**Time Series Analysis**

**Statistical Data Analysis XIII (Part- C)**

**STAT-4109**

**Contents**

**Problem: -1**

1. Generating Random number from Normal distribution. Conver into quarterly and monthly time series. Page-1
2. Time series analysis ofquarterly beer production in Australia (in megaliters) from 1956: Q1 to 1975: Q2. Page-3

**Problem: -2**

Air Passengers data analysis from 1949-160. Page-5

**Problem: -3**

Decomposition and Modeling Data Air Passengers. Page-9

**Problem: -4**

Visualization of data by ggplot2.

1. The a10 dataset, which is supposed to represent monthly anti-diabetic drug subsidy in Australia from 1991 to 2008. Page-13
2. Total quarterly beer production in Australia (in megaliters) from 1956: Q1 to 2010: Q2. Page-15
3. Closing stock prices of GOOG from the NASDAQ exchange, for 1000 consecutive trading days between 25 February 2013 and 13 February 2017. Adjusted for splits. goog200 contains the first 200 observations from goog. Page-17

**Problem: -5**

Analyzing and forecasting monthly rainfall of Bangladesh.

**Problem: -1**

# Generating random number from normal distribution-

x=rnorm(100,0,1)

help(ts)

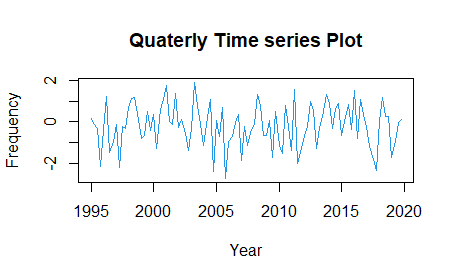
# Creating a time series

quater\_ts=ts(data = x,start = 1995,frequency = 4)

#Creating time series plot for quarterly data #

plot(quater\_ts,col=4,ylab="Frequency",

xlab="Year",main="Quaterly Time series Plot")



**Figure-1.1**

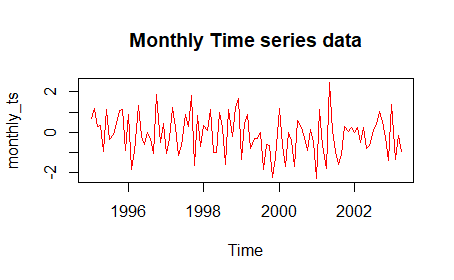
**Comment** Those the quarterly and monthly data are the stationary. cause data form normal distribution, Have equal mean and variance.

#Creating time series plot for monthly data

monthly\_ts=ts(data = x,start = 1995,frequency = 12)

#--------------Plot--------------------#

plot(monthly\_ts,col="red",main="Monthly Time series data")



**Comment:** The plot show data are stationary because sample draw from normal distribution.

#-Detection of tend and get a stationary time series

# install.packages("fpp")

library(fpp)

library(forecast)

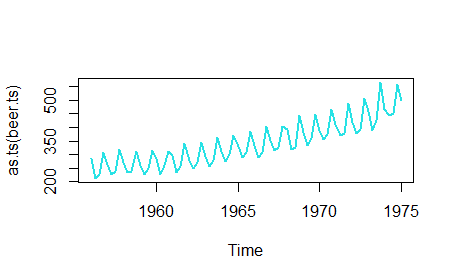
data(ausbeer)

View(ausbeer)

ausbeer

beer.ts=window(ausbeer,start=1956,end=1975)

plot(as.ts(beer.ts),col=5,lwd=2)



**Comment:** Upper tend data.

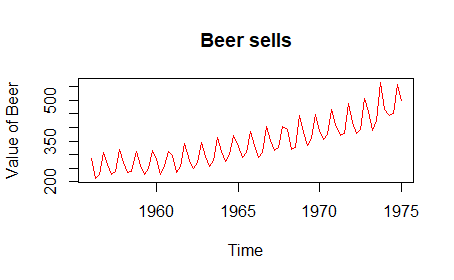
#Creating a moving average that will be close to trend

beer.trend=ma(beer.ts,order = 4,centre = T)

#-Plot trend a MA together.

plot(as.ts(beer.ts),col="red",ylab="Value of Beer",main="Beer sells")

lines(beer.trend,col=6,lwd=1.5)



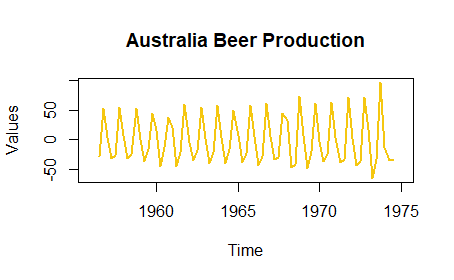
Comment: After perform the moving average method. We get the approximately equal variance of the data set.

#-Removing the trend from the time series

beer.detrend=beer.ts-beer.trend

#-Plot the detrend data-

plot(as.ts(beer.detrend),col=7,main="Australia Beer Production",ylab="Values",lwd=2)



Comment: Now we can see mean and variance are equal. The data set is stationary.

#But in real life problem all data sets are not stationary.

#In this situation we have to convert data into a stationary data.

**Problem: -2**

**Q: Air Passengers data analysis from 1949-160.**

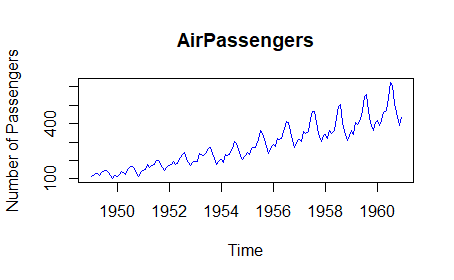
data("Airpassenger")

AirPassengers

#-Understanding and Preparing data.

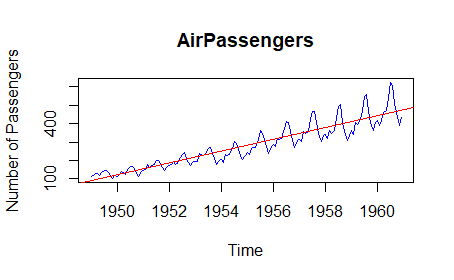
boxplot(AirPassengers~cycle(AirPassengers))

plot(AirPassengers,main="AirPassengers",ylab="Number of Passengers”, col="Blue",lwd=1.5)



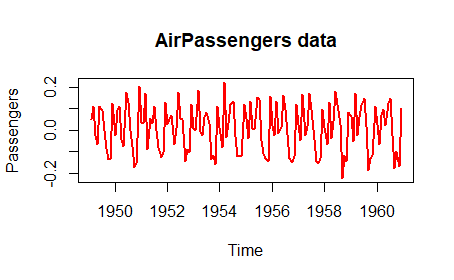
**Comment:** Upward and seasonality Trend data.

abline(lm(AirPassengers~time(AirPassengers)),lwd=1.3,col="red")



#-Make it stationary by making variance constant---

plot(diff(log(AirPassengers)),col="red",lwd=2,ylab="Passengers",main="AirPassengers data")



#-After stationary data we can fit ARIMA model-

#- we need tseries library

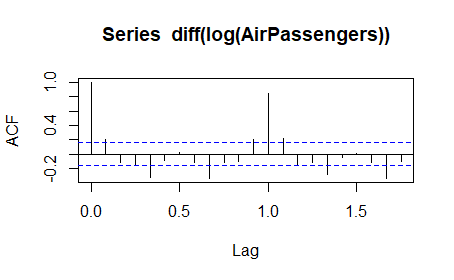
library(tseries)

# -ARIMA(P=Order of AR,D=differencing order,Q= order of MA)

#-Selecting the value of p(AR)

acf(AirPassengers)

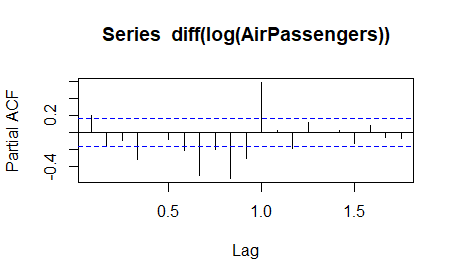
acf(diff(log(AirPassengers)))



**Comment**: Here the number of line cross or touch the blue line is no. of AR. we avoid the for zero value and the data is sessional that's why q=1.

# -Selecting the value of P-

pacf(diff(log(AirPassengers)))



#- Fitting an ARIMA(0,1,1)

# help(arima)

m\_fit=arima(log(AirPassengers),order = c(0,1,1),seasonal =list(order=c(0,1,1),period=12))

m\_fit

#-Forecast for next 10 years-

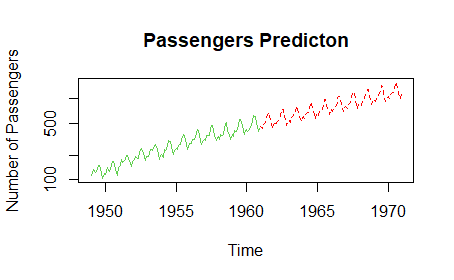
pred=predict(m\_fit,n.ahead = 10\*12)

pred

#-Final prediction-

finalpred=exp(pred$pred)

ts.plot(AirPassengers,finalpred,log="y",lty=c(1:3),col=c(3,'red'),main="Passengers Predicton",ylab="Number of Passengers")



#----Checking accuracy of the prediction.

train\_data=ts(AirPassengers,frequency = 12,start=c(1949,1),end = c(1959,12))

m\_fit2=arima(log(train\_data),order = c(0,1,1),seasonal =list(order=c(0,1,1),period=12))

m\_fit2

#---Predict for one year

pred2=predict(m\_fit2,n.ahead = 1\*12)

pred2

pred3=exp(pred2$pred)

pred3

pred\_1960=round(pred3,0)

true\_1960=tail(AirPassengers,12)

df=data.frame(pred\_1960,true\_1960)

df

Comment: The difference between the true value and the predicted value is the smallest so the accuracy of the model is large.

**Problem: -3**

Decomposition and Modeling Data Air Passengers.

data("AirPassengers")

View(AirPassengers)

AirPassengers

#-Preparing the time series data.

AP=ts(AirPassengers,frequency = 12,start = c(1949,1))

#----Understanding data.

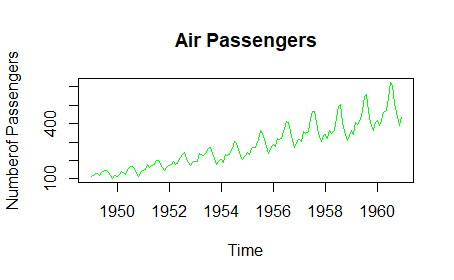
1. Trend(T)

2.Seasonal(S)

3.Cyclical(C)

4.Irregular(I)

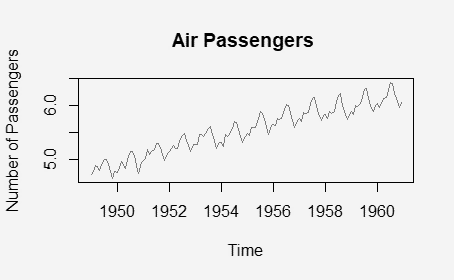
plot(AP,col="green",lwd=1.4,main="Air Passengers",ylab="Numberof Passengers")



#-Log-transform to fix-up variation.

AP2=log(AP)

plot(AP2,lwd=1.2,col="Yellow",main="Air Passengers ",ylab="Number of Passengers") #here we see the variance is same.



#--Decompose-

DAP=decompose((AP2))

DAP$figure

plot(DAP$figure,

type='b',

xlab = 'Month',

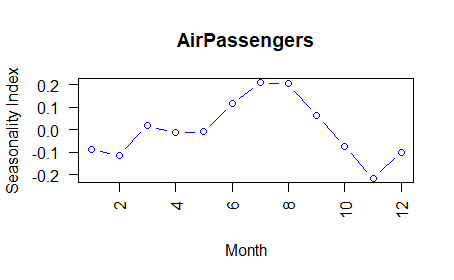
ylab = 'Seasonality Index',

col="blue",

las=2,

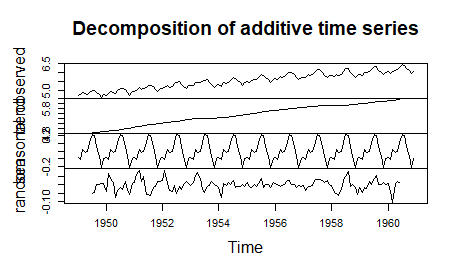
main = "AirPassengers",

lwd=1.3)



#--Decompose plot.

plot(DAP)



#-ARIMA(p,d,q) model

library(forecast)

model\_fit=auto.arima(AP2)

model\_fit

**Output:**

Series: AP2

ARIMA(0,1,1)(0,1,1)[12]

Coefficients:

ma1 sma1

-0.4018 -0.5569

s.e. 0.0896 0.0731

sigma^2 = 0.001371: log likelihood = 244.7

AIC=-483.4 AICc=-483.21 BIC=-474.77

#-Plotting the residuals of the fit model

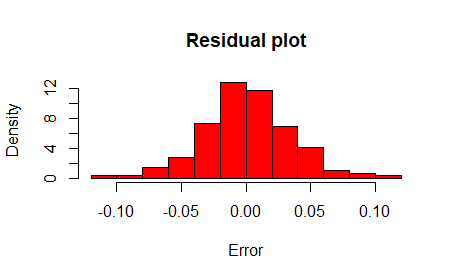
hist(model\_fit$residuals,

col='red',

main = 'Residual plot',

xlab = 'Error',

freq=F)

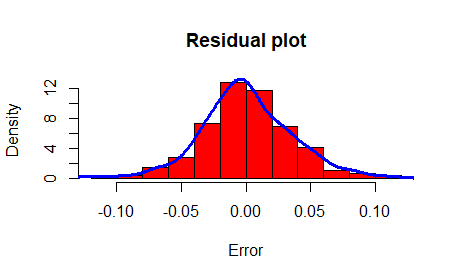


**Comment:** The residuals plot shows the errors are normally distributed.

#- Adding line in the histogram-

lines(density(model\_fit$residuals),

col='blue',lw=3)

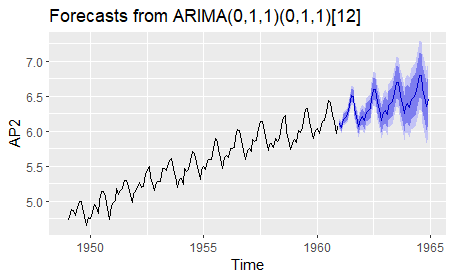


#-Forecast for the next 4 years-

pred=forecast(model\_fit,4\*12)

library(ggplot2)

autoplot(pred) **#deep blue 80% CI and light blue 95% CI.**



#Checking accuracy

accuracy(pred)

|  |
| --- |
| ME RMSE MAE MPE  Training set 0.0005730622 0.03504883 0.02626034 0.01098898  MAPE MASE ACF1  Training set 0.4752815 0.2169522 0.01443892 |
|  |
| |  | | --- | |  | |

**Problem: -4**

Visualization of data by ggplot2.

The a10 dataset, which is supposed to represent monthly anti-diabetic drug subsidy in Australia from 1991 to 2008.

library(fpp)

library(fpp2)

library(forecast)

library(ggplot2)

# help("autoplot")

data(a10)

a10

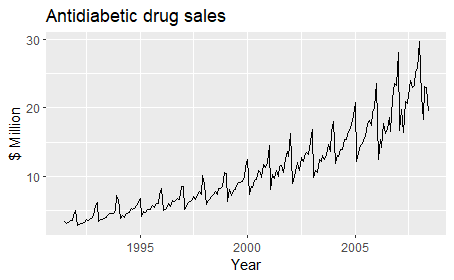
#-Plotting the dataset

autoplot(a10)+

ggtitle("Antidiabetic drug sales")+

ylab("$ Million")+

xlab("Year")



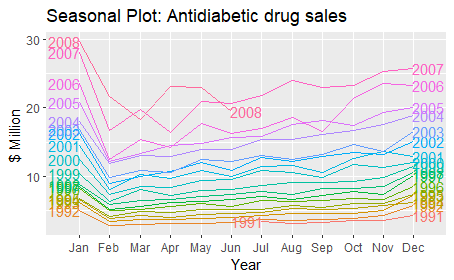
#-Seasonal plot

ggseasonplot(a10,year.labels = T,year.labels.left = T)+

ggtitle("Seasonal Plot: Antidiabetic drug sales")+

ylab("$ Million ")+

xlab("Year")



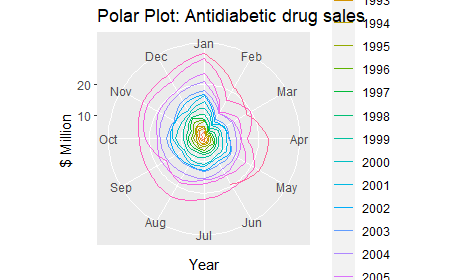
#-Polar seasonal plot-

ggseasonplot(a10,polar =T)+

ggtitle("Polar Plot: Antidiabetic drug sales")+

ylab("$ Million ")+

xlab("Year")



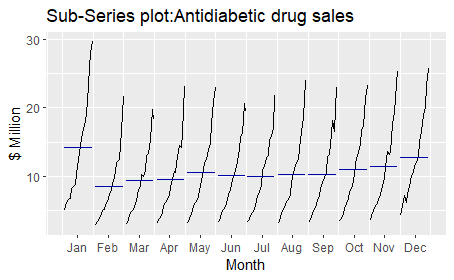
#--Polar seasonal plot

ggseasonplot(a10,polar =T)+

ggtitle("Polar Plot: Antidiabetic drug sales")+

ylab("$ Million ")+

xlab("Year")



#---**Visualizing ausbeer data.**

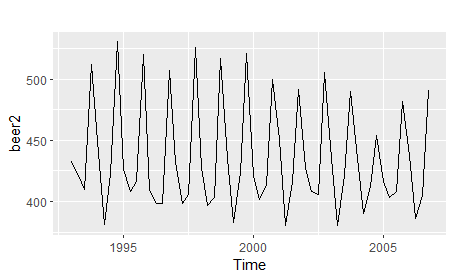
**Total quarterly beer production in Australia (in megaliters) from 1956: Q1 to 2010: Q2**.

data("ausbeer")

ausbeer

beer2=window(ausbeer,start=1993,end=c(2006,4))

autoplot(beer2)



#-Simple forecasting methods.

1. Average

2.Naive (work with most recent values)

3.Seasonal Naive

4.Drift

beerfit1=meanf(beer2,h=12)

beerfit2=rwf(beer2,h=12)

beerfit3=snaive(beer2,h=12)

#We plot full data set

autoplot(window(ausbeer,start=1993))+

autolayer(beerfit1,series="Mean",PI=F)+

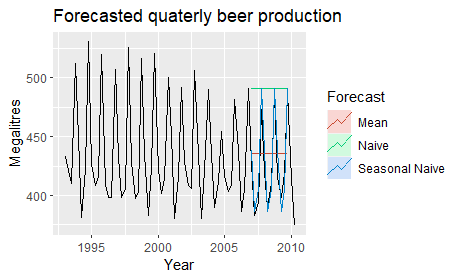
autolayer(beerfit2,series="Naive",PI=F)+

autolayer(beerfit3,series="Seasonal Naive",PI=F)+

xlab("Year")+ylab("Megalitres")+

ggtitle("Forecasted quaterly beer production ")+

guides(colour=guide\_legend(title = "Forecast"))



#--------------------Accuracy Checking-------------------------#

beer3=window(ausbeer,start=2008)

accuracy(beerfit1,beer3)

accuracy(beerfit2,beer3)

accuracy(beerfit3,beer3)

**Output:**

ME RMSE MAE MPE

Training set -2.436072e-14 43.42591 35.19005 -0.928920

Test set -6.892857e+00 36.33196 33.19643 -2.269843

MAPE MASE ACF1 Theil's U

Training set 7.875648 2.395134 -0.11704504 NA

Test set 7.659919 2.259443 -0.07153733 0.7810277

> accuracy(beerfit2,beer3)

ME RMSE MAE MPE

Training set 1.054545 65.06416 54.47273 -0.7865823

Test set -62.500000 71.96353 62.50000 -15.3314577

MAPE MASE ACF1 Theil's U

Training set 12.10955 3.707568 -0.22916007 NA

Test set 15.33146 4.253927 -0.07153733 1.45868

> accuracy(beerfit3,beer3)

ME RMSE MAE MPE MAPE

Training set -1.076923 17.32162 14.69231 -0.3312997 3.415323

Test set -1.500000 12.51000 10.25000 -0.3069842 2.457364

MASE ACF1 Theil's U

Training set 1.000000 -0.2880890 NA

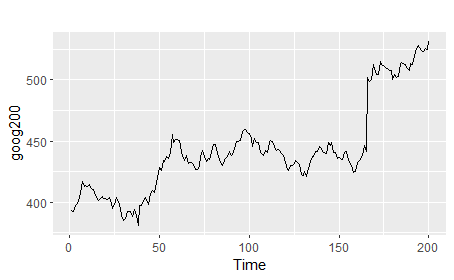
Test set 0.697644 -0.1108185 0.2180679

#-**Another data.**

Closing stock prices of GOOG from the NASDAQ exchange, for 1000 consecutive trading days between 25 February 2013 and 13 February 2017. Adjusted for splits. goog200 contains the first 200 observations from goog.

goog200

autoplot(goog200)



#-Forecast for next 40

googf1=meanf(goog200,h=40)

googf2=rwf(goog200,h=40)

googf3=rwf(goog200,drift=T,h=40)

#--------Plotting the data subset of goog200-----

autoplot(subset(goog200,end = 240))+

autolayer(googf1,series="Mean",PI=F)+

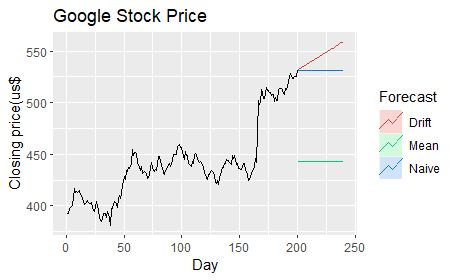
autolayer(googf2,series="Naive",PI=F)+

autolayer(googf3,series="Drift",PI=F)+

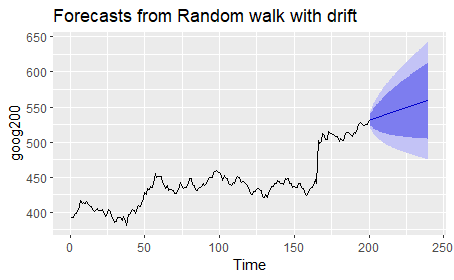
xlab("Day")+ylab("Closing price(us$")+

ggtitle("Google Stock Price")+

guides(colour=guide\_legend(title = "Forecast"))



autoplot(googf3)



**Comment:** Deep blue 80% CI and the light blue 95% prediction.

**Problem: -5**

**#-Analyzing and forecasting monthly rainfall of Bangladesh.**

df=read.csv(file.choose(),header = T,sep = ",")

df

head(df)

View(df)

#--Packages--#

library(ggplot2)

library(dplyr)

library(lubridate)

library(forecast)

#--Preparing data for analysis we take Rajshahi division data.

df1<-filter(df,df$Year>="2007")

df1=filter(df1,Station=="Rajshahi")

View(df1)

df2=select(df1,Rainfall)

df2

#-Convert the value in continuous format.

df2$Rainfall=as.numeric(df2$Rainfall)

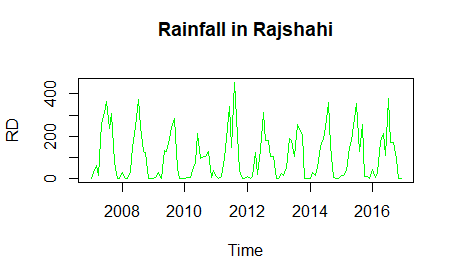
#-Convert into a time series data.

RD=ts(df2$Rainfall,start = 2007,frequency = 12)

RD

#-Time series Plot-

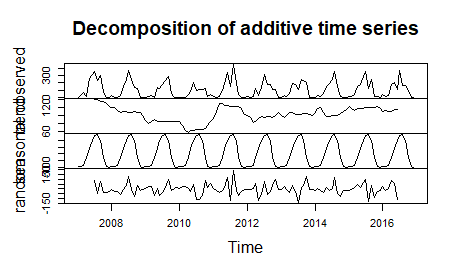
plot(RD,lwd=1.2,col="green",main="Rainfall in Rajshahi")



#--Decompose

RDR=decompose(RD)

plot(RDR)



#--Fitting an arima model

fit\_model=auto.arima(RD)

fit\_model

Series: RD

ARIMA(0,0,0)(2,1,0)[12]

Coefficients:

sar1 sar2

-0.7061 -0.3468

s.e. 0.0922 0.0936

sigma^2 = 5435: log likelihood = -620.14

AIC=1246.28 AICc=1246.51 BIC=1254.33

#--Forecast for next four years-

pred=forecast(fit\_model,4\*12)

autoplot(pred,PI=T)+

ylim(0,550)

