TSA

Yuxiang Ren

#packges

```
library(readxl)
#library(xlsx) #this package doesn't work on Macs very well
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(Kendall)
library(tseries)
library(outliers)

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v tibble  3.1.8     v dplyr   1.1.0
## v readr   2.1.4     v stringr 1.5.0
## v purrr   1.0.1     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(smooth)
```

```
## Loading required package: greybox
## Package "greybox", v1.0.7 loaded.
##
##
## Attaching package: 'greybox'
##
## The following object is masked from 'package:lubridate':
##
##     hm
##
## The following object is masked from 'package:tidyr':
##
##     spread
##
## This is package "smooth", v3.2.0
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(dplyr)
library(smooth)
```

#data

```
raw <- read.csv("./Data/Processed/PM2.5_daily_city_2000_2021.csv")
Table_raw <- raw[c(1:20),]
kable(Table_raw,caption = "Raw data")
```

```
raw_wider <- raw %>% pivot_wider(names_from = city_id, values_from = meanpm) %>%
  mutate(date = as.Date(paste(year, month, day, sep = "-"))) %>%
  select(date, everything())
```

#Beijing 1100 ##data

Table 1: Raw data

year	month	day	city_id	meanpm
2000	1	1	1100	130.86416
2000	1	1	1200	148.09160
2000	1	1	1301	144.00916
2000	1	1	1302	148.45776
2000	1	1	1303	130.70928
2000	1	1	1304	143.41543
2000	1	1	1305	144.08262
2000	1	1	1306	118.38928
2000	1	1	1307	66.89358
2000	1	1	1308	92.74160
2000	1	1	1309	134.39611
2000	1	1	1310	147.80079
2000	1	1	1311	142.80690
2000	1	1	1401	64.93789
2000	1	1	1402	54.55570
2000	1	1	1403	75.32231
2000	1	1	1404	90.57612
2000	1	1	1405	95.47729
2000	1	1	1406	48.40086
2000	1	1	1407	78.33466

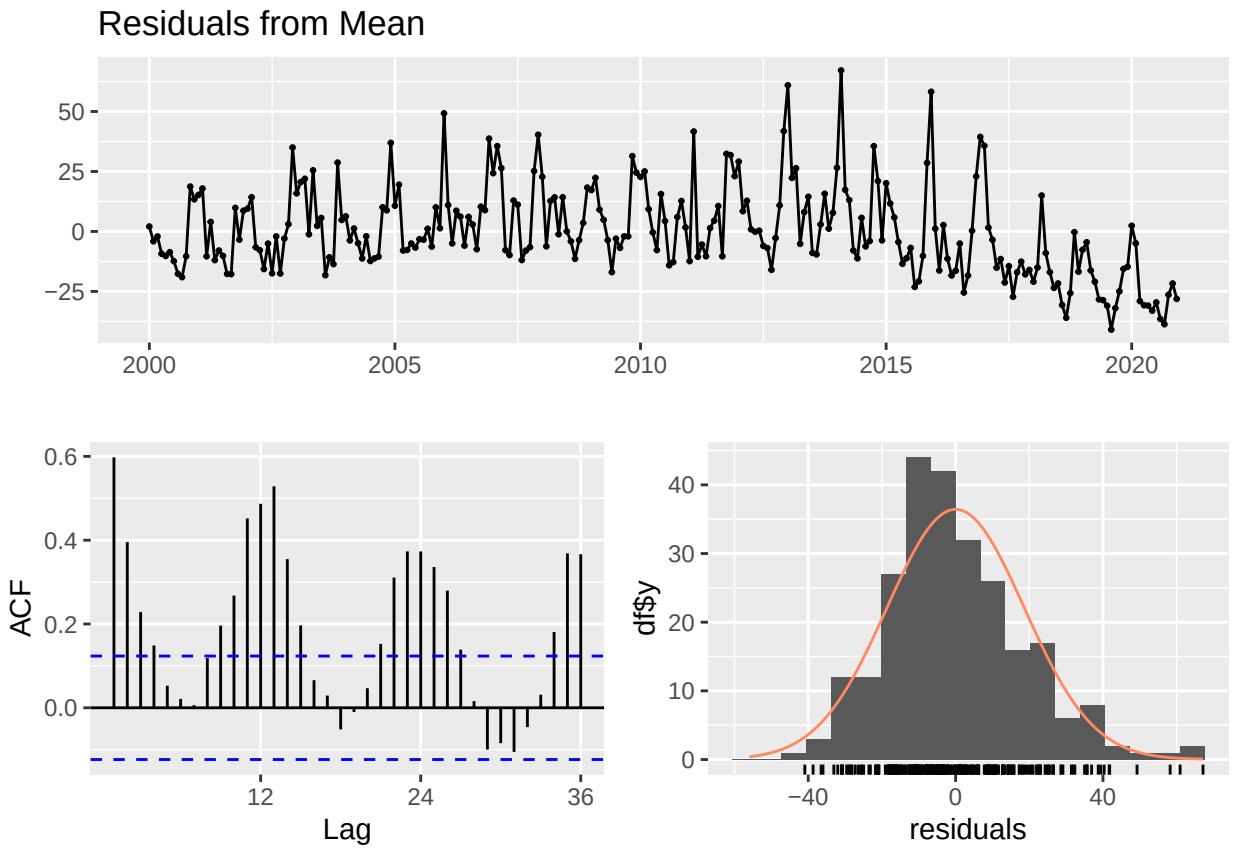
```
Beijing <- raw_wider[,c('date', 'year', 'month', 'day', '1100')] %>% rename(pm2.5 = '1100') %>% group_by
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
#question, mean PM or accumulated PM

Beijing_train <- Beijing %>% filter(date < as.Date("2021-01-01"))
Beijing_test <- Beijing %>% filter(date >= as.Date("2021-01-01"))

ts_Beijing_train <- ts(Beijing_train$monthlyPM,
                       start=c(2000,1), frequency = 12)
ts_Beijing_all <- ts(Beijing$monthlyPM,
                       start=c(2000,1), frequency = 12)
```

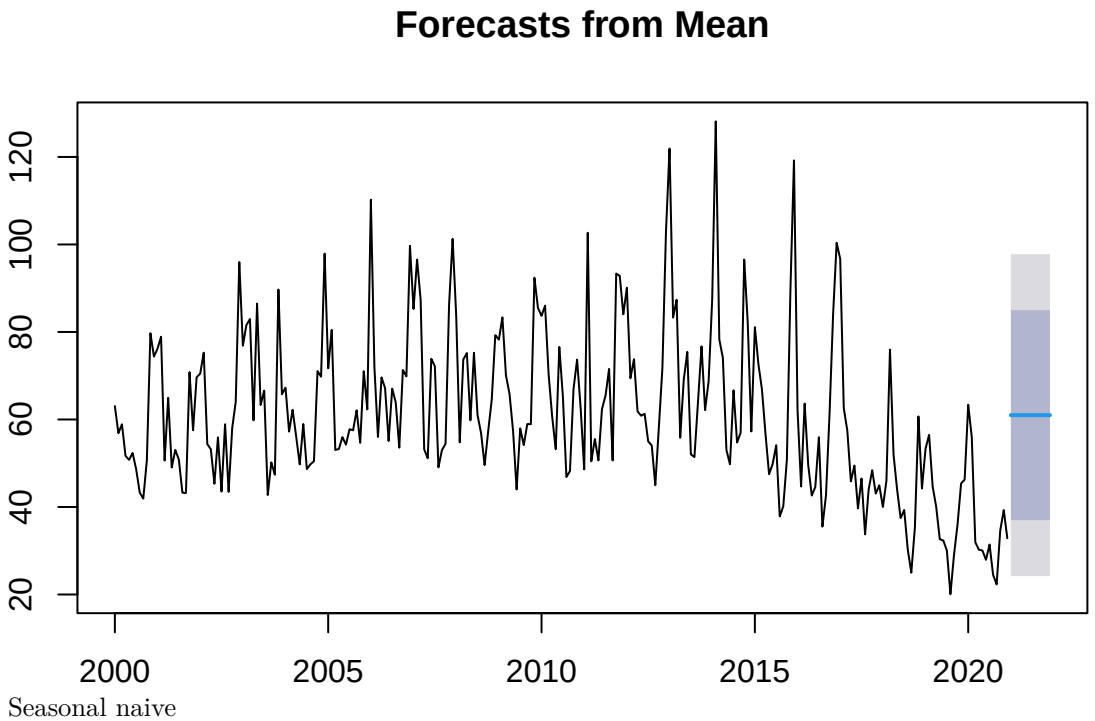
##Model 1: Arithmetic mean

```
AM_Beijing <- meanf(y = ts_Beijing_train, h = 12)
checkresiduals(AM_Beijing)
```



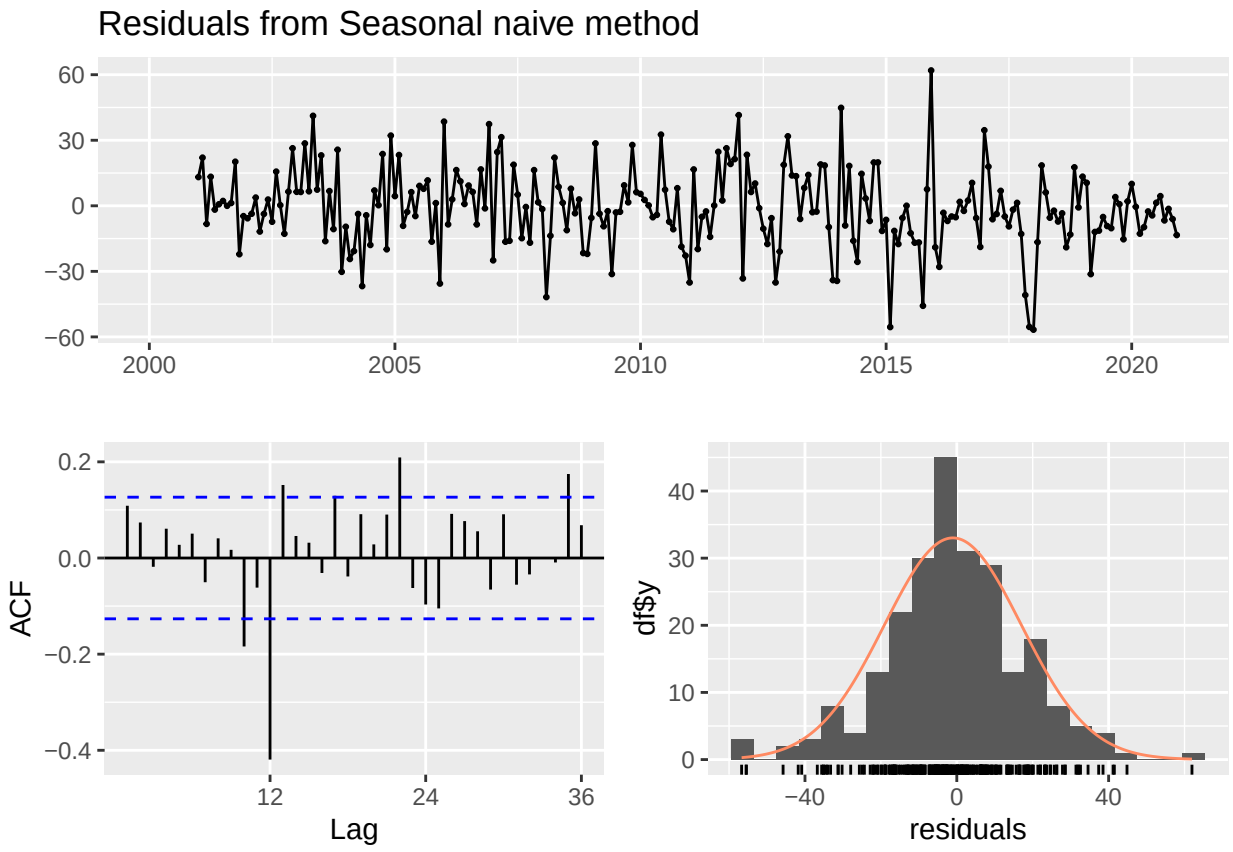

```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 534.57, df = 23, p-value < 2.2e-16
##
## Model df: 1.   Total lags used: 24
```

```
plot(AM_Beijing)
```



