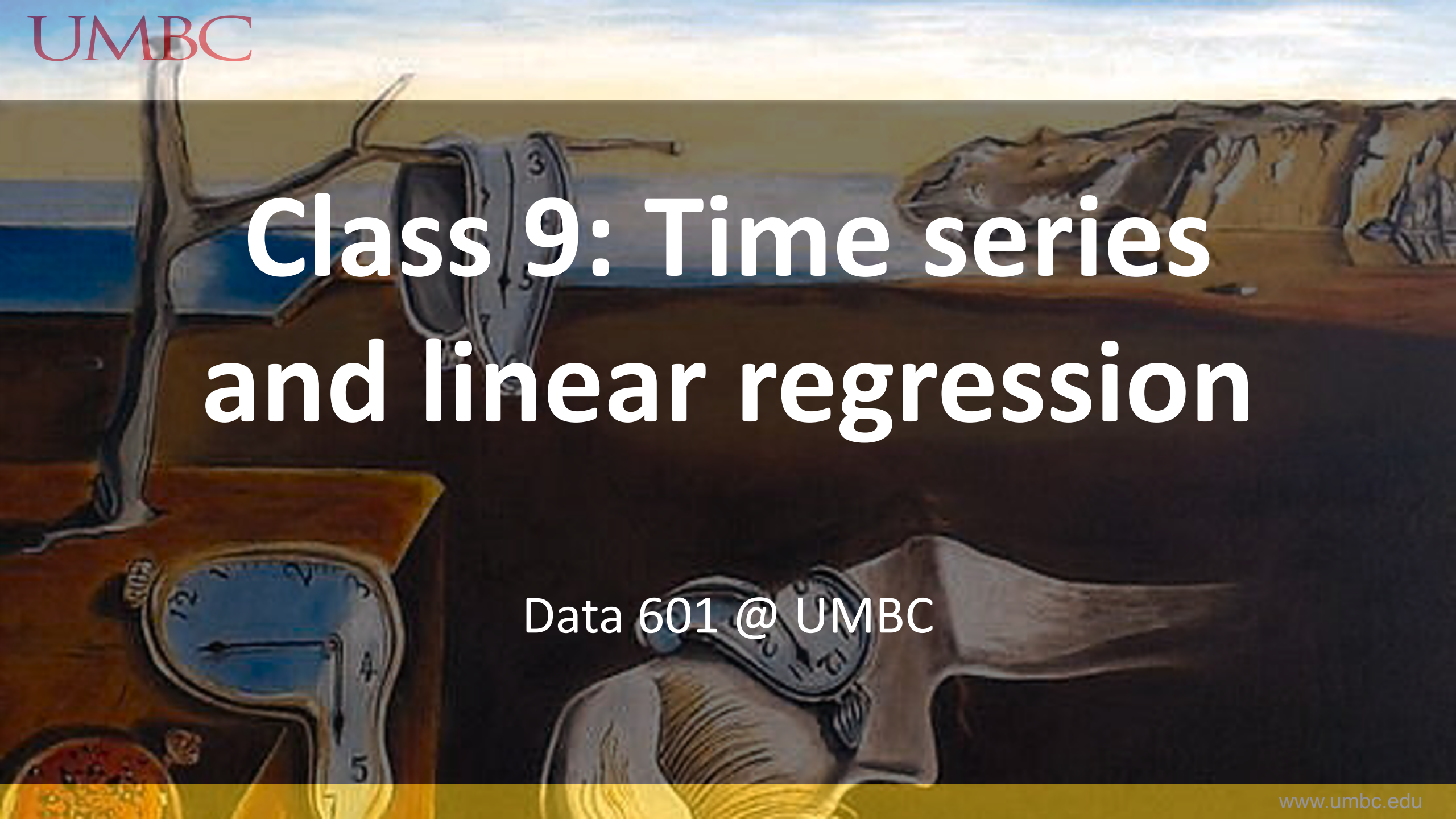


UMBC

Class 9: Time series and linear regression

Data 601 @ UMBC

Outcomes for this evening

By the end of today's class, you should be able to answer the following:

- How to use Linear Regression for trend analysis
- How Fourier Transform helps determine seasonality

Courses offering full semester on time series

- ECON 423 - Time Series and Forecasting: Study of the application of economic statistics to forecasting problems. Topics covered include analysis of cross- section and time-series data, use of published economic indicator series and forecasting methodology.
- STAT 417 - Introduction to Time Series Data Analysis: Concepts in time series analysis, such as stationarity; some commonly used time series models, such as autoregressive moving average models, are introduced using examples. Time series data analysis tools, namely, auto-correlation function (ACF), partial autocorrelation function (PACF), detrending, differencing and forecasting will be discussed using real data sets. Some selected topics from time series modeling, such as transfer function models and intervention models, will be discussed. Data analysis using statistical software such as SPLUS will be an integral part of the course.
- STAT 617 - Time Series Analysis: Theory and applications of time series models, auto-regressive integrated models, Box-Jenkins methodology, forecasting, seasonal models and their applications, spectral theory and estimation of time series models and time series data analysis using statistical packages.

- What is time stamped data?
- Why does time stamped data matter?
- Data cleanup of time series
- Time stamps in Python
- Visualization
- Linear regression for trend analysis
- Fourier transform
- Homework

Time series

Time	Event
…	…
date/time	value
….	….

To make the connection explicit, a column in a Pandas dataframe is of type "series"

Time series

Time	Event 1	Observation	sensor
…	…	…	…
date/time	value	value	value
….	….	…	…

Hmm, this starts to look like a Pandas dataframe...

Activity: What are examples of time series?

- Brainstorm ideas

What creates time series?

Periodic and ad hoc historical data

- Stock market pricing
- Conference attendance count
- Crop yields
- Number of cars on road per year

Physical sensors

- Self-driving car: LIDAR

Internet of Things

- Medical devices
- Industrial control systems
- Self-driving cars

Order distinguishes time series from other data

- When flipping a coin and rolling a die, outcomes are not dependent on previous events – the system does not have memory (stateless)

- Events in a time series are ordered – shuffling the sequence of events loses information

order_distinguishes_time_series.ipynb

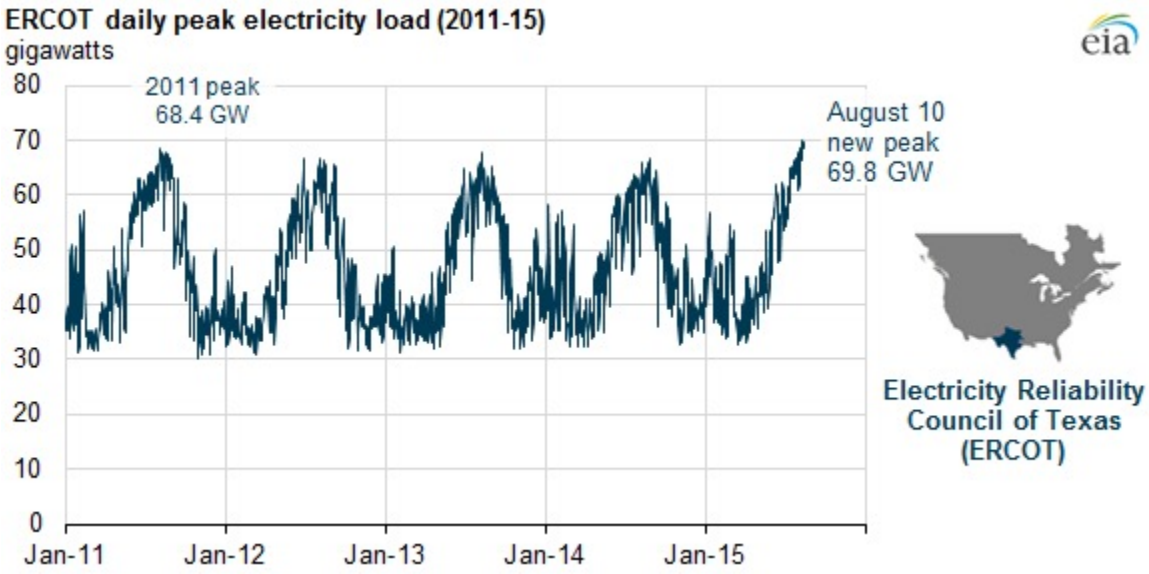
- ~~What is time stamped data?~~
- Why does time stamped data matter?
- Data cleanup of time series
- Time stamps in Python
- Visualization
- Linear regression for trend analysis
- Fourier transform
- Homework

Why care?

