时间序列

姓名：范潇　学号：2254298　日期：2024 年 6 月 11 日

1. (8.1)

我利用 Python 完成了本题的求解。

采用趋势移动平均法所得到的模型的误差为 19.35.

当 $\alpha = 0.3$ 时，直线指数平滑预测模型的误差为 4.43；当 $\alpha = 0.6$ 时，直线指数平滑预测模型的误差为 7.10。

综上，采用 $\alpha = 0.3$ 的直线指数平滑预测模型较优。当使用该模型时，对于 1982 年和 1985 年产量的预测分别为 143.83 亿和 163.12 亿。

本题所用的代码如下:

```python
import numpy as np
from math import *
data = [
    [1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981],
    [80.8, 94.0, 88.4, 101.5, 110.3, 121.5, 134.7, 142.7]
]
val = [80.8, 94.0, 88.4, 101.5, 110.3, 121.5, 134.7, 142.7]
predict = val[:3]
for i in range(3,len(val)):
    predict.append(np.mean(np.array(val[i-3:i])))
err = sqrt(sum((np.array(val[3:])-np.array(predict[3:]))**2)/(len(val)-3))
print("================␣␣Q1␣==================")
print(f"true␣value:{val}")
print(f"prediction:{predict}")
print(f"err:{err}")



s1,s2 = 87.7,87.7
S1=[s1]
S2=[s2]
alpha = 0.3
for i in range(2,len(val)):
    S1.append(alpha*val[i]+(1-alpha)*S1[-1])
```

```
    S2.append(alpha*S1[-1]+(1-alpha)*S2[-1])
exp_prediction = []
exp_prediction.append(s1)
for i in range(1,len(val)):
    exp_prediction.append((1+1/(1-alpha))*S1[i-1]-S2[i-1]/(1-alpha))

err = sqrt(sum((np.array(val[:])-np.array(exp_prediction[:]))**2)/(len(val))
    )
print("================  Q2  ====================")
print("alpha = 0.3")
print(f"true value:{val}")
print(f"prediction:{exp_prediction}")
print(f"err:{err}")

S1=[s1]
S2=[s2]
alpha = 0.6
for i in range(2,len(val)):
    S1.append(alpha*val[i]+(1-alpha)*S1[-1])
    S2.append(alpha*S1[-1]+(1-alpha)*S2[-1])
exp_prediction = []
exp_prediction.append(s1)
for i in range(1,len(val)):
    exp_prediction.append((1+1/(1-alpha))*S1[i-1]-S2[i-1]/(1-alpha))

err = sqrt(sum((np.array(val[:])-np.array(exp_prediction[:]))**2)/(len(val))
    )
print("alpha = 0.6")
print(f"true value:{val}")
print(f"prediction:{exp_prediction}")
print(f"err:{err}")


s1,s2 = 87.7,87.7
S1=[s1]
S2=[s2]
alpha = 0.3
for i in range(2,len(val)):
    S1.append(alpha*val[i]+(1-alpha)*S1[-1])
    S2.append(alpha*S1[-1]+(1-alpha)*S2[-1])
exp_prediction = []
exp_prediction.append(s1)
```

```
for i in range(1,len(val)):
    exp_prediction.append((1+1/(1-alpha))*S1[i-1]-S2[i-1]/(1-alpha))

err = sqrt(sum((np.array(val[:])-np.array(exp_prediction[:]))**2)/(len(val))
    )
print("================  Q3  ====================")
print(S1)
print(S2)
a = 2*S1[-1]-S2[-1]
b = alpha*(S1[-1]-S2[-1])/(1-alpha)
print(a,b)
print("1982预测产量为: ",a+b)
print("1985预测产量为: ",a+4*b)
```

输出为图 1

```
================  Q1 ====================
true value:[80.8, 94.0, 88.4, 101.5, 110.3, 121.5, 134.7, 142.7]
prediction:[80.8, 94.0, 88.4, 87.73333333333335, 94.63333333333333, 100.06666666666666, 111.10000000000001, 122.16666666666667]
err:19.354230086010187
================  Q2 ====================
alpha = 0.3
true value:[80.8, 94.0, 88.4, 101.5, 110.3, 121.5, 134.7, 142.7]
prediction:[87.7, 87.70000000000003, 88.11999999999999, 96.21100000000003, 105.93160000000003, 117.80785, 131.879506, 143.82846189999995]
err:4.434177260866532
alpha = 0.6
true value:[80.8, 94.0, 88.4, 101.5, 110.3, 121.5, 134.7, 142.7]
prediction:[87.7, 87.69999999999999, 88.54000000000002, 104.344, 116.40880000000001, 129.57999999999998, 144.61859200000004, 153.05407360000004]
err:7.102790655938367
================  Q3 ====================
[87.7, 87.91, 91.987, 97.48089999999999, 104.68662999999998, 113.69064099999997, 122.39344869999996]
[87.7, 87.763, 89.0302, 91.56540999999999, 95.50177599999998, 100.95843549999998, 107.38893945999997]
137.39795793999997 6.430503959999996
1982预测产量为:  143.82846189999998
1985预测产量为:  163.11997377999995
```

