STA2023 R lab 7 Hypothesis Testing

```
> # z test (means)
> z.test.pvalue<-function(xbar, mu, sigma, n, tails){
             z=(xbar-mu)*sqrt(n)/sigma
   if (z<0) {pv=pnorm(z, lower.tail = T)} else {pv=pnorm(z,lower.tail = F)}
            z<-round(z, 2)
            v1<-c(z, pv)
            v2<-c(z, 2*pv)
            if(tails==1) {return(v1)}
            if (tails==2) {return(v2)}
            }

># example: with xbar=20, mu=21.1, sigma=1.8, n=30, two tailed test
> z.test.pvalue(9.5,10,0.8,20,2) # yields z (test statistic) and pvalue
[1] -2.80  0.005188608
```

```
> # t test means
> t.test.pvalue<-function(xbar, mu, s, n, tails){
         t=(xbar-mu)*sqrt(n)/s
if (t<0) {pv=pt(t, n-1,lower.tail = T)} else {pv=pt(t, n-1,lower.tail = F)}
         t<-round(t, 2)
         v1<-c(t, pv)
         v2<-c(t, 2*pv)
         if(tails==1) {return(v1)}
        if (tails==2) {return(v2)}
}

># example: with xbar=20, mu=21.1, s=1.8, n=20, two tailed test
> t.test.pvalue(20,21.1,1.8,20,2) #  t test with summary
[1] -2.73  0.0132135
```

With raw data:

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, …)
```

```
> x<-c(1, 1.1, 1.11, 1.21,1.3,1.05,1.3,1.31,1.27,1.33,1.29,1.16,0.99,2.3)
> t.test(x, mu=1.1, alternative="two.sided", conf.level = 0.99) # example
        One Sample t-test

data:  x
t = 1.9328, df = 13, p-value = 0.07535
alternative hypothesis: true mean is not equal to 1.1
99 percent confidence interval:
 1.007442 1.523987
sample estimates:
mean of x
 1.265714
```

```
# Z test proportions:
>z.test.prop.pvalue<-function(x, n, p, tails){
                             z=(x/n - p)/sqrt(p*(1-p)/n)
 if (z<0) {pv=pnorm(z, lower.tail = T)} else {pv=pnorm(z,lower.tail = F)}
                             z<-round(z, 2)
                             v1<-c(z, pv)
                             v2<-c(z, 2*pv)
                             if(tails==1) {return(v1)}
                            if (tails==2) {return(v2)}
}
```

```
> #Z example with: x=130, n=200,p=0.60. one tail.
> z.test.prop.pvalue(130,200,0.60,1)
[1] 1.44  0.07445734
```

```
>#using R function prop.test
> prop.test(130,200,0.60, alternative ="greater", correct = F)

        1-sample proportions test without continuity correction

data:  130 out of 200, null probability 0.6
X-squared = 2.0833, df = 1, p-value = 0.07446
alternative hypothesis: true p is greater than 0.6
95 percent confidence interval:
 0.5928573 1.0000000
sample estimates:
   p
0.65
```

```
Notice that:
> sqrt(2.0833) # taking sqrt of X-squared value, retrieve z:
[1] 1.443364
```

Two proportions using prop.test:

```
> prop.test(x=c(x1,x2),n=c(n1,n2), alternative =" ", correct = F)
Example: compare tow proportions. Are they equal? 35/195 and 53/201


> prop.test(x=c(35,53),n=c(195,201), alternative ="two.sided", correct = F)

2-sample test for equality of proportions without continuity correction

data:  c(35, 53) out of c(195, 201)
X-squared = 4.0594, df = 1, p-value = 0.04393
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.165507576 -0.002881249
sample estimates:
   prop 1    prop 2
0.1794872 0.2636816
```

Two samples:

```
> install.packages("BSDA") # two samples t test with summary
> require(BSDA)
> #tsum.test(mean1,s1,n1,mean2,s2,n2,alt="two.sided",conf.level=.95)
> #Example:
> tsum.test(10,1.1,50,11,.99,48,alt="two.sided",conf.level=.95)

        Welch Modified Two-Sample t-Test

data:  Summarized x and y
t = -4.7341, df = 95.609, p-value = 7.617e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.4193131 -0.5806869
sample estimates:
mean of x mean of y
       10        11
```

Similarly, two sample z test with summary:

```
> #zsum.test(mean1,sigma1,n1,mean2,sigma2,n2,alt="two.sided",conf.level=.95)
```

With raw data:

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

```
> var.test(x,y)# F test to compare two variances
```

If, p value does not disprove equality of vars, use classic t-test witch assume equality of the two variances, as follows:

```
>t.test(x, y, var.equal=TRUE)
```