TimeSeriesAnalysis

Castor

2024-11-04

Table of contents

Preface

R Python

- ;
- ACF AR, MA ARMA, ARIMA
  ;
- ARCH ARCH GARCH IGARCH GARCH-M EGARCH , TGARCH , APARCH GARCH ;
- VAR ;
-

1

Ruey S. Tsay R (Tsay 2013) An Introduction to Analysis of Financial Data with R

1.1

1.2

1.3

1.4

1.5

1.6

1.7

2

2.1

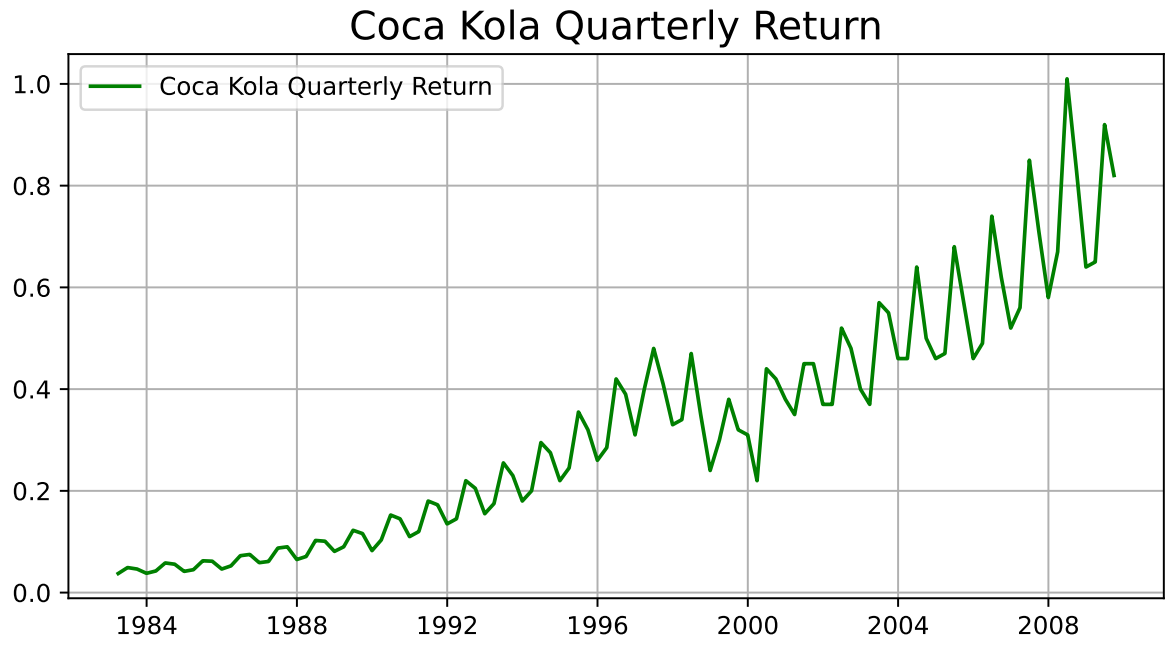
Ruey S. Tsay R (Tsay 2013) An Introduction to Analysis of Financial Data with R “ ”

-
- AR, MA, ARMA
-
-
-
-
-
-
-

2.1.1

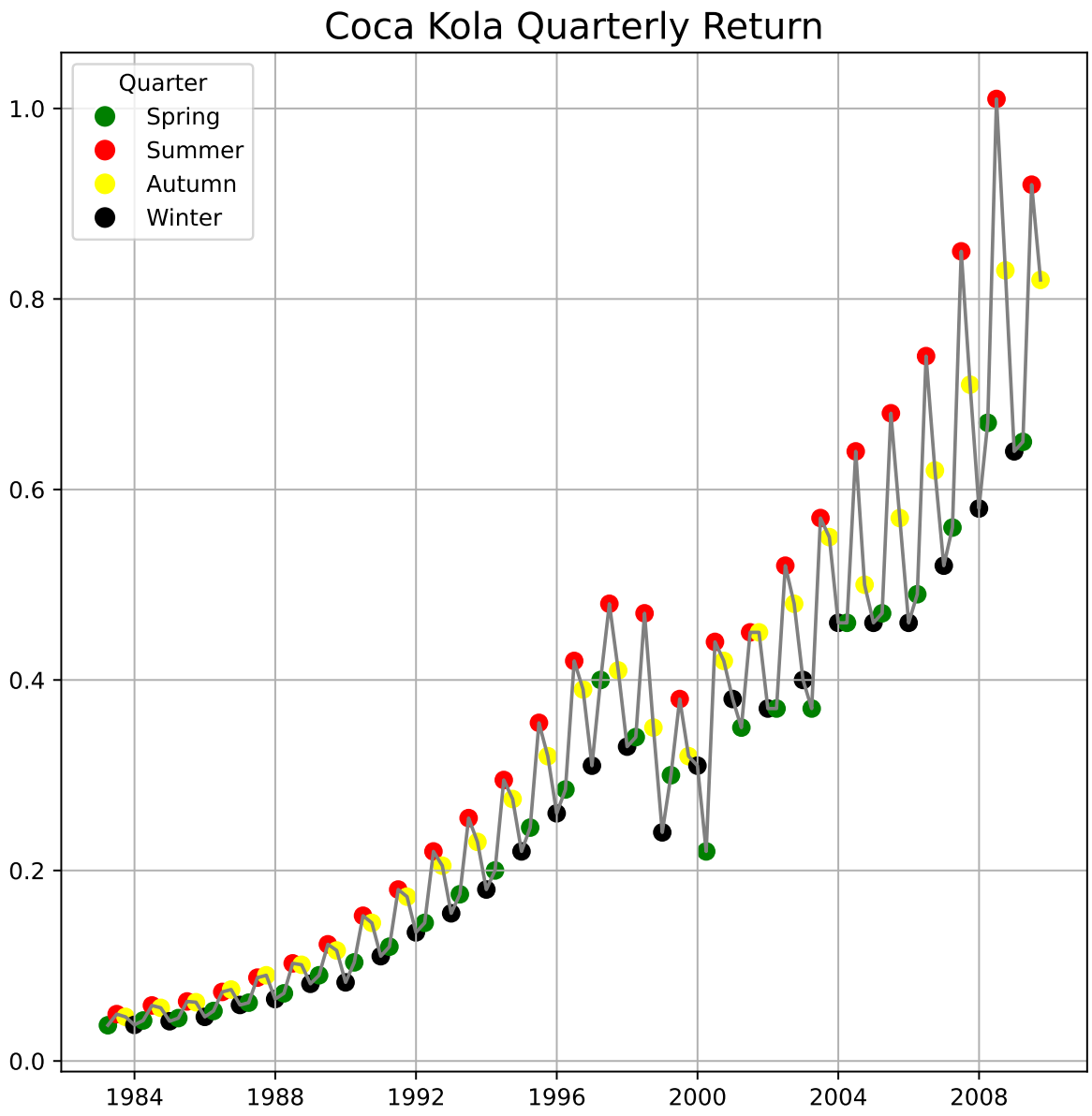
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

raw_data = []
with open("../ftsdata/q-ko-earns8309.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])


data["pends"] = pd.to_datetime(data["pends"], format="%Y%m%d")
data["anntime"] = pd.to_datetime(data["anntime"], format="%Y%m%d")
```

```
data["value"] = pd.to_numeric(data["value"])

plt.figure(figsize=(8, 4))
plt.plot(data["pends"], data['value'], label='Coca Kola Quarterly Return', color='green')
plt.title('Coca Kola Quarterly Return', fontsize=16)
plt.grid(True)
plt.legend()
plt.show()
```




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

raw_data = []
with open("../ftsdata/q-ko-earns8309.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])
```

```
data["pends"] = pd.to_datetime(data["pends"], format="%Y%m%d")
data["anntime"] = pd.to_datetime(data["anntime"], format="%Y%m%d")
data["value"] = pd.to_numeric(data["value"])

data['Date'] = pd.to_datetime(data['pends'])
data.set_index('Date', inplace=True)

data['Year'] = data.index.year
data['Quarter'] = data.index.quarter

cpal = ['green', 'red', 'yellow', 'black']

plt.figure(figsize=(8, 8))

plt.plot(data.index, data['value'], label='Coca Kola Quarterly Return', color='gray')

for i, row in data.iterrows():
    plt.scatter(row.name, row['value'], color=cpal[row['Quarter'] - 1], s=50)

plt.title('Coca Kola Quarterly Return', fontsize=16)
plt.grid(True)

quarter_labels = ['Spring', 'Summer', 'Autumn', 'Winter']
plt.legend([plt.Line2D([0], [0], marker='o', color='w', markerfacecolor=cpal[i], markersize=
           quarter_labels,
           title='Quarter')

plt.show()
```

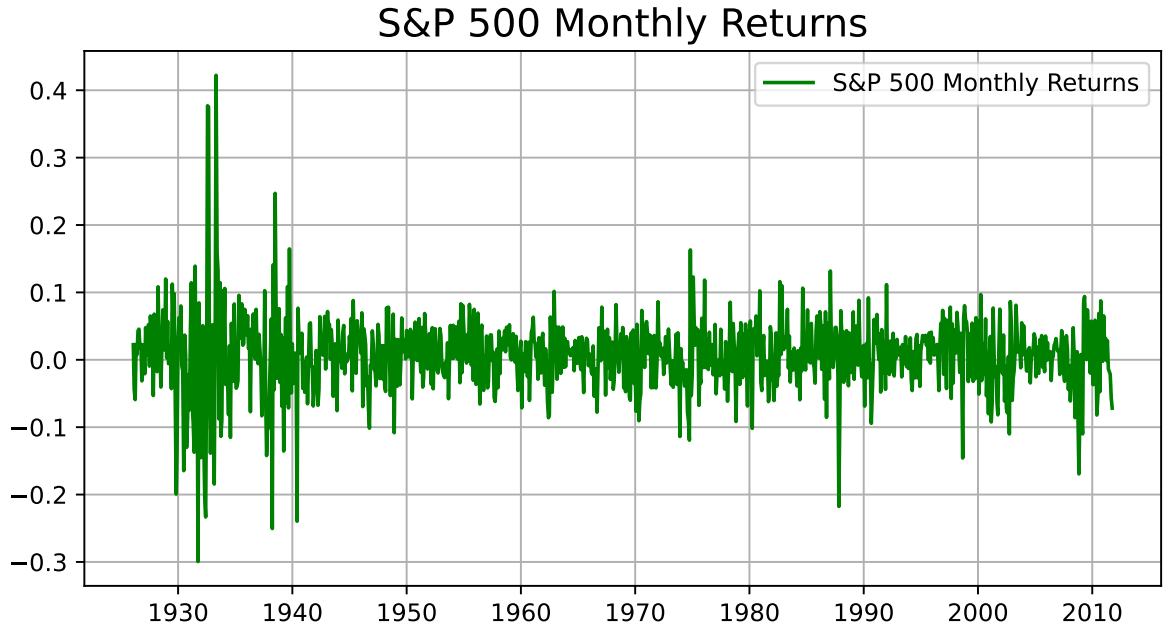


2.1.2 500

0

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

```
raw_data = []
with open("../ftsdata/m-ibmsp-2611.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])

data["date"] = pd.to_datetime(data["date"], format="%Y%m%d")
data["ibm"] = pd.to_numeric(data["ibm"])
data["sp"] = pd.to_numeric(data["sp"])
data.head()

plt.figure(figsize=(8, 4))
plt.plot(data["date"], data['sp'], label='S&P 500 Monthly Returns', color='green')
plt.title('S&P 500 Monthly Returns', fontsize=16)
plt.grid(True)
plt.legend()
plt.show()
```



2.2

500 0

$\{x_t, t = ..., -2, -1, 0, 1, 2, ...\}$, x_t X_t $\{X_t\}$ $X(t, \omega), t \in \mathbb{Z}(\mathbb{Z}$), $\omega \in \Omega$, Ω $\omega_0 \in \Omega$ “ ” $\omega \in \Omega$ $EX_t = \int X_t(\omega)P(d\omega)$ $X_t(\omega)$ $\omega \in \Omega$

$\omega \in \Omega$ “ ”

$\{x_t, t = 1, 2, ..., T\}$ x_t X_t

$\{X_t\}$ $Cov(X_s, X_t)$ $Cov(X_s, X_t) = \gamma_{|t-s|}$ $t - s$,

$$\gamma_k = \text{Cov}(X_{t-k}, X_t), k = 0, 1, 2, ...$$

$\{X_t\}$ $\text{Cov}(X_s, X_t) = \text{Cov}(X_t, X_s)$, $\gamma_{-k} = \gamma_k$ $\gamma_0 = \text{Var}(X_t)$

