4.16

解：3.11 异常值检验

```
infl = result.get_influence()
sm_fr = infl.summary_frame()
sm_fr
```

✓ 0.2s

	dfb_const	dfb_x1	dfb_x2	dfb_x3	cooks_d	standard_resid	hat_diag	dffits_internal	student_resid	dffits
0	-0.199660	-0.150689	0.398722	0.265408	0.166087	-0.893528	0.454180	-0.815075	-0.876041	-0.799124
1	-0.168430	0.250341	-0.009143	-0.066807	0.031145	0.627667	0.240249	0.352959	0.592770	0.333335
2	0.090299	-0.108642	-0.007646	0.015643	0.006201	0.265166	0.260788	0.157499	0.243493	0.144626
3	0.000908	-0.000927	-0.000318	-0.000310	0.000001	-0.004333	0.199347	-0.002162	-0.003955	-0.001974
4	-0.481260	0.868702	0.014848	-1.145821	0.408741	1.753995	0.347018	1.278658	2.293834	1.672198
5	3.194712	-0.295813	-5.623304	4.593966	3.216007	-2.115660	0.741868	-3.586646	-3.832135	-6.496560
6	0.438260	-0.683109	0.251586	-0.741102	0.501103	-1.173475	0.592766	-1.415772	-1.220391	-1.472375
7	-0.970595	0.605480	0.833391	-0.481180	0.289458	-1.162813	0.461294	-1.076026	-1.206058	-1.116043
8	-0.026935	-0.054291	0.098349	0.069962	0.015001	0.409354	0.263664	0.244955	0.379017	0.226802
9	0.563738	-0.766997	-0.042999	0.533884	0.221576	1.064620	0.438825	0.941437	1.079114	0.954255

其中，$standard\_resid_i = ZRE_i = \frac{d_i - \hat{y}_i}{\hat{\sigma}}$

$studard\_resid_i = SRE_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$

$H = X(X^TX)^{-1}X^T$，h_{ii} 是第 i 个对角元素值　$hat\_diag_i = h_{ii}$

$cooks\_d = D_i = \frac{e_i^2}{(p+1)\hat{\sigma}^2} \cdot \frac{h_{ii}}{(1-h_{ii})^2}$

$\overline{ch} = \frac{p}{n} = 0.3$

由计算结果：

$|SRE_6| = 3.832135 > 3$

$\Rightarrow$ 第6个数据为异常值

$h_{55} = 0.741868 \Rightarrow ch_{55} = h_{55} - \frac{1}{10} = 0.641868 > 2\overline{ch}$

$\Rightarrow$ 从杠杆值看，第6个数据是自变量异常值

再 $D_5 = 3.216007 > 1$ 且位居第一

$\Rightarrow$ 第6个数据为异常值是由自变量异常与因变量异常两个原因共同引起的。

Ch 5

5.8

赋值原则：$\alpha_{entry} < \alpha_{removal}$

为保证 α_{entry} 赋值方法：

增大 α_{entry}，使得更多自变量 p-Value 在 α_{entry} 范围内，但仍须保持 $\alpha_{entry} < \alpha_{removal}$

5.10

解：(1) 令 $Y = X\beta + \varepsilon$

$$\hat{\beta} = (X'X)^{-1}X'Y = \begin{pmatrix} 5922.8274 \\ 4.8642 \\ 2.3741 \\ -817.9013 \\ 14.5387 \\ -846.8669 \end{pmatrix}$$

$\Rightarrow \hat{y}_i = 5922.8274 + 4.8642x_{2i} + 2.3741x_{3i} - 817.9013x_{4i} + 14.5387x_{5i} - 846.8669x_{6i}$

