Project NFP - Final Report

Sam Choi, Eric Xu

4/17/2018

Introduction

The Non Farm Payroll (NFP) Report is a monthly government report tracking the number of payrolls in the U.S., excluding farm workers, private household employees, and non-profit organization employees. It is collected and reported by the the U.S. Bureau of Labor Statistics (BLS), and is an influential economic indicator used to analyze the state of the labor market.

The release of monthly NFP Reports has significant ramifications for economic policy makers and market participants. Thus, it is imperative for both economists and traders to have a general forecast of the next month's NFP Report in order to prepare for the volatility that ensues with the release of the report.

The monthly data is collected by the BLS as a component of the Current Employment Statistics (CES) survey. About a third of all NFP workers are surveyed, and the value reported is the result of additional extrapolation and adjustment.

The objective of this analysis is twofold: To determine the extent to which seasonal and non-seasonal trends can be captured using techniques in univariate time series analysis, and to accurately forecast future NFP Reports.

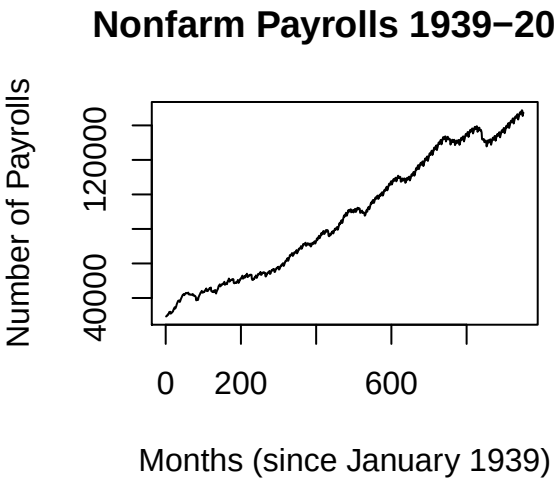
Data Analysis

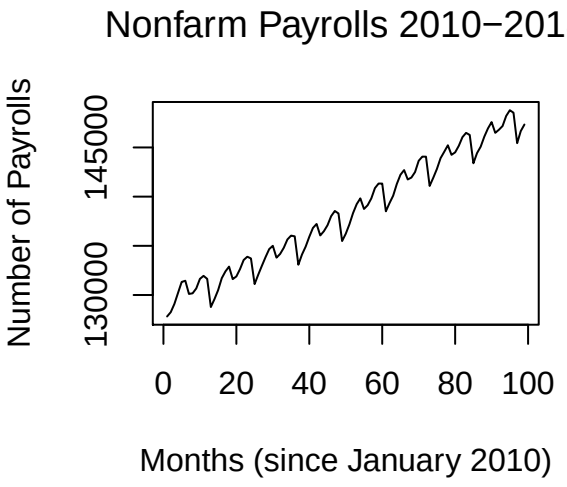
Exploratory Data Analysis

We aim to explore PAYNSA, PAYEMS a dataset of non farm payrolls without seasonal adjustments. The data is recorded in thousands of persons, and contains monthly NFP report values from January 1939 to March 2018.

Throughout the 80 years represented in these datasets, various events have occurred that significantly changed the conditions of the economy. For this reason, we will limit our analysis to the years that followed the financial crisis of 2007/2008. By narrowing our scope to 2010-2018, we aim to provide a more telling analysis of the trends associated with the post-recession economic recovery.

Not Seasonally Adjusted (NSA)

