Assignment Task 23 Jagannath

> library(readr)

> AAPL <- read\_csv("~/assignment data acadgild/assignmnet 21-24/AAPL.csv")

Parsed with column specification:

cols(

Date = col\_character(),

Open = col\_double(),

High = col\_double(),

Low = col\_double(),

Close = col\_double(),

`Adj Close` = col\_double(),

Volume = col\_double()

)

> View(AAPL)

> #Answers

> #\*\*\*\*NOTE\*\*\*\*

> #AAPL is my dataset file

> df<- AAPL

> df$Date <- as.Date(df$Date)

> data = ts(df$Close)

> test = data[60:76]

> data = data[1:59]

> plot(data, main= "Daily Close Price")

> class(data)

[1] "numeric"

> #This tells you that the data series is in a time series format

> start(data)

[1] 1 1

> #This is the start of the time series

> end(data)

[1] 59 1

> #This is the end of the time series

> frequency(data)

[1] 1

> #The

> summary(data)

Min. 1st Qu. Median Mean 3rd Qu. Max.

142 155 165 169 179 210

> plot(data)

> #This will plot the time series

> abline(reg=lm(data~time(data)))

> boxplot(data~cycle(data))

> data = ts(df$Close, frequency = 10)

> plot(data, main = "Daily Close Price")

> decompose(data)

$x

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] 207 202 204 210 208 204 194 192 187 191 194 186 177 177 172 175 174 181 180 179 185 177 175 168

[25] 170 169 169 171 165 164 166 161 157 151 147 157 156 156 158 158 142 148 148 151 153 154 152 150

[49] 153 155 156 157 153 154 153 158 156 155 165 166 167 171 174 174 171 170 169 171 170 171 170 171

[73] 172 171 173 174 174 175 173 175 176 176 175 172 173 179 181 182 184 186 188 187 188 195 191 189

[97] 187 188 189 190 191 194 195 196 197 200 200 201 199 199 199 199 203 204 205 207 207 205 204 205

[121] 201 211 209 212 208 203 203 201 197 186 189 191 190 189 183 187 183 180 179 178 177 178 175 173

[145] 180 183 185 190 193 195 194 194 193 194 198 198 199 199 199 196 200 200 198 202 203 204 204 200

[169] 201 203 202 203 205 204 203 206 203 207 209 209 207 208 210 209 213 208 204 193 197 199 203 201

[193] 200 209 203 202 206 210 210 213 212 203 206 204 206 209 209 206 209 213 213 214 217 224 223 219

[217] 220 221 223 221 218 219 218 221 220 219 224 225 219 221 227 227 224 227 230 236 236 235 234 235

[241] 236 241 240 243 244 247 249 243 243 249 256

$seasonal

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642

[14] 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855

[27] 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155

[40] 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083

[53] -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472

[66] 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158

[79] -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232

[92] 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015

[105] -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129

[118] -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113

[131] 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642

[144] 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855

[157] 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155

[170] 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083

[183] -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472

[196] 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158

[209] -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232

[222] 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015

[235] -0.472 0.855 0.129 -0.158 -0.155 0.113 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129

[248] -0.158 -0.155 0.113 0.232

$trend

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] NA NA NA NA NA 199 198 196 193 189 186 183 182 181 180 179 178 177 177 176 176 175 175 173

[25] 172 170 169 167 165 163 161 160 159 158 157 155 154 153 152 152 153 152 152 151 151 151 152 153

[49] 154 154 154 154 155 156 157 158 159 161 163 165 166 168 169 170 171 171 171 171 171 171 171 172

[73] 172 172 173 173 174 174 174 174 175 175 176 177 178 179 180 181 183 185 187 187 188 189 189 189

[97] 190 191 191 191 192 193 195 196 197 198 198 199 200 200 201 202 203 203 204 204 205 205 206 207

[121] 207 206 206 205 204 202 201 199 197 194 192 190 188 186 185 184 183 182 180 179 179 179 179 181

[145] 182 184 185 187 189 191 193 194 195 196 196 197 197 198 198 199 200 200 200 201 201 202 202 202

[169] 203 203 203 203 203 204 205 205 206 206 207 207 208 208 208 206 205 205 204 203 203 202 202 201

[193] 202 204 205 206 207 207 207 207 208 208 208 208 208 208 208 209 211 213 214 215 217 218 219 220

[217] 220 220 220 220 220 220 220 220 220 221 222 222 223 224 225 227 228 229 231 232 233 234 236 237

[241] 239 240 241 242 243 244 NA NA NA NA NA

$random

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] NA NA NA NA NA 4.277 -3.814 -3.341 -5.776 2.035 7.339 2.307

[13] -4.287 -4.218 -7.217 -5.136 -3.897 3.662 2.796 2.108 8.663 1.238 0.750 -4.935

[25] -1.902 -2.529 0.394 4.215 0.525 0.744 4.466 0.656 -1.296 -6.930 -9.655 0.853

[37] 2.384 3.828 5.779 5.366 -10.638 -4.058 -3.158 -0.451 2.979 1.608 -0.288 -2.988

[49] -0.348 1.127 1.760 2.471 -0.759 -1.639 -3.556 -0.931 -2.920 -6.018 2.489 1.483

[61] -0.100 3.490 5.678 4.023 0.746 -1.524 -1.957 -0.087 -0.533 -0.123 -0.915 -0.692

[73] 0.689 -1.286 0.754 0.208 0.532 1.004 -0.915 0.568 1.099 0.365 -0.591 -4.139

[85] -4.329 -0.831 0.746 0.601 0.806 0.892 1.274 -0.853 0.870 6.555 2.561 -1.428

[97] -3.188 -1.955 -2.097 -1.462 -1.157 0.569 1.381 -0.056 0.784 1.712 1.176 1.933

[109] -0.538 -1.670 -2.175 -2.758 1.231 0.803 1.406 2.670 2.440 -0.017 -1.696 -2.245

[121] -6.271 4.172 3.968 6.495 5.012 -0.389 1.957 1.997 0.545 -8.775 -3.872 0.356

[133] 2.300 2.526 -1.612 1.510 -0.389 -1.841 -0.998 -1.048 -1.642 -0.492 -3.643 -7.276

[145] -1.958 -2.054 -0.282 3.259 3.773 3.766 1.320 -0.050 -1.878 -2.117 2.592 0.367

[157] 2.122 1.191 0.346 -3.529 0.041 -0.435 -1.831 0.947 2.098 1.971 2.241 -2.225

[169] -1.520 0.024 -1.668 0.050 2.407 0.300 -1.015 -0.567 -3.397 1.074 2.254 1.118

[181] -1.274 -0.615 2.672 2.401 8.221 2.946 -0.222 -9.817 -5.710 -3.432 1.688 -0.393

[193] -1.152 5.162 -1.919 -5.387 -0.436 3.318 3.265 5.379 4.578 -5.569 -0.873 -3.570

[205] -1.686 0.395 0.234 -3.606 -1.601 0.372 -1.132 -1.118 0.829 5.633 4.557 -1.717

[217] -0.293 0.518 2.664 0.874 -2.318 -1.386 -2.099 0.589 0.134 -2.720 2.275 2.429

[229] -3.840 -3.058 1.634 0.368 -2.698 -2.033 0.021 3.621 2.864 1.151 -1.388 -2.227

[241] -2.409 0.662 -0.220 1.499 1.268 1.297 NA NA NA NA NA

$figure

[1] 0.232 0.083 -0.642 0.015 -0.472 0.855 0.129 -0.158 -0.155 0.113

$type

[1] "additive"

attr(,"class")

[1] "decomposed.ts"

> decompose(data, type = "multi")

$x

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] 207 202 204 210 208 204 194 192 187 191 194 186 177 177 172 175 174 181 180 179 185 177 175 168

[25] 170 169 169 171 165 164 166 161 157 151 147 157 156 156 158 158 142 148 148 151 153 154 152 150

[49] 153 155 156 157 153 154 153 158 156 155 165 166 167 171 174 174 171 170 169 171 170 171 170 171

[73] 172 171 173 174 174 175 173 175 176 176 175 172 173 179 181 182 184 186 188 187 188 195 191 189

[97] 187 188 189 190 191 194 195 196 197 200 200 201 199 199 199 199 203 204 205 207 207 205 204 205

[121] 201 211 209 212 208 203 203 201 197 186 189 191 190 189 183 187 183 180 179 178 177 178 175 173

