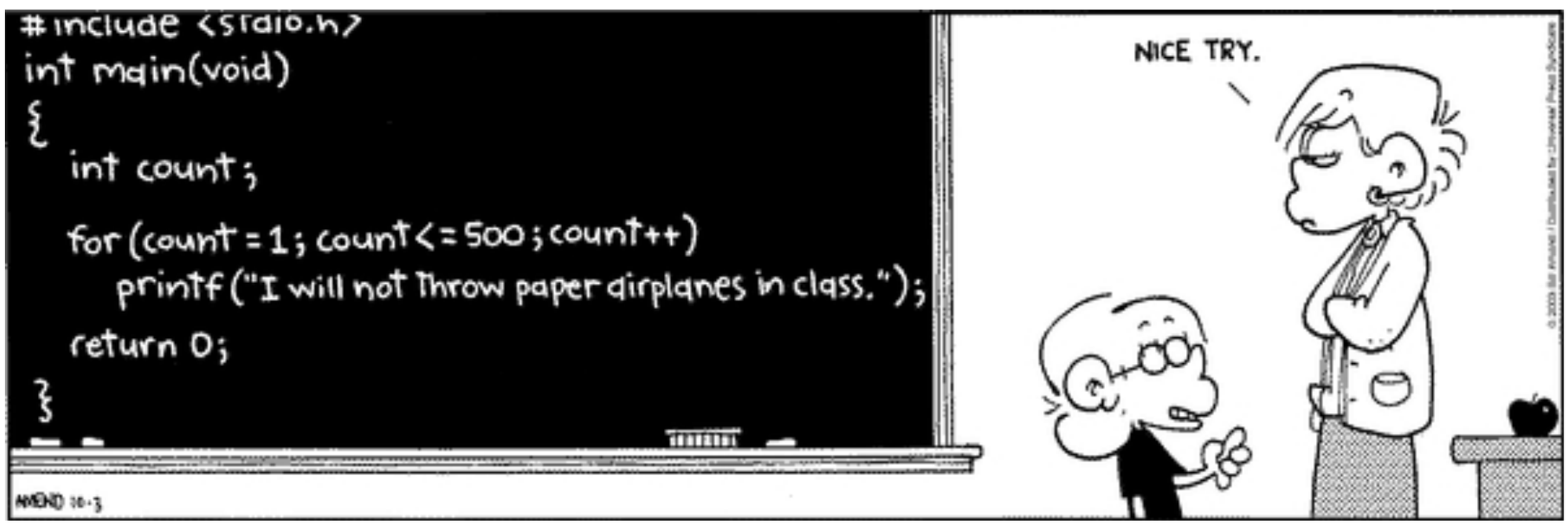
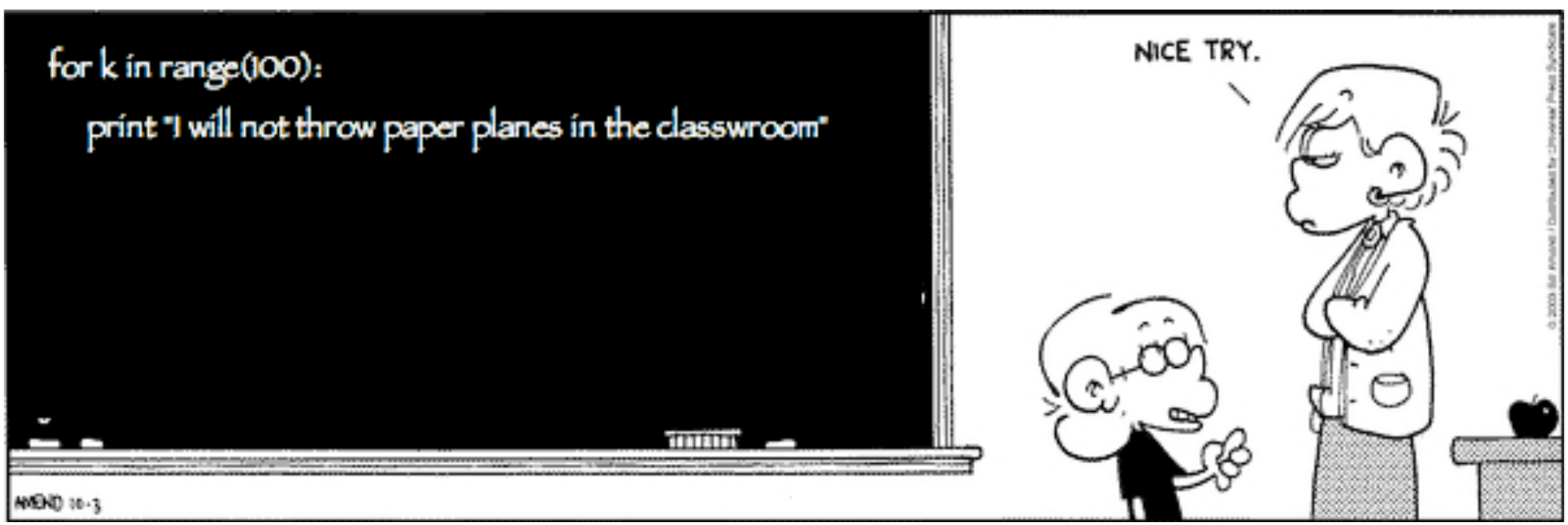
Python Training @ Spot

Massimo Di Pierro and Mark Goetsh



Python Training @ Spot

Massimo Di Pierro and Mark Goetsh



Resources

- https://dl.dropboxusercontent.com/u/18065445/Tmp/SpotTradingSlides.pdf
- https://dl.dropboxusercontent.com/u/18065445/Tmp/notes-spot.pdf
- http://www.learnpython.org/
- https://vimeo.com/20743963
- https://github.com/mdipierro/nlib
- https://github.com/web2py/web2py/blob/master/gluon/dal.py

Goals

- Week 1: Basic Python Syntax + Plotting
- Week 2: Query Data Sources
- Week 3: Practice and more advanced stuff

Why Python

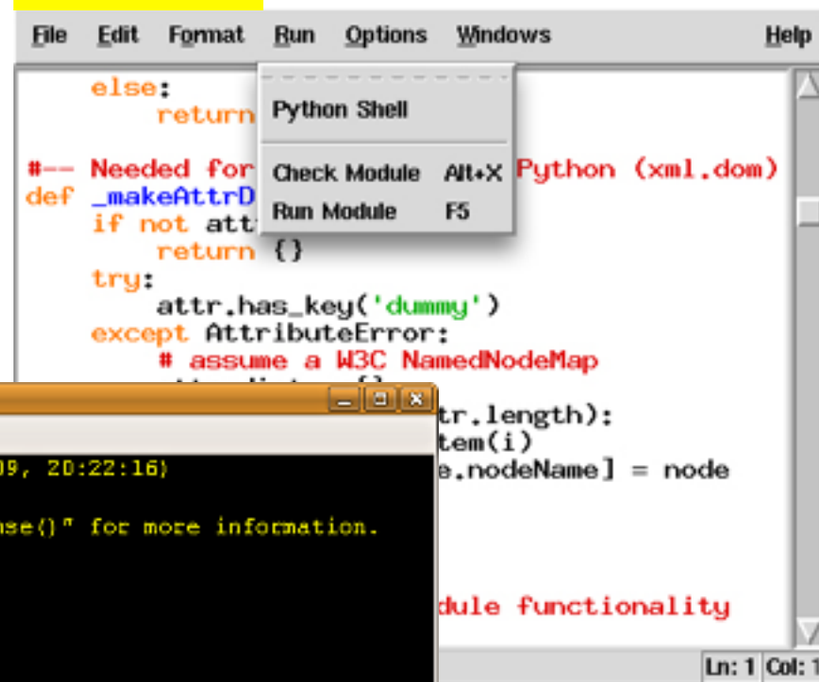
- Simple Syntax
- Very expressive (do more with less code)
- Runs everywhere
- Lots of libraries
- Almost self-documenting
- Not owned by Oracle

Python Ecosystem

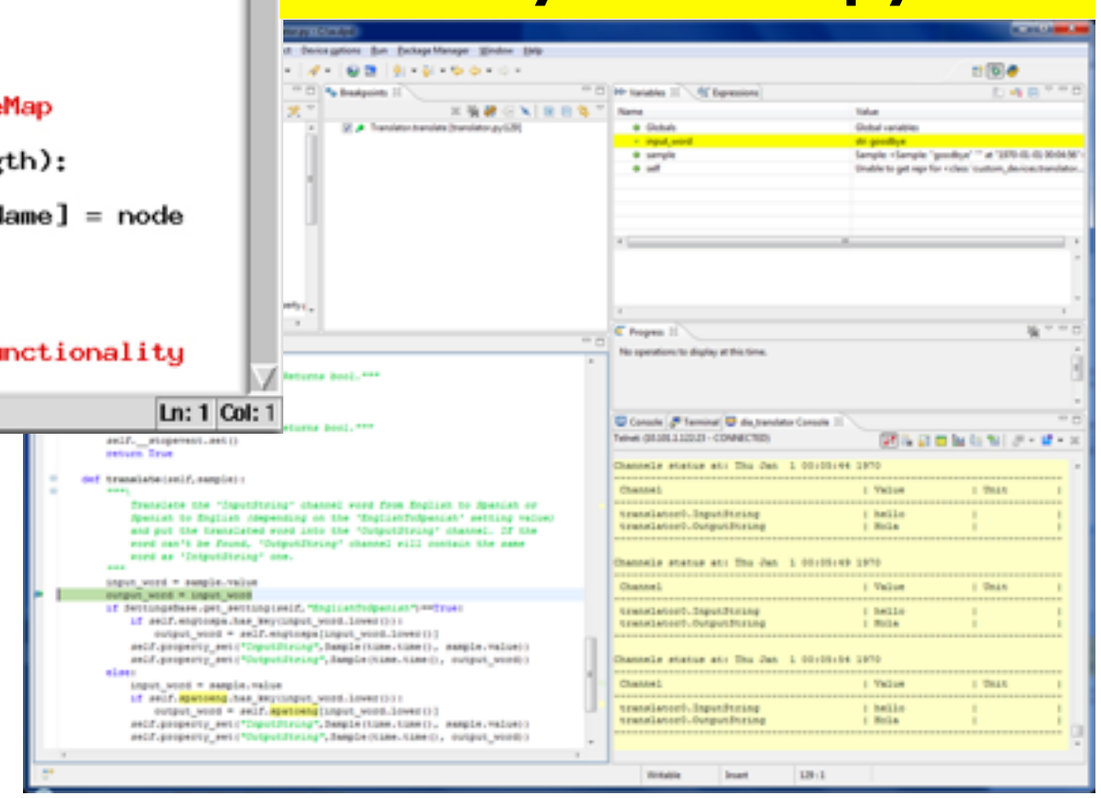
- Languages: 2.X and 3.X
- Dialects: 2.5, 2.6, 2.7
- Interpreters: CPython, Jython, PyPy, ...
- Distributions: "Official", Active State, Enthought, PythonXY, WinPython,...
- Environments: Shell, IDLE, Spyder, IPython

Shells

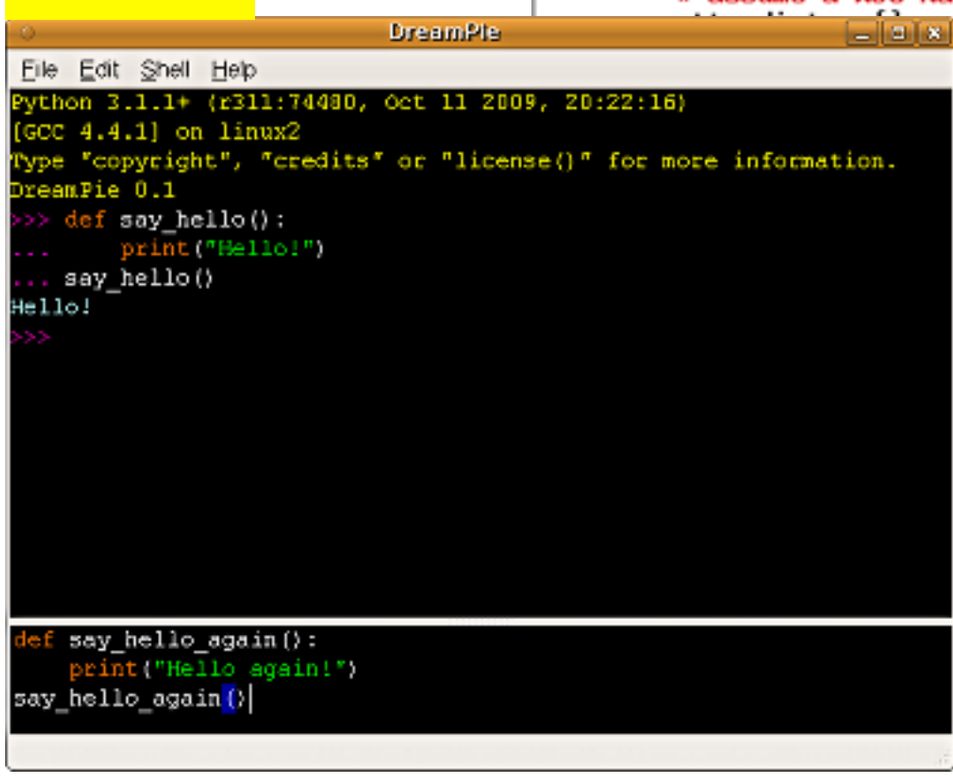
IDLE



WinPython/Spyder



shell



Python Libraries

- Many Built-in (math, datetime, ...)
- Popular (numpy, matplotlib, pyodbc,...)
- Third Party (nlib.py, dal.py, ...)

The Big Picture

Spot Data Sources

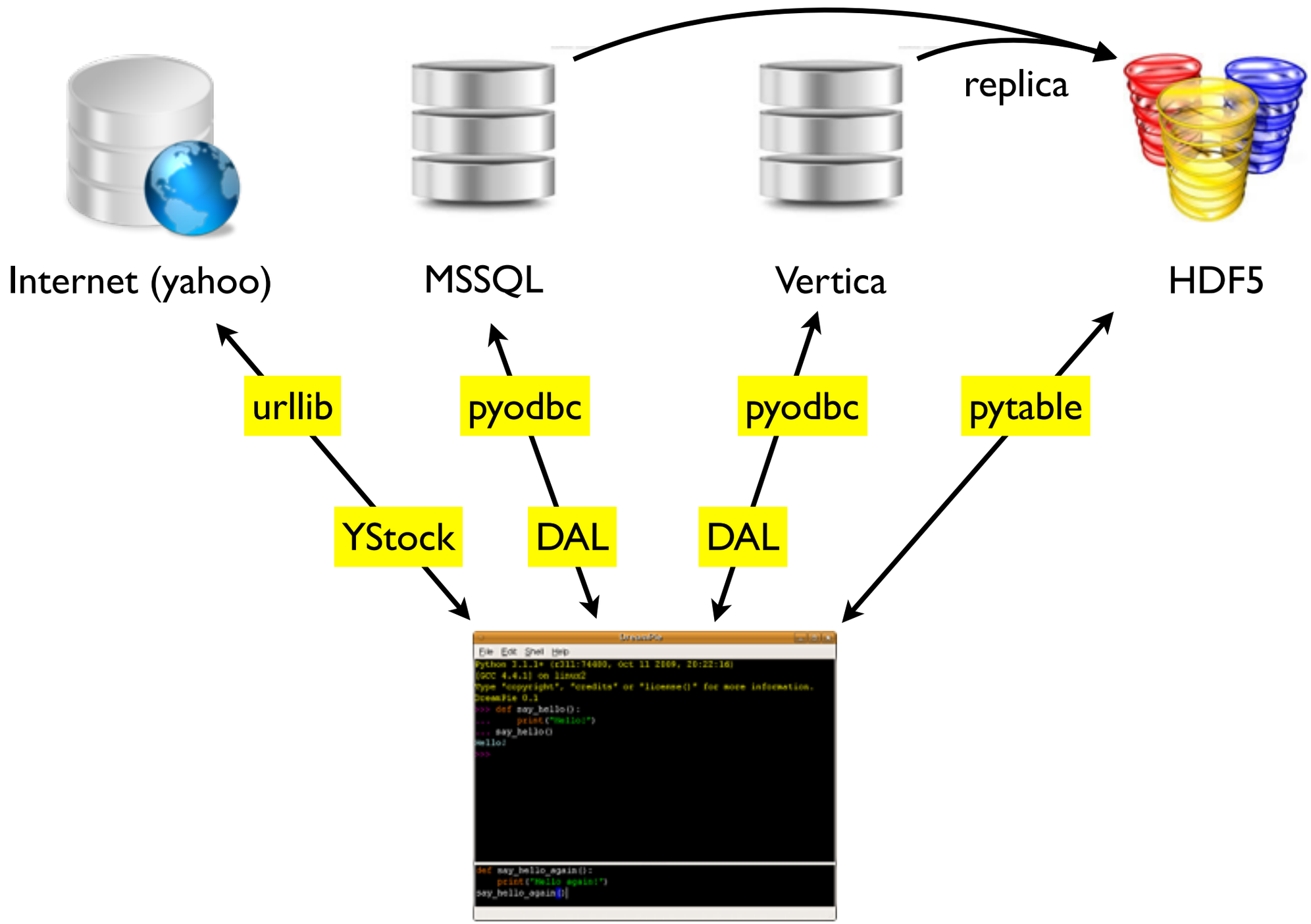
Internet (yahoo)

