# Modeling Japan's Activity Rate in the 21st Century

Time Series Final Project for PSTAT 174

Justin Liu

## Contents

## Abstract

In this paper, we investigate the monthly activity rates of Japanese people ages 25-54 for the years 2000-2021. The activity rate is the percentage of the civilian non-institutional population that is either employed or actively looking for a job.

We seek to identify an appropriate time series model for this data using the Box-Jenkins methodology. We first try a Box-Cox transformation on the original time series data (from January 2000 to December 2020) to stabilize the variance, though the transformation does not significantly change its overall shape. Since the data exhibits a positive linear trend and seasonality, we difference once at both lags 1 and 12, respectively. We then identify several potential models using the ACF and PACF plots of the differenced series and choose a SARIMA$(0,1,2)\times(0,1,7)_{12}$ model with 4 parameters for the original data based on the lowest AICC. After running some diagnostic tests to ensure that the residuals of this model are Gaussian white noise, we use our model to forecast the activity rates for the months between January 2021 and December 2021. Although one prediction fell outside of the prediction interval, the model does an adequate job at forecasting.


## Introduction

The activity rate (sometimes called the labor force participation rate) is a metric that is often used to analyze employment data. It accounts for people who are still searching for jobs and omits certain populations (e.g. members of the military, people in institutions such as prisons and mental health facilities) that may otherwise make the unemployment rate unreliable (Investopedia). To calculate the activity rate $X_t$ (in %) for a given population, we use the formula

$$X_t = \frac{\text{active population at time } t}{\text{civilian non-institutional population at time } t} \times 100$$

where $t$ denotes the time when the statistic was observed. The active population is the number of people in the civilian non-institutional population who are either currently employed (active) or unemployed (potentially active and looking for a job).

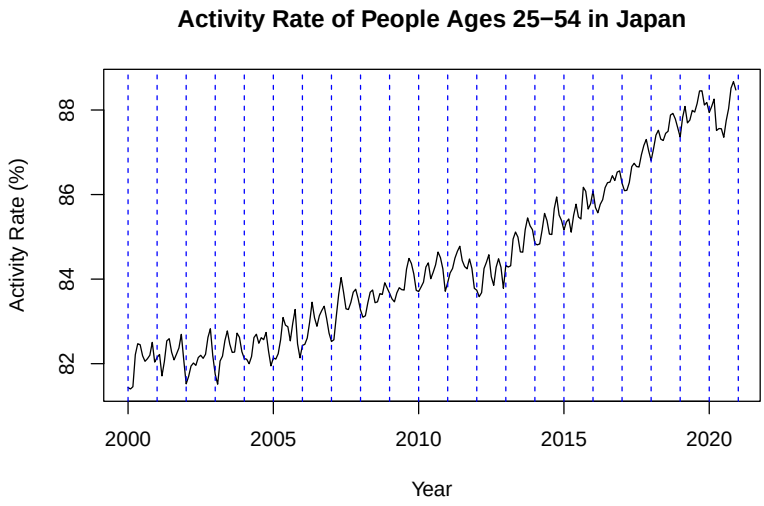
In our analysis, we focus on the monthly activity rates of people ages 25-54 in Japan in the 21st century, so $t$ is a month-year combination (e.g. March 2015). This population includes those who most likely have already graduated from college or are close to retiring. An analysis of this data could be beneficial to economists and sociologists who are interested in studying employment patterns in Japan.

We first use the data from January 2000 to December 2020 to identify and train our model. To this end, we follow the Box-Jenkins methodology, which involves finding a proper transformation of our time series data, differencing it to remove trend and seasonality, and fitting candidate models to find the one that best aligns with our data. After making sure that our model passes several diagnostic tests, we then make predictions for January 2021 to December 2021 and compare them with the actual activity rates to evaluate the performance of our model. Through this process, we choose SARIMA$(0,1,2)\times(0,1,7)_{12}$ to model our original data. Using this model, all but one of the predictions (out of 12) fall within the prediction interval, though the predictions on the test data do not strongly overlap with the actual data.

The source of the data comes from the Federal Reserve Bank of St. Louis (FRED). We utilize the language R to conduct our analysis.