Model 2: Seasonal naive

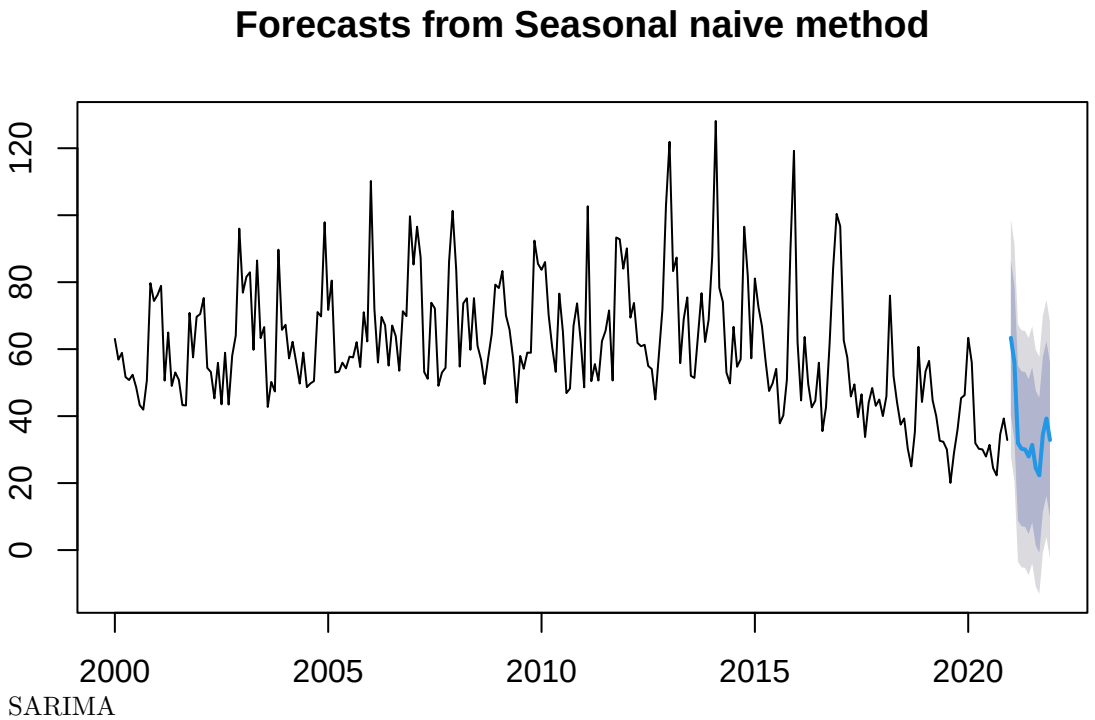
```
SNAIVE_Beijing <- snaive(ts_Beijing_train, h=12, holdout=FALSE)
checkresiduals(SNAIVE_Beijing)
```



##

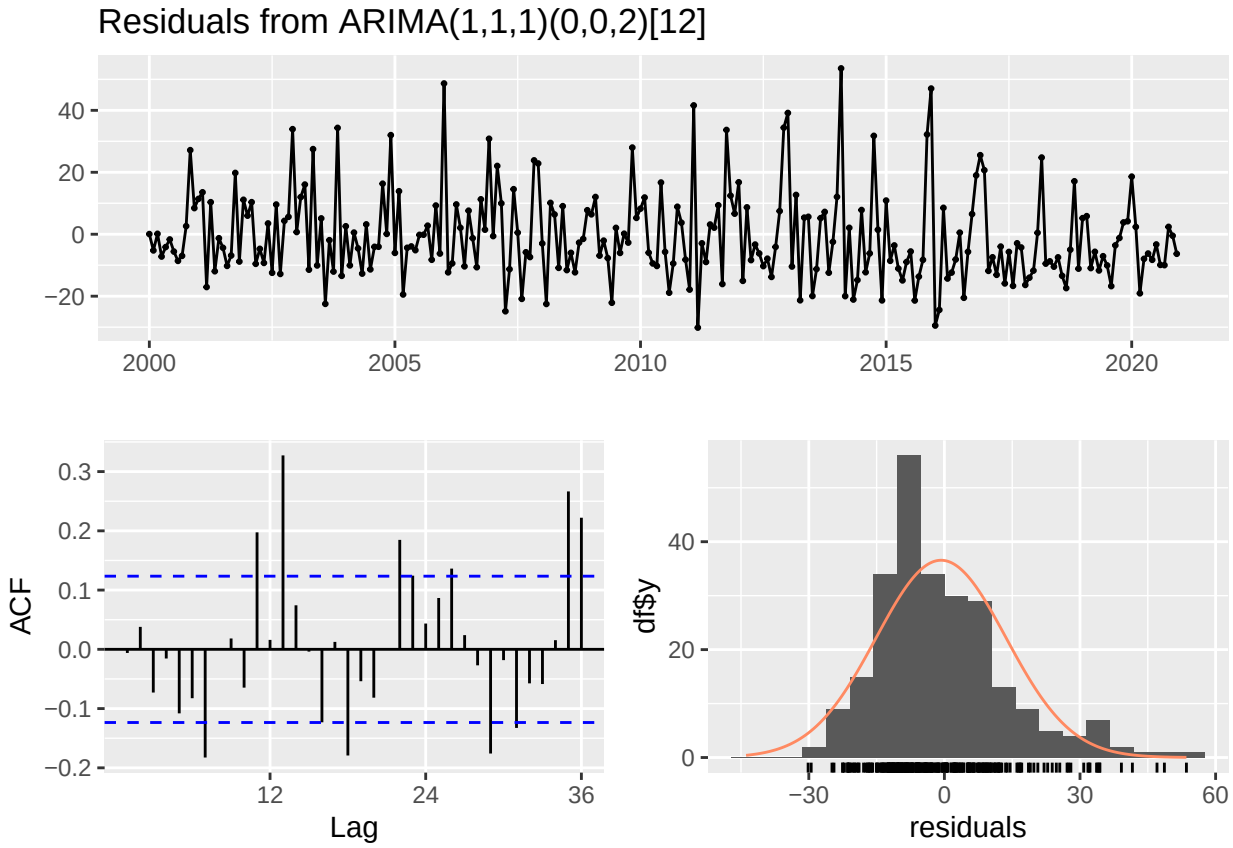
```
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 92.915, df = 24, p-value = 4.72e-10
##
## Model df: 0.   Total lags used: 24
```

```
plot(SNAIVE_Beijing)
```



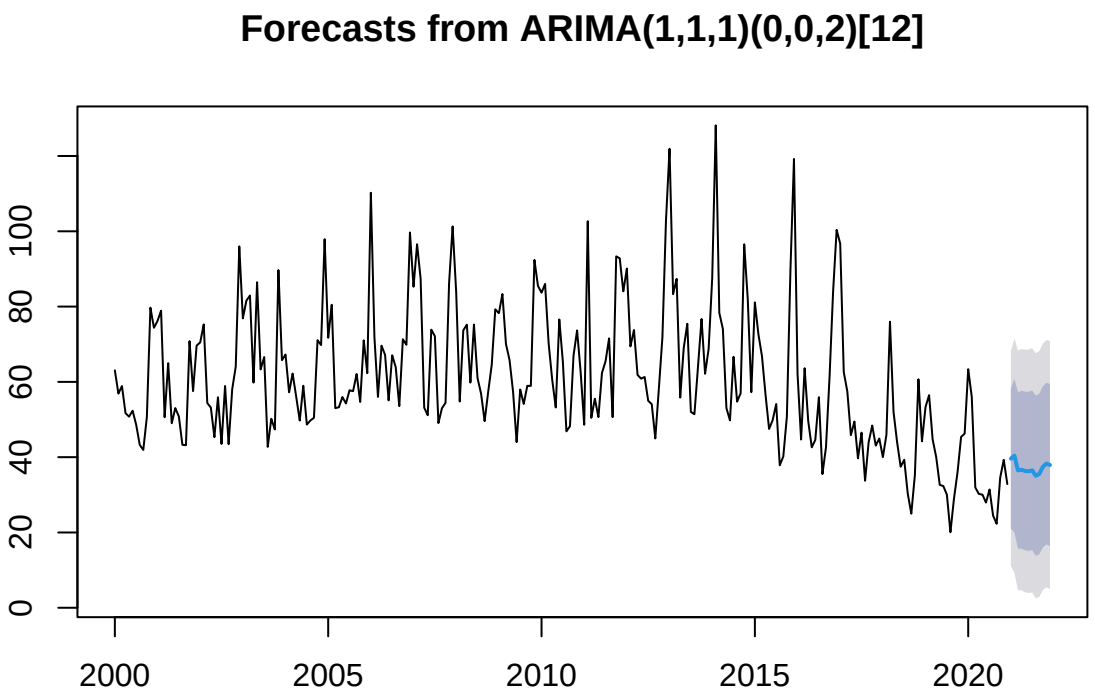
Model 3: SARIMA

```
SARIMA_Beijing_fit <- auto.arima(ts_Beijing_train)
checkresiduals(SARIMA_Beijing_fit)
```



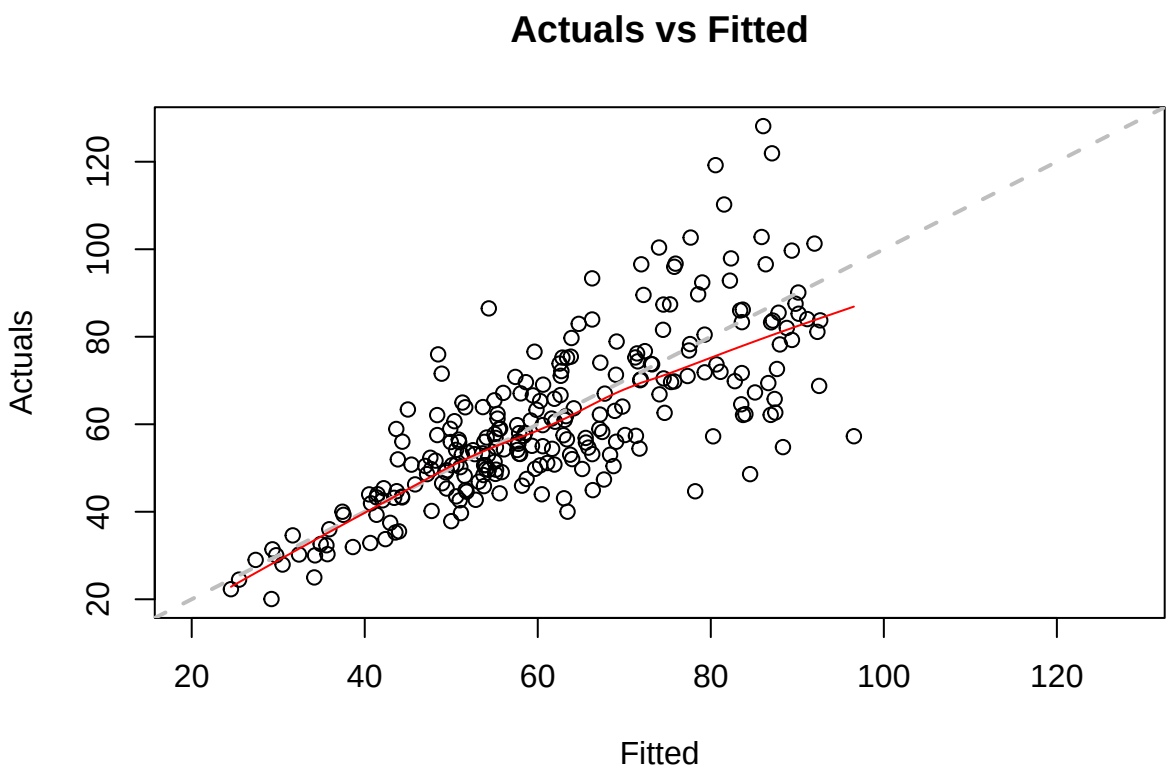

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,0,2)[12]
## Q* = 87.011, df = 20, p-value = 2.458e-10
##
## Model df: 4.   Total lags used: 24
```

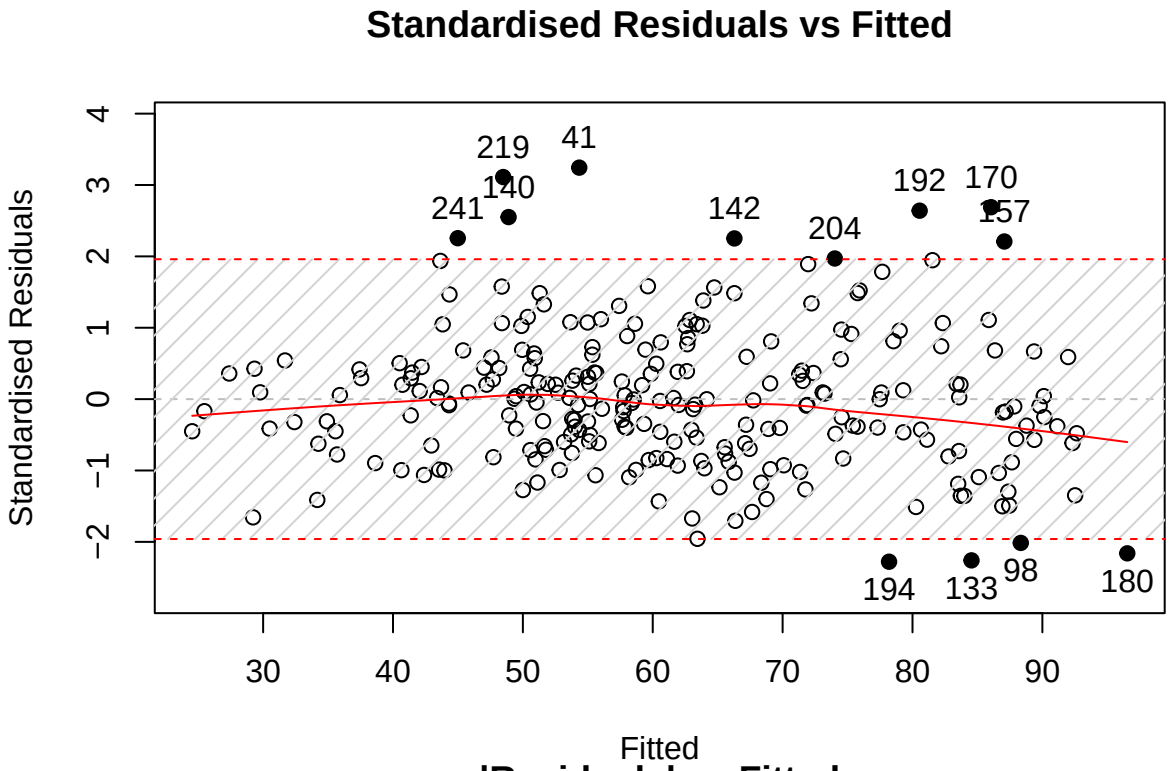
```
#Generating forecasts
#remember auto.arima does not call the forecast() internally so we need one more step
SARIMA_Beijing <- forecast(SARIMA_Beijing_fit,h=12)
plot(SARIMA_Beijing)
```

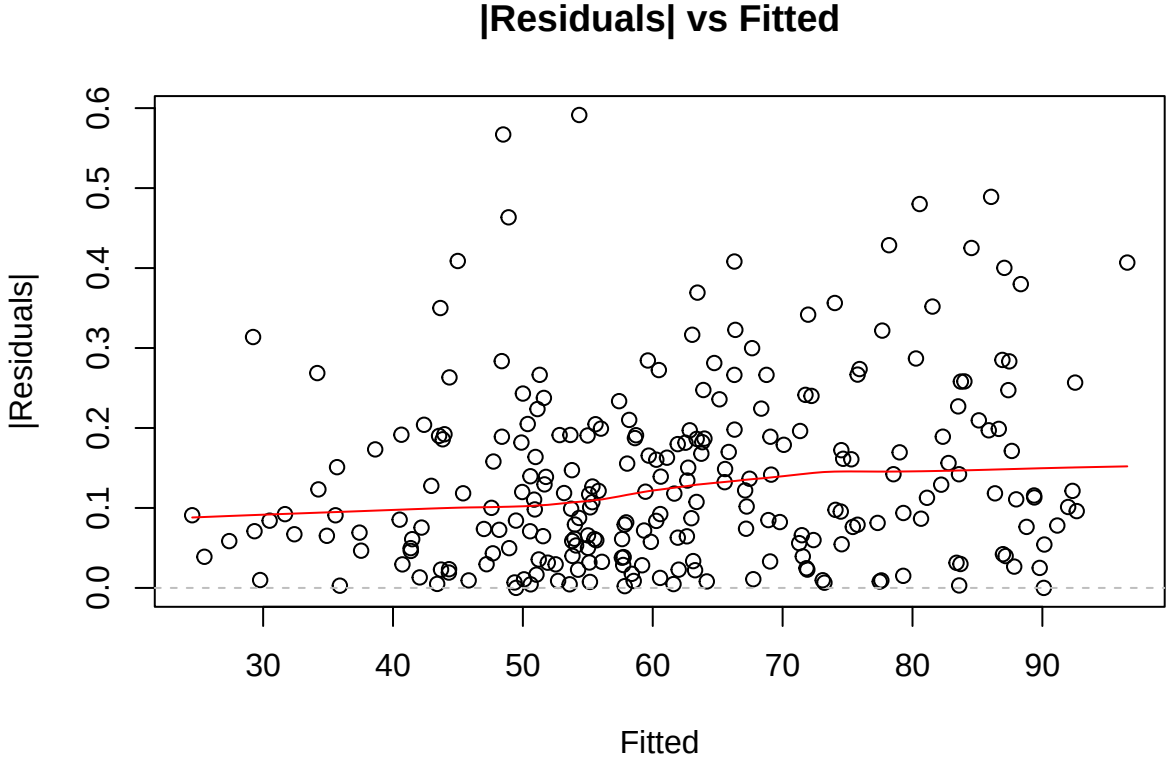


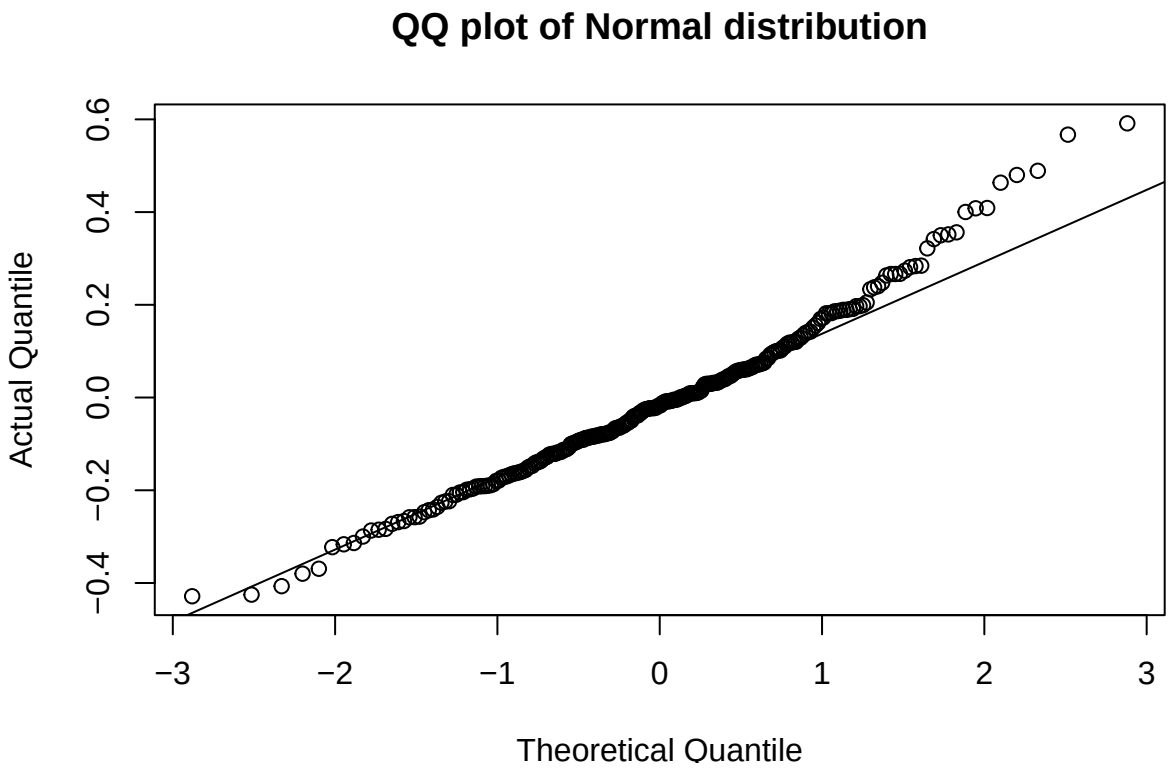
Model 4: Fitting State Space Models to the original (seasonal) series

```
SSES_Beijing <- es(ts_Beijing_train,model="ZZZ",h=12,holdout=FALSE)
plot(SSES_Beijing)
```




Standardised Residuals vs Fitted
Standardised Residuals
Fitted
219
41
140
241
142
204
192
170
157
194
133
98
180


|Residuals| vs Fitted
|Residuals|
Fitted

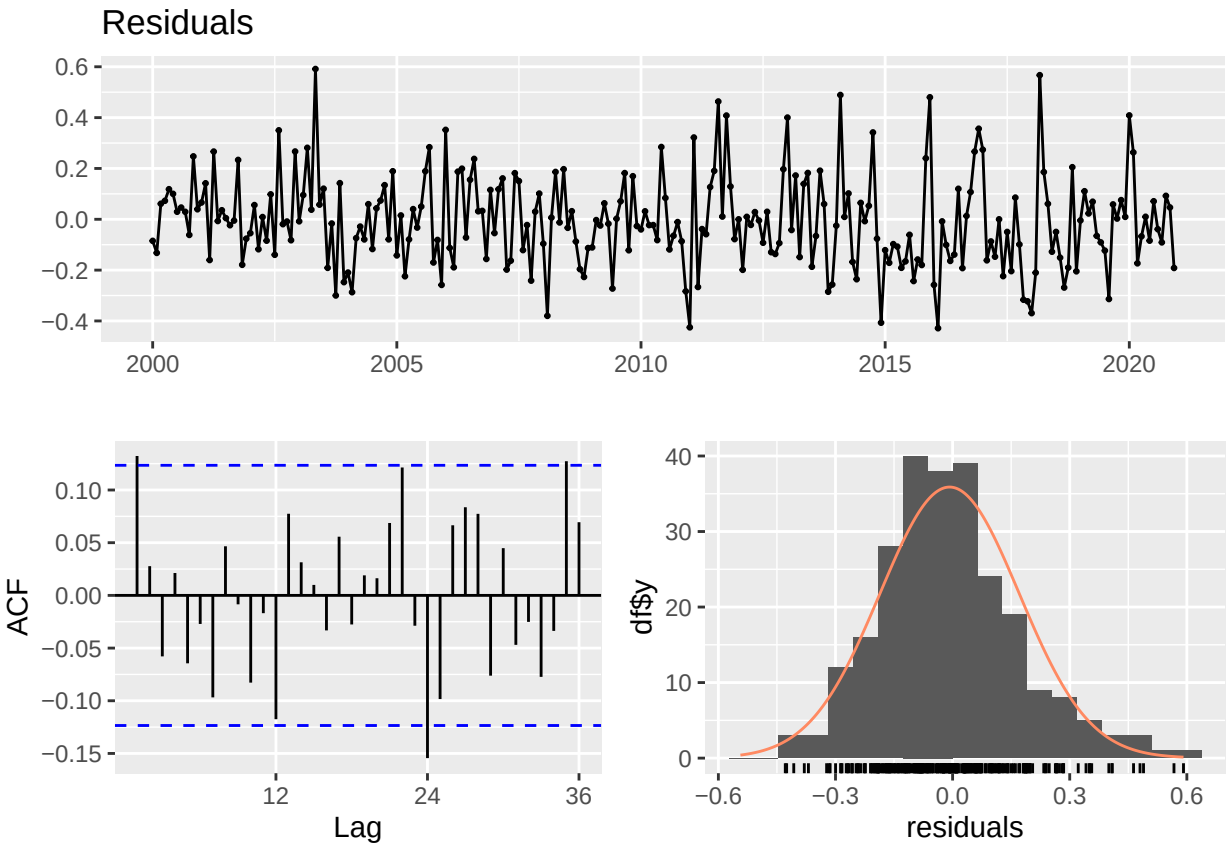



```
checkresiduals(SSES_Beijing)
```

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

Table 2: Forecast Accuracy for Seasonal Data

	ME	RMSE	MAE	MPE	MAPE
MEAN	-28.1059	31.6898	29.2560	-119.0707	120.7650
SNAIVE	-2.4987	13.4377	9.3796	-17.3618	28.6924
SARIMA	-4.3333	14.5863	12.2154	-32.3573	45.6127
SSES	4.8008	12.3008	7.7936	5.2446	19.6416



##accuacy

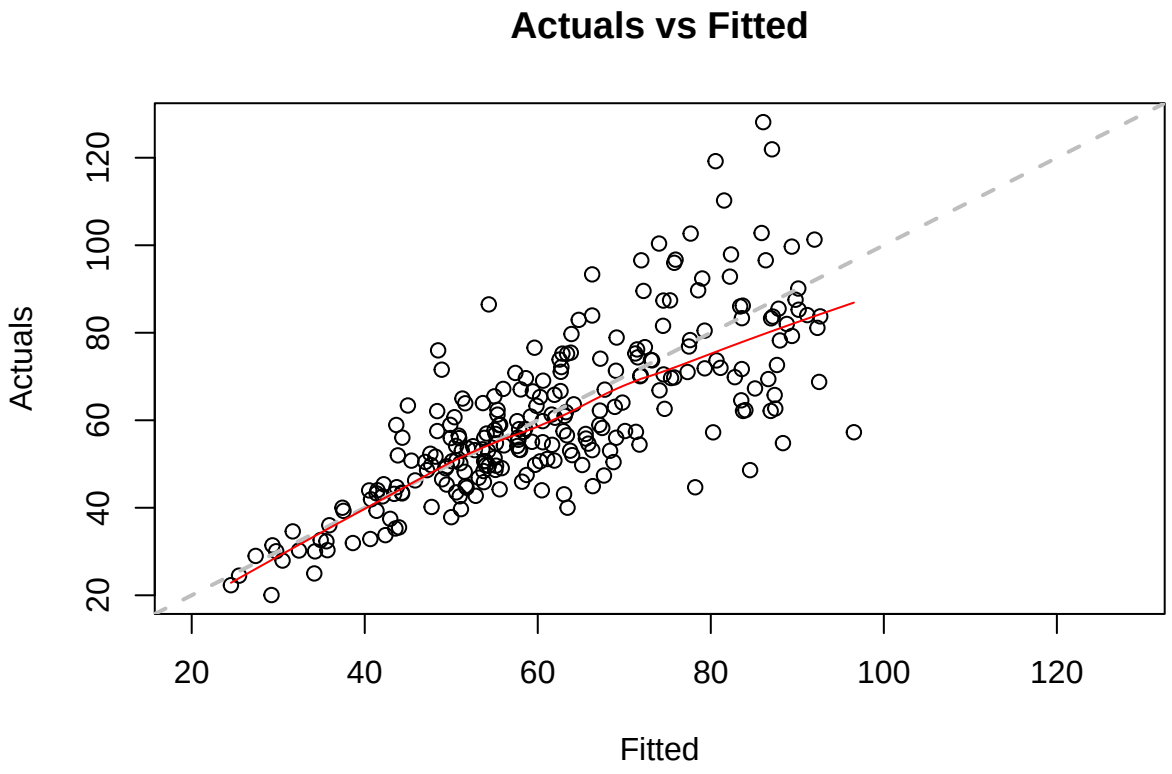
```
MEAN_scores <- accuracy(AM_Beijing$mean,Beijing_test$monthlyPM)
SNAIVE_scores <- accuracy(SNAIVE_Beijing$mean,Beijing_test$monthlyPM)
SARIMA_scores <- accuracy(SARIMA_Beijing$mean,Beijing_test$monthlyPM)
SSES_scores <- accuracy(SSES_Beijing$forecast,Beijing_test$monthlyPM)

Beijing_scores <- as.data.frame(rbind(MEAN_scores, SNAIVE_scores, SARIMA_scores,SSES_scores))
row.names(Beijing_scores) <- c("MEAN", "SNAIVE","SARIMA","SSES")

kbl(Beijing_scores,
      caption = "Forecast Accuracy for Seasonal Data",
      digits = array(4,ncol(Beijing_scores))) %>%
  kable_styling(full_width = FALSE, position = "center") %>%
  #highlight model with lowest RMSE
  kable_styling(latex_options="striped", stripe_index = which.min(Beijing_scores[,"RMSE"]))
```

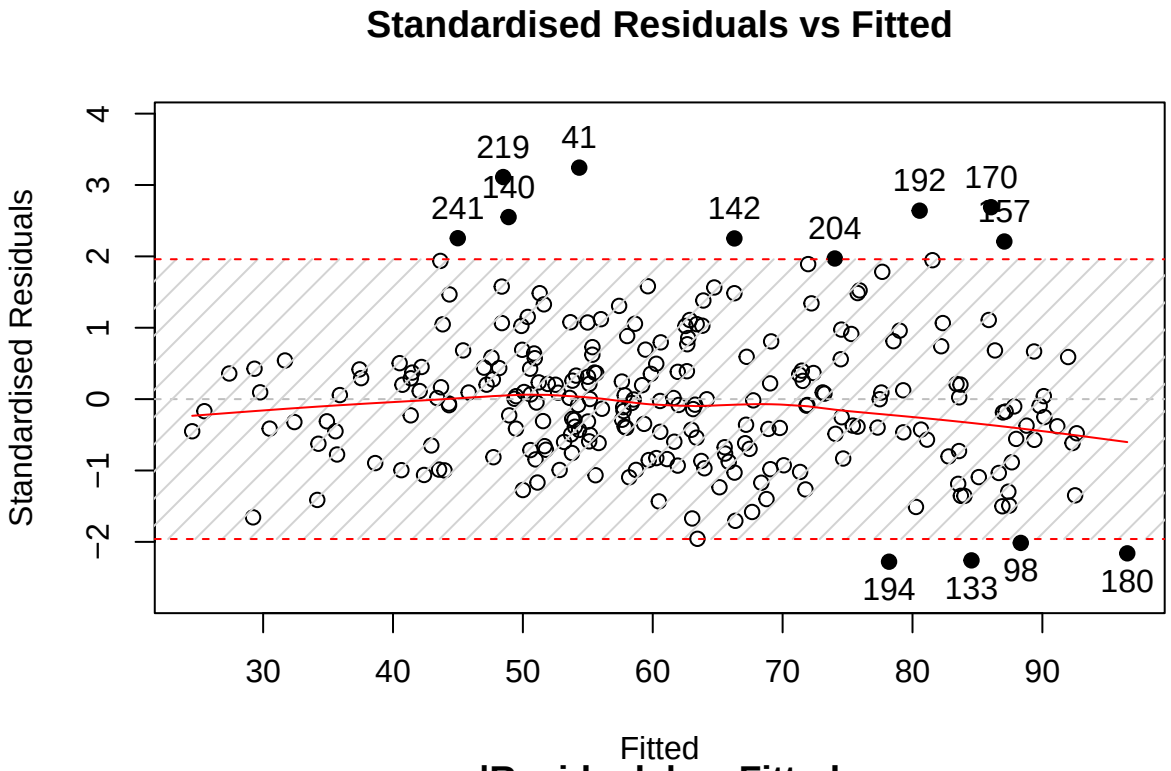
```
best_model_index <- which.min(Beijing_scores[,"RMSE"])
cat("The best model by RMSE is:", row.names(Beijing_scores[best_model_index,]))
```

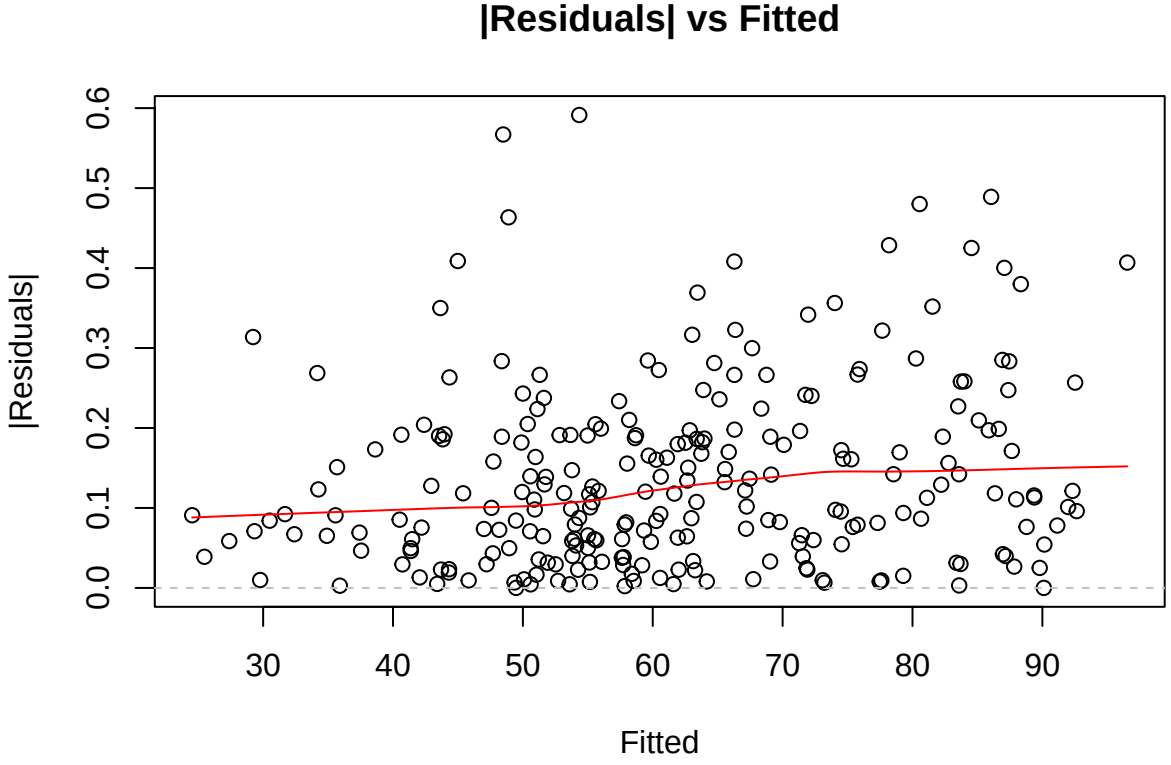
```
## The best model by RMSE is: SSES
```

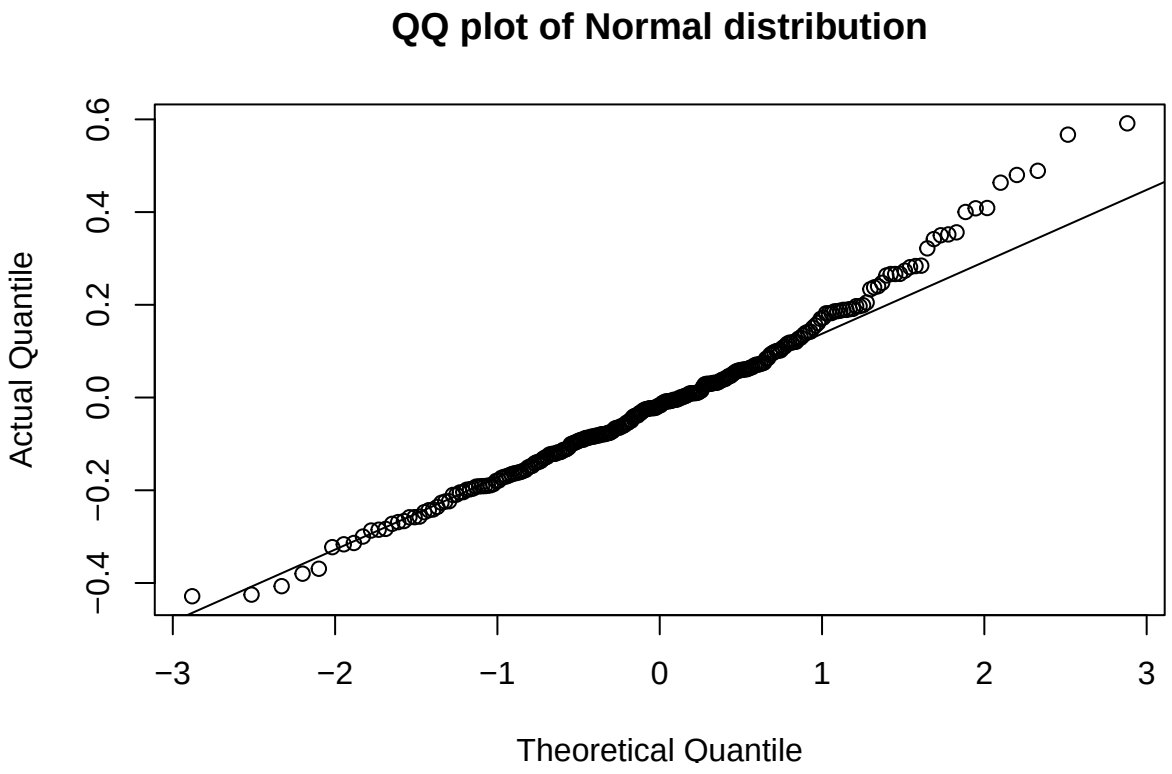
##Final result-2023 ###SSES model, two year result are same

```
for_Beijing_result <- es(ts_Beijing_all,model="ZZZ",h=24,holdout=FALSE)
plot(SSES_Beijing)
```




Standardised Residuals vs Fitted
Standardised Residuals
Fitted
219
41
140
241
142
204
192
170
157
194
133
98
180

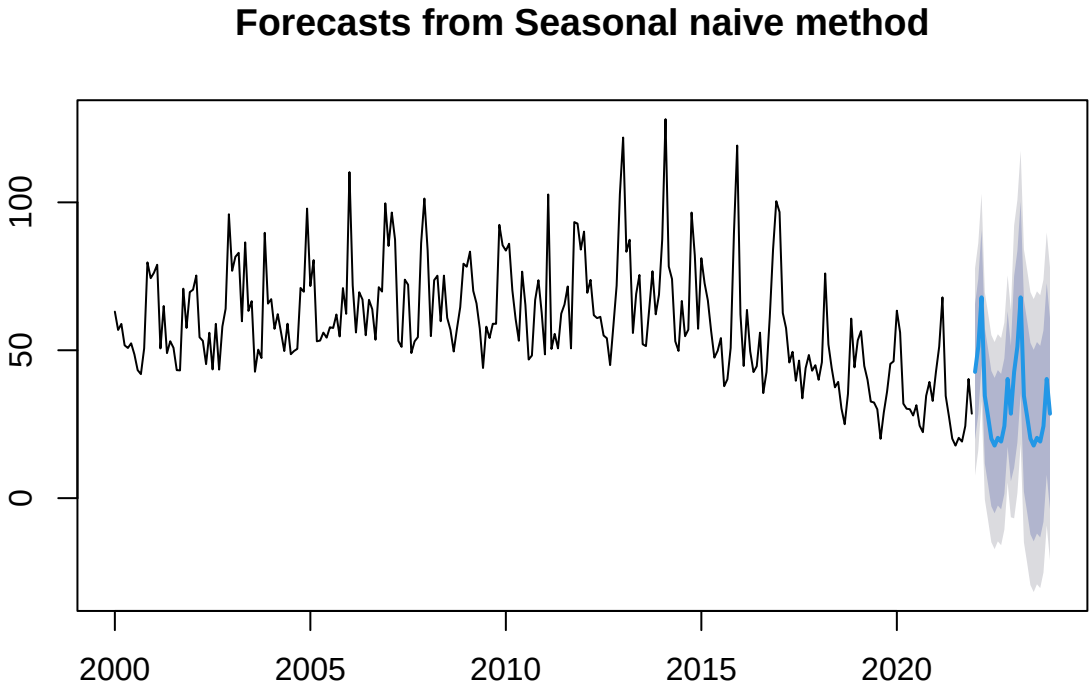

|Residuals| vs Fitted
|Residuals|
Fitted




```
forecast_Beijing <- for_Beijing_result$forecast
```

###SNAIVE same two years

```
for_Beijing_result2 <- snaive(ts_Beijing_all, h=24, holdout=FALSE)
plot(for_Beijing_result2)
```



```
for_Beijing_result2$mean
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2022 42.68474 51.23041 67.88502 34.56747 27.54109 20.13027 17.76858 20.40289
## 2023 42.68474 51.23041 67.88502 34.56747 27.54109 20.13027 17.76858 20.40289
##           Sep      Oct      Nov      Dec
## 2022 19.11793 24.37159 40.29166 28.54776
## 2023 19.11793 24.37159 40.29166 28.54776
```