Time Series Analysis: Intensive Forecasting In R

Stat 133 Post #1, Fall 2017

Edward Yang

11/20/2017

I. Introduction

A Notable Connection: Randomness and Inference

Studying statistics often involves the understanding of uncertainty and how we, as researchers, can interpret stochastic models. In my first post, I explained how R's many built-in functions can be used to analyze data involving a measure of time (time series analysis); however, we only briefly discussed certain topics, such as confidence intervals, that are **able** to be applied in such forecasting models. Thus, in this second post, I will be discussing how we can closely study our datasets to actually **compute** and carry out data manipulation and visualization that, in turn, allow us to form appropriate statistical hypotheses for such datasets that vary over time. We will be using a variety of functions from different packages such as `ggfortify` or `forecast`, differentiating it from the first post that mainly focused on R's built-in functions (with the HoltWinters prediction model being the only exception).

If you are interested in reading my first post, you can find it on our class's Github repository, `ucb-stat133`. From there, you can find the `stat133-posts-fall17` directory, and locate the `lab102` folder (as I am in that lab). You should be able to find my post, titled `post01-edward-yang.pdf`. I chose not to link the source, as I will only be linking appropriate references and our professor has the ability to change the format of the repository.

Note: I will not assume that you have read my first post, as the two posts will differ in approaches and content. Thus, you do not have to read the first post if you do not wish to, but I suggest you to take a look at it (for your own learning purposes).

Many students have learned about confidence intervals in introductory statistics classes, such as AP Statistics in high school or a lower-division statistics class in a university, whereas some students have never taken such a class. To refresh our understanding, a confidence interval is a set of numbers that estimate the value of a population parameter with a specified level of "confidence" through the use of a randomly sampled dataset. Furthermore, we are able to compute deviations from datasetsusing our intuition of "randomness" to make careful generalizations and assumptions that allow us to form inference using basic ideas from hypothesis testing.

Motivation

As a Statistics and Economics major, I am extremely interested in the study of random variables and degrees of entropy in certain data structures. With these random variables, we can compute the probabilities for certain outcomes to occur. Furthermore, I plan on taking Economics 141 (Multivariable Econometrics) and Statistics 150 (Stochastic Processes) in the near future, so I think that it is important to tie the subject of uncertainty with real-life data and make practical conclusions based on our findings with hypothesis testing and confidence intervals. In addition, I will be taking Statistics 153 (Time Series) in a later semester, so this allows me to study graphs involving a time element.

In this post, we will be discussing and tackling the following concepts:

1. Learning how to use `autoplot()` from the `ggfortify` extension to create plots.
2. Displaying various methods to graph and interpret plots with a certain confidence level.
3. Using regression with ARIMA (with `auto.arima()`) to forecast future expectations of plots.
4. Using regression with ETS (with `ets()`) to forecast future expectations of plots.
5. Observing error in certain regression models and distinctions between forecasting methods.
6. Measuring seasonal components of datasets using `monthplot()`.

II. Main Example: Extension of U.S. Unemployment Rates

Preparing the Data

In the last post, we focused on the relationship between inflation rates and unemployment rates in the United States from the years 1948 to 2017 and briefly mentioned the HoltWinters prediction model; however, we did not capitalize on the concept of regression modeling. For that reason, we will be extending our analysis by isolating the two variables and just studying the trend of U.S. monthly unemployment rates from January 1948 to October 2017. All of the data is publicly available online on FRED (Federal Reserve Economic Data). Below is a link to the monthly unemployment rates listed on the FRED website that you can easily download your own CSV file from (Reference 1).

Reference 1: FRED Civilian Unemployment Rate - https://fred.stlouisfed.org/series/UNRATE

Important Note: These datasets (of the URL listed above) update over time. Thus, to make sure you are using the same dataset, specify the date to be 1948-01-01 to 2017-10-01 on the graph above. Then, click 'Download' on the top-right corner to save your own CSV file of the data. You cannot simply use `download.file()` as the online data set is not a CSV file itself at first (it is a graph which you can choose to download a .csv file from).

