

Lab 02C: Week 7 (Solutions)

Contents

1 Quick quiz

- 1.1 Confidence intervals
- 1.2 Bootstrap

2 Group work

3 Questions

- 3.1 Speed of light
- 3.2 Cotinine
- 3.3 Cereal

4 For practice after the computer lab

- 4.1 Further reading
- 4.2 Three blind mice

The **specific aims** of this lab are:

- practice using the bootstrap to construct confidence intervals
- find p-values using permutation tests
- generate discussion and provide an opportunity to practise *statistical thinking* and *communicating statistical concepts*

The unit **learning outcomes** addressed are:

- LO1 Formulate domain/context specific questions and identify appropriate statistical analysis.
- LO3 Construct, interpret and compare numerical and graphical summaries of different data types including large and/or complex data sets.
- LO5 Identify, justify and implement appropriate parametric or non-parametric statistical tests.
- LO8 Create a reproducible report to communicate outcomes using a programming language.

1 Quick quiz

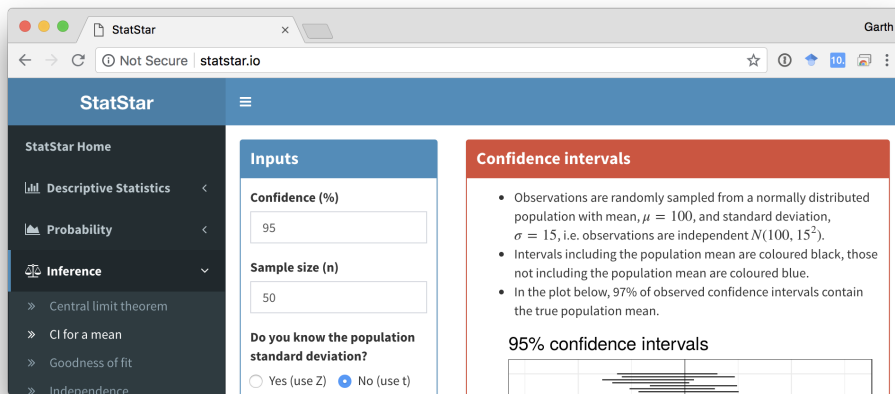
1.1 Confidence intervals

You can explore these ideas at <http://statstar.io> selecting *Inference* on the left and then

CI for a mean.

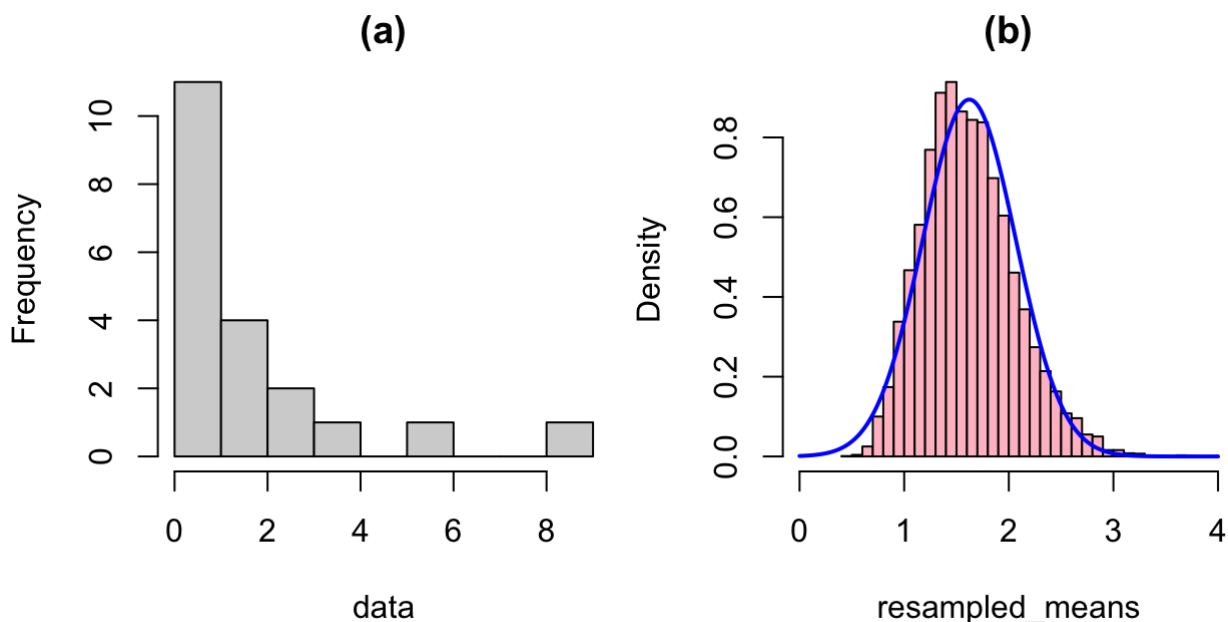
1. If I increase the sample size the confidence interval will get **wider/narrower** (holding all else constant).
2. If I increase the level of confidence (or equivalently decrease α) the confidence interval will get **wider/narrower**.
3. If I do 100 experiments under identical conditions and calculate 95% confidence intervals for each experiment then exactly 95 of those confidence intervals will contain the true population mean.

True/False



1.2 Bootstrap

A researcher wanted to empirically estimate the distribution of the sample mean in the data she had collected. To do this she decided to use bootstrap resampling. Below is a histogram of (a) the raw data and (b) the resampled means with the normal approximation curve.



Which of the following statements is the least correct

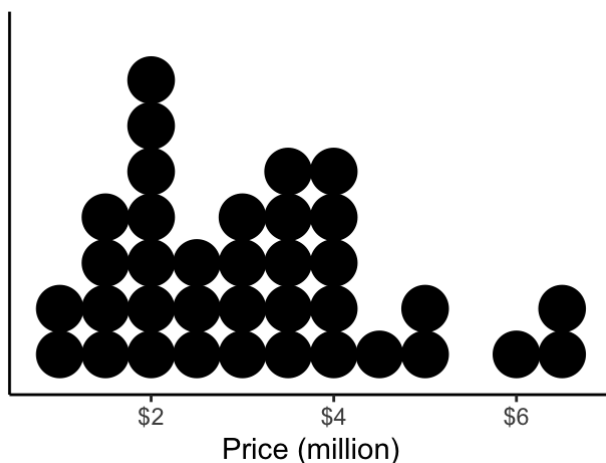
- a. The distribution of the resampled means is right skewed.
- b. The upper bound of the confidence interval derived from the bootstrapped resamples should be greater than the upper bound derived from a normal approximation.
- c. The mean of the resampled means should be greater than the median of the resampled means.
- d. The skew of the resampled means is in the same direction as the skew of the observed data.
- e. The lower bound of the confidence interval derived from the bootstrapped resamples should be less than the lower bound derived from a normal approximation.

e.

2 Group work

We would like to report the average house price for a street of 36 houses. The prices of these house are give in the table below (in millions of dollars).

Prices	1	2	3	4	5	6
1	2.1	2.4	5.9	2.8	2.9	6.4
2	3.4	1.4	1.9	2.2	5.0	3.3
3	3.3	2.9	2.1	6.6	3.5	1.0
4	3.9	2.1	1.6	2.4	1.6	1.9
5	2.0	1.2	4.1	2.9	1.5	5.1
6	3.4	2.3	4.3	4.2	4.1	3.8



The true mean of the houses in the street is \$3.1 million. Unfortunately, you only have money to buy price estimates on 6 houses.

