Class03 - Plotting

Grant Wilson

Phys281

Some info and graphics borrowed from
http://pong.tamu.edu/~kthyng/presentations/visualization.pdf
And http://www-personal.umich.edu/~jpboyd/sciviz_1_graphbadly.pdf

Answers to Exercises

```
import numpy as np


#E2.1
print "E2.1 - Matrix Multiplication"
x = np.array([[1,4,2],[3,6,8],[5,8,9]])
y = np.array([1,4,7])
print “matrix x times vector y = “, x.dot(y)
```

Answers to Exercises

```
import numpy as np

#E2.2
print "E2.2 - Vector Math"
t = np.array([0.,2.,4.,6.,8.,10.])

#the loopy way
y1 = np.array([0.]*len(t))    #or … y1 = np.zeros(len(t)) or y1 = np.empty(len(t))
for i in range(len(t)):
    y1[i] = 4.*t[i]**3 - 3.*t[i]**2 + 7*t[i]

#the vector way
y2 = 4.*t**3 - 3.*t**2 + 7*t

print "y1 = ", y1
print "y2 = ", y2
print "y1-y2 = ", y1-y2
```

Answers to Exercises

```
import numpy as np

#E2.3
print "E2.3 - now using a function"
def funcy(A,B,C):
    t = np.array([0,2,4,6,8,10])
    return A*t**3 + B*t**2 + C*t

#get the user input for A, B, and C
A = float(raw_input("Enter A: "))
B = float(raw_input("Enter B: "))
C = float(raw_input("Enter C: "))
print funcy(A,B,C)
```

Answers to Exercises

```
import numpy as np

#E2.4
print "E2.4 - simple statistics"
x = np.array([ 0.23596479, 0.0269803 , 2.01538686, 1.21755484, -0.18797097,0.13972061,
1.53857571, -1.09488231, 1.07353811, 0.26210775,-0.20792946, -2.10692386, -0.58577368, -
0.07815834, -0.67514258,-0.32785578, 1.2330222 , 0.69836858, -0.26743941, -0.21449487,-
0.90811154, 1.89334483, -0.91871939, 0.6861157 , 0.38594592,0.17821075, -1.54133892, -
0.75497937, 0.35047012, 0.33632286,-0.5326393 , 0.50398025, -0.75764786, 0.62867976,
1.91655106,-0.80132509, -0.9499839 , -0.76363298, -0.99494678, -0.97367112])

print "mean(x) = ", x.mean()
print "median(x) = ", np.median(x)
print "stddev(x) = ", x.std()
```

Class03 - Plotting

Grant Wilson

Phys281

Some info and graphics borrowed from
http://pong.tamu.edu/~kthyng/presentations/visualization.pdf
And http://www-personal.umich.edu/~jpboyd/sciviz_1_graphbadly.pdf

“The aim of good data graphics is to display data accurately and clearly”

—– H. Wainer (1997), pg. 12.

“The greatest value of a picture is when it forces us to notice what we never expected to see.”

—– John Tukey, quoted in (1997), pg. 47.

Goals of Plotting Data

Goals of Plotting Data

- Be honest
- Be clear
- Convey useful information
- Show trends
- Be efficient
- Show ALL data
- KISS

The Fundamental Rule about Graphing

- Before you show it to someone, have something intelligent to say about it.

The Fundamental Rule about Graphing

- Before you show it to someone, have something intelligent to say about it.

