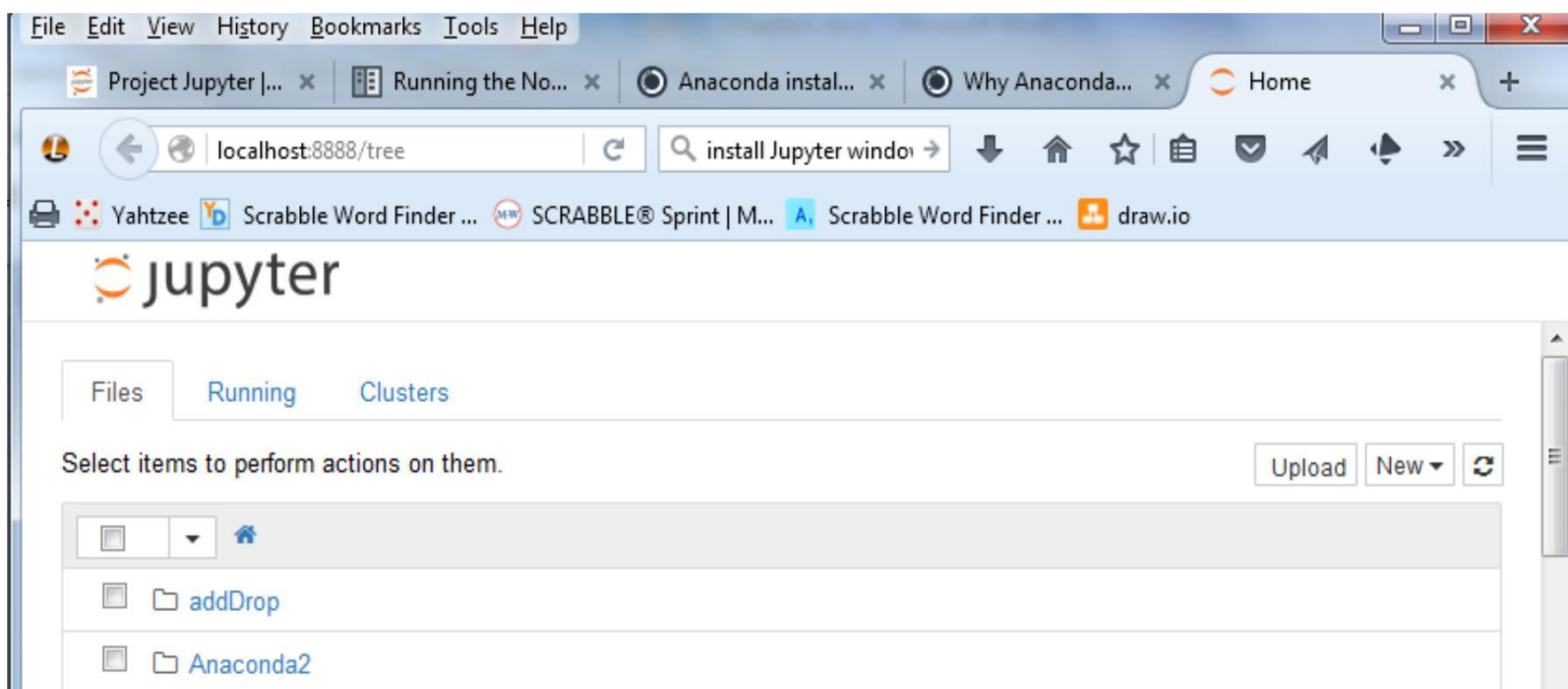


Chapter 1: Introduction to Jupyter



jupyter

Files Running Clusters



Files Running Clusters

Currently running Jupyter processes

Terminals ▾

Terminals are unavailable.

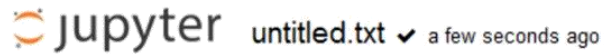
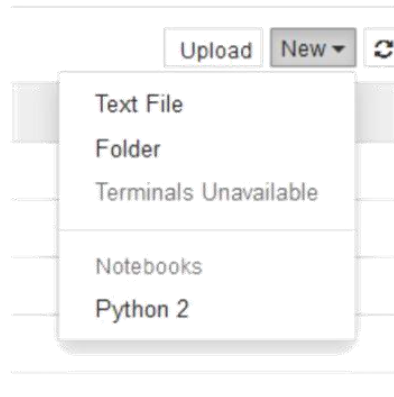
Notebooks ▾

There are no notebooks running.



Files Running Clusters

Clusters tab is now provided by IPython parallel. See [IPython parallel](#) for installation details.



File Edit View Language

1

Untitled Folder

jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Help

Code CellToolbar

In []:

Untitled.ipynb Running

- Folders
- All Notebooks
- Running
- Files
- configuration

Files Running Clusters

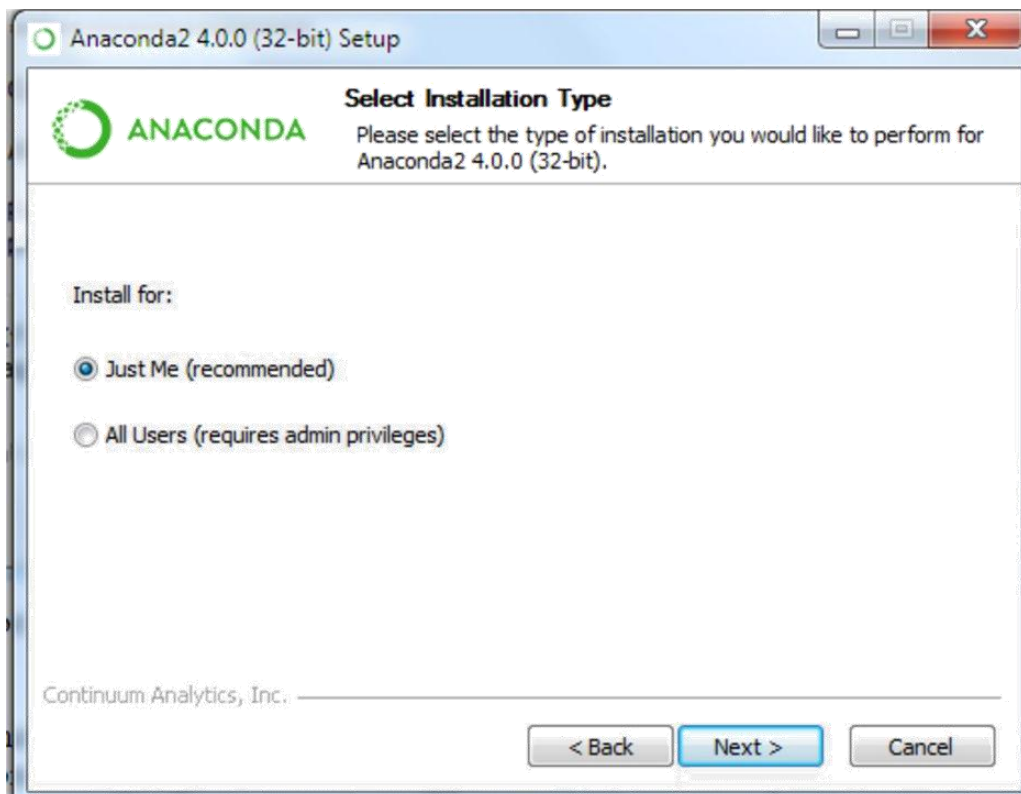
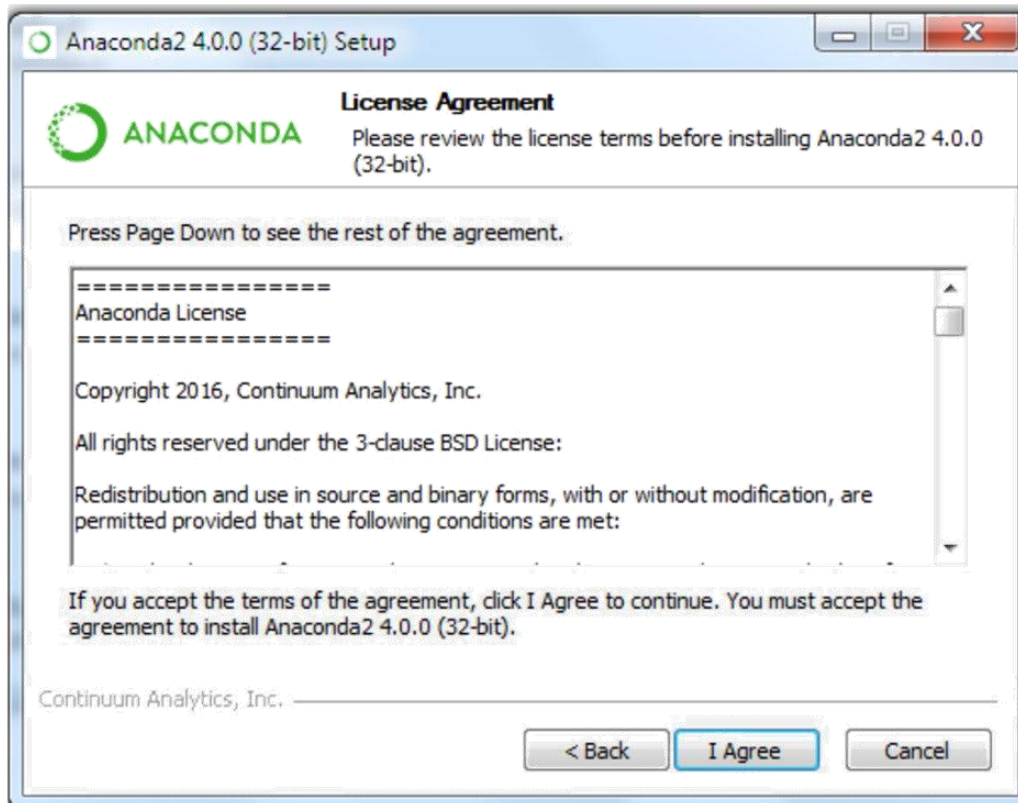
Duplicate Shutdown

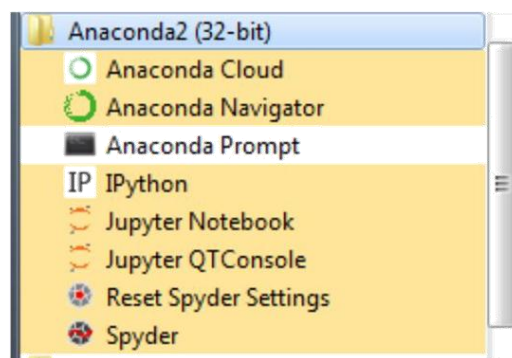
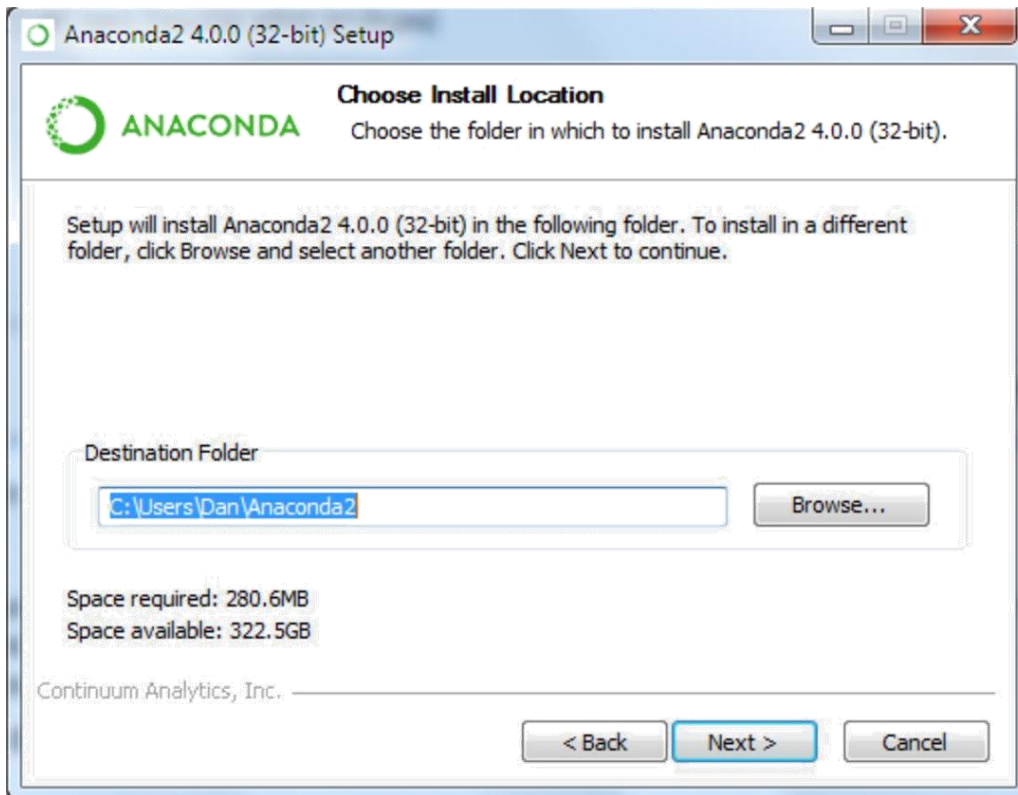
1

Untitled.ipynb Running


- addDrop
- Anaconda2









Files Running Clusters

Duplicate Rename 

1 

 anaconda

Duplicate ×

Are you sure you want to duplicate: addDrop.properties?

