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

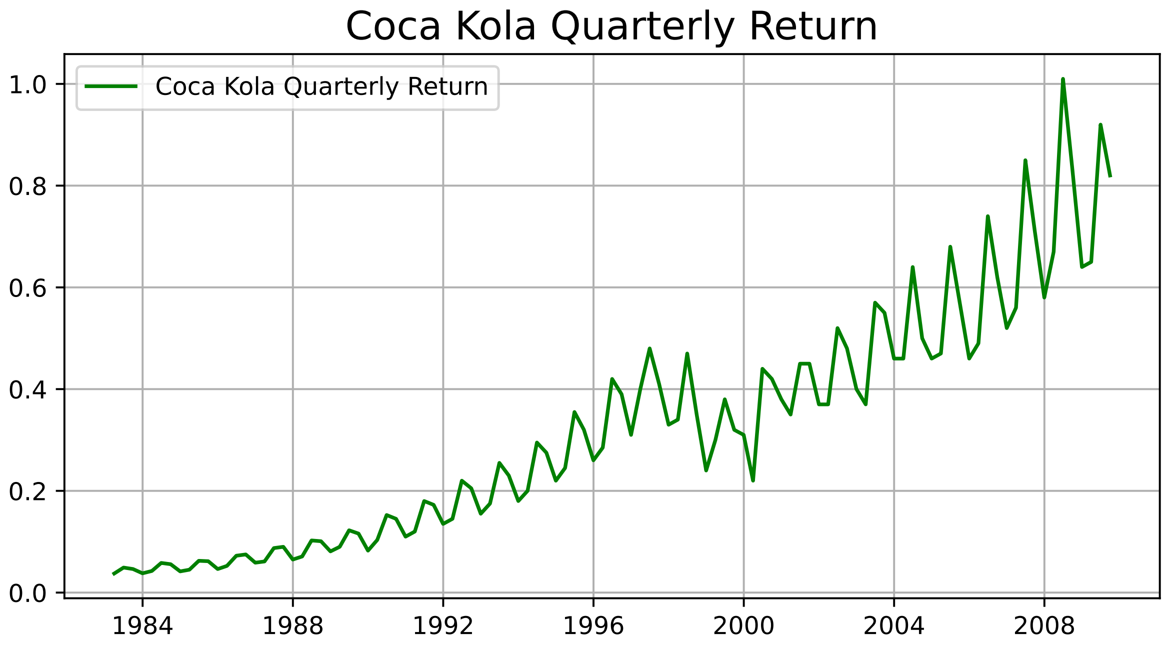
Ruey S. Tsay R (Tsay 2013) An Introduction to Analysis of Financial Data with R “ ”

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

2.1.1

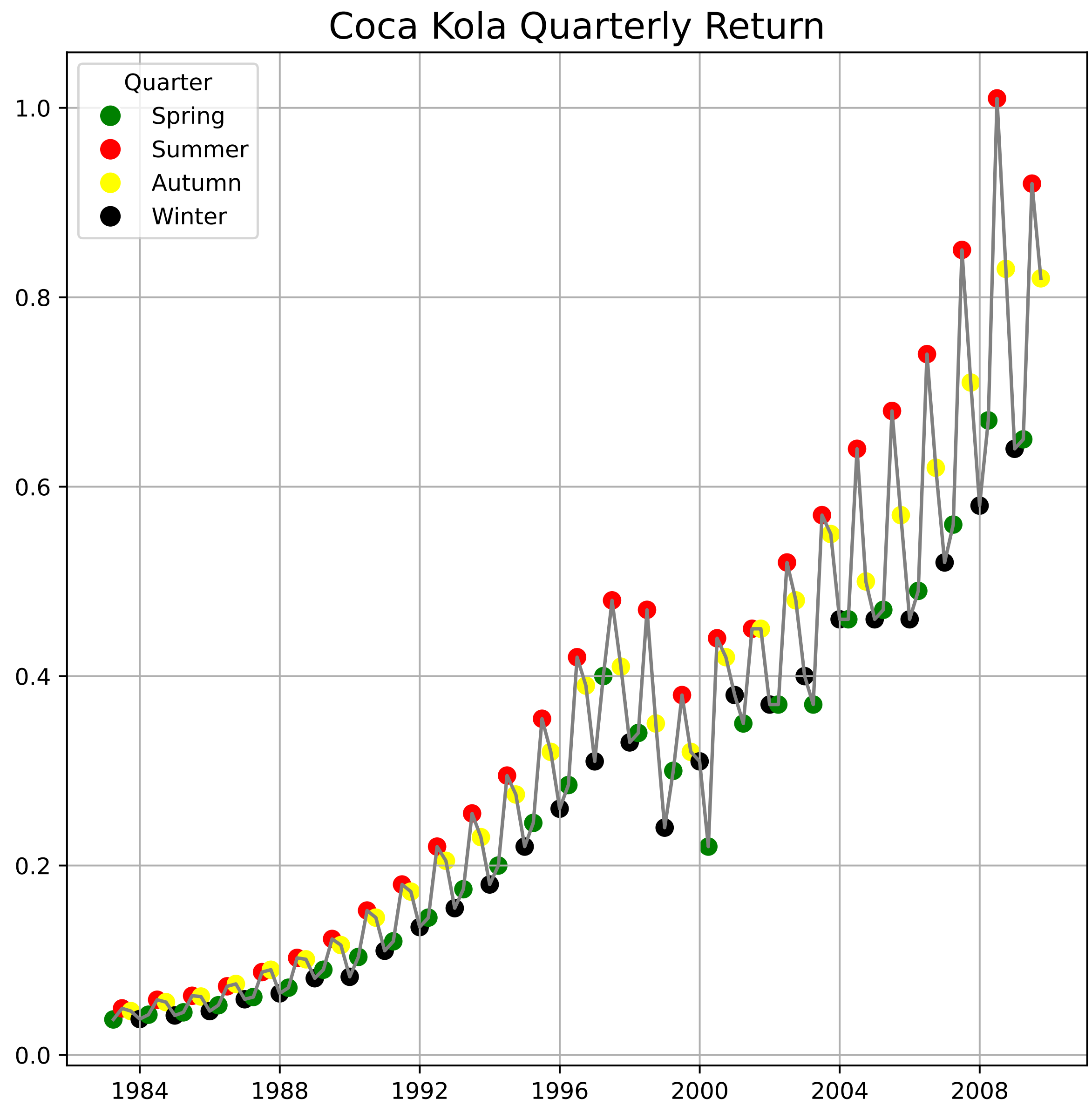
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

