Econometrics Project 1

John Macke, Pujan Thakrar, Mark Vandre

January 17, 2019

Part I Introduction (describe the data, provide some background on the topic, etc.).

The data we are using is retail sales from hardware stores in the US. The data is reported in millions of dollars and is not seasonally adjusted. The frequency of the data is monthly and goes from January 1992 to October 2018.

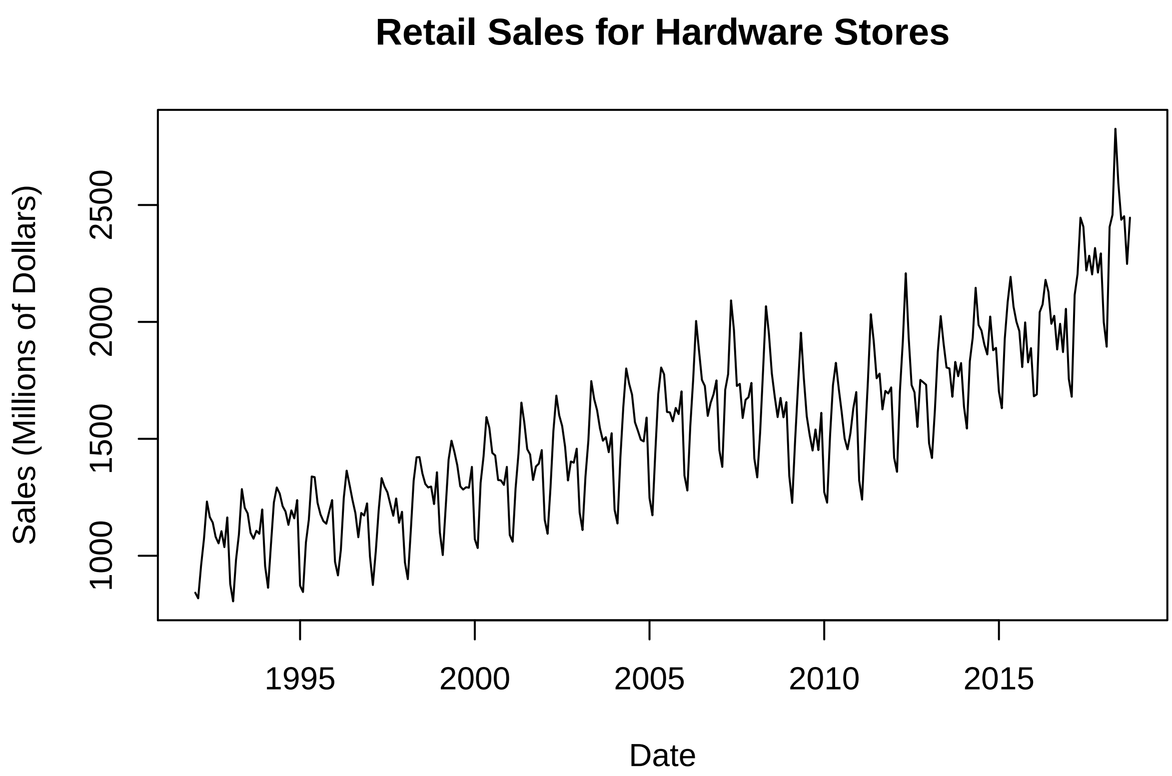
Hardware sales have an increasing trend over time and displays seasonality. This seasonality is expected as we would think that hardware sales typically peak in summer months due to more outdoor projects and construction.

Part II Results (answers and plots). Consists of three parts:

1. Modeling and Forecasting Trend

(a) Show a time-series plot of your data.

```
hardware = read.csv("hardware.csv")                           #load data

hw = ts(hardware,start=1992,frequency = 12)             #create time series
hw_ts = hw[,-1]                                         #remove date column


plot(hw_ts,ylab="Sales (Millions of Dollars)",xlab="Date",      #plot
     main="Retail Sales for Hardware Stores")
```

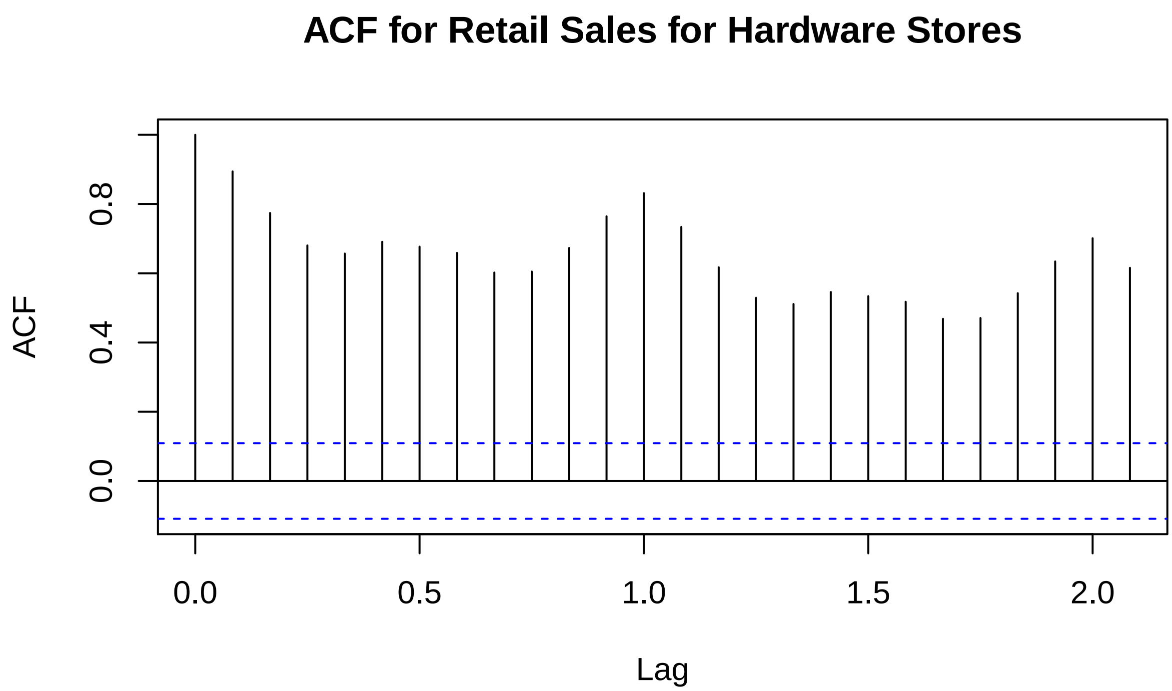


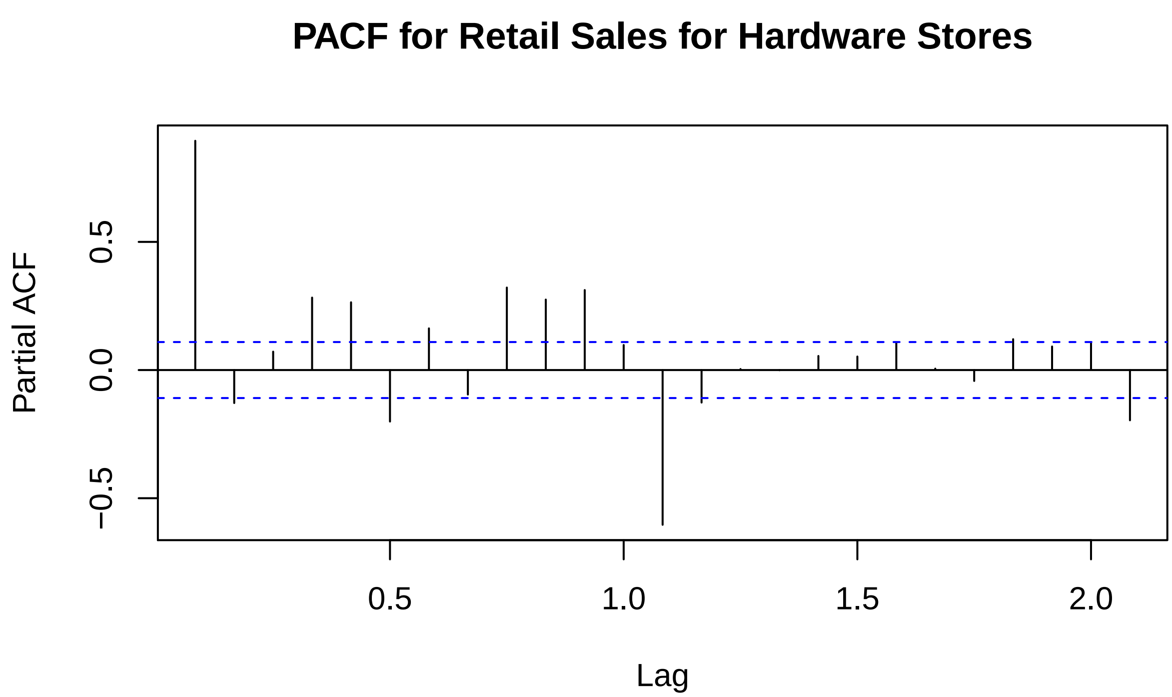
(b) Does your plot in (a) suggest that the data are covariance stationary? Explain your answer.

The plot in (a) suggests that the data are not covariance stationary. The increasing trend of the data suggests that the mean is increasing over time. The variance also appears to be increasing over time. Therefore this cannot be a covariance stationary dataset.

(c) Plot and discuss the ACF and PACF of your data.

```
par(mfrow = c(2,1))
acf(hw_ts, main="ACF for Retail Sales for Hardware Stores")          #ACF
pacf(hw_ts, main="PACF for Retail Sales for Hardware Stores")        #PACF
```





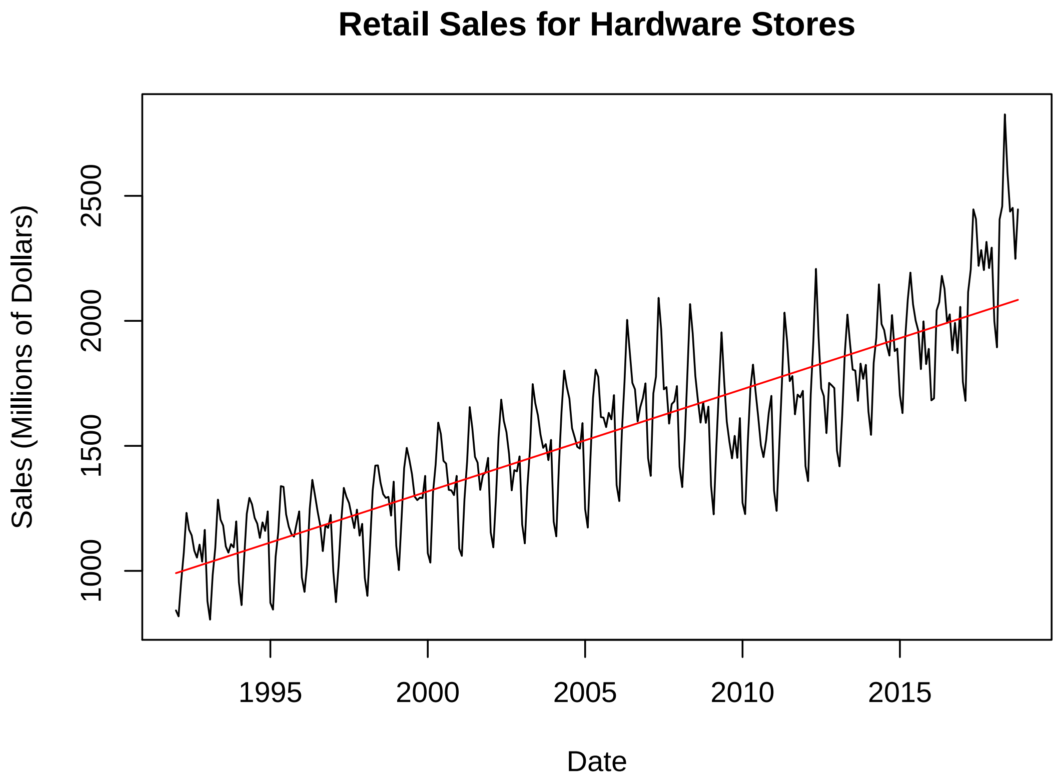
Both the ACF and PACF suggest a high dependence on how retail sales in hardware stores have changed over time. More specifically, retail sales in period t are highly dependent on retail sales from previous periods.

When the data has a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So the ACF of trended time series tend to have positive values that slowly decrease as the lags increase. Our data has a clear increasing trend over time, which is captured in the ACF plot with positive autocorrelations that are decreasing as the lag increases.

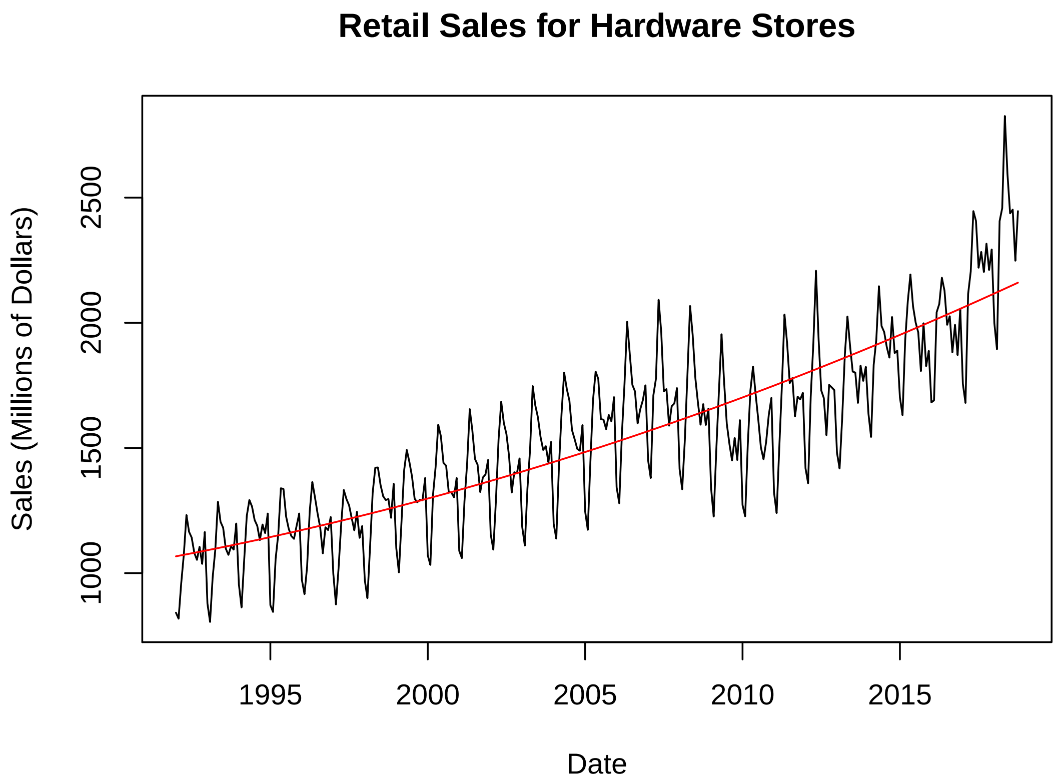
When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags. Our data has a clear seasonal component, which is captured in the ACF plot as the autocorrelations are increasing and decreasing at regular intervals.

(d) Fit a linear and nonlinear (e.g., polynomial, exponential, quadratic + periodic, etc.) model to your series. In one window, show both figures of the original time series plot with the respective fits.

```
t = seq(1992,2018.750,length=length(hw_ts))            #create time dummy variable
par(mfrow=c(2,1))

m1 = lm(hw_ts~t)                                     #model with linear fit
m2 = lm(hw_ts~t+I(t^2))                              #model with quadratic fit


plot(hw_ts,ylab="Sales (Millions of Dollars)", xlab = "Date",  #plot linear fit
     main="Retail Sales for Hardware Stores")
lines(t,m1$fitted.values, col = "red")
plot(hw_ts,ylab="Sales (Millions of Dollars)", xlab = "Date",  #plot quadratic fit
     main="Retail Sales for Hardware Stores")
lines(t,m2$fitted.values, col = "red")
```



Retail Sales for Hardware Stores
Sales (Millions of Dollars)
1000
1500
2000
2500
1995
2000
2005
2010
2015
Date

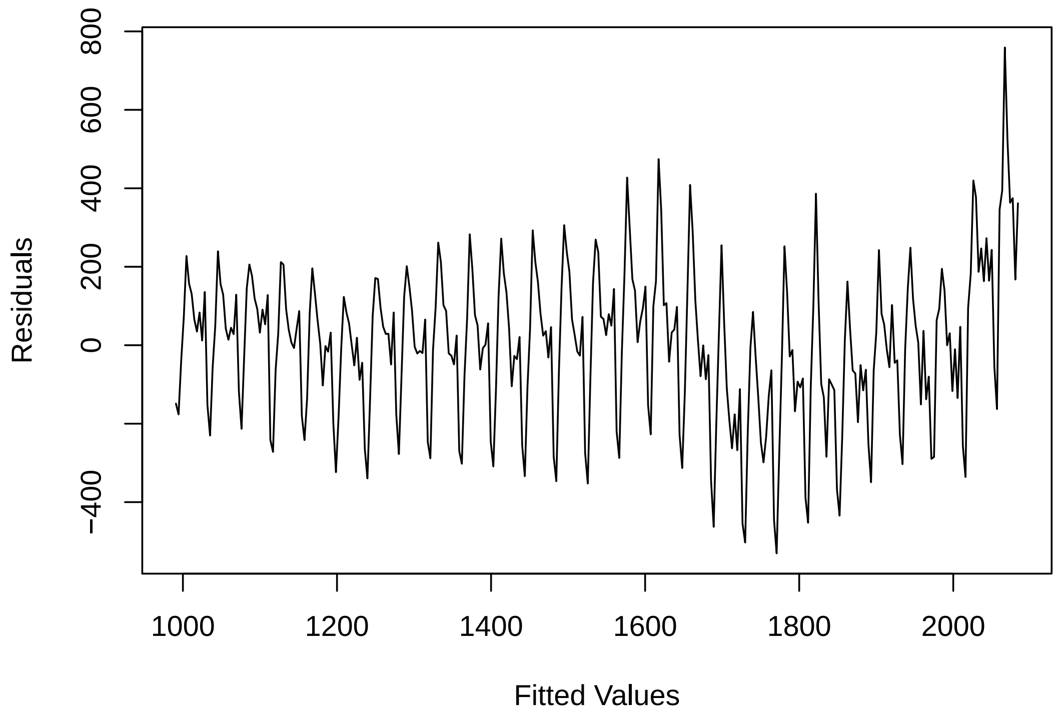


Retail Sales for Hardware Stores
Sales (Millions of Dollars)
1000
1500
2000
2500
1995
2000
2005
2010
2015
Date

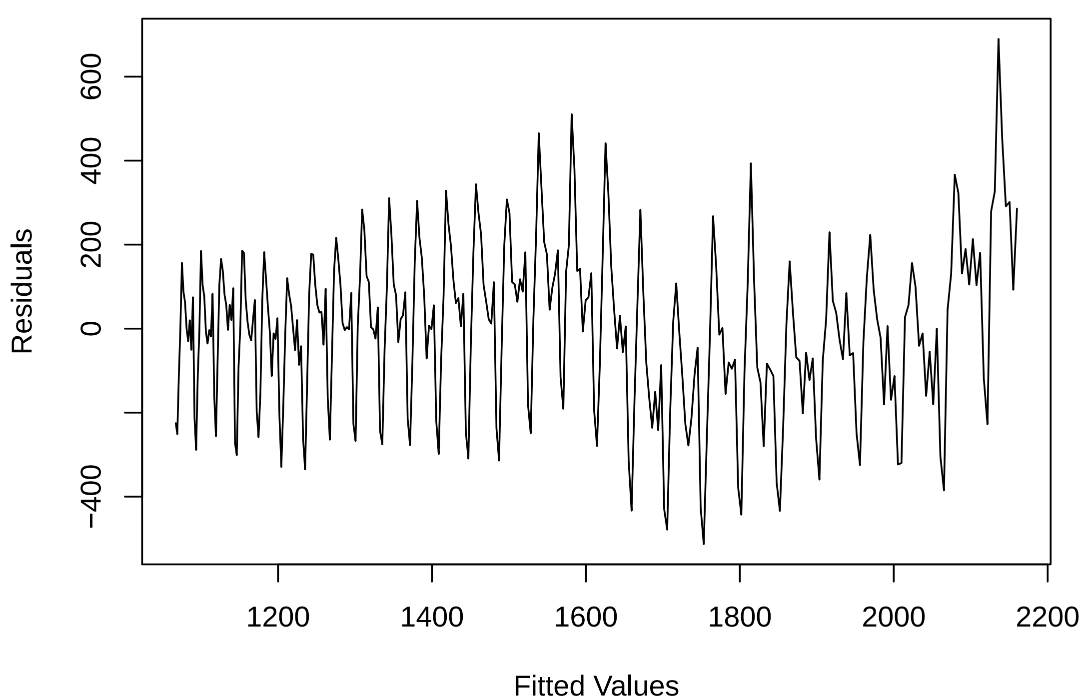
(e) For each model, plot the respective residuals vs. fitted values and discuss your observations.

```r
par(mfrow=c(2,1))

#plot residuals vs. fitted values with linear  fit
plot(m1$fitted.values,m1$residuals,type='l', xlab="Fitted Values", ylab = "Residuals",
     main = "Residuals vs. Fitted Values with Linear Fit")

#plot residuals vs. fitted values with quadratic fit
plot(m2$fitted.values,m2$residuals,type='l', xlab="Fitted Values", ylab = "Residuals",
     main = "Residuals vs. Fitted Values with Quadratic Fit")
```

Residuals vs. Fitted Values with Linear Fit



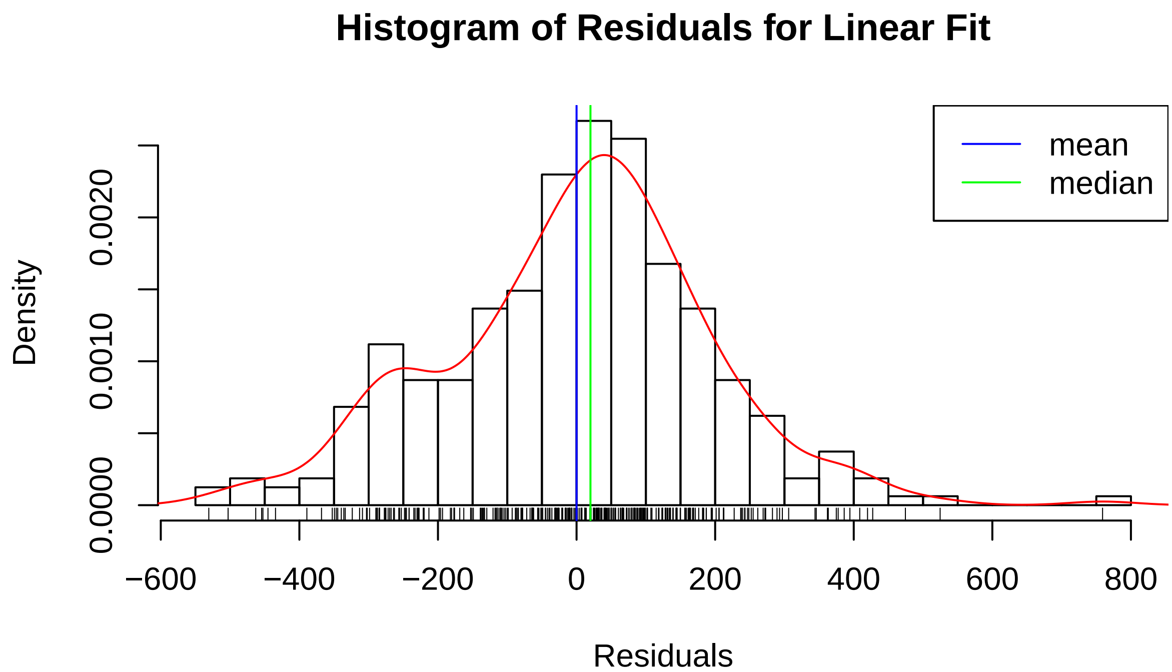
Residuals vs. Fitted Values with Quadratic Fit

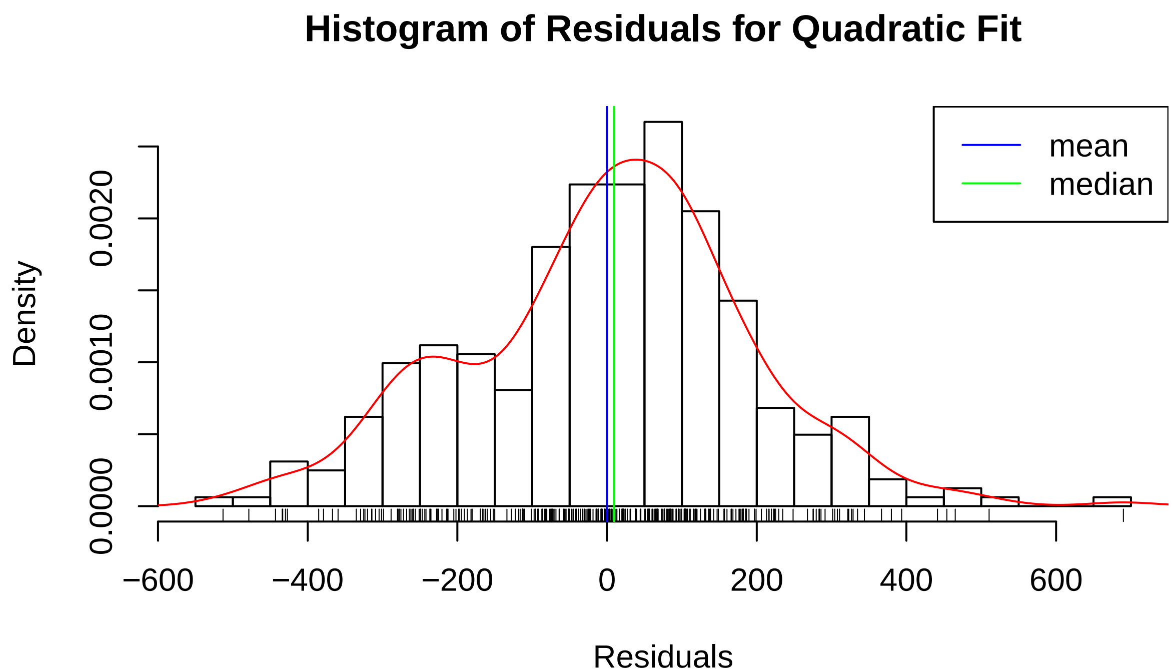


We notice that there is still some structure/pattern remaining in the residuals. We can incorporate this structure into the model to improve the fit. Adding seasonality would likely help make the residuals more random (i.e. white noise). After applying the model fits, the residuals no longer have an increasing trend and the mean is more constant over time.

(f) For each model, plot a histogram of the residuals and discuss your results.

```
par(mfrow=c(2,1))

#Plot histogram for linear fit
hist(m1$residuals,breaks = "FD",xlab="Residuals",
     main="Histogram of Residuals for Linear Fit",probability = TRUE)
lines(density(m1$residuals),col="red")
rug(m1$residuals)
abline(v=mean(m1$residuals),col="blue")
abline(v=median(m1$residuals),col="green")
legend("topright",c("mean","median"),col=c("blue","green"),lty=c(1,1))

#Plot histogram for quadratic fit
hist(m2$residuals,breaks = "FD",xlab="Residuals",
     main="Histogram of Residuals for Quadratic Fit",probability = TRUE)
lines(density(m2$residuals),col="red")
rug(m2$residuals)
abline(v=mean(m2$residuals),col="blue")
abline(v=median(m2$residuals),col="green")
legend("topright",c("mean","median"),col=c("blue","green"),lty=c(1,1))
```





The histogram of residuals for bothe the linear and quadtaic fits are fairly normal. To confirm, we plotted the mean and the median which are fairly close together. This shows that the residuals seem to be normal. As the mean and median are closer together in the quadratic fit, this also indicates that the residuals for the quadratic fit are slightly more normally distributed. From this we can say that our quadratic trend seems to be a better fit for the model. We also see that the mean (blue) is approximately zero for both the linear and quadratic fits, indicating that our forecast will be unbiased.

(g) Perform a Jarque-Berra Test on the two sets of residuals and discuss your results.

```r
jarque.bera.test(m1$residuals)  #jarque-berra test for lineear fit resids
```

```
##
##  Jarque Bera Test
##
## data:  m1$residuals
## X-squared = 3.6618, df = 2, p-value = 0.1603
```

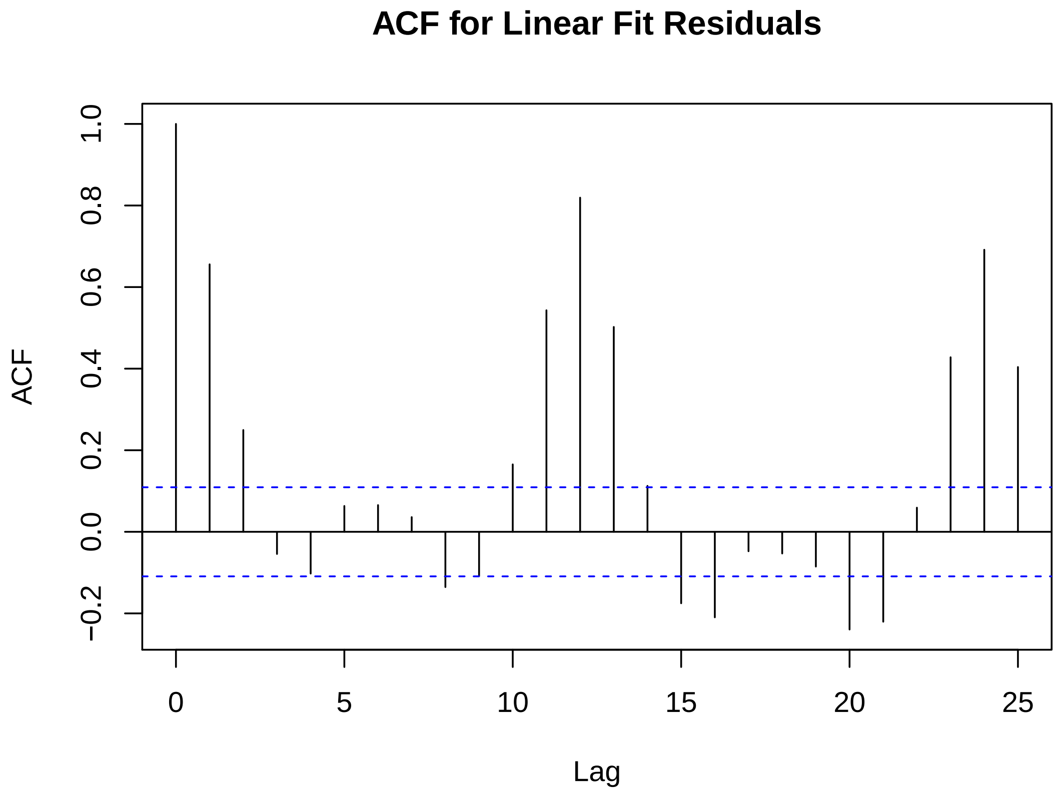
```r
jarque.bera.test(m2$residuals)  #jarque-berra test for quadratic residuals
```

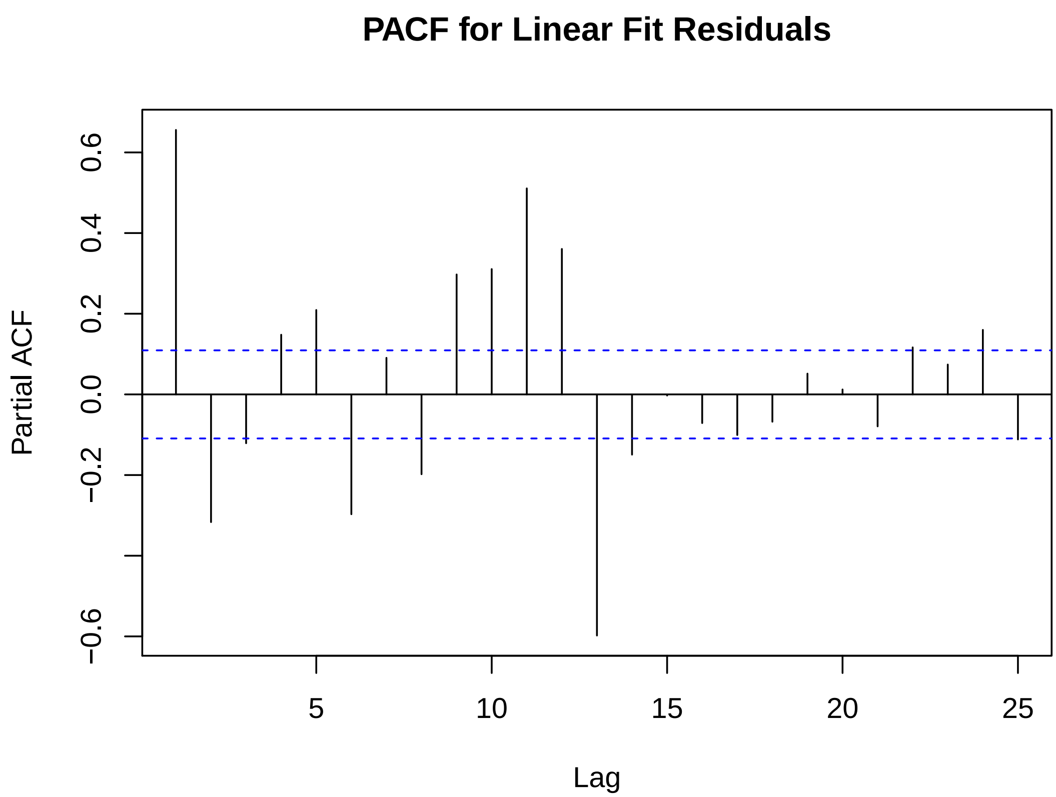
```
##
##  Jarque Bera Test
##
## data:  m2$residuals
## X-squared = 1.1151, df = 2, p-value = 0.5726
```

The p-values are greater than 0.05 so we fail to reject the null of normality. Therefore our residuals are normally distributed. This test also confirms what we were thinking in the previous part - the quadratic trend seems to be a better fit.

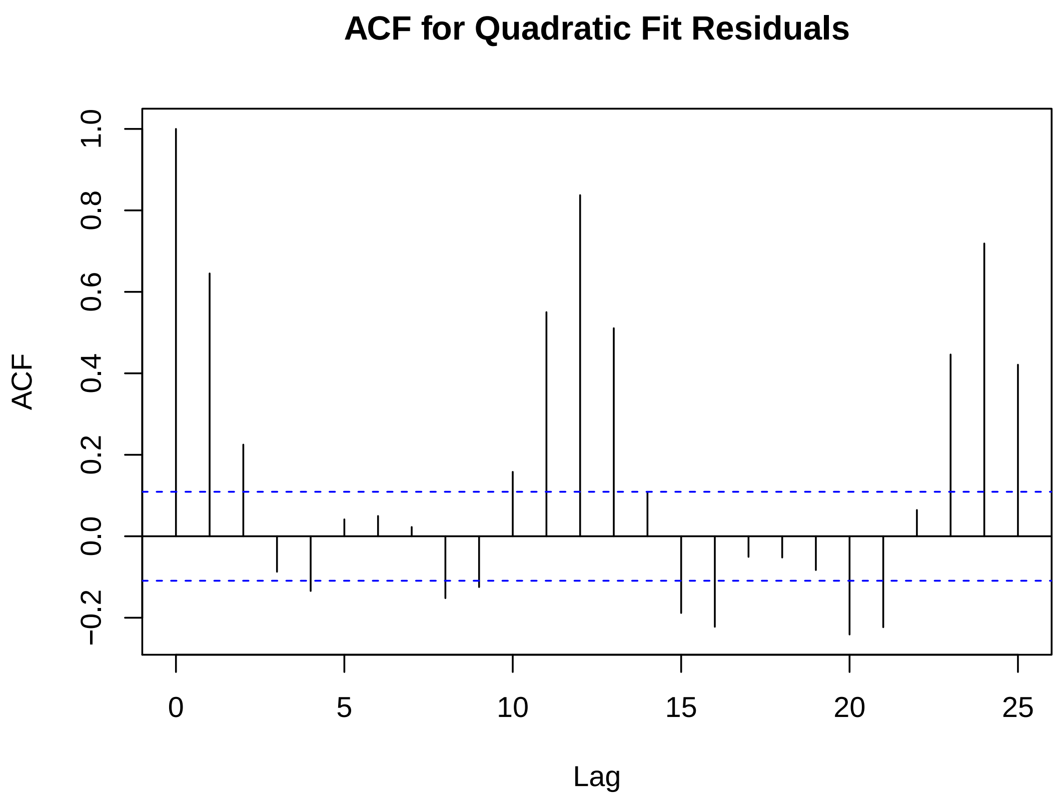
(h) Plot and discuss the ACF and PACF of the residuals from each one of your model fits.

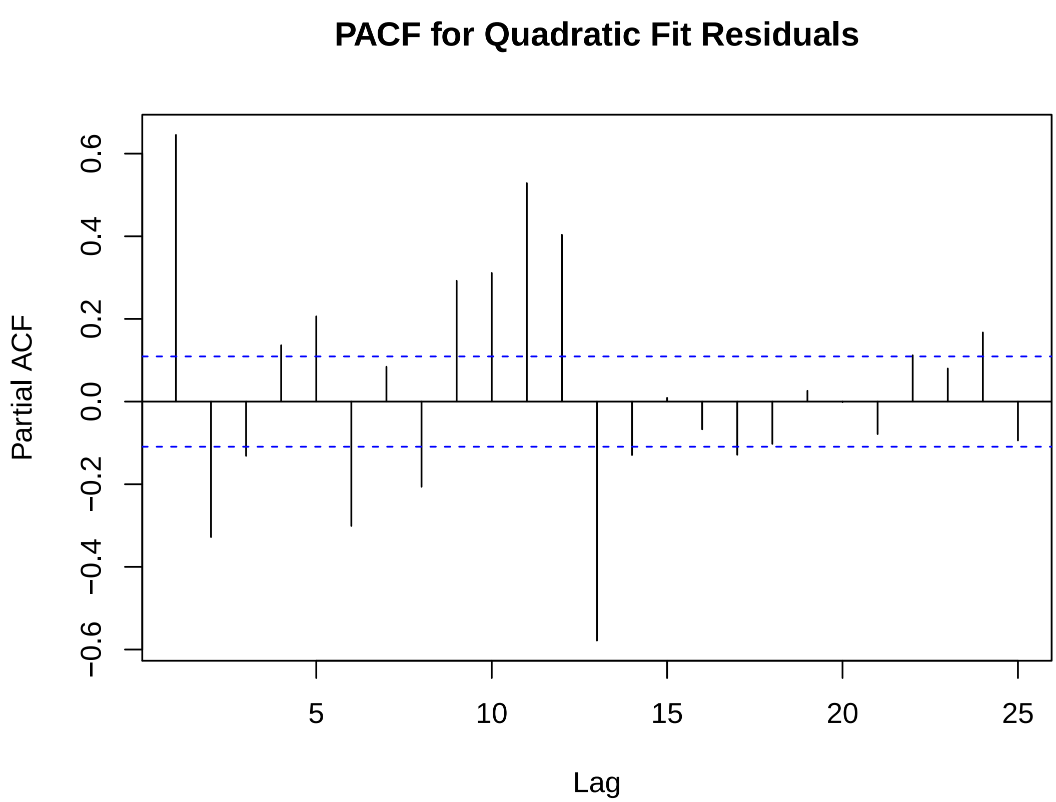
```r
par(mfrow=c(2,1))
#Plot ACF and PACF for linear fit residuals
acf(m1$residuals, main="ACF for Linear Fit Residuals")
pacf(m1$residuals, main = "PACF for Linear Fit Residuals")
```


ACF for Linear Fit Residuals
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
Lag


PACF for Linear Fit Residuals
Partial ACF
0.6
0.4
0.2
0.0
−0.2
−0.6
5
10
15
20
25
Lag

```r
par(mfrow=c(2,1))
#Plot ACF and PACF for quadratic fit residuals
acf(m2$residuals, main="ACF for Quadratic Fit Residuals")
pacf(m2$residuals,main = "PACF for Quadratic Fit Residuals")
```


ACF for Quadratic Fit Residuals
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
Lag


PACF for Quadratic Fit Residuals
Partial ACF
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
5
10
15
20
25
Lag

We can see from the ACF and PACF plots of the residuals for both the linear and quadratic fits that there remains some structure/pattern that could be a result of the seasonality in the series. Incorporating a seasonal component into the model should help capture this structure and improve the estimates of the model.

(i) For each model, discuss the associated diagnostic statistics (R^2, t-distribution, F-distribution, etc.)

```
summary(m1)                              #summary statistics for linear fit
```

```
##
## Call:
## lm(formula = hw_ts ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -530.71 -113.78   20.05  118.74  759.04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -80404.996   2775.712  -28.97   <2e-16 ***
## t               40.861      1.384   29.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 192.4 on 320 degrees of freedom
## Multiple R-squared:  0.7314, Adjusted R-squared:  0.7306
## F-statistic: 871.5 on 1 and 320 DF,  p-value: < 2.2e-16
```

```
summary(m2)                              #summary statistics for quadratic fit
```

```
##
## Call:
## lm(formula = hw_ts ~ t + I(t^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -513.14 -112.65    9.45  110.79  689.90
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.496e+06  7.916e+05   3.153  0.00177 **
## t           -2.529e+03  7.895e+02  -3.203  0.00150 **
## I(t^2)       6.407e-01  1.969e-01   3.255  0.00126 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 189.6 on 319 degrees of freedom
## Multiple R-squared:  0.7401, Adjusted R-squared:  0.7384
## F-statistic: 454.1 on 2 and 319 DF,  p-value: < 2.2e-16
```

Based on the summary statistics, we can see that the F-statistic and adjusted R^2 are fairly similar and indicate that the model is a good fit. But we know that we can still improve our fit by incorporating seasonality. We also notice that the p-values are slightly larger for the quadratic fit, making our variables less significant than in the linear fit model.

(j) Select a trend model using AIC and BIC (show the values obtained from each criterion). Do the selected models agree?

```r
AIC(m1,m2)                                          #model selection with AIC
```

```
##    df      AIC
## m1  3 4304.928
## m2  4 4296.409
```

```r
BIC(m1,m2)                                          #model selection with BIC
```

```
##    df      BIC
## m1  3 4316.252
## m2  4 4311.507
```

Both AIC and BIC agree that the quadratic fit is the better model. Therefore this will be our preferred model.

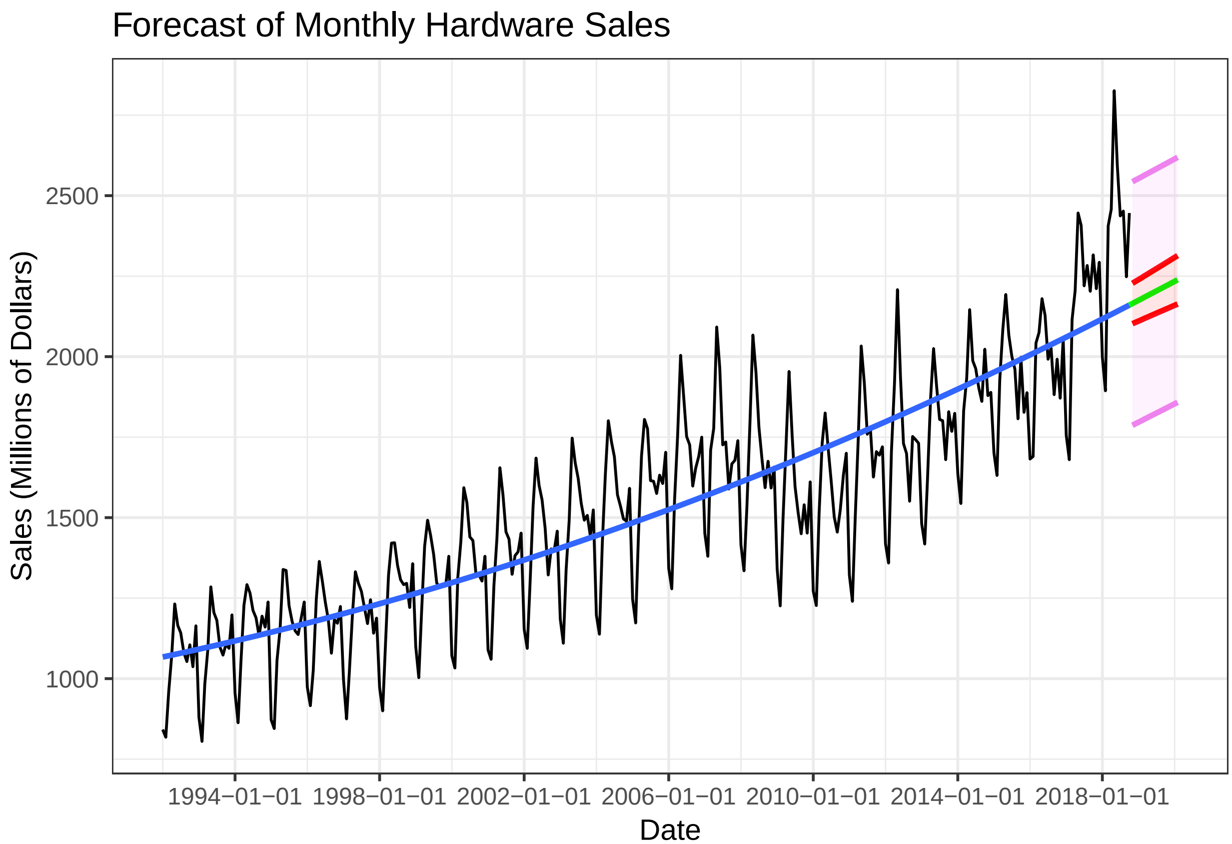
(k) Use your preferred model to forecast h-steps (at least 16) ahead. Your forecast should include the respective uncertainty prediction interval. Depending on your data, h will be in weeks, months, etc.

```r
#create new hardware data fram and time dummy
hardware2 = hardware
colnames(hardware2)[2] = "sales"
hardware2$t = 1:nrow(hardware2)
m4 = lm(sales ~ t + I(t^2), hardware2)
t2 = 1:length(hw_ts)

#solve for prediction intervals
tn2 = data.frame(t = 323:(323+15))
pred_p = predict(m4, tn2, level = 0.95, interval="prediction")
pred_c = predict(m4, tn2, level = 0.95, interval="confidence")

#create separate data frames to store original and forecasted data
known = data.frame(t = 1:322)
known$fc = NA

#store original data
for(i in 1:nrow(known)){
  known[i, "fc"] = hw_ts[i]
}


#store forecasted values
new = cbind(data.frame(pred_c), data.frame(pred_p))
new = new[-4]
colnames(new) = c("fit", "clwr", "cupr", "plwr", "pupr")
new$t = 1:nrow(new)
new$t = new$t + 322
temp = data.frame(fit = m2$fitted.values[322], clwr = NA, cupr = NA, plwr = NA, pupr = NA,
                  t = 322) #connect forecast to original data for plot
new = rbind(temp, new)

#create date for x-axis
known$year = c(rep(1992, 12),rep(1993, 12),rep(1994, 12),rep(1995, 12), rep(1996, 12),
               rep(1997, 12), rep(1998, 12),
  rep(1999, 12), rep(2000, 12), rep(2001, 12), rep(2002, 12), rep(2003, 12),
  rep(2004, 12), rep(2005, 12),
  rep(2006, 12), rep(2007, 12), rep(2008, 12), rep(2009, 12), rep(2010, 12),
```

```r
  rep(2011, 12), rep(2012, 12),
  rep(2013, 12), rep(2014, 12), rep(2015, 12), rep(2016, 12), rep(2017, 12),
  rep(2018, 10))

known$month = known$t %% 12
known[known$month == 0, "month"] = 12
known$day = rep(1, 322)
known$date = paste(as.character(known$year),
                   as.character(known$month), as.character(known$day), sep = "-")
known$date = as.Date(known$date)

new$year = c(rep(2018, 3), rep(2019, 12), rep(2020, 2))
new$month = new$t %% 12
new[new$month == 0, "month"] = 12
new$day = rep(1, 17)
new$date = paste(as.character(new$year),
                 as.character(new$month), as.character(new$day), sep = "-")
new$date = as.Date(new$date)

#plot
ggplot()+
  geom_line(data = known, aes(x = date, y = fc))+
  geom_smooth(data = known, aes(x = date, y = fc), method = "lm",
              formula = y ~ x + I(x^2), se = FALSE)+
  geom_line(data = new, aes(x = date, y = fit), color = "green", size = 1)+
  geom_line(data = new, aes(x = date, y = clwr), color = "red", size = 1)+
  geom_line(data = new, aes(x = date, y = cupr), color = "red", size = 1)+
  geom_line(data = new, aes(x = date, y = plwr), color = "violet", size = 1)+
  geom_line(data = new, aes(x = date, y = pupr), color = "violet", size = 1)+
  geom_ribbon(data = new, aes(x = date, y = fit, ymin = clwr, ymax = cupr),
              fill = "red", alpha = 0.1)+
  geom_ribbon(data = new, aes(x = date, y = fit, ymin = cupr, ymax = pupr),
              fill = "violet", alpha = 0.1)+
  geom_ribbon(data = new, aes(x = date, y = fit, ymin = plwr, ymax = clwr),
              fill = "violet", alpha = 0.1)+
  theme_bw()+
  labs(
    title = "Forecast of Monthly Hardware Sales",
    x = "Date",
    y = "Sales (Millions of Dollars)"
  )+
  scale_x_date(breaks = date_breaks("4 years"))
```

```
## Warning: Ignoring unknown aesthetics: y

## Warning: Ignoring unknown aesthetics: y

## Warning: Ignoring unknown aesthetics: y
## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



**Question 2 Modeling and Forecasting Seasonality (a) Construct and test (by looking at the diagnostic statistics) a model with a full set of seasonal dummies.

```r
fit = tslm(hw_ts~season)          #model with a full set of seasonal dummies
summary(fit)                      #summary statistics for model with seasonal dummies
```

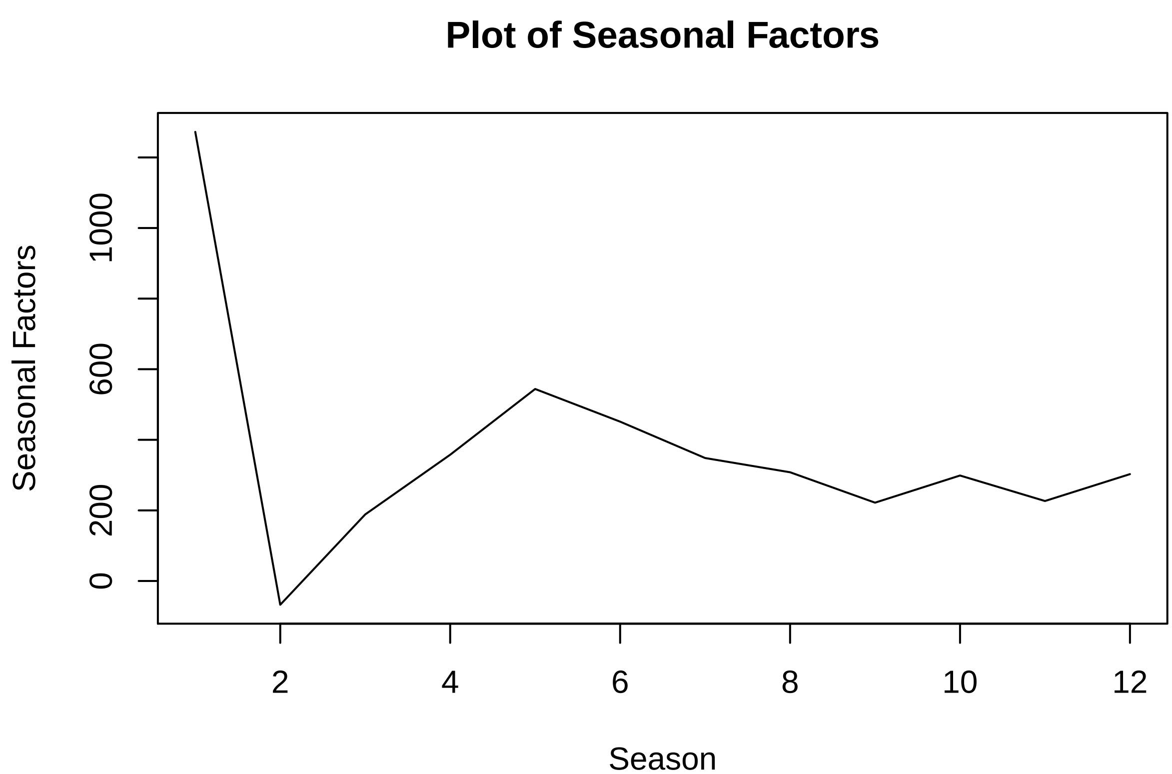
```
##
## Call:
## tslm(formula = hw_ts ~ season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -584.37 -278.17   -9.65  190.91 1009.63
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1272.41      65.03  19.565  < 2e-16 ***
## season2       -67.37      91.97  -0.733 0.464417
## season3       188.52      91.97   2.050 0.041234 *
## season4       357.63      91.97   3.888 0.000123 ***
## season5       543.96      91.97   5.914 8.80e-09 ***
## season6       451.44      91.97   4.908 1.49e-06 ***
## season7       348.37      91.97   3.788 0.000183 ***
## season8       308.04      91.97   3.349 0.000910 ***
```

```
## season9       221.89      91.97   2.413 0.016422 *
## season10      298.81      91.97   3.249 0.001286 **
## season11      226.59      92.85   2.440 0.015234 *
## season12      302.71      92.85   3.260 0.001238 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 337.9 on 310 degrees of freedom
## Multiple R-squared:  0.1973, Adjusted R-squared:  0.1688
## F-statistic: 6.928 on 11 and 310 DF,  p-value: 1.809e-10
```

The adjusted R^2 is much smaller than with the quadratic fit. This means that most of our model was explained by the trend and not by seasonality. The p-value associated with the F-statistic is small so we should keep all of the seasons. We notice that the most significant seasons are the summer months. Additionally, we see that the coefficeints for the summer months are larger. Intuitively this makes sense as we expect hardware store sales to peak during these months due to construction and outdoor projects.

(b) Plot the estimated seasonal factors and interpret your plot.

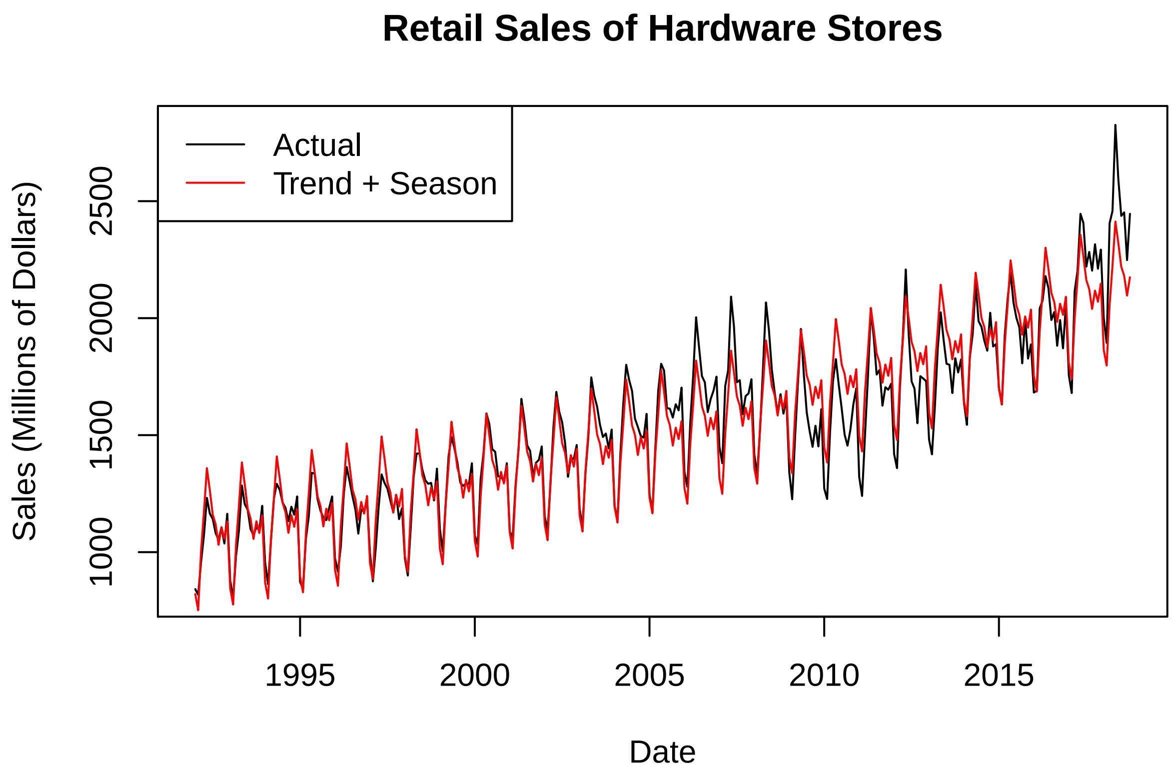
```
#Plot estimated seasonal factors
plot(fit$coefficients,type='l',ylab="Seasonal Factors",xlab="Season",
     main="Plot of Seasonal Factors")
```



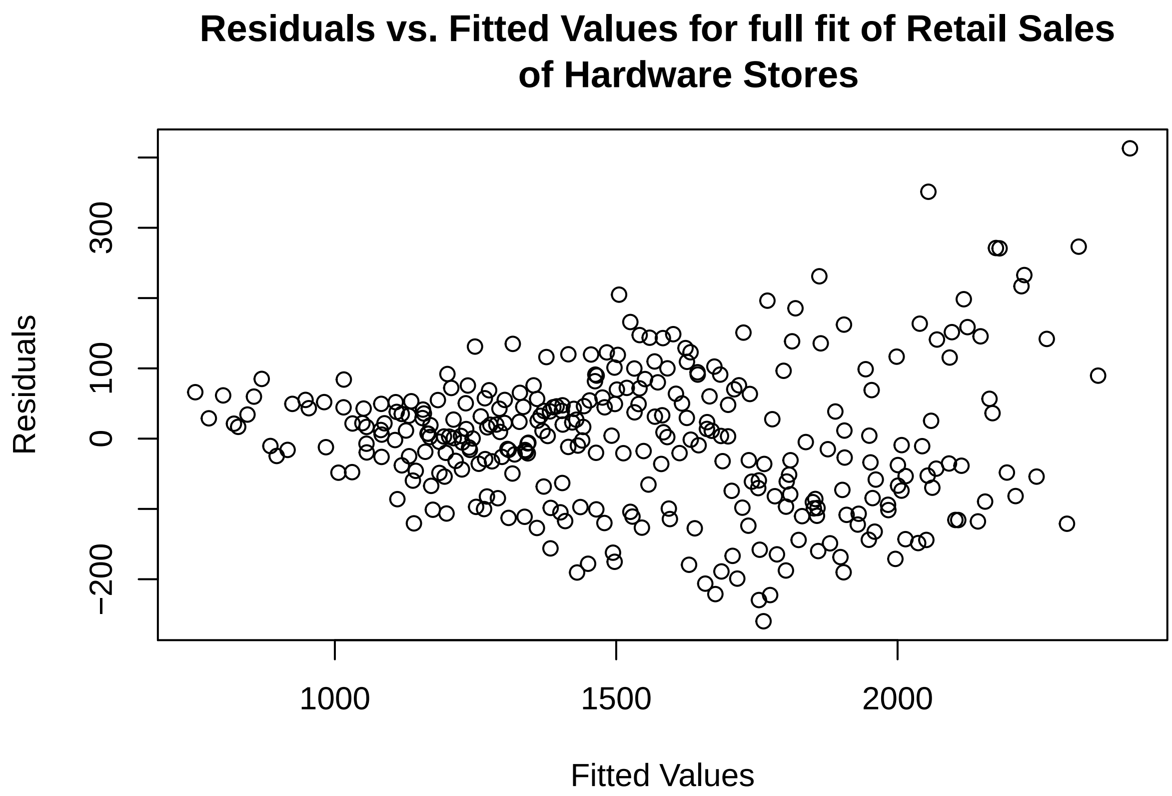
We can see that the summer months have higher sales while the winter and fall months have lower hardware sales. This is what we expected, however the fall and winter sales still aren't as low as expected. This is probably due to a continuous need for hardware store products in those months as well.

(c) In order to improve your model, add the trend model from problem 1 to your seasonal model. We will refer to this model as the full model. For the full model, plot the respective residuals vs. fitted values and discuss your observations.

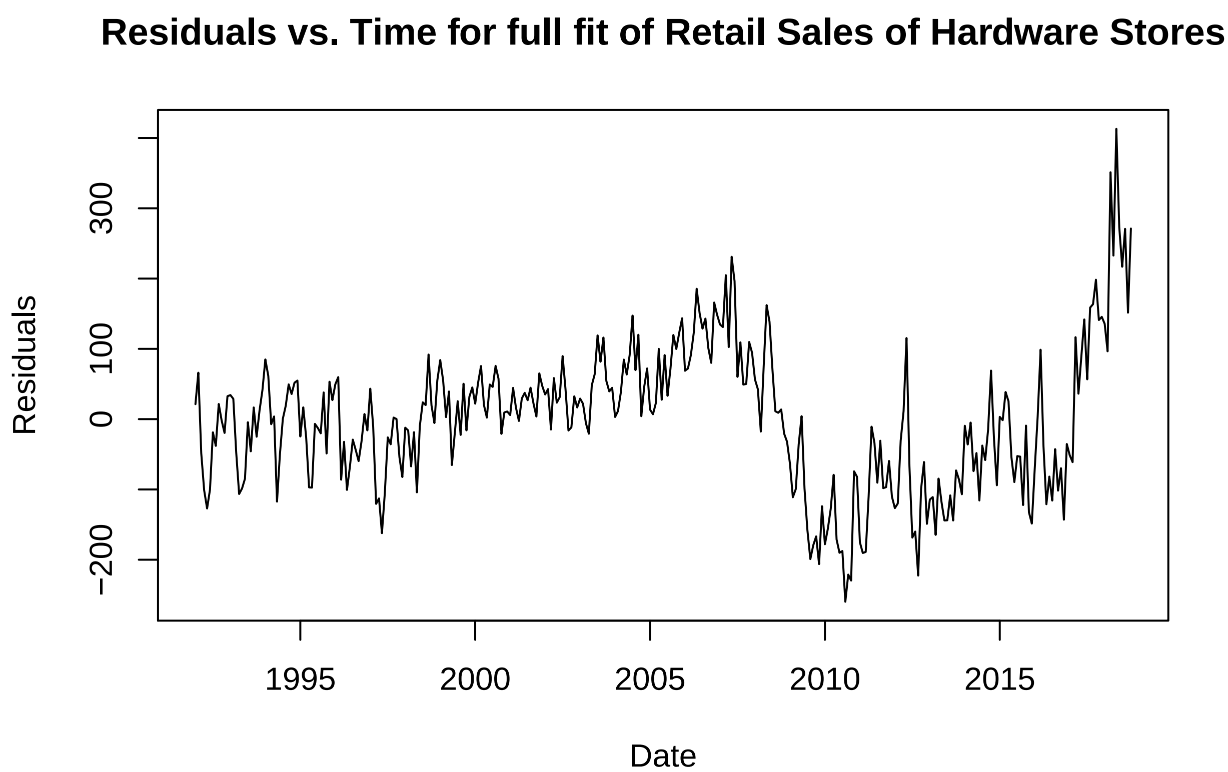
```
#create full model with season and trend
fit.full = tslm(hw_ts~t+I(t^2)+season)

#plot the full fit onto original data
plot(hw_ts,ylab="Sales (Millions of Dollars)", xlab = "Date",
     main = "Retail Sales of Hardware Stores")
lines(fit.full$fitted.values,col="red")
legend("topleft",c("Actual","Trend + Season"),col=c("black","red"),lty = c(1,1))
```




```
#plot residuals vs. fitted values for full fit of hardware stores sales
plot(fit.full$fitted.values,fit.full$residuals, ylab= "Residuals",
xlab = "Fitted Values",main = "Residuals vs. Fitted Values for full fit of Retail Sales
     of Hardware Stores")
```




```
#plot residuals vs. time for full fit of hardware stores sales
plot(t,fit.full$residuals,type='l',ylab="Residuals", xlab = "Date",
     main = "Residuals vs. Time for full fit of Retail Sales of Hardware Stores")
```



The first plot shows the original time series with our fitted model, which includes a trend and seasonal component.

The second plot shows residuals versus fitted observations in a scatterplot. We notice that the variance of the residuals is increasing for larger fitted values meaning that our model doesn't estimate well for larger values of sales, which were typically occurring post-recession.

The third plot shows residuals versus time. We notice a similar pattern with the variance of the residuals increasing with time. This indicates that our model was unable to estimate the drop in sales around 2009 (recession) because we were using a quadratic trend. This meant that our estimates for anything past 2009 were off by a larger amount than pre-recession.

(d) Interpret the respective summary statistics including the error metrics of your full model.

```r
summary(fit.full)                          #summary statistics for full model
```

```
##
## Call:
## tslm(formula = hw_ts ~ t + I(t^2) + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -259.79  -67.10    3.51   55.01  413.03
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.506e+06  4.269e+05   5.870 1.13e-08 ***
## t           -2.538e+03  4.258e+02  -5.962 6.82e-09 ***
## I(t^2)       6.430e-01  1.062e-01   6.057 4.03e-09 ***
```

```
## season2     -7.072e+01  2.781e+01  -2.543   0.0115 *
## season3      1.818e+02  2.781e+01   6.538 2.59e-10 ***
## season4      3.476e+02  2.781e+01  12.497  < 2e-16 ***
## season5      5.305e+02  2.781e+01  19.075  < 2e-16 ***
## season6      4.346e+02  2.781e+01  15.627  < 2e-16 ***
## season7      3.282e+02  2.781e+01  11.799  < 2e-16 ***
## season8      2.844e+02  2.781e+01  10.226  < 2e-16 ***
## season9      1.949e+02  2.782e+01   7.006 1.55e-11 ***
## season10     2.684e+02  2.782e+01   9.648  < 2e-16 ***
## season11     2.160e+02  2.808e+01   7.691 1.98e-13 ***
## season12     2.887e+02  2.808e+01  10.281  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 102.2 on 308 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.924
## F-statistic: 301.2 on 13 and 308 DF,  p-value: < 2.2e-16
```

From the summary table, we see that all the coefficients for trend and seasonal components are statistically significant. This is likely due to our model displaying a heavy seasonal component, which is why the coefficients for the seasons are so large. The adjusted R^2 is 0.924 which is significantly higher than when we just had the trend or seasonal components alone. The F-stat is also significant because its p-value is low, therefore we should include all trend and seasonal components.

(e) Perform a Jarque-Berra Test on the residuals and discuss your results.

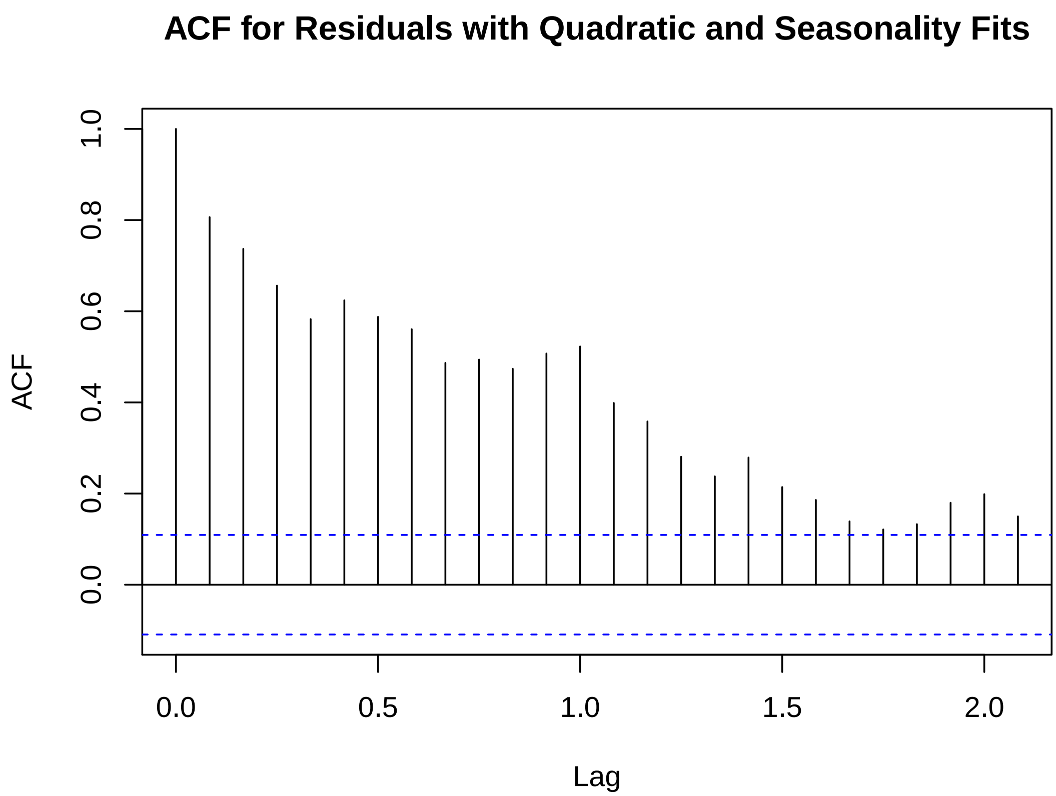
```
jarque.bera.test(fit.full$residuals)      #jarque-bera test for full model
```

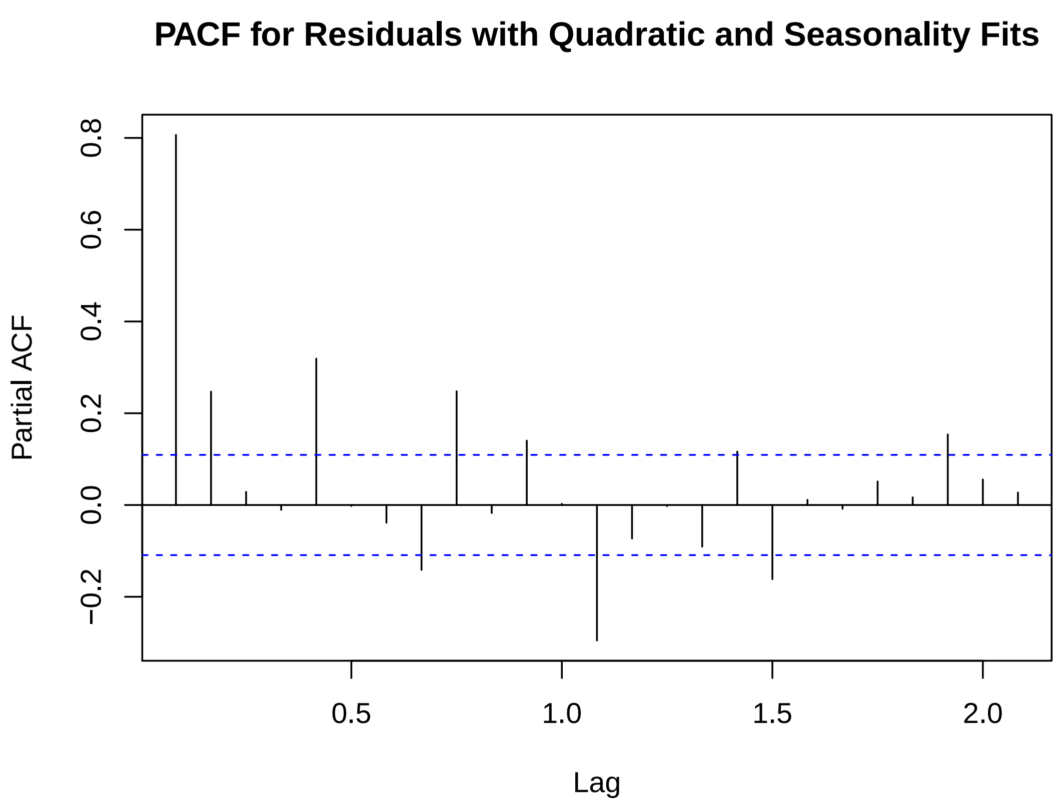
```
##
##  Jarque Bera Test
##
## data:  fit.full$residuals
## X-squared = 17.306, df = 2, p-value = 0.0001746
```

Because of a low p-value, we reject the null of normality. Therefore our residuals are not normal. This seems to be an effect of adding seasonality versus just having the trend. This could be a result of the model having a really good fit for pre-recession data versus post-recession data.

(f) Plot and discuss the ACF and the PACF of the residuals. How do these plots compare to 1(h)?

```
par(mfrow=c(2,1))

#ACF for residuals of full model
acf(fit.full$residuals, main= "ACF for Residuals with Quadratic and Seasonality Fits")

#PACF for residuals of full model
pacf(fit.full$residuals,main="PACF for Residuals with Quadratic and Seasonality Fits")
```


ACF for Residuals with Quadratic and Seasonality Fits
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Lag


PACF for Residuals with Quadratic and Seasonality Fits
Partial ACF
−0.2
0.0
0.2
0.4
0.6
0.8
0.5
1.0
1.5
2.0
Lag

Both the

ACF and PACF are showing that our full model is better than our model that only incorporated the quadratic trend. The autocorrelations are becoming less dependent on time, indicating that our model is acounting for most of the trend and seasonality that is dependent on time.

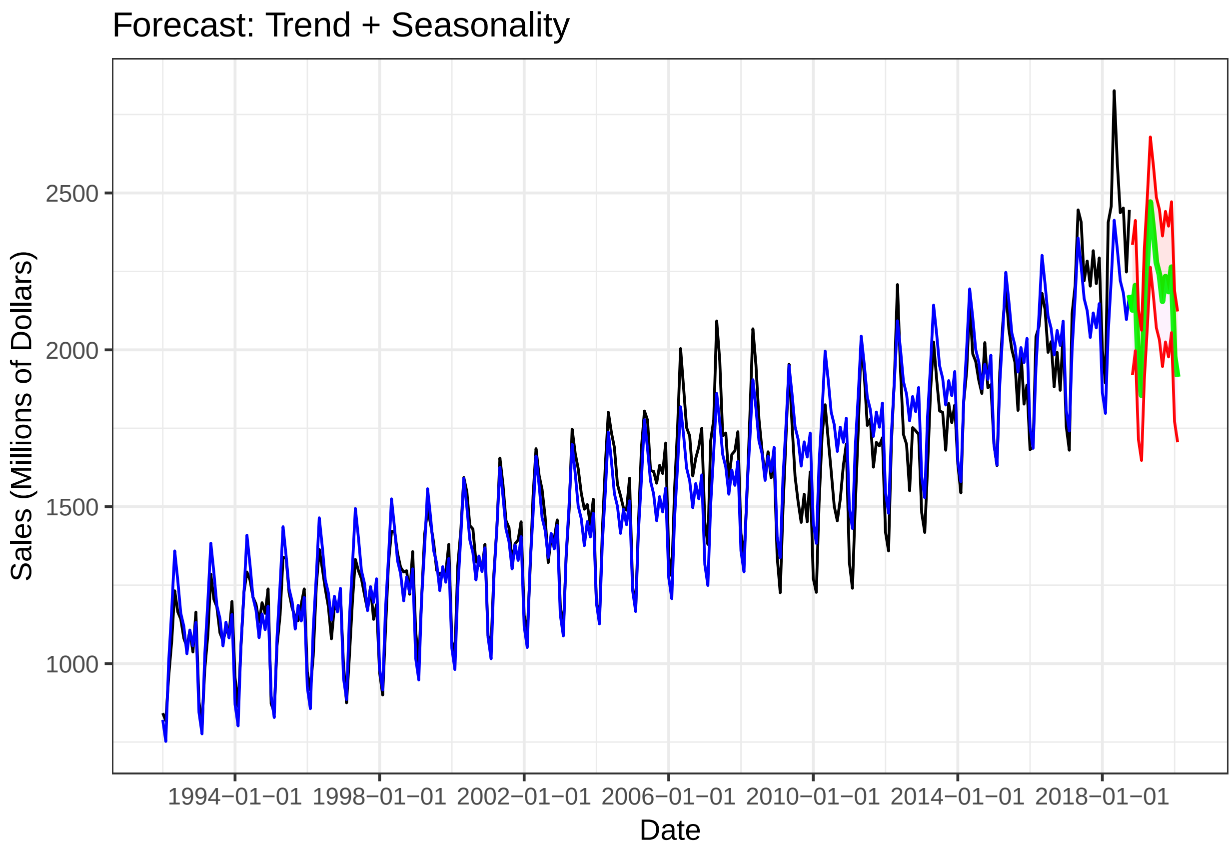
(g) Use the full model to forecast h-steps (at least 16) ahead. Your forecast should include the respective prediction interval.

```
#re-define time dummy
t2 = data.frame(t = 1:(nrow(hardware) + 16))
t2$month = t2$t %% 12
t2[t2$month == 0, "month"] = 12
t2[1:322, "val"] = hardware$MRTSSM44413USN

#fit model using trend and create seasonal dummies
fit2 = lm(val ~ t + I(t^2) + factor(month), t2[1:322, ])
t2[1:322, "fit"] = fit2$fitted.values

t2$season1 = 0
t2$season2 = 0
t2$season3 = 0
t2$season4 = 0
t2$season5 = 0
t2$season6 = 0
t2$season7 = 0
t2$season8 = 0
t2$season9 = 0
t2$season10 = 0
t2$season11 = 0
t2$season12 = 0

t2[t2$month == 1, "season1"] = 1
t2[t2$month == 2, "season2"] = 1
t2[t2$month == 3, "season3"] = 1
t2[t2$month == 4, "season4"] = 1
t2[t2$month == 5, "season5"] = 1
t2[t2$month == 6, "season6"] = 1
t2[t2$month == 7, "season7"] = 1
t2[t2$month == 8, "season8"] = 1
t2[t2$month == 9, "season9"] = 1
t2[t2$month == 10, "season10"] = 1
t2[t2$month == 11, "season11"] = 1
t2[t2$month == 12, "season12"] = 1

#using the model to calculate the fit value for each row
for(i in 323:nrow(t2)){
  t2[i, "fit"] =
  fit2$coefficients[1]+
  fit2$coefficients[2]*t2[i, "t"]+
  fit2$coefficients[3]*(t2[i, "t"]^2)+
  fit2$coefficients[4]*t2[i, "season2"]+
  fit2$coefficients[5]*t2[i, "season3"]+
  fit2$coefficients[6]*t2[i, "season4"]+
  fit2$coefficients[7]*t2[i, "season5"]+
  fit2$coefficients[8]*t2[i, "season6"]+
  fit2$coefficients[9]*t2[i, "season7"]+
```

```r
  fit2$coefficients[10]*t2[i, "season8"]+
  fit2$coefficients[11]*t2[i, "season9"]+
  fit2$coefficients[12]*t2[i, "season10"]+
  fit2$coefficients[13]*t2[i, "season11"]+
  fit2$coefficients[14]*t2[i, "season12"]
}

#get prediction intervals
prediction = predict(fit2, t2, interval="predict")

t2$lwr = NA
t2$upr = NA

t2[323:nrow(t2), "lwr"] = prediction[ 323:nrow(t2), 2]
t2[323:nrow(t2), "upr"] = prediction[ 323:nrow(t2), 3]

#creating date variable for x-axis
t2$day = 1
t2$year = NA
t2[323:nrow(t2), "year"] = c(rep(2018, 2), rep(2019, 12), rep(2020, 2))

t2$date = NA
t2[1:nrow(hardware), "date"] = as.character(hardware$DATE)
t2[323:nrow(t2), "date"] = paste(as.character(t2[323:nrow(t2), "year"]),
                                 as.character(t2[323:nrow(t2), "month"]),
                                 as.character(t2[323:nrow(t2), "day"]), sep = "-")
t2$date = as.Date(t2$date)

#plot
ggplot()+
  geom_line(data = t2[1:322,], mapping = aes(x = date, y = val))+
  geom_line(data = t2[1:322,], mapping = aes(x = date, y = fit), color = "blue")+
  geom_line(data = t2[322:nrow(t2), ], mapping = aes(x = date, y = fit),
          color = "green", size = 1)+
  geom_line(data = t2[322:nrow(t2), ], mapping = aes(x = date, y = upr), color = "red")+
  geom_line(data = t2[322:nrow(t2), ], mapping = aes(x = date, y = lwr), color = "red")+
  theme_bw()+
  geom_ribbon(data = t2[322:nrow(t2), ], aes(x = date, y = fit, ymin = fit, ymax = upr),
            fill = "red", alpha = 0.1)+
  geom_ribbon(data = t2[322:nrow(t2), ], aes(x = date, y = fit, ymin = lwr, ymax = fit),
            fill = "violet", alpha = 0.1)+
  theme_bw()+
  labs(
    title = "Forecast: Trend + Seasonality",
    x = "Date",
    y = "Sales (Millions of Dollars)"
  )+
  scale_x_date(breaks = date_breaks("4 years"))
```

```
## Warning: Ignoring unknown aesthetics: y

## Warning: Ignoring unknown aesthetics: y

## Warning: Removed 1 rows containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Part 3 Conclusions and Future Work (state your conclusion regarding your final model and forecast, and provide some insight as to how it could be improved).

Our final model that incorporates both seasonality and trend is a good fit given our high adjusted R^2. However, the model becomes less accurate after the recession. The drop in sales are not captured by our model. If we were able to capture the drop in sales, we could significantly improve both our model and our forecast.

Part 4 References (include source of your data and any other resources).

Hardware Sales Data: https://fred.stlouisfed.org/series/MRTSSM44413USN