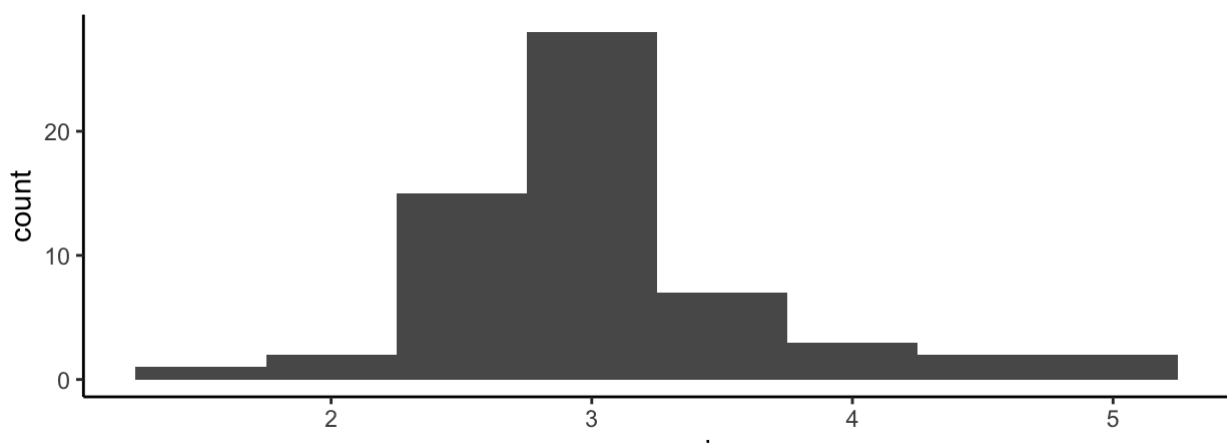
1. Using two rolls of a dice, buy 6 unique housing reports (i.e. without replacement). If you don't carry dice with you you can use `sample(1:6, 2, replace = TRUE)` to simulate rolling two dice. Use these two numbers to identify a random house price in the table above (numbers 1-6 across the top and down the side). Repeat 6 times to get a sample of six prices. (If you hit the same house, roll again.) Report the mean of your sample to the rest of the class. Your tutor will provide a link to an online spreadsheet to aggregate the results.
- What does the distribution of the class' sample means look like?
2. Using your individual sample, estimate the distribution of the sample mean using bootstrap resampling. Use 200 bootstrap resamples. Calculate a bootstrap confidence interval for the population mean average house price of the street.
- Does the true population mean lie in this confidence interval?
 - Does your bootstrap distribution of the sample mean look like the distribution of the means collected from your classmates?

1. For a hypothetical class of 60 students. Each of the 60 students samples 6 property prices (without replacement) from the population. The means of these 60 samples are stored in `res`:

```
population = c(2.1, 2.4, 5.9, 2.8, 2.9, 6.4, 3.4, 1.4, 1.9, 2.2, 5, 3.3,
               3.3, 2.9, 2.1, 6.6, 3.5, 1, 3.9, 2.1, 1.6, 2.4, 1.6, 1.9, 2, 1.2, 4.1,
               2.9, 1.5, 5.1, 3.4, 2.3, 4.3, 4.2, 4.1, 3.8)
res = NULL
set.seed(123)
for (i in 1:60) {
  student_sample = sample(population, 6, replace = FALSE)
  res[i] = mean(student_sample)
}
```

We can plot the distribution of `res` the 60 means from samples of size $n = 6$ as follows:

```
library(ggplot2)
ggplot(data.frame(average_prices = res), aes(x = average_prices)) +
  geom_histogram(binwidth = 0.5) +
  theme_classic()
```



average_prices

2. Pick one sample of size 6:

```
set.seed(1234)
my_sample = sample(population, 6, replace = FALSE)
my_sample
```

```
[1] 2.9 6.6 2.4 1.9 2.9 6.4
```

Take 200 bootstrap resamples from your sample:

```
bs_res = NULL
for(i in 1:200){
  bs_sample = sample(my_sample, 6, replace = TRUE)

  bs_res[i] = mean(bs_sample)
}
```

Bootstrap confidence interval:

```
bs_ci = quantile(bs_res, c(0.025, 0.975))
bs_ci
```

```
2.5% 97.5%
2.4    5.3
```

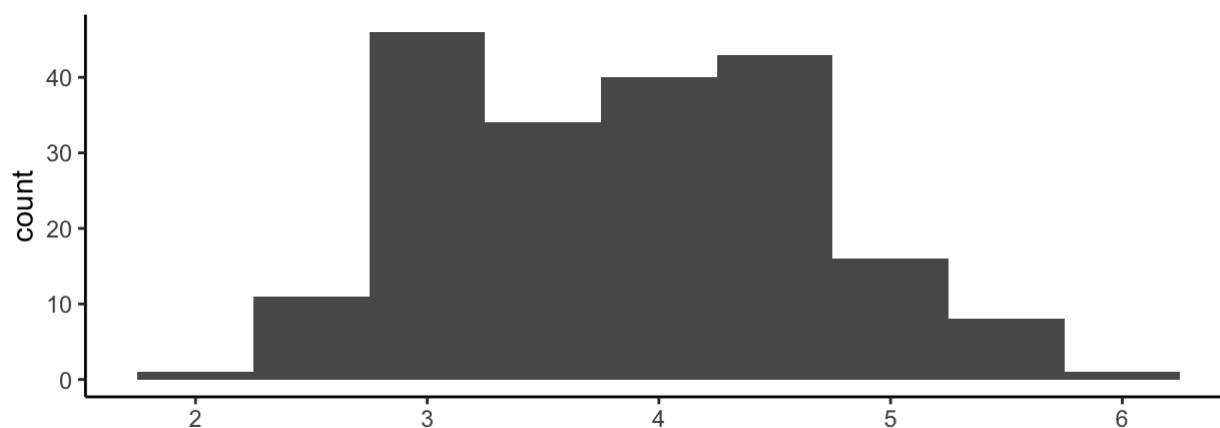
Yes, the true value is inside our bootstrap confidence interval.

```
true_value = 3.1
true_value > bs_ci[1] & true_value < bs_ci[2]
```

```
2.5%
TRUE
```

Distribution of the bootstrap sample mean:

```
ggplot(data.frame(average_prices=bs_res),
  aes(x = average_prices)) +
  geom_histogram(binwidth = 0.5) +
  theme_classic()
```



The key idea of this activity is to get you thinking about:

- taking an individual sample (of size 6)
- taking repeated samples (everyone in the class took a sample)
- the distribution of the mean from repeated samples (sampling distribution of the sample mean)
- bootstrapping from individual samples

3 Questions

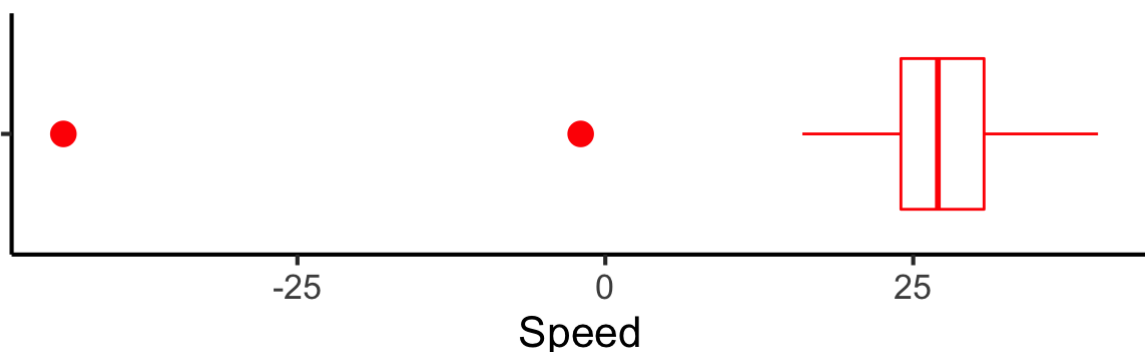
3.1 Speed of light

In the lecture, we discussed a famous dataset where Simon Newcomb measured the time required for light to travel from his laboratory on the Potomac River to a mirror at the base of the Washington Monument and back, a total distance of about 7400 meters. You can download the [data file](https://raw.githubusercontent.com/DATA2002/data/master/speed_of_light.tx) that contains 66 sets of measurements used to estimate the speed of light.

Getting started: Read in the data and check the size of your data.

```
library(tidyverse)
speed_file =
  read_csv("https://raw.githubusercontent.com/DATA2002/data/master/speed_of_light.tx")

speed = speed_file$Speed_of_Light
ggplot(speed_file, aes(x="", y = Speed_of_Light)) +
  geom_boxplot(colour = "red", outlier.size = 4) +
  theme_classic(base_size = 16) +
  labs(x = "", y = "Speed") + coord_flip()
```

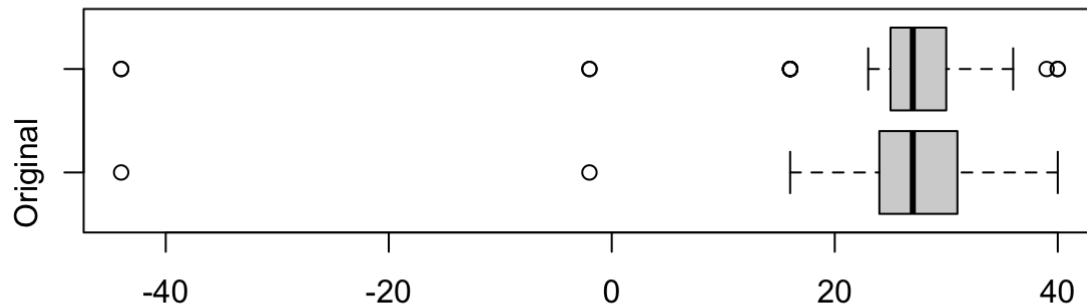


- Generate one bootstrap sample and compare this sampled data with the original data.

The easiest thing would be to generate side by side boxplots:

The easiest thing would be to generate side by side boxplots.

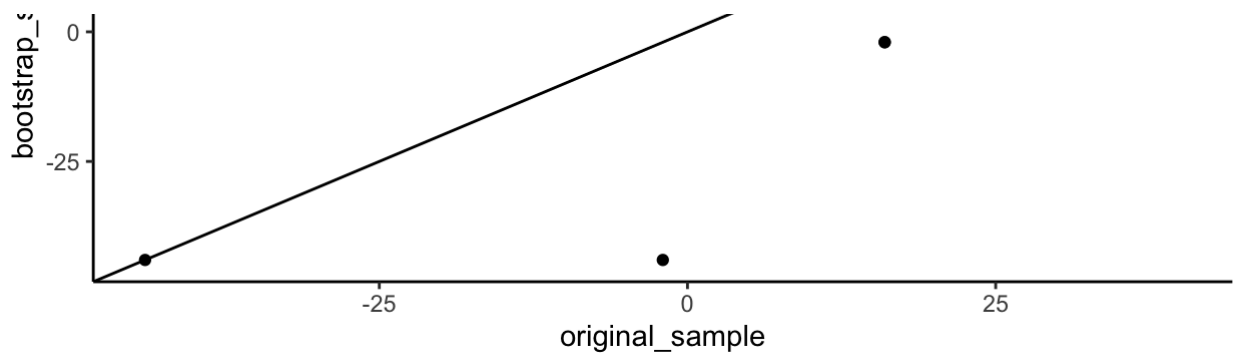
```
speedb1 = sample(speed, replace = TRUE)
boxplot(speed, speedb1, horizontal = TRUE, names = c("Original", "BS"))
```



We could also generate something like a QQ plot, by plotting the sorted original data against the sorted bootstrap data.

```
## qqplot
df = data.frame(original_sample = sort(speed),
                 bootstrap_sample = sort(speedb1))
ggplot(df, aes(x = original_sample, y = bootstrap_sample)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0) +
  theme_classic()
```



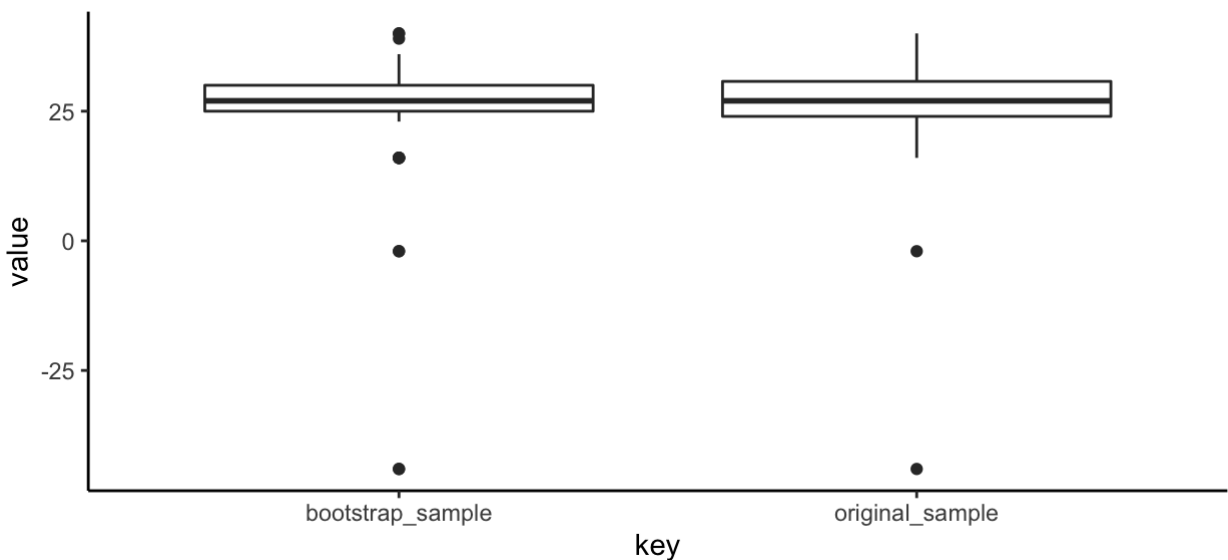


- b. Compute the mean and median of the bootstrap sample and compare with the corresponding values in the original data.

```
df %>%
  summarise_all(.funs = list(mean = mean, median = median))

original_sample_mean bootstrap_sample_mean original_sample_median
1          26.21212          24.78788          27
bootstrap_sample_median
1          27

df %>%
  gather() %>%
  ggplot(aes(x = key, y = value)) + geom_boxplot() + theme_classic()
```



```
# or in base R: mean(speedb1) mean(speed) median(speedb1)
# median(speed) boxplot(df)
```

- c. Draw another bootstrap sample and repeat the comparison. Repeat this 20 times and see if your conclusion changes. Inspect first ten bootstrap estimates of the mean. Visualise the result. *Hint:* Write a for loop (see example in lecture notes).


```

B = 20
speedbmean = numeric(B)
for (i in 1:B) {
  resampled_data = sample(speed, replace = TRUE)
  # boxplot(speed, resampled_data)
  speedbmean[i] = mean(resampled_data)
}
speedbmean[1:10]

[1] 25.04545 27.62121 24.01515 25.98485 27.86364 27.28788 26.12121
[8] 24.81818 25.92424 27.19697

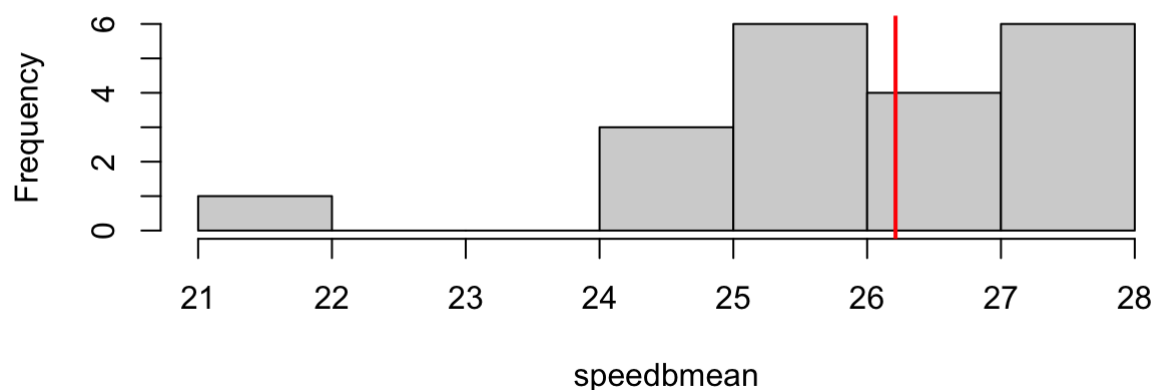
```

```

hist(speedbmean)
abline(v = mean(speed), col = "red", lwd = 2)

```

Histogram of speedbmean



d. Typically one draws a large number of bootstrap samples, say 1000 or more. Try different numbers of bootstrap samples and see how the shape of the histogram changes.

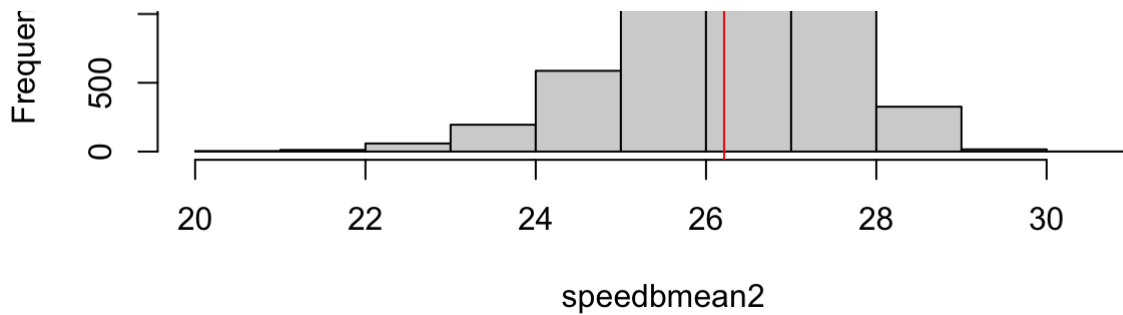
```

B = 5000
speedbmean2 = numeric(B)
for (i in 1:B) {
  resampled_data = sample(speed, replace = TRUE)
  speedbmean2[i] = mean(resampled_data)
}
hist(speedbmean2)
abline(v = mean(speed), col = "red")

```

Histogram of speedbmean2





Once you get up around 1000, 5000, 10000, 100000, the shape of the histogram stays almost exactly the same, i.e. there's not really any advantage to taking larger and larger bootstrap resamples.