We will read in the .csv file and convert it to a times series object with the function `ts()`. The function `ts()` is similar to the function `data.frame()` because it asserts an object to be of a certain type.

```r
# reading in the saved .csv file
# your working directory and save location may be different
unrate_raw <- read.csv('../data/unrate.csv')

# creating a time series object with the data
# specify freq = 12 months per year
unrate <- ts(unrate_raw$UNRATE, start = c(1948, 1), freq = 12)

# showing it is a time series object
str(unrate)
```

```
##  Time-Series [1:838] from 1948 to 2018: 3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
```

Plotting with `autoplot()`

In our dataset, we can use the `autoplot()` function from the `ggfortify` package extension (that requires the `ggplot2` package) to graph the time series objects (objects with class 'ts'). The function does not support certain objects, such as data frame objects. You can alternatively use the function `ts.plot()` if you would like to plot using R's built-in functions. You can read more on the function `autoplot()` and how to work with its arguments in the link below (Reference 2).

Reference 2: Plotting Time Series Functions with Autoplot - https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_ts.html

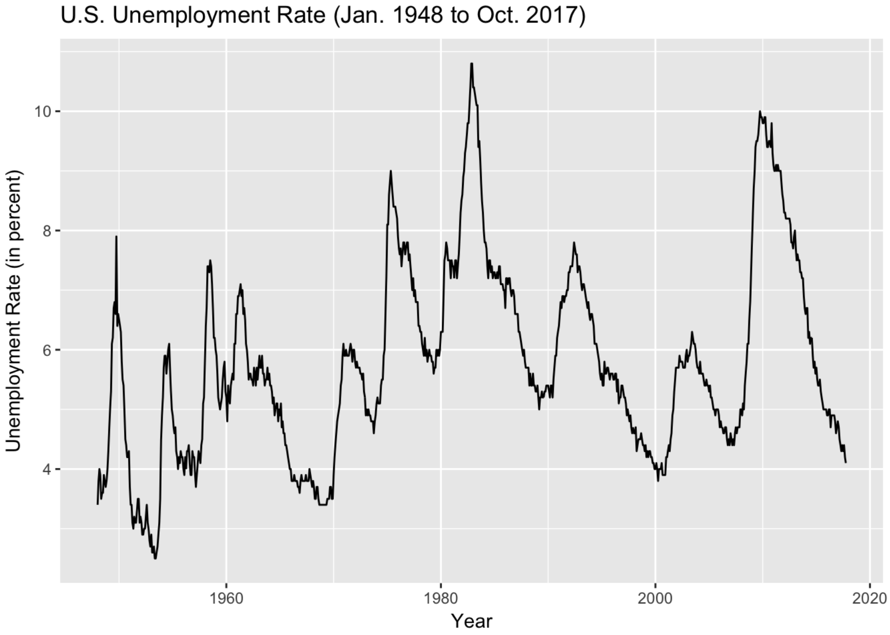
```
# install.packages('ggplot2')
# install.packages('ggfortify')
# ignore the warning message at first when replacing names with library()
library(ggfortify)
```

```
## Loading required package: ggplot2
```

```
## Warning: namespace 'DBI' is not available and has been replaced
## by .GlobalEnv when processing object 'silent'

## Warning: namespace 'DBI' is not available and has been replaced
## by .GlobalEnv when processing object 'silent'
```

```
# autoplotting the data
autoplot(unrate, xlab = 'Year', ylab = 'Unemployment Rate (in percent)',
         main = 'U.S. Unemployment Rate (Jan. 1948 to Oct. 2017)')
```



Description of graph: When looking at this graph, I first think to myself, "What is the difference between this plot and a regular line plot?" Well, we were able to draw out a nicely formatted line plot (with the background similar to when plotting with `ggplot`) with a time series object and not with data frame or tabular objects. Thus, we can see 'Year' on the horizontal axis directly being plotted in order. As a result, we can see the peaks and troughs of the graphs (such as the peak in the early 1980s and later 2000s), highlighting economic recoveries and recessions. In the last post, we highlighted these sudden changes to see if certain changes are remarkable enough to deem as "statistically significant"; however, for this post, we will focus on predicting future (expected) unemployment rate for the next few years through regression modeling using ARIMA.

This introduction with `autoplot()` is just a nice way to start the post and see how the data looks like as a time series plot (it's quite a handy function). Thus, we shall proceed with our main goal of the post – regression.

III. Regression using ARIMA Models

What is ARIMA?

The ARIMA (Autoregressive Integrated Moving Average) model is a useful technique that computes regression along a moving average with respect to time. We will be using it to forecast (or predict) **future** unemployment rates with a specific level of confidence. This model allows us to appropriately extrapolate data to learn more about the plot's past behavior over the last sixty years. To use the ARIMA model, we will use the function `auto.arima()` from the `forecast` package. `auto.arima()` takes in a time series objects and returns a summary of the data computed with a moving average (called an 'ARIMA' object). Do not worry if you do not understand the technicality behind all of the function, as I think the end result is quite elegant and simple to understand. You can read more about the nitty gritty details below (Reference 3).

Reference 3: Forecasting with ARIMA - http://www.forecastingsolutions.com/arima.html

```
# install.packages('forecast')
library(forecast)
```

```
##
## Attaching package: 'forecast'
## The following object is masked from 'package:ggfortify':
##
##     gglagplot
```

```
# creating the ARIMA object
unrate_arima <- auto.arima(unrate)
unrate_arima
```

```
## Series: unrate
## ARIMA(2,1,2)(2,0,2)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1     sar2     sma1     sma2
##       1.3830  -0.4972  -1.3772  0.6262  0.4679  -0.0880  -0.7002  -0.0166
## s.e.  0.3823   0.3578   0.3488  0.2651  0.3303   0.2157   0.3351   0.2962
##
## sigma^2 estimated as 0.03525:  log likelihood=214.61
## AIC=-411.21   AICc=-411   BIC=-368.65
```

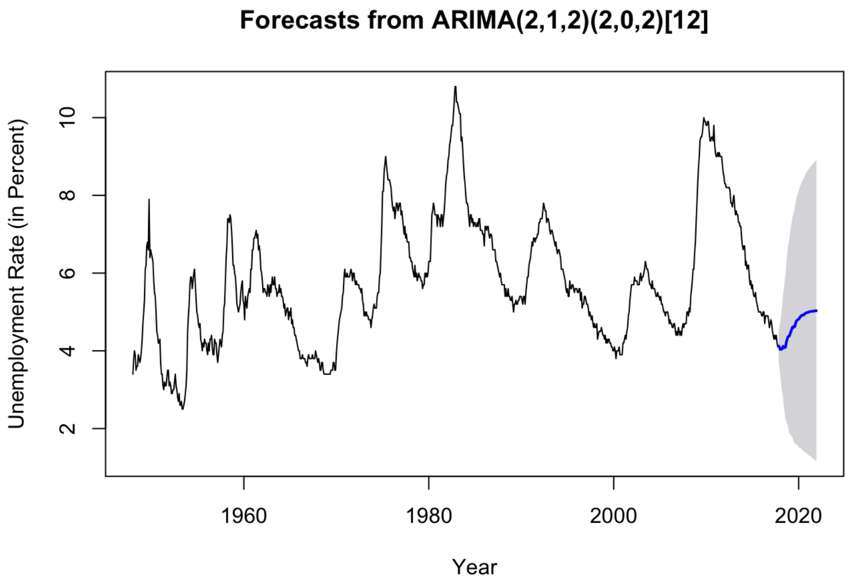
Forecasting using ARIMA

Now, we can plot our predictions of the future using the `forecast()` function (from the `forecast` package) in conjunction with the built-in `plot()` function. We will be specifying optional parameters, level and h, to respectively specify what level of confidence you want to predict at and many more terms (months) to predict. To learn more about what you can forecast in R and how `forecast()` works, feel free to check out the book below (Reference 4).

