МИНИ-ПРОЕКТ ПО АНАЛИЗУ ДАННЫХ ПО ПРОДАЖАМ.

ЦЕЛИ:

Собрать все данные из папки data в один DataFrame, имеющий следующие столбцы: колонки из самих файлов (product_id, quantity), а также имя пользователя (name), и дата этих покупок (date, соответствует названию папки, где лежит папка с пользователем)

1. Выяснить, какой пользователь купил больше всего товаров. Если их несколько, то перечислить имена через запятую с пробелом и в алфавитном порядке.
2. Найти топ-10 товаров по числу проданных единиц за всё время и построить barplot.
3. Визуализировать продажи по дням.
4. Сколько пользователей приобрели какой-либо товар повторно (более 1 раза)? Повтором будем считать покупку товара с одинаковым product_id, совершенную в разные дни.

ОПИСАНИЕ:

Данные имеют следующую структуру:
- записываются для каждого пользователя, совершившего покупки, каждый день
- для каждой даты есть своя папка, внутри неё – папки для каждого пользователя
- внутри каждой папки есть файл data.csv, где и хранятся данные

АНАЛИЗ ДАННЫХ ПО ПРОДАЖАМ с 2020-12-03 по 2020-12-09

1. Выяснить, какой пользователь купил больше всего товаров. Если их несколько, то перечислить имена через запятую с пробелом и в алфавитном порядке.

Ответ: **Alexey_Smirnov, Petr_Smirnov**

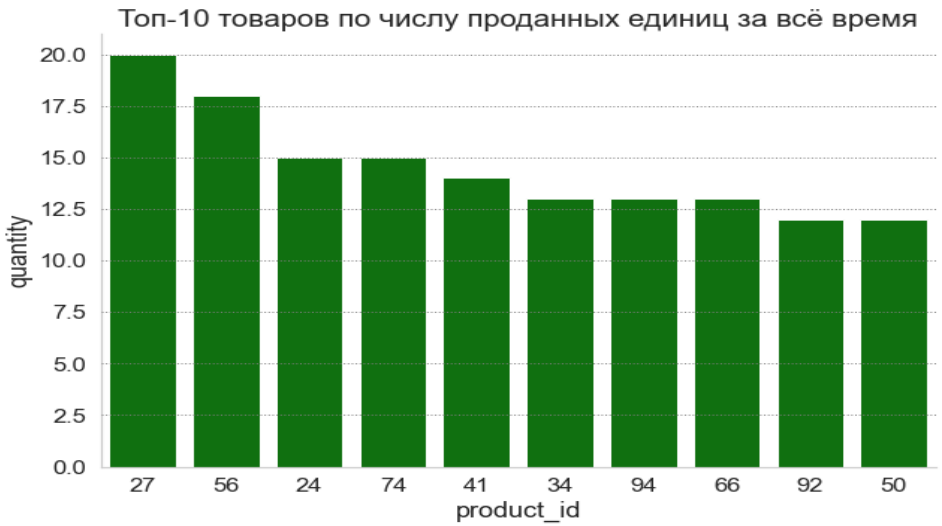
4. Сколько пользователей приобрели какой-либо товар повторно (более 1 раза)? Повтором будем считать покупку товара с одинаковым product_id, совершенную в разные дни.

Ответ:

name	product_id	amount
Anton_Ivanov	15	2
Petr_Fedorov	94	2

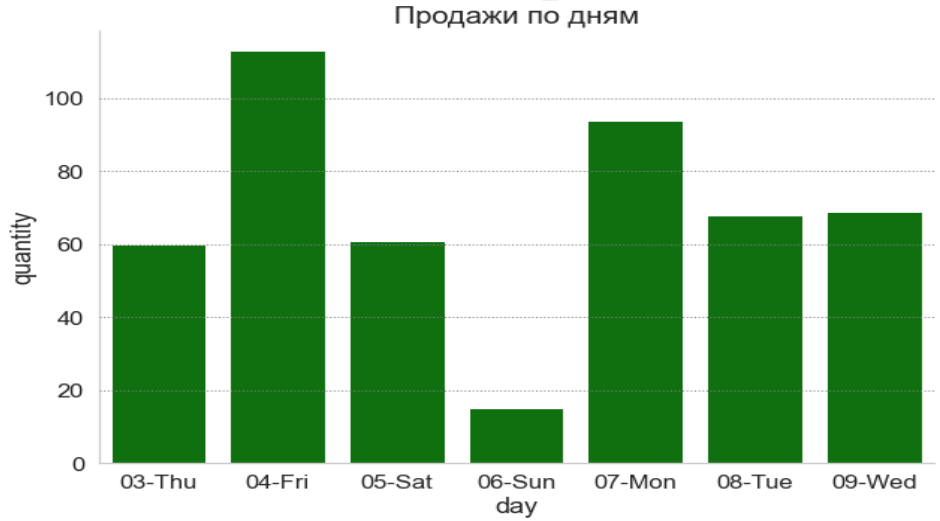
2. Найти топ-10 товаров по числу проданных единиц за всё время и построить barplot.
3. Визуализировать продажи по дням.

