Notebook 4, Module 2, Statistical Inference for Data Science, CAS Applied Data Science, 2019-08-30, G. Conti, S. Haug, University of Bern.

Project work Module 2

This is my project work for Module 2. Plots, numbers and tables for the poster is produced with this Notebook.

The data used is the Iris data set. Available here: https://archive.ics.uci.edu/ml/datasets/iris (https://archive.ics.uci.edu/ml/datasets/iris)

Outline of the data analysis 3x45 min (depending on your background)

- Introduction (situation and goal)
- Descriptive part
- Data filtering with efficiency from fitted models
- Hypothesis testing
- Conclusions

1. Introduction

I am in court as a data science expert. My task is to give advice to the judges. I will have 15 minutes time and plan to bring a poster for the presentation. This way court participants can also look at it in the breaks.

Situation (fictional):

Some new company v4Setosa recently sequenced the genes of the Iris species Setosa and patented it, apparently in order to preserve this species because it is so beautiful. Due this patent it is not allowed to change the plant.

A big farmer and hater of Iris and with a field where Iris is a disturbing weed, has been using a new product from Sonte Manto for a couple of years. The product is supposed to effectively kill Iris plants.

A big Iris lover collected a sample of Iris plants from the farmer's field and thinks the Iris Setosa setal leaves are bigger than normal. She sent the sample to v4setosa, which in turn came to the conclusion that Setosa must have mutated due to the product from Sonte Manto.

So w4setosa sued Sonte Manto with the claim that they have changed the plant with their product. Sonte Manto may risk to pay a billion dollars. The court has asked me to give a neutral and scientific advice.

Data Analysis

- Data collection
  - Request a larger Iris sample from a field far away from the farmer -dataset 0. The data is labeled as the local collector had separate greenhouses for each species.
  - Request a Iris sample from the farmer (collected by some neutral person) - dataset 1. The data is a random mix of all species
- Description of the data
  - Calculate means, variances and correlations
  - Test normality
  - Choose good plots
- Find by **regression** a good way to filter out setosa in the farmer sample (increases the the power of the final test)
- Add a **test** to the filtered sample, checking if the farmer setosa leaves differ significantly from the reference sample
- Conclude
- (Additionally **exclude** all mutation hypotheses not compatible with this experiment)

Definition of significant

We consider p-values below 0.001 as unlikely enough to reject the fluctuation hypothesis. We will for such values advice the court to consider it as a mutation. (This is a weak point that the laywers of Sonte Manto will of course attack.)

2. Description of the datasets

We artificially create the datasets. Out of the first 30 entries from each Iris species make the reference sets (annotated/labeled). With the rest, 20 rows per species, we create a mixed sample being the farmer sample (not anotated).

Import modules we may need

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        from scipy import stats
```

Create the datasets

```
In [2]: df_org     = pd.read_csv('iris.csv',names=['slength','swidth','plength','pwi
        dth','species']) # data type is a string (str), i.e. not converted into numb
        ers
        df_ref_set = df_org[df_org['species']=='Iris-setosa'][:30]
        df_ref_vir = df_org[df_org['species']=='Iris-virginica'][:30]
        df_ref_ver = df_org[df_org['species']=='Iris-versicolor'][:30]
        df_farm_1  = df_org[df_org['species']=='Iris-setosa'][30:]
        df_farm_2  = df_org[df_org['species']=='Iris-virginica'][30:]
        df_farm_3  = df_org[df_org['species']=='Iris-versicolor'][30:]
        df_farm    = df_org[df_org['species']=='Iris-virginica'][30:]
        df_farm    = df_farm_1.append(df_farm_2.append(df_farm_3))
```

The labels in the farmer dataset are unknown, so we remove them to 'simulate' the real situation.

