Intermediate Statistics HW1 Q6

April 15, 2019

```
In [18]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
```

Input the observation values

```
In [12]: obs = pd.Series([0.5160 , 0.2223, -0.1221, 0.1085, 0.0012, -0.0764, -0.0150, 0.0816,0.2
```

1 6-(a)

Make the $max(r_i, 0)$ function

```
In [13]: def max0(x):
             if x>0:
                 return x
             else:
                 return 0
```

Calculate the sample mean

```
In [29]: sample_mean = obs.map(max0).mean()
         print('Sample mean is',sample_mean)
```

```
Sample mean is 0.11785000000000001
```

calculate the sample standard deviation

```
In [30]: sample_sd = obs.map(max0).std()
         print('Samle standard dviation is',sample_sd)
```

```
Samle standard dviation is 0.1690399443261201
```

as we know that sample variance $\hat{\sigma}^2$ follows that

$$\frac{n\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-1}$$

so we solve the inequality

$$2.70 < \frac{10 \times 0.1690^2}{\sigma^2} < 19.02$$

then, 95% C.I for σ^2 is

$$0.0150 < \sigma^2 < 0.1058$$

then, 95% C.I for σ is

$$0.1225 < \sigma^2 < 0.3253$$

2 6-(b)

first calculate the sample mean and sample standard deviation

```
In [34]: mu_hat = obs.mean()
         sd_hat = obs.std()
```

```
In [35]: mu_hat
```

```
Out[35]: 0.07718
```

```
In [36]: sd_hat
```

```
Out[36]: 0.20943995055597414
```

$$\hat{\mu} = 0.07718 \text{ , } \hat{\sigma} = 0.20944$$

We sampling the 10 iid observation$\{x_1, x_2, \cdots x_{10}\}$ and calculate the sample standard error of $max(x_i, 0)$.
Iterate it 10,000 times and sort
the 2.5% quantile and 97.5% quantile will be 95% C.I.

```
In [99]: mc = []
         np.random.seed(0)
         for i in range(10000):
             mc.append(pd.Series(np.random.normal(loc=mu_hat ,scale=sd_hat,size=10)).map(max0).m
```

```
In [100]: sorted_mc = sorted(mc)


          sorted_mc_np = pd.Series(sorted_mc)
          print('95% C.I of r E[X]: ',(sorted_mc_np[249], sorted_mc_np[9749]))
          print('Standard Error of E[x] : ' ,sorted_mc_np.std() )

          sorted_mc_exp = sorted_mc_np.map(lambda x: np.exp(x))
          print('95% C.I of r exp(E[X]): ',(sorted_mc_exp[249], sorted_mc_exp[9749]))
          print('Standard Error of E[x] : ' ,sorted_mc_exp.std() )
```

```
95% C.I of r E[X]:  (0.04590309537487885, 0.2245181261391263)
Standard Error of E[x] :  0.04603271224858704
95% C.I of r exp(E[X]):  (1.0469729495197901, 1.251719399982383)
Standard Error of E[x] :  0.05279352767153702
```

3 6-(c)

Sampling with replacement calculate the sample standard error of $max(x_i, 0)$.
Iterate 10,000 times and sort
The 2.5% quantile and 97.5% quantile will be 95% C.I.

```
In [101]: bts = []
          np.random.seed(0)
          for i in range(10000):
              bts.append(pd.Series(np.random.choice(obs,size=10,replace=True)).map(max0).mean())
```

```
In [102]: sorted_bts = sorted(bts)
          sorted_bts_np = pd.Series(sorted_bts)
          print('95% C.I of standard error : ',(sorted_bts_np[249], sorted_bts_np[9749]))
          print('Standard Error of E[x] : ' ,sorted_bts_np.std() )

          sorted_bts_exp = sorted_bts_np.map(lambda x: np.exp(x))
          print('95% C.I of r exp(E[X]): ',(sorted_bts_exp[249], sorted_bts_exp[9749]))
          print('Standard Error of E[x] : ' ,sorted_bts_exp.std() )
```

```
95% C.I of standard error :  (0.03075, 0.22875)
Standard Error of E[x] :  0.05091737966093325
95% C.I of r exp(E[X]):  (1.031227664741787, 1.2570277427887513)
Standard Error of E[x] :  0.058034644566414016
```

4 6-(d)

We can sample data with out 1 obs, then we can get the 10 sampled vector
Also we can get 10 sample standard error
so, by averaging it we can get the estimator of standard error

```
In [105]: jk = []
          for i in range(10):
              jk.append(obs.drop(i).map(max0).mean())
          print('Jackknife Estimator : ',np.array(jk).mean())
          print('Jackknife Estimator standard error : ',np.array(jk).std())
```

```
Jackknife Estimator :  0.11785000000000001
Jackknife Estimator standard error :  0.017818374653953202
```

5 6-(e)

As jacknife can make n samples it is not suffient to estimate the parameter
Since it does not neet to assume the distribution, I'll use bootstrap