HW2-影响超重的因素

崔孝荣　15320171151887　何嘉欣　15320171151899

March 2019

目录

1 引言

线性回归可以说是实证分析的基础，也是实证分析的起点。除了其简便易于操作的优点外，对回归结果的分析也能够很好地解释变量之间的关系。如今，许多计算机编程软件能够实现多种不同的线性回归，本次作业我们拟以 Python 跟 Stata 两种常见的软件为例，试研究影响 BMI 指数的因素，以及他们之间的关系。

2 数据变量

2.1 数据样本

本次作业运用 2016 年中国家庭追踪调查（CFPS）数据进行实证分析，主要采用成人问卷的数据。CFPS 于 2010 年开始基线调查，并持续对个人进行追踪，本次作业运用 2016 年对 16 岁以上成人的追踪调查结果的数据进行实证分析。

2.2 变量定义

因变量：本次作业的因变量为 BMI 指数，由于 CFPS 数据里没有直接的 BMI 数据，因此我们采用以下计算公式来计算 BMI 指数，$BMI = (weight/2)/(height/100)^2$，备注：CFPS 里的体重数据以斤为单位，身高数据以厘米为单位，身高范围为 0-195 厘米，体重范围为 0-260 斤。

自变量：本次作业的自变量为工作日睡眠时长（$sleep$），年龄（age）及工作日锻炼时长（$exercise$）。工作日睡眠时长范围为 0-24 小时，年龄范围为 16-98 岁。控制变量：本次作业的控制变量为半年内是否有慢性疾病，半年内有慢性疾病为 1，否则为 0。

因变量、自变量的最大值、最小值、均值、标准差见图 1。

```
. summarize height weight age sleep exercise
```

Variable	Obs	Mean	Std. Dev.	Min	Max
height	5,917	166.2631	8.290138	90	195
weight	33,170	122.934	22.99365	50	260
age	33,288	45.76938	16.90851	16	98
sleep	23,463	7.616899	1.45246	.1	24
exercise	13,856	8.312659	10.49891	.1	105

图 1: 因变量，自变量的基本情况

2.3 数据处理

对缺失数据的处理：本次作业没有剔除缺失数据，而是将缺失数据以 . 替代，如在 Stata 中输入以下语句：$replace \quad age = . \quad if \quad age < 0$，再如 $replace \quad height = . \quad if \quad height < 0$。缺失值的情况如图 2。从图 2 可看出，$BMI$ 的缺失情况比较严重，由于身高数据的缺失或是体重数据的缺失导致，而年龄数据的缺失最少，总体而言数据缺失较多，因此本次作业采用 bootstrap regress 方法来减少缺失值对回归效果的影响。

```
. misstable summarize BMI age sleep exercise
```

Variable	Obs=.	Obs>.	Obs<.	Obs<. Unique values	Min	Max
BMI	27,403		5,893	>500	8.891624	93.82716
age	8		33,288	82	16	98
sleep	9,833		23,463	42	.1	24
exercise	19,440		13,856	114	.1	105

图 2: 缺失值情况统计图

3 计量模型

本作业的回归模型如下：

$$BMI = \beta_1 sleep + \beta_2 exercise + \beta_3 age + \gamma X_i + \epsilon_i \tag{1}$$

其中，BMI 指数计算公式如上所述，sleep 表示工作日休息时长，exercise 表示工作日锻炼时长，age 表示受访者年龄，X_i 表示控制变量，本作业我们以半年内是否有慢性疾病作为控制变量，存在慢性疾病为 1，否则为 0。以下是运用 Stata 及 Python 做 bootstrap regress 的结果。

4 stata 回归结果

运用 Stata 做 bootstrap 回归得到图 3 所示结果：（bootstrap 次数为 50）从回归结果可以看出，该模型的解释力度不够，拟合优度仅 0.05，且 $sleep$, $exercise$ 的系数均不显著，只有 age 的系数为显著的。对上述模型系数不显著及解释力度不足的解释为：

（1）影响 BMI 指数的因素有许多，模型中缺少了关键的因素；

（2）模型设置为线性不合适，重新选择模型；

（3）自变量关系与因变量关系不密切；各变量关系散点图：

```
Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Linear regression                               Number of obs     =      1,232
                                                Replications      =         50
                                                Wald chi2(3)      =      64.11
                                                Prob > chi2       =     0.0000
                                                R-squared         =     0.0501
                                                Adj R-squared     =     0.0478
                                                Root MSE          =     3.1571
```

BMI	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal-based [95% Conf.	Interval]
sleep	-.0954034	.0692086	-1.38	0.168	-.2310498	.040243
exercise	.0120514	.0087419	1.38	0.168	-.0050824	.0291853
age	.0699675	.0091854	7.62	0.000	.0519644	.0879705
_cons	20.65543	.62361	33.12	0.000	19.43317	21.87768

