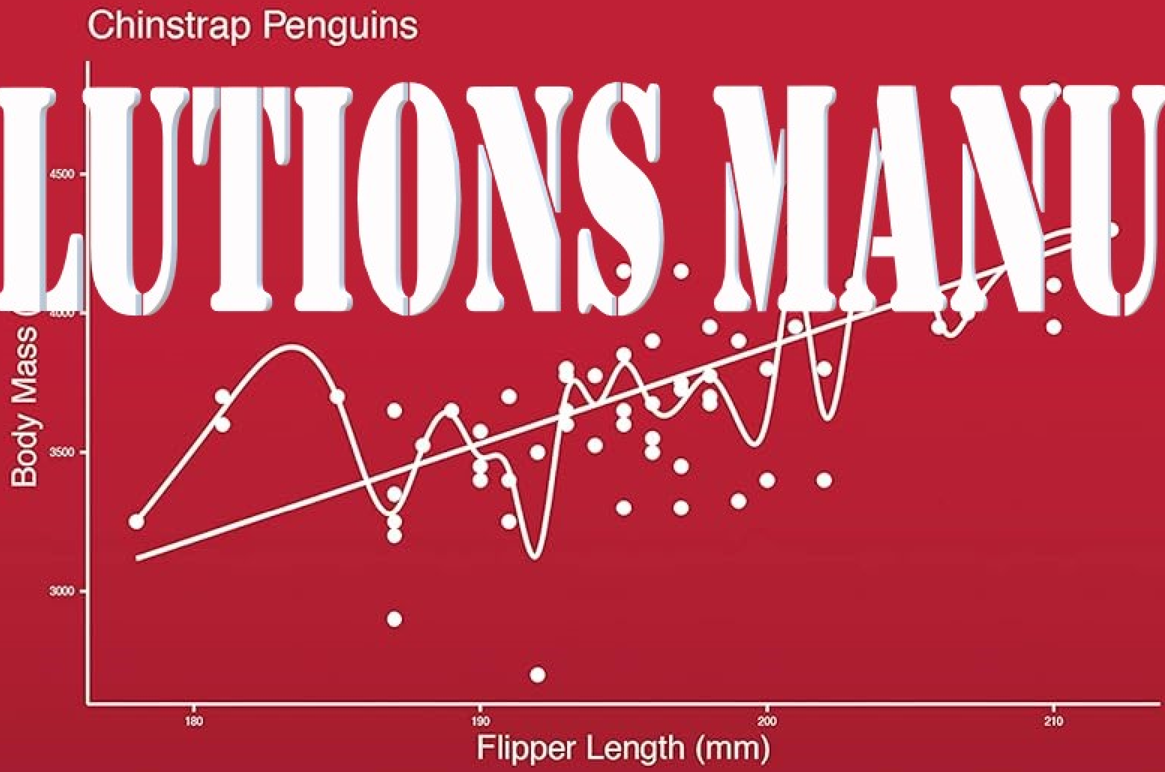
Texts in Statistical Science

Probability, Statistics, and Data

A Fresh Approach Using R



Darrin Speegle

Bryan Clair

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

1

Data in R – Solutions

Problem 1 a. 2 4 6; b. 6 10 12; c. 5

Problem 2 a. 1 2 3 (result not stored into `x`). b. 2 4 6 c. 1 2 3 (result not stored into `x`) d. 2 3 4

Problem 3

```
1:10
```

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

```
1:10 * 2
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
1:10^2
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```
1:10 + 1
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11
```

```
1:(10 * 2)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
rep(c(1,1,2), times = 2)
```

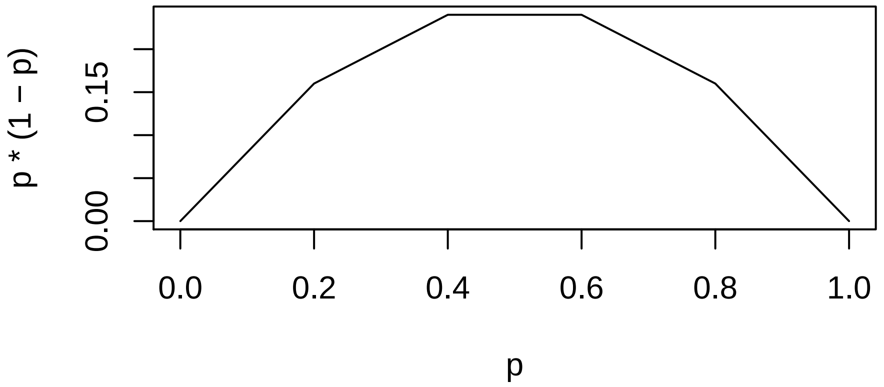
```
## [1] 1 1 2 1 1 2
```

```
seq(from = 0, to = 10, length.out = 5)
```

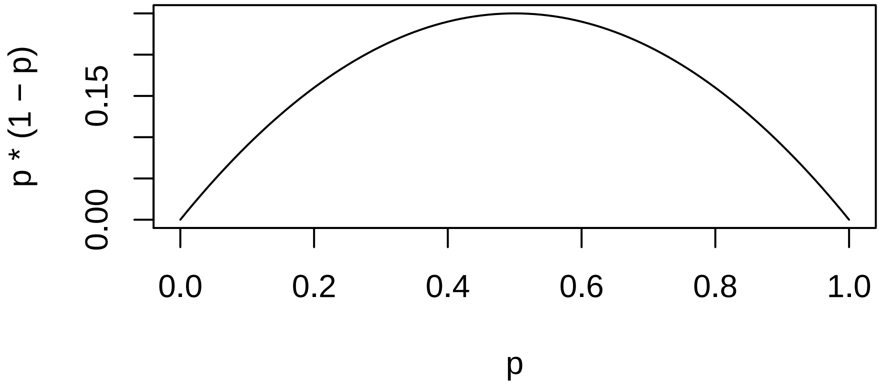
```
## [1]  0.0  2.5  5.0  7.5 10.0
```

Problem 4

```
p <- seq(0,1,0.2)
plot(p,p*(1-p),type='l')
```



```
p <- seq(0,1,0.01)
plot(p,p*(1-p),type='l')
```



Problem 5

```
sum((1:100)^2)
```

```
## [1] 338350
```

Problem 6

```
x <- seq(from = 10, to = 30, by = 2)
length(x)
```

```
## [1] 11
```

```
x[2]
```

```
## [1] 12
```

```
x[1:5]
```

```
## [1] 10 12 14 16 18
```

```
x[1:3*2]
```

```
## [1] 12 16 20
```

```
x[1:(3*2)]
```

```
## [1] 10 12 14 16 18 20
```

```
x > 25
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
```

```
x[x > 25]
```

```
## [1] 26 28 30
```

```
x[-1]
```

```
##  [1] 12 14 16 18 20 22 24 26 28 30
```

```
x[-1:-3]
```

```
## [1] 16 18 20 22 24 26 28 30
```

Problem 7

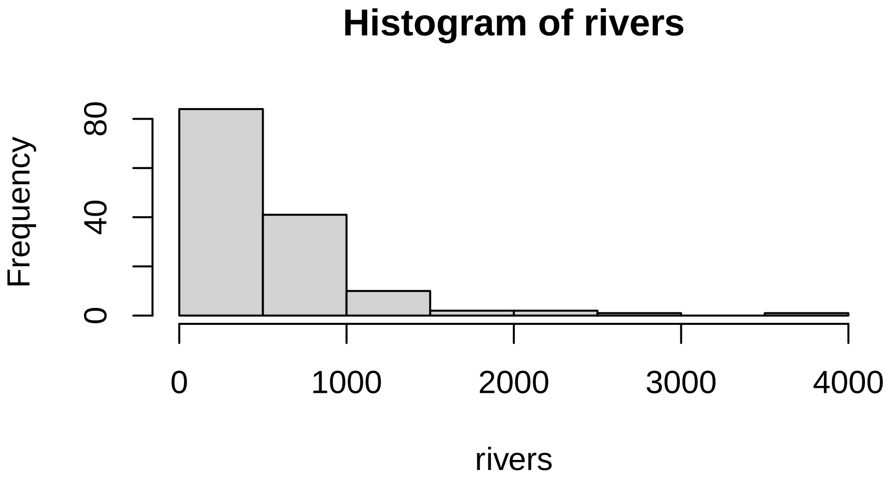
```
mean(rivers)
```

```
## [1] 591.1844
```

```
sd(rivers)
```

```
## [1] 493.8708
```

```
hist(rivers)
```



```
summary(rivers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   135.0   310.0   425.0   591.2   680.0  3710.0
```

```
max(rivers)
```

```
## [1] 3710
```

```
min(rivers)
```

```
## [1] 135
```

```
rivers[rivers > 1000]
```

```
##  [1] 1459 1450 1243 2348 1171 3710 2315 2533 1306 1054 1270 1885 1100 1205 1038
## [16] 1770
```

Problem 8

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

153 observations of 6 variables, named `Ozone`, `Solar.R`, `Wind`, `Temp`, `Month`, `Day`. They are all type `int` except for Wind, which is type `num`. Ozone, Solar.R, and Temp would be more natural as `num` types. Month and Day could be stored as factor types.

Problem 9 a.

```
table(state.region)
```

```
## state.region
##     Northeast         South North Central          West
##             9            16            12            13
```

b.

```
state.name[state.area < 10000]
```

```
## [1] "Connecticut"   "Delaware"      "Hawaii"        "Massachusetts"
## [5] "New Hampshire" "New Jersey"    "Rhode Island"  "Vermont"
```

c.

```
state.name[which.min(state.center$y)]
```

```
## [1] "Florida"
```

Problem 10

```
row.names(mtcars)[mtcars$gear == 4]
```

```
##  [1] "Mazda RX4"      "Mazda RX4 Wag"  "Datsun 710"     "Merc 240D"
##  [5] "Merc 230"       "Merc 280"       "Merc 280C"      "Fiat 128"
##  [9] "Honda Civic"    "Toyota Corolla" "Fiat X1-9"      "Volvo 142E"
```

```
row.names(mtcars)[mtcars$gear == 4 & mtcars$am == 1]
```

```
## [1] "Mazda RX4"      "Mazda RX4 Wag"  "Datsun 710"     "Fiat 128"
## [5] "Honda Civic"    "Toyota Corolla" "Fiat X1-9"      "Volvo 142E"
```

```
row.names(mtcars)[mtcars$gear == 4 | mtcars$am == 1]
```

```
##  [1] "Mazda RX4"      "Mazda RX4 Wag"  "Datsun 710"     "Merc 240D"
##  [5] "Merc 230"       "Merc 280"       "Merc 280C"      "Fiat 128"
##  [9] "Honda Civic"    "Toyota Corolla" "Fiat X1-9"      "Porsche 914-2"
## [13] "Lotus Europa"   "Ford Pantera L" "Ferrari Dino"   "Maserati Bora"
## [17] "Volvo 142E"
```

```
mean(mtcars$mpg[mtcars$carb == 2])
```

```
## [1] 22.4
```

Problem 11

```
mtcars$am <- factor(mtcars$am, levels = c(0, 1), labels = c("auto", "manual"))
table(mtcars$am)
```

```
##
##   auto manual
##     19     13
```

```
table(mtcars$am[mtcars$mpg > 25])
```

```
##
##   auto manual
##      0      6
```

Problem 12 a. 54 observations of 3 variables; b. Beef, Meat, Poultry; c. 645;

d. 190

```
max(hot_dogs[hot_dogs$type == "Beef","calories"])
```

```
## [1] 190
```

Problem 13

a. 70 obs. of 6 variables. `class` and `trade` are type Factor. `sober`, `drinks`, and `n` are type int. `wage` is type num.
b. Profession #23, factory worker, is the lowest paid.
c. 604 workers were surveyed
d. The average wage of all 604 workers was 24.60 shillings/week.

```
library(HistData)
which.min(DrinksWages$wage)
```

```
## [1] 23
```

```
DrinksWages[23,]
```

```
##    class          trade sober drinks wage n
## 23     A factory worker     1      3   12 4
```

```
sum(DrinksWages$n)
```

```
## [1] 604
```

```
sum(DrinksWages$wage * DrinksWages$n)/604
```

```
## [1] 24.59782
```

Problem 14

```
library(Lahman)
```

a. 102816 obs. of 22 variables

c,d,e with base R:

```
max(Batting$X3B)
```

```
## [1] 36
```

```
Batting[which.max(Batting$X3B),c("playerID","yearID","X3B")]
```

```
##        playerID yearID X3B
## 13805 wilsoch01   1912  36
```

```
recent<-Batting[Batting$yearID >= 1960,c("playerID","yearID","X3B")]
recent[which.max(recent$X3B),]
```

```
##        playerID yearID X3B
## 89181 grandcu01   2007  23
```

wilsoch01 hit 36 triples in 1912. Since 1960, It's grandcu01 (Curtis Granderson) with 23 in 2007.

Alternately, c,d,e with dplyr library:

```
head(arrange(Batting,desc(X3B)),1)
```

```
##    playerID yearID stint teamID lgID   G  AB  R   H X2B X3B HR RBI SB CS BB SO
## 1 wilsoch01   1912     1    PIT   NL 152 583 80 175  19  36 11  95 16 NA 35 67
##   IBB HBP SH SF GIDP
## 1  NA   2 23 NA   NA
```

```
recent <- filter(Batting, yearID > 1960)
head(arrange(recent,desc(X3B)),1)
```

```
##    playerID yearID stint teamID lgID   G  AB   R   H X2B X3B HR RBI SB CS BB
## 1 grandcu01   2007     1    DET   AL 158 612 122 185  38  23 23  74 26  1 52
##    SO IBB HBP SH SF GIDP
## 1 141   3   5  5  2    3
```

Problem 15

a. 803 pass
b. 44.76% pass
c. table(bechdel$year)
d. 2010; which.max(table(bechdel$year))
e. 5
f. bechdel[bechdel$binary == "PASS",]
g. bechdel[!is.na(bechdel$domgross),]

2

Probability – Solutions

Problem 1 The probability that one die is twice the other is 1/6.

Problem 2 a. $P(\text{difference} = 0) = 1/6$ b. $P(\text{difference} = 4) = 1/9$

Problem 3

a. The sample space consists of the following 15 ordered pairs: (1,2), (1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6), (5,6).
b. $E = \{(1, 4), (2, 3)\}$.
c. $P(E) = 2/15$.
d. $P(F) = 0$.

Problem 4

a. About 11 percent.

```
mean(replicate(10000, {
  sum(sample(1:8, 2, TRUE)) == 8
}))
```

```
## [1] 0.1092
```

b. About 23 percent.

```
mean(replicate(10000, {
  any(sample(1:8, 2, TRUE) == 2)
}))
```

```
## [1] 0.2282
```

Problem 5 a. 0.80; b. 0.47; c. 0.61; d. 0.013

```
mmdist <- c(0.14, 0.13, 0.20, 0.12, 0.20, 0.21)
mmcolors <- c('Yel','Red','Org','Brn','Grn','Blu')
mean(replicate(10000,any(sample(mmcolors,4,replace=TRUE,prob=mmdist)=='Blu')))
```

```
## [1] 0.6049
```

```
mean(replicate(10000,anyDuplicated(sample(mmcolors,6,replace=TRUE,prob=mmdist)) == 0))
```

```
## [1] 0.0135
```

Problem 6 About 0.065.

```
mean(replicate(10000,{
  bag <- sample(mmcolors,30,replace=TRUE,prob=mmdist);
  blueCount <- sum(bag == 'Blu');
  orangeCount <- sum(bag == 'Org');
  (blueCount >= 9) & (orangeCount >= 6)
}))
```

```
## [1] 0.0647
```

Problem 7 About 38%.

$$\begin{aligned} P(\text{same}) &= P(OO) + P(AA) + P(BB) + P(ABAB) \\ &= 0.45^2 + 0.40^2 + 0.11^2 + 0.04^2 = 0.3762. \end{aligned}$$

Or simulate:

```
types <- c("O", "A", "B", "AB")
p <- c(0.45, 0.40, 0.11, 0.04)
mean(replicate(10000,sample(types,1,prob=p) == sample(types,1,prob=p)))
```

```
## [1] 0.3813
```

Problem 8

```
mean(replicate(10000,sum(sample(1:6,2,replace=TRUE)) == 10))
```

```
## [1] 0.0878
```

```
3/36
```

```
## [1] 0.08333333
```

Problem 9

```
mean(replicate(10000,sum(sample(0:1,7,replace=TRUE)) == 3))
```

```
## [1] 0.2719
```

```
choose(7,3)/128
```

```
## [1] 0.2734375
```

Problem 10 About 0.56.

```
mean(replicate(10000,{roll <- sum(sample(1:6,5,replace=TRUE)); roll >= 15 & roll <= 20}))
```

```
## [1] 0.5519
```

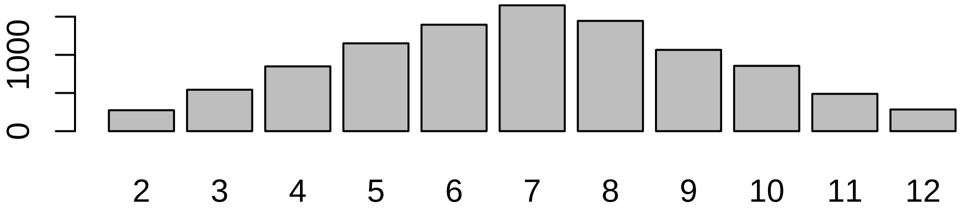
Problem 11

```
mean(replicate(10000, {dieRoll <- sample(1:6, 20, replace = TRUE); 20 %in% cumsum(dieRoll)}))
```

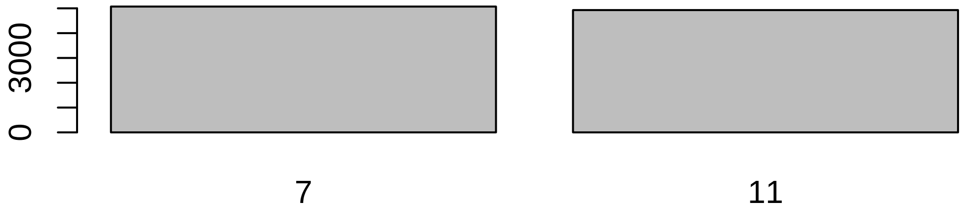
```
## [1] 0.289
```

Problem 12

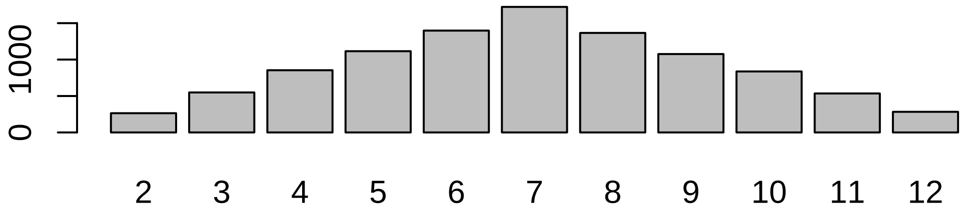
```
rolls <- replicate(10000,sum(sample(1:6,2,replace=TRUE)))
barplot(table(rolls))
```



```
rolls <- replicate(10000,5+sample(c(2,6),1))
barplot(table(rolls))
```



```
rolls <- replicate(10000,sample(c(1,2,2,3,3,4),1)+sample(c(1,3,4,5,6,8),1))
barplot(table(rolls))
```



A sum of two Sicherman dice has the same probability distribution as a sum of two normal dice.

Problem 13 If you happen to be born on day 255, like I was:

```
mean(replicate(10000,any(sample(1:365,199,replace=TRUE)==255)))
```

```
## [1] 0.4201
```

```
1-(364/365)^199
```

```
## [1] 0.420711
```

Problem 14

```
num_hours <- 24 * 365
sim_data <- replicate(10000, {
  birth_hours <- sample(1:num_hours, 100, T)
  anyDuplicated(birth_hours) > 0
})
mean(sim_data)
```

```
## [1] 0.4296
```

The probability is about 43 percent!

Problem 15 The probability is approximately 0.12.

```
most_with_same_birthday <- replicate(10000,max(table(sample(1:365,50,replace=TRUE))))
mean(most_with_same_birthday >= 3)
```

```
## [1] 0.121
```

Problem 16

```
sum(replicate(100000,length(unique(sample(1:20,100,replace=TRUE))) < 20))/100000
```

```
## [1] 0.11546
```

Gives (with rounding) 0.113 consistently

Problem 17 a. 0.048; b. 0.060

```
deck <- rep(c(2:10,10,10,10,11),4)
deals <- replicate(100000, sum(sample(deck,2)))
mean(deals == 21)
```

```
## [1] 0.04828
```

```
mean(deals == 19)
```

```
## [1] 0.06084
```

Problem 18 Between .04 and .05. About .047.

```
pp <- replicate(10000, {
  x1 <- sample(1:1000, 1)
  x2 <- sample(1:x1, 1)
  x3 <- sample(1:x2, 1)
  x4 <- sample(1:x3, 1)
  return(x4 == 1 && x3 > 1)
})
mean(pp)
```

```
## [1] 0.0451
```

Problem 19 About .019.

```
vowels <- c("A","E","I","O","U")
mean(replicate(100000,{
       hand <- sample(fosdata::scrabble$piece,7);
       length(intersect(hand,vowels)) == 0
}))
```

```
## [1] 0.01895
```

Problem 20 a. $\frac{1}{12} \approx 0.083$ b. $\frac{1}{6} \approx 0.167$ c. $\frac{1}{2}$

Problem 21 a. $\frac{1}{15} \approx 0.067$ b. $\frac{2}{15} \approx 0.133$ c. $\frac{1}{2}$

Problem 22 Disjoint: AD, CD; Independent: AB, BD

Problem 23 a. A and C are disjoint. b. B and C are independent. c. $P(B|A) = 1$.

Problem 24 Since B is contained in $A \cup B$,

$$P(A \cup B|B) = \frac{P((A \cup B) \cap B)}{P(B)} = \frac{P(B)}{P(B)} = 1$$

Problem 25 #Solution Needed

Problem 26 $P(A) = 1/6$, $P(B) \approx .42$ and $P(A \cap B) \approx .11$. A and B are not independent since $P(A \cap B) \neq P(A)P(B)$.

```
prob_a <- mean(replicate(10000, {
  die_roll <- sample(1:6, 3, T)
  die_roll[1] >= max(die_roll[2:3])
}))
prob_a
```

```
## [1] 0.4237
```

```
prob_ab <- mean(replicate(10000, {
  die_roll <- sample(1:6, 3, T)
  die_roll[1] >= max(die_roll[2:3]) && die_roll[1] == 5
}))
prob_ab
```

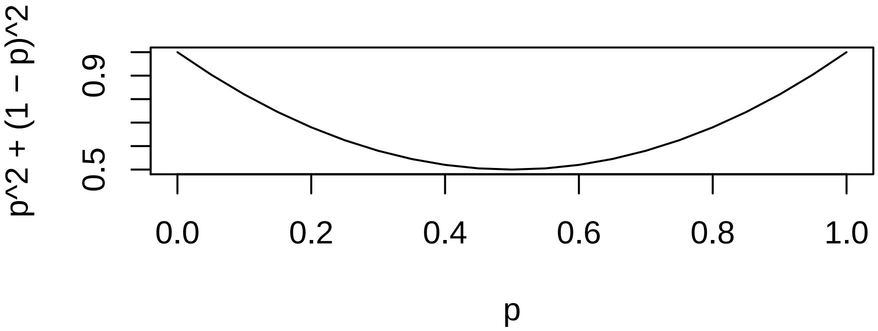
```
## [1] 0.1144
```

Problem 27 The probability is $\frac{p(1-p)}{2p(1-p)} = \frac{1}{2}$.

Problem 28 a. 0.934; b. 0.063.

Problem 29

```
p <- seq(0,1,.05); plot(p,p^2 + (1-p)^2, type='l')
```



a. 1/2; b. $p^2 + (1-p)^2$; d. $P(\text{same}) = 0.505$, $P(\text{diff}) = 0.495$. e. Call same. f. Yes, closer to even.

Problem 30 There is only a 40.5% chance a person testing positive has the disease.

Let D be the event "has disease" and $+$ be the event "tests positive". Given $P(+|D) = 0.999$, $P(-|\text{not } D) = 0.97$ and $P(D) = 0.02$. First, $P(+|\text{not } D) = 0.03$ so the law of total probability gives

$$P(+) = 0.999 \cdot 0.02 + 0.03 \cdot 0.98 = 0.04938$$

Now Bayes' rule gives

$$P(D|+) = P(+|D) \cdot \frac{P(D)}{P(+)} = 0.999 \cdot \frac{0.02}{0.04938} \approx 0.405$$

Problem 31 Let R be the event "red marble" and B_1 and B_2 be the events corresponding to the boxes.

a. With the law of total probability:

$$P(R) = P(R|B_1)P(B_1) + P(R|B_2)P(B_2) = \frac{3}{7}\frac{1}{3} + \frac{2}{7}\frac{2}{3} = \frac{1}{3}$$

b. Using Bayes' rule:

$$P(B_1|R) = P(R|B_1)\frac{P(B_1)}{P(R)} = \frac{3}{7}\frac{1/3}{1/3} = \frac{3}{7} \approx 0.249$$

Problem 32 a. About 84%.

```r
mean(replicate(20000, {
  box <- sample(0:9, 1);
  marblesInBox <- c(rep("R", box), rep("B", 9 - box));
  marblesDrawn <- sample(marblesInBox, 3, TRUE);
  if (all(marblesDrawn == "R")) {
    sample(marblesInBox, 1) == "R"
  } else NA
  }), na.rm = TRUE)
```

```
## [1] 0.83779
```

Here is another approach. Let B be the event "three red marbles drawn" and A the event "four red marbles drawn". Estimate $P(B)$ and $P(A \cap B)$ to get $P(A|B)$. This is somewhat less accurate for the same number of trials.

```r
probB <- mean(replicate(10000, {
  box <- sample(0:9, 1);
  marblesInBox <- c(rep("R", box), rep("B", 9 - box));
  marblesDrawn <- sample(marblesInBox, 3, TRUE);
  all(marblesDrawn == "R")
  }))  #Prob of B

probAB <- mean(replicate(10000, {
  box <- sample(0:9, 1);
  marblesInBox <- c(rep("R", box), rep("B", 9 - box));
  marblesDrawn <- sample(marblesInBox, 4, TRUE);
  all(marblesDrawn == "R")
  } )) #Prob of A cap B

probAB/probB
```

```
## [1] 0.8216944
```

b. About 36%.

```r
mean(replicate(20000, {
  box <- sample(0:9, 1);
  marblesInBox <- c(rep("R", box), rep("B", 9 - box));
  marblesDrawn <- sample(marblesInBox, 3, TRUE);
  if (all(marblesDrawn == "R")) {
    box == 9  # TRUE if box 9, FALSE otherwise
  } else NA   # don't count this trial
  }), na.rm = TRUE)
```

```
## [1] 0.3653223
```

Problem 33 $\binom{10}{4} = 210$.

Problem 34 $\binom{9}{3} = 84$.

Problem 35

a. $6^6 = 46656$.
b. $6! = 720$.
c. $720/46656 = 5/324 \approx 0.015$.

Problem 36

a. $\binom{10}{6} = 210.$
b. $\binom{5}{4}\binom{5}{2} = 50.$
c. $50/210 \approx 0.238$

3

Discrete Random Variables – Solutions

Problem 1 a. $1/4 + 1/2 + 1/8 + 1/8 = 1$. b. $P(X \geq 2) = 1/4$. c. $P(X \geq 2|X \geq 1) = 1/3$. d. $P(X \geq 2 \cup X \geq 1) = P(X \geq 1) = 3/4$

Problem 2

```
X <- sample(0:3, 10000, prob=c(1/4,1/2,1/8,1/8), replace=TRUE)
mean(X == 1)     # part a
```

```
## [1] 0.4947
```

```
table(X)/10000   # part b
```

```
## X
##      0      1      2      3
## 0.2497 0.4947 0.1322 0.1234
```

Problem 3 Need $1 = \sum_x p(x) = C/4 + C/2 + C = 7C/4$ so $C = 4/7$.

Problem 4 For example, $p(x) = 1/2$ when $x = \pm 1$ and $p(x) = 0$ otherwise.

Problem 5 $E[X] = 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = \frac{9}{8} = 1.125$

Problem 6 $E[X] = 1 \cdot \frac{1}{10} + \cdots + 10 \cdot \frac{1}{10} = \frac{1}{10}(1 + 2 + \cdots + 10) = \frac{55}{10} = \frac{11}{2} = 5.5$

```
X <- sample(1:10, 10000, replace=TRUE)
mean(X)
```

```
## [1] 5.5062
```

Problem 7

Exactly:

```
products <- sort(rep(1:6,6)) * (1:6)  # clever way to get all 36 products
sum(products)/36
```

```
## [1] 12.25
```

Or by simulation:

```
mean( replicate(10000, prod(sample(1:6,2,replace=TRUE))))
```

```
## [1] 12.1701
```

Problem 8 If X is the product, $X = 2$, $X = 3$, and $X = 6$ are all equally likely. Then $E(X) = \frac{1}{3}2 + \frac{1}{3}3 + \frac{1}{3}6 = \frac{11}{3}$

Problem 9 The expected number of digits is $1 \cdot \frac{10}{1000} + 2 \cdot \frac{90}{1000} + 3 \cdot \frac{900}{1000} = 2.89$.

Problem 10

a. X can take two values, 1 and 3. $p(1) = .95^2 = 0.9025$, $p(3) = 1 - .95^2 = 0.0975$.
b. $E[X] = 1 \cdot 0.9025 + 3 \cdot 0.0975 = 1.195$.

c.Using simulations:

```
npool <- 90  # number of people in the pool
sim_data <- replicate(10000, {
  tests <- sample(c("pos", "neg"), npool, TRUE, prob = c(5, 95))
  ifelse(any(tests == "pos"), 1+npool, 1)
})
table(sim_data)/10000
```

```
## sim_data
##    1   91
## 0.01 0.99
```

```
mean(sim_data)
```

```
## [1] 90.1
```

d. Pooled testing uses less tests. In this setting pooled testing uses less tests with the pool size 87 or less, although the advantage diminishes after a pool size of 18.

Problem 11 a. Expected value of \$1 bet on red is $1 * (18/38) + (-1) * (20/38) = -2/38 \approx -0.053$. b. Expected value of let it ride, once, is $3 * (18/38)^2 + (-1) * (1 - (18/38)^2) = -0.102$.

Problem 12 a. Bet doubling, you can lose 6 in a row. Seven losses in a row and you are out of money. b. The probability you do not win \$1 is $(20/38)^7 = 0.0112$. The probability you win \$1 is $1 - 0.0112 = 0.9888$. You are about 99% likely to win a dollar with bet doubling! c. Expected value is $1 * 0.9888 + (-127) * 0.0112 = -0.432$. This is not good - you expect to lose, on average, 43 cents every time you play this strategy. d. Let X be the number of wins before you go broke. $X \sim \text{Geom}(0.0112)$, so the expected value is $E(X) = (1-p)/p = 0.9888/0.0112 \approx 88.4$ plays before you go broke.

Problem 13 This code uses some tools that are a little advanced for this chapter.

```
library(plyr)
suppressMessages(suppressWarnings(library(dplyr)))

n <- 10  #flips to sample
k <- 1   #count flips after k-in-a-row successes
trials <- rdply(10000, {
  shots <- sample(0:1,n,replace=TRUE)
  after <- shots[diff(c(rep(0,k+1),cumsum(shots)),k) == k]
  table(factor(after, levels=0:1), useNA="no")
})
trials <- mutate(trials, hit = `1`, miss = `0`, count = hit + miss, p = hit/count) %>%
  select(hit,miss,count,p)
mean(trials$p,na.rm=TRUE)
```

```
## [1] 0.4425184
```

Problem 14

a. \$1 straight: 0.60
b. \$1 front pair: 0.60
c. \$1 back pair: 0.60
d. \$6 6-way: 3.60 (since the bet costs \$6, this is a return of 0.60 on the dollar)
e. \$3 3-way: 1.80 (since the bet costs \$3, this is a return of 0.60 on the dollar)
f. \$1 1-off: $(1 \times 300 + 6 \times 29 + 12 \times 4 + 8 \times 9)/1000 = 0.594$ (this is the worst bet, you don't get your sixty cents on the dollar)

Problem 15 $E[X] = \frac{1}{k}(1 + 2 + \cdots + k) = \frac{k+1}{2}$.

Problem 16 Expected score is 5. Probability of 10 or more correct is:

```
sum(dbinom(10:20,20,.25))
```

```
## [1] 0.01386442
```

Problem 17 Expected number is 9.1 shots. Probability of at least 8 made is:

```
sum(dbinom(8:10,10,.91))
```

```
## [1] 0.94596
```

Problem 18

```
dgeom(19,.09)
```

```
## [1] 0.01499785
```

Problem 19

```
0.047*262  #a
```

```
## [1] 12.314
```

```
1 - pbinom(41,262,0.047) #b
```

```
## [1] 4.627632e-12
```

```
0.5*305 #c
```

```
## [1] 152.5
```

```
1 - pbinom(210,305,0.5) #d
```

```
## [1] 8.791412e-12
```

It seems likely that Dream was cheating.

Problem 20

Let X be a geometric rv with success probability p. Let $q = 1 - p$ be the failure probability. Begin with a geometric series in q:

$$\sum_{x=0}^{\infty} q^x = \frac{1}{1-q}$$

Take the derivative of both sides with respect to q:

$$\sum_{x=0}^{\infty} xq^{x-1} = \frac{1}{(1-q)^2}$$

and a second time:

$$\sum_{x=0}^{\infty} x(x-1)q^{x-2} = \frac{2}{(1-q)^3}$$

Multiply both sides by pq^2:

$$\sum_{x=0}^{\infty} x(x-1)pq^x = \frac{2pq^2}{(1-q)^3}$$

Replace $1-q$ with p and we have shown:

$$E[X(X-1)] = \frac{2q^2}{p^2}$$

Now $E[X] = q/p$ so

$$E[X^2] = E[X(X-1)] + E[X] = \frac{2q^2}{p^2} + \frac{qp}{p^2} = \frac{2q^2+qp}{p^2}$$

Finally,

$$\text{var}(X) = E[X^2] - E[X]^2 = \frac{2q^2+qp}{p^2} - \frac{q^2}{p^2} = \frac{q^2+qp}{p^2} = \frac{q(q+p)}{p^2} = \frac{q}{p^2}.$$

The standard deviation is then

$$\sigma(X) = \frac{\sqrt{1-p}}{p}$$

Problem 21

$$p(y) = \begin{cases} 1/4 & y=-1 \\ 1/2 & y=0 \\ 1/8 & y=1 \\ 1/8 & y=2 \end{cases}$$

$$p(u) = \begin{cases} 1/4 & u=0 \\ 1/2 & u=1 \\ 1/8 & u=4 \\ 1/8 & u=9 \end{cases}$$

$$p(v) = \begin{cases} 1/2 & v=0 \\ 3/8 & v=1 \\ 1/8 & v=2 \end{cases}$$

Problem 22 a. 23; b. -5.

Problem 23

```
pmf <- c(1/4, 1/2, 1/8, 1/8)
mu <- sum((0:3)*pmf)  # mu = 9/8
v <- sum( ((0:3) - mu)^2 * pmf ) # v is the variance
```

The variance is 0.859375, the sd is 0.9270248.

Problem 24

```
# a
pmf <- c(1:6,5:1)/36

# b
mu <- sum(2:12 * pmf)
mu  # mean
```

```
## [1] 7
```

```
sd <- sqrt(sum((2:12)^2 * pmf) - mu^2) # sd
sd
```

```
## [1] 2.415229
```

```
# c
sd^2
```

```
## [1] 5.833333
```

```
2*(35/12)
```

```
## [1] 5.833333
```

Problem 25

a. $9 = \text{Var}(X) = E[X^2] - 4$ so $E[X^2] = 13$.
b. $E[(2X-1)^2] = E[4X^2 - 4X + 1] = 4(13) - 4(2) + 1 = 45$
c. $\text{Var}(2X-1) = E[(2X-1)^2] - E[2X-1]^2 = 45 - 9 - 36$.

Problem 26

$$
\begin{aligned}
E[(X-\mu_X)(Y-\mu_Y)] &= E[XY - X\mu_Y - Y\mu_X + \mu_X\mu_Y] = \\
&= E[XY] - E[X\mu_Y] - E[Y\mu_X] + \mu_X\mu_Y \\
&= E[XY] - \mu_X\mu_Y - \mu_X\mu_Y + \mu_X\mu_Y \\
&= E[XY] - E[X]E[Y] = \text{cov}(X,Y)
\end{aligned}
$$

Problem 27

a. $\text{Var}(X) = 100(.2)(.8) = 16$ and $\text{Var}(Y) = 40 * .5 * .5 = 10$.
b. The value should be 26.

```
X <- rbinom(100000,100,.2)
Y <- rbinom(100000,40,.5)
var(X + Y)
```

```
## [1] 25.93344
```

Problem 28 There are four outcomes of the experiment, HH, HT, TH, TT. All three variables have probability 1/2 of being either 0 or 1. Every pair of values for two of the variables occurs on exactly one outcome of the experiment:

- (X,Y) are (0,0) when the outcome is TT, (0,1) when TH, (1,0) when HT, and (1,1) when HH.
- (X,Z) are (0,0) when the outcome is TH, (0,1) when TT, (1,0) when HT, and (1,1) when HH.
- (Y,Z) are (0,0) when the outcome is HT, (0,1) when TT, (1,0) when TH, and (1,1) when HH.

Since each pair has probability 1/4 and $1/4 = 1/2 \cdot 1/2$, each pair of variables is independent.

However, $P(X=0, Y=0, Z=0) = 0$ since it's impossible for the first and second toss to both be tails while also demanding that the tosses don't match. Since $P(X=0)P(Y=0)P(Z=0) = 1/8 \neq 0$, the variables are not mutually independent.

Problem 29

The mean is $E\left[\frac{X-\mu}{\sigma}\right] = \frac{1}{\sigma}(E[X]-\mu) = 0$. The sd is $\sigma\left(\frac{X-\mu}{\sigma}\right) = \frac{1}{\sigma}(\sigma(X-\mu)) = 1$.

Problem 30

a. Expect 110 to support.
b. $\sigma = \sqrt{200 * .55 * .45}/200 \approx 0.0352$. Margin of error is 7.04%.
c. Probability that poll claims Prop A will fail is 8.87%.

```
pbinom(100,200,.55)
```

```
## [1] 0.08870062
```

d. For 2% margin of error, solve $2\sqrt{.55 * .44/n} = 0.02$ to get $n = 2420$.

Problem 31 The mean and standard deviation are both 1.

```
correct <- replicate(10000,sum(sample(1:27) == 1:27))
mean(correct)
```

```
## [1] 1.0018
```

```
sd(correct)
```

```
## [1] 1.00881
```

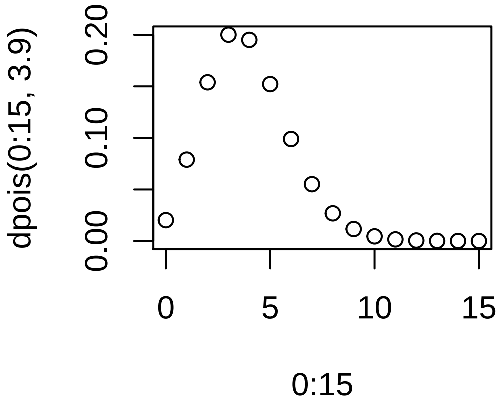
Here is an exact computation for the expected value. There are $n!$ possible draws from the hat. Person 1 gets their number correct $(n-1)!$ of these. The same goes for person 2, and so on. Then the total number of correct assignments across all $n!$ draws is $(n-1)! + \cdots + (n-1)! = n(n-1)! = n!$. So the average number of correct assignments is $n!/n! = 1$.

Problem 32

a. Let X_i be the participant's score on guess i. On the ith trial, the participant has 1/12 chance of scoring 1 and 2/12 chance of scoring 3/4. Their expected score for one guess is then $E[X_i] = 1/12 + 6/48 = 5/24 \approx 0.208$. The expected score on 36 guesses is $36 \cdot 5/24 = 15/2 = 7.5$.
b. For a single guess: $E[X_i^2] = 1 \cdot 1/12 + (3/4)^2 \cdot (2/12) = 17/96$. Then $\text{Var}(X_i) = 17/96 - (5/24)^2 = 77/576$. Since each trial is independent, the variance for the score after 36 guesses is 77/16 so the sd is $\sqrt{77}/4 \approx 2.19$.

Problem 33

```
plot(0:15,dpois(0:15,3.9))
```



b. 3 is most likely
c. $a = 3$.

d. $b = 2$.

Problem 34 The maximum difference is about 0.0014 when $x = 2$.

```
approx_err <- abs(dpois(0:200,2)-dbinom(0:200,200,2/200))
which.max(approx_err)-1
```

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

```
max(approx_err)
```

```
## [1] 0.001362443
```

Problem 35 $\frac{\text{var}(I)}{E[I]} = \frac{\text{var}(Ne)}{E[Ne]} = \frac{e^2\text{var}(N)}{eE[N]} = e\frac{\lambda}{\lambda} = e$.

Problem 36

a. $\sum_{x=0}^{\infty} \frac{\lambda^x}{x!} e^{-\lambda} = e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = e^{-\lambda} e^{\lambda} = 1$, since $\sum_{x=0}^{\infty} \frac{\lambda^x}{x!}$ is the Taylor series of e^{λ}.

b. $E[X] = \sum_{x=0}^{\infty} x \frac{\lambda^x}{x!} e^{-\lambda} = e^{-\lambda} \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-1)!} = \lambda e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = \lambda$.

c. $E[X(X-1)] = \sum_{x=0}^{\infty} x(x-1) \frac{\lambda^x}{x!} e^{-\lambda} = e^{-\lambda} \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-2)!} = \lambda^2 e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = \lambda^2$. Therefore, by $E[X^2] = \lambda^2 + \lambda$ and $\text{Var}(X) = E[X^2] - E[X]^2 = \lambda^2 + \lambda - \lambda^2 = \lambda$.

Problem 37

a. The number of failures before n successes is the number of failures before the first success plus the number of failures between first and second success, and so on.

b. From part a, $E[X] = \sum_{i=1}^{n} E[X_i] = n\frac{p}{1-p}$ and $\text{Var}(X) = \sum_{i=1}^{n} \text{Var}(X_i) = n\frac{p}{(1-p)^2}$.

Problem 38

$\text{Var}(X) = \lambda \leq \frac{\lambda}{1-p} = \frac{np}{(1-p)^2} = \text{Var}(Y)$.

Problem 39

```
scrabble <- fosdata::scrabble
vowels <- c('A','E','I','O','U')
num_vowels <- replicate(10000,{
  hand <- sample(scrabble$piece, 7)
  sum(hand %in% vowels)
})
mean(num_vowels == 7)
```

```
## [1] 0.0015
```

```
mean(num_vowels <= 2)
```

```
## [1] 0.3732
```

```
mean(num_vowels)
```

```
## [1] 2.9478
```

```
sd(num_vowels)
```

```
## [1] 1.274691
```

Problem 40 We can take a large sample of number of rolls with the following code.

```
sim_data <- replicate(10000, {
  cur_value <- 1000
  num_rolls <- 1
  while(cur_value != 1) {
    cur_value <- sample(1:cur_value, 1)
    num_rolls <- num_rolls + 1
  }
  num_rolls
})
```

a. The expected value is about 9.5

```
mean(sim_data)
```

```
## [1] 9.4748
```

b. Here is an approximate pmf.

```
table(sim_data)/10000
```

```
## sim_data
##      2      3      4      5      6      7      8      9     10     11     12
## 0.0011 0.0063 0.0224 0.0465 0.0780 0.1173 0.1306 0.1366 0.1243 0.1024 0.0798
##     13     14     15     16     17     18     19     20     21     22     23
## 0.0588 0.0369 0.0255 0.0137 0.0097 0.0047 0.0023 0.0014 0.0007 0.0007 0.0001
##     25
## 0.0002
```

c. The probability that player 1 wins is the probability that it takes an even number of rolls to get to 1, and is about 1/2.

```
mean(sim_data %% 2 == 0)
```

```
## [1] 0.4936
```

4

Continuous Random Variables – Solutions

Problem 1 a. 3/4; b. 4/5

Problem 2 $C = 3/2$.

Problem 3 For each of the following functions, decide whether the function is a valid pdf, a valid cdf or neither.

a. $h(x) = \begin{cases} 1 & 0 \le x \le 2 \\ -1 & 2 \le x \le 3 \\ 0 & \text{otherwise} \end{cases}$ Not pdf or cdf, has negative values.

b. $h(x) = \sin(x) + 1$ Not pdf, doesn't integrate to 1. Not cdf because not increasing.

c. $h(x) = \begin{cases} 1 - e^{-x^2} & x \ge 0 \\ 0 & x < 0 \end{cases}$ Not pdf, doesn't integrate to 1. Yes cdf. Increasing, and has correct limits.

d. $h(x) = \begin{cases} 2xe^{-x^2} & x \ge 0 \\ 0 & x < 0 \end{cases}$ Yes pdf. Nonnegative and integrates to 1. Not cdf because not increasing.

Problem 4 One possible answer: $f(x) = 2x$ for $0 \le x \le 1$ and $f(x) = 0$ otherwise. It is not possible for $f(x) > 1$ for all x.

Problem 5 It is not possible. $\lim_{x\to\infty} f(x) = 1$ if f is a cdf but the limit is 0 for a pdf.

Problem 6

a. $\int_0^1 3(1-x)^2 dx = 1$, and $f(x) \ge 0$ for all x.

b. $\mu(X) = \int_0^1 x \cdot 3(1-x)^2 dx = 1/4$.
$var(X) = \int_0^1 x^2 \cdot 3(1-x)^2 dx - (1/4)^2 = 0.0375$.
$\sigma(X) \approx 0.194$.

c. 7/8

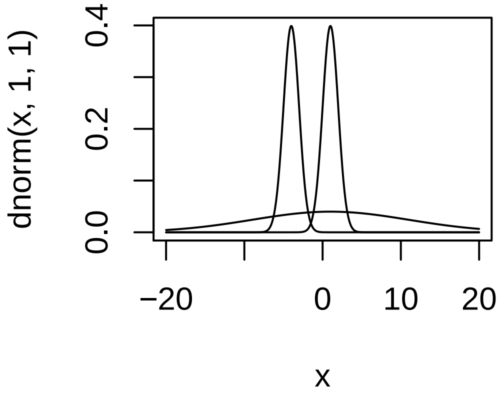
d. $(19/64)/(27/64) = 19/27 \approx .704$.

Problem 7 $\text{Var}(2X + 1) = 4\text{Var}(X) = 12$.

Problem 8 a. 1; b. 2; c. $a = 0$; d. A little bit less than 1/2.

Problem 9

```
x<-seq(-20,20,.01)
plot(x,dnorm(x,1,1), type="l")
lines(x,dnorm(x,1,10), type="l")
lines(x,dnorm(x,-4,1), type="l")
```



Problem 10

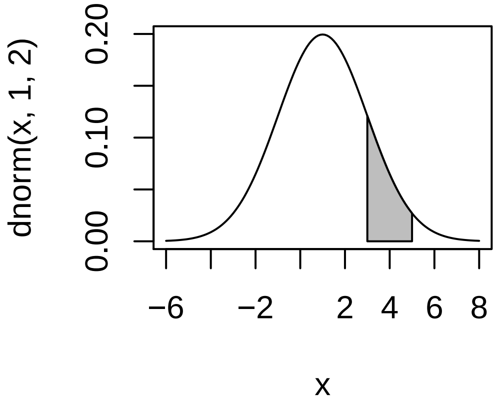
a.

```
pnorm(5,1,2)-pnorm(3,1,2)
```

```
## [1] 0.1359051
```

b.

```
x <- seq(-6,8,.1)
plot(x,dnorm(x,1,2),type="l")
x <- seq(3,5,.1)
polygon(c(3,x,5),c(0,dnorm(x,1,2),0),col="grey")  # tricky - could just shade by hand
```



c. Largest when $a = 0$.

Problem 11

```
pnorm(85, 80, 5, lower.tail = FALSE)
```

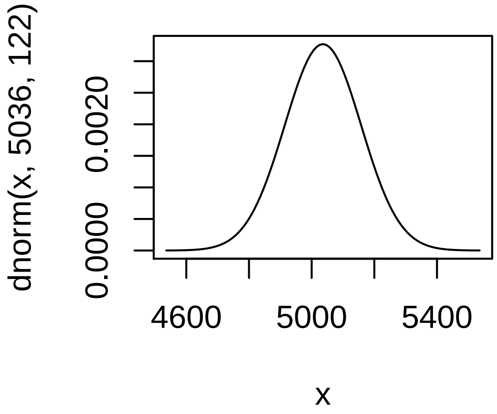
```
## [1] 0.1586553
```

```
1 - pbinom(3, 10, 0.1586553)
```

```
## [1] 0.05974509
```

Problem 12

```
x <- seq(4536,5536,10); plot(x,dnorm(x,5036,122), type="l")
```



b. About 0.384.

c. 5236.7 lbs.

Problem 13

```
X <- rnorm(100000,0,2)
Y <- rnorm(100000,0,1)
var(X+3*Y)
```

```
## [1] 13.12761
```

This is close to $\text{Var}(X) + 9\text{Var}(Y) = 13$.

Problem 14

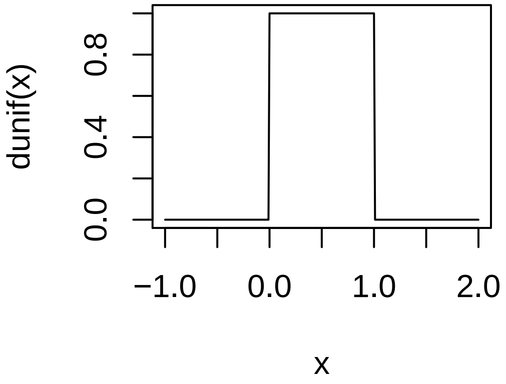
$$\left(\int_{-\infty}^{\infty} e^{-x^2} dx\right)^2 = \int_{-\infty}^{\infty} e^{-x^2}\,dx \int_{-\infty}^{\infty} e^{-y^2}\,dy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(x^2+y^2)}\,dx\,dy$$

$$= \int_0^{2\pi}\int_0^{\infty} e^{-r^2} r\ dr\ d\theta = \int_0^{2\pi} -\frac{1}{2}e^{-r^2}\Big|_{r=0}^{\infty}\ d\theta = \int_0^{2\pi}\frac{1}{2}d\theta = \pi$$

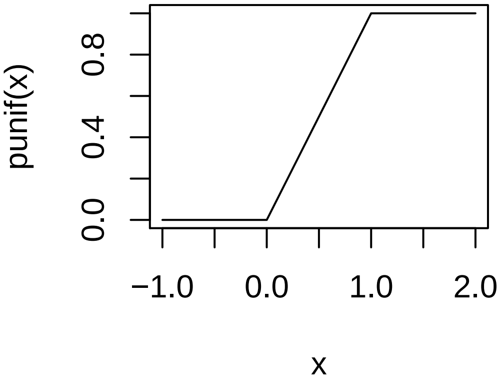
So $\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$.

Problem 15

```
x <- seq(-1,2,.01)
plot(x,dunif(x),type="l")
```

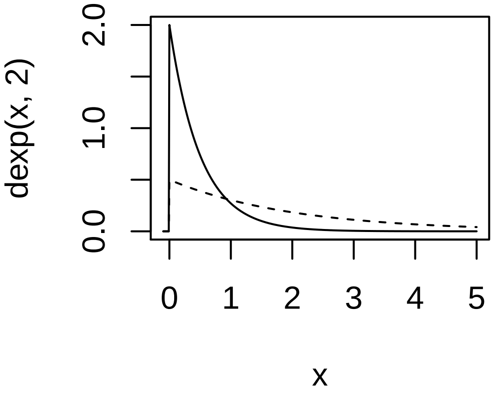


```
plot(x,punif(x),type="l")
```



Problem 16

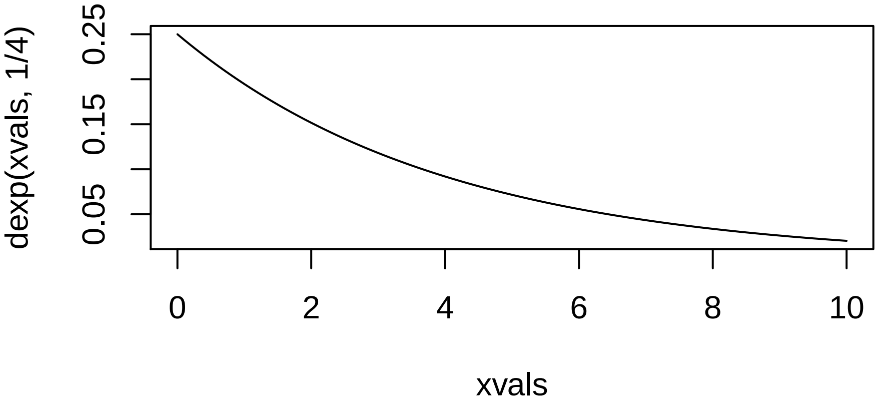
```
x<-seq(-.1,5,.01)
plot(x,dexp(x,2), type="l")
lines(x,dexp(x,1/2), type="l", lty=2) # dashed line
```



Problem 17

a. 4
b. $P(a \leq X \leq a+1)$ is maximized when $a = 0$. The mean is not in $[0, 1]$.

```
xvals <- seq(0,10,.1)
plot(xvals,dexp(xvals, 1/4),type='l')
```



Problem 18 The probability mom gets a text on any given day is 1/6. The probability she gets a text on 3 consecutive days is $(1/6)^3 = 1/216 \approx 0.46\%$

Problem 19

```
component1 <- rexp(10000,1/5)
component2 <- rexp(10000,1/5)
system.fail <- (component1 < 10) & (component2 < 10)
mean(system.fail)
```

```
## [1] 0.7462
```

Alternately, $P(\text{system fail}) = P(c_1 < 10 \cap c_2 < 10) = P(c_1 < 10) \cdot P(c_2 < 10)$. So:

```
pexp(10,1/5)^2
```

```
## [1] 0.7476451
```

Problem 20 Let $X \sim \text{unif}(a, b)$. $E[X] = (a + b)/2$. Then

$$\text{var}(X) = \int_a^b (x - \frac{b-a}{2})^2 dx = \frac{(b-a)^2}{12}.$$

Problem 21 a. The statement says that the probability of waiting a additional time, given that you've already waited b, is the same as the probability of waiting a from the beginning. In other words, the fact that you've waited b doesn't help: the variable has no "memory" of how long you have waited so far.

b. $P(X > a) = 1 - \int_0^a \lambda e^{-\lambda x} dx = e^{-a\lambda}$.

c.

$$P(X > a+b \,|\, X > b) = \frac{P(X > a+b \;\cap\; X > b)}{P(X > b)} = \frac{P(X > a+b)}{P(X > b)} = = \frac{e^{-(a+b)\lambda}}{e^{-b\lambda}} = e^{-a\lambda} = P(X > a).$$

Problem 22

Part a: Y is binomial. $P(Y = 3) \approx 0.155$. Expect $10/6 \approx 1.67$ sixes. $Var(Y) = 50/36 \approx 1.39$.

Part b: U is poisson. $P(U = 2) \approx 0.27$, $E(U) = 2$, $Var(U) = 2$.

Part c: X is uniform. The mean of X is 30 seconds.

Part d: X is exponential. The mean of X is 1/5 hour or 12 minutes. 10 minutes is 1/6 hour, and $P(X \leq 1/6) = 0.565$.

Part e: X is geometric. The mean of X is 1. $P(X \leq 3) \approx 0.94$.

Part f: X is normal. The mean of X is well known to be $98.6°F$. The standard deviation is probably around $1°F$, since $100°F$ is considered a fever.

Problem 23 X is uniform with mean 15 minutes.

Problem 24

$$F(z) = P(Z \leq z) = P\big((X_1 \leq z) \cap (X_2 \leq z) \cap (X_3 \leq z)\big) = P(X_1 \leq z) \cdot P(X_2 \leq z) \cdot P(X_3 \leq z)$$

For $z \in [0, 1]$, $P(X_i \leq z) = z$, so $F(z) = \begin{cases} z^3 & z \in [0,1] \\ 0 & \text{otherwise} \end{cases}$.

Problem 25

```
pexp(4, 1/4)
```

```
## [1] 0.6321206
```

Sixty-three percent change the meeting will start in 4 or fewer minutes.

b.

```
pexp(5, 1/4, lower.tail = FALSE)
```

```
## [1] 0.2865048
```

Twenty-nine percent chance the meeting will start in exactly five minutes.

Problem 26

```
n <- 4
mean(replicate(10000, {
  centers <- sort(runif(n))
  all(diff(centers) <= .5) && (centers[n] - centers[1] > .5)
}))
```

```
## [1] 0.5021
```

The probability that 4 intervals of length 1/2 placed randomly inside of [0, 1] union to form an interval of length of at least 1 is 1/2.

Problem 27

```
sim_data <- replicate(10000, {
  location <- runif(1)
  angle <- runif(1, 0, pi)
  location + abs(1/4 * sin(angle)) > 1 || location - abs(1/4 * sin(angle)) < 0
})
abs(1/mean(sim_data) - pi)
```

```
## [1] 0.07598873
```

Problem 28 Test if sides of length a, b, c make a triangle by checking if the longest edge c is shorter than the sum of the two shorter edges $a + b$. Since our stick has length 1, $a + b + c = 1$ this means that $c < a + b \iff c < 1 - c$ or $c < 1/2$.

```
mean(replicate(10000,{
  breaks <- c(0,sort(runif(2)),1)
  lengths <- diff(breaks)
  longedge <- max(lengths)
  longedge < 1/2
}))
```

```
## [1] 0.2512
```

The exact answer is 1/4.

5

Simulation of Random Variables – Solutions

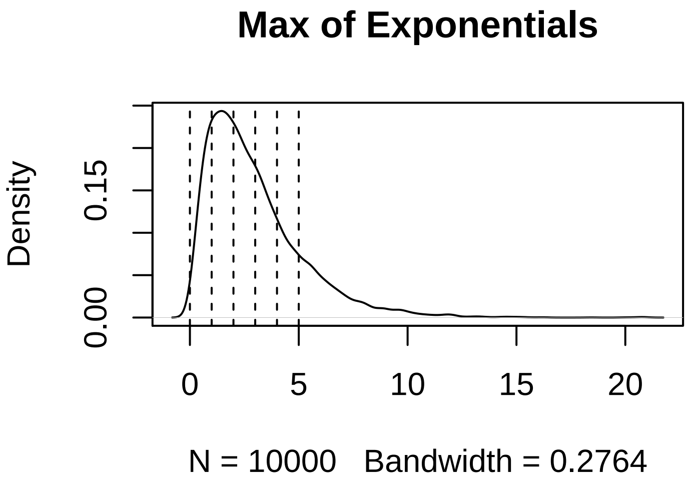
Problem 1

```r
sol <- mean(rnorm(10000)^2 < 2)
```

About 0.838, compare to `pchisq(2, df = 1)`.

Problem 2

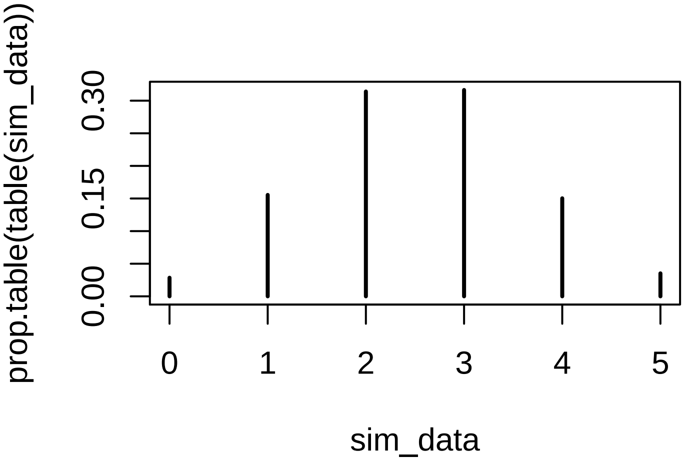
```r
sim_data <- replicate(10000, {
  x <- rexp(1, rate = 1/2)
  y <- rexp(1, rate = 1/2)
  max(x, y)
})
plot(density(sim_data),
     main = "Max of Exponentials")
abline(v = c(0:5), lty = 2)
```



The max area is about when $a = 1$.

Problem 3

```r
sim_data <- replicate(10000, {
  coins <- sample(c("H", "T"), 5, T)
  sum(coins == "H")
})
plot(prop.table(table(sim_data)))
```

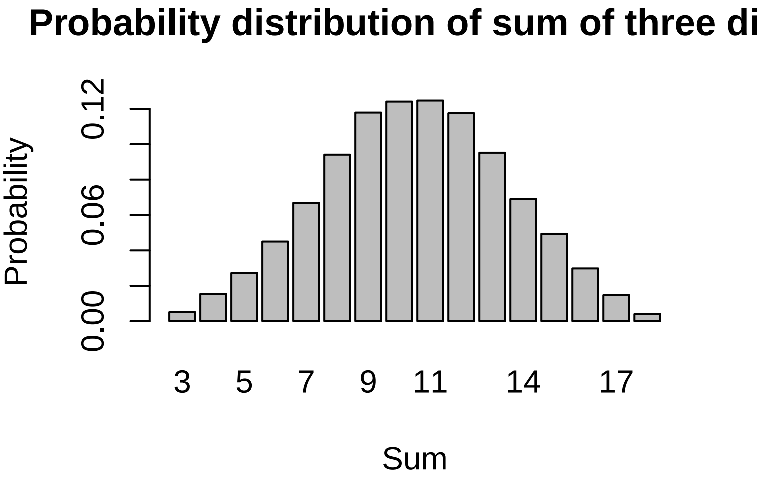


Problem 4

```
sim_data <- replicate(10000, {
  dice <- sample(1:6, 3, replace = TRUE)
  sum(dice)
})
prop.table(table(sim_data))
```

```
## sim_data
##      3      4      5      6      7      8      9     10     11     12     13
## 0.0051 0.0154 0.0272 0.0450 0.0669 0.0941 0.1179 0.1241 0.1247 0.1175 0.0952
##     14     15     16     17     18
## 0.0690 0.0494 0.0298 0.0147 0.0040
```

```
barplot(prop.table(table(sim_data)), main = "Probability distribution of sum of three dice", xlal
```



Problem 5

a. The pmf is approximated below.

```
sim_data <- replicate(10000, {
  balls <- sample(1:7, 2, replace = FALSE)
  sum(balls)
})
```

```
prop.table(table(sim_data))
```

```
## sim_data
##      3      4      5      6      7      8      9     10     11     12     13
## 0.0461 0.0476 0.0957 0.0938 0.1481 0.1378 0.1436 0.0992 0.0958 0.0466 0.0457
```

b. The least liley outcomes are that the sum is 3, 4, 12 or 13, as there is exactly one way to get each of those outcomes. All other outcomes have at least two ways of being achieved.

Problem 6

```
sim_data <- replicate(100000, {
  dice <- sample(1:6, 5, replace = TRUE)
  prod(dice)
})
pmf <- table(sim_data)
which.max(pmf)
```

```
## 360
##  58
```

```
pmf[58]
```

```
##  360
## 3892
```

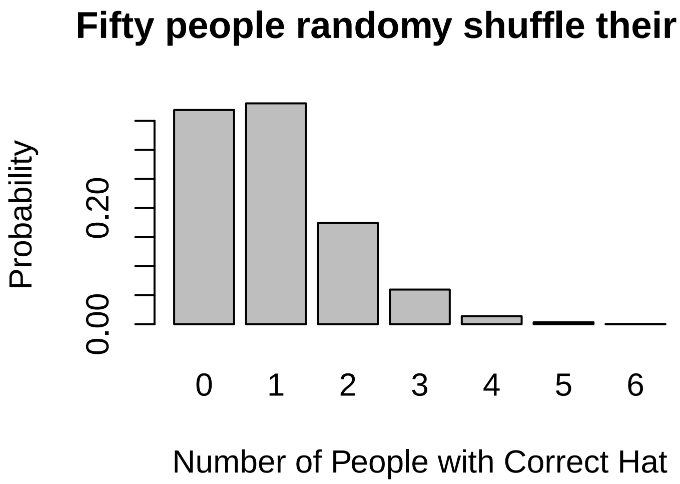
The most likely outcome is that the product is $360 = 2^3 \times 3^2 \times 5$, which has probability about .038.

Problem 7

```
sim_data <- replicate(10000, {
  hats <- sample(50)
  sum(hats == 1:50)
})
prop.table(table(sim_data))
```

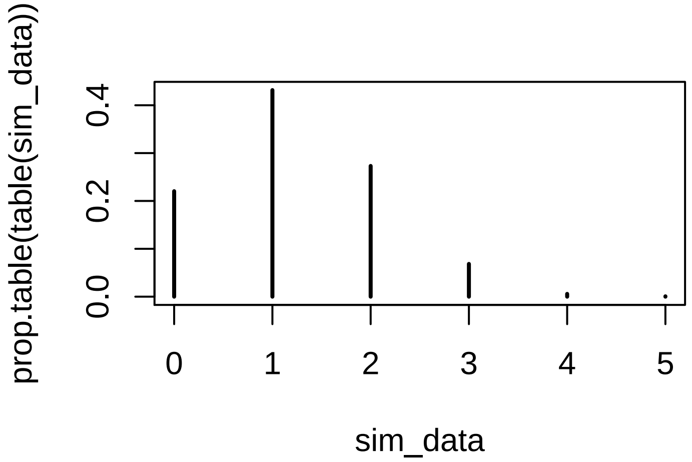
```
## sim_data
##      0      1      2      3      4      5      6
## 0.3687 0.3801 0.1743 0.0596 0.0138 0.0033 0.0002
```

```
barplot(prop.table(table(sim_data)), main = "Fifty people randomy shuffle their hats", xlab = "Nu
```



Problem 8

```
sim_data <- replicate(10000, {
  urns <- sample(1:10, 20, T)
  10 - length(unique(urns))
})
plot(prop.table(table(sim_data)))
```



Most likely outcome is 1 empty. Least likely outcome is 9 empty.

Problem 9

```
sim_data <- replicate(10000, {
  marbles <- c(1, sample(rep(c(-1, 1), times = c(5000, 5005)))) #start by serving person 1, then
  time_end <- min(which(cummax(cumsum(marbles)) - cummin(cumsum(marbles)) == 5)) #first time all
  person_end <- cumsum(marbles)[time_end] #person served last
  if(person_end < 1) {
    person_end <- person_end + 6 #if negative make positive
  }
  person_end
})
table(sim_data)
```

```
## sim_data
##    2    3    4    5    6
## 2029 1985 1957 2065 1964
```

```
prop.table(table(sim_data))
```

```
## sim_data
##      2      3      4      5      6
## 0.2029 0.1985 0.1957 0.2065 0.1964
```

The probability is about 20 percent for each person at the table.

```
all_people <- 1:6
sim_data <- replicate(10000, {
  marbles <- sample(rep(c(-1, 1), times = c(5000, 5005)))
  cur_person <- 1
  served_people <- cur_person
  i <- 1
  while(length(setdiff(all_people, served_people)) > 1) {
    cur_person <- cur_person + marbles[i]
    if(cur_person == 0) {
      cur_person <- 6
    }
    if(cur_person == 7) {
      cur_person <- 1
    }
    served_people <- c(served_people, cur_person)
    i <- i + 1
  }
  setdiff(all_people, served_people)
})
prop.table(table(sim_data))
```

```
## sim_data
##      2      3      4      5      6
## 0.2063 0.1969 0.2004 0.1964 0.2000
```

Problem 10

a. The expected number of ones and twos appears to be the same, about 1/3.

```
sim_data_number_ones <- replicate(10000, {
  die_roll <- sample(1:6, 30, replace = TRUE)
  first_even <- min(which(die_roll %% 2 == 0))
  sum(die_roll[1:first_even] == 1)
})
mean(sim_data_number_ones)
```

```
## [1] 0.3365
```

```
sim_data_number_twos <- replicate(10000, {
  die_roll <- sample(1:6, 30, replace = TRUE)
  first_even <- min(which(die_roll %% 2 == 0))
  sum(die_roll[1:first_even] == 2)
})
mean(sim_data_number_twos)
```

```
## [1] 0.3318
```

b. The pdfs are approximated below. They are definitely not the same, since we can get at most one "2."

```
prop.table(table(sim_data_number_ones))
```

```
## sim_data_number_ones
##      0      1      2      3      4      5      6 
## 0.7453 0.1937 0.0453 0.0114 0.0036 0.0006 0.0001
```

```
prop.table(table(sim_data_number_twos))
```

```
## sim_data_number_twos
##      0      1 
## 0.6682 0.3318
```

c. The variance of the number of ones (≈ 0.44) is higher than the variance of the number of twos ($\approx .22$).

```
var(sim_data_number_ones)
```

```
## [1] 0.4405118
```

```
var(sim_data_number_twos)
```

```
## [1] 0.2217309
```

Problem 11

```
1 + mean(replicate(10000,sum(cumsum(runif(20)) < 1)))
```

```
## [1] 2.7118
```

The exact answer is $e = 2.718\ldots$.

Problem 12

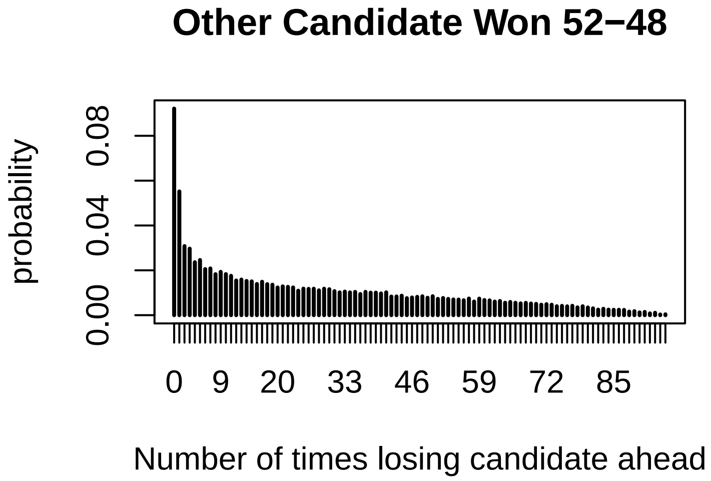
```
sim_data <- replicate(10000, {
  coins <- sample(c(-1, 1), 100, T)
  a <- cumsum(coins)
  sum(a == 0)
})
mean(sim_data)
```

```
## [1] 6.9933
```

About 7. Similar code for the number of times you have more H than T, which gives about 46.5.

Problem 13

```
sim_data <- replicate(50000, {
  votes <- sample(rep(c(1, -1), times = c(48, 52)))
  tally <- cumsum(votes)
  sum(tally > 0)
})
plot(prop.table(table(sim_data)), main = "Other Candidate Won 52-48", xlab = "Number of times lo
```



b.

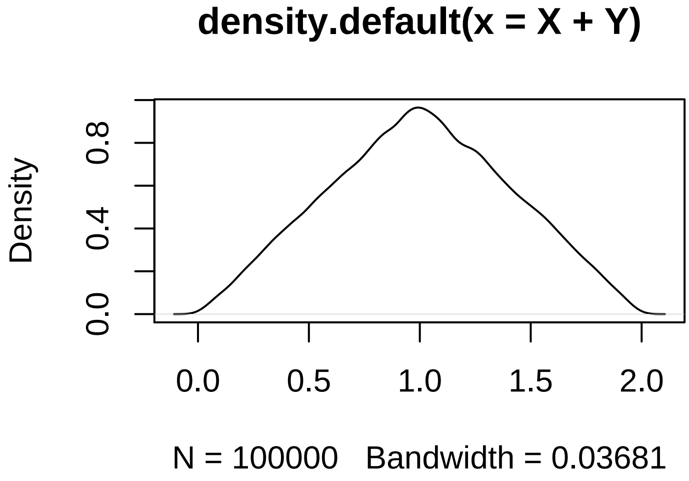
```r
mean(sim_data > 50)
```

```
## [1] 0.19522
```

The probability is about 20 percent that the losing candidate will be ahead more than half of the time.

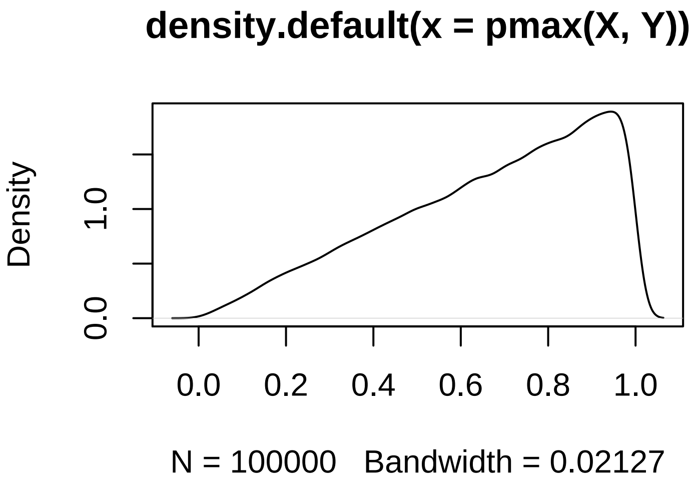
Problem 14

```r
X <- runif(100000,0,1)
Y <- runif(100000,0,1)
plot(density(X+Y))
```



Problem 15

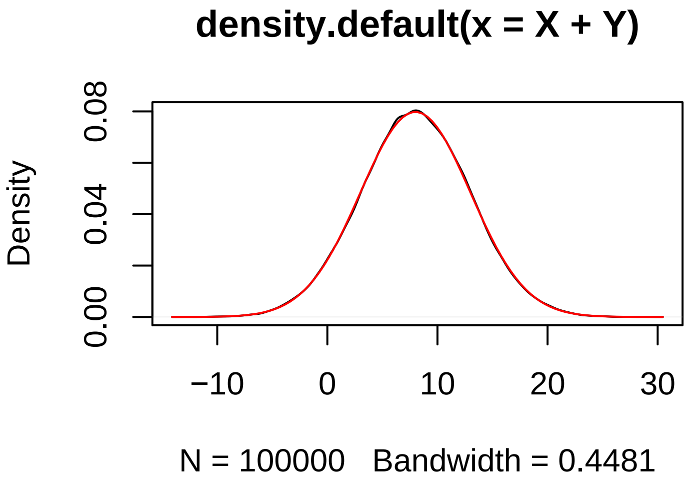
```r
X <- runif(100000,0,1)
Y <- runif(100000,0,1)
plot(density(pmax(X,Y)))
```



b. $P(1/3 \le X \le 2/3)$ is larger than $P(0 \le Z \le 1/3)$.

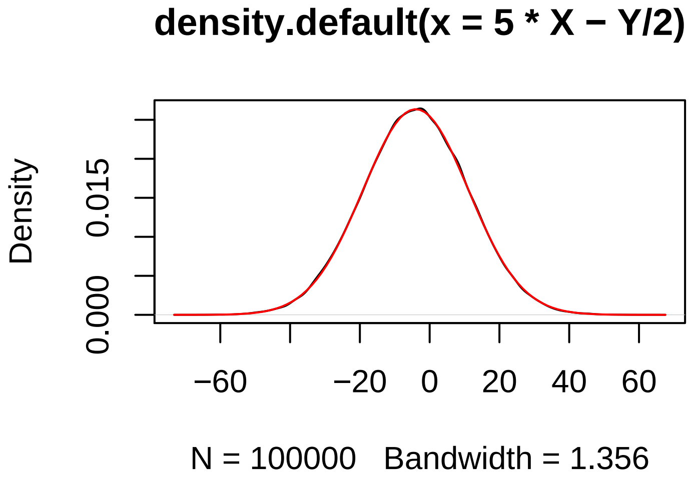
Problem 16 a. Mean 8, sd 5. b.

```
X <- rnorm(100000,0,3)
Y <- rnorm(100000,8,4)
plot(density(X+Y))
curve(dnorm(x,8,5),add=TRUE,col="red")
```



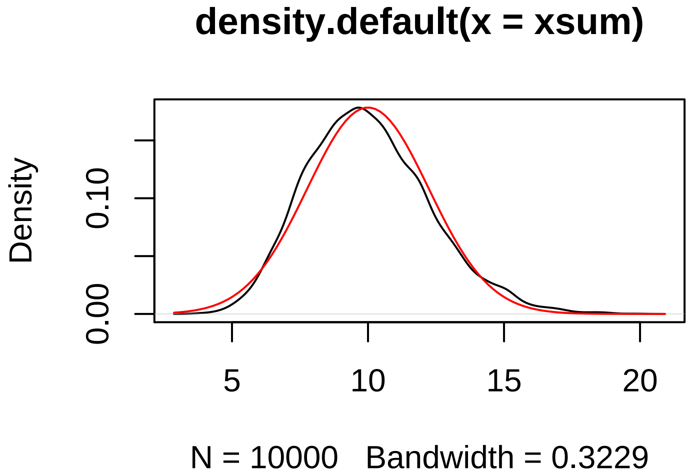
c. Mean -4, sd $\sqrt{15^2 + 2^2} = \sqrt{229} \approx 15.13$. d.

```
plot(density(5*X-Y/2))
curve(dnorm(x,-4,sqrt(229)),add=TRUE,col="red")
```



Problem 17 For Exp(2), $\mu = 1/2$ and $\sigma = 1/2$. Then $\mu(\overline{X}) = 1/2$ and $\sigma(\overline{X}) = 1/2\sqrt{20}$ The sum $X_1 + \cdots + X_{20} = 20\overline{X}$ and so it has mean 10 and sd $20/2\sqrt{20} = \sqrt{5}$.

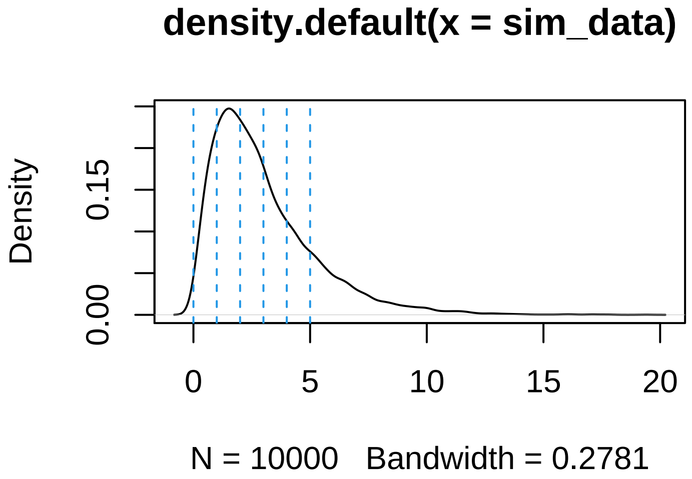
```r
xsum <- replicate(10000, sum(rexp(20,2)))
plot(density(xsum))
curve(dnorm(x,10,sqrt(5)), add=TRUE, col='red')
```



The normal curve fits pretty well, although the sum of exponentials is still skew.

Problem 18

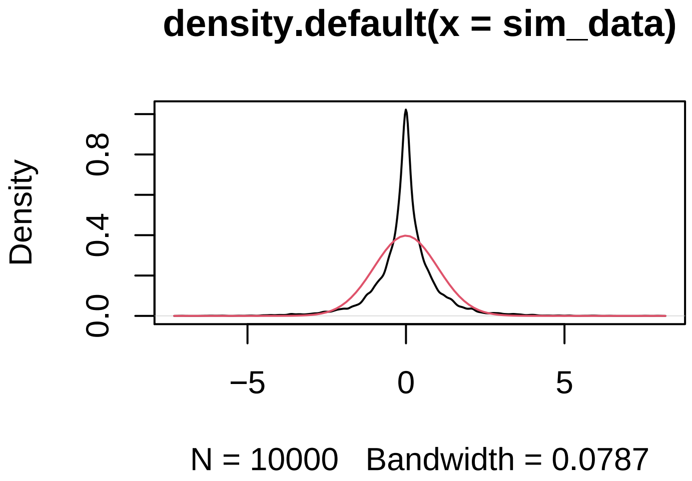
```r
sim_data <- replicate(10000, {
  max(rexp(2, rate = 1/2))
})

plot(density(sim_data))
abline(v = 0:5, lty = 2, col = 4)
```



It appears that $a \approx 1$ so that $P(1 \leq X \leq 2)$ is approximately the largest probability over all intervals of length 1.

Problem 19

```
sim_data <- replicate(10000, {
  x <- rnorm(1)
  y <- rnorm(1)
  x * y
})
plot(density(sim_data))
curve(dnorm(x, mean(sim_data), sd(sim_data)), add = T, col = 2)
```



Curves do not match. Product of normals is not normal.

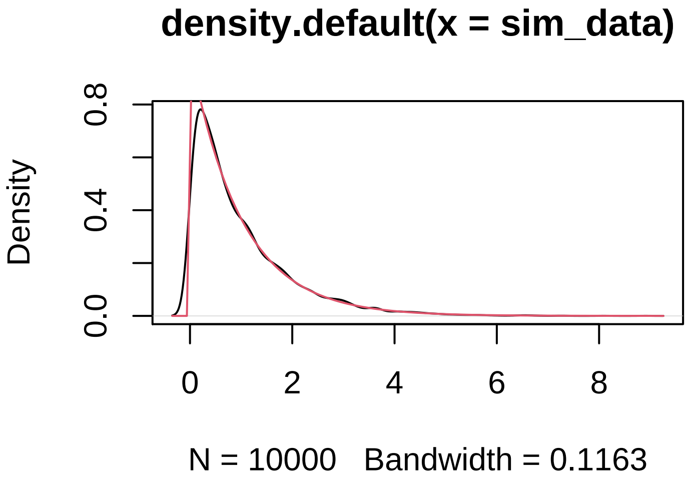
Problem 20

```
sim_data <- replicate(10000, {
  min(rexp(2, 1/2))
})
mean(sim_data)
```

```
## [1] 0.9936384
```

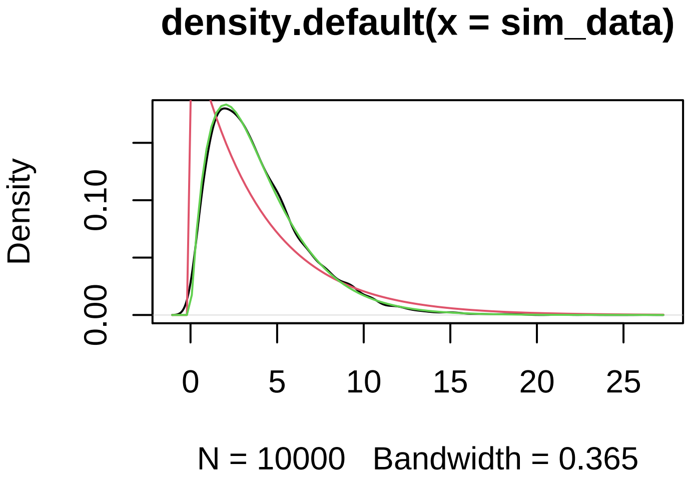
Since the mean of the simulated data is about 1, the rate is also about 1. We check with a density plot.

```
plot(density(sim_data))
curve(dexp(x, 1), add = T, col = 2)
```



Problem 21

```
sim_data <- replicate(10000, {
  x <- rchisq(1, 2)
  y <- rchisq(1, 2)
  x + y
})
plot(density(sim_data))
mu <- mean(sim_data)
curve(dexp(x, rate = 1/mu), add = T, col = 2) #red is exponential
curve(dchisq(x, df = mu), add = T, col = 3) #green is chi-squared. mean of chi-squared is df
```

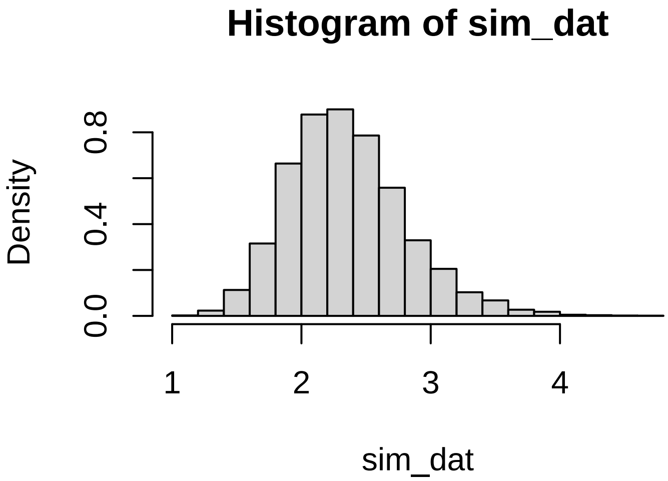


It is χ^2 with 4 degrees of freedom.

Problem 22

a. The maximum of 65 independent normal rvs is right skew with values mostly between 1.5 and 3.6.

```
sim_dat <- replicate(10000, {
  max(rnorm(65))
})
hist(sim_dat, probability = T)
```

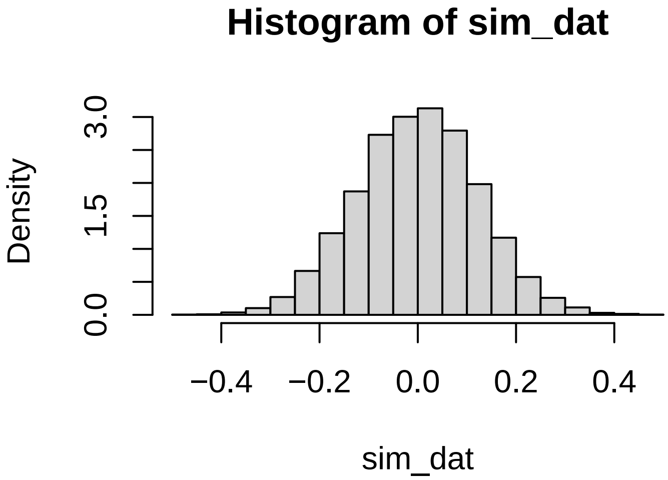


```
quantile(sim_dat, c(.01, .99))
```

```
##       1%      99%
## 1.476763 3.646702
```

b. The mean of 65 independent normal rvs is symmetric and mostly between -0.3 and 0.3.

```
sim_dat <- replicate(10000, {
  mean(rnorm(65))
})
hist(sim_dat, probability = T)
```



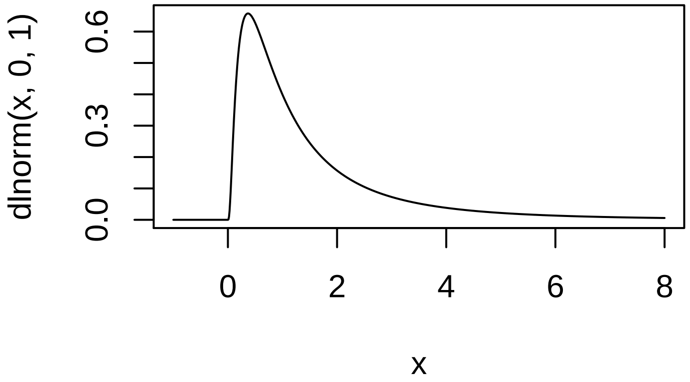
```
quantile(sim_dat, c(.01, .99))
```

```
##         1%        99%
## -0.2878724  0.2905126
```

c. It seems Feynamn thought he was at least 1/4 of a standard deviation smarter than average, but perhaps not 3.5 standard deviations smarter than average.

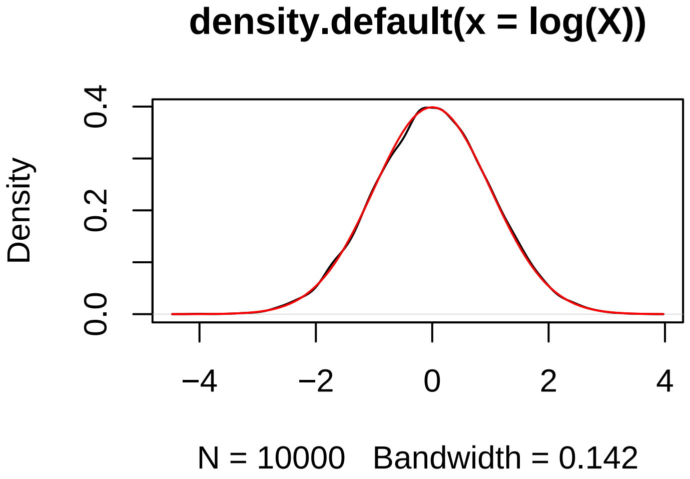
Problem 23 a.

```
x <- seq(-1,8,.01)
plot(x,dlnorm(x,0,1),type='l')
```



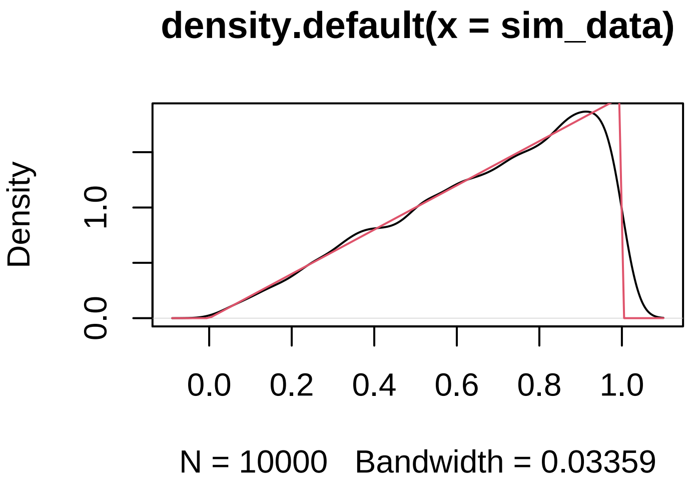
b.

```
X <- rlnorm(10000,0,1)
plot(density(log(X)))
curve(dnorm(x),add=TRUE, col='red')
```



Problem 24

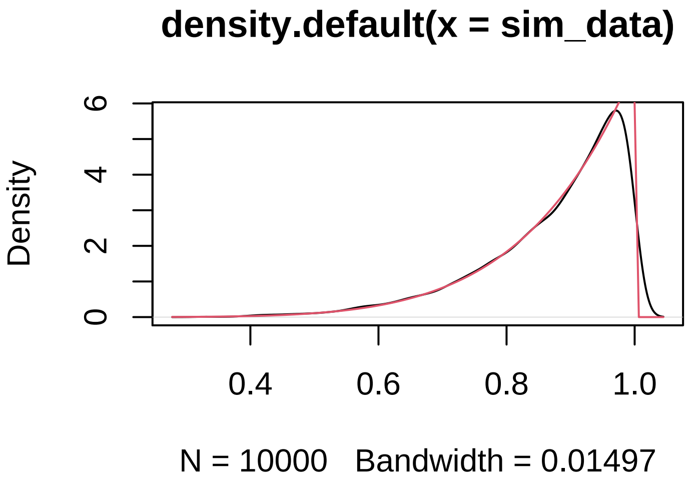
```
sim_data <- replicate(10000, {
  max(runif(2))
})

plot(density(sim_data))
curve(dbeta(x, 2, 1), add = T, col = 2)
```



The max of two uniform random variables on $[0, 1]$ does appear to be a beta random variable with parameters 2 and 1.

Problem 25

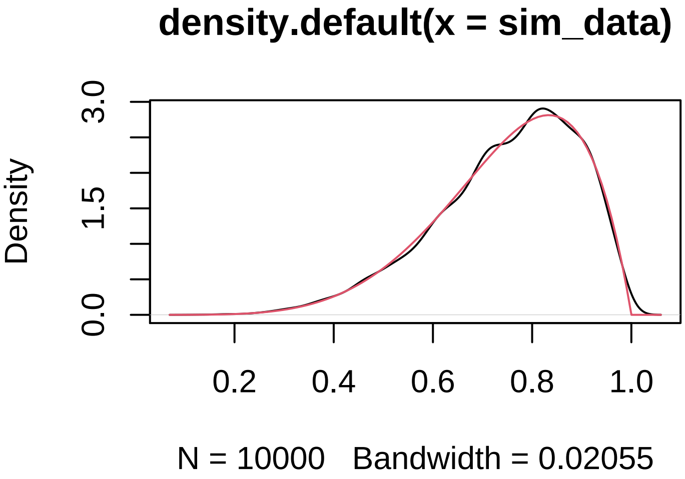
```
sim_data <- replicate(10000, {
  max(runif(7))
})

plot(density(sim_data))
curve(dbeta(x, 7, 1), add = T, col = 2)
```



The max of seven $\text{Unif}(0, 1)$ random variables appears to be a beta random variable with parameters 7 and 1.

Problem 26

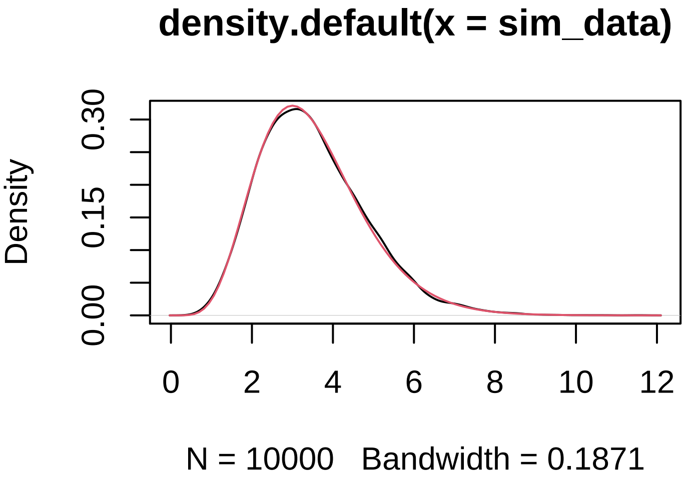
```
sim_data <- replicate(10000, {
  sort(runif(7))[6]
})

plot(density(sim_data))
curve(dbeta(x, 6, 2), add = T, col = 2)
```



The second largest of seven uniform random variables appears to be a beta random variable with parameters 6 and 2.

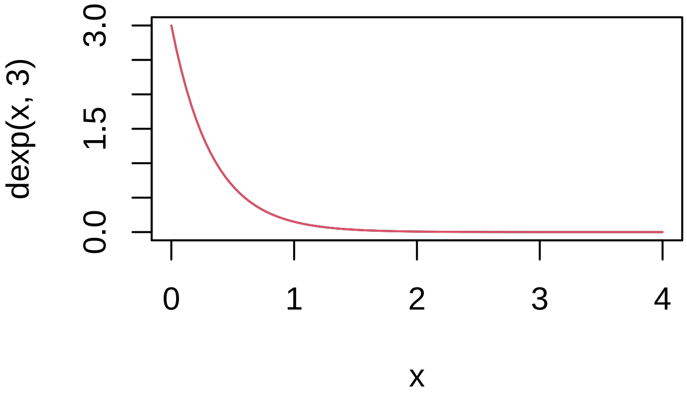
Problem 27

```
sim_data <- replicate(10000, {
  x <- rgamma(1, shape = 3, rate = 2)
  y <- rgamma(1, shape = 4, rate = 2)
  x + y
})
plot(density(sim_data))
curve(dgamma(x, 7, 2), add = T, col = 2)
```



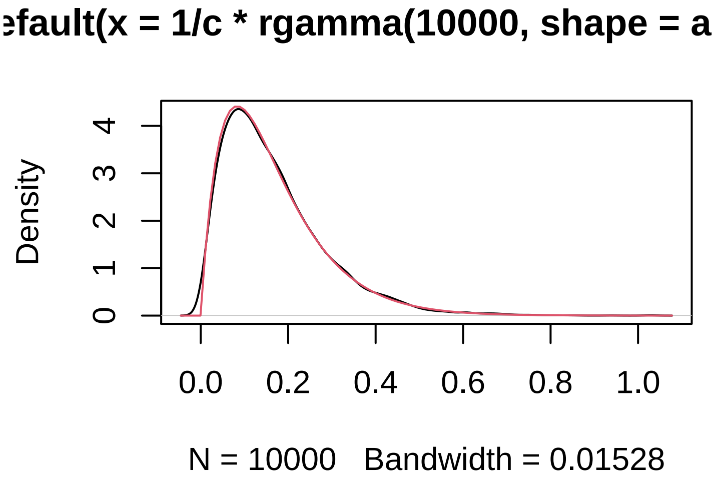
Problem 28

```
curve(dexp(x, 3), from = 0, to = 4)
curve(dgamma(x, shape = 1, rate = 3), add= T, col = 2)
```



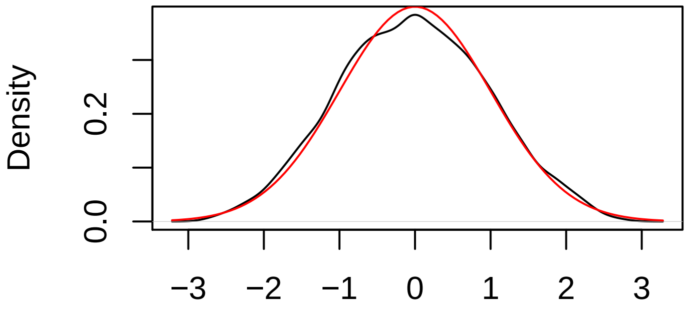
Problem 29

```
alpha <- 2
beta <- 3
c <- 4
plot(density(1/c * rgamma(10000, shape = alpha, rate = beta)))
curve(dgamma(x, shape = alpha, rate = c * beta), add = T, col = 2)
```



Problem 30

```
mu <- 0.5
sigma <- sqrt(1/12)
n <- 3
Xbar <- replicate(10000,mean(runif(n,0,1)))
Z <- (Xbar - mu)/(sigma/sqrt(n))
plot(density(Z), xlab=NA, main=NA)
curve(dnorm(x), add=TRUE, col="red")
```



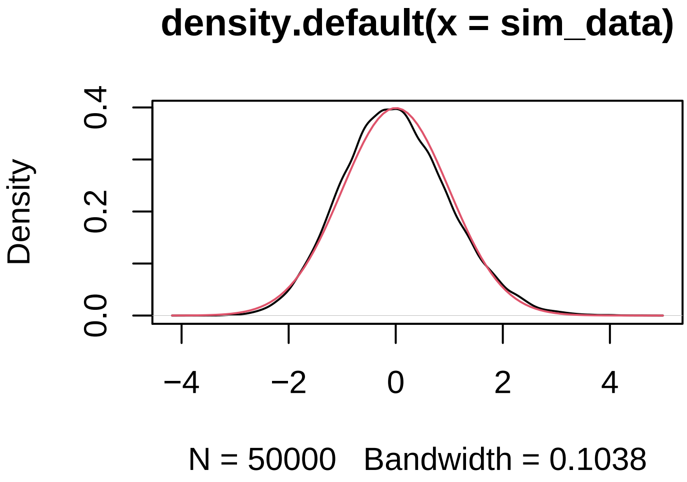
When $n = 3$ as above, the graph begins to look bump shaped. By $n = 6$, it's pretty close to normal.

Problem 31

a. The mean is 1/10 and the standard deviation is also 1/10.

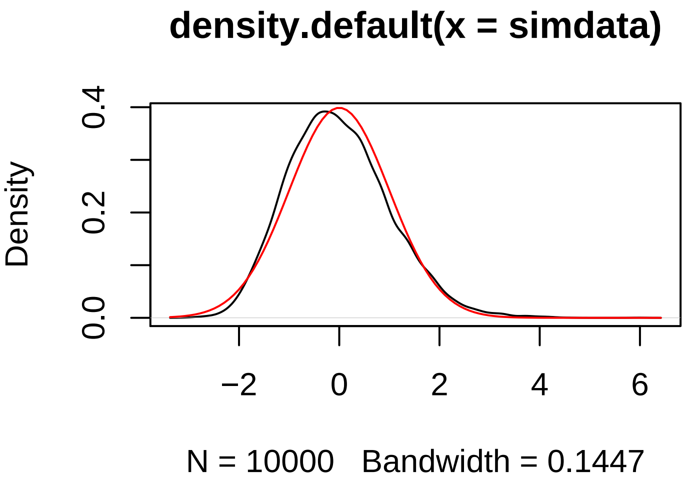
b. Answers will vary, values between 40 and 100 seem like reasonable responses.

```
N <- 70
sim_data <- replicate(50000, {
  dat <- rexp(N, rate = 10)
  (mean(dat) - 1/10)/(1/10/sqrt(N))
})
plot(density(sim_data))
curve(dnorm(x), add = T, col = 2)
```



Problem 32

```
mu <- 2
sigma <- 2
n <- 20
simdata <- replicate(10000,{
  xbar <- mean(rchisq(n,2))
  (xbar - mu)/(sigma/sqrt(n))
})
plot(density(simdata))
curve(dnorm(x), add=TRUE, col="red")
```



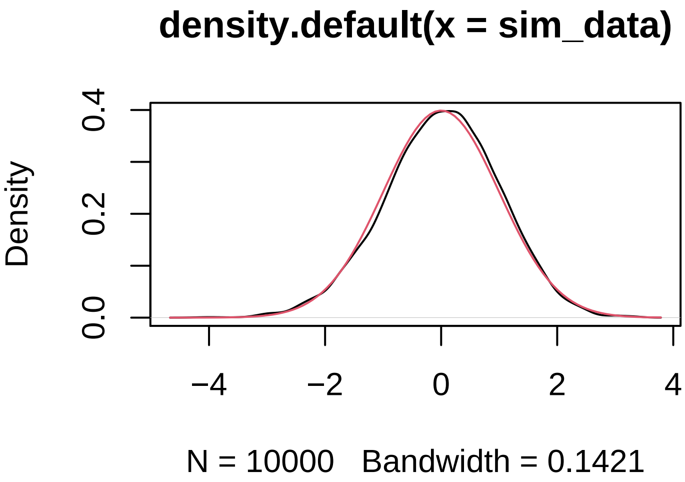
When $n = 20$ as above, the graph is still somewhat skew. By $n = 50$, it's pretty close to normal.

Problem 33

a. The mean is 8 and the standard deviation is $\sqrt{npq} \approx 1.264$.

b. Answers will vary, but numbers between 10 and 30 seem reasonable.

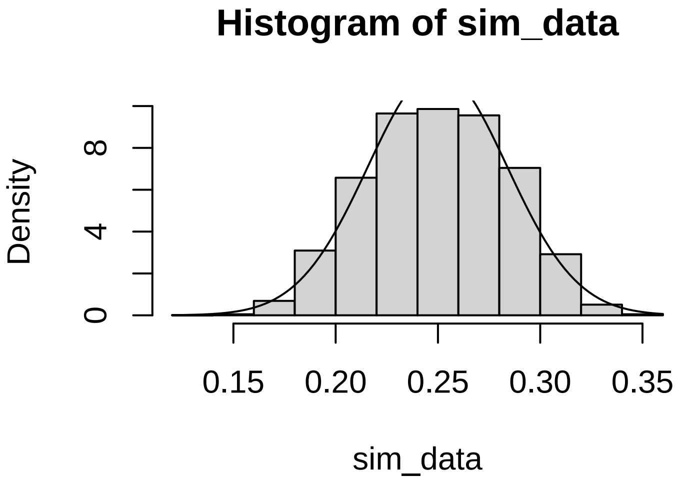
```
N <- 20
sim_data <- replicate(10000, {
  dat <- rbinom(N, 10, .8)
  (mean(dat) - 8)/(1.264/sqrt(N))
})
plot(density(sim_data))
curve(dnorm(x), add = T, col = 2)
```



Problem 34

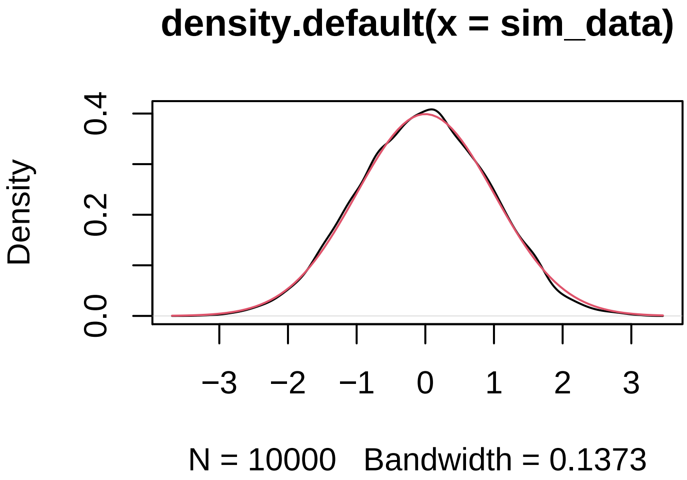
```
sim_data <- replicate(10000, {
  sam <- runif(1000, -1, 1)
  sam <- c(sam, runif(1, 200, 300))
  mean(sam)
})
```

```
hist(sim_data, probability = T)
curve(dnorm(x, mean(sim_data), sd(sim_data)), add = T)
```



Problem 35

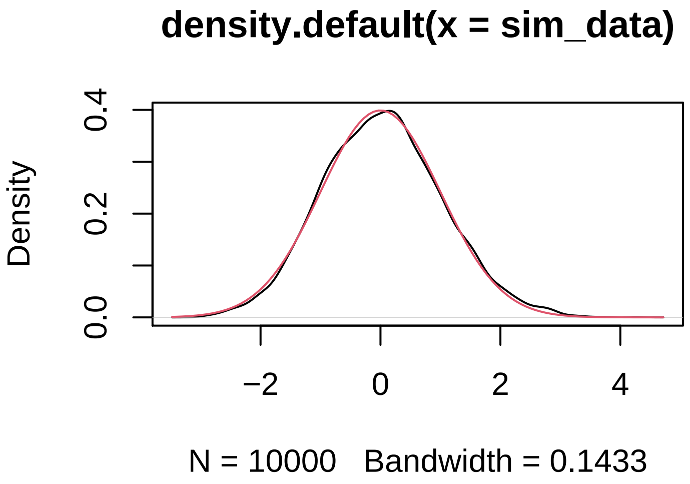
```
N <- 25
med <- 0
sim_data <- replicate(10000, {
  sam <- runif(N, -1, 1)
  (median(sam) - 0)/(1/(2 * sqrt(N) * dunif(0, -1, 1)))
})
plot(density(sim_data))
curve(dnorm(x, 0, 1), add = T, col = 2)
```



About 25 samples.

```
N <- 200
med <- log(2)
sim_data <- replicate(10000, {
  sam <- rexp(N, 1)
  (median(sam) - med)/(1/(2 * sqrt(N) * dexp(med, 1)))
})
```

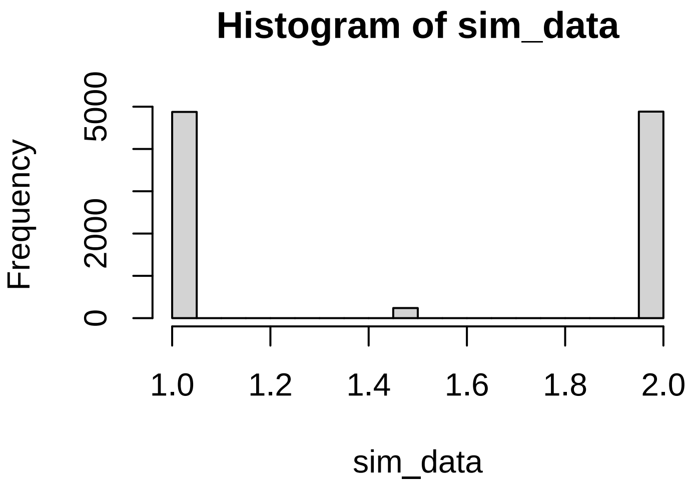
```
plot(density(sim_data))
curve(dnorm(x, 0, 1), add = T, col = 2)
```



About 200 samples before we get a good match.

c.

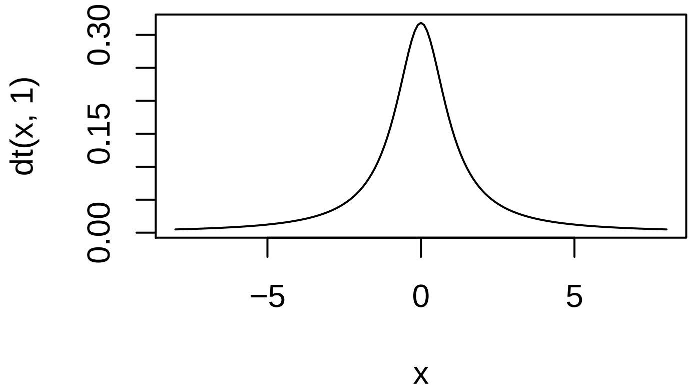
```
N <- 1000
sim_data <- replicate(10000, {
  sam <- rbinom(N, 3, .5)
  median(sam)
})
hist(sim_data)
```



This is clearly not approaching a normal distribution.

Problem 36 a.

```
x <- seq(-8,8,.1)
plot(x,dt(x,1),type='l')
```



b. The means are all over the place

```
replicate(6,mean(rt(100,1)))
```

```
## [1]  0.7500212 -0.6173526  3.6359666  0.1010925 -0.6539421  2.0729001
```

```
replicate(6,mean(rt(1000,1)))
```

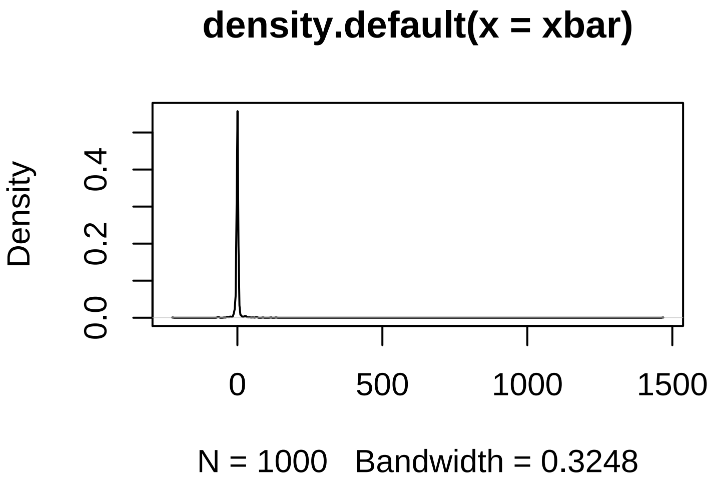
```
## [1]  0.168481481  7.851446000 -0.130871698 -0.007731779 -9.387279027
## [6] 10.162288470
```

```
replicate(6,mean(rt(10000,1)))
```

```
## [1]  1.66553834 10.09726453  0.21268980  2.02553025 -0.63505532  0.03342417
```

c. Try some density plots or histograms. They have the same general look - a sharp spike and really long tails. Not normal.

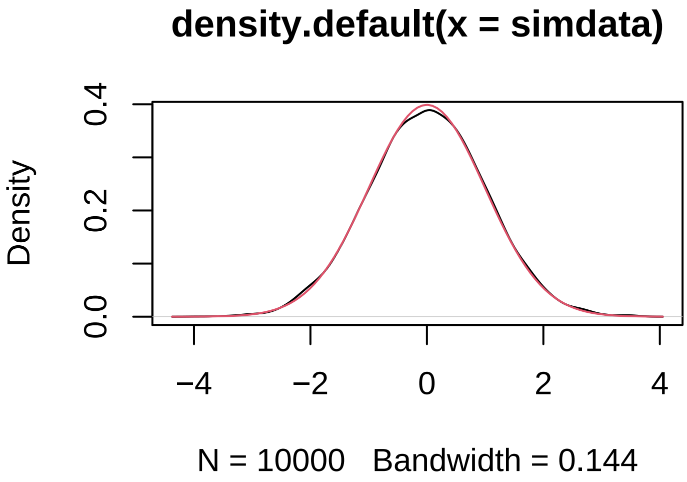
```
xbar <- replicate(1000,mean(rt(1000,1)))
plot(density(xbar))
```



Problem 37

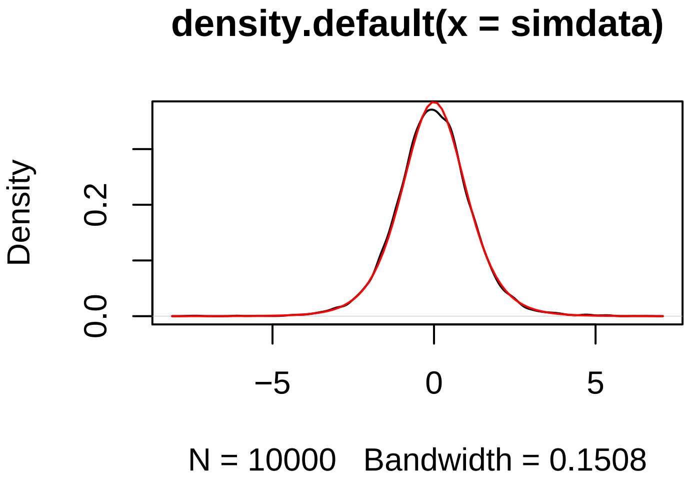
$\overline{X}$ is normal with mean 2 and standard deviation $\sigma/\sqrt{n} = 3/3$, so $a = 2$ and $b = 1$. Check via simulations:

```
simdata <- replicate(10000, {
  dat <- rnorm(9, 2, 3)
  (mean(dat) - 2)/(1)
})
plot(density(simdata))
curve(dnorm(x), add = T, col = 2)
```



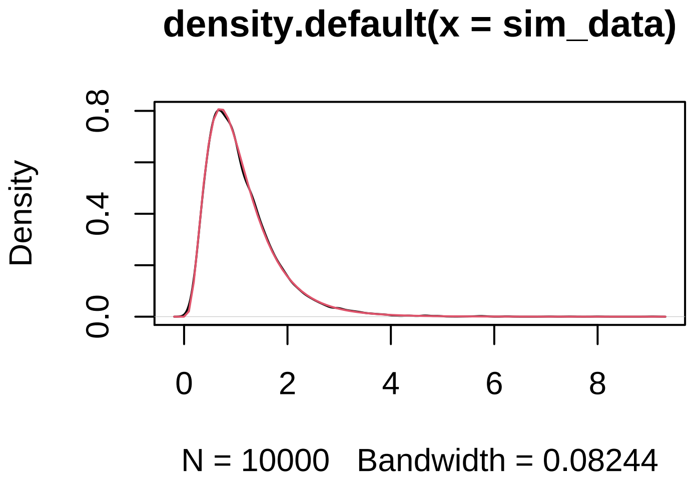
Problem 38

```
simdata <- replicate(10000,{
  x <- rnorm(8,2,3)
  s <- sd(x)
  xbar <- mean(x)
  (xbar - 2) / (s / sqrt(8))
})
plot(density(simdata))
curve(dt(x,7),add=TRUE,col="red")
```



Problem 39

```
sim_data <- replicate(10000, {
  x <- rnorm(10)
  y <- rnorm(20, 1, 2)
  (var(x)/1^2)/(var(y)/2^2)
})

plot(density(sim_data))
curve(df(x, 9, 19), add = T, col = 2)
```



Problem 40

We have:

$$\begin{aligned}\frac{1}{\sigma^2}\left((X_1-(X_1+X_2)/2)^2+(X_2-(X_1+X_2)/2)^2\right)^2 &= \frac{1}{\sigma^2}((X_2/2-X_1/2)^2+(X_1/2-X_2/2)^2)\\ &= \frac{1}{\sigma^2}(\sqrt{2}X_1/2-\sqrt{2}X_2/2)^2\\ &= \left(\frac{1}{\sqrt{2}\sigma}(X_1-X_2)\right)^2.\end{aligned}$$

Since $X_1 - X_2 \sim \text{Norm}(0, \sqrt{2}\sigma)$, $\frac{1}{\sqrt{2}\sigma}(X_1 - X_2) \sim \text{Norm}(0, 1)$, and $\left(\frac{1}{\sqrt{2}\sigma}(X_1 - X_2)\right)^2 \sim \chi^2_1$.

Problem 41

a. With 20 simulations, the five estimates of the expected value of the median are 2.271168, 1.060303, 2.452576, 1.280327 and 1.770506.

```
set.seed(3850)
N <- 20
replicate(5, {sim_data <- replicate(N, {
  dat <- rnorm(5, 2, 4)
  median(dat)
})
mean(sim_data)
})
```

```
## [1] 2.271168 1.060303 2.452576 1.280327 1.770506
```

b. With 1000 simulations, the estimates of the expected value of the median are 1.964379 2.012994 2.021984 2.060744 and 1.903849.

```
set.seed(3850)
N <- 1500
```

```
replicate(5, {sim_data <- replicate(N, {
  dat <- rnorm(5, 2, 4)
  median(dat)
})
mean(sim_data)
})
```

```
## [1] 1.964379 2.012994 2.021984 2.060744 1.903849
```

c. With 20000 simulations, the estimates of the expected value of the median are 1.989225 2.006970 1.995919 2.022799 and 2.014118.

```
set.seed(3850)
N <- 20000
replicate(5, {sim_data <- replicate(N, {
  dat <- rnorm(5, 2, 4)
  median(dat)
})
mean(sim_data)
})
```

```
## [1] 1.989225 2.006970 1.995919 2.022799 2.014118
```

d. It appears that the estimate of the expected value of the median is approaching 2 as the number of replicates goes to infinity.

Problem 42

```
sim_data <- replicate(10000, {
  dat <- rexp(8)
  median(dat)
})
mean(sim_data)
```

```
## [1] 0.7667651
```

Since the expected value of the median of a random sample of 8 exponential random variables is not 1, the median is not an unbiased estimator for the mean.

Problem 43

Repeating several times, it appears that the statistic is an unbiased estimator for σ^2.

```
sim_data <- replicate(50000, {
  dat <- rnorm(10, 1, 3)
  1/10 * sum((dat - 1)^2)
})
mean(sim_data)
```

```
## [1] 8.98643
```

Problem 44

Repeating several times, it appears that the statistic is an unbiased estimator of the variance $\sigma^2 = 4$.

```
sim_data <- replicate(50000, {
  dat <- rnorm(10, 0, 2)
  1/9 * sum((dat - mean(dat))^2)
})
mean(sim_data)
```

```
## [1] 4.00278
```

Problem 45

Repeating several times, it appears that the statistic consistently underestimates the standard deviation $\sigma = 2$.

```
sim_data <- replicate(50000, {
  dat <- rnorm(10, 0, 2)
  sqrt(1/9 * sum((dat - mean(dat))^2))
})
mean(sim_data)
```

```
## [1] 1.946557
```

Problem 46

```
sim_data1 <- replicate(20000, {
  dat <- rnorm(10, 0, 2)
  mu <- mean(dat)
  1/10 * sum((dat - mu)^2)
})
mean((sim_data1 - 4)^2)
```

```
## [1] 3.060738
```

```
sim_data2 <- replicate(20000, {
  dat <- rnorm(10, 0, 2)
  mu <- mean(dat)
  1/10.5 * sum((dat - mu)^2)
})
mean((sim_data2 - 4)^2)
```

```
## [1] 2.903238
```

We see that $\hat{\sigma}_2^2$ has lower MSE.

6

Data Manipulation – Solutions

Problem 1

```
iris_tbl <- mutate(tibble(iris),
                   Sepal.Area = Sepal.Width * Sepal.Length,
                   Petal.Area = Petal.Width * Petal.Length)
iris_tbl
```

```
## # A tibble: 150 x 7
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species Sepal.Area
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>        <dbl>
##  1          5.1         3.5          1.4         0.2 setosa        17.8
##  2          4.9         3            1.4         0.2 setosa        14.7
##  3          4.7         3.2          1.3         0.2 setosa        15.0
##  4          4.6         3.1          1.5         0.2 setosa        14.3
##  5          5           3.6          1.4         0.2 setosa        18
##  6          5.4         3.9          1.7         0.4 setosa        21.1
##  7          4.6         3.4          1.4         0.3 setosa        15.6
##  8          5           3.4          1.5         0.2 setosa        17
##  9          4.4         2.9          1.4         0.2 setosa        12.8
## 10          4.9         3.1          1.5         0.1 setosa        15.2
## # ... with 140 more rows, and 1 more variable: Petal.Area <dbl>
```

Problem 2

```
austen <- fosdata::austen
a.emma <- austen %>% filter(novel == "Emma")
b.vars <- austen %>% select(word, word_length, novel)
c.ordered <- austen %>% arrange(desc(word_length))
# d
austen %>% slice_max(word_length)
```

```
##                    word sentence chapter word_length stop_word sentiment_score
## 1 conscience-stricken     5281      34          19     FALSE               0
## 2 respectable-looking     4334      43          19     FALSE               0
##                   novel
## 1                  Emma
## 2 Pride and Prejudice
```

```
# e
austen %>% summarize(mean(word_length))
```

```
##   mean(word_length)
## 1          4.325518
```

```
f.distinct <- austen %>% distinct(word, word_length, sentiment_score)
```

Problem 3

```
mpg <- ggplot2::mpg
mpg %>% slice_max(hwy) %>% select(manufacturer, model, year, hwy)
```

```
## # A tibble: 2 x 4
##   manufacturer model       year   hwy
##   <chr>        <chr>      <int> <int>
## 1 volkswagen   jetta       1999    44
## 2 volkswagen   new beetle  1999    44
```

```
mpg %>% filter(class == "compact") %>% summarize(cty.mean = mean(cty))
```

```
## # A tibble: 1 x 1
##   cty.mean
##      <dbl>
## 1     20.1
```

```
mpg %>% group_by(class) %>% summarize(cty.mean = mean(cty)) %>% arrange(desc(cty.mean))
```

```
## # A tibble: 7 x 2
##   class      cty.mean
##   <chr>         <dbl>
## 1 subcompact     20.4
## 2 compact        20.1
## 3 midsize        18.8
## 4 minivan        15.8
## 5 2seater        15.4
## 6 suv            13.5
## 7 pickup         13
```

```
mpg %>% mutate(ch.diff = abs(hwy-cty)) %>% slice_min(ch.diff) %>%
  select(manufacturer, model, year, ch.diff)
```

```
## # A tibble: 3 x 4
##   manufacturer model               year ch.diff
##   <chr>        <chr>              <int>   <int>
## 1 nissan       pathfinder 4wd      1999       2
## 2 toyota       4runner 4wd         1999       2
## 3 toyota       toyota tacoma 4wd   1999       2
```

```
mpg %>% group_by(year) %>% summarize(mean.hwy = mean(hwy)) %>% arrange(desc(mean.hwy))
```

```
## # A tibble: 2 x 2
##    year mean.hwy
##   <int>    <dbl>
## 1  2008     23.5
## 2  1999     23.4
```

Problem 4

```
DrinksWages <- HistData::DrinksWages
DrinksWages %>% group_by(class) %>% summarize(mean(wage))
```

```
## # A tibble: 3 x 2
##   class `mean(wage)`
##   <fct>        <dbl>
## 1 A             20.3
## 2 B             27.4
## 3 C             34.0
```

```
DrinksWages %>% filter(n >= 10) %>% mutate(pct.drink = drinks/n) %>% slice_max(n=3,pct.drink)
```

```
##   class   trade sober drinks     wage  n pct.drink
## 1     A  cabmen     1     10 18.41667 11 0.9090909
## 2     B tailors     4     15 27.00000 19 0.7894737
## 3     C   mason     5     17 34.08333 22 0.7727273
```

Problem 5

```
oly12 <- VGAMdata::oly12
oly12 %>% group_by(Country) %>% summarize(medals = sum(Total)) %>% slice_max(medals)
```

```
## # A tibble: 1 x 2
##   Country                  medals
##   <fct>                     <int>
## 1 United States of America     79
```

```
oly12 %>% filter(Sex=="M") %>% group_by(Country) %>%
  summarize(competitors = n(), mean.weight = mean(Weight)) %>%
  filter(competitors >= 10) %>% arrange(desc(mean.weight)) %>% slice_max(n=3,mean.weight)
```

```
## # A tibble: 3 x 3
##   Country   competitors mean.weight
##   <fct>           <int>       <dbl>
## 1 Estonia            21        96.8
## 2 Guatemala          10        80.8
## 3 Finland            26        77.9
```

MovieLens

```
movies <- fosdata::movies
```

Problem 6 *The Shawshank Redemption* is the best movie of all time, with a 4.43 average rating.

```
movies %>% group_by(title) %>%
  summarize(meanRating = mean(rating), numRating = n()) %>%
  filter(numRating >= 30) %>% slice_max(meanRating)
```

```
## # A tibble: 1 x 3
##   title                            meanRating numRating
##   <chr>                                 <dbl>     <int>
## 1 Shawshank Redemption, The (1994)       4.43       317
```

Problem 7 Comedy, with 7196 ratings.

```
movies %>%
  group_by(genres) %>%
  summarize(numRatings = n()) %>%
  slice_max(numRatings)
```

```
## # A tibble: 1 x 2
##   genres numRatings
##   <chr>       <int>
## 1 Comedy       7196
```

Problem 8 *Amelie* has the highest rating of 4.18, *You've Got Mail* has the lowest, 3.12.

```
movies %>% filter(genres == "Comedy|Romance") %>%
  group_by(title) %>% summarize(ratings = n(), mr = mean(rating)) %>%
  filter(ratings >= 50) %>% slice_max(mr)
```

```
## # A tibble: 1 x 3
##   title                                                ratings    mr
##   <chr>                                                  <int> <dbl>
## 1 Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)     120  4.18
```

```
movies %>% filter(genres == "Comedy|Romance") %>%
  group_by(title) %>% summarize(ratings = n(), mr = mean(rating)) %>%
  filter(ratings >= 50) %>% slice_min(mr)
```

```
## # A tibble: 1 x 3
##   title                  ratings    mr
##   <chr>                    <int> <dbl>
## 1 You've Got Mail (1998)      50  3.12
```

Problem 9 The most ratings for a movie with a 4.0 or higher average is *Forrest Gump* with 329.

```
movies %>% group_by(title) %>% summarize(ratings = n(), mr = mean(rating)) %>%
  filter(mr >= 4) %>% slice_max(ratings)
```

```
## # A tibble: 1 x 3
##   title               ratings    mr
##   <chr>                 <int> <dbl>
## 1 Forrest Gump (1994)     329  4.16
```

Problem 10 User 53 gave the highest ratings.

```
movies %>% group_by(userId) %>% summarize(mr = mean(rating)) %>% slice_max(mr)
```

```
## # A tibble: 1 x 2
##   userId    mr
##    <int> <dbl>
## 1     53     5
```

Lahman Batting & Pitching

```
library(Lahman)
```

Problem 11 jennihu01 (Hughie Jennings 1869-1928) was hit by more pitches than any other player.

```
Batting %>% group_by(playerID) %>% summarize(hbp = sum(HBP)) %>% slice_max(hbp)
```

```
## # A tibble: 1 x 2
##   playerID    hbp
##   <chr>     <int>
## 1 jennihu01   287
```

Problem 12 There were 434 doubles in 1871.

```
Batting %>% group_by(yearID) %>% summarize(doubles = sum(X2B)) %>% filter(yearID == 1871)
```

```
## # A tibble: 1 x 2
##   yearID doubles
##    <int>   <int>
## 1   1871     434
```

Problem 13 The team with the most total home runs all time is NYA, the New York (American league) Yankees.

```
Batting %>% group_by(teamID) %>% summarize(hr = sum(HR)) %>% slice_max(hr)
```

```
## # A tibble: 1 x 2
##   teamID    hr
##   <fct>  <int>
## 1 NYA    16309
```

Problem 14 The player with the most runs per game (given 500 games played) is wrighge01, George Wright (1847-1937).

```
Batting %>% group_by(playerID) %>% summarize(games = sum(G), rpg = sum(R)/sum(G)) %>%
  filter(games >= 500) %>% slice_max(rpg)
```

```
## # A tibble: 1 x 3
##   playerID  games   rpg
##   <chr>     <int> <dbl>
## 1 wrighge01   591  1.13
```

Problem 15 a. eggleda01, Dave Eggler (1849-1902) had 2544 at bats but never hit a home run.

```
Batting %>% group_by(playerID) %>% summarize(abs = sum(AB), hrs = sum(HR)) %>%
  filter(hrs == 0) %>% slice_max(abs)
```

```
## # A tibble: 1 x 3
##   playerID    abs   hrs
##   <chr>     <int> <int>
## 1 eggleda01  2544     0
```

b. As of 2020, cuetojo01 (Johnny Cueto) is the active player with the most at bats and no home run.

```
mostRecentYear <- max(Batting$yearID)  # you could just use the number
Batting %>% group_by(playerID) %>%
  summarize(atbats = sum(AB), hrs = sum(HR), finalYear = max(yearID)) %>%
  filter(hrs == 0 & finalYear == mostRecentYear) %>% slice_max(atbats)
```

```
## # A tibble: 1 x 4
##   playerID  atbats   hrs finalYear
##   <chr>      <int> <int>     <int>
## 1 cuetojo01    516     0      2020
```

Problem 16

Solved without taking stint into account

a. grandcu01 is Curtis Granderson. b. In 2020, Kyle Tucker tuckeky01 was the triples leader with 6.

```
Batting %>% filter(yearID >= 1960) %>% slice_max(X3B)
```

```
##    playerID yearID stint teamID lgID   G  AB   R   H X2B X3B HR RBI SB CS BB
## 1 grandcu01   2007     1    DET   AL 158 612 122 185  38  23 23  74 26  1 52
##    SO IBB HBP SH SF GIDP
## 1 141   3   5  5  2    3
```

```
Batting %>% group_by(yearID) %>% slice_max(X3B) %>% ungroup() %>% slice_min(X3B)
```

```
## # A tibble: 1 x 22
##   playerID  yearID stint teamID lgID      G    AB     R     H   X2B   X3B    HR
##   <chr>      <int> <int> <fct>  <fct> <int> <int> <int> <int> <int> <int> <int>
## 1 tuckeky01   2020     1 HOU    AL       58   209    33    56    12     6     9
## # ... with 10 more variables: RBI <int>, SB <int>, CS <int>, BB <int>,
## #   SO <int>, IBB <int>, HBP <int>, SH <int>, SF <int>, GIDP <int>
```

c. In 1962, Maury Wills had 104 stolen bases, 72 more than the next closest player.

```
Batting %>% group_by(yearID) %>% slice_max(SB, n=2) %>%
  summarize(SBlead = max(SB) - min(SB)) %>% slice_max(SBlead)
```

```
## # A tibble: 1 x 2
##   yearID SBlead
##    <int>  <int>
## 1   1962     72
```

```
Batting %>% group_by(yearID) %>% slice_max(SB, n=2) %>% filter(yearID == 1962)
```

```
## # A tibble: 2 x 22
## # Groups:   yearID [1]
##   playerID  yearID stint teamID lgID      G    AB     R     H   X2B   X3B    HR
##   <chr>      <int> <int> <fct>  <fct> <int> <int> <int> <int> <int> <int> <int>
## 1 willsma01   1962     1 LAN    NL      165   695   130   208    13    10     6
## 2 daviswi02   1962     1 LAN    NL      157   600   103   171    18    10    21
## # ... with 10 more variables: RBI <int>, SB <int>, CS <int>, BB <int>,
## #   SO <int>, IBB <int>, HBP <int>, SH <int>, SF <int>, GIDP <int>
```

Solved with taking stint into account - no difference!

a. grandcu01 is Curtis Granderson. b. In 1872, the triples leader was gouldch01 with 8.

```
stinty <- Batting %>% group_by(playerID, yearID) %>% summarize(X3B = sum(X3B), SB = sum(SB)) %>%
stinty %>% filter(yearID >= 1960) %>% slice_max(X3B)
```

```
## # A tibble: 1 x 4
##   playerID  yearID   X3B    SB
##   <chr>      <int> <int> <int>
## 1 grandcu01   2007    23    26
```

```
stinty %>% group_by(yearID) %>% slice_max(X3B) %>% ungroup() %>% slice_min(X3B)
```

```
## # A tibble: 1 x 4
##   playerID  yearID   X3B    SB
##   <chr>      <int> <int> <int>
## 1 tuckeky01   2020     6     8
```

c. In 1962, Maury Wills had 104 stolen bases, 72 more than the next closest player.

```
stinty %>% group_by(yearID) %>% slice_max(SB, n=2) %>%
  summarize(SBlead = max(SB) - min(SB)) %>% slice_max(SBlead)
```

```
## # A tibble: 1 x 2
##   yearID SBlead
##    <int>  <int>
## 1   1962     72
```

```
stinty %>% group_by(yearID) %>% slice_max(SB, n=2) %>% filter(yearID == 1962)
```

```
## # A tibble: 2 x 4
## # Groups:   yearID [1]
##   playerID  yearID   X3B    SB
##   <chr>      <int> <int> <int>
## 1 willsma01   1962    10   104
## 2 daviswi02   1962    10    32
```

Problem 17 a. Cy Young, `youngcy01`, has the most wins with 511.

```
Pitching %>% group_by(playerID) %>% summarize(wins = sum(W)) %>% slice_max(wins)
```

```
## # A tibble: 1 x 2
##   playerID   wins
##   <chr>     <int>
## 1 youngcy01   511
```

b. Cy Young, `youngcy01`, also has the most losses with 316.

```
Pitching %>% group_by(playerID) %>% summarize(losses = sum(L)) %>% slice_max(losses)
```

```
## # A tibble: 1 x 2
##   playerID  losses
##   <chr>      <int>
## 1 youngcy01    315
```

Problem 18 Gus Weyhing, `weyhigu01`, hit 277 batters.

```
Pitching %>% group_by(playerID) %>% summarize(hbp = sum(HBP)) %>% slice_max(hbp)
```

```
## # A tibble: 1 x 2
##   playerID    hbp
##   <chr>     <int>
## 1 weyhigu01   277
```

Problem 19 There were 2885 complete games in 1884, the most ever.

```
Pitching %>% group_by(yearID) %>% summarize(cg = sum(CG)) %>% slice_max(cg)
```

```
## # A tibble: 1 x 2
##   yearID    cg
##    <int> <int>
## 1   1884  2885
```

Problem 20 The pitcher with the highest win percentage among pitchers who have won 100 games is `spaldal01` (Al Spalding, who played 1871-1878).

```
Pitching %>% group_by(playerID) %>%
  summarize(wins = sum(W), wpct = wins/(wins + sum(L))) %>%
  filter(wins >= 100) %>% slice_max(wpct)
```

```
## # A tibble: 1 x 3
##   playerID   wins  wpct
##   <chr>     <int> <dbl>
## 1 spaldal01   252 0.795
```

Problem 21 The pitcher with the highest K-BB ratio among pitchers who have struck out 500 batters is `ueharko01` (Kohi Uehara, who played 2009-2017).

```
Pitching %>% group_by(playerID) %>% summarize(ks = sum(SO), kpct = ks/sum(BB)) %>%
  filter(ks >= 500) %>% slice_max(kpct)
```

```
## # A tibble: 1 x 3
##   playerID     ks  kpct
##   <chr>     <int> <dbl>
## 1 ueharko01   572  7.33
```

Problem 22 Chris Carpenter leads the list, of which we only show the first six.

```
Pitching %>% filter(teamID == 'SLN',yearID == 2006,IPouts >= 30) %>% arrange(ERA) %>%
  select(playerID, IPouts, ERA) %>% head()
```

```
##    playerID IPouts  ERA
## 1 carpech01    665 3.09
## 2 wainwad01    225 3.12
## 3 kinnejo01     75 3.24
## 4 thompbr01    170 3.34
## 5 isrinja01    175 3.55
## 6 loopebr01    220 3.56
```

Problem 23

```
Pitching %>% group_by(yearID) %>% summarize(balks = sum(BK)) %>% arrange(desc(balks)) %>% slice_
```

```
## # A tibble: 5 x 2
##   yearID balks
##    <int> <int>
## 1   1988   924
## 2   1989   407
## 3   1987   356
## 4   1993   298
## 5   1986   289
```

1988 had more than twice as many balks as any other year in history.

Problem 24 First and second are willimi02 and marmoca01. Mitch Williams, the leader, was known as "Wild Thing". Number two on the list is Carlos Mármol.

```
Pitching %>% group_by(playerID) %>%
  summarize(H = sum(H), BB = sum(BB), outs = sum(IPouts)) %>%
  filter(BB > H) %>% arrange(desc(outs)) %>%
  slice_max(n=2,outs)
```

```
## # A tibble: 2 x 4
##   playerID      H    BB  outs
##   <chr>     <int> <int> <int>
## 1 willimi02   537   544  2074
## 2 marmoca01   385   395  1731
```

Problem @stornnames a. Ana and Claudette were each given to 7 storms.

```
dplyr::storms %>%
  distinct(name, year) %>%
  count(name) %>%
  slice_max(n)
```

```
## # A tibble: 14 x 2
##    name          n
##    <chr>     <int>
##  1 Alberto       7
##  2 Ana           7
##  3 Arthur        7
##  4 Barry         7
##  5 Beryl         7
##  6 Bonnie        7
##  7 Chantal       7
##  8 Chris         7
##  9 Claudette     7
## 10 Danielle      7
## 11 Debby         7
## 12 Edouard       7
## 13 Emily         7
## 14 Ernesto       7
```

b. 1995 and 2005 both had 21 named storms.

```
dplyr::storms %>%
  distinct(name, year) %>%
  count(year) %>%
  slice_max(n)
```

```
## # A tibble: 1 x 2
##    year     n
##   <dbl> <int>
## 1  2020    26
```

c. Felix (1995) was the second strongest Felix, after (2007), and had maximum wind speed of 120.

```
dplyr::storms %>%
  group_by(name, year) %>%
  summarize(max_wind = max(wind)) %>%
  summarize(second = nth(max_wind, n = 2, order_by = desc(max_wind))) %>%
  slice_max(second, n = 7)
```

```
## `summarise()` has grouped output by 'name'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 7 x 2
##   name    second
##   <chr>    <int>
## 1 Felix      120
## 2 Harvey     115
## 3 Emily      110
## 4 Karl       110
## 5 Edouard    105
## 6 Gustav      105
## 7 Kate        105
```

Problem 26 a. Approximate values are $n = 18$ and $p = 17/18$. b.

```
colors <- factor(c("Blue", "Orange" ,"Green", "Yellow", "Red", "Brown"))
dd <- data.frame(bag = rep(1:200, times = rbinom(200, 18, 17/18)))
dd$color <- sample(colors, size = nrow(dd), replace = T, prob = c(25, 25, 12.5, 12.5, 12.5, 12.5
dd %>%
  group_by(bag, color, .drop = FALSE) %>%
  count() %>%
  pivot_wider(id = bag, names_from = color, values_from = n)
```

c. The probability is essentially 1. It is very unlikely that any two of the 200 bags will be identical in color distribution.

```
num_bags <- 200
sim_data <- replicate(1000, {
  dd <- data.frame(bag = rep(1:num_bags, times = rbinom(num_bags, 18, 17/18)))
  dd$color <- sample(colors, size = nrow(dd),
                     replace = T,
                     prob = c(25, 25, 12.5, 12.5, 12.5, 12.5))
  dd %>%
    group_by(bag, color, .drop = FALSE) %>%
    count() %>%
    pivot_wider(id = bag, names_from = color, values_from = n) %>%
    distinct() %>%
    nrow() == num_bags
})
mean(sim_data)
```

stringr

```
library(stringr)
```

Problem 27

```
sum(str_detect(words,"ff"))
```

```
## [1] 12
```

```
mean(str_detect(words,"^s"))
```

```
## [1] 0.1214286
```

Problem 28

```
mean(str_detect(sentences,'the'))
```

```
## [1] 0.5666667
```

```
mean(str_detect(sentences,'[Tt]he'))
```

```
## [1] 0.7416667
```

```
mean(word(sentences,1) == 'The')
```

```
## [1] 0.3638889
```

```
tibble(sentences) %>% filter(str_detect(sentences, "x"),str_detect(sentences, "q"))
```

```
## # A tibble: 1 x 1
##   sentences
##   <chr>
## 1 The quick fox jumped on the sleeping cat.
```

```
tibble(sentences) %>% count(word(sentences,-1)) %>% slice_max(n)
```

```
## # A tibble: 4 x 2
##   `word(sentences, -1)`     n
##   <chr>                 <int>
## 1 grass.                    5
## 2 side.                     5
## 3 wall.                     5
## 4 water.                    5
```

Problem 29

```
sum(str_detect(fruit,"berry"))
```

```
## [1] 14
```

```
str_remove(fruit[str_detect(fruit,"fruit")],"fruit")
```

```
## [1] "bread"   "dragon"  "grape"   "jack"    "kiwi "   "passion" "star "
## [8] "ugli "
```

babynames

```
library(babynames)
```

Problem 30 There were 325 popular girls names in 2015. 120 of these end in ‘a’, about 37%.

```
babynames %>% filter(year == 2015, n >= 1000, sex=='F') %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   325
```

```
babynames %>% filter(year == 2015, n >= 1000, sex=='F', str_detect(name,"a$")) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   120
```

Problem 31

```
babynames %>% filter(year == 2013, n > 100) %>%
  group_by(name) %>%
  summarize(genders = n(), popularity = sum(n), gender_ratio = min(n)/max(n)) %>%
  filter(genders > 1) %>%
  filter(gender_ratio > 0.8) %>%
  arrange(desc(popularity)) %>% slice_max(popularity, n=5)
```

```
## # A tibble: 5 x 4
##   name    genders popularity gender_ratio
##   <chr>     <int>      <int>        <dbl>
## 1 Charlie       2       2886        0.849
## 2 Dakota        2       1985        0.833
## 3 Justice       2       1293        0.831
## 4 Milan         2        965        0.986
## 5 Lennon        2        567        0.929
```

Problem 32

All of this code shown should actually run, but is temporarily turned off because of a problem with the phonics install.

Answers may vary. This is really meant to be an open-ended exercise.

Ariana and Arianna (Note that Gabriella and Gabriela are close!)

```
babynames %>%
  filter(year == 2003, sex == "F") %>%
  mutate(phonetic = phonics::metaphone(name)) %>%
  group_by(phonetic) %>%
  filter(n() >= 2) %>%
  top_n(2, n) %>%
  filter(max(n) < 1.2 * min(n)) %>%
  summarize(tot = sum(n),
            names = str_c(name, collapse = " ")) %>%
  arrange(desc(tot))
```

To be sure, we should check all of the phonetic codes with more total names than the 7626 that we chose.

```
babynames %>%
  filter(year == 2003, sex == "F") %>%
  mutate(phonetic = phonics::metaphone(name)) %>%
  filter(phonetic == "KBRL")
```

Gabriela and Gabriella together are 7381, just less than Ariana Arianna.

Jaden Jayden Jadyn

```
babynames %>%
  filter(year == 2003, sex == "F") %>%
  mutate(phonetic = phonics::metaphone(name)) %>%
  group_by(phonetic) %>%
  filter(n() >= 3) %>%
  top_n(3, n) %>%
  filter(max(n) < 1.5 * min(n)) %>%
  summarize(tot = sum(n),
            names = str_c(name, collapse = " ")) %>%
  arrange(desc(tot))
```

Again, we should check JLN, KR and KRBL to see if there any triples that are more common that Jaden.

```
babynames %>%
  filter(year == 2003, sex == "F") %>%
  mutate(phonetic = phonics::metaphone(name)) %>%
  filter(phonetic %in% c("KBRL", "JLN", "KR"), n > 500) %>%
  arrange(phonetic)
```

Problem 33 a. For HTH it is 10. b. For HHT is is 8. c. For TTTT it is 30. d. For THHH it is 16.