<u>False assumptions by some students</u>

- I graph, therefore I think.
- I graph, therefore I work.
- I graph, therefore I progress in understanding

The Fundamental Rule about Graphing

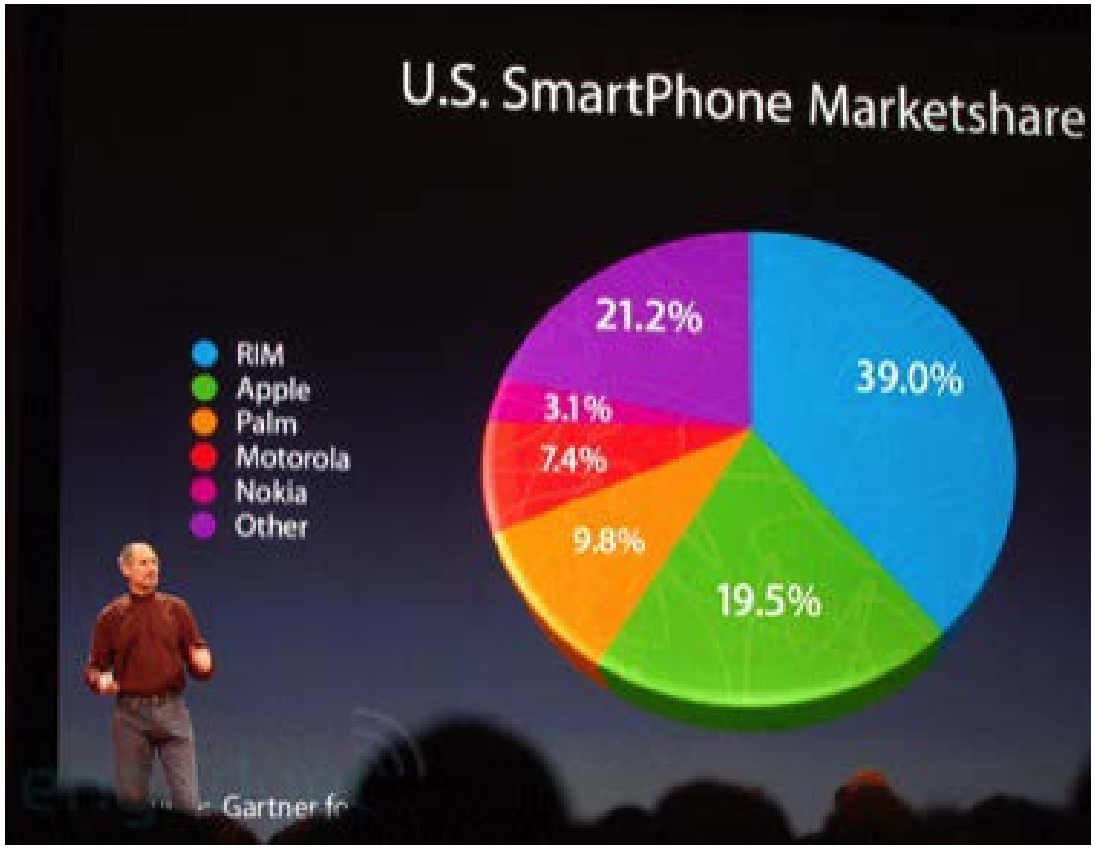
- Before you show it to someone, have something intelligent to say about it.

<u>False assumptions by some students</u>

- I graph, therefore I think.
- I graph, therefore I work.
- I graph, therefore I progress in understanding

