Spring 4-27-2016

A Comparison of Methods to Fit a Model to Simultaneous Time Series

Victoria Marie Moore

A COMPARISON OF METHODS TO FIT A MODEL TO SIMULTANEOUS TIME SERIES

An honors paper submitted to the Department of Mathematics
of the University of Mary Washington
in partial fulfillment of the requirements for Departmental Honors

Victoria Marie Moore

April 2016

By signing your name below, you affirm that this work is the complete and final version of your paper submitted in partial fulfillment of a degree from the University of Mary Washington. You affirm the University of Mary Washington honor pledge: "I hereby declare upon my word of honor that I have neither given nor received unauthorized help on this work."

Victoria M. Moore
(digital signature)

04/27/16

A Comparison of Methods to Fit a Model to Simultaneous Time Series

University of Mary Washington

Victoria M. Moore

April 2016

Contents

Abstract

This research project determines which methods are the most effective for finding a best fit model for simultaneous time series. The type of model used was an Autoregressive Integrated Moving Average (ARIMA) model. Two distinct methods were used when determining what order to assign to the ARIMA model: 1.) using the floor of the average number of autoregressive and moving average terms, and 2.) using the ceiling of the average number of autoregressive and moving average terms. After fitting the model, the Akaike Information Criterion (AIC) value for each method measured the goodness of fit to compare to fitting separate models to each series. Based on the results of this research the most effective method depends on the type of data that is being fitted. In most of the different cases explored, the floor function method and the ceiling function method had very similar results. However, for two specific cases the ceiling function was the more effective method. Therefore, it is important to consider the characteristics of the data that is being fitted to determine the most effective method.

Introduction

Big data has recently become a higher focus in data analysis due to an increased need and desire to analyze large data sets. In many cases these data sets are a collection of points gathered over time. These time series can be analyzed to help identify significant patterns, to form best fit models, and to make predictions. The ability to simultaneously monitor several time series allows for several different data sets to be compared to each other, such as time series sets for different states to look for patterns across the entire country.

A previous study on time series data was completed by University of Mary Washington students in conjunction with the Dahlgren Navy Base (NSWCDD). The research team analyzed big data from the social media network Twitter. The analysis focused on data pulled from tweets in which users posted about their health. The data that was collected over time was examined for 50 different counties. Autoregressive Integrated Moving Average (ARIMA) models were fit to these time series to determine if there were any events leading to increased health problems in a particular area. They also fit one ARIMA model to all 50 counties combined. When fitting this ARIMA model, the research team used the floor function of the average number of autoregressive (AR) and moving average (MA) terms across the sequence of time series to determine the number of AR and MA terms to include in a single ARIMA model fit to the sequence of simultaneous time series.

This research project explored the different methods of fitting a single ARIMA model to a sequence of simultaneous time series. Specifically, the difference between using a floor function versus a ceiling function was analyzed to determine if one method was more effective than the other. Both methods were compared to fitting separate models to each sequence. Eight different types of time series were used to test the effectiveness of these different methods.

Background

This project experimented with fitting models to time series, which are sequences with a set order. For the purpose of this research the Autoregressive Integrated Moving Average (ARIMA) model was chosen. According to Pennsylvania State University, “ARIMA models, also called Box-Jenkins models, are models that may possibly include autoregressive terms, moving average terms, and differencing operations” (Section 3.1). This type of model is generally denoted as ARIMA(p,d,q), where p represents the autoregressive order, the d represents the differencing, and q represents the moving average order. For example, if p=2 and q=1 the ARIMA model would be

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \beta_1 e_{t-1} + e_t ,$$

where the α parameters represent the AR coefficients and β represents the MA coefficient. Throughout this study differencing will be ignored. In other words the model considered will be ARIMA(p,0,q), which is the same as ARMA(p,q). Once the model is fit to the data, it can be used for explaining the behavior of the time series or for making predictions.

When fitting a model to data, it is important to choose the most effective methods to find the best fit model. A statistic used to measure the effectiveness of a fitted model is called the Akaike Information Criterion (AIC). The equation for AIC is

$$AIC = 2k - 2\ln(L) ,$$

where k represents the number of parameters in the model and L is the maximum value of the likelihood function. The likelihood function is the joint probability distribution of data for a specified set of parameter values. The AIC measures the information lost when replacing data with the model data. When several different models are created using different methods, the AIC values can be compared to determine which model is the best fit. Although other goodness-of-fit statistics are available for comparing models, AIC was used for this study because it is the criterion used by the auto.arima command in R to determine and fit best-fit models.

Methods

In order to experiment with different methods of finding the best fit for a sequence of simultaneous time series data, a computer program was written to randomly generate data. For each method analyzed, there were twenty sets of fifty randomly generated time series data points; the same data was used for analyzing each method. The result is a collection of twenty simultaneous time series, each with fifty observations.

Since this study considers ARMA time series models with no differencing terms, only the autoregressive (AR) and moving average (MA) terms were specified during the data generation. The software used for this study was R. The programs written are included in the Appendix. The command arima.sim was used to generate the time series data. The command auto.arima was then used to fit an ARIMA model to each of the time series. (Because ARMA data was generated, any differencing terms included in a best-fit model were ignored.)