Evolution of sensor values over time can be a story that is useful

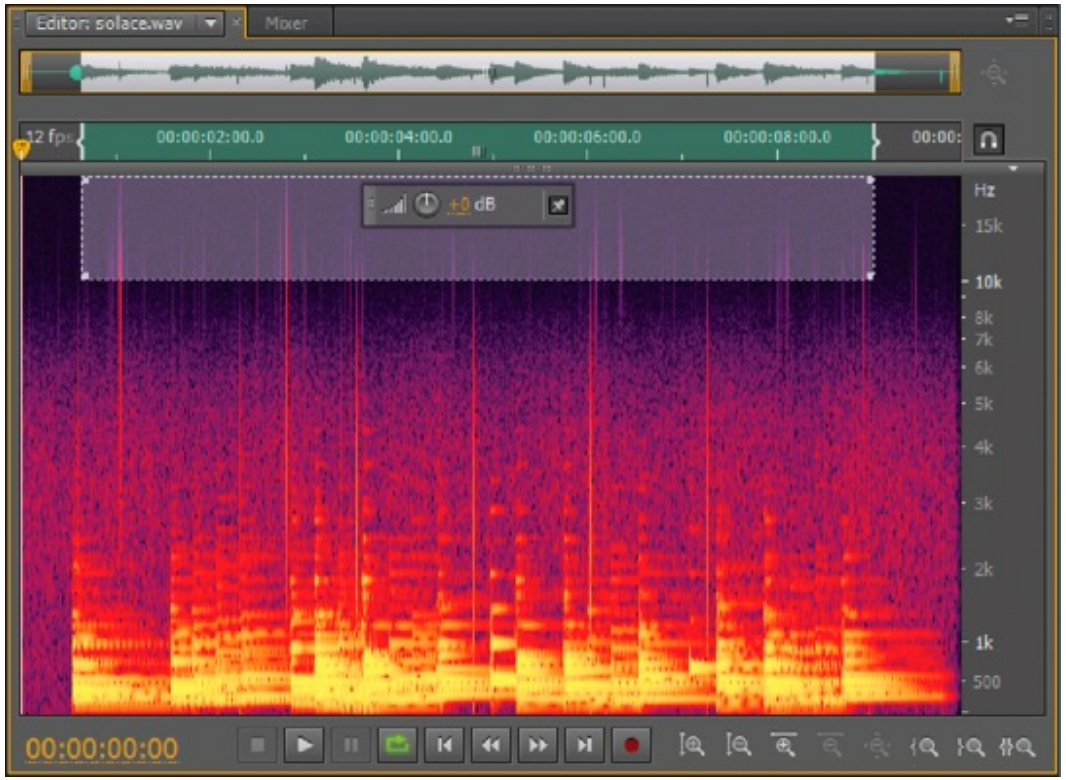
- Pattern recognition
- Anomaly detection
- Predictive analysis (extrapolation)

Application: Load forecasting enables resource planning



Application: Weather forecasting enables you to make more precise plans for the weekend

Application: voice recognition software



Common tasks for analysis of time series

Independent of the domain of application

Distinguish these aspects in a single time series

- Trends
- Changes in trends
- Periodicity (also known as seasonality)
- Noise (also known as residuals)
- Anomalies
- Averages (aka level)

Once univariate analysis has been accomplished, typically want to combine multiple time series sources (multivariate analysis)

- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- Data cleanup of time series
- Time stamps in Python
- Visualization
- Linear regression for trend analysis
- Fourier transform
- Homework

Common data cleanup challenges

For a single data source,

- Which timezone?
- By how much does the clock drift?
- Which time stamp format? See http://strftime.org/
- YY-MM-DD or MM-DD or MM-DD-YY ?
- What granularity (Year/Month/Day/Hour/Minute/Second)?

Each solution has different constraints and different assumptions

Common data cleanup challenges

Inconsistent formats when combining multiple sources

- Same timezone as other sensors? UTC versus local?
- Are all clock sources synchronized?
- Different software has different default formats for time stamps
- Solutions implemented at different times adhere to distinct schemas

Creating a dataframe of time stamped data

Every row is either

- a time bin (with start and stop edges)
  - is the time stamp the start, end, or middle value?

or

- an instantaneous value
  - What if sensor times are not aligned?

How to deal with missing sensor values? Fill in with NaN?

Creating a dataframe of time stamped data

Every row is either

- a time bin (with start and stop edges)
  - is the time stamp the start, end, or middle value?

or

- an instantaneous value
  - What if sensor times are not aligned?

temporal_anomalies_and_bin_alignment.ipynb

- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- ~~Data cleanup of time series~~
- Time stamps in Python
- Visualization
- Linear regression for trend analysis
- Fourier transform
- Homework

Activity: how do you describe now?

1. What is today's day, date, and time

(Write in the chatbox – DO NOT press enter)

Time stamp formats

Reasons for disparity:

- Every vendor has the freedom to choose their own default output
- What seems best for human consumption is a subjective choice
- Different optimization objectives: compactness versus granularity
- Different assumptions or use cases
  - "users would always use local timezone" or
  - "users would always use UTC"

You are not the first person to encounter this

Primary library: https://docs.python.org/3/library/datetime.html

Useful reference page: http://strftime.org/

go to http://strftime.org/

I'll provide an example of strftime in a few slides

Time is relative: Timezones

Naïve assumption: there are 24 timezones

--> Currently 37 Different Local Times in Use (source: https://www.timeanddate.com/time/current-number-time-zones.html)

- Rather than specify "Eastern Time Zone", a common alternative is UTC offset
- For example, **UTC**−05:00 is a time **offset** that subtracts five hours from Coordinated Universal Time (**UTC**).

`timezones.ipynb`

Time is local: Timezones

Naïve assumption: timezones are static

- In North America, UTC-5 is observed in the **Eastern** Time Zone during standard time, and in the Central Time Zone during the other eight months (see Daylight savings time). Source: https://en.wikipedia.org/wiki/UTC%E2%88%9205:00

Reverse engineering the local time based on a location is susceptible to local exceptions; see https://state.1keydata.com/time-zone-state.php

One solution: use Epoch time

`time.time()` function returns the number of seconds since the epoch.

"Epoch" is defined as the start of January 1st, 1970 in UTC.

```
>>> import time
>>> print(time.time())
1355563265.81
```

This can be converted to a string of your choosing:

```
>>> import datetime
>>> print(datetime.datetime.fromtimestamp(
                        time.time()).strftime('%Y-%m-%d %H:%M:%S'))

2012-12-15 01:21:05
```

- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- ~~Data cleanup of time series~~
- ~~Time stamps in Python~~
- Visualization
- Linear regression for trend analysis
- Fourier Transform
- Homework

Plotting Time series: x-axis as datetime

- Data typically has a column of datetime entries
- By default, index in Pandas is integers

Data source: `host_streaming_stats/host_cpu_and_memory.ipynb`
`plotting_timeseries_data.ipynb`

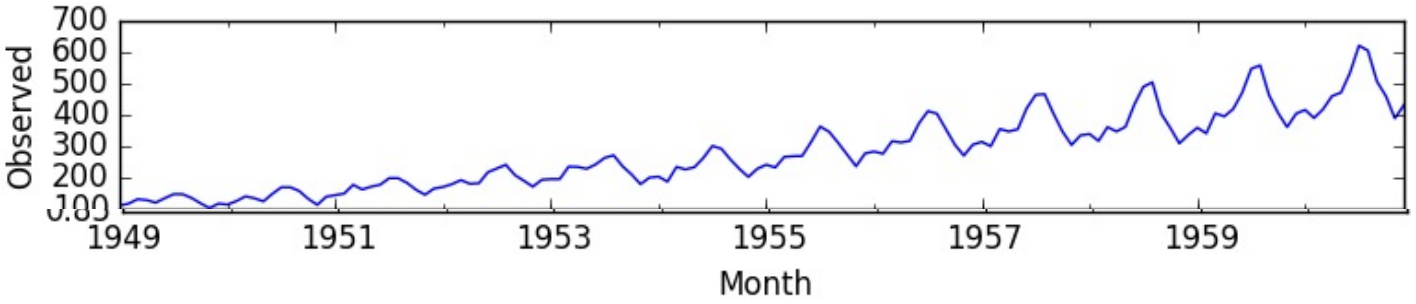
Good overview: https://machinelearningmastery.com/time-series-data-visualization-with-python/

Flat lines and sine waves

- https://seaborn.pydata.org/generated/seaborn.lineplot.html#seaborn.lineplot
- https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html

- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- ~~Data cleanup of time series~~
- ~~Time stamps in Python~~
- ~~Visualization~~
- Decomposing time series
- Linear regression for trend analysis
- Fourier transform
- Homework

Temporal data has multiple factors present

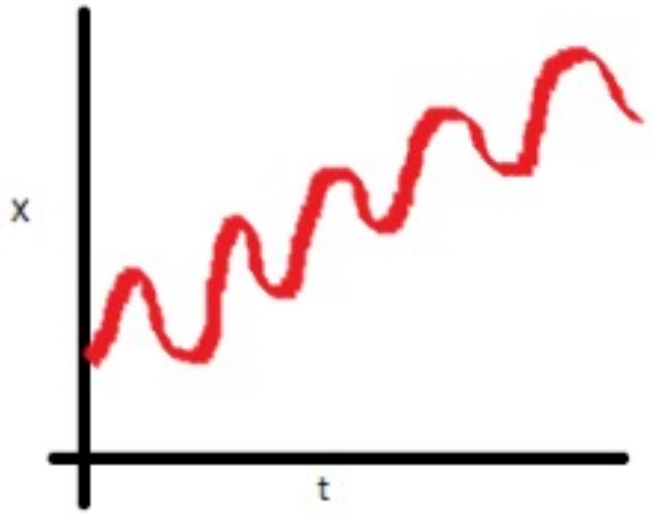


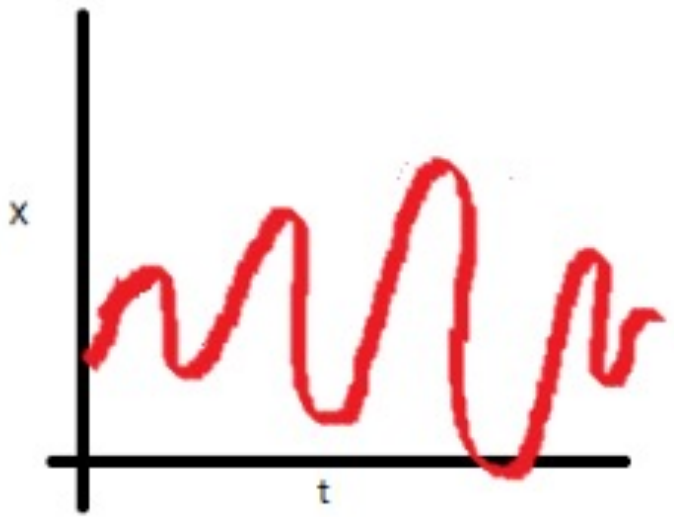
source

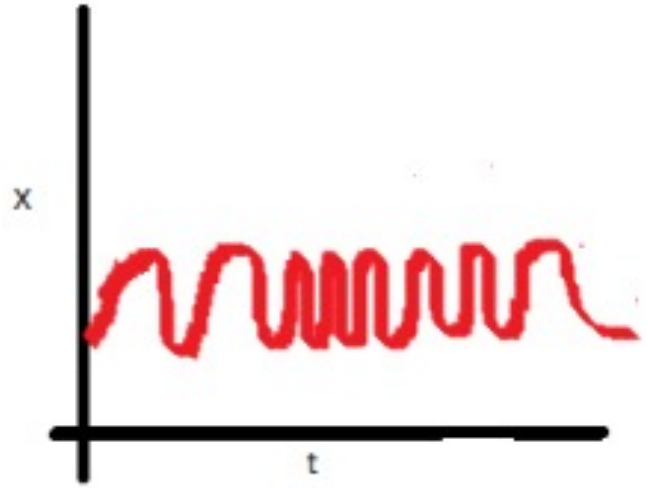
"The Airline Passengers dataset describes the total number of airline passengers over a period of time.

The units are a count of the number of airline passengers in thousands. There are 144 monthly observations from 1949 to 1960."

Stationary = **mean, variance, covariance, do not vary with time**



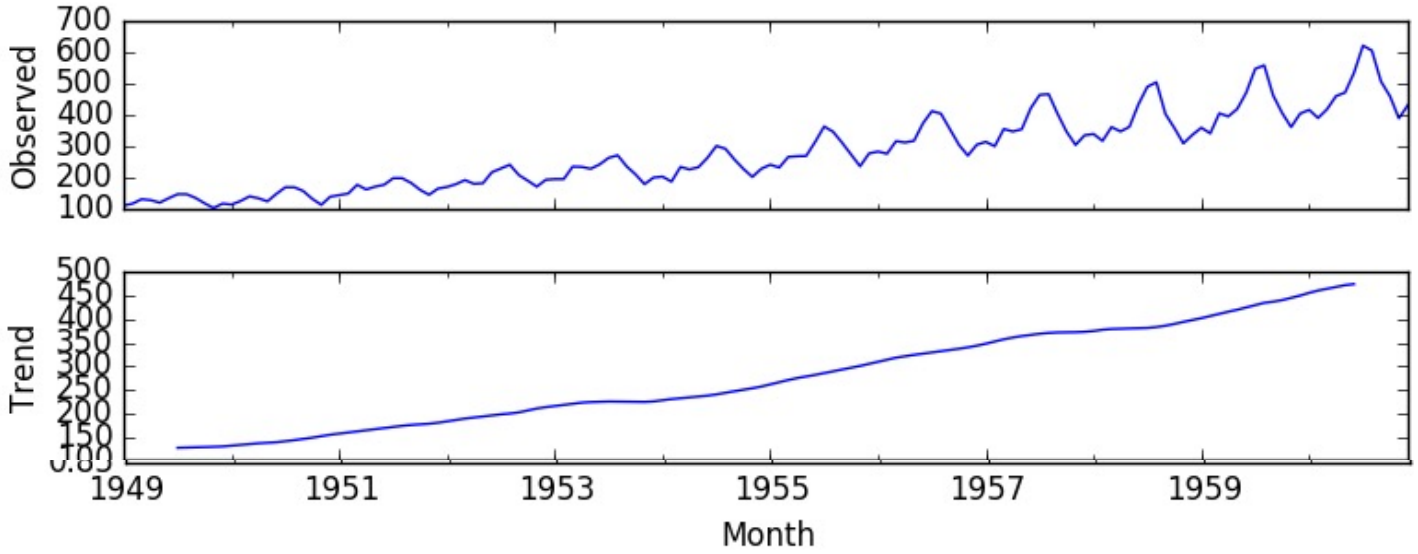




source

Motivation: decomposing temporal data

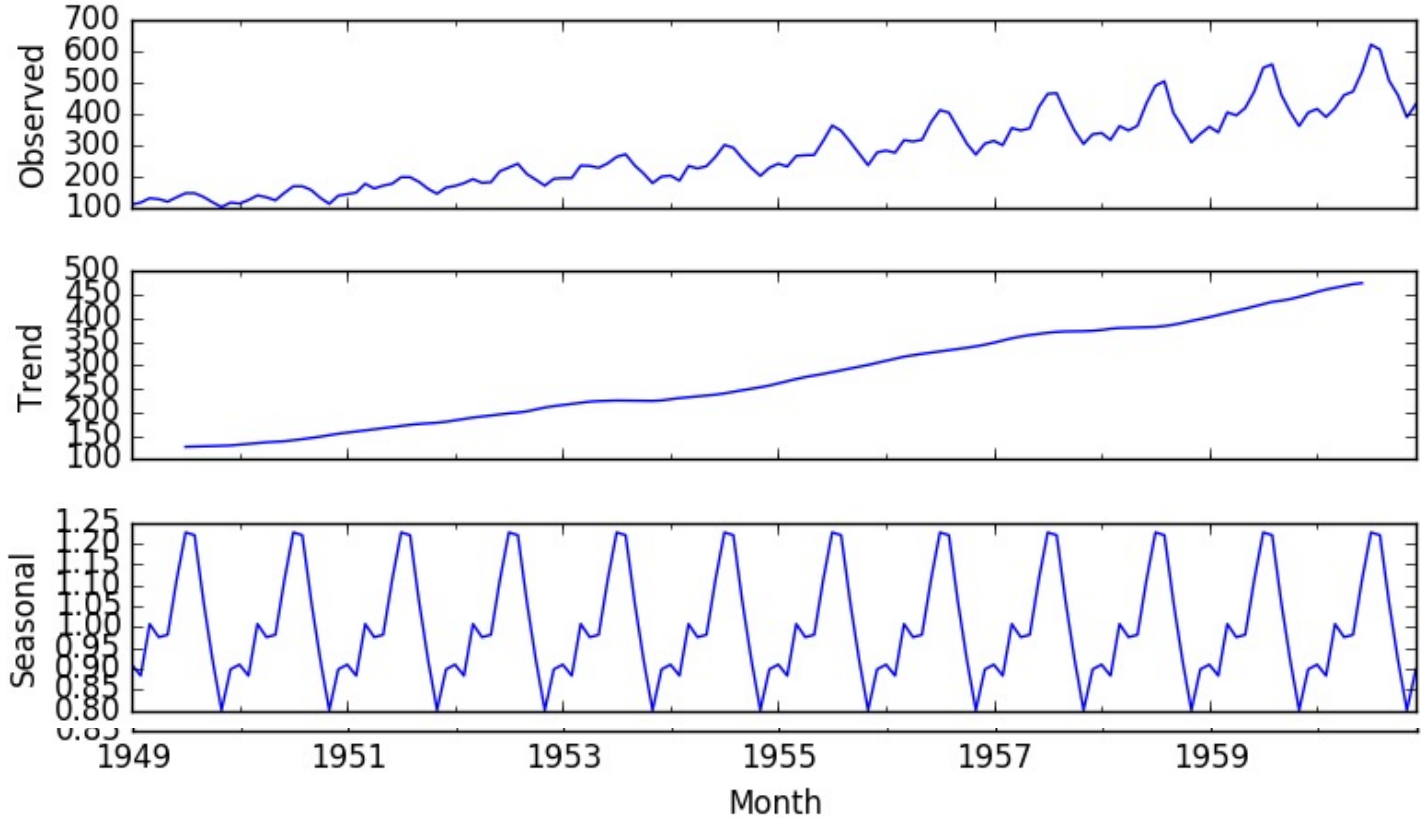
Step 1: separate the trend



source

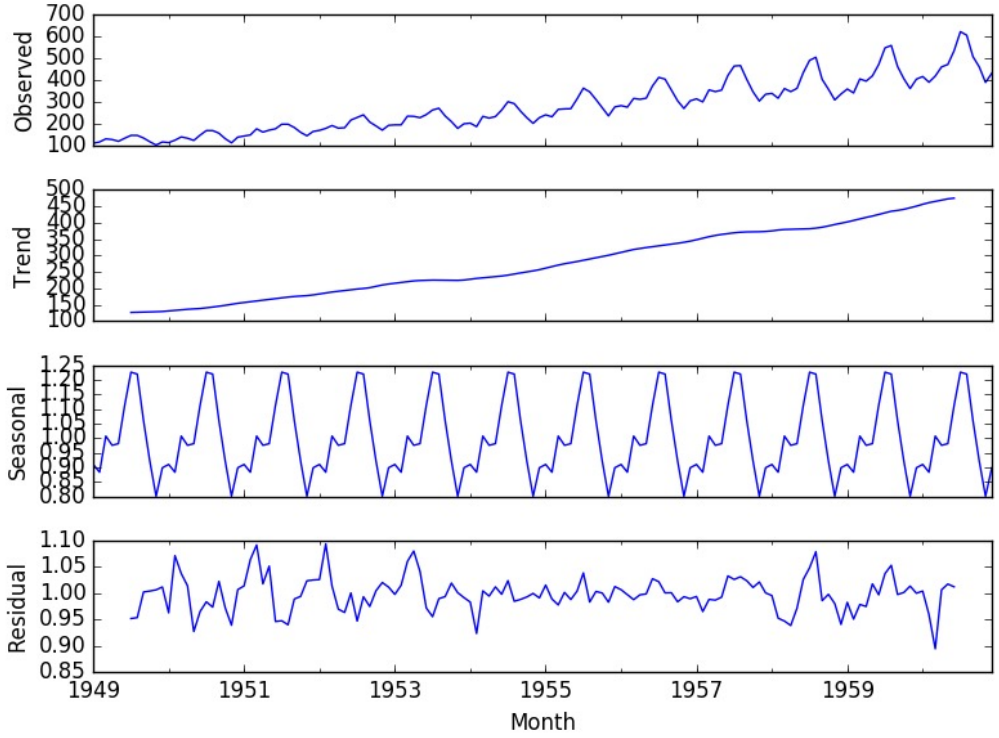
Motivation: decomposing temporal data

Step 2: separate seasonality



source

Motivation: decomposing temporal data



source

Example

- *Data source*: `historical power load_get_files_make_df.ipynb`
- `historical power df_analysis.ipynb`

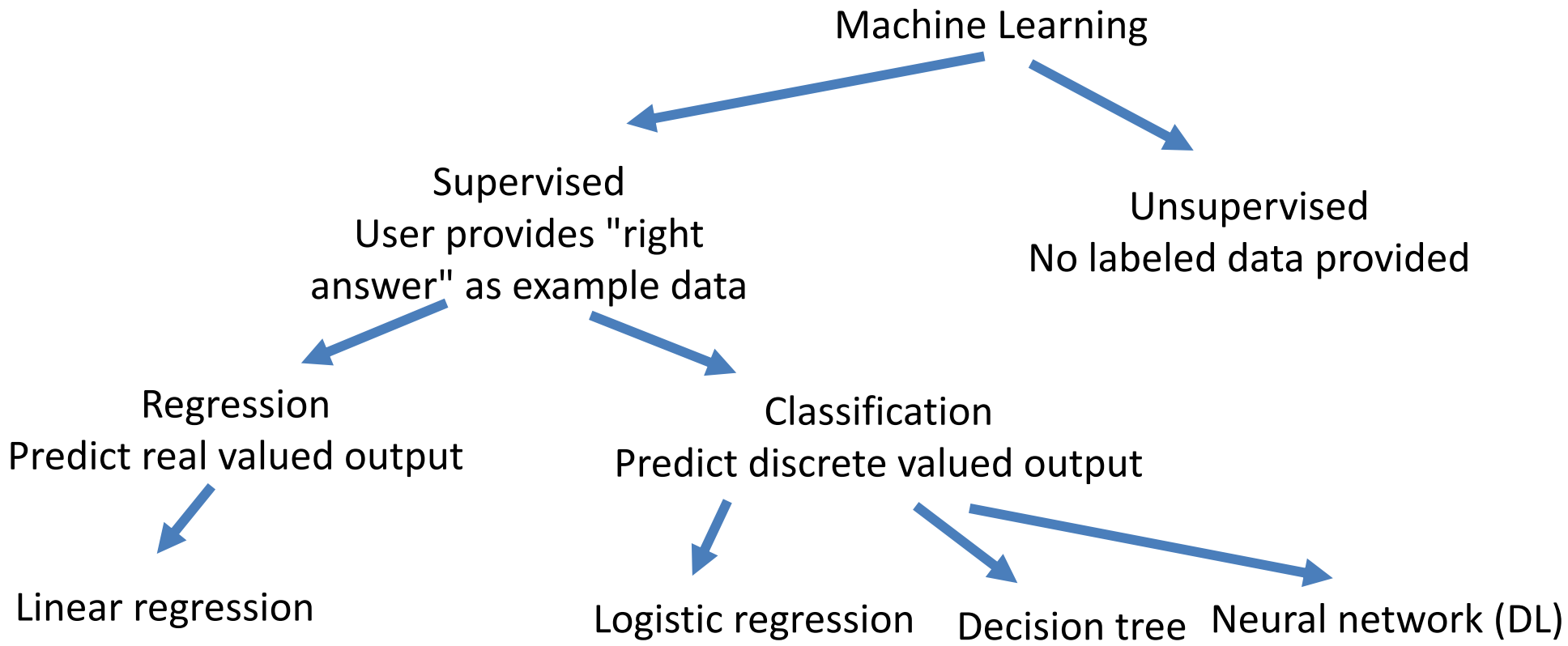
Methods to decompose temporal data

Trend removal

- Differencing
- Rolling average
- Linear regression

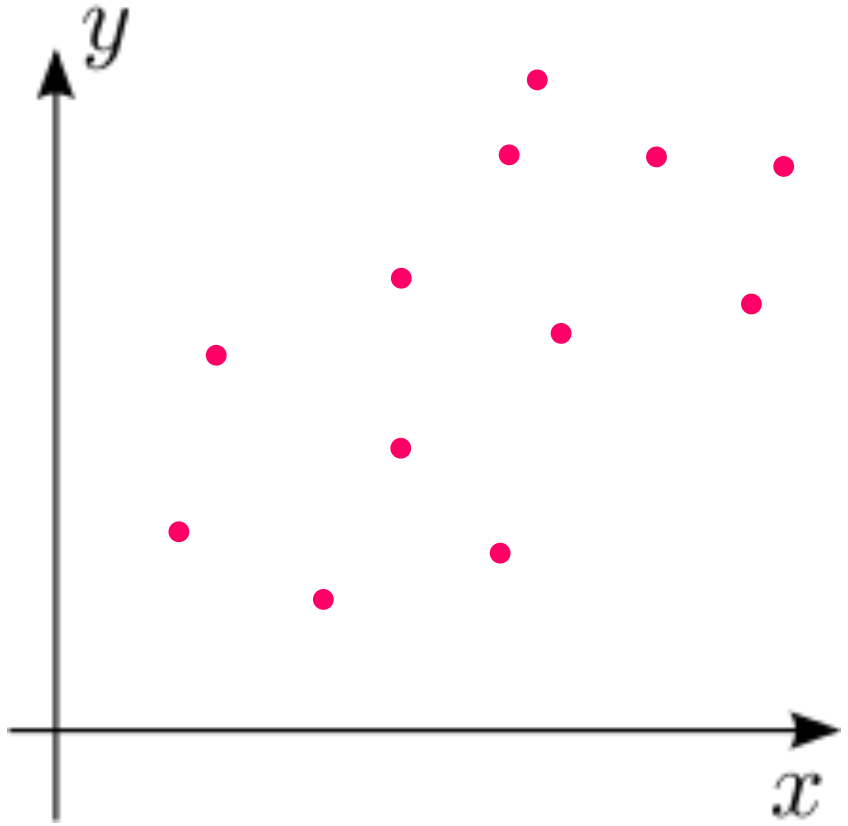
- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- ~~Data cleanup of time series~~
- ~~Time stamps in Python~~
- ~~Visualization~~
- ~~Decomposing time series~~
- Linear regression for trend analysis
- Fourier transform
- Homework

Map of Machine Learning topics



Activity: Draw best fit straight line on plot

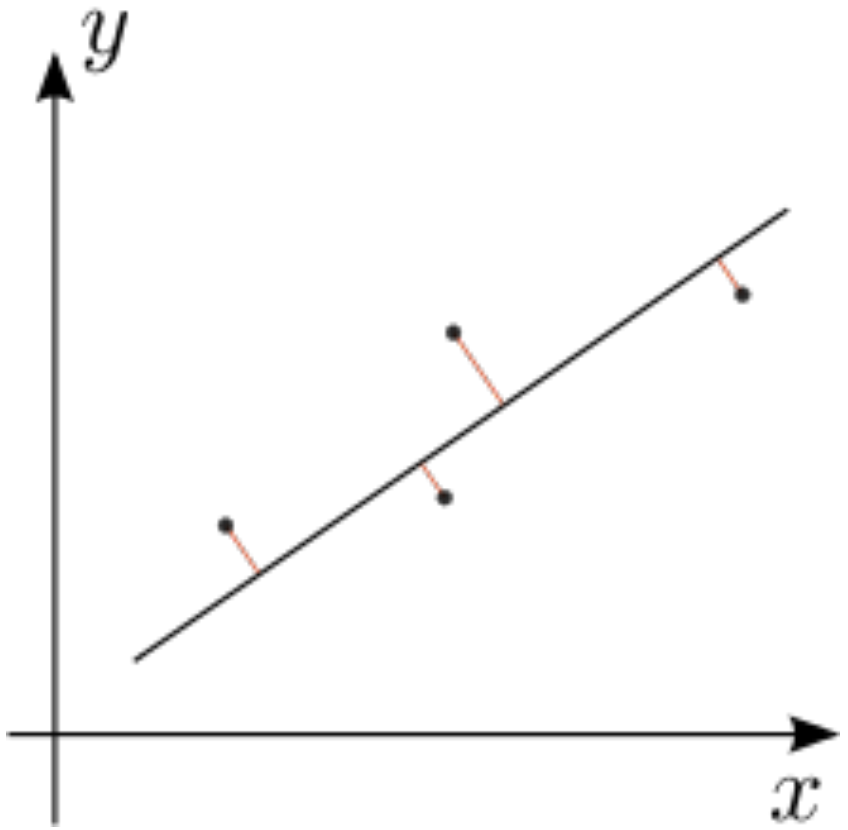
Need a volunteer who is not in Data 602



What you (probably) did

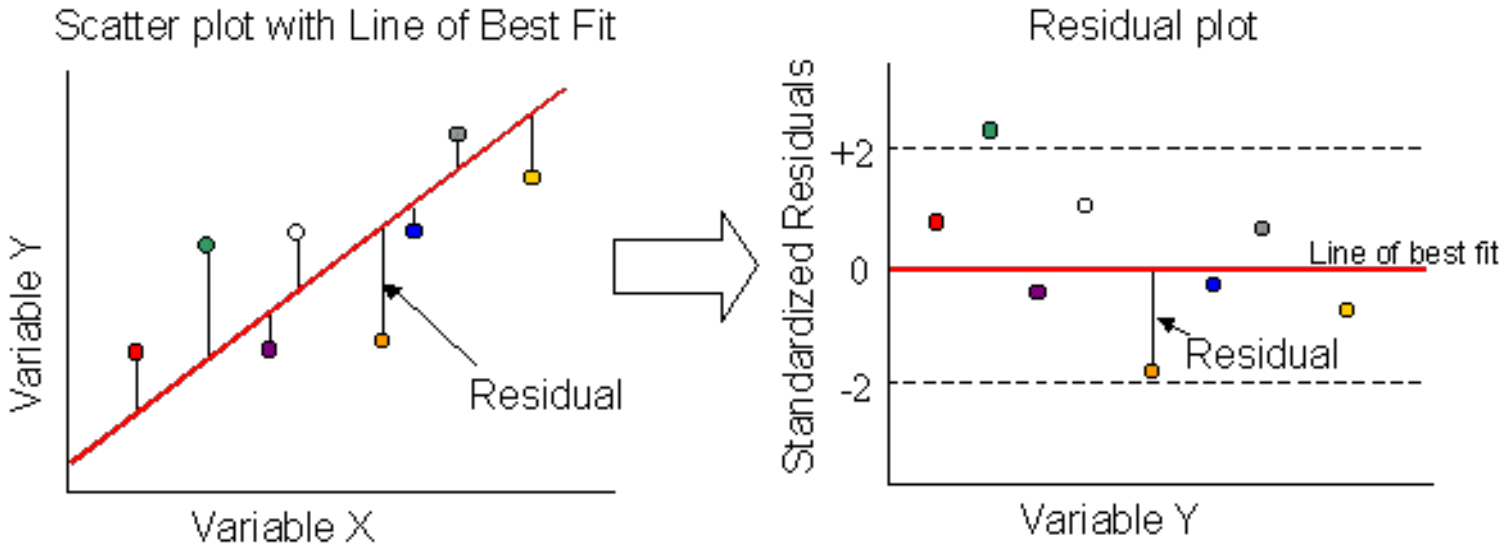
If there is uncertainty in both the x and y coordinates, then use an approach which admits variation in both

https://en.wikipedia.org/wiki/Deming_regression



Linear Regression concepts

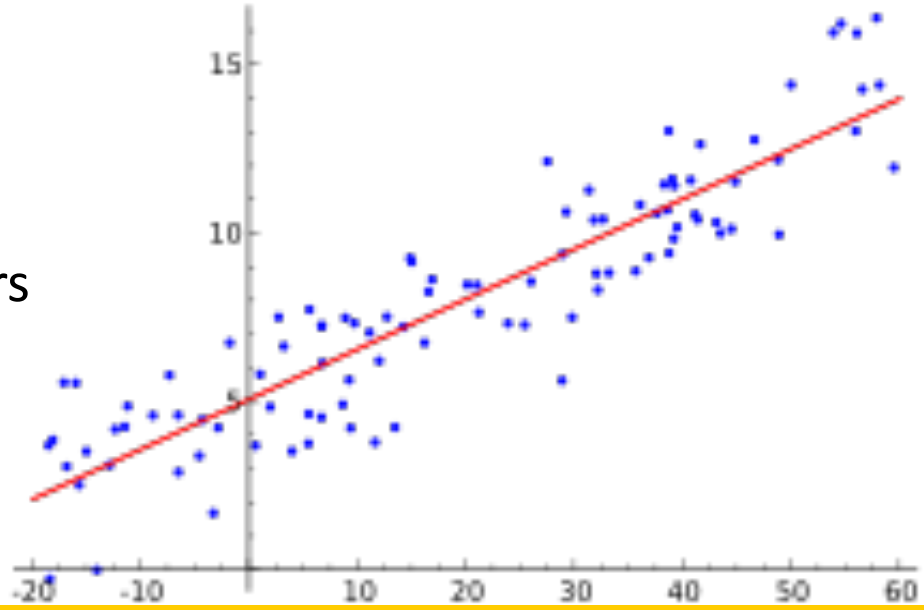
- Predict a target variable by fitting the *best linear relationship* between the dependent (Y) and independent (X) variable.
- The *best fit* is done by making sure that the sum of all the distances between the shape and the actual observations at each point is as small as possible.



Residual = difference between what the current model gives us and the "right" output

Best approach to minimize the residual depends on your data

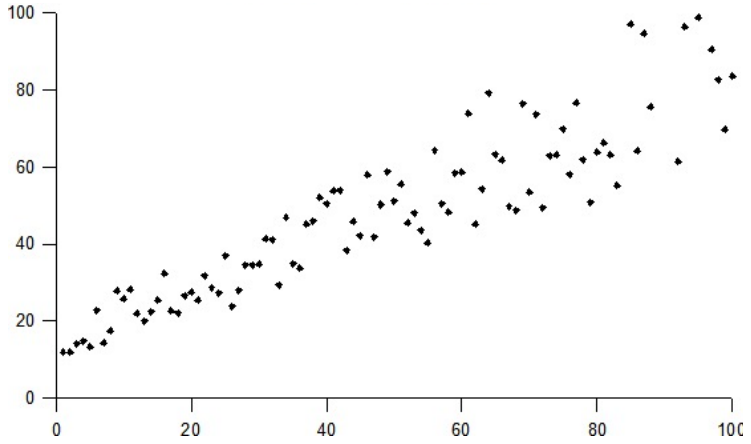
- ordinary least squares for independent and identically distributed errors
- generalized least squares for arbitrary covariance
- weighted least squares for heteroskedastic errors
  (error in y varies with x)
- feasible generalized least squares with autocorrelated errors
  (function repeats)



Residual = difference between what the current model gives us and the "right" output

Best approach to minimize the residual depends on your data

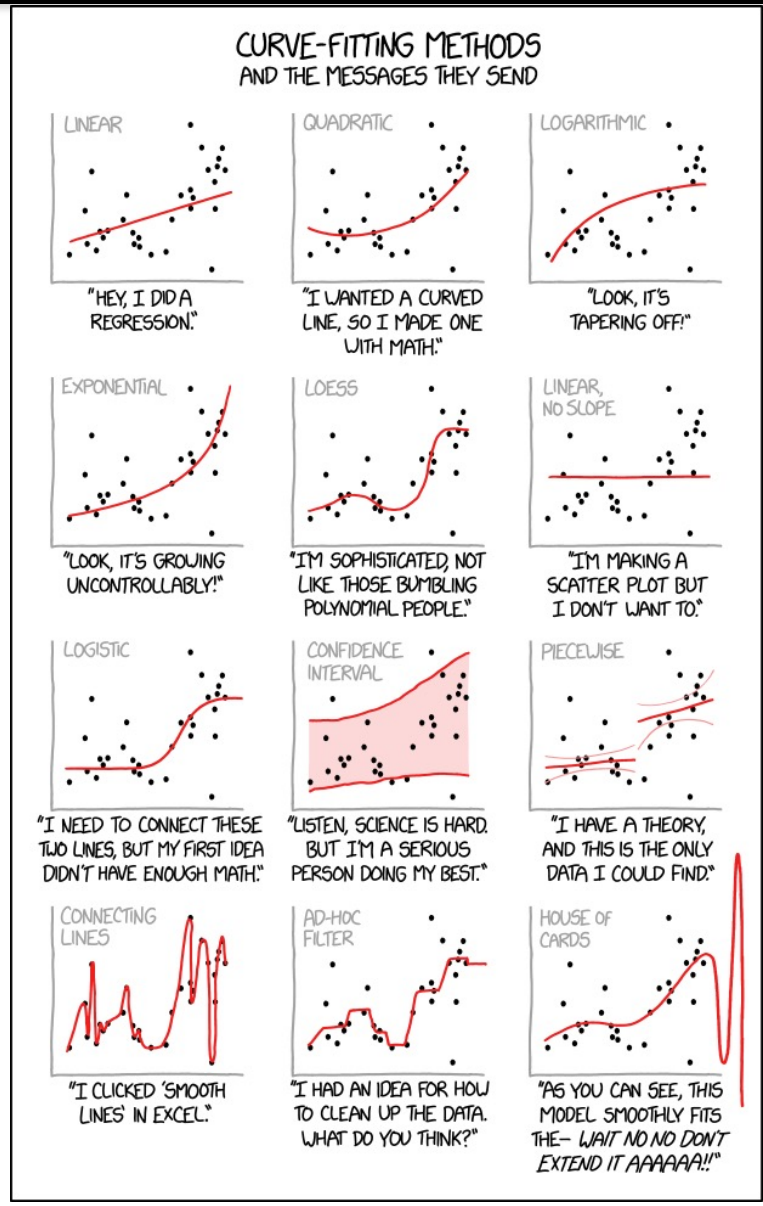
- ordinary least squares for independent and identically distributed errors
- weighted least squares for heteroskedastic errors
  (error in y varies with x)
- generalized least squares for arbitrary covariance
- feasible generalized least squares with autocorrelated errors
  (function repeats)



Residual = difference between what the current model gives us and the "right" output

Best approach to minimize the residual depends on your data

- ordinary least squares for independent and identically distributed errors
- weighted least squares for heteroskedastic errors
  (error in y varies with x)
- generalized least squares for correlated residuals
- feasible generalized least squares with autocorrelated errors
  (function repeats)



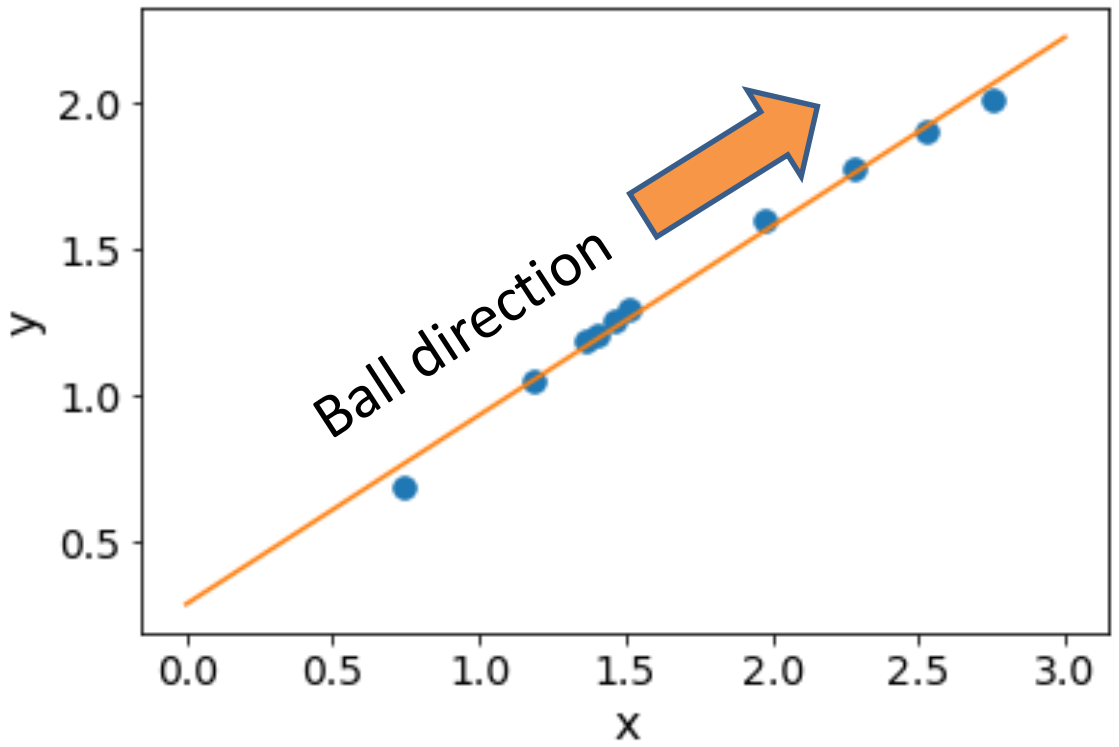
https://xkcd.com/2048

Linear regression = fit data, but for what?

- Prediction
  - Extrapolation = independent variable (x) being evaluated is outside the range of values you already have information on
  - Interpolation = value you are evaluating is within the range of values you already know
- Fitting Trends
- Measure Correlation of two variables

Physics experiment: ski jump

Projectile motion after ball launches

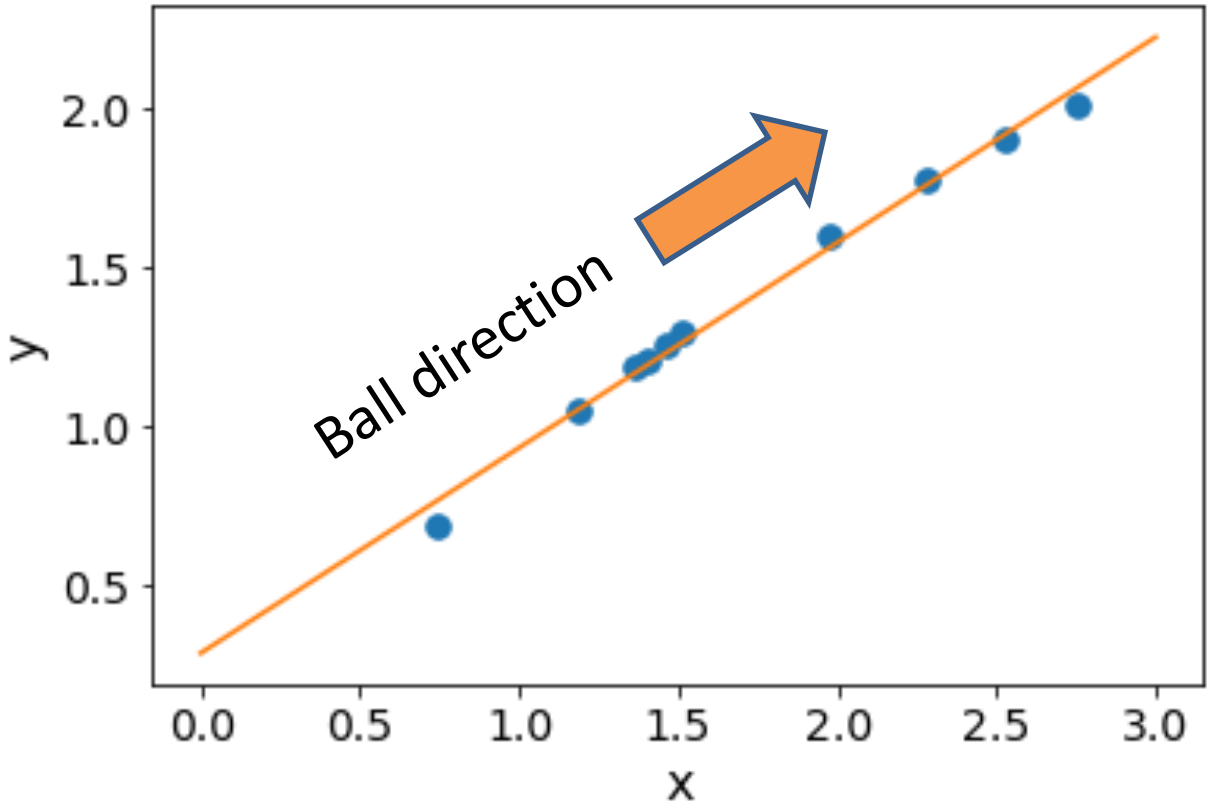


Ramp ends here

X and Y correlation = 0.996179

Y = 0.64672026*X + 0.28288071

Projectile motion *activity*: identify 2 problems



X and Y correlation = 0.996179

Y = 0.64672026*X + 0.28288071

Ramp ends here

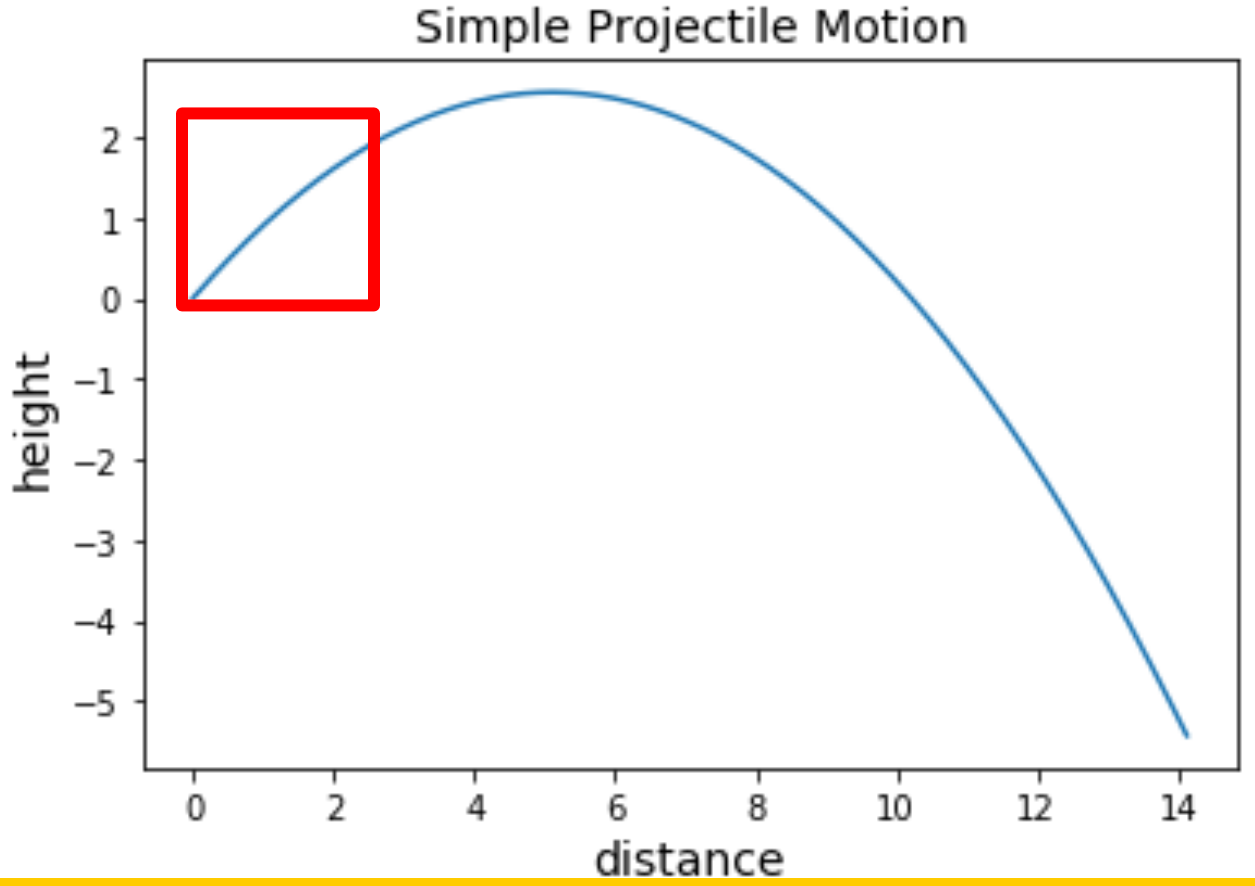
Raise your hand if you have an observation

Issues

- Insufficient sampling in relevant range

While points within the sampling window are correct, the range of interest is the full arc

- Fitting a line to the data appears adequate, but misses the underlying physics of projectile motion: a parabolic path is a second order polynomial



Lesson: Simply trusting data leads to invalid conclusion

- Consistently question your data and your own analysis
- Investigate how data was collected, by whom, for what purpose
- Additional data often helps. Can relevant data be gathered?

- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- ~~Data cleanup of time series~~
- ~~Time stamps in Python~~
- ~~Visualization~~
- ~~Linear regression for trend analysis~~
- Fourier Transform
- Homework

Fourier Transform:

- FFT only applies to data in which the timestamp is uniform

fourier_transform.ipynb

- ~~What is time stamped data?~~
- ~~Why does time stamped data matter?~~
- ~~Data cleanup of time series~~
- ~~Time stamps in Python~~
- ~~Visualization~~
- ~~Linear regression for trend analysis~~
- ~~Fourier transform~~
- Homework

Coding Homework

convert date-time to day of week

```
def convert_datetime_to_dayofweek(datetime_string):
    """
    >>> convert_datetime_to_dayofweek('Jun 1 2005  1:33PM')
    'Wednesday'

    >>> convert_datetime_to_dayofweek('Oct 25 2012  2:17AM')
    'Thursday'
    """
    # your code goes here
    return dayofweek
```

Homework: Imputation

Each XLS or XLSX file contains a dataframe.

https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html

Load each XLS or XLSX file in your Jupyter notebook.

For each dataframe, address missing values by taking the following steps:

1. Count the number of missing values in the dataframe; count the number of missing values per column
2. Plot the distribution of data points using a histogram
3. Create a lag plot (a lag plot shows *t* versus *t+1*)
4. Based on the lag plot, state in a markdown cell whether the order of this data matters.
5. Do one of the following (not both):
   - If the order of the data matters, then interpolate the missing values
   - If the order of the data does not matter, fill in the missing data by sampling from the distribution
6. Create a scatter plot using the columns in dataframe; no Nan entries should be present

Submit a single Jupyter notebook with your analysis of the XLS and XLSX files.

Perform interpolation or sampling programmatically using Python (not manually)

References

Overview

- https://jakevdp.github.io/PythonDataScienceHandbook/03.11-working-with-time-series.html
- http://www.statsoft.com/Textbook/Time-Series-Analysis
- https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm
- https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html
- https://en.wikipedia.org/wiki/Time_series
- https://www.kaggle.com/c/demand-forecasting-kernels-only/discussion/63568
- https://towardsdatascience.com/basic-time-series-manipulation-with-pandas-4432afee64ea

ARIMA and variants

- https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
  https://en.wikipedia.org/wiki/Autoregressive%E2%80%93moving-average_model
  https://www.quantstart.com/articles/Autoregressive-Integrated-Moving-Average-ARIMA-p-d-q-Models-for-Time-Series-Analysis
  http://www.blackarbs.com/blog/time-series-analysis-in-python-linear-models-to-garch/11/1/2016
  https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
  https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/
- https://towardsdatascience.com/playing-with-time-series-data-in-python-959e2485bff8
- https://machinelearningmastery.com/make-sample-forecasts-arima-python/

Tactics

- https://stackoverflow.com/questions/50161140/how-to-plot-a-time-series-array-with-confidence-intervals-displayed-in-python

- https://www.youtube.com/watch?v=tJ-O3hk1vRw
- https://www.youtube.com/watch?v=zmfe2RaX-14
- https://www.youtube.com/watch?v=Prpu_U5tKkE
- https://www.youtube.com/watch?v=NoJr08FNQeg

Tools I didn't mention in class

- https://www.google.com/search?q=facebook+prophet+python
  https://facebook.github.io/prophet/docs/quick_start.html
  https://facebook.github.io/prophet/docs/saturating_forecasts.html#forecasting-growth
  https://facebook.github.io/prophet/docs/trend_changepoints.html#automatic-changepoint-detection-in-prophet

- pip install pystan
  pip install fbprophet

Fourier Transform in Python

- https://ericstrong.org/fast-fourier-transforms-in-python/
- https://gist.github.com/tartakynov/83f3cd8f44208a1856ce
- https://www.oreilly.com/library/view/elegant-scipy/9781491922927/ch04.html

- https://docs.scipy.org/doc/scipy/reference/fftpack.html
- http://scipy-lectures.org/intro/scipy.html#fast-fourier-transforms-scipy-fftpack
- https://docs.scipy.org/doc/scipy/reference/tutorial/fftpack.html

More demos

- https://ipython-books.github.io/101-analyzing-the-frequency-components-of-a-signal-with-a-fast-fourier-transform/

- https://stackoverflow.com/questions/4479463/using-fourier-analysis-for-time-series-prediction