Here is the code for HTH only. For the others, 100 may not be enough flips for the pattern to occur, producing NA. Increase as needed.

```
HTH <- replicate(100000, {
  flips <- paste(sample(c('H','T'), 100, replace=TRUE),collapse='')
  str_locate(flips,'HTH')[2]
})
mean(HTH)
```

Structure of data

```
library(tidyr)
```

Problem 34

a. Jay-Z had five tracks on the charts in 2000.
b. “Independent Women” by Destiny’s Child spent 11 weeks at #1.

```
billboard %>% group_by(artist) %>% summarize(count = n()) %>% slice_max(count)
```

```
## # A tibble: 1 x 2
##   artist count
##   <chr>  <int>
## 1 Jay-Z      5
```

```
billboard %>% pivot_longer(cols = wk1:wk76, names_to = "week", values_to = "rank") %>%
  filter(rank == 1) %>%
  group_by(track) %>% summarize(count = n()) %>% slice_max(count)
```

```
## # A tibble: 1 x 2
##   track                   count
##   <chr>                   <int>
## 1 Independent Women Pa...    11
```

Problem 35

```
ss <- fosdata::scrabble
ss %>% group_by(piece) %>%
  summarize(n = n()) %>%
  print(n = 27)
```

```
## # A tibble: 27 x 2
##    piece     n
##    <chr> <int>
##  1 A         9
##  2 B         2
##  3 blank     2
##  4 C         2
##  5 D         4
##   6 E        12
##   7 F         2
##   8 G         3
##   9 H         2
##  10 I         9
##  11 J         1
##  12 K         1
##  13 L         4
##  14 M         2
##  15 N         6
##  16 O         8
##  17 P         2
##  18 Q         1
##  19 R         6
##  20 S         4
##  21 T         6
##  22 U         4
##  23 V         2
##  24 W         2
##  25 X         1
##  26 Y         2
##  27 Z         1
```

```
lf <- fosdata::letter_frequency %>%
  select(letter, english) %>%
  filter(letter %in% letters) %>%
  mutate(letter = toupper(letter))

new_dat <- left_join(ss, lf, by = c("piece"= "letter")) %>%
  group_by(piece) %>%
  summarize(points = points[1],
            frequency = english[1],
            number = n()) %>%
  filter(piece != "blank")

mutate(new_dat, diff = abs(number/100 - frequency)) %>%
  arrange(desc(diff))
```

```
## # A tibble: 26 x 5
##    piece points frequency number    diff
##    <chr>  <int>     <dbl>  <int>   <dbl>
##  1 H          4   0.0609       2 0.0409
##  2 T          1   0.0906       6 0.0306
##  3 S          1   0.0633       4 0.0233
##  4 I          1   0.0697       9 0.0203
##  5 U          1   0.0276       4 0.0124
##  6 V          4   0.00978      2 0.0102
##  7 G          2   0.0202       3 0.00985
##  8 Z         10   0.00074      1 0.00926
##  9 Q         10   0.00095      1 0.00905
## 10 X          8   0.0015       1 0.0085
## # ... with 16 more rows
```

Problem 36

```
scot_clean <- fosdata::scotland_births %>%
  pivot_longer(!age, names_to = "year", values_to = "births") %>%
  mutate(year = as.integer(str_remove(year,"x")),
         births = as.integer(births))
scot_clean %>% filter(age <= 20) %>%
  group_by(year) %>% summarize(births = sum(births)) %>%
  slice_max(births)
```

```
## # A tibble: 1 x 2
##    year births
##   <int>  <int>
## 1  1967  15457
```

Problem 37

```
fosdata::world_cup %>%
  filter(competition == "2015 FIFA Women's World Cup") %>%
  pivot_longer(cols = contains('team'), names_to = "which", values_to = "team") %>%
  mutate(score = ifelse(which == "team_1", score_1, score_2)) %>%
  select(team,score) %>%
  group_by(team) %>% summarize(score = sum(score)) %>%
  arrange(desc(score)) %>% print(n=24)
```

```
## # A tibble: 24 x 2
##    team        score
##    <chr>       <dbl>
##  1 Germany        20
##  2 USA            14
##  3 Japan          11
##  4 Switzerland    11
##  5 England        10
##  6 France         10
##  7 Cameroon        9
##  8 Norway          9
##  9 Australia       5
## 10 Sweden          5
## 11 Brazil          4
## 12 Canada          4
## 13 China           4
## 14 Colombia        4
## 15 South Korea     4
## 16 Costa Rica      3
## 17 Ivory Coast     3
## 18 Netherlands     3
## 19 Nigeria         3
## 20 Thailand        3
## 21 Mexico          2
## 22 New Zealand     2
## 23 Spain           2
## 24 Ecuador         1
```

Problem 38

```
apply(replicate(10000,sort(runif(5))), 1, mean)
```

```
## [1] 0.1675096 0.3350106 0.5008730 0.6683018 0.8350052
```

```
1:5/6   # for comparison
```

```
## [1] 0.1666667 0.3333333 0.5000000 0.6666667 0.8333333
```

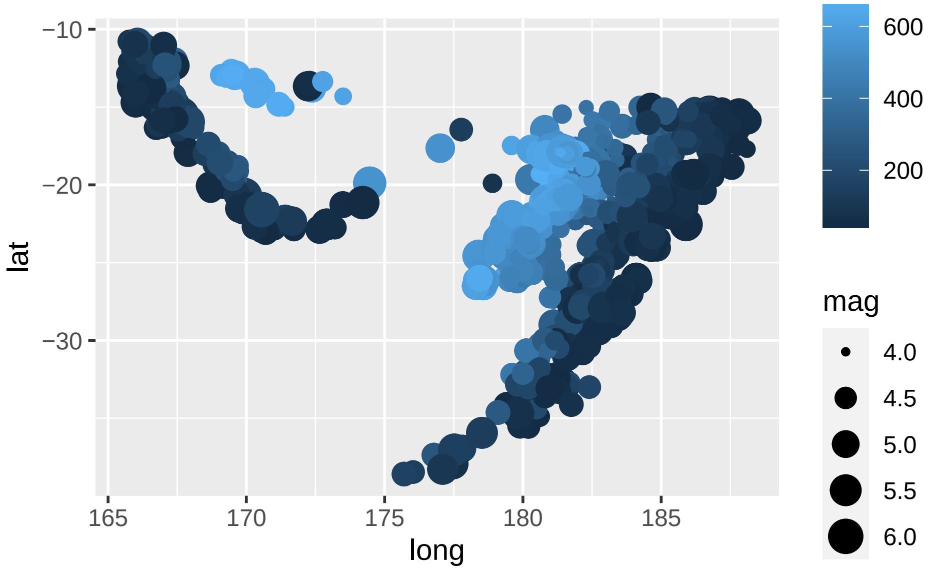
7

Data Visualization with ggplot – Solutions

```
library(ggplot2)
library(dplyr)
library(tidyr)
```

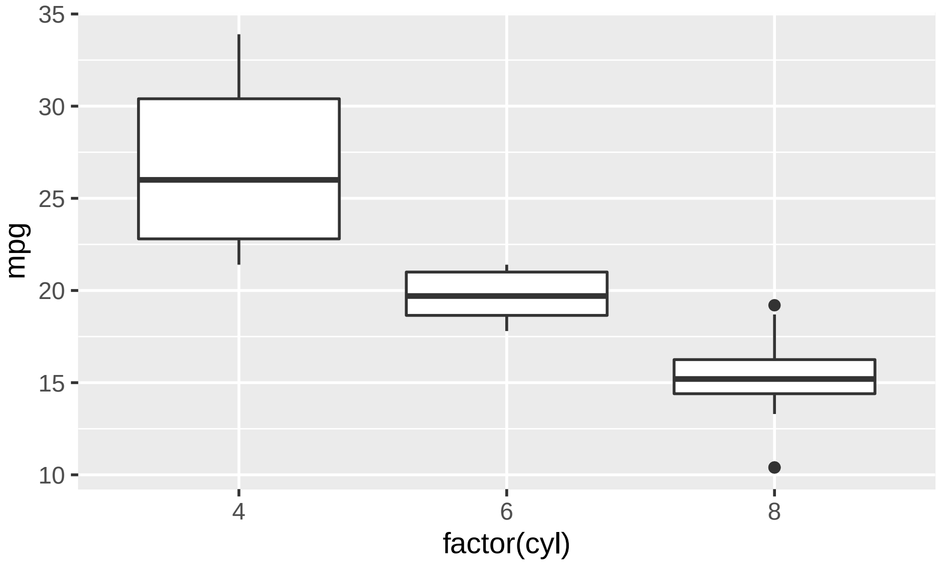
Problem 1

```
ggplot(quakes, aes(x=long, y=lat, color=depth, size=mag)) + geom_point()
```



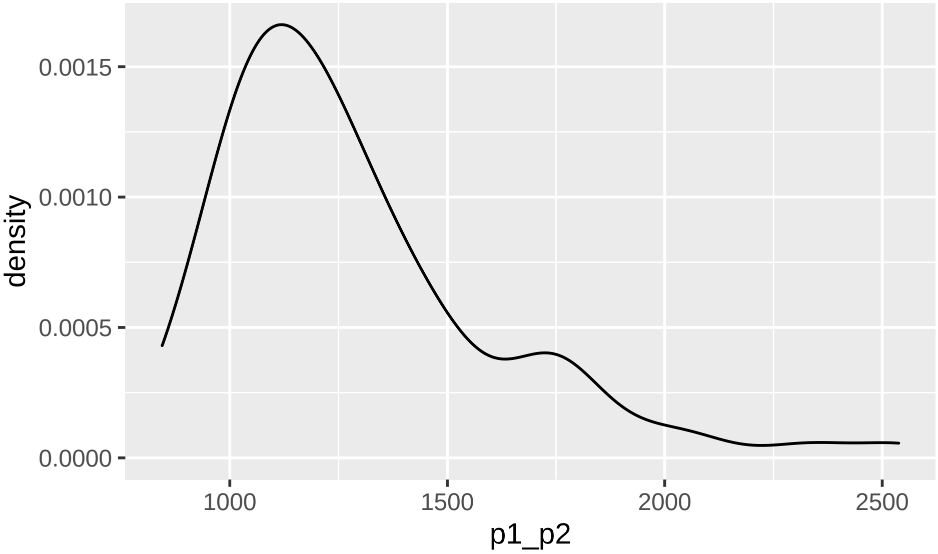
Problem 2

```
ggplot(mtcars, aes(x=factor(cyl), y=mpg)) + geom_boxplot()
```



Problem 3

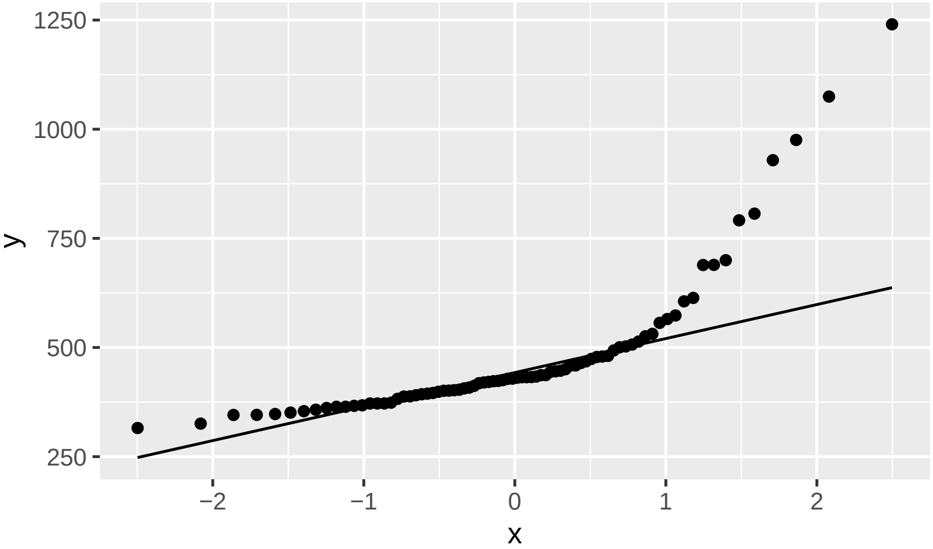
```
fosdata::brake %>% ggplot(aes(x=p1_p2)) + geom_density()
```



a. The data is skew right. b. The most likely time is about 1100 msec.

Problem 4

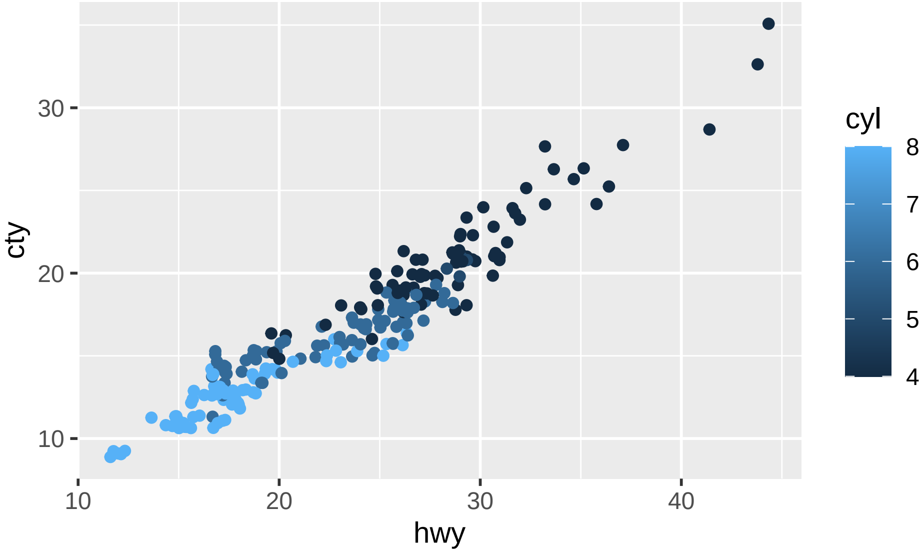
```
fosdata::brake %>% ggplot(aes(sample=latency_p1)) + geom_qq() + geom_qq_line()
```



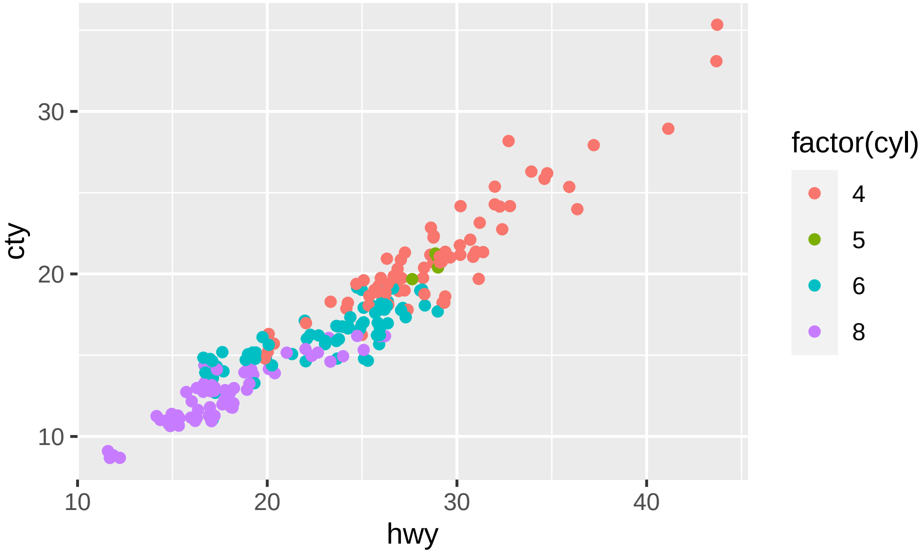
The data is right skew.

Problem 5

```
mpg %>% ggplot(aes(x=hwy, y=cty, color=cyl)) + geom_jitter()
```

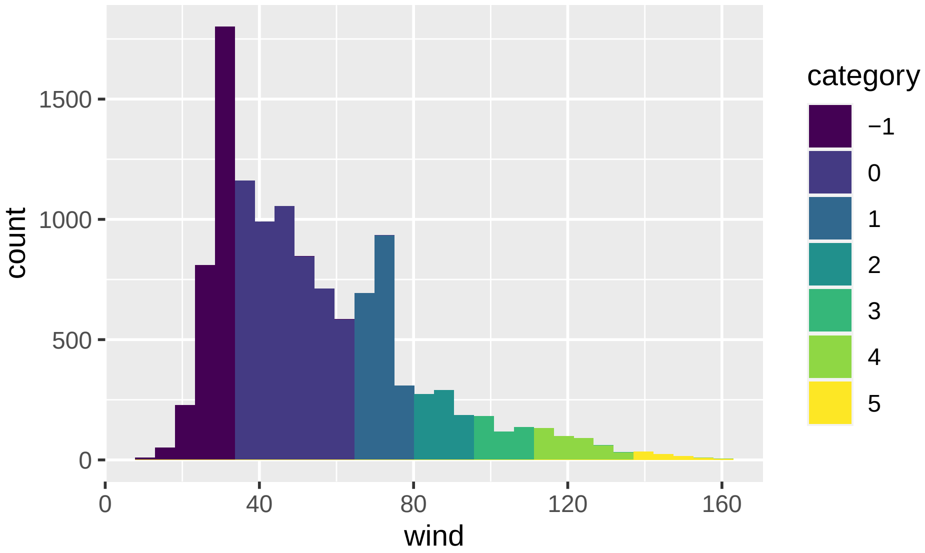


```
mpg %>% ggplot(aes(x=hwy, y=cty, color=factor(cyl))) + geom_jitter()
```

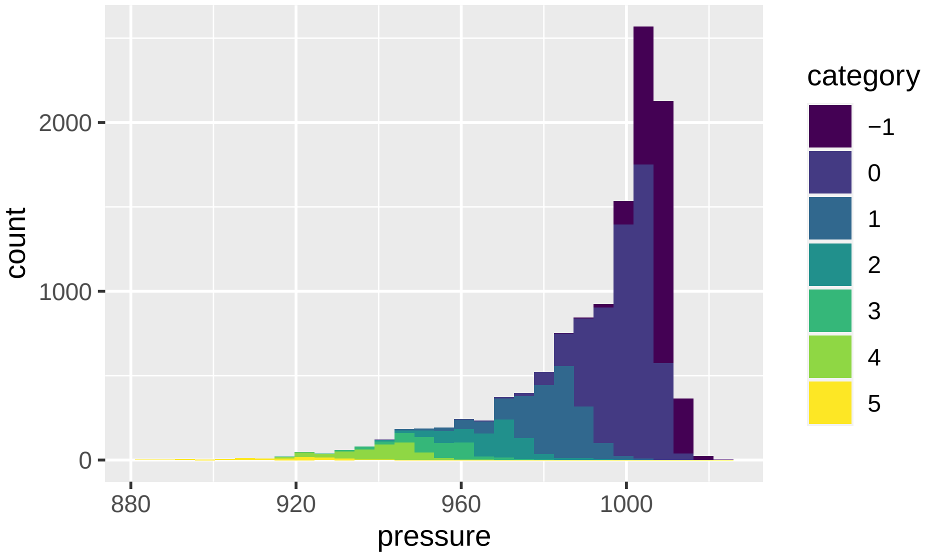


Problem 6

```
storms %>% ggplot(aes(x = wind, fill=category)) + geom_histogram()
```



```
storms %>% ggplot(aes(x = pressure, fill=category)) + geom_histogram()
```

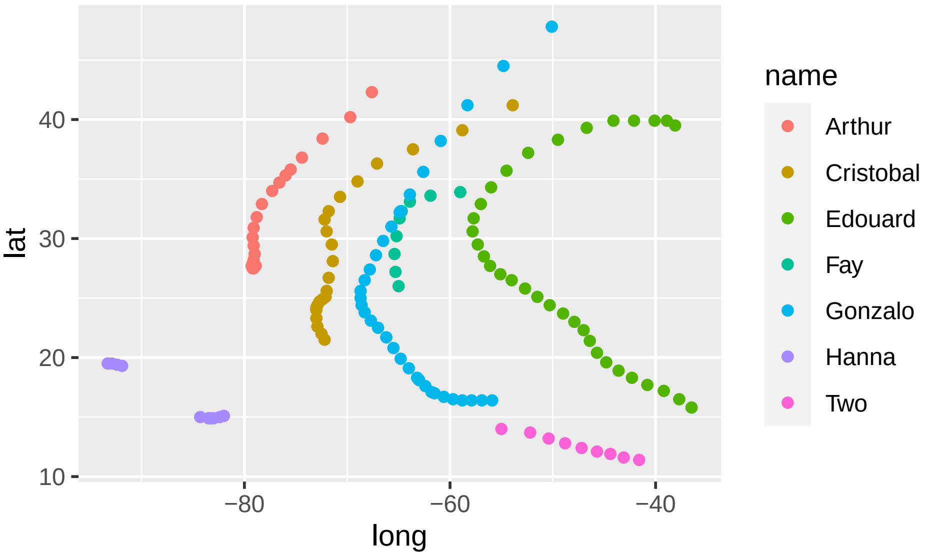


c. The `wind` data is right skew. The `pressure` data is left skew.

d. The `category` variable is an ordered factor. This causes ggplot to choose a color scale that has discrete colors but in a progression.

Problem 7

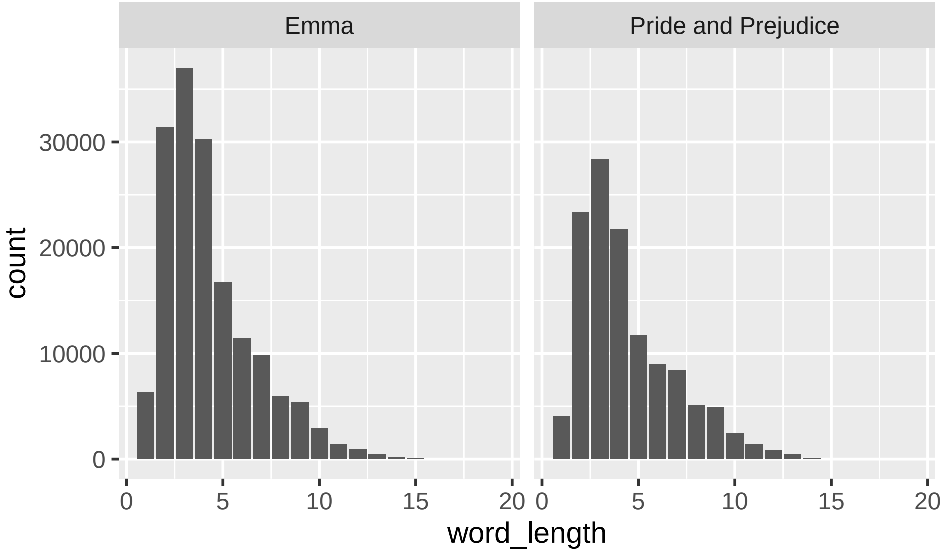
```
storms %>% filter(year ==  2014) %>% ggplot(aes(x = long, y = lat, color=name)) + geom_point()
```



Gonzalo made it furthest North.

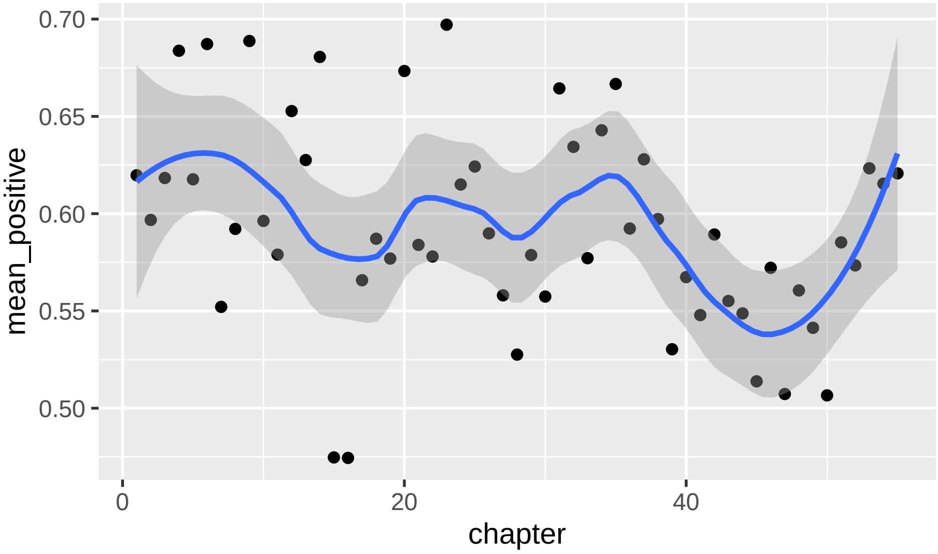
Problem 8

```
fosdata::austen %>%
  ggplot(aes(x = word_length)) +
  geom_bar() +
  facet_wrap(~novel)
```



Problem 9

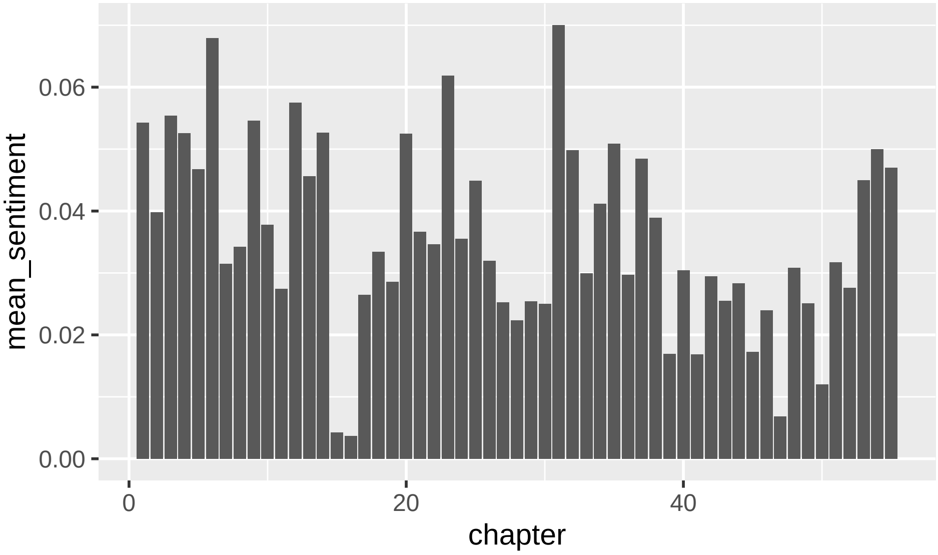
```
fosdata::austen %>%
  filter(novel == "Emma", sentiment_score != 0) %>%
  group_by(chapter) %>%
  summarize(mean_positive = mean(sentiment_score > 0)) %>%
  ggplot(aes(x = chapter, y = mean_positive)) +
  geom_point() +
  geom_smooth(span = .4)
```



`geom_smooth` gives a smoothed version of a line through the points, where `geom_line` doesn't give any extra information over just the points. There is no chapter 3.5, so I would choose `geom_smooth`.

Problem 10

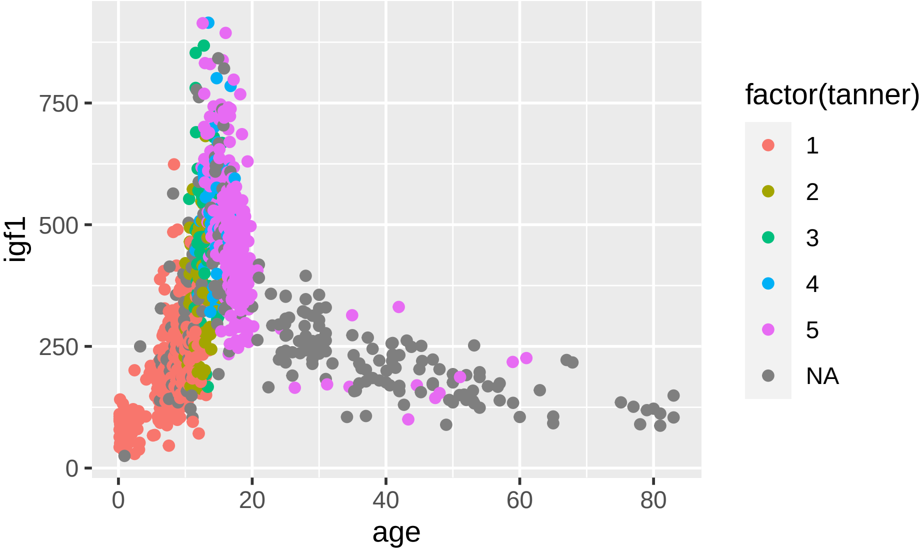
```
fosdata::austen %>%
  filter(novel == "Emma") %>%
  group_by(chapter) %>%
  summarize(mean_sentiment = mean(sentiment_score)) %>%
  ggplot(aes(chapter, y = mean_sentiment)) +
  geom_bar(stat = "identity")
```



Problem 11

```
ISwR::juul %>% ggplot(aes(x=age, y=igf1, color=factor(tanner))) + geom_point()
```

```
## Warning: Removed 326 rows containing missing values (geom_point).
```



Neither `geom_line` or `geom_smooth` works well with this data.

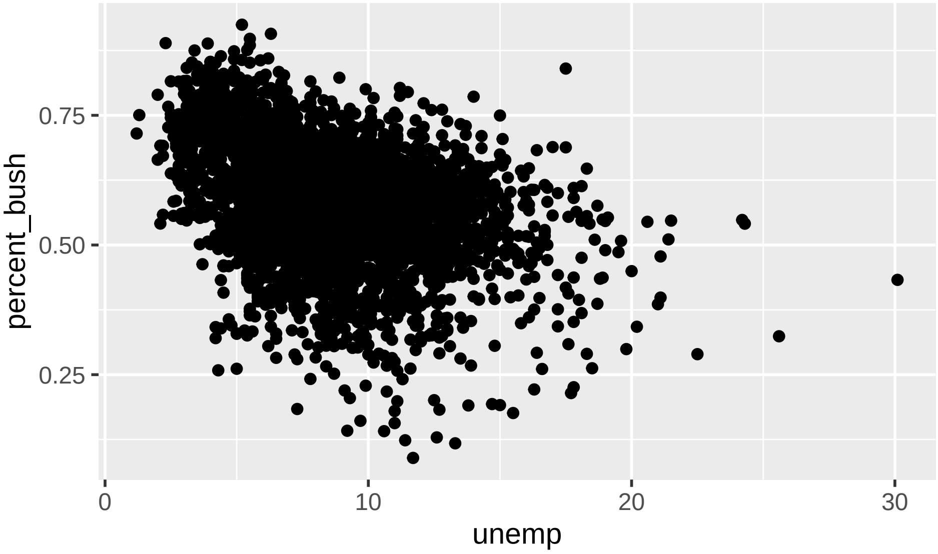
Problem 12

```
pres_election <- fosdata::pres_election
library(mapproj)
library(stringr)
data("unemp")
combined_data <- left_join(pres_election, unemp, by = c("FIPS" = "fips"))
head(combined_data)
```

```
##   year   state state_po  county FIPS      candidate      party candidatevotes
## 1 2000 Alabama       AL Autauga 1001        Al Gore   democrat           4942
## 2 2000 Alabama       AL Autauga 1001 George W. Bush republican          11993
## 3 2000 Alabama       AL Autauga 1001    Ralph Nader      green            160
## 4 2000 Alabama       AL Autauga 1001          Other       <NA>            113
## 5 2000 Alabama       AL Baldwin 1003        Al Gore   democrat          13997
## 6 2000 Alabama       AL Baldwin 1003 George W. Bush republican          40872
##   totalvotes   pop unemp
## 1      17208 23288   9.7
## 2      17208 23288   9.7
## 3      17208 23288   9.7
## 4      17208 23288   9.7
## 5      56480 81706   9.1
## 6      56480 81706   9.1
```

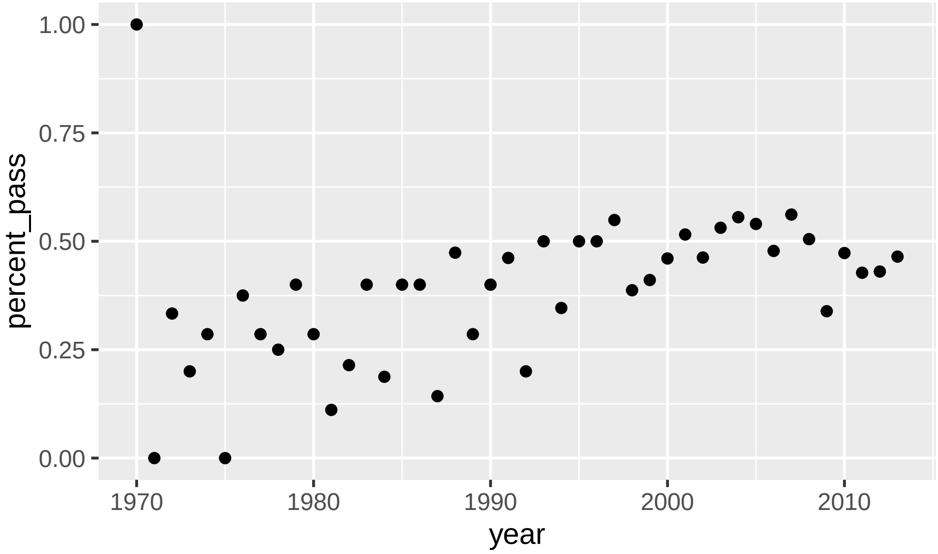
```
combined_data %>%
  filter(year == 2000) %>%
  filter(str_detect(candidate, "Bush")) %>%
  mutate(percent_bush = candidatevotes/totalvotes) %>%
  ggplot(aes(x = unemp, y = percent_bush)) +
  geom_point()
```

```
## Warning: Removed 42 rows containing missing values (geom_point).
```



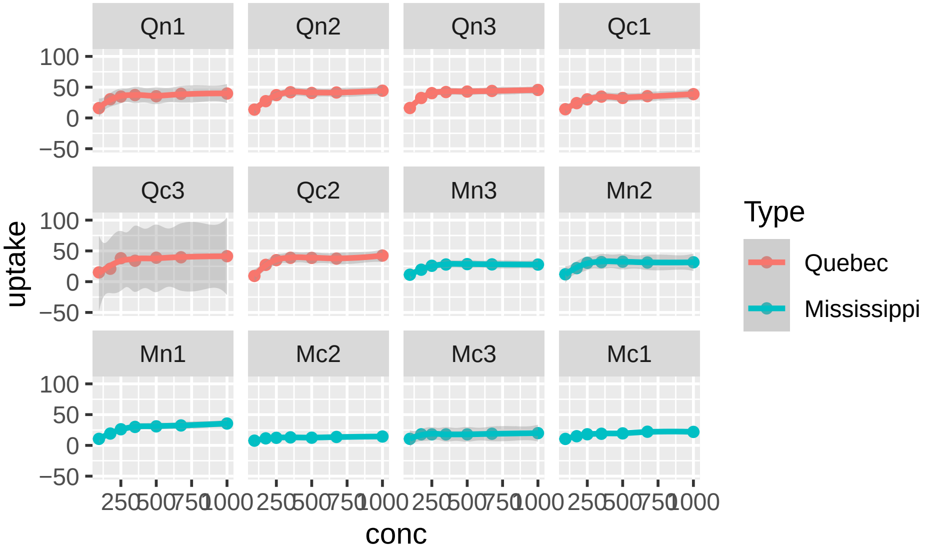
Problem 13

```
bechdel <- fosdata::bechdel
bechdel %>%
  group_by(year) %>%
  summarize(percent_pass = mean(binary == "PASS")) %>%
  ggplot(aes(x = year, y = percent_pass)) +
  geom_point()
```



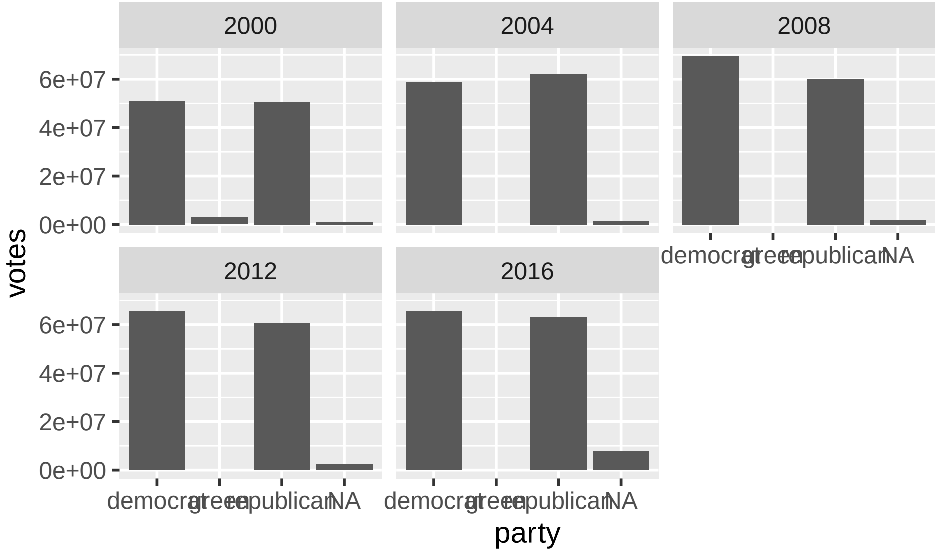
Problem 14

```
ggplot(CO2, aes(x = conc, y = uptake, color = Type)) +
   geom_point() +
   geom_smooth() + facet_wrap(vars(Plant))
```



Problem 15

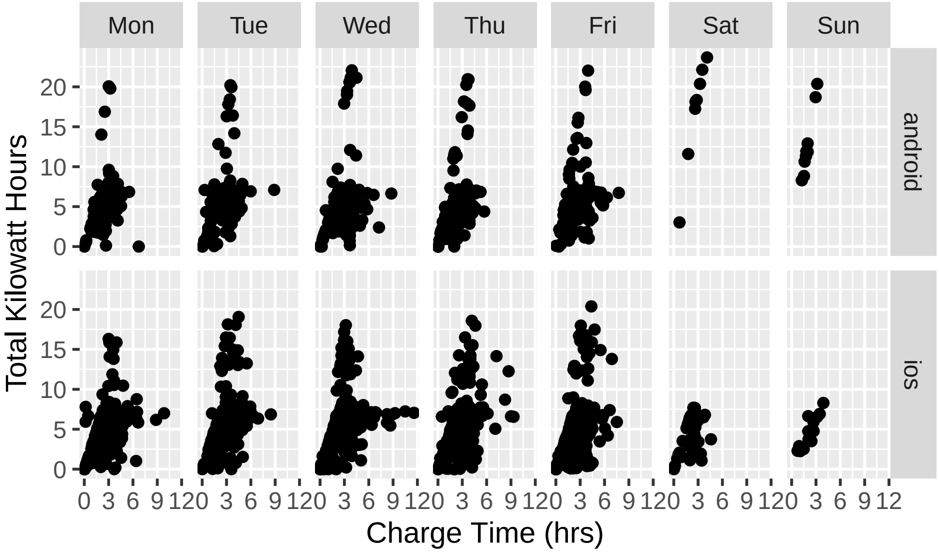
```
fosdata::pres_election %>%
  group_by(year,party) %>%
  summarize(votes = sum(candidatevotes, na.rm=TRUE)) %>%
  ggplot(aes(x=party,y=votes)) + geom_col() + facet_wrap(vars(year))
```



Problem 16

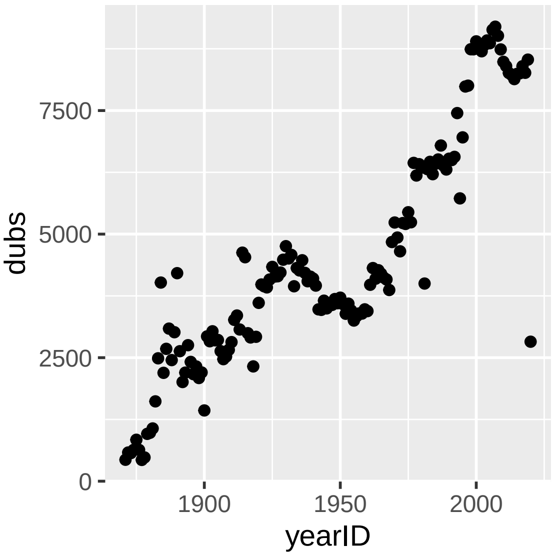
We remove the outlier that has a charge time of over 40 hours.

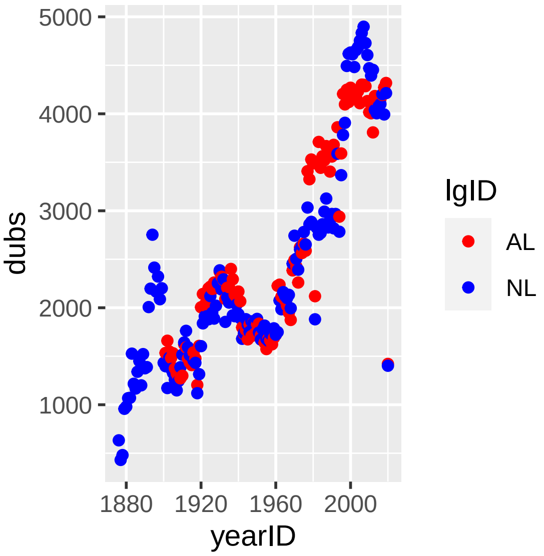
```
cc <- fosdata::ecars
cc <- janitor::clean_names(cc)

cc <- cc %>%
  mutate(weekday = factor(weekday,levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")))

cc %>%
  filter(!str_detect(platform, "web"), charge_time_hrs < 40) %>%
  ggplot(aes(x = charge_time_hrs, y = kwh_total)) +
  geom_point() +
  facet_grid(platform ~ weekday) +
  labs(x = "Charge Time (hrs)",
       y = "Total Kilowatt Hours")
```



Problem 17

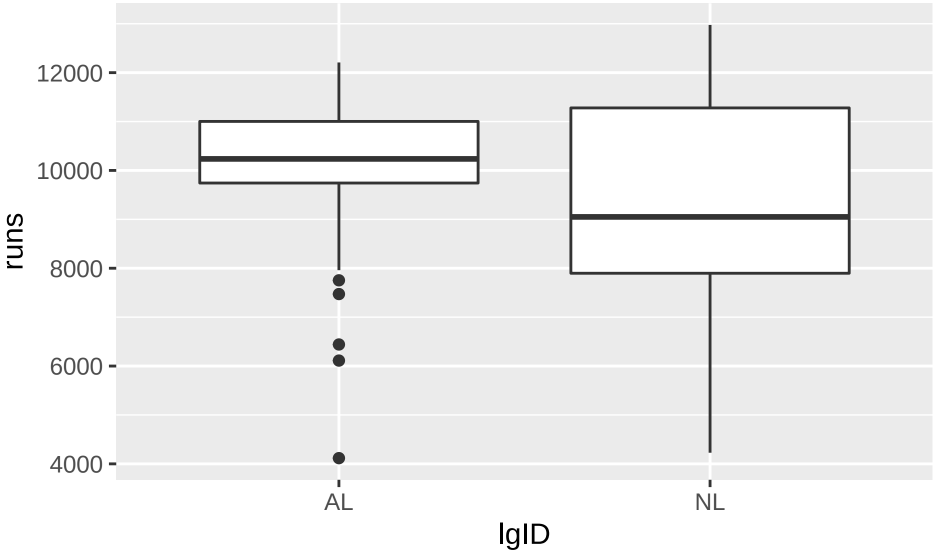
```
library(Lahman)
# a)
Batting %>% group_by(yearID) %>% summarize(dubs=sum(X2B)) %>%
  ggplot(aes(x = yearID, y = dubs)) + geom_point()
# b)
Batting %>% filter(lgID %in% c('AL','NL')) %>%
  group_by(yearID, lgID) %>% summarize(dubs=sum(X2B)) %>%
  ggplot(aes(x = yearID, y = dubs, color=lgID)) +
  scale_color_manual(values = c("red", "blue")) + geom_point()
```





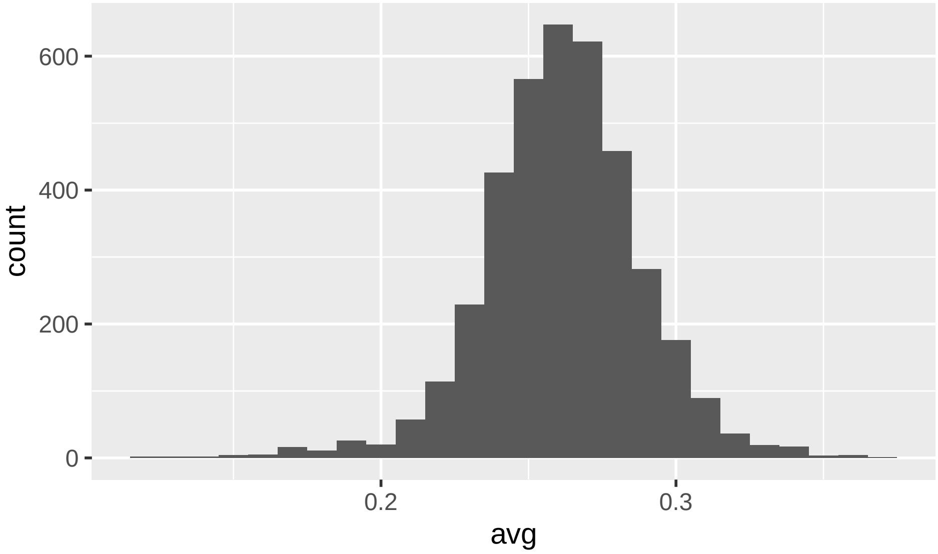
Problem 18

```
Batting %>% filter(yearID >= 1969) %>% group_by(yearID,lgID) %>% summarize(runs=sum(R)) %>%
  ggplot(aes(x = lgID, y = runs)) + geom_boxplot()
```



Problem 19

```
Batting %>% group_by(playerID) %>%
  summarize(avg = sum(H)/sum(AB), abs = sum(AB)) %>%
  filter(abs >= 1000) %>%
  ggplot(aes(x = avg)) + geom_histogram(binwidth = 0.01)
```

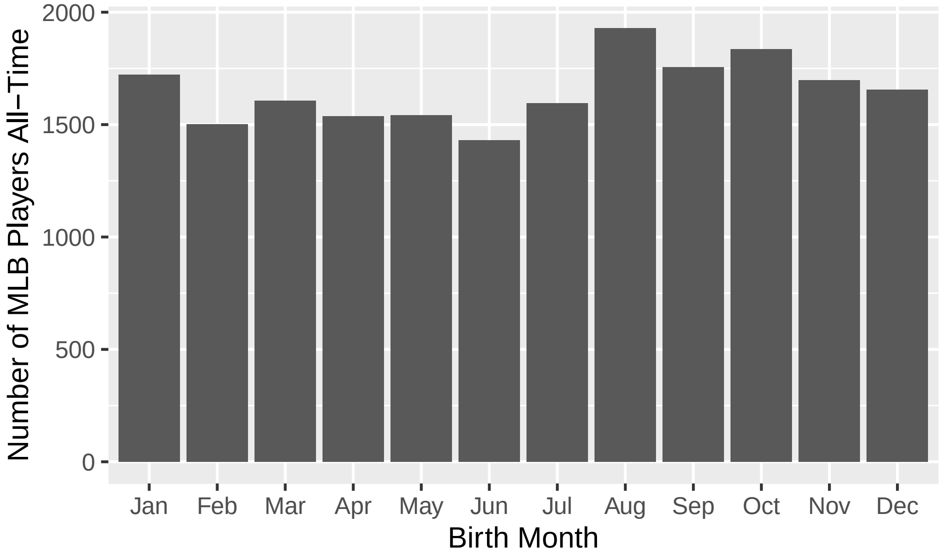


Problem 20

a.

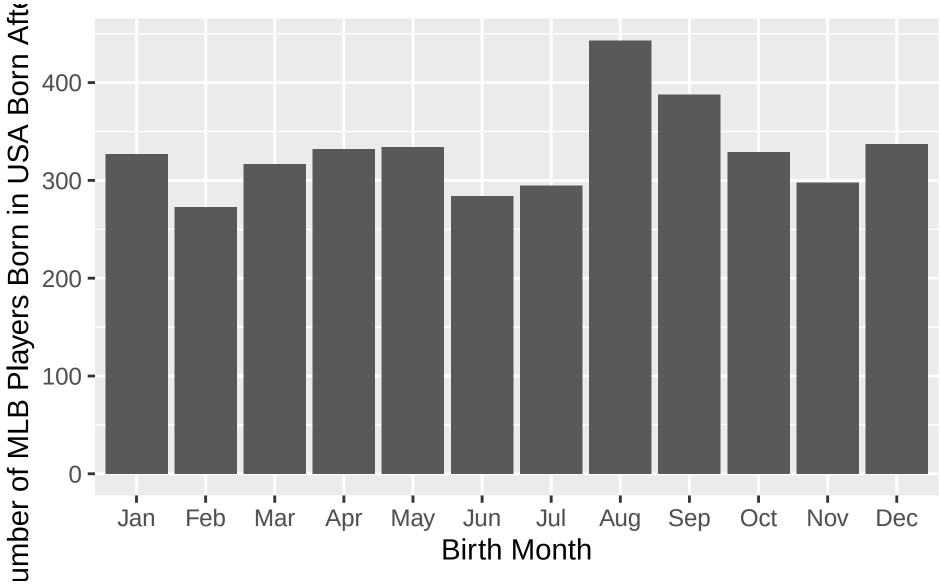
```
Lahman::People %>%
  janitor::clean_names() %>%
  ggplot(aes(x = birth_month)) +
  geom_bar(stat = "count") +
  scale_x_discrete(limits = factor(1:12), labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "
  labs(x = "Birth Month",
       y = "Number of MLB Players All-Time")
```

```
## Warning: Removed 282 rows containing non-finite values (stat_count).
```



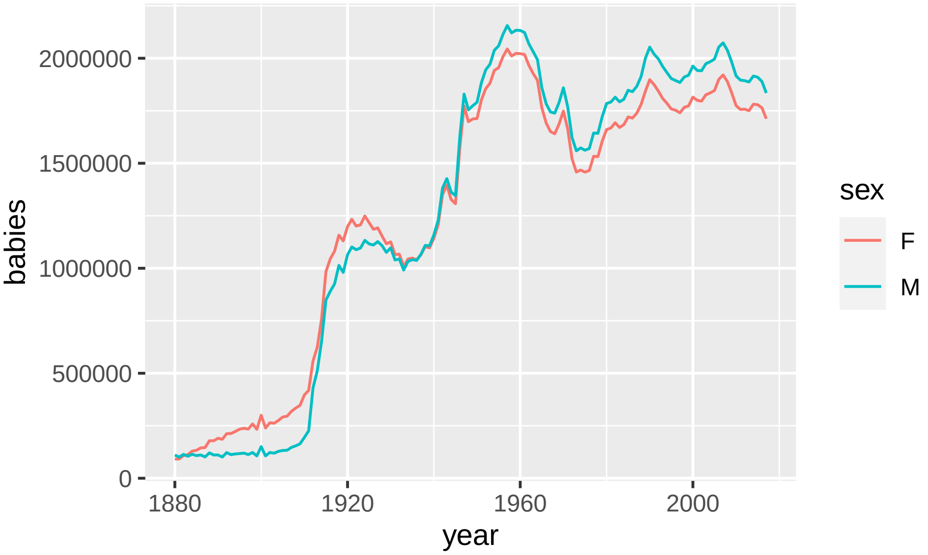
b.

```
Lahman::People %>%
  janitor::clean_names() %>%
  filter(birth_country == "USA", birth_year > 1970) %>%
  ggplot(aes(x = birth_month)) +
  geom_bar(stat = "count") +
  scale_x_discrete(limits = factor(1:12), labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "
  labs(x = "Birth Month",
       y = "Number of MLB Players Born in USA Born After 1970")
```



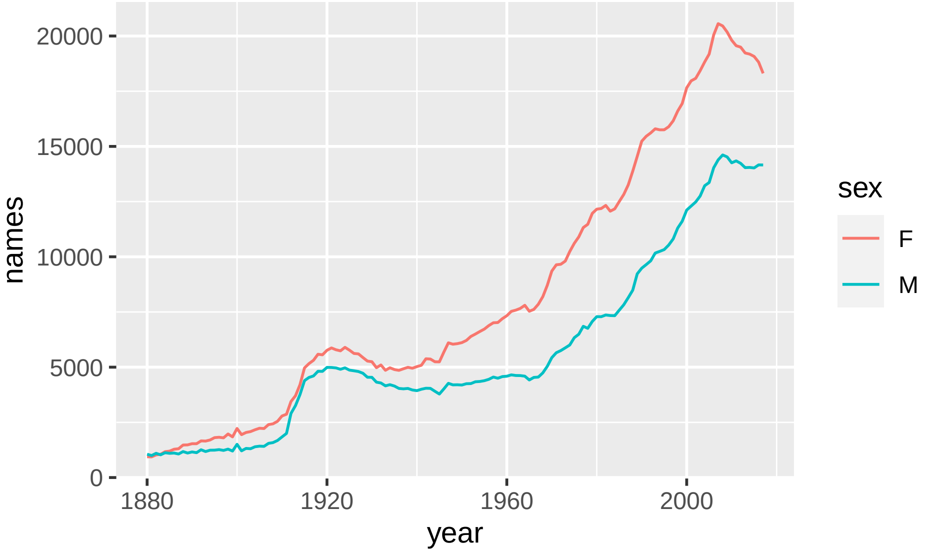
Problem 21

```
library(babynames)
babynames %>% group_by(year,sex) %>% summarize(babies = sum(n)) %>%
  ggplot(aes(x=year,y=babies,color=sex)) + geom_line()
```



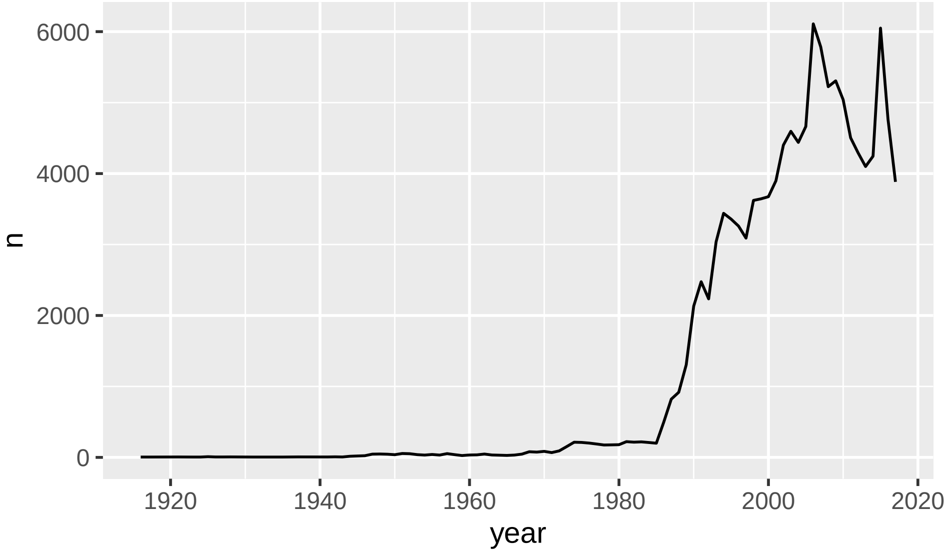
Problem 22

```
babynames %>% group_by(year,sex) %>% summarize(names = n()) %>%
  ggplot(aes(x=year,y=names,color=sex)) + geom_line()
```



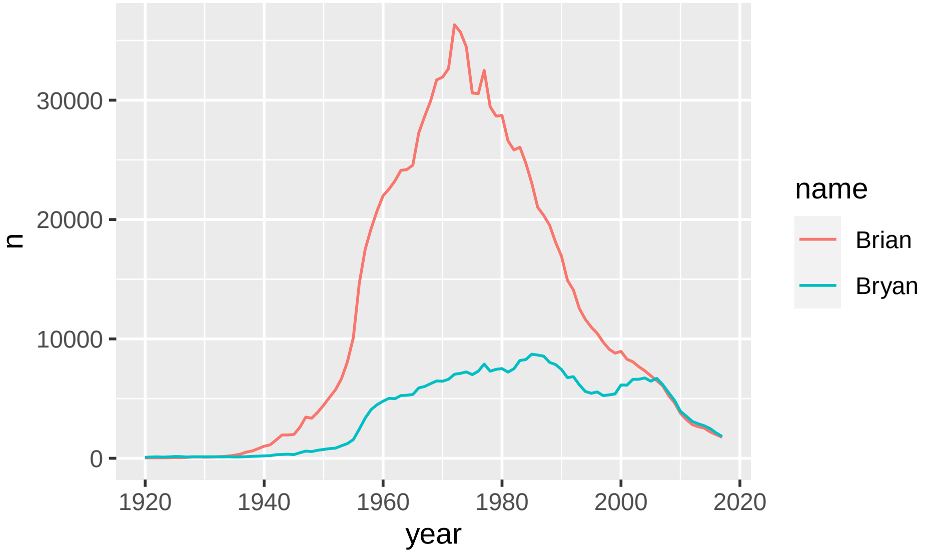
Problem 23 Here is the chart for Alexa Your name may vary.

```
babynames %>% filter(name=="Alexa", sex=="F") %>%
  ggplot(aes(x=year,y=n)) + geom_line()
```



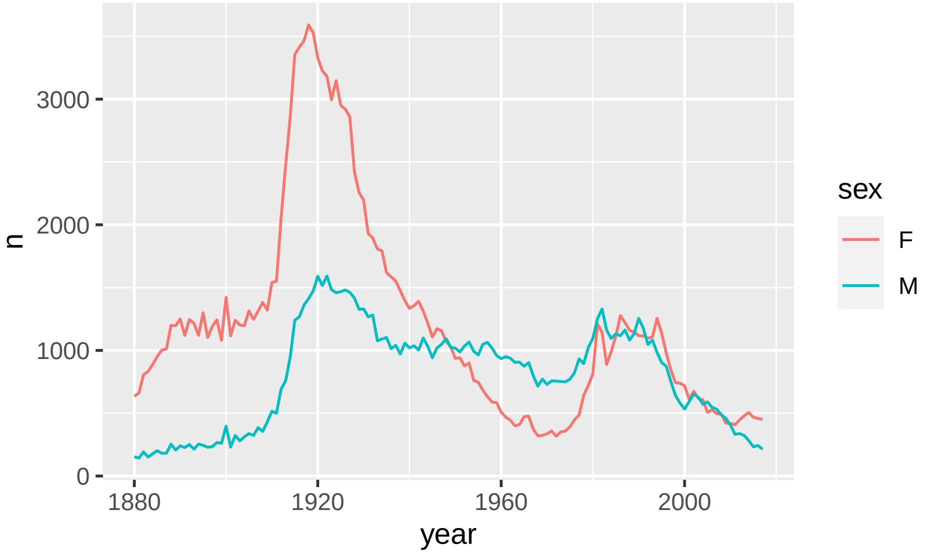
Problem 24

```
babynames %>% filter(name %in% c("Bryan","Brian"), sex=="M", year >= 1920) %>%
  ggplot(aes(x=year,y=n,color=name)) + geom_line()
```



Problem 25

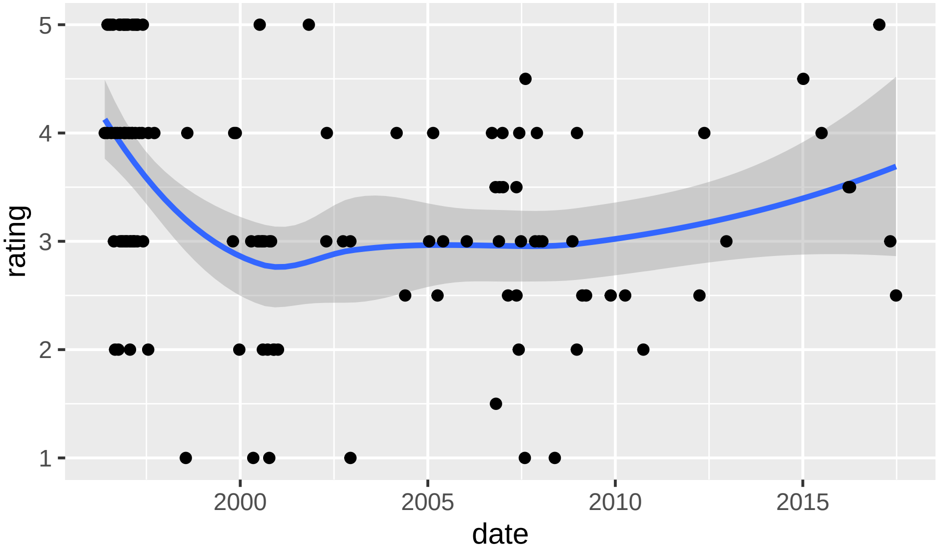
```
babynames %>% filter(name == "Jessie") %>%
  ggplot(aes(x=year,y=n,color=sex)) + geom_line()
```



More female than male Jessie from 1880 to 1950. More male than female Jessie from 1950 to 1980. About the same male and female Jessie since 1980.

Problem 26

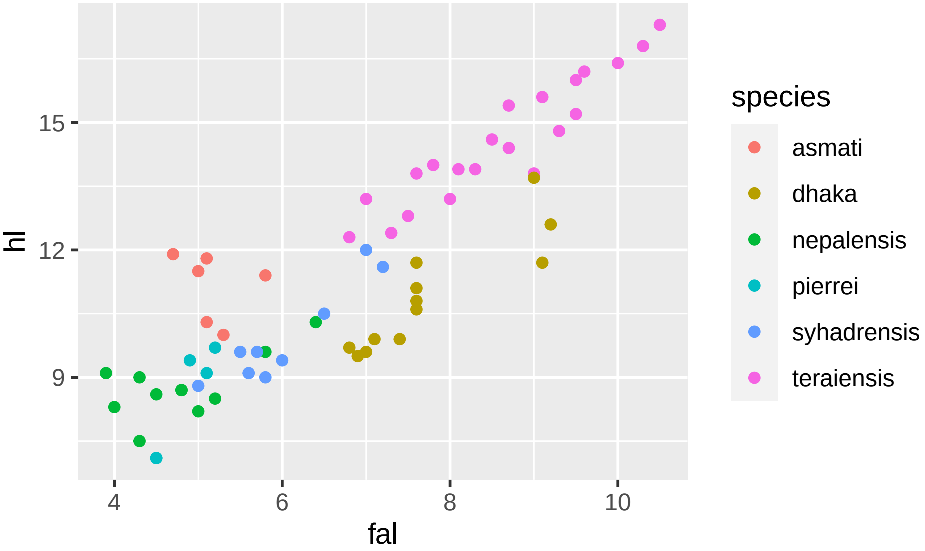
```
fosdata::movies %>%
  filter(title == "Twister (1996)") %>%
  mutate(date = lubridate::as_datetime(timestamp)) %>%
  ggplot(aes(x = date, y = rating)) +
  geom_smooth() +
  geom_point()
```



Problem 27

The dhaka species of frog does not appear to have the same relationship betweem forearm length and head length distance that the other frogs have, which is evidence that this is a new species of frog.

```
ff <- fosdata::frogs
ggplot(ff, aes(x = fal, y = hl, color = species)) +
  geom_point()
```



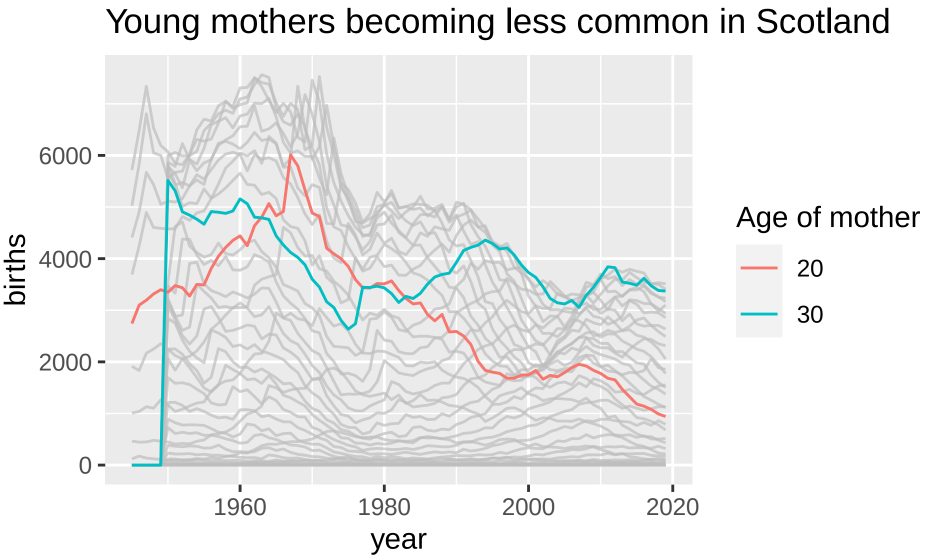
Problem 28

```
library(stringr)
scot_clean <- fosdata::scotland_births %>%
  pivot_longer(!age, names_to = "year", values_to = "births") %>%
  mutate(year = as.integer(str_remove(year,"x")),
         births = as.integer(births))
```

```
scot_clean %>% ggplot(aes(x=year, y=births, color=factor(age))) +
  geom_line() +
  gghighlight::gghighlight(age %in% c(20,30)) +
  labs(title="Young mothers becoming less common in Scotland",
       color="Age of mother")
```

```
## Warning: Tried to calculate with group_by(), but the calculation failed.
## Falling back to ungrouped filter operation...
```

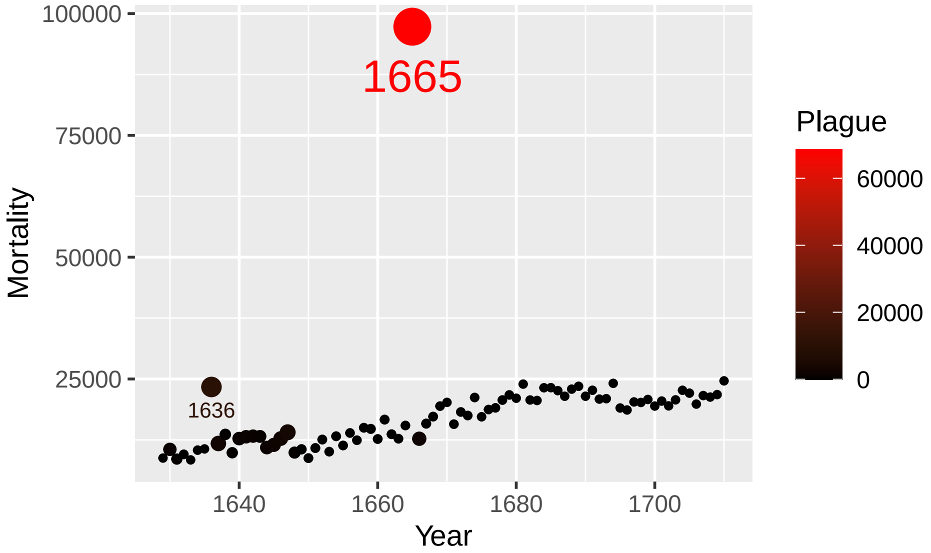
```
## Warning: Using `across()` in `filter()` is deprecated, use `if_any()` or
## `if_all()`.
```



Problem 29

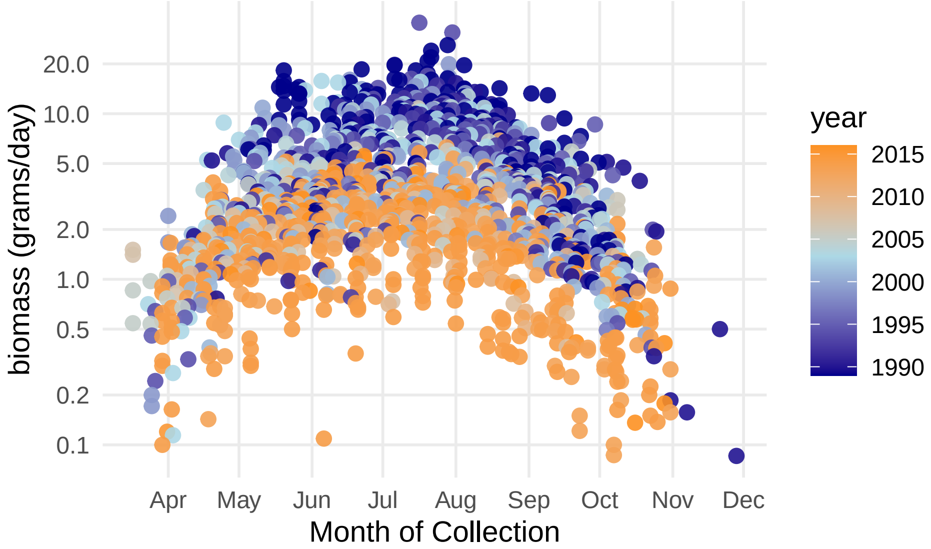
```
Arbuthnot <- HistData::Arbuthnot
Arbuthnot %>% ggplot(aes(x=Year, y=Mortality, size=Plague, color=Plague)) +
  geom_point() +
  scale_color_gradient(low="black",high="red") +
  guides(size=FALSE) +
  geom_text(data=filter(Arbuthnot, Plague > 10000), aes(label=Year), vjust=2)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



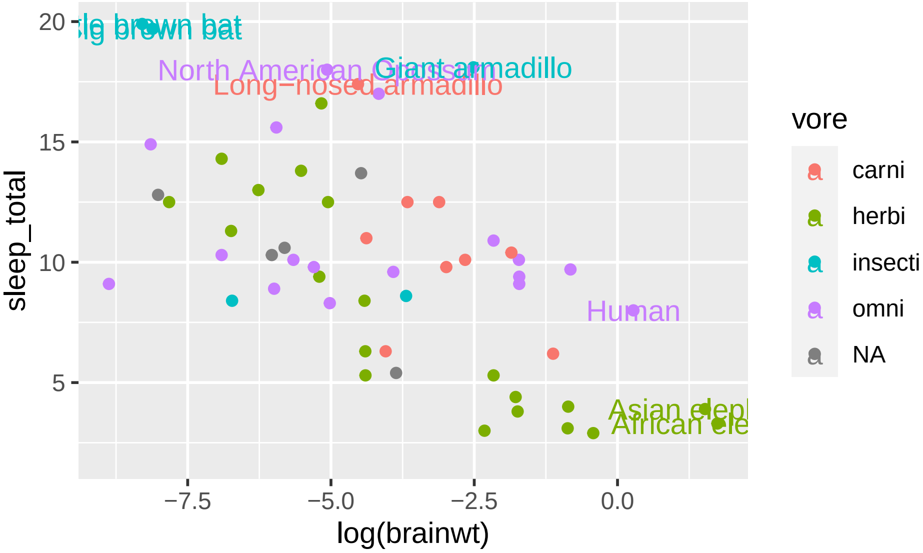
Problem 30

```
library(lubridate)
bb <- fosdata::biomass
bb %>%
  mutate(to = dmy(str_c(to, "-",year)),
         from = dmy(str_c(from, "-", year)),
         day_of_year = (yday(to) + yday(from) )/2,
         grams_per_day = biomass/(yday(to) - yday(from) + 1 )) %>%
  slice_max(grams_per_day, n = 1499) %>%
  #filter(log10(grams_per_day) > 0.001) %>%
  ggplot(aes(x = day_of_year, y = grams_per_day, color = year)) +
  geom_point(alpha = 0.9, size = 2.2) +
  scale_color_gradient2(low = "darkblue", mid = "lightblue", high = "darkorange", midpoint = 2003
  scale_y_log10(breaks = c(.1, .2, .5, 1, 2, 5, 10, 20), minor_breaks = NULL) +
  theme_minimal() +
  scale_x_continuous(breaks = c(90, 120, 151, 181, 212, 243, 273, 304, 334),
                     labels = c("Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),
                     minor_breaks = NULL) +
  labs(y = "biomass (grams/day)",
       x = "Month of Collection")
```



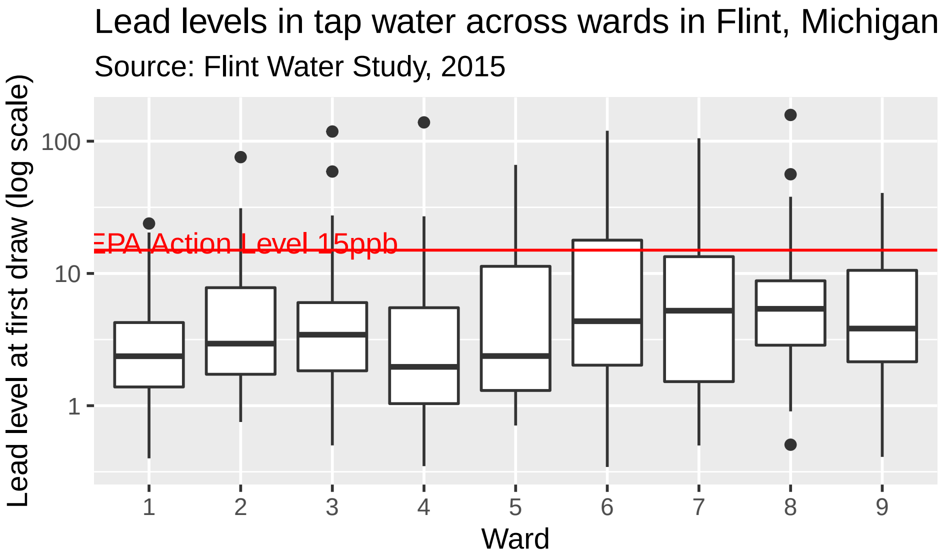
Problem 31

```
msleep %>% ggplot(aes(x=log(brainwt), y=sleep_total, color=vore, label=name)) +
  geom_point() +
  geom_text(data=subset(msleep, brainwt > 1 | sleep_total > 17))
```



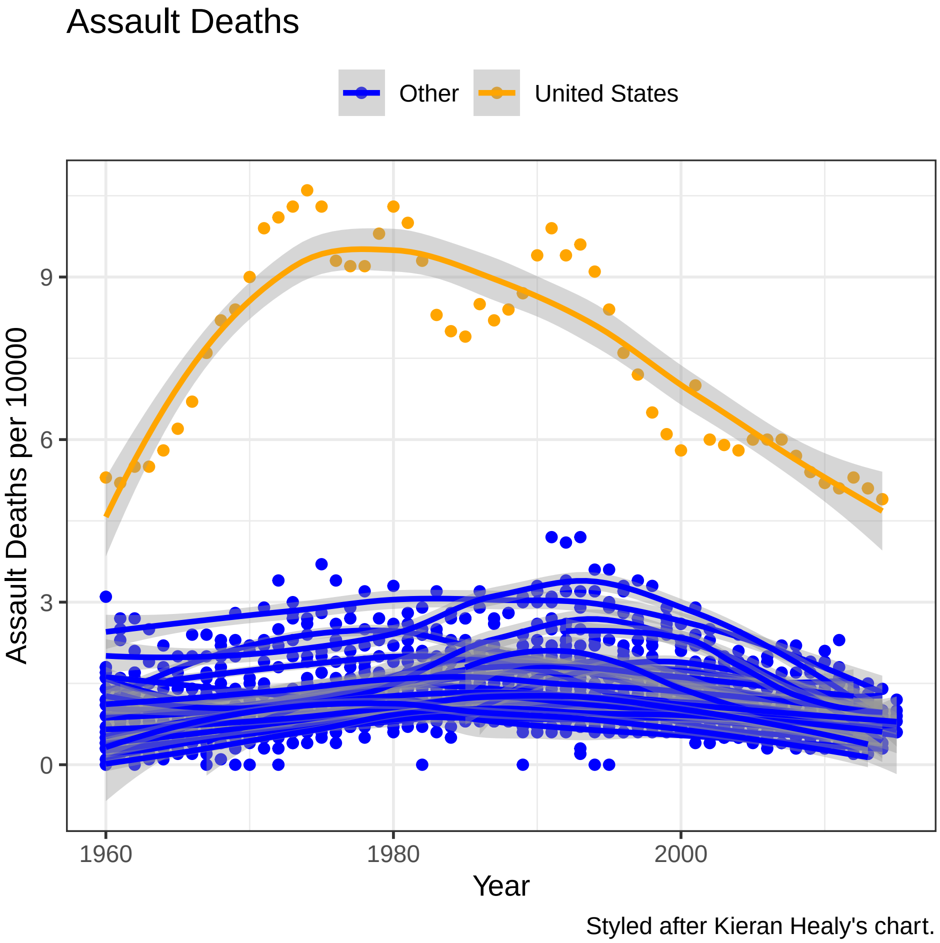
Problem 32

```
fosdata::flint %>% filter(Ward != 0) %>%
  ggplot(aes(x=factor(Ward), y=Pb1)) + geom_boxplot() +
  geom_hline(yintercept = 15,color="red") +
  labs(title="Lead levels in tap water across wards in Flint, Michigan",
       x="Ward",y="Lead level at first draw (log scale)",
       subtitle = "Source: Flint Water Study, 2015") +
  scale_y_log10() +
  annotate("text",x=2.02,y=17,label="EPA Action Level 15ppb",color="red")
```



Problem 33

```
# Load data
assaults.raw <- read.csv("https://raw.githubusercontent.com/kjhealy/assault-deaths/master/data/oe

# Data cleaning: gather multiple Year columns into one variable, fix Year names.
assaults <- assaults.raw %>%
  pivot_longer(cols=-Country, names_to='Year', values_to='Assaults') %>%
  filter(!is.na(Assaults)) %>% mutate(Year = as.numeric(substr(Year,2,6)))

# Create new variable to indicate US or other
assaults <- assaults %>% mutate(US = ifelse (Country == "United States","United States","Other"))

# Countries to remove from chart
badlands <- c("Mexico","Estonia","Chile","Israel")

# Base plot
assaults.plot <- assaults %>% filter(!(Country %in% badlands)) %>%
  ggplot(aes(x=Year, y=Assaults, group=Country, color=US)) + geom_point() + geom_smooth()

# Clean up details
assaults.plot <- assaults.plot +
  scale_color_manual(values=c("Blue","Orange")) +
  labs(title="Assault Deaths", caption="Styled after Kieran Healy's chart.",
       y="Assault Deaths per 10000") +
  theme_bw() +
  theme(legend.position = "top", legend.title = element_blank())

# Display (and suppress the message that geom_smooth() always gives)
suppressMessages(print(assaults.plot))
```

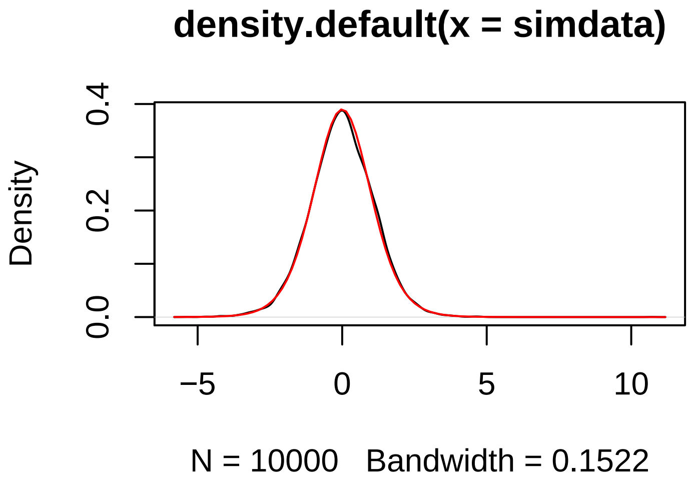

Assault Deaths
Other
United States
Assault Deaths per 10000
9
6
3
0
1960
1980
2000
Year
Styled after Kieran Healy's chart.

8

Inference on the Mean – Solutions

Problem 1

```
simdata <- replicate(10000,{
  x <- rnorm(12,1,3)
  s <- sd(x)
  xbar <- mean(x)
  (xbar - 1) / (s / sqrt(12))
})
plot(density(simdata))
curve(dt(x,11),add=TRUE,col="red")
```



Problem 2 $n = 12$ is the smallest. Here is a table of nearby values:

```
n <- 10:14
data.frame(n,diff = abs(pt(1,n)-pt(0,n) - (pnorm(1)-pnorm(0))))
```

```
##    n        diff
## 1 10 0.011791312
## 2 11 0.010745094
## 3 12 0.009869275
## 4 13 0.009125385
## 5 14 0.008485718
```

Problem 3

```
-qt(0.1,6)
```

```
## [1] 1.439756
```

Problem 4

```
1-pnorm(2)
```

```
## [1] 0.02275013
```

```
1-pt(2,40)
```

```
## [1] 0.02616117
```

```
1-pt(2,20)
```

```
## [1] 0.02963277
```

```
1-pt(2,10)
```

```
## [1] 0.03669402
```

```
1-pt(2,5)
```

```
## [1] 0.05096974
```

Lower DF has more area in the tail of the t distribution.

Problem 5

```
-qt(0.005, 19)
```

```
## [1] 2.860935
```

Problem 6

```
t.test(fosdata::plastics$diameter, conf.level = 0.99)$conf.int
```

```
## [1] 16.49394 20.02190
## attr(,"conf.level")
## [1] 0.99
```

We are 99% confident the true mean precipitation in inches per year is in the interval [30.54601,39.22542].

Problem 7

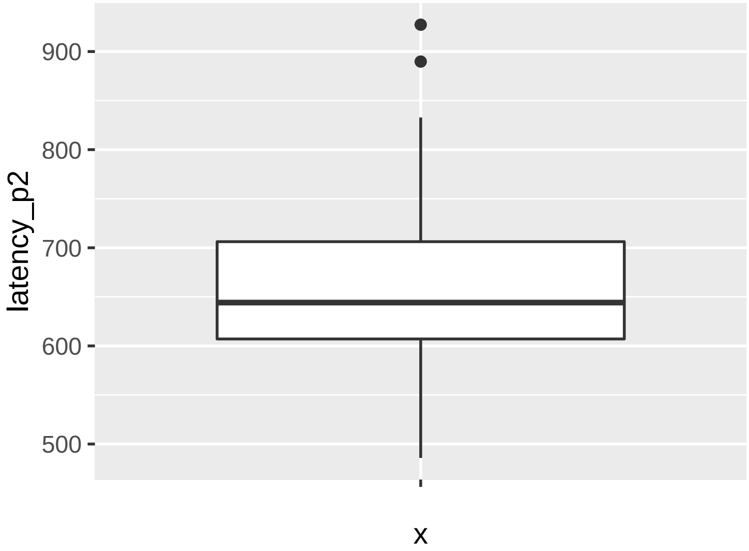
```
t.test(morley$Speed)$conf.int
```

```
## [1] 836.7226 868.0774
## attr(,"conf.level")
## [1] 0.95
```

The confidence interval is [299836.7, 299868.1]. This does not contain the modern accepted speed of light of 299729.

Problem 8

```
brake <- fosdata::brake
young <- filter(brake, age_group == "Young")
young %>%
  ggplot(aes(x = "", y = latency_p2)) +
  geom_boxplot()
```



```
  #stat_qq(distribution = qnorm) could also use this. looks OK

t.test(young$latency_p2)
```

```
##
##  One Sample t-test
##
## data:  young$latency_p2
## t = 43.841, df = 39, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  633.8283 695.1436
## sample estimates:
## mean of x
##  664.4859
```

95 percent CI for the mean time it takes younger drivers to release the accelerator is 634 to 695 ms.

Problem 9

a.

