МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени
первого Президента России Б. Н. Ельцина»

ВЫДЕЛЕНИЕ ТРЕНДОВОЙ СОСТАВЛЯЮЩЕЙ ВРЕМЕННОГО РЯДА

Методические указания к выполнению практического задания № 3

Екатеринбург
2020

Содержание

Введение.....3

1. Задание на лабораторную работу 3

2. Требования к оформлению отчета.....11

Введение

На этой лабораторной работе мы впервые приступаем к декомпозиции временных рядов на простейшие компоненты, одной из которых является **тренд**. В ходе работы студентами будут изучены такие способы построения кривых тренда, как регрессионные методы подгонки, методы скользящего сглаживания, и другие. Более того, регрессионные кривые могут использоваться не только для построения простых трендов, но и более сложных моделей временных рядов.

1. Задание на лабораторную работу

Результатом выполнения лабораторной работы является оформленный отчет в виде *Jupyter*-тетради, в котором должны быть представлены и отражены все нижеперечисленные пункты:

1) Сначала импортируйте в свой код нужные библиотеки, функции и т.д.

```
import numpy as np
import numpy.random as rand
import matplotlib.pyplot as plt
import pandas as pd
from scipy import signal
import scipy.stats as stats
from statsmodels.tsa import api as tsa
%matplotlib inline
```

2) В зависимости от своего варианта, который определяется по последним двум цифрам студ. билета, из таблицы 1 на следующих страницах создать временной ряд из столбца (всего 24 точки). ВР определен на временном интервале от 0 до 1 (далее переменная $\boldsymbol{t}$).

1	2	3	4	5	6	7	8
1,65	23,46	0,54	30,42	20,89	12,60	3,54	15,48
2,59	14,86	2,16	30,56	20,11	18,92	7,81	9,29
6,18	20,14	5,39	29,90	16,41	17,08	12,83	8,26
6,26	21,59	3,48	21,67	18,95	15,51	6,73	5,45
6,44	18,98	4,54	26,31	21,43	8,97	6,29	10,49
7,16	21,77	7,99	28,13	16,54	14,52	15,88	14,47
10,56	20,27	7,95	24,06	11,55	12,77	12,27	9,46
10,93	16,86	7,01	20,55	14,39	12,96	7,84	8,79
9,53	16,23	9,89	24,35	20,66	5,55	10,71	12,96
10,64	18,55	12,35	18,12	15,31	11,09	14,60	15,37
17,43	14,87	12,91	18,69	9,34	9,23	17,48	11,82
14,72	11,98	14,42	14,88	11,39	5,03	12,97	11,34
15,50	14,41	14,13	11,66	11,34	2,15	11,34	20,84
15,01	13,42	18,67	19,83	10,07	8,95	23,82	16,58
17,83	10,44	16,95	14,10	5,95	8,04	19,97	12,47
18,43	8,26	15,84	10,16	4,59	5,68	11,51	7,05
17,69	8,86	19,23	10,08	8,74	0,14	18,07	15,08
19,80	9,53	22,05	5,82	9,96	5,85	22,11	16,97
22,64	6,88	22,59	8,46	3,03	4,21	23,12	13,51
22,86	4,10	21,15	5,50	3,17	2,56	15,52	13,45
21,56	7,61	23,98	3,60	4,45	0,08	20,03	16,55
22,16	4,92	26,45	8,44	4,06	3,87	24,36	18,47
25,82	1,79	29,80	3,04	0,16	1,10	27,02	21,73
26,50	0,10	27,41	0,00	1,52	0,85	21,31	14,04

