Assignment 3 Solutions

Question 1

Determine the values of the following quantities. (Give answers accurate to 2 decimal places.)

(a) t(.10,19)

(b) t(.05,19)

(c) t(.05,29)

(d) t(.05,50)

(e) t(.005,50)

The t-distribution that WebAssign uses is known as a "one-tailed t-distribution". Thus, the specified probability (α) lies in a single tail, ABOVE the upper critical t-value. The "answer" that WebAssign is looking for here is the upper critical t-value (which is positive).

```
In [13]: import scipy.stats as stats
         import numpy as np

         alpha = .10
         df = 20
         T = stats.t(df)
         print ("a: %.2f" % T.ppf(1-alpha))

         alpha = .05
         T = stats.t(df)
         print ("b: %.2f" % T.ppf(1-alpha))

         df = 28
         T = stats.t(df)
         print ("c: %.2f" % T.ppf(1-alpha))

         df = 60
         T = stats.t(df)
         print ("d: %.2f" % T.ppf(1-alpha))

         alpha = .005
         T = stats.t(df)
         print ("e: %.2f" % T.ppf(1-alpha))
```

```
a: 1.33
b: 1.72
c: 1.70
d: 1.67
e: 2.66
```

Question 2

Determine the t critical value(s) that will capture the desired t-curve area in each of the following cases. (Assume that central areas are centered at t = 0. Round your answers to three decimal places.)

(a) Central area = 0.95, df = 10

(b) Central area = 0.95, df = 30

(c) Central area = 0.99, df = 30

(d) Central area = 0.99, df = 60

(e) Upper-tail area = 0.01, df = 20

(f) Lower-tail area = 0.025, df = 5

```
In [15]: df = 15
         T = stats.t(df)
         central_area = .95
         alpha = 1-central_area

         print ("a: %.3f" % T.ppf(1-alpha/2))

         df = 25
         T = stats.t(df)
         print ("b: %.3f" % T.ppf(1-alpha/2))

         central_area = .99
         alpha = 1-central_area
         print ("c: %.3f" % T.ppf(1-alpha/2))

         df = 50
         T = stats.t(df)
         print ("d: %.3f" % T.ppf(1-alpha/2))

         df = 20
         T = stats.t(df)
         alpha = .01
         print ("e: %.3f" % T.ppf(1-alpha))

         df = 5
         lower_area = .025
         T = stats.t(df)
         print ("f: %.3f" % T.ppf(lower_area))
```

```
a: 2.131
b: 2.060
c: 2.787
d: 2.678
e: 2.528
f: -2.571
```

Question 3

Determine the t critical value for a two-sided confidence interval in each of the following situations. (Give answers accurate to 3 decimal places.)

(a) Confidence level = 95%, df = 17

(b) Confidence level = 95%, df = 18

(c) Confidence level = 99%, df = 18

(d) Confidence level = 99%, n = 3

(e) Confidence level = 98%, df = 38

(f) Confidence level = 99%, n = 61

```python
In [19]: df = 5
         T = stats.t(df)
         cl = .95
         alpha = 1-cl
         print ("a: %.3f" % T.ppf(1-alpha/2))

         df = 28
         T = stats.t(df)
         print ("b: %.3f" % T.ppf(1-alpha/2))

         cl = .99
         alpha = 1-cl
         print ("c: %.3f" % T.ppf(1-alpha/2))

         n = 7
         df = n - 1
         T = stats.t(df)
         print ("d: %.3f" % T.ppf(1-alpha/2))

         df = 36
         cl = .98
         alpha = 1-cl
         T = stats.t(df)
         print ("e: %.3f" % T.ppf(1-alpha/2))

         n = 35
         df = n - 1
         cl = .99
         T = stats.t(df)
         print ("f: %.3f" % T.ppf(1-alpha/2))
```

```
a: 2.571
b: 2.048
c: 2.763
d: 3.707
e: 2.434
f: 2.441
```

