HW3: Power Analysis and Bootstrap (Due: Wed, Nov 13, at noon)

Preliminaries

The following function simulates data from a bi-variate distribution for pairs of RVs $[X_i, Y_i]$. Here X_i is exponentially distributed, and the probability of Y_i given $X_i = x_i$ is $P(Y_i|X_i = x_i) = N(\mu + x_i\rho, \sigma^2)$, by construction the correlation between X_i and Y_i is ρ.

```
simXY=function(n,rho,mu=10){
    x=scale(rexp(n))
    y=mu+x*rho+rnorm(n,sd=sqrt(1-rho^2))
    return(data.frame(x,y))
}
```

A simple test

```
DATA=simXY(10000,-.5)
head(DATA)
```

```
##             x         y
## 1 -0.35892732  8.867440
## 2 -0.31949347 11.234041
## 3 -0.44751184  9.443104
## 4 -0.94022084  9.913393
## 5 -0.87733433 10.857067
## 6 -0.06930156 10.078346
```

```
dim(DATA)
```

```
## [1] 10000     2
```

```
cor(DATA)
```

```
##            x          y
## x  1.0000000 -0.4983969
## y -0.4983969  1.0000000
```

1) Power Analysis

We now the use the $simXY()$ function to simulate 30 IID pairs setting `rho=0.1`.

```
 set.seed(120549)
 DATA_1=simXY(n=30,rho=0.1)
 head(DATA_1)
```

```
##             x         y
## 1 -0.37384186 12.412471
## 2 -0.75587129  8.653783
## 3 -0.77233910 11.598143
## 4  1.73863681 10.627966
## 5 -0.18162561  9.284235
```

```
## 6 -0.03993584  9.854104
```

The sample correlation estiamte is

```
cor(DATA_1$x,DATA_1$y)
```

```
## [1] -0.2520449
```

The result printed above is a point estimate based on the sample we have. If we generate another sample, we will obtain a different point estiamte

```
 set.seed(195021)
 DATA_2=simXY(n=30,rho=0.1)
 cor(DATA_2$x,DATA_2$y)
```

```
## [1] 0.1147978
```

The above example illustrate how estimates vary over repeated sampling. To make inferences about population parameters we need to attach to our estimates measures of uncertainty (e.g., SEs) and perform formal hypothesis testing.

The function `cor.test()` can be used for inferences about correlations. It can be used to obtain CIs and to test the hypotheses such as $H_0 : \rho_{x,y} = 0$ Vs $H_0 : \rho_{x,y} \neq 0$. The function useses Fisher's z-transform.

```
test=cor.test(DATA_2$x,DATA_2$y)
print(test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  DATA_2$x and DATA_2$y
## t = 0.6115, df = 28, p-value = 0.5458
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2560616  0.4561995
## sample estimates:
##       cor
## 0.1147978
```

You can get CI, and p-values using the following

```
test$conf.int
```

```
## [1] -0.2560616  0.4561995
## attr(,"conf.level")
## [1] 0.95
```

```
test$p.value
```

```
## [1] 0.5458048
```

With a p-value equal to 0.5458048, we clearly do not reject H_0 which, in this simulation does not hold because we simulated data using $\rho = 0.1$. With this sample size and ρ value, What power do we have to reject the null?

Using Monte Carlo (MC) simulations to esitmate the power to detect a non-zero correlation

The task is to use (at least 1000) Monte Carlo simulations to estimate the power to detect a non-zero correlation between variables `x` and `y` from the bi-variate distribution implemented in `simXY()`, at a significance level of 0.05 for `rho=c(.1,.2,.5)` and `n=c(10,20,50,100,300)`.

Suggestions:

- Simulate data using the `simXY()` function provided above.
- Instead of using two loops (as done in the inclass assigments), consider using `expand.grid()`

```
RES=expand.grid(rho=c(.1,.2,.5),sample_size=c(10,20,50,100,300),replicate=1:1000,pValue=NA)
```

Then, you can store the p-values for each MC replicatle

```
for(i in 1:nrow(RES)){
  # simulate data using n=RES$sample_size[i], rho=RES$rho[i]
  # using the simulated data test the correlation using cor.test()
  # store the p-value in RES$pValue[i]
}
```

Finally, you can evaluate the decision rule (reject if p-value<0.05) using

```
# Set
RES$reject=RES$pValue<0.05
```

And, you can estimate power by counting the proportion of times you rejected H0.

```
library(doBy)
summaryBy(reject~rho+sample_size,data=RES)
```

```
##    rho sample_size reject.mean
## 1  0.1          10          NA
## 2  0.1          20          NA
## 3  0.1          50          NA
## 4  0.1         100          NA
## 5  0.1         300          NA
## 6  0.2          10          NA
## 7  0.2          20          NA
## 8  0.2          50          NA
## 9  0.2         100          NA
## 10 0.2         300          NA
## 11 0.5          10          NA
## 12 0.5          20          NA
## 13 0.5          50          NA
## 14 0.5         100          NA
## 15 0.5         300          NA
```

Report

1.1) Report table with your power estimates by value of $\rho_{x,y}$ and sample size.

1.2) For each value of $\rho_{x,y}$ what sample size you would recommend if one wants to achieve a power of at least 0.8?

2) Bootstrap CIs (percentile method)

Consider this data set

```
set.seed(195021)
DATA=simXY(n=100,rho=0.3)
COR=cor(DATA$x,DATA$y)
```

As noted earlier, an approximate SE for the estimated correlation (`COR`) can be obtained using

```
n=nrow(DATA)
SE=sqrt((1-COR^2)/(n-2))
SE
```

```
## [1] 0.09883397
```

Therefore, assuming that the sample correlation follows a normal distribution (this assumption may not be adequate when data does not follow a bi-variate normal distribution and sample size is small and/or the correlation is close to zero, -1 or 1), we can get a 95% CI using

```
 CI=COR+c(-1.96,1.96)*SE
 CI
```

```
## [1] 0.01297567 0.40040484
```

Alternatively we can use `cor.test()` to get a CI using Fisher's z-transform.

```
 cor.test(DATA$x,DATA$y)$conf.int
```

```
## [1] 0.01070681 0.38738157
## attr(,"conf.level")
## [1] 0.95
```

Task: Use the data simulated below and 5000 Bootstrap samples to estimate an approximate 95% CI, for each of the data sets simulated below (DATA_30 and DATA_300).

```
set.seed(195021)
DATA_30=simXY(n=30,rho=.5)
```

```
set.seed(195021)
DATA_300=simXY(n=300,rho=.5)
```

Report

- Your Bootstrap 95% CI for each of the sample sizes
- What is the effect of Sample Size on the CIs??
- Does Boostrap give CIs closer to those of `cor.test()$conf.int` or closer to those of the approximate CI assuming normality?
- Does the differences between the trhee methods diminish with sample size?