



Academia de Studii Economice  
Facultatea de Cibernetică, Statistică și Informatică Economică  
Specializarea: Informatică Economică

## Proiect Pachete Software

**Profesor coordonator**  
Conferențiar univ. dr.  
Vreja Lucia-Ovidia

**Studenți**  
Rădulescu Theodor  
Raicea David-Gabriel

# Capitolul Python

## Introducere

Aplicația prezentată este o platformă interactivă construită în Streamlit care are ca scop analiza și vizualizarea evoluției acțiunilor Samsung Electronics, precum și explorarea unor fenomene geopolitice și economice regionale relevante. Folosind tehnici de preprocesare, codificare, scalare, modelare și cartografiere, aplicația oferă utilizatorilor o experiență completă de investigare a datelor financiare și geografice. Aceasta include atât aspecte fundamentale precum tratarea valorilor lipsă, analiza corelațiilor și regresii, cât și funcționalități avansate precum clusterizare KMeans, predicție logistică, precum și vizualizări dinamice pe hărți ale distribuției globale a brandurilor de telefoane sau ale râurilor din Coreea de Sud.

Aplicația este organizată în două secțiuni principale, accesibile prin bara laterală: „Proiect” și „Informații”. În cadrul secțiunii „Proiect” sunt implementate toate funcționalitățile analitice și vizualizările interactive, grupate în mai multe subsecțiuni tematice. Acestea includ atât aspecte fundamentale ale analizei datelor, precum prezentarea setului de date, filtrări, tratarea valorilor lipsă și extreme, codificare, corelații și scalare, cât și etape de prelucrare statistică și vizualizare agregată. De asemenea, aplicația oferă și componente avansate precum analiza geospațială (prin hărți interactive), tehnici de machine learning aplicate pe date financiare (clusterizare, regresie logistică și multiplă), precum și informații economice regionale. Utilizatorul navighează între aceste funcționalități prin intermediul unei interfețe de tip radio button, ceea ce asigură o tranziție intuitivă între conținuturi. Secțiunea „Informații” oferă detalii privind sursa datelor utilizate.

## 1. Variabile globale și fișiere utilizate

Aplicația folosește fișierul Samsung.csv, cu date despre acțiunile Samsung Electronics.

Variabilele globale esențiale includ df (setul original), df\_final (versiunea filtrată și curățată), df\_scaled\_model (versiunea scalată pentru modele predictive), df\_phones (pentru harta brandurilor) și world și cities (pentru hărți geopolitice). Aceste variabile sunt utilizate pe întreg parcursul aplicației, fiind comune tuturor subsecțiunilor.

```
df = pd.read_csv('data/Samsung.csv')
df['Date'] = pd.to_datetime(df['Date']).dt.date
df_final = df.copy()
df_scaled_model = df.copy()
scaler_model = StandardScaler()
scaled_values_model = scaler_model.fit_transform(df_final.select_dtypes(include=[np.number]))
df_scaled_model = df_final.copy()
df_scaled_model[['{col}_scaled' for col in df_final.select_dtypes(include=[np.number]).columns]] =
scaled_values_model
```

## 2. Prezentarea datelor

Fișierul încărcat conține date zilnice privind evoluția acțiunilor Samsung, iar conversia coloanei Date asigură compatibilitatea cu widgeturile de selecție din Streamlit.

Pentru afișarea brută a datelor, aplicația utilizează:

```
st.dataframe(df)
```

Aceasta oferă utilizatorului acces complet la toate înregistrările și permite o evaluare vizuală rapidă. În continuare, fiecare coloană este explicată prin mesaje informative precum:

```
st.info("*** Open ** → Prețul acțiunilor la începutul sesiunii (preț de deschidere).")
```

Acest tip de documentare contextuală ajută utilizatorul să înțeleagă sensul variabilelor financiare fără a fi necesară consultarea unei surse externe.

Pentru o vedere sintetică a structurii datasetului, sunt afișate tipurile de date:

```
st.write({col: str(dtype) for col, dtype in df.dtypes.items()})
```

și dimensiunile acestuia:

```
st.write(f"Dimensiunea setului de date: setul conține {df.shape[0]} rânduri și {df.shape[1]} coloane")
```

La final, sunt calculate și afișate statisticile descriptive de bază:

```
st.dataframe(df.describe())
```

Acestea oferă o imagine de ansamblu asupra distribuției valorilor și permit identificarea eventualelor anomalii încă din faza inițială.

## 3. Filtrări pe baza datelor

În această secțiune, aplicația oferă utilizatorului mai multe moduri interactive de a restrânge vizualizarea datelor din df. Primul tip de filtrare permite selectarea unor coloane și a unui prag minim pentru volumul tranzacționat:

```
col_select = st.multiselect("Selectează coloanele", coloane1)
df_filtrat = df[col_select]
df_filtrat = df_filtrat[df_filtrat["Volume"] >= min_vol]
```

Această abordare este utilă pentru a evidenția doar zilele cu activitate bursieră semnificativă și pentru a analiza doar informațiile relevante.

A doua filtrare permite alegerea unui interval de rânduri:

```
start_row, end_row = st.slider("Afișează următoarele rânduri:", 0, len(df) - 1, (0, len(df) - 1))
df_filtrat1 = df.iloc[start_row:end_row + 1]
```

Aceasta este utilă în special pentru inspecție vizuală rapidă sau testare.

Al treilea mecanism de filtrare implică selecția manuală a unora dintre sesiunile de tranzacționare:

```
sesiuni_selectate = st.multiselect("Selectează sesiunile:", df["Date"].unique().tolist())
df_filtrat2 = df[df["Date"].isin(sesiuni_selectate)]
```

Acest tip de filtrare este flexibil, permițând analiza specifică pentru anumite zile sau evenimente bursiere.

Ultima variantă de filtrare se bazează pe un interval de date, folosind un widget de tip calendar:

```
data_range = st.date_input('Selectează intervalul de sesiuni:', [min_data,max_data])
df_filtrat3 = df[(df['Date'] >= start_data) & (df['Date'] <= end_data)]
```

Aceasta permite selectarea unei ferestre temporale și este extrem de utilă în explorarea evoluției prețurilor pe perioade personalizate.

Toate aceste filtre sunt implementate cu actualizare în timp real, iar rezultatele sunt afișate în `st.dataframe(...)`, oferind o experiență fluidă și control precis asupra datelor analizate.

## Filtrare #4

Selectează intervalul de sesiuni:

2000/01/04 – 2022/05/23

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-01-04	6000	6110	5660	6110	4651.7378	74195000
1	2000-01-05	5800	6060	5520	5580	4248.2324	74680000
2	2000-01-06	5750	5780	5580	5620	4278.6865	54390000
3	2000-01-07	5560	5670	5360	5540	4217.7803	40305000
4	2000-01-10	5600	5770	5580	5770	4392.8848	46880000
5	2000-01-11	5820	6100	5770	5770	4392.8848	59745000
6	2000-01-12	5610	5740	5600	5720	4354.8184	29220000
7	2000-01-13	5600	5740	5560	5710	4347.2051	41190000
8	2000-01-14	5720	5880	5680	5830	4438.5659	49375000
9	2000-01-17	6000	6180	5920	6100	4644.125	63505000

Figure 1. Exemplu de filtrare asupra datelor în Python

## 4. Tratarea valorilor lipsă și a valorilor extreme

Deși setul original nu conține valori lipsă, aplicația simulează această situație pentru scopuri educaționale. Se forțează introducerea unor NaN în coloane numerice folosind:

```
valori_lipsa_index = [5, 10, 15, 20, 25]
for idx, col in zip(valori_lipsa_index, coloane_afectate):
    df_simulat.loc[idx, col] = np.nan
```

Ulterior, utilizatorul poate alege între două metode de tratare: eliminarea rândurilor (dropna) sau completarea valorilor lipsă (fillna). În cazul completării, sunt oferite trei opțiuni: medie, mediană sau zero:

```
df_tratat.fillna(df_tratat.mean(numeric_only=True), inplace=True)
```

Pentru evidențierea celulelor completate, aplicația colorează în verde valorile înlocuite, folosind o funcție de stilizare aplicată pe subsetul afectat.

În partea a doua a secțiunii, sunt tratate valorile extreme pe baza scorului Z:

```
z_scores = np.abs(zscore(df_numeric))
extreme_mask = (z_scores > 3)
df_extreme = df_outlier_test[extreme_mask.any(axis=1)]
```

Aplicația afișează aceste valori într-un dataframe cu fundal portocaliu pentru a le face ușor de identificat. Apoi, utilizatorul alege între afișare, eliminare sau înlocuire cu mediană:

```
for col in df_numeric.columns:
    mask_col = extreme_mask[:, df_numeric.columns.get_loc(col)]
    df_outlier_tratat.loc[mask_col, col] = mediane[col]
```

La final, aplicația aplică decizia selectată pe setul real:

```
df_final.loc[mask_col, col] = mediane_final[col]
```

Este prezentată și o justificare a acestei alegeri, subliniind că înlocuirea cu mediana păstrează structura datelor și reduce impactul valorilor anormale.

Tot în această secțiune este tratată și problema volumelor zero, care, deși nu sunt outliers statistici, sunt eliminate pentru că nu oferă valoare analitică:

```
df[df["Volume"] == 0]
```

Astfel, această subsecțiune oferă o bază solidă pentru curățarea și pregătirea datelor înainte de analize ulterioare.

## Tratarea valorilor extreme

### Detectarea valorilor extreme pe baza Z-score ( $> 3$ sau $< -3$ )

Număr de rânduri cu valori extreme: 144

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-01-04	6000.000	6110.000	5660.000	6110.000	4651.738	74195000
1	2000-01-05	5800.000	6060.000	5520.000	5580.000	4248.232	74680000
20	2000-02-01	5600.000	5680.000	5260.000	5320.000	4050.286	71470000
22	2000-02-03	5250.000	5460.000	4970.000	5130.000	3905.633	142765000
31	2000-02-16	5180.000	5200.000	4780.000	5000.000	3806.660	73115000
33	2000-02-18	5370.000	5630.000	5160.000	5300.000	4035.060	108750000
42	2000-03-02	5880.000	5880.000	5520.000	5880.000	4476.632	117750000
43	2000-03-03	6200.000	6250.000	5880.000	6000.000	4567.992	140990000
56	2000-03-22	5960.000	6300.000	5880.000	6200.000	4720.257	86975000
58	2000-03-24	6320.000	7120.000	6180.000	6800.000	5177.057	90260000

Simulează metode de tratare:

- ☐ Doar evidențiere  
☒ Eliminare  
☐ Înlocuire cu mediană

Figure 2. Tratarea valorilor extreme în Python

## 5. Metode de codificare a datelor

În această etapă, aplicația transformă variabila temporală Date într-o formă mai analitică, extrăgând luna corespunzătoare fiecărei înregistrări:

```
df_encoding["Luna"] = pd.to_datetime(df_encoding["Date"]).dt.month
```

Apoi, în funcție de lună, este generată o nouă coloană Anotimp printr-o funcție personalizată care grupează lunile în cele patru anotimpuri:

```
def anotimp(luna):  
    if luna in [12, 1, 2]:  
        return "Iarna"  
    elif luna in [3, 4, 5]:  
        return "Primavara"  
    elif luna in [6, 7, 8]:  
        return "Vara"  
    else:  
        return "Toamna"  
df_encoding["Anotimp"] = df_encoding["Luna"].apply(anotimp)
```

După această transformare, utilizatorul poate alege una dintre cele două metode de codificare: Label Encoding sau One-Hot Encoding. În cazul primei metode, valorile categorice sunt înlocuite cu etichete numerice:

```
le = LabelEncoder()
df_encoding_label["Anotimp_Label"] = le.fit_transform(df_encoding_label["Anotimp"])
```

Pentru One-Hot Encoding, se generează o coloană binară pentru fiecare anotimp, utilizând funcția `pd.get_dummies()`:

```
df_encoding_ohe = pd.get_dummies(df_encoding.copy(), columns=["Anotimp"])
```

Aplicația explică vizual această transformare, afișând un tabel de referință cu vectorii binari corespunzători fiecărui anotimp.

În final, este aplicată metoda One-Hot Encoding pe setul real de analiză:

```
df_final = pd.get_dummies(df_encoding_real, columns=["Anotimp"])
```

Această alegere este justificată în interfață prin avantajul oferit în analiza agregată și filtrarea clară pe categorii, fără introducerea unei ordini artificiale între anotimpuri.

Mostră random de 12 luni diferite (cu anotimpuri asociate):

	Date	Luna	Anotimp
0	2003-01-10		1 Iarna
1	2008-02-01		2 Iarna
2	2001-03-06		3 Primavara
3	2018-04-06		4 Primavara
4	2014-05-20		5 Primavara
5	2013-06-20		6 Vara
6	2004-07-01		7 Vara
7	2005-08-23		8 Vara
8	2005-09-07		9 Toamna
9	2011-10-04		10 Toamna

Simulează metoda de codificare:

- ☒ Label Encoding  
☐ One-Hot Encoding

Rezultat - Label Encoding:

	Anotimp	Anotimp_Label
0	Iarna	0
1	Primavara	1
2	Toamna	2
3	Vara	3

Număr de intrări per anotimp:

	Anotimp	Număr apariții
0	Primavara	1412
1	Vara	1399
2	Toamna	1345
3	Iarna	1337

Figure 3. Label encoding pe anotimpuri în Python

## 6. Analiza corelațiilor

Această parte a aplicației își propune să evidențieze relațiile dintre variabilele numerice din setul de date prelucrat. Se selectează automat doar coloanele de tip numeric folosind:

```
df_corr = df_final.select_dtypes(include=['float64', 'int64'])
```

Coefficienții de corelație Pearson sunt calculați cu `.corr()` și redați atât în formă tabelară:

```
st.dataframe(df_corr.corr().round(2))
```

cât și vizual, sub forma unei matrici de corelație:



```
sns.heatmap(df_corr.corr(), annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5, ax=ax)
```

Această reprezentare ajută la identificarea rapidă a relațiilor puternice, directe sau inverse, dintre variabile.

Interpretarea este ghidată în interfață. De exemplu, se menționează că variabilele legate de preț (Open, High, Low, Close, Adj Close) sunt extrem de corelate între ele, ceea ce este de așteptat. Se sugerează păstrarea uneia singure în etapele ulterioare pentru a evita multicolaritatea, dar fără a elimina efectiv celelalte.

De asemenea, se observă o corelație negativă moderată între Volume și restul variabilelor, ceea ce poate reflecta comportamente opuse între nivelul de activitate și evoluția prețului.

Această secțiune oferă astfel o bază clară pentru selecția variabilelor în modelele statistice ulterioare.

#### Tabelul coeficienților Pearson:

	Open	High	Low	Close	Adj Close	Volume
Open	1	0.99	1	0.99	0.96	-0.42
High	0.99	1	0.99	1	0.96	-0.42
Low	1	0.99	1	1	0.96	-0.42
Close	0.99	1	1	1	0.96	-0.42
Adj Close	0.96	0.96	0.96	0.96	1	-0.4
Volume	-0.42	-0.42	-0.42	-0.42	-0.4	1

#### Matrice de corelație (Heatmap):

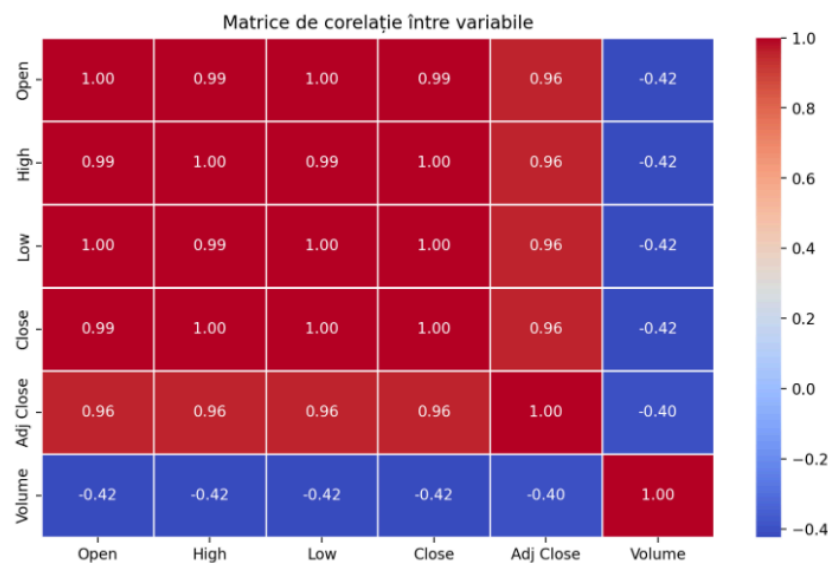


Figure 4. Analiza corelațiilor în Python

## 7. Metode de scalare a datelor

Scopul acestei secțiuni este de a normaliza variabilele numerice pentru a le aduce pe o scară comună, lucru esențial pentru algoritmi de învățare automată. Utilizatorul selectează mai întâi coloanele dorite:

```
coloane_scalare = st.multiselect("Coloane disponibile:", coloane_disponibile, default=coloane_disponibile)
```

Apoi, alege metoda de scalare: Min-Max, Standard sau Robust:

```
metoda_scalare = st.radio("Metodă:", ["Min-Max", "Standard (Z-score)", "Robust"])
```

Fiecare metodă este asociată unui obiect scaler corespunzător din sklearn:

```
scaler = StandardScaler()  
df_scaled_values = scaler.fit_transform(df_numeric[coloane_scalare])
```

Valorile rezultate sunt afișate într-un tabel comparativ cu cele originale:

```
st.dataframe(df_scalare_viz.style.format(precision=3))
```

În plus, aplicația generează două boxploturi pentru a evidenția modificările în distribuția datelor:

```
axs[0].boxplot(...) # original  
axs[1].boxplot(...) # scalat
```

La finalul secțiunii, aplicația anunță decizia de a folosi Standard Scaling (Z-score) în pașii următori, aplicat astfel:

```
scaled_values_model = scaler_model.fit_transform(df_scaled_model.select_dtypes(include=[np.number]))
```

și stocat în df\_scaled\_model pentru a fi folosit ulterior în clusterizare și regresie.

