Practicing T-tests in R

Part 1: Data Prep & 1-sample tests

Nathan Brouwer brouwern@gmail.com @lobrowR

October, 2017

Introduction

This this lab we will practice loading data on human body temperature and heart rate in order to understand t-tests and paired t-tests. We'll focus also on creating different plots to explore that data. Its split into 2 parts. In the 1st part we'll load the data, plot it, and carry out a 1-sample t-test. In the 2nd part we'll do a 2-sample t-test and a paired t-test. A final optional section explores scatterplots and regression.

A note on optional parts of lab

This lab was originally written using only "base R" functions and did not use ggplot, ggpubr, or dplyr. Sections marked as **"OPTIONAL"** and/or **"OLD SCHOOL"** show alternative ways to do things and **can be skipped**. They are useful to look over so you can see other ways that you might encounter people doing things in R; however, they are no longer the easiest, most consistent, or efficient ways to do our tasks for this lab.

Outline of this lab

- Part 1: Setup, visualization & 1-sample t-test
  - Load data from a .csv file
  - Clean data
  - Visualize data
  - Question 1: What is the mean human body temp?
- Part 2: 2-sample t-test & paired t-test
  - Question 2: Is the mean human body statistically different from 98.6?
  - Question 3: How do Male & Female Body Temps compare?
  - Question 4: How is a paired t-test similar to a 1-sample t-test?
  - OPTIONAL: Question 5: What is the relationship between resting heart rate & body temperature

Commands & arguments used

Major commands used: * read.csv() * gghistogram() from the ggplotr packge * sd() * c() * ggboxplot() * ggscatter() * t.test()

References

Original data source:

Shoemaker, A. 1996. What's Normal? – Temperature, Gender, and Heart Rate. Journal of Statistics Education v.4, n.2 ww2.amstat.org/publications/jse/v4n2/datasets.shoemaker.html

Original inspiration

Journal of the American Medical Association entitled "A Critical Appraisal of 98.6 Degrees F, the Upper Limit of the Normal Body Temperature, and Other Legacies of Carl Reinhold August Wunderlich" (Mackowiak, Wasserman, and Levine 1992).

These data are covered in example 11.3 on page 310 in Whitlock and Shulter, 2nd ed.

Part 1: Setup, visualization & 1-sample t-test

Load packages

We will use the ggpubr add on to the ggplot2 package for plotting, and dplyr for data cleaning. (Any red text that pops up can be ignored)

```
library(ggplot2) #plotting tools
library(ggpubr)
library(dplyr)  #data cleaning tools
```

Load Body Temp data

Approach 1: Loading data with RStudios menus

You can follow instructions at to use RStudio's built-in tools for loading data here https://support.rstudio.com/hc/en-us/articles/218611977-Importing-data-with-RStudio

Depending on your version of RStudio there might be a way to do this from the "Tools" section of the drop down menu. Another way is outlined below using the Environment/History/Connections pane.

The basic steps are

- Save the file "bodytemp.csv" to a convenient location
- Locate the panel in RStudio that says "Environment/History/Connections"
- It might be minimized so only those words are visible at the bottom the screen (It might say " Environment/History/Connections/Git" too)
- Enlarge it by clicking on the standard "make it bigger" tab.
- There will be several icons: file folder, disk, spreadsheet with "Import data set" and broom; click on "Import dataset"
- For this lab our data set is in .csv form so we select the top option "From text (base)"
- We can then navigate to where our file is located and select "open"
- A busy screen then opens up; everything should be in order to just click "Import"
- R studio will probably open up a preview of the data in your script viewer.

- You can then explore the data using head(), tail(), dim(), summary(), etc

OPTIONAL: Approach 2: using code

This approach isn't necessary if you have already loaded it as shown above.

