STAT645 - Homework 4

Salih Kilicli

10/1/2019

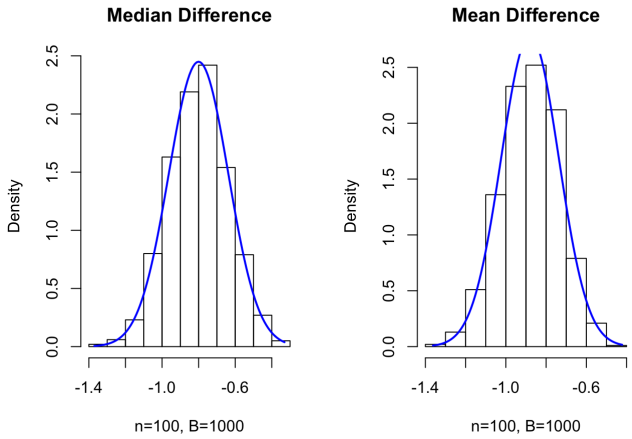
Problem 1: Carry out a simulation to investigate the sampling distribution of a difference in two sample medians and compare it with that of two sample means. Let $n = 100$ (the sample size), and carry out $B = 1000$ simulations. Simulate samples from:

1) Gamma with shape 2 and scale 0.5

2) Gamma with shape 2.5 and scale 0.75; each sample should be of size n = 100.

Simulate the difference in medians (first group minus the second). Report a histogram of the simulated median differences. Overlay a density curve from the normal distribution with mean and variance same as that of the sampling distribution. Do a similar sampling distribution plot for the difference of the sample means, and comment on the differences.

```
B = 1000
n = 100
dmedian = 0*1:B
dmean = 0*1:B

for (i in 1:B)
{
set.seed(i)
s1 = rgamma(n, shape=2, scale=0.5)
s2 = rgamma(n, shape=2.5, scale=0.75)
dmedian[i] = median(s1)-median(s2)
dmean[i] = mean(s1-s2)
}

par(mfrow = c(1, 2))

x1 = seq(min(dmedian), max(dmedian), by = 0.01)
y1 = dnorm(x1, mean=mean(dmedian), sd=sd(dmedian))
hist(dmedian, main = "Median Difference", xlab = "n=100, B=1000", probability= TRUE)
lines(y1 ~ x1, col="blue", lwd=2)

x2 = seq(min(dmean),max(dmean), by=0.01)
y2 = dnorm(x2, mean=mean(dmean), sd=sd(dmean))
hist(dmean, main = "Mean Difference", xlab = "n=100, B=1000", probability= TRUE)
lines(y2 ~ x2, col="blue", lwd=2)
```



The plot of median difference looks skewed as compared to plot of mean difference.

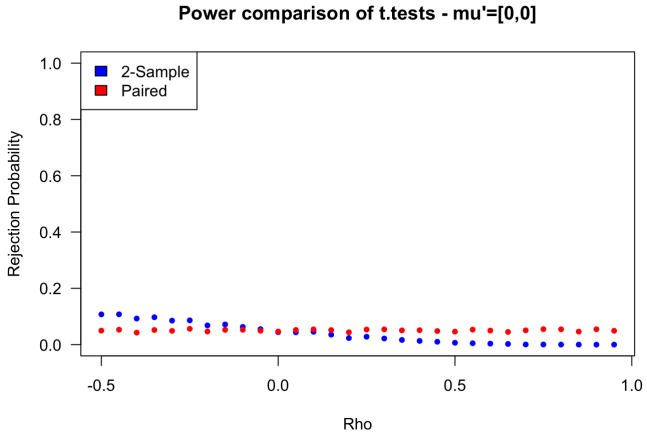
Problem 2: Carry out a simulation to compare the two sample t-test (with equal variance) to the paired t-test under a variety of correlation values. Specifically, for pairwise correlations, σ, of $-0.5, -0.45, \ldots, 0.0, 0.05, \ldots, 0.95$, use simulation (number of simulations, B, equal to 5000) to approximate the power to reject the null hypothesis of equal means against the two-sided alternative hypothesis. Let the sample size n be 30; so, there will be 30 pairs. For the population variance, use $\sigma = 1$ for both groups; i.e.,

