# Homework 1 answer sheet

Isaac Ng, 1002174, Cohort 1

February 12, 2018

## 1

### 1.1

It is likely to be biased. For example, if the question was “Are you commuting”, the answer is likely to be biased to “Yes”

## 2

### 2.1

Control group: the people after the period of time required to clear aspartame. Treatment group: the people at the start of the experiment

### 2.2

Any problems with the sampling method that may have caused the groups to be significantly different (e.g. if a lot more people in the treatment sample were sensitive to aspartame) will affect findings. An experiment using crossover design can eliminate the possibility of this happening.

### 2.3

No. A “higher than average” rate still may not be significant.

## 3

### 3.1

Systematic sampling.

### 3.2

Stratified sampling

### 3.3

Simple Random Sampling

## 4

### 4.1

The data appears skewed to the right.

| | | | | |
|---|---|---|---|---|
| 2.860 | 6.205 | 6.405 | 6.700 | 7.000 |
| min | 25% | median | 75% | max |

Table 1: 5-bar summary of data

### 4.2

6.4335

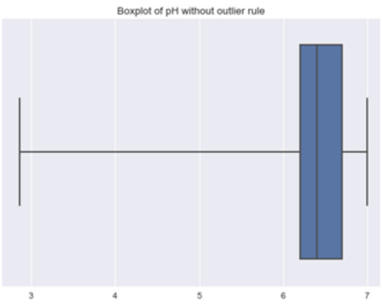
Yes. The trimmed mean is noticeably different from the sample mean of 6.297

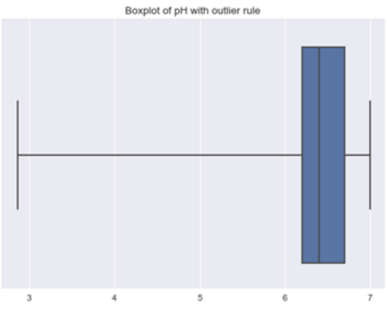
### 4.3

$IQR = 6.70 - 6.205 = 0.495$

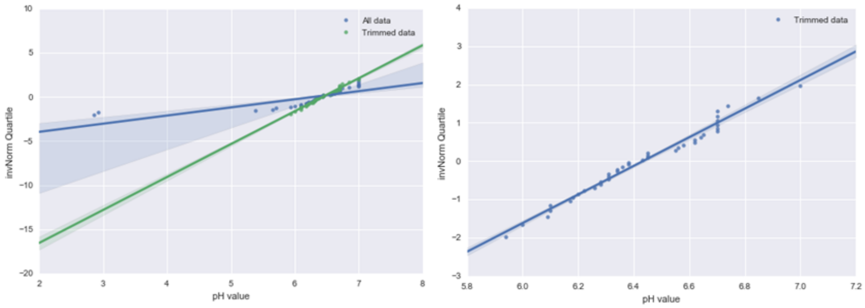
$\sigma = 0.788$

### 4.4





### 4.5



The trimmed data looks normal, while the original data looks skewed.

## 5

### 5.1

$$p = q\pi + \frac{1-q}{12} \tag{1}$$

$$\pi = \frac{p - \frac{1-q}{12}}{q} \tag{2}$$

### 5.2

Yes, it is possible

### 5.3

q=1 or q=0 would not be helpful

## 6

### 6.1

Hospital A is better in both low risk and high risk procedures.

| | Low risk | High risk |
|---|---|---|
| Hospital A | 400/500 = 0.8 | 160/800 = 0.2 |
| Hospital B | 300/500 = 0.6 | 20/200 = 0.1 |

### 6.2

Hospital B is better overall.

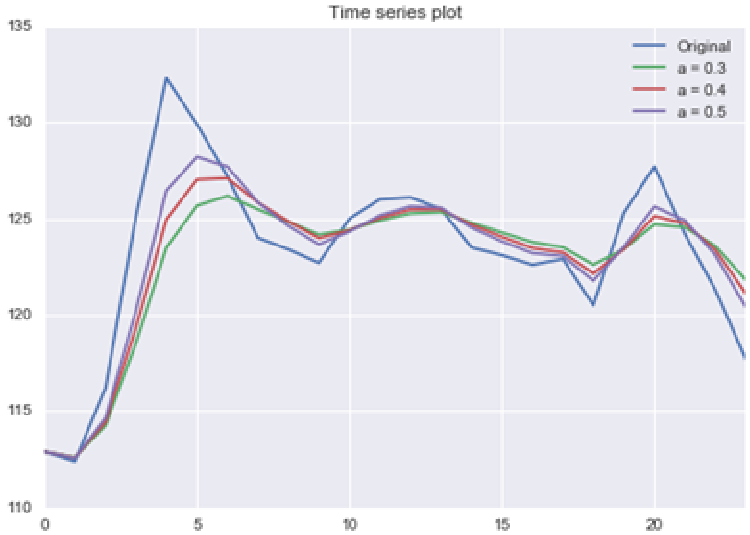
| | |
|---|---|
| Hospital A | $\frac{400+160}{500+800} = 0.4308$ |
| Hospital B | $\frac{300+20}{500+200} = 0.4571$ |

### 6.3

This is Simpson's paradox. As high risk procedures have lower success rates, Hospital A appears worse on the whole because it takes on much more high-risk cases than Hospital B.

## 7

### 7.1



### 7.2

| $\alpha$ | 0.3 | 0.4 | 0.5 |
|---|---|---|---|
| MAPE | 1.74% | 1.46% | 1.22% |

Table 2: MAPE for varying $\alpha$

As the MAPE for $\alpha = 0.5$ is the lowest, we use it to forecast for the next period. Hence our prediction is 120.4

# 8

## 8.1

$$
\begin{aligned}
Var(X) &= E(X^2) = [E(X)^2] \\
Var(X) &= E(X^2) - [E(X)^2] \\
E(X^2) &= \frac{1+4+9+16}{4} \\
&= 7.5 \\
E(X)^2 &= (\frac{1+2+3+4}{4})^2 \\
&= 6.25 \\
Var(X) &= 7.5 - 6.25 \\
Var(X) &= 1.25
\end{aligned}
$$

## 8.2

They are the same.

## 8.3

(calculated with scipy)1.25 They are the same.

# 9

## 9.1

$$
\begin{aligned}
U\ Norm(40, \sqrt{\frac{15^2}{50}}) \\
V\ Norm(40, \sqrt{\frac{15^2}{100}})
\end{aligned}
$$

## 9.2

V. V is the average of a larger number of samples, meaning it has less variance, meaning it is more likely to fall within a range about the distribution mean.

## 9.3

$$
\begin{aligned}
P(35 < U < 45) = 0.982 \\
P(35 < V < 45) = 0.999
\end{aligned}
$$

```
# -*- coding: utf-8 -*-
"""
Created on Sat Feb 03 23:47:29 2018

@author: Isaac Ng
"""
#%% imports
from __future__ import print_function
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import itertools
from scipy.stats import norm
#%% q4
ls = [6.10, 6.74, 6.22, 5.65, 6.38, 6.70, 7.00, 6.43, 7.00, 6.70,
 ↪  6.70, 5.94, 6.28, 6.34, 6.62, 6.55, 2.92, 6.10, 6.20, 6.70,
 ↪  7.00, 6.85, 6.31, 6.26, 6.36, 6.28, 6.38, 6.70, 6.62, 7.00,
 ↪  6.45, 6.31, 2.86, 6.31, 6.09, 6.17, 6.64, 6.45, 7.00, 6.18,
 ↪  6.58, 5.38, 6.34, 7.00, 5.70, 6.65, 6.56, 6.00, 6.70, 6.45]
ls.sort()
df = pd.DataFrame(ls)

#4a
print("5 number summary:")
print(df.describe())
print("\n")

#4b
trimmedls = ls[5:45]
trimmeddf = pd.DataFrame(trimmedls)
print("Trimmed mean = {}".format(float(trimmeddf.mean())))
print("Difference in mean = {}".format(float(trimmeddf.mean() -
 ↪  df.mean())))

#4c
print("IQR =
 ↪  {}".format(float(df.quantile(0.75)-df.quantile(0.25))))
print("Standard deviation = {:.3f} (3dp)".format(np.std(ls, ddof
 ↪  = 1)))

#4d
bp = sns.boxplot(ls, whis = 100).set_title("Boxplot of pH without
 ↪  separating outliers")
```

```
plt.show()
plt.clf()
bp = sns.boxplot(ls, whis = 1.5).set_title("Boxplot of pH,
 ↪  separating outliers")
plt.show()
plt.clf

#4e
ls2 = zip([norm.ppf(float(i)/51) for i in range(1,51)], ls)
y = pd.Series([norm.ppf(float(i)/51) for i in range(1,51)])
y.name = ("invNorm Quartile")
x = pd.Series(ls)
x.name = ("pH value")
qqplot = sns.regplot(x,y, label = "All data")
plt.show()
plt.clf()

ls2 = zip([norm.ppf(float(i)/41) for i in range(1,41)],
 ↪  trimmedls)
y = pd.Series([norm.ppf(float(i)/41) for i in range(1,41)])
y.name = ("invNorm Quartile")
x = pd.Series(trimmedls)
x.name = ("pH value")
qqplot = sns.regplot(x,y, label = "Trimmed data")
#plt.show()
qqplot.legend(loc = "best") #if you leave out the plt.show it
 ↪  plots it on the same axis. but then you can't look at trimmed
 ↪  data in detail.
plt.show()
print("The trimmed data looks normal, while original data looks
 ↪  skewed.")

#%% q7

#7a is so loopable but i'm sleepy
ls = [112.9, 112.4, 116.2, 125.1, 132.3, 129.9, 127.2, 124.0,
 ↪  123.4, 122.7, 125.0, 126.0, 126.1, 125.5, 123.5, 123.1,
 ↪  122.6, 122.9, 120.5, 125.3, 127.7, 124.2, 121.3, 117.7]
df = pd.DataFrame(ls)
ls2 = ls[1:]

ewma1 = pd.ewma(df, alpha = 0.3)
ewma1 = pd.Series(ewma1[0])

ewma2 = pd.ewma(df, alpha = 0.4)
ewma2 = pd.Series(ewma2[0])
```

```
ewma3 = pd.ewma(df, alpha = 0.5)
ewma3 = pd.Series(ewma3[0])

df.drop(0,0)
df.columns = ["Original"]

df['a = 0.3'] = ewma1
df['a = 0.4'] = ewma2
df['a = 0.5'] = ewma3

df.plot(title = "Time series plot")

#7b
n = len(ls2)-1
ape1 = 0
for idx, actual in enumerate(ls2):
    ape1 += abs((actual-ewma1[idx+1])/actual)
    mape1 = 100*ape1/n
ape2 = 0
for idx, actual in enumerate(ls2):
    ape2 += abs((actual-ewma2[idx+1])/actual)
    mape2 = 100*ape2/n
ape3 = 0
for idx, actual in enumerate(ls2):
    ape3 += abs((actual-ewma3[idx+1])/actual)
    mape3 = 100*ape3/n


print("MAPE for a = 0.3 = {:.2f}%".format(mape1))
print("MAPE for a = 0.4 = {:.2f}%".format(mape2))
print("MAPE for a = 0.5 = {:.2f}%".format(mape3))
print("MAPE for a = 0.5 is lowest. Hence our prediction is
↪  {:.1f}".format(ewma3.iloc[-1]))

#%% q8

#8a
#calc var of a sample assuming unif dist of each item in sample
def calc_var(sample):
    sumofsquares = 0.0
    for n in sample:
        sumofsquares += n**2
    expectation_squares = sumofsquares/len(sample)
#    print(expectation_squares)
    expectation_squared = (sum(sample)/float(len(sample)))**2
```

```
#    print(expectation_squared)
    return expectation_squares - expectation_squared

def sample_var(sample):
    return sum([((i-2.5)**2) for i in sample])/2

print("The variance is: {:.2f}".format(calc_var([1,2,3,4])))

#8b
results = {}
for pair in itertools.combinations_with_replacement([1,2,3,4],
↪  2):
    results[pair] = sample_var(pair)
#print(results)
print("The PMF (sorta) is: {}".format(results))

print("The expected value of the sampling variance is:
↪  {}".format(sum(results.values())/len(results)))

#%% q9
#9c

u = (norm.cdf(45, loc = 40, scale = (15**2/50.0)**0.5) -
↪  norm.cdf(35, loc = 40, scale = (15**2/50.0)**0.5))
v = (norm.cdf(45, loc = 40, scale = (15**2/100.0)**0.5) -
↪  norm.cdf(35, loc = 40, scale = (15**2/100.0)**0.5))
print("P(35<U<45) = {:.3}".format(u))
print("P(35<V<45) = {:.3}".format(v))
```