Outliers

(Copyright)

Iván Valero Canales

In [ ]:
```
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt

df = yf.download(
    'AAPL' ,
    start = '2000-01-01' ,
    end = '2010-12-31' ,
    progress = False
)

df = df.loc[:,['Adj Close']]
df.rename(columns = {'Adj Close' : 'adj_close'}, inplace = True)
df

# calculo los rendimientos simples

df['rendimiento_simple'] = df.adj_close.pct_change()
df

# calculo la media aritmetica (movil) y la desviacion estandar

df_rolling = df[['rendimiento_simple']].rolling(window = 21).agg(['mean' , 'std'
df_rolling.columns = df_rolling.columns.droplevel()
df_rolling

df_rolling['rendimiento_simple'] = df.rendimiento_simple # agregamos el rendimie
df_rolling['adj_close'] = df.adj_close # agregamos el adj_close
# tambien podemos usar otro metodo
# df_rolling = df.join(df_rolling)
df_rolling
```

Out[ ]:

Date	mean	std	rendimiento_simple	adj_close
2000-01-03	NaN	NaN	NaN	0.846127
2000-01-04	NaN	NaN	-0.084310	0.774790
2000-01-05	NaN	NaN	0.014633	0.786128
2000-01-06	NaN	NaN	-0.086539	0.718097
2000-01-07	NaN	NaN	0.047369	0.752113
...	...	...	...	...
2010-12-23	0.002274	0.008078	-0.004797	9.784272
2010-12-27	0.001496	0.007040	0.003337	9.816927
2010-12-28	0.001582	0.007040	0.002433	9.840809
2010-12-29	0.001273	0.006982	-0.000553	9.835369
2010-12-30	0.001894	0.005625	-0.005011	9.786084

2766 rows × 4 columns

In [ ]:
```
# renombramos la media y la desviacion

df_rolling.rename(columns = {'mean' : 'media_movil'} , inplace = True)
df_rolling.rename(columns = {'std' : 'desviacion_estandar'} , inplace = True)
df_rolling
```

Out[ ]:

Date	media_movil	desviacion_estandar	rendimiento_simple	adj_close
2000-01-03	NaN	NaN	NaN	0.846127
2000-01-04	NaN	NaN	-0.084310	0.774790
2000-01-05	NaN	NaN	0.014633	0.786128
2000-01-06	NaN	NaN	-0.086539	0.718097
2000-01-07	NaN	NaN	0.047369	0.752113
...	...	...	...	...
2010-12-23	0.002274	0.008078	-0.004797	9.784272
2010-12-27	0.001496	0.007040	0.003337	9.816927
2010-12-28	0.001582	0.007040	0.002433	9.840809
2010-12-29	0.001273	0.006982	-0.000553	9.835369
2010-12-30	0.001894	0.005625	-0.005011	9.786084

2766 rows × 4 columns

In [ ]:
```
# definimos el concepto de outlier como un valor que se encuentre fuera de la me

def identify_outliers(row , n_sigmas = 3):
    x = row['rendimiento_simple']
    mu = row['media_movil']
    sigma = row['desviacion_estandar']

    if (x > mu + 3*sigma) or (x < mu - 3*sigma): # definimos la condicion de out
        return 1
    else:
        return 0
```

In [ ]:
```
# identificar outliers

df_rolling['outlier'] = df_rolling.apply(identify_outliers , axis = 1)

outliers = df_rolling.loc[df_rolling['outlier'] == 1 , ['rendimiento_simple']]

df_rolling
```

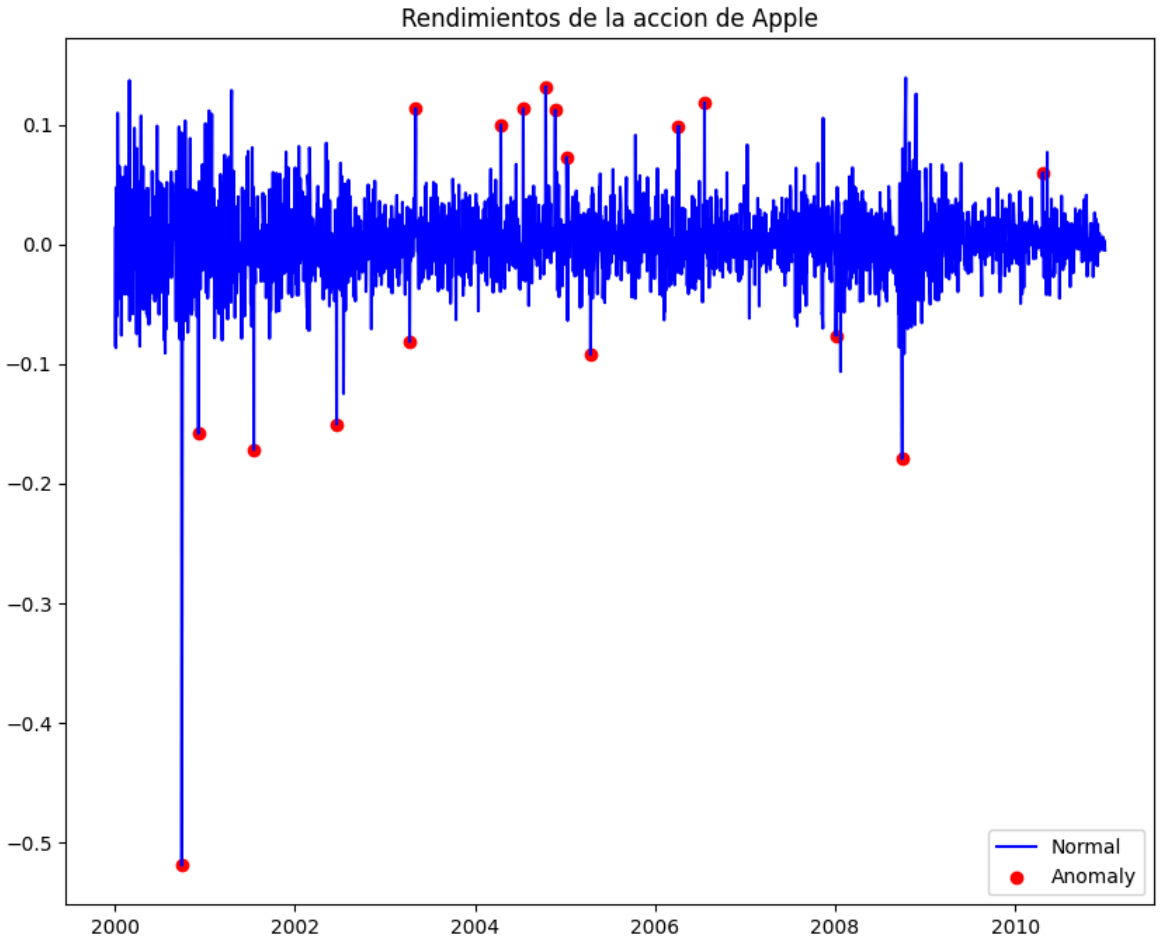
Out[ ]:

Date	media_movil	desviacion_estandar	rendimiento_simple	adj_close	outlier
2000-01-03	NaN	NaN	NaN	0.846127	0
2000-01-04	NaN	NaN	-0.084310	0.774790	0
2000-01-05	NaN	NaN	0.014633	0.786128	0
2000-01-06	NaN	NaN	-0.086539	0.718097	0
2000-01-07	NaN	NaN	0.047369	0.752113	0
...	...	...	...	...	...
2010-12-23	0.002274	0.008078	-0.004797	9.784272	0
2010-12-27	0.001496	0.007040	0.003337	9.816927	0
2010-12-28	0.001582	0.007040	0.002433	9.840809	0
2010-12-29	0.001273	0.006982	-0.000553	9.835369	0
2010-12-30	0.001894	0.005625	-0.005011	9.786084	0

2766 rows × 5 columns

In [ ]:
```
# graficamos los outliers
```

```
fg , ax = plt.subplots(figsize = (10,8))
ax.plot(df_rolling.index , df_rolling.rendimiento_simple , color = 'blue' ,
        label = 'Normal')
ax.scatter(outliers.index , outliers.rendimiento_simple , color = 'red' ,
           label = 'Anomaly')
ax.set_title("Rendimientos de la accion de Apple")
ax.legend(loc='lower right')
plt.show()
```



In [ ]:
```
outliers #fechas en las que han ocurrido dichas anomalias
```

Out[ ]:

Date	rendimiento_simple
2000-09-29	-0.518692
2000-12-06	-0.158089
2001-07-18	-0.171712
2002-06-19	-0.150372
2003-04-11	-0.081420
2003-05-05	0.113492
2004-04-15	0.099850
2004-07-15	0.113254
2004-10-14	0.131573
2004-11-22	0.112017
2005-01-07	0.072811
2005-04-14	-0.092105
2006-04-05	0.098742
2006-07-20	0.118299
2008-01-04	-0.076335
2008-09-29	-0.179195
2010-04-21	0.059814