实验目录

实验内容

1. 基于单变量线性回归的体重预测模型

知识点

1) p-value 可以衡量模型、特征值的有效性

2) R-squared 可以衡量模型的解释能力

3) 线性回归模型需满足线性、正态性、方差齐性、独立性

实验目的

1) 学习使用 statsmodels 建立线性回归模型

2) 学习解读线性回归模型结果

3) 学习线性回归模型诊断

4) 使用 statsmodels 建立线性回归模型，根据身高预测体重

5) 对建立的线性回归模型进行回归诊断

实验步骤

1) 打开 Jupyter，并新建 python 工程

1. 桌面空白处右键，点击 Konsole 打开一个终端

2. 切换至/experiment/jupyter 目录

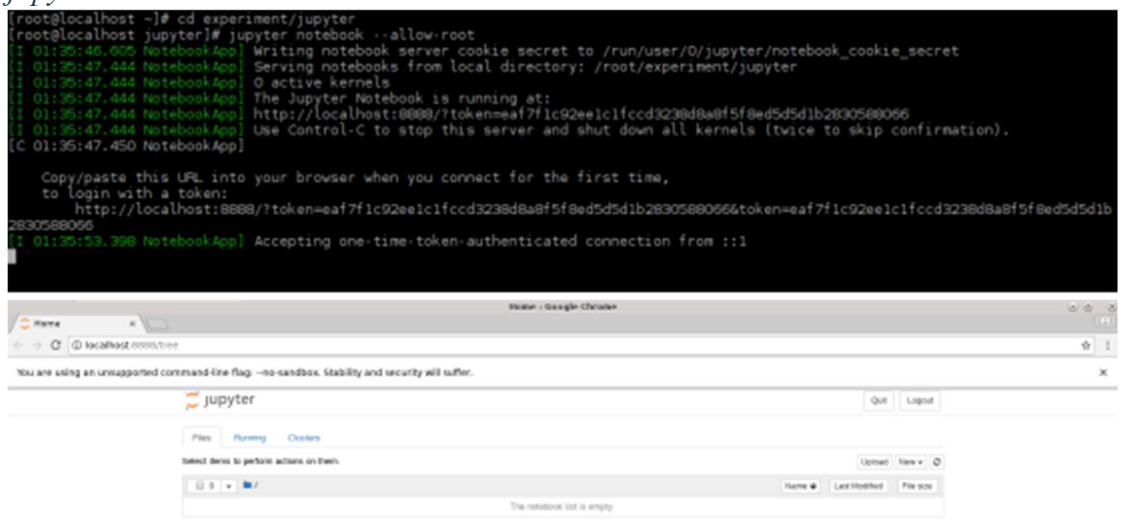
[Command 001]:

cd experiment/jupyter

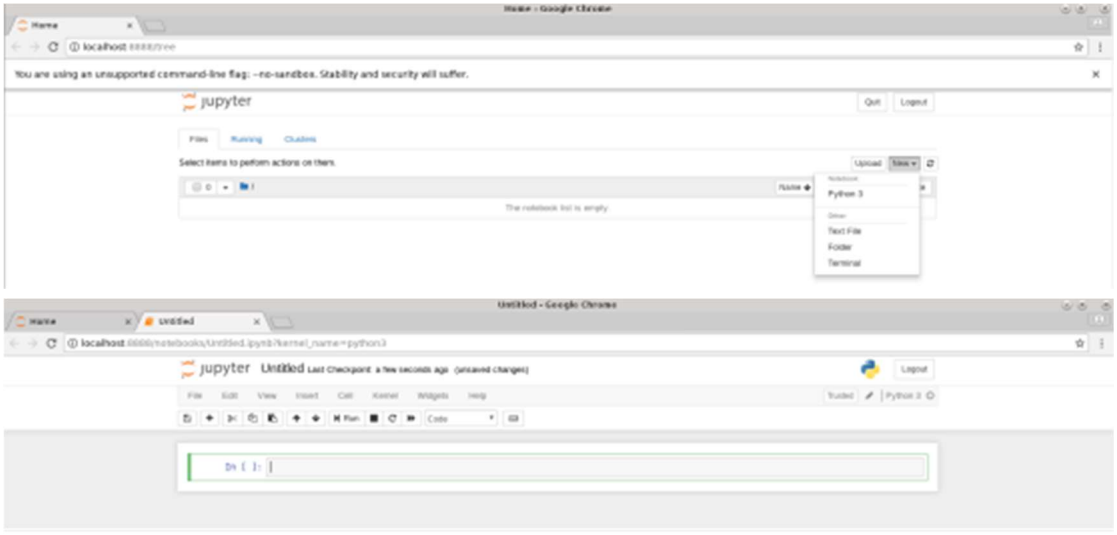
3. 启动 Jupyter，root 用户下运行需加'--allow-root'

[Command 002]:

jupyter notebook --allow-root



4. 依次点击右上角的 New，Python 3 新建 python 工程



5. 点击 Untitled，在弹出框中修改标题名，点击 Rename 确认

2) 读取数据

1. 输入代码后，使用 shift+enter 执行，下同。
2. women.csv 包括 15 名 30-39 岁的美国女性的身高和体重。其中:

   height：身高，连续数值(单位：in)

   weight：体重，连续数值(单位：lbs)