- e. Find a 95% bootstrap confidence interval for the mean using the 2.5 and 97.5 percentiles as the confidence limits. Compare this with a "traditional" confidence interval that uses the t-distribution.

```
quantile(speedbmean, c(0.025, 0.975))

      2.5%      97.5%
22.92538 27.82045

n = length(speed)
mean(speed) + qt(c(0.025, 0.975), n - 1) * sd(speed)/sqrt(n)

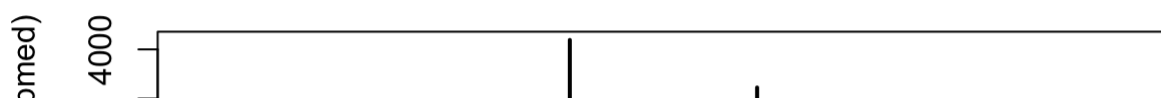
[1] 23.57059 28.85365
```

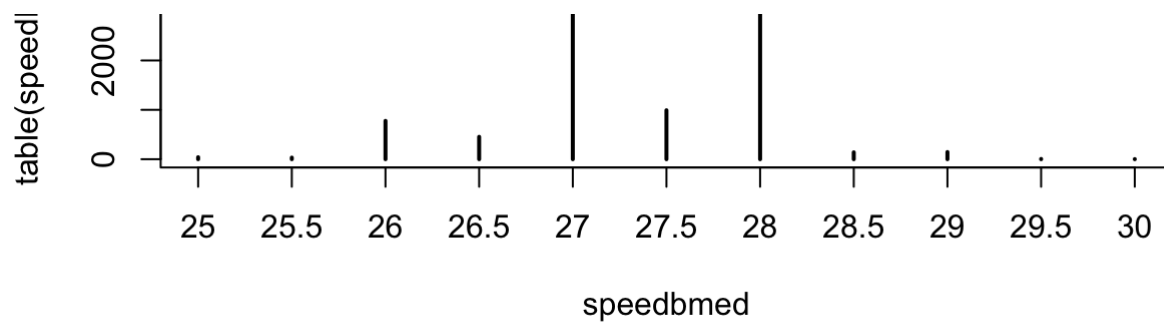
- f. Generate 95% bootstrap confidence intervals calculation for the median and the MAD.

```
B = 10000
speedbmed = speedbmad = numeric(B)
for (i in 1:B) {
  resampled_data = sample(speed, replace = TRUE)
  speedbmed[i] = median(resampled_data)
  speedbmad[i] = mad(resampled_data)
}
# the median distribution is quite discrete
speedbmed[1:10]

[1] 28.0 28.0 27.0 27.0 27.0 27.0 27.0 27.0 27.5 28.0

plot(table(speedbmed))
```





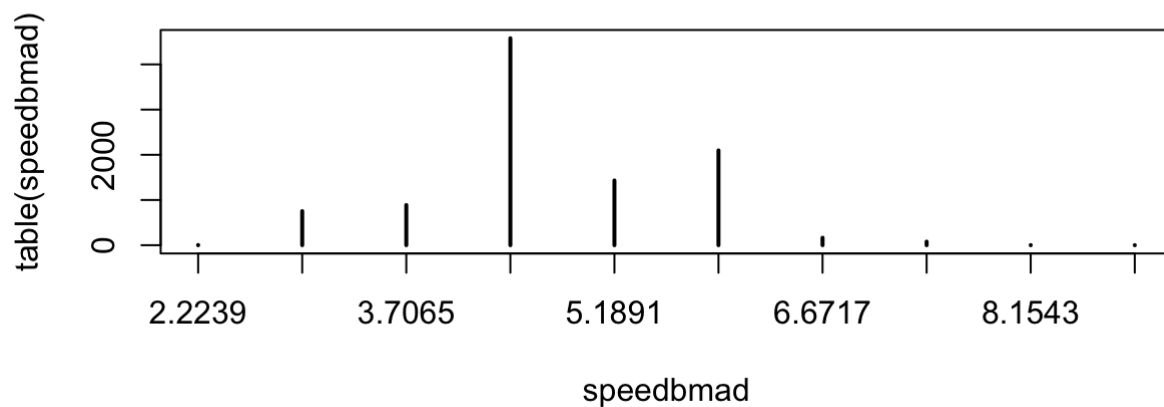
```
hist(speedbmed)
abline(v = median(speed), col = "red")
```



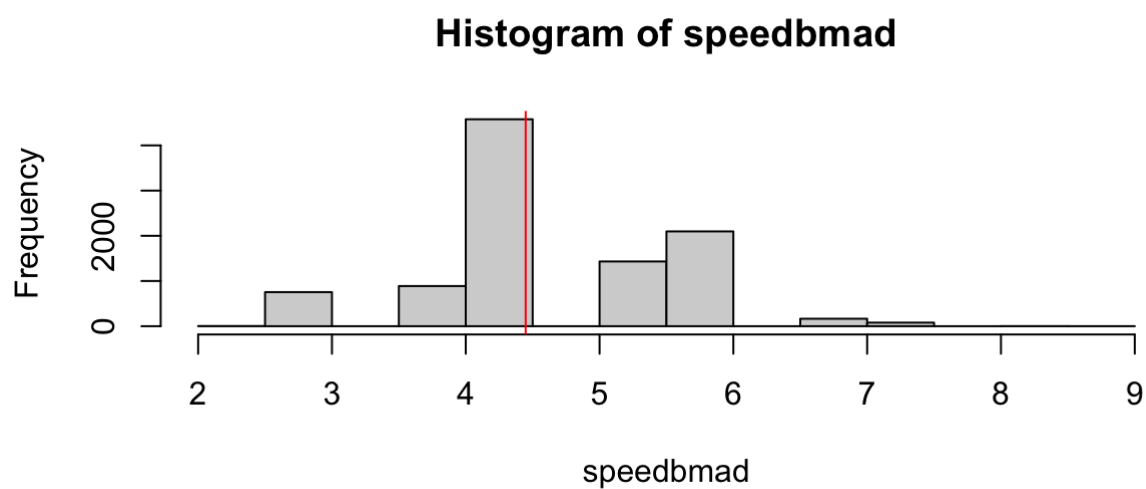
```
quantile(speedbmed, c(0.025, 0.975))
```

```
2.5% 97.5%
26.0  28.5
```

```
# the mad distribution is also quite discrete
plot(table(speedbmad))
```



```
hist(speedbmad)
abline(v = mad(speed), col = "red")
```



```
quantile(speedbmad, c(0.025, 0.975))
```

```
2.5% 97.5%
2.9652 5.9304
```

3.2 Cotinine

A variant of nicotine found in cigarettes is cotinine (which, not coincidentally, is an anagram of nicotine). It is found in the blood stream and the amount is proportional to the amount of exposure a person has to tobacco smoke. Therefore, cotinine is used as an indicator of tobacco smoke exposure. For example, cotinine levels < 10 ng/ml are considered to be consistent with no active smoking; values between 10 - 100 ng/ml are associated with light smoking, or moderate passive exposure while heavy smokers have at least 300 ng/ml. Levels in active smokers typically reach 500 ng/ml.

The following data lists the cotinine levels of 40 passive smokers who are not smokers of tobacco products.

