Model Building 4

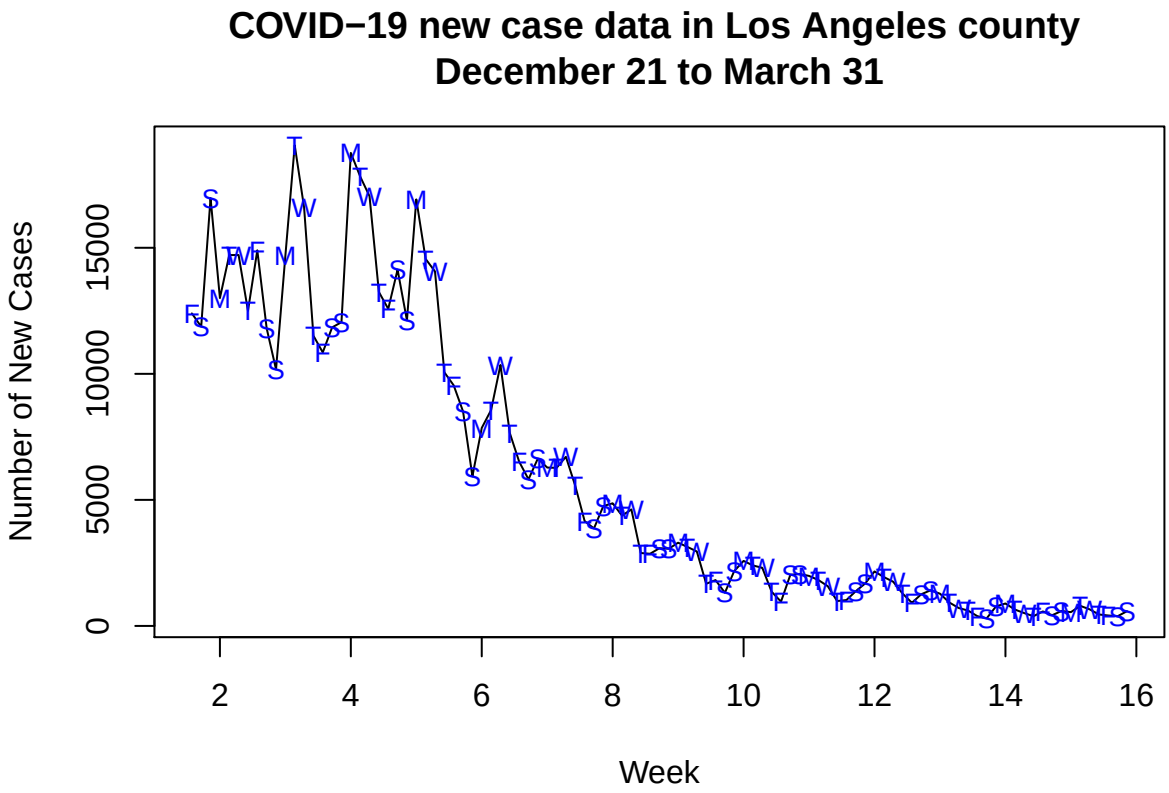
Lian Morales

4/26/2021

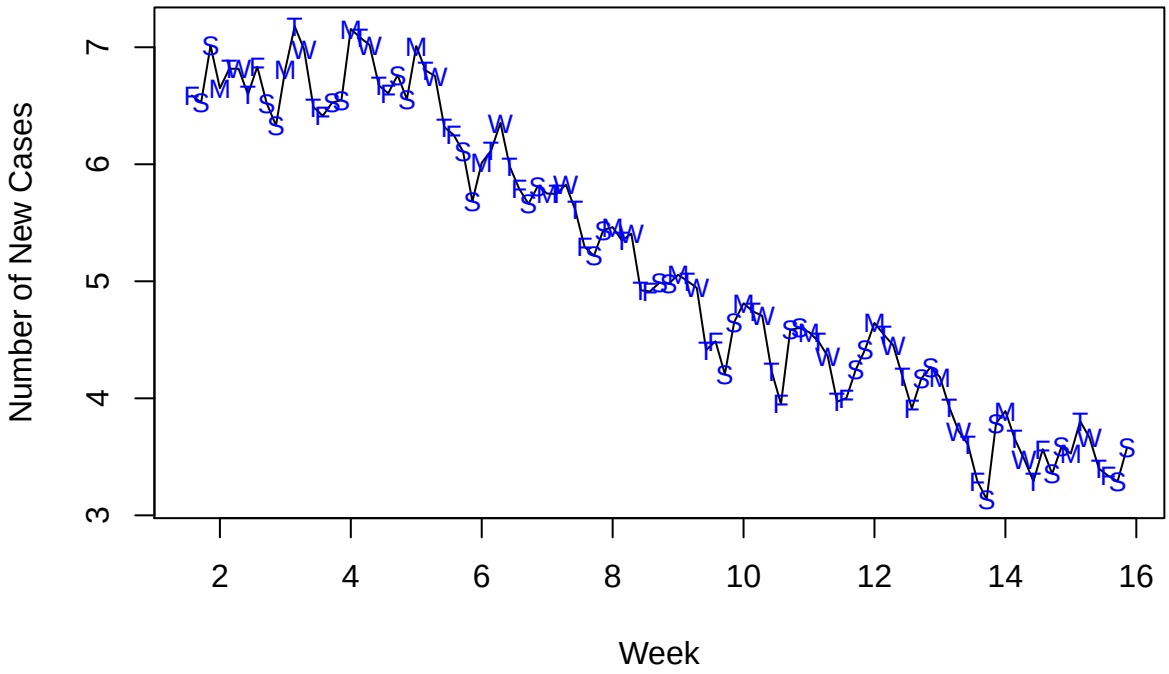
```
# pulling the data from the Los Angeles County GitHub
casedata <- read.csv(text = getURL("https://raw.githubusercontent.com/datadesk/california-coronavirus-da
  filter(county == "Los Angeles") %>%
  mutate(date = date(date), month = month(date)) %>%
  map_df(rev) %>%
  filter(!is.na(new_confirmed_cases) & between(date, date("2020-04-01"),date("2021-03-31")))

# creating the time series
case.ts <- ts(casedata$new_confirmed_cases, start = 1,  frequency = 1)

# averaging dec 25th and 26th
case.ts[269] <- 14711
case.ts[270] <- 14712