MSSQL

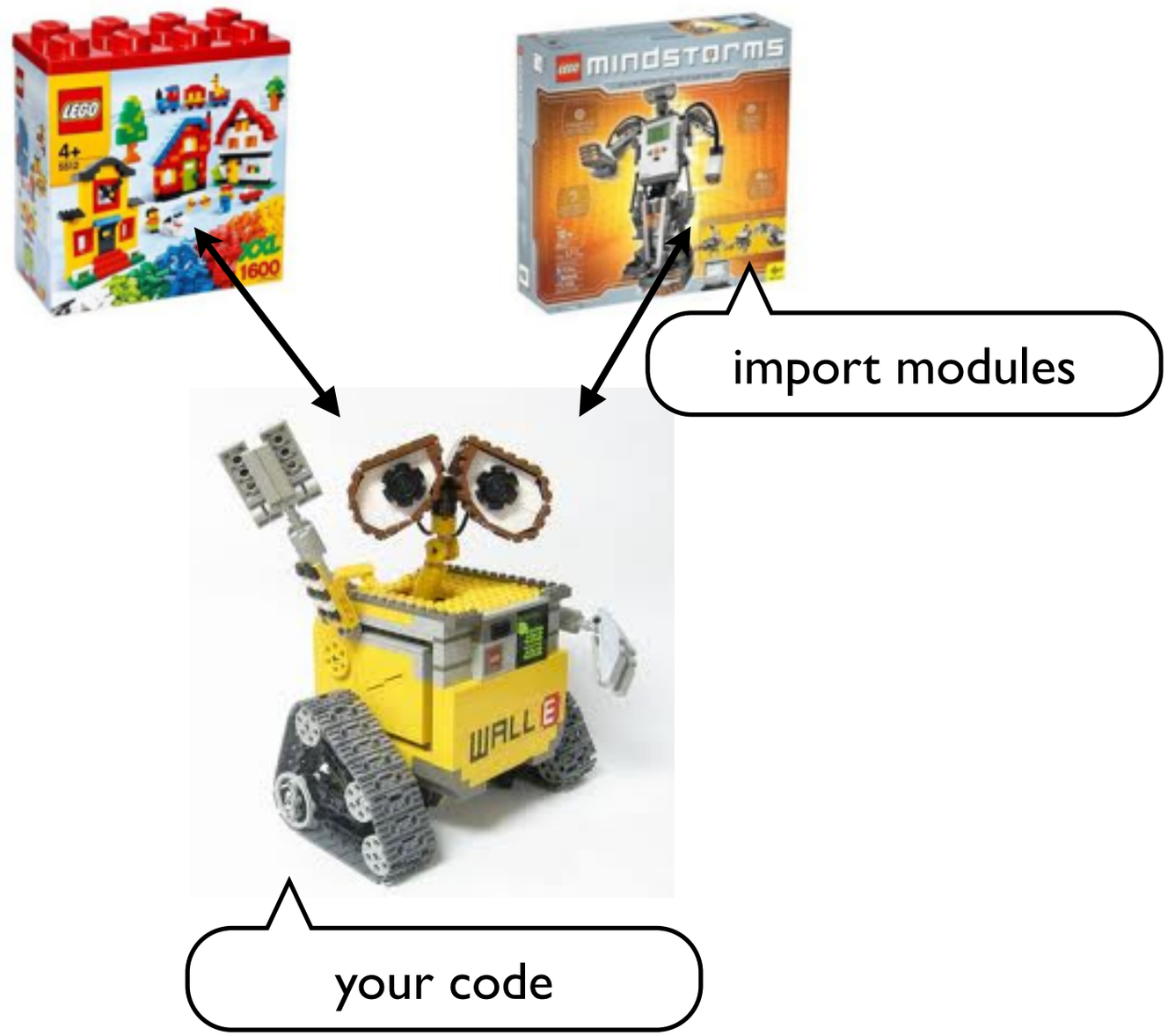
Vertica

HDF5

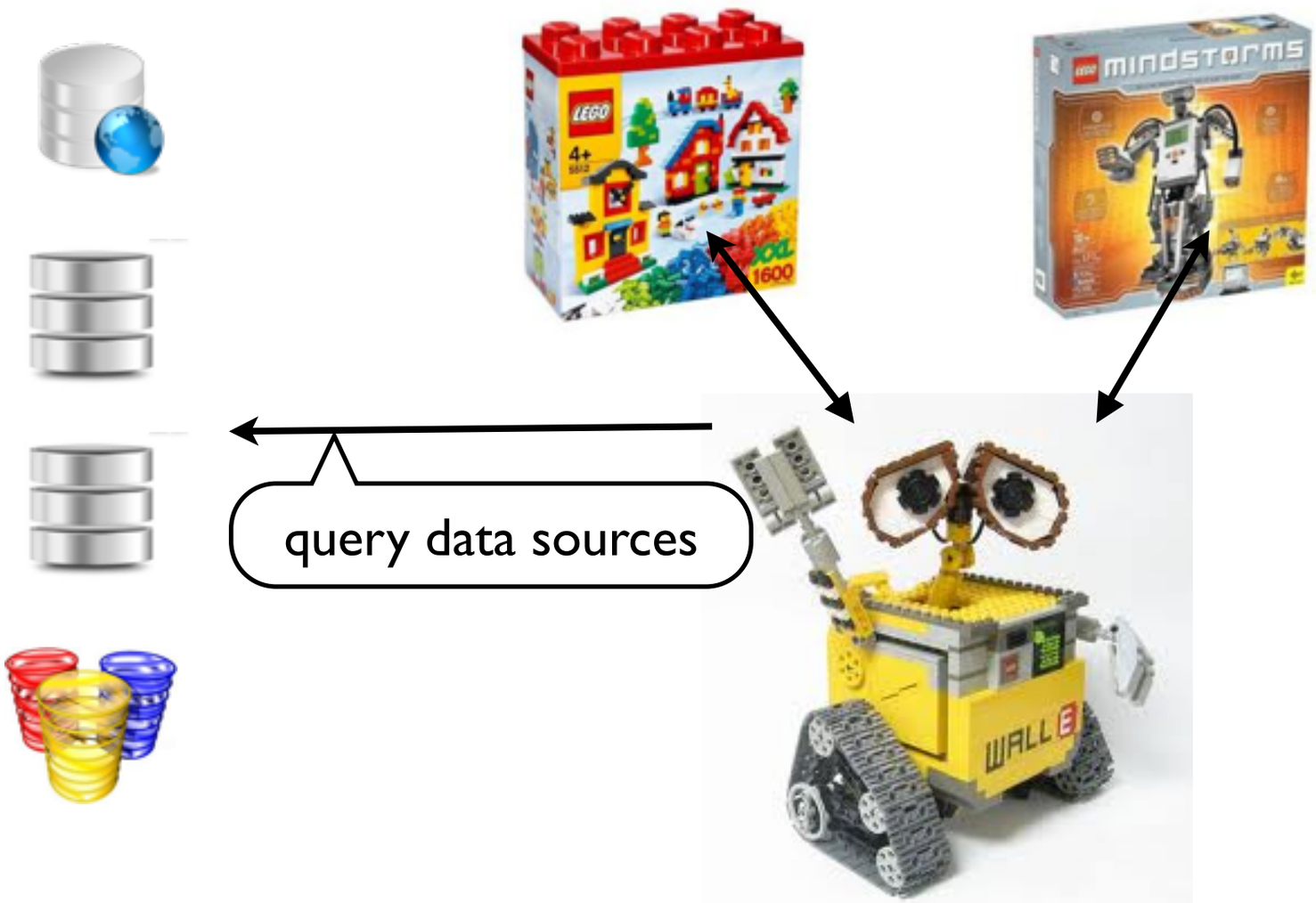
Spot Data Sources



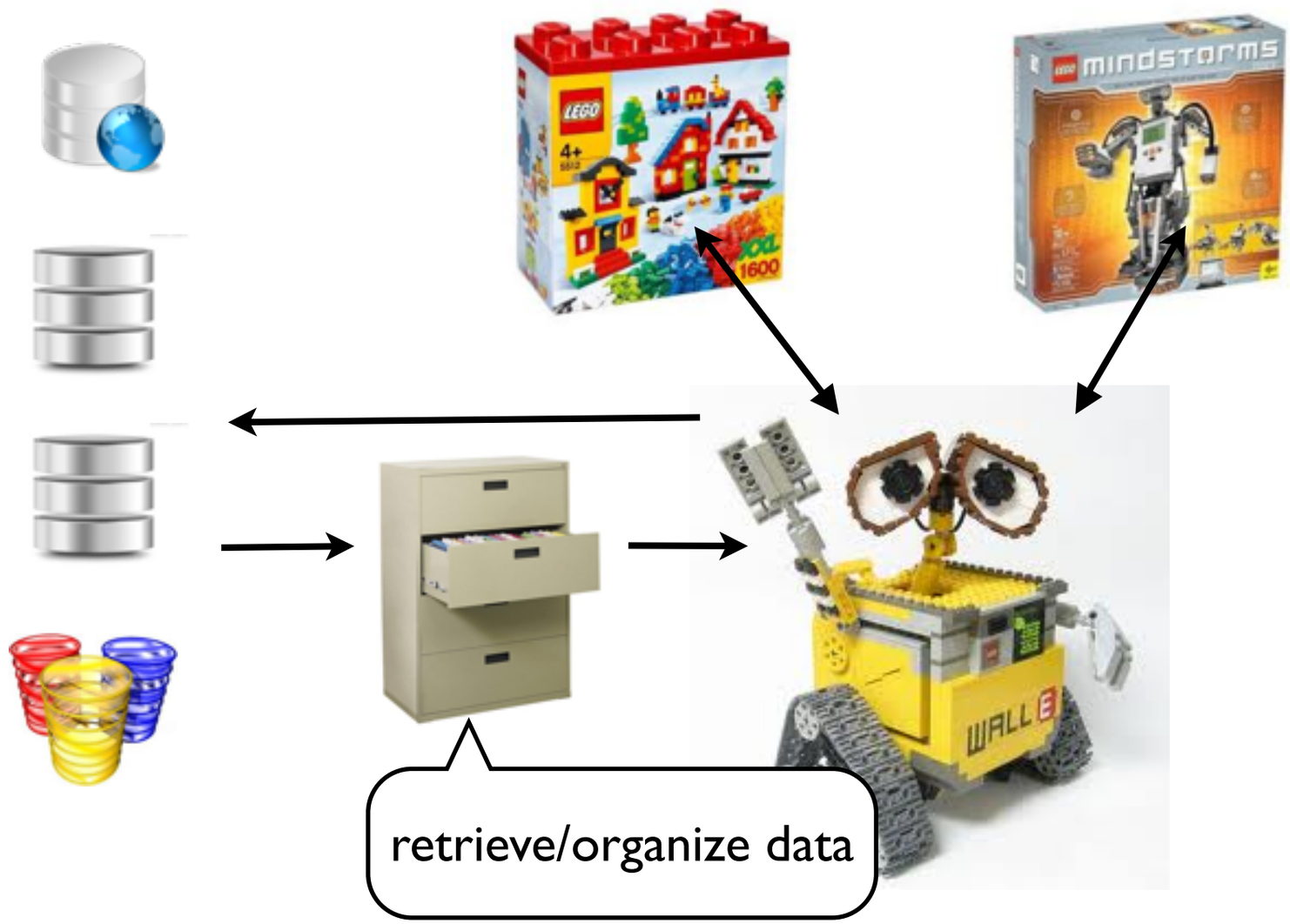
Workflow



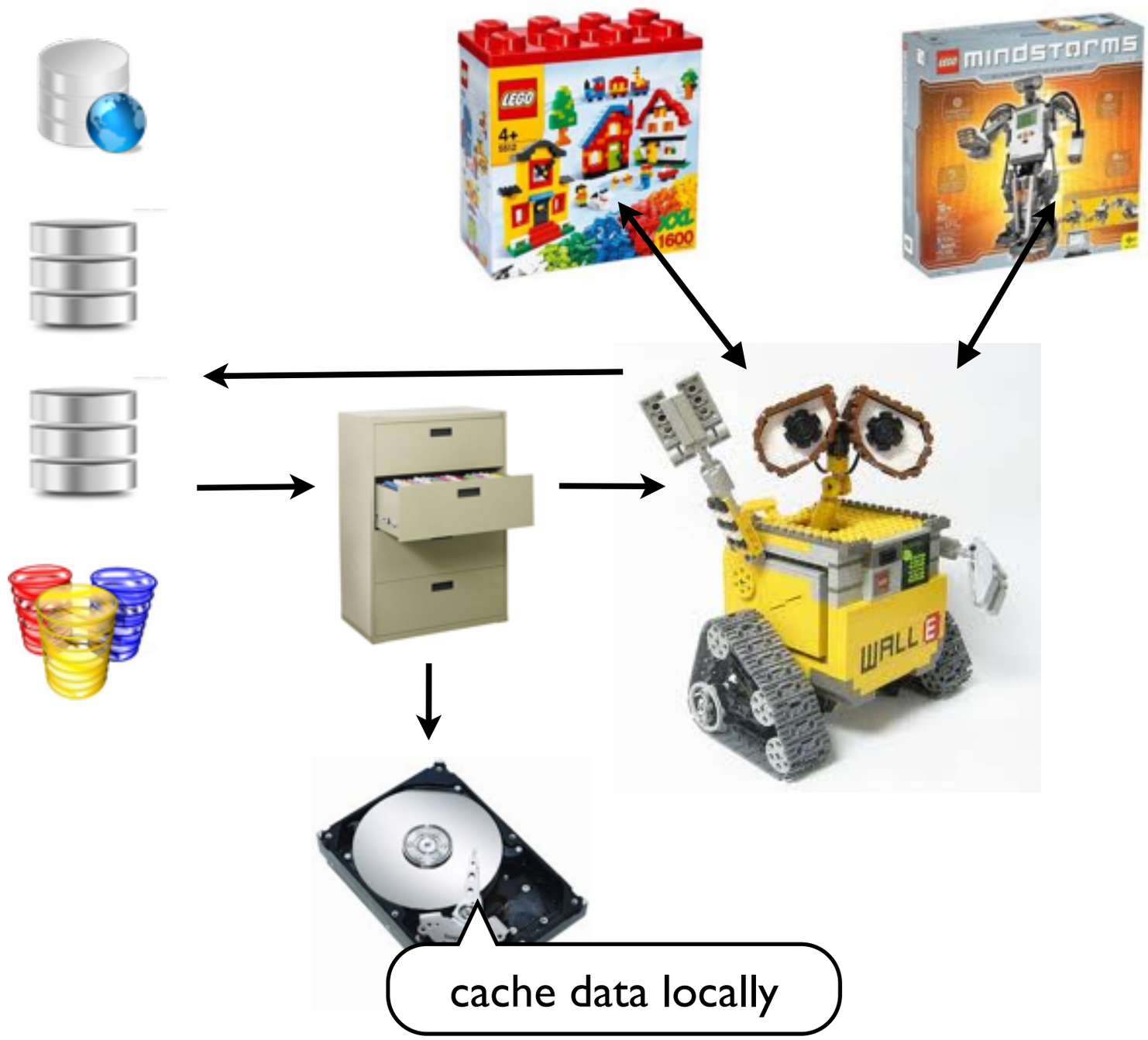
Workflow



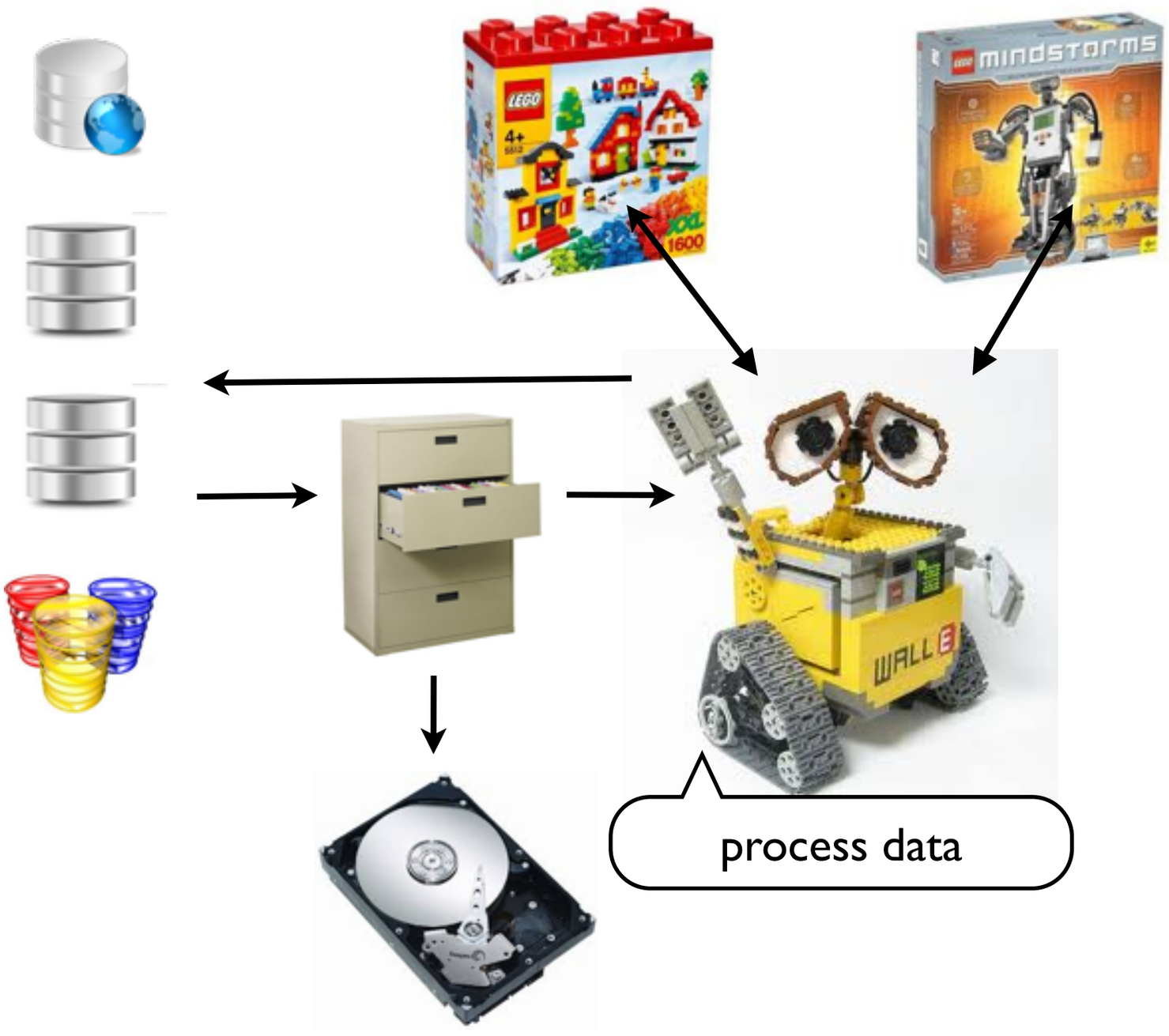
Workflow



Workflow



Workflow



Workflow

Programming

- Data

- Storage for Data

- Logic (Control Structures)

Basic Syntax

Keywords

```
from, import, if, elif, else, for, in, while, break,
continue, def, return, lambda, class, and, or, not,
open, input, ...
```

(lots of functions in modules)

Basic Python Types

```
>>> a = 1
>>> type(a)
<type 'int'>
```

```
>>> a = 1.2
>>> type(a)
<type 'float'>
```

```
>>> a = 'hello'
>>> type(a)
<type 'str'>
```

```
>>> a = [1,2,3]
>>> type(a)
<type 'list'>
```

```
>>> a = (1,2,3)
>>> type(a)
<type 'tuple'>
```

```
>>> a = {'one':1,'two':2}
>>> type(a)
<type 'dict'>
```

```
>>> a = set([1,2,3])
>>> type(a)
<type 'set'>
```

```
>>> import datetime
>>> a = datetime.datetime.now()
>>> type(a)
<type 'datetime.datetime'>
```

list

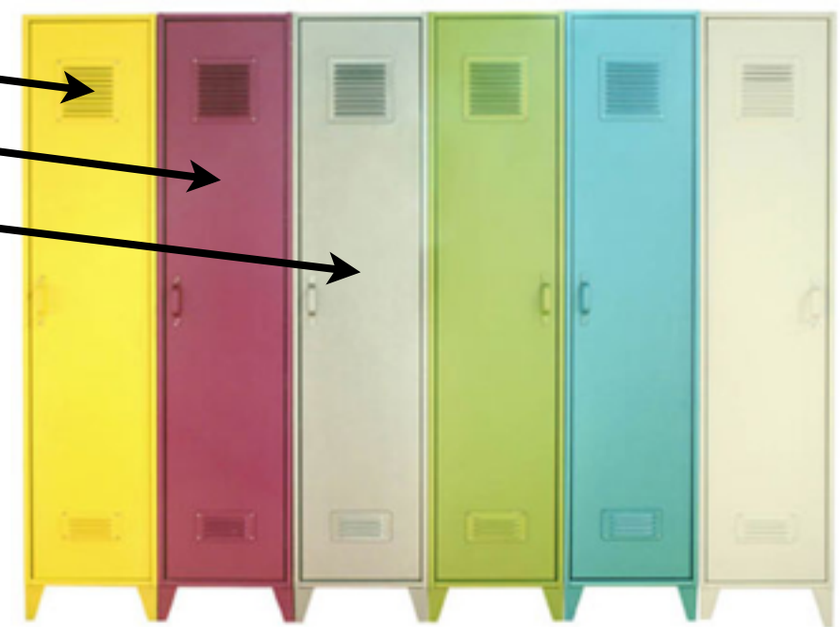
```
a = [3,4,5]        # [3,4,5]
a[0]               # 3
a[1]               # 4
a[2]               # 5
a.append(8)        # [3,4,5,8]
a.insert(0,9)      # [9,3,4,5,
del a[2]           # [9,3,5,8]
a.remove(5)        # [9,3,8]
3 in a             # True
len(a)             # 3
sum(a)             # 20
```

tuple

```
