project_models_analysis

2025-03-27

```
elnino_df <- read.csv("elnino.csv")
```

```
elnino_df$Zonal.Winds <- as.numeric(ifelse(elnino_df$Zonal.Winds == ".", NA, elnino_df$Zonal.Winds))
elnino_df$Meridional.Winds <- as.numeric(ifelse(elnino_df$Meridional.Winds == ".", NA, elnino_df$Meridi
elnino_df$Humidity <- as.numeric(ifelse(elnino_df$Humidity == ".", NA, elnino_df$Humidity))
elnino_df$Air.Temp <- as.numeric(ifelse(elnino_df$Air.Temp == ".", NA, elnino_df$Air.Temp))
elnino_df$Sea.Surface.Temp <- as.numeric(ifelse(elnino_df$Sea.Surface.Temp == ".", NA, elnino_df$Sea.Su

tail(elnino_df)
```

```
##        Observation Year Month Day   Date Latitude Longitude Zonal.Winds
## 178075      178075   98     6  10 980610     8.95   -140.33        -6.8
## 178076      178076   98     6  11 980611     8.96   -140.33        -5.1
## 178077      178077   98     6  12 980612     8.96   -140.32        -4.3
## 178078      178078   98     6  13 980613     8.95   -140.34        -6.1
## 178079      178079   98     6  14 980614     8.96   -140.33        -4.9
## 178080      178080   98     6  15 980615     8.95   -140.33          NA
##        Meridional.Winds Humidity Air.Temp Sea.Surface.Temp
## 178075             -5.3     81.3    27.52            28.17
## 178076             -0.4     94.1    26.04            28.14
## 178077             -3.3     93.2    25.80            27.87
## 178078             -4.8     81.3    27.17            27.93
## 178079             -2.3     76.2    27.36            28.03
## 178080               NA       NA    27.09            28.09
```

```
summary(elnino_df)
```

```
##   Observation          Year          Month             Day
##  Min.   :     1   Min.   :80.0   Min.   : 1.000   Min.   : 1.00
##  1st Qu.: 44521   1st Qu.:92.0   1st Qu.: 4.000   1st Qu.: 8.00
##  Median : 89040   Median :94.0   Median : 6.000   Median :16.00
##  Mean   : 89040   Mean   :93.3   Mean   : 6.505   Mean   :15.72
##  3rd Qu.:133560   3rd Qu.:96.0   3rd Qu.:10.000   3rd Qu.:23.00
##  Max.   :178080   Max.   :98.0   Max.   :12.000   Max.   :31.00
##
##       Date           Latitude          Longitude         Zonal.Winds
##  Min.   :800307   Min.   :-8.8100   Min.   :-180.00   Min.   :-12.400
##  1st Qu.:920116   1st Qu.:-2.0100   1st Qu.:-154.95   1st Qu.: -5.800
##  Median :940601   Median : 0.0100   Median :-111.26   Median : -4.000
##  Mean   :933690   Mean   : 0.4736   Mean   : -54.03   Mean   : -3.305
##  3rd Qu.:960617   3rd Qu.: 4.9800   3rd Qu.: 147.01   3rd Qu.: -1.400
##  Max.   :980623   Max.   : 9.0500   Max.   : 171.08   Max.   : 14.300
##                                                       NA's   :25163
##  Meridional.Winds    Humidity        Air.Temp      Sea.Surface.Temp
##  Min.   :-11.60   Min.   :45.40   Min.   :17.05   Min.   :17.35
##  1st Qu.: -1.70   1st Qu.:77.70   1st Qu.:26.06   1st Qu.:26.77
```

```
##  Median :  0.30   Median :81.20   Median :27.34   Median :28.29
##  Mean   :  0.25   Mean   :81.24   Mean   :26.89   Mean   :27.71
##  3rd Qu.:  2.30   3rd Qu.:84.80   3rd Qu.:28.18   3rd Qu.:29.23
##  Max.   : 13.00   Max.   :99.90   Max.   :31.66   Max.   :31.26
##  NA's   :25162    NA's   :65761   NA's   :18237   NA's   :17007
```

```r
nrow(elnino_df)
```

```
## [1] 178080
```

```r
ncol(elnino_df)
```

```
## [1] 12
```

```r
colnames(elnino_df)
```

```
##  [1] "Observation"      "Year"             "Month"            "Day"
##  [5] "Date"             "Latitude"         "Longitude"        "Zonal.Winds"
##  [9] "Meridional.Winds" "Humidity"         "Air.Temp"         "Sea.Surface.Temp"
```

```r
#elnino_df[6000:8000, 1:8]
```

```r
library(ggplot2)
library(ggfortify)
library(fpp2)
```

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

```
## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
##   autoplot.ets           ggfortify
##   autoplot.forecast      ggfortify
##   autoplot.stl           ggfortify
##   autoplot.ts            ggfortify
##   fitted.ar              ggfortify
##   fortify.ts             ggfortify
##   residuals.ar           ggfortify
```

```
## -- Attaching packages ---------------------------------------------- fpp2 2.5 --
```

```
## v forecast  8.23.0     v expsmooth 2.3
## v fma       2.5
```

```
## Warning: package 'forecast' was built under R version 4.3.3
```

```
##
```

```r
library(astsa)
```

```
## Warning: package 'astsa' was built under R version 4.3.3
```

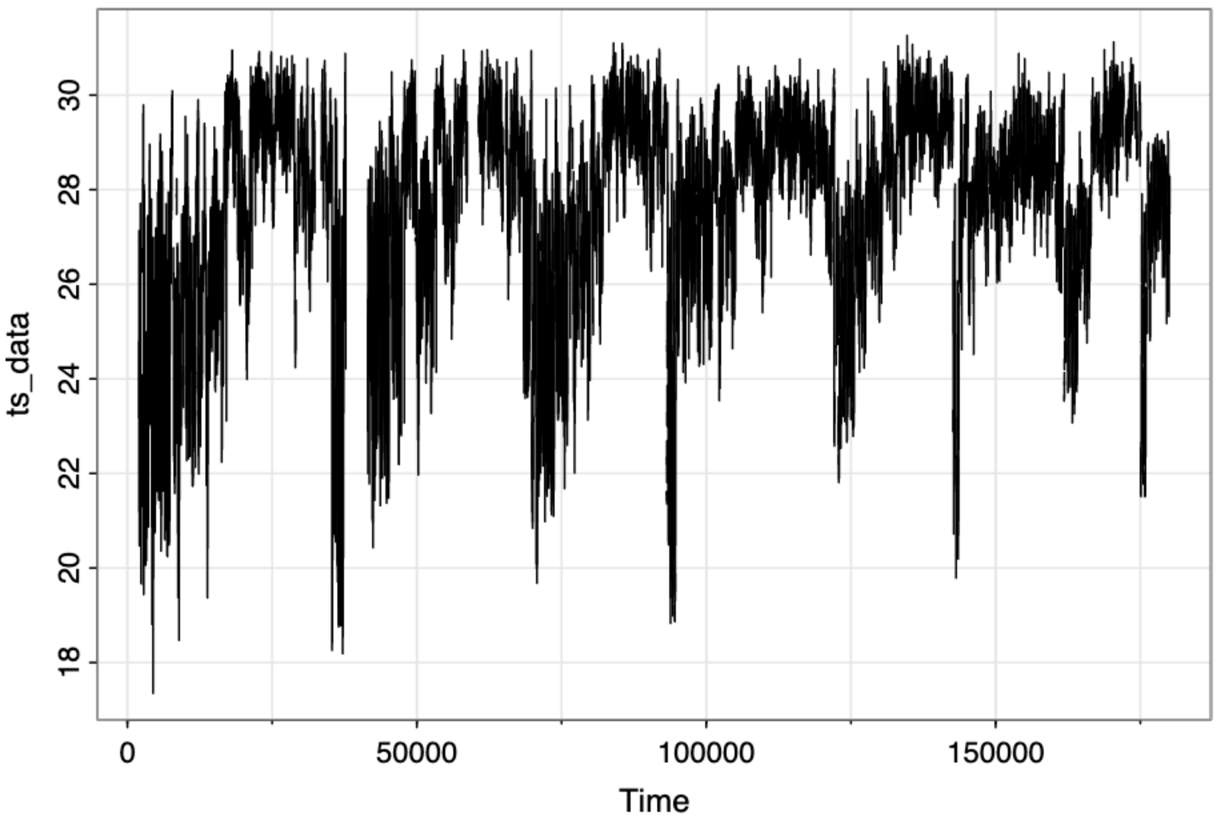
```
##
## Attaching package: 'astsa'
```

```
## The following objects are masked from 'package:fma':
##
##     chicken, sales
```

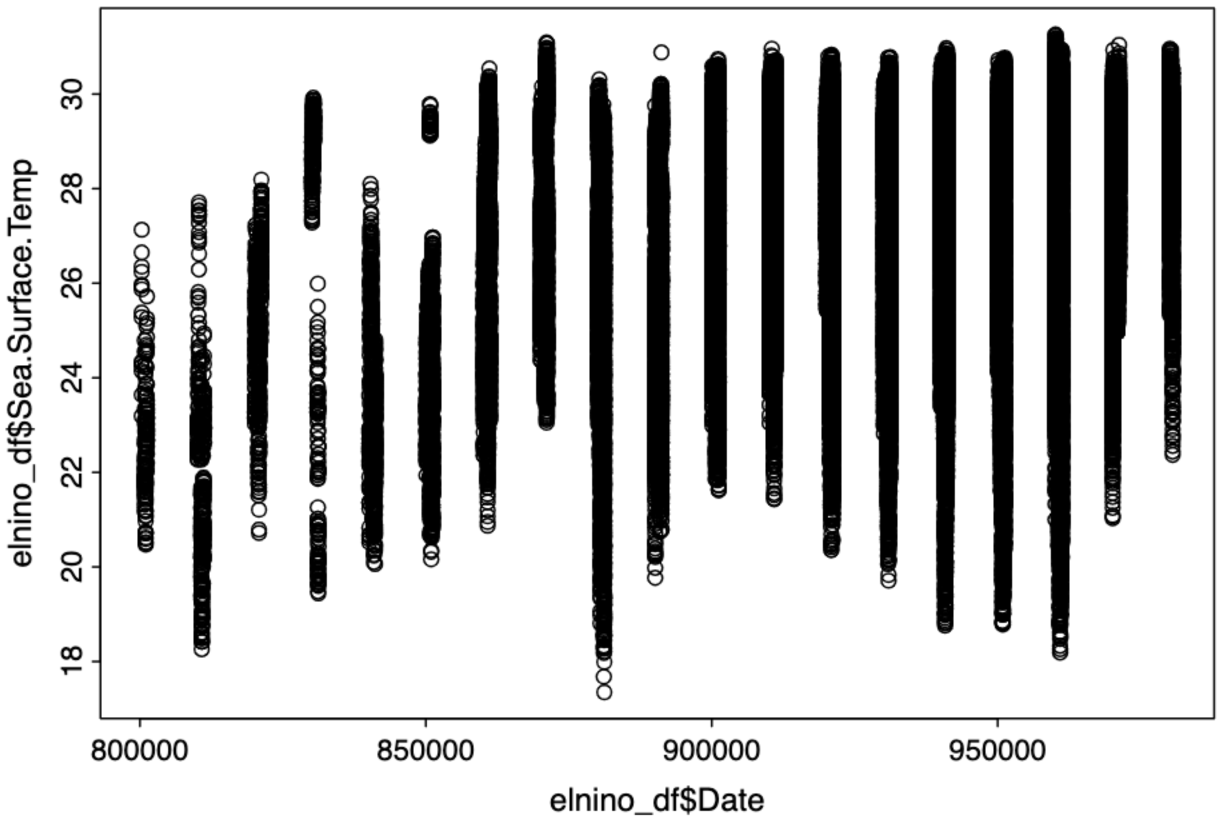
```
## The following object is masked from 'package:forecast':
##
##     gas
```

```
## The following object is masked from 'package:fpp2':
##
##     oil
```

```r
ts_data = ts(c(elnino_df$Sea.Surface.Temp), frequency = 1, start = c(1980, 7,3))
tsplot(ts_data)
```




```r
plot(elnino_df$Date, elnino_df$Sea.Surface.Temp)
```




```r
library(dplyr)
```

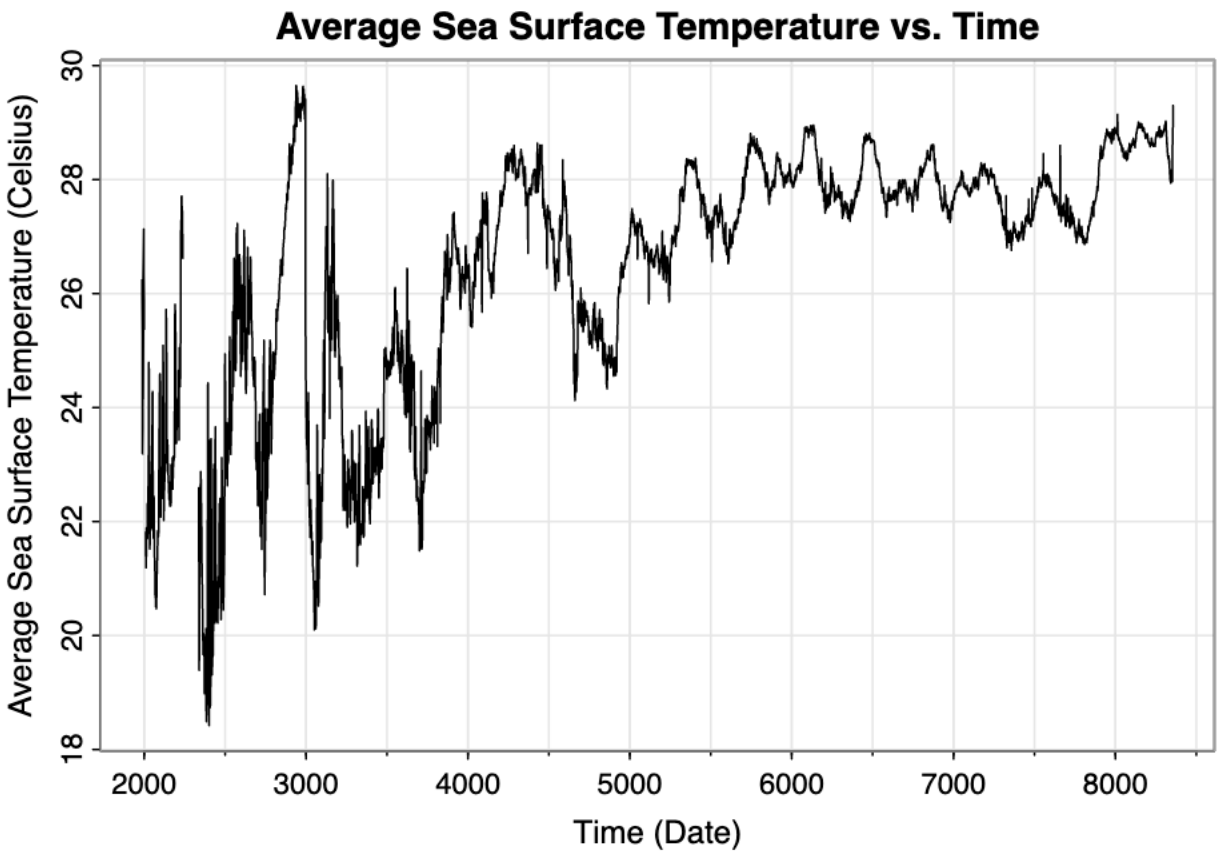
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```r
#elnino_df %>%
#   count(Date)


df_avg <- elnino_df %>%
  group_by(Date) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## i In group 1: `Date = 800307`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
```

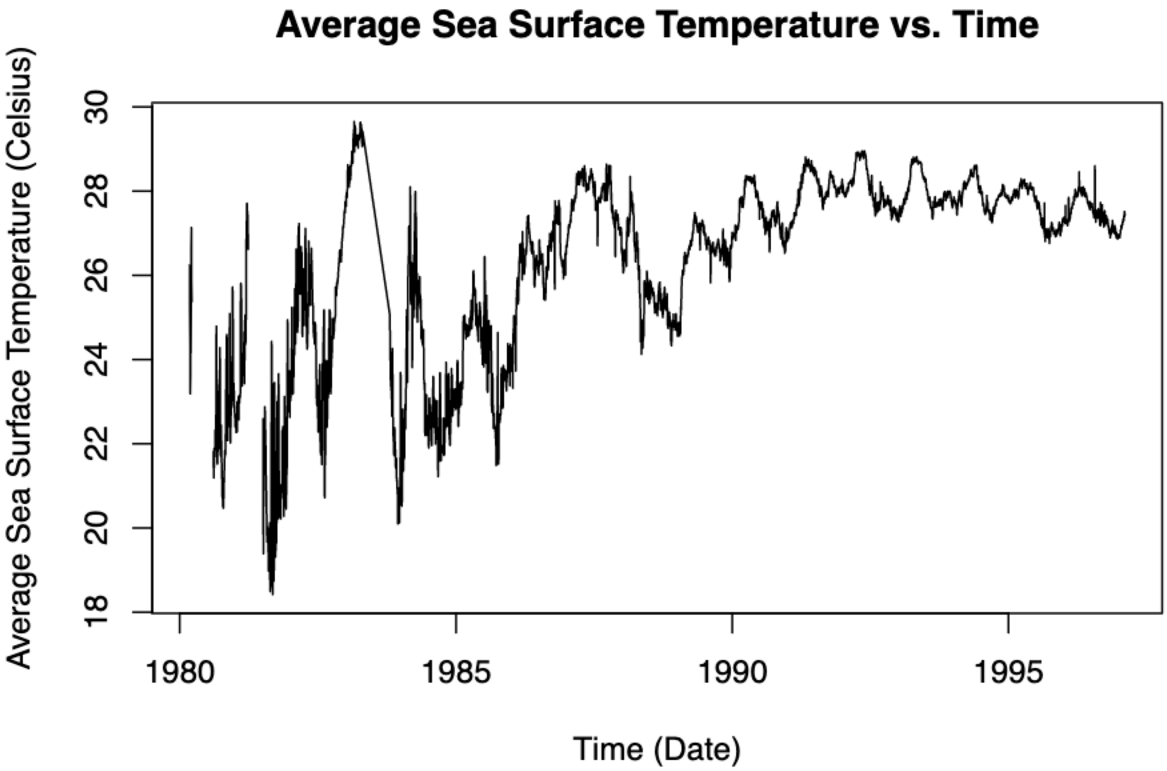
```
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```r
df_avg <- elnino_df %>%
  group_by(Date) %>%
  summarise(Avg.Sea.Surface.Temp = mean(Sea.Surface.Temp, na.rm = TRUE))

ts_data = ts(c(df_avg$Avg.Sea.Surface.Temp), frequency = 1, start = c(1980, 7,3))

tsplot(ts_data, , xlab = "Time (Date)", ylab = "Average Sea Surface Temperature (Celsius)", main = "Ave
```




```r
elnino_df <- elnino_df %>%
  mutate(Date = as.Date(paste0("19", substr(Date, 1, 2), "-",
                                substr(Date, 3, 4), "-",
                                substr(Date, 5, 6)), format="%Y-%m-%d"))
df_avg <- elnino_df %>%
  group_by(Date) %>%
  summarise(Avg.Sea.Surface.Temp = mean(Sea.Surface.Temp, na.rm = TRUE),
            Avg.Zonal.Winds = mean(Zonal.Winds, na.rm=TRUE),
            Avg.Meridional.Winds = mean(Meridional.Winds, na.rm=TRUE),
            Avg.Humidity= mean(Humidity, na.rm=TRUE),
            Avg.Air.Temp= mean(Air.Temp, na.rm=TRUE),
            Avg.Latitude = mean(Latitude, na.rm = TRUE),
            Avg.Longitude = mean(Longitude, na.rm=TRUE))

num_rows <- nrow(df_avg)
df_test <- df_avg[(num_rows - 499):num_rows, ]
```

```r
df_avg <- df_avg[1:(num_rows - 500), ]

plot(x = df_avg$Date, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surfac
```




```r
df_avg = na.omit(df_avg)
nrow(df_avg)
```

```
## [1] 2623
```

```r
head(df_avg)
```

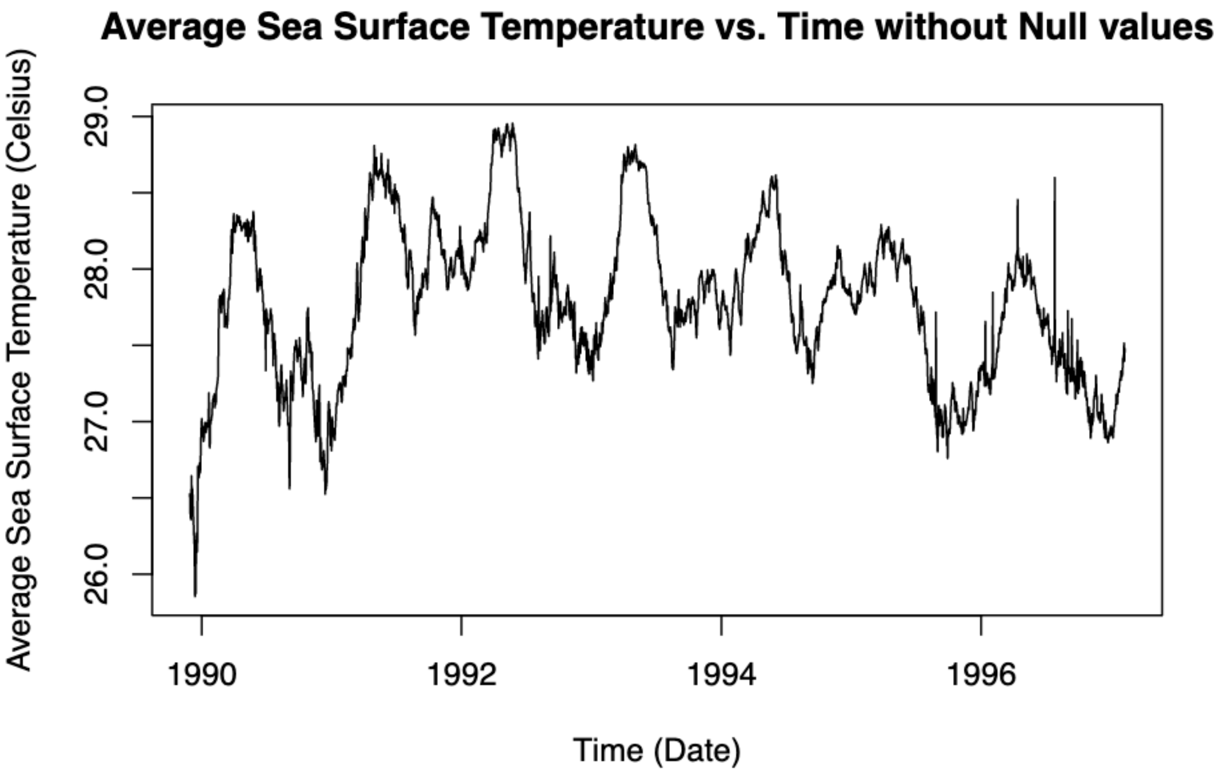
```
## # A tibble: 6 x 8
##   Date       Avg.Sea.Surface.Temp Avg.Zonal.Winds Avg.Meridional.Winds
##   <date>                    <dbl>           <dbl>                <dbl>
## 1 1989-11-28                 26.5           -2.28                1.47
## 2 1989-11-29                 26.5           -2.38                1.56
## 3 1989-11-30                 26.4           -1.86                1.78
## 4 1989-12-01                 26.4           -1.76                1.88
## 5 1989-12-02                 26.4           -2.27                0.545
## 6 1989-12-03                 26.6           -3.64               -0.228
## # i 4 more variables: Avg.Humidity <dbl>, Avg.Air.Temp <dbl>,
## #   Avg.Latitude <dbl>, Avg.Longitude <dbl>
```

```r
tail(df_avg)
```

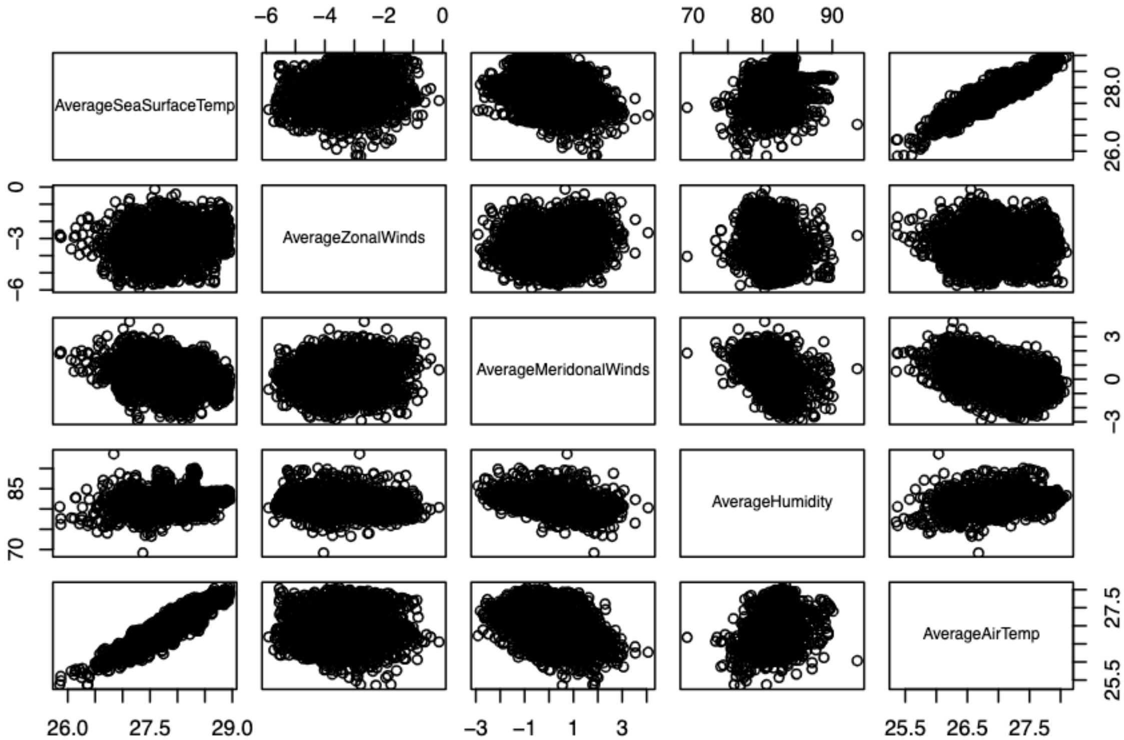
```
## # A tibble: 6 x 8
##   Date       Avg.Sea.Surface.Temp Avg.Zonal.Winds Avg.Meridional.Winds
##   <date>                    <dbl>           <dbl>                <dbl>
## 1 1997-02-03                 27.4           -5.48               -0.873
## 2 1997-02-04                 27.4           -4.80               -0.690
## 3 1997-02-05                 27.4           -4.58               -1.20
```

```
## 4 1997-02-06                    27.5          -4.79               -0.945
## 5 1997-02-07                    27.4          -4.47               -0.488
## 6 1997-02-08                    27.5          -4.63               -1.06
## # i 4 more variables: Avg.Humidity <dbl>, Avg.Air.Temp <dbl>,
## #   Avg.Latitude <dbl>, Avg.Longitude <dbl>
```

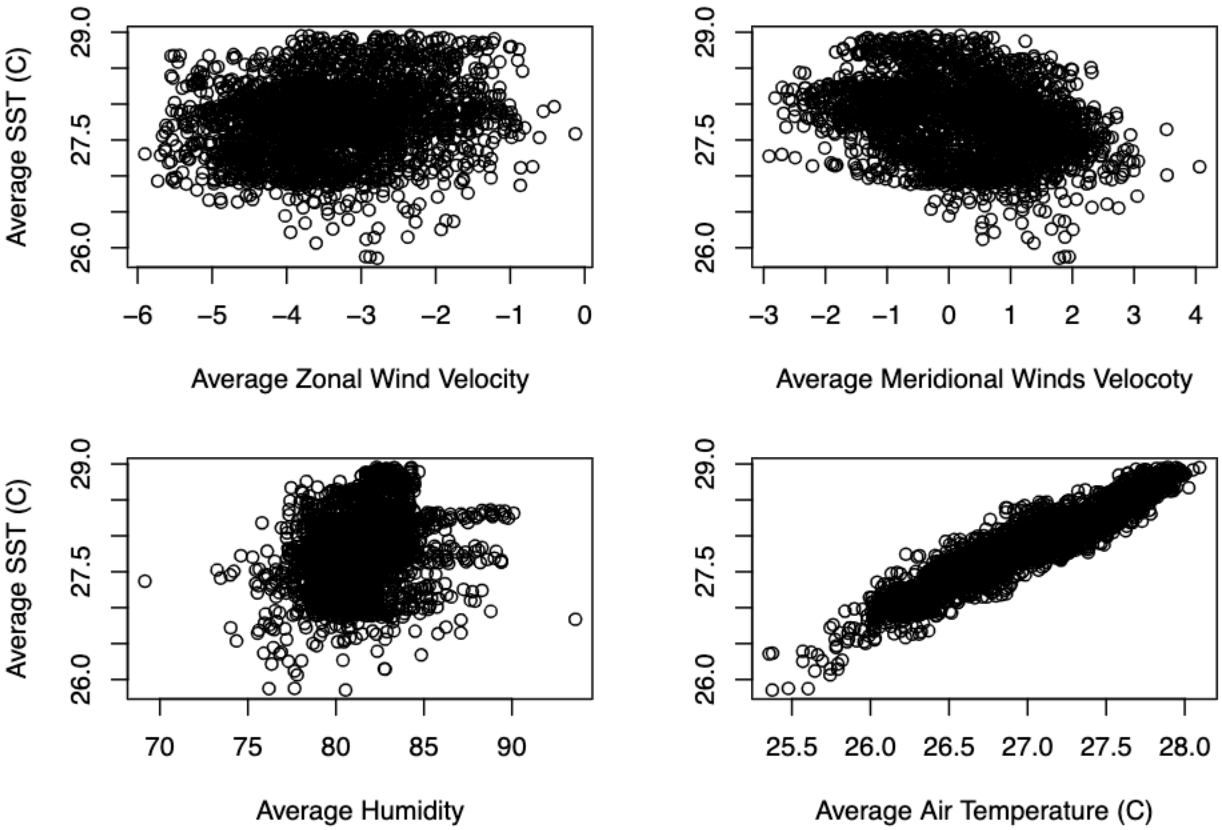
```
plot(x = df_avg$Date, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surfac
```



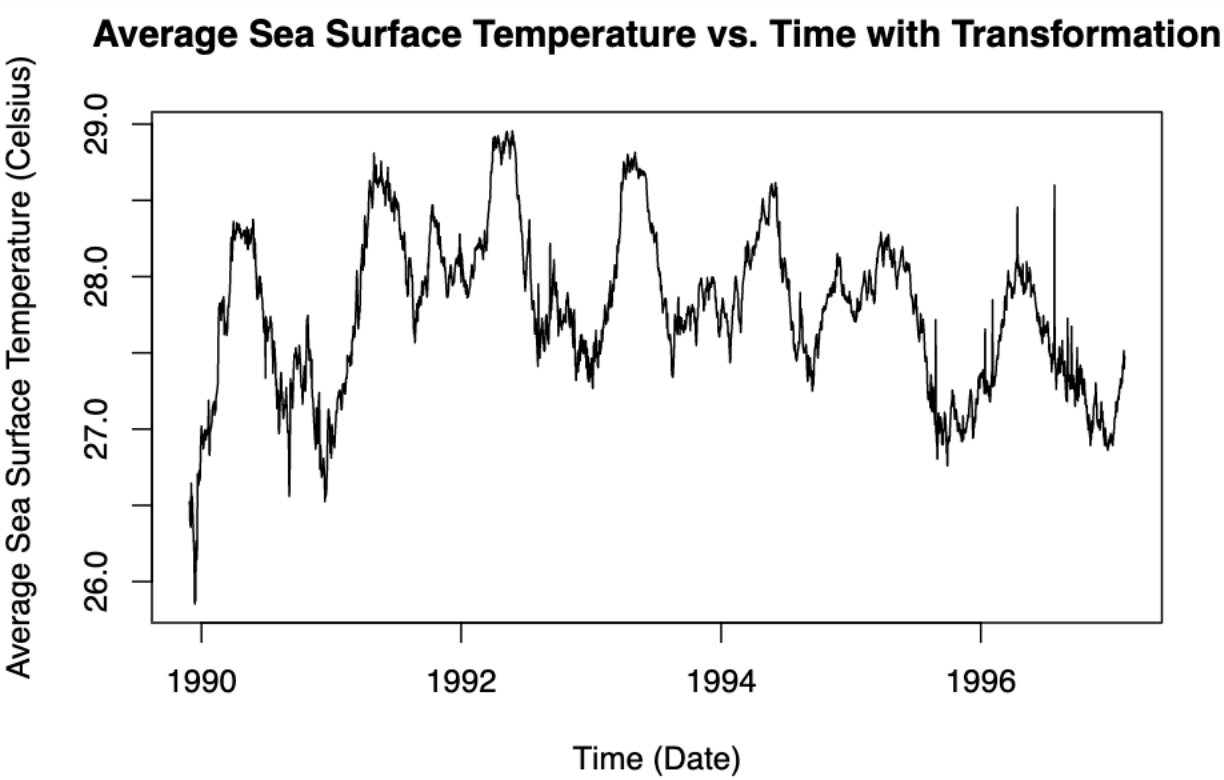

```
pairs(cbind(AverageSeaSurfaceTemp=df_avg$Avg.Sea.Surface.Temp,
            AverageZonalWinds=df_avg$Avg.Zonal.Winds,
            AverageMeridonalWinds=df_avg$Avg.Meridional.Winds,
            AverageHumidity=df_avg$Avg.Humidity,
            AverageAirTemp=df_avg$Avg.Air.Temp))
```




```r
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))

plot(x = df_avg$Avg.Zonal.Winds, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Zonal Wind Velocity",

plot(x = df_avg$Avg.Meridional.Winds, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Meridional Winds

plot(x = df_avg$Avg.Humidity, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Humidity", ylab = "Averag

plot(x = df_avg$Avg.Air.Temp, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Air Temperature (C)", yl
```



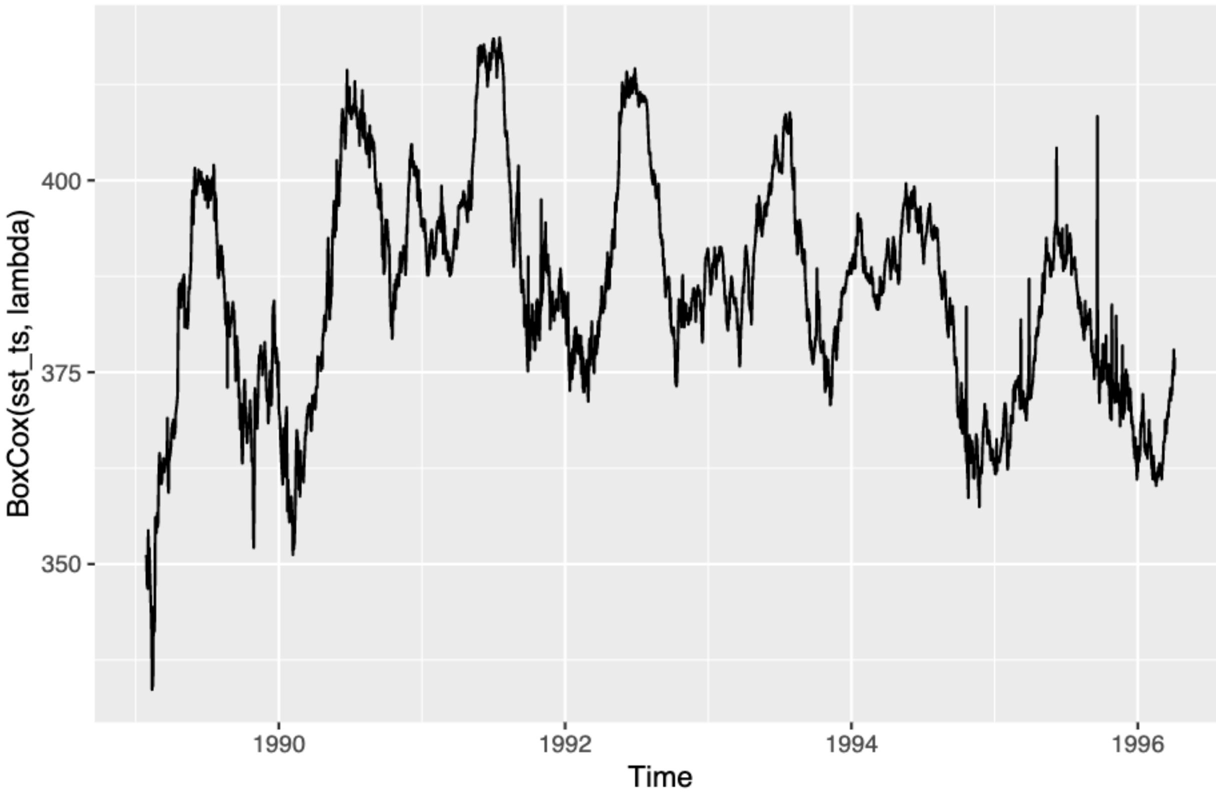

```r
library(fpp2)

plot(x = df_avg$Date, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surfac
```



```r
sst_ts  <- ts(df_avg$Avg.Sea.Surface.Temp, start = c(1989, 28, 11), frequency = 365)

lambda <- BoxCox.lambda(sst_ts)
lambda
```

```
## [1] 1.999924
```

```r
autoplot(BoxCox(sst_ts,lambda))
```

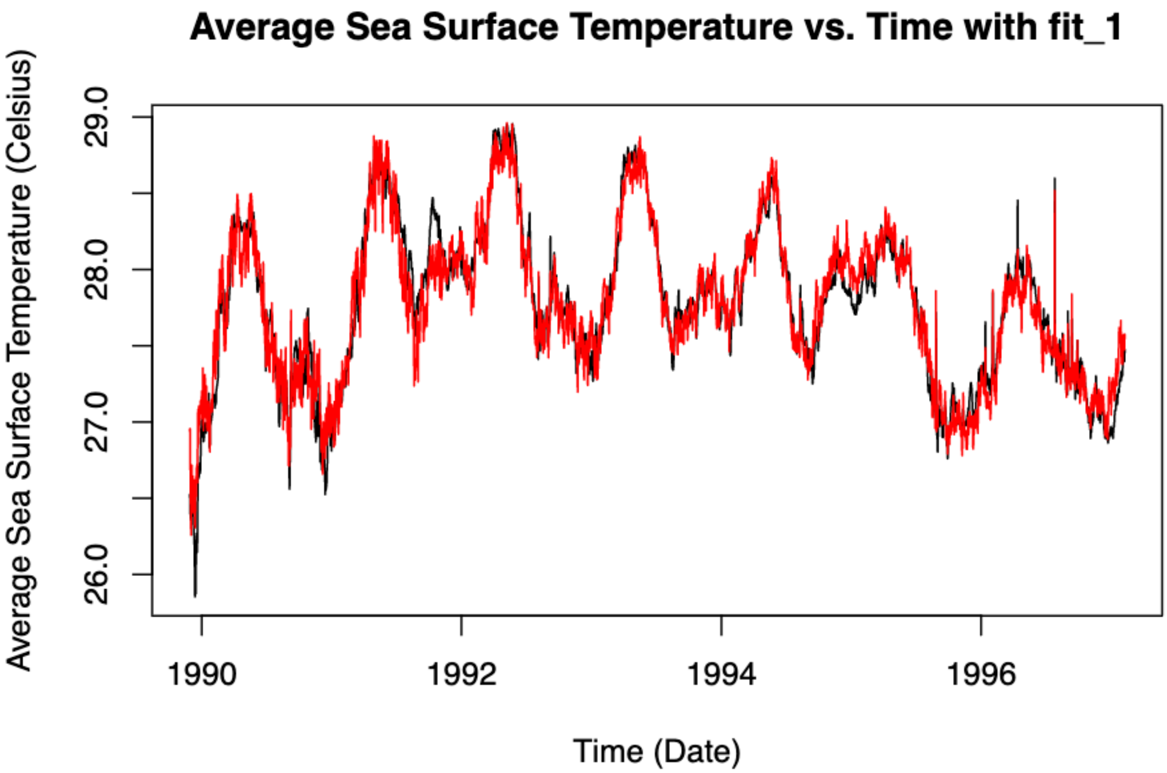



```r
trend <- time(df_avg$Avg.Sea.Surface.Temp)

#model 1
fit_1 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds + Avg.Meridional.Winds + Avg.Humidity + Avg.
summary(fit_1)
```

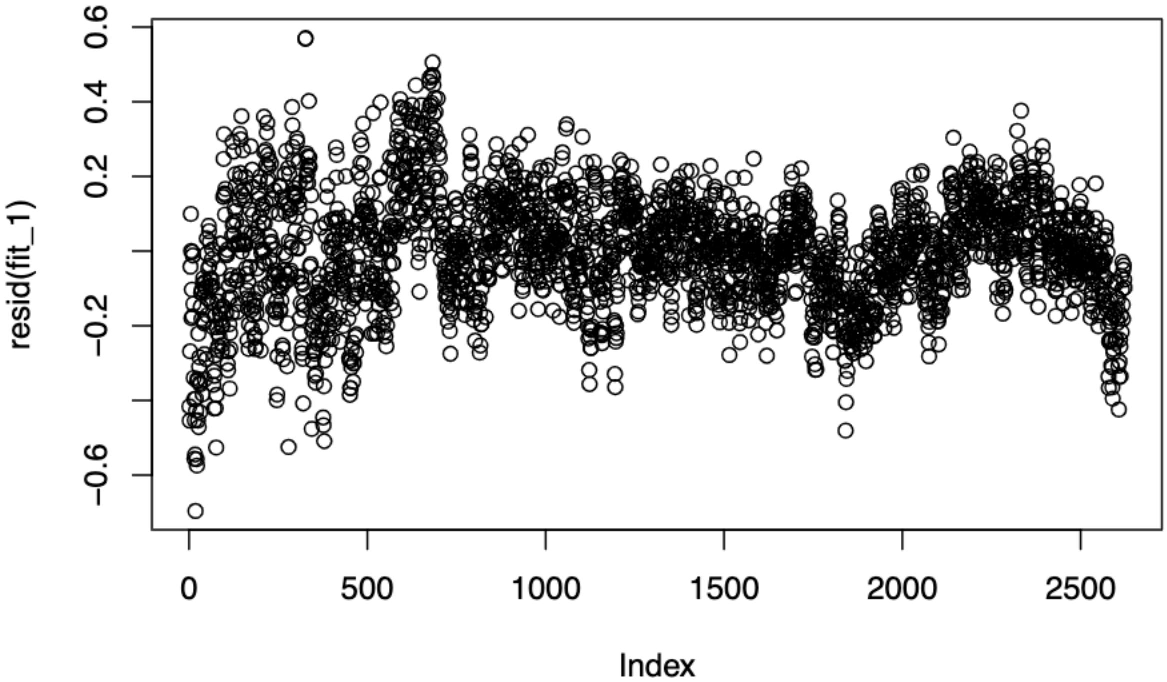
```
##
## Call:
## lm(formula = Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds +
##     Avg.Meridional.Winds + Avg.Humidity + Avg.Air.Temp, data = df_avg,
##     na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69613 -0.09442  0.00439  0.09657  0.57047
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -4.866e-02  2.122e-01  -0.229   0.8186
## trend                 5.588e-07  4.268e-06   0.131   0.8958
## Avg.Zonal.Winds       7.236e-02  3.215e-03  22.504   <2e-16 ***
```

```
## Avg.Meridional.Winds  6.463e-02  3.297e-03  19.602   <2e-16 ***
## Avg.Humidity         -4.197e-03  1.634e-03  -2.569   0.0102 *
## Avg.Air.Temp          1.055e+00  7.060e-03 149.472   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1515 on 2617 degrees of freedom
## Multiple R-squared:  0.9141, Adjusted R-squared:  0.9139
## F-statistic:  5567 on 5 and 2617 DF,  p-value: < 2.2e-16
```

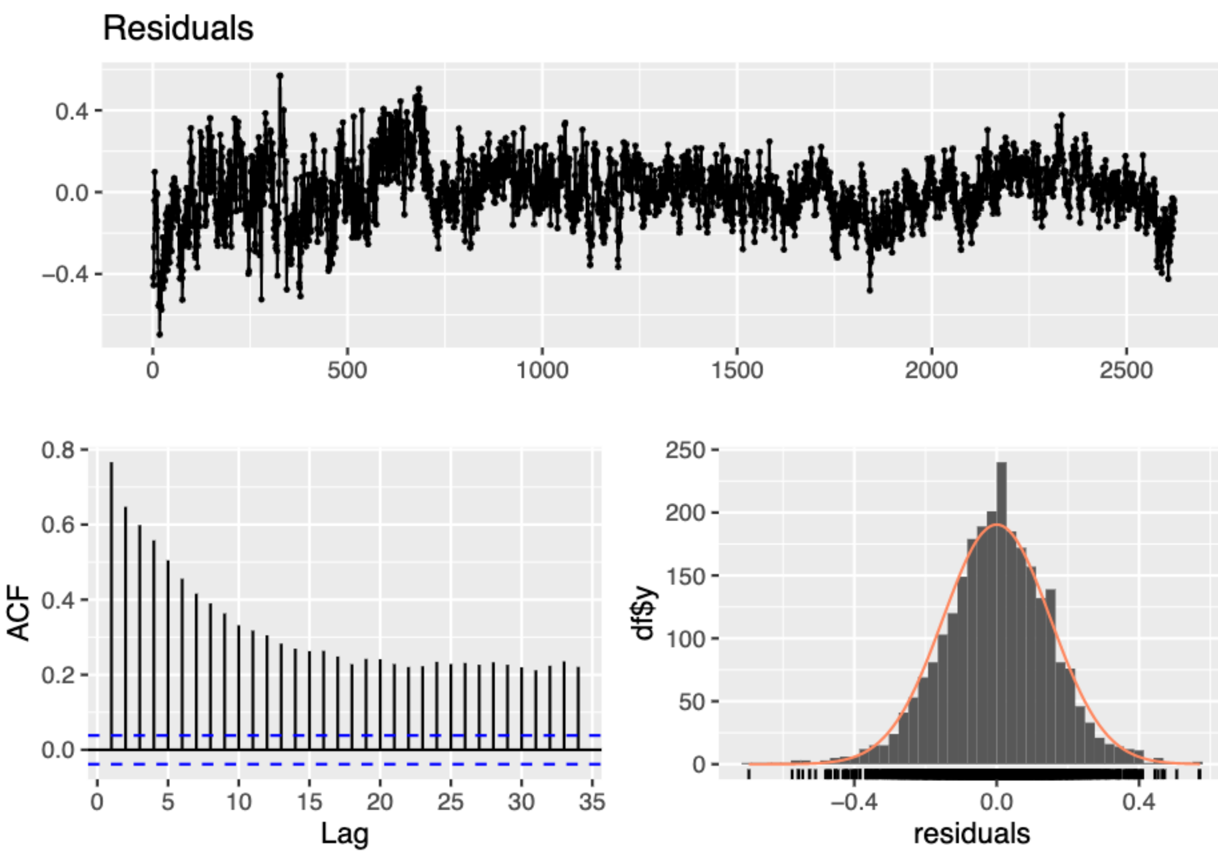
```r
plot(x = df_avg$Date, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surfac

fit1_pred <- predict(fit_1, newdata = df_avg)
lines(x = df_avg$Date, y = fit1_pred, col  = "red")
```




```r
plot(resid(fit_1))
```




```r
#acf2(resid(fit_1), 2000)

checkresiduals(fit_1, test = "LB")
```



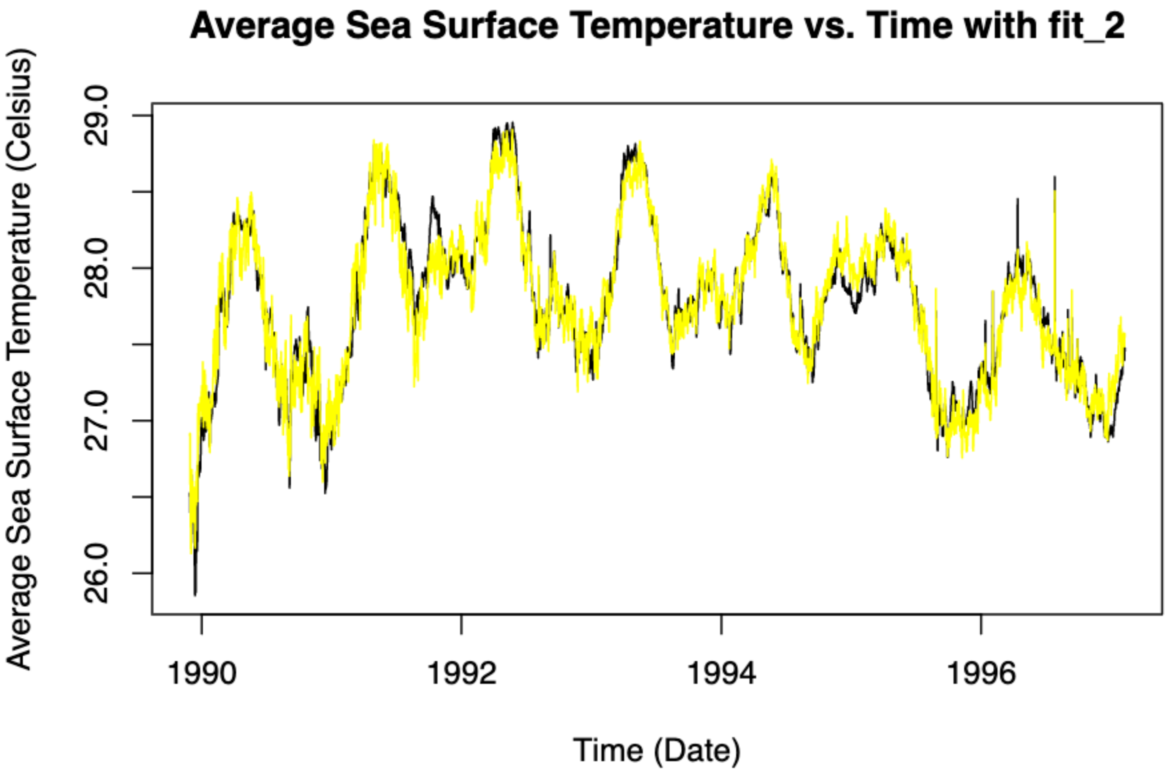

```
##
##  Ljung-Box test
```

```
##
## data:  Residuals
## Q* = 7109.6, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

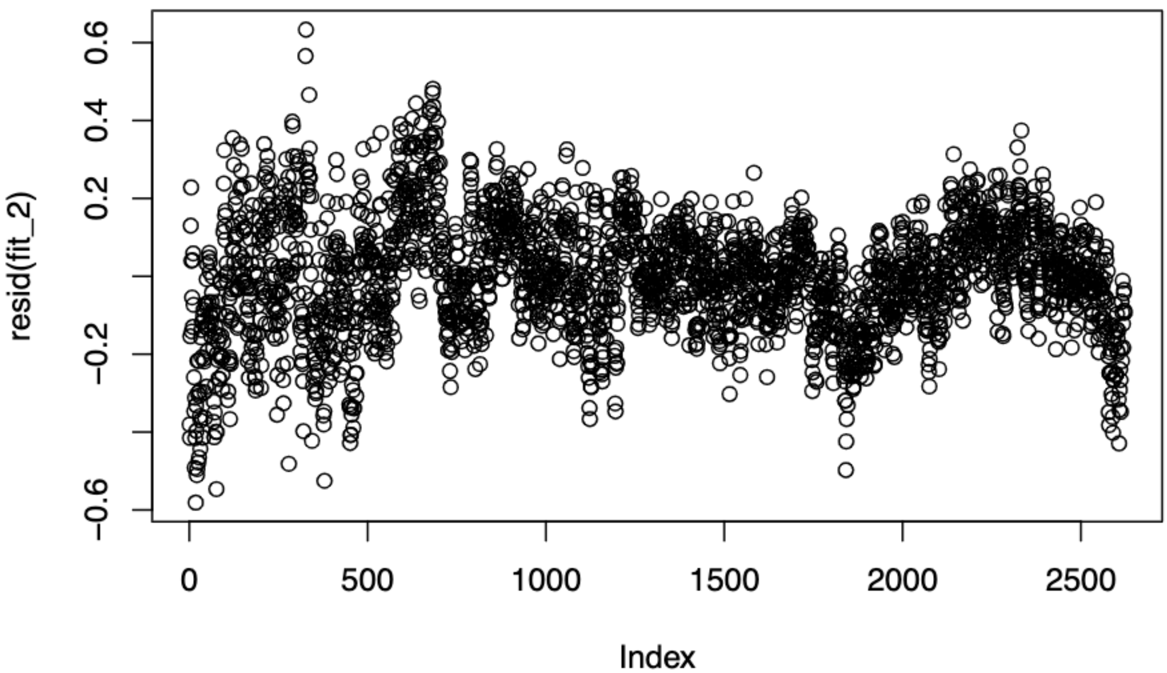
```r
#model 2
df_avg$Avg.Zonal.Winds_m = df_avg$Avg.Zonal.Winds - mean(df_avg$Avg.Zonal.Winds)
df_avg$Avg.Meridional.Winds_m = df_avg$Avg.Meridional.Winds - mean(df_avg$Avg.Meridional.Winds)
df_avg$Avg.Humidity_m = df_avg$Avg.Humidity -  mean(df_avg$Avg.Humidity)
df_avg$Avg.Air.Temp_m = df_avg$Avg.Air.Temp -mean(df_avg$Avg.Air.Temp)


fit_2 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds_m + I(Avg.Zonal.Winds_m^2) + Avg.Meridional.W
summary(fit_2)
```

```
##
## Call:
## lm(formula = Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds_m +
##     I(Avg.Zonal.Winds_m^2) + Avg.Meridional.Winds_m + I(Avg.Meridional.Winds_m^2) +
##     Avg.Humidity_m + I(Avg.Humidity_m^2) + Avg.Air.Temp_m + I(Avg.Air.Temp_m^2),
##     data = df_avg, na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58109 -0.09242  0.00271  0.09889  0.63320
##
## Coefficients:
##                               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                  2.782e+01  8.628e-03 3224.817  < 2e-16 ***
## trend                       -8.191e-06  4.500e-06   -1.820   0.0689 .
## Avg.Zonal.Winds_m            7.482e-02  3.223e-03   23.214  < 2e-16 ***
## I(Avg.Zonal.Winds_m^2)      -7.326e-03  2.391e-03   -3.064   0.0022 **
## Avg.Meridional.Winds_m       6.142e-02  3.279e-03   18.729  < 2e-16 ***
## I(Avg.Meridional.Winds_m^2) -1.225e-02  1.971e-03   -6.211 6.09e-10 ***
## Avg.Humidity_m              -3.572e-03  1.621e-03   -2.203   0.0277 *
## I(Avg.Humidity_m^2)         -3.655e-04  3.408e-04   -1.073   0.2836
## Avg.Air.Temp_m               1.052e+00  7.029e-03  149.661  < 2e-16 ***
## I(Avg.Air.Temp_m^2)         -7.008e-02  1.039e-02   -6.748 1.84e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1489 on 2613 degrees of freedom
## Multiple R-squared:  0.917,  Adjusted R-squared:  0.9168
## F-statistic:  3210 on 9 and 2613 DF,  p-value: < 2.2e-16
```

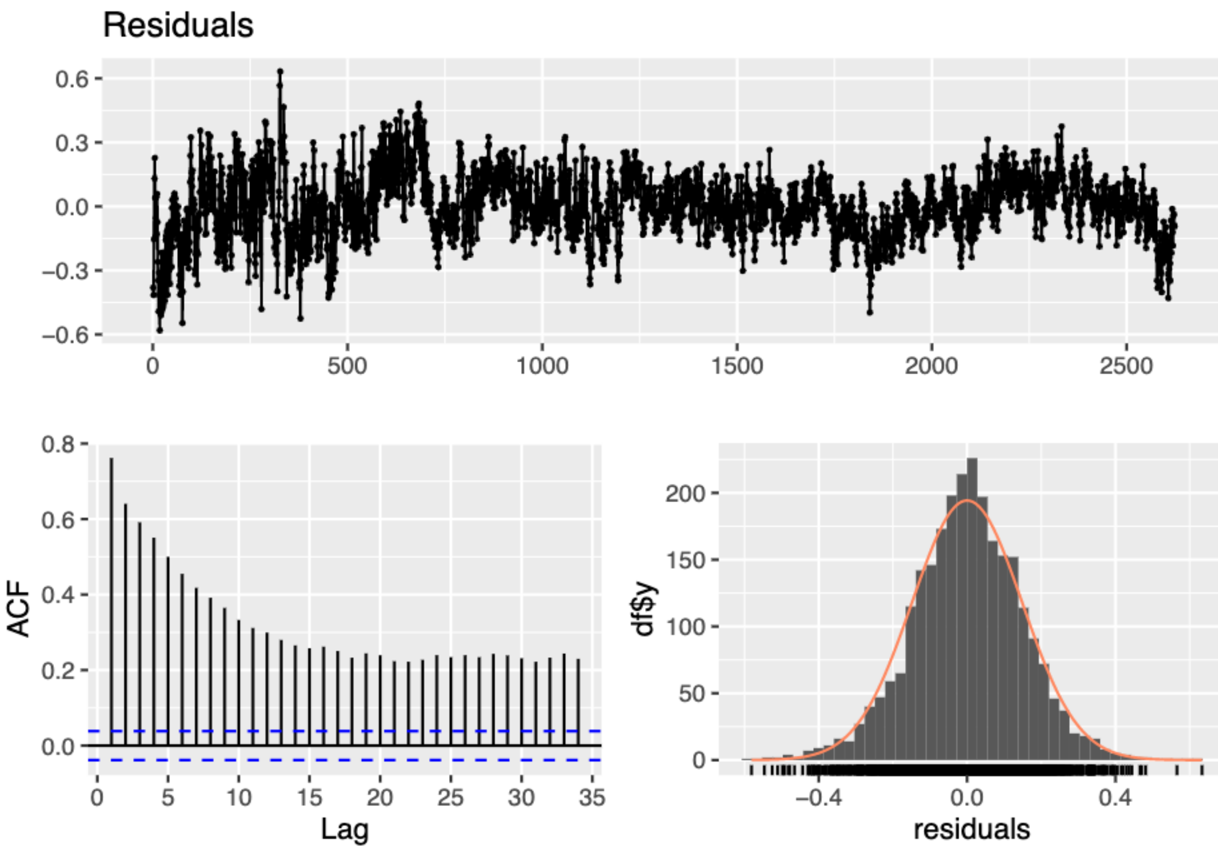
```r
plot(x = df_avg$Date, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surface

fit2_pred <- predict(fit_2, newdata = df_avg)
lines(x = df_avg$Date, y = fit2_pred, col  = "yellow")
```




```r
plot(resid(fit_2))
```




```r
#acf2(resid(fit_2), 2000)

checkresiduals(fit_2, test = "LB")
```



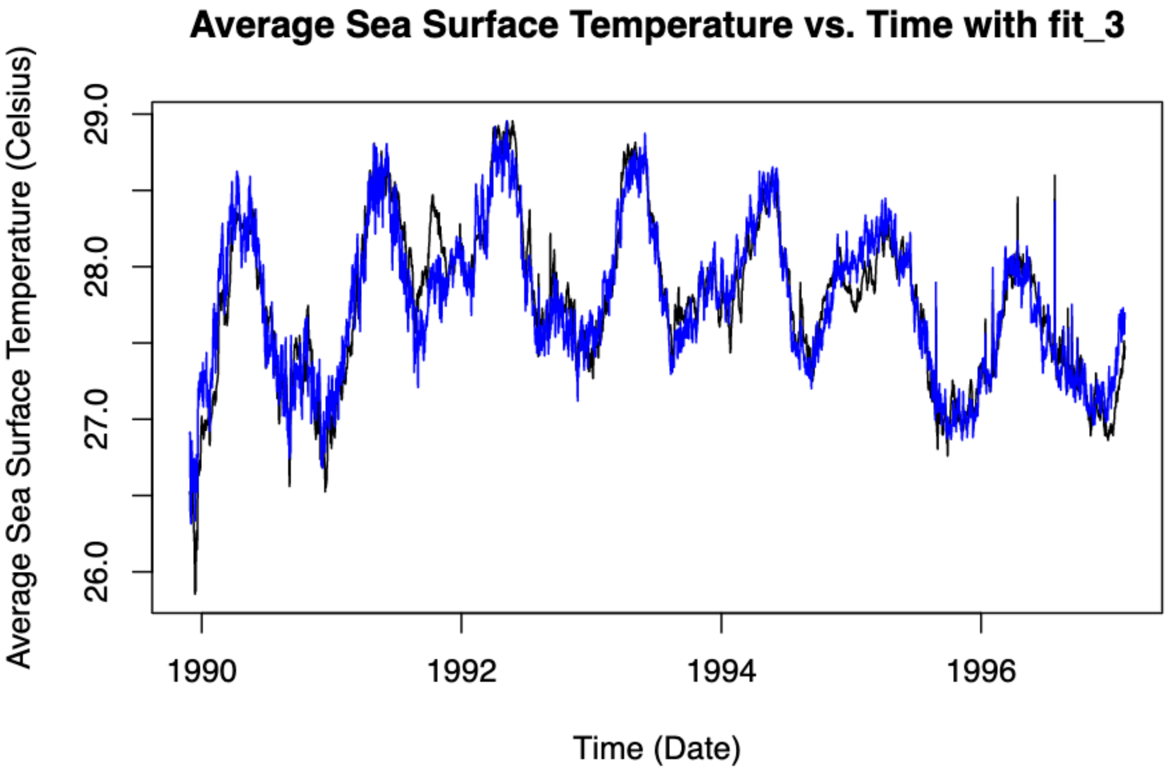

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 7021.4, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```r
#model 3


fit_3 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Air.Temp, data = df_avg)
summary(fit_3)
```

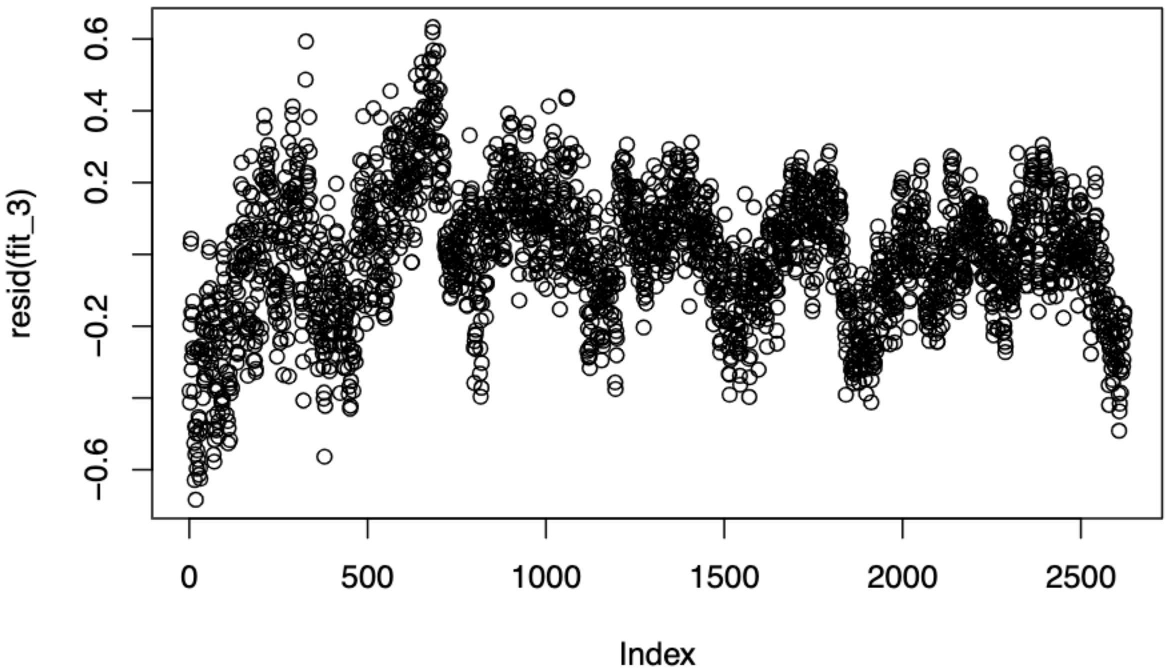
```
##
## Call:
## lm(formula = Avg.Sea.Surface.Temp ~ trend + Avg.Air.Temp, data = df_avg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68342 -0.11400  0.00842  0.12001  0.63301
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.535e+00  1.968e-01   7.798 8.99e-15 ***
## trend        -3.972e-05  4.718e-06  -8.418  < 2e-16 ***
## Avg.Air.Temp  9.773e-01  7.289e-03 134.076  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1821 on 2620 degrees of freedom
## Multiple R-squared:  0.8757, Adjusted R-squared:  0.8756
## F-statistic:  9227 on 2 and 2620 DF,  p-value: < 2.2e-16
```

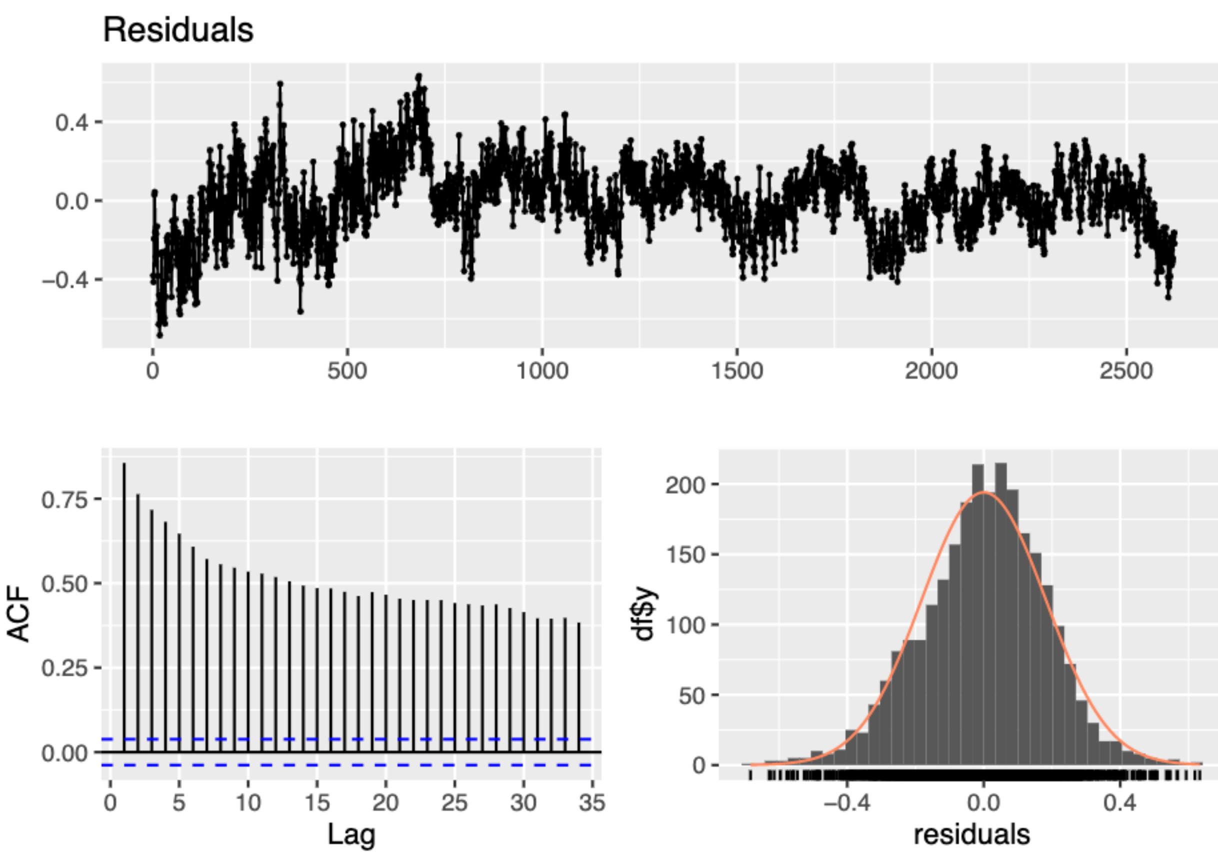
```
plot(x = df_avg$Date, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surfac

fit3_pred <- predict(fit_3, newdata = df_avg)
lines(x = df_avg$Date, y = fit3_pred, col  = "blue")
```




```
plot(resid(fit_3))
```




```r
#acf2(resid(fit_3), 2000)
checkresiduals(fit_3, test = "LB")
```




```
##
##  Ljung-Box test
##
```

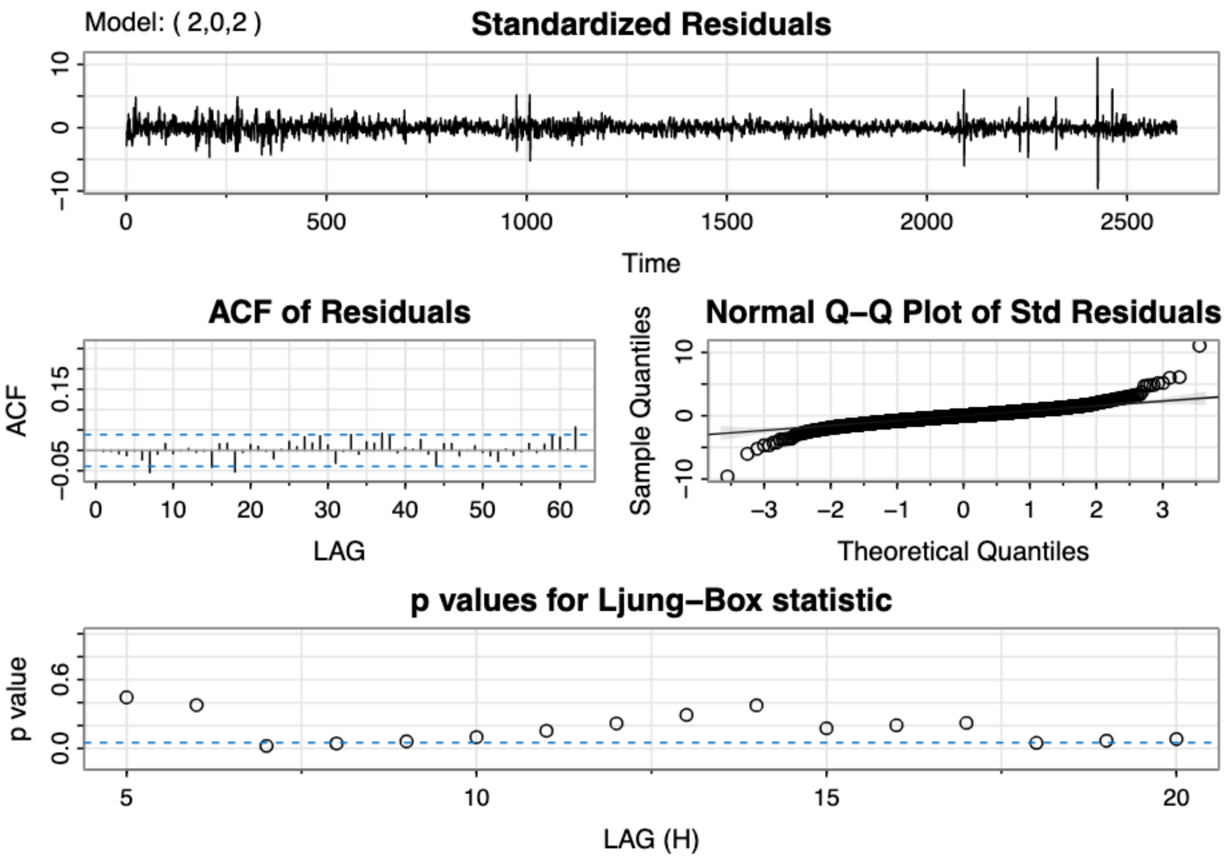
```
## data:  Residuals
## Q* = 11307, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```
#autocorrleated erros for fit1

fit_1 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds + Avg.Meridional.Winds + Avg.Humidity + Avg.A

sarima_mod <- sarima(df_avg$Avg.Sea.Surface.Temp, 2,0,2, xreg=cbind(trend,df_avg$Avg.Zonal.Winds, df_av
```

```
## initial  value -1.891407
## iter   2 value -2.202719
## iter   3 value -2.249806
## iter   4 value -2.342428
## iter   5 value -2.356937
## iter   6 value -2.382783
## iter   7 value -2.404413
## iter   8 value -2.417744
## iter   9 value -2.439352
## iter  10 value -2.468476
## iter  11 value -2.523376
## iter  12 value -2.555050
## iter  13 value -2.569153
## iter  14 value -2.571197
## iter  15 value -2.577946
## iter  16 value -2.585212
## iter  17 value -2.601282
## iter  18 value -2.636502
## iter  19 value -2.687405
## iter  20 value -2.702844
## iter  21 value -2.709890
## iter  22 value -2.715914
## iter  23 value -2.718160
## iter  24 value -2.718419
## iter  25 value -2.724149
## iter  26 value -2.736049
## iter  27 value -2.750666
## iter  28 value -2.756780
## iter  29 value -2.758209
## iter  30 value -2.759394
## iter  31 value -2.761095
## iter  32 value -2.764093
## iter  33 value -2.767149
## iter  34 value -2.769385
## iter  35 value -2.770433
## iter  36 value -2.771111
## iter  37 value -2.772760
## iter  38 value -2.775040
## iter  39 value -2.776937
## iter  40 value -2.777653
## iter  41 value -2.777852
## iter  42 value -2.777953
## iter  43 value -2.778075
## iter  44 value -2.778076
```