Question 4

A random sample of n = 8 E-glass fiber test specimens of a certain type yielded a sample mean interfacial shear yield stress of 32.6 and a sample standard deviation of 3.3. Assuming that interfacial shear yield stress is normally distributed, compute a 95% CI for true average stress. (Give answer accurate to 2 decimal places.)

```python
In [2]: n = 8
        df = n - 1
        xbar = 28.2
        s = 5.0

        cl = .95
        alpha = 1 - cl

        sem = s/np.sqrt(n)

        thigh = stats.t.ppf(1-alpha/2, df)
        tlow = stats.t.ppf(alpha/2, df)

        mulow = xbar + tlow * sem
        muhigh = xbar + thigh * sem

        # print clow, chigh
        print ("(%.02f, %.02f)" % (mulow, muhigh))

        #another way
        c_interval = stats.t.interval(.95, df, loc=xbar, scale=sem)
        print ("(%.02f, %.02f)" % c_interval)
```

```
(24.02, 32.38)
(24.02, 32.38)
```

Question 5

A sample of 14 joint specimens of a particular type gave a sample mean proportional limit stress of 8.06 MPa and a sample standard deviation of 0.66 MPa.

(a) Calculate a 95% lower confidence bound for the true average proportional limit stress of all such joints. (Give answer accurate to 2 decimal places.)

(b) Calculate a 95% lower prediction bound for proportional limit stress of a single joint of this type. (Give answer accurate to 2 decimal places.)

```
In [3]: n = 14
        df = n - 1
        xbar = 8.32
        s = .62

        sem = s/np.sqrt(n)

        # .9 because 5% is on either side of the lower and upper bounds, so 90% is in the middle
        cl = .90

        c_interval = stats.t.interval(cl, df, loc=xbar, scale=sem)

        print ("a. (%.02f, %.02f)" % c_interval)

        s_future = s*np.sqrt((n+1)/n)

        c_interval = stats.t.interval(cl, df, loc=xbar, scale=s_future)
        print ("b. (%.02f, %.02f)" % c_interval)

        a. (8.03, 8.61)
        b. (7.18, 9.46)
```

Question 6

Silicone implant augmentation rhinoplasty is used to correct congenital nose deformities. The success of the procedure depends on various biomechanical properties of the human nasal periosteum and fascia. An article reported that for a sample of 12 (newly deceased) adults, the mean failure strain (%) was 23.5, and the standard deviation was 4.3.

(a) Assuming a normal distribution for failure strain, estimate true average strain in a way that conveys information about precision and reliability. (Use a 95% confidence interval. Give answer accurate to 1 decimal place.)

(b) Predict the strain for a single adult in a way that conveys information about precision and reliability. (Use a 95% prediction interval. Give answer accurate to 1 decimal place.)

How does the prediction compare to the estimate calculated in part (a)?

```
In [12]: n = 17
         xbar = 25.9
         s = 4.3

         sem = s/np.sqrt(n)
         cl = .99

         c_interval = stats.t.interval(cl, df, loc=xbar, scale=sem)
         print ("a. (%.1f, %.1f)" % c_interval)

         s_future = s*np.sqrt((n+1)/n)
         c_interval = stats.t.interval(cl, df, loc=xbar, scale=s_future)
         print ("b. (%.1f, %.1f)" % c_interval)

         a. (22.8, 29.0)
         b. (12.6, 39.2)
```

Question 7

Determine the following. (a) The 97.5th percentile of the chi-squared distribution with v = 5. (Give answer accurate to 2 decimal places.)

(b) The 1th percentile of the chi-squared distribution with v = 5. (Give answer accurate to 2 decimal places.)

(c) $P(13.85 \le \chi2 \le 42.98)$, where χ2 is a chi-squared rv with v = 24. (Give answer accurate to 3 decimal places.)

(d) $P(\chi2 < 14.611$ or $\chi2 > 44.314)$, where χ2 is a chi-squared rv with v = 25. (Give answer accurate to 3 decimal places.)

```
In [5]: df = 5
        print stats.chi2.ppf(0.975, df)

        df = 5
        print stats.chi2.ppf(.01, df)

        df = 24
        print stats.chi2.cdf(42.98, df) - stats.chi2.cdf(13.85, df)

        df = 25
        print 1-(stats.chi2.cdf(44.314, df) - stats.chi2.cdf(14.611, df))

          File "<ipython-input-5-76924b0fde8e>", line 2
            print stats.chi2.ppf(0.975, df)
                      ^
        SyntaxError: invalid syntax
```

Question 8

The amount of lateral expansion (mils) was determined for a sample of n = 9 pulsed-power gas metal arc welds used in LNG ship containment tanks. The resulting sample standard deviation was s = 2.86 mils. Assuming normality, derive a 95% CI for σ2 and for σ. (Round your answers to two decimal places.)

Recall the definition of $\chi^2 = \frac{(n-1)s^2}{\sigma^2}$. Therefore, to find the CI for σ^2, we can use that:

$$\sigma^2 = \frac{(n-1)s^2}{\chi^2}$$

and thus use the confidence interval for χ^2.

```
In [ ]: n = 9
        df = n - 1
        s = 2.86
        cl = .95

        c_interval = stats.chi2.interval(cl, df)
        print c_interval 1 and 2

        #calculate variance interval
        var_high = df*s**2/c_interval[0]
        var_low = df*s**2/c_interval[1]

        print var low, var high
        #std deviation interval
        print np.sqrt(var_low), np.sqrt(var_high)
```

Question 9

The following observations were made on fracture toughness of a base plate of 18% nickel maraging steel (in ksi √in., given in increasing order)].

68.0 71.9 72.6 73.1 73.3 73.5 75.5 75.7 75.8 76.1 76.2

76.2 77.0 77.9 78.1 79.6 79.6 79.9 80.1 82.2 83.7 90.3

Calculate a 99% CI for the standard deviation of the fracture toughness distribution. (Give answer accurate to 2 decimal places.)

```
In [ ]: data = np.array([])
        print data.mean(), data.std(ddof=1)
```

```
In [ ]: n = len(data)
        df = n - 1
        s = data.std(ddof=1)
        cl = .99

        c_interval = stats.chi2.interval(cl, df)
        #chi 2 interval
        print c_interval array

        var_high = df*s**2/c_interval[0]
        var_low = df*s**2/c_interval[1]

        #std deviation interval
        print (np.sqrt(var_low), np.sqrt(var_high))
```

```
In [ ]:
```

```
In [ ]:
```

Question 10

On the basis of extensive tests, the yield point of a particular type of mild steel-reinforcing bar is known to be normally distributed with σ = 100. The composition of the bar has been slightly modified, but the modification is not believed to have affected either the normality or the value of σ.

(a) Assuming this to be the case, if a sample of 25 modified bars resulted in a sample average yield point of 8312 lb, compute a 90% CI for the true average yield point of the modified bar. (Round to the nearest integer.)

(b) How would you modify the interval in part (a) to obtain a confidence level of 92%? (Give answer accurate to 2 decimal places.)

```
In [ ]: n = 25
        sigma = 100
        xbar = 8312
        cl = .9

        sem = sigma/np.sqrt(n)

        # z interval
        c_interval = stats.norm.interval(cl, loc = 0, scale = 1)
        print ("z interval: (%.2f, %.2f)" % c_interval)

        #confidence interval of the mean
        c_interval = stats.norm.interval(cl, loc=xbar, scale=sem)
        print ("a: Confidence Interval: (%.2f, %.2f)" % (c_interval)

        cl = .92
        #z interval
        c_interval = stats.norm.interval(cl, loc=0, scale=1)
        print ("z interval: (%.2f, %.2f)" % c_interval)

        #confidence interval of the mean
        c_interval = stats.norm.interval(cl, loc=xbar, scale=sem)
        print ("b: Confidence Interval: (%.2f, %.2f)" % (c_interval)
```

```
In [ ]:
```