```
In [3]: df_farm['species']='Unknown'
        df_farm.head()
```

Out[3]:

	slength	swidth	plength	pwidth	species
30	4.8	3.1	1.6	0.2	Unknown
31	5.4	3.4	1.5	0.4	Unknown
32	5.2	4.1	1.5	0.1	Unknown
33	5.5	4.2	1.4	0.2	Unknown
34	4.9	3.1	1.5	0.1	Unknown

Describe setosa

In [4]:
```
df_ref_set.describe()
```

Out[4]:

	slength	swidth	plength	pwidth
count	30.000000	30.000000	30.000000	30.000000
mean	5.026667	3.450000	1.473333	0.246667
std	0.372256	0.350123	0.185571	0.100801
min	4.300000	2.900000	1.000000	0.100000
25%	4.800000	3.200000	1.400000	0.200000
50%	5.000000	3.400000	1.500000	0.200000
75%	5.200000	3.675000	1.600000	0.300000
max	5.800000	4.400000	1.900000	0.500000

In [5]:
```
df_ref_vir.describe()
```

Out[5]:

	slength	swidth	plength	pwidth
count	30.000000	30.000000	30.000000	30.000000
mean	6.583333	2.933333	5.603333	2.006667
std	0.688869	0.334595	0.626695	0.255874
min	4.900000	2.200000	4.500000	1.500000
25%	6.225000	2.725000	5.100000	1.800000
50%	6.500000	2.950000	5.600000	2.000000
75%	7.175000	3.150000	5.975000	2.175000
max	7.700000	3.800000	6.900000	2.500000

In [6]:
```
df_ref_ver.describe()
```

Out[6]:

	slength	swidth	plength	pwidth
count	30.000000	30.000000	30.000000	30.000000
mean	6.070000	2.790000	4.333333	1.353333
std	0.545925	0.328371	0.459635	0.211291
min	4.900000	2.000000	3.300000	1.000000
25%	5.700000	2.625000	4.000000	1.300000
50%	6.100000	2.850000	4.500000	1.400000
75%	6.475000	3.000000	4.700000	1.500000
max	7.000000	3.300000	5.000000	1.800000

I'will make a table suited for the poster:

In [7]:
```
#Python code for that table here.
```

Check normality

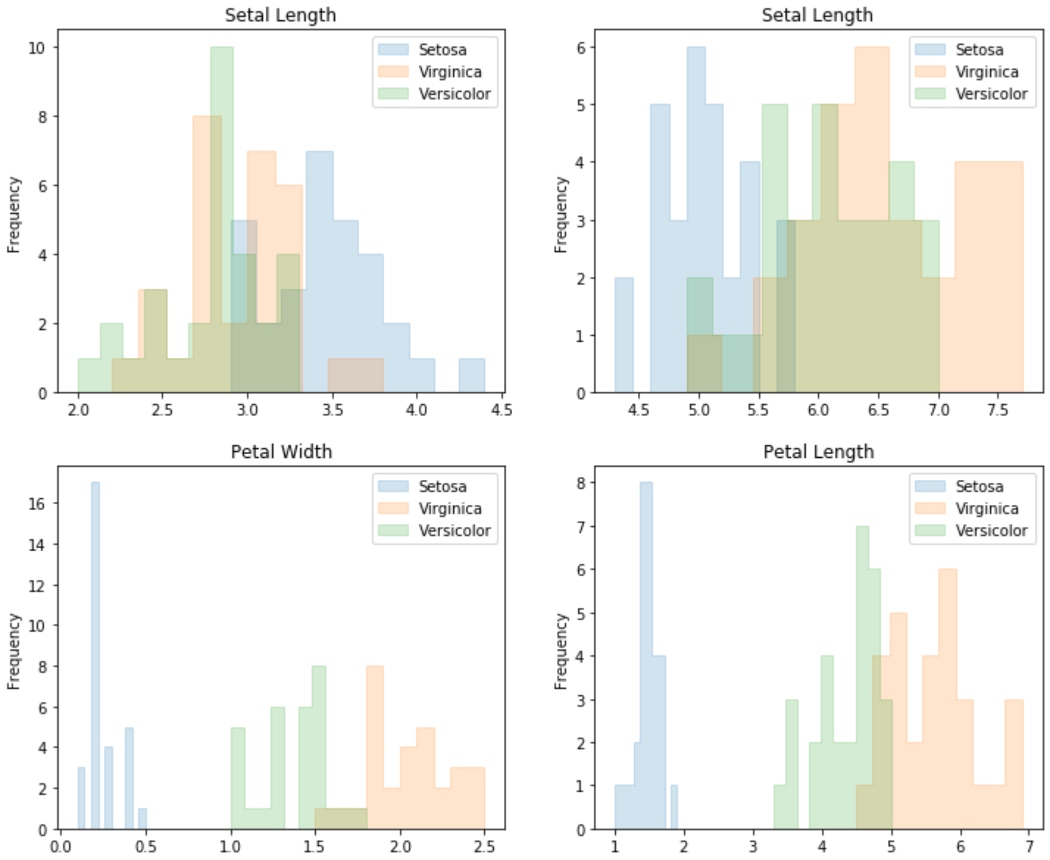
```
In [8]: print('p-values from the normality tests on the setal width (D Agostino-Pear
        son):')
        datasets       = [df_ref_set['swidth'],df_ref_vir['swidth'],df_ref_ver['swi
        dth'],df_farm['swidth'] ]
        dataset_labels = ['Setosa','Virginica','Versicolor','Farmer']
        i = 0
        for dataset in datasets:
          k2, p = stats.normaltest(dataset) # D Agostino-Pearson
          print('%10s %1.2f ' % (dataset_labels[i],p))
          i+=1
```