```
## iter  45 value -2.778092
## iter  46 value -2.778134
## iter  47 value -2.778141
## iter  48 value -2.778167
## iter  49 value -2.778170
## iter  50 value -2.778171
## iter  51 value -2.778171
## iter  52 value -2.778172
## iter  53 value -2.778173
## iter  54 value -2.778174
## iter  55 value -2.778174
## iter  56 value -2.778174
## iter  57 value -2.778174
## iter  58 value -2.778175
## iter  59 value -2.778176
## iter  60 value -2.778176
## iter  61 value -2.778176
## iter  62 value -2.778176
## iter  63 value -2.778177
## iter  64 value -2.778178
## iter  65 value -2.778178
## iter  66 value -2.778179
## iter  67 value -2.778179
## iter  67 value -2.778179
## iter  67 value -2.778179
## final  value -2.778179
## converged
## initial  value -2.775831
## iter   2 value -2.775862
## iter   3 value -2.775896
## iter   4 value -2.775966
## iter   5 value -2.776048
## iter   6 value -2.776079
## iter   7 value -2.776082
## iter   8 value -2.776082
## iter   9 value -2.776082
## iter  10 value -2.776083
## iter  11 value -2.776083
## iter  11 value -2.776083
## iter  11 value -2.776083
## final  value -2.776083
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##           Estimate     SE t.value p.value
## ar1         0.7160 0.6297  1.1371  0.2556
## ar2         0.2728 0.6236  0.4375  0.6618
## ma1         0.0717 0.6284  0.1142  0.9091
## ma2        -0.0976 0.1261 -0.7741  0.4389
## intercept  17.7191 0.4076 43.4690  0.0000
## trend      -0.0001 0.0001 -0.5635  0.5732
##             0.0321 0.0029 10.9856  0.0000
##            -0.0039 0.0025 -1.5919  0.1115
```

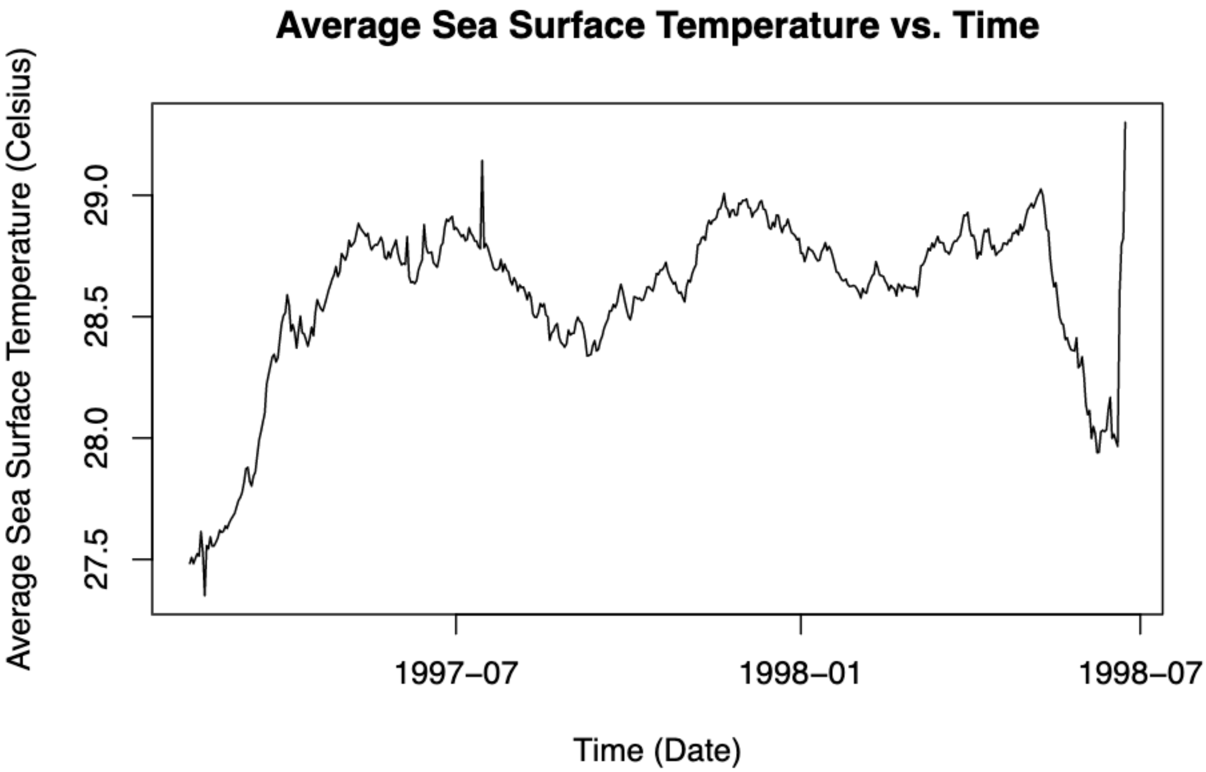
```
##              0.0002 0.0011  0.1465  0.8835
##              0.3808 0.0122 31.3164  0.0000
##
## sigma^2 estimated as 0.003873779 on 2613 degrees of freedom
##
## AIC = -2.705901  AICc = -2.705869  BIC = -2.681276
##
```




```r
resid_mod <- sarima_mod$fit$residuals
#acf2(resid_mod, 2000)

lb_test <- Box.test(ts(resid_mod), lag = 2000, type = "Ljung-Box", fitdf = 0)

lb_pvals <- sapply(1:2000, function(k) Box.test(ts(resid_mod), lag = k, type = "Ljung-Box", fitdf = 0)$p

sum(lb_pvals > 0.05)
```

```
## [1] 1959
```

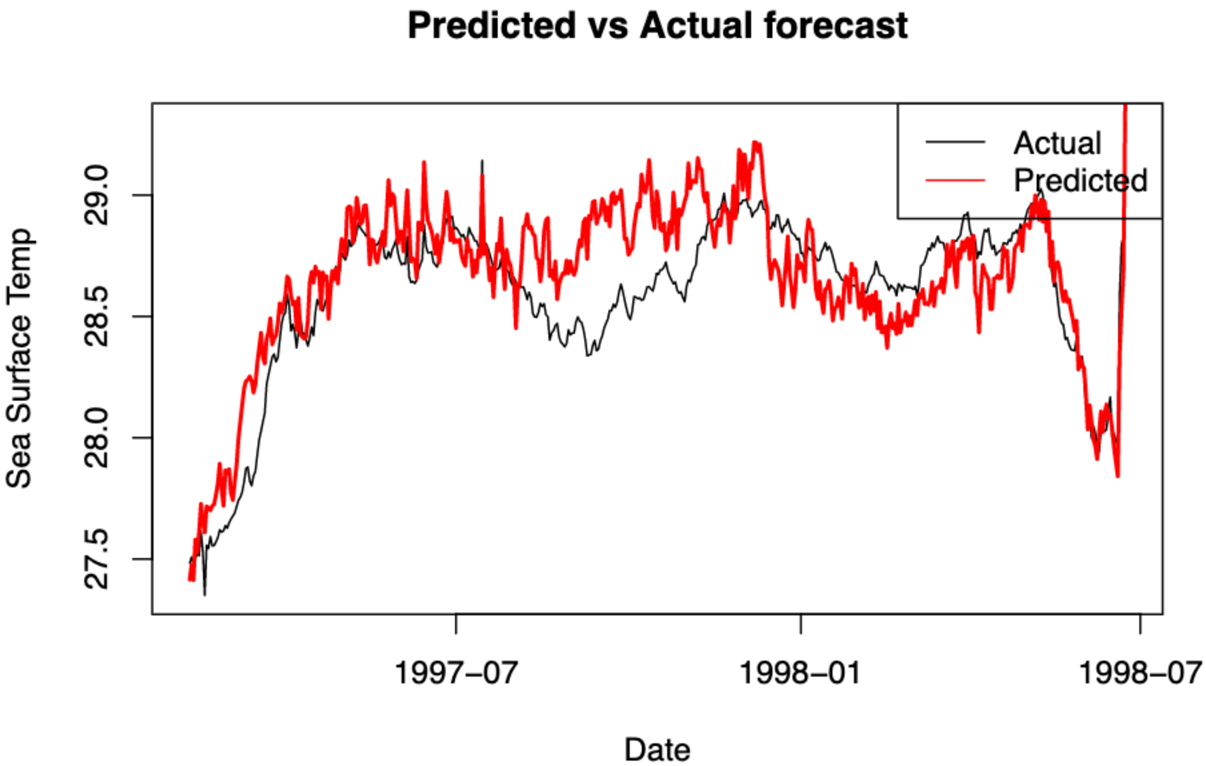
```r
plot(x = df_test$Date, y = df_test$Avg.Sea.Surface.Temp, xlab = "Time (Date)", ylab = "Average Sea Surf
```



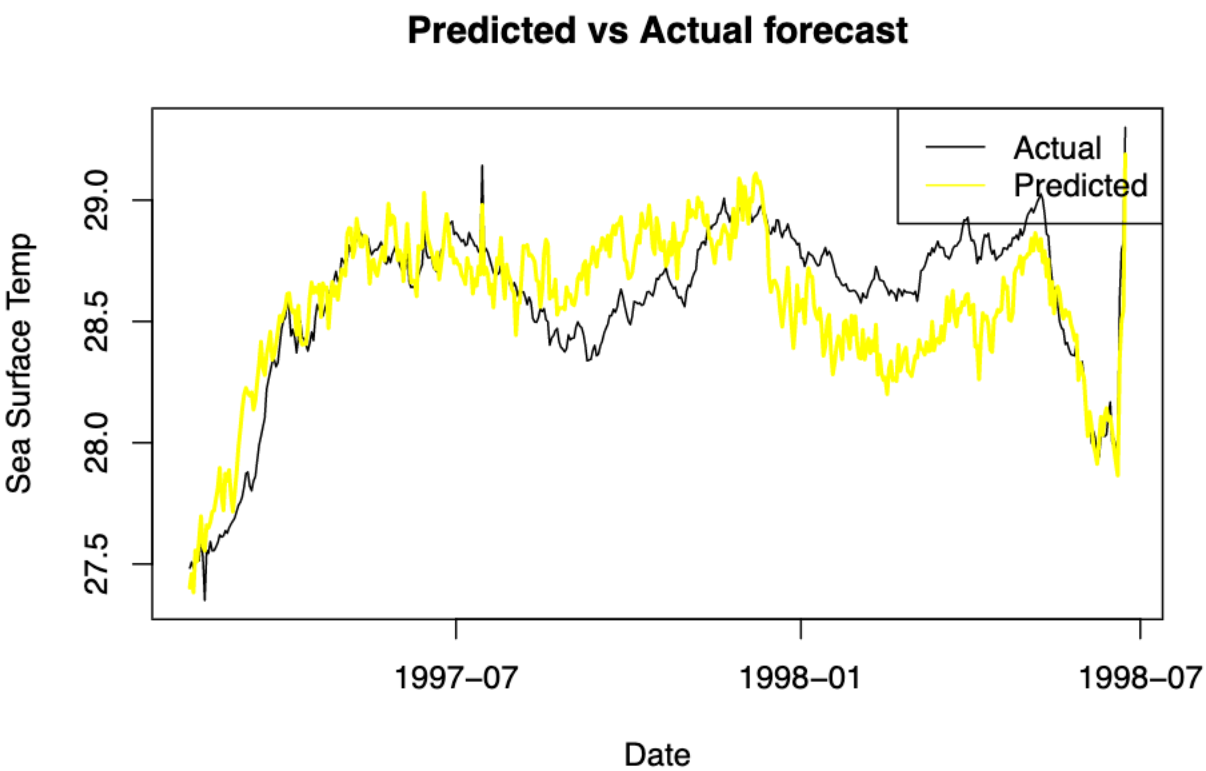

```r
head(df_test)
```

```
## # A tibble: 6 x 8
##   Date       Avg.Sea.Surface.Temp Avg.Zonal.Winds Avg.Meridional.Winds
##   <date>                    <dbl>           <dbl>                <dbl>
## 1 1997-02-09                 27.5           -5.16               -0.892
## 2 1997-02-10                 27.5           -5.43               -0.794
## 3 1997-02-11                 27.5           -5.54               -1.11
## 4 1997-02-12                 27.5           -5.36               -1.08
## 5 1997-02-13                 27.5           -4.83               -1.26
## 6 1997-02-14                 27.5           -5.01               -0.937
## # i 4 more variables: Avg.Humidity <dbl>, Avg.Air.Temp <dbl>,
## #   Avg.Latitude <dbl>, Avg.Longitude <dbl>
```

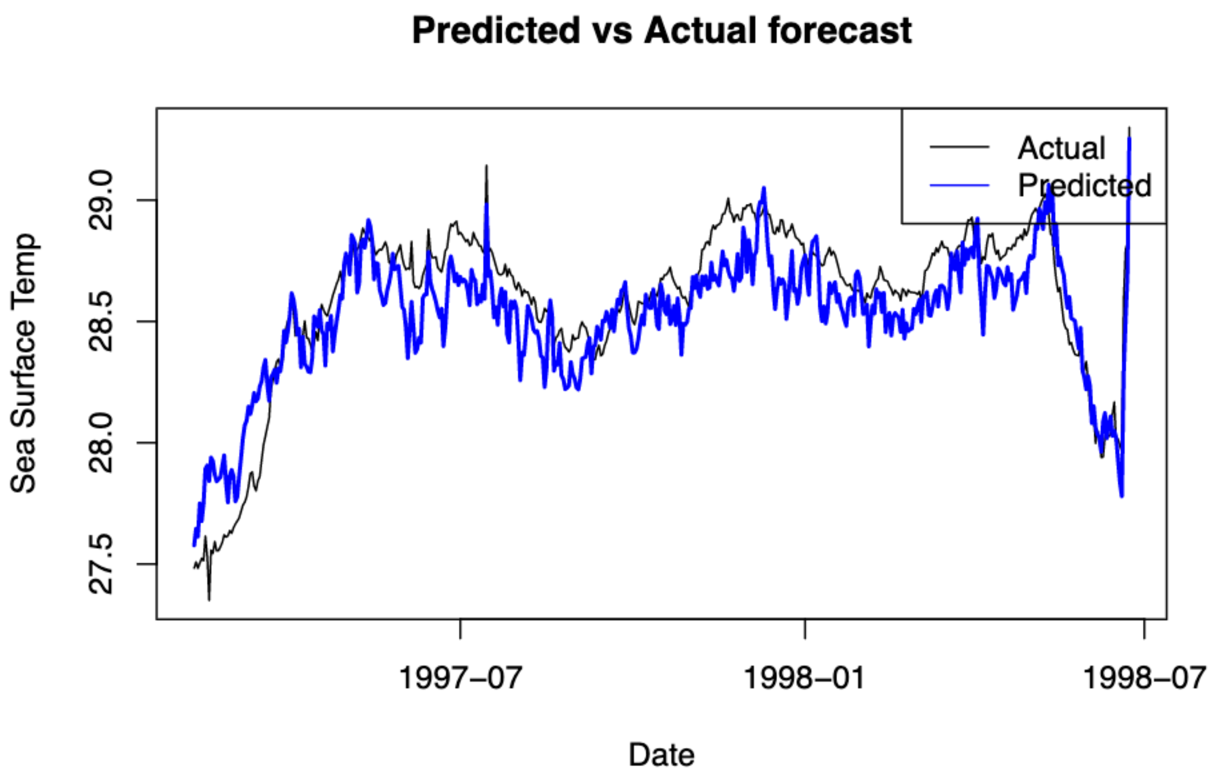
```r
n_train <- nrow(df_avg)
df_test$trend <- (n_train + 1):(n_train + nrow(df_test))
#forecasrt fit1

pred_test <- predict(fit_1, newdata = df_test)

plot(df_test$Date, df_test$Avg.Sea.Surface.Temp, type = "l", col = "black", ylab = "Sea Surface Temp", 
lines(df_test$Date, pred_test, col = "red", lwd = 2)
legend("topright", legend = c("Actual", "Predicted"), col = c("black", "red"), lty = 1)
```




```
#forecasrt fit2
df_test$Avg.Zonal.Winds_m <- df_test$Avg.Zonal.Winds - mean(df_avg$Avg.Zonal.Winds, na.rm = TRUE)
df_test$Avg.Meridional.Winds_m <- df_test$Avg.Meridional.Winds - mean(df_avg$Avg.Meridional.Winds, na.r
df_test$Avg.Humidity_m <- df_test$Avg.Humidity - mean(df_avg$Avg.Humidity, na.rm = TRUE)
df_test$Avg.Air.Temp_m <- df_test$Avg.Air.Temp - mean(df_avg$Avg.Air.Temp, na.rm = TRUE)

pred_test2 <- predict(fit_2, newdata = df_test)

plot(df_test$Date, df_test$Avg.Sea.Surface.Temp, type = "l", col = "black", ylab = "Sea Surface Temp", 
lines(df_test$Date, pred_test2, col = "yellow", lwd = 2)
legend("topright", legend = c("Actual", "Predicted"), col = c("black", "yellow"), lty = 1)
```




```r
#forecasrt fit3
pred_test3 <- predict(fit_3, newdata = df_test)

plot(df_test$Date, df_test$Avg.Sea.Surface.Temp, type = "l", col = "black", ylab = "Sea Surface Temp", 
lines(df_test$Date, pred_test3, col = "blue", lwd = 2)
legend("topright", legend = c("Actual", "Predicted"), col = c("black", "blue"), lty = 1)
```




```r
names(df_avg)
```

```
##  [1] "Date"                   "Avg.Sea.Surface.Temp"   "Avg.Zonal.Winds"
##  [4] "Avg.Meridional.Winds"   "Avg.Humidity"           "Avg.Air.Temp"
##  [7] "Avg.Latitude"           "Avg.Longitude"          "Avg.Zonal.Winds_m"
## [10] "Avg.Meridional.Winds_m" "Avg.Humidity_m"         "Avg.Air.Temp_m"
```

```r
#mse ft_1
mse_1 = mean((pred_test - df_test$Avg.Sea.Surface.Temp)^2)
rmse_1 = sqrt(mse_1)
#mse ft_2
mse_2 = mean((pred_test2 - df_test$Avg.Sea.Surface.Temp)^2)
rmse_2 = sqrt(mse_2)
#mse ft_3
mse_3 = mean((pred_test3 - df_test$Avg.Sea.Surface.Temp)^2)
rmse_3 = sqrt(mse_3)

print("MSE, RMSE of fit_1")
```

```
## [1] "MSE, RMSE of fit_1"
```

```r
c(mse_1, rmse_1)
```

```
## [1] 0.04162505 0.20402218
```

```r
print("MSE, RMSE of fit_2")
```

```
## [1] "MSE, RMSE of fit_2"
```

```r
c(mse_2, rmse_2)
```

```
## [1] 0.04433859 0.21056730
```

```r
print("MSE, RMSE of fit_3")
```

```
## [1] "MSE, RMSE of fit_3"
```

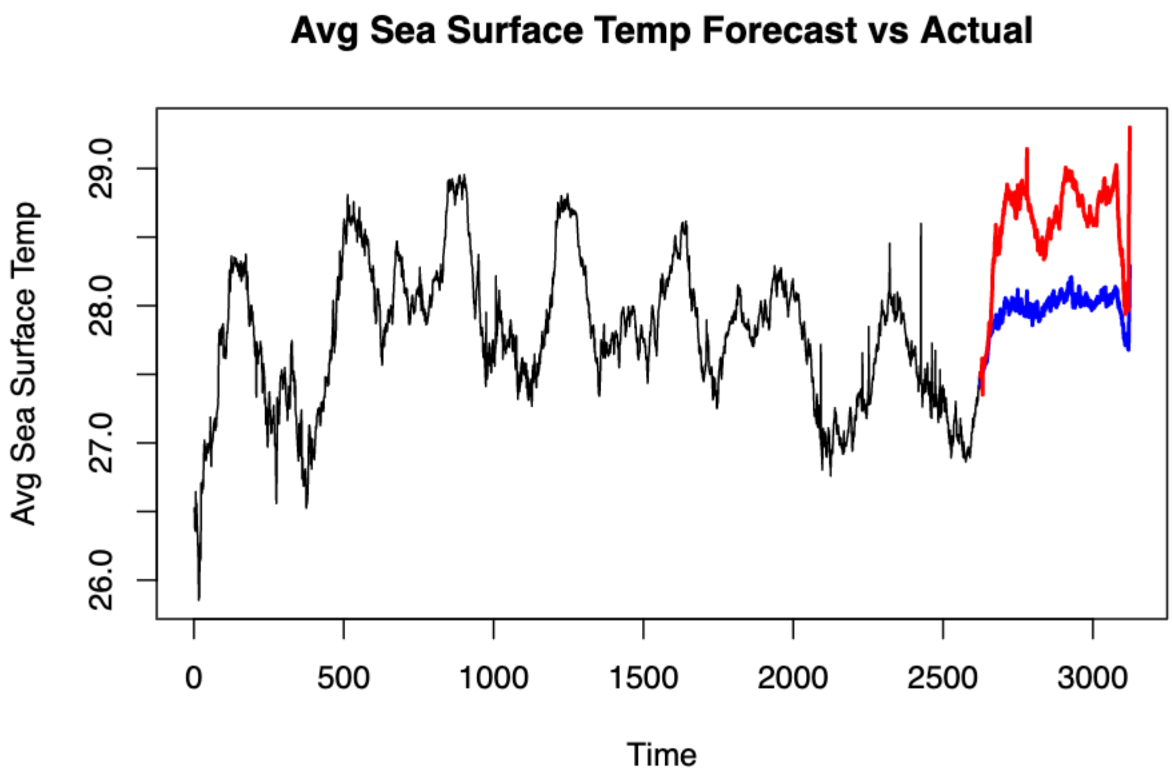
```r
c(mse_3, rmse_3)
```

```
## [1] 0.02669385 0.16338252
```

```r
library(astsa)

y_train <- df_avg$Avg.Sea.Surface.Temp
ts_train <- ts(y_train, frequency = 1)
n_train <- nrow(df_avg)
n_test  <- nrow(df_test)

df_avg$trend  <- seq_len(n_train)
df_test$trend <- seq_len(n_test) + n_train
cols<- c("trend", "Avg.Zonal.Winds", "Avg.Meridional.Winds", "Avg.Humidity", "Avg.Air.Temp")
X_train <- as.matrix(df_avg[, cols])
X_future<- as.matrix(df_test[, cols])

sarima_mod <- sarima(xdata = ts_train, p =2, d =0, q =2, xreg = X_train, details = FALSE)
```

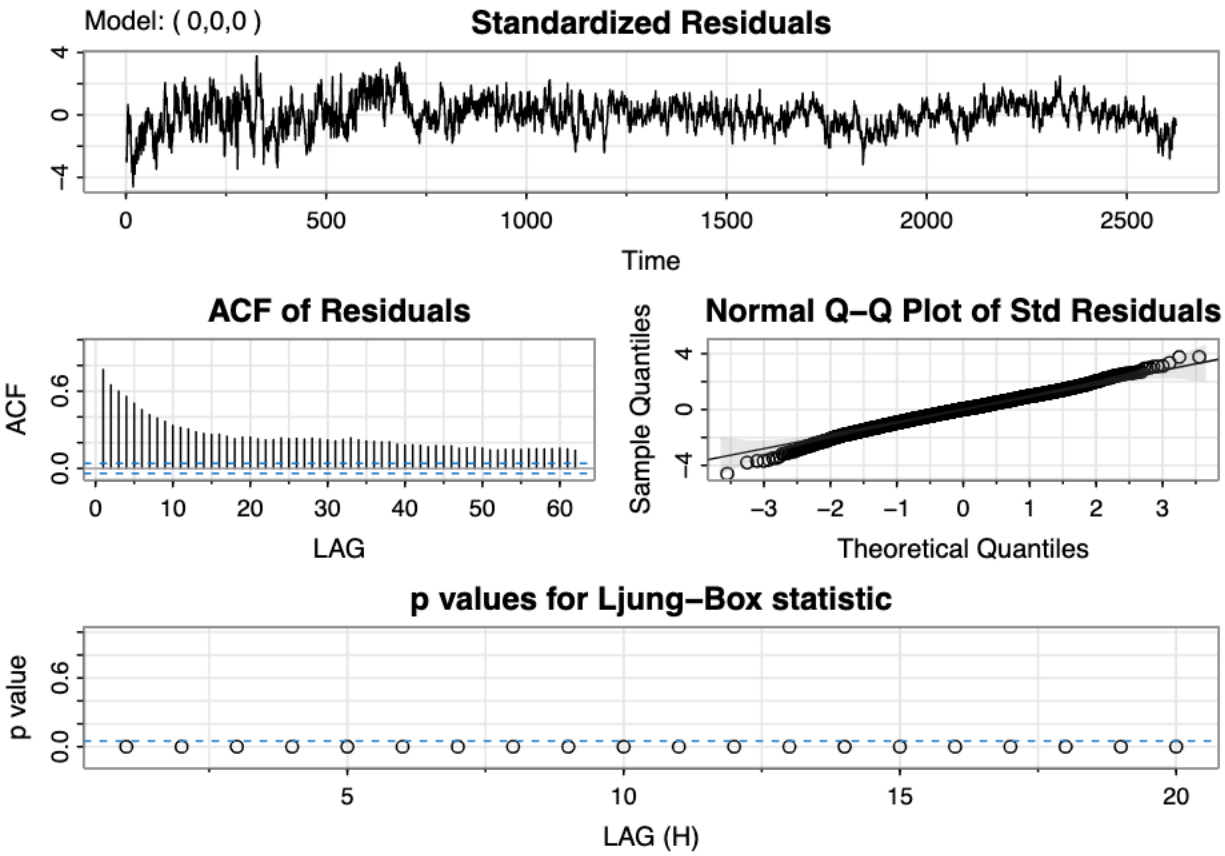
```
## <><><><><><><><><><><><><><>
##
## Coefficients:
##                      Estimate     SE t.value p.value
## ar1                    0.7160 0.6297  1.1371  0.2556
## ar2                    0.2728 0.6236  0.4375  0.6618
## ma1                    0.0717 0.6284  0.1142  0.9091
## ma2                   -0.0976 0.1261 -0.7741  0.4389
## intercept             17.7191 0.4076 43.4690  0.0000
## trend                 -0.0001 0.0001 -0.5635  0.5732
## Avg.Zonal.Winds        0.0321 0.0029 10.9856  0.0000
## Avg.Meridional.Winds  -0.0039 0.0025 -1.5919  0.1115
## Avg.Humidity           0.0002 0.0011  0.1465  0.8835
## Avg.Air.Temp           0.3808 0.0122 31.3164  0.0000
##
## sigma^2 estimated as 0.003873779 on 2613 degrees of freedom
##
## AIC = -2.705901  AICc = -2.705869  BIC = -2.681276
##
```

```r
f <- sarima.for(xdata  = ts_train, n.ahead = nrow(df_test), p = 2, d = 0, q = 2,xreg= X_train, newxreg =

ts_forecast <- ts(f$pred, start = end(ts_train)[1] + 1, frequency = 1)

ts_actual <- ts(df_test$Avg.Sea.Surface.Temp, start= end(ts_train)[1] + 1, frequency = 1)
plot(ts_train, xlim = c(start(ts_train)[1], end(ts_train)[1] + 500), ylim = range(ts_train, ts_forecast

lines(ts_forecast, col = "blue", lwd = 2)
lines(ts_actual,   col = "red",  lwd = 2)
```




```
sarima_mod_temp <- sarima(df_avg$Avg.Sea.Surface.Temp, 0,0,0, xreg=cbind(trend,df_avg$Avg.Zonal.Winds, 
```

```
## initial  value -1.888616
## iter   1 value -1.888616
## final  value -1.888616
## converged
## initial  value -1.888616
## iter   1 value -1.888616
## final  value -1.888616
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##           Estimate     SE  t.value p.value
## intercept  -0.0487 0.2119  -0.2296  0.8184
## trend       0.0000 0.0000   0.0280  0.9777
##             0.0724 0.0032  22.5290  0.0000
##             0.0646 0.0033  19.6238  0.0000
##            -0.0042 0.0016  -2.5721  0.0102
##             1.0553 0.0071 149.6425  0.0000
##
## sigma^2 estimated as 0.02288594 on 2617 degrees of freedom
##
## AIC = -0.9340181  AICc = -0.9340058  BIC = -0.9183473
##
```




```r
mse_out <- mean( (f$pred - df_test$Avg.Sea.Surface.Temp)^2 )
rmse_out <- sqrt(mse_out)

print ("Errors")
```

```
## [1] "Errors"
```

```r
c(mse_out, rmse_out)
```

```
## [1] 0.4310362 0.6565335
```

```r
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.3.3

##
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':
##
##     accuracy
```

```r
library(ModelMetrics)
```

```
## Warning: package 'ModelMetrics' was built under R version 4.3.3

##
## Attaching package: 'ModelMetrics'

## The following objects are masked from 'package:Metrics':
##
```

```
##     auc, ce, logLoss, mae, mse, msle, precision, recall, rmse, rmsle
## The following object is masked from 'package:base':
##
##     kappa
```

```r
library(forecast)

num = length(df_avg)
#model fit_1
AIC(fit_1)/num - log(2*pi) # AIC
```

```
## [1] -190.294
```

```r
BIC(fit_1)/num - log(2*pi) # BIC
```

```
## [1] -187.1321
```

```r
(AICc = log(sum(resid(fit_1)^2)/num) + (num+5)/(num-5-2)) # AICc
```

```
## [1] 4.529892
```

```r
#model fit_2
AIC(fit_2)/num - log(2*pi) # AIC
```

```
## [1] -196.8117
```

```r
BIC(fit_2)/num - log(2*pi) # BIC
```

```
## [1] -191.843
```

```r
(AICc = log(sum(resid(fit_2)^2)/num) + (num+5)/(num-5-2)) # AICc
```

```
## [1] 4.494539
```

```r
#model fit_3
AIC(fit_3)/num - log(2*pi) # AIC
```

```
## [1] -116.2481
```

```r
BIC(fit_3)/num - log(2*pi) # BIC
```

```
## [1] -114.4413
```

```r
(AICc = log(sum(resid(fit_3)^2)/num) + (num+5)/(num-5-2)) # AICc
```

```
## [1] 4.899163
```

```r
#sarima model
# AIC = -2.705856  AICc = -2.705835  BIC = -2.685708

sarima_mod$ICs["AIC"]
```

```
##       AIC
## -2.705901
```

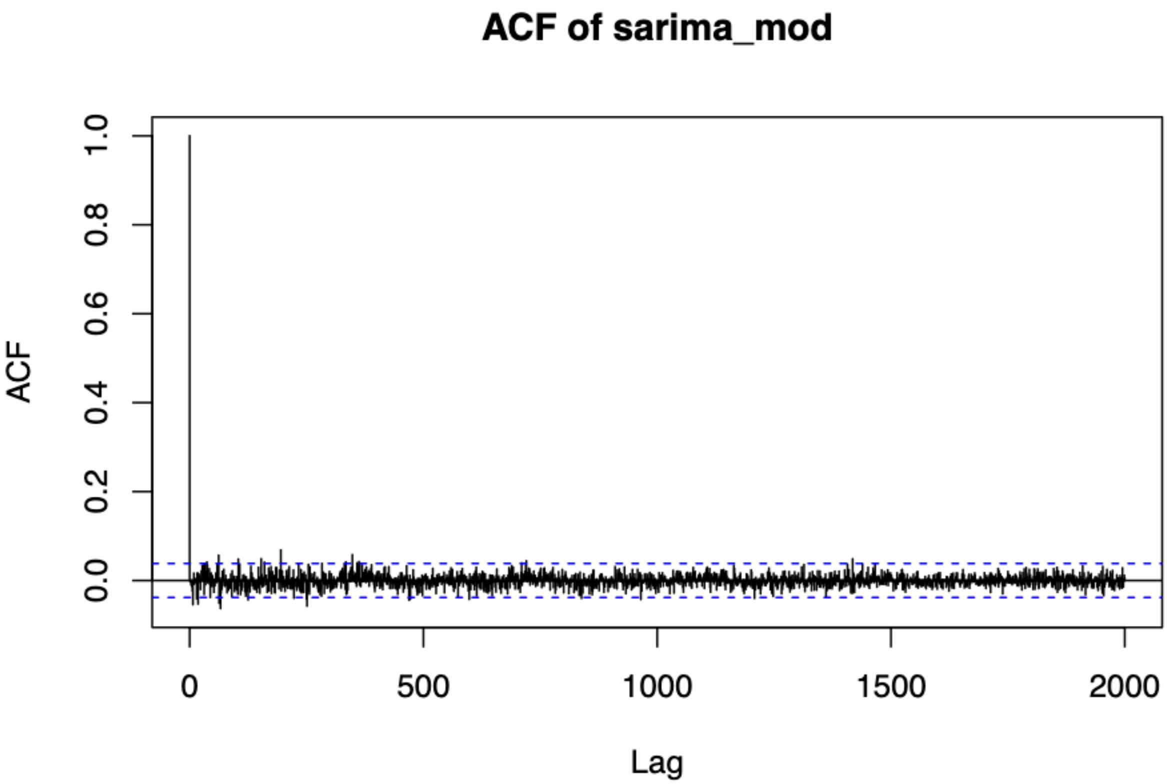
```r
sarima_mod$ICs["BIC"]
```

```
##       BIC
## -2.681276
```

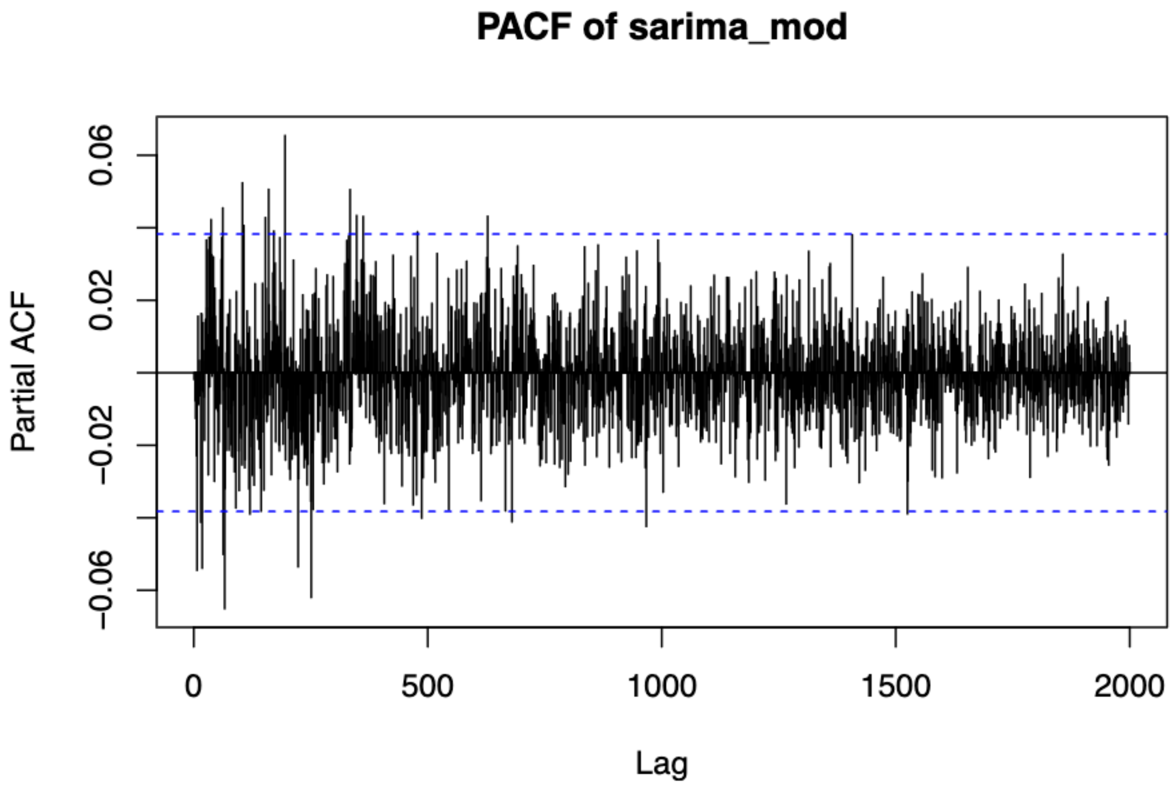
```r
sarima_mod$ICs["AICc"]
```