Cauchy-Schwartz

$$|\gamma_k| = |E[(X_{t-k} - \mu)(X_t - \mu)]| \leq (E(X_{t-k} - \mu)^2 \, E(X_t - \mu)^2)^{1/2} = \gamma_0$$

(weakly stationary time series): $\{X_t\}$

1. $EX_t = \mu$ t
2. $\text{Var}(X_t) = \gamma_0$ t
3. $\gamma_k = \text{Cov}(X_t - k, X_t)$, $k = 1, 2, ...$ t

$\{X_t\}$

Ω,

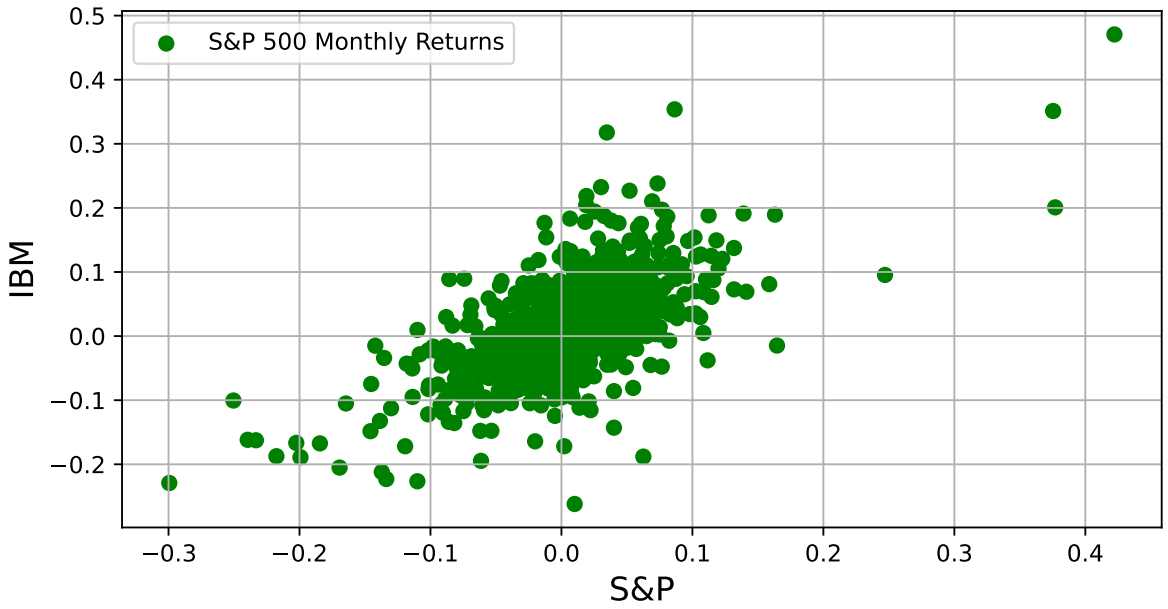
$$\hat{\gamma}_k = \frac{1}{T} \sum_{t=k+1}^{T} (x_{t-k} - \bar{x})(x_t - \bar{x}), k = 0, 1, ..., T - 1$$

$\hat{\gamma}_k$ $1/T$ $1/(T - k)$, $1/(T - k)$

2.3

2.3.1

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

raw_data = []
with open("../ftsdata/m-ibmsp-2611.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])

data["date"] = pd.to_datetime(data["date"], format="%Y%m%d")
data["ibm"] = pd.to_numeric(data["ibm"])
data["sp"] = pd.to_numeric(data["sp"])
data.head()