图 1: 第一题程序输出

2. (8.2)

我利用 Python 完成了本题的求解。

首先我分别测试了 α 值为 $0.1, 0.2, \cdots, 0.9$ 时三次指数平滑法得到的模型拟合误差。经过比较，当 $\alpha = 0.5$ 时，模型拟合误差最小，所以最后选取 $\alpha = 0.5$。这样得到的三次指数平滑法对于 1983 年和 1985 年的全国社会商品零售额分别为 2858.3 亿元和 3465.8 亿元。

本题所用的代码如下:

```
import numpy as np
from math import *
data = [[1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970,
   1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982],
 [696.9, 607.7, 604.0, 604.5, 638.2, 670.3, 732.8, 770.5, 737.3, 801.5,
    858.0, 929.2, 1023.3, 1106.7, 1163.6, 1271.1, 1339.4, 1432.8, 1558.6,
    1800.0, 2140.0, 2350.0, 2570.0]]
val = [696.9, 607.7, 604.0, 604.5, 638.2, 670.3, 732.8, 770.5, 737.3, 801.5,
    858.0, 929.2, 1023.3, 1106.7, 1163.6, 1271.1, 1339.4, 1432.8, 1558.6,
   1800.0, 2140.0, 2350.0, 2570.0]

def third_exp_predict(alpha):
    init = np.mean(val[:3])
    S1,S2,S3 = [init],[init],[init]
    for i in range(1,len(val)):
        S1.append(alpha*val[i]+(1-alpha)*S1[i-1])
        S2.append(alpha*S1[i]+(1-alpha)*S2[i-1])
        S3.append(alpha*S2[i]+(1-alpha)*S3[i-1])
    prediction = [init]
    for i in range(1,len(val)):
        prediction.append((3-3*alpha+alpha*alpha)*S1[i-1]/(1-alpha)**2-(3
           -alpha)*S2[i-1]/(1-alpha)**2 + S3[i-1]/(1-alpha)**2)
    err = sqrt(sum((np.array(prediction)-np.array(val))**2)/len(val))

    print(alpha)
    print(err)
    print(S1)
    print(S2)
    print(S3)
# 用于选取最佳alpha值
# alphas = np.linspace(0.1,0.9,9)
# for i in range(len(alphas)):
#     third_exp_predict(alphas[i])

# alpha取0.5时err最小
best = 0.5
```

```
s1 = 2343.3
s2 = 2122.8
s3 = 1920.8
a = 3*s1-3*s2+s3
b = ((6-5*best)*s1-2*(5-4*best)*s2+(4-3*best)*s3)*best/(2*(1-best)*(1-best))
c = (s1-2*s2+s3)*best*best/(2*(1-best)*(1-best))
# print("1982年预测值：",a)
print("1983年预测值：",a+b+c)
print("1985年预测值：",a+3*b+9*c)
```

3. (8.3)

