Statistics PSET 2

March 13, 2018

1 40.004 Statistics 2018: Problem Set 2

due Monday, 12 March, 2018 at 11:59 pm. Submit on e-dimension.

Adam 1002010

```
In [1]: from scipy import stats
        from scipy import stats
        import numpy as np
```

Question 1:

(A twist on the German Tank Problem). Suppose that the enemy has tanks numbered 0, 1, 2, . . . , N. You observe n of the tanks with replacement at random and note down their numbers. Using the sample mean of these numbers, find an unbiased estimator for the total number of tanks (with justification for your claim).

θ : Total Number of Tanks = N + 1 Let n be sample size $E[\bar{X}] = \frac{1}{n}(E[X_1] + E[X_2] + .. + E[X_n])$

$E[X_i] = \frac{1}{N+1}(0 + 1 + 2 + ... + N)$

$(0 + 1 + 2 + ... + N) = \frac{N}{2}(N + 1)$

$E[\bar{X}] = \frac{N}{2}$

$2E[\bar{X}] = N$

$2E[\bar{X}] + 1 = N + 1$

$E[2\bar{X} + 1] = N + 1$

$\hat{\theta} = 2\bar{X} + 1$

Question 2:

Let S^2 denote the sample variance computed from a random sample of size n from a N (ţ, ^{2}) distribution. Find the probability that the sample variance S^2 exceeds the true variance 2 by a factor of two, i.e., Pr($S^2 \leq 2^2$) when n = 8, 17, 21. Comment on your results. You may use R or Excel or a standard table in the book to find the probabilities.

$Pr(S^2 \leq 2\sigma^2)$

$Pr(\frac{S^2}{2\sigma^2} \leq 1)$

$Pr(\frac{S^2}{2\sigma^2} \leq 1)$

$Pr(\frac{2(n-1)S^2}{2\sigma^2} \geq 2(n-1))$

$\frac{(n-1)S^2}{2\sigma^2} \sim \chi^2_{n-1}$

$P(\chi^2_{n-1} \geq 2(n-1))$

$1 - P(\chi^2_{n-1} \geq 2(n-1))$

```
In [2]: for i in [8,17,21]:
            print("When n = {}".format(i))
            prob = 1-stats.chi2.cdf(2*(i-1), df=i-1)
            print("probability that the sample variance exceeds twice the true variance: {} \n

When n = 8
probability that the sample variance exceeds twice the true variance: 0.0511813534130654

When n = 17
probability that the sample variance exceeds twice the true variance: 0.009999780953104831

When n = 21
probability that the sample variance exceeds twice the true variance: 0.0049954123083075785
```

Question 3:
A random sample of size 100, drawn from a normal distribution, has sample mean x = 16.3.

(a) Calculate the 95% two-sided confidence interval for ţ, if = 6.

```
In [3]: # PART A
        # Calculate the 95% two-sided confidence interval for ţ, if  = 6.
        # normal distribution, using CLT since n is large
        n = 100
        sample_mean = 16.3
        sigma = 6
        z_lower = stats.norm.ppf(0.025) # away from mean
        z_upper = stats.norm.ppf(0.975) # away from mean
        lower_bound, upper_bound = sample_mean+z_lower*sigma/n**0.5, sample_mean+z_upper*sigma/
        print("Lower bound is: {}".format(lower_bound))
        print("Upper bound is: {}".format(upper_bound))

Lower bound is: 15.124021609275967
Upper bound is: 17.475978390724034
```

(b) Calculate the 95% two-sided confidence interval for ţ, if s = 6 and is unknown.

```
In [4]: # PART B
        # Calculate the 95% two-sided confidence interval for ţ, if s = 6 and  is unknown.
        # since sigma is unknown, use t distribution
        df = n - 1
        #degree of freedom
        t_lower = stats.t.ppf(0.025, df=n-1) # away from mean
        t_upper = stats.t.ppf(0.975, df=n-1) # away from mean
        print("Lower bound is: {}".format(sample_mean+t_lower*sigma/n**0.5))
        print("Upper bound is: {}".format(sample_mean+t_upper*sigma/n**0.5))
```

```
Lower bound is: 15.109469829094792
Upper bound is: 17.49053017090521
```

(c) Calculate the upper and lower 95% one-sided confidence intervals for ţ, if s = 6 and is unknown.

```
In [5]: #PART C
        #Calculate the upper and lower 95% one-sided confidence intervals for population mean,
        t_lower = stats.t.ppf(0.05, df=n-1) # away from mean
        t_upper = stats.t.ppf(0.95, df=n-1) # away from mean
        print("Lower bound is: {}".format(sample_mean+t_lower*sigma/n**0.5))
        print("Upper bound is: {}".format(sample_mean+t_upper*sigma/n**0.5))
```

```
Lower bound is: 15.303765306402166
Upper bound is: 17.296234693597835
```

(d) Why is the confidence interval in (b) wider than the CI in (a)?

Answer: When the population variance is unknown, we estimate it using the sample variance. Hence, to allow for this uncertainty in part (b), we should expect a wider interval. As such we use t-distribution and this also explains why t-distribution has heavier tails than normal distribution (to account for the uncertainty)

Question 4:

Let $X_1, . . . , X_n$ be a random sample from a distribution with mean μ and variance ^{2}. Show that $E(\bar{X}^2)$ is a biased estimator of μ^2.

if $E[\bar{X}^2]$ is an unbiased estimator of μ^2, then

$$E[\bar{X}^2] - \mu^2 = 0$$

However,

$$Var(X) = E[\bar{X}^2] - E[\bar{X}]^2$$

$$Var(X) = E[\bar{X}^2] - \mu^2$$

unless X is a constant random variable (which is not) since it follows a distribution as such, $Var(X) \neq 0$

Thus, $E[\bar{X}^2]$ is an biased estimator of μ^2

Question 5

In each of the following cases, state the two competing hypotheses that should be tested and specify which would you set up as the null hypothesis and which one as the alternative hypothesis. Explain your choice briely (a) A consumer watchdog group suspects that a yogurt advertised to be 98% fat free has actually a higher fat content. The group plans to measure the fat contents of 25 yogurt cups (each containing 170 grams) to verify its suspicion.

Let x be percentage of the yogurts mass that is not fats $H_0 : x \geq 98$ \$H_A : x < 98 \$

(b) It is claimed that cloud seeding is an effective technique technique to increase precipitation. \$H_0 : \$ Cloud seeding is an effective technique \$H_A : \$ Cloud seeding is not an effective technique

Question 6:

Consider testing $H_0 : \mu = 0$ vs $H_A : \mu \neq 0$ based on a random sample of size n from a $N(\mu, 1)$ distribution. (a) Calculate the p-values for the following three cases: (i) x = 0.1, n = 100; (ii) x = 0.1, n = 400; (iii) x = 0.1, n = 900. (b) Given the significance level $\alpha = 0.01$, conduct the hypothesis tests for the three cases in (a).

```
In [6]: def normal_hypothesis_test(x,mu,sigma,n, alpha):
            z = (x - mu)/(sigma/n**0.5)
            prob = 1-(stats.norm.cdf(z)-stats.norm.cdf(-z))
            print("Probability of getting value of x at least as extreme as {} at n = {} is \n
            if prob>=alpha:
                print("Since {} >= {} \nwe do not reject our hypothesis".format(prob, alpha))
                return True
            else:
                print("Since {} < {} \nwe reject our hypothesis".format(prob, alpha))
                return False
```

```
In [7]: normal_hypothesis_test(x=0.1, mu=0, n=100, sigma=1, alpha=0.01)
```

```
Probability of getting value of x at least as extreme as 0.1 at n = 100 is
0.31731050786291415
Since 0.31731050786291415 >= 0.01
we do not reject our hypothesis
```

```
Out[7]: True
```

```
In [8]: normal_hypothesis_test(x=0.1, mu=0, n=400, sigma=1, alpha=0.01)
```

```
Probability of getting value of x at least as extreme as 0.1 at n = 400 is
0.04550026389635842
Since 0.04550026389635842 >= 0.01
we do not reject our hypothesis
```

```
Out[8]: True
```

```
In [9]: normal_hypothesis_test(x=0.1, mu=0, n=900, sigma=1, alpha=0.01)
```

```
Probability of getting value of x at least as extreme as 0.1 at n = 900 is
0.002699796063260207
Since 0.002699796063260207 < 0.01
we reject our hypothesis
```

```
Out[9]: False
```

Question 7

A tire company has developed a new tread design. To determine the newly designed tire has a mean of 60,000 miles or more, a random sample of 16 prototype tires are tested. The mean life for this sample is 60,758 miles. Assume that the tire life is normally distributed with unknown ţ and standard deviation = 1500 miles. Test the hypothesis $H_0 : = 60,000$ vs. $H_A : > 60,000$

a) Compute the test statistic and the p-value. Based on the p-value, state whether H0 can be rejected at = 0.01.

```
In [10]: # assume h 0 is true,
         mu = 60000
         n = 16 # sample size
         x = 60758 # sample mean
         sigma = 1500
         test_statistic = (x - mu)/(sigma/ (n-1)**0.5)
         print("Test Statistic: {}".format(test_statistic))
         prob = stats.norm.pdf(test_statistic)
         status = "Reject" if prob < 0.01 else "Do not reject"
         print("p-value: {} ({})".format(prob, status))
```

```
Test Statistic: 1.9571475842834813
p-value: 0.05876834810545223 (Do not reject)
```

b) What is the power of the 0.01-level test in (a) if the true mean life for the new tread design is 61,000 miles?

$P(\frac{x-\mu}{\sigma/\sqrt{n-1}} < Z) = 0.01$
x is the critical value to reject at 0.01 significance level

```
In [11]: critical_value = stats.norm.ppf(0.99)*sigma/(n-1)**0.5+mu
         print("Critical value: {}".format(critical_value))
         beta = stats.norm.cdf((critical_value - 61000)/ sigma/(n-1)**0.5)
         print("Power of test: {}".format(1-beta))
```

```
Critical value: 60900.99065736452
Power of test: 0.5067987384008006
```

c) Suppose that at least 90% power is needed to identify a tread design that has the mean life of 61, 000 miles. How many tires should be tested?

```
In [12]: alpha = 0.01
         beta = 0.1
         n = ((sigma*stats.norm.ppf(1-alpha)+stats.norm.ppf(1-beta)) / (mu-61000))**2
         print("At least {} tires are to be tested".format(np.ceil(n)))
```

```
At least 13.0 tires are to be tested
```

Question 8
Two methods of measuring the atomic weight of carbon (the nominal atomic weight is 12) yielded the following results.

```
In [13]: method_1 = [float(i) for i in "12.0129 12.0072 12.0064 12.0054 12.0016 11.9853 11.994
         method_2 = [float(i) for i in"12.0318 12.0246 12.0069 12.0006 12.0075".split(" ")]
         print("Method 1: {}".format(method_1))
         print("Method 2: {}".format(method_2))
```