	<pre>in [-0.6, -0.7, -0.8, -0.9]: # a ar_model(1, [phi], 18, 5000)</pre>	
plot plt.	<pre>at(f"AR(1) model with phi = {phi}") c_acf(y, lags=50) # b c.show() c_pacf(y, lags=50) # b c.show() el with phi = -0.6 Autocorrelation</pre>	
0.75 - 0.50 - 0.25 -		
0.00 - -0.25 - -0.50 -		
-0.75 - -1.00	0 10 20 30 40 50 Partial Autocorrelation	
0.75 -		
0.25 - 0.00 - -0.25 -	**************************************	
-0.50 - -0.75 - -1.00 -	0 10 20 30 40 50	
1.00 — 0.75 - 0.50 -	el with phi = -0.7 Autocorrelation	
0.25 - 0.00 - -0.25 -		
-0.23 -0.50 - -0.75 - -1.00		
1.00 T 0.75 -	0 10 20 30 40 50 Partial Autocorrelation	
0.50 - 0.25 - 0.00 -	***************************************	
-0.25 - -0.50 - -0.75 -		
-1.00 AR(1) mode	0 10 20 30 40 50 el with phi = -0.8 Autocorrelation	
0.75 - 0.50 - 0.25 -		
0.00 - -0.25 - -0.50 -		
-0.75 - -1.00 -	0 10 20 30 40 50 Partial Autocorrelation	
0.75 - 0.50 - 0.25 -		
0.00 - -0.25 - -0.50 -	***************************************	
-0.75 - -1.00 AR(1) mode	0 10 20 30 40 50 el with phi = -0.9	
1.00 — 0.75 - 0.50 -	Autocorrelation	
0.25 - 0.00 - -0.25 -		
-0.50 - -0.75 - -1.00	0 10 20 30 40 50	
1.00 - 0.75 -	Partial Autocorrelation	
0.50 - 0.25 - 0.00 -	***************************************	
-0.25 - -0.50 - -0.75 -		
[61]: from sta	0 10 20 30 40 50 will check the time series for stationarity using the Augmented Dicker atsmodels.tsa.stattools import adfuller in [-0.6, -0.7, -0.8, -0.9]: ar model(1, [phi], 18, 5000)	r-Fuller test.
y = prin prin prin prin AR(1) mode	ar_model(1, [phi], 18, 5000) nt(f"AR(1) model with phi = {phi}") nt(" ") nt(adfuller(y)) nt(" ") el with phi = -0.6 81438649802, 0.0, 0, 4999, {'1%': -3.431658793968827, '5%': -2.86211	345383404, '10%': -2.567077853953267}, 14014.671388204191)
AR(1) mode (-46.29234 AR(1) mode	81438649802, 0.0, 0, 4999, {'1%': -3.431658793968827, '5%': -2.86211 el with phi = -0.7 48147266736, 0.0, 2, 4997, {'1%': -3.431659318070175, '5%': -2.86211 el with phi = -0.8 3401203077, 0.0, 0, 4999, {'1%': -3.431658793968827, '5%': -2.862118	576928539, '10%': -2.567077977213785}, 14008.55459356905)
AR(1) mode (-307.975	el with phi = -0.9 77737231967, 0.0, 0, 4999, {'1%': -3.431658793968827, '5%': -2.86211 s of the results: p-value > 0.05 - This implies that time-series is non-st	345383404, '10%': -2.567077853953267}, 14159.9281657359)
<pre>What ha [62]: y = ar_m print(f" plot_acf plt.show</pre>		= 1.1 in our function.
plt.show plot_pac plt.show AR(1) mode /var/folde array before y[t] = 0 /Users/man	ef(y, lags=50) # b v() el with phi = 1.1 ers/z6/r8xtfhp56gd9rzqzptq9vv880000gn/T/ipykernel_51865/1885854207.p fore performing this operation. (Deprecated NumPy 1.25.) c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal() rynaborovyk/Desktop/semestr-4/Time Series Analysis/lab4/lib/python3.	9: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from yo 2/site-packages/statsmodels/tsa/stattools.py:693: RuntimeWarning: invalid value encountered in divide
1.00 TO -0.75 -	vf[: nlags + 1] / avf[0] Autocorrelation	
0.50 - 0.25 - 0.00 - -0.25 -		
-0.25 - -0.50 - -0.75 - -1.00 -	_0.04 0.5-	
/Users/mai r[0] = /Users/mai r[k] = /Users/mai	(x ** 2).sum() / n rynaborovyk/Desktop/semestr-4/Time Series Analysis/lab4/lib/python3. (x[0:-k] * x[k:]).sum() / (n - k * adj_needed)	2/site-packages/statsmodels/regression/linear_model.py:1481: RuntimeWarning: overflow encountered in square 2/site-packages/statsmodels/regression/linear_model.py:1483: RuntimeWarning: overflow encountered in multiply 2/site-packages/numpy/core/_methods.py:49: RuntimeWarning: invalid value encountered in reduce
0.75 - 0.50 -	Partial Autocorrelation	
0.25 - 0.00 - -0.25 -		
-0.50 - -0.75 - -1.00	-0.04 -0.02 0.00 0.02 0.04	
Calculat [63]: y = ar_m	ult shows that we can't calculate the ACF and PACF for the AR(1) mode te 5000 values of the AR(2) model with coefficients: 1.3, -0.7 model(2, [1.3, -0.7], 8, 5000) AR(2) model")	I with phi = 1.1.
<pre>plot_acf plt.show</pre>	F(y, lags=50) v() cf(y, lags=50) v()	
0.75 - 0.50 - 0.25 -		
0.00 - -0.25 -		
-0.50 - -0.75 - -1.00 -	0 10 20 30 40 50	
1.00 — 0.75 - 0.50 -	Partial Autocorrelation	
0.25 -		
0.00 -	**************************************	
0.00	0 10 20 30 40 50	