a = (3,4,5)        # (3,4,5)
a[0]               # 3
a[1]               # 4
a[2]               # 5
```

~~a.append(8) # (3,4,5,8)~~
~~a.insert(0,9) # (9,3,4,5,8)~~
~~del a[2] # (9,3,5,8)~~
~~a.remove(5) # (9,3,8)~~

```
3 in a             # True
len(a)             # 3
sum(a)             # 12
```

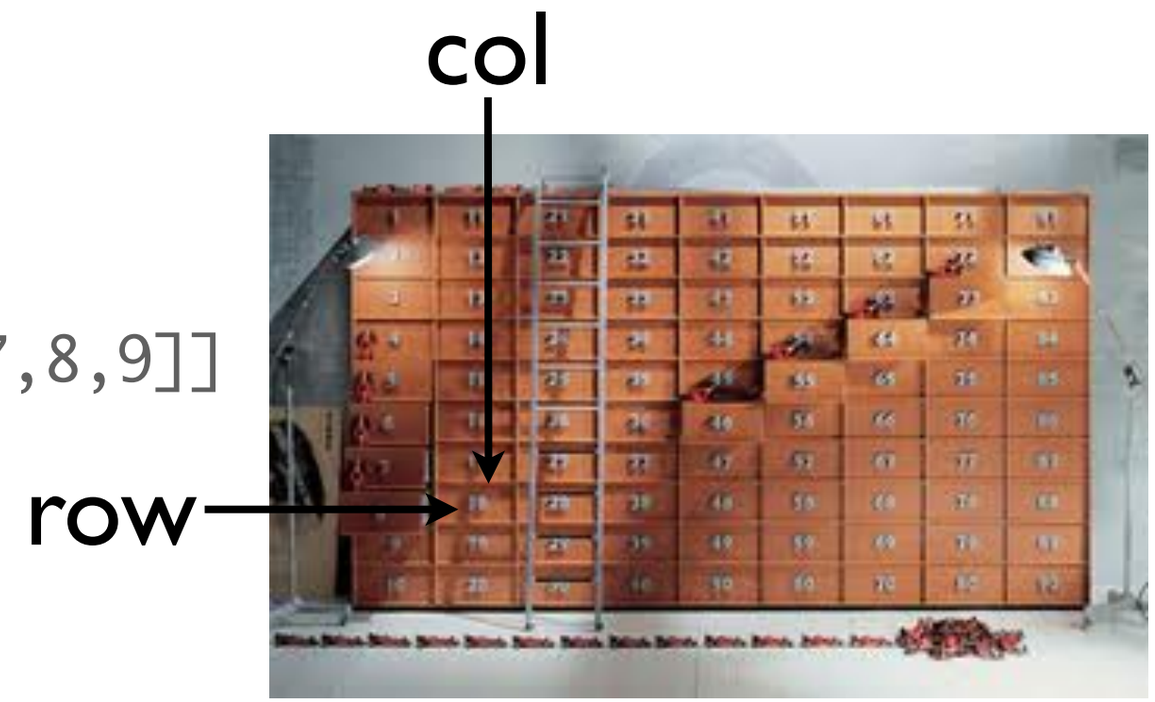


dict(ionary)

```
a = {'x':3,'y':4,'z':5}
a['x']
a['y']
a['z']
a.keys()          # ['x','y','z']
a.values()        # [3,4,5]
del a['y']        # {'x':3,'z':5}
a['t'] = 9        # {'x':3,'z':5,'t':9}
'x' in a          # True
len(a)            # 3
```

list of lists

```
a = [[1,2,3], [4,5,6], [7,8,9]]

a[row][col] = value
print a[row][col]
```

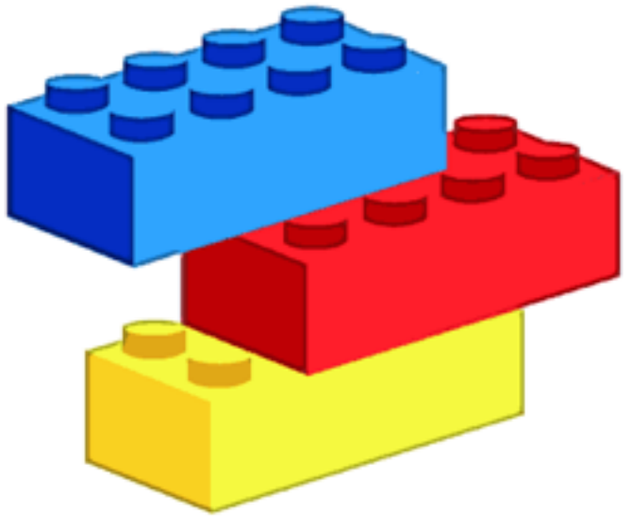


list of dict(ionaries)

```
a = [{'x':1,'y':2}, {'x':4,'y':5}]

a[row][key] = value
print a[row][key]
```

Control Structures

```
if ... elif ... else
for ... in ... break ... continue
while ... break ... continue
def ... return
from ... import ...
try ... except ...
class ...
```



Python Program Example

```
import random

def guess_a_number():
    x = random.randint(0,100)
    for k in range(3):
        print 'attempt number',k
        y = int(input('make a guess: '))
        if y == x:
            print 'You win'
            break
        elif y < x:
            print 'x is higher'
        else:
            print 'x is lower'
    if x!=y:
         print 'Sorry you Lost!'

guess_a_number()
```

Import modules

Functions

Start

if ... elif ... else ...

```
a = 1
b = 2

if a < b:
    print 'a less than b'
elif a > b:
    print 'a greater then b'
else:
    print 'I guess they are the same'
```

for ... in ... break ...

```
for k in range(100):
    if k == 0:
        continue
    if k == 10:
        break
    else:
        print k
```

def ... return

```
def compute_maximum(a,b):
    if a > b:
        return a
    else:
        return b

print compute_maximum(3,7)  # prints 7
```

while

```
a = 2

while a < 10:
    print a
    a = a + 1
```

same as

```
for a in range(2,10):
    print a
```

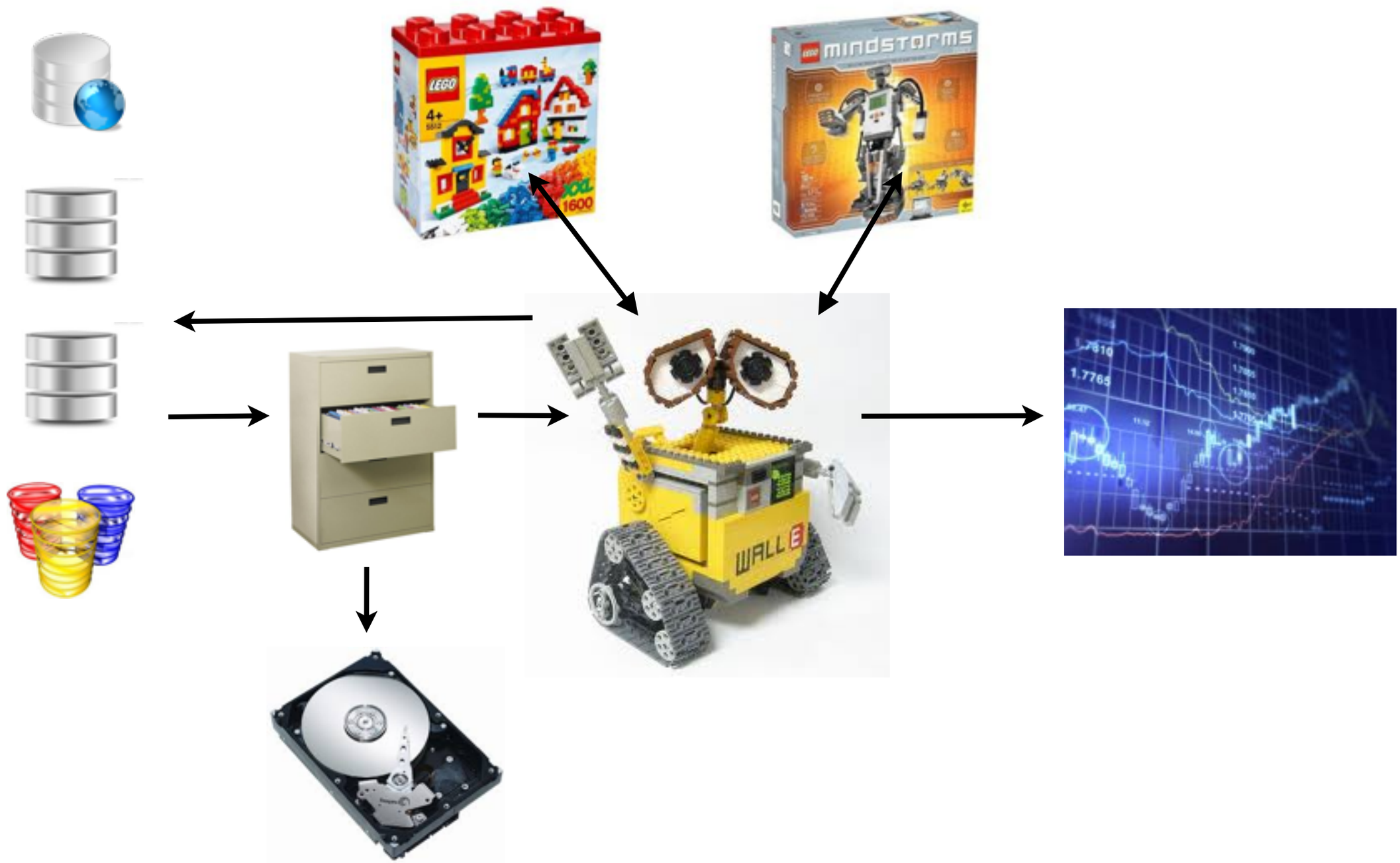
from ... import

```
import math
print math.sin(1)

from math import sin
print sin(1)

from math import *
print sin(1)
```

Speciality Modules

```
from nlib import *
c = YStock('goog').current()
h = YStock('goog').historical()
Canvas().plot(...).save('filename')
storage = PersistentDictionary('storage')

from dbspot import *
db = connect_mssql()
db = connect_vertica()
```

Workflow



Using nlib

YStock

```
>>> symbol = 'aapl'
>>> aapl = YStock(symbol)
>>> c = aapl.current()
>>> print c
{'stock_exchange': '"NasdaqNM"', 'market_cap': '408.3B', '52_week_high': 705.07,
'average_daily_volume': 17939600.0, 'price_sales_ratio': 2.46, 'price': 435.0,
'earnings_per_share': 41.896, 'price_earnings_ratio': 10.59, 'price_book_ratio': 3.08,
'volume': 6176501.0, 'price_earnings_growth_ratio': 0.56, '52_week_low': 385.1,
'short_ratio': 2.3, '200_days_moving_average': 505.657, '50_days_moving_average': 431.59,
'dividend_yield': 1.79, 'dividend_per_share': 7.95, 'ebitda': '57.381B', 'change': -8.86,
'book_value': 144.124}

>>> rows = aapl.historical()
>>> print rows[-1]
{'arithmetic_return': -0.02392575977481637, 'adjusted_vol': 15955700.0, 'adjusted_open':
453.85, 'log_return': -0.024216629657409748, 'adjusted_low': 442.15, 'volume': 15955700.0,
'adjusted_high': 455.2, 'high': 455.2, 'low': 442.15, 'date': datetime.datetime(2013, 5,
14, 0, 0), 'close': 443.86, 'open': 453.85, 'adjusted_close': 443.86}
```

YStock storage

```
>>> symbol = 'aapl'
>>> storage = PersistentDictionary('storage')
>>> if not symbol in storage:
>>>     storage[symbol] = YStock(symbol).historical()
>>> rows = storage[symbol]
```

YStock filter column

```
>>> symbol = 'aapl'
>>> storage = PersistentDictionary('storage')
>>> if not symbol in storage:
>>>     storage[symbol] = YStock(symbol).historical()
>>> rows = storage[symbol]

>>> lr = [row['log_return'] for row in rows if
          row['date'].year == 2012]

>>> print sum(lr)
>>> print len(lr)
>>> print sum(lr)/len(lr)
>>> Canvas().hist(lr).save('aapl_lr_2012.png')
```

list of float

YStock filter column

```
>>> symbol = 'aapl'
>>> storage = PersistentDictionary('storage')
>>> if not symbol in storage:
>>>     storage[symbol] = YStock(symbol).historical()
>>> rows = storage[symbol]

>>> price = [row['adjusted_close'] for row in rows if
            row['date'].year == 2012]

>>> Canvas().plot(price).save('aapl_price_2012.png')
```

list of float, or list of (x,y)

YStock filter column

```
from nlib import *

def geth(symbol, year=2012, column = 'log_return'):
    storage = PersistentDictionary('my_stocks')
    if not symbol in storage:
        storage[symbol] = YStock(symbol).historical()
    h = storage[symbol]
    a = [row[column] for row in h
         if row['date'].year == year]
    return a

>>> price = geth('aapl',2012,column='adjusted_close')
>>> Canvas().plot(price).save('aapl_price_2012.png')
```

Using dbspot

Connect to Vertica

Connect to VERTICA

```
>>> from dbspot import *
>>> db = connect_vertica()
```

List available Tables

```
>>> print db.tables
['stock_trade', 'market_making_quote', 'future_quote', 'spread_trade',
'option_quote', 'spot_point_of_trade', 'point_of_trade', 'future_trade',
'spread_quote', 'stock_quote', 'greeks_snapshot', 'order_state',
'option_trade']
```

Tables and Columns

Select a table

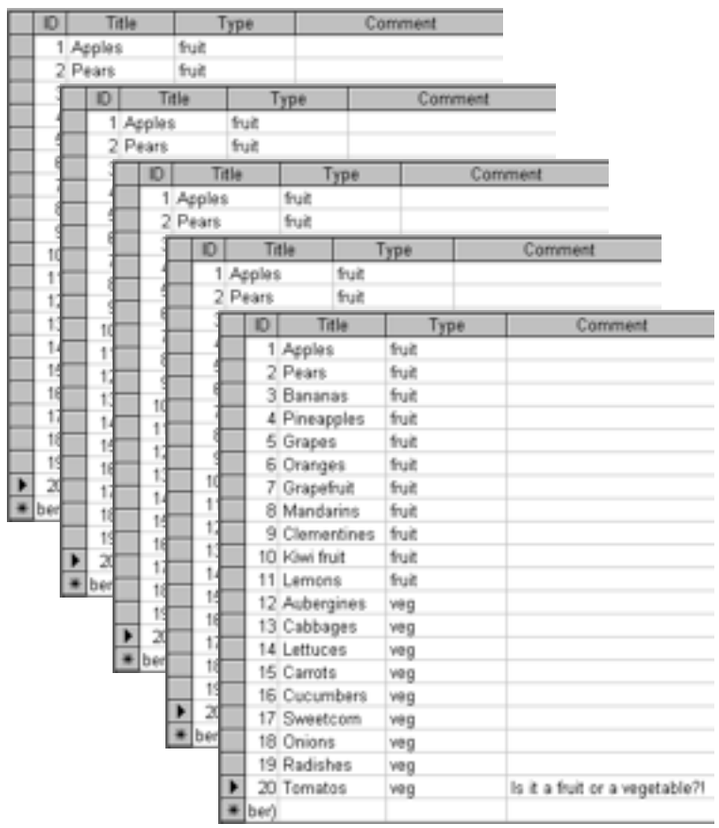
```
>>> Stock = db.stock_trade
```

List available Columns

```
>>> print Stock.fields
['instrument_id', 'symbol', 'timestamp', 'price', 'size', 'exchange',
'volume', 'open', 'close', 'high', 'low', 'trade_status',
'sequence_number', 'bid', 'bid_size', 'bid_exchange', 'ask', 'ask_size',
'ask_exchange', 'quote_sequence_number']
```

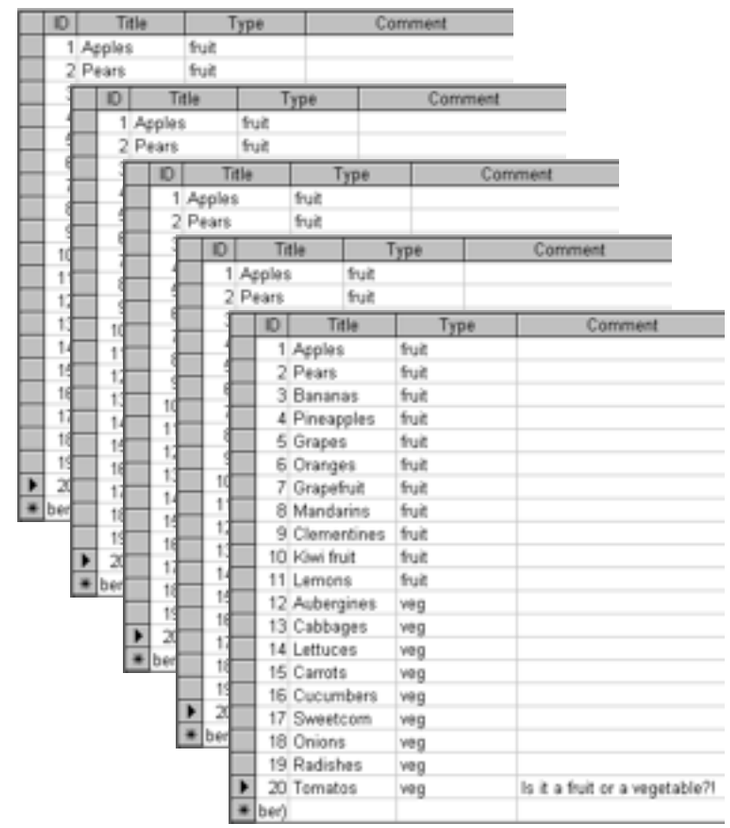
Spot Databases

MSSQL



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!