我利用 Python 完成了本题的求解。

首先我绘制了所给数据的折线图，如图 2。

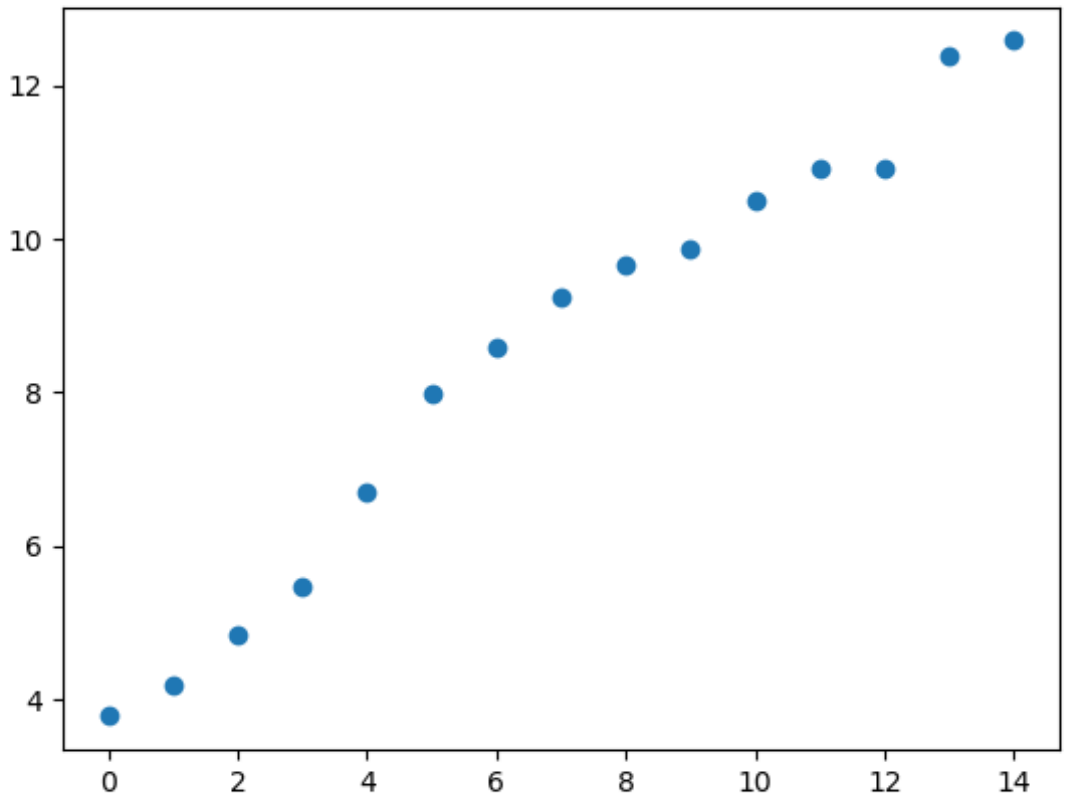

图 2: 第二题数据

可以看到，数据大致成线性，因此采用直线指数平滑预测模型和一阶差分指数平滑模型。

通过分别设定 $\alpha = 0.1, 0.2, \cdots, 0.9$，得到对于直线指数平滑模型，最佳参数为 $\alpha = 0.3$，对应的拟合误差为 0.48；对于一阶差分指数平滑模型，最佳参数为 $\alpha = 0.4$，对应的拟合误差为 0.5.

选取最佳参数后，直线指数平滑模型预测出的 1985 年和 1990 年的粮食产量分别为 13.7，16.4；一阶差分平滑模型预测出的 1985 年和 1990 年的粮食产量分别为 13.1,16.3。

本题所用的代码如下:

```
import matplotlib.pyplot as plt
import numpy as np
from math import *
val = [3.78,4.19,4.83,5.46,6.71,7.99,8.60,9.24,9.67,9.87,10.49,10.92,10.93,
       12.39,12.59,]
plt.figure()
plt.scatter(range(len(val)),  val)
plt.show()
```

```
9
10 def second_exp_predict(alpha):
11     init = np.mean(val[:2])
12     S1,S2 = [init],[init]
13     for i in range(2,len(val)):
14         S1.append(alpha*val[i]+(1-alpha)*S1[-1])
15         S2.append(alpha*S1[-1]+(1-alpha)*S2[-1])
16     prediction = [init]
17     for i in range(1,len(val)):
18         prediction.append((1+1/(1-alpha))*S1[i-1]-S2[i-1]/(1-alpha))
19     err = sqrt(sum((np.array(prediction)-np.array(val))**2)/len(val))
20
21     print(alpha)
22     print(err)
23     print(S1)
24     print(S2)
25
26 def first_diff_predict(alpha):
27     init = np.mean(val[:2])
28     predict_diff =[0]
29     prediction = [val[0]]
30     for i in range(1,len(val)):
31         predict_diff.append(alpha*diff[i-1]+(1-alpha)*predict_diff[i-1])
32         prediction.append(predict_diff[i]+val[i-1])
33
34     err = sqrt(sum((np.array(prediction)-np.array(val))**2)/len(val))
35     print(alpha)
36     print(predict_diff)
37     print(err)
38
39 # 用于选取最佳alpha值
40 # alphas = np.linspace(0.1,0.9,9)
41 # for i in range(len(alphas)):
42 #     print()
43 #     second_exp_predict(alphas[i])
44
45 best = 0.3
46 s1 = 11.33
47 s2 = 10.05
48 a = 2*s1-s2
49 b = best*(s1-s2)/(1-best)
50 print("二次指数平滑1985年预测值: ",a+2*b)
```

```
print("二次指数平滑1990年预测值: ",a+7*b)

diff = [0] + [val[i]-val[i-1] for i in range(1,len(val))]

# 用于选取最佳alpha值
# alphas = np.linspace(0.1,0.9,9)
# for i in range(len(alphas)):
#     print()
#     first_diff_predict(alphas[i])
best = 0.4
diff_predict = best*diff[-1]+(1-best)*0.761
print("一次差分平滑1985年预测值: ",12.59+diff_predict)
print("一次差分平滑1990年预测值: ",12.59+7*diff_predict)
```

4. (8.4)

我在 Matlab 的实时编辑器中完成了本题的求解。

得到的模型为

$$y_t = 1.2276y_{t-1} - 0.68478y_{t-2} + \varepsilon_t - 0.5022\varepsilon_{t-1}$$

其中 $y_t = x_{t+1} - x_t$ 为差分数组。

得到的 10 步预测值分别为 6470, 6879, 7393, 8027, 8743, 9483, 10202, 10882, 11536, 12190.