0.00	0 10 20 30 40 50 Fing the structure of PACFs for AR(1) and AR(2) models: **Model(1, -0.7, 18, 5000)** **AR(1) model")** **Ef(y, lags=50)**	
0.00	0 10 20 30 40 50 Fing the structure of PACFs for AR(1) and AR(2) models: model(1, -0.7, 18, 5000) AR(1) model") Ef(y, lags=50) M(2) MR(2) model") Ef(y, lags=50) M(3) MR(2) model") Ef(y, lags=50) M(4) Compared to the process of the pro	9: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from yo
0.00	0 10 20 30 40 50 Fing the structure of PACFs for AR(1) and AR(2) models: **model(1, -0.7, 18, 5000)** **RR(1) model")** **if(y, lags=50)** **w() **model(2, [1.3, -0.7], 8, 5000)** **RR(2) model")** **if(y, lags=50)** **w() **model(2, [1.3, -0.7], 8, 5000)** **model(2, [1.3, -0	9: DeprecationNarming: Conversion of an array with ndim > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from yo
0.00	0 10 20 30 40 50 Fing the structure of PACFs for AR(1) and AR(2) models: model(1, -0.7, 18, 5000) NR(1) model") Ef(y, lags=50) N() model(2, [1.3, -0.7], 8, 5000) NR(2) model") Ef(y, lags=50) N() el ers/z6/r8xtfhp56gd9rzqzptq9vv880000gn/T/ipykernel_51865/1885854207.pg ore performing this operation. (Deprecated NumPy 1.25.) c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal()	9: DeprecationKarning: Conversion of an array with nulm > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from yo
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(1, -0.7, 18, 5000) if(y, lags=50) if() model(2, [1.3, -0.7], 8, 5000) iR(2) model") if(y, lags=50) if() el ers/z6/r8xtfhp56gd9rzqzptq9vv880000gn/T/ipykernel_51865/1885854207.p ore performing this operation. (Deprecated NumPy 1.25.) c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal() Partial Autocorrelation	9: DeprecationWarning: Conversion of an array with ndin > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from yo
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(1, -0.7, 18, 5900) RR(1) model") if(y, lags=50) i() model(2, [1.3, -0.7], 8, 5000) RR(2) model") if(y, lags=50) i() el ers/z6/r8xtfhp56gd9rzqzptq9vv880000gn/T/ipykernel_51865/1885854207.p ore performing this operation. (Deprecated NumPy 1.25.) c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal() Partial Autocorrelation	9: DeprecationWarning: Conversion of an array with ndia > 0 to a scalar is deprecated, and will error in future. Charm you extract a single element from yo
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(1, -0.7, 18, 5000) RR(1) model") if(y, lags=50) if() model(2, [1.3, -0.7], 8, 5000) RR(2) model") if(y, lags=50) if() el ers/z6/r8xtfhp56gd9rzqzptq9vv880000gn/T/ipykernel_51865/1885854207.p ore performing this operation. (Deprecated NumPy 1.25.) c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal() Partial Autocorrelation Partial Autocorrelation	On DeprecativeWarnings Conversion of an array with notes 5 % to a scalar is deprecated, and will across in facture. Passes you extract a striple element from you
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(1, -e.7, 18, 5000) RR(1) model") f(y, lags=50) i() el ers/z6/r8xtfhp56gd9rzqzptq9vv880000gn/T/ipykernel_51865/1885854207.p ore performing this operation. (Deprecated NumPy 1.25.) c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal() Partial Autocorrelation Partial Autocorrelation	3). DeprecationNorting: Conversion of an energy with eddir > 6 to a scalar is deprecator, and will enter in fature. Ensure you estruct a single absence from yo
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(1, -0.7, 18, 5900) IR(1) model") If(y, 1ags-59) If(y, 1ags-59) If(y, 1ags-59) If(y, 1ags-59) If(y, 1ags-50) If(y, 1ags-	
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(1, -8.7, 18, 5000) ing(1) model") if(y, lags=50) if(y, lags	
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: codel(1, -0.7, 18, 5000) R(1) model') (f(y, lags-50) (/) codel(2, [13, -0.7], 8, 5000) R(2) model') (f(y, lags-50) (/) el era/26/ñatfhp56gd0rqzptq0v88000gm/7/jpykernel_51865/1885354207.p ore performing this operation. (beprecated Numby 1.25.) c in p.dot(phi, y[t-p:t][::-1]) in p.random.normal() Partial Autocorrelation Partial Autocorrelation O 10 20 30 40 50 Correlations in MA models a function that calculates the values of MA(q) model(parameter burning) rocecos(n + b burnin) np.zeros(n + b burnin) rop.zeros(n + b burnin) tj int jere (+q, n + b un-dot) tj int jere (+q, n + b un-dot) liate 5000 values of MA(1) model b) Calculate the autocorrelation (AC de(1), locel') (y, lags-50) # b (y) (y, lags-50) # b (y)	determines how many initial values are discarded)
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: codel(1, -e.7, 18, 5880) KR(1) model") (f(y, lags-50) (X) codel(2, [13, -e.7], 8, 5880) KR(2) model") (f(y, lags-50) (X) core performing this operation. (Deprecated Numby 1.25.) c in p.dot(phi, y(t-p:t[i:-1]) in p. random.normal() Partial Autocorrelation Partial Autocorrelation O 10 20 30 40 50 Correlations in MA models a function that calculate sthe values of MA(q) model(parameter burning the interpretation of the property of the pro	determines how many initial values are discarded)
0.00 -0.250.500.751.00 Compari [64]: y = ar_m print("A plot_pac plt.show AR(1) mode /var/folde array befor y[t] = 0 0.75 - 0.50 - 0.250.500.751.00 AR(2) mode 1.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: indel(1, -e.7, 18, 5000) ing the structure of PACFs for AR(1) and AR(2) models: indel(1, -e.7, 18, 5000) ing the structure of PACFs for AR(1) and AR(2) models: indel(2, [1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models: indel(2, [1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models: indel(2, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(2, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(2, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(2, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(2, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(2, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000) ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000 ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000 ing the structure of PACFs for AR(1) and AR(2) models indel(1, 1.3, -e.7], 8, 5000 ing the structure of PACFs for AR(1) and AR(2) models i	determines how many initial values are discarded)
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: indef(1),0-7, 18, \$989) id(2) model") f(y, lags-59) (v) el electry/fo/fskthp56pdfsrqrptqtww880000gpv7/1pykernel_51865/1885854287.p c + np.dot(phi, y t-pt [::-1]) + np.random.normal() Partial Autocorrelation Partial Autocorrelation O 10 20 30 40 50 Correlations in MA models a function that calculates the values of MA(q) model(parameter burning) to the partial form of the partial form of the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the autocorrelation (AC model(1), [10,18]); 28, 5000) # 0 f(y, lags-50) # 0 cl Autocorrelation	determines how many initial values are discarded)
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: indef(1),0-7, 18, \$989) id(2) model") f(y, lags-59) (v) el electry/fo/fskthp56pdfsrqrptqtww880000gpv7/1pykernel_51865/1885854287.p c + np.dot(phi, y t-pt [::-1]) + np.random.normal() Partial Autocorrelation Partial Autocorrelation O 10 20 30 40 50 Correlations in MA models a function that calculates the values of MA(q) model(parameter burning) to the partial form of the partial form of the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the values of MA(q) model(parameter burning) to the partial function in the calculates the autocorrelation (AC model(1), [10,18]); 28, 5000) # 0 f(y, lags-50) # 0 cl Autocorrelation	determines how many initial values are discarded)
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model (1, -0-2, 18, 5000) ##(3) model*) ##(4) indel*) ##(4) indel*) ##(5) indel*) ##(6) indel*) ##(6) indel*) ##(7) indel*) ##(8) indel*(8) inde	determines how many initial values are discarded)
0.000.250.500.751.00 - Compari [64]:	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model (1, -0-2, 18, 5000) ##(3) model*) ##(4) indel*) ##(4) indel*) ##(5) indel*) ##(6) indel*) ##(6) indel*) ##(7) indel*) ##(8) indel*(8) inde	determines how many initial values are discarded)
0.00	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: redu(10, model) (10, model)	determines how many initial values are discarded)
0.000.250.500.751.00 Compari [64]: y = ar_m	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: post(1, 1, 1, 1), 15, 5890 (1), 16, 16, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18	determines how many initial values are discarded)
0.000.250.500.751.00 - Compari [64]: y = ar m print ("A plot_pac plt. show y = ar my print ("A plot_pac plt. show of array before y[t] = array y[t	0 10 20 30 40 50 ing the structure of PACPs for AR(1) and AR(2) models: second(1, -a, 7, 18, 5986) (10, 100, 100, 100, 100, 100, 100, 100,	determines how many initial values are discarded)
0.000.250.500.751.00 - Compar [64]: y = ar m	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: post(1, 1, 1, 1), 15, 5890 (1), 16, 16, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18	determines how many initial values are discarded)
0.000.250.500.751.00 - Compari [64]:	0 10 20 30 40 50 ing the structure of PACFs for AR(1) and AR(2) models: model(a), (a, 1, 2, 3), (a), (a), (a), (a), (a), (a), (a), (a	determines how many initial values are discarded)
0.000.250.500.751.000.751.000.751.000.1.5how AR(1) mode AR(1) mode AR(2) mode -0.250.500.5	0 10 20 30 40 50 ing the structure of PACTs for ARI1) and ARI21 models. social(1, 10.7, 10, 1000) (10) social(2) (1.5, 10.2), 9, 5000) (10) social(2) (1.5, 10.2), 9, 5000) (10) social(3) (1.5, 10.2), 9, 5000) (10) social(3) (1.5, 10.2), 9, 5000) (10) social(3) (1.5, 10.2), 11.3) is species security (1.5, 10.2) Partial Autocorrelation Partial Autocorrelation Partial Autocorrelation O 10 20 30 40 50 concentrations in MA models of Arian models of Ar	determines how many initial values are discarded)
0.000.250.500.751.001.001.001.001.011.011.011.011.021.031.041.051.00 -	0 10 20 30 40 50 ing the structure of PACEs for AR(1) and AR(2) models: ssss1(1, -6.7, 31, seec) ((1) statis) ((2) statis) ((3) statis) ((4) statis) ((5) statis) ((6) statis) ((6) statis) ((7) statis	determines how many initial values are discarded)
0.000.250.500.751.00 - Compari [64]:	0 10 20 30 40 50 from the structure of PACEs for ANU1) and ANU2) models: code(1, 1, 1, 7, 41, 1006) (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1	determines how many initial values are discarded)
0.000.250.500.751.00 - Compari [64]:	0 10 20 30 40 50 ing the cructure of MACH for ARICS and ARICS models. stability, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	determines how many initial values are discarded)
0.00	0 20 20 30 40 50 rought of structure of PACIDS for ARCID made ARIZI models: detail.1, 19.7, 19. 5960; detail.2, 19. 5960; de	diamondo has reger ful documente sistentials value and diabon. Julium 1994 in Valor to diabits, acrea.
0.00	0	diamondo has reger ful documente sistentials value and diabon. Julium 1994 in Valor to diabits, acrea.
0.00	0 10 20 30 40 50 Ing the souther of PATS for ARILLY models: 40 11 20 30 40 50 For Livery 1, 4, 1982; For Livery 1, 4, 1982	does 1000 to a 100 principal december of a 100 principal d
0.00	0	does 1000 to a 100 principal december of a 100 principal d
0.00	10	does 1000 to a 100 principal december of a 100 principal d
0.00	9 20 20 30 40 50 90 90 90 90 90 90 90 90 90 90 90 90 90	does 1000 to a 100 principal december of a 100 principal d
0.00	19	does 1000 to a 100 principal december of a 100 principal d
0.00	19	does 1000 to a 100 principal december of a 100 principal d
0.00 -0.25 -0.75	0 20 20 30 40 50 20 10 10 10 10 10 10 10 10 10 10 10 10 10 1	does 1000 to a 100 principal december of a 100 principal d
0.00	0 10 20 30 40 50 40 50 40 50 40 50 40 50 40 50 40 50 40 50 40 50 40 50 40 50 40 50 40 50 50 40 50 50 40 50 50 50 50 50 50 50 50 50 50 50 50 50	
0.00	0 20 20 30 40 50 20 10 10 10 10 10 10 10 10 10 10 10 10 10 1	
0.00	10 20 20 20 20 20 20 20	
0.00	10 20 20 20 20 20 20 20	
0.00	### 10	

Lab 5: ARIMA models 1 Task 1 Correlations in AR models

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

y = np.zeros(n + burnin)
Generate the time series according to the AR(p) model

y[t] = c + np.dot(phi, y[t-p:t][::-1]) + np.random.normal()

import numpy as np
import matplotlib.pyplot as plt

def ar_model(p, phi, c, n, burnin=100):

for t in range(p, n + burnin):

Function that calculates the values of AR(p) model(parameter burnin determines how many initial values are discarded):