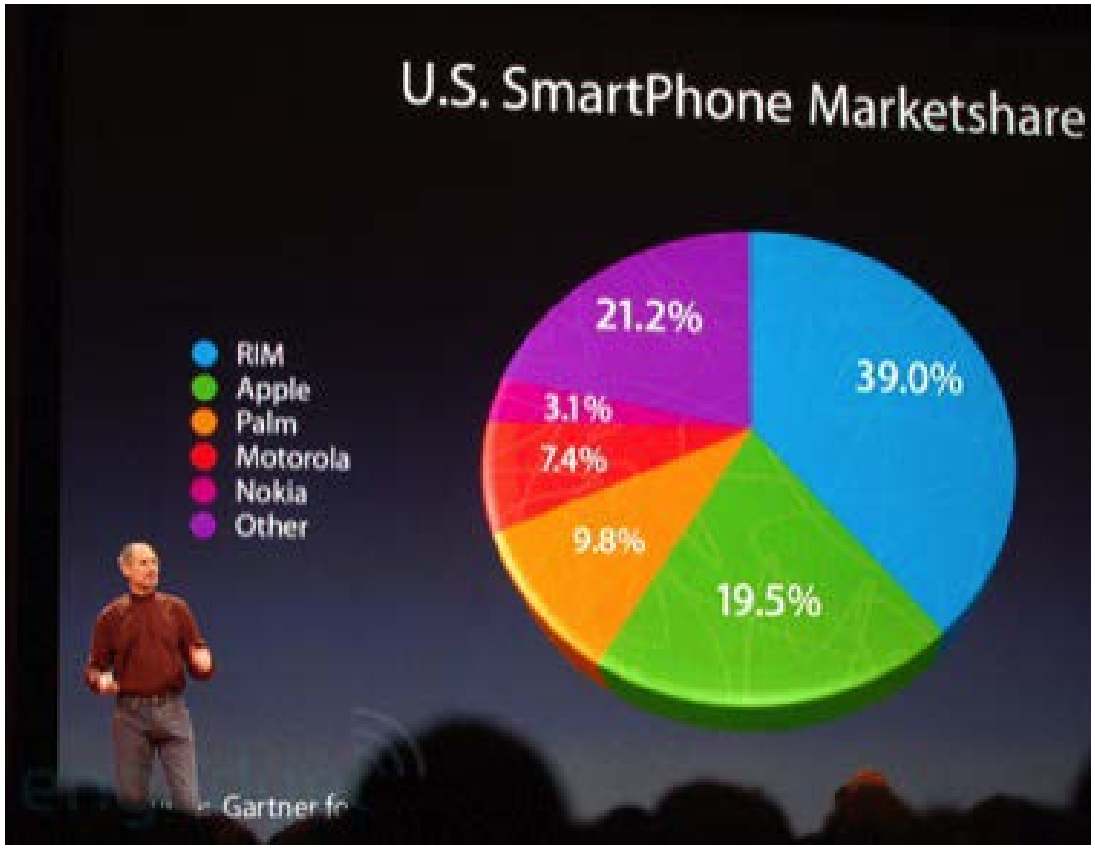
An important lesson: Once you graph it, you own it, and you will be judged by how well you understand it.

What's wrong with this plot?



http://www.engadget.com/2008/01/15/live-from-macworld-2008-steve-jobs-keynote/

What's wrong with this plot?



1) What information does the 3-d-ness of the chart convey?

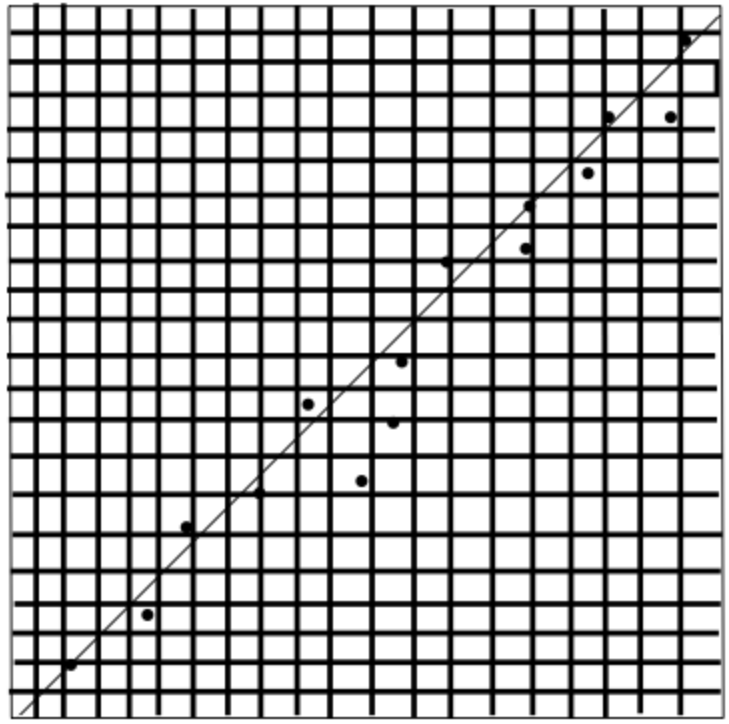
2) What information do the textures convey?

3) Humans are very poor at judging areas so pie-chart doesn't convey more info than the numbers themselves.

Upshot: Don't use Pie-charts for scientific data presentation.

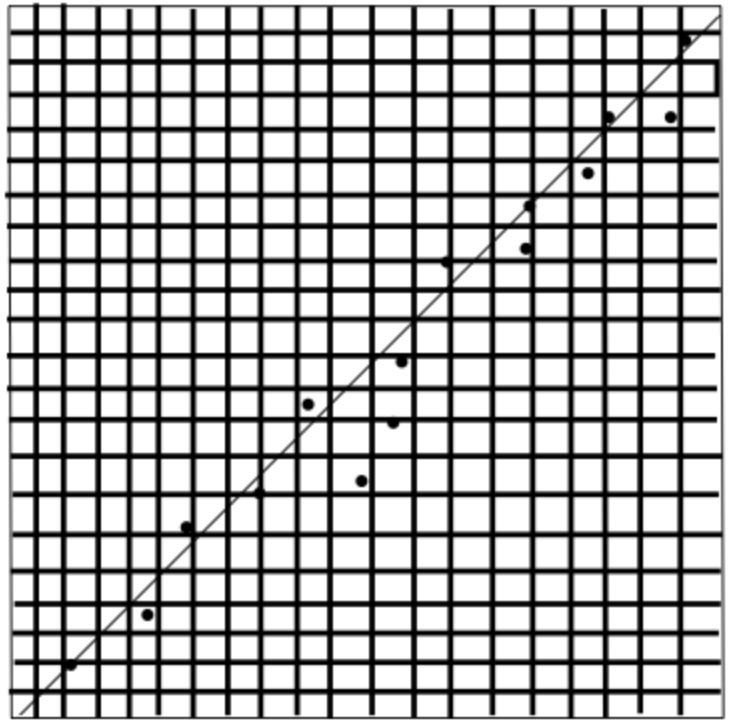
http://www.engadget.com/2008/01/15/live-from-macworld-2008-steve-jobs-keynote/

What's wrong with this plot?



http://www-personal.umich.edu/ jpboyd/sciviz 1 graphbadly.pdf

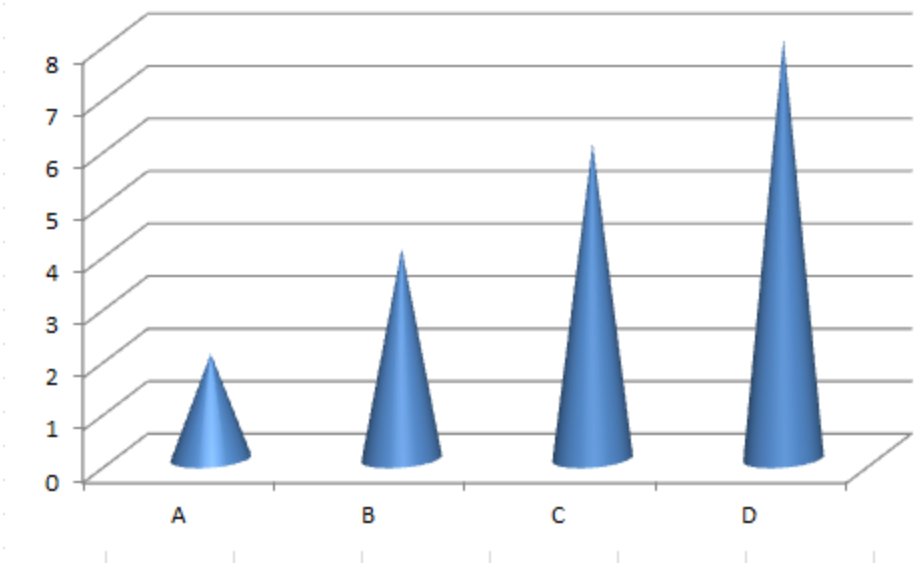
What's wrong with this plot?



