Solutions 6

Jumping Rivers

Method A					
78.64	79.01	79.57	79.52	80.71	79.95
78.50	79.10	81.98	80.09	80.29	80.22

Method B					
81.92	81.12	82.47	82.86	82.89	82.45
82.51	81.11	83.07	82.77	82.38	83.14

We conducted an experiment and collected the data in the tables above. This data set isn't paired.[1]

[1] I intentionally didn't make the data available for download so you would have to think about how to enter the data. You could enter it either Excel and import or directly into Python.

1. Input the data into Python Combine the two data sets into a single data frame.

```
##Data for question 1
## Easier using Excel and export as CSV

import pandas as pd

x = [
  78.64, 79.01, 79.57, 79.52, 80.71, 79.95,
  78.50, 79.10, 81.98, 80.09, 80.29, 80.22
]

y = [
  81.92, 81.12, 82.47, 82.86, 82.89, 82.45,
  82.51, 81.11, 83.07, 82.77, 82.38, 83.14
]

d = pd.DataFrame({
  'method': ['a']*12 + ['b']*12,
  'value': x + y
})
```

2. Exploratory data analysis.

- Construct boxplots, histograms and q-q plots for both data sets. Work out the means and standard deviations. Before carrying out any statistical test, what do you think your conclusions will be? Do you think the variances are roughly equal? Do you think the data conforms to a normal distribution.

```
# Could either do histograms separately or
d.hist(by=d['method'])
plt.show()

# boxplots
d.boxplot(by='method')
plt.show()

import statsmodels.api as sm
sm.qqplot(d.query('method == "a"')['value'], fit=True, line='45')
sm.qqplot(d.query('method == "b"')['value'], fit=True, line='45')

d.groupby('method').agg(['mean','std'])

# normal assumption does not appear too bad, variances don't look the same,
# boxplots suggest a significant difference
```

- Carry out a two sample t-test. Assume that the variances are unequal.

```
import scipy.stats

scipy.stats.ttest_ind(
  d.query('method == "a"')['value'],
  d.query('method == "b"')['value'],
  equal_var=False
)
```

```
## Ttest_indResult(statistic=-7.560307657924682, pvalue=3.0001468265388804e-07)
```

- How does this answer compare with your intuition?

```
answer = """
We found a significant result when we were expecting one
"""
```

- Carry out a two sample t-test, assuming equal variances.

```
scipy.stats.ttest_ind(
  d.query('method == "a"')['value'],
  d.query('method == "b"')['value'],
  equal_var=True
)

# Smaller p-value
```

```
## Ttest_indResult(statistic=-7.5603076579246835, pvalue=1.4892522022968455e-07)
```

- Now carry out a wilcox test, how does the result compare

```
scipy.stats.wilcoxon(
  d.query('method == "a"')['value'],
  d.query('method == "b"')['value']
)
```

```
# much larger p-value, although we still reject the null hypothesis
```

```
## WilcoxonResult(statistic=0.0, pvalue=0.002217721464237049)
```

3. Suppose we are interested whether successful business executives are affected by their zodiac sign. We have collected 4265 samples and obtained the following data

Aries	Taurus	Gemini	Cancer	Leo	Virgo	Libra	Scorpio	Sagittarius	Capricorn	Aquarius	Pisces
348	353	359	357	350	355	359	367	345	362	343	367

Table 1: Zodiac signs of 4265 business executives

- Carry out a χ^2 goodness of fit test on the zodiac data. Are business executives distributed uniformly across zodiac signs? In the notes we used `scipy.stats.chi2_contingency` as we had an example with 2 independent variables, giving rise to a contingency table. Here we only have a single factor so we should use `scipy.stats.chisquare`

```
x = [348, 353, 359, 357, 350, 355, 359, 367, 345, 362, 343, 367]
scipy.stats.chisquare(x)
## Since p > 0.05 we can't reject the null hypothesis.
## However, the question is worded as though we can "prove" the Null
## hypotheis, which we obviously can't do.
```

```
## Power_divergenceResult(statistic=1.949589683470105, pvalue=0.9986653614512341)
```

- What are the expected values for each zodiac sign?

```
import numpy as np
```

```
np.mean(x)
```

```
## 355.4166666666667
```

- The formula for calculating the residuals is given by

$$\frac{\text{observed} - \text{expected}}{\sqrt{\text{expected}}}$$

Which residuals are large?

```
(x - np.mean(x))/np.sqrt(np.mean(x))
```

```
## array([-0.39340499, -0.12818814,  0.19007207,  0.08398534, -0.28731825,
##         -0.0221014 ,  0.19007207,  0.61441903, -0.5525351 ,  0.34920218,
##         -0.65862184,  0.61441903])
```

4. The University of Texas Southwestern Medical Center examined whether the risk of contracting Hepatitis C was related to tattoo use.[2] The data from the study is summarised as follows:

[2] Haley, R. and Fischer, P.R. 2001

	Hepatitis C	No Hepatitis C	Total
Tattoo, Parlour	17	35	52
Tattoo, elsewhere	8	53	61
No tattoo	22	491	513
Total	47	579	626

Table 2: Counts of patients by their Hepatitis C status and whether they had a tattoo from a parlour, from elsewhere or had no tattoo at all.

- Carry out a χ^2 test to determine if the Hepatitis is related to tattoo status.

```
h = [17, 8, 22]
nh = [35, 53, 491]
d = pd.DataFrame({
  'hepatitis': h,
  'no hepatitis': nh
}, index = ['tattoo, parlour', 'tattoo, elsewhere', 'no tattoo'])

chi2, p, dof, ex = scipy.stats.chi2_contingency(d)
p

# suggests we reject the null hypothesis, they are not independent
```

```
## 2.657854691515767e-13
```

- When carrying out χ^2 tests, we should make sure that individual cells have expected values of at least five, otherwise the distributional assumptions may be invalid. What are the expected values of each cell. Which cells have an expected value less than five?

```
ex < 5
```

```
## array([[ True, False],
##        [ True, False],
##        [False, False]])
```

- Since some of the cells have expected values slightly less than five, we should ensure that these aren't driving the test statistic. Look at the test residuals. Which residuals are large? What should you do now?

```
d - ex
## Some of the expected values are less then 5
## So consider combining cells.
```

```
##                    hepatitis  no hepatitis
## tattoo, parlour     13.095847    -13.095847
## tattoo, elsewhere    3.420128     -3.420128
## no tattoo          -16.515974     16.515974
```