[145] 180 183 185 190 193 195 194 194 193 194 198 198 199 199 199 196 200 200 198 202 203 204 204 200

[169] 201 203 202 203 205 204 203 206 203 207 209 209 207 208 210 209 213 208 204 193 197 199 203 201

[193] 200 209 203 202 206 210 210 213 212 203 206 204 206 209 209 206 209 213 213 214 217 224 223 219

[217] 220 221 223 221 218 219 218 221 220 219 224 225 219 221 227 227 224 227 230 236 236 235 234 235

[241] 236 241 240 243 244 247 249 243 243 249 256

$seasonal

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[49] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[97] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[145] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[193] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[241] 1 1 1 1 1 1 1 1 1 1 1

$trend

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] NA NA NA NA NA 199 198 196 193 189 186 183 182 181 180 179 178 177 177 176 176 175 175 173

[25] 172 170 169 167 165 163 161 160 159 158 157 155 154 153 152 152 153 152 152 151 151 151 152 153

[49] 154 154 154 154 155 156 157 158 159 161 163 165 166 168 169 170 171 171 171 171 171 171 171 172

[73] 172 172 173 173 174 174 174 174 175 175 176 177 178 179 180 181 183 185 187 187 188 189 189 189

[97] 190 191 191 191 192 193 195 196 197 198 198 199 200 200 201 202 203 203 204 204 205 205 206 207

[121] 207 206 206 205 204 202 201 199 197 194 192 190 188 186 185 184 183 182 180 179 179 179 179 181

[145] 182 184 185 187 189 191 193 194 195 196 196 197 197 198 198 199 200 200 200 201 201 202 202 202

[169] 203 203 203 203 203 204 205 205 206 206 207 207 208 208 208 206 205 205 204 203 203 202 202 201

[193] 202 204 205 206 207 207 207 207 208 208 208 208 208 208 208 209 211 213 214 215 217 218 219 220

[217] 220 220 220 220 220 220 220 220 220 221 222 222 223 224 225 227 228 229 231 232 233 234 236 237

[241] 239 240 241 242 243 244 NA NA NA NA NA

$random

Time Series:

Start = c(1, 1)

End = c(26, 1)

Frequency = 10

[1] NA NA NA NA NA 1.02 0.98 0.98 0.97 1.01 1.04 1.01 0.98 0.98 0.96 0.97 0.98 1.02 1.02

[20] 1.01 1.05 1.01 1.00 0.97 0.99 0.99 1.00 1.03 1.00 1.00 1.03 1.00 0.99 0.96 0.94 1.01 1.02 1.02

[39] 1.04 1.03 0.93 0.97 0.98 1.00 1.02 1.01 1.00 0.98 1.00 1.01 1.01 1.02 0.99 0.99 0.98 1.00 0.98

[58] 0.96 1.01 1.01 1.00 1.02 1.03 1.02 1.01 0.99 0.99 1.00 1.00 1.00 0.99 1.00 1.00 0.99 1.01 1.00

[77] 1.00 1.01 0.99 1.00 1.01 1.00 1.00 0.98 0.98 1.00 1.00 1.00 1.00 1.00 1.01 1.00 1.00 1.04 1.01

[96] 0.99 0.98 0.99 0.99 0.99 0.99 1.00 1.01 1.00 1.01 1.01 1.01 1.01 1.00 0.99 0.99 0.99 1.01 1.00

[115] 1.01 1.01 1.01 1.00 0.99 0.99 0.97 1.02 1.02 1.03 1.03 1.00 1.01 1.01 1.00 0.95 0.98 1.00 1.01

[134] 1.01 0.99 1.01 1.00 0.99 0.99 0.99 0.99 1.00 0.98 0.96 0.99 0.99 1.00 1.02 1.02 1.02 1.01 1.00

[153] 0.99 0.99 1.01 1.00 1.01 1.01 1.00 0.98 1.00 1.00 0.99 1.01 1.01 1.01 1.01 0.99 0.99 1.00 0.99

[172] 1.00 1.01 1.00 1.00 1.00 0.98 1.01 1.01 1.00 0.99 1.00 1.01 1.01 1.04 1.01 1.00 0.95 0.97 0.98

[191] 1.01 1.00 0.99 1.03 0.99 0.97 1.00 1.02 1.02 1.03 1.02 0.97 1.00 0.98 0.99 1.00 1.00 0.98 0.99