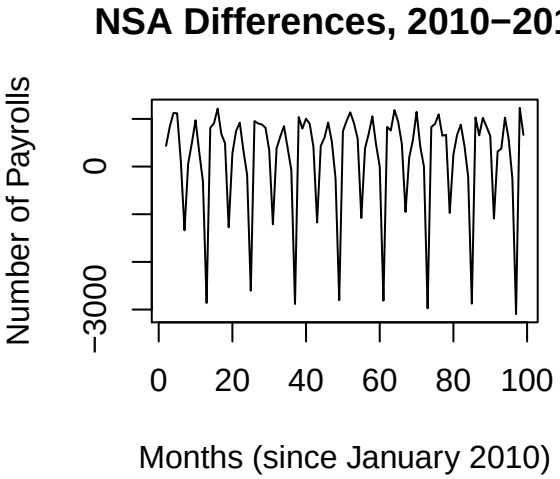


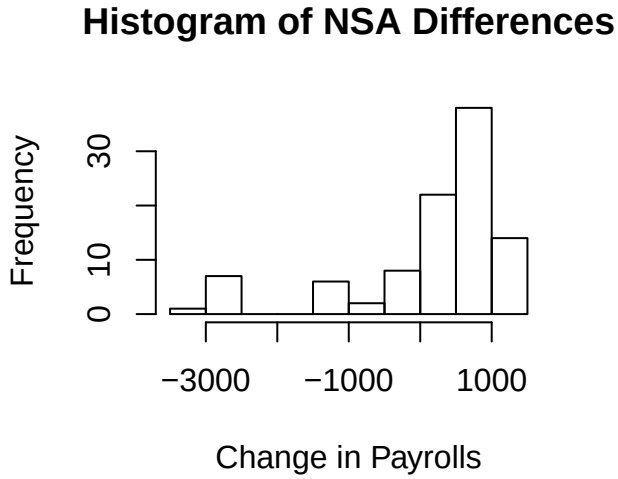


Change in Payrolls

First order differencing of SA and NSA data

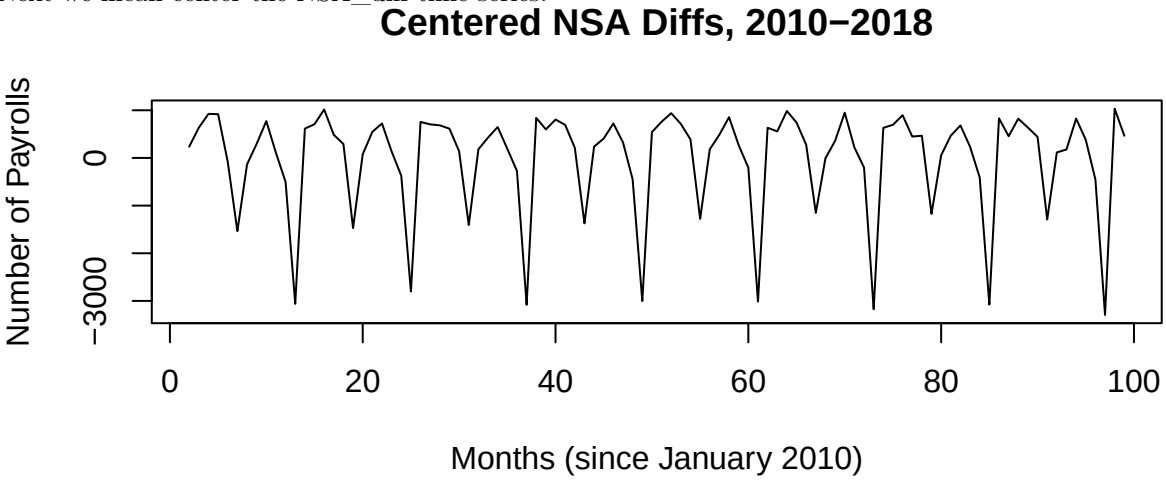
NSA Differences 2010-2018





Taking the first difference seems to have detrended the original time series, and plotting the histogram indicates that the change in total payrolls may follow a left-skewed distribution, characterized by the long left tail above.

Next we mean-center the NSA_diff time series:



The resulting time series above shows signs of stationarity: The plot appears to have a consistent mean and variance with respect to time.

ARIMA Model Building

We will use the Box-Jenkins method to build an ARIMA model for the non-seasonally adjusted (NSA) NFP data. We begin by transforming the data set to achieve stationarity.

Transforming the dataset to achieve stationarity

We begin by transforming the data set to achieve stationarity. We do so by first using a lagged difference with a period of 12 months, producing a time series that resembles a random walk. We then apply differencing again to achieve stationarity.

