Проект

Перед началом выполнения проекта, необходимо изменить имя вашего ноутбука/ПК на вашу фамилию на английском языке (наприм
обязательно выполнить код в ячейках ниже (их 4) (*в случае не выполнения, будут вычиляться 5 баллов из проекта*). В случае е
установлен нужно установить.

В [7]:

```
import os
os.getlogin()
```

Out[7]:

```
'Никита'
```

В [2]:

```
import socket
socket.gethostbyname(socket.gethostname())
```

Out[2]:

```
'26.225.141.103'
```

В [4]:

```
!whoami
```

В [5]:

```
from datetime import datetime

current_time = datetime.now()
print(current_time)
print("Demidovich N.M") # написать здесь свою фамилию и инициалы
```

```
2022-12-24 19:07:37.673293
Demidovich N.M
```

Скачайте таблицы в формате .csv из pgadmin и далее считайте данные таблицы.

Предобработка данных и их изучение.

1. Выведите по 5 строк с каждой таблицы.
2. Введите информацию о каждой таблицы и изучите их (возможно есть какие-то странности. Опишите полученные данные.
3. Проверьте данные на пропуски и дубликаты.
4. Вычислите сводную (описательную) статистику о данных датафреймов (таблиц) и выведите ее.
5. Если в некоторых столбцах нужно изменить данные, измените их и аргументируйте зачем их стоит изменить (например, дата дол
datetime64, а не object).

Задания:

1. Найдите все параметры ПК, имеющие 8x или 40x CD и цену более 600. Отсортируйте по скорости и цене.
2. Для каждого производителя, выпускающего лаптоты с объёмом жесткого диска не менее 10 Гбайт и ОЗУ не менее 64 мб, найти с
Выведите производителей и скорость. *Нарисуйте график зависимости скоростей от полученных моделей ноутбуков.*
3. Найдите номера моделей, тип и цены всех ноутбуков производителя A. Отсортируйте по убыванию цены. Постройте гистограмму
4. Найдите производителя, номер модели и цену среди ноутбуков с наибольшей стоимостью до 1000; *Нарисуйте график зависимо
производителей ноутбуков.*
5. Найдите для каждой модели ПК их количество и максимальное и минимальное ram, сгруппируйте по моделям; переименуйте кол
“max/min_ram”;
6. Проверьте гипотезу: «Самые дорогие ноутбуки у производителя A». Опишите полученный результат.
7. Постройте матрицы корреляции для всех таблиц. Необязательно, но, если будет желание, нарисовать график тепловой карты ма
используя функцию heatmap из библиотеки seaborn.
8. Нарисуйте график (*не графики, на одном графики должно отображаться всё*) зависимости цены ноутбука/ПК от объёма жестко
наблюдения, существует ли какая-то зависимость и т.п.
9. Найдите: a. количество товаров каждого типа у каждого производителя; постройте график pie, на должно отображаться доля каж
самый дорогой товар каждого типа, вывести тип и цену; c. производителей, делающих ноутбуки и пк ценой более 600 долларов,
принтеры, вывести производителя.
10. Выведите новую цену каждого ноутбука и ПК получив её как модель+цена+ram. Дайте колонке название ‘strange_sum’ ;
11. Найти производителей, делающих ноутбуки и ПК, но не принтеры;
12. Найдите производителя ПК и модель, чья цена ниже средней цены ноутбука, а ram и скорость больше в 1.5 и 1.2 раза соответств
13. Написать общий вывод о полученных результатах (какие важные закономерности были вами обнаружены или получены и т.п.)

В [5]:

```
import pandas as pd
# считайте данные таблицы
data = pd.read_csv('laptop.csv')
```

1. Выведите по 5 строк с каждой таблицы.

В [4]:

```
# вывод первых 5 строк фрейма
data.head()
```

Out[4]:

	code	model	speed	ram	hd	price	screen
0	1	1298	350	32	4	700.0	11
1	2	1321	500	64	8	970.0	12
2	3	1750	750	128	12	1200.0	14
3	4	1298	600	64	10	1050.0	15
4	5	1752	750	128	10	1150.0	14

2. Введите информацию о каждой таблицы и изучите их (возможно есть какие-то странности. Опишите полученные данные.

