Item Demand Data Analysis

Tetsuya Chau

3/3/2022

Problem Background

Ensuring that supply equals demand is crucial for any business. Businesses want to ensure their product is available for their customers when the customers want it. The dataset ItemDemand.csv contains the total sales by month for a specific item for 5 consecutive years. The goal of this analysis is to use historical data to project the next years sales

Variable Name and Description

*year - Year

*month - Month

*sales - Total number of item sales

To analyze this dataset, do the following:

Analysis Questions:

0. Similar to what we did for the climate data, create a more continuous measure of time by defining YrMon so we can make plots easier.
1. Create exploratory plots and calculate summary statistics from the time series. Comment on any potential relationships you see between sales and YrMon.
2. Fit a linear regression model to sales using YrMon 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 sales.
3. Determine appropriate values of p, d, q, P, D, Q in your time series model (note you should be able to figure out the seasonal cycle value S). Only consider $p \in \{0,1,2\}$, $d \in \{0,1\}$, $q \in \{0,1,2\}$, $P \in \{0,1\}$, $D \in \{0,1\}$ and $Q \in \{0,1\}$. Discuss how you came to choose your specific values.
4. 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.
5. Fit your chosen time series model and validate any model assumptions you used.
6. Perform a cross-validation of predictions generated from your model for the most recent year of sales. Report the quality of your predictions in terms of RPMSE. Provide a plot of your predictions along with observed sales and 95% prediction interval limits.
7. Forecast the sales forward for 2022. Comment on how executives would be able to use these forecasts to gauge how much of the product is needed.

```
#install.packages("ggplot2")
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
item <- read.csv("/cloud/project/ItemDemand.csv")
   #View(ItemDemand)
```

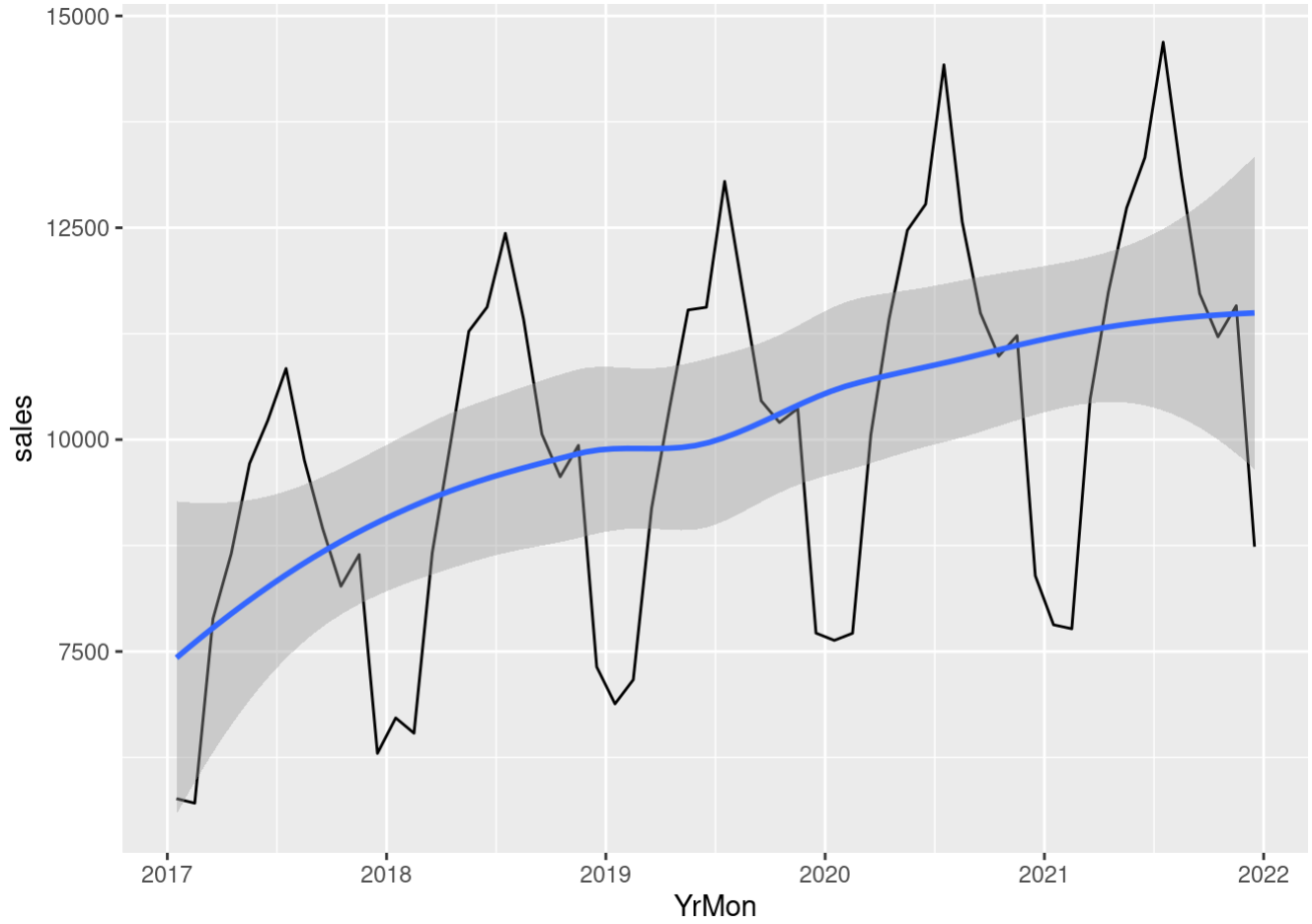
Step 0 - Similar to what we did for the climate data, create a more continuous measure of time by defining YrMon so we can make plots easier.

```
item$YrMon <- item$year + (item$month - 0.5)/12
```

