You will need the following file for this lab:

- Lab10.csv

The file is a comma separated file and you can open it using a text editor such as Notepad to view its data.

The data contains systolic blood pressure for mice with two different genotypes for the D4Mit214 marker. We want to investigate if the systolic blood pressure may be taken to be independent of the genotype. We can attempt to test this by testing if the mean of the two groups are the same. This can be done in R with the function `t.test()`.

Question 1

(2/2 points)
Read the csv file into R as a data frame called `my.data`.

How many rows (observations) does the data frame have?

- 20
- 30
- 40
- 50 ✔

How many columns (variables) does the data frame have?

- 1
- 2 ✔
- 3
- 4

EXPLANATION

Question 2

(1/1 point)

Use the `t.test()` function to derive a test statistic and p-value for testing that the mean in the two groups are identical. Before you do so, create a data frame for each genotype. Which two commands you should use?

- [ ] data1<-my.data$systolic.bp[my.data$Genotype=="AA"]
- [x] data1<-my.data$systolic.bp[my.data$Genotype=="BA"] ✔
- [x] data2<-my.data$systolic.bp[my.data$Genotype=="BB"] ✔
- [ ] data2<-my.data$systolic.bp[my.data$Genotype=="AB"]

✔

EXPLANATION

Question 3

(2/2 points)

Use the `t.test()` function to derive a test statistic and p-value by using the following command.

```
testResult <- t.test(data1,data2)
```

What is the value of statistic element?

- (●) -2.027021 ✔
- ( ) 2.027021
- ( ) -0.04824252
- ( ) 0.04824252

What is the value of p.value element?

- ( ) -2.027021
- ( ) 2.027021

- -0.04824252
- 0.04824252 ✔

EXPLANATION

You can get the statistic and the p-value by running the following command:

```
testResult$statistic
testResult$p.value
```

Question 4

(1/1 point)

This p-value is only valid if the data are normally distributed. Create a density plot for the systolic blood pressure data for mice with genotype BA. Add a density curve for systolic blood pressure data for mice with genotype BB, with a dashed line. What can you conclude from the plot?

- The first set of data, data1, which has genotype BA, resembles the normal distribution

- The second set of data, data2, which has genotype BB, resembles the normal distribution
- Both sets of data resemble the normal distribution
- Neither set of data resembles the normal distribution ✔

EXPLANATION

You can plot the two sets of data using the following command:

```
plot(density(data1))
lines(density(data2),lty=2)
```

It is clear from the plot that none of the two sets of data resembles a bell-curve, which is a prerequisite for normality.

Let's use simulation to obtain a p-value. For this, note that if the two distributions were indeed identical, it wouldn't a priori matter which of the data that had genotype BA and which that had BB, as they would all originate from the same distribution. Following this line of thought, set up the following simulation

- Determine the number `n1` of data with genotype BA.
- Assign `n1` of the data to have genotype BA uniformly at random, and let the rest have genotype BB.
- Calculate the t-test statistic for these two groups having the same mean with the `t.test()` function.

Question 5

(1/1 point)
What is the value of n1?

- 24
- 25
- 26 ✔
- 27

EXPLANATION

You can find the value of n1 by running the following commands:

```
n1<-length(data1)
n1
```

---

1. Create a new data frame from my.data, and assign "BB" as the genotype for all observations.
2. Sample `n1` random numbers with the `sample()` function, and place them in the index `index.temp`.
3. Assign the genotype “BA” to exactly those patients corresponding to `index.temp`.
4. Form the two new groups, and return the test statistic.

With that, you are done with the stochastic part of the simulation.

Question 6

(1/1 point)
The following code presents a possible solution:

```
my.new.data<-my.data
my.new.data$Genotype<-"BB"
index.temp<-sample(1:50,n1)
my.new.data$Genotype[index.temp]<-"BA"
new.data1<-my.new.data$systolic.bp[my.new.data$Genotype=="BA"]
new.data2<-my.new.data$systolic.bp[my.new.data$Genotype=="BB"]
t.test(new.data1,new.data2)$statistic
```

Set the seed to 1234, and run the code. What is the result of the simulation?

- -0.6341197
- 0.945663
- -1.169574 ✔
- 0.3533625

EXPLANATION

Question 7

(1/1 point)

Now that you have done one simulation, wrap the code in a function, and perform 10000 simulations. Store the values of the test statistics in a vector `my.t.values`, and calculate the p-value for the two (original) groups having different means, as the fraction of `my.t.values` that are more extreme (ie., have a higher numerical value) than the test statistic found on **Question 3**.

Set the seed to 554 before running your simulation. What is your calculated p-value from the simulation?

- 0.03416
- 0.03417 ✔
- 0.03418
- 0.03419

EXPLANATION

You can perform the simulations and calculate the p-value using the following code:

```
my.new.data<-my.data
my.new.data$Genotype<-"BB"
doone<-function(){
  index.temp<-sample(1:50,n1)
  my.new.data$Genotype[index.temp]<-"BA"
  new.data1<-my.new.data$systolic.bp[my.new.data$Genotype=="BA"]
  new.data2<-my.new.data$systolic.bp[my.new.data$Genotype=="BB"]
  return(t.test(new.data1,new.data2)$statistic)
}
set.seed(554)
my.t.values<-replicate(100000,doone())
mean((1*(my.t.values)^2>2.2027^2))
```


Download on the
App Store


ANDROID APP ON
Google play