实时编辑器页面如下一页所示。

5. (8.5)

我在 Matlab 的实时编辑器中完成了本题的求解。

首先我绘制了所给数据的折线图，如图 3。

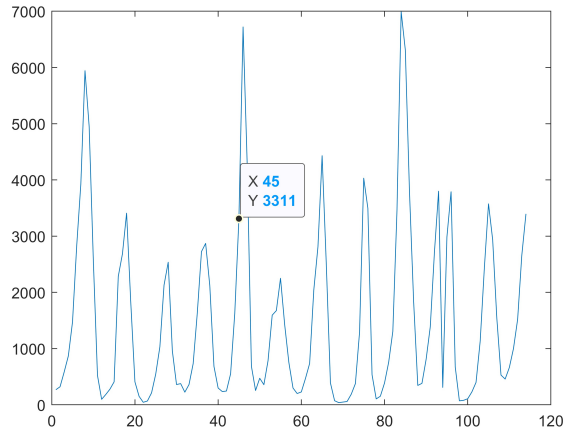

图 3: 第五题数据

可以看到，数据成周期性，周期大致为 10。所以我先对原始数据进行差分处理，转化为差分序列 $y_t = x_{t+10} - x_t$。

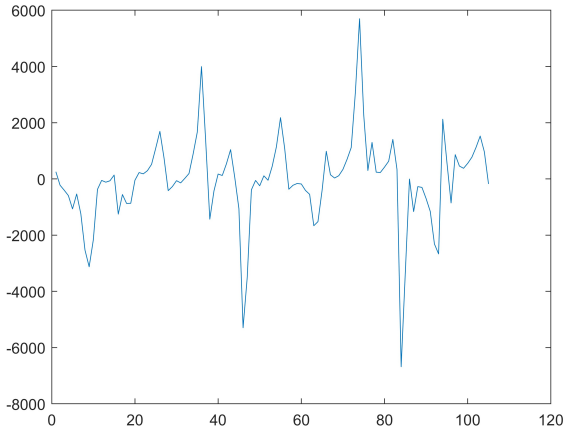

图 4: 差分序列

从图 4可以看到，经过差分后，在大部分时间段内，序列较平稳。

接着，我尝试建立 ARMA 模型，根据 AIC 值选取最佳参数。经比较，当 p=3,q=10 时，AIC 值最小。同时，在该参数下，模型能够通过 LBQ 检验。

通过选取出的模型得到的后两年的预测值分别为 2737 和 2077。

实时编辑器页面如后所示。

6. (8.6)

我在 Matlab 的实时编辑器中完成了本题的求解。

首先我绘制了所给数据的折线图，如图 5。

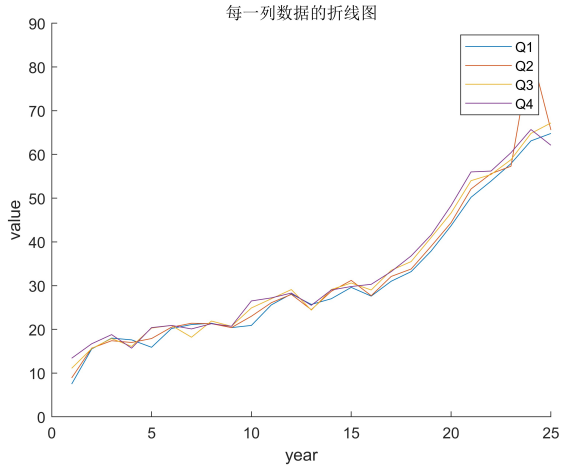

图 5: 第六题数据

可以看到，各个季度的变化趋势基本一致，且相对大小较为稳定。因此，我先以年为单位进行时间序列模型。然后我计算根据所给数据计算出各季度的数据对于一整年数据的贡献平均占比。综合两者，得到对于未来各季度的估计。

对于时间序列，我尝试建立 ARMA 模型，根据 AIC 值选取最佳参数。经比较，当 p=2,q=3 时，AIC 值最小。同时，在该参数下，模型能够通过 LBQ 检验。

通过选取出的模型得到的后两年的预测值分别为 242 和 237。乘以比例系数后，得到后 8 各季度的预测数据分别为 58.3,61.0,61.0,61.9,57.3,59.9,59.9,60.8。

实时编辑器页面如后所示。