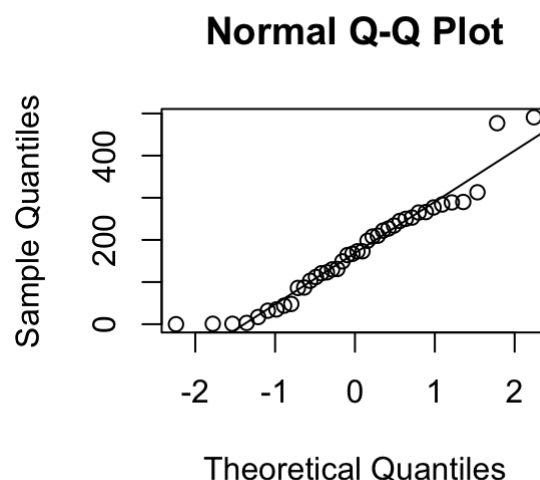
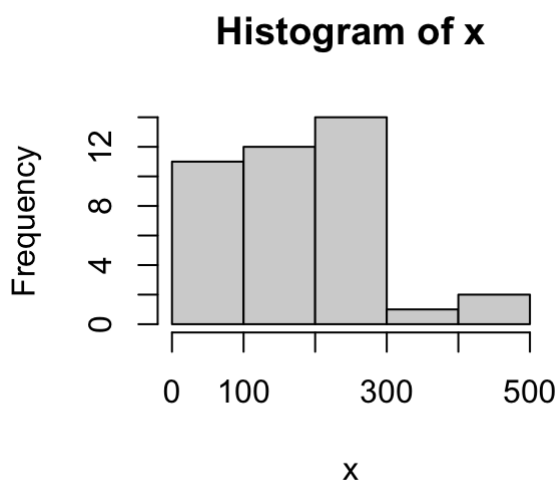
```
x = c(0, 87, 173, 253, 1, 103, 173, 265, 1, 112, 198, 266, 3, 121, 208,
      277, 17, 123, 210, 284, 32, 130, 222, 289, 35, 131, 227, 290, 44, 149,
      234, 313, 48, 164, 245, 477, 86, 167, 250, 491)
```

1. Calculate some simple descriptive measures of the data, construct a histogram and a QQ plot. Provide a brief description of the sample data.

```
summary(x)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	86.75	170.00	172.47	250.75	491.00

```
par(mfrow = c(1, 2))
hist(x)
qqnorm(x)
qqline(x)
```



While the mean and the median are similar, graphically there does appear to be some skewness with two quite large observations. There is some departure from the line in the normal QQ plot, particularly at the extremes, indicating the normality assumption may not be satisfied.

2. Based on your descriptive summary of the data, do you think there are any outlying or unusually

- Based on your descriptive summary of the data, do you think there are any especially large, observations that may impact upon any inferential test that you perform? In your description, take into consideration the summary statistics and histogram of the remaining data.

One may argue that perhaps the skewness is due to a few outlying observations - for example the cotinine levels of 477 and 491. However these may be the cotinine levels of regular heavy smokers. So it would not be impossible to obtain observations as extreme as these - the description of cotinine levels even suggests this. So we should not ignore them or remove them from the data.

3. Using R and the complete sample, perform a standard t test of the hypotheses $H_0: \mu = 130$ vs $H_1: \mu \neq 130$. At the 5% level of significance, what can you conclude about the cotinine levels of the smokers in the population?

```
t.test(x, mu = 130)
```

One Sample t-test

```
data: x
t = 2.248, df = 39, p-value = 0.0303
alternative hypothesis: true mean is not equal to 130
95 percent confidence interval:
 134.2576 210.6924
sample estimates:
mean of x
 172.475
```

The p-value is 0.03, less than 0.05 therefore we reject the null hypothesis at the 5% level of significance and conclude that the true mean cotinine level is not equal to 130.

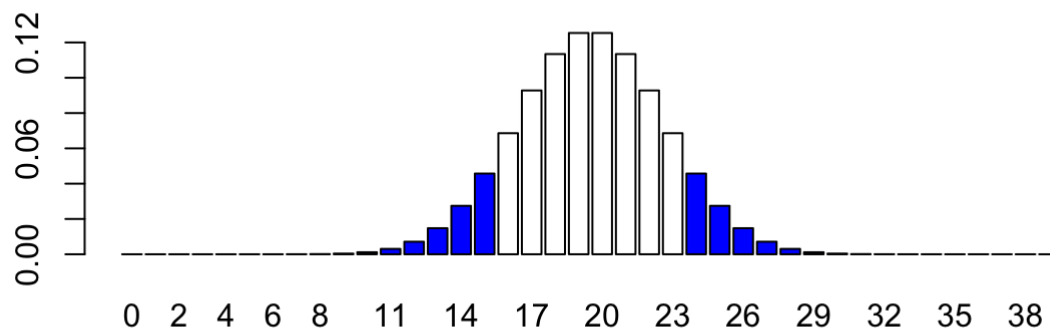
4. Perform a sign test to test $H_0: \mu = 130$ vs $H_1: \mu \neq 130$.

```
# remove any zero differences
x1 = x[x != 130]
bigger = x1 > 130
table(bigger)
```

```
bigger
FALSE  TRUE
   15    24
```

24 out of 39 are greater than 130.

```
barplot(dbinom(0:39, 39, 0.5), names = 0:39, col = c(rep("blue", 16), rep("white",
8), rep("blue", 15)))
```



```
2 * (pbinom(15, 39, 0.5))
```

```
[1] 0.1995909
```

```
2 * (1 - pbinom(23, 39, 0.5))
```

```
[1] 0.1995909
```

```
binom.test(table(bigger))
```

Exact binomial test

```
data: table(bigger)
```

```
number of successes = 15, number of trials = 39, p-value =  
0.1996
```

```
alternative hypothesis: true probability of success is not equal to 0.5
```

```
95 percent confidence interval:
```

```
0.2336393 0.5538094
```

```
sample estimates:
```

```
probability of success  
0.3846154
```

Using the sign test we do not reject the null hypothesis at the 5% level of significance as the p-value is larger than 0.05.

4. Perform a permutation test by generating 10,000 resamples. What conclusion do you reach based on the permutation test p-value?

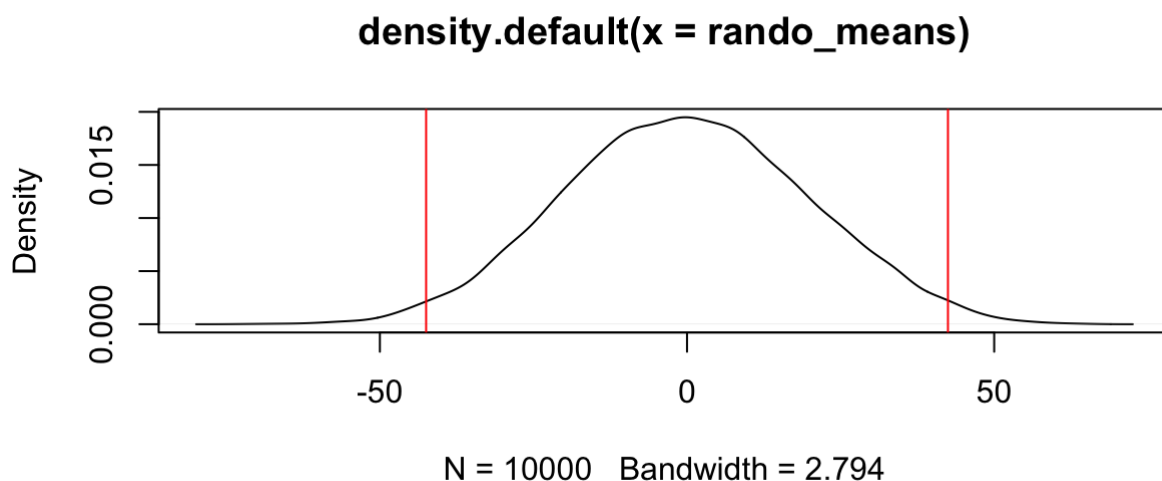
Remembering the importance of exchangeability, we can't just resample our observations with replacement (like in the bootstrap). Since this is a test of a mean, to make our randomisation p-value we need to randomly sample variation around the mean. We do this by randomly assigning a sign change to the differences between the data and the hypothesised mean. In this way, the (mean-adjusted) data stays the same except for a potential sign change which is what generates our null

distribution.

```
repetitions = 10000  
hyp_mean = 130  
n = length(x)
```

Under H_0 , the data has a mean of 130. Re-centre the data by this. Under H_0 , random re-labelling of the absolute values of the centred data should have a mean of zero.