```
##       AICc
## -2.705869
```

```r
res <- sarima_mod$fit$residuals

acf(res, lag.max = 2000,main= "ACF of sarima_mod")
```




```r
pacf(res, lag.max = 2000,main= "PACF of sarima_mod")
```




```r
summary(fit_1)$r.squared
```

```
## [1] 0.9140624
```

```r
summary(fit_2)$r.squared
```

```
## [1] 0.9170474
```

```r
summary(fit_3)$r.squared
```

```
## [1] 0.8756759
```

```r
resid <- sarima_mod$fit$residuals
y <- df_avg$Avg.Sea.Surface.Temp
SSR<- sum(resid^2)
TSS<- sum((y - mean(y))^2)
R2_sar <- 1 - SSR/TSS
R2_sar
```

```
## [1] 0.9854538
```

```r
sarima_mod_temp$ICs["AIC"]
```

```
##        AIC
## -0.9340181
```

```r
sarima_mod_temp$ICs["BIC"]
```

```
##        BIC
## -0.9183473
```

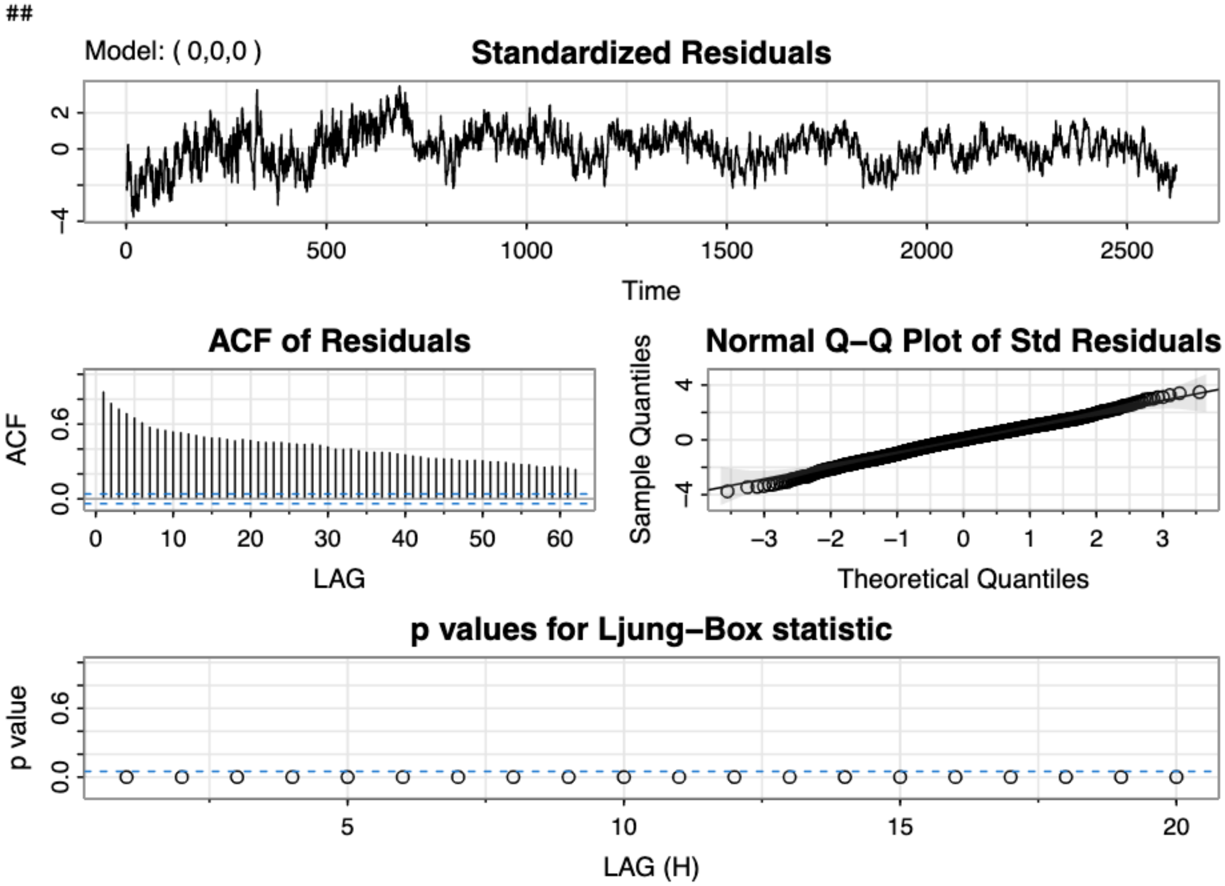
```r
sarima_mod_temp$ICs["AICc"]
```

```
##       AICc
```

```
## -0.9340058
```

```r
#sarima_mod_temp_2 <- sarima(df_avg$Avg.Sea.Surface.Temp, 0,0,0, xreg=cbind(trend, I(df_avg$Avg.Zonal.W

sarima_mod_temp_3 <- sarima(df_avg$Avg.Sea.Surface.Temp, 0,0,0, xreg=cbind(trend, df_avg$Avg.Air.Temp))
```

```
## initial  value -1.703981
## iter   1 value -1.703981
## final  value -1.703981
## converged
## initial  value -1.703981
## iter   1 value -1.703981
## final  value -1.703981
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##           Estimate     SE  t.value p.value
## intercept   1.5349 0.1967   7.8027  0.0000
## trend       0.0000 0.0000  -1.9772  0.0481
##             0.9773 0.0073 134.1525  0.0000
##
## sigma^2 estimated as 0.0331086 on 2620 degrees of freedom
##
## AIC = -0.5670351  AICc = -0.5670316  BIC = -0.5580803
##
```



```r
#5 day forecat

library(astsa)

y_train <- df_avg$Avg.Sea.Surface.Temp
ts_train <- ts(y_train, frequency = 1)
n_train <- nrow(df_avg)
n_test  <- nrow(df_test)

df_avg$trend  <- seq_len(n_train)
df_test$trend <- seq_len(n_test) + n_train
cols<- c("trend", "Avg.Zonal.Winds", "Avg.Meridional.Winds", "Avg.Humidity", "Avg.Air.Temp")
h = 5
X_train <- as.matrix(df_avg[, cols])
X_future<- as.matrix(df_test[, cols])
X_future <- X_future[1:h, , drop = FALSE]
sarima_mod <- sarima(xdata = ts_train, p =2, d =0, q =2, xreg = X_train, details = FALSE)
```

```
## <><><><><><><><><><><><><><>
##
## Coefficients:
##                      Estimate     SE t.value p.value
## ar1                    0.7160 0.6297  1.1371  0.2556
## ar2                    0.2728 0.6236  0.4375  0.6618
## ma1                    0.0717 0.6284  0.1142  0.9091
## ma2                   -0.0976 0.1261 -0.7741  0.4389
## intercept             17.7191 0.4076 43.4690  0.0000
## trend                 -0.0001 0.0001 -0.5635  0.5732
## Avg.Zonal.Winds        0.0321 0.0029 10.9856  0.0000
## Avg.Meridional.Winds  -0.0039 0.0025 -1.5919  0.1115
## Avg.Humidity           0.0002 0.0011  0.1465  0.8835
## Avg.Air.Temp           0.3808 0.0122 31.3164  0.0000
##
## sigma^2 estimated as 0.003873779 on 2613 degrees of freedom
##
## AIC = -2.705901  AICc = -2.705869  BIC = -2.681276
##
```

```r
f <- sarima.for(xdata  = ts_train, n.ahead = h, p = 2, d = 0, q = 2,xreg= X_train, newxreg = X_future, 

ts_forecast <- ts(f$pred, start = end(ts_train)[1] + 1, frequency = 1)

ts_actual <- ts(df_test$Avg.Sea.Surface.Temp[1:h], start= end(ts_train)[1] + 1, frequency = 1)

mse_out <- mean((f$pred - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_out <- sqrt(mse_out)
c(mse_out, rmse_out)
```

```
## [1] 0.003639025 0.060324329
```

```r
#mse ft_1
pred_test = predict(fit_1, newdata = df_test[1:h,])
pred_test1 = predict(fit_2, newdata = df_test[1:h,])
pred_test2 = predict(fit_3, newdata = df_test[1:h,])
```

```r
mse_1 = mean((pred_test - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_1 = sqrt(mse_1)
#mse ft_2
mse_2 = mean((pred_test2 - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_2 = sqrt(mse_2)
#mse ft_3
mse_3 = mean((pred_test3 - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_3 = sqrt(mse_3)

print("MSE, RMSE of fit_1")
```

```
## [1] "MSE, RMSE of fit_1"
```

```r
c(mse_1, rmse_1)
```

```
## [1] 0.00323616 0.05688726
```

```r
print("MSE, RMSE of fit_2")
```

```
## [1] "MSE, RMSE of fit_2"
```

```r
c(mse_2, rmse_2)
```

```
## [1] 0.02578785 0.16058595
```

```r
print("MSE, RMSE of fit_3")
```

```
## [1] "MSE, RMSE of fit_3"
```

```r
c(mse_3, rmse_3)
```

```
## [1] 1.102794 1.050140
```

```r
Box.test(resid(fit_1), type = "Ljung", lag = 2000)
```

```
##
##  Box-Ljung test
##
## data:  resid(fit_1)
## X-squared = 34996, df = 2000, p-value < 2.2e-16
```

```r
Box.test(resid(fit_2), type = "Ljung", lag = 2000)
```

```
##
##  Box-Ljung test
##
## data:  resid(fit_2)
## X-squared = 33664, df = 2000, p-value < 2.2e-16
```

```r
Box.test(resid(fit_3), type = "Ljung", lag = 2000)
```

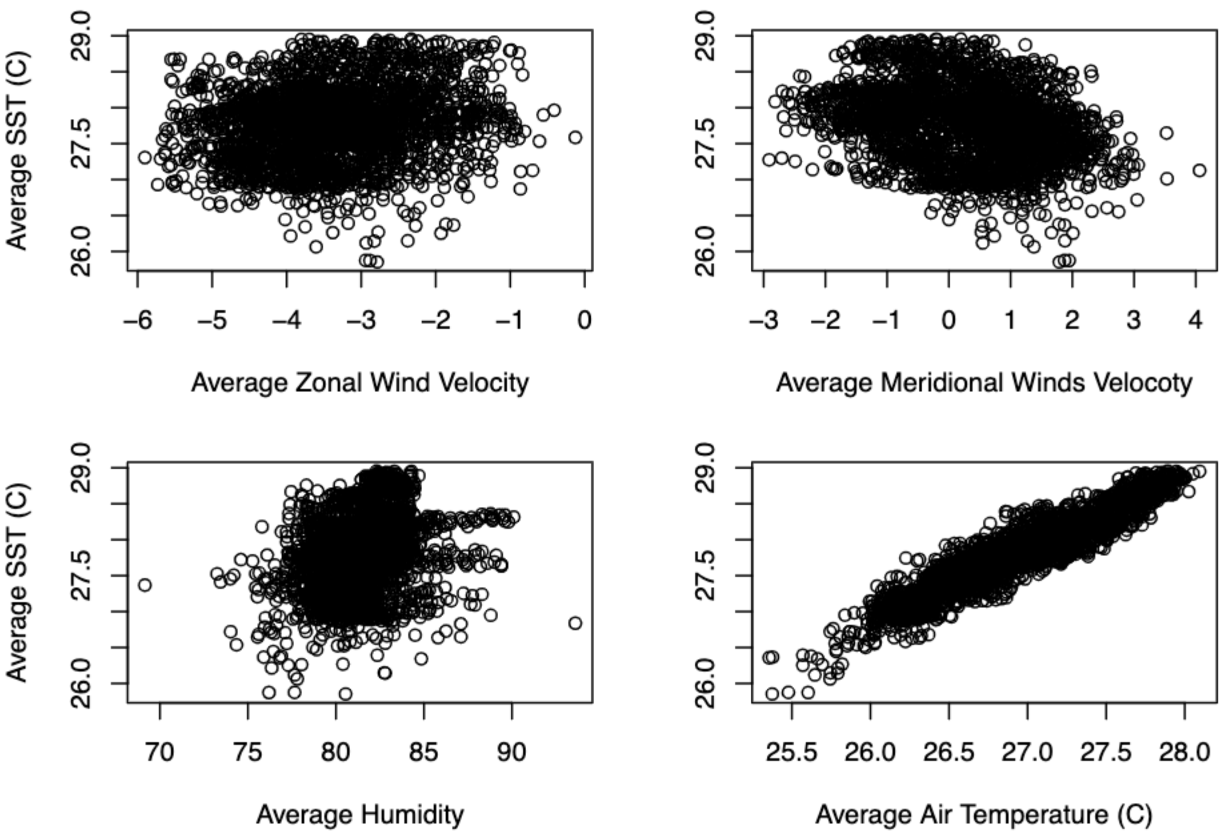
```
##
##  Box-Ljung test
##
## data:  resid(fit_3)
## X-squared = 133776, df = 2000, p-value < 2.2e-16
```

```r
Box.test(sarima_mod$fit$residuals, type = "Ljung", lag = 2000)
```

```
##
```

```
##  Box-Ljung test
##
## data:  sarima_mod$fit$residuals
## X-squared = 1974.5, df = 2000, p-value = 0.6535
```

```r
#Relevant R Code, df_avg represents the original dataset with null values dropped and all repeated date

#Plot of Average SST vs explanatory variables
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))

plot(x = df_avg$Avg.Zonal.Winds, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Zonal Wind Velocity",
plot(x = df_avg$Avg.Meridional.Winds, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Meridional Winds
plot(x = df_avg$Avg.Humidity, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Humidity", ylab = "Averag
plot(x = df_avg$Avg.Air.Temp, y = df_avg$Avg.Sea.Surface.Temp, xlab = "Average Air Temperature (C)", yla
```

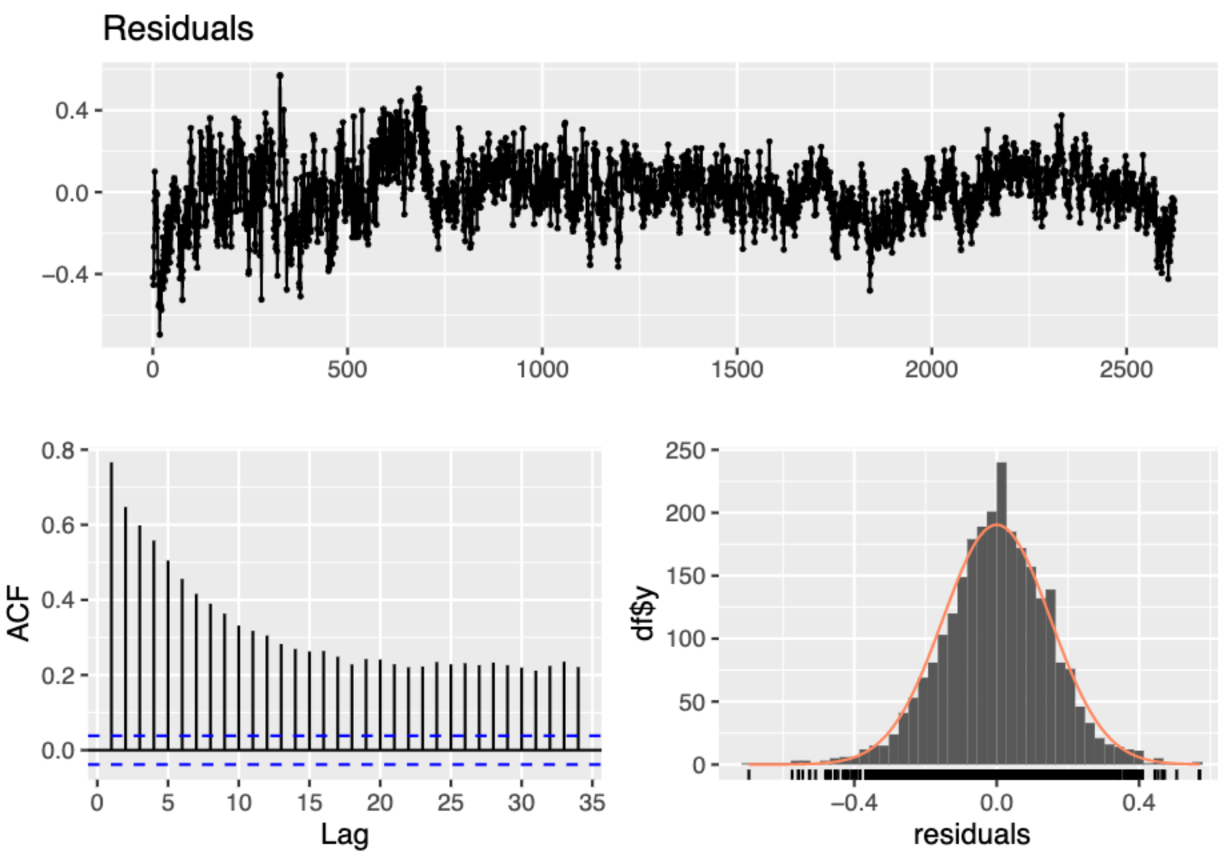



```r
trend <- time(df_avg$Avg.Sea.Surface.Temp)

#model 1
fit_1 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds + Avg.Meridional.Winds + Avg.Humidity + Avg.A
summary(fit_1)
```

```
##
## Call:
## lm(formula = Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds +
##     Avg.Meridional.Winds + Avg.Humidity + Avg.Air.Temp, data = df_avg,
##     na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.69613 -0.09442  0.00439  0.09657  0.57047
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -4.866e-02  2.122e-01  -0.229   0.8186
## trend                   5.588e-07  4.268e-06   0.131   0.8958
## Avg.Zonal.Winds         7.236e-02  3.215e-03  22.504   <2e-16 ***
## Avg.Meridional.Winds  6.463e-02  3.297e-03  19.602   <2e-16 ***
## Avg.Humidity           -4.197e-03  1.634e-03  -2.569   0.0102 *
## Avg.Air.Temp            1.055e+00  7.060e-03 149.472   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1515 on 2617 degrees of freedom
## Multiple R-squared:  0.9141, Adjusted R-squared:  0.9139
## F-statistic:  5567 on 5 and 2617 DF,  p-value: < 2.2e-16
```

```r
#residual plots and acf/pacf plots for residuals to lag 2000
checkresiduals(fit_1, test = "LB")
```

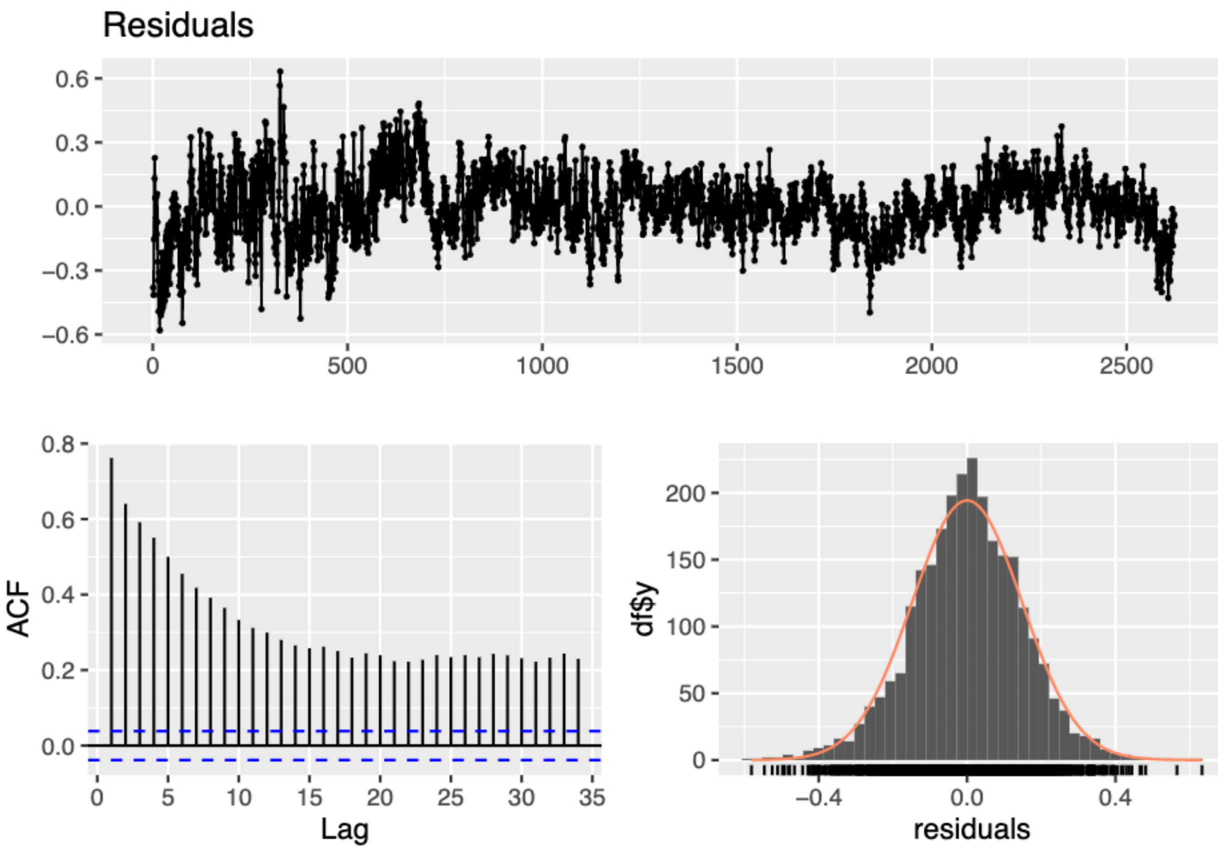



```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 7109.6, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```r
#acf2(resid(fit_1), 2000)

#model 2
df_avg$Avg.Zonal.Winds_m = df_avg$Avg.Zonal.Winds - mean(df_avg$Avg.Zonal.Winds)
df_avg$Avg.Meridional.Winds_m = df_avg$Avg.Meridional.Winds - mean(df_avg$Avg.Meridional.Winds)
df_avg$Avg.Humidity_m = df_avg$Avg.Humidity -  mean(df_avg$Avg.Humidity)
df_avg$Avg.Air.Temp_m = df_avg$Avg.Air.Temp -mean(df_avg$Avg.Air.Temp)

fit_2 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds_m + I(Avg.Zonal.Winds_m^2) + Avg.Meridional.
summary(fit_2)
```

```
##
## Call:
## lm(formula = Avg.Sea.Surface.Temp ~ trend + Avg.Zonal.Winds_m +
##     I(Avg.Zonal.Winds_m^2) + Avg.Meridional.Winds_m + I(Avg.Meridional.Winds_m^2) +
##     Avg.Humidity_m + I(Avg.Humidity_m^2) + Avg.Air.Temp_m + I(Avg.Air.Temp_m^2),
##     data = df_avg, na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58109 -0.09242  0.00271  0.09889  0.63320
##
## Coefficients:
##                               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                  2.782e+01  8.628e-03 3224.817  < 2e-16 ***
## trend                       -8.191e-06  4.500e-06   -1.820   0.0689 .
## Avg.Zonal.Winds_m            7.482e-02  3.223e-03   23.214  < 2e-16 ***
## I(Avg.Zonal.Winds_m^2)      -7.326e-03  2.391e-03   -3.064   0.0022 **
## Avg.Meridional.Winds_m       6.142e-02  3.279e-03   18.729  < 2e-16 ***
## I(Avg.Meridional.Winds_m^2) -1.225e-02  1.971e-03   -6.211 6.09e-10 ***
## Avg.Humidity_m              -3.572e-03  1.621e-03   -2.203   0.0277 *
## I(Avg.Humidity_m^2)         -3.655e-04  3.408e-04   -1.073   0.2836
## Avg.Air.Temp_m               1.052e+00  7.029e-03  149.661  < 2e-16 ***
## I(Avg.Air.Temp_m^2)         -7.008e-02  1.039e-02   -6.748 1.84e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1489 on 2613 degrees of freedom
## Multiple R-squared:  0.917,  Adjusted R-squared:  0.9168
## F-statistic:  3210 on 9 and 2613 DF,  p-value: < 2.2e-16
```

```r
#residual plots and acf/pacf plots for residuals to lag 2000
checkresiduals(fit_2, test = "LB")
```



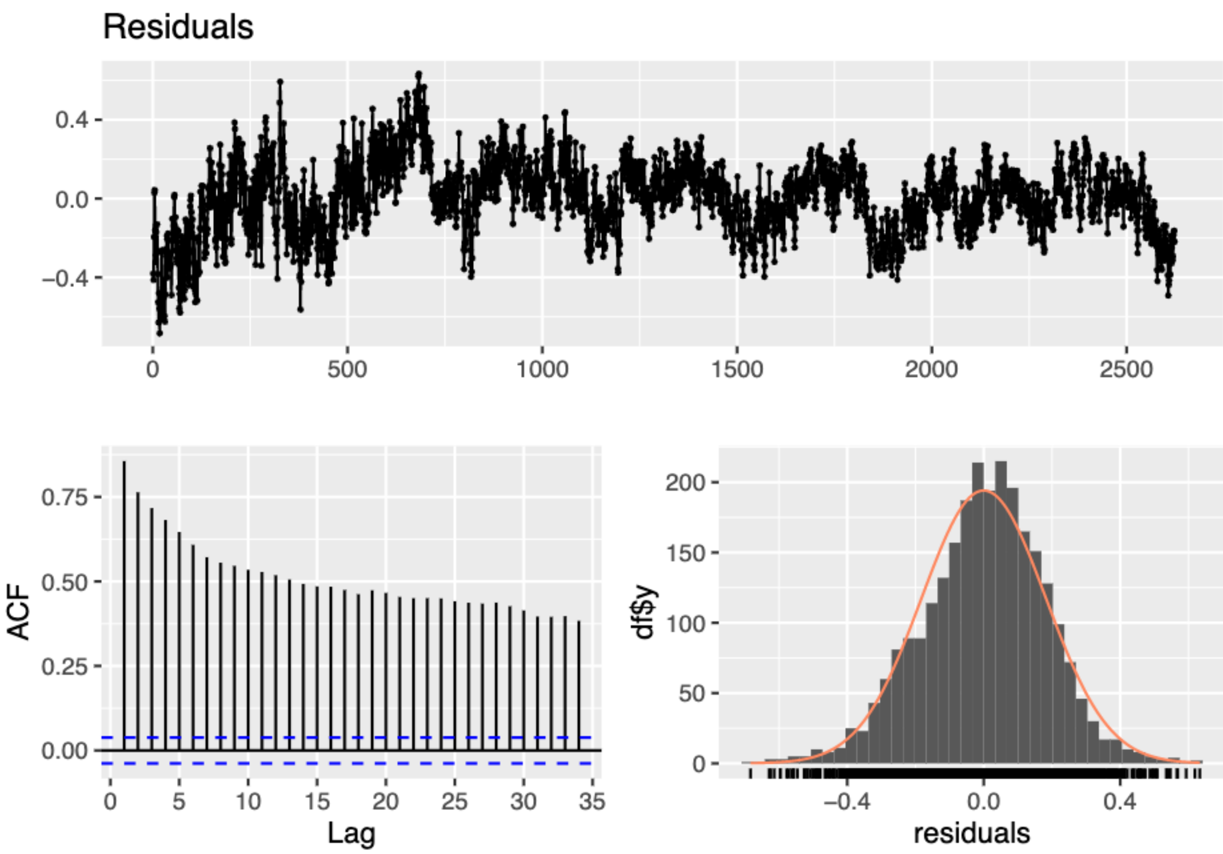

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 7021.4, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```
#acf2(resid(fit_2), 2000)

#model 3
fit_3 <- lm(Avg.Sea.Surface.Temp ~ trend + Avg.Air.Temp, data = df_avg)
summary(fit_3)
```

```
##
## Call:
## lm(formula = Avg.Sea.Surface.Temp ~ trend + Avg.Air.Temp, data = df_avg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68342 -0.11400  0.00842  0.12001  0.63301
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.535e+00  1.968e-01   7.798 8.99e-15 ***
## trend        -3.972e-05  4.718e-06  -8.418  < 2e-16 ***
## Avg.Air.Temp  9.773e-01  7.289e-03 134.076  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1821 on 2620 degrees of freedom
## Multiple R-squared:  0.8757, Adjusted R-squared:  0.8756
## F-statistic:  9227 on 2 and 2620 DF,  p-value: < 2.2e-16
```

```
checkresiduals(fit_3, test = "LB")
```



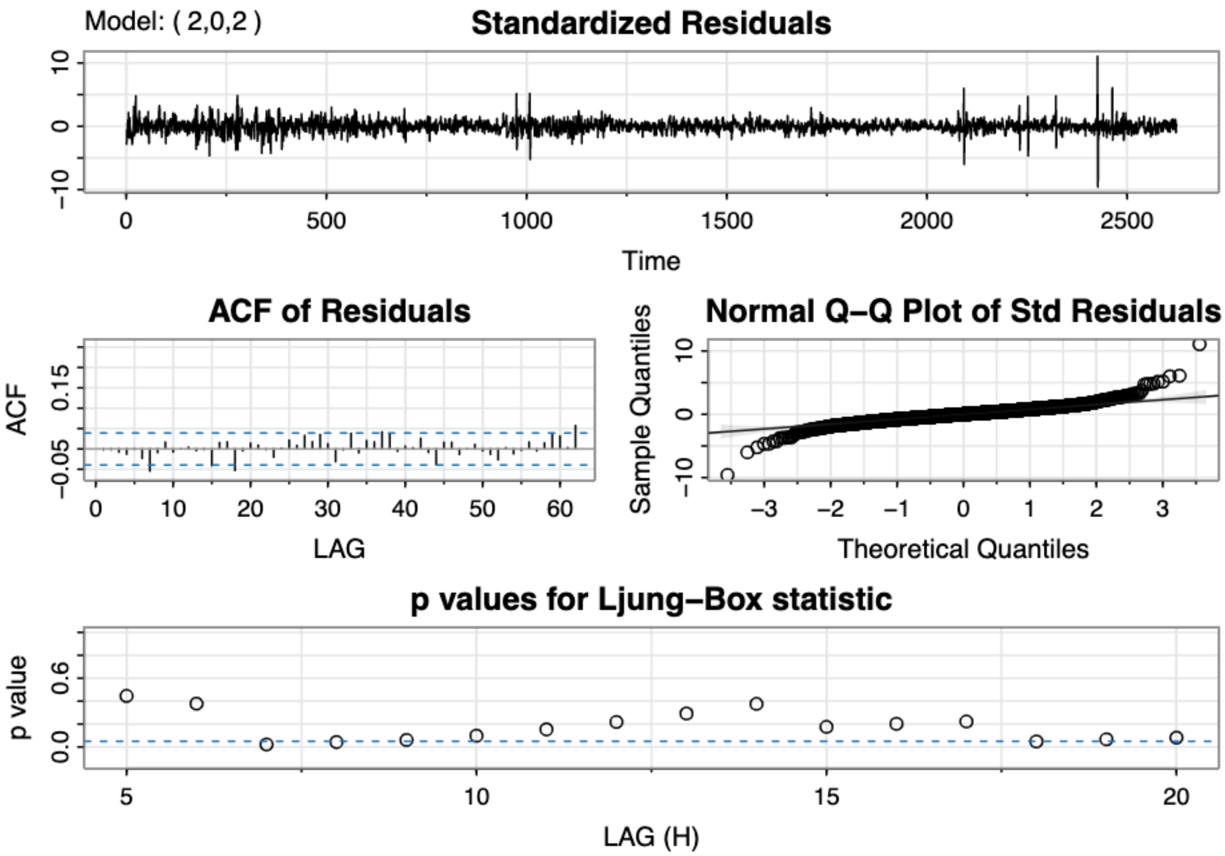

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 11307, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```
#acf2(resid(fit_3), 2000)

#model 4: correlated errors model based on model 1
sarima_mod <- sarima(df_avg$Avg.Sea.Surface.Temp, 2,0,2, xreg=cbind(trend,df_avg$Avg.Zonal.Winds, df_av
```

```
## initial  value -1.891407
## iter   2 value -2.202719
## iter   3 value -2.249806
## iter   4 value -2.342428
## iter   5 value -2.356937
## iter   6 value -2.382783
## iter   7 value -2.404413
## iter   8 value -2.417744
```

```
## iter   9 value -2.439352
## iter  10 value -2.468476
## iter  11 value -2.523376
## iter  12 value -2.555050
## iter  13 value -2.569153
## iter  14 value -2.571197
## iter  15 value -2.577946
## iter  16 value -2.585212
## iter  17 value -2.601282
## iter  18 value -2.636502
## iter  19 value -2.687405
## iter  20 value -2.702844
## iter  21 value -2.709890
## iter  22 value -2.715914
## iter  23 value -2.718160
## iter  24 value -2.718419
## iter  25 value -2.724149
## iter  26 value -2.736049
## iter  27 value -2.750666
## iter  28 value -2.756780
## iter  29 value -2.758209
## iter  30 value -2.759394
## iter  31 value -2.761095
## iter  32 value -2.764093
## iter  33 value -2.767149
## iter  34 value -2.769385
## iter  35 value -2.770433
## iter  36 value -2.771111
## iter  37 value -2.772760
## iter  38 value -2.775040
## iter  39 value -2.776937
## iter  40 value -2.777653
## iter  41 value -2.777852
## iter  42 value -2.777953
## iter  43 value -2.778075
## iter  44 value -2.778076
## iter  45 value -2.778092
## iter  46 value -2.778134
## iter  47 value -2.778141
## iter  48 value -2.778167
## iter  49 value -2.778170
## iter  50 value -2.778171
## iter  51 value -2.778171
## iter  52 value -2.778172
## iter  53 value -2.778173
## iter  54 value -2.778174
## iter  55 value -2.778174
## iter  56 value -2.778174
## iter  57 value -2.778174
## iter  58 value -2.778175
## iter  59 value -2.778176
## iter  60 value -2.778176
## iter  61 value -2.778176
## iter  62 value -2.778176
```

```
## iter  63 value -2.778177
## iter  64 value -2.778178
## iter  65 value -2.778178
## iter  66 value -2.778179
## iter  67 value -2.778179
## iter  67 value -2.778179
## iter  67 value -2.778179
## final  value -2.778179
## converged
## initial  value -2.775831
## iter   2 value -2.775862
## iter   3 value -2.775896
## iter   4 value -2.775966
## iter   5 value -2.776048
## iter   6 value -2.776079
## iter   7 value -2.776082
## iter   8 value -2.776082
## iter   9 value -2.776082
## iter  10 value -2.776083
## iter  11 value -2.776083
## iter  11 value -2.776083
## iter  11 value -2.776083
## final  value -2.776083
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##           Estimate     SE t.value p.value
## ar1         0.7160 0.6297  1.1371  0.2556
## ar2         0.2728 0.6236  0.4375  0.6618
## ma1         0.0717 0.6284  0.1142  0.9091
## ma2        -0.0976 0.1261 -0.7741  0.4389
## intercept  17.7191 0.4076 43.4690  0.0000
## trend      -0.0001 0.0001 -0.5635  0.5732
##             0.0321 0.0029 10.9856  0.0000
##            -0.0039 0.0025 -1.5919  0.1115
##             0.0002 0.0011  0.1465  0.8835
##             0.3808 0.0122 31.3164  0.0000
##
## sigma^2 estimated as 0.003873779 on 2613 degrees of freedom
##
## AIC = -2.705901  AICc = -2.705869  BIC = -2.681276
##
```




```r
#residual plots and acf/pacf plots for residuals to lag 2000
resid_mod <- sarima_mod$fit$residuals
#acf2(resid_mod, 2000)

#MSE/RMSE for models
h = 5 # change to 500 for all of df_test
pred_test = predict(fit_1, newdata = df_test[1:h,])
pred_test1 = predict(fit_2, newdata = df_test[1:h,])
pred_test2 = predict(fit_3, newdata = df_test[1:h,])

mse_1 = mean((pred_test - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_1 = sqrt(mse_1)
mse_2 = mean((pred_test2 - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_2 = sqrt(mse_2)
mse_3 = mean((pred_test3 - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_3 = sqrt(mse_3)

sarima_pred <- sarima.for(xdata = ts_train, n.ahead = h, p =2, d=0, q=2, xreg=as.matrix(df_avg[, cols])
mse_out <- mean((sarima_pred$pred - df_test$Avg.Sea.Surface.Temp[1:h])^2)
rmse_out <- sqrt(mse_out)

#Ljung Box Test
Box.test(resid(fit_1), type = "Ljung", lag = 2000)
```

```
##
##  Box-Ljung test
##
```

```
## data:  resid(fit_1)
## X-squared = 34996, df = 2000, p-value < 2.2e-16
```

```r
Box.test(resid(fit_2), type = "Ljung", lag = 2000)
```

```
##
##  Box-Ljung test
##
## data:  resid(fit_2)
## X-squared = 33664, df = 2000, p-value < 2.2e-16
```

```r
Box.test(resid(fit_3), type = "Ljung", lag = 2000)
```

```
##
##  Box-Ljung test
##
## data:  resid(fit_3)
## X-squared = 133776, df = 2000, p-value < 2.2e-16
```

```r
Box.test(sarima_mod$fit$residuals, type = "Ljung", lag = 2000)
```

```
##
##  Box-Ljung test
##
## data:  sarima_mod$fit$residuals
## X-squared = 1974.5, df = 2000, p-value = 0.6535
```