$$\Sigma = \begin{bmatrix} 1 & \sigma \\ \sigma & 1 \end{bmatrix}.$$

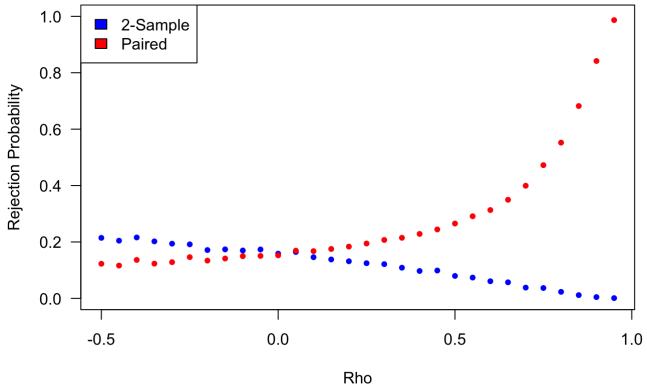
You can simulate correlated data using *mvrnorm* in the *MASS* package of R.

For each of $\mu = [0, 0]$, $\mu = [0, 0.25]$, $\mu = [0, 0.5]$, and $\mu = [0, 0.75]$, do the above and report a plot of σ versus the rejection probabilities. Each plot should have 2 connected line segments, one for the two-sample t-test and one for the paired t-test; use different colors to distinguish between the two. You may use the t.test function of R. What do the plots tell you?

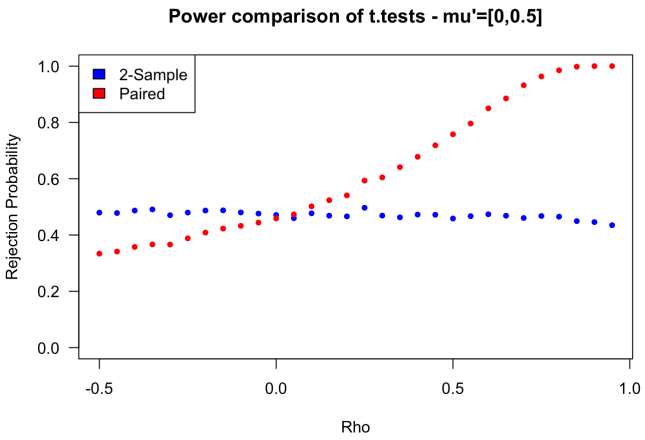
```
library(MASS)

rho = seq(-0.5, 0.95, by=0.05)
m=rbind(c(0,0),c(0,0.25),c(0,0.5),c(0,0.75))
p1=NULL; rej1=NULL;
p2=NULL; rej2=NULL;

  for (i in 1:30){
    sigma=matrix(c(1,rho[i],rho[i],1), ncol=2, byrow=T)

      for (j in 1:5000){
        data=mvrnorm(30, m[1,], sigma)
        p1[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", var.equal=T)$p.value
        p2[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", paired=T)$p.value

      }

  rej1[i]=sum(p1<0.05)/length(p1)
  rej2[i]=sum(p2<0.05)/length(p1)
  }

plot(rho, rej1, ylim=c(0,1), pch=20, ylab="Rejection Probability", xlab="Rho",
     main="Power comparison of t.tests - mu'=[0,0]", col="blue", las=1)
points(rho, rej2, ylim=c(0,1), pch=20, col="red")
legend("topleft",  c("2-Sample", "Paired"), fill=c("blue","red"))
```