3. 使用 pandas 读取 csv 文件

[Code 001]:

```
import pandas as pd
df = pd.read_csv('/root/experiment/datas/women.csv')
# 查看 df 的维度
df.shape
```

```
import pandas as pd
```

```
df = pd.read_csv('/root/experiment/datas/women.csv')
df.shape
```

```
(15, 2)
```

3) 描述性分析与可视化分析

1. 查看 df 的前五项

[Code 002]:

```
df.head()
```

```
df.head()
```

	height	weight
0	58	115
1	59	117
2	60	120
3	61	123
4	62	126

2. 查看 df 中连续变量的统计描述

[Code 003]:

df.describe()

```
df.describe()
```

	height	weight
count	15.000000	15.000000
mean	65.000000	136.733333
std	4.472136	15.498694
min	58.000000	115.000000
25%	61.500000	124.500000
50%	65.000000	135.000000
75%	68.500000	148.000000
max	72.000000	164.000000

3. 查看 df 中各字段的缺失值

[Code 004]:

df.isnull().sum()

```
df.isnull().sum()
```

```
height    0
weight    0
dtype: int64
```

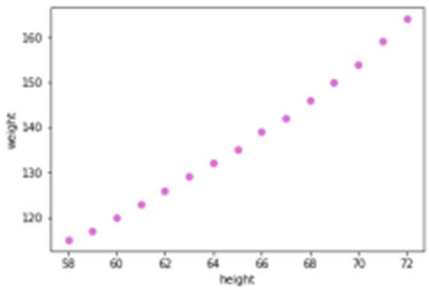
4. 使用 matplotlib 库进行可视化分析（绘图时，由于 jupyter 的问题，执行时可能需重复执行才能显示绘图结果，下同）

[Code 005]:

```
import matplotlib.pyplot as plt
plt.scatter(df['height'],df['weight'],color='orchid')
plt.xlabel('height')
plt.ylabel('weight')
plt.show()
```

```
import matplotlib.pyplot as plt
```

```
plt.scatter(df['height'],df['weight'],color='orchid')
plt.xlabel('height')
plt.ylabel('weight')
plt.show()
```



4) 数据处理

1. 为原始数据增加截距项

[Code 006]:

import statsmodels.api as sm
x = sm.add_constant(df['height'])
y = df['weight']
x.sample(6)

```
import statsmodels.api as sm
```

```
x = sm.add_constant(df['height'])
y = df['weight']
x.sample(6)
```

	const	height
11	1.0	69
2	1.0	60
1	1.0	59
5	1.0	63
4	1.0	62
0	1.0	58

5) 建立模型

1. 使用 OLS 方法建立线性回归模型，并查看模型结果

[Code 007]:

model = sm.OLS(y, x)
model_result = model.fit()
model_result.summary()

```
model = sm.OLS(y, x)
model_result = model.fit()
```

```
model_result.summary()
```

```
/usr/lib/python3.6/lib/python3.6/site-packages/scipy/stats/stats.py:1394: UserWarning: kurtosistest only valid for n>=20
... continuing anyway, n=15
  "anyway, n=%i" % int(n))
```

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.991
Model:	OLS	**Adj. R-squared:**	0.990
Method:	Least Squares	**F-statistic:**	1433.
Date:	Tue, 15 May 2018	**Prob (F-statistic):**	1.09e-14
Time:	02:27:27	**Log-Likelihood:**	-26.541
No. Observations:	15	**AIC:**	57.08
Df Residuals:	13	**BIC:**	58.50
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-87.5167	5.937	-14.741	0.000	-100.343	-74.691
height	3.4500	0.091	37.855	0.000	3.253	3.647

Omnibus:	2.396	**Durbin-Watson:**	0.315
Prob(Omnibus):	0.302	**Jarque-Bera (JB):**	1.660
Skew:	0.789	**Prob(JB):**	0.436
Kurtosis:	2.596	**Cond. No.**	982.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

6) 模型预测

1. 预测并查看结果

[Code 008]:

```
# 使用 predict 对 x 进行预测
y_hat = model_result.predict(x)
# 将预测结果转换为 DataFrame 格式方便转换，并将列名保存为'pred'
y_hat = pd.DataFrame(y_hat ,columns=['pred'])
# 合并原始数据和预测值并查看
df_merge = df.merge(y_hat,left_index=True,right_index=True)
df.head()
```

```
y_hat = model_result.predict(x)
y_hat = pd.DataFrame(y_hat,columns=['pred'])
df_merge = df.merge(y_hat,left_index=True,right_index=True)
df_merge.head()
```

	height	weight	pred
0	58	115	112.583333
1	59	117	116.033333
2	60	120	119.483333
3	61	123	122.933333
4	62	126	126.383333

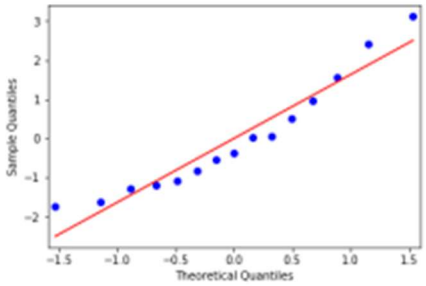
7) 回归诊断

1. 绘制残差 qq 图

[Code 009]:

sm.qqplot(model_result.resid,line='r')
plt.show()

```
sm.qqplot(model_result.resid,line='r')
plt.show()
```



8) 实验结论

? ? ? ? ? ?