### Compararea distribuției (boxplot):

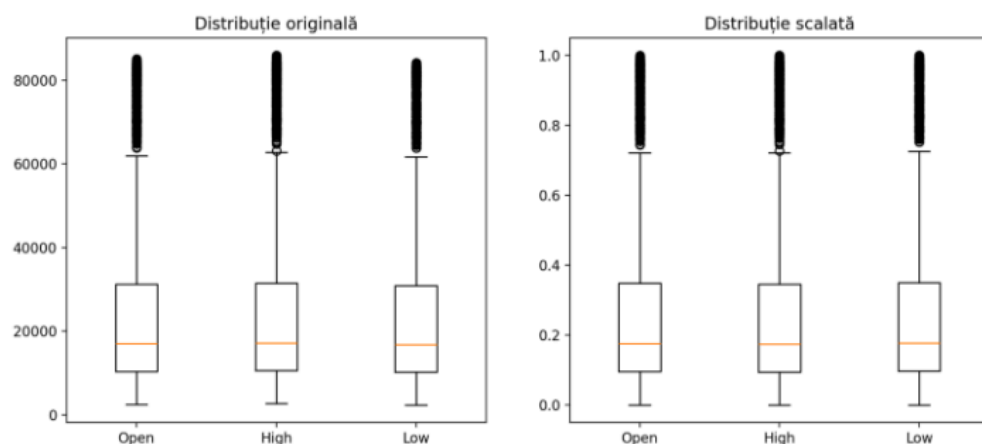


Figure 5. Boxplot pentru distribuția datelor în Python

## 8. Prelucrări statistice, grupare și agregare

Această secțiune permite utilizatorului să grupeze datele după anotimp sau lună și să aplice funcții statistice precum medie, sumă, minim, maxim sau deviație standard. Gruparea este aleasă printr-un selector:

```
grupare = st.selectbox("Alege coloana pentru grupare:", ["Anotimp", "Luna"])
```

Utilizatorul selectează apoi coloanele numerice și funcțiile de agregare dorite:

```
df_agregat = df_stats.groupby(grupare, as_index=False)[coloane_alease].agg(funcții_alease)
```

Rezultatul este afișat într-un tabel colorat gradat:

```
st.dataframe(df_agregat.style.format(precision=2).background_gradient(cmap="Blues", axis=None))
```

Pentru o mai bună interpretare, aplicația permite generarea automată de grafice tip bar chart sau line chart, în funcție de datele agregate:

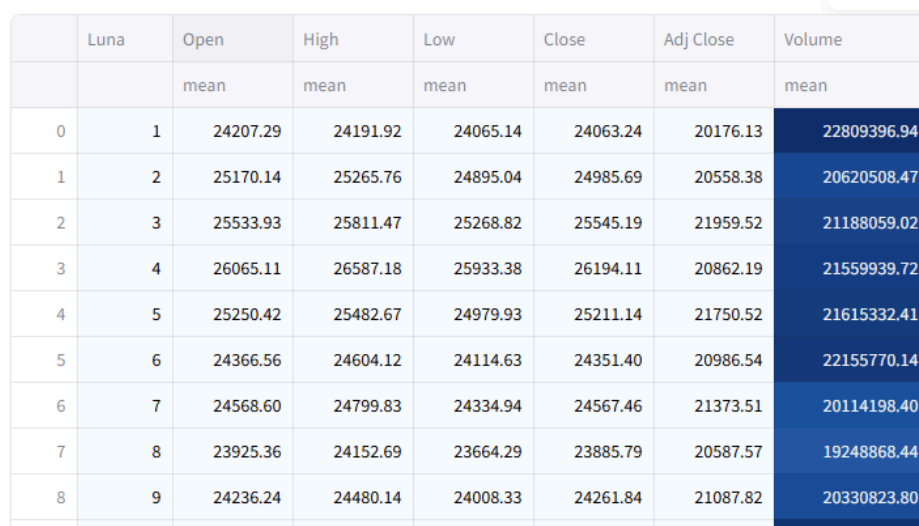
```
ax.bar(x_vals, y_vals) # sau ax.plot(...)
```

Dacă este selectată o singură coloană numerică, se oferă și o reprezentare de tip boxplot, utilă pentru a înțelege distribuția valorilor în funcție de anotimp sau lună:

```
df_boxplot.boxplot(by=grupare, column=col_box, ax=ax2)
```

Această combinație între tabele statistice și vizualizări grafice permite utilizatorului să observe tendințe sezoniere, anomalii sau perioade de vârf în evoluția acțiunilor Samsung.

### Tabelul rezultat (cu agregări):



	Luna	Open	High	Low	Close	Adj Close	Volume
		mean	mean	mean	mean	mean	mean
0	1	24207.29	24191.92	24065.14	24063.24	20176.13	22809396.94
1	2	25170.14	25265.76	24895.04	24985.69	20558.38	20620508.47
2	3	25533.93	25811.47	25268.82	25545.19	21959.52	21188059.02
3	4	26065.11	26587.18	25933.38	26194.11	20862.19	21559939.72
4	5	25250.42	25482.67	24979.93	25211.14	21750.52	21615332.41
5	6	24366.56	24604.12	24114.63	24351.40	20986.54	22155770.14
6	7	24568.60	24799.83	24334.94	24567.46	21373.51	20114198.40
7	8	23925.36	24152.69	23664.29	23885.79	20587.57	19248868.44
8	9	24236.24	24480.14	24008.33	24261.84	21087.82	20330823.80

Figure 6. Agregări pe baza mediei în Python

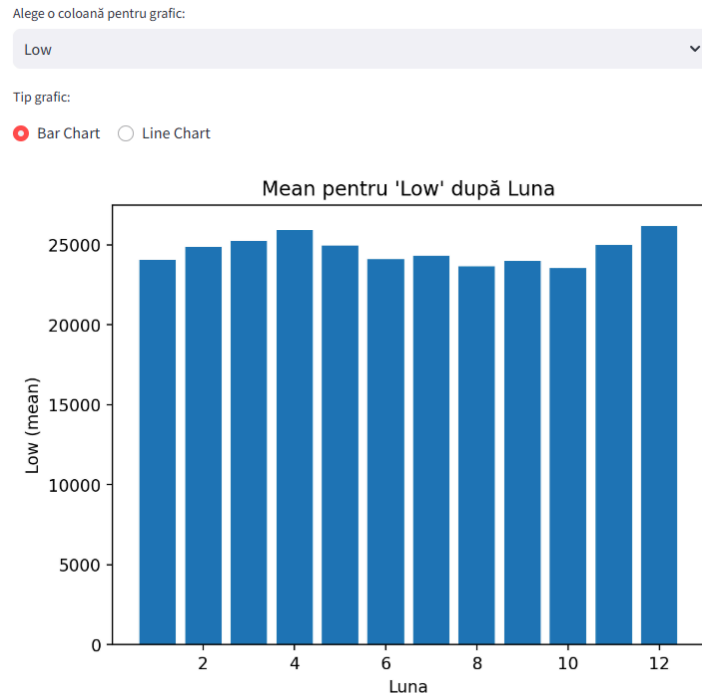


Figure 7. Bar chart pe baza mediei coloanei Low generat în Python

## 9. Harta interactivă a distribuției globale a brandurilor de telefoane

În era tehnologiei mobile, înțelegerea distribuției globale a brandurilor de smartphone-uri este esențială pentru marketing, strategie economică și cercetare de piață. Această secțiune a aplicației oferă o hartă interactivă a cotei de piață a principalelor branduri din fiecare țară, bazată pe date actualizate și prelucrate inteligent.

### 1. Prelucrarea fișierului PhoneBrandsWorld.csv

Fișierul conține pentru fiecare țară:

- numele (Country);
- cele 3 branduri principale și cotele lor de piață (#1 Brand, #1 Market Share, etc.).

Valorile procentuale au fost procesate astfel:

```
df_phones[col] = df_phones[col].str.replace('%', '').astype(float)
df_phones[f'{col} (%)'] = df_phones[col].apply(lambda x: f'{x:.2f}%')
```

Acest lucru permite afisarea coerentă și uniformă în tooltip.

### 2. Îmbinarea cu datele geospațiale

S-a realizat un merge între datele despre țări (din Natural Earth) și cele despre telefoane:

```
df_merged = world.merge(df_phones, left_on="ADMIN", right_on="Country", how="inner")
```

Se obține astfel un GeoDataFrame care conține și geometria fiecărei țări, și informații despre brandurile dominante.

### 3. Colori personalizate pe branduri

Fiecare brand primește o culoare specifică:

```
color_map = {  
    "Apple": "#5C6BC0", "Samsung": "#26A69A", "Xiaomi": "#FFD700", ...  
}  
df_merged["color"] = df_merged["#1 Brand"].map(color_map).fillna("#E0E0E0")
```

Vizualizarea cu Folium

Folosind biblioteca **folium**, s-a creat o hartă interactivă cu următoarele caracteristici:

- stil selectabil de către utilizator (light sau dark);
- țările sunt colorate în funcție de brandul dominant;
- tooltipul afișează:
  - cele 3 branduri principale;
  - cotele lor de piață;
  - numele țării.

Cod pentru harta interactivă:

```
folium.GeoJson(  
    data=df_merged,  
    style_function=..., # stabilește culoarea după brand  
    tooltip=GeoJsonTooltip(fields=..., aliases=..., style=...)  
).add_to(m)
```

### Legendă branduri dominante

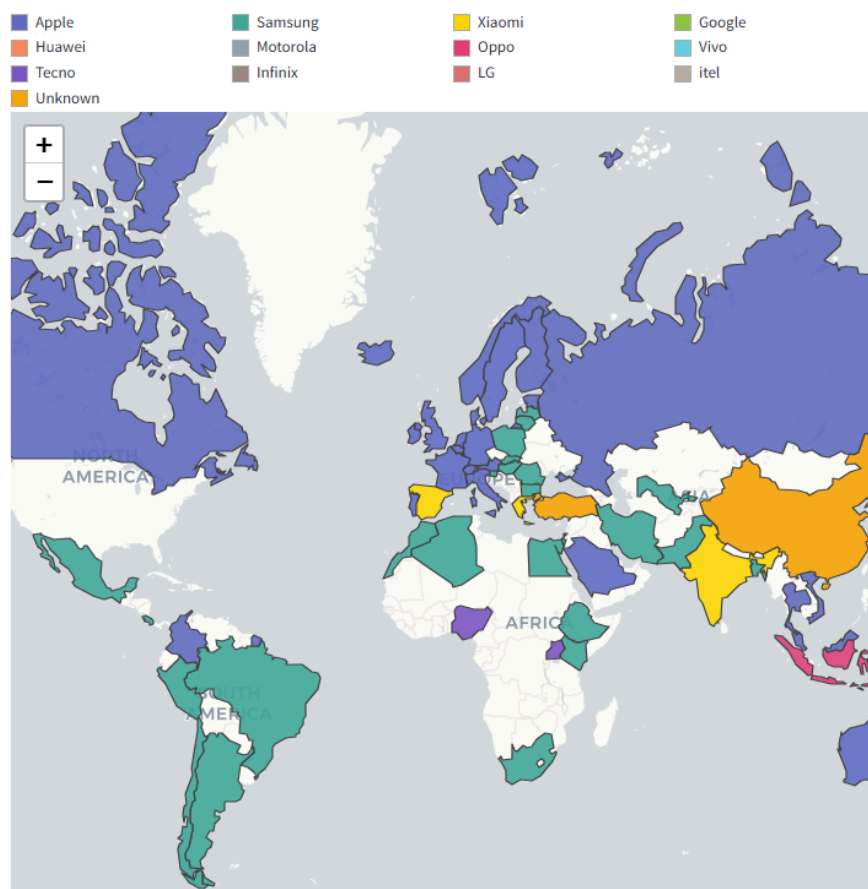


Figure 8. Distribuția în lume a brandurilor de telefoane

## Interpretare

Această hartă oferă informații instantanee despre structura pieței globale:

- Apple domină în SUA, Canada, Japonia;
- Samsung are poziție puternică în Coreea de Sud, Germania, Australia;
- Xiaomi și alte branduri asiatice sunt preferate în India, Pakistan, Indonezia;
- Africa este fragmentată, cu branduri locale ca Tecno, Infinix, itel.

## 10. Harta râurilor din Coreea de Sud

Această secțiune adaugă o componentă esențială în analiza geospațială a Peninsulei Coreene: hidrografia. Prin combinarea datelor geografice despre râuri și granițele administrative, vizualizăm atât rețeaua de ape curgătoare din Coreea de Sud, cât și conexiunile hidrologice cu statele vecine. Aceasta este o analiză de tip infrastructural, ecologic și geopolitic, cu aplicații practice în urbanism, resurse naturale și cooperare transfrontalieră.

Râurile din Coreea de Sud

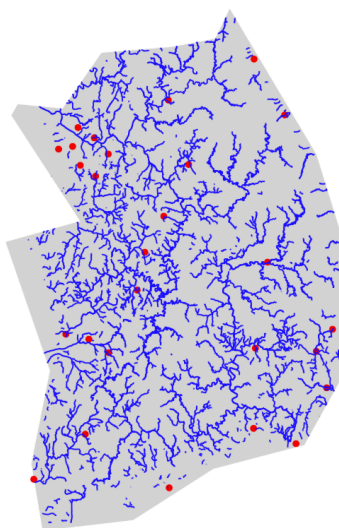


Figure 9. Harta hidrologică a Coreei de Sud

### Selectarea și filtrarea datelor

- S-au folosit date din Natural Earth:
  - world – contururi geografice;
  - cities – poziții orașe;
  - rivers – rețeaua hidrografică.
- S-au selectat doar elementele relevante pentru Coreea de Sud și țările vecine prin `.touches()`.

### Extracția râurilor din Coreea de Sud

```
rivers_in_korea = gpd.overlay(rivers, south_korea, how="intersection")
```

Vizualizarea este realizată cu matplotlib, unde:

- Coreea de Sud este colorată în gri deschis;
- Râurile sunt colorate albastru intens;
- Orașele sud-coreene apar cu roșu.

Pentru a vedea interdependența hidrografică regională, s-a determinat câte râuri din dataset traversează vecinii Coreei de Sud:

```
rauri_pe_tari = rivers_crossing.groupby('ADMIN').size().reset_index(name='Număr râuri')
```

Compararea rețelei de râuri: Sud vs. Nord

S-au selectat și convertit datele pentru Coreea de Sud și Coreea de Nord în EPSG:32652 pentru compatibilitate metrică.

```
rivers_in_coreea = gpd.sjoin(rivers, coreea, how='inner', predicate='intersects')
```

Harta finală arată:

- granițele Coreei de Sud (red) și Nord (blue);
- fundal gri pentru context geografic;
- râurile principale din ambele țări, cu albastru închis.

## 11. Vecinii Coreei de Sud și Granița cu Coreea de Nord

Această secțiune a aplicației oferă o analiză geografică completă a poziției Coreei de Sud în contextul geopolitic est-asiatic. Ne-am propus să:

1. Identificăm vecinii geografici direcți ai Coreei de Sud;
2. Vizualizăm relația spațială dintre Coreea de Sud și Coreea de Nord, incluzând capitalele și centroizii;
3. Calculăm distanța exactă dintre Seul și centrul geografic al Coreei de Nord;
4. Afășăm toate aceste relații într-un mod vizual, interactiv și interpretabil.

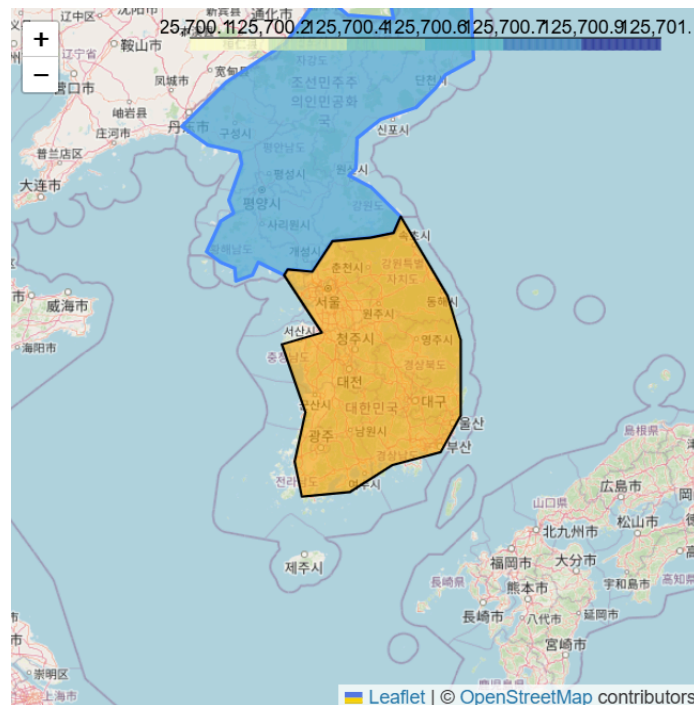


Figure 10. Harta vecinilor Coreei de Sud

### 1. Determinarea vecinilor

Se selectează poligonul Coreei de Sud și se caută toate geometriile din datasetul world care se ating spațial de acesta (folosind `.touches()` din GeoPandas):

```
vecini = world[world.touches(south_korea.iloc[0].geometry)]
```

Se combină aceste entități într-un GeoDataFrame pentru afișare:

```
combinat = gpd.GeoDataFrame(pd.concat([vecini, south_korea]), crs=world.crs)
```

### 2. Calculul suprafeței țărilor

Folosind proiecția EPSG:32652 (UTM zona 52N), se calculează suprafața fiecărei țări vecine în km<sup>2</sup>:

```
neighbours["area_km2"] = neighbours.to_crs(epsg=32652).geometry.area / 1_000_000
```

### 3. Vizualizare interactivă cu Folium

Se creează o hartă cu:

- Coreea de Sud colorată cu portocaliu;
- Țările vecine colorate cu colormap "YlGnBu";
- Tooltipuri interactive ce arată numele și suprafața fiecărei țări.

```
folium.Choropleth(...).add_to(m)
```

```
GeoJson(..., tooltip=GeoJsonTooltip(...)).add_to(m)
```

### 4. Calculul centroizilor și distanței Seul – Coreea de Nord

Centroizii sunt calculați în proiecție metrică, iar Seul este transformat în același sistem (EPSG:32652):

```
seoul_utm = gpd.GeoSeries([Point(126.9780, 37.5665)], crs="EPSG:4326").to_crs(epsg=32652).iloc[0]
```

```
nk_centroid = north_korea.geometry.centroid.iloc[0]
```

Distanța este exprimată în kilometri:

```
dist_km = seoul_utm.distance(nk_centroid) / 1000
```

### 5. Reprezentări avansate cu contextily

Pentru un plus de realism, harta finală este afișată peste un fundal de tip satelit (tiles OpenStreetMap sau Bing), folosind biblioteca contextily. Seul este marcat cu o stea neagră (\*), iar centroizii țărilor sunt evidențiați cu galben.

```
contextily.add_basemap(ax, crs=coreea_ctx.crs.to_string())
```

## 12. Analiza orașelor Coreei de Sud

În cadrul acestui proiect, am explorat datele geografice referitoare la orașele din Coreea de Sud pentru a identifica:

- Zonele potrivite pentru amplasarea unor fabrici departe de aglomerările urbane;
- Orașele care sunt cele mai apropiate de granița cu Coreea de Nord – aspect esențial pentru analiză geopolitică și de securitate.

Am utilizat bibliotecile geopandas, shapely, matplotlib și geopy pentru această analiză, în combinație cu datele oferite de Natural Earth și OpenStreetMap.

### 1. Identificarea zonelor industriale potențiale ( $\geq 3$ km de orașe):



Obiectiv: localizarea spațiilor libere care respectă o distanță de cel puțin 3 km față de orice oraș – o condiție standard în urbanismul industrial.

