Tutorial 8

ST2137-2420

Material

This tutorial covers the topics and concepts from chapter 7, which covers hypothesis tests for comparing means between two groups.

As always, work out each question with R, Python and SAS unless otherwise stated.

Question 1

The number of pages in magazines devoted to advertisements varies widely from magazine-to-magazine and from issue-to-issue within the same magazine. Advertising expenditures, and therefore the number of advertising pages in magazines, tend to be the highest during periods of economic growth. The data in the file `weeklies.txt` gives the number of advertising pages in the current issues of 19 weekly magazines and the number of advertising pages in the same issue of the previous calendar year.

1. Conduct an appropriate t-test at 10% level to assess if the mean difference in advertising expenditure is significantly different from 0. Print out the 90% CI for the mean difference.

R code

```
#R
weeklies <- read.csv("data/weeklies.txt")
weeklies_t_test <- t.test(weeklies$current,
                          weeklies$lastyear,
                          paired = TRUE,
                          conf.level = 0.90 )

cat("The 90% CI is (", format(weeklies_t_test$conf.int, digits=3), ").", sep="")
```

```
The 90% CI is (-19.432 -0.647).
```

Python code

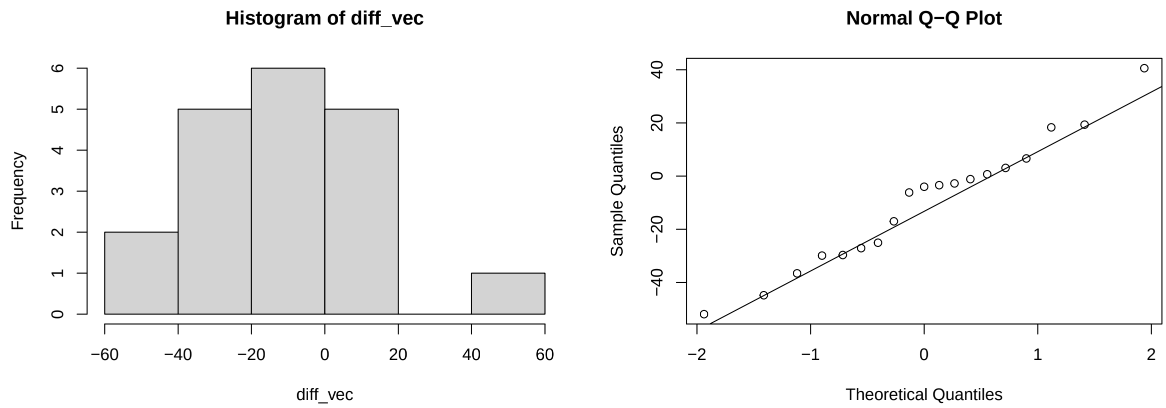
```
import pandas as pd
import numpy as np
from scipy import stats
import statsmodels.api as sm

import matplotlib.pyplot as plt

weeklies = pd.read_csv("data/weeklies.txt")

weeklies_t_test = stats.ttest_rel(weeklies.current, weeklies.lastyear, )
weeklies_ci = weeklies_t_test.confidence_interval(0.90)

#print(weeklies_t_test)
```

```
print(f"The 90% CI is ({weeklies_ci.low:.3f}, {weeklies_ci.high:.3f}).")
## The 90% CI is (-19.432, -0.647).
```

2. Compute the difference (as `diff_vec` = `current - lastyear`) and assess if this variable is Normally distributed using:
   (i) the histogram
   (ii) qq-plot
   (iii) using a hypothesis test of your choice.

R code

```
#R
diff_vec <- weeklies$current - weeklies$lastyear
hist(diff_vec)
qqnorm(diff_vec)
qqline(diff_vec)
ks.test(diff_vec, "pnorm")
```

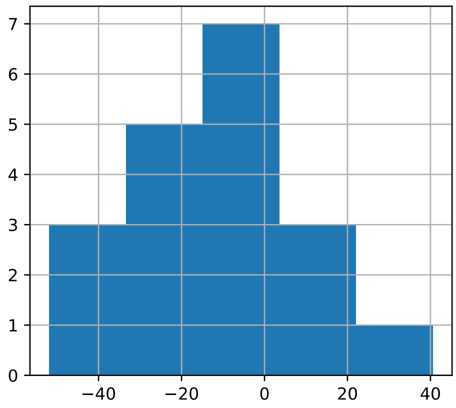


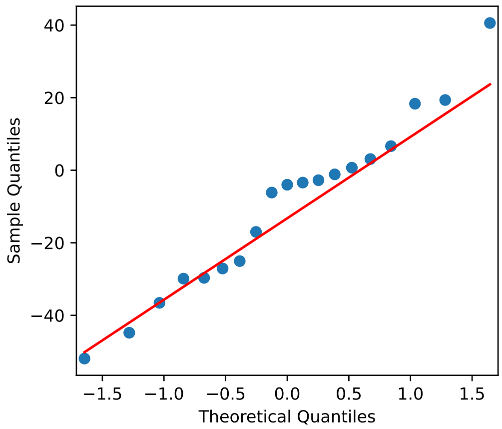

```
    Exact one-sample Kolmogorov-Smirnov test

data:  diff_vec
D = 0.6286, p-value = 8.555e-08
alternative hypothesis: two-sided
```

Python code

```
#Python

diff_vec = weeklies.current - weeklies.lastyear
plt.figure(figsize=(10,4))
plt.subplot(1, 2, 1)
diff_vec.hist(bins=5);
ax2 = plt.subplot(1, 2, 2)
sm.qqplot(diff_vec, line="q", ax=ax2);

ks_out = stats.ks_1samp(diff_vec, stats.norm.cdf)
print(f"The p-value is {ks_out.pvalue:.5f}")
```

```
The p-value is 0.00000
```





3. Write R and Python functions to compute estimates of skewness and kurtosis, as defined in the lecture notes. Apply the functions to `diff`. Ensure that your values agree with these:

```
library(DescTools)
Skew(diff_vec, method=1)
## [1] 0.1283815
# [1] 0.1283815
Kurt(diff_vec, method=1)
## [1] -0.4401952
```

R code

```
skewness_fn <- function(x) {
  x_mean <- mean(x)
  n <- length(x)

  numerator <- mean((x - x_mean)^3)
  denominator <- ((1-1/n)*var(x))^(3/2)
  numerator/denominator
}

kurtosis_fn <- function(x) {
  x_mean <- mean(x)
  n <- length(x)

  numerator <- mean((x - x_mean)^4)
  denominator <- ((1-1/n)*var(x))^2
  numerator/denominator - 3
}
```

Python code

```
def skewness_fn(x):
    x_mean = np.mean(x)
    n = len(x)
    numerator = ((x - x_mean)**3).mean()
    denominator = np.var(x)**(3/2)
    return numerator / denominator
def kurtosis_fn(x):
    x_mean = np.mean(x)
    n = len(x)
```

```
    numerator = ((x - x_mean)**4).mean()
    denominator = np.var(x)**2
    return numerator / denominator - 3
```

4. The paper Kim and White (2004) contains a robust estimate of skewness h_1:

$$h_1 = \frac{Q_3 + Q_1 - 2Q_2}{Q_3 - Q_1}$$

Implement this in R and Python, and apply it to the `diff_vec` data. Is there a large change from part (3)? Can you explain why or why not?

R code

```
## Q3 + Q1 - 2Q2 / (Q3 - Q1)
##
skewness2 <- function(x) {
  qq_tmp <- quantile(x, c(0.25, 0.50, 0.75))
  unname((qq_tmp[3] + qq_tmp[1] - 2*qq_tmp[2])/(qq_tmp[3] - qq_tmp[1]))
}
# skewness2(diff)
# [1] -0.6108358
```

Python code

```
def skewness2(x):
    qq_tmp = np.quantile(x, [0.25, 0.5, 0.75])
    return (qq_tmp[2] + qq_tmp[0] - 2*qq_tmp[1])/(qq_tmp[2] - qq_tmp[0])
```

The non-robust measure of skewness is 0.12, but the robust measure is -0.61. The reason is because of the maximum diff value: 40.58. It corresponds to `TV Guide` magazine, which increased it's expenditure from 82.02 to 41.44. Most of the remaining weeklies decreased their expenditure.

Question 2

The purchasing director for an industrial parts factory is investigating the possibility of purchasing a new type of milling machine. He determines that the new machine will be bought if there is evidence that the parts produced have a higher average breaking strength than those from the old machine. The data file `machine.txt` represents the breaking strength of samples of 50 parts from the old and the new machines.

5. Is there evidence that the purchasing director should buy the new machine? Ensure that you check all assumptions as required.

R code

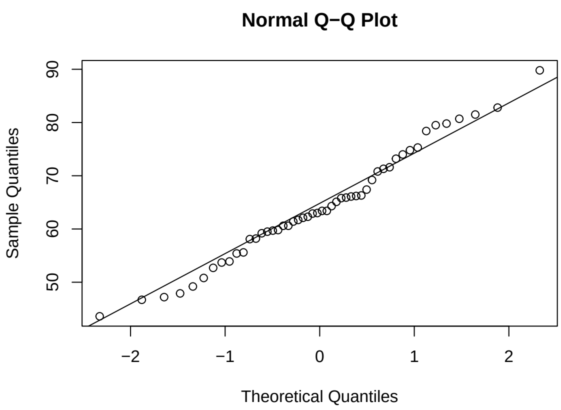
```
library(lattice)
machine <- read.table("data/machine.txt", header=TRUE)
aggregate(strength ~ machine, data=machine,
          function(x) {c(m = mean(x), s = sd(x))})
```

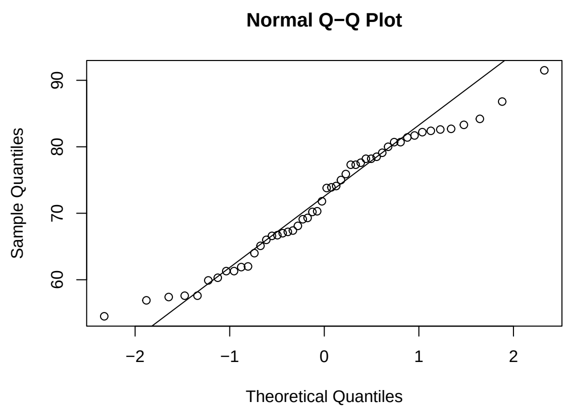
```
  machine strength.m strength.s
1       N  71.972000   9.250241
2       O  64.248000  10.369603
```

```
t.test(strength ~ machine, data=machine, var.equal=TRUE)
```

```
	Two Sample t-test

data:  strength by machine
t = 3.9304, df = 98, p-value = 0.0001581
alternative hypothesis: true difference in means between group N and group O is not equal to 0
95 percent confidence interval:
  3.824174 11.623826
sample estimates:
mean in group N mean in group O 
         71.972          64.248 
```

```
qqnorm(machine$strength[machine$machine == "O"])
qqline(machine$strength[machine$machine == "O"])
qqnorm(machine$strength[machine$machine == "N"])
qqline(machine$strength[machine$machine == "N"])
```





Python code

```
machines = pd.read_csv("data/machine.txt", sep='\\s+')

stats.ttest_ind(machines.strength[machines.machine == "O"],
                machines.strength[machines.machine == "N"])
```

```
TtestResult(statistic=np.float64(-3.9304389129902706), pvalue=np.float64(0.00015810425164191597), df=np
```

Please try out the qqplots in python on your own..

It does appear that the Normality assumption does not hold well for the New machines. We may want to try the non-parametric test. However, given how different the means are, and the distance of the CI from 0, we can be almost sure that we would reject the null hypothesis there too.

Question 3

A flexible working hour program permits employees to design their own 42-hour work week to meet their personal needs. The management of a large manufacturing firm is considering adopting a flextime program for its administrators and professional employees, depending on the success or failure of a pilot program. Ten employees were randomly selected and given a questionnaire designed to measure their attitudes toward their jobs. Each was then permitted to design and follow a flextime workday. After six months, attitudes toward their jobs were again measured. The resulting attitude scores are given in the data file `flextime.txt`. The higher the score, the more favorable the employee's attitude toward his or her work.

6. Use a nonparametric test procedure in SAS to evaluate the success of the pilot flextime program at 5% significance level.
7. Repeat the question above using R and Python.
8. Convert the dataset to long form (use `stack` in R, `pd.DataFrame.stack` in Python). This means that the dataframe should now look like this:

```
   values    ind
1      54 before
2      25 before
3      82 before
4      76 before
5      63 before
6      82 before
7      94 before
8      72 before
9      33 before
10     90 before
11     68  after
12     42  after
13     80  after
14     91  after
15     70  after
16     88  after
17     90  after
18     81  after
19     38  after
20     93  after
```

Solution

This is a paired sample. Hence the appropriate nonparametric test to apply is the Wilcoxon Signed Rank test. The null and alternative hypotheses are:

$$
\begin{aligned}
H_0 &: \text{median of } D_i \text{ is } 0. \\
H_1 &: \text{median of } D_i \text{ is greater than } 0.
\end{aligned}
$$

The output from SAS is as follows:

Variable: _Difference_ (Difference: before - after)

Tests for Location: Mu0=0				
Test	**Statistic**		**p Value**	
Student's t	**t**	-3.16587	**Pr > |t|**	0.0114
Sign	**M**	-3	**Pr >= |M|**	0.1094
Signed Rank	**S**	-23.5	**Pr >= |S|**	0.0137

With SAS, it's not possible to conduct a one-sided non-parametric test. The p-value of the two-sided signed rank test is $0.0137 < 0.05$.

Since we only have 10 observations, the number of non-zero differences would be less than 16; we should be using the exact version of the test.

Now we repeat the test in R and Python.

R code

```r
flextime <- read.table("data/flextime.txt", header=TRUE)
D <- flextime$after - flextime$before
wilcox.test(D,alternative = "greater")
```

```
	Wilcoxon signed rank exact test

data:  D
V = 51, p-value = 0.006836
alternative hypothesis: true location is greater than 0
```

Python code

```python
import pandas as pd
import numpy as np
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols

flextime = pd.read_table("../data/flextime.txt", delimiter = "\\s+")
D = flextime.after - flextime.before
wsr_output = stats.wilcoxon(flextime.after, flextime.before,
                            alternative = "greater", method="exact")
print(f"The p-value is {wsr_output.pvalue:.4f}.")
```

```
The p-value is 0.0068.
```

Before proceeding, we try out a function that is useful in converting the "shape" of a dataset. If you are taking/have taken DSA2101, you would learn more general methods to do this, but here is a function in base R that is already useful.

The `stack` function in R (`pd.DataFrame.stack()` in Python) allows us to stack columns that are originally side-by-side into a long column vector.

R code

```r
flex2 <- stack(flextime, select=-employee)
```

Python code

```python
flex2 = flextime.iloc[:,1:].stack().reset_index()
```

References

Kim, Tae-Hwan, and Halbert White. 2004. "On More Robust Estimation of Skewness and Kurtosis." *Finance Research Letters* 1 (1): 56–73.