```
cows <- fosdata::cows_small
t.test(cows$tk_12)$conf.int
```

```
## [1] -1.954056 -1.642435
## attr(,"conf.level")
## [1] 0.95
```

b.

```
t.test(cows$control)$conf.int
```

```
## [1] 0.03132827 0.11954892
## attr(,"conf.level")
## [1] 0.95
```

c. The `tk_12` interval is wider

d.

```
sd(cows$tk_12)
```

```
## [1] 0.3232691
```

```
sd(cows$control)
```

```
## [1] 0.09151813
```

The sd is larger for the `tk_12` condition.

e. A larger sd gives a wider confidence interval.

Problem 10

```
sigma <- 2
zcritical <- -qnorm(0.025)
xbar <- 3
lower <- xbar - zcritical * sigma / sqrt(5)
upper <- xbar + zcritical * sigma / sqrt(5)
c(lower,upper) # 95% confidence interval
```

```
## [1] 1.246955 4.753045
```

Problem 11 a is II (95%), b is III (99%), c is I (90%). The wider intervals give higher confidence.

Problem 12

a. The 98% confidence interval is $[1.62, 2.98]$. $t_{\alpha/2} = 2.5394832$, the standard error is $S/\sqrt{n} =$ `1.2/sqrt(20)`.
b. 72.2%. Half the width is 0.3, so $t_{\alpha/2} = 0.3\sqrt{n}/S = 1.118$. Then `1 - pt(-1.118, 19)*2` = 0.7224957 is the confidence level.

Problem 13

a. $\mu = 2$.
b.

```
mean(replicate(10000,{
  x <- rexp(10,0.5); ci <- t.test(x)$conf.int; (2 > ci[1] & 2 < ci[2])
}))
```

```
## [1] 0.8997
```

c. The answer is not 95% because the population is skew, and a sample of size 10 is not enough for the $\overline{X}$ distribution to be normal.

Problem 14

```
t.test(ISwR::react, conf.level = .98)$conf.int
```

```
## [1] -1.0365875 -0.5562269
## attr(,"conf.level")
## [1] 0.98
```

We are 98% confident the true mean in determinations of reaction sizes in mm lies lies in the interval [-1.04, -0.56]. The mean is significantly different from zero ($P = 1.115 \times 10^{-13}$).

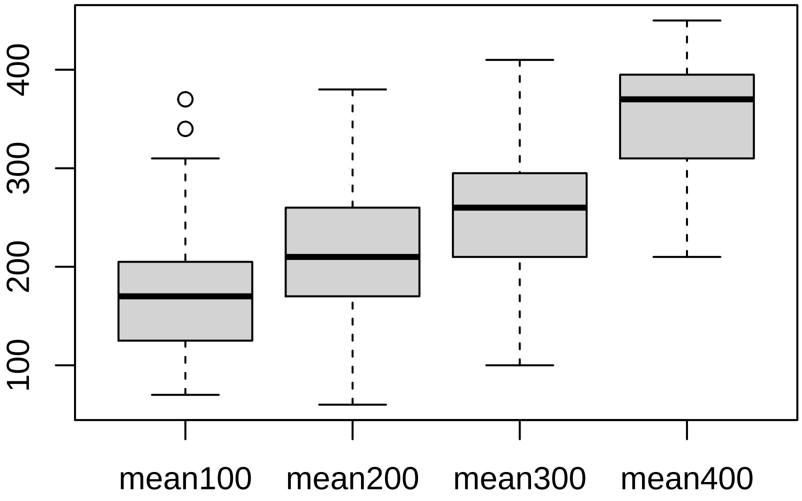
Problem 15

a. The alpha level for $|T| > 1.6$ is `pt(-1.6, 19)*2` = 0.1260951.

b. The rejection region for $\alpha = 0.005$ is $|T| > 3.1737245$, the value coming from `qt(.005/2, 19)`.

Problem 16

```
children <- fosdata::weight_estimate %>% filter(age != "adult")
children %>% select(starts_with("mean")) %>% boxplot()
```



```
t.test(children$mean100, mu=100)$p.value
```

```
## [1] 1.129514e-12
```

```
t.test(children$mean200, mu=200)$p.value
```

```
## [1] 0.1214797
```

```
t.test(children$mean300, mu=300)$p.value
```

```
## [1] 3.497206e-06
```

```
t.test(children$mean400, mu=400)$p.value
```

```
## [1] 1.06494e-08
```

The children's mean estimates were significantly different from the correct weight at 100, 300, and 400g. Their mean estimate for the 200g weight was 214.5, not significantly different from the correct value.

Problem 17

```
bp.obese <- ISwR::bp.obese
t.test(bp.obese$bp, mu=120)
```

```
##
##  One Sample t-test
##
## data:  bp.obese$bp
## t = 3.8986, df = 101, p-value = 0.0001743
## alternative hypothesis: true mean is not equal to 120
## 95 percent confidence interval:
##  123.4479 130.5914
## sample estimates:
```

```
## mean of x
##  127.0196
```

The population is Mexican-American adults in a small California town. There is significant evidence that the mean blood pressure of this population is different from 120 ($P = 0.0002$).

Problem 18

a. The alpha level for $|Z| > 2.2$ is `pnorm(-2.2)*2` $= 0.0278069$.
b. The rejection region for $\alpha = 0.01$ is $|Z| > 2.5758293$, the value coming from `qnorm(.01/2)`.

Problem 19 According to the help (`?bp.obese`), the `obese` variable is a ratio of actual weight to ideal weight. The natural null hypothesis is that this ratio is 1. If μ is the population mean obesity, we test $H_0 : \mu = 1$.

```
t.test(bp.obese$obese, mu=1)
```

```
##
##  One Sample t-test
##
## data:  bp.obese$obese
## t = 12.262, df = 101, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 1
## 95 percent confidence interval:
##  1.262395 1.363684
## sample estimates:
## mean of x
##  1.313039
```

There is significant evidence ($P < 2.2 \times 10^{-16}$) that the population obesity ratio is not 1. This population is obese.

Problem 20

```
pvals <- replicate(10000, {
  sim_data <- rnorm(20)
  alt <- ifelse(mean(sim_data) < 0, "less", "greater")
  t.test(sim_data, mu=0, alternative = alt)$p.value
})
mean(pvals < 0.05)
```

```
## [1] 0.0978
```

Problem 21

a.

```
x <- rnorm(10,4,1)
ci <- t.test(x)$conf.int
print(ci)
```

```
## [1] 3.033752 4.458540
## attr(,"conf.level")
## [1] 0.95
```

```
(4 > ci[1] & 4 < ci[2])
```

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

b.

```
mean(replicate(10000,{
  x <- rnorm(10,4,1); ci <- t.test(x)$conf.int; (4 > ci[1] & 4 < ci[2])
}))
```

```
## [1] 0.9502
```

Problem 22 Since multiple temperature readings are taken from the same person, those values of X_i are dependent.

Problem 23 a.

```
x <- rexp(10,1/4);
t.test(x)$conf.int
```

```
## [1] 2.505336 7.420249
## attr(,"conf.level")
## [1] 0.95
```

Check if 4 is in the interval.

b.

```
mean(replicate(10000,{
  x <- rexp(10,1/4); ci <- t.test(x)$conf.int; (4 > ci[1] & 4 < ci[2])
}))
```

```
## [1] 0.9019
```

The t-test assumes that the population is normal or n is large. $n = 10$ is too small for the highly skew exponential distribution.

c.

```
mean(replicate(10000,{
  x <- rnorm(100,4,1); ci <- t.test(x)$conf.int; (4 > ci[1] & 4 < ci[2])
}))
```

```
## [1] 0.9489
```

```
mean(replicate(10000,{
  x <- rexp(100,1/4); ci <- t.test(x)$conf.int; (4 > ci[1] & 4 < ci[2])
}))
```

```
## [1] 0.9412
```

Now $n = 100$ is large enough for t-test to work reasonably well even with the exponential distribution.

Problem 24 The type I error rate is about 17%.

```
pvals <- replicate(10000, {
  x <- rnorm(20);
  y <- x[1:19] + x[2:20];
  t.test(y)$p.value
  })
mean(pvals < 0.05)
```

```
## [1] 0.1706
```

Problem 25

a. The quality of the painting has no effect on the pain people suffer.

b. Things don't look any different when you bend over and view them between your legs.

c. All mammals empty their bladders in about 21 seconds.

Problem 26

a. All four measurements were better for the didgeridoo group.
b. Quality of sleep did not, because $P = 0.27$ is not significant.
c. Partner reported sleep disturbance is the most significant.

Problem 27

```
t.test(grey_score_avg ~ subspecies, data=fosdata::chimps)
```

```
##
##  Welch Two Sample t-test
##
## data:  grey_score_avg by subspecies
## t = -1.8344, df = 137.73, p-value = 0.06875
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.38456228  0.01441608
## sample estimates:
## mean in group schweinfurthii          mean in group verus
##                      2.427719                     2.612792
```

There is no significant difference in grey score between the subspecies ($P = 0.06875$).

Problem 28

```
with(fosdata::leg_strength, t.test(mean_wii, mean_sid, paired=TRUE))
```

```
##
##  Paired t-test
##
## data:  mean_wii and mean_sid
## t = -10.431, df = 29, p-value = 2.514e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -85.69872 -57.60086
## sample estimates:
## mean of the differences
##               -71.64979
```

There is a significant difference in the measurements ($p = 2.5 \times 10^{-11}$)

Problem 29

```
t.test(nic ~ filter, data=fosdata::cigs)$p.value
```

```
## [1] 6.2683e-28
```

```
t.test(tar ~ filter, data=fosdata::cigs)$p.value
```

```
## [1] 1.70833e-50
```

```
t.test(nic ~ menthol, data=fosdata::cigs)$p.value
```

```
## [1] 0.9558782
```

```
t.test(tar ~ menthol, data=fosdata::cigs)$p.value
```

```
## [1] 0.6403627
```

a. Significant difference. b. Significant difference. c. No significant difference. d. No significant difference.

Overall, filters make a difference, but menthol has no effect.

Problem 30

```
t.test(bp ~ sex, data=bp.obese)
```

```
##
##  Welch Two Sample t-test
##
## data:  bp by sex
## t = 0.46033, df = 98.535, p-value = 0.6463
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.443341  8.731743
## sample estimates:
## mean in group 0 mean in group 1
##        127.9545        126.3103
```

Since the P-value is 0.6463, we fail to reject the null hypothesis. There is not significant evidence that the mean blood pressure of men and women is different. This data is specific to one town in California and should not be generalized to the entire US population.

Problem 31 Let μ_F and μ_R be the true mean decrease in blood pressure for fish oil and for regular oil. We test the null hypothesis $H_0 : \mu_F = \mu_R$.

```
t.test(BP ~ Diet, data=Sleuth3::ex0112)
```

```
##
##  Welch Two Sample t-test
##
## data:  BP by Diet
## t = 3.0621, df = 9.2643, p-value = 0.01308
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   2.039893 13.388678
## sample estimates:
##    mean in group FishOil mean in group RegularOil
##                 6.571429                -1.142857
```

There is a significant difference in blood pressure decrease between fish oil and regular oil (P = 0.01308).

Problem @barnacletest

```
fosdata::barnacles %>% filter(location == "FGB") %>% t.test(log(barnacle_density) ~ deep, data=.
```

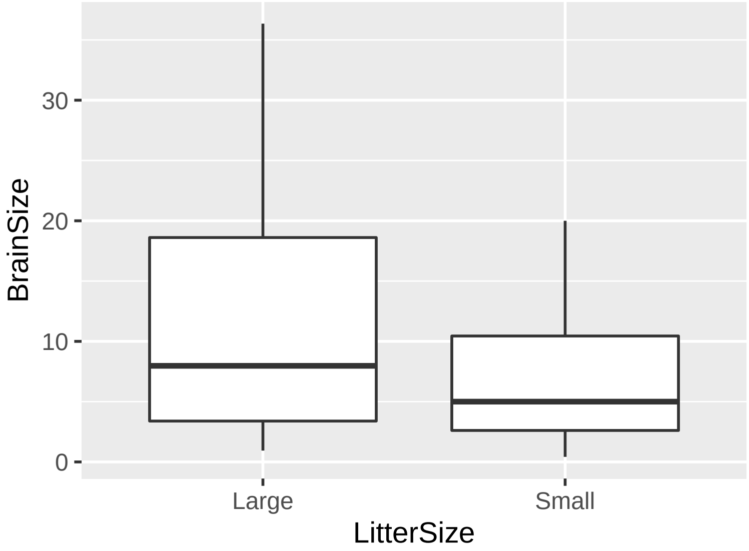
```
##
```

```
##  Welch Two Sample t-test
##
## data:  log(barnacle_density) by deep
## t = -3.1951, df = 29.264, p-value = 0.003339
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.1754965 -0.2581494
## sample estimates:
##    mean in group deep mean in group shallow
##              4.964460              5.681283
```

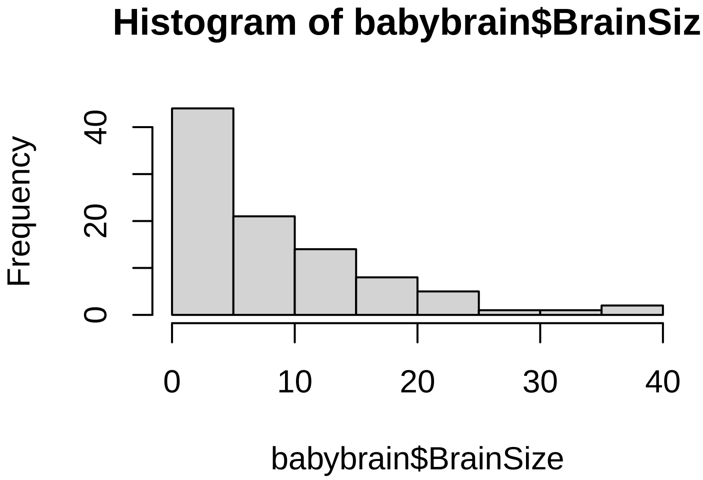
There is a significant difference in barnacle density between deep and shallow reefs, $P = 0.0033$.

Problem 32

```
babybrain <- Sleuth3::ex0333
ggplot(babybrain,aes(x=LitterSize,y=BrainSize))+geom_boxplot()
```

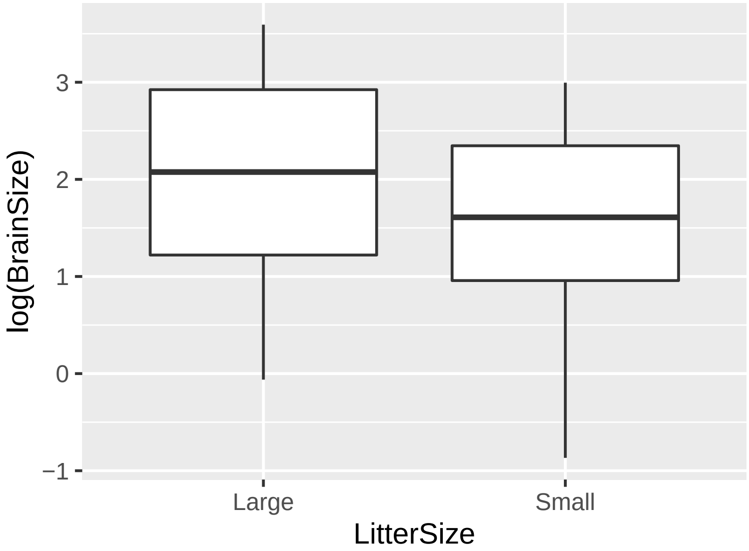


```
hist(babybrain$BrainSize)
```

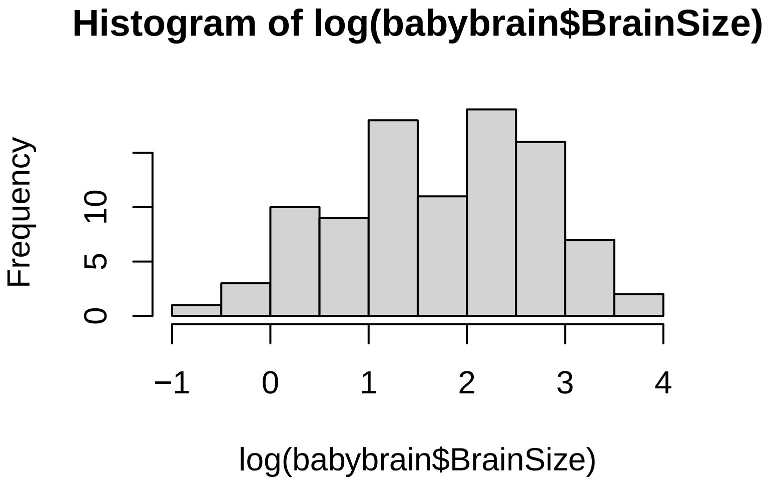


The histogram suggests that BrainSize is not normal. Now take logs:

```
ggplot(babybrain,aes(x=LitterSize,y=log(BrainSize)))+geom_boxplot()
```



```
hist(log(babybrain$BrainSize))
```



The histogram shows the log data is reasonably normal. Using the log of BrainSize:

```
t.test(log(BrainSize) ~ LitterSize, data=babybrain)
```

```
##
##  Welch Two Sample t-test
##
## data:  log(BrainSize) by LitterSize
## t = 1.9669, df = 90.684, p-value = 0.05225
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.003942989  0.797879510
## sample estimates:
## mean in group Large mean in group Small
##            1.949426            1.552458
```

There is no significant difference in brain size between large and small litter mammals (P = 0.052). Without taking the log, the t-test results in a P-value of 0.016, giving a significant difference, but that value is highly suspect due to non-normal data.

Problem 33 Let μ_A and μ_U be the mean brain volume in affected and unaffected twins. We test $H_0 : \mu_A = \mu_U$ against $H_a : \mu_A \neq \mu_U$.

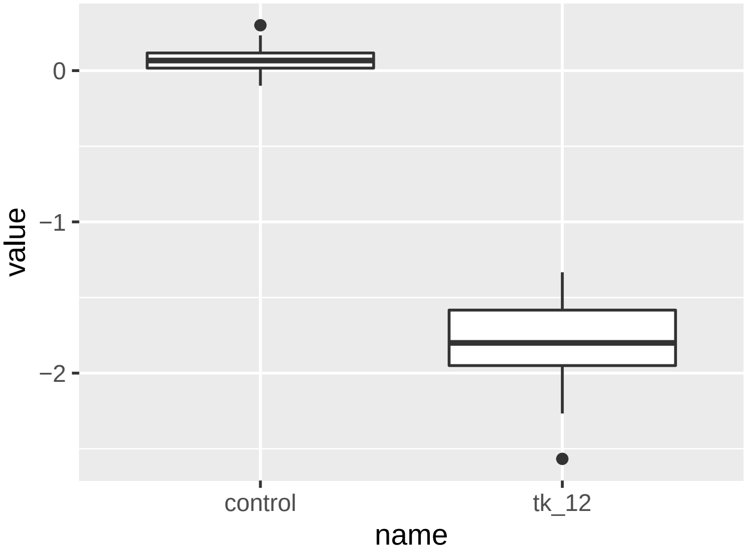
```
twins <- Sleuth3::case0202
t.test(twins$Unaffected,twins$Affected,paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  twins$Unaffected and twins$Affected
## t = 3.2289, df = 14, p-value = 0.006062
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.0667041 0.3306292
## sample estimates:
## mean of the differences
##               0.1986667
```

We find there is a significant difference in brain volume between affected and unaffected twins (P = 0.006).

Problem 34

```
cows <- fosdata::cows_small
cows %>%
  tidyr::pivot_longer(cols = c(control, tk_12)) %>%
  ggplot(aes(name, value)) +
  geom_boxplot() #looks relatively normal, symmetric
```



```
t.test(cows$control, cows$tk_12, paired = TRUE)
```

```
##
##  Paired t-test
##
```

```
## data:  cows$control and cows$tk_12
## t = 23.558, df = 18, p-value = 5.609e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.706584 2.040784
## sample estimates:
## mean of the differences
##                1.873684
```

a. Yes, with a p-value of 5.609 times 10^{-14}, we would reject that the means are the same and conclude that the TK-0.75 cools cows better than nothing.

Problem 35

```
child_tasks <- fosdata::child_tasks
t.test(day_night_completion_time_secs ~ gender, data=child_tasks)
```

```
##
##  Welch Two Sample t-test
##
## data:  day_night_completion_time_secs by gender
## t = -1.2891, df = 49.808, p-value = 0.2033
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.270679  0.931872
## sample estimates:
## mean in group Female   mean in group Male
##             26.44026             28.10967
```

There is no significant difference in mean time to complete the day-night task between Male and Female children ($p = 0.2033$).

Problem 36

```
t.test(age_in_months ~ gender, data=child_tasks)
```

```
##
##  Welch Two Sample t-test
##
## data:  age_in_months by gender
## t = 1.389, df = 56.386, p-value = 0.1703
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.099834 11.601589
## sample estimates:
## mean in group Female   mean in group Male
##             96.68421             91.93333
```

There is no significant difference in mean age between Male and Female children ($p = 0.1703$).

Problem 37

```
sixseven <- fosdata::child_tasks %>% filter(age_group %in% c("6 year olds", "7 year olds"))
t.test(day_night_completion_time_secs ~ age_group, data=sixseven)
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  day_night_completion_time_secs by age_group
## t = 3.5721, df = 29.224, p-value = 0.001251
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.331582 8.573124
## sample estimates:
## mean in group 6 year olds mean in group 7 year olds
##                  32.33824                  26.88588
```

There is a significant difference in the mean time to finish the day night task between the 6 and 7 year old age gropus ($p = 0.001251$). This is significant at the $\alpha = 0.01$ level.

Problem 38

```
t.test(child_tasks$card_sort_preswitch_time_secs, child_tasks$card_sort_postswitch_time_secs, pa
```

```
##
##  Paired t-test
##
## data:  child_tasks$card_sort_preswitch_time_secs and child_tasks$card_sort_postswitch_time_sec
## t = 3.5477, df = 67, p-value = 0.0007152
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2379847 0.8502506
## sample estimates:
## mean of the differences
##               0.5441176
```

The children are significantly faster post-switch, $P = 0.0007$.

Problem 39

```
brake_clipped <- fosdata::brake %>% filter(latency_p2 < 1700)
t.test(latency_p2 ~ age_group, data = brake_clipped)
```

```
##
##  Welch Two Sample t-test
##
## data:  latency_p2 by age_group
## t = 8.0005, df = 56.709, p-value = 7.096e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  199.6023 332.8974
## sample estimates:
##   mean in group Old mean in group Young
##            930.7358            664.4859
```

Even after removing the one large older outlier, the difference between young and old latency is significant ($p = 7 \times 10^{-11}$)

Problem 40

a. Type II error.
b. Type I error.
c. Generally the type I eror would be considered more serious, since it results in the punishment of an innocent person. The type II error lets a criminal get away.

Problem 41

a. The null is that the drug has no effect.
b. A type I error means that an ineffective drug is believed to work.
c. A type II error means that an effective drug is believed to be useless.
d. For ill patients, a type I error means they will receive worthless treatment. A type II error means they are deprived of an effective treatment.
e. For drug manufacturers, a type I error means they produce a worthless drug. A type II error means they miss a drug with potential use.

Problem @powertemp You need 21 in each group, 42 overall.

```
power.t.test(n=NULL, delta=1, sd=0.73, power=.99)
```

```
##
##      Two-sample t test power calculation
##
##               n = 20.59948
##           delta = 1
##              sd = 0.73
##       sig.level = 0.05
##           power = 0.99
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Problem @powerconcrete It has only 21% power.

```
power.t.test(n=5, delta=5, sd=6)
```

```
##
##      Two-sample t test power calculation
##
##               n = 5
##           delta = 5
##              sd = 6
##       sig.level = 0.05
##           power = 0.2129346
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Problem @powertailed

```
# a. 94.5% power
power.t.test(n=40, delta=1, sd=sqrt(3), type="one.sample")
```

```
##
##      One-sample t test power calculation
##
##               n = 40
##           delta = 1
##              sd = 1.732051
##       sig.level = 0.05
##           power = 0.9452242
##     alternative = two.sided
```

```
# b. A little less. Simulation varies but around 93.5-94.
pvals <- replicate(10000,t.test(rt(40,3),mu=1)$p.value)
mean(pvals < 0.05)
```

```
## [1] 0.9372
```

Problem 44 Starting point:

```
power.t.test(n=200, delta=0.2, sd=1, sig.level=0.05, type="one.sample")
```

```
##
##      One-sample t test power calculation
##
##               n = 200
##           delta = 0.2
##              sd = 1
##       sig.level = 0.05
##           power = 0.8036658
##     alternative = two.sided
```

```
power.t.test(n=400, delta=0.2, sd=1, sig.level=0.05, type="one.sample")$power
```

```
## [1] 0.9788416
```

```
power.t.test(n=200, delta=0.4, sd=1, sig.level=0.05, type="one.sample")$power
```

```
## [1] 0.9998784
```

```
power.t.test(n=200, delta=0.2, sd=1, sig.level=0.10, type="one.sample")$power
```

```
## [1] 0.87979
```

In all cases the power increases. You get the most boost from doubling the effect size, almost as much from doubling the sample size, and only about a 10% boost in power from dropping your level of significance.

Problem 42

The authors assumed a difference of abput 10.7 units between the two groups.

```
power.t.test(n = 40 * .7, sd = 14, power = 0.8, sig.level = .05)
```

```
##
##      Two-sample t test power calculation
##
##               n = 28
##           delta = 10.67366
##              sd = 14
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Looking at the paper, they actually assumed a difference of 11 units, which would lead to 38 participants in each group with a dropout rate of exactly 30 percent.

```
power.t.test(delta = 11, sd = 14, power = 0.8, sig.level = .05)$n/.7
```

```
## [1] 37.74772
```

Problem 43

a. A 95 percent confidence interval for the difference in means is $[-13.1, 3.4]$ units.

```
t.test(prwe12m ~ cast_position, data = fosdata::wrist, var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  prwe12m by cast_position
## t = -1.1739, df = 84, p-value = 0.2438
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.091655   3.372615
## sample estimates:
## mean in group 1 mean in group 2
##        15.51282        20.37234
```

b. This matches what was found in the paper.

c. Let's see whether the variances appear equal. They aren't too different in the sample.

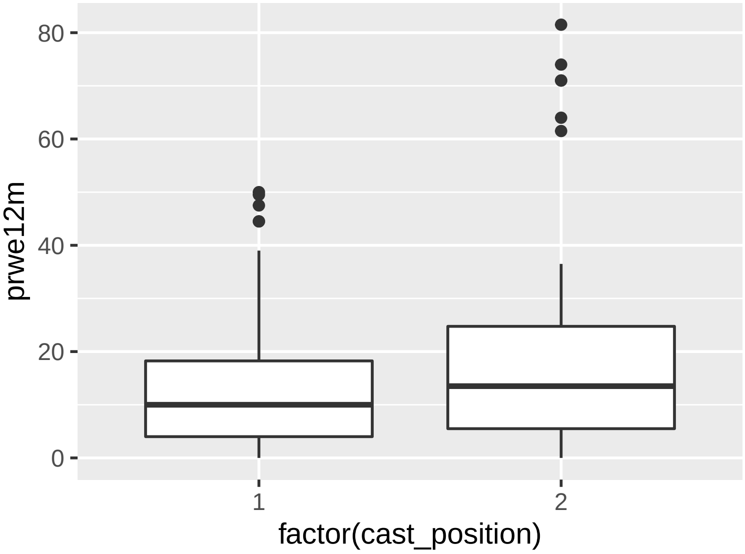
```
fosdata::wrist %>%
  group_by(cast_position) %>%
  summarize(sd = sd(prwe12m, na.rm = T))
```

```
## # A tibble: 2 x 2
##   cast_position    sd
##           <dbl> <dbl>
## 1             1  15.4
## 2             2  21.7
```

And they also don't appear too different using boxplots, though the scores appear right skewed.

```
fosdata::wrist %>%
  ggplot(aes(x = factor(cast_position), y = prwe12m)) +
  geom_boxplot()
```

```
## Warning: Removed 19 rows containing non-finite values (stat_boxplot).
```



Overall, we cannot claim that the authors were incorrect when assuming equal variances. Following the philosophy of this book, we would not have made that assumption, however, unless there is a larger study that validates it.

Problem 45

a.

```
data <- c(-1,2,-3,-4,5)
t.test(data,mu=0)$p.value
```

```
## [1] 0.9096562
```

b. The p-value will not change, because it is based on $t = \frac{\bar{x}-\mu}{s/\sqrt{n}}$. Multiplying the data by 2 will multiply $\bar{x}$ by 2, s by 2, and $\mu = 0$ here. The value of t will not change.

```
t.test(data*2,mu=0)$p.value
```

```
## [1] 0.9096562
```

c.

```
t.test(data*abs(data),mu=0)$p.value
```

```
## [1] 0.9357209
```

Applying the nonlinear transformation to the data changes the p-value.

Problem 46

a.

```
x <- rnorm(20); x[20] <- 1000
t.test(x,mu=5)
```

```
##
##  One Sample t-test
##
## data:  x
## t = 0.90069, df = 19, p-value = 0.379
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  -54.6152 154.6820
## sample estimates:
## mean of x
##  50.03342
```

The data does not show a significant difference from a mean of 5, even though the sample mean was close to 50.

b.

$$\begin{aligned}\lim_{x_n\to\infty}\frac{\overline{X}}{x_n} &= \lim_{x_n\to\infty}\frac{(x_1+\cdots+x_{n-1}+x_n)/n}{x_n}\\ &= \lim_{x_n\to\infty}\left(\frac{1}{n}\right)\left(\frac{x_1}{x_n}+\cdots+\frac{x_{n-1}}{x_n}+\frac{x_n}{x_n}\right)=\frac{1}{n}(0+\cdots+0+1)=\frac{1}{n}.\end{aligned}$$

c.

$$\begin{aligned}\lim_{x_n\to\infty}\frac{S^2}{x_n^2} = \lim_{x_n\to\infty}\frac{\frac{1}{n-1}\sum_{i=1}^n(x_i-\overline{X})^2}{x_n^2} &= \lim_{x_n\to\infty}\frac{1}{n-1}\sum_{i=1}^n\left(\frac{x_i}{x_n}-\frac{\overline{X}}{x_n}\right)^2\\ &= \frac{1}{n-1}\left[(0-\frac{1}{n})^2+\cdots+(0-\frac{1}{n})^2+(1-\frac{1}{n})^2\right]\\ &= \frac{1}{n-1}\left[\frac{1}{n^2}+\cdots+\frac{1}{n^2}+(1-\frac{2}{n}+\frac{1}{n^2})\right]\\ &= \frac{1}{n-1}\left[\frac{n-1}{n^2}+(1-\frac{2}{n}+\frac{1}{n^2})\right]=\frac{1}{n}.\end{aligned}$$

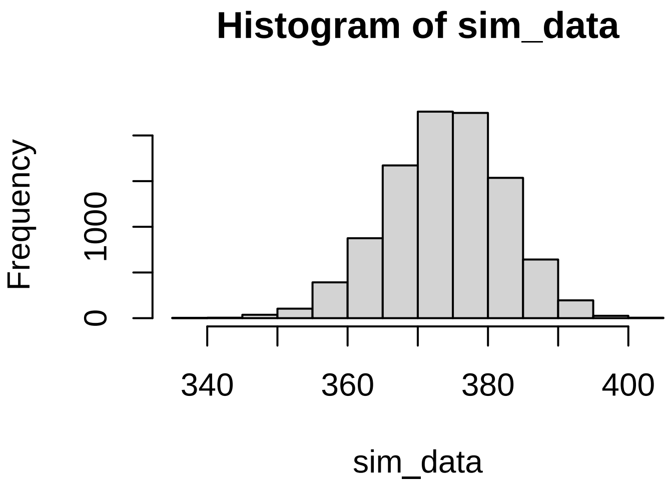
d. By taking square roots in part (c), we get that $\lim_{x_n\to\infty}\frac{S}{x_n}=\frac{1}{\sqrt{n}}$.

$$\begin{aligned}\lim_{x_n\to\infty}\frac{\overline{X}-\mu_0}{S/\sqrt{n}} &= \lim_{x_n\to\infty}\frac{\frac{\overline{X}-\mu_0}{x_n}}{\frac{S/\sqrt{n}}{x_n}}\\ &= \frac{\frac{1}{n}-0}{\frac{1}{\sqrt{n}}/\sqrt{n}}=\frac{1/n}{1/n}=1.\end{aligned}$$

e. As $x_n\to\infty$, t will always be near 1, which is never significant. Therefore, if the data contains a large enough outlier, the t test can never reject the null hypothesis.

Problem 47

```
set.seed(3850)
ww <- fosdata::weight_estimate
a400 <- filter(ww, age == "adult") %>%
  pull(mean400)
sim_data <- replicate(10000, {
  boot <- sample(a400, replace = T)
  mean(boot)
})
hist(sim_data)
```



```
quantile(sim_data, c(.025, .975))
```

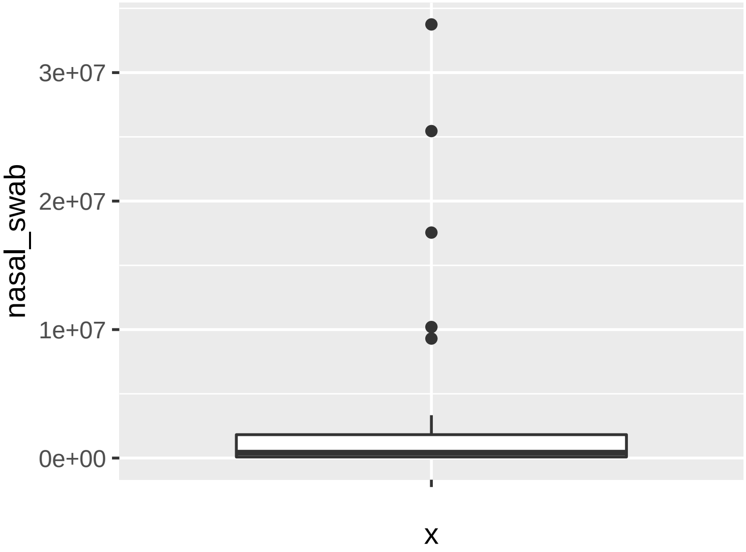
```
##     2.5%    97.5%
## 357.1875 390.0000
```

The 95 percent confidence interval for the mean weight is $[357, 390]$.

Problem 48

a. No, it is not appropriate to use `t.test` even with 37 observations due to the extreme skewness of the population.

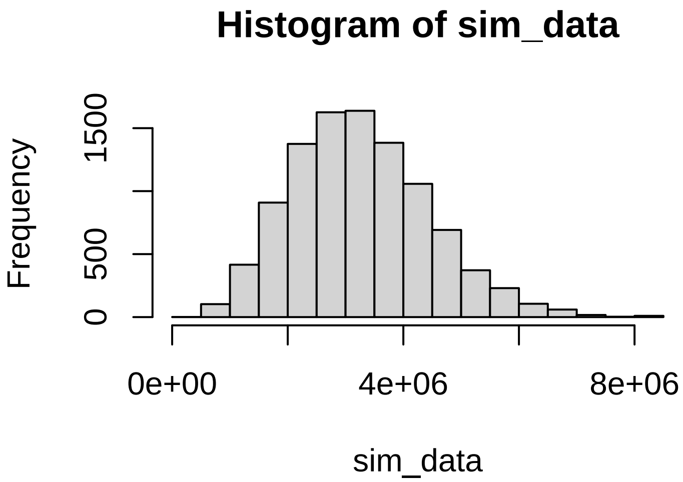
```
masks <- fosdata::masks
ggplot(masks, aes(x = "", y = nasal_swab)) + geom_boxplot()
```



```
e1071::skewness(masks$nasal_swab)
```

```
## [1] 2.882549
```

```
set.seed(3850)
nasal <- masks$nasal_swab
sim_data <- replicate(10000, {
  boot <- sample(nasal, replace = T)
  mean(boot)
})
hist(sim_data)
```



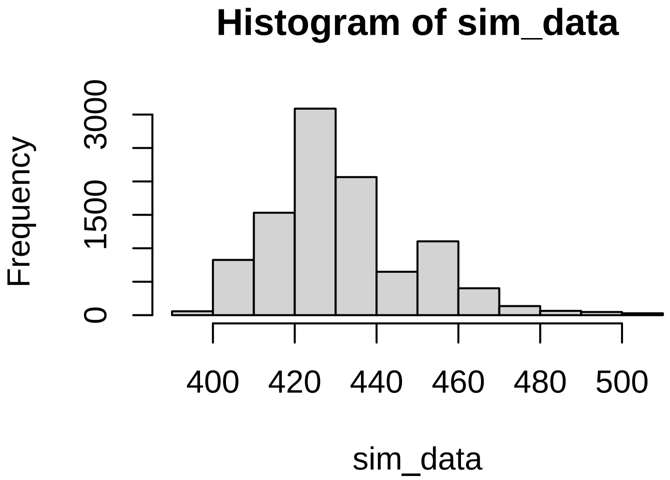
```
quantile(sim_data, c(.025, .975))
```

```
##    2.5%   97.5%
## 1217096 5841024
```

b. The 95 percent confidence interval for the mean viral load is [1217096, 5841024]. The lower bound is about 50 percent higher than the lower bound obtained using `t.test`.

Problem 49

```
set.seed(3850)
lp1 <- fosdata::brake
lp1 <- filter(lp1, age_group == "Old") %>%
  pull(latency_p1)
sim_data <- replicate(10000, {
  boot <- sample(lp1, replace = T)
  median(boot)
})
hist(sim_data)
```



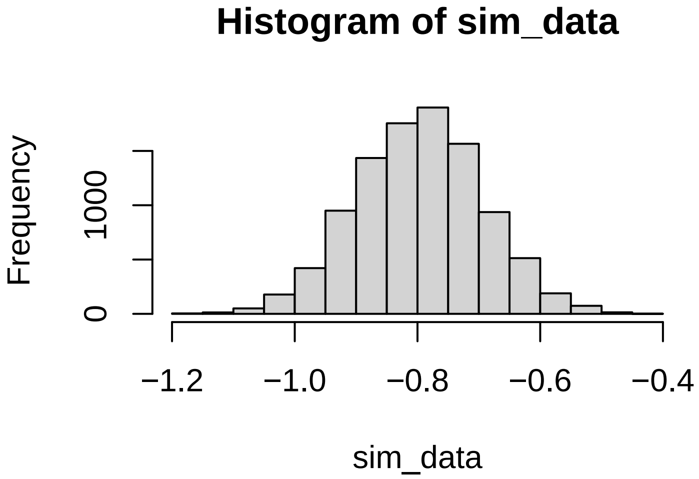
```
quantile(sim_data, c(.025, .975))
```

```
##     2.5%    97.5%
## 404.7436 473.1825
```

A 95 percent confidence interval for the median is $[405, 473]$.

Problem 50

```
set.seed(3850)
react <- ISwR::react
sim_data <- replicate(10000, {
  boot <- sample(react, replace = T)
  mean(boot)
})
hist(sim_data)
```



```
quantile(sim_data, c(.025, .975))
```

```
##       2.5%      97.5%
## -0.9970060 -0.5958084
```

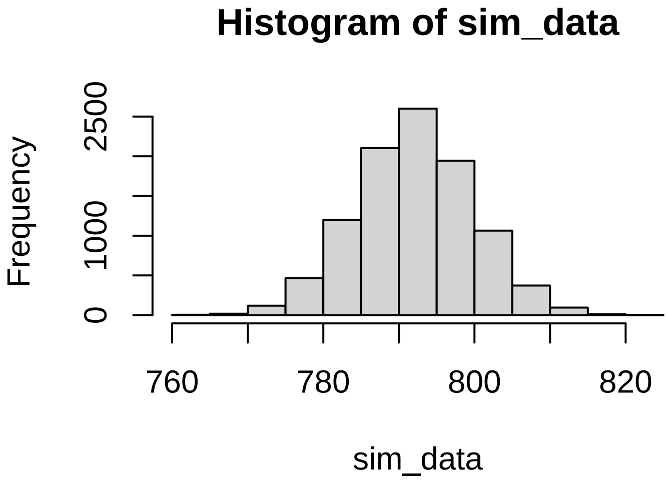
```
t.test(react)
```

```
##
##  One Sample t-test
##
## data:  react
## t = -7.7512, df = 333, p-value = 1.115e-13
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.9985214 -0.5942930
## sample estimates:
##  mean of x
## -0.7964072
```

A 95 percent CI is $[-1, -0.6]$, which compares very closely to `t.test`.

Problem 51

```
speed <- morley$Speed
speed_boot <- speed - mean(speed) + 792
sim_data <- replicate(10000, {
  boot <- sample(speed_boot, replace = T)
  mean(boot)
})
hist(sim_data)
```



```
2 * mean(sim_data > mean(speed))
```

```
## [1] 0
```

With p-value less than 1/10000 we reject the null hypothesis that the true mean of the experiment proposed by Michelson was 792.

9

Rank Based Tests – Solutions

Problem 1 $E[V] = \frac{30\cdot 31}{4} = 232.5$

Problem 2 $P = 0.0094$

```
2*(1-psignrank(600,40))
```

```
## [1] 0.009381349
```

Problem 3 a. The ranks of the n data points are $1, 2, 3, \ldots, n$. Each is positive with probability 0.5, and so the expected value of the sum of the positive ranks is

$$E(V) = 0.5 \cdot (1 + 2 + \cdots + n) = 0.5 \cdot \frac{n(n+1)}{2} = \frac{n(n+1)}{4}$$

b. When $x_n \to \infty$, the rank of x_n will be n and it will be positive. Then

$$E(V) = 0.5 \cdot (1 + 2 + \cdots + (n-1)) + n = 0.5 \cdot \frac{(n-1)n}{2} + n = \frac{n^2 - 3n}{4}$$

Problem 4 Differences from the hypothesized mean are -7, -3, -1, 5. The signed ranks are -4, -2, -1, 3. The test statistic $V = 3$. The expected value of V is $\frac{4\cdot 5}{4} = 5$, so the observed value is two away from the expected value. There are 16 possibilities for V, and they are 0, 1, 2, 3, 3, 4, 5, 6, 4, 5, 6, 7, 7, 8, 9, 10. Of these, ten are at least 2 away from the expected value 5. The p-value is $10/16 = .625$, not significant.

```
wilcox.test(c(-4,0,2,8),mu=3)
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  c(-4, 0, 2, 8)
## V = 3, p-value = 0.625
## alternative hypothesis: true location is not equal to 3
```

Problem 5

a. The effective type I error here is much too large when testing $H_0 : m = 1$.

```
sim_data <- replicate(10000, {
  dat <- rexp(20)
  wilcox.test(dat, mu = 1)$p.value
})
mean(sim_data < .05)
```

```
## [1] 0.1327
```

b. The effective type I error here is much too large when testing $H_0 : m = \log 2$.

```
sim_data <- replicate(10000, {
  dat <- rexp(20)
  wilcox.test(dat, mu = log(2))$p.value
})
mean(sim_data < .05)
```

```
## [1] 0.1012
```

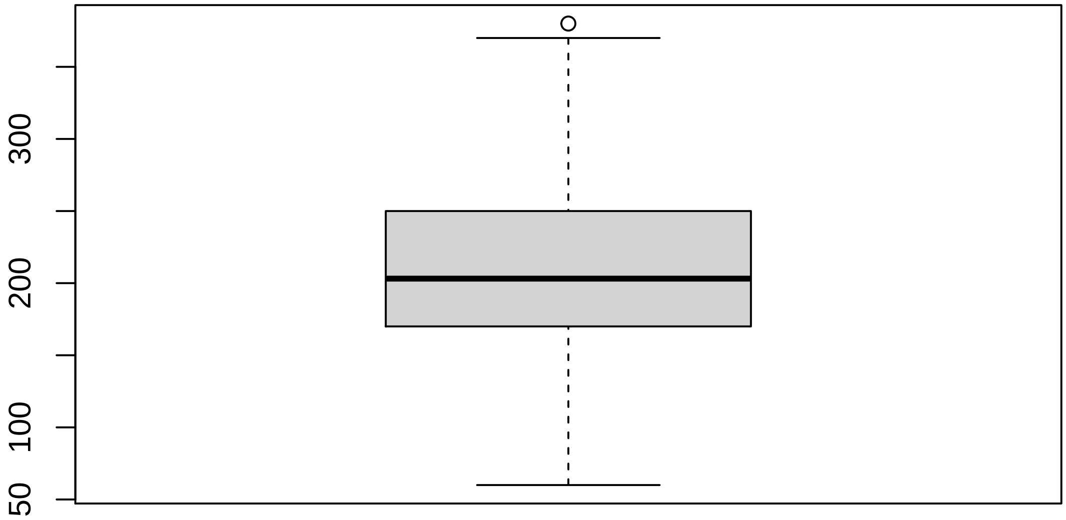
c. However, the effective type I error is just about right when testing $H_0 : m = 0.84$; perhaps still a bit higher than .05.

```
sim_data <- replicate(10000, {
  dat <- rexp(20)
  wilcox.test(dat, mu = 0.84)$p.value
})
mean(sim_data < .05)
```

```
## [1] 0.0532
```

Problem 6

```
mean200 <- fosdata::weight_estimate$mean200
boxplot(mean200)
```



```
wilcox.test(mean200, mu = 200)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  mean200
## V = 1708, p-value = 0.2052
## alternative hypothesis: true location is not equal to 200
```

With $P = 0.2052$ there is no evidence that the mean estimate differs from 200.

Problem 7 Ranks of x are 1, 3. Sum is 4. There are $\binom{5}{2} = 10$ ways to choose two ranks from 1,2,3,4,5. Their sums are 3, 4, 5, 5, 6, 6, 7, 7, 8, 9. Four of these are as far from 6 as the observed value 4. So the p-value is 4/10.

```
x <- c(2, 6)
y <- c(4, 9, 10)
wilcox.test(x,y)
```

```
##
##  Wilcoxon rank sum exact test
##
```

```
## data:  x and y
## W = 1, p-value = 0.4
## alternative hypothesis: true location shift is not equal to 0
```

Problem 8

```
x.clean <- c(53,58)
x.dirty <- c(69, 78, 87, 140)
wilcox.test(x.clean, x.dirty)$p.value
```

```
## [1] 0.1333333
```

```
t.test(x.clean, x.dirty)$p.value
```

```
## [1] 0.09571735
```

```
wilcox.test(x.clean, x.dirty, alternative = "less")$p.value
```

```
## [1] 0.06666667
```

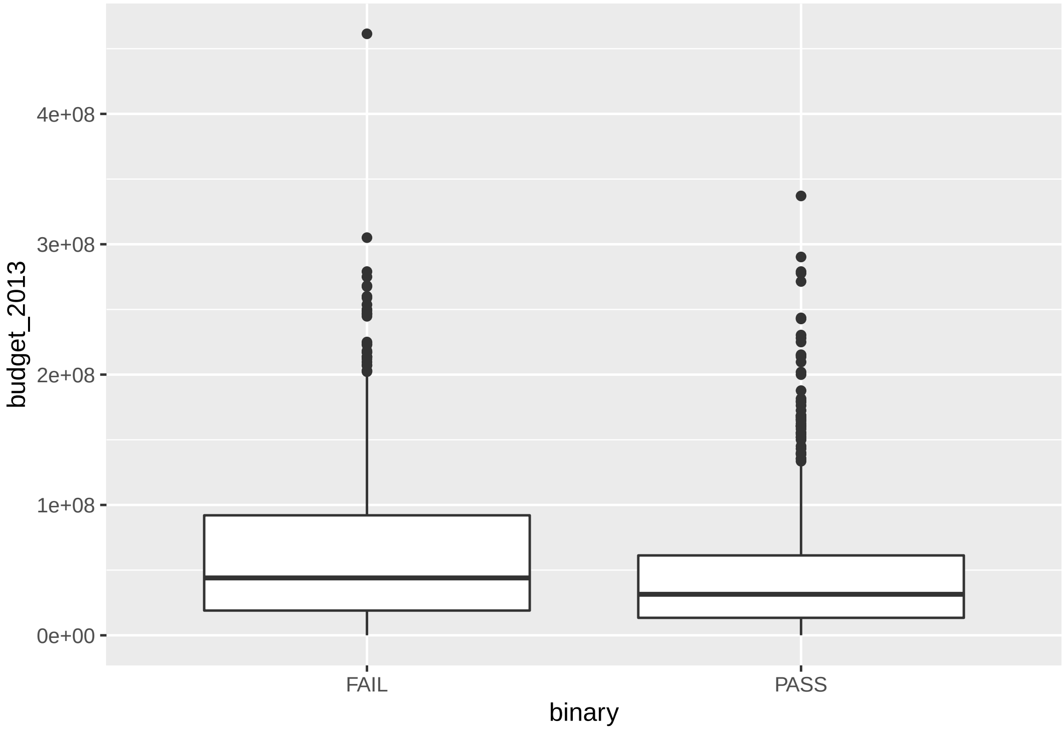
```
t.test(x.clean, x.dirty, alternative = "less")$p.value
```

```
## [1] 0.04785867
```

This experiment provides some evidence that swearing helps to endure the ice bucket, but it's not compelling.

Problem 9

```
fosdata::bechdel %>% ggplot(aes(x=binary, y=budget_2013)) + geom_boxplot()
```



```
wilcox.test(budget_2013 ~ binary, data=fosdata::bechdel)
```

```
##
```

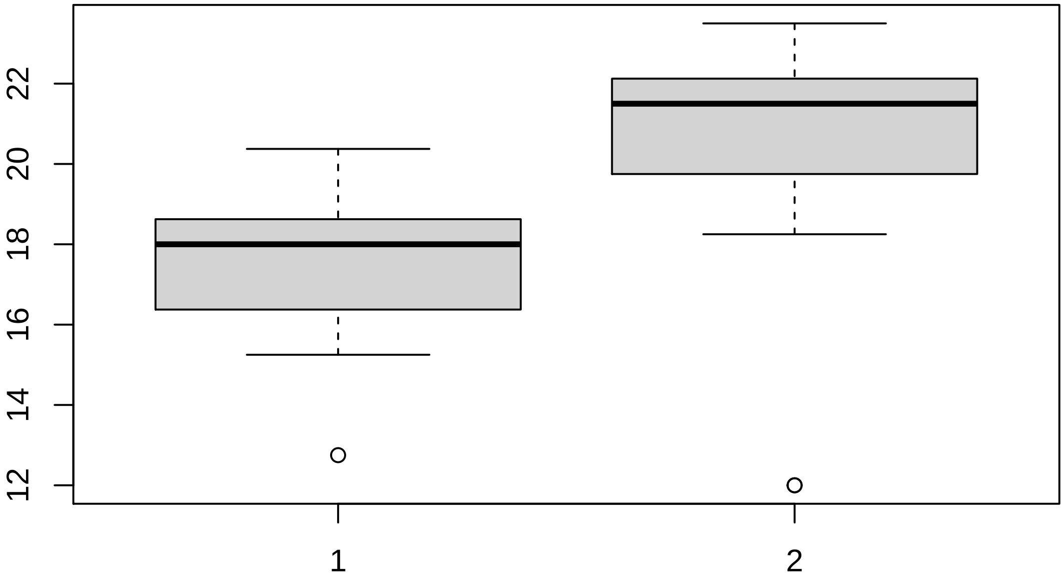
```
##  Wilcoxon rank sum test with continuity correction
##
## data:  budget_2013 by binary
## W = 468296, p-value = 1.093e-10
## alternative hypothesis: true location shift is not equal to 0
```

b. The difference in budget between movies that pass and movies that fail the Bechdel test is significant ($p = 1.1 \times 10^{-10}$). Movies that fail the test have higher budgets.

c. A t-test is inappropriate because the data is right skew and has many outliers.

Problem 10

```
with(HistData::ZeaMays, boxplot(self, cross))
```



The outliers will hinder the t-test.

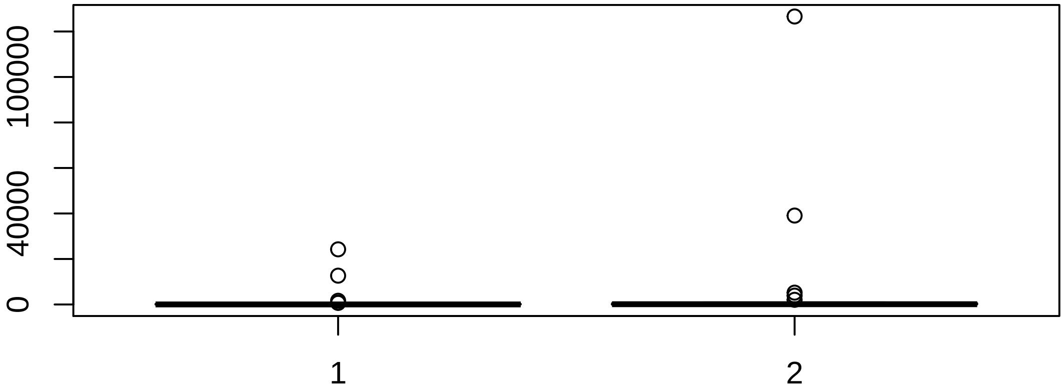
```
with(HistData::ZeaMays, wilcox.test(self, cross, paired=TRUE))
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  self and cross
## V = 24, p-value = 0.04126
## alternative hypothesis: true location shift is not equal to 0
```

There is a significant difference in height between the self and cross fertilized plants ($p = 0.04126$).

Problem 11

```
masks <- fosdata::masks
boxplot(masks$mask_fine,masks$no_mask_fine)
```



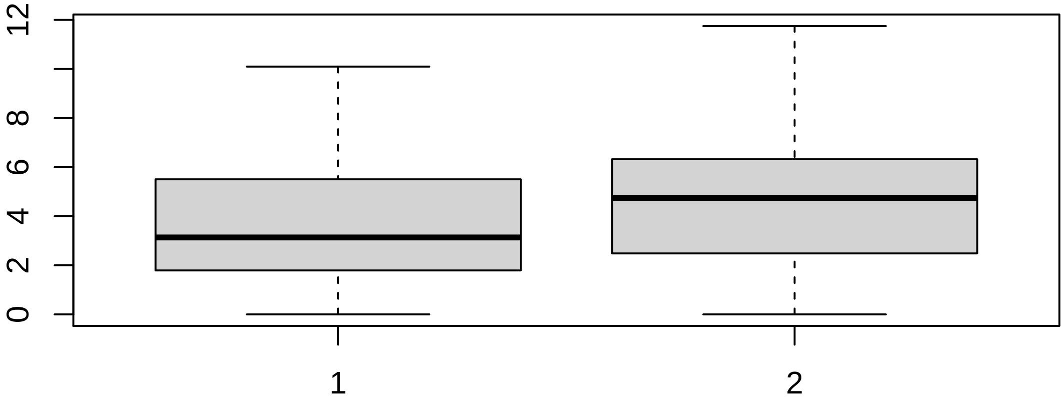
```
t.test(masks$mask_fine,masks$no_mask_fine, paired=TRUE)$p.value
```

```
## [1] 0.1867713
```

```
wilcox.test(masks$mask_fine,masks$no_mask_fine, paired=TRUE)$p.value
```

```
## [1] 0.0002140141
```

```
boxplot(log(1+masks$mask_fine), log(1+masks$no_mask_fine))
```



```
t.test(log(1+masks$mask_fine), log(1+masks$no_mask_fine), paired=TRUE)$p.value
```

```
## [1] 0.000290798
```

b. No significant difference ($P = 0.187$), but this test is inappropriate.

c. Both of the appropriate tests give $P < 0.0003$, suggesting there is a significant difference in influenza particles between mask and no mask. Wearing a mask helps.

Problem 12 a.

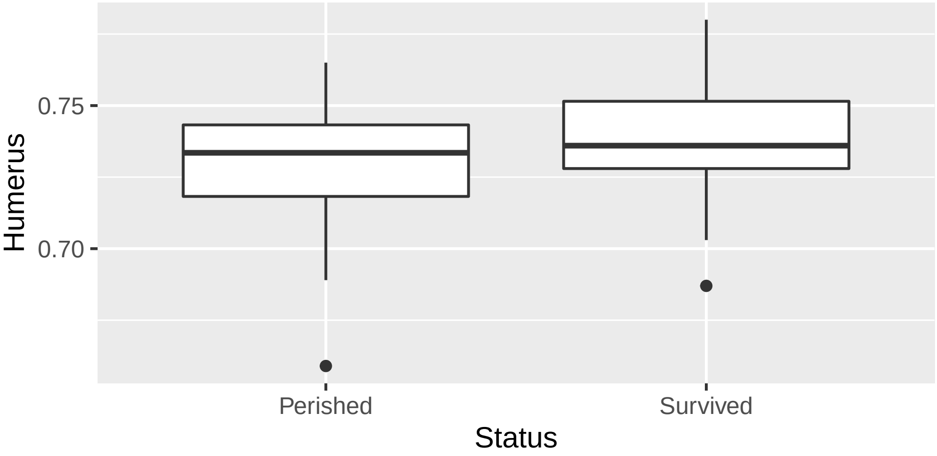
```
wilcox.test(c(-1,2,-3,-4,5))
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  c(-1, 2, -3, -4, 5)
## V = 7, p-value = 1
## alternative hypothesis: true location is not equal to 0
```

b. The P-value remains the same because doubling the values doesn't change their relative ranks.
c. The P-value remains the same because the ranks don't change.
d. The t-test is scale invariant, so doubling the values doesn't change. But the non-linear rescaling in part c does change the P-value for the t-test.

Problem 13

```
ggplot(Sleuth3::ex0221,aes(x=Status, y=Humerus)) + geom_boxplot()
```

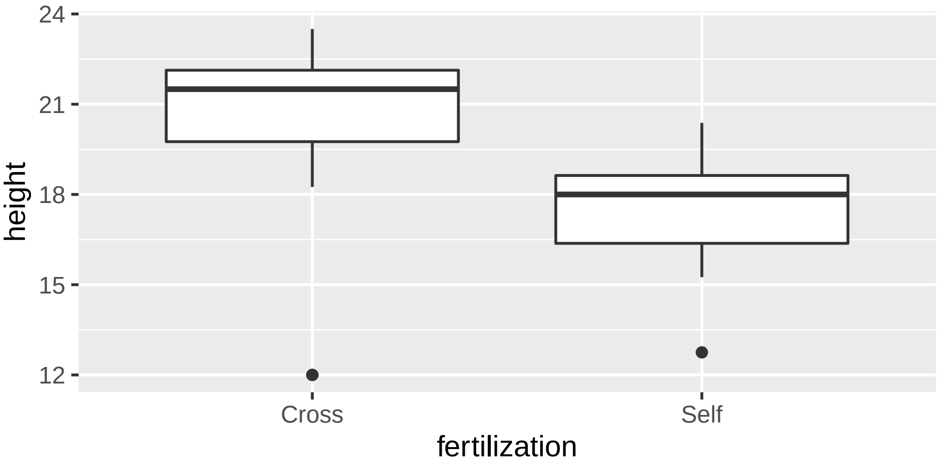


```
wilcox.test(Humerus ~ Status, data=Sleuth3::ex0221)
```

There is not significant evidence (p = 0.1718) that Humerus length is different in the two groups.

Problem 14

```
plants <- Sleuth3::ex0428
tidyr::pivot_longer(plants, everything(), names_to="fertilization", values_to="height") %>%
  ggplot(aes(x=fertilization, y=height)) + geom_boxplot()
```



```
wilcox.test(plants$Cross, plants$Self, paired=TRUE)
```

Both cross- and self-fertilized plants have one serious outlier (and they are not the same pair). The Wilcoxon test is appropriate, and gives significant evidence ($p = 0.04126$) of a difference in height between cross- and self-fertilized plants.

Problem 15

H_0 : antibody levels are the same with/without malaria symtpoms H_1 : antibody levels are not the same.

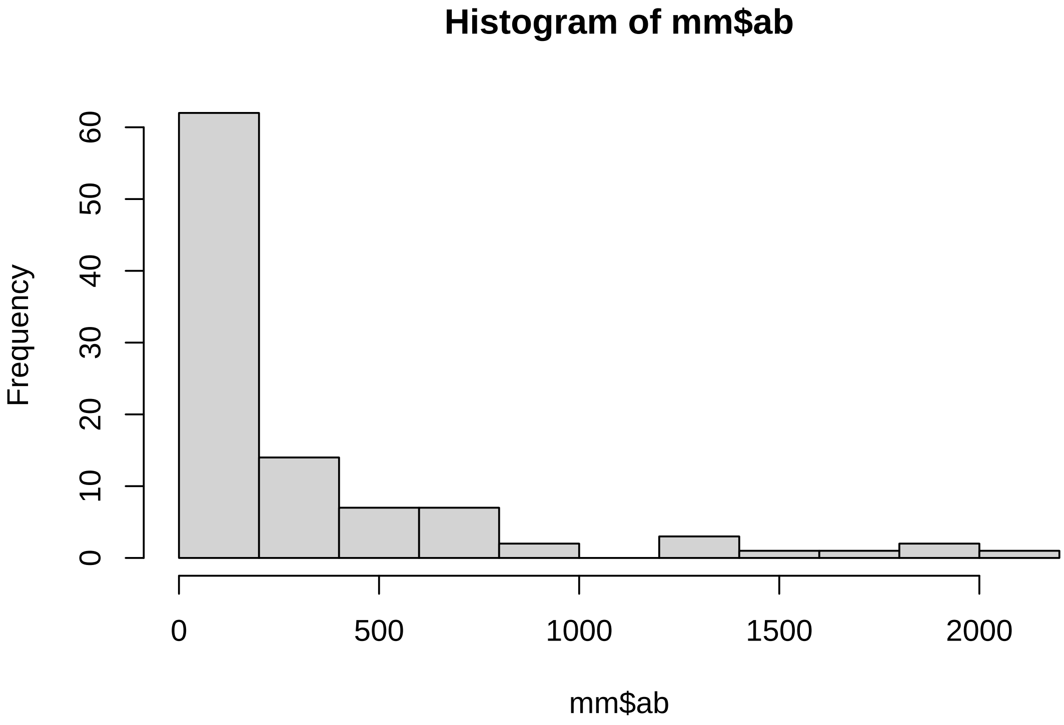
```
mm <- ISwR::malaria
wilcox.test(ab ~ mal, data = mm)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  ab by mal
## W = 1533.5, p-value = 2.127e-05
## alternative hypothesis: true location shift is not equal to 0
```

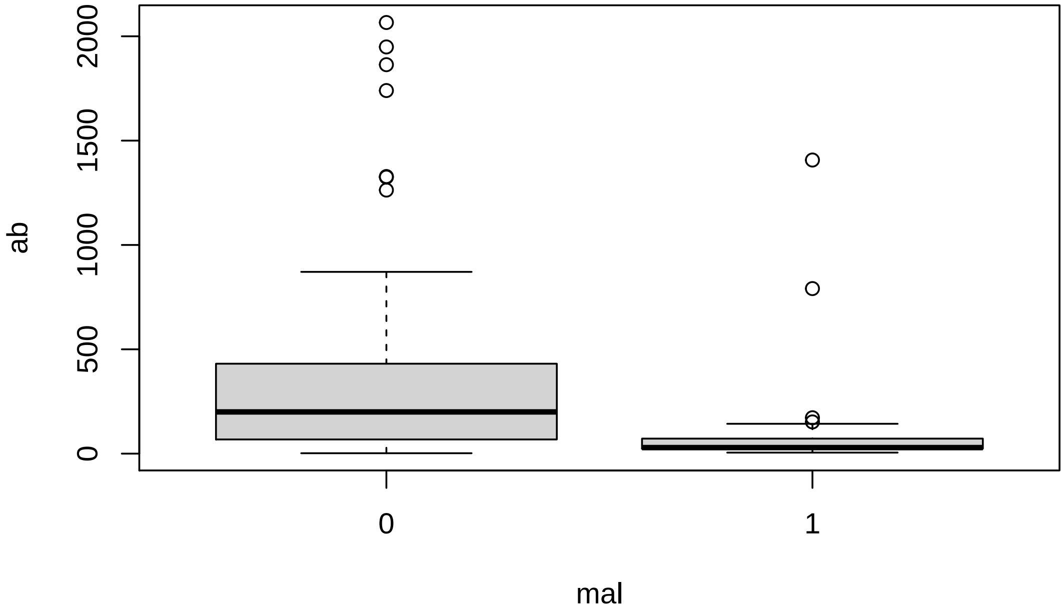
We reject the null hypothesis at the $\alpha = .05$ level.

b.

```
hist(mm$ab)
```



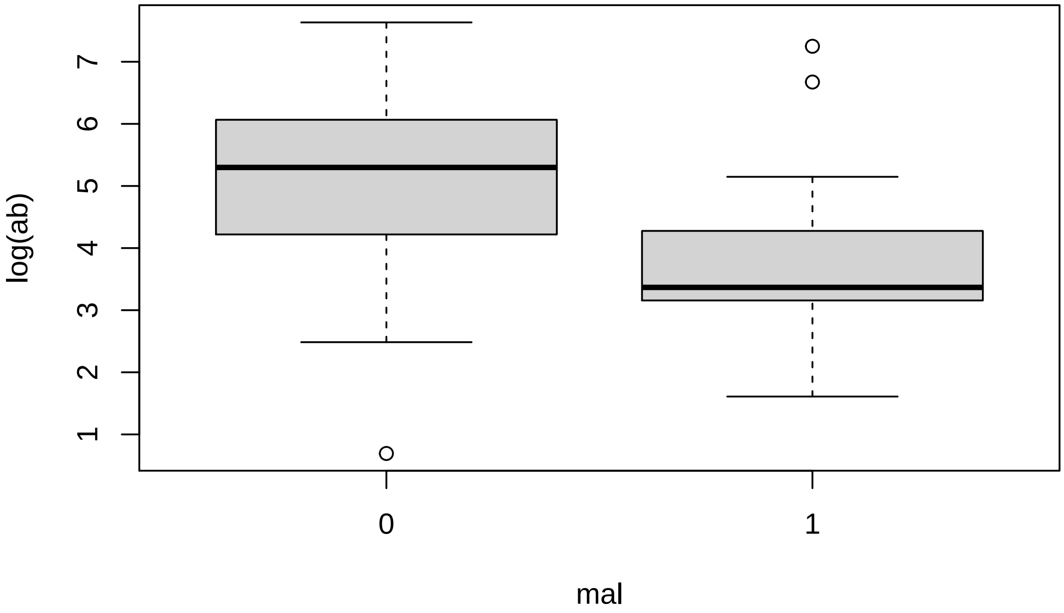
```
boxplot(ab ~ mal, data = mm)
```



T-test is not appropriate due to extreme skewness and outliers.

c.

```
boxplot(log(ab) ~ mal, data = mm)
```



This looks appropriate for a *t*-test.

d.

```
t.test(log(ab) ~ mal, data = mm)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  log(ab) by mal
## t = 4.6398, df = 52.256, p-value = 2.376e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.7831352 1.9764775
## sample estimates:
## mean in group 0 mean in group 1
##        5.126524        3.746717
```

We reject the null hypothesis that the expected value of the logs of the antibody levels are the same for the two groups at the $\alpha = .05$ level. This is the same conclusion we had in the Wilcoxon rank-sum test.

