pandas & ggplot

quick analysis with python and friends

Vincent Warmerdam
Data Scientist

@fishnets88
vincentwarmerdam@godatadriven.com

GoDataDriven
PROUDLY PART OF THE XEBIA GROUP

Who is this guy?

- Data Scientist at GoDataDriven
- Netherlands/USA
- Former Calculus/Statistics Lecturer

- **STACK:**
  - R: ggplot, reshape
  - Python: np, pandas, scikit-learn
    flask, django
  - Javascript: d3, angularjs
  - Hadoop: hive, impala, mahout
- **SIDESTUFF**
  - scala, spark, blender, algorithms & math

How to start.

Make sure you have ipython notebook running.
http://continuum.io/downloads

Also install R/Rstudio, for extra plotting.
http://www.rstudio.com/

Also go to my blog for the .csv files.
http://koaning.com

What is this talk about?

- learning the basics of pandas
- learning the basics of visualisation with ggplot
- combining the two

Part of the talk will be slides, the other part will be done via an iPython-notebook.

Analyzing data.

Things learned in the real world.

- data is domain specific
- I like to see what steps my colleague took
- I need things quick and flexible
- lots of steps are similar
- I like to apply the DRY principle
- I prefer typing to mouseclicking

Analyzing data.

Things learned in the real world, summarised:

Analyzing data.

Other people would agree;

Switching over to python/R

Feels more like this:

Programming Analyst

Why python:

- flexible language
- large community
- performance
- tends to written quickly

Installing Pandas

From command line:

```
> pip install pandas
```

From python (notebook):

```
import pip

def install(package):
   pip.main(['install', package])

install('pandas')
```

Panda Theory

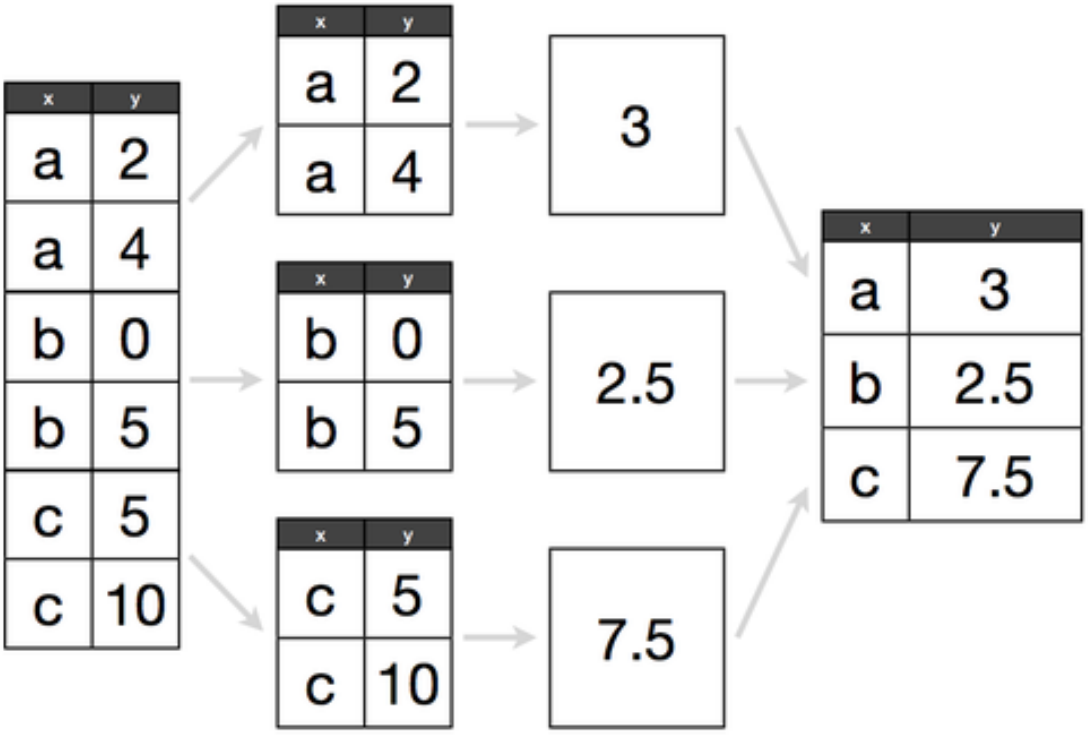
DataFrames at the core. **pa**nel **da**ta **st**ructures

Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure

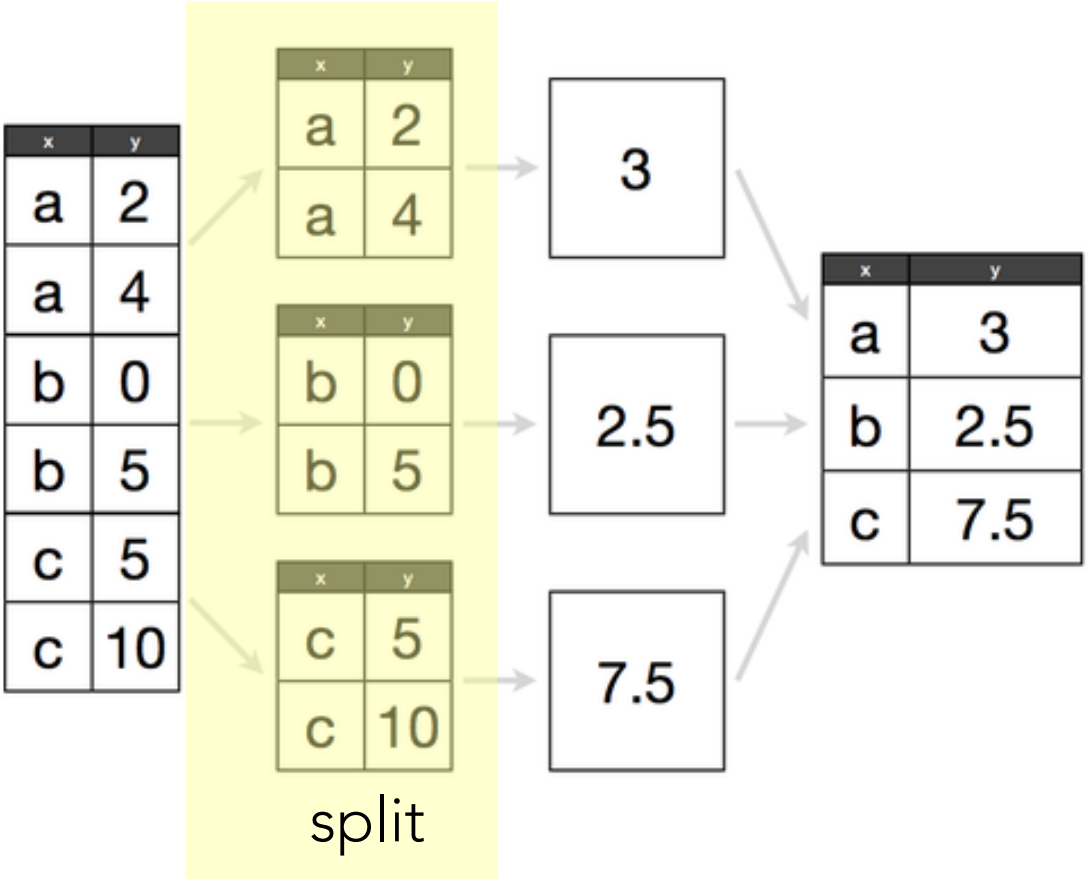
Enough talk. Code!

Start up the notebook.

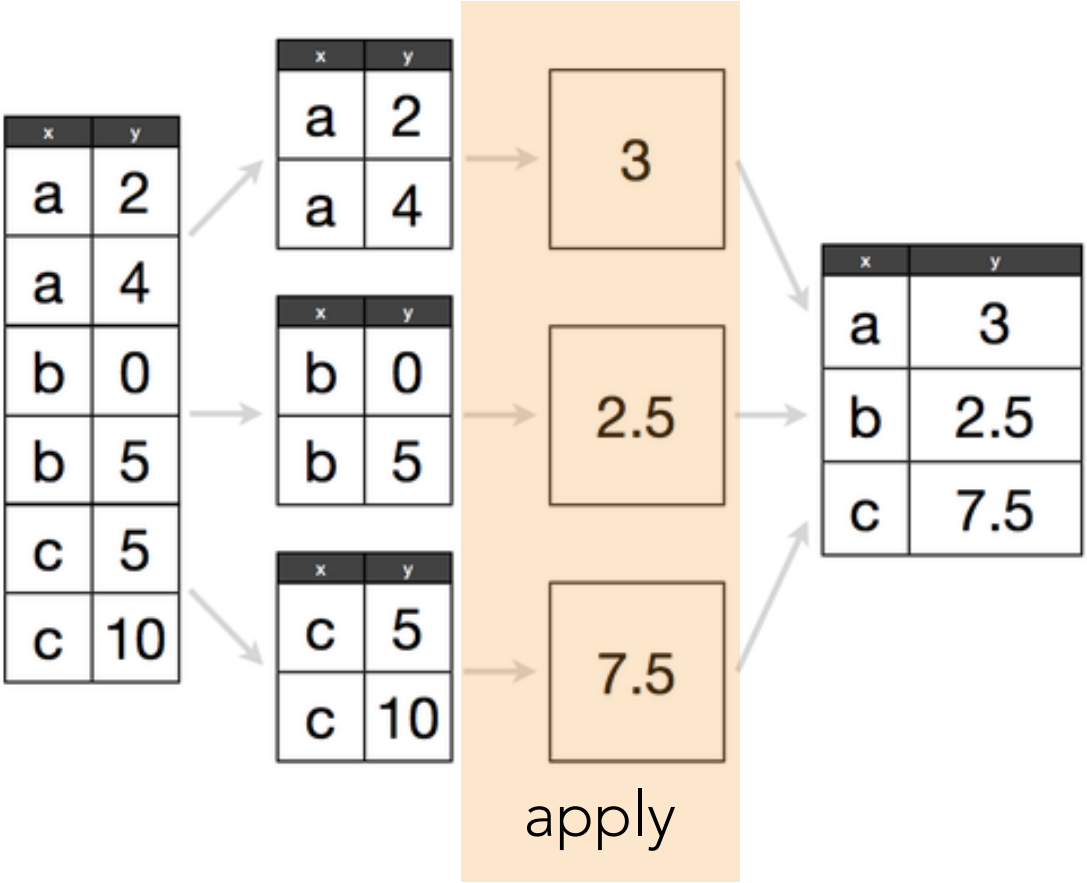
Apply Split Combine



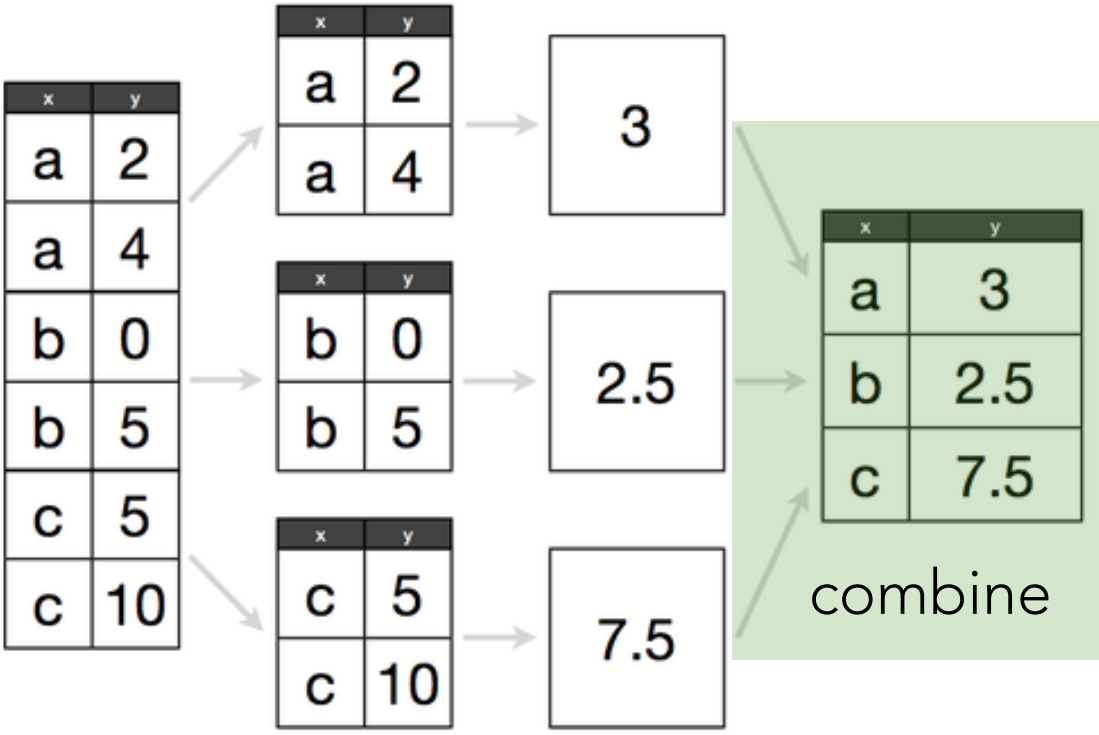
Apply Split Combine



Apply Split Combine



Apply Split Combine



Enough talk. Code!

Start up the notebook.

So why not python?

Why python:

- flexible language
- large community
- performance
- tends to written quickly

Why not python:

