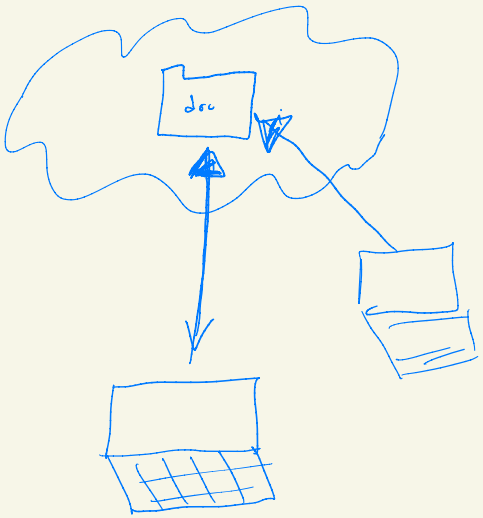
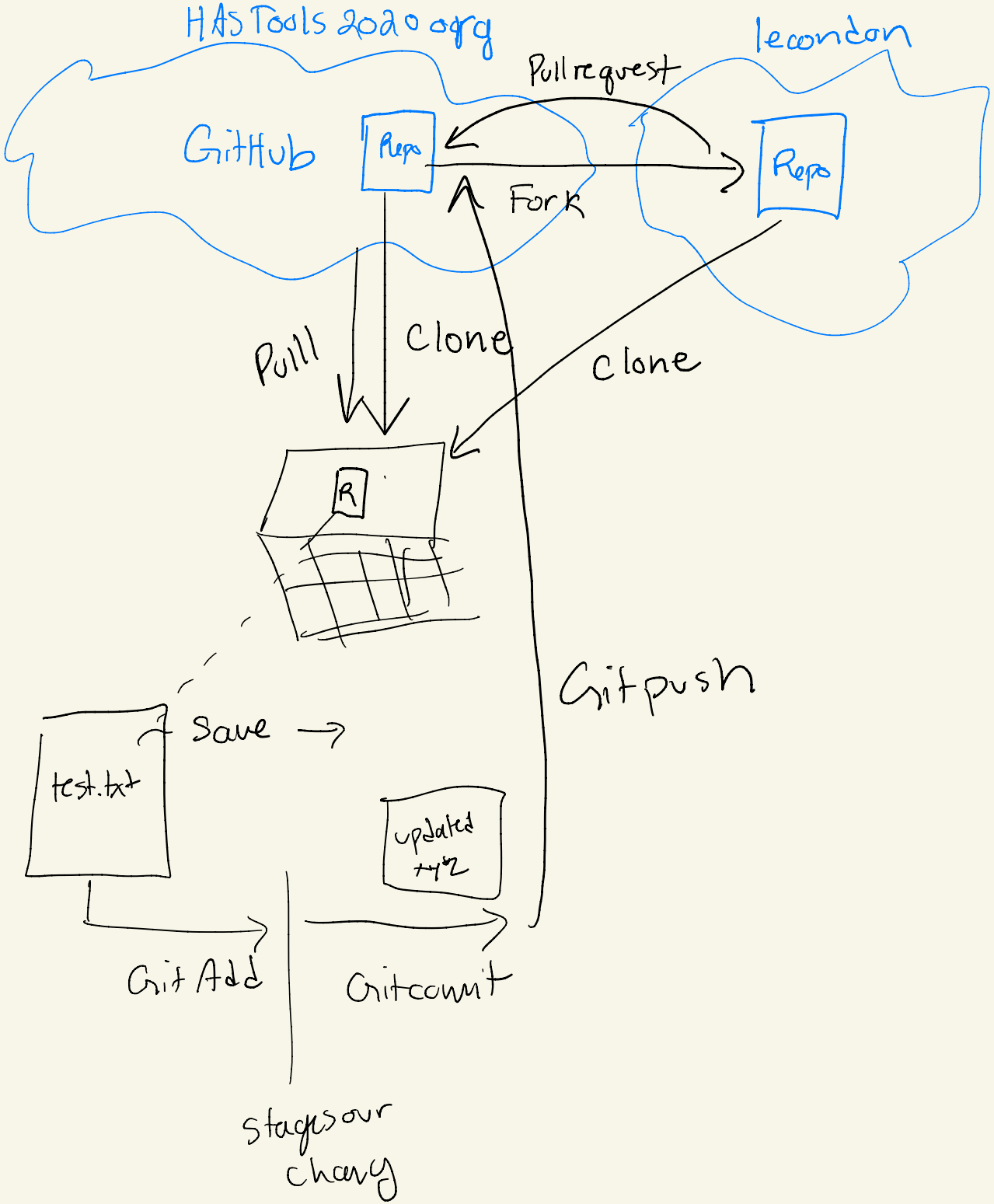
HAS Tools Notes

Fall 2020

Google docs / DropBox / Box





Definitions from week 1

9-1-20

Terminal - Shell - **GitBash**: place we give command directly to the computer (↙ command line)

- language of computer (Bash shell ∈ google bash commands)

GUI - Graphical User Interface: Places where we point & click

Atom: Text editor (notepad, text pad ∈ other text editors). It's the Best
Github integration, designed for coding, lots of handy add ons
Super popular & widely used

Git: Version Control ⇒ tracking changes & making it possible for people to work together

Git Repo ⇒ directory of files with addition tracking features added (hidden .git files)

