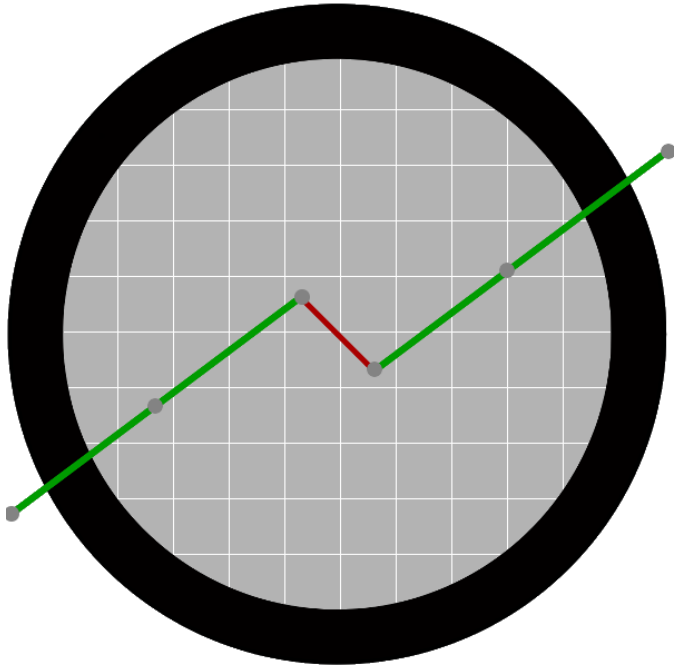
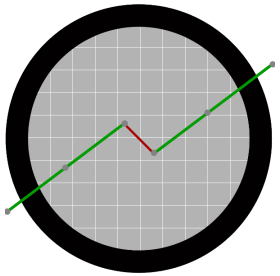
Introduction to
Time Series Analysis



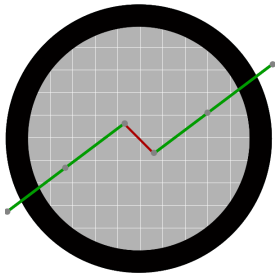
RICE

Fondren Library
Digital Scholarship Services



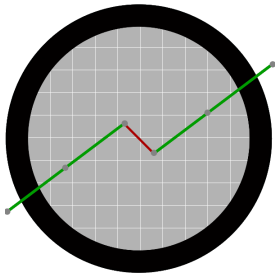
WHAT ARE TIME SERIES?

- Simply, data that are collected in a certain order
- Evenly or unevenly spaced
- Any number of variables



WHAT'S SO SPECIAL ABOUT TIME?

- Many statistical models depend upon data points being independent
  - Linear Regression
  - Logistic Regression
  - Many more!
- With time series, this basic assumption underlying the entire model can be wrong!
- Disastrous results

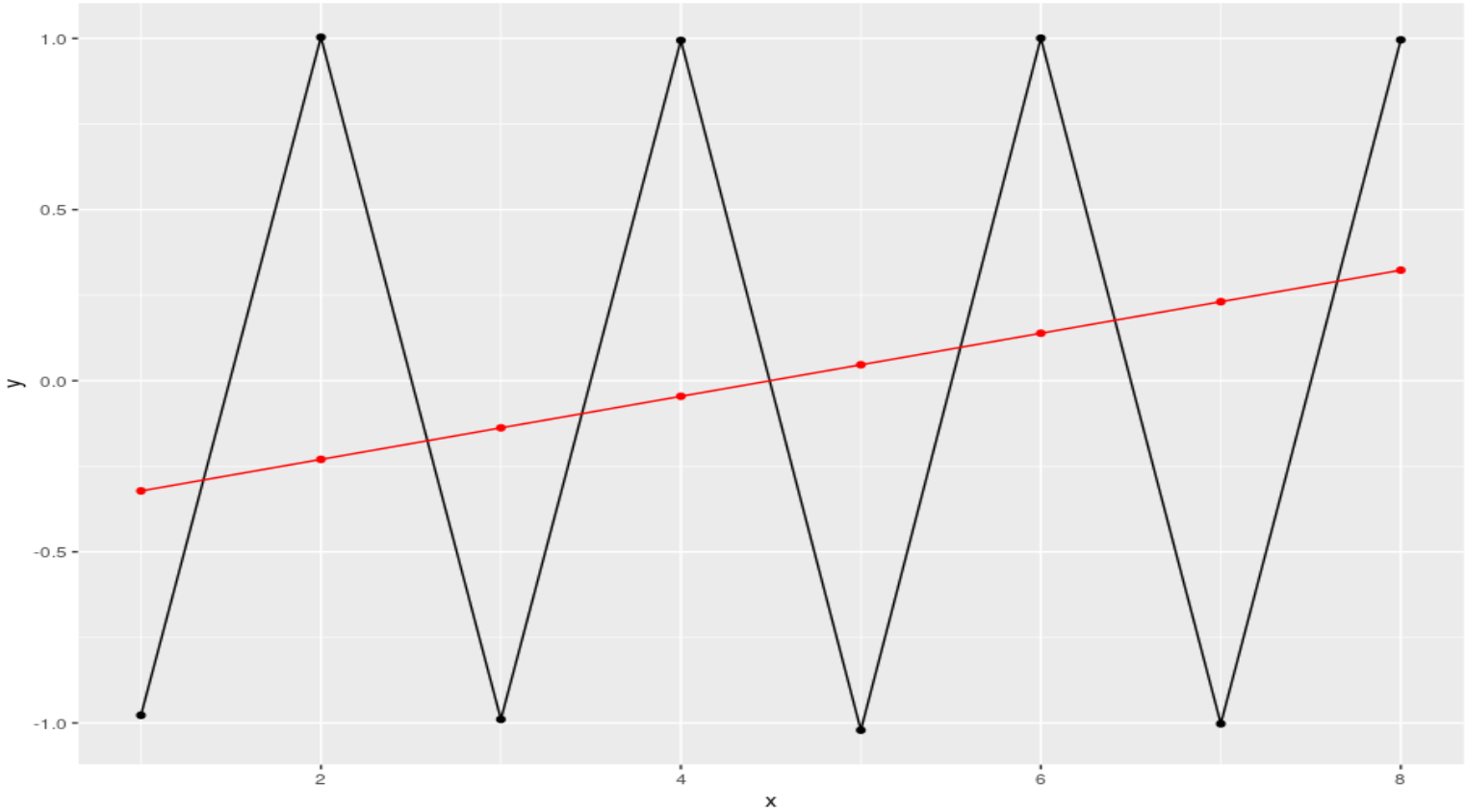


WHY IS THIS IMPORTANT?

- Time series are extremely useful for forecasting
  - Weather
  - Stocks (although this is very hard!)
  - Sports predictions
  - Election predictions
- Well suited to a wide class of problems

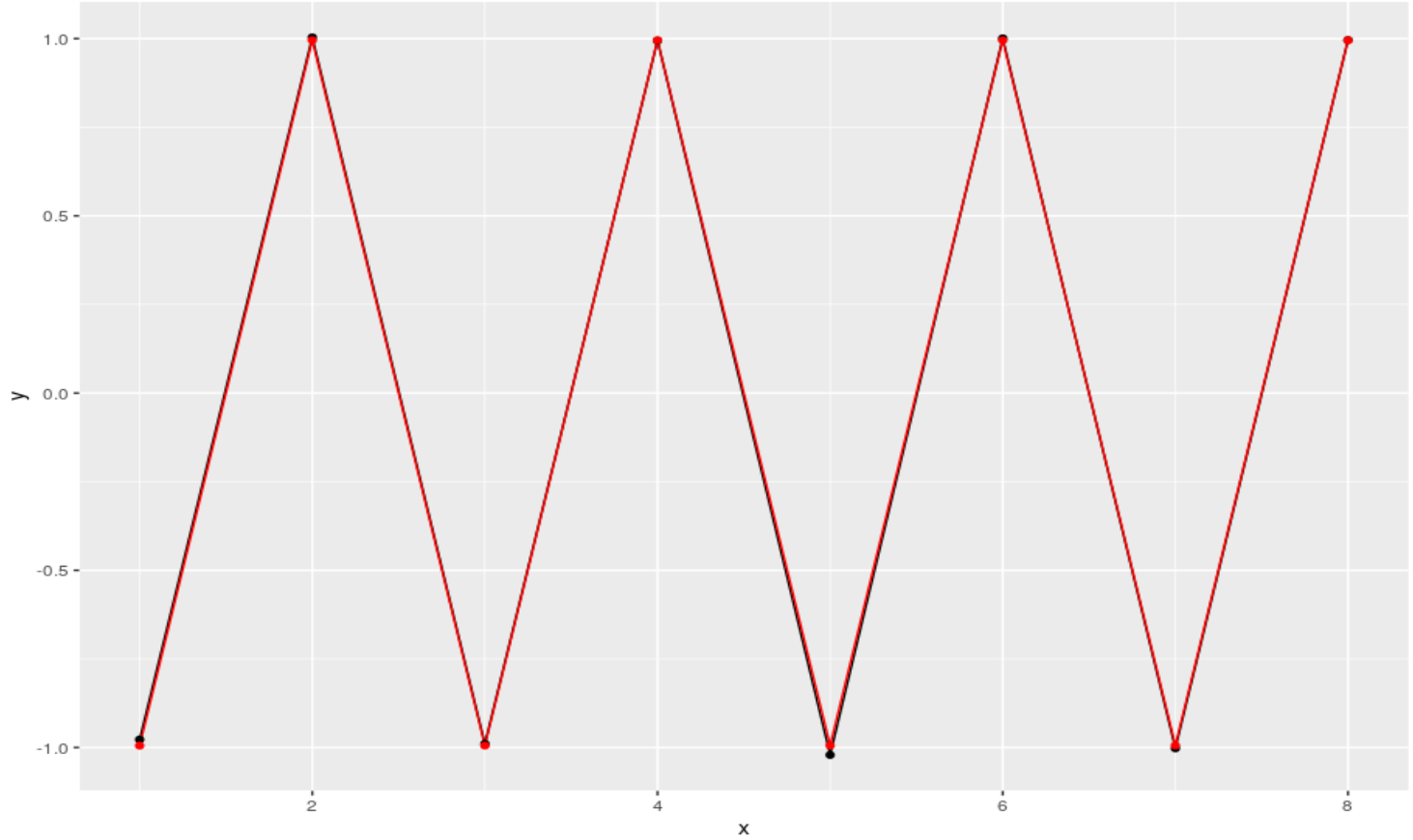
ONE QUICK EXAMPLE

Simple Linear Regression vs. a Time Series Model (ARIMA)



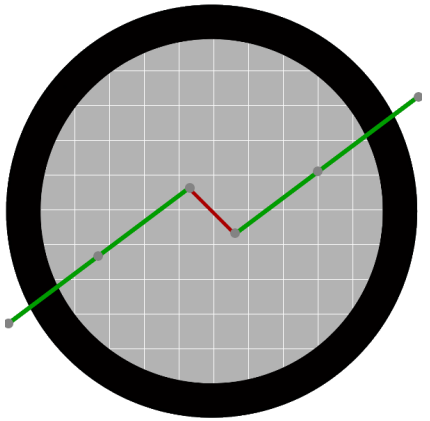
ONE QUICK EXAMPLE

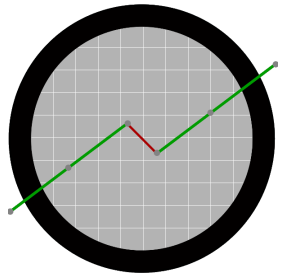
Simple Linear Regression vs. a Time Series Model (ARIMA)



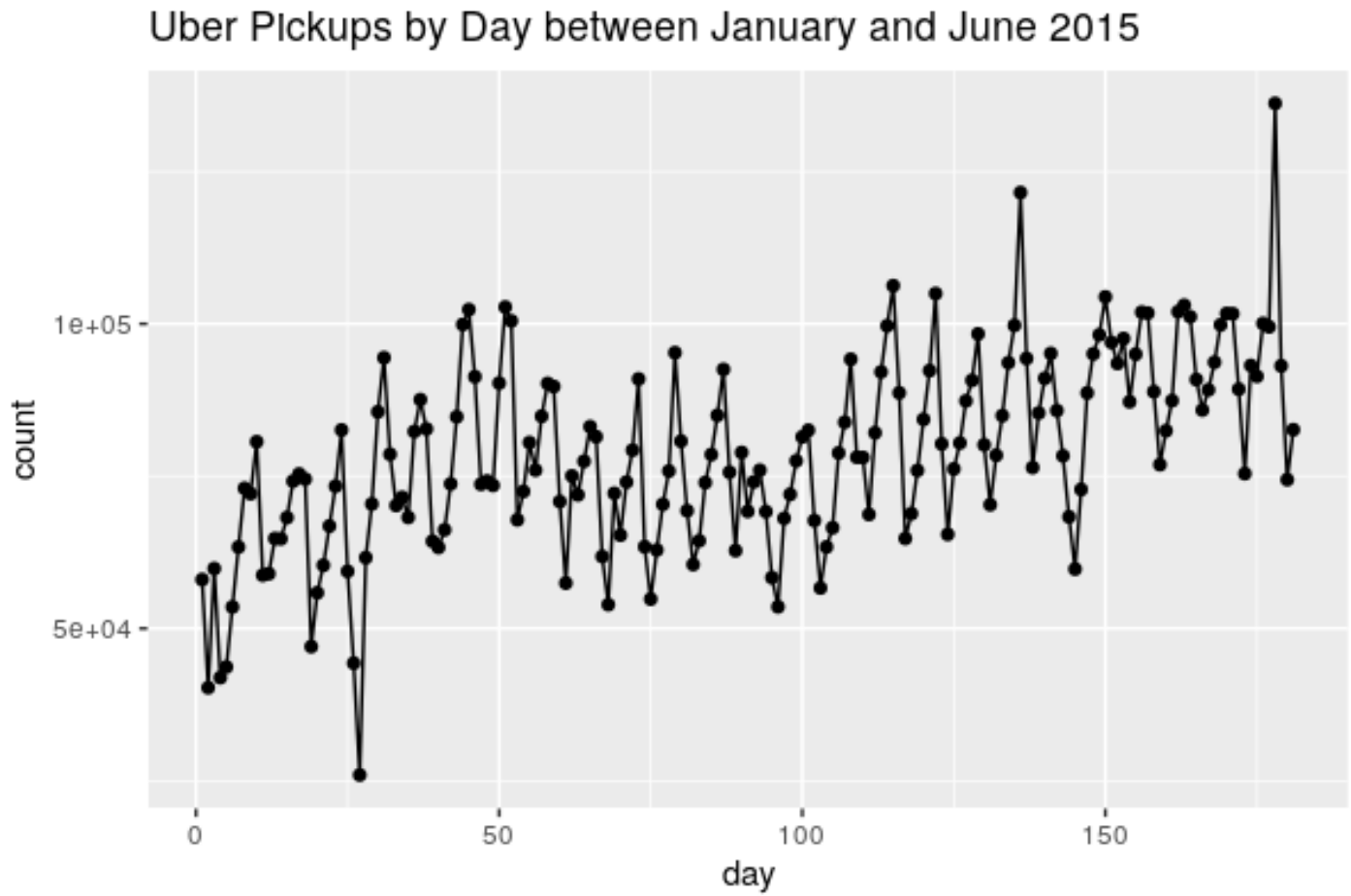
TRENDS AND SEASONALITY

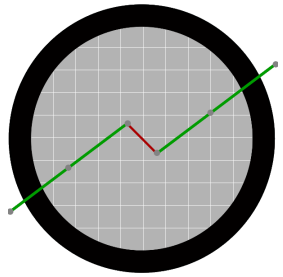
How to filter out the noise and model what really matters



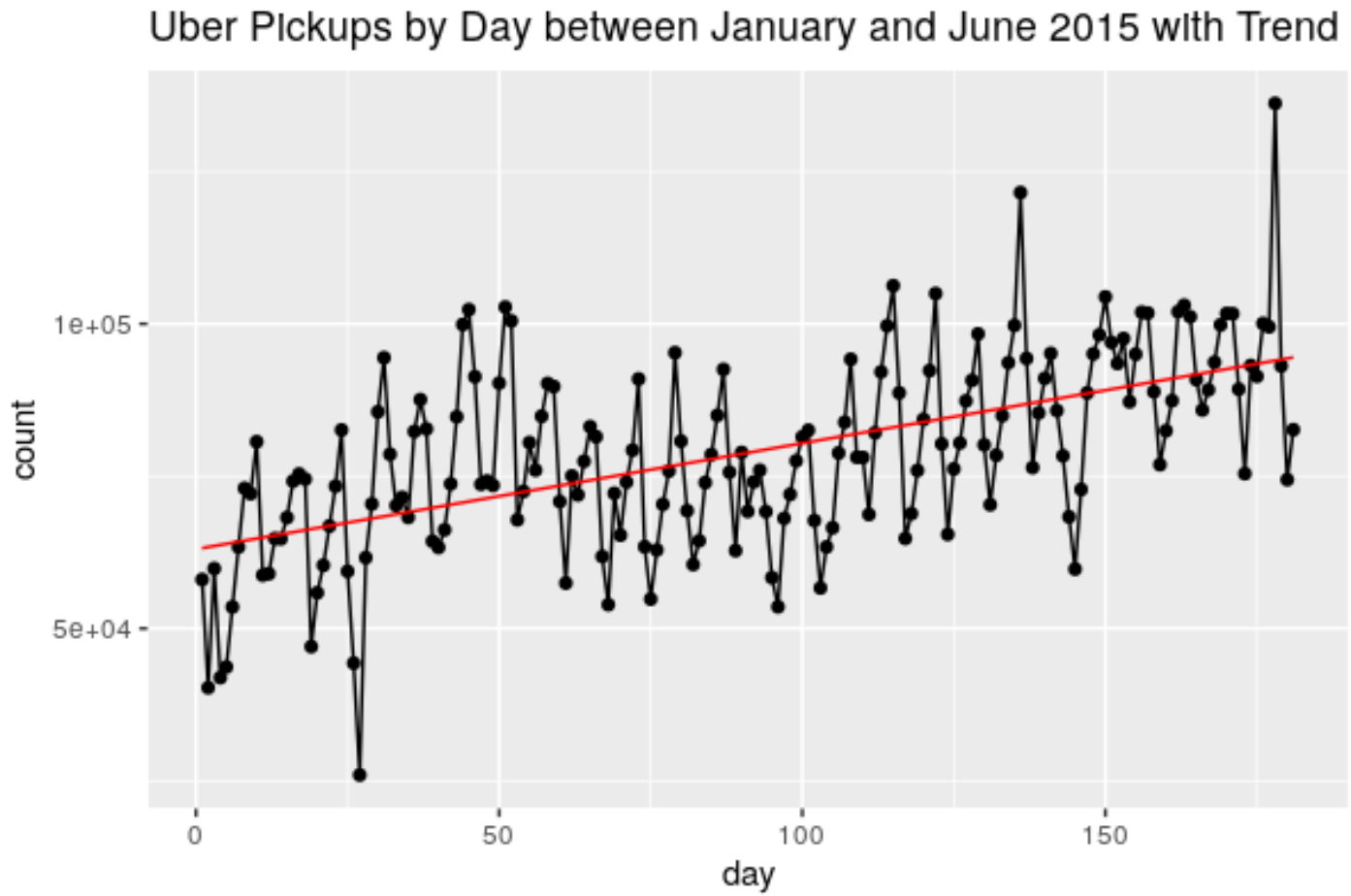


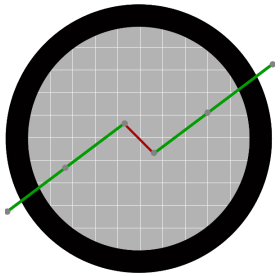
A SAMPLE TIME SERIES





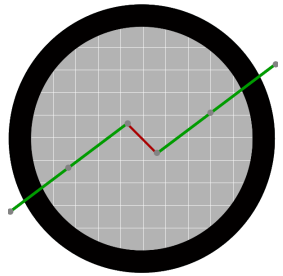
TRENDS


Uber Pickups by Day between January and June 2015 with Trend
count
1e+05
5e+04
0
50
100
150
day