В [10]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   code    6 non-null      int64  
 1   model   6 non-null      int64  
 2   speed   6 non-null      int64  
 3   ram     6 non-null      int64  
 4   hd      6 non-null      int64  
 5   price   6 non-null      float64
 6   screen  6 non-null      int64  
dtypes: float64(1), int64(6)
memory usage: 464.0 bytes
```

3. Проверьте данные на пропуски и дубликаты.
4. Вычислите сводную (описательную) статистику о данных датафреймов (таблиц) и выведите ее.
5. Если в некоторых столбцах нужно изменить данные, измените их и аргументируйте зачем их стоит изменить (например, дата дол
   datetime64, а не object).

В [11]:

```
# проверка на дубли
data.duplicated().sum()
```

Out[11]:

```
0
```

3. Проверьте данные на пропуски и дубликаты.

В [12]:

```
# проверка на пропуски
data.isna().sum()
```

Out[12]:

```
code      0
model     0
speed     0
ram       0
hd        0
price     0
screen    0
dtype: int64
```

Предобработка данных и их изучение:

Задание №1: Выведите по 5 строк с каждой таблицы

B [6]:

```
laptop = pd.read_csv('laptop.csv')
laptop.head()
```

Out[6]:

	code	model	speed	ram	hd	price	screen
0	1	1298	350	32	4	700.0	11
1	2	1321	500	64	8	970.0	12
2	3	1750	750	128	12	1200.0	14
3	4	1298	600	64	10	1050.0	15
4	5	1752	750	128	10	1150.0	14

B [7]:

```
pc = pd.read_csv('pc.csv')
pc.head()
```

Out[7]:

	code	model	speed	ram	hd	cd	price
0	1	1232	500	64	5	12x	600.0
1	10	1260	500	32	10	12x	350.0
2	11	1233	900	128	40	40x	980.0
3	12	1233	800	128	20	50x	970.0
4	2	1121	750	128	14	40x	850.0

B [8]:

```
printer = pd.read_csv('printer.csv')
printer.head()
```

Out[8]:

	code	model	color	type	price
0	1	1276	n	Laser	400.0
1	2	1433	y	Jet	270.0
2	3	1434	y	Jet	290.0
3	4	1401	n	Matrix	150.0
4	5	1408	n	Matrix	270.0

B [9]:

```
product = pd.read_csv('product.csv')
product.head()
```

Out[9]:

	maker	model	type
0	A	1232	PC
1	A	1233	PC
2	A	1276	Printer
3	A	1298	Laptop
4	A	1401	Printer

Задание №2: Введите информацию о каждой таблицы и изучите их (возможно есть какие-то странности. Опишите полученны

В [18]:

```
1 laptop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   code    6 non-null      int64
 1   model   6 non-null      int64
 2   speed   6 non-null      int64
 3   ram     6 non-null      int64
 4   hd      6 non-null      int64
 5   price   6 non-null      float64
 6   screen  6 non-null      int64
dtypes: float64(1), int64(6)
memory usage: 464.0 bytes
```

В [19]:

```
1 pc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   code    12 non-null     int64
 1   model   12 non-null     int64
 2   speed   12 non-null     int64
 3   ram     12 non-null     int64
 4   hd      12 non-null     int64
 5   cd      12 non-null     object
 6   price   12 non-null     float64
dtypes: float64(1), int64(5), object(1)
memory usage: 800.0+ bytes
```

В [20]:

```
1 printer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   code    6 non-null      int64
 1   model   6 non-null      int64
 2   color   6 non-null      object
 3   type    6 non-null      object
 4   price   6 non-null      float64
dtypes: float64(1), int64(2), object(2)
memory usage: 368.0+ bytes
```

В [21]:

```
1 product.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   maker   16 non-null     object
 1   model   16 non-null     int64
 2   type    16 non-null     object
dtypes: int64(1), object(2)
memory usage: 512.0+ bytes
```

В таблице laptop - 6 записей, в таблице pc - 12, в таблице printer - 6, в таблице product, содержащей информацию о всей технике из в количество - 16, в то время как если сложить количество всех записей в 3-х предыдущих таблицах (6+12+6), получится 24. Это связа модели имеют одинаковые характеристики

Задание 3: Проверьте данные на пропуски и дубликаты

В [10]:

```
duplicated = laptop.duplicated().sum()
isna = laptop.isna().sum()
print(duplicated)
print(isna)
```

```
0
code      0
model     0
speed     0
ram       0
hd        0
price     0
screen    0
dtype: int64
```

В [9]:

```
duplicated = pc.duplicated().sum()
isna = pc.isna().sum()
print(duplicated)
print(isna)
```

```
0
code     0
model    0
speed    0
ram      0
hd       0
cd       0
price    0
dtype: int64
```

В [31]:

```
duplicated = printer.duplicated().sum()
isna = printer.isna().sum()
print(duplicated)
print(isna)
```

```
0
code     0
model    0
color    0
type     0
price    0
dtype: int64
```

В [32]:

```
duplicated = product.duplicated().sum()
isna = product.isna().sum()
print(duplicated)
print(isna)
```

```
0
maker    0
model    0
type     0
dtype: int64
```

Пропуски и дубликаты во всех таблицах отсутствуют

Задание 4: Вычислите сводную (описательную) статистику о данных датафреймов (таблиц) и выведите ее

В [45]:

```
laptop.describe()
```

Out[45]:

	code	model	speed	ram	hd	price	screen
count	6.000000	6.000000	6.000000	6.000000	6.00000	6.000000	6.000000
mean	3.500000	1452.833333	566.666667	80.000000	9.00000	1003.333333	13.000000
std	1.870829	231.131492	163.299316	39.191836	2.75681	177.951304	1.549193
min	1.000000	1298.000000	350.000000	32.000000	4.00000	700.000000	11.000000
25%	2.250000	1298.000000	462.500000	64.000000	8.50000	955.000000	12.000000
50%	3.500000	1309.500000	550.000000	64.000000	10.00000	1010.000000	13.000000
75%	4.750000	1642.750000	712.500000	112.000000	10.00000	1125.000000	14.000000
max	6.000000	1752.000000	750.000000	128.000000	12.00000	1200.000000	15.000000

В [46]:

```
1 pc.describe()
```

Out[46]:

	code	model	speed	ram	hd	price
count	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
mean	6.500000	1206.916667	608.333333	88.000000	13.666667	675.000000
std	3.605551	52.397880	153.494645	43.417634	9.670323	261.342687
min	1.000000	1121.000000	450.000000	32.000000	5.000000	350.000000
25%	3.750000	1204.250000	500.000000	56.000000	8.000000	387.500000
50%	6.500000	1232.000000	550.000000	96.000000	10.000000	725.000000
75%	9.250000	1233.000000	750.000000	128.000000	15.500000	875.000000
max	12.000000	1260.000000	900.000000	128.000000	40.000000	980.000000

В [47]:

```
1 printer.describe()
```

Out[47]:

	code	model	price
count	6.000000	6.000000	6.000000
mean	3.500000	1373.333333	296.666667
std	1.870829	72.060160	94.162979
min	1.000000	1276.000000	150.000000
25%	2.250000	1316.250000	270.000000
50%	3.500000	1404.500000	280.000000
75%	4.750000	1426.750000	372.500000
max	6.000000	1434.000000	400.000000

В [48]:

```
1 product.describe()
```

Out[48]:

	model
count	16.000000
mean	1464.500000
std	305.022622
min	1121.000000
25%	1272.000000
50%	1361.000000
75%	1513.000000
max	2113.000000

**Задание №5: Если в некоторых столбцах нужно изменить данные, измените их и аргументируйте зачем их стоит изменить (н
иметь тип данных datetime64, а не object)**

В [125]:

```
1 laptop["model"] = laptop["model"].astype(object)
2 pc["model"] = pc["model"].astype(object)
3 printer["model"] = printer["model"].astype(object)
4 product["model"] = product["model"].astype(object)
```

Наименования моделей должны иметь тип object, а не int64. При желании так же можно изменить тип данных в столбцах "price" с floa
цена каждого продукта является целочисленным числом

Задания:

Задание №1: Найдите все параметры ПК, имеющие 8x или 40x CD и цену более 600. Отсортируйте по скорости и цене

В [130]:

```
new_pc_1 = pc[pc['cd'] == '8x']
new_pc_2 = pc[pc['cd'] == '40x']
new_pc = pd.concat([new_pc_1, new_pc_2])
new_pc = new_pc.sort_values(['speed', 'price'])
new_pc
```

Out[130]:

	code	model	speed	ram	hd	cd	price
6	4	1121	600	128	14	40x	850.0
7	5	1121	600	128	8	40x	850.0
4	2	1121	750	128	14	40x	850.0
2	11	1233	900	128	40	40x	980.0

Задание №2: Для каждого производителя, выпускающего лаптоты с объёмом жесткого диска не менее 10 Гбайт и ОЗУ не ме скорости таких лаптопов. Выведите производителей и скорость. *Нарисуйте график зависимости скоростей от полученных

В [67]:

