Statistical Thinking: Week 4 Lab

Due 12noon Wednesday 2 September 2020

Introduction

Our focus this week (and next week) is on assessing variability in means, using one and two sample t-tests and corresponding confidence intervals. We will also work with both 'paired' and 'upaired' data.

This document is organised in two main sections, Section A and Section B, as shown below. Section B covers three different datasets, each relating to a different variation of the t-test.

A. Instructions (including Lab 4 submission deadline)

B. Lab Exercises
B.1 Single population
B.2 A pair of populations
B.3 Two independent populations

Each analysis of a dataset will focus on five distinct parts:

1. Setting up the analysis
2. Data visualisation
3. Summary statistics
4. Use of the *t.test()* to complete the 5% t-test and obtain the 95% approximate CLT-based 95% confidence interval for the relevant population parameter.
5. Manual calculation of the t-test and confidence interval from part 4.

A. Instructions (including Lab 4 submission deadline)

Before you begin **B. Lab Exercises**, review each of these detailed instructions.

- Create, name and save a new .Rmd file as **Lab04_nnnnnnnn.Rmd**, with your Monash student ID number replacing the segment **nnnnnnnn** in the file name. Including in the **YAML** section
  - a suitable title (e.g. "Statistical Thinking Lab 4")
  - put your name and student-id as the author information
  - set the date "Week 4, 2020"
  - ensure all code chunks[1] will show using the `knitr::opts_chunk$set(echo = TRUE)` command in the initial code chunk.
  - be sure to include all package library commands in the .Rmd
  - use your discretion with regard to further modification of the plots and tables for which **R** commands have been provided.

- Format your **R Markdown** files using (sub-)section heading titles corresponding to the lab exercise section headings. Refer to Lab 3 for other relevant suggestions useful to complete prior to undertaking the lab and for preparing files for submission.

- Lab 4 submissions must be completed before **12noon on Wednesday 2 September 2020**.

Good luck and have fun!

[1] The initial code chunk containing the *kitr::opts_chunk* settings do not need to be displayed.

B. Lab Exercises

The three different datasets for this Lab are:

1: The morley dataset. This is a dataset in the **R** *Datasets* package, which is installed with the base **R** application. Exercises relating to this dataset are in **Section B1**, with more information about the dataset provided at the start of the section.

2: The CBT dataset. This dataset is available in an file named **CBT.csv** on Moodle. You will need to download the **CBT.csv** file and import the data into **R** using the instructions given below. Exercises relating to this dataset are in **Section B2**, following a more detailed description of this data.

3: The birthwt dataset. This dataset is included in the **MASS** package, and information is available in the corresponding **R** help file. Exercises relating to this dataset are in **Section B3**.

For each dataset, you will undertake at least one CLT-based hypothesis test and construct at least CLT-based confidence interval. These will be completed using the *t.test()* **R** function, where the specific options of the functions used will depend on the setting of each dataset. You will also calculate the same t-test quantities and corresponding confidence interval without using the *t.test()* function.

B1: The morley dataset

The **morley** data comes from A. Michelson (1882) Experimental Determination of the Velocity of Light Made at the U.S. Naval Academy, Annapolis. Astronomic Papers, 1 135-8. U.S. Nautical Almanac Office.

The data comes from experiments undertaken during the months of June and July, 1879, concerned with trying to measure the speed of light. The speed of light measure was recorded 100 times, coming from five different experiments (Expt), each recording measurements (Speed) from 20 different runs (Run).

We are interested to see if the experiments undertaken more than 140 years ago resulted in an average speed of light estimate that is statistically different from the currently reported value from Wikipedia. According to Wikipedia[2], the speed of light is 299,792,458 metres per second.

Note that the *Speed* variable in the **morley** data file represents the recorded speed in *kilometres per second* (km/s) less 299000 km/s. Hence, the relevant comparison value from Wikipedia is $\mu_0 = 792.458$.

Therefore, we wish to test $H_0 : \mu = \mu_0$ vs. $H_1 : \mu \neq \mu_0$, where $\mu_0 = 792.458$ and μ represents the population average measurement from the experimental setup undertaken by Michelson back in 1879. For this exercise, assume that all experiments and runs within experiments, are independent. Therefore we will treat the 100 distinct Speed values, in kilometres per second, as a random sample from this population.

Exercise B1.1: Use the *data()* function to load the datafile into your current **R** session, and save it as a tibble named **ME**, as shown in the code chunk below. Briefly explain the contents of the *ME* datafile in your report.

```
data(morley)
ME <- as_tibble(morley)
```

Exercise B1.2: Produce two different visualisations of the **ME** experimental data, positioned side-by-side. First, produce an estimated density plot (smoothed histogram) of the *Speed* variable, irrespective of the *Experiment* or *Run* number. Then add a vertical red line showing the position of the sample average *Speed*, and a vertical black line showing the corresponding value reported by Wikipedia. In a second visualisation, display a collection of violin plots for the *Speed* variable, according to the relevant *Experiment*[3]. After considering the violin plots, discuss whether you think it is reasonable to treat all 100 runs as being independent observations.[4]

[2] Wikipedia article title "Speed of Light", retrieved 23 August 2020.
[3] "Experiments" is a correction. The first version of this Lab incorrectly stated "Runs".
[4] Regardless, we will treat the runs and experiments as being independent for the rest of the analysis!

```
pB1.2density <- ME %>% ggplot(aes(x = Speed, y = ..density..)) +
    geom_density(fill = "cornsilk") + geom_vline(xintercept = mean(ME$Speed),
    colour = "red") + geom_vline(xintercept = 792.458) + xlab("Speed (km/s)")
pB1.2violin <- ME %>% ggplot(aes(x = as_factor(Expt), y = Speed,
    colour = Run)) + geom_violin() + geom_jitter(width = 0.1,
    height = 0.1) + xlab("Experiment") + ylab("Speed (km/s)")
grid.arrange(pB1.2density, pB1.2violin, ncol = 2)
```

Exercise B1.3: Produce a summary table, based on all 100 observations, containing the average Speed ("mean"), the median Speed ("median"), the standard deviation of the Speed variable ("SD") and the interquartile range of the Speed variable ("IQR"). Save the summary tibble as an **R** object named *sME* before you display it. Interpret each of the summary statistics in your report.

```
sME <- ME %>% summarise(n = n(), mean = mean(Speed), median = median(Speed),
    SD = sd(Speed), IQR = IQR(Speed))
sME %>% kable() %>% kable_styling()
```

Exercise B1.4: Use the *t.test()* function to produce each of the following:
i. a p-value from the approximte CLT-based test, and the conclusion of the test; and
ii. an approximate 95% confidence interval for μ, the population mean Speed of light (in km/s) from the Michelson experiment.

```
ttest1 <- t.test(x = morley$Speed, mu = 792.458) %>% tidy()
ttest1
```

An equivalent alternative command is given by:

```
ttest1.alt <- t.test(Speed ~ 1, mu = 792.458, data = morley) %>%
    tidy()
ttest1.alt
```

Exercise B1.5: Without using the *t.test()* command, produce a summary table that displays the estimate of μ, together with the calculated test statistic, given by

$$T = \frac{\bar{Speed} - \mu_0}{SD_{Speed}/\sqrt{n}},$$

and the corresponding critical value for the two-sided test. The critical value, named *Tcrit* in the summary tibble *s2ME* shown below, is equal to the upper 2.5% quantile value from the t distribution with n-1 degrees of freedom. This *Tcrit* value may be obtained using the **R** function *qt()*, evaluated at *q=0.975* and degrees of freedom *df=n-1*. Also report in your final table, the degrees of freedom *df*, the p-value (named *pvalue*) and the end points of a 95% confidence interval, named *lower* and *upper*, respectively. Discuss whether your results agree with those produced in **Exercise B1.4**.

(Note to avoid a clash with the **MASS** package, we need add the package name **dplyr** to the *select()* function, i.e. use *dplyr::select()*.)

```
s2ME <- sME %>% summarise(n, mean, SD, Tstat = sqrt(n) * (mean -
    792.458)/SD, Tcrit = qt(0.975, n - 1), lower = mean - qt(0.975,
    n - 1) * SD/sqrt(n), upper = mean + qt(0.975, n - 1) * SD/sqrt(n))
s2ME <- s2ME %>% mutate(estimate = mean, df = n - 1, pvalue = 2 *
    pt(Tstat, n - 1, lower.tail = FALSE))
s2ME %>% dplyr::select(estimate, Tstat, Tcrit, df, pvalue, lower,
    upper) %>% kable() %>% kable_styling()
```

B2: CBT: Does the treatment have an effect?

Cognitive behavioural therapy (CBT) is a psychological treatment technique that aims to help a person change their thoughts (cognition) and their behavioural patterns. By learning to replace negative thoughts with more positive ones, and to correspondingly modify negative behaviours with the aim of improving feelings of anxiety and/or depression.

Although CBT has been one of the most important treatments for anxiety and depression over many decades now, new methods for delivering the CBT treatment are regularly sought to try to improve the effectiveness of the general technique.

A recent study was undertaken to assess such a new CBT delivery method. In total, 60 people were recruited to voluntarily participate in the study, all participants having had a recent clinically confirmed episode of anxiety or depression, or both.

Each study participant was asked to complete a certain psychological assessment on two occasions, once before the new CBT delivery treatment method was applied, and once at the end of the treatment. As higher scores on the assessment are associated with an increase in anxiety and depression, it is hoped that the individual scores will be reduced following the new CBT treatment, compared with the corresponding scores obtained at the start of the study.

Each participant's scores on these two assessments are contained in a datafile named "CBT.csv", which is available now on the unit Moodle site. Each of the rows of the CBT.csv data file correspond to one of the sixty (60) subjects who participated in the CBT delivery method experiment. Column 1 corresponds to the individual participant's *case* number, while the values stored in columns 2 and 3 of each row (for the *score1* and *score2* measures) relate to the assessment scores for the participant corresponding to the row case number, with *score1* the score of the assessment completed before the start of the CBT therapy and *score2* the score of the assessment completed following the end of the CBT treatment.

In this example, we will want to test whether the population average of assessment score after the CBT treatement, denoted as μ_2 is the same as the population average assessment score before the treatment, denoted as μ_1. Alternatively, is the difference $\delta = \mu_2 - \mu_1 = 0$? This corresponds to the null hypotheses, given equivalently by either $H_0 : \mu_2 - \mu_1 = 0$ or $H_0 : \delta = 0$. We are interested in exploring the alternative hypothesis given by $H_1 : \mu_2 - \mu_1 \neq 0$ or equivalently, $H_1 : \delta \neq 0$.

Exercise B2.1: Use the following 3 step process outlined below to import and load the **CBT** datafile into into your current **R** session. Then, briefly describe the contents of the *CBT* datafile and confirm why this is a 'paired' data situation.

a. Create a sub-directory folder named **data** in your **R** working directory, if you do not already have one.

b. Download the **CBT.csv** file and put the file in the data sub-directory noted above. (This way if we need to run your code our directory structure will match yours.)

c. Import the **CBT.csv** data into R, to create a tibble named **CBT**, using the following **R** command:

```
CBT <- read_csv("data/CBT.csv")
```

Exercise B2.2: Next we want to produce two different visualisations of the **CBT** data, positioned side-by-side, similar to the plots produced for the **morley** data in Section B1. That is, we want to produce an estimated density plot of the main variable of interest, here the difference defined as *Diff = score2 - score1*. Then add a vertical red line showing the position of the sample average *Diff*, and a vertical black line showing the corresponding *null value* $\mu_0 = 0$. In a second visualisation, display a collection of violin plots for a *score* variable. But before these plots can be easily produced using the *ggplot2* package commands, we'll need to wrangle the data a little. So let's break down the task into two parts, labelled below as **Part B2.2a** and **Part B2.2b.**

Note that the two scores from the pre- and post- treatment measurements (i.e. *score1* and *score2*) appear in CBT in separate columns. This is convenient for calculating the *Diff* variable. This format will also useful for plotting the estimated density plot. However, a "longer" format of the **CBT** tibble is better when creating the violin plots.

Part B2.2a: Insert a code chunk and use the *mutate()* command to create the *Diff* variable in the **CBT** tibble. Then add the required commands to produce the estimated density plot for *Diff*, adding a vertical red line at the sample average value of *Diff* and a black vertical line at zero, corresponding to the "null" value that the CBT treatment does not result in change in the assessment score, on average. Save the plot as an **R** object named *pB2.2density*.

INSERT YOUR CODE CHUNK HERE!

Part B2.2b: Change the wider **CBT** tibble into a *longer* format, using the *tidyr pivot_longer()* command. In this case, you will want to put all of the values of the available "score" variables (*score1* and *score2*) into a single column, named *score*, and the corresponding label of *score1* or *score2* in a variable named *assess* (short for "assessment"). We'll also need to replicate the *case* and *Diff* variables accordingly. Then, produce the required voilin plots for the new *score* variable, with one each for the values of *assess* equal to "score1" and "score2". Save the violin plots as an **R** object named *pB2.2violin*. These steps are shown in the code chunk below. Then, add the command required to print the two objects *pB2.2density* and *pB2.2violin* side-by-side on the same line (as was done for the corresponding **morley** plots).

```
CBT_longer <- CBT %>% pivot_longer(cols = 2:3, names_to = "assess",
    values_to = "score")
pB2.2violin <- CBT_longer %>% ggplot(aes(x = as_factor(assess),
    y = score, colour = assess)) + geom_violin() + geom_jitter(width = 0.1,
    height = 0.1) + xlab("Assessment") + ylab("Score")
grid.arrange(pB2.2density, pB2.2violin, ncol = 2)
```

Exercise B2.3: Produce a summary tibble, based on all $n = 60$ observations of *Diff*, that displays the number of observations n, and the sample mean, median, standard deviation and interquartile range. Save and name your summary tibble as *sCBT*, and display it in your Lab report.

Exercise B2.4: Use the corresponding *t.test()* function to produce each of the following for the test of interest:
i. a p-value from the approximate CLT-based test, and the conclusion of the test; and
ii. an approximate 95% confidence interval for δ, the population average difference in assessment scores, post-CBT treatment less pre-CBT treatment.

```
ttest1 <- t.test(x = CBT$Diff) %>% tidy()
ttest1
```

Exercise B2.5: Without using the *t.test()* command, produce a summary table that displays the estimate of δ, the calculate the test statistic, given by

$$T = \frac{\bar{Diff} - \delta_0}{SD_{Diff}/\sqrt{n}},$$

and the corresponding critical value for the two-sided test. This critical value may be obtained using the **R** function *qt()*, evaluated at *q=0.975* and degrees of freedom *df=n-1*. Also report in your final table, the degrees of freedom *df*, the p-value (named *pvalue*) and the end points of a 95% confidence interval, named *lower* and *upper*, respectively. Discuss whther your results agree with those produced in **Exercise B2.4**.

```
s2CBT <- sCBT %>% summarise(n, df = n - 1, mean, SD, Tstat = sqrt(n) *
    (mean - 0)/SD, Tcrit = qt(0.975, n - 1), lower = mean - qt(0.975,
    n - 1) * SD/sqrt(n), upper = mean + qt(0.975, n - 1) * SD/sqrt(n))
```

```
s2CBT <- s2CBT %>% mutate(pvalue = 2 * pt(abs(Tstat), n - 1,
    lower.tail = FALSE))
s2CBT %>% dplyr::select(Tstat, Tcrit, df, pvalue, lower, upper) %>%
    kable() %>% kable_styling()
```

B3: The MASS::birthwt dataset

Exercise B3.1: Load this datafile from the *MASS* package into your **R** session. Take a look at its contents and review the **R** help information. Here we will focus only on the variables named *bwt* and *smoke*. Briefly explain what these two variables represent in your report.

