Aula3_1_AutocovarianciaAutocorrelacao

August 1, 2020

1 Funções de Autocovariância e Autocorrelação

por **Cibele Russo**

Baseado em

- Moretting, P.A.; Toloi, C.M.C. "Análise de Séries Temporais". Blucher, 2004.
- Ehlers, R.S. (2009) Análise de Séries Temporais, http://www.icmc.usp.br/~ehlers/stemp/stemp.pdf. Acessado em 28/06/2020.

Implementações:

- Brownlee, Jason. Introduction to time series forecasting with python: how to prepare data and develop models to predict the future. Machine Learning Mastery, 2017.

Leituras sugeridas:

- https://en.wikipedia.org/wiki/Autocorrelation
- https://otexts.com/fpp2/autocorrelation.html
- https://www.statsmodels.org/devel/generated/statsmodels.graphics.tsaplots.plot_pacf.html
- https://en.wikipedia.org/wiki/Partial_autocorrelation_function

Logo falaremos de modelos autorregressivos, mas antes disso precisamos estudar covariância, correlação e como elas são usadas para descrever a associação entre observações de uma série temporal observada.

Nesta aula, falaremos de

- Função de autocovariância
- Função de autocorrelação
- Gráficos de autocorrelação e autocorrelação parcial

Antes de falar de autocovariância e autocorrelação, o que é covariância e correlação?

Basicamente, a **covariância é uma medida de variabilidade conjunta entre duas variáveis aleatórias**.

Ela mede a força da associação linear entre essas duas variáveis.

E a **correlação é essa medida de associação linear padronizada**, de forma que assuma valores entre -1 e 1.

O sinal da covariância e da correlação indica se as variáveis se associam de forma positiva ou negativa.

1.1 Covariância e correlação

A covariância entre duas variáveis aleatórias X e Y é dada por

$\sigma^2_{XY} = Cov(X,Y) = E[(X - E(X))(Y - E(Y))]$

Sejam σ^2_X e σ^2_Y as variâncias de X e Y, respectivamente.

A correlação entre X e Y é dada por

$$\rho = Cor(X,Y) = \frac{E[(X - E(X))(Y - E(Y))]}{\sqrt{Var(X)}\sqrt{(Var(Y))}} = \frac{\sigma^2_{XY}}{\sqrt{\sigma^2_X}\sqrt{\sigma^2_Y}}$$

Obs: $-1 \leq \rho \leq 1$ e quanto maior a correlação em módulo, mais forte é a associação, positiva ou negativa, entre as duas variáveis.

1.1.1 Coeficiente de correlação de Pearson

O coeficiente de correlação de Pearson entre as variáveis aleatórias X e Y, dada que amostras de X e Y foram observadas, é dado por

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Em Python, podemos usar a função numpy.corrcoef(x,y)

Correlação não significa causalidade!

E então que é autocovariância e autocorrelação?

1.2 Autocovariância

Seja $\{Z_t, t \in \mathbb{Z}\}$ um processo estacionário real com tempo discreto, de média zero.

A função de autocovariância facv para um deslocamento no tempo τ é dada por $\gamma_\tau = E(Z_t Z_{t+\tau})$.

Mais geralmente, a **função de autocovariância** é dada por:

$\gamma(\tau) = E[(Z(t) - \mu)(Z(t+\tau) - \mu)] = Cov(Z(t), Z(t+\tau))$.

$\gamma(\tau)$ é chamado de coeficiente de autocovariância na defasagem τ.

1.3 Autocorrelação

Uma quantidade livre de escala é a **função de autocorrelação**

$\rho(\tau) = \dfrac{\gamma(\tau)}{\gamma(0)}$, para $\tau \in \mathbb{Z}$.

A autocorrelação, também conhecida como correlação serial, é a correlação de um sinal com uma cópia atrasada de si mesma em função do atraso (lag). Informalmente, é a semelhança entre as observações em função do intervalo de tempo entre elas.

(Fonte: Traduzido de https://en.wikipedia.org/wiki/Autocorrelation)

Para entender a autocorrelação, é comum construirmos gráficos e autocorrelação e autocorrelação parcial.

1.4 Autocorrelação em séries temporais

Seja y_k a k-ésima observação da série temporal, e $\bar{y}$ e média amostral da série.

A autocorrelação entre observações com atraso k é dada por

$$\rho_k = \frac{\sum\limits_{t=1}^{n-k}(y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum\limits_{t=1}^{n}(y_t - \bar{y})^2}$$

1.5 Correlograma

O correlograma é uma representação das autocorrelações entre as observações da série temporal.

Ou seja, cada ponto do gráfico representa a correlação entre a série original e a série com o atraso correspondente.

1.6 Autocorrelação parcial

A autocorrelação parcial é uma medida de associação linear de duas variáveis após remover o efeito de outras variáveis que afetam ambas. Ou seja, a autocorrelação parcial com atraso k é a autocorrelação entre y_t e y_{t+k} em que não são contabilizados os atrasos entre 1 e $k-1$.

Na prática, modelos lineares são ajustados para a série "corrente" com a série em atraso como preditor, e os resíduos desse modelo são utilizados para o próximo passo, calcula-se a correlação entre os resíduos e a próxima série em atraso e assim por diante.

1.7 Aplicações

[1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
%matplotlib inline
import statsmodels.api as sm

pkgdir = '/home/cibele/CibelePython/AprendizadoDinamico/Data'

# Dados Passageiros aéreos
df1 = pd.read_csv(f'{pkgdir}/airline_passengers.
↪csv',index_col=0,parse_dates=True)
df1.index.freq = 'MS'

# Dados de nascimentos diários de mulheres
```

```
df2 = pd.read_csv(f'{pkgdir}/DailyTotalFemaleBirths.
↪csv',index_col='Date',parse_dates=True)
df2.index.freq = 'D'
```

[2]:
```
# Funções para cálculo da autocorrelação e autocorrelação parcial

from statsmodels.tsa.stattools import acovf, acf, pacf, pacf_yw, pacf_ols
```

1.8 Funções para cálculo de autocorrelação e autocorrelação parcial

https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.pacf.html

[3]:
```
acf(df1['Milhares de passageiros'])
```

```
/home/cibele/anaconda3/lib/python3.7/site-
packages/statsmodels/tsa/stattools.py:572: FutureWarning: fft=True will become
the default in a future version of statsmodels. To suppress this warning,
explicitly set fft=False.
  FutureWarning
```

[3]:
```
array([1.        , 0.94804734, 0.87557484, 0.80668116, 0.75262542,
       0.71376997, 0.6817336 , 0.66290439, 0.65561048, 0.67094833,
       0.70271992, 0.74324019, 0.76039504, 0.71266087, 0.64634228,
       0.58592342, 0.53795519, 0.49974753, 0.46873401, 0.44987066,
       0.4416288 , 0.45722376, 0.48248203, 0.51712699, 0.53218983,
       0.49397569, 0.43772134, 0.3876029 , 0.34802503, 0.31498388,
       0.28849682, 0.27080187, 0.26429011, 0.27679934, 0.2985215 ,
       0.32558712, 0.3370236 , 0.30333486, 0.25397708, 0.21065534,
       0.17217092])
```

[4]:
```
pacf(df1['Milhares de passageiros'])
```

```
/home/cibele/anaconda3/lib/python3.7/site-
packages/statsmodels/regression/linear_model.py:1406: RuntimeWarning: invalid
value encountered in sqrt
  return rho, np.sqrt(sigmasq)
```

[4]:
```
array([ 1.00000000e+00,  9.54677042e-01, -2.65277317e-01,  5.54695472e-02,
        1.08856215e-01,  8.11257853e-02,  4.12540544e-03,  1.56169553e-01,
        1.03708330e-01,  2.88781439e-01,  2.06918048e-01,  2.41129704e-01,
       -1.58004984e-01, -7.18324604e-01, -8.94806410e-02,  2.21605913e-01,
        1.34622533e-01,  1.15615719e-01,  1.94829396e-01,  9.66561845e-02,
       -2.02158680e-01, -9.36381005e-02, -3.45594572e-01, -1.06170206e-01,
        2.77804723e-01,  5.87815922e-02,  9.86624045e-03,  2.37687367e-01,
        9.40568218e-02, -1.47505422e-01, -1.88609051e-01, -2.52801158e-01,
       -2.57153789e-01, -1.40349613e-01,  1.88263087e-01,  1.30686258e-01,
        5.23902189e-01,  6.91426442e-01,  9.91163921e-01,  3.71021065e+01,
```

```
         -8.85334119e-01])
```

[5]:
```
pacf(df1['Milhares de passageiros'], method='ols')
```

[5]:
```
array([ 1.        ,  0.95893198, -0.32983096,  0.2018249 ,  0.14500798,
        0.25848232, -0.02690283,  0.20433019,  0.15607896,  0.56860841,
        0.29256358,  0.8402143 ,  0.61268285, -0.66597616, -0.38463943,
        0.0787466 , -0.02663483, -0.05805221, -0.04350748,  0.27732556,
       -0.04046447,  0.13739883,  0.3859958 ,  0.24203808, -0.04912986,
       -0.19599778, -0.15443575,  0.04484465,  0.18371541, -0.0906113 ,
       -0.06202938,  0.34827092,  0.09899499, -0.08396793,  0.36328898,
       -0.17956662,  0.15839435,  0.06376775, -0.27503705,  0.2707607 ,
        0.32002003])
```

[6]:
```
pacf_ols(df1['Milhares de passageiros'])
```

[6]:
```
array([ 1.        ,  0.95893198, -0.32983096,  0.2018249 ,  0.14500798,
        0.25848232, -0.02690283,  0.20433019,  0.15607896,  0.56860841,
        0.29256358,  0.8402143 ,  0.61268285, -0.66597616, -0.38463943,
        0.0787466 , -0.02663483, -0.05805221, -0.04350748,  0.27732556,
       -0.04046447,  0.13739883,  0.3859958 ,  0.24203808, -0.04912986,
       -0.19599778, -0.15443575,  0.04484465,  0.18371541, -0.0906113 ,
       -0.06202938,  0.34827092,  0.09899499, -0.08396793,  0.36328898,
       -0.17956662,  0.15839435,  0.06376775, -0.27503705,  0.2707607 ,
        0.32002003])
```

[7]:
```
acf(df2['Births'])
```

[7]:
```
array([ 1.        ,  0.21724118,  0.15287758,  0.10821254,  0.09066059,
        0.09595481,  0.09104012,  0.19508071,  0.14115295,  0.06117859,
        0.04781522,  0.04770662, -0.01964707,  0.02287422,  0.08112657,
        0.11185686,  0.07333732,  0.01501845,  0.07270333,  0.06859    ,
        0.09280107,  0.26386846,  0.14012147,  0.06070286,  0.08716232,
        0.05038825,  0.0650489 ,  0.11466565,  0.1552232 ,  0.12850638,
        0.10358981,  0.09734643,  0.04912286,  0.04022798,  0.05838555,
        0.05359812,  0.10151053,  0.08268663,  0.0912185 ,  0.11192192,
        0.05652846])
```

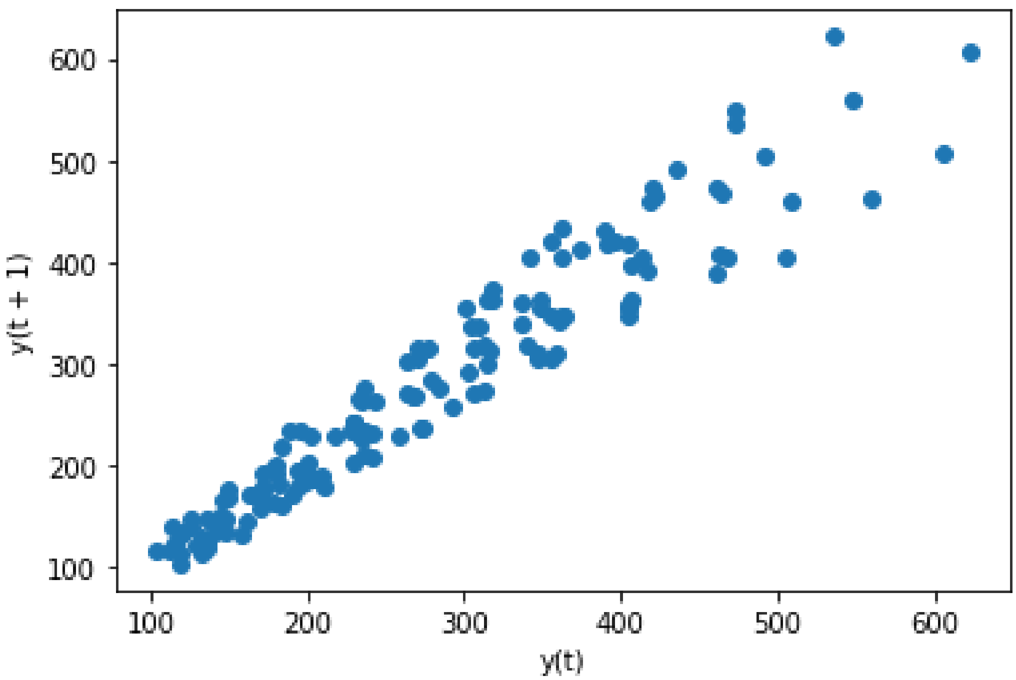
[8]:
```
pacf(df2['Births'], method='ols')
```

[8]:
```
array([ 1.        ,  0.2179641 ,  0.11388341,  0.06139271,  0.05014092,
        0.05597304,  0.0483302 ,  0.16061715,  0.061602  , -0.0245556 ,
       -0.00774957,  0.00782231, -0.07054357,  0.00367697,  0.05073901,
        0.06869818,  0.02855912, -0.03000743,  0.04890835,  0.05079005,
        0.06672663,  0.23464568,  0.01251561, -0.05701977,  0.03051524,
       -0.03035958, -0.00790227,  0.08244362,  0.05410409,  0.00122559,
        0.04213413,  0.03829265, -0.0147851 ,  0.02911748,  0.01617994,
```

```
-0.03759518,  0.03129664,  0.01440593,  0.05191662,  0.07161683,
-0.00544217])
```

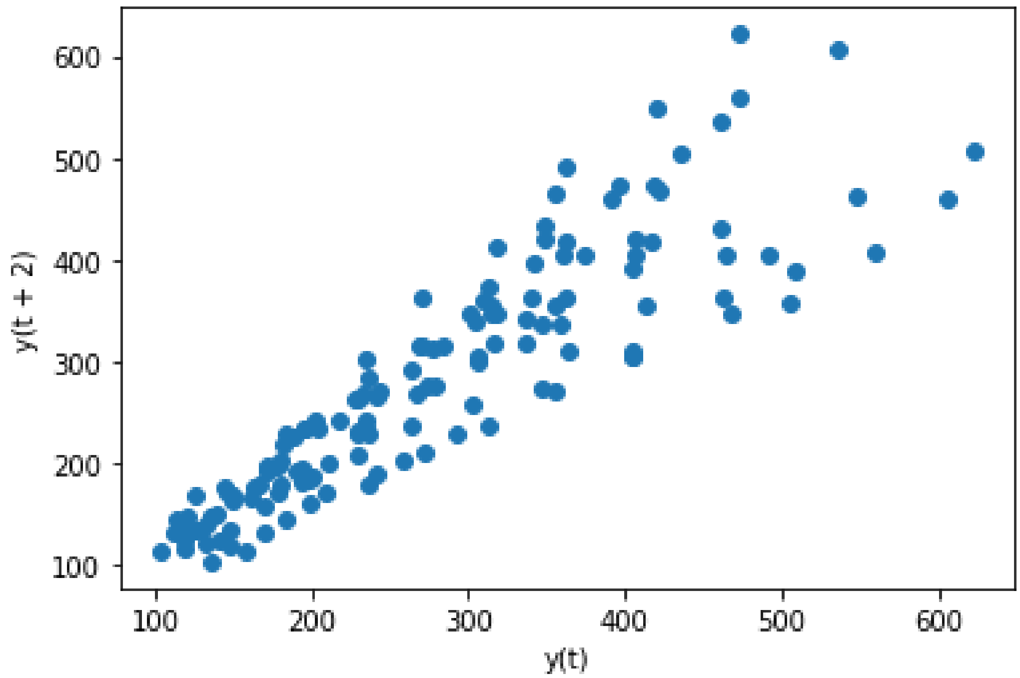
2 Representação gráfica da autocorrelação

[9]:
```python
from pandas.plotting import lag_plot

lag_plot(df1['Milhares de passageiros']);
```



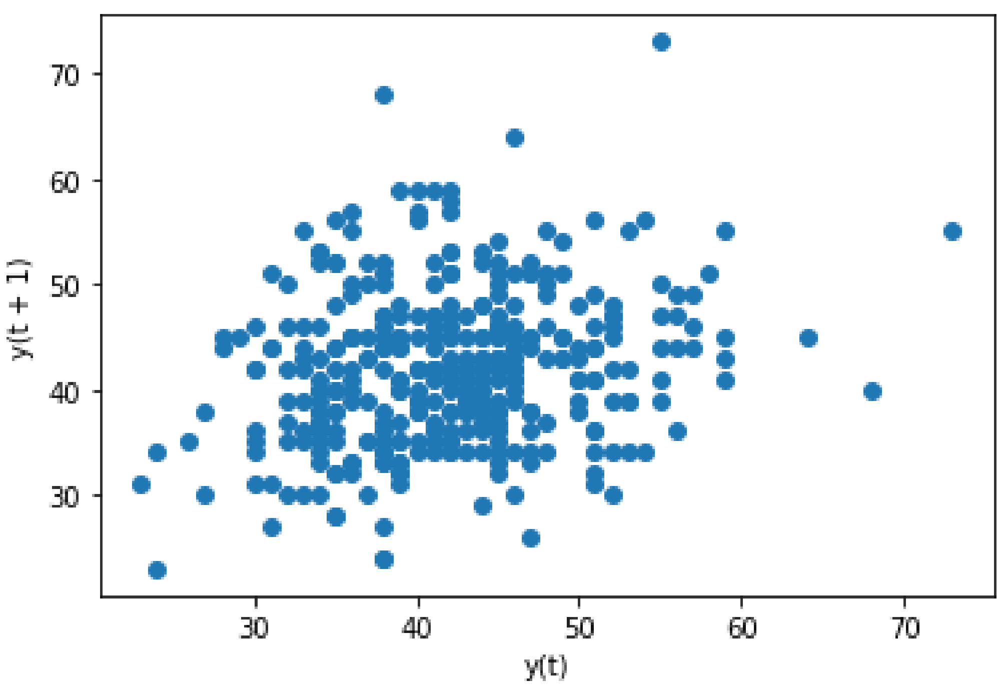
O gráfico acima indica forte correlação entre a série original e a série com atraso 1.

[10]:
```python
from pandas.plotting import lag_plot

lag_plot(df1['Milhares de passageiros'],lag=2);
```



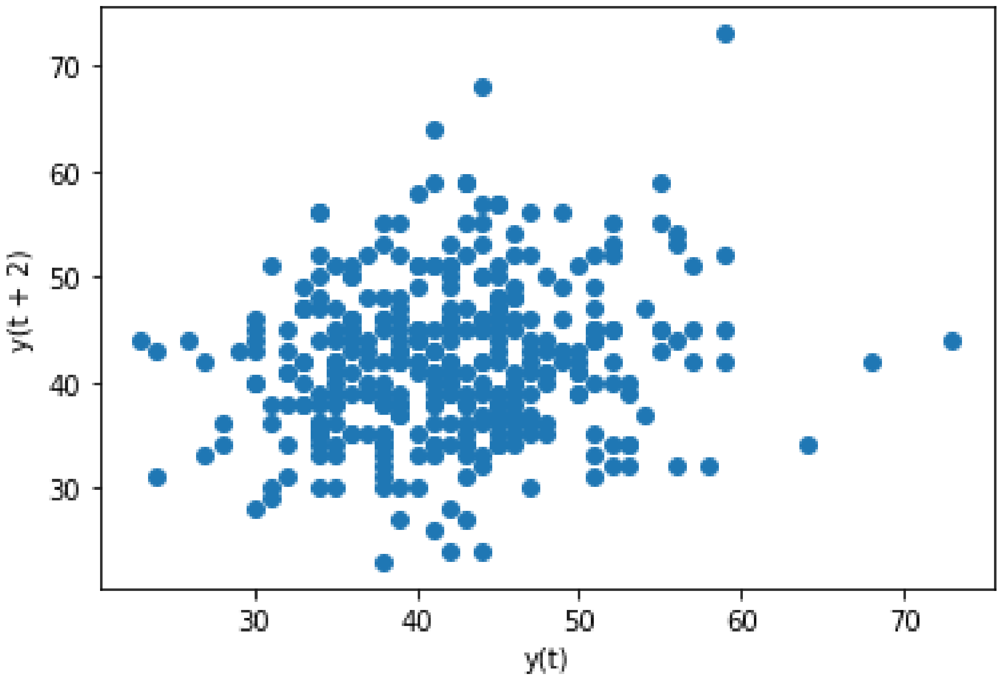
Dados dos nascimentos

[11]:
```
lag_plot(df2['Births']);
```



[12]:
```
lag_plot(df2['Births'], lag=2)
```

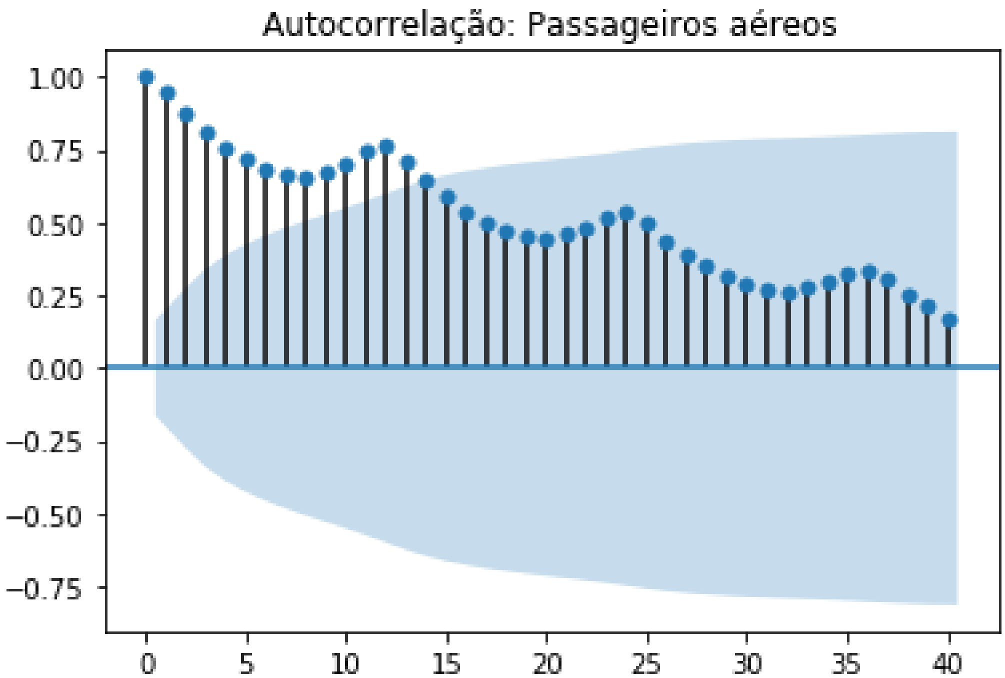
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f763d464b50>



2.1 Representação gráfica da autocorrelação

[13]:
```
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
```

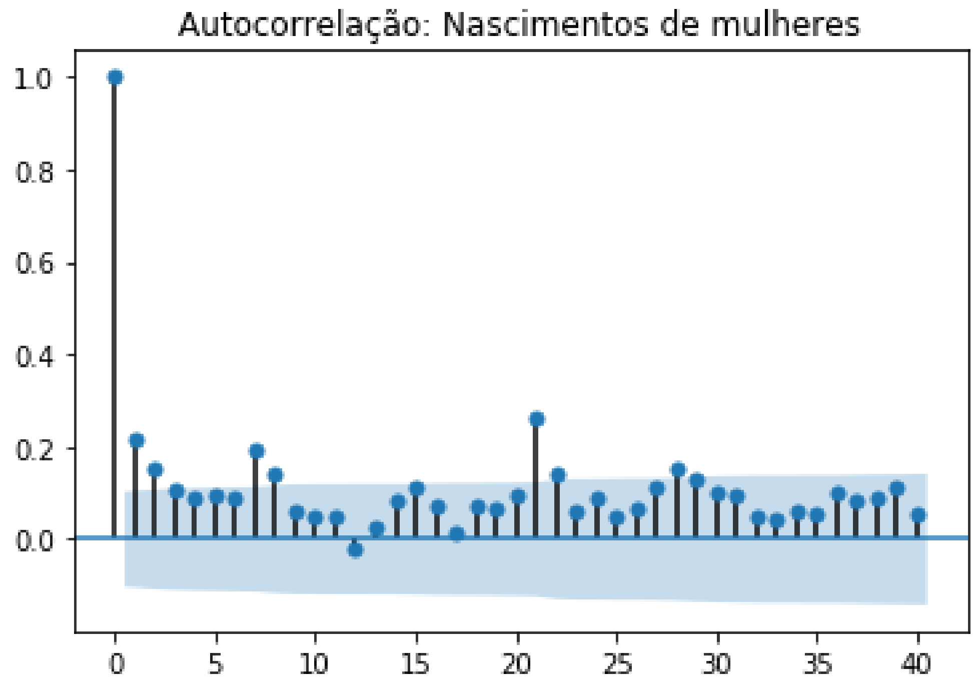
[14]:
```
title = 'Autocorrelação: Passageiros aéreos'
lags = 40
plot_acf(df1['Milhares de passageiros'],title=title,lags=lags);
```



Observe o efeito da sazonalidade e portanto da não-estacionariedade no gráfico da autocorrelação.

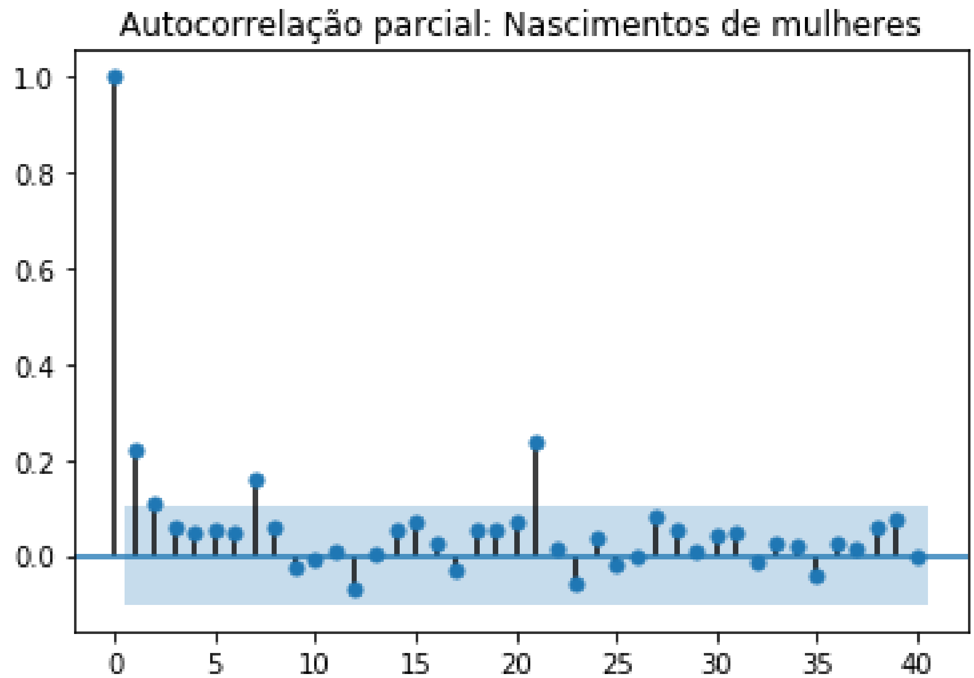
A região em azul representa um intervalo de confiança para a correlação, e quando um ponto ultrapassa essa região temos um indicativo de significância da correlação observada.

[15]:
```
title='Autocorrelação: Nascimentos de mulheres'
lags=40
plot_acf(df2['Births'],title=title,lags=lags);
```



2.2 Representação gráfica da autocorrelação parcial

[16]:
```
title='Autocorrelação parcial: Nascimentos de mulheres'
lags=40
plot_pacf(df2['Births'],title=title,lags=lags);
```



Exercícios:

Obtenha gráficos de autocorrelação e autocorrelação parcial para os dados de COVID-19 do estado de São Paulo. Lembre-se de utilizar os dados completos como fizemos na Aula 2.

Obtenha gráficos de autocorrelação e autocorrelação parcial para os dados da PETR4 da Prática 2.

[ ]:

```

```