```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
data = pd.read_csv("fruit_data.csv")
data['color_score'] = data['color_score'].apply(lambda x: round(x, 2))
df = pd.DataFrame(data, columns=['fruit_name', 'height', 'color_score'])
data.head()
```

	mass	width	height	color_score	fruit_name
0	192	8.4	7.3	0.55	apple
1	150	8.0	6.8	0.59	apple
2	156	7.3	7.2	0.60	apple
3	168	8.2	7.1	0.72	mandarin
4	162	7.5	6.8	0.79	mandarin

با استفاده از کتابخانه ازمون کتابخانه از جمله پاندس توانستیم داده‌ست را خوانیم و در یک متغیر ذخیره کنیم

تمرین 1

```
# 1.class count
print("apple count:",df[df.fruit_name[df.fruit_name == 0].
avg_height_fruit0 = df[df.fruit_name == 0].
```

```
mass        125.120448
width         7.168640
height        7.690240
color_score   0.703040
dtype: float64
```

با استفاده از تابع مخصوص محاسبه میانگین، میانگین را برای هر ویژگی محاسبه کردیم

خطای بود به یک خطات بود متوجها نبود است 0 error

میانگین از اعداد رقمی بصورت ماشه در آن و تابع میانگین هم مخصوص به خودشون میباشد

تمرین 2

```
# 2.std dev
print("std dev")
width = pd.DataFrame(width)
print(width)
```

```
            mass       width      height  color_score
mass    1.000000  0.866026  0.389041  0.092021
width   0.866026  1.000000  0.367487  0.443342
height  0.389041  0.367487  1.000000  0.706828
color   0.092021  0.443342  0.706828  1.000000
fruit   0.471971  0.486773  0.074167  0.329532
```

خروجی واریانس برای دیتاست ماست که درواقع واریانس را برای هر هیچ یا سنوس محاسبه میکند

```
max(dataset.var())
```

```
mass    30197.667086
dtype: float64
```

```
min(dataset.var())
```

```
0.009497058879403
```

به سادگی بیشترین مقدار واریانس و کمترین آن را بدست آوردیم . اما سوال اینا نام هیچ دو هیچ‌ماند . پس به محاسبه ادامه میدهیم

```
cov=np.cov.var(dataset.var())
print("The max value for cov is :",cov,"for column","color_score")
df1 = pd.DataFrame(cov)
df1=df1 - 1
df1=df1.abs()
df1=df1.sort_values(0, ascending=0)
df1.head(1)
```

```
/home/\ DEC_DFIBERTTRANSOL_\LUCR\ 0.007497058403411,\ height\ 0.010278303051,\ 'color_score'\ 0.009497058879403
```

در این مرحله پس از محاسبه واریانس ها به کمک کتابخانه نامپای ، در ادامه با استفاده از دیکشنری یک مقدار میانگین واریانس ها را به هر هیچ ما نسبت میدهیم و در نهایت یک مقدار میانگین واریانس به مقدار خروجی میدهیم

```
print("The max value for cov is :",cov,"for column","color_score")
print(dataset.var())
```

```
The max value for cov is : 1.0
mass    30197.667086
dtype: float64
```

```
print("The max value for cov is :",list(dictvar.keys())[list(dictvar.values()).index(max(dictvar.values))] )
print(dataset.var())
```

```
The max value for cov is : mass
mass    30197.667086
dtype: float64
```

```
print("The max value for cov is :",list(dictvar.keys())[list(dictvar.values()).index(min(dictvar.values))] )
print(dataset.var())
```

```
The max value for cov is : color_score
mass    30197.667086
dtype: float64
```

حالا در انتها توانستیم مقادیر واریانس و نام هیچ یا به شکل دیکشنری ایجاد شده و نوع آماده برای پیدا کردن ماکسیمم و مینی ، پیدا کنیم و بزگردانیم

تمرین 4

```
# 4.max value
print("max value")
cov=np.cov.var(dataset.var())
cov=cov.round()

cov=cov * np.array(cov)
print(cov,cov)
```

```
[1 0.6197534e+09 1.3440364e+09 4.3245054e+09 1.3763174e+09
 1.3440364e+09 1.0000000e+00 4.4124594e+09 1.2011667e+09
 4.3245054e+09 4.4124594e+09 1.0000000e+00 2.0012756e+09
 1.3763174e+09 1.2011667e+09 2.0012756e+09 1.0000000e+00] [1.6197534e+09 1.3440364e+09 4.3245054e+09 1.3763174e+09 1.3440364e+09
 1.0000000e+00 4.4124594e+09 1.2011667e+09 4.3245054e+09 4.4124594e+09
 1.0000000e+00 2.0012756e+09 1.3763174e+09 1.2011667e+09 2.0012756e+09
 1.0000000e+00]
```

با استفاده از کتابخانه numpy و تابع cov میانگین واریانس را برای ماتریس ماست محاسبه کردیم

تمرین 5

```
# 5.std dev
print("std dev")
avg_var = np.array(avg_cov)
dev = list()
for i in range(0,4):
    dev.append(avg_var[i] - avg_var[1])
centered = pd.DataFrame(dev)
centered.rename(index={0:'mass', 1:'width', 2:'height', 3:'color_score'}, inplace=True)
centered.head()
#mse centered is a metric that is centered
```

```
Boston consulting
(1243 - centered [0]) ** 2 + (1243 - centered [1]) ** 2 + (1243 - centered [2]) ** 2 + (1243 - centered [3]) ** 2
(1243 - 1.6197534e+09) ** 2 + (1243 - 1.3440364e+09) ** 2 + (1243 - 4.3245054e+09) ** 2 + (1243 - 1.3763174e+09) ** 2
(1243 - 1.6197534e+09) ** 2 + (1243 - 1.3440364e+09) ** 2 + (1243 - 4.3245054e+09) ** 2 + (1243 - 1.3763174e+09) ** 2
```

	mass	width	height	color_score
0	2.6197534E+18	1.8067264E+18	18.7040644	1.8943127
1	1.8067264E+18	1.0000000E+00	19.4688164	1.4418125
2	18.7040644	19.4688164	1.0000000E+00	4.0051025
3	1.8943127	1.4418125	4.0051025	1.0000000E+00

در این مرحله محیوریم یک آرایه ایجاد کنیم که هر خانه از این آرایه مقادیر یک ستون از دیتاست ما را در خود جای داده است

چون با در مرحله دو ترتیب مقادیر ما به پروپرتی خاصی به خانه مشخص و جدید شده (یا از نام هیچ ها به خانه مشخص در هر مقدار یک ستون از دیتاست ما در هر هیچ به نام هیچ و در هر هیچ به پروپرتی خاصی محاسبه کردیم) پس در این مرحله ما از یک نام هیچ ساخته میشود که برای هیچ ها و به شکل دیکشنری که در هیچ و در یک آرایه جای میگیرد و در یک آرایه جای میگیرد و در یک آرایه جای میگیرد

میدهیم

مثل برای تست کردن ماتلبی و نام هیچ هاتون به ویژگی ها و آرایه ایجاد شده ما از کتابخانه numpy و از کتابخانه numpy و پیمپی و کتابخانه numpy و پیمپی و کتابخانه numpy و پیمپی و