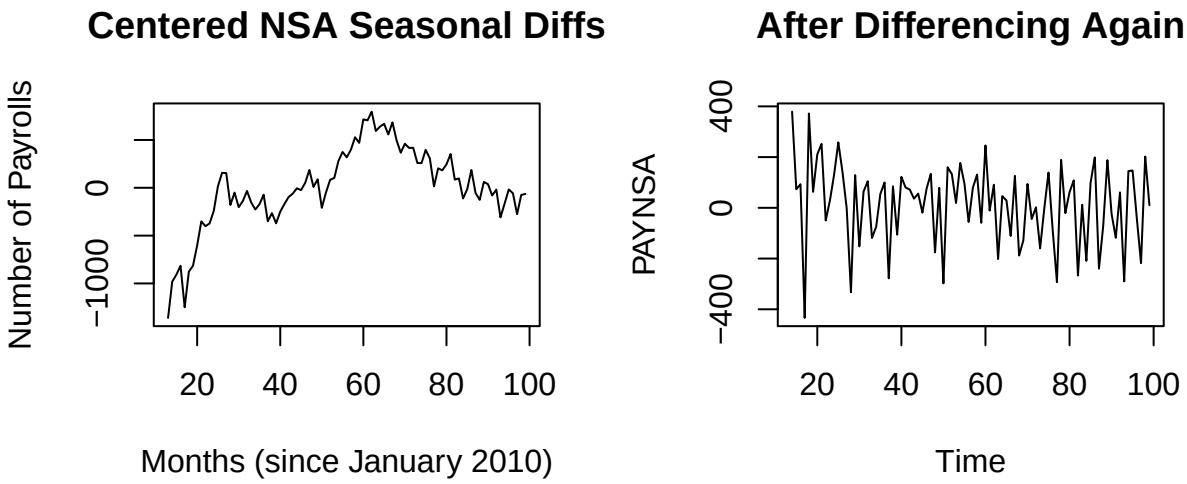


Figure #. NSA data after lagged difference (left) and after lagged difference plus differencing again (right)

This now resembles white noise with fairly consistent variance, with slightly higher variance in the first 20 months of the data.

Model selection

We now inspect the ACF and PACF plots of the transformed time series.

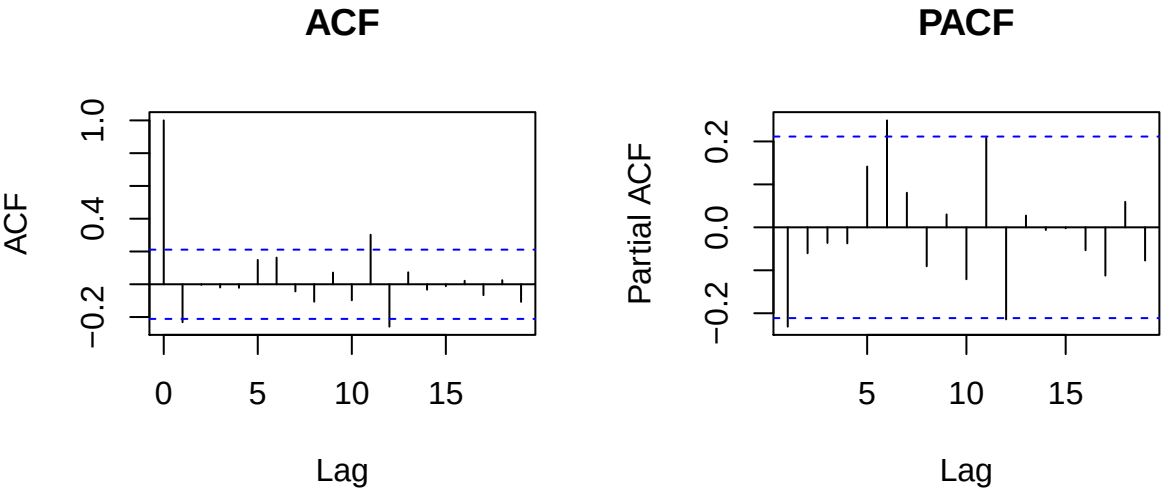


Figure #. ACF and PACF plots of the transformed time series

Neither the ACF nor PACF appear to be significant at any lag. This further supports the notion that our transformed time series is very close to white noise, or an ARMA(0, 0) model. Therefore, fitting any non-zero order ARMA model and proceeding with parameter estimation will likely produce a model with poor predictive properties including bias. Therefore, instead of fitting an ARMA model we turn to spectral analysis to analyze the seasonal variation of our time series from a frequency perspective.

Spectral Analysis

Analyzing the time series using spectral analysis allows us to identify the key frequencies of variation in the non-seasonally adjusted NFP time series.

We begin by transforming the dataset into a stationary time series. We will compare the estimated spectral densities for two methods of transformation: differencing and curve-fitting. We will refer to the time series transformed by curve-fitting as the detrended time series.

Nonparametric Spectral Estimation

We begin by generating raw periodograms for both the differenced and detrended time series.

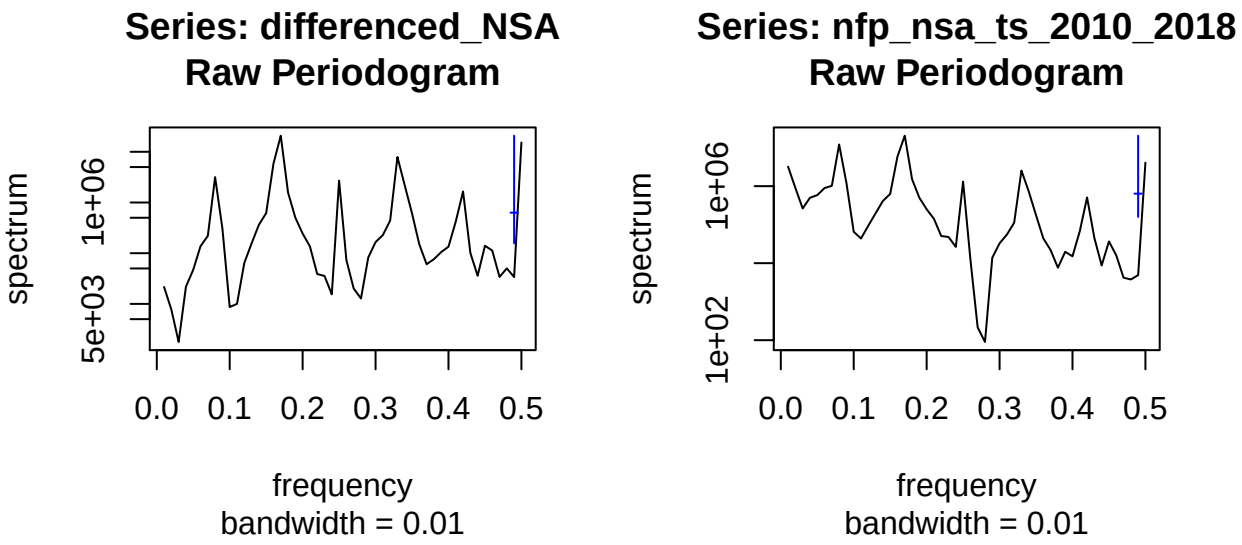


Figure #. Differenced and detrended raw periodograms

As expected, both periodograms are bumpy with similar peaks, but the differenced time series produces a raw periodogram with a much lower baseline in the 0.0 to 0.1 frequency range and the detrended time series produces a raw periodogram with a lower baseline in the 0.26 to 0.28 frequency range. This means in the differenced time series, the peak at a frequency of about 0.08 is more significant, and in the detrended time series, the peaks at 0.25 and 0.33 are more significant.

To reduce the variance of the periodogram and better identify significant peaks, we will apply smoothing to the periodogram using various parameters to find a better estimate of the spectral density. We begin with a daniell kernel with m = 1.

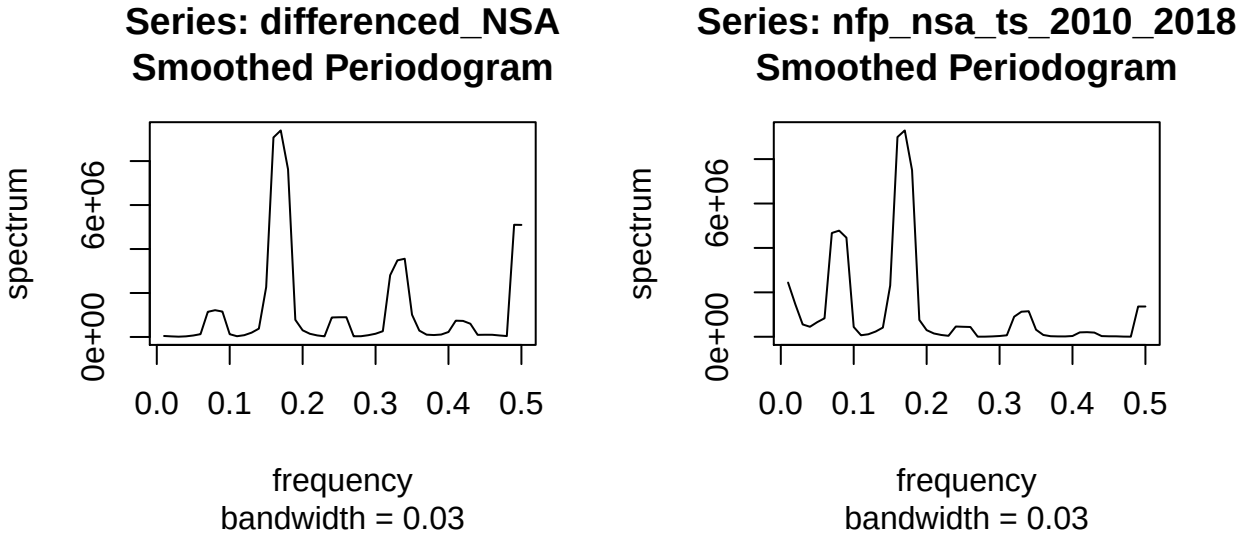


Figure #. Smoothed periodograms of differenced and detrended time series using a Daniell kernel (m = 1)

This looks quite smooth already. Using m = 2 results in wider peaks shown in the Appendix, possibly introducing bias. Therefore, setting m = 1 is an adequate amount of smoothing that allows us to identify significant peaks while minimizing bias.

To further reduce bias and narrow the peaks, we can apply smoothing with the modified Daniell kernel with m = 1. The periodogram is very smooth, so tapering has little effect. This suggests our data set has very clean and predictable cyclical variation, and we can now find the frequencies of the most significant peaks. We set the taper to 0.1 and overlay the local maxima to show where key frequencies might be.

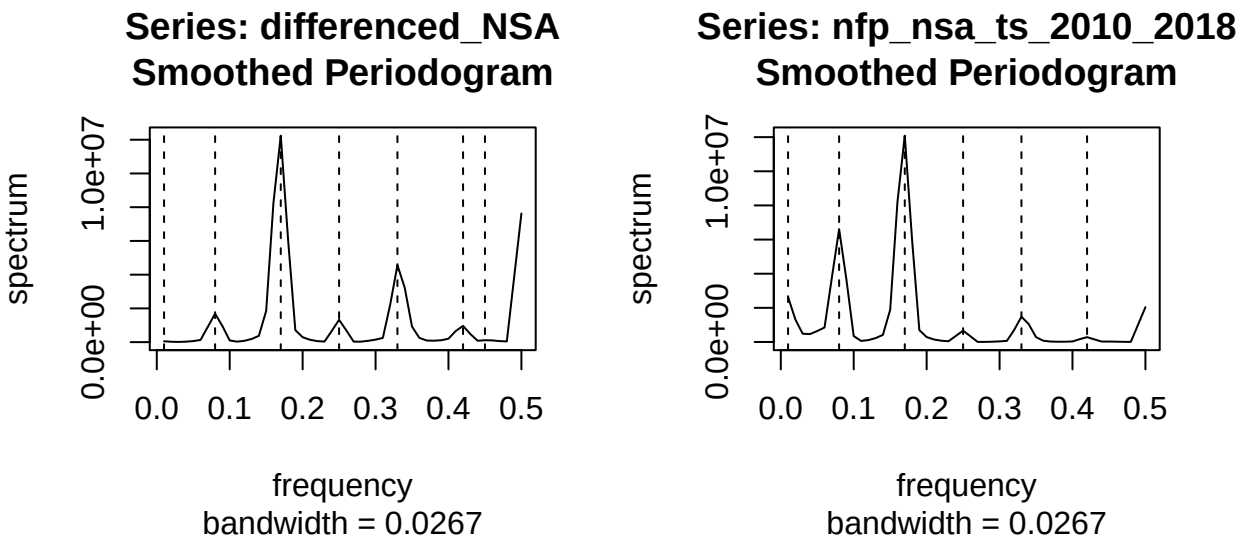


Figure #. Smoothed and tapered periodograms overlaid with local maxima.

Smoothing using the modified Daniell kernel produces much sharper peaks. There does not appear to be much bumpiness remaining so tapering should not be necessary. To see the smoothed and tapered periodogram, see Appendix.

In the differenced time series, we can identify three major peaks: one at a frequency of about 0.17, corresponding to a semiannual cycle, one at a frequency of about 0.5, corresponding to a two-month cycle, and one at a frequency of about 0.33, corresponding to a quarterly cycle.

The smoothed periodogram of the detrended data has two major peaks: one at a frequency of about 0.08, corresponding to an annual cycle, the largest at a frequency of about 0.17, corresponding to a semiannual cycle, and one at a frequency of 0.08, corresponding to an annual cycle. Two other, less significant peaks appear, one at 0.01 and one at 0.5.

Parametric Spectral Estimation

We can now compare our smoothed and tapered periodograms with the estimated spectral density generating using parametric spectral estimation.

