Time Series Project

Xirui Guo

5/7/2022

```r
library(ggplot2)
```

```
## Warning in register(): Can't find generic `scale_type` in package ggplot2 to
## register S3 method.
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(lubridate)
```

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

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

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```r
library(ggpubr)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5
```

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

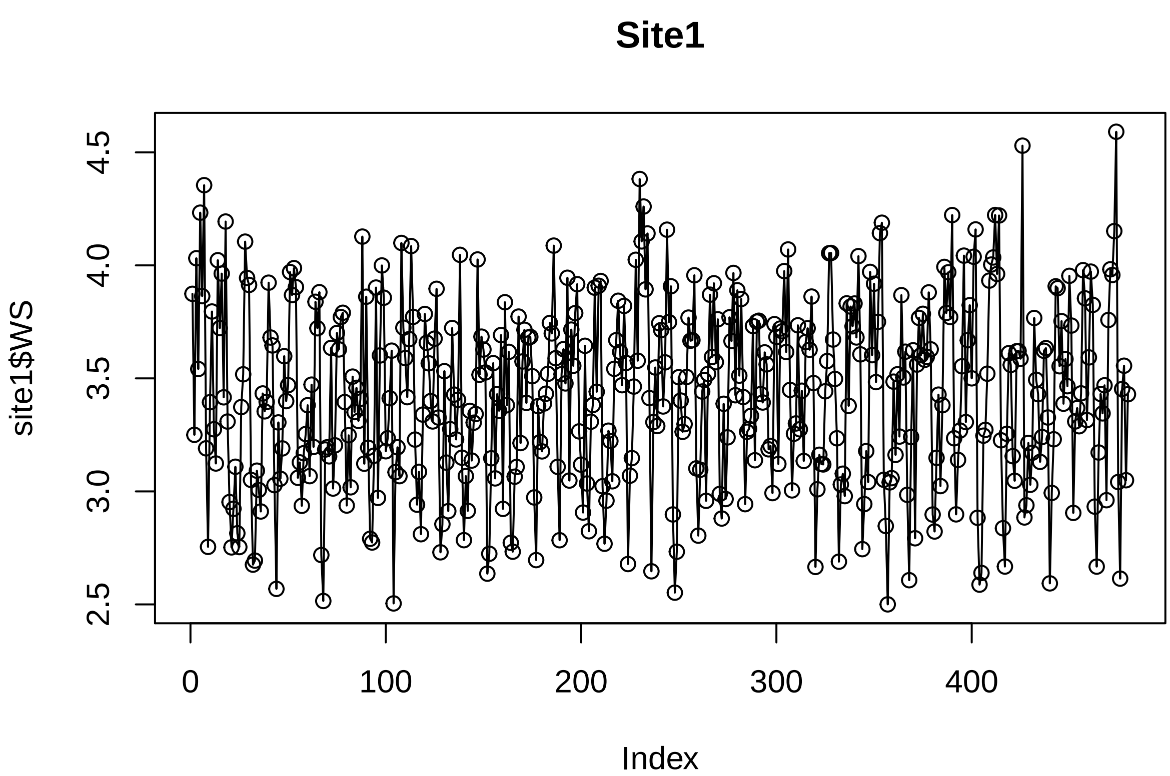
```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:ggpubr':
##
##     gghistogram
```

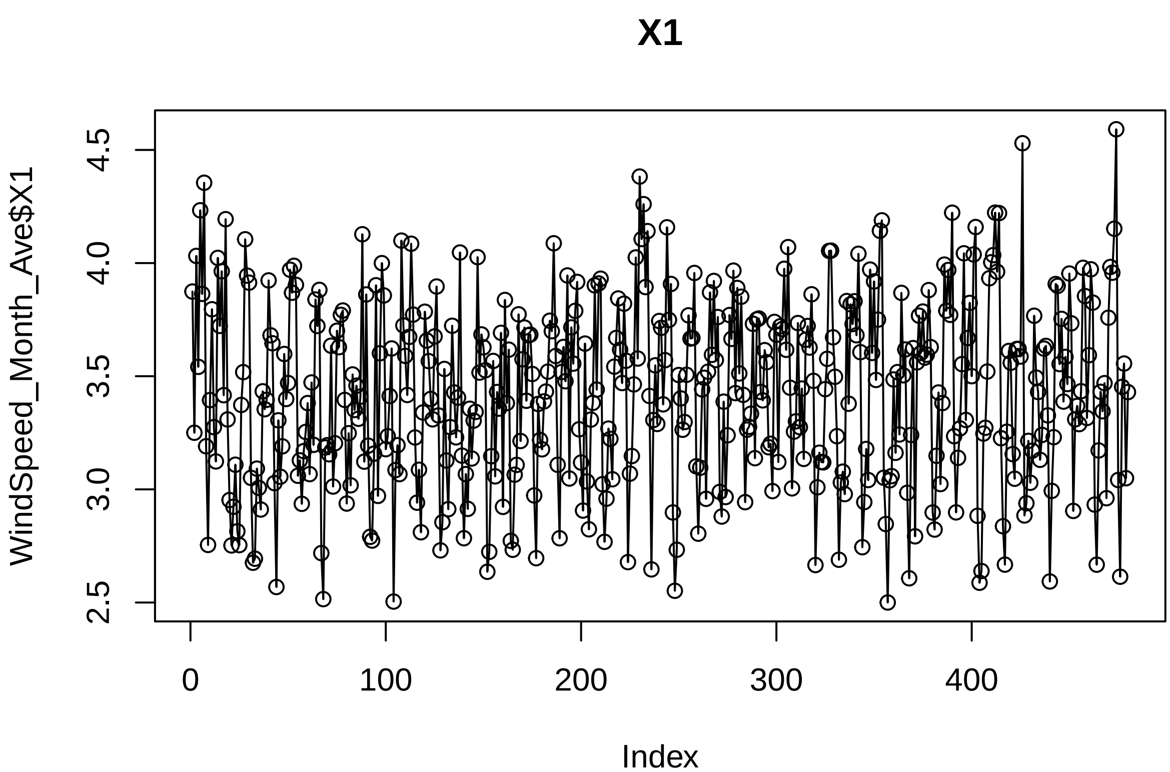
```r
WindSpeed_Month_Ave <- read.csv("WindSpeed_Month_Ave.csv")
#head(WindSpeed_Month_Ave[,1:5])
# dim(WindSpeed_Month_Ave)   # 480*918

lat_lon_WindSpeed_Month_Ave <- read.csv("WS_month_lat_lon.csv")
#head(lat_lon_WindSpeed_Month_Ave)
#tail(lat_lon_WindSpeed_Month_Ave)
# dim(lat_lon_WindSpeed_Month_Ave) # 511680*4

lat_lon_key <- read.csv("lat_lon_index_key.csv")
#head(lat_lon_key)
# dim(lat_lon_key) # 916*3
```

```r
# lat_lon_key[1,]
site1 <- lat_lon_WindSpeed_Month_Ave[lat_lon_WindSpeed_Month_Ave$lat==27.5 & lat_lon_WindSpeed_Month_Ave
plot(site1$WS,type="o",main="Site1")
```

Site1




```r
plot(WindSpeed_Month_Ave$X1,type="o",main="X1")
```



Exploratory Data Analysis

```r
set.seed(1)
sample(seq(1,916,by=1),3,replace=F)
```

```
## [1] 836 679 129
```

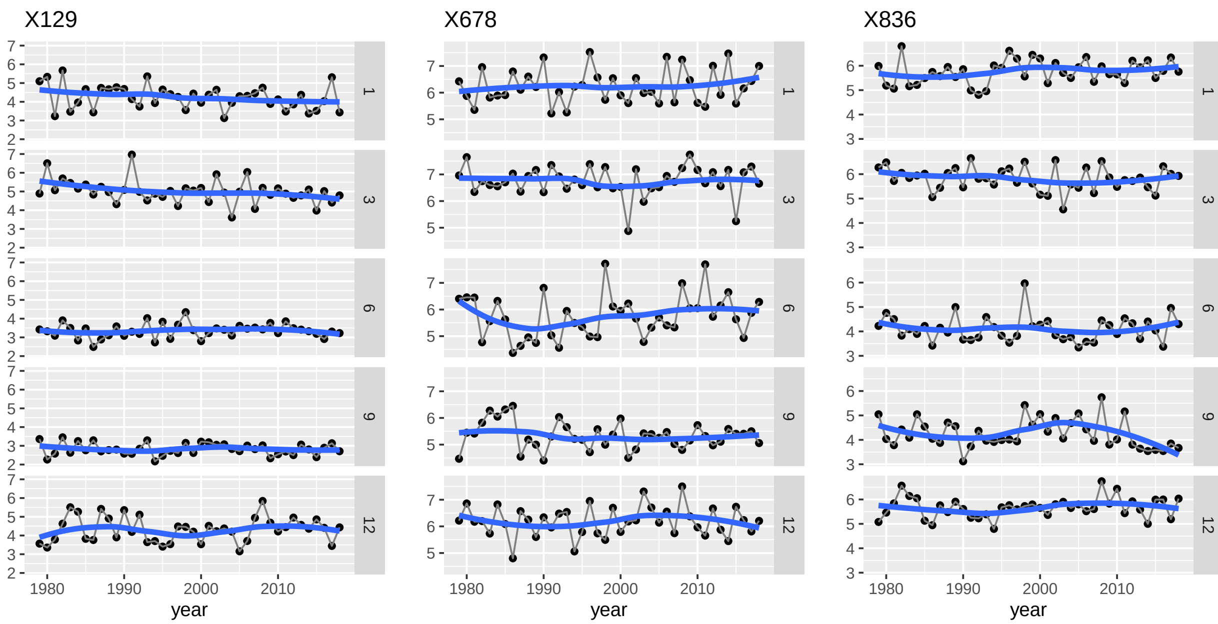
Add two columns

```r
WindSpeed_Month_Ave <- WindSpeed_Month_Ave %>%
  mutate(Date = as.Date(paste(1,month,year),format="%d %m %Y"))%>%
  mutate(Index=1:480)
```

The relationship among same month in different years

```r
data_month <- WindSpeed_Month_Ave %>%
  filter(month == 1|month ==3|month ==6|month ==9|month ==12)

e1 <- ggplot(data_month, aes(x=year, y=X129)) +
  geom_point() +
  geom_line(color = "grey50") +
  facet_grid(month~.) +
  geom_smooth(method = "loess", se = FALSE, lwd =1.5)+
  labs(title="X129",y=" ")

e2 <- ggplot(data_month, aes(x=year, y=X678)) +
  geom_point() +
  geom_line(color = "grey50") +
  facet_grid(month~.) +
  geom_smooth(method = "loess", se = FALSE, lwd =1.5)+
  labs(title="X678",y=" ")

e3 <- ggplot(data_month, aes(x=year, y=X836)) +
  geom_point() +
  geom_line(color = "grey50") +
  facet_grid(month~.) +
  geom_smooth(method = "loess", se = FALSE, lwd =1.5)+
  labs(title="X836",y=" ")

ggarrange(e1,e2,e3,ncol=3,nrow=1)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

