Assignment 2

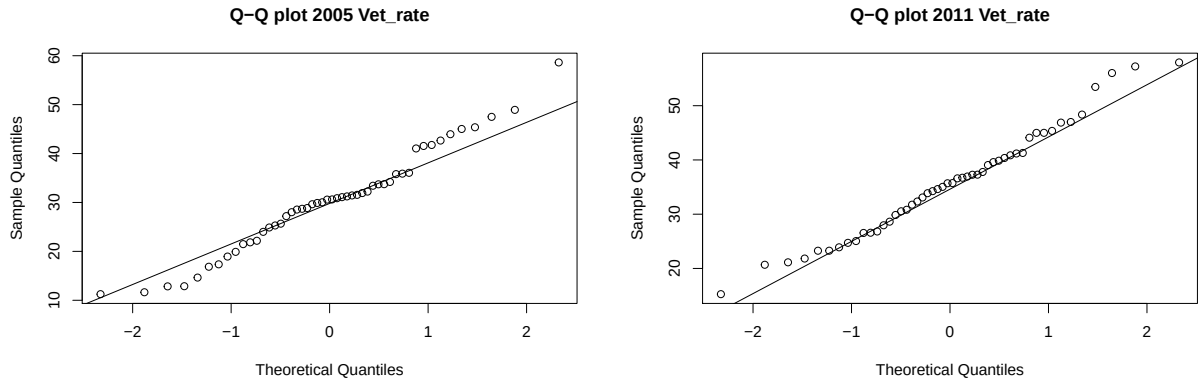
Computational Statistics

11/2019

Names	SNR	ANR
Fernando Catala	2048042	u270800
Francesco Lauria	2041074	u819639
Konstantinos Soiledis	2037001	u226393

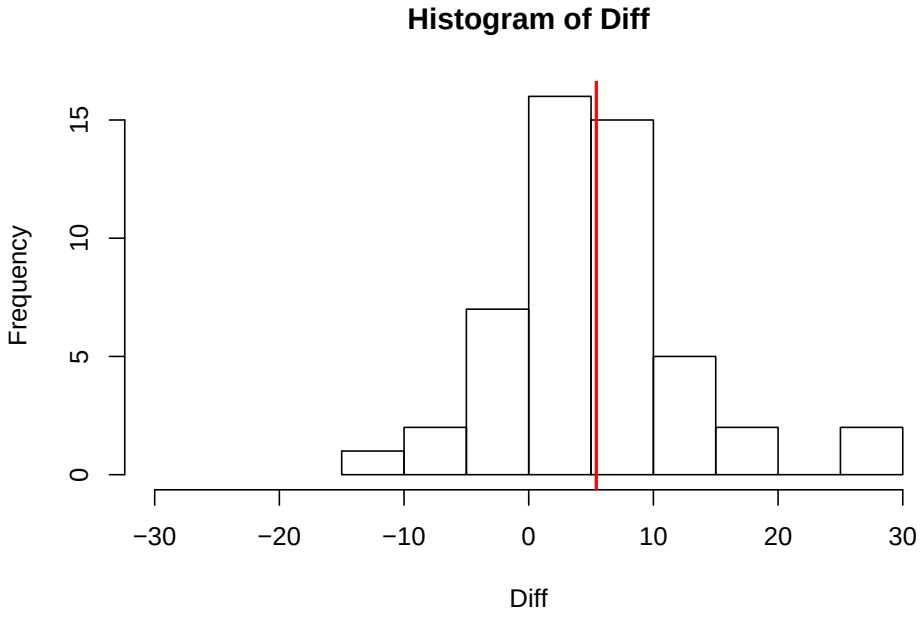
Exercise 1

Question 1. a



The Q-Q plots above check whether the data is normally distributed. In order to have normally distibuted data, the points should be aligned along the theoretical line of 45 degrees. However, it is clear that there are deviations from the line both in 2005 and in 2011. Given these plots, we do not believe that the assumption of normality holds in these particular samples.

The sample means for 2011 and 2005 respectively are $\overline{x}_{2011}$: 35.6882929 and $\overline{x}_{2005}$: 30.2652572. The difference is calculated as $\overline{x}_{2011} - \overline{x}_{2005} = 5.4230357$.



In order to inspect the difference in means, we plotted the histogram of the two features of interest, and we observe that the difference is almost centered in the plot. Furthermore, the plot tends to be familiar to a normal shape. Because of these two reasons we believe that we will fail to reject the H_0.