- Selectăm orașele din Coreea de Sud:

```
cities_sk = cities[cities["SOV0NAME"] == "Korea, South"]
```

- Reproiectăm datele în sistem metric EPSG:32652 (UTM pentru Coreea):

```
sk_utm = south_korea.to_crs(epsg=32652)
```

```
cities_utm = cities_sk.to_crs(epsg=32652)
```

- Creăm buffer de 3 km în jurul fiecărui oraș:

```
buffers["geometry"] = cities_utm.buffer(3000)
```

- Calculăm diferența: zonele din țară care nu sunt în aceste buffere:

```
fabrici_potentiale = gpd.overlay(sk_utm, buffers, how='difference')
```

Vizualizare: folosim matplotlib pentru a evidenția:

- zonele potențiale cu verde deschis;
- orașele cu roșu;
- granița țării cu negru.

Zone din Coreea de Sud pentru potențiale fabrici (minim 3 km distanță de orașe)

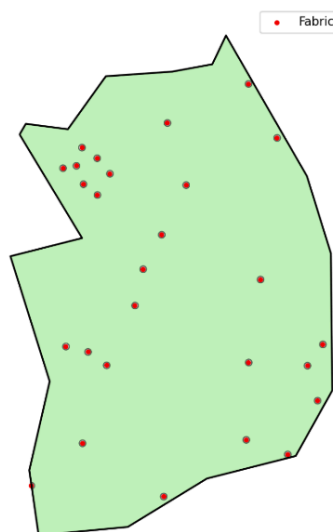


Figure 11. Zone din Coreea de Sud în care s-ar putea construi fabrici

## 2. Distanța dintre orașele sud-coreene și Coreea de Nord

Obiectiv: identificarea orașelor cele mai apropiate de granița nord-coreeană – informație esențială în planificarea infrastructurii, siguranță națională și analiza vulnerabilităților geopolitice.

- Selectăm orașele sud-coreene (cities\_sk) și le reproiectăm în sistem WGS84 (EPSG:4326).
- Selectăm geometria Coreei de Nord și o unificăm:

```
nk_geometry = north_korea.union_all()
```

- Pentru fiecare oraș:

- găsim cel mai apropiat punct de pe granița Coreei de Nord folosind:

```
nearest_points(city.geometry, nk_geometry)
```

- calculăm distanța în km cu geodesic() din geopy.

- Salvăm rezultatele într-un tabel sortat:

```
df_distante = pd.DataFrame(results).sort_values(...)
```

### Distanța de la fiecare oraș din Coreea de Sud până la granița cu Coreea de Nord

Oraș	Distanță până la Coreea de Nord (km)
Goyang	21.36
Bucheon	34.81
Incheon	36.48
Seoul	38.67
Sokcho	38.98
Chuncheon	47.21
Ansan	53.01
Songnam	57.22
Suwon	66.98
Gangneung	91.23

Figure 12. Distanța de la fiecare oraș din Coreea de Sud până la granița vecinilor

## 13. GDP Asia

În cadrul aplicației am inclus o secțiune dedicată analizei geografice a PIB-ului țărilor asiatice. Scopul acestei componente este de a oferi o perspectivă vizuală și comparativă asupra nivelului de dezvoltare economică la nivel continental, folosind atât date numerice cât și hărți geospațiale.

Această analiză integrează biblioteci precum **pandas**, **geopandas** și **matplotlib**. Scopul principal al secțiunii este:

- să îmbinăm date economice (PIB în dolari) cu reprezentări cartografice;
- să realizăm o hartă care evidențiază diferențele între țările asiatice;
- să adăugăm etichete informative pe hartă, direct în dreptul fiecărei țări.

### 1. Filtrarea țărilor asiatice

Datele geografice sunt preluate dintr-un shapefile **world** care include toate țările. Se păstrează doar țările din Asia:

```
asia = world[world["CONTINENT"] == "Asia"].copy()
```

### 2. Încărcarea datelor economice

Se utilizează un fișier CSV **asian\_gdp\_clean.csv** care conține valorile PIB (GDP) pentru fiecare țară:

```
gdp = pd.read_csv("data/asian_gdp_clean.csv")
```

Presupunerea este că în acest fișier există coloana "Country" care se potrivește cu "ADMIN" din asia.

### 3. Combinarea datelor (merge)

Se realizează un join între geometriile țărilor și valorile PIB:

```
merged = asia.merge(gdp, left_on="ADMIN", right_on="Country", how="left")
```

Pentru o afișare coerentă și scalată uniform, datele sunt reproiectate într-un sistem de coordonate metric:

```
proj = merged.to_crs(epsg=3035) # Europe-centric metric projection
```

Apoi, se calculează **centroizii** fiecărei țări pentru a putea afișa etichete texte în centrul hărții:

```
centroids = proj.geometry.centroid.to_crs(merged.crs)
```

Se construiește o hartă folosind matplotlib:

```
fig, ax = plt.subplots(figsize=(15, 10))
```

- Se trasează conturul țărilor asiatice cu o culoare de fundal neutră (**lightgray**);
- Se colorează doar țările care au valori GDP disponibile, folosind colormap **viridis**;
- Se adaugă legendă cu eticheta "GDP (USD)".

```
merged.plot(color='lightgray', edgecolor='black', ax=ax)
```

```
merged[merged['GDP_USD'].notna()].plot(column='GDP_USD', cmap='viridis', legend=True, ax=ax, ...)
```

Etichetele fiecărei țări sunt plasate la centrul său geometric:

```
for x, y, label in zip(centroids.x, centroids.y, proj["ADMIN"]):
```

```
    ax.text(x, y, label, fontsize=7, ha='center', va='center')
```

Harta obținută oferă un impact vizual puternic asupra diferențelor economice dintre țările asiatice:

- Țări precum China, India, Japonia sunt evidențiate prin culori intense (valori GDP ridicate);
- Țări mai mici sau în curs de dezvoltare (ex: Nepal, Bhutan) sunt mai deschise la culoare;
- Absența valorii în fișierul CSV duce la afișarea țării cu fundal gri (lipsă de date).

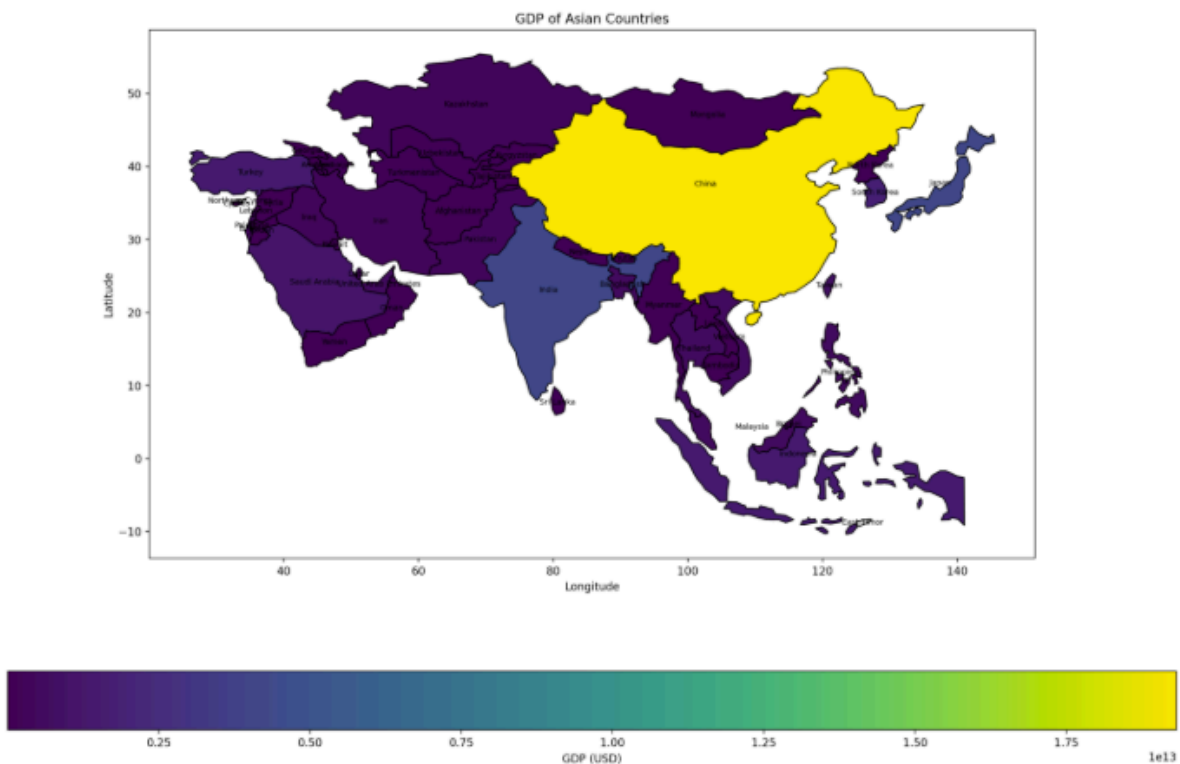


Figure 13. Harta GDP Asia

## 14. Clusterizare KMeans

În contextul analizei bursiere a acțiunilor Samsung, un obiectiv important este identificarea unor tipare ascunse în comportamentul de tranzacționare zilnic. O metodă eficientă în acest sens este

clusterizarea, care presupune gruparea observațiilor în funcție de similaritate. Pentru acest proiect, am utilizat algoritmul **KMeans** – o tehnică de învățare nesupervizată din biblioteca **scikit-learn**.

Am urmărit să grupăm sesiunile de tranzacționare ale acțiunilor Samsung pe baza a două variabile esențiale:

- **Close** – prețul de închidere al zilei;
- **Volume** – volumul tranzacționat.

Datele au fost standardizate anterior, pentru a evita dezechilibrele între scările valorilor (**StandardScaler**), rezultând cadrul **df\_scaled\_model**.

### 1. Alegerea numărului optim de clustere – „k”

Am aplicat două metode complementare pentru identificarea valorii potrivite a lui k:

Elbow Method – alegerea vizuală a lui k:

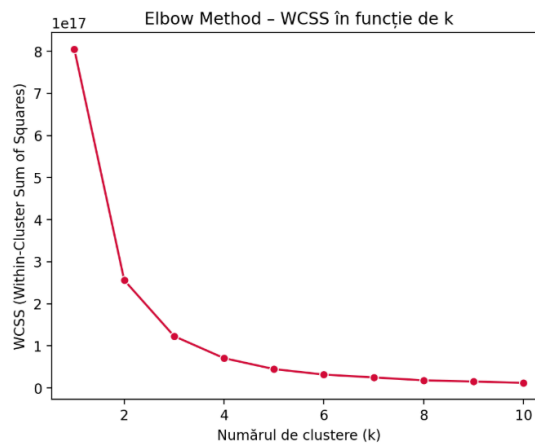


Figure 14. Metoda Elbow în python

**Elbow Method** (Metoda „cotului”):

- Se antrenează modele KMeans pentru valori k = 1..10;
- Se calculează pentru fiecare model WCSS – „Within Cluster Sum of Squares”;
- Se reprezintă grafic valorile WCSS în funcție de k, căutând punctul în care curba „se frânge”.

```
wcss = []  
for k in range(1, 11):  
    model = KMeans(n_clusters=k)  
    model.fit(X)  
    wcss.append(model.inertia_)
```

### Silhouette Scores pentru k = 2..10:

Măsoară cât de apropiat este un punct de clusterul său comparativ cu celelalte cluster. Scorul este între -1 și 1:

~1 → punctul este bine încadrat

~0 → este la graniță între cluster

< 0 → probabil este pus greșit în cluster

	Silhouette Score
2	0.6533
3	0.5904
4	0.5694
5	0.5683
6	0.5532
7	0.5423
8	0.5430
9	0.5385
10	0.5416

Numărul optim de cluster, conform scorului Silhouette, este: **k = 2**

Figure 15. Scorul Silhouette

### Scorul Silhouette:

- Măsoară cât de bine este încadrată o observație în clusterul său (valori între -1 și 1);
- Se calculează pentru k = 2..10;
- Se alege k\_optimal ca fiind valoarea cu scorul maxim.

```
score = silhouette_score(X, labels)
```

După alegerea valorii k\_optimal, am antrenat modelul final:

```
model_final = KMeans(n_clusters=k_optimal)
```

```
cluster_labels = model_final.fit_predict(X)
```

"Etichetele de cluster sunt apoi adăugate în cadrul de date original:

```
df_final["Cluster"] = cluster_labels
```

### Vizualizarea clusterelor folosind scatterplot:

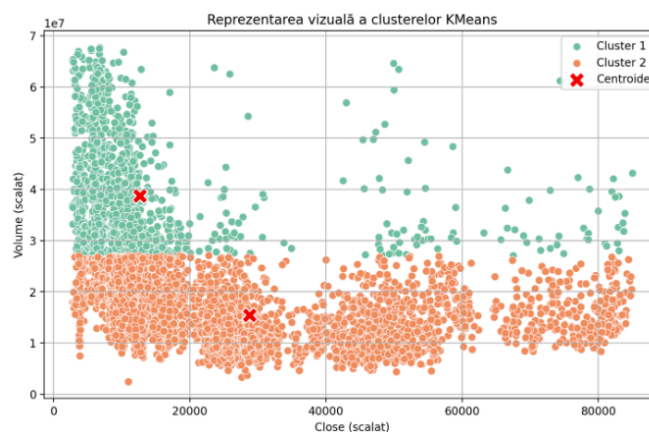


Figure 16. Vizualizare clusterelor folosind scatterplot

Am reprezentat vizual distribuția observațiilor în funcție de **Close** și **Volume**, evidențiind:

- fiecare cluster cu o culoare distinctă;

- centroidele fiecărui grup, marcate cu X roșii.

```
sns.scatterplot(x=X[:, 0], y=X[:, 1], hue=cluster_labels)
```

```
sns.scatterplot(x=model_final.cluster_centers[:, 0], y=model_final.cluster_centers[:, 1], marker='X')
```

## Interpretarea clusterelor identificate

### Număr de observații per cluster

	Cluster	Număr de rânduri
0	0	1322
1	1	4171

### Statistici descriptive pe fiecare cluster

Cluster	Open	High	Low	Close	Adj Close	Volume	Luna	Anotimp_larna	Anotimp_l
0	12752.31	12867.41	12600.6	12648.96	10112.07	38812324.84	6.38	0.24	
1	28755.14	29034.72	28494.21	28765.69	24801.79	15424506.98	6.5	0.24	

Valorile afișate mai sus reprezintă mediile variabilelor numerice din fiecare cluster.

Acestea ne pot ajuta să interpretăm semnificația fiecărui grup:

- Cluster cu **Close** mare și **Volume** mic → zile scumpe cu activitate redusă
- Cluster cu **Volume** mare → zile foarte active, posibil în perioade volatile

### Vizualizarea primelor 3 rânduri din fiecare cluster

#### Cluster 0

	Date	Open	High	Low	Close	Adj Close	Volume	Luna	Anotimp_larna	Anotimp_l
0	2000-01-06	5750	5780	5580	5620	4278.6865	54390000	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1	2000-01-07	5560	5670	5360	5540	4217.7803	40305000	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	2000-01-10	5600	5770	5580	5770	4392.8848	46880000	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### Cluster 1

	Date	Open	High	Low	Close	Adj Close	Volume	Luna	Anotimp_larna	Anotimp_l
0	2000-01-04	6000	6110	5660	6110	4651.7378	18013600	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1	2000-01-05	5800	6060	5520	5580	4248.2324	18013600	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	2000-02-01	5600	5680	5260	5320	4050.2859	18013600	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Clusterizarea ne ajută să identificăm tipare în comportamentul zilnic de tranzacționare,

cum ar fi grupuri cu volum ridicat și preț redus, sau invers.

Eticheta de cluster a fost salvată în setul final (`df_final["Cluster"]`).

Figure 17. Interpretarea clusterelor

Am analizat fiecare cluster:

- numărul de observații;
- mediile pe fiecare coloană numerică (`mean()`);
- exemple concrete din fiecare cluster (primele 3 rânduri).

Tipar identificat:

**Close** mare, **Volume** mic -> Zile scumpe dar cu tranzacții puține (stabilitate sau lipsă de interes)

**Volume** mare, **Close** mic -> Zile cu panică sau tranzacționare excesivă la prețuri mici

Ambele valori ridicate -> Zile intense, posibile în timpul anunțurilor importante

Această tehnică ajută la:

- identificarea tiparelor ascunse fără a avea o variabilă țintă;
- segmentarea temporală a sesiunilor de tranzacționare în grupe semnificative;
- explorarea oportunităților de investiții în funcție de comportamentul zilnic.

Eticheta de cluster a fost salvată în cadrul `df_final["Cluster"]`, putând fi folosită ulterior în alte modele predictive sau analize comparative.

## 15. Regresia logistică

În analiza evoluției financiare a acțiunilor Samsung, un pas important constă în anticiparea trendurilor viitoare ale pieței. În acest scop, am construit și testat un model de **regresie logistică binară**, având ca obiectiv predicția direcției trendului pe termen scurt (5 zile). Această abordare se încadrează în clasa problemelor de clasificare binară și utilizează biblioteci consacrate din Python, precum **scikit-learn**.

Pentru a putea aplica o regresie logistică, am redefinit problema în termeni binari:

- Dacă media valorilor **Close** din următoarele 5 zile este mai mare decât valoarea **Close** de astăzi → **TrendPozitiv\_5zile = 1**
- Altfel → **TrendPozitiv\_5zile = 0**

Această formulare presupune o privire anticipativă asupra datelor, permițând modelului să învețe tipare care preced un trend ascendent.

