STAT 345-01: Nonparametric Statistics

Problem Set 2

Assigned 2018 September 4
Due 2018 September 11

Show your work on all problems! Be sure to give credit to any collaborators, or outside sources used in solving the problems. Note that if using an outside source to do a calculation, you should use it as a reference for the method, and actually carry out the calculation yourself; it's not sufficient to quote the results of a calculation contained in an outside source.

Please hand in parts one and two separately.

1 Part One

1.1 Conover Problems on Hypothesis Testing

Exercise 2.3.6

Problem 2.3.1

Review Problem 2.6.12

1.2 Conover Problems on Binomial Proportion

Exercise 3.1.6

Exercise 3.1.12

Problem 3.1.1

2 Part Two

Please turn in some sort of transcript of your python session, along with answers to the questions posed. If you want to submit electronically, please send either a pdf or a plain ASCII file. No Word documents!

2.1 Estimating the Power Curve for a Two-Tailed t-test

Consider a sample of size $n = 40$ drawn from a normal distribution of mean μ and variance σ^2. A two-tailed t test of significance $\alpha = .09$ rejects the null hypothesis H_0: $\mu = 0$ in favor of the alternative hypothesis $\mu \neq 0$ when

$$\left|\frac{\overline{x}}{s/\sqrt{n}}\right| \geq t_{0.955} \tag{2.1}$$

Determine the power curve when the (sampling distribution is normal) numerically as follows.

(a) Generate $N = 10^3$ samples of $n = 40$ points each from a standard normal distribution $N(0,1)$ as in last week's problem set, and determine the sample mean and sample standard deviation of each (which should be stored in N-point vectors `xbar_j` and `s_j`).

(b) Create a vector of μ values with

```
mu = np.linspace(-1,1,101)
```

(c) Explain why we can use a sample drawn from $N(0,1)$ as a "stand-in" for sample drawn from $N(\mu,1)$ by making the transformation $\overline{x} \to \overline{x} + \mu$ and $s \to s$. (This means we won't have to re-generate the thousand 40-point samples for each value of μ; we can just adjust the thousand $\overline{x}$ and s values and use those to construct the test statistic.)

(d) With the use of the construction `mu[:,None] + xbar_j[None,:]`, produce a 101×10^3 array of $t = \frac{\overline{x}}{s/\sqrt{n}}$ values.

(e) For each of the 101 μ values, find the fraction of t scores which lie in the critical region $|t| \geq t_{0.955}$, using a construction like

```
gamma = np.mean(np.abs(tscore) >= tcrit,axis=-1)
```

(f) Plot $\gamma(\mu)$ versus μ, and verify that $\gamma(0) = \alpha$.

2.2 Confidence Interval for Proportion

Consider the Clopper-Pearson confidence interval for population proportion, as tabulated in Table A4 of Conover and calculated by[1]

```
def ClopperPearsonCI(CL,n,x):
    tailprob = 0.5*(1.-CL)
    lower = stats.beta.ppf(tailprob,x,n-x+1)
    upper = stats.beta.isf(tailprob,x+1,n-x)
    return (lower,upper)
```

(a) Suppose we have a binomial experiment with $n = 30$ trials. For what values of x, the number of successes, does the 90% confidence interval contain $p = 0.20$?

(b) What is the total probability of one of these x values occurring? Compare this actual confidence level to the requested confidence level of 90%.

(c) Repeat the calculation for a confidence level of 97% and a true proportion of $p = 0.35$. (You'll have to use software for this, since these values are not in the tables.

[1]Note that this function can give `nans` if one endpoint of the confidence interval us at 0 or 1. You can remove these with commands like `if np.isnan(lower): lower = 0.` and `if np.isnan(upper): upper = 1.` or `lower[np.isnan(lower)]=0.` and `upper[np.isnan(upper)]=1.`, depending on whether `lower` and `upper` are arrays or not.