```
p-values from the normality tests on the setal width (D Agostino-Pearson):
    Setosa 0.30
 Virginica 0.34
Versicolor 0.23
    Farmer 0.30
```

We conclude that all datasets are normal in the setal width.

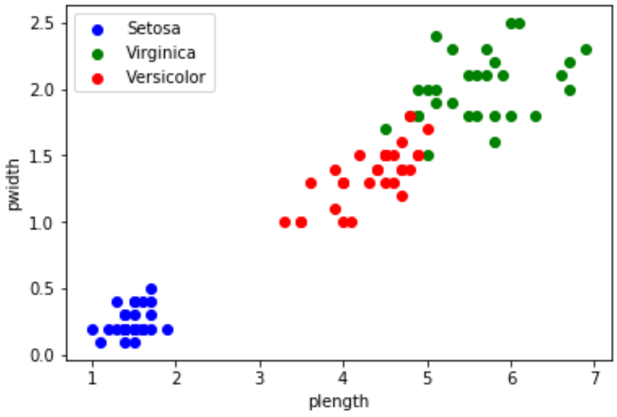
Plot the histograms (looking for fitering posibilities)

In [9]:
```
plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
plt.title('Setal Length')
i=0
for dataset in datasets[0:3]:
    dataset.plot(kind="hist",fill=True,alpha=0.2,histtype='step',label=datas
et_labels[i])
    #print('%10s %1.2f ' % (datasets_labels[i],p))
    i+=1
plt.legend()
plt.subplot(2,2,2)
plt.title('Setal Length')
datasets2 = [df_ref_set['slength'],df_ref_vir['slength'],df_ref_ver['slength
'],df_farm['slength'] ]
i=0
for dataset in datasets2[0:3]:
    dataset.plot(kind="hist",fill=True,alpha=0.2,histtype='step',label=datas
et_labels[i])
    #print('%10s %1.2f ' % (datasets_labels[i],p))
    i+=1
plt.legend()
#plt.show()
plt.subplot(2,2,3)
plt.title('Petal Width')
datasets3 = [df_ref_set['pwidth'],df_ref_vir['pwidth'],df_ref_ver['pwidth'],
df_farm['pwidth'] ]
i=0
for dataset in datasets3[0:3]:
    dataset.plot(kind="hist",fill=True,alpha=0.2,histtype='step',label=datas
et_labels[i])
    #print('%10s %1.2f ' % (datasets_labels[i],p))
    i+=1
plt.legend()
#plt.show()
plt.subplot(2,2,4)
plt.title('Petal Length')
datasets4 = [df_ref_set['plength'],df_ref_vir['plength'],df_ref_ver['plength
'],df_farm['plength'] ]
i=0
for dataset in datasets4[0:3]:
    dataset.plot(kind="hist",fill=True,alpha=0.2,histtype='step',label=datas
et_labels[i])
    #print('%10s %1.2f ' % (datasets_labels[i],p))
    i+=1
plt.legend()
plt.show()
```



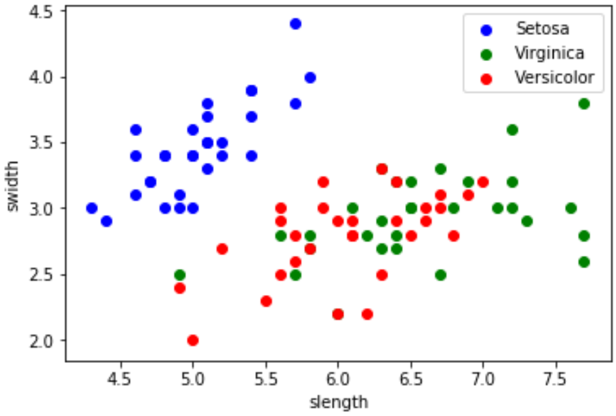
It is clear that we filter out setosa more or less completly with p_width<0.7 cm and plength<2.5 cm. Let's look at the populations in this this space as a scatter plot.

```
In [10]: plt.subplot(1,1,1)
plt.scatter(df_ref_set['plength'],df_ref_set['pwidth'],c='b',label='Setosa')
plt.scatter(df_ref_vir['plength'],df_ref_vir['pwidth'],c='g',label='Virginica')
plt.scatter(df_ref_ver['plength'],df_ref_ver['pwidth'],c='r',label='Versicolor')
plt.xlabel('plength')
plt.ylabel('pwidth')
plt.legend()
plt.show()
```



A discrimination should work very well. However, we don't see the full distribution and this is just by eye. I will take a look at the sepal version of the plot, too.

In [11]:
```
plt.subplot(1,1,1)
plt.scatter(df_ref_set['slength'],df_ref_set['swidth'],c='b',label='Setosa')
plt.scatter(df_ref_vir['slength'],df_ref_vir['swidth'],c='g',label='Virginic
a')
plt.scatter(df_ref_ver['slength'],df_ref_ver['swidth'],c='r',label='Versicol
or')
plt.xlabel('slength')
plt.ylabel('swidth')
plt.legend()
plt.show()
```



Also here a good separation seems easy by using a line. Fitting a line which best separates between two samples belongs to Module 3. Here we will choose by eye two cuts in the petal plane and estimate the efficiency of the filter by using generated values from fitted models.

3. Filtering out Setosa from the farmer sample

Let us fit normal distributions to the reference samples. Then generate 100 000 rows with these distributions, plot the scatter plots with this simulated data, define the fitering cuts and calculate the efficiency of our filter.

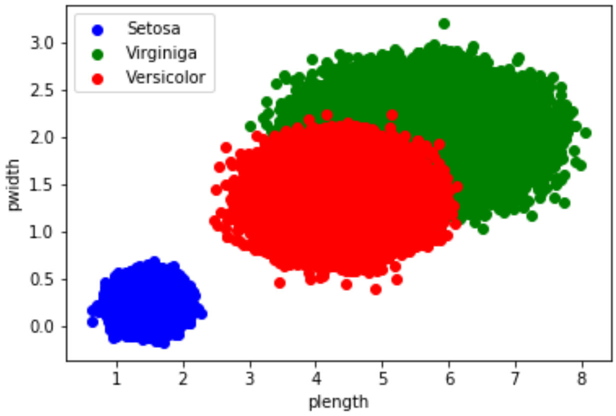
In [12]:

```
import scipy
mean, sd = scipy.stats.norm.fit(df_ref_set['plength'])
df_gen_pl_set = scipy.stats.norm.rvs(mean,sd,100000)
mean, sd = scipy.stats.norm.fit(df_ref_set['pwidth'])
df_gen_pw_set = scipy.stats.norm.rvs(mean,sd,100000)

mean, sd = scipy.stats.norm.fit(df_ref_vir['plength'])
df_gen_pl_vir = scipy.stats.norm.rvs(mean,sd,100000)
mean, sd = scipy.stats.norm.fit(df_ref_vir['pwidth'])
df_gen_pw_vir = scipy.stats.norm.rvs(mean,sd,100000)

