Block5 Lab work

Huizhi Lin

November 22, 2017

## Exercise 1

In the first exercise we are using the built-in data set precip on precipitation data from 70 US cities (see also book page 119). a. Inspect the data set.

precip

## Mobile Juneau Phoenix   
## 67.0 54.7 7.0   
## Little Rock Los Angeles Sacramento   
## 48.5 14.0 17.2   
## San Francisco Denver Hartford   
## 20.7 13.0 43.4   
## Wilmington Washington Jacksonville   
## 40.2 38.9 54.5   
## Miami Atlanta Honolulu   
## 59.8 48.3 22.9   
## Boise Chicago Peoria   
## 11.5 34.4 35.1   
## Indianapolis Des Moines Wichita   
## 38.7 30.8 30.6   
## Louisville New Orleans Portland   
## 43.1 56.8 40.8   
## Baltimore Boston Detroit   
## 41.8 42.5 31.0   
## Sault Ste. Marie Duluth Minneapolis/St Paul   
## 31.7 30.2 25.9   
## Jackson Kansas City St Louis   
## 49.2 37.0 35.9   
## Great Falls Omaha Reno   
## 15.0 30.2 7.2   
## Concord Atlantic City Albuquerque   
## 36.2 45.5 7.8   
## Albany Buffalo New York   
## 33.4 36.1 40.2   
## Charlotte Raleigh Bismark   
## 42.7 42.5 16.2   
## Cincinnati Cleveland Columbus   
## 39.0 35.0 37.0   
## Oklahoma City Portland Philadelphia   
## 31.4 37.6 39.9   
## Pittsburg Providence Columbia   
## 36.2 42.8 46.4   
## Sioux Falls Memphis Nashville   
## 24.7 49.1 46.0   
## Dallas El Paso Houston   
## 35.9 7.8 48.2   
## Salt Lake City Burlington Norfolk   
## 15.2 32.5 44.7   
## Richmond Seattle Tacoma Spokane   
## 42.6 38.8 17.4   
## Charleston Milwaukee Cheyenne   
## 40.8 29.1 14.6   
## San Juan   
## 59.2

1. Assume we know the world average precipitation is 38. We expect that the average from the 70 US cities is lower. Using an hypothesis test, can you show this to be true? Although this is not a statistics lecture, try to give an answer as complete as possible. Use the function t.test to calculate the t-test.

t.test(precip, mu=38, alternative="less")

##   
## One Sample t-test  
##   
## data: precip  
## t = -1.901, df = 69, p-value = 0.03074  
## alternative hypothesis: true mean is less than 38  
## 95 percent confidence interval:  
## -Inf 37.61708  
## sample estimates:  
## mean of x   
## 34.88571

H0: average from the 70 US cities = average precipitation (38) Ha: average from the 70 US cities < average precipitation (38) This is a one sample one-side t-text. The average from the 70 US cities is 34.89. p-value = 0.03074. If we chose alpha = 0.05, than we reject the null hypothesis.

## Exercise 2

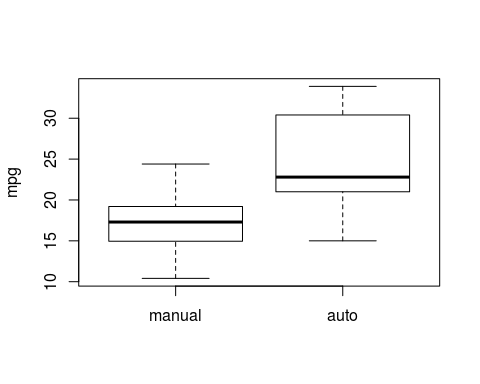
In this exercise, we practise the two-sample t-test using the built-in dataset 'mtcars' (see page 126). Our goal will be to find out if manual transmission cars have a higher mileage than automatic cars. We will assume equal variances.