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()
```

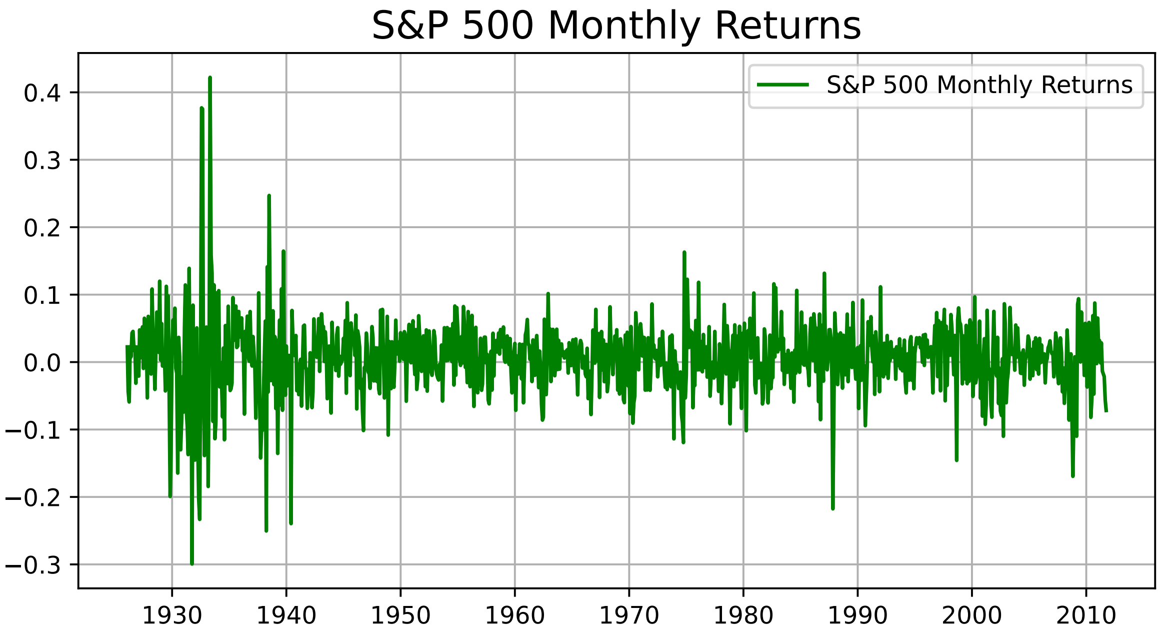


2.1.2 500

0

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

```python
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

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

$\omega \in \Omega$ “ ”

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

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

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

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

Cauchy-Schwartz

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

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

1. $EX_t = \mu$ t
2. Var$(X_t) = \gamma_0$ t
3. γ_k =Cov$(X_t - k, X_t)$, $k = 1, 2, \ldots$ t

$\{X_t\}$

Ω,

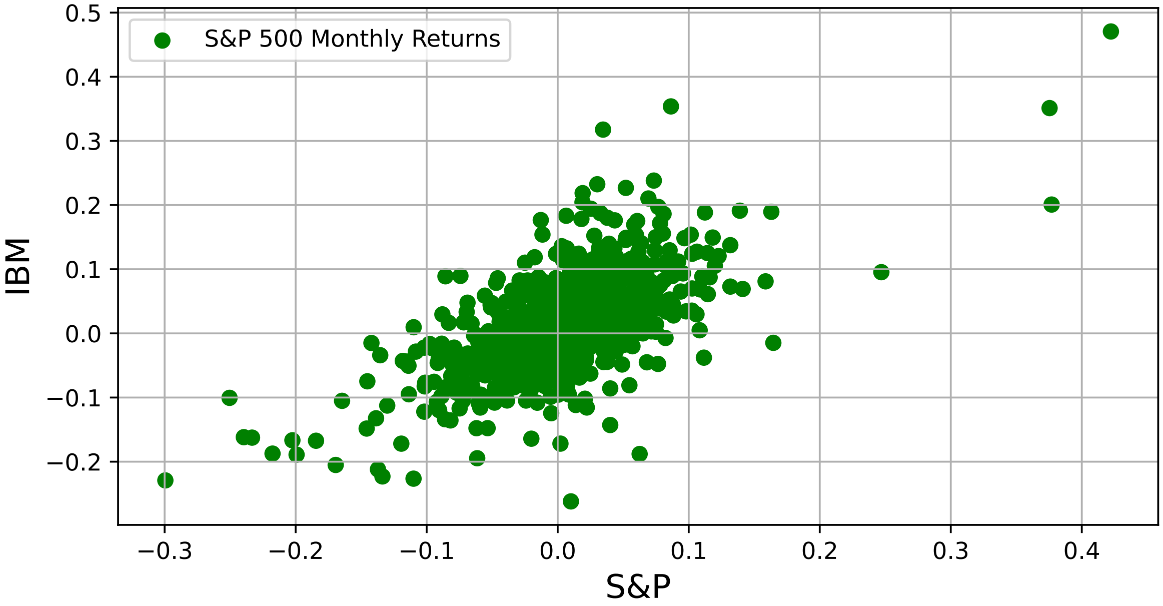
$$\hat{\gamma}_k = \frac{1}{T} \sum_{t=k+1}^{T} (x_{t-k} - \bar{x})(x_t - \bar{x}), k = 0, 1, \ldots, T - 1$$

