International Journal of Current Research and Review
DOI: 10.7324/IJCRR.2017.9244

Research Article

IJCRR
Section: General Science
Sci. Journal Impact Factor 4.016
ICV: 71.54

Forecasting of Monthly Mean Rainfall in Rayalaseema

J. C. Ramesh Reddy[1], T. Ganesh[2], M. Venkateswaran[3], PRS Reddy[1]

[1]Department of Statistics, Sri Venkateswara University, Tirupati, India; [2]Department of Mathematics, NIT Hamirpur, Himachal Pradesh, India; [3]Department of H&S, S V Engineering College, Tirupati, India.

ABSTRACT

Seasonal Autoregressive Integrated Moving Average (SARIMA) model in analyzing the forecast of monthly mean rainfall in Rayalaseema (India) using R is discussed. We have checked the different ARIMA models according to the structure of the data, finally we found that the ARIMA (5,0,1)(2,0,0)[12] has been fitted to the data and the significance test has been made by using lowest AIC and BIC values.

Key Words: Box-Jenkins Methodology, SARIMA model, AIC & BIC

INTRODUCTION

Andhra Pradesh is one of the states of Indian, in which we have taken the rainfall data of Rayalaseema which statistical reports called the drought area in the state. In this blog, we have done the analysis like forecasting of annual rainfall in Rayalaseema for coming years. For the experiment, we have taken data of Mean Annual Rainfall from www.data.gov.in. The data is having the information of mean annual rainfall from year 1901 to 2016.

In this experiment we have taken the help of R programming that is now one of most useful software in the field of data science and statistics. For the analysis, first column of the dataset is chosen to do analysis that is having annual mean rainfall information in mm unit.

METHODOLOGY

ARIMA models are capable of modeling a wide range of seasonal data. A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

ARIMA (p, d, q) $(P, D, Q)_m$: the first parenthesis represents the non-seasonal part of the model and second represents the seasonal part of the model, where m= number of periods per season. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model. The additional seasonal terms are simply multiplied with the non-seasonal terms.

ACF/PACF

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, an $ARIMA(0,0,0)(0,0,1)_{12}$ model will show: a spike at lag 12 in the ACF but no other significant spikes. The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36,….etc. Similarly, an $ARIMA(0,0,0)(1,0,0)_{12}$ model will show: exponential decay in the seasonal lags of the ACF a single significant spike at lag 12 in the PACF. In considering the appropriate seasonal orders for an ARIMA model, restrict attention to the seasonal lags. The modeling procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components of the model.

Seasonal autoregressive integrated moving average (SARIMA) model for any variable involves mainly four steps: Identification, Estimation, Diagnostic checking and Forecasting. The basic form of SARIMA model is denoted by $SARIMA(p,d,q)X(P,D,Q)_S$ and the model is given by

Corresponding Author:
J. C. Ramesh Reddy, Department of Statistics, Sri Venkateswara University, Tirupati, India.
E-mail: rameshreddy_insurance@yahoo.co.in

ISSN: 2231-2196 (Print) **ISSN:** 0975-5241 (Online)
Received: 12.10.2017 **Revised:** 09.11.2017 **Accepted:** 07.12.2017

$\phi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D Z_t = \theta_q(B)\Theta_Q(B^s)a_t$, where Z_t is the time series value at time t and ϕ, Φ, θ *and* Θ are polynomials of order of p, P, q and Q respectively. B is the backward shift operator, $B^s Z_t = Z_{t-s}$ and $\nabla = (1-B)$. Order of seasonality is represented by s. Non-seasonal and seasonal difference orders are denoted by d and D respectively. White noise process is denoted by a_t. The Box-Jenkins methodology involves four steps (Box et al., 1994): (i) identification (ii) estimation (iii) diagnostic checking and (iv) forecasting. First, the original series must be transformed to become stationary around its mean and its variance. Second, the appropriate order of p and q must be specified using autocorrelation and partial autocorrelation functions. Third, the value of the parameters must be estimated using some non-linear optimization procedure that minimizes the sum of squares of the errors or some other appropriate loss function. Diagnostic checking of the model adequacy is required in the fourth step. This procedure is continued until an adequate model is obtained. Finally, the future forecasts generate using minimum mean square error method (Box et al. 1994). SARIMA models are used as benchmark models to compare the performance of the other models developed on the same data set. The iterative procedure of SARIMA model building was explained by Kumari et al. (2013), Boiroju (2012), Rao (2011) and Box et al. (1994).

ARIMA ()

By default, the `arima()` command in R sets c=μ=0 when d>0 and provides an estimate of μ when d=0. The parameter μ is called the "intercept" in the R output. It will be close to the sample mean of the time series, but usually not identical to it as the sample mean is not the maximum likelihood estimate when p+q>0. The `arima()` command has an argument `include. mean` which only has an effect when d=0 and is `TRUE` by default. Setting `include. mean=FALSE` will force μ=0.

The `Arima()` command from the forecast package provides more flexibility on the inclusion of a constant. It has an argument `include.mean` which has identical functionality to the corresponding argument for `arima()`. It also has an argument `include.drift` which allows μ≠0μ≠0 when d=1. For the d>1, no constant is allowed as a quadratic or higher order trend is particularly dangerous when forecasting. The parameter μ is called the "drift" in the R output when d=1.

This is also an argument include.constant which, if TRUE will see include.mean=TRUE if d=0 and include.drift=TRUE when d=1. If include.constant=FALSE. Both include.mean and include.drift will be set to FALSE. If include.constant is used, the values of include.mean=TRUE and include.drift=TRUE are ignored.

AUTO.ARIMA ()

The auto.arima() function automates the inclusion of a constant. By default, for d=0 or d=1, a constant will be included if it improves the AIC value; for d>1, the constant is always omitted. If allow drift=FALSE is specified, then the constant is only allowed when d=0.

There is another function arima() in R which also fits an ARIMA model. However, it does not allow for the constant cc unless d=0, and it does not return everything required for the forecast() function. Finally, it does not allow the estimated model to be applied to new data (which is useful for checking forecast accuracy). Consequently, it is recommended that you use Arima() instead.

Modeling Procedure

When fitting an ARIMA model to a set of time series data, the following procedure provides a useful general approach.

1. Plot the data. Identify any unusual observations.
2. If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
3. If the data are non-stationary: take first differences of the data until the data are stationary.
4. Examine the ACF/PACF: Is an AR(pp) or MA(qq) model appropriate?
5. Try your chosen model(s), and use the AICc to search for a better model.
6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
7. Once the residuals look like white noise, calculate forecasts.

AIC and BIC are both penalized-likelihood criteria. They are sometimes used for choosing best predictor subsets in regression and often used for comparing non-nested models, which ordinary statistical tests cannot do. The AIC or BIC for a model is usually written in the form [-2log$L$ + kp], where L is the likelihood function, p is the number of parameters in the model, and k is 2 for AIC and log(n) for BIC.

AIC is an estimate of a constant plus the relative distance between the unknown true likelihood function of the data and the fitted likelihood function of the model, so that a lower AIC means a model is considered to be closer to the truth. BIC is an estimate of a function of the posterior probability of a model being true, under a certain Bayesian setup, so that a lower BIC means that a model is considered to be more likely to be the true model. Both criteria are based on various assumptions and asymptotic approximations. Each, despite its heuristic usefulness, has therefore been criticized as having questionable validity for real world data. But despite various subtle theoretical differences, their only difference in practice is the size of the penalty; BIC penalizes model complexity more heavily. The only way they should disagree is when AIC chooses a larger model than BIC.

AIC and BIC are both approximately correct according to a different goal and a different set of asymptotic assumptions. Both sets of assumptions have been criticized as unrealistic. Understanding the difference in their practical behavior is easiest if we consider the simple case of comparing two nested models. In such a case, several authors have pointed out that IC's become equivalent to likelihood ratio tests with different alpha levels. Checking a chi-squared table, we see that AIC becomes like a significance test at alpha=.16, and BIC becomes like a significance test with alpha depending on sample size, e.g., .13 for n = 10, .032 for n = 100, .0086 for n = 1000, .0024 for n = 10000. Remember that power for any given alpha is increasing in n. Thus, AIC always has a chance of choosing too big a model, regardless of n. BIC has very little chance of choosing too big a model if n is sufficient, but it has a larger chance than AIC, for any given n, of choosing too small a model.

So what's the bottom line? In general, it might be best to use AIC and BIC together in model selection. For example, in selecting the number of latent classes in a model, if BIC points to a three-class model and AIC points to a five-class model, it makes sense to select from models with 3, 4 and 5 latent classes. AIC is better in situations when a false negative finding would be considered more misleading than a false positive, and BIC is better in situations where a false positive is as misleading as, or more misleading than, a false negative.

Forecasting of rainfall in Rayalaseema

The data has been taken to predict the rain fall of Rayalaseema area of Andhra Pradesh from the year 1951 to 2016. The data has monthly rainfall for each year. In this section, we have to check forecasting model to this data using one of statistical tool R software. In R software majorly we need packages for forecasting model. Using these packages is predicting the model for the Rayalaseema data. The packages are '**ggplot2**', '**forecast**' and '**tseries**'. Install the above mentioned packages using *install.packages()* function and call that packages using library function as below:

Library ('ggplot2') # calls the packages using Library ().

Library ('forecast')

Library ('tseries')

The data has been converted into ".*csv*" file and then read the data into the R programming.

CA<-read.csv ("D:/ganesh/statistics Softwares/project/Analysis2017/Ganesh/CA.csv") # to read the data

To find the summery of the rainfall of Rayalaseema, the following command can be use:

Summary (CA) # to see basic details of data set

Summary details:

Rayalaseema	Min	1st Quartile	Median	Mean	3rd Quartile	Max
	0.00	8.17	43.20	57.83	88.45	327.60

The minimum and maximum rainfall is 0.00 and 327.60, first and third quartiles are 8.17 and 88.45. Based on the quartile, the deviation is 40.14.Average rainfall of Rayalaseema is 57.83and median rainfall is 43.20 Depending on the above summery, we cannot give any decision about the rainfall of Rayalaseema.

To know the pattern of the data, we have applied the below mentioned time series command.

Myts<- ts (CA [, 3], start=c (1951, 1), end = c (2016, 12), frequency = 12) # Applying time series to data

If you draw the graph of the rainfall, we can observe whether the data is in stationary or not. To check the stationary of the data we have applied Augmented Dickey-Fuller test.The test is significant ($p<0.05$*), so that the data is stationary and we can observe in the graph of the data and its difference.

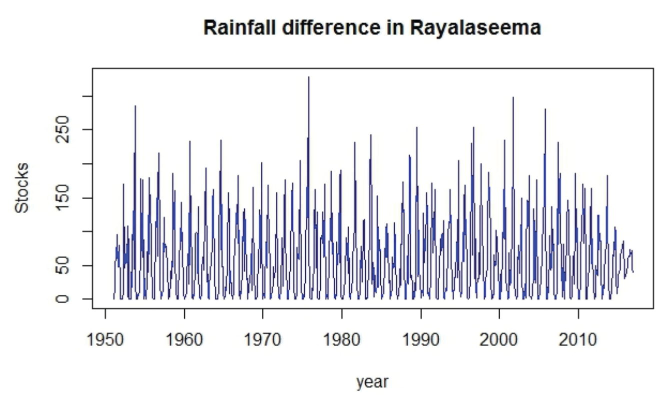
Plot (Myts, xlab='year', ylab = 'Stocks', main="Rainfall difference in Rayalaseema Area", col="blue") # Graph for time series data

adf.test (myts, alternative = "stationary") # Stationary checking

Augmented Dickey-Fuller Test

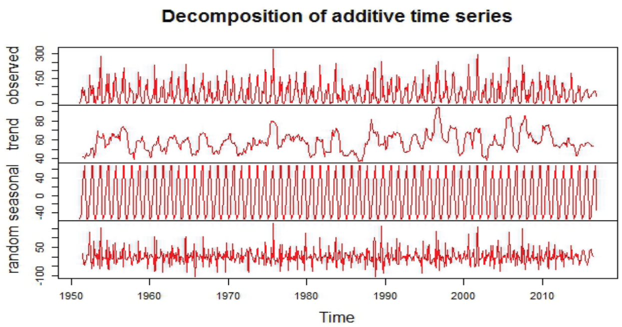
Dickey-Fuller = -13.41, Lag order = 9, p-value = 0.01

Alternative hypothesis: stationary



Dec <- decompose (Myts) # using decompose function to see the decompose details.

Plot (Dec, col="red") # plot of decompose values



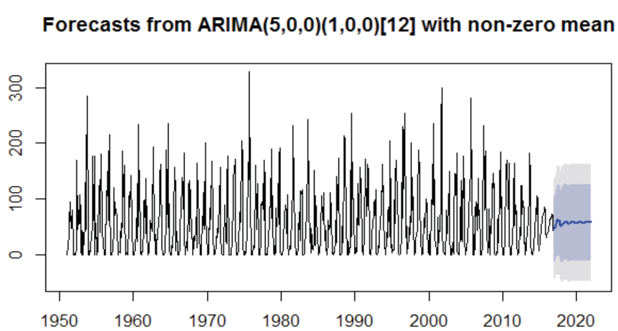
The four graphs are the original data, seasonal component, trend component and the remainder and this shows the periodic seasonal pattern extracted out from the original data and the trend. There is a bar at the right hand side of each graph to allow a relative comparison of the magnitudes of each component. For this data the change in trend is less than the variation doing to the monthly variation.

ARIMA model for data

We have considered the different models to check the least AIC and BIC values of the model so that, the model will be the fitted one. Basing on the different models we made a comparison with AIC and BIC values. And finally we applied auto ARIMA to find the best fitted model to the data.

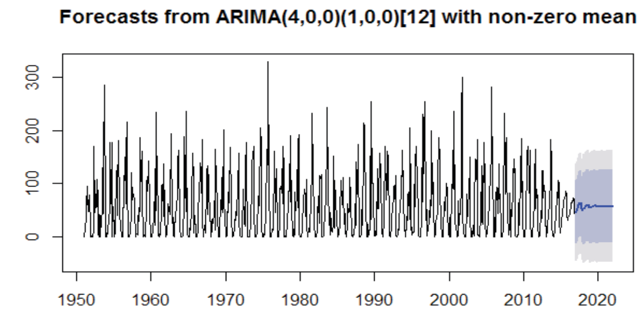
ARIMA (5,0,0)(1,0,0)[12] with non-zero mean

Coefficients:	**AR (1)**	**AR (2)**	**AR (3)**	**AR (4)**	**AR (5)**	**SAR (1)**	**Intercept**
Estimate	0.2102	0.1528	-0.0311	-0.1143	-0.2313	0.3027	57.7582
SE	0.0377	0.0357	0.0355	0.0364	0.0365	0.0435	2.2698
sigma^2 estimated as 2060			Log likelihood=-4146.29				
AIC=8308.59			BIC=8345.98				



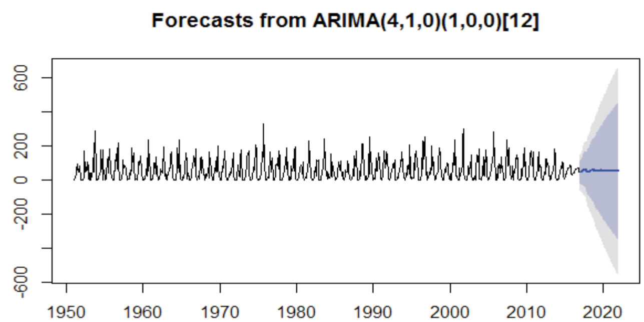
ARIMA (4,0,0)(1,0,0)[12] with non-zero mean

Coefficients:	AR (1)	AR (2)	AR (3)	AR (4)	SAR (1)	Intercept
Estimate	0.1909	0.1444	-0.0557	-0.1235	0.4178	57.6438
SE	0.0428	0.0374	0.0363	0.0404	0.0454	3.3252
sigma^2 estimated as 2158		Log likelihood=-4165.08				
AIC=8344.10		BIC=8376.88				



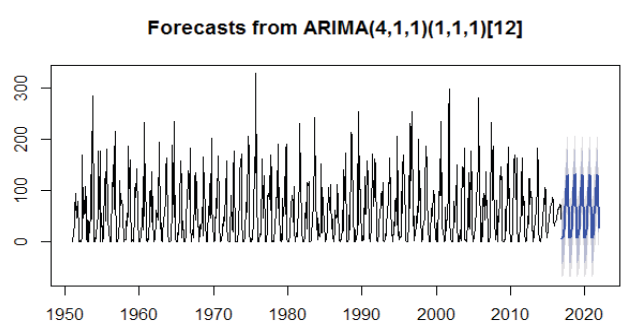
ARIMA (4, 1, 0) (1, 0, 0) [12] with non-zero mean

Coefficients:	**AR (1)**	**AR (2)**	**AR (3)**	**AR (4)**	**SAR (1)**
Estimate	-0.6452	-0.3461	-0.2183	-0.0844	0.5338
SE	0.0427	0.0508	0.0465	0.0362	0.0380
sigma^2 estimated as 2717			Log likelihood=-4251.93		
AIC=8515.86			BIC=8543.909		



ARIMA (4, 1, 1) (1, 1, 1) [12] with non-zero mean

Coefficients:	**AR (1)**	**AR (2)**	**AR (3)**	**AR (4)**	**MA (1)**	**SAR (1)**	**SMA (1)**
Estimate	-0.0552	0.0470	0.0015	-0.0052	-1.00	0.0219	-1.00
SE	0.0361	0.0359	0.0359	0.0360	0.0082	0.0366	0.0179
sigma^2 estimated as 1463					Log likelihood=-3975.38		
AIC=7966.76					BIC=8004.155		



Model Comparison

Model	**AIC**	**BIC**
ARIMA (5,0,0)(1,0,0)[12]	8308.59	8345.98

ARIMA (4,0,0)(1,0,0)[12]	8344.10	8376.88
ARIMA (4,1,0)(1,0,)[12]	8515.86	8543.909
ARIMA (4,1,1)(1,1,1)[12]	7966.76	8004.155

From the above table, we can observe clearly, the model ARIMA (4,1,1)(1,1,1)[12] has the least AIC and BIC values, it may be considered as the best fitted model, but the seasonal differences have been taken is very high so that it has the least values of AIC and BIC.

ARIMAfit<- auto.arima (Myts, approximation=FALSE, trace=FALSE) # to build forecasting model

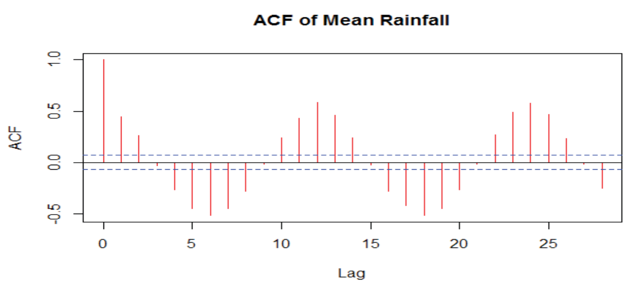
The ACF plot of the residuals from the ARIMA (5,0,1)(2,0,0)[12] model shows all correlations within the threshold limits indicating that the residuals are behaving like white noise. A portmanteau test returns a large p-value, also suggesting the residuals are white noise. The PACF shown is suggestive of model. So an initial candidate model is an ARIMA (5,0,1)(2,0,0)[12]. There are no other obvious candidate models.

ARIMA (5,0,1)(2,0,0)[12] with non-zero mean

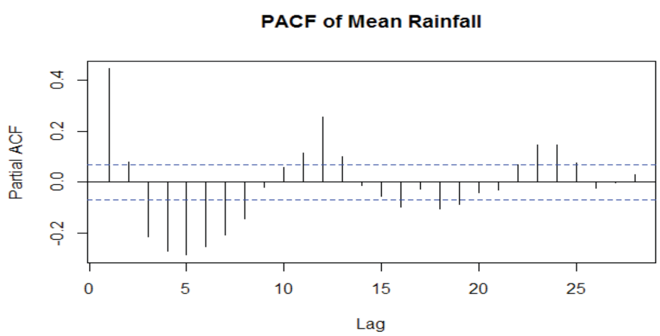
Coefficients:	**AR (1)**	**AR (2)**	**AR (3)**	**AR (4)**	**AR (5)**	**MA(1)**	**SAR (1)**	**SAR (2)**	**Intercept**
Estimate	0.7234	0.0330	-0.1027	-0.0740	-0.1413	-0.6341	0.2233	0.2073	57.6463
SE	0.0681	0.0448	0.0450	0.0442	0.0433	0.0611	0.0438	0.0438	1.7368
sigma^2 estimated as 1893				Log likelihood=-4146.29					
AIC=8237.93			AIC=8238.21			BIC=8284.68			

To check the details of ACF and PCF of Rain fall data.

Acf (ts (Myts), main='ACF of Mean Rainfall', col="red")



Pacf (ts (Myts), main='PACF of Mean Rainfall', col="green")



The ACF plot of the residuals from the ARIMA (5,0,1)(2,0,0)[12] model shows all correlations within the threshold limits indicating that the residuals are behaving like white noise. A portmanteau test returns a large p-value, also suggesting the residuals are white noise. The PACF shown is suggestive of model. So an initial candidate model is an ARIMA (5,0,1)(2,0,0)[12]. There are no other obvious candidate models.

Automate ARIMA model for the data is ARIMA (5,0,1)(2,0,0)[12] with non-zero mean. The predicted values for coast area rain fall details using ARIMA method of (5, 0, 1) and (2, 0, 0).

By applying auto ARIMA, we got the best fitted model which has the lowest AICc. When models are compared using AICc values, it is important that all models have the same orders of differencing. However, when comparing models using a test set, it does not matter how the forecasts were produced - the comparisons are always valid. The given model has been passed the residual tests. In practice, we would normally use the best model we could find, even if it did not pass all tests. Forecasts from the ARIMA$(5,0,1)(2,0,0)_{12}$ model are shown in the figure below.

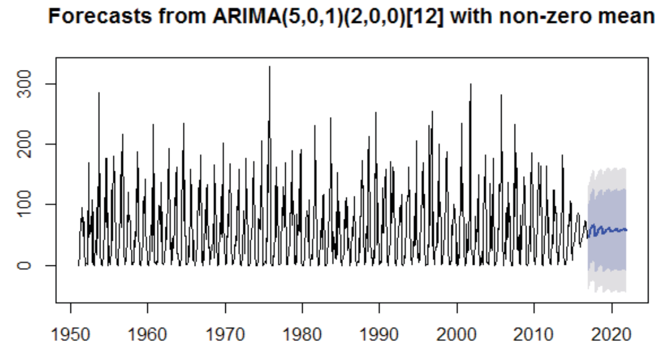
Fact <- forecast (ARIMAfit, h=60) #forecasting values

Print(Fact) # to print the forecast values

Point	**Forecast**	**Lo 80**	**Hi 80**	**Lo 95**	**Hi 95**
Jan-2017	45.33226	-10.4307277	101.0953	-39.94987	130.6144
Feb-2017	48.1612	-7.8237748	104.1462	-37.46043	133.7828
Mar-2017	50.29227	-5.956505	106.541	-35.73281	136.3173
Apr-2017	52.84735	-3.4249532	109.1196	-33.21371	138.9084
May-2017	59.44104	2.8870376	115.995	-27.05084	145.9329
Jun-2017	62.01219	3.9964491	120.0279	-26.71523	150.7396
Jul-2017	62.08568	3.1296584	121.0417	-28.07977	152.2511
Aug-2017	65.75252	6.2449894	125.2601	-25.2564	156.7614
Sep-2017	66.08503	6.4282978	125.7418	-25.15207	157.3221
Oct-2017	65.85715	6.1984433	125.5159	-25.38297	157.0973
Nov-2017	51.4987	-8.2326991	111.2301	-39.85259	142.85
Dec-2017	48.10909	-11.8014097	108.0196	-43.51611	139.7343
Jan-2018	50.1024	-12.2841868	112.489	-45.30965	145.5145
Feb-2018	51.48627	-11.1145982	114.0871	-44.25349	147.226
Mar-2018	53.06941	-9.6402545	115.7791	-42.83675	148.9756
Apr-2018	54.13202	-8.5783353	116.8424	-41.77519	150.0392
May-2018	58.44933	-4.2974872	121.1962	-37.51365	154.4123

Jun-2018	60.00279	-2.9242789	122.9299	-36.23586	156.2414
Jul-2018	59.95357	-3.1376914	123.0448	-36.53619	156.4433
Aug-2018	62.78722	-0.4166723	125.9911	-33.87479	159.4492
Sep-2018	61.66552	-1.5792236	124.9103	-35.05897	158.39
Oct-2018	62.46053	-0.7875231	125.7086	-34.26902	159.1901
Nov-2018	54.05634	-9.1974776	117.3102	-42.68203	150.7947
Dec-2018	51.95587	-11.3222068	115.234	-44.8196	148.7313
Jan-2019	53.37781	-11.9920234	118.7476	-46.59672	153.3523
Feb-2019	54.27363	-11.1702762	119.7175	-45.81419	154.3614
Mar-2019	55.07619	-10.4209702	120.5734	-45.09307	155.2455
Apr-2019	55.85413	-9.6430781	121.3513	-44.31521	156.0235
May-2019	58.19644	-7.3212648	123.7141	-42.00424	158.3971
Jun-2019	59.08536	-6.5545169	124.7252	-41.30217	159.4729
Jul-2019	59.09479	-6.6433739	124.833	-41.44306	159.6326
Aug-2019	60.48876	-5.3152116	126.2927	-40.14973	161.1273
Sep-2019	60.30485	-5.521743	126.1314	-40.36823	160.9779
Oct-2019	60.43063	-5.3974835	126.2587	-40.24478	161.106
Nov-2019	55.57223	-10.2601893	121.4047	-45.10977	156.2542
Dec-2019	54.3961	-11.4519109	120.2441	-46.30974	155.1019
Jan-2020	55.12399	-11.1423872	121.3904	-46.22169	156.4697
Feb-2020	55.6099	-10.6853997	121.9052	-45.78001	156.9998
Mar-2020	56.11794	-10.1948192	122.4307	-45.29867	157.5346
Apr-2020	56.51371	-9.7993254	122.8267	-44.90332	157.9307
May-2020	57.934	-8.3834653	124.2515	-43.48981	159.3578
Jun-2020	58.45663	-7.8880614	124.8013	-43.00882	159.9221
Jul-2020	58.45003	-7.9193194	124.8194	-43.05313	159.9532
Aug-2020	59.34946	-7.0373351	125.7363	-42.18038	160.8793
Sep-2020	59.07578	-7.3176021	125.4692	-42.46414	160.6157
Oct-2020	59.26803	-7.1259903	125.6621	-42.27286	160.8089
Nov-2020	56.43993	-9.9548542	122.8347	-45.10213	157.982
Dec-2020	55.74088	-10.6574239	122.1392	-45.80656	157.2883
Jan-2021	56.19744	-10.3952197	122.7901	-45.64725	158.0421
Feb-2021	56.49123	-10.1108071	123.0933	-45.3678	158.3503
Mar-2021	56.77099	-9.8374551	123.3794	-45.09783	158.6398
Apr-2021	57.02086	-9.5876444	123.6294	-44.84805	158.8898
May-2021	57.82398	-8.78635	124.4343	-44.04773	159.6957
Jun-2021	58.12541	-8.496669	124.7475	-43.76427	160.0151
Jul-2021	58.12626	-8.5058527	124.7584	-43.77876	160.0313
Aug-2021	58.61632	-8.0227644	125.2554	-43.29937	160.532
Sep-2021	58.51716	-8.124473	125.1588	-43.40242	160.4368
Oct-2021	58.58611	-8.055751	125.228	-43.33382	160.506
Nov-2021	56.94724	-9.6949745	123.5895	-44.97323	158.8677
Dec-2021	56.54713	-10.0965836	123.1908	-45.37563	158.4699

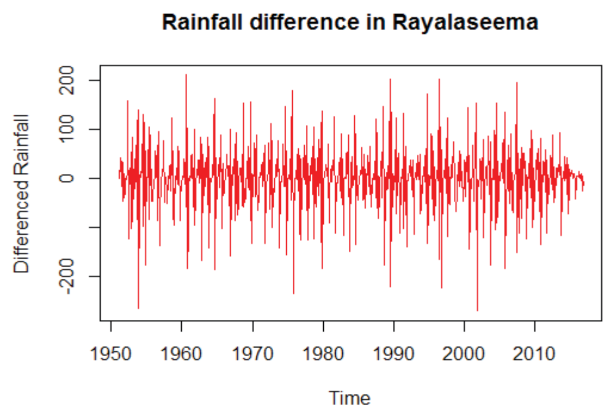
plot(fact)# Plot the actual and forecast values



Forecasting for Seasonal Differences

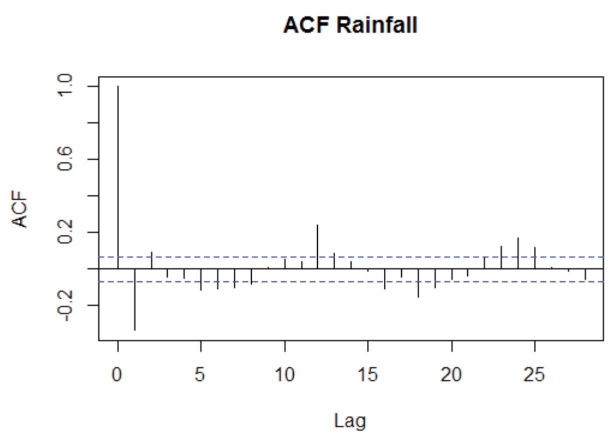
In this section, we have considered the rainfall data with differences. The same interpretation has been carried out for the below mentioned model.

plot (diff (Myts), main="Rainfall difference in Rayalaseema", ylab='Differenced Rainfall', col="red") #Graph for time series difference data

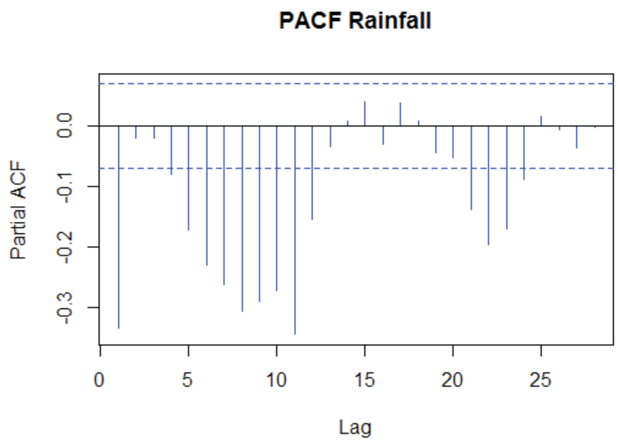


To check the details of ACF and PCF of Rain fall data differences.

Acf (ts (diff (Myts)), main='ACF Rainfall')



Pacf (ts (diff (Myts)), main='PACF Tractor Sales', col="green")



y<- auto.arima (diff (Myts), approximation=FALSE, trace=FALSE) # difference in data

print (y) # display the result

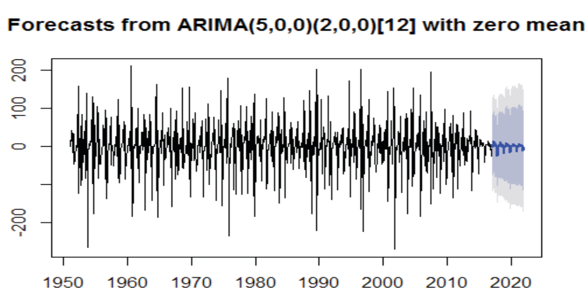
ARIMA(5,0,0)(2,0,0)[12] with zero mean

Coefficients:	AR (1)	AR (2)	AR (3)	AR (4)	AR (5)	SAR (1)	SAR (2)
Estimate	-0.8082	-0.5649	-0.4126	-0.2739	-0.1600	0.3970	0.3641
SE	0.0392	0.0507	0.0513	0.0466	0.0354	0.0332	0.0344
sigma^2 estimated as 2360				Log likelihood=-4195.66			
AIC= 8407.32		AICc=8407.51			BIC = 8444.71		

factdiff<- forecast (y, h=60)

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2017	-3.5388468	-65.80016	58.72247	-98.75931	91.68162
Feb 2017	4.63466545	-75.41920	84.68853	-117.79715	127.06648
Mar 2017	3.62490528	-76.61753	83.86734	-119.09531	126.34512
Apr 2017	4.32915406	-75.93147	84.58978	-118.41888	127.07719
May 2017	13.2618257	-67.02343	93.54708	-109.52388	136.04753
Jun 2017	4.26567270	-76.02472	84.55607	-118.52789	127.05924
Jul 2017	-1.18738607	-81.65869	79.28392	-124.25762	121.88285
Aug 2017	6.75960785	-73.94287	87.46208	-116.66418	130.18339
Sep 2017	0.84780732	-79.86421	81.55982	-122.59057	124.28618
Oct 2017	-0.03304655	-80.74581	80.67971	-123.47256	123.40647
Nov 2017	-24.9634002	-105.6775	55.75076	-148.40506	98.47826
Dec 2017	-5.94312923	-86.65736	74.77111	-129.38490	117.49864
Jan 2018	0.12147751	-84.41756	84.66051	-129.16982	129.41278
Feb 2018	3.14557391	-83.88612	90.17727	-129.95792	136.24907
Mar 2018	3.30576695	-83.75880	90.37034	-129.84800	136.45954
Apr 2018	2.49754940	-84.57009	89.56519	-130.66092	135.65601
May 2018	10.1841675	-76.88782	97.25616	-122.98095	143.34929
Jun 2018	3.33952546	-83.73322	90.41228	-129.82675	136.50580
Jul 2018	-0.62255427	-87.72234	86.47723	-133.83018	132.58507
Aug 2018	6.23843050	-80.89724	93.37410	-127.02408	139.50094
Sep 2018	-1.72694914	-88.86420	85.41030	-134.99187	131.53797
Oct 2018	1.52023270	-85.61714	88.65760	-131.74487	134.78534
Nov 2018	-18.9954775	-106.1330	68.14212	-152.26094	114.26998
Dec 2018	-4.68945692	-91.82707	82.44815	-137.95493	128.57602
Jan 2019	-1.24133106	-94.29538	91.81271	-143.55521	141.07255
Feb 2019	2.93698209	-93.82761	99.70157	-145.05169	150.92565
Mar 2019	2.63238930	-94.17734	99.44212	-145.42531	150.69009
Apr 2019	2.56796062	-94.24607	99.38199	-145.49633	150.63225
May 2019	8.87147994	-87.94848	105.6914	-139.20187	156.94483
Jun 2019	2.87861529	-93.94251	99.69974	-145.19651	150.95374
Jul 2019	-0.67953764	-97.54186	96.18279	-148.81768	147.45861
Aug 2019	4.93781590	-91.97788	101.8535	-143.28196	153.15759
Sep 2019	-0.37687460	-97.29482	96.54107	-148.60009	147.84634
Oct 2019	0.59150567	-96.32662	97.50963	-147.63197	148.81499
Nov 2019	-16.6299923	-113.5484	80.28846	-164.85398	131.59399
Dec 2019	-4.02553541	-100.9440	92.89294	-152.24955	144.19847
Jan 2020	-0.44856822	-99.85830	98.96116	-152.48263	151.58549
Feb 2020	2.31124223	-98.72662	103.3491	-152.21284	156.83532
Mar 2020	2.24863578	-98.80992	103.3071	-152.30708	156.80435
Apr 2020	1.92879176	-99.13171	102.9893	-152.62992	156.48750
May 2020	7.22985272	-93.83338	108.2930	-147.33302	161.79273
Jun 2020	2.35867135	-98.70507	103.4224	-152.20498	156.92232
Jul 2020	-0.49643569	-101.5782	100.5853	-155.08773	154.09486

Aug 2020	4.23162394	-96.87385	105.3371	-150.39586	158.85911
Sep 2020	-0.77838325	-101.8848	100.3281	-155.40742	153.85065
Oct 2020	0.78832582	-100.3182	101.8949	-153.84083	155.41748
Nov 2020	-13.5180298	-114.6247	87.58869	-168.14741	141.11135
Dec 2020	-3.30548815	-104.4122	97.80124	-157.93488	151.32390
Jan 2021	-0.63003524	-103.8275	102.5674	-158.45699	157.19691
Feb 2021	1.98687011	-102.5747	106.5484	-157.92625	161.89999
Mar 2021	1.85111607	-102.7275	106.4297	-158.08807	161.79030
Apr 2021	1.70068342	-102.8795	106.2809	-158.24098	161.64235
May 2021	6.10020845	-98.48227	110.6826	-153.84488	166.04530
Jun 2021	1.98444791	-102.5984	106.5673	-157.96130	161.93020
Jul 2021	-0.44449381	-105.0426	104.1536	-160.41360	159.52462
Aug 2021	3.47772843	-101.1403	108.0957	-156.52178	163.47724
Sep 2021	-0.44622687	-105.0651	104.1726	-160.44702	159.55457
Oct 2021	0.52831938	-104.0906	105.1472	-159.47258	160.52922
Nov 2021	-11.4213593	-116.0404	93.19773	-171.42245	148.57973
Dec 2021	-2.77790842	-107.3970	101.8411	-162.77900	157.22319



CONCLUSION

The data has been fitted to the ARIMA (5, 0, 1) (2, 0, 0) [12] model for rainfall of Rayalaseema. Augmented Dickey-Fuller Test has been tested for stationarity of the data. Basing on the p-value (p=0.01), the data has been stationary and we have applied for auto ARIMA to find and check the best model using R. We have made the interpretation basing on the AIC and BIC values of the model. The lowest AIC and BIC will give us the best fit of the forecast model. Based on auto ARIMA, the best fitted model has been found ARIMA (5, 0, 1) (2, 0, 0)[12], which has the seasonality. The prediction values and its graphs have been shown.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

REFERENCES

1. Alfaro, E. (2004). A Method for Prediction of California Summer Air Surface Temperature, Eos, Vol. 85, No. 51, 21 December 2004.
2. Anisimov O.A., (2001). Predicting Patterns of Near-Surface Air Temperature Using Empirical Data, Climatic Change, Vol. 50, No. 3, 297-315.
3. Box, G. E. P., Jenkins, G. M. & Reinsel, G. C. (1994). Time Series Analysis Forecasting and Control, 3rd ed., Englewood Cliffs, N.J. Prentice Hall.
4. Brunetti, M., Buffoni, L., Maugeri, M. & Nanni, T., (2000). Trends of minimum and maximum daily temperatures in Italy from 1865 to 1996. Theor. Appl. Climatol., 66, 49–60.
5. FAN Ke, (2009). Predicting Winter Surface Air Temperature in Northeast China, Atmospheric and Oceanic Science Letters, Vol. 2, NO. 1, 14−17.
6. Hejase, H.A.N. & Assi, A.H. (2012). Time-Series Regression Model for Prediction of Mean Daily Global Solar Radiation in Al-Ain, UAE, ISRN Renewable Energy, Vol. 2012, Article ID 412471, 11 pages.
7. J. C. Ramesh Reddy, T. Ganesh, M. Venkateswaran & P. R. S. Reddy (2017), “Rainfall Forecast of Andhra Pradesh Using Artificial Neural Networks”, International Journal of Current Research and Modern Education, Volume 2, Issue 2, Page Number 223-234. DOI: http://doi.org/10.5281/zenodo.1050435.
8. J. C. Ramesh Reddy, T. Ganesh, M. Venkateswaran, PRS Reddy (2017), Forecasting of Monthly Mean Rainfall in Coastal Andhra, International Journal of Statistics and Applications 2017, 7(4): 197-204. DOI: 10.5923/j.statistics.20170704.01.
9. Lee, J.H., Sohn, K. (2007). Prediction of monthly mean surface air temperature in a region of China, Advances in Atmospheric Sciences, Vol. 24, 3, 503-508.
10. Stein, M. & Lloret, J. (2001). Forecasting of Air and Water Temperatures for Fishery Purposes with Selected Examples from Northwest Atlantic, J. Northw. Atl. Fish. Sci., Vol. 29, 23-30.