# dec 21 - mar 31
case.ts.2.2 <- ts(case.ts[265:365], start = 1, frequency = 1)
```

December 21 - March 31

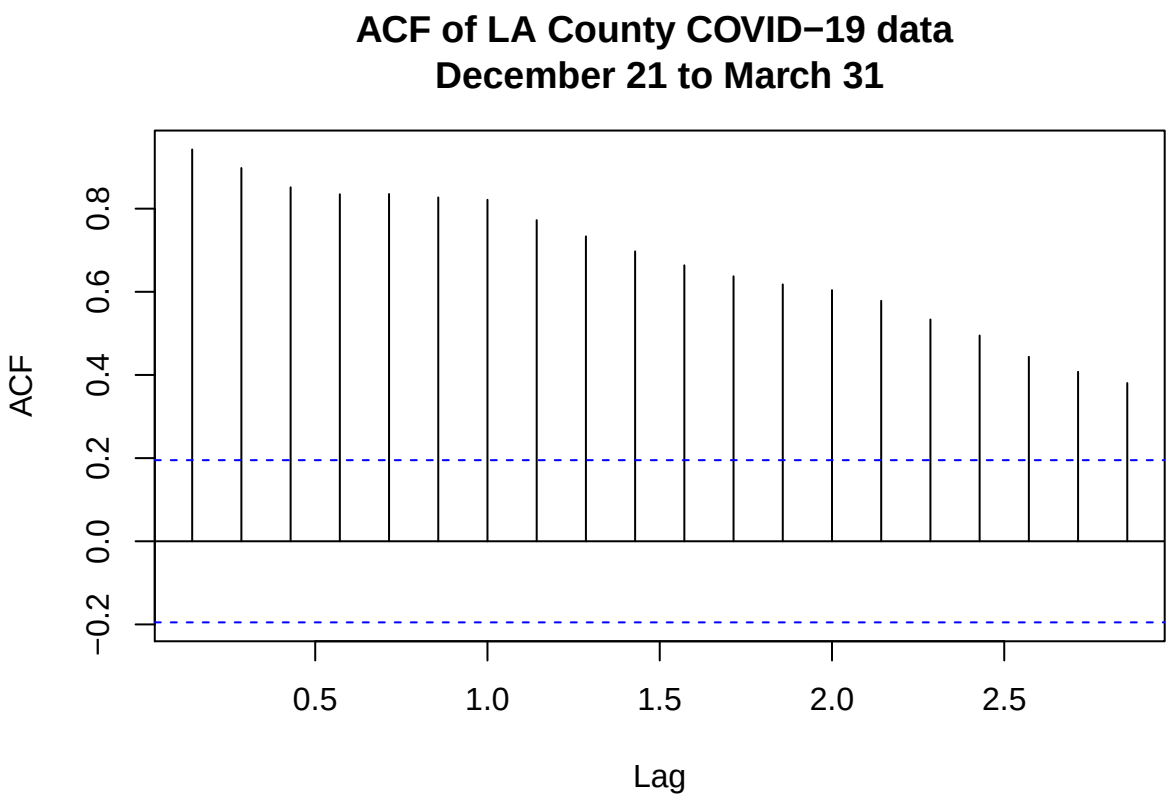
```
set.seed(13)
# transform data October 1 - December 20 to lambda + add seasonality
trans.seasonal.ts.2.2  <- ts(case.ts.2.2^0.2, frequency = 7,start = c(1,5))
seasonal.ts.2.2  <- ts(case.ts.2.2, frequency = 7,start = c(1,5))
trans.ts.2.2 <- ts(case.ts.2.2^0.2)
# plot ts of seasonal data
week. <- season(seasonal.ts.2.2)
plot(seasonal.ts.2.2, ylab = "Number of New Cases", xlab = "Week", main = "COVID-19 new case data in Los
points(seasonal.ts.2.2, pch = as.vector(week.), col = "blue", cex = 0.8)
```