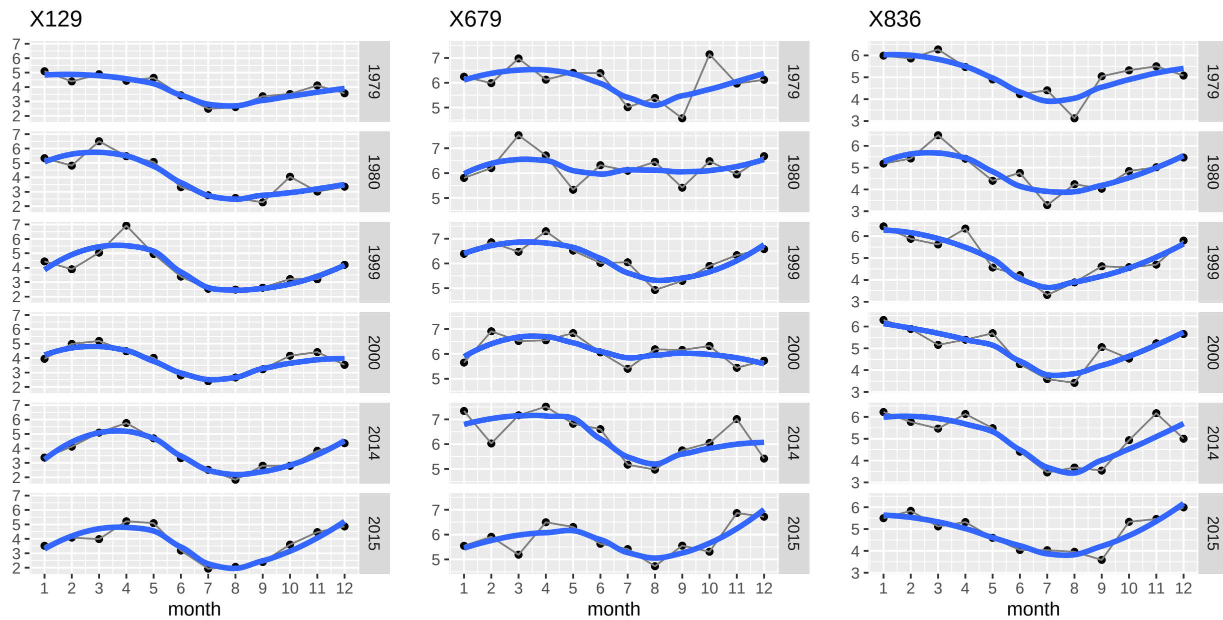


The relationship among different months within one year

```r
year_list <- c(1:24,241:264,421:444)

g1<-ggplot(WindSpeed_Month_Ave[year_list,], aes(x=month, y=X129)) +
  geom_point() +
  geom_line(color = "grey50") +
  facet_grid(year~.) +
  geom_smooth(method = "loess", se = FALSE, lwd =1.5)+
  scale_x_continuous(breaks=seq(1, 12, 1))+
  labs(title="X129",y=" ")

g2<-ggplot(WindSpeed_Month_Ave[year_list,], aes(x=month, y=X679)) +
  geom_point() +
  geom_line(color = "grey50") +
  facet_grid(year~.) +
  geom_smooth(method = "loess", se = FALSE, lwd =1.5)+
  scale_x_continuous(breaks=seq(1, 12, 1))+
  labs(title="X679",y=" ")

g3<-ggplot(WindSpeed_Month_Ave[year_list,], aes(x=month, y=X836)) +
  geom_point() +
  geom_line(color = "grey50") +
  facet_grid(year~.) +
  geom_smooth(method = "loess", se = FALSE, lwd =1.5)+
  scale_x_continuous(breaks=seq(1, 12, 1))+
  labs(title="X836",y=" ")

ggarrange(g1,g2,g3,nrow=1,ncol=3)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

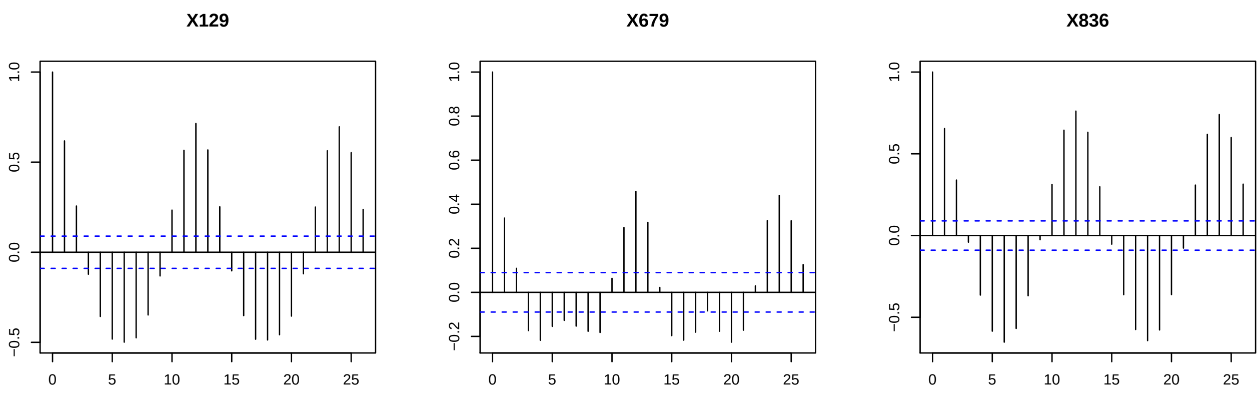



```
# d_max <- WindSpeed_Month_Ave %>%
#   filter(month==4)
#
# d_min <- WindSpeed_Month_Ave %>%
#   filter(month==8)
#
# gg1<-ggplot(WindSpeed_Month_Ave[1:60,], aes(x=Date, y=X129)) +
#   geom_point(color="grey55")+
#   geom_line(color = "grey50")+
#   geom_point(data=d_max[1:5,],aes(x=Date, y=X129),color="red")+
#   geom_label(data=d_max[1:5,],aes(x=Date+50, y=X129+0.1,label=month))+
#   geom_point(data=d_min[1:5,],aes(x=Date, y=X129),color="green")+
#   geom_label(data=d_min[1:5,],aes(x=Date+50, y=X129-0.1,label=month))+
#   ggtitle("1979-1983")
#
# gg2<-ggplot(WindSpeed_Month_Ave[241:300,], aes(x=Date, y=X129)) +
#   geom_point(color="grey55")+
#   geom_line(color = "grey50")+
#   geom_point(data=d_max[21:25,],aes(x=Date, y=X129),color="red")+
#   geom_label(data=d_max[21:25,],aes(x=Date+50, y=X129+0.1,label=month))+
#   geom_point(data=d_min[21:25,],aes(x=Date, y=X129),color="green")+
#   geom_label(data=d_min[21:25,],aes(x=Date+50, y=X129-0.1,label=month))+
#   ggtitle("1999-2003")
#
# gg3<-ggplot(WindSpeed_Month_Ave[421:480,], aes(x=Date, y=X129)) +
#   geom_point(color="grey55")+
#   geom_line(color = "grey50")+
#   geom_point(data=d_max[36:40,],aes(x=Date, y=X129), color="red")+
#   geom_label(data=d_max[36:40,],aes(x=Date+50, y=X129+0.1,label=month))+
#   geom_point(data=d_min[36:40,],aes(x=Date, y=X129),color="green")+
```

```
#   geom_label(data=d_min[36:40,],aes(x=Date+50, y=X129-0.1,label=month))+
#   ggtitle("2014-2018")
#
# ggarrange(gg1,gg2,gg3,nrow=3,ncol=1)
```

Data Trainning