S-au construit manual mai multe variabile predictor (features) pentru a reflecta dinamica pieței:

```
df_trend5["Return"] = df_trend5["Close"].pct_change()
df_trend5["MA_3"] = df_trend5["Close"].rolling(3).mean()
df_trend5["MA_3_diff"] = df_trend5["Close"] - df_trend5["MA_3"]
df_trend5["Volume_change"] = df_trend5["Volume"].pct_change()
```

- **MA\_3\_diff** surprinde dacă prețul este peste/ sub media mobilă pe 3 zile
- **Volume\_change** reflectă impulsul de tranzacționare

Clasa țintă este definită astfel:

```
df_trend5["TrendPozitiv_5zile"] = (
    (df_trend5["Close"].shift(-1) + ... + df_trend5["Close"].shift(-5)) / 5 > df_trend5["Close"]
).astype(int)
```

Au fost utilizate și variabile de anotimp codificate prin One-Hot Encoding (Anotimp\_Iarna, etc.), astfel că vectorul de caracteristici final conține:

- Return, MA\_3\_diff, Volume\_change, Luna

- 4 variabile Anotimp\_\* → total: 8 predictori

Modelul logistic a fost implementat folosind:

```
model = LogisticRegression(class_weight='balanced', max_iter=1000)
```

- class\_weight='balanced': pentru a corecta dezechilibrul între clase
- max\_iter=1000: se asigură convergența modelului

Setul de date a fost împărțit în:

- 80% pentru antrenare (X\_train, y\_train)
- 20% pentru testare (X\_test, y\_test)

Modelul produce două tipuri de ieșiri:

- Eticheta binară (y\_pred)
- Probabilitatea apartenenței la clasa pozitivă (y\_prob)

## Rezultatele modelului:

**Acuratețe: 0.5023**

**F1 Score: 0.4986**

**ROC AUC Score: 0.5158**

*Figure 18. Rezultatele modelului regresiei logistice binare*

S-au evaluat 3 metrici esențiale:

- Acuratețe – procentul total de predicții corecte;
- F1 Score – media armonică între precizie și sensibilitate;
- ROC AUC Score – capacitatea de discriminare între clase.

```
accuracy_score(y_test, y_pred)
```

```
f1_score(y_test, y_pred)
```

```
roc_auc_score(y_test, y_prob)
```



Matricea de confuzie:

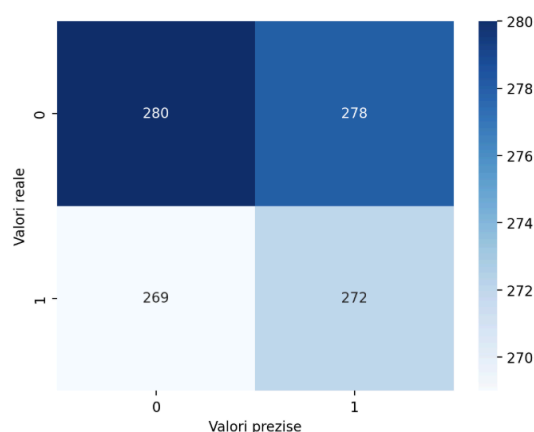


Figure 19. Matricea de confuzie regresie logistica binara

Curba ROC:

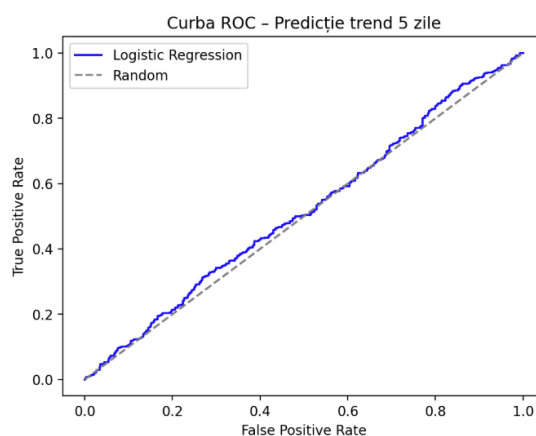


Figure 20. Curba ROC regresie logistica binara

De asemenea, a fost afișată:

- matrice de confuzie (True Positives, False Positives etc.);
- curbă ROC – care compară rata de adevărate pozitive vs. false pozitive.

Rezultatele modelului logistic au indicat o performanță modestă, cu un AUC ușor peste 0.5, ceea ce echivalează cu o capacitate de discriminare puțin mai bună decât o clasificare aleatorie.

Cu toate acestea, matricea de confuzie a fost echilibrată (~270 exemple corect clasificate pentru fiecare clasă), ceea ce indică faptul că modelul învață un tipar slab, dar real în date.

## 16. Regresia multiplă

În această secțiune am implementat un model de regresie liniară multiplă pentru a estima evoluția prețului de închidere (“Close”) al acțiunilor Samsung. Această tehnică aparține clasei modelelor explicative și este una dintre cele mai frecvent utilizate metode în statistică și econometrie pentru înțelegerea relației dintre o variabilă dependentă și mai mulți predicatori.

Pentru estimarea valorii “Close” am folosit variabilele dependente “Low” - prețul minim al sesiunii de tranzacționare, “Volume” - volumul de acțiuni tranzacționate și “Luna” - luna calendaristică, pentru a surprinde eventuale efecte sezoniere.

Pentru estimarea modelului, am utilizat biblioteca statsmodels și metoda OLS (Ordinary Least Squares), care urmărește minimizarea erorii pătratice totale dintre valorile reale și cele prezise.

- Am extras predictorii relevanți din setul de date curățat df\_final, folosind:

```
X = df_final[["Low", "Volume", "Luna"]]
y = df_final["Close"]
```

- A fost adăugată o constantă în matricea predictivă (add\_constant), astfel încât modelul să poată estima și interceptul:

```
X_const = sm.add_constant(X)
```

- Modelul a fost antrenat

```
model = sm.OLS(y, X_const).fit()
```

- Rezultatele au fost evaluate și interpretate pe baza:

- coeficientului de determinare  $R^2$
- valorii RMSE
- graficului real vs. prezis

#### Rezumatul modelului: ☞

OLS Regression Results

Dep. Variable:	Close	R-squared:	0.994
Model:	OLS	Adj. R-squared:	0.994
Method:	Least Squares	F-statistic:	2.837e+05
Date:	Fri, 30 May 2025	Prob (F-statistic):	0.00
Time:	15:55:01	Log-Likelihood:	-48249.
No. Observations:	5493	AIC:	9.651e+04
Df Residuals:	5489	BIC:	9.653e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	304.4222	76.204	3.995	0.000	155.032	453.812
Low	0.9999	0.001	835.521	0.000	0.998	1.002
Volume	-8.456e-06	1.94e-06	-4.356	0.000	-1.23e-05	-4.65e-06
Luna	14.4850	6.199	2.337	0.019	2.333	26.637

Omnibus:	15965.925	Durbin-Watson:	1.997
Prob(Omnibus):	0.000	Jarque-Bera (JB):	642208760.067
Skew:	-40.013	Prob(JB):	0.00
Kurtosis:	1676.180	Cond. No.	8.69e+07

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 8.69e+07. This might indicate that there are strong multicollinearity or other numerical problems.

Figure 21. Rezumatul modelului regresie multiple

Modelul a avut un coeficient  $R^2 \approx 0.994$ , ceea ce indică faptul că **99,4% din variația prețului Close** este explicată prin combinația liniară a celor trei variabile alese. Acest rezultat este excepțional din punct de vedere statistic, demonstrând o capacitate ridicată de ajustare.

Coeficientul lui **Low** este foarte apropiat de 1, ceea ce este așteptat: în piețele bursiere, prețul de închidere este adesea apropiat de minimul sesiunii.

Coeficientul lui **Volume** este negativ și semnificativ, sugerând că zilele cu volum mare de tranzacționare pot corespunde unor corecții bursiere sau episoade de volatilitate ridicată.

Variabila **Luna** introduce o ușoară sezonality, putând reflecta comportamente recurente ale investitorilor în funcție de perioadele anului (ex. început sau final de trimestru).

Grafic: Valori reale vs. Valori prezise

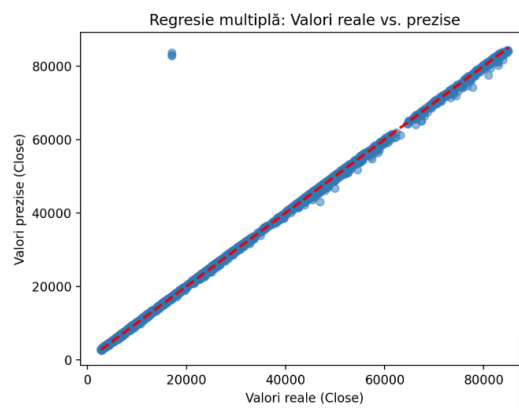


Figure 22. Grafic regresie multiplă

Pentru o analiză vizuală, am realizat un **grafic de tip scatterplot** care compară valorile reale **Close** cu cele prezise. Punctele s-au aliniat aproape perfect de-a lungul diagonalei  $y = x$ , confirmând calitatea predicțiilor.

# Capitolul SAS

## 1. Crearea unui set de date SAS din fișiere externe

Importarea datelor dintr-un fișier .csv extern (în cazul nostru, Samsung.csv) într-un set de date SAS pentru a putea fi utilizat în analizele viitoare.

Informațiile necesare sunt: calea către fișier și cunoașterea structurii fișierului.

Vom folosi **infile** cu opțiunea **dsd** (delimiter-sensitive data) pentru fișiere CSV și **input** pentru a defini variabilele.

```
data samsung;  
    infile '/home/u64223636/ProiectPSW/Samsung.csv' dsd firstobs=2;  
    input Date :ymmdd10. Open High Low Close Volume;  
    format Date yymmdd10.;  
run;
```

Datele vor fi încărcate într-un set denumit “samsung”:

Total rows: 5621 Total columns: 6

	Date	Open	High	Low	Close	Volume
1	2000-01-04	6000	6110	5660	6110	4651.737793
2	2000-01-05	5800	6060	5520	5580	4248.232422
3	2000-01-06	5750	5780	5580	5620	4278.686523
4	2000-01-07	5560	5670	5360	5540	4217.780273
5	2000-01-10	5600	5770	5580	5770	4392.884766
6	2000-01-11	5820	6100	5770	5770	4392.884766
7	2000-01-12	5610	5740	5600	5720	4354.818359
8	2000-01-13	5600	5740	5560	5710	4347.205078
9	2000-01-14	5720	5880	5680	5830	4438.565918
10	2000-01-17	6000	6180	5920	6100	4644.125
11	2000-01-18	6160	6160	5980	6100	4644.125
12	2000-01-19	6000	6040	5960	5960	4537.538574
13	2000-01-20	5860	6040	5820	6040	4598.444824

Figure 23. Setul de date “samsung”.

Pentru a evita pierderea datelor la închiderea sesiunii, am creat un set de date permanent denumit “samsung\_perm” folosind:

```
libname proiect '/home/u64223636/ProiectPSW';  
data proiect.samsung_perm;  
    set samsung;  
run;
```

Astfel, vom putea folosi oricând fișierul “samsung\_perm.sas7bdat”.

## 2. Crearea și folosirea de formate definite de utilizator

Vom defini formate cu scopul de a transforma variabile brute în forme ușor interpretabile. Astfel, vom clasifica valorile de închidere (Close) în categorii intuitive, precum Scăzut, Mediu și Ridicat. Variabila numerică Close este necesară.

Acest lucru se realizează prin utilizarea *proc format*:

```
proc format;
  value pret_fmt
    low - <5000 = 'Scăzut'
    5000 - <8000 = 'Mediu'
    8000 - high = 'Ridicat';
run;
data proiect.samsung_perm;
  set proiect.samsung_perm;
  length CategorieClose $10;
  CategorieClose = put(Close, pret_fmt.);
run;
title "Primele obs cu clasificarea pe Close";
proc print data=proiect.samsung_perm(obs=10);
  var Date Close CategorieClose;
run;
```

**Primele obs cu clasificarea pe Close**

Obs	Date	Close	CategorieClose
1	2000-01-04	6110	Mediu
2	2000-01-05	5580	Mediu
3	2000-01-06	5620	Mediu
4	2000-01-07	5540	Mediu
5	2000-01-10	5770	Mediu
6	2000-01-11	5770	Mediu
7	2000-01-12	5720	Mediu
8	2000-01-13	5710	Mediu
9	2000-01-14	5830	Mediu
10	2000-01-17	6100	Mediu

Figure 24. Observații clasificate

### 3. Procesarea iterativă și condițională a datelor

În analiza datelor financiare istorice ale companiei Samsung, procesarea condițională este esențială pentru clasificarea și interpretarea evoluțiilor pieței.

Vom grupa observațiile în funcție de anul în care au avut loc prin crearea unei variabile ce va clasifica fiecare observație în intervale de ani.

```
data proiect.samsung_perm;  
  set proiect.samsung_perm;  
  An = year(Date);  
  length GrupAni $ 12;  
  if 2000 <= An <= 2005 then GrupAni = '2000–2005';  
  else if 2006 <= An <= 2010 then GrupAni = '2006–2010';  
  else if 2011 <= An <= 2015 then GrupAni = '2011–2015';  
  else if 2016 <= An <= 2020 then GrupAni = '2016–2020';  
  else if An >= 2021 then GrupAni = '2021–2025';  
run;  
  
proc print data=proiect.samsung_perm(obs=10);  
  var Date An GrupAni;  
  title "Clasificare pe intervale de ani";  
run;
```

Clasificare pe intervale de ani			
Obs	Date	An	GrupAni
1	2000-01-04	2000	2000–2005
2	2000-01-05	2000	2000–2005
3	2000-01-06	2000	2000–2005
4	2000-01-07	2000	2000–2005
5	2000-01-10	2000	2000–2005
6	2000-01-11	2000	2000–2005
7	2000-01-12	2000	2000–2005
8	2000-01-13	2000	2000–2005
9	2000-01-14	2000	2000–2005
10	2000-01-17	2000	2000–2005

Figure 25. Clasificarea pe intervale de ani

Dorim să determinăm dacă în fiecare zi prețul a crescut, a scăzut sau a stagnat. S-au folosit valorile din Open și Close, iar metoda a constat în instrucțiuni if și missing() pentru a genera o variabilă Trend.

```
data proiect.samsung_perm;  
  set proiect.samsung_perm;  
  length Trend $ 10;  
  if missing(Open) or missing(Close) then Trend = "Necunoscut";  
  else if Close > Open then Trend = "Creștere";  
  else if Close < Open then Trend = "Scădere";  
  else Trend = "Stagnare";  
run;
```

```
proc print data=proiect.samsung_perm(obs=10);
  var Date Open Close Trend;
  title "Etichetare trend zilnic";
run;
```

Etichetare trend zilnic				
Obs	Date	Open	Close	Trend
1	2000-01-04	6000	6110	Creștere
2	2000-01-05	5800	5580	Scădere
3	2000-01-06	5750	5620	Scădere
4	2000-01-07	5560	5540	Scădere
5	2000-01-10	5600	5770	Creștere
6	2000-01-11	5820	5770	Scădere
7	2000-01-12	5610	5720	Creștere
8	2000-01-13	5600	5710	Creștere
9	2000-01-14	5720	5830	Creștere
10	2000-01-17	6000	6100	Creștere

*Figure 26. Etichetare trend zilnic*

Vom evalua cât de mare este volumul zilnic și îl încadrăm în trei niveluri. Au fost folosite valorile din Volume, iar metoda de calcul a fost select-when cu praguri numerice.

```
data proiect.samsung_perm;
  set proiect.samsung_perm;
  length NivelVolum $ 12;
  select;
    when (Volume < 5000000) NivelVolum = 'Mic';
    when (5000000 <= Volume < 20000000) NivelVolum = 'Mediu';
    when (Volume >= 20000000) NivelVolum = 'Ridicat';
    otherwise NivelVolum = 'Necunoscut';
  end;
run;

proc print data=proiect.samsung_perm(obs=10);
  var Date Volume NivelVolum;
run;
```

Obs	Date	Volume	NivelVolum
1	2000-01-04	4651.74	Mic
2	2000-01-05	4248.23	Mic
3	2000-01-06	4278.69	Mic
4	2000-01-07	4217.78	Mic
5	2000-01-10	4392.88	Mic
6	2000-01-11	4392.88	Mic
7	2000-01-12	4354.82	Mic
8	2000-01-13	4347.21	Mic
9	2000-01-14	4438.57	Mic
10	2000-01-17	4644.13	Mic

Figure 27. Încadrarea volumul de tranzacționare

Simulând o investiție de 10.000 KRW cu 5% dobândă, dorim să aflăm în câți ani se dublează. Am folosit do until și o condiție pe coloana Total.

```
data dublare;
  Total = 10000;
  Dobanda = 0.05;
  An = 0;
  do until (Total >= 20000);
    An + 1;
    Total + Dobanda * Total;
    output;
  end;
  format Total dollar12.2;
run;

proc print data=dublare;
run;
```

Obs	Total	Dobanda	An
1	\$10,500.00	0.05	1
2	\$11,025.00	0.05	2
3	\$11,576.25	0.05	3
4	\$12,155.06	0.05	4
5	\$12,762.82	0.05	5
6	\$13,400.96	0.05	6
7	\$14,071.00	0.05	7
8	\$14,774.55	0.05	8
9	\$15,513.28	0.05	9
10	\$16,288.95	0.05	10
11	\$17,103.39	0.05	11
12	\$17,958.56	0.05	12
13	\$18,856.49	0.05	13
14	\$19,799.32	0.05	14
15	\$20,789.28	0.05	15

Figure 28. Simulare investiție



## 4. Crearea de subseturi de date

Dorim să extragem doar înregistrările din anul 2020. Au fost folosite datele din Date, iar metoda de filtrare s-a bazat pe funcția year() și instrucțiunea if.

```
data samsung_2020;  
  set proiect.samsung_perm;  
  if year(Date) = 2020;  
run;  
  
proc print data=samsung_2020(obs=10);  
run;
```

Obs	Date	Open	High	Low	Close	Volume	CategorieClose	An	GrupAni	Trend	NivelVolum
1	2020-01-02	55500	56000	55000	55200	51557.59	Ridicat	2020	2016-2020	Scădere	Mic
2	2020-01-03	56000	56600	54900	55500	51837.80	Ridicat	2020	2016-2020	Scădere	Mic
3	2020-01-06	54900	55600	54600	55500	51837.80	Ridicat	2020	2016-2020	Creștere	Mic
4	2020-01-07	55700	56400	55600	55800	52118.01	Ridicat	2020	2016-2020	Creștere	Mic
5	2020-01-08	56200	57400	55900	56800	53052.02	Ridicat	2020	2016-2020	Creștere	Mic
6	2020-01-09	58400	58600	57400	58600	54733.24	Ridicat	2020	2016-2020	Creștere	Mic
7	2020-01-10	58800	59700	58300	59500	55573.85	Ridicat	2020	2016-2020	Creștere	Mic
8	2020-01-13	59600	60000	59100	60000	56040.87	Ridicat	2020	2016-2020	Creștere	Mic
9	2020-01-14	60400	61000	59900	60000	56040.87	Ridicat	2020	2016-2020	Scădere	Mic
10	2020-01-15	59500	59600	58900	59000	55106.85	Ridicat	2020	2016-2020	Scădere	Mic

Figure 29. Observațiile din 2020

Extragem observațiile din anul 2021 în care prețul de închidere a depășit 7000. Au fost folosite Date și Close, iar metoda aplicată a fost where în cadrul proc print.

```
proc print data=proiect.samsung_perm;  
  where year(Date) = 2021 and Close > 7000;  
  var Date Close;  
run;
```

Obs	Date	Close
5281	2021-01-04	83000
5282	2021-01-05	83900
5283	2021-01-06	82200
5284	2021-01-07	82900
5285	2021-01-08	88800
5286	2021-01-11	91000
5287	2021-01-12	90600
5288	2021-01-13	89700
5289	2021-01-14	89700
5290	2021-01-15	88000
5291	2021-01-18	85000
5292	2021-01-19	87000
5293	2021-01-20	87200
5294	2021-01-21	88100
5295	2021-01-22	86800