Reference 4: Forecasting - ARIMA Models - https://www.otexts.org/fpp/8/1

```
# using the forecast function with a 95% confidence level for ARIMA model
# the argument h specifies how many more months to predict (50)
unrate_forecast_arima <- forecast(unrate_arima, level = c(95), h = 50)

# plotting the predictions for the next 50 months (ending with Dec. 2021)
plot(unrate_forecast_arima, xlab = 'Year',
     ylab = 'Unemployment Rate (in Percent)')
```



Description of graph: Based on this graph, we can see that, using an algorithm that computes a moving average, the unemployment rate is expected to increase within the next four years. Intuitively, without using the ARIMA model, we can note that we (end of 2017) are currently in a period of time with great economic status (low unemployment rate). Thus, we are expected to have our unemployment rate bounce back up if history repeats itself, and there is also a zero level bound for unemployment rate so it cannot go down much further. Lastly, we can observe the blue area that is a margin of error that comes from computing the ARIMA model with a specific level of confidence. This tells us that the expected unemployment rate is 95% confident to land anywhere in between this margin within the next given years. Note how the margin of error is “fanning” outwards. This is because it is harder to predict later in the future (say, 2021) versus next year (2018).

Extension: Conclusions and Statistics of ARIMA Models

To study our predictions of our ARIMA model of the unemployment rate, we can use the function `ggtsdiag()` from the `ggfortify` package, which returns a diagonsis of the specified ARIMA object. Then, we can use our knowledge of hypothesis testing to draw practical conclusions from our ARIMA model.

When appropriately using `ggtsdiag()`, three plots will show up:

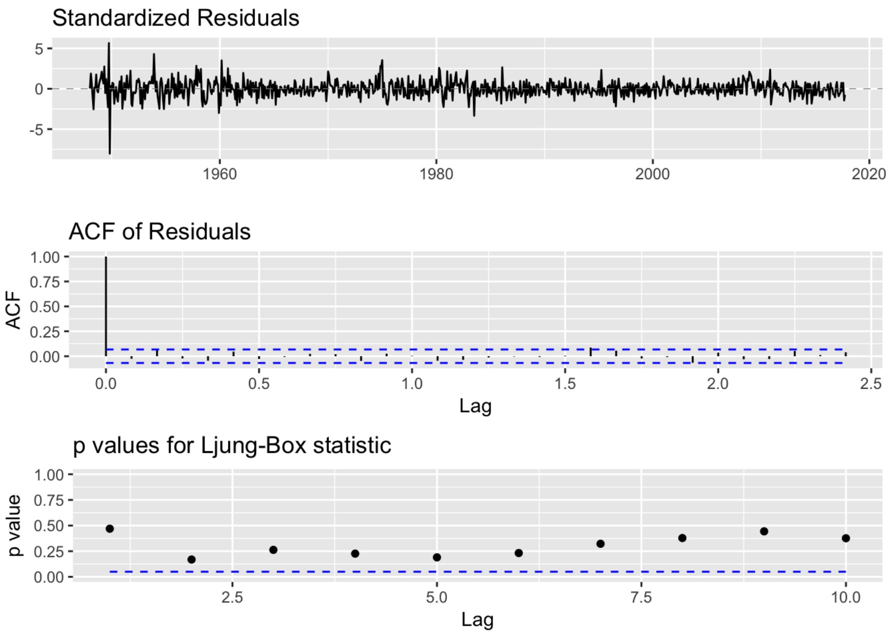
1. **Standardized Residuals**: Displays the standarized residuals of each of the plots and measuring statistically significant patterns that deviate from the moving average.
2. **ACF of Residuals**: Displays autocorrelation of the plots to see how well the future correlates with the past. Autocorrelation is the measurement of correlation given a certain deviation, called lag, that is measured in months. This enables us to create a null hypothesis

(rather, an abstraction) that there is no autocorrelation in the errors of the unemployment rate.

3. **p values of Ljung-Box statistic**: This process is similar to a hypothesis test. Any values higher than the blue-dotted line means that we have no evidence to reject our null hypothesis that there is no autocorrelation in the errors of the unemployment rate. As we can see, the ACF of our residuals matches with this. You can read more about what it means to have large p-values (Reference 5).

