

עבודה בקורס סדנת מומחים

אלינה בוחנוב

341349819

שאלה 1.

בכל סעיף מתחילים בספריות ובקריאת הקבצים.

```
import pandas as pd
import dateutil
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import Kmeans
```

```
afi = pd.read_csv(r"/home/elina/Documents/sadnat mumhim/stocks/afi.csv", skiprows=2 )
airportcity = pd.read_csv(r"/home/elina/Documents/sadnat mumhim/stocks/airportcity.csv", skiprows=2 )
ובו'.
```

(א)

כדי למצוא את הממוצע כותבים את הפונקציה `data_ret_mean` ומשתמשים בה עבור כל מניה ומניה.

```
def data_ret_mean(df):
    df['Date'] = pd.to_datetime(df.Date)
    df.sort_values(by=['Date'], inplace=True, ascending=True)
    df = df.set_index(['Date'])
    df_returns = df['Adjusted Closing Price'].pct_change().mean()
    return df_returns
```

על מנת למצוא סטיית התקן כותבים את הפונקציה `data_ret_var`.

```
def data_ret_var(df):
    df['Date'] = pd.to_datetime(df.Date)
    df.sort_values(by=['Date'], inplace=True, ascending=True)
    df = df.set_index(['Date'])
    df_variance = df['Adjusted Closing Price'].pct_change().std()
    return df_variance
```

```
afi_returns = data_ret_mean(afi)
afi_variance = data_ret_var(afi)
```

עושים את זה עבור כל מניה.

```
Company Returns Variance
afi 0.003082 0.066578
airportcity 0.003086 0.066827
alony 0.001449 0.044826
amot 0.002013 0.049980
azrieli 0.001723 0.049287
```

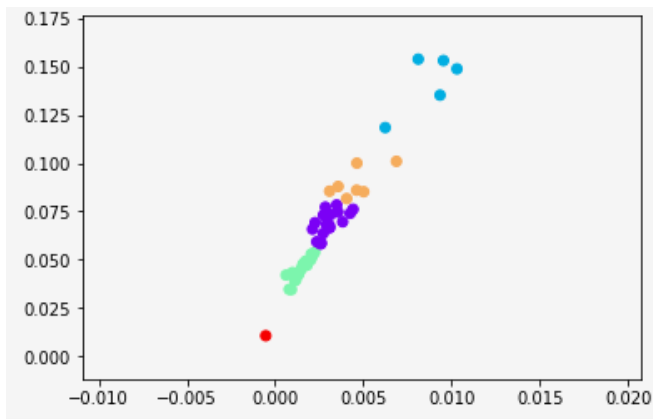
מוסיפים את התוצאות לאותו `data - frame`.

K means clustering algorithm ותוצאות.

```
#k-means
X = ret_var.values
kmeans = KMeans(n_clusters = 5).fit(X)
```

```
centroids = kmeans.cluster_centers_
plt.scatter(X[:,0],X[:,1], c = kmeans.labels_, cmap="rainbow")
plt.show()
```

```
#getting the cluster number of each company
Company = pd.DataFrame(ret_var.index)
cluster_labels = pd.DataFrame(kmeans.labels_)
df = pd.concat([Company, cluster_labels],axis = 1)
print(df.head(20))
```



Companies and their clusters.

	Company	0			
0	afi	4	25	hilan	4
1	airportcity	4	26	icl	2
2	alony	0	27	indus	2
3	amot	0	28	israel con	2
4	azrieli	0	29	isramco	0
5	bayside	0	30	leumi	0
6	bazan	2	31	lvpsn	1
7	bezeq	4	32	matrix	4
8	big	0	33	mega_or	4
9	clal insun	4	34	melisron	4
10	compuge	1	35	mizrahi	0
11	delek dri	4	36	nice	4
12	delek gro	2	37	nova	4
13	discount	0	38	partner	1
14	elbit	4	39	paz	0
15	electra	0	40	phoenix	4
16	energix	2	41	reit1	0
17	enlight e	2	42	sapiens	2
18	equital	4	43	sella	0
19	fibi	3	44	shikun	4
20	fibi holdi	0	45	shufersal	2
21	formula	4	46	strauss	0
22	gazit	0	47	summit	4
23	hapoalim	0	48	teva	1
24	harel	4	49	tower	1

(ב)

משתמשים באותה פונקציה מהסעיף הקודם כדי למצוא את הממוצע.

אחרי זה מייצרים את הוקטור של משקלות, מחשבים את ה - cov ואת ה - var כדי לחשב את ה - beta.

```
#weights
```

```
weighted_returns = (wts * market_ret)
market_ret = weighted_returns.sum(axis=1)
market_ret = market_ret.to_frame()
market_ret.columns = ['Market Return']
f_column = market_ret['Market Return']
#returns of every stock and the market
weighted_returns = pd.concat([weighted_returns,f_column], axis = 1)
#cov
cov_matrix = weighted_returns.cov()
covar = cov_matrix.iloc[:, -1]
covar = covar.drop(covar.index[len(covar)-1])
covar = covar.to_frame()
#var
var_returns = market_ret.var()
```

על מנת לקבל את התשואות השנתיות כותבים את הפונקציה

```
def dataset_year(df):
```

```

df.sort_values(by=['Date'], inplace=True, ascending=True)
df['Date'] = df['Date'].apply(lambda x: dateutil.parser.parse(x))
year = df["Date"].dt.to_period("Y")
agg = df.groupby([year])
id_df = agg['Adjusted Closing Price'].apply(lambda x: pd.Series(x.values)).unstack()
id_df = id_df.rename(columns={i: 'Adjusted Closing Price{}'.format(i + 1) for i in range(id_df.shape[1])})
id_df = id_df.pct_change(244, axis=1)
id_df = id_df.T
id_df = id_df.tail(1)
return id_df

```

עושים את זה עבור כל מניה.

מוסיפים את התוצאות לאותו data - frame
מקבלים 6 עמודות (d=6).

Date	2013	2014	...	2017	2018
Company			...		
afi	0.541372	-0.042839	...	0.298182	-0.038408
airportcity	0.755219	0.063409	...	0.191954	0.062211
alony	0.053222	0.109003	...	0.254549	-0.031531
amot	0.170671	0.147135	...	0.368404	-0.080793
azrieli	0.218788	0.130356	...	0.185819	-0.044621
bayside	0.349780	0.134962	...	0.316002	-0.049382
bazan	-0.378373	-0.046040	...	0.315215	0.117088
bezeq	0.630842	0.327038	...	-0.219027	-0.276277
big	0.383535	0.177523	...	-0.020099	-0.120707
clal_insurance	0.233521	-0.185683	...	0.268866	-0.161814
compugen	0.612083	0.021010	...	-0.555476	-0.059249
delek_drill_l	0.188396	-0.211906	...	-0.201817	0.011398

K means clustering algorithm ותוצאות.

```

#k-means
X = ret.values
kmeans = KMeans(n_clusters=5).fit_predict(X, y=None, sample_weight=None)
ret['cluster'] = kmeans

```

Company	cluster		
afi	2	hilan	0
airportcity	0	icl	3
alony	2	indus	2
amot	2	israel cor	3
azrieli	2	isramco	2
bayside	2	leumi	2
bazan	2	lvpsn	3
bezeq	0	matrix	2
big	0	mega or	0
clal_insurance	2	melisron	0
compugen	0	mizrahi	2
delek_drill_l	2	nice	2
delek_group	0	nova	2
discount	2	partner	2
elbit	0	paz	2
electra	2	phoenix	2
energix	1	reit1	2
enlight_energy	1	sapiens	0
equital	2	sella	2
fibi	2	shikun	2
fibi_holdings	2	shufersa	2
formula	0	strauss	2
gazit	2	summit	0
hapoalim	2	teva	0
harel	2	tower	4

(ד)

על מנת לקבל את התשואות החודשיות כותבים את הפונקציה `dataset_month`.

```
def dataset_month(df):  
    df.sort_values(by=['Date'], inplace=True, ascending=True)  
    df['Date'] = df['Date'].apply(lambda x: dateutil.parser.parse(x))  
    df_month = df["Date"].dt.to_period("m")  
    df_agg = df.groupby([df_month])  
    df_df = df_agg['Adjusted Closing Price'].apply(lambda x: pd.Series(x.values)).unstack()  
    df_df = df_df.rename(columns={i: 'Adjusted Closing Price{}'.format(i + 1) for i in range(df_df.shape[1])})  
    df_df = df_df.pct_change(24, axis=1)  
    df_df = df_df.T  
    df_df = df_df.tail(1)  
    return df_df
```

עושים אותו תהליך כמו בסעיפים הקודמים ומוסיפים את התוצאות לאותו `data - frame`.

Date	2013-01	2013-02	...	2018-11	2018-12
Company					
afi	0.081344	0.046049	...	0.125843	-0.009711
airportcity	-0.053238	0.010441	...	0.047253	0.099593
alony	-0.026754	-0.052684	...	0.009327	0.104935
amot	-0.015128	0.018438	...	-0.087570	-0.020906
azrieli	0.004651	0.012508	...	0.017320	0.043618
[5 rows x 72 columns]					

מקבלים 72 עמודות ($d=72$).

K means clustering algorithm ותוצאות.

```
#k-means  
X = df.values  
kmeans = KMeans(n_clusters=5).fit_predict(X, y=None, sample_weight=None)  
ret['cluster'] = kmeans
```

Company	cluster		
afi	1	icl	2
airportcity	1	indus	1
alony	1	israel cor	2
amot	1	isramco	1
azrieli	1	leumi	1
bayside	1	lvpsn	2
bazan	1	matrix	1
bezeq	1	mega or	0
big	1	melisron	1
clal insurance	0	mizrahi	1
compugen	3	nice	1
delek drill l	1	nova	0
delek group	1	partner	1
discount	1	paz	1
elbit	1	phoenix	0
electra	1	reit1	1
energix	0	sapiens	1
enlight energy	0	sella	1
equital	1	shikun	1
fibi	1	shufersa	0
fibi holdings	0	strauss	1
formula	1	summit	1
gazit	1	teva	3
harel	0	tower	4
hilan	1		

קלסטרים הם האוסף של נקודות נתונים המצטברות יחד על סמך קווי דמיון מסוימים. אפשר לראות שבכל סעיף מקבלים תוצאות קלסטריזציה שונות, מפני שמתבססים על פרמטרים שונים.

שאלה 2

(א)

בחרתי את הסקטור של הבנקים: discount, hapoalim, leumi, mizrahi.
קודם כל עושים מיון לפי התהריך, ככה יהיה יותר נוח לעבוד עם הנתונים.

```
discount['Date']=pd.to_datetime(discount.Date)
discount.sort_values(by=['Date'], inplace=True, ascending=True)
discount = discount.set_index(['Date'])
```

וכו'.

אחר כך מחשבים את התשואות, עושים את זה באמצעות pct_change().

```
#daily returns
discount_daily_returns = discount['Adjusted Closing Price'].pct_change()
hapoalim_daily_returns = hapoalim['Adjusted Closing Price'].pct_change()
leumi_daily_returns = leumi['Adjusted Closing Price'].pct_change()
mizrahi_daily_returns = mizrahi['Adjusted Closing Price'].pct_change()
```

מייצרים את ה-data-frame עם כל התשואות.

```
all_returns = pd.concat([discount_daily_returns[:,],hapoalim_daily_returns[:,], leumi_daily_returns[:,],
mizrahi_daily_returns], axis=1, keys=['discount_dr', 'hapoalim_dr', 'leumi_dr', 'mizrahi_dr'])
```

על מנת למצוא כותבים את הפונקציה crosscorr.

```
def crosscorr(datax, datay, lag=0, wrap=False):
```

```
    """ Lag-N cross correlation.
```

```
    Shifted data filled with NaNs
```

```
    Parameters
```

```
    -----
```

```
    lag : int, default 0
```

```
    datax, datay : pandas.Series objects of equal length
```

```
    Returns
```

```
    -----
```

```
    crosscorr : float
```

```
    """
```

```
    if wrap:
```

```
        shiftedy = datay.shift(lag)
```

```
        shiftedy.iloc[:lag] = datay.iloc[-lag:].values
```

```
        return datax.corr(shiftedy)
```

```
    else:
```

```
        return datax.corr(datay.shift(lag))
```

תוצאות:

	r=0	r=1	r=2	r=3	r=4
discount/discount	1.000000	-0.149100	-0.028256	-0.094132	-0.126771
discount/hapoalim	0.609616	-0.036767	-0.022405	-0.034408	-0.079993
discount/leumi	0.760012	-0.047199	-0.026541	-0.072656	-0.087985
discount/mizrahi	0.584486	-0.111665	0.015667	-0.017811	-0.069994

	r=0	r=1	r=2	r=3	r=4
hapoalim/hapoalim	1.000000	-0.108394	-0.102887	-0.099796	-0.075200
hapoalim/discount	0.609616	-0.062134	-0.068071	-0.046315	-0.123823
hapoalim/leumi	0.706167	-0.061143	-0.082336	-0.053739	-0.092368
hapoalim/mizrahi	0.668712	-0.097574	-0.027485	-0.046961	-0.105825

	r=0	r=1	r=2	r=3	r=4
leumi/leumi	1.000000	-0.127035	-0.082070	-0.073371	-0.125508
leumi/discount	0.760012	-0.107563	-0.071309	-0.040441	-0.120574
leumi/hapoalim	0.706167	-0.031250	-0.071466	-0.065582	-0.113767
leumi/mizrahi	0.605507	-0.102831	-0.009379	-0.018465	-0.095775

	r=0	r=1	r=2	r=3	r=4
mizrahi/mizrahi	1.000000	-0.127035	-0.082070	-0.073371	-0.125508
mizrahi/discount	0.760012	-0.107563	-0.071309	-0.040441	-0.120574
mizrahi/hapoalim	0.706167	-0.031250	-0.071466	-0.065582	-0.113767
mizrahi/leumi	0.605507	-0.102831	-0.009379	-0.018465	-0.095775

ניתן לראות שיש קורלציות משמעותיות בסקטור של הבנקים כאשר $r = 0$.

כאשר בוחרים r גדול יותר אין השפעה משמעותית בין בנק אחד לשני.

(ב)

עשים אותו תהליך אבל עם ערך מוחלט של התשואות.

	r=0	r=1	r=2	r=3	r=4
discount/discount	1.000000	0.237290	0.130762	0.117078	0.044793
discount/hapoalim	0.623249	0.212104	0.144803	0.105718	0.067666
discount/leumi	0.721475	0.201733	0.113763	0.054990	-0.002030
discount/mizrahi	0.631792	0.255073	0.104073	0.067959	0.006391

r = 0.01.

	r=0	r=1	r=2	r=3	r=4
hapoalim/hapoalim	1.000000	0.271473	0.156169	0.079608	0.026867
hapoalim/discount	0.623249	0.263494	0.158742	0.059993	0.013529
hapoalim/leumi	0.643376	0.219798	0.145480	0.048858	0.013023
hapoalim/mizrahi	0.686869	0.261045	0.130435	0.063653	0.011688

	r=0	r=1	r=2	r=3	r=4
leumi/leumi	1.000000	0.206546	0.151198	0.076194	0.066097
leumi/discount	0.721475	0.232194	0.142014	0.107259	0.091971
leumi/hapoalim	0.643376	0.198805	0.169813	0.113636	0.113118
leumi/mizrahi	0.567802	0.219496	0.135144	0.069492	0.083072

	r=0	r=1	r=2	r=3	r=4
mizrahi/mizrahi	1.000000	0.206546	0.151198	0.076194	0.066097
mizrahi/discount	0.721475	0.232194	0.142014	0.107259	0.091971
mizrahi/hapoalim	0.643376	0.198805	0.169813	0.113636	0.113118
mizrahi/leumi	0.567802	0.219496	0.135144	0.069492	0.083072

גם פה יש קורלציות משמעותיות כאשר $r = 0$.
נוסף לכך ניתן לראות שכל הקורלציות יותר גבוהות במקרה של הערך המוחלט של התשואות.

(ג)

בסעיף הזה בחרתי מניות של `big`, `hapoalim`, `bezeq`, `teva`.

	r=0	r=1	r=2	r=3	r=4
big/big	1.000000	0.214609	0.133233	0.059676	0.017962
big/hapoalim	0.606020	0.243445	0.104125	0.023000	-0.008821
big/bezeq	0.471231	0.132738	0.126251	0.022084	-0.031360
big/teva	0.253706	0.075710	0.020155	-0.035775	-0.051816

	r=0	r=1	r=2	r=3	r=4
hapoalim/hapoalim	1.000000	-0.108394	-0.102887	-0.099796	-0.075200
hapoalim/big	0.532914	0.105312	0.043390	-0.008394	-0.064671
hapoalim/bezeq	0.227093	-0.010583	-0.038670	-0.062329	-0.025917
hapoalim/teva	-0.148205	-0.040578	0.072879	-0.006444	-0.014364

	r=0	r=1	r=2	r=3	r=4
bezeq/bezeq	1.000000	-0.028109	-0.087261	-0.078059	-0.086672
bezeq/big	0.264540	0.005844	0.011131	-0.003964	0.063306
bezeq/hapoalim	0.227093	-0.024802	-0.019546	0.092961	0.026195
bezeq/teva	0.301720	-0.081865	0.006435	-0.001369	-0.047661

	r=0	r=1	r=2	r=3	r=4
teva/teva	1.000000	-0.079531	-0.101102	-0.086374	-0.056300
teva/big	-0.014246	0.019259	-0.005952	-0.019036	-0.043729
teva/hapoalim	-0.148205	0.005243	-0.016574	0.018677	0.069501
teva/bezeq	0.301720	0.012721	-0.031543	-0.037687	-0.021828

ניתן לראות שיש קורלציות יותר משמעותיות בין מניית `big` למניות של החברות האחרות.
במקרים אחרים יש הרבה קורלציות שליליות במיוחד בכל זוג עם `teva`.

עבשיו התוצאות עבור ערך מוחלט של התשואות.

	r=0	r=1	r=2	r=3	r=4
big/big	1.000000	-0.113115	-0.117223	-0.065651	-0.016546
big/hapoalim	0.532914	-0.180262	-0.108401	-0.051000	-0.030535
big/bezeq	0.264540	-0.034304	-0.086358	-0.026079	0.008429
big/teva	-0.014246	0.055043	-0.005820	-0.016986	-0.011032

	r=0	r=1	r=2	r=3	r=4
hapoalim/hapoalim	1.000000	0.271473	0.156169	0.079608	0.026867
hapoalim/big	0.606020	0.193716	0.181205	0.117163	0.100944
hapoalim/bezeq	0.492359	0.132506	0.165676	0.082468	0.045847
hapoalim/teva	0.320975	0.063922	0.034727	0.009333	-0.020635

	r=0	r=1	r=2	r=3	r=4
bezeq/bezeq	1.000000	0.108423	0.157364	0.122809	0.054679
bezeq/big	0.471231	0.090045	0.133118	0.041470	0.041131
bezeq/hapoalim	0.492359	0.138246	0.098241	0.037868	-0.000617
bezeq/teva	0.360247	0.058175	0.009894	-0.026249	-0.047164

	r=0	r=1	r=2	r=3	r=4
mizrahi/mizrahi	1.000000	0.206546	0.151198	0.076194	0.066097
mizrahi/discount	0.721475	0.232194	0.142014	0.107259	0.091971
mizrahi/hapoalim	0.643376	0.198805	0.169813	0.113636	0.113118
mizrahi/leumi	0.567802	0.219496	0.135144	0.069492	0.083072

מהתוצאות ניתן לראות שהקורלציות המשמעותיות יותר הן הקורלציות בזוגות עם mizrahi.

שאלה 3

(א)

קודם כל מצאתי Discount Factors עבור כל הנקודות זמן ע"י הנוסחה

$$D(T) = e^{-r \cdot T}$$

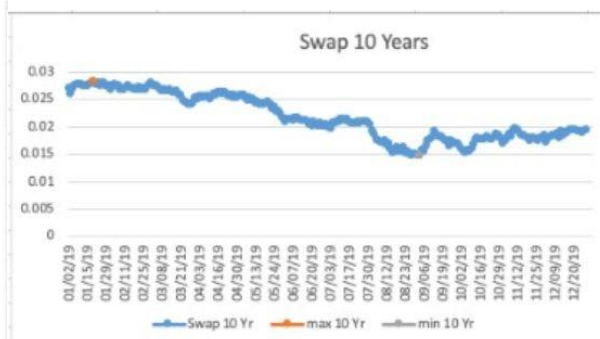
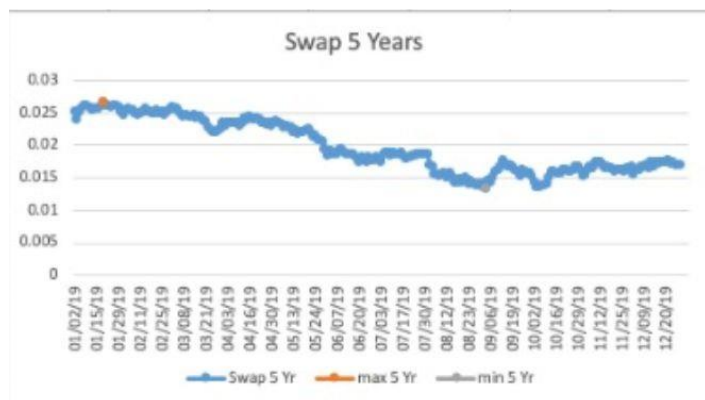
אחר כך מצאתי ערכים של שער הריבית עבור 4 שנים ע"י אינטרפולציה.

מחשבים annualized swap rates ע"י הנוסחה הבאה:

$$C_i = \frac{D_0 - D_i}{\sum_{i=1}^N (T_i - T_{i-1}) D_i}$$

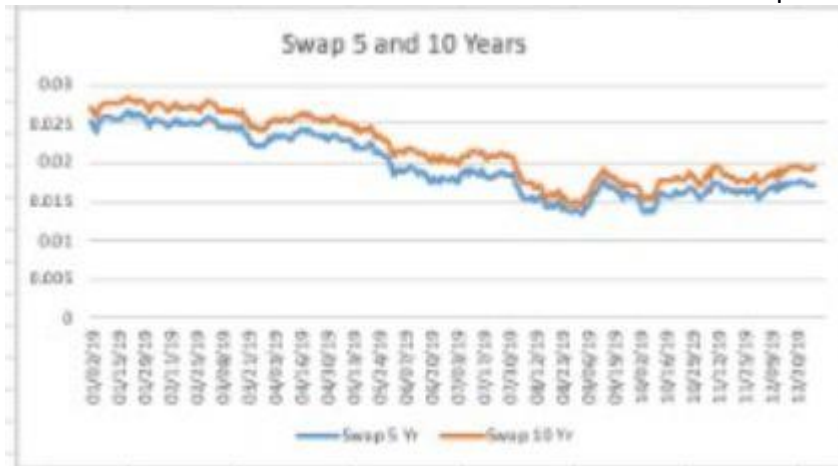
הערך המקסימלי מסומן בצבע כתום, הערך המינימלי מסומן בצבע אפור.

Swap 5 Yr	Swap 10 Yr	max 5 Yr	min 5 Yr	max 10 Yr	min 10 Yr
0.02521	0.0270409	MIN/A	MIN/A	MIN/A	MIN/A
0.02399	0.0259912	MIN/A	MIN/A	MIN/A	MIN/A
0.02521	0.0271359	MIN/A	MIN/A	MIN/A	MIN/A
0.02562	0.0274532	MIN/A	MIN/A	MIN/A	MIN/A
0.02613	0.0277755	MIN/A	MIN/A	MIN/A	MIN/A
0.02603	0.0278652	MIN/A	MIN/A	MIN/A	MIN/A
0.02593	0.0278596	MIN/A	MIN/A	MIN/A	MIN/A
0.02552	0.0275475	MIN/A	MIN/A	MIN/A	MIN/A
0.02562	0.0275488	MIN/A	MIN/A	MIN/A	MIN/A
0.02562	0.0276473	MIN/A	MIN/A	MIN/A	MIN/A
0.02573	0.0277527	MIN/A	MIN/A	MIN/A	MIN/A
0.02613	0.0279701	MIN/A	MIN/A	MIN/A	MIN/A
0.02654	0.0283878	0.02654	MIN/A	0.028388	MIN/A



(ב)

בתוצאות שקיבלתי לא קיים מצב ש - swap – 5 שנים עלה מעל swap ל-10 שנים. אבל ניתן לראות שיש נקודות בהן swap – 5 שנים עלה ממש קרוב לרמת ה- swap ל-10 שנים.



(ג)

על מנת למצוא את ה - zero coupon rates השתמשתי בנוסחאות הבאות:
מחשבים Discount Factors:

$$DF_i = \frac{1 - S_i * D_i}{1}$$

מחשבים Zero Coupon Rates:

$$ZC_i = -\ln(DF_i)$$

(פה $T_i - T_{i-1} = 1$)

הסיבה לשינוי בשערי הריבית היא שכסף אינו סטטי אך ערכו ממשיך להשתנות עם החברה ועם המצבה הכלכלי. שווי המטבע של מדינה נקבע ע"י השוק והוא קשור לתנאים הכלכליים ומדיניות כמו אינפלציה, סחר חוץ, תעסוקה, ריביות, קצב צמיחה ותנאים גיאופוליטיים.

Years	1	2	3	4	5	6	7	8	9	10
Swap Rate SEK	0.14	0.13	0.14	0.16	0.18	0.21	0.25	0.29	0.33	0.37
DF SEK	0.877193	0.776277	0.680945	0.587021	0.497476	0.411137	0.32891	0.254969	0.191706	0.1399312
ZC Rate SEK	0.131028	0.253246	0.384274	0.532694	0.698209	0.888829	1.111973	1.366615	1.651794	1.96660441
Swap Rate NOK	1.8	1.83	1.79	1.76	1.75	1.75	1.74	1.74	1.75	1.75
DF NOK	0.357143	0.126199	0.045233	0.016389	0.005959	0.002167	0.000791	0.000289	0.000105	3.8169E-05
ZC Rate NOK	1.029619	2.069896	3.095938	4.111168	5.122769	6.13437	7.142328	8.150286	9.161887	10.1734879
Swap Rate DKK	-0.24	-0.23	-0.21	-0.19	-0.13	-0.13	-0.1	-0.06	-0.01	0.03
DF DKK	1.315789	1.708817	2.16306	2.670445	3.069477	3.528134	3.920149	4.170371	4.212496	4.08980202
ZC Rate DKK	-0.27444	-0.5358	-0.77152	-0.98224	-1.12151	-1.26077	-1.36613	-1.42801	-1.43806	-1.4084966