Question 1. b

1. b. a

The test's null hypothesis is $H_0 : \mu_{2011} - \mu_{2005} = 0$ while the alternative hypothesis is stated as $H_1 : \mu_{2011} - \mu_{2005} > 0$.Below a paired t-test is performed:

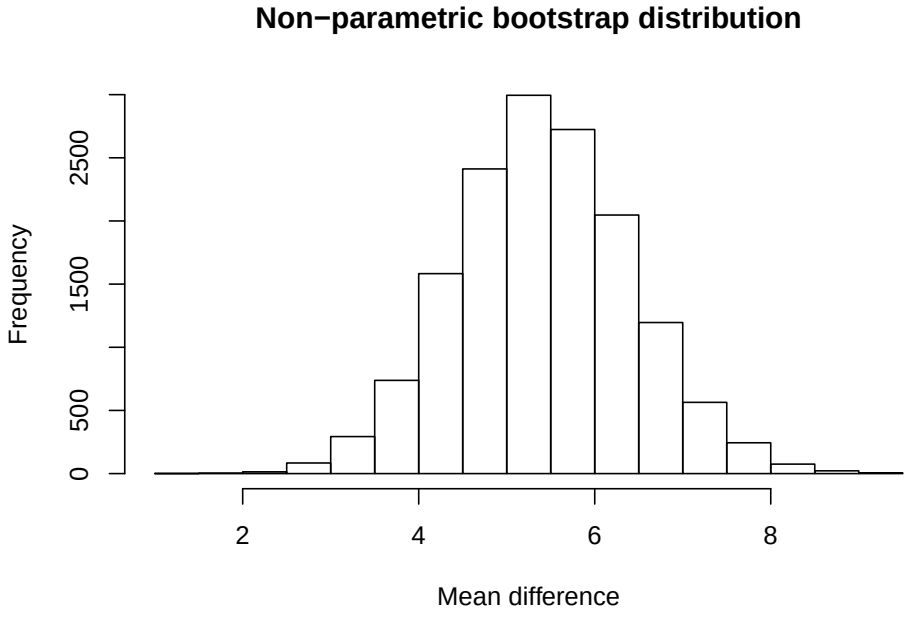
```
##
##  Paired t-test
##
## data:  vet_rate_2011 and vet_rate_2005
## t = 5.3279, df = 49, p-value = 1.245e-06
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  3.716564      Inf
## sample estimates:
## mean of the differences
##                5.423036
```

1. b. b

The reported t-statstic is 5.3279485, with a corresponding p-value of 1.2449175×10^{-6} and a confidence interval of: 3.7165636, ∞ at 95% confidence.

Question 1. c

The plot for the bootstrap is:



After calculating the difference in means of the samples, and after assuming that these means are the true population means (plug-in principle), we calculate the mean of the bootstrap distribution. Finally, we

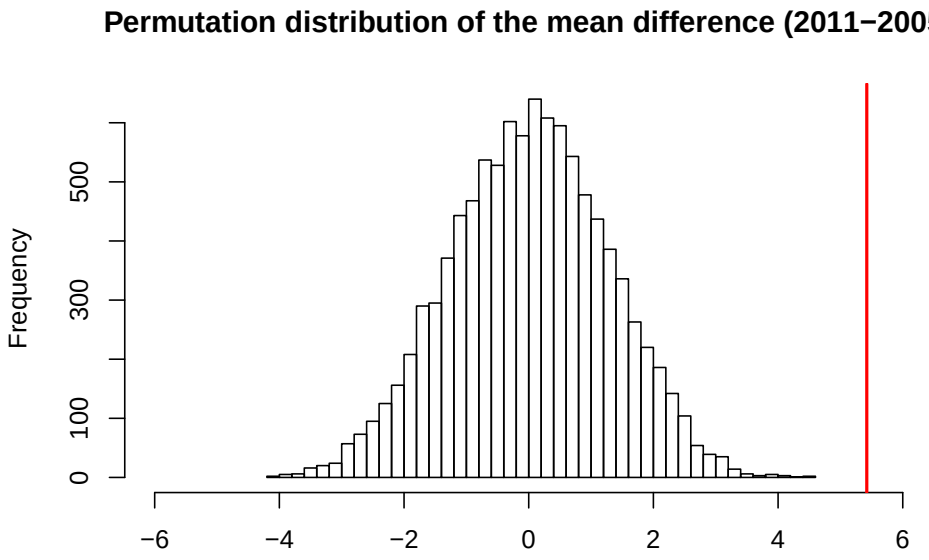
calculate the difference and report it as the bias. The value of the bias is calculated as: -0.0067007. The bias is fairly small, practically 0.

Furthermore, one property of bootstrap distribution is to approximate the shape of the actual sampling distribution. From our bootstrap distribution we see that the data are normally shaped.

As a result, these observations allow a computation of confidence intervals.

Both interval methods deliver the same solutions. At 95% of confidence, we found 7.088152 to be the lower-bound of the bootsrap confidence interval. Similarly, the percentile interval is 7.0981226 . Therefore, we confirm assumptions of normality and unbiasness. Since zero is not included in the interval, we confirm the significance of the p-value. Overall, the similarity between the CI methods confirm that having done the t-test and the bootstrap was adequate.

Question 1. d



The p-value of the permutation test is: 10^{-4}.

The p-value is very small, which means that there is a very low probability to obtain the data under the null hypothesis. The null hypothesis is that there is no difference between 2011 and 2005. Then, we reject it and find that there is a true difference.

Conclusions

Both in bootstrap and permutation methods we reject the null hypothesis, which means that the distributions of the differences in means are not equal. So far, we maintain the conclusion we got in the t-test. In the first place we assumed, but we weren't sure about the normal distribution of our data, so we could not trust the p-value of the t-test. However, we performed a non-parametric bootstrap and a permutation to relax the assumption of normality and still got a significant difference.

In conclusion we can trust the p-value we got from the permutation test. As a result, we can say that there is a true difference in means of the suicide rates between 2011 and 2005.

Exercise 2

Question 2. 1

Both in Student's and Welch's t-tests the hypothesis are the same:

2 samples:
H_0: $\mu_1 = \mu_2$.
H_1: $\mu_1 \neq \mu_2$.

Permutation:
H_0: sample 1 and sample 2 are distributed randomly.
H_1: sample 1 and sample 2 are not distributed randomly, there is a pattern.

Question 2. 2

The model runs with fixed values, n2 = 100 and sigma_squared1 = 1.

Under the null hypothesis, which is stated as $H_0 : \mu 1 = \mu 2$, the results reported are demonstrated below:

```
##   n1.i. sigma_squared2.j. levelS levelW levelP
## 1    10                 1 0.1404 0.0508 0.1378
## 2    10                 2 0.0522 0.0530 0.0506
## 3    10                10 0.0006 0.0488 0.0006
## 4   100                 1 0.0500 0.0496 0.0484
## 5   100                 2 0.0464 0.0464 0.0444
## 6   100                10 0.0488 0.0486 0.0452
## 7   200                 1 0.0288 0.0522 0.0274
## 8   200                 2 0.0474 0.0464 0.0452
## 9   200                10 0.1166 0.0472 0.1132
```

In order to control Type I Error, we would expect values closer to 0.05.

- In conditions 4, 5 & 6 where sample sizes are equal, we see that levels are the most similar.
- In conditions 1 & 9, we see that Student's and Permutation have particularly bad control compared to Welch's.
- In condition 3, we see that Welch has almost full control, while the other two models are a bit far from optimal.
- In conditions 2, 5 & 8 where variances are equal, we see that levels are the most similar.
- Across conditions, permutation is usually outperformed by the t-tests. Additionally, Welch usually performs the best.

Under the alternative hypothesis, which is stated as $H_1 : \mu 1 \neq \mu 2$, the results reported are demonstrated below:

```
##   n1.i. sigma_squared2.j. PowerS PowerW PowerP
## 1    10                 1 0.7538 0.5176 0.7460
## 2    10                 2 0.5648 0.5032 0.5548
## 3    10                10 0.0354 0.4128 0.0330
## 4   100                 1 0.9998 0.9998 1.0000
## 5   100                 2 0.9990 0.9990 0.9990
## 6   100                10 0.8158 0.8136 0.8078
## 7   200                 1 1.0000 1.0000 1.0000
## 8   200                 2 0.9998 0.9998 0.9998
## 9   200                10 0.9292 0.8462 0.9252
```

In order to control Type II Error, we would expect values closer to 1.

- We see a full control of Type II Error in conditions 4, 5, 7 & 8, where sample sizes are large and variances are small and similar.
- In the 3rd condition, the power is higher in Welch's, because variances are quite different (equal variances are assumptions of Student's) and n1 is too little (which plays against the Permutation). This effect dilutes in condition 6, because both sample sizes are large and equal.
- Finally, this effect doesn't hold true anymore in condition 9, where n1 increases to 200. Values from condition 4 onwards are particularly higher than 1-3, since power is correlated with the sample size.

Conclusions

Advantages:

Permutation tests are accurate, while t-tests are approximations. Modern computational methods allow to conduct permutation tests, while in the past it was more efficient to conduct t-tests.

Permutation tests exist for any test statistic, regardless of whether or not its distribution is known. On the other hand, t-tests assume that the population follows a normal distribution.

Disadvantages:

We cannot test whatever H_0 with permutation tests, as permutation requires the assumption that there is randomness in the data. If there is no reason to assume a priori that our data may be randomly distributed, there is no point in conducting a permutation.

Permutation tests, do not assume anything about the type of the distribution, therefore are very good tools for hypothesis testing under many conditions. However, their computational demands can pose an issue, especially in large samples. The same applies if many tests have to be conducted at the same time.

Our permutation with S = 5000 and B = 499 took 0.5 hours to run, while when we tried with S = 10000 and B = 999 it took 2.5 hours, where accuracy didn't seem much better to justify the time.

Overall, we cannot say that we should always use a permutation instead of a t-test. We must consider its disadvantages. However, if we meet the conditions of randomness, it is advisable to do a permutation, as it outperforms the t-test in level of accuracy.

APPENDIX CODE

```
knitr::opts_chunk$set(echo = TRUE)

#Load data
VetSuicides_2005 <- read.csv("input/VetSuicides_2005.csv")
VetSuicides_2011 <- read.csv("input/VetSuicides_2011.csv")

## store the interested variables
vet_rate_2005<-VetSuicides_2005$vet_rate
vet_rate_2011<-VetSuicides_2011$vet_rate

#---------------------------------- Q1.a ----------------------------------------
#Checking distribution
{qqnorm(vet_rate_2005,main = "Q-Q plot 2005 Vet_rate")
qqline(vet_rate_2005)}

{qqnorm(vet_rate_2011,main = "Q-Q plot 2011 Vet_rate")
qqline(vet_rate_2011)}

#Computing means
sample_mean_2011 <- mean(vet_rate_2011)

sample_mean_2005 <- mean(vet_rate_2005)

#Dataset to compute
Diff <- vet_rate_2011 - vet_rate_2005

#Get a histogram of the difference in means.
hist(Diff, xlim = c(-30,30))
abline(v = mean(Diff), col=2, lwd=2)


#---------------------------------- Q1.b ----------------------------------------
#Performing t-test
ttest <- t.test(vet_rate_2011,vet_rate_2005, var.equal = TRUE, alternative="greater",
       paired=TRUE)
ttest

#---------------------------------- Q1.c ----------------------------------------

## NON PRAMETRIC BOOTSTRAP:
non_parametric_simulation<-function(B,vet_rate_2011,vet_rate_2005){
  result<-matrix(nrow=B,ncol=2)
  colnames(result)<-c("Mean_2011","Mean_2005")
  for (i in 1:B)
  {
    indices<-sample(c(1:50),replace=TRUE)
    bootsample_2011<-vet_rate_2011[indices]
    mean_2011 <- mean(bootsample_2011)
    bootsample_2005<-vet_rate_2005[indices]
    mean_2005 <- mean(bootsample_2005)
    result[i,]<-c(mean_2011,mean_2005)
```

```
  }
  return(result)
}

set.seed(12345)
result<-non_parametric_simulation(15000,vet_rate_2011,vet_rate_2005)

# sampling distribution of the paired different in means
bootstrap_distribution<-(result[,1]-result[,2])

#Plotting bootstrap distribution
hist(bootstrap_distribution,
     main = "Non-parametric bootstrap distribution",
     xlab = "Mean difference")
#Computing Bias
obs_mean<-mean(vet_rate_2011)-mean(vet_rate_2005)
boot_mean<-mean(bootstrap_distribution)
bias<-boot_mean-obs_mean # NO BIAS
#Confidence intervals

#Get the lenght
N<-length(vet_rate_2005)
M<-length(vet_rate_2011)

SE_bootstrap<-sd(bootstrap_distribution)
alfa <- 0.05
tcv <- qt(1-alfa,N+M-2)

upper <- boot_mean+(tcv*SE_bootstrap)

upper_quant <- quantile(bootstrap_distribution,0.95)

#---------------------------------- Q1.d ----------------------------------------

#Permutation
all_data<-matrix(nrow = 50, ncol = 2)
all_data[1:50,1]<-vet_rate_2011
all_data[1:50,2]<-vet_rate_2005
colnames(all_data)<-c("vet_rate_2011","vet_rate_2005")

set.seed(12345)

# Permutation procedure:
B <- 9999  #nr of permutation samples
out <- matrix(nrow = B, ncol = 2)
groups<-matrix(nrow = 50, ncol = 2)
colnames(groups)<-c("group_1","group_2")
for (i in 1:B){
  for(row in 1:50){
    groups[row,]<-sample(all_data[row,],2)
  }
  mean_group1<-mean(groups[,1])
  mean_group2<-mean(groups[,2])
```

```
  out[i,] <- c(mean_group1,mean_group2)
}

#Calculate p-value
diff<-out[,1]-out[,2]
obsdiff<-mean(vet_rate_2011)-mean(vet_rate_2005)
pval<-(1+sum(diff>(obsdiff)))/(B+1)

#Creating the plot.
a <- hist(diff,50)

#Permutation distribution
barcols = a$breaks
barcols[a$breaks<round(obsdiff,0)] = 0
barcols[a$breaks>=round(obsdiff,0)] = 2
hist(diff,50, xlab = "",
      main = "Permutation distribution of the mean difference (2011-2005)",
      col=barcols,xlim = c(-6,6))
abline(v=obsdiff, col =2, lwd = 2)

#--------------------------------- Q2.2 ----------------------------------------
#Monte carlo simulation

#inputs
sigma_squared2 = c(1, 2, 10) ; n1 = c(10, 100, 200)
n2<- 100 ; sigma_squared1 = 2
S = 5000; B = 499
set.seed(3)


MonteCarlo=function(S,B,n1,n2,sigma_squared1,sigma_squared2,mu1,mu2){

  result <- matrix(nrow = S ,ncol = 3 )
  colnames(result)<-c("student p", "welch p", "perm")

  for (s in 1:S){
    #Getting 2 sample data.
    out1<-matrix(rnorm(n1,mu1,sqrt(sigma_squared1)))
    out2<-matrix(rnorm(n2,mu2,sqrt(sigma_squared2)))

    #Getting the Welch and Student
    Welcht <- t.test(out1, out2, alternative = "two.sided")
    Studentt <- t.test(out1, out2, var.equal = T, alternative = "two.sided")

    ## Permutation

    #Getting a combined data set.
    all_out <- rbind(out1,out2)
    #Getting the difference in means.
    obs_diff <- mean(out1) - mean(out2)


    # Permutation procedure:
```

```
    out <- matrix(nrow = B, ncol = 2)

      for (i in 1:B){
        permnrs <- sample(n1+n2) #this is the re-sampling without replacement
        xbar_group1 <- mean(all_out[permnrs[1:n1],1])  # in this way we label the sample
        xbar_group2 <- mean(all_out[permnrs[(n1+1):(n1+n2)],1])
        out[i,] <- c(xbar_group1,xbar_group2)
      }

    #calculate p-value
    diff<-out[,1]-out[,2]
    pval<-(1 + sum(abs(diff) > abs(obs_diff)))/(B+1) # larger or different

    ##Storing results
    result[s,]<-c(Studentt$p.value, Welcht$p.value, pval)
  }
    return(result)
}


#Null hypothesis, Computing Levels, 9 cases.
mu1=0;mu2=0

Null_Hypothesis_level<- data.frame(row.names = c("n1","sigma_sq_2","Student lvl",
                                    "Welch LvL", "Permutation LvL"))
for (i in 1:3){
  for (j in 1:3){
    result <- MonteCarlo(S,B,n1[i],n2,sigma_squared1,sigma_squared2[j],mu1,mu2)

    #Getting levels
    levelS <- sum(result[,1]<0.05)/S
    levelW <- sum(result[,2]<0.05)/S
    levelP <- sum(result[,3]<0.05)/S

    Result_level <- data.frame(n1[i],sigma_squared2[j],levelS,levelW,levelP)

    Null_Hypothesis_level <- rbind.data.frame(Null_Hypothesis_level,
                                              Result_level)
  }
}

#Alternative hypothesis, computing Power, 9 cases.
mu1=0;mu2=1

Alt_Hypothesis_power<- data.frame(row.names = c("n1","sigma_sq_2","Student Power",
                                    "Welch Power", "Permutation Power"))
for (i in 1:3){
  for (j in 1:3){
    result <- MonteCarlo(S,B,n1[i],n2,sigma_squared1,sigma_squared2[j],mu1,mu2)

    #Getting power
    PowerS <- sum(result[,1]<0.05)/S
    PowerW <- sum(result[,2]<0.05)/S
```

```
    PowerP <- sum(result[,3]<0.05)/S

    Result_level <- data.frame(n1[i],sigma_squared2[j],PowerS, PowerW, PowerP)
    Alt_Hypothesis_power <- rbind.data.frame(Alt_Hypothesis_power,
                                             Result_level)
  }
}


#Matrix demonstration Level.
Null_Hypothesis_level

#Matrix demonstration Power
Alt_Hypothesis_power
```