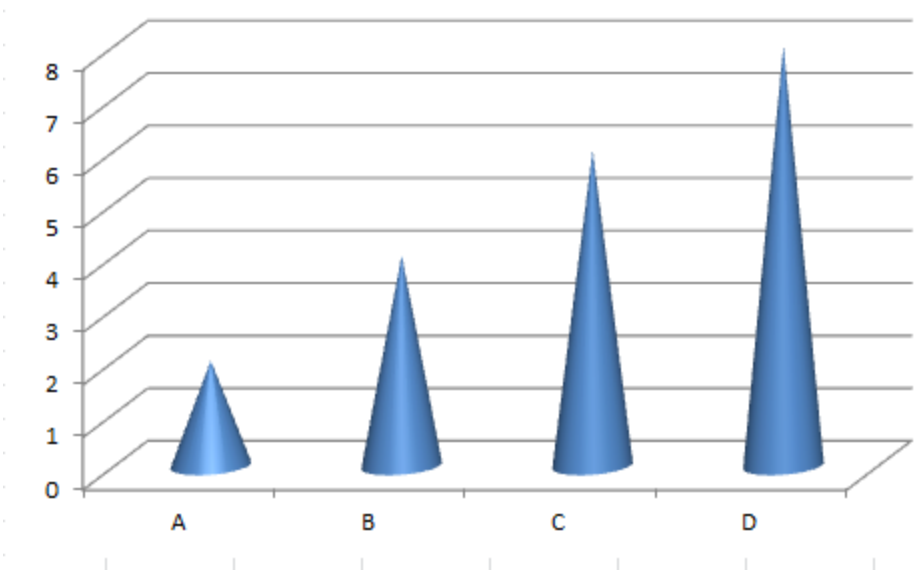
1) No axis labels.
2) Grid obfuscates data.
3) No plot title.
4) No error bars
   (but perhaps none are needed)
5) No legend.

http://www-personal.umich.edu/ jpboyd/sciviz 1 graphbadly.pdf

What's wrong with this plot?

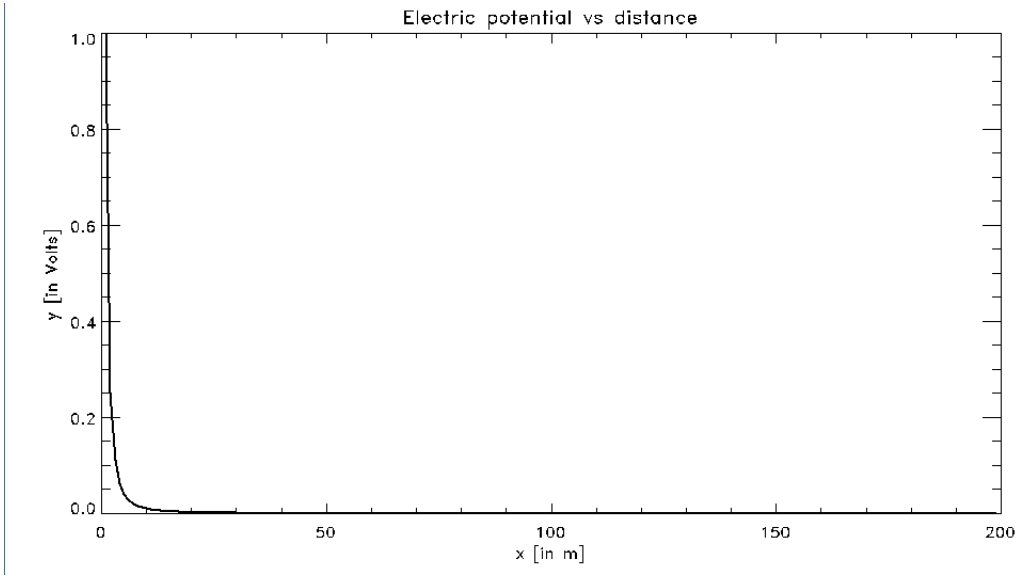


What's wrong with this plot?

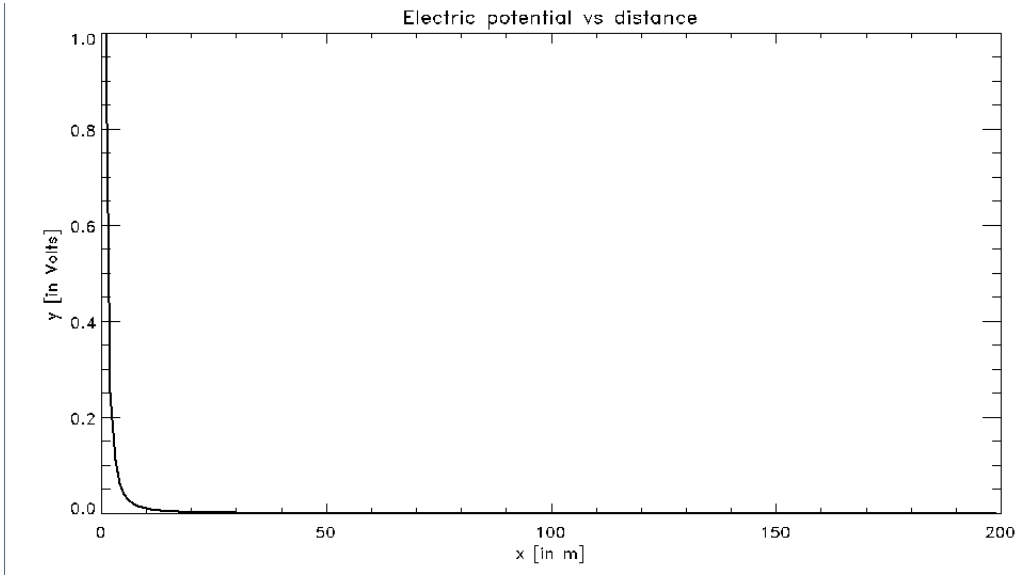


1) No axis labels.
2) No plot title.
3) No legend.
4) Is 3-d really necessary?
5) What's up with them cones?

What's wrong with this plot?



What's wrong with this plot?

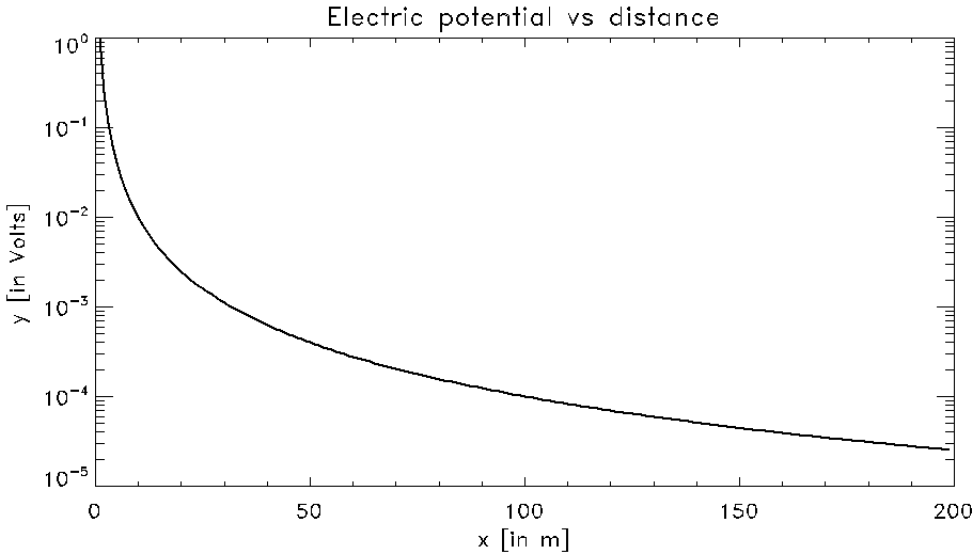


1) Most of the plot area is unused.

2) The graph doesn't clearly show what's going on at x>10.

3) Oh, and the labels are hard to read.

… instead …



1) If you have a huge range, use a <u>log axis</u>.

2) Make your labels ridiculously large. Even these are still too small.

Setting up Plotting with ipython

- In order to plot interactively we need to start ipython with the –matplotlib qt directive

  ipython --matplotlib qt

- Then, once inside ipython, we need to import the plotting libraries
  - *from matplotlib import pyplot as plt*

Setting up Plotting with ipython

- In order to plot interactively we need to start ipython with the –matplotlib qt directive

  ipython --matplotlib qt

Starting ipython this way allows our plotting to be interactive.

- Then, once inside ipython, we need to import the plotting libraries
  - *from matplotlib import pyplot as plt*

Read this as “from the matplotlib package, import the module pyplot but let me call it ‘plt’ to save typing.

From this point on, EVERYTHING associated with the plotting will start with plt.
(note this is very different from the tutorial you looked at)

Why not use –pylab?

- What is pylab?
  - “pylab provides a procedural interface to the matplotlib object-oriented plotting library. It is modeled closely after Matlab(TM). Therefore, the majority of plotting commands in pylab have Matlab(TM) analogs with similar arguments. Important commands are explained with interactive examples.”

- Sounds good on the surface, but let’s look at what it does:
  - *import numpy*
  - *import matplotlib*
  - *from matplotlib import pylab, mlab, pyplot*
  - *np = numpy*
  - *plt = pyplot*
  - *from IPython.core.pylabtools import figsize, getfigs*
  - *from pylab import **
  - *from numpy import **

Why not use –pylab?

- Pylab pollutes the namespace by importing over 950 items
- Starting Pylab is irreversible
- Pylab replaces built-in functions
- Some programs may work in ipython but not in python

Bottom line about --pylab

- Don’t use it. Be in control of your computing environment.

Our first plot

ipython --matplotlib qt

- *from matplotlib import pyplot as plt*
- *plt.plot([0,2,4,6,8])*

Our first plot

ipython --matplotlib qt

- *from matplotlib import pyplot as plt*
- *plt.plot([0,2,4,6,8])*
- *plt.plot([0,2,4,6,8], 'o')*

Our first plot

ipython --matplotlib qt

- *from matplotlib import pyplot as plt*
- *plt.plot([0,2,4,6,8])*
- *plt.plot([0,2,4,6,8], 'o')*

- Now try just
  - *plot([0,2,4,6,8], 'o')*

- To close the plot window
  - *plt.close()*

Trying out the Matplotlib tutorial

- Together, let's try the following

  ipython --matplotlib qt

  - *from matplotlib import pyplot as plt*
  - *import numpy as np*
  - *X = np.linspace(-np.pi, np.pi, 256, endpoint=True)*
  - *C = np.cos(X)*
  - *S = np.sin(X)*
  - *plt.plot(X,C)*
  - *plt.plot(X,S)*

Trying out the Matplotlib tutorial

- Together, let's try the following

  ipython --matplotlib qt
  - *from matplotlib import pyplot as plt*
  - *import numpy as np*
  - *X = np.linspace(-np.pi, np.pi, 256)*
  - *C = np.cos(X)*
  - *S = np.sin(X)*
  - *plt.plot(X,C)*
  - *plt.plot(X,S)*

- Now, type that same sequence into a text file called "simple_plot.py" and run it from a new ipython instance.

