In [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
import simfin as sf

sf.set_api_key('6fdb2dc3-d138-4b2c-9bc9-57041a206c28')
sf.set_data_dir('simfin_data')
```

In [ ]:
```
import warnings
warnings.filterwarnings('ignore')

from IPython.display import Image, display
```

# Long-Term Stock Forecasting

In [ ]:
```
display(Image(filename='meme.jpg', width=800, height=600))
```


**Урок по инвестированию**

*Покупай акцию всякий раз, когда график выглядит как белка, сидящая на плече клоуна. Это называется технический анализ.*

Этот ноутбук посвящен изучению вопроса предсказания дохода от акций в долгосрочной перспективе.

## Из далека

### Вспомнить всё

- **Акция** — ценная бумага, которая со временем как то меняет свою стоимость.
- **Портфель** — набор акций.
- **Стоимость ценной бумаги** — случайная величина $X_t : \Omega \to \mathrm{R}$
- **Доходность** — случайная величина $R_t = \frac{X_{t+1} - X_t}{X_t}$
- Доход == Выручка, однако Доход — не Прибыль

Обратите внимание на то, что доходность у нас считается относительно следующего дня! То есть $R_t$ в каком то смысле показывает прирост(или убыток), который произойдет в следующий день:

Если $R_t > 1$, то завтра акция будет стоить дороже.

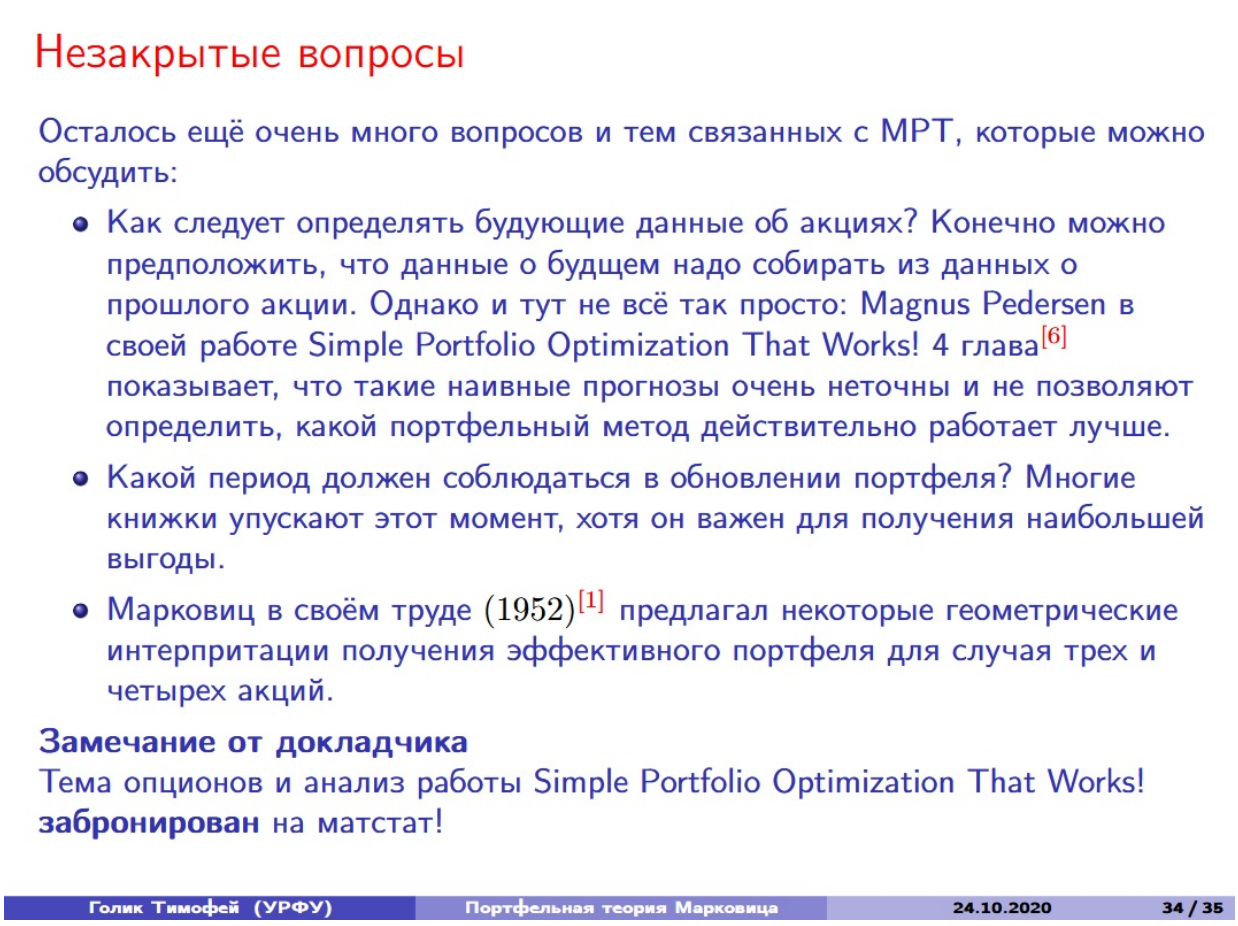
У этого есть технические причины, о которых можно почитать в разделе дополнительных материалов.

Прошлый доклад с тервера был посвящён составлению портфеля, оптимизации и диверсификации. В нем мы постарались ответить на вопрос:

> Как мне вложиться в имеющийся набор акций, чтобы в будущем получить наибольшую доходность ?

Мы тогда дали решение и строгое математическое обоснование. Однако в конце доклада был поставлен вопрос: Как следует определять будущие данные об акциях?

In [ ]:
```
display(Image(filename='slide.png', width=800, height=600))
```




Действительно: мы конечно можем протестировать нашу модель на каких то уже известных нам данных, но как собрать портфель на практике?

## Наивное предсказание

Покажем, что предсказывать будущую стоимость акции, основываясь на прошлом - плохая идея. Для этого будем использовать данные от `simfin`.

In [ ]:
```
# Интерфейс для работы с рынком USA
hub = sf.StockHub(market='us', refresh_days_shareprices=100)

# Скачаем данные об акциях за каждый день
df_daily_prices = hub.load_shareprices(variant='daily')
```

```
Dataset "us-shareprices-daily" on disk (24 days old).
- Loading from disk ... Done!
```

Мы например можем запросить данные об акциях Apple( `AAPL` )

In [ ]:
```
df_daily_prices.loc["AAPL"].head(5)
```

Out[ ]:

| Date | SimFinId | Open | High | Low | Close | Adj. Close | Volume | Dividend | Shares Outstanding |
|---|---|---|---|---|---|---|---|---|---|
| **2019-06-06** | 111052 | 45.77 | 46.37 | 45.54 | 46.30 | 44.54 | 90105244 | NaN | 1.840430e+10 |
| **2019-06-07** | 111052 | 46.63 | 47.98 | 46.44 | 47.54 | 45.73 | 122737572 | NaN | 1.840430e+10 |
| **2019-06-10** | 111052 | 47.95 | 48.84 | 47.91 | 48.15 | 46.31 | 104883404 | NaN | 1.840430e+10 |
| **2019-06-11** | 111052 | 48.72 | 49.00 | 48.40 | 48.70 | 46.85 | 107731528 | NaN | 1.840430e+10 |
| **2019-06-12** | 111052 | 48.49 | 48.99 | 48.35 | 48.55 | 46.70 | 73012756 | NaN | 1.840430e+10 |

Это конечно круто, но использовать в сыром виде данные об акциях не очень удобно:

- Для одно временного промежутка данные об акциях могут отсутствовать, из-за чего будем видеть `NAN`
- Много лишнего
- Нет значения доходности

Давайте исправим этот момент. Пока просто получим значение доходности для каждой акции обработав `NAN`-ы и остальные тонкие моменты

In [ ]:
```
# В этом скрипте происходит вся магия по обработке данных
import data_loader

# 1. Убираем `плохие` тикеты
# 2. Заполняем данные в те дни, когда стоят nan-ы
# 3. Будем считать дневной доход для каждого дня
daily_returns = data_loader.prepare_data(df_daily_prices)
```

In [ ]:
```
daily_returns
```

Out[ ]:

| Ticker | A | AA | AAL | AAON | AAP | AAPL | AAWW | AB | ABBV | ABCB | ... | ZGNX | ZION |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date | | | | | | | | | | | | | |
| 2019-06-06 | 1.005130 | 0.993753 | 1.017958 | 1.013813 | 1.002887 | 1.026718 | 1.049750 | 0.995338 | 1.004723 | 0.991412 | ... | 1.008792 | 0.986535 |
| 2019-06-07 | 1.011108 | 1.006770 | 0.994773 | 0.995564 | 0.988196 | 1.012683 | 1.010532 | 1.009953 | 0.993788 | 1.016726 | ... | 1.019995 | 1.004457 |
| 2019-06-10 | 0.997179 | 1.004323 | 0.994089 | 0.986314 | 0.991260 | 1.011661 | 1.028040 | 0.992464 | 1.015881 | 1.021152 | ... | 0.986177 | 1.013311 |
| 2019-06-11 | 1.002829 | 0.976566 | 1.016848 | 0.996773 | 0.989273 | 0.996798 | 0.979483 | 1.028621 | 1.005155 | 0.993959 | ... | 1.003313 | 0.994253 |
| 2019-06-12 | 1.000148 | 1.029873 | 1.064003 | 1.024280 | 1.000446 | 0.999786 | 1.037457 | 1.033504 | 1.004798 | 1.010130 | ... | 1.008890 | 1.012662 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2024-05-02 | 1.014067 | 1.023073 | 0.999278 | 0.862602 | 1.018589 | 1.059807 | 1.000000 | 1.018146 | 1.018501 | 1.023285 | ... | 1.000000 | 1.021045 |
| 2024-05-03 | 1.008309 | 1.014851 | 1.057762 | 0.971959 | 0.967928 | 0.990905 | 1.000000 | 0.999028 | 0.993544 | 0.998578 | ... | 1.000000 | 1.001895 |
| 2024-05-06 | 1.004657 | 1.006775 | 0.972696 | 1.029640 | 1.008734 | 1.003760 | 1.000000 | 0.985728 | 0.998853 | 0.995117 | ... | 1.000000 | 1.002601 |
| 2024-05-07 | 1.010841 | 0.975236 | 1.011930 | 0.981192 | 1.003711 | 1.001873 | 1.000000 | 0.990457 | 0.987117 | 1.002658 | ... | 1.000000 | 1.007311 |
| 2024-05-08 | 1.016440 | 1.019597 | 1.004161 | 1.009910 | 1.011502 | 1.010007 | 1.000000 | 1.020598 | 0.999677 | 1.008768 | ... | 1.000000 | 1.003512 |

1240 rows × 1838 columns

Теперь займемся анализом реальных данных: изучим как изменялось скользящие среднее, стандартное квадратичное отклонение, а так же рассмотрим корреляцию между двумя акциями

Выберем, например, акции Apple( `AAPL` ) и Amazon( `AMZN` )

In [ ]:
```
windows = [20, 60, 250]
tickers = ['AAPL', 'AMZN']