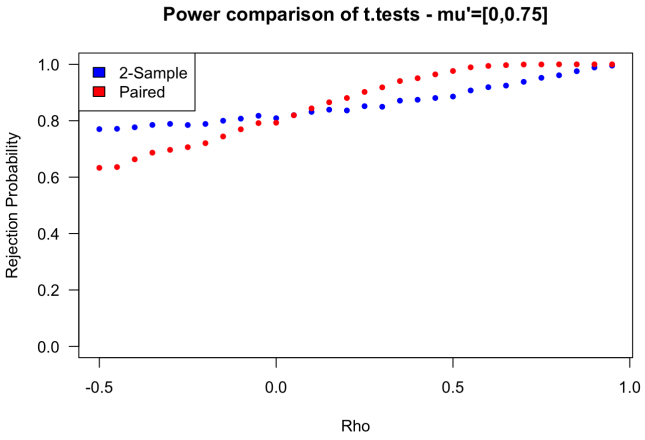

```
p1=NULL; rej1=NULL;
p2=NULL; rej2=NULL;

  for (i in 1:30){
    sigma=matrix(c(1,rho[i],rho[i],1), ncol=2, byrow=T)

      for (j in 1:5000){
        data=mvrnorm(30, m[2,], sigma)
        p1[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", var.equal=T)$p.value
        p2[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", paired=T)$p.value
      }

  rej1[i]=sum(p1<0.05)/length(p1)
  rej2[i]=sum(p2<0.05)/length(p2)
  }

plot(rho, rej1, ylim=c(0,1), pch=20, ylab="Rejection Probability", xlab="Rho",
     main="Power comparison of t.tests - mu'=[0,0.25]", col="blue", las=1)
points(rho, rej2, ylim=c(0,1), pch=20, col="red")
legend("topleft",  c("2-Sample", "Paired"), fill=c("blue","red"))
```



```
p1=NULL; rej1=NULL;
p2=NULL; rej2=NULL;

  for (i in 1:30){
    sigma=matrix(c(1,rho[i],rho[i],1), ncol=2, byrow=T)

      for (j in 1:5000){
        data=mvrnorm(30, m[3,], sigma)
        p1[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", var.equal=T)$p.value
        p2[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", paired=T)$p.value
      }

  rej1[i]=sum(p1<0.05)/length(p1)
  rej2[i]=sum(p2<0.05)/length(p2)
  }

plot(rho, rej1, ylim=c(0,1), pch=20, ylab="Rejection Probability", xlab="Rho",
     main="Power comparison of t.tests - mu'=[0,0.5]", col="blue", las=1)
points(rho, rej2, ylim=c(0,1), pch=20, col="red")
legend("topleft",  c("2-Sample", "Paired"), fill=c("blue","red"))
```




```
p1=NULL; rej1=NULL;
p2=NULL; rej2=NULL;

  for (i in 1:30){
    sigma=matrix(c(1,rho[i],rho[i],1), ncol=2, byrow=T)

      for (j in 1:5000){
        data=mvrnorm(30, m[4,], sigma)
        p1[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", var.equal=T)$p.value
        p2[j]=t.test(data[,1], data[,2], mu=0, alternative="two.sided", paired=T)$p.value
      }

  rej1[i]=sum(p1<0.05)/length(p1)
  rej2[i]=sum(p2<0.05)/length(p2)
  }

plot(rho, rej1, ylim=c(0,1), pch=20, ylab="Rejection Probability", xlab="Rho",
     main="Power comparison of t.tests - mu'=[0,0.75]", col="blue", las=1)
points(rho, rej2, ylim=c(0,1), pch=20, col="red")
legend("topleft",  c("2-Sample", "Paired"), fill=c("blue","red"))
```

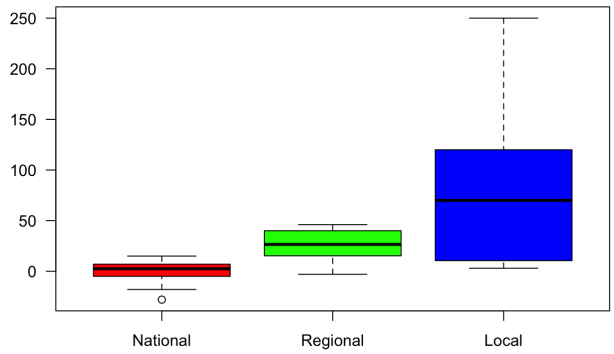


It is apparent from the plots that the power of paired t-test is higher than 2 sample t-test since it takes higher probabilities in similar cases as rho increases.

Problem 3: Consider the calories data in "calories.txt." These contain data on the difference between the true and reported calorie contents (we will use the differences per item)

(a) Construct side by side box plots of the calorie differences by region. Comment.

```
library(nlme)
setwd("~/Desktop/STAT645/Data")
cal = read.delim("calories.txt")
calories=cal[,3]
region = factor(c(rep(1,20),rep(2,12),rep(3,8)))
boxplot(calories ~ region, names=c("National","Regional","Local"), col=rainbow(3), las=1)
```



Both variability and the median in the calorie information increases from National to Local. Also, there is an outlier for the National region with a minimal value.