Ответы:



Топ – 10 товаров по числу проданных единиц за всё время.

product_id	quantity
27	20
56	18
24	15
74	15
41	14
34	13
94	13
66	13
92	12
50	12



Продажи по дням.

day	quantity
03-Thu	60
04-Fri	113
05-Sat	61
06-Sun	15
07-Mon	94
08-Tue	68
09-Wed	69

Ниже вставлена копия скрипта Phyton из файла “mini_project.ipynb” (jupyter notebook) который был использован для поиска решений проекта.

Анализ проекта

Сбор данных из всех каталогов в один DataFrame

In [1]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Create an empty DataFrame to add read data from 'data.csv' files to it
df = pd.DataFrame()
# path relative path to data folder
path = r".\data"

# Walk all branches of the folder tree to search for files 'data.csv'
for root, dirs, files in os.walk(path):
    # check file name match 'data.csv'
    for name_fls in files:
        # if substring name_fls is not found, -1 is returned
        if not name_fls.find('data.csv') == -1 :
            # collect the path
            path_csv = f'{root}\\{name_fls}'
            # read 'data.csv'
            data_csv = pd.read_csv(path_csv, usecols=[1, 2])
            lst = path_csv.split("\\")
            # add new columns from parsed path 'lst'
            data_csv['name'] = lst[-2]
            data_csv['date'] = pd.to_datetime(lst[-3])
            # collect all DataFrame
            df = pd.concat([df, data_csv])
```

Пользователь который купил больше всего товаров

In [2]:
```
# calculate the maximum value
max_byu = df.groupby('name', as_index=False) \
            .agg({'quantity':'sum'})['quantity']\
```

```
            .max()
# compare with the maximum value
name_max = df.groupby('name', as_index=False) \
             .agg({'quantity':'sum'}) \
             .rename(columns={'quantity':'total'})\
             .query('total == @max_byu')
# formation of a string with names
', '.join([str(i) for i in name_max['name'].sort_values()])
```

Out[2]:
```
'Alexey_Smirnov, Petr_Smirnov'
```

Топ-10 товаров по числу проданных единиц за всё время

In [3]:
```
product_top10 = df.groupby('product_id', as_index=False) \
                  .agg({'quantity':'sum'}) \
                  .sort_values('quantity', ascending=False) \
                  .head(10)
product_top10
```

Out[3]:

	product_id	quantity
22	27	20
42	56	18
21	24	15
51	74	15
32	41	14
27	34	13
66	94	13
46	66	13
65	92	12
38	50	12

Объем продаж по дням

In [4]:
```
df['day'] = df['date'].dt.strftime('%d-%a')
sales_day = df.groupby(['day'], as_index=False)\
                .agg({'quantity':'sum'})
sales_day
```

Out[4]:

	day	quantity
0	03-Thu	60
1	04-Fri	113
2	05-Sat	61
3	06-Sun	15
4	07-Mon	94
5	08-Tue	68
6	09-Wed	69

Пользователи которые совершили повторные покупки в разные дни

In [5]:
```
repeat_purchases = df.drop_duplicates(['name', 'product_id','date'])\
                     .groupby(['name', 'product_id'], as_index=False)\
                     .agg({'date':'count'})\
                     .query('date > 1')\
                     .rename(columns={'date':'amount'})

