DSCI 554 LAB 1

TOOLING, WEB DEV, LINE AND PIE CHARTS WITH GOOGLE SHEETS AND GOOGLE CHARTS.

Dr. Luciano Nocera

USC Viterbi
School of Engineering
Integrated Media Systems Center

🧰 TOOLING: CHROME, NODE, VS CODE

INSTALL GOOGLE CHROME

Download and install from https://www.google.com/chrome

🚨 Make Chrome the default browser for assignments & labs

INSTALL NODE

Download and install node LTS from https://nodejs.org

INSTALL VISUAL STUDIO CODE

Download and install from https://code.visualstudio.com

Install `code`:

```
View > Command Palette... > Shell command: Install 'code' command in PATH
```

💡 Learn most used keyboard shortcuts: Windows, macOS, Linux

🧰 TOOLING: GIT

CREATE A GITHUB ACCOUNT WITH YOUR USC USERNAME!

🚨 Assignments from non USC accounts will not be graded!

INSTALL GIT

- Mac (in terminal install xcode dev): `xcode-select --install`
- Windows: download and install from https://git-scm.com

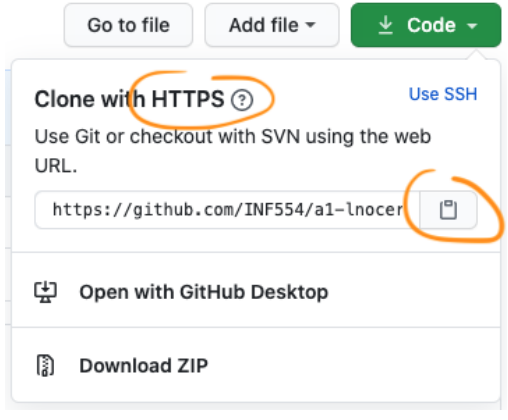
INSTALL GITHUB DESKTOP

Download and install from https://desktop.github.com

💡 In Windows use GitBash to run shell commands or the terminal in VS Code

GETTING STARTED WITH ASSIGNMENTS

1. Accept the assignment from the course home-page or Blackboard under Content > Week 1
2. Get assignment GitHub link:



3. Clone repository:
   a. GitHub Desktop (preferred): `File → Clone Repository`
   b. VS Code: `Git Clone` command
   c. Command line (create a personal access token first):

```
$ mkdir hw
$ cd hw
$ git clone https://github.com/DSCI554/a1-lnocera-1.git
```

4. Open the folder in VS Code:

```
$ cd a1-lnocera-1
$ code -n .
```

STARTER CODE TOUR

```
ASSIGNMENT.md  #assignment rubric in markdown
README.md  #repository README in markdown
index.html
lab/  #lab files
  ex1.html   #lab exercise file
  ex2.html
  simple-exe/
    faithful.csv
    node_modules
    package-lock.json
    package.json
    simple-exe
  simple-html/
    index.html
    node_modules
    package-lock.json
    package.json
  simple-vue/
    README.md
    babel.config.js
    node_modules
    package-lock.json
    package.json
    public
    src
    vue.config.js
  style.css
node_modules  #node modules created when running npm install
package-lock.json
package.json  #package.json file
```

🔨 SIMPLE-EXE: NODE COMMAND LINE INTERFACE (CLI)

Node.js (JavaScript runtime environment)

```
$ node
> console.log('hello JavaScript')
> a = [1, 'b', 'c', 2]
> a.concat(a)
```

This is how to generate files for a CLI executable using npm (JavaScript package manager):

If you cut-and-paste the commands below do not copy the comments starting with #

```
$ npm init
$ npm install commander csv-parse #fetch libs in node_modules and adds references in package.json
$ cat > mycli  #create mycli executable
#!/usr/bin/env node
console.log('hello mycli');
$ chmod +x mycli
$ ./mycli
$ code -n .  #open current folder in VS Code
```

💡 Debug with `console.log()` and VS Code debugger (launch.json file)

🔨 SIMPLE-HTML: WEB PAGE

💡 Debug in Chrome with DevTools

💡 In Windows use GitBash to execute or VS Code terminal

To debug in Chrome:

1. Open index.html in Chrome
2. Open DevTools: `View → Developer → Javascript Console`

DevTools shortcuts:

Action	Mac	Windows / Linux
Open whatever panel you used last	`Command+Option+I`	`F12` or `Control+Shift+I`
Open the **Console** panel	`Command+Option+J`	`Control+Shift+J`
Open the **Elements** panel	`Command+Shift+C` or `Command+Option+C`	`Control+Shift+C`

🔨 SIMPLE-VUE: VUE.JS WEB APP

Run the Vue.js (JavaScript framework) app in VS Code:

```
$ cd simple-vue
$ code -n .  #open simple-vue project
$ npm install #only when updating package.json with new packages
$ npm run serve #use vue cli (see in package.json)
```

💡 Debug with console.log() or Vue DevTools extension and Chrome

To create a new Vue.js app:

```
$ npm install -g @vue/cli  #install vue CLI globally
$ vue --version
$ vue create my-app  #choose default vue 2!
```

To debug in Chrome update the devtool property inside vue.config.js:

```
$ cat > vue.config.js
module.exports = {
  configureWebpack: {
    devtool: 'source-map'
  }
}
```

GRAPHING WITH GOOGLE SHEETS

1. Read the documentation:
   - Types of charts & graphs in Google Sheets
   - Add & edit a chart or graph
   - Line charts
   - Pie charts
   - Edit your chart's axes
   - Create & use pivot tables
2. Format the data
3. Generate the graph

LAB EXERCISES

🏋🏾 EX1: LINE CHART IN GOOGLE SHEETS USING CCSE COVID-19 DATASET

- Follow the instructions in `ex1.html`

🏋🏾 EX2: IMPLEMENT A GOOGLE CHARTS PIE CHART

- Follow the instructions in `ex2.html`

💡 Implemented and test one step at the time then **commit & push**!

🚨 Make sure to **commit & push** by the deadline (15% of grade)

DSCI 554: plotting with Google Sheets

REFERENCE SLIDES

GIT BASICS

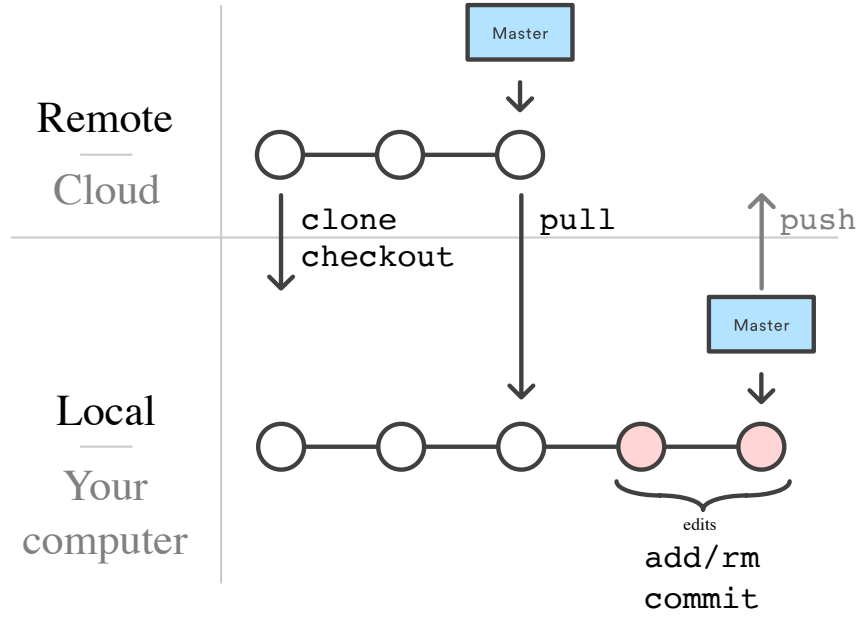
- You will use git to store and version your code
- git snapshots are tracked using a SHA-1 hash
- The git *repository* is located in `.git/`

EXAMPLE SHA-1 HASH

`ca412e3a7968b81b247fa7ae40c2fd5bf7fb3308`

COMMANDS TO KNOW

- **clone:** fetch copy of remote
- **checkout:** create working copy
- **add/rm:** mark files to add
- **commit:** save changes
- **pull:** fetch changes from remote
- **push:** upload changes to remote



MARKDOWN BASICS

- Use Markdown to document your work
- Markdown files have .md extension, e.g., README.md
- Visual Studio Code let's you preview the Markdown
- README.md is rendered as HTML by GitHub

MARKDOWN

```
# Title
## Subtitle
__HTML__ `inline` code:

```html
<h1 style="color: red">Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>
```

A list:

- [hyperlink text](https://guides.github.com/)
- list item
- list item
```

RENDERING OF MARKDOWN

Title

Subtitle

HTML `inline` code:

```
<h1 style="color: red">Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>
```

A list:

- hyperlink text
- list item
- list item

HTML BASICS

💡 Only the contents of the body tag are visible!

💡 Spaces, tabs and newlines outside of html elements are not rendered!

BODY

```
<h1>Header</h1>

<h2>Sub-header</h2>

<p>paragraph with text</p>     and some more text
directly in the body
<p>
  Some text with a
  <a href="some_url">hyperlink</a>
</p>
```

RENDERING OF BODY

HEADER

SUB-HEADER

paragraph with text

and some more text directly in the body

Some text with a hyperlink

UN DATA FOR THE ASSIGNMENT

A1 is modeled after Cairo's example from this week reading so it might help if you read that first.

The goal is to create a toy dataset we can use for the rest of the class.

Find a UNData dataset to use from the Datamarts page: