

Εργαστηριακή Άσκηση 4

Θέμα: Στατιστική Συμπερασματολογία

1.

```
> Height<-c(3.33, 5.87, 6.31, 1.98, 4.86, 2.65, 2.78, 2.21, 0.45, 1.51, 0.56,
3.68, 2.16, 3.15, 0.84, 3.95, 4.16, 2.99, 2.15, 6.78, 2.84, 4.23, 2.89, 2.56, 2.65,
1.54, 0.98, 0.87)

> Position<-c(1,2,2,2,1,2,1,1,2,1,2,1,1,1,1,1,1,2,2,1,2,1,1,2,2,2)

> Height_north<-Height[Position==1]
> Height_south<-Height[Position==2]
```

 - a.

```
> hist(Height_north)
> hist(Height_north, nclass=8)
> qqnorm(Height_north)
> qqline(Height_north)
> length(Height_north)
> mean (Height_north)
> t.test(Height_north, mu=2)
```
 - b.

```
> hist(Height_south)
> qqnorm(Height_south)
> qqline(Height_south)
> length(Height_south)
> mean (Height_south)
> table(Height_south)
> wilcox.exact(Height_south, mu=1.5, alternative="greater",
conf.level=0.98)
```
 - c.

```
> t.test(Height_north, mu=2, alternative="less", conf.level=0.90)
```
 - d.

```
> boxplot(Height_north, Height_south ,names=c('North Greece','South
Greece'), main="Height (meters) of a particular plant according to its
Geographical Position")
> var(Height_north)
> var(Height_south)
> var.test(Height_north, Height_south)
> t.test(Height_north, Height_south, var.equal=FALSE)
> wilcox.exact(Height_north, Height_south)
```
 - e.

```
> t.test(Height_north, Height_south, var.equal=FALSE, conf.level=0.99)
> wilcox.test(Height_north, Height_south, conf.level=0.99)
```

- f. `> t.test(Height_north, Height_south, var.equal=FALSE,
alternative="greater", conf.level=0.90)
> wilcox.test(Height_north, Height_south, ,alternative="greater",
conf.level=0.90)`
2. a. `> 8*0.5
> binom.test(3,8, p=0.5)
> binom.test(3,8, p=0.5, conf.level=0.98)
> binom.test(3,8, p=0.5, conf.level=0.90)`
- b. `> 80*0.5
> prop.test(30,80, p=0.5)
> prop.test(30,80, p=0.5, conf.level=0.98)
> prop.test(30,80, p=0.5, conf.level=0.90)`
- c. `> 800*0.5
> prop.test(300,800, p=0.5)
> prop.test(300,800, p=0.5, conf.level=0.98)
> prop.test(300,800, p=0.5, conf.level=0.90)`
3. a. `> coin<-c(rep('A', 50), rep('B', 40))
> head<-c(rep('No', 30), rep('Yes', 20), rep('No', 30), rep('Yes', 10))
> T=table(head,coin)
> T
> prop.table(T)
> prop.table(T,1)
> prop.table(T,2)`
- b. `> x<-c(20,10)
> n<-c(50,40)
> p<-sum(x)/sum(n)
> sum(n)*p
> sum(n)*(1-p)
> prop.test(x,n,correct="false")
> prop.test(x,n)`
- c. `> prop.test(x,n, alternative="greater", correct="false")
> prop.test(x,n, alternative="greater")`
- d. `> prop.test(x,n, alternative="less", correct="false")
> prop.test(x,n, alternative="less")`