|  |  |  |
| --- | --- | --- |
| **What is a Stationary Model?** | 1. Subpopulations of have the same mean for each *t*. Restated, **the mean does not depend on time** (*t*). 2. Subpopulations of X for a given time have a finite and constant variance for all t. **Restated, the variance does not depend on time.** 3. The correlation of X\_(t\_1 )and X\_(t\_2 ) depends only on t\_2- t\_1. That is, the **correlation between data points depends only on how far apart they are in time**, not where they are in time.   Unit 1:  1.4.4: Determining if temperature data is stationary. Concept check says yes. Thought I believe the answer is no. Seasonal data, temperature is dependent on the month of a year  1.7.6: Using ACF to determine which data noctula or lavon is stationary  Async videos says Noctula, however the ACFs shows Lavon being similar in lags (negative vs. positive) after accounting for noise. | **Mean**  **Var**  xdf = read.csv(file.choose(),header = TRUE)  x = as.numeric(paste(xdf$Adj.Close))  x = x[!is.na(x)]  n=length(x) #n = 1509  nlag=1508 #n-1  m=mean(x)  v=var(x,na.rm = TRUE)  n = length(x)  gamma0=var(x)\*(n-1)/n  gamma0  aut=acf(x,lag.max=1508) #n-1  sum=0  for (k in 1:nlag) {sum=sum+(1-k/n)\*aut$acf[k+1]\*gamma0}  vxbar=2\*sum/n+gamma0/n #note the mult of sum by 2  vxbar  **Compare ACF**  plotts.sample.wge(lavon)  x = lavon  n = length(lavon)  acf(x[1:trunc(n/2)], lag.max = trunc(n/2))  acf(x[round(n/2):n], lag.max = round(n/2))  **gamma and rho**  n = length(x)  var(x)\*(n-1)/n  See Gamma And Rho Calculation Spreadsheet  ar <- acf(x)  ar$acf    x = gen.sigplusnoise.wge(n=100,coef = c(3,1.5), freq = c(0.05,0.35), psi=c(0,2)) |
| **Spectral Analysis** | **Periodic functions**  *f* (*x*) is periodic function with period *p* if *p* is the smallest value such that *f* (*x*) *= f* (*x + kp*)for all *x* and integer *k.*  **Psuedo-periodic data**  Data are pseudo-periodic with period *p* if *p* is the smallest value such that a cycle appears to repeat itself.  **Aperiodic functions/data**  *f* (*x*) is non-periodic (*aperiodic*) if no such *p* exists. | **Frequency** =number of cycles per unit (period = cycle)  =1/period    parzen.wge(x) |
| **Filtering** | **High-pass filters:**   * “Pass” high-frequency behavior and “filter out” lower-frequency behavior * A difference is a high-pass filter   **Low-pass filters:**   * “Pass” low-frequency behavior and “filter out” higher-frequency behavior * The 5-point moving average smoother is a low-pass filter   **Butterworth Filter:** | **Difference**  dif = diff(x, lag = 1)  **5 Point Moving Average Filter**  ma = stats::filter(x, rep(1,5)) / 5  ma.= ma[!is.na(ma)]  plot(ma, type = "l") |
| **Autoregressive Models: AR(1)** | AR(1) Process is stationary if and only if || < 1  When ||>1, the realization is explosive and is not typical of realizations seen in practice.  Positive   * ***Realizations*** seem to be “wandering,” aperiodic in nature. * ***Autocorrelations*** are damped exponentials. * ***Spectral densities*** have peaks at *f* = 0, which is consistent with the behavior of the realizations.   Negative   * ***Realizations*** seem to be “oscillating,” that is, if *Xt* is above the mean, then the strong tendency is for *Xt*+1 to be below the mean and so on. * ***Autocorrelations*** are damped, oscillating exponentials. ***Spectral densities*** have peaks at *f* = .5 (i.e. a cycle length of 2). This is consistent with the up-and-down behavior in the realizations. | **Zero Means formula**  Backshift  Characteristic equation  Root |