In each of the eight cases examined by these programs, three different AIC values were recorded to compare across the different methods. The first AIC recorded was from the best fit model fit to each of the twenty different time series, which allows there to be different numbers of AR and MA terms for each data set. A best fit model was found for each of the simultaneous series using a program that determined the numbers of AR and MA terms and also estimated the coefficients for these terms. A possibly different best fit model was found for each of the twenty time series. The second and third AIC values were calculated by using the same specified numbers of AR and MA terms for each of the twenty sets of data. In these methods the results from the first method were used to find the average number of AR and MA terms from fitting a separate model to each of the twenty time series. For one method the floor of the average was used to determine the number of terms to estimate; the ceiling of the average was used in the last method. The auto.arima command was then used to fit the same ARMA model to each of the twenty time series using the specified number of coefficients. The exact procedure for one of the eight cases studied is described below. In order to illustrate the process used, Table 1 shows the possible methods for three example series. For Method 1, auto.arima determined the best values for p and q for each series. Consequently, Method 1 uses different models for each time series whereas Methods 2 and 3 use the same model.

Example Series			
Series	**Method 1 Model Fit**	**Method 2 Model Fit**	**Method 3 Model Fit**
1	p=2, q=1	p=1, q=0	p=2, q=1
2	p=1, q=1	p=1, q=0	p=2, q=1
3	p=2, q=0	p=1, q=0	p=2, q=1
	Average p = 1.67, Average q = 0.67		

Table 1: Model fits for example series

The first case examined time series data generated using two fixed autoregressive coefficients, 0.1 and 0.5, and one fixed moving average coefficient, 0.2. Thus the equation is

$$x_t = 0.1x_{t-1} + 0.5x_{t-2} + 0.2e_{t-1} + e_t .$$

Furthermore, a small standard deviation of $\sqrt{0.15}$ was used for the random error term. After the data had been generated, an ARIMA model was fit to the data. In addition to the AIC value, the number of AR and MA terms were extracted from this case and stored in separate matrices for further analysis. The mean, minimum, maximum, and standard deviation of the AIC values were calculated to be used when comparing the efficiency of the different methods. The next method explored in this case was that of taking the floor of the average number of AR and MA terms. These values were used to fit an ARIMA model of the order (floor of the average number of AR, 0, floor of the average number of MA) to the generated data. The AIC values were stored and used to calculate the same statistics as previously mentioned. The last method substituted the ceiling function for the floor function. Thus the generated data was fit to an ARIMA model of the order (ceiling of the average number of AR, 0, ceiling of the average number of MA) and the AIC values were stored.

All subsequent cases in this project followed the same procedures as in the first case. The second case had the same conditions as the first case where time series data was generated using two fixed autoregressive coefficients, 0.1 and 0.5, and one fixed moving average coefficient, 0.2. However, a larger standard deviation of $\sqrt{0.5}$ was used. The third and fourth cases both examined time series data generated using two autoregressive terms and one moving average term, all of which had random coefficients between 0 and 1. The difference between these cases was that the third case used a smaller standard deviation of $\sqrt{0.15}$ and the fourth case used a larger standard deviation of $\sqrt{0.5}$. The fifth and sixth cases were nearly identical to cases three and four, except that the random coefficients were between 0 and 0.5. As before, case five had the lower standard deviation while case six had the higher. Finally, cases seven and eight examined an even mixture of varying numbers of AR and MA terms with random coefficients between 0 and 1. The mixture consisted of four sets of five time series. The four different sets were created as follows: 1.) one AR term and one MA term, 2.) one AR term and two MA terms, 3.) two AR terms and one MA term, and 4.) two AR terms and two MA terms. The four sets were combined to form a set of 20 time series, as there had been in the previous cases. The difference between these cases was that the seventh case used a smaller standard deviation of $\sqrt{0.15}$ and the eighth case used a larger standard deviation of $\sqrt{0.5}$. Table 2 below displays all of the information for the eight cases.

Case	p, q	values	Standard Deviation
1	Fixed: p=2, q=1	Fixed: AR1=0.1, AR2=0.5, MA1=0.2	$\sqrt{0.15}$
2	Same as above	Same as above	$\sqrt{0.5}$
3	Fixed: p=2, q=1	Random between 0 and 1	$\sqrt{0.15}$
4	Same as above	Same as above	$\sqrt{0.5}$
5	Fixed: p=2, q=1	Random between 0 and 0.5	$\sqrt{0.15}$
6	Same as above	Same as above	$\sqrt{0.5}$
7	Even mixture: p (0 to 2) q (0 to 2)	Random between 0 and 1	$\sqrt{0.15}$
8	Same as above	Same as above	$\sqrt{0.5}$

Table 2: Parameters for the eight cases studied

These cases were chosen to determine what aspects of times series data impact the effectiveness of the model fitting methods. This study focuses on the impact of the values of coefficients of an ARMA(2,1) series (fixed, random ranging from 0 to 1, or random ranging from 0 to 0.5) and the amount of random variation of the series.

