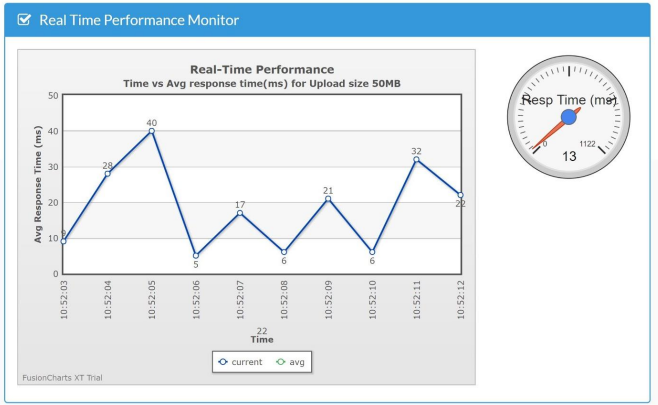
CAL STATE LA
CALIFORNIA STATE UNIVERSITY, LOS ANGELES

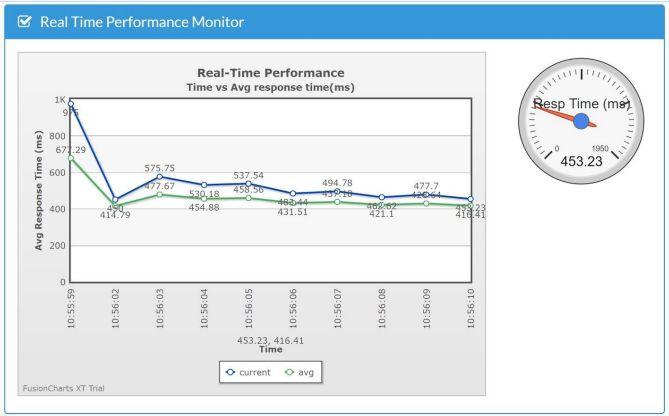
NASA Direct-STEM Linear Regression Modeling

Isaac Cano
David San
Davis Louie
Faculty Advisor: Dr. Jiang Guo

Our Project

- Performance of Evaluation for Cloud Service
  - Web services are software that utilize access to the internet
  - Analyzes how well a cloud service performs
    - Performance
    - Reliability
    - Availability
    - Scalability
- Capable performance testing between multiple domains:
  - Compares the response times between each domain
  - domain1.com vs domain2.com vs ... domainN.com





What is Pinging?

- To send an echo-request packet to (an IP address) and use the echo reply to determine whether another computer on the network is operational and the speed at which the data is being transferred
- In simpler terms, Pinging means to check the connection between two networks
- The goal is to predict how long it takes to ping to a specific server and receive a reply
    - The time is generally given at the end in the measurement of milliseconds

How to Collect and Record Ping Data

- Through a command line:
  - ping youtube.com
  - Add parameter options:
    - -c, -w, -i etc.

```
imcano@DESKTOP-8NHEKFE:/mnt/c/Windows/System32$ ping youtube.com -c 10 -i 0.5
PING youtube.com (172.217.11.174) 56(84) bytes of data.
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=1 ttl=54 time=13.9 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=2 ttl=54 time=13.8 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=3 ttl=54 time=13.9 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=4 ttl=54 time=13.4 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=5 ttl=54 time=14.8 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=6 ttl=54 time=13.8 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=7 ttl=54 time=15.0 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=8 ttl=54 time=14.7 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=9 ttl=54 time=15.1 ms
64 bytes from lax28s15-in-f14.1e100.net (172.217.11.174): icmp_seq=10 ttl=54 time=14.5 ms

--- youtube.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 4507ms
rtt min/avg/max/mdev = 13.466/14.334/15.109/0.562 ms
```

- Now, what do we need to record?
  - Pen and paper
  - Patience
  - Make sure to have coffee! (very important)

How can we make this easier?

- We can automate it
  - Write a python script
    - Run the command
    - Save the information to text file
    - Organize the information
- What can we do with all this data?
  - Graph
  - Analyze
  - Predict

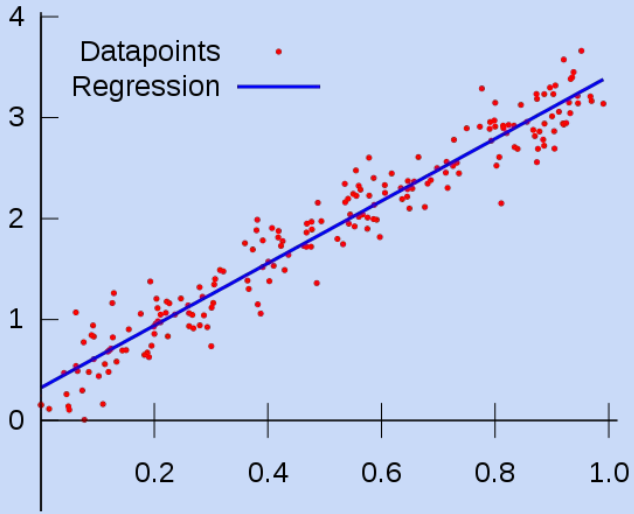
ping.py

```python
import argparse
import csv
import subprocess
import sys


parser = argparse.ArgumentParser()
parser.add_argument('domain', help='name or IP of the website', type=str)
parser.add_argument('-p','--parameters', help='https://www.tutorialspoint.com/unix_commands/ping.htm')
args = parser.parse_args()

# list will contain each statement that builds the cmd
usr_cmd = []
# list contains ['ping', 'somedomainname']
usr_cmd.extend(('ping', args.domain))

# when params are passed in add them to the array
if args.parameters:
    usr_cmd.extend(args.parameters.split(' '))
# when no params are included add -c 3
# pings 3 times
# fail safe, prevents from infinitely pinging
else:
    param = '-c'
    count = '3'
    usr_cmd.extend((param, count))

# By default -D is added
# -D includes a timestamp
time = '-D'
usr_cmd.append(time)

# prints out user command
print(usr_cmd)

# will store parsed data
unix_time = []
resp_time = []

# Saves the ping data to raw_data.txt
with open('raw_data.log', 'w') as out:
    # Runs users request through the terminal
    result = subprocess.Popen(usr_cmd, stdout=subprocess.PIPE)

    for line in iter(result.stdout.readline, b''):
        # convert line to a string
        str_line = line.decode('utf-8')
        # displays to terminal
        sys.stdout.write(str_line)
        # writes to file
        out.write(str_line)
```

What is Linear Regression

- Linear Regression is a modelled relationship between an independent variable (x-axis) and a dependent variable (y-axis)
- The linear regression is determined to be a best fit line based on data points from a graph
- Generally this formula can follow the form:

$$Y = mx + b \text{ (slope-intercept form!)}$$

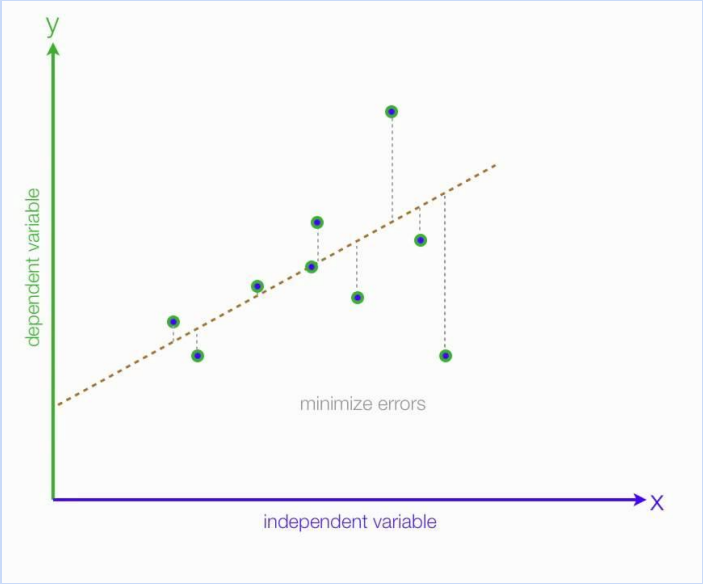


How is it Useful?

- Linear regression can be used to predict the next outcome with respect to the independent variable
- We can then test the accuracy of these results by comparing them to its real value
    - Example: Ping for youtube.com at 1:30pm on Friday
        - The model predicted the ping was 150ms
        - The actual ping was 15ms

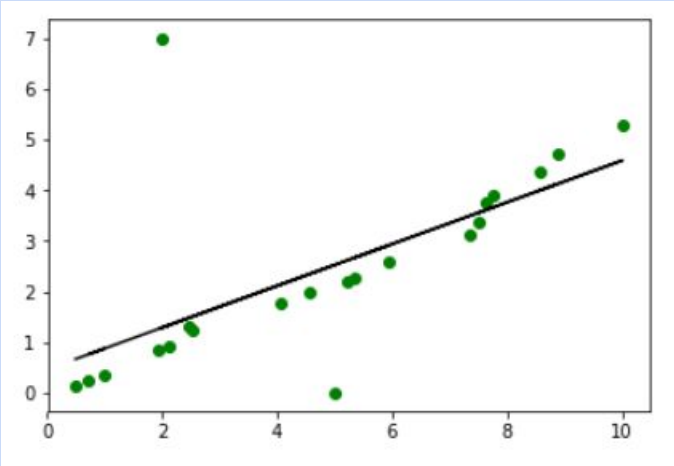
Making a Linear Regression Graph

- What you need:
  - A Computer!
  - A Computer Science Major named Isaac
  - Ubuntu Bash (mandatory with Isaac's implementation)
  - Anaconda, Excel, or Google Sheets
  - Python (version 3.6+)
  - Data (CSV file)
  - Jupyter Notebook (optional)
  - Visual Studio Code (optional)



Linear Regression Python Example

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt

from sklearn.linear_model import LinearRegression

# Reading data from file location (\\ is the escape character for \)
df = pd.read_csv('C:\\Users\\TallGuy\\DIRECTSTEM\\Summer_2018-PTWS\\candy_bar_prices.csv')

# display the head (first five rows) from the data
df.head()

# LinearRegression will expect an array of shape (n, 1) for the "Training data"
X = df['Candy Bar Weight in Grams'].values[:,np.newaxis]

# "Target data" is array of shape (n,)
y = df['Candy Bar Prices in USD'].values

# Setting the Linear Regression model to a variable and fitting the data
lr_model = LinearRegression()
lr_model.fit(X, y)

# ploting the points into the scatter plot
plt.scatter(X, y,color='g')
# ploting the best line of fit for the scatter plot
plt.plot(X, lr_model.predict(X),color='k')

# displaying the plot graph
plt.show()
```



Things to Consider

- Ping different sites and graph a linear regression line for each for visual

  www.yahoo.com, www.ibm.com, www.google.com,
  www.intel.com, www.amd.com, www.ge.com,
  www.cisco.com, www.twitter.com, www.facebook.com

- Calculate the average ping time for each site
- Obtain different ping results throughout different times of the day
  - Similarly, obtain different ping results for different days
- How often should you ping for reliable data?
- When should we use Logistic Regression?

Thank you & Questions