mean, sd = scipy.stats.norm.fit(df_ref_ver['plength'])
df_gen_pl_ver = scipy.stats.norm.rvs(mean,sd,100000)
mean, sd = scipy.stats.norm.fit(df_ref_ver['pwidth'])
df_gen_pw_ver = scipy.stats.norm.rvs(mean,sd,100000)

plt.subplot(1,1,1)
plt.scatter(df_gen_pl_set,df_gen_pw_set,c='b',label='Setosa')
plt.scatter(df_gen_pl_vir,df_gen_pw_vir,c='g',label='Virginiga')
plt.scatter(df_gen_pl_ver,df_gen_pw_ver,c='r',label='Versicolor')
plt.xlabel('plength')
plt.ylabel('pwidth')
plt.legend()
plt.show()
```



Filter choice: plength<2.5 and pwidth<1.0

Filter efficiency:

- Setosa 1.0 (100%)
- Virginica 0.0
- Versicolor 0.0

3. Filter the farmer data

In [13]:

```
df_farm     = df_farm.sort_values('slength')
df_farm_set = df_farm[df_farm['plength'] < 2.5]
df_farm_set
```

Out[13]:

	slength	swidth	plength	pwidth	species
42	4.4	3.2	1.3	0.2	Unknown
38	4.4	3.0	1.3	0.2	Unknown
41	4.5	2.3	1.3	0.3	Unknown
47	4.6	3.2	1.4	0.2	Unknown
30	4.8	3.1	1.6	0.2	Unknown
45	4.8	3.0	1.4	0.3	Unknown
37	4.9	3.1	1.5	0.1	Unknown
34	4.9	3.1	1.5	0.1	Unknown
49	5.0	3.3	1.4	0.2	Unknown
40	5.0	3.5	1.3	0.3	Unknown
35	5.0	3.2	1.2	0.2	Unknown
43	5.0	3.5	1.6	0.6	Unknown
46	5.1	3.8	1.6	0.2	Unknown
44	5.1	3.8	1.9	0.4	Unknown
39	5.1	3.4	1.5	0.2	Unknown
32	5.2	4.1	1.5	0.1	Unknown
48	5.3	3.7	1.5	0.2	Unknown
31	5.4	3.4	1.5	0.4	Unknown
36	5.5	3.5	1.3	0.2	Unknown
33	5.5	4.2	1.4	0.2	Unknown

4. Make the hypothesis test

What is the probability that the farmer's Setosa sample is a just fluctuation of the fitted model?

First on the non-filtered farmer data:

In [14]:

```
stats.ttest_ind(df_farm['swidth'],df_ref_set['swidth'])
```

Out[14]:

```
Ttest_indResult(statistic=-4.439847029716977, pvalue=2.6025613930449344e-05)
```

These are not from the same population. What about the filtered data:

In [15]:

```
stats.ttest_ind(df_farm_set['swidth'],df_ref_set['swidth'])
```

Out[15]:

```
Ttest_indResult(statistic=-0.7237814582186126, pvalue=0.4727127408773476)
```

Clearly from the same population. Some more tests:

```
In [16]: stats.ttest_ind(df_farm_set['slength'],df_ref_set['slength'])
Out[16]: Ttest_indResult(statistic=-0.503874959098223, pvalue=0.6166541850289631)
```

```
In [17]: stats.ttest_ind(df_farm_set['plength'],df_ref_set['plength'])
Out[17]: Ttest_indResult(statistic=-0.4620898274411638, pvalue=0.6461046018163226)
```

Conclusion

Our data analysis cannot confirm the claim of v4setosa. There is absolutely no evidence for the claim in our independent study. We advice the court to judge for not guilty.

(And we sent a very large bill which the looser v4setosa had to pay. v4setosa went bankrott and we cannot sleep anymore...)

Remarks regarding the poster session

I used four hours for this small analysis. To work it out, doing more tests, think through, study here and there, make nice tables and plots, create the poster, I would probably need another 10 hours.

Considering that you may need longer due to less experience, this is what is expected from you for the poster session (about 30h work).

```
In [ ]:
```