Results

The following tables display the AIC values for each method for the eight different cases. The values in the tables were used to calculate the average percentage increase in the average

AIC to compare the second and third methods with the first method. That is, the percent increase in average AIC indicates how much worse the approach using the same number of AR and MA terms in models fit to a simultaneous sequence of time series does compared to fitting separate models to each of those time series.

<table>
<tr><th colspan="4">Fixed p=2 and q=1, set coefficients (AR1=0.1, AR2=0.5, MA1=0.2), small SD = $\sqrt{0.15}$</th></tr>
<tr><th></th><th>Best fit models AIC</th><th>arima floor AIC</th><th>arima ceiling AIC</th></tr>
<tr><td><b>Mean</b></td><td>47.17119</td><td>59.24048</td><td>58.9313</td></tr>
<tr><td><b>Min</b></td><td>30.38086</td><td>37.8228</td><td>37.8228</td></tr>
<tr><td><b>Max</b></td><td>61.51179</td><td>87.23085</td><td>87.23085</td></tr>
<tr><td><b>SD</b></td><td>8.727144</td><td>11.02587</td><td>10.83969</td></tr>
</table>

Table 3: Results for Case 1

<table>
<tr><th colspan="4">Fixed p=2 and q=1, set coefficients(AR1=0.1, AR2=0.5, MA1=0.2), large SD = $\sqrt{0.5}$</th></tr>
<tr><th></th><th>Best fit models AIC</th><th>arima floor AIC</th><th>arima ceiling AIC</th></tr>
<tr><td><b>Mean</b></td><td>110.1417</td><td>118.2341</td><td>118.5153</td></tr>
<tr><td><b>Min</b></td><td>93.4908</td><td>98.87386</td><td>98.87386</td></tr>
<tr><td><b>Max</b></td><td>129.493</td><td>135.3692</td><td>135.3692</td></tr>
<tr><td><b>SD</b></td><td>10.12309</td><td>11.05805</td><td>10.85481</td></tr>
</table>

Table 4: Results for Case 2

<table>
<tr><th colspan="4">Fixed p=2 and q=1, random coefficients (1,0,1), small SD = $\sqrt{0.15}$</th></tr>
<tr><th></th><th>Best fit models AIC</th><th>arima floor AIC</th><th>arima ceiling AIC</th></tr>
<tr><td><b>Mean</b></td><td>55.99704</td><td>83.73349</td><td>74.66904</td></tr>
<tr><td><b>Min</b></td><td>51.00926</td><td>79.95844</td><td>56.54016</td></tr>
<tr><td><b>Max</b></td><td>60.98481</td><td>87.50853</td><td>87.50853</td></tr>
<tr><td><b>SD</b></td><td>7.053781</td><td>5.338721</td><td>16.14755</td></tr>
</table>

Table 5: Results for Case 3

<table>
<tr><th colspan="4">Fixed p=2 and q=1, random coefficients (1,0,1),
large SD = $\sqrt{0.5}$</th></tr>
<tr><th></th><th>Best fit models AIC</th><th>arima floor AIC</th><th>arima ceiling AIC</th></tr>
<tr><td><b>Mean</b></td><td>107.3178</td><td>110.6135</td><td>109.9475</td></tr>
<tr><td><b>Min</b></td><td>89.75717</td><td>85.97857</td><td>85.97857</td></tr>
<tr><td><b>Max</b></td><td>121.5015</td><td>123.2994</td><td>123.2994</td></tr>
<tr><td><b>SD</b></td><td>11.97373</td><td>15.23562</td><td>13.72446</td></tr>
</table>

Table 6: Results for Case 4

<table>
<tr><th colspan="4">Fixed p=2 and q=1, random coefficients (1,0,0.5),
small SD = $\sqrt{0.15}$</th></tr>
<tr><th></th><th>Best fit models AIC</th><th>arima floor AIC</th><th>arima ceiling AIC</th></tr>
<tr><td><b>Mean</b></td><td>50.42602</td><td>64.63026</td><td>64.36691</td></tr>
<tr><td><b>Min</b></td><td>30.26033</td><td>42.03672</td><td>42.03672</td></tr>
<tr><td><b>Max</b></td><td>69.0756</td><td>103.345</td><td>103.345</td></tr>
<tr><td><b>SD</b></td><td>9.30843</td><td>18.33312</td><td>17.90962</td></tr>
</table>

Table 7: Results for Case 5

<table>
<tr><th colspan="4">Fixed p=2 and q=1, random coefficients (1,0,0.5),
large SD = $\sqrt{0.5}$</th></tr>
<tr><th></th><th>Best fit models AIC</th><th>arima floor AIC</th><th>arima ceiling AIC</th></tr>
<tr><td><b>Mean</b></td><td>107.4342</td><td>124.3044</td><td>124.0324</td></tr>
<tr><td><b>Min</b></td><td>86.17369</td><td>96.18998</td><td>96.18998</td></tr>
<tr><td><b>Max</b></td><td>129.4788</td><td>170.5216</td><td>170.5216</td></tr>
<tr><td><b>SD</b></td><td>11.09719</td><td>23.51569</td><td>22.95412</td></tr>
</table>

Table 8: Results for Case 6

Even mixture of p and q, random coefficients (1,0,1), small SD = $\sqrt{0.15}$			
	Best fit models AIC	**arima floor AIC**	**arima ceiling AIC**
Mean	54.721	95.04898	90.63482
Min	26.98535	47.3221	42.07901
Max	73.21344	195.5535	195.5535
SD	14.23153	43.53112	44.23242

Table 9: Results for Case 7

Even mixture of p and q, random coefficients (1,0,1), large SD = $\sqrt{0.5}$			
	Best fit models AIC	**arima floor AIC**	**arima ceiling AIC**
Mean	109.239	130.5417	130.5296
Min	85.49435	87.69461	87.69461
Max	128.4546	189.7017	189.7017
SD	13.18627	26.50782	25.37934

Table 10: Results for Case 8

As would be expected, the mean AIC was larger for all three methods for the time series data with more variability; this result can be seen by comparing the results for Case 1 and Case 2, for example. Also, the mean AIC was always lower when fitting separate models for each sequence compared to fitting the same model to each sequence. When comparing the AIC statistics in almost all of the cases, excluding Cases 3 and 7, it appears that there is not much difference in the AIC values between using the floor versus the ceiling function to estimate the number of terms to be used in the ARIMA model. There is a notable difference between the results in Cases 3 and 7, however. In these simulations the AIC statistics from using the ceiling function are closer to the AIC values from fitting separate best fit models. Therefore it appears that in some circumstances it can be more accurate to use the ceiling function instead of the floor function to determine the number of AR and MA terms to fit to a sequence of simultaneous time series. To determine which specific circumstances call for the ceiling function, it is beneficial to consider the average percentage increase from the best fit models. These percentages are shown in Table 11.

Percent increase from the average AIC from the best fit models		
Case	**Floor Method**	**Ceiling Method**
1	25.59%	24.93%
2	7.35%	7.60%
3	49.53%	33.34%
4	3.07%	2.45%
5	28.17%	27.65%
6	15.70%	13.57%
7	73.70%	65.63%
8	19.50%	19.49%

Table 11: Percent increase from the best fit model

The differences between the average AIC values ranged from 0.01% to 16.19%. There were smaller increases in the average AIC in cases where there was more variability (i.e, Cases 2, 4, 6 and 8). In these cases there was little to no difference (less than 3%) in the AIC values between the floor and ceiling methods. On the other hand, when there was less variability such as in cases 3 and 7, there was a larger difference. That is, using the ceiling function to determine the number of AR and MA terms in these cases produced a smaller average increase in average AIC compared to fitting separate models to each time series. Case 5 differs from Case 3 in that a narrower range of possible values was used for the AR and MA coefficients. Therefore, based on the research done in this study, using the ceiling function to determine the number of AR and MA terms is more effective when there is less variability in the time series and when the AR and MA terms have a smaller range of possible values.

Conclusion

The simulations for this study showed that in most of the situations tested both the ceiling and the floor functions were equally efficient in fitting the time series. Case 3 consisted of time series data that was generated using two autoregressive terms and one moving average term, all of which had random coefficients between 0 and 1, and a small standard deviation. Case 7 was made up of time series data created by an even mixture of varying numbers of AR and MA terms with random coefficients between 0 and 1, and a small standard deviation. In both of these cases, using the ceiling of the average number of AR and MA terms was the most effective method. It more accurately fit the data and would be more effective for explaining the behavior of the data. In conclusion, when fitting an ARIMA model to time series data it is important to consider the characteristics of the data and to test the efficiency of different methods in order to choose the best fit model.

Future Studies

This project focused on eight specific cases. Further studies could be conducted to determine if one method is more efficient than the other for different types of data that were not considered here. Six of the eight cases focused on ARIMA sets of order (2,0,1). Therefore, it would be beneficial to explore ARIMA sets of other various orders such as, (2,0,2), (1,0,1), etc. Additionally, the last two cases which examined an even mixture of orders could be altered to be an uneven mixture to analyze the effects of added randomness. Other ranges of possible coefficients could be considered as well as different amounts of variability, including varying the standard deviation for each series. Further aspects that could be changed are the number of simultaneous sequences and the number of observations in each sequence.

Other methods for simultaneously fitting a collection of time series data could be explored. There is a “grid search” method which is similar to the all possible subsets model selection in regression. This method could start with order (1,0,0) fit to all of the time series and tries all possible models up to order $(k_1, 0, k_2)$ for some specified k_1 and k_2. Another method would be to combine all of the sequences into one sequence and find a best fit model. This method can be used if the series have roughly the same average and seasonality. Finally, another method would be vector auto regression for panel data, which is often used in econometrics.

References

Wei, William W. S. Time Series Analysis: Univariate and Multivariate Methods. Redwood City, CA: Addison-Wesley Pub., 1990. Print.

"Welcome to STAT 510!" Welcome to STAT 510! The Pennsylvania State University, 2016. Web. 25 Mar. 2016.

Acknowledgements

I would like to thank Dr. Debra Hydorn for her support and guidance throughout this research project. I would also like to thank Dr. Suzanne Sumner and Dr. Melody Denhere for serving as members on my thesis committee.

Appendix

Program 1

```
#Case 1- fixed p=2 and q=1, sd small

library("forecast")

simulation.matrix<-data.frame()
p<-data.frame()
q<-data.frame()
fit.aic<-data.frame()

for (i in 1:20){
        data<-arima.sim(n=50,list(ar=c(0.1,0.5),ma=.2),sd=sqrt(0.15))
        fit<-auto.arima(data)
        simulation.matrix<-rbind(simulation.matrix,data)

        arma=fit$arma
        p<-rbind(p,arma[1])
        q<-rbind(q,arma[2])
        fit.aic <- rbind(fit.aic,fit$aic)
}

mean(fit.aic[,])
min(fit.aic[,])
max(fit.aic[,])
sd(fit.aic[,])

p.floor<-floor(mean((p[,])))
q.floor<-floor(mean(q[,]))
p.ceiling<-ceiling(mean(p[,]))
q.ceiling<-ceiling(mean(q[,]))


# Method 1 Floor (Case 1)

aic<-data.frame()

for(i in 1:20) {
        data2<-t(simulation.matrix[i,])
        fit2<-arima(data2,order=c(p.floor,0,q.floor))
        aic.value=fit2$aic
        aic<-rbind(aic,aic.value)
}

mean(aic[,])
min(aic[,])
max(aic[,])
sd(aic[,])

#Method 2 Ceiling (Case 1)

aic2<-data.frame()

for(i in 1:20) {
        data3<-t(simulation.matrix[i,])
        fit3<-arima(data3,order=c(p.ceiling,0,q.ceiling))

        aic.value2=fit3$aic
        aic2<-rbind(aic,aic.value2)
}

mean(aic2[,])
min(aic2[,])
max(aic2[,])
sd(aic2[,])
```

Program 2

```
#Case 2- fixed p=2 and q=1, sd large

library("forecast")

simulation.matrixL<-data.frame()
pL<-data.frame()
qL<-data.frame()
fit.aicL<-data.frame()

for (i in 1:20){
        dataL<-arima.sim(n=50,list(ar=c(0.1,0.5),ma=.2),sd=sqrt(0.5))
        fitL<-auto.arima(dataL)
        simulation.matrixL<-rbind(simulation.matrixL,dataL)

        armaL=fitL$arma
        pL<-rbind(pL,armaL[1])
```

```
	qL<-rbind(qL,armaL[2])
	fit.aicL <- rbind(fit.aicL,fitL$aic)
}

mean(fit.aicL[,])
min(fit.aicL[,])
max(fit.aicL[,])
sd(fit.aicL[,])

p.floorL<-floor(mean((pL[,])))
q.floorL<-floor(mean(qL[,]))
p.ceilingL<-ceiling(mean(pL[,]))
q.ceilingL<-ceiling(mean(qL[,]))

# Method 1 Floor (Case 2)

aicL<-data.frame()

for(i in 1:20) {
	data2L<-t(simulation.matrixL[i,])
	fit2L<-arima(data2L,order=c(p.floorL,0,q.floorL))

	aic.valueL=fit2L$aic
	aicL<-rbind(aicL,aic.valueL)
}

mean(aicL[,])
min(aicL[,])
max(aicL[,])
sd(aicL[,])

#Method 2 Ceiling (Case 2)

aic2L<-data.frame()

for(i in 1:20) {
	data3L<-t(simulation.matrixL[i,])
	fit3L<-arima(data3L,order=c(p.ceilingL,0,q.ceilingL))

	aic.value2L=fit3L$aic
	aic2L<-rbind(aicL,aic.value2L)
}

mean(aic2L[,])
min(aic2L[,])
max(aic2L[,])
```

```
sd(aic2L[,])
```

Program 3

```
#Case 3- fixed random p=2 and q=1, sd small, runif(1,0,1)

library("forecast")

simulation.matrixC2<-data.frame()
pC2<-data.frame()
qC2<-data.frame()
fit.aicC2<-data.frame()

for (i in 1:20){
        dataC2<-arima.sim(n=50,list(ar=c(runif(1,0,1),runif(1,0,1)),
                ma=runif(1,0,1)),sd=sqrt(0.15))
        fitC2<-auto.arima(dataC2)
        simulation.matrixC2<-rbind(simulation.matrixC2,dataC2)

        armaC2=fitC2$arma
        pC2<-rbind(pC2,armaC2[1])
        qC2<-rbind(qC2,armaC2[2])
        fit.aicC2 <- rbind(fit.aicC2,fitC2$aic)
}

mean(fit.aicC2[,])
min(fit.aicC2[,])
max(fit.aicC2[,])
sd(fit.aicC2[,])

p.floorC2<-floor(mean((pC2[,])))
q.floorC2<-floor(mean(qC2[,]))
p.ceilingC2<-ceiling(mean(pC2[,]))
q.ceilingC2<-ceiling(mean(qC2[,]))


# Method 1 Floor (Case 3)

aicC2<-data.frame()

for(i in 1:20) {
        data2C2<-t(simulation.matrixC2[i,])
        fit2C2<-arima(data2C2,order=c(p.floorC2,0,q.floorC2))

        aic.valueC2=fit2C2$aic
```

```
    aicC2<-rbind(aicC2,aic.valueC2)
}

mean(aicC2[,])
min(aicC2[,])
max(aicC2[,])
sd(aicC2[,])

#Method 2 Ceiling (Case 3)

aic2C2<-data.frame()

for(i in 1:20) {
    data3C2<-t(simulation.matrixC2[i,])
    fit3C2<-arima(data3C2,order=c(p.ceilingC2,0,q.ceilingC2))

    aic.value2C2=fit3C2$aic
    aic2C2<-rbind(aicC2,aic.value2C2)
}

mean(aic2C2[,])
min(aic2C2[,])
max(aic2C2[,])
sd(aic2C2[,])
```

Program 4

```
#Case 4- fixed random p=2 and q=1, sd large, runif(1,0,1)

library("forecast")

simulation.matrixLC2<-data.frame()
pLC2<-data.frame()
qLC2<-data.frame()
fit.aicLC2<-data.frame()

for (i in 1:20){
    dataLC2<-arima.sim(n=50,list(ar=c(runif(1,0,1),runif(1,0,1)),
        ma=runif(1,0,1)),sd=sqrt(0.5))
    fitLC2<-auto.arima(dataLC2)
    simulation.matrixLC2<-rbind(simulation.matrixLC2,dataLC2)

    armaLC2=fitLC2$arma
    pLC2<-rbind(pLC2,armaLC2[1])
    qLC2<-rbind(qLC2,armaLC2[2])
```

```
	fit.aicLC2 <- rbind(fit.aicLC2,fitLC2$aic)
}

mean(fit.aicLC2[,])
min(fit.aicLC2[,])
max(fit.aicLC2[,])
sd(fit.aicLC2[,])

p.floorLC2<-floor(mean((pLC2[,])))
q.floorLC2<-floor(mean(qLC2[,]))
p.ceilingLC2<-ceiling(mean(pLC2[,]))
q.ceilingLC2<-ceiling(mean(qLC2[,]))

# Method 1 Floor (Case 4)

aicLC2<-data.frame()

for(i in 1:20) {
	data2LC2<-t(simulation.matrixLC2[i,])
	fit2LC2<-arima(data2LC2,order=c(p.floorLC2,0,q.floorLC2))

	aic.valueLC2=fit2LC2$aic
	aicLC2<-rbind(aicLC2,aic.valueLC2)
}

mean(aicLC2[,])
min(aicLC2[,])
max(aicLC2[,])
sd(aicLC2[,])

#Method 2 Ceiling (Case 4)

aic2LC2<-data.frame()

for(i in 1:20) {
	data3LC2<-t(simulation.matrixLC2[i,])
	fit3LC2<-arima(data3LC2,order=c(p.ceilingLC2,0,q.ceilingLC2))

	aic.value2LC2=fit3LC2$aic
	aic2LC2<-rbind(aicLC2,aic.value2LC2)
}

mean(aic2LC2[,])
min(aic2LC2[,])
max(aic2LC2[,])
sd(aic2LC2[,])
```

Program 5

```
#Case 5- fixed random p=2 and q=1, sd small, runif(1,0,0.5)

library("forecast")

simulation.matrixC3<-data.frame()
pC3<-data.frame()
qC3<-data.frame()
fit.aicC3<-data.frame()

for (i in 1:20){
        dataC3<-arima.sim(n=50,list(ar=c(runif(1,0,0.5),runif(1,0,0.5)),
                ma=runif(1,0,0.5)),sd=sqrt(0.15))
        fitC3<-auto.arima(dataC3)
        simulation.matrixC3<-rbind(simulation.matrixC3,dataC3)

        armaC3=fitC3$arma
        pC3<-rbind(pC3,armaC3[1])
        qC3<-rbind(qC3,armaC3[2])
        fit.aicC3 <- rbind(fit.aicC3,fitC3$aic)
}

mean(fit.aicC3[,])
min(fit.aicC3[,])
max(fit.aicC3[,])
sd(fit.aicC3[,])

p.floorC3<-floor(mean((pC3[,])))
q.floorC3<-floor(mean(qC3[,]))
p.ceilingC3<-ceiling(mean(pC3[,]))
q.ceilingC3<-ceiling(mean(qC3[,]))

# Method 1 Floor (Case 5)

aicC3<-data.frame()

for(i in 1:20) {
        data2C3<-t(simulation.matrixC3[i,])
        fit2C3<-arima(data2C3,order=c(p.floorC3,0,q.floorC3))

        aic.valueC3=fit2C3$aic
        aicC3<-rbind(aicC3,aic.valueC3)
}
```

```
mean(aicC3[,])
min(aicC3[,])
max(aicC3[,])
sd(aicC3[,])

#Method 2 Ceiling (Case 5)

aic2C3<-data.frame()

for(i in 1:20) {
        data3C3<-t(simulation.matrixC3[i,])
        fit3C3<-arima(data3C3,order=c(p.ceilingC3,0,q.ceilingC3))

        aic.value2C3=fit3C3$aic
        aic2C3<-rbind(aicC3,aic.value2C3)
}

mean(aic2C3[,])
min(aic2C3[,])
max(aic2C3[,])
sd(aic2C3[,])
```