rets = daily_returns[tickers]
```

Теперь для каждого окна посчитаем статистики, а затем объединим результат

In [ ]:
```
list_mean = []
list_std = []
list_corr = []

for window in windows:
    # stack ещё по дефолту отбрасывает Nan-ы у первых окон. Поэтому нам не надо их самими убирать
    mean = rets.rolling(window=window).mean().stack().rename(str(window) + ' Days')
    std = rets.rolling(window=window).std().stack().rename(str(window) + ' Days')
    corr = rets.rolling(window=window).corr().stack().rename(str(window) + ' Days')

    # Переименуем
    corr.index.rename(['Date', 'Ticker', 'Ticker2'], inplace=True)

    list_mean.append(mean)
    list_std.append(std)
    list_corr.append(corr)

# Представляем в виде фрэймов pandas
df_mean = pd.concat(list_mean, axis=1).dropna()
df_std = pd.concat(list_std, axis=1).dropna()
df_corr = pd.concat(list_corr, axis=1).dropna()
```

Посмотрим на то, что у нас получилось

In [ ]:
```
df_mean
```

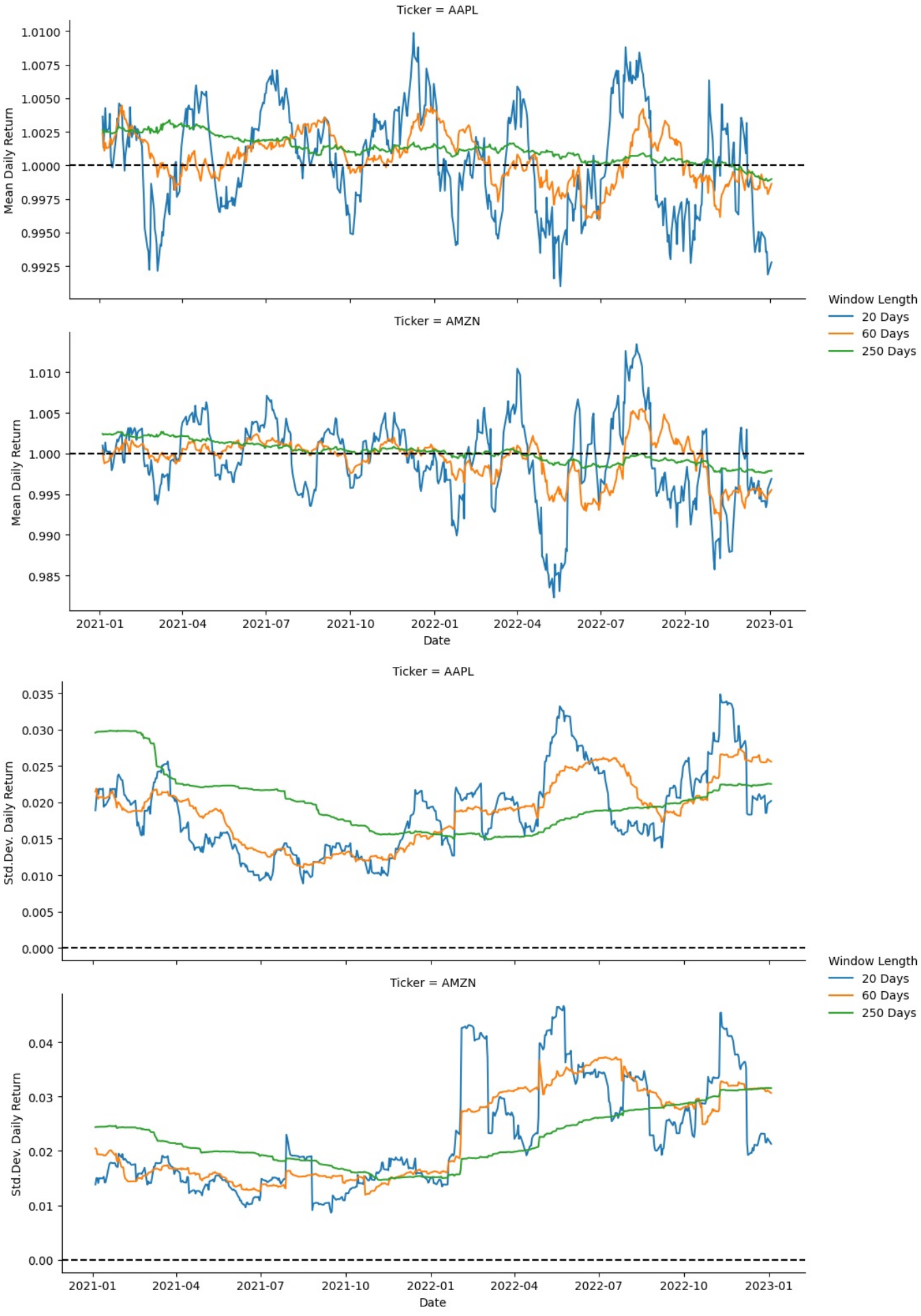
Out[ ]:

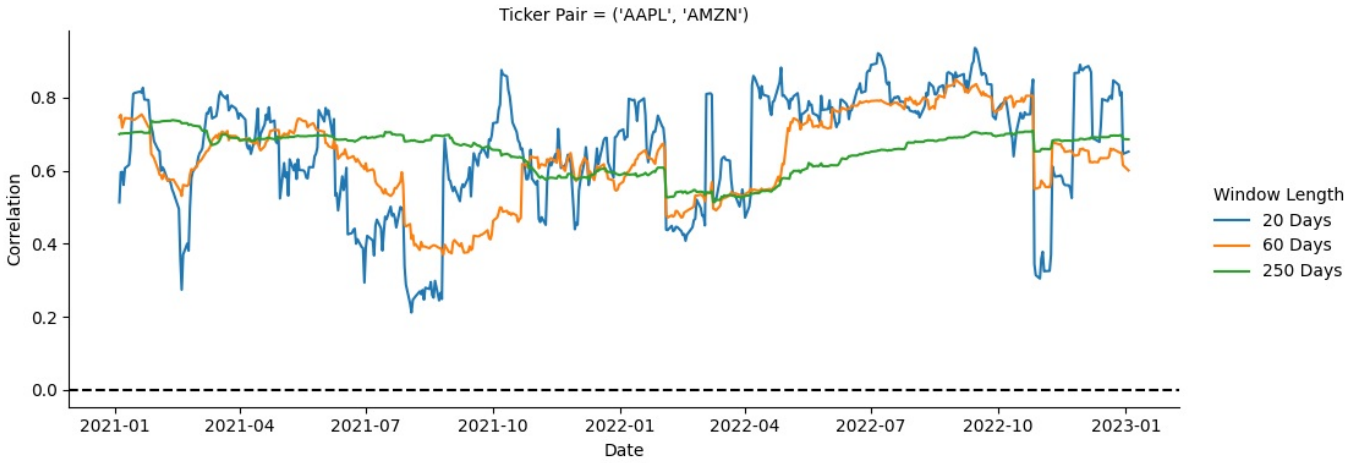
| Date | Ticker | 20 Days | 60 Days | 250 Days |
|---|---|---|---|---|
| **2020-06-02** | **AAPL** | 1.004623 | 1.004150 | 1.002617 |
| | **AMZN** | 1.003414 | 1.005774 | 1.001572 |
| **2020-06-03** | **AAPL** | 1.003681 | 1.002807 | 1.002476 |
| | **AMZN** | 1.002334 | 1.004810 | 1.001430 |
| **2020-06-04** | **AAPL** | 1.004582 | 1.003861 | 1.002539 |
| **...** | **...** | ... | ... | ... |
| **2024-05-06** | **AMZN** | 1.000991 | 1.001426 | 1.002452 |
| **2024-05-07** | **AAPL** | 1.004477 | 0.999721 | 1.000308 |
| | **AMZN** | 1.000714 | 1.001560 | 1.002302 |
| **2024-05-08** | **AAPL** | 1.002813 | 1.000076 | 1.000343 |
| | **AMZN** | 1.000280 | 1.002051 | 1.002262 |

1982 rows × 3 columns

Осталось только отобразить эти данные. Конечно, сам процесс отрисовки имеет свои тонкости, однако это не столь важно для нашего повествования.

In [ ]:
```
import plot_drawer

def get_from_range(df, start, end):
    df = df.copy()
    df.index = df.index.set_levels(pd.to_datetime(df.index.levels[0]), level=0)
    return df.loc[start:end]

