TSA FINAL REPORT:

Yuxiang_Ren

Spring 2023

Contents

Abstract

Introduction

Method (Data Processing)

Data

Three datasets used in the project were collected from EDI Data Portal, including:

1. North Temperate Lakes LTER: Phytoplankton - Madison Lakes Area 1995 – current;(Magnuson & H.Stanley, 2022)
2. North Temperate Lakes LTER: Physical Limnology of Primary Study Lakes 1981 – current;
3. North Temperate Lakes LTER: Chemical Limnology of Primary Study Lakes: Nutrients, pH and Carbon 1981 – current.

The three files respectively record the water body phytoplankton information, physical information and chemical information of multiple lakes in the Wisconsin range. We analyzed these data at the beginning stage to screen out suitable research subjects, including the target lake, and primary algae responsible for

Table 1: Site Information

Site	Observation Date Count
ME	402
MO	355
WI	23
FI	1

Table 2: Division level Total Biomass (mg/L)

Division	Count	Total Biomass	Max Biomass	Min Biomass	Mean Biomass
Cyanophyta	8581	1824.25213	76.0000	0.00e+00	0.2125920
Bacillariophyta	1914	378.01888	13.4028	3.19e-05	0.1975020
Chlorophyta	4368	244.95205	84.6924	0.00e+00	0.0560788
Cryptophyta	1876	76.11009	2.9630	3.63e-05	0.0405704
Pyrrhophyta	415	29.12950	5.3194	0.00e+00	0.0701916

blooms. First, we chose Mendota Lake (ME) for this project, as it has more time measurement data compared to other lakes, which might be more conducive to time series analysis and obtaining more reliable results (Table 1). Second, to obtain information on dominant species that may cause water blooms, we accumulated the biomass of algae from different divisions and considered the algae with the highest total biomass to be the main contributor to water bloom outbreaks. It is worth noting that the original data records the biomass of specific algal species on the observation day. Therefore, to obtain division-level data, we summed the biomass of all species within the same division on the same day to obtain the biomass information for the division. The result shows that the dominant division is Cyanophyta, which is also consistent with other studies (Table 2)[Brock (2012)](Beversdorf, 2015).

After identifying the target lake and algal division, we cleaned and combined the three data tables. The following are the data cleaning steps:

a. Integrate the phytoplankton data according to lakeid, sampledate, depth range, and division to obtain the biomass information of each division on the observation day. Then, filter out all data with a lake id of Mendota and a division of Cyanophyta.
b. Filter out the physical and chemical information of Lake Mendota. Considering that the original data records information at different depths on the same observation day, we calculated the average of all environmental data at depths of 0-8m, which correspond to the depths mentioned in the algae information. It is worth noting that on some dates, the depth of the algae information is 0-2m, and in these cases, we used the average environmental data for 0-2m.
c. Based on the sampling date and depth range, we combined these data together (Table 3).

d. We averaged the data monthly and used the zoo function (na.approx, rule = 2) to fill in NA values. Due to this method is not suitable for filling in NA values at the beginning of data, data before 1996 were removed.
e. The final dataset includes dates (from 1996 to December 2020), temperature, total nitrogen, total phosphorus, and biomass (Table 4).

Table 3: rawdata

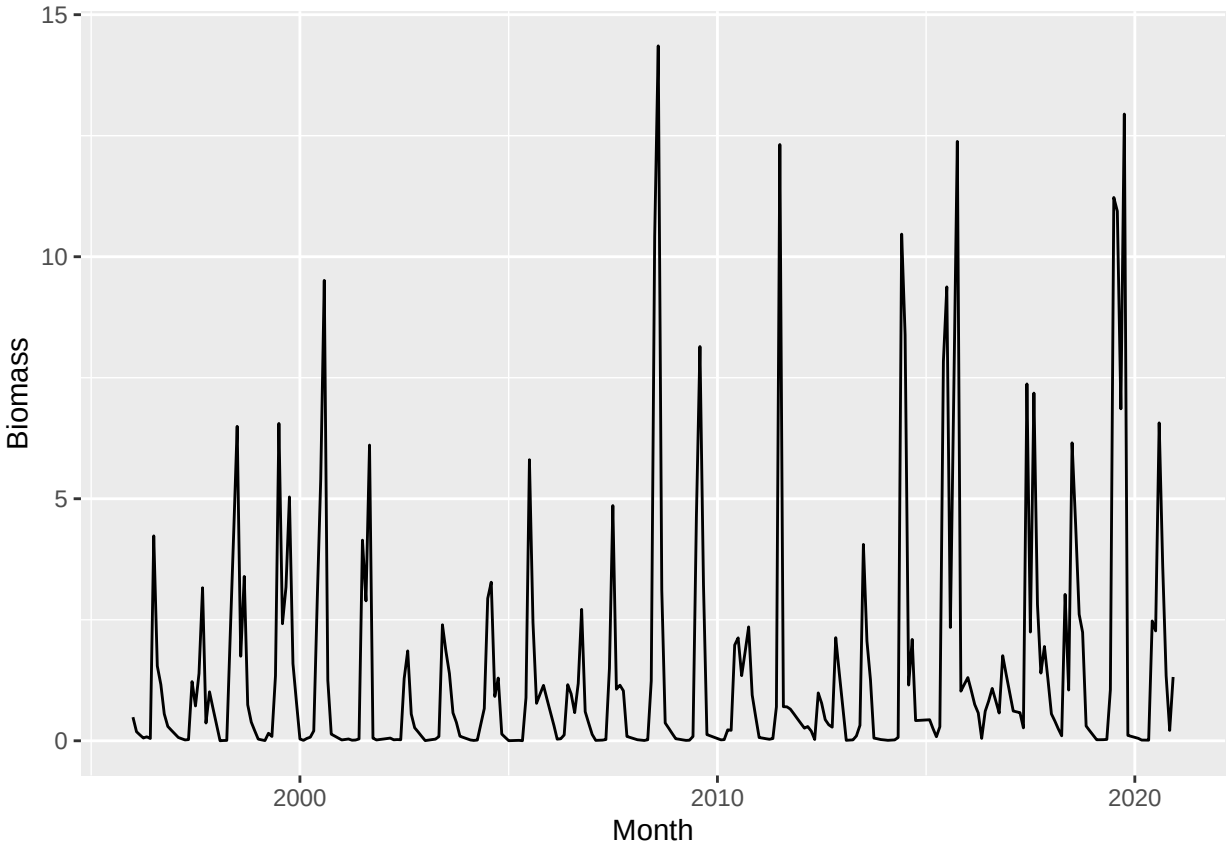
lakeid	sampledate	total_biomass	Temperature	date_diff	TN	TP
ME	1995-01-24	0.0128965	NA	NA	NA	NA
ME	1995-03-28	0.0013183	NA	63	NA	NA
ME	1995-04-11	0.0017578	NA	14	NA	NA
ME	1995-04-24	0.0019157	NA	13	NA	NA
ME	1995-05-23	0.8052959	13.64444	29	0.7305	0.0895000
ME	1995-06-06	0.0738443	16.84444	14	0.7195	0.0756667

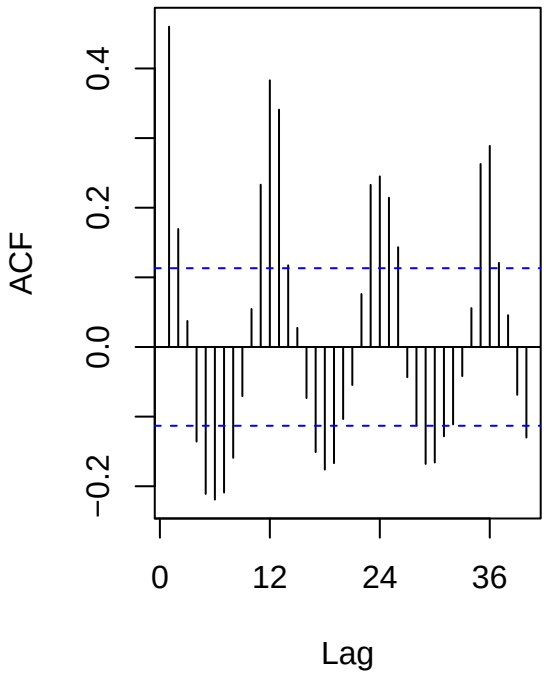
Table 4: Final Data

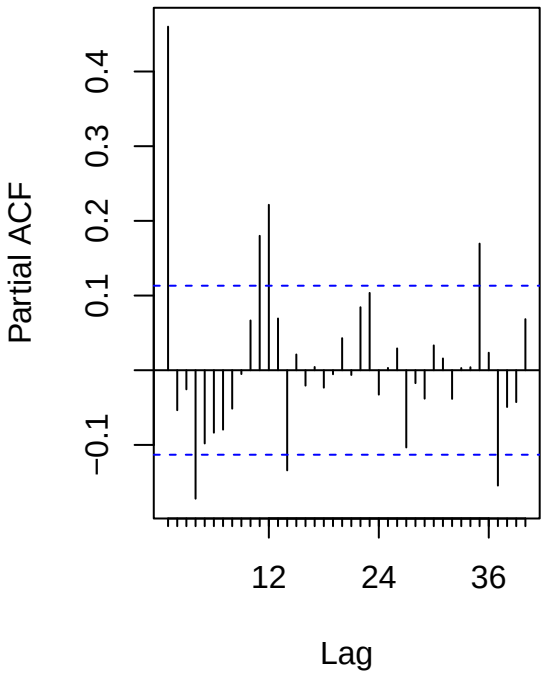
date	Temperature	TN	TP	Biomass
1996-01-01	3.702222	0.7277500	0.1163333	0.4872730
1996-02-01	5.104444	0.7835000	0.1096667	0.1878542
1996-03-01	6.506667	0.8120556	0.1080556	0.1224752
1996-04-01	7.908889	0.8406111	0.1064444	0.0570962
1996-05-01	9.311111	0.8691667	0.1048333	0.0817296

Result (1)

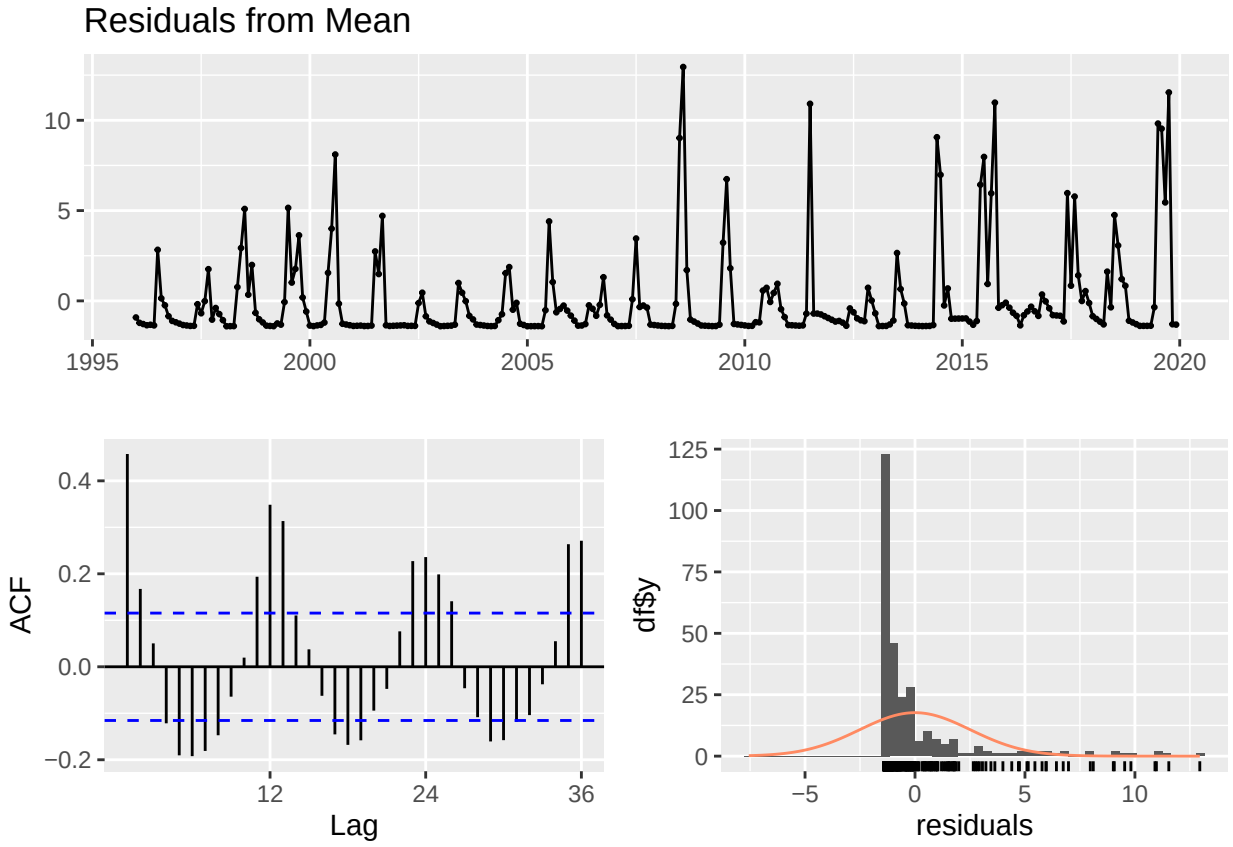
Original (1996-2020)





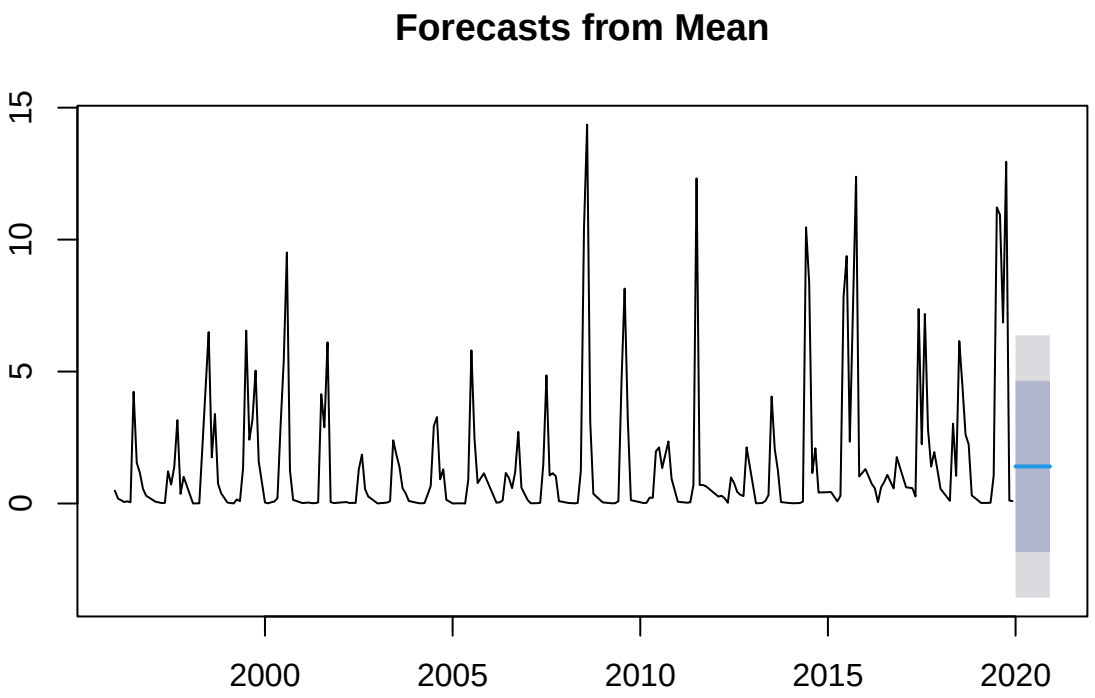



```
# Model 1: Arithmetic mean
# The meanf() has no holdout option
MEAN_seas <- meanf(y = ts_biomass, h = 12)
checkresiduals(MEAN_seas)
```

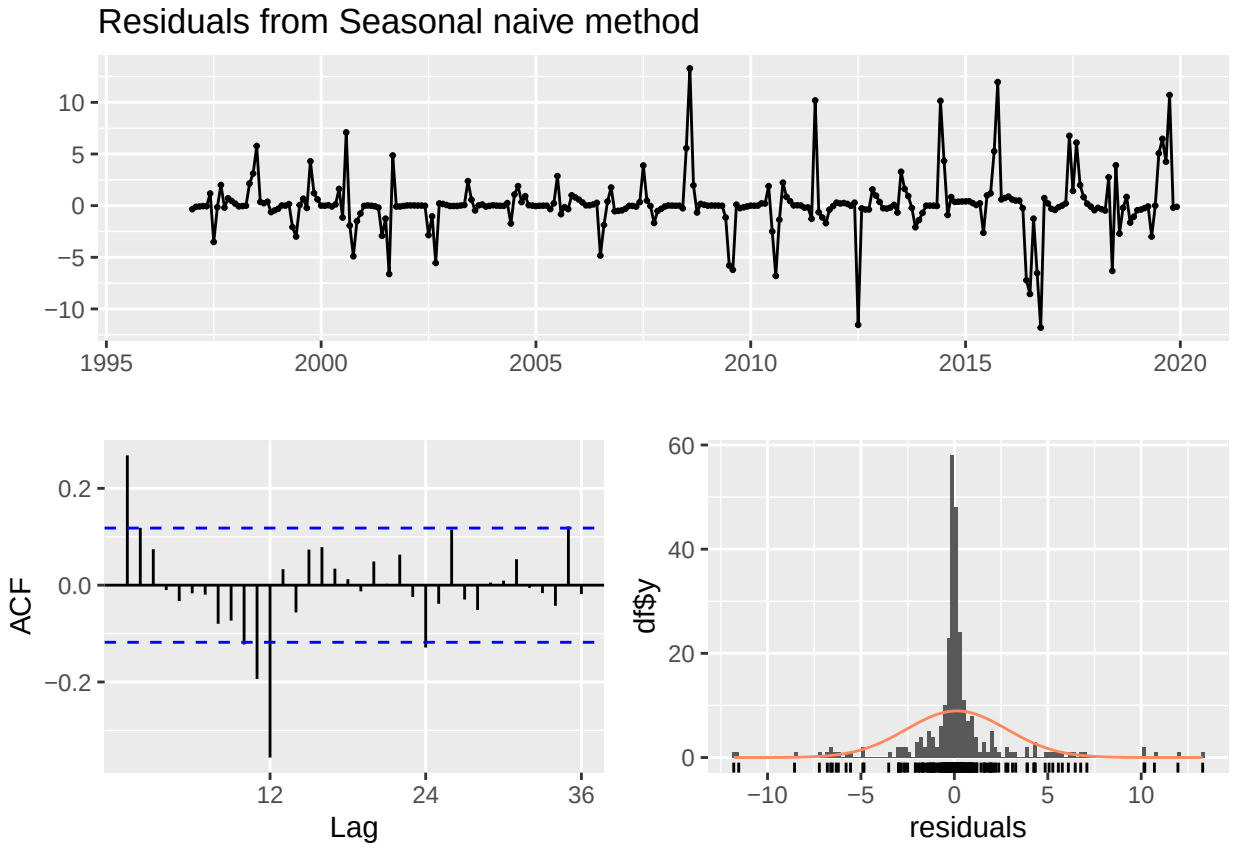


```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 258.95, df = 23, p-value < 2.2e-16
##
## Model df: 1.   Total lags used: 24
```

```
plot(MEAN_seas)
```

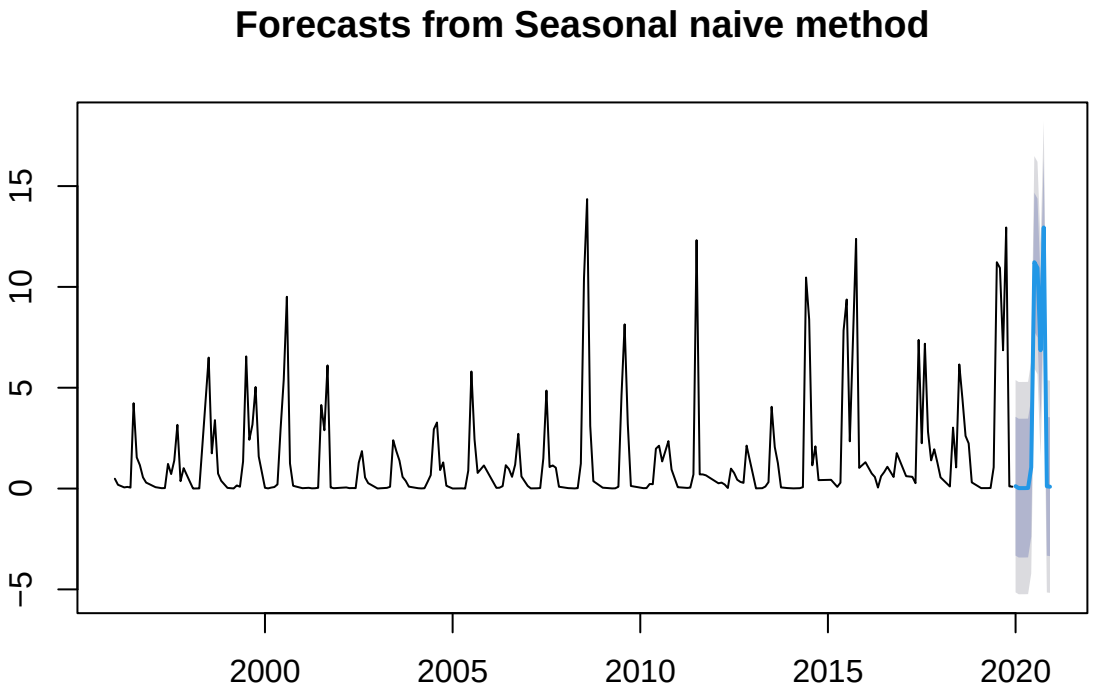



```
# Model 2: Seasonal naive
SNAIVE_seas <- snaive(ts_biomass, h=12, holdout=FALSE)
checkresiduals(SNAIVE_seas)
```

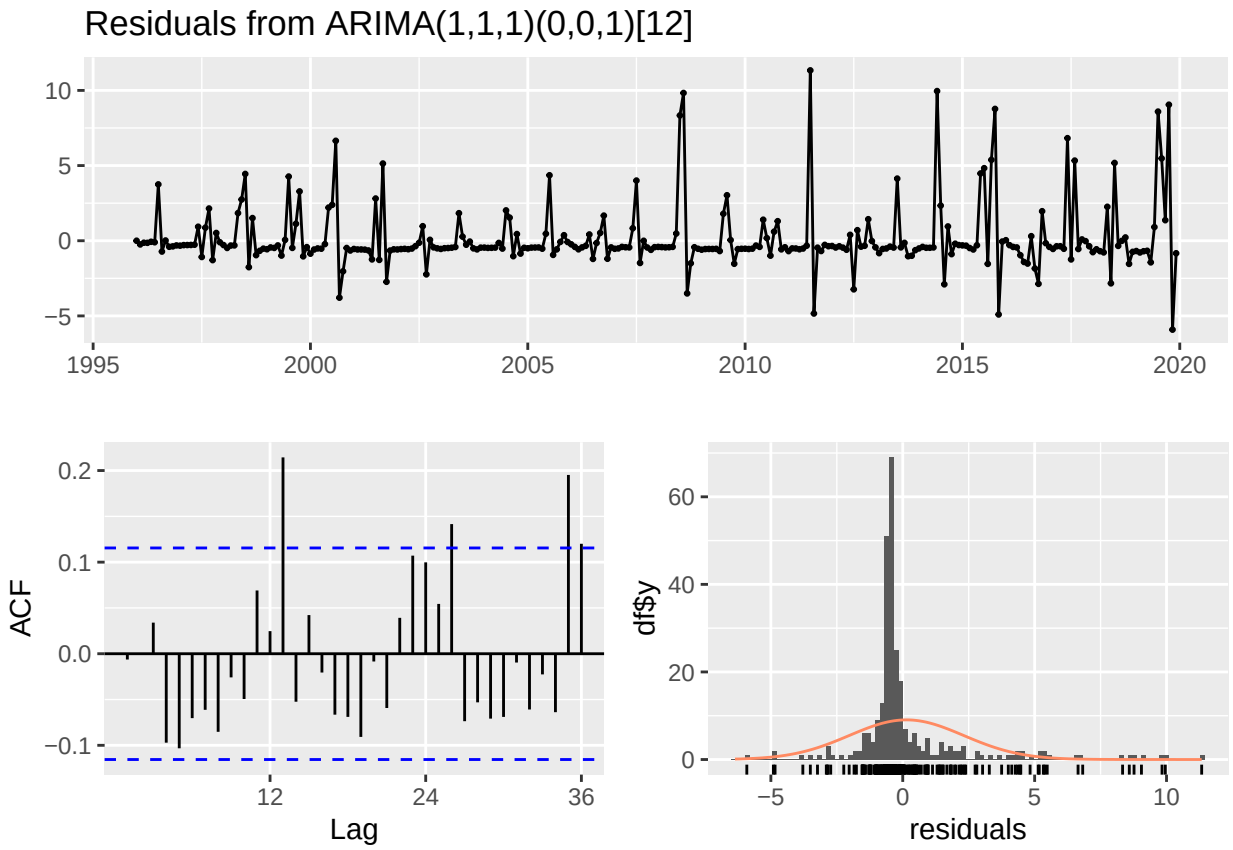



```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 93.651, df = 24, p-value = 3.556e-10
##
## Model df: 0.   Total lags used: 24
```

```
plot(SNAIVE_seas)
```

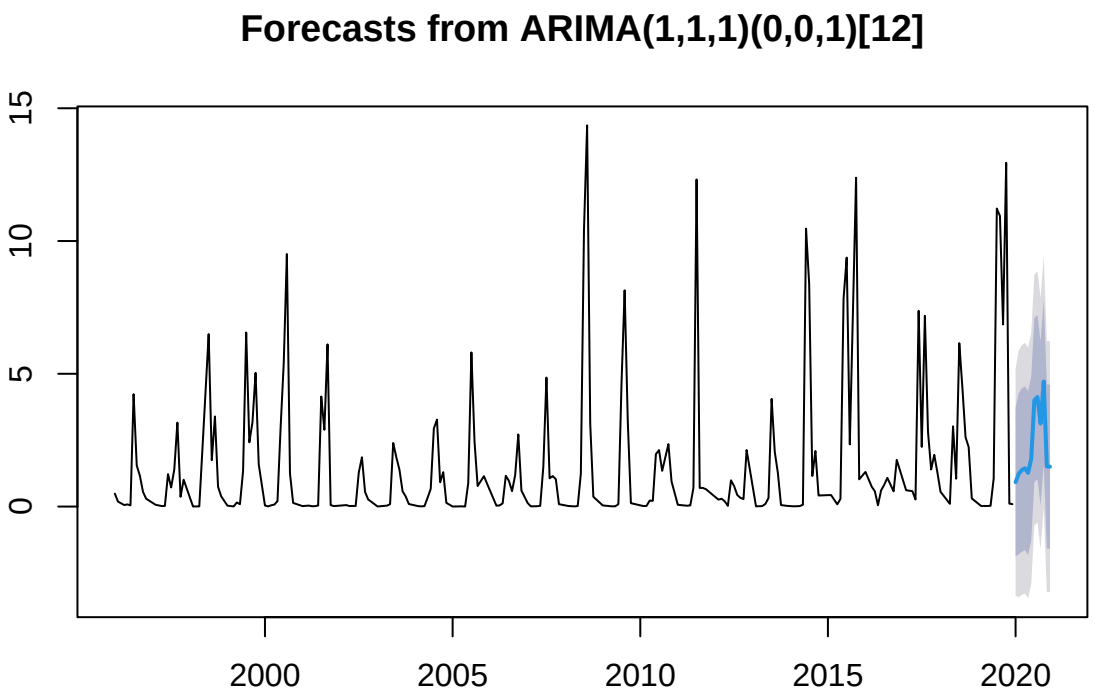



```
# Model 3: SARIMA

SARIMA_autofit <- auto.arima(ts_biomass)
checkresiduals(SARIMA_autofit)
```

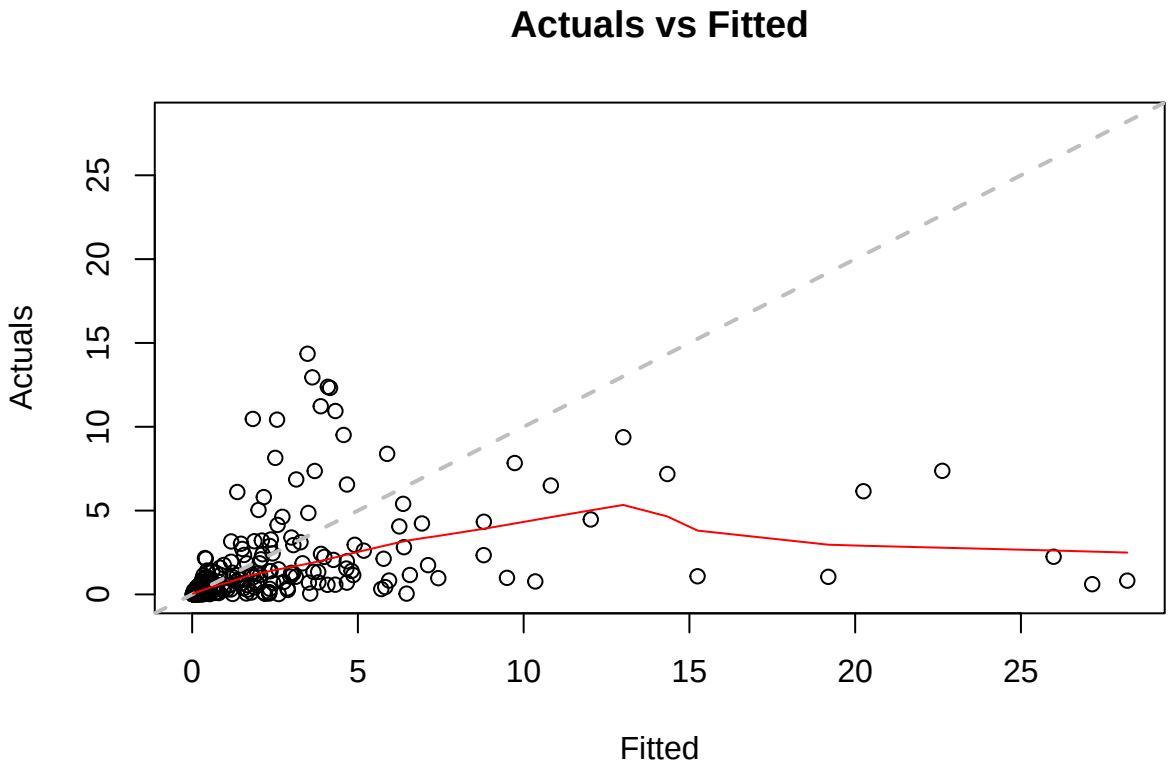


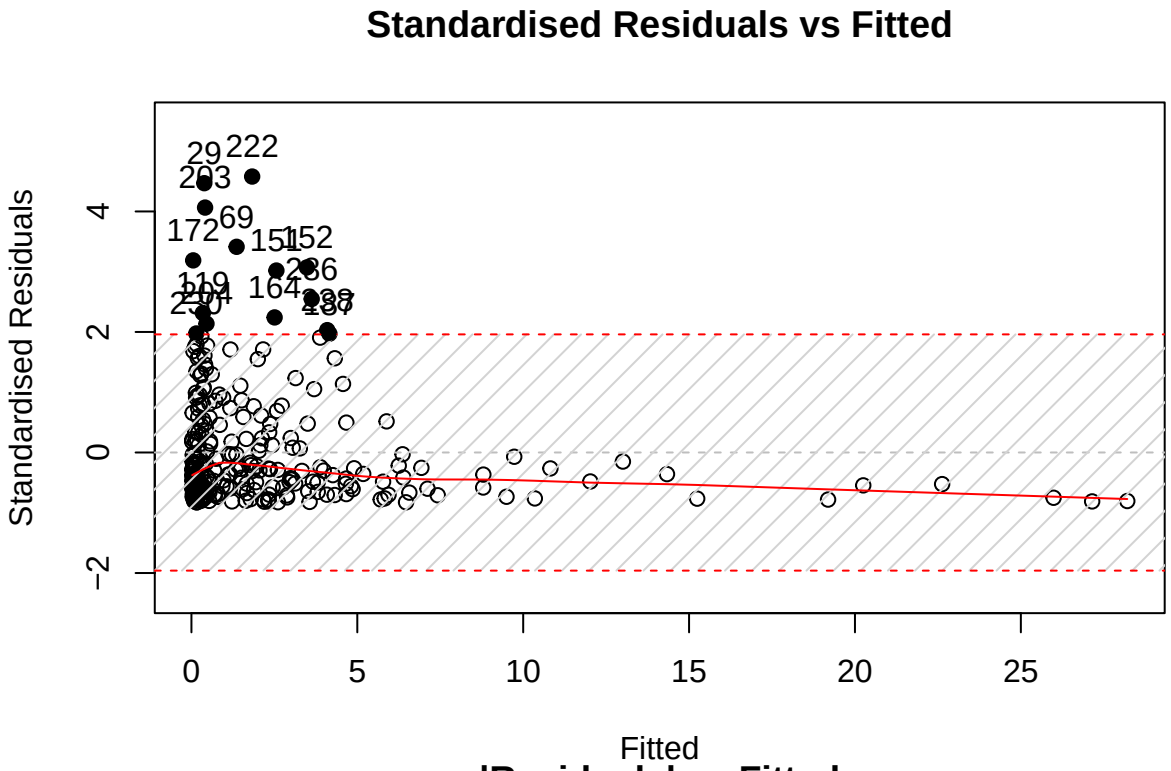

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,0,1)[12]
## Q* = 42.8, df = 21, p-value = 0.003333
##
## Model df: 3.   Total lags used: 24
```

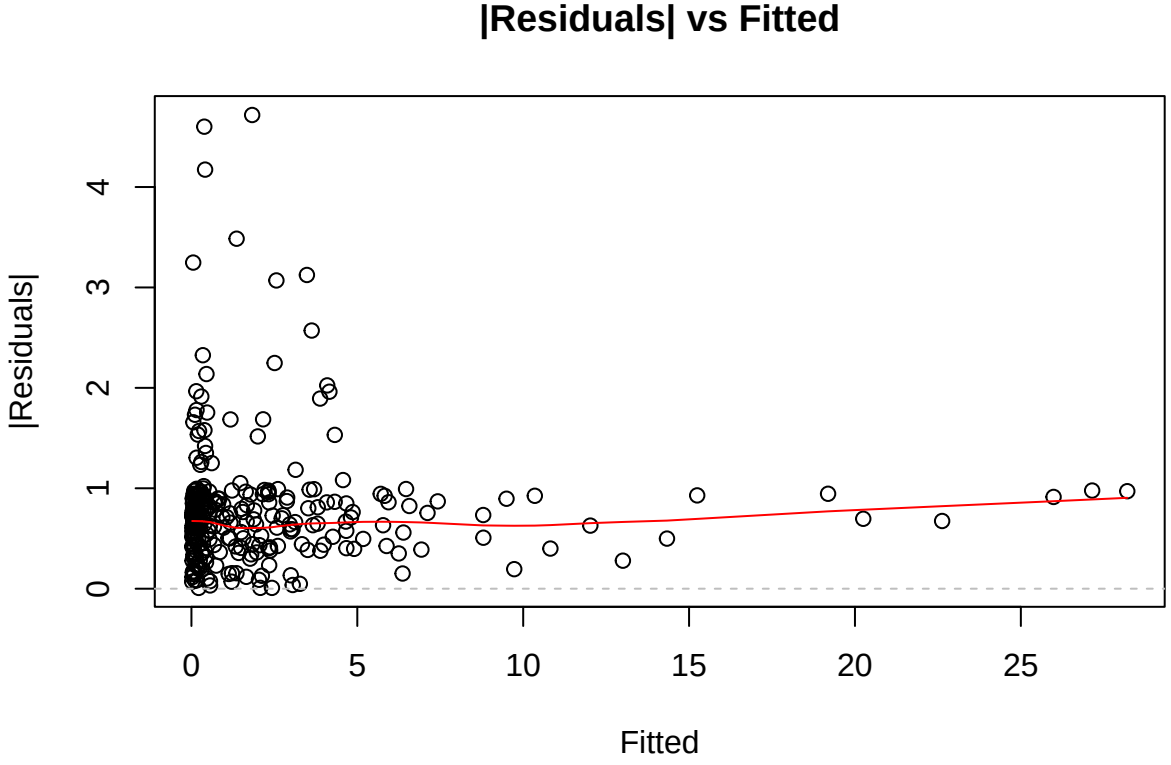
```
#Generating forecasts
#remember auto.arima does not call the forecast() internally so we need one more step
SARIMA_for <- forecast(SARIMA_autofit,h=12)
plot(SARIMA_for)
```

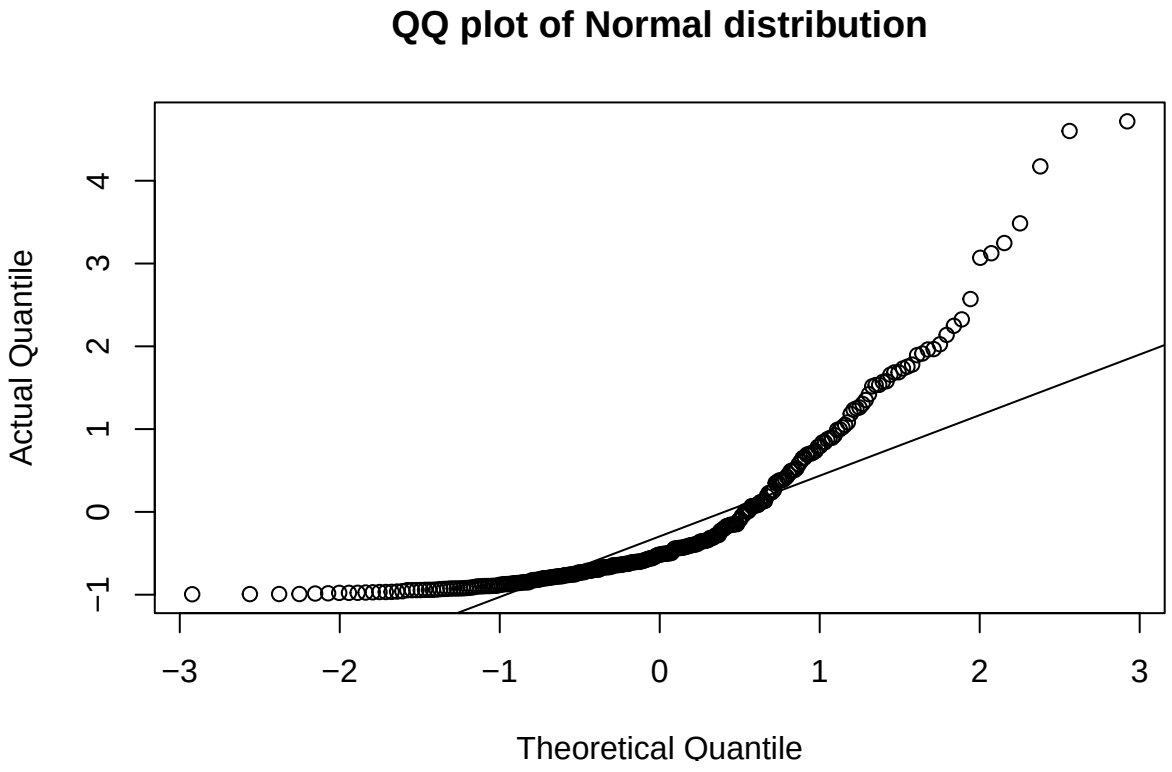



```
# Model 4: SS Exponential smoothing
SSES_seas <- es(ts_biomass,model="ZZZ",h=12,holdout=FALSE)
plot(SSES_seas)
```



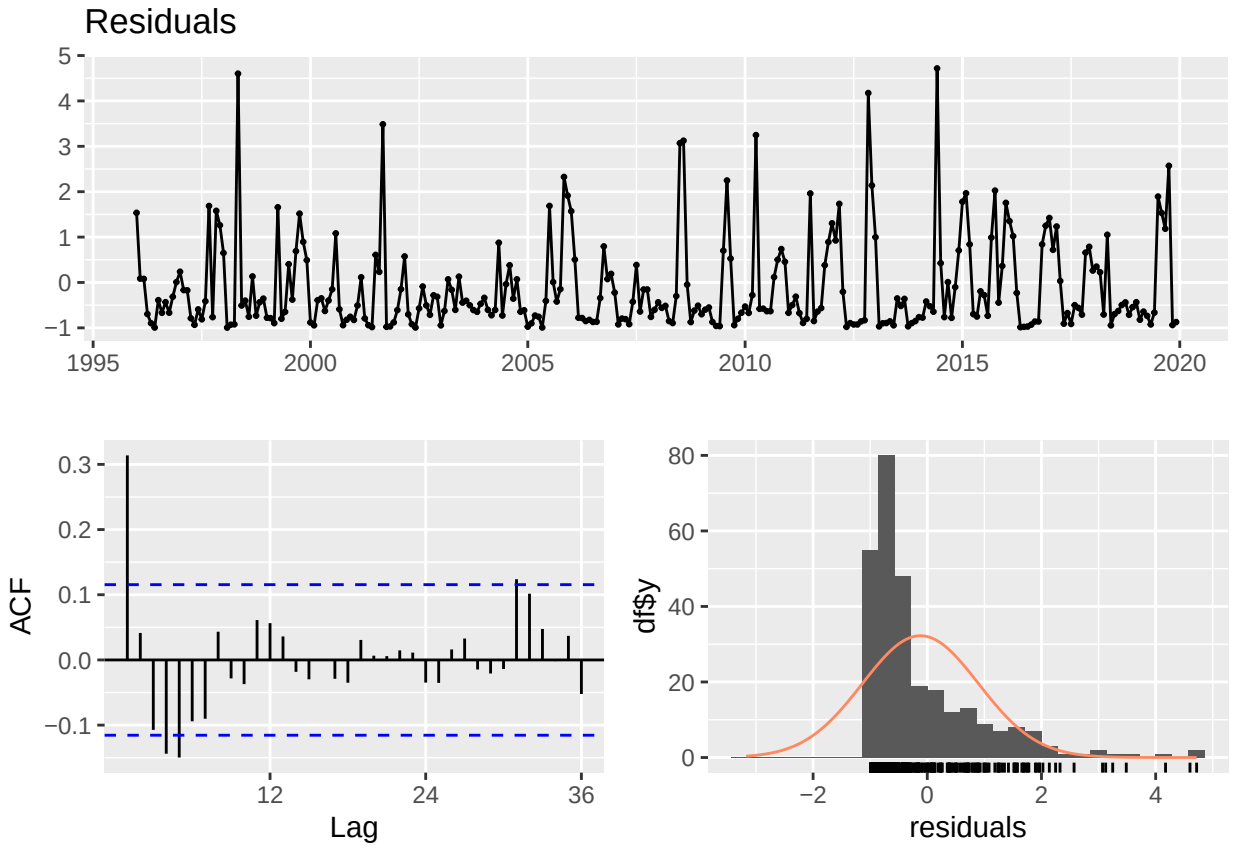

Standardised Residuals vs Fitted
Standardised Residuals
Fitted
29
222
203
69
172
151
152
286
119
164
−2
0
2
4
0
5
10
15
20
25


|Residuals| vs Fitted
|Residuals|
Fitted
0
1
2
3
4
0
5
10
15
20
25



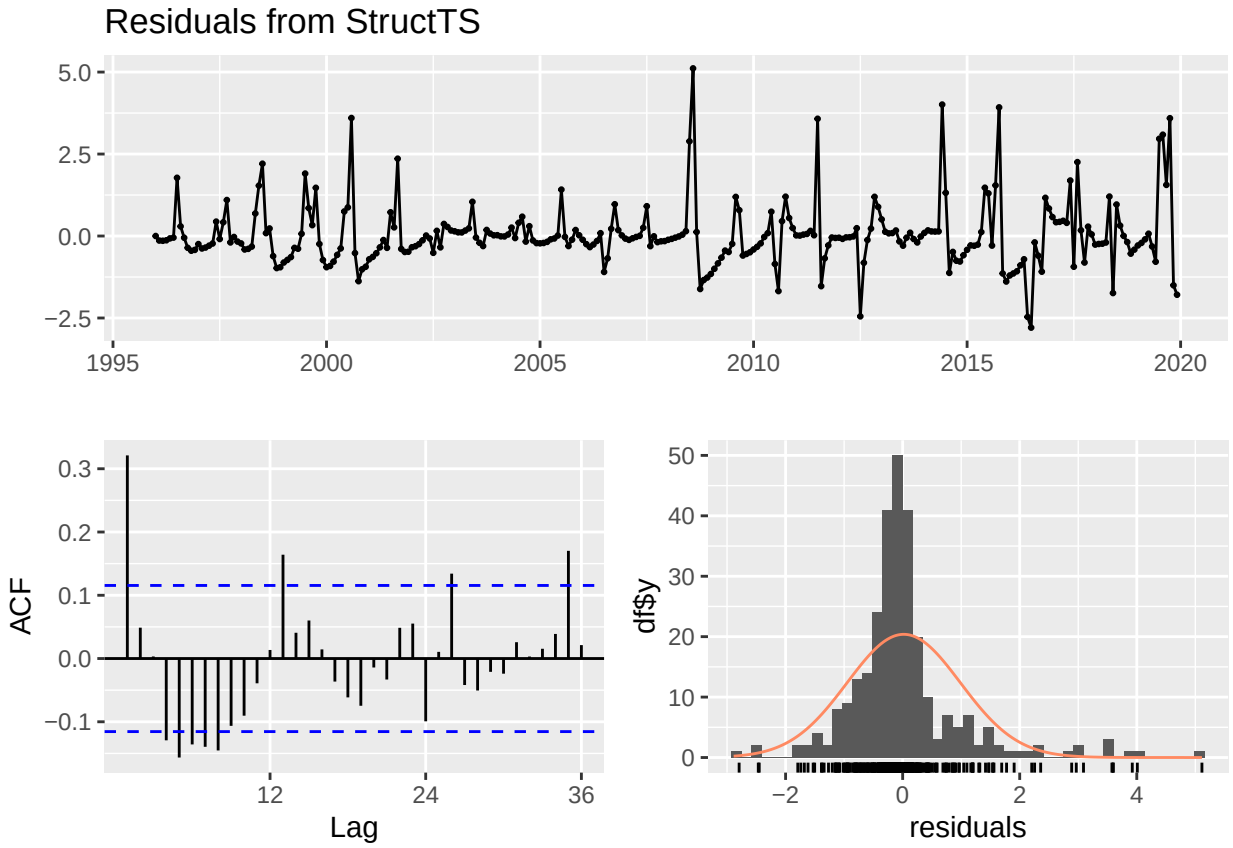

