International Journal of Statistics and Applications 2017, 7(4): 197-204
DOI: 10.5923/j.statistics.20170704.01

Forecasting of Monthly Mean Rainfall in Coastal Andhra

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

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

Abstract In this paper, forecasting of monthly mean rainfall of coastal Andhra (India) having seasonal autoregressive integrated moving average (SARIMA) model using R is discussed. We found that the ARIMA (1,0,0)(2,0,0)[12] has been fitted to the data and the significance test has been made by using lowest AIC and BIC values.

Keywords Box-Jenkins Methodology, SARIMA model, AIC & BIC

1. Introduction

A lot of variation can be seen from North to South and East to West of India. Top side of country is having a range of Mountains that starts from Jammu & Kashmir to Arunachal Pradesh; Middle part of country is having plains. Most of the south part of country is covered by sea. These parameters are responsible for the variation of climate that leads to cause of variations in rainfall that is why some parts of India are rich in rainfall and some parts of India are rain deficient.

In this blog, we have done analysis like forecasting of annual rainfall of Coastal Andhra 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 demanded 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.

2. Methodology

ARIMA models are capable of modelling 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.

2.1. 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 modelling 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.

2.2. SARIMA

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 $SARIMA(p,d,q)(P,D,Q)_m$ and the model is given by $\varphi_p(B)\Phi_p(B^m)\nabla^d\nabla_m^d Z_t = \theta_q(B)\Theta_Q(B^m)a_t$ 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, $B^m Z_t = Z_{t-m}$ and $\nabla = (1-B)$. Order of seasonality is represented by *m*. 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

* Corresponding author:
ganimsc2007@gmail.com (T. Ganesh)
Published online at http://journal.sapub.org/statistics
Copyright © 2017 Scientific & Academic Publishing. All Rights Reserved

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).

2.3. 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 d > 1, no constant is allowed as a quadratic or higher order trend is particularly dangerous when forecasting. The parameter μμ 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.

2.4. 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.

2.5. Modelling 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.

2.6. AIC and BIC

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 [-2logL + 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 behaviour 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=0.16, and BIC becomes like a significance test with alpha depending on sample size, e.g., 0.13 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.

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.

3. Forecasting of Rainfall of Coastal Andhra

The data has been taken to predict the rain fall of coastal 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 coastal 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 coastal Andhra, the following command can be use:

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

Summery details:

	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Coastal Andhra	0.00	12.85	59.50	84.72	140.53	410.50

The minimum and maximum rainfall is 0.00 and 410.50, first and third quartiles are 12.85 and 140.53. Basing on the quartile, the deviation is 63.84. Average rainfall of coastal is 84.42 and median rainfall is 59.50. Depending on the above summery, we cannot give any decision about the rainfall of coastal Andhra.

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

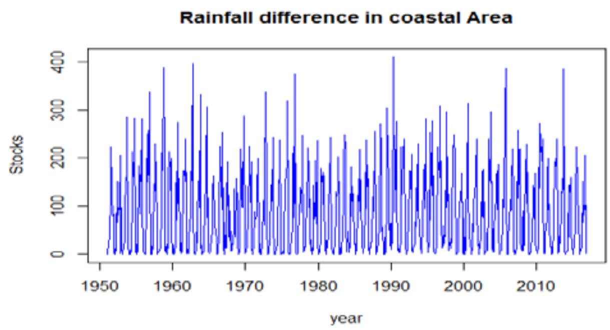
Myts <- ts (CA [, 3], start=c (1951, 1), end = c (2016, 12), frequency = 12) # Applying time series to data.

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.

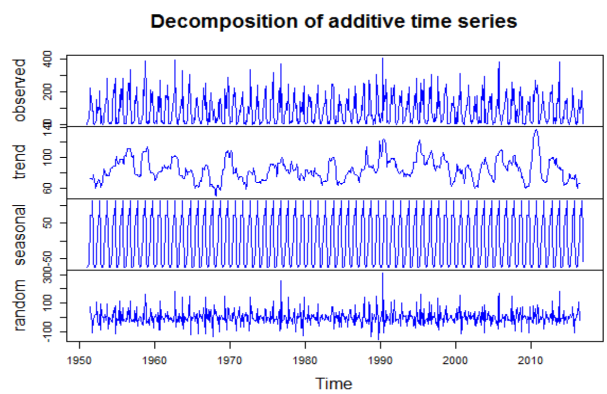
Plot (Myts, xlab='year', ylab = 'Stocks', main="Rainfall difference in coastal Area", col="blue") # Graph for time series data.

