Forecasting the phytoplankton growth in Mendota Lake

Group 7: Yuxiang Ren, Kassie Huang, Yu Huan

Spring 2023 - ENVIRON790 TSA

Contents

Abstract

Algae bloom, also known as harmful algal blooms (HABs), occurs when there is an excessive growth of phytoplankton in a waterbody. Damage from harmful algal blooms (HABs) significantly impacts the environment and human life in various ways, highlighting the importance of predicting and providing early warnings for HABs.

In this project, three datasets were obtained from the EDI Data Portal with a focus on Mendota Lake (ME) due to its suitability for time series analysis based on its extensive time series data. Cyanophyte biomass was selected as the primary variable of interest, as this division had the highest total biomass and was determined to be the primary contributor to algae bloom outbreaks.

Among the five models evaluated, the SARIMA model demonstrated the highest accuracy. Additional investigation was conducted on how temperature, TN, and TP could improve model performance. We found out that adding temperature and TN slightly improved the accuracy of the original model, but the improvement was very small that the use of autoregression may be a more practical and cost-effective option for real-world forecasting.

Introduction

Algae bloom, also known as harmful algal blooms (HABs), occurs when there is an excessive growth of phytoplankton in a water body. This overgrowth can be caused by various factors, including agriculture, which contributes to the development of HABs in several ways. Agricultural practices such as nutrient runoff from excessive fertilizer use, soil erosion from unsustainable practices, livestock waste management, aquaculture, and inefficient irrigation systems can all lead to increased levels of nitrogen and phosphorus in nearby water bodies (Chakraborty, 2017).

Damage from harmful algal blooms (HABs) significantly impacts the environment and human life in various ways. These consequences include creating dead zones where aquatic life cannot survive due to oxygen depletion, disrupting the natural balance of aquatic ecosystems and leading to biodiversity loss, contaminating drinking water supplies with toxins that pose health risks, and affecting recreational water activities (Joyce, 2000). Moreover, HABs have economic repercussions on industries dependent on clean water, such as commercial fishing, aquaculture, and tourism, leading to financial losses and negatively impacting local economies (Treuer, 2021).

Therefore, predicting and providing early warnings for harmful algal blooms (HABs) has become increasingly important. Governments in many regions have started to monitor relevant data to establish early warning systems for HAB occurrences. Lakes that adopt water quality-based early warning mechanisms tend to have a greater potential to predict HAB events in advance compared to those relying on biomass or remote sensing images. In this study, we aim to predict algal populations in lakes using time series analysis methods and identify the most suitable forecasting model. By analyzing historical data and examining trends, we hope to better understand the factors influencing algal growth and develop effective strategies for predicting and managing harmful algal blooms.

Data Processing

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 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

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

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.

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

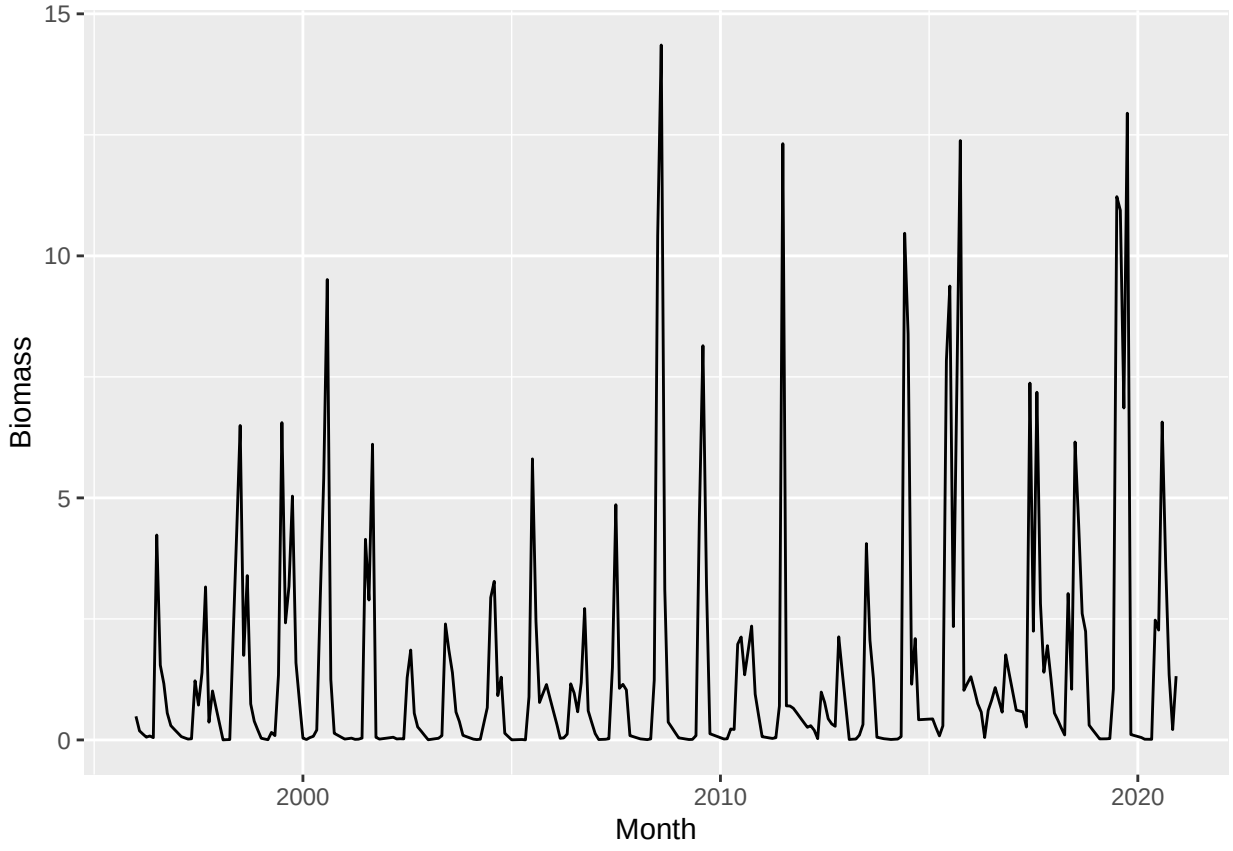
date	Temperature	TN	TP	Biomass
1996-01-01	3.702222	0.7277500	0.1163333	0.4872730
1996-02-01	5.104444	0.7835000	0.1096667	0.1878542
1996-03-01	6.506667	0.8120556	0.1080556	0.1224752
1996-04-01	7.908889	0.8406111	0.1064444	0.0570962
1996-05-01	9.311111	0.8691667	0.1048333	0.0817296

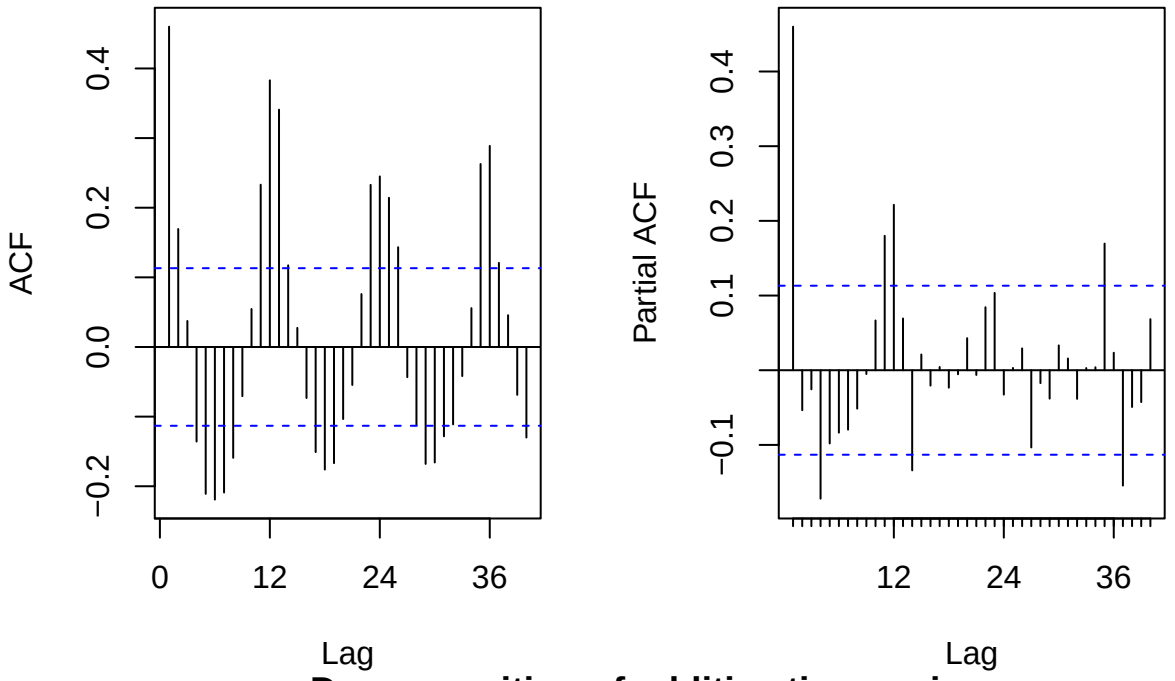
e. The final dataset includes dates (from 1996 to December 2020), temperature, total nitrogen, total phosphorus, and biomass (Table 4).

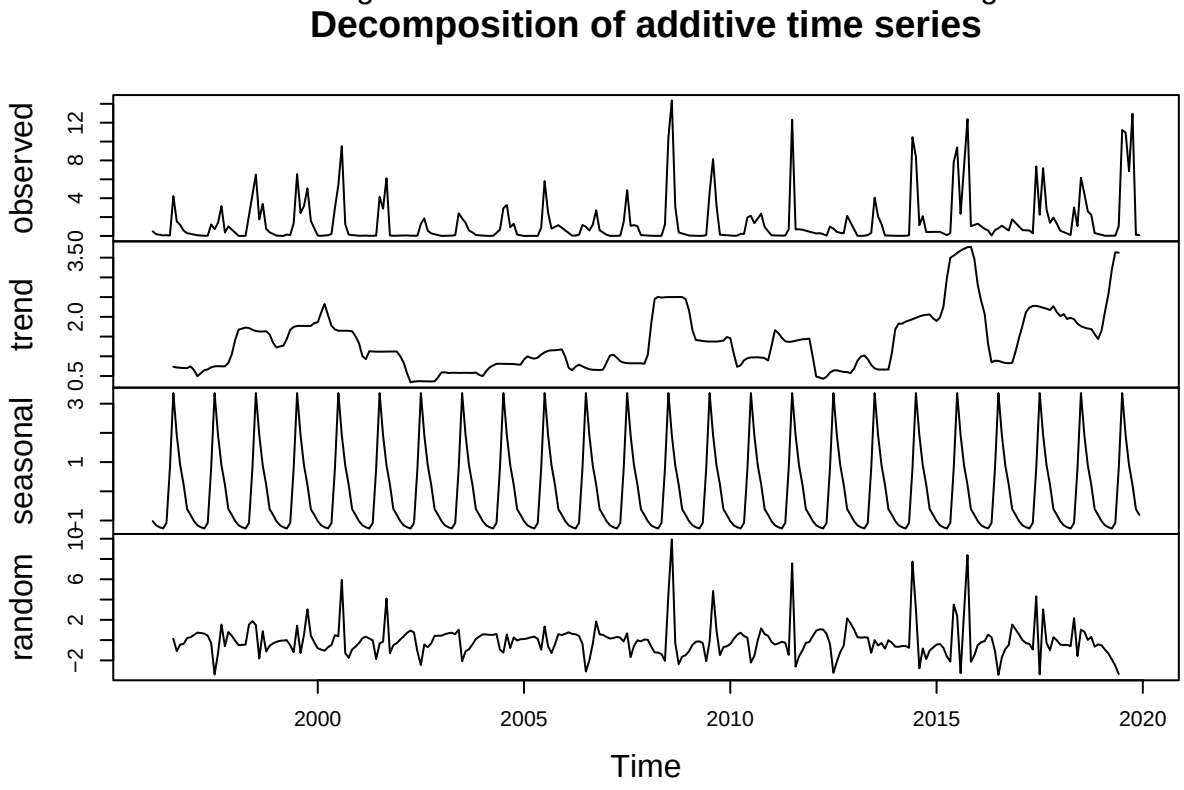
Results

Result1 Auto-regression

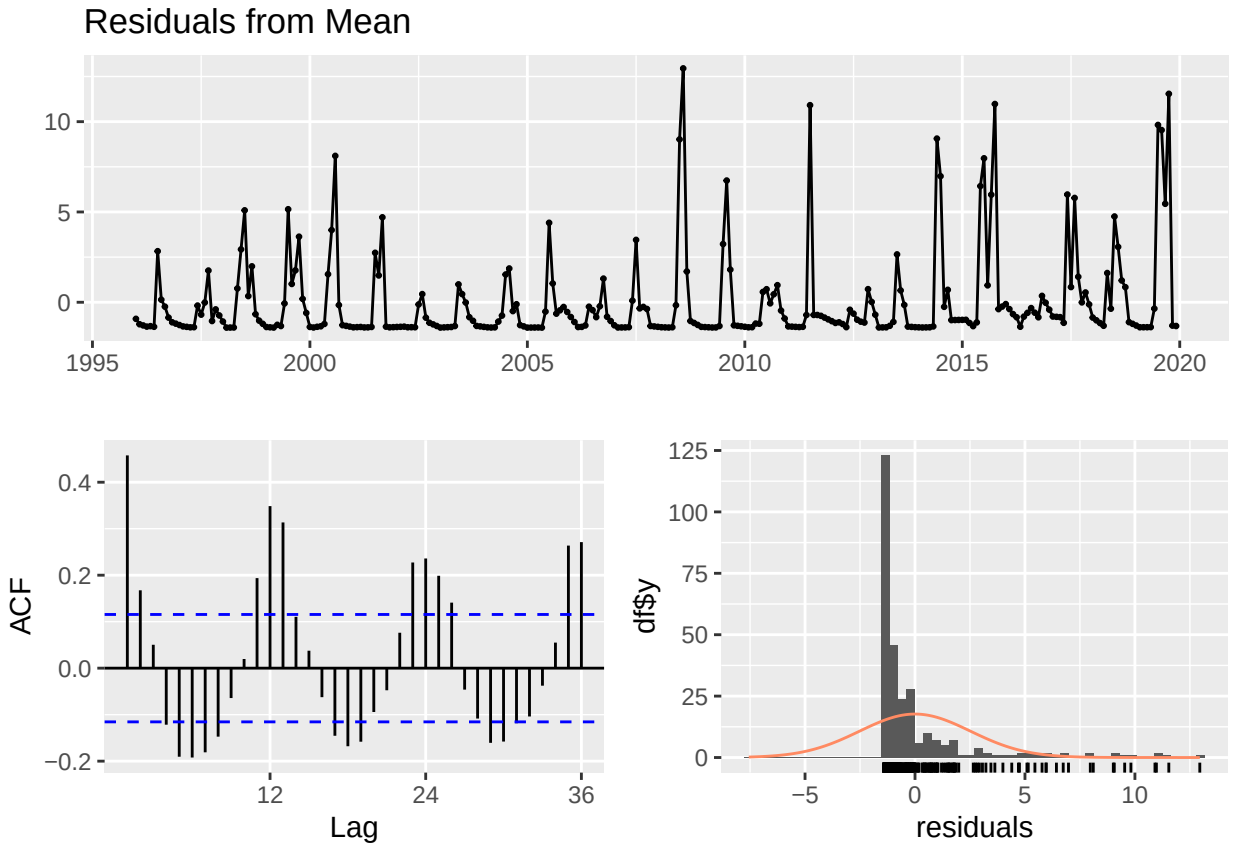
Original (1996-2020)





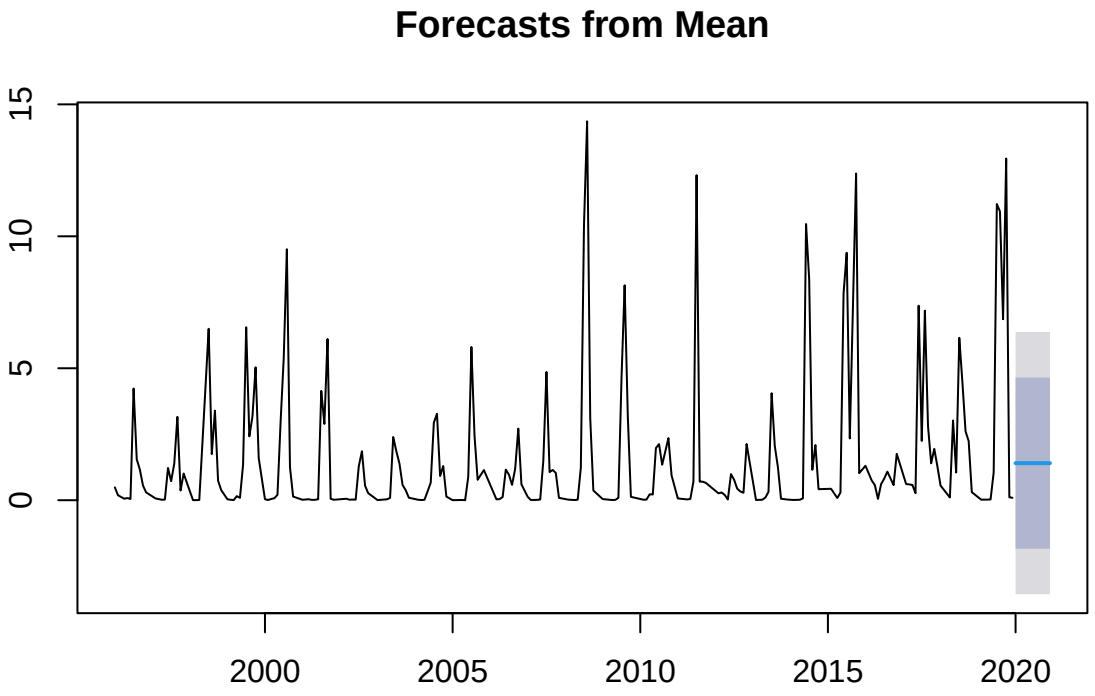



```
# Model 1: Arithmetic mean
# The meanf() has no holdout option
MEAN_seas <- meanf(y = ts_biomass, h = 12)
checkresiduals(MEAN_seas)
```

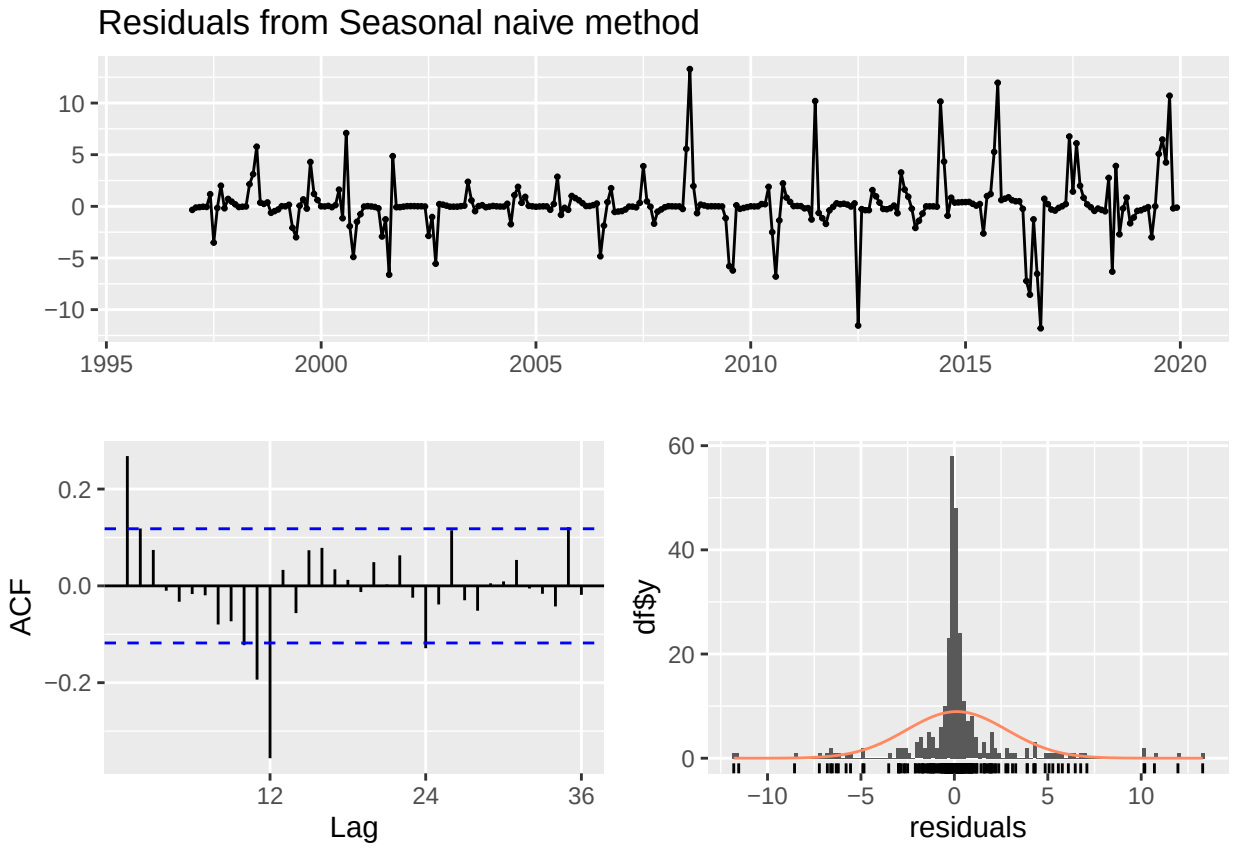



```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 258.95, df = 23, p-value < 2.2e-16
##
## Model df: 1.   Total lags used: 24
```

```
plot(MEAN_seas)
```

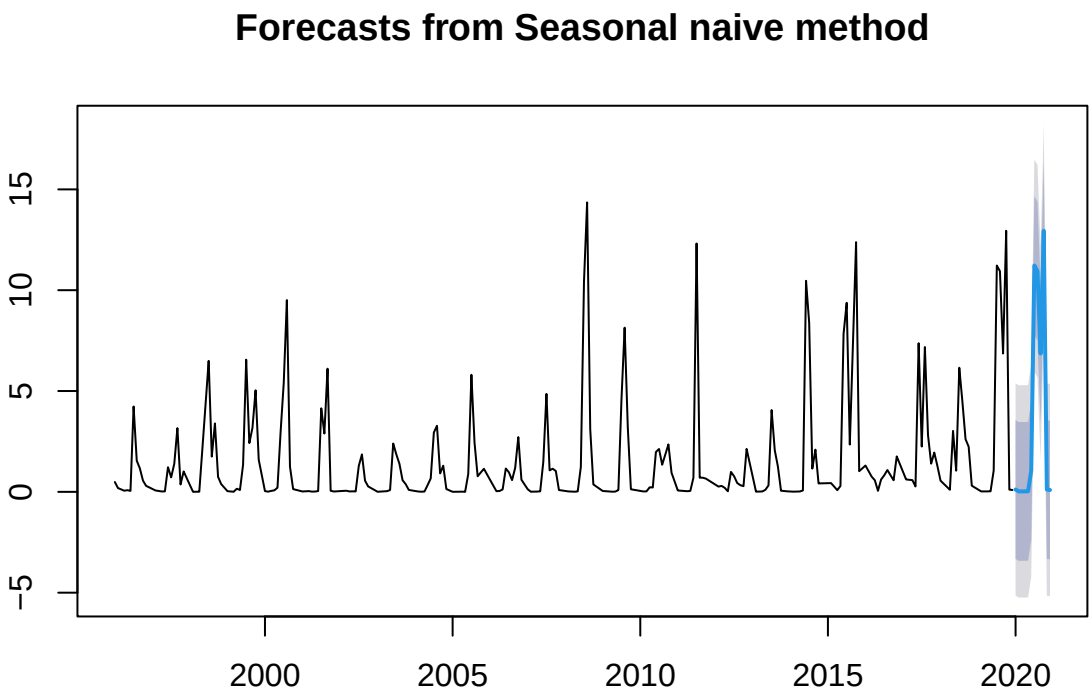



```
# Model 2: Seasonal naive
SNAIVE_seas <- snaive(ts_biomass, h=12, holdout=FALSE)
checkresiduals(SNAIVE_seas)
```

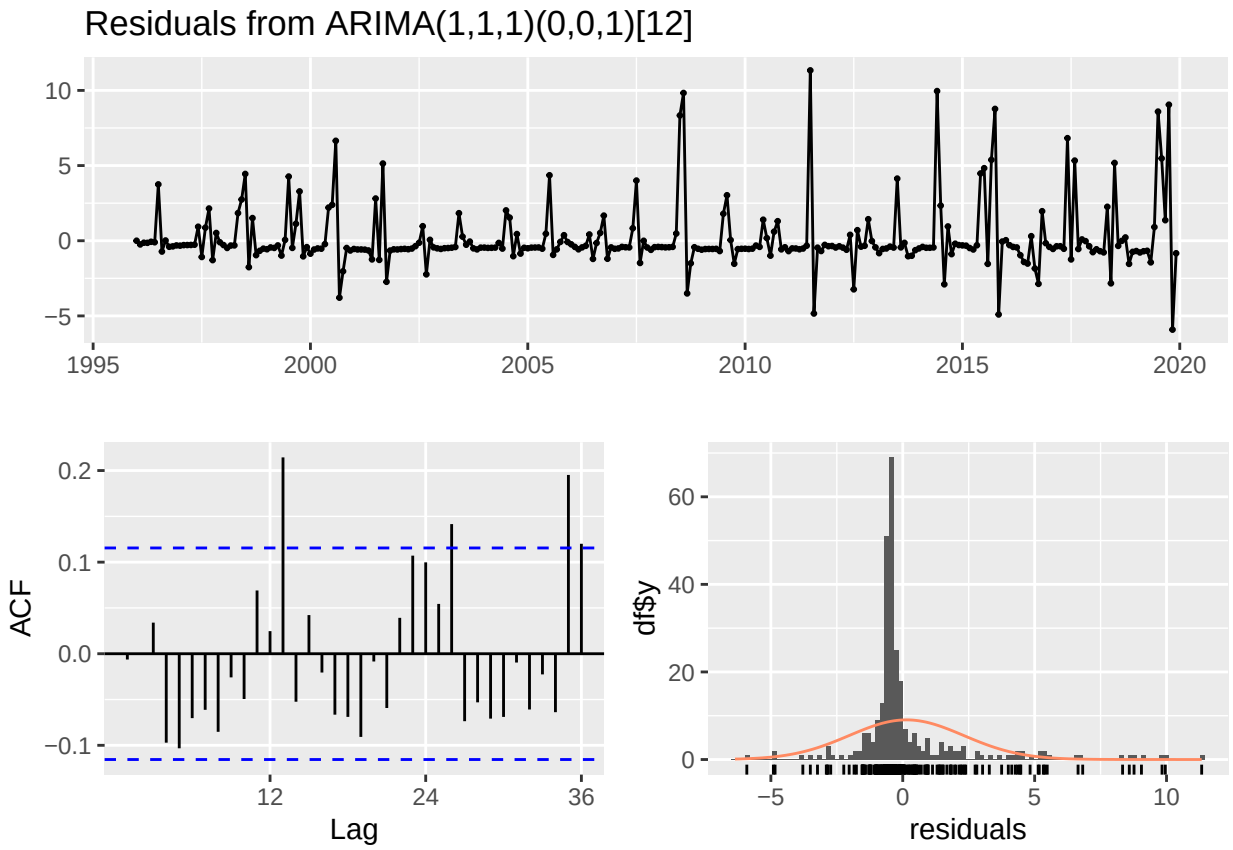


```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 93.651, df = 24, p-value = 3.556e-10
##
## Model df: 0.   Total lags used: 24
```

```
plot(SNAIVE_seas)
```

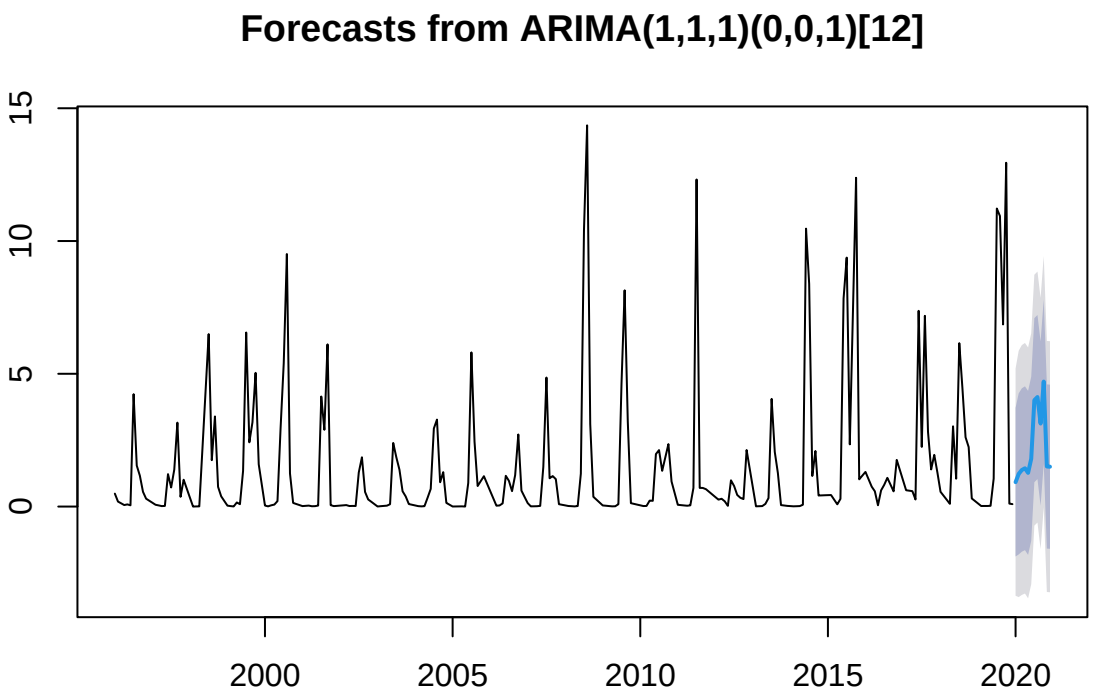



```
# Model 3: SARIMA

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

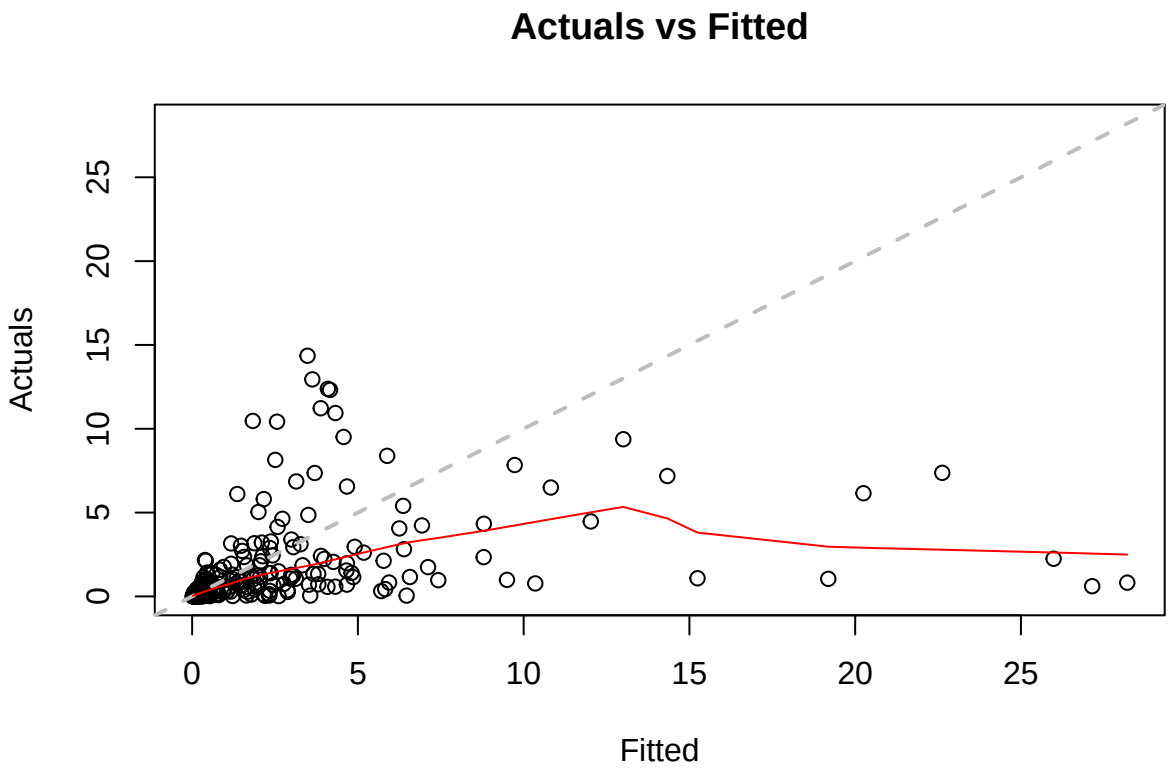


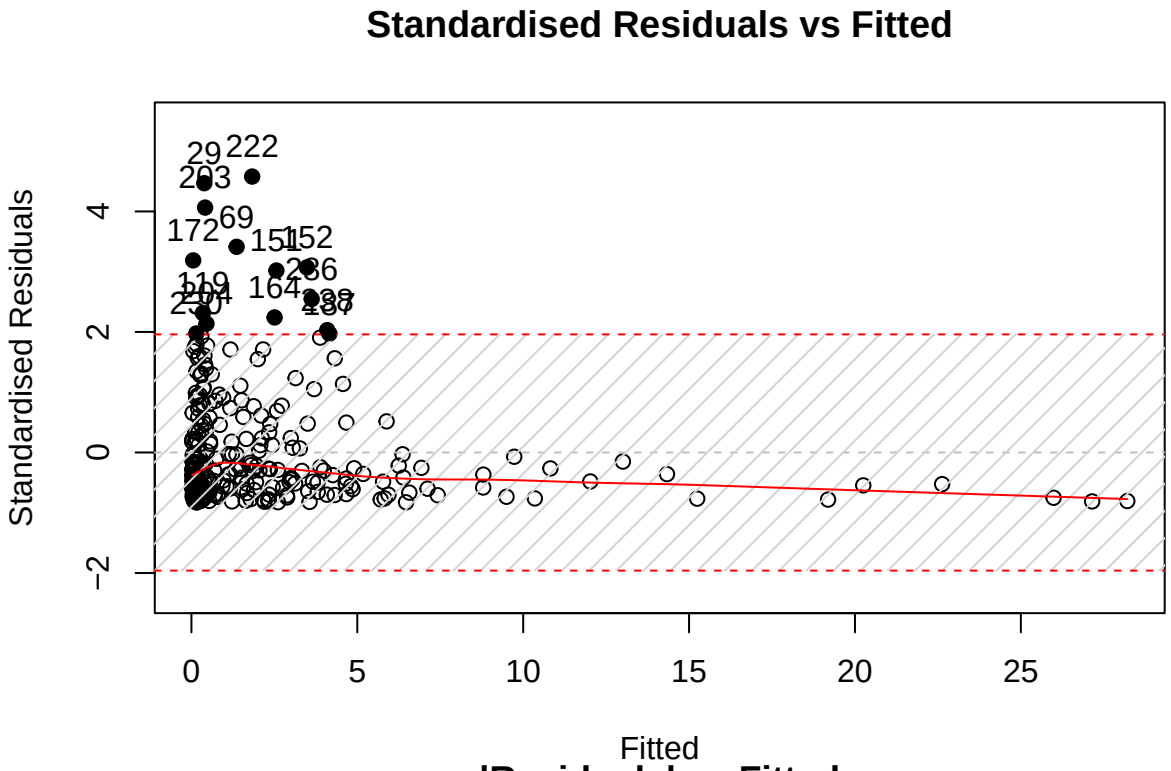

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,0,1)[12]
## Q* = 42.8, df = 21, p-value = 0.003333
##
## Model df: 3.   Total lags used: 24
```

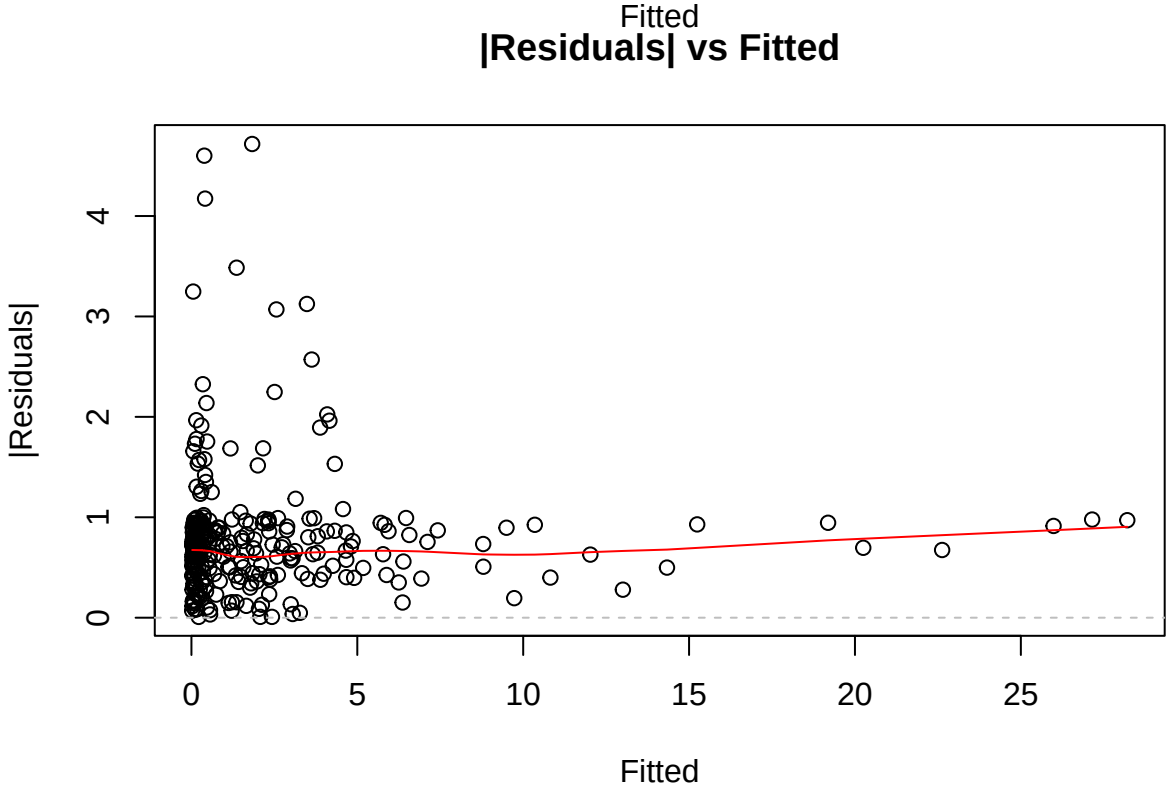
```
#Generating forecasts
#remember auto.arima does not call the forecast() internally so we need one more step
SARIMA_for <- forecast(SARIMA_autofit,h=12)
plot(SARIMA_for)
```

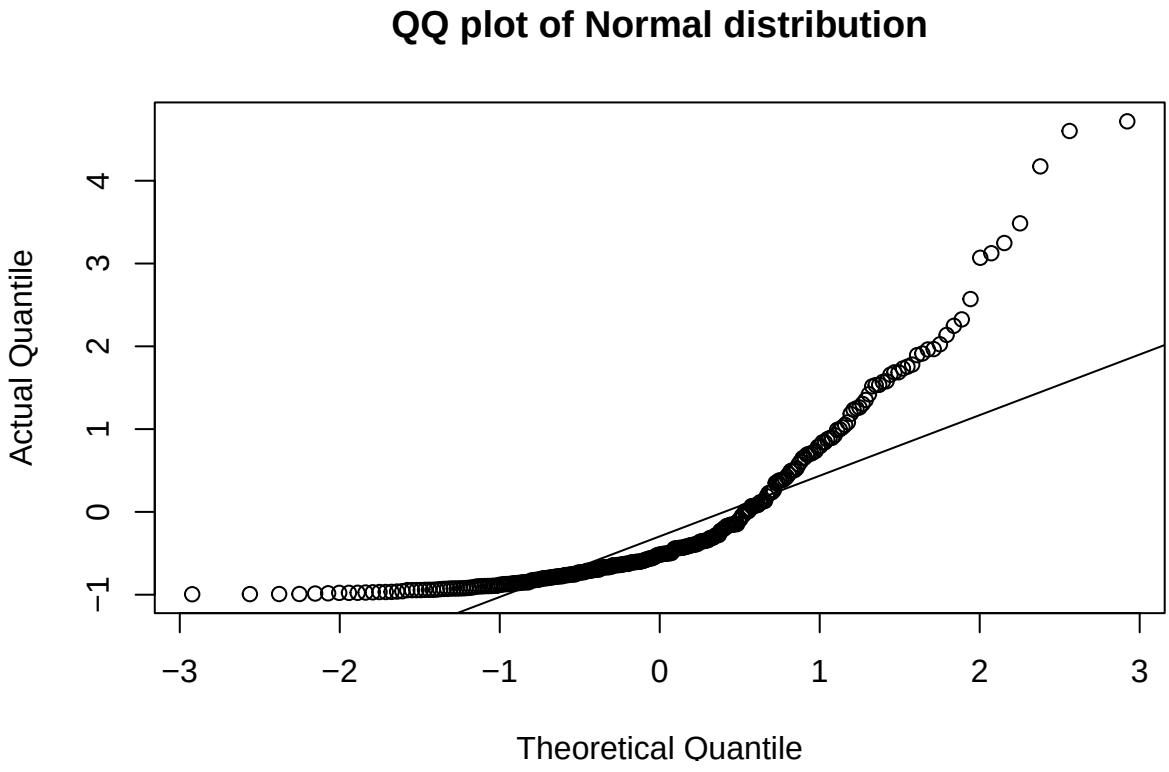



```
# Model 4: SS Exponential smoothing
SSES_seas <- es(ts_biomass,model="ZZZ",h=12,holdout=FALSE)
plot(SSES_seas)
```



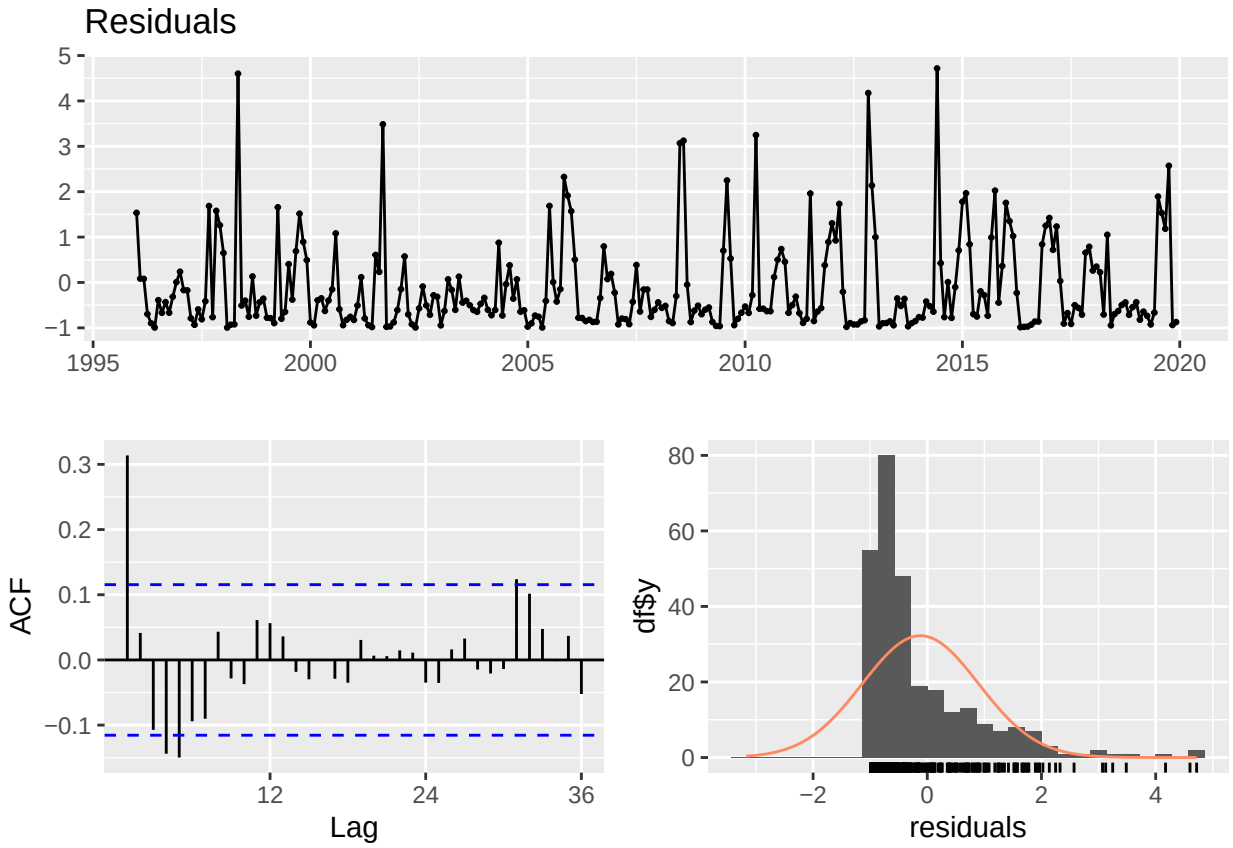

Standardised Residuals vs Fitted
Standardised Residuals
Fitted
29
222
203
69
172
151
152
119
164
−2
0
2
4
0
5
10
15
20
25


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



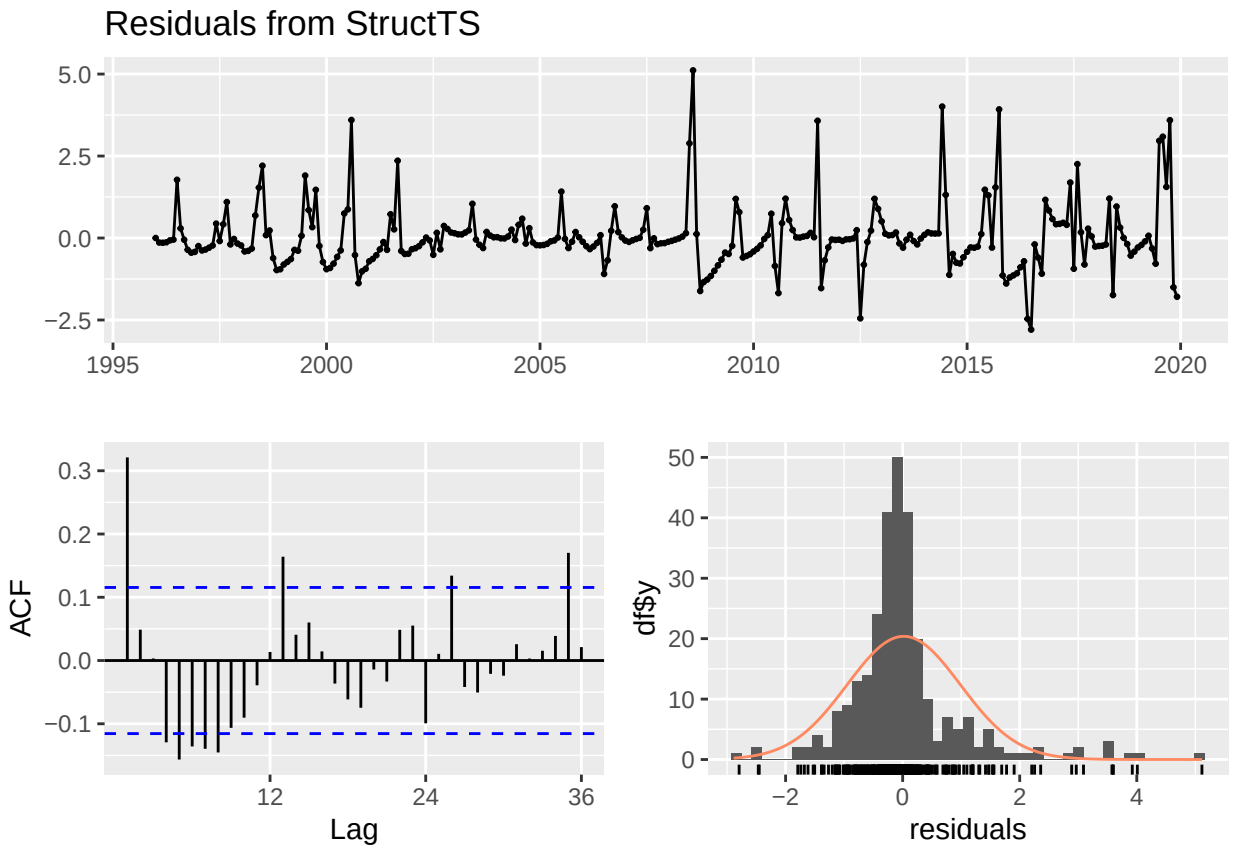

```
checkresiduals(SSES_seas)
```

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

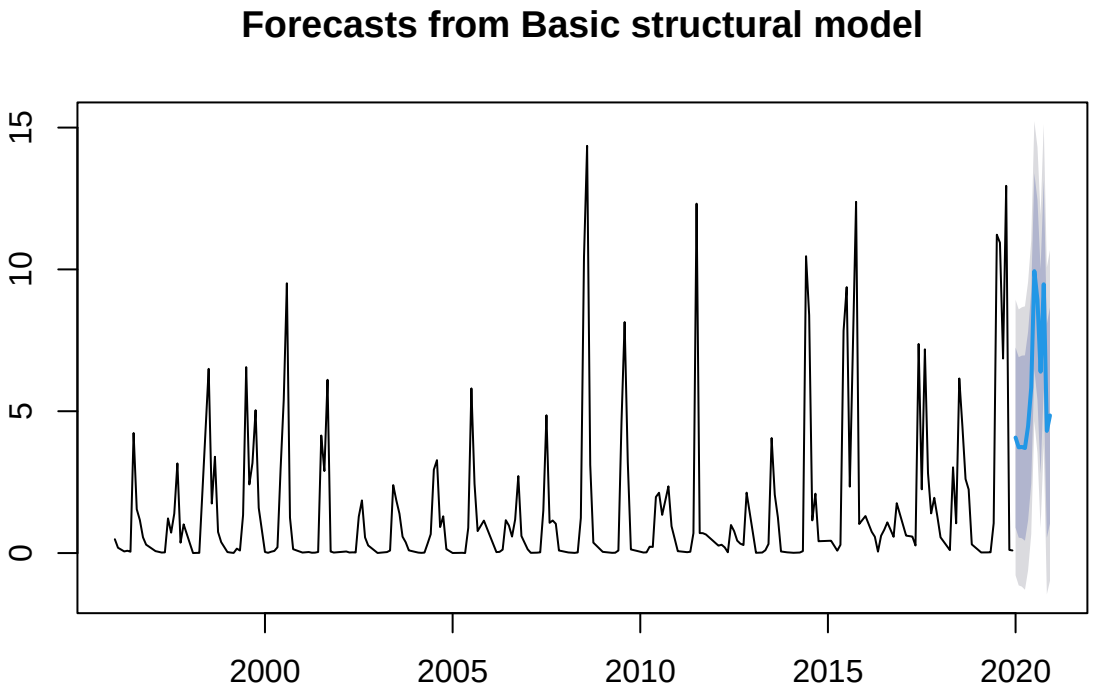



```
# Model 5: SS with StructTS()

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,]))
```

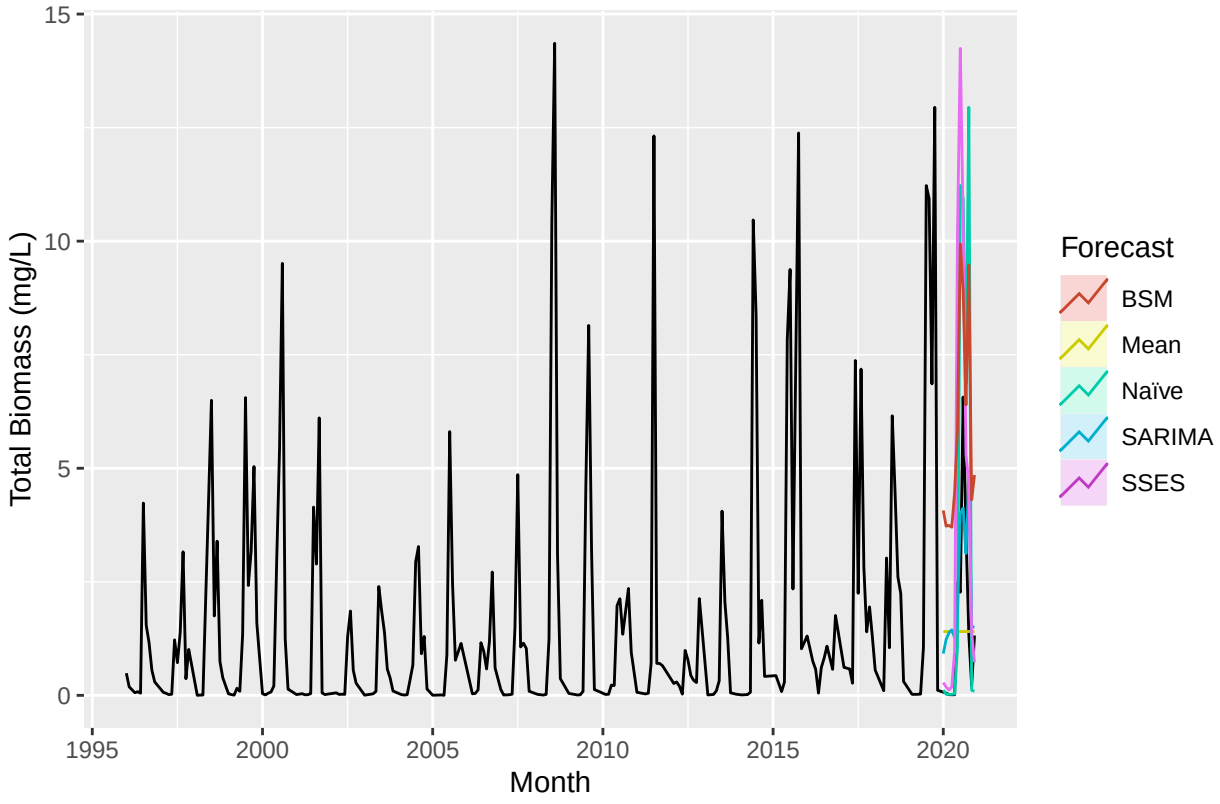
```
## The best model by RMSE is: SARIMA
```

Table 5: Forecast Accuracy for Seasonal Data

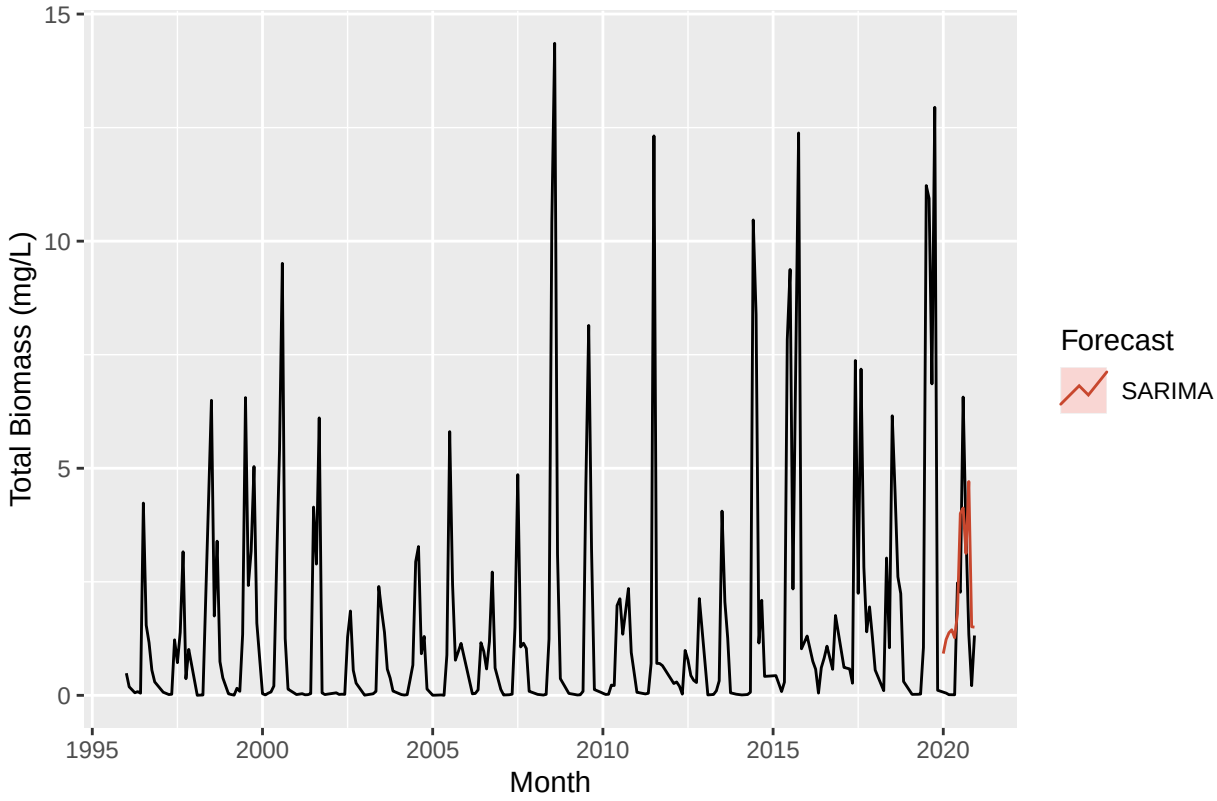
	ME	RMSE	MAE	MPE	MAPE
MEAN	0.09077	1.91972	1.45817	-2993.9993	3030.8195
SNAIVE	-2.12360	4.54636	2.58677	-122.4177	164.9659
SARIMA	-0.75532	1.58841	1.35418	-2867.0051	2879.9488
SSES	-2.55079	4.40694	2.64893	-998.5011	1005.9502
BSM	-4.29032	4.61610	4.29032	-8743.8441	8743.8441

```
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"]))
```

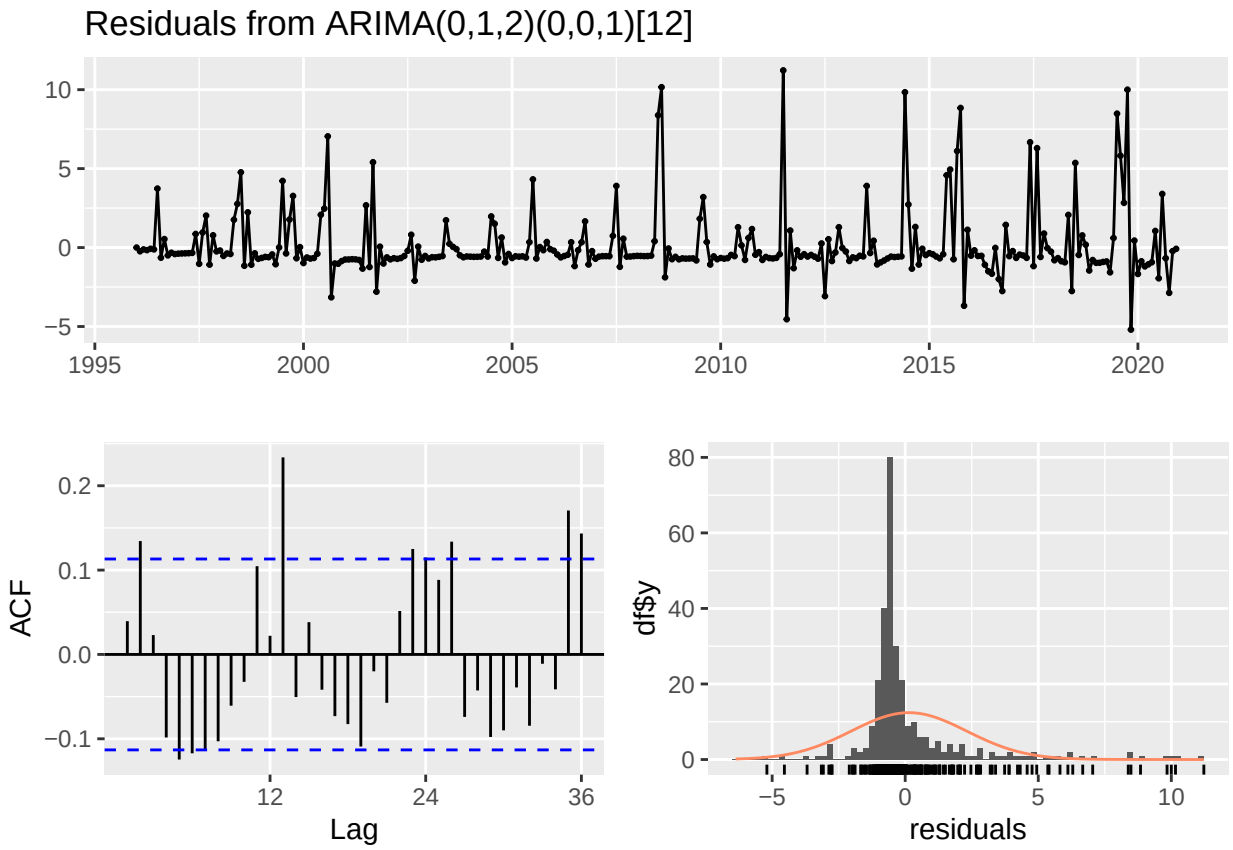
```
autoplot(ts_biomass_data) +
  autolayer(MEAN_seas, PI=FALSE, series="Mean") +
  autolayer(SNAIVE_seas, PI=FALSE, series="Naïve") +
  autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  autolayer(SSES_seas$forecast, series="SSES") +
  autolayer(SS_for,PI=FALSE,series="BSM") +
  xlab("Month") + ylab("Total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




```
autoplot(ts_biomass_data) +

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

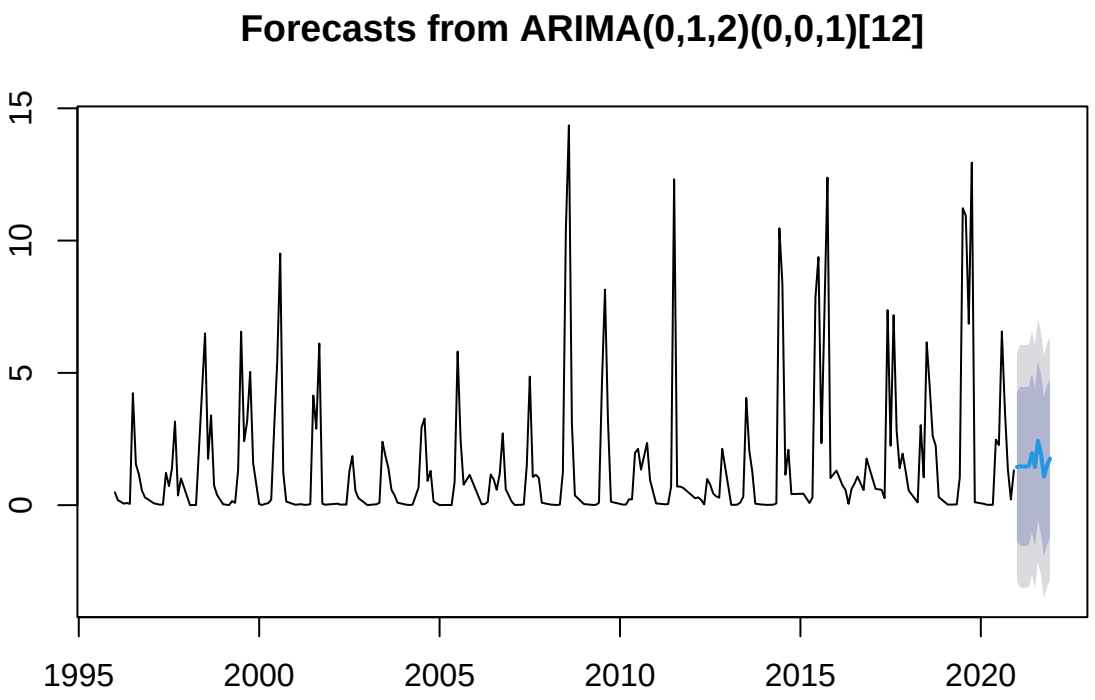



```
# Forecast

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



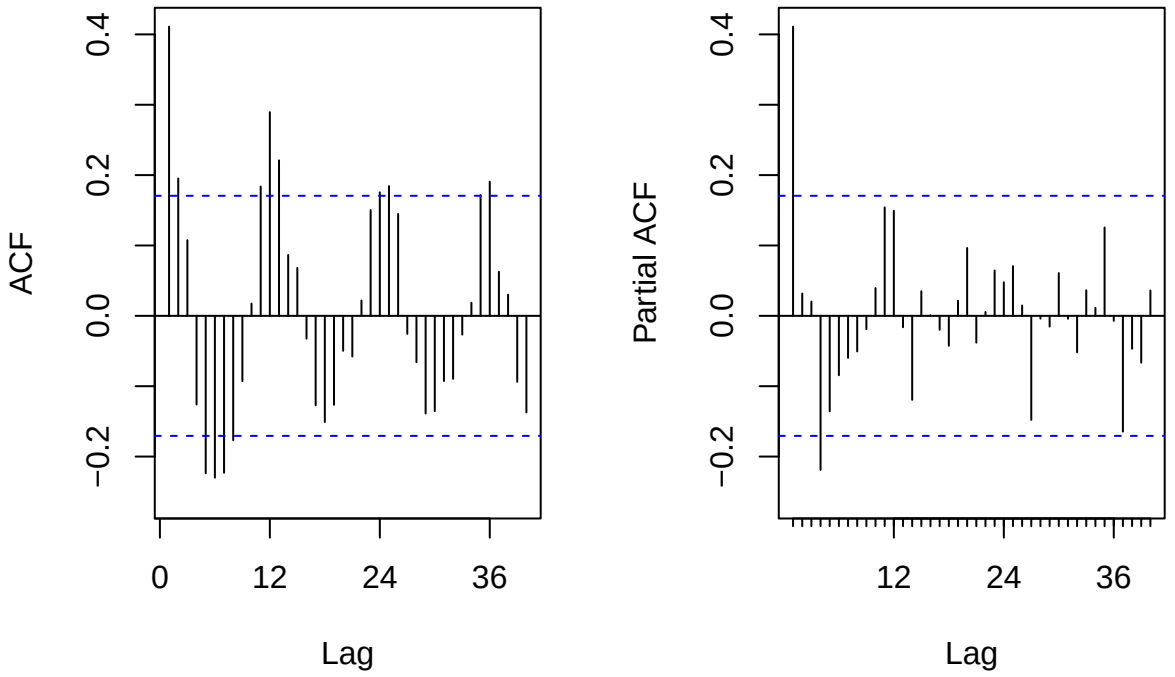

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)(0,0,1)[12]
## Q* = 68.766, df = 21, p-value = 5.534e-07
##
## Model df: 3.   Total lags used: 24
```

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

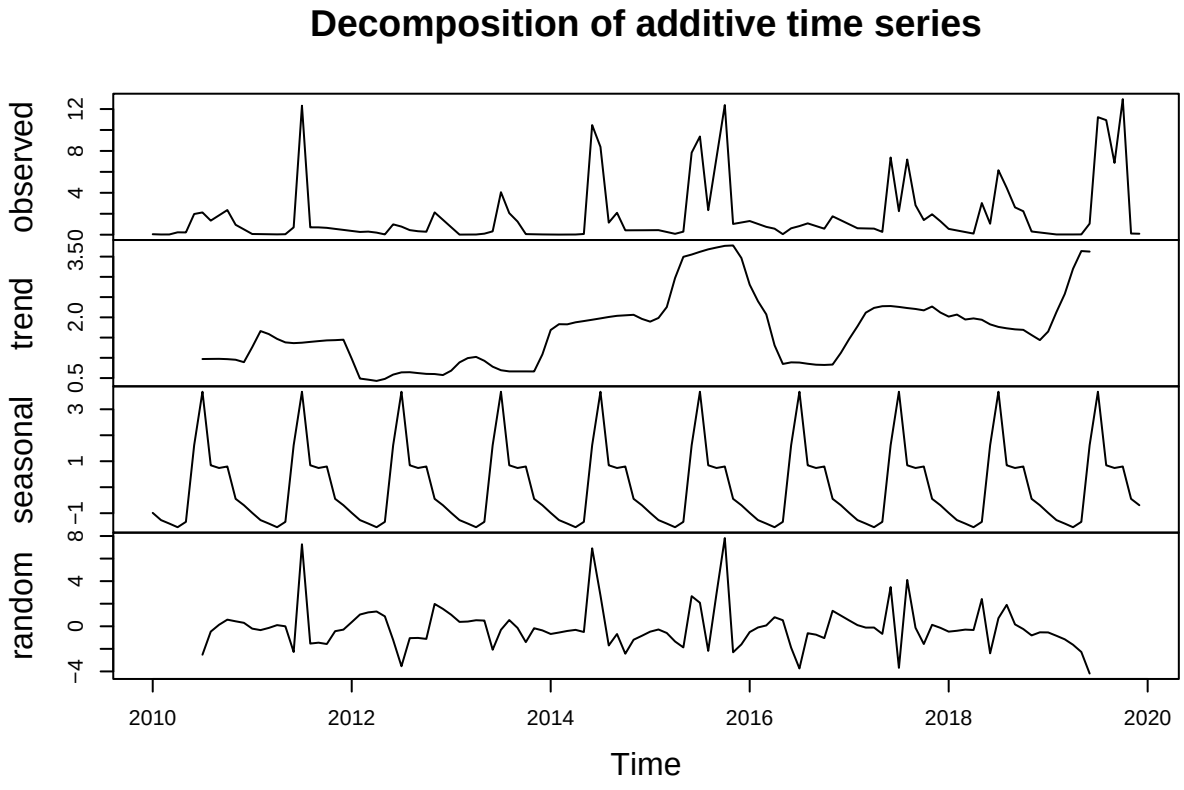


Use recent ten-year data (2010-2020) to forecast

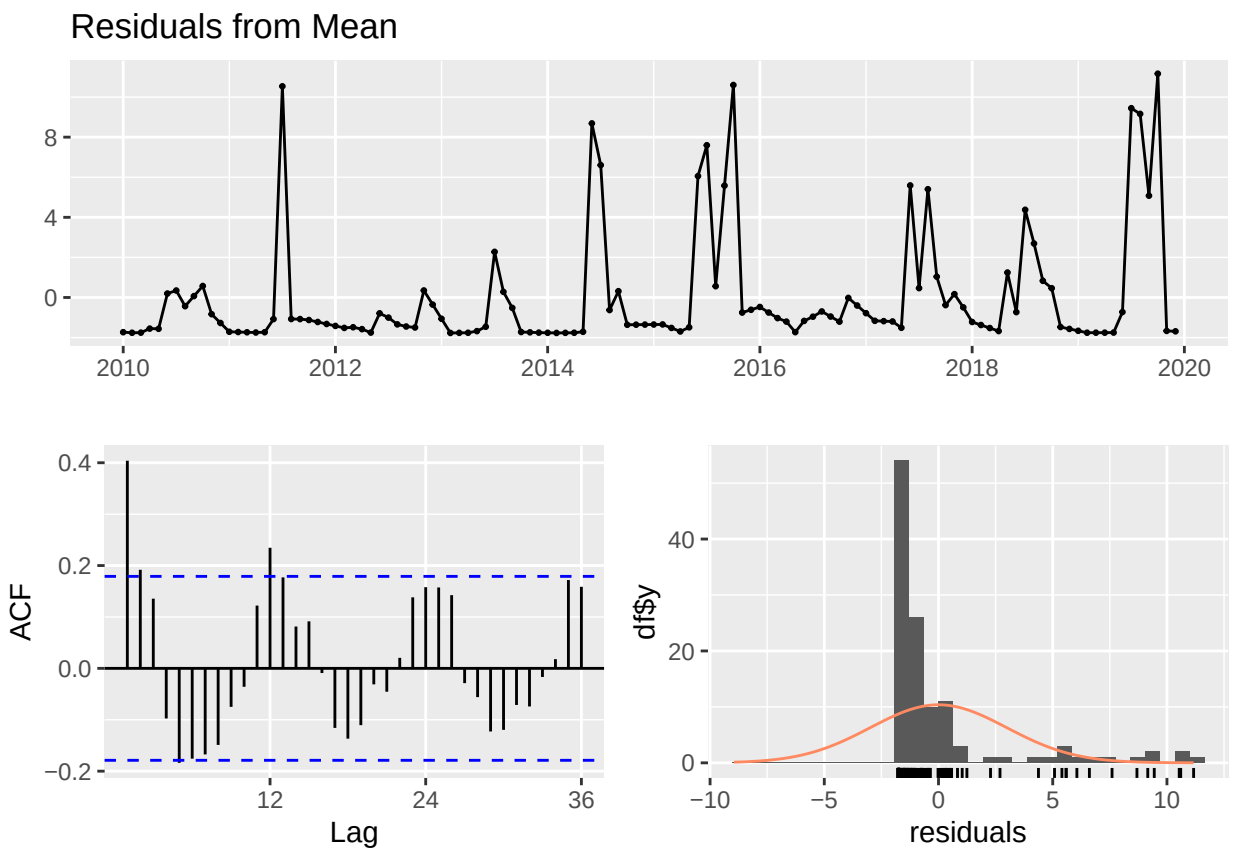
```
# Change the time span

# 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]


#ACF and PACF plots
par(mfrow=c(1,2))
ACF_Plot <- Acf(ts_biomass_data, lag = 40, plot = TRUE,main="")
PACF_Plot <- Pacf(ts_biomass_data, lag = 40, plot = TRUE,main="")
```




```
#Plot ts decompose
decompose_biomass_data <- decompose(ts_biomass,"additive")

plot(decompose_biomass_data)
```

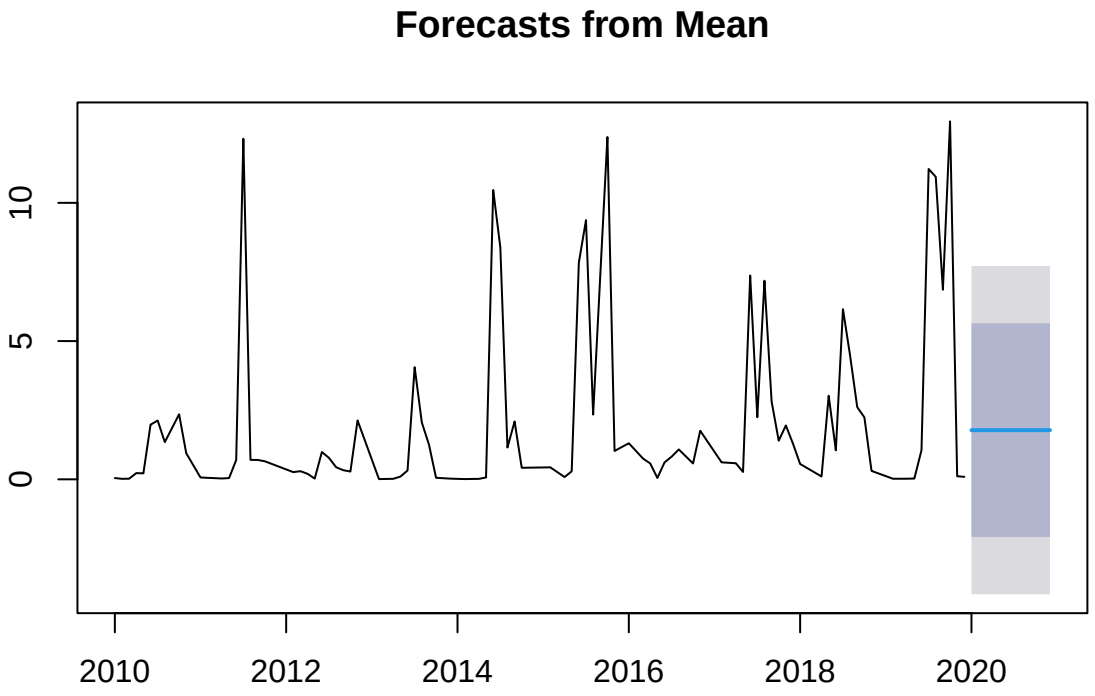


```
# Model 1: Arithmetic mean
# The meanf() has no holdout option
MEAN_seas <- meanf(y = ts_biomass, h = 12)
checkresiduals(MEAN_seas)
```

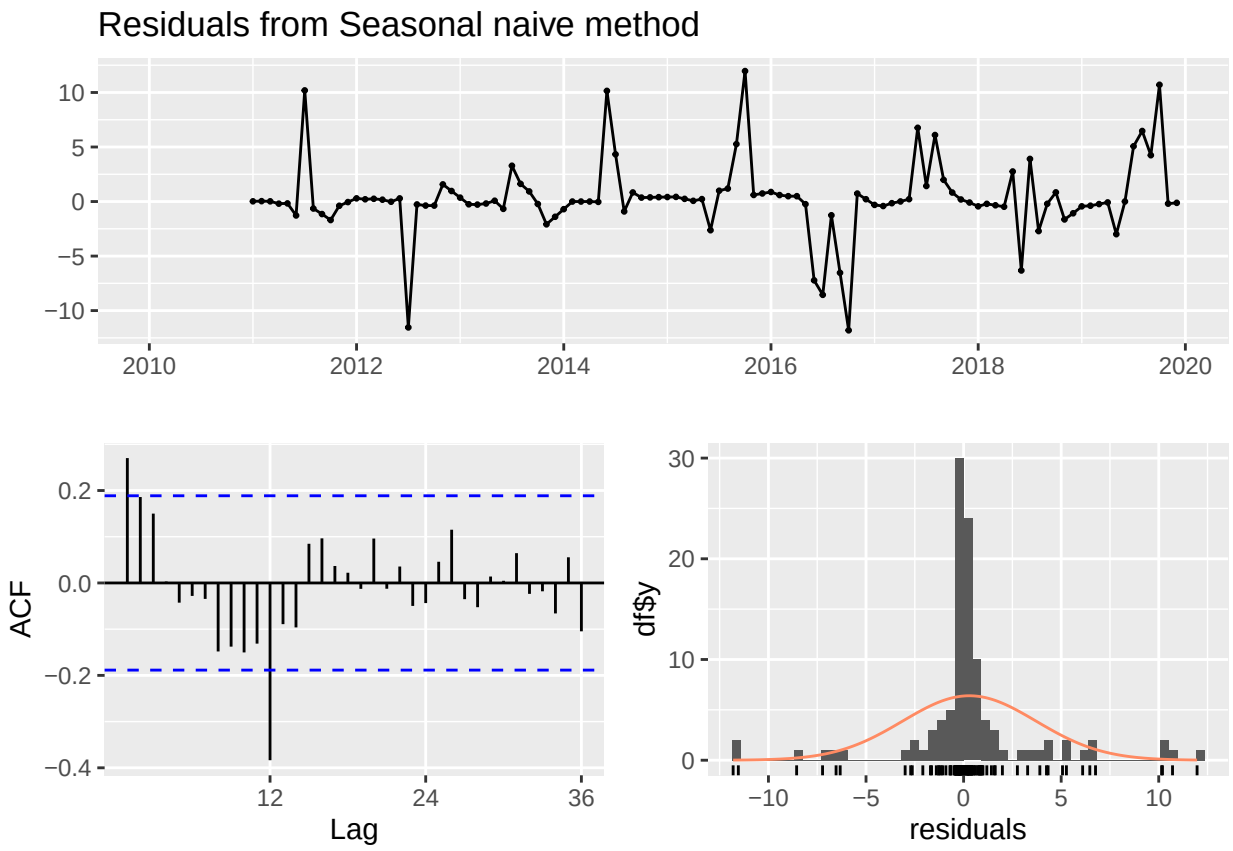



```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 73.247, df = 23, p-value = 3.795e-07
##
## Model df: 1.   Total lags used: 24
```

```
plot(MEAN_seas)
```

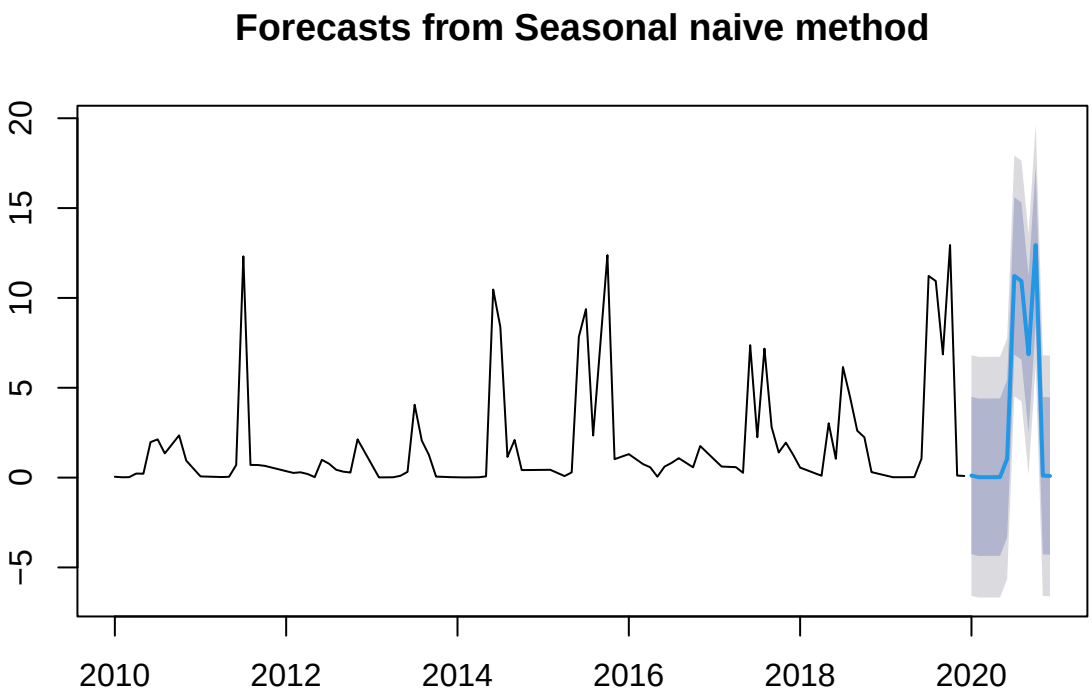



```
# Model 2: Seasonal naive
SNAIVE_seas <- snaive(ts_biomass, h=12, holdout=FALSE)
checkresiduals(SNAIVE_seas)
```

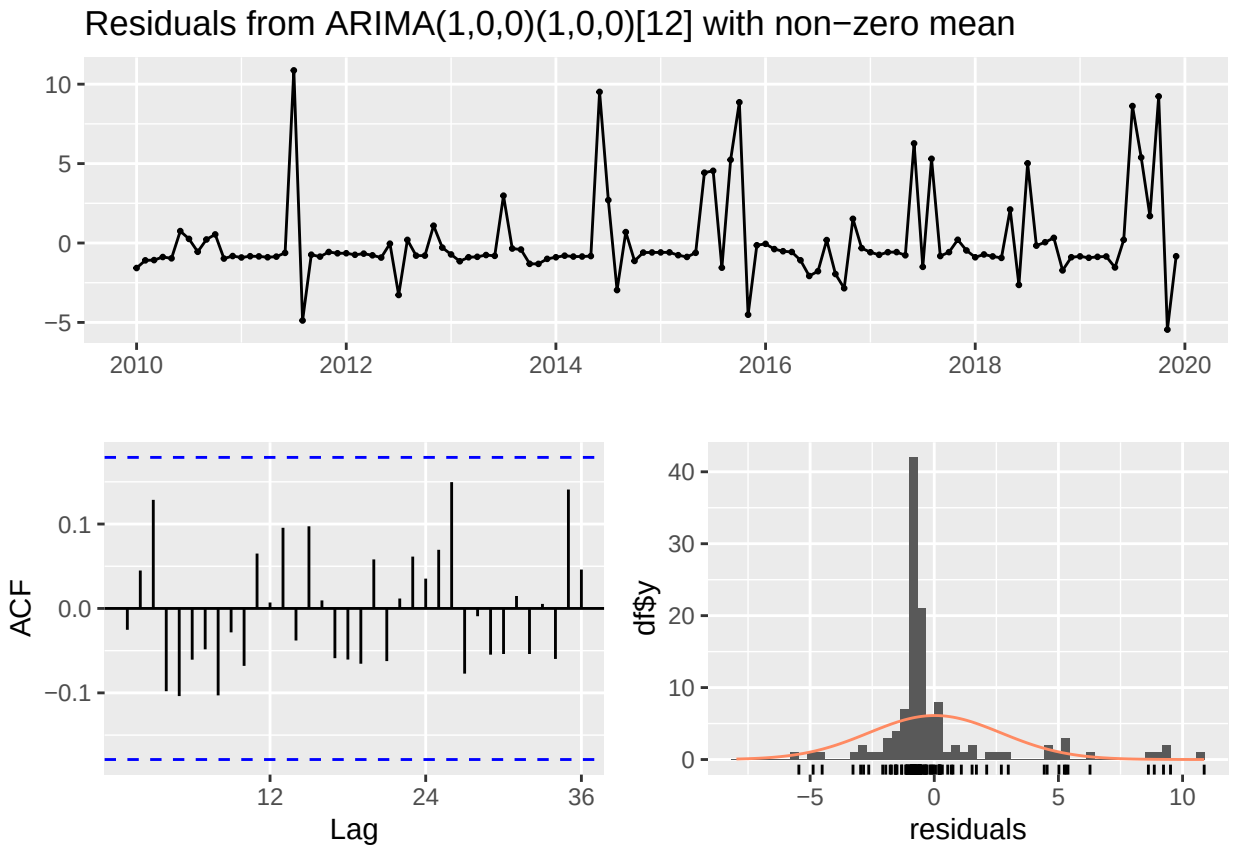


```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 49.511, df = 24, p-value = 0.001633
##
## Model df: 0.   Total lags used: 24
```

```
plot(SNAIVE_seas)
```

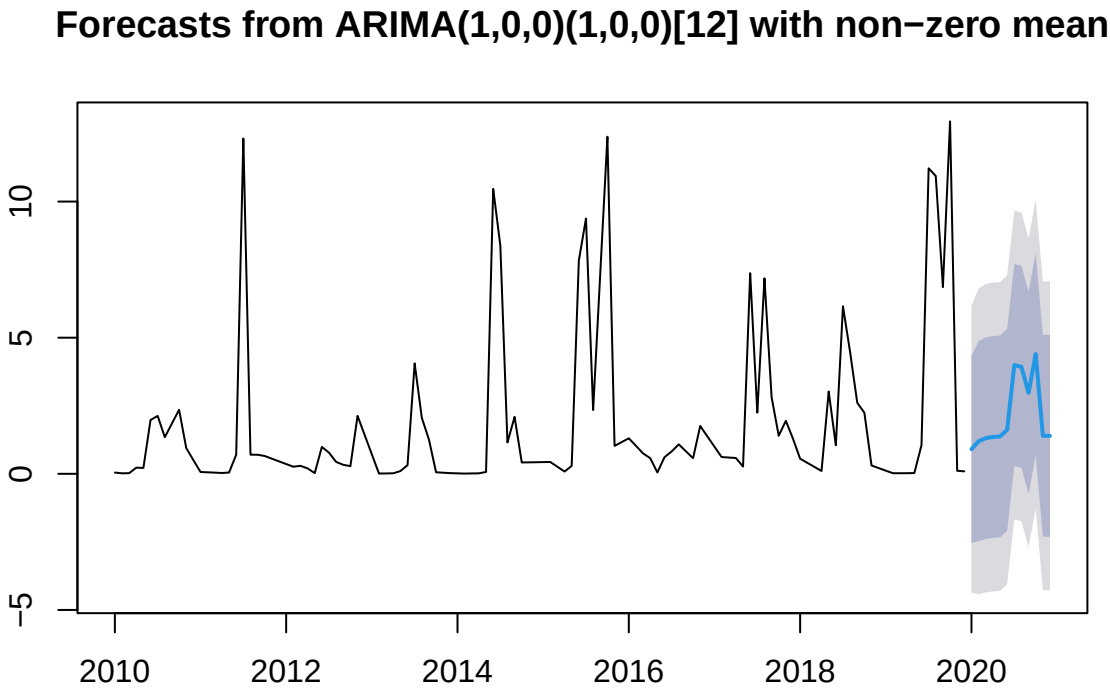



```
# Model 3: SARIMA

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

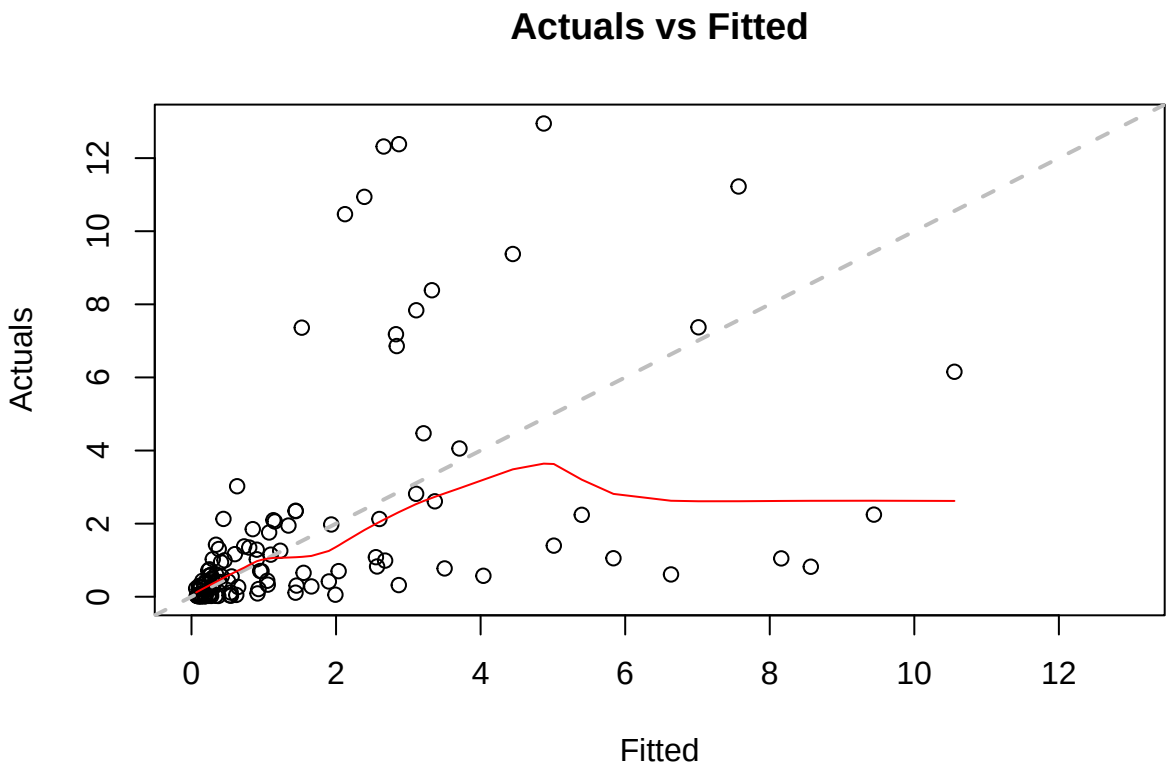


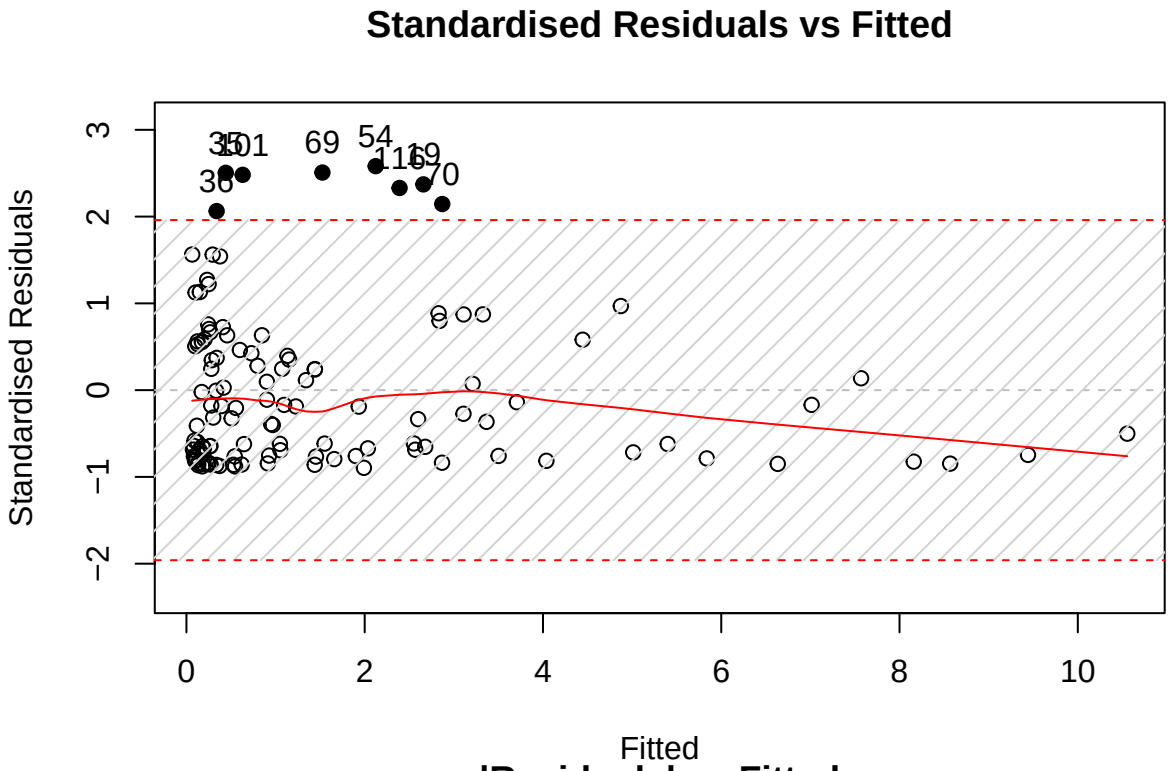

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,0,0)[12] with non-zero mean
## Q* = 14.711, df = 22, p-value = 0.8743
##
## Model df: 2.   Total lags used: 24
```

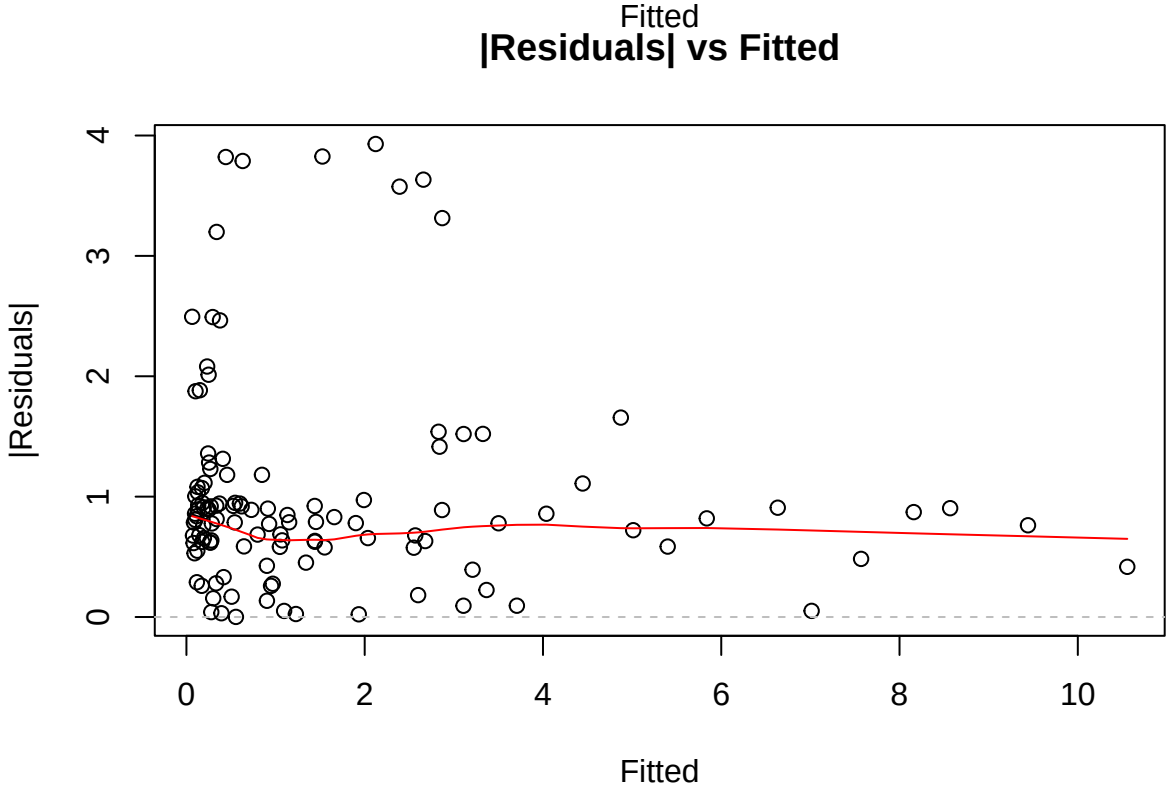
```
#Generating forecasts
#remember auto.arima does not call the forecast() internally so we need one more step
SARIMA_for <- forecast(SARIMA_autofit,h=12)
plot(SARIMA_for)
```

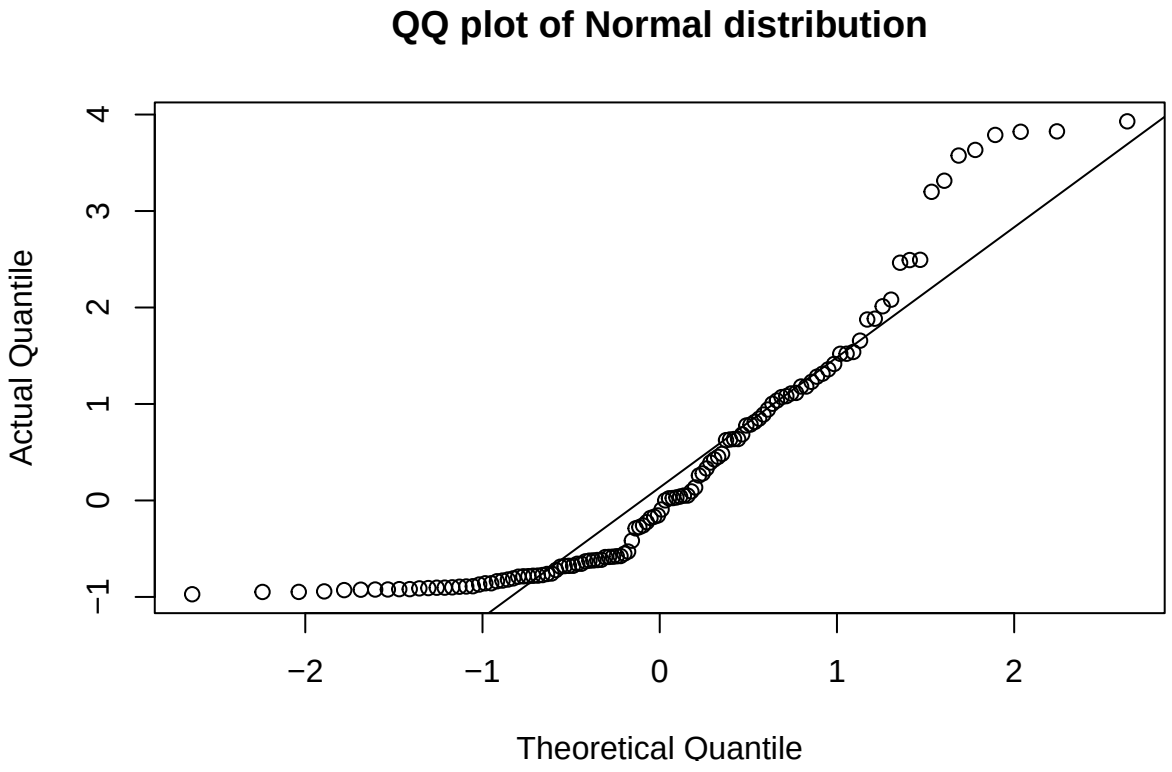



```
# Model 4: SS Exponential smoothing
SSES_seas <- es(ts_biomass,model="ZZZ",h=12,holdout=FALSE)
plot(SSES_seas)
```



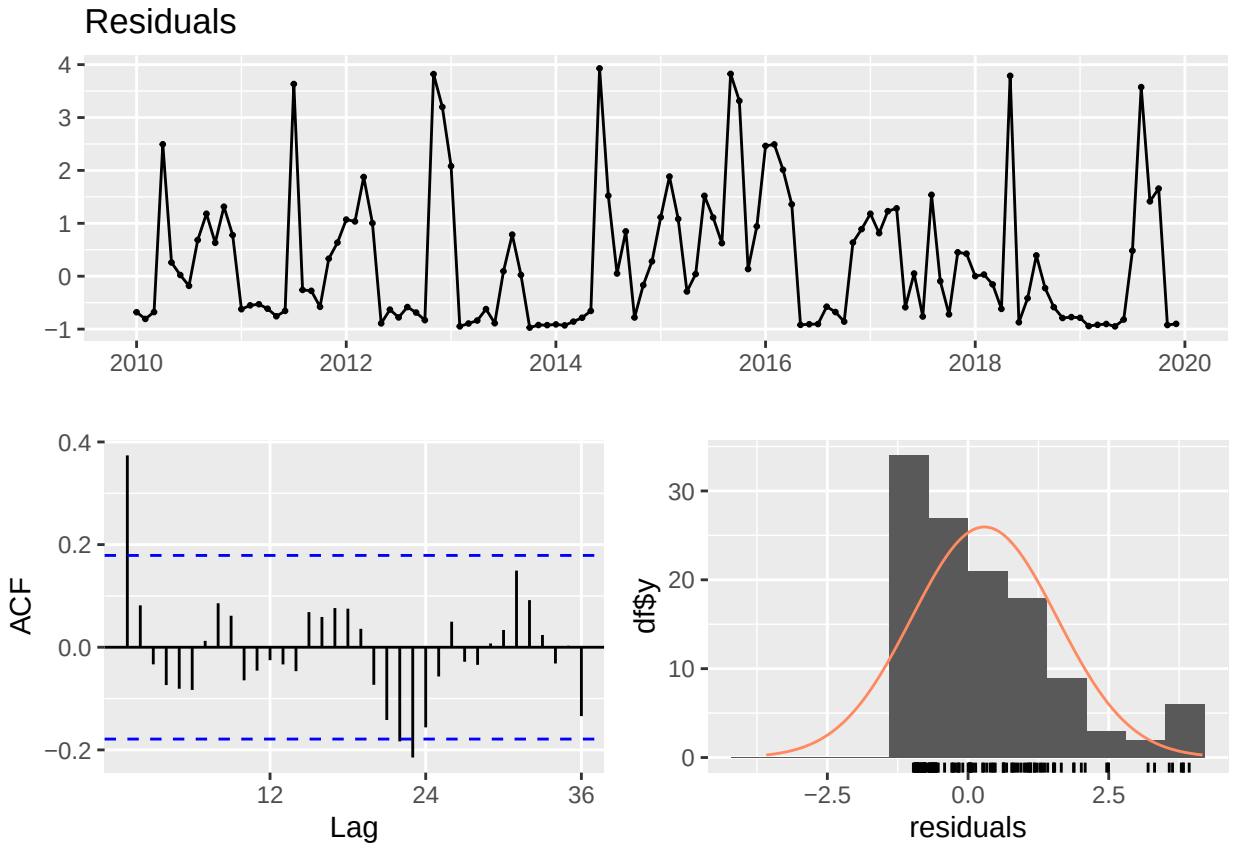

Standardised Residuals vs Fitted
Standardised Residuals
Fitted
35
101
69
54
116
69
70
36
3
2
1
0
−1
−2
0
2
4
6
8
10


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



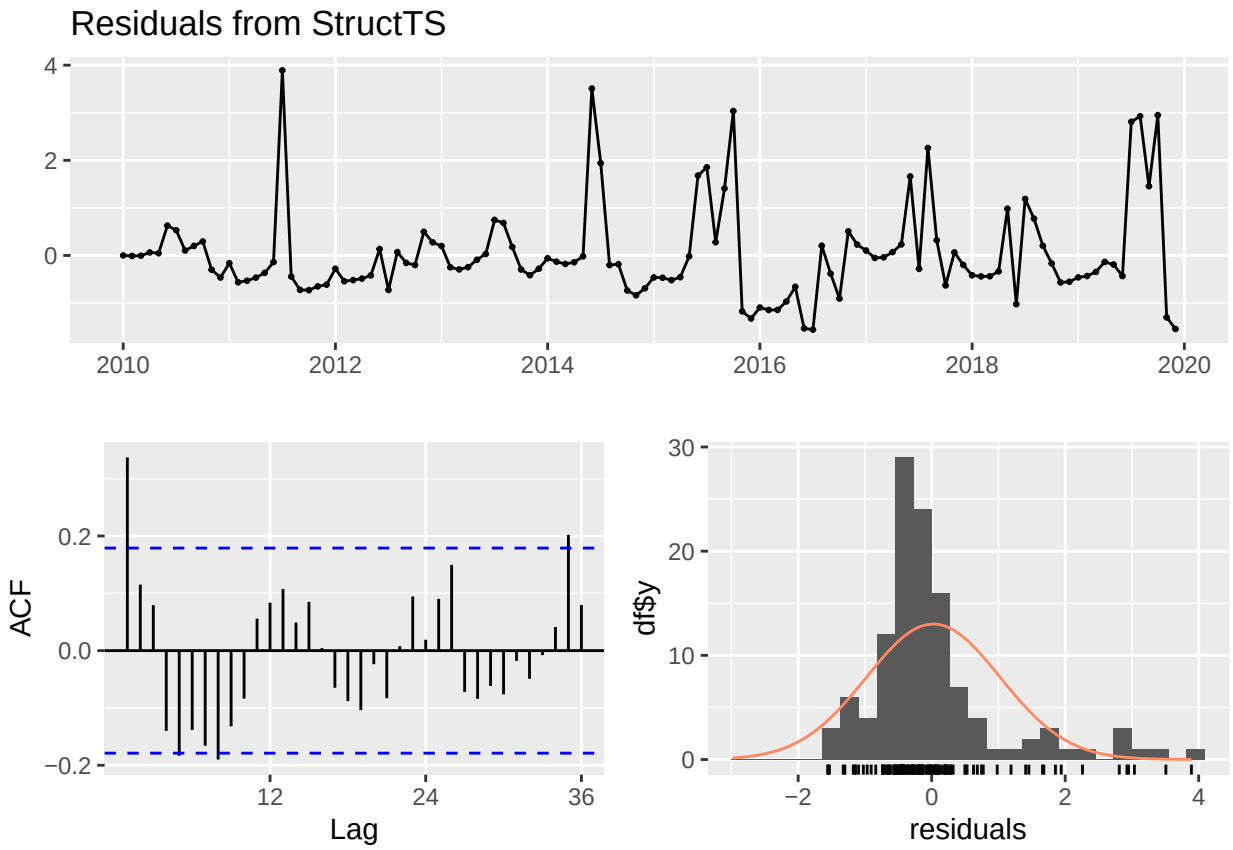

```
checkresiduals(SSES_seas)
```

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

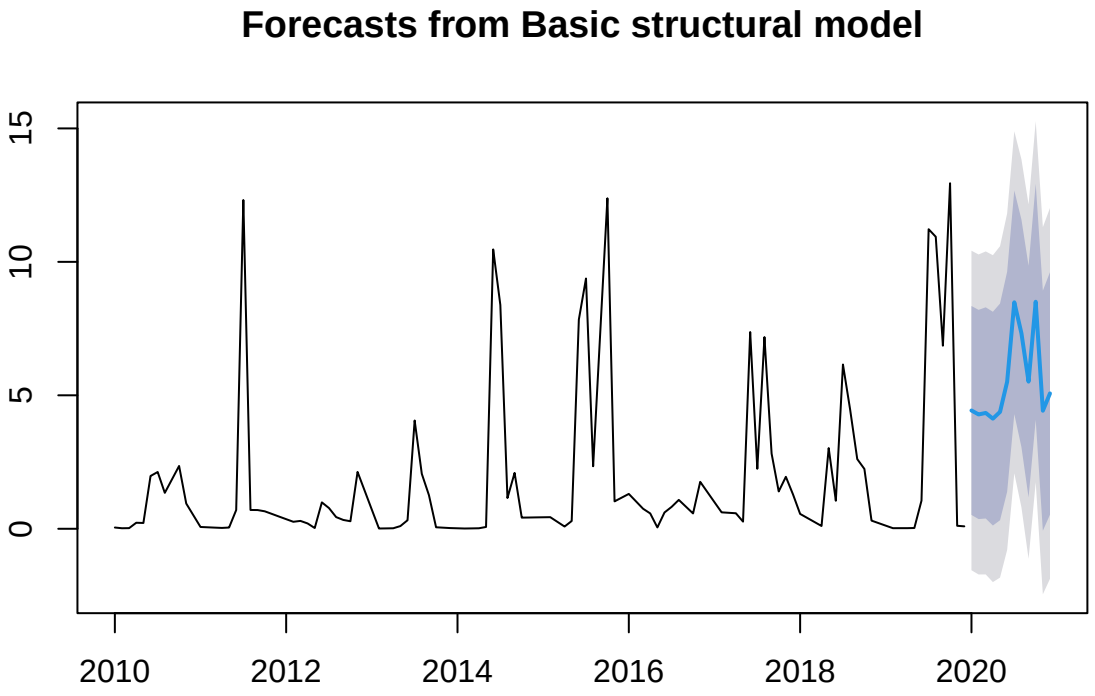



```
# Model 5: SS with StructTS()

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,]))
```

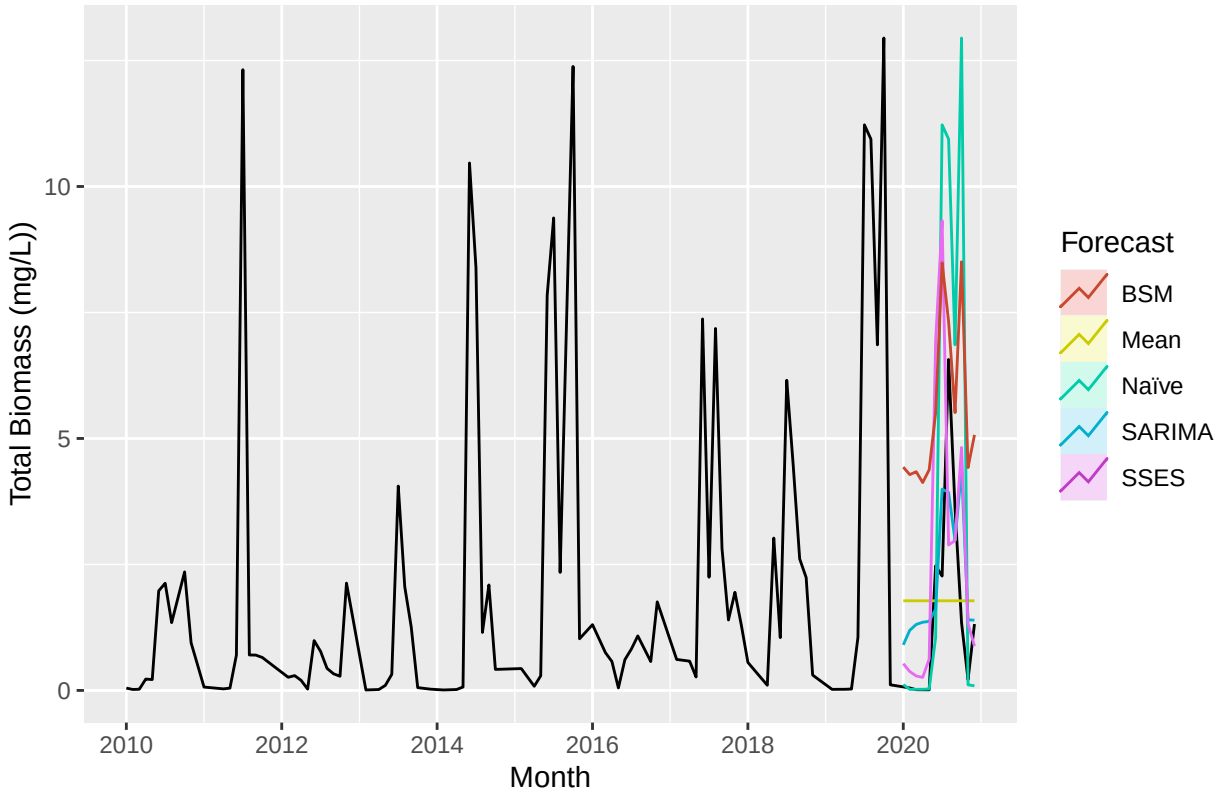
```
## The best model by RMSE is: SARIMA
```

Table 6: Forecast Accuracy for Seasonal Data

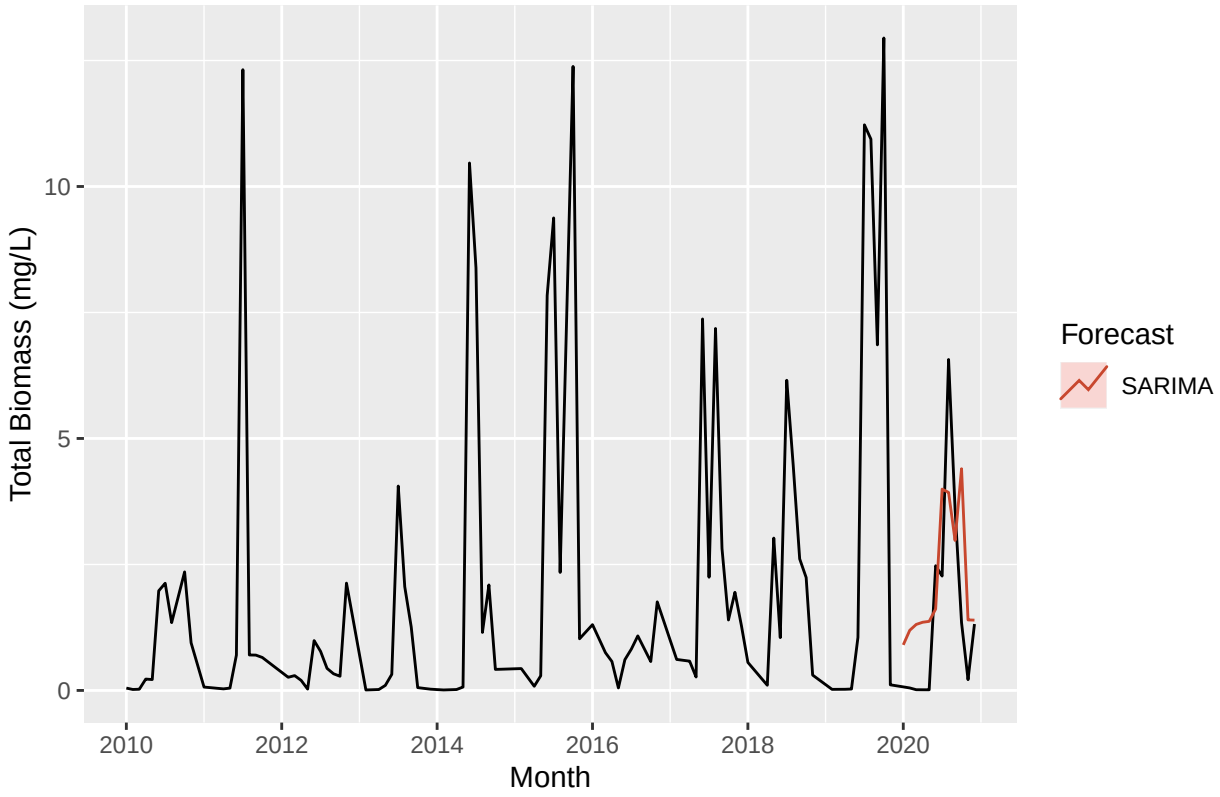
	ME	RMSE	MAE	MPE	MAPE
MEAN	-0.28323	1.93838	1.58284	-3817.5199	3846.3960
SNAIVE	-2.12360	4.54636	2.58677	-122.4177	164.9659
SARIMA	-0.65751	1.55783	1.34461	-2824.6163	2839.9915
SSES	-1.10662	2.85708	1.89745	-919.7008	937.4871
BSM	-4.03626	4.34690	4.03626	-9410.6202	9410.6202

```
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"]))
```

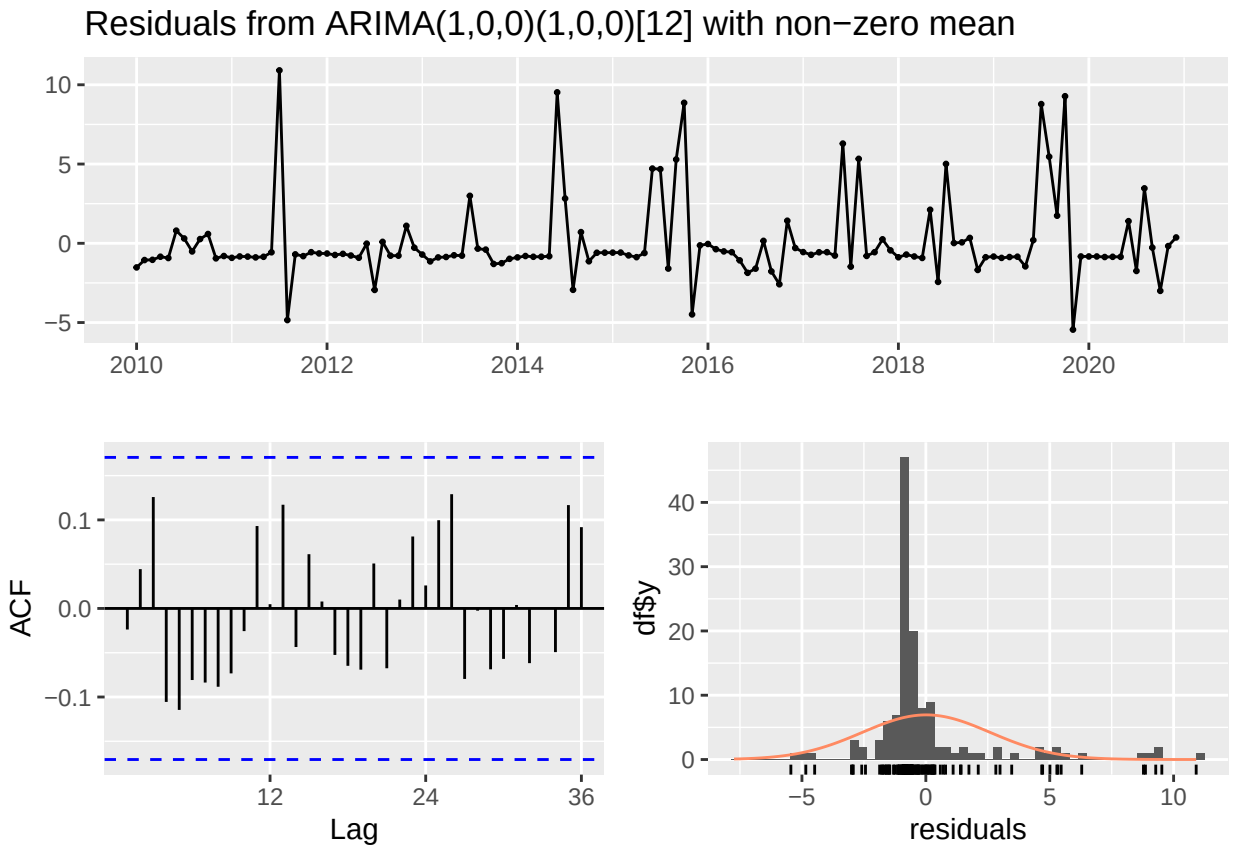
```
autoplot(ts_biomass_data) +
  autolayer(MEAN_seas, PI=FALSE, series="Mean") +
  autolayer(SNAIVE_seas, PI=FALSE, series="Naïve") +
  autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  autolayer(SSES_seas$forecast, series="SSES") +
  autolayer(SS_for,PI=FALSE,series="BSM") +
  xlab("Month") + ylab("Total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




```
autoplot(ts_biomass_data) +

autolayer(SARIMA_for,PI=FALSE, series="SARIMA") +
  xlab("Month") + ylab("Total Biomass (mg/L)") +
  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)
```

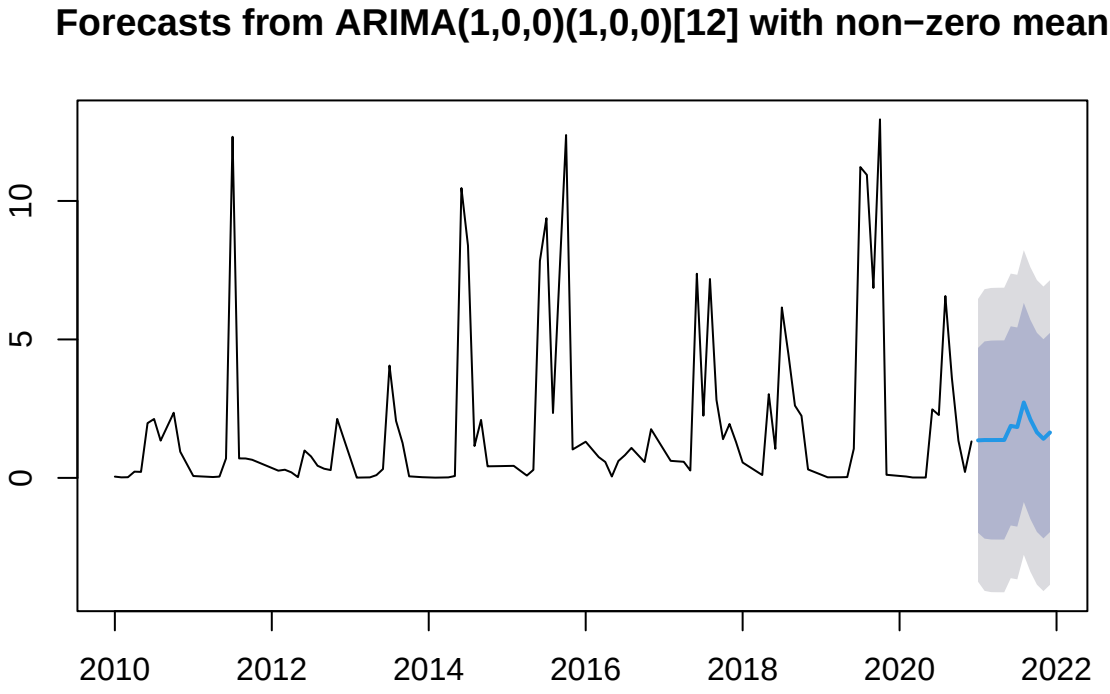
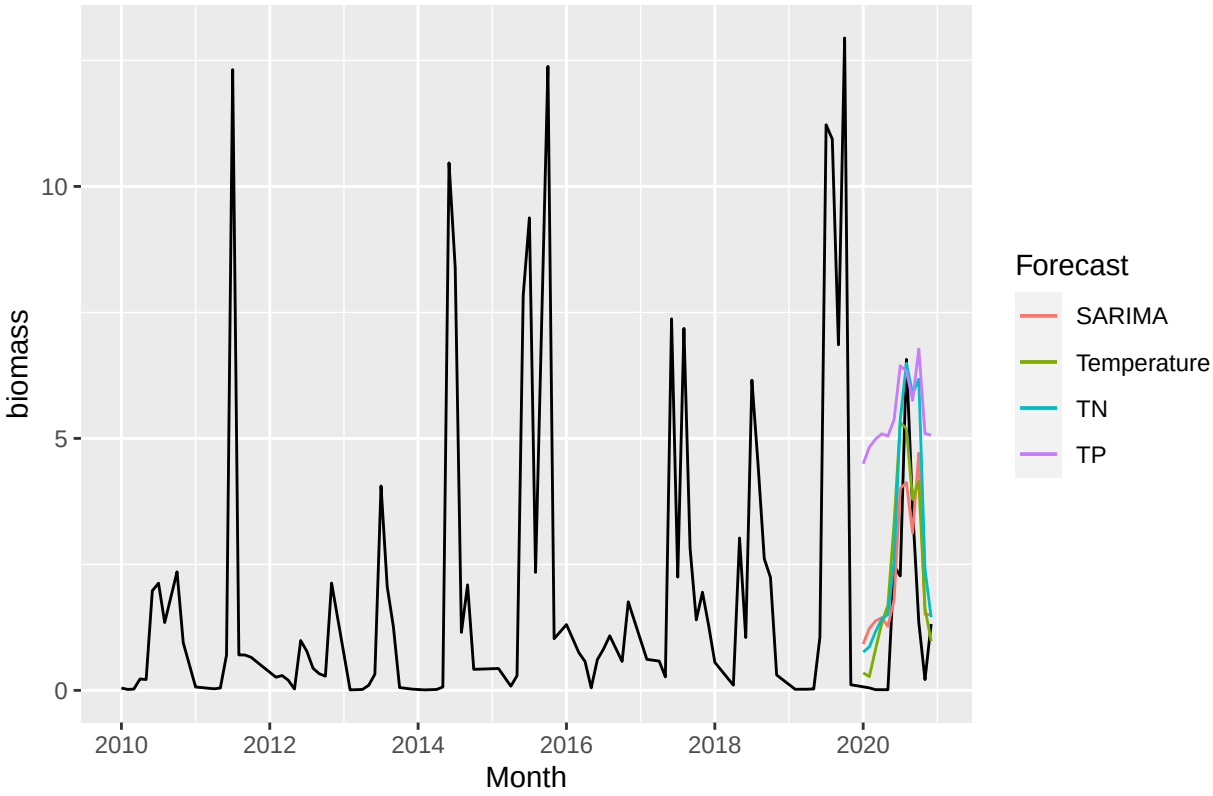

Forecasts from ARIMA(1,0,0)(1,0,0)[12] with non−zero mean
0
5
10
2010
2012
2014
2016
2018
2020
2022

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

Result2 Exogenous Variables: Temperature, TN and TP



Other than biomass itself, that there may be exogenous variables that also influence phytoplankton biomass, and incorporating these variables into the model could potentially improve our ability to forecast the phytoplankton biomass. Therefore, we searched the existing literature to identify key variables.

As shown in the table below, it appears that temperature, nitrogen, and phosphorus appear throughout the literature. In terms of the mechanism by which they affect phytoplankton biomass, the three variables fall into two groups: First, temperature affects the metabolic rates of organisms. As temperature increases, the metabolic rates of phytoplankton and their predators increase at different rates and reach their growth optima at different times. Second, phytoplankton growth is often limited by nutrient availability. Nitrogen (N) and phosphorus (P) are the two such nutrients, so increasing TN and TP could lead to increased phytoplankton growth rates and hence their biomass.

Table 8: Therefore, we added the three variables individually to the best performing model, i.e. SARIMA, to see if the model performance would be improved accordingly. Table 7 shows the forecast accuracy of adding each explanatory variables, where the first row shows the accuracy of the original SARIMA model, and the following rows show the accuracy of the new models with one exogenous variable added.

Author	Year	Key findings
Yuan and Pollard (Yuan, 2018)	2018	The mean biomass of zooplankton individuals was best predicted by physical lake characteristics such as geographic location, mean annual temperature, affecting the relationship between zooplankton (Z) and phytoplankton (P) biomasses.
Borics et al. (Borics, 2013)	2013	By incorporating lake depth, TP, TN and lake use as independent and Chl-a as dependent variables into different models, the predictive models explained 50% of the variance.
Cai et al. (Cai, 2012)	2012	Multiple stepwise linear regression revealed that EAWT, dissolved total phosphorus (DTP), and TP explained 99.2% of the variation of Chl-a in Meiliang Bay.
Burgmer and Hillebrand (Burgmer, 2011)	2011	Altered temperature regimes strongly affected algal biomass and diversity by interdependently altering competitive and consumer interactions.

According to Table 7, in terms of Mean Error (ME), none of the three new models improved the performance, but worsened it. On the other hand, adding temperature helped to slightly improve the model's accuracy in all other metrics (RMSE, MAE, MPE, and MAPE), and adding TN slightly improved the model's accuracy in terms of MPE and MAPE.

Figure X shows forecast results of different models. Consistent with the accuracy metrics, temperature and TN improved the performance of the original SARIMA model, especially its ability to forecast the peak in 2020. Also, adding temperature improved the prediction of low values. In contrast, adding TP significantly worsened the model's overall accuracy, as it only focuses on the peaks.

Discussion

Of the five different models we tried, the SARIMA model performed best in terms of accuracy. To further explore how exogenous variables could improve the performance of the model, we tried adding temperature, TN, and TP to the model, and temperature and TN slightly improved the performance of the original model. Specifically, adding temperature improved the original model's ability to handle extreme variation, as reflected by a lower RMSE.

However, all of the models we examined have relatively high (absolute value) MPE and MAPE. A possible explanation could be that many of the actual values of phytoplankton biomass are close to zero. As a result, the MPE and MAPE can go very high as they calculate the relative error of a near-zero actual value.

There are several outstanding limitations of our study. First, we only added one exogenous variable at a time, ignoring the combination of different exogenous variables could produce a better forecast. Second, the data of the exogenous variables we used in the forecast is the actual observed data rather than the forecast from their historical level. A major obstacle here was that our exogenous variable contained too many NAs which made it difficult for any model to forecast.

Therefore, the improvement from adding exogenous variables could be significantly smaller than our results, given the difficulty in accurately forecasting these exogenous variables. From our perspective, given that improvement is still quite subtle even using the real observed data, sticking to auto-regression may be a simpler and more economical option in real-world forecasting.

Reference

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.

Borics, L. M., Gábor; Nagy. (2013). Which factors affect phytoplankton biomass in shallow eutrophic lakes? *Hydrobiologia*, *714*(1). https://doi.org/10.1007/s10750-013-1525-6

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

Burgmer, H., Tanja; Hillebrand. (2011). Temperature mean and variance alter phytoplankton biomass and biodiversity in a long-term microcosm experiment. *Oikos*, *120*, 922–933.

Cai, G. Z., Lin-lin; Zhu. (2012). Effects of temperature and nutrients on phytoplankton biomass during bloom seasons in taihu lake. *Water Science and Engineering*, *5*(4), 361–374.

Chakraborty, P. S., Subhendu; Tiwari. (2017). Effects of fertilizers used in agricultural fields on algal blooms. *The European Physical Journal Special Topics*, *226*(1), 2119–2133.

Joyce, S. (2000). The dead zones: Oxygen-starved coastal waters. *Environmental Health Perspectives*, *108*, A120–A125.

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

Treuer, C. L., Galen; Kirchhoff. (2021). Challenges of managing harmful algal blooms in US drinking water systems. *Nature Sustainability*, *4*, 958–964.

Yuan, A. I., Lester L.; Pollard. (2018). Changes in the relationship between zooplankton and phytoplankton biomasses across a eutrophication gradient. *Limnology and Oceanography*, *63*(6), 2493–2507.