[210] 1.00 0.99 0.99 1.00 1.03 1.02 0.99 1.00 1.00 1.01 1.00 0.99 0.99 0.99 1.00 1.00 0.99 1.01 1.01

[229] 0.98 0.99 1.01 1.00 0.99 0.99 1.00 1.01 1.01 1.00 0.99 0.99 0.99 1.00 1.00 1.01 1.01 1.00 NA

[248] NA NA NA NA

$figure

[1] 1 1 1 1 1 1 1 1 1 1

$type

[1] "multiplicative"

attr(,"class")

[1] "decomposed.ts"

> par(mfrow=c(1,2))

> plot(decompose(data, type = "multi"))

> # creating seasonal forecast

> library(forecast)

Registered S3 method overwritten by 'xts':

method from

as.zoo.xts zoo

Registered S3 method overwritten by 'quantmod':

method from

as.zoo.data.frame zoo

Registered S3 methods overwritten by 'forecast':

method from

fitted.fracdiff fracdiff

residuals.fracdiff fracdiff

Warning message:

package ‘forecast’ was built under R version 3.6.1

> par(mfrow=c(1,1))

> seasonplot(data)

> # lags

> lag(data,10)

Error: `x` must be a vector, not a ts object, do you want `stats::lag()`?

Call `rlang::last\_error()` to see a backtrace

> lag.plot(data)

> # Partial auto correlation

> pac <- pacf(data)

> pac$acf

, , 1

[,1]

[1,] 0.9734

[2,] 0.0659

[3,] 0.0651

[4,] -0.0056

[5,] -0.0589

[6,] -0.0052

[7,] 0.0193

[8,] -0.0528

[9,] 0.0359

[10,] -0.0778

[11,] 0.0171

[12,] 0.0222

[13,] 0.0103

[14,] -0.0052

[15,] -0.0401

[16,] -0.0344

[17,] 0.0276

[18,] -0.0329

[19,] 0.0333

[20,] -0.0409

[21,] -0.0402

[22,] 0.0425

[23,] 0.0404

> # Auto correlation

> ac <- acf(data)

> ac$acf

, , 1

[,1]

[1,] 1.00

[2,] 0.97

[3,] 0.95

[4,] 0.93

[5,] 0.91

[6,] 0.89

[7,] 0.87

[8,] 0.85

[9,] 0.83

[10,] 0.81

[11,] 0.79

[12,] 0.77

[13,] 0.75

[14,] 0.73

[15,] 0.71

[16,] 0.69

[17,] 0.67

[18,] 0.65

[19,] 0.63

[20,] 0.61

[21,] 0.59

[22,] 0.57

[23,] 0.55

[24,] 0.54

> #the decay of ACF chart is very slow, which means that the population is not stationary

> # we now intend to regress on the difference of logs rather than log directly.

> #Let's see how ACF and PACF curve come out after regressing on the difference

> # looking at ACF and PACF graph it is clear that the time series is not stationary

> #------------------------------------------

> pacf(diff(log(data)))

> acf(diff(log(data)))

> tbl <- stl(data, 'periodic')

> stab <- seasadj(tbl)

> seasonplot(stab, 12)

> # unit root for stationarity

> # The Augmented Dicky Fuller Test for

> library(tseries)

‘tseries’ version: 0.10-47

‘tseries’ is a package for time series analysis and computational finance.

See ‘library(help="tseries")’ for details.

Warning message:

package ‘tseries’ was built under R version 3.6.1

> adf.test(data)

Augmented Dickey-Fuller Test

data: data

Dickey-Fuller = -3, Lag order = 6, p-value = 0.3

alternative hypothesis: stationary

> acf(log(data))

> pacf(log(data))

> acf(diff(log(data)))

> pacf(diff(log(data)))

> #main part start

> data = ts(na.omit(AAPL1$Close ), frequency=10)

Error in na.omit(AAPL1$Close) : object 'AAPL1' not found

> decomp = stl(data, s.window="periodic") #decompose

> #main part start

> data = ts(na.omit(AAPL$Close ), frequency=10)

> decomp = stl(data, s.window="periodic") #decompose

> deseasonal\_cnt <- seasadj(decomp)

> plot(decomp)

> adf.test(data, alternative = "stationary")

Augmented Dickey-Fuller Test

data: data

Dickey-Fuller = -3, Lag order = 6, p-value = 0.3

alternative hypothesis: stationary

> #since it's p value is 0.14 which is greater than 0.05

> #we have to do further processing by changing the value out of(p,d,q) of d.