repeat_purchases
```

Out[5]:

	name	product_id	amount
37	Anton_Ivanov	15	2
92	Petr_Fedorov	94	2

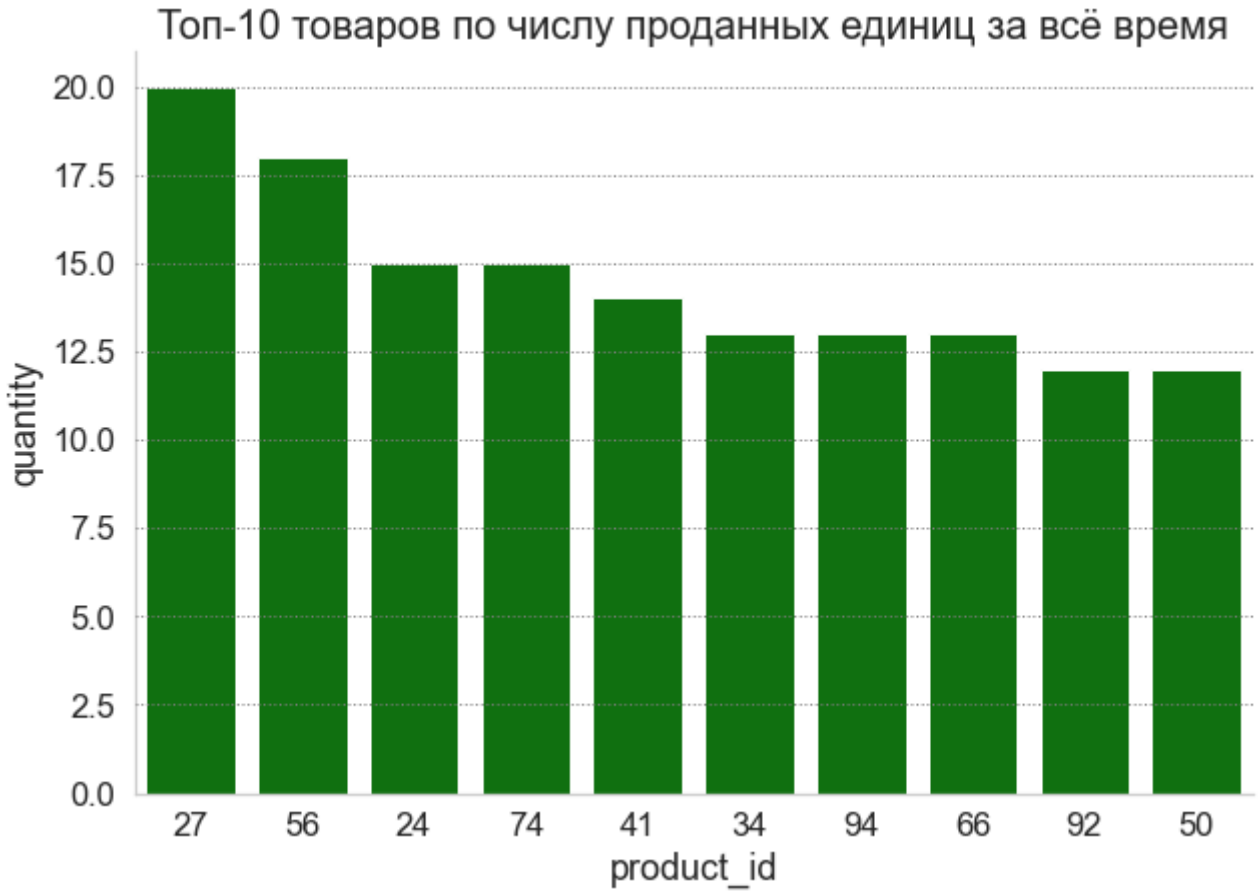
Визуализация

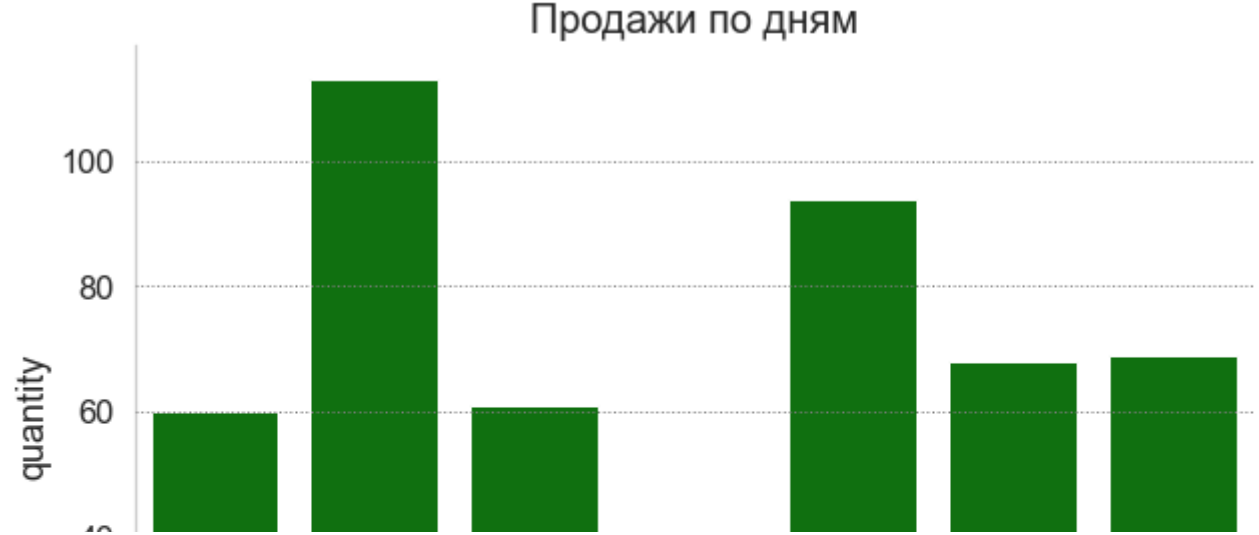
In [6]:
```
# creating a style for charts
sns.set_style("whitegrid",
```

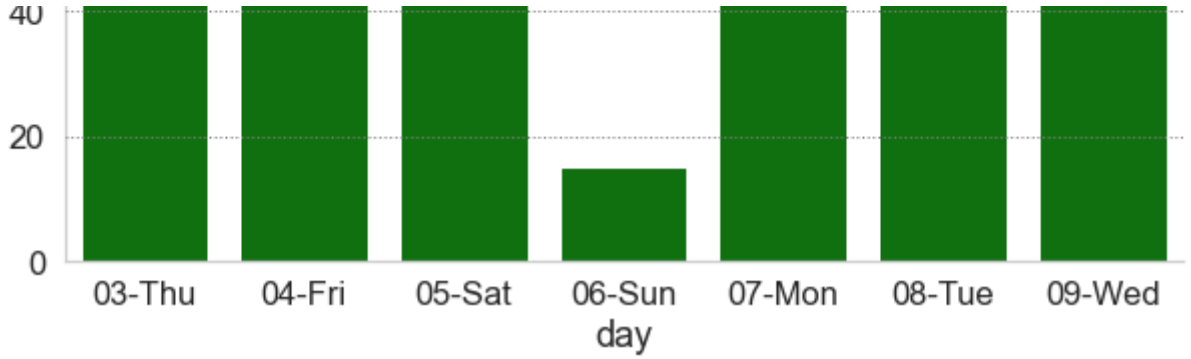
```
              {'axes.grid':True,
               'grid.color':'grey',
               'grid.linestyle':':',
               'axes.axisbelow':False,
               'axes.spines.top':False,
               'axes.spines.right':False,
              })
sns.set_context("notebook", font_scale=1.5,
                rc={'axes.titlesize':20.0})
# construction barplot
plt.figure(figsize=(10, 15))
plt.subplot(2, 1, 1)
top10 = sns.barplot(x='product_id',
                    y='quantity',
                    color='green',
                    order=product_top10['product_id'],
                    data=product_top10)
top10.set_title("Топ-10 товаров по числу проданных единиц за всё время")

plt.subplot(2, 1, 2)
sales = sns.barplot(x='day',
                        y='quantity',
                        color='green',
                        data=sales_day)
sales.set_title("Продажи по дням")
```

Out[6]:
```
Text(0.5, 1.0, 'Продажи по дням')
```


Топ-10 товаров по числу проданных единиц за всё время
quantity
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
27
56
24
74
41
34
94
66
92
50
product_id


Продажи по дням
quantity
100
80
60



Выгрузка данных в excel

In [11]:
```
# writer 'df', 'name_max', 'product_top10', 'sales_day', 'repeat_purchases' to "output.xlsx"
with pd.ExcelWriter("output.xlsx") as writer:
    df.to_excel(writer, sheet_name="data", index=False)
    name_max.to_excel(writer, sheet_name="name_max", index=False)
    product_top10.to_excel(writer, sheet_name="product_top10", index=False)
    sales_day.to_excel(writer, sheet_name="sales_day", index=False)
    repeat_purchases.to_excel(writer, sheet_name="repeat_purchases", index=False)
```