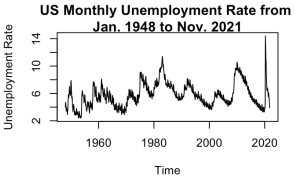
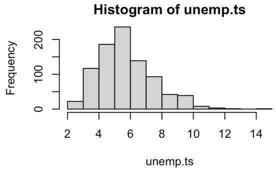
Time Series Analysis: Unemployment Rates in the US

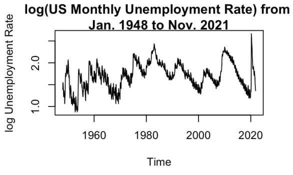
ST566: TIME SERIES ANALYTICS

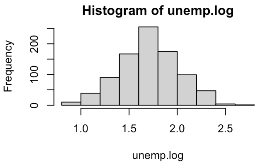
Introduction

The unemployment rate is defined as the percentage of unemployed workers in the total labor force. Workers are considered unemployed if they currently do not work. The total labor force is the combined population of employed and unemployed people within an economy. Unemployment tends to be cyclical and decreases when the economy expands as companies contract more workers to meet growing demand. Monthly unemployment rates in the US from January 1948 to November 2021 are shown below. The goal of this analysis is to conduct a comprehensive analysis of the unemployment rate time series data using statistical methodologies learned in *ST566: Time Series Analytics*. The aspects of decomposing the series, ARMA, ARIMA modeling and frequency domain analysis will be summarized and described.









Methods and Results

Evaluation and Decomposition

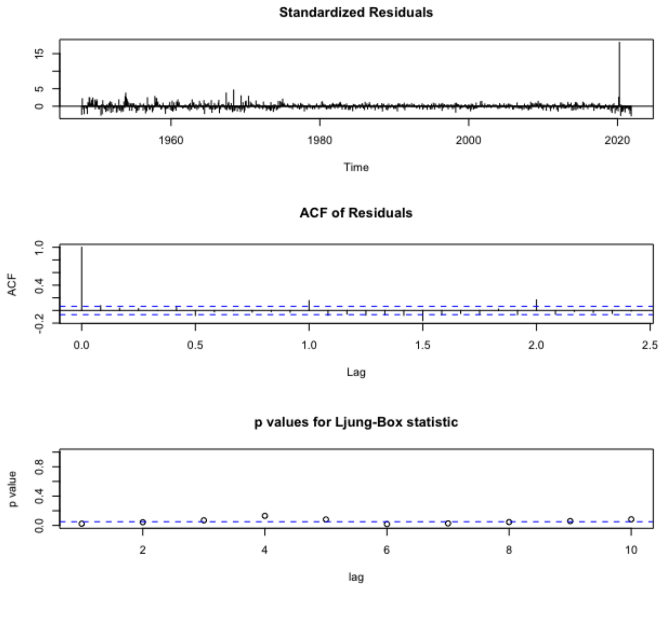
The time plot of the original series shows a cubic trend and a seasonality pattern, clearly depicting the non-stationarity of the series. There are noticeable inconsistencies and changes in variance, with the most notable variance gap appearing in 2020. The logarithm of the original series is taken to help stabilize the data. The histogram of the original series shows a long tail to the right of the distribution, suggesting an exponential or long-tail distribution. As shown above, the log transformation helps to remove exponential variance that exists within the original series, albeit trend and seasonality patterns still exist. The histogram of the log of the series now shows a more uniform distribution of observations. By eliminating the exponential variance and now depicting a more constant variance on the log plot, taking the log of the original series is an appropriate transformation to stabilize the data.

Trend and seasonality were analyzed and removed. To get a better idea of the long-term trend in unemployment rates in the US, kernel smoothing was performed to estimate the non-linear trend and to fit the series with the best balance between smoothness and variability. The function *loess()* was used in R to fit the nonparametric curve. The span parameter, which specifies the percentage of data points used in the local fit, of the loess function was set at 0.15 to control the smoothness of the estimated trend. An abrupt change in the series occurred in 2020. During this time period, unemployment rates spiked, reaching their highest peak before rapidly declining, which differs from previous time periods. Other notable changes in the series where there was a spike in unemployment rates occurred in 1983 and 2010. To examine the seasonal pattern, the residuals of the nonparametric trend (loess) were plotted by year. The evidence of seasonality is clear in the residuals due to the presence of variations that occur at specific regular intervals. To get a clearer picture of seasonality alone, the month was collected for each time point, then the means were calculated at each month. After deducting each point by the corresponding monthly mean, a time series object containing only seasonality was created.

Upon removing both trend and seasonality from the series, the plot was obtained by performing differencing to remove both trend and seasonality at the same time. The R function *decompose()* was also used to automatically decompose the series into seasonal, trend and irregular components using moving averages. The decompose function additionally allowed for being able to quickly compare all the plots using a single function. Examining the residual plots there does not appear to be any discernible long-term trend, seasonality or obvious pattern that needs to be transformed. There however is a clear outlier in the residual plot in 2020. The residual noise appears to be random, thus stationarity has been met. The greatest source of variation in US monthly unemployment rates is the cyclic pattern, higher unemployment rates during impactful events, such as COVID, in 2020. Upon removal of long-term trend and seasonality, the residual noise displays stationarity, and no obvious patterns are visibly noticed within the series. Next,ARIMA model will be fit to the residual series and examine the ACF and PACF plots to find the best-fitted model for the series.

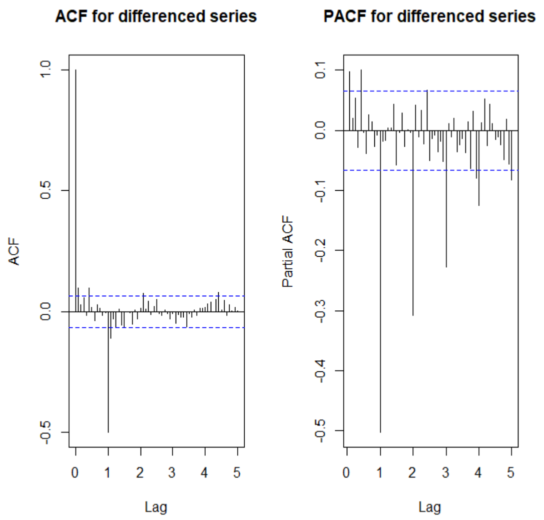
ARMA MODEL

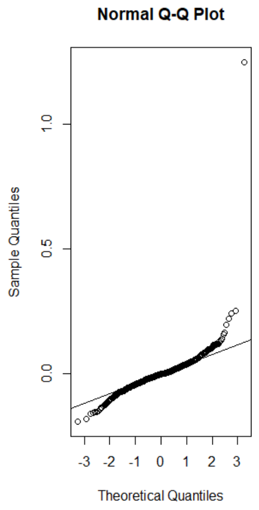
In addition, an ARMA analysis was also performed in order to see if a different type of model could perform best against the Holt Winters and SARIMA models. Using the residual series, ACF and PACF plots were generated and examined to determine appropriate models. The ACF plot decreases to zero and the PACF plot is non-zero at lag 3. This suggests that the most viable models are an AR(3) model or an ARMA model. Several models were fit: AR(1), AR(2), AR(3), ARMA(1, 1) and ARMA(1, 2). Both the AR(3) and the ARMA(1,1) had similar AIC (-2291.2 and 2289.8, respectively) but the former was chosen. In terms of performance, the plots below show the AR(3) is not a suitable model for the unemployment dataset. The standardized residuals in the first plot do not look as if they are white noise. It is also apparent in the ACF plot that some serial correlation in those residuals are significant at lag 1, 1.5 and 2. Moreover, all the p-values for the Ljung-Box statistic are small, with some dropping down to significance level. This strongly suggests that the residuals are not independent and therefore not white noise.



SARIMA MODEL

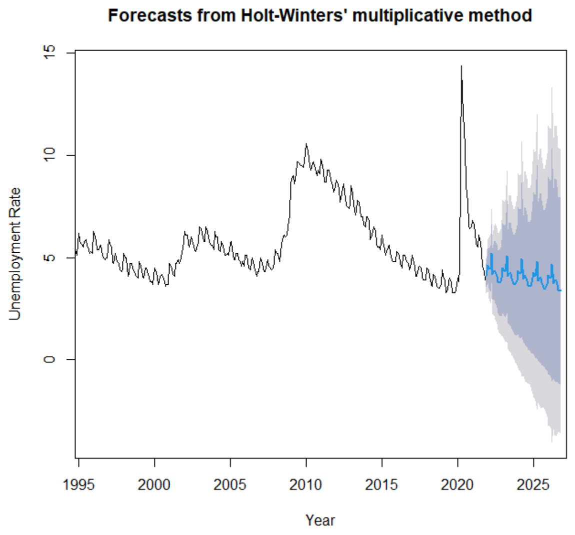
In an attempt to address the issues with the ARMA model fit, a first order difference of the log of the series was performed, then a seasonal difference of period 12. From the the ACF and PACF plots of the seasonal differenced series below, it looks like the best model might be a SARIMA(1,1,1)x(0,1,2)_12, but a couple other SARIMA models were fit and selected the best model based on AIC. From the models that were fit, the SARIMA(1,1,1)x(0,1,2)_12 resulted in the lowest AIC. Examining the ACF and PACF of the residual series from the model fit, it appears to be that of white noise. Moreover, there don't appear to be any issues with the Ljung-Box statistic, but there are some concerns about the normality of the residuals due to the outliers discussed previously, which can be seen in the QQ plot of the residuals.



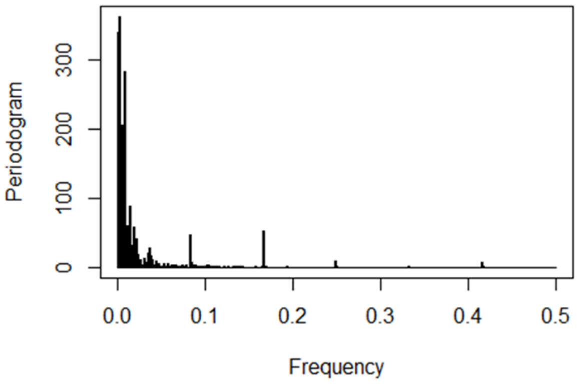


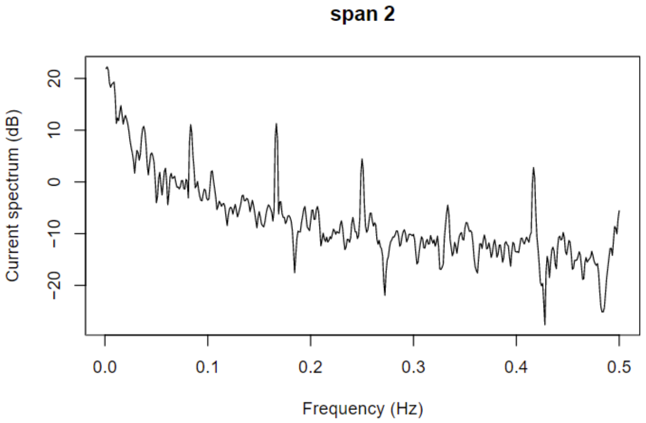
Holt-Winters and Forecasting

Due to the concerns about the normality of the residuals, it seemed best to use an alternative method to forecast. So the Holt-Winters forecasting method was used to forecast 5 years ahead, as seen below.



Spectral Analysis





A spectral analysis has been performed of the unemployment data. The periodogram (left plot) generally shows the importance of the different frequencies and from the look of it, there are specific frequencies that contribute more compared to a white noise process which should have an approximately equal contribution of each frequency. It can be seen that low frequencies have a high contribution - seen as peaks in the plot - and can be correlated to seasonality and trend which causes a large value of the first cosine coefficient and is of less importance for us. Additionally, it can be seen that higher frequencies in steps have some contribution which is correlated to the seasonality with steps of approximately i/12 with i=1…n which is a multiple of the seasonal frequency. Furthermore, the data has been smoothed (right plot) in periodograms to reduce the variance while not introducing too much bias. The span size of 2 seemed to provide good results. The smoothed periodogram shows again, as the initial periodogram, peaks at multiples of the seasonal frequency, showing that the seasonality of the data is an important factor.

Discussion

Multiple investigations of the unemployment data have been performed. The methods used included evaluating and decomposing the data, ARMA, SARIMA, as well as Holt-Winters forecasting, and spectral analysis.

