TV Viewership

Elise Gertsch and Spencer Wadsworth

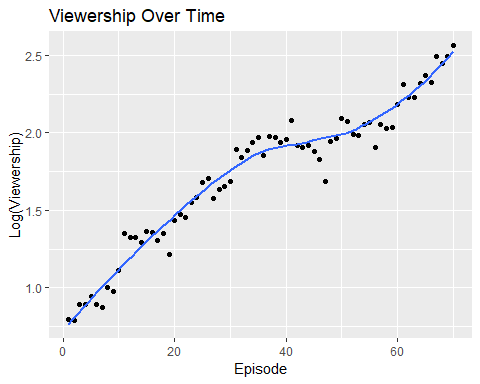
March 5, 2019

1. Because the change in viewership is highly important in determining whether to keep producing the show, use the log-transformed Viewers variable in all your analysis below. This way, the change from one show to the next corresponds to a percentage increase or decrease in viewership.

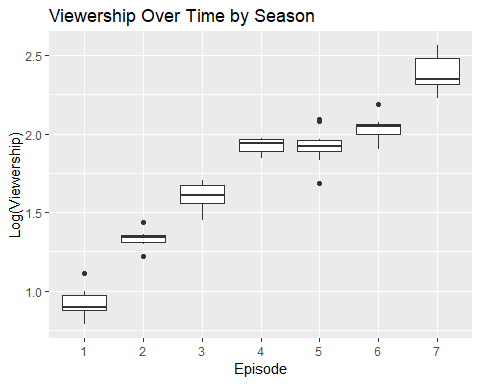
tv$Viewers <- log(tv$Viewers)

1. Create exploratory plots and calculate summary statistics from the time series. Comment on any potential relationships you see between log(Viewers) and ShowNum (note, we are using ShowNum to denote “time” in this analysis).

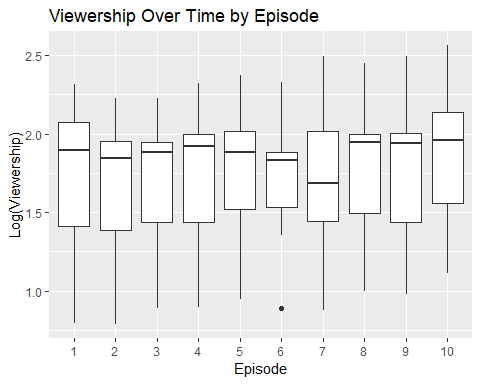
ggplot(data = tv, aes(x = ShowNum, y = Viewers)) + geom\_point() + ggtitle("Viewership Over Time") + labs(x = "Episode", y = "Log(Viewership)") + geom\_smooth(se = FALSE)



ggplot(data = tv, aes(x = Season, y = Viewers)) + geom\_boxplot() + ggtitle("Viewership Over Time by Season") + labs(x = "Episode", y = "Log(Viewership)")



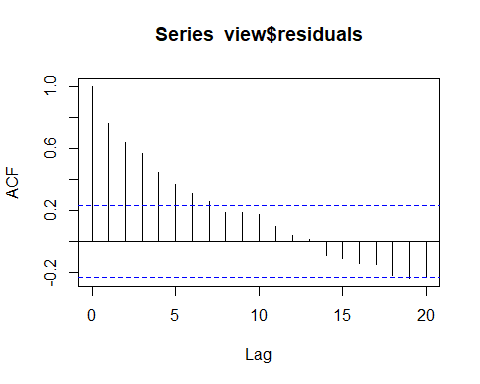
ggplot(data = tv, aes(x = Episode, y = Viewers)) + geom\_boxplot() + ggtitle("Viewership Over Time by Episode") + labs(x = "Episode", y = "Log(Viewership)")



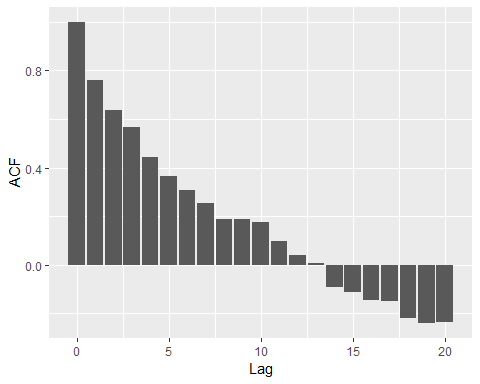
## Season Mean Log Viewership Standard Deviation Median  
## 1 1 0.918 0.096 0.894  
## 2 2 1.332 0.055 1.338  
## 3 3 1.599 0.087 1.609  
## 4 4 1.922 0.049 1.938  
## 5 5 1.922 0.116 1.919  
## 6 6 2.037 0.072 2.045  
## 7 7 2.377 0.116 2.349  
## 8 Overall 1.729 0.461 1.883

1. Fit a linear regression model to log(Viewers) using ShowNum as the explanatory variable. Determine if there is temporal correlation in the residuals which should be accounted for in your model. Discuss what this temporal correlation means for viewership.

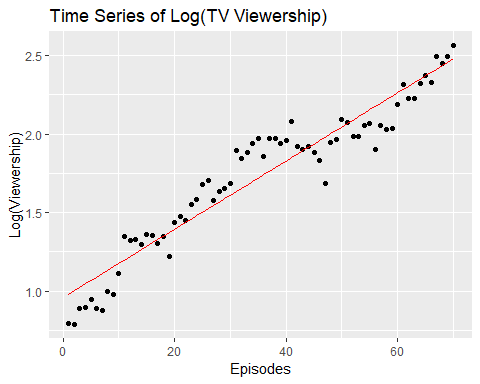
view <- lm(Viewers ~ ShowNum, data = tv)  
  
my.ACF <- acf(view$residuals, lag.max = 20)



ACF.dframe <- data.frame(Lag = my.ACF$lag, ACF = my.ACF$acf)  
ggplot(data = ACF.dframe, aes(x = Lag, y = ACF)) + geom\_col()



ggplot(data = tv, mapping = aes(x = ShowNum, y = Viewers)) + geom\_point() + labs(x = "Episodes", y = "Log(Viewership)") + ggtitle("Time Series of Log(TV Viewership)") + geom\_line(aes(x = ShowNum, y = fitted(view)), col = "red")



#The temporal correlation shown in the plot below suggests there is correlation between each succeeding episode in a season: i.e. episode 1 is closely correlated with 2 but less correlated with 3. Episode 2 is closely correlated with 3 but less with 4. Etc. This means we can expect the viewership is any immediately succeeding episode to be related to viewership in the previous episode.

1. Fixing d = 0 and D = 1, determine appropriate values of p, q, P, Q in your time series model (note you should be able to figure out the seasonal cycle value S). Only consider p{0,1,2}, q{0,1,2}, P{0,1} and Q{0,1}. Discuss how you came to choose your specific values.

X <- matrix(tv$ShowNum, ncol=1)  
ts\_models <- c()  
for (i in 0:2) {  
 for (j in 0:2) {  
 for (k in 0:1) {  
 for (l in 0:1) {  
 ts\_models <- rbind(ts\_models, c(i, 0, j, k, 1, l))  
 }  
 }  
 }  
}  
  
  
AIC.vals <- rep(NA, nrow(ts\_models))  
  
for (m in 1:nrow(ts\_models)) {  
 my.mod <- sarima(log(tv$Viewers), p = ts\_models[m, 1],  
 d = ts\_models[m, 2],  
 q = ts\_models[m, 3],   
 P = ts\_models[m, 4],   
 D = ts\_models[m, 5],  
 Q = ts\_models[m, 6],  
 S = 10,   
 xreg = X, details = FALSE)  
 AIC.vals[m] <- my.mod$AIC  
}  
  
ts\_models[which.min(AIC.vals),]

## [1] 2 0 0 0 1 1

#The model printed here is our chosen SARIMA model. This was chosen by creating 36 different models with using each of the specified values and checking the AIC values for each of the models. The model with the lowest AIC is the one that claims our choice.

1. Write down your selected time series regression model in terms of population parameters including your specification for the time series component of the residuals. Explain the meaning of any parameters in your model (including the time series components). Explain how statistical inference for your model can be used to predict the viewership moving forward.

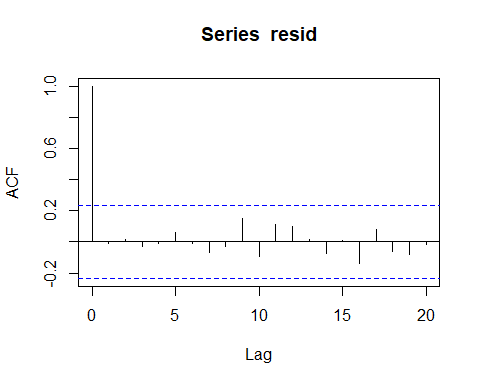
# 2 0 0 0 1 1  
#p = 2 shows decently high short term correlation, or there is high correlation between viewership in neighboring episodes. But since it is slowly decaying, there continues to be correlation even with non neighboring episodes.  
#D = 1 specifies that there exists a change from season 1 to season 2  
#Q = 1 shows seasonal correlation, or there is correlation between corresponding episodes of neighboring seasons of the series, though it decays quickly so season 3 does not have a close correlation with season 1.

1. Fit your chosen time series model and validate any model assumptions you used.

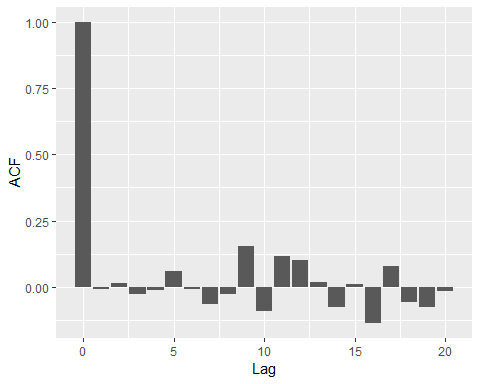
m <- which.min(AIC.vals)  
ts <- sarima(tv$Viewers, p = ts\_models[m, 1],  
 d = ts\_models[m, 2],  
 q = ts\_models[m, 3],  
 P = ts\_models[m, 4],   
 D = ts\_models[m, 5],  
 Q = ts\_models[m, 6],  
 S = 10,   
 xreg = X, details = FALSE)   
# view coefficients  
ts$ttable

## Estimate SE t.value p.value  
## ar1 0.6406 0.1236 5.1816 0.0000  
## ar2 0.2856 0.1251 2.2820 0.0263  
## sma1 -0.7828 0.2268 -3.4506 0.0011  
## xreg 0.0254 0.0043 5.9171 0.0000

resid <- resid(ts$fit)  
coef <- ts$ttable[4, 1]  
fitted <- X%\*%coef  
  
# ACF  
my.ACF <- acf(resid, lag.max = 20)

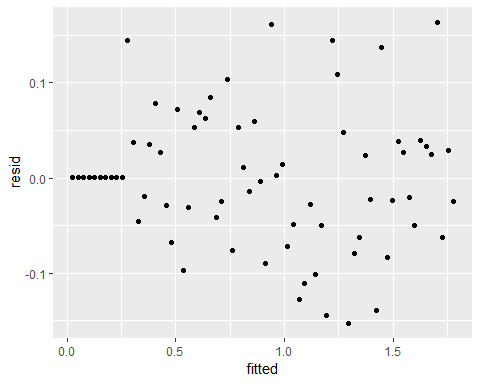


ACF.dframe <- data.frame(Lag = my.ACF$lag, ACF = my.ACF$acf)  
ggplot(data = ACF.dframe, aes(x = Lag, y = ACF)) + geom\_col()



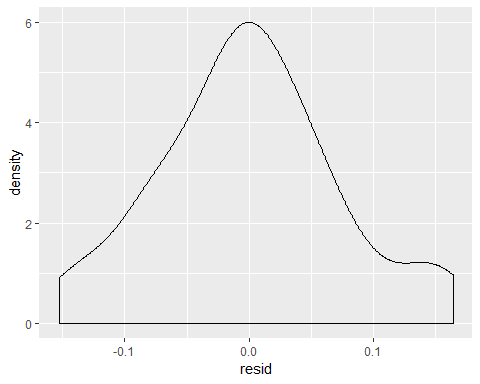
# the ACF plot shows that our data is not autocorrelated and is therefore independent  
  
# fitted v residuals  
ggplot(data = tv, aes(x = fitted, y = resid)) + geom\_point()

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



# the residuals are evenly distributed around 0  
  
# histogram of residuals  
ggplot(data = tv, aes(x = resid)) + geom\_density()

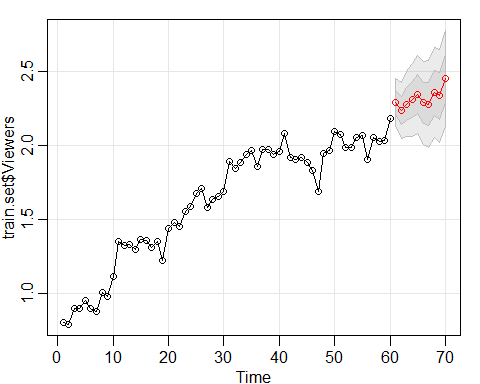
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



# the density curve is approximately normal

1. Perform a cross-validation of predictions generated from your model for the most recent season of shows. Report the quality of your predictions in terms of RPMSE. Provide a plot of your predictions along with observed viewership and 95% prediction interval limits.

train.set <- tv[1:(nrow(tv) - 10),]  
test.set <- tv[-(1:(nrow(tv) - 10)),]  
X.train <- X[1:(nrow(tv) - 10),]  
X.test <- X[-(1:(nrow(tv) - 10)),]   
  
forecast <- sarima.for(train.set$Viewers, p = 2, d = 0, q = 0, P = 0, D = 1, Q = 1, S = 10,  
 xreg = X.train, n.ahead = 10, newxreg = X.test)



# rpmse  
rpmse <- (forecast$pred - test.set$Viewers)^2 %>% mean() %>% sqrt()  
rpmse

## [1] 0.09846837

# CI  
low <- forecast$pred - qt(0.975, df = nrow(train.set) - 4)\*forecast$se  
up <- forecast$pred + qt(0.975, df = nrow(train.set) - 4)\*forecast$se  
cvg <- ((test.set$Viewers > low) & (test.set$Viewers < up)) %>% mean()  
cvg

## [1] 1

#The coverage seems a little too good here. We tried predicting the last 20 observations instead of just the last 10 and got a coverage of .7. Perhaps it's just the low sample size for validation and the luck of the draw that resulted in such good prediction intervals.

1. Determine if viewership is increasing or decreasing. Support your conclusions with appropriate hypothesis tests and confidence intervals.

#According to our graph above, viewership appears to be increasing. We will verify with a hypothesis test and confidence interval.  
  
ts$ttable[4,4]

## [1] 0

#The p-value of 0 suggests viewership is increasing.  
  
ts$ttable[4,1] + (c(-1, 1)\*qt(0.975, df=nrow(tv) - 4)\*ts$ttable[4,2])

## [1] 0.01681477 0.03398523

#The 95% confidence interval of 0.0168, 0.0340 has no negative values within it, so   
#we may conclude viewership is increasing.

1. Season 8 is already in production. Forecast the log(Viewers) forward for season 8. Comment on how executives would be able to use these forecasts to gauge if the show should continue into a ninth season.

pred\_view <- 71:80  
my\_for <- sarima.for(tv$Viewers, xreg=X,  
 p=2, d=0, q=0, P=0, D=1, Q=1, S=10,  
 n.ahead=10, newxreg=pred\_view)

