Two-Sample T Tests in R

Hannah Waddel

Steps for a Two-Sample T Test

Recall that a two-sample T test is used to compare the mean of a continuous outcome across two groups (categories). The general strategy for a two-sample T test is as follows:

1. Perform descriptive analyses for each variable (both the continuous outcome variable and the categorical group variable)

- In SAS, you could use PROC UNIVARIATE or PROC FREQ. R has its own way to perform the same functions.

2. Assess whether the required conditions for the two-sample t-tests are met:

- Are both sample sizes "large"? (≥ 30?)
- If not, are both samples approximately normal?

3. Choose between the pooled (equal variance) and unpooled (unequal variance/Satterthwaite) two-sample t-tests.

- If *a priori* (before seeing the data) it can be justified that the population variances are equal, use the pooled t-test
- If *a priori* it can be justified that the population variances are unequal, use the unpooled t-test.
- If prior information about equality of population variances is not available, then either
  - Perform the F-test for equality of variances (if data are *known* to be normally distributed)
  - Choose the t-test with the larger p-value, to be more conservative (less likely to reject H_0)

4. After these steps are done, perform the appropriate t-test, report your p-value, and make a conclusion.

Reading in the Data

For this example, our research question is the following: are the mean ages of male and female stroke patients the same?

Our null and alternative hypotheses are thus:

$$H_0 : \mu_1 = \mu_2$$
$$H_A : \mu_1 \neq \mu_2$$

Where:

μ_1 = Mean age of male stroke patients

μ_2 = Mean age of female stroke patients

We are going to use a dataset from a stroke clinical trial to answer the question. The data are saved as **"strokestudy.csv"**.

Download the data and save them to your computer. I recommend creating a folder for this class, and the R examples.

To import and export data, we first need to set our working directory using the **setwd()** function. The working directory is similar to the libname statement in SAS–it sets the default location of any files you import or export from R.

In RStudio, you can select Session > Set Working Directory > Choose Directory... and select the folder where you saved your .csv file. You can save the line of code that changed the working directory from the "history" tab in RStudio, and insert it back into your R script, by clicking on the **setwd()** code that appears in the history and clicking "to source" at the top of the history.

On my own computer, the file is stored in the directory "Documents\Work\Bios 591P 2022\R Materials\1 Introduction\Two Sample T Test" folder. The following piece of code sets the correct working directory.

```
setwd("~/Documents/Work/Bios 591P 2022/R Materials/1 Introduction/Two Sample T Test")
```

Now, we have set up the working directory. R will import and export datasets from this folder.

To read in the data, we will use the **read.csv()** function in R. The argument to this function is a *string* (text with quotation marks around it) with the name of your .csv file. Make sure that the text in the string exactly matches the name and extension (.csv, the file type) of your data file!

Using the **<-** operator, we save the data to the name *strokestudy*. Once we do this, we can reference the dataset by this name.

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

If you want to see the data you have just read in, either run:

```
print(strokestudy)
```

to print out the dataset in the console. Or

```
View(strokestudy)
```

to view the data in the viewer window.

NOTE: What happens when you run the following code instead of saving the data as strokestudy?

```
read.csv("strokestudy.csv")
```

Perform descriptive statistics

Now that we have read in the data, we can perform a descriptive analysis to get a better sense of the variables of interest.

In SAS, we could use PROC CONTENTS to get a sense of how many observations and variables we have in our dataset, and the variable names in the dataset. Let's do something similar in R.

The first function we will use is **dim()**. It takes a dataset as its argument, and returns the number of rows and columns in the dataset

```
dim(strokestudy)
```

```
## [1] 323   6
```

We have 323 observations (rows) in the dataset, and 6 variables (columns).

To get the variable names, we use the **names()** function.

```
names(strokestudy)
```

```
## [1] "Patient" "arm"     "age"     "gender"  "SBP"     "ldl"
```

We see that the names of the variables are Patient, arm, age, gender, SBP, and ldl.

Our categorical variable of interest is gender, and our continuous variable is age.

Let's see how many males and females we have in the study. In R we will use the **table()** function.

We must specify which variable we are interested in tallying in the dataset. To extract the gender variable from the strokestudy dataset, we use the $ operator. First, see what happens when you simply type

```
strokestudy$gender
```

into the console. Observe that you get a list of all the gender values for each patient, in order of patients.

Now, when we run the code

```
table(strokestudy$gender)
```

```
##
##   1   2
## 194 129
```

From the output, we can see that we have 194 patients with gender 1 (male) and 129 patients with gender 2 (female). This is much higher than the threshold of 30 patients in each group, so the Central Limit Theorem applies and we can use the two-sample T test.

If there had been less than 30 patients, and we wanted to do some analysis of the continous outcome (age), we could use the following code to do something similar to PROC UNIVARIATE. We use the functions **mean()**, **var()**, **median()** to get the mean, variance, and median of the age in the sample.

```
mean(strokestudy$age)
```

```
## [1] 60.95046
```

```
var(strokestudy$age)
```

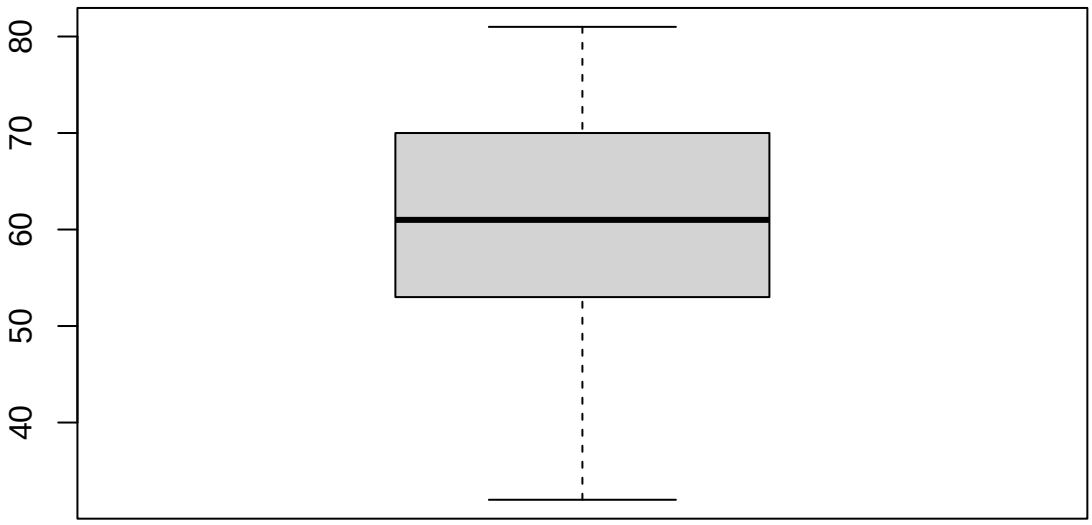
```
## [1] 124.2149
```

```
median(strokestudy$age)
```

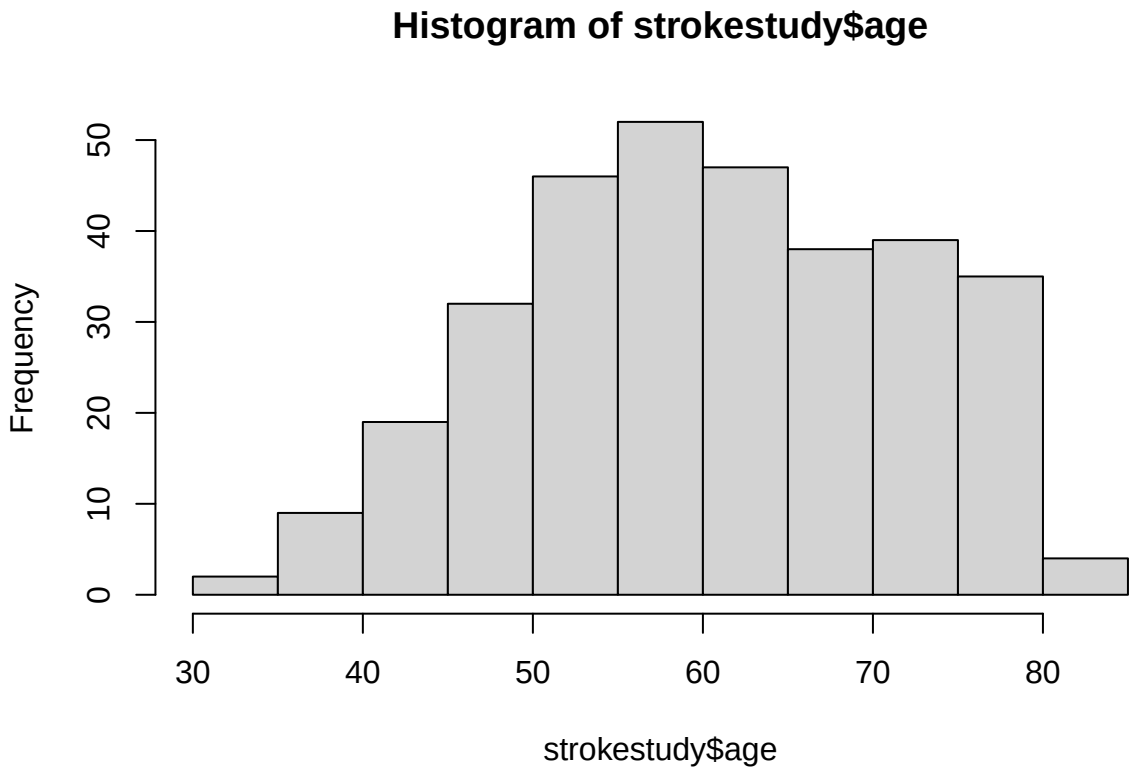
```
## [1] 61
```

We can also use the functions **boxplot()** and **hist()** to get a sense of the distribution of the age variable.

```
boxplot(strokestudy$age)
```



```
hist(strokestudy$age)
```

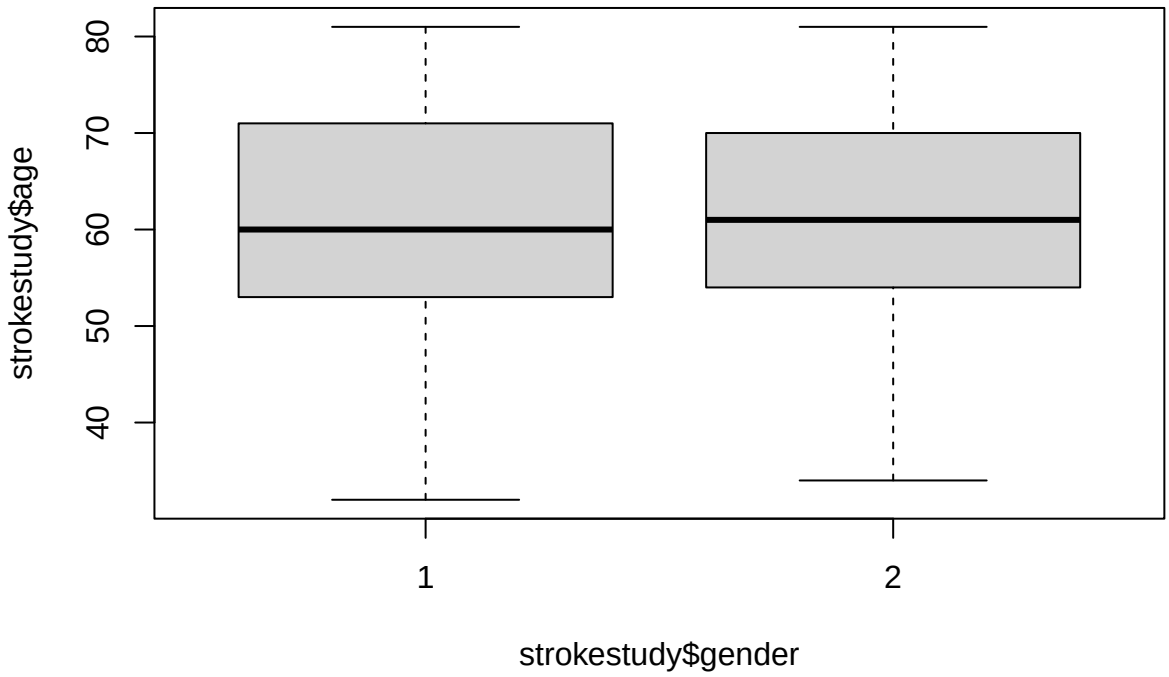


The plots are a bit rough right now, but we will learn ways to customize graphs later.

Suppose we want to split the graphs of age by gender? To do this, we use the ~ operator. This operator will appear again when we are doing linear modeling. Our *dependent* variable go on the left side of the ~, and *independent* variable on the right side. Our dependent variable, or outcome, is age, and our independent variable is gender.

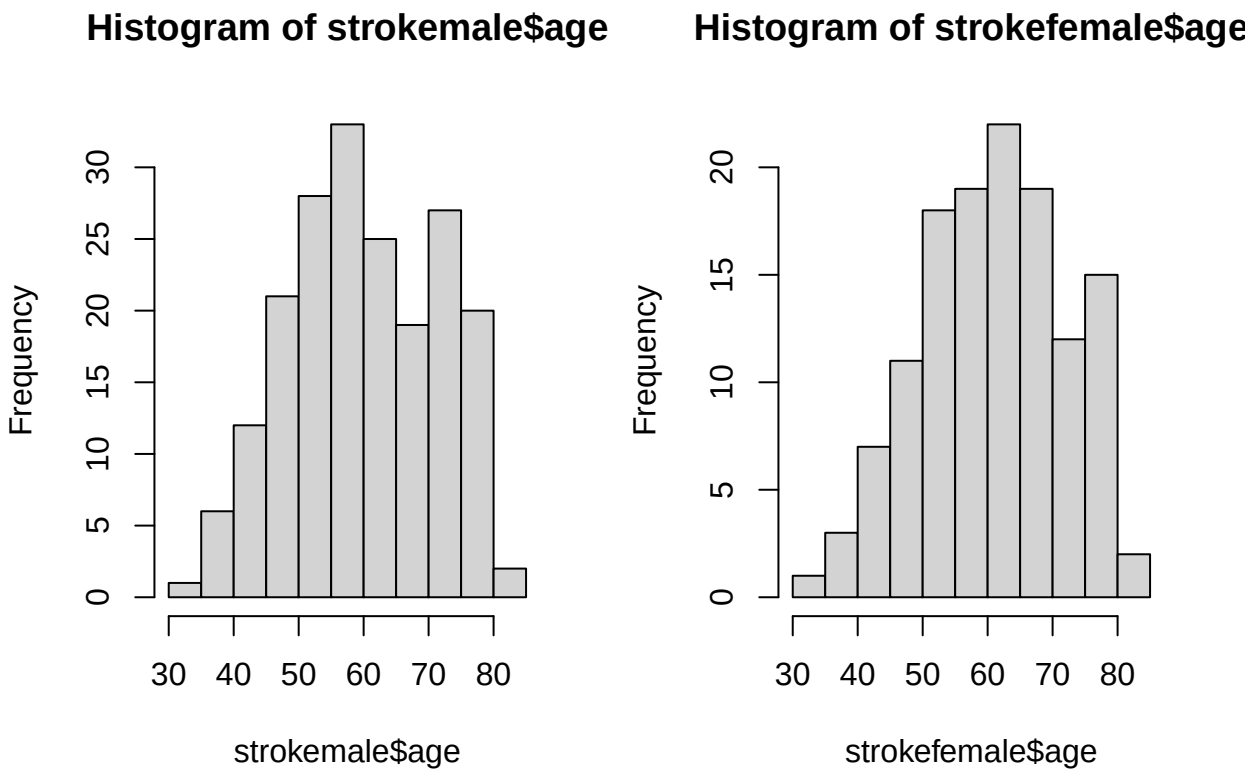
So, to split age into boxplots by gender, we now use the following syntax since we are using two variables in the graph:

```
boxplot(strokestudy$age ~ strokestudy$gender)
```



Creating a divided histogram is more complicated for an introductory R lesson, and it is complicated without installing more packages. I will demonstrate the code, but we will go into further detail in a later lesson, when we cover plots and graphs specifically.

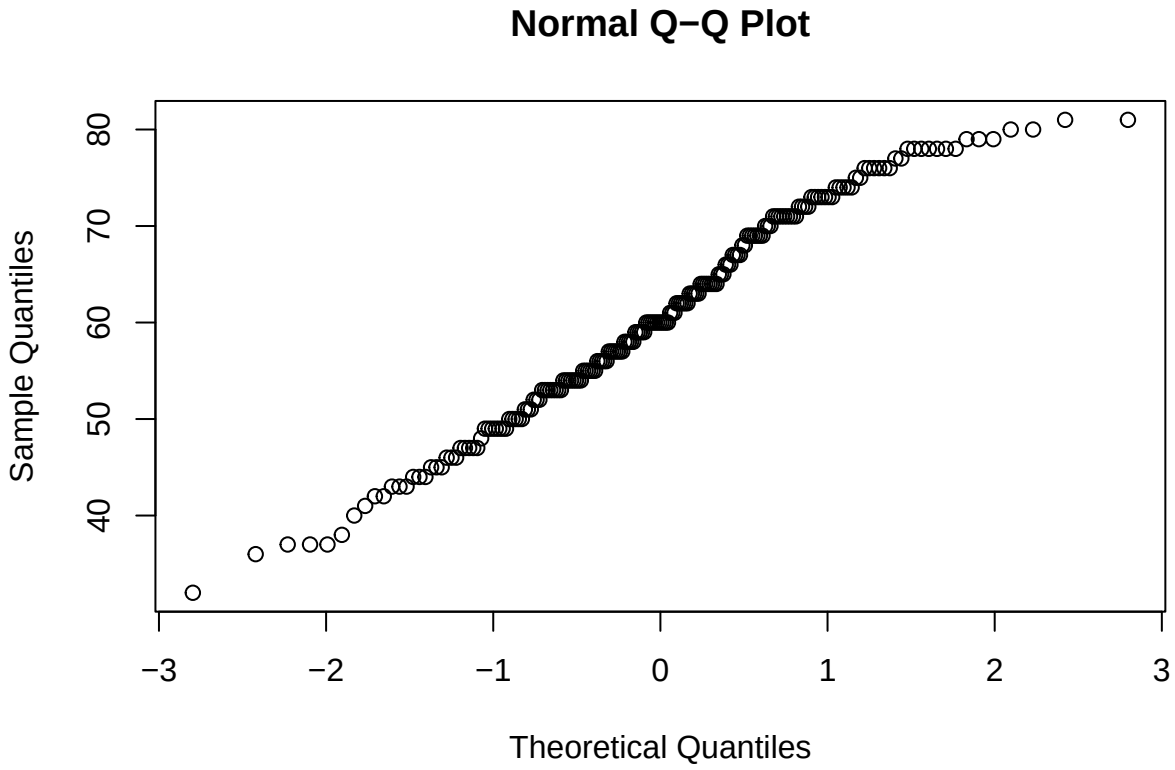
```
par(mfrow=c(1,2)) #Setting the graph area to include two graphs, in 1 row and 2 columns

strokemale <- strokestudy[which(strokestudy$gender==1),] #Copying the male patients to a new dataset
hist(strokemale$age) #histogram of male stroke patients age

strokefemale <- strokestudy[which(strokestudy$gender==2),] #Copying the female patients to a new datase
hist(strokefemale$age) #histogram of female stroke patients age
```