Explore plotting on your own

- Run through the tutorial at http://www.loria.fr/~rougier/teaching/matplotlib/ producing each successive plot shown.

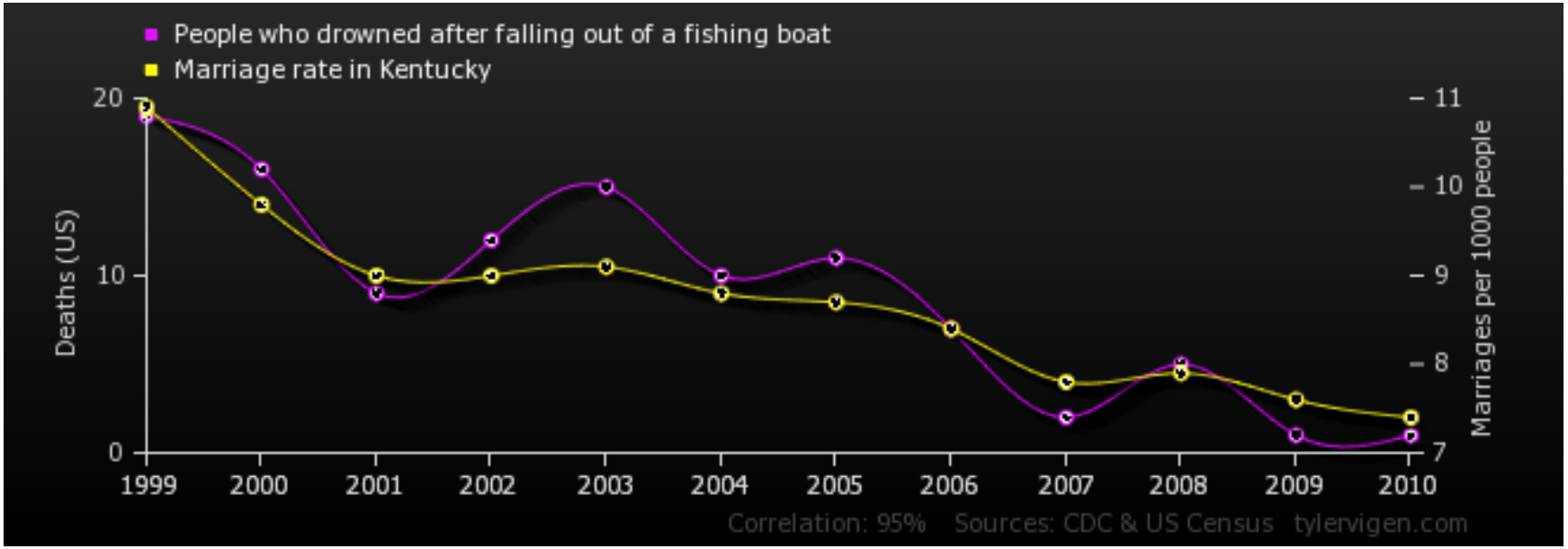
- Show either me or one of the TAs after you complete each plot.

Cannon Ball Game

- Download cannon_ball.py from the class Moodle page.
- Try playing the game and get a feel for what happens.
- Read and understand the code in the game.
- Now, add some plots to make the game more visually interesting.

Exercise

- Reproduce the following plot from http://www.tylervigen.com/.



dates: 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010

US fishing boat drownings: 19, 16, 9, 12, 15, 10, 11, 7, 2, 5, 1, 1

Marriage rate in Kentucky (per 1000 people): 10.9, 9.8, 9, 9, 9.1, 8.8, 8.7, 8.4, 7.8, 7.9, 7.6,7.4