While some methods, like the decomposition, provide insight into the data behavior, it was ultimately an insufficient approach. Some of the challenges included correctly modeling the trend as well as influence of outside factors which resulted in outliers in the observed series. Therefore, the decision to use Holt-Winters method for forecasting was the best path forward, since other models performed poorly. Finally, the spectral analysis showed some significance of the seasonality in the data but did not bring that much insight as the analysis is probably better suited for different applications, like for example in engineering.

R CODE APPENDIX

```
#### read unemployment data
unemp <- read.csv("unemployment_rate_data.csv")
#remove extraneous columns from data
unemp <- unemp[1:2]
#head(unemp)

#### transform data into time series object
unemp.ts <- ts(unemp[2], start=c(1948,1),end=c(2021,11), frequency=12)
#print(unemp.ts)
# take logarithm of monthly unemployment rate to stabilize data
unemp.log <- log(unemp.ts)

#### plot the original series
par(mfrow =c(2, 2))
plot(unemp.ts, ylab = "Unemployment Rate",
     main = "US Monthly Unemployment Rate from \nJan. 1948 to Nov. 2021")
# histogram of original series
hist(unemp.ts)
#### time series plot of logarithms of series
plot(unemp.log, ylab = "log Unemployment Rate",
     main = "log(US Monthly Unemployment Rate) from \nJan. 1948 to Nov.
2021")
# histogram of log series
hist(unemp.log)
```

```
# create a time variable
unemp.time <- time(unemp.ts)

##### TREND ANALYSIS (Kernel Smoothing) FOR LOG
# fit a nonparametric trend
unemp.loess.log <- loess(unemp.log ~ unemp.time, span = 0.15)
# get the trend
unemp.loess.pred.log <- predict(unemp.loess.log)
# change it into a time series object
unemp.loess.trend.log <- ts(unemp.loess.pred.log, start=c(1948,1),
end=c(2021,11), frequency = 12)
# overlay the trend on the time plot
plot(unemp.log, ylab = "log(Unemployment Rate)")
lines(unemp.loess.trend.log, col = "blue", lty = 1, lwd = 5)
```

```
##### Trend removal
# cubic trend removed & formatted into time series
# unemp.loess.trend.log
# get the residuals
unemp.res.log <- unemp.loess.log$residuals
# change it into time series
unemp.res.log <- ts(unemp.res.log, start=c(1948,1), end=c(2021,11), deltat =
1/12)
# plot it
plot(unemp.res.log, xlab = "Year", ylab = "Residuals after removing trend")
```

```
##### Seasonality removal
unemp.month <-factor(cycle(unemp.res.log))
fit.season <-lm(unemp.res.log ~ unemp.month)
# estimates seasonal components
unemp.season <-ts(fit.season$fitted, start=c(1948,1), end=c(2021,11), deltat
= 1/12)

#### With both linear trend and seasonality removed
unemp.rand <-ts(unemp.res.log - unemp.season, start=c(1948,1),
end=c(2021,11), deltat = 1/12)

### plot the decomposition (trend, seasonality, and residual)
par(mfrow =c(2, 2))
plot(unemp.log, xlab = "Year", ylab = "log(Unemployment Rate)")
plot(unemp.loess.trend.log, xlab = "Year", ylab = "Trend")
plot(unemp.season, xlab = "Year", ylab = "Seasonality")
plot(unemp.rand, xlab = "Year", ylab = "Random")
```

```
#### DIFFERENCING: remove both trend and seasonality at the same time
#plot(diff(unemp.log), ylab = "Difference")



#First order difference

diff1 <- c(NA, diff(unemp.log))

diff1 <- ts(diff1, start=c(1948,1), end=c(2021,11), deltat = 1/12)

plot(diff1, xlab = "Year", ylab = "First Order Differenced Series")



