Quiz 7

March 26, 2020

[1]:
```
# importing the necessary library
import numpy as np
import pandas as pd

from scipy.stats import binom

# mathematical constants
from math import e
```

[2]:
```
# load the data
X = pd.Series(np.sort([1, 6, 1, 4, 7, 8, 4, 6, 5, 2,
                       1, 6, 5, 3, 1, 8, 7, 8, 1, 2,
                       6, 1, 6, 4, 9, 2, 6, 9, 6, 9,
                       5, 2, 3, 6, 6, 9, 2, 8, 7, 3]))

# X now represents the ordered sample
```

[3]:
```
# to compare X to the exponential distribution, we will use the Cramer-Von␣
 ↪Mises Test
# thus, we will need a function for the exponential CDF
def exponentialCDF(x, theta):
    return (1 - e**(-x / theta))

# the exponentialCDF function defined above needs the parameter theta
# we can approximate theta using the Maximum Likelihood Estimator for the␣
 ↪Exponential Distribution
```

[4]:
```
# now we can define a function for calculating the Cramer-Von Mises Test␣
 ↪Statistic

def CVM_Test(X, distributionParameter):

    # get the length of this ordered list
    n = len(X)

    # initialize a variable to store the sum of squared differences
    sumTotal = 0
    # calculate the sum of squared distances
    for i in range(n):
        ithTerm = (i+0.5)/n # i goes begins at 0, so (i+1) - 0.5 = i+0.5
        sumTotal += (exponentialCDF(X[i], distributionParameter) - ithTerm)**2

    # return the test statistic
    return (1 / (12 * n)) + sumTotal
```

[5]:

```
# let's test this function on Example 13.8.2

testX = pd.Series(np.sort([5.2, 8.4, 0.9, 0.1, 5.9,
                           17.9, 3.6, 2.5, 1.2, 1.8,
                           1.8, 6.1, 5.3, 1.2, 1.2,
                           3.0, 3.5, 7.6, 3.4, 0.5,
                           2.4, 5.3, 1.9, 2.8, 0.1]))

CVM_Test(testX, 3.7)
```

[5]:

```
0.05093701139921777
```

[6]:

```
# now that we know the function is correct...
# lets answer question #1 using alpha = 0.05

# MLE for exponential distribution is n/sum(Xi)
mle = sum(X) / len(X)
```

[7]:

```
# now calculate the Cramer-Von Mises Test Statistic
observedCM = CVM_Test(X, mle)
```

[8]:

```
# then since we used an approximation for the parameter of the distribution␣
 ↪under the null,
# we need to apply Stephen's modifications
observedCM = (1 + 0.16 / len(X)) * observedCM
```

[9]:

```
# Let's compare the result to critical value for alpha = 0.5
print("Question #1:")
print("")
print(observedCM, ">", 0.225, "=", observedCM>0.225)

# print the results
if (observedCM>0.255):
    print("Thus we reject the null.")
else:
    print("Thus we fail to reject the null.")
```

```
Question #1:
```

```
0.6066946310765385 > 0.225 = True
Thus we reject the null.
```

[10]:

```python
# Now lets make a function for the sign test
def sgnTest(X, m, alternative='greater'):

    # count the observations less than m
    t = len([x for x in X if x < m])

    # calculate n
    n = len(X)

    if alternative=='greater':
        return binom.cdf(t, n, 0.5)
    elif alternative=='smaller':
        return 1 - binom.cdf(t-1, n, 0.5)
```

[11]:

```python
# now lets get our observed test statistic for question 2
observedSgn = sgnTest(X,7,'smaller')
```

[12]:

```python
# Let's compare the result to critical value for alpha = 0.1
print("Question #2:")
print("")
print(observedSgn, "<", 0.1, "=", observedSgn<0.1)

# print the results
if (observedSgn<0.1):
    print("Thus we reject the null.")
else:
    print("Thus we fail to reject the null.")
```

```
Question #2:

0.0032132880478457O8 < 0.1 = True
Thus we reject the null.
```