

In [312...

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

sns.set(style='whitegrid', font_scale=1.3, palette='Set2')
%matplotlib inline

import requests
from time import sleep
import json
from statistics import mean
import plotly
import plotly.graph_objects as go
import plotly.express as px
import scipy.stats as sps
import warnings
warnings.filterwarnings('ignore')
plotly.offline.init_notebook_mode()
```

## Analysis of diversification effects, 1

В данной работе я сравнила ETF IGM и несколько акций ведущих технических компаний.

Обоснование выбора инструментов (компаний, с акциями которых работала): Adobe разрабатывает продукты для дизайнеров, фотографов, видеографов. Так как пандемия поддерживала и даже увеличивала спрос на онлайн-контент, дела у компании шли хорошо. Также мне было известно, что за последние 5 лет акции компании росли.

Почему PayPal? Компания была интересна из-за развития криптовалюты. Совсем недавно был максимум прибыли, люди продолжают многое покупать онлайн.

Microsoft — так как это просто очень известная, стабильно растущая компания

In [269...

```
prof_IGM = pd.DataFrame(pd.read_csv('IGM_prof.csv'))
prof_Adobe = pd.DataFrame(pd.read_csv('ADBE_prof.csv'))
prof_Microsoft = pd.DataFrame(pd.read_csv('MSFT_prof.csv'))
prof_PayPal = pd.DataFrame(pd.read_csv('PYPL_prof.csv'))
```

Диверсификация — это в общем случае стратегия управления рисками, которая объединяет в портфель разные инвестиции. Все это делается для ограничения подверженности риску отдельного актива. Обоснование этого метода заключается в том, что портфель, состоящий из различных видов активов, в среднем будет приносить более высокую долгосрочную доходность и снизить риск любого отдельного холдинга или ценной бумаги.

In [270...

```
prof_IGM = prof_IGM.loc[3:4]
for index in prof_IGM.index:
    for col in prof_IGM.columns:
        if col != 'IGM':
            prof_IGM.loc[index, col] = float(prof_IGM.loc[index, col].replace(
prof_IGM
```

Out[270...

	IGM	2016	2017	2018	2019	2020
3	Sum profit	0.151489	0.574965	0.610978	1.271361	2.270652
4	Annual profit	0.151489	0.287483	0.203659	0.31784	0.45413

Рассчитали прибыль в год с помощью данных, с сайта "yahoo" по формуле  $\frac{(D+P_C-P_O)}{P_O}$ , где  $D$  - это дивиденды,  $P_O$  и  $P_C$  цена покупки и продажи акций соответственно.

In [271...

```
prof_IGM = prof_IGM.transpose()
prof_IGM.columns = ['Sum profit', 'Annual profit']
prof_IGM = prof_IGM.drop(index = 'IGM')
prof_IGM
```

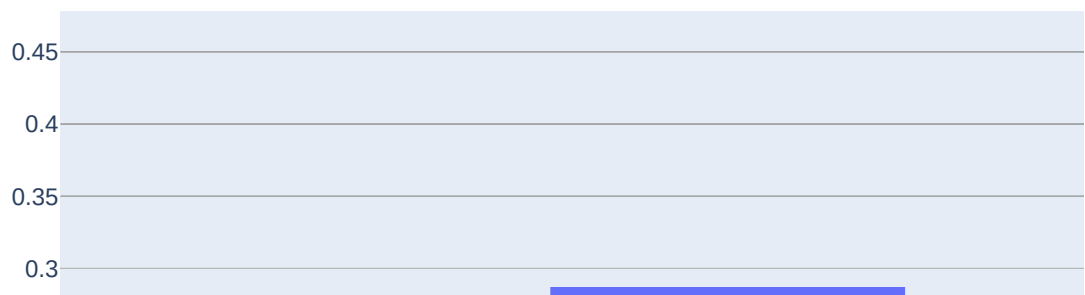
Out[271...

	Sum profit	Annual profit
2016	0.151489	0.151489
2017	0.574965	0.287483
2018	0.610978	0.203659
2019	1.271361	0.31784
2020	2.270652	0.45413

In [272...

```
fig = go.Figure(
    data=[go.Bar(x=['2016', '2017', '2018', '2019', '2020'], y=prof_IGM['Annual profit'])],
    layout_title_text="IGM profit"
)
fig.show()
```

IGM profit



Построили гистограмму зависимости прибыли от года для IGM, аналогичную процедуру проделываем для 3 выбранных компаний.

```
In [273... prof_Adobe.columns= ['Adobe', '2016', '2017', '2018', '2019', '2020']
```

```
In [274... prof_Adobe = prof_Adobe.loc[2:3]
for index in prof_Adobe.index:
    for col in prof_Adobe.columns:
        if col != 'Adobe':
            prof_Adobe.loc[index, col] = float(prof_Adobe.loc[index, col].rep
prof_Adobe
```

```
Out[274...
      Adobe      2016      2017      2018      2019      2020
2  Sum profit  0.121826  0.909556  1.465294  2.593876  4.449711
3  Annual profit 0.121826  0.454778  0.488431  0.648469  0.889942
```

```
In [275... prof_Adobe = prof_Adobe.loc[2:3]
prof_Adobe = prof_Adobe.transpose()
prof_Adobe.columns = ['Sum profit', 'Annual profit']
prof_Adobe = prof_Adobe.drop(index = 'Adobe')
```

```
In [276... fig = go.Figure(
    data=[go.Bar(x=['2016', '2017', '2018', '2019', '2020'], y=prof_Adobe['Ar
    layout_title_text="Adobe profit"
)
fig.show()
```

### Adobe profit



In [278... `prof_Microsoft.columns= ['Microsoft', '2016', '2017', '2018', '2019', '2020']`

In [279... `prof_Microsoft = prof_Microsoft.loc[3:4]`  
`for index in prof_Microsoft.index:`  
 `for col in prof_Microsoft.columns:`  
 `if col != 'Microsoft':`  
 `prof_Microsoft.loc[index, col] = float(prof_Microsoft.loc[index,`  
`prof_Microsoft`

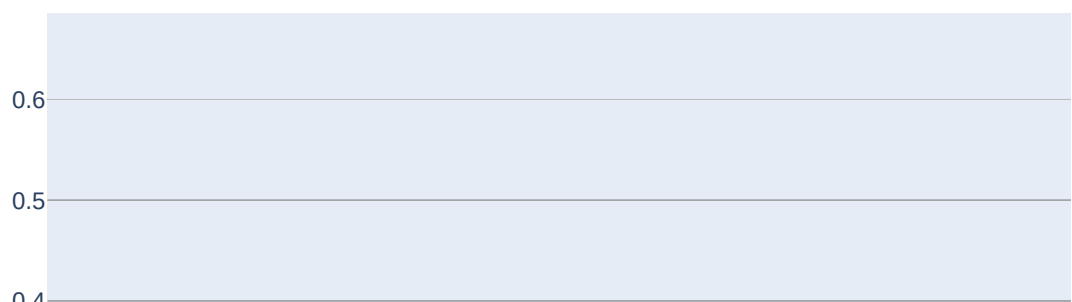
Out[279... 

	Microsoft	2016	2017	2018	2019	2020
3	Sum profit	0.171024	0.631075	0.957842	2.025957	3.255891
4	Annual profit	0.171024	0.315538	0.319281	0.506489	0.651178

In [280... `prof_Microsoft = prof_Microsoft.loc[3:4]`  
`prof_Microsoft = prof_Microsoft.transpose()`  
`prof_Microsoft.columns = ['Sum profit', 'Annual profit']`  
`prof_Microsoft = prof_Microsoft.drop(index = 'Microsoft')`

In [281... `fig = go.Figure(`  
 `data=[go.Bar(x=['2016', '2017', '2018', '2019', '2020'], y=prof_Microsoft`  
 `layout_title_text="Microsoft profit"`  
`)`  
`fig.show()`

Microsoft profit



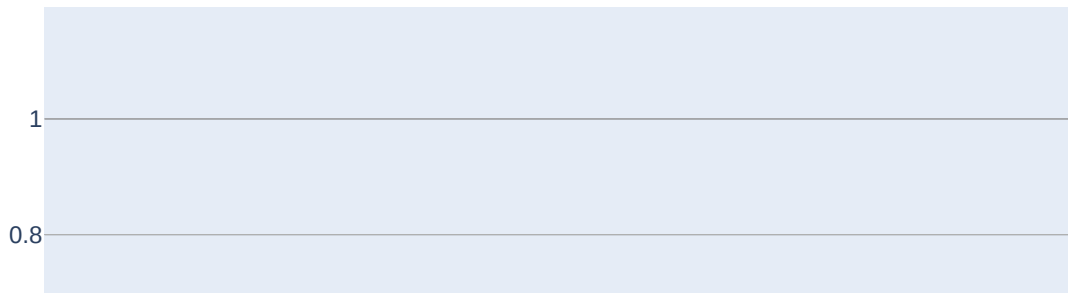
```
In [282... prof_PayPal.columns= ['PayPal', '2016', '2017', '2018', '2019', '2020']
```

```
In [283... prof_PayPal = prof_PayPal.loc[2:3]
for index in prof_PayPal.index:
    for col in prof_PayPal.columns:
        if col != 'PayPal':
            prof_PayPal.loc[index, col] = float(prof_PayPal.loc[index, col].r
```

```
In [284... prof_PayPal = prof_PayPal.loc[2:3]
prof_PayPal = prof_PayPal.transpose()
prof_PayPal.columns = ['Sum profit', 'Annual profit']
prof_PayPal = prof_PayPal.drop(index = 'PayPal')
```

```
In [285... fig = go.Figure(
    data=[go.Bar(x=['2016', '2017', '2018', '2019', '2020'], y=prof_PayPal['A
    layout_title_text="PayPal profit"
)
fig.show()
```

### PayPal profit



```
In [309...
```

prof\_IGM

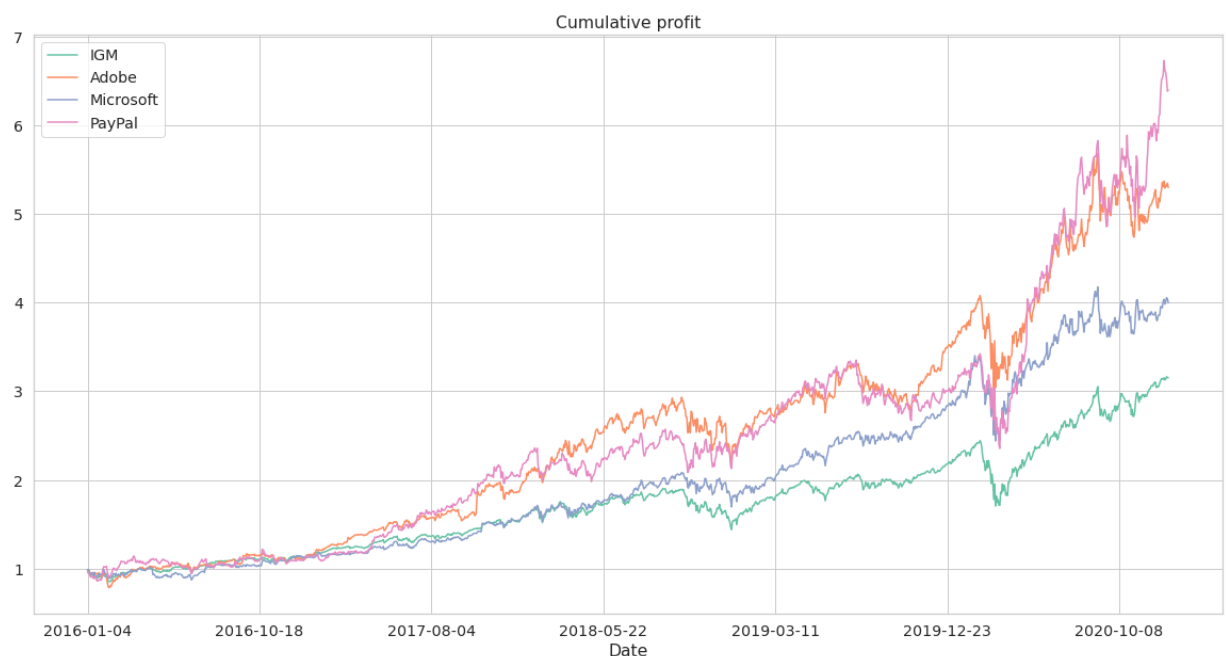
Out[309...

	Sum profit	Annual profit
<b>2016</b>	0.151489	0.151489
<b>2017</b>	0.574965	0.287483
<b>2018</b>	0.610978	0.203659
<b>2019</b>	1.271361	0.31784
<b>2020</b>	2.270652	0.45413

Еще раз зададимся вопросом дохода, но теперь рассчитаем совокупную доходность и построим график, чтобы сравнить поведение акций за анализируемые годы.

In [307...

```
cumulative_returns = (1 + stocks_df).cumprod()
cumulative_returns.plot(figsize=(20, 10), title="Cumulative profit")
plt.show()
```



Видим, что этот график еще раз наглядно показывает различия наших акций, которые мы уже замечали на отдельных гистограммах. И снова видно, что доход у Adobe и PayPal наибольший, а у IGM наименьший.

## Теперь займемся исследованием рисков для изучаемых объектов

In [294...

```
df_IGM = pd.DataFrame(pd.read_csv('IGM.csv'))
df_Adobe = pd.DataFrame(pd.read_csv('ADBE.csv'))
df_Microsoft = pd.DataFrame(pd.read_csv('MSFT.csv'))
df_PayPal = pd.DataFrame(pd.read_csv('PYPL.csv'))
```

Снова скачиваем данные за последние 5 лет о всех компаниях.

In [295...

```
df_IGM = df_IGM[['Date', 'Close']]
df_IGM = df_IGM.set_index('Date')
```

```

df_Adobe = df_Adobe[['Date', 'Close']]
df_Adobe = df_Adobe.set_index('Date')
df_Microsoft = df_Microsoft[['Date', 'Close']]
df_Microsoft = df_Microsoft.set_index('Date')
df_PayPal = df_PayPal[['Date', 'Close']]
df_PayPal = df_PayPal.set_index('Date')
stocks_df = pd.concat([df_IGM, df_Adobe, df_Microsoft, df_PayPal], axis='columns')
stocks_df.columns = ['IGM', 'Adobe', 'Microsoft', 'PayPal']
stocks_df

```

Out[295]...

	IGM	Adobe	Microsoft	PayPal
Date				
2015-12-31	110.690002	93.940002	55.480000	36.200001
2016-01-04	108.290001	91.970001	54.799999	34.750000
2016-01-05	107.980003	92.339996	55.049999	34.310001
2016-01-06	106.540001	91.019997	54.049999	33.980000
2016-01-07	103.389999	89.110001	52.169998	33.130001
...	...	...	...	...
2020-12-23	346.220001	496.910004	221.020004	239.440002
2020-12-24	347.440002	499.859985	222.750000	238.639999
2020-12-28	350.149994	498.950012	224.960007	235.729996
2020-12-29	349.179993	502.109985	224.149994	231.029999
2020-12-30	348.929993	497.450012	221.679993	231.509995

1259 rows × 4 columns

Теперь рассчитаем ежедневное процентное изменение акций и посмотрим на графике как менялось поведение акций за 5 лет:

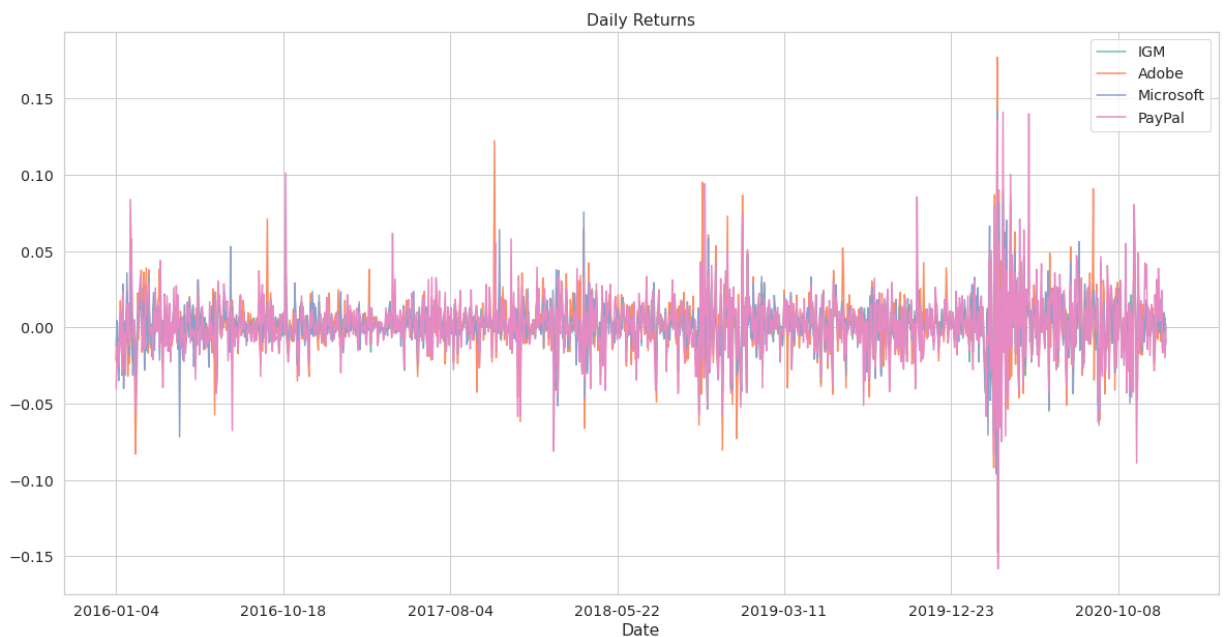
In [296]...

```

stocks_df = stocks_df.pct_change().dropna()

stocks_df.plot(figsize=(20, 10), title="Daily Returns")
plt.show()

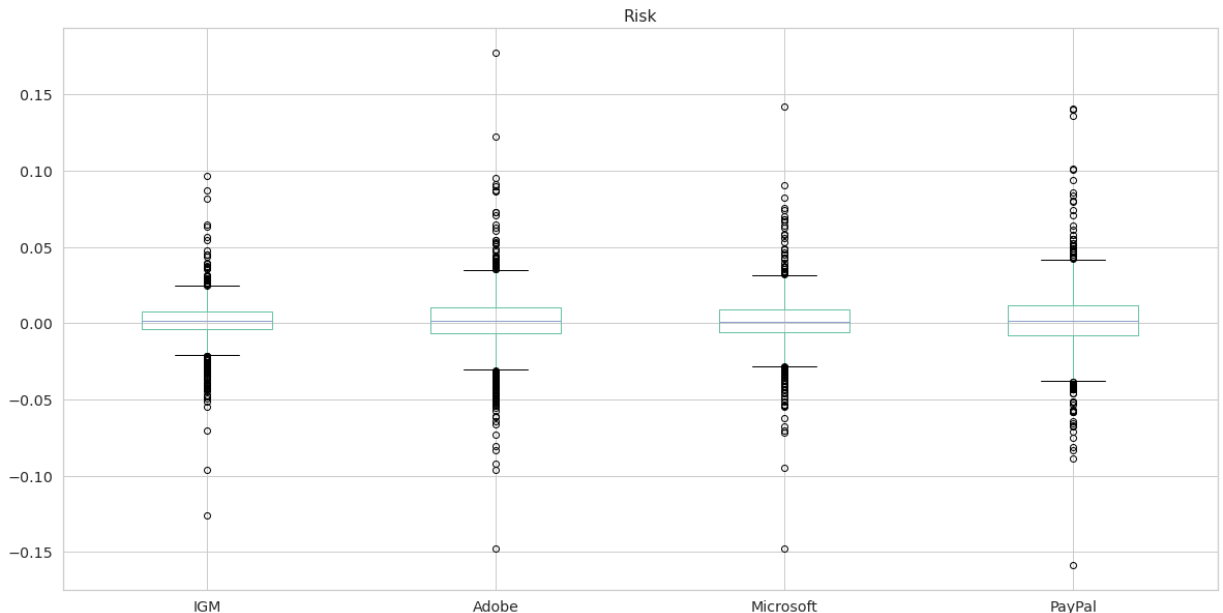
```



Из графика видно, что Adobe и PayPal являются наиболее волатильными из проанализированных акций, в то время как IGM, по-видимому, является наименее волатильной акцией. Чтобы это заметить, следует сосредоточиться на количестве пиков на графике: чем больше их количество, тем выше волатильность акции и, значит, больше связанный с ней инвестиционный риск.

In [304...

```
stocks_df.plot.box(figsize=(20, 10), title="Risk")
plt.show()
```



По графику boxplot, который как раз обычно используется для анализа средних и медианных значений без учета разбросов, видим очередное подтверждение нашей теории. Хотя в целом значения различаются весьма не сильно, но у Adobe и PayPal снова наибольший, а у IGM наименьший размер ящика, что опять доказывает наибольшую волатильность первых экций.

Теперь рассчитаем стандартное отклонение для каждой акции.

In [258...

```
stocks_df.std()
```

Out[258...

```
IGM          0.014600
Adobe        0.020322
Microsoft    0.017496
PayPal       0.021532
dtype: float64
```

Стандартное отклонение в годовом исчислении.

In [259...

```
stocks_df.std() * np.sqrt(252)
```

Out[259...

```
IGM          0.231775
Adobe        0.322594
Microsoft    0.277738
PayPal       0.341812
dtype: float64
```

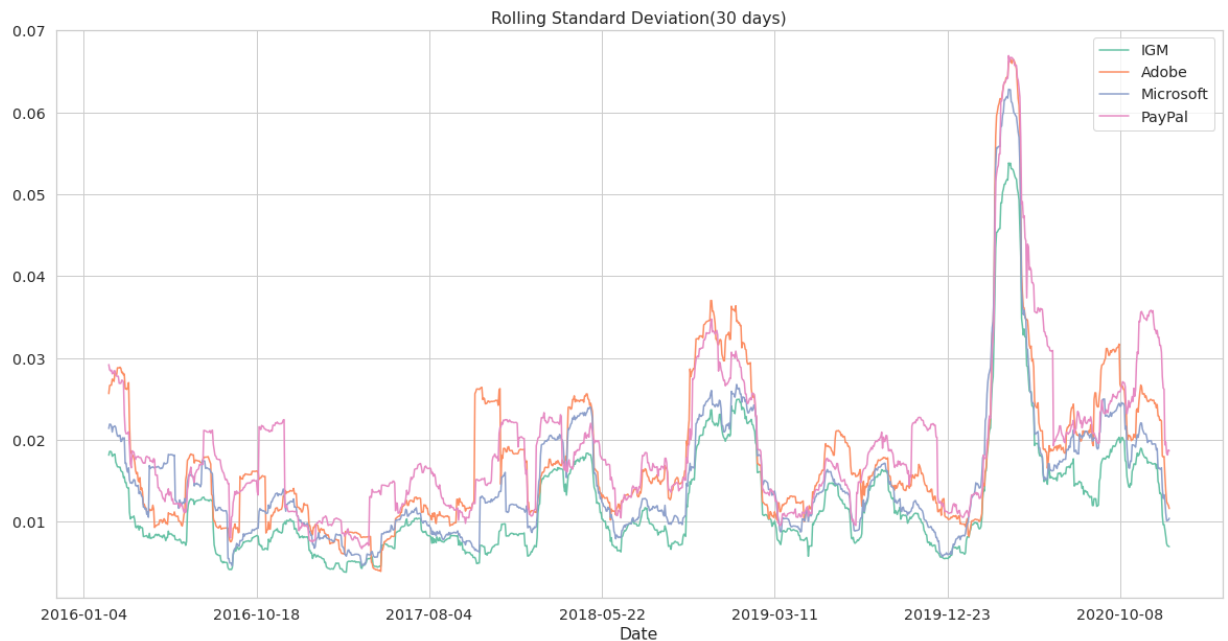
Затем воспользуемся методом скользящей средней, который является одним из самых универсальных и популярных индикаторов на рынке, который может быть использован для торговли по тренду. (Методом подбора посмотрим с каким коэффициентом следует



его брать для наибольшей выразительности). Сделаем коэффициент равным 30, так как жто ближе всего к среднему числу дней в месяце, а по такому интервалу логичнее всего судить об акциях на мой взгляд.

In [305...

```
stocks_df.rolling(window=30).std().plot(figsize=(20, 10), title="Rolling Star
```



Отметим интересный факт, что года примерно одинаково влияют на волатильность акций, подъем и спад их всех приходится примерно на одни и те же года.

И в конце рассчитаем еще один показатель риска, который часто используется и учитывается "yahoo" как один из 5 самых показательных - годовой коэффициент Шарпа.

In [261...

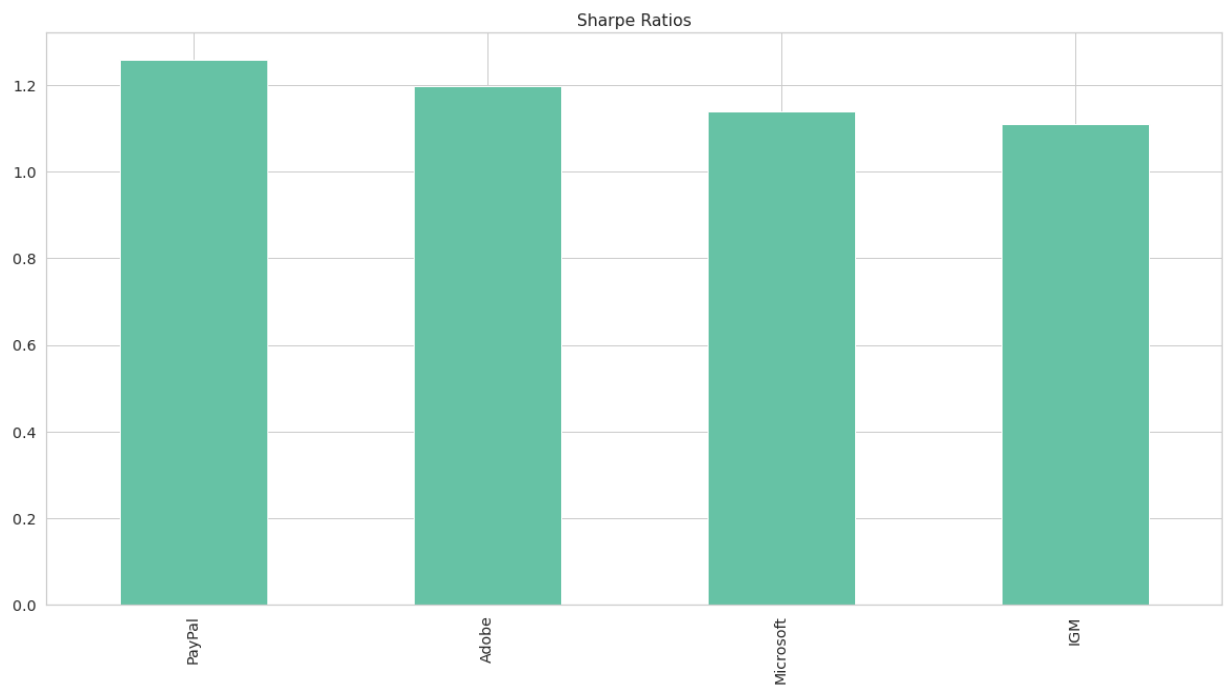
```
sharpe_ratios = (stocks_df.mean() * 252) / (stocks_df.std() * np.sqrt(252))
sharpe_ratios = sharpe_ratios.sort_values(ascending=False)
sharpe_ratios
```

Out[261...

```
PayPal      1.258328
Adobe       1.196438
Microsoft   1.138533
IGM         1.109277
dtype: float64
```

In [262...

```
sharpe_ratios.plot(figsize=(20, 10), kind="bar", title="Sharpe Ratios");
```



Как мы видим, все коэффициенты получились больше 1, то есть все они являются оптимальными.

Таким образом, хочется подытожить, что если мы, как инвесторы, хотим выбрать самое "доходное" решение, то нам стоит выбрать PayPal или Adobe для инвестирования, однако если нам важнее стабильность, то есть наименьшая волатильность, то, что логично, стоит выбирать IGM (я бы сказала, что в этом и есть смысл ETF, ведь если у вас есть пакет акций, то даже если у каких-то из них сильный упадок, то другие будут это компенсировать и в среднем все будет стабильно, а вот у одиночной компании такого не будет).