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

2.3

2.3.1

```python
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

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

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

```python
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

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

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

$\rho_k = \gamma_k/\gamma_0$, $X_t - k$ 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

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

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

$\hat{\rho}_k$ ρ_k

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

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

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

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

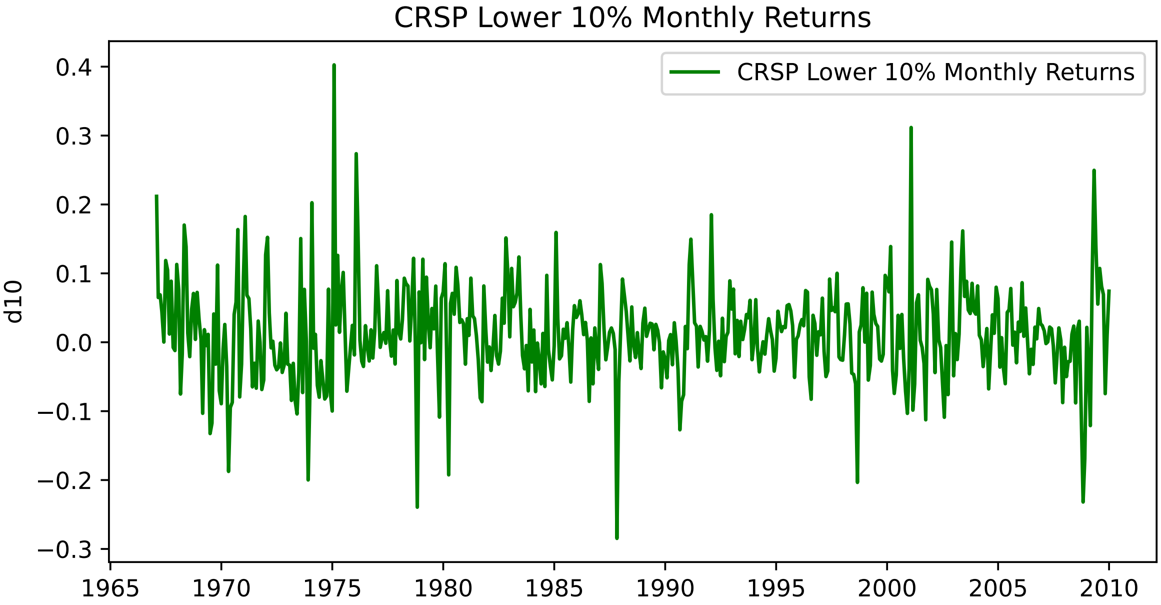
2.3.2.1 CRSP 10

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()
```

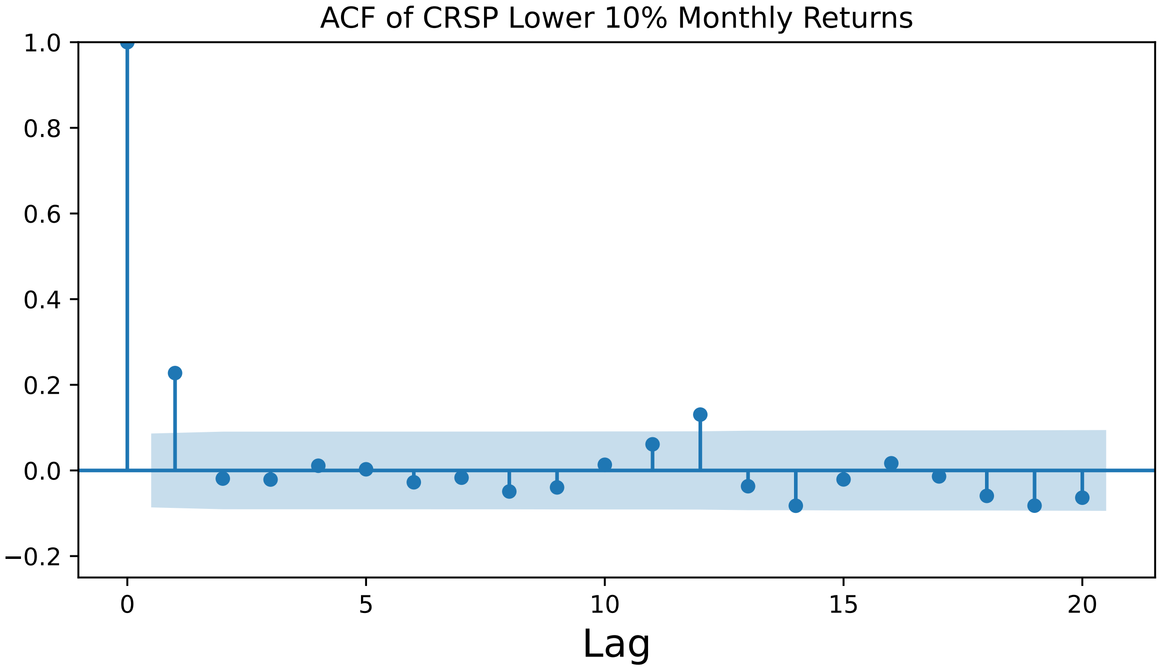
```
plt.figure(figsize=(8, 4))
plt.plot(data["dec10"], label='CRSP Lower 10% Monthly Returns', color="green")
plt.title('CRSP Lower 10% Monthly Returns')
plt.ylabel('d10')
plt.legend()
plt.show()
```



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)
```

```python
data.head()

from statsmodels.graphics.tsaplots import plot_acf

plt.figure(figsize=(8, 4))
ax = plt.gca()  #
plot_acf(data["dec10"], ax=ax, lags=20)
ax.set_ylim(-0.25, 1)
ax.set_xlabel('Lag', fontsize=16)  #
plt.title('ACF of CRSP Lower 10% Monthly Returns')
plt.show()
```