1. Inspect the data. Make a boxplot showing the distributions of the variable miles per gallon (mpg) for automatic and manual transmission cars.

head(mtcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

manual <- which(mtcars$am==0)  
auto <- which(mtcars$am==1)  
boxplot(mtcars$mpg[manual], mtcars$mpg[auto], ylab="mpg", names=c("manual", "auto"))



1. Do the test. What are your conclusions?

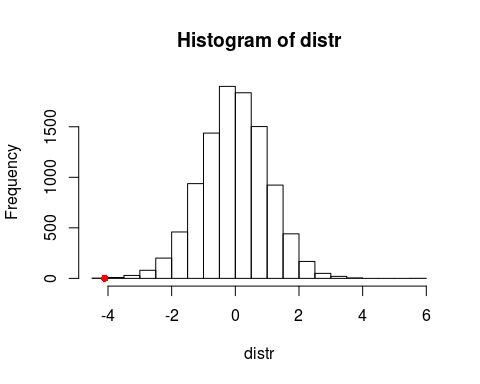
t.test(mtcars$mpg[manual], mtcars$mpg[auto], var.equal = TRUE, alternative="less")

##   
## Two Sample t-test  
##   
## data: mtcars$mpg[manual] and mtcars$mpg[auto]  
## t = -4.1061, df = 30, p-value = 0.0001425  
## alternative hypothesis: true difference in means is less than 0  
## 95 percent confidence interval:  
## -Inf -4.250255  
## sample estimates:  
## mean of x mean of y   
## 17.14737 24.39231

The miles per galon (mpg) for manual transmission cars if significantly lower than the mpg for automatic transmission cars.

1. To calculate the p-value, We use the fact that our test statistic follows approximately a t-18 distribution if the H0 is true. If we did not know this we could also have used a permutation test. To find the distribution in a permutation test we permute the group labels, ensuring no significant difference exists anymore between the two groups. In this distribution, we find the actual test statistic value and we determine the p-value in the permuted distribution.

distr <- sapply(1:10000, function(i) {  
 index <- sample(mtcars$am)  
 manual <- which(index==0)  
 auto <- which(index==1)  
 t.test(mtcars$mpg[manual], mtcars$mpg[auto], var.equal = TRUE, alternative="less")$statistic  
})  
hist(distr)  
real\_t\_statistic <- t.test(mtcars$mpg[manual], mtcars$mpg[auto], var.equal = TRUE, alternative="less")$statistic  
points(real\_t\_statistic,0, col="red", pch=16)



sprintf("%f", sum(distr<real\_t\_statistic)/length(distr))

## [1] "0.000200"

p-value = 0.000300

1. Explain an eventual discrepancy between the p-value from the t-test and the permutation test. How could you make these p-values more similar? In task b we used t-18 distribution, and in task c we used permutation test to find the distribution. Because the permutation used our data (permute the froup labels), therefore, the distribution we got is different from the t-18 distribution. To make these p-values more similar, we may need increase our sample size.

## Exercise 3

High-quality expression profiles were successfully derived from 52 prostate tumours and 50 non-tumour prostate samples from patients undergoing surgery. Oligonucleotide microarrays containing probes for approximately 12600 genes. Since prostate tumours are among the most heterogeneous of cancers, both histologically and clinically, the goal here is to classify tumour and non-tumour samples. The dataset consists of 102 prostate tissues of which 50 are normal and 52 tumour samples. The number of gene expression levels is 12600.

1. Load in the data file Singh.rda.

load("Singh.rda")

The variable Singh contains a list with 4 named elements, X, y, Xt, yt, which are respectively the training design matrix, training classes, test design matrix and test classes. The groups are indicated by a 1 or -1 according as tissue is normal/tumour. There are 12600 gene expressions. We focus in this exercise on the training part of the data (in variable Singh$X).