Problem 16

```
flint <- fosdata::flint
t.test(flint$Pb2, flint$Pb3, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  flint$Pb2 and flint$Pb3
## t = 1.6526, df = 270, p-value = 0.09957
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.270442 14.551320
## sample estimates:
## mean of the differences
##                6.640439
```

a. We find no significant difference in lead levels after 45 seconds and 2 minutes (p $=$ 0.10)
b. A plot of this data shows that it is very skew. The assumptions of the t-test are not satisfied.
c.

```
t.test(log(flint$Pb2), log(flint$Pb3), paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  log(flint$Pb2) and log(flint$Pb3)
## t = 9.0056, df = 270, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.4067597 0.6343692
## sample estimates:
## mean of the differences
##               0.5205644
```

Now the difference is significant (p $< 2.2 \times 10^{-16}$)

d.

```
wilcox.test(flint$Pb2, flint$Pb3, paired=TRUE)
```

```
##
```

```
##  Wilcoxon signed rank test with continuity correction
##
## data:  flint$Pb2 and flint$Pb3
## V = 30172, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

The difference is significant ($p < 2.2 \times 10^{-16}$)

e. Both the t test of the logged data and the Wilcoxon test agree the difference in lead levels is highly significant. We conclude that the log of the lead levels after running the water for 45 seconds are on average between .4 and .63 log parts per million more than the log of the lead levels after running the water for 2 minutes. To find a confidence interval for the Wilcoxon sign-rank test, we can set `conf.int = TRUE` as follows:

```
wilcox.test(flint$Pb2, flint$Pb3, conf.int = T, paired = T)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  flint$Pb2 and flint$Pb3
## V = 30172, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  0.5539025 1.1950478
## sample estimates:
## (pseudo)median
##       0.813967
```

This says that we estimate that the (pseudo)-median of the lead in the water after running for 45 seconds is .8 ppm higher than the pseudo-median of the lead in the water after running the water for 2 minutes.

Problem 17

```
sim_data <- replicate(10000, {
  dat <- runif(30, -.5, 1)
  c(t.test = t.test(dat)$p.value, wilcox = wilcox.test(dat)$p.value)
})
apply(sim_data, 1, function(x) mean(x < .05))
```

```
## t.test wilcox
## 0.8722 0.8049
```

We see that `t.test` is considerably more powerful (87 percent to 80 percent) in this context.

Problem 18

```
mean(replicate(10000,t.test(
  x <- c(rnorm(20),sample(c(-10,10),1)))$p.value < 0.05))
mean(replicate(10000,wilcox.test(
  x <- c(rnorm(20),sample(c(-10,10),1)))$p.value < 0.05))
```

The type I error rate for the t-test is near 0: it almost never rejects H_0. The Wilcoxon test rejects H_0 at close to the intended rate of 0.05.

Problem 19

```
mean(replicate(10000,t.test(rnorm(100,0.1,1))$p.value < .05))
mean(replicate(10000,wilcox.test(rnorm(100,0.1,1))$p.value < .05))
mean(replicate(10000,t.test(rnorm(1000,0.1,1))$p.value < .05))
mean(replicate(10000,wilcox.test(rnorm(1000,0.1,1))$p.value < .05))
```

The t-test rejects H_0 about 17% of the time when $n = 100$. The Wilcox test is not as powerful, and rejects H_0 only 16% of the time. When $n = 1000$ both tests are much more powerful and reject H_0 about 87% of the time. Again, the t test is a little better than the Wilcox test.

Problem 20 a. The true mean is 0, which means that H_0 is false. Type I error is impossible, there can only be Type II error b. 0 c. 0

```
mean(replicate(10000,t.test(rt(20,3),mu=1)$p.value > 0.05))
```

```
## [1] 0.2204
```

```
mean(replicate(10000,wilcox.test(rt(20,3),mu=1)$p.value > 0.05))
```

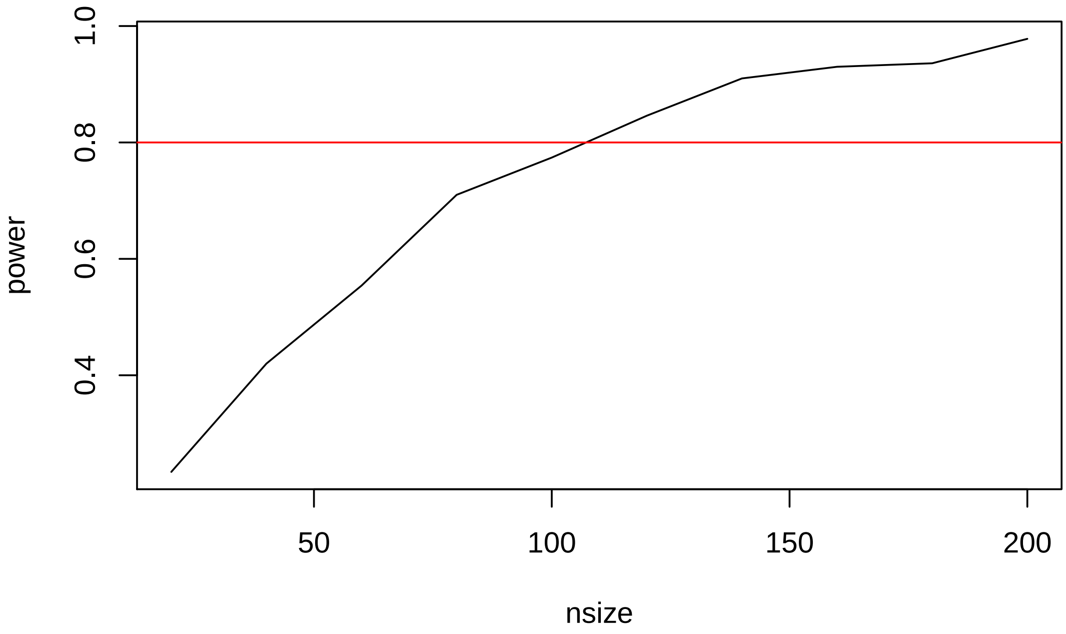
```
## [1] 0.128
```

d. `t.test` rejects the null about 78% of the time. The probability of type II error is about 0.22.
e. `wilcox.test` rejects the null about 88% of the time. The probability of type II error is about 0.12.
f. Wilcoxon is more powerful here, since it rejects the false null more frequently.

Problem 21

Recommend a sample size of around 100, or maybe a little more.

```
nsize <- seq(20,200,20)
power <- purrr::map(seq(20,200,20),
  function (nsize) {
    pvals <- replicate(500,
      wilcox.test(rnorm(nsize, mean=0), rnorm(nsize, mean=0.4))$p.value
      )
    mean(pvals < 0.05)
  })
plot(nsize,power,type='l')
abline(h=0.8,col='red')
```



Problem 22

Let's restrict to people who have seen only one of the movies and perform a Wilcoxon rank-sum test.

```
sleepy <- fosdata::movies %>%
  filter(stringr::str_detect(title, "Sleepless in S|While You Were S"))
sleepy <- sleepy %>%
  group_by(userId) %>%
  mutate(count = n()) %>%
  filter(count == 1)
wilcox.test(rating ~ title, data = sleepy)
```

```
## Warning in wilcox.test.default(x = c(1, 3, 3, 2.5, 4, 4.5, 3, 3.5, 4, 3, :
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  rating by title
## W = 711.5, p-value = 0.237
## alternative hypothesis: true location shift is not equal to 0
```

There is no statistically significant difference at the $\alpha = .05$ level in the ratings between a person who saw Sleepless and a person who saw While. The effect size is $A = .427$, which means we estimate that about 43 percent of people who see Sleepless in Seattle will rate it higher than people who see While You Were Sleeping.

```
effsize::VD.A(rating ~ title, data = sleepy)
```

```
##
## Vargha and Delaney A
##
## A estimate: 0.4273273 (negligible)
```

Problem 23

```
sharks <- fosdata::sharks
head(sharks)
```

```
##      av     music scary dangerous vicious peaceful beautiful graceful
## 1 audio   ominous     7         7       4        1         3        3
## 2 video uplifting     7         4       4        4         7        7
## 3 video   ominous     2         4       2        7         7        7
## 4 video   silence     5         6       3        4         5        5
## 5 audio   silence     4         5       5        1         1        1
## 6 video   silence     4         5       3        4         4        4
##   free_response conserve gender age race_ethnicity annual_income
## 1      dramatic        6      1  27              1             2
## 2          many        6      2  60              1             1
## 3 unpredictable        4      1  54              1             5
## 4        creepy        5      2  30              1             3
## 5          Ugly        7      2  21              1             1
## 6   fascinating        6      2  35              1             3
##   political_views
## 1               3
## 2               4
## 3               3
## 4               3
## 5               4
## 6               2
```

```
sharks %>%
  filter(av == "video") %>%
  filter(stringr::str_detect(music, "uplif|omin")) %>%
  with(wilcox.test(vicious ~ music))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  vicious by music
## W = 6343.5, p-value = 0.02904
## alternative hypothesis: true location shift is not equal to 0
```

There is a statistically significant difference in the participants' ratings of how vicious sharks are based on the music they heard.

```
sharks %>%
  filter(av == "video") %>%
  filter(stringr::str_detect(music, "uplif|omin")) %>%
  mutate(music = factor(music, levels = c("ominous", "uplifting"))) %>%
  with(effsize::VD.A(vicious ~ music))
```

```
##
## Vargha and Delaney A
##
## A estimate: 0.5865465 (small)
```

The effect size is $A = .587$, which means roughly 59 percent of people rate sharks as more vicious after hearing ominous music while watching the video than other people do after watching the uplifting music video.

Problem 24

```
wilcox.test(length ~ region, data = fosdata::plastics)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length by region
## W = 4678, p-value = 5.163e-05
## alternative hypothesis: true location shift is not equal to 0
```

We reject the null hypothesis that the populations are the same.

b. In the paper they computed the following test statistic: $W + 134 \times 135/2 = 13723$.

```
4678 + 134 * 135 / 2
```

```
## [1] 13723
```

c. Vargha and Delaney's A is about 0.35, as given by:

```
effsize::VD.A(length ~ region, data = fosdata::plastics)
```

```
##
## Vargha and Delaney A
##
## A estimate: 0.345648 (small)
```

This roughly means that if you randomly select two pieces of plastic as in the experiment, about 35 percent of the time the plastic from the Arctic will be longer than the plastic from Europe, where we assume that ties are broken by flipping a fair coin.

Problem 25 Solution needed.

We didn't discuss effect size for paired tests or one sample tests. This problem probably needs to go.

Problem 26

We choose a few values of N and see that the power seems to be getting close to 1. Indeed, when $N = 100$, all 1000 trials resulted in rejecting the null hypothesis.

```
Ns <- c(20, 50, 100)
sapply(Ns, function(N) {
  sim_data <- replicate(1000, {
    dat1 <- rnorm(N)
    dat2 <- rnorm(N, 1, 1)
    wilcox.test(dat1, dat2)$p.value
  })
  mean(sim_data < .05)
})
```

```
## [1] 0.823 0.997 1.000
```

Problem 27

a. The probability appears to be about 0.5 that $X > Y$.

```
mean(rnorm(10000) > rnorm(10000, 0, 5))
```

```
## [1] 0.5141
```

b.

```
set.seed(3850)
Ns <- c(10, 100, 1000, 5000)
sapply(Ns, function(N) {
  sim_data <- replicate(1000, {
    dat1 <- rnorm(N)
    dat2 <- rnorm(N, 0, 5)
    wilcox.test(dat1, dat2)$p.value
  })
  mean(sim_data < .05)
})
```

```
## [1] 0.067 0.082 0.096 0.080
```

We estimate the probabilities of rejecting the null hypothesis to be 0.067, 0.082, 0.096 and 0.080.

c. The Wilcoxon rank sum test does not appear to be consistent in this context.

10

Tabular Data – Solutions

Problem 1

```
cern <- fosdata::cern %>%
  mutate(platform = factor(platform, levels = c("Facebook", "Twitter", "Tw Frenc", "Google+", "I
         type = factor(type, levels = c("News", "GWII", "TBT", "Wow")))
xtabs(~ type + platform, data = cern) %>%
  addmargins()
```

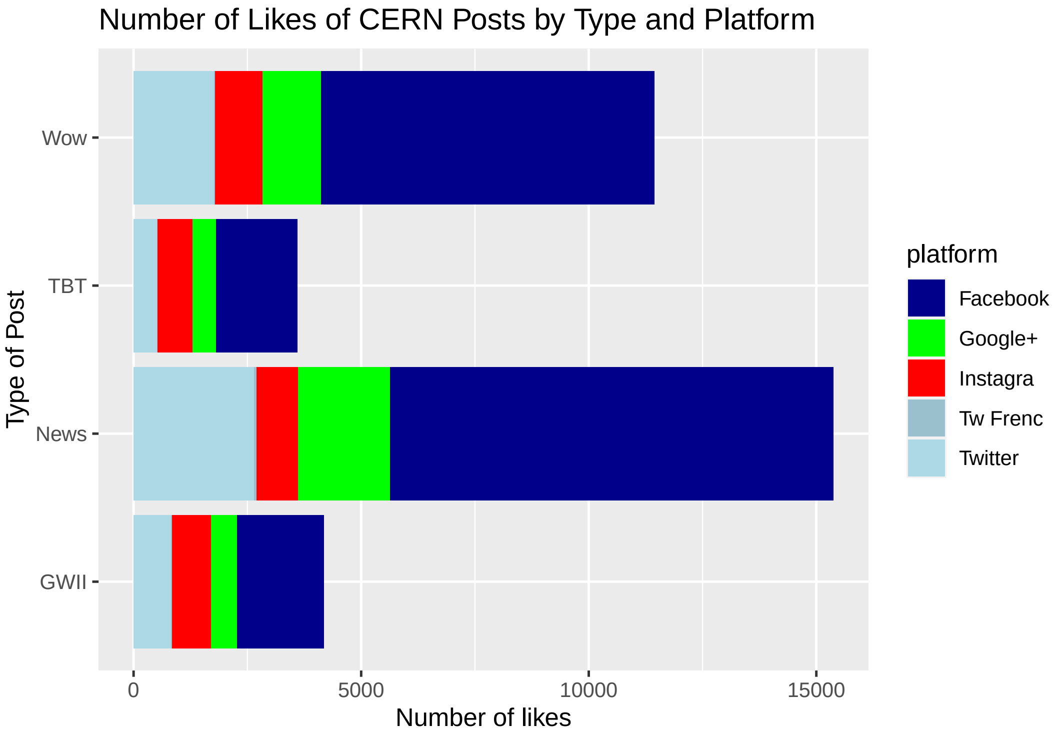
```
##       platform
## type   Facebook Twitter English Twitter French Google+ Instagram Sum
##   News       24              23             17      22         8  94
##   GWII        8               8              8       8         8  40
##   TBT         8               8              8       8         8  40
##   Wow         8               8              8       8         8  40
##   Sum        48              47             41      46        32 214
```

Problem 2

Colored by platform; it might be fun to try to get the colors to match the official platform colors more closely.

```
fosdata::cern %>%
  group_by(platform, type) %>%
  summarize(count = sum(likes)) %>%
  ggplot(aes(x = type, y = count, fill = platform)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("darkblue", "green", "red", "lightblue3", "lightblue")) +
  coord_flip() +
  labs(title = "Number of Likes of CERN Posts by Type and Platform",
       y = "Number of likes",
       x = "Type of Post")
```

```
## `summarise()` has grouped output by 'platform'. You can override using the
## `.groups` argument.
```



Problem 3

By hand, we sum all of the outcomes that are as or less likely than 12:

```
p1 <- dbinom(12, 20, 0.4)
binom_probs <- dbinom(0:20, 20, 0.4)
sum(binom_probs[binom_probs <= p1])
```

```
## [1] 0.1074783
```

and we confirm it is the same as `binom.test`.

```
binom.test(12, 20, p = 0.4)$p.value
```

```
## [1] 0.1074783
```

Problem 4

We compute the p-value using a normal approximation and continuity correction.

```
p <- 0.4
mu <- 100 * p
sdev <- sqrt(100 * p * (1 - p))

pnorm(33.5, mu, sdev) * 2 #this would be 33 without continuity correction
```

```
## [1] 0.1845726
```

We check with `prop.test`.

```
prop.test(33, 100, p = 0.4)$p.value
```

```
## [1] 0.1845726
```

Problem 5

Based on this data, it appears that Shaq was indeed better than 50 percent at the line, 95 percent confidence interval $[.518, .537]$.

```r
binom.test(x = 5935, n = 11252)
```

```
##
##  Exact binomial test
##
## data:  5935 and 11252
## number of successes = 5935, number of trials = 11252, p-value =
## 5.954e-09
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5181866 0.5367227
## sample estimates:
## probability of success
##              0.5274618
```

Problem 6

```r
binom.test(20245, 40000, conf.level = .99)
```

```
##
##  Exact binomial test
##
## data:  20245 and 40000
## number of successes = 20245, number of trials = 40000, p-value =
## 0.01448
## alternative hypothesis: true probability of success is not equal to 0.5
## 99 percent confidence interval:
##  0.4996730 0.5125756
## sample estimates:
## probability of success
##               0.506125
```

99% CI [0.4996730, 0.5125756]

$H_0 : p = .5, Ha : p \neq .5$, where p is the probability that a coin will land on the same side that it was tossed from

Yes, sufficient evidence to reject at the alpha = .05 level, p = .01448

Problem 7

```r
mean(replicate(10000, {
  coin_tosses <- sample(0:1, 100, T)
  any(sapply(10:100, function(x) {
    binom.test(x = sum(coin_tosses[1:x]), n = x)$p.value
  }) < .05)
}))
```

About 20 percent.

Problem 8

a. $H_0 : p = 1/6$ vs $H_a : p \neq 1/6$, where p is the true percentage of times the die comes up 6.

b. The thing that is random here is not the number of successes, but the number of rolls of the die. `binom.test` computes probabilities of getting various numbers of successes on the number of trials that we ran, and the number of successes is not random in this experiment.

c.

```
pr <- dnbinom(x = 460, size = 100, prob = 1/6)
neg_binom_probs <- dnbinom(0:1000, size = 100, prob = 1/6)
sum(neg_binom_probs[neg_binom_probs >= pr])
```

```
## [1] 0.4848174
```

We get a p-value of 0.485. Note that applying `binom.test(100, 560, 1/6)` gives a slighlty different p-value of .4609.

d. We do not reject H_0.

Problem 9

The p-value is $p = .1153$. Note that this test statistic is possible with the combination `c(56, 56, 38)`.

```
pchisq(4.32, df = 2, lower.tail = FALSE)
```

```
## [1] 0.1153251
```

Problem 10

```
chisq.test(x = c(1357, 1321, 1946, 1182, 2052, 2142), p = c(.14, .13, .2, .12, .2, .21))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(1357, 1321, 1946, 1182, 2052, 2142)
## X-squared = 5.5799, df = 5, p-value = 0.3493
```

We do not reject the null hypothesis that the color of candies follows the distribution given in the problem.

Problem 11

```
benford <- log10(1 + 1 / (1:9))
first_digit <- substr(as.character(fosdata::bechdel$budget),1,1)
chisq.test(table(first_digit),p=benford)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(first_digit)
## X-squared = 7.4061, df = 8, p-value = 0.4935
```

```
first_digit <- substr(as.character(fosdata::bechdel$intgross),1,1)
chisq.test(table(first_digit),p=benford)
```

```
##
##  Chi-squared test for given probabilities
```

```
##
## data:  table(first_digit)
## X-squared = 4.8665, df = 8, p-value = 0.7717
```

```
first_digit <- substr(as.character(fosdata::bechdel$domgross),1,1)
fosdata::bechdel[which.min(first_digit),"title"]
```

```
## [1] "I Come with the Rain"
```

```
first_digit <- first_digit[first_digit != '0']
chisq.test(table(first_digit),p=benford)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(first_digit)
## X-squared = 9.9697, df = 8, p-value = 0.2672
```

All budget and earnings figures are consistent with Benford's Law.

Problem 12 Generally population data follows Benford's law, as long as all incorporated cities and towns are included.

Problem 13

```
goals <- fosdata::world_cup %>%
  filter(competition == "2014 FIFA Men's World Cup") %>%
  tidyr::pivot_longer(cols = contains("score"), values_to = "score") %>%
  pull(score)   # pull extracts the "score" column as a vector
table(goals)
```

```
## goals
##  0  1  2  3  4  5  7
## 37 44 27 12  5  2  1
```

```
lambda <- mean(goals)
expected_goals <- 104 * c(dpois(0:3, lambda),
                          ppois(3, lambda, lower.tail = FALSE))
observed_goals <- c(37, 44, 27, 12, 8)
chi_2 <- sum((observed_goals - expected_goals)^2 / expected_goals)
chi_2
```

```
## [1] 7.357173
```

```
pchisq(chi_2, df = 3, lower.tail = FALSE)
```

```
## [1] 0.06134411
```

We do not reject the null at the $\alpha = 0.05$ level. The observed data is consistent with a Poisson distribution ($P = 0.06$).

Problem 14

We first confirm the table given in the text:

```
aa <- fosdata::austen %>%
  filter(stringr::str_detect(novel, "Pride"),
         chapter == 1)
nn <- aa %>%
  group_by(word) %>%
  summarize(num_repeat = n() - 1) %>%
  count(num_repeat)
print(t(nn))
```

```
##            [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## num_repeat    0    1    2    3    4    5    6    7    8     9    10    11    12
## n           201   50   16   13   12    2    5    5    2     1     4     2     1
##            [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
## num_repeat    13    14    16    17    20    21    28    30
## n              2     1     2     1     1     1     1     1
```

If it is Poisson, an estimate for the mean is 1.64:

```
mu <- sum(nn$num_repeat * nn$n)/sum(nn$n)
mu
```

```
## [1] 1.638889
```

We will need to combine some of the cells. For $\lambda = 1.64$, we will need to combine the cells larger than 4 in order to have expected values all at least 5.

```
ceiling(ppois(1:10, mu, lower.tail = FALSE) * sum(nn$n))
```

```
##  [1] 158  74  28   9   3   1   1   1   1   1
```

We perform the test as follows:

```
chisq.test(x = c(nn$n[1:5], sum(nn$n[6:nrow(nn)])), p = c(dpois(0:4, mu), ppois(4, mu, lower.tai
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(nn$n[1:5], sum(nn$n[6:nrow(nn)]))
## X-squared = 478.73, df = 5, p-value < 2.2e-16
```

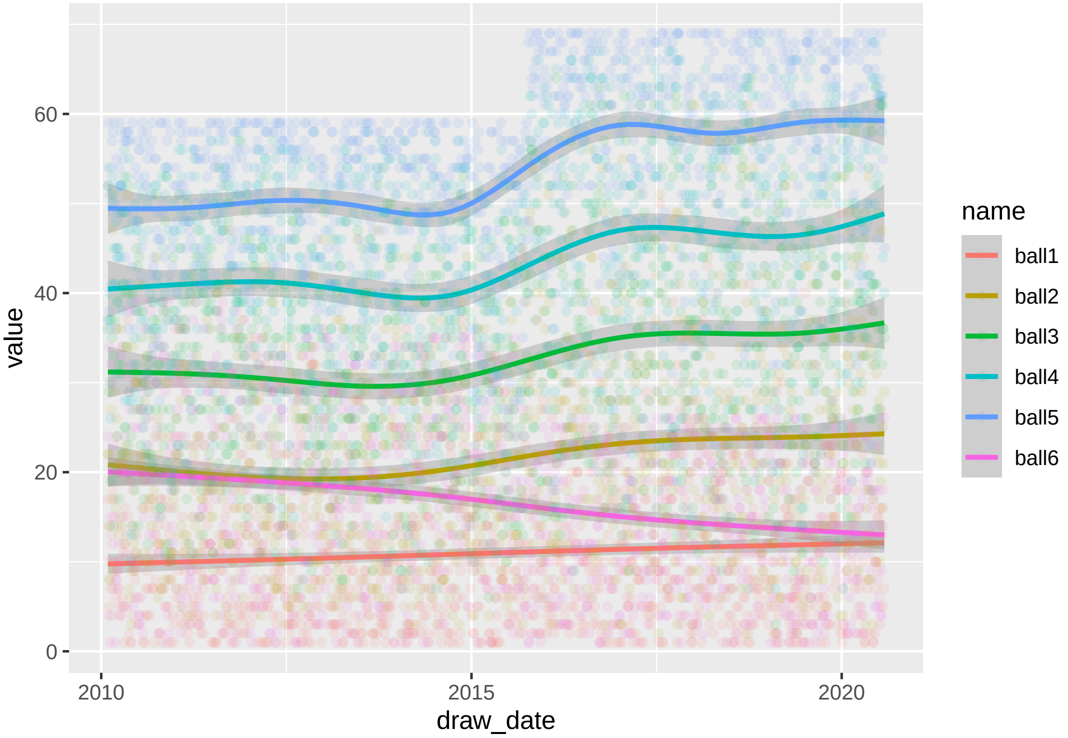
We reject the null hypothesis that the number of repetitions of words follows a Poisson distribution.

Problem 15

a. A couple of things: the balls are arranged in increasing order and there was a change in the rules at some point after 2015.

```
pow <- fosdata::powerball
pow <- pow %>%
  janitor::clean_names() %>%
  tidyr::pivot_longer(cols = matches("^b")) %>%
  mutate(draw_date = lubridate::ymd(draw_date))

ggplot(pow, aes(x = draw_date, y = value, color = name)) +
  geom_smooth() +
  geom_point(alpha = .1)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



b. We reject the null hypothesis that all numbers ever drawn follow a uniform distribution.

```
observed_alltime <- pow %>%
  count(value) %>%
  pull(n)
chisq.test(observed_alltime)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed_alltime
## X-squared = 579.1, df = 68, p-value < 2.2e-16
```

c. We fail to reject the null hypothesis that the ball numbers drawn after October 4, 2015 are uniformly distributed.

```
observed <- pow %>%
  filter(draw_date > lubridate::ymd("2015-10-04"), !stringr::str_detect(name, "6")) %>%
  count(value) %>%
  pull(n)
chisq.test(observed)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 70.209, df = 68, p-value = 0.4034
```

d. Ball 1 is not uniform because it is the minimum of the 5 white balls.

```
observed_ball1 <- pow %>%
  filter(draw_date > lubridate::ymd("2015-10-04"), stringr::str_detect(name, "1")) %>%
  count(value) %>%
  pull(n)
chisq.test(observed_ball1)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed_ball1
## X-squared = 371.49, df = 41, p-value < 2.2e-16
```

Problem 16

a. We estimate the mean to be 1.32 and the standard deviation to be 0.39.

```
adipose <- fosdata::adipose
mu <- mean(adipose$hdl)
sdev <- sd(adipose$hdl)
mu
```

```
## [1] 1.32037
```

```
sdev
```

```
## [1] 0.394878
```

b. The dividing points are:

```
breaks <- qnorm(seq(0, 1, length.out = 8), mu, sdev)
breaks
```

```
## [1]      -Inf 0.8988103 1.0968896 1.2492875 1.3914533 1.5438511 1.7419305
## [8]       Inf
```

c. We get the following tabled values.

```
observed <- table(cut(adipose$hdl, breaks = breaks))
observed
```

```
##
## (-Inf,0.899]  (0.899,1.1]   (1.1,1.25]  (1.25,1.39]  (1.39,1.54]  (1.54,1.74]
##            9           15           11           15           11           10
##  (1.74, Inf]
##           10
```

d. The χ^2 test statistic is:

```
expected <- nrow(adipose) /8
chisq <- sum((observed - expected)^2/expected)
chisq
```

```
## [1] 4.973765
```

e. The p-value associated with this test of normality is $p = 0.42$, with degrees of freedom number of bins - 1 - number of parameters estimated from data = 8 - 1 - 2 = 5.

```
pchisq(chisq, 5, lower.tail = FALSE)
```

```
## [1] 0.4190903
```

f. There is not sufficient evidence to conclude that HDL is not normally distributed.

Problem 17

```
temp <- UsingR::normtemp$temperature
splits <- qnorm(seq(0, 1, length.out = 11), mean(temp), sd(temp))
bins <- sapply(1:130, function(x) sum(temp[x] < splits))
pchisq(sum((table(bins) - 13)^2/13), df = 7, lower.tail = TRUE) #p-value pf 0.313, do not reject
```

```
## [1] 0.3118989
```

Problem 18

```
binom.test(162, 300)
```

```
##
##  Exact binomial test
##
## data:  162 and 300
## number of successes = 162, number of trials = 300, p-value = 0.1841
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4817787 0.5974236
## sample estimates:
## probability of success
##                   0.54
```

95 percent confidence interval:

0.4817787 0.5974236

$H_0 : p = 5, H_a : p \neq .5$, where p is the probability that this participant will toss Heads while trying to toss heads

Not sufficient evidence to reject, p = 0.1841

```
prop.test(c(162, 175), c(300, 300))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(162, 175) out of c(300, 300)
## X-squared = 0.97483, df = 1, p-value = 0.3235
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.12599523  0.03932856
## sample estimates:
##    prop 1    prop 2
## 0.5400000 0.5833333
```

Not sufficient evidence that the two participants have different probabilities of getting heads

Problem 19

```
prop.test(x = c(265, 308), n = c(5036, 4426))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
```

```
## data:  c(265, 308) out of c(5036, 4426)
## X-squared = 11.625, df = 1, p-value = 0.0006508
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.026886740 -0.007048591
## sample estimates:
##     prop 1     prop 2
## 0.05262113 0.06958879
```

We reject the null hypothesis ($p = 0.0006508$) that the proportions of patients receiving CABG in the two groups is the same. A 95 percent confidence interval for the difference in proportions is $[0.007, 0.027]$, with the younger group getting the procedure more frequently than the older group.

Problem 20

```
binom.test(143, 175)
```

```
##
##  Exact binomial test
##
## data:  143 and 175
## number of successes = 143, number of trials = 175, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.7517804 0.8714358
## sample estimates:
## probability of success
##              0.8171429
```

Reject H_0 that the probability that people will predict H on the first toss is .5, conclude that it is higher than .5 (p < .05).

Problem 21

```
prop.test(c(47, 16), c(54, 51))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(47, 16) out of c(54, 51)
## X-squared = 31.583, df = 1, p-value = 1.911e-08
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.3818790 0.7314108
## sample estimates:
##    prop 1    prop 2
## 0.8703704 0.3137255
```

Yes, sufficient evidence to reject H_0. The order that you name Heads/Tails or Tails/Heads impacts the probability that a person puts Heads or Tails as the first item.

Problem 22

There is not sufficient evidence $p = 0.07$ to conclude that the proportion of children who prefer blue and white striped marble to a red marble is different than the proportion of children who prefer 3 red + 1 blue and white to 4 red marbles.

```
prop.test(x = c(43, 32), n = c(48, 44))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(43, 32) out of c(48, 44)
## X-squared = 3.2833, df = 1, p-value = 0.06999
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.01065217  0.34777338
## sample estimates:
##    prop 1    prop 2
## 0.8958333 0.7272727
```

Problem 23

```
prop.test(c(30, 51), c(46, 51 + 78))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(30, 51) out of c(46, 51 + 78)
## X-squared = 7.9926, df = 1, p-value = 0.004697
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.08064158 0.43300857
## sample estimates:
##    prop 1    prop 2
## 0.6521739 0.3953488
```

We reject the null hypothesis that the proportion of extended stays is the same for scald patients as for burn patients (p = .004697).

Problem 24

```
bb <- babynames::babynames
bb <- filter(bb, year %in% c(1978, 1982))
bb %>%
  mutate(is_reagan = name == "Reagan") %>%
  with(xtabs(n ~ is_reagan + year))
```

```
##          year
## is_reagan    1978    1982
##     FALSE 3174048 3507578
##     TRUE      220      86
```

```
prop.test(c(220, 86), c(3174268, 3507664 ))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(220, 86) out of c(3174268, 3507664)
## X-squared = 72.024, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
```

```
##  3.396719e-05 5.561196e-05
## sample estimates:
##       prop 1       prop 2
## 6.930732e-05 2.451774e-05
```

Reject the null hypothesis that the probability of a baby named Reagan is the same in both years.

The probability was higher in 1978 than in 1982.

Problem 25

We start by filtering the desired observations and creating a cross table.

```
dogs1 <- fosdata::dogs %>%
  filter(trial == 1, condition == 1)
xtabs( ~ conform + start_direction, data = dogs1)
```

```
##        start_direction
## conform  0  1
##       0 13  8
##       1  1 10
```

Based on this table, it seems that directing dogs away from the trained position might well impact their decision.

```
chisq.test(xtabs( ~ conform + start_direction, data = dogs1))
```

```
## Warning in chisq.test(xtabs(~conform + start_direction, data = dogs1)): Chi-
## squared approximation may be incorrect

## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  xtabs(~conform + start_direction, data = dogs1)
## X-squared = 6.1766, df = 1, p-value = 0.01295
```

With a p-value of 0.01295, we would reject the null hypothesis that the facing direction is independent of whether the dog stays or switches. Care should be taken, since the χ^2 approximation may be incorrect.

** Problem 26**

a. Here is the table. It appears that perhaps uplifting music may well lead to a lower danger score.

```
sharks <- fosdata::sharks
shark_xtabs <- xtabs(~ music + dangerous, data = sharks)
shark_xtabs
```

```
##            dangerous
## music        1  2  3  4  5  6  7
##   ominous    3  3  8 17 42 68 62
##   silence    3  6 15 23 39 55 66
##   uplifting 19 11 16 17 42 48 53
```

b. We reject the null hypothesis at the $\alpha = .05$ level that the level of dangerous has the same distribution for each type of music heard.

```
chisq.test(shark_xtabs)
```

```
##
##  Pearson's Chi-squared test
##
## data:  shark_xtabs
## X-squared = 34.759, df = 12, p-value = 0.0005115
```

Problem 27 Solution needed. Leaving this because there is a warning message when you run prop.test with this data.

We infer that 61/65 and 18/26 test-takers passed in the two groups. There is sufficient evidence $p = .005234$ to conclude that the proportions of people who pass in the two groups is different.

```
prop.test(c(61, 18), c(65, 26))
```

```
## Warning in prop.test(c(61, 18), c(65, 26)): Chi-squared approximation may be
## incorrect
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(61, 18) out of c(65, 26)
## X-squared = 7.7969, df = 1, p-value = 0.005234
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.03245253 0.45985516
## sample estimates:
##    prop 1    prop 2
## 0.9384615 0.6923077
```

Problem 28

```
aa <- fosdata::bicycle_signage
xtabs(~ bike_move_right2 + treatment, data = aa)
```

```
##                   treatment
## bike_move_right2 1_None 2_STR 3_SLM 4_BMUFL
##       0_Disagree    326   268   315     336
##       1_Agree       163   154   139     123
```

```
xtabs(~ bike_move_right2 + treatment, data = aa) %>%
  prop.table(margin = 2)
```

```
##                   treatment
## bike_move_right2    1_None     2_STR     3_SLM   4_BMUFL
##       0_Disagree 0.6666667 0.6350711 0.6938326 0.7320261
##       1_Agree    0.3333333 0.3649289 0.3061674 0.2679739
```

There were more people who disagreed than agreed in each group. Proportion-wise they seem similar but the highest percentage that disagreed were in BMUFL group.

```
row_probs <- xtabs(~ bike_move_right2 + treatment, data = aa) %>%
  rowSums() %>%
  prop.table()
```

```
col_probs <- xtabs(~ bike_move_right2 + treatment, data = aa) %>%
  colSums() %>%
  prop.table()

expected <- row_probs %*% t(col_probs) * nrow(aa)
observed <- xtabs(~ bike_move_right2 + treatment, data = aa)
test_stat <- sum((expected - observed)^2/expected)
pchisq(test_stat, df = 3, lower.tail = FALSE)
```

```
## [1] 0.01536776
```

we reject the null hypothesis that agreement is independent of the type of signage observed (p = .01536776).

Problem 29 Imagine filling the table by first putting a number in the first column, then the second column, then the third column. Once the first two columns are filled, the third must be chosen to make the total 1107. This is why this table has only two degrees of freedom.

If the first number is 0, there are 1108 ways to fill in the second number. If the first number is 1, there are 1107 ways to fill the second number, and so on. Thus there are $1108 + 1107 + \cdots + 2 + 1 + 0$ ways to fill the table, or $\frac{1108 \cdot 1109}{2} = 614386$ ways.

11

Simple Linear Regression – Solutions

Problem 1

a.

```
lm(waiting ~ eruptions, data=faithful)
```

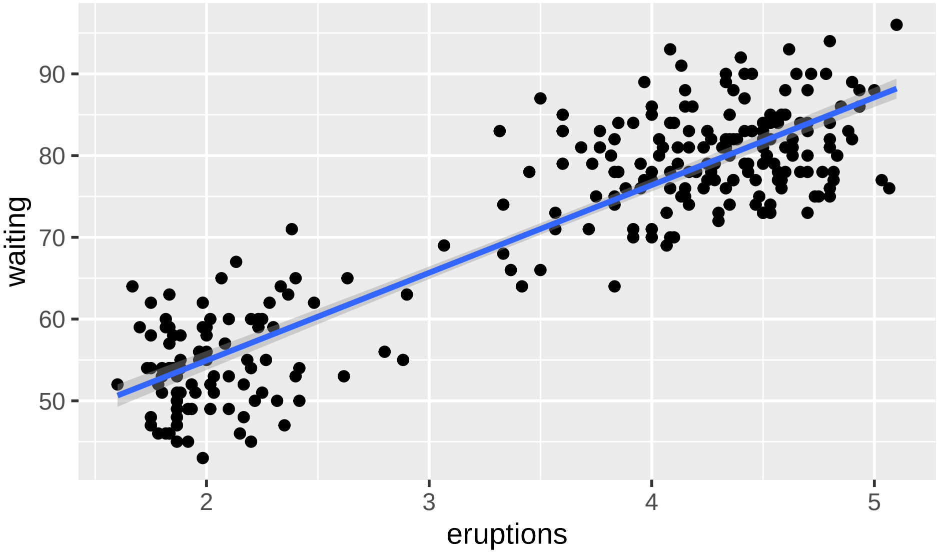
```
##
## Call:
## lm(formula = waiting ~ eruptions, data = faithful)
##
## Coefficients:
## (Intercept)    eruptions
##       33.47        10.73
```

The equation is $\widehat{\text{waiting}} = 33.47 + 10.73 \times \text{eruptions}$

b.

```
ggplot(faithful, aes(x=eruptions, y=waiting)) + geom_point() + geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



c.

```
faithful.mod <- lm(waiting ~ eruptions, data=faithful)
predict(faithful.mod, data.frame(eruptions = 4.3))
```

```
##        1
## 79.61186
```

Problem 2

```
barn_mod <- lm(barnacle_density ~ depth, data=fosdata::barnacles)
barn_mod
```

```
##
## Call:
## lm(formula = barnacle_density ~ depth, data = fosdata::barnacles)
##
## Coefficients:
## (Intercept)        depth
##      518.87       -10.53
```

```
predict(barn_mod, data.frame(depth = 30))
```

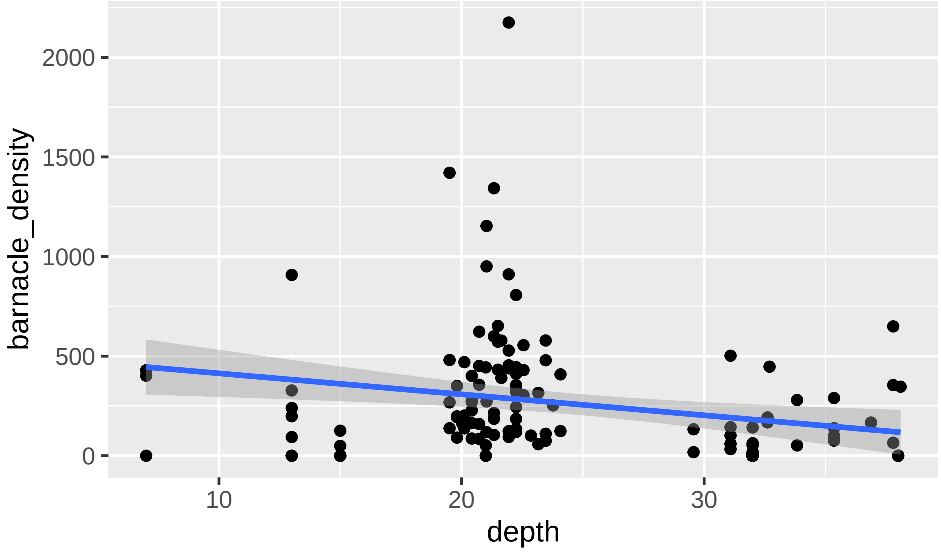
```
##        1
## 203.0197
```

a. The regression line is `barnacle_density` = 518.87 - 10.53 `depth`.
b. Expect a mean of 203 barnacles per square meter at 30 meters depth.

Note that this regression line does not follow the data closely.

```
ggplot(fosdata::barnacles, aes(x = depth, y = barnacle_density)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Problem 3

```
pub5young <- filter(ISwR::juul, tanner == 5 & age < 20)
igf.mod <- lm(igf1 ~ age, data=pub5young)
igf.mod
```

```
##
## Call:
## lm(formula = igf1 ~ age, data = pub5young)
##
```

```
## Coefficients:
## (Intercept)          age
##     1135.49       -38.94
```

```
predict(igf.mod, data.frame(age = 16))
```

```
##       1
## 512.517
```

The regression line is $\widehat{\text{igf1}} = 1135.49 - 38.94 \cdot \text{age}$. The slope of -38.94 means that for each additional year of age, the igf1 level drops by 38.94. We predict an igf1 level of 512.5 for the 16 year old.

Problem 4

```
gentoo <- palmerpenguins::penguins %>% filter(species == "Gentoo")
```

a.

```
gentoo %>% filter(!is.na(body_mass_g)) %>%
  mutate(badfit = -3037.2 + 34.6 * flipper_length_mm) %>%
  summarize(sum((body_mass_g-badfit)^2))
```

```
## # A tibble: 1 x 1
##   `sum((body_mass_g - badfit)^2)`
##                             <dbl>
## 1                       61818803.
```

b.

```
lm(body_mass_g ~ flipper_length_mm, data=gentoo)$coefficients
```

```
##       (Intercept) flipper_length_mm
##        -6787.2806           54.6225
```

The best fit line is `body_mass_g` = -6787.2806 + 54.6225 * `flipper_length_mm`.

c.

```
gentoo %>% filter(!is.na(body_mass_g)) %>%
  mutate(badfit = -6787.2806 + 54.6225 * flipper_length_mm) %>%
  summarize(sum((body_mass_g-badfit)^2))
```

```
## # A tibble: 1 x 1
##   `sum((body_mass_g - badfit)^2)`
##                             <dbl>
## 1                       15696203.
```

The sum of squared errors is much smaller for the best fit line.

Problem 5

a. Strongly negative.
b. Weak.
c. Strongly positive.
d. Strongly negative.

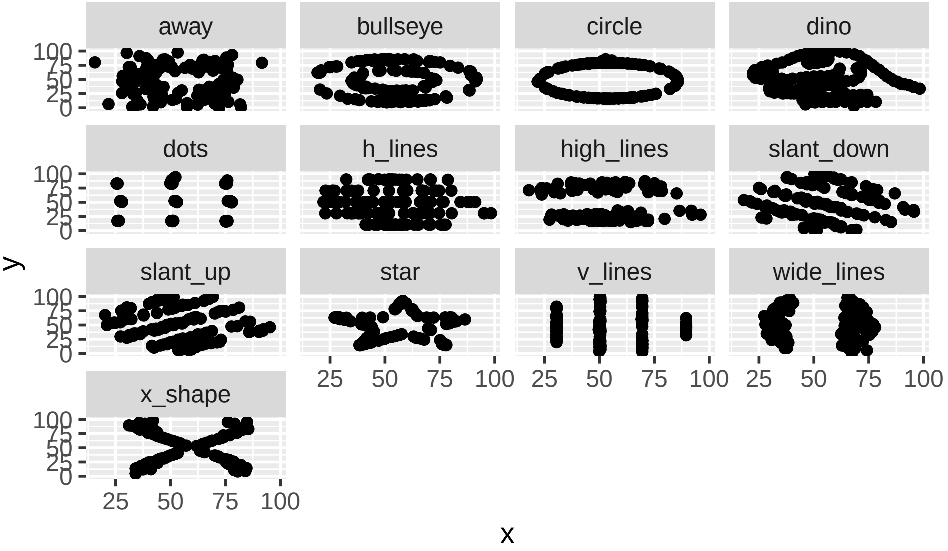
Problem 6

```
dd <- datasauRus::datasaurus_dozen
dd %>%
  group_by(dataset) %>%
  summarize(mux = mean(x),
            muy = mean(y),
            sdx = sd(x),
            sdy = sd(y),
            corxy = cor(x, y))
```

```
## # A tibble: 13 x 6
##    dataset      mux   muy   sdx   sdy   corxy
##    <chr>      <dbl> <dbl> <dbl> <dbl>   <dbl>
##  1 away        54.3  47.8  16.8  26.9 -0.0641
##  2 bullseye    54.3  47.8  16.8  26.9 -0.0686
##  3 circle      54.3  47.8  16.8  26.9 -0.0683
##  4 dino        54.3  47.8  16.8  26.9 -0.0645
##  5 dots        54.3  47.8  16.8  26.9 -0.0603
##  6 h_lines     54.3  47.8  16.8  26.9 -0.0617
##  7 high_lines  54.3  47.8  16.8  26.9 -0.0685
##  8 slant_down  54.3  47.8  16.8  26.9 -0.0690
##  9 slant_up    54.3  47.8  16.8  26.9 -0.0686
## 10 star        54.3  47.8  16.8  26.9 -0.0630
## 11 v_lines     54.3  47.8  16.8  26.9 -0.0694
## 12 wide_lines  54.3  47.8  16.8  26.9 -0.0666
## 13 x_shape     54.3  47.8  16.8  26.9 -0.0656
```

These summary statistics are all very similar, but the plots below are not.

```
ggplot(dd, aes(x = x, y = y)) +
  geom_point() +
  facet_wrap(vars(dataset))
```



Problem 7

Wealth. Availability of health care. Attitude towards self-care. Race. Urban/suburban/rural living. Empolyment situation. Eating habits. Smoking cigarettes. Amount of driving. Risk taking behavior.

Problem 8

a.

```
x <- rnorm(100000)
cor(x,x^2)
```

```
## [1] -0.001745225
```

b. No, X and X^2 are not independent.

Problem 9 The line goes through $(3, 2)$ and has slope $0.7\frac{2}{1}$ so the line is $\hat{y} = 1.4(x - 3) + 2$ or $\hat{y} = 1.4x - 2.2$.

Problem 10 The regression line is

$$\hat{y} = r\frac{s_y}{s_x}(x - \bar{x}) + \bar{y} = \bar{y} - r\frac{s_y}{s_x}\bar{x} + r\frac{s_y}{s_x}x$$

so

$$\hat{\beta}_0 = \bar{y} - r\frac{s_y}{s_x}\bar{x} = \frac{s_x\bar{y} - rs_y\bar{x}}{s_x}$$

Problem 11

The solution is $\beta_0 = \bar{y} - \bar{x}$.

Compute the SSE as

$$SSE = \sum \epsilon_i^2 = \sum (y_i - \beta_0 - x_i)^2$$

Now take the derivative with respect to β_0:

$$\frac{d}{d\beta_0}SSE = \sum 2(y_i - \beta_0 - x_i)(-1) = 2\left(-\sum y_i + \sum \beta_0 + \sum x_i\right)$$

Set equal to zero and divide by 2:

$$0 = -\sum y_i + \sum \beta_0 + \sum x_i = n\beta_0 + \sum x_i - \sum y_i$$

So:

$$\beta_0 = \frac{1}{n}\sum y_i - \frac{1}{n}\sum x_i = \bar{y} - \bar{x}$$

Notice that this value of β_0 causes the line $y = \beta_0 + x$ to pass through the point $(\bar{x}, \bar{y})$.

Problem 12

Compute the SSE as

$$SSE = \sum \epsilon_i^2 = \sum (y_i - \beta_1 x_i)^2$$

Now take the derivative with respect to β_1:

$$\frac{d}{d\beta_1}SSE = \sum 2(y_i - \beta_1 x_i)(-x_i) = 2\left(-\sum x_i y_i + \beta_1 \sum x_i^2\right)$$

Set equal to zero and solve for β_1 to get:

$$\beta_1 = \frac{\sum x_i y_i}{\sum x_i^2}$$

Problem 13

a. By symmetry of x and y and the previous solution, we get

$$\gamma = \frac{\sum x_i y_i}{\sum y_i^2},$$

and the line of best fit is $y = (1/\gamma)x$.

b. Almost any collection of points will work, for example $(0, 0), (0, 1), (1, 1)$.

```
x <- c(0,0,1)
y <- c(0,1,1)

sum(x * y)/sum(x^2)
```

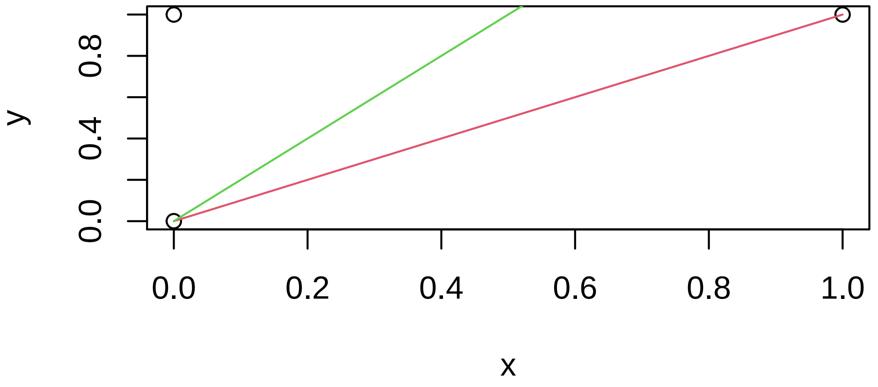
```
## [1] 1
```

```
sum(x * y)/sum(y^2)
```

```
## [1] 0.5
```

Lines of best fit are $y = x$ to minimize vertical error and $y = 2x$ to minimize horizontal error.

```
plot(x, y)
curve(1 * x, add = T, col = 2)
curve(2*x, add = T, col = 3)
```



Problem 14

1. Not good - non linear.
2. Not good - heteroscedastic.
3. Good.
4. Ok but one serious outlier.
5. Not good - not normal.
6. Looks good.
7. Looks good.
8. Looks good.

Problem 15

```
ans <- carData::Anscombe
```

a. Yes, a higher proportion of youth would seem to require a higher edcuation spend on a per-capita basis.

b.

```
cor(ans$young,ans$education)
```

```
## [1] 0.3114855
```

c.

```
lm(education ~ young, data=ans)
```

```
##
## Call:
## lm(formula = education ~ young, data = ans)
##
## Coefficients:
## (Intercept)        young
##    -20.4247       0.6039
```

Each additional youth (per 1000 people) requires about 60 cents per person (or $600 per 1000 people). The slope is significant (P = 0.0261)

d.The residuals look reasonable, except for Alaska (AK) which spends way more on education than other states, but also has a lot of youths.

```
ans49 <- filter(ans,education < 370)
lm(education ~ young, data=ans49)
```

```
##
## Call:
## lm(formula = education ~ young, data = ans49)
##
## Coefficients:
## (Intercept)        young
##    146.5564       0.1294
```

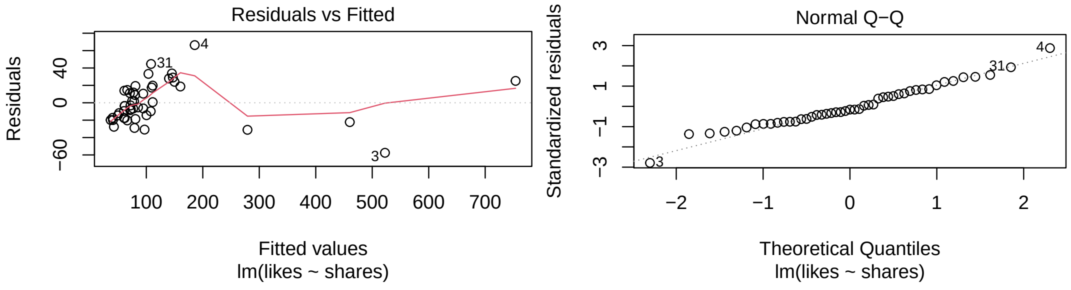
Removing Alaska, the slope of line is no longer significantly different from zero (P = 0.631). There appears to be no relationship between percentage of youth and educational expenditures.

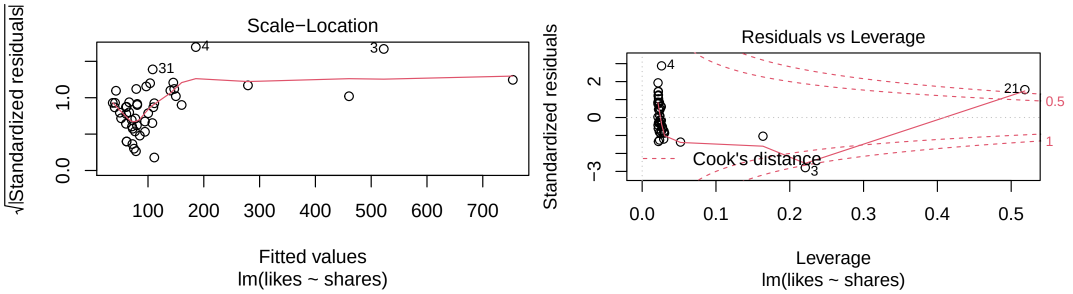
Problem 16

```
cern_twitter <- fosdata::cern %>% filter(platform == "Twitter")
```

a.

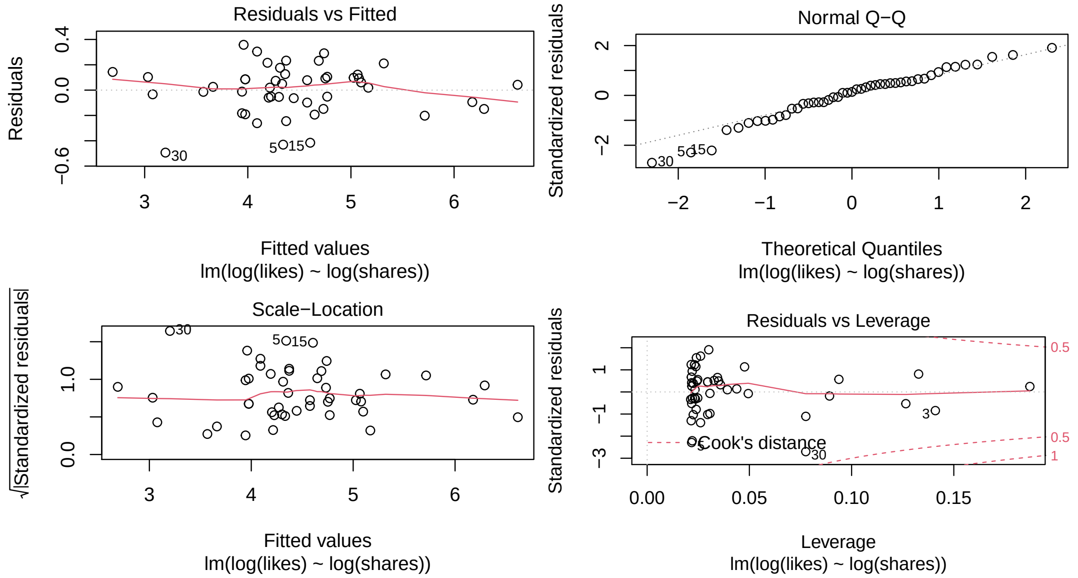
```
lm(likes ~ shares, data=cern_twitter) %>% plot()
```





b.

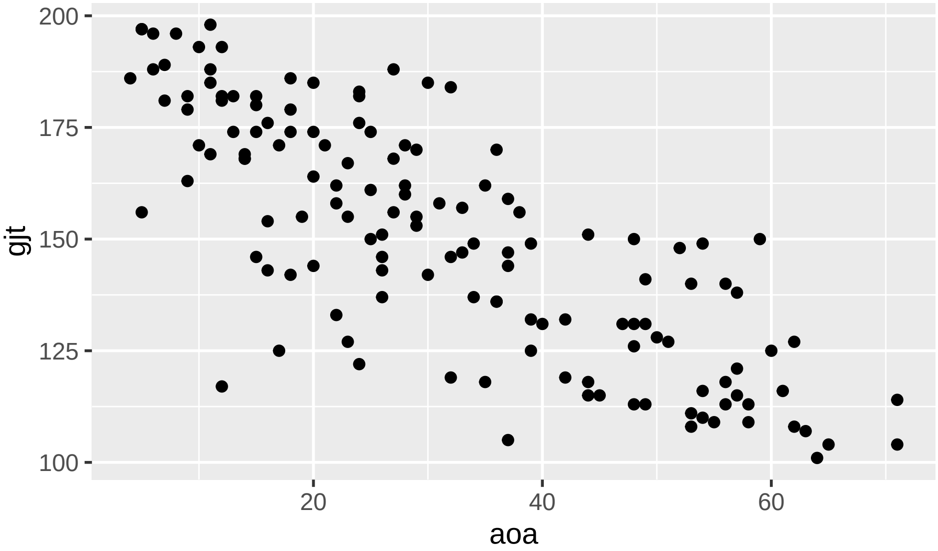
```
lm(log(likes) ~ log(shares), data=cern_twitter) %>% plot()
```



c. The log model is better. In the first model, the residuals vs fitted plot shows that there is still a linear trend among the samller values, and there are high leverage outliers that probably caused it. The residual plots for the log model look very good.

Problem 17

```
dd <- fosdata::crit_period
ggplot(dd, aes(x = aoa, y = gjt)) +
  geom_point()
```

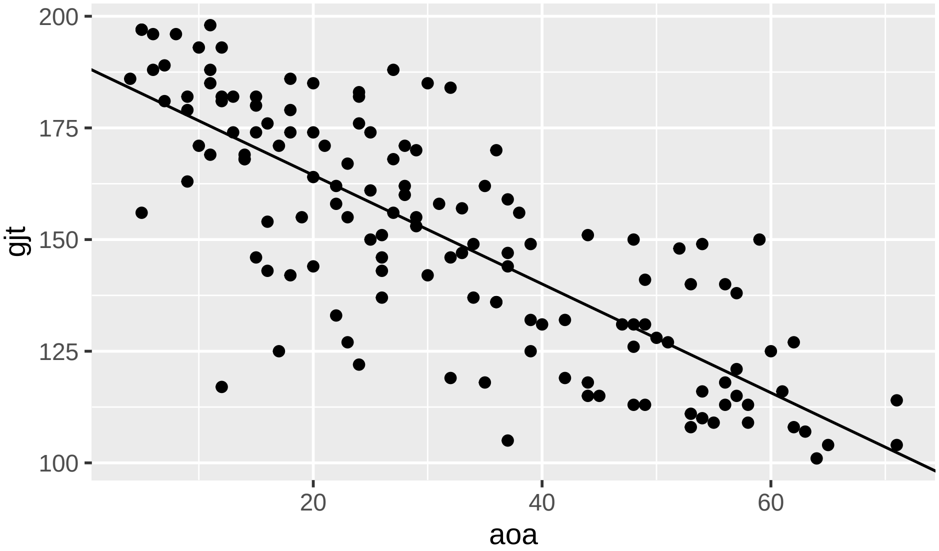


b.

```
mod <- lm(gjt ~ aoa, data = dd)
mod$coefficients
```

```
## (Intercept)         aoa
##  188.774823   -1.217982
```

```
ggplot(dd, aes(x = aoa, y = gjt)) +
  geom_point() +
  geom_abline(slope = mod$coefficients[2], intercept = mod$coefficients[1])
```

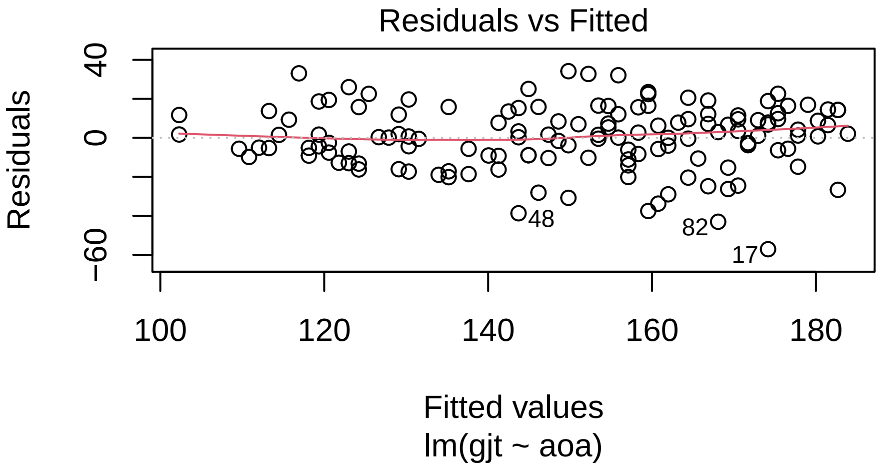


c.

The slope is the mean change in grammaticality judgment test scores for each year later that the person started learning the language.

d.

```
plot(mod, which = 1)
```



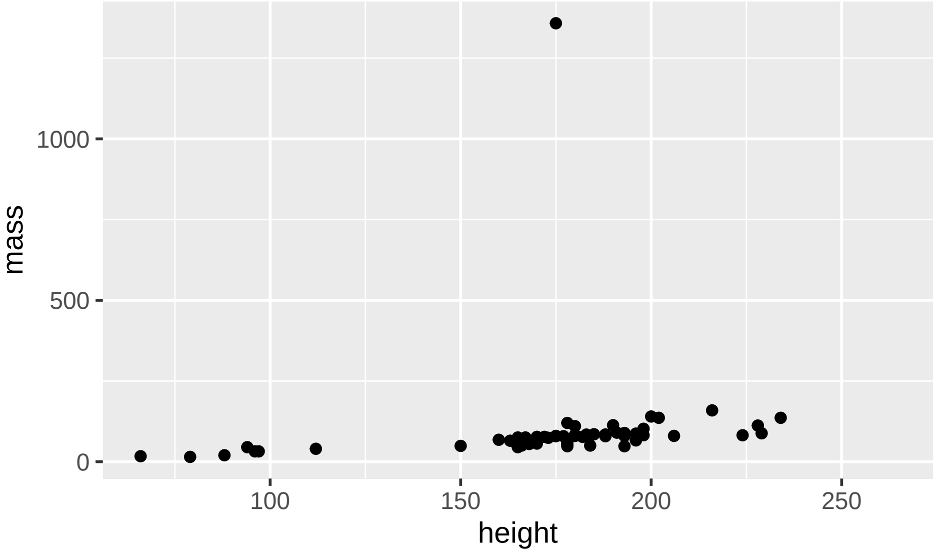
There is not a pronounced elbow, but if you squint then you might see something. There does not seem to be a strong elbow in the residuals.

Problem 18

a.

```
ggplot(starwars, aes(x = height, y = mass)) +
  geom_point()
```

```
## Warning: Removed 28 rows containing missing values (geom_point).
```



b.

```
mod_sw <- lm(mass ~ height, data = starwars)
summary(mod_sw)
```

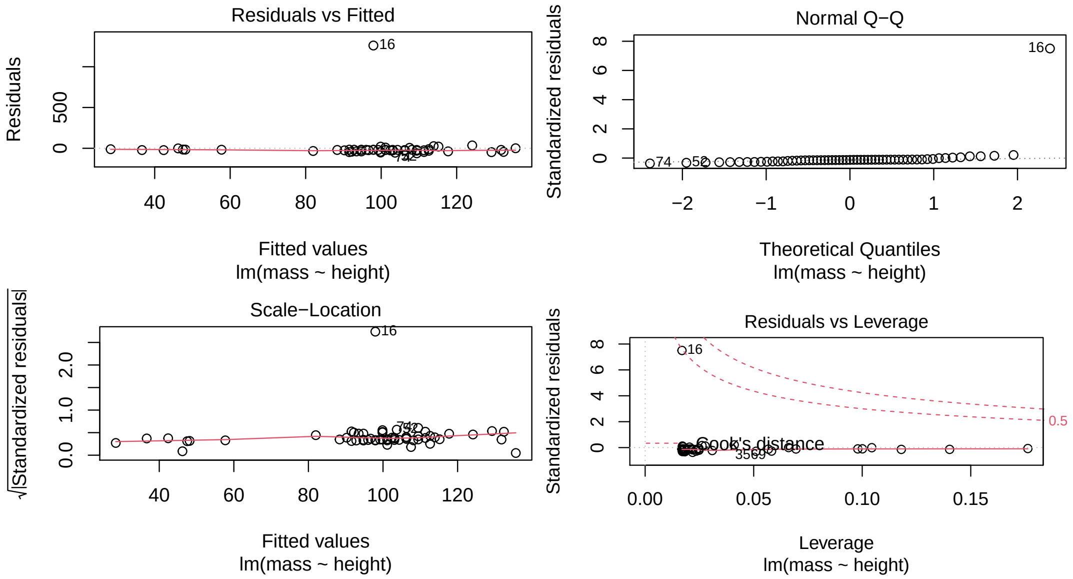
```
##
## Call:
## lm(formula = mass ~ height, data = starwars)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -61.43  -30.03  -21.13  -17.73 1260.06
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.8103   111.1545  -0.124    0.902
## height        0.6386     0.6261   1.020    0.312
##
## Residual standard error: 169.4 on 57 degrees of freedom
##   (28 observations deleted due to missingness)
## Multiple R-squared:  0.01792,    Adjusted R-squared:  0.0006956
## F-statistic:  1.04 on 1 and 57 DF,  p-value: 0.312
```

Equation of line of best fit is `mass` = -13.8 + 0.6386 `heght`.

c.

```
plot(mod_sw)
```



The residuals are dominated by the outlier, which is observation number 16.

```
starwars[16, "name"]
```

```
## # A tibble: 1 x 1
##   name
##   <chr>
## 1 Jabba Desilijic Tiure
```

Better known as “Jabba the Hutt”

d.

```
mod_sw_no_jabba <- lm(mass ~ height, data = starwars[-16,])
coefficients(mod_sw_no_jabba)
```

```
## (Intercept)      height
## -32.5407582   0.6213599
```

The intercept and slope have changed quite a bit, though they are easily within two standard errors of the original estimates with Jabba. The biggest difference in this model is that the standard errors are much smaller, as are the *p*-values.

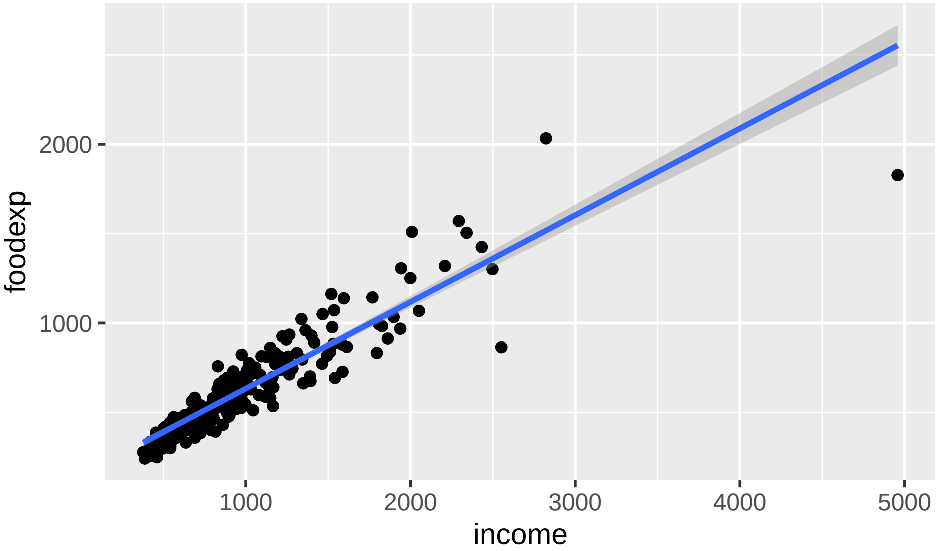
```
summary(mod_sw_no_jabba)
```

```
##
## Call:
## lm(formula = mass ~ height, data = starwars[-16, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.382  -8.212   0.211   3.846  57.327
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -32.54076   12.56053  -2.591   0.0122 *
## height        0.62136    0.07073   8.785 4.02e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.14 on 56 degrees of freedom
##   (28 observations deleted due to missingness)
## Multiple R-squared:  0.5795, Adjusted R-squared:  0.572
## F-statistic: 77.18 on 1 and 56 DF,  p-value: 4.018e-12
```

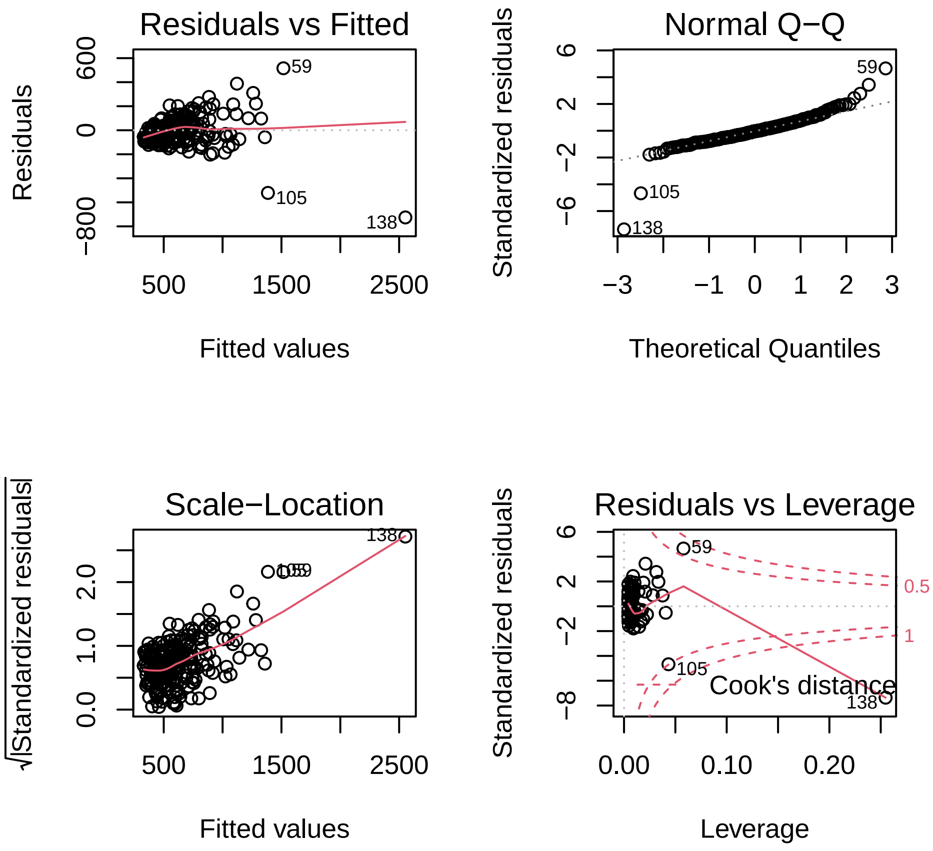
Problem 19

```
suppressMessages(library(quantreg))
data(engel)
engel %>% ggplot(aes(x=income, y=foodexp)) + geom_point() + geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
par(mfrow=c(2,2))  # get all four residual plots in one image
plot(lm(foodexp ~ income, data=engel))
```



The residuals are heteroskedastic, which you can see on the Residuals vs. Fitted plot and on the Scale-Location plot. The Normal Q-Q plot shows that the residuals are not normally distributed. The leverage plot indicates that point 138 is a high-leverage outlier, and by looking at our plot of the data with the line you can tell that the outlier has pulled the line downwards.

This data is not appropriate for a simple linear regression. Removing the outlier and taking the log of both variables improves the situation.

Problem 20

```
pub5young <- filter(ISwR::juul, tanner == 5 & age < 20)
igf.mod <- lm(igf1 ~ age, data=pub5young)
predict(igf.mod, data.frame(age = 16), interval="predict")
```

```
##       fit      lwr     upr
## 1 512.517 302.6871 722.347
```

The prediction interval is $[302.6871, 722.347]$.

Problem 21

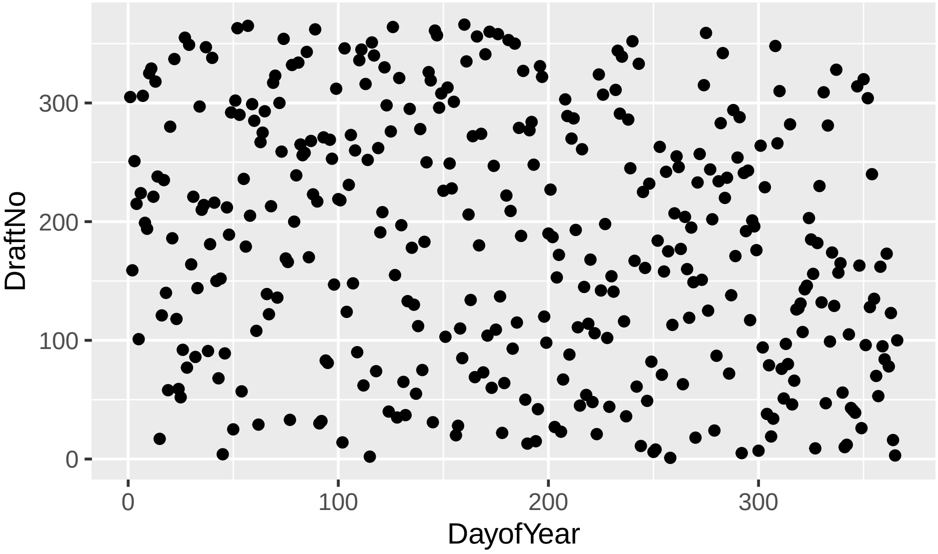
```
cern_twitter <- fosdata::cern %>% filter(platform == "Twitter")
lm(log(likes) ~ log(shares), data=cern_twitter) %>% confint()
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.3064638 0.8769978
## log(shares) 0.7832336 0.9054010
```

The 95% confidence intervals are $[0.78, 0.91]$ for the slope and $[0.31, 0.88]$ for the intercept.

Problem 22 a.

```
fosdata::draft %>% ggplot(aes(x=DayofYear, y=DraftNo)) + geom_point()
```



It's hard to see much of a pattern in the plot.

b.

```
summary(lm(DraftNo ~ DayofYear, data=fosdata::draft))$coefficients
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 225.0092222 10.8119658 20.811130 6.271511e-64
## DayofYear    -0.2260594  0.0510617 -4.427181 1.263829e-05
```

There is a significant relationship between day of year and draft number ($p = 1.264 \times 10^{-5}$). It is generally accepted that the capsules were not well mixed and that led to an uneven distribution of draft numbers over the days of the year, causing days later in the year to have earlier draft numbers.

Problem 23

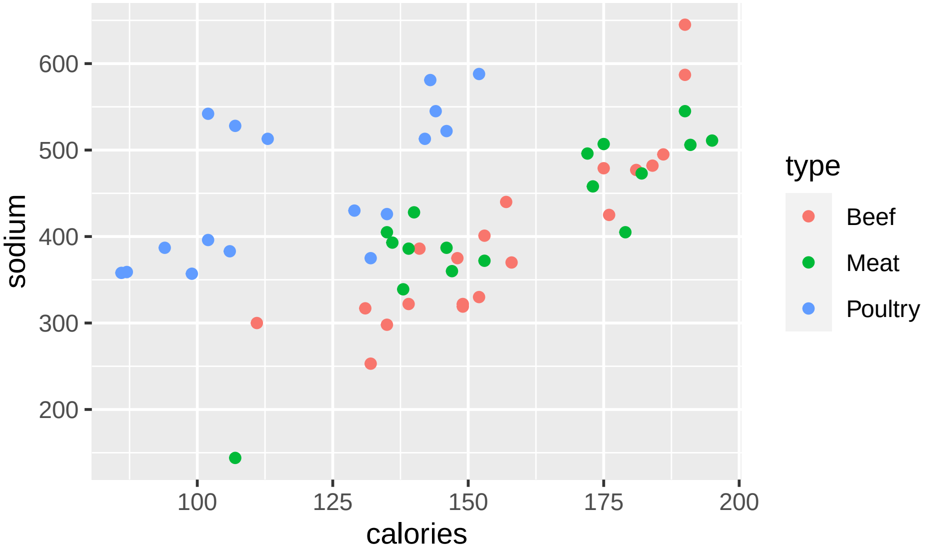
```
gentoo <- palmerpenguins::penguins %>% filter(species == "Gentoo")
summary(lm(body_mass_g ~ flipper_length_mm, data=gentoo))$coefficients
```

```
##                    Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)       -6787.2806 1092.551940 -6.212318 7.649742e-09
## flipper_length_mm    54.6225    5.028244 10.863137 1.330279e-19
```

There is a significant correlation between flipper length and body mass ($P = 1.33 \times 10^{-19}$).

Problem 24 a.

```
hot_dogs <- fosdata::hot_dogs
ggplot(hot_dogs,aes(x=calories,y=sodium,color=type)) + geom_point()
```



b. We remove the Poultry hot dogs because they clearly have more sodium per calorie than the Beef/Meat hot dogs.

```
bmdogs <- filter(hot_dogs,type != "Poultry")
dog.mod <- lm(sodium ~ calories, data=bmdogs)
dog.mod
```

```
##
## Call:
## lm(formula = sodium ~ calories, data = bmdogs)
##
## Coefficients:
## (Intercept)     calories
##    -160.580        3.613
```

The regression line is $\widehat{\text{sodium}} = -160.580 + 3.613 \times \text{calories}$.

c.

```
predict(dog.mod, data.frame(calories = 140), interval = "predict")
```

```
##        fit      lwr      upr
## 1 345.1826 244.4656 445.8995
```

The model predicts a sodium level of 345.2 for a 140 calorie hot dog. It predicts that 95% of beef/meat hot dogs with 140 calories would have sodium in the interval $[244.5, 445.9]$.

Proboem 25

There are significant relationships between log of range and log of acorn size in both regions (Atlantic $p = 0.000386$, California $p = 0.0354$).

```
acorns <- fosdata::acorns %>%
  filter(Range > 13) %>%
  mutate(logrange = log(Range), logsize = log(Acorn_size))
acorns %>% filter(Region == "Atlantic") %>% lm(logrange ~ logsize, data=.) %>% summary()
```

```
##
## Call:
```

```
## lm(formula = logrange ~ logsize, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0466 -0.4789  0.2227  0.6507  1.4194
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.4175     0.2028  41.513  < 2e-16 ***
## logsize       0.7640     0.1876   4.073 0.000386 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8883 on 26 degrees of freedom
## Multiple R-squared:  0.3896, Adjusted R-squared:  0.3661
## F-statistic: 16.59 on 1 and 26 DF,  p-value: 0.000386
```

```
acorns %>% filter(Region == "California") %>% lm(logrange ~ logsize, data=.) %>% summary()
```

```
##
## Call:
## lm(formula = logrange ~ logsize, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67789 -0.26166  0.01577  0.32085  0.54167
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.0080     0.1909  31.464 1.13e-09 ***
## logsize       0.3271     0.1294   2.528   0.0354 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4148 on 8 degrees of freedom
## Multiple R-squared:  0.444,  Adjusted R-squared:  0.3745
## F-statistic: 6.389 on 1 and 8 DF,  p-value: 0.03538
```

Problem 26

a.

```
library(Lahman)
scoring <- Batting %>% group_by(yearID) %>%
  summarize(HR = sum(HR), R = sum(R))
```

b.

```
ggplot(data=scoring, aes(x=HR,y=R,color=yearID)) + geom_point()
```

c.

```
scoring <- filter(scoring, yearID >= 1970)
```

```
ggplot(data=scoring, aes(x=HR,y=R)) + geom_point()
```

d.

```
scoring.mod <- lm(R ~ HR, data=scoring)
summary(scoring.mod)
```

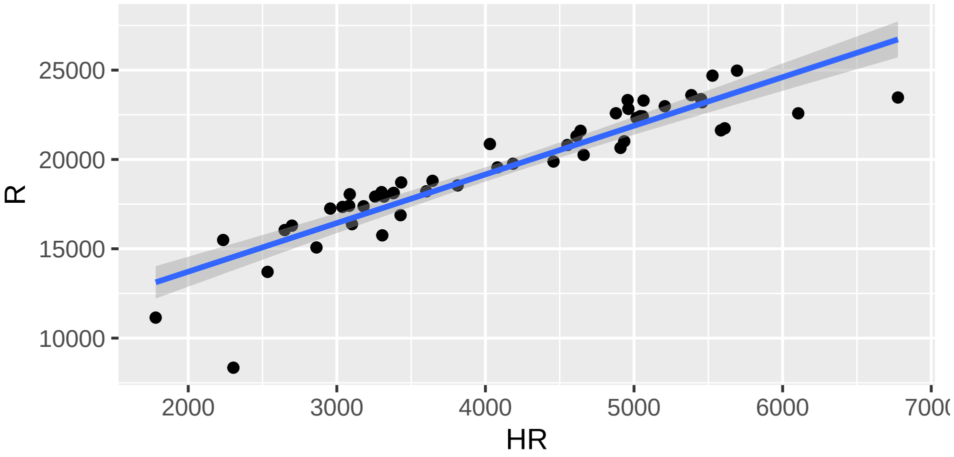
```
##              Estimate  Std. Error  t value     Pr(>|t|)
## (Intercept) 8272.4546 740.8144732 11.16670 4.553117e-15
## HR             2.7224   0.1732006 15.71819 8.978120e-21
```

The model predicts 2.84 additional R for each additional HR. There is very significant evidence ($P < 10^{-25}$) that the slope, which is about 2.84, is not 1.

e.

```
ggplot(data=scoring, aes(x=HR,y=R)) + geom_point() + geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



f.

```
predict(scoring.mod, data.frame(HR = 4000),interval="prediction")
```

```
##        fit      lwr      upr
## 1 19162.06 16295.18 22028.94
```

We predict between 17822 and 21174 runs scored when 4000 home runs are hit.

Problem 27

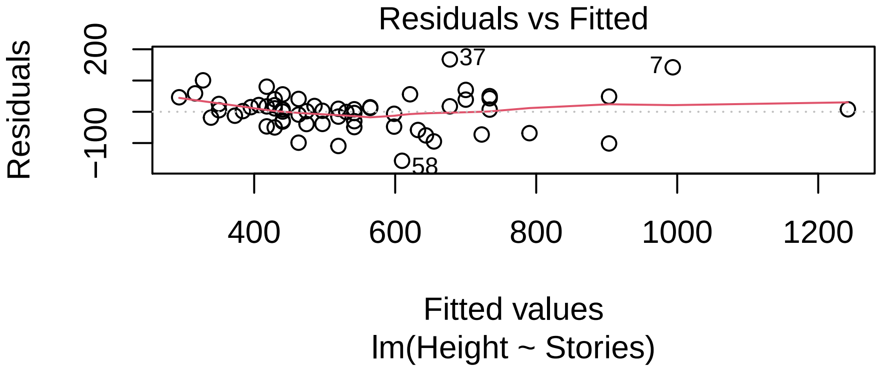
```
tall <- Sleuth2::ex0728
```

a.

```
ggplot(data=tall, aes(x=Stories, y=Height)) + geom_point()
```

b. Plotting the residuals, we find buildings 7 and 37 are outliers - they are tall for the number of stories.

```
plot(lm(Height ~ Stories, data=tall),which=1)
```



```
tall[c(7,37),]
```

```
##    Year Height Stories
## 7  1974   1136      80
## 37 1989    845      52
```

c.

```
tall$HeightPerStory <- tall$Height/tall$Stories
summary(lm(HeightPerStory ~ Year, data=tall))
```

```
##                Estimate  Std. Error   t value   Pr(>|t|)
## (Intercept) 43.88695806 22.47785722  1.952453 0.05571661
## Year        -0.01521083  0.01139974 -1.334314 0.18731624
```

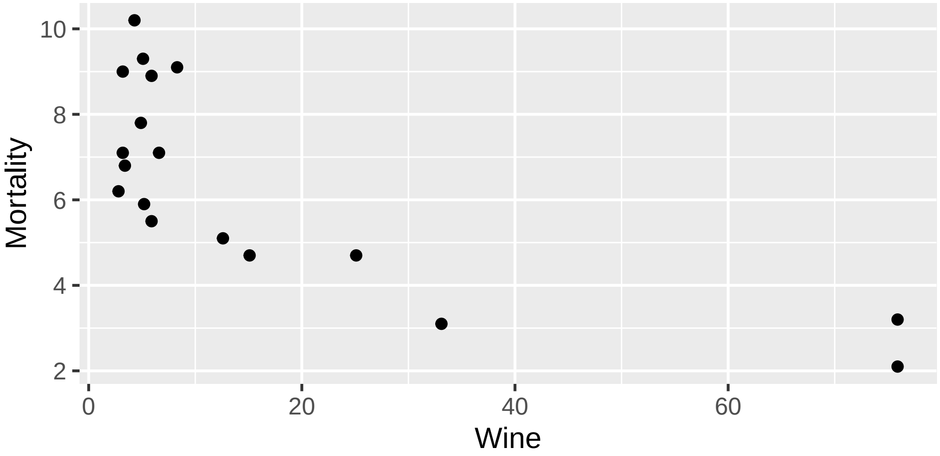
There is not significant evidence that height per story has changed over time. The slope of the regression line is -0.015, which is not significantly different from zero (P = 0.1873).

Problem 28

```
wine <- Sleuth3::ex0823
```

a.

```
ggplot(data=wine, aes(x=Wine, y=Mortality)) + geom_point()
```



The data does not look linear. A transformation is needed. Using log(Wine) improves the situation dramatically.

b.

```
summary(lm(Mortality ~ log(Wine), data=wine))
```

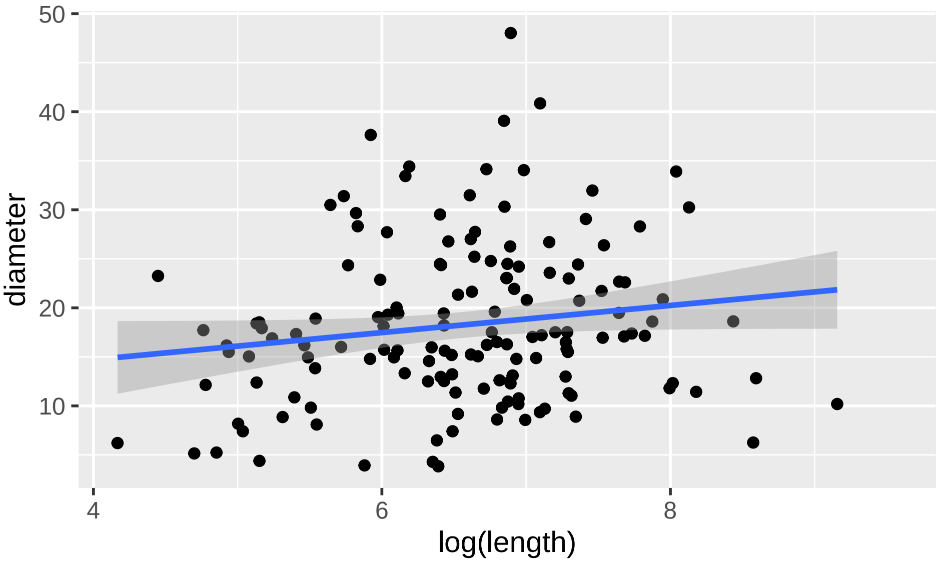
```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 10.279524  0.8316433 12.360497 1.338392e-09
## log(Wine)   -1.771155  0.3467517 -5.107847 1.053553e-04
```

There is significant evidence ($p = 0.0001$) that Mortality decreases with increased log(Wine).

c. No. Data points represent entire countries and may say nothing about individuals. Additionally, there are many other confounding factors (e.g. the wealth of the country) that make it hard to claim causality between wine consumption and mortality.

Problem 29

```
plastics <- fosdata::plastics
ggplot(data=plastics, aes(x=log(length), y=diameter)) +
  geom_point() + geom_smooth(method="lm")
```



c.

```
summary(lm(diameter ~ log(length), data=plastics))
```

```
##             Estimate Std. Error  t value   Pr(>|t|)
## (Intercept) 9.187940  4.8342757 1.900582 0.05931433
## log(length) 1.381558  0.7292569 1.894473 0.06012619
```

There is not significant evidence of a relationship between diameter and the log of length, $p = 0.06$.

Problem 30

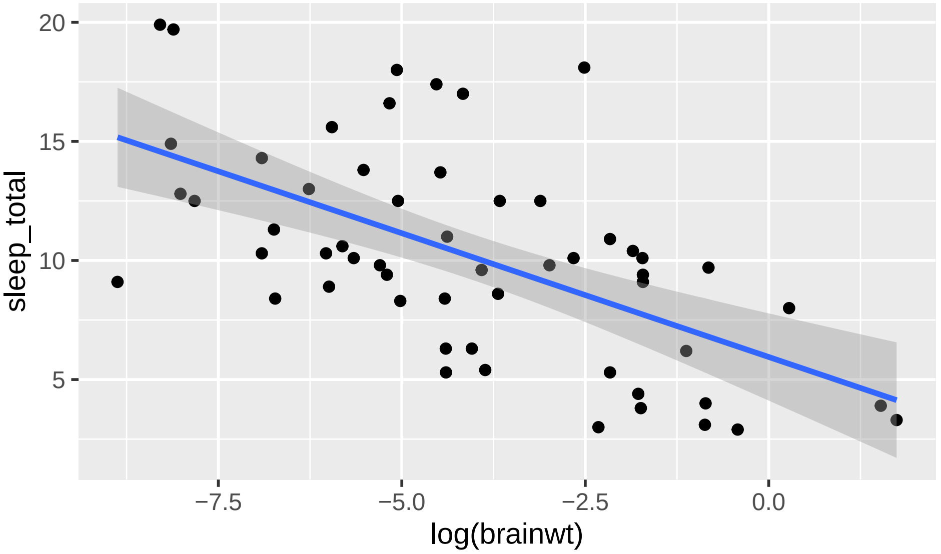
a.

```
msleep %>% ggplot(aes(y=sleep_total,x=log(brainwt)))+geom_point() + geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 27 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 27 rows containing missing values (geom_point).
```



b. The correlation is -0.594 and the relationship is significant ($p = 1.36 \times 10^{-6}$).

```
cor(msleep$sleep_total, log(msleep$brainwt), use="complete.obs")
```

```
## [1] -0.5944555
```

c. The plots look great - no concerns at all.

Problem 31 As expected, the true mean at $x = x_0$ lies in the 95% confidence interval 95% of the time.

```
simdata <- replicate(1000,{
  x <- runif(30,0,10)
  y <- 1 + 2*x + rnorm(30)
  ci <- predict(lm(y~x), data.frame(x = x[1]), interval = "confidence")
  (ci[2] < 1 + 2*x[1]) & (1 + 2*x[1] < ci[3])
})
mean(simdata)
```

```
## [1] 0.947
```

Problem 32

As expected, the new data lies in the prediction interval 95% of the time.

```
simdata <- replicate(1000,{
  x <- runif(30,0,10)
  y <- 1 + 2*x + rnorm(30)
  x_star <- runif(1,0,10)
  y_star <- 1 + 2*x_star + rnorm(1)
  pi <- predict(lm(y~x), data.frame(x = x_star), interval = "prediction")
  (pi[2] < y_star) & (y_star < pi[3])
})
mean(simdata)
```

```
## [1] 0.949
```

Problem 33 The success rate varies, in these examples from a low of 0.891 to a high of 0.980. This is because the quality of the regression line depends on the data, sometimes coming closer to the true line, other times further away. Once the regression line is fixed, the probability of new data landing in the prediction intervals is related to how lucky we were with the line.

```
replicate(10, {  # 10 is "a few" times
  x <- runif(30,0,10)
  y <- 1 + 2*x + rnorm(30)
  simdata <- replicate(1000,{
    x_star <- runif(1,0,10)
    y_star <- 1 + 2*x_star + rnorm(1)
    pi <- predict(lm(y~x), data.frame(x = x_star), interval = "prediction")
    (pi[2] < y_star) & (y_star < pi[3])
  })
  mean(simdata)
})
```

```
##  [1] 0.891 0.972 0.912 0.914 0.978 0.958 0.980 0.969 0.958 0.976
```

Problem 34 a. About 46-49% of the time.

```
simdata <- replicate(1000,{
  x <- runif(30,0,10)
  y <- x^2 + rnorm(30)
  x_star <- 0
  y_star <- x_star^2 + rnorm(1)
  pi <- predict(lm(y~x), data.frame(x = x_star), interval = "prediction")
  (pi[2] < y_star) & (y_star < pi[3])
})
mean(simdata)
```

```
## [1] 0.48
```

b. Essentially 100% of the time.

```
simdata <- replicate(1000,{
  x <- runif(30,0,10)
  y <- x^2 + rnorm(30)
  x_star <- 5
  y_star <- x_star^2 + rnorm(1)
  pi <- predict(lm(y~x), data.frame(x = x_star), interval = "prediction")
  (pi[2] < y_star) & (y_star < pi[3])
})
mean(simdata)
```

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

Problem 35 About 2.6-2.7 when $x = 10$ and about 2.8-2.9 when $x = 1$.

```
x <- 1:19
res10 <- replicate(1000,{
  y <- 1 + 2*x + rnorm(19,0,3)
  lm(y ~ x)$residuals[x==10]
})
res1 <- replicate(1000,{
  y <- 1 + 2*x + rnorm(19,0,3)
  lm(y ~ x)$residuals[x==1]
})
sd(res1)
```

```
## [1] 2.758172
```

```
sd(res10)
```

```
## [1] 2.828756
```

Problem 36

```
chinstrap <- filter(palmerpenguins::penguins, species == "Chinstrap")
loo_zero <- function(test_obs) {
  test <- chinstrap[test_obs,]
  train <- chinstrap[-test_obs,]
  mod0 <- lm(body_mass_g ~ flipper_length_mm + 0, data=train)
  bm0 <- predict(mod0, newdata=test)
  bm0 - test$body_mass_g
}
err_zero <- sapply(1:68, loo_zero)
mean(err_zero^2)
```

```
## [1] 100660.5
```

With a MSE of 100660.5, this model has a higher MSE than the linear model with intercept, which has an MSE of 91245.43.

Problem 37

```
loo_spline <- function(test_obs, degree) {
  test <- chinstrap[test_obs,]
  train <- chinstrap[-test_obs,]
  mod2 <- smooth.spline(x = train$flipper_length_mm,
                        y = train$body_mass_g,
                        df = degree)
  bm2 <- predict(mod2, test$flipper_length_mm)$y
  bm2 - test$body_mass_g
}
err_spline18 <- sapply(1:68, loo_spline, degree=18)
mean(err_spline18^2)
```

```
## [1] 93175.27
```

```
err_spline10 <- sapply(1:68, loo_spline, degree=10)
mean(err_spline10^2)
```

```
## [1] 95496.97
```

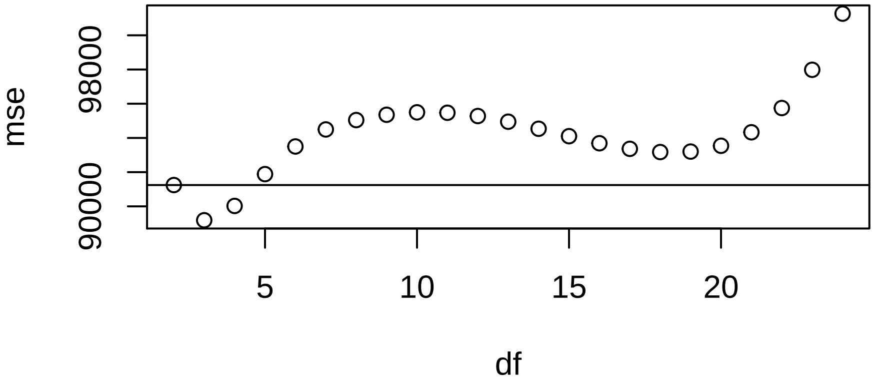
```
err_spline3 <- sapply(1:68, loo_spline, degree=3)
mean(err_spline3^2)
```

```
## [1] 89187.98
```

The model with 3df does better than the linear model, which has an MSE of 91245.43.

Problem 38

```
mse_spline <- function(degree) {
  err_spline <- sapply(1:68, loo_spline, degree=degree)
  mean(err_spline^2)
}
df <- 2:24
mse <- sapply(df,mse_spline)
plot(df,mse)
abline(h=91245.43)
```



The best model is the spline with 3 df, which has a MSE of 89188. With 2df, the spline model is a line and it matches the least squares regression line.

Problem 39

```
child_tasks <- fosdata::child_tasks
loo_linear <- function(test_obs) {
  test <- child_tasks[test_obs,]
  train <- child_tasks[-test_obs,]
  mod_linear <- lm(stt_cv_trail_b_secs ~ age_in_months, data=train)
  stt <- predict(mod_linear, newdata=test)
  stt - test$stt_cv_trail_b_secs
}
err_linear <- sapply(1:nrow(child_tasks), loo_linear)
mean(err_linear^2)
```

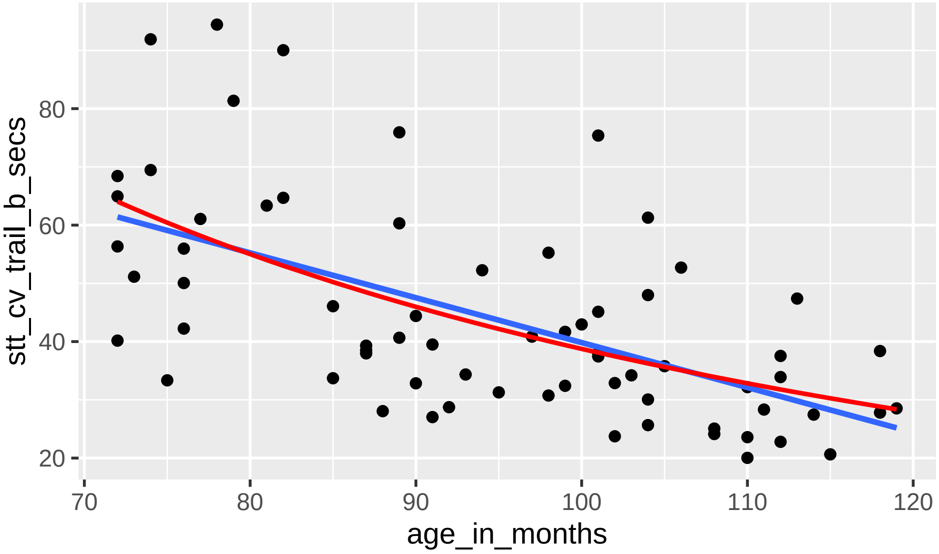
```
## [1] 218.7925
```

```
loo_inverse <- function(test_obs) {
  test <- child_tasks[test_obs,]
  train <- child_tasks[-test_obs,]
  mod_inverse <- lm(stt_cv_trail_b_secs ~ I(1/age_in_months), data=train)
  stt <- predict(mod_inverse, newdata=test)
  stt - test$stt_cv_trail_b_secs
}
err_inverse <- sapply(1:nrow(child_tasks), loo_inverse)
mean(err_inverse^2)
```

```
## [1] 217.9428
```

a. The MSE for the linear model is 218.8.
b. The MSE for the inverse model is 217.9.
c. The inverse model is a better predictor, according to LOOCV.
d.

```r
mod_inverse <- lm(stt_cv_trail_b_secs ~ I(1/age_in_months), data=child_tasks)
child_tasks$fit_inverse <- predict(mod_inverse, child_tasks)
child_tasks %>%
  ggplot(aes(x=age_in_months, y=stt_cv_trail_b_secs)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE) +
  geom_line(aes(y=fit_inverse), color="red", size=0.8)
```



12

Analysis of Variance and Comparison of Multiple Groups – Solutions

Problem 1

```
chimps <- fosdata::chimps
chimps %>%
  count(population)
```

```
##   population  n
## 1      NGOGO 75
## 2       NIRC 43
## 3        TAI 47
```

$n_1 = 43, n_2 = 75$ and $n_3 = 47$.

```
chimps %>%
  group_by(population) %>%
  summarize(mu = mean(grey_score_avg))
```

```
## # A tibble: 3 x 2
##   population    mu
##   <chr>      <dbl>
## 1 NGOGO       2.43
## 2 NIRC        2.87
## 3 TAI         2.37
```

$x_{1\cdot} = 2.87, x_{2\cdot} = 2.43$ and $x_{3\cdot} = 2.37$

Problem 2

The F test statistic is $F_{2,87}$.

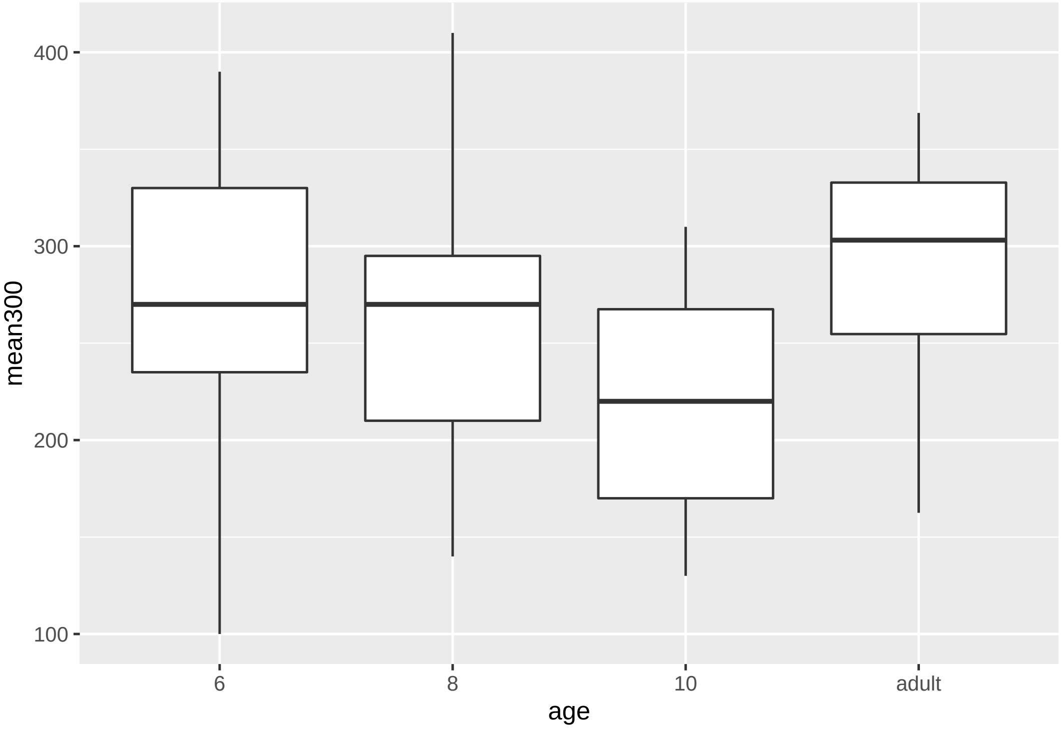
Problem 3

```
x <- rep(letters[1:4], each = 25)
sim_data <- replicate(10000, {
  y <- rnorm(100, 1, 2)
  anova(lm(y ~ x))[1,4]
})

hist(sim_data, probability = T)
curve(df(x, 3, 96), add = T, col = 2)
```

The histogram of simulated F statistics is consistent with the $F_{3,96}$ distribution.

Problem 4

```
ww <- fosdata::weight_estimate
ggplot(ww, aes(x = age, y = mean300)) +
  geom_boxplot() #looks good
```



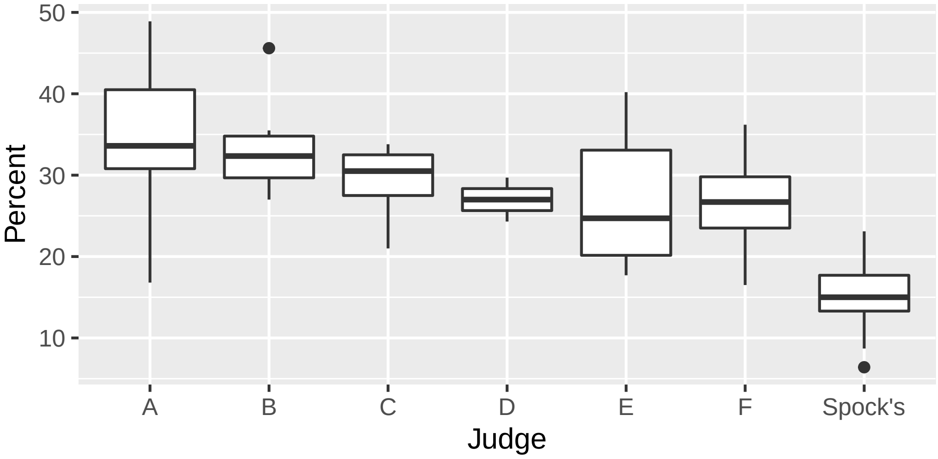
```
anova(lm(mean300 ~ age, data = ww))
```

```
## Analysis of Variance Table
##
## Response: mean300
##           Df Sum Sq Mean Sq F value  Pr(>F)
## age        3  60786 20262.1  4.9018 0.00362 **
## Residuals 76 314155  4133.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reject H_0 and conclude at least one mean is different

Problem 5

```
spock <- Sleuth3::case0502
ggplot(spock, aes(x=Judge, y=Percent)) + geom_boxplot()
```



a. Spock's venire pretty clearly has a lower percent female than the other judges do.

b.

```
spock$isSpock <- spock$Judge == "Spock's"
judges.mod <- lm(Percent ~ Judge, data = filter(spock, !isSpock))
anova(judges.mod)
```

```
## Analysis of Variance Table
##
## Response: Percent
##           Df  Sum Sq Mean Sq F value Pr(>F)
## Judge      5  326.46  65.292  1.2183 0.3239
## Residuals 31 1661.33  53.591
```

Since $P = 0.3239$, there is no significant evidence that the percent of women varies among the venires of judges A-F.

c.

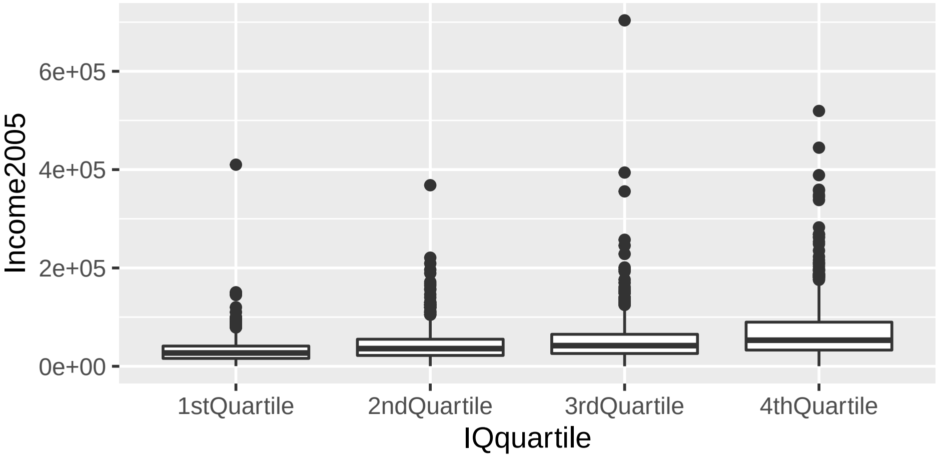
```
t.test(Percent ~ isSpock, data=spock)
```

```
##
##  Welch Two Sample t-test
##
## data:  Percent by isSpock
## t = 7.1597, df = 17.608, p-value = 1.303e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  10.49935 19.23999
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            29.49189            14.62222
```

There is significant evidence ($P = 1.3 \times 10^{-6}$) that Spock's judge has a different percent of women on his juries than the other judges do.

Problem 6

```
iq <- Sleuth3::ex0524
ggplot(data=iq, aes(x=IQquartile, y=Income2005))+geom_boxplot()
```



```
anova(lm(Income2005 ~ IQquartile, data=iq))
```

```
## Analysis of Variance Table
##
## Response: Income2005
##               Df     Sum Sq    Mean Sq F value    Pr(>F)
## IQquartile     3 4.9337e+11 1.6446e+11  82.442 < 2.2e-16 ***
## Residuals   2580 5.1466e+12 1.9948e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is significant evidence ($P < 2 \times 10^{-16}$) that Income2005 varies among IQ quartiles. However, the residuals show the data is skew and also heteroskedastic. The assumptions for ANOVA were not met, which casts doubt on the results. A transformation is called for. Taking the logarithm of Income2005 improves the situation, and gives the same P-value.

Problem 7

```
anova(lm(weight ~ feed, data=chickwts))
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value    Pr(>F)
## feed       5 231129   46226  15.365 5.936e-10 ***
## Residuals 65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is a significant difference in mean weight of chicks between different feed types ($P = 6 \times 10^{-10}$). The assumptions for ANOVA appear to be satisfied.

Problem 8

```
bar.mod <- lm((Y1+Y2) ~ Var,data=MASS::immer)
anova(bar.mod)
```

```
## Analysis of Variance Table
##
## Response: (Y1 + Y2)
##           Df Sum Sq Mean Sq F value Pr(>F)
## Var         4  10620  2655.0  1.2937 0.2993
## Residuals 25  51308  2052.3
```

There is not a significant difference in yield among the varieties. Checking the residuals, one area of concern is that the T variety has both the highest yield and highest variance.

Problem 9 a. Sprays C,D, and E are more effective. A, B, and F are less effective.

b. We reject H_0, the three effective sprays do not have the same mean (P = 0.0088)

```
effective <- filter(InsectSprays, spray %in% c('C','D','E'))
anova(lm(count ~ spray, data=effective))
```

```
## Analysis of Variance Table
##
## Response: count
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## spray      2  48.167 24.0833  5.4873 0.008763 **
## Residuals 33 144.833  4.3889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c. The three ineffective sprays do not significantly differ in mean.

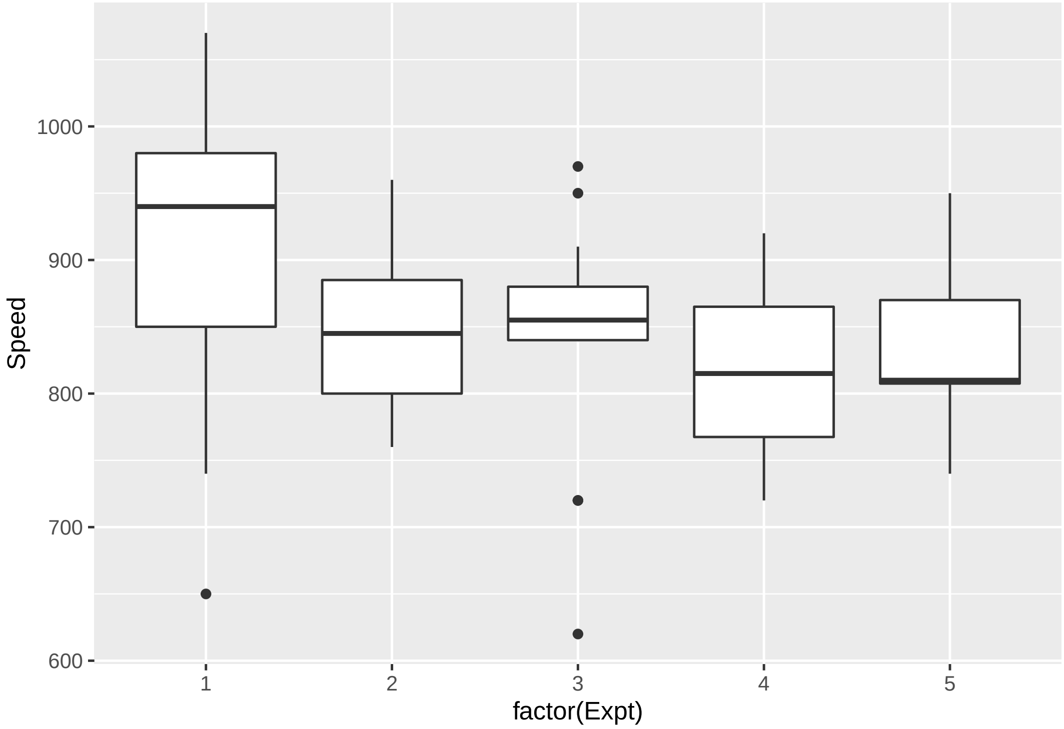
```
ineffective <- filter(InsectSprays, spray %in% c('A','B','F'))
anova(lm(count ~ spray, data=ineffective))
```

```
## Analysis of Variance Table
##
## Response: count
##           Df Sum Sq Mean Sq F value Pr(>F)
## spray      2  28.67  14.333  0.5435 0.5858
## Residuals 33 870.33  26.374
```

Problem 10

a.

```
ggplot(morley, aes(x = factor(Expt), y = Speed)) +
  geom_boxplot()
```



The boxplots indicate there may be a difference in the variances between groups.

b.

```
morley %>%
  group_by(Expt) %>%
  summarize(n = n(),
            sd = sd(Speed))
```

```
## # A tibble: 5 x 3
##    Expt     n    sd
##   <int> <int> <dbl>
## 1     1    20 105.
## 2     2    20  61.2
## 3     3    20  79.1
## 4     4    20  60.0
## 5     5    20  54.2
```

The simulation in the Error simulations section indicate that with equal sample sizes, a factor of 2 difference in standard deviations may not cause a large problem, so ANOVA or oneway.test could be appropriate.

c.

```
anova(lm(Speed ~ factor(Expt), data = morley))
```

```
## Analysis of Variance Table
##
## Response: Speed
##              Df Sum Sq Mean Sq F value   Pr(>F)
## factor(Expt)  4  94514 23628.5  4.2878 0.003114 **
## Residuals    95 523510  5510.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We conclude ($p = 0.003114$) that not all of the means are the same across all experiments.

d.

```
oneway.test(Speed ~ factor(Expt), data = morley)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  Speed and factor(Expt)
## F = 3.0061, num df = 4.000, denom df = 47.044, p-value = 0.02738
```

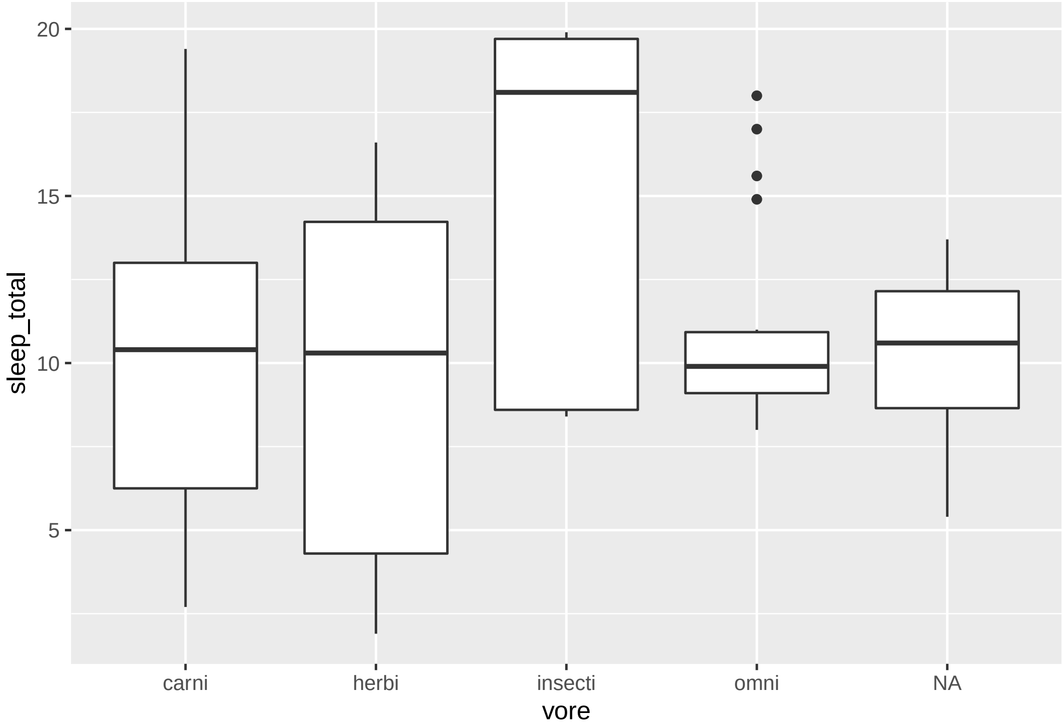
We conclude ($p = 0.02738$) that not all of the means are the same across all experiments.

Problem @msleepanova

```
table(msleep$vore)
```

```
##
##   carni   herbi insecti    omni
##      19      32       5      20
```

```
msleep %>% ggplot(aes(x=vore, y=sleep_total)) + geom_boxplot()
```



```
oneway.test(sleep_total ~ vore, data=msleep)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  sleep_total and vore
## F = 1.4047, num df = 3.000, denom df = 16.586, p-value = 0.2766
```

```
anova(lm(sleep_total ~ vore, data=msleep))
```

```
## Analysis of Variance Table
##
## Response: sleep_total
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## vore       3  133.72  44.573  2.2353 0.09143 .
## Residuals 72 1435.73  19.941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c. There is not significant evidence that the group means are different ($p = 0.2766$)
d. ANOVA reports $p = 0.9143$.

Problem 11

```
set.seed(3850)
dd <- data.frame(group = rep(letters[1:4], times = c(20, 20, 20, 100)))

sim_data <- replicate(10000, {
  dd$value <- c(rnorm(60), rnorm(100, 0, 2))
  anova(lm(value ~ group, data = dd))[1,5]
})
mean(sim_data < .05)
```

```
## [1] 0.0028
```

The effective type I error rate is approximately .0028.

Problem 12

```
set.seed(3850)
dd <- data.frame(group = rep(letters[1:4], times = c(20, 20, 20, 20)))

sim_data <- replicate(10000, {
  dd$value <- c(rnorm(60), rnorm(20, 0, 2))
  anova(lm(value ~ group, data = dd))[1,5]
})
mean(sim_data < .05)
```

```
## [1] 0.0632
```

The effective type I error rate is approximately 0.632, which is closer to the desire value of .05.

Problem 13

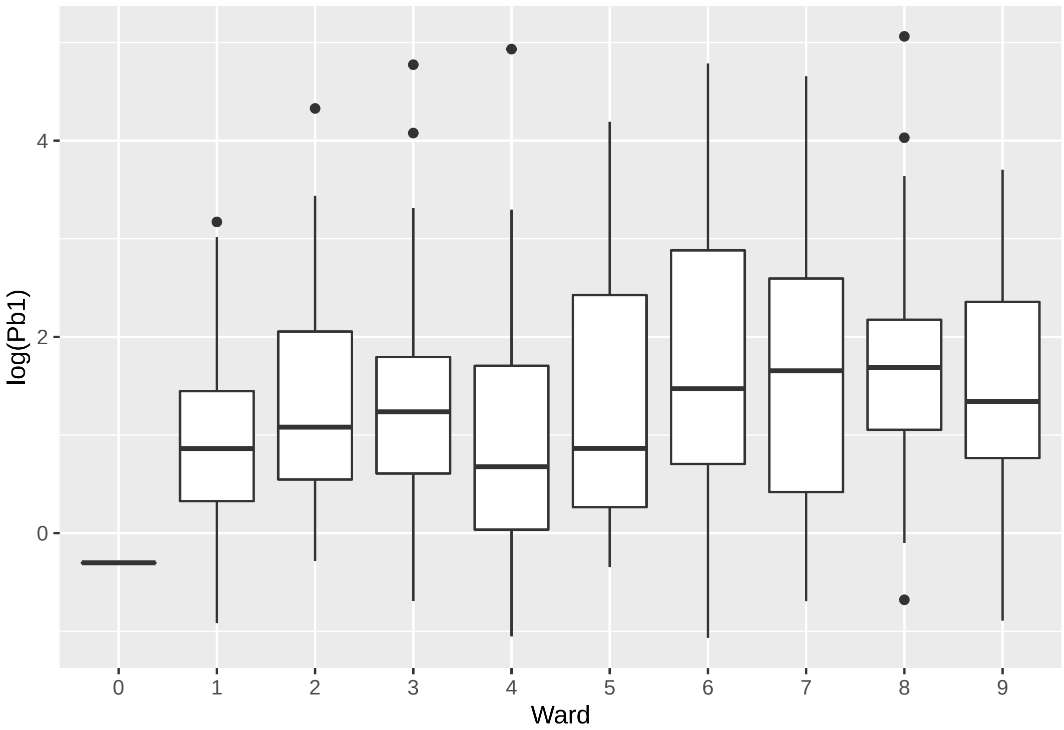
We do both parts together. The power associated with ANOVA is about 0.17, while that associated with pairwise t tests with Holm correction is about 0.12.

```
set.seed(3850)
dd <- data.frame(group = rep(letters[1:6], each = 15))

sim_data <- replicate(1000, {
  dd$value <- c(rnorm(45, 0, 3), rnorm(45, 1, 3))
  anova_reject <- anova(lm(value ~ group, data = dd))[1,5] < .05
  pairwise_reject <- any(pairwise.t.test(dd$value, dd$group)$p.value < .05, na.rm = T)
  c(anova_reject, pairwise_reject)
})

apply(sim_data, 1, function(x) mean(x))
```

```
## [1] 0.171 0.124
```

Problem 14

```
flint <- fosdata::flint
flint %>% ggplot(aes(x=Ward, y=log(Pb1)))+geom_boxplot()
```



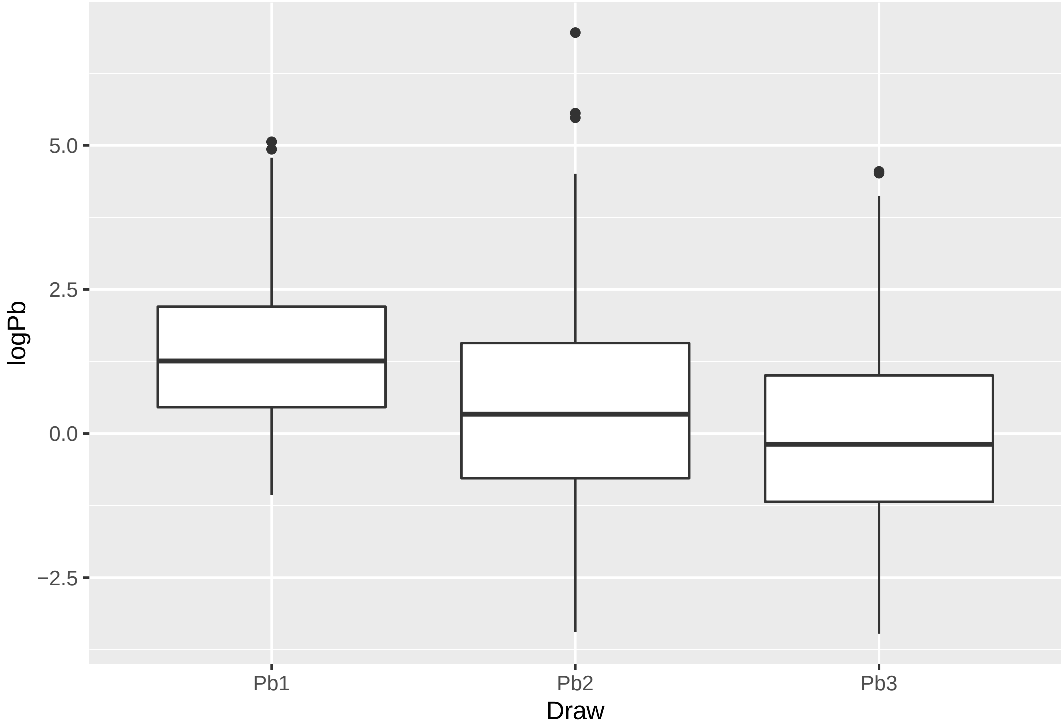
```
filter(flint, Ward != 0) %>% lm(log(Pb1) ~ Ward, data=.) %>% anova()
```

```
## Analysis of Variance Table
##
## Response: log(Pb1)
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Ward        8  22.61  2.8267  1.7191 0.09412 .
## Residuals 261 429.17  1.6443
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is not a significant difference in the $log(Pb1)$ level across the 9 wards of Flint ($p = 0.09412$).

Problem 15

```
flint_tidy <- fosdata::flint %>%
  tidyr::pivot_longer(cols = starts_with("Pb"), names_to = "Draw", values_to = "Pb") %>%
  mutate(logPb = log(Pb))
ggplot(flint_tidy, aes(x=Draw, y = logPb)) + geom_boxplot()
```



```
pairwise.t.test(x = flint_tidy$logPb, g = flint_tidy$Draw, paired=TRUE)
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  flint_tidy$logPb and flint_tidy$Draw
##
##     Pb1    Pb2
## Pb2 <2e-16 -
## Pb3 <2e-16 <2e-16
##
## P value adjustment method: holm
```

The lead levels (after log) are normally distributed and have roughly equal variances. However, the independence assumption for ANOVA is not satisfied because each house is repeated in all three draws. The pairwise t-test shows a significant difference between all three log lead levels, with $p < 2 \times 10^{-16}$ for all three comparisons.

Problem 16

```
sim_data <- replicate(10000, {
  three_groups <- data.frame(values = rnorm(90, rep(c(0, 0.5, 1), each = 30)),
                             groups = factor(rep(1:3, each = 30)))
  mean(pairwise.t.test(three_groups$values,
                 three_groups$groups,
                 p.adjust.method = "holm")$p.value < .05, na.rm = TRUE)
})
mean(sim_data)
```

```
## [1] 0.5757
```

About 0.57.

```
sim_data <- replicate(10000, {
  three_groups <- data.frame(values = rnorm(90, rep(c(0, 0.5, 1), each = 30)),
                             groups = factor(rep(1:3, each = 30)))
  mean(pairwise.t.test(three_groups$values,
                  three_groups$groups,
                  p.adjust.method = "bonferroni")$p.value < .05, na.rm = TRUE)

})
mean(sim_data)
```

```
## [1] 0.5147333
```

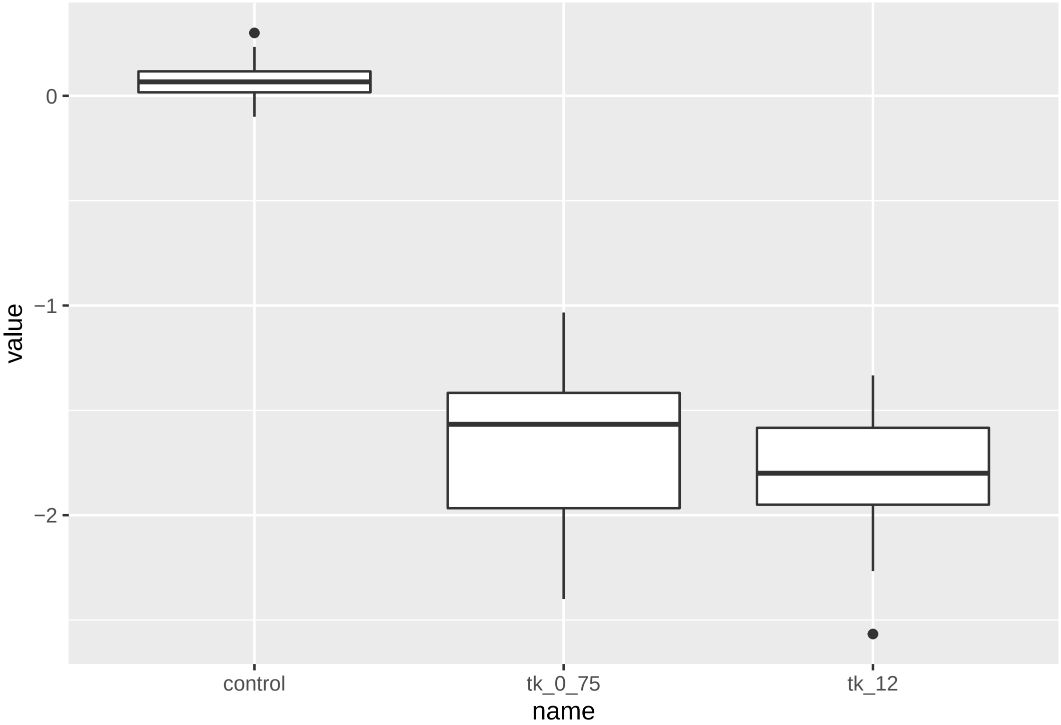
About 0.51.

Problem 17

a. The observations are **not** independent, so it would be inappropriate to use one-way ANOVA as described in this chapter.

b.

```
cows <- fosdata::cows_small
library(tidyr)
cows <- cows %>%
  pivot_longer(cols = !matches("cow"))

ggplot(cows, aes(x = name, y = value)) +
  geom_boxplot()
```



The standard deviations do not seem to be equal across the groups, so we will use `pool.sd = FALSE`.

```
pairwise.t.test(cows$value, cows$name, paired = T, pool.sd = F)
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  cows$value and cows$name
##
##         control tk_0_75
## tk_0_75 7.5e-14 -
## tk_12   1.7e-14 0.15
##
## P value adjustment method: holm
```

There is a significant difference between control and TK 0.75 $p = 7.5 \times 10^{-14}$ as well as between the control and TK 12 $p = 1.7 \times 10^{-14}$. There is not a significant difference between the nozzles $p = 0.15$.

Problem 18

a. The pairwise t-test on 6 groups requires 15 tests.

b. Each test has a probability of a type I error of .05.

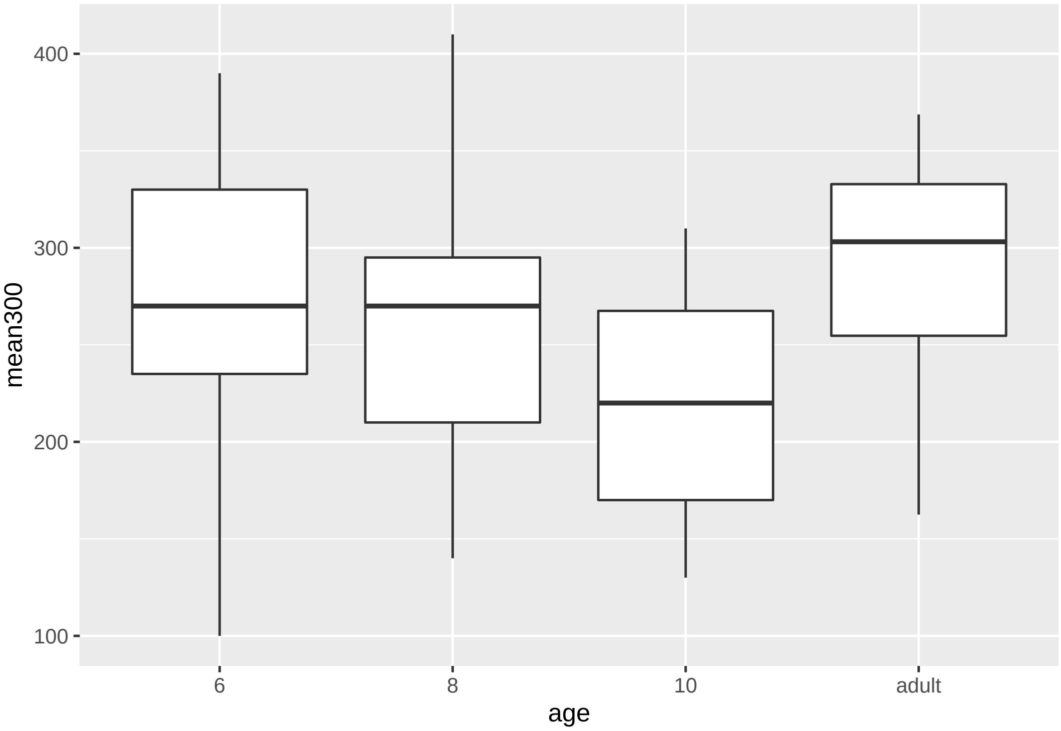
Assuming the tests are independent,

$$
\begin{aligned}
P(\text{Error on any test}) &= 1 - P(\text{No error on any test}) \tag{1}\\
&= 1 - P(\text{No error on test 1}) \cdot P(\text{No error on test 2}) \cdots P\,(\text{No error on test } M) \tag{2}\\
&= 1 - (0.95)^M \tag{3}
\end{aligned}
$$

b. When $k = 6$, $M = \binom{6}{2} = 15$, and $1 - .95^{15} \approx 0.54$. Thus FWER is approximately 54%. There is a 54% chance that one of our 15 t-tests will be wrong. From the chapter, the simulated FWER is 35%. The assumption that the pairwise tests are independent was faulty. If one group is different, it will have large differences with every other group.

Problem 19

```r
ww <- fosdata::weight_estimate
ggplot(ww, aes(x = age, y = mean300)) +
  geom_boxplot() #looks good
```



```r
pairwise.t.test(ww$mean300, ww$age)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  ww$mean300 and ww$age
##
##       6     8     10
## 8     0.805 -     -
## 10    0.040 0.083 -
## adult 0.805 0.702 0.003
##
```

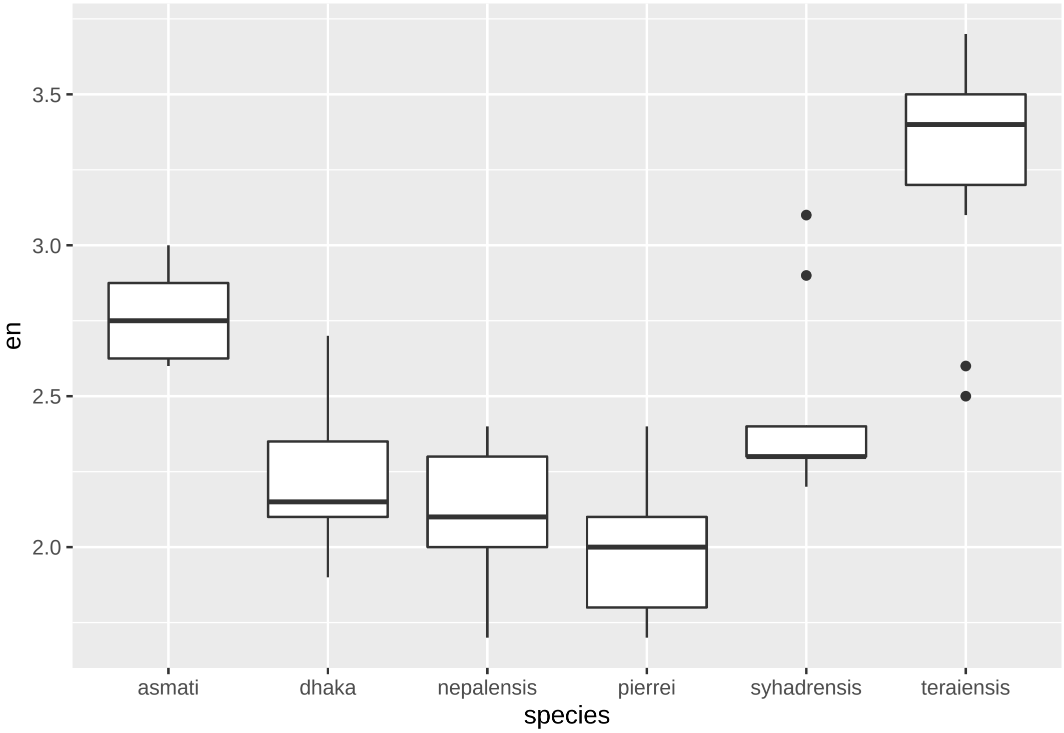
```
## P value adjustment method: holm
```

We see that there is a significant difference between adults and 10 year olds $p = .003$, as well as between 10 year olds and 6 year olds $p = .04$.

Problem 20

a. The sample standard deviations are within a factor of 2 of one another, and even though the design is pretty highly unbalanced (factor of 3.5 between lowest and highes number in groups) we continue with the analysis.

```
frogs <- fosdata::frogs
ggplot(frogs, aes(x = species, y = en)) +
  geom_boxplot()
```



```
frogs %>%
  group_by(species) %>%
  summarize(mu = mean(en),
            sd = sd(en),
            n = n())
```

```
## # A tibble: 6 x 4
##   species        mu    sd     n
##   <chr>       <dbl> <dbl> <int>
## 1 asmati       2.77 0.163     6
## 2 dhaka        2.23 0.246    12
## 3 nepalensis   2.11 0.226    11
## 4 pierrei      2    0.274     5
## 5 syhadrensis  2.44 0.324     9
## 6 teraiensis   3.33 0.314    21
```

b. Not all means are the same.

```
mod <- lm(en ~ species, data = frogs)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: en
##           Df Sum Sq Mean Sq F value    Pr(>F)
## species    5 17.666  3.5333  46.386 < 2.2e-16 ***
## Residuals 58  4.418  0.0762
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

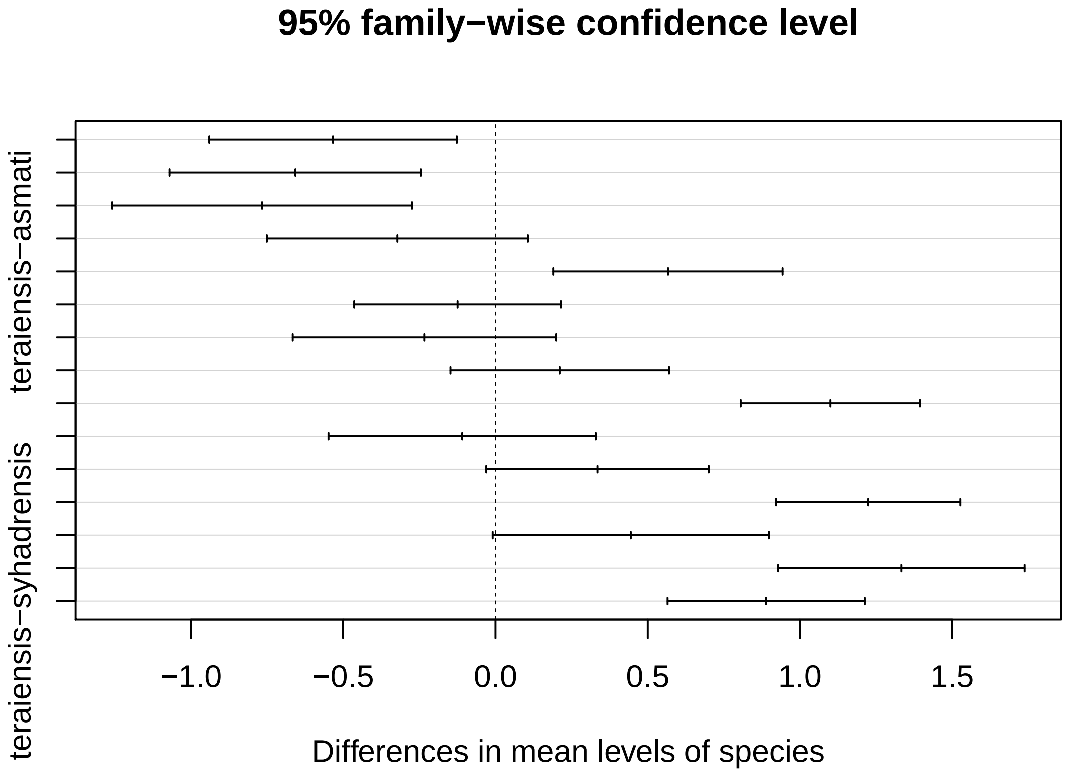
c. To perform Tukey's post-hoc test, we need to use aov.

```
mod_aov <- aov(en ~ species, data = frogs)
tukey_frog <- TukeyHSD(mod_aov)
tukey_frog
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = en ~ species, data = frogs)
##
## $species
##                              diff          lwr        upr     p adj
## dhaka-asmati           -0.5333333 -0.940023867 -0.1266428 0.0036786
## nepalensis-asmati      -0.6575758 -1.070382281 -0.2447692 0.0002345
## pierrei-asmati         -0.7666667 -1.259193362 -0.2741400 0.0003398
## syhadrensis-asmati     -0.3222222 -0.750911685  0.1064672 0.2467470
## teraiensis-asmati       0.5666667  0.190144397  0.9431889 0.0005713
## nepalensis-dhaka       -0.1242424 -0.463766841  0.2152820 0.8879612
## pierrei-dhaka          -0.2333333 -0.666288468  0.1996218 0.6094198
## syhadrensis-dhaka       0.2111111 -0.147556226  0.5697784 0.5151609
## teraiensis-dhaka        1.1000000  0.805659002  1.3943410 0.0000000
## pierrei-nepalensis     -0.1090909 -0.547796032  0.3296142 0.9770086
## syhadrensis-nepalensis  0.3353535 -0.030234067  0.7009411 0.0900014
## teraiensis-nepalensis   1.2242424  0.921507128  1.5269777 0.0000000
## syhadrensis-pierrei     0.4444444 -0.009237839  0.8981267 0.0580600
## teraiensis-pierrei      1.3333333  0.928584055  1.7380826 0.0000000
## teraiensis-syhadrensis  0.8888889  0.564830115  1.2129477 0.0000000
```

The eyes-to-nostril measurement distinguishes many pairs of species of frogs. The species teraiensis is distinguished from all 5 other species, and asmati is distinguished from all other species except for syhadrensis. Syhadrensis is only distinguished from teraiensis. Dhaka, nepalensis and pierrei are not distinguished from each other or syhadrensis, but are from teraiensis and asmati. A nice way to present this is through a plot:

```
plot(tukey_frog)
```



or as a table: species with the same letter do not have distinguised eye-nostril length.

Species	Mean eye-nostril length	grouping
teraiensis	3.33	a
asmati	2.766667	b
syhadrensis	2.444444	bc
dhaka	2.233333	c
nepalensis	2.109091	c
pierrei	2.000000	c

The table above can be obtained via `agricolae::HSD.test(mod_aov, "species", console = T)`

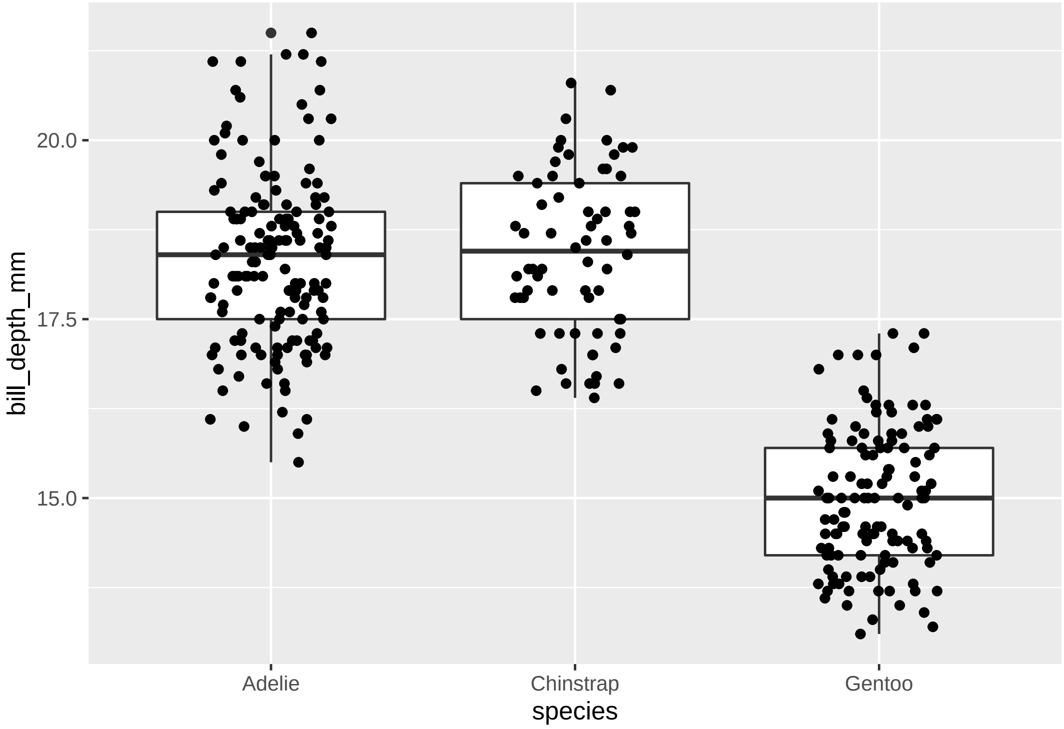
Problem 21

a.