- Save the file "bodytemp.csv" to a convenient location
- Set your "working directory" to whee bodytemp.csv is located
- Load the data from a .csv file sing read.csv()

```
    bodytemp <- read.csv("bodytemp.csv")
```

IN CASE OF PROBLEMS: Approach 3: Building data by hand

NOTE: if you have trouble loading the data, go to the very end of this page; there is code to replicate the data directly in R w/o loading a .csv file. You can copy and paste from the pdf directly into a script file to make the data set.

data Exploration

Look at the Body Temp bodytemp

```
#size of data
dim(bodytemp)

#first few rows
head(bodytemp)

#last few rows
tail(bodytemp)
```

Look at overall summary

```
summary(bodytemp)
```

Data cleaning:

Convert "1" to "male"" & "2" to "female""

The original file has male = 1 and female = 2. Let's change these to letters. We'll use the recode() function in the dplyr package. As an optional example, you can see how to do it with the base R command ifelse()

Gender is coded as 1 or 2. R is treating this as a numeric variable.

```
summary(bodytemp$gender)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     1.0     1.5     1.5     2.0     2.0
```

We can see that it only takes on two values by "nesting" the command factor() within summary(). There are 65 "s and 65 2s.

```
summary(factor(bodytemp$gender))
```

```
##  1  2
## 65 65
```

Load dplyr if you haven't already. (any red text that pops up can be ignored)

```
library(dplyr)
```

Now we'll "decode" the 1s and 2s in the gender columns.

```
gender.MF <- recode(bodytemp$gender,
                    "1" = "male",
                    "2" = "female")
```

Look at the new column

```
summary(gender.MF)
```

We have "character" data right now. We can covert it to a categorical aka factor variable using factor()

```
#convert w/ factor()
gender.MF <- factor(gender.MF)

#check w/ summary
summary(gender.MF)
```

```
## female   male
##     65     65
```

OPTIONAL: the as.factor() function

Sometimes you will see the function as.factor() used. In this context, it does the same thing as factor

```
        #convert w/ factor()
        gender.MF <- as.factor(gender.MF)

        #check w/ summary
        summary(gender.MF)
```

```
## female   male
##     65     65
```

OPTIONAL: recode using ifelse()

Here is an alternative approach using the **ifelse()** command. What we'll do is have R look at the gender column in the bodytemp data and if the value equals 1, change it to "male", else change it to "female". So, if its a 1, change it to male, if it its not a 1, it must be a 2 so change it to female. We'll put these changes into a new data object.

```
        gender.MF <- ifelse(bodytemp$gender == 1,"male","female")
```

Note that we use == to set up the logical argument.

Add new column to existing data

Overwrite the old dataframe with a new version that has our recode gender variable

```
#add new object to existing data
bodytemp <- data.frame(bodytemp, gender.MF)
```

Look sat the new data

```
summary(bodytemp)
```

```
##       temp            gender       heartrate      gender.MF
##  Min.   : 96.30   Min.   :1.0   Min.   :57.00   female:65
##  1st Qu.: 97.80   1st Qu.:1.0   1st Qu.:69.00   male  :65
##  Median : 98.30   Median :1.5   Median :74.00
##  Mean   : 98.25   Mean   :1.5   Mean   :73.76
##  3rd Qu.: 98.70   3rd Qu.:2.0   3rd Qu.:79.00
##  Max.   :100.80   Max.   :2.0   Max.   :89.00
```

(Note: R will actually automatically makes the gender.MF column into categorical bodytemp if we had skipped that factor() step).

Data Exploration

Question 1: What is the mean human body temp?

Conventional wisdom is that "normal" body temp is **98.6** degrees F when taken with an oral thermometer. Eg, via Wikipedia "Normal human body temperature, also known as normothermia or euthermia, is the typical temperature range found in humans. The normal human body temperature range is typically stated as 36.5–37.5 °C" https://en.wikipedia.org/wiki/Human_body_temperature

Is this true for our sample of 130 men and women? We'll calculate the mean, then plot the distribution of the data and overlay the mean.

```
summary(bodytemp)
```

```
##       temp            gender      heartrate      gender.MF
##  Min.   : 96.30   Min.   :1.0   Min.   :57.00   female:65
##  1st Qu.: 97.80   1st Qu.:1.0   1st Qu.:69.00   male  :65
##  Median : 98.30   Median :1.5   Median :74.00
##  Mean   : 98.25   Mean   :1.5   Mean   :73.76
##  3rd Qu.: 98.70   3rd Qu.:2.0   3rd Qu.:79.00
##  Max.   :100.80   Max.   :2.0   Max.   :89.00
```

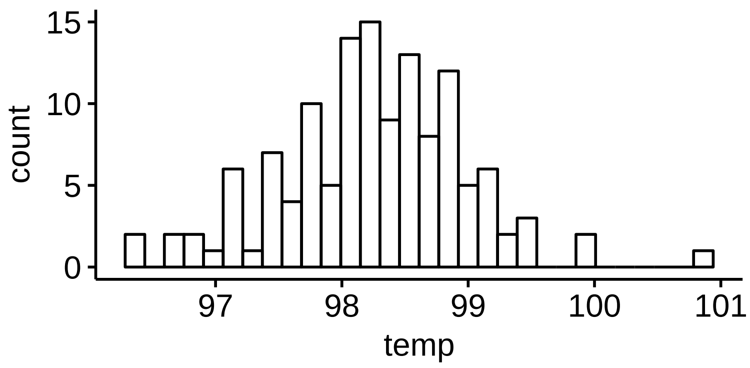
The mean and the median are both about 0.3 degrees *below* **98.6.**

Let's look at the distribution of the data to get a sense for what's going one.

Graph 1: histogram

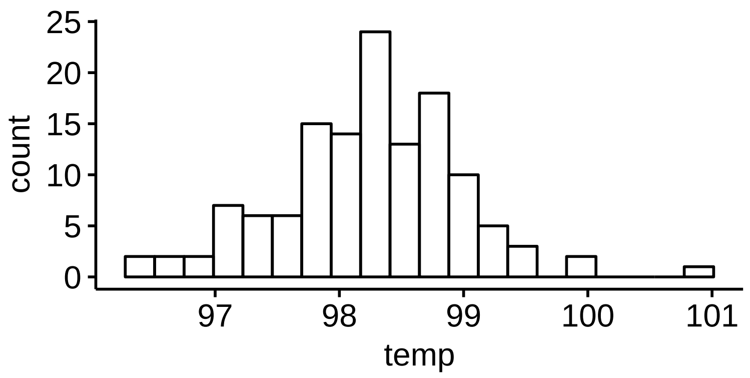
- We can make a histogram using the gghistogram() function from ggpubr.

```
#Load libraries if you haven't already
#library(ggplot)
#library(ggpubr)

gghistogram(bodytemp,
            x = "temp")
```



The default is a bit boxy looking, so we can change the look by setting "bins = 20". This looks a little smoother. Note the outlier at around 101 degrees.

```r
gghistogram(bodytemp,
            x = "temp",
            bins = 20)
```



We can add some more information to the plot using gghistogram; add = "mean" shows us where the mean is located. (we could do "median" if we wanted)

```r
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "median")
```

OPTIONAL: OLD SCHOOL: the hist() command

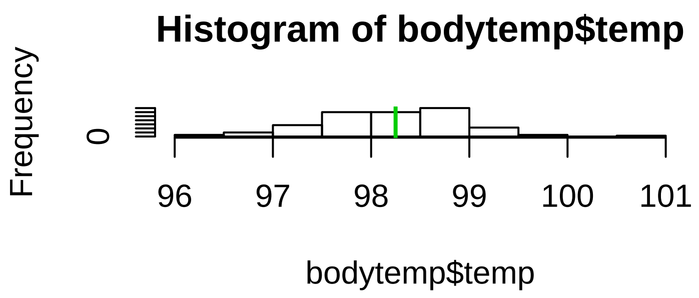
For historical reference, here's how this would be made using R's original base plotting functions

- We can add a line for the **mean** (y.hat) using the command **abline()**.

- We give **abline** the argument **v=** to tell it to make a vertical line.
- We also pass abline the argument **col="green"** to change the color to green
- the name of the color must be in quotes. col = green will not work.
- Finally, we pass abline the argument "lwd=2" to increase the size of the line.

- **lwd** means "line width"

```r
## OPTIONAL
      #1st, Calculate the mean and store into an R object
      overall.mean <- mean(bodytemp$temp)

      #make the histogram of the $temp column
      hist(bodytemp$temp)

      #add a line for the mean
      abline(v = overall.mean, col = 3, lwd = 2)
```

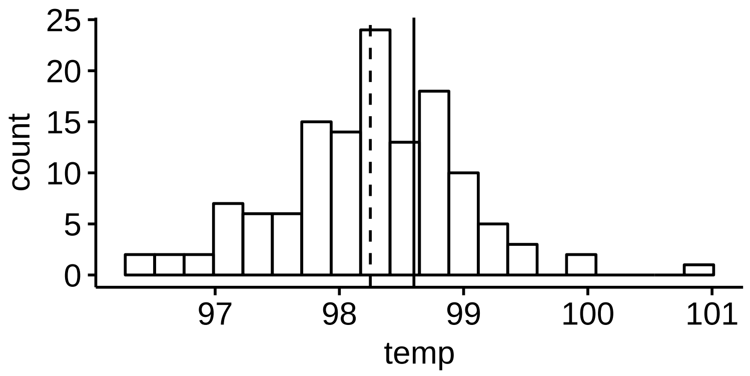


Graph 2:histogram with more details

Let's add some information to the graph. We'll do this using the geom_vline() function (vline = vertical line)

First, let's add a reference line for the assumed mean of 98.6 Don't forget the "+" after gghistogram()!

```r
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "mean") +
  geom_vline(xintercept =  98.6)
```



Note that 98.6 is **higher** than the mean and the median.

We can change the color and size of the line at 98.6

```r
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "mean") +
  geom_vline(xintercept =  98.6,
             color = "red",
             size = 2)
```

Next, let's plot where +/- 1 standard deviation is. We'll need the mean and the SD

```r
#Calculate the mean
overall.mean <- mean(bodytemp$temp)

#Calculate the standard deviation
```

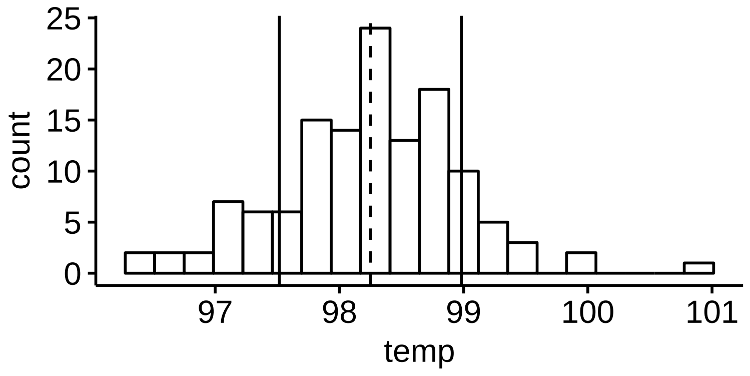
```
overall.sd <- sd(bodytemp$temp)
```

Plot the SD on the histogram using geom_vline().
We'll plot

- mean + SD, and
- mean - SD

Don't forget the "+" after each geom_vline() statement!

```
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "mean") +
  geom_vline(xintercept = overall.mean + overall.sd) +
  geom_vline(xintercept = overall.mean - overall.sd)
```



We can easily change the colors of these reference lines

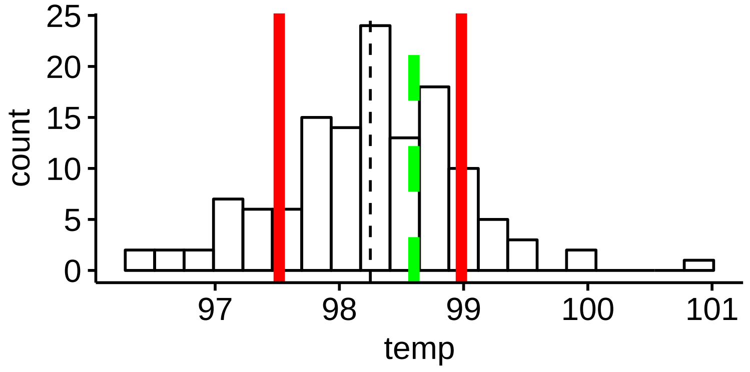
```
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "mean") +
  geom_vline(xintercept = overall.mean + overall.sd,
             color = "red") +
  geom_vline(xintercept = overall.mean - overall.sd,
             color = "Red")
```

And their thickness

```
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "mean") +
  geom_vline(xintercept = overall.mean + overall.sd,
             color = "red",
             size = 2) +
  geom_vline(xintercept = overall.mean - overall.sd,
             color = "red",
             size = 2)
```

Now add the line for 98.6 degrees F We can make it dashed using "linetype = 2". Don't forget the "+" after each geom_vline() statement.

```
gghistogram(bodytemp,
            x = "temp",
            bins = 20,
            add = "mean") +
  geom_vline(xintercept = overall.mean + overall.sd,
             color = "red",
             size = 2) +
  geom_vline(xintercept = overall.mean - overall.sd,
             color = "red",
             size = 2) +
  geom_vline(xintercept = 98.6,
             color = "green",
             size = 2,
             linetype = 2)
```



Conclusion: is the body temp = 98.6?

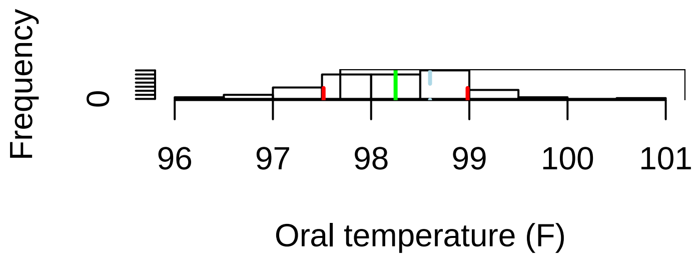
In our sample, the body temp is 98.25. This is less than the assumed mean of 98.6. 98.6 however is well within +/- 1 SD of our mean.

OPTIONAL: OLD SCHOOL: Graph 2: histogram with more details

Again, this is all optional and just for reference.

First, calculate sum summary stats and save them into R objects.

```
## OPTIONAL
      #Calculate the mean
      overall.mean <- mean(bodytemp$temp)

      #Calculate the standard deviation
      overall.sd <- sd(bodytemp$temp)

      #make the histogram of the $temp column
      hist(bodytemp$temp, xlab = "Oral temperature (F)",main = "")

      #add a green line for the mean
      abline(v = overall.mean, col = "green", lwd = 2)

      # Add lines for +/- 1 SD, make the color RED
      ##plus 1 SD.  Note that I add the SD to the mean
      abline(v = overall.mean+overall.sd, col = "red", lwd = 2, lty = 2)

      ##minus 1 SD Note that I subtract the SD from the mean
      abline(v = overall.mean-overall.sd, col = 2, lwd = 2, lty = 2)

      #Add a line for where 98.6 is
      abline(v = 98.6, col = "lightblue", lwd = 2, lty = 4)

      #Add a legend
      legend.text <- c("mean","median","+/- 1 SD","98.6 degrees F")
      colors.used <- c("green","blue","red","lightblue")
      legend("topright", legend = legend.text,  #what each line is
                 col = colors.used,      #what each color is
                 lty = c(1,2,2,4),       #the type of dashs used
                 lwd = 2)                #the width of the line
```



Question 2: Is the average human body statistically different from 98.6?

The mean and median of our sample are both less than the conventional value of 98.6. But could this just be due to **sampling variation**? We can test the hypothesis that this difference is not due to chance using a one-sample t-test. This is a bit different than the standard t-test used to compare two groups, but is a great set up for a paired t-test.

One-sample t-test

- We will test the hypothesis that the mean of this sample of data is different than the conventional wisdom that the "normal" human temperature is **98.6 degrees F.**

- The **null hypothesis (Ho)** is that the mean of this sample equals 98.6
- Recall that **"mu"** is the population mean (aka the population parameter).

First, we make an object that contains our value for mu, 98.6. Then we carry out a one-sample t-test.

```
normal.temp <- 98.6
```

Now conduct the t.test. Note that there is no "~" to say "this relates to that". Instead, we declare "mu = normal.temp", which tells R what our hypothesis is.

```
t.test(bodytemp$temp,
      mu = normal.temp)
```

```
##
##  One Sample t-test
##
## data:  bodytemp$temp
## t = -5.4548, df = 129, p-value = 2.411e-07
## alternative hypothesis: true mean is not equal to 98.6
## 95 percent confidence interval:
##  98.12200 98.37646
## sample estimates:
## mean of x
##  98.24923
```

Visualizing the output of the 1-sample t-test

Visualizes the effect size.

Evaluate the results

The p value is MUCH lower than the traditional cutoff of **0.05**.

This 1-sample t-test indicates that the mean temperature of this sample is less than 98.6 and the difference is very unlikely to be due to chance.

The most plausible range of values for the true, biologically determined population mean (mu) is between **98.12200** and **98.37646**, as defined by the 95% confidence interval (CI). The traditionally assumed value of 98.6 is therefore a very implausible value!

Note, however, than many people did have a higher temp than 98.6; there is significant variation in the body temp of healthy individual.