```
checkresiduals(SSES_seas)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```

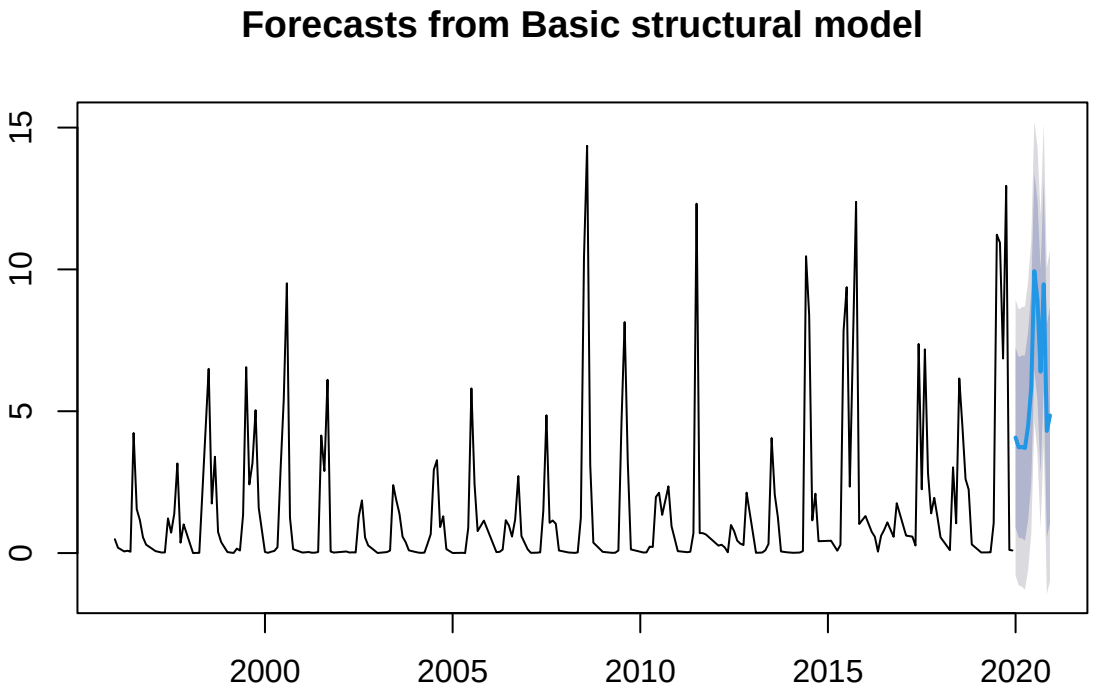



```
# Model 5: SS with StructTS()

SS_seas <- StructTS(ts_biomass,
                    type="BSM",fixed=c(0,0.001,0.3,NA))   #this function has convergence issues
checkresiduals(SS_seas)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
#Generating forecasts
# StructTS() does not call the forecast() internally so we need one more step
SS_for <- forecast(SS_seas,h=12)
plot(SS_for)
```




```
#Model 1: Arithmetic mean
MEAN_scores <- accuracy(MEAN_seas$mean,last_obs)  #store the performance metrics

#Model 2: Seasonal naive
SNAIVE_scores <- accuracy(SNAIVE_seas$mean,last_obs)

# Model 3:  SARIMA
SARIMA_scores <- accuracy(SARIMA_for$mean,last_obs)

# Model 4:  SSES
SSES_scores <- accuracy(SSES_seas$forecast,last_obs)

# Model 5:  BSM
SS_scores <- accuracy(SS_for$mean,last_obs)


#create data frame
seas_scores <- as.data.frame(rbind(MEAN_scores, SNAIVE_scores, SARIMA_scores,SSES_scores,SS_scores))
row.names(seas_scores) <- c("MEAN", "SNAIVE","SARIMA","SSES","BSM")

#choose model with lowest RMSE
best_model_index <- which.min(seas_scores[,"RMSE"])
cat("The best model by RMSE is:", row.names(seas_scores[best_model_index,]))
```

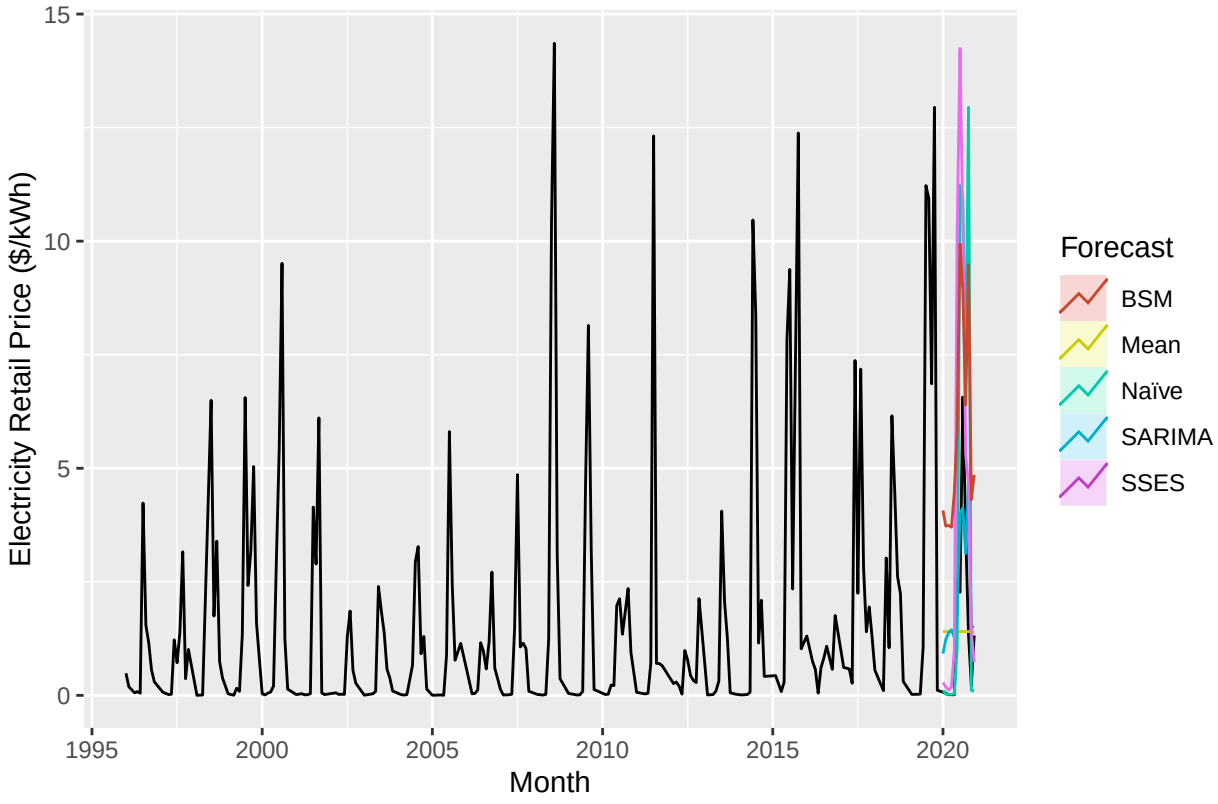
```
## The best model by RMSE is: SARIMA
```

Table 5: Forecast Accuracy for Seasonal Data

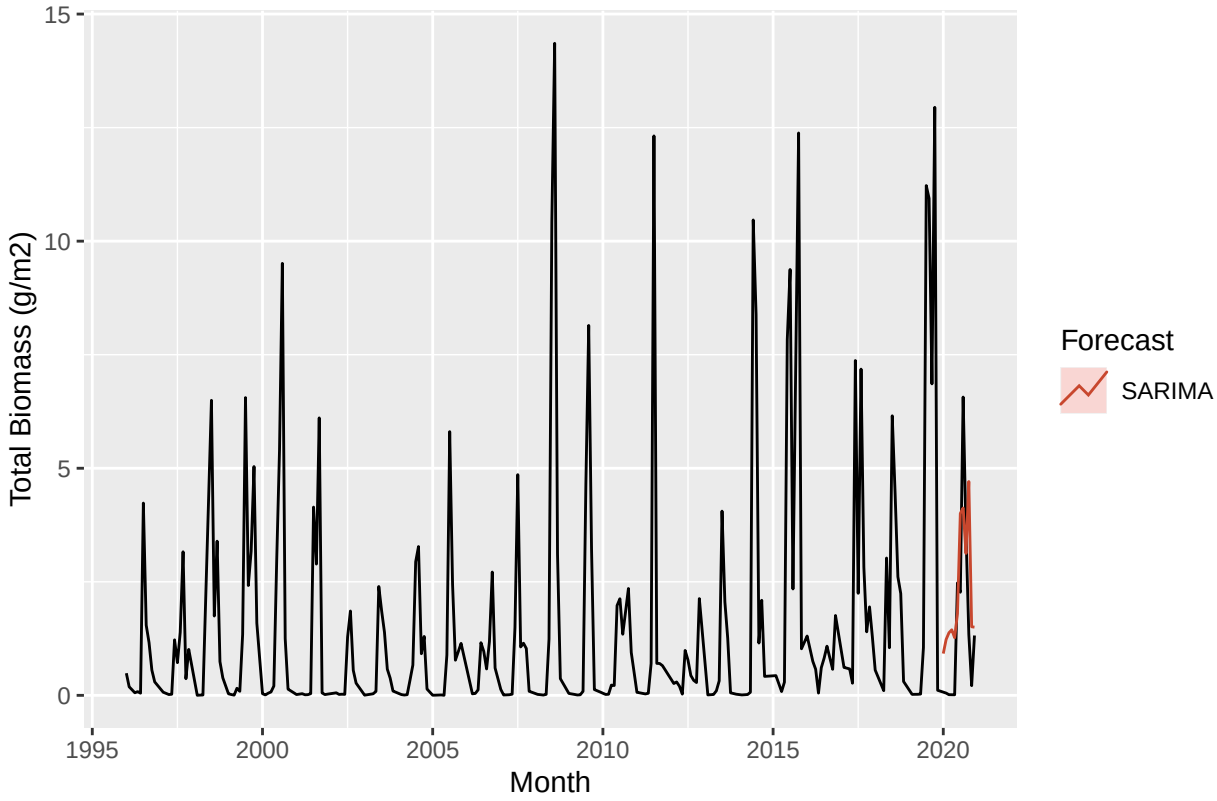
	ME	RMSE	MAE	MPE	MAPE
MEAN	0.09077	1.91972	1.45817	-2993.9993	3030.8195
SNAIVE	-2.12360	4.54636	2.58677	-122.4177	164.9659
SARIMA	-0.75532	1.58841	1.35418	-2867.0051	2879.9488
SSES	-2.55079	4.40694	2.64893	-998.5011	1005.9502
BSM	-4.29032	4.61610	4.29032	-8743.8441	8743.8441

```
kbl(seas_scores,
      caption = "Forecast Accuracy for Seasonal Data",
      digits = array(5,ncol(seas_scores))) %>%
  kable_styling(full_width = FALSE, position = "center") %>%
  #highlight model with lowest RMSE
  kable_styling(latex_options="striped", stripe_index = which.min(seas_scores[,"RMSE"]))
```

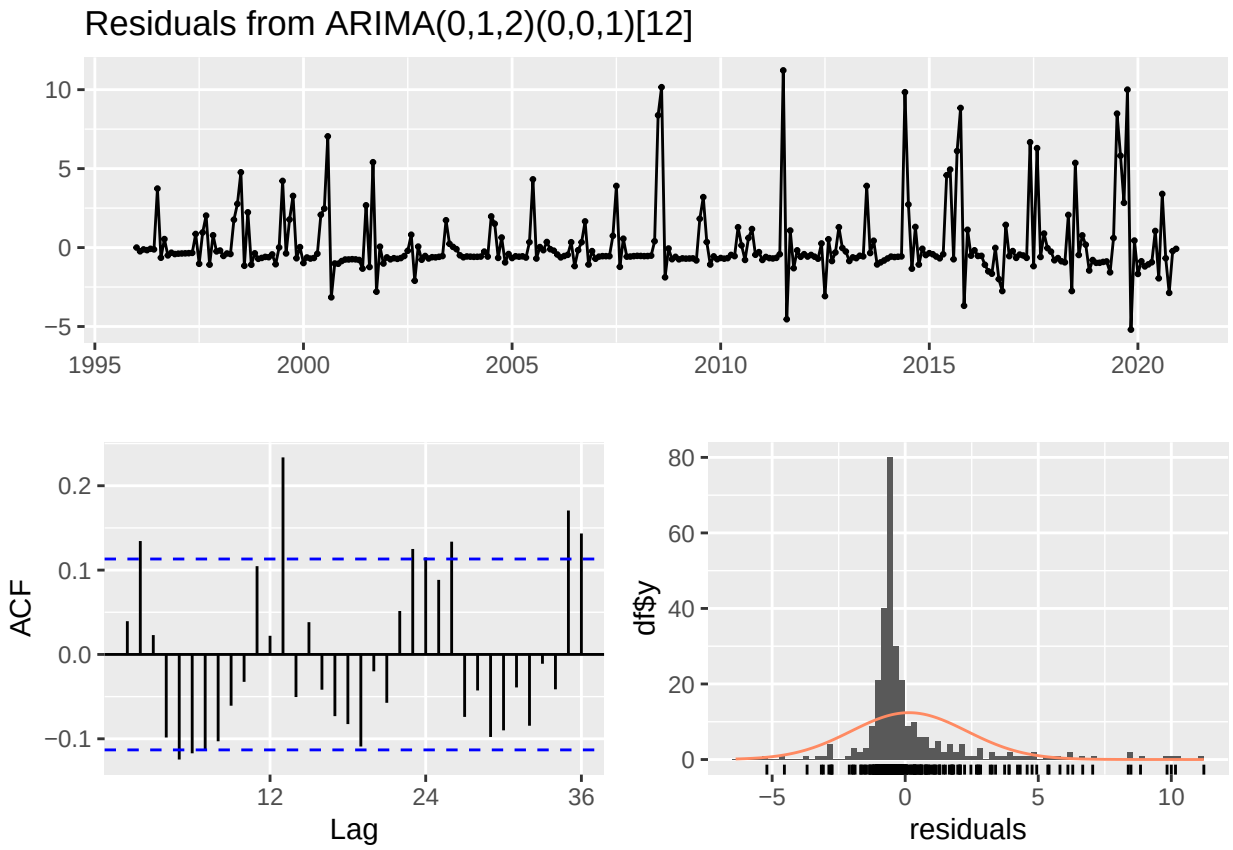
```
autoplot(ts_biomass_data) +
  autolayer(MEAN_seas, PI=FALSE, series="Mean") +
  autolayer(SNAIVE_seas, PI=FALSE, series="Naïve") +
  autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  autolayer(SSES_seas$forecast, series="SSES") +
  autolayer(SS_for,PI=FALSE,series="BSM") +
  xlab("Month") + ylab("Electricity Retail Price ($/kWh)") +
  guides(colour=guide_legend(title="Forecast"))
```




```
autoplot(ts_biomass_data) +

autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  xlab("Month") + ylab("Total Biomass (g/m2)") +
  guides(colour=guide_legend(title="Forecast"))
```

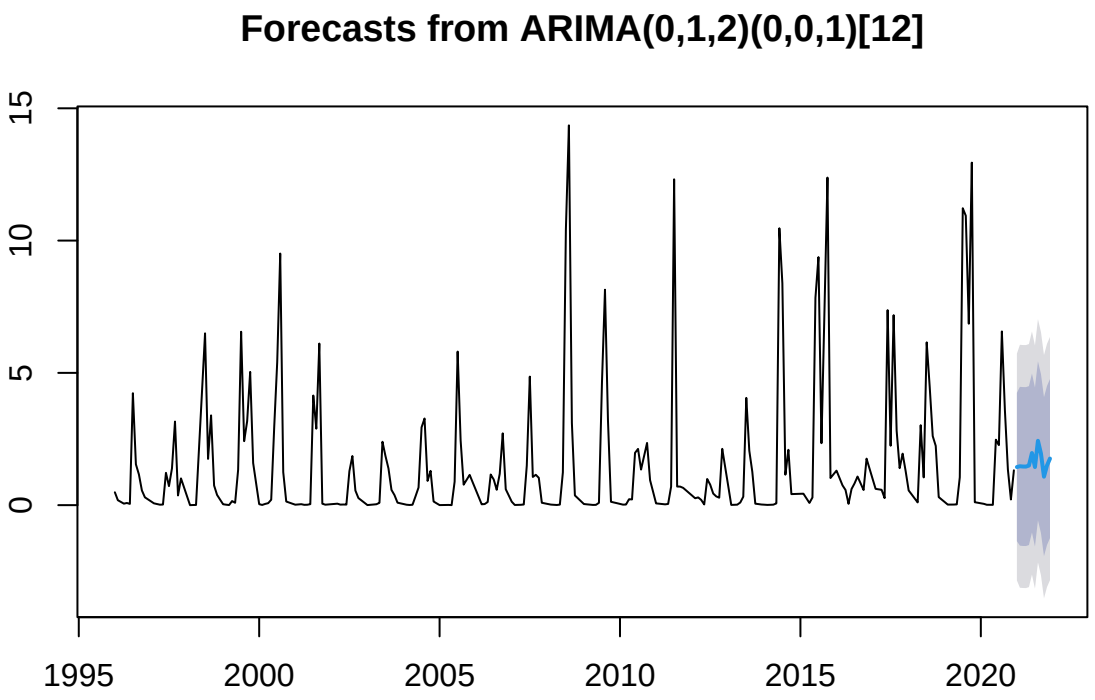



```
# Forecast

SARIMA_autofit_new <- auto.arima(ts_biomass_data)
checkresiduals(SARIMA_autofit_new)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)(0,0,1)[12]
## Q* = 68.766, df = 21, p-value = 5.534e-07
##
## Model df: 3.   Total lags used: 24
```

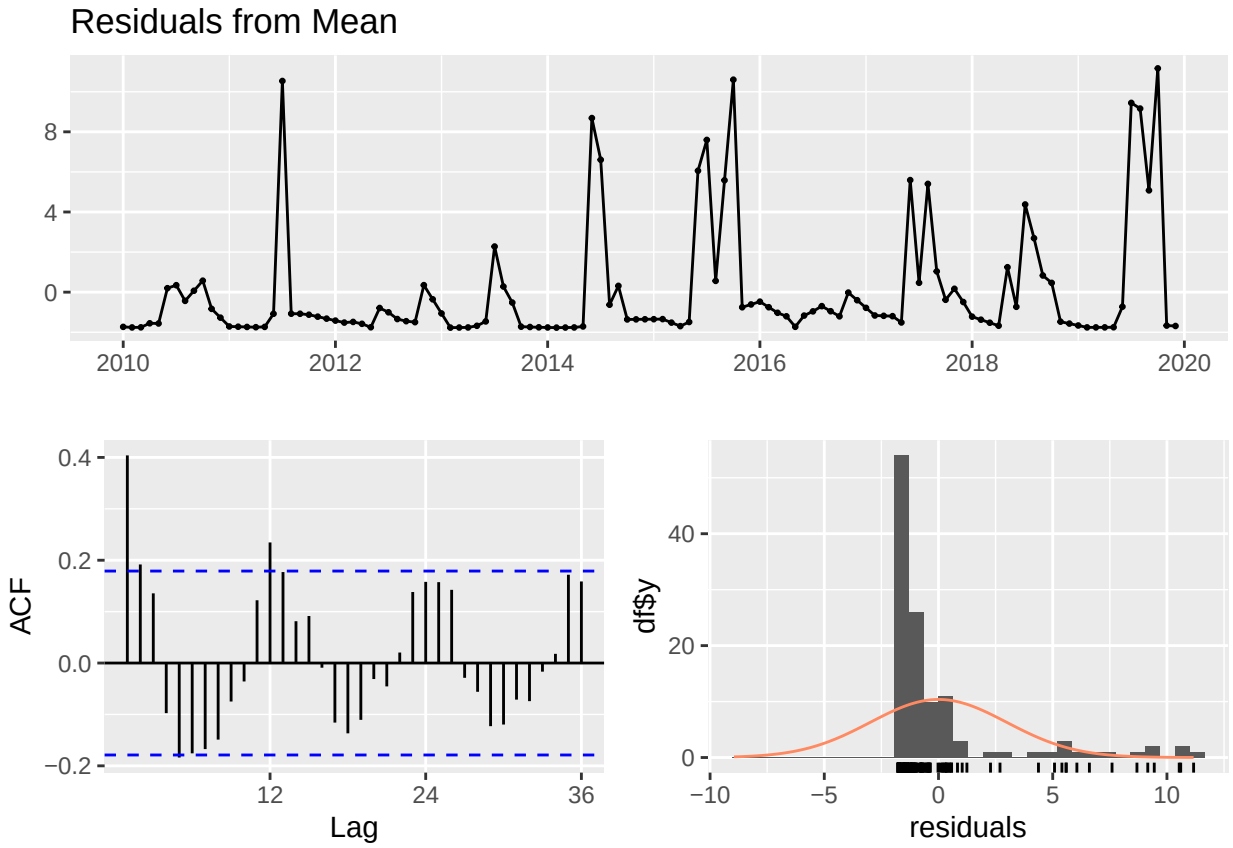
```
SARIMA_for_new <- forecast(SARIMA_autofit_new,h=12)
plot(SARIMA_for_new)
```



Use recent ten-year data (2010-2020) to forecast

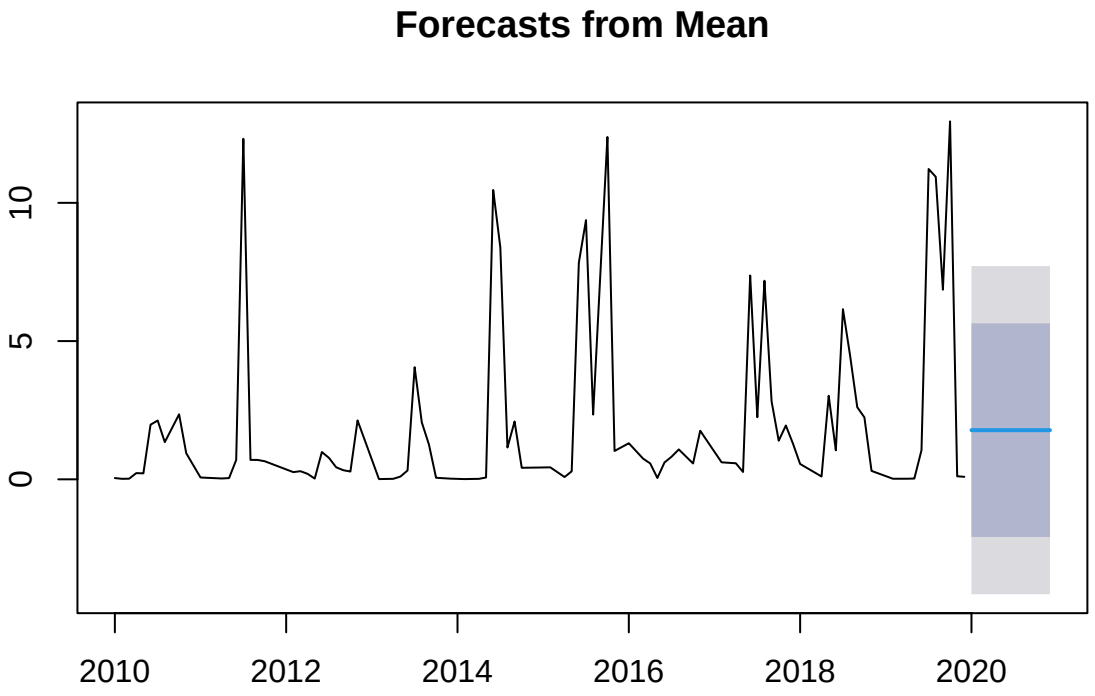
```
# Change the time span

# Transform to time series format

ts_biomass_data <- ts(
  biomass_data_frame[169:300,2],
  start=c(year(biomass_data_frame$Month[169]),month(biomass_data_frame$Month[169])),
  frequency=12)

ts_biomass <- ts(
  biomass_data_frame[169:288,2],
  start=c(year(biomass_data_frame$Month[169]),month(biomass_data_frame$Month[169])),
  frequency=12)


last_obs <- ts_biomass_data[121:132]
```

```
# Model 1: Arithmetic mean
# The meanf() has no holdout option
MEAN_seas <- meanf(y = ts_biomass, h = 12)
checkresiduals(MEAN_seas)
```

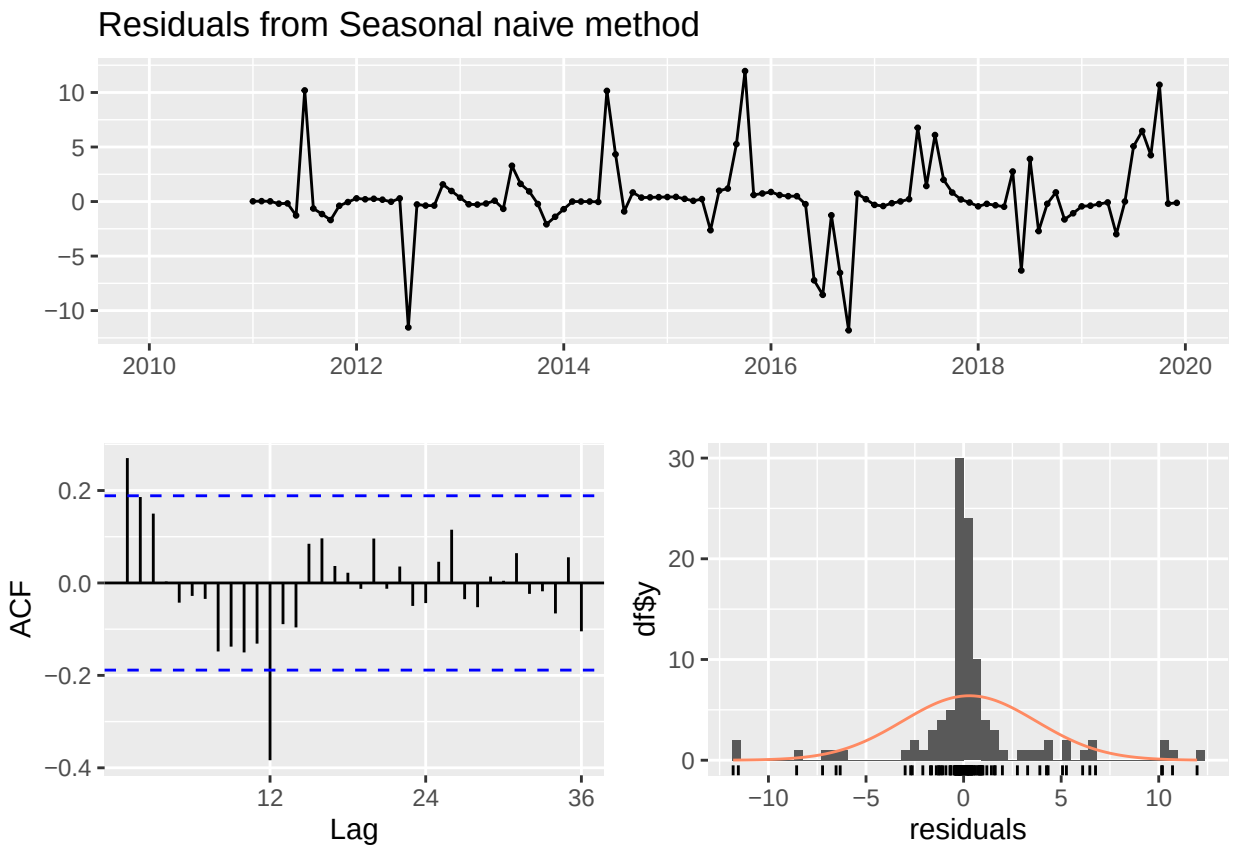



```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 73.247, df = 23, p-value = 3.795e-07
##
## Model df: 1.   Total lags used: 24
```

```
plot(MEAN_seas)
```

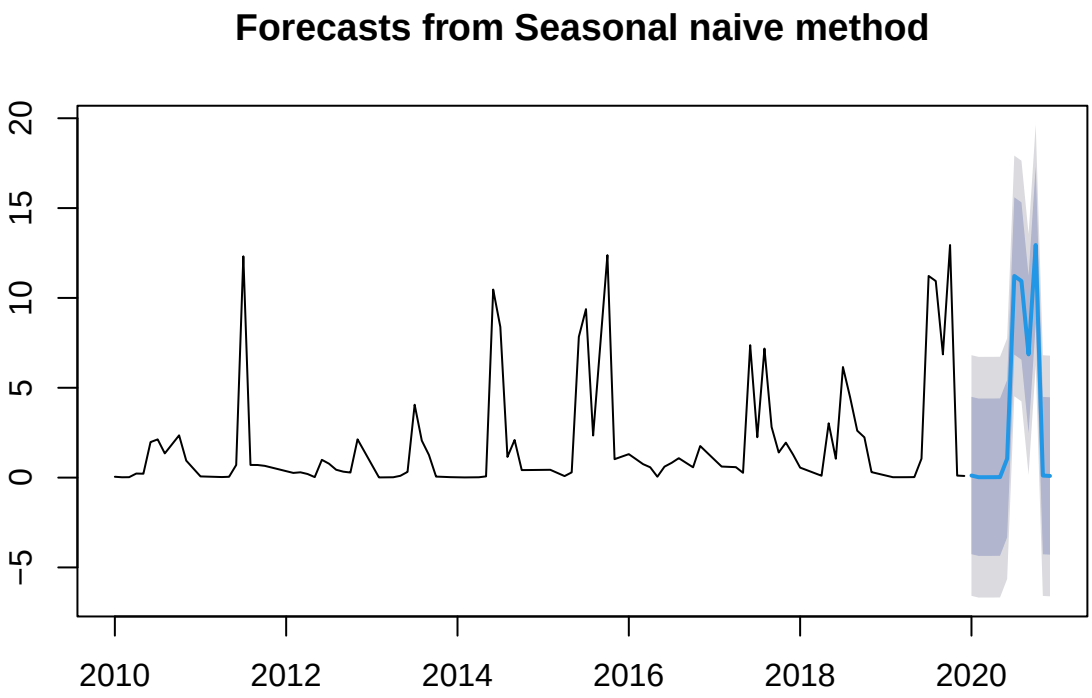



```
# Model 2: Seasonal naive
SNAIVE_seas <- snaive(ts_biomass, h=12, holdout=FALSE)
checkresiduals(SNAIVE_seas)
```

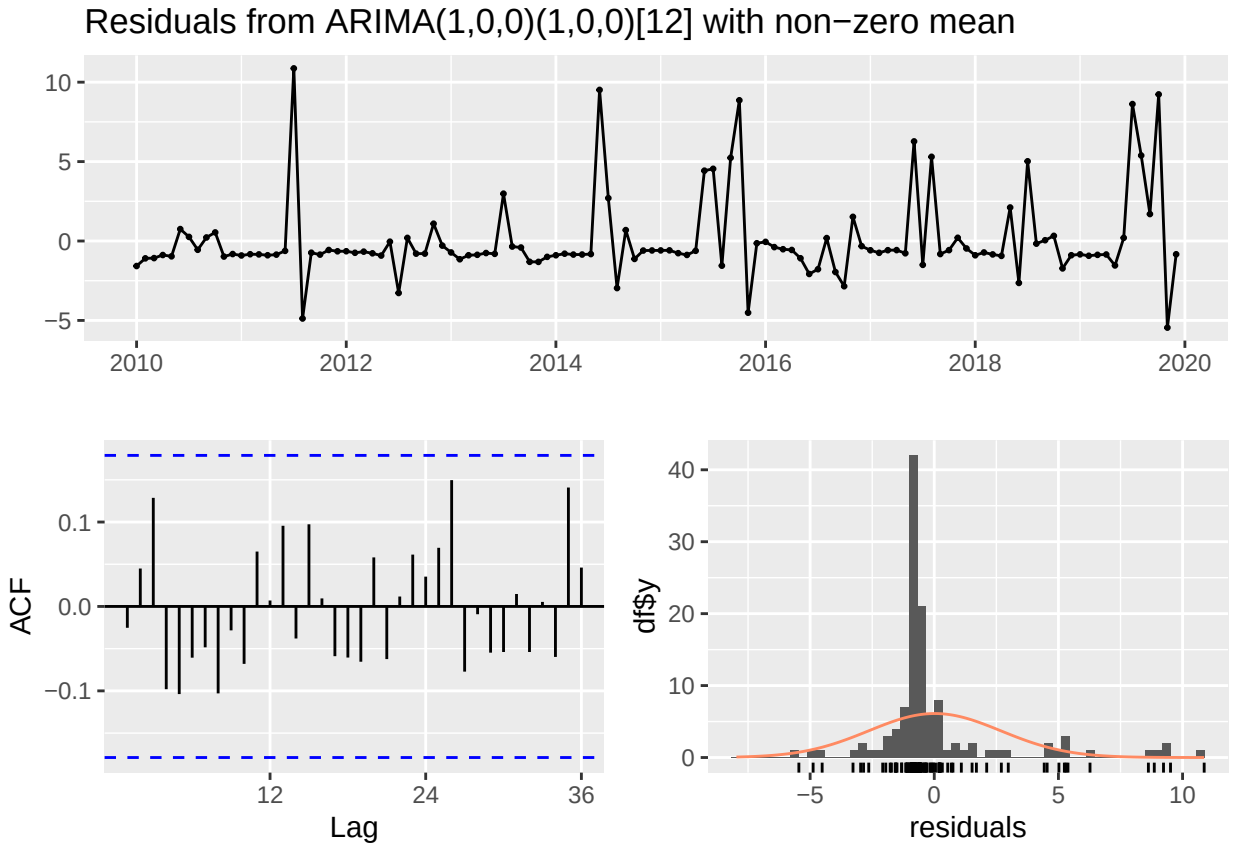


```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 49.511, df = 24, p-value = 0.001633
##
## Model df: 0.   Total lags used: 24
```

```
plot(SNAIVE_seas)
```

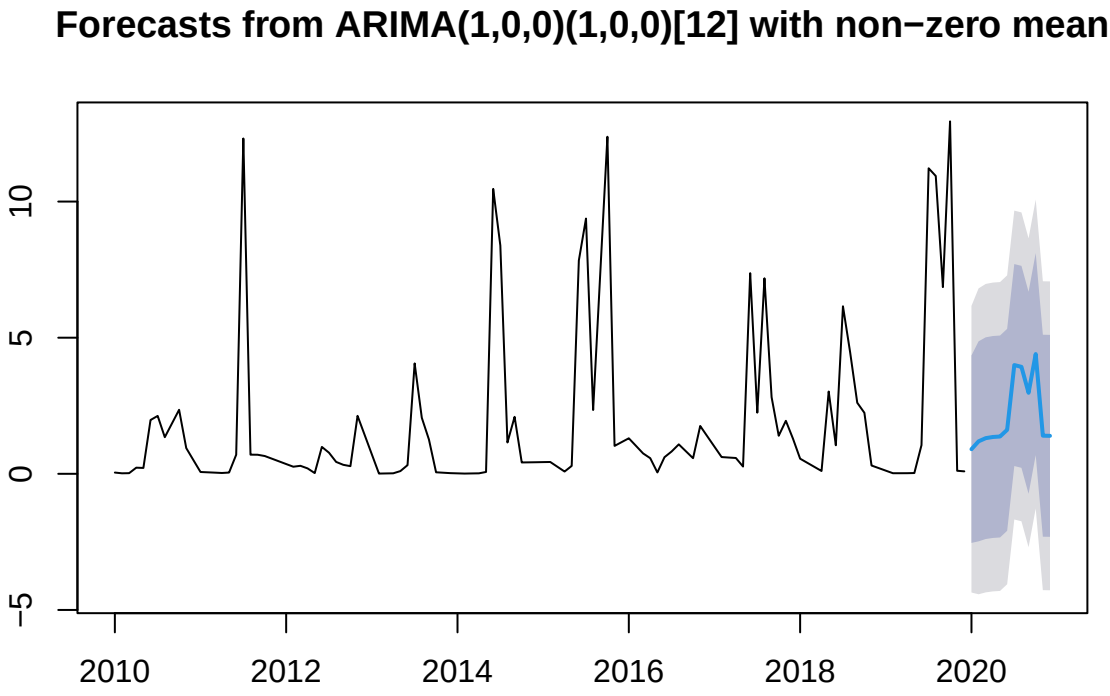



```
# Model 3: SARIMA

SARIMA_autofit <- auto.arima(ts_biomass)
checkresiduals(SARIMA_autofit)
```

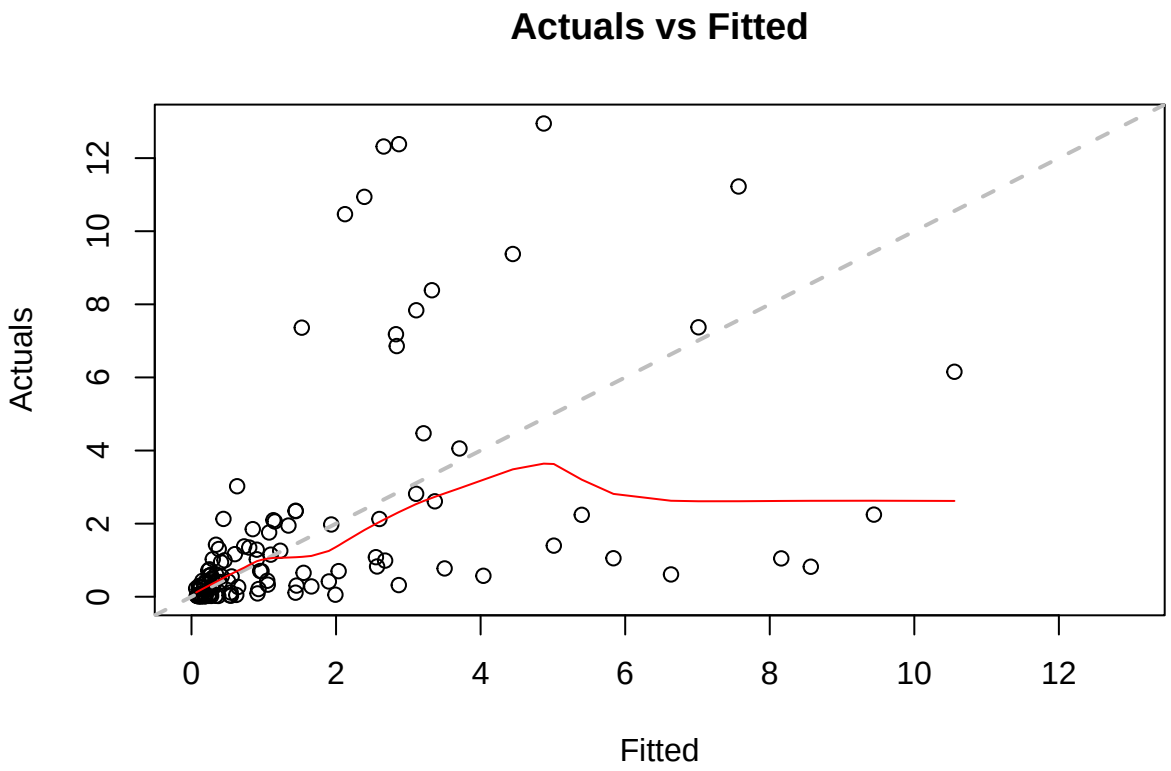


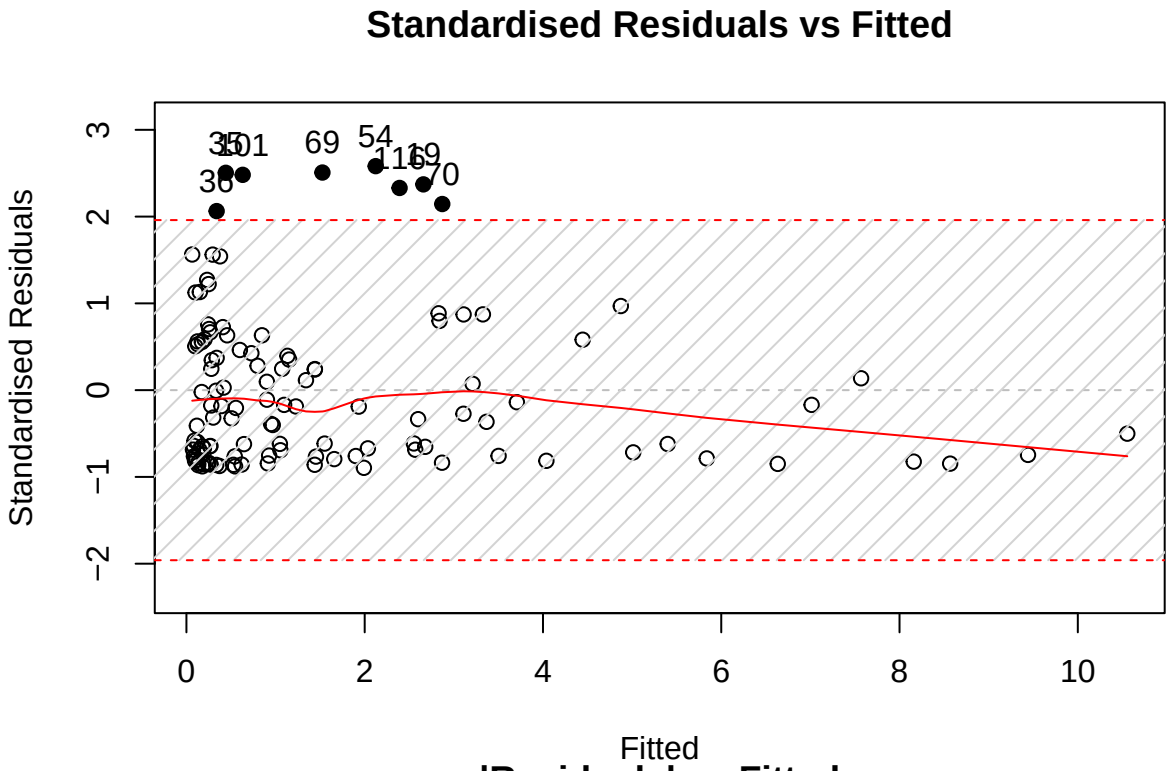

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,0,0)[12] with non-zero mean
## Q* = 14.711, df = 22, p-value = 0.8743
##
## Model df: 2.   Total lags used: 24
```

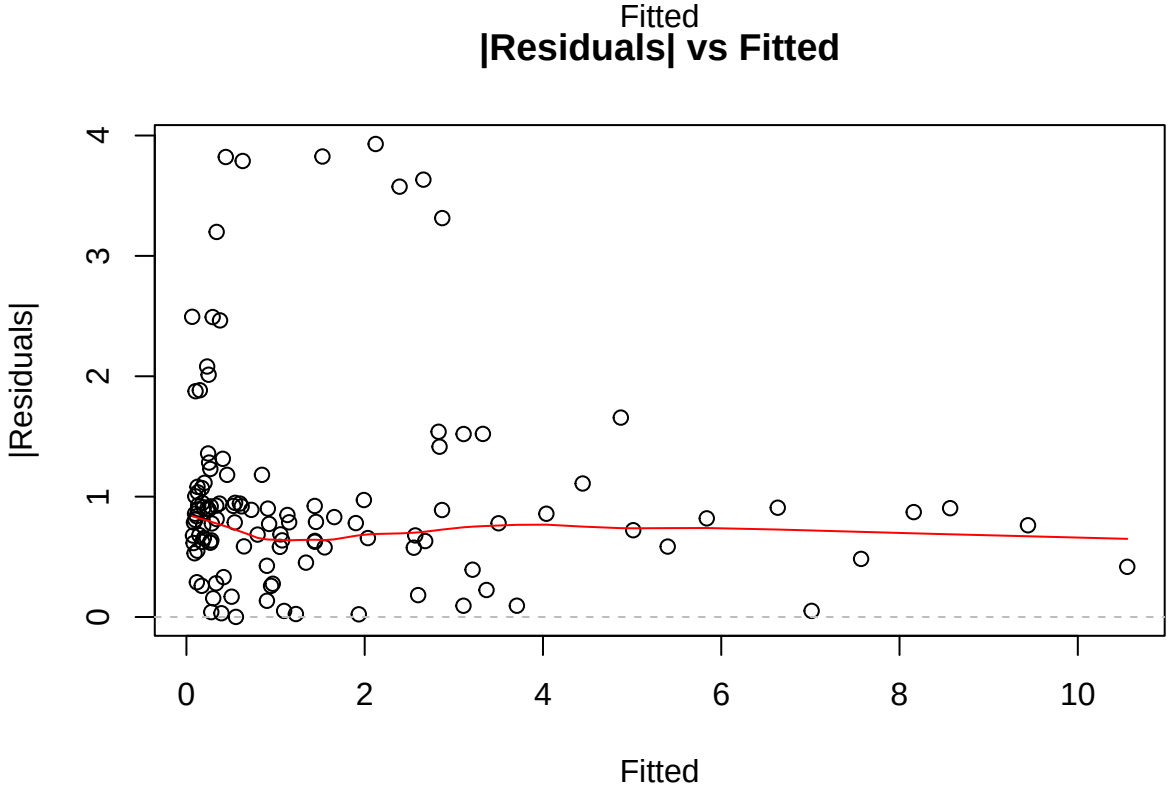
```
#Generating forecasts
#remember auto.arima does not call the forecast() internally so we need one more step
SARIMA_for <- forecast(SARIMA_autofit,h=12)
plot(SARIMA_for)
```

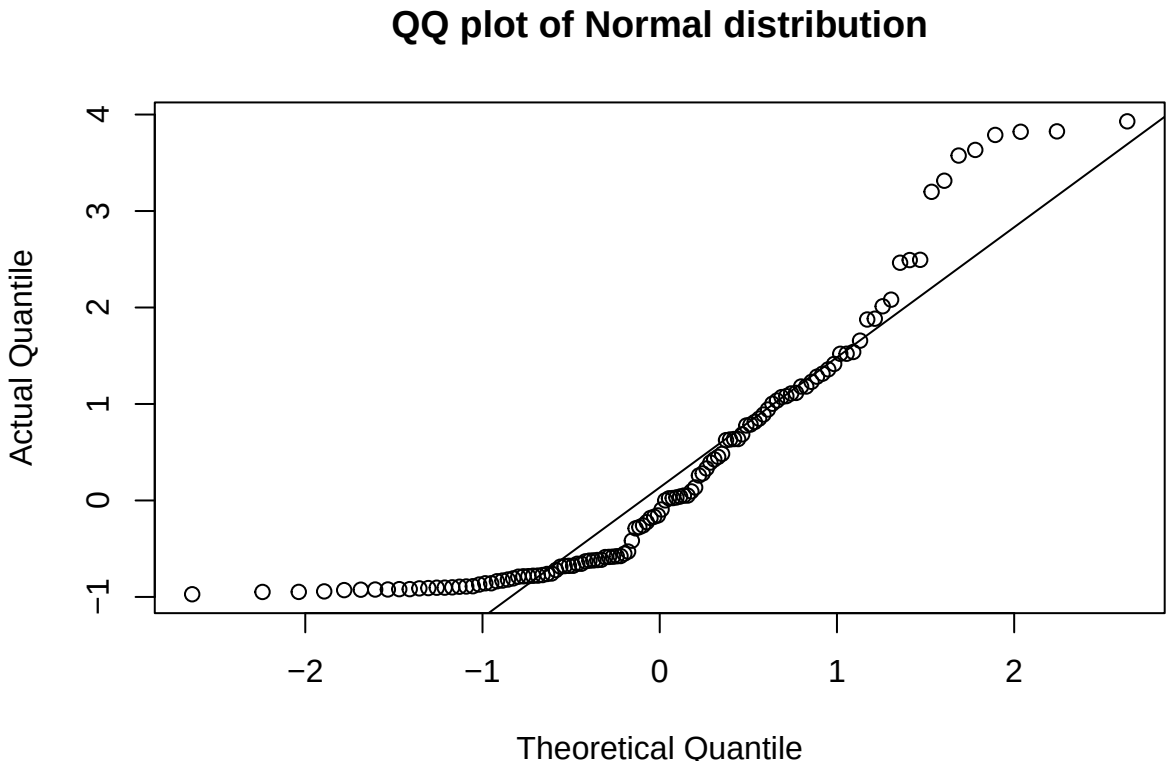



```
# Model 4: SS Exponential smoothing
SSES_seas <- es(ts_biomass,model="ZZZ",h=12,holdout=FALSE)
plot(SSES_seas)
```



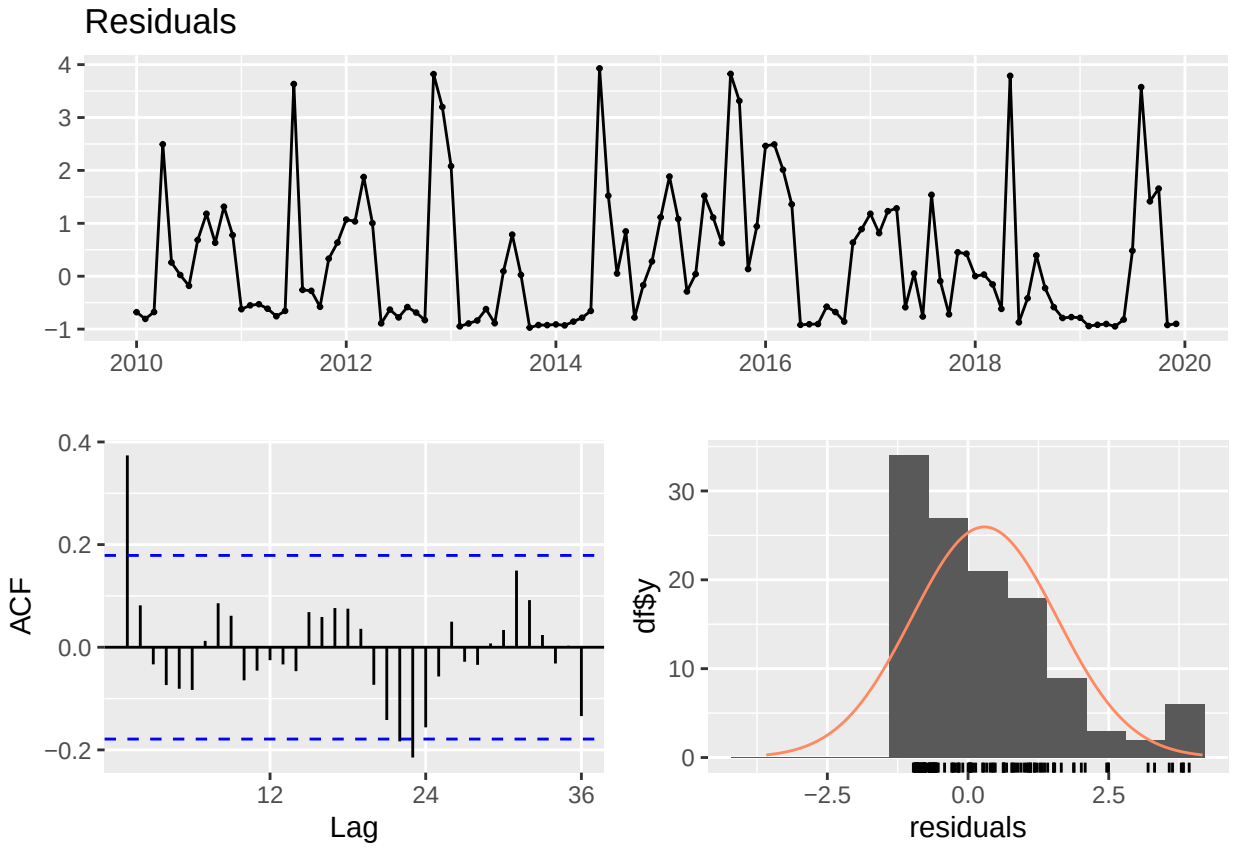

Standardised Residuals vs Fitted
Standardised Residuals
Fitted
35
101
69
54
116
69
70
36
3
2
1
0
−1
−2
0
2
4
6
8
10


|Residuals| vs Fitted
|Residuals|
Fitted
4
3
2
1
0
0
2
4
6
8
10



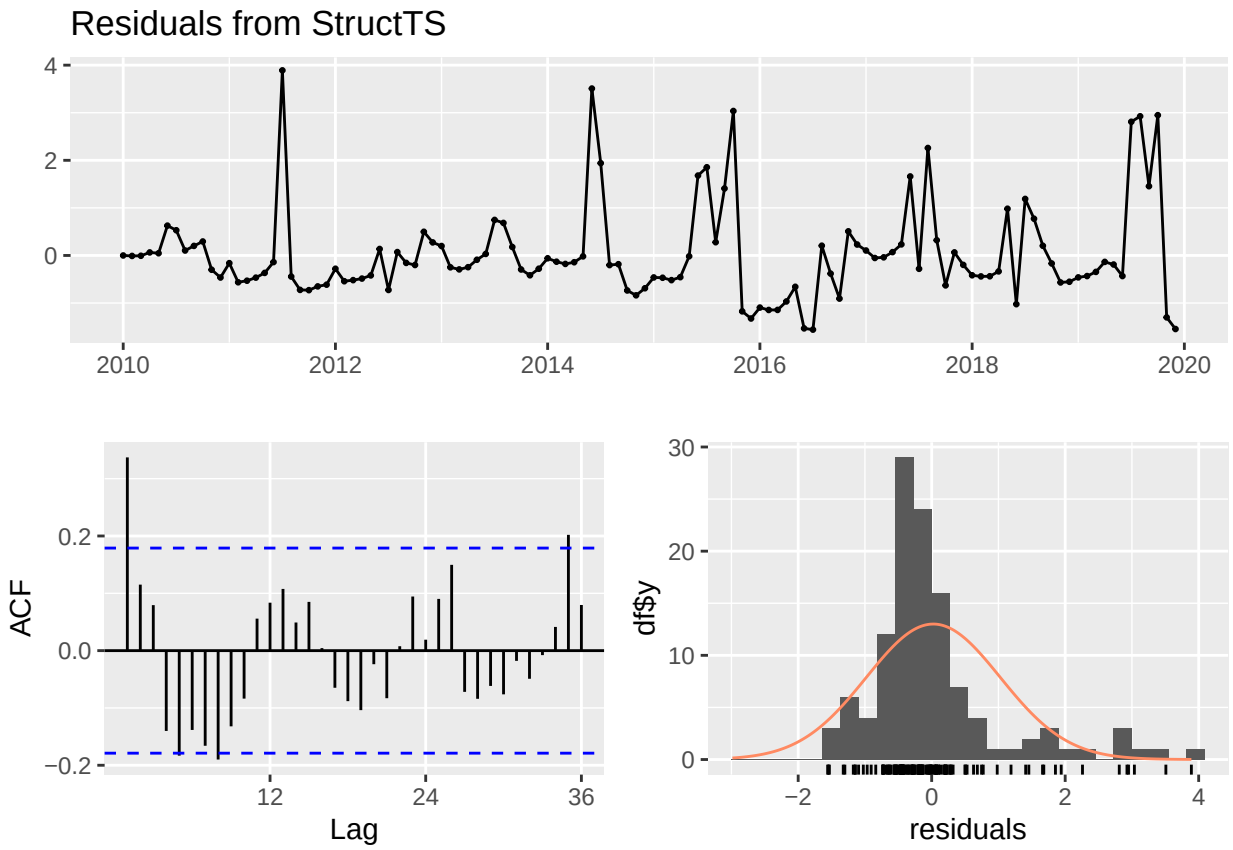

```
checkresiduals(SSES_seas)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```

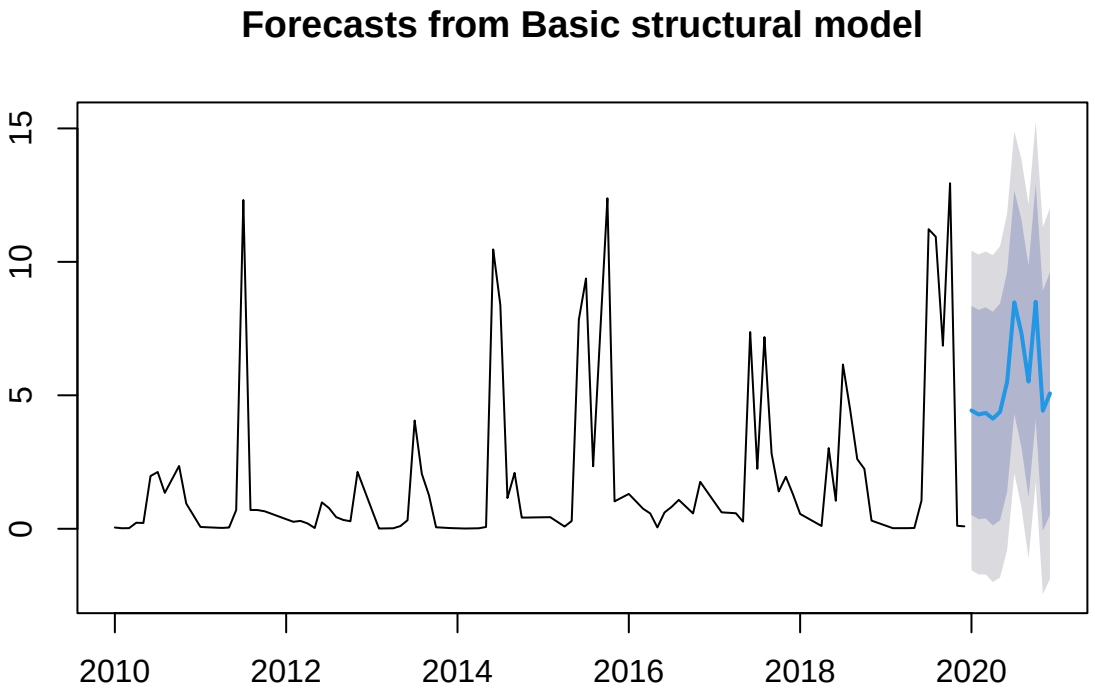



```
# Model 5: SS with StructTS()

SS_seas <- StructTS(ts_biomass,
                    type="BSM",fixed=c(0,0.001,0.3,NA))   #this function has convergence issues
checkresiduals(SS_seas)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
#Generating forecasts
# StructTS() does not call the forecast() internally so we need one more step
SS_for <- forecast(SS_seas,h=12)
plot(SS_for)
```




```
#Model 1: Arithmetic mean
MEAN_scores <- accuracy(MEAN_seas$mean,last_obs)  #store the performance metrics

#Model 2: Seasonal naive
SNAIVE_scores <- accuracy(SNAIVE_seas$mean,last_obs)

# Model 3:  SARIMA
SARIMA_scores <- accuracy(SARIMA_for$mean,last_obs)

# Model 4:  SSES
SSES_scores <- accuracy(SSES_seas$forecast,last_obs)

# Model 5:  BSM
SS_scores <- accuracy(SS_for$mean,last_obs)


#create data frame
seas_scores <- as.data.frame(rbind(MEAN_scores, SNAIVE_scores, SARIMA_scores,SSES_scores,SS_scores))
row.names(seas_scores) <- c("MEAN", "SNAIVE","SARIMA","SSES","BSM")

#choose model with lowest RMSE
best_model_index <- which.min(seas_scores[,"RMSE"])
cat("The best model by RMSE is:", row.names(seas_scores[best_model_index,]))
```

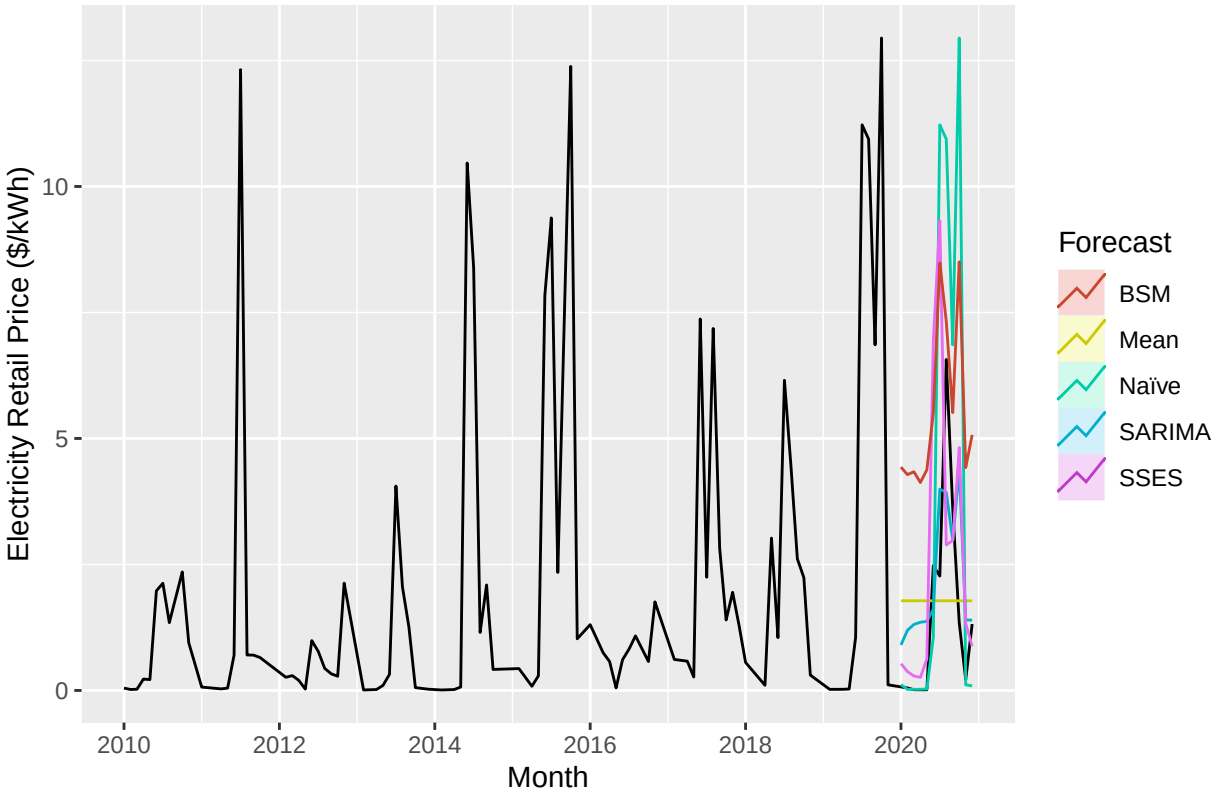
```
## The best model by RMSE is: SARIMA
```

Table 6: Forecast Accuracy for Seasonal Data

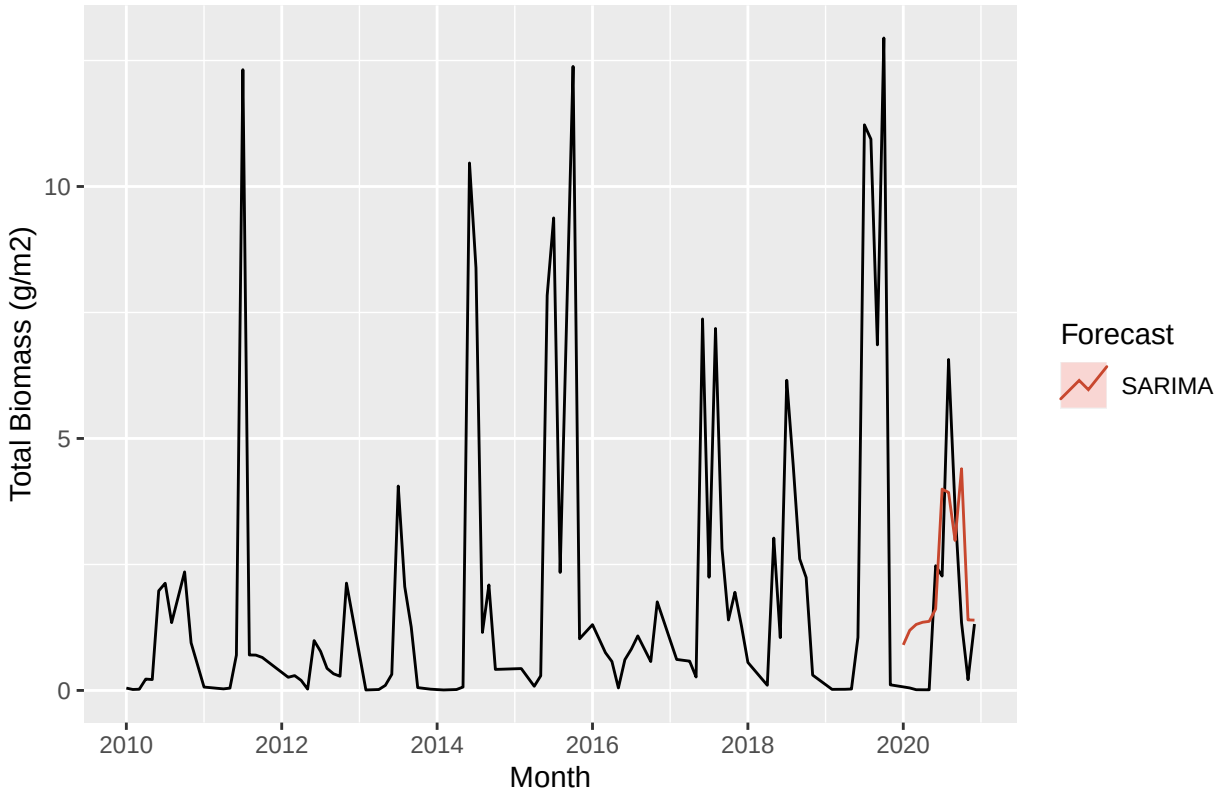
	ME	RMSE	MAE	MPE	MAPE
MEAN	-0.28323	1.93838	1.58284	-3817.5199	3846.3960
SNAIVE	-2.12360	4.54636	2.58677	-122.4177	164.9659
SARIMA	-0.65751	1.55783	1.34461	-2824.6163	2839.9915
SSES	-1.10662	2.85708	1.89745	-919.7008	937.4871
BSM	-4.03626	4.34690	4.03626	-9410.6202	9410.6202

```
kbl(seas_scores,
      caption = "Forecast Accuracy for Seasonal Data",
      digits = array(5,ncol(seas_scores))) %>%
  kable_styling(full_width = FALSE, position = "center") %>%
  #highlight model with lowest RMSE
  kable_styling(latex_options="striped", stripe_index = which.min(seas_scores[,"RMSE"]))
```

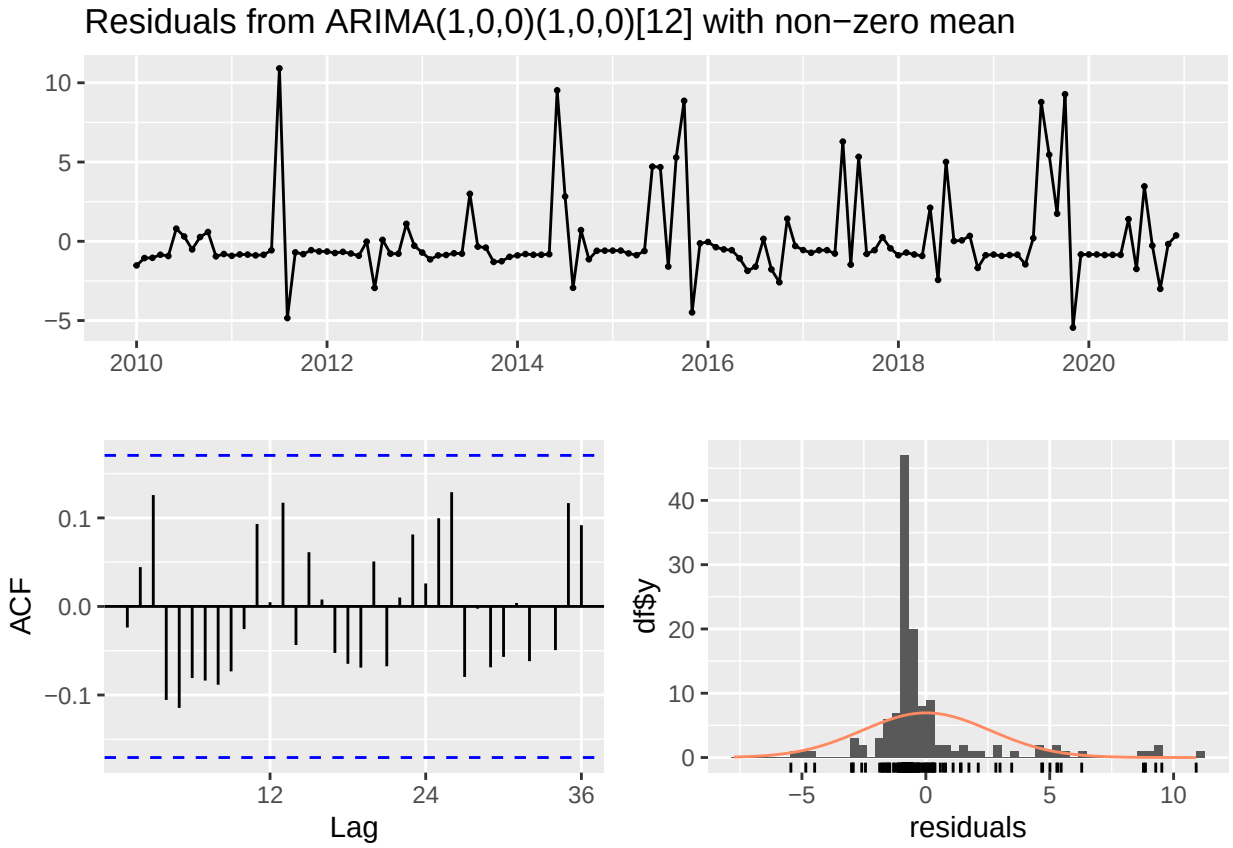
```
autoplot(ts_biomass_data) +
  autolayer(MEAN_seas, PI=FALSE, series="Mean") +
  autolayer(SNAIVE_seas, PI=FALSE, series="Naïve") +
  autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  autolayer(SSES_seas$forecast, series="SSES") +
  autolayer(SS_for,PI=FALSE,series="BSM") +
  xlab("Month") + ylab("Electricity Retail Price ($/kWh)") +
  guides(colour=guide_legend(title="Forecast"))
```




```
autoplot(ts_biomass_data) +

autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  xlab("Month") + ylab("Total Biomass (g/m2)") +
  guides(colour=guide_legend(title="Forecast"))
```




```
# Forecast

SARIMA_autofit_new <- auto.arima(ts_biomass_data)
checkresiduals(SARIMA_autofit_new)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,0,0)[12] with non-zero mean
## Q* = 18.099, df = 22, p-value = 0.7001
##
## Model df: 2.   Total lags used: 24
```

```
SARIMA_for_new <- forecast(SARIMA_autofit_new,h=12)
plot(SARIMA_for_new)
```

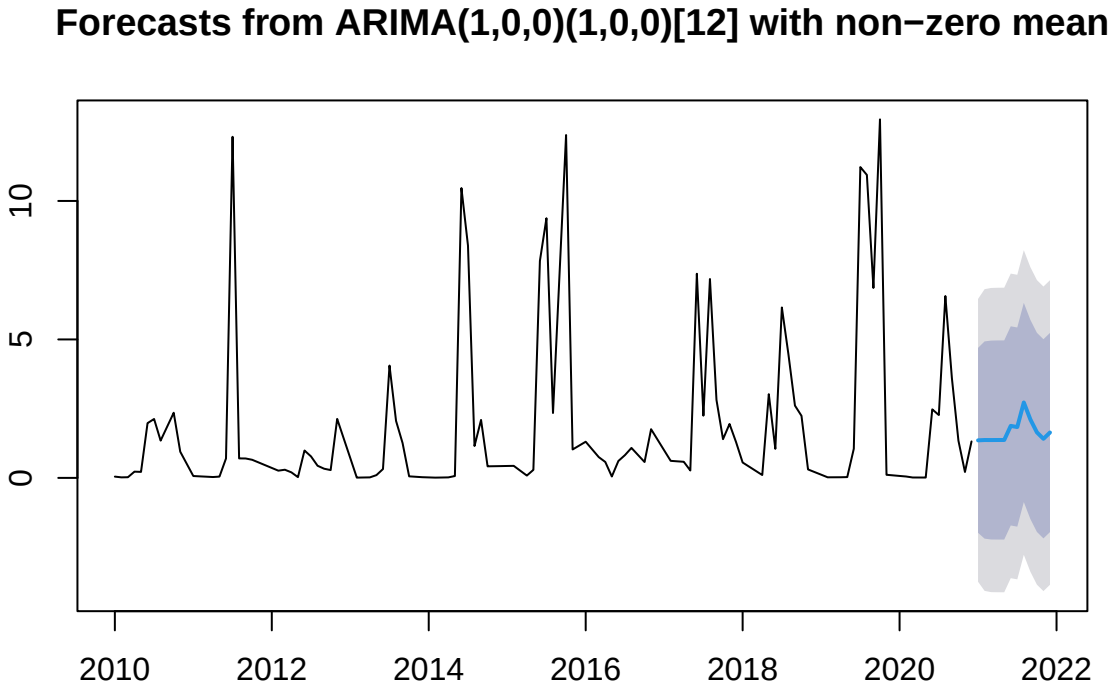
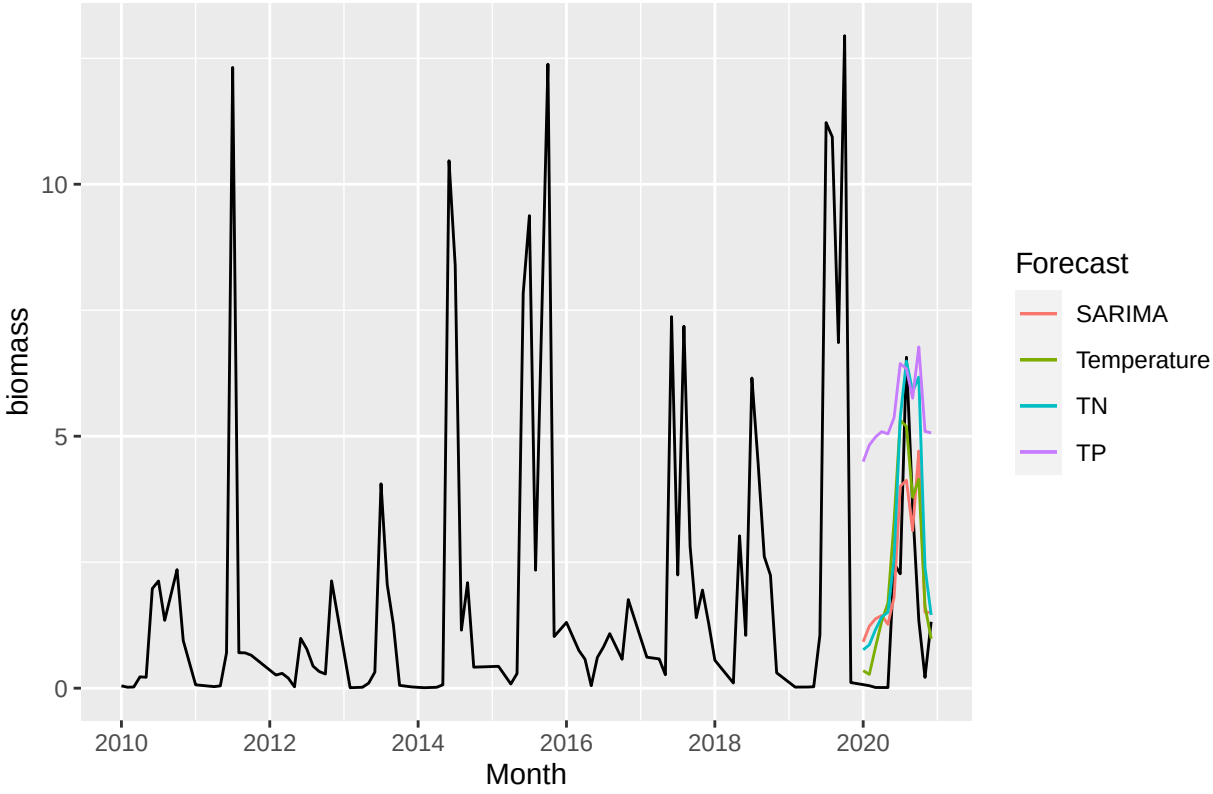

Forecasts from ARIMA(1,0,0)(1,0,0)[12] with non−zero mean
0
5
10
2010
2012
2014
2016
2018
2020
2022

Table 7: Forecast Accuracy for Explanatory Variable

	ME	RMSE	MAE	MPE	MAPE
SARIMA	-0.7553	1.5884	1.3542	-2867.005	2879.949
Temperature	-0.9066	1.5060	1.1919	-2538.268	2546.039
TN	-1.5077	2.0333	1.5205	-2864.176	2864.371
TP	-3.9441	4.2430	3.9827	-10924.066	10924.653

Result (2) env-biomass Kexin



Discussion

Reference

```
#To add reference, you need first add reference in reference in reference.bib.
#Than use"[@uniqueID]" to site it.
```


Beversdorf, T. R. M., Lucas J; Miller. (2015). Long-term monitoring reveals carbon–nitrogen metabolism key to microcystin production in eutrophic lakes. *Frontiers in Microbiology*, *6*, 456.

Brock, T. D. (2012). *A eutrophic lake: Lake mendota, wisconsin*. Springer Science & Business Media.

Magnuson, S. R. C., J.J., & H.Stanley, E. (2022). North temperate lakes LTER: Phytoplankton - madison lakes area 1995 - current. Retrieved from https://portal.edirepository.org/nis/mapbrowse?scope=knb-lter-ntl&identifier=88&revision=31