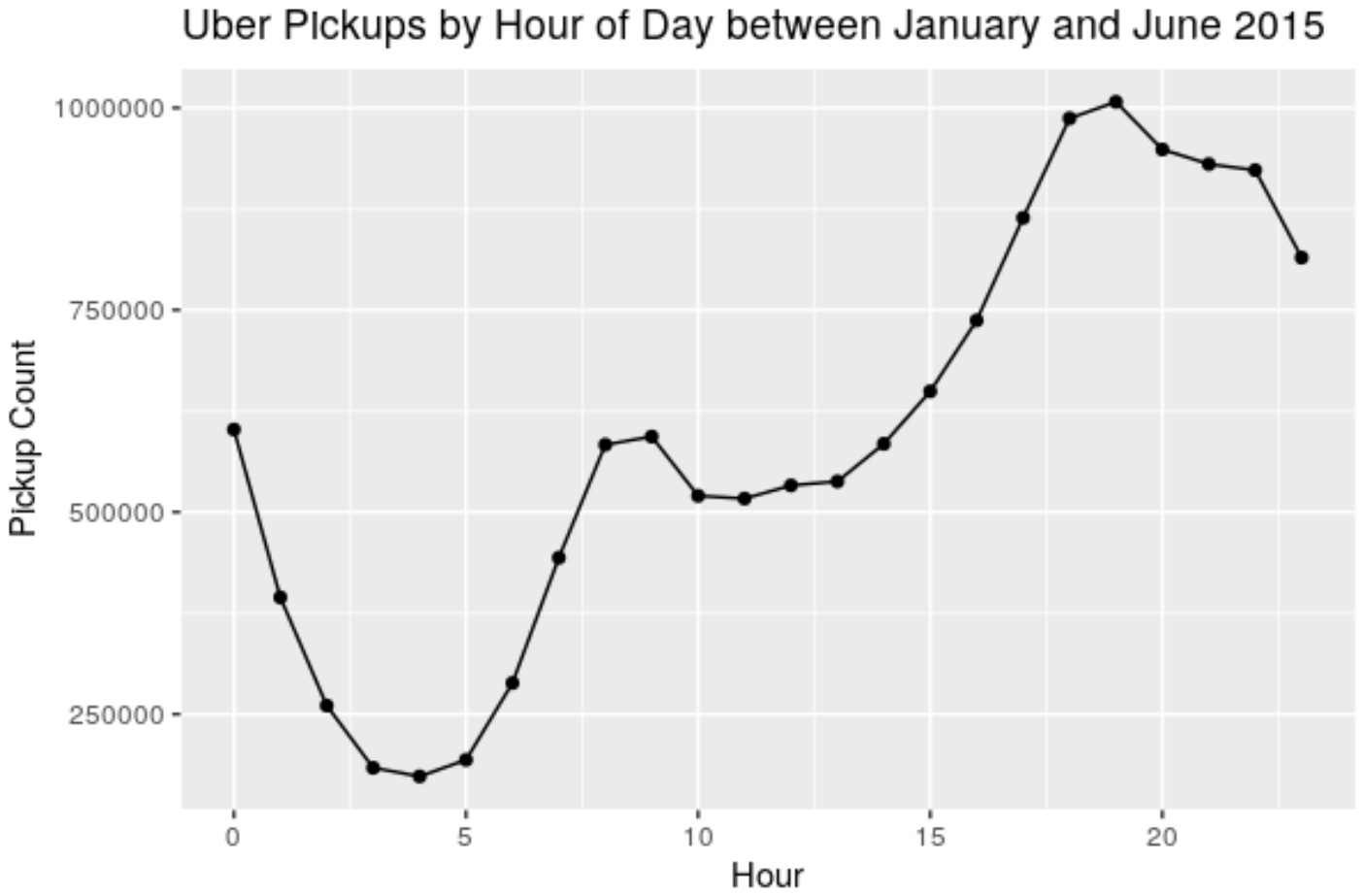


PERIODIC COMPONENTS

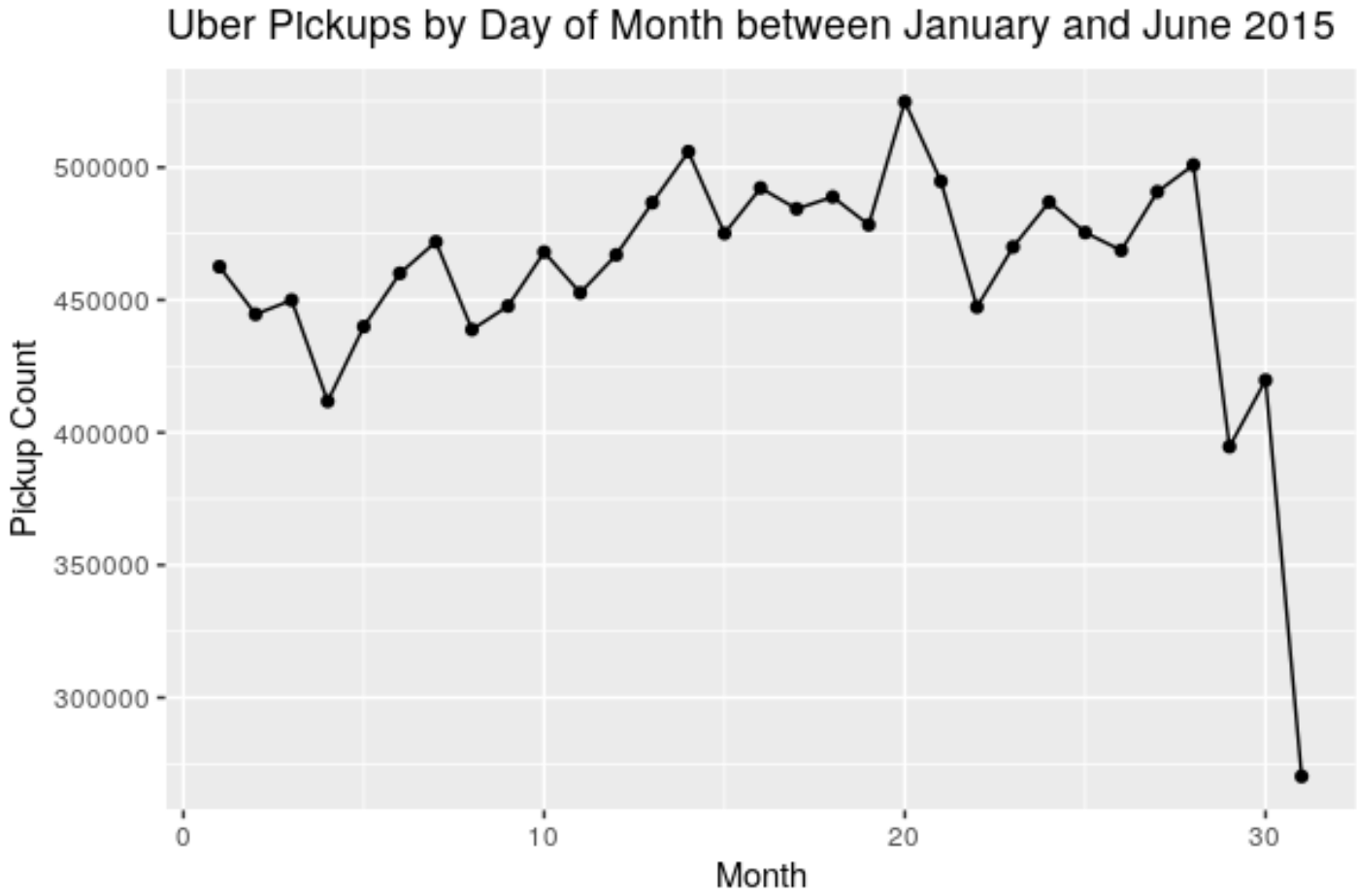
- Cycles that repeat
- Commonly by month, day of week, day of year, etc.
- Also called seasonality

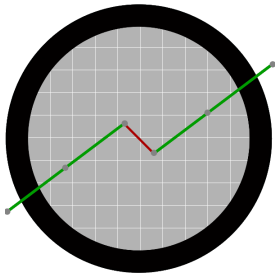


BY HOUR OF DAY

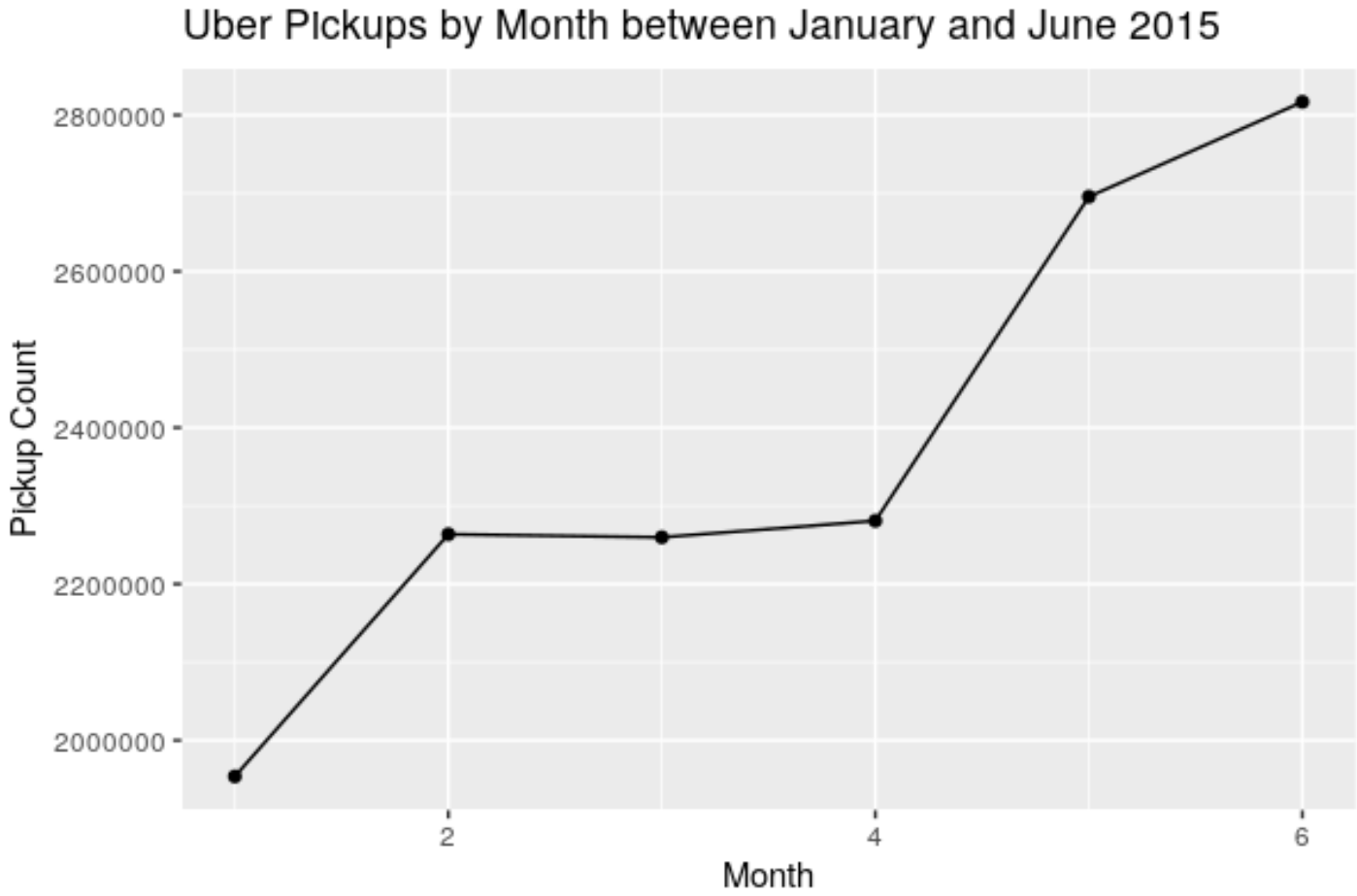


BY DAY OF MONTH

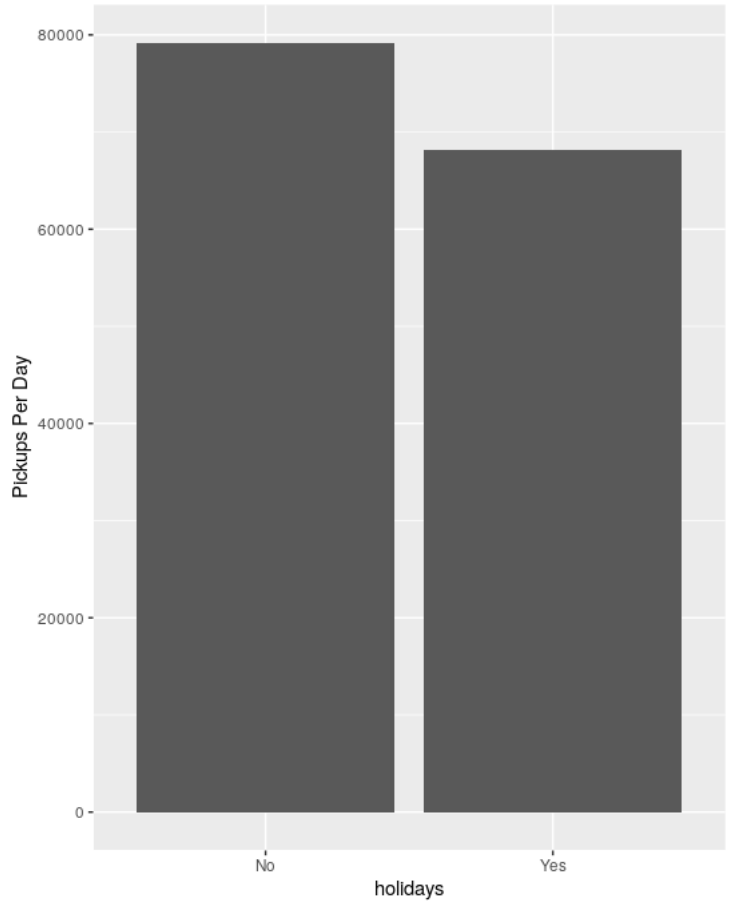


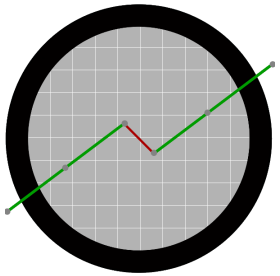


BY MONTH

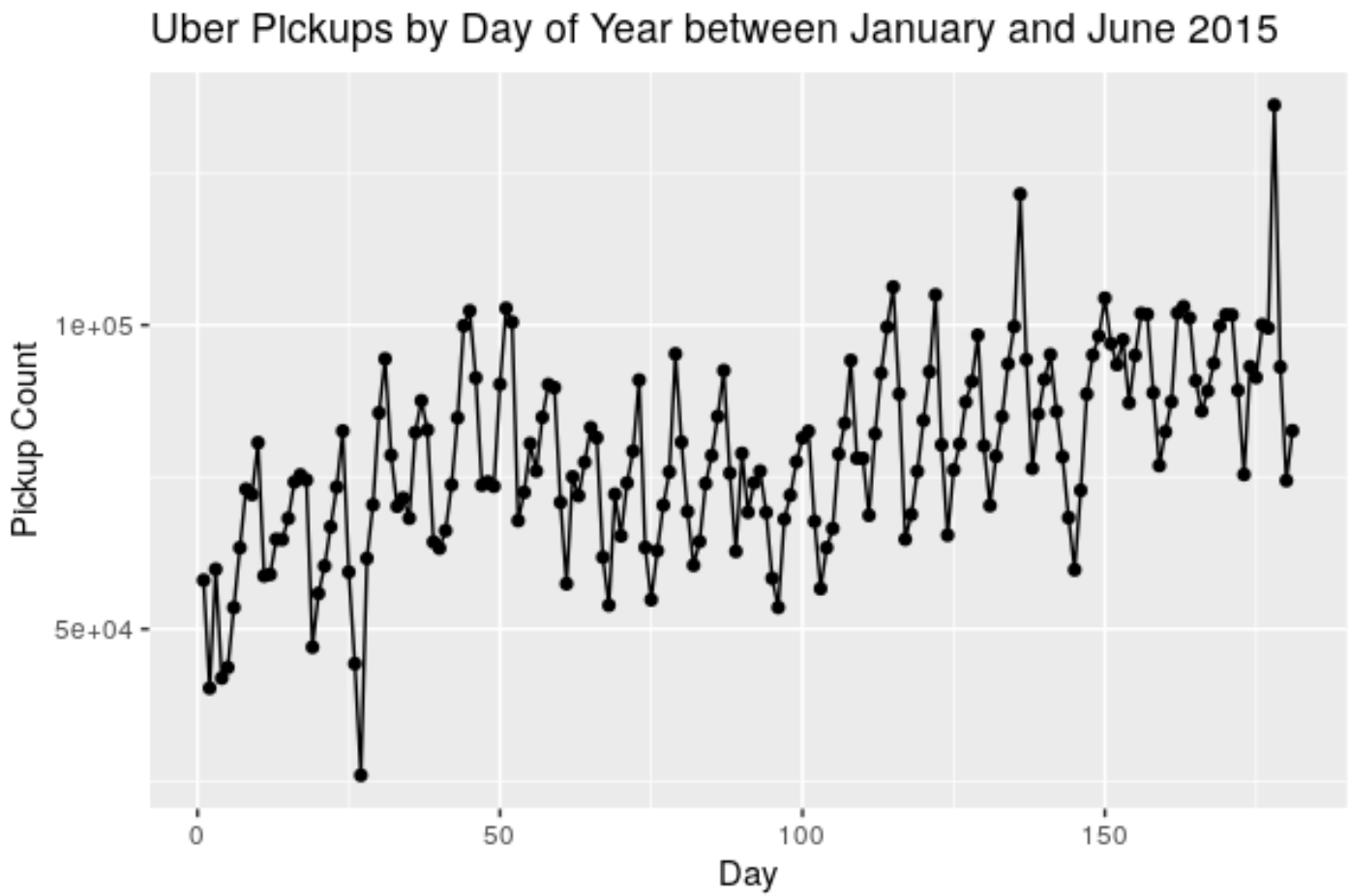

Uber Pickups by Month between January and June 2015
Pickup Count
2800000
2600000
2400000
2200000
2000000
2
4
6
Month

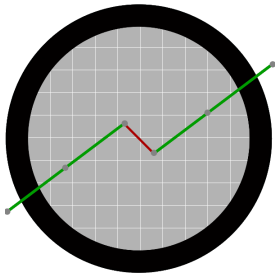
HOLIDAYS





LOOKING BACK AT IT



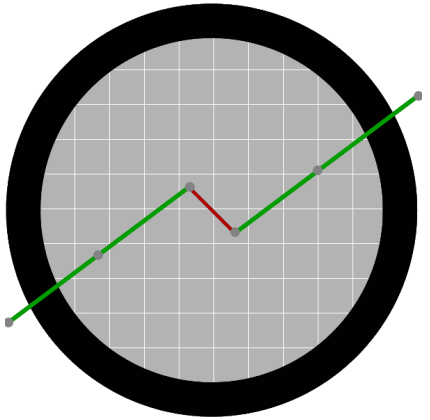


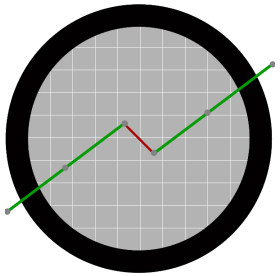
WHITE NOISE

- Not very easy to quantify, but the assumption is that there is some randomness we haven't accounted for
- The statistician's approach: assume this randomness behaves nicely so we can build some great models!

HOW TO MAKE YOUR DATA STATIONARY

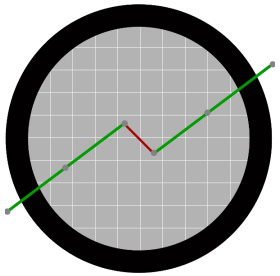
Make your data appetizing for modeling





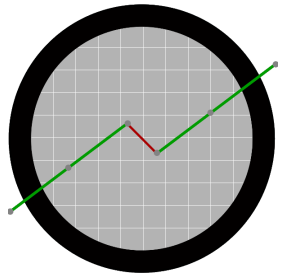
WHAT IS STATIONARY DATA?

- Intuitively, it doesn’t matter where you are in the series
  - If it’s Monday or Friday, you can model it the same way
- Once we have our data stationary (or at least close), we can build amazing models!
- Can’t have a trend
  - If we do, we need to know when we are
- Should minimize interaction between data points



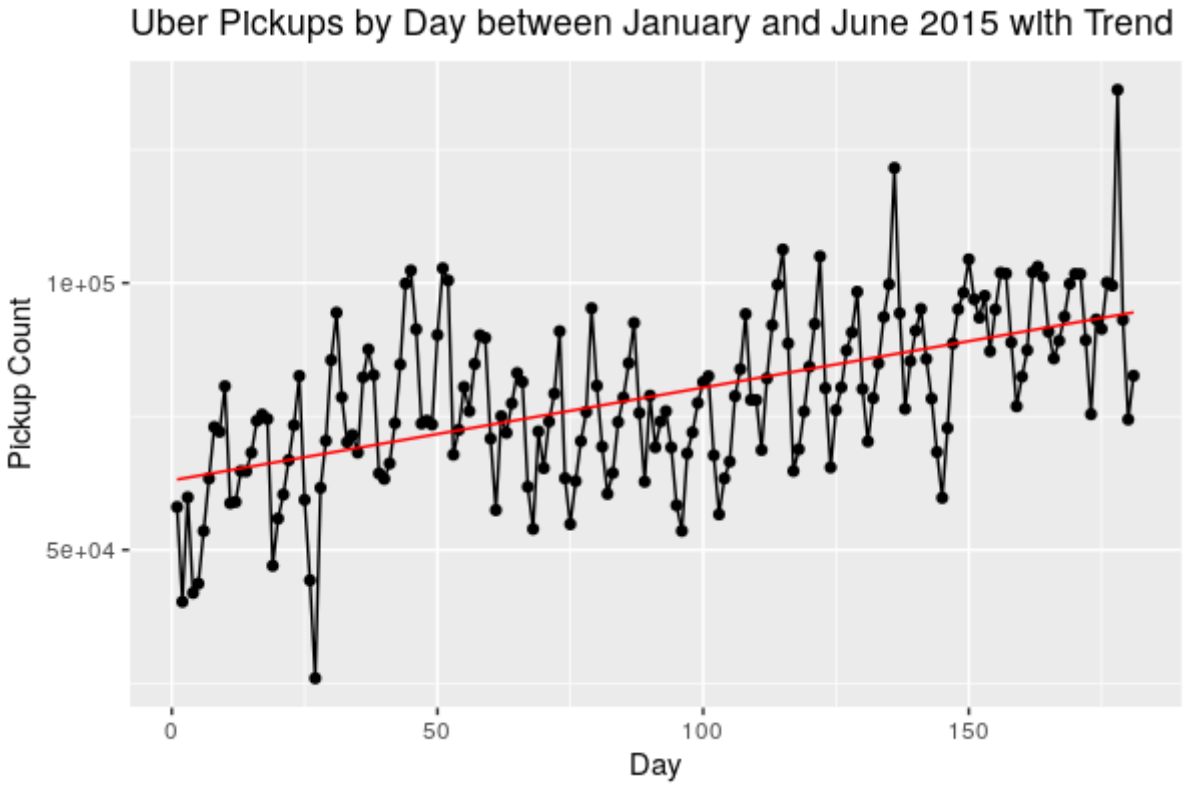
DETRENDING

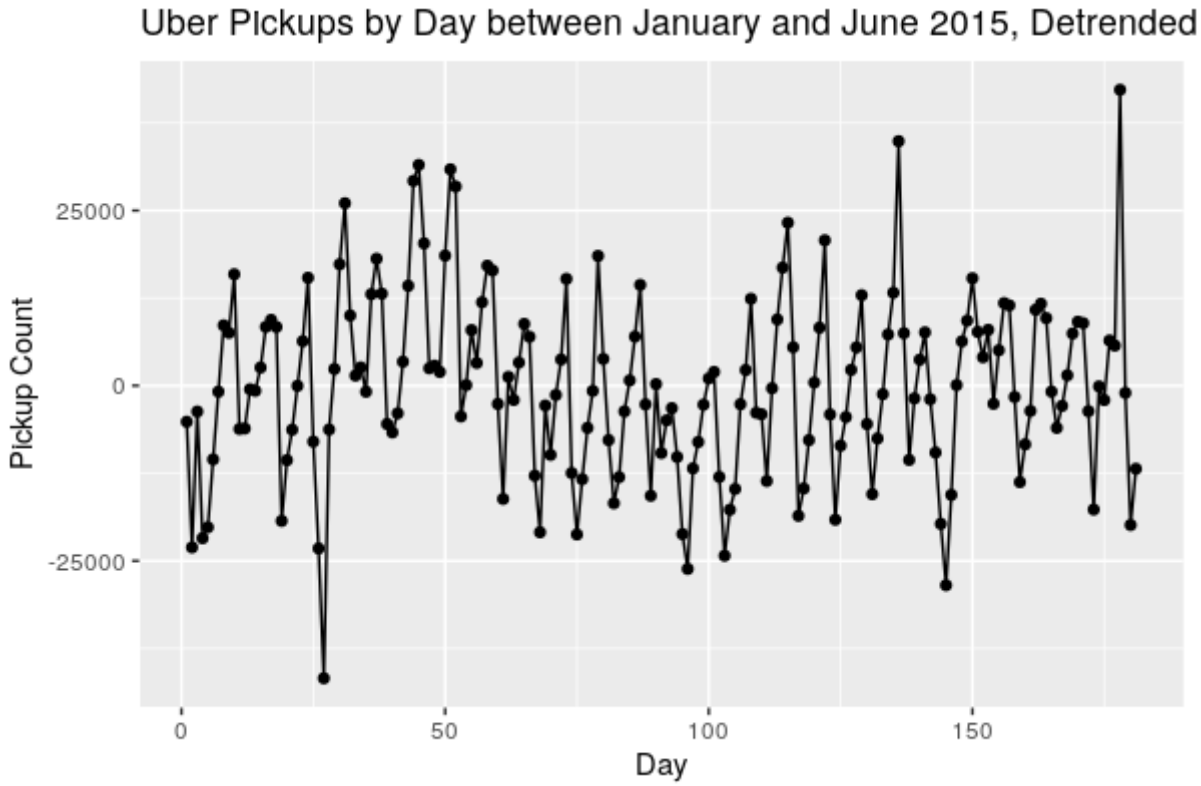
- Subtracting our estimate of the data looking at trend only
- But where do we get the trend from?
  - We have to estimate it ourselves!
  - What kind of trend is it? Trends don’t have to be linear!
  - Something else that could go wrong…
  - Can we do better?

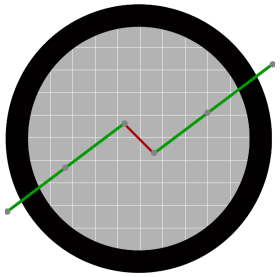


DETRENDING EXAMPLE

- Subtracting our estimate of the data looking at trend only

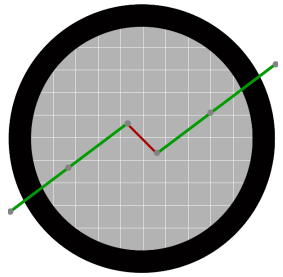




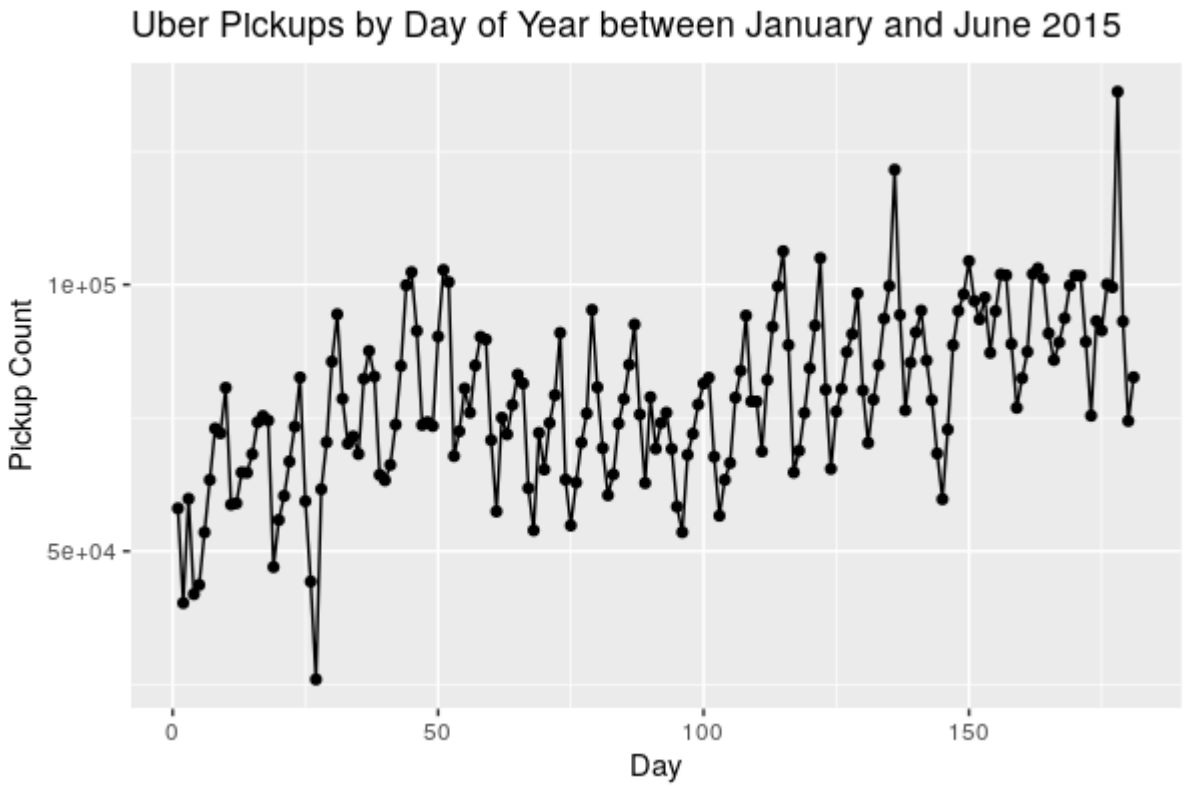


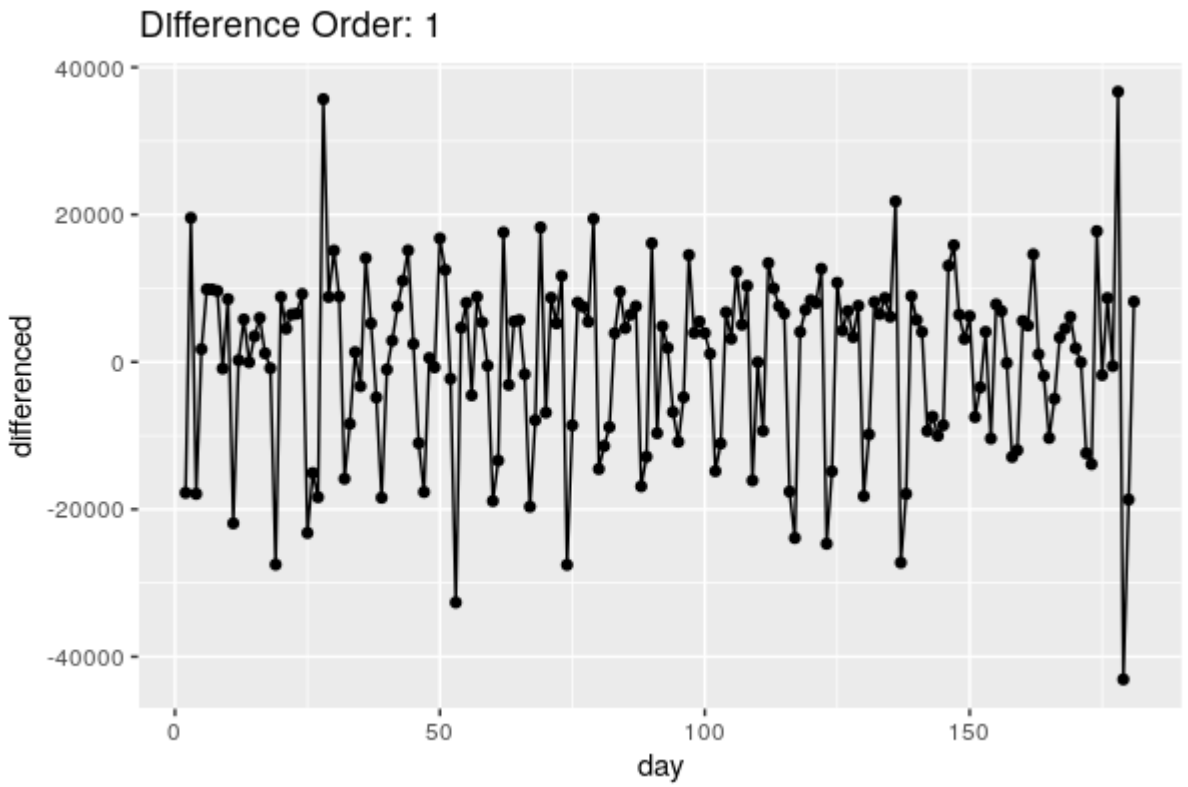
DIFFERENCING

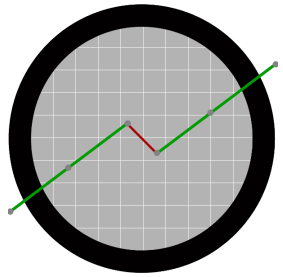
- Subtracting values separated by some constant value
- Say we have a list of 1, 7, 10, 13, and 2
  - First order difference: 7-1, 10-7, 13-10, 2-13
  - Third order difference: 13-1, 2-7
  - Losing data at the start!
- If there's a linear trend, first order differencing will get rid of it!
  - You can do higher orders for quadratic trends, etc.
- Also used for removing some cyclic effects
  - Weekly cycle? Difference by 7!
- Can do many of these at once