1. Inspect the data and the elements of the list. Make sure you understand the structure.

names(Singh)

## [1] "X" "y" "Xt" "yt"

dim(Singh$X)

## [1] 102 12600

Singh$y

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [24] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [47] 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  
## [70] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  
## [93] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

102 rows: tissue samples 12600 columns: gene expressions Singh$y has the indicator for normal(1)/tumour(-1) samples.

1. Create 2 separate variables containing all gene expressions. One for the healthy individuals and one for the diseased individuals.

healthy <- Singh$X[Singh$y==1,] # 1=healthy  
diseased <- Singh$X[Singh$y==-1,] #-1=diseased

1. Do a two sample t-test one the first gene. Interpret de outcome. Is there a significant difference in expression level for gene 1 between the 2 groups?

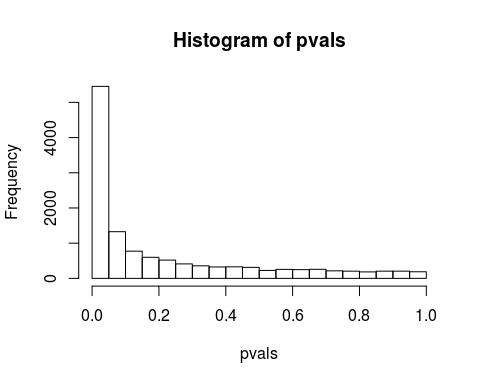
t.test(healthy[,1], diseased[,1], "two.sided")

##   
## Welch Two Sample t-test  
##   
## data: healthy[, 1] and diseased[, 1]  
## t = -1.8275, df = 71.946, p-value = 0.07176  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -9.0422383 0.3926998  
## sample estimates:  
## mean of x mean of y   
## -9.594000 -5.269231

This is a two sided t-test. p-value is 0.07176, which is not significant. Therefore we accept the null hypothesis: there is not difference in expression level for gene 1 between the 2 groups.

1. Do a two sample t-test on all 12600 genes in the training set. How many significant differences do you find (use alpha=0.05)? Create a histogram of all p-values

pvals <- sapply(1:ncol(healthy), function(i) {  
 t.test(healthy[,i], diseased[,i], "two.sided")$p.value  
})  
hist(pvals)



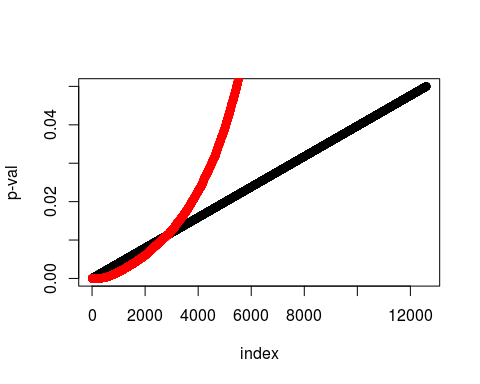
sum(pvals<0.05)

## [1] 5454

There are 5454 t-test of these genes showed a significant (p-value <0.05) difference in expression.

1. Do pairwise comparisons according to Benjamini-Hochberg (see slides for details). How many significant differences are you left with?

j=1:length(pvals)  
m=length(pvals)  
delta=0.05  
bhfdr <- delta\*(j/m)  
  
plot(bhfdr, xlab='index', ylab='p-val')  
points(sort(pvals), col='red')



max(which(sort(pvals)<=bhfdr))

## [1] 2842

There are 2842 genes show significant differences after the Benjamini-Hochberg FDR.

## Exercise 4

García-Arenzana et al. (2014) tested associations of 25 dietary variables with mammographic density, an important risk factor for breast cancer, in Spanish women. For these 25 variables the obtained individual p-values. We are going to use this data set to practice multiple testing corrections.

1. The first step will be to load the data into R. The data file is given as an Excel file which means we need an extra package to read the xlsx file. There are several packages that can do this, but we will use the package XLConnect. \*I can't install the XLConnect package, so I change the file into csv file and load in with "read.csv" command.

theData <- read.csv("dietaryvariables.csv")

1. R has built in methods to adjust a series of p-values either to control the family-wise error rate or to control the false discovery rate. Not all these methods have been discussed (in depth) in the lecture. However, we can still use them (provided we do it correct). The methods Holm, Hochberg, Hommel, and Bonferroni control the family-wise error rate. These methods attempt to limit the probability of even one false discovery (a type I error, incorrectly rejecting the null hypothesis when there is no real effect), and so are all relatively strong (conservative). The methods BH (Benjamini-Hochberg, which is the same as FDR in R) and BY control the false discovery rate. These methods attempt to control the expected proportion of false discoveries.

See the help page of the function p.adjust on information how to use these methods.

?p.adjust

These methods require only the p-values to adjust and the number of p-values that are being compared.

1. First, we order the data by p-value. Check if data is ordered the way we intended

theData = theData[order(theData$Raw.p),]  
theData

## Food Raw.p  
## 20 Total\_calories 0.001  
## 12 Olive\_oil 0.008  
## 25 Whole\_milk 0.039  
## 24 White\_meat 0.041  
## 15 Proteins 0.042  
## 11 Nuts 0.060  
## 5 Cereals\_and\_pasta 0.074  
## 23 White\_fish 0.205  
## 3 Butter 0.212  
## 22 Vegetables 0.216  
## 18 Skimmed\_milk 0.222  
## 16 Red\_meat 0.251  
## 9 Fruit 0.269  
## 7 Eggs 0.275  
## 1 Blue\_fish 0.340  
## 10 Legumes 0.341  
## 4 Carbohydrates 0.384  
## 13 Potatoes 0.569  
## 2 Bread 0.594  
## 8 Fats 0.696  
## 19 Sweets 0.762  
## 6 Dairy\_products 0.940  
## 17 Semi-skimmed\_milk 0.942  
## 21 Total\_meat 0.975  
## 14 Processed\_meat 0.986

1. Perform p-value adjustments using the available methods in the p.adjust function. The available methods are "bonferroni", "Benjamini Hochberg", "holm", "hochberg", "hommel", "Benjamini & Yekutieli". Save all adjusted p-values in the theData variable.

# bonferroni  
theData$Bonferroni <- p.adjust(theData$Raw.p, method = "bonferroni")  
# Benjamini Hochberg  
theData$BH <- p.adjust(theData$Raw.p, method = "BH")  
# holm  
theData$Holm <- p.adjust(theData$Raw.p, method = "holm")  
# hochberg  
theData$Hochberg <- p.adjust(theData$Raw.p, method = "hochberg")  
# hommel  
theData$Hommel <- p.adjust(theData$Raw.p,method = "hommel")  
# Benjamini & Yekutieli  
theData$BY <- p.adjust(theData$Raw.p, method = "BY")

Inspect the resulting theData variable.

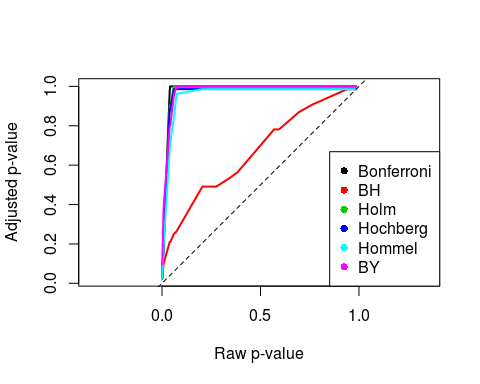
theData

## Food Raw.p Bonferroni BH Holm Hochberg Hommel  
## 20 Total\_calories 0.001 0.025 0.0250000 0.025 0.025 0.025  
## 12 Olive\_oil 0.008 0.200 0.1000000 0.192 0.192 0.192  
## 25 Whole\_milk 0.039 0.975 0.2100000 0.897 0.882 0.682  
## 24 White\_meat 0.041 1.000 0.2100000 0.902 0.882 0.697  
## 15 Proteins 0.042 1.000 0.2100000 0.902 0.882 0.714  
## 11 Nuts 0.060 1.000 0.2500000 1.000 0.986 0.840  
## 5 Cereals\_and\_pasta 0.074 1.000 0.2642857 1.000 0.986 0.962  
## 23 White\_fish 0.205 1.000 0.4910714 1.000 0.986 0.986  
## 3 Butter 0.212 1.000 0.4910714 1.000 0.986 0.986  
## 22 Vegetables 0.216 1.000 0.4910714 1.000 0.986 0.986  
## 18 Skimmed\_milk 0.222 1.000 0.4910714 1.000 0.986 0.986  
## 16 Red\_meat 0.251 1.000 0.4910714 1.000 0.986 0.986  
## 9 Fruit 0.269 1.000 0.4910714 1.000 0.986 0.986  
## 7 Eggs 0.275 1.000 0.4910714 1.000 0.986 0.986  
## 1 Blue\_fish 0.340 1.000 0.5328125 1.000 0.986 0.986  
## 10 Legumes 0.341 1.000 0.5328125 1.000 0.986 0.986  
## 4 Carbohydrates 0.384 1.000 0.5647059 1.000 0.986 0.986  
## 13 Potatoes 0.569 1.000 0.7815789 1.000 0.986 0.986  
## 2 Bread 0.594 1.000 0.7815789 1.000 0.986 0.986  
## 8 Fats 0.696 1.000 0.8700000 1.000 0.986 0.986  
## 19 Sweets 0.762 1.000 0.9071429 1.000 0.986 0.986  
## 6 Dairy\_products 0.940 1.000 0.9860000 1.000 0.986 0.986  
## 17 Semi-skimmed\_milk 0.942 1.000 0.9860000 1.000 0.986 0.986  
## 21 Total\_meat 0.975 1.000 0.9860000 1.000 0.986 0.986  
## 14 Processed\_meat 0.986 1.000 0.9860000 1.000 0.986 0.986  
## BY  
## 20 0.09539895  
## 12 0.38159582  
## 25 0.80135122  
## 24 0.80135122  
## 15 0.80135122  
## 11 0.95398954  
## 5 1.00000000  
## 23 1.00000000  
## 3 1.00000000  
## 22 1.00000000  
## 18 1.00000000  
## 16 1.00000000  
## 9 1.00000000  
## 7 1.00000000  
## 1 1.00000000  
## 10 1.00000000  
## 4 1.00000000  
## 13 1.00000000  
## 2 1.00000000  
## 8 1.00000000  
## 19 1.00000000  
## 6 1.00000000  
## 17 1.00000000  
## 21 1.00000000  
## 14 1.00000000

bonferroni: only the first one is significant Benjamini Hochberg: only the first one is significant holm: only the first one is significant hochberg: only the first one is significant hommel: only the first one is significant Benjamini & Yekutieli: none is significant *Not sure about this interptation*

1. Plot all the different p-values in a single figure.

X <- theData$Raw.p  
Y <- cbind(theData$Bonferroni, theData$BH, theData$Holm, theData$Hochberg, theData$Hommel, theData$BY)  
matplot(X, Y, xlab="Raw p-value", ylab="Adjusted p-value", type="l", asp=1, col=1:6, lty=1, lwd=2)  
legend('bottomright',legend = c("Bonferroni", "BH", "Holm", "Hochberg", "Hommel", "BY"), col=1:6, cex=1, pch=16)  
abline(0, 1, col=1, lty=2, lwd=1)



1. Interpret the last figure. What do the lines represent? Which method is most strict, which one the most lenient? The lines represent how the adjusted p-value increases along with the increasing raw p-value. BH method is the most lenient, when the raw p-value increases, the adjusted p-value increases the slowest when compare to other method. All other methods are very strict, the adjusted p-value reach unsignificant level super quick. Among them Bonferroni may be the most strict one, because it reach the 1.0 adjusted p-value quickest.