# опционально. Можно рисовать график для всего доступного промежутка
if True:
    start_year = '2021'
    end_year = '2023'
    df_mean = get_from_range(df_mean, start_year, end_year)
    df_std = get_from_range(df_std, start_year, end_year)
    df_corr = get_from_range(df_corr, start_year, end_year)

plot_drawer.draw_return_statatistics_plots(df_mean, df_std, df_corr, tickers, windows)
```


Ticker = AAPL
Mean Daily Return
1.0100
1.0075
1.0050
1.0025
1.0000
0.9975
0.9950
0.9925
Ticker = AMZN
Mean Daily Return
1.010
1.005
1.000
0.995
0.990
0.985
2021-01
2021-04
2021-07
2021-10
2022-01
2022-04
2022-07
2022-10
2023-01
Date
Window Length
20 Days
60 Days
250 Days
Ticker = AAPL
Std.Dev. Daily Return
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Ticker = AMZN
Std.Dev. Daily Return
0.04
0.03
0.02
0.01
0.00
2021-01
2021-04
2021-07
2021-10
2022-01
2022-04
2022-07
2022-10
2023-01
Date
Window Length
20 Days
60 Days
250 Days




Из графиков мы можем сделать следующие выводы

- Для окна в 20 дней доходность имеет высокую волатильность(сильно изменчива со временем)
- По мере увеличения окна статистики начинают вести себя менее скачкообразно
- Окно в 250 дней выглядит более предсказуемым
- Хоть и для окна в 250 дней мы можем наблюдать какой-то тренд, но даже так статистика в некоторых моментах отклоняется от него.

### А что если использовать статистику для окна в 1 год для предсказания будущего в 1 месяц?

Пользоваться таким допущением нельзя! **Статистика подсчитанная за период в 1 год не содержит в себе информации об промежутках меньших 1 года**. Это видно по самим графикам. Да и не трудно придумать простой пример.

Пусть у нас доходность по месяцам представляла следующее: `[1.02, 1.02, 1.02, 1.02, 1.02, 1.02, 1.02, 1.02, 1.02, 1.02, 1.02, 0.85]`

Тогда годовая доходность будет составлять

In [ ]:
```
(11 * 1.02 + 0.85) / 12
```

Out[ ]:
```
1.0058333333333334
```

Но считать, что в следующий месяц стоимость акции возрастет на `0.5%` глупо.

# Long-Term Stock Forecasting

## Теоретическая модель

Мы можем получить доход от акции двумя способами:

- Дивиденды — выплаты, которые даются владельцам акции соразмерно с их долей владением
- Изменение в стоимости — вчера дешевле, завтра подороже

Сделаем следующее предположение:

Если акция даёт нам дивиденды, то полученные средства мы сразу вкладываем на покупку ещё акций этой же компании.

Тогда общий доход (обозначим его за $\textbf{Total Return}_t$) в момент времени $t$

$$\text{Total Return}_t = \text{Shares}_t \cdot \text{Share Price}_t$$

Где

- $\text{Share Price}$ - цена акции
- $\text{Share}$ - кол-во акций.

Таким образом $\text{Total Return}_t$ учитывает теперь доходы и с дивидендов(так как они будут влиять на количество акций, которыми мы владеем), и с изменением цен ($\text{Share Price}_t$).

Теперь ввёдем такое понятие, как доходность пересчитанная на годовое значение (Его обозначим как $\textbf{Ann Return}_t$), **которое мы получем в будущем если купим акций вот сейчас**.

$$\text{Ann Return}_t = \left( \frac{\text{Total Return}_{t+\text{Years}}}{\text{Total Return}_t} \right)^{1/\text{Years}} - 1.$$

И теперь, если мы вспомним чему равно $\text{Total Return}_t$, получим следующее:

$$\text{Ann Return}_t = \left( \frac{\text{Shares}_{t+\text{Years}} \cdot \text{Share Price}_{t+\text{Years}}}{\text{Shares}_t \cdot \text{Share Price}_t} \right)^{1/\text{Years}} - 1$$

## Мультипликаторы

Мультипликаторы в инвестициях — это показатели, которые позволяют сравнивать компании друг с другом без учёта их масштаба и других факторов.

Перечислим некоторые из них:

- P/E (Price to Earnings) — отношение капитализации компании к чистой прибыли за год.
- P/Sales (Price to Sales) — отношение капитализации к годовой выручке.
- PEG (Price Earnings Growth) — отношение P/E к прогнозируемому росту прибыли или средним темпам прироста прибыли за 5 лет.

Более формально $\text{P/Sales}_t$ определяется следующим образом

$$\text{P/Sales}_t = \frac{\text{Share Price}_t \cdot \text{Total Assets}_t}{\text{Sales Per Share}_t \cdot \text{Total Assets}_t} = \frac{\text{Share Price}_t}{\text{Sales Per Share}_t},$$

где

- $\text{Share Price}_t$ - цена акции
- $\text{Total Assets}_t$ - кол-во акций
- $\text{Sales Per Share}_t$ - часть дохода, которая приходится на каждую акцию. Если к примеру у компании $500,000$ акций, а доход $\$1,000,000$, тогда

$$\text{Sales Per Share} = \frac{1,000,000}{500,000} = 2.$$

То есть на каждую акцию приходится $\$2$ дохода.

$\text{P/Sales}$ показывает насколько сильно акция переоценена на рынке. Если $\text{P/Sales} > 5$, то акции переоценены, её стоимость не пропорциональна её доходности. Такое значение свойственно для технологических и инновационных компаний, в которые верит рынок. И напротив, если $\text{P/Sales} < 1$, то акции недооценены: компания имеет значительную выручку по сравнению с её рыночной стоимостью, рынок не верит в рост акции.

Теперь запишем стоимость акции как

$$\text{Share Price}_t = \text{P/Sales}_t \cdot \text{Sales Per Share}_t$$

и подставить в формулу годовой доходности

$$\text{Ann Return}_t = \left( \overbrace{\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t}}^{\text{Reinvest Dividends}} \cdot \overbrace{\frac{\text{P/Sales}_{t+\text{Years}}}{\text{P/Sales}_t}}^{\text{Revaluation}} \cdot \overbrace{\frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}}^{\text{Sales Growth}} \right)^{\frac{1}{\text{Years}}} - 1.$$

Мы получили формулу для годовой доходности, которую можно разбить на 3 компоненты:

- $\text{Reinvest Dividends}$ — **Реинвистируемые Дивиденды** — ожидаемый рост доход от дивидентов, которые мы сразу вложим в акцию
- $\text{Revaluation}$ — **Ревальвация** — ожидаемый рост значения мультипликатора
- $\text{Sales Growth}$ — **Рост продаж/доходности** — ожидаемый рост дохода компании

Заметим, что эта формула точно отвечает на вопрос о годовой доходности акции спустя годы, если мы знаем точные значения трех параметров. Если мы например уверены, что какое-то из 3 параметров будет иметь определённое значение в будущем, то мы можем его подставить и упростить дальнейший анализ.

# Прогнозирование среднего дохода

Конечно мы не знаем будущего, поэтому не можем сказать точные значения трех параметров: $\text{Reinvest Dividends}$, $\text{Revaluation}$ и $\text{Sales Growth}$. Поэтому лучшее что мы можем сделать — найти матожидание доходности и стандартное квадратичное отклонение, чтобы понимать, как сильно отстраняется настоящее значение от среднего.

Немного напомним об обозначениях:

- $\text{P/Sales}$ — отношение капитализации к годовой выручке.

- $\text{Sales Per Share}_t$ - часть дохода, которая приходится на каждую акцию.
- $\text{Shares}$ - число наших акций во владении
- $\text{Ann Return}_t$ - показатель за некоторый промежуток времени, который приводится к годовому

Запишем матожидание от $\text{Ann Return}_t$:

$$E[\text{Ann Return}_t] = E\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t} \cdot \frac{\text{P/Sales}_{t+Years}}{\text{P/Sales}_t} \cdot \frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}} - 1\right]$$

Из-за того, что $\text{P/Sales}_t$ нам известна(константа в момент времени $t$), мы можем по свойства матожидания её извлечь

$$E[\text{Ann Return}_t] = \frac{E\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t} \cdot \text{P/Sales}_{t+\text{Years}} \cdot \frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}}\right]}{\text{P/Sales}_t^{1/\text{Years}}} - 1$$

А теперь сделаем допущение, предположив что 3 компоненты не зависят друг от друга. Да это неправильно суждение и мы скорее всего будем получать какие-то ошибки при расчётах. Но ведь попробовать никто не запрещает =)

$$E[\text{Ann Return}_t] = \frac{E\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t}\right)^{1/\text{Years}}\right] \cdot E[(\text{P/Sales}_{t+\text{Years}})^{1/\text{Years}}] \cdot E\left[\left(\frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}}\right]}{\text{P/Sales}_t^{1/\text{Years}}} - 1$$

На первый взгляд может показаться, что всё стало ещё запутаннее, но давайте разберёмся с каждой компонентой:

- $\text{Shares}_{t+\text{Years}}$ — это будущее количество акций в нашем владении. Так как мы договорились реинвестировать деньги полученные с дивидендов в покупку ещё акций, то выражение

${\color{green}\left( \dfrac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t} \right)^{1/\text{Years}}}
п р и б л и з и т е л ь н о р а в н о г о д о в о м у д о х о д у д и в и д е н т о в к с т о и м о с т и а к ц и и в t$, которая определяется следующей формулой:

$$\text{Dividend Yield}_t + 1 = \frac{\text{Dividend TTM}}{\text{Share Price}} + 1$$

Где **Dividend TTM** — дивиденды полученные за год с одной акции (**Trailing Twelve Months**). Приблизительно равно потому, что дивиденты выплачиваются по кварталам да и количество акций, которые мы купим после выплат дивидендов определяется стоимостью акций в этот день, а дни в формулах у нас немного разные. И всё же будем считать, что

$$E\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t}\right)^{1/\text{Years}}\right] \simeq E[\text{Dividend Yield} + 1]$$

- $E\left[\left(\frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}}\right]$ — это ничто иное как рост продаж на одну акцию в переводе на годовой показатель. Если мы предположим, что рост продаж не зависит от одного года к другому, то мы можем переписать

$$E\left[\left(\frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}}\right] \simeq E[\text{Sales Growth Rate} + 1]$$

Где $\text{Sales Growth Rate}$ - годовой темп роста выручки.

Объединяя, мы получаем одну из двух главных формул для нашей модели по предсказанию:

$$E[\text{Ann Return}_t] = \frac{a}{\text{P/Sales}_t^{1/\text{Years}}} - 1$$

Где параметр $a$ может быть подсчитан из трёх параметров, которые обсуждали ранее

$$a \simeq E[\text{Dividend Yield} + 1] \cdot E[(\text{P/Sales}_{t+\text{Years}})^{1/\text{Years}}] \cdot E[\text{Sales Growth Rate} + 1]$$

## Прогнозирование std доходности

Теперь найдем $\text{Std}[\text{Ann Return}_t]$.

$$\text{Std}[\text{Ann Return}_t] = \text{Std}\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t} \cdot \frac{\text{P/Sales}_{t+\text{Years}}}{\text{P/Sales}_t} \cdot \frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}} - 1\right]$$

Пользуемся свойствами и тем, что $\text{P/Sales}_t$ нам известно:

$$\text{Std}[\text{Ann Return}_t] = \frac{\text{Std}\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t} \cdot \text{P/Sales}_{t+\text{Years}} \cdot \frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}}\right]}{\text{P/Sales}_t^{1/\text{Years}}}$$

Обозначим числитель $b$. Получаем вторую ключевую формулу для нашей модели:

$$\text{Std}[\text{Ann Return}_t] = \frac{b}{\text{P/Sales}_t^{1/\text{Years}}}$$

Где параметр $b$ может быть посчитан используя рассуждения из прошлого раздела для параметра $a$:

$$b = \text{Std}\left[\left(\frac{\text{Shares}_{t+\text{Years}}}{\text{Shares}_t} \cdot \text{P/Sales}_{t+\text{Years}} \cdot \frac{\text{Sales Per Share}_{t+\text{Years}}}{\text{Sales Per Share}_t}\right)^{1/\text{Years}}\right] \simeq \simeq \text{Std}[(\text{Dividend Yield} + 1) \cdot \text{P/Sales}^{1/\text{Years}} \cdot (\text{Sales Growth Rate} + 1)]$$

Больше ничего со стандартным отклонением не получится сделать, даже если будем считать внутренние компоненты независимыми. Поэтому оставим всё как есть.

## Кривые доходности

Посмотрим на графике как будет устроена зависимость среднегодовой доходности Ann Return от значения мультипликатора **на момент покупки акции**.

$$E[\text{Ann Return}] = \frac{1}{\text{P/Sales}^{1/\text{Years}}} - 1$$

In [ ]:
```
x = np.linspace(0.4, 2.2, 50)
a = 1
years = [1, 2, 3, 4, 5, 8, 10]

plt.figure(figsize=(8,5))
plt.rcParams['figure.dpi'] = 150

for y in years:
    plt.plot(x, a / x**(1/y) - 1, label=f'Year = {y}')

plt.axvline(x=1, color='black')
plt.axhline(y=0, color='black')
plt.legend()
plt.xlabel('P/Sales')
plt.ylabel('Ann Return')
plt.grid()
plt.show()
```

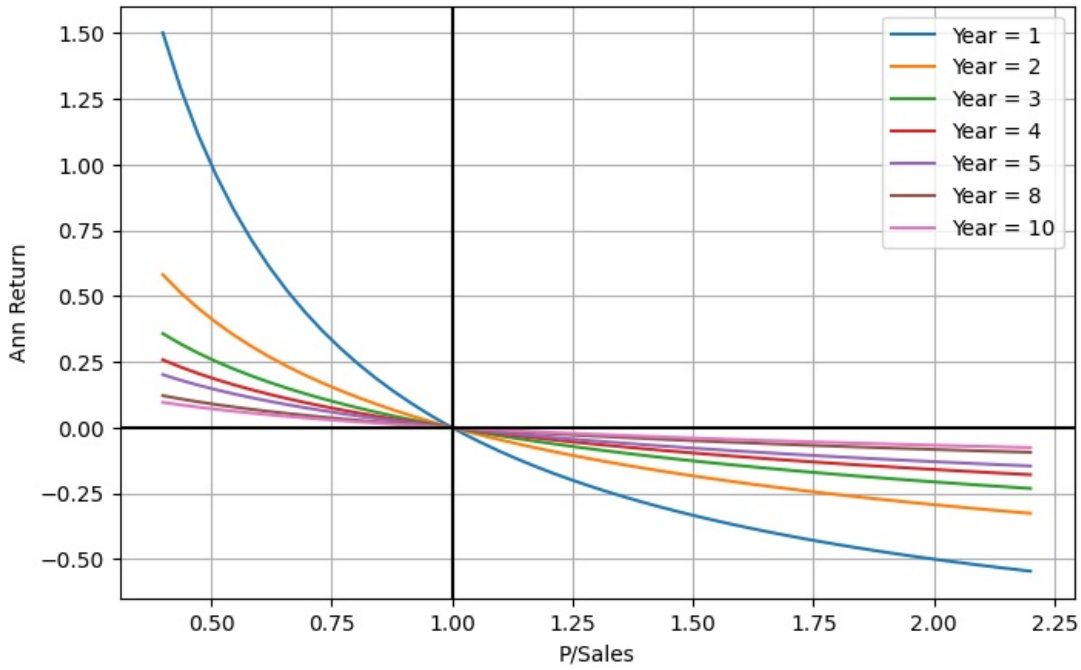



График получился вполне логичным, если понимать суть значения P/Sales. Если акция переоценена (P/Sales > 5), то среднегодовая доходность будет отрицательной(даже несмотря на то, что в будущем компания сможет вырваться вверх), но если акции наоборот недооценены(P/Sales < 1), то мы скорее всего ожидаем прирост стоимости акции в связи с большим притоком денег в компанию

## Тестим!

Посмотрим как теория работает на практике. Передадим нашей модели все данные о будущем, чтобы проверить, что при правильных прогнозах трех параметров она будет выдавать результат, схожий с реальностью.

In [ ]:
```
from data_loader2 import load_stock_data

ticker = 'PG'

df_PG = load_stock_data(ticker, dividend_TTM=True, earnings=False)
```

In [ ]:
```
df_PG.dropna()
```

Out[ ]:

| Date | Total Return | Share-price | Sales Per Share | P/Sales | Sales Growth | Book-Value Per Share | P/Book | Dividend TTM | P/Dividend | Dividend Yield |
|---|---|---|---|---|---|---|---|---|---|---|
| 1994-06-30 | 5.120126 | 13.343750 | 11.070000 | 1.205397 | -0.008065 | 3.230000 | 4.131192 | 1.272088 | 10.489645 | 0.095332 |
| 1994-07-01 | 5.192069 | 13.531250 | 11.073014 | 1.222002 | -0.007773 | 3.231699 | 4.187040 | 1.272527 | 10.633366 | 0.094044 |
| 1994-07-02 | 5.192069 | 13.531250 | 11.076027 | 1.221670 | -0.007481 | 3.233397 | 4.184840 | 1.272967 | 10.629694 | 0.094076 |
| 1994-07-03 | 5.192069 | 13.531250 | 11.079041 | 1.221338 | -0.007189 | 3.235096 | 4.182643 | 1.273407 | 10.626025 | 0.094109 |
| 1994-07-04 | 5.192069 | 13.531250 | 11.082055 | 1.221006 | -0.006897 | 3.236795 | 4.180448 | 1.273846 | 10.622358 | 0.094141 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2020-06-26 | 114.507416 | 115.230003 | 28.513607 | 4.041229 | 0.055080 | 18.851639 | 6.112466 | 3.058132 | 37.679867 | 0.026539 |
| 2020-06-27 | 115.312337 | 116.040003 | 28.517705 | 4.069051 | 0.055167 | 18.851230 | 6.155567 | 3.058571 | 37.939282 | 0.026358 |
| 2020-06-28 | 116.117259 | 116.850004 | 28.521803 | 4.096866 | 0.055255 | 18.850820 | 6.198670 | 3.059011 | 38.198622 | 0.026179 |
| 2020-06-29 | 116.922180 | 117.660004 | 28.525902 | 4.124673 | 0.055342 | 18.850410 | 6.241774 | 3.059451 | 38.457887 | 0.026002 |
| 2020-06-30 | 118.820198 | 119.570000 | 28.530000 | 4.191027 | 0.055334 | 18.850000 | 6.343236 | 3.059890 | 39.076567 | 0.025591 |

9498 rows × 10 columns

Договоримся о том, как будем оценивать качество наших предсказаний. Мы будем использовать 3 статистики:

- MAE — средняя абсолютная ошибка.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |y_t - x_t|$$

- MSE — средняя квадратичная ошибка.

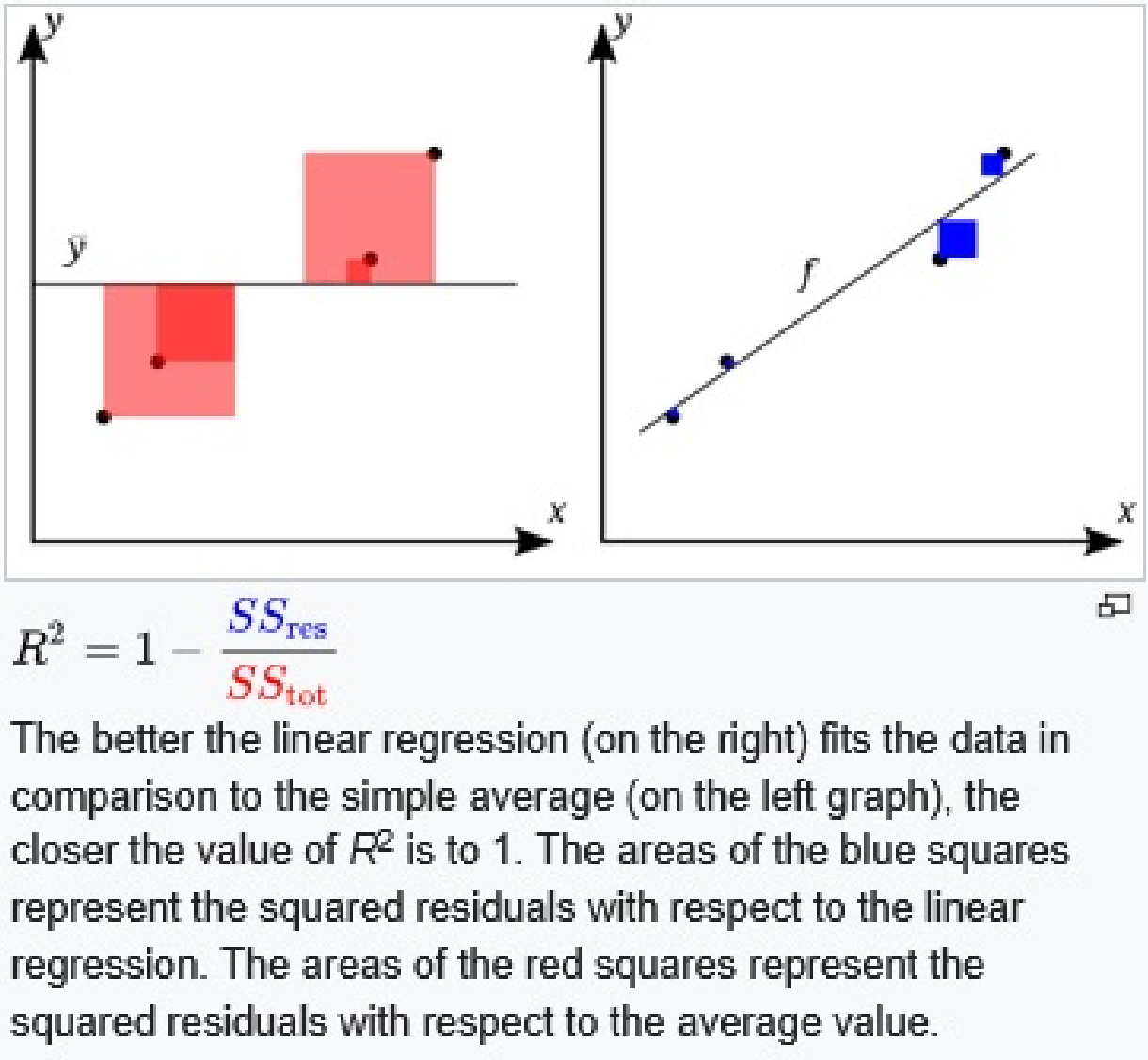
$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} (y_t - x_t)^2$$

- $R^2$ — коэффициент детерминации

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_{t=1}^{n}(y_t - x_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

  Чем ближе значение к 1, тем лучше модель предсказала данные. Действительно, если наша модель предсказывает значения лучше, чем простое среднее(То есть её MSE меньше чем MSE для среднего), то $R^2 > 0$ растет, а если наоборот результаты хуже, то соответственно $R^2 < 0$ уменьшается.

In [ ]:
```
display(Image(filename='r2_explanation.png', width=800, height=600))
```



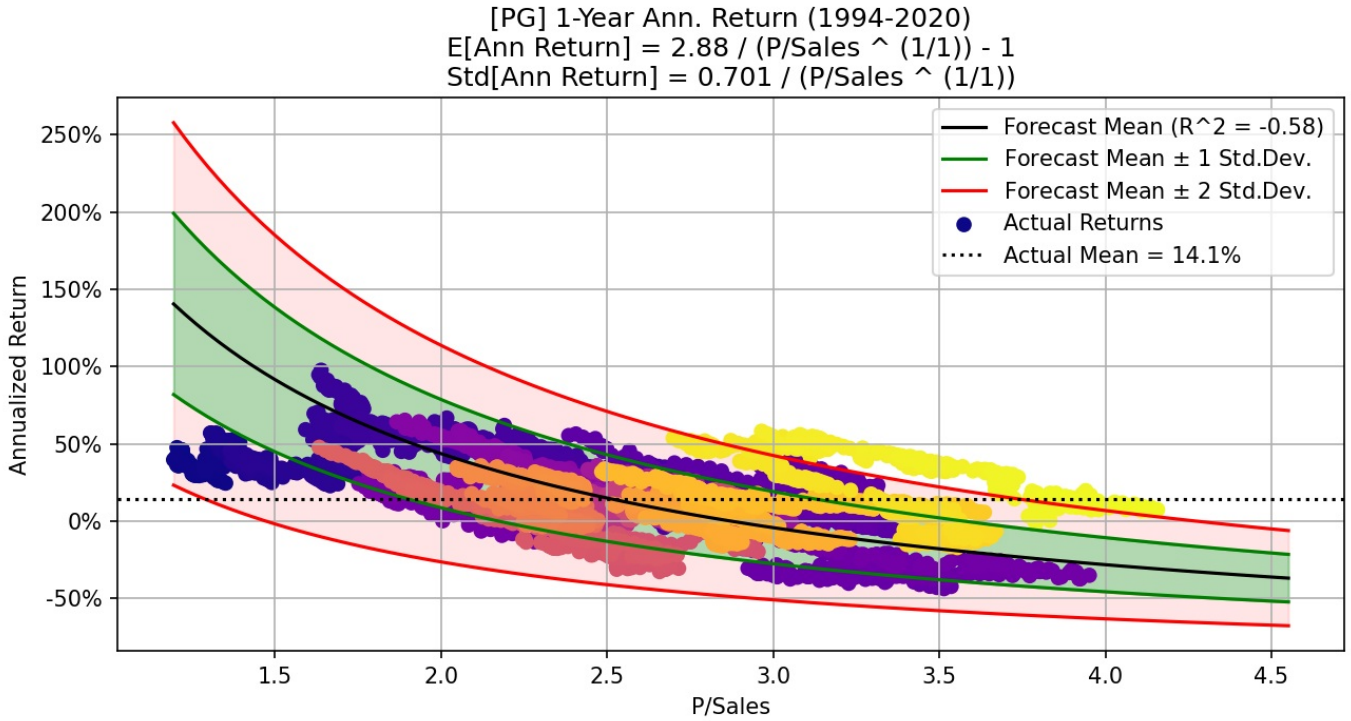

Теперь выведем следующую гипотезу: $H_0$ — наша модель предсказывает хуже исторического среднего, $H_1$ — наша модель предсказывает значения лучше исторического среднего. Для проверки воспользуемся критерием Стьюдента для зависимых выборок `ttest_rel`:

$$\delta(\overrightarrow{\text{Model Error}}, \overrightarrow{\text{Baseline Error}}) = \begin{cases} H_0 : \mu_{\text{Model Error}} \geq \mu_{\text{Baseline Error}} \\ H_1 : \mu_{\text{Model Error}} < \mu_{\text{Baseline Error}} \end{cases}$$

Если p-value будет мало, то это будет свидетельствовать о том, что наши данные при предположении об истинности нулевой гипотезы маловероятны, а значит гипотезу $H_0$ придётся отвергнуть.

In [ ]:
```
from plot_drawer2 import plot_ann_returns

plot = plot_ann_returns(years=1, ticker=ticker, df=df_PG, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    18.3%           14.8%           1.00e+00
MSE:    6.37e-02        4.02e-02        1.00e+00
R^2:    -0.58
```

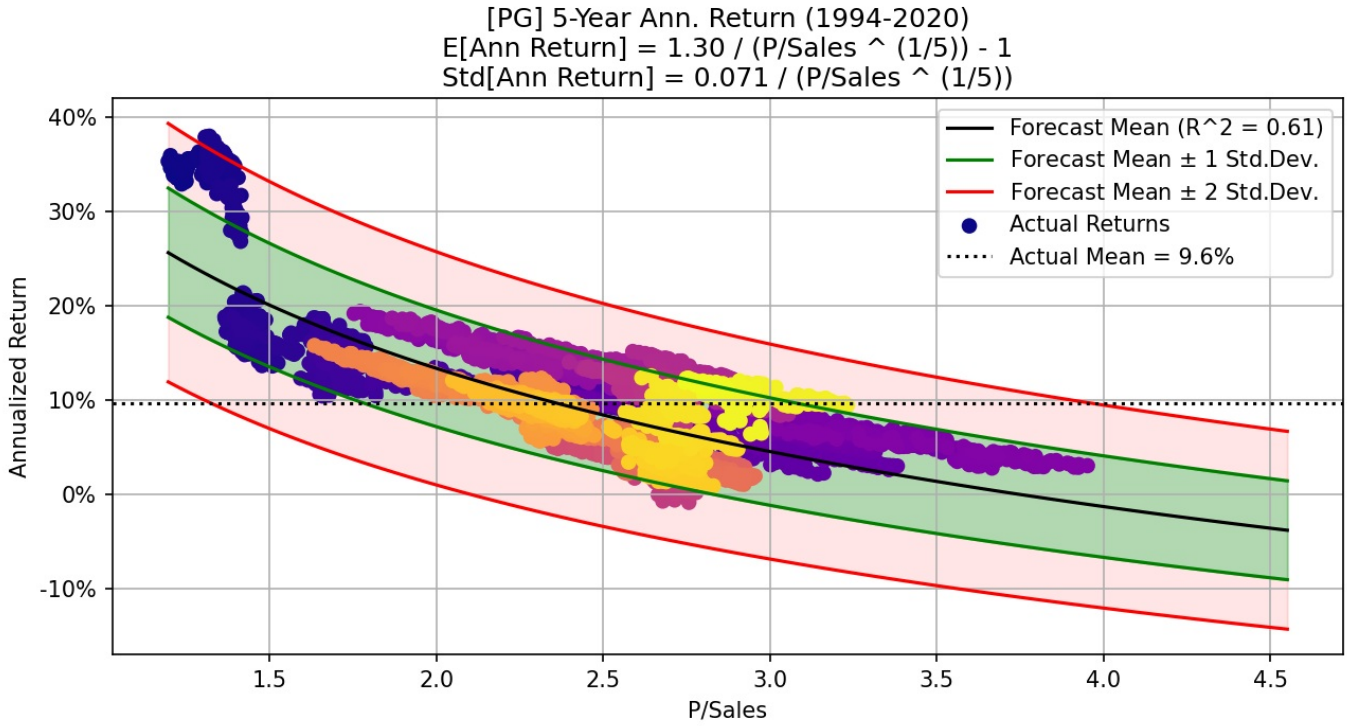



Раскрашенные точки показывают исторические значения годовой доходности при определённом значении мультипликатора. Как же были получены эти точки?

Например, 1 января 1995 года стоимость акции PG составляла USD 15.55. Год позднее 1 января 1996 года стоимость акции была USD 20.77, что даёт доходность $20.77/15.55 - 1 \simeq 33.6$, но так же ещё были выплаты дивидендов, поэтому общий доход составлял около 45.8. К 1 январю 1995 $P/Sales = 1.34$. Значит надо по оси $x$ отложить 1.34, а по оси $y$ значение 45.8. Так мы проделываем для каждого дня начиная с 1994 заканчивая 2020. Причем с увеличением времени цвет переходит от темно фиолетового к светло желтому, чтобы понимать, какая была тенденция.

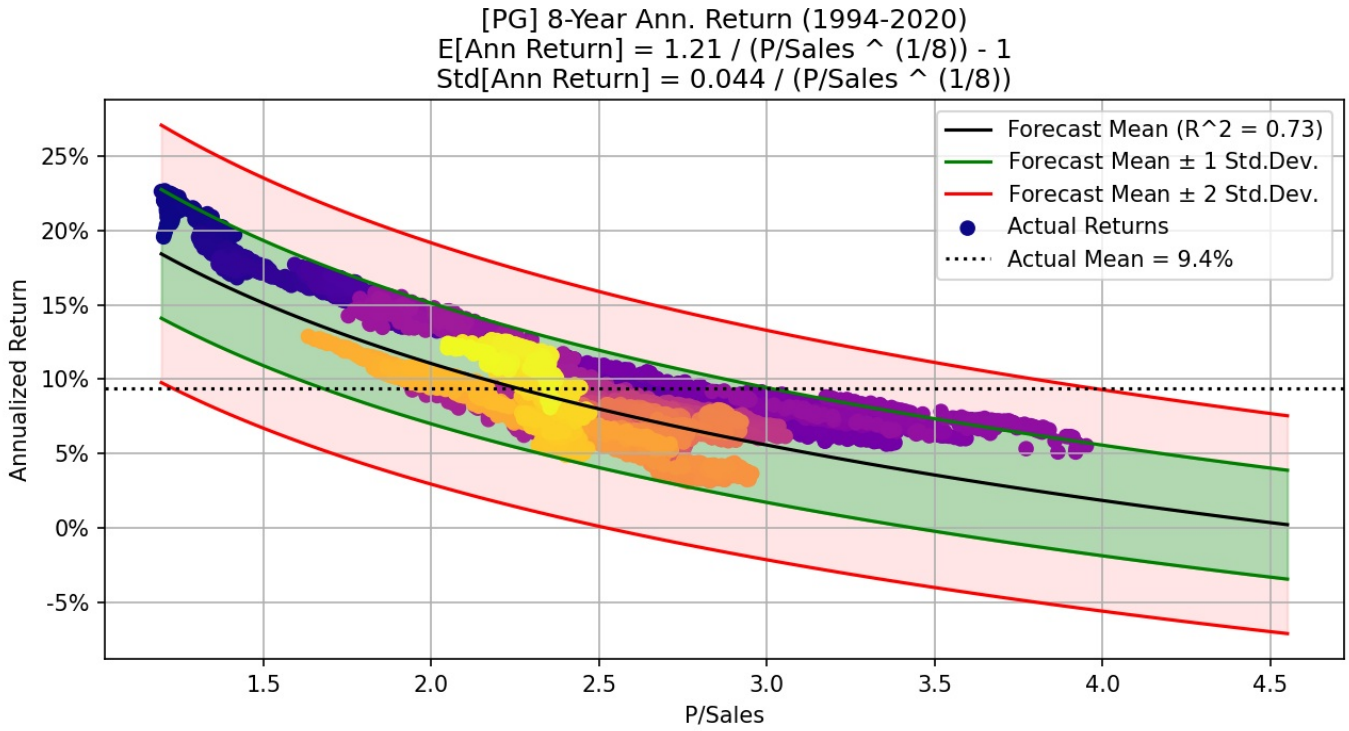
In [ ]:
```
plot = plot_ann_returns(years=5, ticker=ticker, df=df_PG, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    3.1%            4.3%            7.56e-199
MSE:    1.48e-03        3.83e-03        5.63e-119
R^2:    0.61
```




In [ ]:
```
plot = plot_ann_returns(years=8, ticker=ticker, df=df_PG, print_stats=True)
```

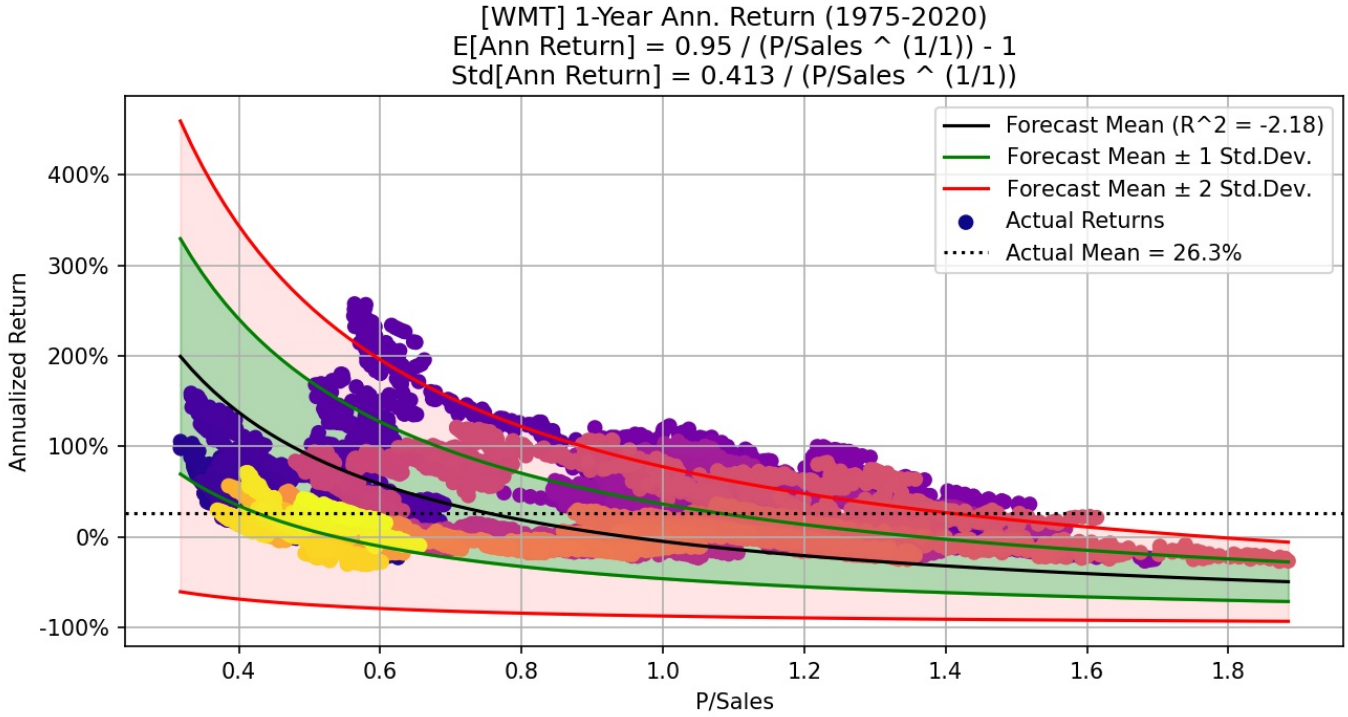
```
        Forecast        Baseline        p-value
================================================
MAE:    1.7%            3.0%            0.00e+00
MSE:    4.10e-04        1.52e-03        7.70e-276
R^2:    0.73
```




Как мы видим, в модели, охватывающей год, ошибки очень большие и даже хуже среднего. Он вышел слишком зашумленным и случайным Но ситуация меняется при рассмотрении в промежутке 5-ти лет, и ещё сильнее - 8-ми лет. Ошибки уже заметно меньше(при рассмотрении 8-ми летнего случая мы попали в границы одного Std), а R растёт. То есть характеристика P/Sales очень хорошо справляется в более долгосрочном рассмотрении. Однако, это не всегда так.

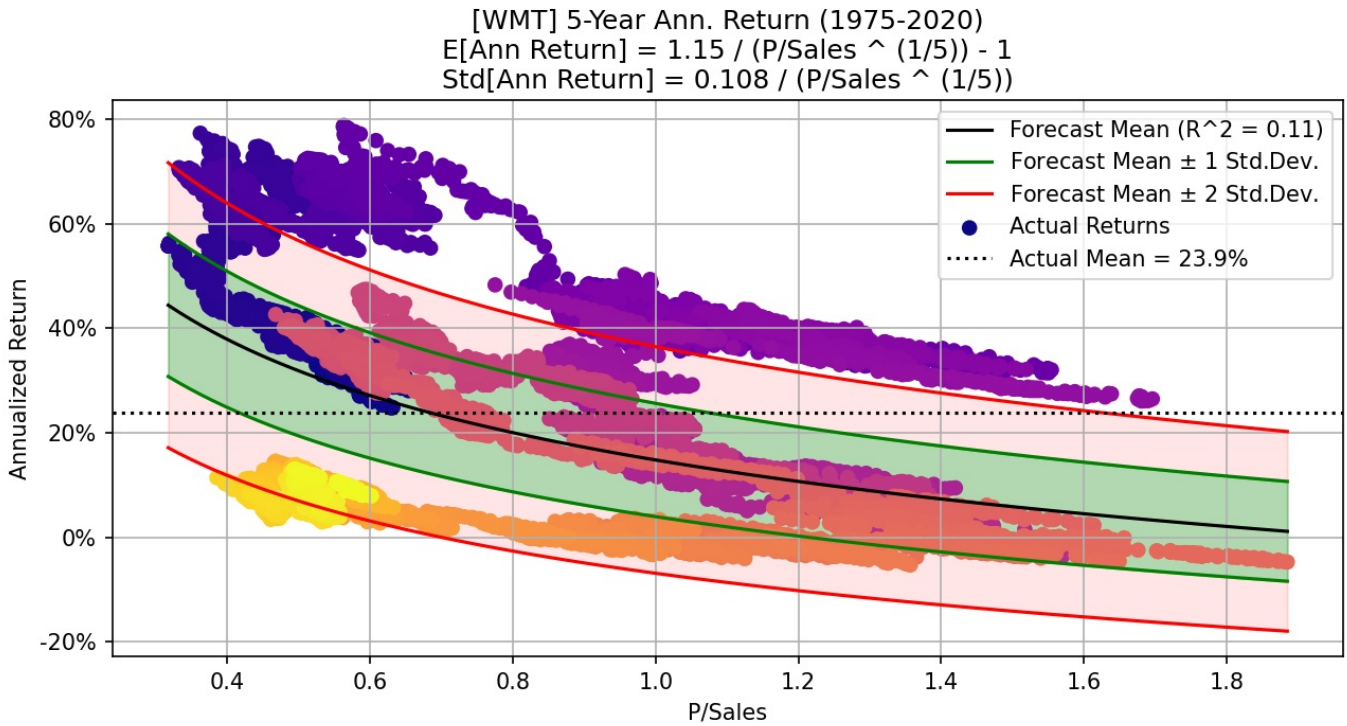
In [ ]:
```
ticker = 'WMT'

df_WMT = load_stock_data(ticker, dividend_TTM=True)

plot = plot_ann_returns(years=1, ticker=ticker, df=df_WMT, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    58.0%           27.8%           1.00e+00
MSE:    4.36e-01        1.37e-01        1.00e+00
R^2:    -2.18
```



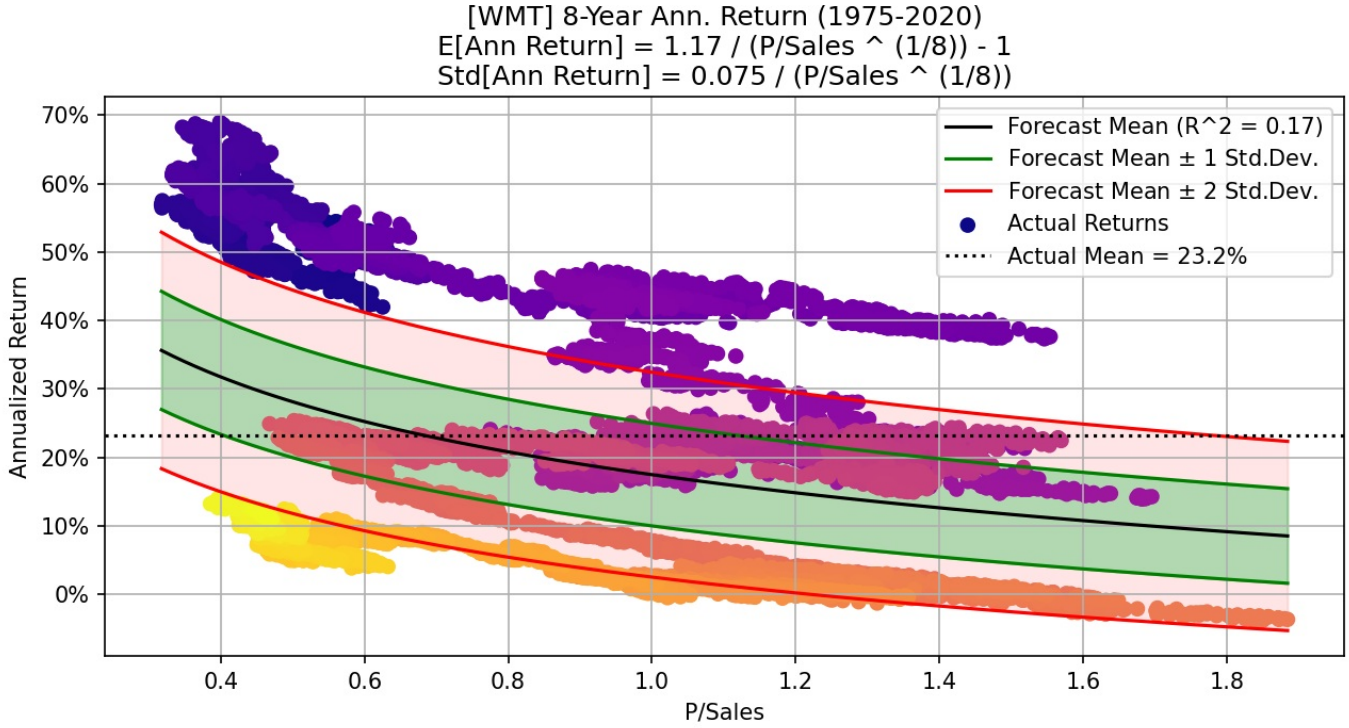

In [ ]:
```
plot = plot_ann_returns(years=5, ticker=ticker, df=df_WMT, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    17.4%           18.8%           4.92e-65
MSE:    4.12e-02        4.65e-02        3.03e-55
R^2:    0.11
```




In [ ]:
```
plot = plot_ann_returns(years=8, ticker=ticker, df=df_WMT, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    14.8%           15.5%           4.52e-45
MSE:    2.95e-02        3.54e-02        2.64e-187
R^2:    0.17
```

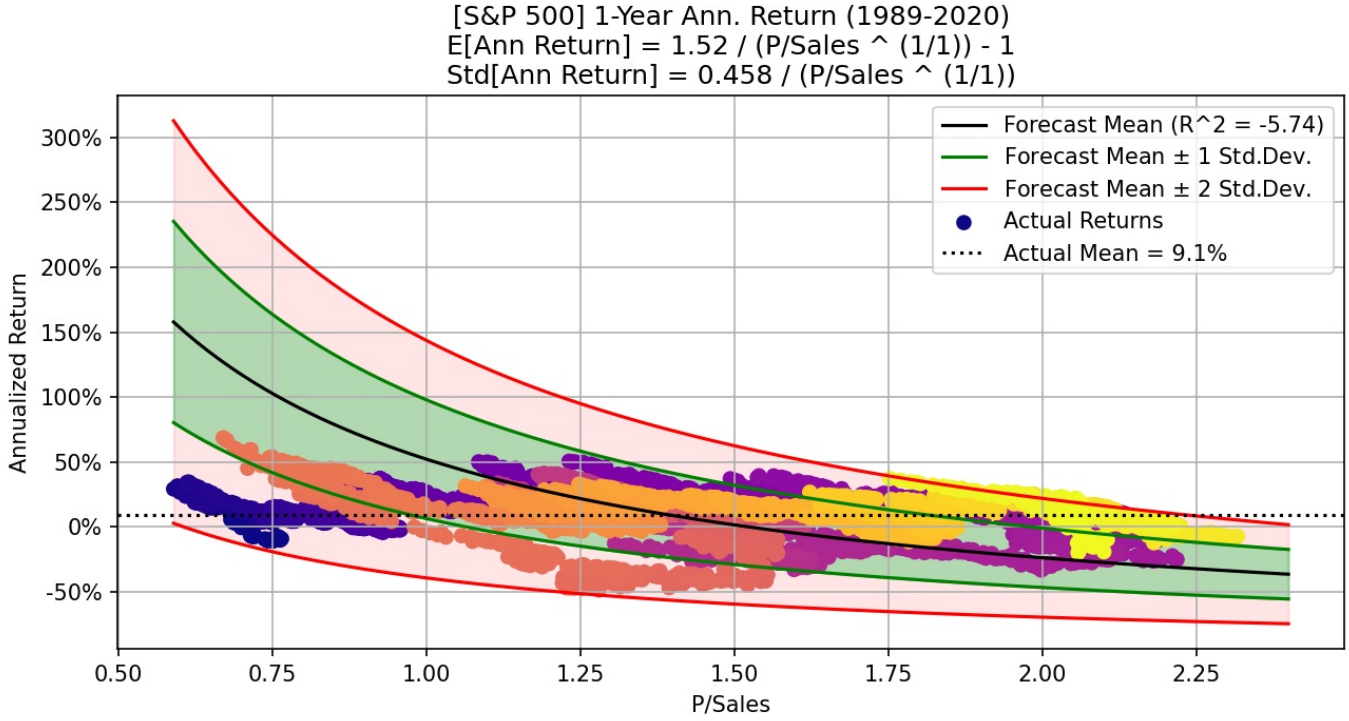



Также, как и в первом случае, в промежутке в год всё достаточно плохо - отрицательный R и большие ошибки. Но, в отличие от первого случая, в промежутках на 5 и 8 лет ситуация сильно не улучшается - ошибка действительно становится меньше, R растёт, но прогноз всё равно очень слабый.

На самом деле, этот алгоритм может неплохо работать и с пакетами акций, например, с S&P 500

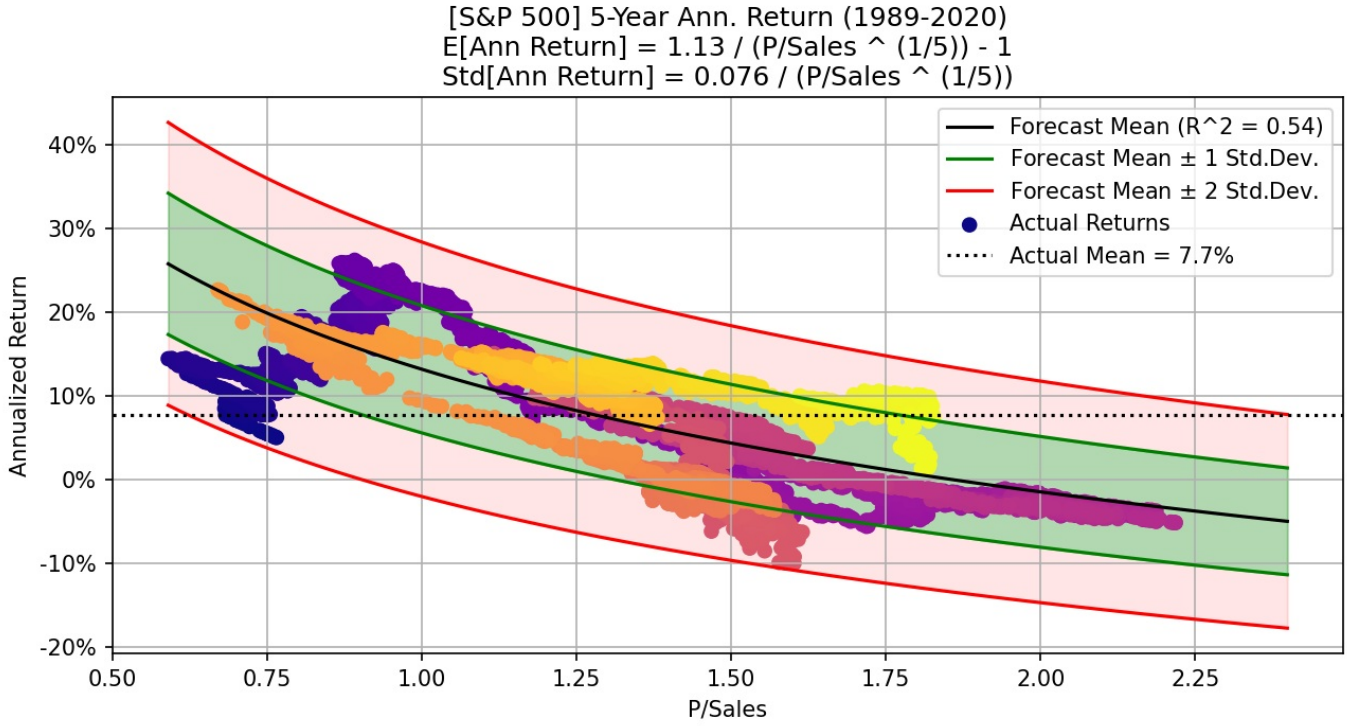
In [ ]:
```
ticker = 'S&P 500'