```
new_laptop_1 = laptop[laptop['hd'] >= 10]
new_laptop_2 = laptop[laptop['ram'] >= 64]
new_laptop = pd.concat([new_laptop_1, new_laptop_2])
new_laptop = pd.merge(product['maker'], new_laptop['speed'], left_index = True, right_index = True)
new_laptop
```

Out[67]:

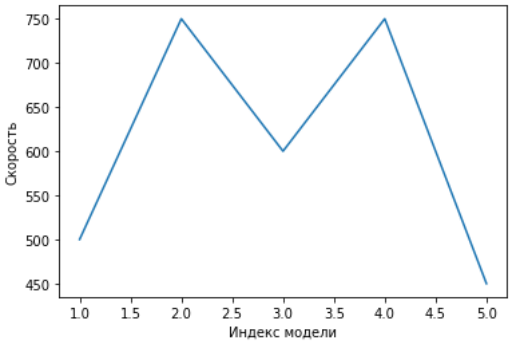
	maker	speed
1	A	500
2	A	750
2	A	750
3	A	600
3	A	600
4	A	750
4	A	750
5	A	450
5	A	450

В [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

В [65]:

```
plt.plot(new_laptop['speed'])
plt.ylabel("Скорость")
plt.xlabel("Индекс модели")
plt.locator_params(axis = 'x', nbins = 10)
plt.show()
```



Задание №3: Найдите номера моделей, тип и цены всех ноутбуков производителя А. Отсортируйте по убыванию цены. Пост изменения цены

В [129]:

```
new_laptop = pd.merge(laptop, product, how = 'inner')
new_laptop = product[product['maker'] == 'A']
new_laptop = new_lap[['maker', 'model', 'type', 'price']]
new_laptop = new_lap.sort_values('price', ascending = 0)
new_laptop
```

Out[129]:

	model	type	price	maker
5	1752	Laptop	1150.0	A
1	1298	Laptop	1050.0	A
2	1298	Laptop	950.0	A
0	1298	Laptop	700.0	A

Задание №4: Найдите производителя, номер модели и цену среди ноутбуков с наибольшей стоимостью до 1000; *Нарисуйте цен от всех производителей ноутбуков

В [18]:

```
new_laptop = pd.merge(laptop, product, how = 'inner')
new_laptop = new_laptop[new_laptop['price'] < 1000]
new_laptop = new_laptop[new_laptop['price'] == max(new_laptop['price'])]
new_laptop = new_laptop[['maker', 'model', 'price']]
new_laptop
```

Out[18]:

	maker	model	price
3	C	1321	970.0

В [120]:

```
#боже как...
```

Задание №5: Найдите для каждой модели ПК их количество и максимальное и минимальное ram, сгруппируйте по моделям; макс. и мин. в “max/min_ram”

В [123]:

```
new_pc = pd.merge(pc, product, how = 'inner')
new_pc = new_pc[new_pc['ram'] == max(new_pc['ram'])]
```

В [122]:

```
#боже как...
```

Задание №6: Проверьте гипотезу: «Самые дорогие ноутбуки у производителя А». Опишите полученный результат

В [13]:

```
new_laptop = pd.merge(laptop, product, how = 'inner')
pd.pivot_table(new_laptop, index=['maker'], values = ['price'])
```

Out[13]:

maker	price
A	962.5
B	1200.0
C	970.0

Как мы видим из этой таблицы, самые дорогие ноутбуки у производителя B

Задание №7: Постройте матрицы корреляции для всех таблиц. Необязательно, но, если будет желание, нарисовать график т корреляции используя функцию heatmap из библиотеки seaborn

В [121]:

```
#боже как...
```

Задание №8: Нарисуйте график (*не графики, на одном графики должно отображаться всё*) зависимости цены ноутбука/ПК диска. Опишите ваши наблюдения, существует ли какая-то зависимость и.т.п

В [72]:

```
pc = pc[['hd', 'price']].sort_values('hd')
laptop = laptop[['hd', 'price']].sort_values('hd')
```

В [54]:

