Python ARIMA

Python

1 ARIMA

1970 GEP Box GM Jenkins Time Series Analysis: Forecasting and Control

ARIMA

1 (AR, autoregression)

$\{Y_t\}$ $Y_t = c + \phi_1 \cdot Y_{t-1} + \mu_t$ μ_t $\{Y_t\}$ $AR(1)$ p $AR(p)$

2 (MA, moving average)

Y_t $Y_t = c + \theta \cdot \mu_{t-1} + \mu_t$ μ_t $\{Y_t\}$ $MA(q)$ q $MA(q)$

3)ARMA ARIMA

Y_t $Y_t = c+\phi \cdot Y_{t-1}+\theta \cdot \mu_{t-1}+\mu_t$ μ_t $\{Y_t\}$ $ARMA(1,1)$ p q $ARMA(p,q)$ 1 $ARIMA(p,1,q)$

$ARIMA(p,1,q)$

2 Python

Python

pandas, numpy, scipy, statsmodels matplotlib

```
from __future__ import print_function
import pandas as pd
import numpy as np
from scipy import  stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.graphics.api import qqplot
```

90

```
#input the data
dta=[10930,10318,10595,10972,7706,6756,9092,10551,9722,10913,11151,8186,6422,6337,11649,1165
```

```
## change data type
dta=np.array(dta,dtype=np.float)
dta=pd.Series(dta)
dta.index = pd.Index(sm.tsa.datetools.dates_from_range('1911','2000'))
dta.plot(figsize=(12,8))
# general view of the data
dta.plot(figsize=(12,8))
plt.show()
```

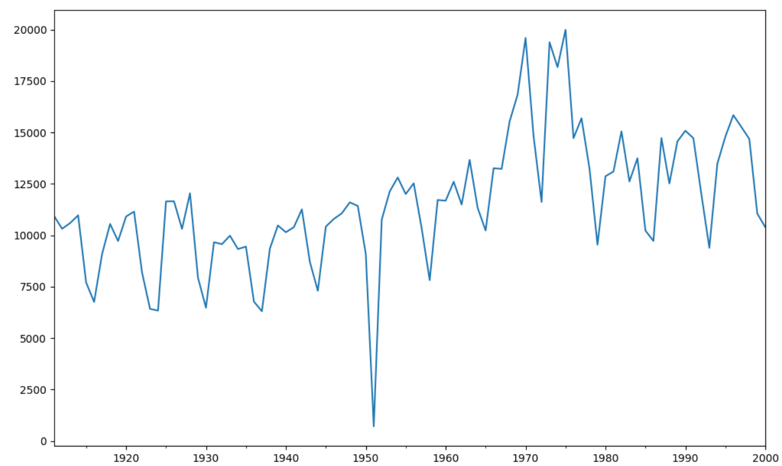

Figure 1:

```
## difference 1-order
fig = plt.figure(figsize=(12,8))
ax1= fig.add_subplot(111)
diff1 = dta.diff(1)
diff1.plot(ax=ax1)
plt.show()
```

```
## difference 2-order
fig = plt.figure(figsize=(12,8))
ax2= fig.add_subplot(111)
diff2 = dta.diff(2)
```

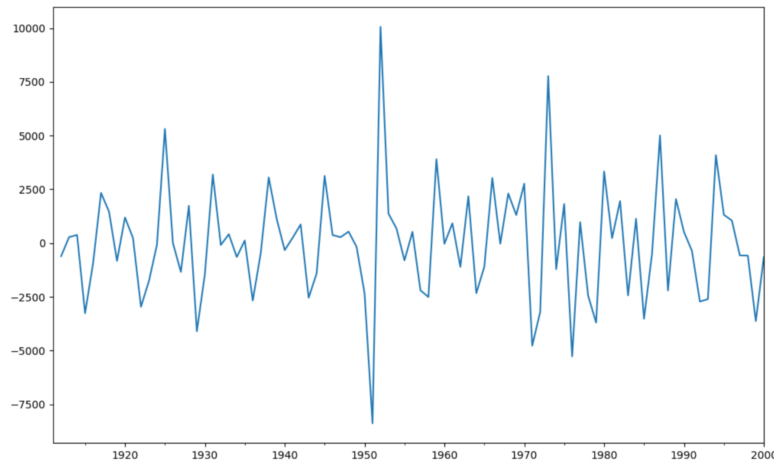

Figure 2:

```
diff2.plot(ax=ax2)
plt.show()
```

(acf, autocorrelation function) (pacf, partial autocorrelation funcition) acf pacf

```
## acf and pacf
diff1= dta.diff(1)
dta=dta.diff1
fig = plt.figure(figsize=(12,8))
ax1=fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(dta,lags=40,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(dta,lags=40,ax=ax2)
plt.show()
```

$lag = 8$ AIC BIC $AR(7)$ $AR(8)$

```
## model specification, AIC and BIC
arma_mod70 = sm.tsa.ARMA(dta,(7,0)).fit()
print(arma_mod70.aic,arma_mod70.bic,arma_mod70.hqic)
arma_mod30 = sm.tsa.ARMA(dta,(0,1)).fit()
print(arma_mod30.aic,arma_mod30.bic,arma_mod30.hqic)
arma_mod71 = sm.tsa.ARMA(dta,(7,1)).fit()
print(arma_mod71.aic,arma_mod71.bic,arma_mod71.hqic)
```

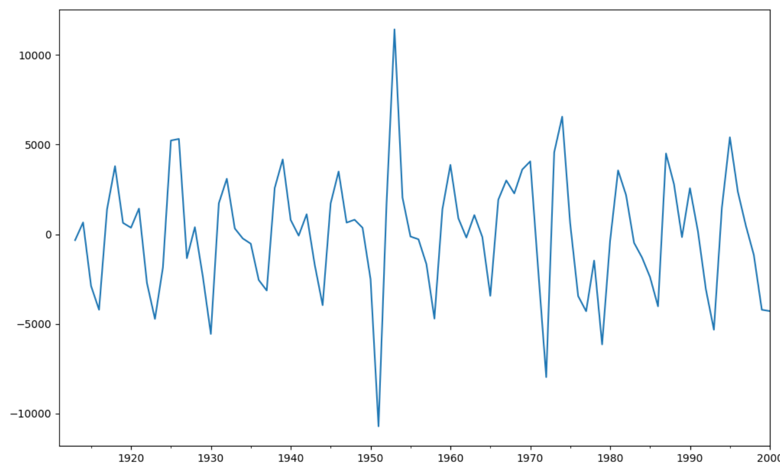

Figure 3:

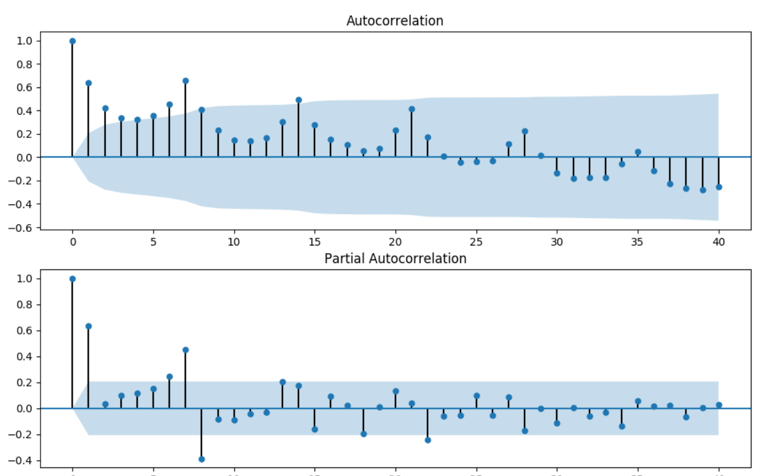

Figure 4:

```
arma_mod80 = sm.tsa.ARMA(dta,(8,0)).fit()
print(arma_mod80.aic,arma_mod80.bic,arma_mod80.hqic)
```

AR 8 MA 0 $ARIMA(8,1,0)$

- ACF PACF

```
## residual
resid = arma_mod80.resid
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(resid.values.squeeze(), lags=40, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(resid, lags=40, ax=ax2)
plt.show()
```

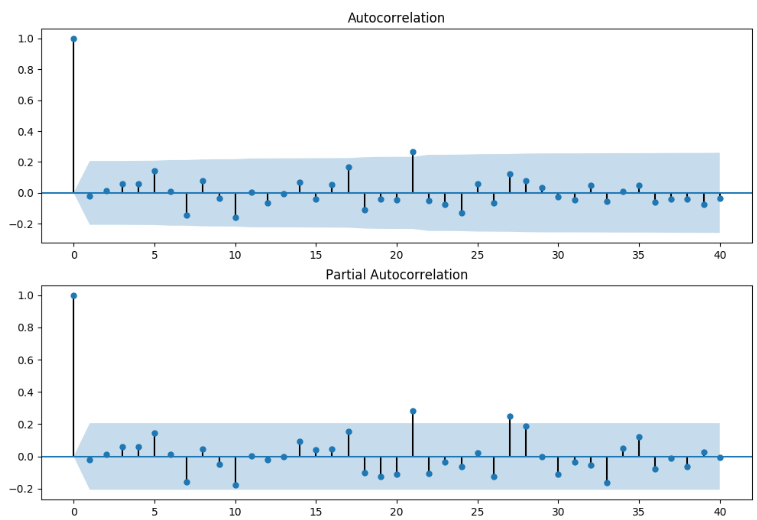

Figure 5:

-

  ```
  ##Normarity test
  print(stats.normaltest(resid))
  fig = plt.figure(figsize=(12,8))
  ax = fig.add_subplot(111)
  fig = qqplot(resid, line='q', ax=ax, fit=True)
  plt.show()
  ```

NormaltestResult(statistic=14.604690828509506, pvalue=0.00067395621352767988)

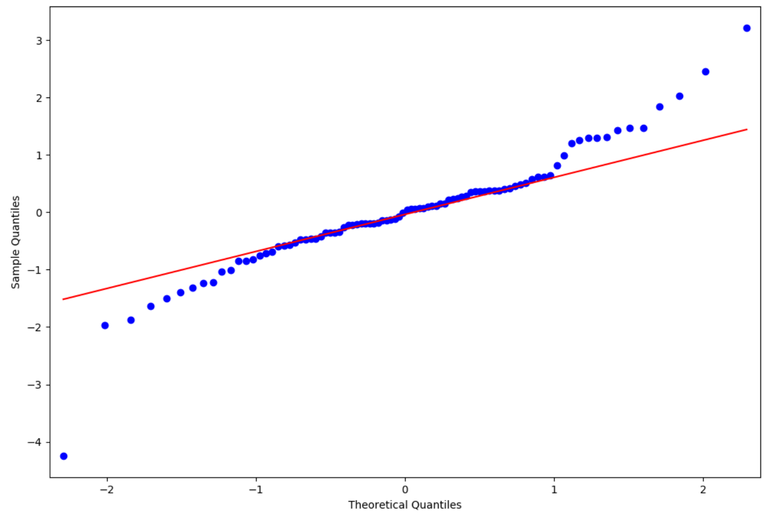

Figure 6:

- D-W

```
##D-W test
print(sm.stats.durbin_watson(arma_mod80.resid.values))
```

    2.04167903544

- Ljunng-Box

```
##Ljunng-Box test
r,q,p = sm.tsa.acf(resid.values.squeeze(), qstat=True)
data = np.c_[range(1,41), r[1:], q, p]
table = pd.DataFrame(data, columns=['lag', "AC", "Q", "Prob(>Q)"])
print(table.set_index('lag'))
```

3

Stata/SAS/Eviews Python Python ARIM

1. Autoregressive Moving Average (ARMA): Sunspots data, url: http://statsmodels.sourceforge.net/devel/examples/notebooks/generated/tsa_arma_0.html#autoregressive-moving-average-arma-sunspots-data.
2. Python_Statsmodels ARIMA , url:http://blog.csdn.net/hal_sakai/article/details/51965657.
3. [ ] · ( ), ( ). —— (2), .
4. , .
5. . (2005). . , *25*(9), 2179-2181.