GitHub: Cloud server for git repos (github.com)
pull, push, clone ⇒ how we interact between local (on our computer) and remote repo (on the cloud)

GitHub Desktop: GUI for managing github repos, lets you do things like clone, pull, push

GitKraken: GUI for managing github repos - more powerful than Github desktop

Week 1 pitfalls

Common pitfalls:

1) Putting a repo inside a repo: Don't do it! if you do them make a fresh clone

2) Terminal - git -commit and forget the "-m"
   ↳ control c to quit the operation in your terminal
   ↳ less prone to error if you commit through Atom or Git Kraken

3) Dont do clone on one drive: do it locally instead

4) No spaces in directory names

9-10-20

- Environments, packages & modules

Environment: A working "space" environment where you setup all of your software dependencies (engine) => lets you have multiple versions of thing ∈ software & packages & swap between them

Packages: A set of tools, A directory of python files with an "__init__" file
when we install a package we get this locally

Modules: A file of python code => define functions ~~& etc~~ classes & variables

Step 1: Activate environment
Step 2: Install package (only has to happen 1 time in your environment)
Step 3: Top of python script load package

Lists:
- Lists are 1D arrays [x, x, x, x,]
- not data type dependent (ints, floats, strings, booleans) ← but must be the same with list
- use [ ]
- lists in lists [[ ] [ ]]
- indexing start at 0
  - ↳ indexing/slicing => pull out parts of our dat structure
- * list comprehension
- Data structure
- Has its own method = eg append

* function in python ( ) — function x (arg1, arg2 - - -)
method in python — object.method ( )

Control flow - Conditionals, for loops, while loops
- conditional statements: if else else if
  - if something is true then do something
  - if it is not true then do something else
  - if the first thing is not true but some some second thing is true then do - - -

- Can as many nested conditionals as you want

has to be something with a T/F answer

- if <statement0> :
    <statement>
    <s 2>
    <s 3>

indentation matters!

} only happens so is true.

```
x = 5
if x == 5 :
    print("Here")

print("done")
```

<statement 4> ← outside my if statement

For loops:

- Repeats some action given a range of values
- Can have as many nested as you want

index

range of values that i will take

For i in <range> :
    <Statement 1>
    <S 2>
    <S 3>

indented

} in our for loop

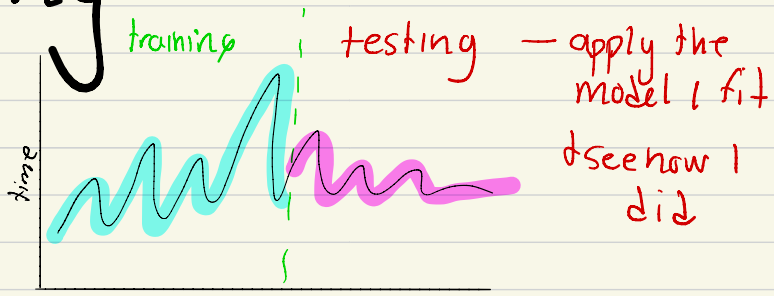
```
for i in range10 :
    print(i)

print(done)
```

[s 4)

Autoregressive Modeling

$y = f(x)$

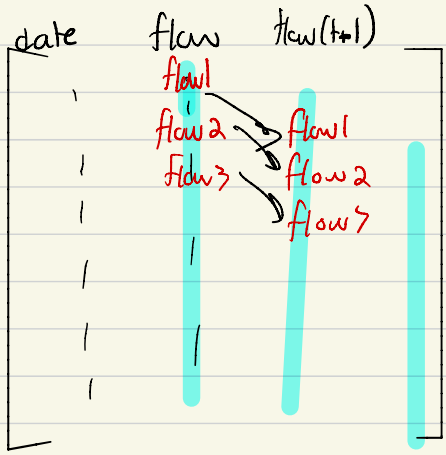
$x(t)$ $x(t-n)$

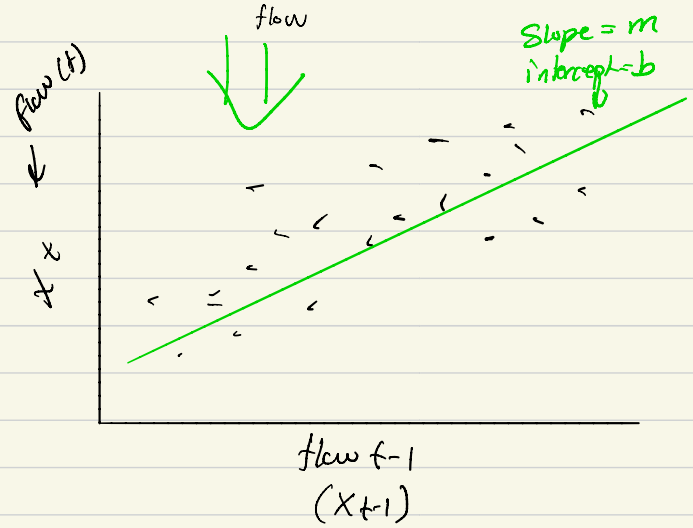


— apply the model & fit & see how I did

$X_t = mX_{t-1} + b$

slope intercept





↙ Can add in additional timesteps

$X_t = m(t-1) + n(t-2) + \cdots\cdots + b$

$X_t = m(t-3) + b$