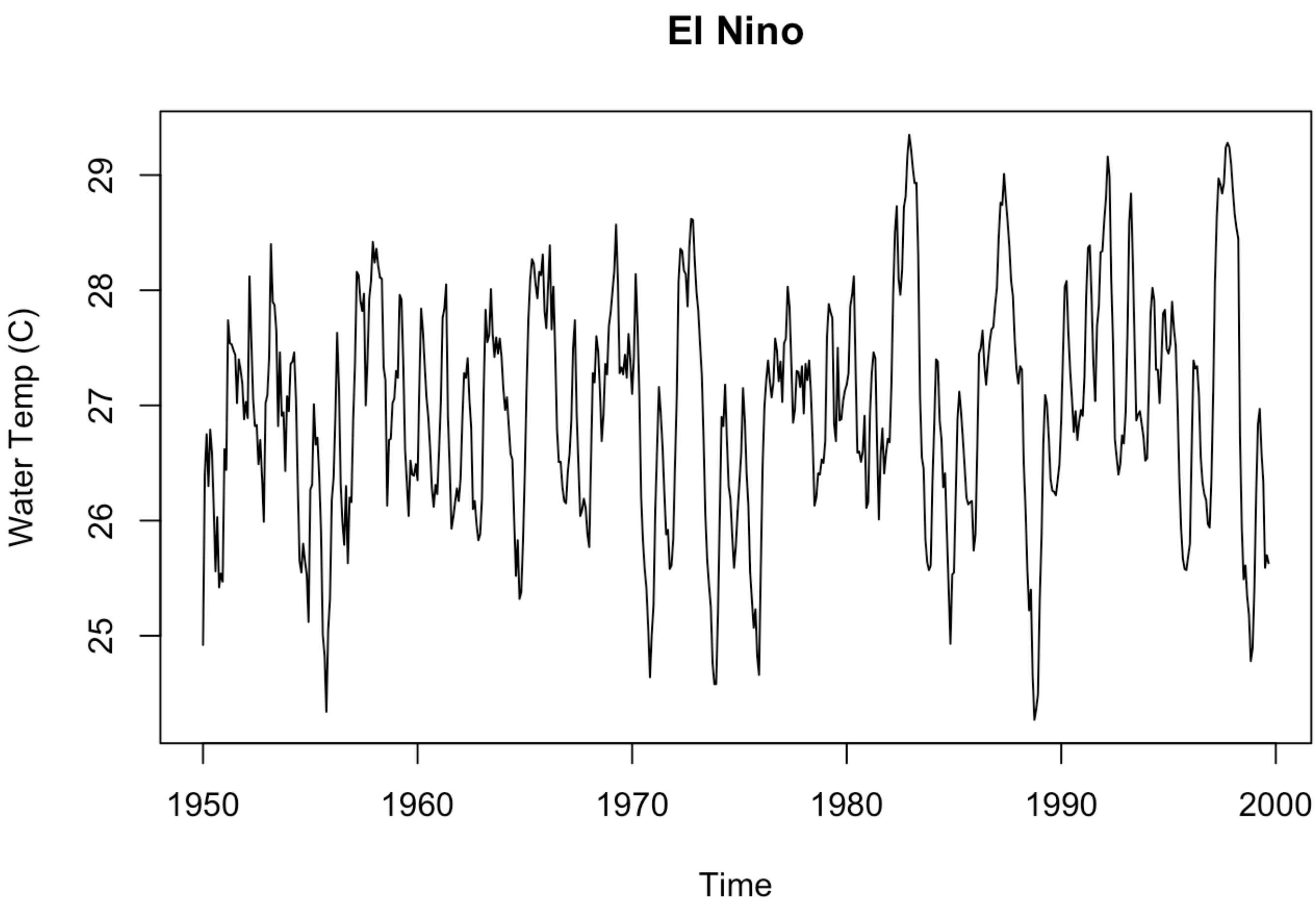
Hunter Garfield ECON 522 Final

Problem 1

```
d = read.csv("nino.csv")
d = ts(d, start = 1950, frequency = 12)
```

Problem 2

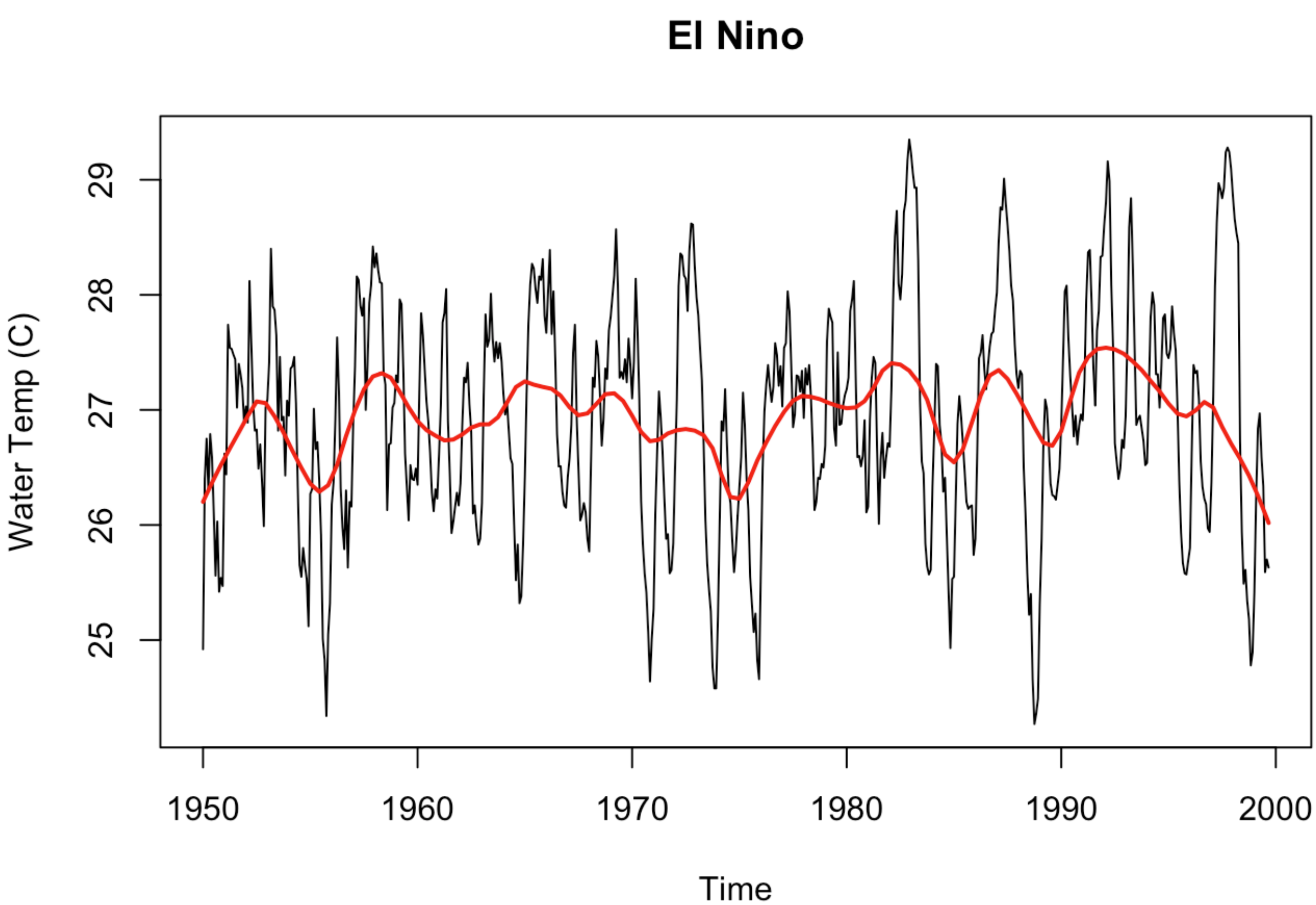


Problem 3

Based on only the plot of the data above, it is hard to tell if a Box-Cox transformation would be useful. Box-Cox transformations are used to stabilize sample variance and make data sets more normally distributed. While we may not care so much about the normal distribution aspect (although we would if it also applied to our residuals), we can tell that there may be a little bit of heteroskedasticity in our model because the distances between peaks and valleys varies quite a bit over time. Therefore, a Box-Cox transformation may be helpful if we specify the value of lambda correctly.

It may be a good idea to analyze this data differenced because, though it does look somewhat like white noise, it is easy to tell that the data has a mean (around 27 by the looks of the plot). It is not hard to imagine that the mean of ocean temperatures for the last 50 years would be dependent on time (global warming,etc.). Since we prefer to work with stationary data when using a lot of our fitting methods, de-meaning this data by differencing it would be helpful.

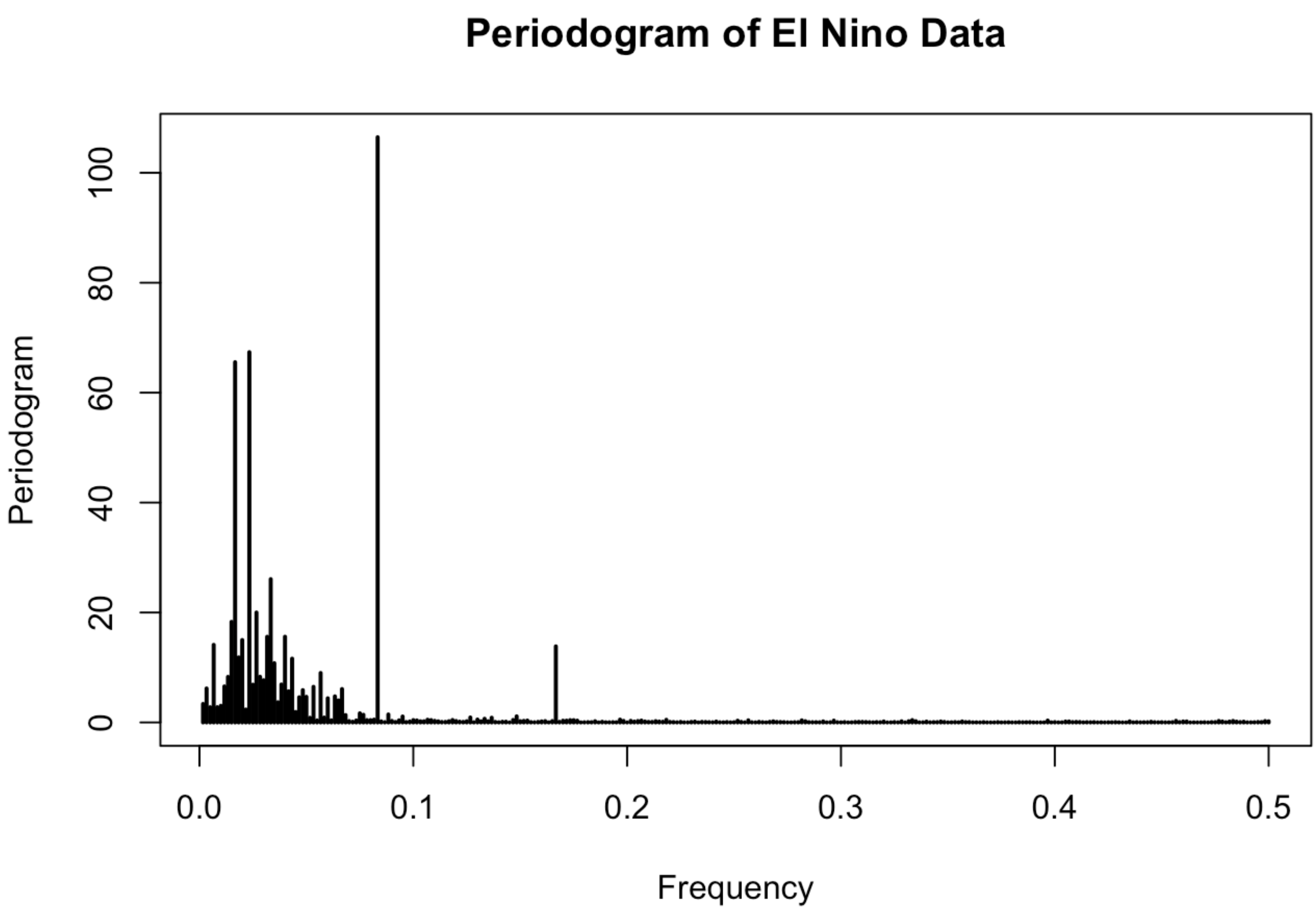
Problem 4



Based on the lowess smoother, it does appear that there is cyclicality in this data for periods longer than a year. It is not perfectly symmetric, but we can see a gradual sine wave that starts in 1950 and ends a couple of years before 1960. The wave breaks up a little bit after this, but we can still see upward and downward movements, and it looks like there is a similar wave between the periods just after 1980 and just before 1990.

Problem 5

```
trend = time(d)
fit = lm(d~0+trend)
detrend = d - fit$fitted.values
per = periodogram(detrend, main="Periodogram of El Nino Data") #requires "TSA" pac
kage
```



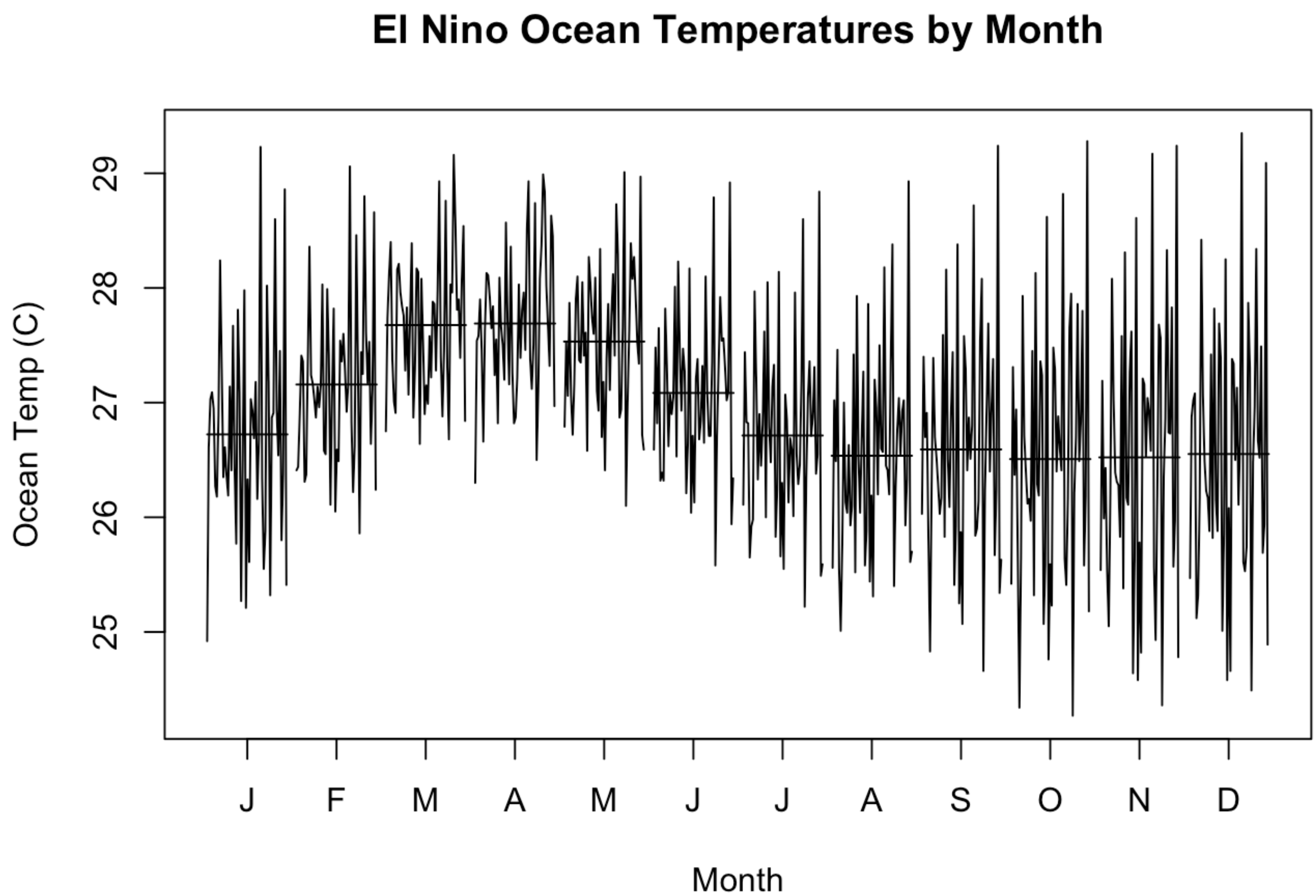
We can tell by the periodogram output by the function that there are a few peak frequencies of interest. The main one clearly occurs just before the 0.1 frequency, and we are also interested in the ones that occur in between the tallest peak and zero. We can find out what these frequencies are using the code below:

```
m1 = max(per$spec) #find highest value
m1ind = which(per$spec==max(per$spec)) #find index of highest value
f1 = per$freq[m1ind] #find corresponding frequency by index
per$spec = per$spec[-m1ind] #get rid of first highest and search for second
per$freq = per$freq[-m1ind]

#rinse and repeat
m2 = max(per$spec)
m2ind = which(per$spec == max(per$spec))
f2 = per$freq[m2ind]
per$spec = per$spec[-m2ind] #get rid of second highest and search for third
per$freq = per$freq[-m2ind]

m3 = max(per$spec)
m3ind = which(per$spec == max(per$spec))
f3 = per$freq[m3ind]

(cbind(f1,f2,f3))
```

```
##              f1         f2         f3
## [1,] 0.08333333 0.02333333 0.01666667
```

The frequencies pulled out of the periodogram data are .08333, .02333, and .01667 (listed in order of peak height). These numbers tell us that there is a periodic signal of 1/f1 = 12, 1/f2 = 42.85, and 1/f3 = 60 months in the data. In other words, the data, though noisy, completes a full cycle after this many months (different cycles for each frequency). This is important because it tells us that we will probably need to adjust for seasonality when trying to model this data.

Problem 6



We can tell by the plot of ocean temperatures by month that there is definitely a little bit of seasonality in this data. While temperature seems to stay pretty steady from August through January, it is pretty clearthat ocean temperatures rise significantly starting in February, peak is March and April, and then decline again to a similar level as the later months by July. We can also see that there is a lot less variability in the data when the ocean temperatures are high (February-June) and that the variability picks up a lot in the later months. This may be because overall ocean temperatures have gotten hotter over the years, so we see a lot more change in temperature when the ocean is supposed to be cold and is actually warm (Sept - Dec), than we do when it is warm when we expect it to be.

Problem 7

```
dum = factor(cycle(d))
mm = model.matrix(d~0+dum)
fit = lm(d~0+mm)
```

Regression of Nino on Monthly dummies

	Ocean Temp
January	26.723*** (0.125)

February	27.158*** (0.125)
March	27.676*** (0.125)
April	27.690*** (0.125)
May	27.533*** (0.125)
June	27.084*** (0.125)
July	26.713*** (0.125)
August	26.537*** (0.125)
September	26.592*** (0.125)
October	26.508*** (0.127)
November	26.522*** (0.127)
December	26.552*** (0.127)
Observations	597
R^2	0.999
Adjusted R^2	0.999
Residual Std. Error	0.887 (df = 585)
F Statistic	45,951.290*** (df = 12; 585)
Notes:	***Significant at the 1 percent level.
	**Significant at the 5 percent level.
	*Significant at the 10 percent level.

Fitting the data to a matrix of monthly dummy indicators shows us that there is definitely an association between ocean temperature and the month of the year. In fact, it appears that our monthly dummies explain 99% of the variability in ocean temperature (looking at the Rsquared). This will probably be a useful result to use when modeling the data down the line.

Problem 8

I started off this problem by only fitting the AR component of the arima model until I found one that worked best. An AR(3) was a pretty good step up from one and two. Once I created an AR(4), however, I realized that the AIC barely decreased and that the fourth AR term wasn't significant, so I stuck with an AR(3)

```
arima(d, order = c(3,0,0))
```

```
##
## Call:
## arima(x = d, order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3  intercept
##       1.1734  -0.1395  -0.1860    26.9212
## s.e.  0.0406   0.0630   0.0407     0.1034
##
## sigma^2 estimated as 0.1486:  log likelihood = -279.05,  aic = 566.1
```

```
arima(d, order = c(4,0,0))
```

```
##
## Call:
## arima(x = d, order = c(4, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3     ar4  intercept
##       1.1870  -0.1306  -0.2672  0.0695    26.9185
## s.e.  0.0413   0.0631   0.0633  0.0415     0.1109
##
## sigma^2 estimated as 0.1479:  log likelihood = -277.65,  aic = 565.3
```

Next, I started testing how many MA terms to add. The ARMA(3,2) was a big step up from ARMA(3,1), but ARMA(3,3) didn't add much and, again, didn't have a significant coefficient, so I went with ARMA(3,2):

```
arima(d, order = c(3,0,2)) #Moving on with this
```

```
##
## Call:
## arima(x = d, order = c(3, 0, 2))
##
## Coefficients:
##          ar1     ar2      ar3     ma1      ma2  intercept
##       0.5129  0.8174  -0.5532  0.6979  -0.1752    26.9199
## s.e.  0.0804  0.0823   0.0780  0.0943   0.0989     0.1064
##
## sigma^2 estimated as 0.1458:  log likelihood = -273.46,  aic = 558.92
```

```
arima(d, order = c(3,0,3))
```

```
##
## Call:
## arima(x = d, order = c(3, 0, 3))
##
## Coefficients:
##          ar1     ar2      ar3     ma1     ma2     ma3  intercept
##       0.3241  0.8545  -0.4638  0.8881  0.0358  0.1292    26.9170
## s.e.  0.1338  0.0262   0.1150  0.1349  0.1935  0.0746     0.1116
##
## sigma^2 estimated as 0.1447:  log likelihood = -271.58,  aic = 557.16
```

Next I looked at the integrated term. Usually we don't set a value of d greater than one, but I also tried using two to see if it would add any more information. I found that adding this term actually decreased the significance of some of my estimates and increased the AIC of the model, so I decided not to include an integrated term at all.

Adding monthly dummies as an external regressor probably provided the biggest boost to the usefulness of this model.

```
arima(d, order = c(3,0,2), xreg = mm,include.mean = F)
```

```
##
## Call:
## arima(x = d, order = c(3, 0, 2), xreg = mm, include.mean = F)
##
## Coefficients:
##          ar1     ar2      ar3     ma1      ma2     dum1     dum2     dum3
##       0.8798  0.9658  -0.8936  0.1130  -0.8481  26.7332  27.1670  27.6868
## s.e.  0.0337  0.0190   0.0303  0.0508   0.0515   0.0870   0.0871   0.0871
##          dum4     dum5     dum6     dum7     dum8     dum9    dum10
##       27.6975  27.5411  27.0882  26.7173  26.5346  26.5892  26.4822
## s.e.   0.0872   0.0872   0.0872   0.0872   0.0871   0.0870   0.0871
##         dum11    dum12
##       26.4981  26.5236
## s.e.   0.0871   0.0871
##
## sigma^2 estimated as 0.09726:  log likelihood = -152.88,  aic = 339.77
```

All of the monthly dummies are significant along with the ARMA terms, and the AIC dropped from 559 to 340. This tells us that the specifications of this model are much better than the previous ones that we have looked at.

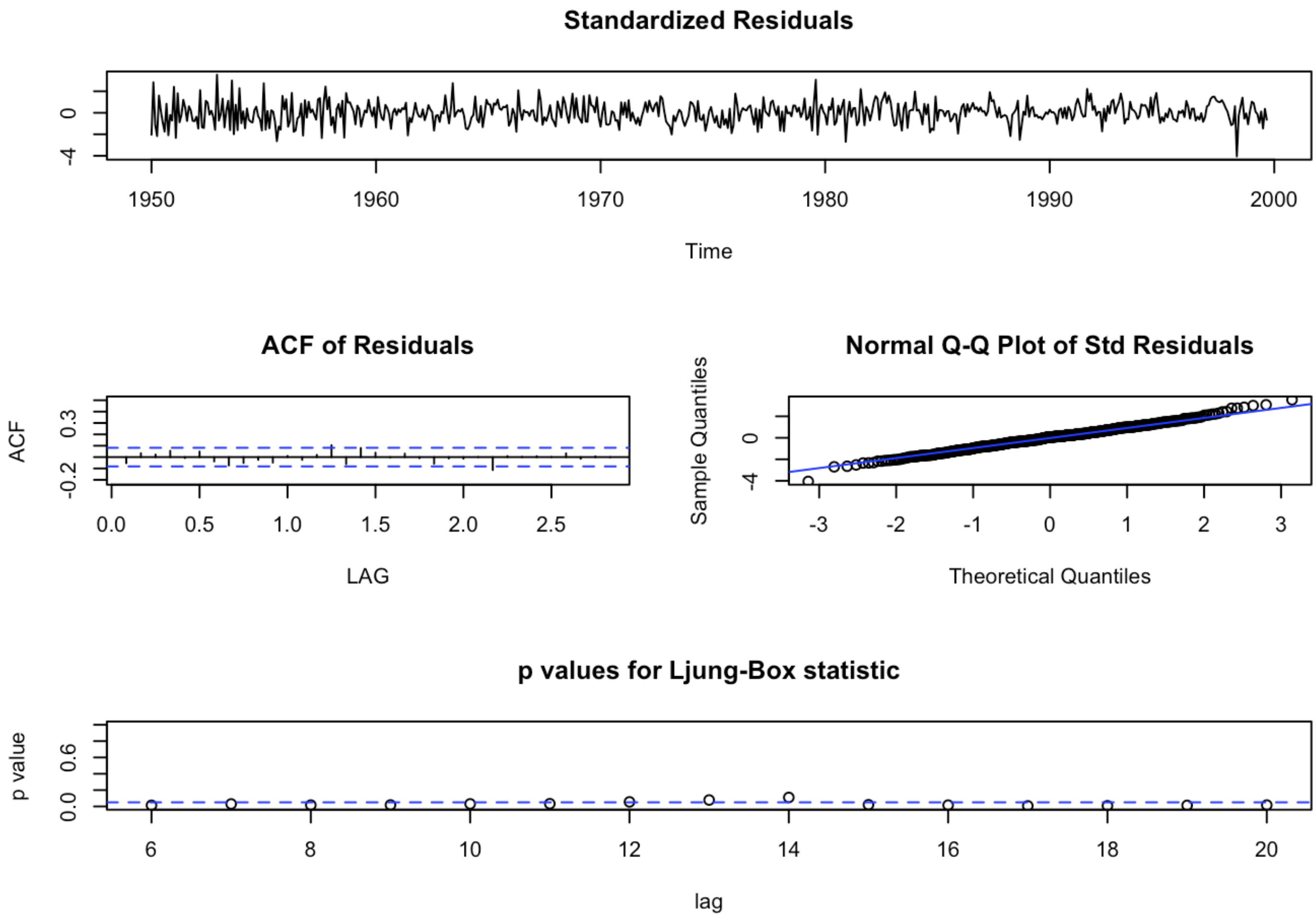
Finally, I included seasonal terms in my arima model. Basically every combination of seasonal terms that I tried made the model worse. The coefficients on the terms were never significant and the AIC always increased. The most helpful specification I could add was a seasonal AR(1), but even this increased the AIC a little bit and still wasn't significant.

```
arima(d, order = c(3,0,2), seasonal = list(order = c(1,0,0), period=12), xreg = m
m,include.mean = F)
```

```
##
## Call:
## arima(x = d, order = c(3, 0, 2), seasonal = list(order = c(1, 0, 0), period = 1
2),
##     xreg = mm, include.mean = F)
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2    sar1     dum1     dum2
##       0.9712  0.7827  -0.8041  -0.0053  -0.7096  0.0296  26.7301  27.1677
## s.e.  0.0839  0.1475   0.0770   0.1017   0.1021  0.0452   0.0889   0.0890
##          dum3     dum4     dum5     dum6     dum7     dum8     dum9
##       27.6844  27.6978  27.5383  27.0888  26.7139  26.5346  26.5854
## s.e.   0.0890   0.0891   0.0891   0.0891   0.0891   0.0890   0.0889
##         dum10    dum11    dum12
##       26.4823  26.4939  26.5238
## s.e.   0.0890   0.0890   0.0890
##
## sigma^2 estimated as 0.09701:  log likelihood = -151.94,  aic = 339.88
```

Based on these results, it appears that an ARIMA(3,0,2) with seasonal dummies as external regressors is the best model for this data. To test this, we can look at some of the output from the sarima() command:

```
sar = sarima(d, 3, 0, 2, 0, 0, 0, 0, xreg = mm[, -1], details = FALSE)
```



The top plot of the residuals looks good, they appeared to be centered at zero and without a lot of heteroskedasticity (although there is some in the periods right before 1990 and 2000). The ACF plot also looks good, there only appear to be two small lags at which the residuals are correlated but even these are borderline, so we know that the errors aren't correlated with each other for the most part. The normal Q-Q plot also looks great, the fact that almost every point lies on the straight line indicates that our assumption about the normal distribution of the errors is probably correct, save for a few outliers at the tails. The Ljung-Box plot is a little bit worrisome, we can see from the points below the dotted blue line that at all the lags except 12, 13, and 14, we may have residuals that are not independent. While this may be an issue, I was unable to get anything significantly different from changing the model (I tried taking out the dummies, adding different types of seasonality, and changing the ARMA specifications). Therefore, it may be an issue that we have to deal with and keep in mind when forecasting using this model.