|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Stationary Conditions Examples**  **Unit1** | **Example Met** | **Met Write up** | **Lecture Example Slide** | **Example Not Met** | **Not Met Write up** | **Lecture Examples Slides** | **Code Realization, ACF, Comparison ACF** |
| **Condition 1**  Mean does not depend on time Subpopulations of have the same mean for each *t. Eg. Slides41-43* | https://lh6.googleusercontent.com/IvcdTKARHlGQoy6Fix9Ge3W0Pl49RPo1P-LtmB6Ab2TRrYzze7LIgUkSDZcpFZDnEwis2S_0u1ZOebd8qlRjt4rFCsKGcbRZZ22f8ONRe73G9_CoGBk2QUzMvIiPXU5wY1MAsycx | Just judging from the series and not knowing much about consumer sentiment it is reasonable to believe that the mean consumer sentiment is not dependent on time. That is, if we were able to observe another realization of this series, we might be just as likely to observe troughs were there are peaks and peaks where there are troughs as any other outcome.  Wandering or Aperiodic data. |  | https://lh6.googleusercontent.com/SEpRaDQQcewd966Lh_RPFO8OFj50viAwhHVwMyHE1O8ItQr2ZoPfjrYTIZzdEjYhieewR2TyXLQgOfdN9vjsKL_flT2_C2A0fiQeXQrjSYsSJ-hr1sa5EW96TxmtCUfWwW2e6MCN | Looks like there is at least one frequency in the series and that there may be some wandering behavior as well. We may need to know more about the beer industry here but it is likely that beer is constantly produced to keep up with demand which is probably seasonal. If this is the case then the mean beer production would be conditional on the month and thus not stationary. Pseudo cyclic, or cyclic behavior due to periods being present in the data |  | ```{r}  Realization = gen.arma.wge(500,.95,0,plot = TRUE,sn = 784)  ```  ```{r}  acf(Realization[1:250],plot = TRUE)  acf(Realization[251:500],plot = TRUE)  ``` |
| **Condition 2**  The variance does not depend on time. Subpopulations of *X* for a given time have a finite and constant variance for all *t*. Eg. 49-51 | https://lh6.googleusercontent.com/IvcdTKARHlGQoy6Fix9Ge3W0Pl49RPo1P-LtmB6Ab2TRrYzze7LIgUkSDZcpFZDnEwis2S_0u1ZOebd8qlRjt4rFCsKGcbRZZ22f8ONRe73G9_CoGBk2QUzMvIiPXU5wY1MAsycx | There is not a lot of evidence against constant (and finite) variance. Again, we have to imagine how other realizations would behave. |  | https://lh6.googleusercontent.com/SEpRaDQQcewd966Lh_RPFO8OFj50viAwhHVwMyHE1O8ItQr2ZoPfjrYTIZzdEjYhieewR2TyXLQgOfdN9vjsKL_flT2_C2A0fiQeXQrjSYsSJ-hr1sa5EW96TxmtCUfWwW2e6MCN | Again, this may take a little domain knowledge as it is quite possibly that the variance in production is increasing or at least changing over time. This could be because of the addition of breweries or a shifting in tourist populations / numbers(as an example.) |  |  |
| **Condition 3**  The correlation between data points depends only on how far apart they are in time, not where they are in time. The correlation of and depends only on | https://lh4.googleusercontent.com/rk94lnWCZjH8yt9nYfytN65ZlUaHfpfJe6_ITdTAmQHaWfq84DW1qMtsSlBwzv8YEa1k0edKONmgeTOllahPSYjLdQhJlT8SykLiheJfORFmVJAkQsCmiuh79w3kderc1kUNR-x_ | Judging from the ACFs, there is little to no evidence that the covariance is different from the first half of the series to the second. Judging from the ACFs, the covariance seems to not depend on where in the series we are, only how far apart the observations are. |  | https://lh5.googleusercontent.com/eiL9HzC5atW82CjdFapuC998qQWOUX_JkNd-OJI1BD8YFCD2scmW7Gc2CjdxlF_6OVm4Z25r2ib34LJKMiH9GjPthiP7n-2-cTLnsSjco21DzrngNWIQazknJ9kIbrrq47NljyyOhttps://lh5.googleusercontent.com/r-vvqmm1mfFhj5Y5lsYtAiAZQDYSms88x0zP3jGEGhj1b5h3EeaLpWHHlRoSANR3RHW9ATx-CiWK20c60JJ4T9FNWzoOnXp-n9MOz_sdOMePOpPCBrBj1igU51_wIz9AMH4BcYu-https://lh3.googleusercontent.com/SAr5R8u-5S23BwnNaeznsxYfrdBVY0GpCob_sMsKZOr5Tw95zjVx_fu6r7pXpFPi7OvdPU8gaOEctp2-20y9outY9NnMeIEA3xfHRzJH0GvdOeIeVK2cR6z6zCFfIdmVjNePwjG2 | Judging from the ACFs, there is strong evidence that the covariance structure is dependent on where we are in the series. |  |  |
| **White Noise Simulation** |  | *y* = gen.arma.wge(*n* = 250)  Time = seq(1,250,length.out = 250)  plot(Time,*y*) |  | *y* = gen.arma.wge(*n* = 250)  Time = seq(1,250,length.out = 250)  plot(Time,*y,*type = “l”) |  | plotts.wge(*y*) #from tswge package | **Variance** |
| Unit 1 Equations  HW Solutions for Example | Sample Mean | Expected Value | Variance (Std Dev of Data)  Standard Deviation | Auto Covariance  Gamma 0 is variance of the data Equation below gives it for higher gammas    Gamma below is autocovariance | Auto Correlation = Serial Correlation  (Standardized autocovariance)    Rho 0 is always 1    Gamma sub k dividied by the std dev to get rho sub k  Example HW1 | **Covariance/Autocovariance**  Next, consider two random continuous variables *X* and *Y*  We would like a measure of how much they move together.  That is, as *X* gets bigger, does *Y* tend to get smaller, larger, or is *Y* **independent** of *X*?  This can be measured with the **covariance** of *X* and *Y*: ***Cov(X,Y).*** | **Correlation/Autocorrelation**  The correlation between *X* and *Y* allows us to obtain a measure of dependence between *X* and *Y* that is scaled so that it must be between –1 and 1 inclusive.  Correlation, which you are very familiar with, is simply the scaled covariance; we simply divide the covariance by the standard deviation of both *X* and *Y* and we have the correlation.  \*Know that covariance and correlation (or autocovariance/autocorrelation) are basically the same, except that correlation is standardized (always between -1 and 1) and covariance is based on the units of measurement\* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Realization, ACF, Spectral Density Interpretations**  **Unit 2** | **Graph Display** | **Period / Frequ Calculation** | **Spectral Density** | **Write Up** | **Paired Realizations with Spectral Density Examples**  **Unit 2 BreakOut** | **Code for Realization, ACF, Parzen Window** |
| Period |  | Period – 2pi  Frequency = 1/2pi = .159 |  | This is a sin period with a frequency of .159. As we have highlighted to the left the spectral density shows a peak reflecting our calculated frequency at .159. | =  = | ```{r}  data(airlog)  plotts.wge(airlog) #plot realization  parzen.wge(airlog) #plot spectral density  parzen.wge(airlog,trunc = 70) #plot spectral density with different truncation point  plotts.sample.wge(airlog) #realization, Acf, periodogram, spectral density  ``` |
| Pseudo Period |  | Period = 12 months  Frequency = 1/12 = .083333 |  | This is a pseudo period of 12 month time frame of temp data. We can see a peak in the spectral density of our calculated frequency of .03333. | =  =  = | ```{r}  data(sunspot.classic)  parzen.wge(sunspot.classic)  plotts.sample.wge(sunspot.classic) #realization, Acf, periodogram, spectral density  ``` |
| Aperiodic / Wandering |  | Period: Aperiodic  Frequency = N/A |  | Intuitively, for the length of this data set, the series has not displayed evidence of completing a single period/cycle. Consequently, the shortest the period could be here is 250, making the biggest the largest the frequency could be is . | =  = |  |
|  |  |  |  |  |  | The Nyquist Frequency  This is the highest observable frequency when sampling at the integers because you need at leaset two observable points to observe one cycle. When sampling at the integers, a series with a frequency higher than .5 may become aliased with a series with lower frequency. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Frequency Filters Unit 3** |  | **Original Data** | **Filtered Data** | **Example** |  | **Code** |
| High-pass filter | “Pass” high-frequency behavior and “filter out” lower-frequency behavior  Difference filter |  |  |  |  | ```{r}  data(fig1.21a)  plot(fig1.21a, type = "l")  ```  ```{r}  dif = diff(fig1.21a,lag = 1)  plot(dif,type = "l")  ```  ```{r}  Realization = gen.sigplusnoise.wge(200,coef = c(5,0),freq = c(.1,0), vara = 10, sn = 1)  ```  ```{r}  dif = diff(Realization,lag = 1)  plot(dif,type = "l")  parzen.wge(dif[!is.na(dif)])  ``` |
| Low-pass filter | “Pass” low-frequency behavior and “filter out” higher-frequency behavior  Moving average Filter |  |  |  |  | ```{r}  plot(fig1.21a, type = "l")  ```  ```{r}  ma = filter(fig1.21a,rep(1,5))/5  plot(ma,type = "l")  ```  ```{r}  Realization = gen.sigplusnoise.wge(200,coef = c(5,0),freq = c(.1,0), vara = 10, sn = 1)  ```  ```{r}  ma = filter(Realization,rep(1,5))/5  plot(ma,type = "l")  parzen.wge(ma[!is.na(ma)])  ``` |
| General Linear Process  Linear Filter with white noise input` |  |  | AR, MA, and ARMA are all special cases of GLPs  This is useful for confidence intervals |  |  |  |
| **Model Type** | **Model Equation** | **GLP Process Graph** | **Stationarity** | **Condition 1** | **Condition 2** | **Condition 3** |
| Autoregressive Model  AR(p) p = 1  AR(1)  AR(1) is stationary is and only if  Zero Mean Form:    Second Zero Mean Form    Backshift Form  From    To backshift form below |  | -Are very useful for describing stationary data that move forward in time  - is called the moving average constant  -The AR(1) model says that the value of the process at a time depends on the value of the process at time plus a random noise component.  - This is a sensible way to describe the way a time series might progress in time.  -This is similar to the simple linear regression model, but in this case, the “independent variable” is a value of the dependent variable at a prior time period | An AR(1) process is stationary if and only if  Stationary  Not Stationary | -Mean does not depend on time | -Note the variance is finite as long as , and also note that it does not depend on t-time | -Note that the decreases exponentially with and only depends on and not .  **Bonus: Spectral Density**    -Monotonically increasing or decreasing in depending on |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit 4** | **Notes** | **Equations** | **Roots** | **Stationary if** | **Condition 1(Mean)** | **Condition 2(Variance)** | **Condition 3** |
| AR(2)  If my = 0 then:    Moving average constant:    Alternate zero mean form | -This model “looks like” a multiple regression model with two independent variables.  -But in this case, the “independent variables” are values of the dependent variable at the two previous time periods.  -It specifies that the value at time *t* is a linear combination of values at the two previous time periods plus a random noise component that enters the model at time *t*. | Zero Mean Form:  Operator Notation:  Characteristic Polynomial:  Characteristic Equation:  Example:  Zero Mean Form  Operator Notation:  Characteristic Equation:  Roots: | Either  -2 Real Roots  -r1&r2 complex conjugate pairs | All roots are **outside** the unit circle  **|*r*|-1 < 1**  (abs recip of root)  *(\* |r|-1 > 1 is “explosive”)*  ***Stationary if and only if the roots of the characteristic equation are outside of the unit circle. >1*** | Expected value of x t is mu if stationary and does not depend on time | Variance does not depend on time | In the AR(2) case the spectral density behavior depends on the roots being real of complex    If complex: between 0 and .5  If real: peak at 0 or .5  Real Root > 0:  peak at *f* = 0  Real Root < 0:  peak at *f* = .5  Complex Roots:  Peak at *system freq* |
| AR(p)    Where    NOTE:    First-Order Factors:  -Associated with Real Roots  -Contribute AR(1) type behavior to the AR(p) model  -Are associated w’ “system freq” f0=0 if a is negative  Second-Order Factors:  -Associated with complex conj roots  -Contribute cyclic AR(2) type behavior to the AR(p) model associated with “system freq”    -Coefficients alone cannot tell us the AR(p) model characteristics and therefore pick out their behaviors from a graph. Therefore, we us the factor tables to derive the characteristics and chart the models. | -Looks like a multiple regression model  -Says the value at time t is a linear combination of the p previous values plus a random noise component  -Looks complicated | Zero Mean Form:    Or    Operator Form:  or    Characteristic Equation: | Broken into  AR(1)  See if |*φ*1| < 1  AR(2)  -2 Real  -Complex Conjugate Roots | All roots are **outside** the unit circle  AR(1)  |*φ*1| < 1  AR(2)  |*r*|-1 < 1  (abs recip of root)  *(\* |r|-1 > 1 is “explosive”)*  ***Stationary if and only if the roots of the characteristic equation are outside of the unit circle. >1*** | Factoring the AR(p)  -Roots of a general pth order polynomial equation cannot always be found using mathematical formulas (such as the quadratic formula)  -Can be solved using numerical methods  1.Real Roots  2.Complex conjugate Roots  -Can always be factored as a product of:  --First-order (linear) factors  --Second-order (quadratic) factors | AR(3) Model    Characteristic Equation:    Factor Characteristic Equation:    AR(4) Model  Characteristic Equation:      Characteristic Equation: | NOTE:    First-Order Factors:  -Associated with Real Roots  -Contribute AR(1) type behavior to the AR(p) model  -Are associated w’ “system freq” f0=0 if a is negative  Second-Order Factors:  -Associated with complex conj roots  -Contribute cyclic AR(2) type behavior to the AR(p) model associated with “system freq”    AR(p) models reflect a mixture of the first- and second- order behaviors in the:  Realizations  Autocorrelations  Spectral Densities |
| **Factor Table:**  Provide information about roots of characteristic equation to help determine stationarity and underlying frequency domains of | **Factor**  First- or second- order factor  or | **Root**  Root(s) of firs- or second-order equations associated w/ factor:  or | | **Modulus**  1/abs reciprocal  **Absolute reciprocal of root**  , where is the root in second column  Note: If for all roots, then the process is **stationary** | | **System frequency**  First-order factors:  & Second-order factors | |
| **Dominant Behavior Models**  Roots closest to the unit circle behavior will dominate the visual characteristics in the realization, acf, and spectral density  Model A: roots equally close to the unit circle (close to 1)  Model A-r: real root closer to the unit circle (closer to 1)  Model A-c: complex root closer to the unit circle (closer to 1) |  |  | |  | |  | |
| **Factor Table Example 1:**    This is an AR(3)  Is this stationary? | This AR(3) is made up of an AR(1) [first-order factor] and an AR(2) [second-order factor] | This model has first-order positive real root. This is associated with AR(1) behaviors:  -Realization: wandering  -ACF: slowly dampening  -Spectral Density: peak at 0  This model has a second -order complex conjugate roots. This is associated with complex conjugate behaviors:  -Realization: pseudo-cyclic behavior showing frequency of about w/ cycle length of  -ACF: Damped sinusoidal autocorrelations w/ a period (cycle length) of about 6  -Spectral Density: peak at | | Upon reviewing the absolute reciprocals, we can quickly see this third-order polynomial model is stationary. Since both of the above absolute reciprocals are < 1 that means the roots associated which those absolute reciprocals are outside the unit circle and therefore the model is stationary.  We choose to look at the absolute reciprocal since the complex conjugate root is not easily interpretable as >1 or outside of the unit circle for stationary judgement. | | Since this is an AR(3) the model will show a combination of all the behaviors that have been listed thus far. Below are the realization, acf, and spec den of the data we just broke down with the table.    **Code**  - X\_t - 1.95X\_t-1 + 1.85X\_t-2 - .855X\_t-3 = a\_t  ```{r}  #Factor table code  factor.wge(phi = c(1.95,-1.85,.855))  ```  - Plot a realization along with true autocorrelations and spectral density  ```{r}  plotts.true.wge(phi = c(1.95,-1.85,.855))  ``` | |
| **Factor Table Example 2:**    Is this AR(4) Stationary? | Broken down into two second order models | 2 pairs of complex conjugate roots | | Both of these are < 1 so it is stationary | | We think we would see two frequency in this data .06 and .45.  Below realization reflects both the low and high frequencies that we see above. We see the sinusoidal behavior from complex conjugate roots. In the spectral frequency we see the double spike for both frequencies above .06 and .45. | |
| **Factor Table Example 3:** | This 4 order polynomial is broken down into 2 first- and 1 second-order | We see 1 positive and 1 negative root. As well as a pair of complex conjugate roots. | | The above highlighted abso. recip. is >1, therefore it is explosively nonstationary. | | We see the explosive behavior below in the realizations. | |
| **Roots Examples** | **Step 1** | **Step 2** | **Step 3** | | | | **Answer** |
|  | Characteristic Equation: | Factored Characteristic Equation: | Calculate Roots:  1.25 > 1: outside the unit circle  1.33 >1 : outside the unit circle | | | | Due to both real roots being outside the unit circle it implies that is stationary |
|  | Operator Notation:  Characteristic Equation: | Factored Characteristic Equation: | Calculate Roots:  .667 < 1: inside the unit circle  10>: outside the unit circle | | | | Due to only one root being outside the unit circle (one is inside the unit circle) this model is nonstationary. If realizations were generated, they would be explosive. |
|  | Characteristic Equation: | Factored Characteristic Equation can Calculate Roots:  THEY ARE COMPLEX therefore we solve with quadratic formula.  1.1180 > 1: outside the unit circle | | | | | Due to both complex conjugate roots being outside the unit circle it implies that is stationary |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit 5** |  |  | Mean | Variance | Realizations | Autocorrelations | Spectral Density |
| **MA(q) Models**    Theta’s are a function of white noise terms, AR(p) are a coefficient of Xt  See below the theta’s (q) are the negitve psi weights:    Zero Mean Form:  or    Operator zero mean form:    MA-Characteristic Equation: | MA(q) is a finite General Linear Process (GLP) and is always stationary.  All roots are **outside** the unit circle | Example:    MA(1) Zero Mean Form    MA(1) Operator Form    MA(1) Zero Mean Operator Form:    MA(1) Characteristic Equation: | Stationary process  Does not depend on time | Does not depend on t  Stationary Process | Difficult to discern and typically unrealistic due to *ρk* = 0 when *k* > | Rho 0 is always 1  Rho 1 is above equation  Rhok = 0 for k>1, this is always the case. There is no slow dampening. It drops off immediately. | The moving average spectral density will have dips in them instead of peaks like the AR(1)  *φ*1 > 0: dip at *f* = 0  *φ*1 < 0: dip at *f* = .5 |
| **Model Multiplicity**  It the case when we have 2 different models that have the same autocorrelations. We prevent this by restricting our models to **invertible models**. | **Invertibility**  An MA model is invertible if, and only if, all roots are outside the unit circle: |z|>1 | Invertibility Example 1  Xt = at - .8at-1  Xt = at -.8Bat  = (1-.8B)at  Characteristic Equation  1-.8B = 0  Root:  Z = 1/.8 ; |z| > 1  Root is outside the unit circle and therefore this equation is invertible. |  | Invertibility Example 2  Xt = at – 1.25 at-1  Characteristic Equation:  1-1.25B = z  Root:  Z = 1/1.25; |z| < 1  Root is inside the unit circle and therefore this equation could have model multiplicity |  |  | Code |
| **ARMA(p,q) Models**    Where    Zero Mean Form:    Operator Notation: | Cancellation |  | ARMA Model Spectral Densities  **ARMA Model**    **AR Component** | **MA Component** |  | **ARMA Model** | ARMA Model Spectral Densities  Are a blend of the AR and MA characteristics    Note: the y axis is what changes to accommodate the blend of characteristics |
| **ARMA(p,q) Psi Weights**  AR, MA AND ARMA processes can all be expressed as a general linear process or GLP  GLP: | MA Psi Weights    AR Psi Weights | Psi Weights  -Are used in prediction limits on forecasts  -They are more difficult for AR(p) models, p>1 and ARMA(p,q) models  \*\*Because psi weight for an AR model are infinite  -We leverage code to calculate psi weights psi.weights.wge | ARMA Summary  -An MA(q) is an ARMA(0,q) model  -An AR(p) is an ARMA(p,0) model  -A stationary an invertible ARMA(p,q) model can be writte in 2 forms  GLP:    Infinite Order: |  |  |  | -An ARMA(p,q) can be written as an infinite order AR(form 2)  -That is, an ARMA(p,q) model may be able to be approximated by a higher order AR |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit 6** |  |  |  |  |  |  |  |
| **Deterministic Signal-Plus Noise Models**  **St is a deterministic signal**  **Zt is a zero-mean, stationary process**    Xt is non-stationary because we have a non-constant mean. This means the mean (mu) depends on time.  Simple Signal    Quadratic Signal    Cyclic Signal Modeled by Cos | **It is very difficult to determine from the naked eye if a deterministic signal is present in the data or not.** | Code:  ```{r}  #Generate a linear trend with white noise with noise variance 100  gen.sigplusnoise.wge(100,b0=2,b1=4,vara = 100)  #Intercept 0; slope 0; white noise variance of 10, therefore this is white noise  gen.sigplusnoise.wge(100,b0=0,b1=0,vara = 10)  #Generate 100 realization, intercept:0; slope:0, with noise from an AR(1) model with phi = .975  gen.sigplusnoise.wge(100,b0=0,b1=0,phi=.975,vara=10)  ## This shows us something with a lot of serial correlation  ``` | Code Example:    - X\_t = 5 cos(2\*pi(.1)t + 2.5) + Z\_t: Z\_t~AR(1)  - 5 is the coefficient  - 2\*pi is the phase shift  - .1 is frequency  - 10 is period  - Z\_t is generated noise from an AR(1)  ```{r}  gen.sigplusnoise.wge(100,coef = c(5,0),freq = c(.1,0), psi = c(.25,0), phi = .975, vara = 10)  ``` |  |  |  | Code Example:  - Factored equation we were give the below model version:  - In order to get the coefficients we use mult.wge  ```{r}  #equation put into mult.wge to create coefficients  parms = mult.wge(c(.975),c(.2,-.45),c(-.53))  #to see model coeffecients from above inside of parms  parms$model.coef  ```  Once Coefficients are found this would be how you could write the model:    - Below is a small generated version from the above model to see how different they can all appear  ```{r}  gen.arma.wge(160,phi = parms$model.coef,vara = 1)  ``` |
| **ARIMA(p,d,q) Models**  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)dXt satisfy (stationary) ARMA(*p,q*) model, where *d* is a non-negative integer.          and    Lie outside the unit circle. And    Clearly has roots inside the unit circle \*\*The Non-Stationary Part\*\*  They are in fact d roots of 1! | **Autocorrelation Function:**  For ARIMA models, φ1 = 1 and the variance of Xt is infinite, so the **autocorrelations are undefined**  Unboundedly large because as approaches 1 then approaches .  ARIMA(0,1,0) required a new definition for autocorrelations: Extended Autocorrelations for ARIMA -> we continue to call them autocorrelations. | **ARIMA(0,1,0)** | **ARIMA Summary**  NOTE: The (1-B)d factor dominates the stationary components.  -The realization  -The ACF  -The spectral densities have peak at f = 0  are all dominated in traits  For d>1, this domination is even more striking  The true autocorrelations will all equal 1  All sample autocorrelations will always damp (in part becase of the way they are calculated)  **Slowly dampening sample autocorrelations is an indication of ARIMA data and if we should difference the data in order to create a stationary data set to model.** |  |  |  |  |
| **Seasonal Models**  **ARUMA**        **Note that this model has:**   * Stationary factors: *j*(*B*)and *q*(*B*) * An ARIMA-type factor: (1 - *B*)*d* * A seasonal factor: 1 - *Bs*   **Or generally, with ARIMA component**    *\*****(1-Bs)*** *is seasonal factor*  *All have unit roots and absolutely reciprocals of 1, is what makes the seasonal component the non-stationary component\*\* is observed in the Factor Tables below* | All roots are **outside** the unit circle (AR and MA)  *\*AR and MA Must be stationary and invertable.*  ***(1-Bs) and [potentially]***  **(1-*B*)d  are non-stationary components** | Example:  For quarterly sales data, this model says that sales in the current quarter are equal to the sales for this quarter last year plus a random-noise term. | Note major difference from ARIMA model | Notes about Quarterly Model Example:  -Quarterly behavior is present in realization (but not as clear as it was in the initial realization of length n=20)    -Sample autocorrelations at lags 4, 8… are large, which is consistent with the seasonable model.  -> That is Xt, Xt+4, Xt+8 would be expected to be “similar”    -The spectral estimate has peaks at f=0, .25, and .5 | ***Realization***  *Seasonal* cyclic behavior combined with AR and MA behaviors | **Autocorrelations**  ACF will damp slowly, but **has peaks at lags that are multiples of *s***  Combo of AR and MA, but **dominated** by **(1-*B*s) and(1-*B*)d because both have |*r*|-1 = 1 (ON the unit circle)** | **Spectral Density**  Combo of AR and MA; but **dominated by frequencies of (1-*B*s) factors (they all have |*r*|-1 = 1) in a seasonal model** |
| **ARUMA** How to Stationarize  **This is done by changing the code since it is Bs** |  | ```{r}  x=gen.aruma.wge(n=80, s=4, sn = 81) #tswge function to generate ARIMA and Seasonal Models  Dif = artrans.wge(x,c(0,0,0,1)) #Take out the (1-B^4)  aic5.wge(Dif) #Check the structure of the noise  ``` |  |  |  |  |  |
| **ARUMA / Seasonal Factor Tables Code**  S = 4  factor.wge(phi=c(0,0,0,1))  factor.wge(phi=c(rep(0,3),1))  S = 12  factor.wge(phi=c(0,0,0,0,0,0,0,0,0,0,0,1))  factor.wge(phi=c(rep(0,11),1))  factor.wge(c(0,0,0,1))  factor.wge(c(0,0,0,0,0,0,0,0,0,0,0,1))  factor(c(rep(0,3),1))  factor(c(rep(0,11),1))  Seasonal model with s = 5  factor(c(rep(0,4),1)) |  | 1-B4 | 1-B12 |  |  |  | -Calculating forecasts by hand, won’t require it but might have a full one done and ask to pull out psi weights etc  -ASE in bonus  -Probability Limits = Confidence intervals |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unit 7** |  |  |  |  |
| Forecasting is extrapolation. It’s important to be very careful when extrapolating. But sometimes we need to extrapolate, aka sometimes we need to forecast! | Forecast future behavior ofa time series given a finite realization of it’s past.  -The use of ARMA models for this purpose is popular. | In this unit, we will assume that we know the true model p, q, as well the phis() and thetas () |  |  |
| is the forecast of given data up to time  This is where X is the actual value, X\_hat is the forecast  is the forecast origin  is the lead time, ie. The number of time unit (steps ahead) which we want to forecast |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Forecasting Using an AR(1)**      ***As forecasts get farther away from the last known point the forecast approaches the mean*** | We know X1 – X10 and we want to forecast X12  To obtain X12 we use the below equation:  but we don’t know mu(), X11, or a12. This has to be calculated iteratively. Which means we calculate X11 before X12  First use instead of , new formula:    Iteratively:      =0 because it is equal to the unconditional mean which is zero  we know all    … to forcast we calculate | Example of AR(1) Forecast  X1…..,X80 observed from  X80=21.77, =24.17, t0=80; (data are in tswge file fig6.inf) | The AR(1) forecast model just dampens back to the mean see graphical display of calculations found to the left below.  Positive Forecasting AR(1)  We see the graph approach the mean    Negative Forecasting AR(1)  We see a heavy sinusoidal pattern, like the ACF as the forecast approaches the mean | - Forecasting a positive AR(1) model  ```{r}  data(fig6.1nf)  plotts.wge(fig6.1nf)  ```  ```{r}  fore.arma.wge(fig6.1nf,phi=.8,n.ahead=20,plot=TRUE,limits=FALSE)  ```  - Forecasting a negative AR(1) model  ```{r}  x1 = gen.arma.wge(100, phi = -.8)  fore.arma.wge(x1, phi=-.8,n.ahead = 20,plot = TRUE, limits = FALSE)  ``` |
| **Forecasting Using an AR(2)**    Reduces to the below:      ***As forecasts get farther away from the last known point the forecast approaches the mean*** |  | We observe a sinusoidal pattern in the prediction that finds it’s way back to the mean. Like the ACF and **complex conjugate roots** |  | ```{r}  x2=gen.arma.wge(n=75,phi=c(1.6,-.8),sn=24)  x2=x2+25  plotts.wge(x2)  ```  - n.ahead is how many forward you want to predict\*\*\*  ```{r}  fore.arma.wge(x2,phi=c(1.6,-.8),n.ahead=50,limits=FALSE)  ``` |
| **ARMA((p,q) Forecasting**  Basic Formula    Very computationally intensive. Example is For Live Assignment Unit 7 | fore.arma.wge for an ARMA (2,1) While the forecast has slightly changed, very much resembles an AR(2) | - ARMA(2,1) Example  ```{r}  arma1 = gen.arma.wge(n=75, phi = c(1.6,-.8), theta = -.9, sn=24)  fore.arma.wge(arma1, phi = c(1.6,-.8), theta = -.9, n.ahead = 20, limits = FALSE)  ```  Limits = TRUE, FALSE, creates limits in code  - ARMA(1,1) Example  ```{r}  fore.arma.wge(arma1, phi = .8, theta = -.9, n.ahead = 20, limits = FALSE)  ``` | ARMA(1,1) we also see it mimicking the AR(1) behavior while the MA(1) component has slightly changes the forecasting model. |  |
| **Probability Limits**  We are trying to make sure the actual values from the forecasts will fall within our probability limits | Any ARMA(p,q) model can be expressed as a General Linear Process    and at is normal white noise (mean 0 and variance s2a | Probability Limit for ARMA Facts | Note about Psi weights    Recall: The ratio of two polynomials is a (possibly infinite order) polynomial  Example Code:  - (1 - 1.2B + .6B2) Xt = (1 - .5B)at  - Change lag.max for number of psi's you need  ```{r}  psi.weights.wge(phi=c(1.2,-.6),theta=.5,lag.max=10)  ```  - Concept Checl 7.6  ```{r}  #Generate Example Model data  cc7.6 = gen.arma.wge(n=200, phi = c(.4,-.6,.8))  #Model Data  fore.arma.wge(cc7.6, phi = c(.4,-.6,.8))  #Grab Sigma\_hat a  fore.arma.wge(cc7.6, phi = c(.4,-.6,.8))$se  psi.weights.wge(phi = c(.4,-.6,.8),lag.max=10)  ``` | Probability Limit Calculation Example  Formula for the half-widths      Calculating Forecasts and half widths in tswge |
| **Checking our forecasts:**  Average Square Error is calculated by below formula    The lower the ASE the better the model is forecasting | Canadian Lynx Example and Code  ```{r}  data(llynx)  plotts.wge(llynx)  Lynxf\_AR4 = fore.arma.wge(llynx,phi=c(1.3, -0.7,0.1,-0.2), n.ahead=12, limits=FALSE, lastn = TRUE)  ASE\_AR4 = mean((Lynxf\_AR4$f -llynx[103:114])^2)  ASE\_AR4  ``` |  |  |  |
| **Forecasting with an ARIMA model** | Methods similar to ARMA but will only use tswge for fore.arumo.wge  Forecasts limits for non-stationary models become unbounded: because we are modeling unstationary data the forecasts become unbounded | Example ARIMA(0,1,0)        **\*\*Continues to forecast the last data value since it’s + aa that has as much of a chance of going up as it does down. Therefore, the ARIMA forecast is just the last value. Boring but accurate.\*\***  xd1=gen.aruma.wge(n=75,d=1,sn=74)  fore.aruma.wge(xd1,d=1,n.ahead=5,limits=T)  ARIMA(1,1,0)  - Example: forecasts using the ARIMA(1,1,0) model  - Arima term of 1  - AR term of .8  - Non stationary data  - Therefore will not converge to mean but increase to some point  ```{r}  x=gen.aruma.wge(n=50,phi=.8,d=1,sn=15)  fore.aruma.wge(x,phi=.8,d=1,n.ahead=20 , limits = FALSE)  ``` | Example ARIMA(0,2,)  IN STEAD of the most recent value to be use. The most recent TREND (slope) is continued in the forecast.      TO CODE THIS WE WOULD CHANGE D = 2  - Example: forecasts using the ARIMA(0,2,0) model  - No AR term  - No MA term  - 2 d terms (ARIMA) terms  ```{r}  x=gen.aruma.wge(n=50,phi=.8,d=1,sn=15)  fore.aruma.wge(x,d=2,n.ahead=20, limits = T)  ``` |  |
| **Forecasting with Seasonal ARIMA models**  Simple quarterly seasonal model | Given quarterly data, this very simple model forecasts the current quarter to be the value observed for this quarter last year.  Forecasts value are an exact replica of the last s value  Example s=12      **\*\*ONLY THE LAST SEASONAL BEHAVIOR IS FORECASTED> THERE IS NO TREND)\*\*** | ```{r}  x=gen.aruma.wge(n=20,s=4, sn = 6)  fore.aruma.wge(x,s=4,n.ahead=24,lastn=FALSE,plot=TRUE, limits=FALSE)  ```  - Using lastn in order to compare our forecast to the last 8 points of known data  ```{r}  x=gen.aruma.wge(n=20,s=4, sn = 6)  fore.aruma.wge(x,s=4,n.ahead=8,lastn=TRUE,plot=TRUE,limits=FALSE)  ```  - ARIMA forecasting with seasonal and phi (AR) component  - Looks at the lst 4 seasonal points and just continues to forcast those.  ```{r}  x=gen.aruma.wge(n=20,phi=.8,s=4,sn = 6)  fore.aruma.wge(x,phi=.8,s=4,n.ahead=24,limits=FALSE)  ``` | ARIMA w/ Seasonal and Trend Component  Seasonal Model w/ Trend = Airline Model  (1-B6)(1-B) |  |
| **Forecasting data with a deterministic trend aka data that was produced from a process with a signal plus noise model** | Forecasting Strategy Linear Case    Forecast Limits | x=gen.sigplusnoise.wge(n=50,b0=10,b1=.2, phi=c(.8,-.6))  #  xfore=fore.sigplusnoise.wge(x,linear=TRUE,n.ahead=20,lastn=F,limits=F) | Notice that the realization has a line with a cyclic noise. That cyclic noise is coming from an AR(2) with complex conjugate root. Early forecasts show this behavior. But as the data gets father away from the last known data point we see that it starts to approach the mean. However, since this is a signal plus noise model the mean is the slope of the regression line. |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Final Exam Review** |  |  |  |  |  |  |  |
| Burg Estimates |  | Cannot yield nonstationary models |  | Dicky Fuller | Ho of Dickey Fuller is the model does have a factor with a root of 1  Ho: model has a root of +1 | Ha of Dickey Fuller is the model does not have a factor with a root of 1  Ha: the model does not have a root of +1 |  |
| Yule Walker Estimates | Become much less accurate than Burg and ML estimates as phi gets close to 1. | Cannot yield nonstationary models |  | Cochrane Orcutt Test |  |  |  |
| Maximum Likelihood |  | Can yield nonstationary models |  | MLR w/ correlated errors | Have univariate responses |  |  |
| Criterion  AIC, AICC, BIC | BIC favors smaller models and penalizes accordingly |  |  | VAR models | Have multivariate responses |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Examples**  **Zero Mean Form** | **Stationarity** | **Autocorrelation Functions** | **Example Formula** | **Realization Behavior** | **Autocorrelation Behaviors (ACF)** | **Spectra**  **l Density Behaviors** | **Code** |
| AR(1) Positive (Unit 3)    Operator Notation  Characteristic Polynomial  Characteristic Equation  Means  Stationary if | Stationary if | for *k* ≥ 1 | Example 1    Example 2 | Seem to be “wandering,” aperiodic in nature.  *φ*1 > 0: *wandering* | Are damped exponentials  *φ*1 > 0: *exponential* *damping*  Closer to 1 therefore dampening slower  Farther from 1 therefore dampening faster | Have peaks at *f* = 0, which is consistent with the behavior of the realizations  *φ*1 > 0: peak at *f* = 0  closer to 1 therefore strong peak at 0  farther from 1 therefore lower peak at 0 | ```{r}  gen.arma.wge(n=100,phi=.95,sn=5)  ```  ```{r}  plotts.true.wge(phi=.95)  ``` |
| AR(1) Negative (Unit 3) | Stationary if |*φ*1| < 1 | for *k* ≥ 1 | Examples 1    Example 2 | 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.  *φ*1 < 0: *oscillating* | Are damped, oscillating exponentials. For example, for *ϕ*1 = -.95, then *r*1 = -.95, which is consistent with the behavior described above for realizations.  Closer to 1 therefore dampening slower  Farther from 1 therefore dampening faster | Have peaks at *f* = .5 (i.e. a cycle length of 2). This is consistent with the up-and-down behavior in the realizations.  closer to 1 therefore strong peak at .5  farther from 1 therefore lower peak at .5 | ```{r}  gen.arma.wge(n=100,phi=-.95,sn=5)  ```  ```{r}  plotts.true.wge(phi=-.95)  ``` |
| AR(2) /AR(p) Two Real Roots Positive | All roots are **outside** the unit circle  **|*r*|-1 < 1**  (abs recip of root)  *(\* |r|-1 > 1 is “explosive”)* | Variance |  | Seem to be “wandering,” aperiodic in nature.  *φ*1 > 0: *wandering* | Are damped exponentials  *φ*1 > 0: *exponential* *damping* | Have peaks at *f* = 0, which is consistent with the behavior of the realizations  *φ*1 > 0: peak at *f* = 0 | - Two Positive Real Roots  -Xt - .2X\_t-1 - .48X\_t-2  - Heavy oscillating behavior in ACFs, high peak at .5 due to double negative behavior driving graph behaviors. Heavy oscillating in Realization.  ```{r}  x4.1=gen.arma.wge(200,phi = c(.2,.48))  plotts.sample.wge(x4.1)  plotts.true.wge(phi = c(.2,.48))  ``` |
| AR(2) /AR(p) Two Real Roots Negative | All roots are **outside** the unit circle  **|*r*|-1 < 1**  (abs recip of root)  *(\* |r|-1 > 1 is “explosive”)* | Variance | Example Equation: | Would look just like a negative like above. | Would look just like a negative like above. | Would look just like a negative like above. | - Two Negative Roots  - Xt + 1.4X\_t-1 + .48X\_t-2  ```{r}  x4.3=gen.arma.wge(200,phi = c(-1.4,-.48))  plotts.sample.wge(x4.3)  ``` |
| AR(2) / AR(p) Real Roots One Positive & One Negative | All roots are **outside** the unit circle  **|*r*|-1 < 1**  (abs recip of root)  *(\* |r|-1 > 1 is “explosive”)* | Variance | Example Equation: | Wandering (due to  1 - .8*B*) and high-frequency behavior (due to 1 + .6*B*) | Damped exponential (due to 1 - .8*B*) with a hint of oscillatory behavior for small lags (due to 1 + .6*B*) | Have peaks at *f* = 0 and *f* = .5 because of the positive and negative traits showing. Higher at peak zero because the root for positive is close to the edge of the unit circle | - One Positive and One Negative  -Xt - 1.4X\_t-1 + .48X\_t-2  - Because the positive root is so much larger than the negative root you barely see any of the negative root freq represented in the spectral density plot. As well as ACF and Realizations  ```{r}  x4.2=gen.arma.wge(200,phi=c(1.4,-.48))  plotts.sample.wge(x4.2)  ``` |
| AR(2) Complex Conjugate Roots | All roots are **outside** the unit circle  **|*r*|-1 < 1**  (abs recip of root)  *(\* |r|-1 > 1 is “explosive”)* | Variance | Example Equation:  Characteristic Equation: | Has realizations that show a pseudo-cyclic behavior with cycle length of about 1/f0, where f0 is given  Example  The realization is pseudo-cuclic with about 7 cycles in the series of length 100. That is, the cycle length is about 100/7 = 14 | -A stationary AR(2) model whose characteristic equation has complex conjugate roots has autocorrelation w/ the appearance of damped sinusoidal curve  Example autocorrelations have a damped sinusoidal behavior with cycle length about 13 or 14, which is consistent with f0=.0738. | The sinusoidal function has frequency f0 where:    Shows peak at the system’s frequency.  Example:    We see the spectral density reflect our calculation from above with peak at around .07 | - Complex Conjugate Roots  - 1-1.6z+.8z^2 = 0  ```{r}  x4.4=gen.arma.wge(200,phi = c(1.6,-.8))  plotts.sample.wge(x4.4)  plotts.true.wge(phi = c(1.6,-.8))  ``` |
| AR(3) Model |  | For AR(p) models we use the factor table in tswge. We are able to pull the behaviors of the realization, acf, and spectral density. See factor table examples for more info: | Upon reviewing the absolute reciprocals, we can quickly see this third-order polynomial model is stationary. Since both of the above absolute reciprocals are < 1 that means the roots associated which those absolute reciprocals are outside the unit circle and therefore the model is stationary.  We choose to look at the absolute reciprocal since the complex conjugate root is not easily interpretable as >1 or outside of the unit circle for stationary judgement. | -Realization: wandering  -Realization: pseudo-cyclic behavior showing frequency of about w/ cycle length of | -ACF: slowly dampening  -ACF: Damped sinusoidal autocorrelations w/ a period (cycle length) of about 6 | -Spectral Density: peak at 0  -Spectral Density: peak at | - X\_t - 1.95X\_t-1 + 1.85X\_t-2 - .855X\_t-3 = a\_t  ```{r}  #Factor table code  factor.wge(phi = c(1.95,-1.85,.855))  ```  - Plot a realization along with true autocorrelations and spectral density  ```{r}  plotts.true.wge(phi = c(1.95,-1.85,.855))  ``` |
| MA(1) q1 Positive | All roots are outside the unit circle: always stationary | Variance | Already in zero mean form: |  | ACF shows a hard cut off at lag 1 for MA(1) as expected. With Theta positive you see a oscillating acf.  This is telling me that observations 1 unit apart are negatively correlated. Observations after 1 are not correlated. | With theta being > 0 we see a dip at 0 | ```{r}  gen.arma.wge(n=100,theta=.99,sn=5)  plotts.true.wge(theta=c(.99))  ``` |
| MA(1) q1 Negative | All roots are outside the unit circle: always stationary | Variance | Already in zero mean form: |  | ACF shows a hard drop after lag 1 for MA(1) as expected. With theta negative you see a positively rapid drop off. This is telling me observations 1 unit apart are positively correlated. Observations after 1 are not correlated. | With theta being < 0 we see a dip at .5. | ```{r}  gen.arma.wge(n=100,theta=-.99,sn=5)  plotts.true.wge(theta=c(-.99))  ``` |
| MA(2) q1 is positive and q2 is negative | All roots are outside the unit circle: always stationary | Variance | Already in zero mean form: |  | ACF shows a hard drop after lag 2 for MA(2) as expected. This is telling me that observations 1 unit apart are negatively correlated. It is also telling me observations 2 units apart are slightly positively correlated. Observations more than 2 units apart are not correlated at all. |  | ```{r}  gen.arma.wge(n=100,theta=c(.9,-.4))  plotts.true.wge(theta=c(.9,-.4))  ``` |
| ARMA(4,3) | This model is both stationary and invertible because all roots are outside the unit circle. |  | Jet Fuel Example  AIC: Akaike’s Information Criterion  Given a set of data, the AIC is used to evaluate and compare the quality of models  Given a set of models, the model with the **lowest** AIC is thought to have the most quality  Jet Fuel Example Code:  ```{r}  #AR(1)  aic.wge(jetA$Price,p = 1, q = 0)$value  #AR(2)  aic.wge(jetA$Price,p = 2, q = 0)$value  #ARMA(1,1)  aic.wge(jetA$Price,p = 1, q = 1)$value  ```  ```{r}  aic5.wge(jetA$Price)  ```  Southwest Example Code:  ```{r}  swa = read.csv("https://raw.githubusercontent.com/BivinSadler/MSDS-6373-Time-Series/master/Unit%205/SWADelay.csv", header=TRUE)  plotts.wge(swa$arr\_delay)  plotts.sample.wge(swa$arr\_delay)  aic5.wge(swa$arr\_delay)  ``` | Combo of AR and MA | Combo of AR and MA; factors w/ **|*r*|-1 closest to 1 will dominate model behaviors**  Here we see that the dominant traits will be that of the AR components by running a factor table    AR Factor Table see that the sys freq has abso recip of .975 likely the closest to the unit circle (closest to 1)    MA Factor Table see that the sys freq has abso recip of .9 not as close as the AR components | Combo of AR and MA; factors w/ **|*r*|-1 closest to 1 will dominate model behaviors** Again compare factor tables for knowing which components will dominate the Spectral Density. | ```{r}  plotts.true.wge(phi = c(.3,.9,.1,-.8075),theta = c(-.9,-.8,-.72))  ```  -AR(4) componenets    ```{r}  plotts.true.wge(phi = c(.3,.9,.1,-.8075))  # AR factors  factor.wge(c(.3,.9,.1,-.8075))  ```  - MA(3) Components    ```{r}  plotts.true.wge(theta = c(-.9,-.8,-.72))  # MA factors  factor.wge(c(-.9,-.8,-.72))  ``` |
| ARIMA(0,1,0)    Differenced to | All roots are **outside** the unit circle (AR and MA)  *\*AR and MA Must be stationary and invertable.* **(1-*B*)d is non-stationary component** | **Autocorrelation Function:**  For ARIMA models, φ1 = 1 and the variance of Xt is infinite, so the **autocorrelations are undefined** | Code for ARIMA  x = gen.arima.wge(200,phi = 0, var = 1,d = 1,sn = 31)  acf(x) | To make the ARIMA(0,1,0) model stationary, we simply need to “difference” the data. As we saw in the filtering unit, the first difference is t = *Xt* – *Xt*-1) and, by the definition of the model, is equal to white noise (*at*). Since white noise is a stationary process, we can simply take the first difference to “stationarize” these data. In other words, differencing the data will remove the (1 – *B*) so that: | Code for differencing the data  Xtilde = artrans.wge(x,1)  plotts.wge(Xtilde)  acf(Xtilde)  Below we see a realization and ACF that are consisten with an ARMA(1,1) [white noise] so we run an ACI5 function  #R Code:  Xtilde = artrans.wge(x,1)  plotts.wge(Xtilde)  acf(Xtilde) | Code All together:  x = gen.arima.wge(200,phi = 0, var = 1,d = 1,sn = 31)  artrans.wge(x,1)  aic5.wge(artans.wge(x,1)) |  |
| ARIMA(2,1,0)  This model is in factored form | All roots are **outside** the unit circle (AR and MA)  *\*AR and MA Must be stationary and invertable.* **(1-*B*)d is non-stationary component** |  |  | We observe a heavy wandering behavior the trait of an 1-B or d = 1 component. We also see cyclic behavior, attached to the AR(2) component. | The true autocorrelations staying at 1 – key indicator that we are dealing with an ARIMA 1-B component | Slowly dampening ACF with little to no indication of cyclic behavior – key 1-b CLUE to knowing the 1-B component is in the model. | - ARIMA(2,1,0)  - (1-1.5B+.8B^2)(1-B)X\_t = a\_t  ```{r}  A = gen.arima.wge(200, phi=c(1.5,-.8), var = 1, d = 1, sn=31)  acf(A)  ```  \*\*THIS MODEL IS NOT IN PROPER FORM FOR FACTOR TABLE\*\*  - Below code provides the coefficients of the model and the factor table.  ```{r}  model = mult.wge(fac1 = c(1.5,-.8), fac2 = 1)  factor.wge(model$model.coef)  ``` |
| ARIMA(2,2,1)  This model is in factored form: | All roots are **outside** the unit circle (AR and MA)  *\*AR and MA Must be stationary and invertable.* **(1-*B*)d is non-stationary component** |  |  | Diff 1 Data Realization noise    Diff 2 Data Realizations noise to examine | Diff 1 Autocorrelations    Diff 2 Autocorelations | Must difference the data \*2 | - ARIMA(2,2,1)  - (1 - 1.5B + .8B^2) (1-B)^2 X\_t = (1+.8B)a\_t  - First generate data and graph ACF  ```{r}  arima2.2 = gen.arima.wge(200,phi = c(1.5,-.8), theta = -.8, d=2,vara = 1, sn = 21)  acf(arima2.2)  ```  - Then glance at a parzen window  ```{r}  parzen.wge(arima2.2, trunc = 40)  ```  - Next find the coefficients and then breakout the factor table  - List AR in fac1, break out 2 d componenets in fac 2 & 3  - Create separate ma factor table  ```{r}  coef\_arima2.2 = mult.wge(fac1 = c(1.5,-.8), fac2 = 1, fac3 = 1)  #AR and D factor table  factor.wge(coef\_arima2.2$model.coef)  #MA factor table  factor.wge(-.8)  ```  -Next step we take out the wandering component (1-B)^2 and model the remainder of the data.  - Since there are 2 we need to diff twice  ```{r}  #Removal of first d component - differencing the data  arima2.2Dif1 = artrans.wge(arima2.2,1)  #Removal of second d component - differencing the data  arima2.2Dif2 = artrans.wge(arima2.2Dif1,1)  ```  - Next review the parzen window and see that with the nonstationary component removed, we see the peak and freq from our factor table before .0917.  ```{r}  parzen.wge(arima2.2Dif2)  ```  - Run a new AIC5 function to check the 'structure of the noise' - and see what model best fits this new stationary component of our ARIMA(2,2,1) = we would expect to see the ARMA(2,1) model left  ```{r}  aic5.wge(arima2.2Dif2)  ``` |
| **Seasonal Model Example**  **ARUMA**  Pure Seasonal Model |  | S=4 : seasonality component |  | We can see some quarterly behavior. | We can see the peaks in the ACF at 4, 8, 12, 16 that are indicative of a seasonal model with s = 4. | We see a peak at f=0, f=.25, f=.5. This is due to the factorization of the 1-B4 factor. | ```{r}  x1=gen.aruma.wge(n=80, s=4, sn = 6)  plotts.sample.wge(x1)  ``` |
| **Seasonal Component Model**  **ARUMA**  **AR(2), MA(1), Seasonal 4** |  | S=4 : seasonality component |  | We can see some quarterly behavior maybe a bit of wandering in the first part of the realization that is reflective of the AR component. | Still see dominate seasonal behavior of the ACF points of 4, 8, 12, etc. | The spectral density has altered. The AR and MA factors have affected the spectral density the most.    To evaluate further we look at the factor table of the AR and MA components.    As we can see the AR(2) component is producing a f=.1383, therefore we see a lift and at that frequency. Also the MA(1) component has a freq at f=.5 and since it is a ‘dip’ for MA we see a lowered .5 freq. | Adding phis for AR and theta for MA.  ```{r}  x2=gen.aruma.wge(n=80, phi=c(1,-.6),s=4,theta=-.5, sn=6)  plotts.sample.wge(x2)  ``` |
| **Seasonal Component Model**  **ARUMA**  **AR(2), MA(1), Seasonal 12** |  | S=12 : seasonality component |  | We can see a monthly seasonal cycle | Here we can see a full cycle completed at period 12, 24, 36… etc | We see a peak at f=0 and f=,5 and then we see a ripple behavior. This is caused by the factorization of the 1-B12 factor. | ```{r}  x3=gen.aruma.wge(n=180, phi=c(1,-.6),s=12,theta=-.5, sn=6)  plotts.sample.wge(x3,lag.max=48)  ``` |
| **Stationarize an ARUMA (Seasonal Model)** |  | ```{r}  x=gen.aruma.wge(n=80, phi = c(.4,.6,-.74), theta = c(-.7), s=12, sn = 31)  Dif = artrans.wge(x,c(rep(0,11),1)) #Take out the (1-B^12)  aic5.wge(Dif) #Check the structure of the noise  ``` |  |  |  |  |  |
| **ARUMA**  Degree 15 ARUMA polynomial |  | ```{r}  factor.wge(c(-.2,.4,.49,0,0,0,0,0,0,0,0,1,.2,-.4,-.49))  ``` | Can be factored into  (1-.88B)(1+1.08B+.55B2)(1-B12) Xt = (1+.92) at  By comparing factor tables for s=12 | And the factor table for the polynomial: | Another example:  *(.3B – .8B2 + 1B5 – .3B6 + .8B7*) *Xt* = (1 + .29*B*)*at*  *Can be factored into the below*  *(1 + .3B – .8B2)(1 – B5) Xt* = (1 + .29*B*)*at*  *Comparing the two factor tables* |  |  |
| **ARUMA**  **Airline Models**  Airline models must contain a (1-B) and (1-B12) components |  |  | All models fit for this data:  Note not an airline model missing 1-B    Below is an airline model for the data    Another Airline Model |  |  |  |  |