df_SP_500 = load_stock_data(ticker, dividend_TTM=True, earnings=False)

plot = plot_ann_returns(years=1, ticker=ticker, df=df_SP_500, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    30.2%           11.5%           1.00e+00
MSE:    1.67e-01        2.49e-02        1.00e+00
R^2:    -5.74
```



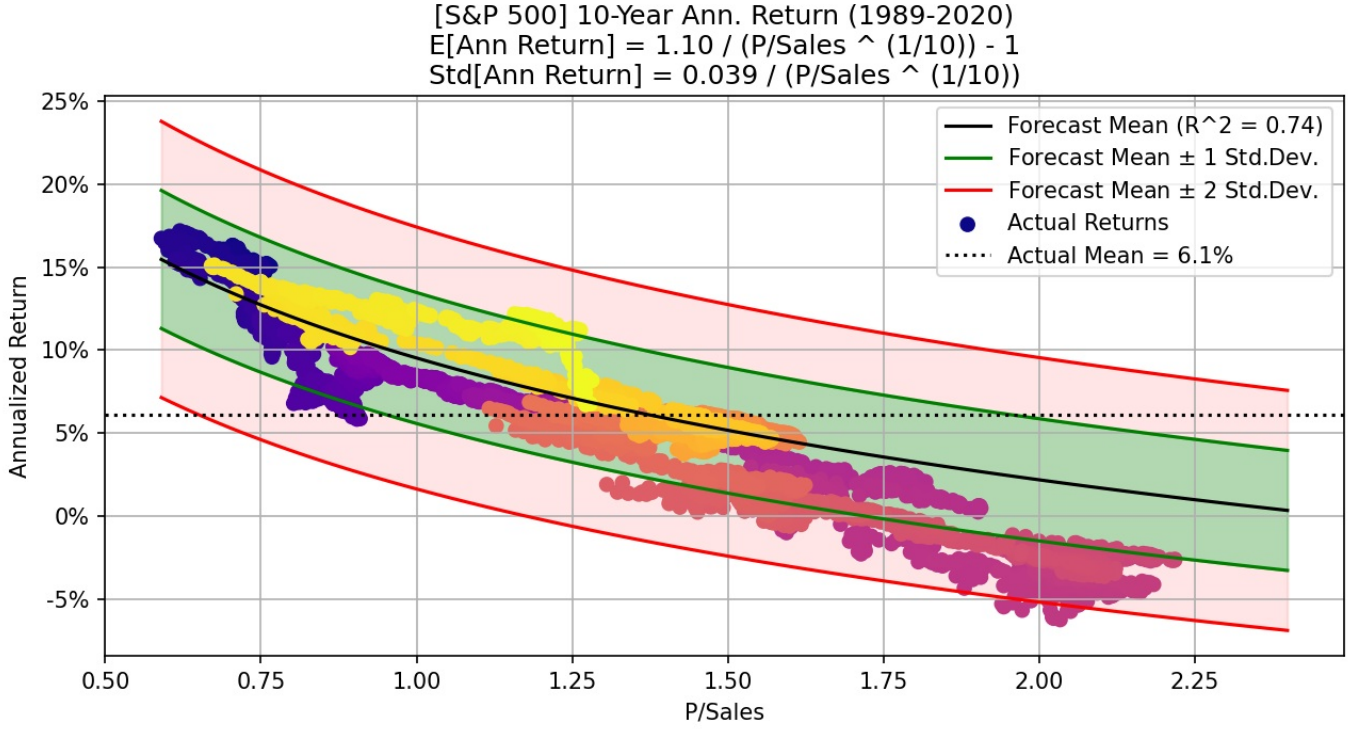

In [ ]:
```
plot = plot_ann_returns(years=5, ticker=ticker, df=df_SP_500, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    4.7%            7.0%            0.00e+00
MSE:    3.08e-03        6.77e-03        0.00e+00
R^2:    0.54
```




In [ ]:
```
plot = plot_ann_returns(years=10, ticker=ticker, df=df_SP_500, print_stats=True)
```

```
        Forecast        Baseline        p-value
================================================
MAE:    1.8%            3.6%            0.00e+00
MSE:    5.78e-04        2.21e-03        0.00e+00
R^2:    0.74
```




## Выводы по работе

- Наша теоретическая модель подтвердилась - формула даёт характерные кривые, хорошо описывающие полученные данные, и показывает, что при высоких P/Sales темпы роста акции действительно ниже.
- Обратная зависимость P/Sales и доходности. На всех рассмотренных горизонтах (1, 5 и 10 лет) наблюдается чёткая отрицательная связь между мультипликатором цены к выручке (P/Sales) на момент покупки и последующей среднегодовой доходностью: чем ниже P/Sales, тем выше доходность.
- Усиление эффекта с ростом горизонта инвестирования. Корреляция между P/Sales и доходностью становится сильнее по мере увеличения срока удержания акций. Для 5- и 10-летних периодов эффект более выражен, чем для однолетних. Также

заметим, что можно рассматривать не только определённый момент (прогноз спустя год, спустя 3 года), но и прогноз на промежутке времени - например в период 2-4 года, 4-6 лет. Подробнее об этом можно почитать в Appendix

- Использование P/Sales как базового фундаментального критерия позволяет сконструировать портфель с повышенным долгосрочным потенциалом доходности без сложных моделей и машинного обучения.

Источники и литература:


- Pedersen, M. E. H. (2020). Long-Term Stock Forecasting. Hvass Labs.
- Pedersen, M. E. H. (2021). Simple Portfolio Optimization That Works!
- Серия статей, поясняющая работу рынка