Reference 5: High Ljung-Box p-values at large lags - https://stats.stackexchange.com/questions/91706/high-ljung-box-p-values-at-large-lags

```
# by default, cannot specify an axis label for standardized results (time/year)
ggtsdiag(unrate_arima)
```



Standardized Residuals

ACF of Residuals

p values for Ljung-Box statistic

Description of graphs: Based on our standardized residuals plot, we can see that the errors waver pretty consistently in a pattern around zero, which depicts a possible zig-zag pattern in our original plot of unemployment rates (which is true). This is actually backed up by our result with the `forecast()` function, since the ARIMA regression prediction was an increase in the next few years. The autocorrelation of residuals shows that there is no reason to reject the idea that there is no autocorrelation. Thus, there may be autocorrelation, which is also supported by our original plot. Lastly, the p-values for the Ljung-Box statistic is rather difficult to understand, but it follows with our preceding result with the autocorrelation of residuals (large p-values lead us to not reject our original hypothesis). Are there different ways to study more about such observations? Yes; we will go through another forecasting technique called ETS (exponential smoothing) in the next section.

IV. Regression using ETS Models

What is ETS?

ETS modeling is a form of exponential smoothing and forecasting in time series analysis that puts more weight on more recent observations. Unlike ARIMA’s moving average across all dates, ETS relies on a weighted moving average corresponding to a particular parameter (alpha) to state the amount of weight on the more recent seasons than in the past. Reference 6 (below) discusses the reason of using exponential modeling to forecast future expectations of certain parameters, such as unemployment rate. It is a different approach to studying and predicting data through the knowledge of a data’s trend and seasonal factors.

Reference 6: Error, Trend, and Seasonality - http://ellisp.github.io/blog/2016/11/27/ets-friends

Forecasting using ARIMA

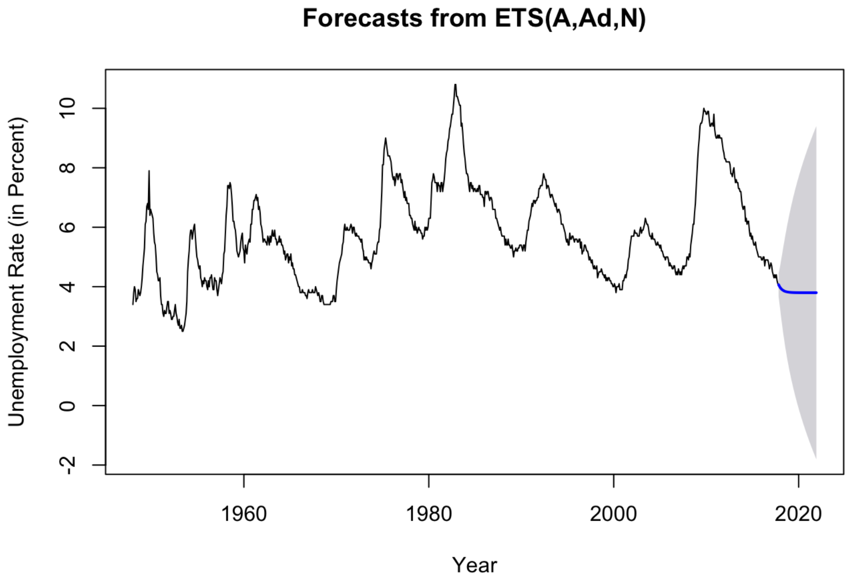
Similar to the ARIMA model, we will be using the `forecast()` function (from the `forecast` package) to predict again along with the same optional parameters (level and r). In contrast to the ARIMA model, however, we will be using the `ets()` function in place of the `auto.arima()` function.

```
# creating the ARIMA object
unrate_ets <- ets(unrate)
unrate_ets
```

```
## ETS(A,Ad,N)
##
## Call:
##  ets(y = unrate)
##
##   Smoothing parameters:
##     alpha = 0.7291
##     beta  = 0.3449
##     phi   = 0.8094
##
##   Initial states:
##     l = 3.2106
##     b = 0.3434
##
##   sigma:  0.1957
##
##      AIC     AICc      BIC
## 2918.843 2918.944 2947.229
```

```
# using the forecast function with 95% confidence for ETS model
# the argument h specifies how many more months to predict (50)
unrate_forecast_ets <- forecast(unrate_ets, level = c(95), h = 50)

# plotting the predictions for the next 50 months (ending with Dec. 2021)
plot(unrate_forecast_ets, xlab = 'Year',
     ylab = 'Unemployment Rate (in Percent)')
```



Description of graph: This result looks different to the result that we had earlier using ARIMA modeling. As with the ARIMA model, I have included a 95% confidence interval with this ETS forecast plot to see the possibility of variability and error using this model. As statisticians, we should try to study why unemployment rate seems to be much more “constant” with this model compared to the ARIMA model. Perhaps it is the recent decrease after the Great Recession (around 2009-2010) that contributed to this expected decline in unemployment rate. Further, using an ETS model puts more weight on these recent observations, so that is also another factor to consider. Now, let us study the Great Recession.

V. The Great Recession (2007 to 2009)

Background Information

As mentioned in the first post, the Great Recession is a period of sharp economic decline that took place in December 2007 and ended in June 2009 (according to Reference 7). Note that this is not the same reference as the one in the first post.

Reference 7: The Great Recession - https://www.investopedia.com/terms/g/great-recession.asp

During this period of decline, how we can measure exactly when the period of decline is first noticeable. As seen in our first post, unemployment rate has a trend, seasonal, and “remainder” component that plays a role in the pattern of consistent up’s and down’s in the original FRED data. Thus, how do we account for this factor and isolate the data so we get a unemployment rate that takes this “seasonal component into account to measure the noticeable effects of the Great Recession?

When does the Great Recession Start?

We will use the built-in R function `monthplot()` to isolate the “seasonal” portion of the time series. Each season corresponds to a unique month. We can these measurements of monthly trends in unemployment and see if there is any sort of variation in unemployment rate between months (Reference 8).

Reference 8: How to Replicate A Monthly Cycle Chart in R - https://stackoverflow.com/questions/5826703/how-to-replicate-a-monthly-cycle-chart-in-r

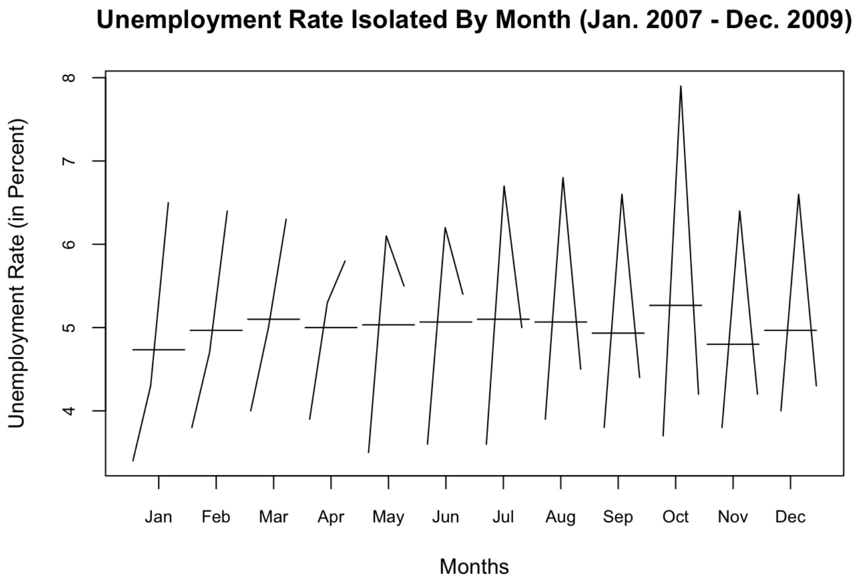
If you would like more information on different types of plots, involving deseasonalization and detrending, check out Chapter 6 of the book below (Reference 9).

Reference 9: Forecasting - Time Series Models - https://www.otexts.org/fpp/6/1

```
# creating time series object only for years 2007 to 2009 (36 months)
unrate_grec <- ts(unrate_raw$UNRATE,
                  start = c(2007, 1), end = c(2009, 12), freq = 12)
unrate_grec
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2007 3.4 3.8 4.0 3.9 3.5 3.6 3.6 3.9 3.8 3.7 3.8 4.0
## 2008 4.3 4.7 5.0 5.3 6.1 6.2 6.7 6.8 6.6 7.9 6.4 6.6
## 2009 6.5 6.4 6.3 5.8 5.5 5.4 5.0 4.5 4.4 4.2 4.2 4.3
```

```
# plotting the monthly decomposition of the time series object
monthplot(unrate_grec, labels = month.abb, cex.axis = 0.8,
          xlab = 'Months', ylab = 'Unemployment Rate (in Percent)',
          main = 'Unemployment Rate Isolated By Month (Jan. 2007 - Dec. 2009)')
```



Description of graph: Based on this plot, we can see that each of the monthly unemployment rates have a gradual change in shape over time. Within each month contains three miniscule points (for each year from 2007 to 2009 in order) plotted and linked together to form a lineplot to create the corner points. We can see a noticeable change, as we see that the graph changes shape over time. From this, how do we see the start and end of the Great Recession from this graph? We can see that the graph first starts low (around 4 percent) but begins to rise sharply (as an overall average to above 5 to 6 percent) around December 2007 to January/February 2008. Also, we can see that the Great Recession ended around April 2009 to June 2009 by the noticeable change in shape between each of the months (they become upside-down 'V' shapes).

This matches Reference 7's dates of the Great Recession as well, but we know that all of our conclusions are visually subjective. Much of statistics is based on our own judgments and interpretations of datasets, but we can construct certain models such as the ARIMA model to estimate with confidence.

VI. Conclusion

Reference 10: Intro. to Forecasting - https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials

Take-Away Message

The tutorial above (Reference 10) is an efficient and comprehensive guide to how we approached most of our ideas in forecasting in R. The point of this post wass to highlight how we can use R and its packages to study specific datasets in-depth through the use of forecasting and data manipulation with respect to time series. We combined the ideas of time series analysis from the first post with new, innovative abstractions using seasonal and exponential modeling. Most importantly, we realized that our intuition is extremely important for making final conclusions about our data. We do not work with certainty in models; rather, we work to approach a level of confidence that we are satisfied with.

Summary

Firstly, we learned how to plot time series objects with the `autoplot()` function (instead of the `ts.plot()` function) through the `ggfortify` package. Secondly, we used ARIMA modeling to forecast unemployment rates with a certain confidence level for the next 50 months through the use of the `auto.arima()` function. From ARIMA modeling, we learned how to summarize our predictions with the `ggtsdiag()` function to break up the summary statistics of the ARIMA model. In addition to the ARIMA model, we used `ets()` to exponentially model our predictions with a weighted moving average (more weights on the more recent observations). Lastly, we used `monthplot()` to observe seasonal changes in our data to study the subjective commencement and ending of the Great Recession period. R proves to be very useful for converting certain objects to different types to uniquely and appropriately perform data manipulation and forecasting on them.

VII. References

All of the references below are listed in order in which they appear within the post for ease.

Reference 1: FRED Civilian Unemployment Rate - https://fred.stlouisfed.org/series/UNRATE

Reference 2: Plotting Time Series Functions with Autoplot - https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_ts.html

Reference 3: Forecasting with ARIMA - http://www.forecastingsolutions.com/arima.html

Reference 4: Forecasting - ARIMA Models - https://www.otexts.org/fpp/8/1

Reference 5: High Ljung-Box p-values at large lags - https://stats.stackexchange.com/questions/91706/high-ljung-box-p-values-at-large-lags

Reference 6: Error, Trend, and Seasonality - http://ellisp.github.io/blog/2016/11/27/ets-friends

Reference 7: The Great Recession - https://www.investopedia.com/terms/g/great-recession.asp

Reference 8: How to Replicate A Monthly Cycle Chart in R - https://stackoverflow.com/questions/5826703/how-to-replicate-a-monthly-cycle-chart-in-r

Reference 9: Forecasting - Time Series Models - https://www.otexts.org/fpp/6/1

Reference 10: Intro. to Forecasting - https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials