual data points this becomes

  $$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

  where ϵ_i are the error terms
- we make assumptions about ϵ_i, that on average, the values of ϵ_i are 0

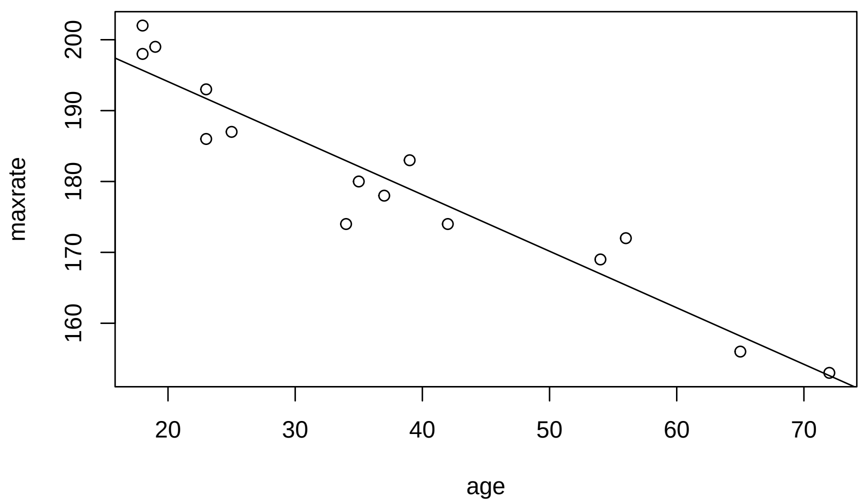
3.2.1 `lm` function

```
res <- lm(maxrate ~ age, data=heartrate)
res
```

```
##
## Call:
## lm(formula = maxrate ~ age, data = heartrate)
##
## Coefficients:
## (Intercept)          age
##    210.0485      -0.7977
```

Visualizing the regression line

```
plot(maxrate ~ age, data=heartrate)
abline(res)
```



Chapter 4

Footnotes and citations

4.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

4.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [**?**] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 5

Blocks

5.1 Equations

Here is an equation.

$$f\left(k\right) = \binom{n}{k} p^k \left(1 - p\right)^{n-k} \tag{5.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (5.1).

5.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 5.1.

Theorem 5.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

5.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 6

Sharing your book

6.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book-all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.