# V\_1.2\_march\_20\_18

# Python resources:

|  |  |  |
| --- | --- | --- |
| Gallery | <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks#programming-and-computer-science> |  |
| azure | https://notebooks.azure.com |  |
| Ibm |  |  |
| Python Data Science | <https://notebooks.azure.com/jakevdp/libraries/PythonDataScienceHandbook/tree/notebooks> |  |
| Python for data analysis | <https://github.com/wesm/pydata-book> |  |
| Python for developers | <http://nbviewer.jupyter.org/github/ricardoduarte/python-for-developers/tree/master/> |  |
| Introduction to Computer Science | <https://github.com/yoavram/CS1001.py> |  |
| Quantum computing | * [Information](https://developer.ibm.com/open/openprojects/qiskit/) about the quantum computing community and qiskit toolkit. * [Github](https://github.com/QISKit/qiskit-api-py) notebook example (scroll down) illustrating how to access a quantum computer. * [Developers' Guide](https://developer.ibm.com/code/2017/05/17/developers-guide-to-quantum-qiskit-sdk/) to quantum qiskit SDK.   <https://www.research.ibm.com/ibm-q/> |  |
| Object oriented programming: | <http://nbviewer.jupyter.org/github/mbakker7/exploratory_computing_with_python/blob/master/notebook12_oop/py_exploratory_comp_12_sol.ipynb>  (http://mbakker7.github.io/exploratory\_computing\_with\_python/) |  |
| Error handling and testing: | <https://notebooks.azure.com/garth-wells/libraries/CUED-IA-Computing-Michaelmas/html/09%20Error%20handling%20and%20testing.ipynb> |  |
| System of equations/calculus applications: | <http://nbviewer.jupyter.org/github/mbakker7/exploratory_computing_with_python/blob/master/notebook6_linear_systems/py_exploratory_comp_6_sol.ipynb> |  |
|  |  |  |
|  |  |  |

# yml file install

conda env export -f environment.yml

conda env create -f environment.yml

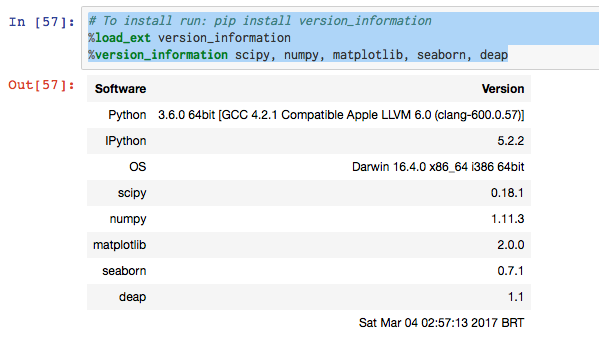
Notebook sources

# Load version information:

<http://nbviewer.jupyter.org/github/lmarti/evolutionary-computation-course/blob/master/AEC.04%20-%20Evolutionary%20Strategies%20and%20Covariance%20Matrix%20Adaptation.ipynb>

*# To install run: pip install version\_information*

%**load\_ext** version\_information

%**version\_information** scipy, numpy, matplotlib, seaborn, deap

# Python to R

<http://hafen.github.io/rbokeh/#background>

map plot

contour image plot

# Data cleaning

|  |  |  |  |
| --- | --- | --- | --- |
| Missing data |  |  | missingno <https://github.com/ResidentMario/missingno> |
| Data cleaning and exploring |  |  |  |
|  |  |  | <https://github.com/NathanEpstein/Dora>  Cleansing functions include:   * Reading data with missing and poorly scaled values * Imputing missing values * Scaling values of input variables |
| Data cleaning |  |  | <https://github.com/rhiever/datacleaner> |
| data-cleaning-with-python/ |  |  | https://www.dataquest.io/blog/data-cleaning-with-python/ |
|  |  |  |  |

# Data time

|  |  |  |  |
| --- | --- | --- | --- |
| Arrow /dat-time manipulation |  |  | <http://arrow.readthedocs.io/en/latest/> |
| Downsample /upsample |  |  | 2+ times series against each other |