plt.figure(figsize=(8, 4))
plt.scatter(data["sp"], data['ibm'], label='S&P 500 Monthly Returns', color='green')
plt.xlabel("S&P", fontsize=14)
plt.ylabel("IBM", fontsize=14)
plt.grid(True)
plt.legend()
plt.show()
```



IBM 500

X Y

$$\rho(X,Y) = \rho_{xy} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}} = \frac{E[(X-\mu_x)(Y-\mu_y)]}{\sqrt{E(X-\mu_x)^2 E(Y-\mu_y)^2}}$$

(X,Y) (x_t, y_t), $t = 1, 2, \ldots, T,$ Pearson

$$\hat{\rho}_{xy} = \frac{\sum_{t=1}^{T}(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^{T}(x_t - \bar{x})^2 \sum_{t=1}^{T}(y_t - \bar{y})^2}}$$

X Y (Spearman) X ( ) Y Spearmam rank correlation

tau(Kendall's τ) $(X,Y), (X_1,Y_1), (X_2,Y_2)$ (X,Y) X Y tau

$$\tau = P\left[(X_1 - X_2)(Y_1 - Y_2) > 0\right] - P\left[(X_1 - X_2)(Y_1 - Y_2) < 0\right]$$

IBM

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

raw_data = []
with open("../ftsdata/m-ibmsp-2611.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])

data["date"] = pd.to_datetime(data["date"], format="%Y%m%d")
data["ibm"] = pd.to_numeric(data["ibm"])
data["sp"] = pd.to_numeric(data["sp"])
data.head()

pearson_corr = data['ibm'].corr(data['sp'])
spearman_corr = data['ibm'].corr(data['sp'], method='spearman')
kendall_corr = data['ibm'].corr(data['sp'], method='kendall')

print("Pearson correlation:", pearson_corr)
print("Spearman correlation:", spearman_corr)
print("Kendall correlation:", kendall_corr)
```

```
Pearson correlation: 0.6395978546773113
Spearman correlation: 0.6065788974589758
Kendall correlation: 0.4328065703413303
```

2.3.2

$\{X_t\}$ $\{\gamma_k\}$

$$\rho(X_{t-k}, X_t) = \frac{\mathrm{Cov}(X_{t-k}, X_t)}{\sqrt{\mathrm{Var}(X_{t-k})\,\mathrm{Var}(X_t)}} = \frac{\gamma_k}{\sqrt{\gamma_0\gamma_0}} = \frac{\gamma_k}{\gamma_0},\ k = 0, 1, \dots,\ \forall t$$

$\rho_k = \gamma_k/\gamma_0$, $X_t - k\,X_t$ t $\{\rho_k, k = 0, 1, ...\}$ $\{X_t\}$ (Autocorrelation function, ACF) $\rho_0 = 1$

$\{X_t\}$ $\rho_k = 0, k = 1, 2, ...,$ $\{X_t\}$ $\{X_t\}$ $\{X_t\}$

ρ_k

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0},\ k = 0, 1, ...$$

$\hat{\rho}_0 = 1$ $\hat{\rho}_k, k = 1, 2, ...$

$\hat{\rho}_k$ ρ_k

$\{X_t\}$ $\hat{\rho}_k(k > 0)$ $\mathrm{N}(0, \frac{1}{T})$

$\{\varepsilon_t\}$ q $\{\psi_j, j = 0, 1, ..., q\}$ $\psi_0 = 1$,

$$X_t = \mu + \sum_{j=0}^{q} \psi_j \varepsilon_{t-j},\ t \in \mathbb{Z},$$

$\{X_t, t = 1, ..., T\}$ ACF $k > q$ $\sqrt{T}\hat{\rho}_k$ $\mathbb{N}(0, 1 + 2\sum_{j=0}^{q} \rho_j^2)$, Bartlett