Step 1 - Create exploratory plots and calculate summary statistics from the time series. Comment on any potential relationships you see between sales and YrMon.

```
ggplot(data=item, mapping = aes(x=YrMon, y=sales))+
 geom_line()+geom_smooth()
```

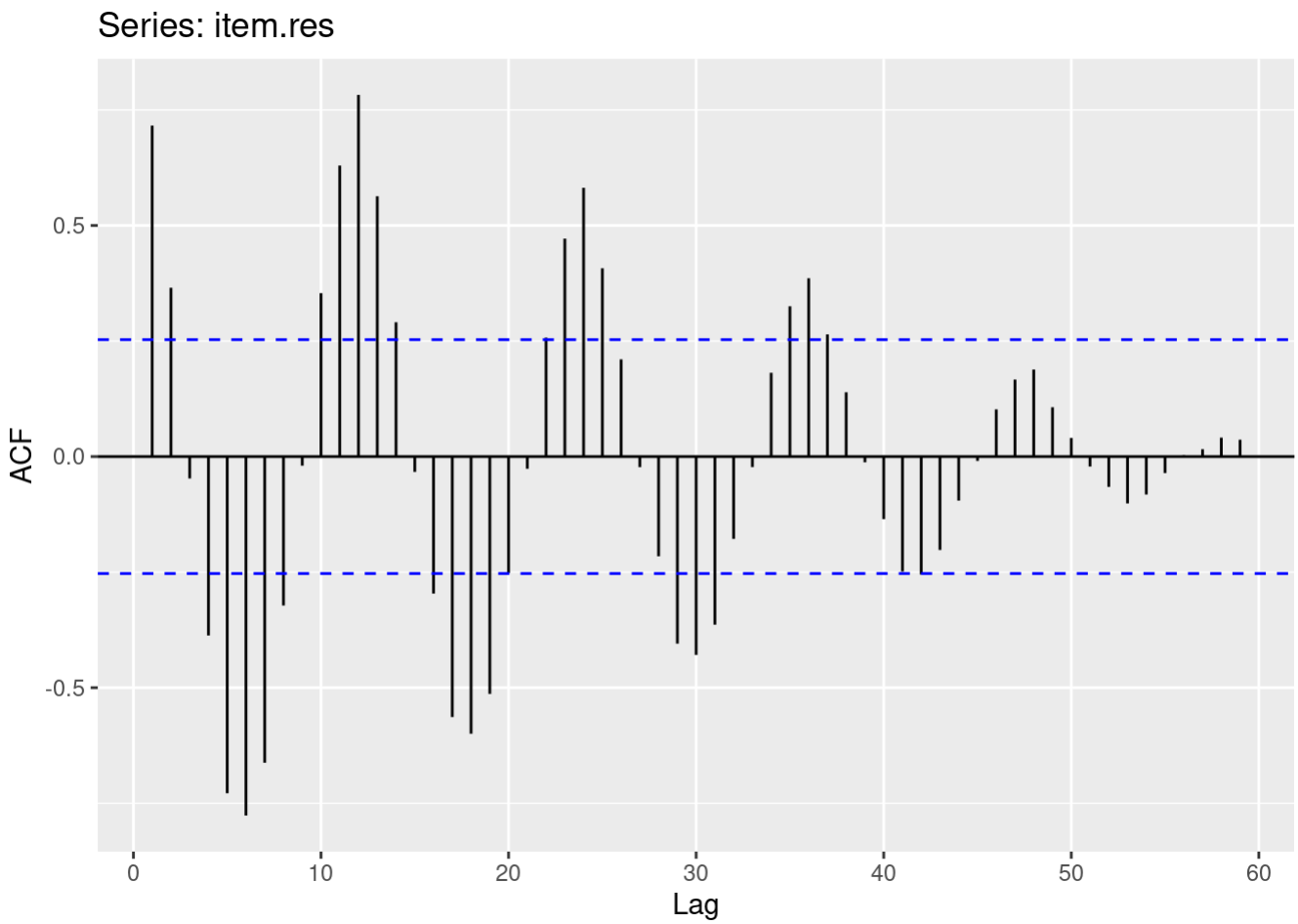
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Sales increase over the years and summer season seems to be the season when sales is the highest.

Step 2 - Fit a linear regression model to sales using YrMon 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 sales.

```
item.lm <- lm(sales~YrMon, data=item)

item.res <- resid(object=item.lm, type="pearson")

ggAcf(item.res, lag.max=nrow(item))
```



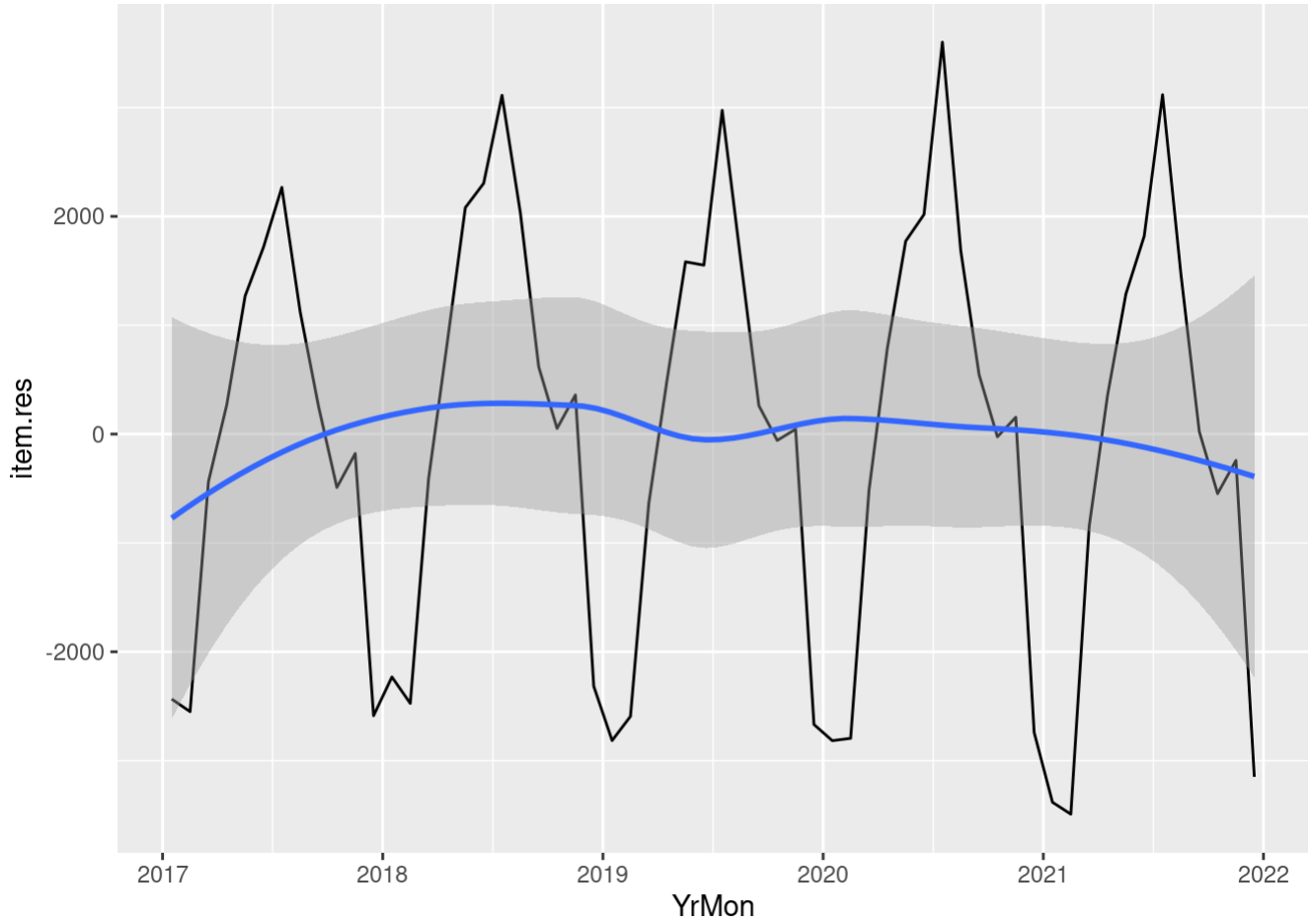
There is a temporal correlation because the month sales for each year increase on average.

There is a correlation in the lags.

The correlation between the months look like they decrease over the years on average.

```
ggplot(data=item, mapping = aes(x=YrMon, y=item.res))+
 geom_line()+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Step 3 - Determine appropriate values of p, d, q, P, D, 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}, d∈{0,1}, q∈{0,1,2}, P∈{0,1}, D∈{0,1} and Q∈{0,1}. Discuss how you came to choose your specific values.

```
my.ts1 <- ts(data=item$sales, frequency=12, start = c(1, 1))

X1 <- model.matrix(item.res~-1+YrMon, data=item)

auto.arima(my.ts1, stepwise=FALSE, max.p=2, max.q=2, max.P=1, max.Q=1, max.d=1, max.D=1, ic="ai
c", xreg = X1)
```

```
## Series: my.ts1
## Regression with ARIMA(0,0,2) errors
##
## Coefficients:
##          ma1     ma2   intercept     YrMon
##       1.2949  0.8265  -1335622.8  666.2777
## s.e.  0.1244  0.1305    585642.8  289.9939
##
## sigma^2 = 1267012:  log likelihood = -506.13
## AIC=1022.26   AICc=1023.37   BIC=1032.73
```

I found these specific values by taking the max of the numbers in each set of numbers.

It checks which combinations have the lowest AIC/BIC scores and generates the best ARIMA values.

Step 4 - 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.

y = XB + e

e ~ SARIMA(p,d,q,P,D,Q)s

-y is the response variable which is the sales

-X is the year-month

-B is a vector of the beta coefficient which is the coefficient of the year-month

-e is error ~ SARIMA(p,d,q,P,D,Q)s

-p is the autoregressive order (0)

-d is the differencing order in time (0)

-q is the moving average order (2)

-P is the seasonal autoregressive order (0)

-D is the seasonal differencing order (0)

-Q is the seasonal moving average order (0)

-s is the season (12)

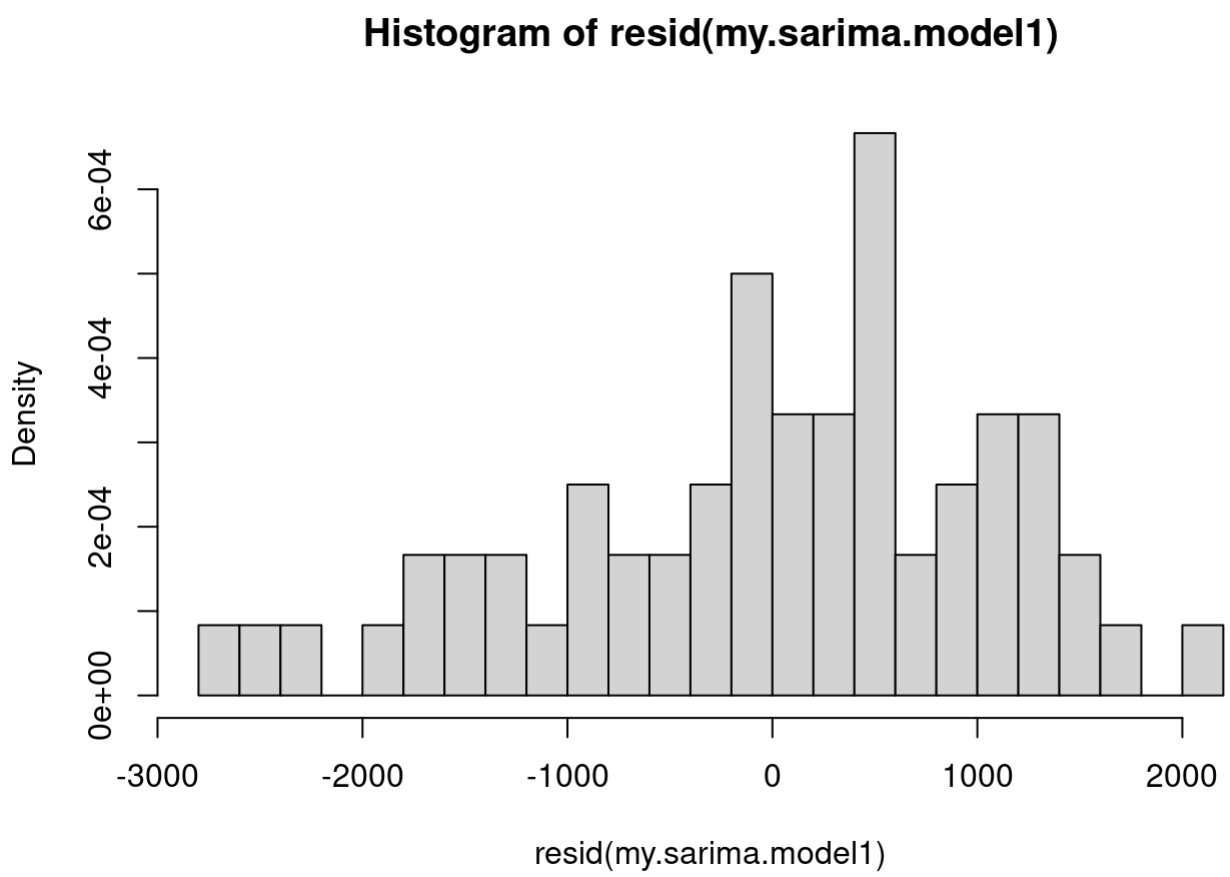
This model can be used to make an inference of the sales based on the year and the month.

Step 5 - Fit your chosen time series model and validate any model assumptions you used.

```
my.sarima.model1 <- Arima(my.ts1, order=c(0,0,2),seasonal=c(0,0,0), xreg=X1)
coef(my.sarima.model1)
```

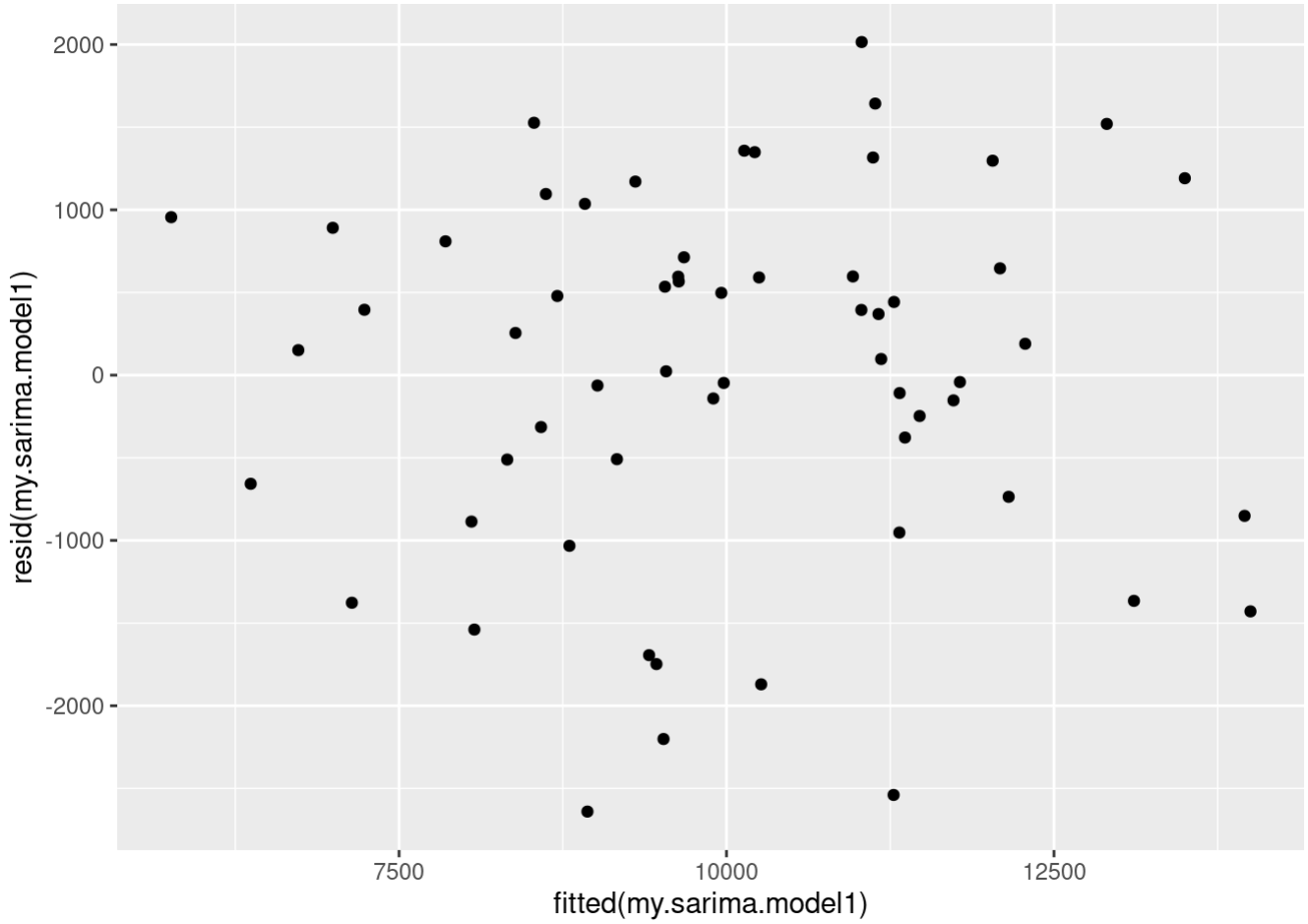
```
##           ma1           ma2     intercept         YrMon
##  1.294927e+00  8.264610e-01 -1.335623e+06  6.662777e+02
```