ACF Lag k ACF $\hat{\rho}_k$

ACF $\pm\frac{2}{\sqrt{T}}$ $\hat{\rho}_k$ 95%

ACF $k = 0$ $\hat{\rho}_0 = 1$

$\hat{\rho}_1$ $\hat{\rho}_{12}$ (1/12)

2.3.3

$\{X_t\}$ $\hat{\rho}_k(k \geq 1)$ $N(0, 1/T)$ H_0

$$t = \sqrt{T}\hat{\rho}_k$$

$|t| >$qnorm$(1-\alpha/2)$ $\alpha = 0.05$, qnorm$(1-\alpha/2) \approx 2$, $\hat{\rho}_1$ $\pm 2/\sqrt{T}$ H_0, $\hat{\rho}_k$ $\pm 2/\sqrt{T}$ H_0, t (± 3)
$\{X_t\}$ $X_t = \mu + \sum_{j=0}^{q} \psi_j \varepsilon_{t-j}$ Bartlet

$$t = \frac{\hat{\rho}_k}{\sqrt{\frac{1}{T}\left(1 + 2\sum_{j=1}^{k-1} \hat{\rho}_j^2\right)}},\ k > q$$

t qnorm$(1-\alpha/2)$

2.3.3.1 TODO 3.3

2.3.4 Ljung-Box

$\rho_k = 0, k = 1, 2, \cdots$ ρ_k
Box Pierce(G. Box & Pierce, 1970) (Portmanteau statistic)

$$Q_*(m) = T\sum_{j=1}^{m} \hat{\rho}_j^2$$

$$H_0 : \rho_1 = \cdots = \rho_m = 0 \longleftrightarrow H_a :$$

$\{X_t\}$ $Q_*(m)$ $\chi^2(m)$ α, $Q_*(m) >$qchisq$(1-\alpha, m)$ H_0, m
Ljung Box(Ljung & Box, 1978)

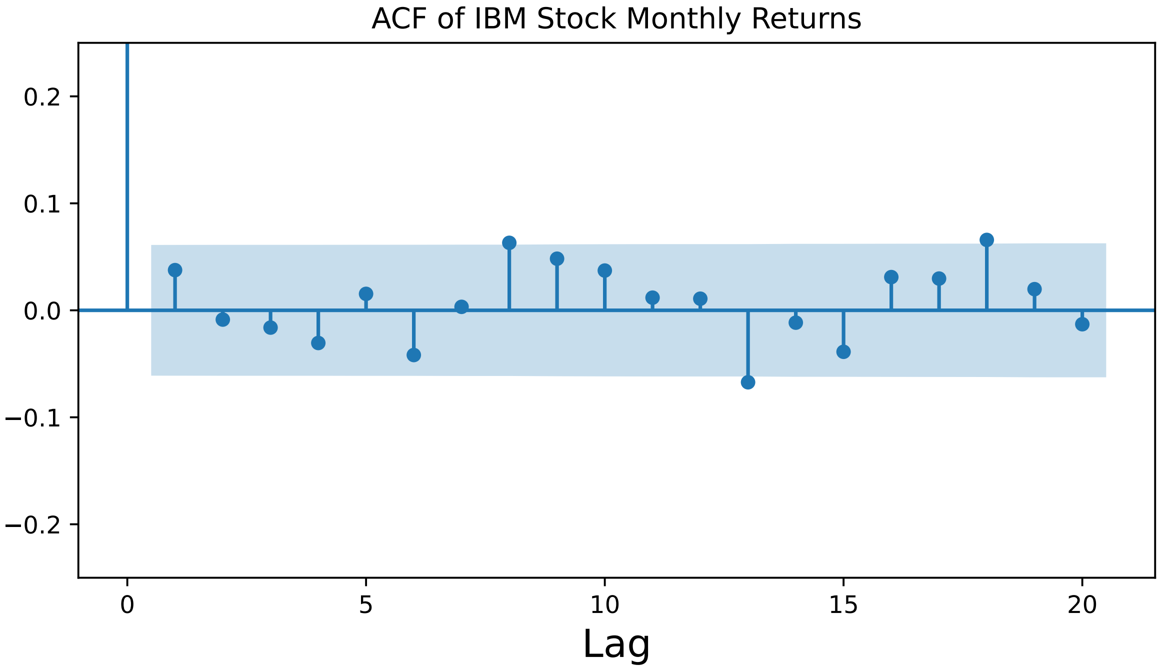
$$Q(m) = T(T+2)\sum_{j=1}^{m} \frac{\hat{\rho}_j^2}{T-j}$$

$\chi^2(m)$ $Q(m) >$qchisq$(1-\alpha, m)$ H_0, Ljung-Box m
ARMA(p, q) $m - (p + q)$

2.3.4.1 IBM

ACF

```
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.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))
ax = plt.gca()  #
plot_acf(data["ibm"], ax=ax, lags=20)
ax.set_ylim(-0.25, 0.25)
ax.set_xlabel('Lag', fontsize=16)  #
plt.title('ACF of IBM Stock Monthly Returns')
plt.show()
```



ACF

Ljung-Box $m = 12$ $m = 24$

```
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.set_index("date", inplace=True)
data = data.apply(pd.to_numeric)
data.head()

from statsmodels.stats.diagnostic import acorr_ljungbox
```

```
#   Ljung-Box
lb_test_12 = acorr_ljungbox(data["ibm"], lags=[12], return_df=True)
print(lb_test_12)

lb_test_24 = acorr_ljungbox(data["ibm"], lags=[24], return_df=True)
print(lb_test_24)
```

```
      lb_stat  lb_pvalue
12  13.097984   0.361959
      lb_stat  lb_pvalue
24  35.384127   0.062905
```

0.05 IBM

LB

```
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.set_index("date", inplace=True)
data = data.apply(pd.to_numeric)
data.head()

#   Ljung-Box
lb_test_12 = acorr_ljungbox(np.log(data["ibm"] + 1), lags=[12], return_df=True)
print(lb_test_12)

lb_test_24 = acorr_ljungbox(np.log(data["ibm"] + 1), lags=[24], return_df=True)
print(lb_test_24)
```

```
      lb_stat  lb_pvalue
```

```
12  12.814366   0.382677
      lb_stat  lb_pvalue
24  34.505798   0.076073
```

0.05

2.3.4.2 CRSP 10

[CRSP 10]

```
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.stats.diagnostic import acorr_ljungbox


#   Ljung-Box
lb_test_12 = acorr_ljungbox(data["dec10"], lags=[12], return_df=True)
print(lb_test_12)

lb_test_24 = acorr_ljungbox(data["dec10"], lags=[24], return_df=True)
print(lb_test_24)
```

```
      lb_stat  lb_pvalue
12  41.059699   0.000048
      lb_stat  lb_pvalue
24  56.245617   0.000212
```

0.05 CRSP 10

TREVOR S. BREUSCH (1978) LESLIE G. GODFREY (1978) (LM) AR MA ARMA

2.4

$\{X_t\}$ $\{X_t\}$ (white noise) $\{X_t\}$ N$(0, \sigma^2)$ $\{X_t\}$ (Gaussian)

$(\rho_0 = 1$)

(ACF),

$\{\varepsilon_t\}$ $\mathrm{Var}(\varepsilon_t) = \sigma^2$, $\{\psi_j\}$ $\sum_j \psi_j^2 < \infty, \psi_0 = 1$,

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

$\{X_t\}$ ( ) ε_t t (innovation) (shock) $\varepsilon_{t+j}(j \geq 1)$ X_t, X_{t-1} $\{X_t - 1, X_{t-2}, ...\}$ $\{\varepsilon_t - 1, \varepsilon_{t-2}, ...\}$

$\{\varepsilon_t\}$ ( ) (3.1) $\{X_t\}$ **Wold** $\{X_t\}$

AR MA ARMA

$\{\psi_j\}$ ψ

$$EX_t = \mu, \quad \mathrm{Var}(X_t) = \sigma^2 \sum_{j=0}^{\infty} \psi_j^2,$$

$$\begin{aligned}
\gamma_k =& \mathrm{Cov}(X_t, X_{t-k}) = E\left[\left(\sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}\right)\left(\sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i-l}\right)\right] \\
=& E\left(\sum_{i,j=0}^{\infty} \psi_i \psi_j \varepsilon_{t-i} \varepsilon_{t-j-k}\right) = \sigma^2 \sum_{i,j=0}^{\infty} \psi_i \psi_j \delta_{i-j-k} \\
=& \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+k}
\end{aligned}$$

δ_k $k = 0$ 1, $k \neq 0$ 0

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\sum_{j=0}^{\infty} \psi_j \psi_{j+k}}{1 + \sum_{j=1}^{\infty} \psi_j^2},\ k \geq 0$$

$\psi_j \to 0, j \to \infty,$ $\rho_k \to 0, k \to \infty,$

2.5

3

3.1

$\rho_1 \neq 0$, X_t X_{t-1} X_{t-1} X_t

$$X_t = \phi_0 + \phi_1 X_{t-1} + \varepsilon_t$$

(Autoregression model), AR(1) $\{\varepsilon_t\}$ σ^2, ε_t $X_{t-1}, X_{t-2}, \ldots$ $|\phi_1| < 1$ $\{\varepsilon_t\}$

$Y_i = \phi_0 + \phi_1 x_i + \varepsilon_i$ ε_t X_{t-1} $t-1$

AR(1) (Markov) X_t $X_{t-1}, X_{t-2}, \ldots$ X_{t-1} X_{t-1} $X_{t-1}, X_{t-2}, \ldots$ X_t, X_{t-1}

$$E(X_t|X_{t-1}) = \phi_0 + \phi_1 X_{t-1}, \quad \mathrm{Var}(X_t|X_{t-1}) = \sigma^2$$

$X_t - 1 = x_{t-1}$ X_t $\phi_0 + \phi_1 x_{t-1}$, σ^2

$$\mathrm{Var}(X_t) = \frac{\sigma^2}{1 - \phi_1^2}$$

$|\phi_1| < 1$, X_t $X_t - 1 = x_{t-1}$ $X_t - 1$ X_t, X_t

AR(1) AR(p)

$$X_t = \phi_0 + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t$$

$\{\varepsilon_t\}$ σ^2, ε_t $X_t - 1, X_{t-2}, \ldots$ $\phi_1, \ldots, \phi_p$

$$1 - \phi_1 z - \cdots - \phi_p z^p = 0$$

z_* $|z_*| > 1$, AR(p) “ ” (R. S. Tsay, 2013) AR(1) $|\phi_1| < 1$.

$\{\varepsilon_t\}$

3.2

$\{\xi_t, t \in \mathbb{Z}\}$ B

$$B\xi_t = \xi_{t-1}, \quad B^j\xi_t = \xi_{t-j}, \; j \in \mathbb{Z}$$

$P(z) = \sum_{j=0}^{\infty} a_j z^j,\ \{a_j\}$ $\sum_{j=0}^{\infty} |a_j| < \infty$ ( ), $P(z)$

$$P(B)\xi_t = \sum_{j=0}^{\infty} a_j \xi_{j-t}.$$

$P(B)$ ( ) $P(B)\xi_t$ $\{\xi_t\}$

$Q(z) = \sum_{j=0}^{\infty} b_j z^j, \{b_j\}$ $C(z) = P(z)Q(z) = \sum_{j=0}^{\infty} c_j z^j,$

$$c_j = \sum_{i=0}^{j} a_i b_{j-i}, \; j = 0, 1, ...$$

$\{c_j\}$ $\{c_j\}$ $\{a_j\}$ $\{b_j\}$ $\{\xi_t\}$

$$P(B)Q(B)\xi_t = Q(B)P(B)\xi_t = C(B)\xi_t, \; t \in \mathbb{Z}$$

$\xi_t \equiv \xi$ t $B^j\xi = \xi$ $B^j 1 = 1, P(B)1 = P(1)$

$D(z) = \frac{P(z)}{Q(z)}, Q(z) \neq 0$ $|z| \leq 1$ z $D(z) = \sum_{j=0}^{\infty} d_j z^j, \{d_j\}$

$$P(B)\xi_t = Q(B)D(B)\xi_t = D(B)Q(B)\xi_t, \; t \in \mathbb{Z}$$

3.3 TODO: AR(1)

3.4 TODO: AR(1)

3.5 TODO: AR(2)

3.5.0.1 TODO: (GNP)

3.6 AR(p)

AR(p) ACF ACF -1

$$\mu = \frac{\phi_0}{1 - \phi_1 - \cdots - \phi_p}$$

(ACF) ( )

$$(1 - \phi_1 B - \cdots - \phi_p B^p)\rho_j = 0, \quad j = 1, 2, ...$$

AR(p)

3.7

AR p p AIC

$X_1, ..., X_n, Y$

$$L(Y|X_1, \dots, X_n) = \underset{\hat{Y}=b_0+b_1X_1+\cdots+b_nX_n}{\operatorname{argmin}} E(Y - \hat{Y})^2$$

$X_1, ..., X_n$ Y $Y - L(Y|X_1, ..., X_n)$ $Z - L(Z|X_1, ..., X_n)$ Y Z $X_1, ..., X_n$ $n = 1, 2, ...,$

$$L(X_t|X_{t-1}, ..., X_{t-n}) = \phi_{n0} + \phi_{n1}X_{t-1} + ... + \phi_{nn}X_{t-n}$$

$\phi_{nj}, j = 0, 1, ..., n$ t ϕ_{nn} $\{X_t\}$ $\{\phi_{nn}\}$ $\{X_t\}$ **(PACF)**

ϕ_{nn} X_t $X_t - n$ $X_t - 2, ..., X_t - n + 1$ $\phi_1 1$ ρ_1

ϕ_{nn} $\hat{\phi}_{nn}, n = 1, 2, ...$

$\{X_t\}$ AR(p)

$$X_t = \phi_0 + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t,\ \phi_p \neq 0$$

$X_{t-1}, X_{t-2}, ...$ X_t $X_t - 1, ..., X_{t-p}$, $X_{t-p-1}, X_{t-p-2}, ...$ $\phi_{kk} = 0, k > p$ AR

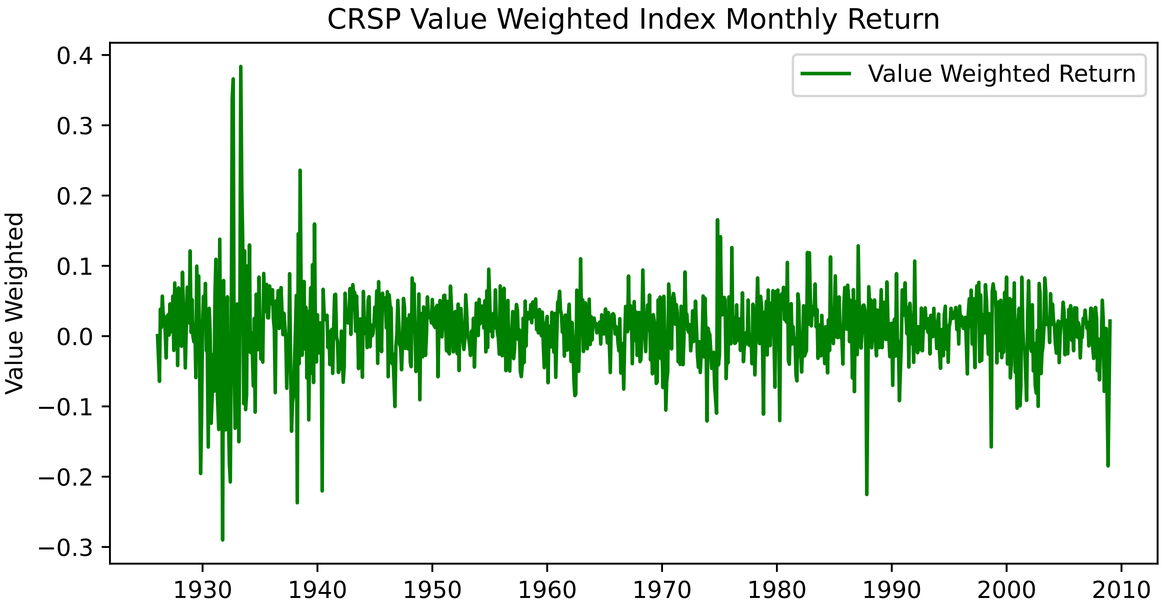
AR(p) $\hat{\phi}_{kk}$

- $T \to \infty$ $\hat{\phi}_p p \to \phi_p \neq 0$
- $k > p, \hat{\phi}_{kk} \to 0(\hat{T} \to \infty)$
- $k > p, \hat{\phi}_{kk}$ $\frac{1}{T}$