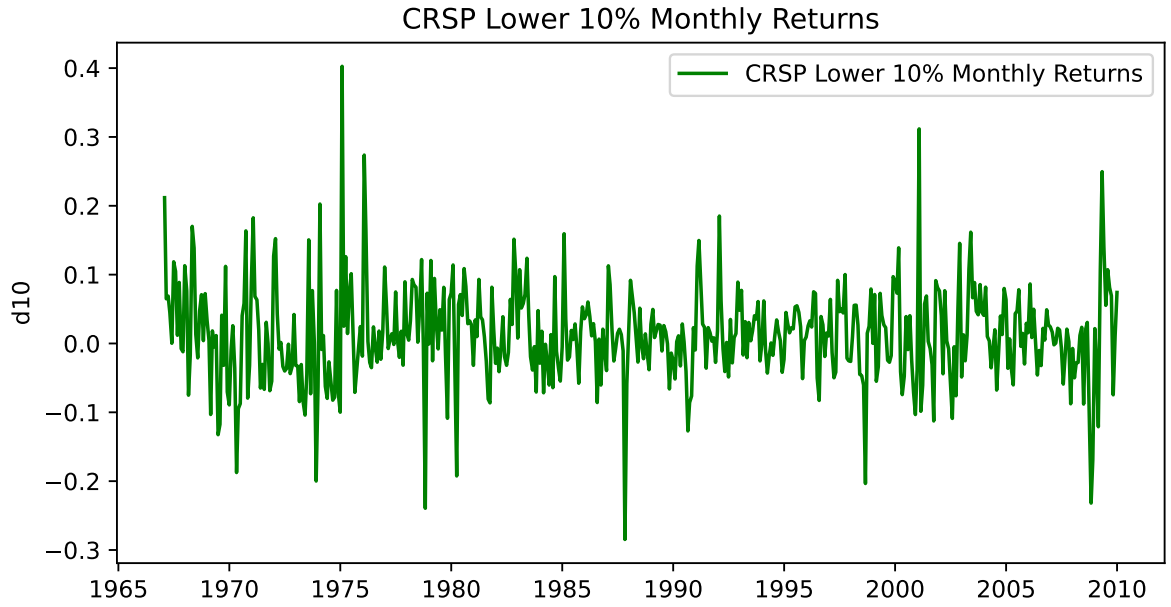
2.3.2.1 CRSP 10

10 NYSE AMEX NASDAQ 10%

- CRSP Center for Research in Security Prices, Chicago Booth
- NYSE(The New York Stock Exchange,),
- AMEX(American Stock Exchange,)
- NASDAQ(National Association of Securities Dealers Automated Quotations)

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

raw_data = []
with open("../ftsdata/m-dec12910.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])

data["date"] = pd.to_datetime(data["date"], format="%Y%m%d")
data.set_index("date", inplace=True)
data = data.apply(pd.to_numeric)
data.head()
```

```
plt.figure(figsize=(8, 4))
plt.plot(data["dec10"], label='CRSP Lower 10% Monthly Returns', color="green")
plt.title('CRSP Lower 10% Monthly Returns')
plt.ylabel('d10')
plt.legend()
plt.show()
```

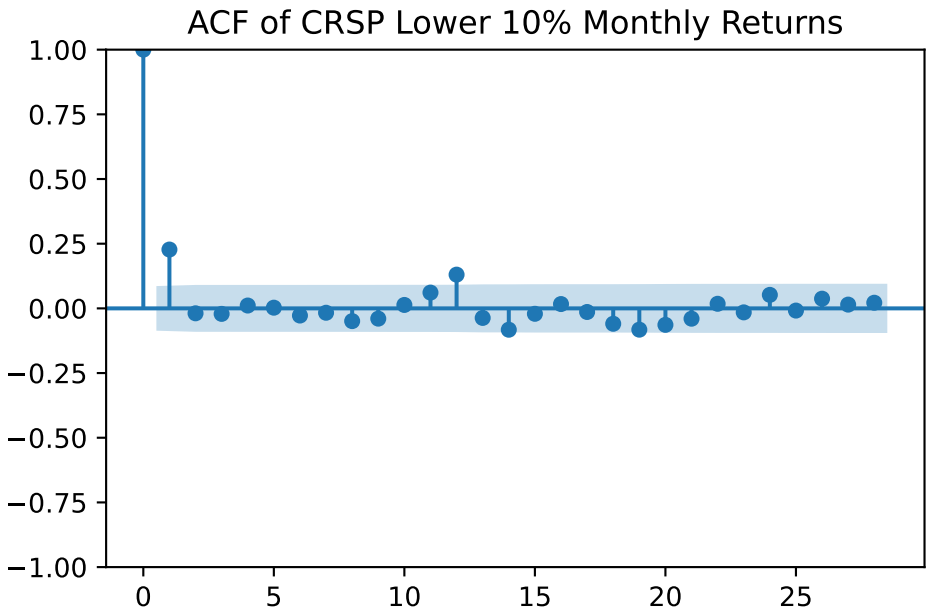


ACF

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

raw_data = []
with open("../ftsdata/m-dec12910.txt", "r", encoding="utf-8") as file:
    for line in file.readlines():
        line = line.strip("\n").strip(" ").replace("\t", " ").split(" ")
        line = list(filter(lambda x: x != "", line))
        raw_data.append(line)
data = pd.DataFrame(raw_data[1:], columns=raw_data[0])

data["date"] = pd.to_datetime(data["date"], format="%Y%m%d")
data.set_index("date", inplace=True)
data = data.apply(pd.to_numeric)
```

```
data.head()

from statsmodels.graphics.tsaplots import plot_acf

plt.figure(figsize=(8, 4))
plot_acf(data["dec10"])
plt.title('ACF of CRSP Lower 10% Monthly Returns')
plt.show()
```

```
<Figure size 2400x1200 with 0 Axes>
```



2.4

2.5