## Exploratory Data Analysis

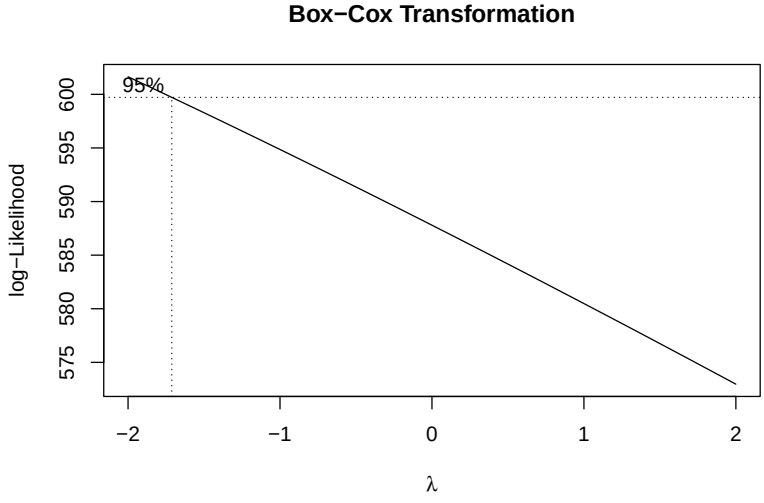
Before we start identifying and training our model, we need to first look at the data we are working with. We plot our training data, which consists of the monthly activity rates of Japanese people ages 25-54 between January 2000 to December 2020.




From this plot, we notice some interesting patterns. The activity rate appears to increase with each year, suggesting that the active population in Japan is getting closer to the civilian non-institutional population. Within each year, there is generally a peak during the spring, a small dip during the summer, and then another peak during the fall. The lowest activity rate in any given year tends to be either at the beginning or end of the year. However, the data for the year 2020 doesn't conform to these patterns as there is a noticeable dip (around a 1% decrease) that starts around April. This corresponds to the time period when Japan declared a state of emergency on April 7, 2020, in response to COVID-19 (Prime Minister's Office of Japan). We will keep this in mind when making our final conclusions.

## Transformation

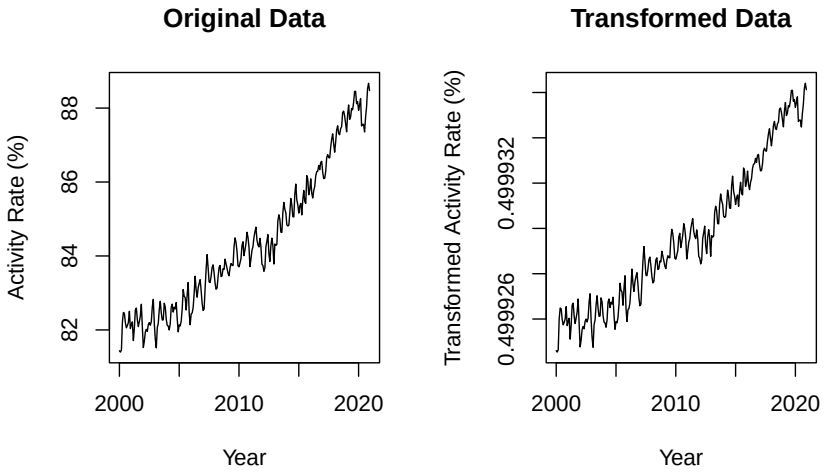
In order to apply the Box-Jenkins methodology, we need to ensure that the variance of the time series is stabilized. We can achieve this by performing a Box-Cox transformation, which essentially finds an appropriate value of $\lambda$ to transform the response.



Although it is not clear in the graph above, the chosen value is $\lambda = -2$ since that is where the maximum likelihood value is achieved. Therefore, we transform our original response variable $X_t$ to obtain a new variable $Y_t$, which is defined as

$$Y_t = -\frac{1}{2}(X_t^{-2} - 1)$$

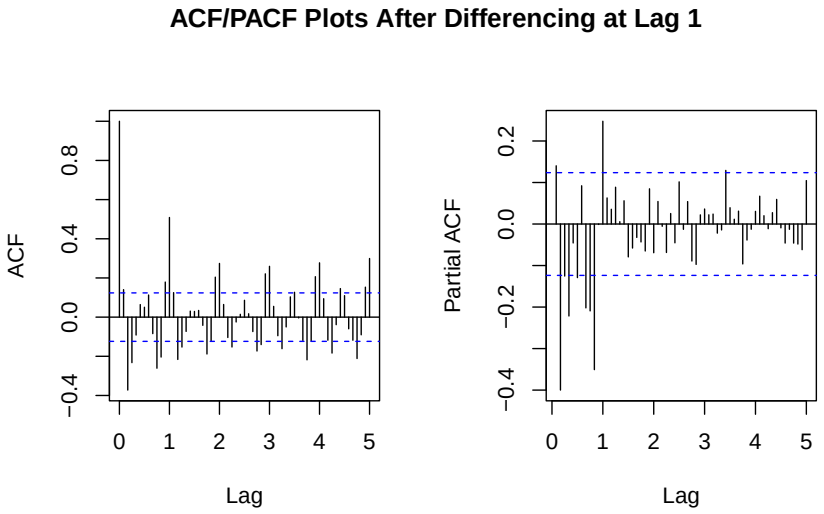
We plot the original and transformed time series below to see if the transformation is necessary.