> Acf(data, main='')

> Pacf(data, main='')

> #thus we change d again n again so that we get desired p value

> data = diff(deseasonal\_cnt, differences = 1)

> plot(data)

> adf.test(data, alternative = "stationary")

Augmented Dickey-Fuller Test

data: data

Dickey-Fuller = -5, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

Warning message:

In adf.test(data, alternative = "stationary") :

p-value smaller than printed p-value

> data = diff(deseasonal\_cnt, differences = 2)

> plot(data)

> adf.test(data, alternative = "stationary")

Augmented Dickey-Fuller Test

data: data

Dickey-Fuller = -11, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

Warning message:

In adf.test(data, alternative = "stationary") :

p-value smaller than printed p-value

> Acf(data, main='ACF for Differenced Series')

> Pacf(data, main='PACF for Differenced Series')

> # Automatic ARIMA Model

> model2 <- auto.arima(deseasonal\_cnt,seasonal = FALSE)

> model2

Series: deseasonal\_cnt

ARIMA(1,2,1)

Coefficients:

ar1 ma1

-0.032 -0.970

s.e. 0.066 0.021

sigma^2 estimated as 12.7: log likelihood=-670

AIC=1347 AICc=1347 BIC=1357

> tsdisplay(residuals(model2), lag.max=15, main='Seasonal Model Residuals')

> #tsdisply helps in display overall of various things

> plot(forecast(model2, h=15))

> accuracy(model2)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.34 3.5 2.6 0.18 1.4 0.28 -0.0066

> # more running model on deseasonal\_cnt(deseasonal data)

> model3 <- arima(deseasonal\_cnt, order=c(1,2,7))

> model3

Call:

arima(x = deseasonal\_cnt, order = c(1, 2, 7))

Coefficients:

ar1 ma1 ma2 ma3 ma4 ma5 ma6 ma7

-0.74 -0.24 -0.83 0.023 0.088 0.011 -0.058 0.090

s.e. 0.14 0.15 0.14 0.084 0.095 0.091 0.068 0.076

sigma^2 estimated as 12.3: log likelihood = -667, aic = 1351

> tsdisplay(residuals(model3), lag.max=15, main='Seasonal Model Residuals')

> plot(forecast(model3, h=15))

> accuracy(model3)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.25 3.5 2.5 0.13 1.3 0.98 -0.0083

> # taking random order

> model4 <- arima(deseasonal\_cnt, order = c(4,2,7))

Warning message:

In arima(deseasonal\_cnt, order = c(4, 2, 7)) :

possible convergence problem: optim gave code = 1

> model4

Call:

arima(x = deseasonal\_cnt, order = c(4, 2, 7))

Coefficients:

ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4 ma5 ma6 ma7

-0.7 -0.11 -0.43 -0.8 -0.27 -0.69 0.32 0.503 -0.87 -0.046 0.222

s.e. NaN 0.09 NaN NaN NaN 0.09 NaN 0.078 NaN 0.071 0.065

sigma^2 estimated as 11.6: log likelihood = -662, aic = 1349

Warning message:

In sqrt(diag(x$var.coef)) : NaNs produced

> tsdisplay(residuals(model4), lag.max=15, main='Seasonal Model Residuals')

> accuracy(model4)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.23 3.4 2.5 0.12 1.3 0.97 0.0095

> plot(forecast(model4, h=15))

> # taking random order

> model5 <- arima(deseasonal\_cnt, order = c(4,2,4))

> model5

Call:

arima(x = deseasonal\_cnt, order = c(4, 2, 4))

Coefficients:

ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4

0.531 0.30 -0.78 -0.099 -1.571 0.23 1.23 -0.857

s.e. 0.098 0.13 0.10 0.070 0.078 0.18 0.18 0.077

sigma^2 estimated as 11.7: log likelihood = -664, aic = 1345

> tsdisplay(residuals(model5), lag.max=15, main='Seasonal Model Residuals')

> accuracy(model5)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.29 3.4 2.5 0.15 1.3 0.97 -0.0051

> plot(forecast(model5, h=15))

> # taking random order

> model6 <- arima(deseasonal\_cnt, order = c(3,2,5))

> model6

Call:

arima(x = deseasonal\_cnt, order = c(3, 2, 5))

Coefficients:

ar1 ar2 ar3 ma1 ma2 ma3 ma4 ma5

0.65 0.23 -0.81 -1.70 0.43 1.20 -1.01 0.110

s.e. 0.11 0.16 0.11 0.13 0.27 0.22 0.12 0.075

sigma^2 estimated as 11.7: log likelihood = -664, aic = 1345

> tsdisplay(residuals(model6), lag.max=15, main='Seasonal Model Residuals')

> accuracy(model6)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.28 3.4 2.5 0.15 1.3 0.97 -0.00046

> plot(forecast(model6, h=15))

> # taking random order

> model7 <- arima(deseasonal\_cnt, order = c(0,2,1))

> model7

Call:

arima(x = deseasonal\_cnt, order = c(0, 2, 1))

Coefficients:

ma1

-0.97

s.e. 0.02

sigma^2 estimated as 12.6: log likelihood = -671, aic = 1345

> tsdisplay(residuals(model7), lag.max=15, main='Seasonal Model Residuals')

> accuracy(model7)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.35 3.5 2.6 0.18 1.4 0.99 -0.034

> plot(forecast(model7, h=15))

> # taking random order

> model8 <- arima(deseasonal\_cnt, order = c(1,2,0))

> model8

Call:

arima(x = deseasonal\_cnt, order = c(1, 2, 0))

Coefficients:

ar1

-0.471

s.e. 0.056

sigma^2 estimated as 19.6: log likelihood = -724, aic = 1452

> tsdisplay(residuals(model8), lag.max=15, main='Seasonal Model Residuals')

> accuracy(model8)

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.064 4.4 3.3 0.026 1.7 1.3 -0.16

> plot(forecast(model8, h=15))

> # Holt Winters Exponential Smoothing Model

> model9 <- HoltWinters(deseasonal\_cnt, gamma = F)

> summary(model9)

Length Class Mode

fitted 747 mts numeric

x 251 ts numeric

alpha 1 -none- numeric

beta 1 -none- numeric

gamma 1 -none- logical

coefficients 2 -none- numeric

seasonal 1 -none- character

SSE 1 -none- numeric

call 3 -none- call

> tsdisplay(residuals(model9), lag.max=15, main='Seasonal Model Residuals')

> plot(forecast(model9, h=15))

> accuracy(forecast(model9, h=15))

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.37 3.6 2.6 0.19 1.4 0.29 -0.0054

> # ETS

> model10 <- ets(deseasonal\_cnt)

> summary(model10)

ETS(A,N,N)

Call:

ets(y = deseasonal\_cnt)

Smoothing parameters:

alpha = 0.9933

Initial states:

l = 207.1544

sigma: 3.6

AIC AICc BIC

2028 2028 2039

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.19 3.5 2.6 0.065 1.4 0.28 -0.0025

> tsdisplay(residuals(model10), lag.max=15, main='Seasonal Model Residuals')

> plot(forecast(model10, h=15))

> accuracy(forecast(model10, h=15))

ME RMSE MAE MPE MAPE MASE ACF1

Training set 0.19 3.5 2.6 0.065 1.4 0.28 -0.0025

> # Making predictions for next 15 days

> predicted <- forecast(model4, 15)

> predicted

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

26.10 255 251 260 249 262

26.20 256 250 263 247 266

26.30 257 250 265 246 269

26.40 258 250 267 245 271

26.50 260 250 269 245 274

26.60 260 250 270 244 276

26.70 263 251 274 245 280

26.80 264 251 276 245 282

26.90 265 252 278 245 285

27.00 266 252 281 244 289

27.10 267 252 282 243 291

27.20 269 253 285 244 294

27.30 270 252 287 243 296

27.40 272 253 290 243 300

27.50 273 253 293 243 303

> # comparing data with forecast

> predicted$residuals[60:76]

[1] 2.883 0.468 5.857 3.457 -2.040 -1.304 -4.035 -0.074 0.786 -2.574 0.593 -3.413 1.846 0.703

[15] -2.443 2.905 -2.227

>

Graphs