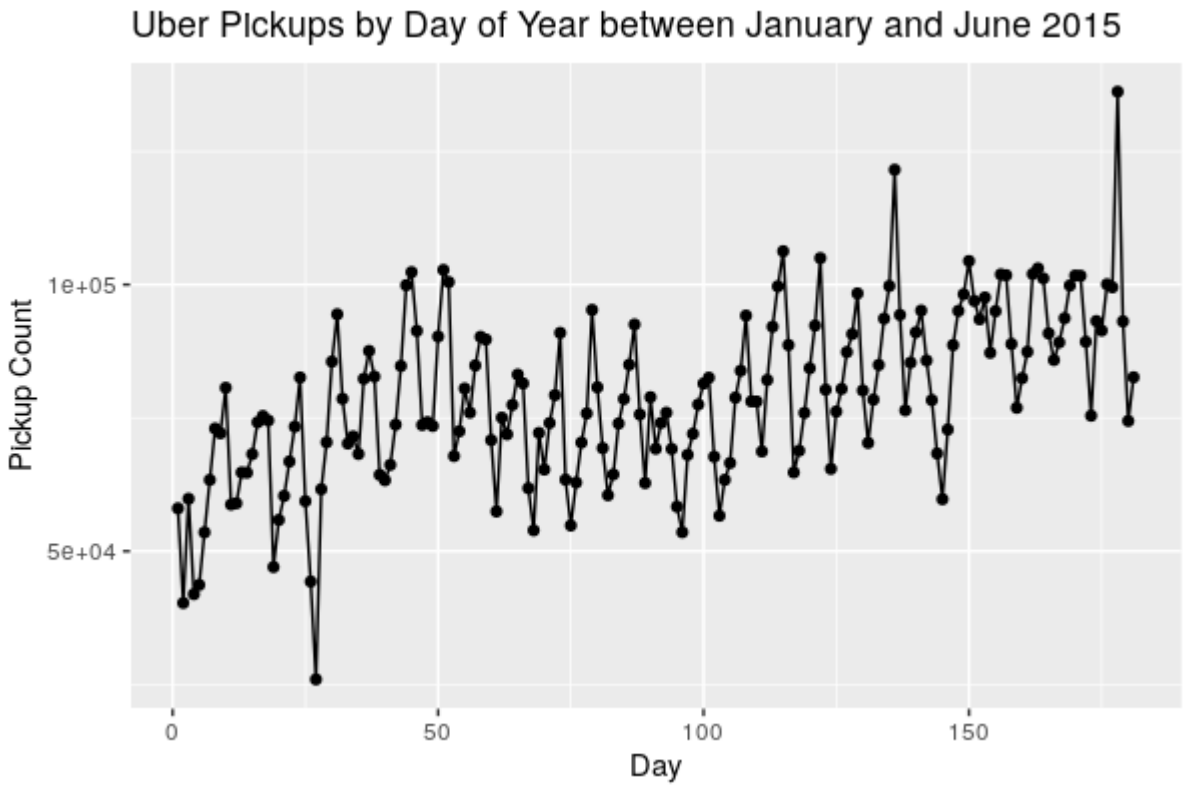
DIFFERENCING EXAMPLE: 1st ORDER

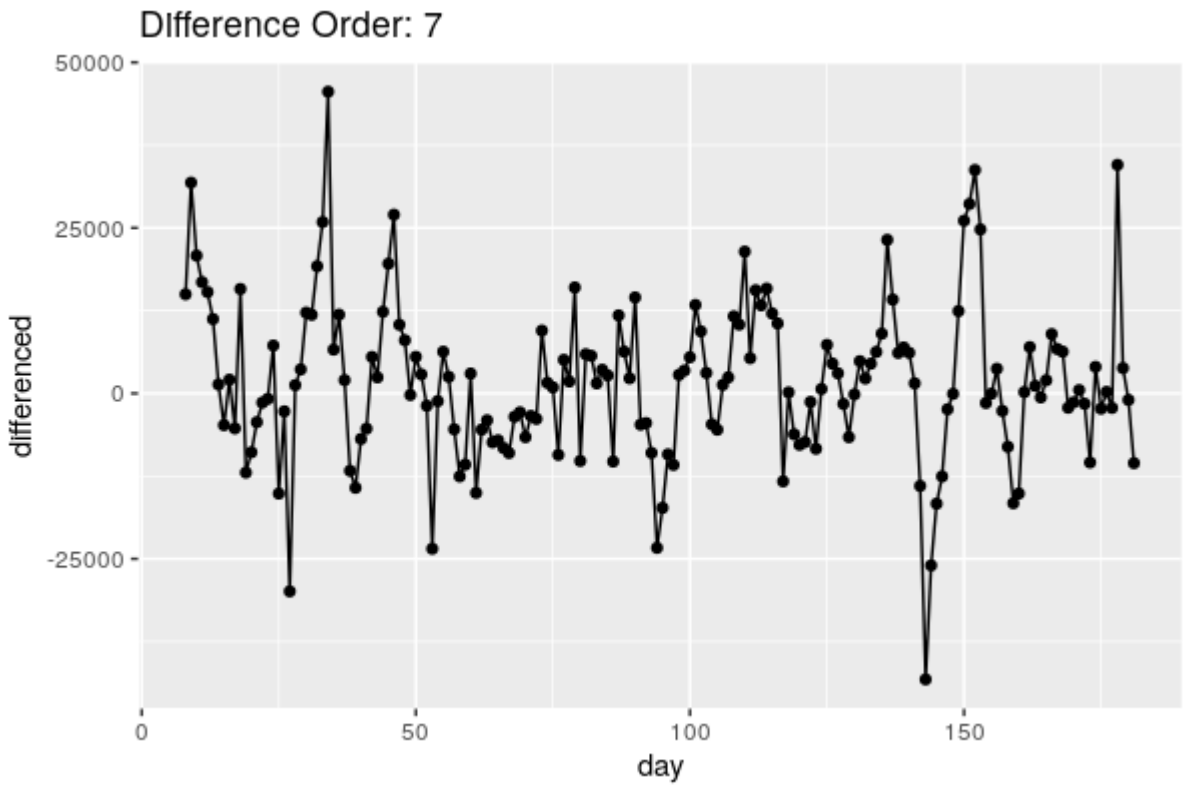


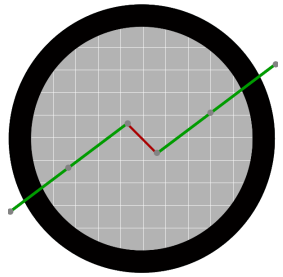




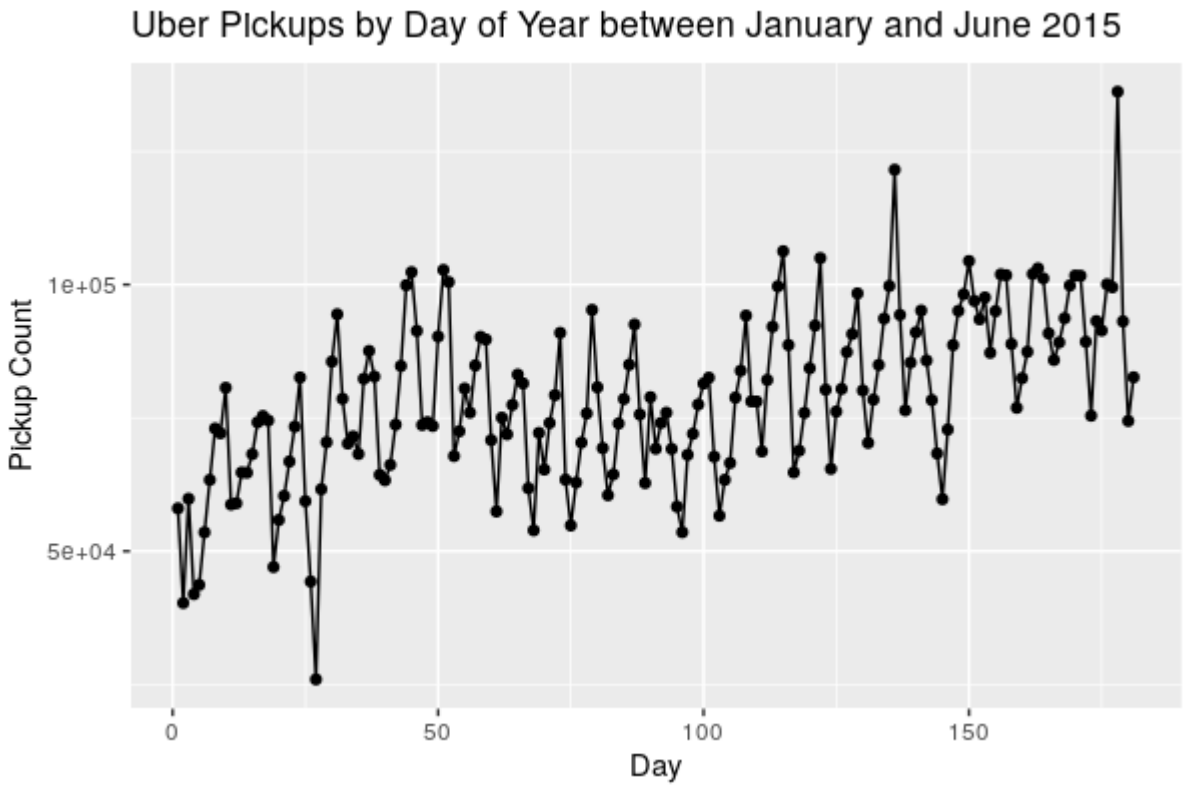
DIFFERENCING EXAMPLE: 7th ORDER

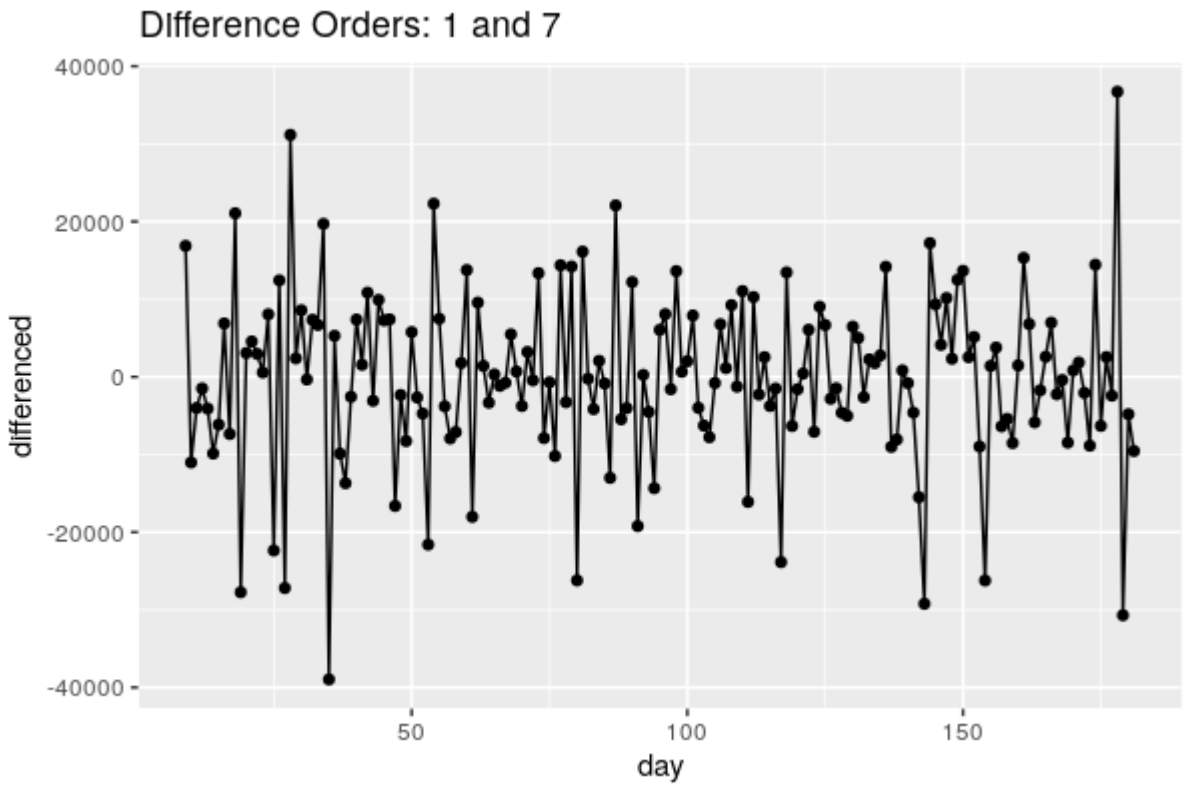


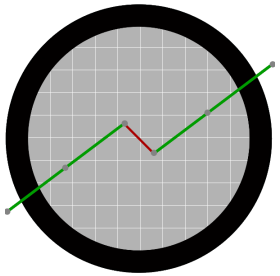




DIFFERENCING EXAMPLE: BOTH!



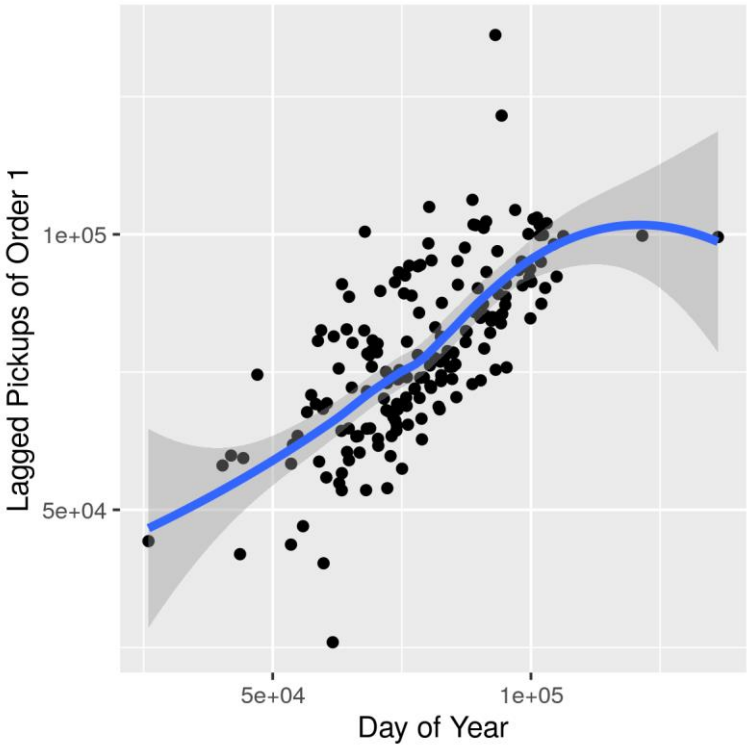


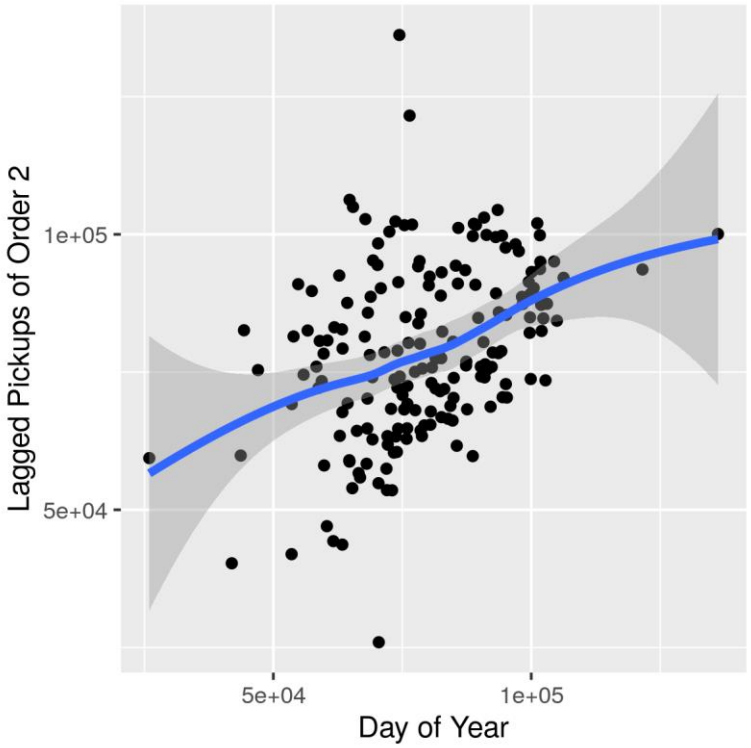


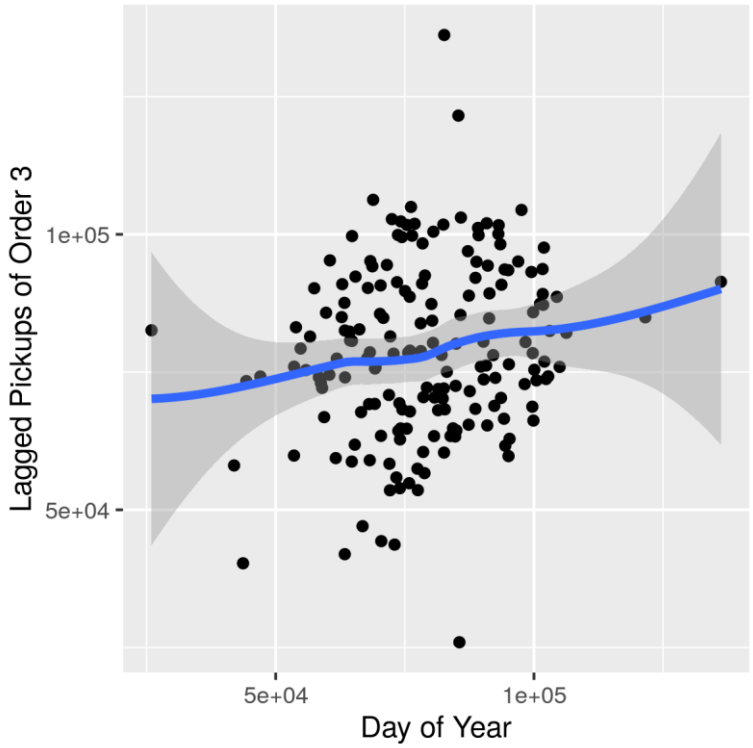
LAGGED SCATTERPLOTS

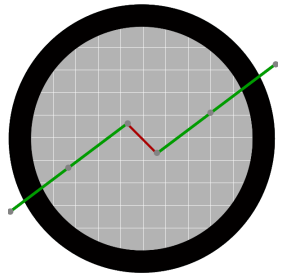
- How do you know what orders are likely to help?
- Lagged scatterplots!
  - Essentially, plotting one point versus a previous point
- If there's an obvious trend (either positive or negative) it's probably a good idea to try differencing!