# Seasonal Difference

diff12 <- c(NA, diff(diff1, lag = 12))

diff12 <- ts(diff12, start=c(1948,1), end=c(2021,11), deltat = 1/12)

plot(diff12, xlab = "Year", ylab = "Seasonal Difference at Lag 12")
```

```
# Check the ACF and PACF of differenced series
 par(mfrow = c(1, 2))
 acf(diff12, lag.max = 60, na.action = na.pass,
 main = "ACF for differenced series")


 pacf(diff12, lag.max = 60, na.action = na.pass,
 main = "PACF for differenced series")



# Fitting models based on AFC and PACF plots, then selecting based on AIC

# SARIMA(1,1,4) x (0,1,2)_12

model1 <- arima(unemp.log, order = c(1, 1, 4),

                seasonal = list(order = c(0, 1, 2), period = 12))



model1



# SARIMA(1,1,1) x (0,1,2)_12

model2 <- arima(unemp.log, order = c(1, 1, 1),

                seasonal = list(order = c(0, 1, 2), period = 12))

model2



# SARIMA(3,1,4) x (0,1,2)_12

model3 <- arima(unemp.log, order = c(3, 1, 4),

                seasonal = list(order = c(0, 1, 2), period = 12))

model3



# SARIMA(4,1,4) x (0,1,2)_12

model4 <- arima(unemp.log, order = c(4, 1, 4),

                seasonal = list(order = c(0, 1, 2), period = 12))
```

```
df <- rbind("SARIMA(1,1,4) x (0,1,2)_12" = c(model1$aic, model1$loglik),
            "SARIMA(1,1,1) x (0,1,2)_12" = c(model2$aic, model2$loglik),
            "SARIMA(3,1,4) x (0,1,2)_12" = c(model3$aic, model3$loglik),
            "SARIMA(4,1,4) x (0,1,2)_12" = c(model4$aic, model4$loglik))
df <- as.data.frame(df)
names(df) <- c("AIC", "Log Likelihood")
df

#### Model Diagnostics on model3 = SARIMA(1,1,1) x (0,1,2)_12
## Fitting residuals and ACF/PACF plots
par(mfrow = c(1, 2))
res <- model2$residuals
acf(res, lag.max = 36)
pacf(res, lag.max = 36)

# Diagnostic plots
tsdiag(model2)

# QQ plot of residuals
# Outlier but probably normal enough
qqnorm(res)
qqline(res)
```

```
#### Forecasting

# Get forecasted values, h=60

pred <- predict(model2, n.ahead = 60)


# Plot forecasted values, change xlim to get better view of forecast

plot(unemp.log, xlim = c(1996, 2026), ylim = c(-1, 3.25),

     main = "Forecast of Log(Unemployment Rate)",

     ylab = "log(Unrate)")


#### Forecasted values

lines(pred$pred, col = "red")


#### 95% forecasting limits

lines(pred$pred-2*pred$se,col='blue')

lines(pred$pred+2*pred$se,col='blue')


#### Legend

legend("bottomleft", legend = c("Forecasted Values", "95% Forecasting Limits"),

       col = c("red", "blue"), lty = 1)


#### Backtransformed

# Plot forecasted values, change xlim to get better view of forecast

plot(unemp.ts, xlim = c(1996, 2026), ylim = exp(c(-1, 3.25)),

     main = "Forecast of Unemployment Rate",

     ylab = "Unemployment Rate")
```

```
#### Forecasted values

lines(exp(pred$pred), col = "red")


#### 95% forecasting limits

lines(exp(pred$pred-2*pred$se),col='blue')

lines(exp(pred$pred+2*pred$se),col='blue')


#### Legend

legend("topleft", legend = c("Forecasted Values", "95% Forecasting Limits"),

       col = c("red", "blue"), lty = 1)


#### Holt-Winters Forecasting

#### Holt-Winters Forecasting

library(forecast)

fore.log.unemp <- hw(log(unemp.ts), seasonal = "multiplicative", h=60)

plot(fore.log.unemp, xlim = c(1996, 2026))

fore.unemp <- hw(unemp.ts, seasonal = "multiplicative", h=60)

plot(fore.unemp, xlim = c(1996, 2026), ylab = "Unemployment Rate", xlab =
"Year")
```

```
#### DECOMPOSE the series
unemp.dec.log <- decompose(unemp.log)
# plot everything
plot(unemp.dec.log)
```

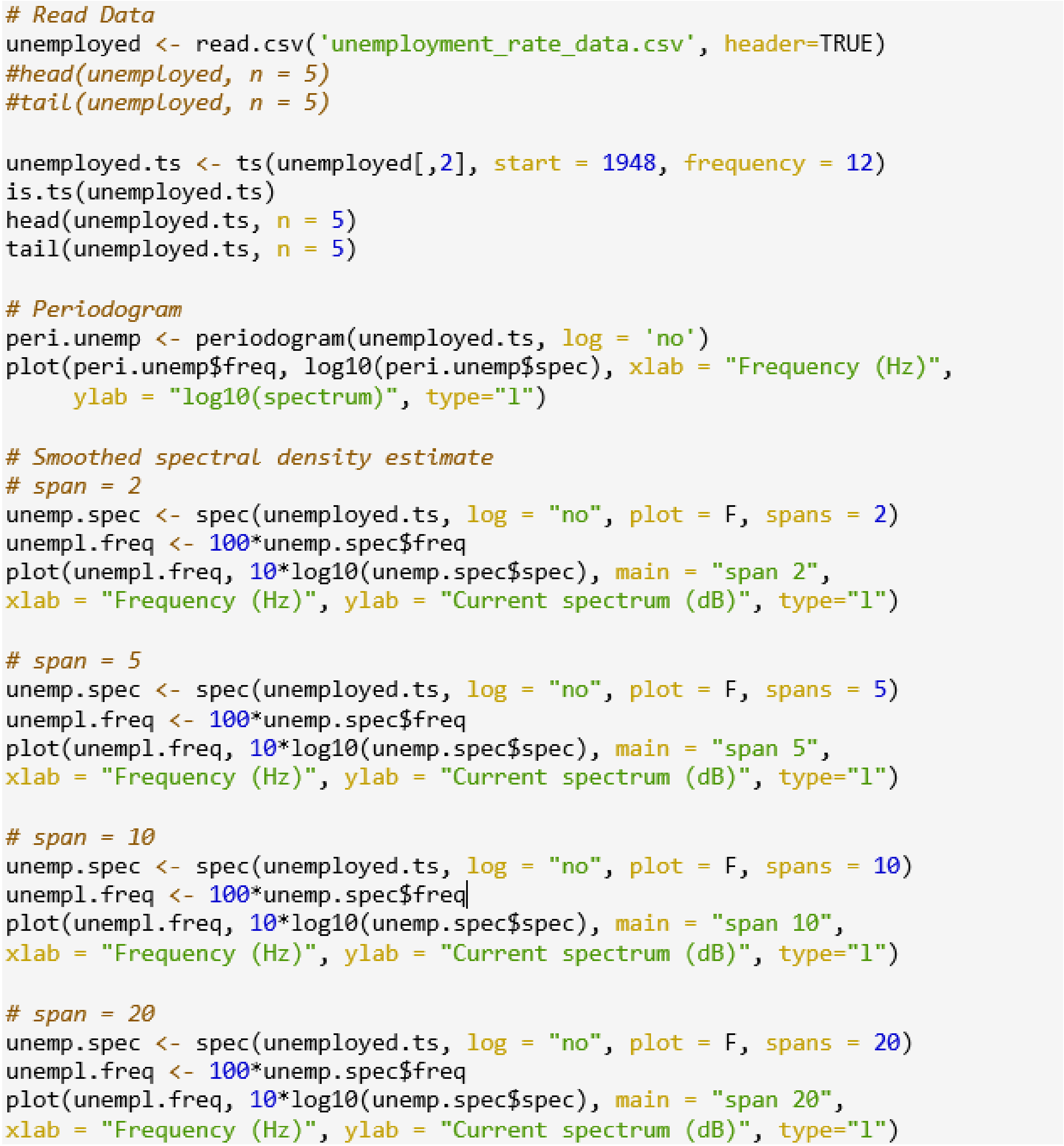
ARMA Model assessment

```
# read unemployment data

unemp <- read.csv("unemployment_rate_data.csv")

unemp <- unemp[1:2] # subset series


# transform data into time series object

unemp.ts <- ts(unemp[2], start=c(1948,1),end=c(2021,11), frequency=12)

unemp.log <- log(unemp.ts)


# plot original series

plot(unemp.ts, main = "Unemployment series (original)")


# Plot the log series

plot(unemp.log, main = "Unemployment series (Log)")


# create a time variable

unemp.time <- time(unemp.ts)


# fit a nonparametric trend

unemp.loess.log <- loess(unemp.log ~ unemp.time, span = 0.15)


# get the trend

unemp.loess.pred.log <- predict(unemp.loess.log)


# change it into a time series object

unemp.loess.trend.log <- ts(unemp.loess.pred.log, start=c(1948,1),
end=c(2021,11), frequency = 12)
```

```
# Remove Trend

unemp.res.log <- unemp.loess.log$residuals


# change it into time series

unemp.res.log <- ts(unemp.res.log, start=c(1948,1), end=c(2021,11), deltat =
1/12)


# Remove Seasonality

unemp.month <-factor(cycle(unemp.res.log))

fit.season <-lm(unemp.res.log ~ unemp.month)


# estimate seasonal components

unemp.season <-ts(fit.season$fitted, start=c(1948,1), end=c(2021,11), deltat
= 1/12)


# Remove seasonality and trend

unemp.rand <-ts(unemp.res.log - unemp.season, start=c(1948,1),
end=c(2021,11), deltat = 1/12)


# check acf and pacf of the residual series

par(mfrow = c(1, 2))

acf(unemp.rand, main = "ACF of the residual series",lag.max=60)

pacf(unemp.rand, main = "PACF of the residual series",lag.max=60)


# might be AR or ARMA so let's try different models and grab the lowest AIC

for (p in seq(1, 3)) {
```

```r
  t_out <- arima(unemp.rand, order = c(p = p,d = 0,q = 0), method = "ML",
include.mean = F)

  model_name <- paste0("AR(", p, ") AIC: ", t_out$aic)

  print(model_name)

}



# loop for ARMA

for (p in seq(1, 2)) {

  t_out <- arima(unemp.rand, order = c(p = p, d = 0, q = 1), method = "ML",
include.mean = F)

  model_name <- paste0("ARMA(", p, ", 1) AIC: ", t_out$aic)

  print(model_name)

}

# Fit the best model we found, which is the AR(1)

fit.ar3 <- arima(unemp.rand, order = c(p = 3, d = 0, q = 0), method = "ML",
include.mean = F)



## Diagnostics ##

par(mfrow = c(1, 2))

res <- fit.ar3$residuals



# Plot Residuals

acf(res, main = "ACF for residuals")

pacf(res, main = "PACF for residuals")



par(mfrow = c(1, 1))

tsdiag(fit.ar1)
```

Spectral Analysis R Code



```
# Read Data
unemployed <- read.csv('unemployment_rate_data.csv', header=TRUE)
#head(unemployed, n = 5)
#tail(unemployed, n = 5)

unemployed.ts <- ts(unemployed[,2], start = 1948, frequency = 12)
is.ts(unemployed.ts)
head(unemployed.ts, n = 5)
tail(unemployed.ts, n = 5)

# Periodogram
peri.unemp <- periodogram(unemployed.ts, log = 'no')
plot(peri.unemp$freq, log10(peri.unemp$spec), xlab = "Frequency (Hz)",
     ylab = "log10(spectrum)", type="l")

# Smoothed spectral density estimate
# span = 2
unemp.spec <- spec(unemployed.ts, log = "no", plot = F, spans = 2)
unempl.freq <- 100*unemp.spec$freq
plot(unempl.freq, 10*log10(unemp.spec$spec), main = "span 2",
xlab = "Frequency (Hz)", ylab = "Current spectrum (dB)", type="l")

# span = 5
unemp.spec <- spec(unemployed.ts, log = "no", plot = F, spans = 5)
unempl.freq <- 100*unemp.spec$freq
plot(unempl.freq, 10*log10(unemp.spec$spec), main = "span 5",
xlab = "Frequency (Hz)", ylab = "Current spectrum (dB)", type="l")

# span = 10
unemp.spec <- spec(unemployed.ts, log = "no", plot = F, spans = 10)
unempl.freq <- 100*unemp.spec$freq
plot(unempl.freq, 10*log10(unemp.spec$spec), main = "span 10",
xlab = "Frequency (Hz)", ylab = "Current spectrum (dB)", type="l")

# span = 20
unemp.spec <- spec(unemployed.ts, log = "no", plot = F, spans = 20)
unempl.freq <- 100*unemp.spec$freq
plot(unempl.freq, 10*log10(unemp.spec$spec), main = "span 20",
xlab = "Frequency (Hz)", ylab = "Current spectrum (dB)", type="l")
```