STA2023 Answers to Review 02 questions in R Studio

1.

```
> x<-0:3
> px<-c(0.111,0.215,0.450,0.224)
> library("Weighted.Desc.Stat", lib.loc="~/R/win-library/3.5")
> mean<-w.mean(x,px);sigma<-w.sd(x,px)
> mean;sigma
[1] 1.787
[1] 0.9152218
> min=mean-2*sigma; max=mean+2*sigma
> min;max
[1] -0.04344366
[1] 3.617444
> # The min value is zero (a negative result has no physical meaning). Since
zero is the min usual value, having no obese is not an "unusual" result.
```

2.

```
> y<-c(85, -4)
> py<-c(1/36,35/36)
> w.mean(y,py)
[1] -1.527778
```

3.

```
> dbinom(x=5,size=12,prob=0.25)
[1] 0.1032414
> dbinom(5,12,0.25) # may not include x, size, prob. Just keep the syntax.
[1] 0.1032414
```

4.

```
> # machine works if up to 3 components fail. P(x<=3) given by pbinom:
> pbinom(3,11,0.2)
[1] 0.8388608
```

5.

```
> sum(dbinom(2:6,6,0.5)) # “at least two” adding up probs from 2 to 6.
[1] 0.890625
```

6.

```
> sum(dbinom(72:73,73,.94)) # prob that 72 or 73 passengers show up.
[1] 0.0618183
```

7.

```
> dbinom(2,9,.14)
[1] 0.2454979
```

8.

```
> 17*1/4
[1] 4.25
```

9.

```
> mean=800*0.68;sd=sqrt(800*0.68*0.32)
> mean;sd
[1] 544
[1] 13.19394
> 544-2*13.19;544+2*13.19
[1] 517.62
[1] 570.38
> #494 is less than the minimum usual value (about 518); therefore, it would
be unusual to get 494 consumers who recognize the brand.
```

10.

```
 > punif(5,0,8, lower.tail = F)
 [1] 0.375

 > 1-punif(5,0,8) # alternative
 [1] 0.375
```

11.

```
> pnorm(1.88)-pnorm(-1.88)
[1] 0.9398919
```

12.

```
> pnorm(-1.82, lower.tail=F)
[1] 0.9656205

> 1-pnorm(-1.82) #alternative
[1] 0.9656205
```

13.

```
> pnorm(0.97)
[1] 0.8339768
```

14.

```
> qnorm(0.30,100,15)
[1] 92.13399
```

15.

```
> qnorm(0.75,63.6,2.5)
[1] 65.28622
```

16.

```
> pnorm(300,268,15, lower.tail = F)
[1] 0.0164487
```

17.

```
> pnorm(5.48,5.67,0.070)+pnorm(5.82,5.67,0.070, lower.tail = F)
[1] 0.01938323
> 0.01938323*100
[1] 1.938323
```

Note: Rejected coins: weight less than 5.48 or greater than 5.82

18.

```
> # central limit theorem, sample size not equal to 1. Therefore, we need to
divide the sd by sqrt(n)
> pnorm(72.8,70,10/sqrt(25), lower.tail = F)
[1] 0.08075666
```

19.

```
> pnorm(9.1,9.3,1.1/sqrt(70))
[1] 0.06410482
```

20.

```
> pnorm(98.50,98.20,0.62/sqrt(19))
[1] 0.982534
```

21.

```
> install.packages("epitools")
> library("epitools", lib.loc="~/R/win-library/3.5")
> binom.approx(162,195,conf.level=.95)
    x   n proportion    lower     upper conf.level
1 162 195  0.8307692 0.778142 0.8833965       0.95
```

22.

```
> binom.approx(21,380,conf.level=.95)
   x   n proportion      lower      upper conf.level
1 21 380 0.05526316 0.03228953 0.07823679       0.95
```

23.

```
> # using formula, R as a calculator:
> n=2.575^2*0.25/0.028^2
> ceiling(n) # in order to round up n.
[1] 2115

> n # instead of "ceiling" simply type "n" and round up the result yourself:
[1] 2114.357
So that 2114.357 becomes 2115.
```

```
> # using a function:
> n.sample.prop<-function(conf.level,phat,error){
+     n=(qnorm((1-conf.level)/2))^2*phat*(1-phat)/error^2
+     return(ceiling(n))
+     }
> n.sample.prop(.99,0.5,0.028)
[1] 2116
```

Note:

```
> # the final answer differs by 1 since qnorm(1-alpha/2) is greater that 2.57
5, it is indeed:
> qnorm(1-.01/2)
[1] 2.575829
```

24.

```
> # t conf interval (Using formula, R as a calculator)
> n=27;xbar=76.2;s=21.4; alpha=0.05 # alpha = 1-0.95 =0.05
> se<-qt(1-alpha/2,n-1)*s/sqrt(n)
>
> xbar-se;xbar+se
[1] 67.73444
[1] 84.66556
```

25.

```
> sample<-c(72.2,71.1,74.5,76.6,85.9,77.7)
> t.test(sample, conf.level = .90)
        One Sample t-test

data:  sample
t = 35.173, df = 5, p-value = 3.495e-07
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 71.96027 80.70640
sample estimates:
mean of x
 76.33333
```

26.

```
> sample2<-c(7.3,10.8,9.1,8.4,11.8,7.7,6.4,11.8,10,12.3)
> t.test(sample2, conf.level = .95)

        One Sample t-test

data:  sample2
t = 14.452, df = 9, p-value = 1.558e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  8.063553 11.056447
sample estimates:
mean of x
     9.56
```

27.

```
> # using formula, R as a calculator:
> n=(1.96*60/2)^2  # formula is   n=(z_alpha/2 * sigma/error)^2
> ceiling(n) # rounding up
[1] 3458
```

28.

```
> # t conf interval (Using formula, R as a calculator)
> n=187;xbar=19;s=3.3; alpha=0.052 # alpha = 1-0.98 =0.02
> se<-qt(1-alpha/2,n-1)*s/sqrt(n)
> xbar-se;xbar+se
[1] 18.52805
[1] 19.47195
```