(2) 用 Python 写算法并附用法如下：

```
def backwards(data,label,cols_all):
    p = True
    # 全部变量，一个都不剔除，计算初始aic
    X_col = cols_all.copy()
    X = sm.add_constant(data[X_col])
    y = data[label]
    ols = sm.OLS(y, X).fit()
    AIC_None_value = ols.aic
    while p:
        # 剔除一个字段提取aic最小的字段
        AIC = {}
        for col in cols_all:
            # print(col)
            X_col = [i for i in cols_all if i!=col]
            X = sm.add_constant(data[X_col])
            ols = sm.OLS(y, X).fit()
            AIC[col] = ols.aic
        AIC_min_value = min(AIC.values())
        AIC_min_key = min(AIC, key=AIC.get)
        # 如果最小的aic小于不剔除该变量时的aic，则剔除该变量，否则停止
        if AIC_min_value < AIC_None_value:
            cols_all.remove(AIC_min_key)
            AIC_None_value = AIC_min_value
            p = True
        else:
            break
    select_col = cols_all
    ######################## 提取
    X = sm.add_constant(data[select_col])
    ols = sm.OLS(y, X).fit()
    summary = ols.summary()
    AIC = ols.aic
    return select_col,summary,AIC
```

✓ 0.1s

```
name = data.columns
select_col,summary,AIC = backwards(data,name[6],list(name[1:6]))
print(select_col,AIC)
summary
```

✓ 0.3s

```
['x2', 'x3', 'x4', 'x6'] 253.95484522374255

D:\py\lib\site-packages\scipy\stats\stats.py:1603: UserWarning: kurtosistest only valid for n>=20 ... continuing "
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

OLS Regression Results

Dep. Variable:	y	R-squared:	0.824
Model:	OLS	Adj. R-squared:	0.759
Method:	Least Squares	F-statistic:	12.84
Date:	Thu, 07 Apr 2022	Prob (F-statistic):	0.000397
Time:	23:02:14	Log-Likelihood:	-121.98
No. Observations:	16	AIC:	254.0
Df Residuals:	11	BIC:	257.8
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	6007.3203	2245.481	2.675	0.022	1065.051	1.09e+04
x2	5.0681	1.360	3.727	0.003	2.075	8.061
x3	2.3078	0.486	4.750	0.001	1.238	3.377
x4	-824.2614	167.776	-4.913	0.000	-1193.535	-454.988
x6	-862.6990	232.489	-3.711	0.003	-1374.403	-350.995

Omnibus:	2.659	Durbin-Watson:	2.478
Prob(Omnibus):	0.265	Jarque-Bera (JB):	1.105
Skew:	0.617	Prob(JB):	0.575
Kurtosis:	3.368	Cond. No.	3.10e+04

(3)（Python 自己写函数太麻烦了找不到对应的库，故使用 R）

```
library(readxl)

data <- read_xls("data.xls")
# x <- data[2:6]
# y <- data[7]

ols <- lm(y~., data=data[-1])
ols_step <- step(ols,direction='both')
print(summary(ols_step))
```

10:1 (Top Level)

Console Terminal Jobs

R 4.1.2 · C:/Users/Moriarty/Desktop/数学/应用统计方法/hw第七周/

```
> print(summary(ols_step))

Call:
lm(formula = y ~ x2 + x3 + x4 + x6, data = data[-1])

Residuals:
    Min      1Q  Median      3Q     Max
-877.60 -262.89  -19.53  172.95 1165.46

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 6007.3203  2245.4805   2.675 0.021589 *
x2             5.0681     1.3600   3.727 0.003343 **
x3             2.3078     0.4858   4.750 0.000600 ***
x4          -824.2614   167.7764  -4.913 0.000462 ***
x6          -862.6990   232.4888  -3.711 0.003437 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 597 on 11 degrees of freedom
Multiple R-squared:  0.8235,    Adjusted R-squared:  0.7594
F-statistic: 12.84 on 4 and 11 DF,  p-value: 0.0003973

>
```

(4) 采用两种方法最终都是剔除了 x_5，

但后退法从全模型开始，通过最小 AIC 信息统计量，每一步剔除一个变量，中途不会再选入变量；逐步回归法有进有出，在加入新变量后的步骤中若新加入新变量使得原有的变量不显著（很不显著），则需删去原来的那个变量，最终保留最后得到的子集是最优回归子集。

PB19051769

马