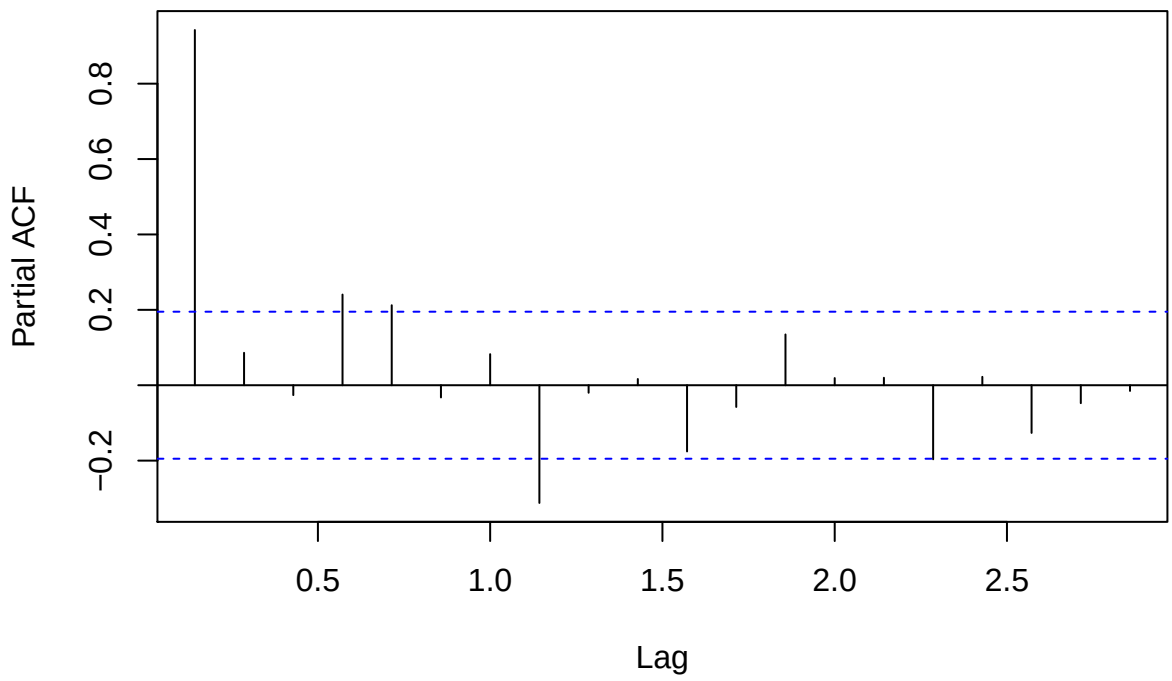

```
week. <- season(trans.seasonal.ts.2.2)
plot(trans.seasonal.ts.2.2, ylab = "Number of New Cases", xlab = "Week", main = "Lamda-transformed COVI
points(trans.seasonal.ts.2.2, pch = as.vector(week.), col = "blue", cex = 0.8)
```



```r
# acf of transformed data
acf(seasonal.ts.2.2, main = "ACF of LA County COVID-19 data \nDecember 21 to March 31")
```




```r
# pacf of transformed data
pacf(seasonal.ts.2.2, main = "PACF of LA County COVID-19 data \nDecember 21 to March 31")
```



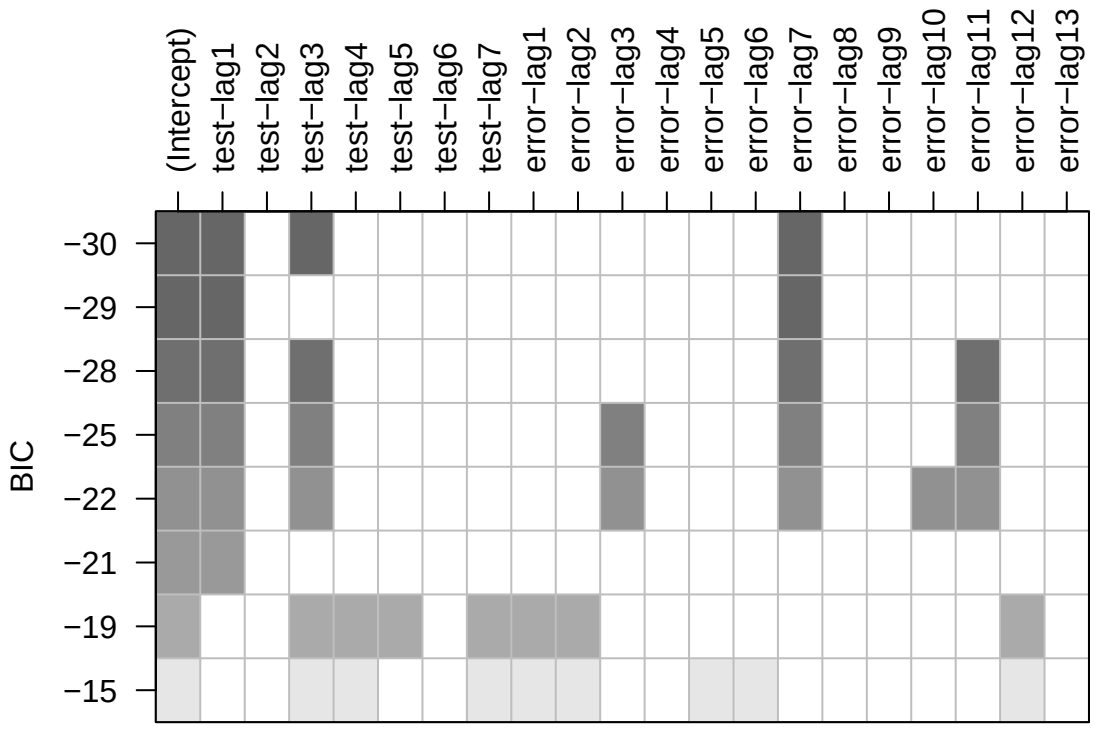

```
# best subsets ARMA approach
eacf(trans.seasonal.ts.2.2)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 o o x x o o x o o x x  o  o  x
## 2 x o o x o o x o o o x  o  o  o
## 3 x o o x o o x o o o o  o  o  o
## 4 x x o o o o x o o o o  o  o  o
## 5 x x o x o o o o o o o  o  o  o
## 6 x o x x o o x o o o o  o  o  o
## 7 o o x x o o o o o o o  o  o  o
```

```
eacf(diff(trans.seasonal.ts.2.2, lag = 7))
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x o o o o o o  o  o  o
## 1 o x o o x o x o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 o x o o o o x o o o o  o  o  o
## 5 o x o o o o x o o o o  o  o  o
## 6 x x o o o x x o o o o  o  o  o
## 7 x x o o o o x o o o o  o  o  o
```

```
plot(armasubsets(y=diff(trans.ts.2.2,lag=7), nar= 7, nma = 13, y.name = "test", ar.method = 'ols'))
```




```
auto.arima(trans.seasonal.ts.2.2)
```

```
## Series: trans.seasonal.ts.2.2
## ARIMA(3,1,2) with drift
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2    drift
##       0.9620  -0.5989  -0.2510  -1.3706  0.8304  -0.0322
## s.e.  0.1253   0.1356   0.1178   0.0989  0.0696   0.0104
##
## sigma^2 estimated as 0.04263:  log likelihood=17.74
## AIC=-21.49   AICc=-20.27   BIC=-3.25
```

```
auto.arima(trans.ts.2.2)
```

```
## Series: trans.ts.2.2
## ARIMA(0,1,3) with drift
##
## Coefficients:
##          ma1      ma2      ma3    drift
##       -0.2309  -0.1892  -0.2612  -0.0344
## s.e.   0.0952   0.1070   0.1048   0.0078
##
## sigma^2 estimated as 0.05548:  log likelihood=4.46
## AIC=1.08   AICc=1.72   BIC=14.1
```

ARIMA(3,0,0)x(0,1,1) # dif of arma subset ARIMA(3,1,2) # arima sim ARIMA(0,1,3) # arima sim

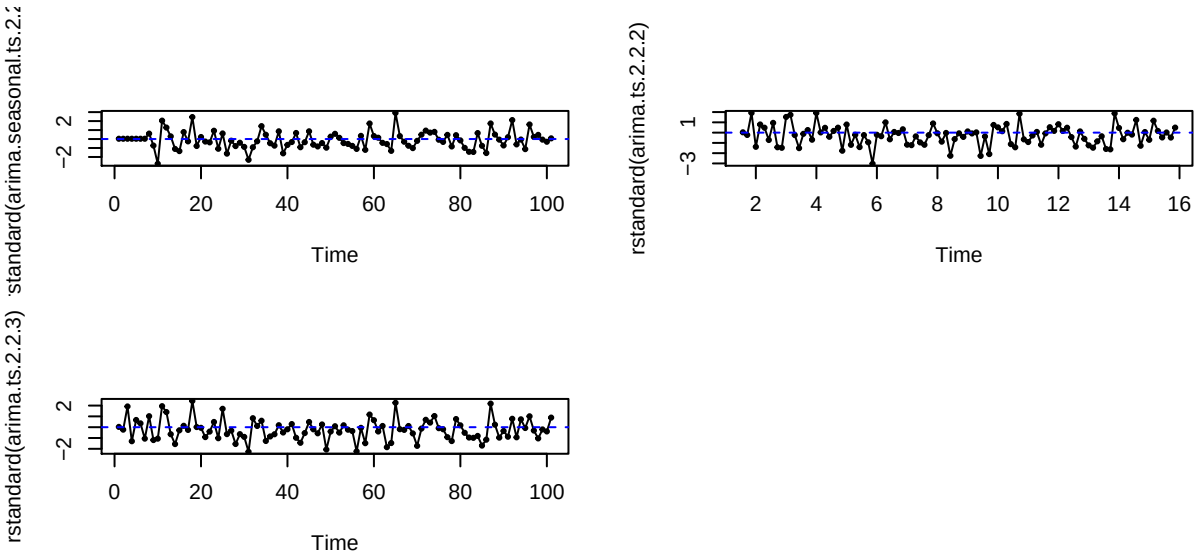
Completing analysis with suggested models

```
# create aimra models for seasonal and transformed data
arima.seasonal.ts.2.2.1 <- arima(trans.ts.2.2, order = c(3,0,0), seasonal = list(order = c(0,1,1), peri
arima.ts.2.2.2 <- arima(trans.seasonal.ts.2.2, order = c(3,1,2))
```

```
## Warning in log(s2): NaNs produced
```

```
arima.ts.2.2.3 <- arima(trans.ts.2.2, order = c(0,1,3))
```

```
par(mfrow = c(3,2))
# plot residuals
plot(rstandard(arima.seasonal.ts.2.2.1),type = "o", pch=20)
abline(h=0,lty=2,col="blue")
plot(rstandard(arima.ts.2.2.2),type = "o", pch=20)
abline(h=0,lty=2,col="blue")
plot(rstandard(arima.ts.2.2.3),type = "o", pch=20)
abline(h=0,lty=2,col="blue")
```




```
# AIC
arima.seasonal.ts.2.2.1$aic
```

```
## [1] -11.90926
```

```
arima.ts.2.2.2$aic
```

```
## [1] -17.27363
```

```
arima.ts.2.2.3$aic
```

```
## [1] 6.177964
```

```r
# residual tests

# check for correlation of error terms
Box.test(rstandard(arima.seasonal.ts.2.2.1), type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rstandard(arima.seasonal.ts.2.2.1)
## X-squared = 0.27936, df = 1, p-value = 0.5971
```

```r
Box.test(rstandard(arima.ts.2.2.2), type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rstandard(arima.ts.2.2.2)
## X-squared = 0.45794, df = 1, p-value = 0.4986
```

```r
Box.test(rstandard(arima.ts.2.2.3), type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rstandard(arima.ts.2.2.3)
## X-squared = 0.35416, df = 1, p-value = 0.5518
```

```r
# check for normality of error terms
shapiro.test(rstandard(arima.seasonal.ts.2.2.1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(arima.seasonal.ts.2.2.1)
## W = 0.97812, p-value = 0.09159
```

```r
shapiro.test(rstandard(arima.ts.2.2.2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(arima.ts.2.2.2)
## W = 0.9867, p-value = 0.4103
```

```r
shapiro.test(rstandard(arima.ts.2.2.3))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(arima.ts.2.2.3)
## W = 0.97873, p-value = 0.1024
```

```r
# check for independence of error terms
runs(rstandard(arima.seasonal.ts.2.2.1))
```

```
## $pvalue
## [1] 0.274
##
## $observed.runs
## [1] 45
##
## $expected.runs
## [1] 50.90099
##
## $n1
## [1] 56
##
## $n2
## [1] 45
##
## $k
## [1] 0
```

```r
runs(rstandard(arima.ts.2.2.2))
```

```
## $pvalue
## [1] 0.81
##
## $observed.runs
## [1] 51
##
## $expected.runs
## [1] 49.31683
##
## $n1
## [1] 61
##
## $n2
## [1] 40
##
## $k
## [1] 0
```

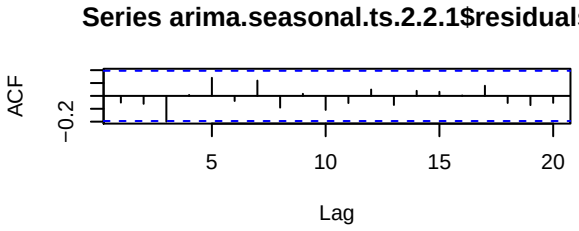
```r
runs(rstandard(arima.ts.2.2.3))
```

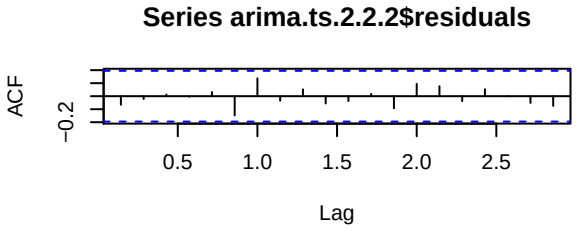
```
## $pvalue
## [1] 0.325
##
## $observed.runs
## [1] 53
##
## $expected.runs
## [1] 47.89109
##
```

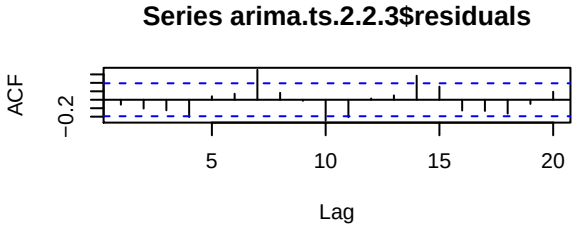
```
## $n1
## [1] 64
##
## $n2
## [1] 37
##
## $k
## [1] 0
```

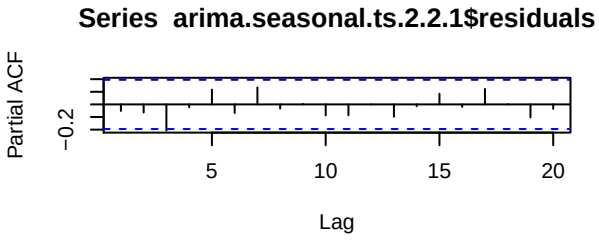
Visualize residuals

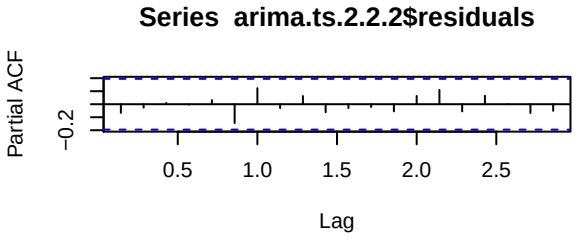
```
layout(matrix(c(1,2,3,4,5,0), nrow = 3, ncol = 2, byrow = TRUE))

acf(arima.seasonal.ts.2.2.1$residuals)
acf(arima.ts.2.2.2$residuals)
acf(arima.ts.2.2.3$residuals)


# pacf of seasonal and transformed residuals
pacf(arima.seasonal.ts.2.2.1$residuals)
pacf(arima.ts.2.2.2$residuals)
```





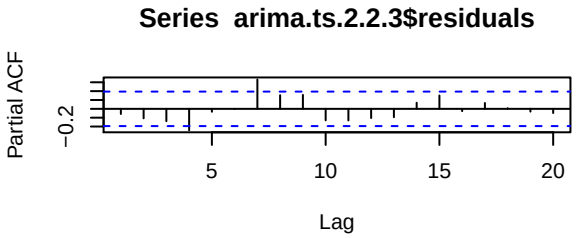


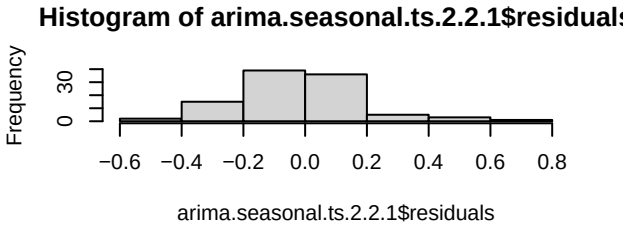


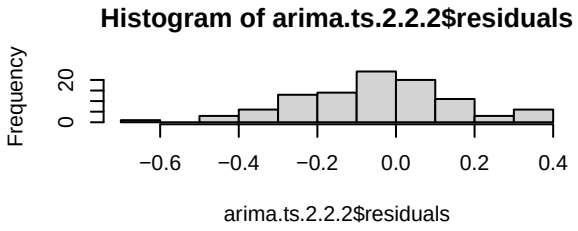


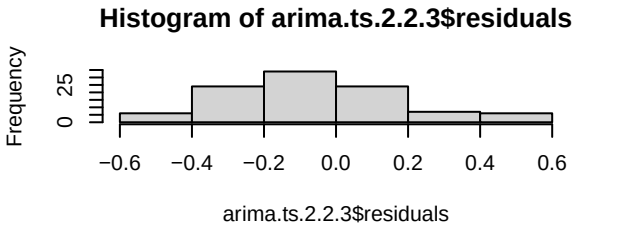

```
pacf(arima.ts.2.2.3$residuals)


# Histogram of residuals
```

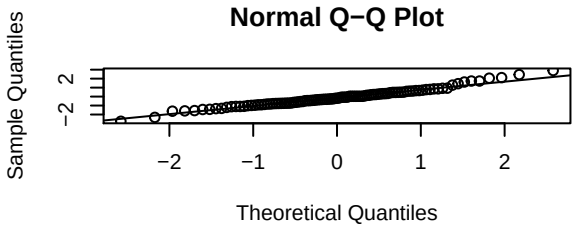
```
hist(arima.seasonal.ts.2.2.1$residuals)
hist(arima.ts.2.2.2$residuals)
hist(arima.ts.2.2.3$residuals)

# qqnorm plots of residuals
qqnorm(rstandard(arima.seasonal.ts.2.2.1))
qqline(rstandard(arima.seasonal.ts.2.2.1))
```



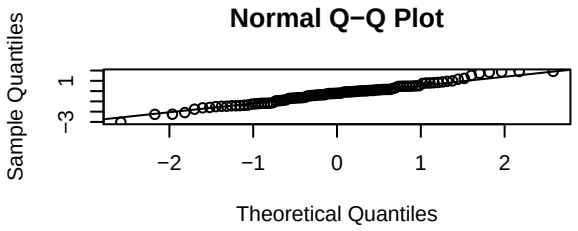


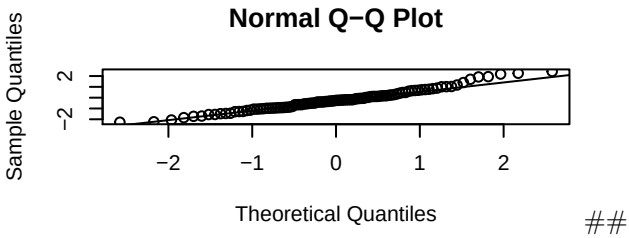







```
qqnorm(rstandard(arima.ts.2.2.2))
qqline(rstandard(arima.ts.2.2.2))

qqnorm(rstandard(arima.ts.2.2.3))
qqline(rstandard(arima.ts.2.2.3))
```



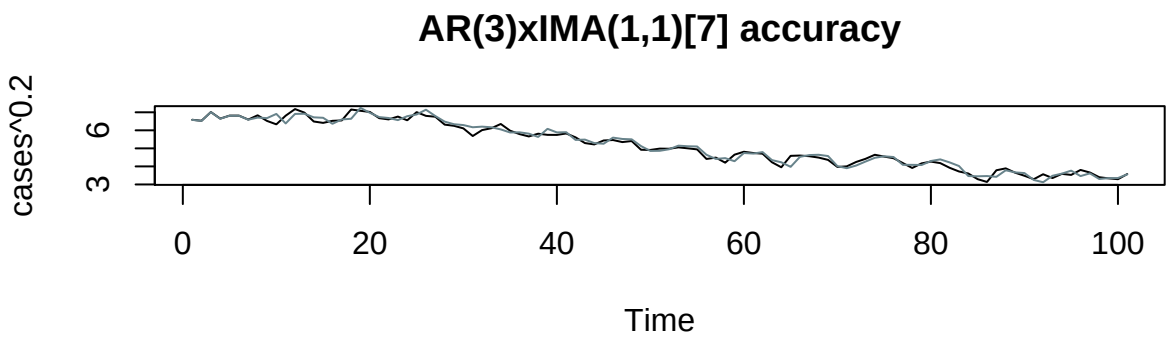


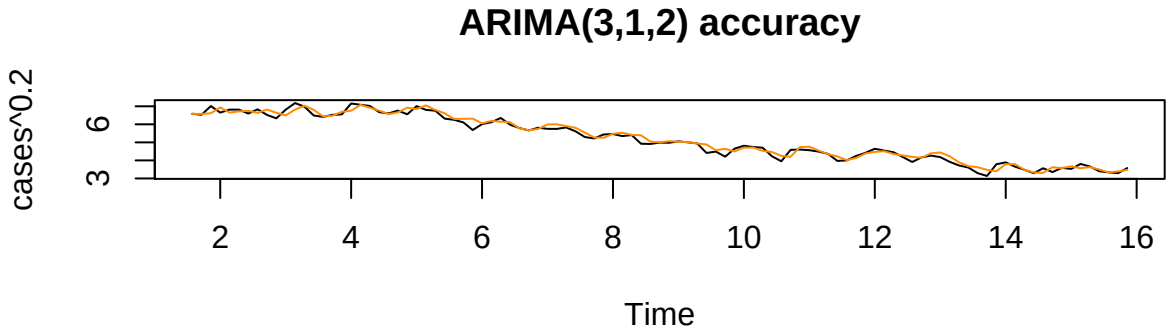
##

Forecasting

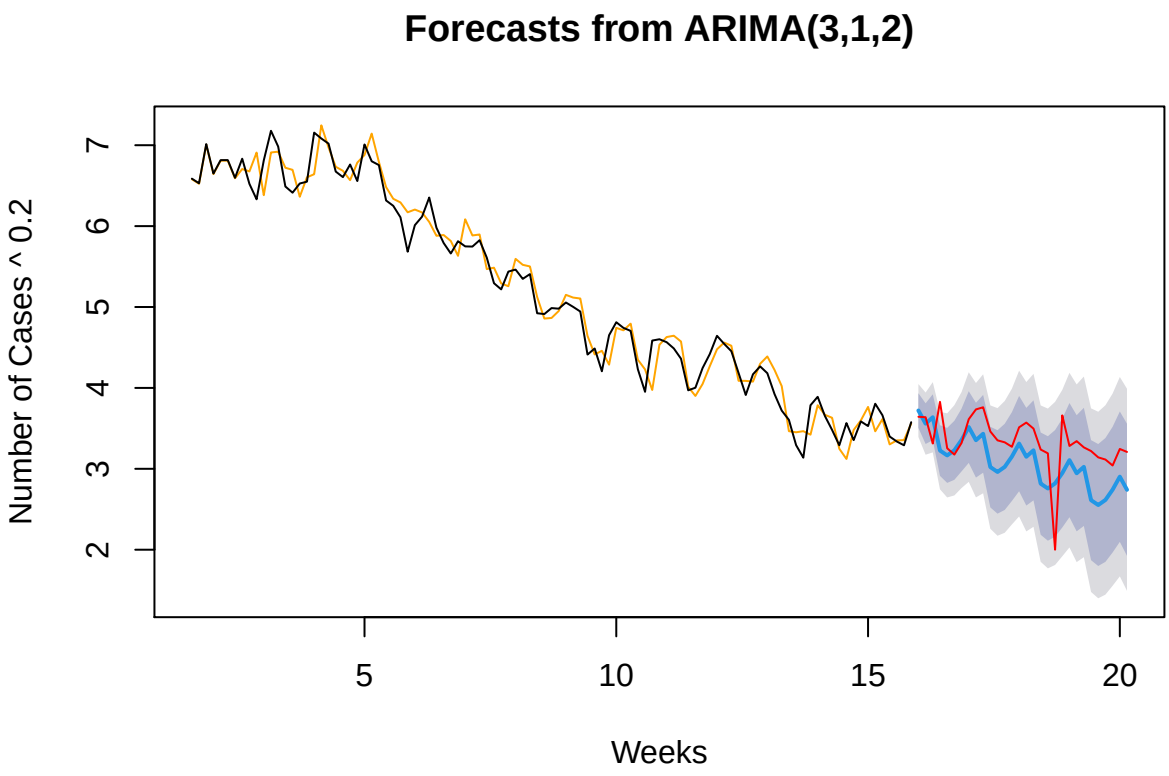
```
fit.1 <- Arima(trans.ts.2.2, order = c(3,0,0), seasonal = list(order = c(0,1,1), period = 7))
fit.1.ts <- ts(fit.1$fitted, frequency = 7, start = c(1,5))
fit.2 <- Arima(trans.seasonal.ts.2.2, order = c(3,1,2))
```

```
par(mfrow = c(2,1))
plot(trans.ts.2.2, ylab = "cases^0.2", main = "AR(3)xIMA(1,1)[7] accuracy")
lines(fit.1$fitted, col = "lightblue4")
plot(trans.seasonal.ts.2.2, ylab = "cases^0.2", main = "ARIMA(3,1,2) accuracy")
lines(fit.2$fitted, col = "darkorange")
```






```
testdata <- read.csv(text = getURL("https://raw.githubusercontent.com/datadesk/california-coronavirus-da
  filter(county == "Los Angeles") %>%
  mutate(date = date(date), month = month(date)) %>%
  map_df(rev) %>%
  filter(!is.na(new_confirmed_cases) & between(date, date("2021-04-01"),date("2021-04-30")))

testdata.ts <- ts((testdata$new_confirmed_cases)^0.2, frequency = 7, start = c(16,1))
par(mfrow = c(1,1))
plot(forecast(fit.1.ts,h=30), col = "orange", main = "Forecasts from ARIMA(3,1,2)",ylab="Number of Cases
lines(trans.seasonal.ts.2.2)
lines(testdata.ts, col = "red")
```


Forecasts from ARIMA(3,1,2)
Number of Cases ^ 0.2
2
3
4
5
6
7
5
10
15
20
Weeks