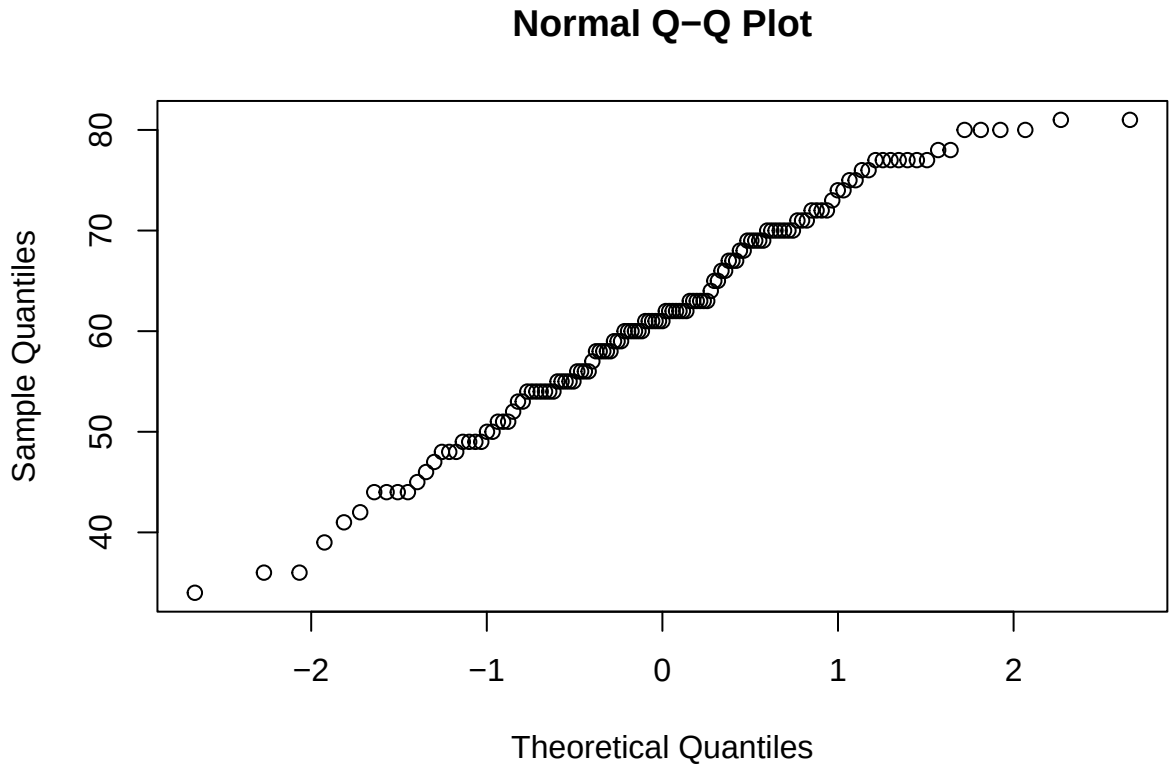

```
par(mfrow=c(1,1)) #Setting the graph area to only include 1 plot
```

We will also use the new datasets we created for males and females to assess normality with a Q-Q plot. When you give a numeric variable to the **qqnorm()** function, R will return the Q-Q plot. Recall that a linear Q-Q plot indicates normality in the variable.

```
qqnorm(strokemale$age)
```

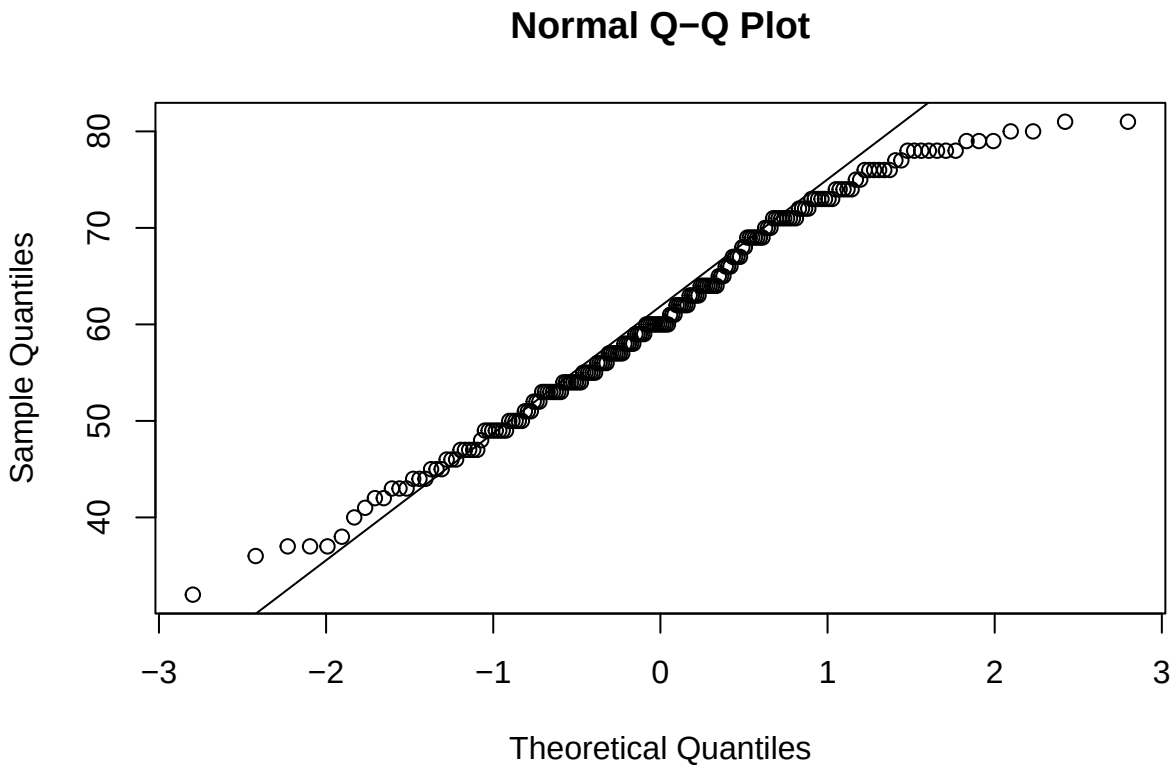


```
qqnorm(strokefemale$age)
```

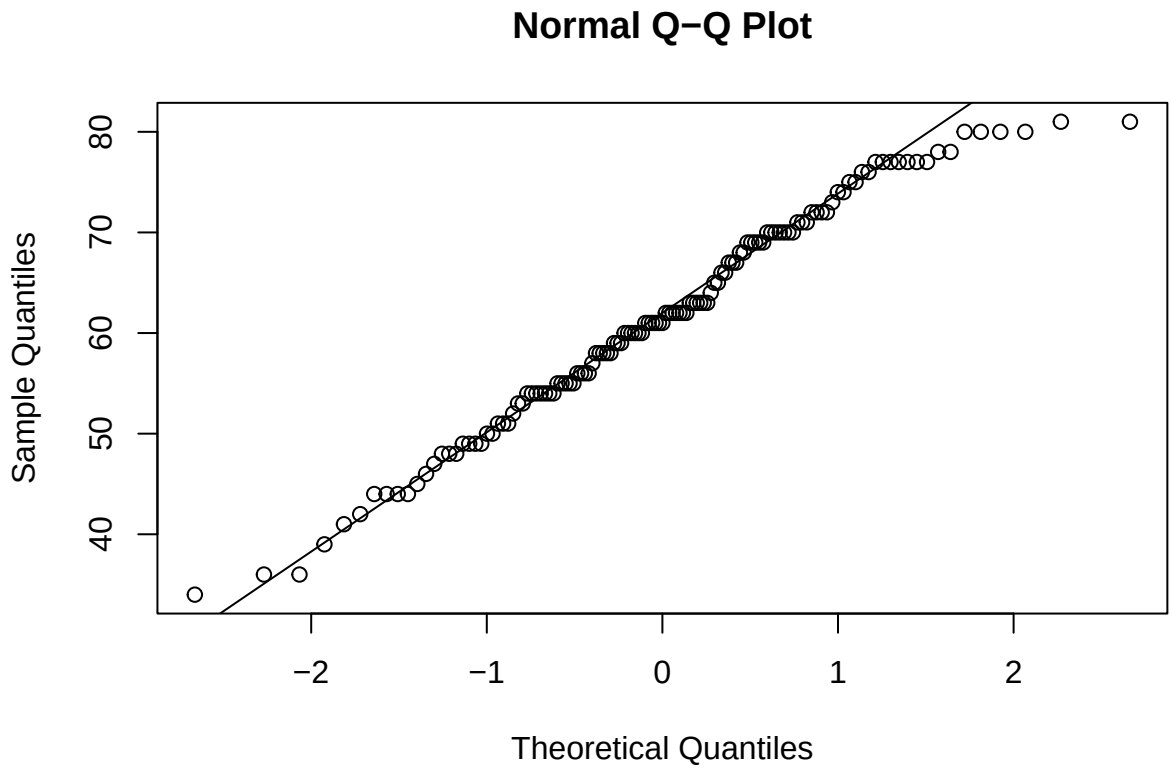


If we want to draw the Q-Q line on top of the plot, we can use the **qqline()** function. However, the **qqline()** function does not initialize a new plot in R, so you will get an error if you run **qqline()** without first running **qqnorm()**.

```
qqnorm(strokemale$age)
qqline(strokemale$age)
```



```
qqnorm(strokefemale$age)
qqline(strokefemale$age)
```



Looking at the histograms and Q-Q plots, we see that the data are approximately normal.

Select Pooled vs. Unpooled T Test

Now that we have decided to move forward with the two-sample T test, we must decide whether to use the pooled or unpooled T test.

We do not have any prior information on whether or not the population variance of age is equal between male and female stroke patients.

Our data are also *approximately* normally distributed, and recall that the F test for equal variances requires our data to be normally distributed, not approximately normal. Thus, we will use the conservative approach and select the T test which has the highest p-value.

It is possible to conduct the F test in R, and the details for that are at the end of this document.

Perform the T-Test

Now we test our hypothesis. To do this, we use the **t.test()** function in R. Note that the **t.test()** function requires us to use the ~ operator again.

```
t.test(strokestudy$age ~ strokestudy$gender)
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  strokestudy$age by strokestudy$gender
## t = -0.69971, df = 278.56, p-value = 0.4847
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  -3.365961  1.600597
## sample estimates:
## mean in group 1 mean in group 2
##        60.59794        61.48062
```

The output includes our T statistic (t =), our degrees of freedom (df =), and our p-value. It tells us what the alternative hypothesis was, and gives a 95% confidence interval for the difference in means. It also gives us the estimates of the mean in the male and female groups.

By default, **t.test()** uses the unpooled (Satterthwaite/Welch) T test. However, we can specify whether we use the pooled or unpooled test. **t.test()** also uses a two-sided T test, by default, but you can change this.

Let's try the pooled T test. To make it pooled, we set the function argument "var.equal=TRUE"

```
t.test(strokestudy$age ~ strokestudy$gender,var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  strokestudy$age by strokestudy$gender
## t = -0.69657, df = 321, p-value = 0.4866
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  -3.375718  1.610354
## sample estimates:
## mean in group 1 mean in group 2
##        60.59794        61.48062
```

Comparing the p-value between the pooled and unpooled T tests, we see that they are similar, but that the pooled variance T test has a slightly higher p-value. Since we are taking the conservative approach, we use the pooled T test to make our conclusion.

To make our conclusion, we make our decision to reject or fail to reject H_0, state our p-value, and put our conclusion back in the original context of the research question.

"Fail to reject H_0. With $p = 0.4866$, which is greater than $\alpha = 0.05$, there is not evidence to suggest that male and female stroke patients have a different mean age."

Appendix: F Test for Equality of Variances

Assuming that our data were normally distributed (though they were only approximately normal for this example), we can conduct the F Test for Equality of Variances. In order to do this, we use the **var.test()** function.

The null and alternative hypotheses for the F test are as follows:

$$H_0 : \sigma_1^2 = \sigma_2^2$$

$$H_A : \sigma_1^2 \neq \sigma_2^2$$

Where:

σ_1^2 = Variance of age for males

σ_2^2 = Variance of age for females

We use the ~ operator again to divide patients by gender.

```
var.test(strokestudy$age ~ strokestudy$gender)
```

```
##
##  F test to compare two variances
##
## data:  strokestudy$age by strokestudy$gender
## F = 1.0456, num df = 193, denom df = 128, p-value = 0.7912
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7574836 1.4284400
## sample estimates:
## ratio of variances
##           1.045579
```

Our conclusion in this case is:

"Fail to reject H_0. With p = 0.7912, there is not evidence to suggest that the variance of age is different between males and females".

Then, in keeping with this conclusion, we would use the pooled T Test.