Figure 30. Observațiile din anul 2021 în care prețul de închidere a depășit 7000

Pentru a lucra doar cu variabilele importante pentru vizualizare, au fost selectate Date, Open, Close și Volume, iar metoda a fost keep= în instrucțiunea set.

```
data samsung_trimmed;
  set proiect.samsung_perm(keep=Date Open Close Volume);
run;

proc print data=samsung_trimmed(obs=10);
run;
```

Obs	Date	Open	Close	Volume
1	2000-01-04	6000	6110	4651.74
2	2000-01-05	5800	5580	4248.23
3	2000-01-06	5750	5620	4278.69
4	2000-01-07	5560	5540	4217.78
5	2000-01-10	5600	5770	4392.88
6	2000-01-11	5820	5770	4392.88
7	2000-01-12	5610	5720	4354.82
8	2000-01-13	5600	5710	4347.21
9	2000-01-14	5720	5830	4438.57
10	2000-01-17	6000	6100	4644.13

Figure 31. Alegerea variabilelor importante

## 5. Utilizarea de funcții SAS

Vom calcula media dintre Open și Close, iar apoi o vom rotunji. Pentru asta, am folosit funcțiile mean și round.

```
data proiect.samsung_perm;
  set proiect.samsung_perm;
  PretMediu = mean(Open, Close);
  PretRotunjit = round(Close, 100);
run;

proc print data=proiect.samsung_perm(obs=10);
  var Date Open Close PretMediu PretRotunjit;
run;
```

Obs	Date	Open	Close	PretMediu	PretRotunjit
1	2000-01-04	6000	6110	6055	6100
2	2000-01-05	5800	5580	5690	5600
3	2000-01-06	5750	5620	5685	5600
4	2000-01-07	5560	5540	5550	5500
5	2000-01-10	5600	5770	5685	5800
6	2000-01-11	5820	5770	5795	5800
7	2000-01-12	5610	5720	5665	5700
8	2000-01-13	5600	5710	5655	5700
9	2000-01-14	5720	5830	5775	5800
10	2000-01-17	6000	6100	6050	6100

Figure 32. Prețul mediu și rotunjit

Pentru a extrage luna și ziua pentru analiza sezonieră, am folosit funcțiile month și day aplicate pe coloana Date.

```
data proiect.samsung_perm;
  set proiect.samsung_perm;
  Luna = month(Date);
  Zi = day(Date);
run;

proc print data=proiect.samsung_perm(obs=10);
  var Date Luna Zi;
run;
```

Obs	Date	Luna	Zi
1	2000-01-04	1	4
2	2000-01-05	1	5
3	2000-01-06	1	6
4	2000-01-07	1	7
5	2000-01-10	1	10
6	2000-01-11	1	11
7	2000-01-12	1	12
8	2000-01-13	1	13
9	2000-01-14	1	14
10	2000-01-17	1	17

Figure 33. Extragerea lunii și a zilei

Dorim să măsurăm cât de mare a fost variația de preț într-o zi raportată la prețul de deschidere, construind astfel un indicator de volatilitate. Apoi, clasificăm această volatilitate în „Scăzută”, „Moderată” și „Ridicăată”.

```
data proiect.samsung_perm;
  set proiect.samsung_perm;

  if Open > 0 then VolRel = round((High - Low) / Open, 0.001);
  else VolRel = .;

  length TipVolatilitate $10;
  select;
    when (VolRel < 0.01) TipVolatilitate = "Scăzută";
    when (0.01 <= VolRel < 0.03) TipVolatilitate = "Moderată";
    when (VolRel >= 0.03) TipVolatilitate = "Ridicăată";
    otherwise TipVolatilitate = "Necunoscut";
  end;
run;

proc print data=proiect.samsung_perm(obs=10);
  var Date Open High Low VolRel TipVolatilitate;
  title "Calculul volatilității zilnice și clasificarea ei";
run;
```

Obs	Date	Open	High	Low	VolRel	TipVolatilitate
1	2000-01-04	6000	6110	5660	0.075	Ridicată
2	2000-01-05	5800	6060	5520	0.093	Ridicată
3	2000-01-06	5750	5780	5580	0.035	Ridicată
4	2000-01-07	5560	5670	5360	0.056	Ridicată
5	2000-01-10	5600	5770	5580	0.034	Ridicată
6	2000-01-11	5820	6100	5770	0.057	Ridicată
7	2000-01-12	5610	5740	5600	0.025	Moderată
8	2000-01-13	5600	5740	5560	0.032	Ridicată
9	2000-01-14	5720	5880	5680	0.035	Ridicată
10	2000-01-17	6000	6180	5920	0.043	Ridicată

Figure 34. Calculul volatilitatii zilnice și clasificarea ei

## 6. Combinarea seturilor de date

Presupunem că avem două fișiere cu date Samsung pe două perioade diferite. Vom împărți fișierul tău **Samsung.csv** în două părți și le vom concatena. La concatenare adăugăm o variabilă nouă, **Pret**, care este calculată în funcție de prețul de închidere.

```
data samsung1 samsung2;
  set proiect.samsung_perm;
  if _N_ <= 100 then output samsung1; /* _N_ este o variabilă automată în SAS care reține numărul */
  else output samsung2; /* observației curente. Cu acest IF punem primele 100 de observații să fie */
run;                                /* puse în samsung1 */

data samsung_total;
  set samsung1 samsung2;
  if Close < 5000 then Pret = 0;
  else if Close < 8000 then Pret = 60;
  else Pret = 25;
run;

proc print data=samsung_total (obs=10);
  title 'Concatenarea a doua subseturi Samsung si calculul pretului';
run;
```

Obs	Date	Open	High	Low	Close	Volume	CategorieClose	Pret
1	2000-01-04	6000	6110	5660	6110	4651.74	Mediu	60
2	2000-01-05	5800	6060	5520	5580	4248.23	Mediu	60
3	2000-01-06	5750	5780	5580	5620	4278.69	Mediu	60
4	2000-01-07	5560	5670	5360	5540	4217.78	Mediu	60
5	2000-01-10	5600	5770	5580	5770	4392.88	Mediu	60
6	2000-01-11	5820	6100	5770	5770	4392.88	Mediu	60
7	2000-01-12	5610	5740	5600	5720	4354.82	Mediu	60
8	2000-01-13	5600	5740	5560	5710	4347.21	Mediu	60
9	2000-01-14	5720	5880	5680	5830	4438.57	Mediu	60
10	2000-01-17	6000	6180	5920	6100	4644.13	Mediu	60

Figure 35. Concatenarea a două subseturi și calculul prețului

Pornind de la setul de date permanent [proiect.samsung\\_perm](#), , care conține informații bursiere despre acțiunile companiei Samsung (inclusiv prețul de închidere – **Close**), dorim să redenumim variabila **Close** în **PretFinal**.

Această redenumire reflectă mai bine scopul analizei ulterioare, unde prețul de închidere este interpretat drept prețul final relevant pentru decizii de tranzacționare.

```
data samsung_redenumit;
  set proiect.samsung_perm (rename=(Close=PretFinal));
run;

proc print data=samsung_redenumit(obs=10);
  title 'Redenumirea variabilei Close in PretFinal';
run;
```

Redenumirea variabilei Close in PretFinal							
Obs	Date	Open	High	Low	PretFinal	Volume	CategorieClose
1	2000-01-04	6000	6110	5880	6110	4651.74	Mediu
2	2000-01-05	5800	6060	5520	5580	4248.23	Mediu
3	2000-01-06	5750	5780	5580	5620	4278.69	Mediu
4	2000-01-07	5560	5670	5360	5540	4217.78	Mediu
5	2000-01-10	5600	5770	5580	5770	4392.88	Mediu
6	2000-01-11	5820	6100	5770	5770	4392.88	Mediu
7	2000-01-12	5610	5740	5600	5720	4354.82	Mediu
8	2000-01-13	5600	5740	5560	5710	4347.21	Mediu
9	2000-01-14	5720	5880	5680	5830	4438.57	Mediu
10	2000-01-17	6000	6180	5920	6100	4644.13	Mediu

*Figure 36. Redenumirea variabilei Close în PretFinal*

Setul de date original [proiect.samsung\\_perm](#), care conține valorile de tranzacționare zilnică pentru acțiunile Samsung, a fost împărțit anterior în două subseturi: [samsung1](#) și [samsung2](#). Aceste subseturi conțin observații diferite, dar structura lor este identică.

Pentru a interclasa aceste seturi, trebuie să le sortăm mai întâi după variabila **Date**. Apoi, folosind clauza **BY**, le combinăm astfel încât înregistrările să fie ordonate cronologic în noul set de date rezultat.

```
proc sort data=samsung1; by Date; run;
proc sort data=samsung2; by Date; run;
data samsung_interclasat;
  set samsung1 samsung2;
  by Date;
run;
```

```
proc print data=samsung_interclasat (obs=10);
  title 'Interclasarea datelor Samsung dupa Data';
run;
```

Interclasarea datelor Samsung dupa Data							
Obs	Date	Open	High	Low	Close	Volume	CategorieClose
1	2000-01-04	6000	6110	5660	6110	4651.74	Mediu
2	2000-01-05	5800	6060	5520	5580	4248.23	Mediu
3	2000-01-06	5750	5780	5580	5620	4278.69	Mediu
4	2000-01-07	5560	5670	5360	5540	4217.78	Mediu
5	2000-01-10	5600	5770	5580	5770	4392.88	Mediu
6	2000-01-11	5820	6100	5770	5770	4392.88	Mediu
7	2000-01-12	5610	5740	5600	5720	4354.82	Mediu
8	2000-01-13	5600	5740	5560	5710	4347.21	Mediu
9	2000-01-14	5720	5880	5680	5830	4438.57	Mediu
10	2000-01-17	6000	6180	5920	6100	4644.13	Mediu

Figure 37. Interclasarea datelor Samsung dupa Data

Fuziunea unu-la-unu este aplicabilă atunci când avem **câte o singură observație corespunzătoare pentru fiecare valoare a unei variabile cheie în ambele seturi**. Este esențial ca aceste seturi de date să fie **sortate în prealabil** după variabila comună, altfel SAS va returna o eroare sau va combina incorect datele.

În acest caz, vom crea un nou set de date care calculează volatilitatea zilnică (diferența dintre cel mai mare și cel mai mic preț al zilei: **High - Low**), și îl vom fuziona cu setul original pe baza variabilei **Date**. Astfel, fiecare înregistrare din setul final va conține, pe lângă datele bursiere standard, și o nouă variabilă calculată – **Volatilitate**.

```
data info_volatilitate;
  set proiect.samsung_perm (keep=Date Close);
  Volatilitate = High - Low;
run;

proc sort data=info_volatilitate; by Date; run;
proc sort data=proiect.samsung_perm; by Date; run;

data samsung_fuziune;
  merge proiect.samsung_perm info_volatilitate;
  by Date;
run;

proc print data=samsung_fuziune (obs=10);
  title 'Fuziunea unu-la-unu pe baza Date';
run;
```

### Fuziunea unu-la-unu pe baza Date

Obs	Date	Open	High	Low	Close	Volume	CategorieClose	Volatilitate
1	2000-01-04	6000	.	.	6110	4651.74	Mediu	.
2	2000-01-05	5800	.	.	5580	4248.23	Mediu	.
3	2000-01-06	5750	.	.	5620	4278.69	Mediu	.
4	2000-01-07	5580	.	.	5540	4217.78	Mediu	.
5	2000-01-10	5600	.	.	5770	4392.88	Mediu	.
6	2000-01-11	5820	.	.	5770	4392.88	Mediu	.
7	2000-01-12	5610	.	.	5720	4354.82	Mediu	.
8	2000-01-13	5600	.	.	5710	4347.21	Mediu	.
9	2000-01-14	5720	.	.	5830	4438.57	Mediu	.
10	2000-01-17	6000	.	.	6100	4644.13	Mediu	.

Figure 38. Fuziunea unu la unu pe baza Date

Ne dorim să îmbogățim setul de date **Samsung** cu o coloană care conține reduceri de preț în funcție de volumul tranzacționat.

Pentru ca fuziunea să fie posibilă, trebuie să existe o **variabilă comună** cu valori identice în ambele seturi de date. Deoarece în setul original **samsung\_perm**, volumul (**Volume**) are valori continue (ex. 13.245.000), iar în tabelul de reduceri (**volume\_red**) avem doar câteva praguri fixe (ex. 1.000.000, 5.000.000 etc.), a fost necesar să construim o versiune modificată a setului Samsung, în care valorile de **Volume** au fost rotunjite sau grupate pentru a corespunde celor din tabelul de reduceri.

```

/* Setul auxiliar cu reduceri definite pentru volume fixe */
data volume_red;
input Volume VolumeReducere;
    datalines;
1000000 0.02
5000000 0.05
10000000 0.10
15000000 0.15
;
run;

/* Rotunjirea volumului în setul Samsung pentru a permite fuziunea */
data samsung_std;
    set proiect.samsung_perm;

    if Volume < 3000000 then Volume = 1000000;
    else if Volume < 7500000 then Volume = 5000000;
    else if Volume < 12500000 then Volume = 10000000;
    else Volume = 15000000;
run;

/* Sortarea ambelor seturi de date după variabila comună */
proc sort data=samsung_std; by Volume; run;
proc sort data=volume_red; by Volume; run;

```

```

/* Fuziune reală pe baza variabilei Volume */
data samsung_merge;
  merge samsung_std(in=a) volume_red(in=b);
  by Volume;
  if a;
  PretRedus = round(Close * (1 - VolumeReducere), 0.01);
run;

proc print data=samsung_merge(obs=10);
  title 'MERGE Samsung cu Volume Reduceri pe Volume rotunjit';
run;

```

**MERGE Samsung cu Volume Reduceri pe Volume rotunjit**

Obs	Date	Open	High	Low	Close	Volume	CategorieClose	VolumeReducere	PretRedus
1	2000-01-04	6000	6110	5660	6110	1000000	Mediu	0.02	5987.8
2	2000-01-05	5800	6060	5520	5580	1000000	Mediu	0.02	5468.4
3	2000-01-06	5750	5780	5580	5620	1000000	Mediu	0.02	5507.6
4	2000-01-07	5560	5670	5360	5540	1000000	Mediu	0.02	5429.2
5	2000-01-10	5600	5770	5580	5770	1000000	Mediu	0.02	5654.6
6	2000-01-11	5820	6100	5770	5770	1000000	Mediu	0.02	5654.6
7	2000-01-12	5610	5740	5600	5720	1000000	Mediu	0.02	5605.6
8	2000-01-13	5600	5740	5560	5710	1000000	Mediu	0.02	5595.8
9	2000-01-14	5720	5880	5680	5830	1000000	Mediu	0.02	5713.4
10	2000-01-17	6000	6180	5920	6100	1000000	Mediu	0.02	5978.0

Figure 39. Setul de date Samsung cu Volume Reduceri cu ajutorul MERGE

Presupunem că dorim să comparăm datele despre acțiunile **Samsung** din două perioade diferite: una anterioară (**samsung\_vechi**) și una recentă (**samsung\_nou**). Scopul este să observăm care înregistrări apar doar într-unul din seturi și care sunt comune.

Pentru aceasta, folosim opțiunea **IN=** pentru a marca dacă o observație provine din fiecare dintre cele două surse. Rezultatul este afișat cu ajutorul instrucțiunii **PUT**, deoarece variabilele definite prin **IN=** sunt variabile temporare și nu pot fi afișate direct prin **PROC PRINT**.

```

Date=2000-01-04  inVechi=1  inNou=0  Close=6110
Date=2000-01-05  inVechi=1  inNou=0  Close=5580
Date=2000-01-06  inVechi=1  inNou=0  Close=5620
Date=2000-01-07  inVechi=1  inNou=0  Close=5540
Date=2000-01-10  inVechi=1  inNou=0  Close=5770
Date=2000-01-11  inVechi=1  inNou=0  Close=5770
Date=2000-01-12  inVechi=1  inNou=0  Close=5720
Date=2000-01-13  inVechi=1  inNou=0  Close=5710
Date=2000-01-14  inVechi=1  inNou=0  Close=5830
Date=2000-01-17  inVechi=1  inNou=0  Close=6100
Date=2000-01-18  inVechi=1  inNou=0  Close=6100
Date=2000-01-19  inVechi=1  inNou=0  Close=5960
Date=2000-01-20  inVechi=1  inNou=0  Close=6040
Date=2000-01-21  inVechi=1  inNou=0  Close=5880
Date=2000-01-24  inVechi=1  inNou=0  Close=5700
Date=2000-01-25  inVechi=1  inNou=0  Close=5440
Date=2000-01-26  inVechi=1  inNou=0  Close=5480
Date=2000-01-27  inVechi=1  inNou=0  Close=5520
Date=2000-01-28  inVechi=1  inNou=0  Close=5820
Date=2000-01-31  inVechi=1  inNou=0  Close=5580
Date=2000-02-01  inVechi=1  inNou=0  Close=5320
.....

```

Figure 40. Secțiune din rezultatul compararea datelor cu ajutorul IN=



## 7. Proceduri specifice SQL

Pentru a exemplifica utilizarea procedurilor specifice SQL am introdus un nou set de date EWY, ce este un ETF din Coreea de Sud care contine mai multe actiuni la firme Coreene precum Samsung (cu o proportie de aproximativ 25%), aceste ETF este reprezentat cu în moneda USD.

Se realizează o joncțiune internă (INNER JOIN) între tabelele Samsung și EWY, păstrând doar înregistrările care au aceeași dată (date) în ambele tabele. Astfel, sunt comparate valorile acțiunilor Samsung și ale ETF-ului EWY pentru zilele în care există date disponibile simultan.

```
PROC SQL;  
CREATE TABLE work.inner_join AS  
SELECT * FROM sams_renamed AS s  
INNER JOIN ewy_renamed AS e ON s.date = e.date;  
QUIT;
```

	Date	open_s	high_s	low_s	Close	volume_s	price	open_e	high_e	low_e	volume_e	change_pct
1	2000-05-15	6290	6530	6220	6440	4902.977539	19.44	19.44	19.44	19.44	5.00K	-1.27%
2	2000-05-16	6510	6800	6510	6800	5177.057129	19.81	19.81	19.81	19.81	0.30K	1.90%
3	2000-05-17	7000	7340	6920	6920	5268.416504	19.44	19.44	19.44	19.44	0.40K	-1.87%
4	2000-05-18	6740	7000	6680	6800	5177.057129	19.25	19.31	19.31	19.25	0.80K	-0.98%
5	2000-05-19	6740	6900	6600	6900	5253.189453	19.31	19.31	19.31	19.31	18.00K	0.31%
6	2000-05-22	6720	6830	6620	6670	5078.083008	17.44	17.94	17.94	17.44	70.50K	-9.68%
7	2000-05-23	6500	6700	6350	6380	4857.297852	17.88	17.94	18.12	17.88	51.30K	2.52%
8	2000-05-24	6160	6370	5920	6200	4720.257324	17.69	17.69	17.69	17.56	10.60K	-1.06%
9	2000-05-25	6330	6380	5930	6000	4567.991699	17.88	18.12	18.31	17.88	44.30K	1.07%
10	2000-05-26	5900	5990	5600	5600	4263.460938	16.94	16.94	17.06	16.94	21.00K	-5.26%
11	2000-05-30	5460	5460	5460	5460	4156.872559	17.94	18.12	18.12	17.94	32.10K	5.90%
12	2000-05-31	6000	6290	5960	6160	4689.805664	19.25	19.12	19.25	19.12	55.00K	7.30%
13	2000-06-01	6200	6410	6120	6300	4796.390625	19.19	19.31	19.38	19.19	11.50K	-0.31%
14	2000-06-02	6690	6900	6510	6620	5040.017578	20.5	20.5	20.5	20.19	10.40K	6.83%
15	2000-06-05	6920	6980	6680	6740	5131.377441	21	21.19	21.19	21	3.10K	2.44%
16	2000-06-06	6740	6740	6740	6740	5131.377441	20.75	21.12	21.12	20.75	10.20K	-1.19%
17	2000-06-07	6600	6980	6510	6800	5177.057129	21.25	21.5	21.5	20.5	44.10K	2.41%

Figure 41. Tabel INNER JOIN Samsung EWY

Se realizeaza o joncțiune LEFT JOIN pastrandu-se toate datele din tabelul Samsung,diferent dacă există corespondență în EWY. Dacă pentru o anumită zi nu există date în EWY, câmpurile asociate vor fi completate cu valori lipsă (missing).

```
PROC SQL;  
CREATE TABLE work.left_join AS  
SELECT * FROM sams_renamed AS s  
LEFT JOIN ewy_renamed AS e  
ON s.date = e.date;  
QUIT;
```

	Date	open_s	high_s	low_s	Close	volume_s	price	open_e	high_e	low_e	volume_e	change_pct
101	2000-05-23	6500	6700	6350	6380	4857.297852	17.88	17.94	18.12	17.88	51.30K	2.52%
102	2000-05-24	6160	6370	5920	6200	4720.257324	17.69	17.69	17.69	17.56	10.60K	-1.06%
103	2000-05-25	6330	6380	5930	6000	4567.991699	17.88	18.12	18.31	17.88	44.30K	1.07%
104	2000-05-26	5900	5990	5600	5600	4263.460938	16.94	16.94	17.06	16.94	21.00K	-5.26%
105	2000-05-29	5260	5680	5240	5440	4156.872559						
106	2000-05-30	5460	5460	5460	5460	4156.872559	17.94	18.12	18.12	17.94	32.10K	5.90%
107	2000-05-31	6000	6290	5960	6160	4689.805664	19.25	19.12	19.25	19.12	55.00K	7.30%
108	2000-06-01	6200	6410	6120	6300	4796.390625	19.19	19.31	19.38	19.19	11.50K	-0.31%
109	2000-06-02	6690	6900	6510	6620	5040.017578	20.5	20.19	20.5	20.19	10.40K	6.83%
110	2000-06-05	6920	6980	6680	6740	5131.377441	21	21.19	21.19	21	3.10K	2.44%
111	2000-06-06	6740	6740	6740	6740	5131.377441	20.75	21.12	21.12	20.75	10.20K	-1.19%
112	2000-06-07	6600	6980	6510	6800	5177.057129	21.25	21.5	21.5	20.5	44.10K	2.41%
113	2000-06-08	6660	6960	6530	6530	4971.497559	20.25	20.44	20.44	20.25	12.00K	-4.71%
114	2000-06-09	6600	7040	6530	7020	5344.551758	22.12	22.12	22.12	22.12	1.30K	9.23%
115	2000-06-12	7200	7280	7030	7190	5473.976563	22	22.56	22.56	22	12.10K	-0.54%
116	2000-06-13	7060	7100	6870	7060	5375.004883	21.12	21	21.31	21	14.50K	-4.00%

Figure 42. Tabel LEFTJOIN Samsung EWY

În momentul realizării unui FULL JOIN se combină toate datele din ambele tabele. Sunt incluse atât zilele care apar doar în Samsung, cât și cele care apar doar în EWY, rezultând un set complet de date pe întreaga perioadă.

```
PROC SQL;
CREATE TABLE work.full_join AS
SELECT * FROM sams_renamed AS s
FULL JOIN ewy_renamed AS e
ON s.date = e.date;
QUIT;
```

Total rows: 5839 Total columns: 12

Date	open_s	high_s	low_s	Close	volume_s	price	open_e	high_e	low_e	volume_e	chang
201 2000-10-10	3700	3810	3590	3660	2786.475586	15.31	15.44	15.44	15.31	0.50K	2.07%
202 2000-10-11	3450	3460	3210	3220	2451.48877	14.56	14.5	14.56	14.5	5.00K	-4.90%
203 2000-10-12	3140	3280	3140	3140	2390.582031	13.75	14	14	13.5	2.20K	-5.56%
204 2000-10-13	3000	3090	2830	3030	2306.835449	14	13.69	14	13.69	71.90K	1.82%
205 2000-10-16	3030	3030	3030	3030	2306.835449	14	14.06	14.06	14	0.70K	0.00%
206 2000-10-17	2980	3080	2730	2740	2086.049561	13.31	13.5	13.5	13.31	28.80K	-4.93%
207 2000-10-18	2540	2760	2420	2730	2078.436279	13.25	13.31	13.44	13.25	15.80K	-0.45%
208 2000-10-19	2730	3050	2690	2900	2207.863037	13.69	13.44	13.69	13.44	13.30K	3.32%
209 2000-10-20	3240	3330	3190	3330	2535.234863	14.56	14.5	14.62	14.5	5.00K	6.36%
210 2000-10-23	3490	3520	3160	3200	2436.263184	14.44	14.44	14.5	14	6.90K	-0.82%
211 2000-10-24	3180	3420	3100	3340	2542.848633	14.44	14.25	14.44	14.06	20.70K	0.00%
212 2000-10-25	3200	3260	3130	3210	2443.875488	13.88	14.38	14.38	13.88	35.70K	-3.88%
213 2000-10-26	2930	3030	2740	2890	2200.249512	13.69	13.75	14	13.69	2.80K	-1.37%
214 2000-10-27	2890	2890	2890	2890	2200.249512	13.5	13.56	13.62	13.25	5.00K	-1.39%
215 2000-10-30	2800	2880	2720	2750	2093.663086	13.19	13.69	13.69	13.19	20.80K	-2.30%
216 2000-10-31	2750	2850	2630	2850	2169.795654	13.69	13.62	13.75	13.56	4.00K	3.79%

Figure 43. Tabel FULL JOIN Samsung EWY

Determinarea prețurilor maxime și minime lunare se realizează pentru fiecare lună din fiecare an, valorile maxime și minime ale prețurilor de închidere (close pentru Samsung și price pentru EWY). Acest tip de agregare ajută la analiza volatilității lunare.

```
PROC SQL;
SELECT YEAR(date) AS an, MONTH(date) AS luna,
       MAX(close) AS max_samsung, MIN(close) AS min_samsung
FROM proiect.samsung_perm
GROUP BY calculated an, calculated luna;
QUIT;
```

/\* Preț maxim și minim lunar pentru EWY \*/

```
PROC SQL;
SELECT YEAR(date) AS an, MONTH(date) AS luna,
       MAX(price) AS max_ewy, MIN(price) AS min_ewy
FROM proiect.ewy
GROUP BY calculated an, calculated luna;
QUIT;
```

an	luna	max_samsung	min_samsung
2000	1	6110	5440
2000	2	5760	4800
2000	3	7660	5120
2000	4	7300	5400
2000	5	6920	5460
2000	6	7640	6300
2000	7	7760	5730
2000	8	6460	5470
2000	9	5540	3800
2000	10	3940	2730
2000	11	3780	3100
2000	12	3780	3100
2001	1	4800	3160
2001	2	4290	3740
2001	3	4340	3610
2001	4	4700	3640
2001	5	4700	4210
2001	6	4470	3720
2001	7	3940	3340
2001	8	3990	3660
2001	9	3930	2810
2001	10	3710	2810
2001	11	4620	3580
2001	12	5670	4490

an	luna	max_ewy	min_ewy
2000	5	19.81	16.94
2000	6	22.12	19.19
2000	7	22	17.94
2000	8	20.31	17.38
2000	9	18.81	14.75
2000	10	15.88	13.19
2000	11	15.19	12.62
2000	12	14.12	12
2001	1	15.94	12.44
2001	2	15.27	13.77
2001	3	14	12
2001	4	14.01	11.01
2001	5	15.36	13.87
2001	6	15.07	13.81
2001	7	14.26	12.3
2001	8	14.23	13.2
2001	9	13.53	10.81
2001	10	13.23	11.07
2001	11	17.27	13.49
2001	12	18.36	16.33

Figure 44. Preț maxim și minim lunar pentru Samsung și EWY

Se calculează media lunară a prețului de închidere pentru Samsung și EWY. Anul și luna sunt extrase direct din variabila DATE folosind funcțiile YEAR() și MONTH(). Gruparea se face pe baza acestor valori, permițând o analiză lunară sincronizată între cele două surse financiare.

```

PROC SQL;
CREATE TABLE work.medie_lunara AS
SELECT YEAR(s.date) AS an, MONTH(s.date) AS luna,
       MEAN(s.close) AS medie_samsung,
       MEAN(e.price) AS medie_ewy
FROM proiect.samsung_perm s
INNER JOIN proiect.ewy e
ON s.date = e.date
GROUP BY calculated an, calculated luna;
QUIT;

```

	an	luna	medie_samsung	medie_ewy
1	2000	5	6360.8333333	18.5225
2	2000	6	7011.3636364	20.684090909
3	2000	7	7047.5	20.589
4	2000	8	6099.0909091	19.034545455
5	2000	9	4518	16.579
6	2000	10	3270.4545455	14.378181818
7	2000	11	3335.7142857	14.285714286
8	2000	12	3339	13.041
9	2001	1	4065.7142857	14.531904762
10	2001	2	4029.4736842	14.578421053
11	2001	3	3916.6666667	12.945238095
12	2001	4	4115.5	12.552
13	2001	5	4493.6363636	14.62
14	2001	6	4073.8095238	14.504761905

Figure 45. Preț mediu lunar Samsung și EWY

Se caută zilele în care acțiunile Samsung au înregistrat o creștere (close > open), iar ETF-ul EWY a scăzut (valoare negativă în change\_pct). Pentru a face posibilă filtrarea numerică, câmpul change\_pct (stocat ca text cu %) este convertit într-o valoare numerică (change\_num) prin funcția INPUT după eliminarea simbolului % cu SUBSTR.

```
DATA proiect.ewy_clean;
  SET proiect.ewy;
  change_num = INPUT(SUBSTR(change_pct, 1, LENGTH(change_pct)-1), BEST.);
RUN;

PROC SQL;
CREATE TABLE work.zile_opuse AS
SELECT s.date, s.close, s.open, e.price, e.change_pct
FROM proiect.samsung_perm s
JOIN proiect.ewy_clean e ON s.date = e.date
WHERE s.close > s.open AND change_num < 0;
QUIT;
```

	Date	Close	Open	price	change_pct
1	2020-02-19	60200	59800	61.07	-0.02%
2	2020-02-21	59200	58800	58.37	-1.42%
3	2020-02-25	57900	56200	55.29	-0.05%
4	2020-03-05	57800	57600	56.32	-2.19%
5	2020-06-18	52300	52200	57.24	-0.31%
6	2020-06-19	52900	52600	56.75	-0.86%
7	2020-06-24	52900	51900	57.5	-0.36%
8	2020-06-26	53300	52800	57.07	-0.57%
9	2020-07-24	54200	54000	58.91	-0.14%
10	2020-09-03	56400	55600	63.3	-1.06%
11	2020-09-21	59200	59100	65.47	-0.86%
12	2020-09-23	58600	58400	63.43	-1.58%
13	2020-09-24	57800	57700	62.8	-0.99%
14	2020-10-12	60400	60000	67.99	-0.51%

Figure 46. Zile cu creștere Samsung & scădere EWY

Se selectează doar lunile ianuarie, februarie și martie, apoi se calculează media valorii de închidere pentru fiecare an. Se obține o imagine de ansamblu asupra performanței din primul trimestru.

```
DATA work.q1;
  SET sams.samsung;
  IF MONTH(date) IN (1, 2, 3);
RUN;
```

```

PROC SQL;
SELECT YEAR(date) AS an,
      MEAN(close) AS medie_Q1
FROM work.q1
GROUP BY calculated an;
QUIT;

```

an	medie_Q1
2000	5677.188
2001	3990
2002	6579.375
2003	6022.656
2004	10587.54
2005	9899.375
2006	13393.02
2007	11732.9
2008	11340.66
2009	10082.46
2010	15840
2011	18655.33
2012	22920.97
2013	29840
2014	25983.28
2015	27821.67
2016	23910.33
2017	39141.94
2018	49073.11
2019	43960.17
2020	55891.94
2021	83886.67
2022	73403.45

Figure 47. – Medie trimestru Q1 Samsung

Se identifică înregistrările în care volumul de tranzacționare este zero și se înlocuiesc cu valoarea lipsă (.), conform convenției SAS. Acest pas este important pentru evitarea erorilor în analizele ulterioare.

```

DATA work.samsung_clean;
SET proiect.samsung_perm;
IF volume = 0 THEN volume = .;
RUN;

```

	Date	Open	High	Low	Close	Volume
1401	2005-05-17	9800	9860	9670	9790	7453.441406
1402	2005-05-18	9890	9900	9780	9790	7453.441406
1403	2005-05-19	9860	10020	9850	9970	.
1404	2005-05-20	9990	10040	9900	9970	7590.47998
1405	2005-05-23	9920	9970	9890	9960	7582.866211
1406	2005-05-24	9930	9970	9860	9900	7537.187988
1407	2005-05-25	9870	9930	9670	9690	7377.304688
1408	2005-05-26	9630	9700	9590	9640	7339.237305
1409	2005-05-27	9800	9860	9730	9800	7461.052246
1410	2005-05-30	9800	9880	9800	9870	7514.348633
1411	2005-05-31	9850	9880	9730	9780	7445.827637

Figure 48. Remedierea înregistrărilor cu volum 0

Se adaugă o coloană nouă care calculează diferența procentuală dintre prețul de închidere Samsung și prețul ETF-ului EWY pentru aceeași zi. Formula aplicată este:

$((\text{close} - \text{price}) / \text{price}) * 100$

Această valoare permite comparația directă între cele două instrumente.

```
PROC SQL;
CREATE TABLE work.diferente_pct AS
SELECT s.date, s.close, e.price,
       ((s.close - e.price) / e.price) * 100 AS pct_diff
FROM proiect.samsung_perm s
JOIN proiect.ewy e ON s.date = e.date;
QUIT;
```

	Date	Close	price	pct_diff
1	2000-05-15	6440	19.44	33027.572016
2	2000-05-16	6800	19.81	34226.09793
3	2000-05-17	6920	19.44	35496.707819
4	2000-05-18	6800	19.25	35224.675325
5	2000-05-19	6900	19.31	35632.780943
6	2000-05-22	6670	17.44	38145.412844
7	2000-05-23	6380	17.88	35582.326622
8	2000-05-24	6200	17.69	34948.049746
9	2000-05-25	6000	17.88	33457.04698
10	2000-05-26	5600	16.94	32957.85124
11	2000-05-30	5460	17.94	30334.782609
12	2000-05-31	6160	19.25	31900

Figure 49. Diferența procentuală Samsung vs EWY

Se extrag, pentru fiecare lună din fiecare an, cele mai mari variații procentuale înregistrate de ETF-ul EWY (col. change\_pct). Această analiză permite evidențierea celor mai volatile perioade din punct de vedere al performanței pieței.

```
PROC SQL;
SELECT YEAR(date) AS an, MONTH(date) AS luna,
       MAX(change_num) AS max_var_pct
FROM proiect.ewy_clean
GROUP BY calculated an, calculated luna;
QUIT;
```

an	luna	max_var_pct
2000	5	7.3
2000	6	9.23
2000	7	3.29
2000	8	6.44
2000	9	5.03
2000	10	6.36
2000	11	4.59
2000	12	6.51
2001	1	11.01
2001	2	3.23
2001	3	5.64

Figure 50. Identificarea lunilor cu variații maxime EWY

## 8. Lucrul cu masive

Calcularea Volatilității și etichetare în funcție de prag

```
DATA proiect.samsung_volatil;  
  SET proiect.samsung_perm;  
  
  ARRAY val[2] High Low;  
  
  * Calculăm volatilitatea zilnică;  
  Volatilitate = ABS(val[1] - val[2]);  
  
  * Clasificăm volatilitatea;  
  IF Volatilitate > 200 THEN Tip = 'Ridicata';  
  ELSE IF Volatilitate > 100 THEN Tip = 'Medie';  
  ELSE Tip = 'Scazuta';  
  
RUN;  
  
PROC PRINT DATA=proiect.samsung_volatil (obs=10);  
  TITLE 'Volatilitatea Samsung și clasificarea';  
RUN;
```

Volatilitatea Samsung și clasificarea								
Obs	Date	Open	High	Low	Close	Volume	Volatilitate	Tip
1	2000-01-04	6000	6110	5660	6110	4651.74	450	Ridicata
2	2000-01-05	5800	6060	5520	5580	4248.23	540	Ridicata
3	2000-01-06	5750	5780	5580	5620	4278.69	200	Medie
4	2000-01-07	5580	5670	5360	5540	4217.78	310	Ridicata
5	2000-01-10	5600	5770	5580	5770	4392.88	190	Medie
6	2000-01-11	5820	6100	5770	5770	4392.88	330	Ridicata
7	2000-01-12	5810	5740	5600	5720	4354.82	140	Medie
8	2000-01-13	5600	5740	5560	5710	4347.21	180	Medie
9	2000-01-14	5720	5880	5680	5830	4438.57	200	Medie
10	2000-01-17	6000	6180	5920	6100	4644.13	260	Ridicata

Figure 51. Volatilitatea Samsung și clasificare

Comparare zilnică între EWY și Samsung (masiv dublu)

```
DATA comparatie_diff;  
  SET proiect.comparatie;  
  
  ARRAY ewy[4] Open_EWY High_EWY Low_EWY Close_EWY;  
  ARRAY samsung[4] Open_Samsung High_Samsung Low_Samsung Close_Samsung;  
  ARRAY dif[4] Dif_Open Dif_High Dif_Low Dif_Close;  
  
  DO i = 1 TO 4;  
    dif[i] = samsung[i] - ewy[i];  
  END;  
  DROP i;  
RUN;
```

```
PROC PRINT DATA=comparatie_diff (obs=10);
  TITLE 'Diferențe zilnice Samsung vs EWY';
RUN;
```

Diferențe zilnice Samsung vs EWY														
Obs	data	Open_Samsung	High_Samsung	Low_Samsung	Close_Samsung	Price_EWY	Open_EWY	High_EWY	Low_EWY	Close_EWY	Dif_Open	Dif_High	Dif_Low	Dif_Close
1	15MAY2000	6290	6530	6220	6440	19.44	19.44	19.44	19.44	.	6270.56	6510.56	6200.56	.
2	16MAY2000	6510	6800	6510	6800	19.81	19.81	19.81	19.81	.	6490.19	6780.19	6490.19	.
3	17MAY2000	7000	7340	6920	6920	19.44	19.44	19.44	19.44	.	6980.56	7320.56	6900.56	.
4	18MAY2000	6740	7000	6680	6800	19.25	19.31	19.31	19.25	.	6720.69	6980.69	6660.75	.
5	19MAY2000	6740	6900	6600	6900	19.31	19.31	19.31	19.31	.	6720.69	6880.69	6580.69	.
6	22MAY2000	6720	6830	6620	6670	17.44	17.94	17.94	17.44	.	6702.06	6812.06	6602.56	.
7	23MAY2000	6500	6700	6350	6380	17.88	17.94	18.12	17.88	.	6482.06	6681.88	6332.12	.
8	24MAY2000	6160	6370	5920	6200	17.69	17.69	17.69	17.56	.	6142.31	6352.31	5902.44	.
9	25MAY2000	6330	6380	5930	6000	17.88	18.12	18.31	17.88	.	6311.88	6361.09	5912.12	.
10	26MAY2000	5900	5990	5600	5800	16.94	16.94	17.06	16.94	.	5883.06	5972.94	5583.06	.

Figure 52. Diferențe zilnice Samsung și EWY

## 9. Prelucrarea datelor prin crearea de rapoarte și aplicarea de analize statistice

Afișăm un raport detaliat privind evoluția valorii Close în fiecare an.

Au fost folosite datele Date și Close, iar ca metode de calcul s-au aplicat proc sort pentru sortare și proc print cu by, sum și etichete personalizate.

```
data samsung_raport;
  set proiect.samsung_perm;
  An = year(Date);
run;

proc sort data=samsung_raport;
  by An;
run;

proc print data=samsung_raport label sumlabel='Total #byval(An)' grandtotal_label='Total' noobs;
  by An;
  var Date Open Close;
  sum Close;
  format Date date9.;
  label Date = 'Data'
        Open = 'Deschidere'
        Close = 'Închidere';
  title "Raport anual: evoluția valorii de închidere";
run;
```



### Raport anual: evoluția valorii de închidere

An=2000		
Data	Deschidere	Inchidere
04JAN2000	6000	6110
05JAN2000	5800	5580
06JAN2000	5750	5620
07JAN2000	5560	5540
10JAN2000	5600	5770
11JAN2000	5820	5770
12JAN2000	5610	5720
13JAN2000	5600	5710
14JAN2000	5720	5830
17JAN2000	6000	6100
18JAN2000	6160	6100
19JAN2000	6000	5960
20JAN2000	5860	6040

Figure 53. Raport anual: evoluția valorii de închidere

Dorim să analizăm distribuția valorii de închidere a acțiunilor pentru a detecta medii, abateri și valori extreme.

Au fost folosite datele Close, iar ca metode procedura proc univariate cu plot, histogram, și nextrobs.

```
proc univariate data=proiect.samsung_perm plot;  
  var Close;  
  histogram Close;  
  id Date;  
  title "Statistici descriptive și distribuție pentru prețul de închidere";  
run;  
  
proc univariate data=proiect.samsung_perm nextrobs=5 nextrobs=0;  
  var Close;  
  id Date;  
  title "Valori extreme distincte pentru prețul de închidere";  
run;
```

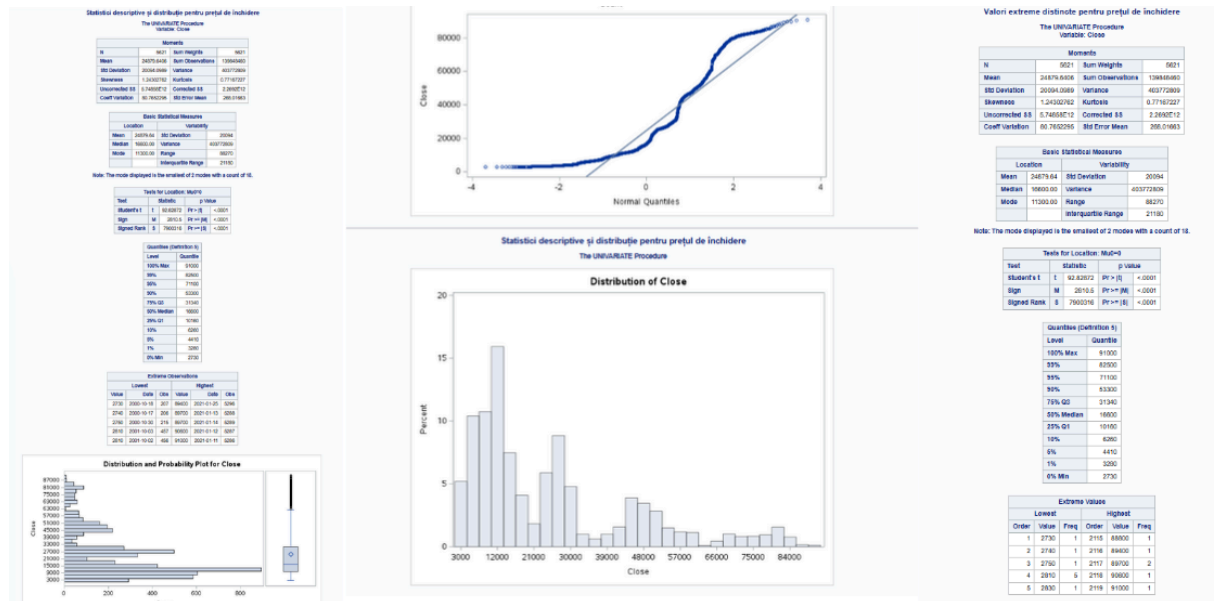


Figure 54. Analiza distribuției valorii de închidere

În vederea generării statisticilor agregate pentru fiecare an (medie, minim, maxim), au fost folosite coloanele Date, Open, Close, iar metodele au fost proc means cu by și var.

```
data samsung_an;
  set proiect.samsung_perm;
  An = year(Date);
run;

proc sort data=samsung_an;
  by An;
run;

proc means data=samsung_an n mean min max;
  by An;
  var Open Close;
  title "Statistici agregate anuale pentru Open și Close";
run;
```

Statistici agregate anuale pentru Open și Close				
The MEANS Procedure				
An=2000				
Variable	N	Mean	Minimum	Maximum
Open	259	5370.12	2540.00	7780.00
Close	259	5361.58	2730.00	7760.00
An=2001				
Variable	N	Mean	Minimum	Maximum
Open	261	3889.23	2760.00	5600.00
Close	261	3898.74	2810.00	5670.00
An=2002				
Variable	N	Mean	Minimum	Maximum
Open	261	6850.34	5580.00	8460.00
Close	261	6856.90	5470.00	8640.00
An=2003				
Variable	N	Mean	Minimum	Maximum
Open	261	7480.69	5320.00	9600.00
Close	261	7488.24	5390.00	9600.00
An=2004				
Variable	N	Mean	Minimum	Maximum
Open	262	9675.00	8000.00	12740.00
Close	262	9674.69	8040.00	12740.00
An=2005				
Variable	N	Mean	Minimum	Maximum
Open	259	10722.93	8760.00	13200.00
Close	259	10733.36	8700.00	13220.00
An=2006				
Variable	N	Mean	Minimum	Maximum
Open	248	12768.47	10880.00	14760.00
Close	248	12762.42	10980.00	14800.00

Figure 55. Statistici agregate anuale pentru Open și Close

Pentru a analiza câte zile au fost marcate de o volatilitate scăzută, moderată sau ridicată, vom folosi coloanele create în exemplele precedente (VolRel, TipVolatilitate). Acest lucru va fi realizat prin intermediul metodei proc freq.

```
proc freq data=proiect.samsung_perm;
  tables TipVolatilitate / nocum nopercnt;
  title "Frecvența categoriilor de volatilitate zilnică";
run;
```

Frecvența categoriilor de volatilitate zilnică	
The FREQ Procedure	
TipVolatilitate	Frequency
Moderată	3985
Ridică	1273
Scăzută	363

Figure 56. Frecvența categoriilor de volatilitate zilnică

Dorim să evaluăm dacă există corelație între prețul de închidere (Close) și volumul tranzacționat (Volume).

Au fost folosite variabilele Close și Volume, iar ca metodă de calcul a fost aplicată procedura corr cu specificarea expresă a variabilelor în var și with.

```
proc corr data=proiect.samsung_perm;
  var Volume;
  with Close;
  title "Corelația dintre prețul de închidere și volumul tranzacționat";
run;
```



Figure 57. Corelația dintre prețul de închidere și volumul tranzacționat

Valoarea coeficientului Pearson este foarte apropiată de 1, ceea ce indică o corelație pozitivă foarte puternică între volumul tranzacționat (Volume) și prețul de închidere (Close). Cu alte cuvinte, în zilele cu volum mai mare, prețul de închidere tinde să fie mai mare.

În scopul verificării dacă volumul tranzacționat poate fi un predictor pentru prețul de închidere, voi folosi analiza de regresie liniară prin utilizarea Volume ca variabilă independentă și Close dependentă prin procedura reg.

```
proc reg data=proiect.samsung_perm;
  model Close = Volume;
  title "Regresie liniară: estimarea prețului Close în funcție de Volume";
run;
quit;
```

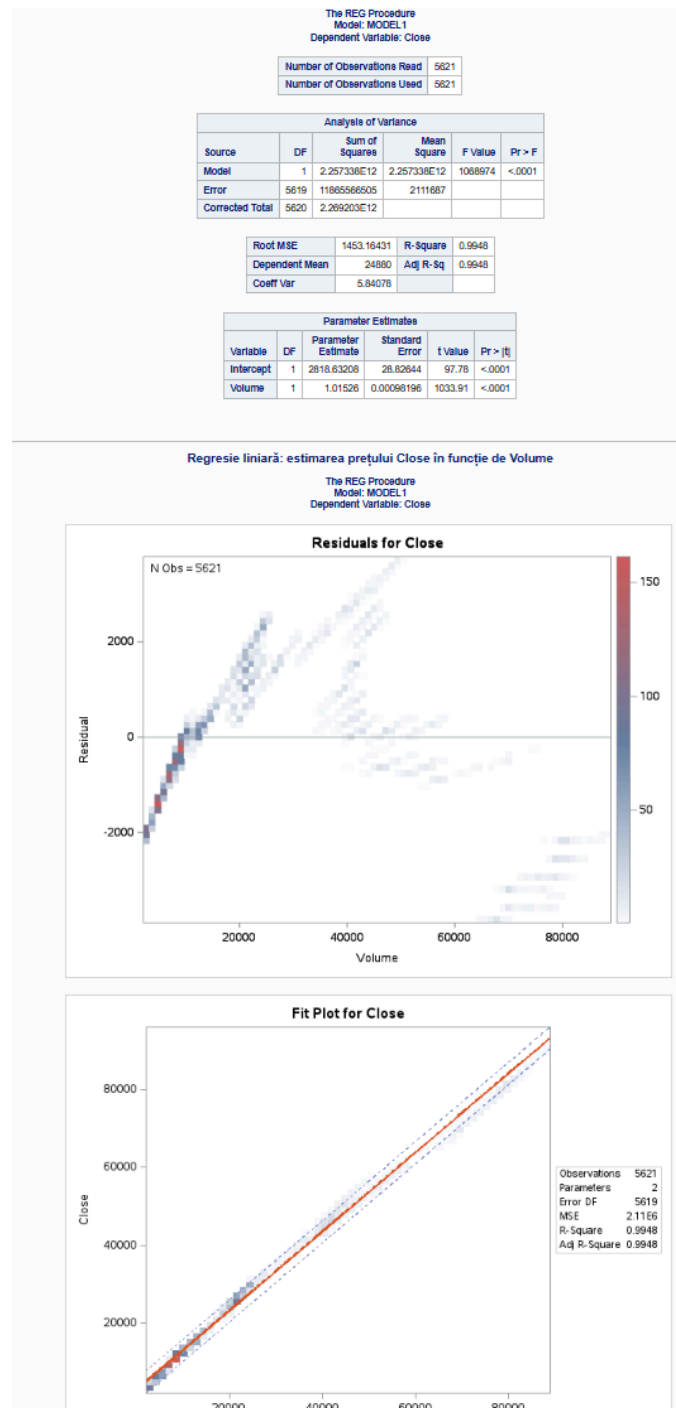


Figure 58. Regresie liniară: estimarea prețului Close în funcție de Volume

Modelul de regresie liniară indică o relație foarte puternică între volumul tranzacționat și prețul de închidere ( $R^2 = 0.9948$ ). Fiecare unitate în plus la Volume determină, în medie, o creștere cu 1.10 în Close, coeficientul fiind semnificativ statistic.

## 10. Generarea de grafice

Pentru a vizualiza cum a evoluat prețul Close pe parcursul anilor, au fost folosite Date și Close, iar ca metodă de calcul proc gplot cu simboluri și interpolare liniară.

```
symbol value=dot i=join color=blue width=1;  
title "Evolutia in timp a pretului de inchidere";
```

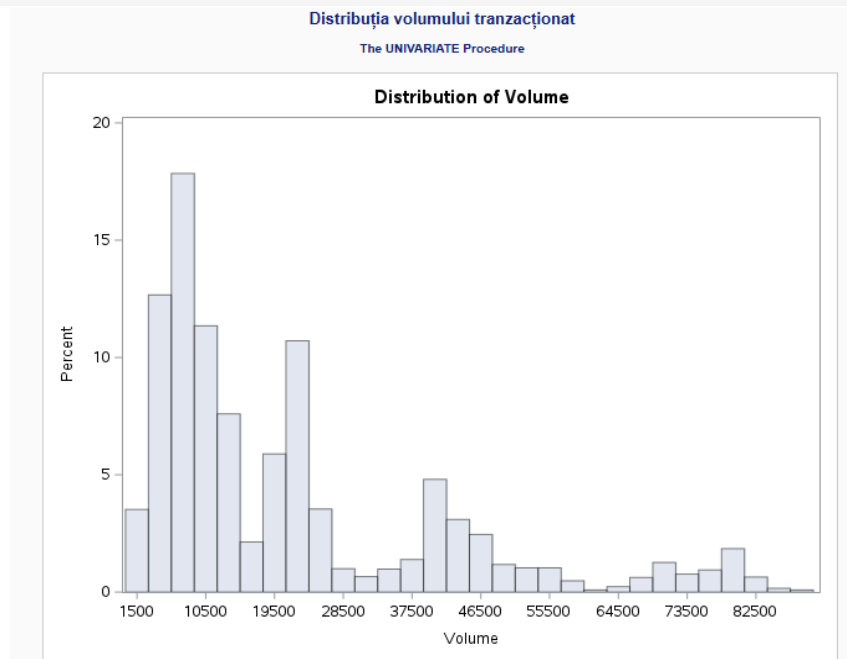
```
proc gplot data=proiect.samsung_perm;  
  plot Close*Date;  
run;  
quit;
```



*Figure 59. Evoluția în timp a prețului de închidere*

Dorim să vizualizăm distribuția volumului tranzacționat zilnic. Au fost folosite datele Volume, iar metoda aleasă a fost proc univariate cu histogram.

```
proc univariate data=proiect.samsung_perm noprint;
  var Volume;
  histogram Volume;
  title "Distribuția volumului tranzacționat";
run;
```



*Figure 60. Distribuția volumului de tranzacționat*

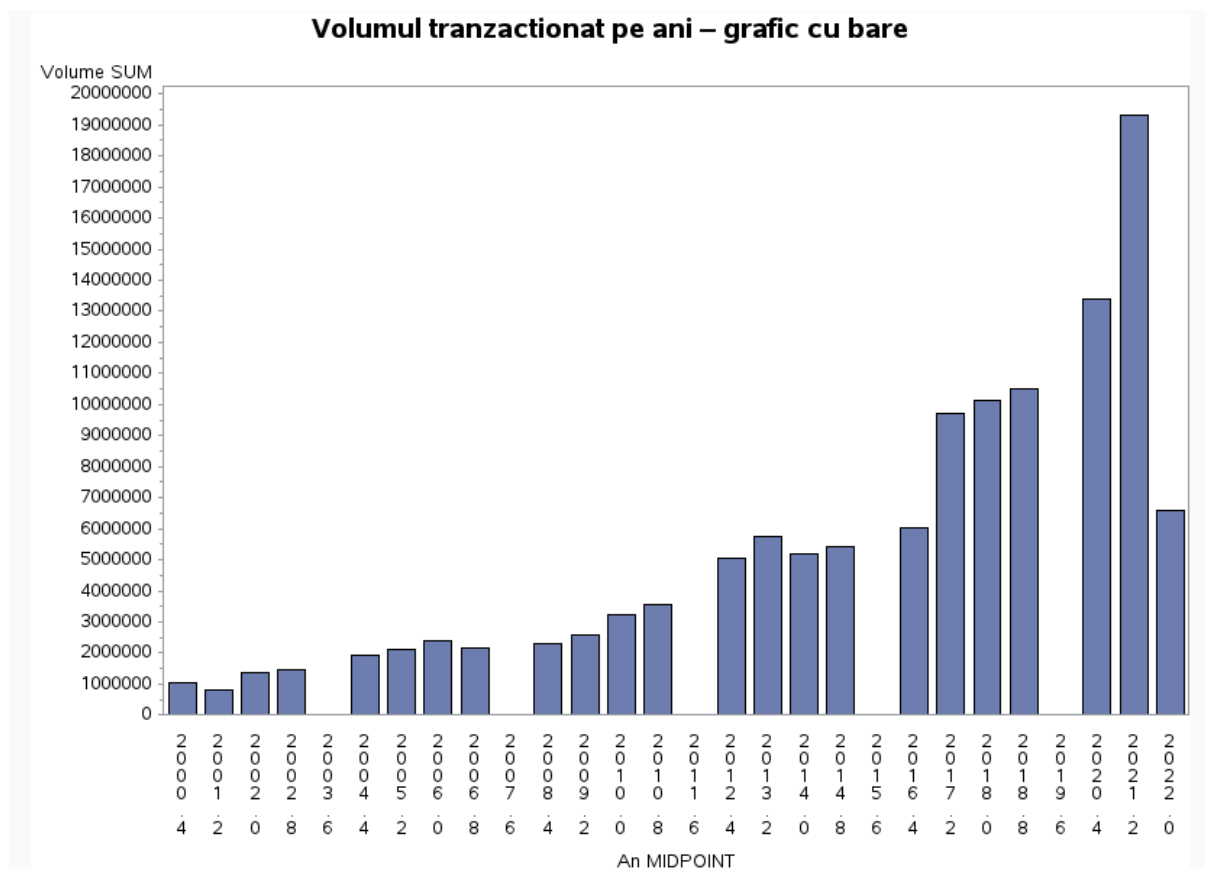
Vom afișa totalul volumului tranzacționat pentru fiecare an într-un grafic cu bare.

Au fost folosite year(Date) și Volume, cu proc gchart și opțiuni sumvar, type.

```
data vol_per_an;
  set proiect.samsung_perm;
  An = year(Date);
run;

title "Volumul tranzacționat pe ani – grafic cu bare";
pattern value=solid;

proc gchart data=vol_per_an;
  vbar An / sumvar=Volume type=sum;
run;
quit;
```



*Figure 61. Volumul tranzacționat pe ani*

## 11. Utilizarea SAS ML

În contextul investițiilor bursiere, anticiparea variației prețului acțiunilor este esențială pentru luarea unor decizii informate. Pe baza datelor din fișierul propus, se poate construi un model de învățare automată care să prezică dacă în ziua următoare prețul acțiunii va crește sau nu.

Această problemă poate fi formulată ca o clasificare binară, în care variabila țintă (Target) va avea valoarea:

- 1 dacă prețul de închidere a doua zi ( $Close_{t+1}$ ) este mai mare decât cel din ziua curentă ( $Close_t$ );
- 0 în caz contrar.

Pentru evitarea conflictelor ce pot fi cauzate de exercițiile anterioare, vom citi din nou setul de date, lucrând cu el din biblioteca `work`. La început, vom crea variabila țintă.



```

data work.samsung_target;
  set work.samsung;
  Close_t = Close;
  Close_t_plus1 = lag(Close);
  Target = (Close_t_plus1 > Close_t);
run;

data work.samsung_target;
  set work.samsung_target;
  if _N_ > 1;
run;

```

În plus, vom crea variabile cu valoare predictivă, și anume volatilitatea în timpul zilei, dacă acțiunea a crescut în timpul zilei și volatilitatea relativă la prețul de deschidere.

```

data work.samsung_features;
  set work.samsung_target;

  DailyRange = High - Low;
  PriceChange = Close - Open;
  Volatility = (High - Low) / Open;

  format DailyRange PriceChange Volatility 8.4;
run;

```

Cum nu avem valori lipsă, putem trece la standardizarea valorilor, dat fiind faptul că un astfel de algoritm este sensibil la magnitudine.

```

/* remove la target pt scalare */
data work.samsung_for_scaling;
  set work.samsung_features;
  retain Target;
run;

proc stdize data=work.samsung_for_scaling
  out=work.scaled_vars
  method=range;
  var Open High Low Close Volume DailyRange PriceChange Volatility;
run;

data work.samsung_scaled;
  merge work.scaled_vars work.samsung_features(keep=Target);
run;

data work.samsung_scaled;
  set work.samsung_scaled;
  Adj_Close = 'Adj Close'n;
  drop 'Adj Close'n;
run;

```

Până acum, setul de date este de forma:

Obs	Date	Open	High	Low	Close	Adj Close	Volume	Close_t	Close_t_plus1	Target	DailyRange	PriceChange	Volatility
1	2000-01-05	0.0371467839	0.0350914504	0.0355904488	0.0322873003	4248.232422	0.4547696617	5580	6110	1	0.0740	0.3609	0.6033
2	2000-01-06	0.0365770283	0.032113994	0.0362884704	0.0327404554	4278.686523	0.3312121304	5620	5580	0	0.0274	0.3709	0.2254
3	2000-01-07	0.0344120328	0.030944279	0.0337620579	0.0318341452	4217.780273	0.2454404287	5540	5620	1	0.0425	0.3832	0.3613
4	2000-01-10	0.0348678213	0.0320076563	0.0362884704	0.034439787	4392.884766	0.285479402	5770	5540	0	0.0260	0.4045	0.2198
5	2000-01-11	0.0373746582	0.0355168014	0.0384703721	0.034439787	4392.884766	0.363821619	5770	5770	0	0.0452	0.3799	0.3674
6	2000-01-12	0.0349817885	0.0316886431	0.0365181442	0.0338733432	4354.818359	0.17793746	5720	5770	1	0.0192	0.3978	0.1617
7	2000-01-13	0.0349878213	0.0316886431	0.0360587965	0.0337600544	4347.205078	0.250829705	5710	5720	1	0.0247	0.3978	0.2083
8	2000-01-14	0.0362351869	0.0331773713	0.0374368397	0.0351195197	4438.565918	0.3006728983	5830	5710	0	0.0274	0.3978	0.2266
9	2000-01-17	0.0394257065	0.0363675032	0.040192926	0.0381783165	4644.125	0.3897186311	6100	5830	0	0.0356	0.3966	0.2808
10	2000-01-18	0.0412488605	0.0361548277	0.0408819476	0.0381783165	4644.125	0.2756142861	6100	6100	0	0.0247	0.3788	0.1893

Figure 62. Setul de date ML

În continuare, vom împărți setul în două subseturi: de antrenament (70%) și de tesare (30%).

```
%let train_prop = 0.7;
```

```
proc surveyselect data=work.samsung_scaled out=work.samsung_split outall
  method=srs
  rate=&train_prop
  seed=123;
run;

data work.train work.test;
  set work.samsung_split;
  if selected then output work.train;
  else output work.test;
run;
```

The SURVEYSELECT Procedure	
Selection Method	Simple Random Sampling
Input Data Set	SAMSUNG_SCALED
Random Number Seed	123
Sampling Rate	0.7
Sample Size	3934
Selection Probability	0.7
Sampling Weight	0
Output Data Set	SAMSUNG_SPLIT

Figure 63. Împărțirea setului de date

Vom stoca variabilele independente într-o macrovariabilă:

```
proc sql noprint;
  select name into :indepVars separated by ' '
  from dictionary.columns
  where libname='WORK' and memname='SAMSUNG_SCALED'
    and name not in ('Target' 'Close_t' 'Close_t_plus1' 'selected');
quit;

%put &indepVars;
```

Astfel, datele sunt pregătite pentru a antrena, pentru început, un model de regresie logistică. Acesta va prezice probabilitatea ca Target să fie 1.

```
proc logistic data=work.train descending outmodel=work.logitmodel;
  model Target(event='1') = &indepVars;
  score data=work.test out=work.test_pred_logit outroc=roc_logit;
run;
```

Model Information	
Data Set	WORK.TRAIN
Response Variable	Target
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	3934
Number of Observations Used	3934

Response Profile		
Ordered Value	Target	Total Frequency
1	1	1818
2	0	2116

Probability modeled is Target=1.

Model Convergence Status	
Convergence criterion (GCONV=1E-8) satisfied.	

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	5433.087	3984.554
SC	5439.384	4041.051
-2 Log L	5431.087	3986.554

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	1464.5327	8	<.0001
Score	917.2773	8	<.0001
Wald	687.3251	8	<.0001

Figure 64. Output regresie logistica 1

DailyRange =	0.04658 * Intercept + 12.8822 * High - 11.9288 * Low
PriceChange =	0.4067 * Intercept - 9.80559 * Open + 9.86257 * Close

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-4.3132	0.9229	21.8408	<.0001
Date	1	0.000184	0.000054	9.1503	0.0025
Open	1	346.0	23.6599	213.8433	<.0001
High	1	129.9	24.9330	27.1475	<.0001
Low	1	157.8	33.5209	22.1487	<.0001
Close	1	-629.3	29.3402	459.9988	<.0001
Volume	1	0.3126	0.6743	0.2150	0.6429
DailyRange	0	0	.	.	.
PriceChange	0	0	.	.	.
Volatility	1	1.9229	0.8165	5.5461	0.0185
Adj_Close	1	-0.00003	0.000041	0.5146	0.4732

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Date	1.000	1.000	1.000
Open	>999.999	>999.999	>999.999
High	>999.999	>999.999	>999.999
Low	>999.999	>999.999	>999.999
Close	<0.001	<0.001	<0.001
Volume	1.387	0.365	5.126
Volatility	6.840	1.381	33.891
Adj_Close	1.000	1.000	1.000

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	85.3	Somers' D	0.707
Percent Discordant	14.7	Gamma	0.707
Percent Tied	0.0	Tau-a	0.351
Pairs	3846888	c	0.853

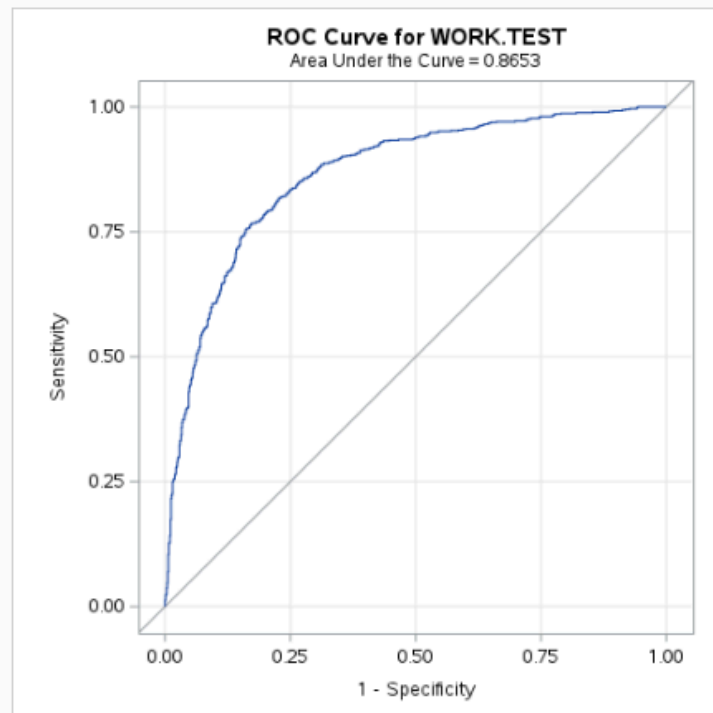


Figure 65. Output regresie logistica 2

Variabilele legate de preț (Open, High, Low, Close) au o influență majoră și semnificativă asupra probabilității ca prețul să crească în ziua următoare. Adj\_Close nu aduce valoare adăugată modelului.

$c = 0.853$  (AUC ROC) => Modelul are foarte bună capacitate de discriminare (85.3%) și clasifică corect în majoritatea cazurilor.

În continuare, am convertit probabilitățile în clase binare, afișând matricea de confuzie pentru cele două praguri (0.3 și 0.5).

```
data work.test_pred_logit;
  set work.test_pred_logit;
  pred_class_05 = (P_1 > 0.5);
  pred_class_03 = (P_1 > 0.3);
run;

title 'Matrice de confuzie – Regresie logistică, prag 0.5';
proc freq data=work.test_pred_logit;
  tables Target * pred_class_05 / nocol;
run;

title 'Matrice de confuzie – Regresie logistică, prag 0.3';
proc freq data=work.test_pred_logit;
  tables Target * pred_class_03 / nocol;
run;
```

**Matrice de confuzie – Regresie logistică, prag 0.5**

The FREQ Procedure

Frequency Percent Row Pct		Table of Target by pred_class_05		
		pred_class_05		
Target		0	1	Total
0	772 45.79 85.02	138 8.07 14.98	908 53.86	
1	208 12.34 26.74	570 33.81 73.26	778 46.14	
Total	980 58.13	708 41.87	1688 100.00	

**Matrice de confuzie – Regresie logistică, prag 0.3**

The FREQ Procedure

Frequency Percent Row Pct		Table of Target by pred_class_03		
		pred_class_03		
Target		0	1	Total
0	455 26.99 50.11	453 26.87 49.89	908 53.86	
1	48 2.85 6.17	730 43.30 93.83	778 46.14	
Total	503 29.83	1183 70.17	1688 100.00	

Figure 66. Matricea de confuzie

Pentru pragul de precizie de 0.5, am creat manual matricea de confuzie pentru afișarea unui raport de evaluare mai intuitiv.

```
data evaluare;
    set work.test_pred_logit;

    TP = (Target = 1 and pred_class_05 = 1);
    TN = (Target = 0 and pred_class_05 = 0);
    FP = (Target = 0 and pred_class_05 = 1);
    FN = (Target = 1 and pred_class_05 = 0);
run;

proc means data=evaluare sum noprint;
    var TP TN FP FN;
    output out=confuzie_sum sum=TP TN FP FN;
run;

data evaluare_finala;
    set confuzie_sum;

    Total = TP + TN + FP + FN;
    Accuracy = (TP + TN) / Total;
    Precision = TP / (TP + FP);
    Recall = TP / (TP + FN);
    F1 = 2 * (Precision * Recall) / (Precision + Recall);
run;

title "Raport de evaluare a modelului logistic (prag 0.5)";
proc print data=evaluare_finala label noobs;
    var TP TN FP FN Accuracy Precision Recall F1;
    label
        TP = "True Positive"
        TN = "True Negative"
        FP = "False Positive"
        FN = "False Negative"
        Accuracy = "Acuratețe"
        Precision = "Precizie"
        Recall = "Recall (Sensibilitate)"
        F1 = "F1-Score";
run;
```

Raport de evaluare a modelului logistic (prag 0.5)							
True Positive	True Negative	False Positive	False Negative	Acuratețe	Precizie	Recall (Sensibilitate)	F1-Score
570	772	136	208	0.79597	0.80737	0.73265	0.76819

Figure 67. Raport de evaluare a modelului logistic

Modelul de regresie logistică antrenat pentru a prezice creșterea zilnică a prețului acțiunilor Samsung a obținut rezultate foarte bune la pragul de decizie standard de 0.5. Cu o acuratețe de aproximativ 80%, modelul a fost capabil să clasifice corect majoritatea cazurilor. De asemenea, a prezentat o precizie ridicată (0.807) și un recall bun (0.733), reușind să detecteze majoritatea zilelor reale de creștere. Scorul F1 (0.768) reflectă un echilibru solid între precizie și sensibilitate, ceea ce face ca acest model să fie eficient.

# Bibliografie

Acțiuni Samsung Electronics -

<https://www.kaggle.com/datasets/ranugadisansagamage/samsung-stocks>

Distribuția globală a brandurilor de telefoane -

<https://www.johnsphones.com/research/the-most-popular-phone-brands-in-every-country>

Acțiuni EWY (Ishares Msci South Korea ETF)-

<https://www.investing.com/etfs/ishares-south-korea-index-historical-data>



## List Of Images

- [Figure 1. Exemplu de filtrare asupra datelor în Python](#)
- [Figure 2. Tratarea valorilor extreme în Python](#)
- [Figure 3. Label encoding pe anotimpuri în Python](#)
- [Figure 4. Analiza corelațiilor în Python](#)
- [Figure 5. Boxplot pentru distribuția datelor în Python](#)
- [Figure 6. Agregări pe baza mediei în Python](#)
- [Figure 7. Bar chart pe baza mediei coloanei Low generat în Python](#)
- [Figure 8. Distribuția în lume a brandurilor de telefoane](#)
- [Figure 9. Harta hidrologica a Coreei de Sud](#)
- [Figure 10. Harta vecinilor Coreei de Sud](#)
- [Figure 11. Zone din Coreea de Sud în care s-ar putea construi fabrici](#)
- [Figure 12. Distanța de la fiecare oraș din Coreea de Sud până la graniță vecinilor](#)
- [Figure 13. Harta GDP Asia](#)
- [Figure 14. Metoda Elbow în python](#)
- [Figure 15. Scorul Silhouette](#)
- [Figure 16. Vizualizare clusterelor folosind scatterplot](#)
- [Figure 17. Interpretarea clusterelor](#)
- [Figure 18. Rezultatele modelului regresiei logistice binare](#)
- [Figure 19. Matricea de confuzie regresie logistica binara](#)
- [Figure 20. Curba ROC regresie logistica binara](#)
- [Figure 21. Rezumatul modelului regresie multiple](#)
- [Figure 22. Grafic regresie multiplă](#)
- [Figure 23. Setul de date "samsung".](#)
- [Figure 24. Observatii clasificate](#)
- [Figure 25. Clasificarea pe intervale de ani](#)
- [Figure 26. Etichetare trend zilnic](#)
- [Figure 27. Încadrarea volumul de tranzactionare](#)
- [Figure 28. Simulare investiție](#)
- [Figure 29. Observatiile din 2020](#)
- [Figure 30. Observatiile din anul 2021 în care prețul de închidere a depășit 7000](#)
- [Figure 31. Alegerea variabilelor importante](#)
- [Figure 32. Prețul mediu și rotunjit](#)
- [Figure 33. Extragerea lunii și a zilei](#)
- [Figure 34. Calculul volatilitatii zilnice și clasificarea ei](#)
- [Figure 35. Concatenarea a două subseturi și calculul prețului](#)
- [Figure 36. Redenumirea variabilei Close în PretFinal](#)
- [Figure 37. Interclasarea datelor Samsung dupa Data](#)
- [Figure 38. Fuziunea unu la unu pe baza Date](#)
- [Figure 39. Setul de date Samsung cu Volume Reduceri cu ajutorul MERGE](#)
- [Figure 40. Secțiune din rezultatul compararea datelor cu ajutorul IN=](#)
- [Figure 41. Tabel INNER JOIN Samsung EWY](#)
- [Figure 42. Tabel LEFTJOIN Samsung EWY](#)
- [Figure 43. Tabel FULL JOIN Samsung EWY](#)
- [Figure 44. Preț maxim și minim lunar pentru Samsung și EWY](#)

- [Figure 45. Pret mediu lunar Samsung și EWY](#)
- [Figure 46. Zile cu creștere Samsung & scădere EWY](#)
- [Figure 47. – Medie trimestru Q1 Samsung](#)
- [Figure 48. Remedierea înregistrărilor cu volum 0](#)
- [Figure 49. Diferența procentuală Samsung vs EWY](#)
- [Figure 50. Identificarea lunilor cu variații maxime EWY](#)
- [Figure 51. Volatilitatea Samsung și clasificare](#)
- [Figure 52. Diferențe zilnice Samsung și EWY](#)
- [Figure 53. Raport anual: evoluția valorii de închidere](#)
- [Figure 54. Analiza distribuției valorii de închidere](#)
- [Figure 55. Statistici agregate anuale pentru Open și Close](#)
- [Figure 56. Frecvența categoriilor de volatilitate zilnică](#)
- [Figure 57. Corelația dintre prețul de închidere și volumul tranzacționat](#)
- [Figure 58. Regresie liniară: estimarea prețului Close în funcție de Volume](#)
- [Figure 59. Evoluția în timp a prețului de închidere](#)
- [Figure 60. Distribuția volumului de tranzacționat](#)
- [Figure 61. Volumul tranzacționat pe ani](#)
- [Figure 62. Setul de date ML](#)
- [Figure 63. Împărțirea setului de date](#)
- [Figure 64. Output regresie logistica 1](#)
- [Figure 65. Output regresie logistica 2](#)
- [Figure 66. Matricea de confuzie](#)
- [Figure 67. Raport de evaluare a modelului logistic](#)