9	10	11	12	13	14	15	16
12,19	23,75	18,47	76,88	8,48	24,78	3,07	10,22
8,41	28,00	14,87	69,88	10,43	22,55	6,26	10,06
14,68	33,01	21,51	74,55	18,97	30,85	7,46	13,34
8,64	16,78	9,07	59,75	6,37	23,88	6,48	11,92
32,94	18,16	16,02	72,21	9,86	27,78	1,64	8,81
22,61	20,05	11,12	66,85	1,29	12,71	5,41	8,10
45,92	3,18	23,45	69,91	13,23	25,25	6,18	12,51
23,63	16,11	6,45	68,05	8,50	25,70	16,93	11,16
18,59	21,66	14,21	72,59	11,68	34,44	2,71	8,77
36,22	20,16	8,18	42,83	10,17	23,18	6,94	4,87
50,10	24,71	14,50	67,04	14,18	29,81	8,35	10,57
46,22	15,63	3,86	56,63	2,79	22,26	11,59	10,37
23,63	16,27	10,14	61,10	26,63	22,97	5,98	6,88
47,30	18,99	9,99	44,88	15,69	16,37	10,77	9,13
40,03	21,12	14,47	52,90	20,32	22,82	14,71	10,31
56,53	8,34	0,65	46,03	17,28	14,19	14,66	7,13
38,41	14,96	8,97	46,72	22,87	16,40	11,77	3,52
51,47	17,17	2,47	46,48	23,80	7,23	27,10	0,14
6,29	20,24	12,58	31,63	28,81	13,05	9,69	6,35
35,41	8,31	3,12	21,72	28,59	4,63	22,31	5,30
67,79	12,36	6,81	21,40	35,68	3,19	19,73	1,46
74,21	14,59	0,43	11,40	35,72	4,55	25,88	1,09
79,12	21,72	4,65	10,06	39,44	0,94	29,00	2,40
45,10	28,69	5,91	0,42	40,04	11,07	32,18	1,92

17	18	19	20	21	22	23	24
11,54	0,54	6,86	11,43	10,41	4,89	15,45	5,93
0,80	4,33	3,91	7,60	7,70	3,10	11,94	3,88
12,76	3,73	6,66	12,15	10,39	5,19	11,93	5,08
11,18	5,18	6,38	10,39	10,73	1,02	18,66	5,98
8,90	2,50	8,35	11,44	12,31	6,25	12,69	7,77
8,49	3,72	6,16	10,94	9,58	5,06	10,01	6,67
11,38	4,78	7,68	13,54	11,53	5,96	8,81	6,55
10,93	5,72	7,12	11,87	11,55	6,27	10,86	6,27
9,40	3,69	8,61	13,35	13,98	6,56	11,49	8,23
9,30	4,80	5,87	11,72	10,07	6,43	10,78	6,61
12,43	6,35	7,76	13,58	11,44	6,45	10,38	7,40
11,03	6,89	7,07	10,56	11,00	6,26	13,07	7,48
10,88	6,38	8,37	11,04	11,16	7,00	10,81	8,08
11,33	5,93	8,69	8,96	9,49	4,51	12,73	7,00
13,86	9,17	6,83	11,38	10,41	5,93	12,11	6,16
14,98	9,31	6,17	9,26	9,15	6,53	15,74	5,73
12,66	4,07	6,98	9,38	8,48	6,98	17,71	7,23
12,98	9,47	3,84	8,04	5,41	8,96	15,31	3,86
18,09	12,28	4,75	10,98	6,44	5,78	11,15	5,63
17,49	13,32	4,05	7,95	6,15	5,87	18,12	5,66
14,97	9,87	4,88	7,67	3,17	6,21	20,81	5,71
14,42	12,73	0,51	4,69	0,47	3,33	19,90	2,62
21,29	16,73	2,60	7,16	1,80	5,21	19,15	3,89
20,66	17,05	0,90	4,17	1,26	4,63	22,43	3,44

3) Построить график заданного ряда.
4) Рассчитать регрессионную модель тренда первого порядка, то есть линейный тренд $\tau(t) = \beta_0 + \beta_1 t$. Для этого:
5) Сначала произвести оценку регрессионной модели $y = X\beta$.

   Для этого потребуется в матричном виде решить эту систему линейных уравнений.

6) Для линейного тренда $X = \begin{vmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{vmatrix}, y = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{vmatrix}, \beta = \begin{vmatrix} \beta_0 \\ \beta_1 \end{vmatrix}.$
7) Для решения Вам пригодится функция:

   B = np.linalg.lstsq(X,Y)

8) Из этого результата **B** коэффициенты находятся в нулевом элементе. Построим получившийся тренд:

```
B = B[0]    # забираем из результата коэффициенты β
print(B)
plt.figure(figsize = (10, 5))
plt.plot(t, Y)                    # строим исходный ВР
plt.plot(t, B[0] + B[1] * t , 'r')    # строим его тренд
plt.show()
```

9) Кроме матричных расчетов в Python, несомненно, существуют и готовые функции построения регрессионных кривых. Воспользуемся ими из нескольких библиотек.

10) На основе построения полиномиальных кривых из **numpy**:

```
bb = np.polyfit(t, Y, 1) # полиномиальная кривая 1-го порядка
plt.figure(figsize = (10, 5))
plt.plot(t, Y)
plt.plot(t, bb[1] + bb[0]*t, 'r') # Внимание! Коэф. β в другом порядке
plt.show()
```

Чтобы не ошибиться в порядке коэффициентов, лучше использовать функцию **poly1d**:

```
p = np.poly1d(bb)       # создаем экземпляр полинома
plt.figure(figsize = (10, 5))
plt.plot(t, Y)
# считаем значения полинома на заданной временной сетке
plt.plot(t, p(t), 'g')
plt.show()
```

11) На основе линейной регрессии из **scipy.stats**:

```
out = stats.linregress(t, Y)
print(out)          # выведет все коэффициенты и статистику регрессии
plt.figure(figsize = (10, 5))
plt.plot(t, Y)              # строим график кривой вместе с трендом
plt.plot(t, out.intercept + out.slope*t, 'r')
plt.show()
```

12) На основе подгонки кривых **curve_fit** из **scipy.optimize**:

```
def func(t, b0, b1):      # описываем функцию тренда
    return b0 + b1 * t    # линейный тренд с 2 параметрами

from scipy.optimize import curve_fit
popt, pcov = curve_fit(func, t, Y)    # проводим подгонку МНК
print(popt)       # получаем коэффициенты b0 & b1
print(pcov)       # ковариационная матрица ошибок подгонки
```

13) На основе библиотеки **sklearn**:

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(t.reshape(-1,1), Y)
print(reg.coef_)          # здесь выведется линейный коэффициент b1
print(reg.intercept_)     # здесь выведется коэффициент b0 (смещение)
print(reg.score(t.reshape(-1,1), Y))
# здесь будет выведена «оценка» (равная R^2) полученной регрессии,
чем ближе она к 1.0, тем лучше тренд
```

14) На основе **statsmodel**:

```
import statsmodels.api as sm
x_ = sm.add_constant(t.reshape(-1,1)) # создаем простую модель
                                      # вида τ(t) = β0 + β1t
smm = sm.OLS(Y, x_)   # используем Метод Наименьших Квадратов
                      # (МНК) (Ordinary Least Squares = OLS)
res = smm.fit()       # подгоняем параметры модели по МНК
print(res.params)     # получаем результирующие коэффициенты
```

15) Пример, приведенный выше, гораздо ближе по своей реализации уже к методам **машинного обучения**: сначала задается «форма» решаемой задачи, затем определяется метод ее решения и уже самим решением занимается ЭВМ.

16) Удостоверьтесь, что во всех реализациях получились одинаковые коэффициенты линейного тренда.

17) Аналогичным образом постройте модель тренда **второй** и **третьей** степени. Также постройте модель **экспоненциального** тренда: $\tau(t) = \beta_0 e^{\beta_1 t}$ Учтите, что не все приведенные выше методы для этого подойдут.

18) Все найденные тренды разной степени и модели нанесите на один график. Удостоверьтесь, что получились правильные результаты.

19) Теперь построим тренд методом сглаживания. Для этого напишите следующую функцию:

```
def smooth(x, window_len):
    if window_len<3:
        return x
    s=np.r_[2*x[0]-x[window_len-1::-1], x, 2*x[-1]-x[-1:-window_len:-1]]
    w=np.ones(window_len, 'd')
    y=np.convolve(w/w.sum(), s, mode='same')
    return y[window_len:-window_len+1]
```

20) Затем вызовите эту функцию для сглаживания ряда, например:

```
Smoothed_data = smooth(Y, 3) # сглаживание по 3 точкам
```

21) Постройте тренды, полученные методом скользящего сглаживания по **трем, семи** и **одиннадцати** точкам. Постройте каждый из них **отдельно**, но вместе с исходным ВР (то есть всего 3 рисунка по 2 графика в каждом).

22) Постройте **собственную функцию** сглаживания по трем точкам, на основе формул из лекции 4.
23) Постройте **собственную функцию** сглаживания по семи точкам, на основе формул из лекции 4.
24) Сравните получившиеся результаты – тренды должны получиться одинаковыми (отличия могут возникнуть только по краям временного интервала).
25) Наконец, постройте тренд методом **экспоненциального сглаживания**, самостоятельно подобрав его параметр (который лежит в диапазоне от 0 до 1).
26) Не забудьте в отчет-тетрадь добавить необходимые рисунки.

2. Требования к оформлению отчета

Отчет в Jupyter-тетради должен обязательно содержать: номер лабораторной работы, ФИО студента, номер варианта (либо студенческий номер), номер группы, результаты выполнения работы с комментариями студента (комментарии пишутся после #) и изображениями.