Other than changing the scale of the $y$-axis, the Box-Cox transformation does not significantly change the overall shape of the original time series. Therefore, we will continue our analysis with the original data $X_t$ (i.e. without the transformation).

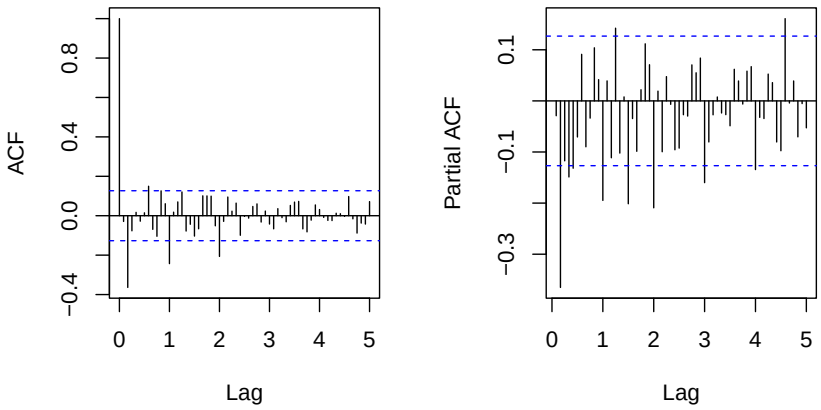
# Differencing

We note that there is a positive linear trend in the original data, so we difference once at lag 1 to remove it.



The original time series has a variance of 3.801 while the differenced time series (at lag 1) has a variance of 0.080, so we move forward with the differenced series. However, the ACF plot exhibits a periodic shape even after removing the linear trend. The large lags at $k = 12, 24, 36, ...$ suggest that there is still some seasonality in the data (recall the dips and peaks within each year), so we also difference the time series once at lag 12.

**ACF/PACF Plots After Differencing at Lags 1 & 12**




After differencing at both lags 1 and 12, the time series has a variance of 0.073, which is less than the variance of the time series differenced only at lag 1. Since we have removed the trend and seasonality in the data, we can move onto identifying the model.

# Model Identification

Since our original data had both trend and seasonality, it most likely follows a SARIMA$(p, d, q) \times (P, D, Q)_s$ model where $p$, $d$, and $q$ are the non-seasonal components of the model, while $P$, $D$, $Q$, and $s$ are the seasonal components of the model.

We differenced once at both lags 1 and 12, so we have $d = 1$, $D = 1$, and $s = 12$.

We first focus on the ACF plot we obtained at the end of the previous section. For the non-seasonal component (i.e. within years), we observe that lags 2 and possibly 7 are significant. For the seasonal component (i.e. between years), we note that lags 12 and 24 are significant since they are outside of the confidence interval. Therefore, we consider $q = 2, 7$ and $Q = 2$.

Now we look at the PACF plot for the same series. For the non-seasonal component, we observe that lags 2, 4, and possibly 5 are significant. For the seasonal component, we note that lags 12, 24, 36, and 48 are significant. Therefore, we consider $p = 2, 4, 5$ and $P = 4$.

We propose the following models:

- **Model 1:** SARIMA$(0, 1, 2) \times (0, 1, 2)_{12}$ (i.e. $q = 2$ and $Q = 2$)
- **Model 2:** SARIMA$(0, 1, 2) \times (0, 1, 7)_{12}$ (i.e. $q = 7$ and $Q = 2$)
- **Model 3:** SARIMA$(2, 1, 0) \times (4, 1, 0)_{12}$ (i.e. $p = 2$ and $P = 4$)
- **Model 4:** SARIMA$(4, 1, 0) \times (4, 1, 0)_{12}$ (i.e. $p = 4$ and $P = 4$)
- **Model 5:** SARIMA$(5, 1, 0) \times (4, 1, 0)_{12}$ (i.e. $p = 5$ and $P = 4$)

We fit each of these models to the original data and ensure that all of the parameters are not within 2 standard errors of 0. The parameters whose confidence intervals contained 0 were dropped from each model.

(The exception here is for Model 5, where trying to set the non-significant parameters to 0 returns an error. In this case, we use the model without any parameters set to 0.) We then find the AICC for each model and compare them in the table below. (Model 4 returns an error, so the corresponding values are listed as NA.)

| | **Model 1** | **Model 2** | **Model 3** | **Model 4** | **Model 5** |
|---|---|---|---|---|---|
| AICC | $-46.34177$ | $-50.52189$ | $-25.03965$ | NA | $-29.02515$ |
| # of parameters estimated | 4 | 4 | 5 | NA | 9 |

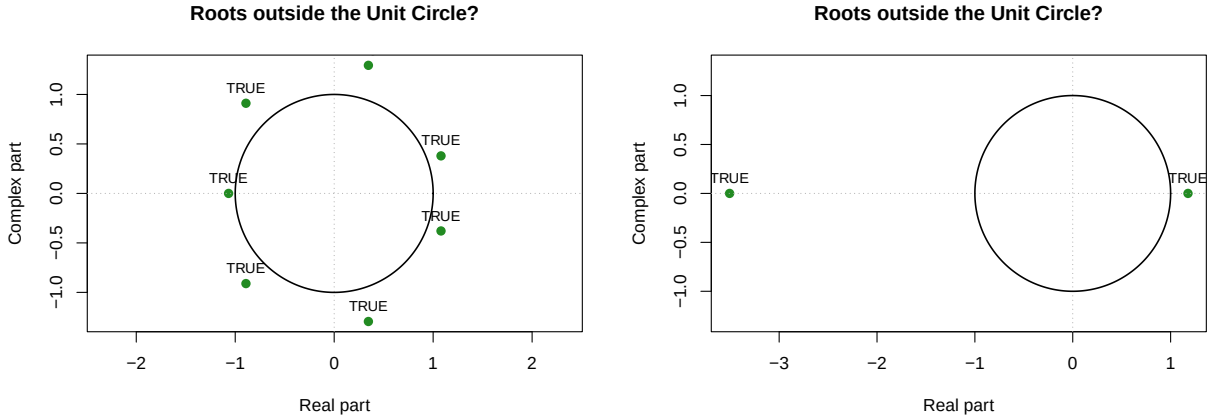
Since we are trying to minimize the AICC, then Model 2 appears to be the most optimal model. It is also comparable to Model 1 since both share the same number of parameters, and we usually want to choose the model with the least number of parameters by the principle of parsimony. Therefore, we use SARIMA$(0,1,2) \times (0,1,7)_{12}$ in the analysis that follows. Our model for the activity rate $X_t$ is

$$(1-B)(1-B^{12})X_t = (1 - 0.5389_{(0.0700)}B^2 + 0.2455_{(0.0721)}B^7)(1 - 0.5640_{(0.0728)}B^{12} - 0.2422_{(0.0696)}B^{24})Z_t$$

where $B$ is the backshift operator and $Z_t$ is a white noise process. The standard errors for the parameter estimates are shown in the subscripts.

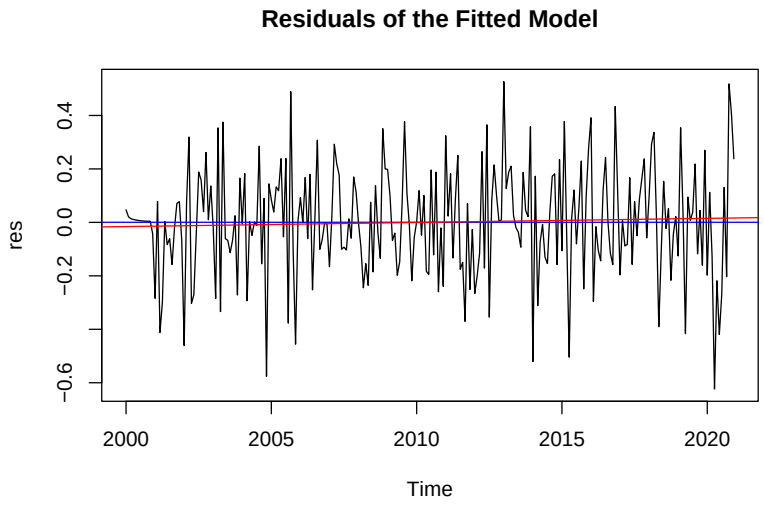
## Stationarity and Invertibility

We want to ensure that our model is both stationary and invertible. Since this is a pure MA process, then it is stationary since it is a linear combination of $Z_t$'s. The roots for both the non-seasonal and seasonal components of the MA part of the model (left and right plots, respectively) are outside of the unit circle, so the model is also invertible.

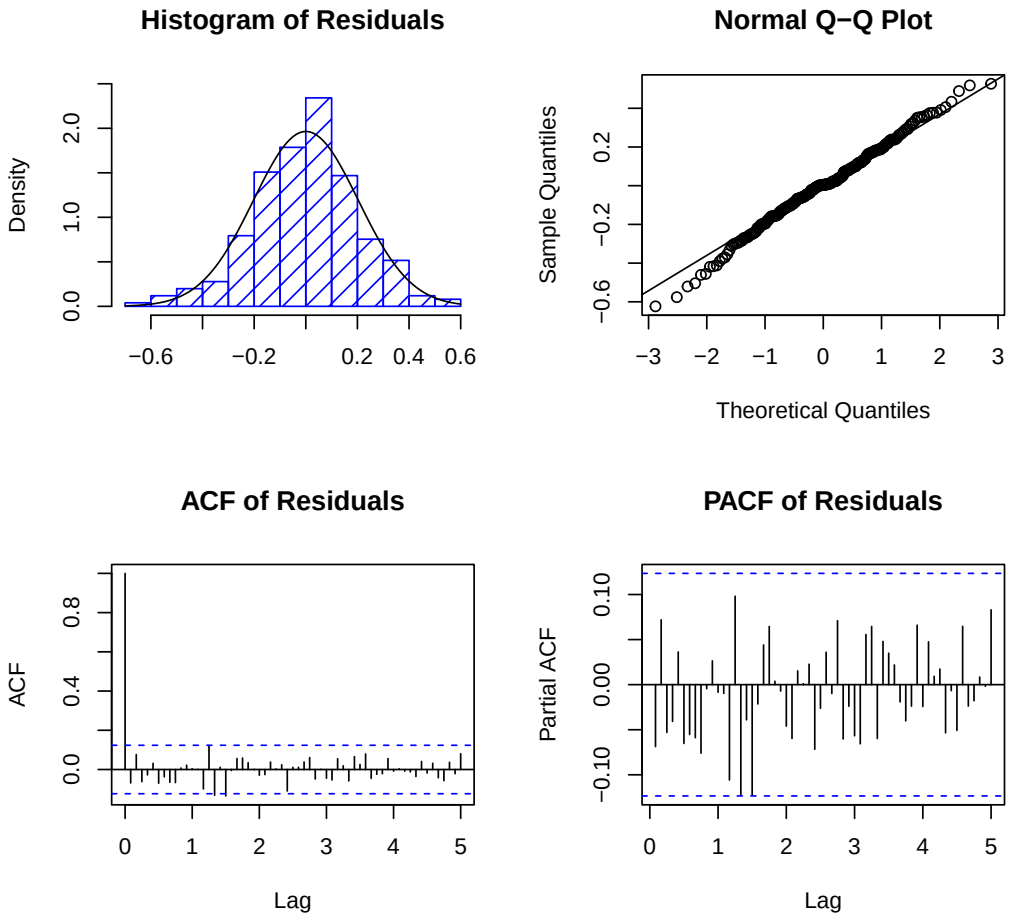



## Diagnostic Checking

Before using our chosen model for forecasting, we need to make sure that it passes several diagnostic tests. More specifically, the residuals of this model should be similar to Gaussian white noise. We plot the residuals on the following page.




The residuals are approximately centered around mean 0 (indicated by the blue line) and don't appear to follow a linear trend (indicated by the red line), which are characteristics of Gaussian white noise. We then visually check whether the residuals roughly follow a normal distribution.



We note the following:

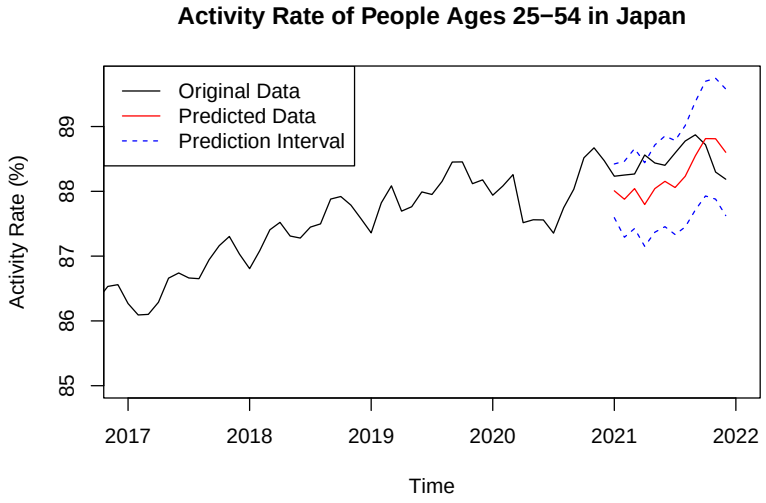
- The histogram has a bell-curved distribution.
- The points in the Q-Q plot follow a straight line for the most part, though there is a bit of deviation at the ends.
- For both the ACF and PACF plots, there are no significant lags for $k \geq 1$. The lags are generally within the confidence intervals, so they are not significantly different from 0.

We also perform the Shapiro-Wilk test and fail to reject the null hypothesis that the residuals are normally distributed since $p = 0.4341 > 0.05$. Combined with the visual checks, we can be confident that the residuals do indeed follow a normal distribution.

To test whether the residuals are correlated or linearly dependent, we use the Box-Pierce and Ljung-Box tests. Since these tests return $p = 0.09814 > 0.05$ and $p = 0.07474 > 0.05$, respectively, we fail to reject the null hypothesis that the residuals are uncorrelated. To test whether the residuals are non-linearly dependent, we use the McLeod-Li test. Since this test returns $p = 0.1092 > 0.05$, we fail to reject the null hypothesis that the residuals do not have non-linear dependence. We also apply Yule-Walker estimation on the residuals, which returns AR(0). Therefore, it is reasonable to assume that the residuals follow a Gaussian white noise process.

# Forecasting

After passing the diagnostic tests, we can predict the activity rates for each month in 2021 using the chosen model.




Out of the 12 test points, only one of them (barely) falls outside of the prediction interval. Since 2020 saw the effects of COVID-19, the predictions for 2021 are slightly lower than the actual data since the model takes into account this recent past. Because of this, there is not much overlap between the predicted and actual data. Overall, the model does a decent job in forecasting, though there may be another model that is better suited for these activity rates.

# Conclusion

In this paper, we have used a SARIMA$(0,1,2)\times(0,1,7)_{12}$ model with 4 parameters for the monthly activity rates of people ages 25-54 in Japan (denoted as $X_t$) for the years 2000-2020. Our model equation is

$$(1-B)(1-B^{12})X_t = (1-0.5389_{(0.0700)}B^2+0.2455_{(0.0721)}B^7)(1-0.5640_{(0.0728)}B^{12}-0.2422_{(0.0696)}B^{24})Z_t$$

Although our model had the lowest AICC out of all of the candidate models and passed all of the diagnostic tests, it tended to underestimate the activity rates for 2021. This is most likely because the data from 2020 was included in our training set, which had an irregular pattern of activity rates compared to other years due to the prevalence of COVID-19. However, as most of the predictions still fell within the prediction interval, this model still performed relatively well.

I would like to thank Professor Feldman for her support on this project.

# References

**Note:** The sources are listed in order of appearance.

1. *Investopedia.* "Labor Force Participation Rate: Purpose, Formula, and Trends." (link)
2. *FRED.* "Activity Rate: Aged 25-54: All Persons for Japan." (link)
3. *Prime Minister's Office of Japan.* "[COVID-19] Declaration of a State of Emergency in response to the Novel Coronavirus Disease (April 16)." (link)

# Appendix

## Exploratory Data Analysis

```
# load packages
library(tidyverse)
library(qpcR)
library(MASS)
library(UnitCircle)

# read data and keep observations between Jan. 2000 - Dec. 2021
activity <- read.csv("data/japan-activity-rate.csv") %>%
  rename(ACTIVITY_RATE = 2) %>%
  filter("2000" < DATE & DATE < "2022")

# create training and test sets
activity.train <- activity %>% slice(1:252)  # Jan. 2000 - Dec. 2020
activity.test <- activity %>% slice(253:264) # Jan. 2021 - Dec. 2021

# convert training set into a time series object
activity.ts <- activity.train %>%
  pull(ACTIVITY_RATE) %>%
  ts(start = c(2000, 1), frequency = 12)

# plot time series for training set
plot(activity.ts,
```

```
     main = "Activity Rate of People Ages 25-54 in Japan",
     xlab = "Year",
     ylab = "Activity Rate (%)")

# add blue dashed lines for each year
abline(v = 2000:2021,
       col = "blue",
       lty = "dashed")
```

### Transformation

```
# determine the optimal value of lambda
t <- time(activity.ts)
bcTransform <- boxcox(activity.ts ~ t, plotit = T)
title("Box-Cox Transformation")

# transform the response variable
lambda <- bcTransform$x[which.max(bcTransform$y)]
activity.ts.bc <- (1/lambda) * (activity.ts^lambda - 1)

# chosen value of lambda
lambda
```

```
## [1] -2
```

```
# 1 row with 2 plots
par(mfrow = c(1, 2))

# plot original data
plot(activity.ts,
     main = "Original Data",
     xlab = "Year",
     ylab = "Activity Rate (%)")

# plot Box-Cox transformed data
plot(activity.ts.bc,
     main = "Transformed Data",
     xlab = "Year",
     ylab = "Transformed Activity Rate (%)")
```

### Differencing

```
# difference once at lag 1 to remove linear trend
d1 <- diff(activity.ts, 1)

# plot ACF and PACF of the differenced series
par(mfrow = c(1, 2))
acf(d1, lag.max = 60, main = "")
pacf(d1, lag.max = 60, main = "")
```

```r
title("ACF/PACF Plots After Differencing at Lag 1",
      line = -1,
      outer = TRUE)

var(activity.ts) # variance of original series
```

```
## [1] 3.80066
```

```r
var(d1) # variance of series differenced at lag 1
```

```
## [1] 0.07995984
```

```r
# difference once at lag 12 to remove seasonality
dd12 <- diff(d1, 12)

# plot ACF and PACF
par(mfrow = c(1, 2))
acf(dd12, lag.max = 60, main = "")
pacf(dd12, lag.max = 60, main = "")
title("ACF/PACF Plots After Differencing at Lags 1 & 12",
      line = -1,
      outer = TRUE)

var(dd12) # variance of series differenced at both lags 1 & 12
```

```
## [1] 0.07301666
```

## Model Identification

**Model 1: SARIMA** $(0, 1, 2) \times (0, 1, 2)_{12}$

```r
# fit model
model1 <- arima(activity.ts,
                order = c(0, 1, 2),
                seasonal = list(order = c(0, 1, 2), period = 12),
                method = "ML")
model1
```

```
##
## Call:
## arima(x = activity.ts, order = c(0, 1, 2), seasonal = list(order = c(0, 1, 2),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1      ma2     sma1     sma2
##       -0.1415  -0.4839  -0.5532  -0.2501
## s.e.   0.0611   0.0646   0.0724   0.0694
##
## sigma^2 estimated as 0.04403:  log likelihood = 28.25,  aic = -46.5
```

**Model 2: SARIMA**$(0,1,2) \times (0,1,7)_{12}$

```
# fit model
model2 <- arima(activity.ts,
                order = c(0, 1, 7),
                seasonal = list(order = c(0, 1, 2), period = 12),
                method = "ML")
model2
```

```
##
## Call:
## arima(x = activity.ts, order = c(0, 1, 7), seasonal = list(order = c(0, 1, 2),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1      ma2      ma3      ma4     ma5      ma6     ma7     sma1
##       -0.0439  -0.5113  -0.0404  -0.0835  0.0734  -0.1269  0.1895  -0.5382
## s.e.   0.0653   0.0658   0.0700   0.0814  0.0734   0.0759  0.0723   0.0765
##          sma2
##       -0.2539
## s.e.   0.0715
##
## sigma^2 estimated as 0.04131:  log likelihood = 34.81,  aic = -49.62
```

```
# ma1, ma3, ma4, ma5, ma6 are within 2 standard errors of 0 -> set them to 0
model2 <- arima(activity.ts,
                order = c(0, 1, 7),
                seasonal = list(order = c(0, 1, 2), period = 12),
                fixed = c(0, NA, 0, 0, 0, 0, NA, NA, NA),
                method = "ML")
model2
```

```
##
## Call:
## arima(x = activity.ts, order = c(0, 1, 7), seasonal = list(order = c(0, 1, 2),
##     period = 12), fixed = c(0, NA, 0, 0, 0, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ma1      ma2  ma3  ma4  ma5  ma6     ma7     sma1     sma2
##         0  -0.5389    0    0    0    0  0.2455  -0.5640  -0.2422
## s.e.    0   0.0700    0    0    0    0  0.0721   0.0728   0.0696
##
## sigma^2 estimated as 0.04294:  log likelihood = 30.63,  aic = -51.27
```

**Model 3: SARIMA**$(2,1,0) \times (4,1,0)_{12}$

```
# fit model
model3 <- arima(activity.ts,
                order = c(2, 1, 0),
                seasonal = list(order = c(4, 1, 0), period = 12),
                method = "ML")
model3
```

```
##
## Call:
## arima(x = activity.ts, order = c(2, 1, 0), seasonal = list(order = c(4, 1, 0),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2     sar1     sar2     sar3     sar4
##       -0.0695  -0.3559  -0.4644  -0.4709  -0.3426  -0.1923
## s.e.   0.0611   0.0608   0.0681   0.0733   0.0737   0.0719
##
## sigma^2 estimated as 0.04819:  log likelihood = 19.34,  aic = -24.67
```

```
# ar1 is within 2 standard errors of 0 -> set it to 0
model3 <- arima(activity.ts,
                order = c(2, 1, 0),
                seasonal = list(order = c(4, 1, 0), period = 12),
                fixed = c(0, NA, NA, NA, NA, NA),
                method = "ML")
```

```
## Warning in arima(activity.ts, order = c(2, 1, 0), seasonal = list(order = c(4, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
model3
```

```
##
## Call:
## arima(x = activity.ts, order = c(2, 1, 0), seasonal = list(order = c(4, 1, 0),
##     period = 12), fixed = c(0, NA, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ar1      ar2     sar1     sar2     sar3     sar4
##         0  -0.3530  -0.4645  -0.4616  -0.3348  -0.1890
## s.e.    0   0.0609   0.0679   0.0733   0.0738   0.0717
##
## sigma^2 estimated as 0.04851:  log likelihood = 18.69,  aic = -25.38
```

**Model 4: SARIMA** $(4,1,0) \times (4,1,0)_{12}$

```
# fit model - returns an error
model4 <- arima(activity.ts,
                order = c(4, 1, 0),
                seasonal = list(order = c(4, 1, 0), period = 12),
                method = "ML")
```

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

## Warning in log(s2): NaNs produced

## Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE, : non-finite finite-differe
```

**Model 5: SARIMA**$(5,1,0) \times (4,1,0)_{12}$

```
# fit model
model5 <- arima(activity.ts,
                order = c(5, 1, 0),
                seasonal = list(order = c(4, 1, 0), period = 12),
                method = "ML")
model5
```

```
##
## Call:
## arima(x = activity.ts, order = c(5, 1, 0), seasonal = list(order = c(4, 1, 0),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5     sar1     sar2     sar3
##       -0.1363  -0.4365  -0.1762  -0.1695  -0.1234  -0.4586  -0.4429  -0.3325
## s.e.   0.0646   0.0650   0.0717   0.0672   0.0664   0.0692   0.0751   0.0746
##          sar4
##       -0.1929
## s.e.   0.0723
##
## sigma^2 estimated as 0.04606:  log likelihood = 24.88,  aic = -29.77
```

```
# ar5 is within 2 standard errors of 0 -> set it to 0
# however, this returns an error
# we stick with the model without any parameters set to 0
model5 <- arima(activity.ts,
                order = c(5, 1, 0),
                seasonal = list(order = c(4, 1, 0), period = 12),
                fixed = c(NA, NA, NA, NA, 0, NA, NA, NA, NA),
                method = "ML")
```

```
## Warning in arima(activity.ts, order = c(5, 1, 0), seasonal = list(order = c(4, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

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

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

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

```
## Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE, : non-finite finite-differe
```

```
# get the AICC for each model
c("Model 1" = AICc(model1),
  "Model 2" = AICc(model2),
  "Model 3" = AICc(model3),
  "Model 4" = NA,           # returns an error
  "Model 5" = AICc(model5)) # none of the parameters are set to 0
```

```
##   Model 1   Model 2   Model 3   Model 4   Model 5
## -46.34177 -50.52189 -25.03965        NA -29.02515
```

### Stationarity and Invertibility

```
# check that the roots are outside of the unit circle
par(mfrow = c(1, 2))
uc.check(c(1, 0, -0.5389, 0, 0, 0, 0, 0.2455), print_output = FALSE)
uc.check(c(1, -0.5640, -0.2422), print_output = FALSE)
```

### Diagnostic Checking

```
# plot residuals
res <- residuals(model2)
plot(res, main = "Residuals of the Fitted Model")

# plot mean (blue) and trend (red) lines
t <- time(res)
abline(h = mean(res), col = "blue")
abline(lm(res ~ t), col = "red")

# 2 rows with 2 plots per row
par(mfrow = c(2, 2))

# histogram for the residuals with normal curve overlaid
hist(res,
     density = 10,
     breaks = 10,
     col = "blue",
     xlab = "",
     prob = TRUE,
     ylim = c(0, 2.5),
     main = "Histogram of Residuals")
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = TRUE)

# Q-Q plot for the residuals
qqnorm(res)
qqline(res)

# ACF and PACF plots for the residuals
acf(res, lag.max = 60, main = "ACF of Residuals")
pacf(res, lag.max = 60, main = "PACF of Residuals")

# run Shapiro-Wilk test
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99412, p-value = 0.4341
```

```r
# get h to use in the tests that follow
n <- length(activity.ts)
h <- round(sqrt(n))

Box.test(res, lag = h, type = "Box-Pierce", fitdf = 4) # Box-Pierce test
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 18.619, df = 12, p-value = 0.09814
```

```r
Box.test(res, lag = h, type = "Ljung-Box", fitdf = 4) # Ljung-Box test
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 19.615, df = 12, p-value = 0.07474
```

```r
Box.test(res^2, lag = h, type = "Ljung-Box", fitdf = 0) # McLeod-Li test
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 23.172, df = 16, p-value = 0.1092
```

```r
# use Yule-Walker estimation to fit residuals
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.04123
```

## Forecasting

```r
# get both the training and test data
activity.ts.all <- activity %>%
  slice(1:264) %>%
  pull(ACTIVITY_RATE) %>%
  ts(start = c(2000, 1), frequency = 12)

# calculate the upper and lower bounds of the prediction interval
preds <- predict(model2, n.ahead = 12)
upper.bound <- preds$pred + 2 * preds$se
```

```r
lower.bound <- preds$pred - 2 * preds$se

# plot the original data
plot(activity.ts.all,
     main = "Activity Rate of People Ages 25-54 in Japan",
     ylab = "Activity Rate (%)",
     xlim = c(2017, 2022),
     ylim = c(85, max(upper.bound)))

# plot the upper and lower bounds of the prediction interval
lines(upper.bound, col = "blue", lty = "dashed")
lines(lower.bound, col = "blue", lty = "dashed")

# plot the predictions
lines(seq(2021, 2022, length.out = 13)[1:12],
      preds$pred,
      col = "red")

# add legend
legend("topleft",
       legend = c("Original Data", "Predicted Data", "Prediction Interval"),
       col = c("black", "red", "blue"),
       lty = c("solid", "solid", "dashed"))
```