Exercise B3.2: Produce at least one appropriate visualisation of the *bwt* variable, in relation to the smoking status of the mother during pregnancy. Note you may find the discussion in Section 6.4.2 of the R Graphics Cookbook helpful. You are not required to produce exactly the same type of plots as in **Exercise B1.2** or **Exercise B2.2**, rather produce what you think is of interest.

INSERT YOUR CODE CHUNK HERE!

An easy solution: just copy and paste from the cookbook! But they can do more...

```
birthwt_mod <- birthwt %>% mutate(smoke = as.factor(smoke))
pB3.2density <- ggplot(birthwt_mod, aes(x = bwt, fill = smoke)) +
    geom_density(alpha = 0.3)
pB3.2density
```

Exercise B3.3: Produce a summary tibble that displays the number of observations where the mother smoked *n1*, and the sample mean birthweight of babies born to these mothers (*mean1*), along with the corresponding standard deviation (*SD1*).Save and name your summary tibble as *sBWT1*, and display it in your Lab report. Repeat this for the birthweights of babies born to mothers who did not smoke, resulting in variables *n0*, *mean0* and *SD0*. Save and name this summary tibble as *sBWT0*, and also display it in your Lab report.

The code chunk below will help to get you started. Note that in the **birthwt** datafile, the variable *bwt* is stored as an integer. This needs to be changed to numeric in order to apply the *mean()* and *sd()* functions.

```
sBWT1 <- birthwt %>% filter(smoke == 1) %>% summarise(n1 = n(),
    mean = mean(as.numeric(bwt)), SD1 = sd(as.numeric(bwt)))
sBWT0 <- birthwt %>% filter(smoke == 0) %>% summarise(n1 = n(),
    mean = mean(as.numeric(bwt)), SD1 = sd(as.numeric(bwt)))
sBWT1 %>% kable() %>% kable_styling()
sBWT0 %>% kable() %>% kable_styling()
```

Exercise B3.4: For this setting, the test of interest is

$$H_0 : \mu_{smoke1} = \mu_{smoke0} \quad \text{vs.} \quad H_1 : \mu_{smoke1} \neq \mu_{smoke0},$$

or alternatively,

$$H_0 : \mu_{smoke1} - \mu_{smoke0} = 0 \quad \text{vs.} \quad H_1 : \mu_{smoke1} - \mu_{smoke0} \neq 0.$$

In this case, μ_{smoke1} and μ_{smoke0} are the population mean birthweight of babies whose mothers smoke, or do not smoke, during pregnancy, respectively.

Use the corresponding *t.test()* function to produce each of the following for the test of interest:
i. a p-value from the approximate CLT-based test[5], and the conclusion of the test; and
ii. an approximate 95% confidence interval for $\delta = \mu_{smoke1} - \mu_{smoke0}$. Interpret the output in the context of the setting.

The code chunk below will get you started!

```
bws1 <- birthwt %>% filter(smoke == 1) %>% pull(bwt)
bws1 <- as.numeric(bws1)
```

```
bws0 <- birthwt %>% filter(smoke == 0) %>% pull(bwt)
bws0 <- as.numeric(bws0)
ttest1 <- t.test(x = bws1, y = bws0) %>% tidy()
ttest1
```

Exercise B3.5 (optional): If you are interested, you can attempt to manually calculate the t-test and 95% confidence interval for δ, but note that the formula for the degrees of freedom identified in the lecture videos is not the one computed by the *t.test()* function.

[5]Consider the *t.test()* option "var.equal". If you are unsure of the most appropriate value to use, you can try both and see if your conculsion changes.