



מבוא למדע הנתונים

ד"ר אריה יעקבי

"The Sexiest Job of the 21st Century"
(Davenport and Patil)

סמסטר ב' תשפ"ב

במצגת זו השתמשתי בין השאר בחומרים של ד"ר אורית גולדמן

אפליקציה פשוטה לאופטימיזציה של תיק מניות

DATA CLEANING AND SIMPLE FINTECH ANALYSIS

אופטימיזציה פשוטה של תיק מניות

מה נלמד בשיעור זה

❖ כיצד מורידים נתוני מניות באמצעות פייתון

❖ כיצד נחקור את טיב נתוני המניות

❖ כיצד נבצע אופטימיזציה על תיק המניות

ספריות שנשתמש בהן

לא לשכוח להתקין ספריות אלה
באמצעות:

```
Pip install pandas_datareader
Pip install plotly
Pip install matplotlib
Pip install seaborn
```

נביא את הספריות המתאימות ע"י מיקום הפקודות הבאות בראש הקובץ

```
from pandas_datareader.data import DataReader
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
```

הספרייה

pandas_datareader

הספרייה pandas_datareader מאפשרת
לספרייה Pandas להגיע למקורות מידע
שונים.

הפונקציות מ- pandas_datareader.data
ו- pandas_datareader.wb מחלצות נתונים
ממקורות אינטרנט שונים לתוך pandas
DataFrame

❖ נכון לעכשיו המקורות הבאים נתמכים:

- ❖ [Tiingo](#)
- ❖ [IEX](#)
- ❖ [Alpha Vantage](#)
- ❖ [Econdb](#)
- ❖ [Enigma](#)
- ❖ [Quandl](#)
- ❖ [St. Louis FED \(FRED\)](#)
- ❖ [Kenneth French's data library](#)
- ❖ [World Bank](#)
- ❖ [OECD](#)
- ❖ [Eurostat](#)
- ❖ [Thrift Savings Plan](#)
- ❖ [Nasdaq Trader symbol definitions](#)
- ❖ [Stooq](#)
- ❖ [MOEX](#)
- ❖ [Naver Finance](#)
- ❖ [Yahoo Finance](#)

❖ אנו נשתמש באפליקציה זו להורדת נתוני מניות מ- [Yahoo Finance](#)

הספריה pandas_datareader

Yahoo Finance מספקת נתונים על שוק המניות

נקודות הקצה הבאות זמינות:

- yahoo : לאחר מחירי מניות יומיים (גבוה, פתיחה, סגירה, נפח, סגירה מותאמת)
- yahoo-actions : לאחר פעולות היסטוריות של תאגידים (דיבידנדים ופיצולי מניות)
- Yahoo dividends : לאחר דיבידנדים היסטוריים

```
import pandas_datareader.data as web
import datetime as dt
```

```
df = web.DataReader('GE', 'yahoo', start=('2019-10-09'=dne, '2019-09-10)
print(df.head())
```

| Date | High | Low | Open | Close | Volume | Adj Close |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|
| 2019-09-10 | 74.160004 | 71.199997 | 71.279999 | 73.120003 | 7827150.0 | 72.313095 |
| 2019-09-11 | 74.879997 | 72.480003 | 73.199997 | 74.879997 | 7136863.0 | 74.053665 |
| 2019-09-12 | 76.160004 | 73.760002 | 75.199997 | 74.080002 | 8514388.0 | 73.262497 |
| 2019-09-13 | 75.599998 | 73.120003 | 74.480003 | 74.720001 | 5698675.0 | 73.975334 |
| 2019-09-16 | 75.360001 | 73.360001 | 74.400002 | 75.040001 | 5718550.0 | 74.292145 |

```
start = dt.datetime(29, 1, 2010)
end = dt.datetime.today()
actions = web.DataReader('GOOG', 'yahoo-actions', start, end)
print(actions.head())
```

| | action | value |
|------------|--------|----------|
| 2015-04-27 | SPLIT | 0.997262 |
| 2014-03-27 | SPLIT | 0.4995 |

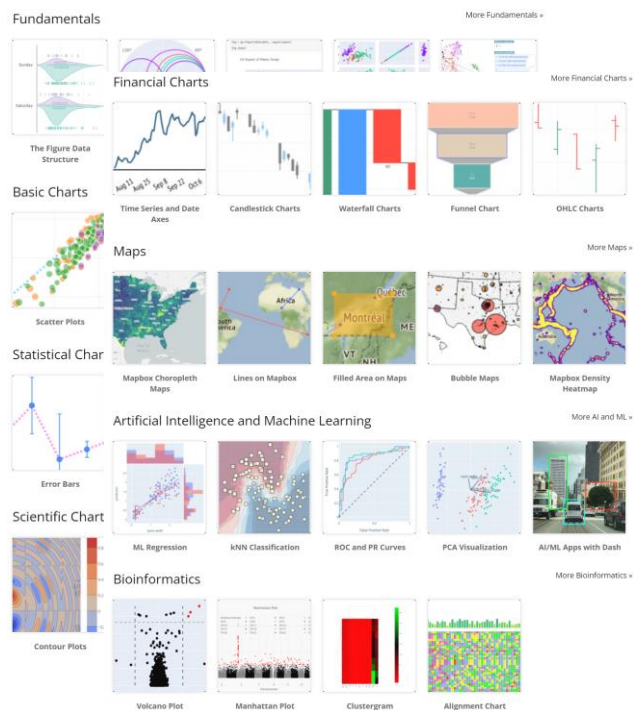
```
dividends = web.DataReader('IBM', 'yahoo-dividends', start, end)
print(dividends.head())
```

| | action | value |
|------------|----------|----------|
| 2022-05-09 | DIVIDEND | 1.65 |
| 2022-02-10 | DIVIDEND | 1.64 |
| 2021-11-09 | DIVIDEND | 1.64 |
| 2021-08-09 | DIVIDEND | 1.567878 |
| 2021-05-07 | DIVIDEND | 1.567878 |

הספריה plotly

הספריה plotly

ספריית הגרפיקה Plotly מייצרת גרפים אינטראקטיביים באיכות פרסום. דוגמאות לאופן שבו ניתן ליצור תרשימי קווים, תרשימי פיזור, תרשימי שטח, תרשימי עמודות, סרגלי שגיאה, תרשימי תיבה, היסטוגרמות, מפות חום, תרשימי משנה, צירים מרובים, תרשימי קוטב ותרשימי בועות.



הספריה plotly.express

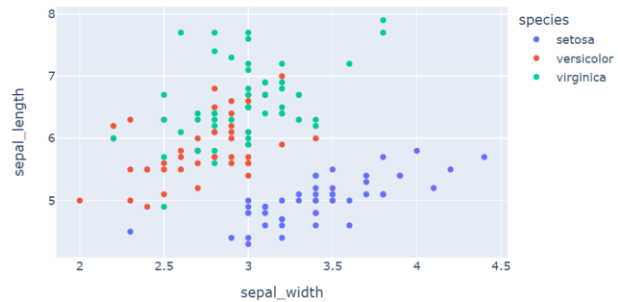
הספריה (plotly.express) מיובאת בדרך כלל כ-px, מכילה פונקציות שיכולות ליצור תמונות שלמות בבת אחת, והוא מכונה Plotly Express או Plotly Express

Plotly Express הוא חלק מובנה מספריית Plotly, והוא נקודת ההתחלה המומלצת ליצירת התמונות הנפוצות ביותר. כל פונקציה של Plotly Express משתמשת באובייקטים גרפיים באופן פנימי ומחזירה מופע של Figure: plotly.graph_objects

נייצר גרף נקודות פיזור עם קו מגמה באמצעות הספריה plotly

דוגמה זו מראה כיצד להשתמש בקו מגמה של plotly.express

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species")
fig.show()
```



הספריה matplotlib

הספריה matplotlib

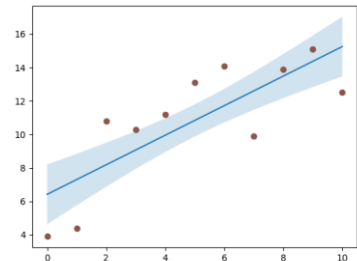
ספריית הגרפיקה הבסיסית והחשובה ביותר בפיתוח.

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

```
N = 21
x = np.linspace(0, 10, 11)
y = [3.9, 4.4, 10.8, 10.3, 11.2, 13.1, 14.1, 9.9, 13.9, 15.1, 12.5]
```

```
# fit a linear curve and estimate its y-values and their error.
a, b = np.polyfit(x, y, deg=1)
y_est = a * x + b
y_err = x.std() * np.sqrt(1/len(x) +
                        (x - x.mean())**2 / np.sum((x - x.mean())**2))
```