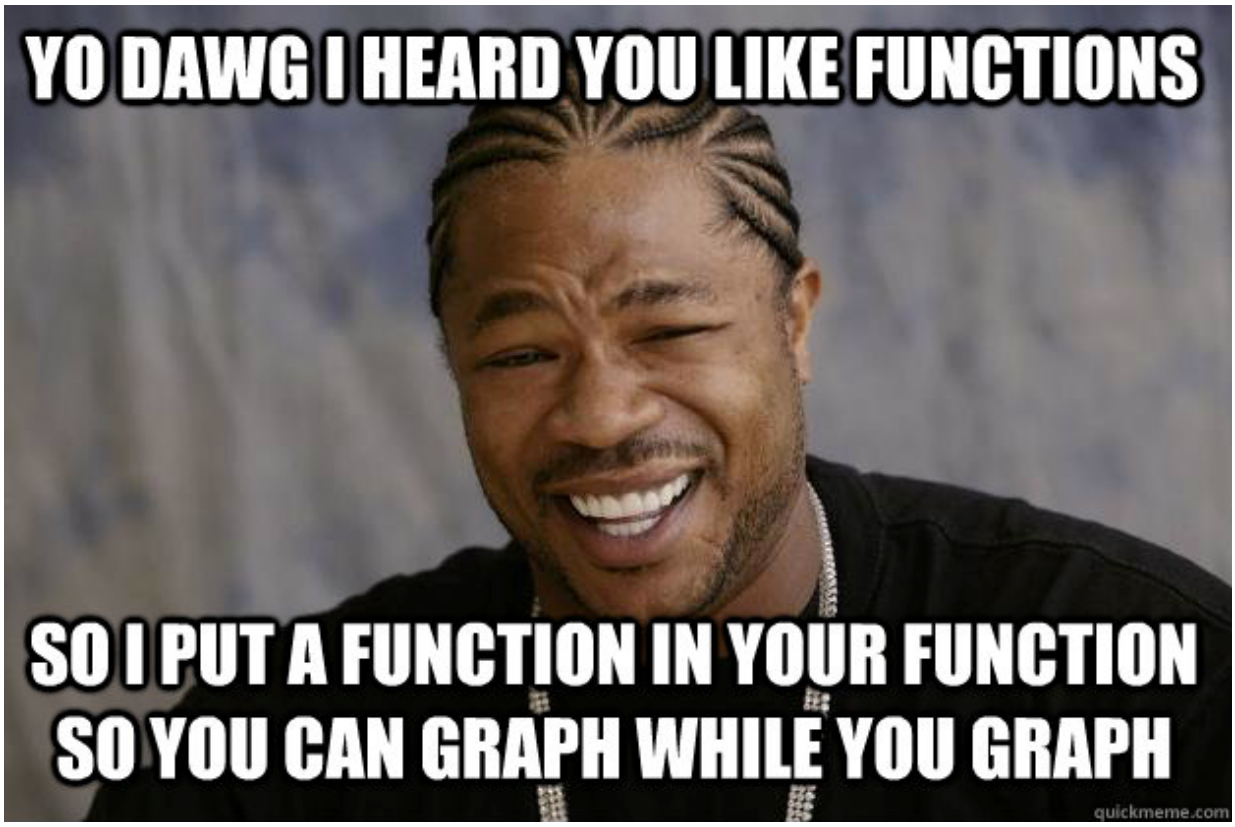
- ggplot2

Matplotlib

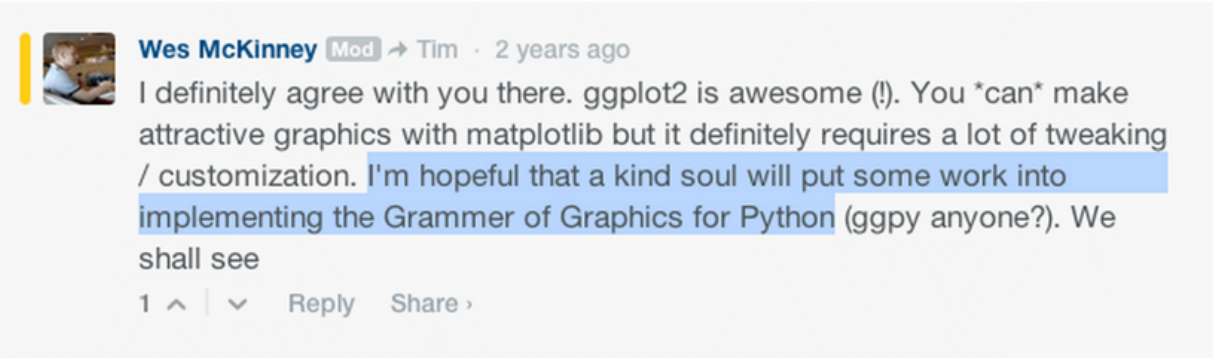
This is the code needed to plot a linechart.

```
#!/usr/bin/env python
from pylab import *
#from matplotlib.pyplot import *
#from numpy import arange

if 1:
    figure(figsize=(7, 4))
    ax = subplot(121)
    ax.set_aspect(1)
    plot(arange(10))
    xlabel('this is a xlabel\n(with newlines!)')
    ylabel('this is vertical\ntest', multialignment='center')
    #ylabel('this is another!')
    text(2, 7,'this is\nyet another test',
         rotation=45,
         horizontalalignment = 'center',
         verticalalignment   = 'top',
         multialignment      = 'center')

    grid(True)

    subplot(122)

    text(0.29, 0.7, "Mat\nTTp\n123", size=18,
         va="baseline", ha="right", multialignment="left",
         bbox=dict(fc="none"))

    text(0.34, 0.7, "Mag\nTTT\n123", size=18,
         va="baseline", ha="left", multialignment="left",
         bbox=dict(fc="none"))

    text(0.95, 0.7, "Mag\nTTT$^{A^A}$\n123", size=18,
         va="baseline", ha="right", multialignment="left",
         bbox=dict(fc="none"))

    xticks([0.2, 0.4, 0.6, 0.8, 1.],
           ["Jan\n2009","Feb\n2009","Mar\n2009", "Apr\n2009", "May\n2009"])

    axhline(0.7)
    title("test line spacing for multiline text")

subplots_adjust(bottom=0.25, top=0.8)
draw()
show()
```

Matplotlib


YO DAWG I HEARD YOU LIKE FUNCTIONS
SO I PUT A FUNCTION IN YOUR FUNCTION
SO YOU CAN GRAPH WHILE YOU GRAPH
quickmeme.com

Creators Agree



Wes McKinney Mod → Tim · 2 years ago

I definitely agree with you there. ggplot2 is awesome (!). You *can* make attractive graphics with matplotlib but it definitely requires a lot of tweaking / customization. I'm hopeful that a kind soul will put some work into implementing the Grammer of Graphics for Python (ggpy anyone?). We shall see

1 Reply Share ›

> matplotlib is powerful...but its plotting commands remain rather verbose, and its no-frills, default output looks much more like Excel circa 1993 than ggplot circa 2013. ~ Jake Vanderplas, Matplotlib & the Future of Visualization in Python (@jakevdp)

ggplot2, many better, much cool

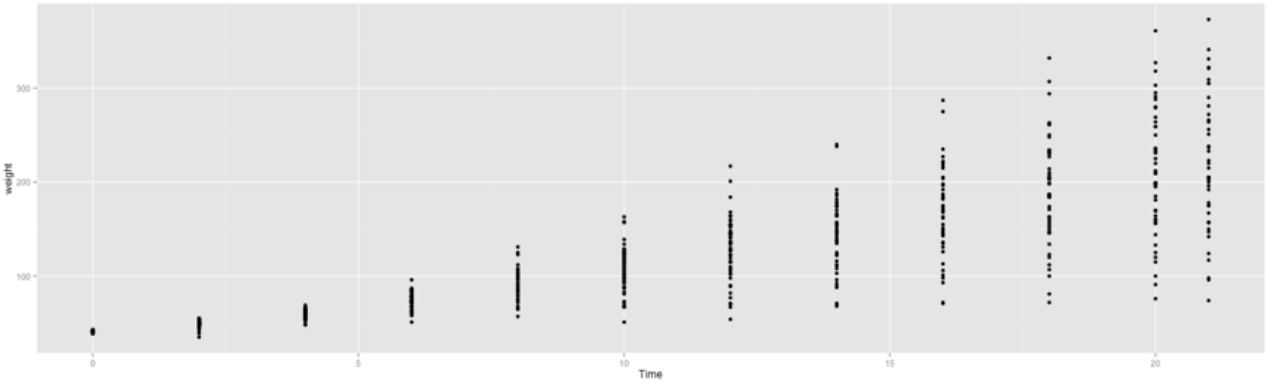
Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight)
)
```

ggplot2, many better, much cool

Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight)
)
```



ggplot2, many better, much cool

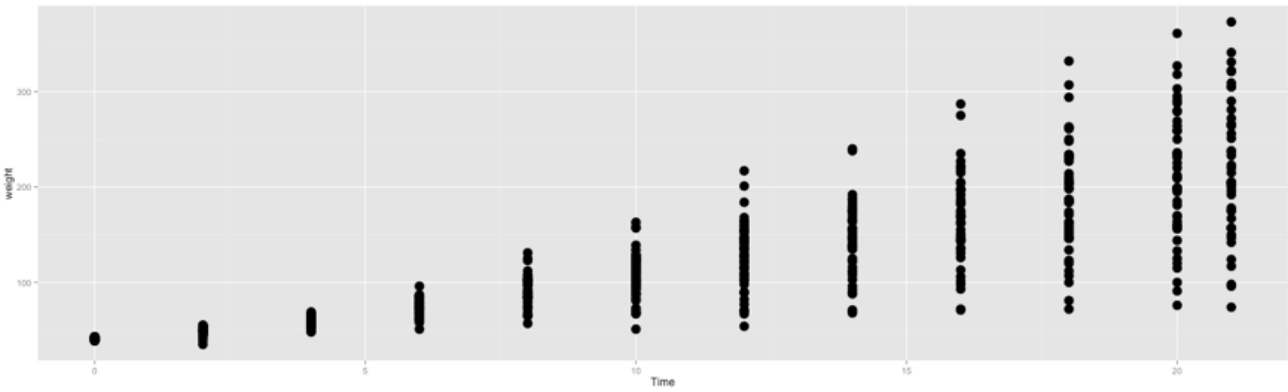
Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight), size = 5
)
```

ggplot2, many better, much cool

Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight), size = 5
)
```



ggplot2, many better, much cool

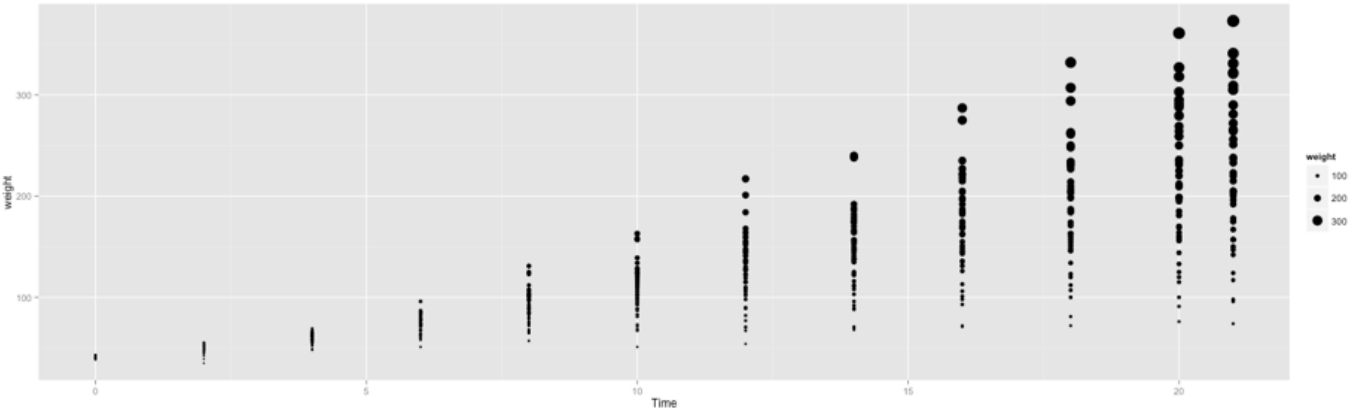
Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight, size = weight)
)
```

ggplot2, many better, much cool

Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight, size = weight)
)
```



ggplot2, many better, much cool

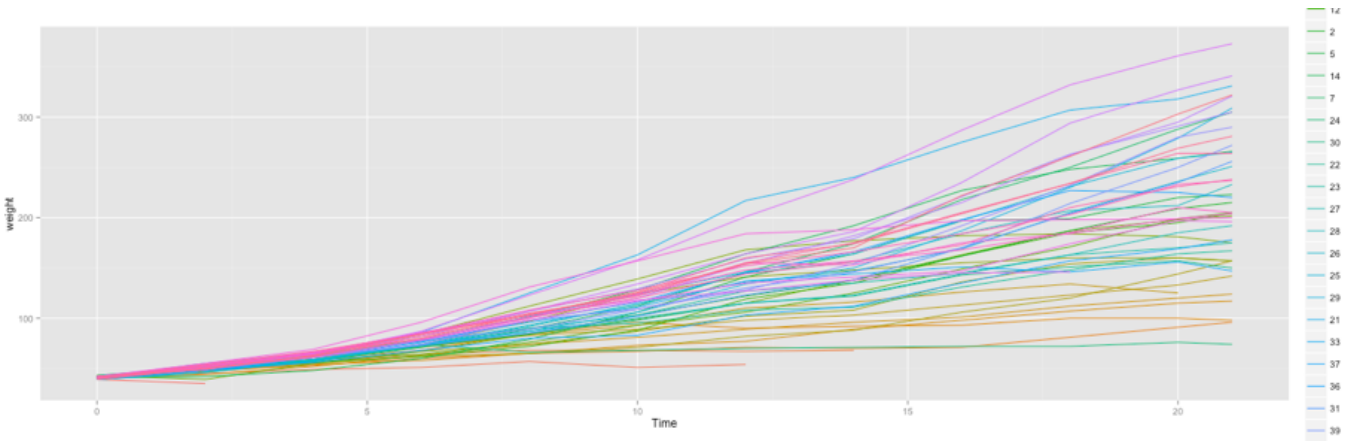
Simple example, what should this code do?

```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight, color = Chick)
)
```

ggplot2, many better, much cool

Simple example, what should this code do?

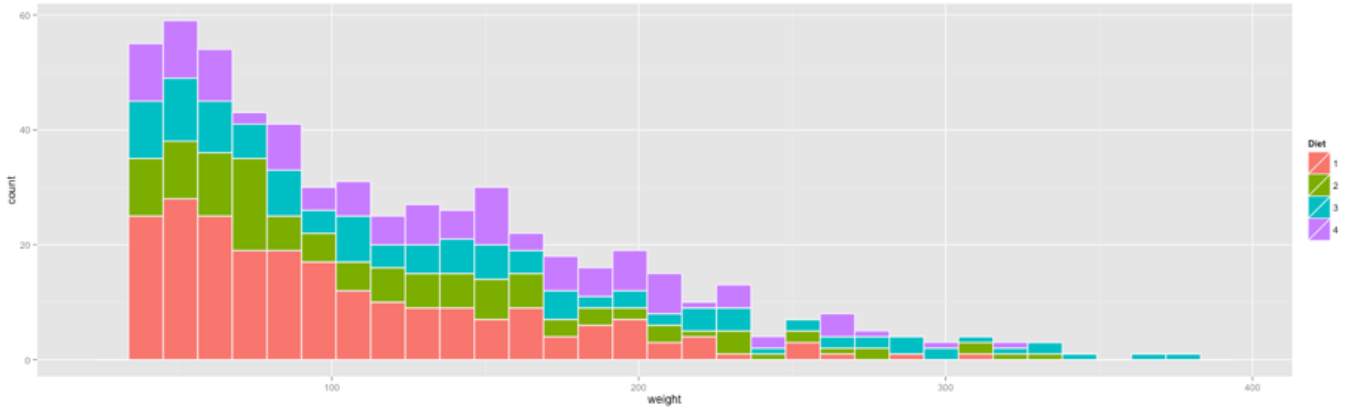
```
p = ggplot()
p + geom_point(
   data=ChickWeight,
   aes(x=Time, y=weight, color = Chick)
)
```



ggplot2, many better, much cool

Simple example, what should this code do?

```
p + geom_histogram(
   data=ChickWeight,
   aes(x=weight, fill=Diet)
)
```

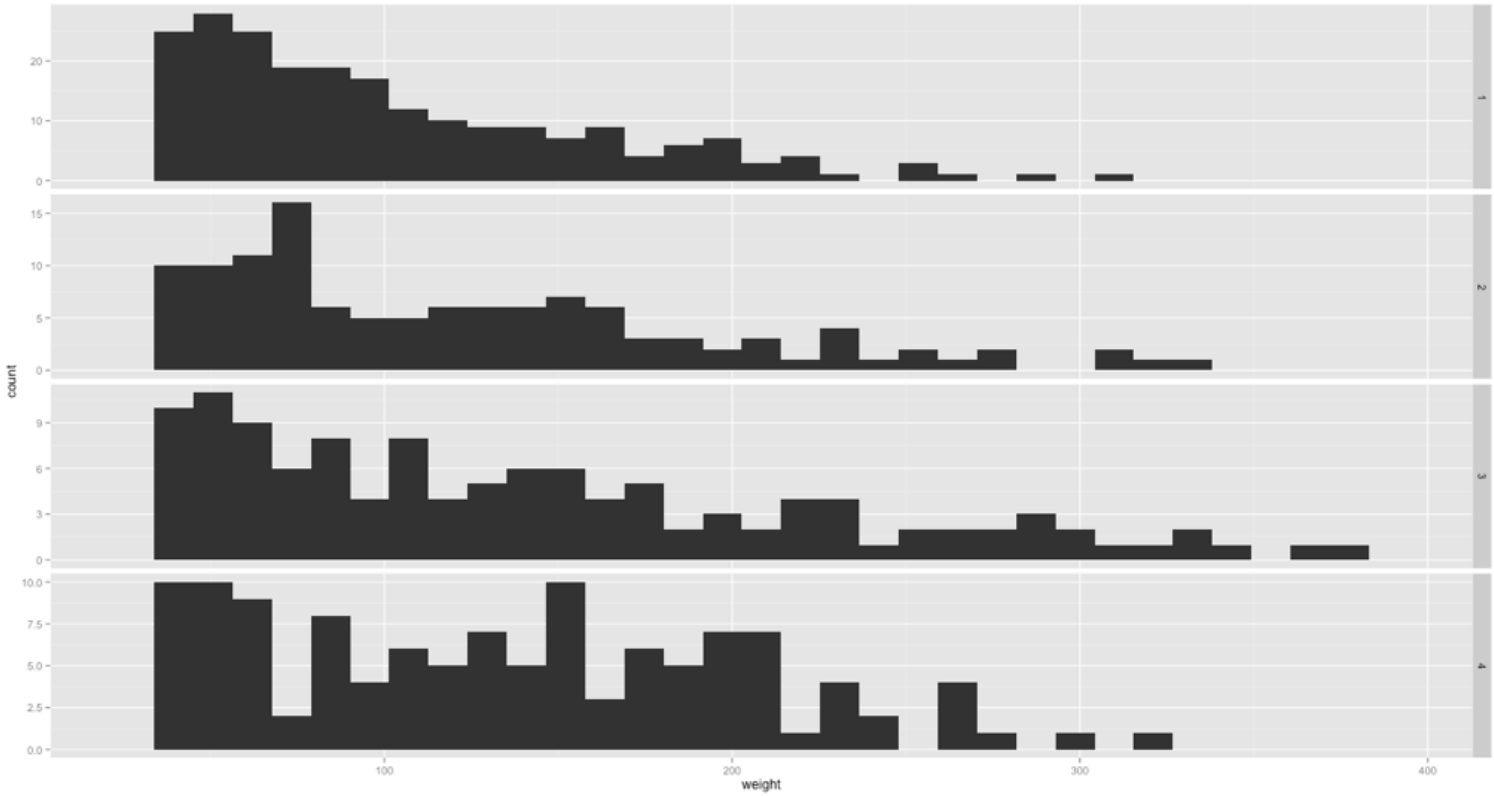


ggplot2, many better, much cool

Simple example, what should this code do?

```
p = p + geom_histogram(
   data=ChickWeight,
   aes(x=weight)
)
p = p + facet_grid( Diet ~ . , scales="free")
```

ggplot2, many better, much cool

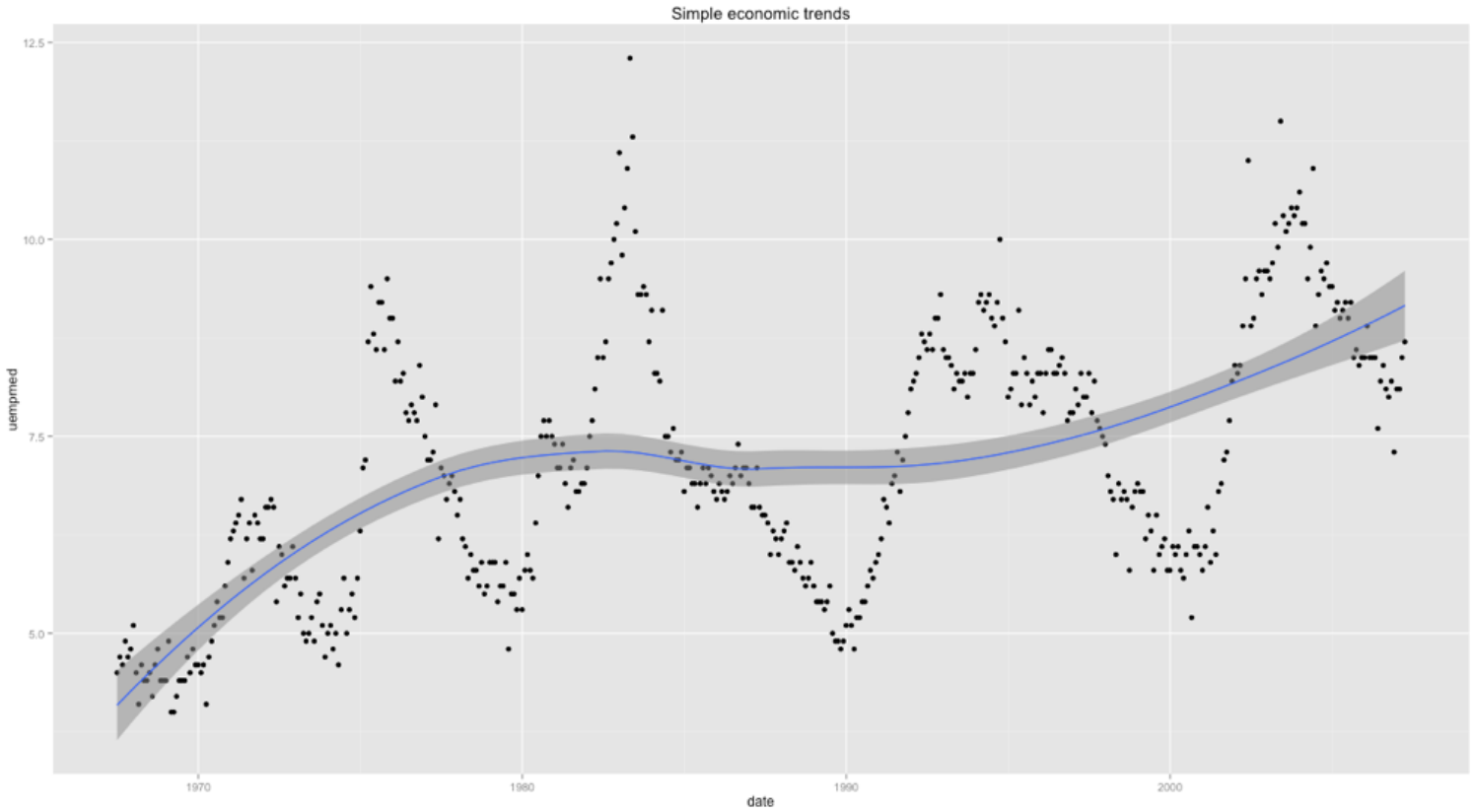


ggplot2, many better, much cool

Moar sugar!

```
p = p + geom_point(
  data=economics,
  aes(date, uempmed)
)
p = p + geom_smooth(
  data=economics,
  aes(date, uempmed)
)
p + ggtitle('Simple economic trends')
```

ggplot2, many better, much cool



Pipe Pandas to R

When R is installed, install python module

```
> pip install rpy2
```

When R is installed make sure ipython knows it

```
%load_ext rpy2.ipython
```

You can then push dataframes to R

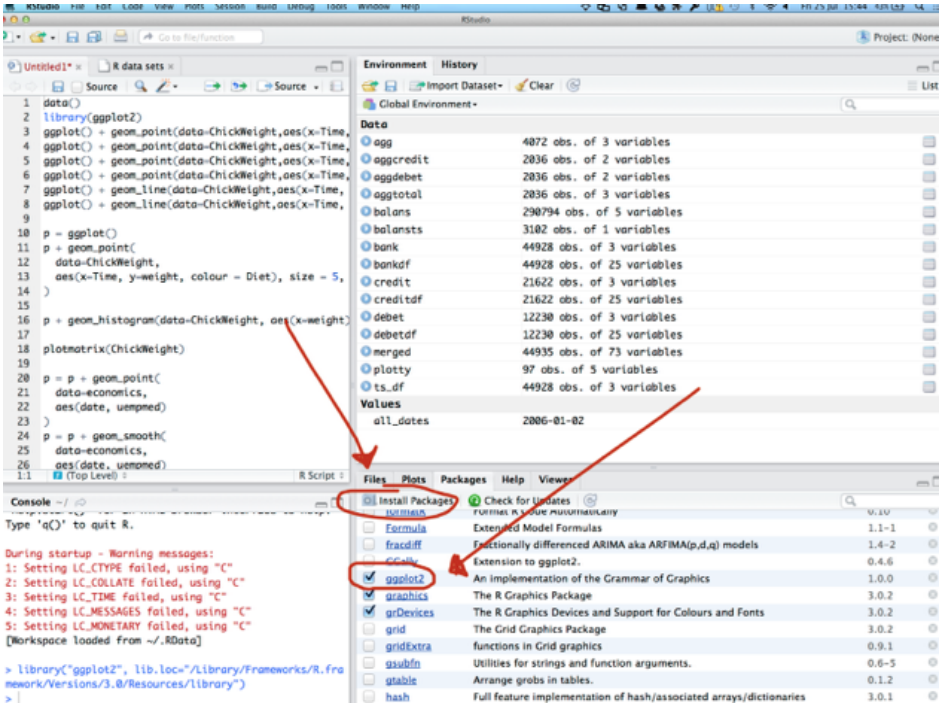
```
%Rpush df
```

And run Rcode when you start a block with

```
%%R -w 1200 -o df
```

Install ggplot2 in R

When R is installed, open R studio, install ggplot2.



Final Challenge

Go to my blog and download

```
whatweare.csv
```

Can you figure out what this dataset describes?

Moar Python Packages!

Machine Learning:

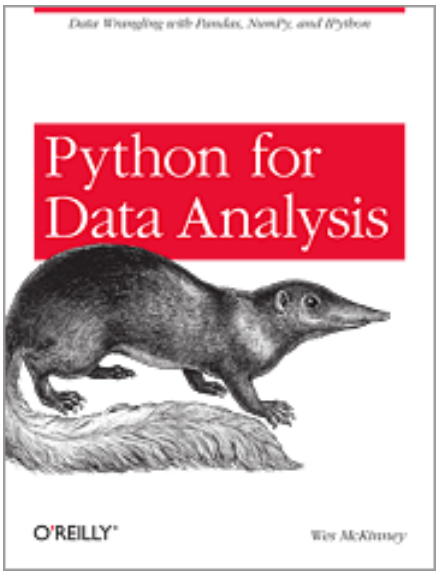
- Scikit-Learn
- PyBrain
- PyMC

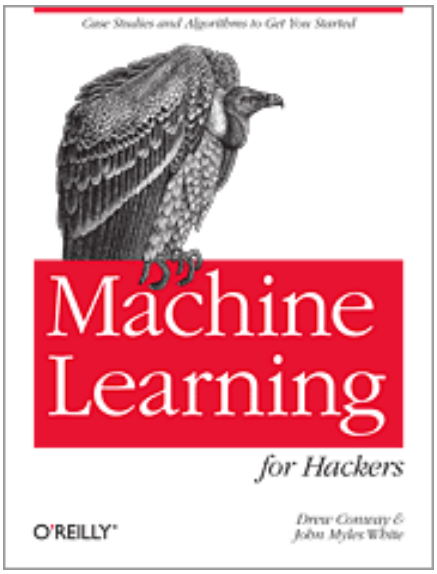
Visualisation:

- ggplot
- matplotlib
- bokeh
- vincent
- igraph

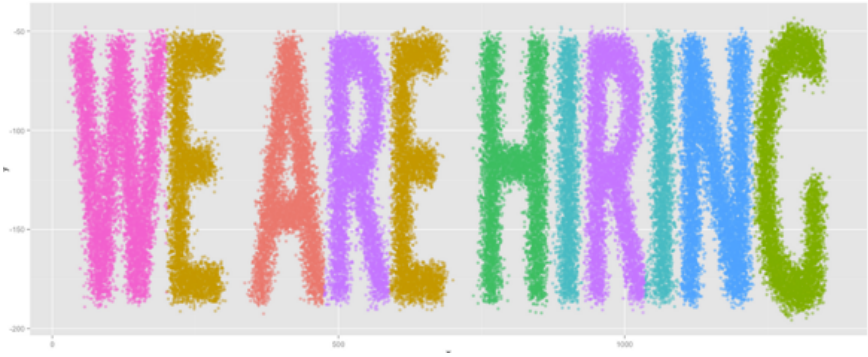
d3?

Proper Books.





GoDataDriven



Vincent Warmerdam
Data Mining Scientist

@fishnets88
vincentwarmerdam@godatadriven.com