FHG Case - Predictive Modeling

Eli (Ilya) Bolotin

08/05/2018

In the first part of this project we dealt with incorrect or missing data by computing and retrieving missing values (where possible). Unfortunately, we were still left with missing values that could not be computed or recovered. To deal with these missing values we will use imputation.

Load libraries

```
# load VIM
library(VIM)

# load MICE
library(mice)

# load Amelia
library(Amelia)

# load GGPlot
library(ggplot2)

# load forecasting and time series libraries
library(forecast)
library(tseries)
```

Stage 1: Imputation with MICE and AMELIA

Pre-imputation analysis

Import our dataset for imputation

```
data <- read.csv("dataset_for_imputation.csv", header=TRUE, sep=",")
```

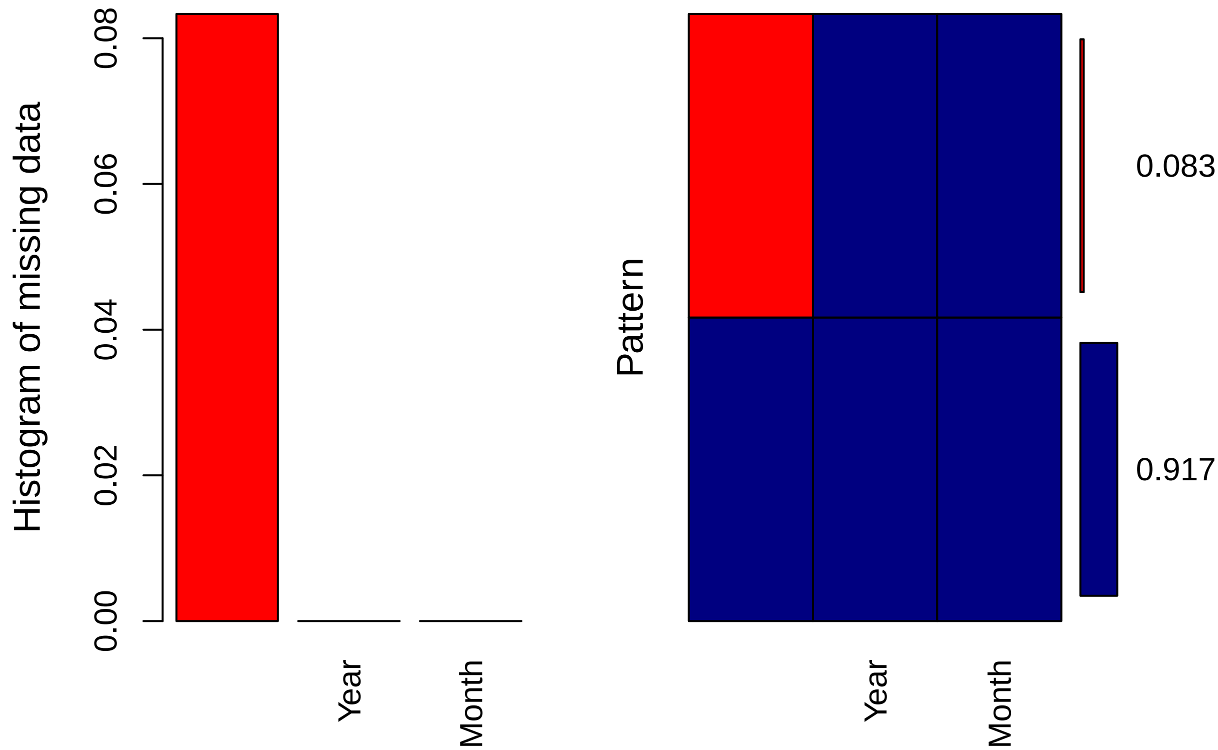
Find out how many NAs we have as percentage of all rows, for every variable

```
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(data,2,pMiss)
```

```
## Incoming.Examinations                  Year                 Month
##              8.333333              0.000000              0.000000
```

Result is 8.33% of observations in the Incoming.Examinations column have NAs. View this in graph form.

```
aggr_plot <- aggr(data, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
↪  labels=names(data), ylab=c("Histogram of missing data","Pattern"))
```



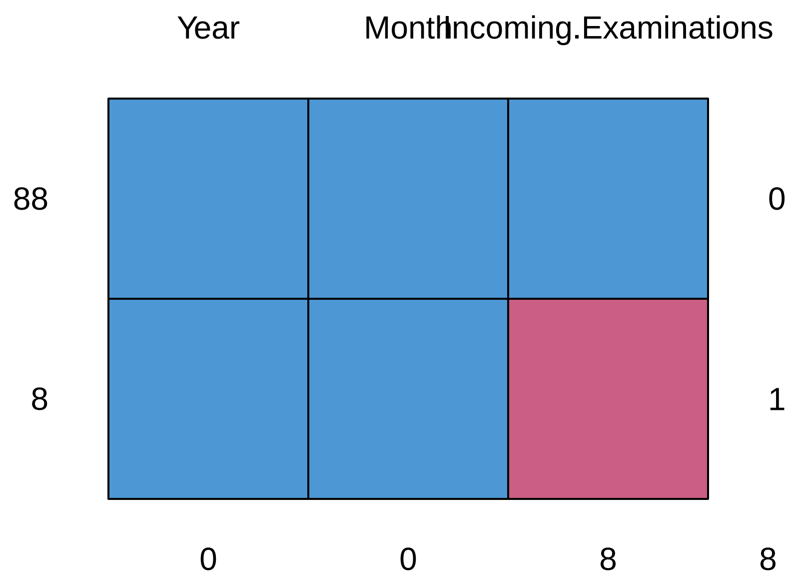

```
##
## Variables sorted by number of missings:
##               Variable      Count
## Incoming.Examinations 0.08333333
##                   Year 0.00000000
##                  Month 0.00000000
```

To deal with these NAs, we will use imputation (the data has been cleaned previously). In this analysis, 2 forms of imputation are tested. The first is MICE imputation and the second is AMELIA.

Imputation with MICE

Let's start by getting a better sense of missing data.

```
md.pattern(data)
```




```
##    Year Month Incoming.Examinations
## 88    1     1                     1 0
## 8     1     1                     0 1
##       0     0                     8 8
```

This tells us there are 8 missing values in the Incoming.Examinations column.

Let's impute the data these missing values.

```
# default method is predictive mean matching
imputed_data <- mice(data, m=10, maxit=10, seed=500, print=F)
```

MICE will generate 10 sets of multiple imputations (with 10 iterations per set). We will:

1. Fit each set with a linear model
2. Then select 3 out of 10 sets
3. Pool the fitted datasets into one
4. Then review summary the summary information.

See below:

```
# fit a linear model to the imputed data
mice_fit <- with(imputed_data, lm(Incoming.Examinations ~ Year + Month))

# review summary of linear model for imputation 1
summary(mice_fit$analyses[[1]])
```

```
##
## Call:
```

```
## lm(formula = Incoming.Examinations ~ Year + Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -915.6 -458.7 -137.5  380.2 1768.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.228e+06  5.155e+04 -23.818  < 2e-16 ***
## Year         6.119e+02  2.566e+01  23.851  < 2e-16 ***
## Month        5.252e+01  1.703e+01   3.084  0.00269 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 576 on 93 degrees of freedom
## Multiple R-squared:  0.8615, Adjusted R-squared:  0.8585
## F-statistic: 289.2 on 2 and 93 DF,  p-value: < 2.2e-16
```

```r
# review summary of linear model for imputation 2
summary(mice_fit$analyses[[2]])
```

```
##
## Call:
## lm(formula = Incoming.Examinations ~ Year + Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -917.9 -459.9 -138.7  375.9 1767.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.226e+06  5.139e+04  -23.86  < 2e-16 ***
## Year         6.109e+02  2.557e+01   23.89  < 2e-16 ***
## Month        5.262e+01  1.697e+01    3.10  0.00256 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 574.1 on 93 degrees of freedom
## Multiple R-squared:  0.8619, Adjusted R-squared:  0.8589
## F-statistic: 290.1 on 2 and 93 DF,  p-value: < 2.2e-16
```

```r
# review summary of linear model for imputation 2
summary(mice_fit$analyses[[3]])
```

```
##
## Call:
## lm(formula = Incoming.Examinations ~ Year + Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -933.2 -440.8 -148.6  377.9 1760.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.222e+06  5.147e+04 -23.739  < 2e-16 ***
```

```
## Year         6.088e+02  2.561e+01  23.771  < 2e-16 ***
## Month        5.592e+01  1.700e+01   3.289  0.00142 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 575 on 93 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.858
## F-statistic: 287.9 on 2 and 93 DF,  p-value: < 2.2e-16
```

```r
# pool the fitted models for all imputed datasets to come up with overall regression
mice_pooled_fit <- pool(mice_fit)

# adjusted R squared
pool.r.squared(mice_fit, adjusted = TRUE)
```

```
##               est     lo 95     hi 95 fmi
## adj R^2 0.8579779 0.7942441 0.9031312 NaN
```

```r
# summarize pooled linear model
summary(mice_pooled_fit)
```

```
##                  estimate   std.error  statistic       df    p.value
## (Intercept) -1.224184e+06 51633.75773 -23.708984 90.75259 0.00000000
## Year         6.100202e+02    25.69475  23.741046 90.75261 0.00000000
## Month        5.481914e+01    17.10999   3.203925 90.09275 0.00187097
```

Base on the pooled fit, we can now create a cleaned and completed dataset. Let's do that next.

```r
completedDataMice <- complete(imputed_data)
completedDataMice
```

```
##    Incoming.Examinations Year Month
## 1                    362 2006     1
## 2                    436 2006     2
## 3                    362 2006     3
## 4                    490 2006     4
## 5                    508 2006     5
## 6                    393 2006     6
## 7                    393 2006     7
## 8                    596 2006     8
## 9                    634 2006     9
## 10                   613 2006    10
## 11                   545 2006    11
## 12                   411 2006    12
## 13                   398 2007     1
## 14                   311 2007     2
## 15                   664 2007     3
## 16                   680 2007     4
## 17                   442 2007     5
## 18                   467 2007     6
## 19                   566 2007     7
## 20                   806 2007     8
## 21                   732 2007     9
## 22                   886 2007    10
## 23                   776 2007    11
## 24                   698 2007    12
```

```
## 25                                875 2008      1
## 26                                840 2008      2
## 27                                724 2008      3
## 28                               1115 2008      4
## 29                                997 2008      5
## 30                                775 2008      6
## 31                                886 2008      7
## 32                               1041 2008      8
## 33                               1011 2008      9
## 34                                775 2008     10
## 35                                939 2008     11
## 36                               1004 2008     12
## 37                               1004 2009      1
## 38                               1065 2009      2
## 39                               1263 2009      3
## 40                                962 2009      4
## 41                               1004 2009      5
## 42                               1429 2009      6
## 43                               1205 2009      7
## 44                                890 2009      8
## 45                               1320 2009      9
## 46                               1276 2009     10
## 47                               1757 2009     11
## 48                               2043 2009     12
## 49                               1491 2010      1
## 50                               1595 2010      2
## 51                               1578 2010      3
## 52                               1604 2010      4
## 53                               1758 2010      5
## 54                               1595 2010      6
## 55                               1457 2010      7
## 56                               1607 2010      8
## 57                               1808 2010      9
## 58                               1866 2010     10
## 59                               1934 2010     11
## 60                               2294 2010     12
## 61                               2294 2011      1
## 62                               2334 2011      2
## 63                               1973 2011      3
## 64                               2262 2011      4
## 65                               2259 2011      5
## 66                               2217 2011      6
## 67                               2739 2011      7
## 68                               2772 2011      8
## 69                               3383 2011      9
## 70                               2869 2011     10
## 71                               2239 2011     11
## 72                               2789 2011     12
## 73                               2789 2012      1
## 74                               3455 2012      2
## 75                               2940 2012      3
## 76                               2968 2012      4
## 77                               3466 2012      5
## 78                               3037 2012      6
```

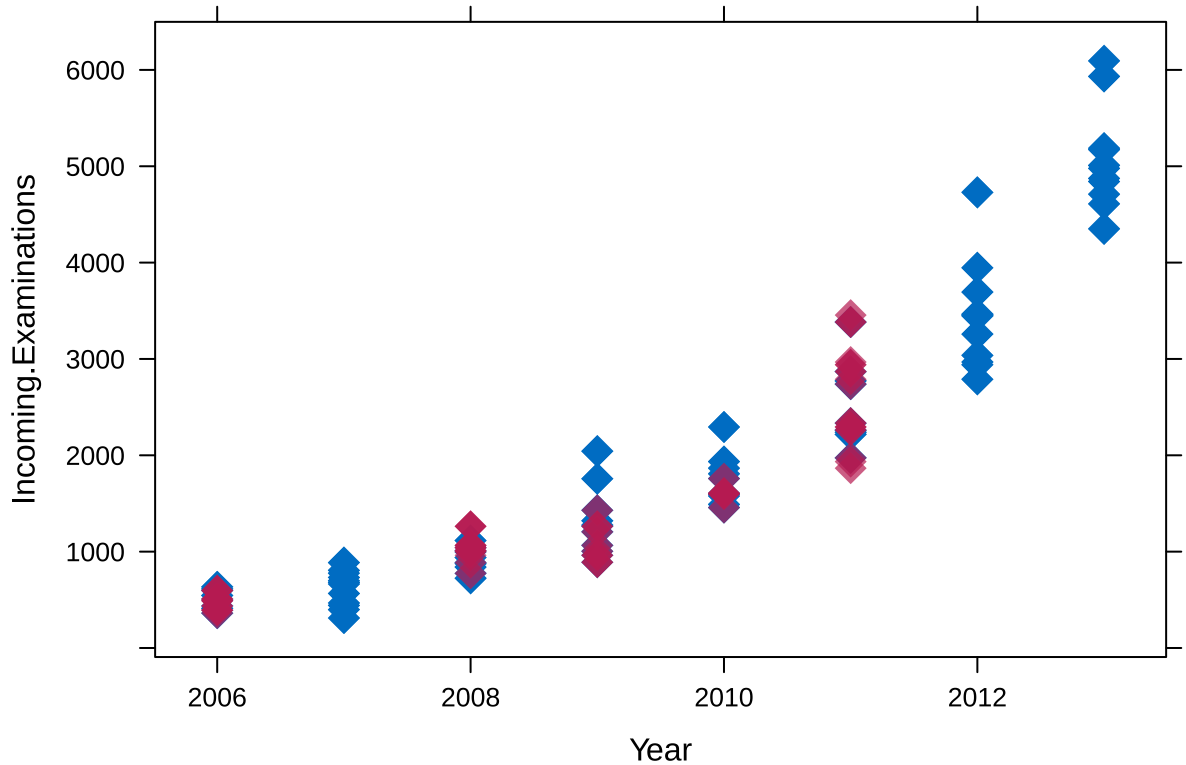
```
## 79                      3946 2012      7
## 80                      3459 2012      8
## 81                      3446 2012      9
## 82                      3258 2012     10
## 83                      4729 2012     11
## 84                      3694 2012     12
## 85                      4610 2013      1
## 86                      4841 2013      2
## 87                      5172 2013      3
## 88                      4351 2013      4
## 89                      5187 2013      5
## 90                      4710 2013      6
## 91                      5010 2013      7
## 92                      4978 2013      8
## 93                      5008 2013      9
## 94                      6094 2013     10
## 95                      4874 2013     11
## 96                      5933 2013     12
```

Create CSV with cleaned, completed data.

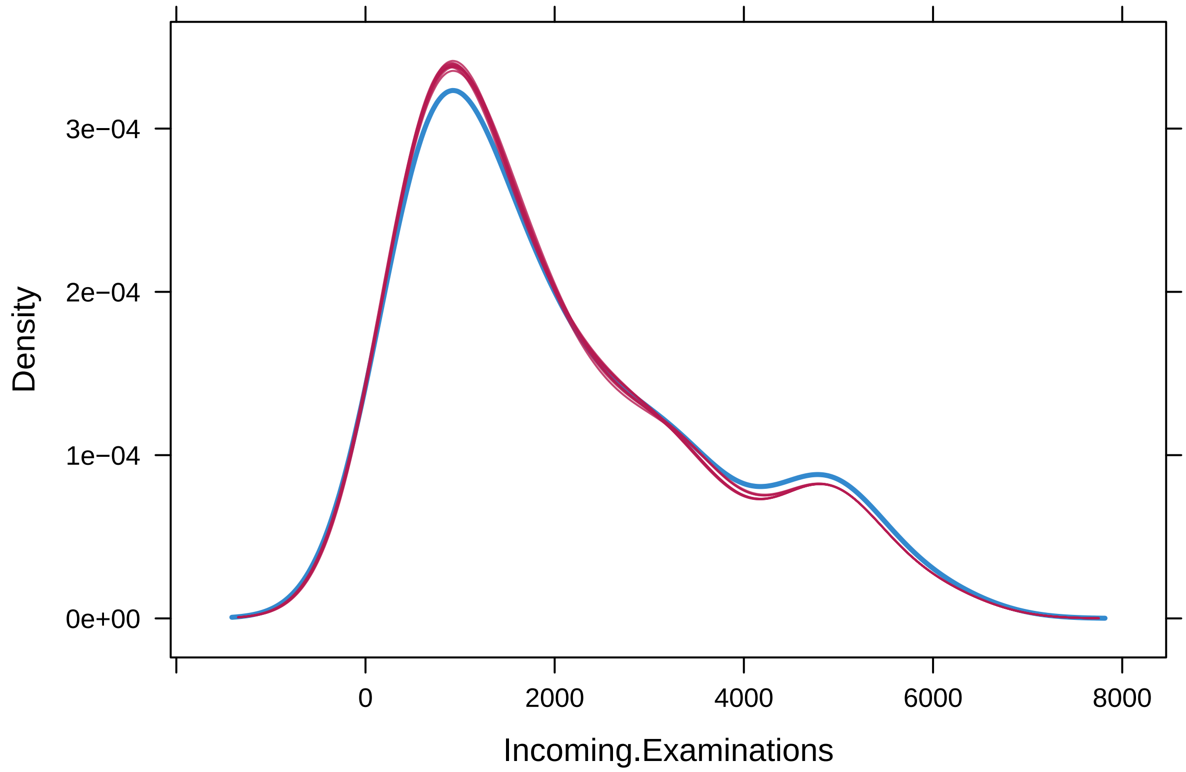
```
write.csv(completedDataMice, "cleaned_dataset.csv", row.names=F)
```

Compare the (cleaned) imputed data to the original data.

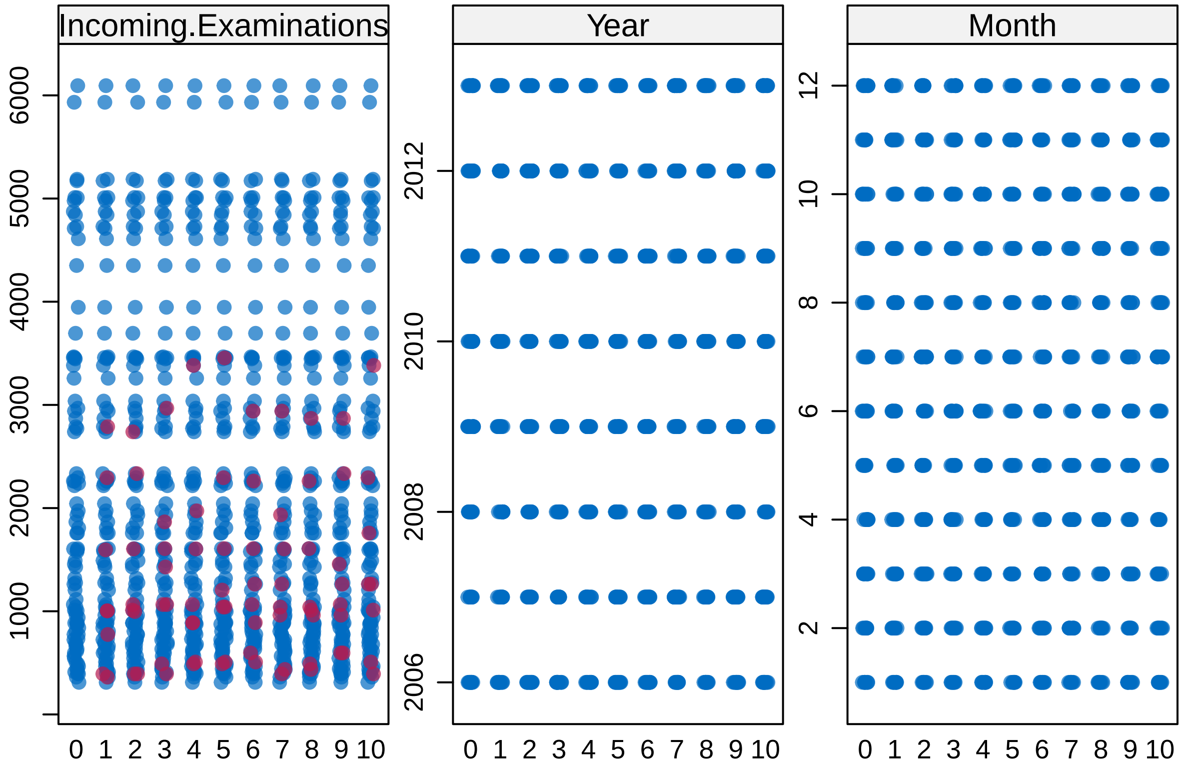
```
# blue points are observed, red are imputed. The overlap tells us that the imputed values
↪  are plausible
xyplot(imputed_data, Incoming.Examinations ~ Year, pch=18,cex=2)
```




```r
# Density plot of the imputed dataset (96 observations)
densityplot(imputed_data, n=96)
```



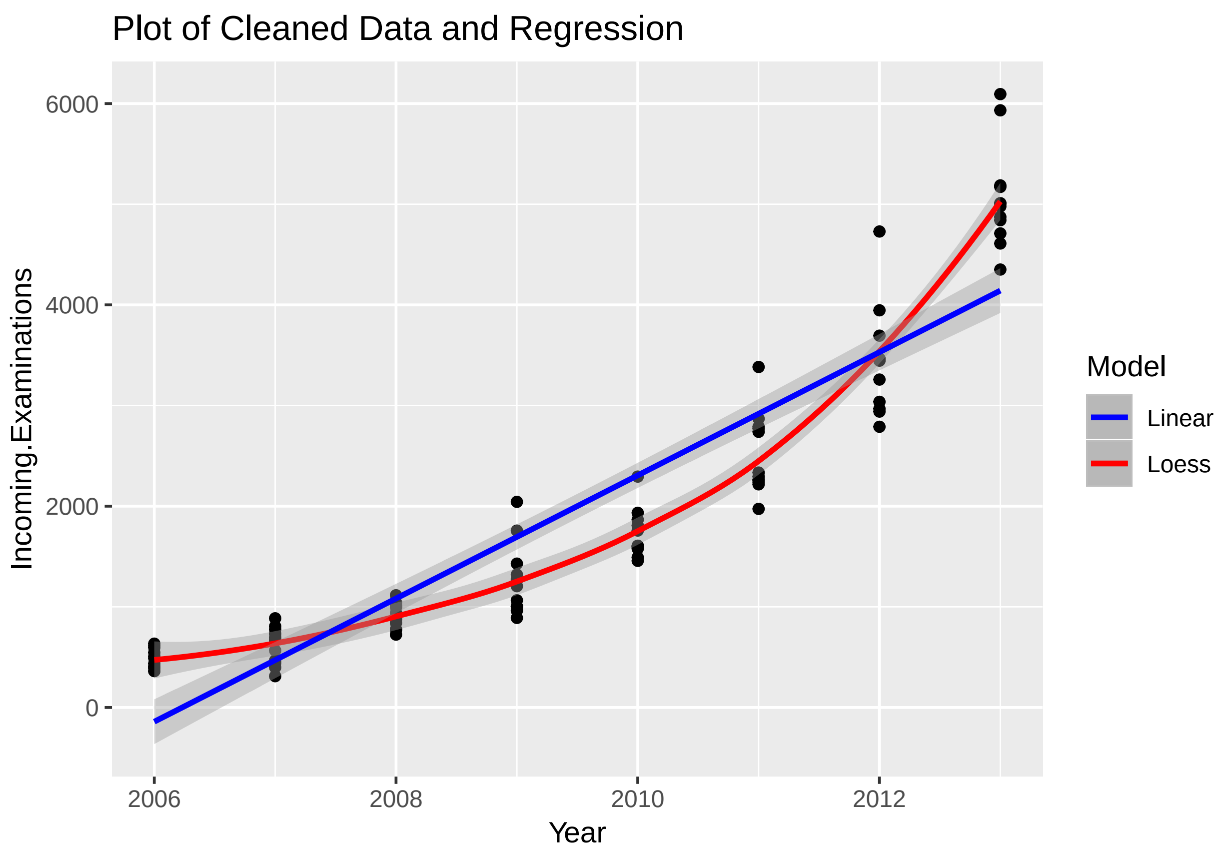

```r
# Distributions of the imputed values for each variable by imputed dataset
stripplot(imputed_data, pch = 20, cex = 1.2)
```



As shown in the charts above, our imputed values are both plausible and follow a similar distribution as the other values in set.

Next, we plot the linear and loess models against our imputed dataset (set 1) to check which fits better.

```
ggplot(completedDataMice, aes(x = Year, y = Incoming.Examinations)) +
  geom_point() +
  geom_smooth(method="loess", aes(colour="Loess")) +
  geom_smooth(method="lm", aes(colour="Linear")) +
  ggtitle("Plot of Cleaned Data and Regression") +
  scale_colour_manual(name="Model", values=c("blue","red"))
```



As you can see, the observations do not follow a linear model. Nonlinear regression would be better suited to fit this data.

Imputation with AMELIA

In this section, we will test another method of value imputation to compare it with MICE imputation above.

Let's import our dataset:

```
data <- read.csv("dataset_for_imputation.csv", header=TRUE, sep=",")
```

Run AMELIA imputation:

```
amelia_imp <- amelia(data, m=10, parallel = "multicore", ts="Year", p2s=0)
```

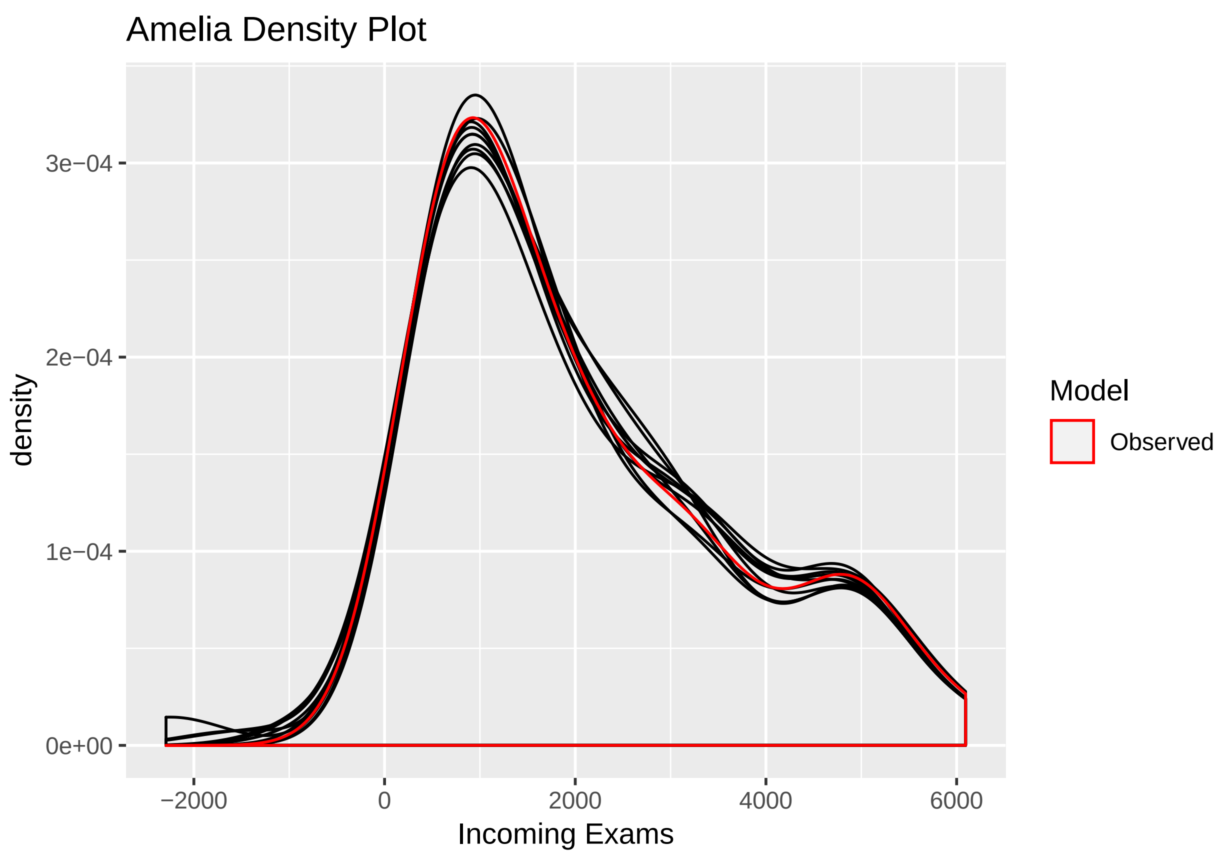
Convert list to dataframe and round values.

```
amelia_imp <- amelia_imp$imputations
amelia_ds <- do.call(rbind.data.frame, amelia_imp)

# round values
amelia_ds <- round(amelia_ds[,], 0)
```

Evaluate the distributions of Amelia imputations

```
  ggplot(amelia_imp[[1]]) +
    geom_density(aes(x=amelia_imp[[1]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[2]]$Incoming.Examinations)) +
```

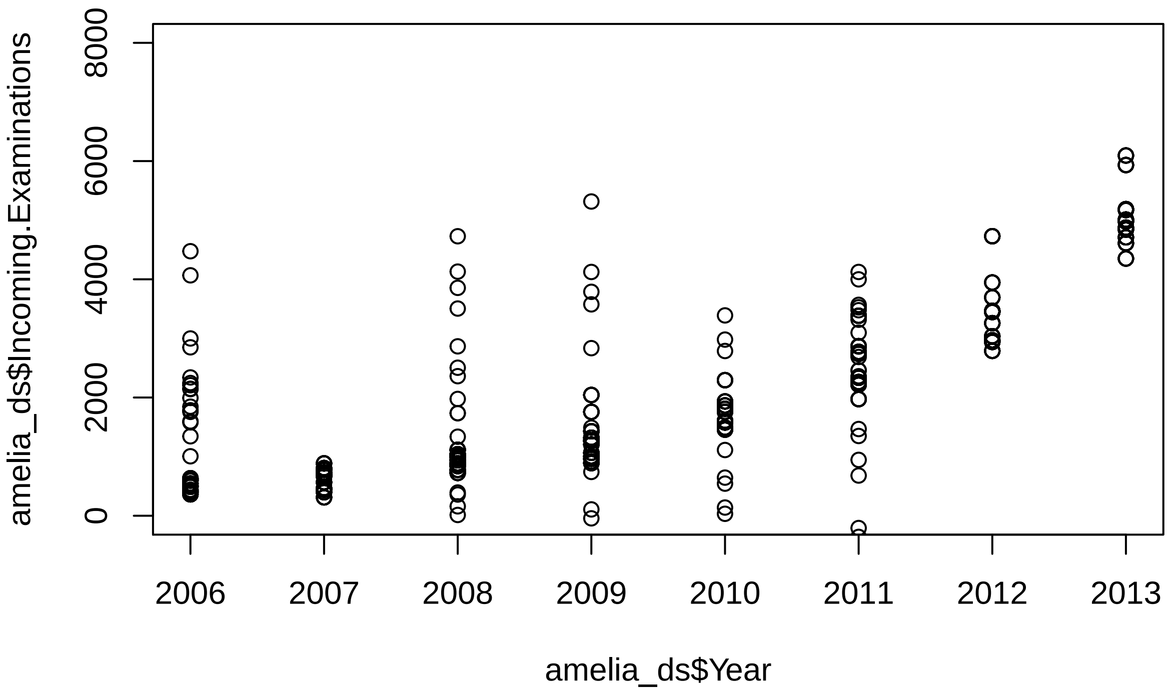
```
    geom_density(aes(x=amelia_imp[[3]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[4]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[5]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[6]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[7]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[8]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[9]]$Incoming.Examinations)) +
    geom_density(aes(x=amelia_imp[[10]]$Incoming.Examinations)) +
    geom_density(aes(x=data$Incoming.Examinations, col="Observed")) +
    scale_colour_manual(name="Model", values=c("red")) +
    labs(title="Amelia Density Plot", x="Incoming Exams")
```



Clearly, every imputation follows a similar distribution.

Plot our dataset with the (AMELIA) imputed observations.

```
plot(amelia_ds$Incoming.Examinations ~ amelia_ds$Year, ylim=c(0,8000))
```



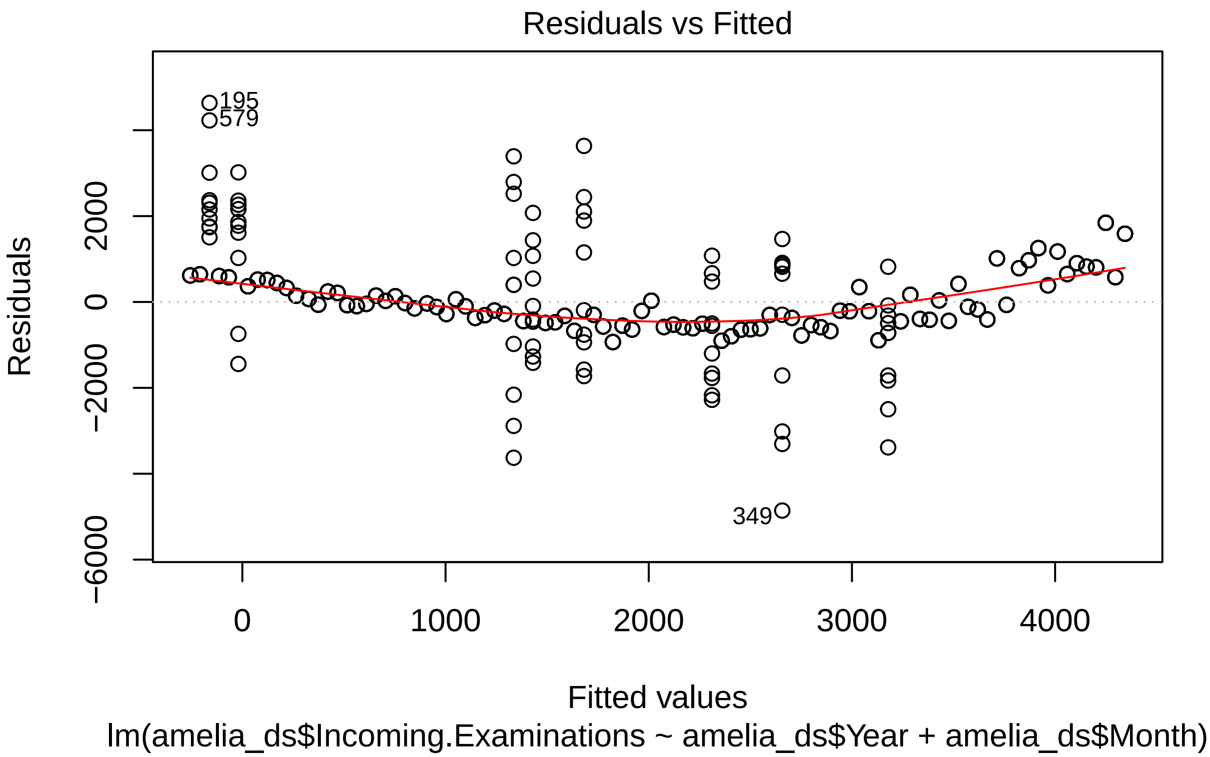
Next, we would like to fit a regression on Amelia imputations, then analyze this regression, and finally, plot regression.

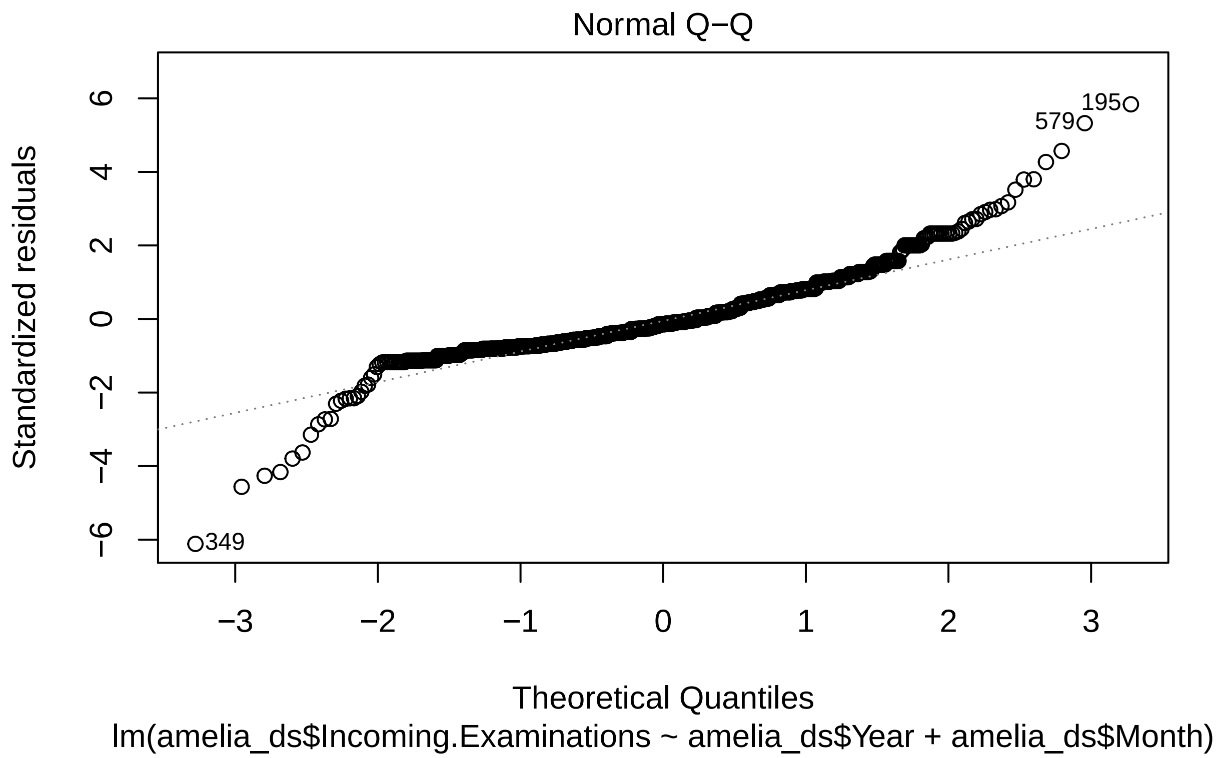
```
# run a linear fit on the first imputed Amelia dataset
amelia_fit <- with(amelia_ds, lm(amelia_ds$Incoming.Examinations ~ amelia_ds$Year +
↪  amelia_ds$Month)) #
summary(amelia_fit)
```

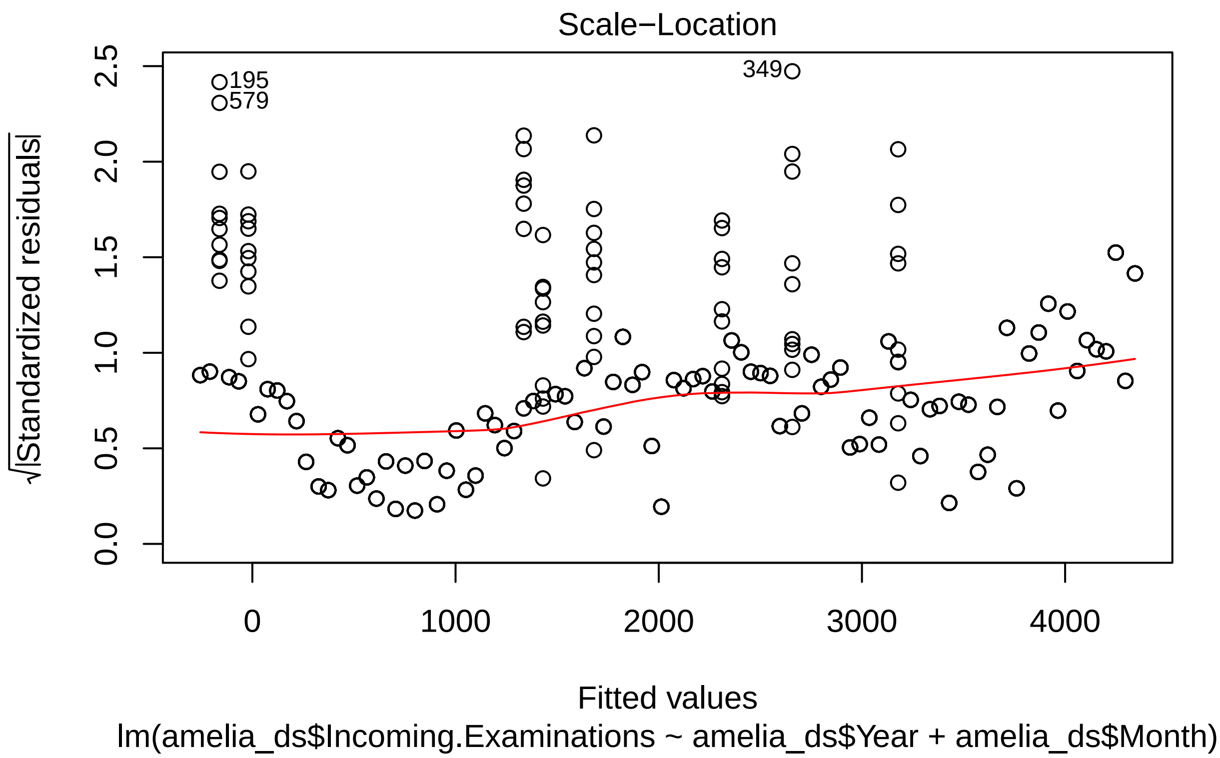
```
##
## Call:
## lm(formula = amelia_ds$Incoming.Examinations ~ amelia_ds$Year +
##     amelia_ds$Month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4857.2  -487.8  -112.6   405.8  4636.6
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.169e+06  2.253e+04 -51.899  < 2e-16 ***
## amelia_ds$Year   5.827e+02  1.121e+01  51.976  < 2e-16 ***
## amelia_ds$Month  4.738e+01  7.441e+00   6.367 2.99e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 795.9 on 957 degrees of freedom
## Multiple R-squared:  0.7413, Adjusted R-squared:  0.7407
```

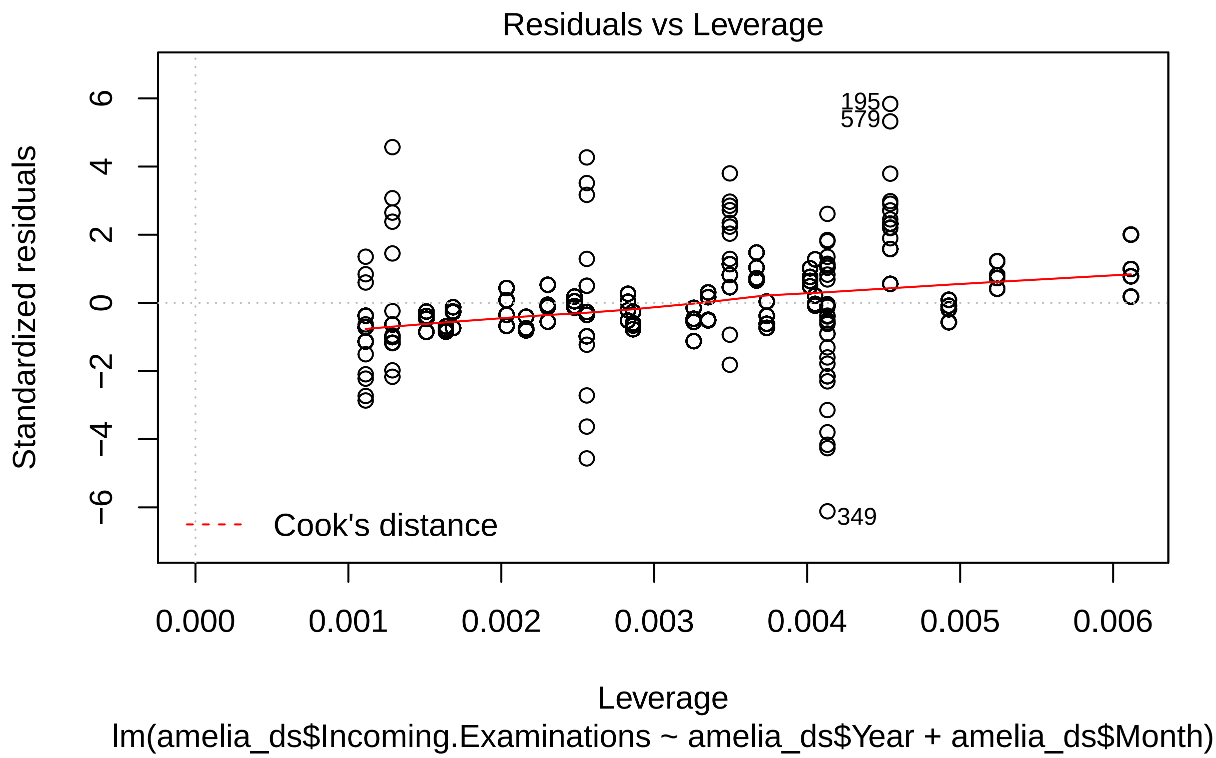
```
## F-statistic:  1371 on 2 and 957 DF,  p-value: < 2.2e-16
```

```r
plot(amelia_fit)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(amelia_ds$Incoming.Examinations ~ amelia_ds$Year + amelia_ds$Month)
195
579
349
−6
−4
−2
0
2
4
6
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
Fitted values
lm(amelia_ds$Incoming.Examinations ~ amelia_ds$Year + amelia_ds$Month)
195
579
349
0.0
0.5
1.0
1.5
2.0
2.5
0
1000
2000
3000
4000



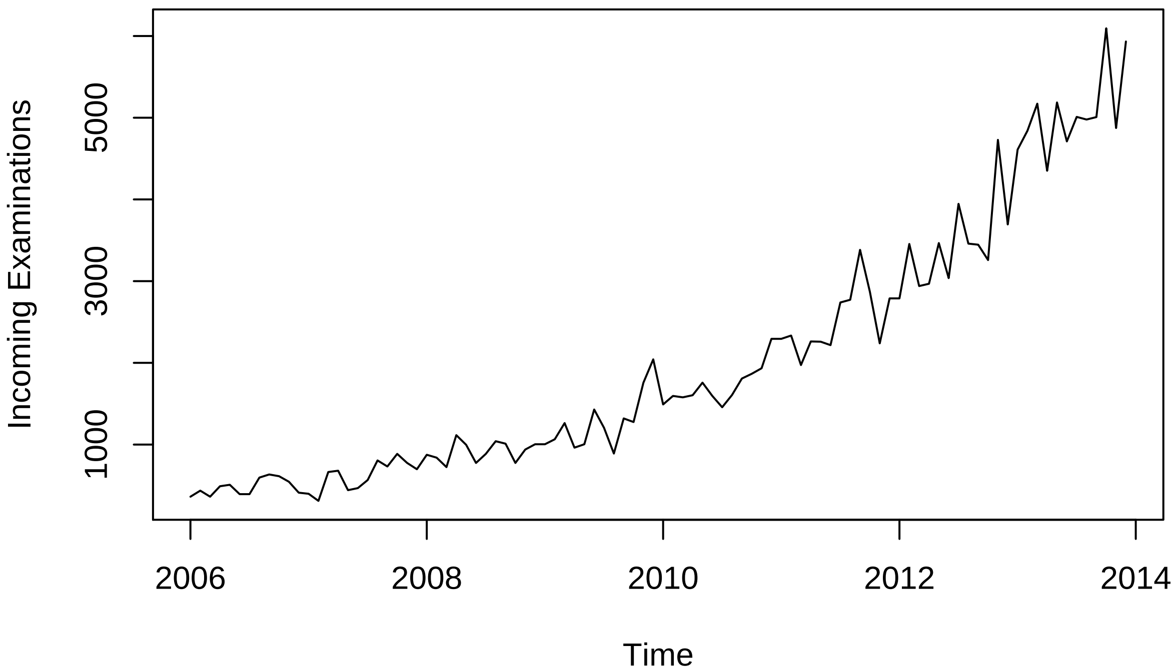
Stage 2: Forecasting

Now that we have dealt with missing values by using imputation, our next goal is to forecast demand of incoming examinations. In this stage, we will use ARIMA and Holt's exponential smoothing to generate predictive models.

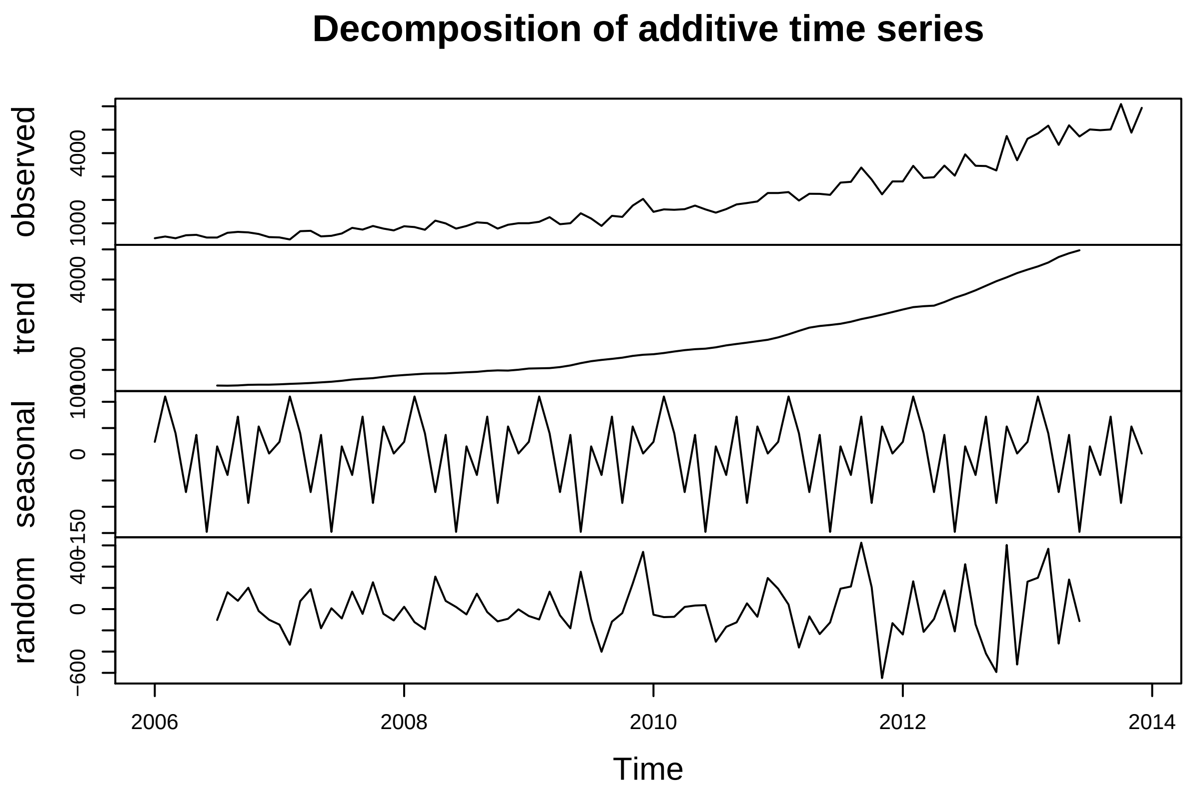
Forecasting with ARIMA (with MICE imputed dataset)

Step 1: review time series and ensure that the time series is stationary.

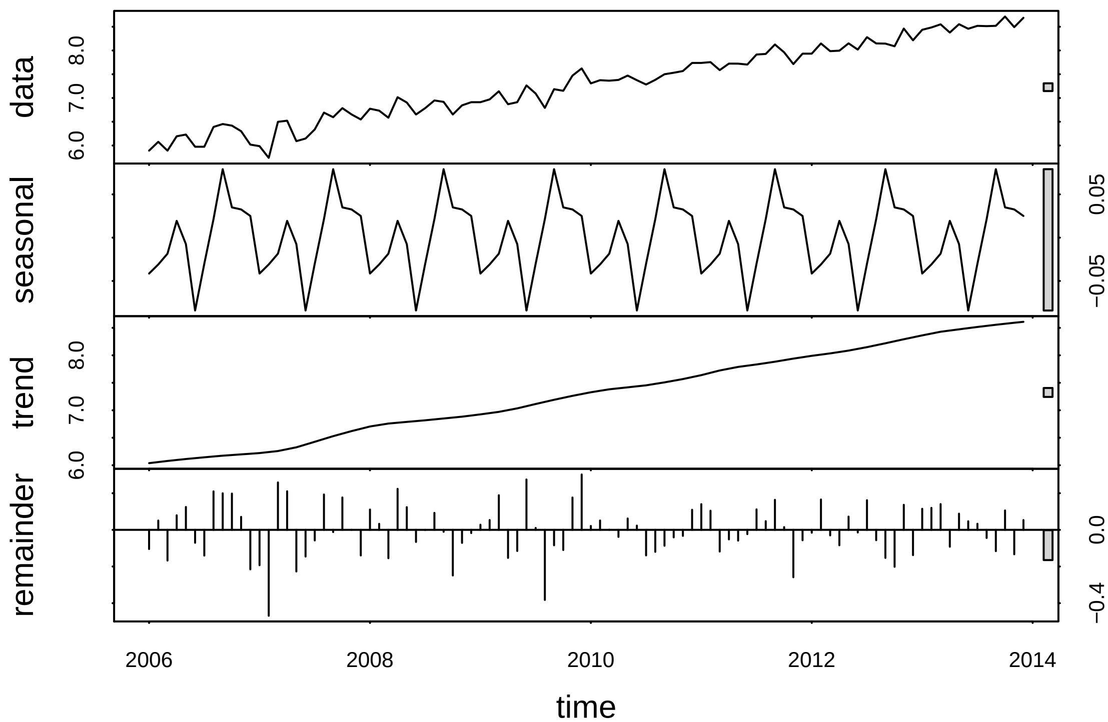
```
# create time series
ts = ts(completedDataMice$Incoming.Examinations, start=c(2006, 1), end=c(2013,12),
↪  frequency=12)

# plot the time series
plot(ts, xlab="Time", ylab="Incoming Examinations")
```



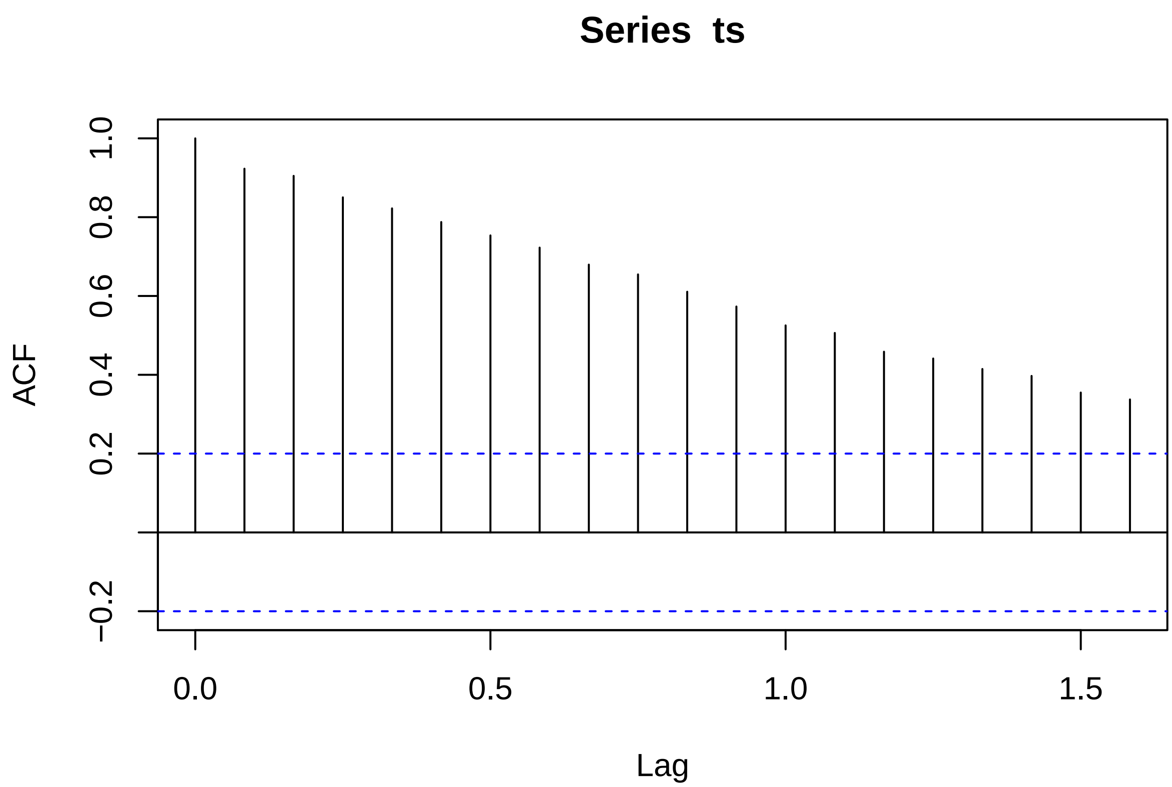

```
# decompose the time series to understand the prevalence of three componenets: trend,
↪  seasonality, and error/irregularity
components.ts = decompose(ts)

# we see that the data is not stationary
plot(components.ts)
```




```
# we can test this another way by using seasonal decomposition in loess smoothing
plot(stl(log(ts), s.window="period"))
```




```r
# autocorrelation function shows data is not stationary. Correlates set of observations
↪  at current time to set of observations at k periods earlier.
acf(ts)
```



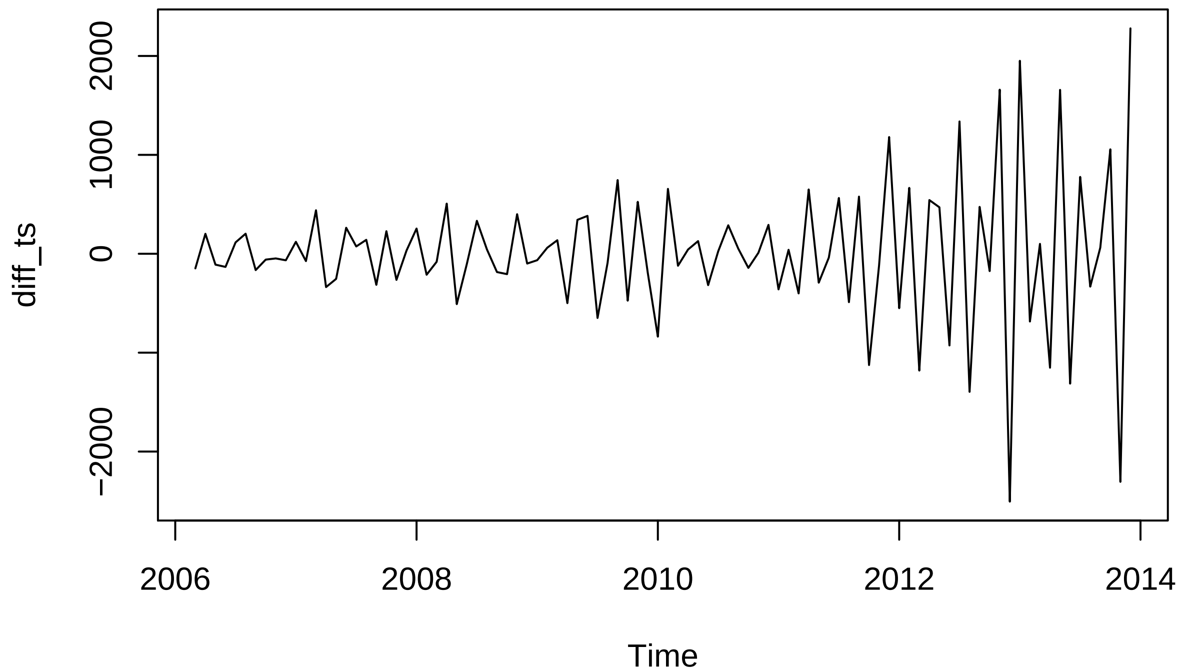

```r
# run ADF test to confirm our hypothesis that data is not stationary
adf.test(ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts
## Dickey-Fuller = -0.31208, Lag order = 4, p-value = 0.9884
## alternative hypothesis: stationary
```

```r
# estimate the number of differences required to make a given time series stationary
ndiffs(ts)
```

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

```r
# differentiate the data to make it stationary. We differentiate it twice to remove what
↪  appears to be a quadratic trend.
diff_ts <- diff(ts, differences=2)

# take a look at the differentiated data
plot(diff_ts)
```

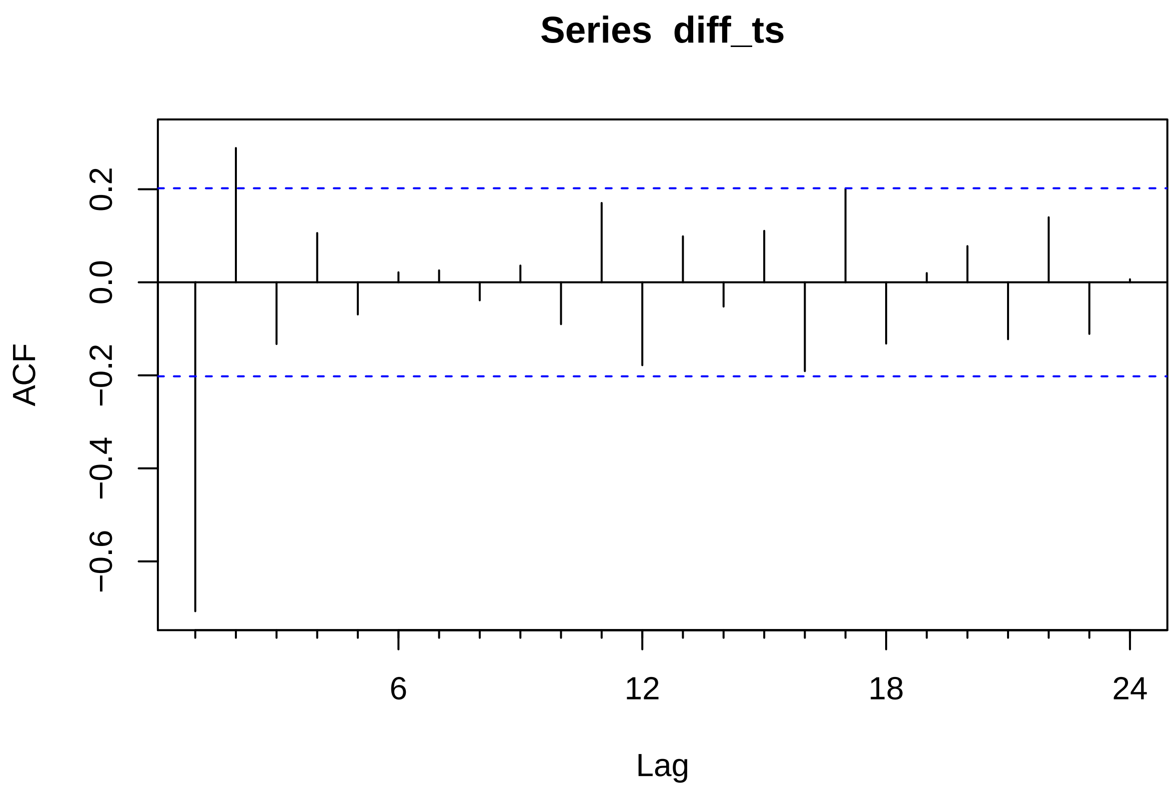



```r
# run ADF to confirm data is now stationary
adf.test(diff_ts)
```

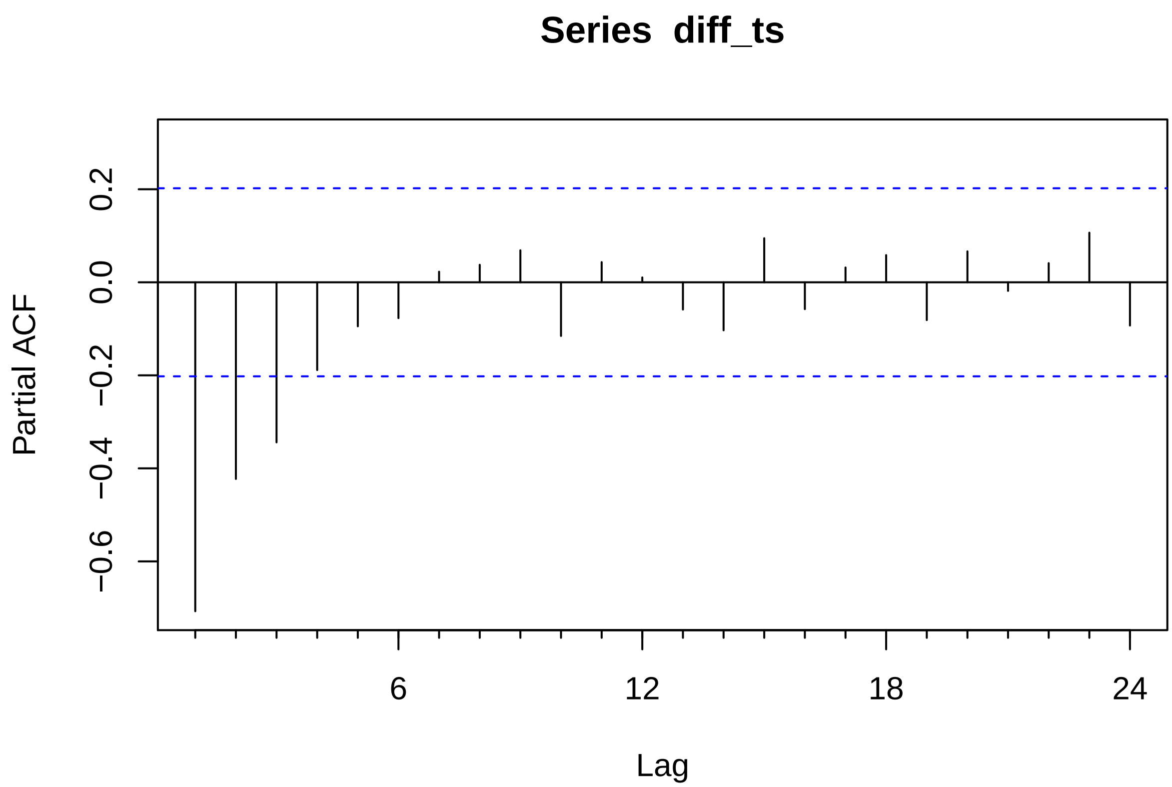
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_ts
## Dickey-Fuller = -8.4792, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Step 2: identify reasonable model or models (possible values of p and q)

```r
# review the ACF chart of the differenced time series. This plots the AC at various lags
↪  for the stationary/differenced time series.
Acf(diff_ts)
```




```
# correlation between an observation at current period and an observation at k periods
↪  earlier with observations between removed.
Pacf(diff_ts)
```



Step 3: fit the model

```
# (p, d, q). d = 2 for nonseasonal differences applied. The lag at which the PACF cuts
↪ off is the indicated number of AR terms (p). The lag at which the ACF cuts off is the
↪ indicated number of MA terms (q).
fit <- Arima(ts, order=c(7,2,6))

# Automated forecasting using an ARIMA model
fitauto <- auto.arima(ts)

# lower S^2, meaning points are closer to the line
summary(fit)
```

```
## Series: ts
## ARIMA(7,2,6)
##
## Coefficients:
##           ar1     ar2    ar3    ar4      ar5      ar6     ar7      ma1
##       -0.5466  0.2136  0.606  0.900  -0.1045  -0.2040  0.1025  -1.3558
## s.e.   0.1277  0.1717  0.176  0.111   0.1452   0.1465  0.1300   0.1674
##           ma2      ma3     ma4     ma5      ma6
##       -0.1624  -0.0394  0.1334  1.3699  -0.9419
## s.e.   0.3626   0.3858  0.3599  0.3970   0.1980
##
## sigma^2 estimated as 86696:  log likelihood=-668.53
## AIC=1365.06   AICc=1370.37   BIC=1400.66
##
```

```
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE     MASE
## Training set 34.12339 270.4629 195.3399 -0.8952236 12.34182 0.291775
##                     ACF1
## Training set -0.03209728
```

```r
summary(fitauto)
```

```
## Series: ts
## ARIMA(1,1,1) with drift
##
## Coefficients:
##           ar1      ma1    drift
##       -0.2949  -0.4919  54.9360
## s.e.   0.1428   0.1282  12.8896
##
## sigma^2 estimated as 103157:  log likelihood=-681.93
## AIC=1371.87   AICc=1372.31   BIC=1382.09
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.6192313 314.4183 222.5433 -7.53879 15.81654 0.3324082
##                      ACF1
## Training set 9.147416e-05
```

```r
# correlation of custom ARIMA fitted values to actual values
cor(fitted(fit),ts)^2
```

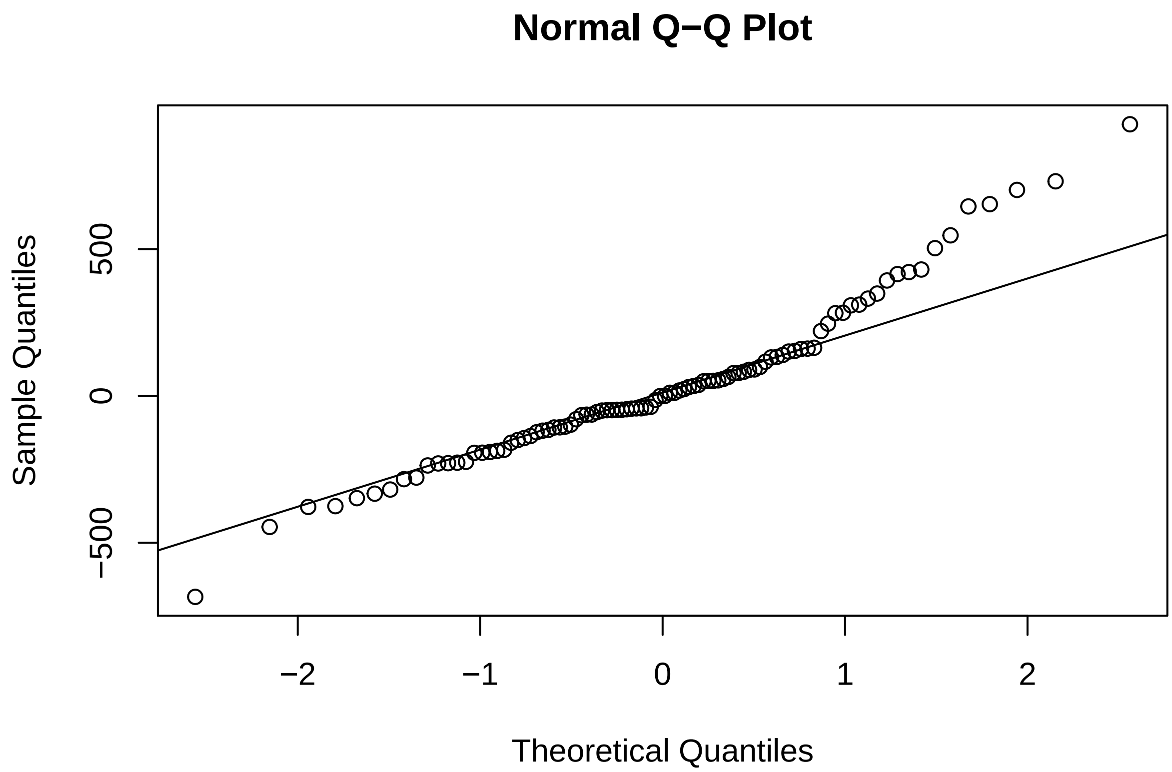
```
## [1] 0.9691572
```

```r
# correlation of custom ARIMA fitted values to actual values
cor(fitted(fitauto),ts)^2
```

```
## [1] 0.9602115
```

Step 4: evaluate the model's fit, accuracy and residuals

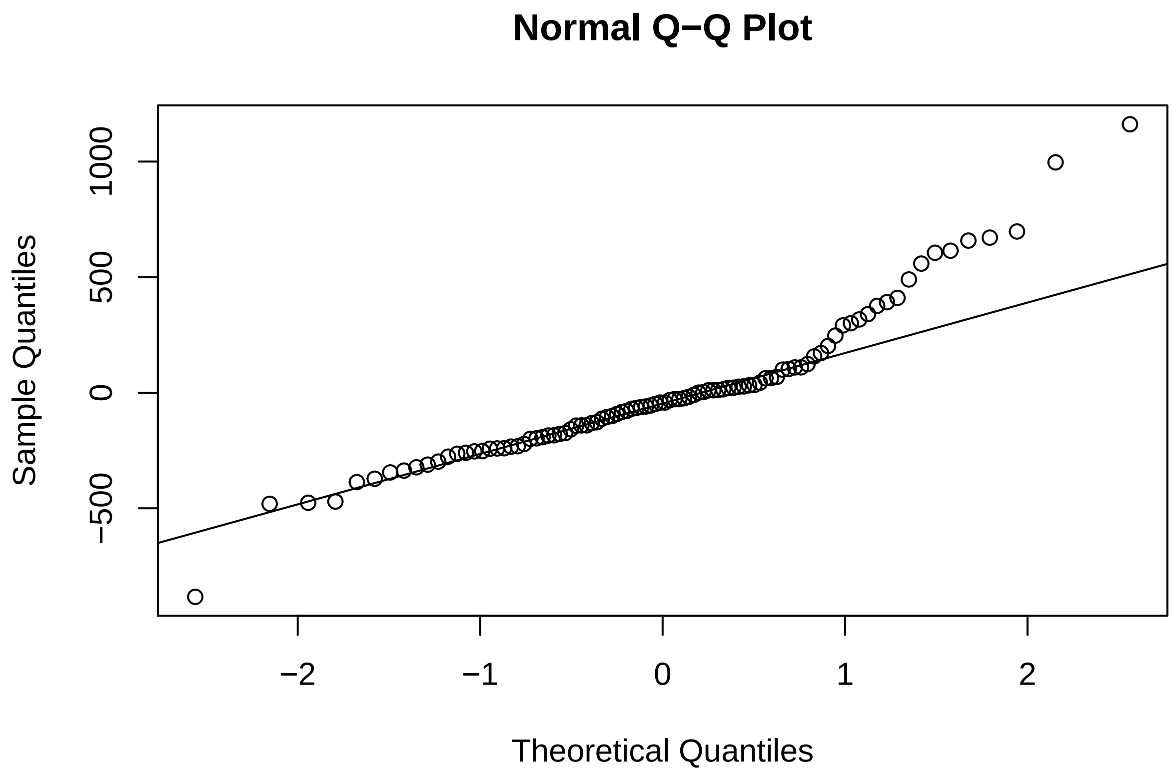
```r
# checking distribution & Ljung-Box for ARIMA
# chart a Q-Q plot to test if the data is normally distributed
qqnorm(fit$residuals)
qqline(fit$residuals) # add line
```




```r
# the results of this test are not significant, suggesting that the autocorrelations
↪  don't differ from zero.
Box.test(fit$residuals, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  fit$residuals
## X-squared = 0.10203, df = 1, p-value = 0.7494
```

```r
# checking distribution & Ljung-Box for auto ARIMA
qqnorm(fitauto$residuals)
qqline(fitauto$residuals)
```




```r
Box.test(fitauto$residuals, type="Ljung-Box") # lowest chi-square suggesting better fit,
↪  and highest p-value suggesting low AC and nonsignificance. But this is due to mean
↪  error.
```

```
##
##  Box-Ljung test
##
## data:  fitauto$residuals
## X-squared = 8.2865e-07, df = 1, p-value = 0.9993
```

```r
# checking accuracy
accuracy(fit) # check accuracy for ARIMA fit.
```

```
##                    ME     RMSE      MAE        MPE     MAPE     MASE
## Training set 34.12339 270.4629 195.3399 -0.8952236 12.34182 0.291775
##                     ACF1
## Training set -0.03209728
```

```r
accuracy(fitauto) # check accuracy for ARIMA auto fit
```

```
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.6192313 314.4183 222.5433 -7.53879 15.81654 0.3324082
##                      ACF1
## Training set 9.147416e-05
```

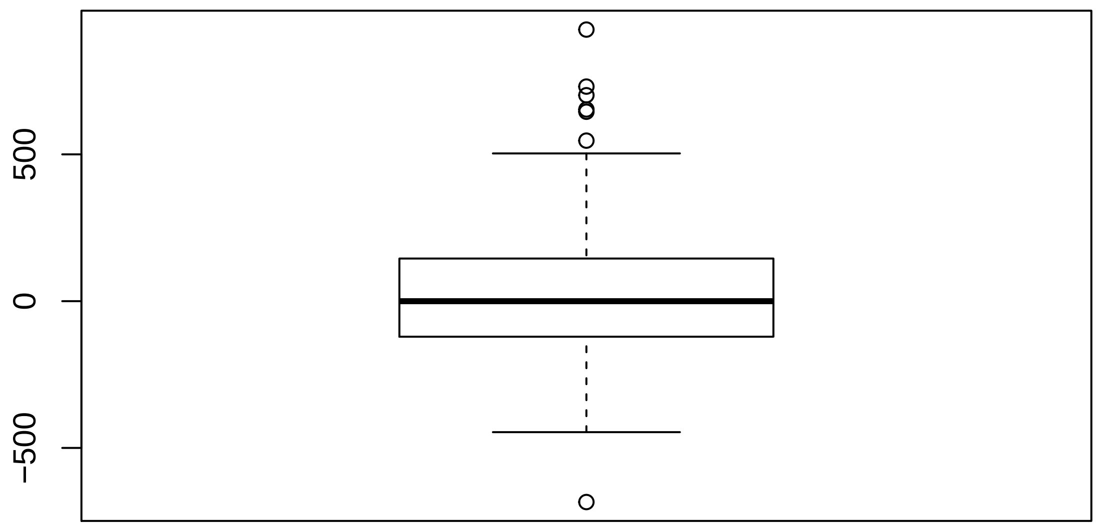
```r
# checking residuals
summary(residuals(fit))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -684.2587 -119.7795   -0.0792   34.1234  142.3442  925.1034
```

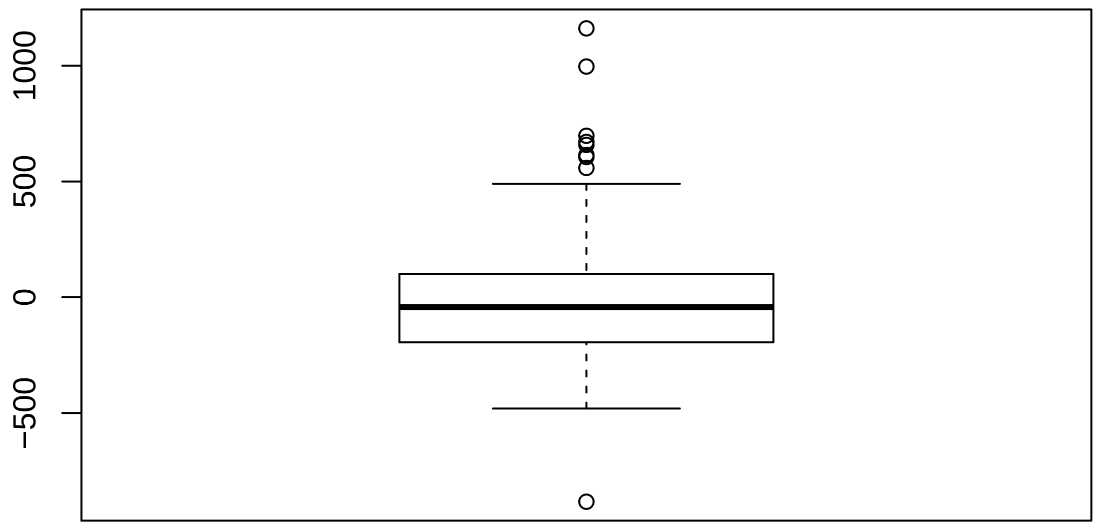
```r
summary(residuals(fitauto))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -883.1740 -193.6513  -42.5549   -0.6192  100.6924 1161.3995
```

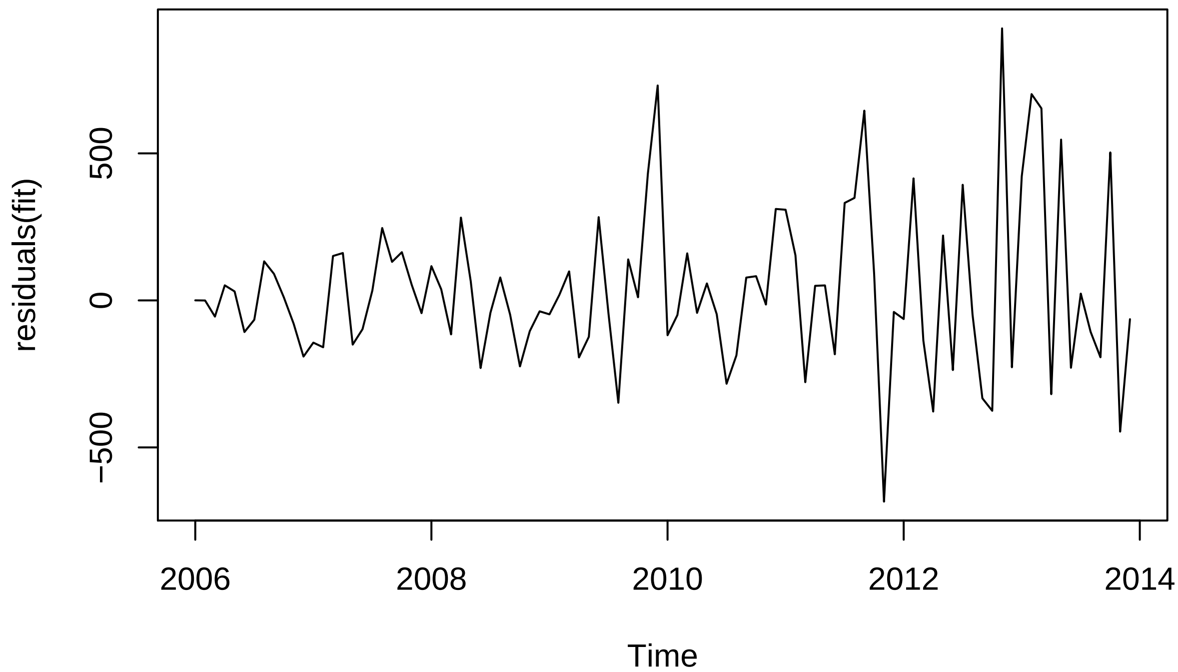
```r
boxplot(residuals(fit))
```



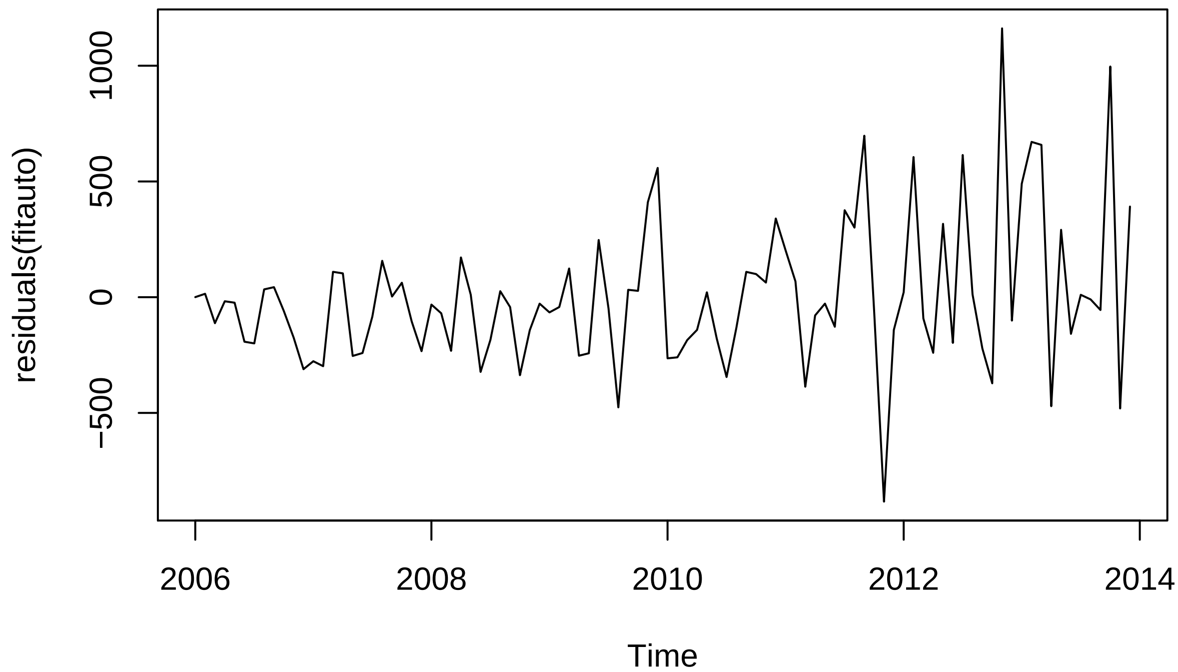

```r
boxplot(residuals(fitauto))
```



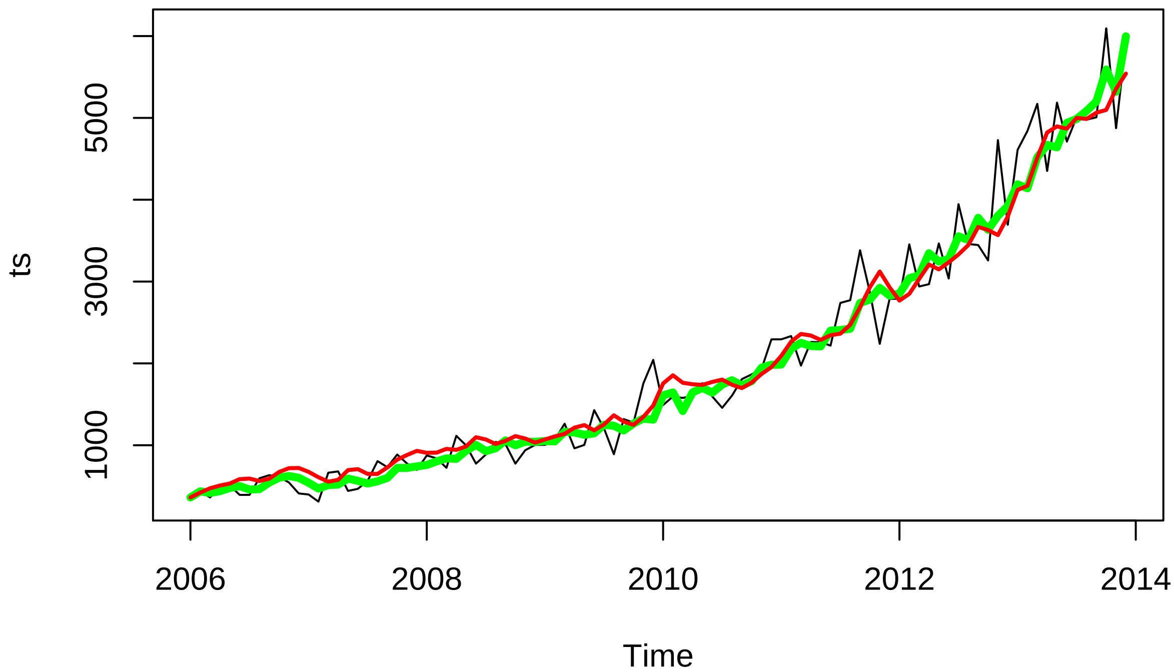

```
plot(residuals(fit))
```




```
plot(residuals(fitauto))
```

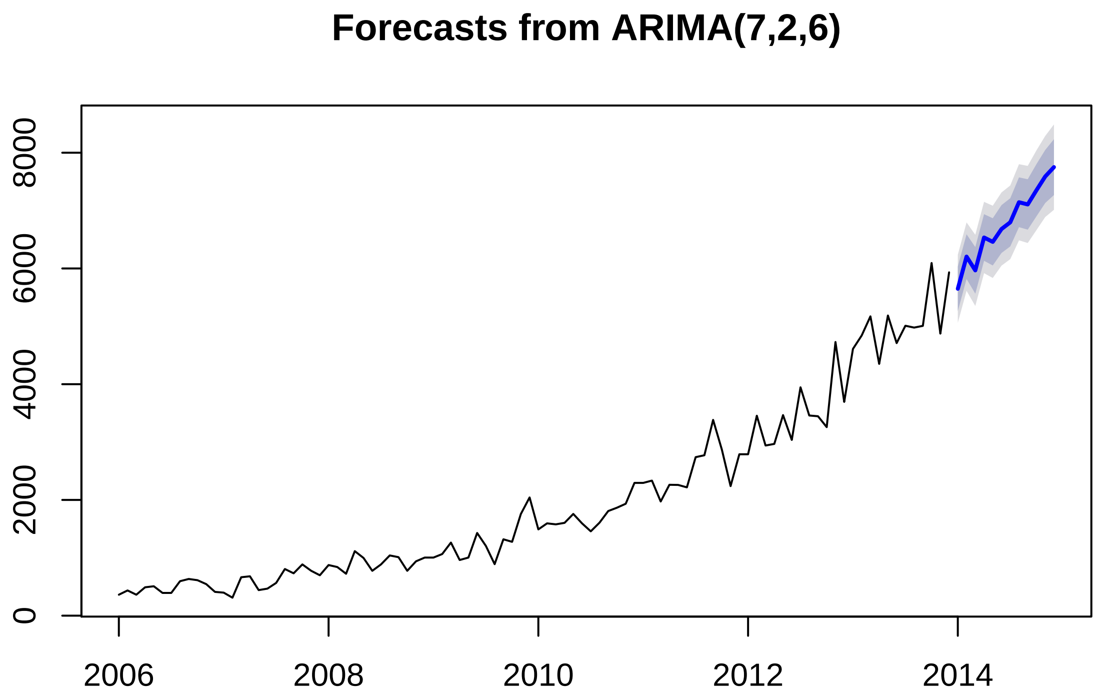



```r
# compare ARIMA models to time series
plot(ts)
lines(fitted(fit), col="green", lwd="4")
lines(fitted(fitauto), col="red", lwd="2")
```

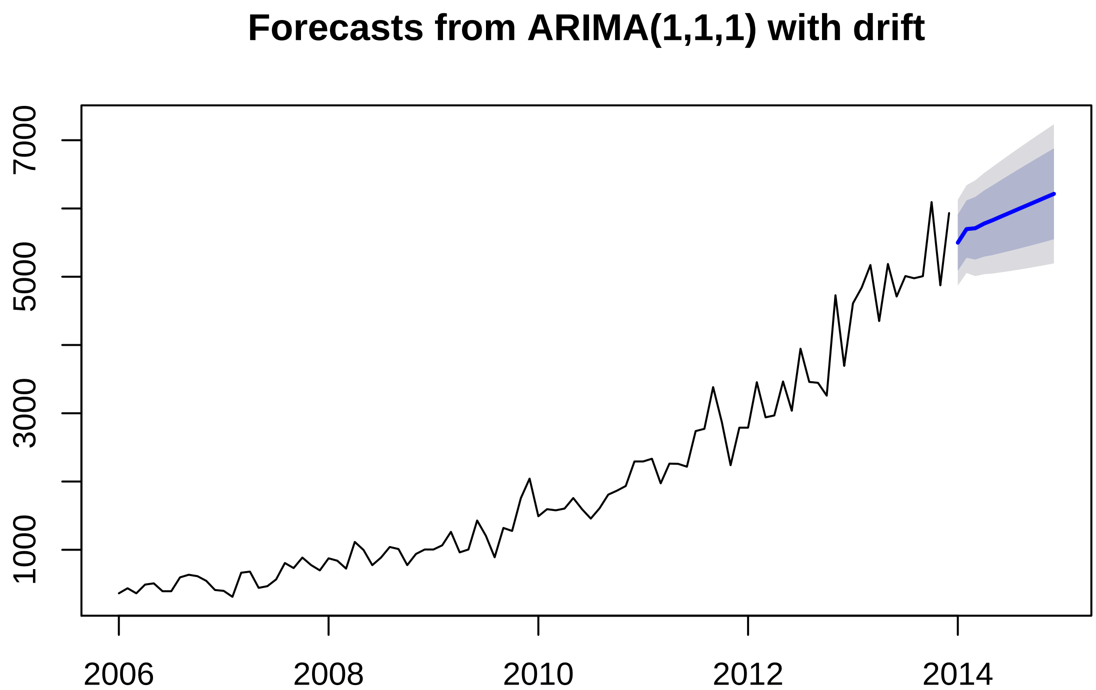


Step 5: make forecasts and show predictions

```
# create forecast for custom ARIMA model
forecast_arima <- forecast(fit, h=12)

 # create forecast for auto ARIMA model
forecast_arimaauto <- forecast(fitauto, h=12)

# plot custom ARIMA forecast
plot(forecast_arima)
```




```
# plot  auto generated forecast
plot(forecast_arimaauto)
```




```
# show forecasted incoming exams for next 12 months using custom ARIMA model
forecast_arima
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2014       5648.928 5264.640 6033.216 5061.211 6236.646
## Feb 2014       6204.625 5818.351 6590.898 5613.870 6795.379
## Mar 2014       5966.686 5564.426 6368.946 5351.482 6581.890
## Apr 2014       6534.957 6132.134 6937.779 5918.892 7151.021
## May 2014       6459.474 6051.289 6867.659 5835.209 7083.738
## Jun 2014       6681.212 6267.628 7094.796 6048.689 7313.735
## Jul 2014       6797.003 6380.862 7213.145 6160.570 7433.437
## Aug 2014       7143.707 6713.717 7573.698 6486.093 7801.322
## Sep 2014       7105.961 6670.395 7541.526 6439.821 7772.100
## Oct 2014       7352.476 6902.050 7802.903 6663.609 8041.344
## Nov 2014       7588.955 7130.822 8047.087 6888.302 8289.607
## Dec 2014       7750.958 7268.747 8233.169 7013.480 8488.436
```

```
# show forecasted incoming exams for next 12 months using the auto ARIMA model
forecast_arimaauto
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2014       5499.183 5087.574 5910.793 4869.681 6128.686
## Feb 2014       5698.272 5277.412 6119.132 5054.622 6341.922
## Mar 2014       5710.692 5251.658 6169.725 5008.660 6412.723
## Apr 2014       5778.167 5293.639 6262.696 5037.145 6519.190
## May 2014       5829.405 5318.063 6340.747 5047.375 6611.435
## Jun 2014       5885.432 5349.355 6421.509 5065.573 6705.290
```

```
## Jul 2014       5940.046 5380.119 6499.974 5083.711 6796.382
## Aug 2014       5995.077 5412.333 6577.821 5103.847 6886.308
## Sep 2014       6049.985 5445.268 6654.703 5125.150 6974.821
## Oct 2014       6104.930 5479.015 6730.844 5147.675 7062.184
## Nov 2014       6159.863 5513.444 6806.282 5171.251 7148.475
## Dec 2014       6214.800 5548.508 6881.092 5195.794 7233.805
```

Forecasting with Holt's (with MICE imputed dataset)

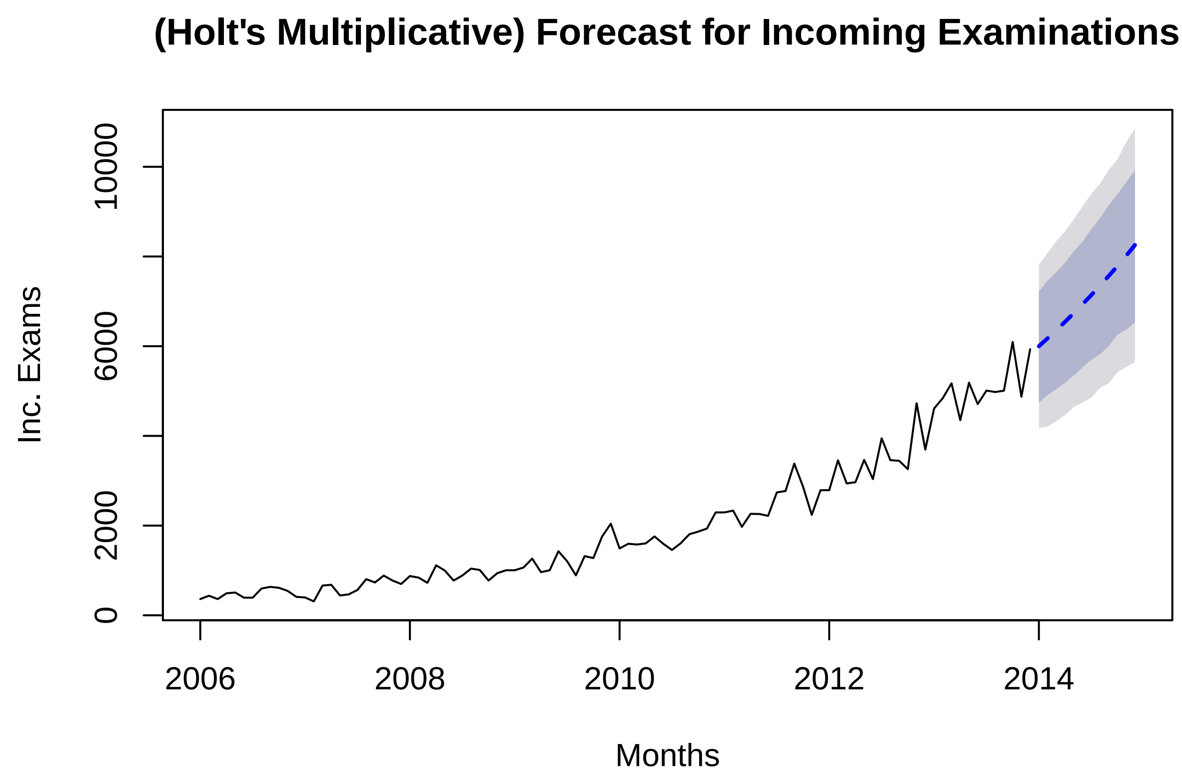
```
# generate time series
ts = ts(completedDataMice$Incoming.Examinations, start=c(2006, 1), end=c(2013,12),
↪  frequency=12)

# Holt's approach but with multiplicative error and multiplicative trend.
fitholts_m <- ets(ts, model="MMN")

# Holt's model (additive)
fitholts_a <- ets(ts, model="AAN")

# check Holt's multiplicative accuracy.
accuracy(fitholts_m)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 9.777434 269.1309 198.3447 -0.629297 12.30327 0.2962633
##                    ACF1
## Training set -0.1297505
```

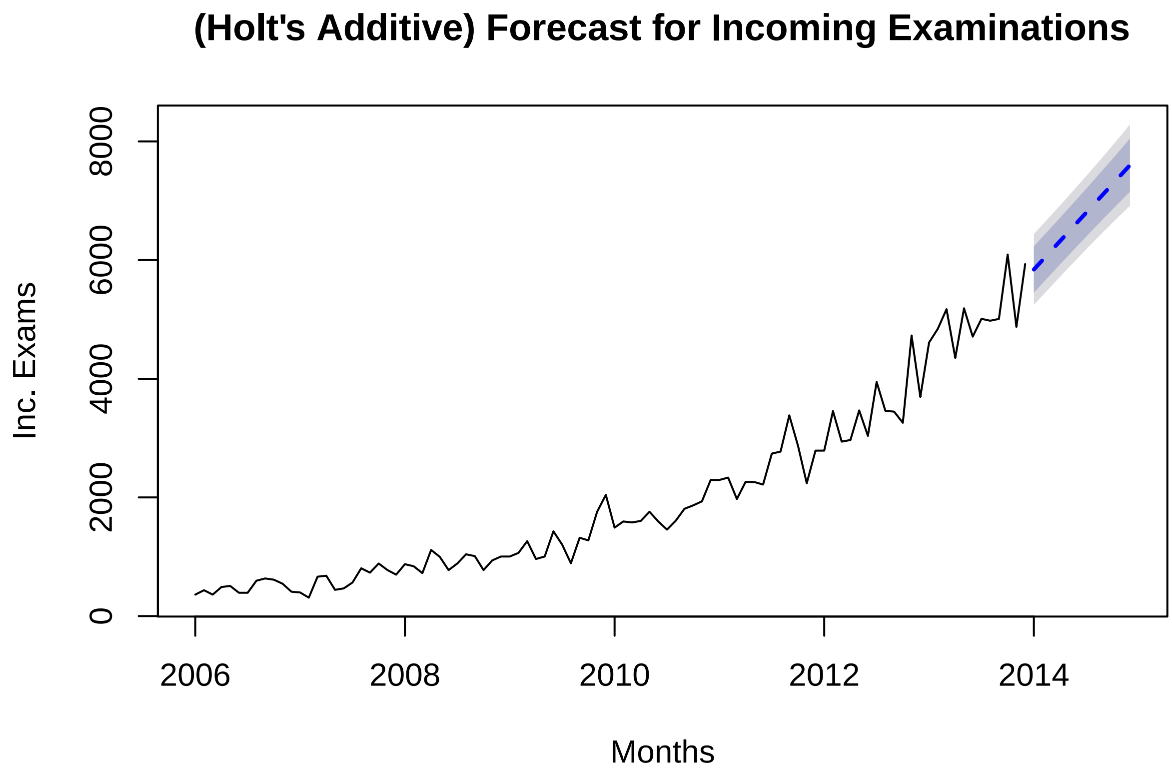
```
# check Holt's additive model accuracy
accuracy(fitholts_a)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 67.84433 298.2579 208.7351 0.2865199 13.20034 0.3117831
##                      ACF1
## Training set -0.009329275
```

```
# create Holts multiplicative forecast
forecast_m <- forecast(fitholts_m, 12)

# create Holts additive forecast for comparison
forecast_a <- forecast(fitholts_a, 12)

# plot Holts multiplicative forecast
plot(forecast_m, main="(Holt's Multiplicative) Forecast for Incoming Examinations",
↪  ylab="Inc. Exams", xlab="Months", flty=2)
```




```r
# plot Holts forecast
plot(forecast_a, main="(Holt's Additive) Forecast for Incoming Examinations", ylab="Inc.
↪  Exams", xlab="Months", flty=2)
```




```r
# show forecasted incoming exams for next 12 months by Holts_m model
forecast_m
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## Jan 2014       5998.765 4745.877 7213.437 4177.846  7824.393
## Feb 2014       6175.419 4911.081 7463.346 4206.967  8074.614
## Mar 2014       6357.274 5044.117 7649.880 4332.045  8342.059
## Apr 2014       6544.485 5183.895 7866.999 4465.023  8563.324
## May 2014       6737.209 5345.225 8114.385 4649.733  8831.467
## Jun 2014       6935.608 5530.809 8330.830 4741.420  9118.043
## Jul 2014       7139.850 5695.271 8609.541 4856.907  9396.467
## Aug 2014       7350.106 5820.520 8856.302 5071.857  9625.589
## Sep 2014       7566.554 5997.112 9143.919 5169.957  9927.934
## Oct 2014       7789.376 6258.686 9387.505 5425.262 10167.111
## Nov 2014       8018.760 6370.831 9655.999 5532.910 10554.243
## Dec 2014       8254.898 6530.365 9922.261 5645.952 10847.938
```

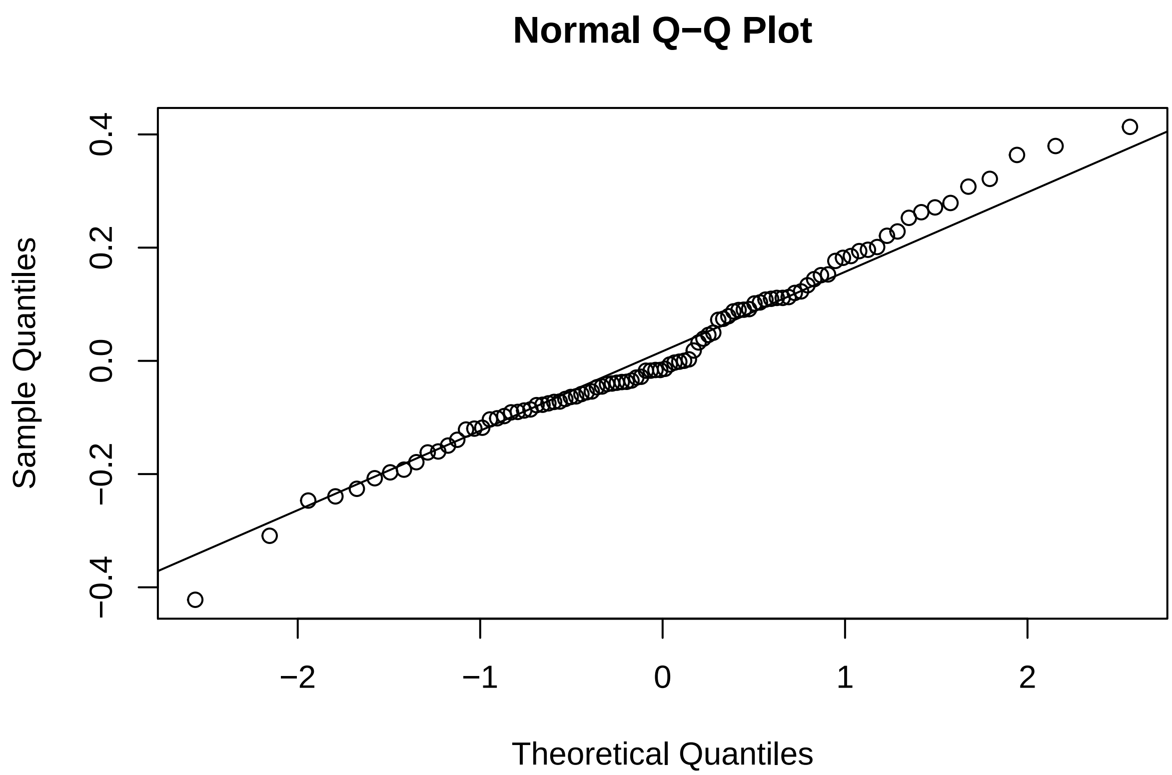
```r
# show forecasted incoming exams for next 12 months by Holts_a model
forecast_a
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2014       5841.241 5450.788 6231.695 5244.094 6438.389
## Feb 2014       6001.130 5610.291 6391.968 5403.394 6598.866
## Mar 2014       6161.018 5769.316 6552.721 5561.961 6760.076
## Apr 2014       6320.907 5927.673 6714.141 5719.507 6922.306
## May 2014       6480.795 6085.181 6876.410 5875.755 7085.835
## Jun 2014       6640.684 6241.666 7039.701 6030.439 7250.928
```

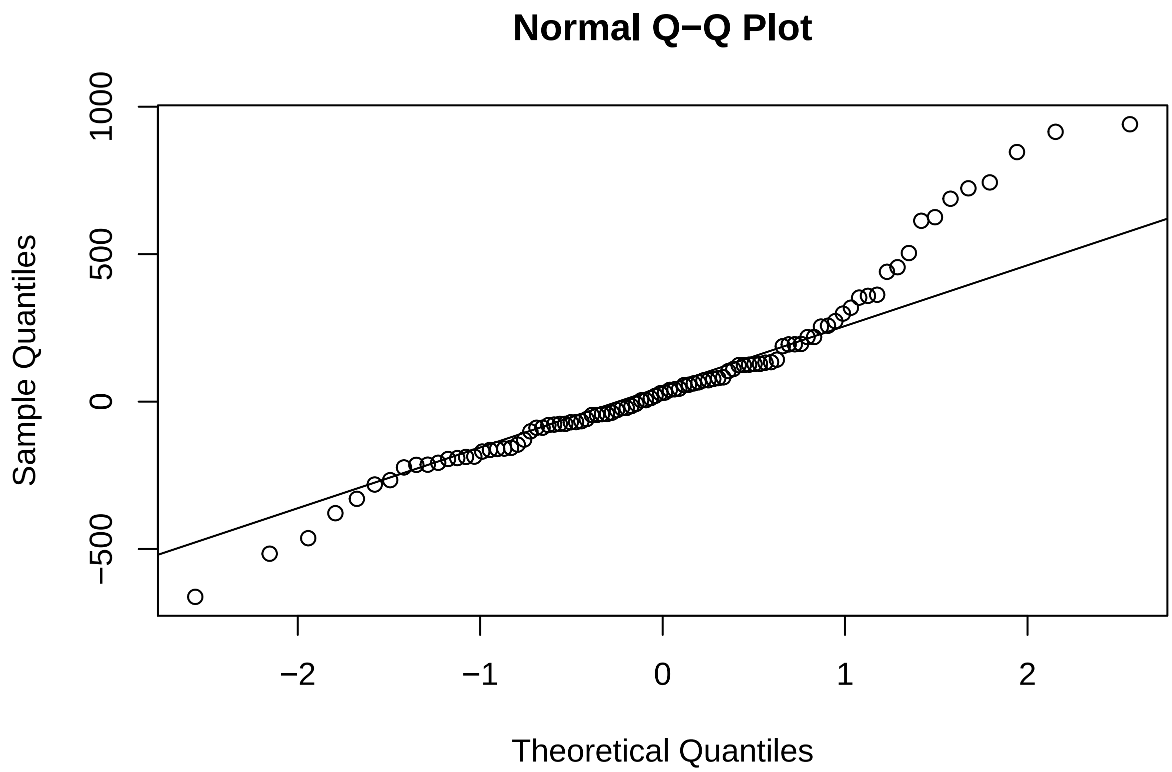
```
## Jul 2014       6800.572 6396.969 7204.175 6183.315 7417.829
## Aug 2014       6960.460 6550.946 7369.975 6334.162 7586.759
## Sep 2014       7120.349 6703.473 7537.225 6482.792 7757.906
## Oct 2014       7280.237 6854.448 7706.026 6629.049 7931.425
## Nov 2014       7440.126 7003.796 7876.455 6772.817 8107.434
## Dec 2014       7600.014 7151.463 8048.565 6914.014 8286.014
```

Check residuals & summaries of Holt's multiplicative and additive models.

```
qqnorm(fitholts_m$residuals)
qqline(fitholts_m$residuals)
```




```
qqnorm(fitholts_a$residuals)
qqline(fitholts_a$residuals)
```




```r
summary(fitholts_m$residuals)
```

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.42203 -0.07770 -0.01506  0.01810  0.11162  0.41333
```

```r
summary(fitholts_a$residuals)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -662.17  -88.55   29.49   67.84  189.29  940.47
```