```
fig, ax = plt.subplots()
ax.plot(x, y_est, '-')
ax.fill_between(x, y_est - y_err, y_est + y_err, alpha=0.2)
ax.plot(x, y, 'o', color='tab:brown')
fig.show()
plt.waitforbuttonpress()
```



הספריה seaborn

הספריה seaborn

Seaborn היא ספריית הדמיית נתונים של Python המבוססת על matplotlib

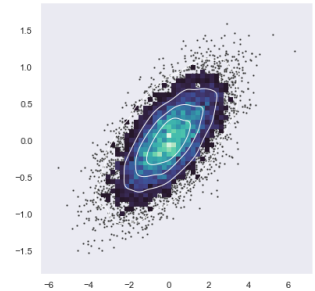
הספריה מספקת ממשק ברמה גבוהה לציוור גרפיקה סטטיסטית אטרקטיבית ואינפורמטיבית

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="dark")
```

```
# Simulate data from a bivariate Gaussian
n = 10000
mean = [0, 0]
cov = [(2, .4), (.4, .2)]
rng = np.random.RandomState(0)
x, y = rng.multivariate_normal(mean, cov, n).T
```

```
# Draw a combo histogram and scatterplot with density contours
f, ax = plt.subplots(figsize=(6, 6))
sns.scatterplot(x=x, y=y, s=5, color=".15")
sns.histplot(x=x, y=y, bins=50, pthresh=.1, cmap="mako")
sns.kdeplot(x=x, y=y, levels=5, color="w", linewidths=1)
```

```
f.show()
plt.waitforbuttonpress()
```



קריאת נתוני המניות מהאתר Yahoo Finanace ברשת הווב

בוא נקבל כמה נתונים עבור תגי (tickers) המניות שאנו רוצים לכלול בתיק שלנו. הספרייה pandas-datareader

מספקת שיטה למשוך נתוני מחירי מניות מהאינטרנט ולאחסן אותם ב-DataFrame

אם לא תעברו את תאריכי ההתחלה והסיום, מחירי המניות יינתנו עבור טווח התאריכים המלא הזמין. שים לב שטווח התאריכים עשוי להיות שונה עבור מניות שונות בהתאם למועד הרשימה.

```
# Read the stocks from the web (Yahoo Finance)
```

```
start_date = '2013/01/01'
end_date = '2022/4/17'
```

```
tickers = ['MA', 'FB', 'V', 'AMZN', 'JPM', 'BA']
stocks_df = DataReader(tickers, 'yahoo', start=start_date,
end=end_date)['Adj Close']
print(stocks_df.head())
```

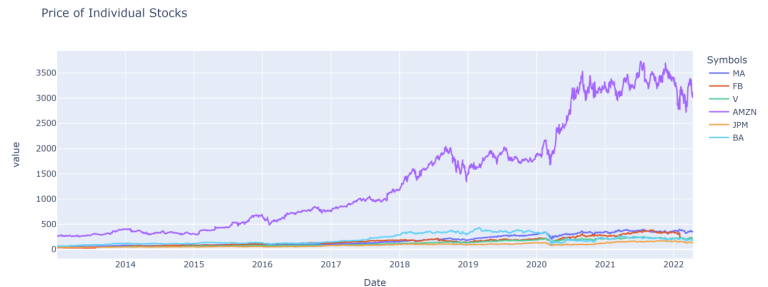
| Symbols | Date | MA | FB | V | AMZN | JPM | BA |
|-----------|-----------|------------|-----------|-----------|-----------|------------|----|
| 64.583885 | 34.679985 | 257.309998 | 36.372139 | 28.000000 | 48.250706 | 2013-01-02 | |
| 64.919052 | 34.610092 | 258.480011 | 36.400234 | 27.770000 | 48.319725 | 2013-01-03 | |
| 65.103401 | 35.223557 | 259.149994 | 36.697517 | 28.760000 | 48.317844 | 2013-01-04 | |
| 63.796150 | 35.262383 | 268.459991 | 36.959698 | 29.420000 | 49.155178 | 2013-01-07 | |
| 62.120140 | 35.332272 | 266.380005 | 37.303810 | 29.059999 | 48.993401 | 2013-01-08 | |

גרף קו של המחירים

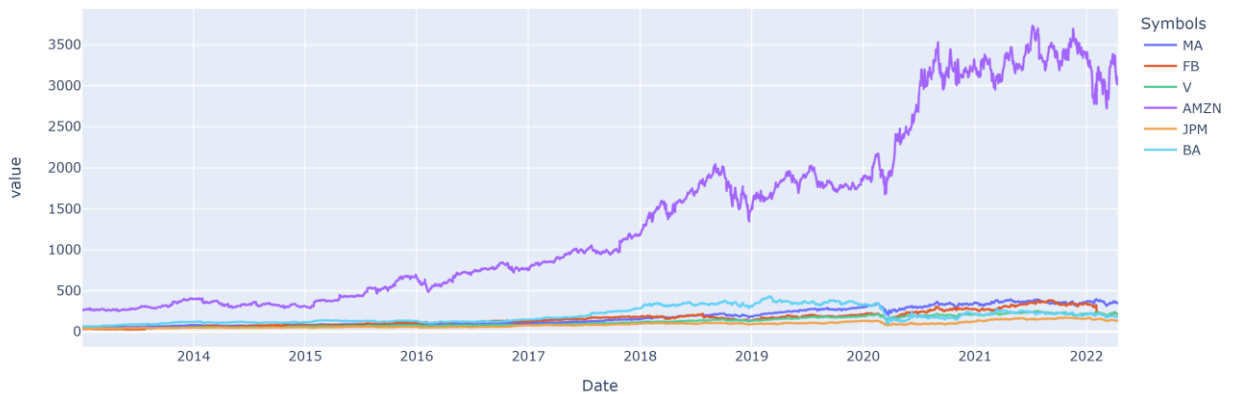
כאן אנו משתמשים בספריית plotly כדי ליצור גרף קו של המחירים.

שלא כמו בספריית Matplotlib זה גרף אינטראקטיבי ותוכלו לרחף עם העכבר בהצביעו על הקווים על מנת לקבל פרטים.

```
#Plot the stocks
fig_price = px.line(stocks_df, title='Price of Individual Stocks')
fig_price.show()
```



Price of Individual Stocks



כפי שניתן לראות לעיל, נראה שאמזון שולטת בקנה המידה של הגרף מכיוון שהמחיר המוחלט של המניה גבוה מאוד. הגרפים של כל המניות האחרות שטוחים. גרף כזה אינו שימושי במיוחד כדי להשוות את הביצועים היחסיים של המניות. כדי לטפל בזה, בואו נראה כיצד נוכל למדוד טוב יותר את הביצועים של מניה על ידי בחינת המושגים של תשואות יומיות ותנודתיות

חישוב תשואות יומיות

Daily Returns

התשואות היומיות של מניה הן הרווח (או ההפסד) החלקי ביום נתון ביחס ליום הקודם, הוא ניתן על ידי:

$$\frac{\text{Current day's close price} - \text{previous day's close price}}{\text{previous day's close price}}$$

מחיר הסגירה של היום הנכחי פחות מחיר סגירה של היום הקודם לחלק למחיר הסגירה של היום הקודם

מכיוון שזהו ערך יחסי, הוא מספק השוואה הוגנת יותר בין תשואות מניות ללא קשר למחירי המניות האבסולוטיים.

ניתן להשתמש במתודה `pct_change()` בכדי לקבל את התשואות היומיות ביעילות.

```
daily_returns = stocks_df.pct_change().dropna()
print("\nDaily returns")
print(daily_returns.head())
```

המתודה `pct_change()` מחשבת את השינוי באחוזים בין האלמנט הנוכחי לאלמנט קודם. שימו לב לשימוש במתודה `dropna()` שדואגת להוריד את כל השורות עם ערכי `<NA>`

The daily returns

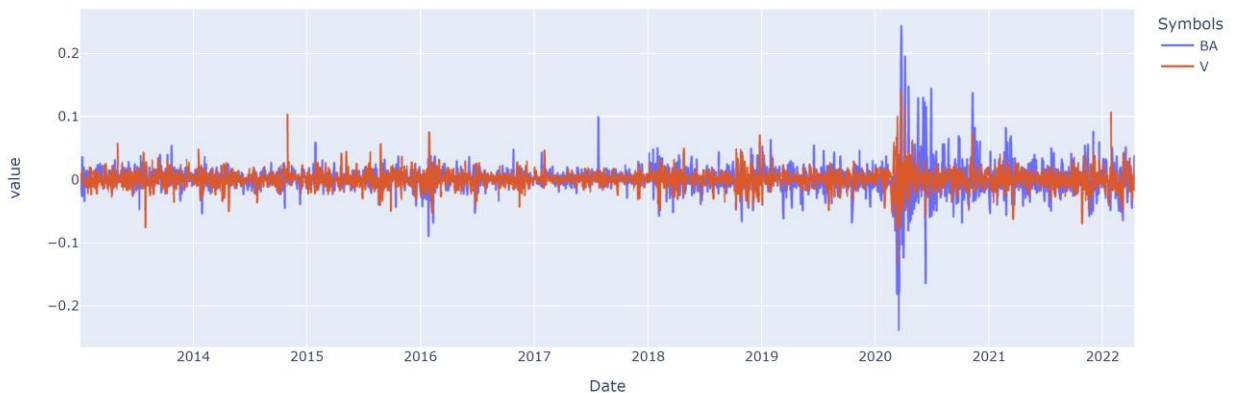
| Symbols | MA | FB | V | AMZN | JPM | BA |
|---------|----------|----------|----------|----------|----------|----------|
| Date | 0.005190 | 0.002015 | 0.004547 | 0.000772 | 0.008214 | 0.001430 |
| | 0.002840 | 0.017725 | 0.002592 | 0.008167 | 0.035650 | 0.000039 |
| | 0.020080 | 0.001102 | 0.035925 | 0.007144 | 0.022949 | 0.017330 |
| | 0.026271 | 0.001982 | 0.007748 | 0.009310 | 0.012237 | 0.003291 |
| | 0.035479 | 0.000659 | 0.000113 | 0.015248 | 0.052650 | 0.028137 |

בואו נייצר גרף קוי של התשואות היומיות של 2 מהמניות, בואינג (BA) וויזה (V), לאורך כל תקופת הזמן.

```
fig = px.line(daily_returns[['BA', 'V']], title='Daily Returns')
fig.show()
```



Daily Returns



אנו רואים שהן נוטות לתנודות עדינות סביב 0. יש לציין שהתנודות הן הרבה יותר גדולות בתקופה של תנודתיות גבוהה (כלומר במהלך התרסקות קוביד במרץ 2020).

חישוב התנודתיות Daily - Volatility

התנודתיות היומית היא ההפרש הממוצע בין התשואה ביום נתון לבין התשואה הממוצעת לאורך תקופת הזמן.

מבחינה מתמטית, זוהי רק סטיית התקן של התשואות היומיות.

תנודתיות היא אחד ממדדי הסיכון בכך שהשקעות בתנודתיות גבוהה יכולות לשאת סיכון גדול יותר.

The Daily Volatility

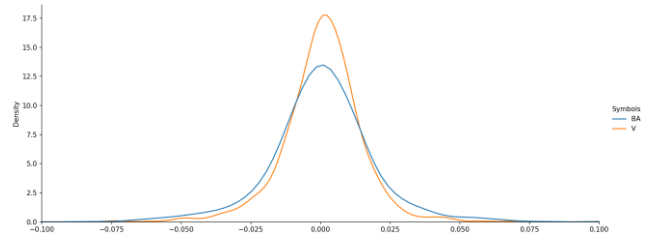
Symbols

| | |
|--------|----------|
| MA | 0.017019 |
| FB | 0.022239 |
| V | 0.015621 |
| AMZN | 0.019330 |
| JPM | 0.016772 |
| BA | 0.023948 |
| dtype: | float64 |

כאן אנו רואים כי ל-BA יש תנודתיות מעט גבוהה יותר, בהשוואה לזו של V.

```
print("\nDaily Volatility")
print(daily_returns.std())
```

```
sns.displot(data=daily_returns[['BA', 'V']], kind = 'kde', aspect = 2.5)
plt.xlim(-0.1, 0.1)
```



כאשר אנו משווים את גרף הצפיפות של התשואות היומיות שלהם, אנו יכולים לראות של-V יש עקומה צרה יותר עם שיא גבוה יותר, בעוד ל-BA יש עקומה רחבה יותר המעידה על סטיית תקן גבוהה יותר ומכאן התנודתיות.

ציור של התשואות המצטברות Plot Individual Cumulative Returns

ניתן לחשב בקלות את התשואות המצטברות של המניה על ידי הוספת 1 לתשואות היומיות ולקחת את המוצר המצטבר על פני כל התקופה.

כאן אנו מתווים את התשואות המצטברות של מניות החל מהשקעה ראשונית של \$100 (כלומר כמה השקעה של \$100 בהתחלה בכל אחת מהמניות תיתן תשואה לאורך תקופת הזמן?)

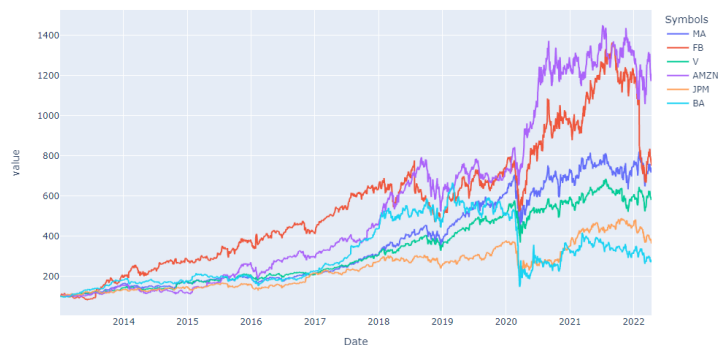
זוהי השוואה הוגנת לביצועי המניות.

```
def plot_cum_returns(data, title):
```

```
    daily_cum_returns = 1 + data.dropna().pct_change()
    daily_cum_returns = daily_cum_returns.cumprod()*100
    fig = px.line(daily_cum_returns, title=title)
    return fig
```

```
fig_cum_returns = plot_cum_returns(stocks_df,
    'Cumulative Returns of Individual Stocks Starting with $100')
fig_cum_returns.show()
```

Cumulative Returns of Individual Stocks Starting with \$100



מטריצת קורלציות בין המניות Correlation Matrix between Stocks

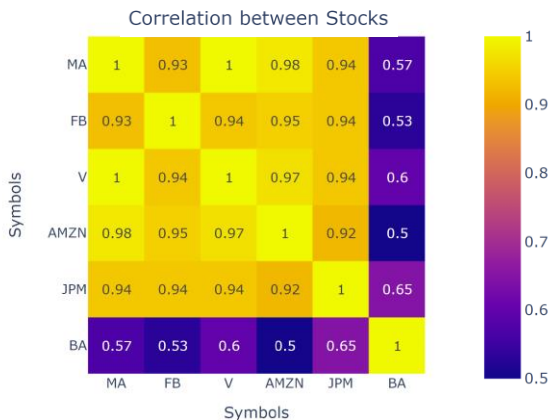
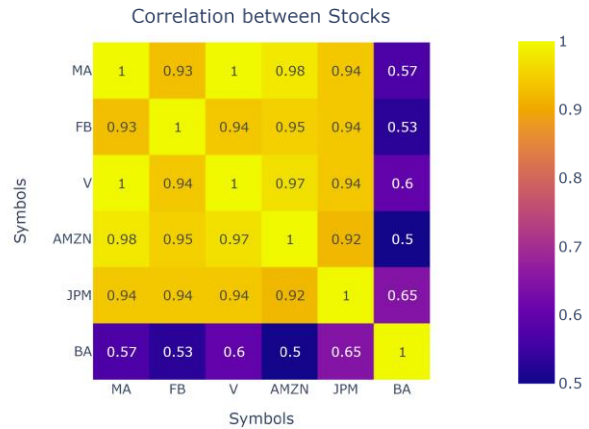
מטריצת הקורלציות נותנת לנו את מקדמי המתאם (קורלציה) בין כל זוג מניות.

מקדמי מתאם הם אינדיקטורים לחוזק הקשר הליניארי בין שני משתנים שונים. זהו ערך מ-0 עד 1, כאשר 1 מציין את הקשר החזק ביותר.

- ערך חיובי מצביע על קשר חיובי, כלומר שני המשתנים נעים יחד.
- ערך שלילי מציין קשר שלילי, כלומר שני המשתנים קשורים ביחס הפוך.
- ערך אפס מציין שאין קשר.

ל-Pandas יש שיטת `corr()` שמאפשרת לנו ליצור את המטריצה.

```
corr_df = stocks_df.corr().round(2) # 2 decimal places
fig_corr = px.imshow(corr_df, text_auto=True, title = 'Correlation between Stocks')
fig_corr.show()
```



באופן כללי (אם כי לא תמיד), מחירי המניות נוטים לנוע יחדיו (עלייה בשוק השוורים, ירידה בשוק הדובי), ולכן קרוב לוודאי שהמתאם יהיה חיובי כפי שמוצג להלן.

שימו לב גם לאופן שבו בואינג (BA) נמצאת בקורלציה חלשה למניות אחרות במהלך תקופה זו. ייתכן שהסיבה לכך היא שהוא בתעשייה שונה מאוד, או בגלל החדשות הרעות שפקדו אותו בשנים האחרונות.

ושימו לב כיצד החברות הדומות מאסטקארד (MA) וויזה (V) נמצאות בקורלציה כמעט מושלמת

חלק מתקדם שמטפל באופטימיזציה של תיק המניות ודורש ידע גבוה יותר

שימוש בספריית PYPORTFOLIOOPT שמיועדת לאופטימיזציה של תיקי
מניות בשיטות מתקדמות

Expected Returns and Covariance Matrix

באופן דומה למטריצת הקורלציות, ישנה
מטריצת הקו-ואריאנס שמודדת אם מניות נעות
באותו כיוון (a positive covariance) או בכיוונים
מנוגדים (a negative covariance).

היא משמשת לחישוב התגודתיות של כל תיק
המניות, וכמו כן, היא משמשת מנהלי התיקים כדי
לנמק את הסיכון של כל פורטפוליו תיק המניות.

ספריית pyportfolioopt מקלה על קבלת המטריצה הזו,
וגם את התשובה השנתית הממוצעת של כל
המניות. אנו זקוקים לתוצאות אלה בתור כקלט על
מנת למצוא את פורטפוליו המניות האופטימלי.

```
mu = expected_returns.mean_historical_return(stocks_df)
S = risk_models.sample_cov(stocks_df)
print("\nThe volatility of the whole portfolio of stocks")
print(mu)
```

Symbols

| | |
|------|----------|
| MA | 0.241051 |
| FB | 0.242676 |
| V | 0.209486 |
| AMZN | 0.304666 |
| JPM | 0.149305 |
| BA | 0.118103 |

dtype: float64

תשואות תיקים, שיעור ללא סיכון, תגודתיות וִיחַס Sharpe

בדומה למניות, גם לתיק יש תשואה צפויה ותגודתיות. התשואה הצפויה שלו מתקבלת ממשקל כל מניה כפול התשואה הצפויה של המניה למעלה ולאחר מכן סכימה שלהם. ניתן לחשב את התגודתיות שלו ממטריצת השונות כפי שהוזכר קודם לכן.

יחס שארפ

יחס ה- Sharpe של תיק מודד את התשואה שלו ביחס לשיעור חסר הסיכון (למשל ריבית האוצר האמריקאי) והסיכון שלו (סטיית תקן). זה ניתן על ידי:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

כאשר

R_p היא התשואה על תיק (פורטפוליו) המניות

R_f הוא השיעור ללא סיכון

σ_p היא סטיית התקן של התיק

הצגה של ה- Efficient Frontier ו- Sharpe Ratio המקסימלי

❖ The Modern Portfolio Theory (MPT) היא תיאוריה שמציגה מודל לפיתוח תיק נכסים הממקסם את התשואה הצפויה עבור רמת סיכון נתונה.

❖ התיאוריה מניחה שהאדם הממוצע הוא שונא סיכון. מכאן שעבור רמה נתונה של תשואה צפויה, התיק הפחות מסוכן תמיד מועדף.

❖ מערך התיקים האופטימליים המציעים את הסיכון הנמוך ביותר (תגודתיות) עבור רמה נתונה של תשואה צפויה יוצר את ה- Efficient Frontier. הוא מיוצג על ידי עקומה בגרף תשואה לעומת תגודתיות.

❖ כמובן, תיק היחס המקסימלי של Sharpe נמצא על הגבול היעיל. הגבול היעיל ותיק היחס המקסימלי של Sharpe מצוינים גם הם בגרף.

❖ כדי לייצג הכל בצורה ויזואלית, נבנה פונקציה שמייצרת 1000 תיקים עם המניות שלנו, מייצרת למניות משקלים אקראיים ומתארת את התשואות והתגודתיות שלהן.

לסיכום

❖ מה למדנו היום?

❖ כיצד מורידים נתוני מניות באמצעות פייתון

❖ כיצד נחקור את טיב נתוני המניות

❖ כיצד נבצע אופטימיזציה על תיק המניות

חשוב!

למדו את המושגים החדשים הם
חשובים

להתראות בשבוע הבא!