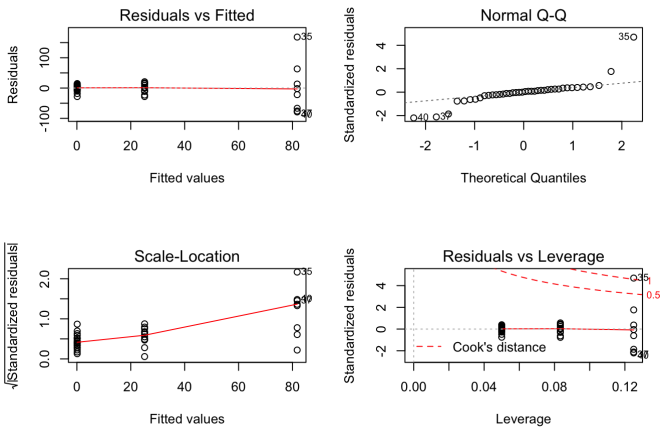
(b) Fit an ordinary least squares model using lm. Report the coefficient estimates, standard errors, and p-values.

```
model1 = lm(calories ~ region)
summary(model1)
```

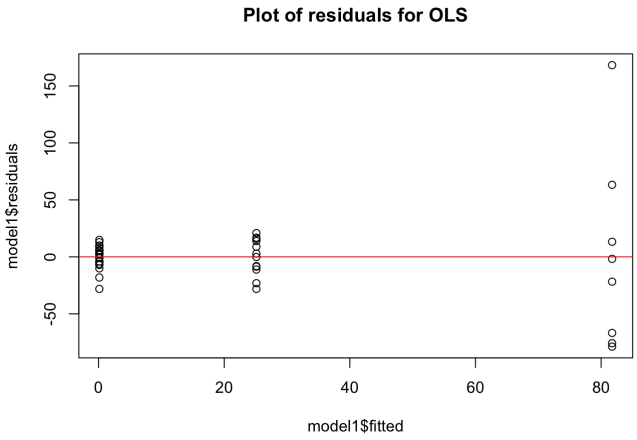
```
##
## Call:
## lm(formula = calories ~ region)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -78.750  -9.000   0.875   9.875 168.250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.125      8.566   0.015   0.9884
## region2       25.000     13.989   1.787   0.0821 .
## region3       81.625     16.026   5.093 1.06e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.31 on 37 degrees of freedom
## Multiple R-squared:  0.4123, Adjusted R-squared:  0.3805
## F-statistic: 12.98 on 2 and 37 DF,  p-value: 5.361e-05
```

(c) What assumption of the ordinary least squares model appears to be violated?

```
par(mfrow=c(2,2))
plot(model1)
```




```
par(mfrow=c(1,1))
plot(model1$fitted, model1$residuals,  main="Plot of residuals for OLS")
abline(h=0, col="red")
```



Clearly, from "sqrt(standard residuals) vs fitted values" (bottom left) or "fitted values vs residuals" plots we can see that the error variance is increasing. Therefore, **constant error variance** assumption is violated.

(d) Fit a weighted least squares model using gls, allowing for different variances for each region. Report the coefficient estimates, standard errors,and p-values. Comment on how these compare to the ordinary least squares results.

```
model2 = gls(calories ~ 1+region, weights = varIdent(form=~1|region))
summary(model2)
```

```
## Generalized least squares fit by REML
##   Model: calories ~ 1 + region
##   Data: NULL
##        AIC     BIC    logLik
##   337.1125 346.778 -162.5562
##
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | region
##  Parameter estimates:
##        1        2        3
## 1.000000 1.527862 7.981460
##
## Coefficients:
##              Value Std.Error  t-value p-value
## (Intercept)  0.125  2.352455 0.053136  0.9579
## region2     25.000  5.202386 4.805487  0.0000
## region3     81.625 29.780567 2.740881  0.0094
##
##  Correlation:
##         (Intr) regin2
## region2 -0.452
## region3 -0.079  0.036
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.67335201 -0.60595979  0.07500991  0.74971757  2.00371677
##
## Residual standard error: 10.5205
## Degrees of freedom: 40 total; 37 residual
```

Coefficient estimates are exactly same; however, std. errors and t-values therefore p-values are fixed (are different).

Problem 4: For the "calories.txt" data: Use the bootstrap to test the null hypothesis H_0 that the population median per-item calorie difference equals zero.

(a) Report a 95% BCa confidence interval for the population median. Comment on whether it supports H_0 or not.

```
library(boot)
setwd("~/Desktop/STAT645/Data")
cal = read.delim("calories.txt")
calories=cal[,3]

boot_median = function(calories, indices){
boot_data = calories[indices]
return(median(boot_data))
}

newdata=calories-median(calories)
model4a=boot(calories, boot_median, 10000)
boot.ci(model4a, type="bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = model4a, type = "bca")
##
## Intervals :
## Level       BCa
## 95%   ( 3, 15 )
## Calculations and Intervals on Original Scale
```

Since 0 is not in the confidence interval we reject the Null hypothesis. There it **doesn't** support the H_0.