| **AR(2)** | An AR(2) model is stationary if and only if the roots of the characteristic equation are greater than 1 in absolute value (lie outside the unit circle).  AR(2) will have 2 roots, both r1 and r2 are real or they appear as complex conjugate pairs, r1 = a + bi and r2 = a – bi (you cannot square root a negative value).  AR(2) with 2 Real Roots (One Positive \ One Negative)   * System Frequency peaks at 0 and .5 * Depending on which root is closest to unit circle the dominant characteristics with appear   + ACF Damped exponential (positive phi)   + ACF Oscillatory behavior (negative phi)   AR(2) with 2 Positive Real Roots   * Wandering behavior with system frequency at 0 * ACF Damped exponentially   AR(2) with 2 Negative Real Roots   * System Frequency at .5 * ACF has oscillatory behavior   AR(2) with complex conjugate roots   * System Frequency peak between 0 and .5 * Realization will have pseudo cyclic behavior | Zero means form  Backshift  Characteristic equation  Quadratic formula  If complex conjugate roots  Validate with factors.wge()  Calculate System Frequency for Complex Conjugate Roots |
| **AR(p)** | An AR(*p*) model is stationary if and only if the roots of the characteristic equation are greater than 1 in absolute value.  AR(*p*) models reflect a mixture of these *first-* and *second-order* behaviors in the:   * Realizations * Autocorrelations * Spectral densities |  |
| MA(q) | * Function of white noise * MA is written as a GLP and is always stationary * Spectral Densities do not have “peaks”, they have “dips”   Invertibility   * 2 models can have to same system frequency and * An MA(*q*) model is invertible if and only if the roots of the MA-characteristic equation, *q*(*z*) = 0, are greater than 1 in absolute value. | MA(q)  Zero Mean Form  Backshift  MA-Characteristic equation  MA(1) Formulas  MA(2) Formulas |
| **ARMA(p,q)** | **a stationary and invertible**  ARMA(*p,q*) process if:   1. Roots of *j*(*z*) = 0 are all outside the unit circle 2. Roots of *θ*(*z*) = 0 are all outside the unit circle |  |
| **AIC5** | Use AIC5 to identify the optimal model  Aic5.wge(x,p=5, q=5). May need to change p or q to increase the ARMA(p,q) values |  |
| **Psi Weights** | MA(q) is already is GLP Form  Used to establish prediction limits on forecasts  Psi.weights.wge(x, lag.max=5) |  |
| **Signal-Plus-Noise Models** | Xt = st + Zt   * st is deterministic signal (linear) * Zt is a zero-mean, stationary process |  |
| ARIMA(p,d,q) | The *autoregressive integrated moving average process* of orders *p, d*, and *q* (denoted ARIMA(*p,d,q*)) is a process, *Xt ,* whose differences (1*- B*)*d X*t satisfy a (stationary) ARMA(*p,q*) model, where *d* is a non-negative integer.  All roots are outside the unit circle. However (1-B)d has a root of 1, resulting in a non-stationary model.  reaches infinity when , thus the ACF will be close to one, and true autocorrelations are equal to 1 for all k. Sample ACF will dampen due to nature of calculation.  The (1*- B*)*d* factor dominates the stationary components in realization, autocorrelations, and spectral densities (*f* = 0). | Formula  gen.arima.wge()  Use artrans.wge(x,1) for each d we want to take a difference from |