Duplicate

Cancel

-  addDrop-Copy1.properties
-  addDrop-test.properties
-  addDrop.properties

Rename file ×

Enter a new file name:

addDrop.properties

OK

Cancel

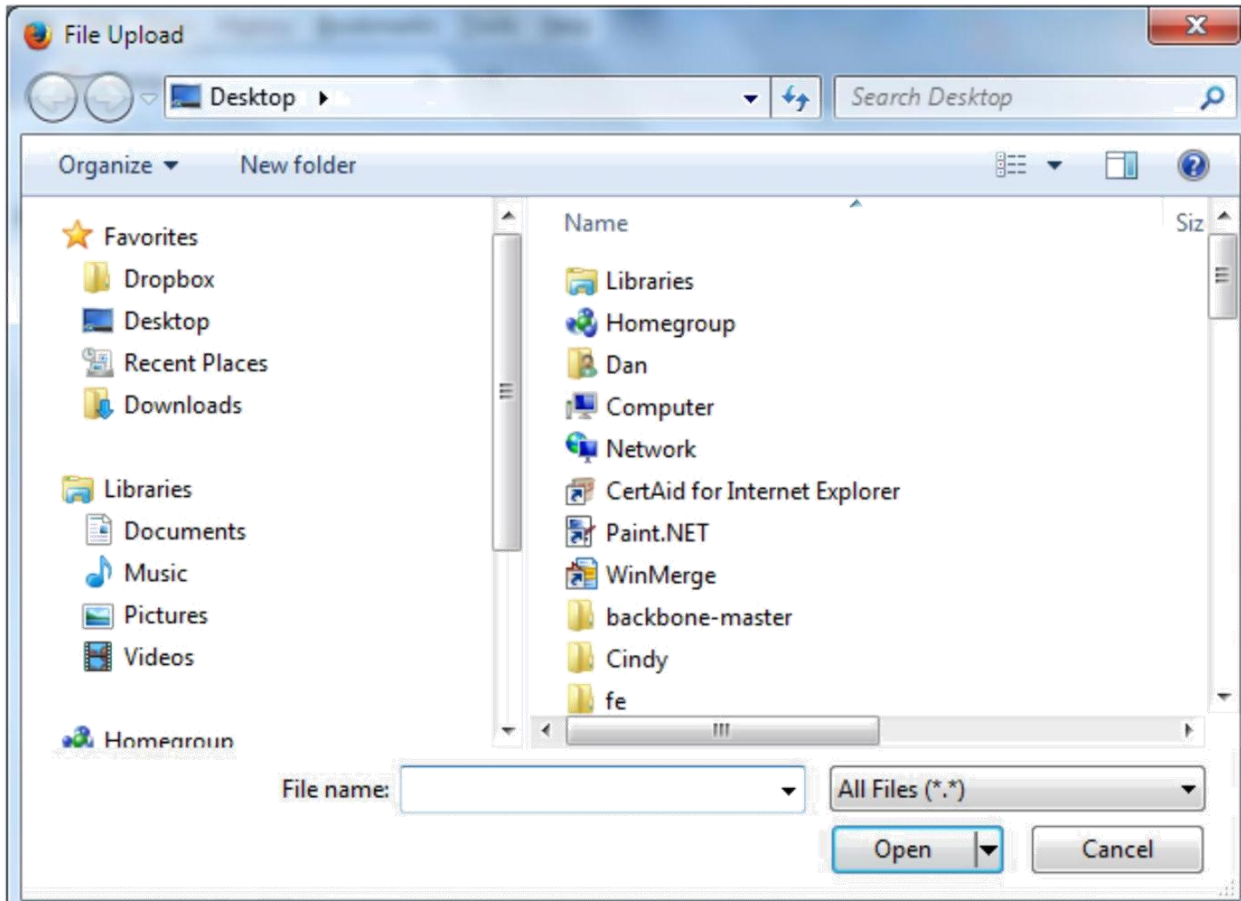
Delete

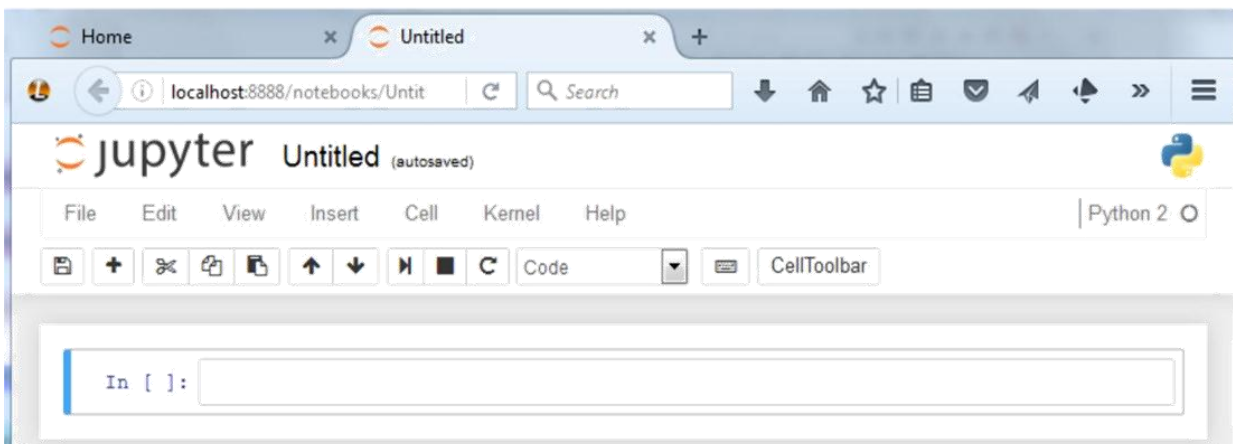
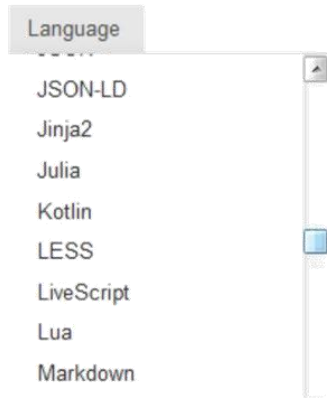
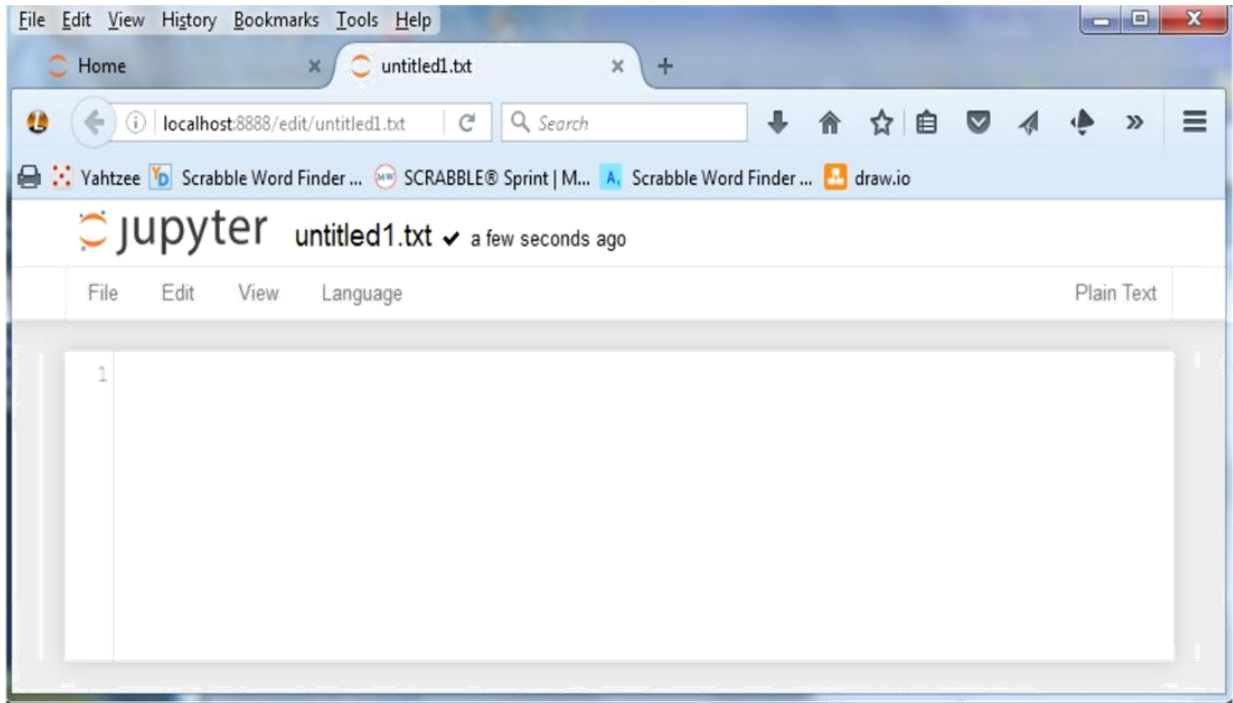


Are you sure you want to permanently delete: addDrop.properties?

Delete

Cancel





Edit Notebook Metadata



Manually edit the JSON below to manipulate the metadata for this Notebook. We recommend putting custom metadata attributes in an appropriately named sub-structure, so they don't conflict with those of others.

```
1 {  
2   "kernel_spec": {  
3     "name": "python2",  
4     "display_name": "Python 2",  
5     "language": "python"  
6   },  
7   "language_info": {  
8     "mimetype": "text/x-python",  
9     "nbconvert_exporter": "python",  
10    "name": "python",  
11    "pygments_lexer": "ipython2",  
12    "version": "2.7.11",  
13    "file_extension": ".py",  
14    "codemirror_mode": {  
15      "version": 2,  
16      "name": "ipython"  
17    }  
18  }  
19 }
```

OK

Cancel

Find and Replace



Aa

.*



Find

Replace

Replace All

File Edit **View** Insert Cell Kernel Help

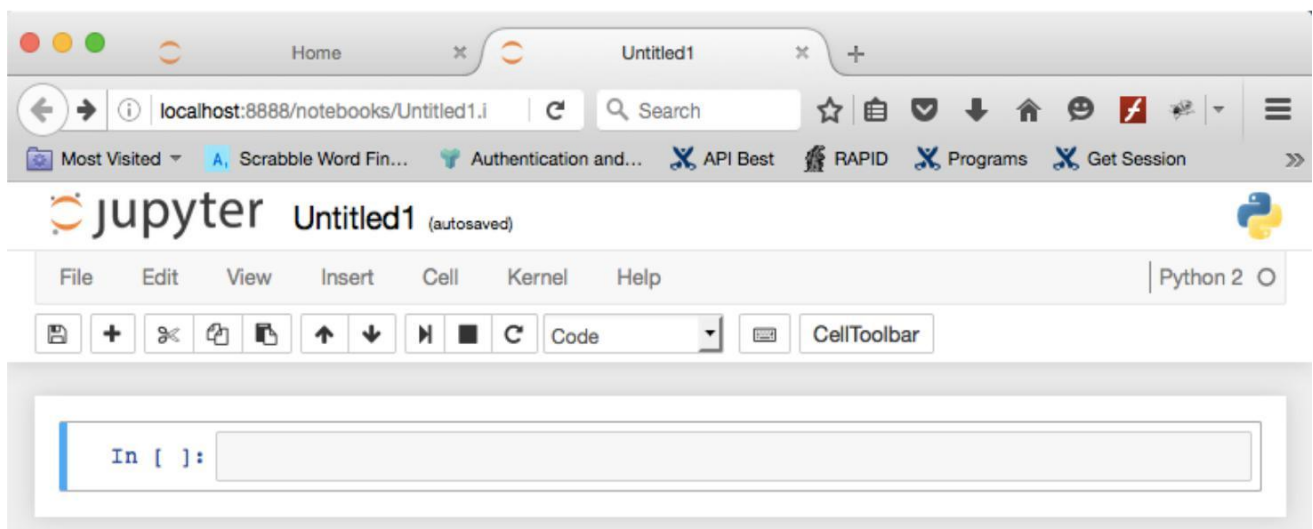
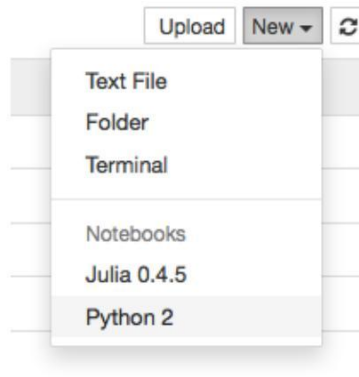
CellToolbar

- Toggle Header
- Toggle Toolbar
- Cell Toolbar**

- None
- Edit Metadata
- Raw Cell Format
- Slideshow

In []:

Chapter 2: Jupyter Python Scripting



The screenshot shows a web browser window with the URL `localhost:8888/notebooks/Learning Jupyter Chapter 2`. The page title is "Learning Jupyter Chapter 2 (unsaved changes)". The Jupyter interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The code cells contain the following Python code:

```
In [ ]: name = "Dan"
        age = 37

In [ ]: print(name + ' is ' + str(age) + ' years old.')
```

The screenshot shows a file manager window with the user name "dtoomey". The "Favorites" sidebar shows "iCloud Drive". The main pane displays a table of files:

Name	Date Modified
Learning Jupyter Chapter 2.ipynb	Today, 3:46 PM

The screenshot shows the Jupyter file browser interface with the URL `localhost:8888/tree#`. The page title is "jupyter". The file list shows the following items:

<input type="checkbox"/>	webex-cps-common	
<input type="checkbox"/>	Workspaces	
<input type="checkbox"/>	Learning Jupyter Chapter 2.ipynb	Running
<input type="checkbox"/>	Untitled.ipynb	

Learning Jupyter Chapter 2 (autosaved)

File Edit View Insert Cell Kernel Help Python 2

```
In [1]: name = "Dan"
        age = 37
```

```
In [2]: print(name + ' is ' + str(age) + ' years old.')
        Dan is 37 years old.
```

```
Last login: Tue Apr 26 15:28:27 on ttys001
bos-mpdc7:~ dtomey$ /Users/dtoomey/anaconda/bin/jupyter_mac.command ; exit;
[I 15:40:33.331 NotebookApp] Serving notebooks from local directory: /Users/dtoomey
[I 15:40:33.331 NotebookApp] 0 active kernels
[I 15:40:33.331 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 15:40:33.331 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[I 15:44:57.489 NotebookApp] Creating new notebook in
[I 15:44:58.104 NotebookApp] Kernel started: 03451fbb-0f73-4814-90ff-f53a4d0efae5
[I 15:46:58.062 NotebookApp] Saving file at /Untitled1.ipynb
[I 16:14:30.651 NotebookApp] Saving file at /Learning Jupyter Chapter 2.ipynb
[I 16:24:10.949 NotebookApp] Saving file at /Learning Jupyter Chapter 2.ipynb
[I 16:32:10.965 NotebookApp] Saving file at /Learning Jupyter Chapter 2.ipynb
[I 16:34:32.798 NotebookApp] Saving file at /Learning Jupyter Chapter 2.ipynb
[I 16:35:33.945 NotebookApp] Saving file at /Learning Jupyter Chapter 2.ipynb
```





```
In [ ]: # import the datasets package
        from sklearn import datasets
```

```
In [ ]: # pull in the iris data
        iris_dataset = datasets.load_iris()
        # grab the first two columns of data
        X = iris_dataset.data[:, :2]
```

```
In [ ]: # calculate some basic statistics
        x_count = len(X.flat)
        x_min = X[:, 0].min() - .5
        x_max = X[:, 0].max() + .5
        x_mean = X[:, 0].mean()
```

```
In [ ]: # display our results
        x_count, x_min, x_max, x_mean
```

 **jupyter** Python Data Access (unsaved changes) 

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar



```
In [1]: # import the datasets package
        from sklearn import datasets

In [2]: # pull in the iris data
        iris_dataset = datasets.load_iris()
        # grab the first two columns of data
        X = iris_dataset.data[:, :2]

In [3]: # calculate some basic statistics
        x_count = len(X.flat)
        x_min = X[:, 0].min() - .5
        x_max = X[:, 0].max() + .5
        x_mean = X[:, 0].mean()

In [4]: # display our results
        x_count, x_min, x_max, x_mean

Out[4]: (300, 3.7999999999999998, 8.4000000000000004, 5.8433333333333337)
```

 **jupyter** Python Pandas (autosaved) 

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

```
In [ ]: # we are just using csv handling, but pandas are extensive
        from pandas import *

In [ ]: # we are using the machine learning training set
        training_set = read_csv('train.csv')
        training_set.head()

In [ ]: # break out the set of male vs female
        male = training_set[training_set.sex == 'male']
        female = training_set[training_set.sex == 'female']

In [ ]: # calculate the different survival rates
        womens_survival_rate = float(sum(female.survived))/len(female)
        mens_survival_rate = float(sum(male.survived))/len(male)
        womens_survival_rate
        mens_survival_rate
```


Jupyter Python Pandas (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

				Heath (Lily May Peel)	female	35.0	0	0	17000	8.05
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05

```
In [22]: # break out the set of male vs female
male = training_set[training_set.Sex == 'male']
female = training_set[training_set.Sex == 'female']
```

```
In [23]: # calculate the different survival rates
womens_survival_rate = float(sum(female.Survived))/len(female)
mens_survival_rate = float(sum(male.Survived))/len(male)
womens_survival_rate, mens_survival_rate
```

Out[23]: (0.7420382165605095, 0.18890814558058924)

Jupyter Python Pandas (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

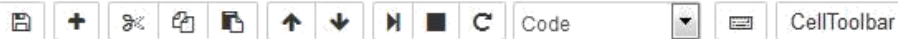
Code CellToolbar

```
In [20]: # we are just using csv handling, but pandas are extensive
from pandas import *
```

```
In [21]: # we are using the machine learning training set
training_set = read_csv('C:/book2/chapter2/train.csv')
training_set.head()
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92



```
In [ ]: import pandas
import matplotlib

# Enable inline plotting
%matplotlib inline
```

```
In [ ]: # define our two columns of data
baby_name = ['Alice', 'Charles', 'Diane', 'Edward']
number_births = [96, 155, 66, 272]
```

```
In [ ]: # create a data set from the two columns
dataset = list(zip(baby_name, number_births))
dataset
```

```
In [ ]: # create a Python dataframe from the dataset
df = pandas.DataFrame(data = dataset, columns=['Name', 'Number'])
df
```

```
In [ ]: # plot the data
df['Number'].plot()
```

```
# Enable inline plotting
%matplotlib inline

In [41]: # define our two columns of data
baby_name = ['Alice', 'Charles', 'Diane', 'Edward']
number_births = [96, 155, 66, 272]

In [42]: # create a data set from the two columns
dataset = list(zip(baby_name, number_births))
dataset

Out[42]: [('Alice', 96), ('Charles', 155), ('Diane', 66), ('Edward', 272)]

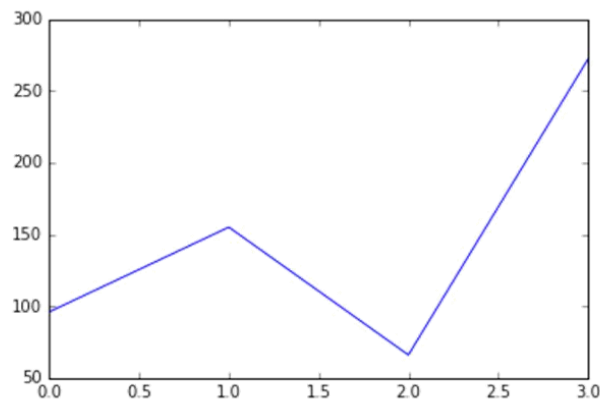
In [43]: # create a Python dataframe from the dataset
df = pandas.DataFrame(data = dataset, columns=['Name', 'Number'])
df
```

Out[43]:

	Name	Number
0	Alice	96
1	Charles	155
2	Diane	66
3	Edward	272

```
In [27]: # plot the data
df['Number'].plot()
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x47cf8f0>



Jupyter Python Random Numbers (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Code CellToolbar

```
In [*]: import pylab
import random

random.seed(113)

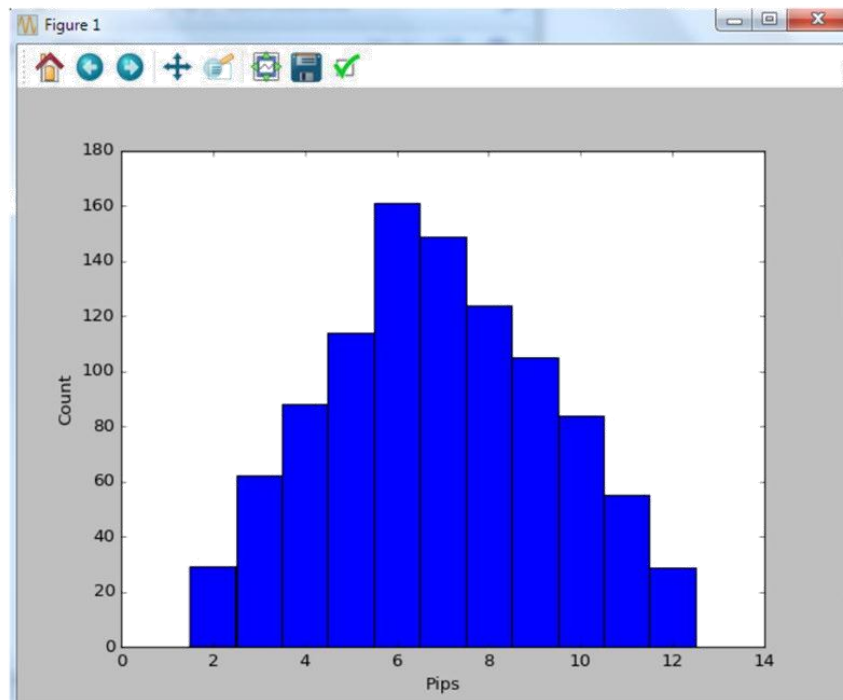
samples = 1000

dice = []
for i in range(samples):
    total = random.randint(1,6) + random.randint(1,6)
    dice.append(total)

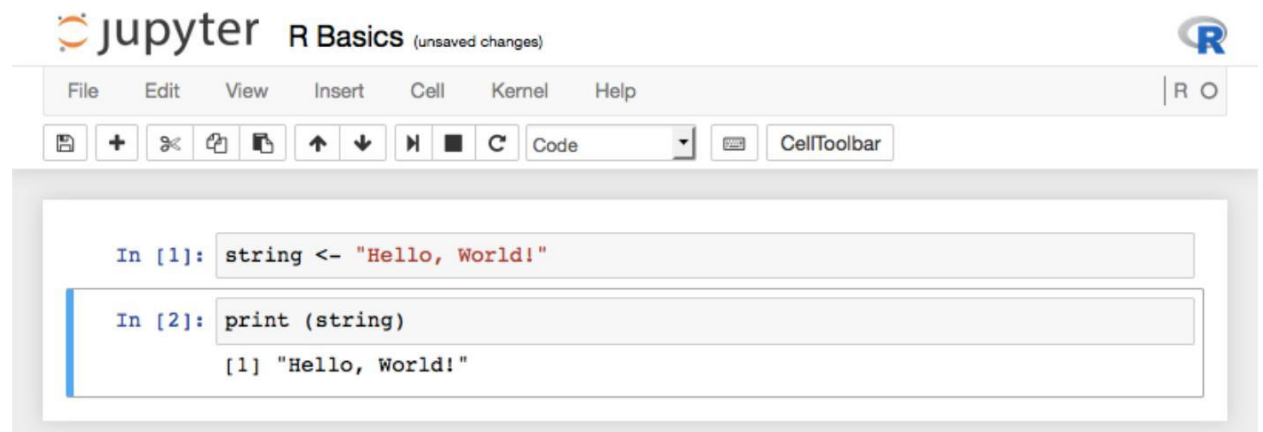
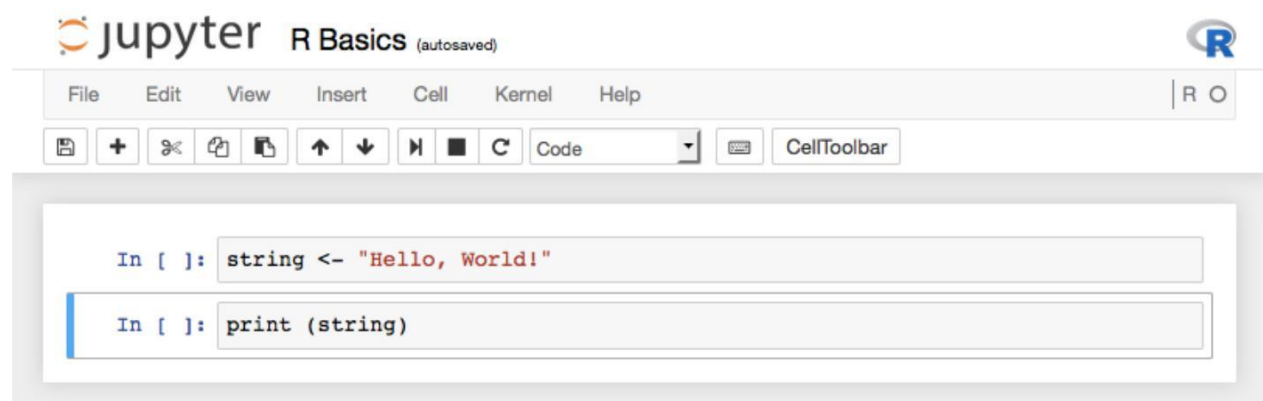
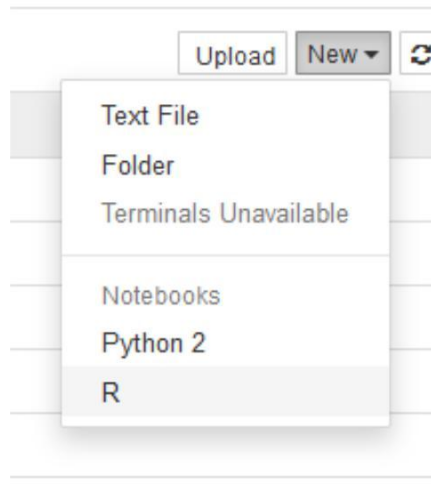
print "Throw two dice", samples, "times."
print "Mean of", pylab.mean(dice)
print "Median of", pylab.median(dice)
print "Std Dev", pylab.std(dice)

pylab.hist(dice, bins= pylab.arange(1.5,12.6,1.0))
pylab.xlabel('Pips')
pylab.ylabel('Count')
pylab.show()
```

Throw two dice 1000 times.
Mean of 6.905
Median of 7.0
Std Dev 2.45397127123



Chapter 3: R Scripting



In []: `data(iris)`

In []: `summary(iris)`

In []: `plot(iris)`

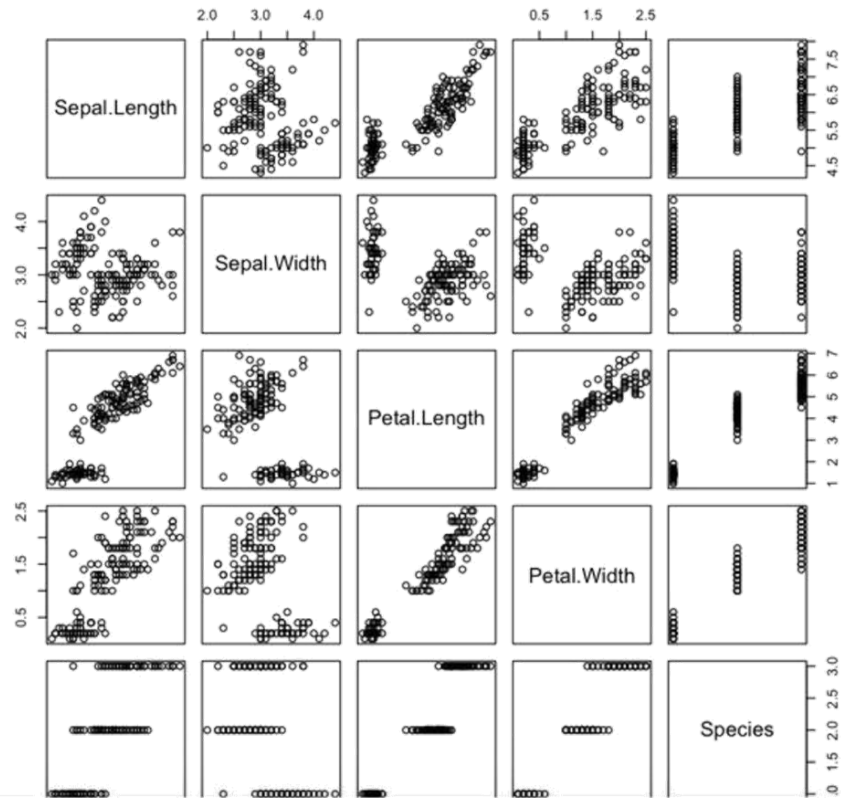
In [1]: `data(iris)`

In [2]: `summary(iris)`

```

Out[2]:
  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
Min.   :4.300  Min.   :2.000  Min.   :1.000  Min.   :0.100
1st Qu.:5.100  1st Qu.:2.800  1st Qu.:1.600  1st Qu.:0.300
Median :5.800  Median :3.000  Median :4.350  Median :1.300
Mean   :5.843  Mean   :3.057  Mean   :3.758  Mean   :1.199
3rd Qu.:6.400  3rd Qu.:3.300  3rd Qu.:5.100  3rd Qu.:1.800
Max.   :7.900  Max.   :4.400  Max.   :6.900  Max.   :2.500
  Species
setosa   :50
versicolor:50
virginica :50
  
```

```
In [3]: plot(iris)
```



localhost:8888/notebooks/persp

Yahzee | Scrabble Word Finder ... | SCRABBLE® Sprint | M... | Scrabble Word Finder ... | draw.io

jupyter persp (autosaved)

File Edit View Insert Cell Kernel Help | R O

CellToolbar

```
In [ ]: example(persp)
```

```
In [1]: example(persp)

persp> require(grDevices) # for trans3d

persp> ## More examples in demo(persp) !!
persp> ## -----
persp> 
persp> # (1) The Obligatory Mathematical surface.
persp> #   Rotated sinc function.
persp> 
persp> x <- seq(-10, 10, length= 30)

persp> y <- x

persp> f <- function(x, y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }

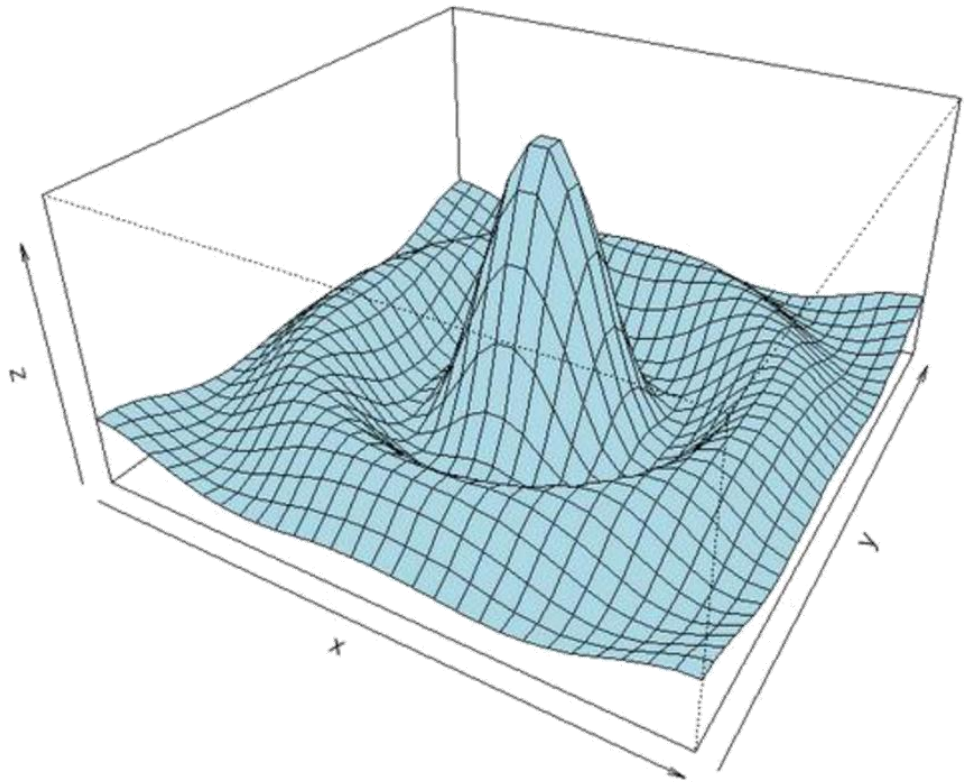
persp> z <- outer(x, y, f)

persp> z[is.na(z)] <- 1

persp> op <- par(bg = "white")

persp> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue"
)

persp> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue"
```



make sure lattice package is installed

```
In [11]: library("lattice")
```

use the automobile data from ics.edu use the automobile data from ics.edu

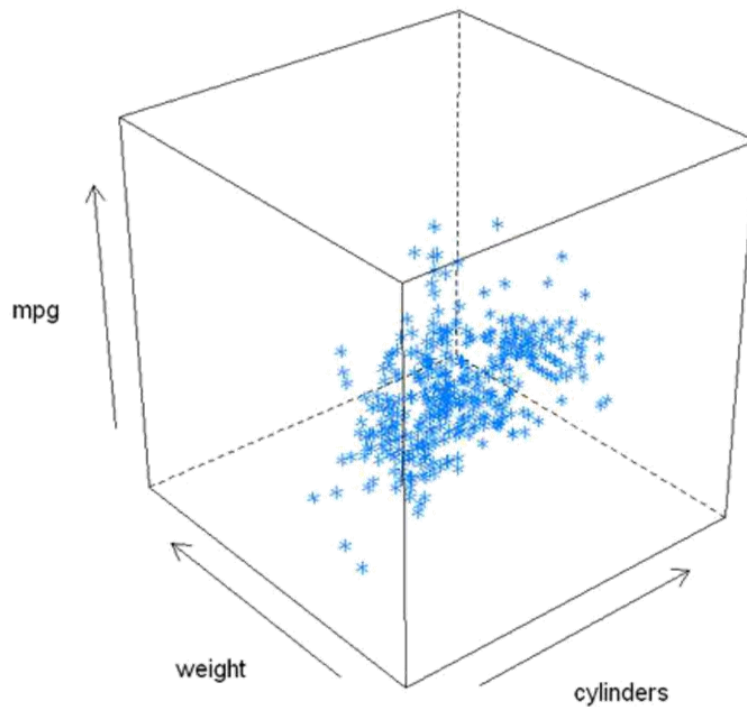
```
In [12]: mydata <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/e
< |-----| >
```

define more meaningful column names for the display

```
In [13]: colnames(mydata) <- c("mpg", "cylinders", "displacement", "horsepower", "weight"
< |-----| >
```

3-D plot with number of cylinders on x axis, weight of the vehicle on the y axis and miles per gallon on the z axis

```
In [ ]: cloud(mpg~cylinders*weight, data=mydata)
```



load the wheat data set from uci.edu

```
In [ ]: wheat <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/002
```

define useful column names

```
In [ ]: colnames(wheat) <-c("area", "perimeter", "compactness", "length", "width", "asy
```

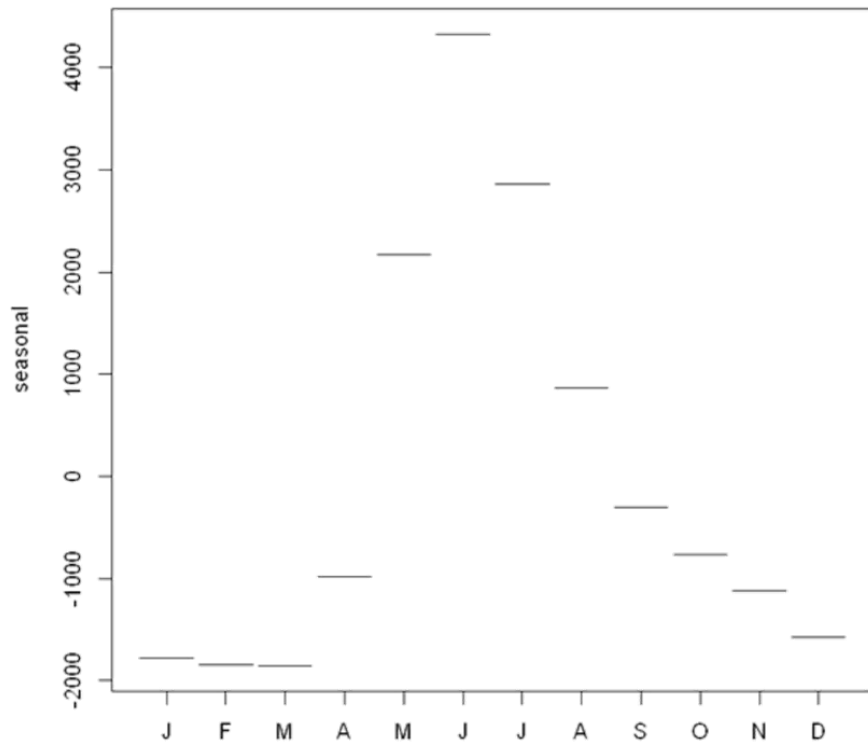
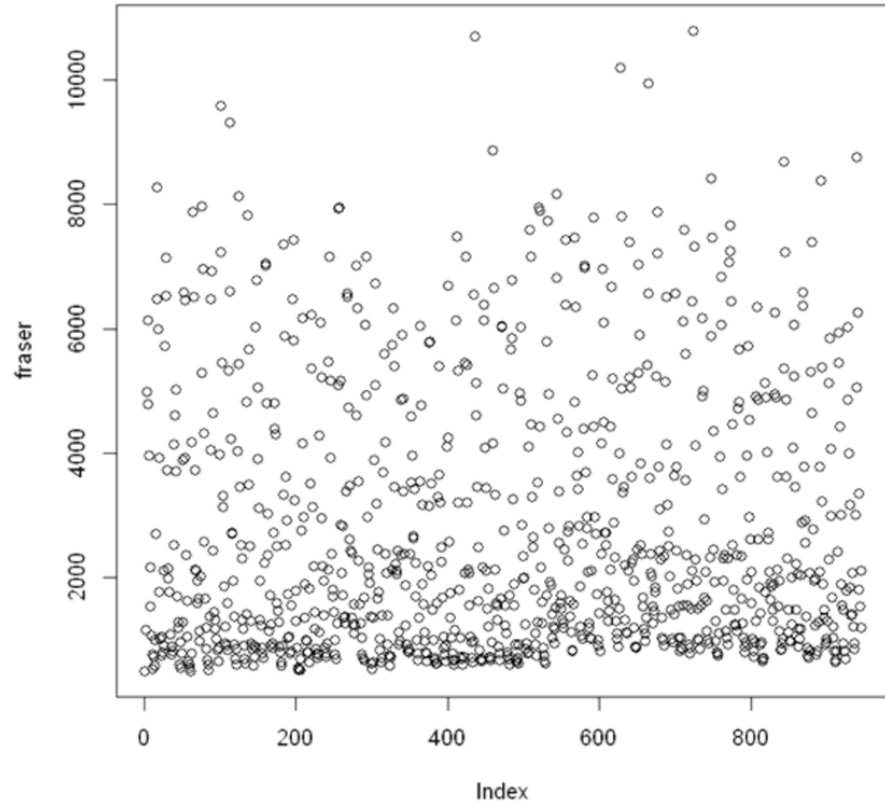
exclude incomplete cases from the data

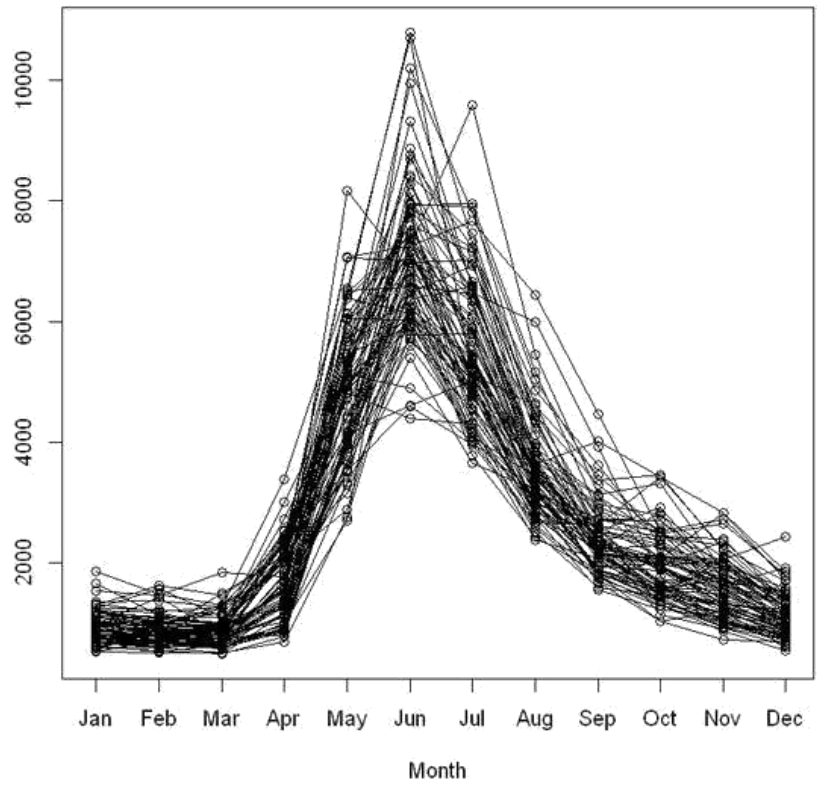
```
In [ ]: wheat <- wheat[complete.cases(wheat),]
```

```
In [ ]: calculate the clusters
```

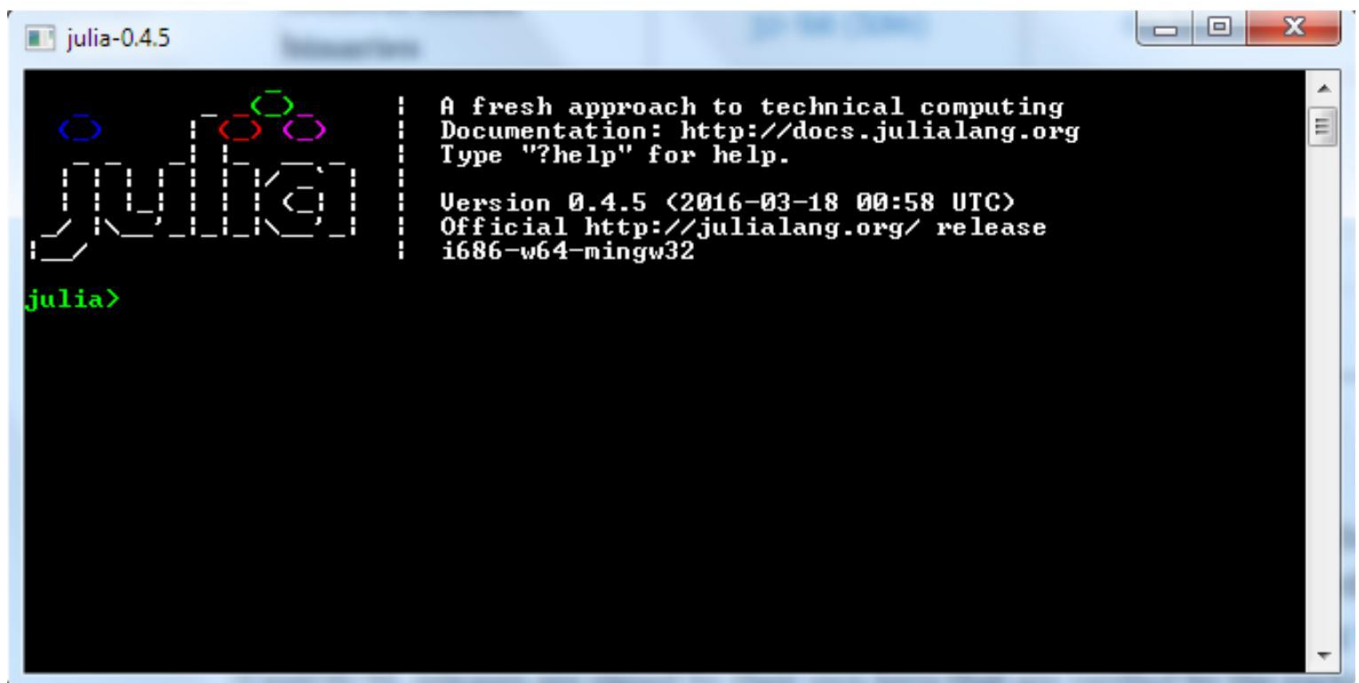
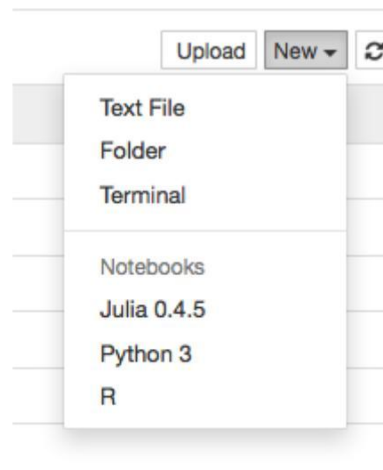
```
In [ ]: fit <- kmeans(wheat, 5)
```

```
In [ ]: fit
```

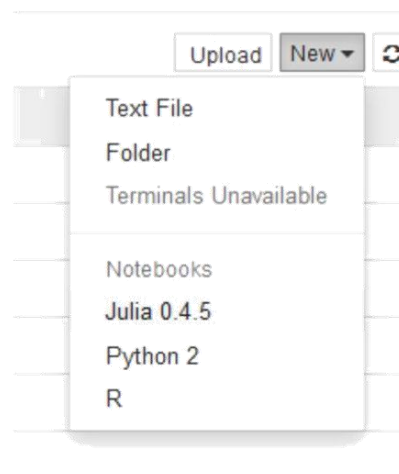




Chapter 4: Julia Scripting



```
Mark julia-0.4.5
julia> Pkg.add("IJulia")
INFO: Cloning cache of BinDeps from git://github.com/JuliaLang/BinDeps.jl.git
INFO: Cloning cache of Compat from git://github.com/JuliaLang/Compat.jl.git
INFO: Cloning cache of Conda from git://github.com/lathaf/Conda.jl.git
INFO: Cloning cache of IJulia from git://github.com/JuliaLang/IJulia.jl.git
INFO: Cloning cache of JSON from git://github.com/JuliaLang/JSON.jl.git
INFO: Cloning cache of LibExpat from git://github.com/amitmurthy/LibExpat.jl.git
INFO: Cloning cache of Nettle from git://github.com/staticfloat/Nettle.jl.git
INFO: Cloning cache of SHA from git://github.com/staticfloat/SHA.jl.git
INFO: Cloning cache of URIParser from git://github.com/JuliaWeb/URIParser.jl.git
INFO: Cloning cache of WinRPM from git://github.com/JuliaLang/WinRPM.jl.git
INFO: Cloning cache of ZMQ from git://github.com/JuliaLang/ZMQ.jl.git
INFO: Cloning cache of Zlib from git://github.com/dcjones/Zlib.jl.git
INFO: Installing BinDeps v0.3.21
INFO: Installing Compat v0.7.18
INFO: Installing Conda v0.2.0
INFO: Installing IJulia v1.1.9
INFO: Installing JSON v0.5.0
INFO: Installing LibExpat v0.1.2
INFO: Installing Nettle v0.2.3
INFO: Installing SHA v0.1.2
INFO: Installing URIParser v0.1.3
INFO: Installing WinRPM v0.1.15
INFO: Installing ZMQ v0.3.1
INFO: Installing Zlib v0.1.12
INFO: Building WinRPM
WARNING: skipping repodata/repomd.xml, not in cache -- call WinRPM.update() to d
ownload
WARNING: skipping repodata/repomd.xml, not in cache -- call WinRPM.update() to d
```



localhost:8888/notebooks/Julia: Search

jupyter Julia Iris (unsaved changes) Julia 0.4.5

File Edit View Insert Cell Kernel Help

Code CellToolbar

```
In [ ]: using RDatasets, DataFrames, Gadfly

        define the size of the plot area

In [ ]: set_default_plot_size(5inch, 5inch/golden);

        plot out the iris sepal width by the sepal length

In [ ]: describe(dataset("datasets", "iris"))

In [ ]: plot(dataset("datasets", "iris"), x="SepalWidth", y="SepalLength", color="Species")
```

```
In [7]: using RDatasets, DataFrames
```

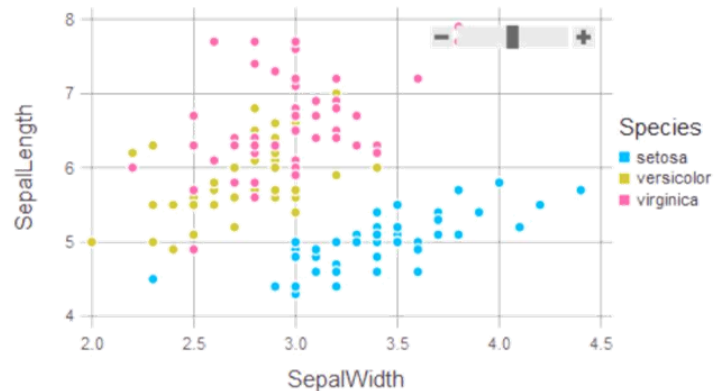
define the size of the plot area

```
In [8]: set_default_plot_size(5inch, 5inch/golden);
```

plot out the iris sepal width by the sepal length

```
In [9]: plot(dataset("datasets", "iris"), x="SepalWidth", y="SepalLength", color="Species")
```

Out[9]:





```
In [*]: using Gadfly
        plot(x=rand(7), y=rand(7))
```

```
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\ArrayViews.ji for module ArrayViews.
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\StatsBase.ji for module StatsBase.
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\StatsFuns.ji for module StatsFuns.
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\Gadfly.ji for module Gadfly.
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\Codecs.ji for module Codecs.
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\FixedPointNumbers.ji for module FixedPointNumbers.
INFO: Recompiling stale cache file C:\Users\Dan\.julia\lib\v0.4\Colors.ji for module Colors.
```



```
In [1]: using RDatasets
        describe(dataset("datasets", "iris"))
```

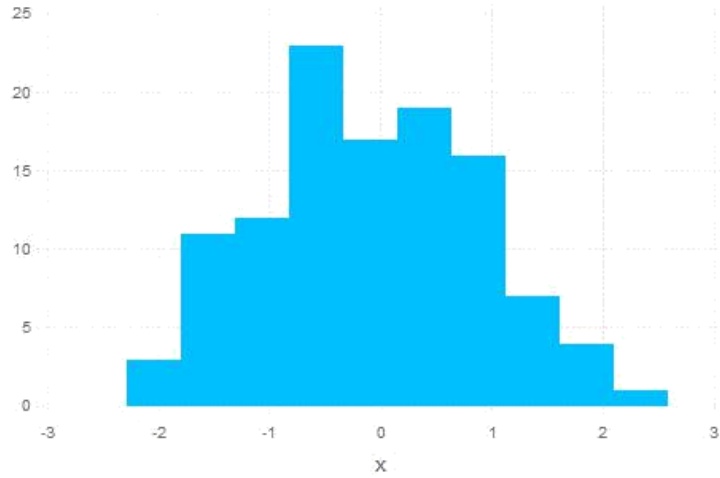
```
SepalLength
Min      4.3
1st Qu.  5.1
Median   5.8
Mean     5.843333333333332
3rd Qu.  6.4
Max      7.9
NAs      0
NA%      0.0%
```

```
SepalWidth
Min      2.0
```




```
In [1]: using Gadfly
        srand(111)
        plot(x=randn(113), Geom.histogram(bincount=10))
```

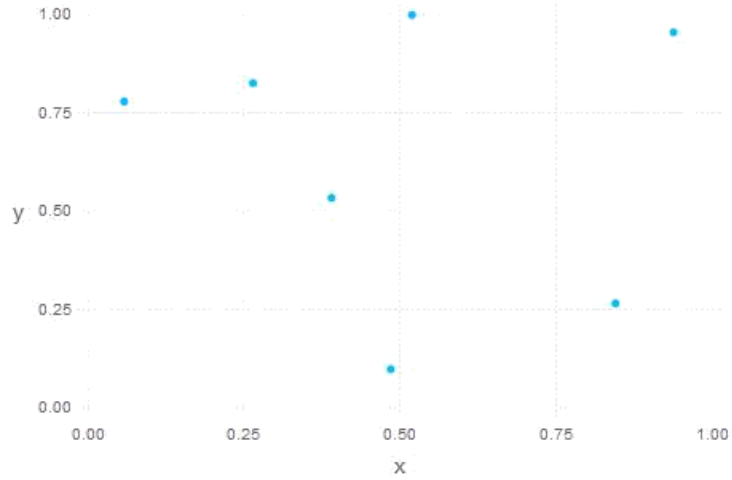
Out[1]:





```
In [1]: using Gadfly
        srand(111)
        plot(x=rand(7), y=rand(7))
```

Out[1]:

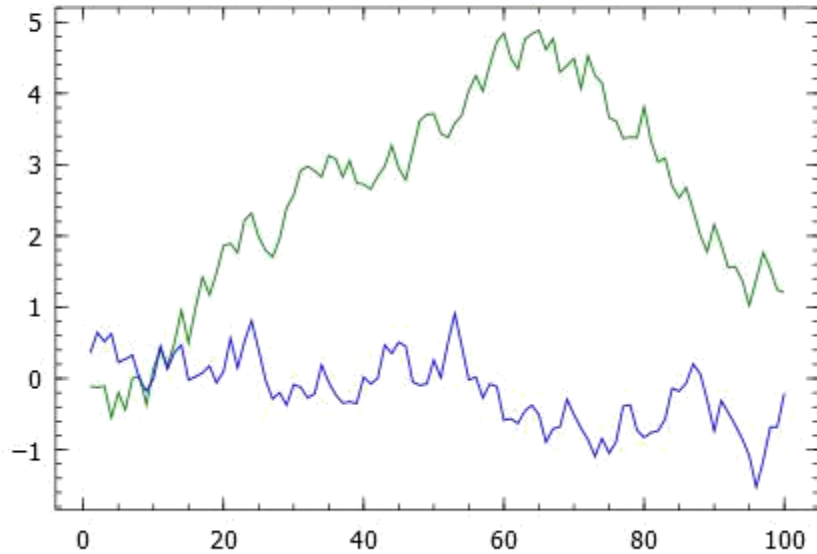


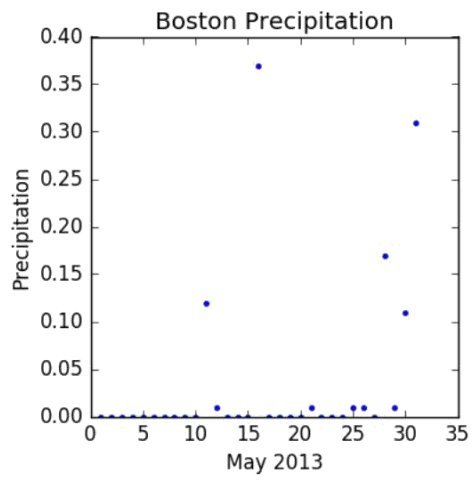
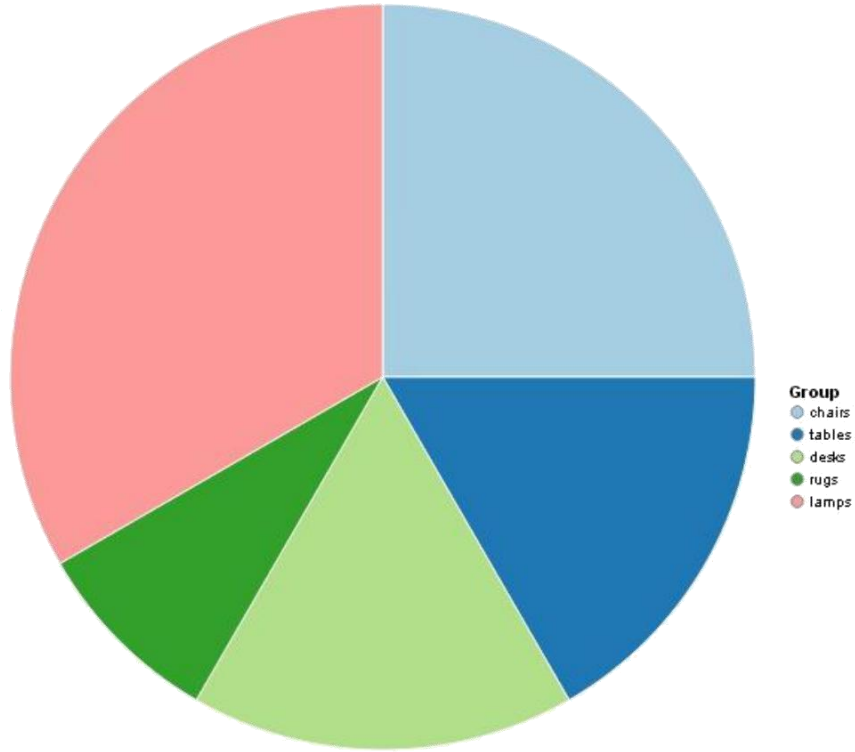


```
In [8]: using Winston
        srand(111)

        # generate a plot
        pl = plot(cumsum(rand(100) .- 0.5), "g", cumsum(rand(100) .- 0.5))

        # display the plot
        display(pl)
```







```
In [1]: addprocs(1)
        srand(111)
        r = remotecall(rand, 2, 3, 4)
        s = @spawnat 2 1 .+ fetch(r)
        fetch(s)
```

```
Out[1]: 3x4 Array{Float64,2}:
 1.17558  1.35232  1.9891  1.22328
 1.36165  1.2355  1.39344  1.17494
 1.95311  1.30926  1.54958  1.84229
```

```
In [3]: function larger(x, y)
        if (x>y)
            return x
        end
        return y
    end
    println(larger(7,8))
```

8



```
In [2]: ismatch(r"^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", "(781)244-1212")
```

```
Out[2]: true
```

```
In [1]: ismatch(r"^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", "-781-244-1212")
```

```
Out[1]: false
```



```
In [13]: using FactCheck
          f(x) = x^3
          facts("cubes") do
              @fact f(2) --> 8
              @fact f(2) --> 7
          end

          cubes
          Failure :: (line:-1) :: fact was false
            Expression: f(2) --> 7
            Expected: 7
            Occurred: 8
          Out of 2 total facts:
            Verified: 1
            Failed: 1

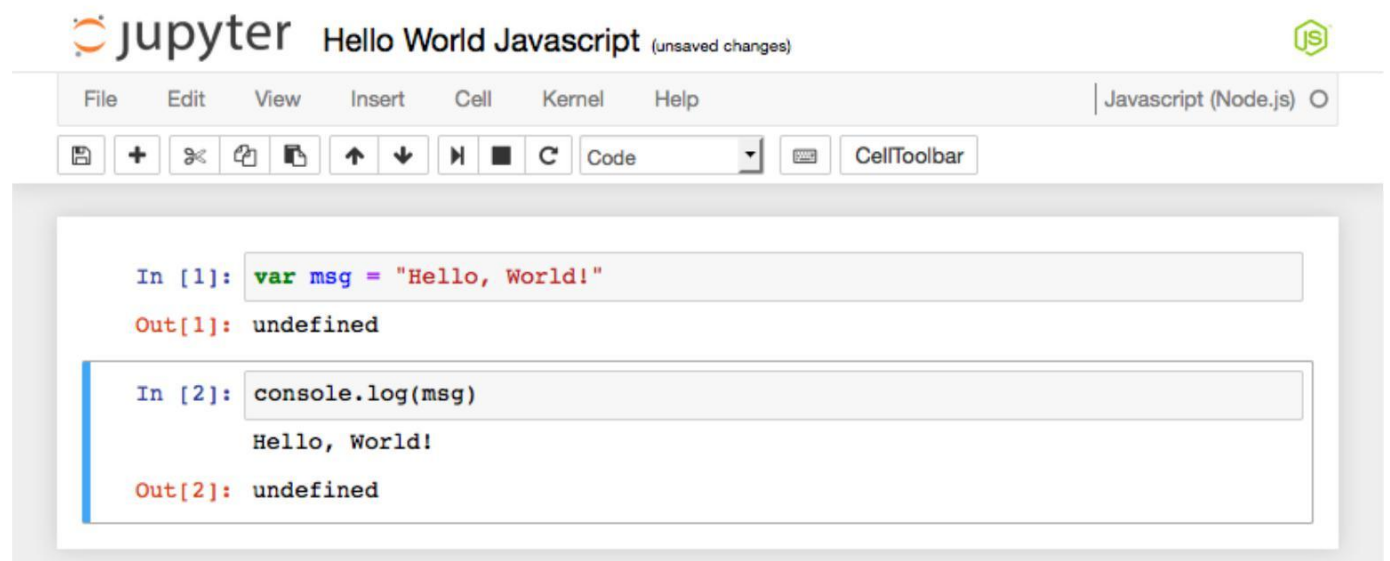
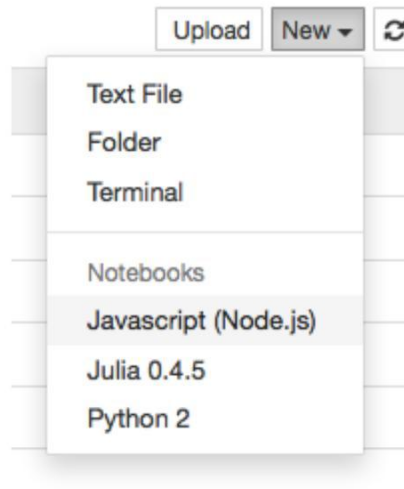
          Out[13]: delayed_handler (generic function with 4 methods)
```

```
In [9]: using Base.Test
        f(x) = x^3
        @test f(2) == 8
        @test f(2) == 7
```

```
LoadError: test failed: 8 == 7
  in expression: f(2) == 7
while loading In[9], in expression starting on line 4

  in error at error.jl:21
  in default_handler at test.jl:28
  in do_test at test.jl:53
```

Chapter 5: JavaScript Coding



A screenshot of the Jupyter notebook interface. The title bar shows 'jupyter Hello World Javascript (unsaved changes)' and a JavaScript icon. The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The toolbar contains icons for file operations and a 'Code' dropdown menu. The notebook content consists of two code cells:

```
In [1]: var msg = "Hello, World!"  
Out[1]: undefined
```

```
In [2]: console.log(msg)  
        Hello, World!  
Out[2]: undefined
```



```
In [1]: var fs = require("fs");
var d3 = require("d3");
var _ = require("lodash");

fs.readFile("data/animals.tsv", "utf8", function(error, data) {
  data = d3.tsv.parse(data);
  console.log(JSON.stringify(data));

  var maxWeight = d3.max(data, function(d) { return d.avg_weight; });
  console.log(maxWeight);
});
```

Out[1]: undefined

```
[{"name": "lion", "avg_weight": "400"}, {"name": "tiger", "avg_weight": "400"}, {
  "name": "human", "avg_weight": "150"}, {"name": "elephant", "avg_weight": "2000"}]
400
```

```
In [17]: const stats = require("stats-analysis");

var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];

//standard deviation
var my_stddev = stats.stdev(arr).toFixed(2);

//mean
var my_mean = stats.mean(arr).toFixed(2);

//median
var my_median = stats.median(arr);

//median absolute deviation
var my_mad = stats.MAD(arr);

// Outlier detection. Returns indexes of outliers
var my_outliers = stats.indexOfOutliers(arr);

// Remove the outliers
var my_without_outliers = stats.filterOutliers(arr);

//display our stats
console.log("Raw data is ", arr);
console.log("Standard Deviation is ", my_stddev);
console.log("Mean is ", my_mean);
console.log("Median is ", my_median);
console.log("Median Abs Deviation is " + my_mad);
console.log("The outliers of the data set are ", my_outliers);
console.log("The data set without outliers is ", my_without_outliers);
```



```
In [27]: //load the JSON dataset
//http://www.carqueryapi.com/api/0.3/?callback=?&cmd=getModels&make=ford
var fords = require('/Users/dtoomey/fords.json');

//display how many Ford models are in our data set
console.log("There are " + fords.Models.length + " Ford models in the data set");

//loop over the set
var index = 1
for(var i=0; i<fords.Models.length; i++) {

    //get this model
    var model = fords.Models[i];

    //pull it's name
    var name = model.model_name;

    //if the model name does not have numerics in it
    if(! name.match(/[0-9]/i)) {
        //display the model name
        console.log("Model " + index + " is a " + name);
        index++;
    }

    //only display the first 5
    if (index>5) break;
}
```

```

There are 147 Ford models in the data set
Model 1 is a Aerostar
Model 2 is a Anglia
Model 3 is a Artic
Model 4 is a Aspire
Model 5 is a Bantam

Out[27]: 5

```

```
In [8]: // create a canvas 200 by 200 pixels
var Canvas = require('canvas')
    , Image = Canvas.Image
    , canvas = new Canvas(200, 200)
    , ctx = canvas.getContext('2d')
    , string = "Jupyter!";

// place our string on the canvas
ctx.font = '30px Impact';
ctx.rotate(.1);
ctx.fillText(string, 50, 100);

var te = ctx.measureText(string);
ctx.strokeStyle = 'rgba(0,0,0,0.5)';
ctx.beginPath();
ctx.lineTo(50, 102);
ctx.lineTo(50 + te.width, 102);
ctx.stroke();

//create an html img tag, with embedded graphics
console.log('');

<img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAMgAAADICAYAAActW
K6eAAAABmJLR0QA/wD/AP+gvaetAAAJYE1EQVR4nO3da4wVZxnA8T8LRviUUmhLEWhTa9W2Vm
M1rjRGjbdovFStEGO9JUaNMd7SePvgLX7pF0U1Jka/aLVpqomXSqxaL8SqtKZGKVraAtzSFrZ
```

Jupyter!

```
In [14]: //set random seed
var seedrandom = require('seedrandom');
var rng = seedrandom('Jupyter');

//setup plotly
var plotly = require('plotly')(username=<username>, api_key=<api key>)

var x = [];

for (var i = 0; i < 500; i ++) {
    x[i] = Math.random();
}

require('plotly')(username, api_key);

var data = [
    {
        x: x,
        type: "histogram"
    }
];
var graphOptions = {filename: "basic-histogram", fileopt: "overwrite"};
plotly.plot(data, graphOptions, function (err, msg) {
    console.log(msg);
});
```

```
Out[14]: undefined
{ streamstatus: undefined,
  url: 'https://plot.ly/~dantoomey/1',
```

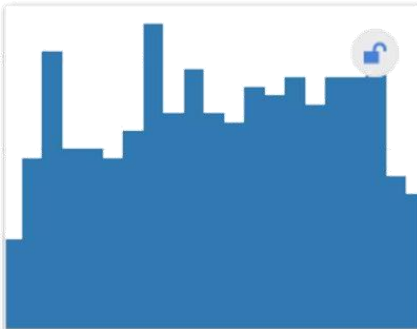
+ NEW

- Home
- Shared with me
- Recent
- Favorite
- Trash

Home

SHOW

Plots
Grids
Folders
Private
Public



basic-histogram

a few seconds ago 0 views

x
0.387097295607
0.279918756103
0.833789543016
0.23877594457
0.244038771605
0.995358167449
0.156761112856
0.540174414404
0.148360050982

basic-histogram Grid

a few seconds ago 0 views



File Edit View Insert Cell Kernel Help Javascript (Node.js)

+
⌕
↺
↻
↵
⏪
⏩
⏹
Code
CellToolbar

```

In [5]: //thread function - invoked for every number in items array
function async(arg, callback) {
  console.log('triple \''+arg+'\ ', and return 2 seconds later');
  setTimeout(function() { callback(arg * 3); }, 2000);
}

//function called once - after all threads complete
function final() { console.log('Done', results); }

//list of numbers to operate upon
var items = [ 0, 1, 1, 2, 3, 5, 7, 11 ];

//results of each step
var results = [];

//loop the drives the whole process
items.forEach(function(item) {
  async(item, function(result){
    results.push(result);
    if(results.length == items.length) {
      final();
    }
  })
});
    
```

```
triple '0', and return 2 seconds later
triple '1', and return 2 seconds later
triple '1', and return 2 seconds later
triple '2', and return 2 seconds later
triple '3', and return 2 seconds later
triple '5', and return 2 seconds later
triple '7', and return 2 seconds later
triple '11', and return 2 seconds later
```

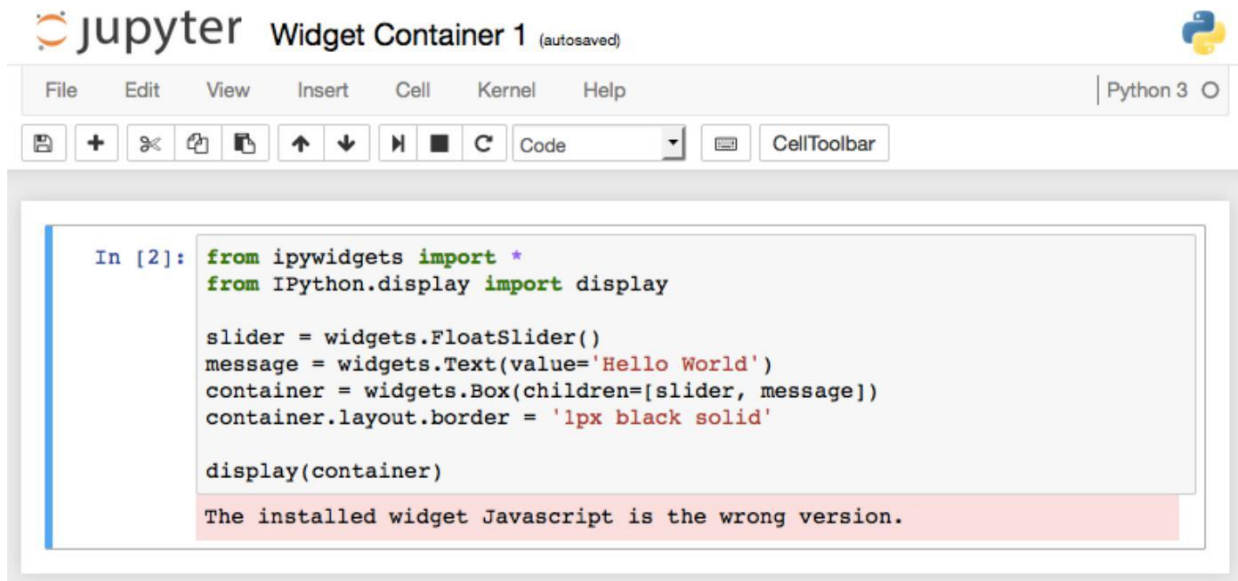
Out[5]: undefined

```
Done [ 0, 3, 3, 6, 9, 15, 21, 33 ]
```

Out[23]: undefined

```
rows = 42
training_size = 28
test_size = 14
Decision Tree is { data:
  [ { 'mpg,cylinders,displacement,horsepower,weight,acceleration,modelyear,maker': 'Bad,8,400,170,4746,12,71,America' },
    { 'mpg,cylinders,displacement,horsepower,weight,acceleration,modelyear,maker': 'Bad,8,400,170,4746,12,71,America' } ] }
```

Chapter 6: Interactive Widgets



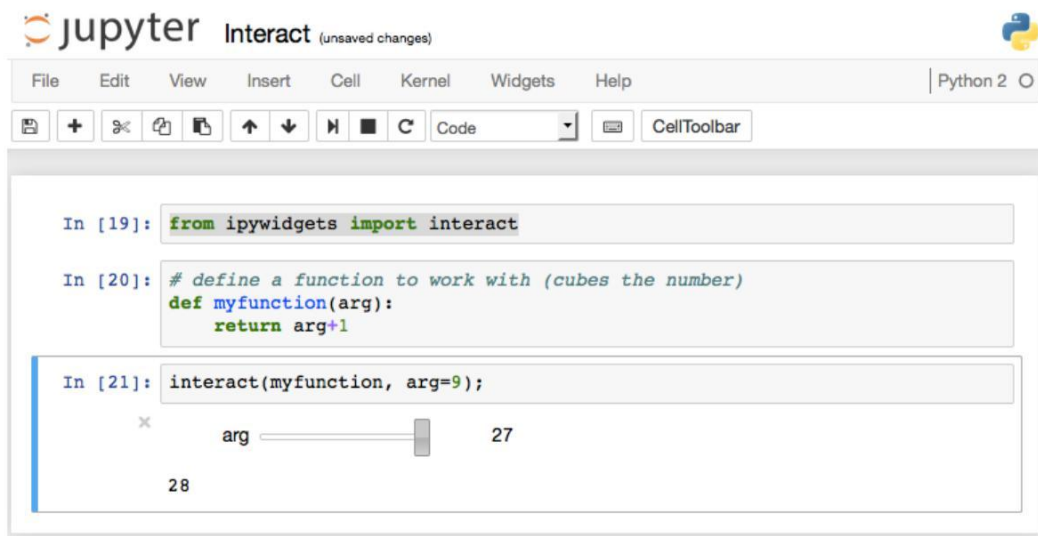
The screenshot shows a Jupyter Notebook titled "Widget Container 1 (autosaved)" running on Python 3. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The code cell contains the following Python code:

```
In [2]: from ipywidgets import *
        from IPython.display import display

        slider = widgets.FloatSlider()
        message = widgets.Text(value='Hello World')
        container = widgets.Box(children=[slider, message])
        container.layout.border = '1px black solid'

        display(container)
```

Below the code, a red error message is displayed: "The installed widget Javascript is the wrong version."



The screenshot shows a Jupyter Notebook titled "Interact (unsaved changes)" running on Python 2. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar. The code cell contains the following Python code:

```
In [19]: from ipywidgets import interact

In [20]: # define a function to work with (cubes the number)
        def myfunction(arg):
            return arg+1

In [21]: interact(myfunction, arg=9);
```

Below the code, an interactive slider widget is displayed. The slider is labeled "arg" and has a value of 27. The output of the function is 28.



```
In [4]: from ipywidgets import interact
```

```
In [5]: def myfunction(x):  
        return x
```

```
In [6]: interact(myfunction, x=False);
```

x x

True



```
In [4]: from ipywidgets import interact
```

```
In [5]: def myfunction(x):  
        return x
```

```
In [6]: interact(myfunction, x="Hello World");
```

x

u'Hello World'



```
In [4]: from ipywidgets import interact
```

```
In [5]: def myfunction(x):  
        return x
```

```
In [6]: interact(myfunction, x=('red', 'green'));
```

x

x

u'green'



```
In [5]: from ipywidgets import interactive
```

```
In [6]: def myfunction(x):  
        return x
```

```
In [7]: w = interactive(myfunction, x= "Hello World ");
```

```
In [8]: from IPython.display import display  
        display(w)
```

x

x

u'Hello World '



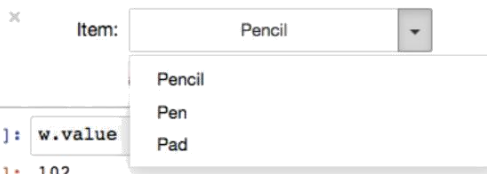
```
In [5]: import ipywidgets as widgets
```

```
In [6]: widgets.FloatProgress(  
    value=45,  
    min=0,  
    max=100,  
    step=5,  
    description='Percent:',  
)
```



```
In [5]: import ipywidgets as widgets  
        from IPython.display import display
```

```
In [6]: w = widgets.Dropdown(  
    options={'Pen': 7732, 'Pencil': 102, 'Pad': 33331},  
    description='Item:',  
)  
        display(w)
```



```
In [7]: w.value
```

```
Out[7]: 102
```



```
In [1]: from ipywidgets import widgets
        from IPython.display import display
```

```
In [2]: def handle_submit(sender):
        print(text.value)
```

```
In [3]: text = widgets.Text()
        text.on_submit(handle_submit)
        display(text)
```

x

Dan



```
In [7]: from ipywidgets import widgets
        from IPython.display import display
        text = widgets.Text()
        display(text)
```

x

```
In [8]: def handle_submit(sender):
        print(text.value)
```

```
In [9]: text.on_submit(handle_submit)
```



```
In [4]: from ipywidgets import widgets
        from IPython.display import display
        text = widgets.Text()
        display(text)
```

test

test

```
In [5]: def handle_submit(sender):
        print(text.value)
```

```
In [6]: text.on_submit(handle_submit)
```



```
In [13]: from ipywidgets import widgets
        from IPython.display import display

        button = widgets.Button(description="Submit");
        display(button)

        def on_button_clicked(widget):
            print("Clicked Button:" + widget.description);

        button.on_click(on_button_clicked);
```

Submit

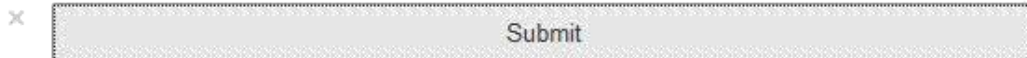


```
In [13]: from ipywidgets import widgets
         from IPython.display import display

         button = widgets.Button(description="Submit");
         display(button)


         def on_button_clicked(widget):
             print("Clicked Button:" + widget.description);

         button.on_click(on_button_clicked);
```



Clicked Button:Submit

```
Out[1]: ['_view_name',
         'orientation',
         'color',
         '_view_module',
         'height',
         'disabled',
         'visible',
         'border_radius',
         'border_width',
         '_model_module',
         'font_style',
         'layout',
         'min',
         '_range',
         'background_color',
         'slider_color',
         'width',
         'continuous_update',
         'font_family',
         '_dom_classes',
         'description',
         '_model_name',
         'max',
         'border_color',
         'readout',
         'padding',
         'font_weight',
         'step',
         'border_style',
         'font_size',
         'msg_throttle',
         'value',
         'margin']
```

 **Jupyter** Adjust Property (unsaved changes)



File Edit View Insert Cell Kernel Widgets Help

Python 2

          Code  CellToolbar

```
In [1]: from ipywidgets import *
        Text(value='You can not change this text!', disabled=True)|
```





```
In [2]: from ipywidgets import *
w = IntSlider()
original = w.value
w.value = 5
original, w.value
```

Out[2]: (0, 5)



```
In [3]: from ipywidgets import widgets
from IPython.display import display
```

```
In [4]: button = widgets.Button(description="Click Me!")
display(button)
```



```
Button clicked.
Button clicked.
Button clicked.
Button clicked.
```

```
In [5]: def on_button_clicked(b):
        print("Button clicked.")

        button.on_click(on_button_clicked)
```



```
In [2]: from ipywidgets import *
        from IPython.display import display

        slider = widgets.FloatSlider()
        message = widgets.Text(value='Hello World')
        container = widgets.Box(children=[slider, message])
        container.layout.border = '1px black solid'

        display(container)
```

x



```
In [1]: from ipywidgets import *
        from IPython.display import display

        container = widgets.Box()
        container.layout.border = '1px black solid'
        display(container)

        slider = widgets.FloatSlider()
        message = widgets.Text(value='Hello World')
        container.children=[slider, message]
```

x

Chapter 7: Sharing and Converting Jupyter Notebooks

The screenshot shows a GitHub repository page for 'danieltoomey / notebooks'. At the top, there is a navigation bar with 'Code', 'Issues 0', 'Pull requests 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. Below this, a message states 'No description or website provided. — Edit'. A summary bar indicates '2 commits', '1 branch', and '0 releases'. Below the summary bar, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', and 'Finish'. A commit message from 'danieltoomey' is visible, stating 'Add files via upload'. Below the commit message, there is a list of files: 'README.md' (first commit) and 'Stats Analysis.ipynb' (Add files via upload). At the bottom, there is a section for 'README.md' with the word 'notebooks' displayed in a large, bold font.

danieltoomey / notebooks Unwatch 1

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

2 commits 1 branch 0 releases

Branch: master New pull request Create new file Upload files Finish

danieltoomey committed on GitHub Add files via upload Latest

README.md first commit

Stats Analysis.ipynb Add files via upload

README.md

notebooks

Branch: master

notebooks / Stats Analysis.ipynb

 **danieltoomey** Add files via upload

1 contributor

85 lines (84 sloc) | 2.18 KB

```
In [17]: const stats = require("stats-analysis");

var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];

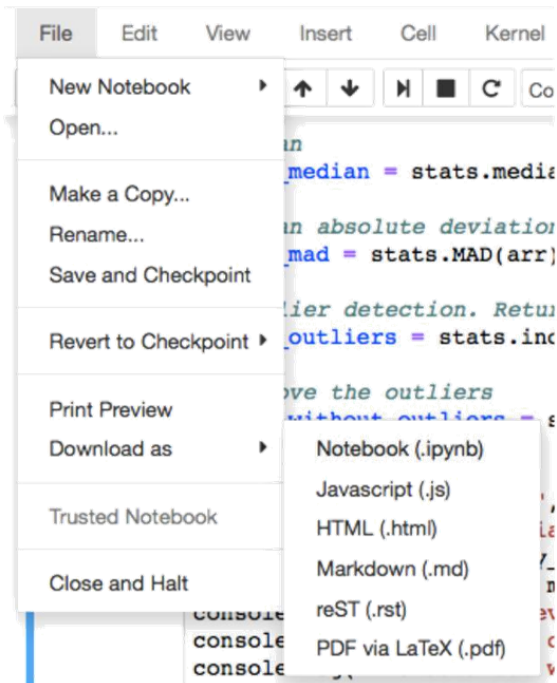
//standard deviation
var my_stddev = stats.stdev(arr).toFixed(2);

//mean
var my_mean = stats.mean(arr).toFixed(2);

//median
var my_median = stats.median(arr);

//median absolute deviation
var my_mad = stats.MAD(arr);

// Outlier detection. Returns indexes of outliers
var my_outliers = stats.indexOfOutliers(arr);
```



```
In [17]: const stats = require("stats-analysis");

var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];

//standard deviation
var my_stddev = stats.stdev(arr).toFixed(2);

//mean
var my_mean = stats.mean(arr).toFixed(2);

//median
var my_median = stats.median(arr);

//median absolute deviation
var my_mad = stats.MAD(arr);

// Outlier detection. Returns indexes of outliers
var my_outliers = stats.indexOfOutliers(arr);

// Remove the outliers
var my_without_outliers = stats.filterOutliers(arr);

//display our stats
console.log("Raw data is ", arr);
```

```
In [17]: const stats = require("stats-analysis");

var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];

//standard deviation
var my_stddev = stats.stdev(arr).toFixed(2);

//mean
var my_mean = stats.mean(arr).toFixed(2);

//median
var my_median = stats.median(arr);

//median absolute deviation
var my_mad = stats.MAD(arr);

// Outlier detection. Returns indexes of outliers
var my_outliers = stats.indexOfOutliers(arr);

// Remove the outliers
var my_without_outliers = stats.filterOutliers(arr);

//display our stats
console.log("Raw data is ", arr);
console.log("Standard Deviation is ", my_stddev);
console.log("Mean is ", my_mean);
console.log("Median is ", my_median);
console.log("Median Abs Deviation is " + my_mad);
console.log("The outliers of the data set are ", my_outliers);
console.log("The data set without outliers is ", my_without_outliers);
```

```
Raw data is [ 98, 98.6, 98.4, 98.8, 200, 120, 98.5 ]
Standard Deviation is 35.07
Mean is 116.04
Median is 98.6
Median Abs Deviation is 0.200000000000000284
The outliers of the data set are [ 4, 5, 6 ]
The data set without outliers is [ 98, 98.6, 98.4, 98.8 ]
```

Out[17]: undefined

```
Stats+Analysis-1.rst | Stats+Analysis.md
4  const stats = require("stats-analysis");
5
6  var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];
7
8  //standard deviation
9  var my_stddev = stats.stdev(arr).toFixed(2);
10
11 //mean
12 var my_mean = stats.mean(arr).toFixed(2);
13
14 //median
15 var my_median = stats.median(arr);
16
17 //median absolute deviation
18 var my_mad = stats.MAD(arr);
19
20 // Outlier detection. Returns indexes of outliers
21 var my_outliers = stats.indexOfOutliers(arr);
22
23 // Remove the outliers
24 var my_without_outliers = stats.filterOutliers(arr);
25
26 //display our stats
27 console.log("Raw data is ", arr);
28 console.log("Standard Deviation is ", my_stddev);
29 console.log("Mean is ", my_mean);
30 console.log("Median is ", my_median);
31 console.log("Median Abs Deviation is " + my_mad);
32 console.log("The outliers of the data set are ", my_outliers);
33 console.log("The data set without outliers is ", my_without_outliers);
34
35
36 ...
37
38 Raw data is [ 98, 98.6, 98.4, 98.8, 200, 120, 98.5 ]
```

```
6   var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];
7
8   //standard deviation
9   var my_stddev = stats.stdev(arr).toFixed(2);
10
11  //mean
12  var my_mean = stats.mean(arr).toFixed(2);
13
14  //median
15  var my_median = stats.median(arr);
16
17  //median absolute deviation
18  var my_mad = stats.MAD(arr);
19
20  // Outlier detection. Returns indexes of outliers
21  var my_outliers = stats.indexOfOutliers(arr);
22
23  // Remove the outliers
24  var my_without_outliers = stats.filterOutliers(arr);
25
26  //display our stats
27  console.log("Raw data is ", arr);
28  console.log("Standard Deviation is ", my_stddev);
29  console.log("Mean is ", my_mean);
30  console.log("Median is ", my_median);
31  console.log("Median Abs Deviation is " + my_mad);
32  console.log("The outliers of the data set are ", my_outliers);
33  console.log("The data set without outliers is ", my_without_outliers);
34
35
36
37
38  .. parsed-literal::
39
40  Raw data is [ 98, 98.6, 98.4, 98.8, 200, 120, 98.5 ]
```

Stats Analysis

July 14, 2016

```
In [17]: const stats = require("stats-analysis");

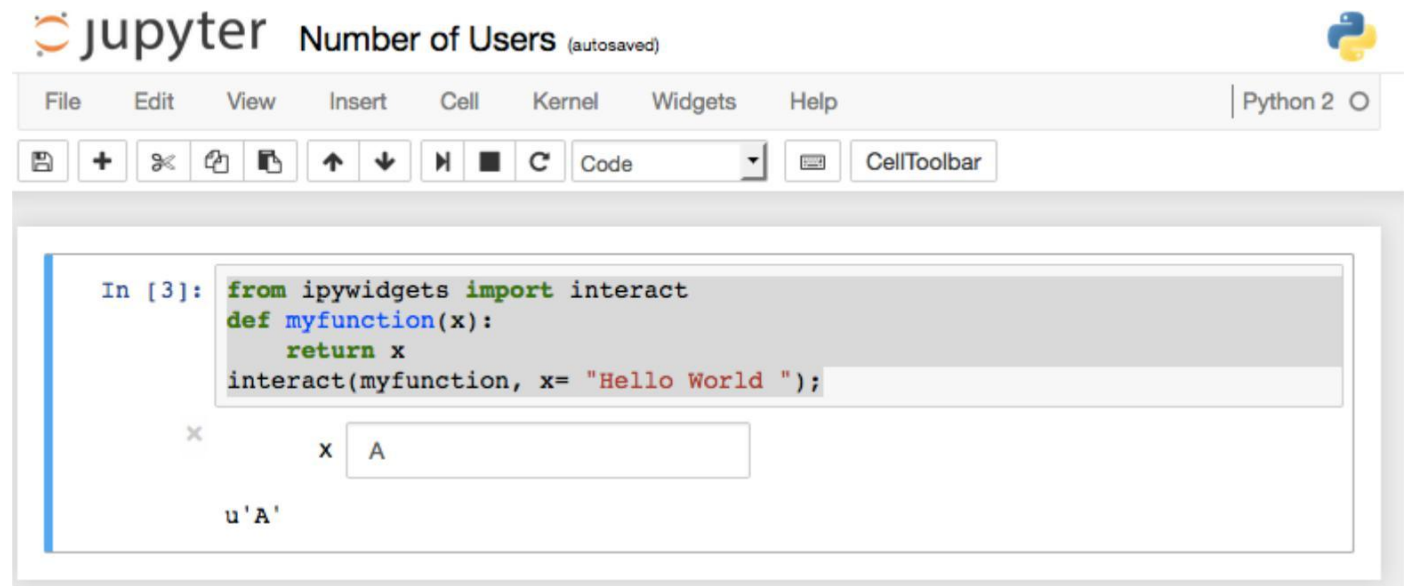
var arr = [98, 98.6, 98.4, 98.8, 200, 120, 98.5];

//standard deviation
var my_stddev = stats.stdev(arr).toFixed(2);

//mean
var my_mean = stats.mean(arr).toFixed(2);

//median
var my_median = stats.median(arr);
```

Chapter 8: Multiuser Jupyter Notebooks



The screenshot shows the Jupyter Notebook interface. At the top, the logo "jupyter" is followed by the text "Number of Users (autosaved)" and a Python logo. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". On the right side of the menu bar, it says "Python 2". Below the menu bar is a toolbar with icons for saving, adding, undo, redo, copy, paste, and other actions. The main area contains a code cell with the following code:

```
In [3]: from ipywidgets import interact
def myfunction(x):
    return x
interact(myfunction, x= "Hello World ");
```

Below the code, there is an interactive widget. It consists of a label "x" followed by a text input field containing the letter "A". Below the input field, the output is displayed as "u'A'".

jupyter

Sign in

Username:

Password:

Sign In

Select items to perform actions on them.

Upload New ↻

<input type="checkbox"/>	<input type="checkbox"/>	anaconda
<input type="checkbox"/>	<input type="checkbox"/>	AndroidStudioProjects

Stop My Server My Server

My Server

Home

192.168.99.100:8888/tree

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New ↻

Notebook list empty.

Upload New ↕ ↻

Text File

Folder

Terminal

Notebooks

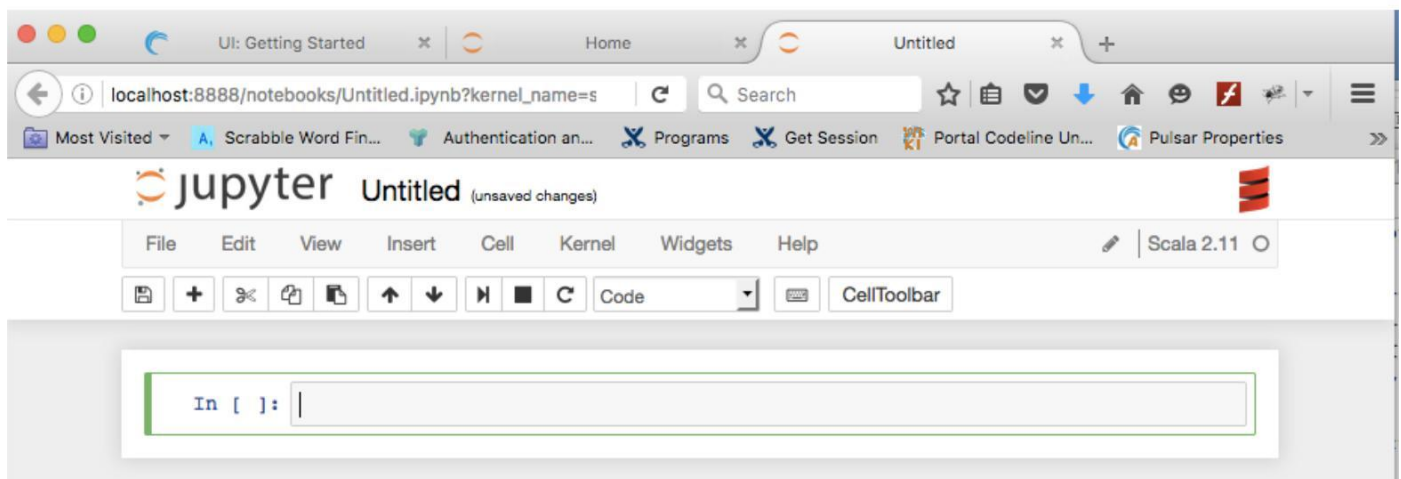
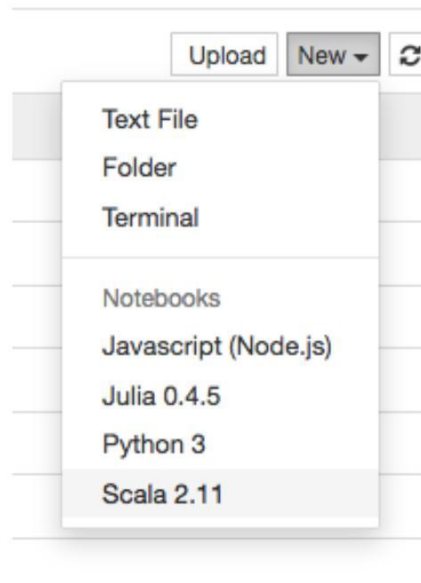
Apache Toree - Scala

Python 2

Python 3

R

Chapter 9: Jupyter Scala Published





```
In [3]: val name = "Dan"
        val age = 37
        show(name + " is " + age)

        "Dan is 37"

        name: String = "Dan"
        age: Int = 37
```



```
        total += x;
    }
    val mean:Double = total / count;

import scala.io.Source
filename: String = "iris.data"
array: collection.mutable.ArrayBuffer[Float] = ArrayBuffer(
  5.1F,
  4.9F,
  4.7F,
  4.6F,
  5.0F,
  5.4F,
  4.6F,
  5.0F,
  4.4F,
  4.9F,
  5.4F,
  4.8F,
  4.8F,
  4.3F,
  5.8F,
  5.7F,
  5.4F,
  5.1F,
  5.7F,
  ...
count: Int = 150
min: Double = 4.300000190734863
max: Double = 7.900000095367432
total: Double = 876.4999990463257
mean: Double = 5.843333326975505
```



```

var females_survived = 0
for (line <- Source.fromFile(filename).getLines) {
  var cols = line.split(",").map(_.trim);
  var sex = cols(5);
  if (sex == "male") {
    males = males + 1;
    if (cols(1).toInt == 1) {
      males_survived = males_survived + 1;
    }
  }
  if (sex == "female") {
    females = females + 1;
    if (cols(1).toInt == 1) {
      females_survived = females_survived + 1;
    }
  }
}
}
val mens_survival_rate = males_survived.toFloat/males.toFloat
val womens_survival_rate = females_survived.toFloat/females.toFloat

```

```

import scala.io.Source
filename: String = "train.csv"
males: Int = 577
females: Int = 314
males_survived: Int = 109
females_survived: Int = 233
mens_survival_rate: Float = 0.18890814F
womens_survival_rate: Float = 0.7420382F

```



```

}
val max = dice.reduceLeft(_ max _)
for( i <- 0 to 11) {
  var str = ""
  for( j <- 1 to dice(i)/3) {
    str = str + "X"
  }
  print(i+1, str, "\n")
}

(1,XXXXXXX,
) (2,XXXXXXXXXXXXXXXXXXXX,
) (3,XXXXXXXXXXXXXXXXXXXXXXXX,
) (4,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (5,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (6,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (7,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (8,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (9,XXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (10,XXXXXXXXXXXXXXXXXXXXXXXXXXXX,
) (11,XXXXXXXXXXXX,
) (12,,
)

r: util.Random = scala.util.Random@a5edf54
samples: Int = 1000
dice: Array[Int] = Array(23, 48, 76, 112, 148, 164, 135, 114, 78, 71, 31,
0)
max: Int = 164

```

In []:



```

In [1]: var factor = 7
val multiplier = (i:Int) => i * factor
val a = multiplier(11)
val b = multiplier(12)

factor: Int = 7
multiplier: Int => Int = <function1>
a: Int = 77
b: Int = 84

```



```
In [10]: def squared(x: Int): Int = x * x
def cubed(x: Int): Int = x * x * x
def process(a: Int, processor: Int => Int): Int = {processor(a) }
val fiveSquared = process(5, squared)
val sevenCubed = process(7, cubed)

defined function squared
defined function cubed
defined function process
fiveSquared: Int = 25
sevenCubed: Int = 343
```



```
In [1]: def matchTest(x: Any): Any = x match {
  case 7 => "seven"
  case "two" => 2
  case _ => "something"
}

val isItTwo = matchTest("two")
val isItTest = matchTest("test")
val isItSeven = matchTest(7)

defined function matchTest
isItTwo: Any = 2
isItTest: Any = something
isItSeven: Any = seven
```



```
In [4]: case class Car(brand: String, model: String)
        val buickLeSabre = Car("Buick", "LeSabre")

defined class Car
buickLeSabre: $user.Car = Car("Buick", "LeSabre")
```

```
In [5]: def carType(car: Car) = car match {
        case Car("Honda", "Accord") => "sedan"
        case Car("GM", "Denali") => "suv"
        case Car("Mercedes", "300") => "luxury"
        case Car("Buick", "LeSabre") => "sedan"
        case _ => "Car: is of unknown type"
        }

        val typeOfBuick = carType(buickLeSabre)

defined function carType
typeOfBuick: String = "sedan"
```



```
In [0]: def calculate (amount: Int): Int = { amount = amount + 1; return amount; }

        var balance = 100
        val result = calculate(balance);

Main.scala:23: reassignment to val
                def calculate (amount: Int): Int = { amount = amount + 1; r
return amount; } ; var balance = { () =>
```




```
In [2]: var mutableList = List(1, 2, 3);
        var immutableList = scala.collection.immutable.List(4, 5, 6);
        mutableList.updated(1,400);
        immutableList.updated(1,700);

mutableList: List[Int] = List(1, 2, 3)
immutableList: List[Int] = List(4, 5, 6)
res1_2: List[Int] = List(1, 400, 3)
res1_3: List[Int] = List(4, 700, 6)
```



```
In [1]: def divide(dividend:Int, divisor:Int): Float = { dividend.toFloat / divisor
        divide(40, 5)
        divide(divisor = 40, dividend = 5)

defined function divide
res0_1: Float = 8.0F
res0_2: Float = 0.125F
```



```
In [1]: trait Color {
        def isRed(): Boolean
      }

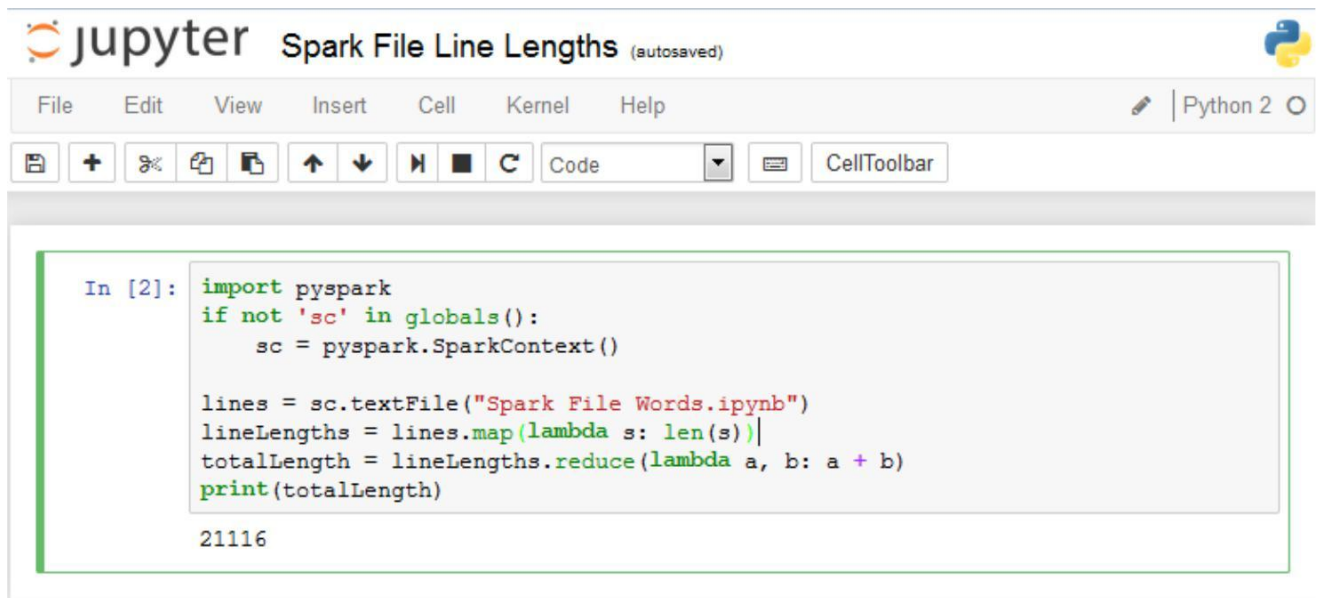
      class Red extends Color {
        def isRed() = true
      }

      class Blue extends Color {
        def isRed() = false
      }

      var red = new Red();
      var blue = new Blue();
      red.isRed()
      blue.isRed()

      defined trait Color
      defined class Red
      defined class Blue
      red: $user.Red = cmd0$$user$Red@1029c76f
      blue: $user.Blue = cmd0$$user$Blue@29bd267d
      res0_5: Boolean = true
      res0_6: Boolean = false
```

Chapter 10: Jupyter and Big Data

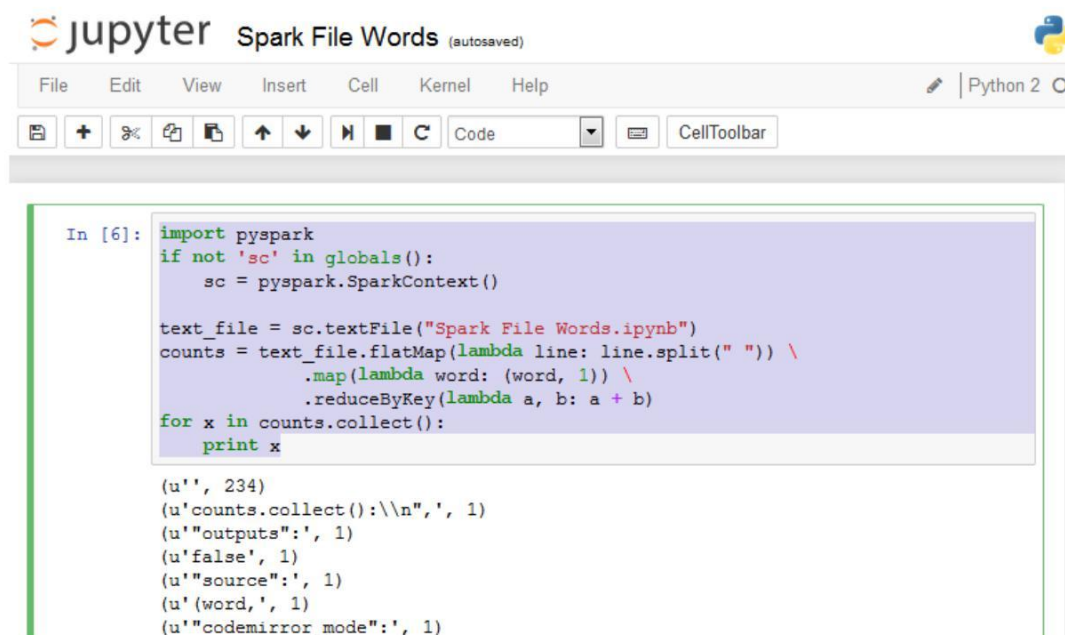


The screenshot shows a Jupyter Notebook interface with the title "Spark File Line Lengths (autosaved)". The notebook is running Python 2. The code cell contains the following Python code:

```
In [2]: import pyspark
if not 'sc' in globals():
    sc = pyspark.SparkContext()

lines = sc.textFile("Spark File Words.ipynb")
lineLengths = lines.map(lambda s: len(s))
totalLength = lineLengths.reduce(lambda a, b: a + b)
print(totalLength)
```

The output of the code is the integer 21116.



The screenshot shows a Jupyter Notebook interface with the title "Spark File Words (autosaved)". The notebook is running Python 2. The code cell contains the following Python code:

```
In [6]: import pyspark
if not 'sc' in globals():
    sc = pyspark.SparkContext()

text_file = sc.textFile("Spark File Words.ipynb")
counts = text_file.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a + b)
for x in counts.collect():
    print x
```

The output of the code is a list of tuples representing word counts:

```
(u'', 234)
(u'counts.collect():\n', 1)
(u'"outputs":', 1)
(u'false', 1)
(u'"source":', 1)
(u'(word,', 1)
(u'"codemirror_mode":', 1)
```



```
In [2]: import pyspark
if not 'sc' in globals():
    sc = pyspark.SparkContext()

text_file = sc.textFile("Spark File Words.ipynb")
sorted_counts = text_file.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a + b) \
    .sortByKey()
for x in sorted_counts.collect():
    print x

(u'', 779)
(u'', 4)
(u\"(u\\'',', 1)
(u\"(u\\'(<ipython-input-5-78889f28c230>\\'',', 1)
(u\"(u\\'(word,\\'',', 1)
```



```
In [8]: import pyspark
import random

if not 'sc' in globals():
    sc = pyspark.SparkContext()

NUM_SAMPLES = 1000

def sample(p):
    x,y = random.random(),random.random()
    return 1 if x*x + y*y < 1 else 0

count = sc.parallelize(xrange(0, NUM_SAMPLES)) \
    .map(sample) \
    .reduce(lambda a, b: a + b)

print "Pi is roughly %f" % (4.0 * count / NUM_SAMPLES)

Pi is roughly 3.208000
```



Code CellToolbar

```
In [12]: import pyspark
if not 'sc' in globals():
    sc = pyspark.SparkContext()

textFile = sc.textFile("access_log")
print(textFile.count(), "access records")

gets = textFile.filter(lambda line: "GET" in line)
print(gets.count(), "GETs")

posts = textFile.filter(lambda line: "POST" in line)
print(posts.count(), "POSTs")

other = textFile.subtract(gets).subtract(posts)
print(other.count(), "Other")
for x in other.collect():
    print x

(1546, 'access records')
(1525, 'GETs')
(14, 'POSTs')
(7, 'Other')
64.246.94.152 - - [08/Mar/2004:20:09:57 -0800] "HEAD /twiki/bin/view/Main/S
pamAssassinDeleting HTTP/1.1" 200 0
```



```

number = abs(int(number))

# simple tests
if number < 2:
    return False

# 2 is prime
if number == 2:
    return True
# other even numbers aren't
if not number & 1:
    return False

# check whether number is divisible into it's square root
for x in range(3, int(number**0.5)+1, 2):
    if number % x == 0:
        return False

#if we get this far we are good
return True

# create a set of numbers to 100,000
numbers = sc.parallelize(xrange(100000))

# count out the number of primes we found
print numbers.filter(is_it_prime).count()

```

9592



```
In [11]: import pyspark

if not 'sc' in globals():
    sc = pyspark.SparkContext()

sentences = sc.textFile('2600raid.txt') \
    .glom() \
    .map(lambda x: " ".join(x)) \
    .flatMap(lambda x: x.split(" "))
print(sentences.count(), "sentences")

bigrams = sentences.map(lambda x:x.split()) \
    .flatMap(lambda x: [(x[i],x[i+1]),1] for i in range(0,len(x)-1)])
print(bigrams.count(), "bigrams")

frequent_bigrams = bigrams.reduceByKey(lambda x,y:x+y) \
    .map(lambda x:(x[1],x[0])) \
    .sortByKey(False)
frequent_bigrams.take(10)
```

```
(220, 'sentences')
(3463, 'bigrams')
```

```
Out[11]: [(36, (u'of', u'the')),
(15, (u'the', u'mall')),
(12, (u'At', u'this')),
(12, (u'on', u'the')),
(12, (u'to', u'the')),
(11, (u'the', u'guards')),
```

```
In [32]: import pyspark
import csv
import operator
import itertools
import collections

if not 'sc' in globals():
    sc = pyspark.SparkContext()

years = {}
occupations = {}
guests = {}

#YEAR,GoogleKnowlege_Occupation,Show,Group,Raw_Guest_List
with open('daily_show_guests.csv', 'rb') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        year = row['YEAR']
        if years.has_key(year):
            years[year] = years[year] + 1
        else:
            years[year] = 1

        occupation = row['GoogleKnowlege_Occupation']
        if occupations.has_key(occupation):
            occupations[occupation] = occupations[occupation] + 1
        else:
            occupations[occupation] = 1
```



```
    guest = row['Raw_Guest_List']
    if guests.has_key(guest):
        guests[guest] = guests[guest] + 1
    else:
        guests[guest] = 1

syears = sorted(years.items(), key=operator.itemgetter(1), reverse=True)
soccupations = sorted(occupations.items(), key=operator.itemgetter(1), reverse=True)
sguests = sorted(guests.items(), key=operator.itemgetter(1), reverse=True)

print syears[:5]
print soccupations[:5]
print sguests[:5]
```

```
[('2000', 169), ('1999', 166), ('2003', 166), ('2013', 166), ('2010', 165)]
[('actor', 596), ('actress', 271), ('journalist', 180), ('author', 102), ('Journalist', 72)]
[('Fareed Zakaria', 19), ('Denis Leary', 17), ('Brian Williams', 16), ('Paul Rudd', 13), ('Ricky Gervais', 13)]
```