```
pp <- palmerpenguins::penguins
ggplot(pp, aes(x = species, y = bill_depth_mm)) +
  geom_boxplot() +
  geom_jitter(height = 0, width = 0.2)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



```
pp %>%
  group_by(species) %>%
  summarize(n = n(),
            sd = sd(bill_depth_mm, na.rm = T))
```

```
## # A tibble: 3 x 3
##   species       n    sd
##   <fct>     <int> <dbl>
## 1 Adelie      152 1.22
## 2 Chinstrap    68 1.14
## 3 Gentoo      124 0.981
```

Appears to be roughly equal variances. Data appears symmetric with no outliers. Sample sizes are not equal, which could cause problems if the variances are not the same.

b.

```
anova(lm(bill_depth_mm ~ species, data = pp))
```

```
## Analysis of Variance Table
##
## Response: bill_depth_mm
##             Df Sum Sq Mean Sq F value    Pr(>F)
## species      2 903.97  451.98  359.79 < 2.2e-16 ***
## Residuals 339 425.87    1.26
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have evidence that the bill length is not the same across all of the species.

c.

```
TukeyHSD(aov(bill_depth_mm ~ species, data = pp))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = bill_depth_mm ~ species, data = pp)
##
## $species
##                         diff        lwr        upr     p adj
## Chinstrap-Adelie  0.07423062 -0.3110995  0.4595607 0.8928875
## Gentoo-Adelie    -3.36424379 -3.6847143 -3.0437733 0.0000000
## Gentoo-Chinstrap -3.43847441 -3.8371903 -3.0397586 0.0000000
```

We see that the mean bill length in mm of gentoo penguins is different from adelie and chinstrap penguins, while we fail to detect a difference between chinstrap and adelie penguins.

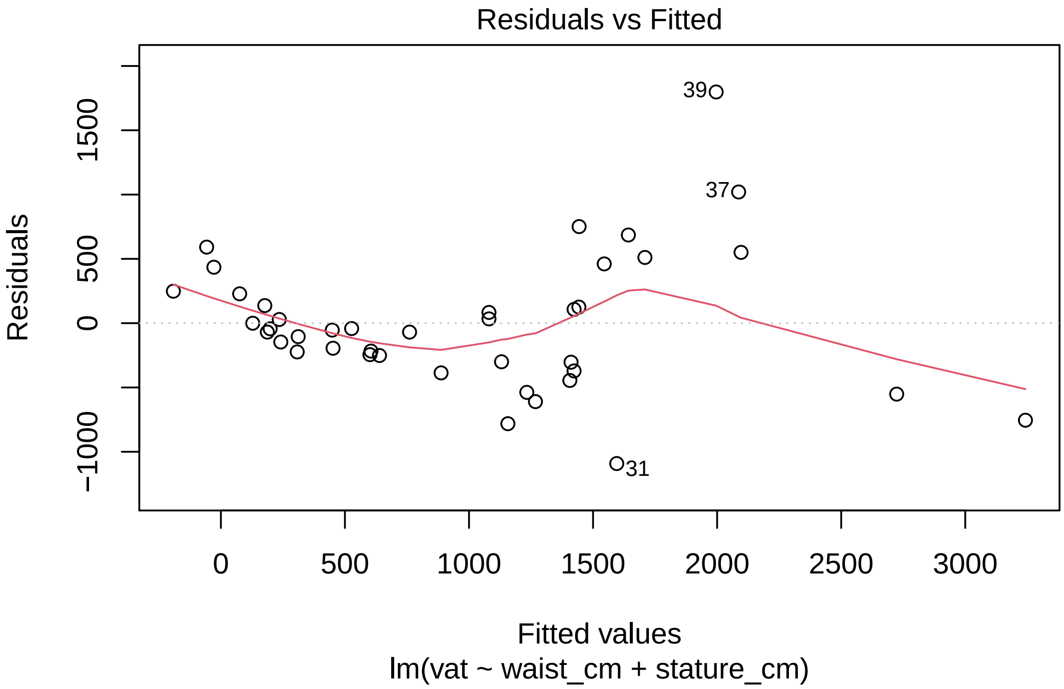
13

Multiple Regression – Solutions

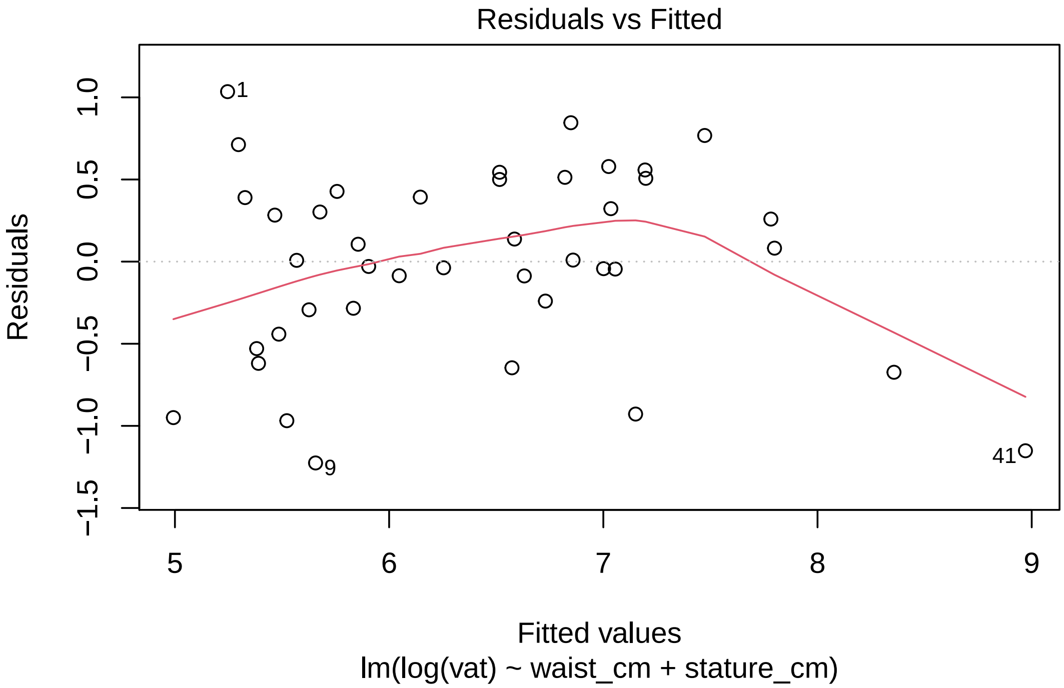
Problem 1

The residual vs. fitted plot for log of vat versus waist and stature looks like it has equal variance and no obvious trends. The residual vs. fitted of vat versus waist and stature has an obvious trend and is clearly not well modeled.

```
adipose <- fosdata::adipose

adipose %>% filter(sex == "Male") %>% lm(vat ~ waist_cm + stature_cm, data=.) %>% plot(which = 1
```



```
adipose %>% filter(sex == "Male") %>% lm(log(vat) ~ waist_cm + stature_cm, data=.) %>% plot(whic
```



Problem 2

a.

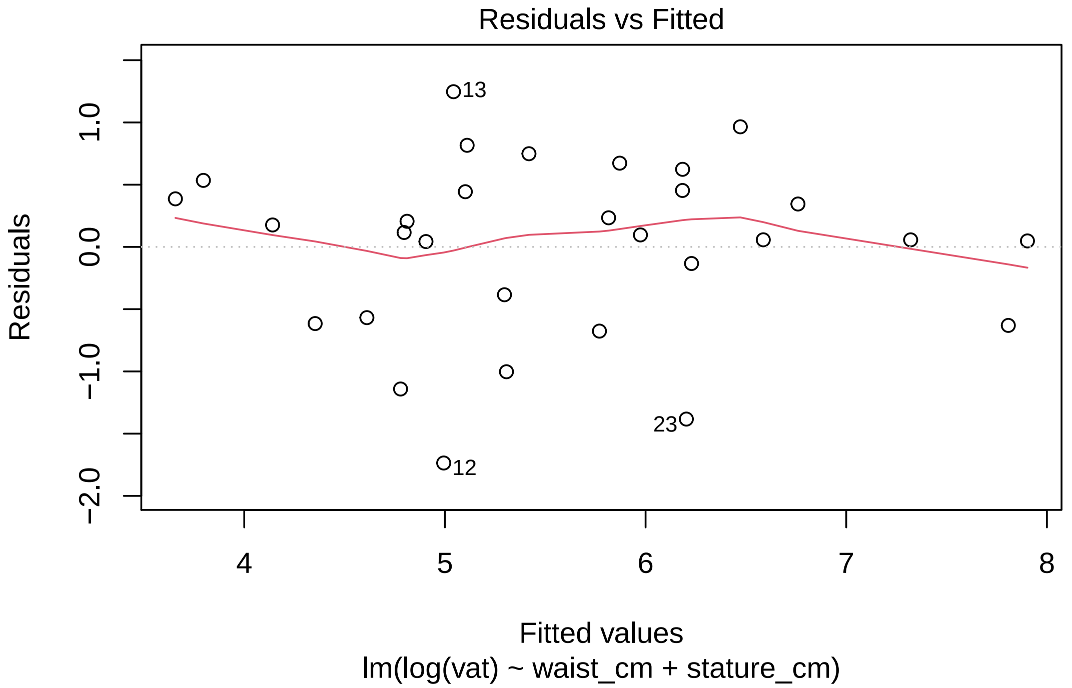
```
adipose <- fosdata::adipose %>%
  filter(sex == "Female", vat > 5)
```

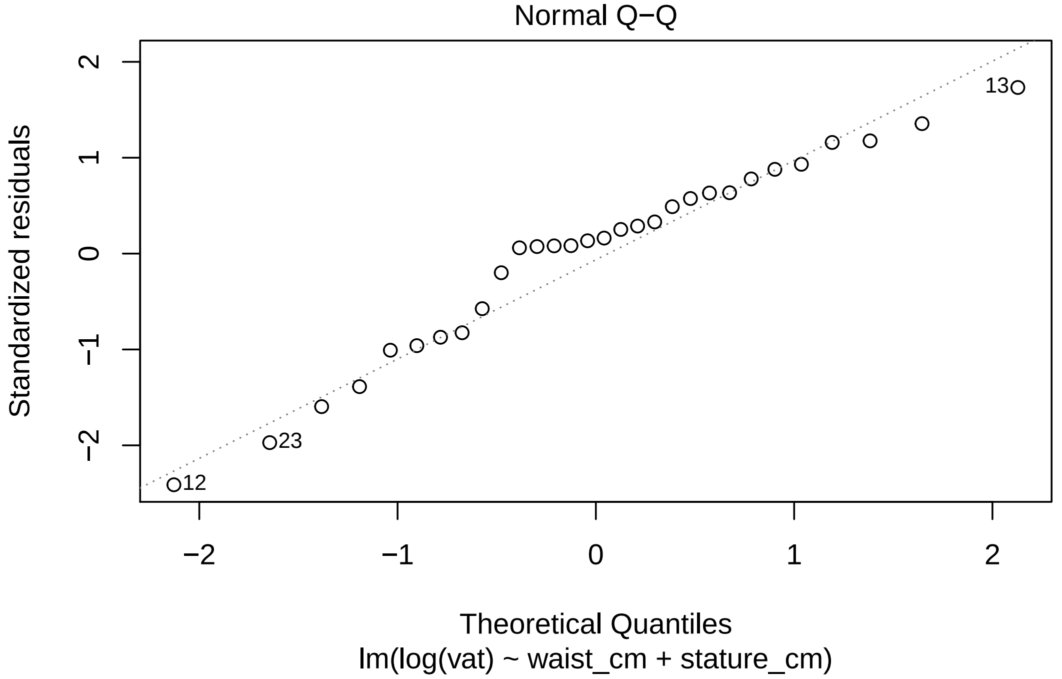
b.

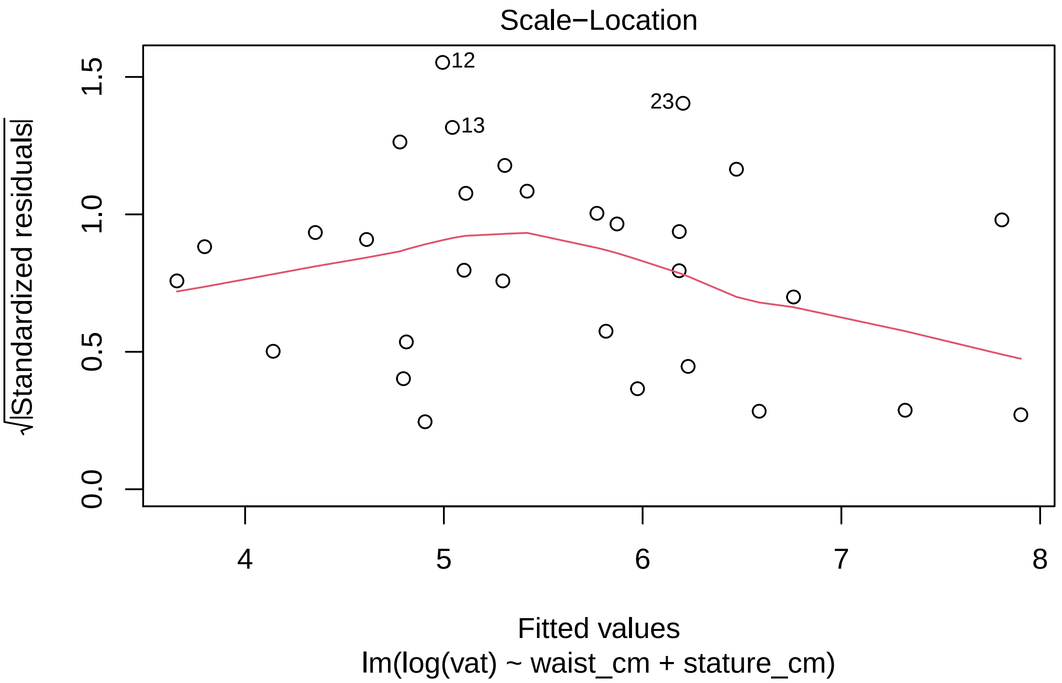
```
mod <- lm(log(vat) ~ waist_cm + stature_cm, data = adipose)
summary(mod)
```

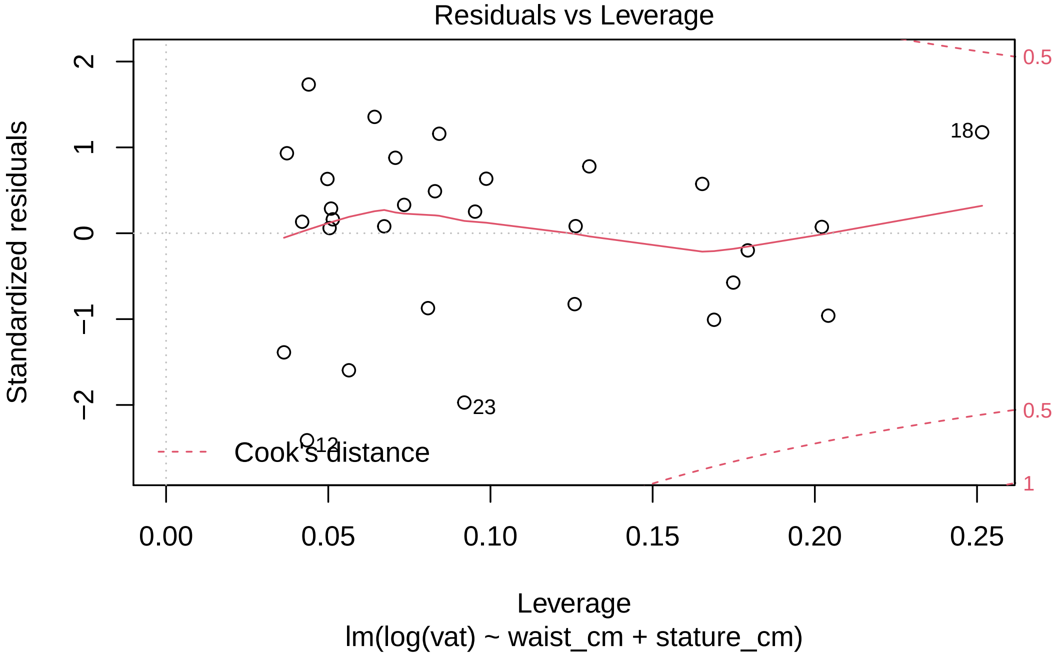
```
##
## Call:
## lm(formula = log(vat) ~ waist_cm + stature_cm, data = adipose)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7358 -0.5221  0.1062  0.4510  1.2470
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.15120    3.36640   0.936    0.358
## waist_cm     0.07888    0.01008   7.829 2.03e-08 ***
## stature_cm  -0.02445    0.02021  -1.210    0.237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7361 on 27 degrees of freedom
## Multiple R-squared:  0.6955, Adjusted R-squared:  0.673
## F-statistic: 30.84 on 2 and 27 DF,  p-value: 1.066e-07
```

```
plot(mod)
```


Residuals vs Fitted
Residuals
−2.0
−1.0
0.0
1.0
4
5
6
7
8
13
12
23
Fitted values
lm(log(vat) ~ waist_cm + stature_cm)


Normal Q−Q
Standardized residuals
−2
−1
0
1
2
−2
−1
0
1
2
13
23
12
Theoretical Quantiles
lm(log(vat) ~ waist_cm + stature_cm)





The model is $\widehat{\log}(\text{vat}) = 3.1512 + .0788 \times \text{waistcm} - .02445 \times \text{staturecm}$, and the residuals look pretty good.

c. The p-value for the test $H_0 : \beta_1 = 0$ versus $H_a : \beta_1 \neq 0$ is 2×10^{-8}.

d. A 95 percent confidence interval for β_2 is $[-0.066, 0.017]$.

```
confint(mod)
```

```
##                   2.5 %      97.5 %
## (Intercept) -3.75607615 10.05847045
## waist_cm     0.05821073  0.09955814
## stature_cm  -0.06592624  0.01702660
```

e. A 95 percent prediction interval for `vat` for the new patient is $[2.94, 6.09]$.

```
predict(mod, newdata = data.frame(waist_cm = 70, stature_cm = 170), interval = "pre")
```

```
##        fit      lwr      upr
## 1 4.516638 2.942904 6.090373
```

Problem 3

```
acorns <- fosdata::acorns %>%
  filter(Region == "Atlantic") %>%
  mutate(logrange = log(Range), logsize = log(Acorn_size))
lm(logrange ~ Tree_height + logsize, data=acorns) %>% summary()
```

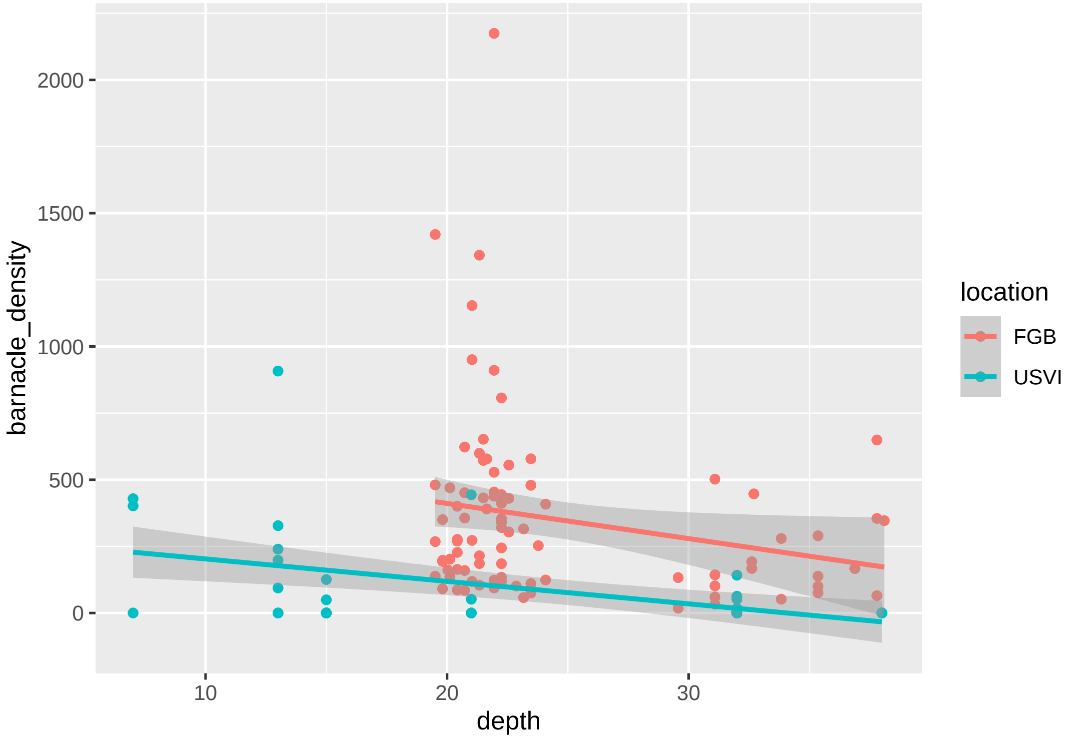
```
##
## Call:
## lm(formula = logrange ~ Tree_height + logsize, data = acorns)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9718 -0.4362  0.2291  0.6259  1.3560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.97007    0.37173  21.441  < 2e-16 ***
## Tree_height  0.02794    0.01961   1.425 0.166652
## logsize      0.70977    0.18784   3.779 0.000873 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8713 on 25 degrees of freedom
## Multiple R-squared:  0.4354, Adjusted R-squared:  0.3902
## F-statistic: 9.639 on 2 and 25 DF,  p-value: 0.0007885
```

Only the log of acorn size is significant. Removing the tree height variable reduces R^2 from 0.4354 to 0.3896, a fairly small change.

Problem 4

a.

```
barnacles <- fosdata::barnacles
barnacles %>% ggplot(aes(x = depth, y = barnacle_density, color=location)) +
  geom_point() + geom_smooth(method="lm")
```



b.

```
barn_mod <- lm(barnacle_density ~ location + depth, data=barnacles)
summary(barn_mod)$coefficients
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)   602.89390  87.605372  6.881928 2.202625e-10
## locationUSVI -270.92428  51.697139 -5.240605 6.230035e-07
## depth         -10.24441   3.371167 -3.038832 2.867762e-03
```

c.

```
predict(barn_mod, data.frame(depth = c(30,30), location = c("FGB","USVI")))
```

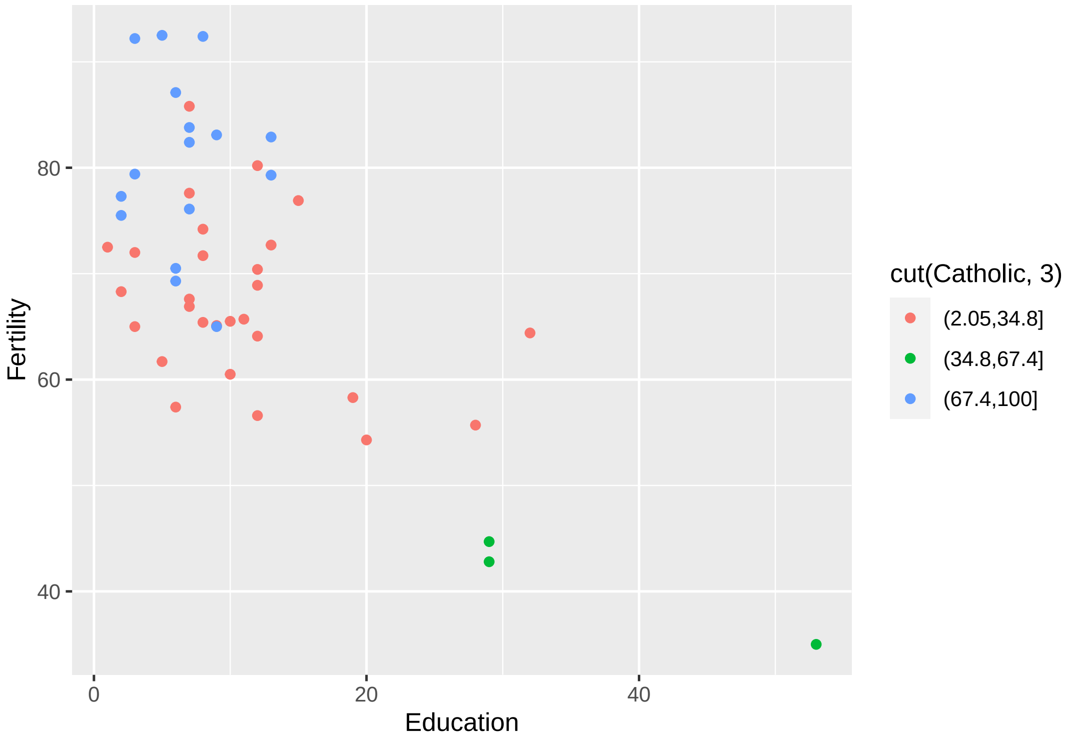
```
##         1         2
## 295.56157  24.63728
```

d. The coefficient means that the predicted mean barnacle density is 270.9 less in USVI than in FGB at any depth.

Problem 5

a. Provinces are either almost entirely Catholic or almost entirely Protestant. There are only three provinces with roughly equal representation of both religions, and these are the city of Geneva and two adjacent provinces.
b.

```
swiss %>% ggplot(aes(x=Education, y=Fertility, color=cut(Catholic,3))) + geom_point()
```



c.

```
summary(lm(Fertility ~ Agriculture + Examination + Education + Catholic + Infant.Mortality, data=
```

```
##
## Call:
## lm(formula = Fertility ~ Agriculture + Examination + Education +
##     Catholic + Infant.Mortality, data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2743  -5.2617   0.5032   4.1198  15.3213
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
## Agriculture      -0.17211    0.07030  -2.448  0.01873 *
## Examination      -0.25801    0.25388  -1.016  0.31546
## Education        -0.87094    0.18303  -4.758 2.43e-05 ***
## Catholic          0.10412    0.03526   2.953  0.00519 **
## Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.165 on 41 degrees of freedom
## Multiple R-squared:  0.7067, Adjusted R-squared:  0.671
## F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

All variables are significant except `Examination`.

d. Removing Examination lowered R^2 from 0.671 to 0.6707, barely any change.

Problem 6

a. We choose a sample size of 30.

```
dd <- data.frame(x1 = runif(30, -2, 2),
                 x2 = runif(30, -2, 2),
                 y = 2 + rnorm(30))
```

b.

```
mod <- lm(y ~ x1 + x2, data = dd)
```

c.

```
summary(mod)$fstatistic[1]
```

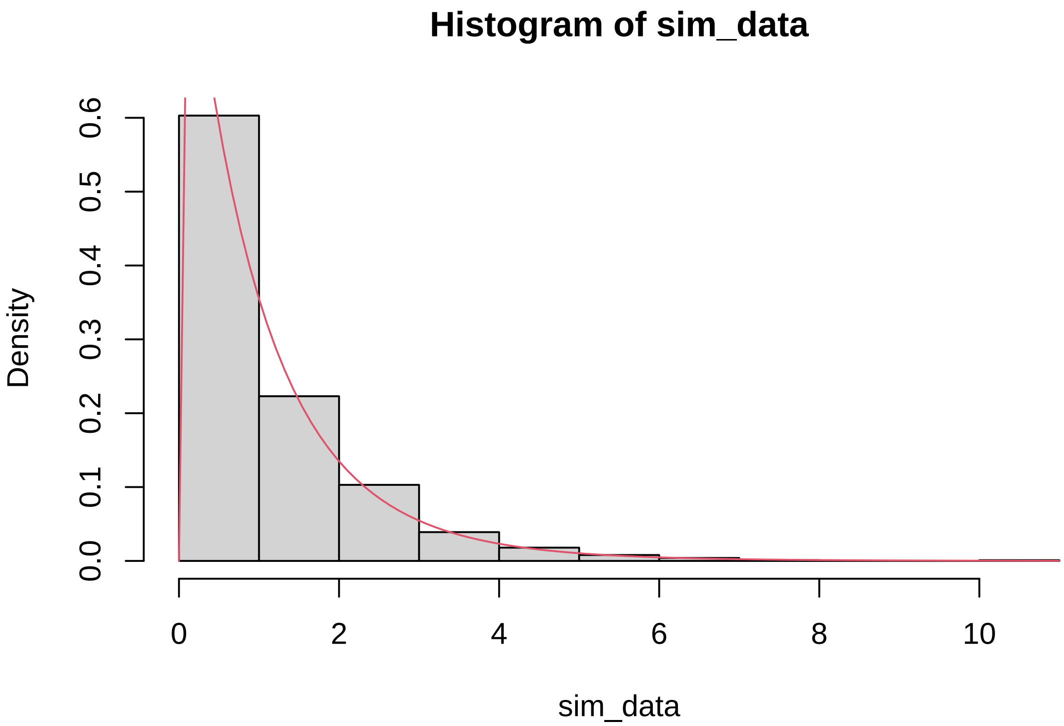
```
##    value
## 2.182037
```

d.

```
sim_data <- replicate(1000, {
  dd <- data.frame(x1 = runif(30, -2, 2),
                 x2 = runif(30, -2, 2),
                 y = 2 + rnorm(30))
  mod <- lm(y ~ x1 + x2, data = dd)
  summary(mod)$fstatistic[1]
})
```

e. The histogram of F statistics is consistent with an F distribution with 2, 27 degrees of freedom.

```
hist(sim_data, probability = T)
curve(df(x, 2, 27), add = T, col = 2)
```



Problem 7

c.

```
set.seed(3850)
sim_data <- replicate(10000, {
  df <- data.frame(x1 = runif(20, -2, 2),
                   x2 = runif(20, -2, 2))
  df$y <- 1 + 2 * df$x1 + rexp(20)
  mod <- lm(y ~ x1 + x2, data = df)
  summary(mod)$coefficients[3,4]
})
mean(sim_data < .05)
```

```
## [1] 0.0439
```

d. We get an effective type I error rate of .0439, which is pretty close to the value of .05 that would be expected if the error were normal.

Problem 8

```
penguins <- palmerpenguins::penguins
```

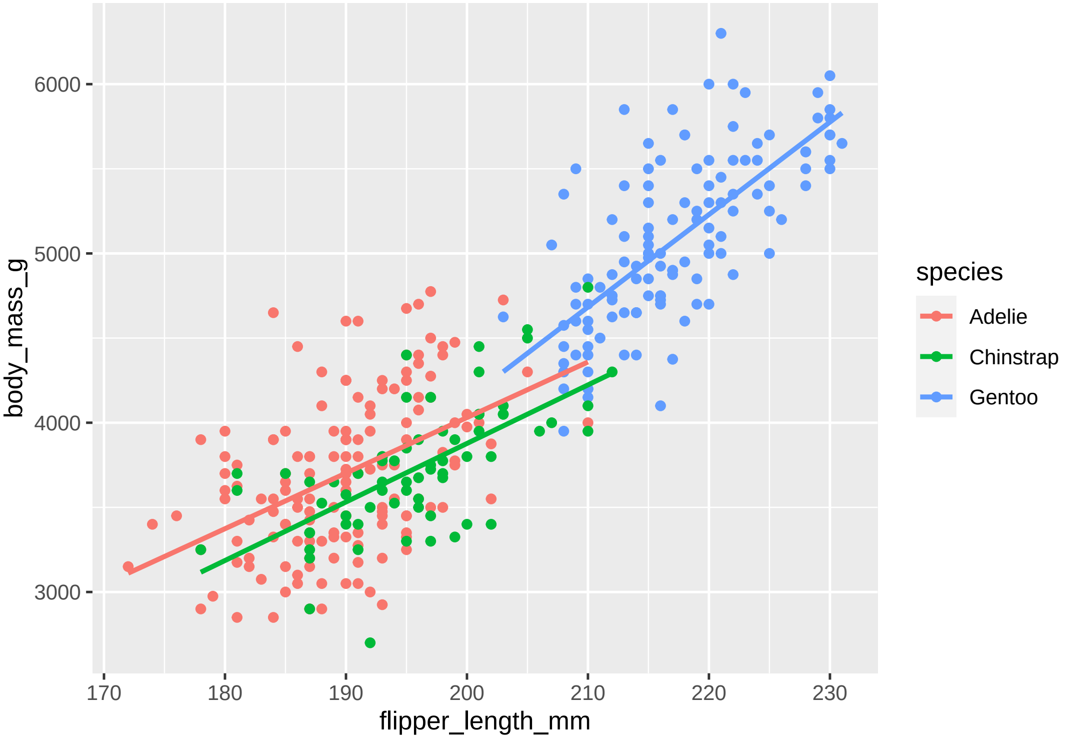
a.

```
penguins %>% ggplot(aes(x=flipper_length_mm,y=body_mass_g, color=species)) + geom_point() + geom_
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



b.

```
penguins_Vslope <- lm(body_mass_g ~ flipper_length_mm * species, data=penguins)
summary(penguins_Vslope)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm * species, data = penguins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -911.18 -251.93  -31.77  197.82 1144.81
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -2535.837    879.468  -2.883  0.00419 **
## flipper_length_mm                    32.832      4.627   7.095 7.69e-12 ***
## speciesChinstrap                   -501.359   1523.459  -0.329  0.74229
## speciesGentoo                     -4251.444   1427.332  -2.979  0.00311 **
## flipper_length_mm:speciesChinstrap    1.742      7.856   0.222  0.82467
## flipper_length_mm:speciesGentoo      21.791      6.941   3.139  0.00184 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 370.6 on 336 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7896, Adjusted R-squared:  0.7864
## F-statistic: 252.2 on 5 and 336 DF,  p-value: < 2.2e-16
```

c. The interaction term with gentoo is significant, but not with chinstrap. That's clear in the plot, because gentoo has a very different slope from adelie, but chinstrap is almost the same slope as adelie.

Problem 9 MSE for equal slopes is 142K, for variable slopes the MSE is 139K. The variable slopes model has better predictive value.

```
# loocv
loo_Eslope <- function(k) {
  train <- penguins[-k,]
  test  <- penguins[k,]
  penguins_Eslope <- lm(body_mass_g ~ flipper_length_mm + species, data=train)
  test$body_mass_g - predict(penguins_Eslope, test)
}
errs_E <- sapply(1:nrow(penguins), loo_Eslope)
mean(errs_E^2, na.rm=TRUE)
```

```
## [1] 142423.5
```

```
loo_Vslope <- function(k) {
  train <- penguins[-k,]
  test  <- penguins[k,]
  penguins_Vslope <- lm(body_mass_g ~ flipper_length_mm * species, data=train)
  test$body_mass_g - predict(penguins_Vslope, test)
}
errs_V <- sapply(1:nrow(penguins), loo_Vslope)
mean(errs_V^2, na.rm=TRUE)
```

```
## [1] 139117.5
```

Problem 10

a. We will want to recode `species` as a factor.

```
fish <- fosdata::fish
fish <- mutate(fish, species = factor(species, labels = c("Bream", "Whitefish", "Roach", "Parkki'
```

b.

```
mod1 <- lm(weight ~ length1 * species, data = fish)
summary(mod1)
```

```
##
## Call:
## lm(formula = weight ~ length1 * species, data = fish)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -193.36  -39.05   -1.74   23.79  333.24
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1015.0498   116.3407  -8.725 5.99e-15 ***
## length1                    54.1075     3.8093  14.204  < 2e-16 ***
## speciesWhitefish          -45.5700   220.1870  -0.207 0.836333
## speciesRoach              685.6736   160.5456   4.271 3.52e-05 ***
## speciesParkki             733.3764   186.5252   3.932 0.000131 ***
## speciesSmelt              989.9905   236.4418   4.187 4.90e-05 ***
## speciesPike               -525.7745   150.6962  -3.489 0.000644 ***
## speciesPerch               395.8747   121.2186   3.266 0.001364 **
## length1:speciesWhitefish     1.1570     7.4408   0.155 0.876646
## length1:speciesRoach       -30.7883     6.5178  -4.724 5.45e-06 ***
## length1:speciesParkki      -30.7997     8.5715  -3.593 0.000448 ***
## length1:speciesSmelt       -50.8884    18.5815  -2.739 0.006948 **
## length1:speciesPike         -0.9127     4.4032  -0.207 0.836088
## length1:speciesPerch       -15.1961     4.0111  -3.789 0.000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.76 on 144 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9547, Adjusted R-squared:  0.9507
## F-statistic: 233.7 on 13 and 144 DF,  p-value: < 2.2e-16
```

c. The expected weight is 232.6.

```
predict(mod1, newdata = data.frame(species = "Roach", length1 = 24.1))
```

```
##        1
## 232.6181
```

d. The expected difference is 54.1075 - 30.7883 = 23.3. We check using `predict`.

```
predict(mod1, newdata = data.frame(species = "Roach", length1 = 23.1)) - predict(mod1, newdata =
```

```
##        1
## 23.31926
```

Problem 11

```
conver <- fosdata::conversation %>%
  mutate(gender = factor(gender, labels = c("Male", "Female")))
mod <- lm(proportion_words ~ gender + f1_psychopathy + f2_psychopathy + total_psychopathy + attra
summary(mod)
```

```
##
## Call:
## lm(formula = proportion_words ~ gender + f1_psychopathy + f2_psychopathy +
##     total_psychopathy + attractiveness + fighting_ability + strength +
##     height + median_income + highest_class_rank + major_presige +
##     dyad_status_difference, data = conver)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25584 -0.08240 -0.00371  0.09083  0.30167
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.3235746  0.0340683   9.498  < 2e-16 ***
## genderFemale            0.0280451  0.0188090   1.491   0.1376
## f1_psychopathy         -1.2741589  0.7630936  -1.670   0.0966 .
## f2_psychopathy         -0.7282022  0.4090278  -1.780   0.0766 .
## total_psychopathy       1.7194755  0.9930276   1.732   0.0849 .
## attractiveness          0.0018322  0.0094146   0.195   0.8459
## fighting_ability       -0.0279085  0.0141531  -1.972   0.0500 .  
## strength                0.0307189  0.0139481   2.202   0.0288 *  
## height                 -0.0010481  0.0095064  -0.110   0.9123    
## median_income          -0.0001206  0.0003411  -0.354   0.7240    
## highest_class_rank      0.0783975  0.0190457   4.116 5.66e-05 ***
## major_presige          -0.0008824  0.0008975  -0.983   0.3267    
## dyad_status_difference -0.0310579  0.0119824  -2.592   0.0103 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1255 on 197 degrees of freedom
## Multiple R-squared:  0.2452, Adjusted R-squared:  0.1992 
## F-statistic: 5.332 on 12 and 197 DF,  p-value: 8.689e-08
```

We start with psychopathy. It seems unlikely that we will be able to remove all of those variables.

```
mod2 <- lm(proportion_words ~ gender + attractiveness + fighting_ability + strength + height + me
anova(mod2, mod)
```

```
## Analysis of Variance Table
## 
## Model 1: proportion_words ~ gender + attractiveness + fighting_ability + 
##     strength + height + median_income + highest_class_rank + 
##     major_presige + dyad_status_difference
## Model 2: proportion_words ~ gender + f1_psychopathy + f2_psychopathy + 
##     total_psychopathy + attractiveness + fighting_ability + strength + 
##     height + median_income + highest_class_rank + major_presige + 
##     dyad_status_difference
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1    200 3.4803                                  
## 2    197 3.1043  3   0.37596 7.9527 4.947e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we remove `f1_psychopathy` then the other two are significant.

```
mod3 <- lm(proportion_words ~ gender + f2_psychopathy + total_psychopathy + attractiveness + figh
summary(mod3)
```

```
## 
## Call:
## lm(formula = proportion_words ~ gender + f2_psychopathy + total_psychopathy + 
##     attractiveness + fighting_ability + strength + height + median_income + 
##     highest_class_rank + major_presige + dyad_status_difference, 
##     data = conver)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.24861 -0.08977 -0.00654  0.09817  0.32739 
## 
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             0.3154439  0.0338704   9.313  < 2e-16 ***
## genderFemale           0.0215050  0.0184796   1.164 0.245940    
## f2_psychopathy        -0.0455914  0.0132358  -3.445 0.000698 ***
## total_psychopathy      0.0615409  0.0134764   4.567 8.70e-06 ***
## attractiveness         0.0008450  0.0094384   0.090 0.928752    
## fighting_ability      -0.0315326  0.0140487  -2.245 0.025905 *  
## strength               0.0355131  0.0137109   2.590 0.010307 *  
## height                -0.0044023  0.0093335  -0.472 0.637683    
## median_income         -0.0001753  0.0003410  -0.514 0.607861    
## highest_class_rank     0.0801281  0.0191032   4.194 4.12e-05 ***
## major_presige         -0.0004910  0.0008702  -0.564 0.573234    
## dyad_status_difference -0.0316386  0.0120313  -2.630 0.009218 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1261 on 198 degrees of freedom
## Multiple R-squared:  0.2345, Adjusted R-squared:  0.192 
## F-statistic: 5.513 on 11 and 198 DF,  p-value: 1.144e-07
```

Next, we conisder gender, attractiveness, fighting ability, strength and height. We remove attractiveness, height and gender, and check with ANOVA whether we have fone too far. We haven't.

```
mod4 <- lm(proportion_words ~ f2_psychopathy + total_psychopathy + fighting_ability + strength +
anova(mod4, mod3)
```

```
## Analysis of Variance Table
## 
## Model 1: proportion_words ~ f2_psychopathy + total_psychopathy + fighting_ability + 
##     strength + median_income + highest_class_rank + major_presige + 
##     dyad_status_difference
## Model 2: proportion_words ~ gender + f2_psychopathy + total_psychopathy + 
##     attractiveness + fighting_ability + strength + height + median_income + 
##     highest_class_rank + major_presige + dyad_status_difference
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    201 3.1751                           
## 2    198 3.1483  3  0.026866 0.5632 0.6399
```

```
summary(mod4)
```

```
## 
## Call:
## lm(formula = proportion_words ~ f2_psychopathy + total_psychopathy + 
##     fighting_ability + strength + median_income + highest_class_rank + 
##     major_presige + dyad_status_difference, data = conver)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.25876 -0.09129 -0.00019  0.09600  0.32195 
## 
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             0.3243515  0.0328122   9.885  < 2e-16 ***
## f2_psychopathy         -0.0438687  0.0127421  -3.443  0.00070 ***
## total_psychopathy       0.0588353  0.0126376   4.656 5.86e-06 ***
## fighting_ability       -0.0324825  0.0139584  -2.327  0.02096 *
## strength                0.0361225  0.0135819   2.660  0.00845 **
## median_income          -0.0001606  0.0003381  -0.475  0.63538
## highest_class_rank      0.0795558  0.0187405   4.245 3.34e-05 ***
## major_presige          -0.0003717  0.0008515  -0.437  0.66293
## dyad_status_difference -0.0308093  0.0115947  -2.657  0.00851 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1257 on 201 degrees of freedom
## Multiple R-squared:  0.2279, Adjusted R-squared:  0.1972
## F-statistic: 7.418 on 8 and 201 DF,  p-value: 1.202e-08
```

Finally, we remove major prestige and median income. There is not sufficient evidence to add back in any variables, so this is our final model.

```
mod5 <- lm(proportion_words ~ f2_psychopathy + total_psychopathy + fighting_ability + strength +
anova(mod5, mod)
```

```
## Analysis of Variance Table
##
## Model 1: proportion_words ~ f2_psychopathy + total_psychopathy + fighting_ability +
##     strength + highest_class_rank + dyad_status_difference
## Model 2: proportion_words ~ gender + f1_psychopathy + f2_psychopathy +
##     total_psychopathy + attractiveness + fighting_ability + strength +
##     height + median_income + highest_class_rank + major_presige +
##     dyad_status_difference
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    203 3.1816
## 2    197 3.1043  6  0.077295 0.8175 0.5575
```

```
anova(mod5, mod4)
```

```
## Analysis of Variance Table
##
## Model 1: proportion_words ~ f2_psychopathy + total_psychopathy + fighting_ability +
##     strength + highest_class_rank + dyad_status_difference
## Model 2: proportion_words ~ f2_psychopathy + total_psychopathy + fighting_ability +
##     strength + median_income + highest_class_rank + major_presige +
##     dyad_status_difference
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    203 3.1816
## 2    201 3.1751  2 0.0064956 0.2056 0.8143
```

b. The R-squared is 0.2264, which means that there is quite a lot of variance in the proportion of words spoken that is not explained by the variables in our model.

```
summary(mod5)
```

```
##
## Call:
## lm(formula = proportion_words ~ f2_psychopathy + total_psychopathy +
##     fighting_ability + strength + highest_class_rank + dyad_status_difference,
##     data = conver)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.26024 -0.09503 -0.00538  0.09827  0.32163 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             0.30455    0.01077  28.289  < 2e-16 ***
## f2_psychopathy         -0.04468    0.01261  -3.543 0.000492 ***
## total_psychopathy       0.05950    0.01254   4.747 3.90e-06 ***
## fighting_ability       -0.03411    0.01360  -2.509 0.012891 *  
## strength                0.03699    0.01346   2.749 0.006510 ** 
## highest_class_rank      0.08098    0.01851   4.376 1.93e-05 ***
## dyad_status_difference -0.03021    0.01143  -2.642 0.008875 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1252 on 203 degrees of freedom
## Multiple R-squared:  0.2264, Adjusted R-squared:  0.2035 
## F-statistic:   9.9 on 6 and 203 DF,  p-value: 1.411e-09
```

Problem 12

```
adipose <- fosdata::adipose
```

We start by restricting to data without missing values and selecting the numeric values.

```
adipose <- filter(adipose, sex == "Male") %>%
  select(-sex, -health)
adipose <- adipose[complete.cases(adipose),]
mod <- lm(log(vat) ~ ., data = adipose)
summary(mod)
```

```
## 
## Call:
## lm(formula = log(vat) ~ ., data = adipose)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.04690 -0.12191  0.04265  0.23827  0.80437 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.145379   3.496459  -0.042 0.967178    
## age          0.024221   0.006329   3.827 0.000815 ***
## ldl         -0.033052   0.130529  -0.253 0.802260    
## hdl         -0.351291   0.432211  -0.813 0.424336    
## trig         0.059594   0.153990   0.387 0.702165    
## glucose     -0.126080   0.058778  -2.145 0.042280 *  
## stature_cm  -0.000354   0.018990  -0.019 0.985282    
## waist_cm     0.039148   0.021379   1.831 0.079526 .  
## hips_cm      0.019607   0.017994   1.090 0.286683    
## bmi          0.050773   0.055113   0.921 0.366088    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4948 on 24 degrees of freedom
## Multiple R-squared:  0.8484, Adjusted R-squared:  0.7916
## F-statistic: 14.93 on 9 and 24 DF,  p-value: 8.075e-08
```

We now consider the blood measurements in order: ldl, trig, hdl and glucose. By the time we remove three variables, glucose is significant.

```
mod2 <- lm(log(vat) ~ . - ldl - trig - hdl, data = adipose)
summary(mod2)
```

```
##
## Call:
## lm(formula = log(vat) ~ . - ldl - trig - hdl, data = adipose)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10970 -0.20784  0.01699  0.24957  0.70500
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.353566   3.343956  -0.106 0.916576
## age          0.021758   0.005080   4.283 0.000208 ***
## glucose     -0.125757   0.054440  -2.310 0.028769 *
## stature_cm  -0.002515   0.018190  -0.138 0.891070
## waist_cm     0.052714   0.017342   3.040 0.005212 **
## hips_cm      0.012866   0.015490   0.831 0.413493
## bmi          0.040419   0.052796   0.766 0.450575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4801 on 27 degrees of freedom
## Multiple R-squared:  0.8395, Adjusted R-squared:  0.8038
## F-statistic: 23.53 on 6 and 27 DF,  p-value: 1.531e-09
```

```
anova(mod2, mod)
```

```
## Analysis of Variance Table
##
## Model 1: log(vat) ~ (age + ldl + hdl + trig + glucose + stature_cm + waist_cm +
##     hips_cm + bmi) - ldl - trig - hdl
## Model 2: log(vat) ~ age + ldl + hdl + trig + glucose + stature_cm + waist_cm +
##     hips_cm + bmi
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     27 6.2226
## 2     24 5.8747  3   0.34791 0.4738 0.7034
```

Next, we consider body shape measurements.

```
mod3 <- lm(log(vat) ~ age + glucose  + waist_cm, data = adipose)
summary(mod3)
```

```
##
## Call:
## lm(formula = log(vat) ~ age + glucose + waist_cm, data = adipose)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0565 -0.3456  0.1562  0.3486  0.6354
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.371374   0.589777  -0.630   0.5337
## age          0.021621   0.004434   4.877 3.30e-05 ***
## glucose     -0.136376   0.051736  -2.636   0.0132 *
## waist_cm     0.074688   0.007467  10.003 4.55e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.476 on 30 degrees of freedom
## Multiple R-squared:  0.8247, Adjusted R-squared:  0.8071
## F-statistic: 47.03 on 3 and 30 DF,  p-value: 1.865e-11
```

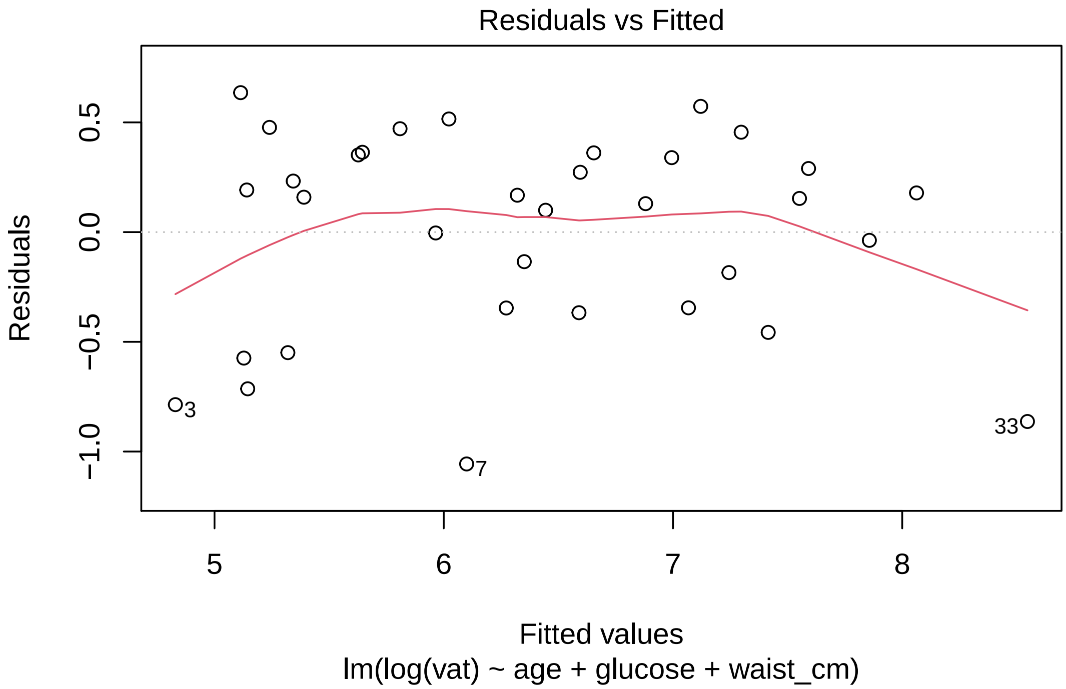
```
anova(mod3, mod2)
```

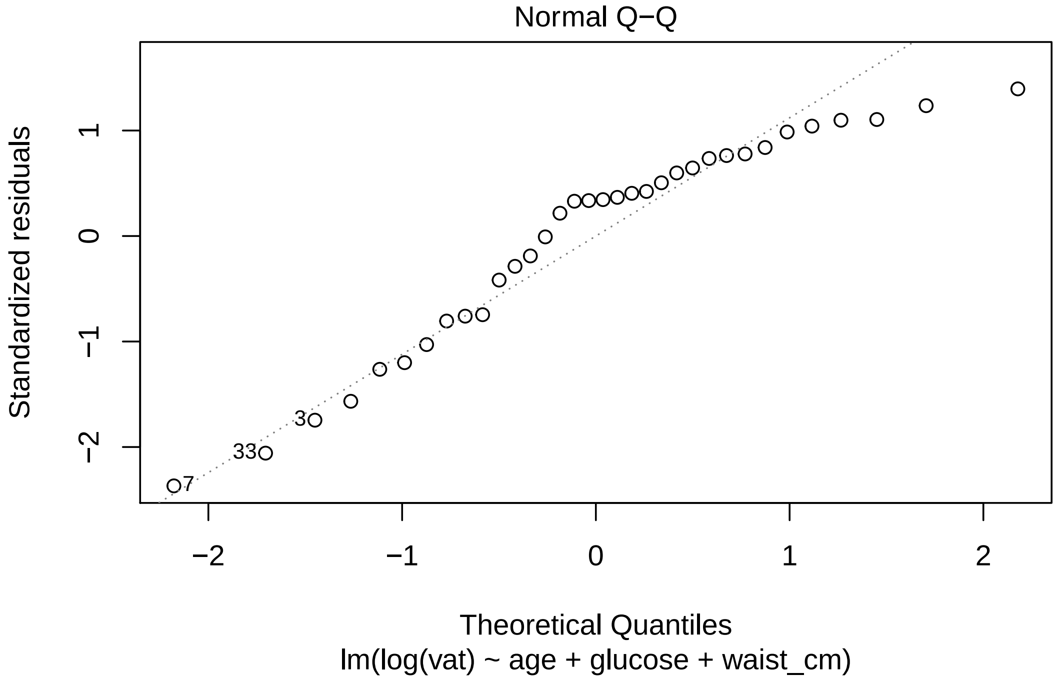
```
## Analysis of Variance Table
##
## Model 1: log(vat) ~ age + glucose + waist_cm
## Model 2: log(vat) ~ (age + ldl + hdl + trig + glucose + stature_cm + waist_cm +
##     hips_cm + bmi) - ldl - trig - hdl
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1     30 6.7972
## 2     27 6.2226  3   0.57457 0.831 0.4885
```

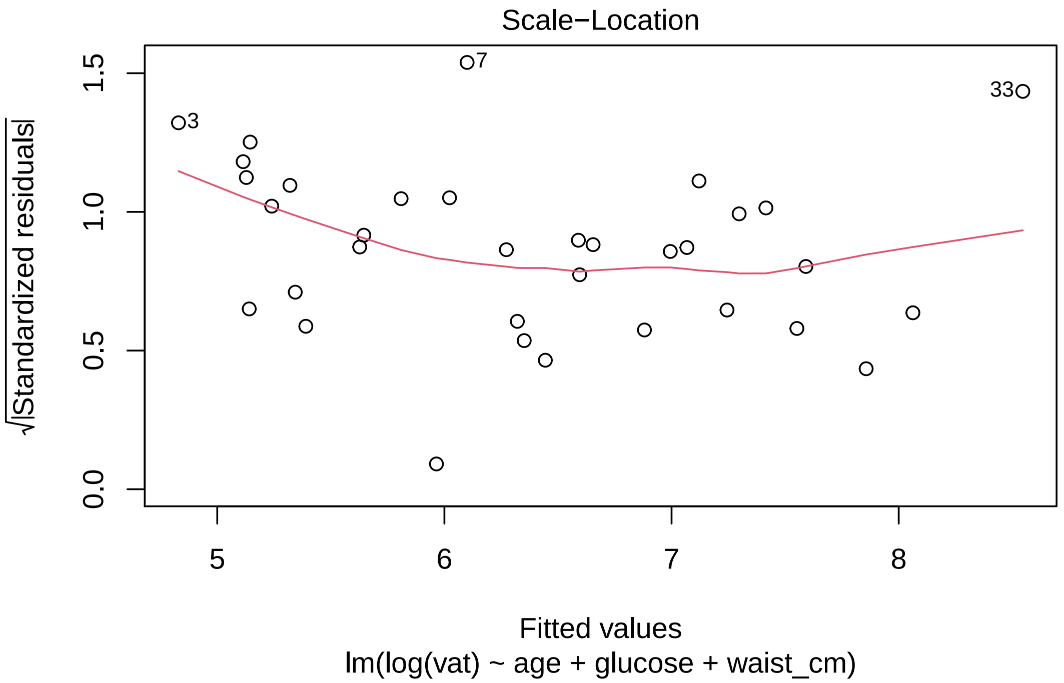
```
anova(mod3, mod)
```

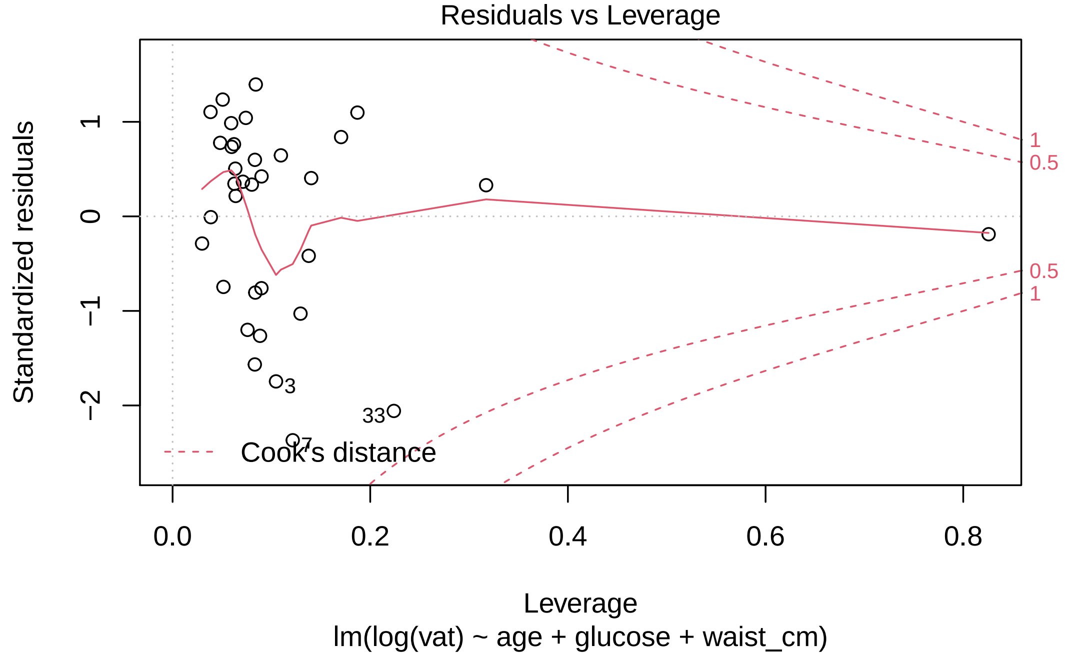
```
## Analysis of Variance Table
##
## Model 1: log(vat) ~ age + glucose + waist_cm
## Model 2: log(vat) ~ age + ldl + hdl + trig + glucose + stature_cm + waist_cm +
##     hips_cm + bmi
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     30 6.7972
## 2     24 5.8747  6   0.92248 0.6281 0.7063
```

```
plot(mod3)
```


Residuals vs Fitted
Residuals
0.5
0.0
−0.5
−1.0
3
7
33
5
6
7
8
Fitted values
lm(log(vat) ~ age + glucose + waist_cm)


Normal Q−Q
Standardized residuals
1
0
−1
−2
7
33
3
−2
−1
0
1
2
Theoretical Quantiles
lm(log(vat) ~ age + glucose + waist_cm)





The final model depends on age, glucose and waist measurement in centimeters.

b. The expected value for the log of the vat is about 4.8.

```
predict(mod3, newdata = data.frame(age = 20, glucose = 4.6, waist_cm = 72))
```

```
##        1
## 4.811295
```

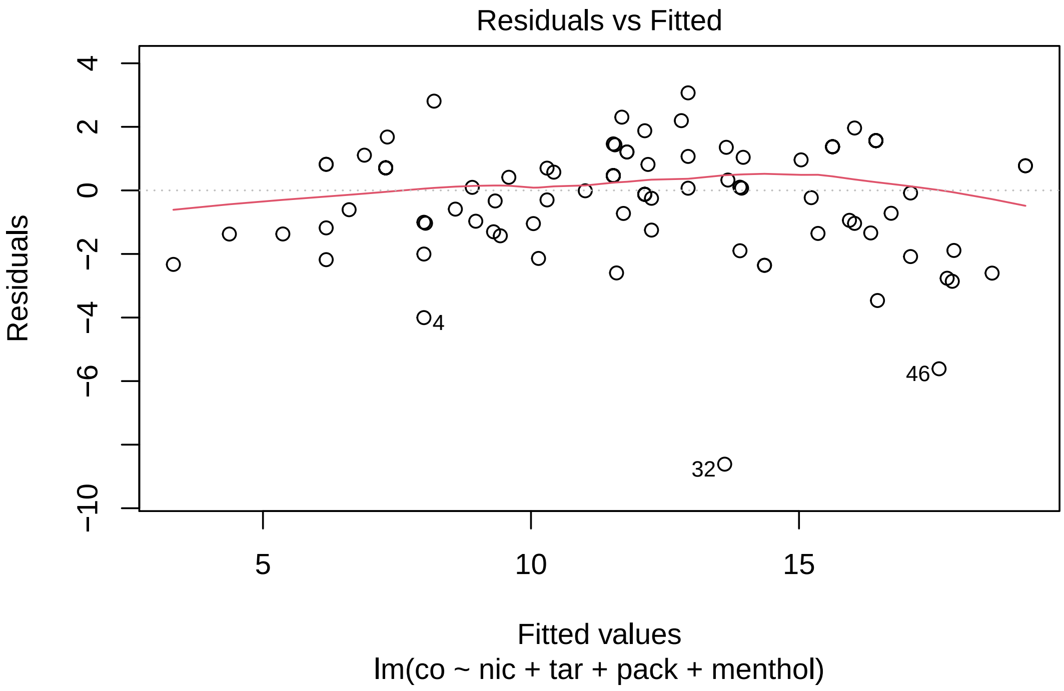
Problem 13

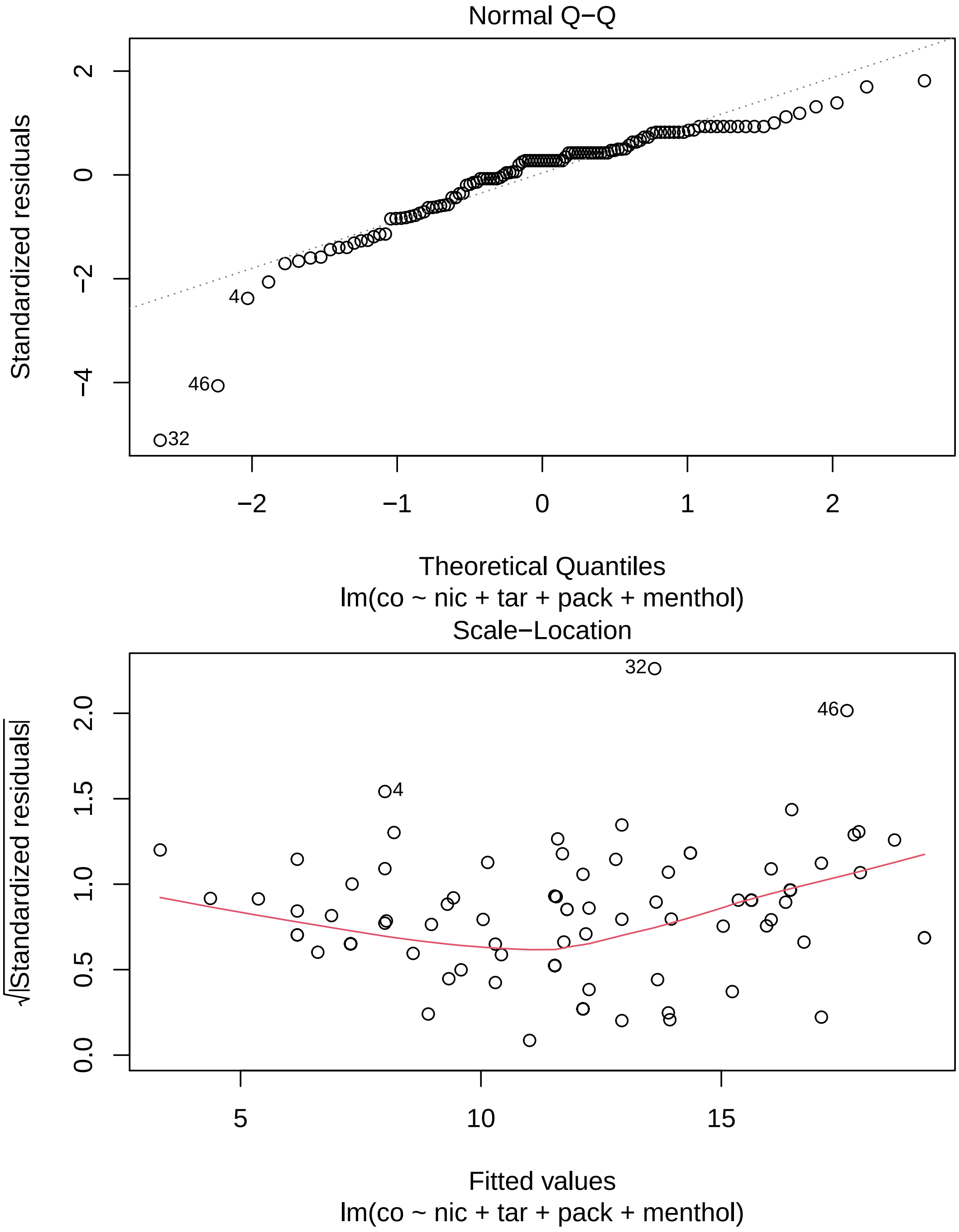
Model on all of the variables doesn't look too bad, except for one outlier. We build it again without the outlier, and it does not change the end model.

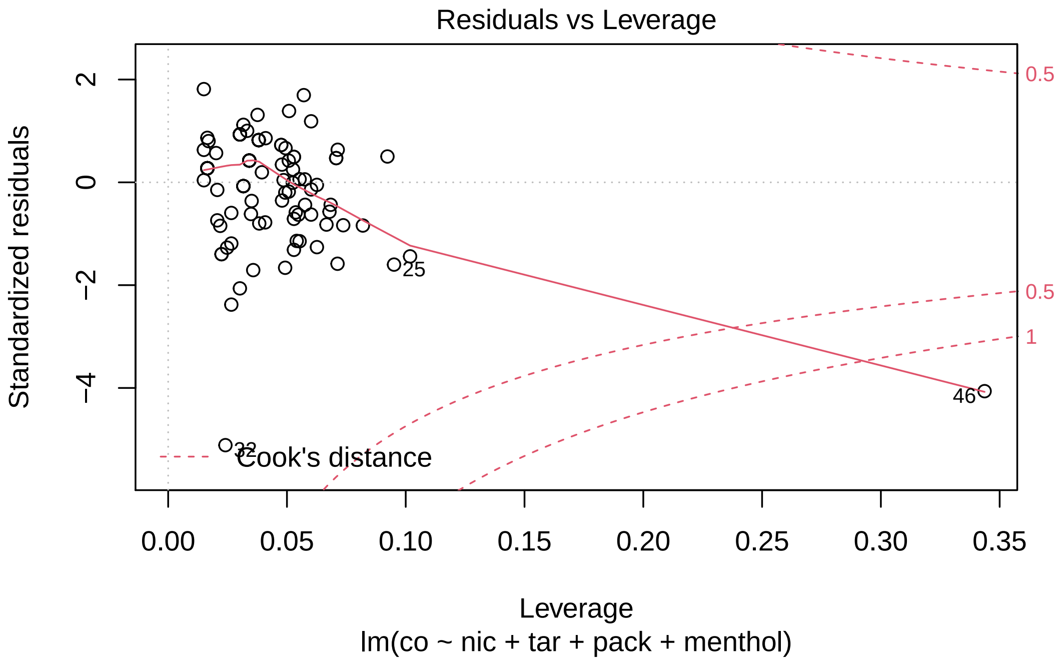
```
cigs <- fosdata::cigs_small
mod <- lm(co ~ nic + tar + pack + menthol, data = cigs)
summary(mod)
```

```
##
## Call:
## lm(formula = co ~ nic + tar + pack + menthol, data = cigs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6135 -0.9626  0.4642  1.0982  3.0690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6156     0.5975   4.377 2.69e-05 ***
## nic          -6.8253     1.4624  -4.667 8.45e-06 ***
## tar           1.3953     0.0997  13.994  < 2e-16 ***
## packSP        1.1405     0.3875   2.943  0.00395 **
## mentholyes   -0.8085     0.3477  -2.325  0.02184 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.705 on 113 degrees of freedom
## Multiple R-squared:  0.8306, Adjusted R-squared:  0.8246
## F-statistic: 138.5 on 4 and 113 DF,  p-value: < 2.2e-16
```

```
plot(mod)
```



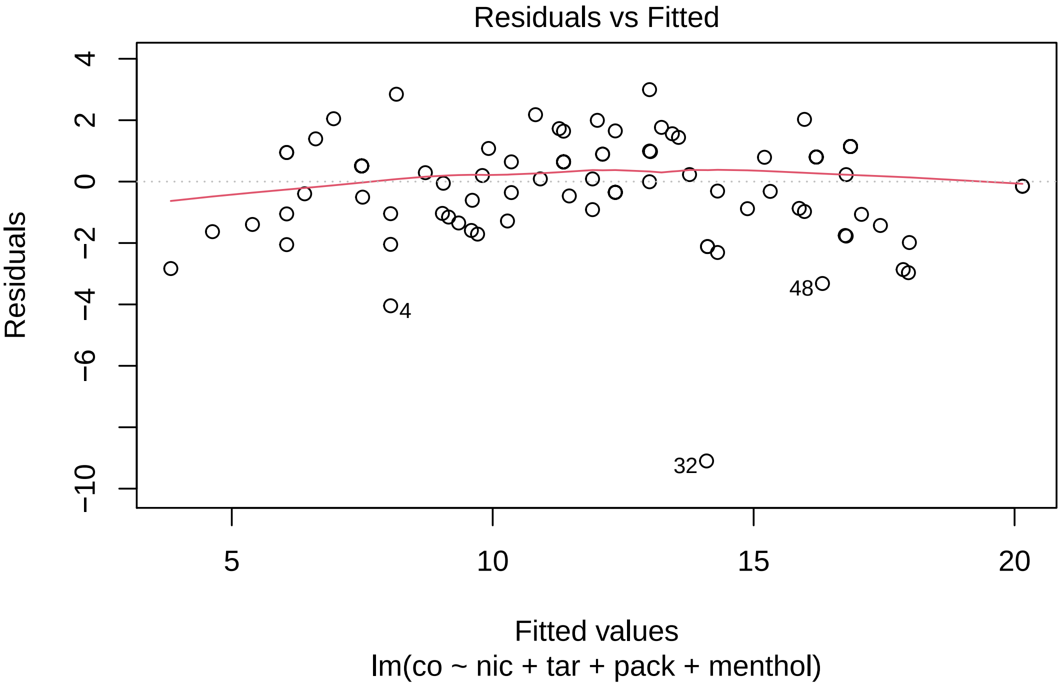

Normal Q−Q
Standardized residuals
2
0
−2
−4
4
46
32
−2
−1
0
1
2
Theoretical Quantiles
lm(co ~ nic + tar + pack + menthol)
Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
32
46
4
5
10
15
Fitted values
lm(co ~ nic + tar + pack + menthol)

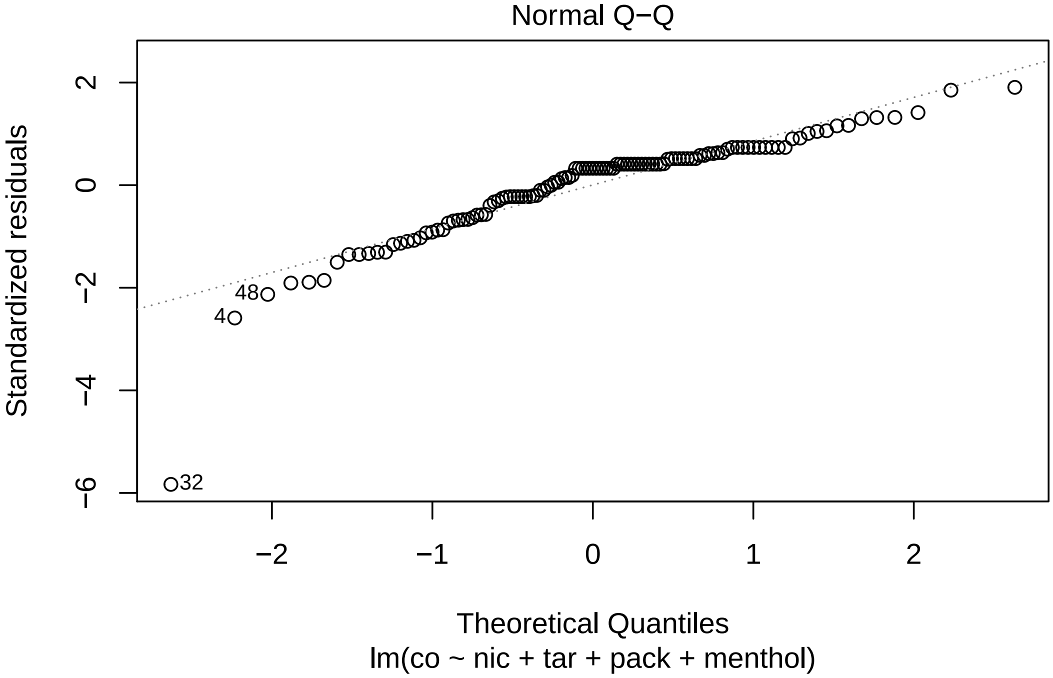


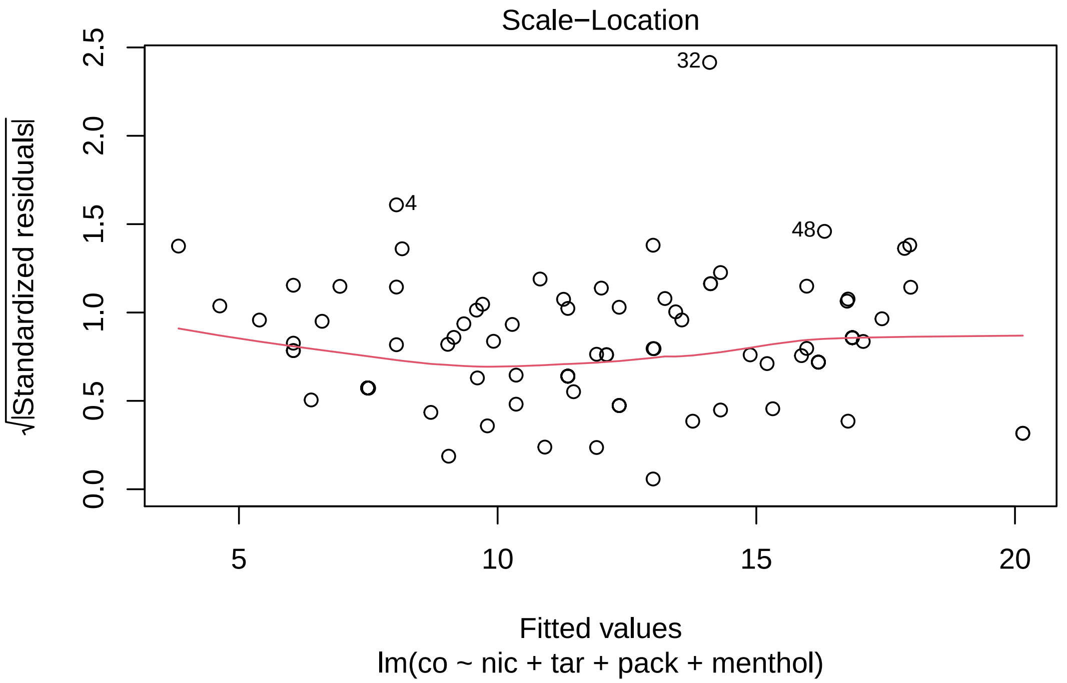
```
mod2 <- lm(co ~ nic + tar + pack + menthol, data = cigs[-46,])
summary(mod2)
```

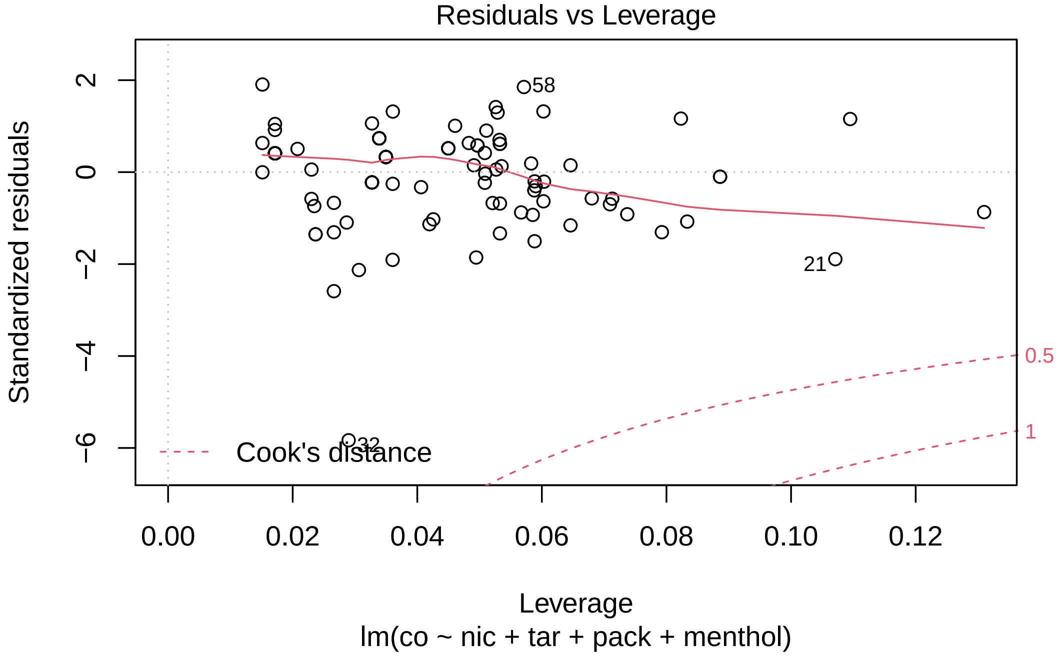
```
##
## Call:
## lm(formula = co ~ nic + tar + pack + menthol, data = cigs[-46,
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0978 -0.8734  0.5109  0.8943  2.9947
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.2763     0.5748   5.700 9.85e-08 ***
## nic         -10.9251     1.6492  -6.624 1.26e-09 ***
## tar           1.6477     0.1090  15.110  < 2e-16 ***
## packSP        0.8996     0.3639   2.472   0.0149 *
## mentholyes   -0.6560     0.3246  -2.021   0.0457 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.583 on 112 degrees of freedom
## Multiple R-squared:  0.8553, Adjusted R-squared:  0.8501
## F-statistic: 165.5 on 4 and 112 DF,  p-value: < 2.2e-16
```

```
plot(mod2)
```


Residuals vs Fitted
Residuals
4
2
0
−2
−4
−6
−10
5
10
15
20
4
48
32
Fitted values
lm(co ~ nic + tar + pack + menthol)


Normal Q−Q
Standardized residuals
2
0
−2
−4
−6
−2
−1
0
1
2
48
4
32
Theoretical Quantiles
lm(co ~ nic + tar + pack + menthol)





Problem 14

```
fish <- fosdata::fish
fish <- mutate(fish, species = factor(species, labels = c("Bream", "Whitefish", "Roach", "Parkki'
fish <- mutate(fish, sex = factor(sex, labels = c("female", "male")))
```

There are 87 missing values out of 159 for sex. We will build two models. One with sex and one without.

```
mod <- lm(weight ~ . - obs, data= fish)
summary(mod)
```

```
##
## Call:
## lm(formula = weight ~ . - obs, data = fish)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.885  -51.295    3.312   40.924  179.551
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1552.958    283.274  -5.482 9.56e-07 ***
## speciesWhitefish   318.357    148.613   2.142   0.0364 *
## speciesRoach       321.289    137.657   2.334   0.0231 *
## speciesParkki      232.010     89.058   2.605   0.0116 *
## speciesSmelt       816.873    191.054   4.276 7.20e-05 ***
## speciesPike         40.704    205.619   0.198   0.8438
## speciesPerch       339.866    162.635   2.090   0.0410 *
## length1            -63.078     47.607  -1.325   0.1904
## length2             42.598     55.572   0.767   0.4465
## length3             57.830     32.437   1.783   0.0798 .
## height_percent      12.319      8.006   1.539   0.1293
## width_percent       -2.781     10.061  -0.276   0.7832
## sexmale             45.009     26.532   1.696   0.0952 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.16 on 58 degrees of freedom
##   (88 observations deleted due to missingness)
## Multiple R-squared:  0.9729, Adjusted R-squared:  0.9673
## F-statistic: 173.4 on 12 and 58 DF,  p-value: < 2.2e-16
```

We start by removing the width and length variables that are not significant.

```
mod2 <- lm(weight ~ . - obs - width_percent - length2 - length1, data = fish)
summary(mod2)
```

```
##
## Call:
## lm(formula = weight ~ . - obs - width_percent - length2 - length1,
##     data = fish)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -168.928  -50.978    5.568   47.884  188.693
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1495.489    268.881  -5.562 6.28e-07 ***
## speciesWhitefish   253.354    111.789   2.266  0.02699 *
## speciesRoach       250.485    103.583   2.418  0.01860 *
## speciesParkki      159.669     47.468   3.364  0.00133 **
## speciesSmelt       727.259    158.315   4.594 2.23e-05 ***
## speciesPike        -10.781    184.708  -0.058  0.95365
## speciesPerch       257.373    100.567   2.559  0.01299 *
## length3             41.732      1.503  27.772  < 2e-16 ***
## height_percent      13.116      6.814   1.925  0.05892 .
## sexmale             38.989     25.672   1.519  0.13399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.54 on 61 degrees of freedom
##   (88 observations deleted due to missingness)
## Multiple R-squared:  0.9719, Adjusted R-squared:  0.9678
## F-statistic: 234.8 on 9 and 61 DF,  p-value: < 2.2e-16
```

```
anova(mod2, mod)
```

```
## Analysis of Variance Table
##
## Model 1: weight ~ (obs + species + length1 + length2 + length3 + height_percent +
##     width_percent + sex) - obs - width_percent - length2 - length1
## Model 2: weight ~ (obs + species + length1 + length2 + length3 + height_percent +
##     width_percent + sex) - obs
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     61 338956
## 2     58 327610  3     11346 0.6696 0.5742
```

Now, we remove sex, but since sex isn't important, we redo the entire thing using all of the data.Note the increase in the degrees of freedom of the F statistic.

```
mod <- lm(weight ~ species + length1 + length2 + length3 + height_percent + width_percent, data =
summary(mod)
```

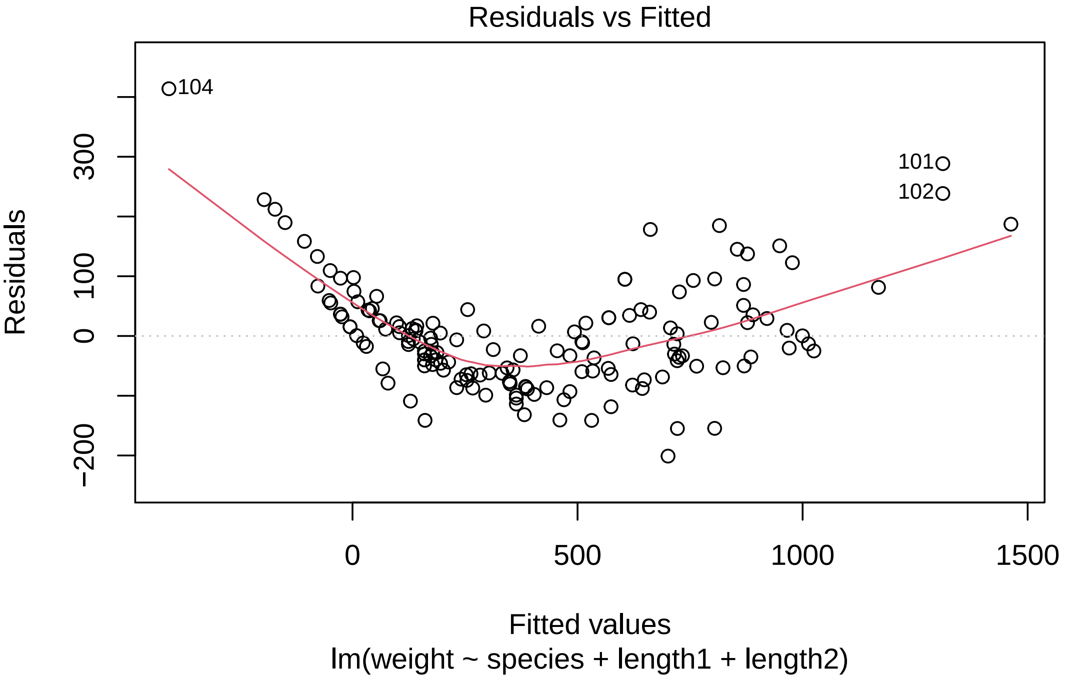
```
##
## Call:
## lm(formula = weight ~ species + length1 + length2 + length3 +
##     height_percent + width_percent, data = fish)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -186.12  -57.05  -14.46   37.86  412.29
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1130.347    209.690  -5.391 2.76e-07 ***
## speciesWhitefish    99.787    100.831   0.990 0.323984
## speciesRoach       112.210     97.841   1.147 0.253317
## speciesParkki      135.807     69.902   1.943 0.053963 .
## speciesSmelt       512.020    144.834   3.535 0.000546 ***
## speciesPike       -128.026    153.487  -0.834 0.405579
## speciesPerch       149.673    127.973   1.170 0.244082
## length1            -65.276     35.766  -1.825 0.070032 .
## length2             63.253     44.872   1.410 0.160771
## length3             35.476     27.827   1.275 0.204380
## height_percent       4.533      5.777   0.785 0.434005
## width_percent        6.972      8.304   0.840 0.402460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91.62 on 146 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9395, Adjusted R-squared:  0.9349
## F-statistic: 205.9 on 11 and 146 DF,  p-value: < 2.2e-16
```

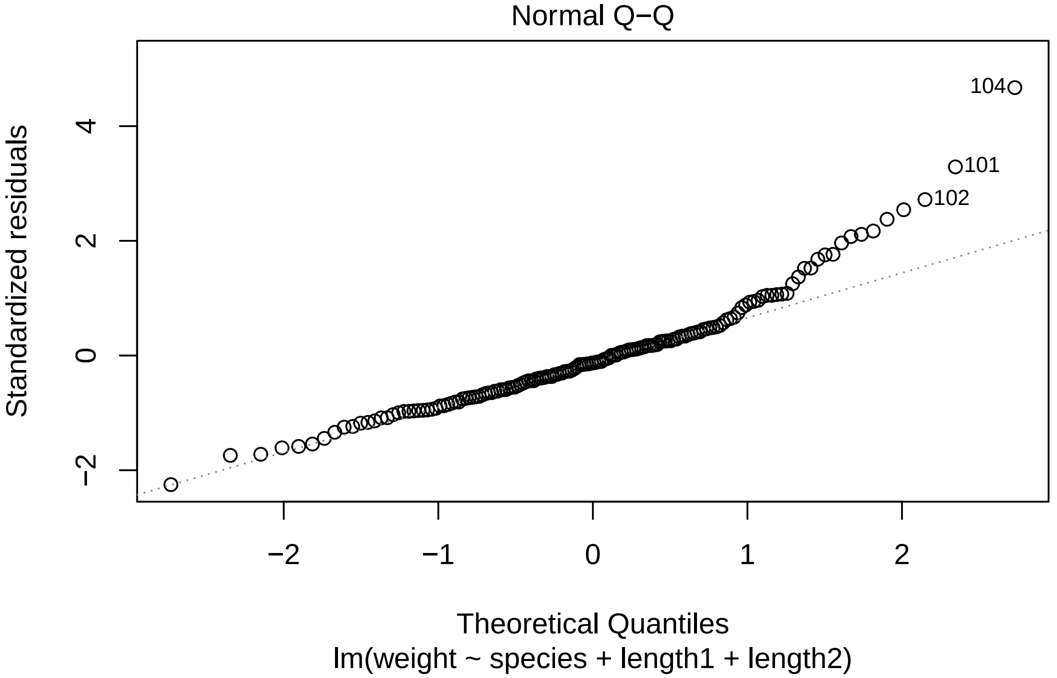
Now, length1 and length2 are both significant.

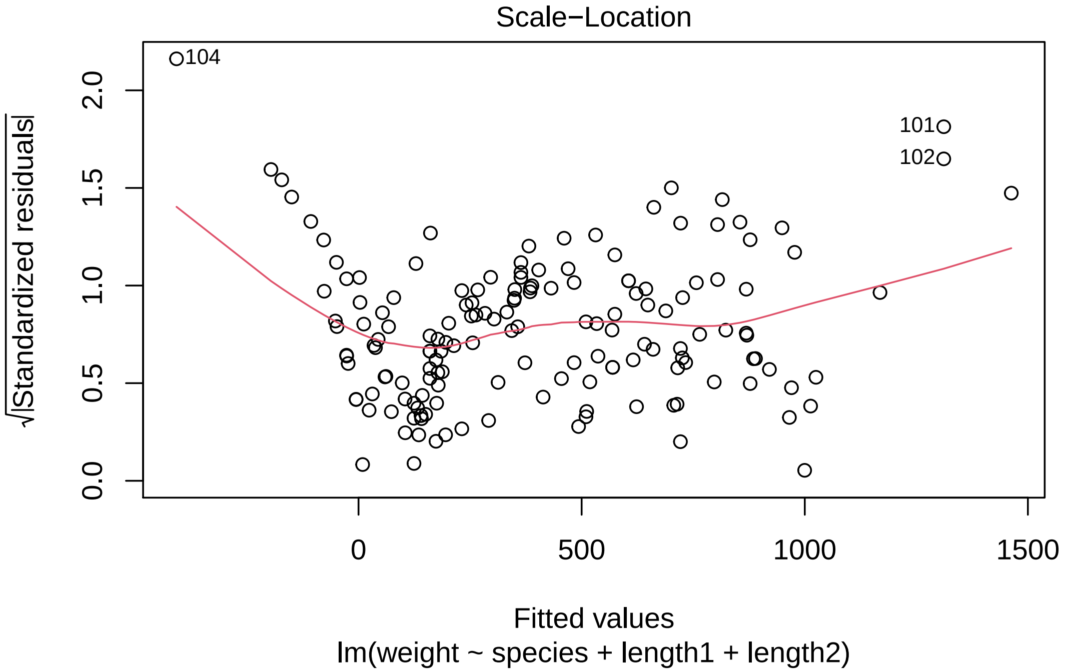
```
mod2 <-  lm(weight ~ species + length1 + length2, data = fish)
summary(mod2)
```

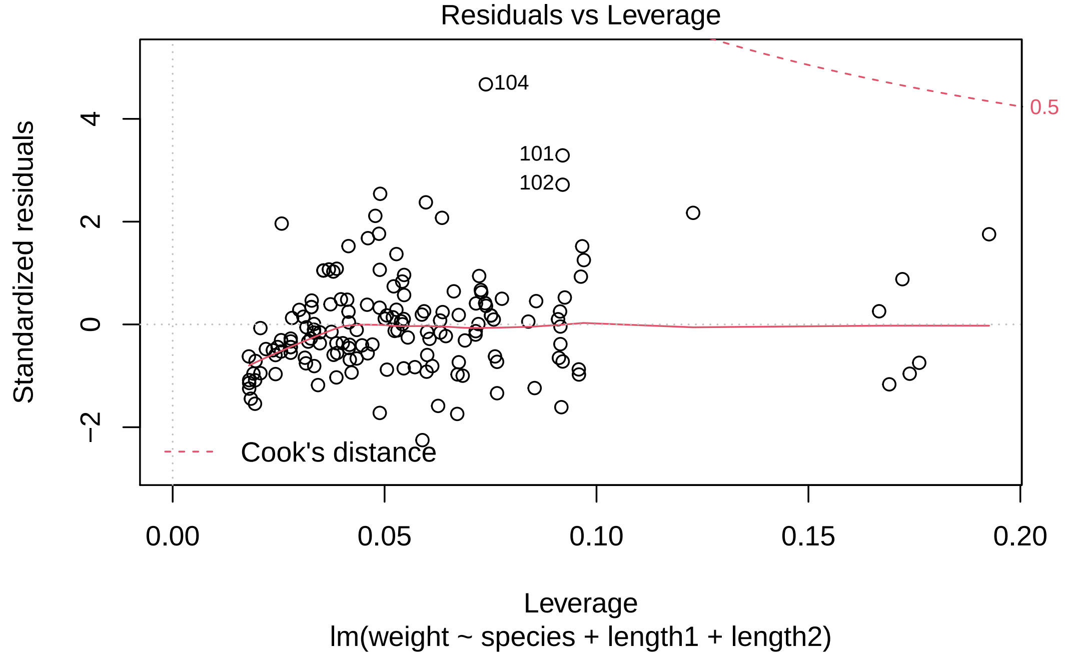
```
##
## Call:
## lm(formula = weight ~ species + length1 + length2, data = fish)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -201.04  -58.32  -11.45   36.03  413.83
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -767.767     50.563 -15.184  < 2e-16 ***
## speciesWhitefish   -7.794     41.357  -0.188  0.85077
## speciesRoach        2.433     34.590   0.070  0.94402
## speciesParkki      75.009     38.358   1.956  0.05239 .
## speciesSmelt      303.503     49.489   6.133  7.4e-09 ***
## speciesPike      -362.627     35.691 -10.160  < 2e-16 ***
## speciesPerch       -7.411     24.243  -0.306  0.76027
## length1           -74.661     35.493  -2.104  0.03710 *
## length2           110.382     33.450   3.300  0.00121 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.05 on 149 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9376, Adjusted R-squared:  0.9343
## F-statistic:   280 on 8 and 149 DF,  p-value: < 2.2e-16
```

```
plot(mod2)
```


Residuals vs Fitted
104
101
102
Residuals
−200
0
100
300
0
500
1000
1500
Fitted values
lm(weight ~ species + length1 + length2)


Normal Q−Q
104
101
102
Standardized residuals
−2
0
2
4
−2
−1
0
1
2
Theoretical Quantiles
lm(weight ~ species + length1 + length2)





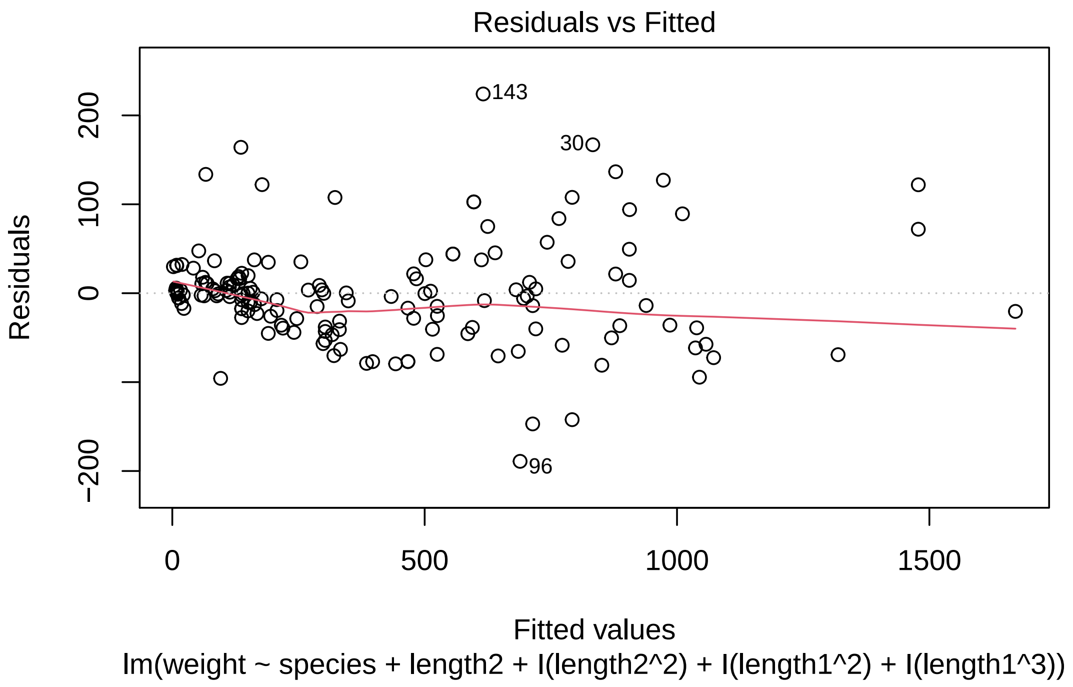
The final model depends on species, length1 and length2. However, we have significant curvature in the residuals! This leads me to believe that there should be some higher order terms. We add length1 and length2 squared and cubed, and then remove the ones that aren't significant.

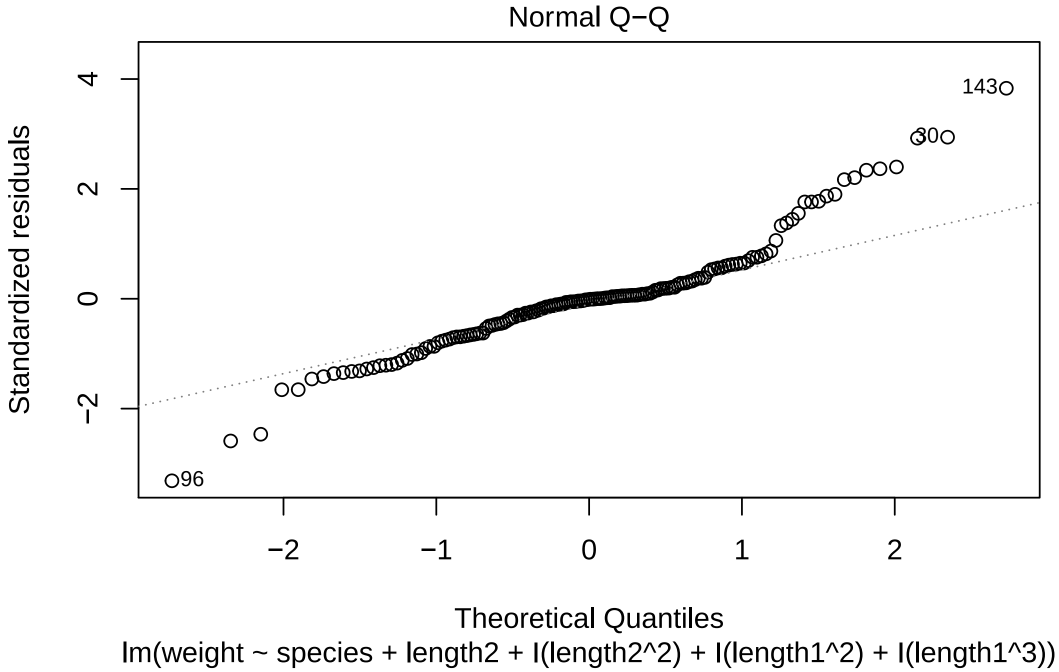
```
mod3 <- lm(weight ~ species + length2 + I(length2^2) + I(length1^2) + I(length1^3), data = fish)
summary(mod3)
```

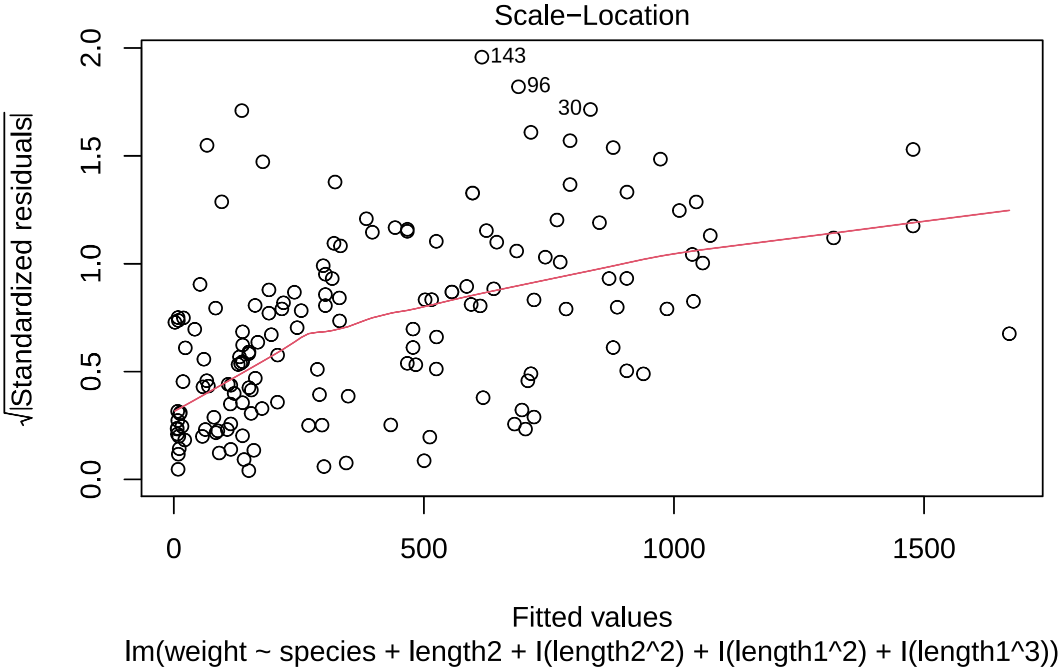
```
##
## Call:
```

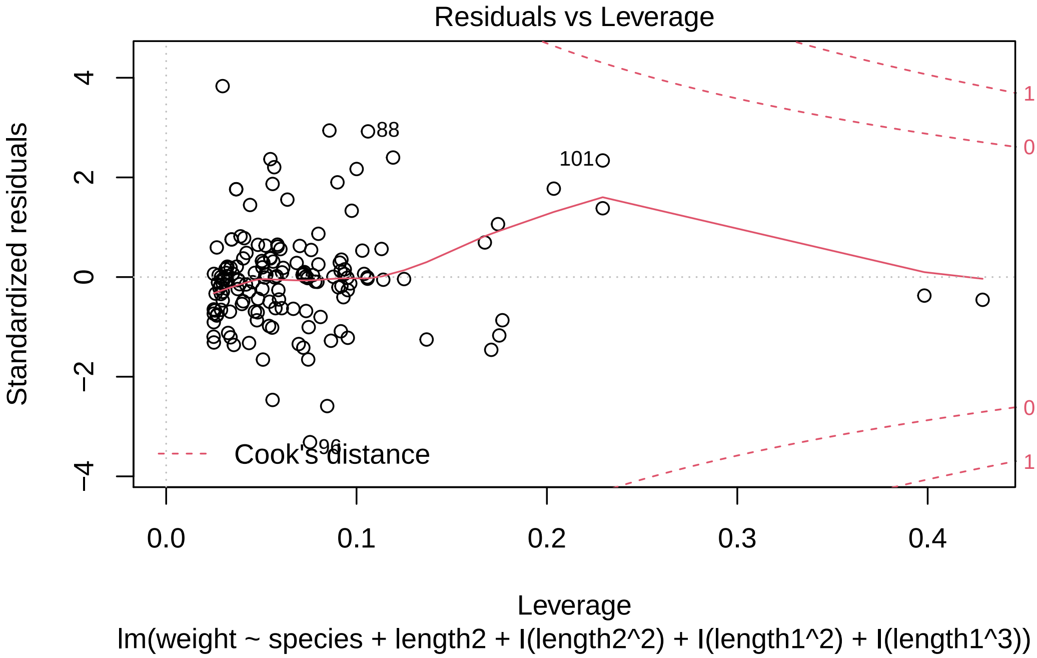
```
## lm(formula = weight ~ species + length2 + I(length2^2) + I(length1^2) +
##     I(length1^3), data = fish)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -189.102  -30.713   -0.669   18.472  224.067
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.152e+02  9.458e+01   3.332 0.001090 **
## speciesWhitefish -8.120e+00  2.702e+01  -0.300 0.764230
## speciesRoach     -5.562e+01  2.321e+01  -2.397 0.017803 *
## speciesParkki    -1.015e+01  2.607e+01  -0.389 0.697503
## speciesSmelt     -4.756e+01  3.723e+01  -1.277 0.203532
## speciesPike      -4.855e+02  2.750e+01 -17.655  < 2e-16 ***
## speciesPerch     -6.033e+01  1.716e+01  -3.515 0.000584 ***
## length2          -4.388e+01  8.438e+00  -5.200 6.58e-07 ***
## I(length2^2)      2.155e+00  3.794e-01   5.682 6.93e-08 ***
## I(length1^2)     -1.436e-01  4.015e-01  -0.358 0.721003
## I(length1^3)     -1.724e-02  2.758e-03  -6.251 4.19e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.36 on 147 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9744, Adjusted R-squared:  0.9727
## F-statistic: 559.9 on 10 and 147 DF,  p-value: < 2.2e-16
```

```
plot(mod3)
```




Normal Q−Q
Standardized residuals
4
2
0
−2
143
30
96
−2
−1
0
1
2
Theoretical Quantiles
lm(weight ~ species + length2 + I(length2^2) + I(length1^2) + I(length1^3))


Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
143
96
30
0
500
1000
1500
Fitted values
lm(weight ~ species + length2 + I(length2^2) + I(length1^2) + I(length1^3))



This is definitely better. The R-squared value is much higher and the residual plots look better as well.

Problem 15

```
tlc <- ISwR::tlc
mod <- lm(tlc ~ ., data = tlc)
summary(mod)
```

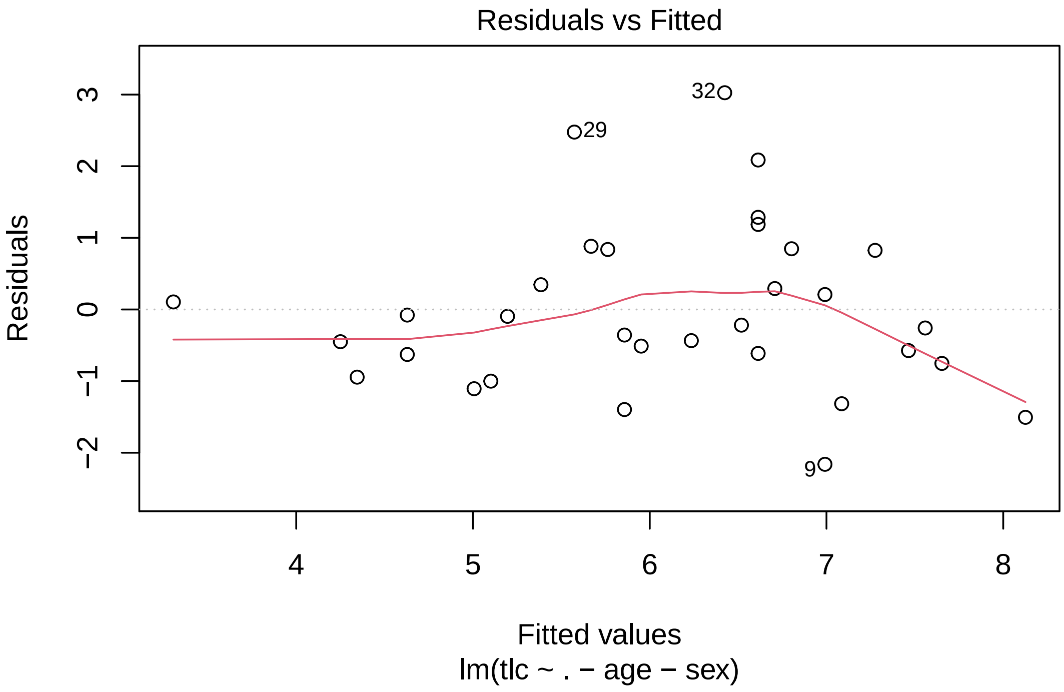
```
##
## Call:
## lm(formula = tlc ~ ., data = tlc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9572 -0.8404  0.0445  0.5793  2.6097
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.24066    3.44489  -2.682  0.01212 *
## age         -0.02502    0.02353  -1.063  0.29665
## sex          0.69705    0.49944   1.396  0.17379
## height       0.08955    0.02455   3.647  0.00107 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.156 on 28 degrees of freedom
## Multiple R-squared:  0.5422, Adjusted R-squared:  0.4932
## F-statistic: 11.05 on 3 and 28 DF,  p-value: 5.822e-05
```

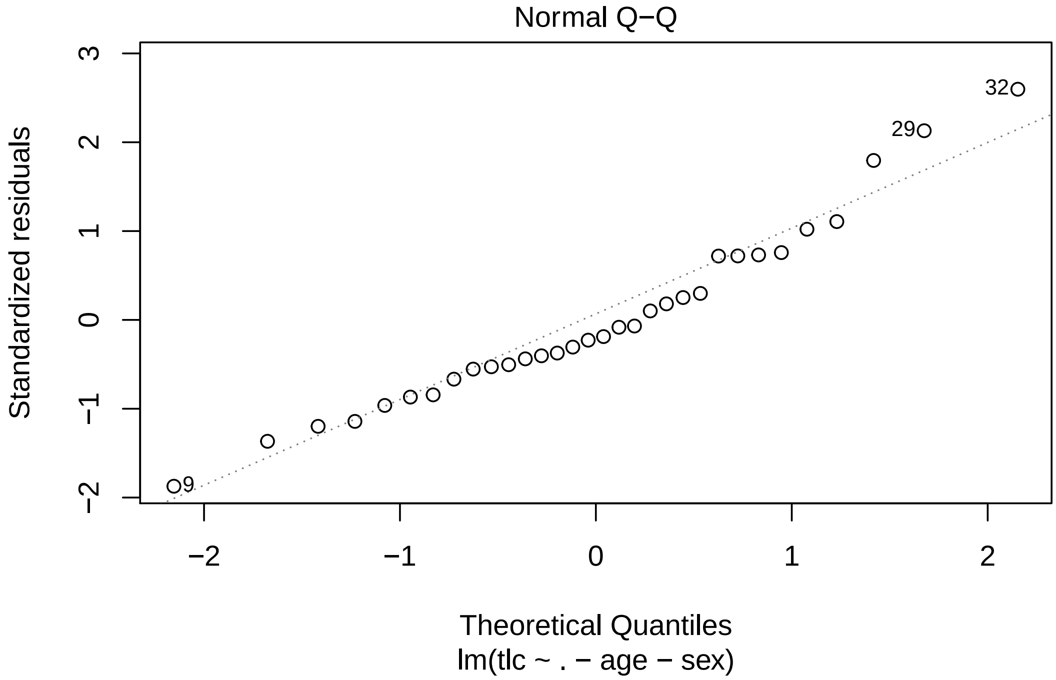
```
mod2 <- lm(tlc ~ . - age - sex, data= tlc)
summary(mod2)
```

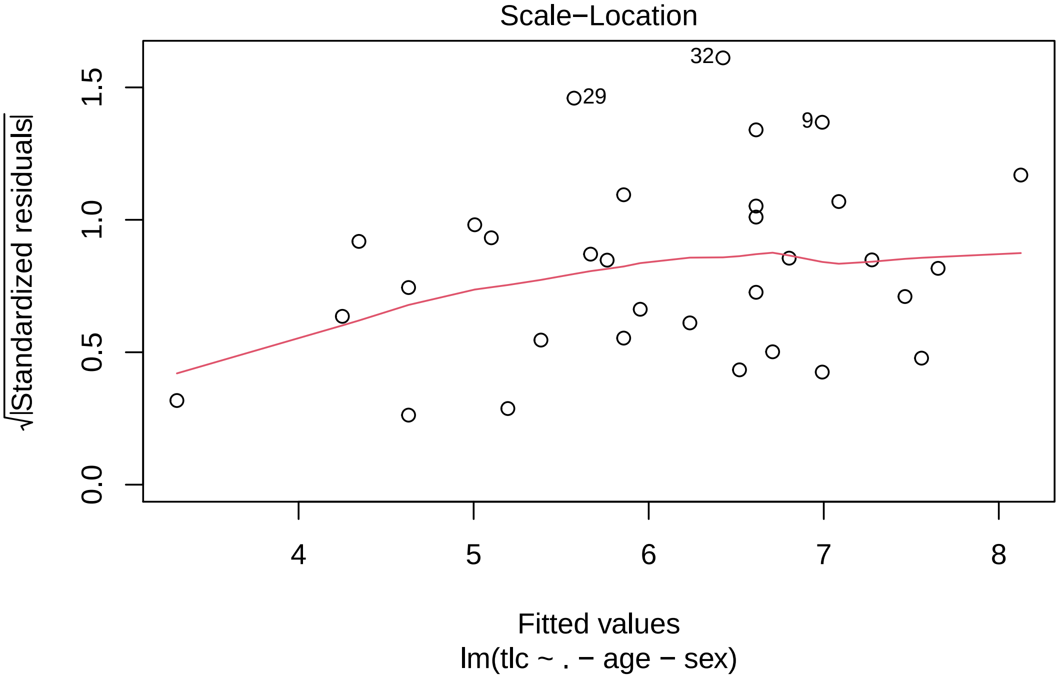
```
##
## Call:
## lm(formula = tlc ~ . - age - sex, data = tlc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1614 -0.6594 -0.2387  0.8281  3.0257
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.74025    2.99108  -3.256   0.0028 **
## height       0.09453    0.01782   5.305 9.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.186 on 30 degrees of freedom
## Multiple R-squared:  0.484,  Adjusted R-squared:  0.4668
## F-statistic: 28.14 on 1 and 30 DF,  p-value: 9.853e-06
```

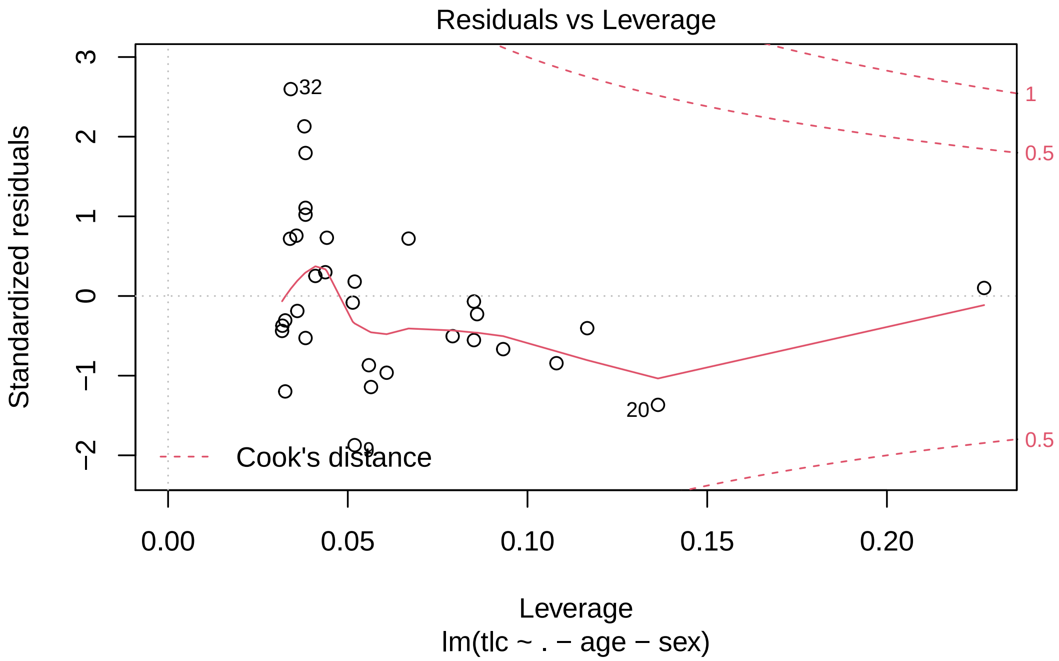
We can model `tlc` on `height`. The residuals look pretty good.

```
plot(mod2)
```




Normal Q−Q
Standardized residuals
−2
−1
0
1
2
3
Theoretical Quantiles
lm(tlc ~ . − age − sex)
32
29
9


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
4
5
6
7
8
Fitted values
lm(tlc ~ . − age − sex)
32
29
9



Problem 16

```
cystfibr <- ISwR::cystfibr
str(cystfibr)
```

```
## 'data.frame':    25 obs. of  10 variables:
##  $ age   : int  7 7 8 8 8 9 11 12 12 13 ...
##  $ sex   : int  0 1 0 1 0 0 1 1 0 1 ...
##  $ height: int  109 112 124 125 127 130 139 150 146 155 ...
##  $ weight: num  13.1 12.9 14.1 16.2 21.5 17.5 30.7 28.4 25.1 31.5 ...
##  $ bmp   : int  68 65 64 67 93 68 89 69 67 68 ...
##  $ fev1  : int  32 19 22 41 52 44 28 18 24 23 ...
##  $ rv    : int  258 449 441 234 202 308 305 369 312 413 ...
##  $ frc   : int  183 245 268 146 131 155 179 198 194 225 ...
##  $ tlc   : int  137 134 147 124 104 118 119 103 128 136 ...
##  $ pemax : int  95 85 100 85 95 80 65 110 70 95 ...
```

```
mod <- lm(pemax ~ ., data = cystfibr)
summary(mod)
```

```
##
## Call:
## lm(formula = pemax ~ ., data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.338 -11.532   1.081  13.386  33.405
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.0582   225.8912   0.779    0.448
## age          -2.5420     4.8017  -0.529    0.604
## sex          -3.7368    15.4598  -0.242    0.812
## height       -0.4463     0.9034  -0.494    0.628
## weight        2.9928     2.0080   1.490    0.157
## bmp          -1.7449     1.1552  -1.510    0.152
## fev1          1.0807     1.0809   1.000    0.333
## rv            0.1970     0.1962   1.004    0.331
## frc          -0.3084     0.4924  -0.626    0.540
## tlc           0.1886     0.4997   0.377    0.711
##
## Residual standard error: 25.47 on 15 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.4197
## F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

We start by removing the other lung function data. None of the others become significant as we remove them one at a time.

```
mod2 <- lm(pemax ~ . - tlc - frc - rv - fev1, data = cystfibr)
anova(mod2, mod)
```

```
## Analysis of Variance Table
##
## Model 1: pemax ~ (age + sex + height + weight + bmp + fev1 + rv + frc +
##     tlc) - tlc - frc - rv - fev1
## Model 2: pemax ~ age + sex + height + weight + bmp + fev1 + rv + frc +
##     tlc
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     19 12129.2
## 2     15  9731.2  4      2398 0.9241 0.4758
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = pemax ~ . - tlc - frc - rv - fev1, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.194  -9.412  -2.425   9.157  40.112
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 280.4482   124.9556   2.244   0.0369 *
## age          -3.0750     3.6352  -0.846   0.4081
## sex         -11.5281    10.3720  -1.111   0.2802
## height       -0.6853     0.7962  -0.861   0.4001
## weight        3.5546     1.5281   2.326   0.0312 *
## bmp          -1.9613     0.9263  -2.117   0.0476 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.27 on 19 degrees of freedom
## Multiple R-squared:  0.548,  Adjusted R-squared:  0.429
## F-statistic: 4.606 on 5 and 19 DF,  p-value: 0.006388
```

Next we remove age, height and sex in that order.

```
mod3 <- lm(pemax ~ bmp + weight, data = cystfibr)
summary(mod3)
```

```
##
## Call:
## lm(formula = pemax ~ bmp + weight, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.924 -13.399   4.361  16.642  48.404
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 124.8297    37.4786   3.331 0.003033 **
## bmp          -1.0054     0.5814  -1.729 0.097797 .
## weight        1.6403     0.3900   4.206 0.000365 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.31 on 22 degrees of freedom
## Multiple R-squared:  0.4749, Adjusted R-squared:  0.4271
## F-statistic: 9.947 on 2 and 22 DF,  p-value: 0.0008374
```

```
anova(mod3, mod2)
```

```
## Analysis of Variance Table
##
## Model 1: pemax ~ bmp + weight
## Model 2: pemax ~ (age + sex + height + weight + bmp + fev1 + rv + frc +
##     tlc) - tlc - frc - rv - fev1
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     22 14090
## 2     19 12129  3    1961.3 1.0241 0.4041
```

```
anova(mod3, mod)
```

```
## Analysis of Variance Table
##
## Model 1: pemax ~ bmp + weight
## Model 2: pemax ~ age + sex + height + weight + bmp + fev1 + rv + frc +
##     tlc
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     22 14090.5
## 2     15  9731.2  7    4359.3 0.9599 0.4928
```

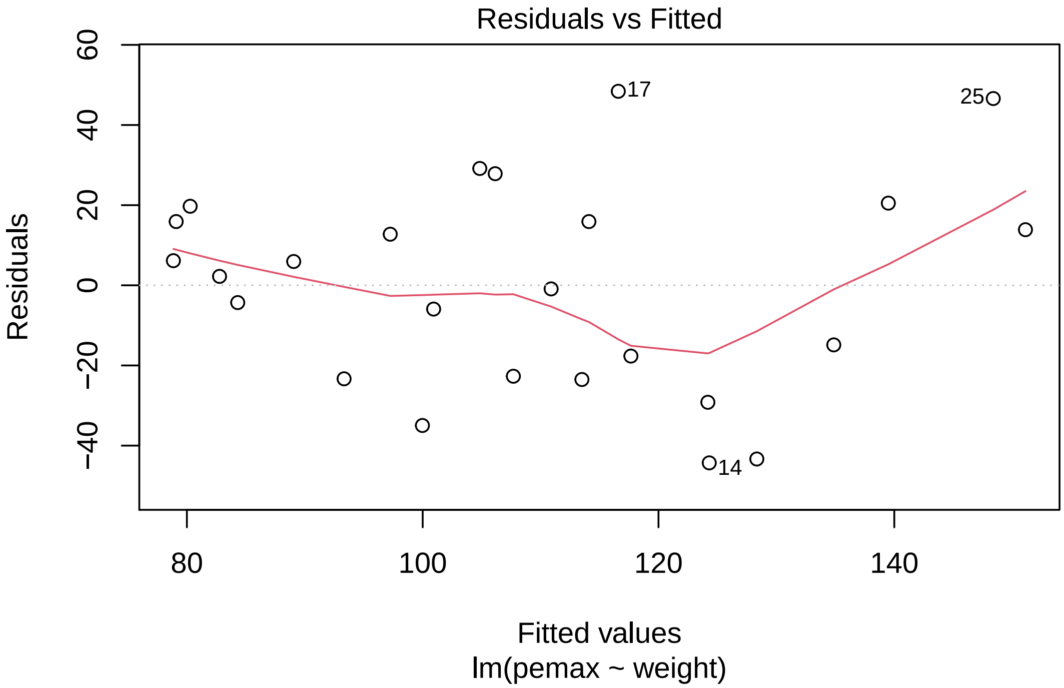
Finally, we remove bmp.

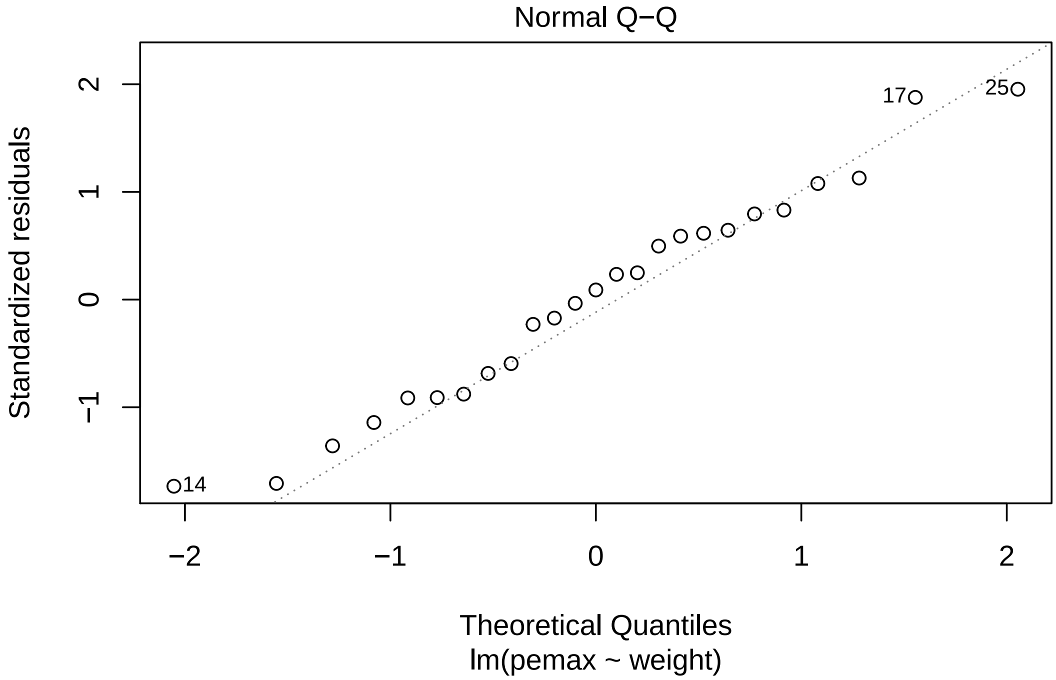
```
mod4 <- lm(pemax ~ weight, data = cystfibr)
summary(mod4)
```

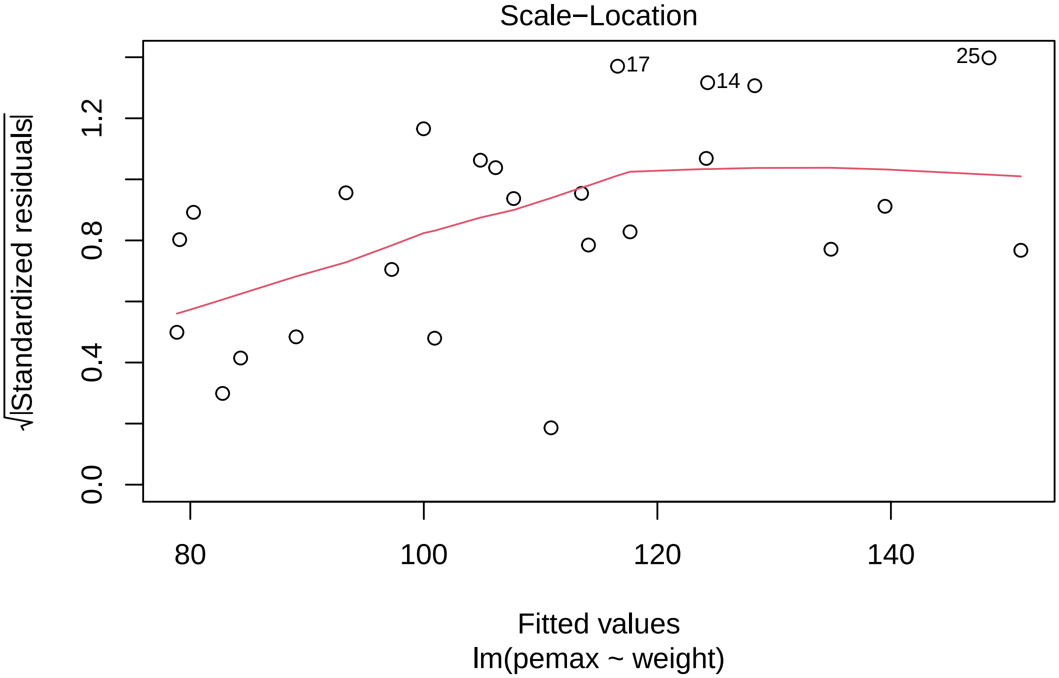
```
##
## Call:
## lm(formula = pemax ~ weight, data = cystfibr)
##
```

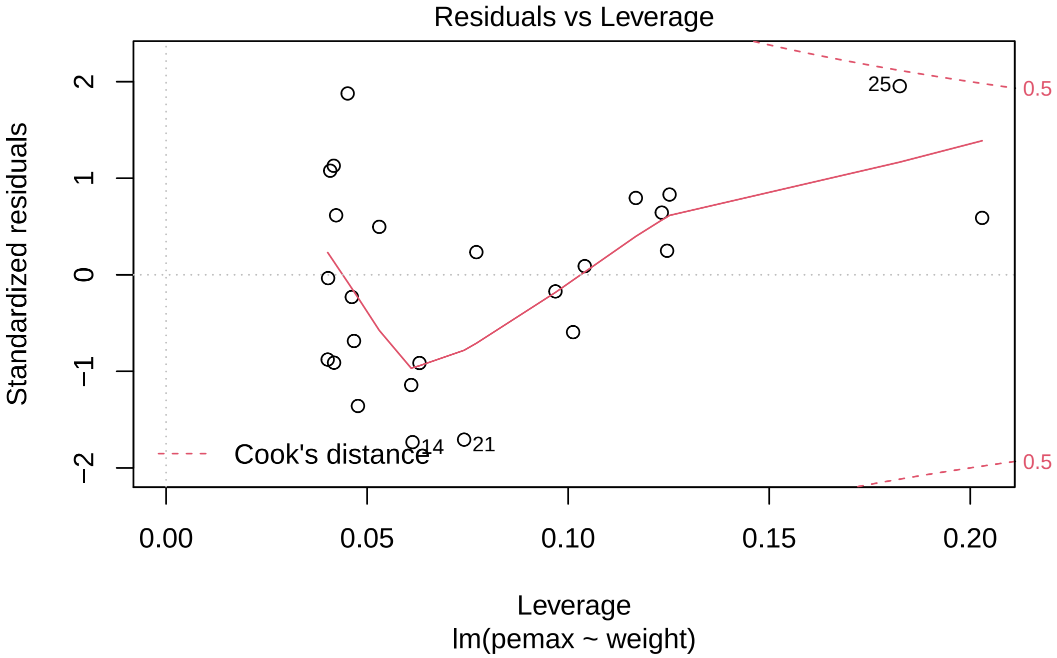
```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.30 -22.69   2.23  15.91  48.41
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  63.5456    12.7016   5.003 4.63e-05 ***
## weight        1.1867     0.3009   3.944 0.000646 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.38 on 23 degrees of freedom
## Multiple R-squared:  0.4035, Adjusted R-squared:  0.3776
## F-statistic: 15.56 on 1 and 23 DF,  p-value: 0.0006457
```

```
plot(mod4)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
17
25
14
−2
−1
0
1
2
Theoretical Quantiles
lm(pemax ~ weight)


Scale−Location
√|Standardized residuals|
1.2
0.8
0.4
0.0
17
14
25
80
100
120
140
Fitted values
lm(pemax ~ weight)



Residuals look pretty good.

Problem 17

a. It is very likely that if the forecast is wrong for one of the stations, that it will also be wrong for other nearby stations. Therefore, the residuals of the model are unlikely to be independent. It is also possible that if the forecast is wrong on one day, it may again be wrong on the next day, but that is less obvious.

b.

```
seoul <- fosdata::seoulweather %>%
  filter(station == 1)
seoul <- seoul %>%
  mutate(error_tmax = ldaps_tmax_lapse - next_tmax) %>%
  select(-next_tmax, -next_tmin, -date, -lat, -lon, -dem, -slope, -station)
```

c. We use `MASS::stepAIC`, as it tends to do better for predicting future values than an ad hoc variable selection.

```
mod <- lm(error_tmax ~ . , data = seoul)
summary(mod)
```

```
##
## Call:
## lm(formula = error_tmax ~ ., data = seoul)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1051 -0.7283 -0.0339  0.7074  4.0881
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -3.6170733  2.1941062  -1.649  0.10034
## present_tmax     -0.1296248  0.0466110  -2.781  0.00578 **
## present_tmin      0.1074699  0.0723564   1.485  0.13857
## ldaps_r_hmin     -0.0135397  0.0131679  -1.028  0.30471
## ldaps_r_hmax     -0.0017452  0.0179719  -0.097  0.92271
## ldaps_tmax_lapse  0.3676989  0.0845830   4.347 1.92e-05 ***
## ldaps_tmin_lapse -0.2056147  0.1036328  -1.984  0.04821 *
## ldaps_ws          0.1078292  0.0345181   3.124  0.00197 **
## ldaps_lh         -0.0069830  0.0049759  -1.403  0.16159
## ldaps_cc1         1.1051601  0.5579116   1.981  0.04857 *
## ldaps_cc2         0.6173516  0.7123480   0.867  0.38687
## ldaps_cc3        -0.0999945  0.7163087  -0.140  0.88908
## ldaps_cc4         1.5103592  0.5212935   2.897  0.00406 **
## ldaps_ppt1       -0.0313750  0.0457898  -0.685  0.49378
## ldaps_ppt2       -0.1252231  0.0561651  -2.230  0.02656 *
## ldaps_ppt3        0.0162914  0.0654041   0.249  0.80347
## ldaps_ppt4       -0.0179376  0.0950707  -0.189  0.85048
## solar_radiation  -0.0001477  0.0002089  -0.707  0.48014
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.323 on 285 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.1812, Adjusted R-squared:  0.1323
## F-statistic: 3.709 on 17 and 285 DF,  p-value: 1.903e-06
```

```
mod <- MASS::stepAIC(mod)
```

```
## Start:  AIC=187.03
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_r_hmax +
##     ldaps_tmax_lapse + ldaps_tmin_lapse + ldaps_ws + ldaps_lh +
##     ldaps_cc1 + ldaps_cc2 + ldaps_cc3 + ldaps_cc4 + ldaps_ppt1 +
##     ldaps_ppt2 + ldaps_ppt3 + ldaps_ppt4 + solar_radiation
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_r_hmax      1     0.017 498.81 185.04
## - ldaps_cc3         1     0.034 498.83 185.06
## - ldaps_ppt4        1     0.062 498.86 185.07
## - ldaps_ppt3        1     0.109 498.90 185.10
## - ldaps_ppt1        1     0.822 499.62 185.53
## - solar_radiation   1     0.875 499.67 185.56
## - ldaps_cc2         1     1.314 500.11 185.83
## - ldaps_r_hmin      1     1.850 500.65 186.16
## <none>                          498.80 187.03
## - ldaps_lh          1     3.447 502.24 187.12
## - present_tmin      1     3.861 502.66 187.37
## - ldaps_cc1         1     6.867 505.66 189.18
## - ldaps_tmin_lapse  1     6.890 505.68 189.19
## - ldaps_ppt2        1     8.700 507.49 190.27
## - present_tmax      1    13.536 512.33 193.15
## - ldaps_cc4         1    14.692 513.49 193.83
## - ldaps_ws          1    17.079 515.87 195.24
## - ldaps_tmax_lapse  1    33.075 531.87 204.49
##
```

```
## Step:  AIC=185.04
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc2 +
##     ldaps_cc3 + ldaps_cc4 + ldaps_ppt1 + ldaps_ppt2 + ldaps_ppt3 +
##     ldaps_ppt4 + solar_radiation
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_cc3         1     0.028 498.84 183.06
## - ldaps_ppt4        1     0.074 498.89 183.09
## - ldaps_ppt3        1     0.103 498.91 183.11
## - ldaps_ppt1        1     0.828 499.64 183.55
## - solar_radiation   1     0.919 499.73 183.60
## - ldaps_cc2         1     1.329 500.14 183.85
## - ldaps_r_hmin      1     2.006 500.82 184.26
## <none>                          498.81 185.04
## - ldaps_lh          1     3.486 502.30 185.16
## - present_tmin      1     3.918 502.73 185.41
## - ldaps_tmin_lapse  1     6.885 505.70 187.20
## - ldaps_cc1         1     6.915 505.73 187.22
## - ldaps_ppt2        1     8.689 507.50 188.28
## - present_tmax      1    13.895 512.71 191.37
## - ldaps_cc4         1    14.904 513.72 191.97
## - ldaps_ws          1    17.063 515.87 193.24
## - ldaps_tmax_lapse  1    33.216 532.03 202.58
##
## Step:  AIC=183.06
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc2 +
##     ldaps_cc4 + ldaps_ppt1 + ldaps_ppt2 + ldaps_ppt3 + ldaps_ppt4 +
##     solar_radiation
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_ppt4        1     0.081 498.92 181.11
## - ldaps_ppt3        1     0.088 498.93 181.11
## - ldaps_ppt1        1     0.827 499.67 181.56
## - solar_radiation   1     0.909 499.75 181.61
## - ldaps_cc2         1     1.322 500.16 181.86
## - ldaps_r_hmin      1     2.222 501.06 182.41
## <none>                          498.84 183.06
## - ldaps_lh          1     3.459 502.30 183.16
## - present_tmin      1     3.969 502.81 183.46
## - ldaps_tmin_lapse  1     6.866 505.71 185.20
## - ldaps_cc1         1     6.914 505.75 185.23
## - ldaps_ppt2        1     8.663 507.50 186.28
## - present_tmax      1    15.117 513.96 190.11
## - ldaps_ws          1    17.474 516.31 191.49
## - ldaps_cc4         1    22.518 521.36 194.44
## - ldaps_tmax_lapse  1    33.669 532.51 200.85
##
## Step:  AIC=181.11
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc2 +
##     ldaps_cc4 + ldaps_ppt1 + ldaps_ppt2 + ldaps_ppt3 + solar_radiation
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_ppt3        1     0.057 498.98 179.15
## - ldaps_ppt1        1     0.809 499.73 179.60
## - solar_radiation   1     0.924 499.84 179.67
## - ldaps_cc2         1     1.301 500.22 179.90
## - ldaps_r_hmin      1     2.298 501.22 180.50
## <none>                          498.92 181.11
## - ldaps_lh          1     3.521 502.44 181.24
## - present_tmin      1     3.932 502.85 181.49
## - ldaps_tmin_lapse  1     6.786 505.71 183.20
## - ldaps_cc1         1     7.087 506.01 183.38
## - ldaps_ppt2        1     8.658 507.58 184.32
## - present_tmax      1    15.037 513.96 188.11
## - ldaps_ws          1    17.456 516.38 189.53
## - ldaps_cc4         1    24.301 523.22 193.52
## - ldaps_tmax_lapse  1    33.635 532.55 198.88
##
## Step:  AIC=179.14
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc2 +
##     ldaps_cc4 + ldaps_ppt1 + ldaps_ppt2 + solar_radiation
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_ppt1        1     0.848 499.82 177.66
## - solar_radiation   1     0.986 499.96 177.74
## - ldaps_cc2         1     1.385 500.36 177.98
## - ldaps_r_hmin      1     2.263 501.24 178.52
## <none>                          498.98 179.15
## - ldaps_lh          1     3.469 502.45 179.24
## - present_tmin      1     3.892 502.87 179.50
## - ldaps_tmin_lapse  1     6.811 505.79 181.25
## - ldaps_cc1         1     7.041 506.02 181.39
## - ldaps_ppt2        1     8.609 507.59 182.33
## - present_tmax      1    15.245 514.22 186.26
## - ldaps_ws          1    17.817 516.79 187.78
## - ldaps_cc4         1    25.201 524.18 192.07
## - ldaps_tmax_lapse  1    34.062 533.04 197.15
##
## Step:  AIC=177.66
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc2 +
##     ldaps_cc4 + ldaps_ppt2 + solar_radiation
##
##                    Df Sum of Sq    RSS    AIC
## - solar_radiation   1     0.850 500.67 176.17
## - ldaps_cc2         1     1.740 501.56 176.71
## - ldaps_r_hmin      1     2.365 502.19 177.09
## - ldaps_lh          1     3.121 502.95 177.54
## <none>                          499.82 177.66
## - present_tmin      1     3.689 503.51 177.89
## - ldaps_cc1         1     6.204 506.03 179.40
## - ldaps_tmin_lapse  1     6.584 506.41 179.62
## - ldaps_ppt2        1    10.826 510.65 182.15
## - present_tmax      1    14.965 514.79 184.60
## - ldaps_ws          1    17.374 517.20 186.01
## - ldaps_cc4         1    25.198 525.02 190.56
## - ldaps_tmax_lapse  1    33.321 533.15 195.21
##
## Step:  AIC=176.17
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc2 +
##     ldaps_cc4 + ldaps_ppt2
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_cc2         1     1.869 502.54 175.30
## - ldaps_lh          1     2.695 503.37 175.80
## - ldaps_r_hmin      1     2.958 503.63 175.96
## <none>                          500.67 176.17
## - present_tmin      1     4.284 504.96 176.76
## - ldaps_cc1         1     5.541 506.22 177.51
## - ldaps_tmin_lapse  1     6.196 506.87 177.90
## - ldaps_ppt2        1    10.811 511.49 180.65
## - present_tmax      1    15.527 516.20 183.43
## - ldaps_ws          1    16.774 517.45 184.16
## - ldaps_cc4         1    24.692 525.37 188.76
## - ldaps_tmax_lapse  1    33.217 533.89 193.64
##
## Step:  AIC=175.3
## error_tmax ~ present_tmax + present_tmin + ldaps_r_hmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc4 +
##     ldaps_ppt2
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_r_hmin      1    2.5562 505.10 174.84
## <none>                          502.54 175.30
## - present_tmin      1    4.2519 506.80 175.86
## - ldaps_lh          1    4.5474 507.09 176.03
## - ldaps_tmin_lapse  1    5.5054 508.05 176.60
## - ldaps_ppt2        1    9.2126 511.76 178.81
## - present_tmax      1   14.3322 516.88 181.82
## - ldaps_cc1         1   14.5385 517.08 181.94
## - ldaps_ws          1   17.2698 519.81 183.54
## - ldaps_cc4         1   27.6559 530.20 189.53
## - ldaps_tmax_lapse  1   31.3690 533.91 191.65
##
## Step:  AIC=174.84
## error_tmax ~ present_tmax + present_tmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_lh + ldaps_cc1 + ldaps_cc4 +
##     ldaps_ppt2
##
##                    Df Sum of Sq    RSS    AIC
## - ldaps_lh          1     3.003 508.10 174.64
## <none>                          505.10 174.84
## - present_tmin      1     3.566 508.67 174.97
## - ldaps_ppt2        1    10.235 515.34 178.92
## - ldaps_tmin_lapse  1    12.532 517.63 180.27
## - present_tmax      1    12.539 517.64 180.27
## - ldaps_cc1         1    12.759 517.86 180.40
## - ldaps_ws          1    15.871 520.97 182.21
## - ldaps_cc4         1    25.198 530.30 187.59
## - ldaps_tmax_lapse  1    56.425 561.53 204.93
##
## Step:  AIC=174.64
## error_tmax ~ present_tmax + present_tmin + ldaps_tmax_lapse +
##     ldaps_tmin_lapse + ldaps_ws + ldaps_cc1 + ldaps_cc4 + ldaps_ppt2
##
##                    Df Sum of Sq    RSS    AIC
## - present_tmin      1     2.385 510.49 174.06
## <none>                          508.10 174.64
## - ldaps_ppt2        1     8.855 516.96 177.87
## - ldaps_tmin_lapse  1     9.605 517.71 178.31
## - ldaps_cc1         1    12.887 520.99 180.22
## - ldaps_ws          1    13.361 521.46 180.50
## - present_tmax      1    14.106 522.21 180.93
## - ldaps_cc4         1    26.998 535.10 188.32
## - ldaps_tmax_lapse  1    65.407 573.51 209.33
##
## Step:  AIC=174.06
## error_tmax ~ present_tmax + ldaps_tmax_lapse + ldaps_tmin_lapse +
##     ldaps_ws + ldaps_cc1 + ldaps_cc4 + ldaps_ppt2
##
##                    Df Sum of Sq    RSS    AIC
## <none>                          510.49 174.06
## - ldaps_tmin_lapse  1     7.308 517.80 176.36
## - ldaps_ppt2        1     8.444 518.93 177.03
## - present_tmax      1    11.743 522.23 178.95
## - ldaps_ws          1    14.048 524.54 180.28
## - ldaps_cc1         1    15.940 526.43 181.37
## - ldaps_cc4         1    25.638 536.13 186.90
## - ldaps_tmax_lapse  1    64.431 574.92 208.07
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = error_tmax ~ present_tmax + ldaps_tmax_lapse + ldaps_tmin_lapse +
##     ldaps_ws + ldaps_cc1 + ldaps_cc4 + ldaps_ppt2, data = seoul)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3607 -0.7442 -0.0393  0.7119  4.2661
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.67038    1.11228  -4.199 3.55e-05 ***
## present_tmax     -0.10171    0.03904  -2.605 0.009653 **
## ldaps_tmax_lapse  0.32143    0.05268   6.102 3.29e-09 ***
## ldaps_tmin_lapse -0.12427    0.06047  -2.055 0.040754 *
## ldaps_ws          0.09073    0.03184   2.849 0.004691 **
## ldaps_cc1         1.21893    0.40161   3.035 0.002619 **
## ldaps_cc4         1.38843    0.36071   3.849 0.000145 ***
## ldaps_ppt2       -0.11198    0.05069  -2.209 0.027940 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.315 on 295 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.162,  Adjusted R-squared:  0.1421
## F-statistic: 8.146 on 7 and 295 DF,  p-value: 4.521e-09
```

d. About 16 percent of the next day error is explained by the model.