| Seasonal Models – ARIMA(p,d,q) with s = # | Contain factor (1 – Bs) | gen.aruma.wge() s and d can be used  Use artrans.wge(x, c(rep(0,s-1), 1))  (1-B4)   |  |  |  |  | | --- | --- | --- | --- | | **Factor** | **Roots** | **Abs Recip** | ***f*** | | 1 - *B* | 1 | 1 | 0 | | 1 + *B2* | ±*i* | 1 | .25 | | 1 + *B* | 1 | 1 | .5 | |
| **Steps to Analyze Realization** | 1. Create a model and plot    1. plotss.wge    2. Note distinctive features    3. Meantion any domain knowledge on data 2. If phis \ thetas, determine if stationary \ invertible using factors    1. factors.wge    2. All phis and thetas need to be outside unit circle    3. Note frequency and periods, aperiodic vs. pseudo frequency 3. If ARIMA or ARUMA, analyze true auto correlation and spectral density 4. Remove nonstationary data    1. artrans.wge(x,1) for each d in (1-B)d    2. artrans.wge(x,c(rep(0,s-1),1)) where (1-Bs) 5. Analyze plotts    1. Plotts.sample.wge    2. Reference back to factor table 6. Determine best ARMA(p,q) model using AIC5   Steps to take out the non-stationary portions a realization   * Determine if seasonal or adjustment with a (1-b) (artrans.wge) * Analyze plots again * Adjust if needed * Finalize with aic5 to determine best model to create estimates with | (1-B12)   |  |  |  |  | | --- | --- | --- | --- | | **Factor** | **Root(s)** | **Abs Recip** | ***f*** | | 1 – *B* | 1 | 1 | 0 | | tmp.gif | .866 + .5*i* | 1 | .083 | | 1 - *B* + *B2* | .5 + .866*i* | 1 | .167 | | 1 + *B2* | +*i* | 1 | .25 | | 1 + *B* + *B2* | -.5 + .866*i* | 1 | .333 | | tmp.gif | -.866 + .5*i* | 1 | *.417* | | 1 + *B* | *-*1 | 1 | .5 | |
| **Forecast AR(p)** | See equation  When generating data add the mean in to the equation | fore.arma.wge(x,phi=0,theta=0,n.ahead=20,plot=TRUE,limits=FALSE) |
| **Psi Weights** | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |  |  | 1 | +.7B | +.24B2 | -0.132B3 | -0.302B4 | -0.2832B5 | -0.159216B6 |  | |  | 1-1.2B+.6B2 | 1 | -.5B |  |  |  |  |  |  | |  |  | 1 | -(+)1.2B | +(-).6B2 |  |  |  |  |  | |  |  |  | .7B | -0.6B2 |  |  |  |  |  | |  |  |  | (-).7B | -(+)0.84B2 | +(-)0.42B3 |  |  |  |  | |  |  |  |  | 0.24B2 | -0.42B3 |  |  |  |  | |  |  |  |  | (-)0.24B2 | -(+)0.288B3 | +(-)0.144B4 |  |  |  | |  |  |  |  |  | -0.132B3 | -0.1440B4 |  |  |  | |  |  |  |  |  | -(+)0.132B3 | +(-).1584B4 | -(+)0.0792B5 |  |  | |  |  |  |  |  |  | -0.302B4 | +0.0792B5 |  |  | |  |  |  |  |  |  | -(+)0.302B4 | +(-)0.36288B5 | -(+)0.1812B6 |  | |  |  |  |  |  |  |  | --0.28368B5 | +0.1812B6 |  | |  |  |  |  |  |  |  | --(+)0.28368B5 | +(-)0.340416B6 |  | |  |  |  |  |  |  |  |  | --0.159216B6 |  | | |
| **Forecast** | Confident Intervals   * Need to calculate * sum(x\_hat$resid^2) / (length(x) - 2) # 2 phis | CI Formula  Noise Variance  #use $wvm from fore.arma.wge model  sum(x\_hat$resid^2) / (length(x) - 2) # 2 phis # Sigma\_sq\_a (variance)  sqrt(sum(x\_hat$resid^2) / (length(x) - 4) ) #sigma\_a |
| Cross Validation | fore.arma.wge(x,phi=0,theta=0,n.ahead=20,plot=TRUE,limits=FALSE)  Use lastn=TRUE to forecast the last n.ahead values in the series  Use Average Square Error to estimate how well your model did | s <- length(x) - 30 + 1 #30 = n.ahead value  n <- length(x)  ASE <- mean((x\_hat$f-x[s:n])^2)  ASE |
| **Forecasting with ARIMA** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |