Group No. 104: Tornado Prediction and Impact Analysis

First Name	Last Name	Monday or Tuesday class
Alisha Anna	Jose	Monday
Archana	Subramaniyan	Monday

Table of Contents

1. Introduction

Natural disasters around the world have always had a serious impact on the people and property. The destruction varies largely from property damage to even many more deaths of people. United States is prone to such serious natural disasters for more than 80 years from the past and is still suffering because of such natural disasters. At present, we have intelligent disaster mitigation plans in place which helps us to plan and manage the resources efficiently in such a way that the loss incurred by any disaster is reduced.

Tornado is one such natural disaster and has had severe bad impacts on the lives of people and on the property and agriculture. Tornado is the most violent of all atmospheric that has had severe impacts on people and property. Around 1200 tornadoes hit the US yearly. Hence, National Oceanic and Atmospheric Administration's National Weather Service Storm Prediction Center generates severe reports and maintains it in its Geographic Information System(GIS) database.

In this project we collected data from the GIS report on tornadoes and examined the occurrence of tornadoes over 60-65 years in the past, apply the different analytical methodologies and deduce the relationship between several factors that would aid in predicting the severity of the tornadoes, property loss and crop loss caused by any tornado in future. This study would not only help to predict the tornadoes severity and loss incurred but also determines if the application of science and technology in disaster management has really reduced the impact of such disasters on people, property and agriculture.

2. Data

The data set belongs to Database of tornado activity from 1950 to 2016 created by NOAA's National Weather Service Storm Prediction Center (data set available at http://www.spc.noaa.gov/gis/svrgis/) to enhance understanding of where tornados happen, indicators of damage, and weather conditions associated with tornados.

Metadata available at http://www.spc.noaa.gov/wcm/data/SPC_severe_database_description.pdf

Reference: https://www.kaggle.com/jtennis/spctornado

The data contains 62208 records.

Attributes:

No	Attribute	Description
1	Om	Tornado number
2	Yr	Year when respective Tornado occurred
3	Mo	Month of the year when respective Tornado occurred
4	Dy	Day of the year when respective Tornado occurred
5	Date	Exact date of tornado
6	Time	Exact time of tornado
7	Tz	Timezone of the region where tornado occurred

8	St	State where tornado occurred
9	Stf	State FIPS number
10	Stn	Monitoring station of tornado
11	Mag	Magnitude of tornado
12	Inj	Injuries because of tornado
13	Fat	Fatalities because of tornado
14	Loss	Estimated Property loss because of tornado in millions of dollars
15	Closs	Estimated Crop loss because of tornado in millions of dollars
16	Slat	Tornado starting latitude in degrees
17	Slon	Tornado starting longitude in degrees
18	Elat	Tornado ending latitude in degrees
19	Elon	Tornado ending longitude in degrees
20	Len	Length of tornado in miles
21	Wid	Width of tornado in yards
22	Fc	Altered or unaltered f-scale rating

3. Problems to be Solved

Below are the list of problems that would be researched in this project using the above mentioned dataset

1) Is the distance travelled by the tornadoes of magnitude 3 and 4 are the same?
2) Is the number of injured people affected by the states Alabama and Texas are the same?
3) Is the average number of fatalities caused by the tornadoes is 4?
4) Is the number of states affected by tornadoes of magnitude 0 to 3 , is 3?
5) Is the property loss incurred by different regions across the US the same or not?
6) Is the no. of injuries due to tornadoes of different magnitudes are the same or not?
7) Is it possible to deduce magnitude category from length, width and distance travelled by a tornado?
8) Is it possible to deduce the property loss category from magnitude category of tornado, region of occurrence and no. of states affected by a tornado?
9) Is it possible to make a prediction on the length of the tornadoes that would occur in future date?

4. Data Processing

The dataset available in the Kaggle site for Tornadoes is not a clean data that can be directly used for analysis and for solving the different problems posted in the section 2 of this document. We had to apply a lot of pre-processing methods to make the data suitable for the analysis and for solving the problems and we have provided the list below

a) Calculated distance travelled by the tornado from start and end latitude-longitude pairs in the data set using distHaversine() function in geosphere package of R.

distHaversine give the shortest distance between latitude-longitude pairs considering spherical feature of earth and ignoring ellipsoidal feature.

b) Calculated average length and width by month for years 1950-2016 using aggregate () function in R.

c) Computed yearly average of length and width, and filled it for the months for which data were missing.

d) Property loss values follow different format over years. Manipulated it to be of the similar format without altering information for ease of analysis.

Prior to 1996, property loss data was categorical ranging from 1 to 9 (1 : <50$, 2: 50-500$, 3: 500-5000$,…..9:>5000,000,000$.

e) Grouped the States into five geographic Regions, MidWest, NorthEast, SouthEast, SouthWest and West for ease of data analysis.

f) Categorical data in numbers replaced by corresponding category labels, eg: Property loss categories (1950-1995), magnitude scale(0-5), etc.

5. Methods and Process

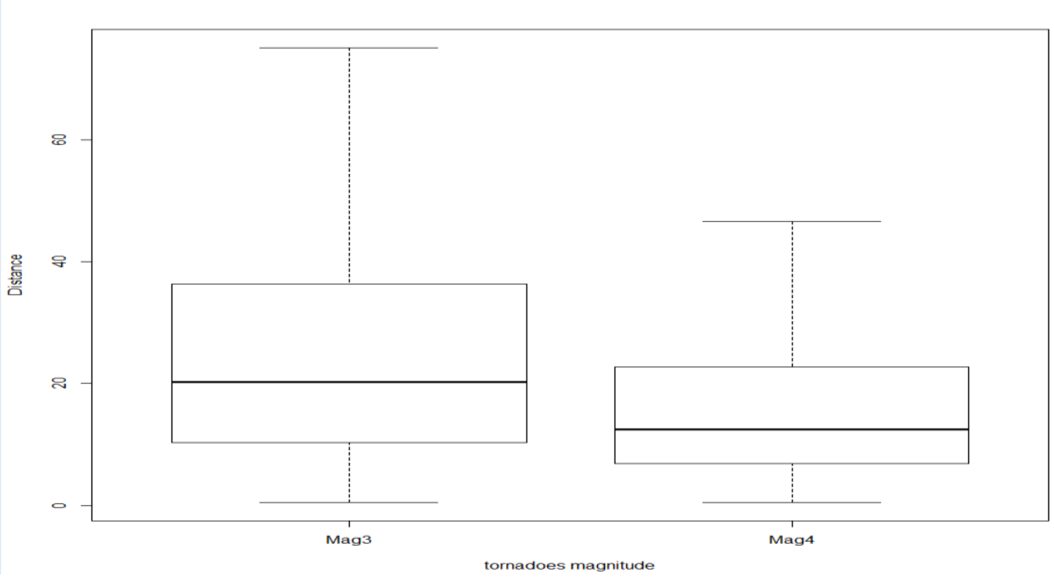
1) Hypothesis #1:

H_0: Distance travelled by tornadoes of magnitude 3 and 4 are the same

H_a: Distance travelled by tornadoes of magnitude 3 is more than that of 4

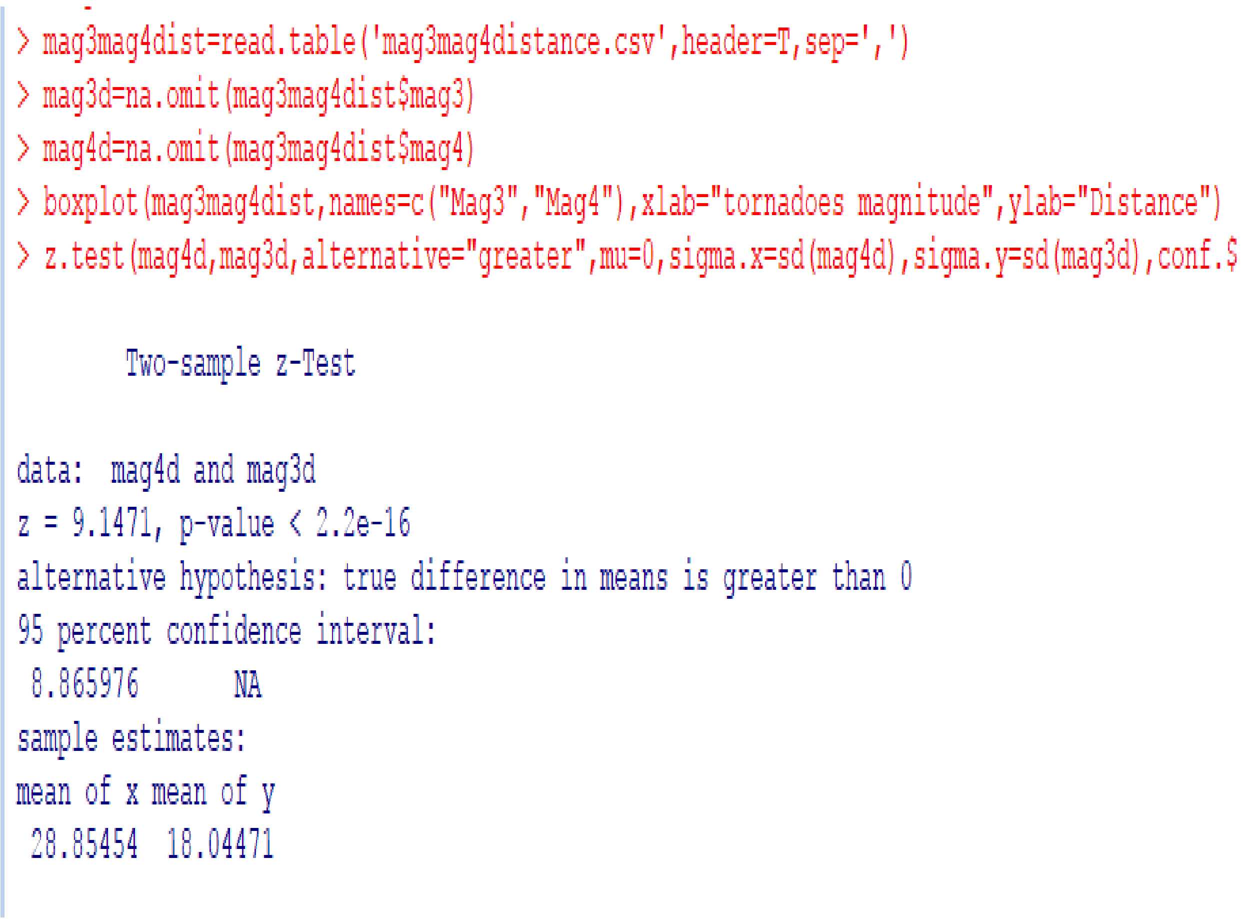
Data used: mag3mag4distance.csv

Box plot showing the distribution of the distance data for magnitude 3 and 4



It is clear from the box plot that magnitude 3 has a lengthier distance travelled than mag4. But we use hypothesis testing to confirm this.

Here, the two samples we used are two independent samples and are of size (2202/655) more than 30. Hence, we perform two sample one tailed hypothesis testing



```
> mag3mag4dist=read.table('mag3mag4distance.csv',header=T,sep=',')
> mag3d=na.omit(mag3mag4dist$mag3)
> mag4d=na.omit(mag3mag4dist$mag4)
> boxplot(mag3mag4dist,names=c("Mag3","Mag4"),xlab="tornadoes magnitude",ylab="Distance")
> z.test(mag4d,mag3d,alternative="greater",mu=0,sigma.x=sd(mag4d),sigma.y=sd(mag3d),conf.$

        Two-sample z-Test

data:  mag4d and mag3d
z = 9.1471, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 8.865976       NA
sample estimates:
mean of x mean of y 
 28.85454  18.04471 
```

From the screenshot, we could see that the p-value is less than 0.05. Hence at 95% confidence level, we have no enough evidence to accept the null hypothesis. Hence, **at 95% confidence level, we conclude that the tornadoes of magnitude 3 travel a longer distance compared to the tornadoes of magnitude 4**.

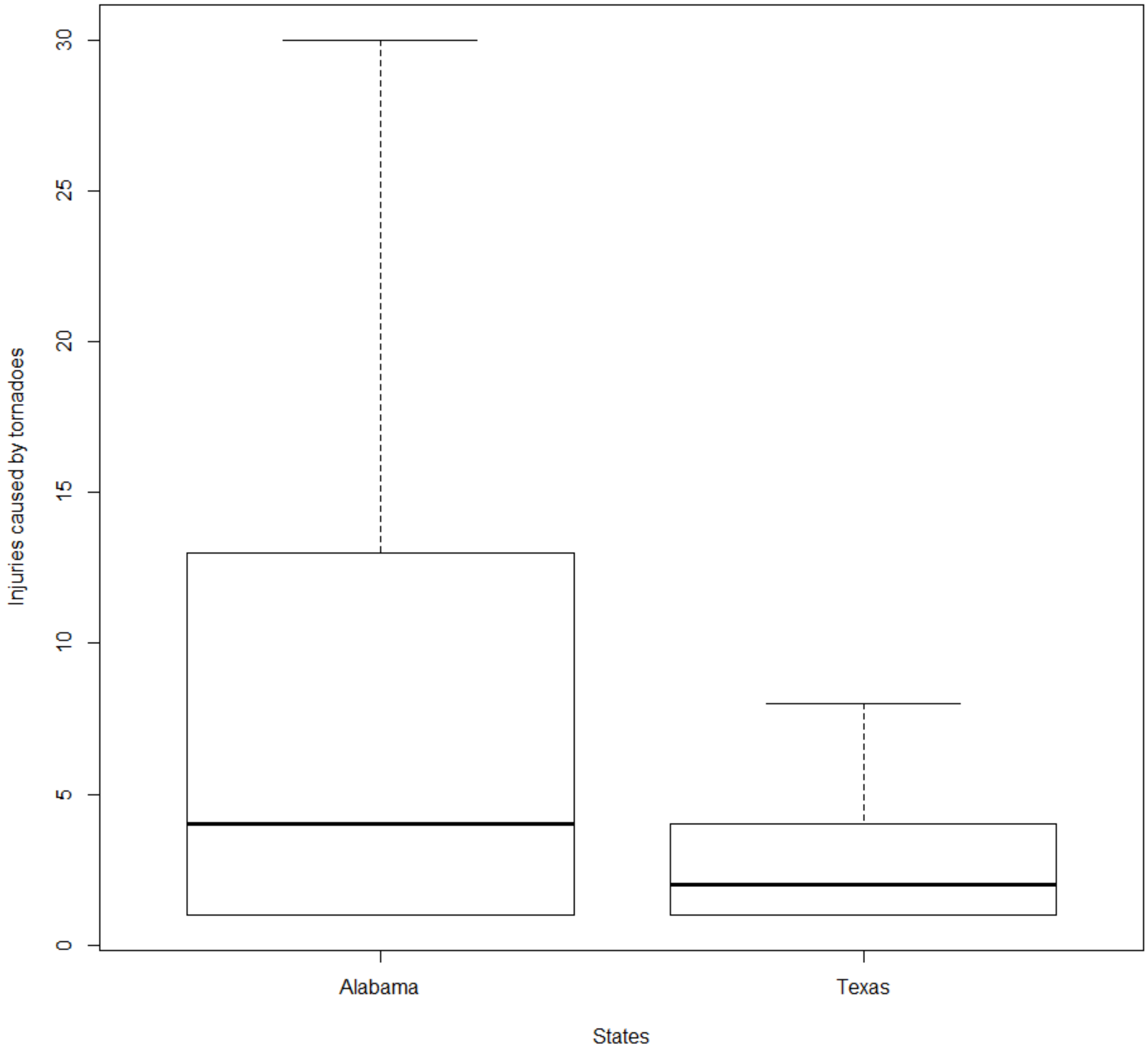
2) Hypotheis#2:

H_0: Alabama and Texas have the same number of injuries list caused by tornadoes

H_a: Alabama and Texas injuries list is not the same

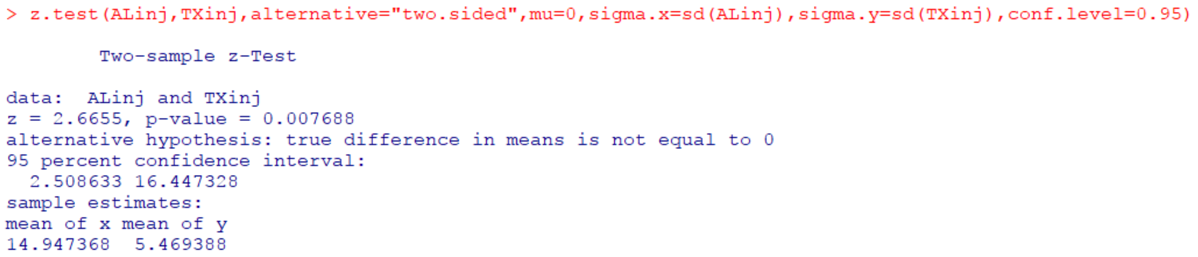
Data used: ALTXinjuries.csv

Box plot showing the distribution of the injury data of Alabama and Texas



It is clear from the box plot that state Alabama has a higher average of injury than that of state Texas. But we use hypothesis testing to confirm this.

Here, the two samples we used are two independent samples and are of size (58,50) more than 30. Hence, we perform two sample two tailed hypothesis testing



```
> z.test(ALinj,TXinj,alternative="two.sided",mu=0,sigma.x=sd(ALinj),sigma.y=sd(TXinj),conf.level=0.95)

        Two-sample z-Test

data:  ALinj and TXinj
z = 2.6655, p-value = 0.007688
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  2.508633 16.447328
sample estimates:
mean of x mean of y 
14.947368  5.469388 
```

From the above z test screenshot, we could see that p-value is less than 0.05. Hence at 95% confidence level, we do not have enough evidence to accept the null hypothesis. **Hence at 95% confidence level, we conclude that the number of people injured by tornadoes in states Texas and Alabama are not the same and that they are different.**

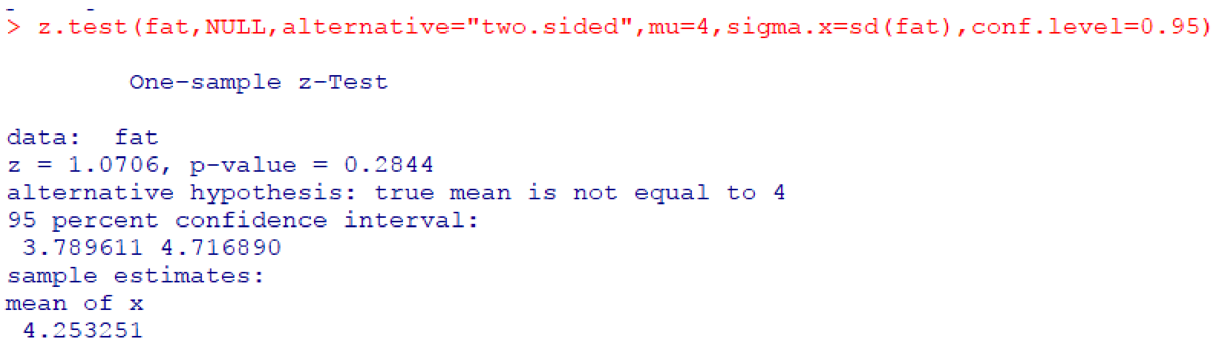
3) Hypothesis#3:

H_0: Average number of fatalities caused by tornadoes is 4

H_a: Average number of fatalities caused by tornadoes is not 4

Data used: fatalities.csv

Since the sample size of fatalities data (1616) > 30, we use z-test to perform one sample two tailed hypothesis testing to confirm if our assumption is right or not



```
> z.test(fat,NULL,alternative="two.sided",mu=4,sigma.x=sd(fat),conf.level=0.95)

        One-sample z-Test

data:  fat
z = 1.0706, p-value = 0.2844
alternative hypothesis: true mean is not equal to 4
95 percent confidence interval:
 3.789611 4.716890
sample estimates:
mean of x 
 4.253251 
```

From the above z test screenshot, we could see that the p-value > 0.05. Hence at 95% confidence level, we have enough evidence to accept the null hypothesis. Hence, **at 95% confidence level, we conclude that the average number of fatalities caused by tornadoes is 4.**

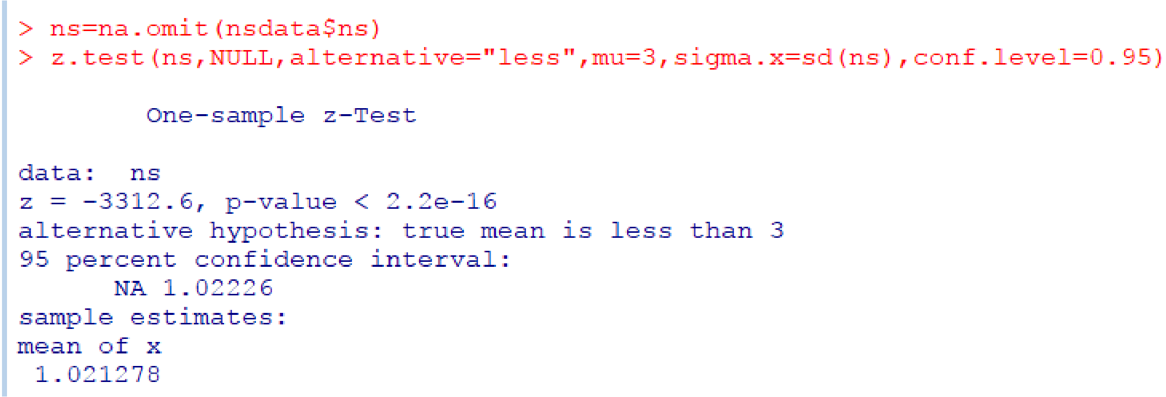
4) Hypothesis#4:

H_0: No of states affected by tornado of magnitudes 0 to 3 is 3

H_a: No of states affected by tornado of magnitude less than 3 is less than 3

Data used: noofstates.csv

Since the sample size of no of states data (61380) > 30, we use z-test to perform one sample one tailed hypothesis testing to confirm if our assumption is right or not.



```
> ns=na.omit(nsdata$ns)
> z.test(ns,NULL,alternative="less",mu=3,sigma.x=sd(ns),conf.level=0.95)

        One-sample z-Test

data:  ns
z = -3312.6, p-value < 2.2e-16
alternative hypothesis: true mean is less than 3
95 percent confidence interval:
      NA 1.02226
sample estimates:
mean of x
 1.021278
```

Here, the z-stat falls within the rejection region and the p-value is also less than 0.05. Hence, we do not have enough evidence to accept the null hypothesis at 95% confidence level which in turn favors the alternate hypothesis. Hence, **at 95% confidence level, it can be concluded that the no of states affected by tornado of magnitudes 0 to 3 are less than 3.**

5) Comparison of property loss incurred by different regions – ANOVA

Test data set*: ploss19962016.csv*

Sample size: 12406 records

Attributes: Region, property loss in dollars

Response variable: Property loss

Region groups: Midwest, NorthEast, SouthEast, SouthWest and West

Assumptions:

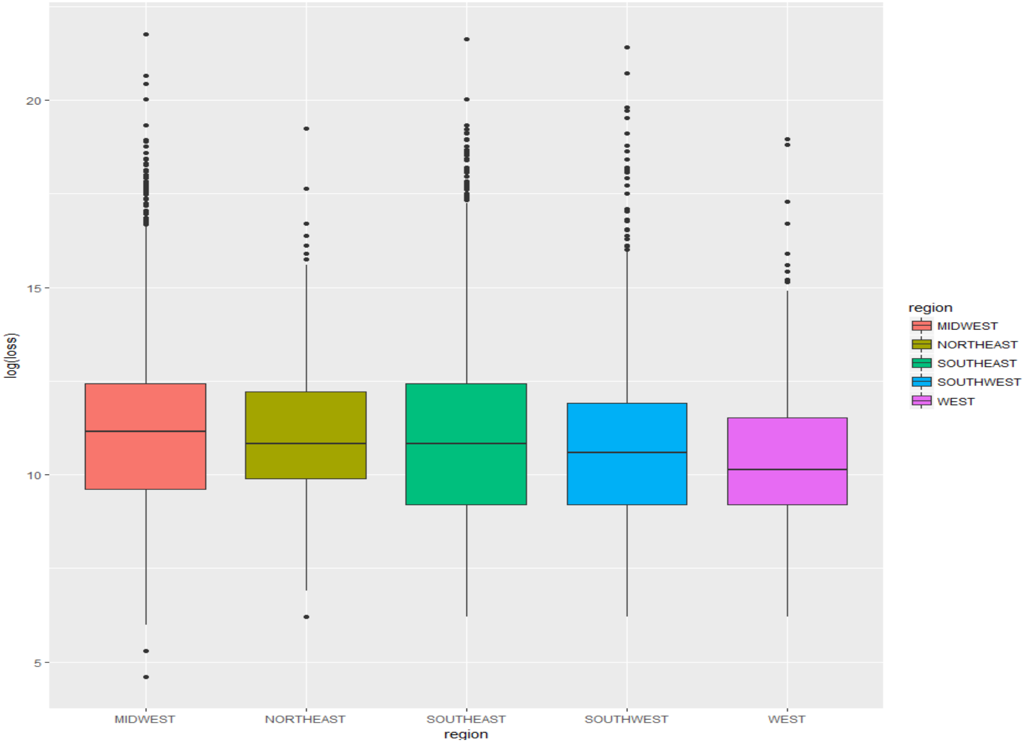
a) samples of losses incurred for each region is independent
b) populations from which the samples are taken is normal with unknown average loss
c) populations have the same standard deviation.

The ANOVA F-test tests the hypotheses:

H_0: Average property loss for all the regions are the same.

H_a: Not all the averages are equal

Box Plot:



The group means appears to be almost similar, with slight difference in in-group variation, from the analysis of boxplot.

ANOVA model:

```
> #build anova model for region Vs loss
>
>
>
> AnovaForRegionVsLoss = lm(loss~region)
> summary(AnovaForRegionVsLoss)

Call:
lm(formula = loss ~ region)

Residuals:
       Min         1Q     Median         3Q        Max
  -3194897   -2574608   -1929041   -1739041 2797485392

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      2614608     683870   3.823 0.000132 ***
regionNORTHEAST -1562193    2143537  -0.729 0.466143
regionSOUTHEAST  -675567     880828  -0.767 0.443115
regionSOUTHWEST   580789    1203583   0.483 0.629424
regionWEST      -1398018    2324741  -0.601 0.547608
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42570000 on 12401 degrees of freedom
Multiple R-squared:  0.0001665, Adjusted R-squared:  -0.000156
F-statistic: 0.5162 on 4 and 12401 DF,  p-value: 0.7238
```

F-statistic is 0.5 with p=0.72 (>0.05). This indicates our null hypothesis is true, that the average loss incurred by different regions are the same.

T-test on individual parameters also suggest the difference in averages are not significant for all the regions as all p-values > 0.05.

The model needs to be evaluated for model assumptions by residual analysis before concluding (Section 6.1)

6) Comparison of injuries due to tornadoes of different magnitude – ANOVA

Test data set*: mag injuries.csv*

Sample size: 7719 records

Attributes: Magnitude (0-5 scale), no. of people injured

Response variable: Magnitude

Magnitude groups: mag0, mag1, mag2, mag3, mag4, mag5.

Assumptions:

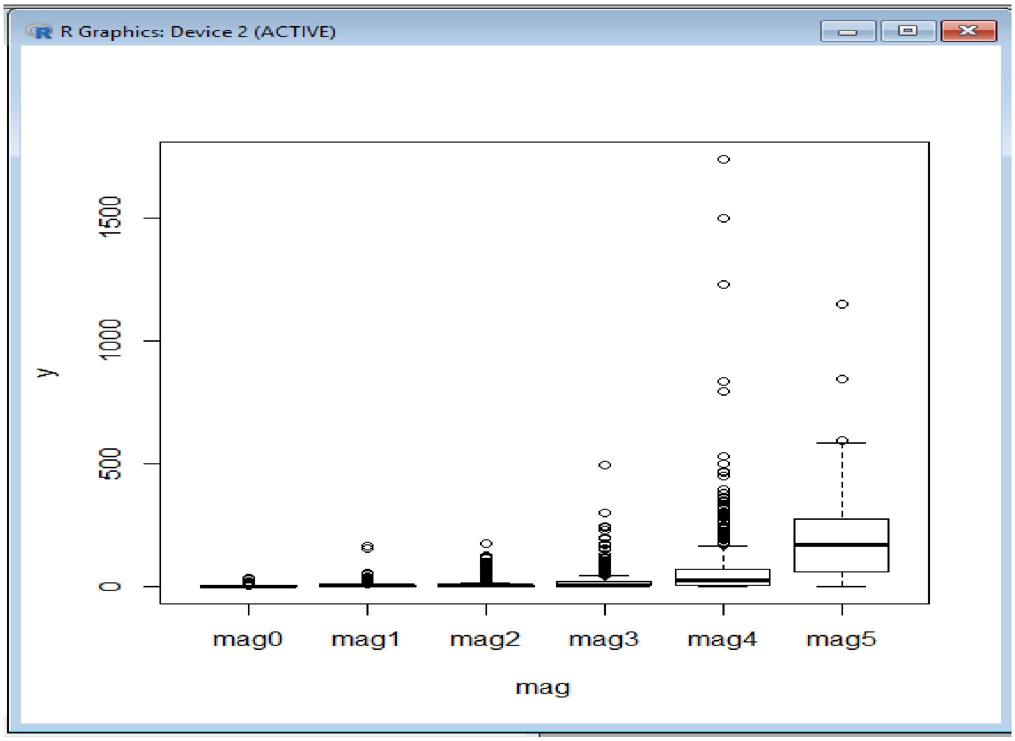
a) samples of injuries due to each tornado category is independent
b) populations from which the samples are taken is normal with unknown average no. of injuries
c) populations have the same standard deviation.

The ANOVA F-test tests the hypotheses:

H_0: Average no. of injuries due to all magnitude categories are the same.

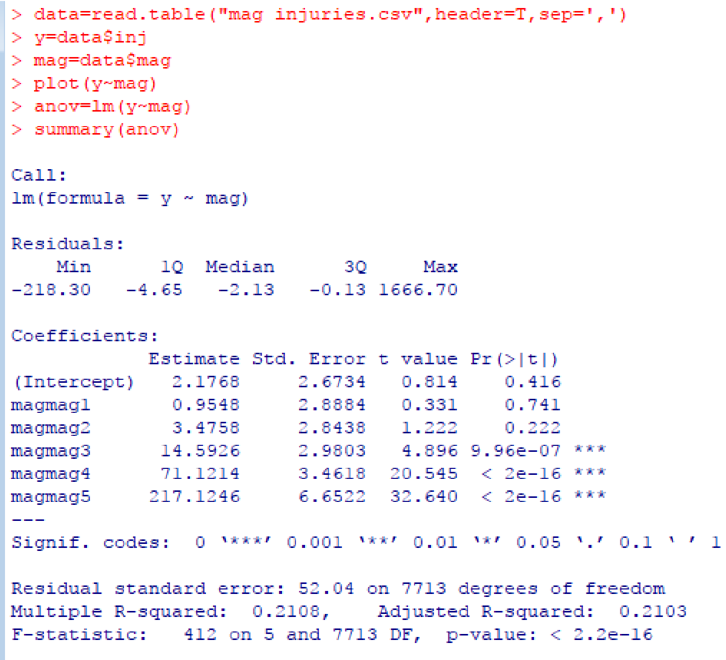
H_a: Not all the averages are equal

Box Plot:



The group means appears to be almost similar except for mag4 and mag5, with slight difference in in-group variation, from the analysis of boxplot.

ANOVA model:



```
> data=read.table("mag injuries.csv",header=T,sep=',')
> y=data$inj
> mag=data$mag
> plot(y~mag)
> anov=lm(y~mag)
> summary(anov)

Call:
lm(formula = y ~ mag)

Residuals:
    Min      1Q  Median      3Q     Max 
-218.30   -4.65   -2.13   -0.13 1666.70 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.1768     2.6734   0.814    0.416    
magmag1       0.9548     2.8884   0.331    0.741    
magmag2       3.4758     2.8438   1.222    0.222    
magmag3      14.5926     2.9803   4.896 9.96e-07 ***
magmag4      71.1214     3.4618  20.545  < 2e-16 ***
magmag5     217.1246     6.6522  32.640  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 52.04 on 7713 degrees of freedom
Multiple R-squared:  0.2108,	Adjusted R-squared:  0.2103 
F-statistic:   412 on 5 and 7713 DF,  p-value: < 2.2e-16
```

F-statistic is 412 and p-value<0.05, hence we can reject null hypothesis.

Also, t-test for individual parameters indicate that the difference in means is significant only for mag3, mag4 ang mag5 with respect to mag0 as p-values<0.05. The difference in means is insignificant for mag1 and mag2 with respect to mag0 as p-values>0.05.

The model needs to be evaluated for model assumptions by residual analysis before concluding (Section 6.1)

7) Group Magnitude in terms of Length, width and distance travelled by a tornado – KNN Classification

Test data set: *mag categ prediction.csv*

Sample size: 11797

Attributes: length, width, distance, magnitude

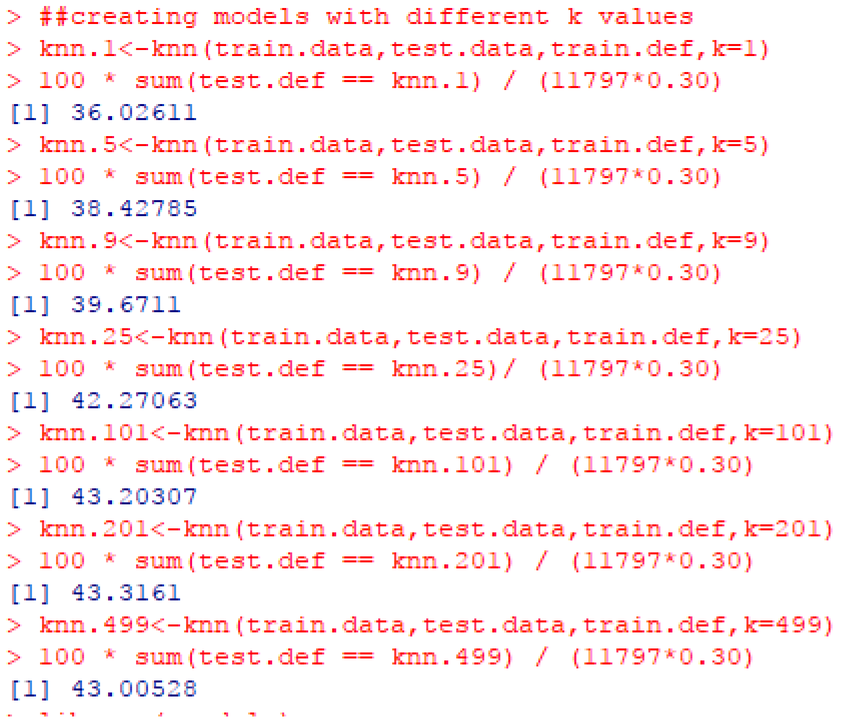
Label: Magnitude

Features: Length, width, distance

Data split: 70/30 split by Hold out Evaluation as the sample size is large. (note: Attempted 80/20, 75/25 splits, with different k-values, but result was better with 70/30. Hence, documenting for 70/30 split alone)

Data preparation: Convert dependent variable magnitude to factor. Normalize numeric variables length, width and distance.

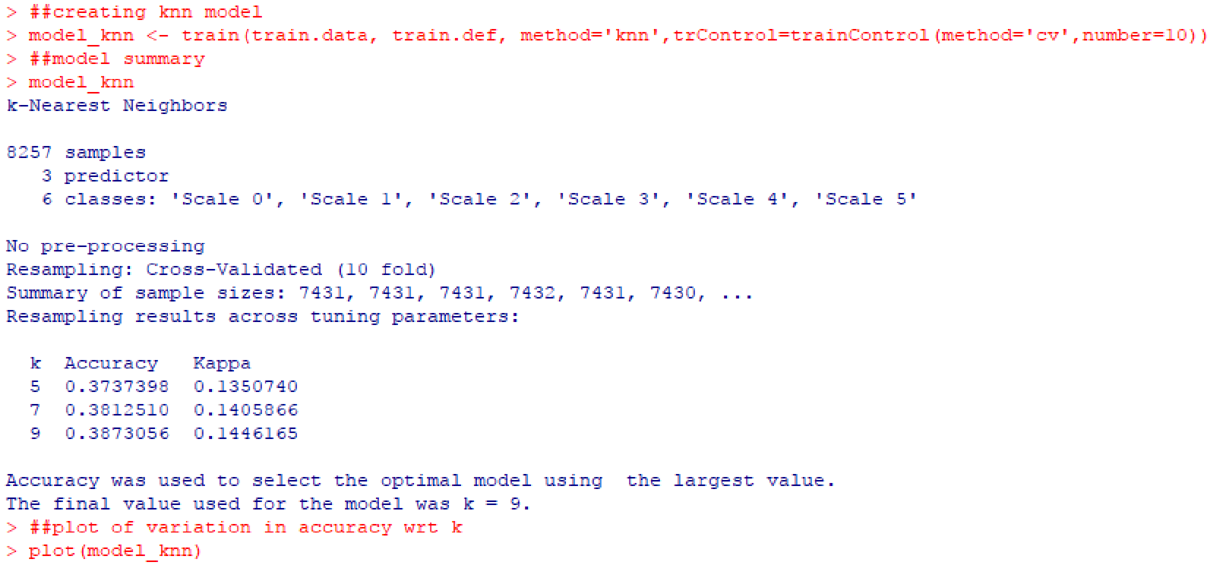
Method: Build K-Nearest Neighbour model for k=1, 5,9,25,101,201,499 and calculate the accuracy.



```
> ##creating models with different k values
> knn.1<-knn(train.data,test.data,train.def,k=1)
> 100 * sum(test.def == knn.1) / (11797*0.30)
[1] 36.02611
> knn.5<-knn(train.data,test.data,train.def,k=5)
> 100 * sum(test.def == knn.5) / (11797*0.30)
[1] 38.42785
> knn.9<-knn(train.data,test.data,train.def,k=9)
> 100 * sum(test.def == knn.9) / (11797*0.30)
[1] 39.6711
> knn.25<-knn(train.data,test.data,train.def,k=25)
> 100 * sum(test.def == knn.25)/ (11797*0.30)
[1] 42.27063
> knn.101<-knn(train.data,test.data,train.def,k=101)
> 100 * sum(test.def == knn.101) / (11797*0.30)
[1] 43.20307
> knn.201<-knn(train.data,test.data,train.def,k=201)
> 100 * sum(test.def == knn.201) / (11797*0.30)
[1] 43.3161
> knn.499<-knn(train.data,test.data,train.def,k=499)
> 100 * sum(test.def == knn.499) / (11797*0.30)
[1] 43.00528
```

The accuracy levels are less for almost all models with k=201 being the better model.

We also tried model by training data by knn and resampling by cross validation.



```
> ##creating knn model
> model_knn <- train(train.data, train.def, method='knn',trControl=trainControl(method='cv',number=10))
> ##model summary
> model_knn
k-Nearest Neighbors

8257 samples
   3 predictor
   6 classes: 'Scale 0', 'Scale 1', 'Scale 2', 'Scale 3', 'Scale 4', 'Scale 5'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7431, 7431, 7431, 7432, 7431, 7430, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.3737398  0.1350740
  7  0.3812510  0.1405866
  9  0.3873056  0.1446165

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was k = 9.
> ##plot of variation in accuracy wrt k
> plot(model_knn)
```

The method took k=9 as the best model, but accuracy is only 0.38

Model evaluation for accuracy needs to be carried out using crosstable/confusion matrix (section 6.1)

8) Group Property Loss category in terms of Magnitude category of tornado, region of occurrence and no. of states affected – Naïve Bayes Classification

Test data set: *property loss prediction.csv*

Sample size: 9705

Attributes: Property loss, magnitude, region, no. of states

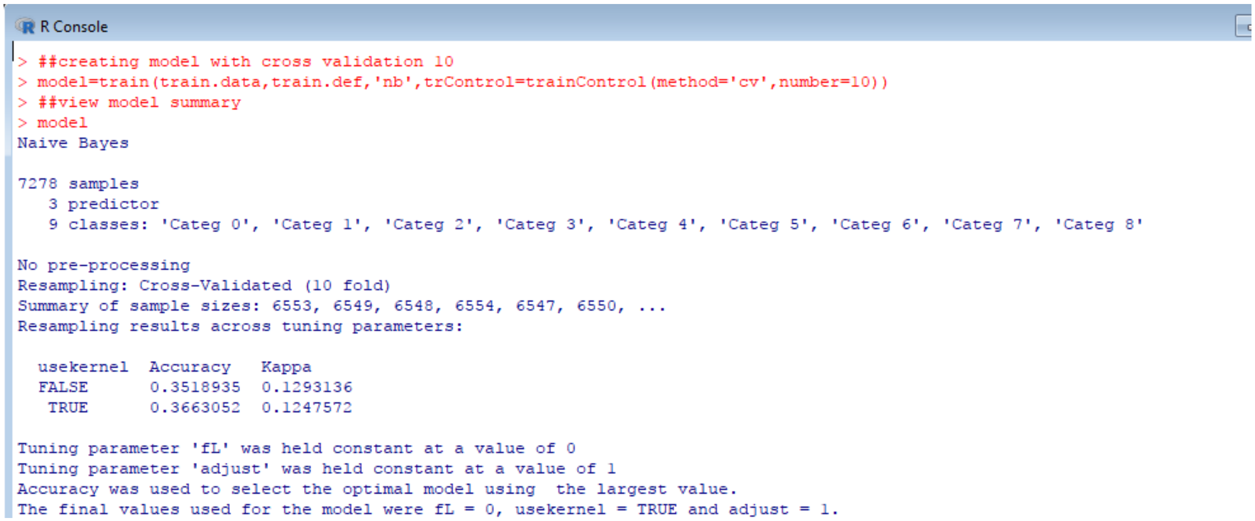
Label: Property loss

Features: magnitude, region, no. of states

Data split: 10-folds cross validation as the sample size is less

Data preparation: Convert dependent variable magnitude to factor. Normalize numeric variables length, width and distance.

Method: We use Naïve Bayes since the features are nominal.



```
> ##creating model with cross validation 10
> model=train(train.data,train.def,'nb',trControl=trainControl(method='cv',number=10))
> ##view model summary
> model
Naive Bayes

7278 samples
   3 predictor
   9 classes: 'Categ 0', 'Categ 1', 'Categ 2', 'Categ 3', 'Categ 4', 'Categ 5', 'Categ 6', 'Categ 7', 'Categ 8'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6553, 6549, 6548, 6554, 6547, 6550, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.3518935  0.1293136
   TRUE      0.3663052  0.1247572

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

The accuracy of the model obtained is less 0.35.

We need to further evaluate the model using predictions to conclude (Sec 6.1)

9) Time series analysis on Tornadoes Length data:

Tornadoes occurred over the years have had different lengths throughout. Time series analysis is an attempt to learn the distribution of length of tornadoes over the years. Since the daily data collected were not continuous, we took the average length data per month and used the new monthly data for the time series analysis. There were data missing for 3 months in three different years. So, we arrived at the average length of the year and filled this average for the respective months which missed the length data

Data used: torn-length-monthly-data.csv

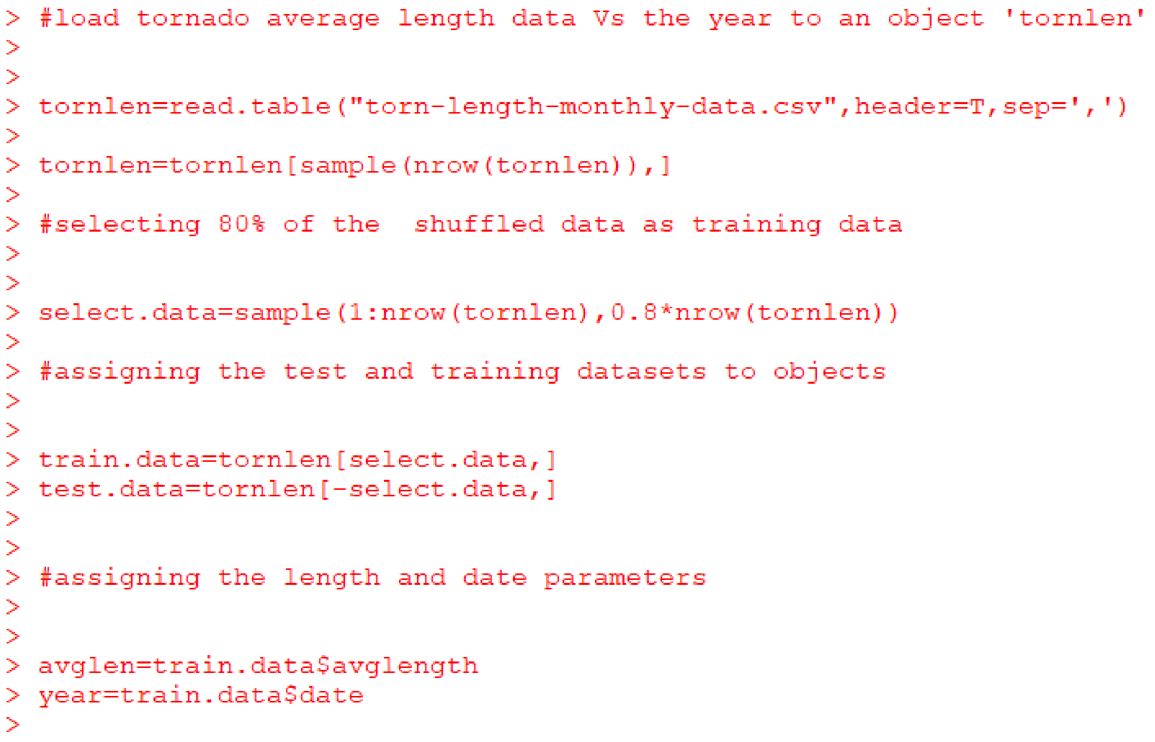
Data spans from 1950 to 2016 covering 67 years and 67X12 = 804 months.

Process followed for Time Series:

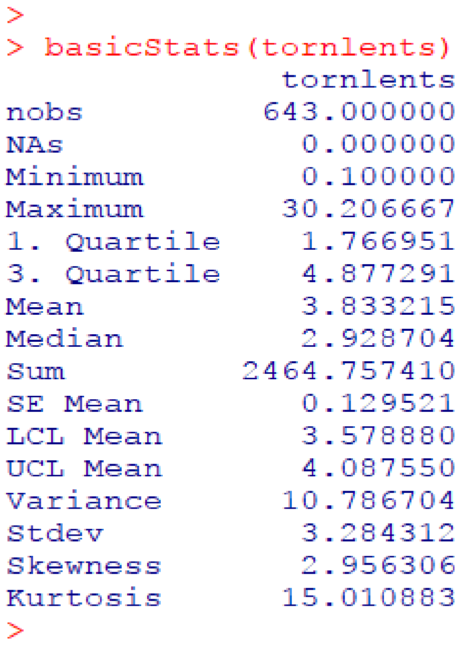
- Loading time series data to an object
- Split the data into training and testing dataset
- Creation of Time Series Object for the training dataset
- Preliminary Data Analysis to test assumptions
- Decide if transformation or differencing is required
- Identify p and q values for building time series models – AR/MA/ARMA
- Build models
- Perform residual analysis of every model
- Evaluate the models based on RMSE
- Choose the best time series model
- Plot the predictions and forecasts graph with predicted values

Every step followed is as detailed below with appropriate screenshots

Loading time series data and split-up



```
> #load tornado average length data Vs the year to an object 'tornlen'
>
>
> tornlen=read.table("torn-length-monthly-data.csv",header=T,sep=',')
>
> tornlen=tornlen[sample(nrow(tornlen)),]
>
> #selecting 80% of the  shuffled data as training data
>
>
> select.data=sample(1:nrow(tornlen),0.8*nrow(tornlen))
>
> #assigning the test and training datasets to objects
>
>
> train.data=tornlen[select.data,]
> test.data=tornlen[-select.data,]
>
>
> #assigning the length and date parameters
>
>
> avglen=train.data$avglength
> year=train.data$date
>
```



```
>
> basicStats(tornlents)
                 tornlents
nobs            643.000000
NAs               0.000000
Minimum           0.100000
Maximum          30.206667
1. Quartile       1.766951
3. Quartile       4.877291
Mean              3.833215
Median            2.928704
Sum            2464.757410
SE Mean           0.129521
LCL Mean          3.578880
UCL Mean          4.087550
Variance         10.786704
Stdev             3.284312
Skewness          2.956306
Kurtosis         15.010883
>
```

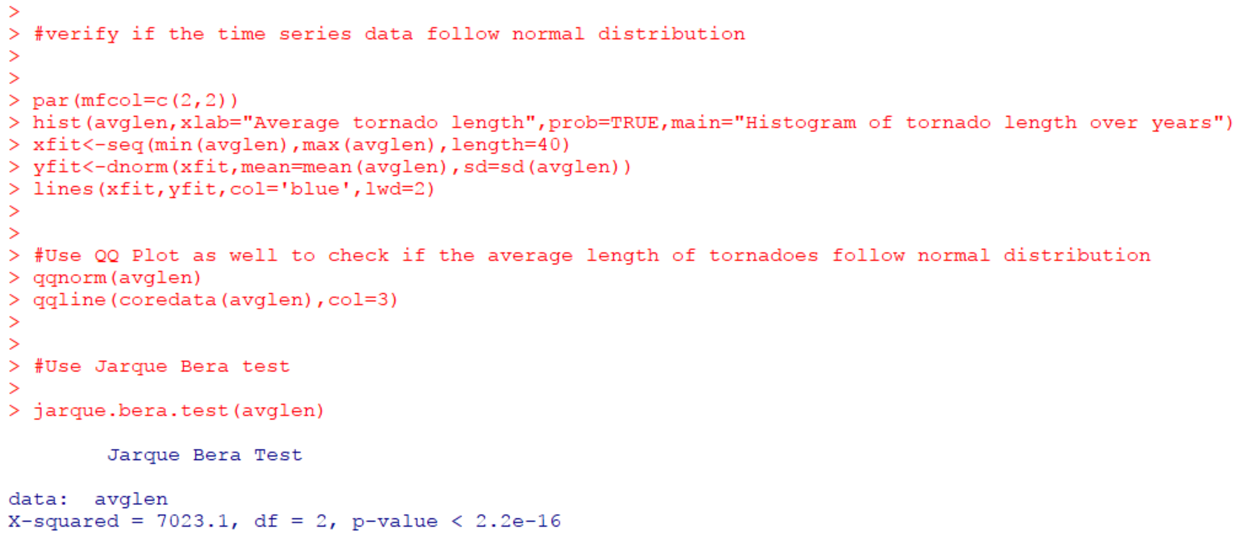
As you can see here, we have split the data into test and training and created the time series object

Preliminary data analysis to test assumptions of time series:

Normal Distribution: Most important assumption for time series which the time series data is expected to meet.

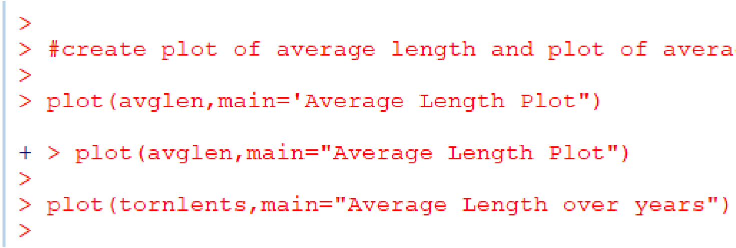
Tornadoes length data meets this requirement?

To test this, we plotted the histogram, QQ plot and Jarque Bera test as shown below



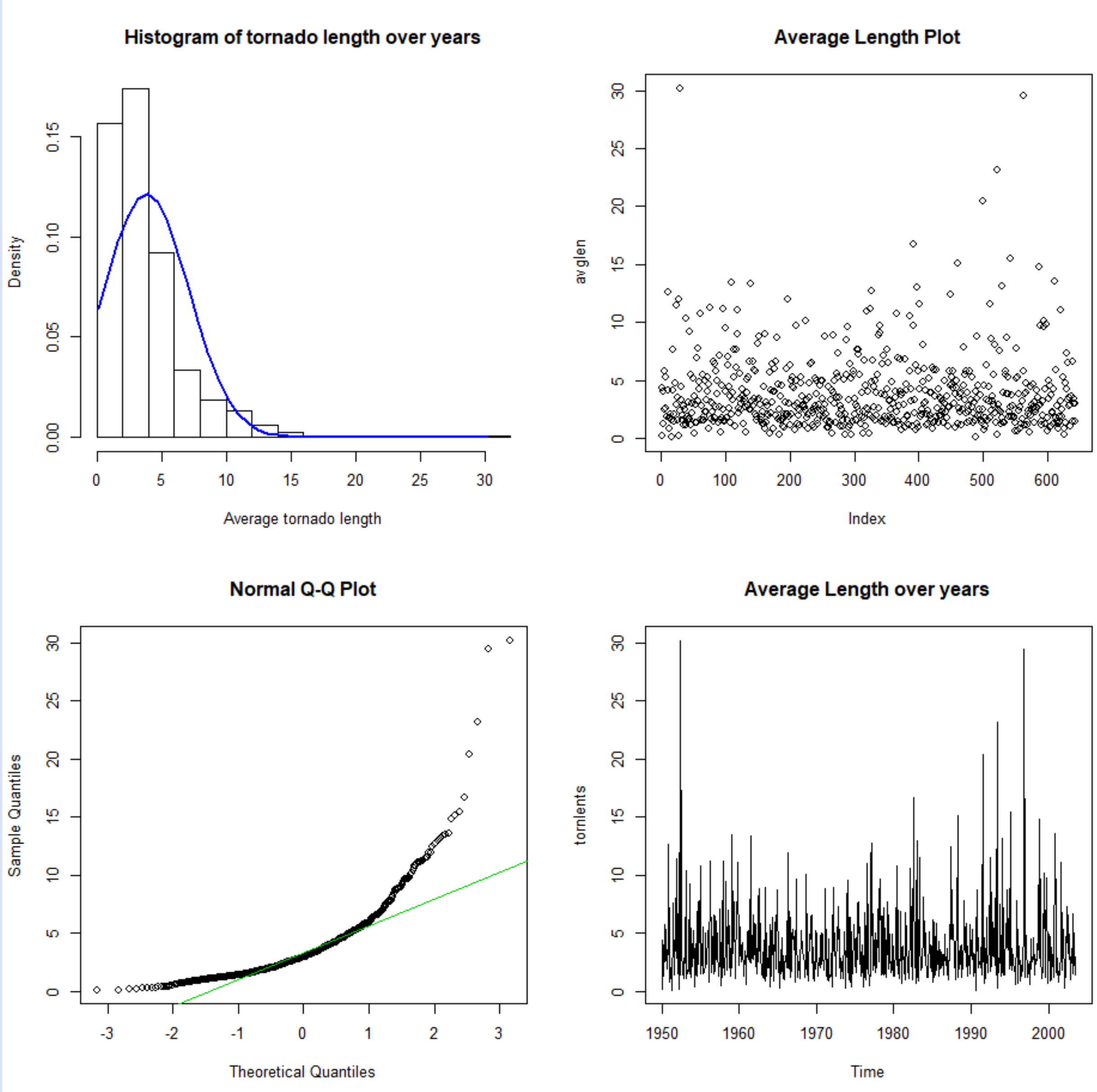
```
>
> #verify if the time series data follow normal distribution
>
>
> par(mfcol=c(2,2))
> hist(avglen,xlab="Average tornado length",prob=TRUE,main="Histogram of tornado length over years")
> xfit<-seq(min(avglen),max(avglen),length=40)
> yfit<-dnorm(xfit,mean=mean(avglen),sd=sd(avglen))
> lines(xfit,yfit,col='blue',lwd=2)
>
>
> #Use QQ Plot as well to check if the average length of tornadoes follow normal distribution
> qqnorm(avglen)
> qqline(coredata(avglen),col=3)
>
>
> #Use Jarque Bera test
>
> jarque.bera.test(avglen)

        Jarque Bera Test

data:  avglen
X-squared = 7023.1, df = 2, p-value < 2.2e-16
```

We created the time and time series plots too



```
>
> #create plot of average length and plot of average length time series object too
>
> plot(avglen,main='Average Length Plot")

+ > plot(avglen,main="Average Length Plot")
>
> plot(tornlents,main="Average Length over years")
>
```

Resulting graphs and plots of Normality tests and time plots



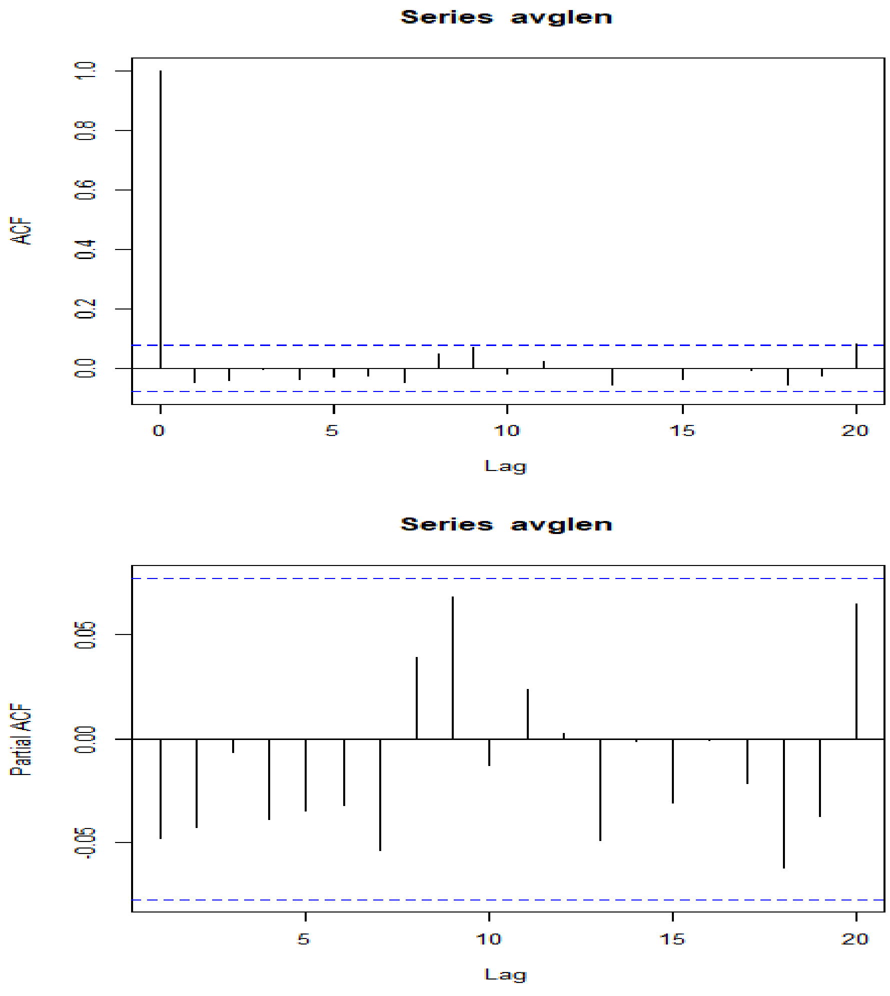
It is clear from the above histogram, QQ Plot and Jarque Bera Test that the tornadoes length data **doesn't follow normal distribution**

Stationarity:

Stationarity refers to the constant variance and constant mean of the tornadoes average length throughout the years which is one of the critical assumptions that the time series data is expected to meet.

Let's see if our data meets this stationarity requirement. We plot the ACF plot to verify this.

```
> #plot to verify the stationarity assumption of the data
>
> acf(avglen,plot=F,lag.max=20,na.action=na.pass)

Autocorrelations of series 'avglen', by lag

     0      1      2      3      4      5      6      7      8      9     10     11     12 
 1.000 -0.048 -0.040 -0.002 -0.036 -0.030 -0.025 -0.047  0.048  0.070 -0.020  0.024  0.001 
    13     14     15     16     17     18     19     20 
-0.055  0.000 -0.035 -0.001 -0.008 -0.054 -0.024  0.082 
> pacf(avglen,plot=F,lag.max=20,na.action=na.pass)

Partial autocorrelations of series 'avglen', by lag

     1      2      3      4      5      6      7      8      9     10     11     12     13 
-0.048 -0.042 -0.006 -0.039 -0.035 -0.032 -0.054  0.039  0.068 -0.013  0.023  0.002 -0.049 
    14     15     16     17     18     19     20 
-0.001 -0.031  0.000 -0.021 -0.062 -0.037  0.065 
> acf(avglen,plot=T,lag.max=20,na.action=na.pass)
> pacf(avglen,plot=T,lag.max=20,na.action=na.pass)
>
```



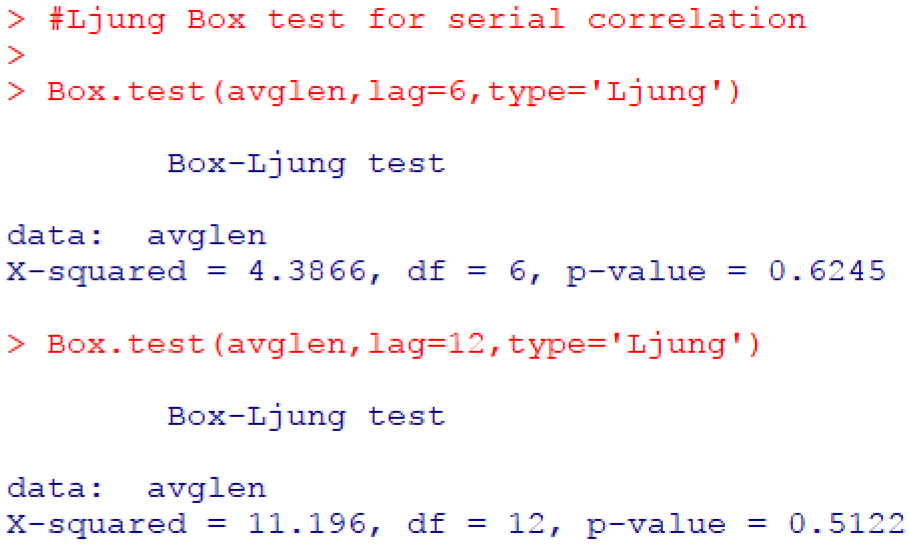
There is no single non-zero auto-correlation value present and the acf values decay slowly in the ACF plot and hence this confirms that the tornadoes length data is not stationary

Serial Correlation:

In time series, the data should be correlated to the past data and only then the prediction and model building can be done. This aspect of correlation between the current and past data is referred to as serial correlation.

Tornadoes length data meets this requirement?

To answer this, we need to perform the Ljung Box Test on the tornadoes length data as shown below



```
> #Ljung Box test for serial correlation
>
> Box.test(avglen,lag=6,type='Ljung')

        Box-Ljung test

data:  avglen
X-squared = 4.3866, df = 6, p-value = 0.6245

> Box.test(avglen,lag=12,type='Ljung')

        Box-Ljung test

data:  avglen
X-squared = 11.196, df = 12, p-value = 0.5122
```

The hypothesis statements considered in this Ljung Box test are as follows:

H_0: Time series data has serial correlation and are not white noise

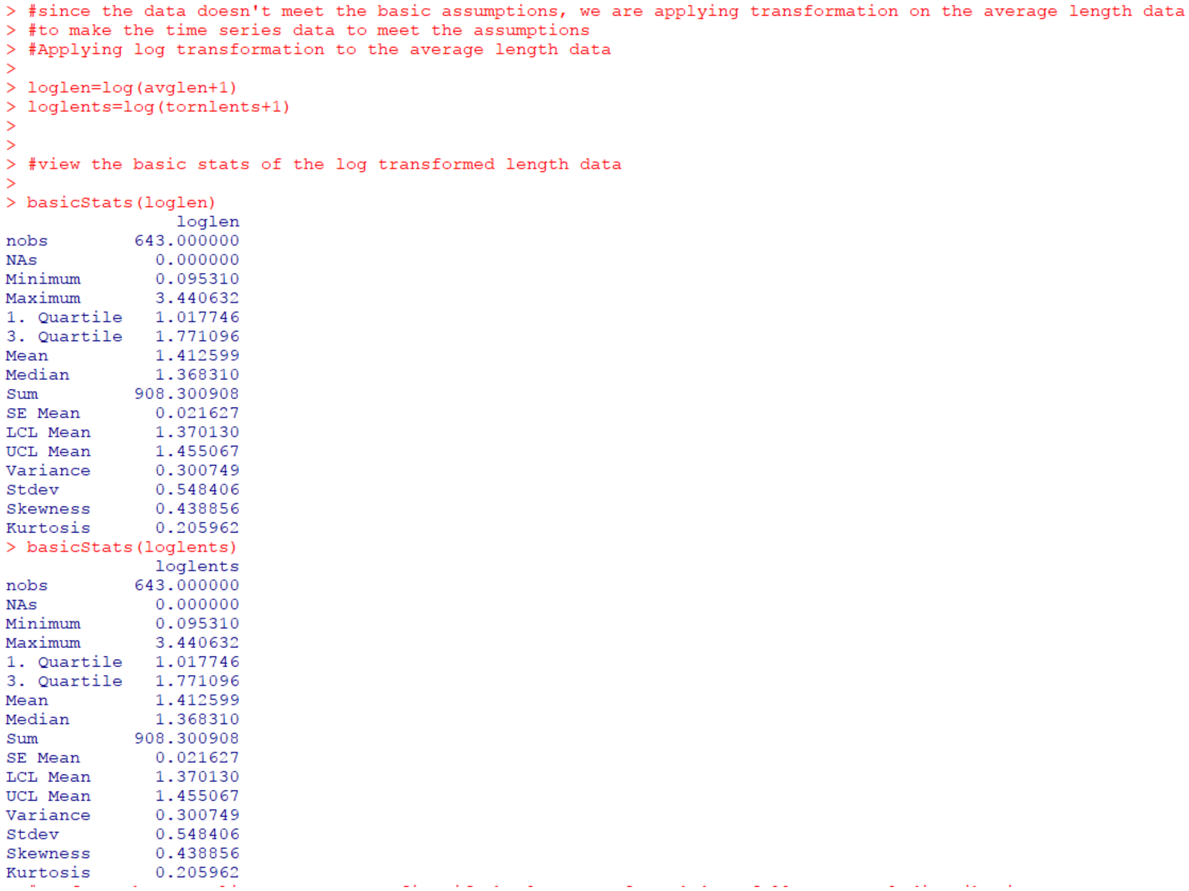
H_a: Time series data has no serial correlation and are white noise

From the above test result, we could see that the p-value of this test at both lags 6 & 12 are more than 0.05 favoring the null hypothesis. Hence, this indicates **that the tornadoes length data is not serial correlated**

<u>Decide if transformation or differencing is required</u>

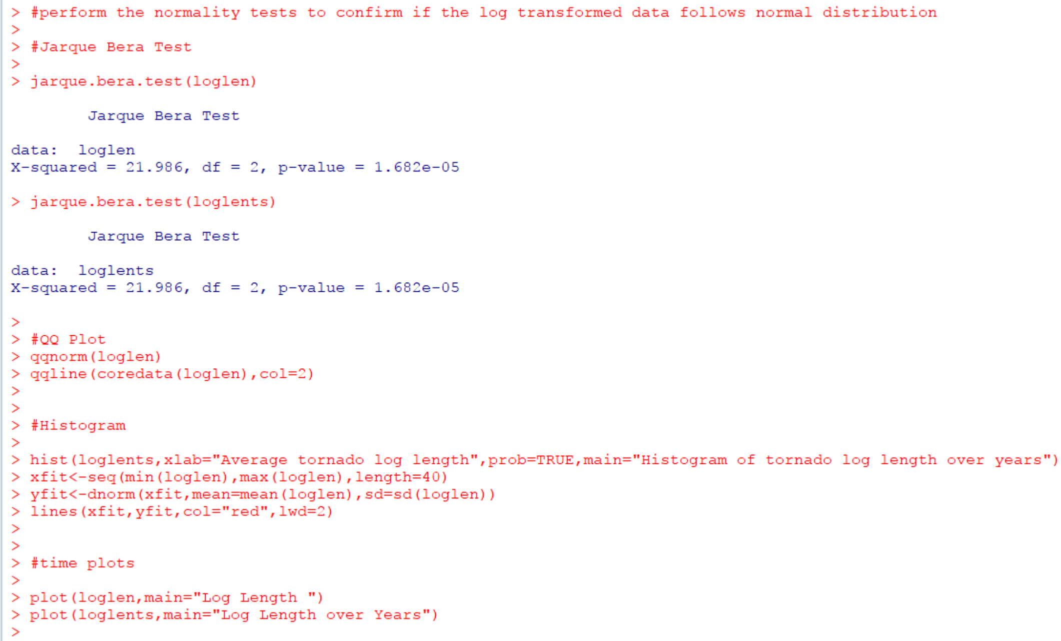
It is obvious from the above screenshots that the data doesn't meet the requirements of time series data. So, we applied log transformation on the tornado length data since log transformation normalizes the data. We need to perform the preliminary analysis on this log transformed data as shown below.

Log transformed data:



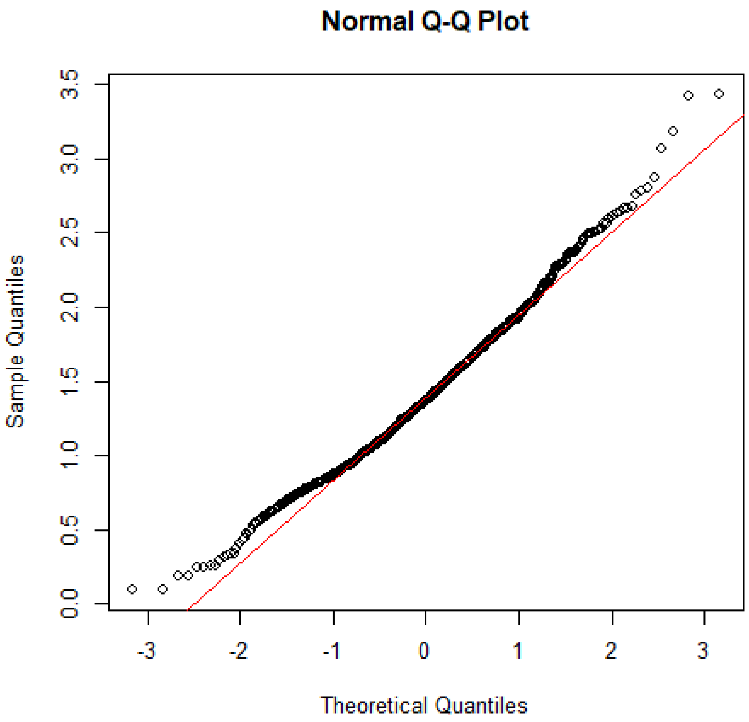
```
> #since the data doesn't meet the basic assumptions, we are applying transformation on the average length data
> #to make the time series data to meet the assumptions
> #Applying log transformation to the average length data
>
> loglen=log(avglen+1)
> loglents=log(tornlents+1)
>
>
> #view the basic stats of the log transformed length data
>
> basicStats(loglen)
                loglen
nobs        643.000000
NAs           0.000000
Minimum       0.095310
Maximum       3.440632
1. Quartile   1.017746
3. Quartile   1.771096
Mean          1.412599
Median        1.368310
Sum         908.300908
SE Mean       0.021627
LCL Mean      1.370130
UCL Mean      1.455067
Variance      0.300749
Stdev         0.548406
Skewness      0.438856
Kurtosis      0.205962
> basicStats(loglents)
              loglents
nobs        643.000000
NAs           0.000000
Minimum       0.095310
Maximum       3.440632
1. Quartile   1.017746
3. Quartile   1.771096
Mean          1.412599
Median        1.368310
Sum         908.300908
SE Mean       0.021627
LCL Mean      1.370130
UCL Mean      1.455067
Variance      0.300749
Stdev         0.548406
Skewness      0.438856
Kurtosis      0.205962
```

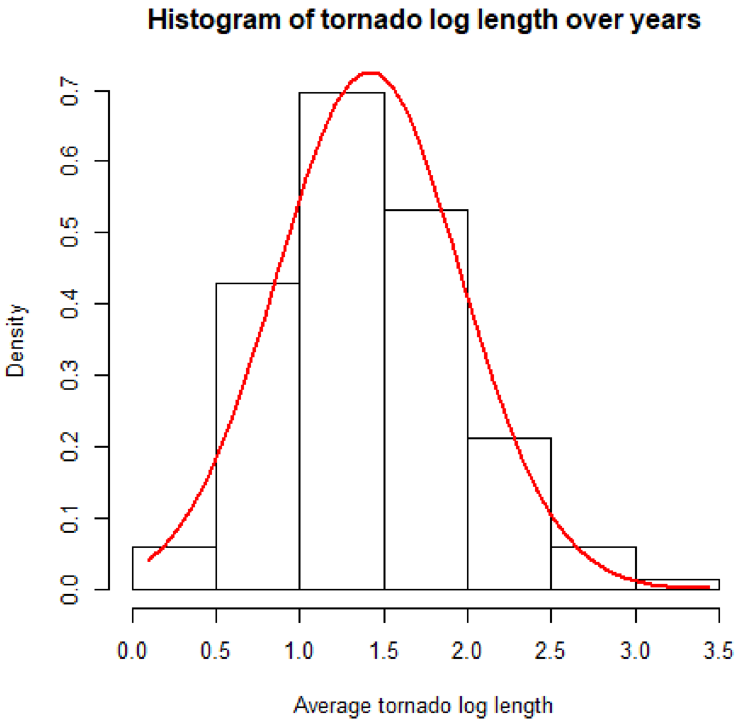
Normality tests and time plots on log transformed data:



```
> #perform the normality tests to confirm if the log transformed data follows normal distribution
>
> #Jarque Bera Test
>
> jarque.bera.test(loglen)

        Jarque Bera Test

data:  loglen
X-squared = 21.986, df = 2, p-value = 1.682e-05

> jarque.bera.test(loglents)

        Jarque Bera Test

data:  loglents
X-squared = 21.986, df = 2, p-value = 1.682e-05

>
> #QQ Plot
> qqnorm(loglen)
> qqline(coredata(loglen),col=2)
>
>
> #Histogram
>
> hist(loglents,xlab="Average tornado log length",prob=TRUE,main="Histogram of tornado log length over years")
> xfit<-seq(min(loglen),max(loglen),length=40)
> yfit<-dnorm(xfit,mean=mean(loglen),sd=sd(loglen))
> lines(xfit,yfit,col="red",lwd=2)
>
>
> #time plots
>
> plot(loglen,main="Log Length ")
> plot(loglents,main="Log Length over Years")
>
```

Log transformed data normality test results:

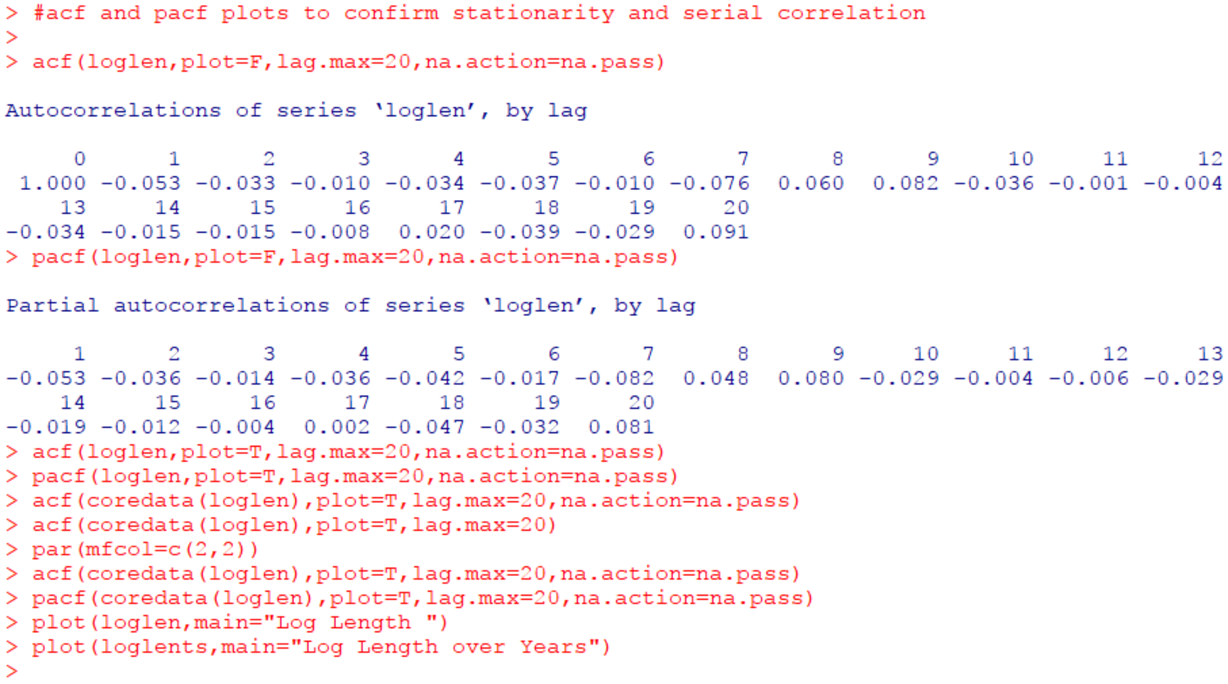




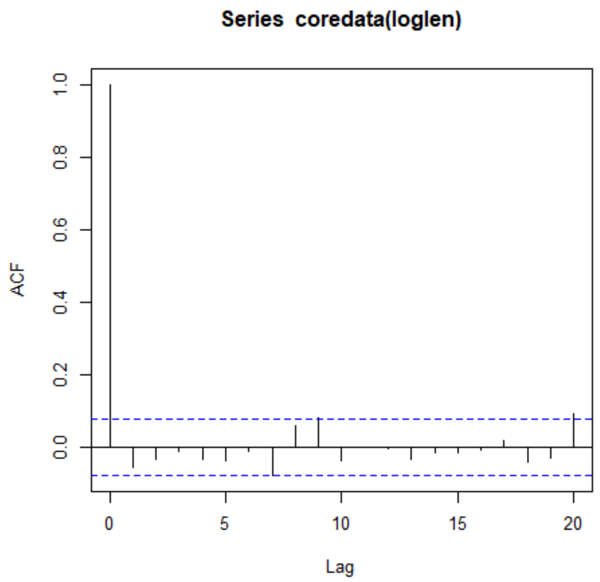
It is now clear from the above screenshots that the log transformed data follows normal distribution. The skewness and kurtosis values in the Basic Stats also prove this.

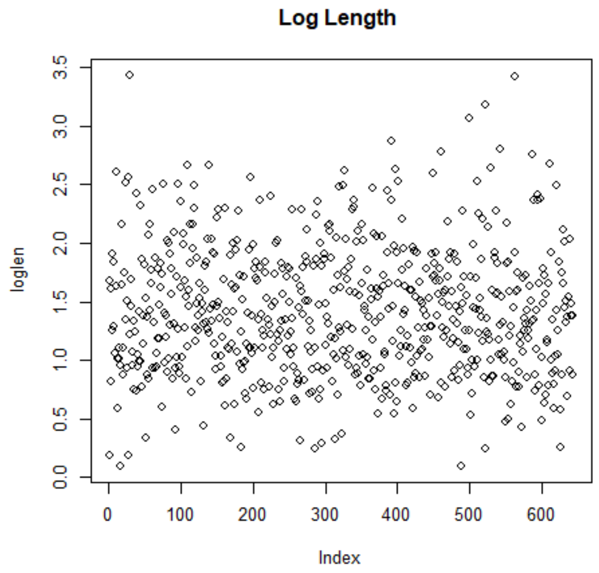
Stationarity tests:

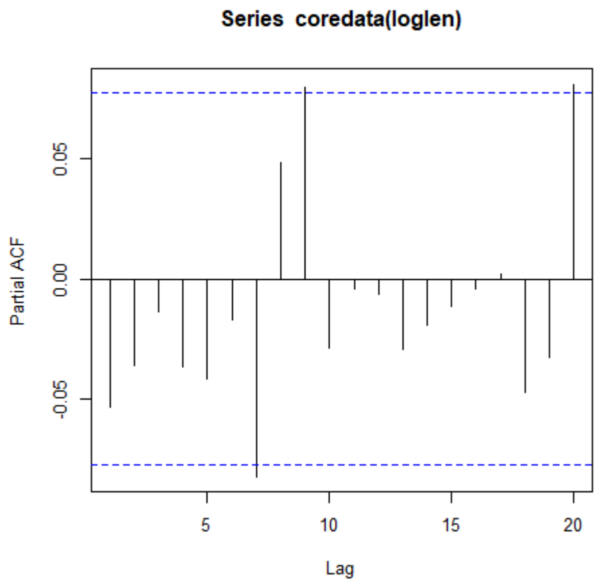
We plotted the ACF and PACF plots to confirm the stationarity as given below

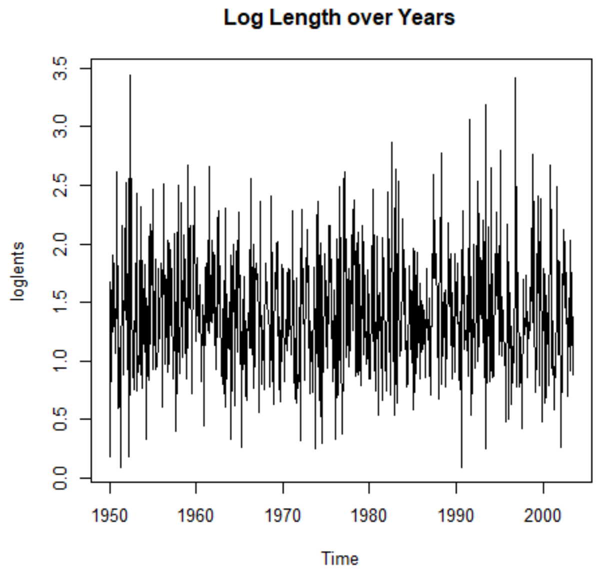


```
> #acf and pacf plots to confirm stationarity and serial correlation
>
> acf(loglen,plot=F,lag.max=20,na.action=na.pass)

Autocorrelations of series 'loglen', by lag

     0      1      2      3      4      5      6      7      8      9     10     11     12 
 1.000 -0.053 -0.033 -0.010 -0.034 -0.037 -0.010 -0.076  0.060  0.082 -0.036 -0.001 -0.004 
    13     14     15     16     17     18     19     20 
-0.034 -0.015 -0.015 -0.008  0.020 -0.039 -0.029  0.091 
> pacf(loglen,plot=F,lag.max=20,na.action=na.pass)

Partial autocorrelations of series 'loglen', by lag

     1      2      3      4      5      6      7      8      9     10     11     12     13 
-0.053 -0.036 -0.014 -0.036 -0.042 -0.017 -0.082  0.048  0.080 -0.029 -0.004 -0.006 -0.029 
    14     15     16     17     18     19     20 
-0.019 -0.012 -0.004  0.002 -0.047 -0.032  0.081 
> acf(loglen,plot=T,lag.max=20,na.action=na.pass)
> pacf(loglen,plot=T,lag.max=20,na.action=na.pass)
> acf(coredata(loglen),plot=T,lag.max=20,na.action=na.pass)
> acf(coredata(loglen),plot=T,lag.max=20)
> par(mfcol=c(2,2))
> acf(coredata(loglen),plot=T,lag.max=20,na.action=na.pass)
> pacf(coredata(loglen),plot=T,lag.max=20,na.action=na.pass)
> plot(loglen,main="Log Length ")
> plot(loglents,main="Log Length over Years")
>
```







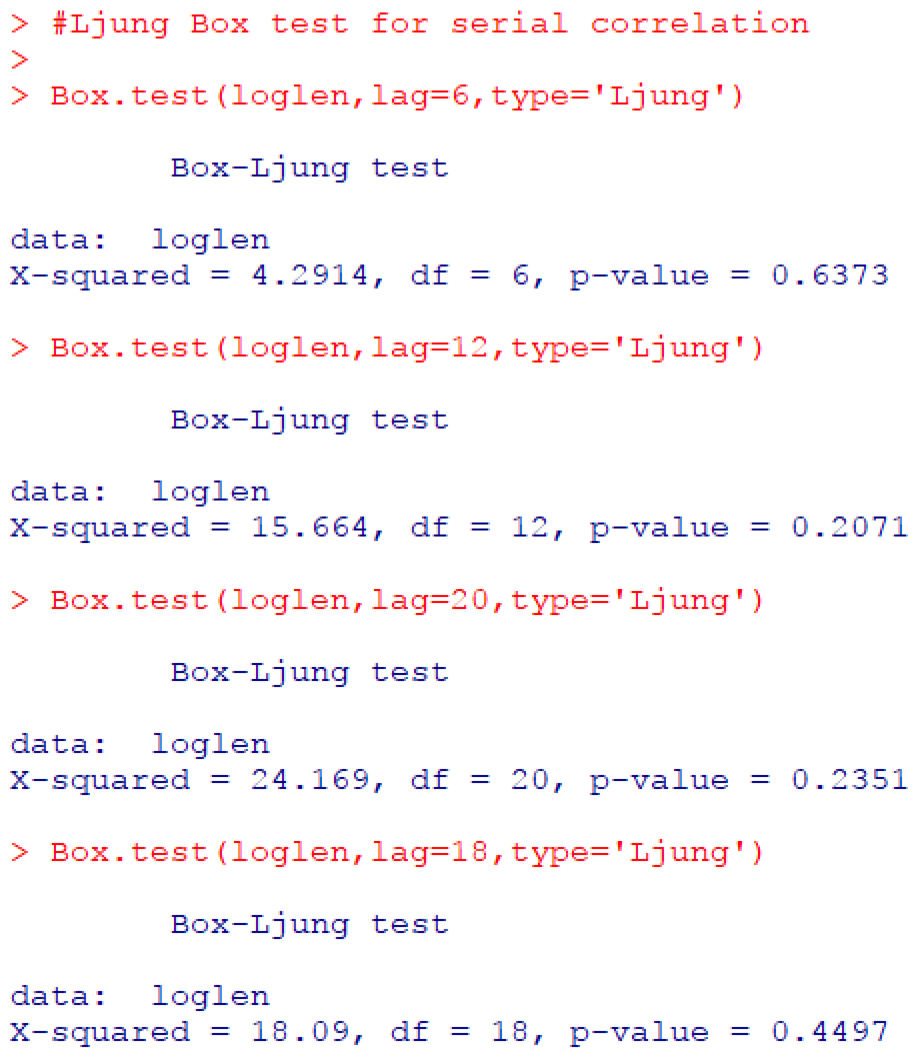


There is one non-zero autocorrelation value present in the ACF plot indicating serial correlation but it is not stationary.

We need to confirm this with the Ljung Box test as well

Serial correlation:

We performed Ljung Box test on the log transformed data to test serial correlation as shown below



```
> #Ljung Box test for serial correlation
>
> Box.test(loglen,lag=6,type='Ljung')

        Box-Ljung test

data:  loglen
X-squared = 4.2914, df = 6, p-value = 0.6373

> Box.test(loglen,lag=12,type='Ljung')

        Box-Ljung test

data:  loglen
X-squared = 15.664, df = 12, p-value = 0.2071

> Box.test(loglen,lag=20,type='Ljung')

        Box-Ljung test

data:  loglen
X-squared = 24.169, df = 20, p-value = 0.2351

> Box.test(loglen,lag=18,type='Ljung')

        Box-Ljung test

data:  loglen
X-squared = 18.09, df = 18, p-value = 0.4497
```

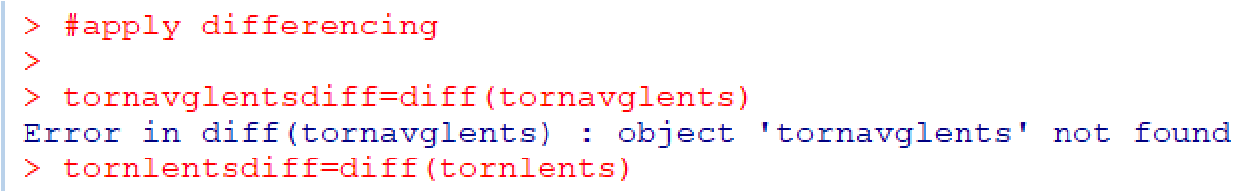
As stated earlier, the null hypothesis in Ljung Box test is ‘there is no serial correlation and the time series data is white noise’. Here again, the p-value is > 0.05 favoring the null hypothesis and hence the log transformed data is not serially correlated.

Hence we can conclude that , though **the log transformed data is normally distributed, it is not stationary and serial correlated and cannot be used for time series analysis**.

We no need to apply differencing on the length data to see if they make the length data meet the time series assumptions.

First Differencing:

Differencing is applied on the time series object of tornadoes length data as shown below.

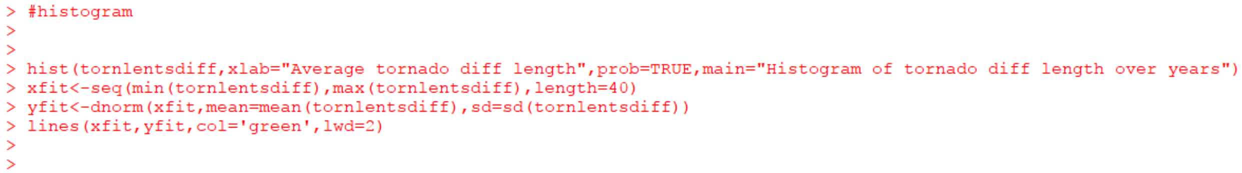


```
> #apply differencing
>
> tornavglentsdiff=diff(tornavglents)
Error in diff(tornavglents) : object 'tornavglents' not found
> tornlentsdiff=diff(tornlents)
```

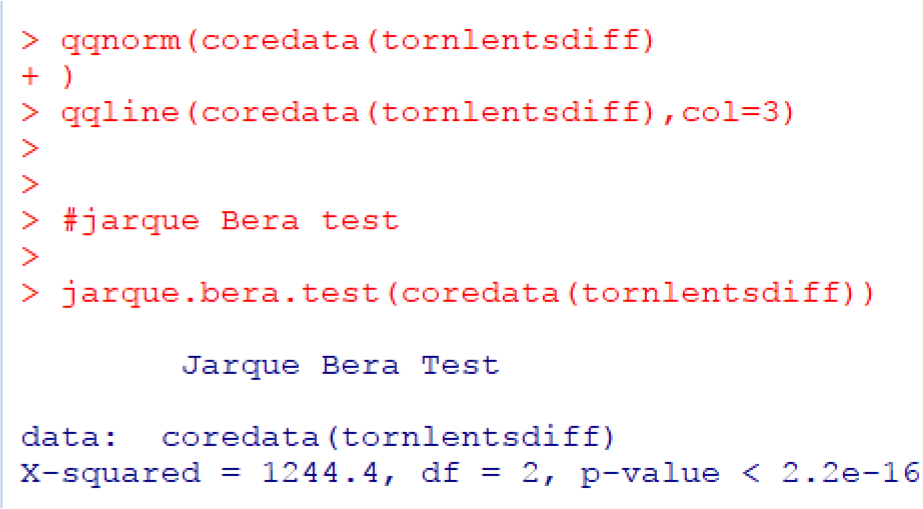
The preliminary analysis process should be now followed for this differenced time series object as well.

Normality tests:

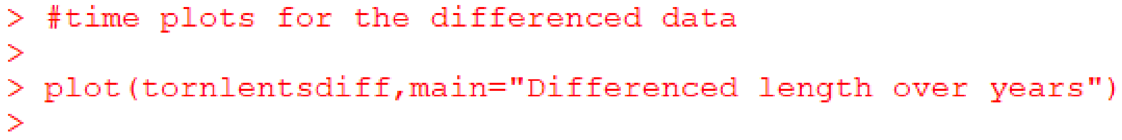
We plotted the histogram, QQ plot and performed Jarque Bera tests on the first differencing applied time series object as shown below:



```
> #histogram
>
>
> hist(tornlentsdiff,xlab="Average tornado diff length",prob=TRUE,main="Histogram of tornado diff length over years")
> xfit<-seq(min(tornlentsdiff),max(tornlentsdiff),length=40)
> yfit<-dnorm(xfit,mean=mean(tornlentsdiff),sd=sd(tornlentsdiff))
> lines(xfit,yfit,col='green',lwd=2)
>
>
```

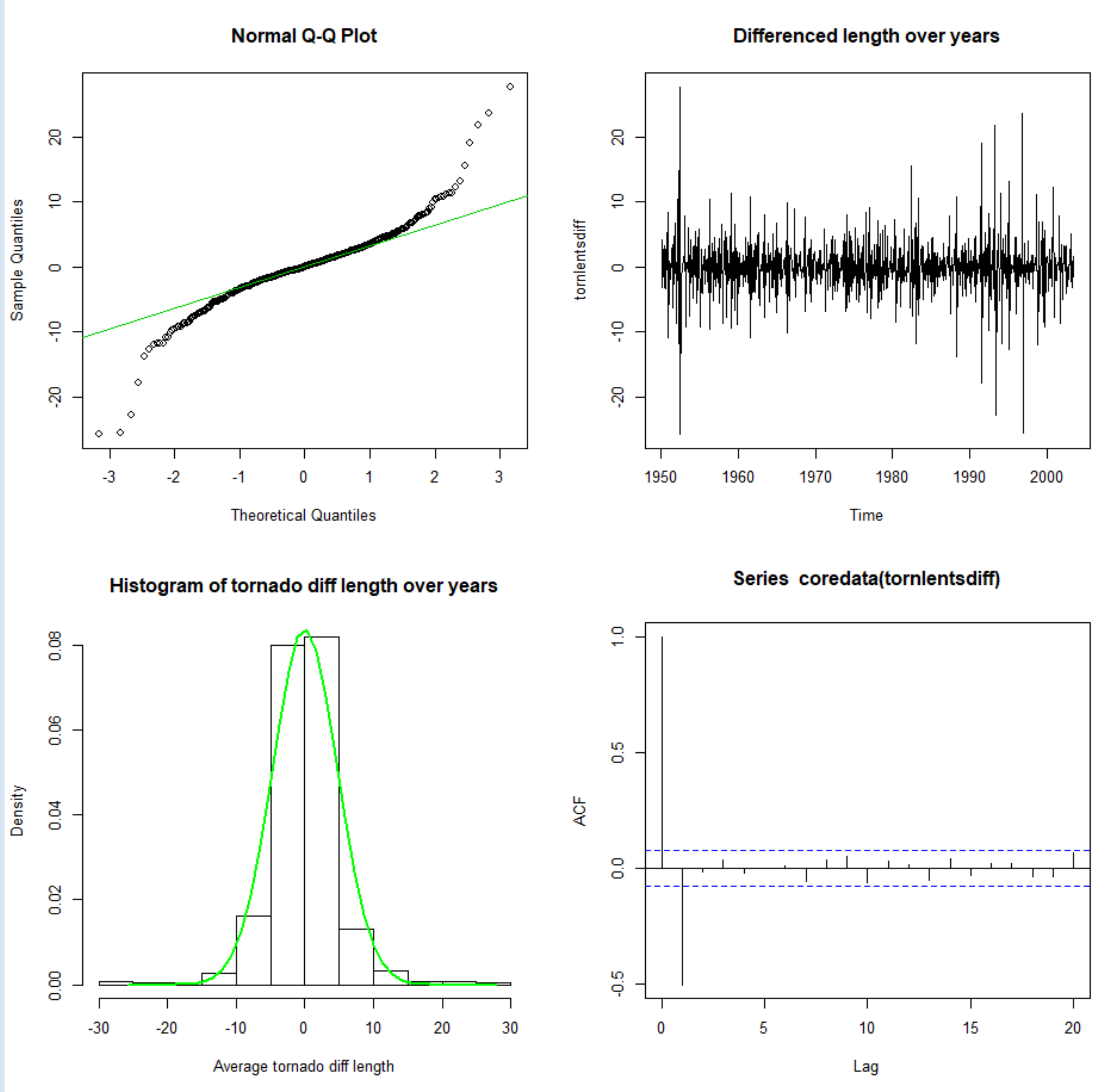


```
> qqnorm(coredata(tornlentsdiff)
+ )
> qqline(coredata(tornlentsdiff),col=3)
>
>
> #jarque Bera test
>
> jarque.bera.test(coredata(tornlentsdiff))

        Jarque Bera Test

data:  coredata(tornlentsdiff)
X-squared = 1244.4, df = 2, p-value < 2.2e-16
```



```
> #time plots for the differenced data
>
> plot(tornlentsdiff,main="Differenced length over years")
>
```

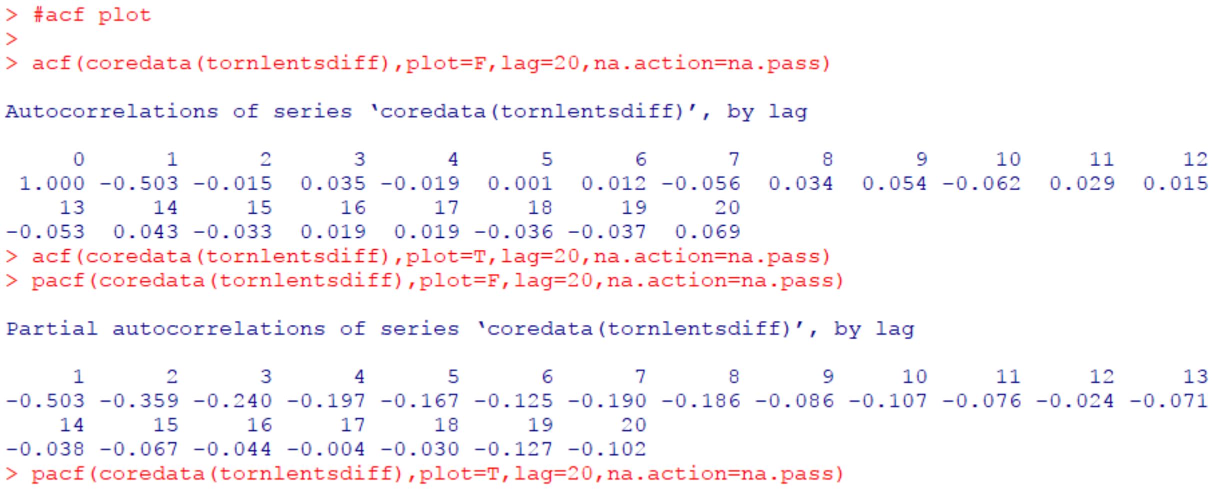
Normality test results:



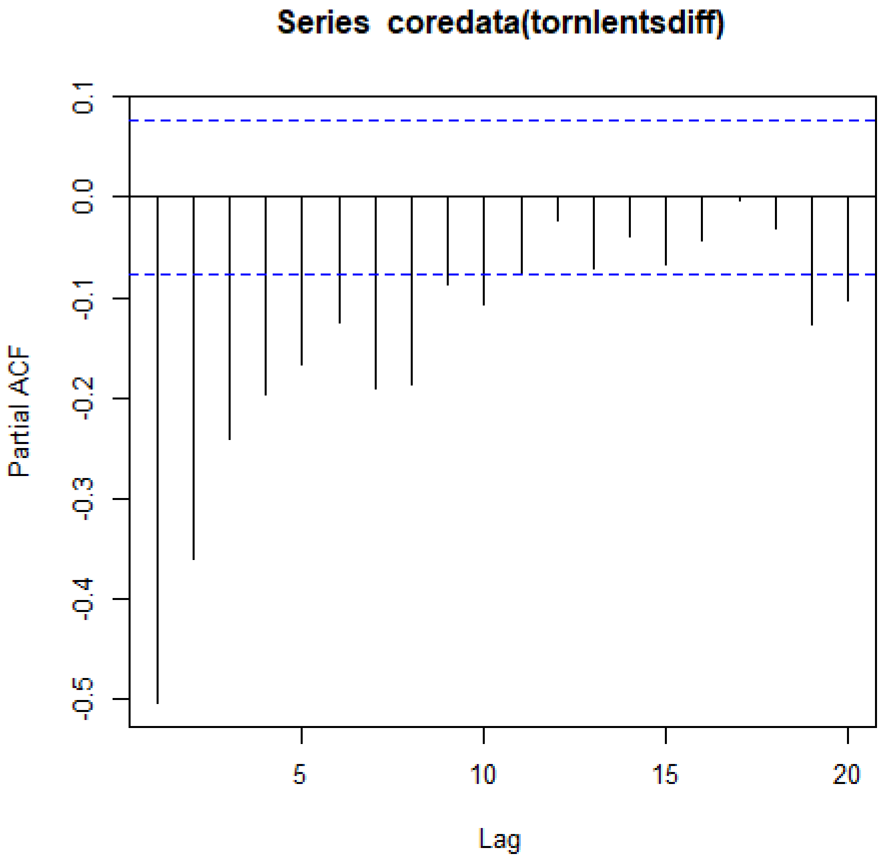
We could see from the normality plots and test results that the **differencing applied length data now follows normal distribution meeting the first assumption required for time series analysis**

Stationarity:

The ACF plot given above shows that there is quick decay of the acf values and there is atleast one non-zero auto-correlation value confirming that the differencing applied length data meets the stationarity assumption of time series analysis. The code used and the pacf plot of the differenced data is as given below.

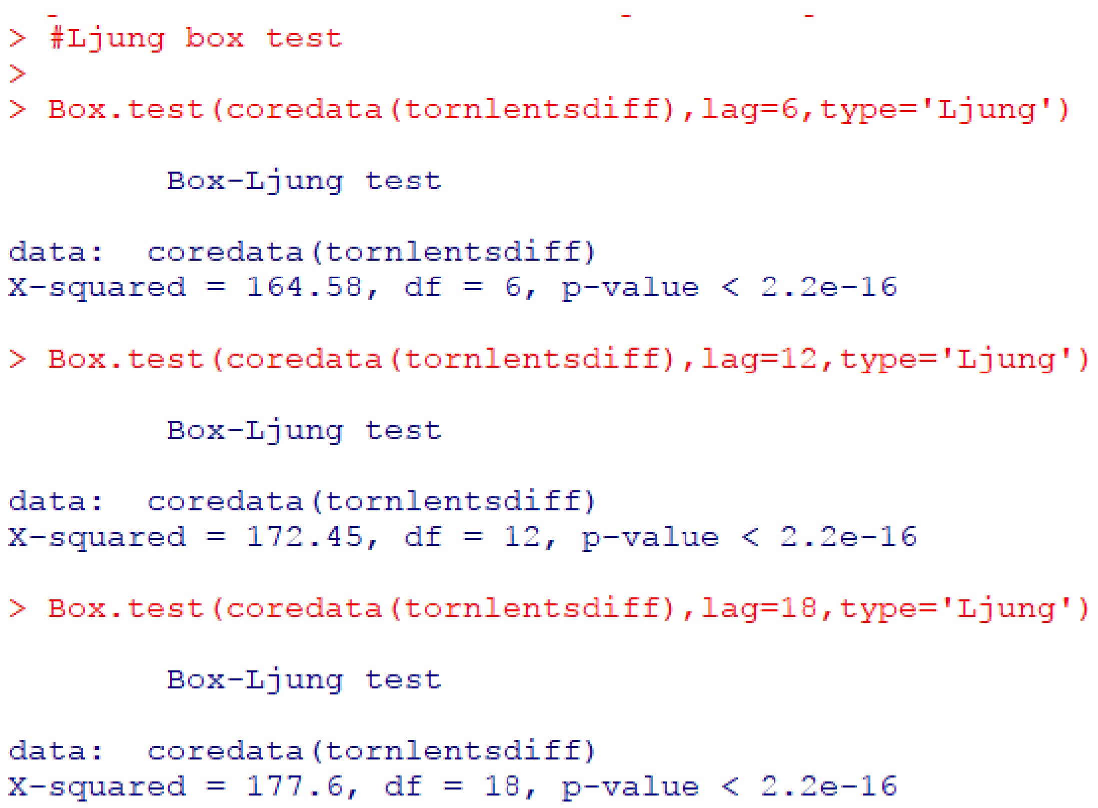


```
> #acf plot
> 
> acf(coredata(tornlentsdiff),plot=F,lag=20,na.action=na.pass)

Autocorrelations of series ‘coredata(tornlentsdiff)’, by lag

     0      1      2      3      4      5      6      7      8      9     10     11     12 
 1.000 -0.503 -0.015  0.035 -0.019  0.001  0.012 -0.056  0.034  0.054 -0.062  0.029  0.015 
    13     14     15     16     17     18     19     20 
-0.053  0.043 -0.033  0.019  0.019 -0.036 -0.037  0.069 
> acf(coredata(tornlentsdiff),plot=T,lag=20,na.action=na.pass)
> pacf(coredata(tornlentsdiff),plot=F,lag=20,na.action=na.pass)

Partial autocorrelations of series ‘coredata(tornlentsdiff)’, by lag

     1      2      3      4      5      6      7      8      9     10     11     12     13 
-0.503 -0.359 -0.240 -0.197 -0.167 -0.125 -0.190 -0.186 -0.086 -0.107 -0.076 -0.024 -0.071 
    14     15     16     17     18     19     20 
-0.038 -0.067 -0.044 -0.004 -0.030 -0.127 -0.102 
> pacf(coredata(tornlentsdiff),plot=T,lag=20,na.action=na.pass)
```



Serial correlation:

We performed the Ljung box test on the first differencing applied tornadoes length data as shown below



```
> #Ljung box test
>
> Box.test(coredata(tornlentsdiff),lag=6,type='Ljung')

        Box-Ljung test

data:  coredata(tornlentsdiff)
X-squared = 164.58, df = 6, p-value < 2.2e-16

> Box.test(coredata(tornlentsdiff),lag=12,type='Ljung')

        Box-Ljung test

data:  coredata(tornlentsdiff)
X-squared = 172.45, df = 12, p-value < 2.2e-16

> Box.test(coredata(tornlentsdiff),lag=18,type='Ljung')

        Box-Ljung test

data:  coredata(tornlentsdiff)
X-squared = 177.6, df = 18, p-value < 2.2e-16
```

Here, the p-value is less than 0.05 and hence we reject the null hypothesis ‘no serial correlation exists in the time series data and it is white noise . Hence, we can conclude that the differencing applied time series data is serial correlated.

Hence, **the first differencing applied tornadoes length data now satisfies all the three assumptions – normal distribution, stationarity and serial correlation – of the time series modelling**.

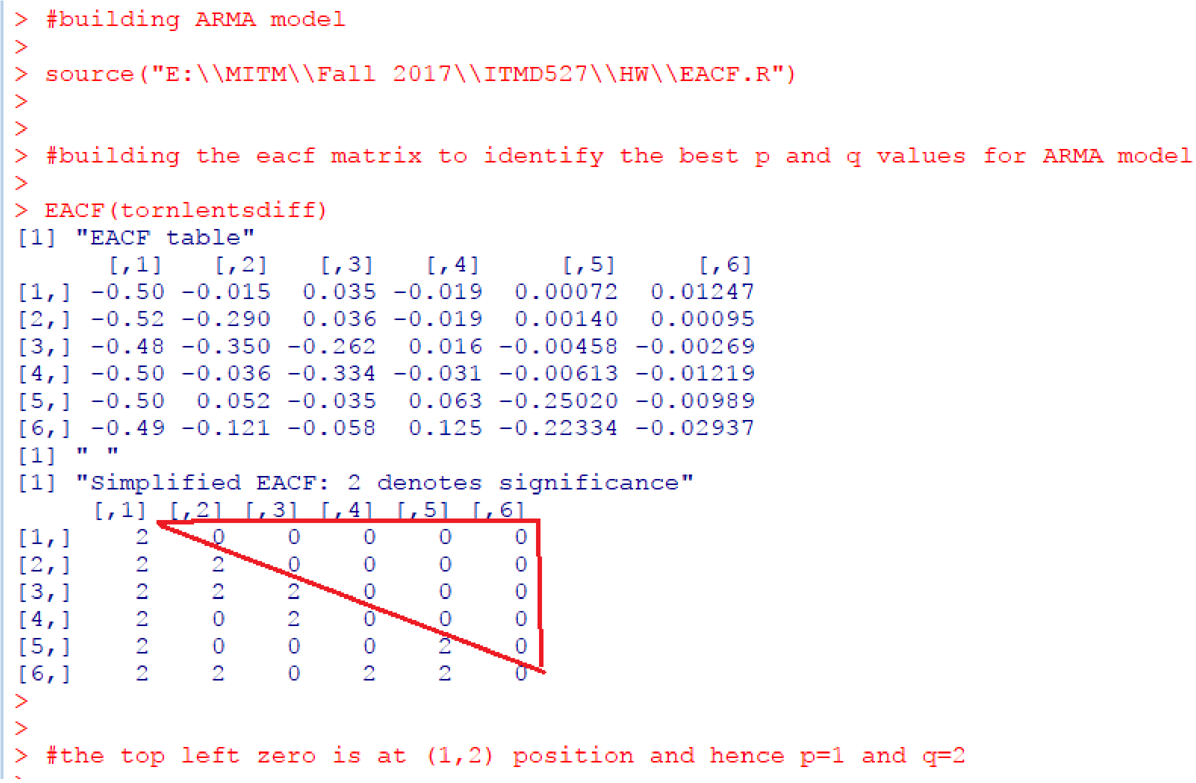
The next step in time series is to identify the p and q values from the PACF and ACF plots respectively to build models

Identify p and q values for building time series models – AR/MA/ARMA:

From the ACF plot, we could see that after lag1 the successive lags are all within the zero bounds of the ACF plot. The ACF plot cuts off at q. Hence the **q value is identified to be 1**.

From the PACF plot, we could see that after lag10, the successive lags are all within the zero bounds of the PACF plot. The PACF plot cuts off at lag p. Hence the **p value is identified to be 10.**

We can also identify p and q values for building ARMA models automatically using the EACF plot as shown below:



```
> #building ARMA model
>
> source("E:\\MITM\\Fall 2017\\ITMD527\\HW\\EACF.R")
>
>
> #building the eacf matrix to identify the best p and q values for ARMA model
>
> EACF(tornlentsdiff)
[1] "EACF table"
      [,1]   [,2]   [,3]   [,4]     [,5]     [,6]
[1,] -0.50 -0.015  0.035 -0.019  0.00072  0.01247
[2,] -0.52 -0.290  0.036 -0.019  0.00140  0.00095
[3,] -0.48 -0.350 -0.262  0.016 -0.00458 -0.00269
[4,] -0.50 -0.036 -0.334 -0.031 -0.00613 -0.01219
[5,] -0.50  0.052 -0.035  0.063 -0.25020 -0.00989
[6,] -0.49 -0.121 -0.058  0.125 -0.22334 -0.02937
[1] " "
[1] "Simplified EACF: 2 denotes significance"
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    2    0    0    0    0    0
[2,]    2    2    0    0    0    0
[3,]    2    2    2    0    0    0
[4,]    2    0    2    0    0    0
[5,]    2    0    0    0    2    0
[6,]    2    2    0    2    2    0
>
>
> #the top left zero is at (1,2) position and hence p=1 and q=2
```

Build Models:

We will now build Auto-Regressive (AR(p)), Moving Average(MA(q)), ARMA(p,q) models with the order values (p & q) we identified from the ACF,PACF and EACF plots respectively.

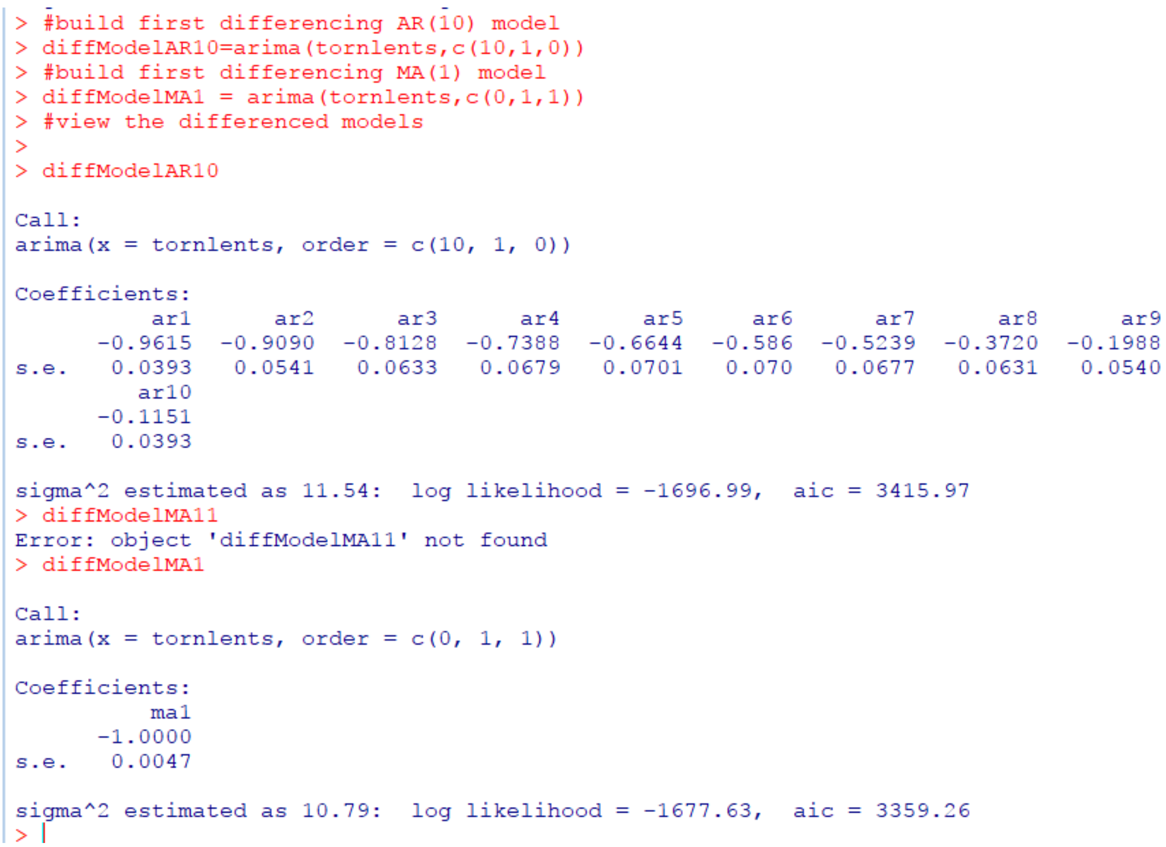
We will build all the models using ARIMA () function available in R.

Note: since we cannot use the model built using ARMA () function for prediction purposes, we will build the ARMA model also using the ARIMA () function in R.

Since p=10 from PACF plot and q = 1 from ACF plot, we will be building AR (10) and MA (1) models as shown below:

AR (10) and MA (1) models:

Since we have applied differencing, the order of AR (10) model will be (10,1,0) and MA(1) will be (0,1,1) as shown below



```
> #build first differencing AR(10) model
> diffModelAR10=arima(tornlents,c(10,1,0))
> #build first differencing MA(1) model
> diffModelMA1 = arima(tornlents,c(0,1,1))
> #view the differenced models
>
> diffModelAR10

Call:
arima(x = tornlents, order = c(10, 1, 0))

Coefficients:
          ar1      ar2      ar3      ar4      ar5     ar6      ar7      ar8      ar9
      -0.9615  -0.9090  -0.8128  -0.7388  -0.6644  -0.586  -0.5239  -0.3720  -0.1988
s.e.   0.0393   0.0541   0.0633   0.0679   0.0701   0.070   0.0677   0.0631   0.0540
         ar10
      -0.1151
s.e.   0.0393

sigma^2 estimated as 11.54:  log likelihood = -1696.99,  aic = 3415.97
> diffModelMA11
Error: object 'diffModelMA11' not found
> diffModelMA1

Call:
arima(x = tornlents, order = c(0, 1, 1))

Coefficients:
          ma1
      -1.0000
s.e.   0.0047

sigma^2 estimated as 10.79:  log likelihood = -1677.63,  aic = 3359.26
> |
```

Once the models are built, we should analyze the residuals of every model that is built to check if the residuals meet the below requirements:

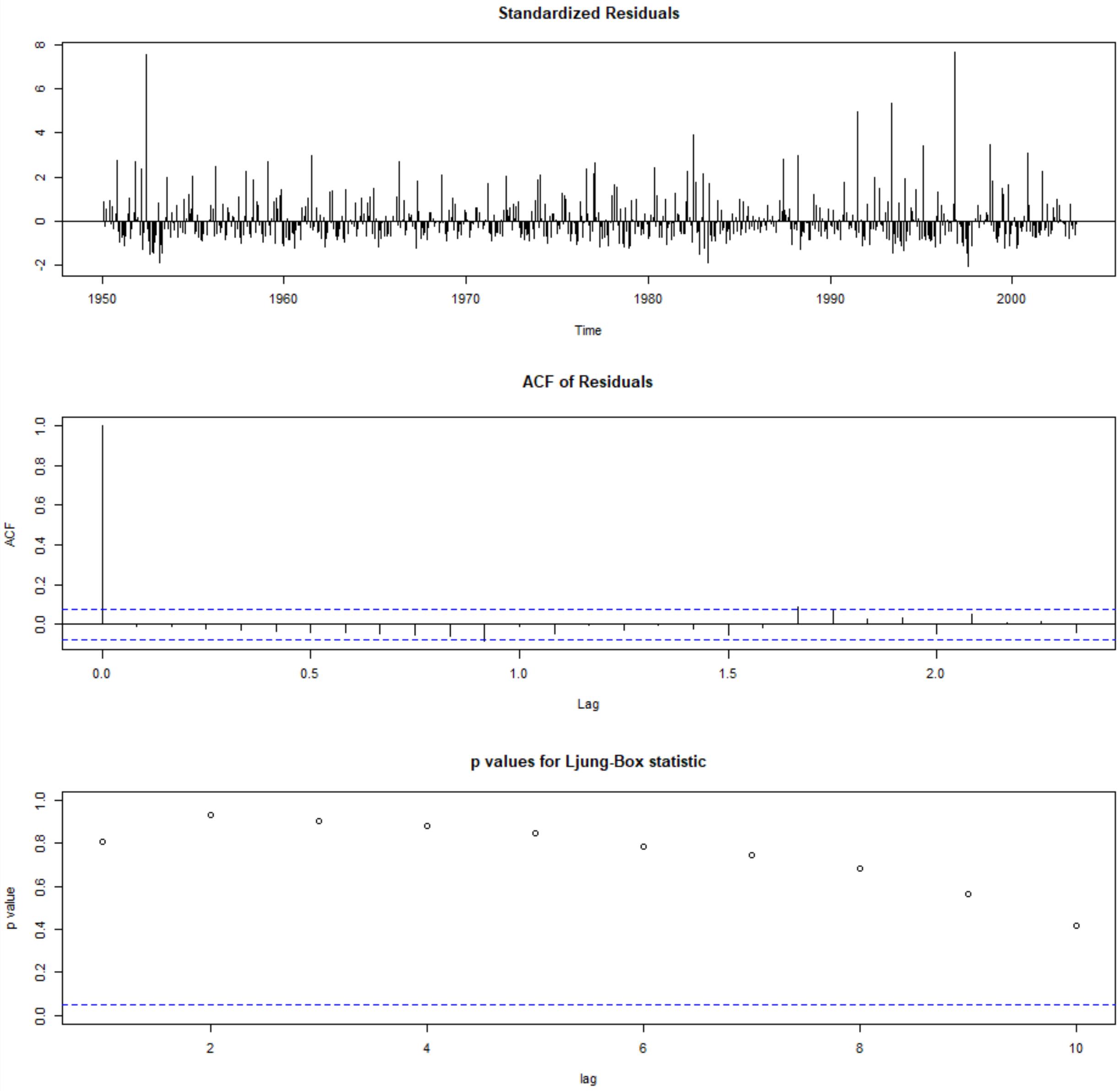
Residuals must follow normal distribution

Residuals must be white noise.

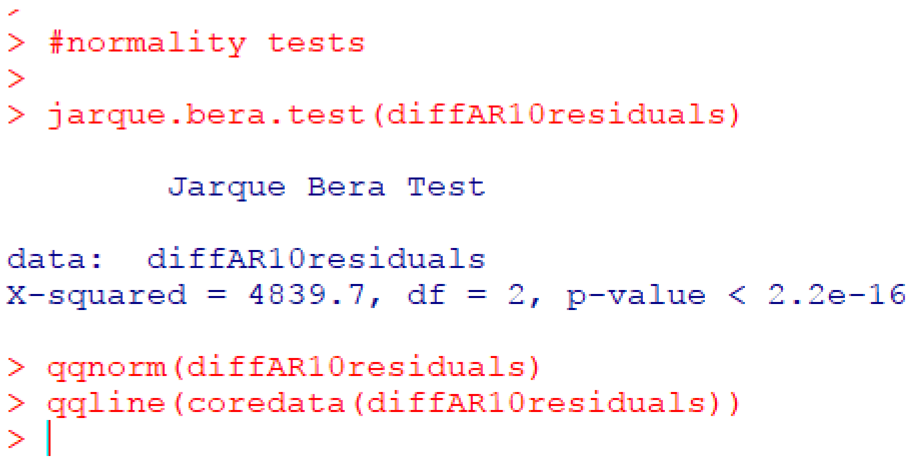
We test all these assumptions of residuals using the residual plots, QQ Plot, jarque Bera test and Ljung Box test.

These are carried out as shown below.

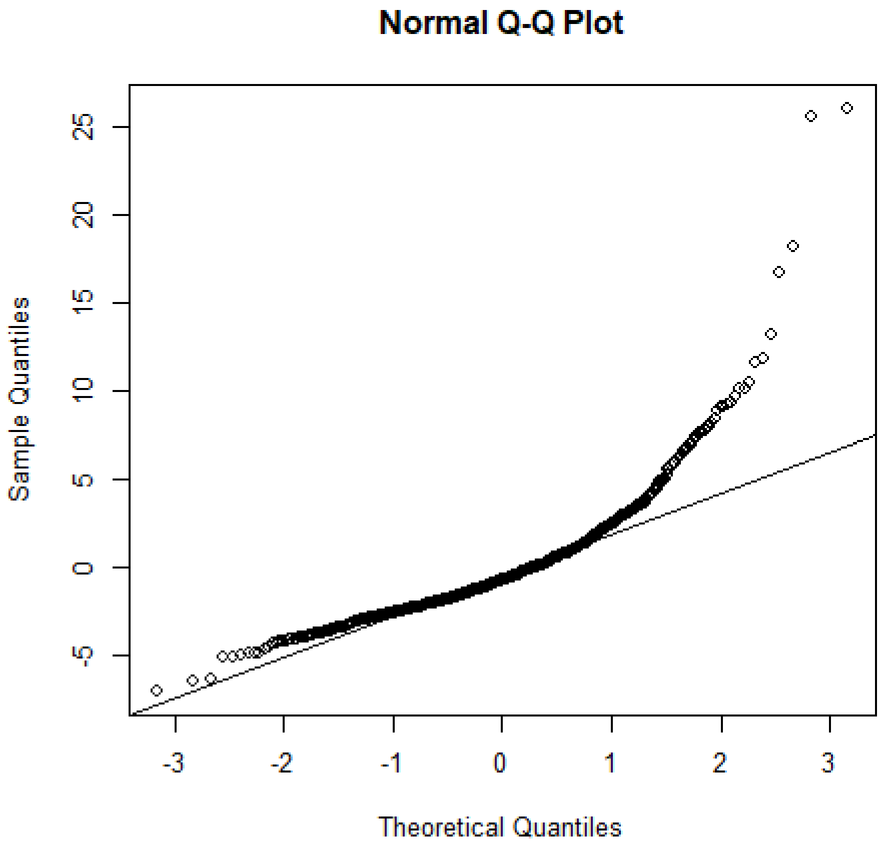
Residual plots of AR (10) model:



Normality tests of AR (10) model residuals:

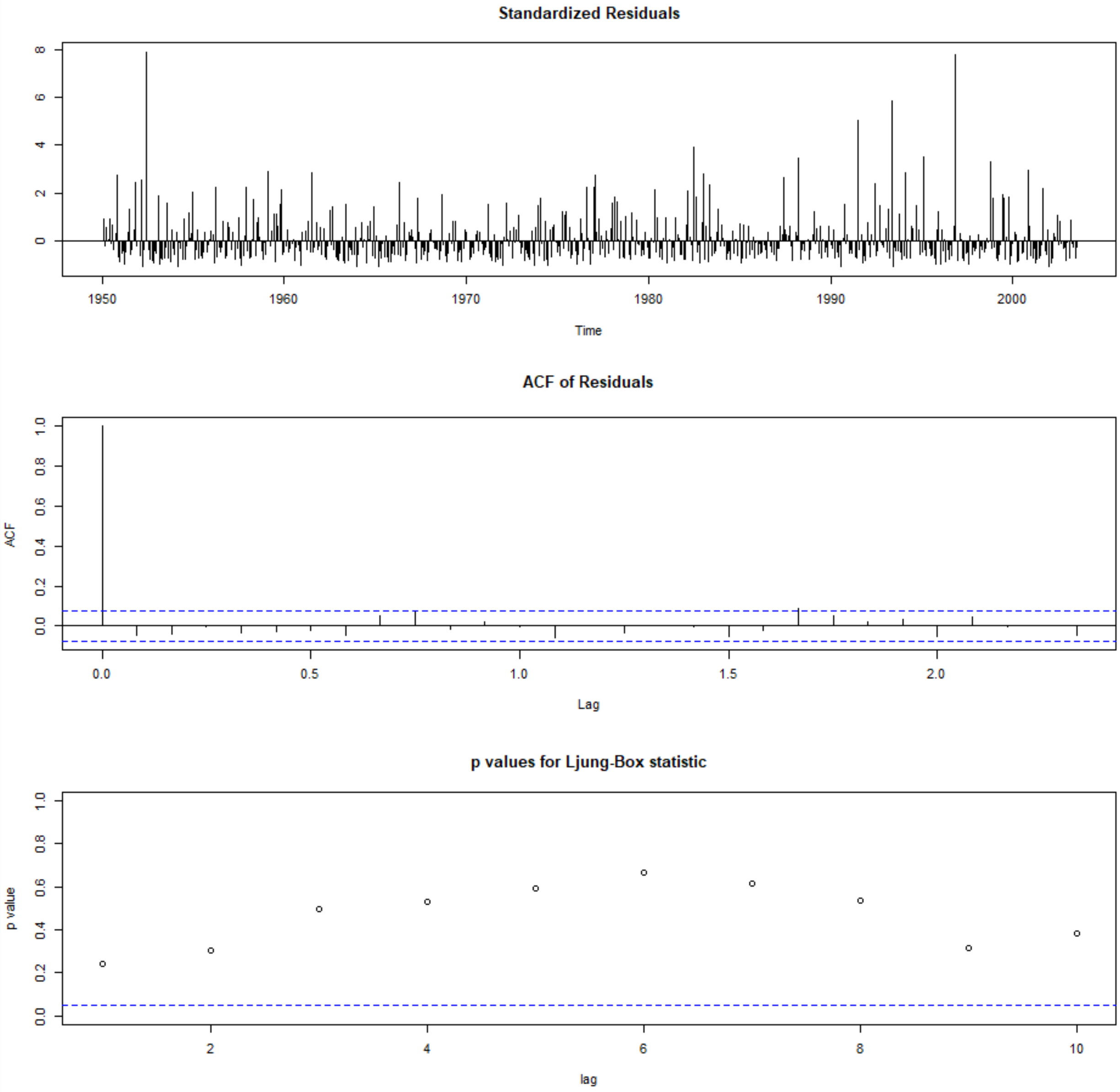


```
> #normality tests
>
> jarque.bera.test(diffAR10residuals)

        Jarque Bera Test

data:  diffAR10residuals
X-squared = 4839.7, df = 2, p-value < 2.2e-16

> qqnorm(diffAR10residuals)
> qqline(coredata(diffAR10residuals))
> |
```

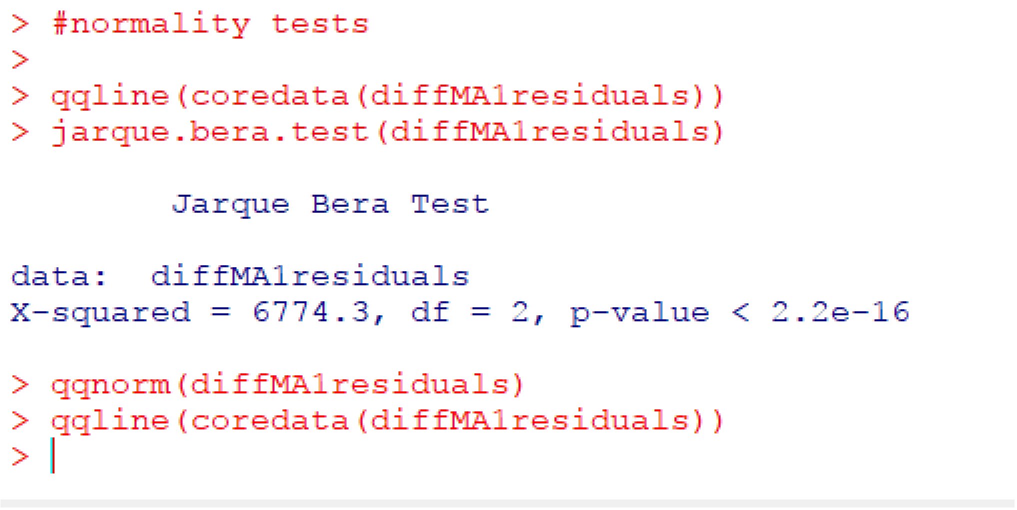


From the above plots and the p-value < 0.05 in the jarque bera test, **we could see that the residuals of AR (10) model follows normal distribution.**

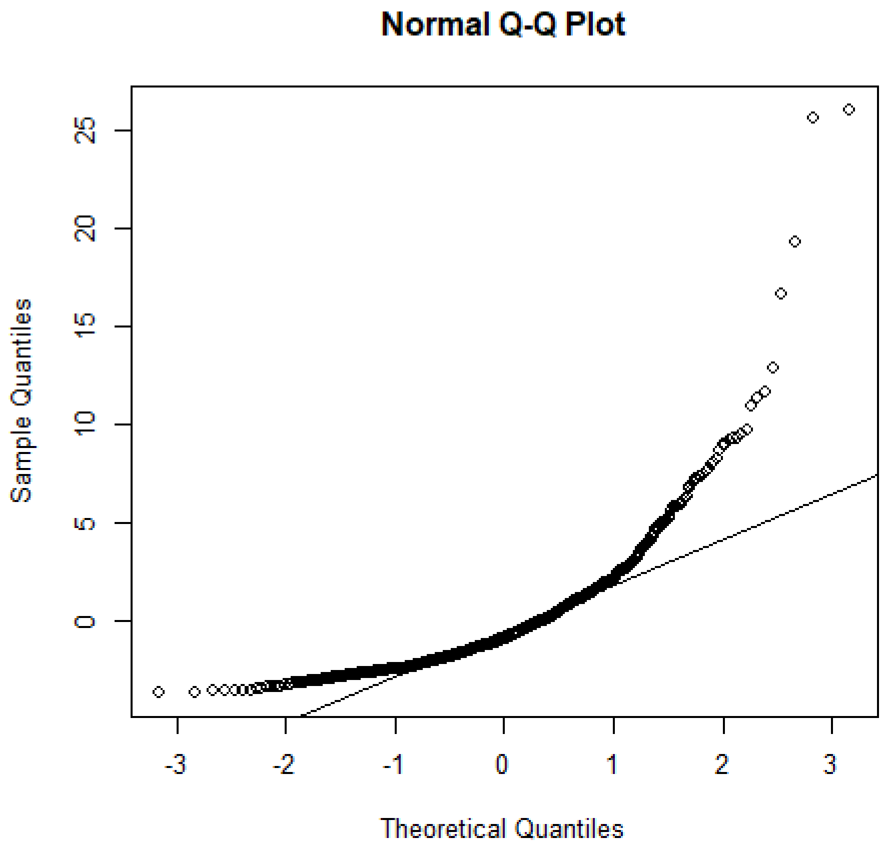
Residual plots of MA (1) model:



Normality tests of MA (1) model



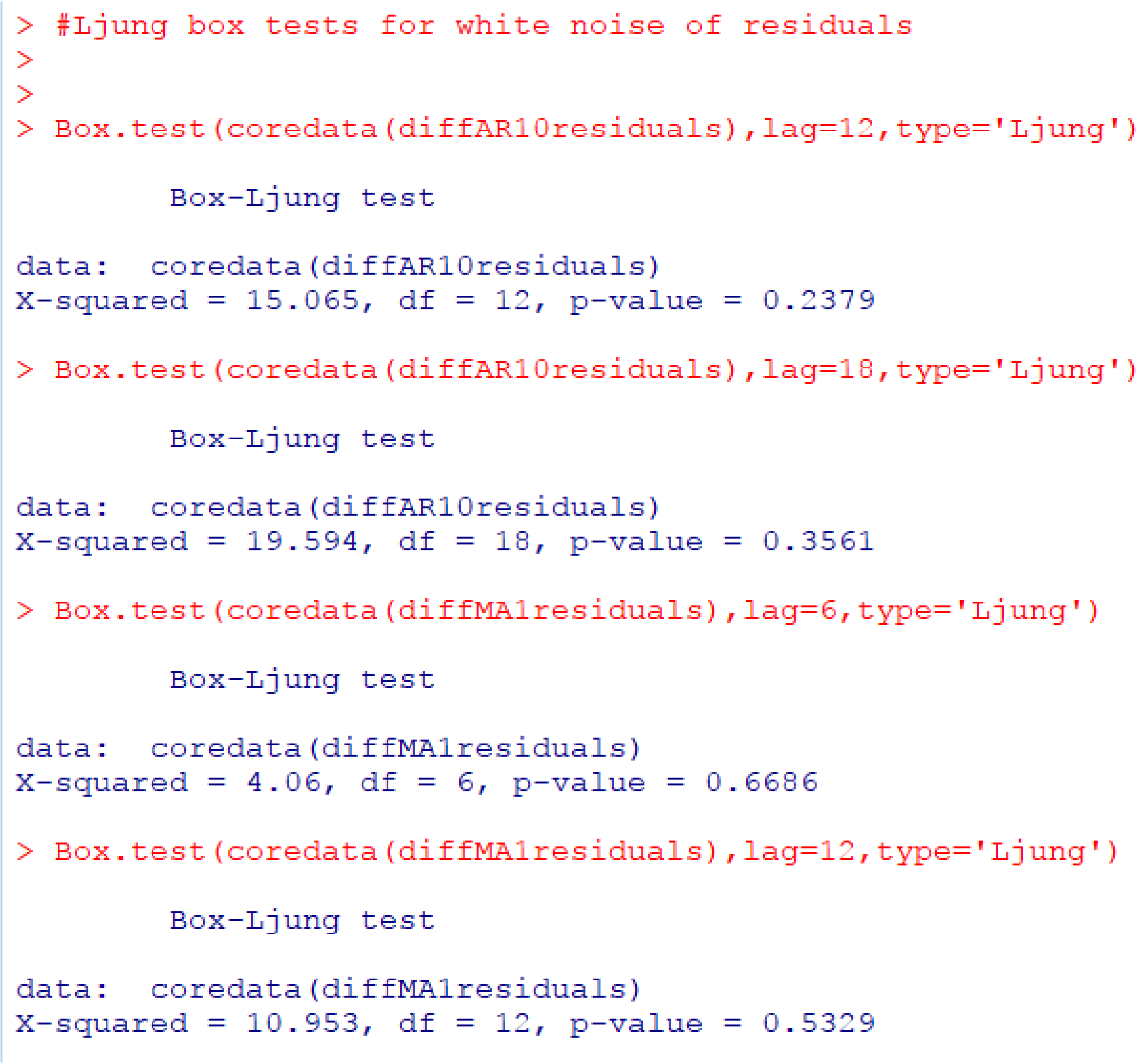
```
> #normality tests
>
> qqline(coredata(diffMA1residuals))
> jarque.bera.test(diffMA1residuals)

        Jarque Bera Test

data:  diffMA1residuals
X-squared = 6774.3, df = 2, p-value < 2.2e-16

> qqnorm(diffMA1residuals)
> qqline(coredata(diffMA1residuals))
> |
```



From the above plots and p-value < 0.05 in Jarque Bera test, we could see that **the residuals of MA (1) model also follow normal distribution.**

We will now see if the residuals are white noise or not for these two models.

Ljung Box test for residuals white noise:



```
> #Ljung box tests for white noise of residuals
>
>
> Box.test(coredata(diffAR10residuals),lag=12,type='Ljung')

        Box-Ljung test

data:  coredata(diffAR10residuals)
X-squared = 15.065, df = 12, p-value = 0.2379

> Box.test(coredata(diffAR10residuals),lag=18,type='Ljung')

        Box-Ljung test

data:  coredata(diffAR10residuals)
X-squared = 19.594, df = 18, p-value = 0.3561

> Box.test(coredata(diffMA1residuals),lag=6,type='Ljung')

        Box-Ljung test

data:  coredata(diffMA1residuals)
X-squared = 4.06, df = 6, p-value = 0.6686

> Box.test(coredata(diffMA1residuals),lag=12,type='Ljung')

        Box-Ljung test

data:  coredata(diffMA1residuals)
X-squared = 10.953, df = 12, p-value = 0.5329
```

The hypothesis statements in Ljung Box test for residuals are as given below

H_0: Residuals are not white noise

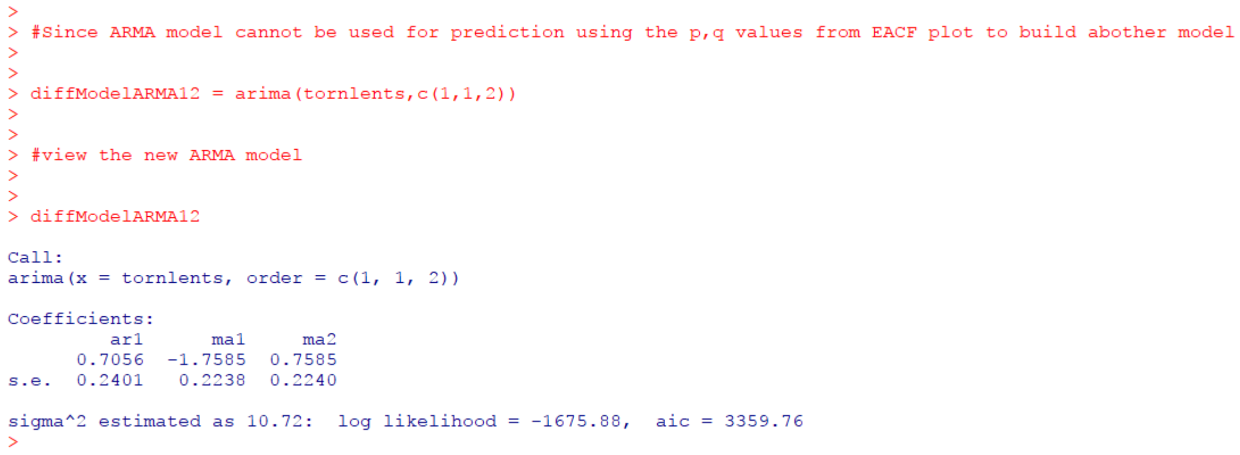
H_a: Residuals are white noise

As shown in the above plot, the p-values are more than 0.05 at both lags for both the models.

Hence, we do not have enough evidence to accept the above null hypothesis. Hence, we can conclude that **the residuals of both models are white noise**.

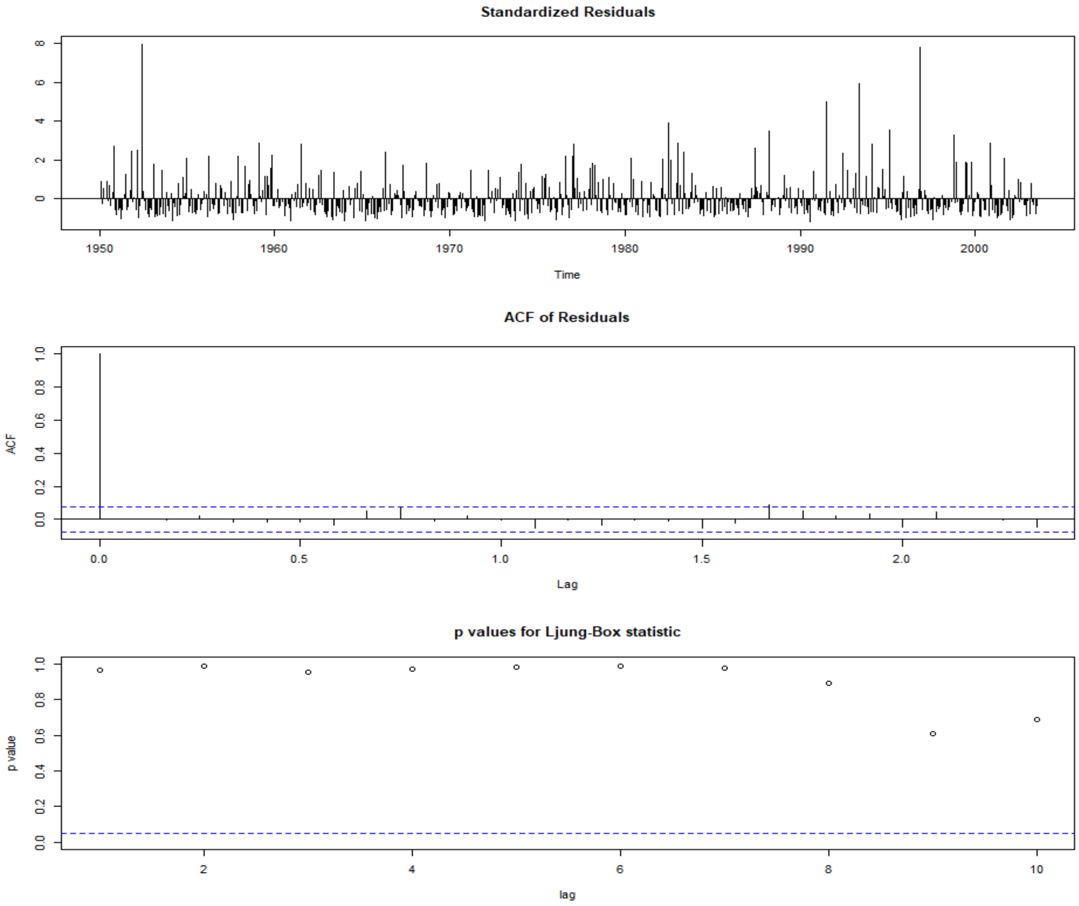
ARMA (1,2) model

From EACF plot, we got the p and q values to 1 and 2. We will now build ARMA (1,2) model as shown below:



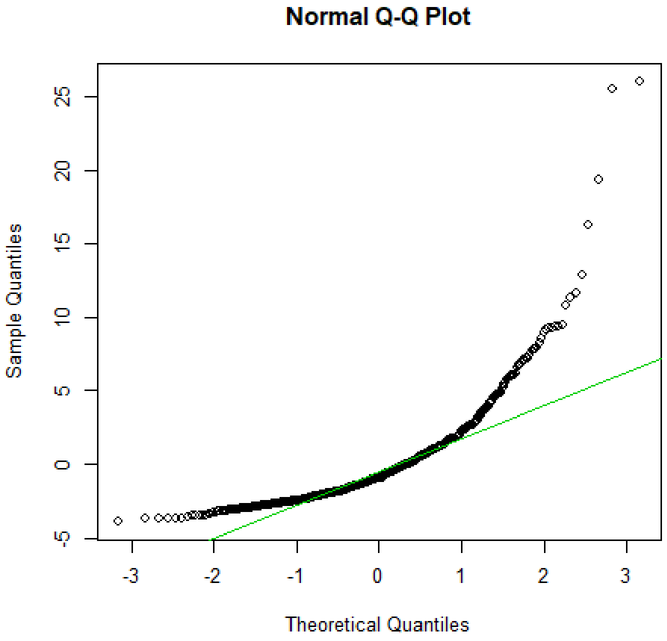
```
>
> #Since ARMA model cannot be used for prediction using the p,q values from EACF plot to build abother model
>
>
> diffModelARMA12 = arima(tornlents,c(1,1,2))
>
>
> #view the new ARMA model
>
>
> diffModelARMA12

Call:
arima(x = tornlents, order = c(1, 1, 2))

Coefficients:
         ar1      ma1     ma2
      0.7056  -1.7585  0.7585
s.e.  0.2401   0.2238  0.2240

sigma^2 estimated as 10.72:  log likelihood = -1675.88,  aic = 3359.76
>
```

Residual plots of ARMA (1,2) model

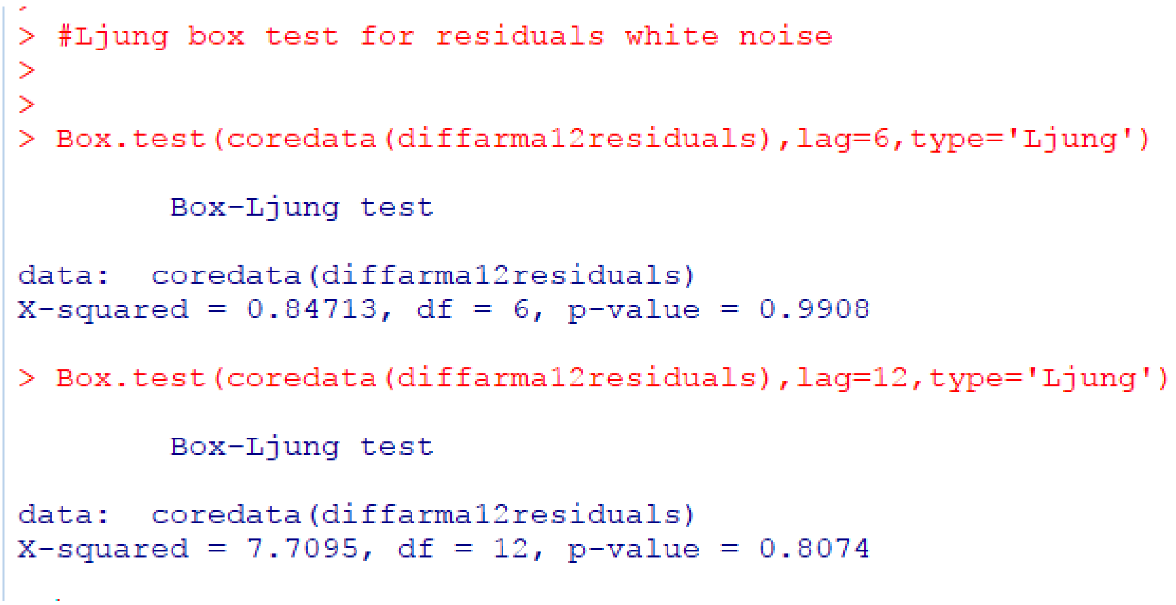


Normality tests of ARMA (1,2) residuals

```
> 
> #residual analysis of new ARMA model
> 
> 
> diffarma12residuals = diffModelARMA12$residuals
> 
> 
> 
> #residual plots
> 
> tsdiag(diffModelARMA12)
> 
> 
> #qqplot
> qqnorm(coredata(diffarma12residuals))
> qqline(coredata(diffarma12residuals),col=3)
> 
> #jarque bera test for normality
> jarque.bera.test(coredata(diffarma12residuals))

        Jarque Bera Test

data:  coredata(diffarma12residuals)
X-squared = 6802.3, df = 2, p-value < 2.2e-16
```



Ljung Box test for ARMA (1,2) model residuals white noise



```
> #Ljung box test for residuals white noise
>
>
> Box.test(coredata(diffarma12residuals),lag=6,type='Ljung')

        Box-Ljung test

data:  coredata(diffarma12residuals)
X-squared = 0.84713, df = 6, p-value = 0.9908

> Box.test(coredata(diffarma12residuals),lag=12,type='Ljung')

        Box-Ljung test

data:  coredata(diffarma12residuals)
X-squared = 7.7095, df = 12, p-value = 0.8074
```

It is clear from the QQ plot that most of the residuals are along the QQ line and the p-value in Jarque Bera test is < 0.05. This indicates that **the residuals of ARMA (1,2) model follow normal distribution.**

Also, from the Box test, the p-values are more than 0.05 at both lags. Hence **the residuals of ARMA (1,2) model are white noise.**

A detailed summary of all the 3 models built with the training data is as given below:

Criteria	AR (10)	MA (1)	ARMA (1,2)
AIC	3415.97	3359.26	3359.76
Residuals white noise?	yes	Yes	Yes
Residuals normally distributed	Yes	Yes	Yes
Qualified Model?	Yes	Yes	Yes

6. Evaluations and Results

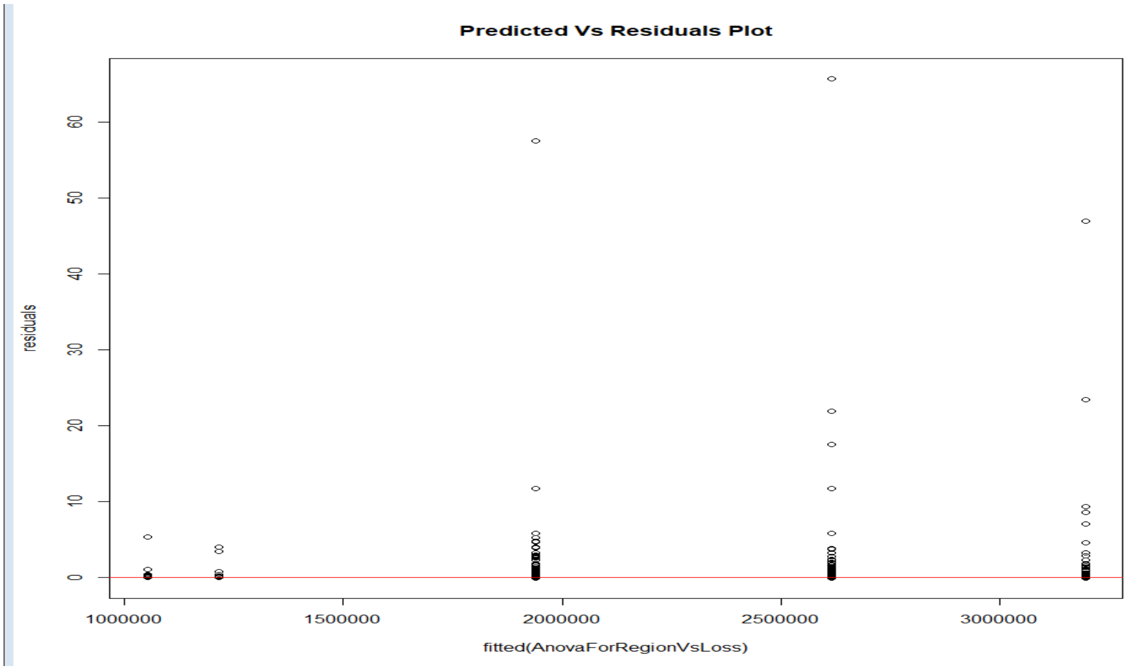
6.1. Evaluation Methods

ANOVA model evaluation (Comparison of property loss incurred by different regions)

Residual Analysis:

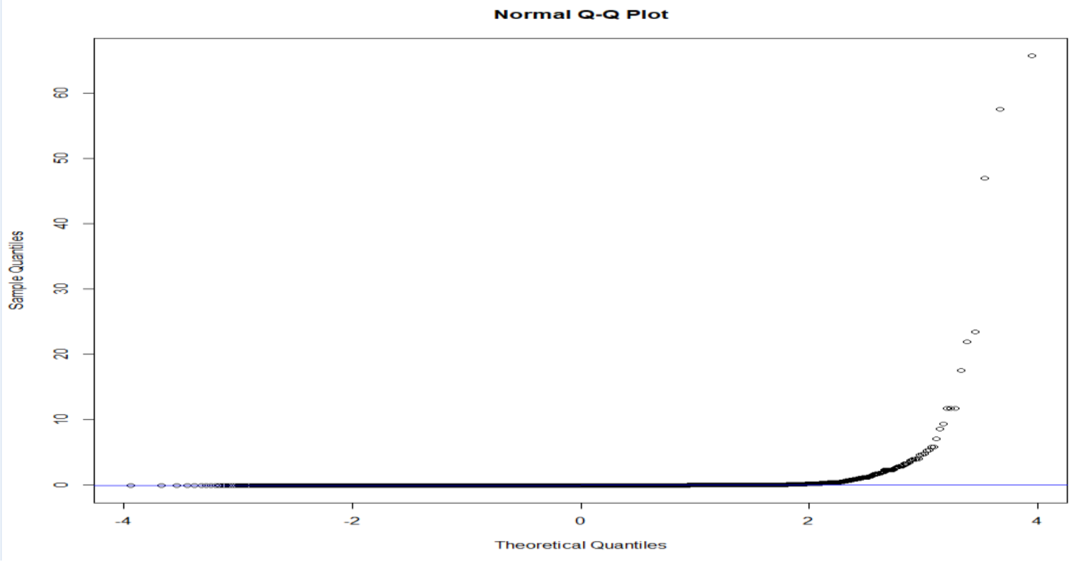
a) Residual plot – is the within group variability constant?
b) Normal Probability plot – does it follow normal distribution?

Predicted vs Residual plot:



Here, the within- group variability is not constant.

Normal Probability plot:



The residuals do not follow normal distribution.

Hence, our model is not sufficient to conclude on the ANOVA test hypothesis.

Hence, we apply build ANOVA model with log transformed loss variable.

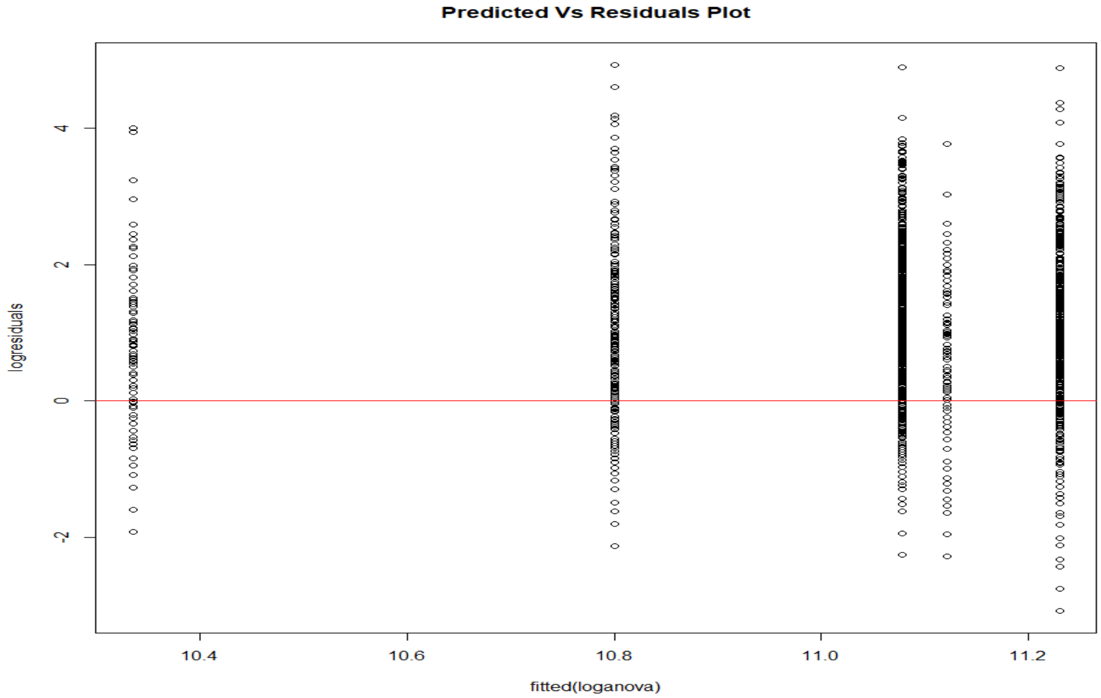
```
> #log transformation
>
>
> loganova=lm(log(loss)~region)
> summary(loganova)

Call:
lm(formula = log(loss) ~ region)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6246 -1.5896 -0.2033  1.1994 10.6165 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     11.22981    0.03460 324.600  < 2e-16 ***
regionNORTHEAST -0.10852    0.10844  -1.001 0.316972    
regionSOUTHEAST -0.15239    0.04456  -3.420 0.000628 ***
regionSOUTHWEST -0.42985    0.06089  -7.060 1.76e-12 ***
regionWEST      -0.89474    0.11760  -7.608 2.98e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.153 on 12401 degrees of freedom
Multiple R-squared:  0.00747,   Adjusted R-squared:  0.00715 
F-statistic: 23.33 on 4 and 12401 DF,  p-value: < 2.2e-16
```

F-statistic is 23.33 (which is greater than the previous model) with p-value < 0.05. This indicates that we can reject null hypothesis at 95% confidence interval.

Also, t-test on individual parameters suggest that the difference in means in not significant for NorthEast region with respect to MidWest as p-value>0.05. But, it is significant for SouthEast, SouthWest and West with respect to MidWest as p-values>0.05.

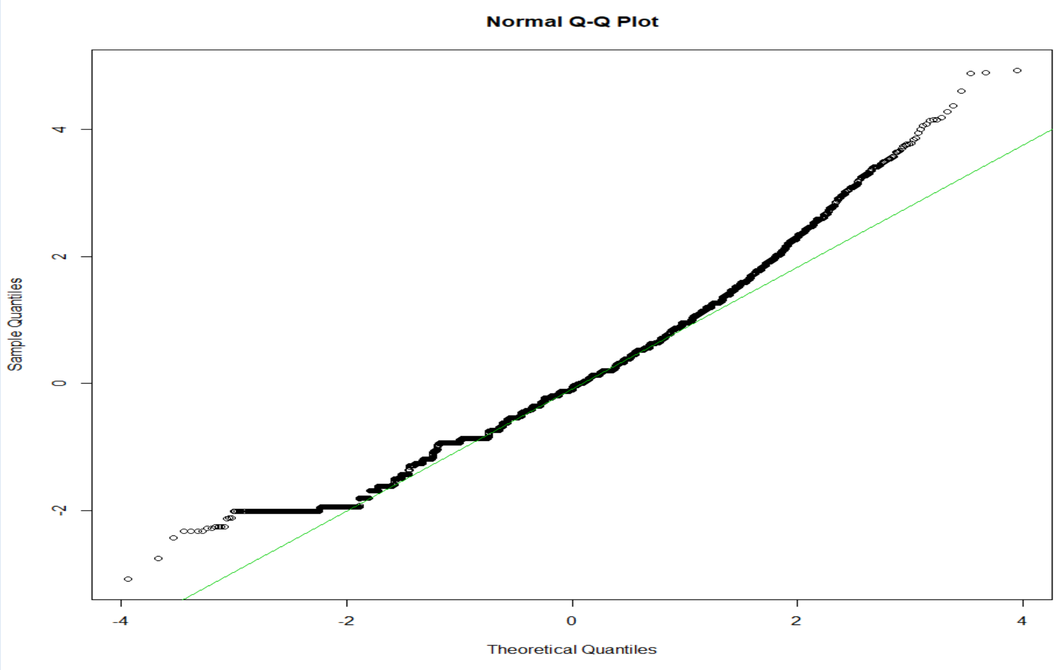
Model evaluation for the log transformed ANOVA model:

Residual analysis:



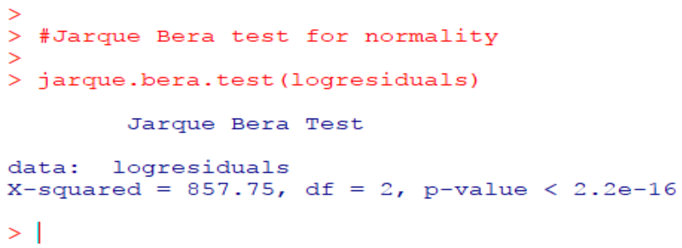
The within group variability is now constant.

Normal Probability plot



The residuals follow normal distribution.

To confirm we do Jarque-Bera test for Normality



```
>
> #Jarque Bera test for normality
>
> jarque.bera.test(logresiduals)

        Jarque Bera Test

data:  logresiduals
X-squared = 857.75, df = 2, p-value < 2.2e-16

> |
```

Since, p-value<0.05, we can confirm the residuals follow normal distribution.

Thus, the log transformed model confirms our assumptions on normality and constant variance.

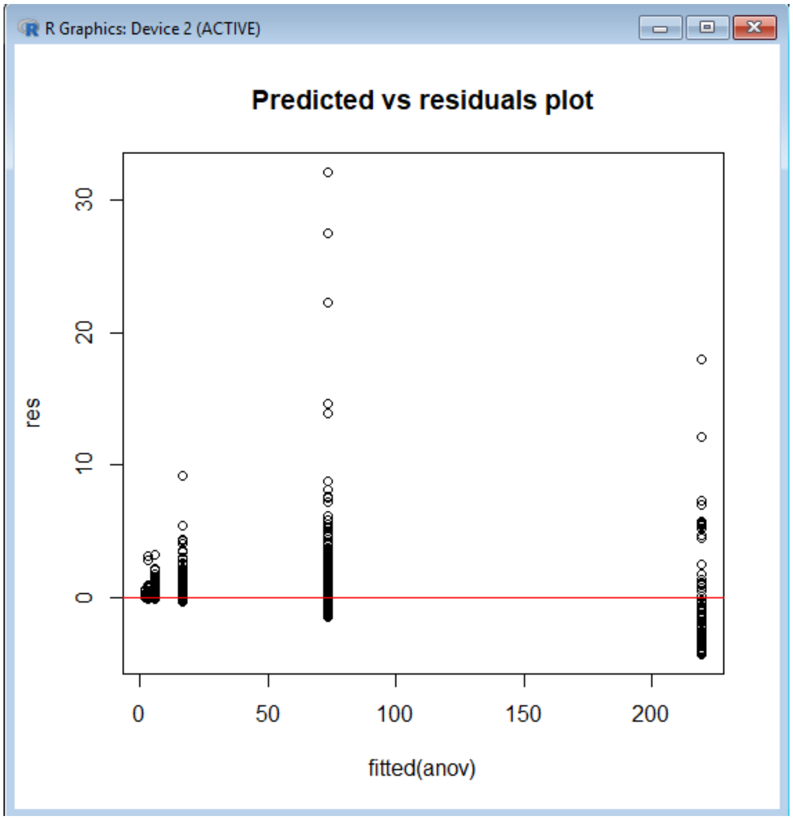
Hence, the log transformed model is the best fitted ANOVA model with which we conclude that we can reject null hypothesis, and the difference in mean loss in some regions are significant.

ANOVA model evaluation (Comparison of injuries due to tornadoes of different magnitudes)

Residual Analysis:

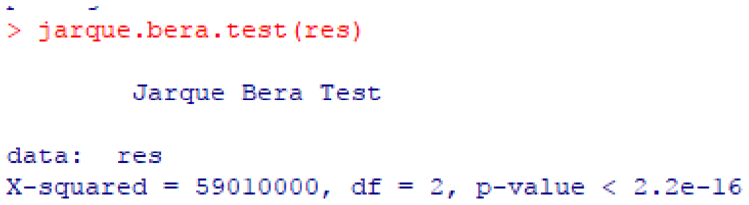
a) Residual plot – is the within group variability constant?
b) Normal Probability plot – does it follow normal distribution?

Predicted vs Residual plot:



The within group variance is almost constant.

Test for normality:



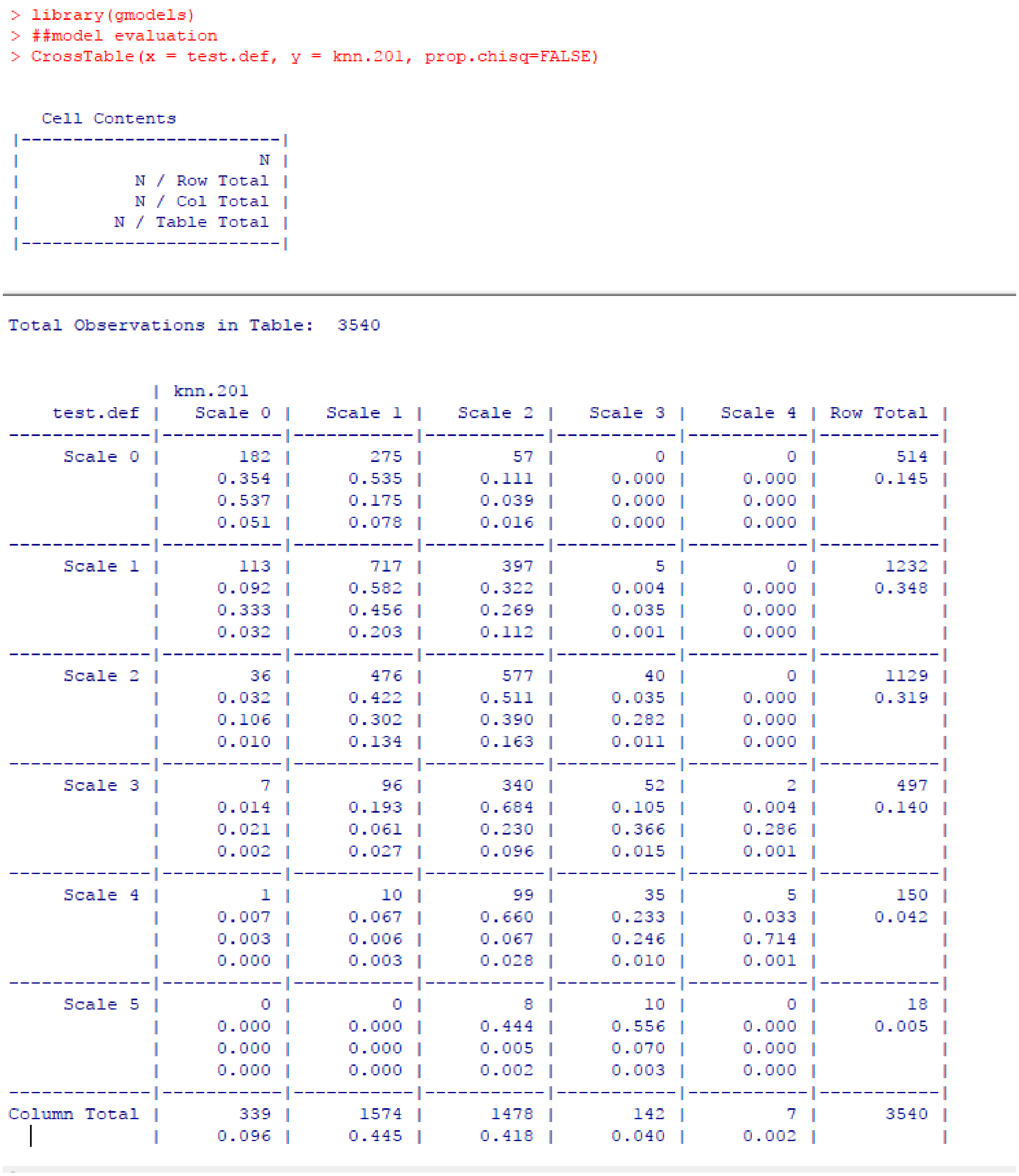
```
> jarque.bera.test(res)

        Jarque Bera Test

data:  res
X-squared = 59010000, df = 2, p-value < 2.2e-16
```

Since p<0.05 for Jarque Bera test on residuals, we say the residuals follow normal distribution.

Hence, our model is sufficient to conclude on the ANOVA test hypothesis and we reject null hypothesis, i.e., the difference in injuries is significant for some mag categories.

KNN Classification Model Evaluation (Group Magnitude in terms of Length, width and distance travelled by a tornado)

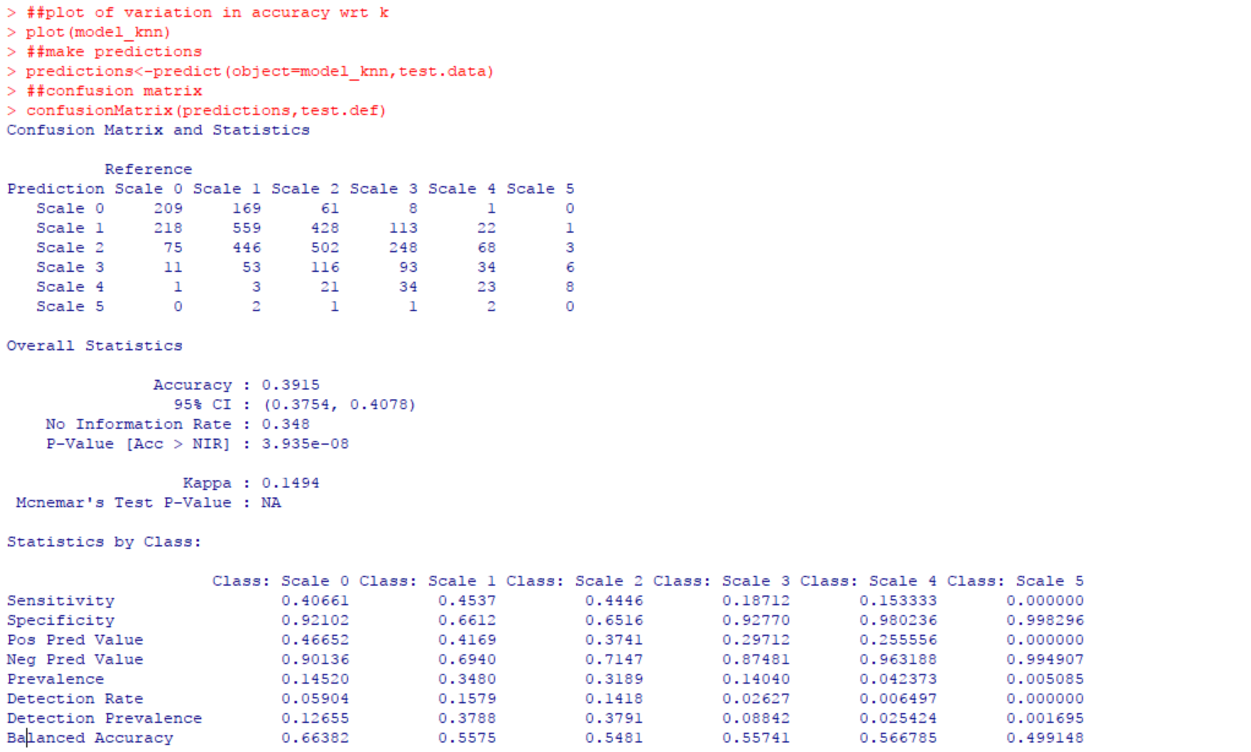
We evaluate the kNN model, knn.201 obtained for k=201 which had better accuracy.



```
> library(gmodels)
> ##model evaluation
> CrossTable(x = test.def, y = knn.201, prop.chisq=FALSE)


   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  3540


             | knn.201
    test.def |   Scale 0 |   Scale 1 |   Scale 2 |   Scale 3 |   Scale 4 | Row Total |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
     Scale 0 |       182 |       275 |        57 |         0 |         0 |       514 |
             |     0.354 |     0.535 |     0.111 |     0.000 |     0.000 |     0.145 |
             |     0.537 |     0.175 |     0.039 |     0.000 |     0.000 |           |
             |     0.051 |     0.078 |     0.016 |     0.000 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
     Scale 1 |       113 |       717 |       397 |         5 |         0 |      1232 |
             |     0.092 |     0.582 |     0.322 |     0.004 |     0.000 |     0.348 |
             |     0.333 |     0.456 |     0.269 |     0.035 |     0.000 |           |
             |     0.032 |     0.203 |     0.112 |     0.001 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
     Scale 2 |        36 |       476 |       577 |        40 |         0 |      1129 |
             |     0.032 |     0.422 |     0.511 |     0.035 |     0.000 |     0.319 |
             |     0.106 |     0.302 |     0.390 |     0.282 |     0.000 |           |
             |     0.010 |     0.134 |     0.163 |     0.011 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
     Scale 3 |         7 |        96 |       340 |        52 |         2 |       497 |
             |     0.014 |     0.193 |     0.684 |     0.105 |     0.004 |     0.140 |
             |     0.021 |     0.061 |     0.230 |     0.366 |     0.286 |           |
             |     0.002 |     0.027 |     0.096 |     0.015 |     0.001 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
     Scale 4 |         1 |        10 |        99 |        35 |         5 |       150 |
             |     0.007 |     0.067 |     0.660 |     0.233 |     0.033 |     0.042 |
             |     0.003 |     0.006 |     0.067 |     0.246 |     0.714 |           |
             |     0.000 |     0.003 |     0.028 |     0.010 |     0.001 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
     Scale 5 |         0 |         0 |         8 |        10 |         0 |        18 |
             |     0.000 |     0.000 |     0.444 |     0.556 |     0.000 |     0.005 |
             |     0.000 |     0.000 |     0.005 |     0.070 |     0.000 |           |
             |     0.000 |     0.000 |     0.002 |     0.003 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
Column Total |       339 |      1574 |      1478 |       142 |         7 |      3540 |
             |     0.096 |     0.445 |     0.418 |     0.040 |     0.002 |           |
```

As per above cross table, the prediction results on test data does not seem good, as for example 182 records of magnitude 0 were predicted correctly as mag0, 275 which were actually of mag0 was predicted incorrectly as Scale 1, which is not preferred.

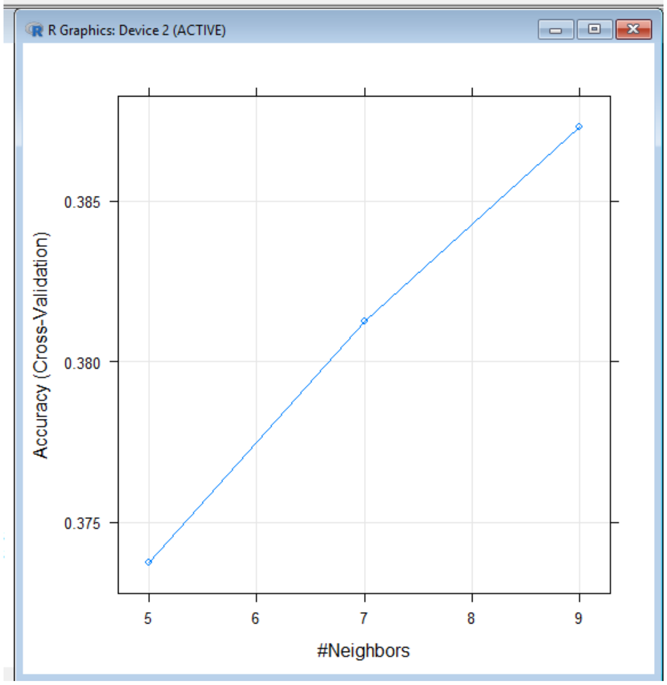
Also, on evaluating data obtained by training and resampling by cross validation, the overall accuracy is less 0.39.



```
> ##plot of variation in accuracy wrt k
> plot(model_knn)
> ##make predictions
> predictions<-predict(object=model_knn,test.data)
> ##confusion matrix
> confusionMatrix(predictions,test.def)
Confusion Matrix and Statistics

          Reference
Prediction Scale 0 Scale 1 Scale 2 Scale 3 Scale 4 Scale 5
   Scale 0     209     169      61       8       1       0
   Scale 1     218     559     428     113      22       1
   Scale 2      75     446     502     248      68       3
   Scale 3      11      53     116      93      34       6
   Scale 4       1       3      21      34      23       8
   Scale 5       0       2       1       1       2       0

Overall Statistics

               Accuracy : 0.3915
                 95% CI : (0.3754, 0.4078)
    No Information Rate : 0.348
    P-Value [Acc > NIR] : 3.935e-08

                  Kappa : 0.1494
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Scale 0 Class: Scale 1 Class: Scale 2 Class: Scale 3 Class: Scale 4 Class: Scale 5
Sensitivity                 0.40661         0.4537         0.4446        0.18712       0.153333       0.000000
Specificity                 0.92102         0.6612         0.6516        0.92770       0.980236       0.998296
Pos Pred Value              0.46652         0.4169         0.3741        0.29712       0.255556       0.000000
Neg Pred Value              0.90136         0.6940         0.7147        0.87481       0.963188       0.994907
Prevalence                  0.14520         0.3480         0.3189        0.14040       0.042373       0.005085
Detection Rate              0.05904         0.1579         0.1418        0.02627       0.006497       0.000000
Detection Prevalence        0.12655         0.3788         0.3791        0.08842       0.025424       0.001695
Balanced Accuracy           0.66382         0.5575         0.5481        0.55741       0.566785       0.499148
```

As per the confusion matrix, the accuracy of predicted data is less.

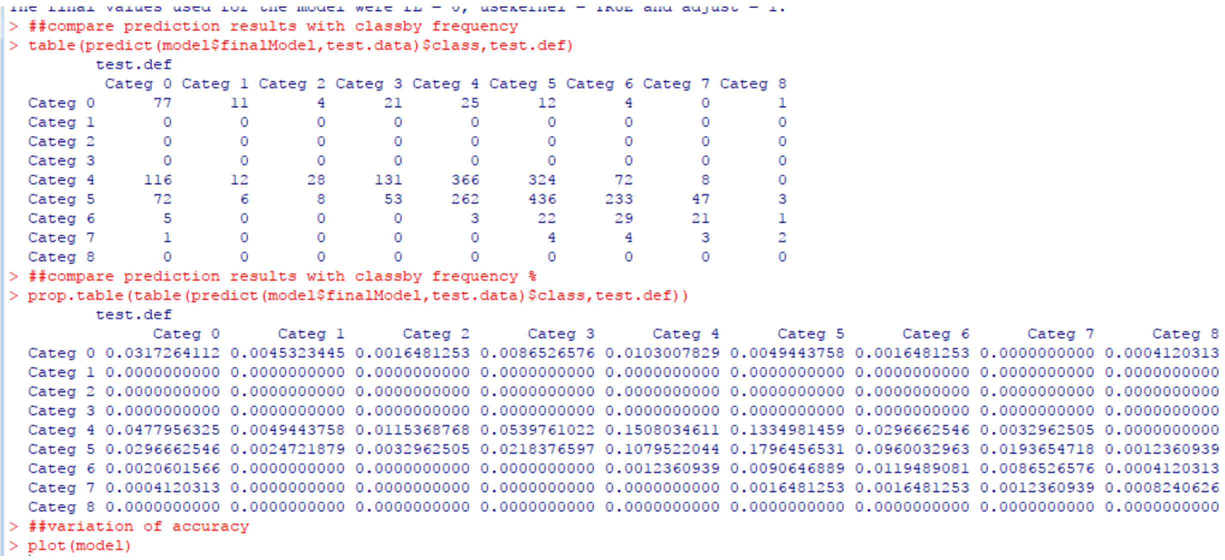
We plot the variation of accuracy with respect to k-value. We found, accuracy is of range 0.32 to 0.39.



Hence, we conclude that we cannot group magnitude category based on length, width and distance of tornado.

Naïve Bayes Classification Model Evaluation (Group Property Loss category in terms of Magnitude category of tornado, region of occurrence and no. of states affected)

We predict on test data set, to evaluate the model.

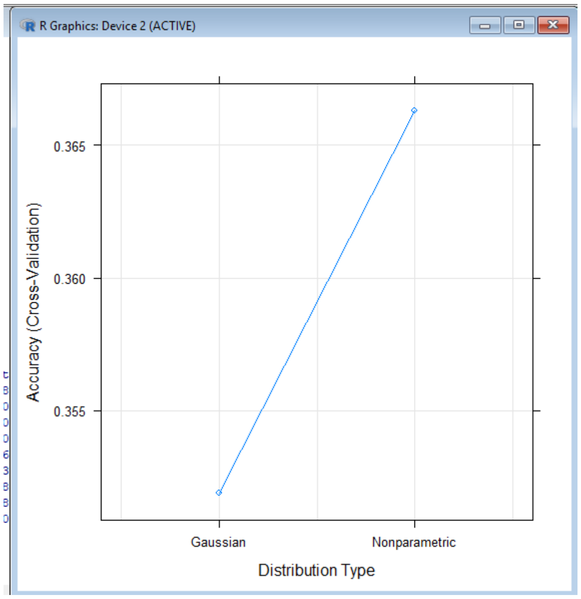


```
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
> ##compare prediction results with classby frequency
> table(predict(model$finalModel,test.data)$class,test.def)
         test.def
          Categ 0 Categ 1 Categ 2 Categ 3 Categ 4 Categ 5 Categ 6 Categ 7 Categ 8
  Categ 0      77      11       4      21      25      12       4       0       1
  Categ 1       0       0       0       0       0       0       0       0       0
  Categ 2       0       0       0       0       0       0       0       0       0
  Categ 3       0       0       0       0       0       0       0       0       0
  Categ 4     116      12      28     131     366     324      72       8       0
  Categ 5      72       6       8      53     262     436     233      47       3
  Categ 6       5       0       0       0       3      22      29      21       1
  Categ 7       1       0       0       0       0       4       4       3       2
  Categ 8       0       0       0       0       0       0       0       0       0
> ##compare prediction results with classby frequency %
> prop.table(table(predict(model$finalModel,test.data)$class,test.def))
         test.def
               Categ 0      Categ 1      Categ 2      Categ 3      Categ 4      Categ 5      Categ 6      Categ 7      Categ 8
  Categ 0 0.0317264112 0.0045323445 0.0016481253 0.0086526576 0.0103007829 0.0049443758 0.0016481253 0.0000000000 0.0004120313
  Categ 1 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
  Categ 2 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
  Categ 3 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
  Categ 4 0.0477956325 0.0049443758 0.0115368768 0.0539761022 0.1508034611 0.1334981459 0.0296662546 0.0032962505 0.0000000000
  Categ 5 0.0296662546 0.0024721879 0.0032962505 0.0218376597 0.1079522044 0.1796456531 0.0960032963 0.0193654718 0.0012360939
  Categ 6 0.0020601566 0.0000000000 0.0000000000 0.0000000000 0.0012360939 0.0090646889 0.0119489081 0.0086526576 0.0004120313
  Categ 7 0.0004120313 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0016481253 0.0016481253 0.0012360939 0.0008240626
  Categ 8 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
> ##variation of accuracy
> plot(model)
```

The first prediction table shows the actual no. of predicted values and the second shows the % of data predicted on various categories.

The prediction results are not good, as for example, for loss category 0, 77 samples were predicted correctly as category 0, but 11 which are actually category 1 were predicted wrongly to be of category 1.

We also evaluated the accuracy variation by model plot.



The accuracy seems to be varying between 0.33 and 0.37. Hence, **we conclude we cannot accurately predict loss category based on magnitude, region and no. of states affected.**

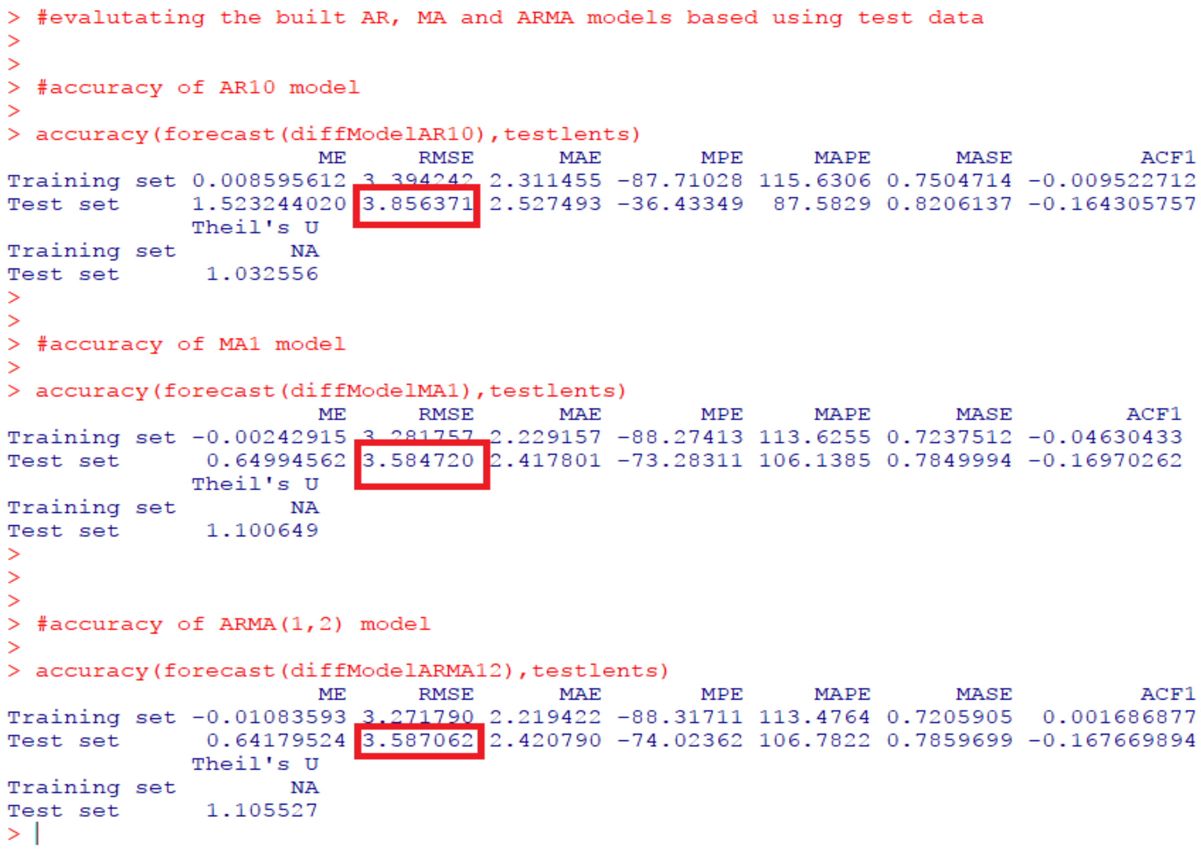
Time series modelling of Tornadoes length data

Evaluation method used for splitting data for time series

We used the Hold-out evaluation method to split the data into training and testing data

Evaluation method applied for choosing the best model

We used the accuracy () function in R to apply the models built using training data on the test data to arrive at the RMSE values of all the models that we built. It is this RMSE value that we use as the criteria to choose the best model as shown below

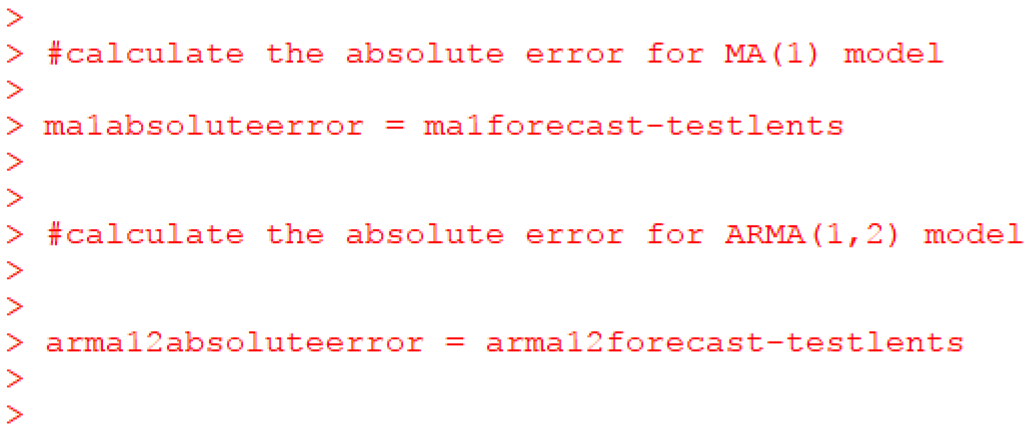


```
> #evalutating the built AR, MA and ARMA models based using test data
>
>
> #accuracy of AR10 model
>
> accuracy(forecast(diffModelAR10),testlents)
                      ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 0.008595612 3.394242 2.311455 -87.71028 115.6306 0.7504714 -0.009522712
Test set     1.523244020 3.856371 2.527493 -36.43349  87.5829 0.8206137 -0.164305757
             Theil's U
Training set        NA
Test set      1.032556
>
>
> #accuracy of MA1 model
>
> accuracy(forecast(diffModelMA1),testlents)
                      ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -0.00242915 3.281757 2.229157 -88.27413 113.6255 0.7237512 -0.04630433
Test set      0.64994562 3.584720 2.417801 -73.28311 106.1385 0.7849994 -0.16970262
             Theil's U
Training set        NA
Test set      1.100649
>
>
>
> #accuracy of ARMA(1,2) model
>
> accuracy(forecast(diffModelARMA12),testlents)
                      ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set -0.01083593 3.271790 2.219422 -88.31711 113.4764 0.7205905  0.001686877
Test set      0.64179524 3.587062 2.420790 -74.02362 106.7822 0.7859699 -0.167669894
             Theil's U
Training set        NA
Test set      1.105527
> |
```

We could see that the RMSE values of MA (1) model and ARMA (1,2) model are the lowest compared to that of AR (10) model

But MA (1) and ARMA (1,2) model have very close RMSE values and hence we have a doubt if both these RMSEs are the same or not. So, to confirm this, we will perform two paired sample two-tailed hypothesis testing on absolute errors and make a conclusion on the best time series model for prediction.

Calculating absolute errors for MA (1) model and ARMA (1,2) models as shown below



```
>
> #calculate the absolute error for MA(1) model
>
> ma1absoluteerror = ma1forecast-testlents
>
>
> #calculate the absolute error for ARMA(1,2) model
>
>
> arma12absoluteerror = arma12forecast-testlents
>
>
```

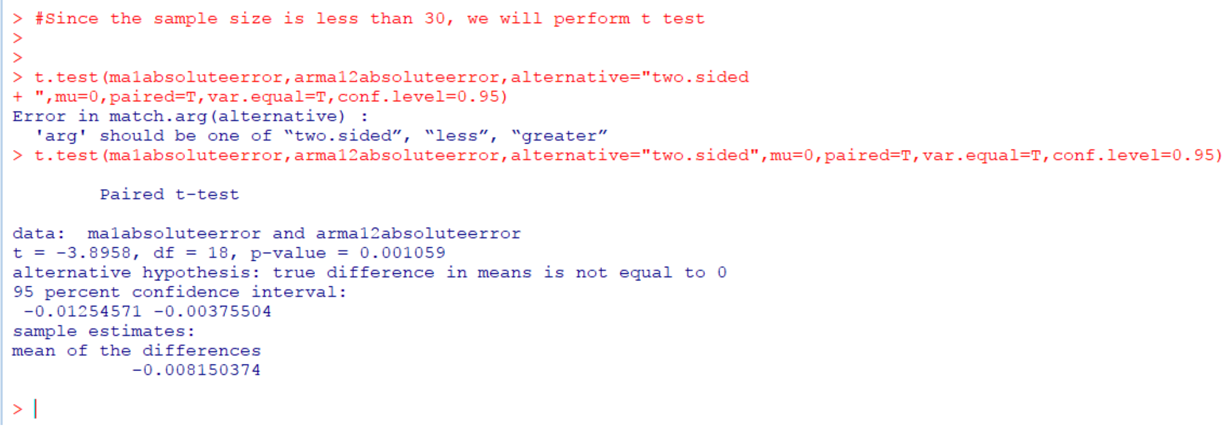
Since both the models are built on the same data and the mean errors are also for the same data, we will carry out two sample paired hypothesis.

Next step is to perform the two paired samples two tailed hypothesis testing. Since the sample size here is less than 30, we perform t-test.

The Hypothesis statements are:

H_0: Errors of models MA (1) and ARMA (1,2) are the same

H_a: Errors of models MA (1) and ARMA (1,2) are not the same



```
> #Since the sample size is less than 30, we will perform t test
>
>
> t.test(ma1absoluteerror,arma12absoluteerror,alternative="two.sided
+ ",mu=0,paired=T,var.equal=T,conf.level=0.95)
Error in match.arg(alternative) :
  'arg' should be one of "two.sided", "less", "greater"
> t.test(ma1absoluteerror,arma12absoluteerror,alternative="two.sided",mu=0,paired=T,var.equal=T,conf.level=0.95)

        Paired t-test

data:  ma1absoluteerror and arma12absoluteerror
t = -3.8958, df = 18, p-value = 0.001059
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01254571 -0.00375504
sample estimates:
mean of the differences
           -0.008150374

> |
```

From the above hypothesis testing screenshot, we see that the p-value is less than 0.05. Hence, we do not have enough evidence to accept the null hypothesis at 95% confidence level. Hence, we can conclude that the errors of models MA (1) and ARMA (1,2) are not the same.

Hence, we confirm that **model MA (1) is the best model for tornado length prediction for different months in future**

6.2. Results and Findings

✓ **By Hypothesis Testing**

- **Average distance travelled by tornadoes of magnitude 3 is more than that of magnitude 4**
- **No of injured people affected by tornadoes in states Alabama and Texas are not the same**
- **Average number of fatalities caused by tornadoes is 4**

- **Number of states affected by tornado of magnitude 0 to 3 is less than 3**

✓ **Comparison of Average loss incurred across different regions by log transformed ANOVA model**

The average loss incurred across various regions are not the same. The difference is significant for SouthEast, SouthWest and West with respect to MidWest, and insignificant for NorthEast with respect to MidWest.

✓ **Comparison of injuries due to tornadoes of different magnitudes by ANOVA model**

The average no. of injuries due to tornadoes of different magnitude categories are not the same. The difference is significant for magnitude categories 3,4 and 5 with respect to scale 0. But, it is insignificant for magnitude categories 1 and 2 with respect to 0.

✓ **Group Magnitude in terms of Length, width and distance travelled by a tornado by KNN classification**

We cannot deduce magnitude category of tornado accurately based on length, width and distance.

✓ **Group Property Loss category in terms of Magnitude category of tornado, region of occurrence and no. of states affected by Naïve Bayes Classification**

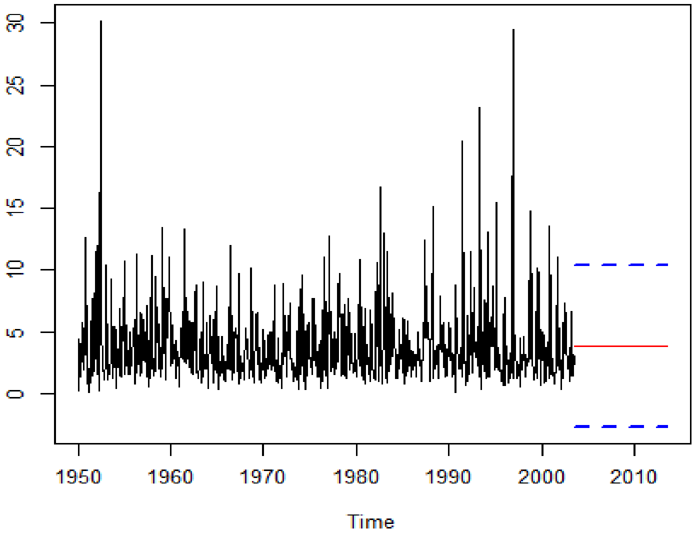
We cannot deduce property loss category accurately based on magnitude, region and no. of states affected.

✓ **Prediction of tornado average length for future years:**

On evaluating the different models based on the RMSE value, we conclude that **the MA (1) model is the best model in predicting the tornado length data for the coming years.**

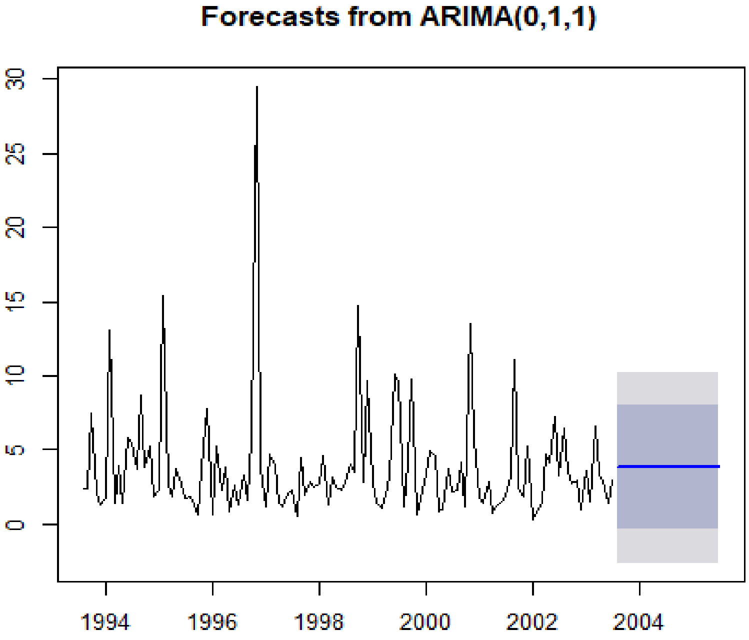
The predictions graph and the forecast graph for the next few years are as given below

Prediction Graph:



The lines in blue represent the prediction interval and the one in red indicates the predicted tornado length for the years from 2004.

Forecast Graph



The line in blue represents the forecasted tornado length for the years after 2004, the dark gray portion represent the 80% confidence interval and the light gray portion represents the 95% confidence interval.

7. Conclusions and Future Work

7.1. Conclusions

We were able to predict, and forecast tornadoes based on the time series model. But, we could not find any effective relationship between the tornado and attributes defining a tornado, that could help predict the magnitude or the impact on people and property.

7.2. Limitations

The data provides a huge amount of information regarding the characteristics of the tornado. But, it has not taken into consideration, the geographic conditions of the region where it occurred. This could have helped identify the factors contributing to the tornado and its intensity.

Also, the data being vast, a deeper knowledge on the various analytics techniques was required, along with better understanding of the domain of the dataset.

7.3. Potential Improvements or Future Work

There is scope for various analytical models on the data set, especially time based. And, with attributes pertaining to the geographic conditions at the region of occurrence during the time could help identify the factors, and hence predict the intensity and extend of loss on people and property.