# Anonymize data

|  |  |  |  |
| --- | --- | --- | --- |
| scrubadub |  | * Names (proper nouns) * Email addresses * URLs * Phone numbers * username/password combinations * Skype usernames * Social security numbers | <http://scrubadub.readthedocs.io/en/stable/index.html> |
|  |  |  | <https://github.com/krasch/presentations/blob/master/pydata_Berlin_2016.pdf> |

# Statistics

|  |  |  |  |
| --- | --- | --- | --- |
| 2 samples/ covariance |  |  | http://nbviewer.jupyter.org/github/lmarti/evolutionary-computation-course/blob/master/AEC.04%20-%20Evolutionary%20Strategies%20and%20Covariance%20Matrix%20Adaptation.ipynb |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Probability

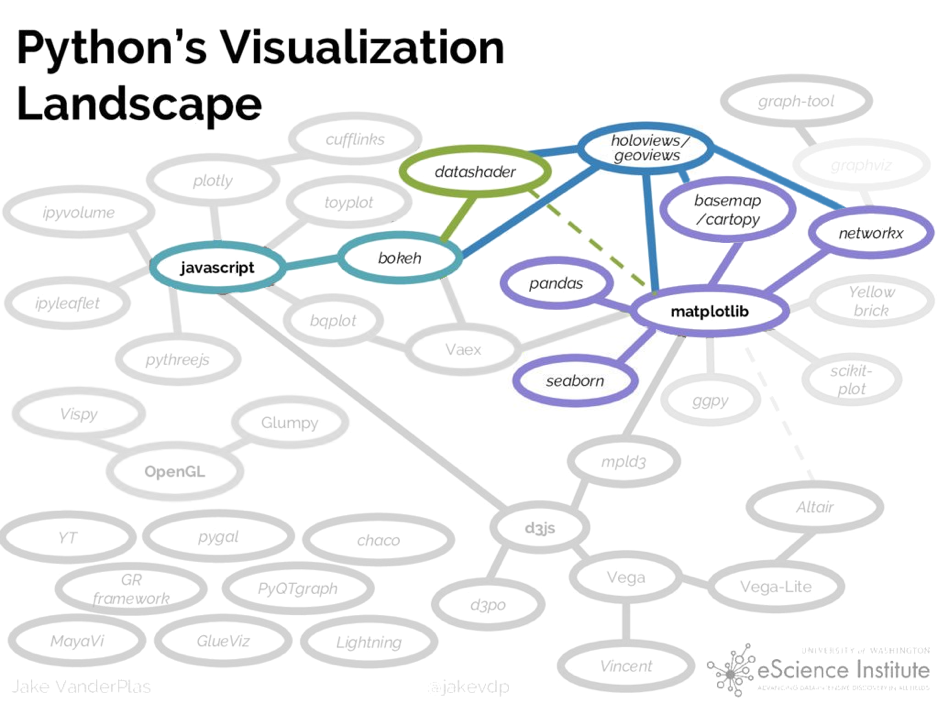
|  |  |  |  |
| --- | --- | --- | --- |
| Discrete random |  |  | http://nbviewer.jupyter.org/github/mbakker7/exploratory\_computing\_with\_python/blob/master/notebook9\_discrete\_random\_variables/py\_exploratory\_comp\_9\_sol.ipynb |
| Continuous random |  |  | http://nbviewer.jupyter.org/github/mbakker7/exploratory\_computing\_with\_python/blob/master/notebook10\_continuous\_random\_variables/py\_exploratory\_comp\_10\_sol.ipynb |
| Hypothesis |  |  | http://nbviewer.jupyter.org/github/mbakker7/exploratory\_computing\_with\_python/blob/master/notebook11\_hypothesis\_test/py\_exploratory\_comp\_11\_sol.ipynb |

# Test If normal for outlier tests

# Visualization

PYVIZ

<https://pyviz.github.io/pyviz/tutorial/>



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# OTHER:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Processed to |  |  |
| General plotting | API Figures and Subplots[¶](http://nbviewer.jupyter.org/github/pydata/pydata-book/blob/2nd-edition/ch09.ipynb#Figures-and-Subplots)Line PlotsBar PlotsHistograms and Density PlotsScatter or Point PlotsFacet Grids and Categorical DataOther Python Visualization Tools[¶](http://nbviewer.jupyter.org/github/pydata/pydata-book/blob/2nd-edition/ch09.ipynb#Other-Python-Visualization-Tools) |  | <http://nbviewer.jupyter.org/github/pydata/pydata-book/blob/2nd-edition/ch09.ipynb#> |
| Plot two categories against two continuous variables |  | Prettyplotlib | https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
| Plot categories by numerical variables |  |  |  |
|  | Plot CEC /Sic by consumption and price /kWhr |  | Statsmodels |
| Smoothed plot |  |  | df = pd.DataFrame({  https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
|  | Tutorial – daily average consumption |  | df = pd.DataFrame({  https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
| Network graph |  |  | G=nx.random\_geometric\_graph(200,0.125)  https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
| Network graph |  |  | https://anaconda.org/anaconda-enterprise/interactive-dashboards-and-charts/notebook |
|  | Total load by structure by time |  | https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
| Network - weighted |  |  | mport networkx as nx  https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
| Time-series  High & low variances | candlestick |  | https://anaconda.org/ijstokes/pythondataviz-ipynb/notebook |
| 3d plot |  |  | Matplotlib 3D Static Rendering https://anaconda.org/ijstokes/python-dataviz-3h/notebook |
|  | 3d plot heatmap consumption – aggregate day, week, month |  | Matplotlib 3D Static Rendering https://anaconda.org/ijstokes/python-dataviz-3h/notebook |
| Plot categoriacal vs continuous |  |  | http://nbviewer.jupyter.org/github/mbakker7/exploratory\_computing\_with\_python/blob/master/notebook8\_pandas/py\_exploratory\_comp\_8\_sol.ipynb |
| Plot 3 axis |  |  | http://nbviewer.jupyter.org/github/mbakker7/exploratory\_computing\_with\_python/blob/master/notebook8\_pandas/py\_exploratory\_comp\_8\_sol.ipynb |
| Aggregate and plot time-series  (week, month, year) |  |  | import numpy as np  measles\_by\_year = df[["Year","measles"]].groupby("Year").aggregate(np.sum)  measles\_by\_year.plot();  <https://anaconda.org/jbednar/exploring_data/notebook>  [dataset of the number of cases of measles and pertussis (per 100,000 people) over time in each state](http://graphics.wsj.com/infectious-df-and-vaccines/#b02g20t20w15):  df = pd.read\_csv('data/diseases.csv.gz') |
|  | Add interactive data |  | By default, the tools below ignore the Pandas index, so we'll make it into a real column for the rest of this notebook:  measles\_by\_year = measles\_by\_year.reset\_index()  hv.Curve(measles\_by\_year) |
|  | Add lines and text |  | m = hv.Curve(measles\_by\_year) \* hv.VLine(1963) \* hv.Text(1963, 27000, " Vaccine introduced", halign='left')  m |
|  | Drop down menu |  | ds = hv.Dataset(df, ['Year', 'State'], 'measles').aggregate(function=np.sum)  measles\_by\_state = ds.to(hv.Curve, 'Year', 'measles')  measles\_by\_state \* hv.VLine(1963) |
|  | Look at facet (season -weekday by weekend) By customer on top 10 list |  | https://anaconda.org/jbednar/exploring\_data/notebook |
|  | Facet |  | %%opts Curve [width=200, height=100]  measles\_by\_state.select(State=states, Year=(1930, 2005)).grid('State') \* hv.VLine(1963)  a |
|  | Add error bands on trend |  | agg = ds.aggregate('Year', function=np.mean, spreadfn=np.std)  (hv.Curve(agg) \* hv.ErrorBars(agg,vdims=['measles', 'measles\_std'])).redim.range(measles=(0, None)) \* hv.VLine(1963) |
|  |  |  |  |
|  | Categorical /heat map by Category  By time |  | url = 'https://raw.githubusercontent.com/blmoore/blogR/master/data/measles\_incidence.csv'  data = pd.read\_csv(url, skiprows=2, na\_values='-')  yearly\_data = data.drop('WEEK', axis=1).groupby('YEAR').sum().reset\_index()  measles = pd.melt(yearly\_data, id\_vars=['YEAR'], var\_name='State', value\_name='Incidence')  heatmap = hv.HeatMap(measles, label='Measles Incidence')  aggregate = hv.Dataset(heatmap).aggregate('YEAR', np.mean, np.std)  marker = hv.Text(1963, 800, u'\u2193 Vaccine introduced', halign='left')  agg = hv.ErrorBars(aggregate) \* hv.Curve(aggregate).opts(plot=dict(xrotation=90))  hm\_opts = dict(width=900, height=500, tools=['hover'], logz=True, invert\_yaxis=True,  xrotation=90, labelled=[], toolbar='above', xaxis=None)  overlay\_opts = dict(width=900, height=200, show\_title=False)  vline\_opts = dict(line\_color='black')  opts = {'HeatMap': {'plot': hm\_opts}, 'Overlay': {'plot': overlay\_opts}, 'VLine': {'style': vline\_opts}}  (heatmap + agg \* marker).opts(opts).cols(1) |
|  | Average price paid per month Segment  -customer |  | heatmap |
| Time-series  Data-time | Date and Time Data Types Calculations Date and Time Data Types |  | http://nbviewer.jupyter.org/github/pydata/pydata-book/blob/2nd-edition/ch11.ipynb |
| Decomposition of time-series | Seasonal, | Trend increasing | <https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-visualization-with-python-3> |
| Plot year over year | Minimum Daily Temperature Yearly Line Plots | Behavoiur changing? | from pandas import Series  from pandas import DataFrame  from pandas import TimeGrouper  from matplotlib import pyplot  series = Series.from\_csv('daily-minimum-temperatures.csv', header=0)  groups = series.groupby(TimeGrouper('A'))  years = DataFrame()  for name, group in groups:  years[name.year] = group.values  years.plot(subplots=True, legend=False)  pyplot.show()  <https://machinelearningmastery.com/time-series-data-visualization-with-python/> |
| 2+ times series against each other |  |  | https://anaconda.org/chezka-sino/climate\_timeseries-part1/notebook |
| 2+ times series against each other |  |  | https://anaconda.org/defusco/comparing-time-series/notebook |
|  | ARIMA |  | https://anaconda.org/anaconda-enterprise/time-series/notebook |
|  | Customer prediction for next year (3 years data) |  | https://anaconda.org/anaconda-enterprise/time-series/notebook |
| Smoothed vs discrete |  |  | https://demo.bokehplots.com/apps/weather |
| Time changing bubbleplots |  | kW – demand change vs KW-base | https://demo.bokehplots.com/apps/gapminder |
| Map-heatmap |  | Changing load per week, hour, year | http://bokeh.pydata.org/en/latest/docs/gallery/texas.html |
| Rolling window |  |  | https://anaconda.org/jbednar/stock\_dashboard/notebook |
| Categorical data by groups |  |  | https://bokeh.pydata.org/en/latest/docs/user\_guide/categorical.html#userguide-categorical |
| Time spreads by period |  |  | https://bokeh.pydata.org/en/latest/docs/user\_guide/categorical.html#userguide-categorical |
| Widgets (Bokeh) |  |  | <https://bokeh.pydata.org/en/latest/docs/user_guide/interaction/widgets.html#userguide-interaction-widgets> |
| Slice anddice time-series with stats |  |  | <http://holoviews.org/reference/streams/bokeh/BoundsX.html>  https://raw.githubusercontent.com/ioam/holoviews/master/examples/reference/streams/bokeh/BoundsX.ipynb |
| Contour |  |  | https://notebooks.azure.com/garth-wells/libraries/CUED-IA-Computing-Michaelmas/html/08%20Plotting.ipynb |
| Pairwise plots | HoloViews + Bokeh statistical plots |  | <https://anaconda.org/jbednar/exploring_data/notebook> |
| MAP – many categories by location | Houston district 29 |  | <https://anaconda.org/jbednar/gerrymandering/notebook> |
| MAP - heatmpa |  |  | <https://anaconda.org/bfiedler/timeseries/notebook> |
| MAP |  |  | Many data <https://anaconda.org/jbednar/nyc_taxi/notebook> |
| Graph viz |  |  | <https://anaconda.org/jobbyworld/testgraphvizcloud/notebook> |
| Draw diagram and plot over it |  |  | https://anaconda.org/kaophet/data-viz-draft-3-bokeh-shot-chart-interactive/notebook |
| Rangefinder with plotly |  |  | https://anaconda.org/elkeerdei/time-series/notebook |
| Dashboards – multiple slice and dice |  |  | https://anaconda.org/ashermeyers/seattle\_historical\_dashboard/notebook |
| 3d |  |  | http://nbviewer.jupyter.org/github/lmarti/evolutionary-computation-course/blob/master/AEC.04%20-%20Evolutionary%20Strategies%20and%20Covariance%20Matrix%20Adaptation.ipynb |
| 3d |  |  | https://notebooks.azure.com/garth-wells/libraries/CUED-IA-Computing-Michaelmas/html/08%20Plotting.ipynb |
|  |  |  |  |
|  |  |  |  |

# Tables

|  |  |  |  |
| --- | --- | --- | --- |
| Tables |  |  | <https://github.com/HHammond/PrettyPandas> |
| Tables |  |  | <https://pypi.python.org/pypi/tabulate> |

# Vizualization Servers

Dashboards using Bokeh

<http://go.continuum.io/supercharge-your-data-science-team/>

|  |  |  |  |
| --- | --- | --- | --- |
| Interactive web apps |  |  | <https://github.com/dgrtwo/gleam> |
| Bokeh Server |  |  | https://bokeh.pydata.org/en/latest/docs/user\_guide/server.html#userguide-server |
|  |  |  |  |
| Pyxley |  |  | https://github.com/stitchfix/pyxley |
| pubnub |  |  | https://www.pubnub.com/tutorials/javascript/realtime-dashboard-with-python-and-epoch/ |
| Shiny |  |  | conda install -c r r-shinydashboard |

Amazing dashboarding/visualization resources

<https://github.com/obazoud/awesome-dashboard#graphite>

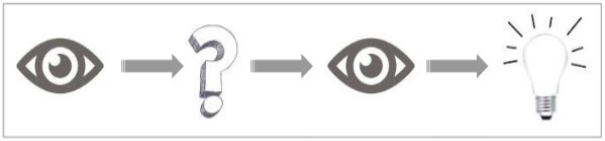
<https://opensource.com/business/16/11/open-source-dashboard-tools-visualizing-data>

Graphana

<https://grafana.com/cloud/grafana>

# Visualization

1. Explain
2. Explore



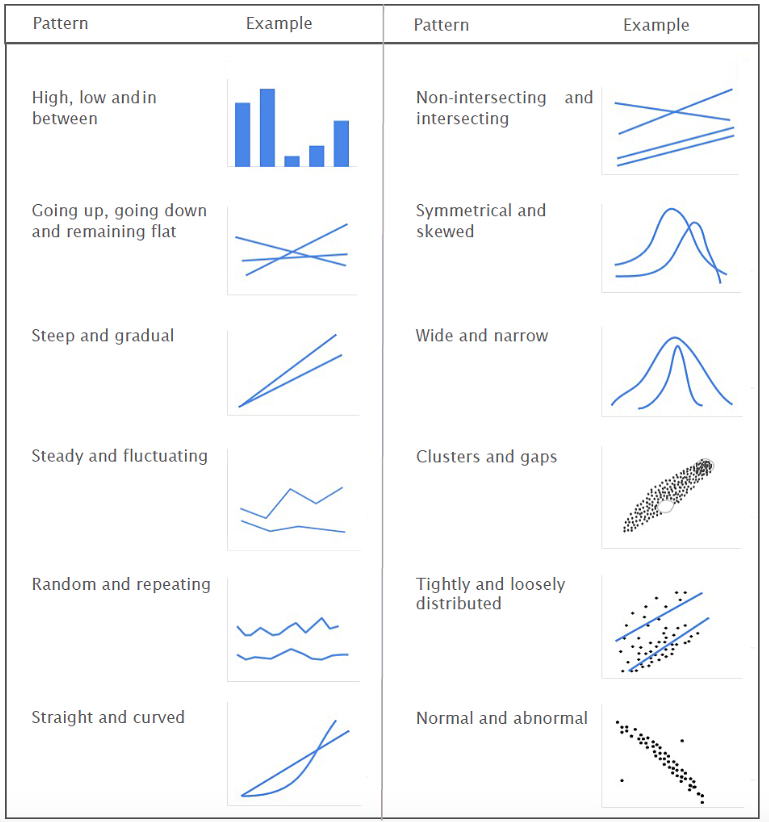


Explain

* Answer a question. E.g., How much sales did we have last quarter?
* Support a decision. E.g., We need to stock more football jerseys as they were sold out on most days last week
* Communicate information. E.g., Revenue is on track for this quarter
* Increase efficiency. E.g., ‘Technical specifications’ is the most viewed section in the product
* page. It should be given more visibility.
* Tell a story – time domain

Explore

* Pose new questions
* Explore and discover



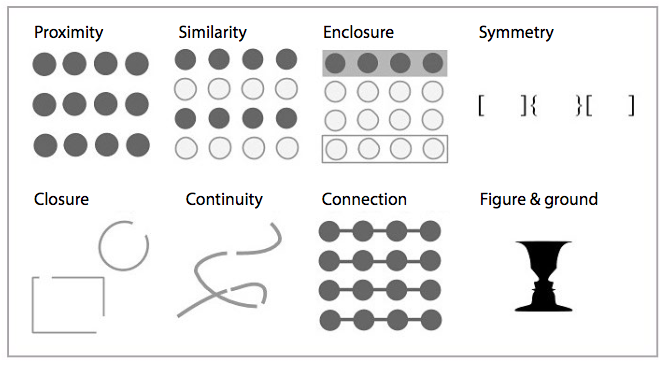
Using the Gestalt Principles to Bring Out Patterns in Visualizations. Gestalt principles describe how our mind organizes individual elements into groups. We can use these principles to highlight patterns that are important, and downplay other patterns. The image below

illustrates the principles of Gestalt which are relevant to visualization.

Here’s what we notice from each of the illustrations:

* Proximity: We see three rows of dots instead of four columns of dots because they are closer horizontally than vertically.
* Similarity: We see similar looking objects as part of the same group.
* Enclosure: We group the first four and and last four dots as two rows instead of eight dots.
* Closure: We automatically close the square and circle instead of seeing three disconnected paths.
* Continuity: We see one continuous path instead of three arbitrary ones.
* Connection: We g roup the connected dots as belonging to the same group.
* Symmetry: We see three pairs of symmetrical brackets rather than six individual brackets.
* Figure & ground: We either notice the two faces, or the vase. Whichever we notice becomes the figure, and the other the ground

<https://www.fusioncharts.com/whitepapers/downloads/Principles-of-Data-Visualization.pdf>



# Algorithms

# <https://github.com/gudnm/codekatas>

# Deploy python

<http://go.continuum.io/wp-productionizing-deploying-secure-scalable-ds-projects/>

<http://go.continuum.io/download-ebook-breaking-data-science-open/>

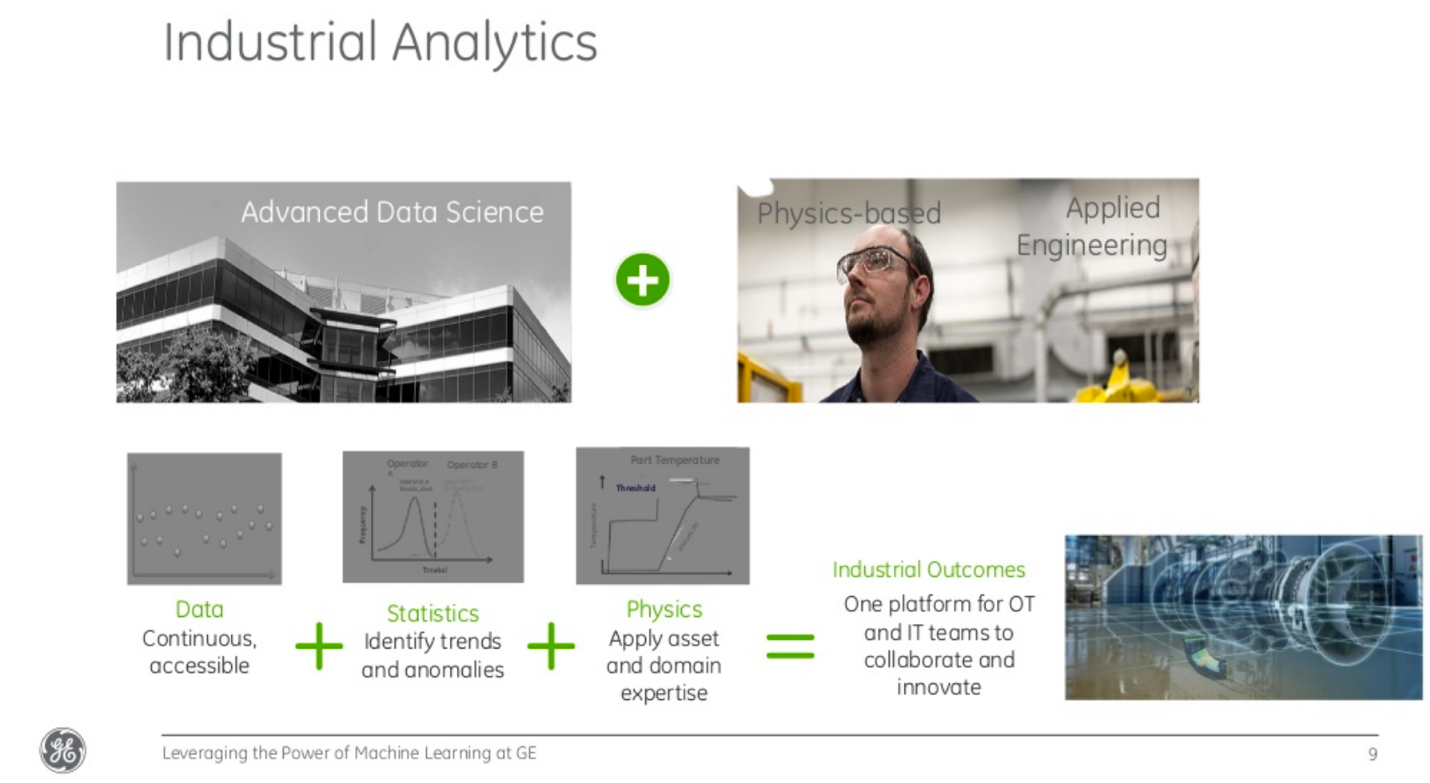
# Data science team

<http://go.continuum.io/registration-overcome-5-dysfunctions/>

# Analytics applications

Industrial Analytics

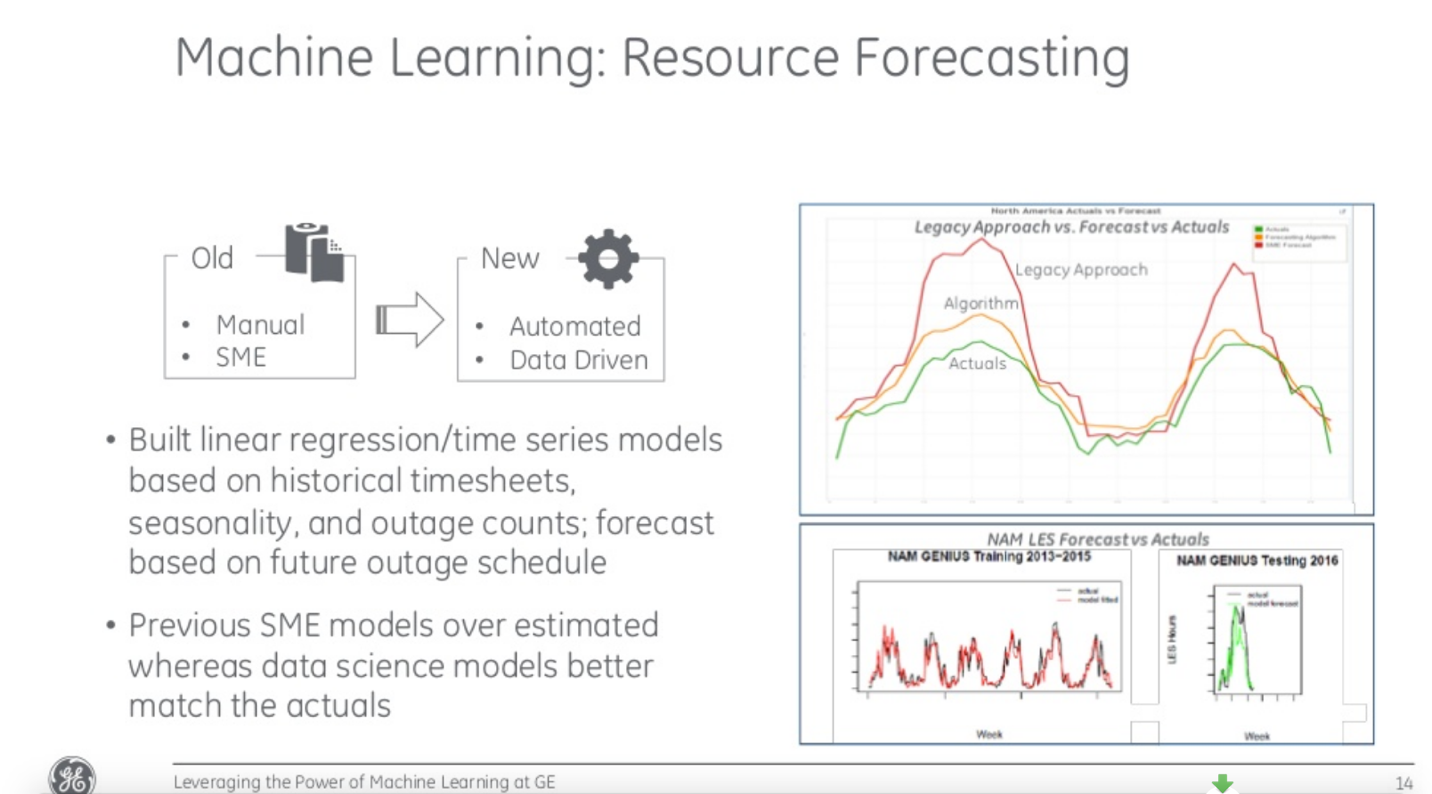
Combining data-driven and physics-based to achieve higher accuracy



<https://www.slideshare.net/continuumio/leveraging-the-power-of-machine-learning-at-ge-anacondacon-2017>

Leveraging the Power of Machine Learning at GE Girish Modgil

Improving upone manual and SME processes



<https://www.slideshare.net/continuumio/leveraging-the-power-of-machine-learning-at-ge-anacondacon-2017>

Leveraging the Power of Machine Learning at GE Girish Modgil