Program 6

```
#Case 6- fixed random p=2 and q=1, sd large, runif(1,0,0.5)

library("forecast")

simulation.matrixLC3<-data.frame()
pLC3<-data.frame()
qLC3<-data.frame()
fit.aicLC3<-data.frame()

for (i in 1:20){
        dataLC3<-arima.sim(n=50,list(ar=c(runif(1,0,0.5),runif(1,0,0.5)),
                ma=runif(1,0,0.5)),sd=sqrt(0.5))
        fitLC3<-auto.arima(dataLC3)
        simulation.matrixLC3<-rbind(simulation.matrixLC3,dataLC3)

        armaLC3=fitLC3$arma
        pLC3<-rbind(pLC3,armaLC3[1])
        qLC3<-rbind(qLC3,armaLC3[2])
        fit.aicLC3 <- rbind(fit.aicLC3,fitLC3$aic)
}

mean(fit.aicLC3[,])
```

```
min(fit.aicLC3[,])
max(fit.aicLC3[,])
sd(fit.aicLC3[,])

p.floorLC3<-floor(mean((pLC3[,])))
q.floorLC3<-floor(mean(qLC3[,]))
p.ceilingLC3<-ceiling(mean(pLC3[,]))
q.ceilingLC3<-ceiling(mean(qLC3[,]))

# Method 1 Floor (Case 6)

aicLC3<-data.frame()

for(i in 1:20) {
        data2LC3<-t(simulation.matrixLC3[i,])
        fit2LC3<-arima(data2LC3,order=c(p.floorLC3,0,q.floorLC3))

        aic.valueLC3=fit2LC3$aic
        aicLC3<-rbind(aicLC3,aic.valueLC3)
}

mean(aicLC3[,])
min(aicLC3[,])
max(aicLC3[,])
sd(aicLC3[,])

#Method 2 Ceiling (Case 6)

aic2LC3<-data.frame()

for(i in 1:20) {
        data3LC3<-t(simulation.matrixLC3[i,])
        fit3LC3<-arima(data3LC3,order=c(p.ceilingLC3,0,q.ceilingLC3))

        aic.value2LC3=fit3LC3$aic
        aic2LC3<-rbind(aicLC3,aic.value2LC3)
}

mean(aic2LC3[,])
min(aic2LC3[,])
max(aic2LC3[,])
sd(aic2LC3[,])
```

Program 7

```
#Case 7- even mixture of random p and q, sd small
```

```
library("forecast")

simulation.matrixC4<-data.frame()
pC4<-data.frame()
qC4<-data.frame()
fit.aicC4<-data.frame()

for (i in 1:5){
    dataC4<-arima.sim(n=50,list(ar=c(runif(1,0,1)),ma=runif(1,0,1)),sd=sqrt(0.15))
    fitC4<-auto.arima(dataC4)
    simulation.matrixC4<-rbind(simulation.matrixC4,dataC4)

    armaC4=fitC4$arma
    pC4<-rbind(pC4,armaC4[1])
    qC4<-rbind(qC4,armaC4[2])
    fit.aicC4 <- rbind(fit.aicC4,fitC4$aic)
}

for (i in 1:5){
    dataC4<-arima.sim(n=50,list(ar=c(runif(1,0,1),runif(1,0,1)),
        ma=runif(1,0,1)),sd=sqrt(0.15))
    fitC4<-auto.arima(dataC4)
    simulation.matrixC4<-rbind(simulation.matrixC4,dataC4)

    armaC4=fitC4$arma
    pC4<-rbind(pC4,armaC4[1])
    qC4<-rbind(qC4,armaC4[2])
    fit.aicC4 <- rbind(fit.aicC4,fitC4$aic)
}

for (i in 1:5){
    dataC4<-arima.sim(n=50,list(ar=c(runif(1,0,1)),
        ma=c(runif(1,0,1),runif(1,0,1))),sd=sqrt(0.15))
    fitC4<-auto.arima(dataC4)
    simulation.matrixC4<-rbind(simulation.matrixC4,dataC4)

    armaC4=fitC4$arma
    pC4<-rbind(pC4,armaC4[1])
    qC4<-rbind(qC4,armaC4[2])
    fit.aicC4 <- rbind(fit.aicC4,fitC4$aic)
}

for (i in 1:5){
    dataC4<-arima.sim(n=50,list(ar=c(runif(1,0,1),runif(1,0,1)),
        ma=c(runif(1,0,1),runif(1,0,1))),sd=sqrt(0.15))
    fitC4<-auto.arima(dataC4)
```

```
        simulation.matrixC4<-rbind(simulation.matrixC4,dataC4)

        armaC4=fitC4$arma
        pC4<-rbind(pC4,armaC4[1])
        qC4<-rbind(qC4,armaC4[2])
        fit.aicC4 <- rbind(fit.aicC4,fitC4$aic)
}

mean(fit.aicC4[,])
min(fit.aicC4[,])
max(fit.aicC4[,])
sd(fit.aicC4[,])

p.floorC4<-floor(mean((pC4[,])))
q.floorC4<-floor(mean(qC4[,]))
p.ceilingC4<-ceiling(mean(pC4[,]))
q.ceilingC4<-ceiling(mean(qC4[,]))

# Method 1 Floor (Case 7)

aicC4<-data.frame()

for(i in 1:20) {
        data2C4<-t(simulation.matrixC4[i,])
        fit2C4<-arima(data2C4,order=c(p.floorC4,0,q.floorC4))

        aic.valueC4=fit2C4$aic
        aicC4<-rbind(aicC4,aic.valueC4)
}

mean(aicC4[,])
min(aicC4[,])
max(aicC4[,])
sd(aicC4[,])

#Method 2 Ceiling (Case 7)

aic2C4<-data.frame()

for(i in 1:20) {
        data3C4<-t(simulation.matrixC4[i,])
        fit3C4<-arima(data3C4,order=c(p.ceilingC4,0,q.ceilingC4))

        aic.value2C4=fit3C4$aic
        aic2C4<-rbind(aicC4,aic.value2C4)
}
```

```
mean(aic2C4[,])
min(aic2C4[,])
max(aic2C4[,])
sd(aic2C4[,])
```

Program 8

```
#Case 8- even mixture of random p and q, sd large

library("forecast")

simulation.matrixLC4<-data.frame()
pLC4<-data.frame()
qLC4<-data.frame()
fit.aicLC4<-data.frame()

for (i in 1:5){
        dataLC4<-arima.sim(n=50,list(ar=c(runif(1,0,1)),ma=runif(1,0,1)),sd=sqrt(0.5))
        fitLC4<-auto.arima(dataLC4)
        simulation.matrixLC4<-rbind(simulation.matrixLC4,dataLC4)

        armaLC4=fitLC4$arma
        pLC4<-rbind(pLC4,armaLC4[1])
        qLC4<-rbind(qLC4,armaLC4[2])
        fit.aicLC4 <- rbind(fit.aicLC4,fitLC4$aic)
}

for (i in 1:5){
        dataLC4<-arima.sim(n=50,list(ar=c(runif(1,0,1),runif(1,0,1)),
                ma=runif(1,0,1)),sd=sqrt(0.5))
        fitLC4<-auto.arima(dataLC4)
        simulation.matrixLC4<-rbind(simulation.matrixLC4,dataLC4)

        armaLC4=fitLC4$arma
        pLC4<-rbind(pLC4,armaLC4[1])
        qLC4<-rbind(qLC4,armaLC4[2])
        fit.aicLC4 <- rbind(fit.aicLC4,fitLC4$aic)
}

for (i in 1:5){
        dataLC4<-arima.sim(n=50,list(ar=c(runif(1,0,1)),
                ma=c(runif(1,0,1),runif(1,0,1))),sd=sqrt(0.5))
        fitLC4<-auto.arima(dataLC4)
        simulation.matrixLC4<-rbind(simulation.matrixLC4,dataLC4)
        armaLC4=fitLC4$arma
        pLC4<-rbind(pLC4,armaLC4[1])
        qLC4<-rbind(qLC4,armaLC4[2])
        fit.aicLC4 <- rbind(fit.aicLC4,fitLC4$aic)
}

for (i in 1:5){
        dataLC4<-arima.sim(n=50,list(ar=c(runif(1,0,1),runif(1,0,1)),
                ma=c(runif(1,0,1),runif(1,0,1))),sd=sqrt(0.5))
        fitLC4<-auto.arima(dataLC4)
        simulation.matrixLC4<-rbind(simulation.matrixLC4,dataLC4)

        armaLC4=fitLC4$arma
        pLC4<-rbind(pLC4,armaLC4[1])
        qLC4<-rbind(qLC4,armaLC4[2])
        fit.aicLC4 <- rbind(fit.aicLC4,fitLC4$aic)
}

mean(fit.aicLC4[,])
min(fit.aicLC4[,])
max(fit.aicLC4[,])
sd(fit.aicLC4[,])

p.floorLC4<-floor(mean((pLC4[,])))
q.floorLC4<-floor(mean(qLC4[,]))
p.ceilingLC4<-ceiling(mean(pLC4[,]))
q.ceilingLC4<-ceiling(mean(qLC4[,]))

# Method 1 Floor (Case 8)

aicLC4<-data.frame()

for(i in 1:20) {
        data2LC4<-t(simulation.matrixLC4[i,])
        fit2LC4<-arima(data2LC4,order=c(p.floorLC4,0,q.floorLC4))

        aic.valueLC4=fit2LC4$aic
        aicLC4<-rbind(aicLC4,aic.valueLC4)
}

mean(aicLC4[,])
min(aicLC4[,])
max(aicLC4[,])
sd(aicLC4[,])

#Method 2 Ceiling (Case 8)
```

```
aic2LC4<-data.frame()

for(i in 1:20) {
	data3LC4<-t(simulation.matrixLC4[i,])
	fit3LC4<-arima(data3LC4,order=c(p.ceilingLC4,0,q.ceilingLC4))

	aic.value2LC4=fit3LC4$aic
	aic2LC4<-rbind(aicLC4,aic.value2LC4)
}

mean(aic2LC4[,])
min(aic2LC4[,])
max(aic2LC4[,])
sd(aic2LC4[,])
```