Project 1

Authors

Authors: Alvin Ng, Jenny Wang, Ruixi Zhou, Austin Lee

Introduction

In this project, we will be forecasting attendence for a non-profit organization based in Irvine that aims to promote child-development. We hope to provide management insights on their data so they can adjust their resources accordingly and prevent unnecessary costs.

The data used in this project is the Monthly attendence of the non-profit organization. The data contains daily ticket entries excluding entries from birthday parties and field trips (Attendence), if it is federal holiday (Fed Holiday), if the day is a Saturday or Sunday (Weekend), and if the museum is closed (Closed). Data is collected from it's opening day, September 1st 2009, until present, January 20th 2019.

1A. Time series plot of the data

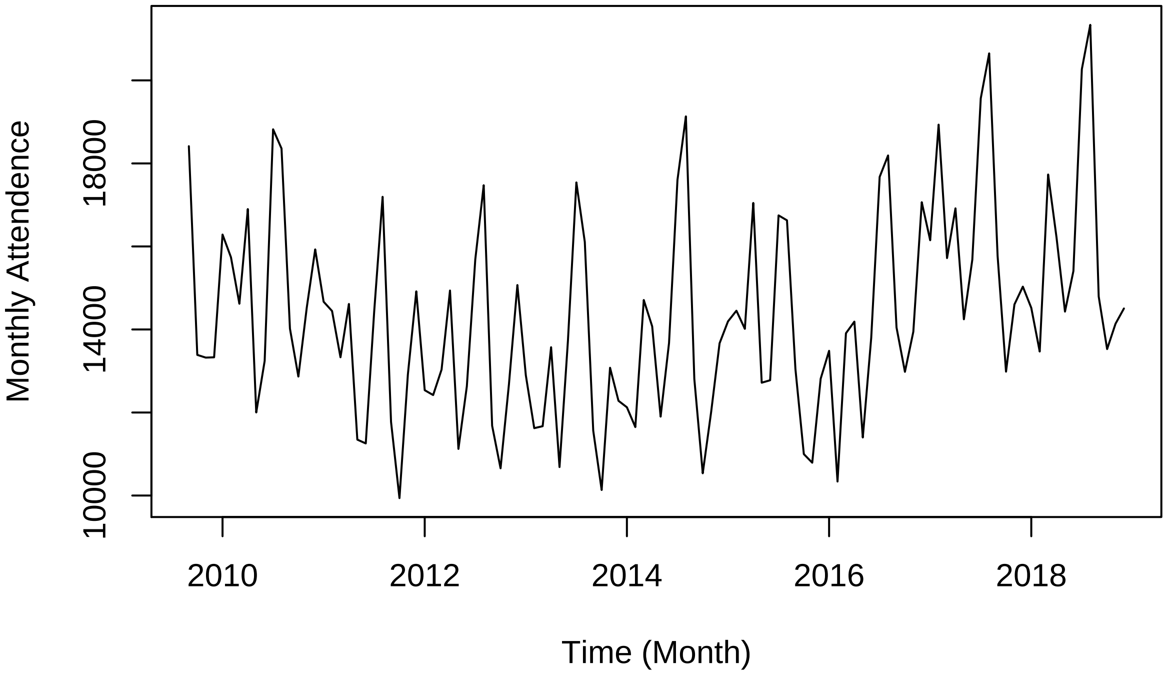
```r
#Here we read in our excel data
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.5.2
```

```r
library(forecast)
```

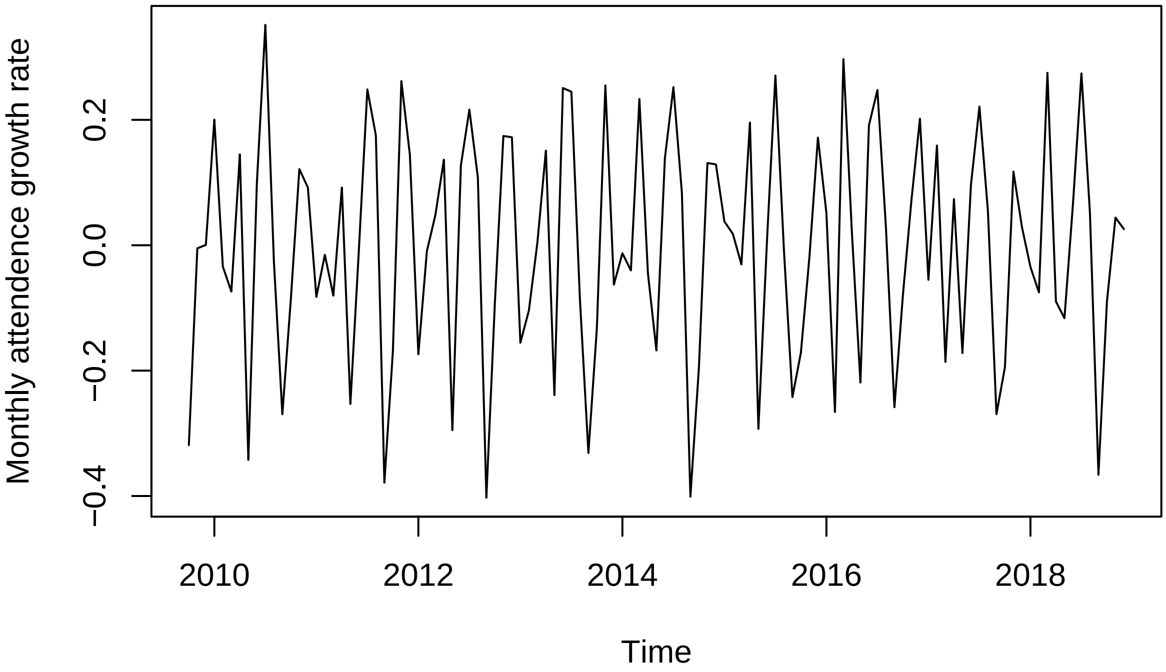
```
## Warning: package 'forecast' was built under R version 3.5.2
```

```r
Monthly_data <- read_excel("PretendCity Daily Attendence.xlsx",
    sheet = "Monthly")
# create a time series model from the data set
ts_data <- ts(Monthly_data$Attendence, start = c(2009,9), frequency = 12)
# plot the time series model
plot(ts_data, ylab="Monthly Attendence", xlab="Time (Month)")
```



1B. Covariance stationary

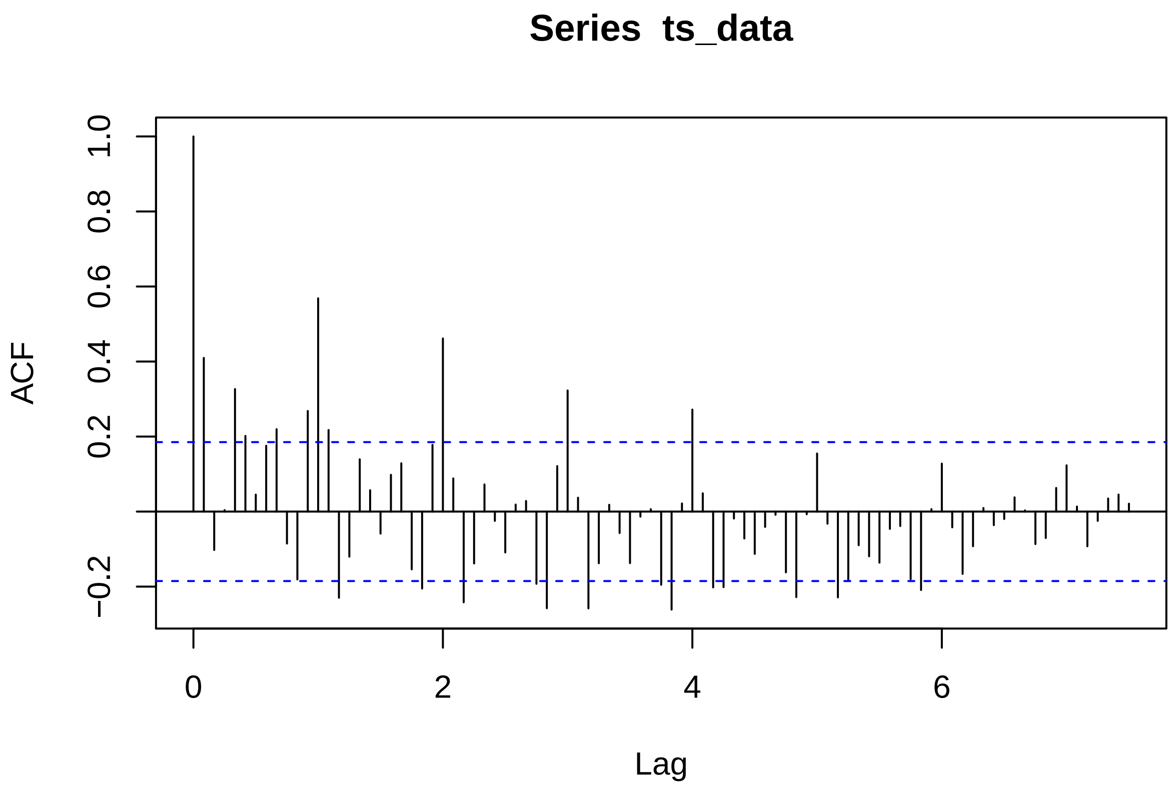
```r
# plot the covariance of the time series by taking the first difference of the log of the time series
# the difference of the log of the time series give us the percentage point changes between each point
plot(diff(log(ts_data)), ylab="Monthly attendence growth rate")
```



The plot above suggest that the observed data is covariance stationary because it has the same variance mean.

1C. ACF

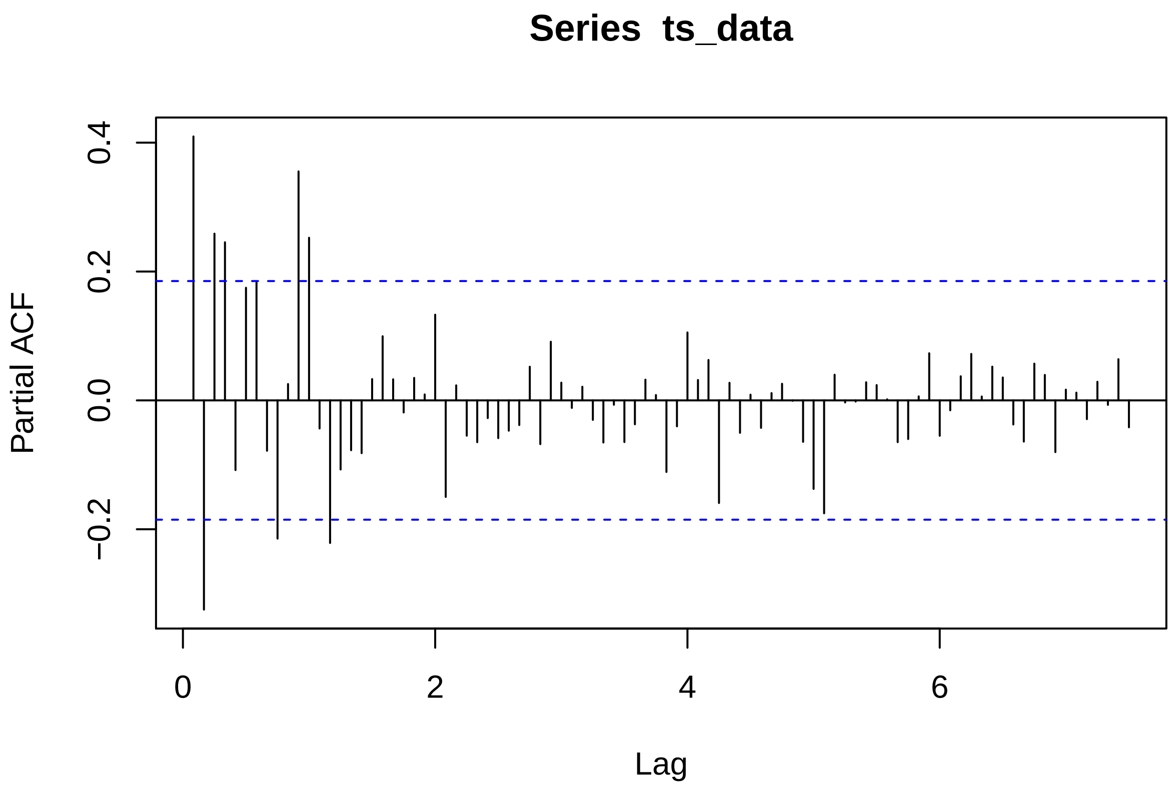
```
# finding the autocorrelation function of the time series model
acf(ts_data, lag.max = 90)
```



The graph above shows trends and cycle indictating that simply logging the function does not capture all the dynamics of the data, such as seasonality. We also see that around december and January, there is a spike in ACF, implying time dependence.

1C. PACF

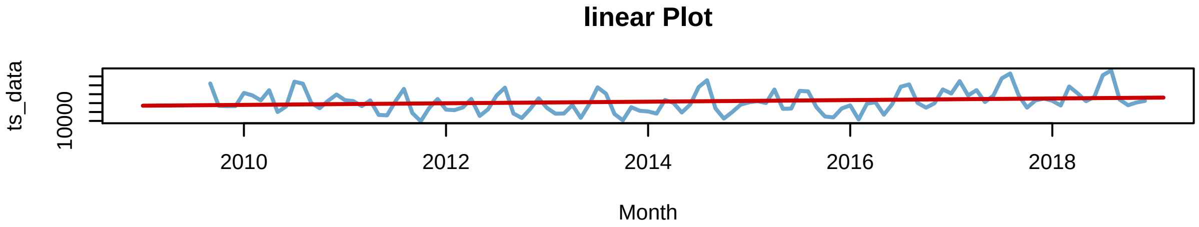
```r
# finding the partial autocorrelation funciton of the time series model, removing all the information i
pacf(ts_data, lag.max = 90)
```

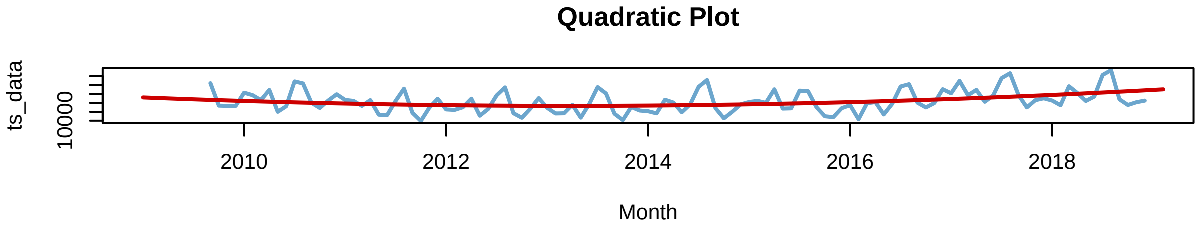


After the initial two years, PACF decreased significantly, implying that after removing information between time t and t+1, the data is no longer time dependence.

1D. Linear and Non-Linear Fit

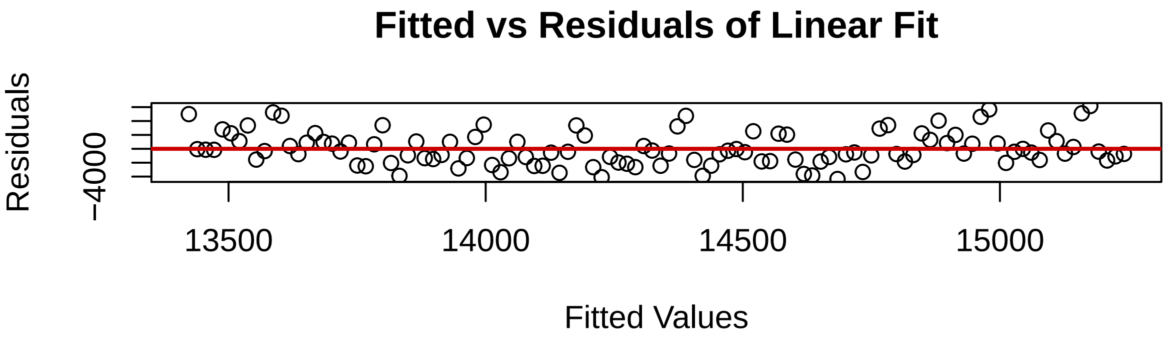
```
# Create a sequence to show the length of time

t <- seq(2009, 2019.1,length=length(ts_data))
# Linear fit of data vs time
lin_fit = tslm(ts_data~t)
# Quadratic fit of data vs time
quad_fit=tslm(ts_data~t+I(t^2))

# Plot both fits against the original data
par(mfrow=c(3,1))
plot(ts_data, xlab="Month", lwd=2, col='skyblue3', xlim=c(2009,2019), main="linear Plot")
lines(t,lin_fit$fit,col="red3",lwd=2)
plot(ts_data, xlab="Month", lwd=2, col='skyblue3', xlim=c(2009,2019), main="Quadratic Plot")
lines(t,quad_fit$fit,col="red3",lwd=2)
```

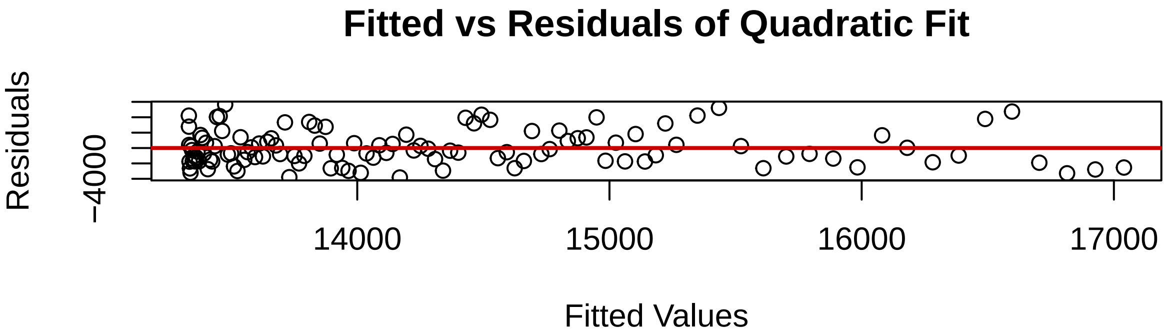




1E. Residuals vs Fitted Values

```
# Plot Linear and Quadratic model to compare the residuals (predicted) and fitted values (actual)
par(mfrow=c(2,1))
plot(as.vector(fitted(lin_fit)),as.vector(residuals(lin_fit)), ylab="Residuals",xlab="Fitted Values", ma
abline(h=0,lwd=2,col = "red3")
plot(as.vector(fitted(quad_fit)),as.vector(residuals(quad_fit)), ylab="Residuals",xlab="Fitted Values",
abline(h=0,lwd=2,col = "red3")
```

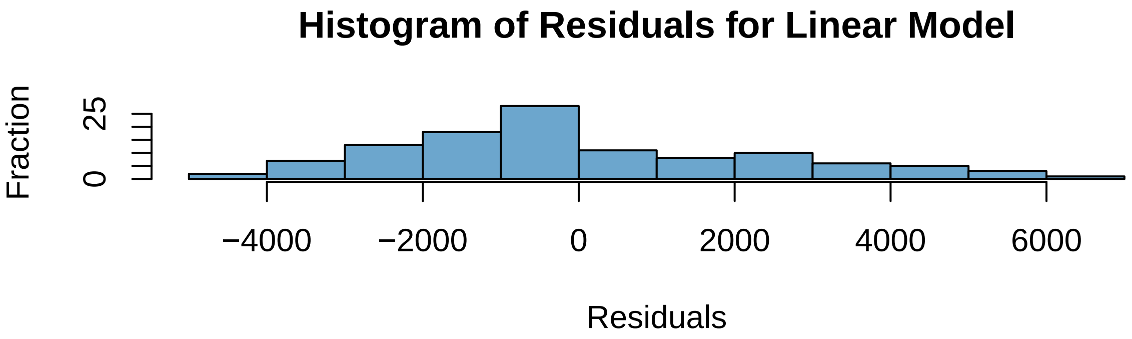


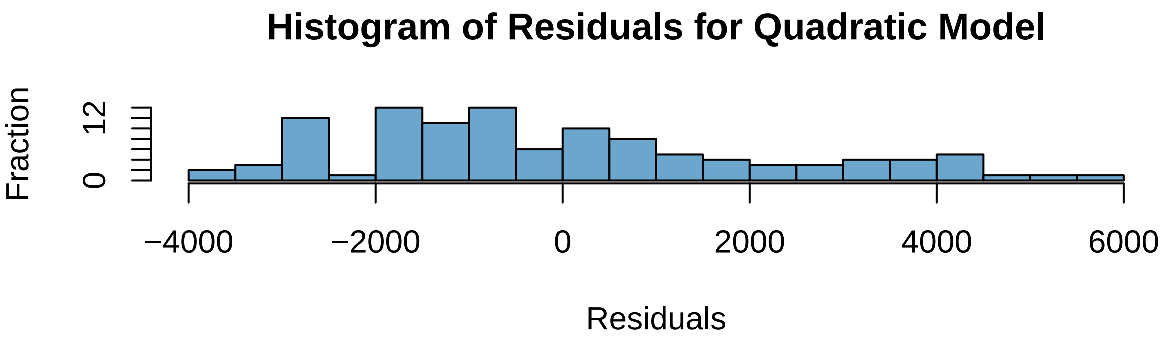


In both of these graphs the residuals form an almost horiztonal line around zero and appear to bounce randomly between zero. This indicates that both the linear and quadratic relationship could be reasonable. One difference to note would be that the residuals in the linear model seem to be evenly spread horizontally while in the quadratic model they appear to be clustered more densely on the left side. However, the variance still seems to be evenly spread so the relationship should still be reasonable.

1F. Histogram of Residuals

```
# Plot the residuals (error amount) or both the linear and quadratic fit
par(mfrow=c(2,1))
hist(lin_fit$res,15,col="skyblue3",xlab="Residuals",ylab="Fraction",main="Histogram of Residuals for Li
hist(quad_fit$res,15,col="skyblue3",xlab="Residuals",ylab="Fraction",main="Histogram of Residuals for Qu
```





For both cases the residuals are centered around zero, which means that there is not a trend in the residuals which is an indication that this is a good model. However, the residual in both models have a tail on the right, indicating there might be a some dynamics we are not capturing.

1G. Diagnostic Statistics

```
# Run the statistics of both the linear and quadratic fit
summary(lin_fit)
```

```
##
## Call:
## tslm(formula = ts_data ~ t)
##
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -4347  -1626   -458   1146   6161 
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -348276.1   155287.5  -2.243   0.0269 *
## t               180.0       77.1   2.335   0.0214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2400 on 110 degrees of freedom
## Multiple R-squared:  0.04723,    Adjusted R-squared:  0.03857
## F-statistic: 5.453 on 1 and 110 DF,  p-value: 0.02135
```

```
summary(quad_fit)
```

```
##
## Call:
## tslm(formula = ts_data ~ t + I(t^2))
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3832  -1679   -537   1338   5657
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.327e+08  1.120e+08   3.864 0.000190 ***
## t           -4.299e+05  1.112e+05  -3.865 0.000189 ***
## I(t^2)       1.068e+02  2.761e+01   3.867 0.000188 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2261 on 109 degrees of freedom
## Multiple R-squared:  0.1622, Adjusted R-squared:  0.1468
## F-statistic: 10.55 on 2 and 109 DF,  p-value: 6.485e-05
```

For both models the adjusted R squared is very low, with the quadratic model slightly larger at 0.1622 compared the linear model of 0.04723. This indicates that neither model is a very good fit as the amount of error is still very large, though the quadratic fit is slightly better. The F-statistic for both are also large with 10.55 for the quadratic and 5.453 for the linear fit. A high F-stat means that we can reject the null hypothesis that the group means are equal. Also, in the quadratic model the t-values show that all of the variables are significant whereas in the linear model the two variables are less statistically significant. Therefore, while both models are not good fits, the quadratic performs slightly better than the linear.

1H. AIC and BIC

```
#AIC and BIC functions to run the AIC and BIC for the linear and quadradic model
AIC(lin_fit,quad_fit)
```

```
##          df      AIC
## lin_fit   3 2065.303
## quad_fit  4 2052.904
```

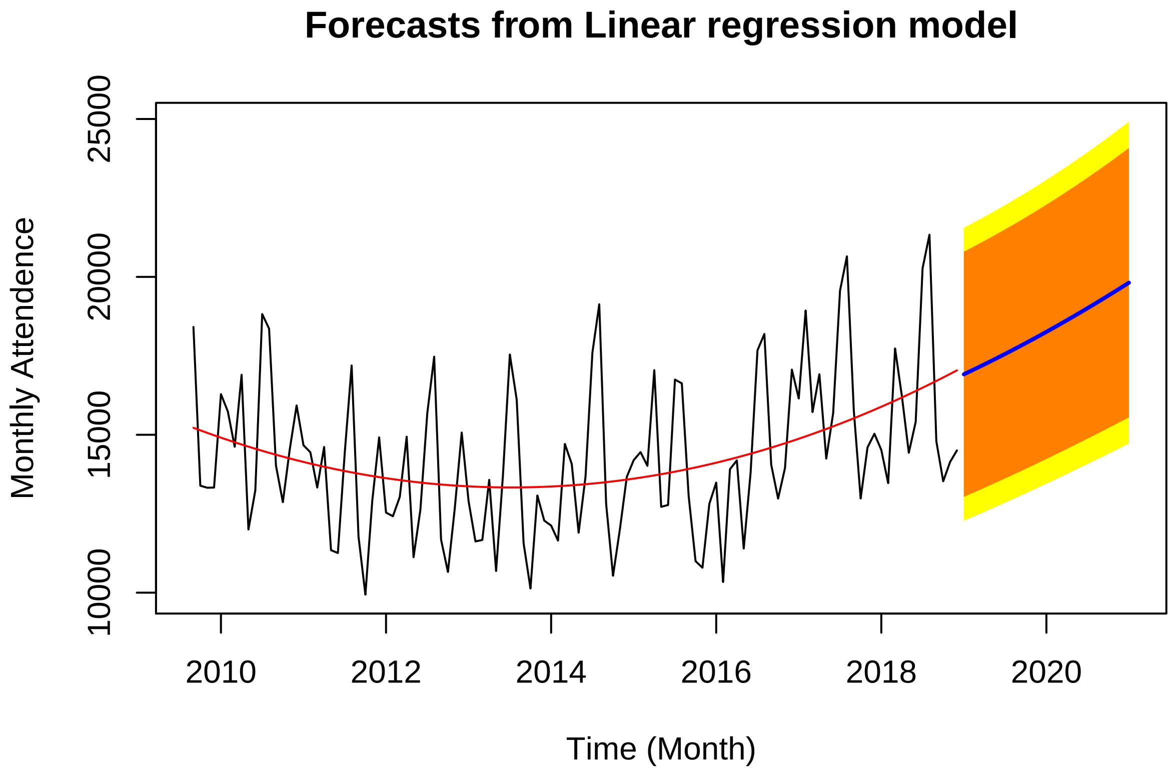
```
BIC(lin_fit,quad_fit)
```

```
##          df      BIC
## lin_fit   3 2073.459
## quad_fit  4 2063.778
```

According to both AIC and BIC, the quadradic fit model has a better goodness of fit when compared to the linear fit model.

1I. Forecast and Prediction Interval

```
#Forecast the plot of the quadradic fit
#We also need to fit the data using data.frame to reformat the data
quad_fit_forecast <- forecast(quad_fit, level = c(90,95), newdata=data.frame(t=seq(2019, 2021,by=(1/12)
plot(quad_fit_forecast,ylab="Monthly Attendence", xlab="Time (Month)", shadecols="oldstyle")
lines(quad_fit_forecast$fitted, col="red")
```



2A. Seasonal Diagonstics

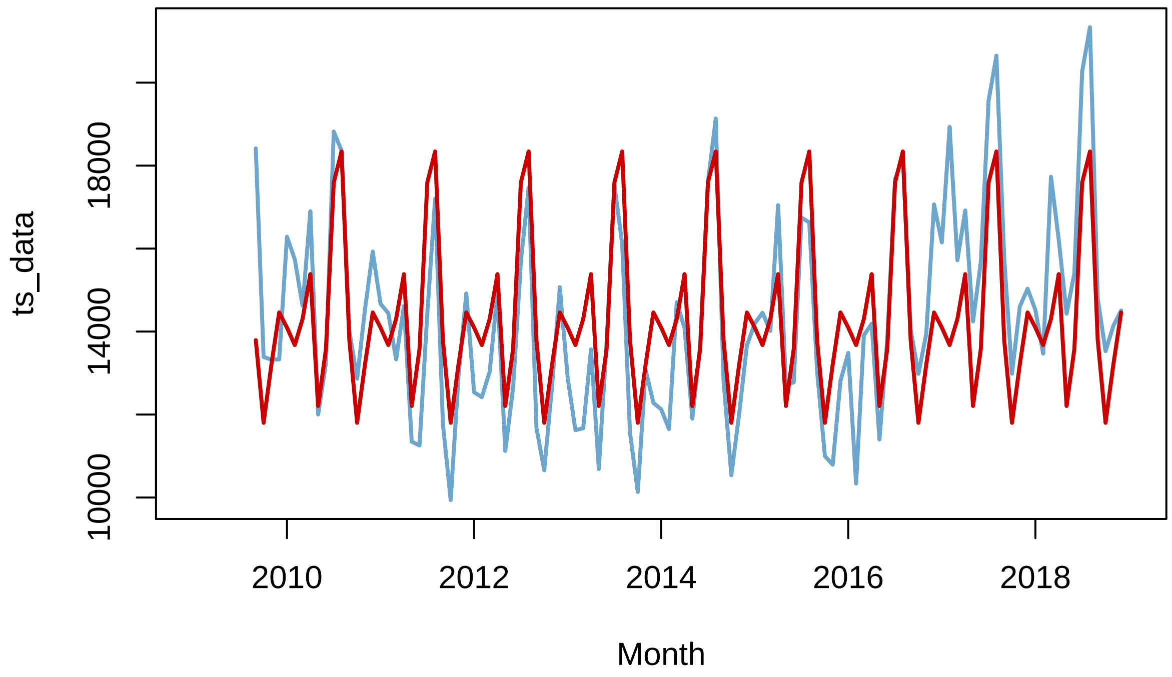
```
#Creating a time series regression and creating the summary information
season_fit <- tslm(ts_data ~ season)
summary(season_fit)
```

```
##
## Call:
## tslm(formula = ts_data ~ season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3337.3 -1157.5   -87.5  1093.7  5259.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14096.3      562.4  25.063  < 2e-16 ***
## season2       -422.0      795.4  -0.531  0.59690
## season3        208.2      795.4   0.262  0.79403
## season4       1288.7      795.4   1.620  0.10835
## season5      -1890.9      795.4  -2.377  0.01934 *
## season6       -517.9      795.4  -0.651  0.51647
## season7       3496.8      795.4   4.396 2.75e-05 ***
## season8       4245.7      795.4   5.338 5.89e-07 ***
## season9       -303.0      775.3  -0.391  0.69671
## season10     -2295.5      775.3  -2.961  0.00383 **
## season11      -894.5      775.3  -1.154  0.25131
## season12       364.8      775.3   0.471  0.63901
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1687 on 100 degrees of freedom
## Multiple R-squared:  0.572,  Adjusted R-squared:  0.525
## F-statistic: 12.15 on 11 and 100 DF,  p-value: 3.454e-14
```

At the 5% level, most of our seasonal coefficients are statistically insignificant. The R^2, a measure of goodness of fit is measured at .572. The F statistic stands at a 12.15, meaning at least some of the variables should be included in the model because we reject the null hypothesis that each coefficient equals 0.

2B. Seasonal Plot

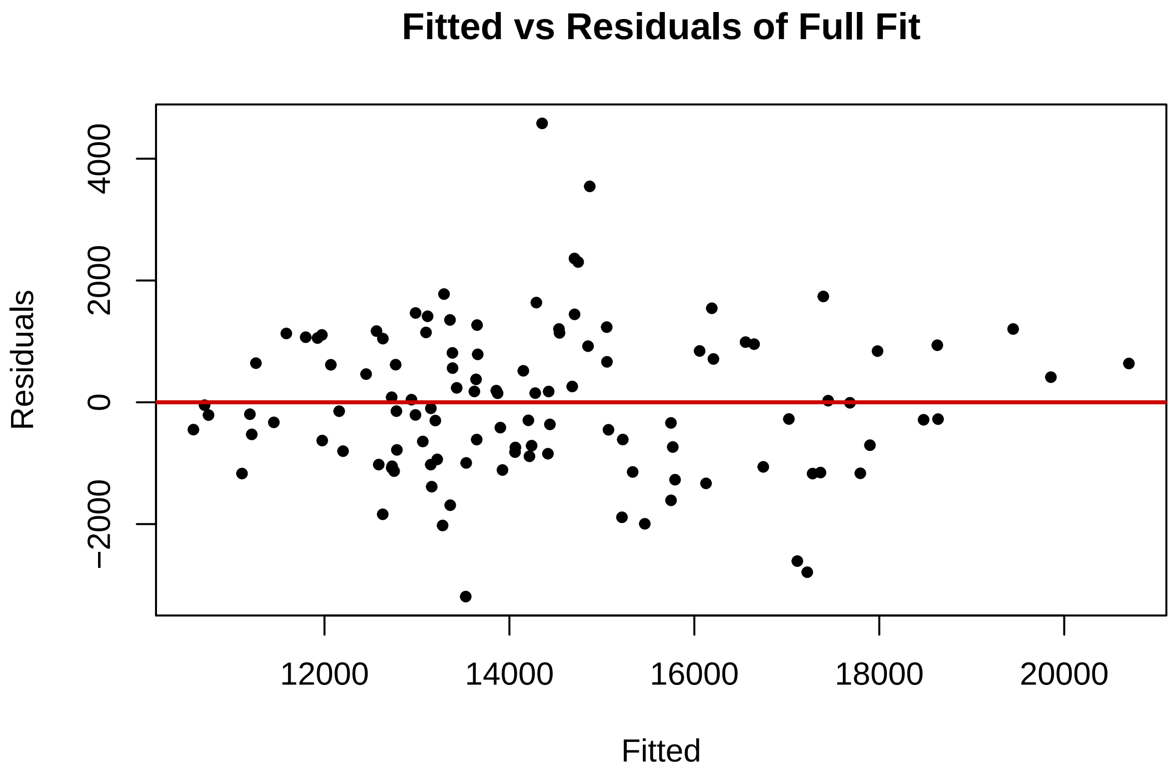
```
#We can create the seasonal plot by using the plot function of our data, and our fitted values using th
#line function.
plot(ts_data, xlab="Month", lwd=2, col='skyblue3', xlim=c(2009,2019))
lines(season_fit$fit,col="red3",lwd=2)
```



Although it appears most of our seasonal coefficients were statistically insiginficant, there appears to be a good fit between the observations and our seasonal predictors. This would indicate that at least some of the observations for attendance may be due to seasonal factors.

2C. Full Model

```
#Fitting the model with both seasonal coefficients and the quadratic fit from before and plotting the f
full_fit <- tslm(ts_data ~ season + t + I(t^2))
plot(as.vector(fitted(full_fit)),as.vector(residuals(full_fit)), pch = 20, ylab="Residuals",xlab="Fitte
abline(h=0,lwd=2,col = "red3")
```



The plot seems to improved on the quadratic residual vs fitted values plot in that it seems to have spread out the cluster of points in the quadratic plot. The points are still spreaded across zero without any specific patterns.

2D. Full Model Statistics

```
#summary to see statistics of the model
summary(full_fit)
```

```
##
## Call:
## tslm(formula = ts_data ~ season + t + I(t^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3190.6  -824.9  -122.4   862.5  4580.2
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.830e+08  6.486e+07   7.446 3.77e-11 ***
## season2     -4.307e+02  6.133e+02  -0.702  0.48418    
## season3      1.889e+02  6.133e+02   0.308  0.75876    
## season4      1.257e+03  6.134e+02   2.049  0.04315 *  
## season5     -1.937e+03  6.135e+02  -3.158  0.00211 ** 
## season6     -5.810e+02  6.136e+02  -0.947  0.34599    
## season7      3.415e+03  6.137e+02   5.565 2.28e-07 ***
## season8      4.143e+03  6.139e+02   6.750 1.05e-09 ***
## season9     -5.473e+02  5.985e+02  -0.914  0.36274    
## season10    -2.552e+03  5.985e+02  -4.264 4.62e-05 ***
## season11    -1.166e+03  5.986e+02  -1.948  0.05430 .  
## season12     7.673e+01  5.987e+02   0.128  0.89829    
## t           -4.797e+05  6.441e+04  -7.449 3.73e-11 ***
## I(t^2)       1.191e+02  1.599e+01   7.451 3.69e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1301 on 98 degrees of freedom
## Multiple R-squared:  0.7507, Adjusted R-squared:  0.7176 
## F-statistic:  22.7 on 13 and 98 DF,  p-value: < 2.2e-16
```

```
#showing the error metrics
accuracy(full_fit)
```

```
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 3.249781e-14 1216.921 959.5311 -0.715121 6.759969 0.6464013
##                   ACF1
## Training set 0.3703759
```

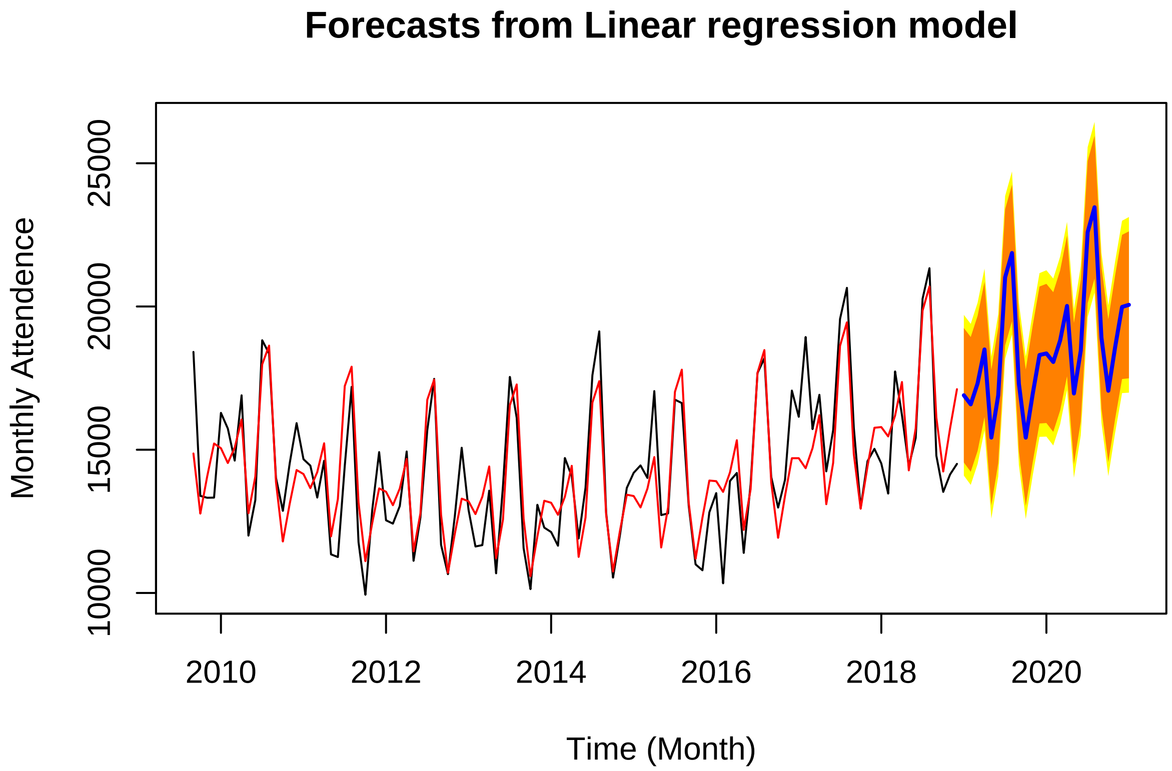
```
accuracy(season_fit)
```

```
##                        ME    RMSE      MAE       MPE     MAPE      MASE
## Training set 1.136868e-13 1594.33 1280.028 -1.239203 9.094446 0.8623085
##                   ACF1
## Training set 0.6281369
```

At a 5% significance level, more seasonal coefficients are statistically significant than just the seasonal model. The adjusted R^2 also went up to 0.7176 which is around 0.2 higher meaning higher percentage of the variations are explaiend. The F-statistics also seems to stay significant. For the error metrics, the model has a MAE of 959.53, and RMSE of 1216.92 which are relatively small considering the data ranges from 9938 to 21337. The MAPE of the full model is also smaller than the the MAPE of the seasonal model meaning the percentage of mean absolute error is lowered by using this full model.

2E. Full Model Forecast

```
full_fit_forecast <- forecast(full_fit, level = c(90,95), newdata = data.frame(t=seq(2019,2021, by = (1
plot(full_fit_forecast,ylab="Monthly Attendence", xlab="Time (Month)", shadecols="oldstyle")
# red line indicate the fitted model
# black line indicate the original data
lines(full_fit_forecast$fitted, col="red")
```



```
# this is the prediction of monthly attendence numbers with 90% and 95% confidence interval
full_fit_forecast
```

```
##          Point Forecast    Lo 90    Hi 90    Lo 95    Hi 95
## Jan 2019       16901.89 14554.48 19249.29 14096.58 19707.19
## Feb 2019       16583.69 14231.90 18935.48 13773.15 19394.24
## Mar 2019       17317.40 14961.01 19673.79 14501.36 20133.44
## Apr 2019       18501.02 16139.83 20862.21 15679.24 21322.80
## May 2019       15424.31 13058.12 17790.51 12596.55 18252.07
## Jun 2019       16899.85 14528.45 19271.25 14065.87 19733.83
## Jul 2019       21016.73 18639.93 23393.53 18176.30 23857.17
## Aug 2019       21867.52 19485.12 24249.92 19020.40 24714.64
## Sep 2019       17300.86 14929.11 19672.60 14466.47 20135.25
## Oct 2019       15421.46 13043.48 17799.43 12579.62 18263.29
## Nov 2019       16935.24 14550.80 19319.67 14085.68 19784.79
## Dec 2019       18307.00 15915.87 20698.13 15449.44 21164.56
## Jan 2020       18360.96 15931.42 20790.50 15457.50 21264.42
## Feb 2020       18062.62 15624.71 20500.53 15149.16 20976.08
## Mar 2020       18816.19 16369.64 21262.73 15892.41 21739.97
## Apr 2020       20019.66 17564.22 22475.10 17085.25 22954.07
## May 2020       16962.81 14498.22 19427.40 14017.47 19908.16
## Jun 2020       18458.20 15984.21 20932.20 15501.62 21414.79
## Jul 2020       22594.94 20111.31 25078.58 19626.84 25563.05
## Aug 2020       23465.59 20972.07 25959.11 20485.67 26445.51
## Sep 2020       18918.78 16435.79 21401.78 15951.44 21886.12
## Oct 2020       17059.24 14565.34 19553.13 14078.87 20039.61
## Nov 2020       18592.87 16087.79 21097.96 15599.14 21586.61
## Dec 2020       19984.49 17467.95 22501.04 16977.06 22991.93
## Jan 2021       20058.31 17492.60 22624.02 16992.11 23124.50
```

Conclusion

The final full model including seasonal dummies and the quadratic fit does seem fit better than having them fit separately. This would suggest the existence of seasonality and trend in the monthly attendance of the PretendCity Children Museum.

In the future, we may create a better model by dropping our statistically insignificant seasonal coefficients. We may want to consider gathering more data and fitting in cyclical components make our model more robust. However, the restraints of this project do not allow the inclusion of cyclical components.

References

Monthly Attendence excel spreadsheet by Alvin Ng with information from PretendCity Children Museum in Irvine, CA.