ACF PACF $\pm\frac{2}{\sqrt{T}}$ PACF

3.7.0.1 CRSP

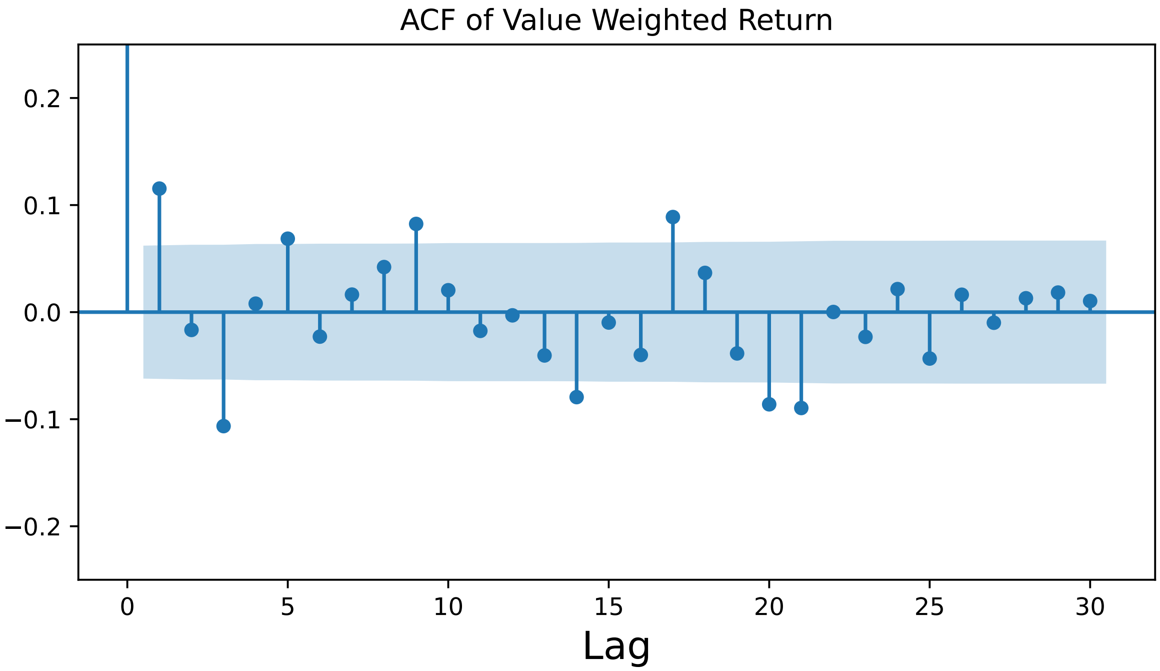
```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

raw_data = []
with open("../ftsdata/m-ibm3dx2608.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["vwrtn"], label='Value Weighted Return', color="green")
plt.title('CRSP Value Weighted Index Monthly Return')
plt.ylabel('Value Weighted')
plt.legend()
plt.show()
```



ACF

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

raw_data = []
with open("../ftsdata/m-ibm3dx2608.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
```

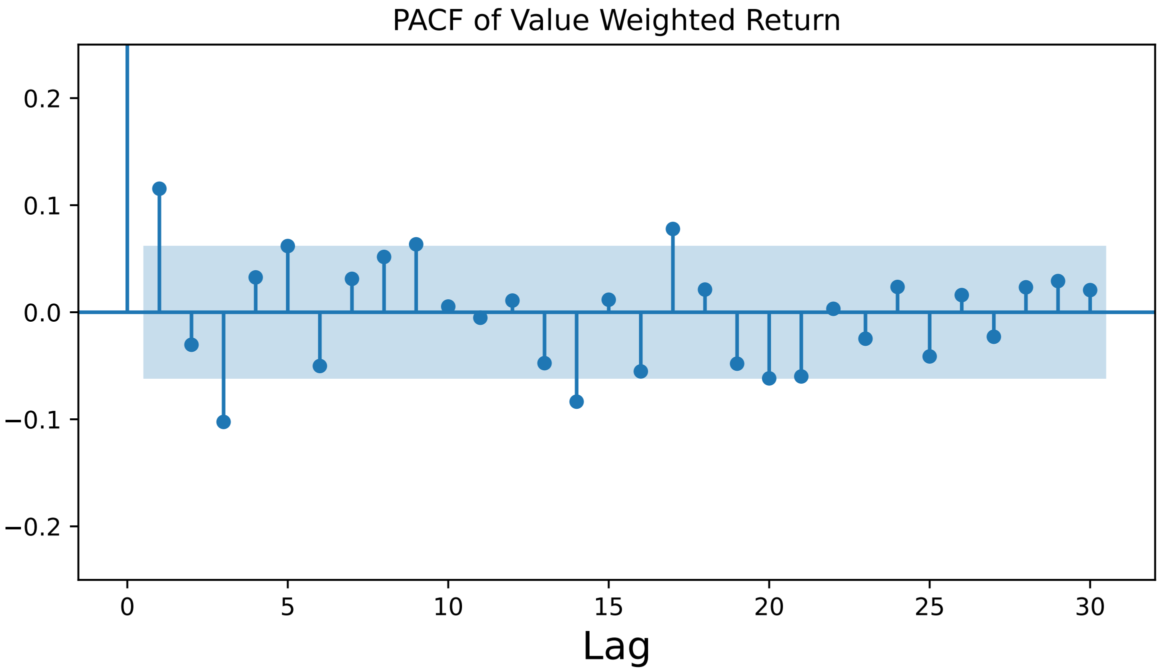
```python
plt.figure(figsize=(8, 4))
ax = plt.gca()  #
plot_acf(data["vwrtn"], ax=ax, lags=30)
ax.set_ylim(-0.25, 0.25)
ax.set_xlabel('Lag', fontsize=16)  #
plt.title('ACF of Value Weighted Return')
plt.show()
```



ACF $k = 21$ PACF

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

raw_data = []
with open("../ftsdata/m-ibm3dx2608.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_pacf

plt.figure(figsize=(8, 4))
ax = plt.gca()  #
plot_pacf(data["vwrtn"], ax=ax, lags=30)
ax.set_ylim(-0.25, 0.25)
ax.set_xlabel('Lag', fontsize=16)  #
plt.title('PACF of Value Weighted Return')
plt.show()
```



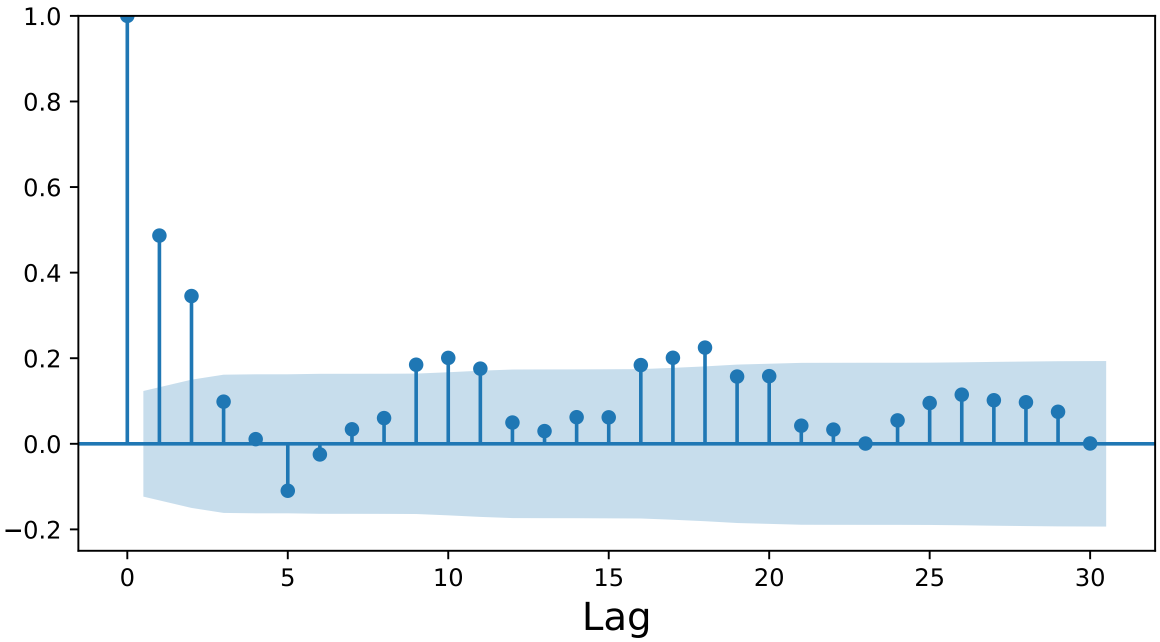
PACF $p = 3$ PACF $k = 17$

3.7.0.2 (GNP)

TODO: (GNP) GNP p

ACF

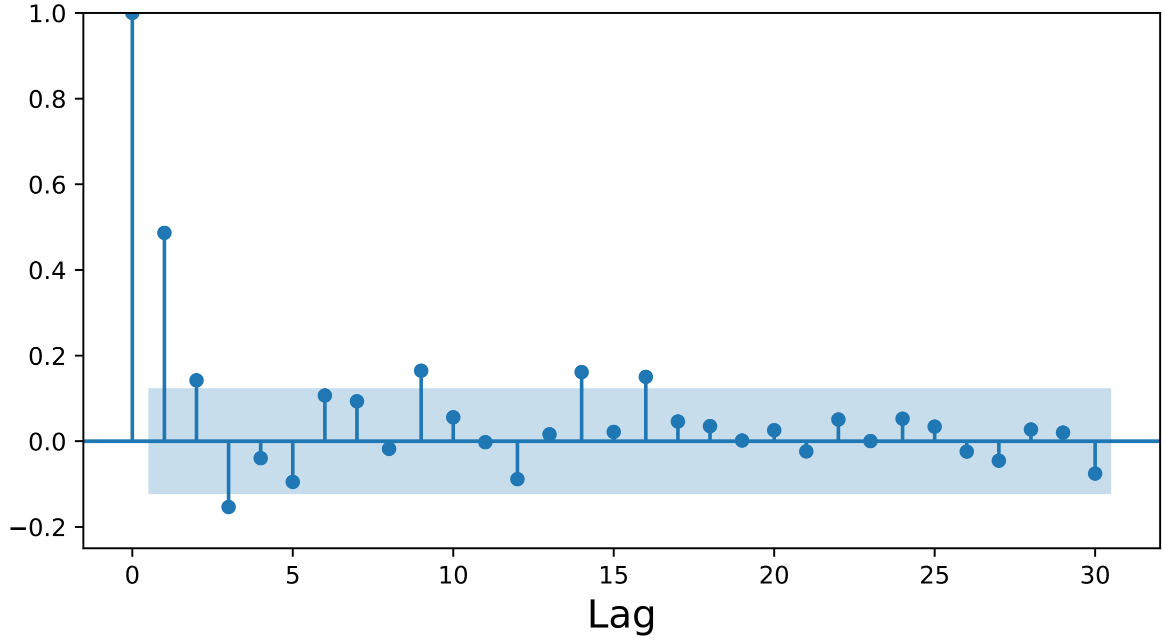
```
import pandas as pd

#
da = pd.read_csv("../ftsdata/q-gnp4710.txt", sep='\s+', dtype=float)
data = pd.DataFrame(da["VALUE"].values, index=pd.date_range(start="1947-01", periods=len(da)
data.head()

from statsmodels.graphics.tsaplots import plot_acf

plt.figure(figsize=(8, 4))
ax = plt.gca()  #
plot_acf(np.diff(np.log(data["value"])), ax=ax, lags=30)
ax.set_ylim(-0.25, 1)
ax.set_xlabel('Lag', fontsize=16)  #
plt.title('')
plt.show()
```



ACF PACF

```
import pandas as pd

#
da = pd.read_csv("../ftsdata/q-gnp4710.txt", sep='\s+', dtype=float)
```

```
data = pd.DataFrame(da["VALUE"].values, index=pd.date_range(start="1947-01", periods=len(da)
data.head()

from statsmodels.graphics.tsaplots import plot_pacf

plt.figure(figsize=(8, 4))
ax = plt.gca()  #
plot_pacf(np.diff(np.log(data["value"])), ax=ax, lags=30)
ax.set_ylim(-0.25, 1)
ax.set_xlabel('Lag', fontsize=16)  #
plt.title('')
plt.show()
```



PACF = 3, 9, 14, 16 AR(3)