LAGGED SCATTERPLOTS EXAMPLE

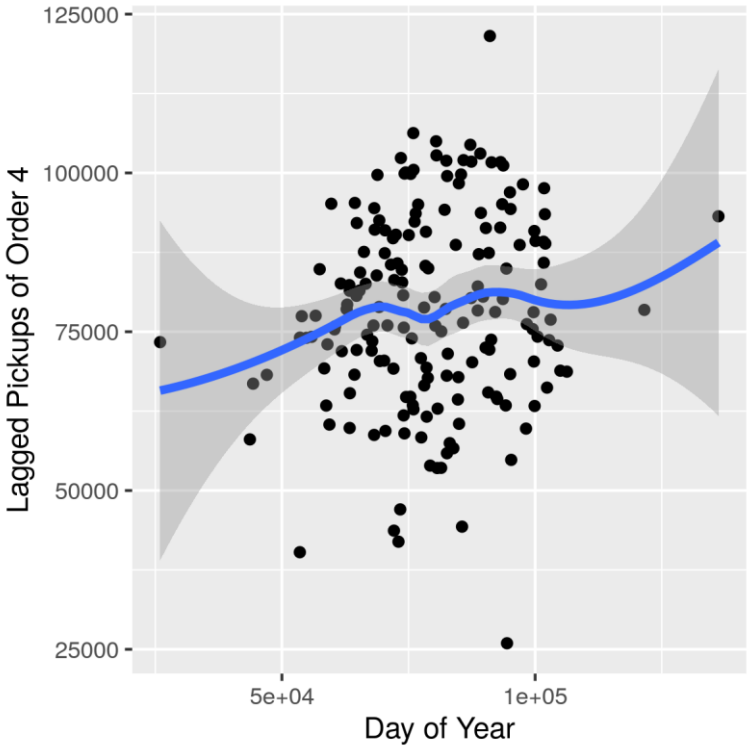


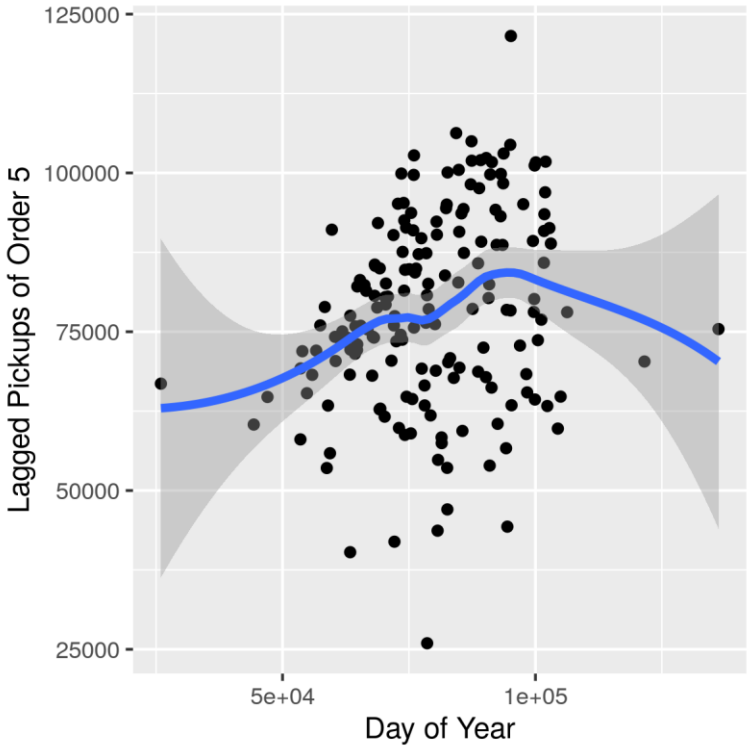


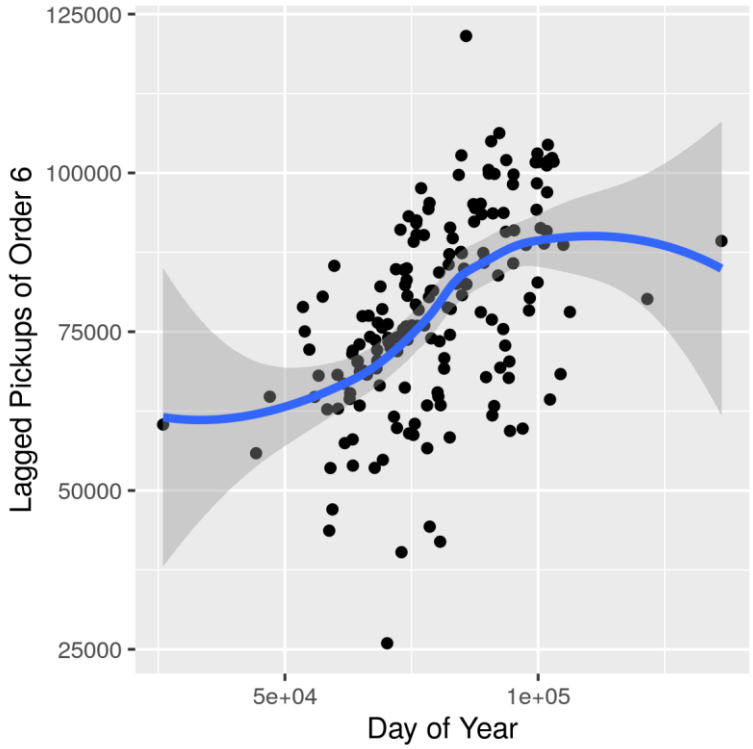




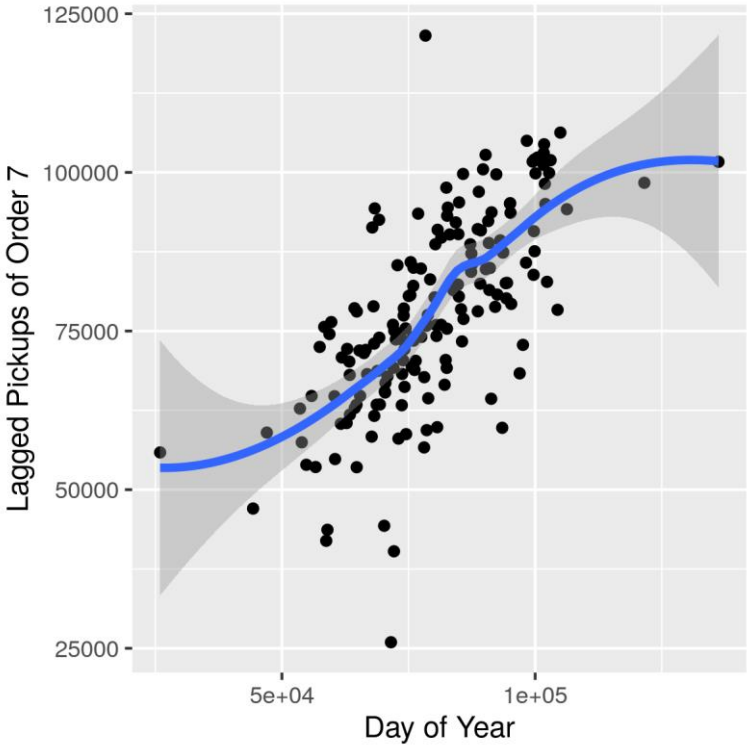
LAGGED SCATTERPLOTS EXAMPLE

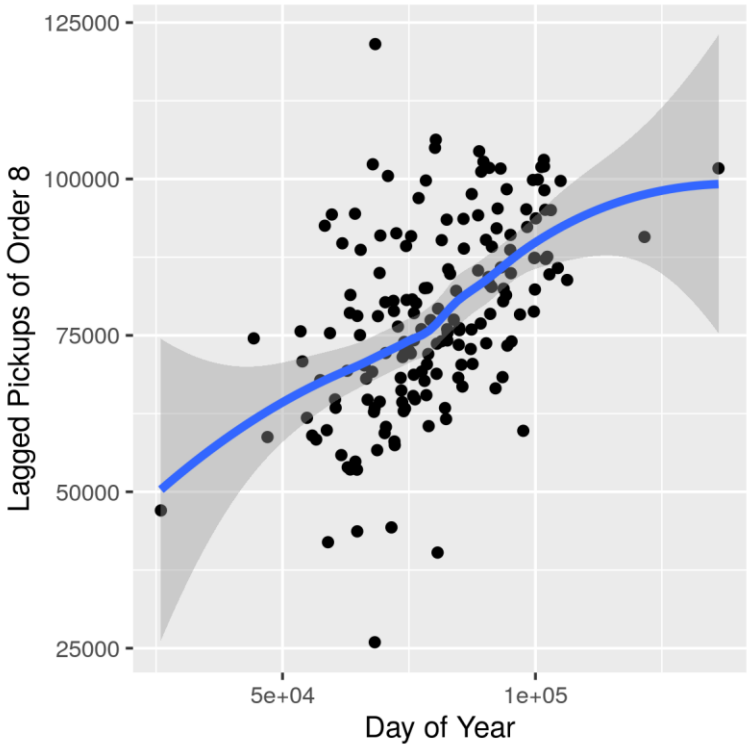


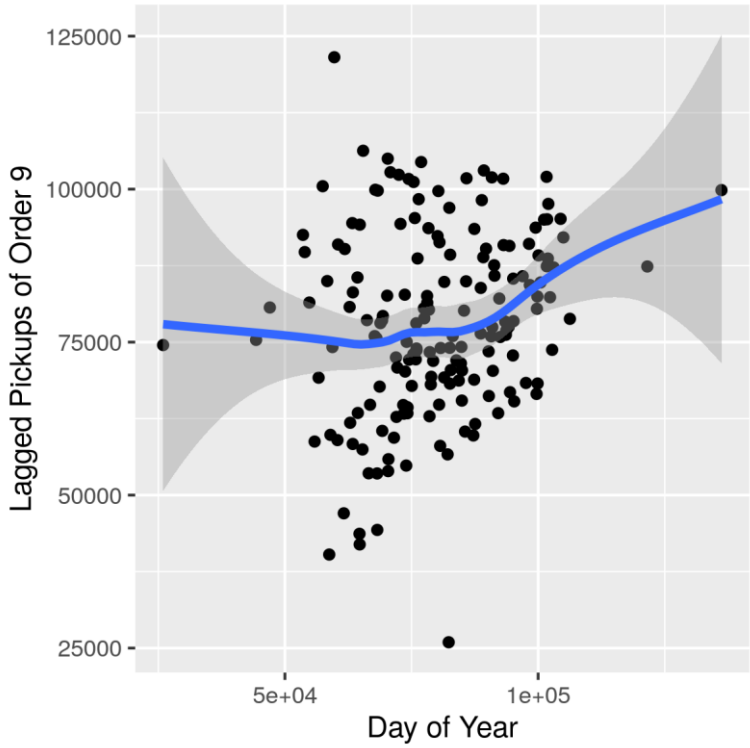


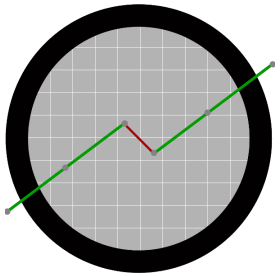


LAGGED SCATTERPLOTS EXAMPLE









HOW DO I KNOW I'M STATIONARY?

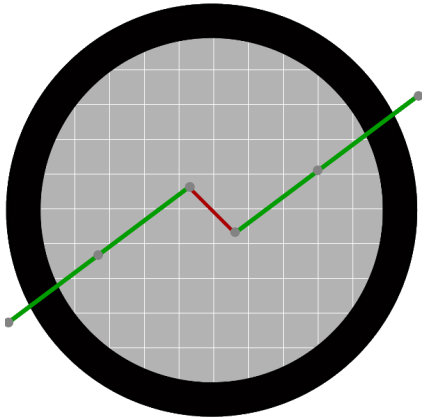
- The Augmented Dickey-Fuller test to the rescue!
  - The p-value in the output is a measure of how stationary the series is
  - If you have less than ~0.05, you're all set to model!
- Python

```
from statsmodels.tsa.stattools import adfuller
adfuller(Your Time Series)
```

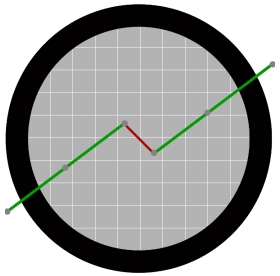
- R

```
library(tseries)
adf.test(Your Time Series)
```

A FEW NOTES

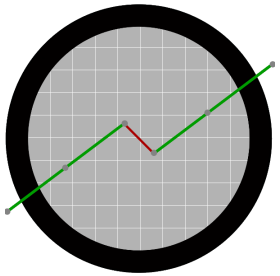


On modeling, forecasting, evaluating, plus explaining what these plots mean



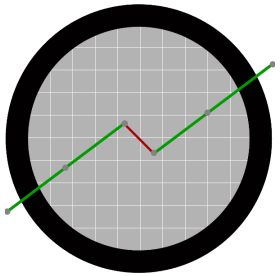
MY MODELING PHILOSOPHY

- “All models are wrong, but some are useful” – George Box
  - You’ll never get everything right
- Try a bunch of things and see what works!
- Interpretability matters
  - If we can predict well but we have absolutely no idea why, is the model really what we want?



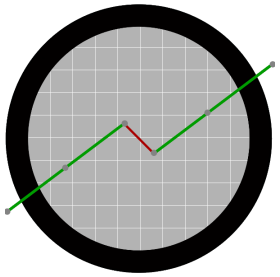
MODELING FOR FORECASTING

- If we use all our data to model, and then “forecast” based on that, we’re cheating!
  - A model could just memorize what we put in and get a perfect score
- The real test of a model is its ability to forecast on unknown data
- Solution: choose a point (usually ~70% of the way through the data set) and split the data into two chunks: before and after
- Model with the first chunk, and check forecasting ability on the second chunk



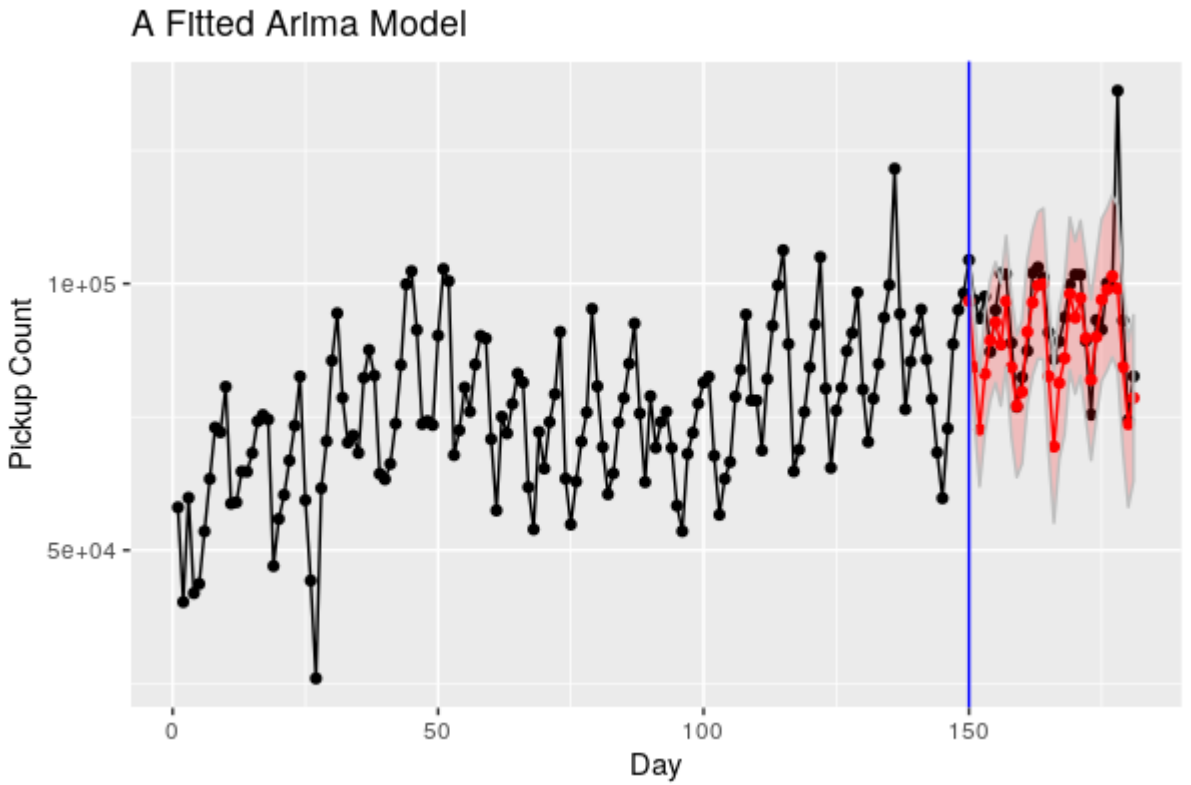
EVALUATING MODELS

- Common statistical measures don’t apply well to time series
- I’m going to be using mean relative error, which measures how close our predictions are on average to the true value
  - Another common metric: Mean Root Squared Error



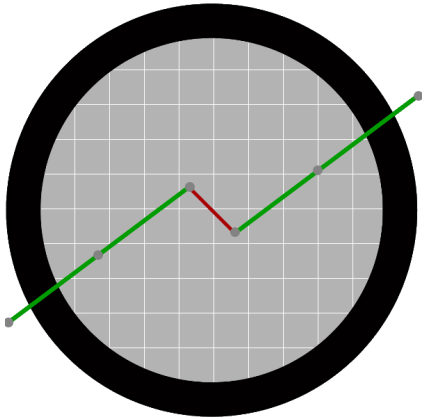
A SAMPLE FORECAST GRAPH

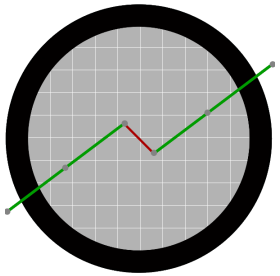
- Vertical blue line: switch from train to test
- Red: predictions
- Faint red ribbon: a prediction interval, if applicable
  - Basically, “I think it’s probably in here somewhere”



ARIMA MODELS

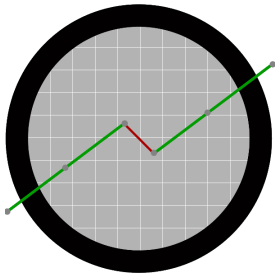
Interpretable, simple and speedy time series modeling





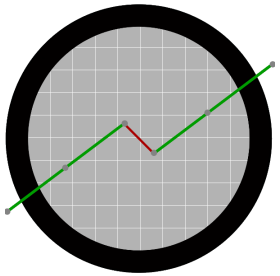
ARIMA MODELS

- Auto-Regressive Integrated Moving Average models
- Pros:
  - Great at producing flexible models, solid support of periodic behavior and trending
  - Parameters are fairly easy to interpret
  - Widely used and mathematically well-understood
- Cons:
  - Bit of a pain to figure out the right parameters
  - Not great at dealing with holidays, etc



WHAT IS AUTOCORRELATION?

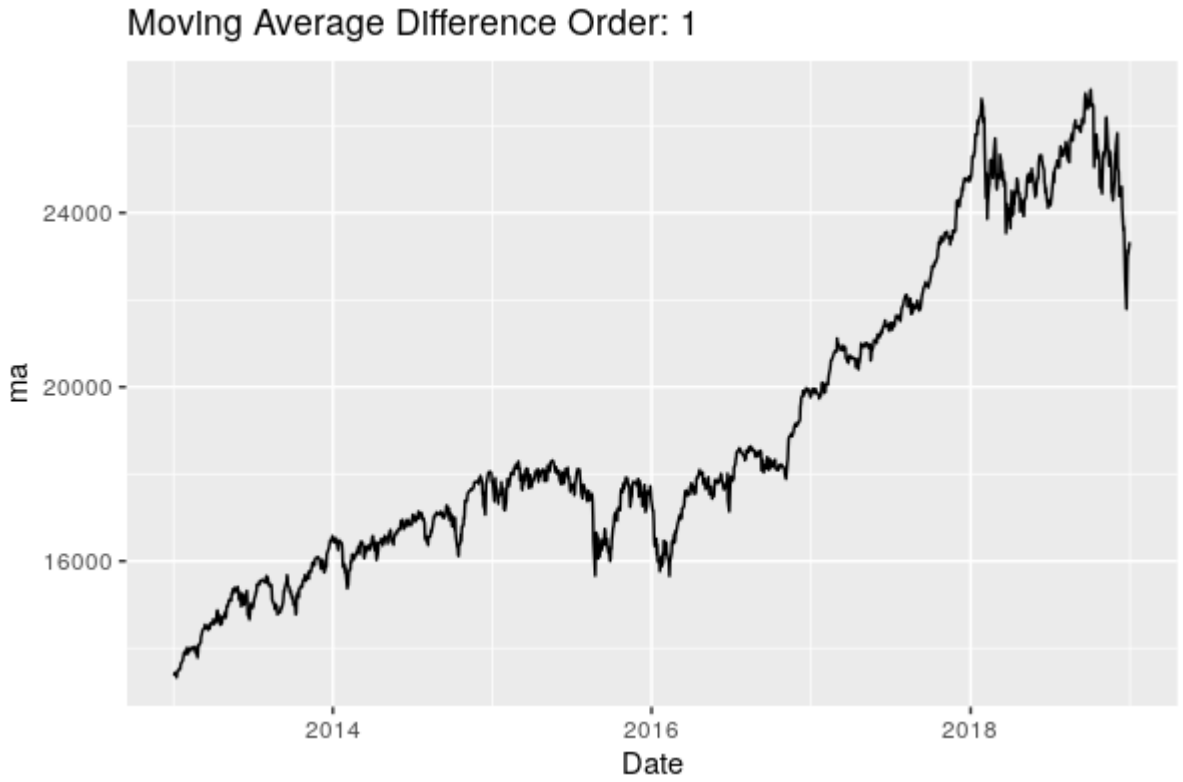
- A measure of how much values depend on previous values
- The “autoregressive” part of ARIMA
  - So we’re modeling based on the assumption that these relationships matter and aren’t just random noise

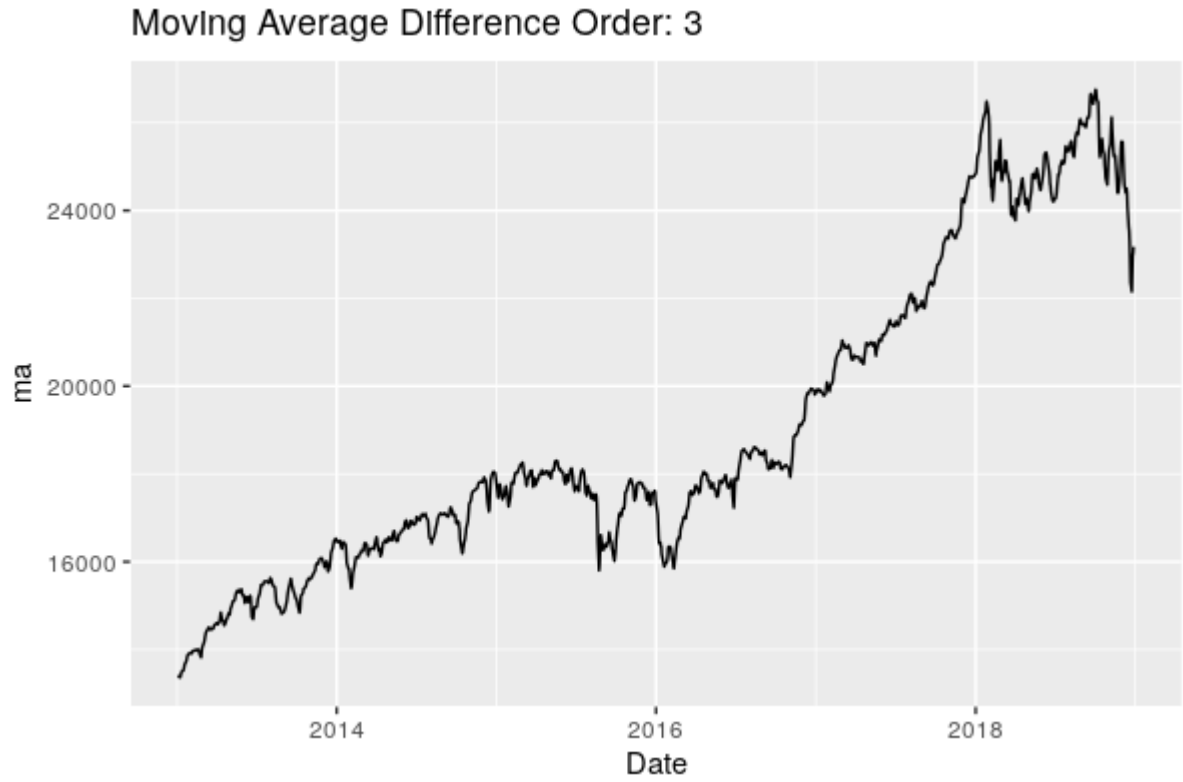


WHAT'S A MOVING AVERAGE?

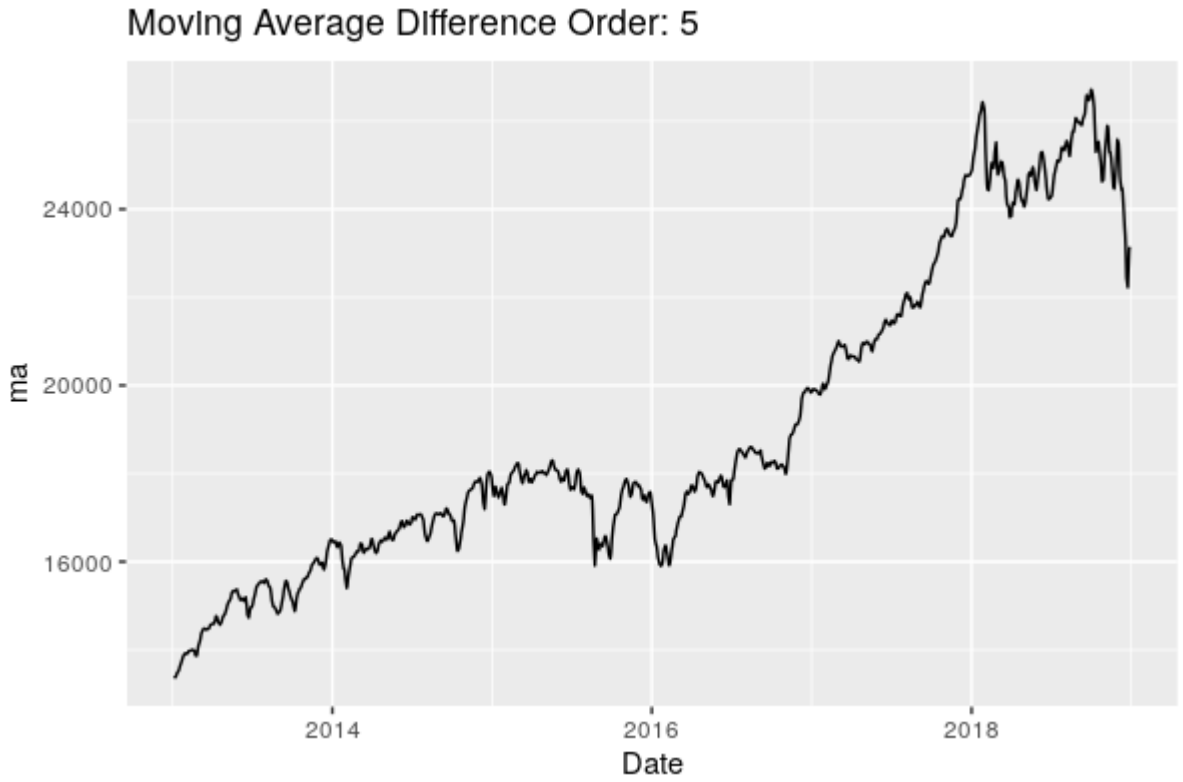
- Essentially, a sliding window of interest upon which we take an average
- Say we have a list of 1, 7, 10, 13, and 2
- A moving average of order 3 would take all triples of components and average them
  - (1 + 7 + 10)/3, (7 + 10 + 13)/3, (10 + 13 + 2)/3
  - Notice we're losing data points!
- A kind of smoothing to reduce the effect of massive spikes

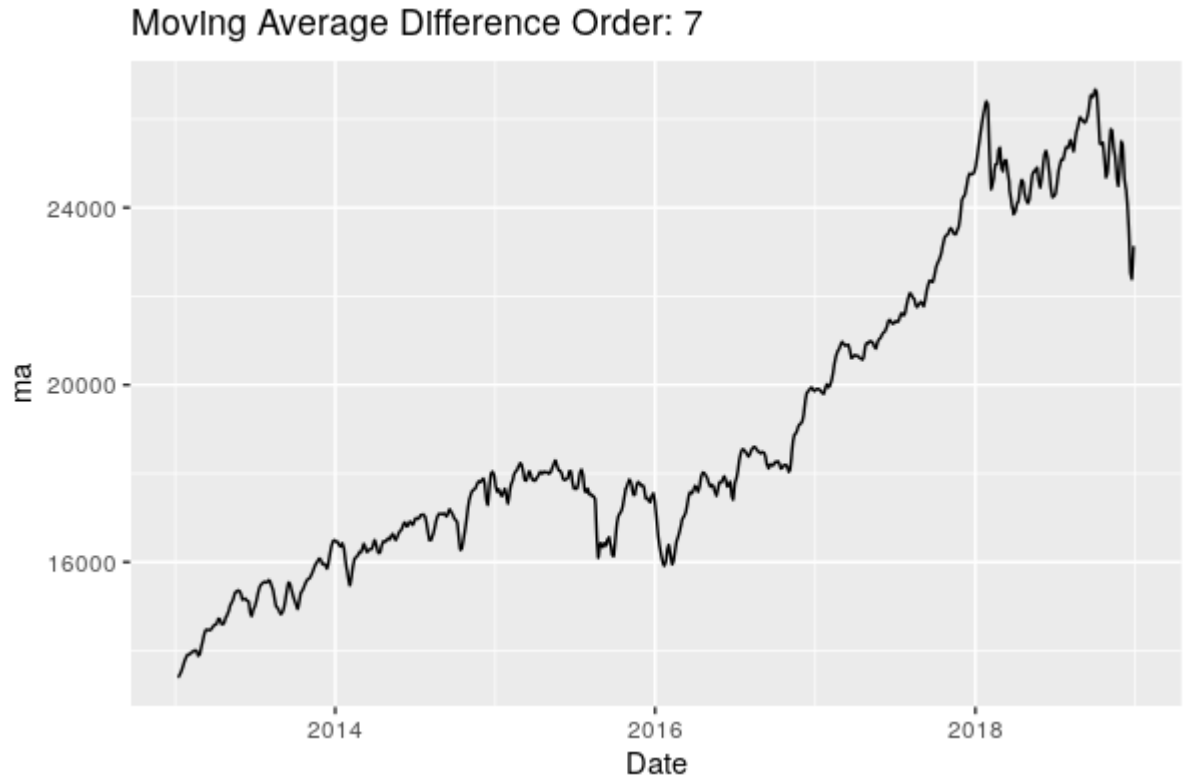
MOVING AVERAGES IN ACTION: 1 & 3

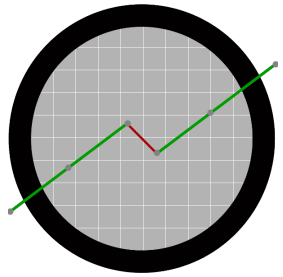




MOVING AVERAGES IN ACTION: 5 & 7

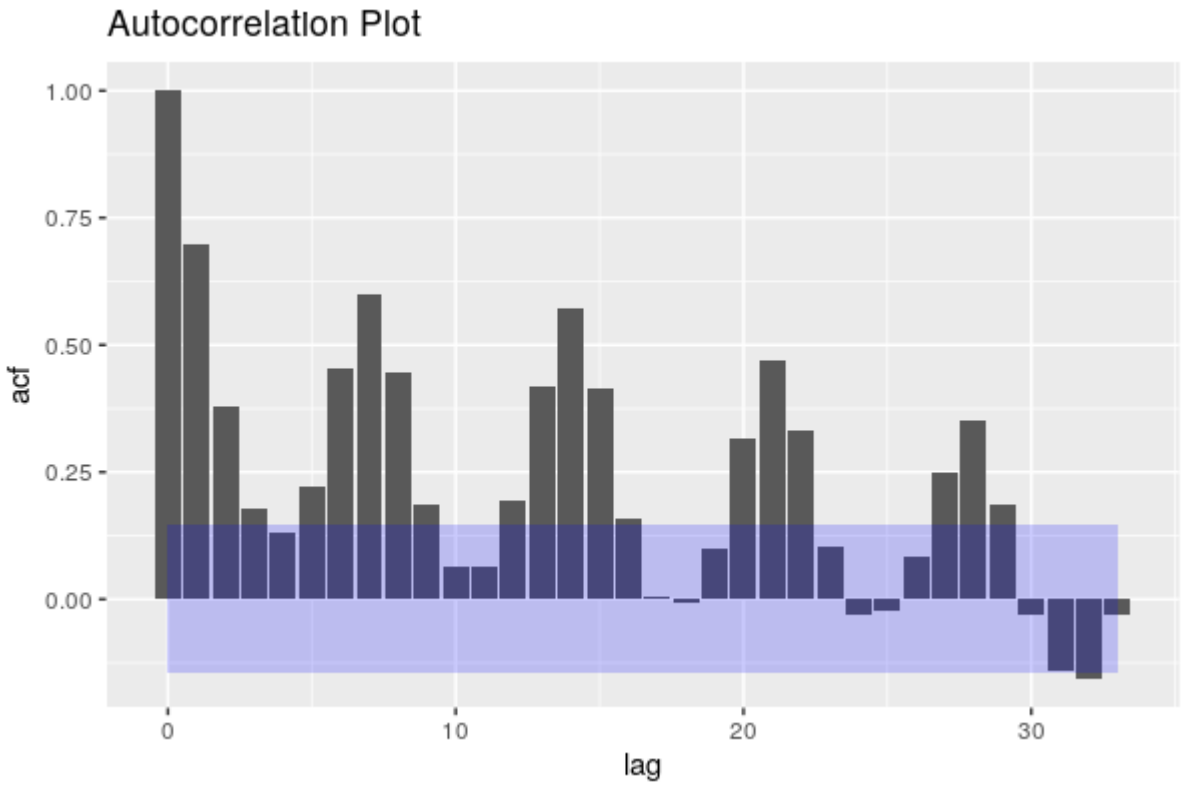


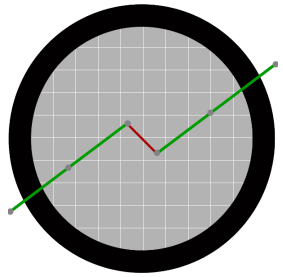




AUTOCORRELATION PLOTS

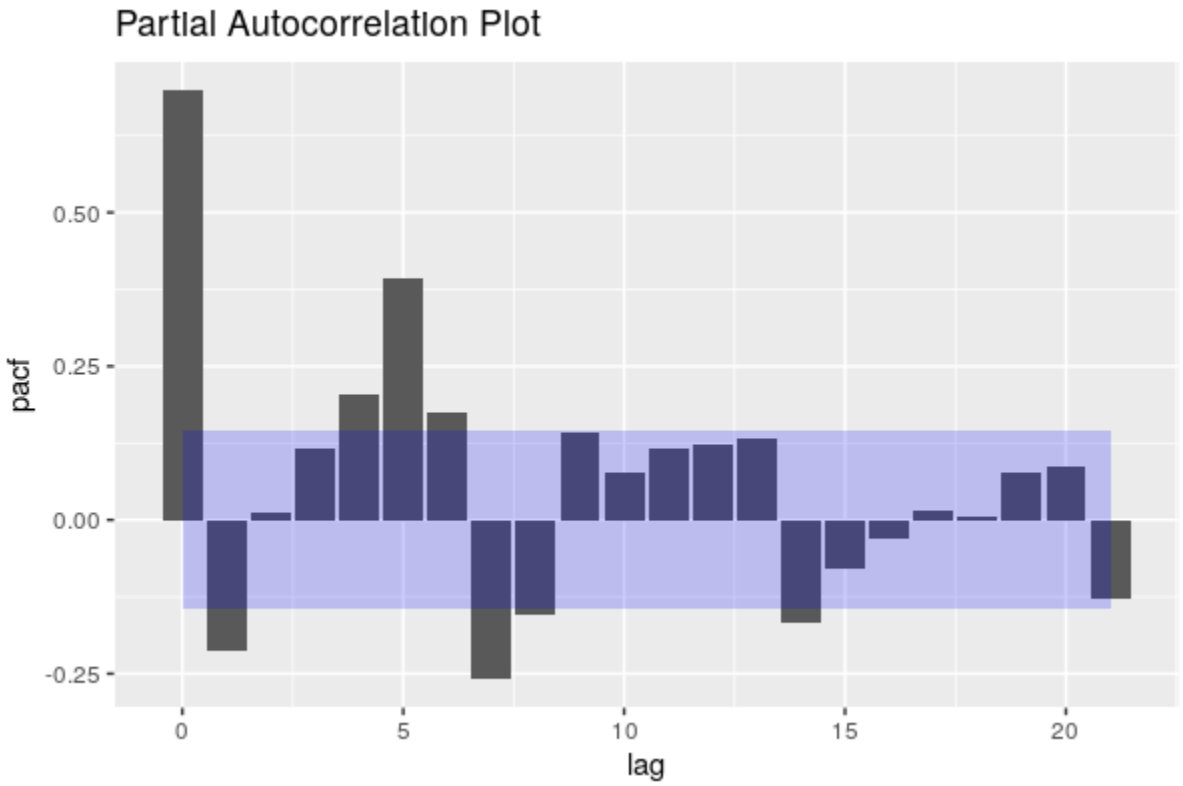
- A plot of a range of autocorrelations for different orders
  - How far back do we look?
- Find the lag to the right of which the columns are (mostly) within the blue region

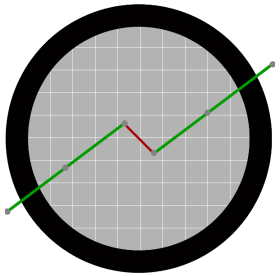




PARTIAL AUTOCORRELATION PLOTS

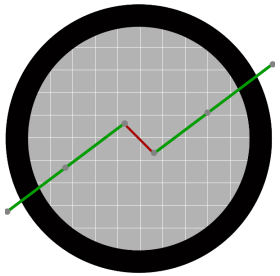
- A plot of a range of partial autocorrelations for different orders
  - How much smoothing do we apply?
- Estimates the moving average component
- Find the lag to the right of which the columns are (mostly) within the blue region





ARIMA COMPONENTS

- Some number of autoregressive components (p)
  - How many? Check the autocorrelation plot!
- A differencing order (d)
  - What's the order? Check the lagged scatterplots!
- Some number of moving average components (q)
  - How many? Check the partial autocorrelation plot!
- Or, you can just experiment!



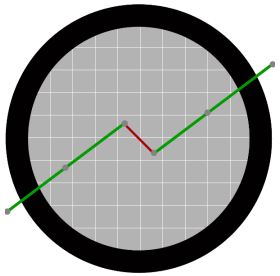
BUILDING THE MODEL

- R

```
arima(Your Time Series, order=c(p, d, q))
```

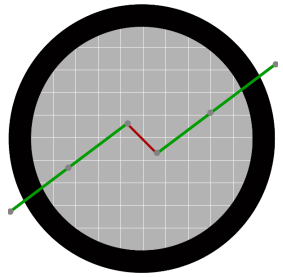
- Python

```
from statsmodels.tsa.statespace import sarimax
sarimax.SARIMAX(Your Time Series, order=(p, d, q))
```



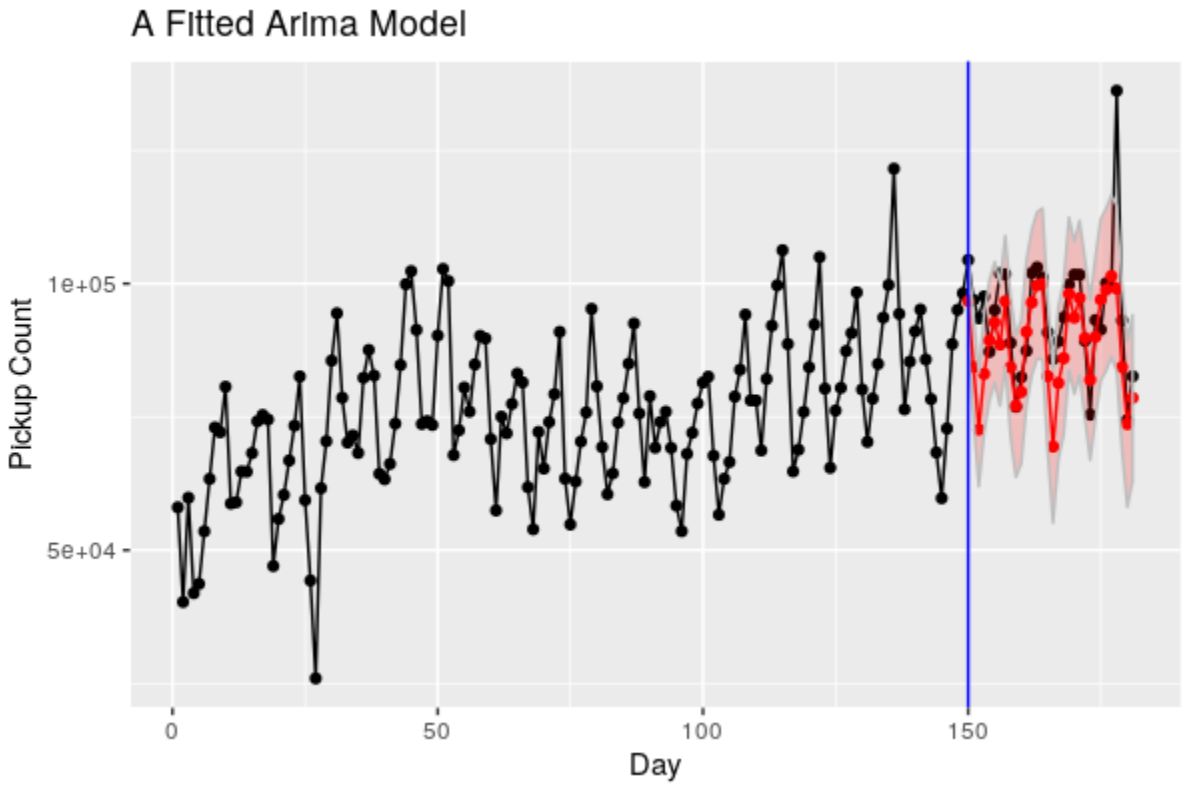
A FEW THINGS ABOUT ARIMA

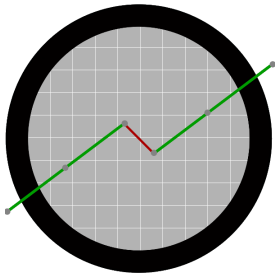
- Modeling is slow, especially when compared to our next class of models
- Figuring out what values to use for the orders is not an exact science
  - Our plots give us a good idea, but it's a good idea to try a few more in the general area
  - Can do a grid search, for example
    - Automatically try a bunch of models and see what works



AN ARIMA MODEL

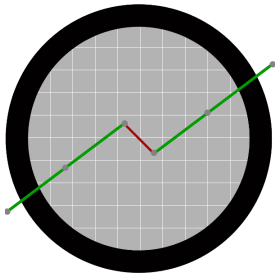
- Mean Relative Error: 5.6%
- Follows peaks nicely!
  - Only major miss is the huge spike





ARIMA IN SUMMARY

- Explore your data with plots
  - Plot by day, day of week, day of month, month, hour, minute, etc.
    - This will help you get a feel for the data
  - Try out some lags, moving averages, and detrending
  - If you can get a stationary data series, remember what you did to get there!
- Estimate parameters
  - Use (partial)autocorrelation function to figure out the order of moving average and autoregressive components
  - Use lagged scatterplots to deal with
- Model, forecast, and repeat!

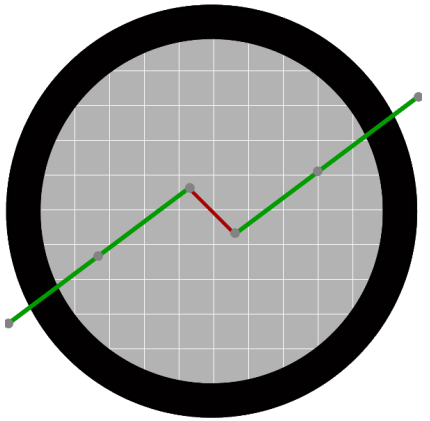


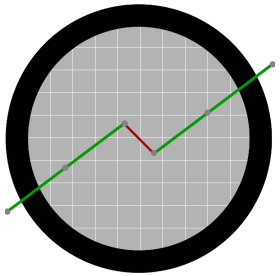
ARIMA EXTENSIONS

- Won’t explore these here
- Seasonal ARIMA: better support for seasonal cycles (longer range)
  - Even more parameters to figure out
- ARIMAX: use external variables to help prediction
  - Maybe if we knew the unemployment rate, that would help us predict inflation
  - Usually pretty simple
- SARIMAX: both!

PROPHET MODELS

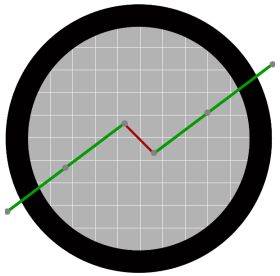
Facebook's in-house forecasting tool





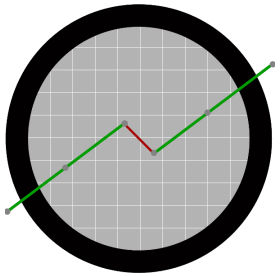
PROPHET MODELS

- Cutting up a forecast
- Pros:
  - Fast and scalable
  - Parameters are very easy to understand
  - Deals with holidays very elegantly
  - Splitting models up allows even more flexibility
  - Performs quite well! Results often better than ARIMA
- Cons:
  - Not as widely used (released in Sept. 2017)!
  - The jury's still out on this one



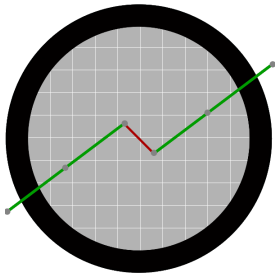
CHANGEPOINTS

- Segment data into various sections and model them differently
- Why? Maybe things actually have changed
  - In business, there can be some major shakeups
  - It doesn’t always make sense to predict based on what happened twenty years ago. Is that data even still applicable?
- Automatically detected by prophet



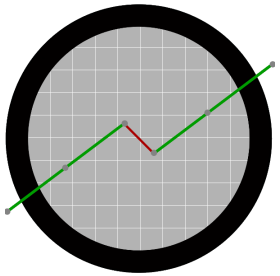
HOLIDAYS

- Large list of holidays
- Maybe things do change on a New Year's Day
  - People are going to want to be picked up after partying, but are drivers going to be working?



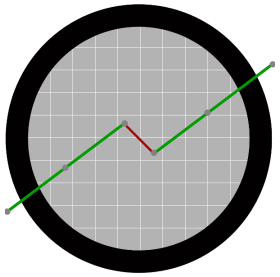
SEASONALITY

- The cyclic components we noted earlier
  - Do things repeat every day? week? hour?
- Extremely easy to deal with in prophet
- Just enable the seasonalities in the modeling call



SOME ANNOYING CONVENTIONS

- Prophet requires that your data be arranged in a very specific way
  - ARIMA doesn't
- Rename your date column to 'ds' and convert it to a datetime format
- Rename the variable you want to model to 'y'



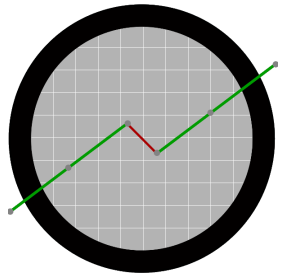
BUILDING THE MODEL

- R

```
library(prophet)
prophet(dataframe, yearly.seasonality=TRUE, …)
```

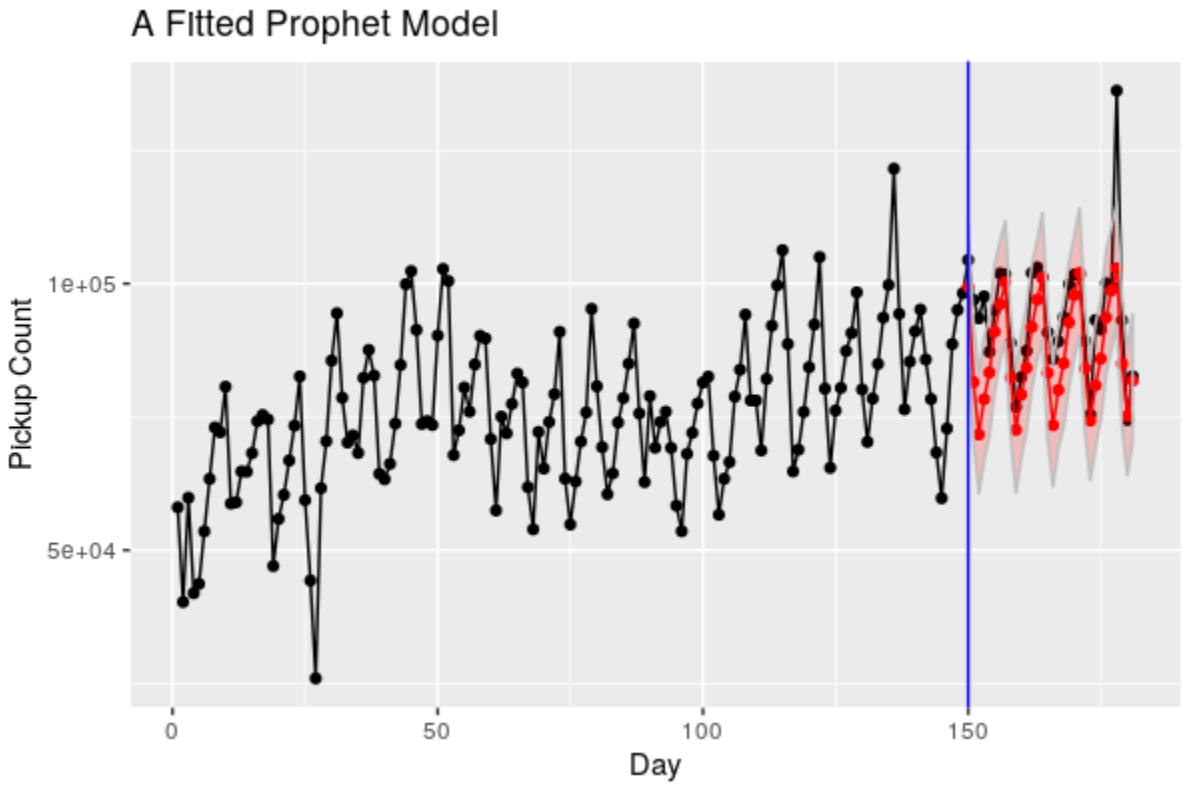
- Python

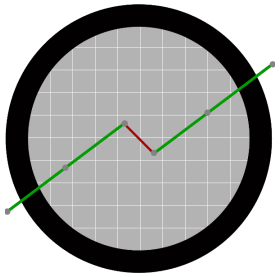
```
import fbprophet
fbprophet.Prophet(dataframe,
    yearly_seasonality=TRUE, …)
```



A PROPHET MODEL

- Mean Relative Error: 7.4%
- Very interested in cycles
- Not bad, but slightly worse than ARIMA
  - Much faster, though





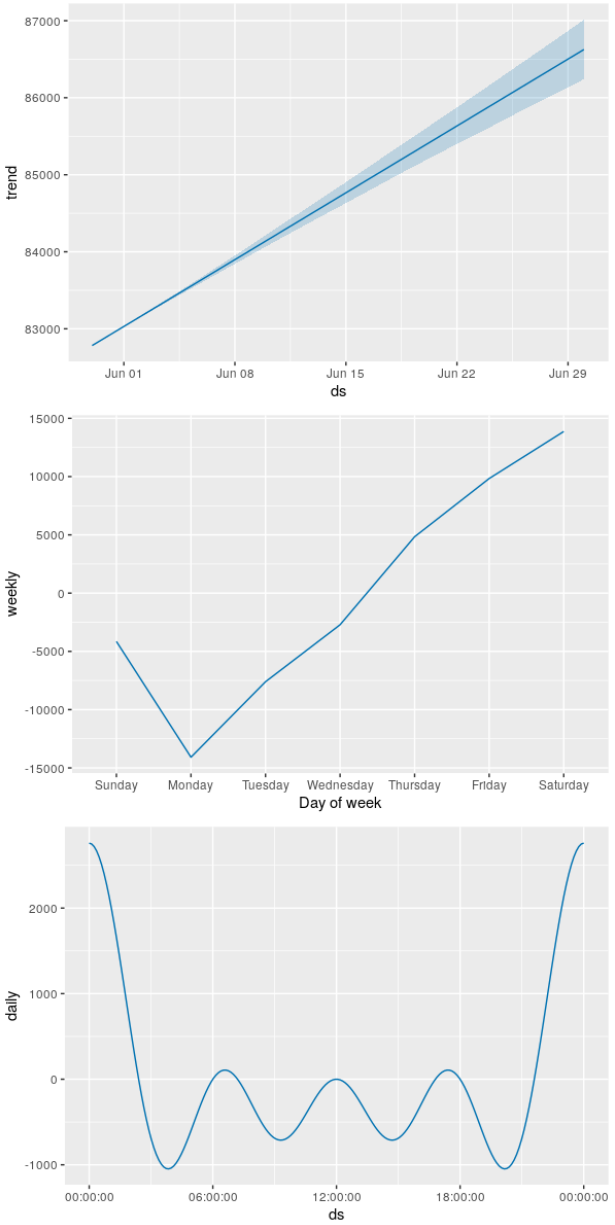
DECOMPOSING

- Look at individual components of the model
  - trend, weekly and daily seasonality
- R

`prophet_plot_components`

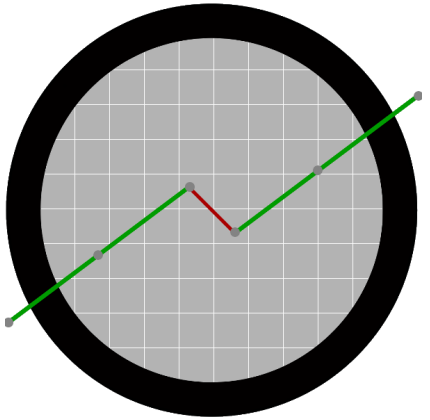
- Python

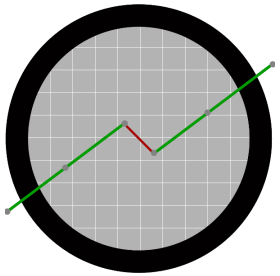
`model.plot_components`



NEURAL NETWORK APPROACHES

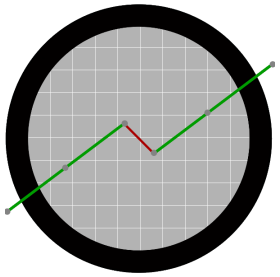
Breaking out the big guns





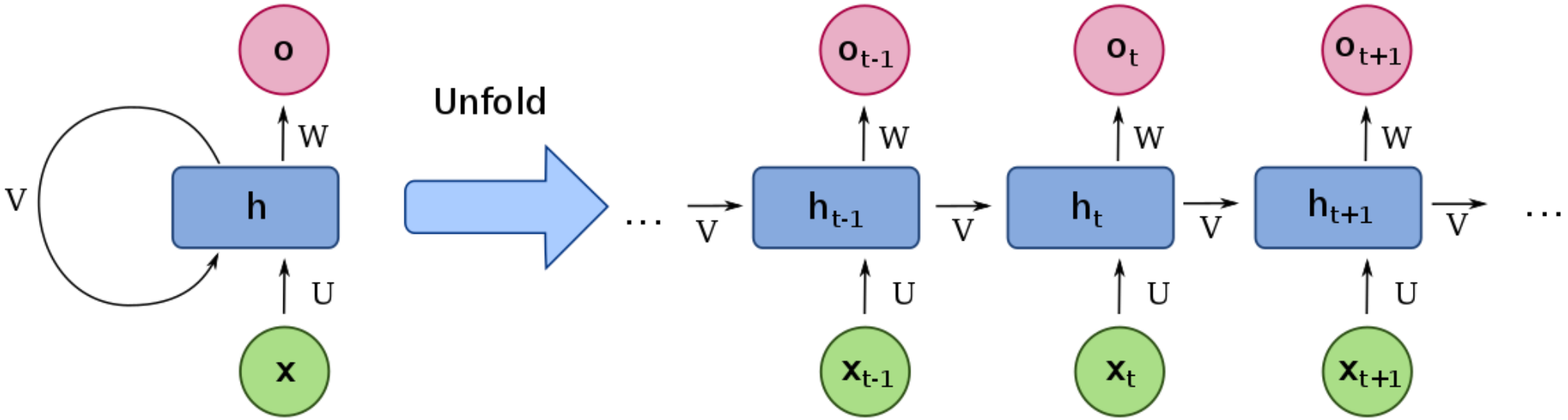
NEURAL NETWORK MODELS

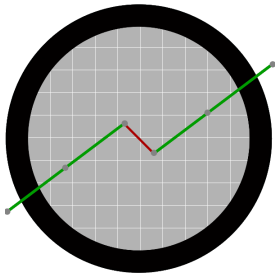
- Incomprehensible but extremely powerful
- Pros:
  - Often extremely accurate
- Cons:
  - More or less impossible to understand what any of the parameters mean
  - Can take a very long time to train the model (can be days)
  - Lots of moving parts and things that can go wrong
- But depending on the problem, that one pro may outweigh all the cons



RECURRENT NEURAL NETWORKS

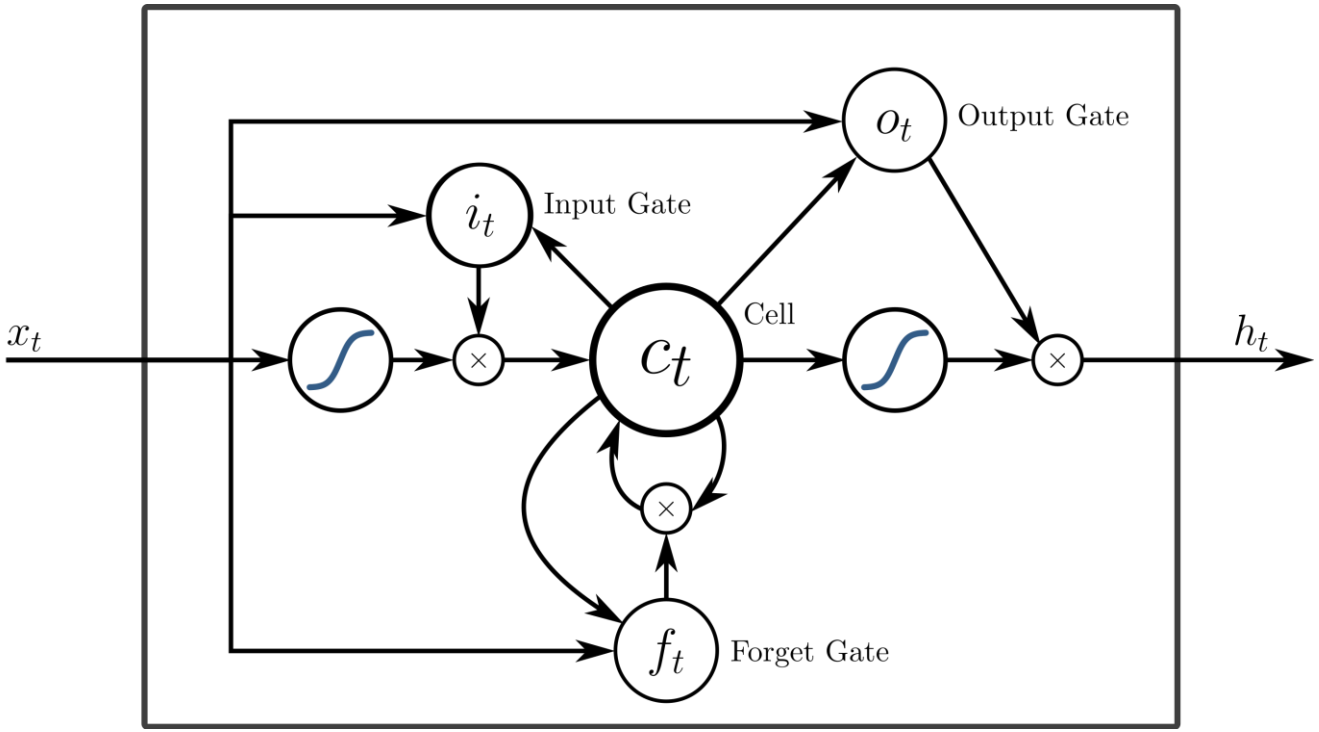
- Simplest time series applicable neural network model
- Neural net that feeds back into itself

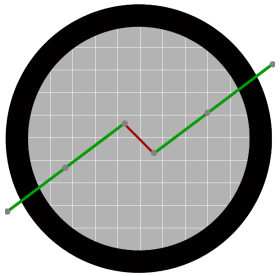




LONG SHORT-TERM MEMORY

- Cells and gates
  - Forget gate
  - Input gate
  - Output gate
- Basically, LSTM models ‘learn’ how long they need to pay attention
- Don’t worry about the details!





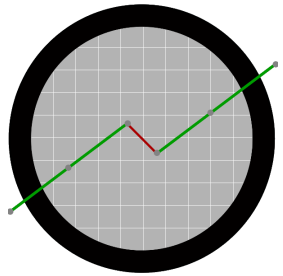
BUILDING AN RNN MODEL

- R

```
library(keras)
keras_model_sequential() %>%
    layer_simple_rnn(units) …
```

- Python

```
import keras.models
model = models.Sequential([
    SimpleRNN(units) …
])
```



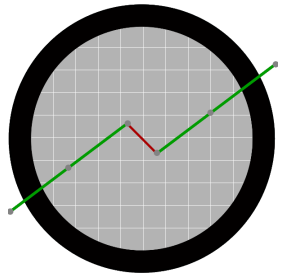
BUILDING AN LSTM MODEL

- R

```
library(keras)
keras_model_sequential() %>%
    layer_lstm(units) …
```

- Python

```
import keras.models
model = models.Sequential([
    LSTM(units) …
])
```

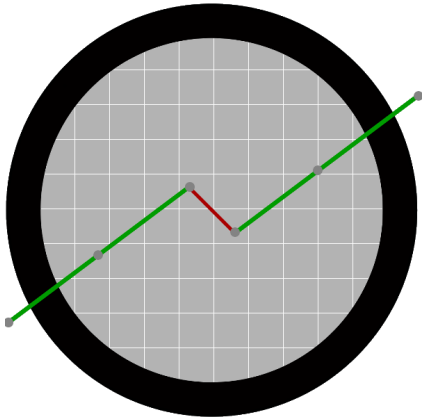


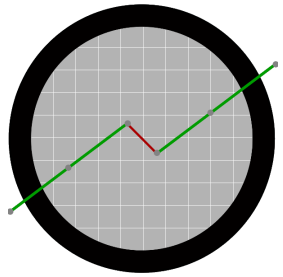
LSTM HIGHLIGHT

- If given enough data, an LSTM can produce ridiculously good results.
- Here's the best model I made on the Dow Jones data:

COMPARISON AND SUMMARY

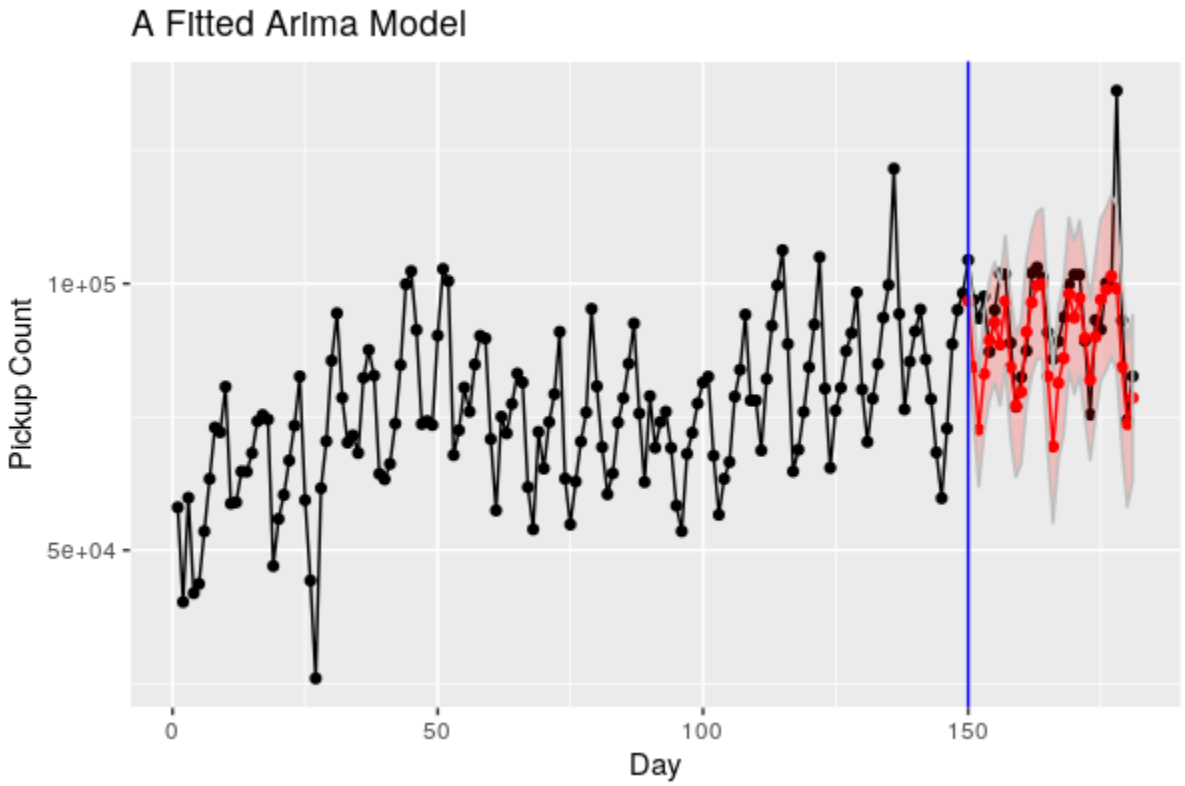
A look back at our models

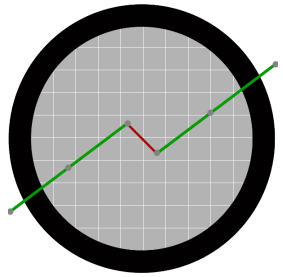




MODEL COMPARISON: ARIMA

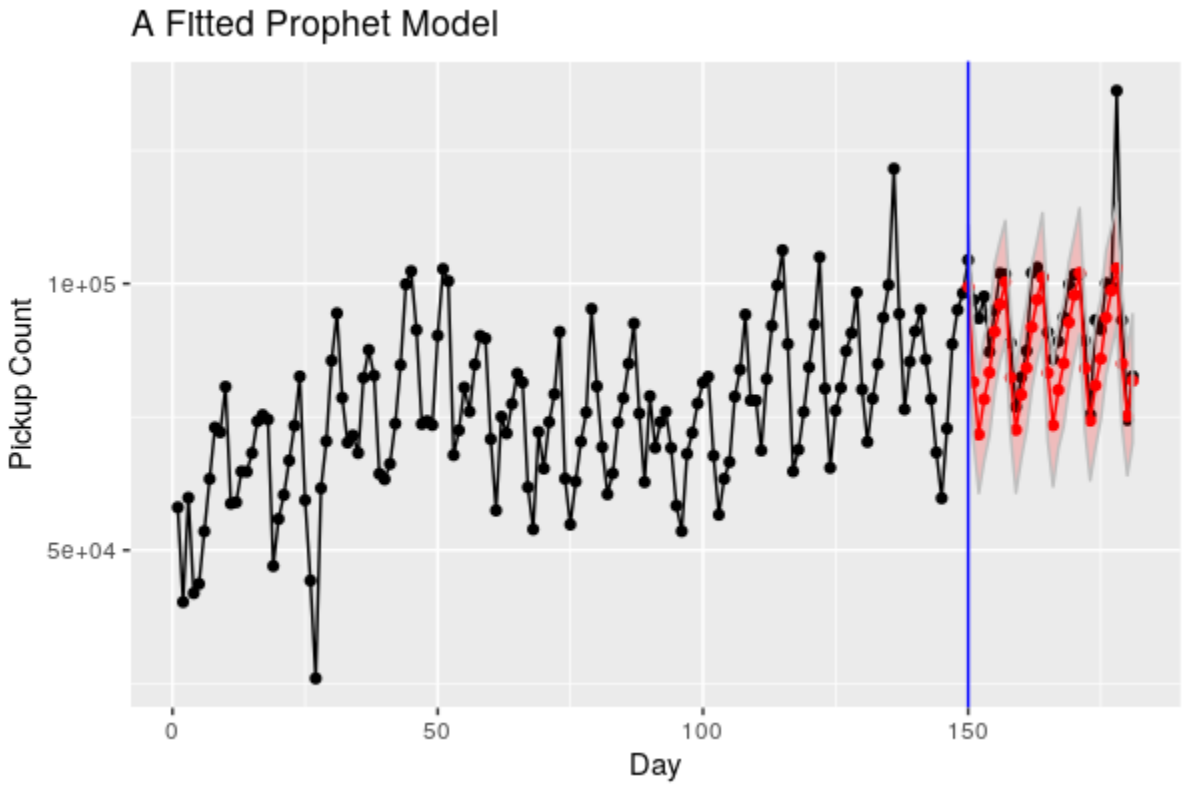
- Mean Relative Error: 5.6%
- Follows peaks nicely!
  - Only major miss is the huge spike

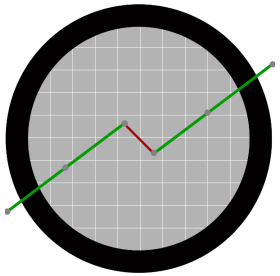




MODEL COMPARISON: PROPHET

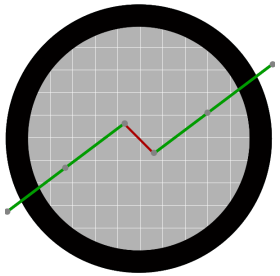
- Mean Relative Error: 7.4%
- Very interested in cycles
- Not bad, but slightly worse than ARIMA
  - Much faster, though





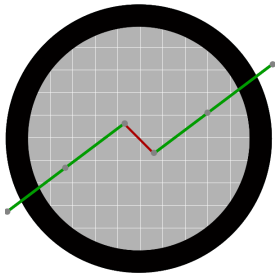
MODEL STRENGTHS

- ARIMA
  - Flexible, interpretable, not too complicated to estimate parameters, easy to tweak
- Prophet
  - Great for business-style forecasting (Facebook's interest), fast, interpretable, holidays, longer seasonality
- Simple RNN
  - Somewhat less complicated than LSTM
- LSTM
  - Highest ceiling, extreme flexibility



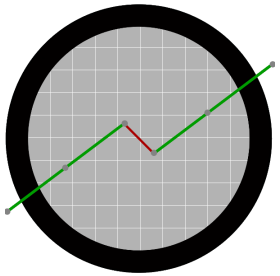
MODEL WEAKNESSES

- ARIMA
  - Bad at longer seasonal trends, never forgets
- Prophet
  - Annoying interface, not a mature package
- Simple RNN
  - All the drawbacks of LSTM, but also less flexibility
- LSTM
  - Can be very finicky to train, uninterpretable, extremely slow and resource-intensive, requires tons of data



MY PERSONAL ADVICE

- Visualize your data extensively
  - Lots of great insights can be found
  - Lagged scatterplots, plots by day/week/hour of day, (partial)-autocorrelation plots
- Make it stationary
  - This will help you understand the data more
- Start with a Prophet model
  - Quick and fast
- Try out a few ARIMA models
  - Tweak the parameters a little bit
- If, and only if, these models don’t meet your needs, build and tune an LSTM



ACKNOWLEDGMENTS

- Plots were created using the ggplot2 package in R
- Shumway and Stoffer's *Time Series Analysis and Its Applications With R Examples, Fourth Edition* published by Springer was used as the main reference for the ARIMA section
- *Forecasting at Scale* by Sean J. Taylor and Benjamin Letham was used as the main reference for the Prophet section
- The pictures under Neural Networks are licensed for free use under Wikimedia Commons
- Uber data provided by a Rice faculty member, not made public here
- Dow Jones data (used in the exercises) downloaded from Yahoo! Finance