```
par(mfcol=c(1,3),mar = c(3, 3, 3, 3))
a1<-acf(WindSpeed_Month_Ave$X129,main="X129")
a2<-acf(WindSpeed_Month_Ave$X679, main="X679")
a3<-acf(WindSpeed_Month_Ave$X836, main="X836")
```



De-seasonality

```
desea <- function(k,plot=FALSE){
  v<-c()
  name <- colnames(WindSpeed_Month_Ave)[k]
  value <- WindSpeed_Month_Ave[1:480,k]
  Index <- WindSpeed_Month_Ave$Index[1:480]
  for (i in seq(0,1,0.001)){
    #I(Index^3)+Index+
    m<-lm(value~I(sin(i*Index))+I(sin(2*i*Index))+I(cos(i*Index)))
    v<-c(v,summary(m)$sigma)
  }

  num <- (which(v==min(v))-1)*0.001
  m<-lm(value~I(sin(num*Index))+I(sin(2*num*Index))+I(cos(num*Index)))
  y<-residuals(m)

  if (plot==TRUE){
    main_exp1 <- paste("acf coef_i: ",num, "variable: ", name)
    main_exp2 <- paste("pacf coef_i: ",num, "variable: ", name)
    graph1 <- acf(y,main=main_exp1)
    graph2 <- pacf(y,main=main_exp2)
    return(list(y,m,graph1,graph2))
```

```
  }

  return(list(y,m))
}
```

```
par(mfcol=c(2,3),mar = c(3, 3, 3, 3))
desea(131,T)[[3]]
```

```
##
## Autocorrelations of series 'y', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.121  0.020 -0.047  0.006 -0.043 -0.014 -0.027  0.026 -0.079 -0.039
##     11     12     13     14     15     16     17     18     19     20     21
## -0.021  0.028 -0.017  0.018  0.003 -0.020 -0.101 -0.029 -0.016 -0.029 -0.043
##     22     23     24     25     26
##  0.052  0.027  0.039 -0.003  0.017
```

```
desea(681,T)[[3]]
```

```
##
## Autocorrelations of series 'y', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.016  0.094 -0.011  0.012  0.007 -0.030  0.005  0.080 -0.031  0.016
##     11     12     13     14     15     16     17     18     19     20     21
## -0.046  0.036 -0.001 -0.057 -0.058  0.004 -0.048  0.041 -0.049 -0.023 -0.020
##     22     23     24     25     26
## -0.044  0.022  0.020  0.024  0.137
```

```
desea(838,T)[[3]]
```

```
##
## Autocorrelations of series 'y', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.054  0.058  0.076  0.083  0.003 -0.018  0.040  0.005  0.099 -0.050
##     11     12     13     14     15     16     17     18     19     20     21
##  0.062  0.053  0.015 -0.089  0.024  0.068  0.003 -0.024 -0.044  0.026 -0.113
##     22     23     24     25     26
## -0.035 -0.001  0.021 -0.072  0.006
```

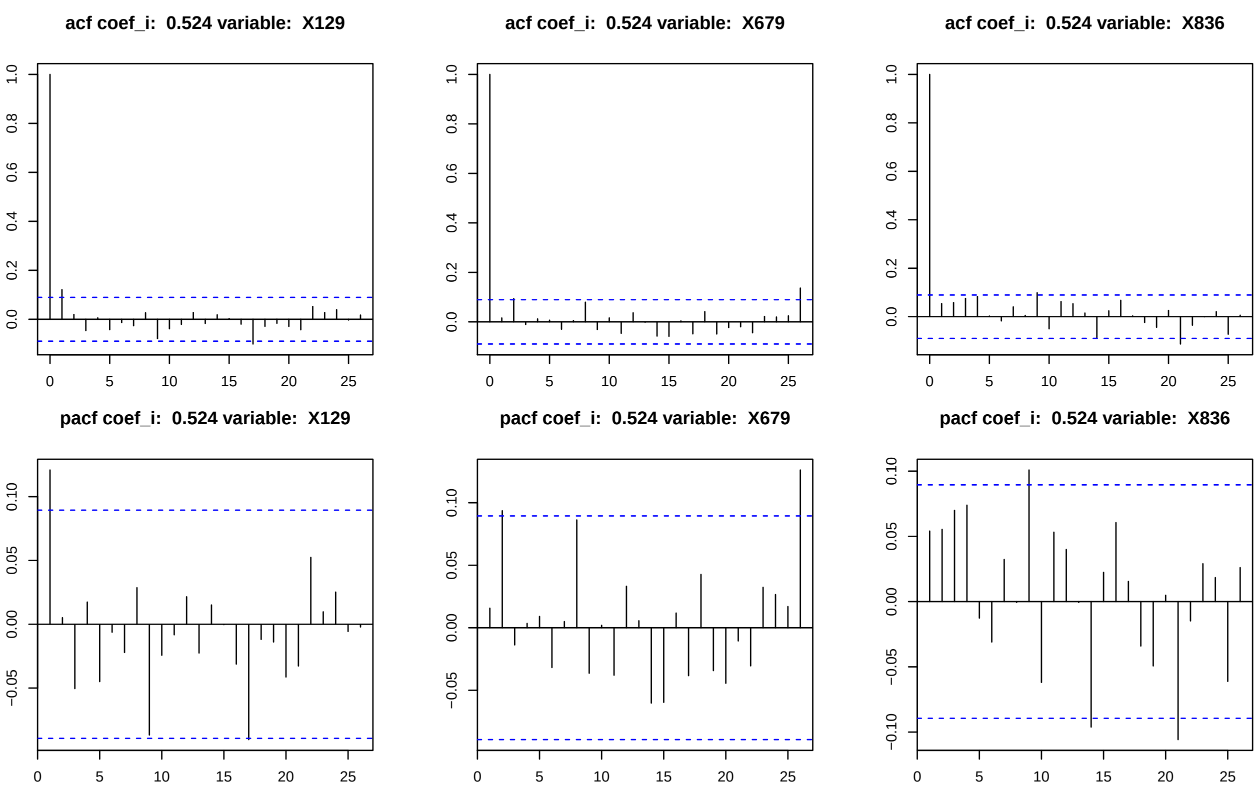
```
desea(131,T)[[4]]
```

```
##
## Partial autocorrelations of series 'y', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.121  0.005 -0.050  0.017 -0.045 -0.006 -0.022  0.029 -0.087 -0.024 -0.008
##     12     13     14     15     16     17     18     19     20     21     22
##  0.022 -0.023  0.015  0.000 -0.031 -0.090 -0.012 -0.014 -0.041 -0.033  0.052
##     23     24     25     26
##  0.010  0.025 -0.006 -0.002
```

```r
desea(681,T)[[4]]
```

```
##
## Partial autocorrelations of series 'y', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.016  0.094 -0.014  0.003  0.009 -0.032  0.005  0.086 -0.036  0.002 -0.038
##     12     13     14     15     16     17     18     19     20     21     22
##  0.033  0.006 -0.060 -0.060  0.012 -0.038  0.043 -0.034 -0.044 -0.011 -0.030
##     23     24     25     26
##  0.032  0.027  0.017  0.126
```

```r
desea(838,T)[[4]]
```




```
##
## Partial autocorrelations of series 'y', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.054  0.055  0.070  0.074 -0.013 -0.031  0.032 -0.001  0.101 -0.062  0.053
##     12     13     14     15     16     17     18     19     20     21     22
##  0.040 -0.001 -0.096  0.022  0.061  0.015 -0.034 -0.049  0.005 -0.106 -0.015
##     23     24     25     26
##  0.029  0.018 -0.061  0.026
```

Model Selection:

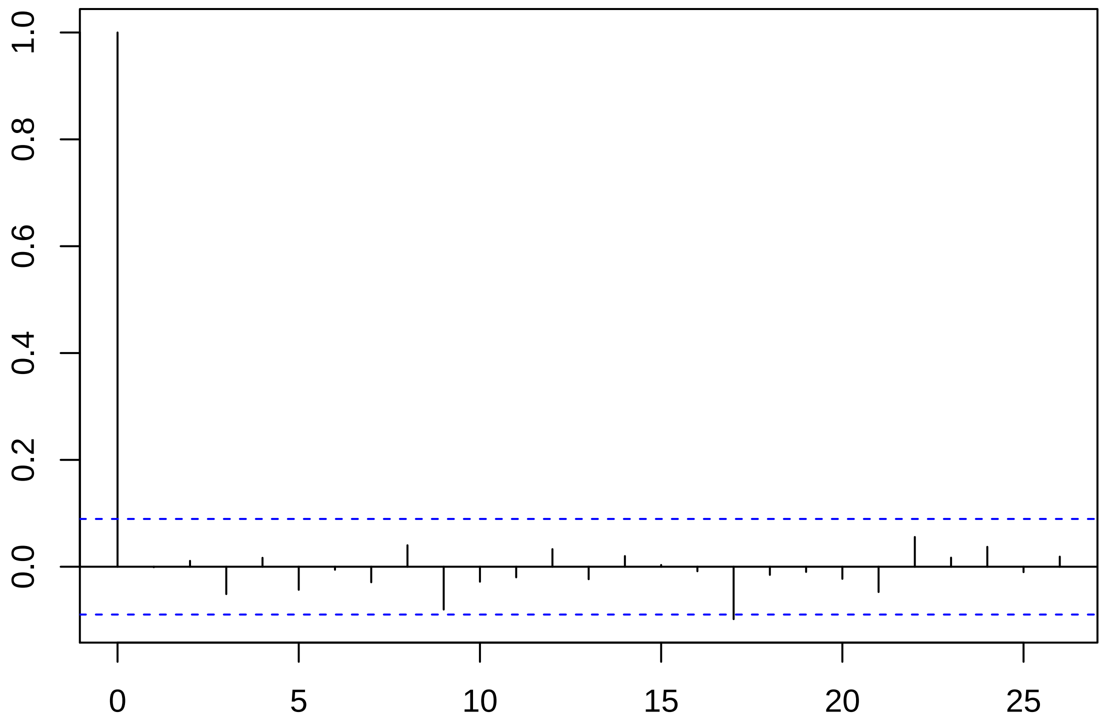
ARMA(0,0) vs AR(1) vs auto.arima

```
WindSpeed_Month_Ave1 <- read.csv("WindSpeed_Month_Ave.csv")

model_selection <- function(k,plot=FALSE){



  y <- desea(k)[[1]]
  order_1 <- c(1,0,0)
  Arma_fit_1 <- Arima(y=y,order=order_1)
  resids_1 <- Arma_fit_1$residuals
  # plot(resids,type="o",main="ARMA Resids Best Order")
  # acf(resids,main="ARMA Resids Best Order")
  # order
  # Arma_fit$aic
  # forecast(Arma_fit,h=6)

  order_0 <- c(0,0,0)
  Arma_fit_0 <- Arima(y=y,order=order_0)
  resids_0 <- Arma_fit_0$residuals

  model2 <- auto.arima(y=WindSpeed_Month_Ave1[,k])

  order2 <- c(model2$arma[1],model2$arma[6],model2$arma[2])
  seasonal2 <- c(model2$arma[3],model2$arma[7],model2$arma[4])
  period2 <- model2$arma[5]

  Arma_fit2 <- Arima(y=WindSpeed_Month_Ave1[,k],order=order2,seasonal = list(order=seasonal2, period=pe

  model_par2 <- data.frame("AR"=model2$arma[1],"MA"=model2$arma[2],"d"=model2$arma[6],
                           "SAR"=model2$arma[3],"D"=model2$arma[7],"SMA"=model2$arma[4],
                           "Period"= model2$arma[5])

  resids2 <- Arma_fit2$residuals
  #plot(resids,type="o",main="ARMA Resids Best Order")
  #acf(resids2,main="ARMA Resids Best Order")

  res_col <- paste("ARIMA", model2$arma[1],model2$arma[6],model2$arma[2],
                   model2$arma[3],model2$arma[7],model2$arma[4],model2$arma[5])

  df_result <- data.frame("ARMA(0,0,0)" =
                           c(Arma_fit_0$aic,Arma_fit_0$aicc,Arma_fit_0$bic),
                           "ARMA(1,0,0)" =
                           c(Arma_fit_1$aic,Arma_fit_1$aicc,Arma_fit_1$bic),
                         "unprocessed variable" =
                           c(Arma_fit2$aic,Arma_fit2$aicc,Arma_fit2$bic),
                         row.names = c("AIC","AICC","BIC"))
  colnames(df_result) <- c("ARMA(0,0)","AR(1)",res_col)
```

```r

  min_aic = min(Arma_fit_0$aic,Arma_fit_1$aic,Arma_fit2$aic)

  if (plot==TRUE & min_aic==Arma_fit2$aic){
    # same output as auto.arima$arma
    main_exp1 <- paste("ARIMA:", model2$arma[1],model2$arma[6],model2$arma[2],
                       model2$arma[3],model2$arma[7],model2$arma[4],model2$arma[5])
    graph1 <- acf(resids2, main=main_exp1)
    return(list(Arma_fit2,df_result,model_par2,graph1))
  }
  else if (plot==TRUE & min_aic==Arma_fit_1$aic){
    main_exp2 <- paste("ARMA:", 1, 0, 0)
    graph2 <- acf(resids_1,main = main_exp2)
    return(list(Arma_fit_1,df_result,order_1,graph2))
  }
  else if (plot==TRUE & min_aic==Arma_fit_0$aic){
    main_exp2 <- paste("ARMA:", 0, 0, 0)
    graph2 <- acf(resids_0,main = main_exp2)
    return(list(Arma_fit_0,df_result,order_0,graph2))
  }
  else if (plot==FALSE & min_aic==Arma_fit2$aic){
    return(list(Arma_fit2,df_result,model_par2))
  }
  else if (plot==FALSE & min_aic==Arma_fit_1$aic){
    return(list(Arma_fit_1,df_result,order_1))
  }
  else if (plot==FALSE & min_aic==Arma_fit_0$aic){
    return(list(Arma_fit_0,df_result,order_0))
  }

}
```

```r
par(mar=c(3,3,3,3))
model_selection(131,T)
```

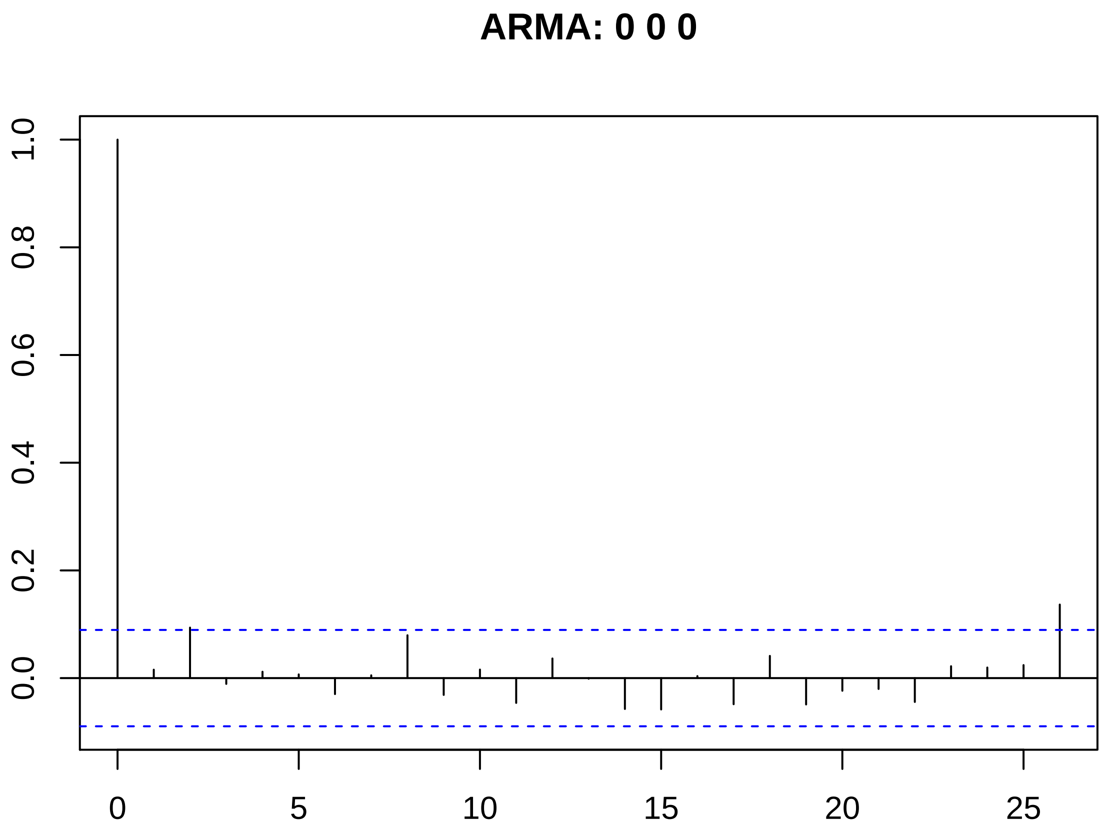
ARMA: 1 0 0




```
## [[1]]
## Series: y
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.1212  0.0003
## s.e.  0.0454  0.0280
##
## sigma^2 = 0.292:  log likelihood = -384.67
## AIC=775.34   AICc=775.39   BIC=787.86
##
## [[2]]
##      ARMA(0,0)    AR(1) ARIMA 5 0 1 0 0 0 1
## AIC   780.4260 775.3373            1017.079
## AICC  780.4511 775.3877            1017.384
## BIC   788.7735 787.8587            1050.469
##
## [[3]]
## [1] 1 0 0
##
## [[4]]
##
## Autocorrelations of series 'resids_1', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000 -0.001  0.011 -0.051  0.017 -0.043 -0.006 -0.029  0.040 -0.080 -0.028
##     11     12     13     14     15     16     17     18     19     20     21
```

```
## -0.020  0.033 -0.023  0.020  0.003 -0.008 -0.098 -0.015 -0.010 -0.023 -0.047
##     22     23     24     25     26
##  0.056  0.017  0.037 -0.010  0.019
```

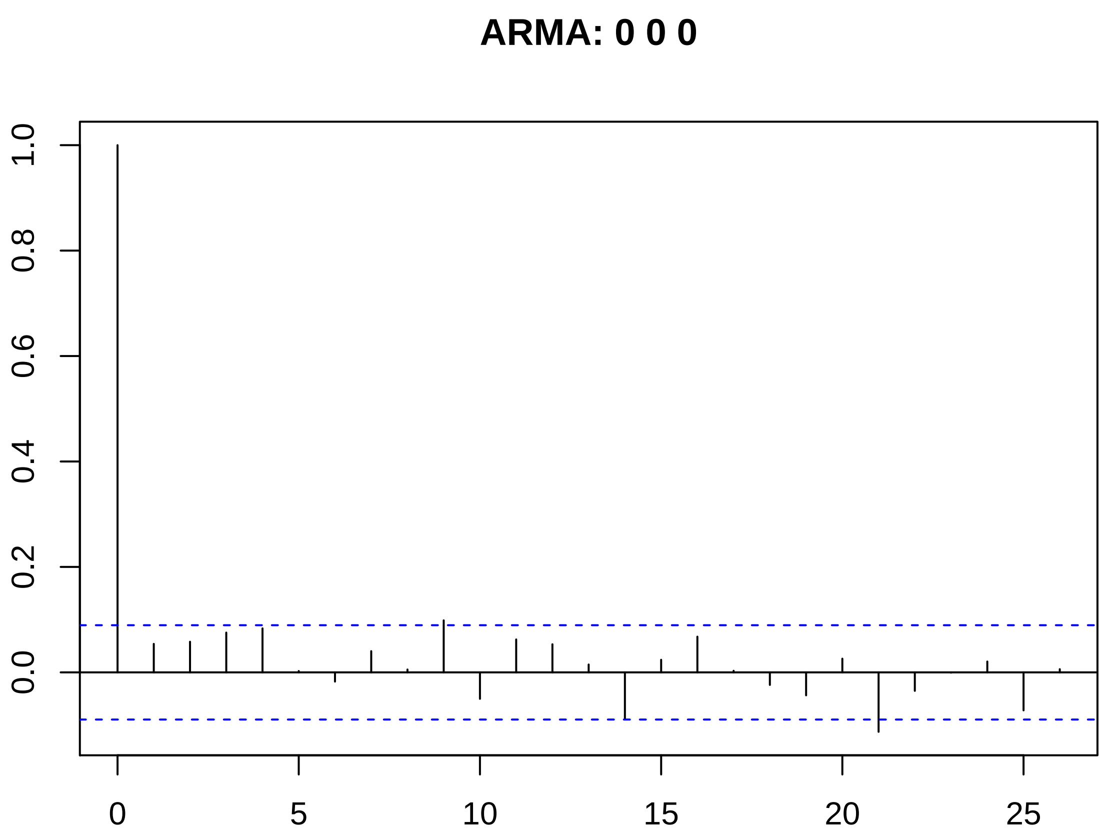
```r
par(mar=c(3,3,3,3))
model_selection(681,T)
```




```
## [[1]]
## Series: y
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.0000
## s.e.  0.0271
##
## sigma^2 = 0.3527:  log likelihood = -430.49
## AIC=864.98   AICc=865   BIC=873.32
##
## [[2]]
##      ARMA(0,0)    AR(1) ARIMA 3 0 1 0 0 0 1
## AIC   864.9750 866.8573            1061.002
## AICC  865.0002 866.9077            1061.180
## BIC   873.3226 879.3787            1086.045
##
## [[3]]
## [1] 0 0 0
```

```
##
## [[4]]
##
## Autocorrelations of series 'resids_0', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.016  0.094 -0.011  0.012  0.007 -0.030  0.005  0.080 -0.031  0.016
##     11     12     13     14     15     16     17     18     19     20     21
## -0.046  0.036 -0.001 -0.057 -0.058  0.004 -0.048  0.041 -0.049 -0.023 -0.020
##     22     23     24     25     26
## -0.044  0.022  0.020  0.024  0.137
```

```r
par(mar=c(3,3,3,3))
model_selection(838,T)
```




```
## [[1]]
## Series: y
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##        mean
##       0.000
## s.e.  0.021
##
## sigma^2 = 0.2121:  log likelihood = -308.41
## AIC=620.81   AICc=620.84   BIC=629.16
##
```

```
## [[2]]
##      ARMA(0,0)    AR(1) ARIMA 5 0 1 0 0 0 1
## AIC   620.8138 621.4080            865.7558
## AICC  620.8389 621.4584            866.0616
## BIC   629.1613 633.9293            899.1461
##
## [[3]]
## [1] 0 0 0
##
## [[4]]
##
## Autocorrelations of series 'resids_0', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.054  0.058  0.076  0.083  0.003 -0.018  0.040  0.005  0.099 -0.050
##     11     12     13     14     15     16     17     18     19     20     21
##  0.062  0.053  0.015 -0.089  0.024  0.068  0.003 -0.024 -0.044  0.026 -0.113
##     22     23     24     25     26
## -0.035 -0.001  0.021 -0.072  0.006
```

```
# auto.arima(y=WindSpeed_Month_Ave1[,131])
# auto.arima(y=WindSpeed_Month_Ave1[,681])
# auto.arima(y=WindSpeed_Month_Ave1[,838])
```

Conclusion

```
prediction <- function(s=3,e=918){
  result <- data.frame("Date"=as.Date(c("2019-01-01","2019-02-01","2019-03-01",
                                        "2019-04-01","2019-05-01","2019-06-01")))

  for(j in s:e){
    model <- model_selection(j)[[1]]

    var_1 <- model_selection(j)[[3]][1]

    if (var_1 ==1){
      n <- ncol(result)
      fore_data <- predict(desea(j)[[2]],data.frame(Index=c(481:486)))+
      data.frame(forecast(model,h=6))$Point.Forecast
      result[,(n+1)]<- fore_data
    } else{
      n <- ncol(result)
      fore_data <- predict(desea(j)[[2]],data.frame(Index=c(481:486)))
      result[,(n+1)]<- fore_data
    }

  }
  colnames(result)[2:(e-s+2)] <-colnames(WindSpeed_Month_Ave)[s:e]
  return(result)
}
```

Predicted Results

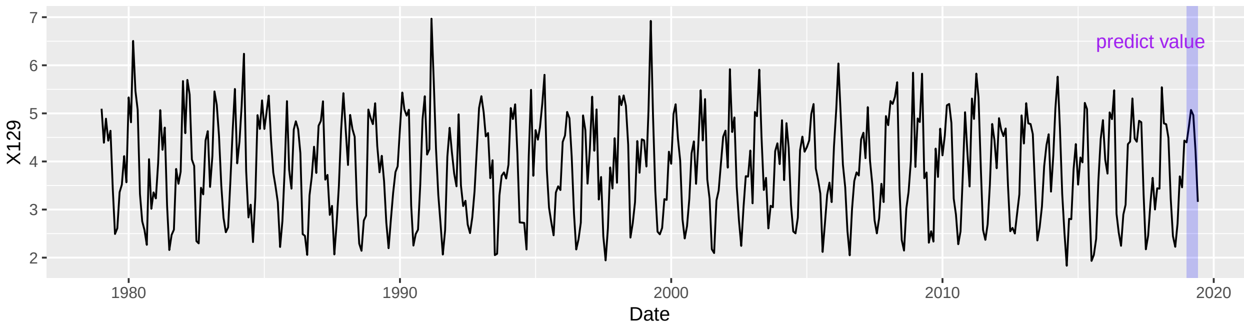
```r
p1<-prediction(131,131)
p2<-prediction(681,681)
p3<-prediction(838,838)

m1<- merge(p1,p2,by="Date")
merge(m1,p3,by="Date")
```

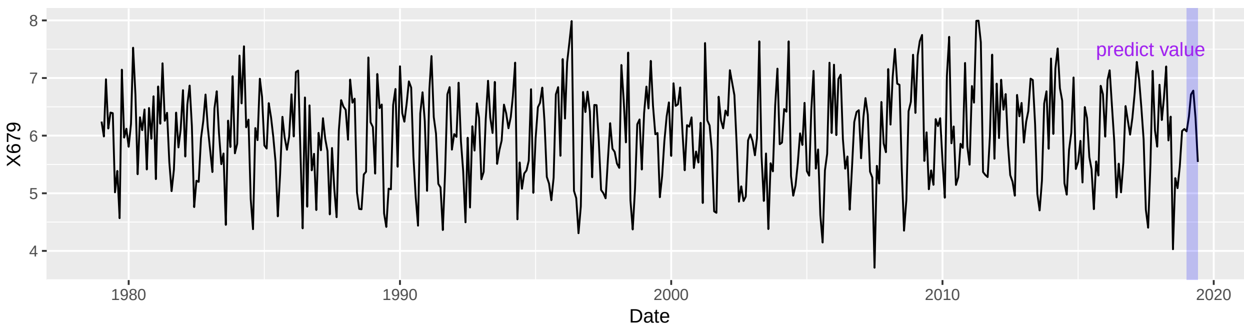
```
##         Date     X129     X679     X836
## 1 2019-01-01 4.396814 6.073694 5.749882
## 2 2019-02-01 4.730077 6.333063 5.844623
## 3 2019-03-01 5.071447 6.715117 5.839573
## 4 2019-04-01 4.962324 6.780959 5.509089
## 5 2019-05-01 4.213535 6.304962 4.801518
## 6 2019-06-01 3.159696 5.543563 3.993592
```

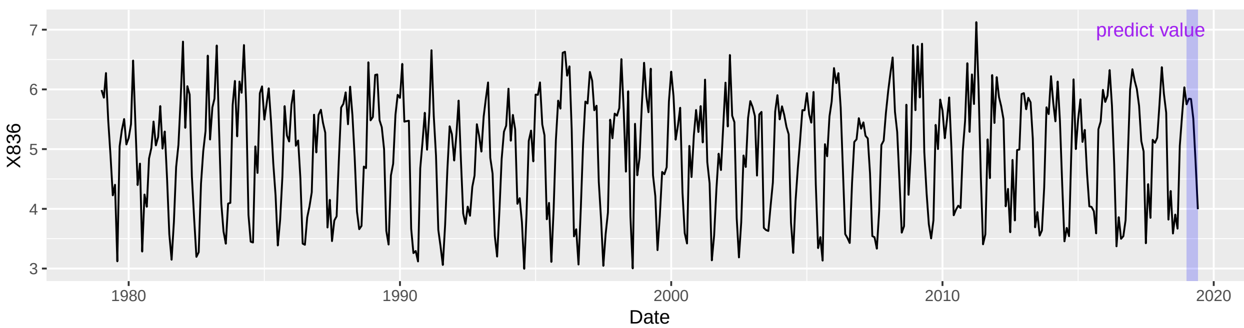
The Graph of Predicted Results

```r
WindSpeed_Month_Ave2 <- WindSpeed_Month_Ave1 %>%
  mutate(Date = as.Date(paste(1,month,year),format="%d %m %Y"))
```

```r
# X129
all_data <- rbind(WindSpeed_Month_Ave2[,c(919,131)],prediction(131,131))
res1<-ggplot(all_data,aes(Date,X129)) +
  geom_line()+
  annotate("rect", xmin = as.Date("2019-01-01"), xmax = as.Date("2019-06-01"),
           ymin = -Inf, ymax = Inf, fill = "blue",alpha = .2)+
  annotate("text", x = as.Date("2015-09-01"), y = 6.5, label ="predict value",
             color = "purple", hjust = 0)

# X679
all_data2 <- rbind(WindSpeed_Month_Ave2[,c(919,681)],prediction(681,681))
res2<-ggplot(all_data2,aes(Date,X679)) +
  geom_line()+
  annotate("rect", xmin = as.Date("2019-01-01"), xmax = as.Date("2019-06-01"),
           ymin = -Inf, ymax = Inf, fill = "blue",alpha = .2)+
  annotate("text", x = as.Date("2015-09-01"), y = 7.5, label ="predict value",
             color = "purple", hjust = 0)

# X836
all_data <- rbind(WindSpeed_Month_Ave2[,c(919,838)],prediction(838,838))
res3<-ggplot(all_data,aes(Date,X836)) +
  geom_line()+
  annotate("rect", xmin = as.Date("2019-01-01"), xmax = as.Date("2019-06-01"),
           ymin = -Inf, ymax = Inf, fill = "blue",alpha = .2)+
  annotate("text", x = as.Date("2015-09-01"), y = 7, label ="predict value",
             color = "purple", hjust = 0)

ggarrange(res1,res2,res3,nrow=3,ncol=1)
```








```
# the prediction code for directly using auto.arima
# prediction1 <- function(s=3,e=918,p,q){
#   result <- data.frame("Date"=as.Date(c("2019-01-01","2019-02-01","2019-03-01",
#                                          "2019-04-01","2019-05-01","2019-06-01")))
#   for(j in s:e){
#     model <- model_selection(p,q,j)[[1]]
#     n <- ncol(result)
#     result[,(n+1)]<- data.frame(forecast(model,h=6))$Point.Forecast
#   }
#   colnames(result)[2:(e-s+2)] <-colnames(WindSpeed_Month_Ave)[s:e]
#   return(result)
# }
# prediction1(131,131,1,0)
# plot(forecast(model_selection(1,0,131)[[1]],h=6))
```