图 3: Stata 的 Bootstrap 回归结果

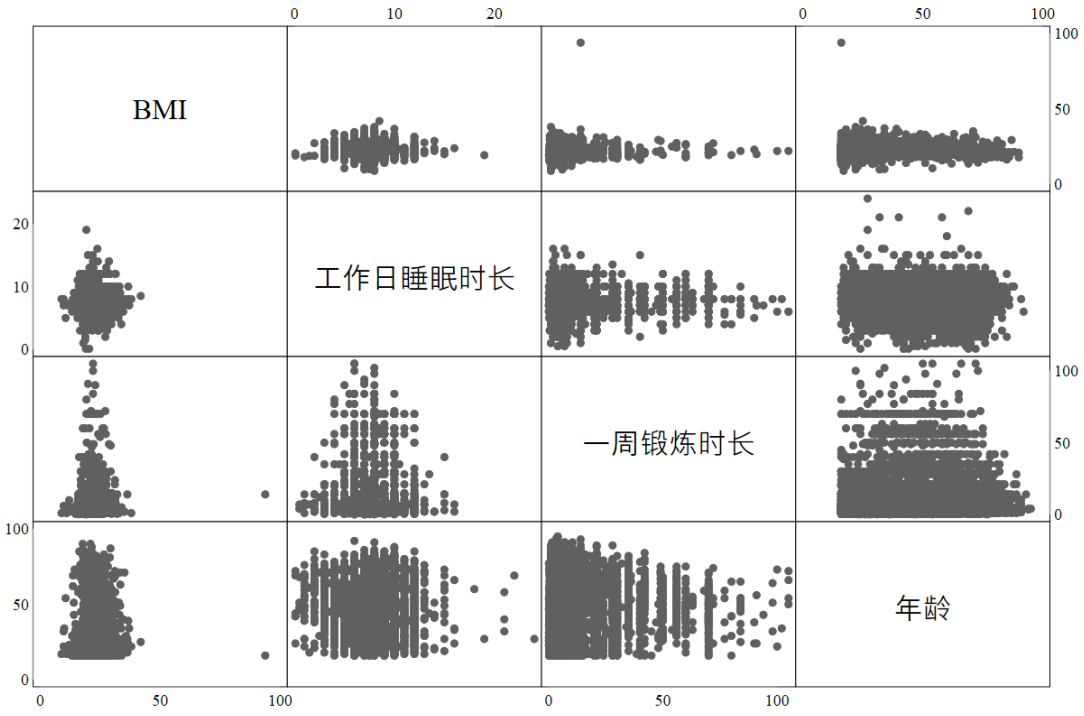

图 4: 各变量关系散点图

5 python 处理

5.1 数据处理

首先将数据导入 python，并进行数据清洗，检查是否含有缺省值、删除缺省值所在的记录进行后续分析。（本部分代码见附录）

画出变量的箱线图，通过箱线图我们可以看出年龄和一周运动时间的数据是比较分散的，睡眠时间和 BMI 数据则相对集中。

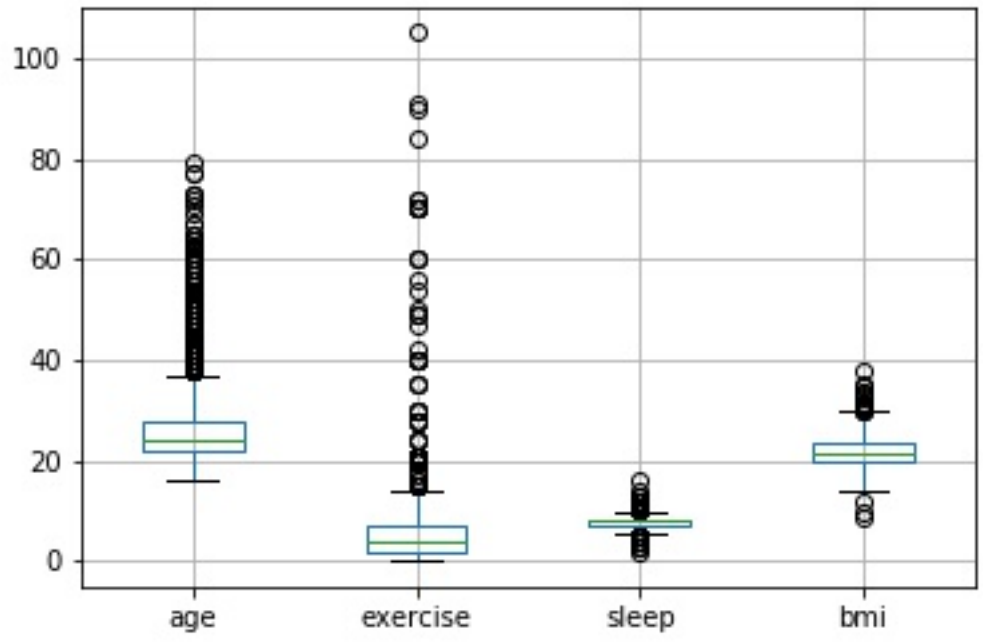

图 5: 箱型图

5.2 处理步骤

利用 python 的 sklearn 包进行线性回归。

第一步，建立训练集与测试集，利用 sklearn 中的 train_test_split 函数，将 80% 的随机样本设为训练集，剩余的 20% 的数据设为测试集。

第二步，将第一步得到的训练集中的特征值与标签值放入 LinearRegression() 模型中且使用 fit 函数进行训练，在模型训练完成之后，利用 intercept_ 与 coef_ 方法会得到所对应的方程式（线性回归方程式）需要估计的参数。

第三步，对数据集进行预测与模型测评。通过 prodict 和 score 对模型进行测评，并且画出测试集上预测值和真实值的差距（见图 6）

5.3 回归结果与效果

通过上述的分析过程以及软件代码的运行，我们可以得到

$$BMI = -0.0346sleep + 0.1076exercise + 0.0667age + 19.861 + \epsilon_i \tag{2}$$

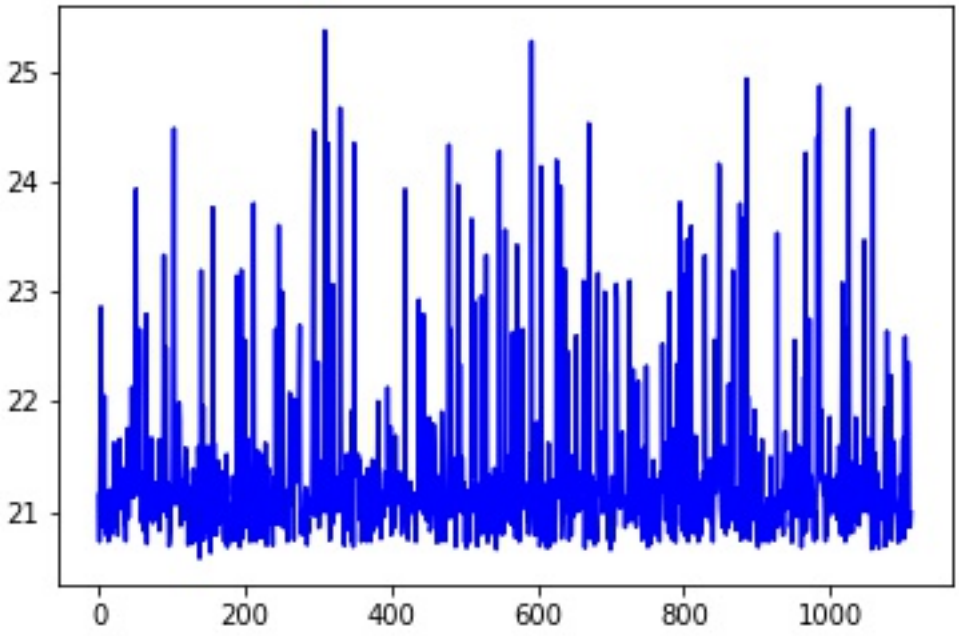

图 6: 测试集测试结果

通过学习效果的分析，我们发现这几个变量对超重的影响是不显著的，说明模型缺少了关键的决定性的变量。（结果同 stata 分析的结论）

参考文献

[1] A.Collin Cameron, Pravin K Trivedi. *Microeconometrics Using Stata*. Stata Press. 706 pp.

6 附录

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cross_validation import train_test_split
from sklearn import linear_model

#导入数据
dt=pd.read_stata(r"C:\Users\smile\Desktop\data_drop_allnone.dta")
#处理缺省值
np.isnan(dt).any()
dt.dropna(inplace=True)
np.isnan(dt).any()
#数据描述
print(dt.describe())
dt.boxplot()
plt.savefig("boxplot.jpg")
plt.show
print(dt.corr())

#通过加入一个参数kind='reg'，seaborn可以添加一条最佳拟合直线和95%的置信带。
sns.pairplot(dt, x_vars=['age','exercise','sleep'], y_vars='bmi', size=7, aspect=0.8,kind = 'reg')
plt.savefig("pairplot.jpg")
plt.show()
#training
X_train,X_test,Y_train,Y_test = train_test_split(dt.ix[:,:3],dt.bmi,train_size=.80)
print("原始数据特征:",dt.ix[:,:3].shape,",训练数据特征:",X_train.shape,",测试数据特征:",X_test.shape)
print("原始数据标签:",dt.bmi.shape,",训练数据标签:",Y_train.shape,",测试数据标签:",Y_test.shape)

model=linear_model.LinearRegression()
model.fit(X_train,Y_train)
a= model.intercept_#截距
b= model.coef_#回归系数
print("最佳拟合线:截距",a,",回归系数： ",b)

score=model.score(X_test,Y_test)
print(score)
Y_pre=model.predict(X_test)
print(Y_pre)

plt.plot(range(len(Y_pre)),Y_pre,"b",label='predict')
plt.savefig('predict.jpg')
plt.show()
```

图 7: python 代码