Vertica



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!

Table Names

Database vertica

- stock_trade
- market_making_quote
- future_quote
- spread_trade
- option_quote
- spot_point_of_trade
- point_of_trade
- future_trade
- spread_quote
- stock_quote
- greeks_snapshot
- order_state
- option_trade

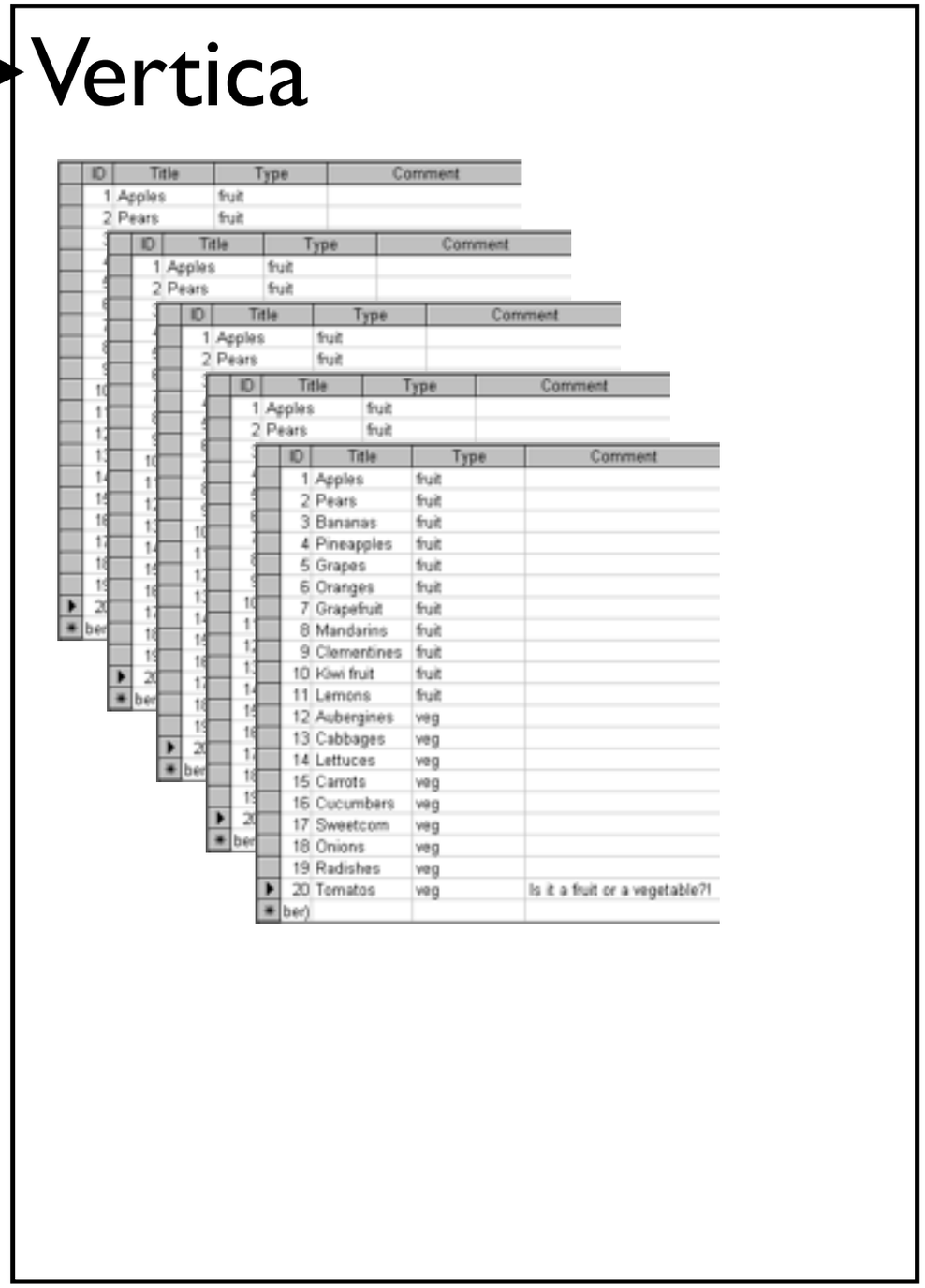
Database mysql

- FutureOptionContractVIEW
- InstrumentSectorFE
- StockVIEW
- OptionContractDetailsVIEW
- FutureContractVIEW

Overview

```
db = connect_vertica()
```

the connection



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!

Overview

```
db = connect_vertica()
Stock = db.stock_trade
```

the table
"stock_trade"

Vertica

ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
```

a column

Vertica

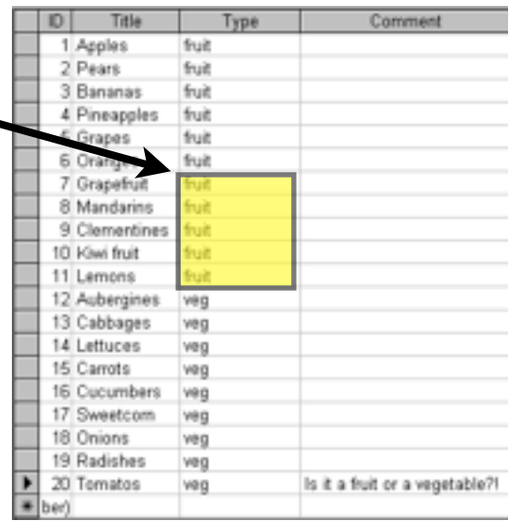
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'
```

a query, it identifies those record with column "symbol" equal to "AAPL"

Vertica



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'
```

```
rows=db(q).select(Stock.ALL)
```

gets the records matching the query condition

list of records matching the query condition

Vertica

ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'

rows=db(q).select(Stock.ALL)
rows[0]
```

first record matching the query condition

Vertica

ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'

rows=db(q).select(Stock.ALL)
rows[1]
```

second record matching the query condition

Vertica

ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'

rows=db(q).select(Stock.ALL)
rows[2]
```

third record matching the query condition

Vertica

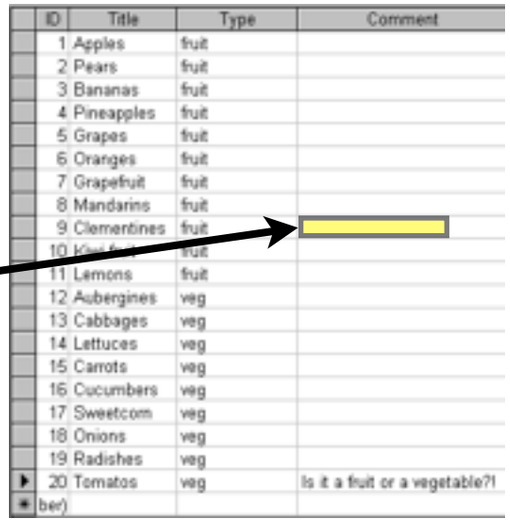
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'

rows=db(q).select(Stock.ALL)
rows[2]
print rows[2].timestamp
```

"timestamp" value of third record matching the query condition

Vertica



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q = Stock.symbol=='AAPL'

rows=db(q).select(Stock.ALL)
rows[2]
print rows[2].price
```

"price" value of third record matching the query condition

Vertica

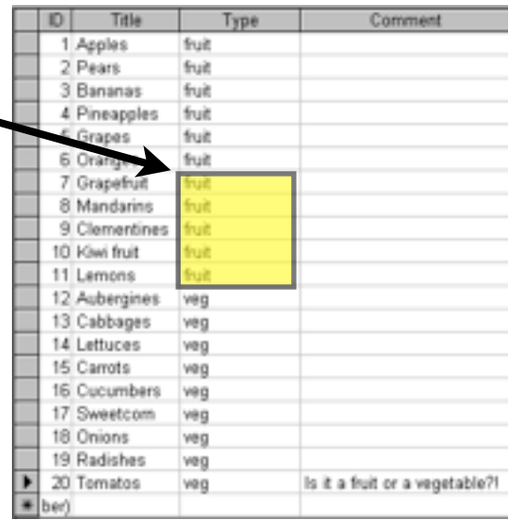
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
```

filter by symbol=='AAPL'

Vertica



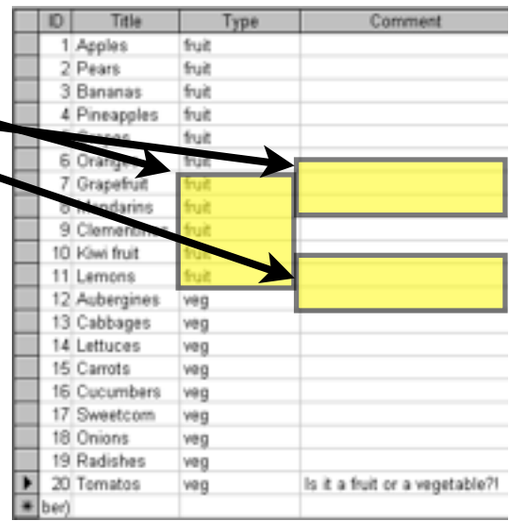
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
```

filter by "yesterday"

Vertica



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
[illegible]	[illegible]	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
q3=q1&q2
```

filter by both (&)

Vertica

ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
q3=q1&q2

rows=db(q3).select(Stock.ALL)
```

get records

Vertica

ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
q3=q1&q2

rows=db(q3).select(Stock.ALL)
rows[0]
```

get first AAPL trade yesterday

Vertica

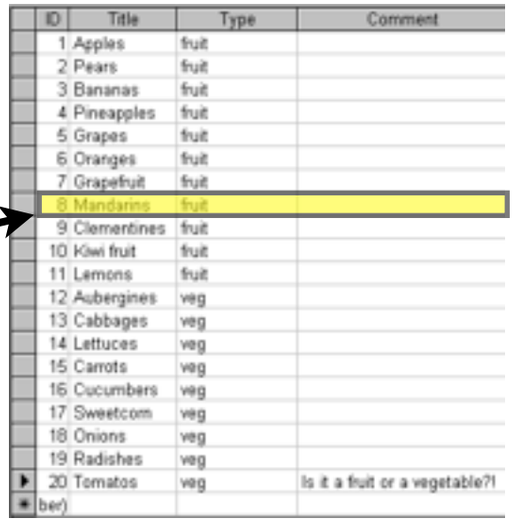
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
q3=q1&q2

rows=db(q3).select(Stock.ALL)
rows[1]
```

get second AAPL trade yesterday

Vertica



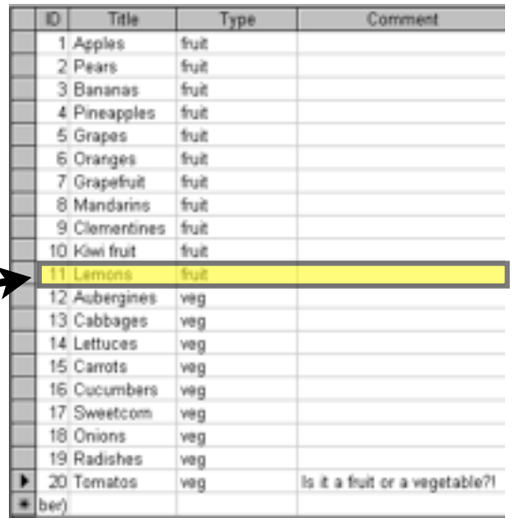
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
q3=q1&q2

rows=db(q3).select(Stock.ALL)
rows[2]
```

get third AAPL trade yesterday

Vertica



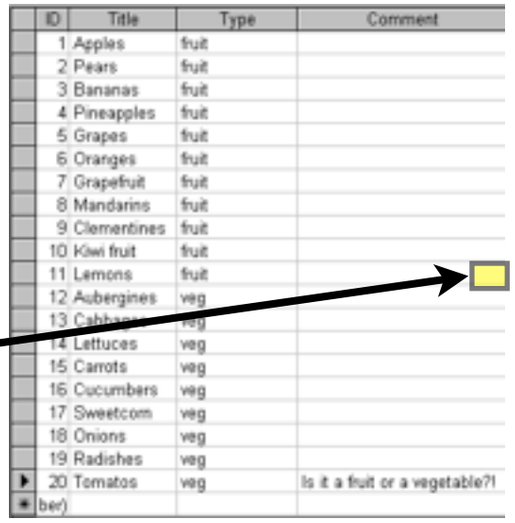
ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Overview

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1=Stock.symbol=='AAPL'
q2=daypast(Stock.timestamp,1)
q3=q1&q2

rows=db(q3).select(Stock.ALL)
rows[2].volume
```

get "volume" of third AAPL trade yesterday

Vertica



ID	Title	Type	Comment
1	Apples	fruit	
2	Pears	fruit	
3	Bananas	fruit	
4	Pineapples	fruit	
5	Grapes	fruit	
6	Oranges	fruit	
7	Grapefruit	fruit	
8	Mandarins	fruit	
9	Clementines	fruit	
10	Kiwi fruit	fruit	
11	Lemons	fruit	
12	Aubergines	veg	
13	Cabbages	veg	
14	Lettuces	veg	
15	Carrots	veg	
16	Cucumbers	veg	
17	Sweetcorn	veg	
18	Onions	veg	
19	Radishes	veg	
20	Tomatos	veg	Is it a fruit or a vegetable?!
ber)			

Filter by day

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL)
```

days in the past (1 = yesterday)

get all columns for table Stock

Filter and Order

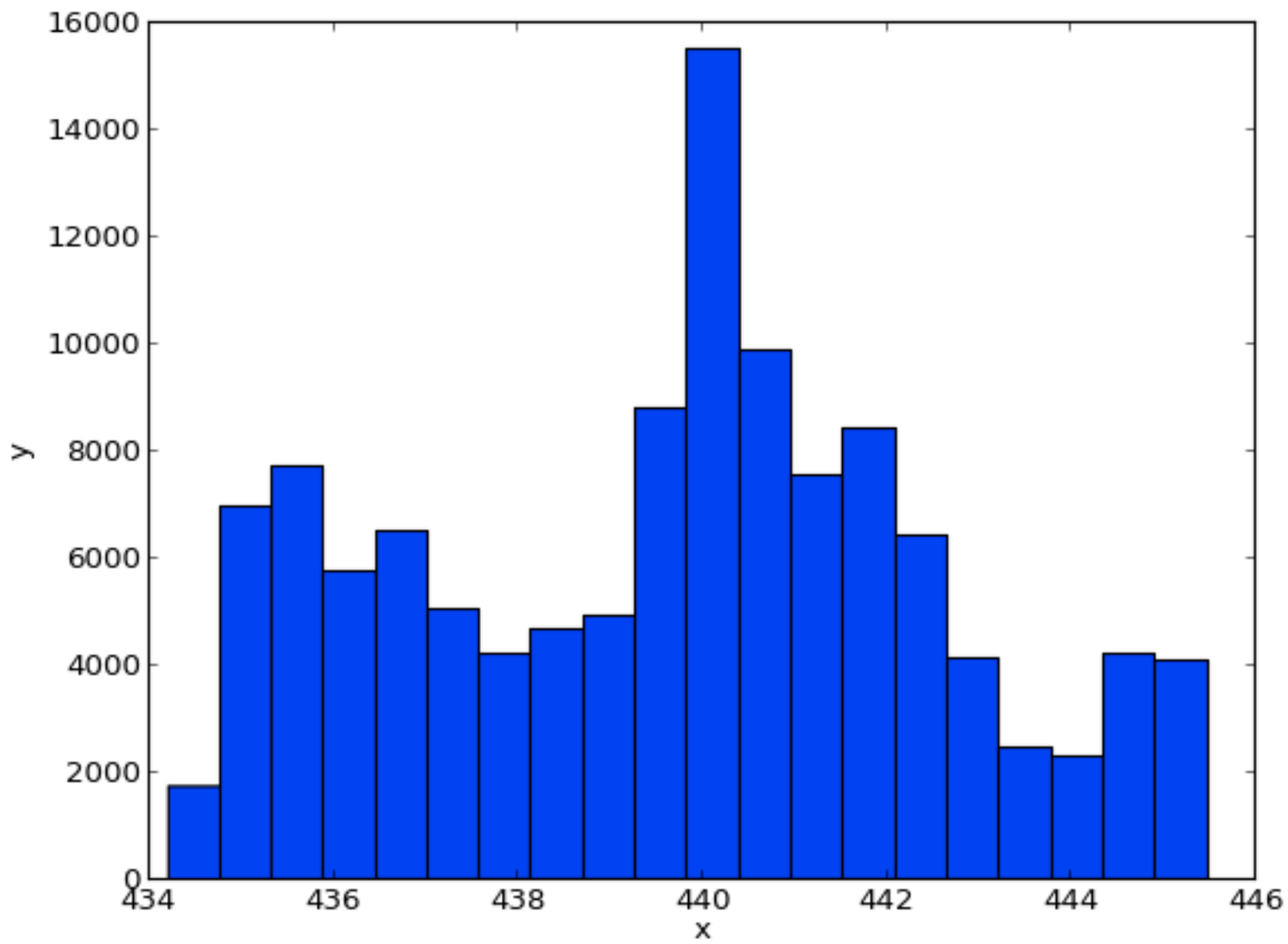
```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL, orderby=Stock.timestamp)
```

Order records by "timestamp" column

Hist Price

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL, orderby=Stock.timestamp)
data = [row.price for row in rows]
Canvas().hist(data).save('aapl.yesterday.trades.png')
```

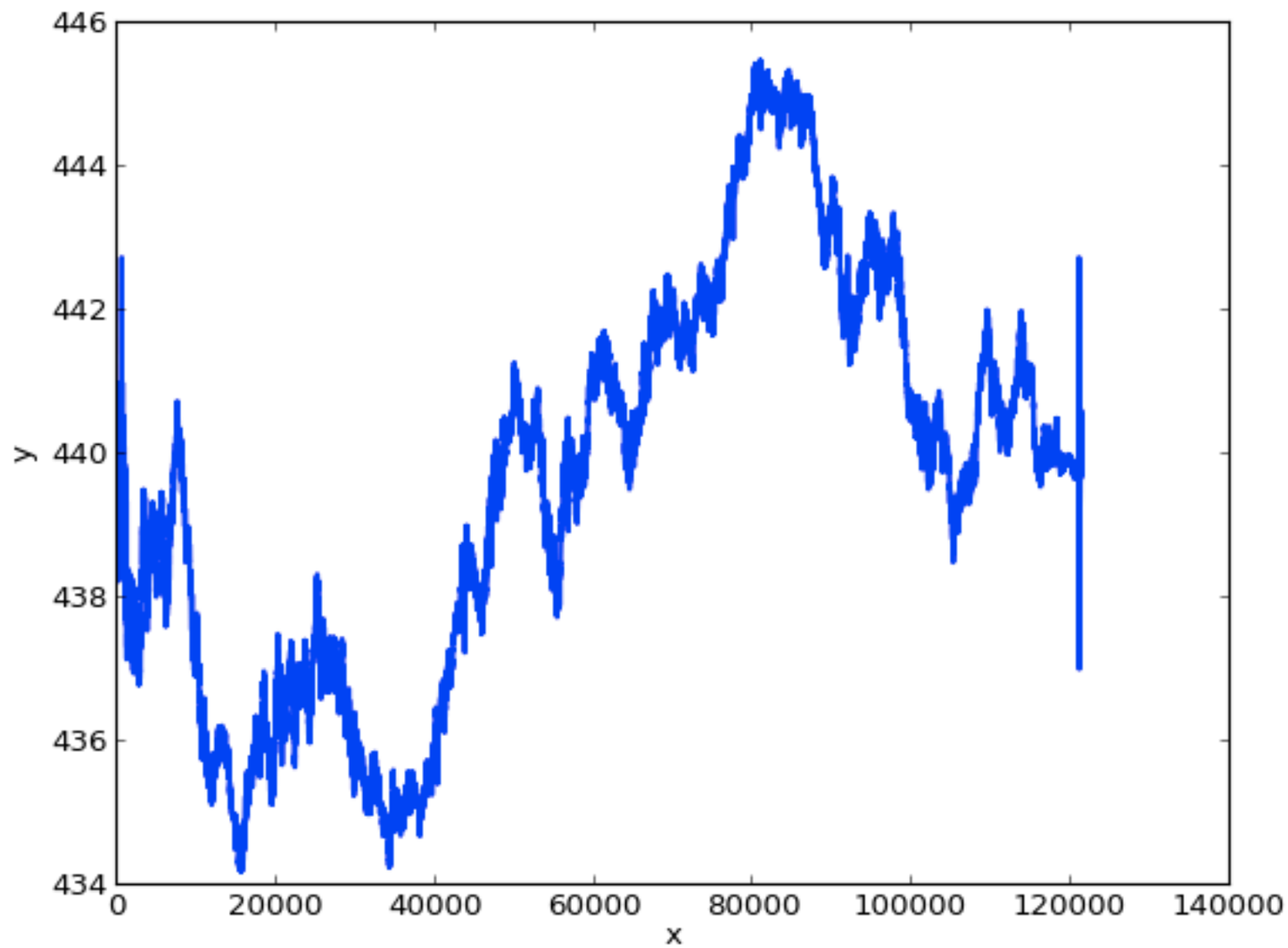
Histogram of Price



Plotting Price

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL, orderby=Stock.timestamp)
data = [row.price for row in rows]
Canvas().plot(data).save('aapl.yesterday.trades.hist.png')
```

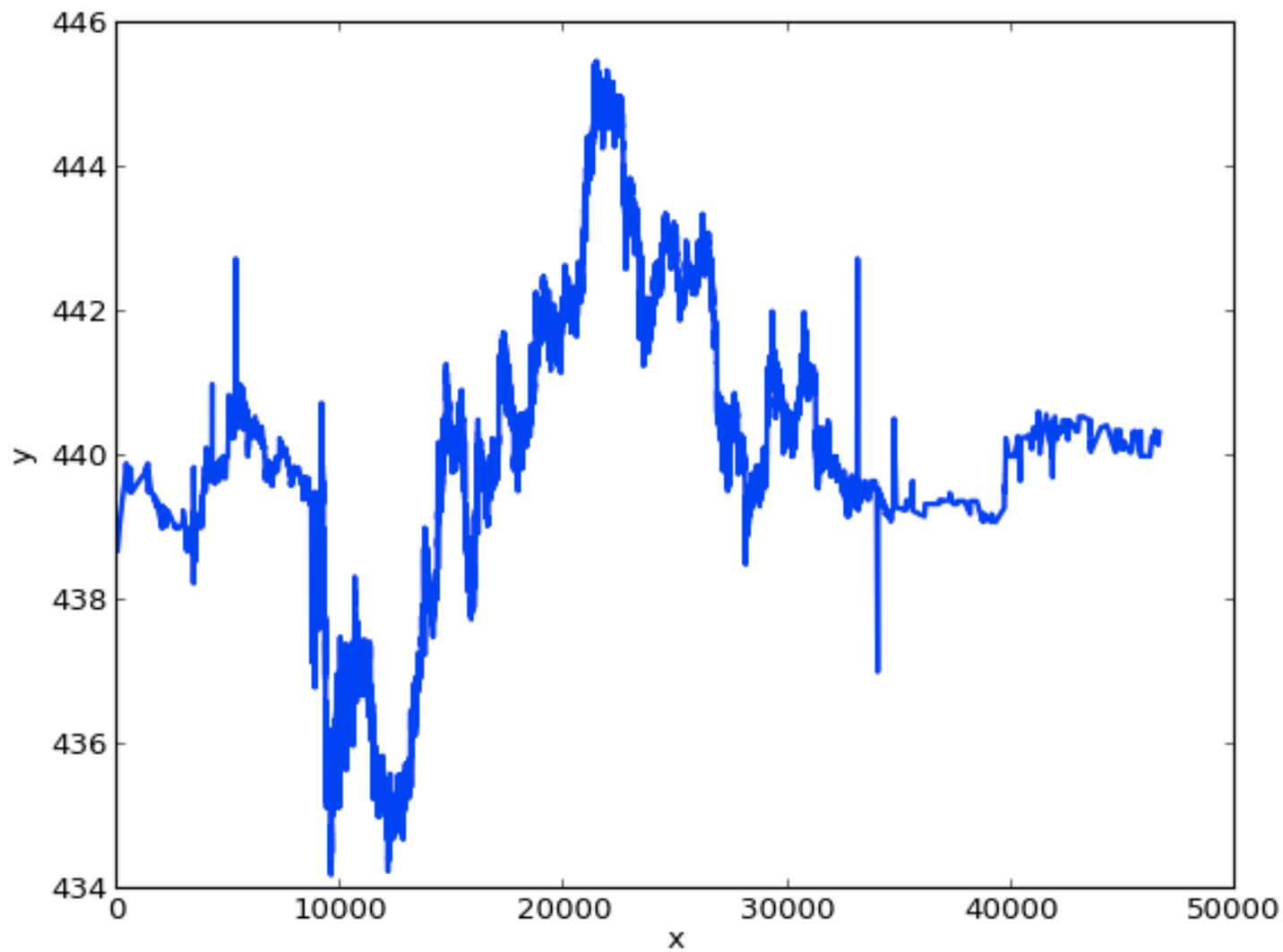
Plotting Price



Plotting Price (better)

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL, orderby=Stock.timestamp)
data = [((row.timestamp-rows[0].timestamp).seconds,
         row.price) for row in rows]
Canvas().plot(data).save('aapl.yesterday.trades.png')
```

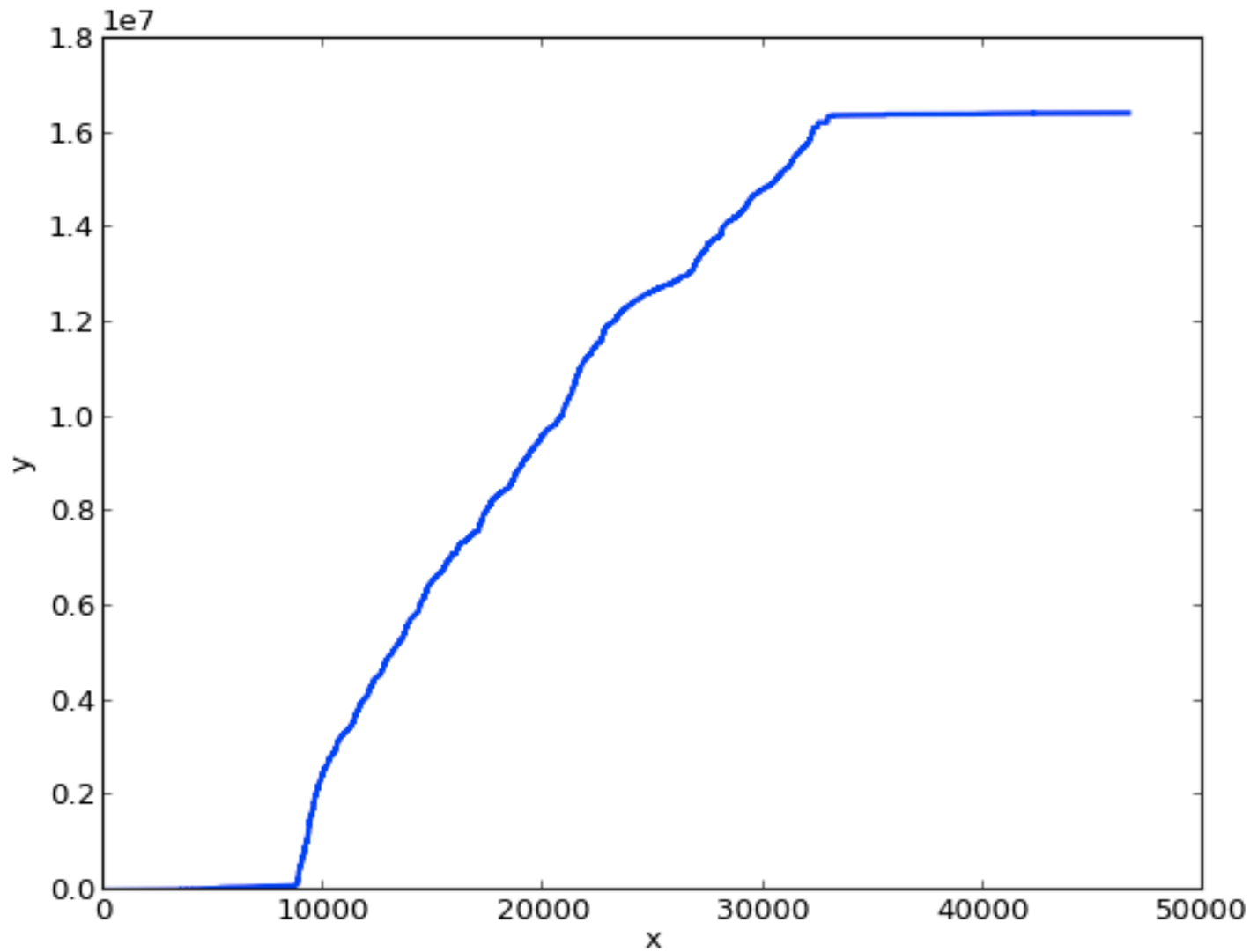
Plotting Price (better)



Plotting Volume

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL, orderby=Stock.timestamp)
data = [((row.timestamp-rows[0].timestamp).seconds,
         row.volume) for row in rows]
Canvas().plot(data).save('aapl.yesterday.trade_vol.png')
```

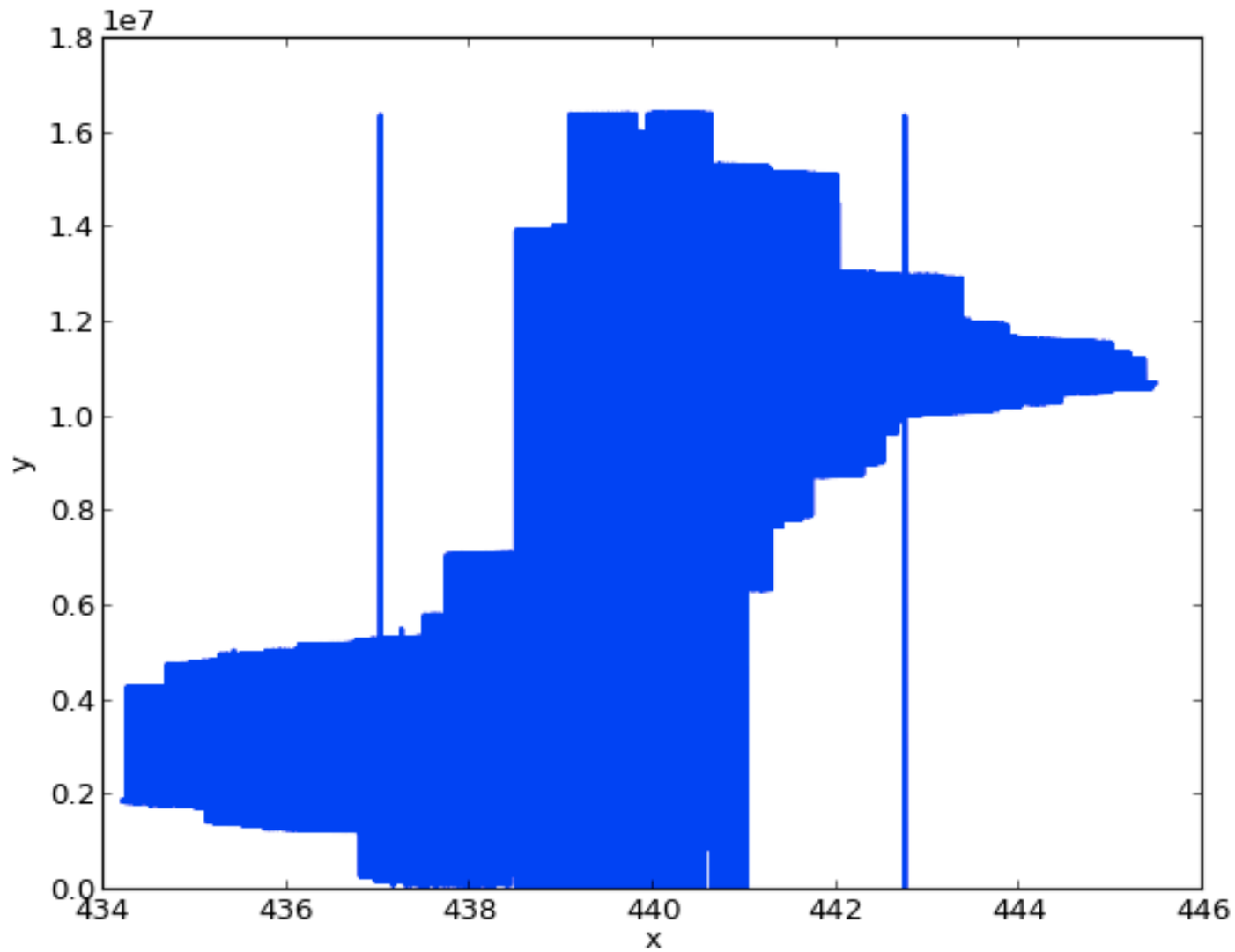
Plotting Volume



Price vs Volume

```
db = connect_vertica()
Stock = db.stock_trade
Stock.symbol
q1 = Stock.symbol=='AAPL'
q2 = daypast(Stock.timestamp,1)
q3 = q1&q2

rows = db(q3).select(Stock.ALL, orderby=Stock.timestamp)
data = [(row.price, row.volume) for row in rows]
data.sort()
Canvas().plot(data).save('aapl.yesterday.p_vs_vol.png')
```

Price vs Volume



To be continued....