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

Augmented Dickey-Fuller Test
Dickey-Fuller = -12.829, Lag order = 9, p-value = 0.01
Alternative hypothesis: stationary



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



The above four graphs represents 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.

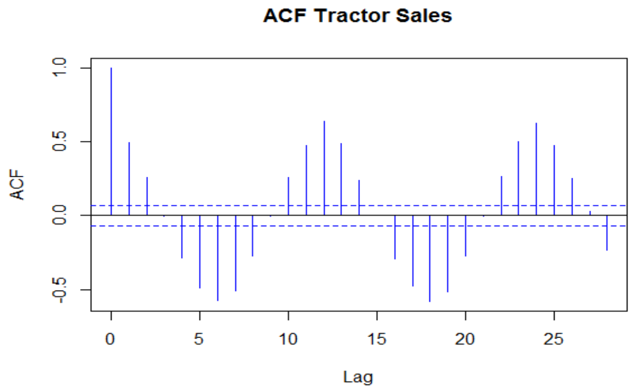
ARIMA model for data: we are using auto arima method for find forecasting model for data set.

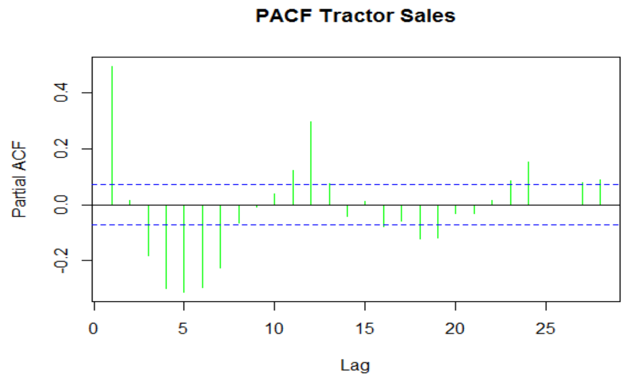
ARIMAfit <- auto.arima (Myts, approximation=FALSE, trace=FALSE) # to build forecasting model

To check the details of ACF and PCF of Rain fall data.
Acf (ts (Myts), main='ACF Tractor Sales’, col="blue")
Pacf (ts (Myts), main='PACF Tractor Sales’, col="green")

The entire document should be in Times New Roman. The font sizes to be used are specified in Table 1.

The size of a lower-case "j" will give the point size by measuring the distance from the top of an ascender to the bottom of a descender.





The ACF plot of the residuals from the ARIMA (1,0,0)(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 (1,0,0)(2,0,0)[12]. There are no other obvious candidate models.

3.1. ARIMA (1,0,0)(2,0,0)[12] with Non-zero Mean

Coefficients:	AR (1)	SAR (1)	SAR (2)	Mean
Estimate	0.0656	0.3897	0.3617	83.1067
SE	0.0377	0.0314	0.0335	8.5691
sigma^2 estimated as 3580		Log likelihood=-4366.76		
AIC=8743.52	AICc=8743.6	BIC=8766.89		

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

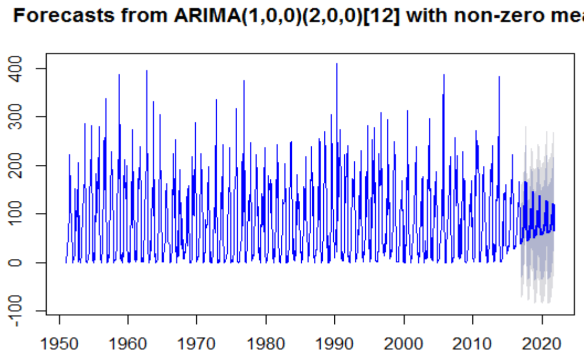
By applying auto ARIMA, we got the best fitted model which has the lowest AICc. When models are compared using AICc values, it is important that all models have the same orders of differencing. However, when comparing models using a test set, it does not matter how the forecasts were produced - the comparisons 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(1,0,0)(2,0,0)[12] model are shown in the figure below.

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

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2017	25.62493	-51.051177	102.3010	-91.641056	142.8909
Feb 2017	25.66684	-51.173953	102.5076	-91.851009	143.1847
Mar 2017	32.02999	-44.811511	108.8715	-85.488942	149.5489
Apr 2017	40.65928	-36.182216	117.5008	-76.859649	158.1782
May 2017	97.90564	21.064139	174.7471	-19.613294	215.4246
Jun 2017	86.90636	10.064860	163.7479	-30.612573	204.4253
Jul 2017	138.02755	61.186046	214.8690	20.508613	255.5465
Aug 2017	111.54306	34.701564	188.3846	-5.975869	229.0620
Sep 2017	102.43148	25.589983	179.2730	-15.087450	219.9504
Oct 2017	130.08825	53.246754	206.9298	12.569321	247.6072
Nov 2017	51.08534	-25.756157	127.9268	-66.433590	168.6043
Dec 2017	58.98300	-17.858504	135.8245	-58.535937	176.5019
Jan 2018	30.64249	-51.804198	113.0892	-95.448841	156.7338
Feb 2018	35.03598	-47.433990	117.5060	-91.090957	161.1629
Mar 2018	35.41754	-47.052532	117.8876	-90.709552	161.5446
Apr 2018	43.48308	-38.986989	125.9532	-82.644009	169.6102
May 2018	113.79586	31.325781	196.2659	-12.331239	239.9229

Jun 2018	78.57996	-3.890110	161.0500	-47.547130	204.7071
Jul 2018	108.41365	25.943573	190.8837	-17.713447	234.5407
Aug 2018	101.31229	18.842213	183.7824	-24.814807	227.4394
Sep 2018	94.54197	12.071895	177.0120	-31.585125	220.6691
Oct 2018	145.83557	63.365499	228.3056	19.708479	271.9627
Nov 2018	41.10696	-41.363111	123.5770	-85.020131	167.2341
Dec 2018	78.51460	-3.955477	160.9847	-47.612497	204.6417
Jan 2019	41.86761	-49.522977	133.2582	-97.902229	181.6374
Feb 2019	43.59490	-47.832168	135.0220	-96.230732	183.4205
Mar 2019	46.04545	-45.381772	137.4727	-93.780419	185.8713
Apr 2019	52.31020	-39.117019	143.7374	-87.515666	192.1361
May 2019	100.41966	8.992444	191.8469	-39.406204	240.2455
Jun 2019	82.71717	-8.710056	174.1444	-57.108704	222.5430
Jul 2019	112.83628	21.409057	204.2635	-26.989591	252.6621
Aug 2019	100.48818	9.060954	191.9154	-39.337693	240.3140
Sep 2019	94.55370	3.126480	185.9809	-45.272167	234.3796
Oct 2019	124.54746	33.120240	215.9747	-15.278407	264.3733
Nov 2019	55.15585	-36.271374	146.5831	-84.670022	194.9817
Dec 2019	72.59045	-18.836767	164.0177	-67.235415	212.4163
Jan 2020	48.05710	-47.038026	143.1522	-97.378343	193.4925
Feb 2020	50.31955	-44.791035	145.4301	-95.139540	195.7786
Mar 2020	51.41256	-43.698101	146.5232	-94.046641	196.8718
Apr 2020	56.77161	-38.339047	151.8823	-88.687587	202.2308
May 2020	100.95527	5.844611	196.0659	-44.503929	246.4145
Jun 2020	81.31736	-13.793299	176.4280	-64.141839	226.7766
Jul 2020	103.84698	8.736321	198.9576	-41.612219	249.3062
Aug 2020	96.46605	1.355396	191.5767	-48.993144	241.9252
Sep 2020	91.70425	-3.406403	186.8149	-53.754944	237.1634
Oct 2020	121.94812	26.837463	217.0588	-23.511077	267.4073
Nov 2020	57.02099	-38.089666	152.1316	-88.438207	202.4802
Dec 2020	77.34736	-17.763294	172.4580	-68.111834	222.8066
Jan 2021	54.52979	-43.670596	152.7302	-95.654741	204.7143
Feb 2021	56.03631	-42.177155	154.2498	-94.168222	206.2408
Mar 2021	57.34873	-40.864788	155.5622	-92.855885	207.5533
Apr 2021	61.70340	-36.510115	159.9169	-88.501212	211.9080
May 2021	96.32517	-1.888347	194.5387	-53.879444	246.5298
Jun 2021	82.26848	-15.945036	180.4820	-67.936132	232.4731
Jul 2021	101.94376	3.730236	200.1573	-48.260861	252.1484
Aug 2021	94.60052	-3.612997	192.8140	-55.604093	244.8051
Sep 2021	90.59808	-7.615444	188.8116	-59.606540	240.8027
Oct 2021	113.23422	15.020700	211.4477	-36.970396	263.4388
Nov 2021	62.82999	-35.383525	161.0435	-87.374621	213.0346
Dec 2021	77.05807	-21.155454	175.2716	-73.146551	227.2627

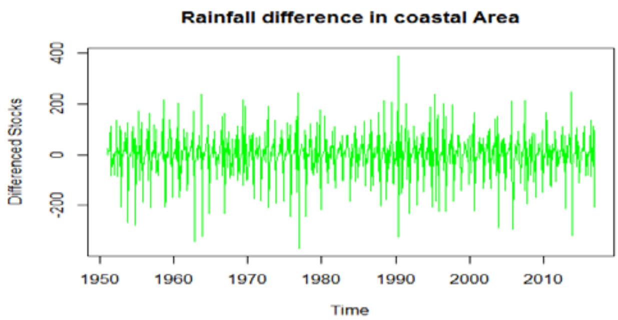
plot(fact) # Plot the actual and forecast values



3.2. 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.

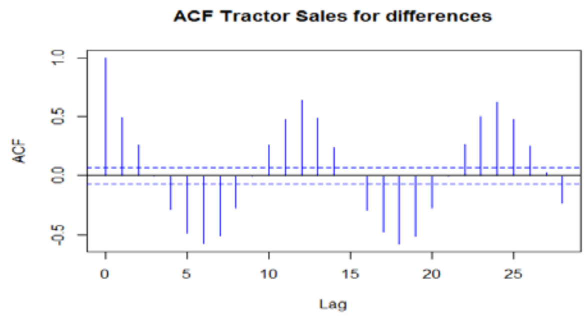
Plot (diff (Myts), main="Rainfall difference in coastal Area", ylab='Differenced Stocks', col="green") Graph for time series difference data

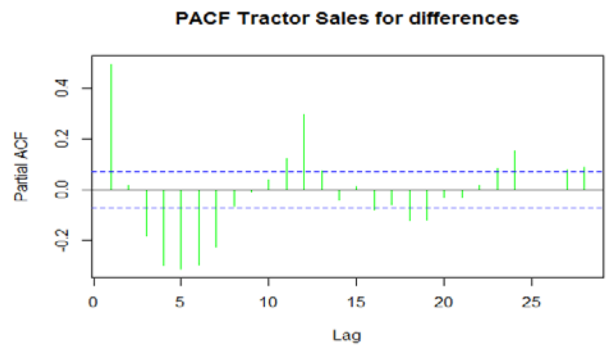


To check the details of ACF and PCF of Rain fall data differences.

Acf (ts (diff (Myts)), main='ACF Tractor Sales', col="blue")

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





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

3.3. ARIMA (5,0,0)(2,0,0)[12] with Zero Mean

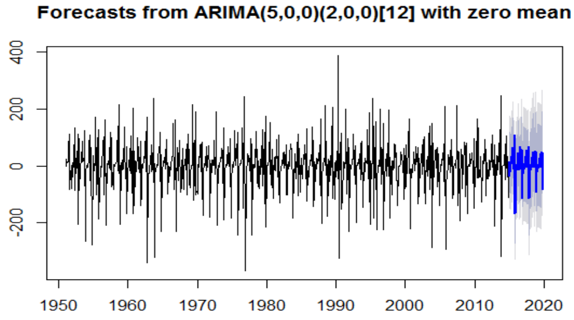
Coefficients:	AR (1)	AR (2)	AR (3)	AR (4)	AR (5)	SAR (1)	SAR (2)
Estimate	-0.7705	-0.5696	-0.4344	-0.3261	-0.1415	0.4248	0.3806
SE	0.0382	0.0477	0.0492	0.0455	0.0358	0.0335	0.0339
sigma^2 estimated as 4286				Log likelihood=-4432.65			
AIC=8881.3		AICc=8881.49			BIC=8918.69		

Fact<- forecast (ARIMAfit, h=60)

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2017	-81.895900	-165.79903	2.007230	-210.2147	46.42286
Feb 2017	13.403017	-92.51827	119.324303 -	148.5896	175.39564
Mar 2017	17.759524	-88.18104	123.700092	-144.2626	179.78164
Apr 2017	5.414171	-100.53293	111.361271	-156.6179	167.44628
May 2017	59.618513	-46.32965	165.566674	-102.4152	221.65224
Jun 2017	-5.633938	-111.93382	100.665949	-168.2056	156.93771
Jul 2017	47.002709	-59.31734	153.322762	-115.5998	209.60520
Aug 2017	-25.452518	-131.97881	81.073771	-188.3704	137.46538
Sep 2017	-9.121570	-115.64802	97.404878	-172.0397	153.79657
Oct 2017	30.139089	-76.38747	136.665646	-132.7792	193.05740
Nov 2017	-85.677162	-192.20834	20.854019	-248.6025	77.24822
Dec 2017	9.724583	-96.81238	116.261544	-153.2096	172.65880

Jan 2018	-36.110640	-148.30198	76.080696	-207.6925	135.47120
Feb 2018	10.609217	-105.05671	126.275149	-166.2866	187.50499
Mar 2018	5.325888	-110.34327	120.995047	-171.5748	182.22660
Apr 2018	7.236671	-108.43324	122.906579	-169.6652	184.13853
May 2018	75.875857	-39.79483	191.546544	-101.0272	252.77891
Jun 2018	-34.984855	-150.71584	80.746133	-211.9801	142.01042
Jul 2018	30.373891	-85.35959	146.107373	-146.6252	207.37298
Aug 2018	-7.394764	-123.16418	108.374648	-184.4488	169.65927
Sep 2018	-7.264840	-123.03428	108.504605	-184.3189	169.78925
Oct 2018	55.430234	-60.33922	171.199684	-121.6239	232.48433
Nov 2018	-114.183930	-229.95426	1.586402	-291.2394	62.87151
Dec 2018	40.237753	-75.53351	156.009022	-136.8191	217.29463
Jan 2019	-46.505507	-171.39715	78.386132	-237.5108	144.49977
Feb 2019	9.610638	-120.52290	139.744180	-189.4114	208.63272
Mar 2019	9.021187	-121.11735	139.159727	-190.0085	208.05091
Apr 2019	5.135150	-125.00486	135.275157	-193.8968	204.16712
May 2019	54.921528	-75.21899	185.062042	-144.1112	253.95427
Jun 2019	-17.007416	-147.23911	113.224273	-216.1796	182.16477
Jul 2019	30.791195	-99.44521	161.027598	-168.3882	229.97059
Aug 2019	-12.827209	-143.11750	117.463079	-212.0890	186.43459
Sep 2019	-6.557507	-136.84784	123.732826	-205.8194	192.70436
Oct 2019	35.017990	-95.27236	165.308343	-164.2439	234.27989
Nov 2019	-81.113798	-211.40540	49.177800	-280.3776	118.15001
Dec 2019	20.794983	-109.49809	151.088054	-178.4711	220.06104
Jan 2020	-33.498935	-167.97667	100.978804	-239.1649	172.16702
Feb 2020	8.120306	-128.89367	145.134283	-201.4245	217.66510
Mar 2020	5.859270	-131.15712	142.875662	-203.6892	215.40776
Apr 2020	4.935482	-132.08156	141.952520	-204.6140	214.48496
May 2020	52.207107	-84.81030	189.224510	-157.3429	261.75714
Jun 2020	-20.538840	-157.60072	116.523038	-230.1569	189.07921
Jul 2020	24.639945	-112.42404	161.703929	-184.9813	234.26122
Aug 2020	-8.263500	-145.35386	128.826863	-217.9251	201.39812
Sep 2020	-5.550491	-142.64088	131.539896	-215.2121	204.11116
Oct 2020	35.970839	-101.11955	173.061233	-173.6908	245.63250
Nov 2020	-77.912655	-215.00367	59.178363	-287.5753	131.74996
Dec 2020	24.146927	-112.94480	161.238653	-185.5168	233.81063
Jan 2021	-31.929179	-172.69928	108.840924	-247.2185	183.36012
Feb 2021	7.107114	-135.87650	150.090728	-211.5675	225.78168
Mar 2021	5.922235	-137.06351	148.907975	-212.7556	224.60005
Apr 2021	4.050942	-138.93540	147.037281	-214.6278	222.72968
May 2021	43.079751	-99.90685	186.066350	-175.5994	261.75888
Jun 2021	-15.197766	-158.22331	127.827777	-233.9365	203.54093
Jul 2021	22.185488	-120.84199	165.212968	-196.5562	240.92714
Aug 2021	-8.392018	-151.44256	134.658519	-227.1689	210.38490
Sep 2021	-4.853499	-147.90406	138.197059	-223.6304	213.92345
Oct 2021	28.607734	-114.44283	171.658298	-190.1692	247.38469
Nov 2021	-63.967814	-207.01892	79.083290	-282.7456	154.80997
Dec 2021	18.171964	-124.87976	161.223692	-200.6068	236.95070

Plot (fact)



4. Conclusions

The data has been fitted to the ARIMA (5, 0, 0) (2, 0, 0)[12] model for rainfall of coastal Andhra. 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, 0) (2, 0, 0)[12], which has the seasonality. The prediction values and its graphs have been shown.

ACKNOWLEDGEMENTS

The authors thank the referees for their valuable comments which have improved the presentation of the paper.

REFERENCES

[1] Alfaro, E. (2004). A Method for Prediction of California Summer Air Surface Temperature, Eos, Vol. 85, No. 51, 21.

[2] Anisimov O.A., (2001). Predicting Patterns of Near-Surface Air Temperature Using Empirical Data, Climatic Change, Vol. 50, 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, 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] 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.

[8] 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.