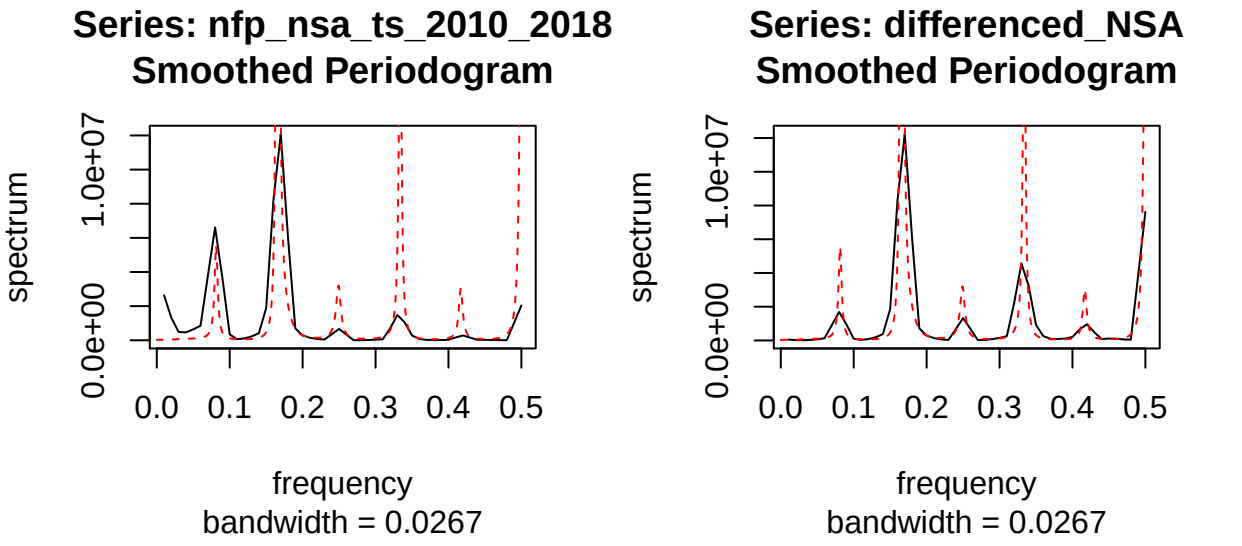


Figure #. Smoothed and tapered periodograms overlaid with parametric spectral estimator.

The parametric spectral estimator has peaks at frequencies 0.08, 0.17, 0.25, 0.33, 0.42, and 0.50. This matches the frequencies of peaks seen in the differenced and detrended time series. However, the size of the peaks at these frequencies differs between the two time series.

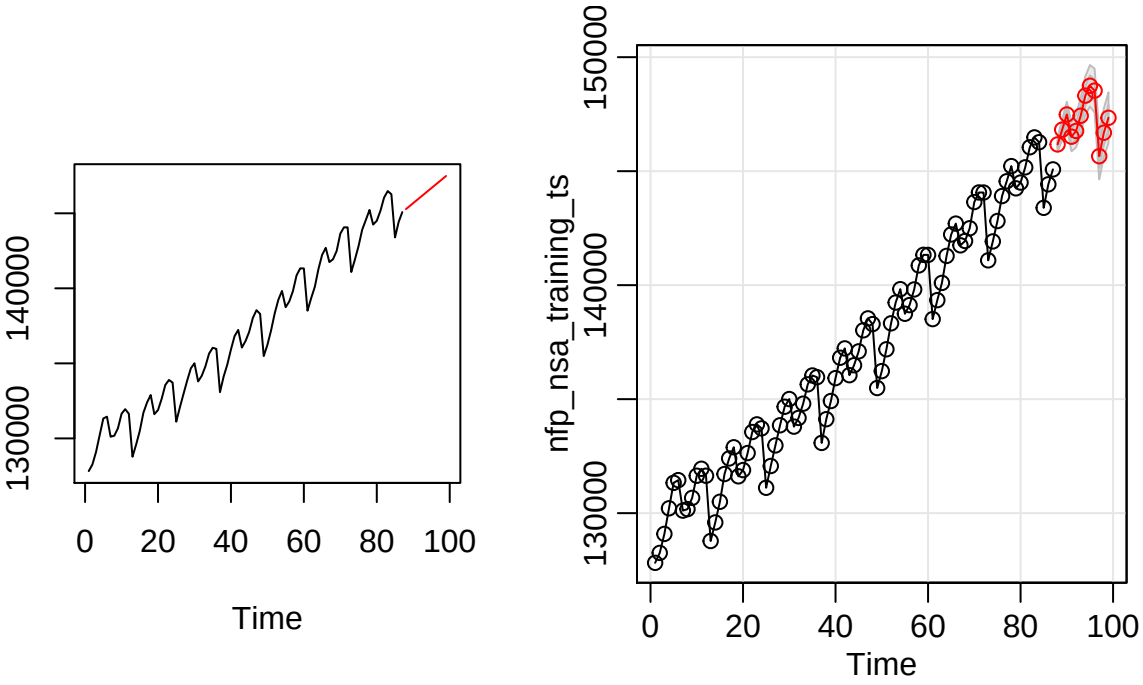
The peaks at 0.08, 0.17 match the detrended time series periodogram in frequency and size, with much less significant peaks at the other frequencies. The differenced time series has its three major peaks at 0.17, 0.33, and 0.50, with much less significant peaks at the other frequencies.

Forecasting

Using data from January 2010 to March 2017, we aim to forecast Nonfarm payrolls for the following 12 months (April 2017 to March 2018). We do this by partitioning our dataset into a training portion that will be used to fit our models, and a testing portion that will be used to measure the accuracy of our forecasts.

ARIMA

We proceed with a non-seasonal ARIMA model, as well as a SARIMA model to fit the general trend of the data, as well as the seasonal trends observed:



We can see that the first graph captures the general direction of increasing payrolls, while the second graph also captures the seasonal behavior of payroll increases and decreases.

We follow up by computing the mean-squared error for both models, and find that the ARIMA forecasts result in a MSE of 1774426, while the SARIMA forecast results in a MSE of 13108. This shows that the SARIMA forecast has not only captured the seasonal trend, but has led to a significantly lower error term (~130 times lower MSE).

Spectral

Conclusion

Key Takeaways

Jobs are added and lost according to a highly consistent seasonal pattern. TODO

Linear job growth since the Great Recession ended. TODO

Interesting Findings

Cycles of variation every quarter, 6 months, year align with the seasons. TODO

Seasonal and first differencing removes cyclical and trend variation, leaving us with essentially white noise. TODO

Different key frequencies highlighted by differencing vs detrending. TODO

Future Analysis

Analyzing other time periods. TODO

Appendix

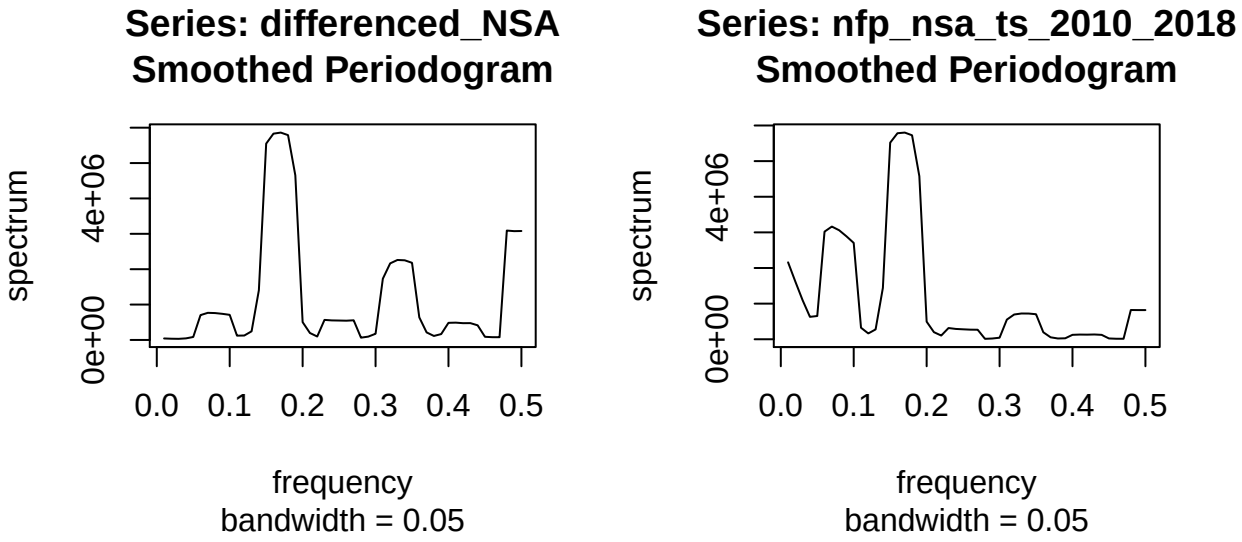


Figure #. Smoothed periodograms of differenced and detrended time series using a Daniell kernel (m = 2)

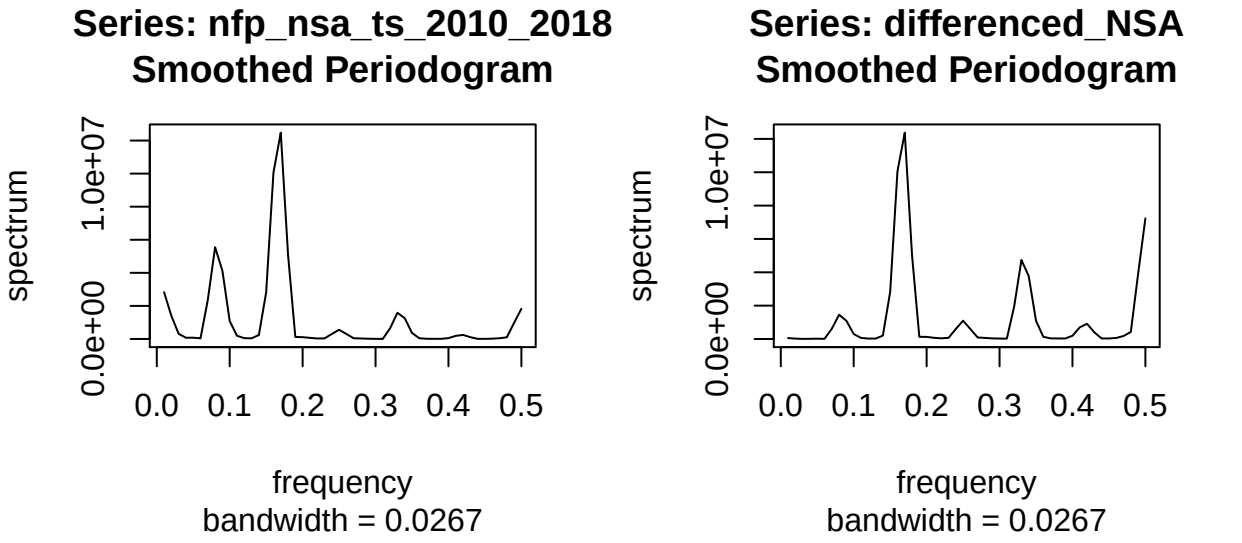


Figure #. Smoothed and tapered periodograms of differenced and detrended time series using a modified Daniell kernel (m = 1)

Code

EDA

```
PAYNSA <- read.csv(file = "PAYNSA.csv", header = TRUE, sep = ",")
nfp_nsa_ts <- ts(PAYNSA[2])
plot.ts(nfp_nsa_ts, main = "Total Nonfarm Payrolls (Not Seasonally Adjusted)", xlab = "Months (since January 1

nfp_nsa_ts_2010_2018 <- ts(PAYNSA[853:951, ][2])
plot.ts(nfp_nsa_ts_2010_2018, main = "Total Nonfarm Payrolls (NSA), 2010-2018", xlab = "Months (since January

NSA_diff <- diff(nfp_nsa_ts_2010_2018, lag = 1, differences = 1)
plot.ts(NSA_diff, main = "NSA Differences, 2010-2018", xlab = "Months (since January 2010)", ylab = "Number of

hist(as.numeric(unlist(NSA_diff)), breaks=15,  main = "Histogram of NSA Payroll changes", xlab = "Number of Pa

NSA_mean <- mean(NSA_diff, na.rm = TRUE)
NSA_mean
```

```
centered_NSA_diff <- NSA_diff - SA_mean
plot.ts(centered_NSA_diff, main = "Centered NSA Diffs, 2010-2018", xlab = "Months (since January 2010)", ylab =
```

Forecasting

```
PAYNSA <- read.csv(file = "PAYNSA.csv", header = TRUE, sep = ",")
nfp_nsa_ts <- ts(PAYNSA[2])
nfp_nsa_training_ts <- ts(PAYNSA[853:939,][2])
nfp_nsa_testing_ts <- ts(PAYNSA[940:951,][2])

nobs <- length(nfp_nsa_training_ts)
fit <- arima(nfp_nsa_training_ts, order=c(0, 1, 0), xreg=1:nobs)
n_pred <- length(nfp_nsa_testing_ts)
forecast <- predict(fit, n_predictions, newxreg=(nobs+1):(nobs+n_pred))

mse <- function(ts1, ts2) {
  sum_squares = 0
  for (i in 1:length(ts1)) {
    diff <- ts1[i] - ts2[i]
    sum_squares = sum_squares + diff^2
  }
  sum_squares/length(ts1)
}

par(mfrow=c(2,1))
ts.plot(nfp_nsa_training_ts, forecast$pred)
sfit <- sarima.for(nfp_nsa_training_ts, n_pred, p=0, d=1, q=0, P=0, D=1, Q=0, S=12)

mse(forecast$pred, nfp_nsa_testing_ts)
mse(sfit$pred, nfp_nsa_testing_ts)
```