```
pc #таблица зависимости цены от объёма жесткого диска ПК
```

Out[54]:

	hd	price
0	5	600.0
5	5	600.0
7	8	850.0
10	8	350.0
1	10	350.0
9	10	400.0
11	10	350.0
4	14	850.0
6	14	850.0
3	20	970.0
8	20	950.0
2	40	980.0

В [55]:

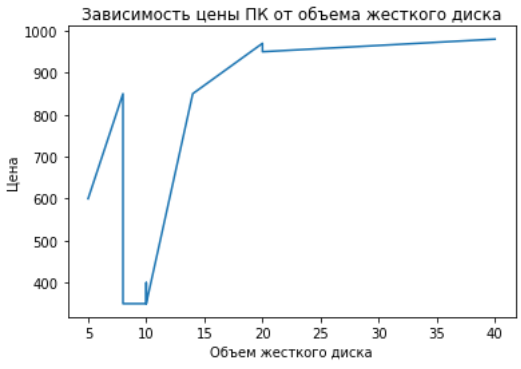
```
laptop #таблица зависимости цены от объёма жесткого диска ноутбука
```

Out[55]:

	hd	price
0	4	700.0
1	8	970.0
3	10	1050.0
4	10	1150.0
5	10	950.0
2	12	1200.0

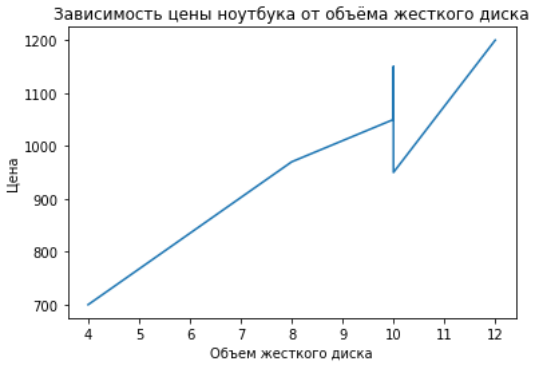
В [73]:

```
plt.plot(pc['hd'], pc['price'])
plt.title("Зависимость цены ПК от объема жесткого диска")
plt.xlabel("Объем жесткого диска")
plt.ylabel("Цена")
plt.show()
```



In [74]:

```
plt.plot(laptop['hd'], laptop['price'])
plt.title("Зависимость цены ноутбука от объёма жесткого диска")
plt.xlabel("Объем жесткого диска")
plt.ylabel("Цена")
plt.show()
```



Как мы видим из графиков, как правило, цена возрастает с увеличением объёма жесткого диска

Задание №9: Найдите:
a. количество товаров каждого типа у каждого производителя; постройте график pie, на должно отображаться доля каждого
b. самый дорогой товар каждого типа, вывести тип и цену;
c. производителей, делающих ноутбуки и пк ценой более 600 долларов, но которые не производят принтеры, вывести прои

In [123]:

```
#боже как...
```

Задание №10: Выведите новую цену каждого ноутбука и ПК получив её как модель+цена+ram. Дайте колонке название 'stran

In [36]:

```
new_table = pd.concat([pc, laptop])
new_table = new_table[['model', 'price', 'ram']]

strange_sum = []
for i in range(len(new_table)):
    strange_sum.append(new_table.iloc[i]['model'] + new_table.iloc[i]['price'] + new_table.iloc[i]['ram'])
new_table['strange_sum'] = strange_sum

new_table
```

Out[36]:

	model	price	ram	strange_sum
0	1232	600.0	64	1896.0
1	1260	350.0	32	1642.0
2	1233	980.0	128	2341.0
3	1233	970.0	128	2331.0
4	1121	850.0	128	2099.0
5	1233	600.0	64	1897.0
6	1121	850.0	128	2099.0
7	1121	850.0	128	2099.0
8	1233	950.0	128	2311.0
9	1232	400.0	32	1664.0
10	1232	350.0	64	1646.0
11	1232	350.0	32	1614.0
0	1298	700.0	32	2030.0
1	1321	970.0	64	2355.0
2	1750	1200.0	128	3078.0
3	1298	1050.0	64	2412.0
4	1752	1150.0	128	3030.0
5	1298	950.0	64	2312.0

Задание №11: Найти производителей, делающих ноутбуки и ПК, но не принтеры

В [10]:

```
new_product = product[product['type'] != 'Printer']
new_product
#а как дальше то блять
```

Out[10]:

	maker	model	type
0	A	1232	PC
1	A	1233	PC
3	A	1298	Laptop
6	A	1752	Laptop
7	B	1121	PC
8	B	1750	Laptop
9	C	1321	Laptop
12	E	1260	PC
14	E	2112	PC
15	E	2113	PC

В [ ]:

```

```