```
diff_data = x - hyp_mean  
dbar = mean(diff_data)  
rando_means = numeric(repetitions)  
  
for (i in 1:repetitions){  
  permuted_signs = sample(c(-1, 1), n, replace = TRUE)  
  permuted_data = permuted_signs * diff_data  
  rando_means[i] = mean(permuted_data)  
}  
pval = mean(abs(rando_means) >= abs(dbar))  
plot(density(rando_means))  
abline(v = dbar, col = "red")  
abline(v = -dbar, col = "red")
```



The permutation test p-value is 0.026 which is less than 0.05 so we reject the null hypothesis.

5. Which of the procedures above provides the more appropriate inference of the population mean. Why?

The permutation test, as it does not assume the observations are sampled from a normal population.

6. Generate 95% bootstrap confidence intervals for the standard deviation and the median absolute deviation from the median (MAD). Plot histograms of both bootstrap distributions. Which of these two estimators of scale is more reliable in this setting.


```

B = 10000
sd_boot = numeric(B)
mad_boot = numeric(B)
for (i in 1:B) {
  boot_sample = sample(x, replace = TRUE)
  sd_boot[i] = sd(boot_sample)
  mad_boot[i] = mad(boot_sample)
}
quantile(sd_boot, c(0.025, 0.975))

```

```

      2.5%      97.5%
89.56265 145.37794

```

```

quantile(mad_boot, c(0.025, 0.975))

```

```

      2.5%      97.5%
80.0604 168.2751

```

```

diff(quantile(sd_boot, c(0.025, 0.975)))

```

```

      97.5%
55.81529

```

```

diff(quantile(mad_boot, c(0.025, 0.975)))

```

```

      97.5%
88.2147

```

The length of the bootstrap confidence interval for the standard deviation is 55.8152892, which is narrower than the length of the confidence interval for the median absolute deviation from the median, 88.2147.

7. Say we were interested in the coefficient of variation, $CV = s/\bar{x}$, for this data set. Generate a 90% bootstrap confidence interval for the coefficient of variation.

```

B = 10000
bs_cv = numeric(B)
for (i in 1:B) {
  boot_sample = sample(x, replace = TRUE)
  bs_cv[i] = sd(boot_sample)/mean(boot_sample)
}
quantile(bs_cv, c(0.05, 0.95))

```

```

      5%      95%
0.5429697 0.8333776

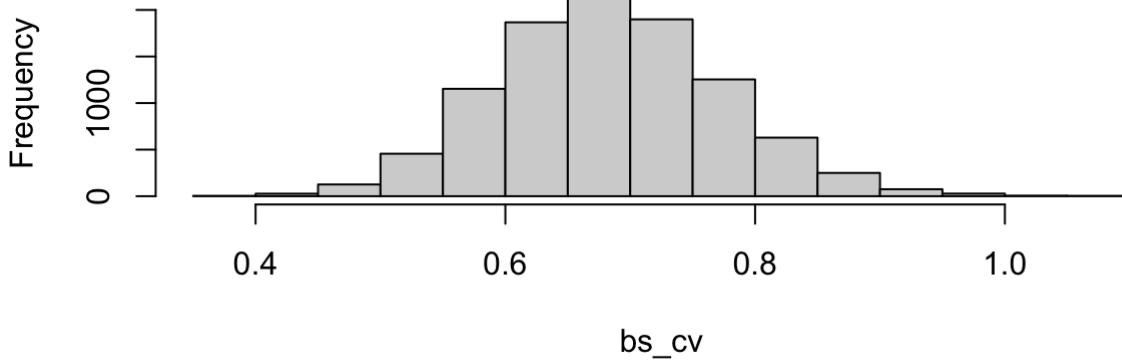
```

```

hist(bs_cv)

```

Histogram of bs_cv



The 90% bootstrap confidence interval for the coefficient of variation is (0.5429697, 0.8333776).

3.3 Cereal

This cereal data is taken from Data Description, Inc. (2021) and can be downloaded from <https://raw.githubusercontent.com/DATA2002/data/master/Cereal.csv>.

One of the variable `mfr` represents the manufacturer of cereal where `A` = American Home Food Products, `G` = General Mills, `K` = Kelloggs, `N` = Nabisco, `P` = Post, `Q` = Quaker Oats, `R` = Ralston Purina.

Please note that the code provided should be used as a guide only and not complete solutions.

Getting started: Read the data from the website. Check the size of your data. Think about what the number of rows actually means.

```
library(tidyverse)
cereal = read_csv("https://raw.githubusercontent.com/DATA2002/data/master/Cereal.csv",
  na = "-1")
# if you've downloaded it to your computer cereal =
# read_delim('Cereals.txt', delim = '\t', na = '-1') Looking at the
# start of the data
dplyr::glimpse(cereal)
```

1. Produce some basic summary statistics the nutrients "sugar" and "sodium"

```
cereal %>%
  drop_na(sugars, sodium) %>%
  summarise(across(.cols = c(sugars, sodium), .fns = list(mean = mean,
    sd = sd, min = min, max = max, q2 = ~quantile(.x, 0.25), q3 = ~quantile(.x,
    0.75)))) %>%
  tidyr::pivot_longer(cols = everything(), names_sep = "_", names_to = c("nutrient",
    "value")) %>%
```

```
knitr::kable(digits = 2)
```

nutrient	mean	sd	min	max	q2	q3
sugars	7.03	4.38	0	15	3.00	11.0
sodium	161.78	82.32	0	320	133.75	212.5

Alternatively we could use the **table1** package to go something similar:

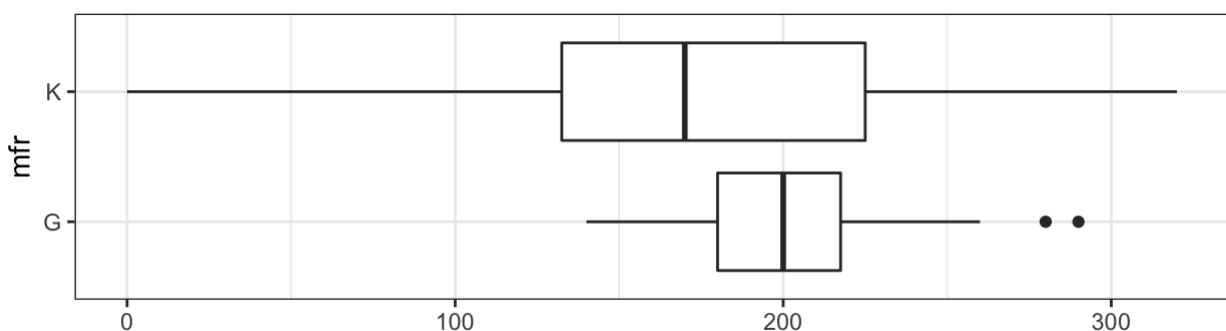
```
# install.packages(table1)
table1::table1(~sugars + sodium, data = cereal)
```

Overall (N=77)	
sugars	
Mean (SD)	7.03 (4.38)
Median [Min, Max]	7.00 [0, 15.0]
Missing	1 (1.3%)
sodium	
Mean (SD)	160 (83.8)
Median [Min, Max]	180 [0, 320]

- Restricting attention to **G** = General Mills and **K** = Kelloggs cereals, visualise the distribution of sodium content between the two manufacturers. Does it look like there is equal variance between the two groups? Could you safely assume normality within each group?

We start by subsetting and visualising the data:

```
test_dat = cereal %>%
  filter(mfr %in% c("G", "K")) %>%
  drop_na(sugars)
test_dat %>%
  ggplot() +
  aes(x = mfr, y = sodium) +
  geom_boxplot() +
  coord_flip() +
  theme_bw()
```

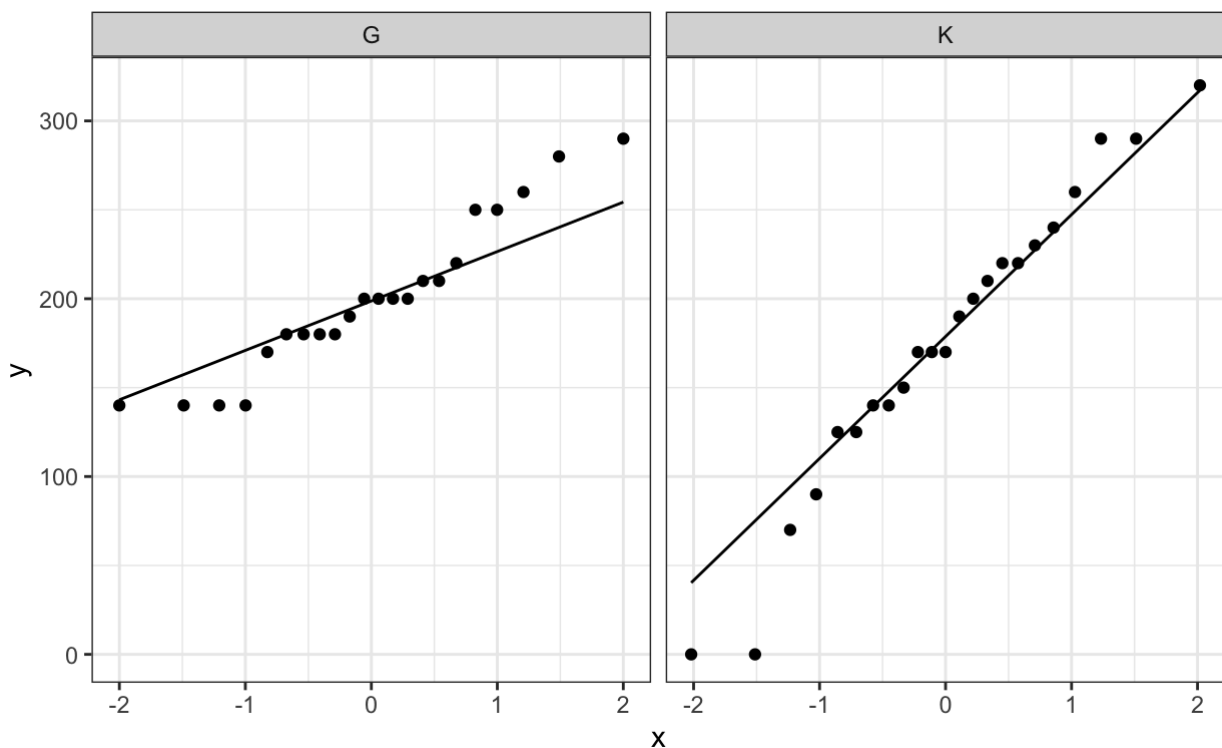


sodium

It looks like there is a difference in spread between two manufacturers, with Kellogg's appearing to have larger variation than General Mills. So we wouldn't be too comfortable using the `equal.var = TRUE` option if we were to perform a `t.test()`.

To assess the normality within each group, we can use a QQ plot and facet by manufacturer:

```
test_dat %>%  
  ggplot(aes(sample = sodium)) + geom_qq() + geom_qq_line() + facet_wrap(~mfr) +  
  theme_bw()
```



In the General Mills QQ plot, there appears to be some departure from the line at both ends of the distribution (indicating heavier tails than a normal distribution). The points are mostly close to the line in the Kellogg's sample, except for two cereals which have zero sodium. We're not totally confident in the normality assumption, though it may still be OK to use a t -test because the observations are not

wildly far from the line and we have a reasonably large number of observations, so we could instead rely on the central limit theorem to ensure the test statistic at least approximately follows a t distribution.

3. Perform a permutation test to test whether there is a significant difference in the mean sodium content between the two manufacturers.

Instead of using a two-sample t -test type test statistic, let's do a permutation test using the Wilcoxon rank-sum test statistic as our measure of discrepancy between the two group means (or medians).

There is a parameter for the `wilcox.test()` function that indicates if a continuity correction should be used when calculating the p-value using the normal approximation. The default is `correct = TRUE`. In previous examples, when we're comparing R output to our "manual" calculation I specify `correct = FALSE` to ensure the results match. In general though it's fine (and probably better) to leave it as the default setting `correct = TRUE`. Note that this has no impact on the permutation test because the continuity correction does not change the test statistic.

```
# specify exact = FALSE because there are ties and I don't want it to
# print the warning message when it defaults to exact = FALSE
t0 = wilcox.test(sodium ~ mfr, data = test_dat, exact = FALSE)
t0
```

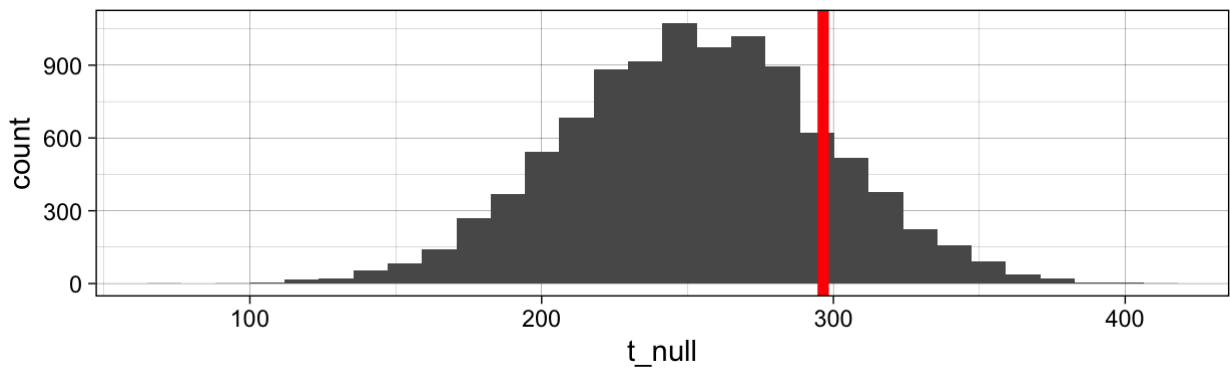
Wilcoxon rank sum test with continuity correction

data: sodium by mfr

W = 296.5, p-value = 0.3274

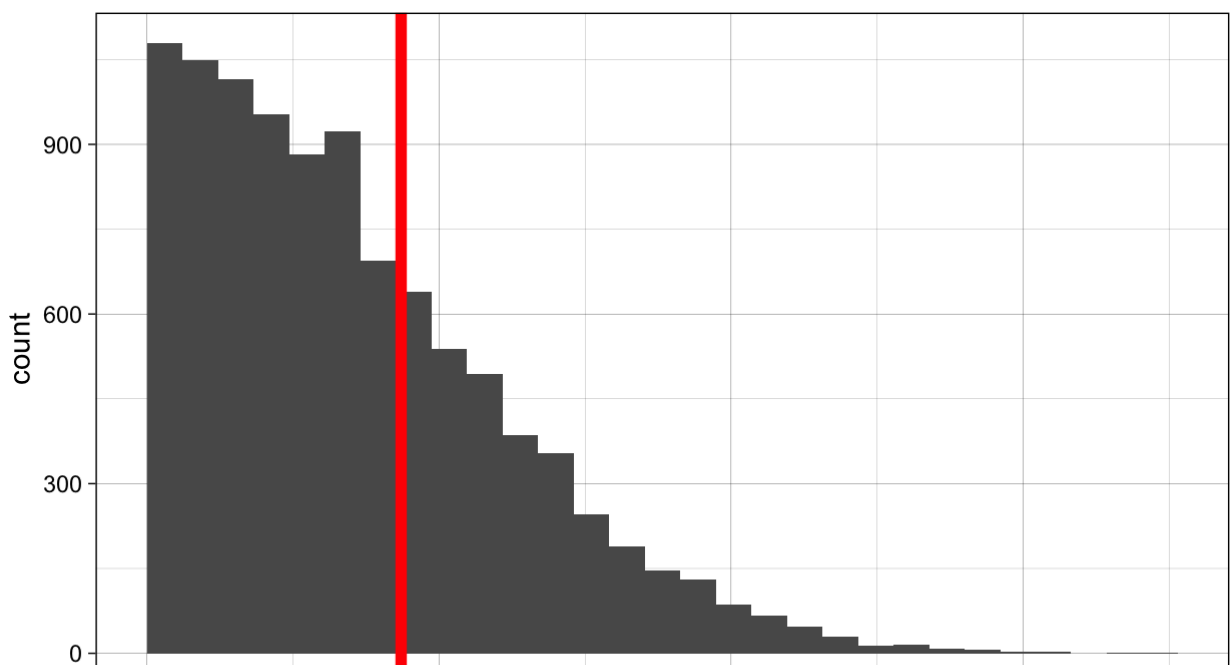
alternative hypothesis: true location shift is not equal to 0

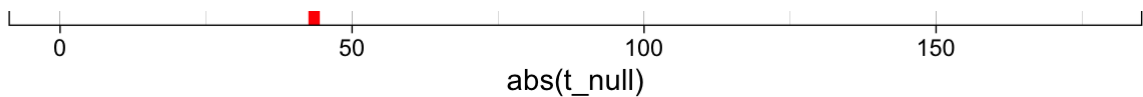
```
B = 10000 # number of permuted samples we will consider
permuted_dat = test_dat # make a copy of the data
t_null = vector("numeric", B) # initialise outside loop
for(i in 1:B) {
  permuted_dat$mfr = sample(test_dat$mfr) # this does the permutation
  t_null[i] = wilcox.test(sodium ~ mfr,
                        data = permuted_dat,
                        exact = FALSE)$statistic
}
data.frame(t_null) %>%
  ggplot() +
  aes(x = t_null) +
  geom_histogram() +
  theme_linedraw() +
  geom_vline(xintercept = abs(t0$statistic), col = "red", lwd = 2)
```



It's a two sided test, so let's center that distribution at its theoretical mean and consider the absolute value of the test statistic distribution. The method of calculating the test statistic given in the lecture notes has a corresponding mean of $n_x(N + 1)/2$. The test statistic used by R is equivalent but a bit different to what we defined in class (see [here](#)), it's different by a shift of $n_x(n_x + 1)/2$ so we need to calibrate our mean to match the definition of the test statistic used by R:

```
n_x = 22
n_y = 23
N = n_x + n_y
mu_W = n_x*(N+1)/2
t_mean = mu_W - n_x*(n_x + 1)/2
data.frame(t_null = t_null - t_mean) %>%
  ggplot() +
  aes(x = abs(t_null)) +
  geom_histogram(boundary = 0) +
  theme_linedraw() +
  geom_vline(xintercept = abs(t0$statistic - t_mean),
            col = "red", lwd = 2)
```





We can calculate our permutation p-value:

```
mean(abs(t_null - t_mean) >= abs(t0$statistic - t_mean))

[1] 0.3345
```

The permutation p-value is large (much larger than 0.05), so we do not reject the null hypothesis and conclude that there's no significant difference between the mean sodium content of the two manufacturers.

We could also do all this using a t-test type test statistic:

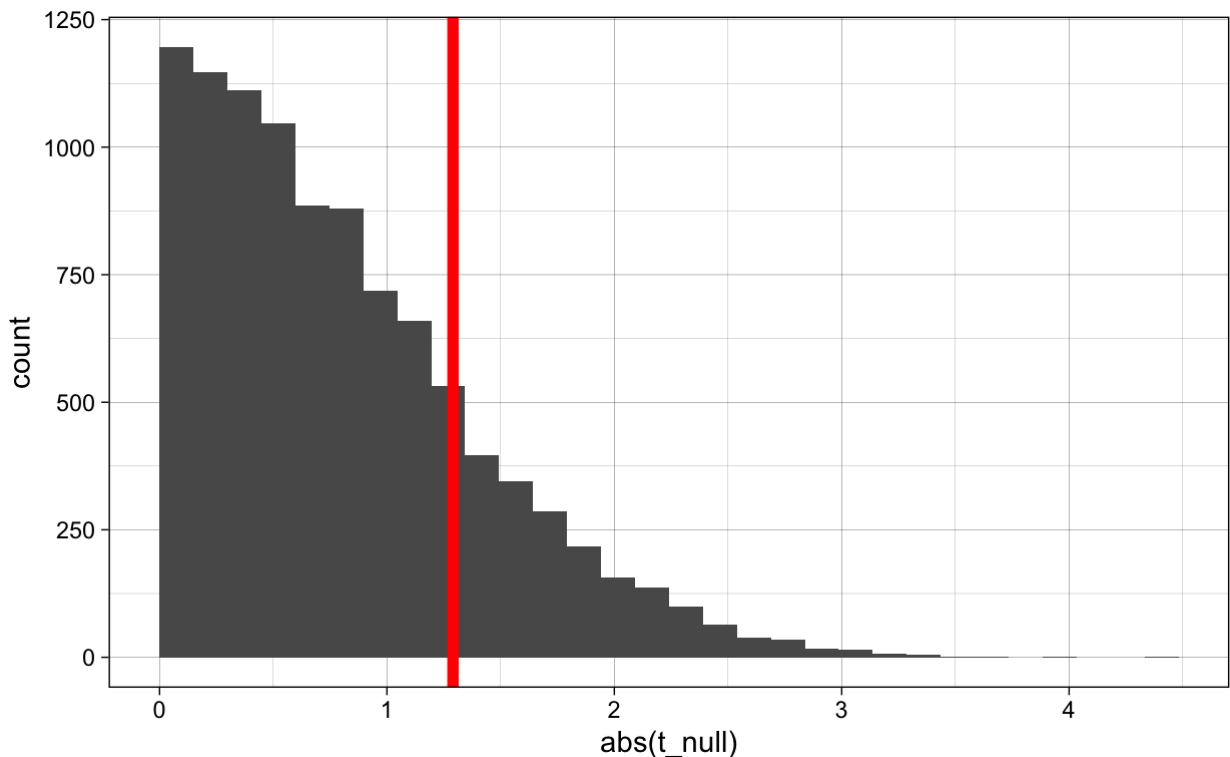
```
t0 = t.test(sodium ~ mfr, data = test_dat)
t0
```

Welch Two Sample t-test

```
data: sodium by mfr
t = 1.2903, df = 33.519, p-value = 0.2058
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -14.78243 66.12630
sample estimates:
mean in group G mean in group K
    200.4545      174.7826
```

```
B = 10000 # number of permuted samples we will consider
permuted_dat = test_dat # make a copy of the data
t_null = vector("numeric", B) # initialise outside loop
for(i in 1:B) {
  permuted_dat$mfr = sample(test_dat$mfr) # this does the permutation
  t_null[i] = t.test(sodium ~ mfr, data = permuted_dat)$statistic
}

data.frame(t_null = t_null) %>%
  ggplot() +
  aes(x = abs(t_null)) +
  geom_histogram(boundary = 0) +
  theme_linedraw() +
  geom_vline(xintercept = abs(t0$statistic), col = "red", lwd = 2)
```



We can calculate our permutation p-value:

```
mean(abs(t_null) >= abs(t0$statistic))
[1] 0.2005
```

Again, the permutation p-value is large (much larger than 0.05), so we do not reject the null hypothesis and conclude that there's no significant difference between the mean sodium content of the two manufacturers.

4 For practice after the computer lab

4.1 Further reading

For permutation testing see Curley and Milewski (2020) sections [13.1](#) and [13.3](#).

4.2 Three blind mice

Three blind mice are put on a diet. Their weights before and after the diet are {2.9, 2.4, 3.1} and {2.8, 2.2, 2.8} respectively. Using a permutation test and the test statistic \bar{d} , where d_i are the observed

weight changes, is there any evidence that the three blind mice lost weight on the diet at a significance level of $\alpha = 0.1$

We will use a permutation test to test whether there is any weight loss. To do this we will permute the signs of the differences.

- **Hypothesis:** $H_0: \mu_d = 0, H_1: \mu_d < 0$
- **Test statistic:** \bar{D} .
- **Assumptions:** That the observations are exchangeable (iid).
- **P-value:**
 - there are 2^3 possible permutations of the signs.
 - For each of permutations we calculate the mean as in the R code below.
 - The number of permutations equal to or more extreme than the original mean of the differences is 1.
 - This gives a p-value of 0.125.
- **Decision:** As the p-value is greater than $\alpha = 0.1$, there is no evidence to suggest that the diet is effective.

```
b = c(2.9, 2.4, 3.1)
a = c(2.8, 2.2, 2.8)
d = a-b
d
```

```
[1] -0.1 -0.2 -0.3
```

```
dbar = mean(d)
permutation_res = round(c(mean(c(1,1,1)*d),
                           mean(c(1,-1,1)*d),
                           mean(c(1,-1,-1)*d),
                           mean(c(1,1,-1)*d),
                           mean(c(-1,1,1)*d),
                           mean(c(-1,-1,1)*d),
                           mean(c(-1,-1,-1)*d),
                           mean(c(-1,1,-1)*d)),2)
p_value = mean(permutation_res <= dbar)
p_value
```

```
[1] 0
```

Note that this fails due to floating point errors. See [here](#) for a brief summary of the issue. Below `tol` is the numerical tolerance level above 0 we're willing to accept is due to floating point calculation errors.

```
tol = 1e-05
p_value = mean(permutation_res - dbar <= tol)
```

p_value

[1] 0.125

References

Curley, James P., and Tyler M. Milewski. 2020. *Psy317I Guidebook*. https://bookdown.org/curleyjp0/psy317I_guides/.
Data Description, Inc. 2021. *The Data and Story Library*. <https://dasl.datadescription.com/>.