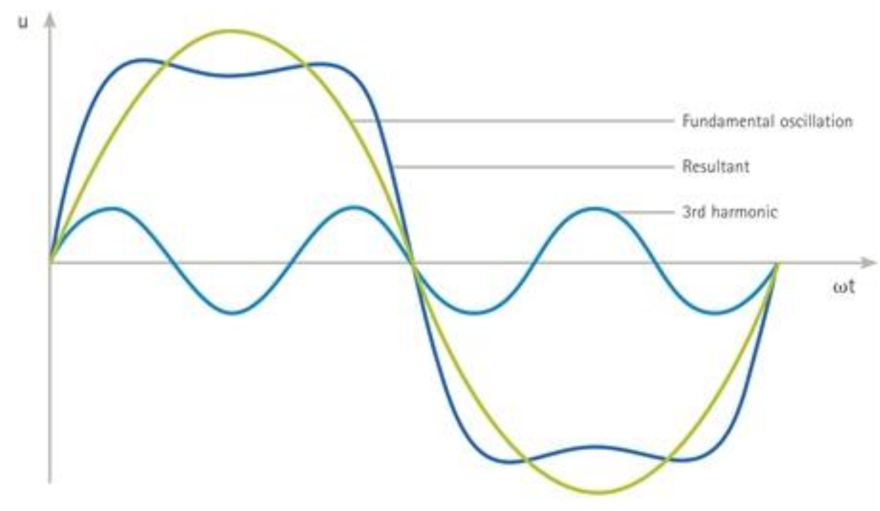
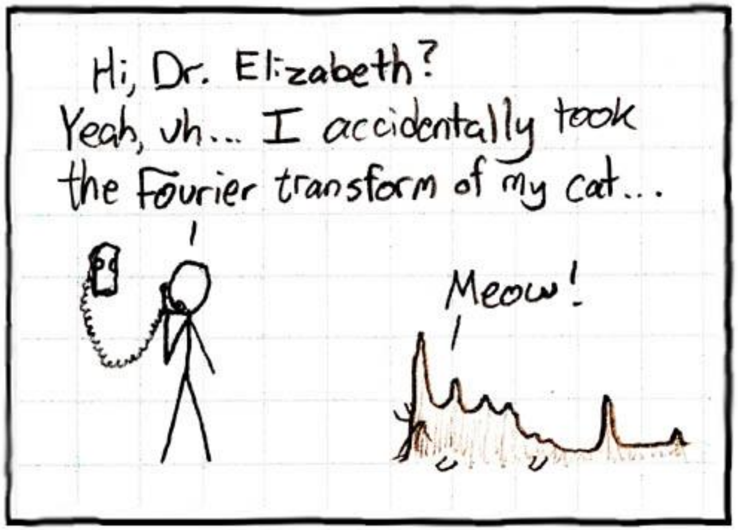
Fourier Analysis for Machine Learning

Presented by: Derek Kane

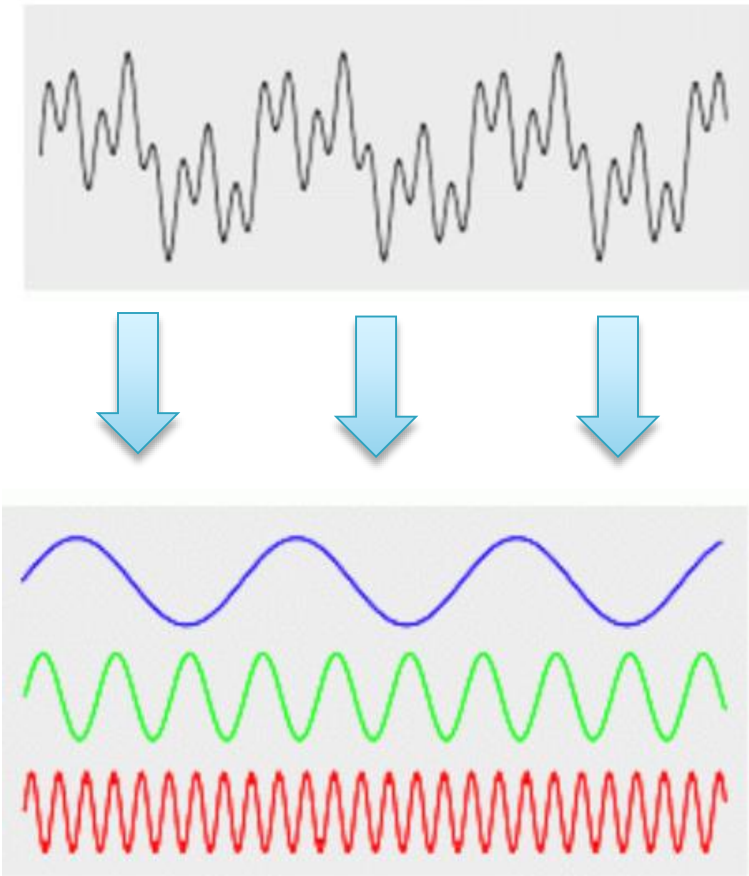
Overview of Topics

- Introduction to Fourier Analysis
- The Fourier Transform
- Econometrics/ Time Series Modeling
- Practical Application Example
  - Manufacturing Order Volume with Artificial Neural Networks





Introduction to Fourier Analysis



- Fourier analysis is a mathematical method used to break down and transform a periodic function into a set of simpler functions.
- These simpler functions can then be summed and transformed back into the original form.
- A period function is a mathematical relationship between a quantity and a variable or variables whose relative values consistently repeat over some regular period of time.

Introduction to Fourier Analysis

- Invented in the early 19th century, French physicist and mathematician Joseph Fourier transformed the partial differentiation equation representing the propagation of heat into a series of simpler trigonometric wave functions. (Ex. sines and cosines)

- These wave functions could be superimposed to reconstitute the original function, thereby providing a simpler, general solution to the problem.

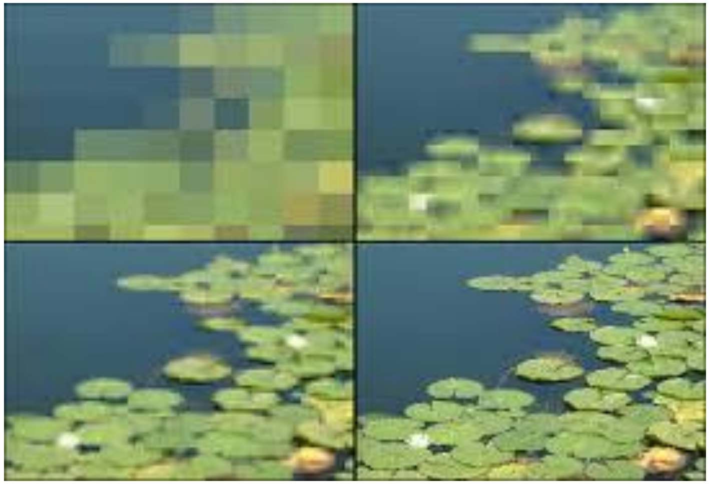
- The Fourier Transform and Fourier's Law were named after his contributions and Joseph Fourier is also generally credited with the discovery of the greenhouse effect.

Joseph Fourier

Fourier Analysis Applications

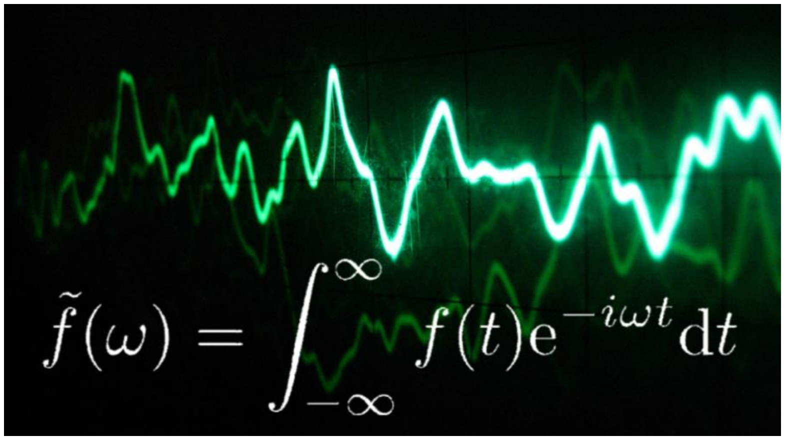
Here are some applications of Fourier Analysis:

- Acoustics
- Signal Processing (.mp3 / .jpeg)
- Option Pricing
- Number Theory
- Statistics / Probability Theory
- Harmonic Analysis
- Image Processing
- Electrical Engineering



Note: Our exploration of the Fourier Analysis/ Fourier Transform will be more conceptual in nature to prepare for the practical application of the technique. There will be less emphasis on the mathematical concepts.

Why is the Fourier Transform so great?



$$\nabla \cdot \mathbf{D} = \rho_V$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

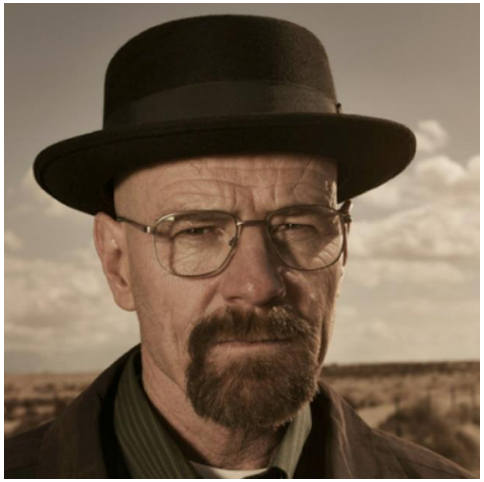
$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}$$

- At the heart of this seemingly intimidating mathematically formula is a beautiful concept.
- This function is essentially <u>transforms</u> a function of time into a function of frequency.
- This "transformation" has a profound impact on the nature of reality itself, due to the relationship it draws between 2 variables.
- This equation is the mathematical bridge between time and frequency, much like Maxwell's equations governing electricity and magnetism.

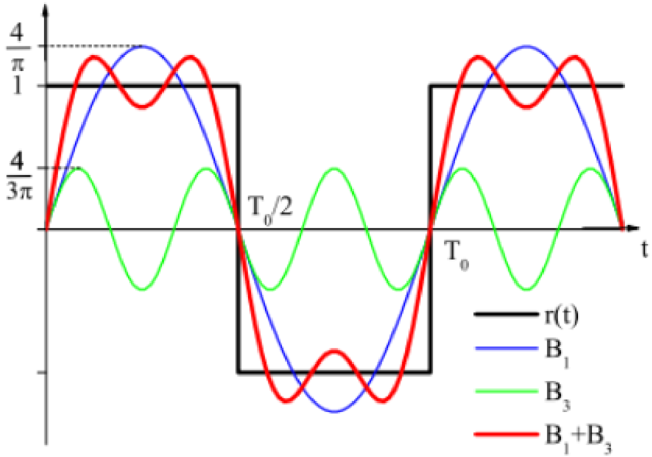
Why is the Fourier Transform so great?

- One of the weirdest results in quantum physics is the Heisenberg Uncertainty Principle.
- This principle states that for a given particle the more precisely its position is defined, the more uncertain its momentum is and vice versa.
- In quantum mechanics, frequency is interchangeable with energy and therefore the energy of a particle is uncertain over arbitrarily small time-frames.
- This allows particles in the quantum regime to 'borrow' enough energy to tunnel through a potential barrier so long as they pay it back in a small enough timeframe to be in keeping with Heisenberg.
- The Heisenberg Uncertainty Principle is essentially a theorem about Fourier Transformations.

$$\sigma_x \sigma_p \geq \frac{\hbar}{2}$$



The Fourier Transform



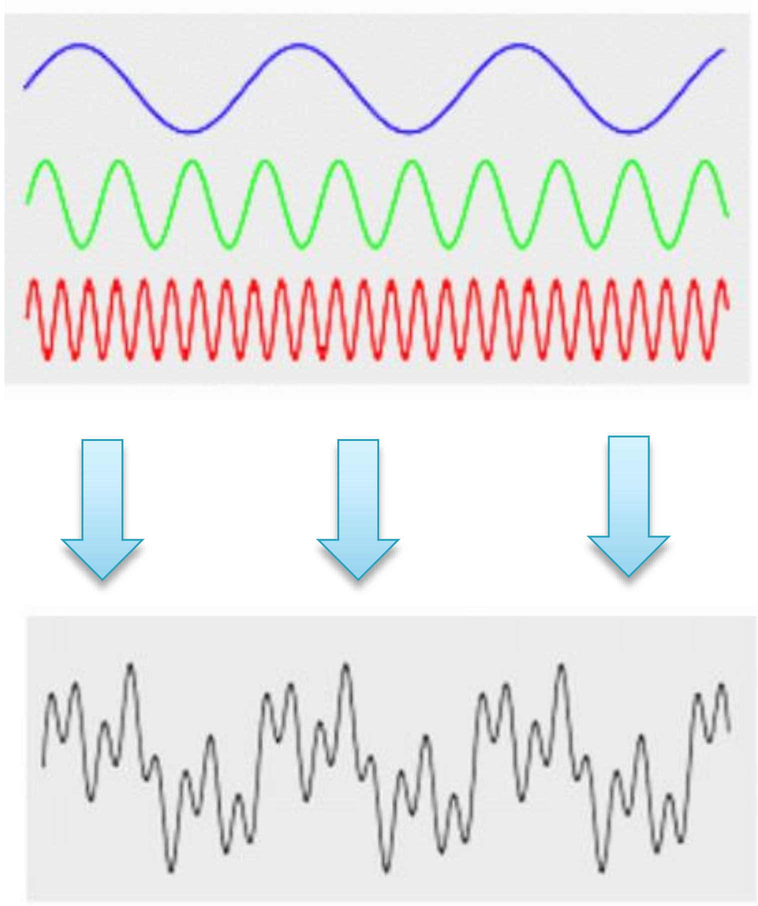
- A Fourier Analysis begins with a Fourier transform.
- The Fourier Transform breaks down, or decomposes, a single, more complicated periodic wave function into a set of simpler elements called a "Fourier Series" that takes the form of sine and cosine waves or complex exponential equations.
- These can then be solved using simpler mathematics and superimposed, or recombined, to yield a solution to the original function via linear combination.
- The decomposed elements in a Fourier Series are sometimes referred to as "harmonics".

The Fourier Transform

- Narrowly defined, Fourier Analysis refers to the process of decomposing the original function into a series of simpler components.

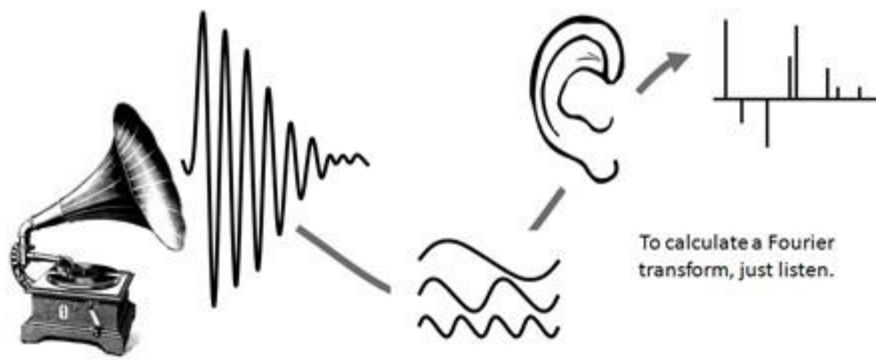
- More generally, it can also include Fourier Synthesis, the process by which the original function is reconstituted by performing an inverse transform.

- Fourier Synthesis essentially runs the Fourier Analysis in reverse.



The Fourier Transform

- Fourier Transformations are like musical notation. We can sing a song from memory but we can also write down what to play.
- If you play the right keys together, it sounds exactly like the original song.
- In other words, we are adding up frequencies to recreate the original waveform.
- Musical notation is what you might call a time resolved Fourier Transform. You're creating slices in time, and at each time step, you are specifying the frequency spectrum (the chords).

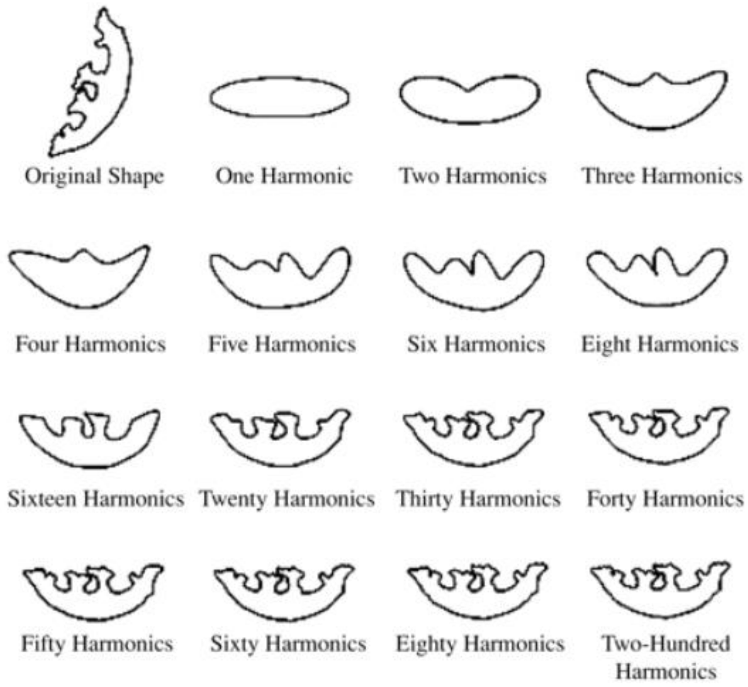


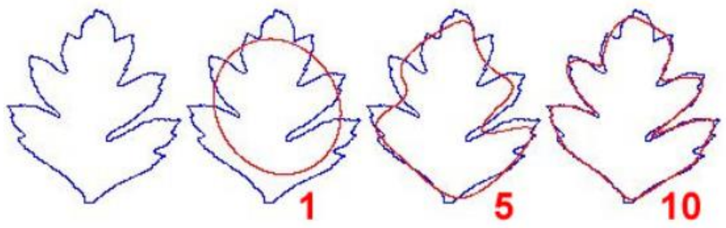
The Fourier Transform

- Another example which fits this idea is related to cooking recipes.
- We can describe a dish by what it tastes like.
- However, we can also describe it by the ingredients used to cook the dish.
- If we add the right ingredients in the right amount, you can recreate the taste.

The Fourier Transform

- The Fourier Transform can also be applied to recreate the shapes of 2 dimensional objects:

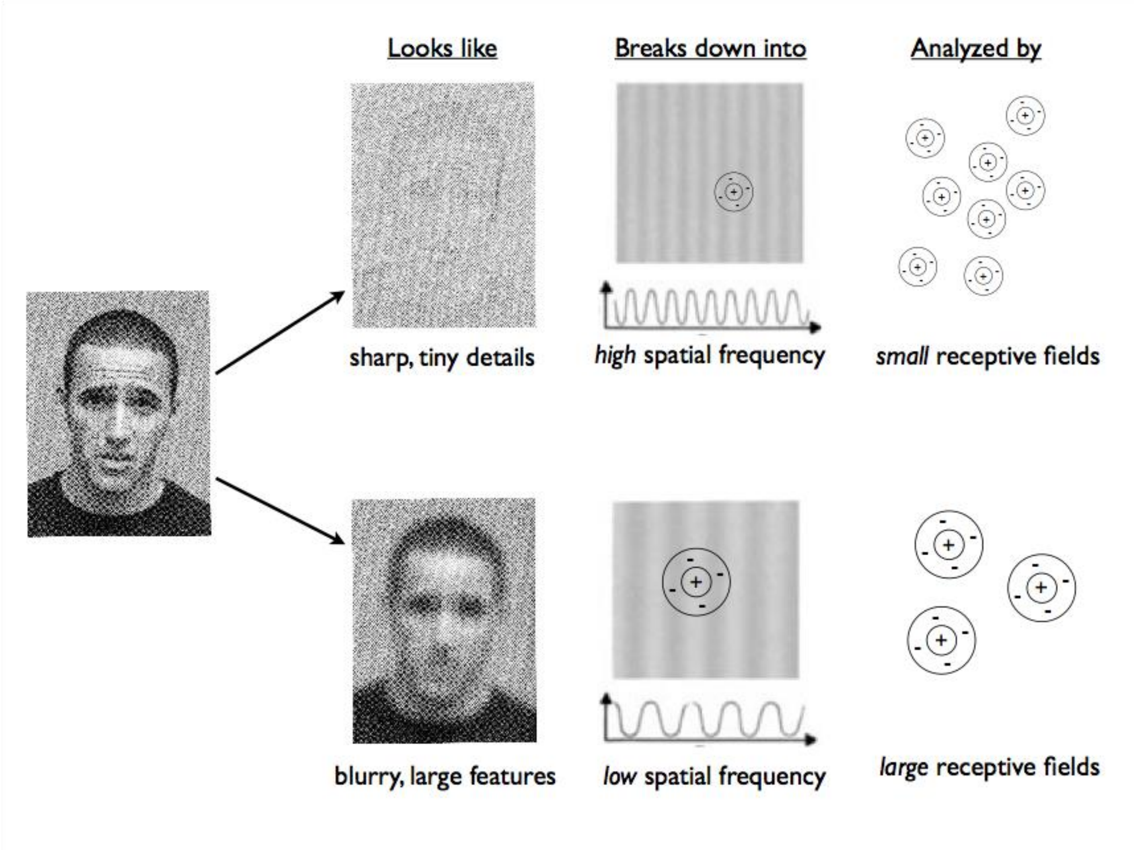




Outline of the leaf at right, summarized using 1, 5, and 10 harmonics (elliptic Fourier analysis carried out)

The Fourier Transform

- Here is a representation of a complex image processing approach through a Fourier Transformation.



The Fourier Transform

Stuart Riffle's elegant explanation in a single sentence. (Discrete Fourier Transform)

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{i2\pi k\frac{n}{N}}$$

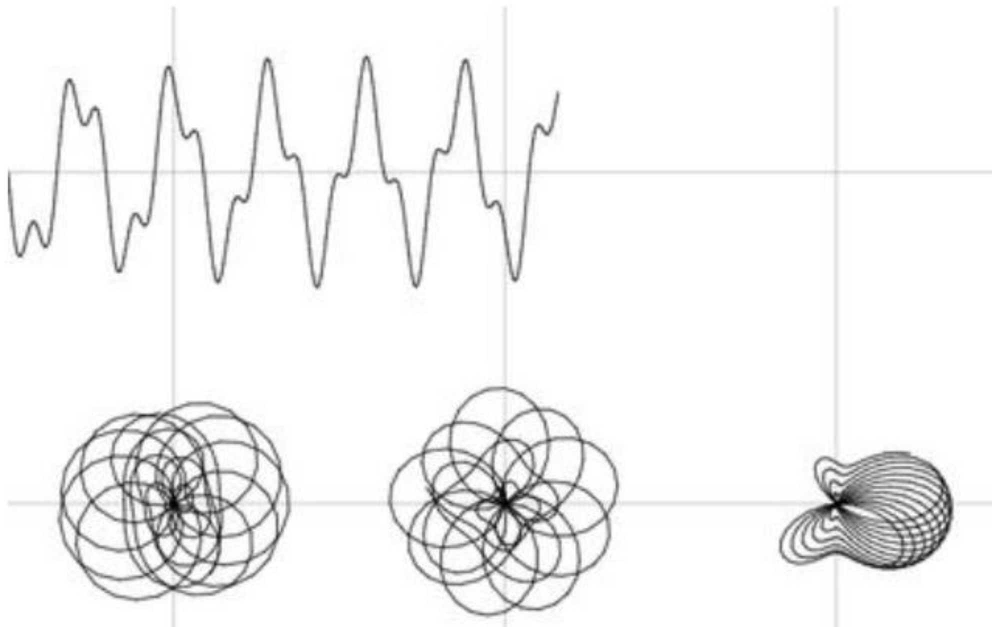
To find the energy at a particular frequency, spin your signal around a circle at that frequency, and average a bunch of points along that path.

The daunting formula involves imaginary numbers and complex summations but the idea is simple.

- Imagine an enormous speaker, mounted on a pole, playing a repeating sound. The speaker is so large, you can see the cone move back and forth with the sound.
- Mark a point on the cone, and now rotate the pole.
- Trace the point from an above-ground view, if the resulting squiggly curve is off-center, then there is frequency corresponding the pole's rotational frequency is represented in the sound.

The Fourier Transform

The upper signal is make up of three frequencies ("notes"), but only the bottom-right squiggle is generated by a rotational frequency matching one of the component frequencies of the signal.



The Fast Fourier Transformation

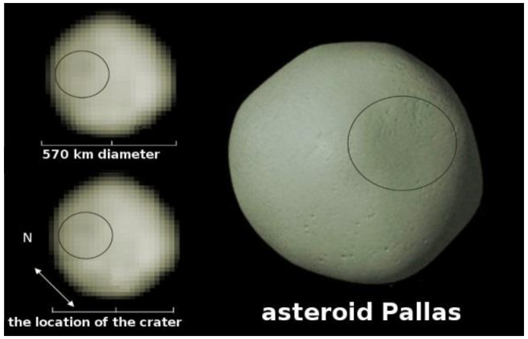
- A Fast Fourier transform (FFT) is an algorithm to compute the Discrete Fourier Transform (DFT) and its inverse.

- An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors.

- As a result, Fast Fourier Transforms are widely used for many applications in engineering, science, and mathematics.

- In 1994 Gilbert Strang described the Fast Fourier Transform as "the most important numerical algorithm of our lifetime“ and it was included in Top 10 Algorithms of 20th Century by the IEEE journal Computing in Science & Engineering.

The Fast Fourier Transformation



- The development of fast algorithm for DFT can be traced to Gauss's unpublished work in 1805 when he needed it to interpolate the orbit of asteroids Pallas and Juno from sample observations.

- His method was very similar to the one published in 1965 by Cooley and Tukey, who are generally credited for the invention of the modern generic FFT algorithm.

- While Gauss's work predated even Fourier's results in 1822, he did not analyze the computation time and eventually used other methods to achieve his goal.

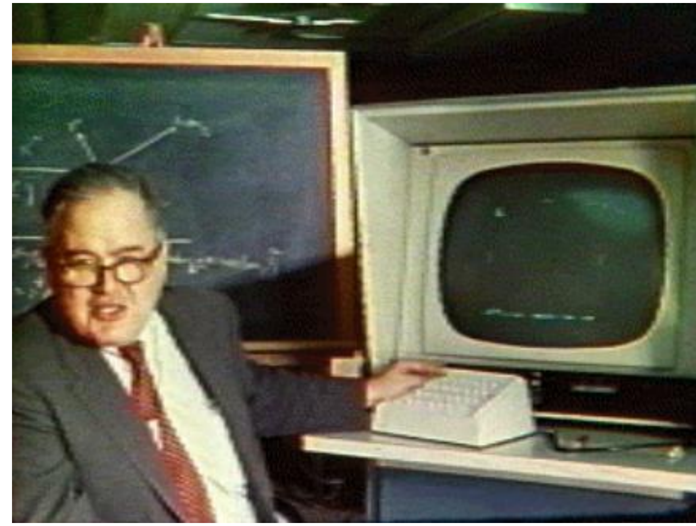
The Fast Fourier Transformation

- John Tukey, who worked at IBM's Watson labs, came up with the idea during a meeting of President Kennedy's Science Advisory Committee where a discussion topic involved detecting nuclear tests from the Soviet Union by setting up sensors that surrounds the country from outside.

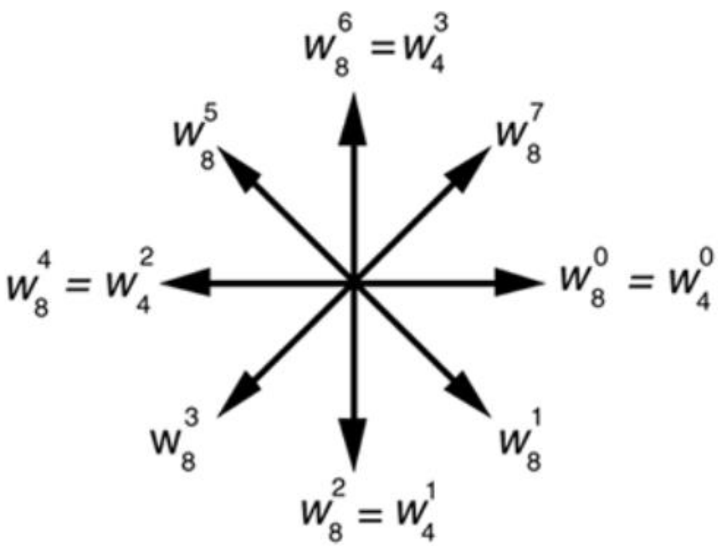
- To analyze the output of these sensors, a Fast Fourier Transform algorithm was needed.

- Tukey's idea was given to Cooley for implementation while hiding the original purpose from Cooley for security reasons.

- As Cooley didn't work at IBM, the patentability of the idea was doubted and the algorithm went to the public domain, which, through the computing revolution of the next decade, made FFT one of the indispensable algorithms in digital signal processing.



The Fast Fourier Transformation



Twiddle Factor

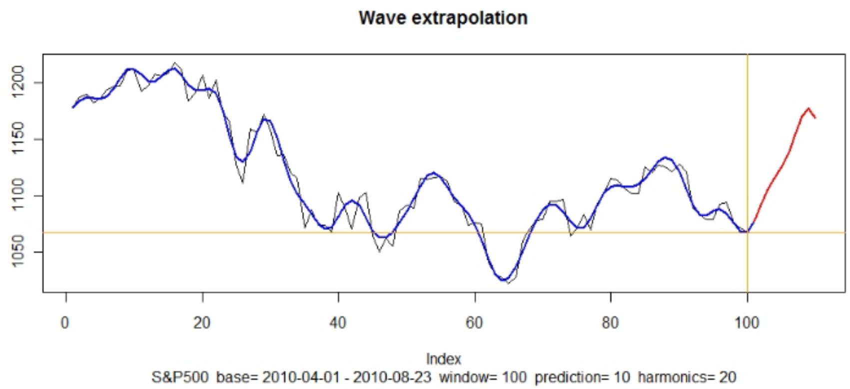
- By far the most commonly used FFT is the Cooley–Tukey algorithm. Although there are many other forms of the Fast Fourier Transformation.
- An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the most important difference is that an FFT is much faster.
- The FFT approach is a divide and conquer algorithm that recursively breaks down a DFT of any composite size N = N1, N2 into many smaller DFT's of sizes N1 and N2, along with O(N) multiplications by complex roots of unity traditionally called twiddle factors (after Gentleman and Sande, 1966).

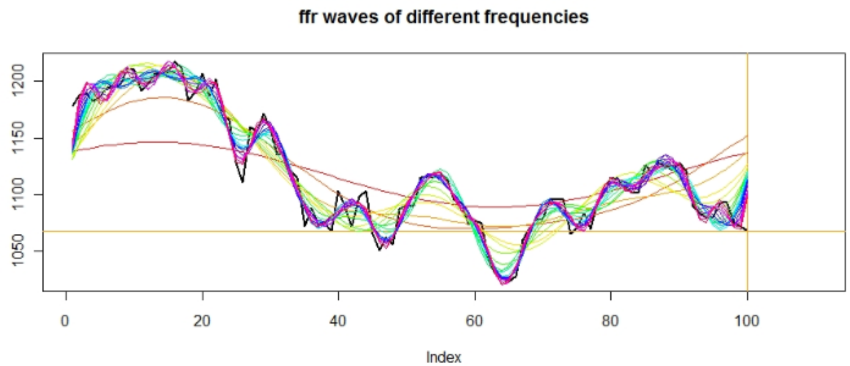
Fourier Analysis and Machine Learning

- Improved, expanded upon, and the core of what has come to be known as the field of harmonic analysis, Fourier Analysis has evolved and progressed to include the study of more abstract and general phenomena.

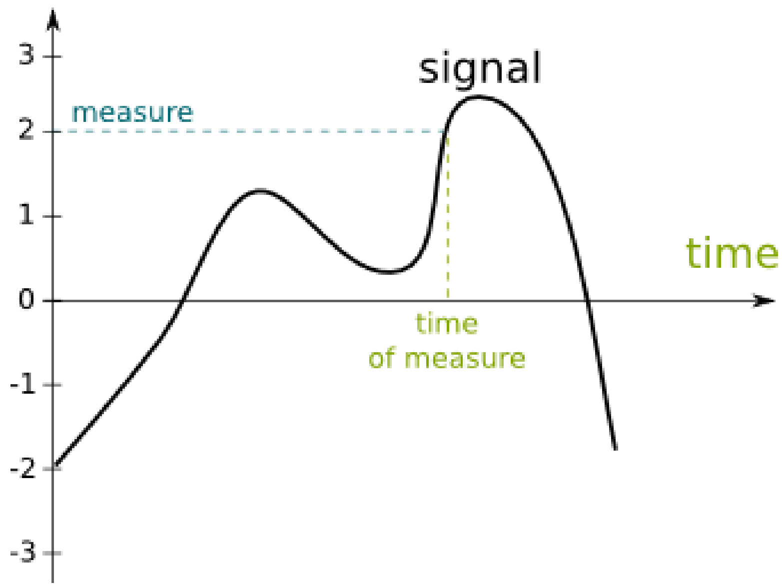
- Fourier Analysis is now used actively, regularly, and widely in econometrics and financial markets theory by researchers and practitioners to forecast.

- Fourier Analysis also is used to analyze and better understand, the nature and behavior of a wide range of time series data and parameters that exhibit non-linear relationships and repeating, wave-like patterns over time.





Fourier Analysis and Machine Learning

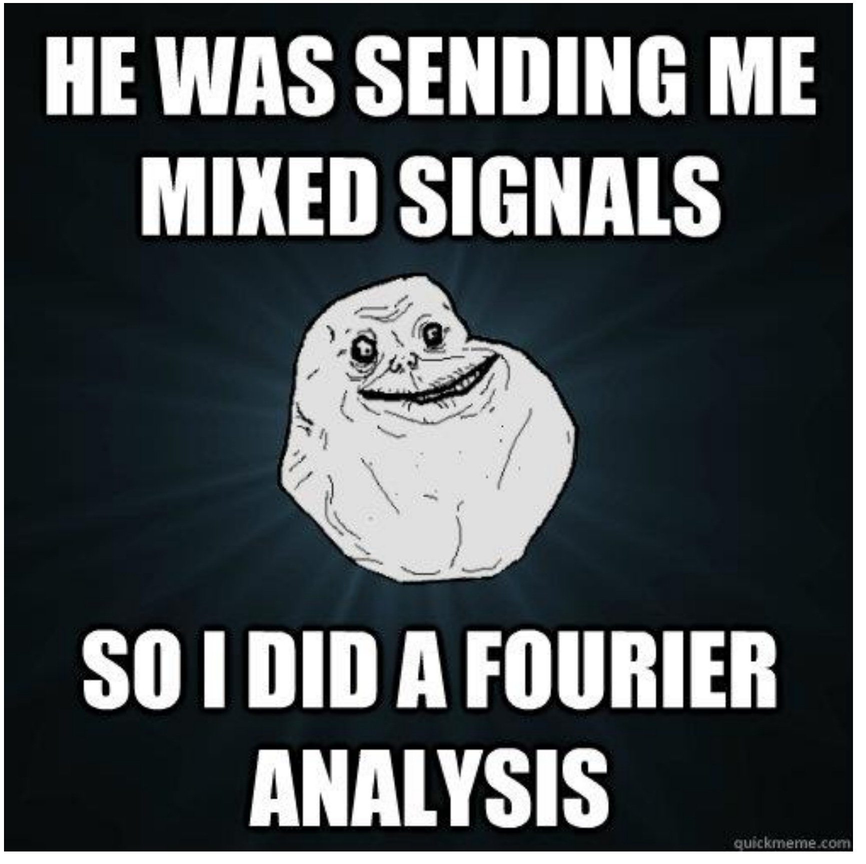


- Among its many applications, it has been used to model long-term economic cycles, the relationship between inflation and the demand for money, and patterns and trends in the stock, foreign exchange, and housing markets, and cycles in the semiconductor industry.

- It is also used as to measure the efficiency of a national economy.

- FFT's are also useful as a high-pass to remove low-frequency periodic variations (e.g. hourly, daily, weekly) and then you can back-transform to the time domain and use that as input for your monitoring tools.

Fourier Analysis and Machine Learning



Practical Example – Manufacturing Order Volume

Manufacturing Order Volume

- Manufacturing is a field where predictive analytics can have a significant impact on operational efficiency and risk.

- Without trustworthy guidance on the expected amount of orders to be fulfilled in the near future, usually a significant stockpile of parts/raw materials and spare (process) capacities are necessary in order to compensate for the variances in the volume of incoming orders.

Our goal is to devise:

- Develop a time series prediction to determine the projected order volume for a manufacturer.
- Utilize the Fast Fourier Transform and artificial neural networks to create the algorithm.

Understanding the data

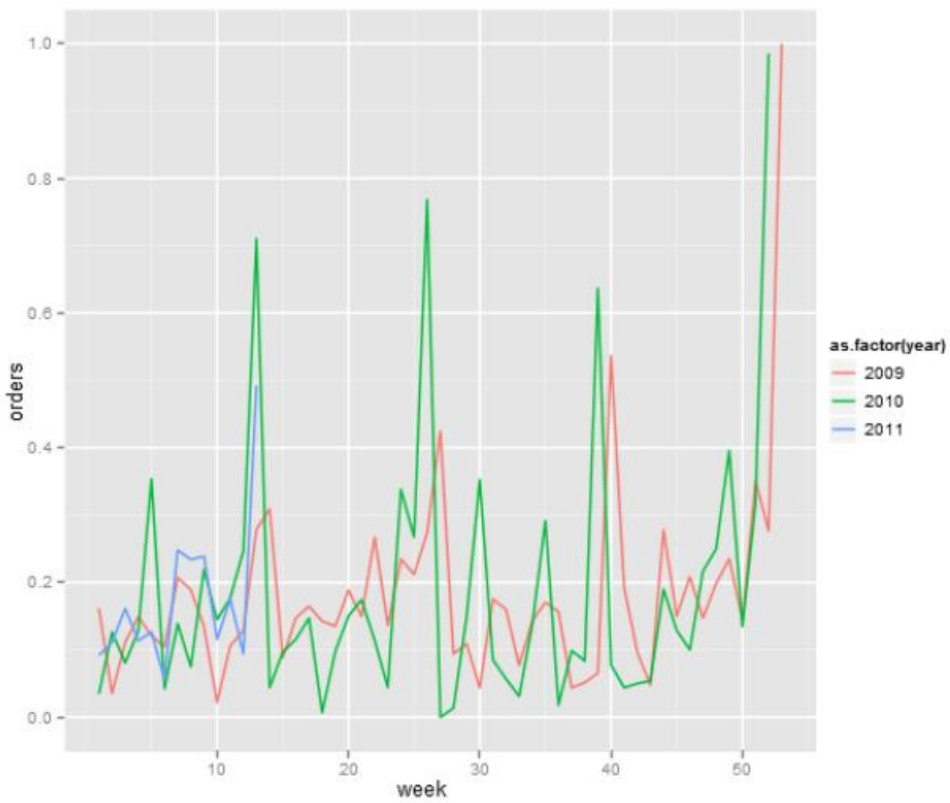
- In order to better prepare the analysis, we must first understand the data we are working with.

year	quarter	week	orders
2009	Q1	1	0.162037
2009	Q1	2	0.0350988
2009	Q1	3	0.1109399
2009	Q1	4	0.14919
2009	Q1	5	0.1209814
2009	Q1	6	0.1044528
2009	Q1	7	0.2074297
2009	Q1	8	0.189102
2009	Q1	9	0.1346347
2009	Q1	10	0.0220552
2009	Q1	11	0.105833
2009	Q1	12	0.129021
2009	Q1	13	0.278845
2009	Q1	14	0.3090167
2009	Q2	15	0.0883416

- We will work on the weekly aggregates of the worldwide amount of orders, normalized into the [0,1] interval.

- The value of the week and quarter columns is relative with respect to the year. The dataset contains values from Q1 2009 through Q1 2011.

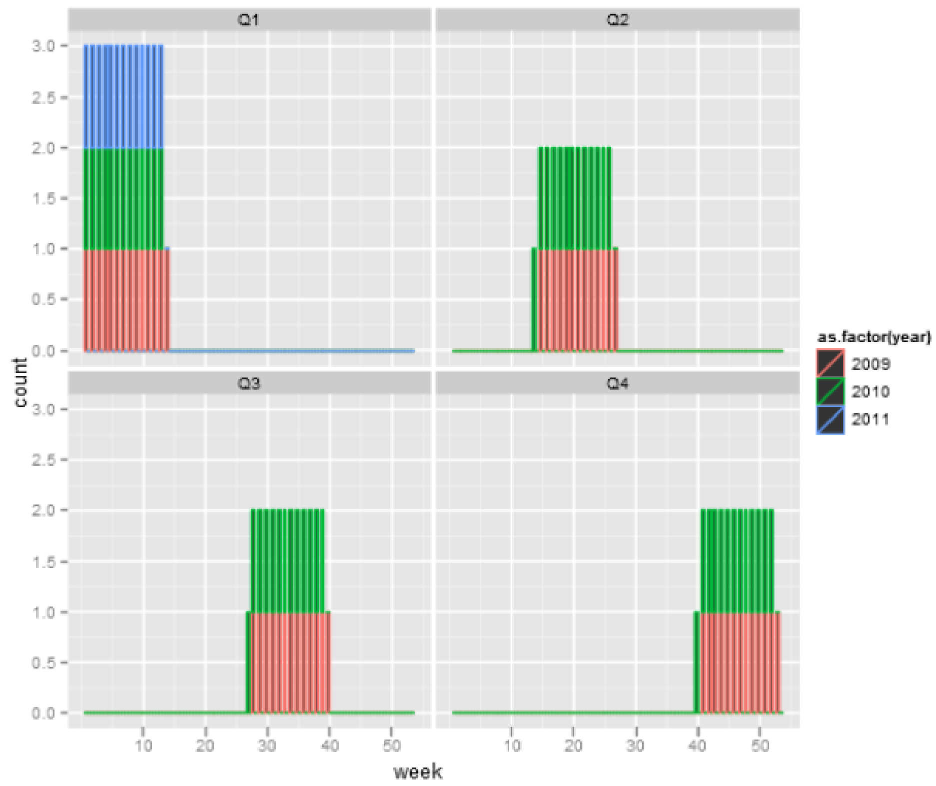
Understanding the data



- We will review the plot of the order volume to see if there is a period frequency of the pattern.
- It is apparent that there is quite much similarity between the two full years and the one full quarter we have data for.
- Also, there seems to be a strong periodicity across the quarters.
- This suggests that there is an underlying structure in the data that can be used for forecasting.

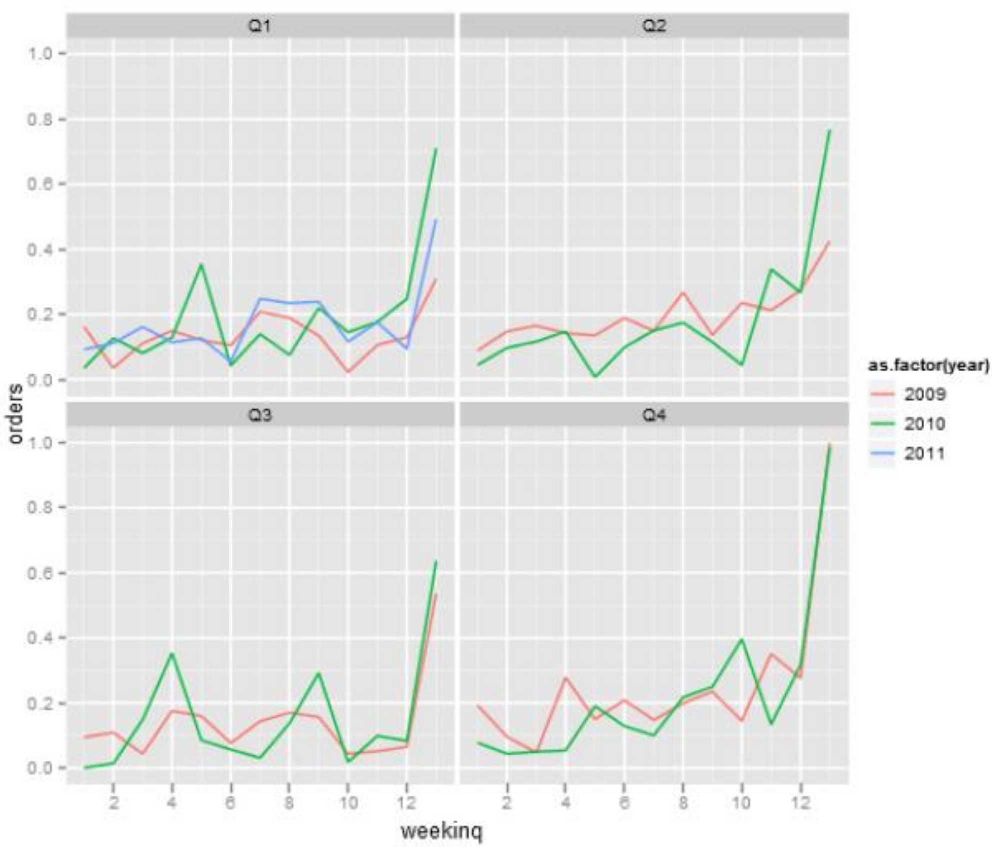
Understanding the data

- However, there seems to be a strange one week difference between the apparent peaks of the two full years.
- A by-quarter histogram of the week values uncovers the cause quickly.
- There was one more business week in the first quarter of 2009 than in the first quarter of 2010 and 2011.
- Consequently, there is a consistent one week difference between the last weeks of the quarters.
- However, due to the sudden surge of orders, from the business point of view this is exactly the week that counts the most.



Understanding the data

The model building approaches we will use later can usually cope with such a one-week offset discrepancy.

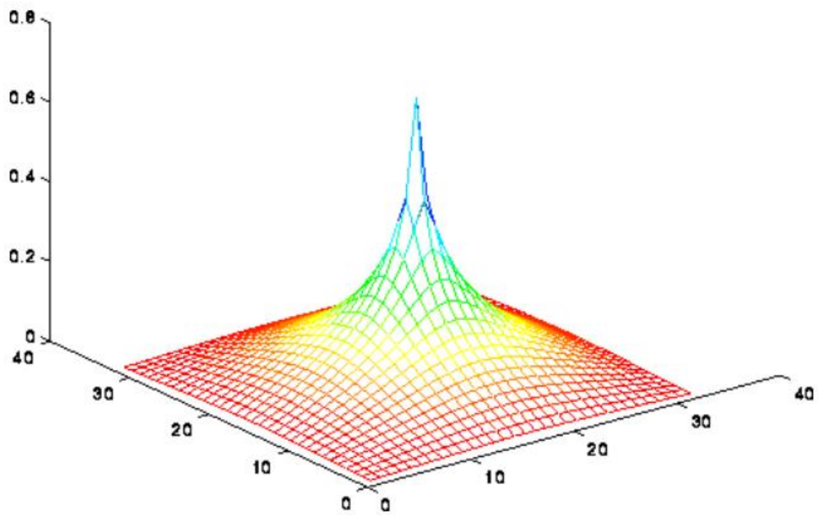


Here is what we can tell now from these plots:

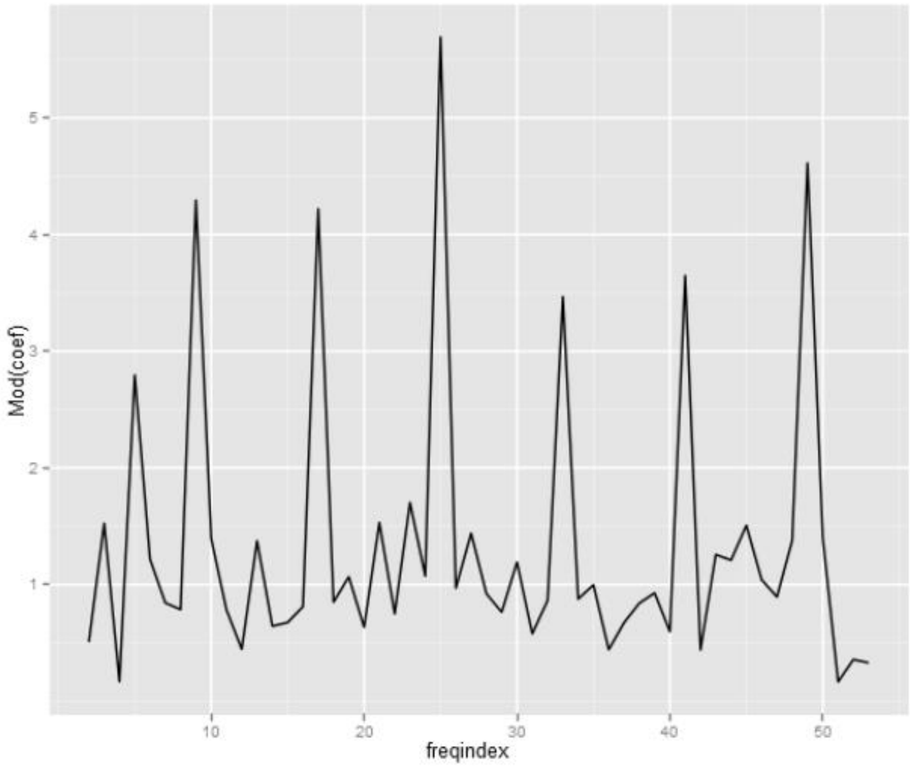
- At the very end of the quarter, there is always a significant spike in the orders; for Q3 and Q4, even the exact amounts are very close.
- During the quarters, there is also significant similarity between the orders.
- On the whole, the time series seems to be stationary and highly periodic; therefore, it should be worthwhile to analyze its characteristics in the frequency domain.

Fast Fourier Transformation

- A visual inspection hinted at a strong periodicity in the time series at the quarter, half year, and year intervals.
- In order to prove this suspicion, we now perform a brief power spectrum analysis of the first two years of the data.
- 2011 Q1 is omitted here; we would like to look at full years to uncover possibly yearly periodicity.
- Also, this analysis will lead us to the training data for the forecasting approach and 2011 Q1 is kept back for testing purposes.



Fast Fourier Transformation



- We apply the common Fast Fourier Transformation, extend the results with a frequency member index and plot the value of the complex modulus w.r.t. the frequency index.
- Note that the zero frequency component is omitted.
- Also, as the input signal is real-valued, it is symmetric in the remaining frequencies; therefore, only the lower half of the spectrum is plotted.
- This results in the following “power spectrum” plot that uses the complex modulus to measure “magnitude”.

Fast Fourier Transformation

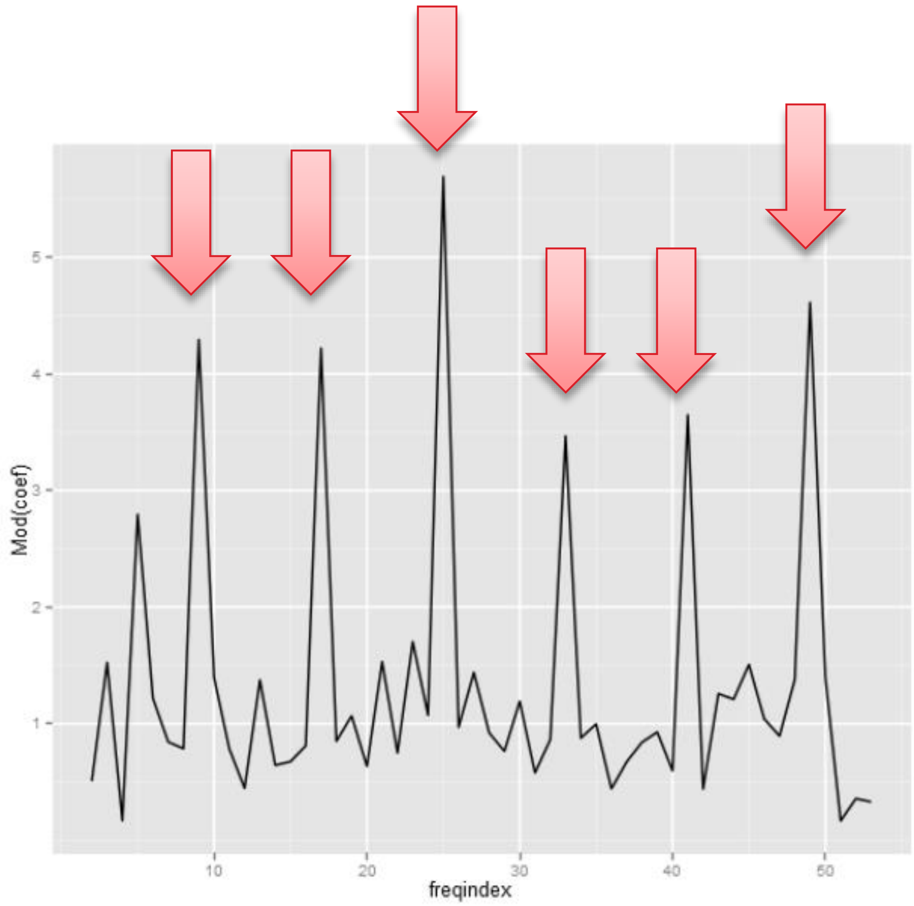
- It is apparent that a number of frequencies have significantly higher amplitude than the others, hinting at a strong underlying periodic nature in the data.

Let's identify these peaks:

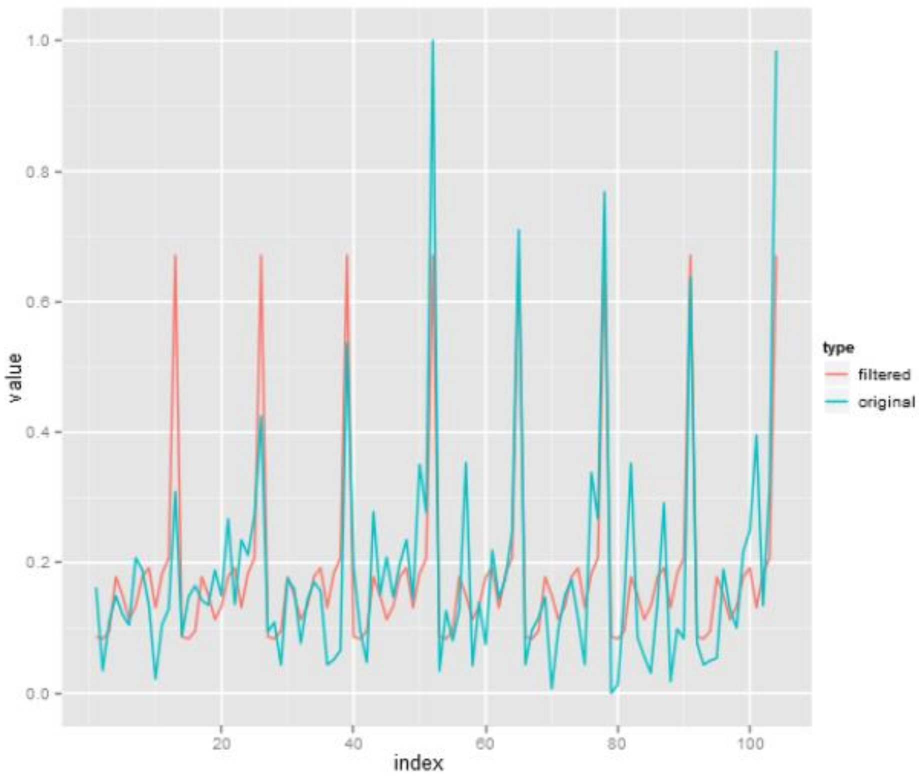
- 0, 8, 16, 24, 32, 40, 48

- What we see is that a component with the frequency of 8 / 104 (2 * 4 quarters in the two years) and its harmonics.

Note: the exact difference of 8 between the peaks seem to dominate the signal.



Fast Fourier Transformation

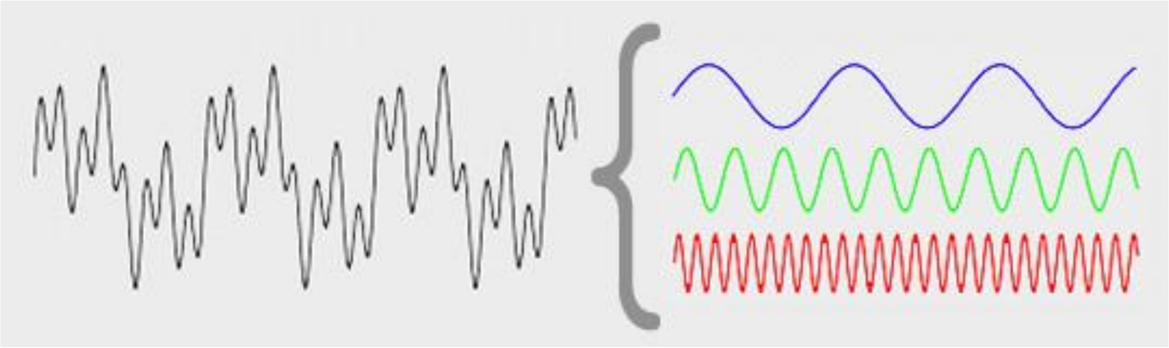


- However, these frequencies alone are insufficient to capture the time series to the precision we would like to.
- To demonstrate this, we eliminate the frequencies with “small magnitude” in a copy of the Fourier Transform.
- The remaining ones are transformed back to the time domain (an inverse FFT call) in order to be compared to the original time series.

Forecasting Methodology

- The signal we deal with shows strong regularity, but is at the same time highly complex (and decidedly nonlinear).

- Therefore, for forecasting purposes we split it into simpler periodic time series and train artificial neural networks for the finite time window forecasting of each simplified component.

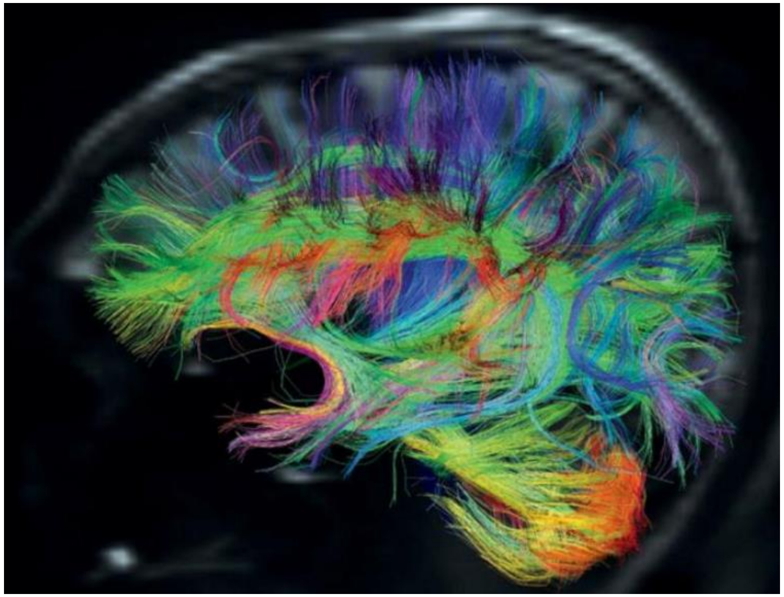


- The “splitting” is based on a non-overlapping partitioning of the frequency domain.

Forecasting Methodology

Therefore our prediction approach is the following:

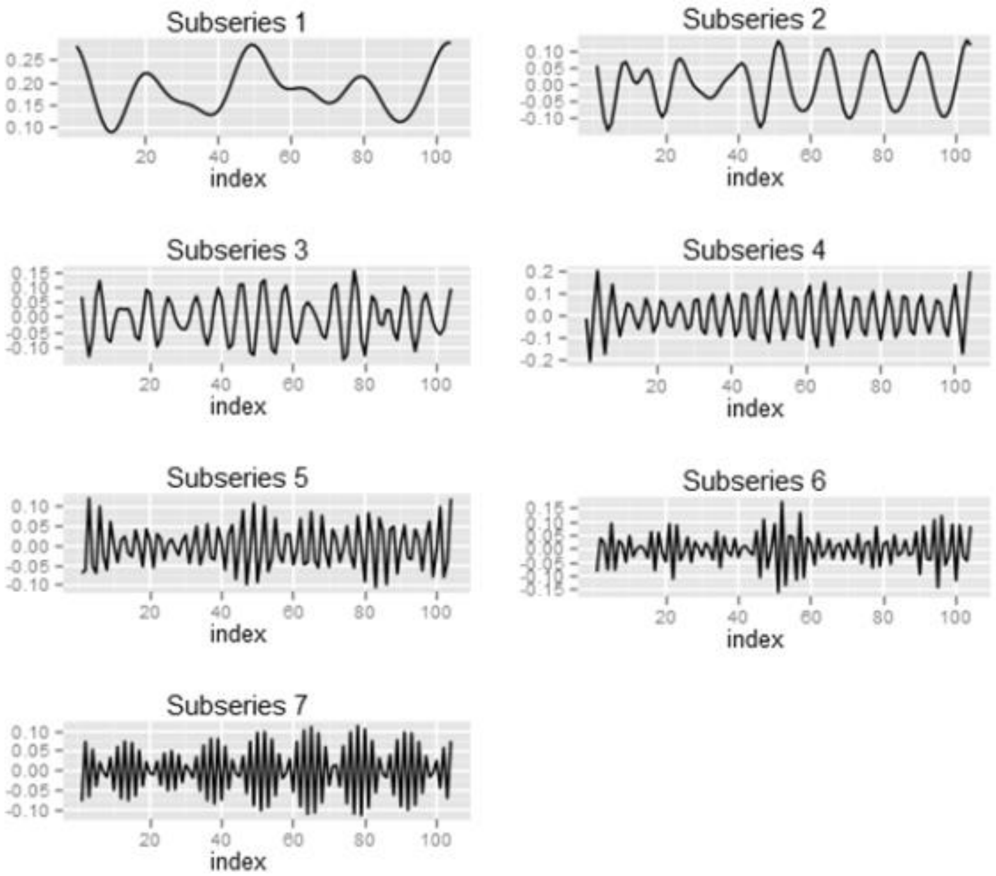
- "New" time series data is concatenated to the training set as it becomes available.
- This new, extended time series is again split with the same band-pass filtering utilized for the training data.
- The new simplified time series set (the elements of which are extended with the images of the new observations) is used to exercise the corresponding forecasting neural network.



Key Point: The order forecast for the un-split signal is reached by summing the outputs of the component-forecasting neural networks.

Signal Decomposition

- We split the frequency domain of the time series into intervals.
- Each interval contains either the fundamental frequency of the strong periodic signal or one harmonic of it.
- This is effectively a band-pass filtering based decomposition.
- For the two year time series and harmonic set, this produces the following decomposition in the time domain.



Neural Network Training

- We use the “Multilayer Perceptron” (MLP) feedforward artificial neural network model to map a finite time window of order observations onto predictions about the orders that can be expected in the future.

- We will have a number of neural networks, each one responsible for learning and predicting a frequency band of the original.

- We assume that "cutting up" the original signal by using frequency bands results in sub-signals that are fundamentally easier to predict.

Note: In the following, we work with a slightly different time series of 105 observations and the main frequency/harmonics 9,17,25,33,41,49,50 in the Fourier Series.)

Neural Network Training

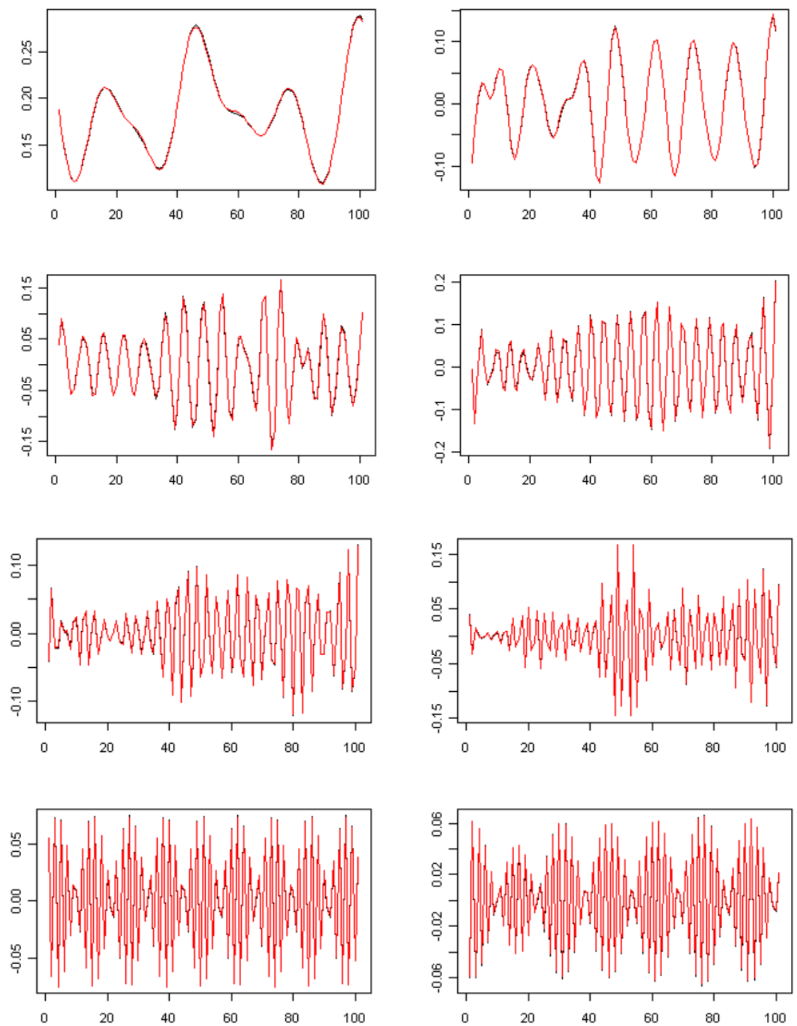
- The results of the neural network trainings for the different decomposed signals can be seen on the figure to the right.

- The red line is the response of the network, and the black is the original time series.

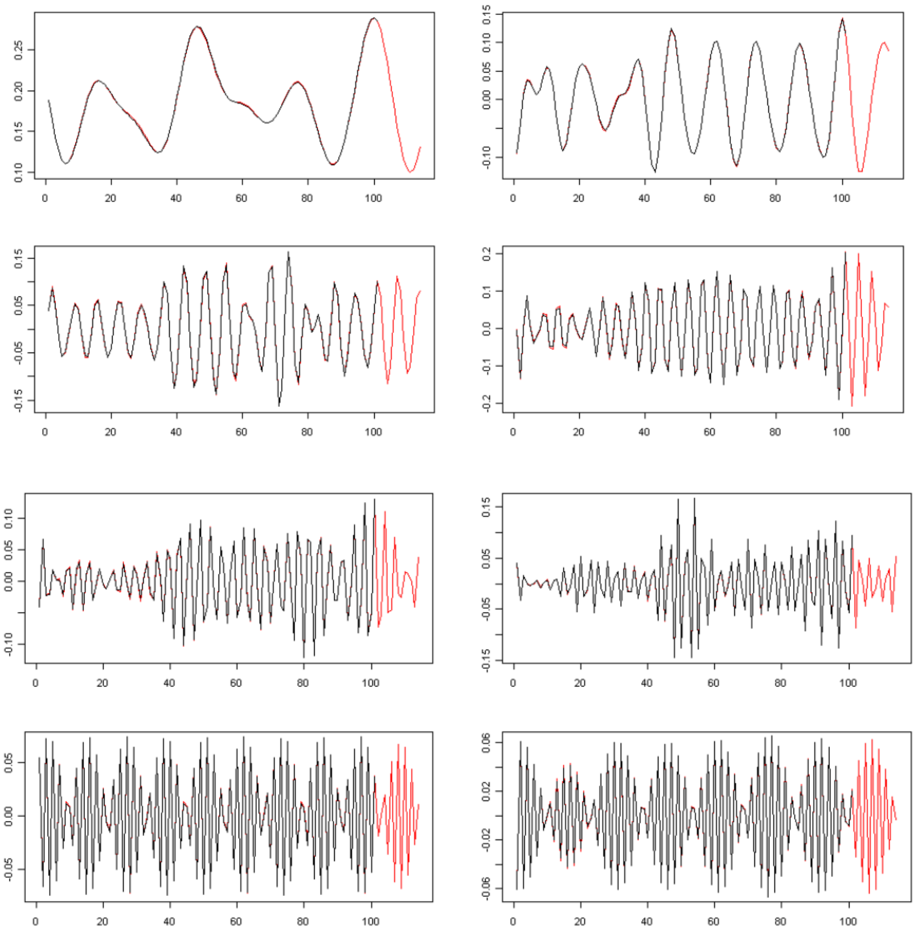
- All training data fits well, and the response of the neural networks at the training samples is accurate.

- The MSE (mean square error) for all training data is smaller than 10^(-5).

- Because the Fourier Transform is a linear operation, the sum of the individual predictions will give the full time series prediction back.



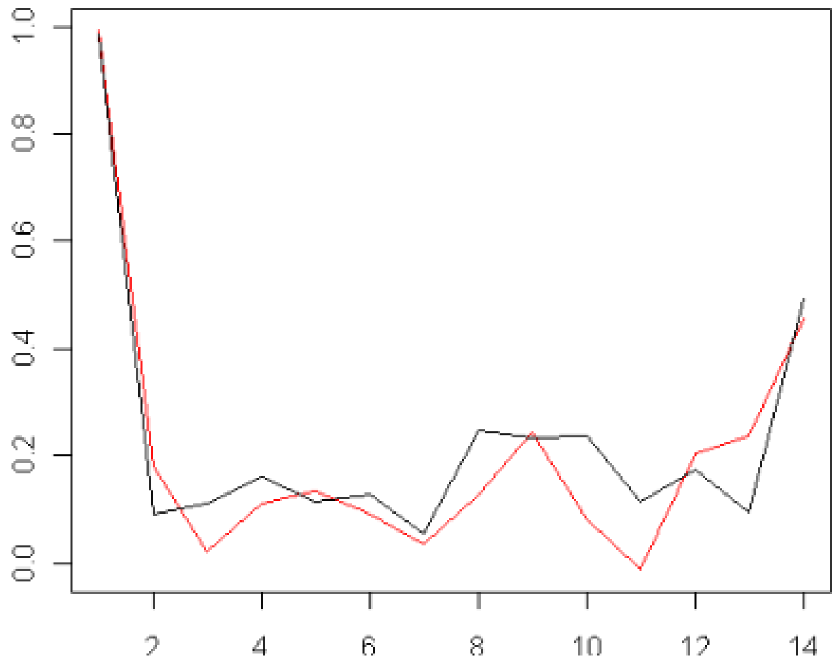
Forecasting the Time Series



- We have mentioned earlier that we have used only the first 8 quarters as training data.
- Now we will use the data for the last, ninth quarter to check how well we can predict with a one quarter horizon.
- In this approach, we predict one week, use the result of the prediction to augment the historical data and then exercise the predictor with this new input again.
- Theoretically, this feedback strategy can be used to predict to arbitrarily long horizons in the future.

Prediction Results

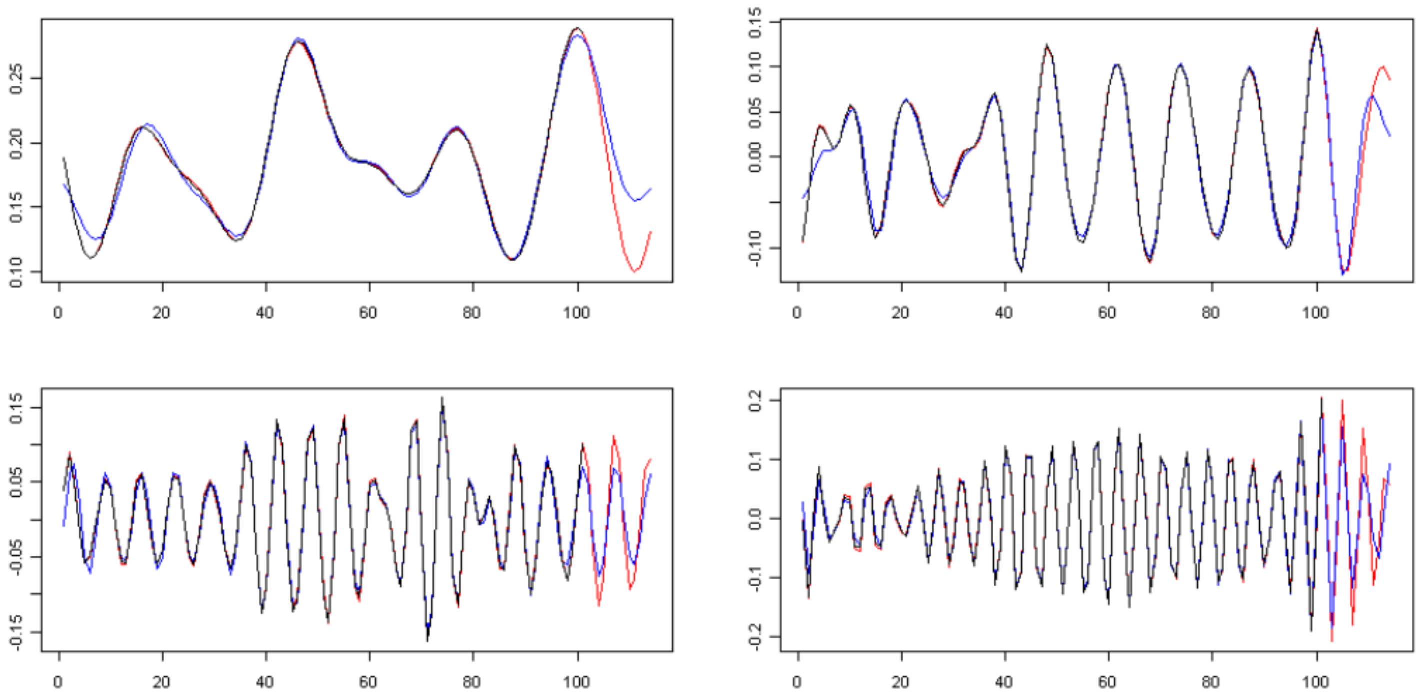
- Here the error is calculated as the RMSE (Root Mean Square Error) for the predicted time series.



- The RMSE value is 0.08.

Prediction Results

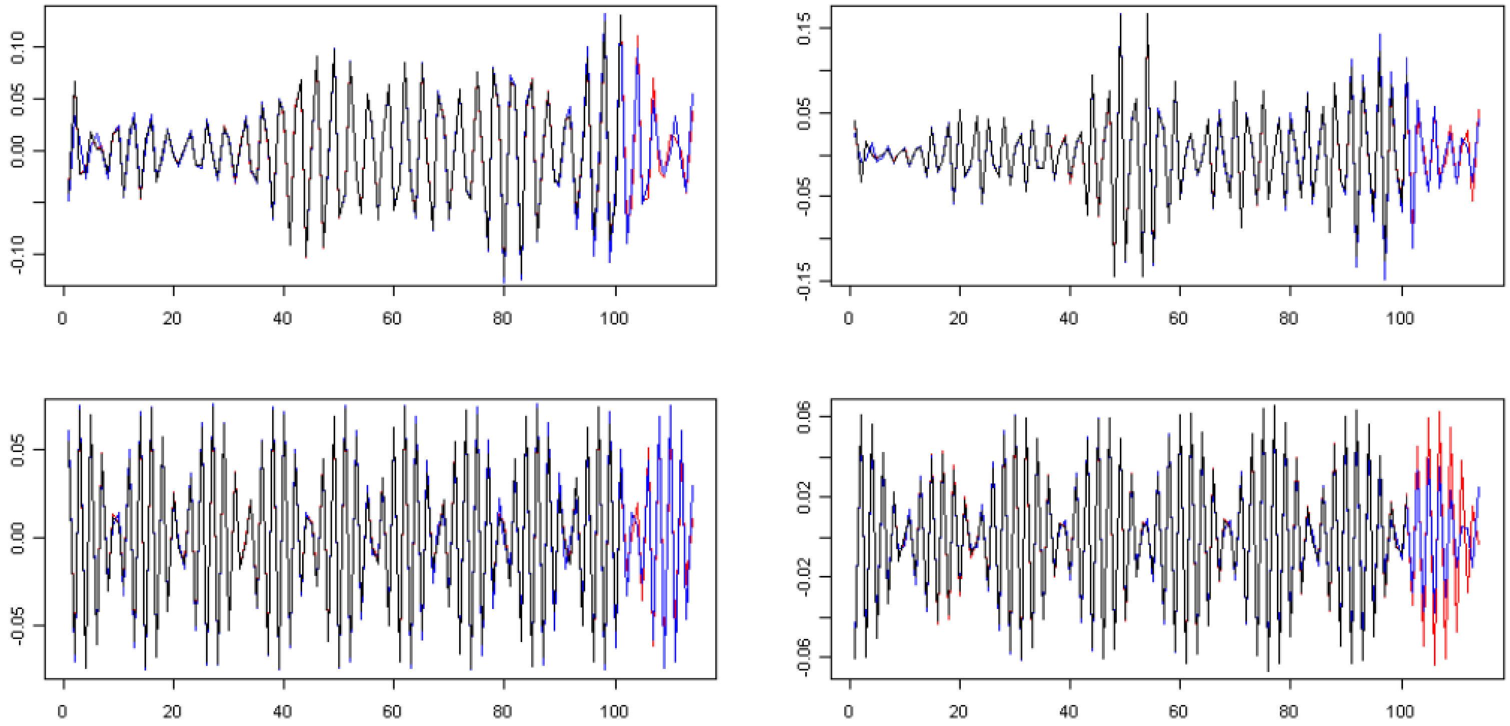
- We may also want to know, what kind of data should be predicted by the single neural networks, thus, what kind of forecast would result in a perfect prediction.



- The training data signals are black, the predicted signals are red, and the blue line are the signals that would have belonged to a perfect prediction.

Prediction Results

- The signals that were predicted more accurately where the signal was periodic, and those signals had the highest errors where the trend was dominant, the first and the second signals.



Final Results

- Here is the final recombination of the prediction for the next 14 periods:

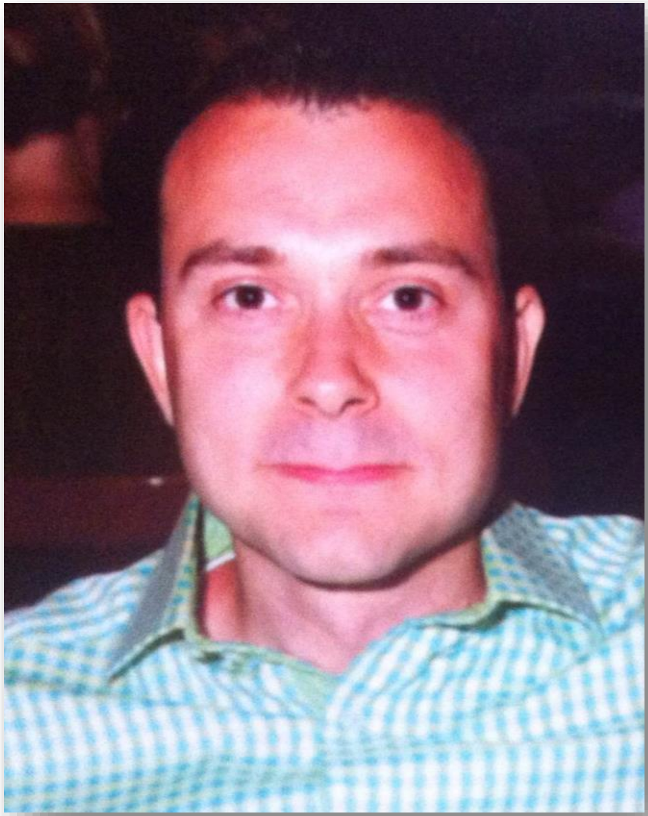
Harmonic 1	Harmonic 2	Harmonic 3	Harmonic 4	Harmonic 5	Harmonic 6	Harmonic 7	Prediction
0.28908	0.11871	0.09944	0.19462	0.12134	0.08516	0.07733	0.98567
0.28375	0.06648	0.07757	-0.01316	-0.06869	-0.08173	-0.08035	0.18387
0.27322	-0.00729	-0.03960	-0.19775	-0.05816	0.03742	0.07904	0.08688
0.25844	-0.07803	-0.13404	0.01146	0.11508	0.02309	-0.07343	0.12257
0.24087	-0.12351	-0.08846	0.18366	-0.04432	-0.06760	0.06382	0.16446
0.22204	-0.13213	0.05713	-0.00767	-0.06341	0.07271	-0.05088	0.09780
0.20366	-0.10567	0.14252	-0.15450	0.08404	-0.03549	0.03524	0.16980
0.18725	-0.05609	0.07233	-0.00080	-0.00920	-0.02508	-0.01781	0.15061
0.17417	0.00082	-0.06639	0.11465	-0.05723	0.07745	-0.00041	0.24306
0.16544	0.05141	-0.12069	0.01613	0.03615	-0.09331	0.01856	0.07370
0.16172	0.08717	-0.04843	-0.06926	0.02926	0.06236	-0.03545	0.18738
0.16324	0.10441	0.05020	-0.03796	-0.04153	0.00222	0.05040	0.29098
0.16986	0.10220	0.07760	0.02323	-0.01702	-0.07009	-0.06237	0.22341
0.18103	0.08084	0.03916	0.06380	0.06057	0.10848	0.07091	0.60479

Year	Quarter	Week	Prediction
2010	Q4	52	0.98567
2011	Q1	1	0.18387
2011	Q1	2	0.08688
2011	Q1	3	0.12257
2011	Q1	4	0.16446
2011	Q1	5	0.09780
2011	Q1	6	0.16980
2011	Q1	7	0.15061
2011	Q1	8	0.24306
2011	Q1	9	0.07370
2011	Q1	10	0.18738
2011	Q1	11	0.29098
2011	Q1	12	0.22341
2011	Q1	13	0.60479

Reminder: We need to add all of the harmonics back together to build the final prediction.

About Me

- Reside in Wayne, Illinois
- Active Semi-Professional Classical Musician (Bassoon).
- Married my wife on 10/10/10 and been together for 10 years.
- Pet Yorkshire Terrier / Toy Poodle named Brunzie.
- Pet Maine Coons' named Maximus Power and Nemesis Gul du Cat.
- Enjoy Cooking, Hiking, Cycling, Kayaking, and Astronomy.
- Self proclaimed Data Nerd and Technology Lover.



Acknowledgements

- http://en.wikipedia.org/wiki/Fourier_analysis
- http://labs.eeb.utoronto.ca/dickinson/MorphometricMethods.HTML
- https://danslimmon.wordpress.com/2014/05/16/wherein-i-rant-about-fourier-analysis/
- http://www.wisegeek.com/what-is-a-fourier-analysis.htm
- http://www.di.fc.ul.pt/~jpn/r/fourier/fourier.html
- http://www.stats.uwo.ca/faculty/aim/tsar/tsar.pdf
- http://www.inside-r.org/howto/time-series-analysis-and-order-prediction-r
- http://www.r-bloggers.com/fft-fast-fourier-transform-of-time-series-promises-and-pitfalls-towards-trading/
- http://www.r-bloggers.com/the-fourier-transform-explained-in-one-sentence/
- http://en.wikipedia.org/wiki/Fast_Fourier_transform
- http://www.reakkt.com/2010/08/different-frequency-fft-waves-for-s.html
- http://pretendy.tumblr.com/post/28938647513/why-the-fourier-transform-is-great-if-you-study
- http://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

Fine