```
hist(resid(my.sarima.model1),freq = FALSE,breaks = 20)
curve(dnorm,from = -3,to = 3,col = "cornflowerblue",lwd = 2,
lty = 2,add = TRUE)
```

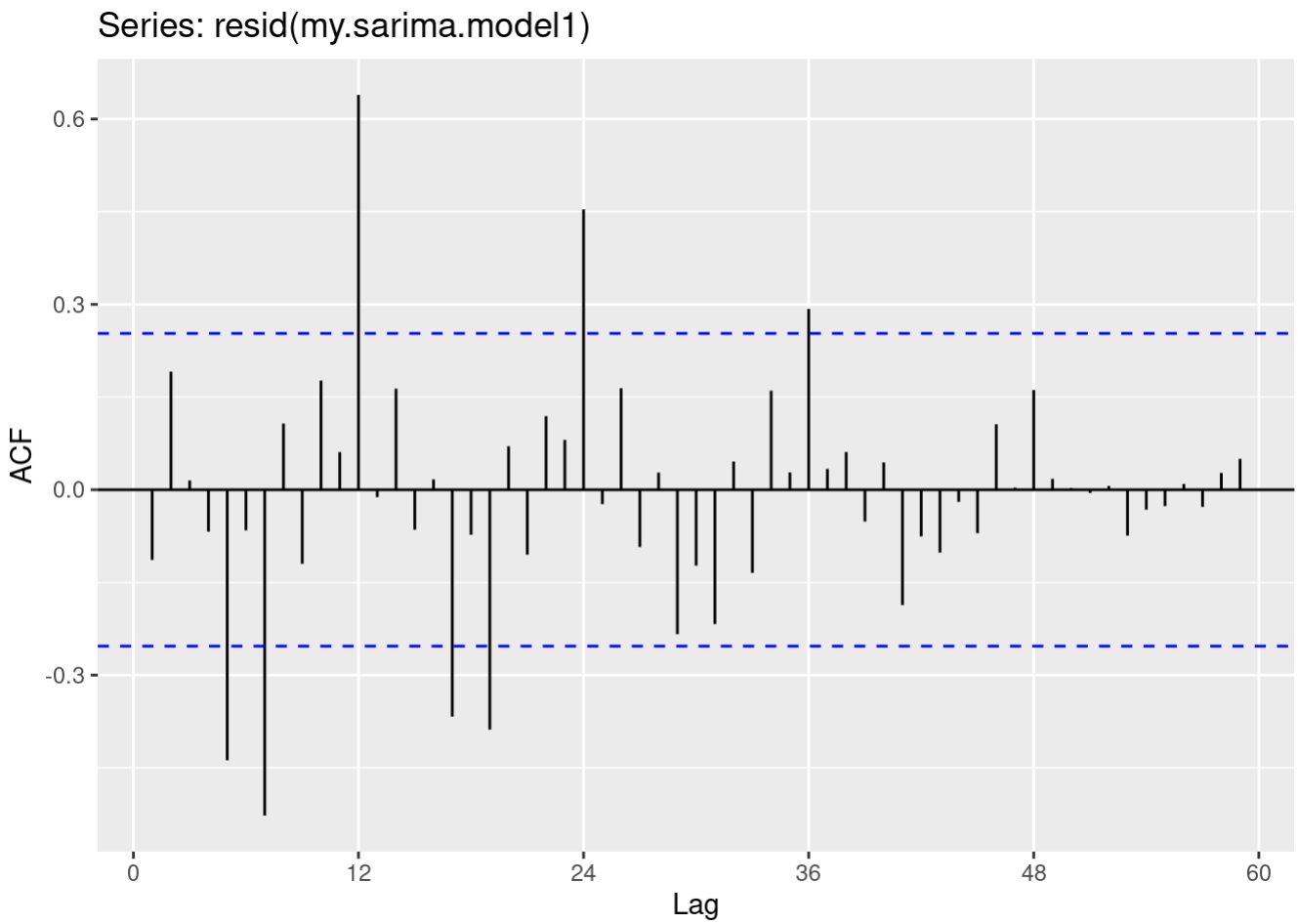



```
ggplot(data=item, mapping = aes(x=fitted(my.sarima.model1), y=resid(my.sarima.model1)))+
 geom_point()
```

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




```
ggAcf(resid(my.sarima.model1), lag.max=nrow(item))
```



Normality - The histogram of the residuals is slightly left skewed but overall, we can say that it meets the normality assumption.

Equal Variance - The scatter plot of the fitted values vs residuals look like the data points are scattered all over so we can say that it meets the equal variance assumption.

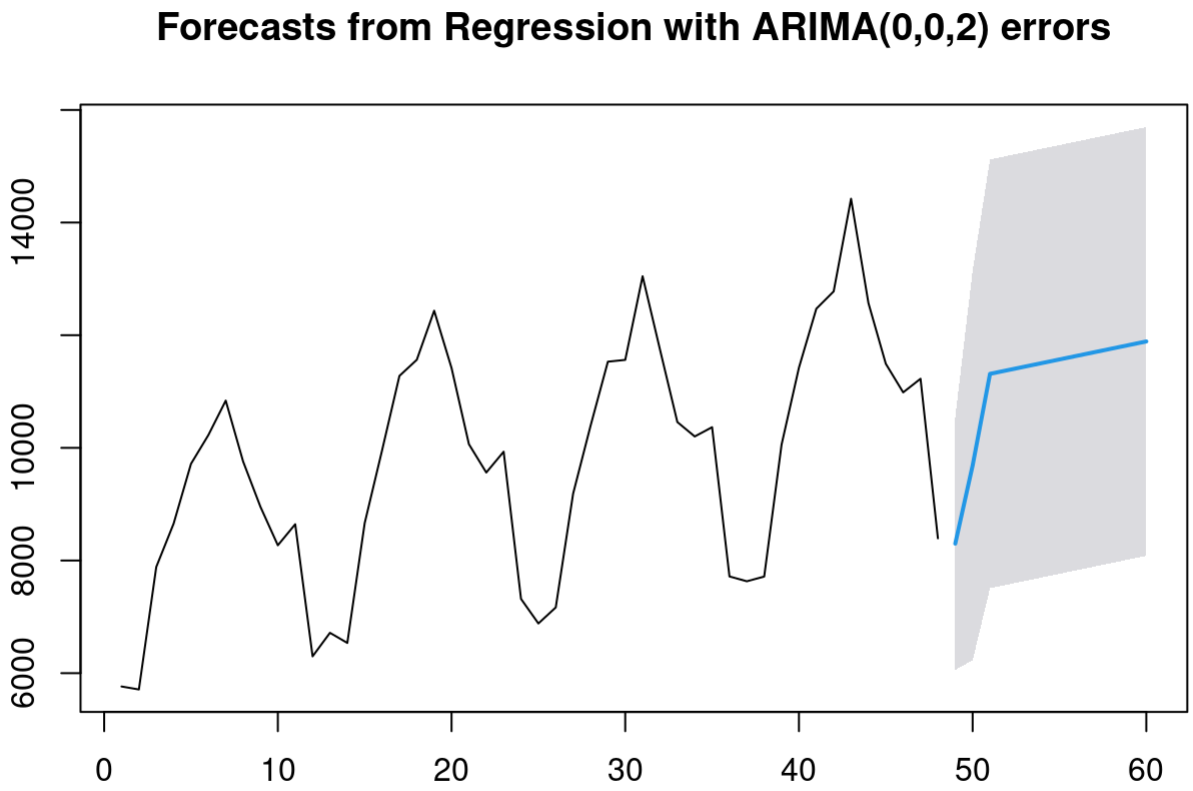
Independence - We don’t assume independence in a time-series data analysis but we do take this into account.

Step 6 - Perform a cross-validation of predictions generated from your model for the most recent year of sales. Report the quality of your predictions in terms of RPMSE. Provide a plot of your predictions along with observed sales and 95% prediction interval limits.

```
n <- 60 #Number of CV studies to run
## Select test observations
#test.obs <- sample(x=1:n, size=n.test)
## Split into test and training sets

testx1.set <- X1[(n-11):n,]
trainx1.set <- X1[1:(n-12),]

test1.set <- item[(n-11):n,]
train1.set <- item[1:(n-12),]

train.model1 <- Arima(my.ts1[1:(n-12)], order=c(0,0,2), seasonal=c(0,0,0), xreg=trainx1.set)

mean <- forecast(train.model1, h=12, xreg=testx1.set, level=0.95)$mean
lwr <- forecast(train.model1, h=12, xreg=testx1.set, level=0.95)$lower
upr <- forecast(train.model1, h=12, xreg=testx1.set, level=0.95)$upper

rpmse <- (test1.set[['sales']]-mean)^2 %>% mean() %>% sqrt()

cvg <- ((test1.set[['sales']] > lwr) & (test1.set[['sales']] < upr)) %>% mean()


print(rpmse)
```

```
## [1] 1631.646
```

```
plot0 <- forecast(train.model1, h=60,xreg=testx1.set,level=.95)
plot(plot0)
```



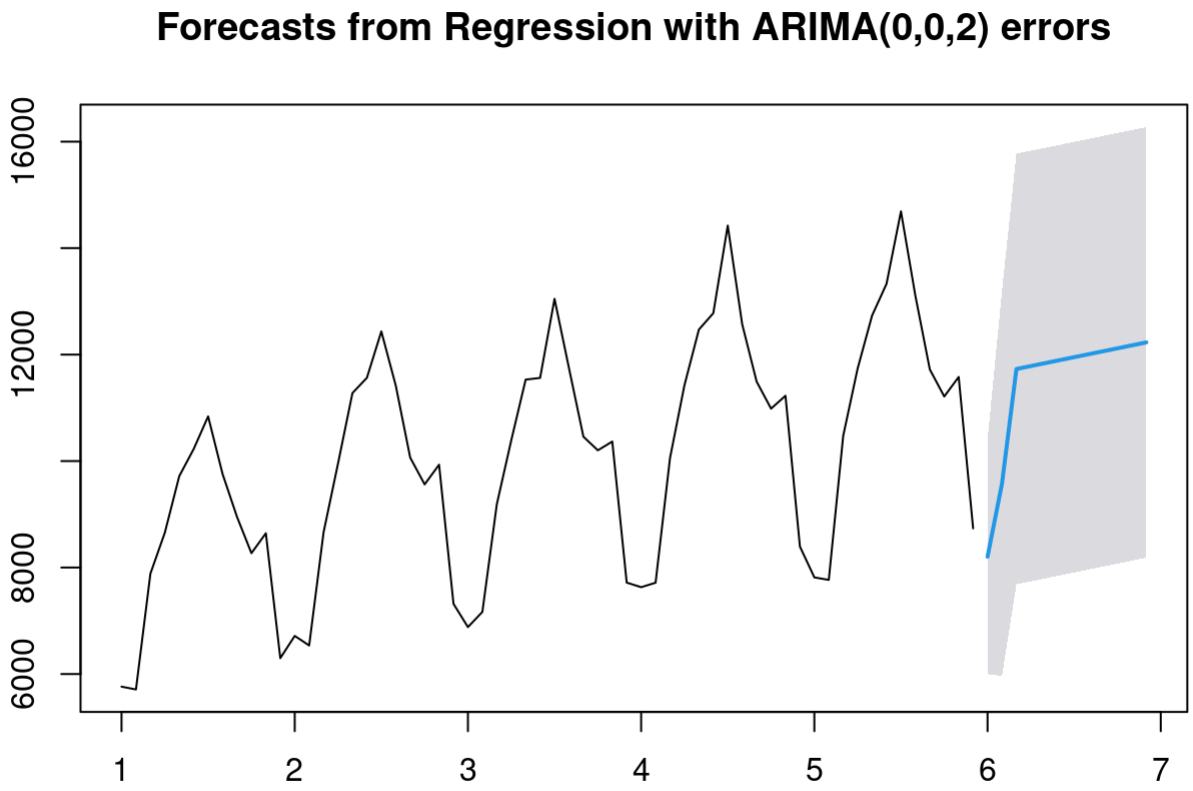
The RPMSE of 1631.646 is smaller than the standard deviation so we can say that the model fits the data well.

Step 7 - Forecast the sales forward for 2022. Comment on how executives would be able to use these forecasts to gauge how much of the product is needed.

```
future_dates1 <- max(item$YrMon) + seq(1/12,1,by=1/12)
X_future1 <- cbind(future_dates1)
names(X_future1) <- names(X1)
preds <- forecast(my.sarima.model1, h=12*1,xreg=X_future1,level=0.95)
```

```
## Warning in forecast.forecast_ARIMA(my.sarima.model1, h = 12 * 1, xreg =
## X_future1, : xreg contains different column names from the xreg used in
## training. Please check that the regressors are in the same order.
```

```
plot(preds)
```



We can use this forecast graph to tell the executives how many items will be in demand in a particular year and month based on previous data. By doing so, we can inform them about how many items they will need to produce in order to meet those predicted demands.