(b) Use the bootstrap to compute a p-value, using the sample median $\bar{y}$ as your test statistic.

```
x = boot(newdata, boot_median, 10000)
tobs = median(calories)
jj=(1:10000)[!is.na(model4a$t)]
p0=sum(as.numeric(x$t>abs(tobs) | x$t<=-abs(tobs)))/length(jj)
cat("p-value using bootstrap method is", p0)
```

```
## p-value using bootstrap method is 0.0058
```

(c) Using a bootstrap approach estimate the bias of the sample median.

```
model4a
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = calories, statistic = boot_median, R = 10000)
##
##
## Bootstrap Statistics :
##     original   bias    std. error
## t1*        9 0.161275    3.600799
```

```
print("Estimate of the bias of the sample median is 0.211075")
```

```
## [1] "Estimate of the bias of the sample median is 0.211075"
```

Problem 5: For the "mouse.txt" data: Use the bootstrap to test the null hypothesis H_0 that the population means of the treatment and control groups are equal.

(a) Report a 95% BCa confidence interval for the difference in population means. Comment on whether it supports H_0 or not.

```
library(boot)
setwd("~/Desktop/STAT645/Data")
mouse = read.delim("mouse.txt",header=F)

boot_diff_means_ci = function(data, ind){
  data = data[ind,]
  if(!all(data[,1]=="C")&!all(data[,1]=="TX")){
  return(mean(data[data[,1]=="TX", 2])- mean(data[data[,1]=="C", 2]))
  } else {return(NA)}
}

model5a = boot(mouse, boot_diff_means_ci, 10000)
summary(model5a)
```

```
##           Length Class      Mode
## t0             1  -none-     numeric
## t          10000  -none-     numeric
## R              1  -none-     numeric
## data           2  data.frame list
## seed         626  -none-     numeric
## statistic      1  -none-     function
## sim            1  -none-     character
## call           4  -none-     call
## stype          1  -none-     character
## strata        16  -none-     numeric
## weights       16  -none-     numeric
```

```
boot.ci(model5a, type="bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 9999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = model5a, type = "bca")
##
## Intervals :
## Level       BCa
## 95%   (-22.07,  87.31 )
## Calculations and Intervals on Original Scale
```

Since 0 is within the interval we fail to reject the Null hypothesis. In other words, the CI we have found supports H_0.

(b) Use the bootstrap to compute a p-value, using the two-sample t-statistic (unequal variances) as your test statistic.

```
diff_pval = function(data, ind){
  tx=data[ind,1]
  y =data[ind,2]
  if(sum(tx=="C")>=2 & sum(tx=="TX")>=2){
    return(t.test(y[tx=="C"], y[tx=="TX"], var.equal=F)$stat)
  } else {return(NA)}
}
data = mouse
y0=y=data[,2]
y[data[, 1]=="TX"]=y[data[,1]=="TX"]-mean(y[data[,1]=="TX"])
y[data[, 1]=="C"]=y[data[,1]=="C"]-mean(y[data[,1]=="C"])

datanew=data
datanew[,2]=y

tobs= t.test(y0[data[,1]=="TX"], y0[data[,1]=="C"], var.equal=F)$stat

model5b = boot(datanew, diff_pval, 10000)


jj=(1:10000)[!is.na(model5b$t)]
p = sum(model5b$t[jj]>abs(tobs) | model5b$t[jj]<= -abs(tobs) )/length(jj)
cat("p-value using bootstrap method is", p)
```

```
## p-value using bootstrap method is 0.324492
```

(c) Compare the bootstrap p-value with the p-value obtained using the standard two-sample t test assuming unequal variances.

```
p1=t.test(y0[data[,1]=="TX"], y0[data[,1]=="C"], var.equal=F)$p.value
cat("p-value using bootstrap is ", p, "whereas p-value using t-test is",p1)
```

```
## p-value using bootstrap is  0.324492 whereas p-value using t-test is 0.3155007
```

p-values in both methods are pretty close to each other.

Problem 6: For the pollution data in "pollute data.csv" [Note: There is at least one missing value in these data. Just remove any records with at least one missing value prior to doing the following analyses.]: Consider the model from 1(d) in homework 3. For each of the "bootstrapping pairs" and "bootstrapping residuals" approaches, approximate the sampling distribution of the sum of the coefficients for percent white collar and percent non white, and report the following:

(a) A histogram of the estimated sampling distribution.

(b) 95% confidence intervals, using the normal-theory approach, the percentile approach, and the BCa approach.