The Unix Shell

Elizabeth Everman

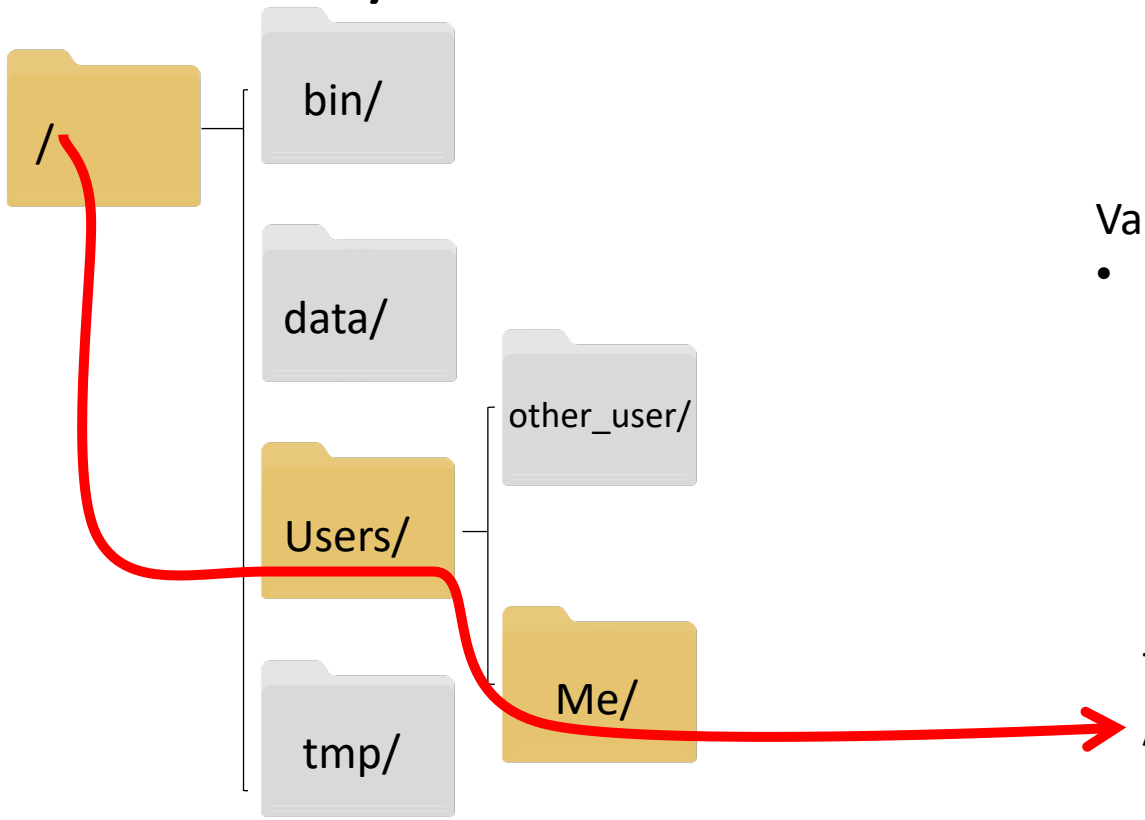
Email: e.everman@ku.edu

GitHub: https://github.com/ereverman/July-August_SWCarpentry_Day1

Clarification on terminology:

- Files = any object with an extension (e.g. .txt, .ppt, .docx)
- Directories = a location where files are stored
- File system = the operating system responsible for managing files and directories

Directory Structure:



Variation in home directory structure:

- Paths will look different on different operating systems:
  - Mac, Linux: /home/user
  - PC: C:\Users\

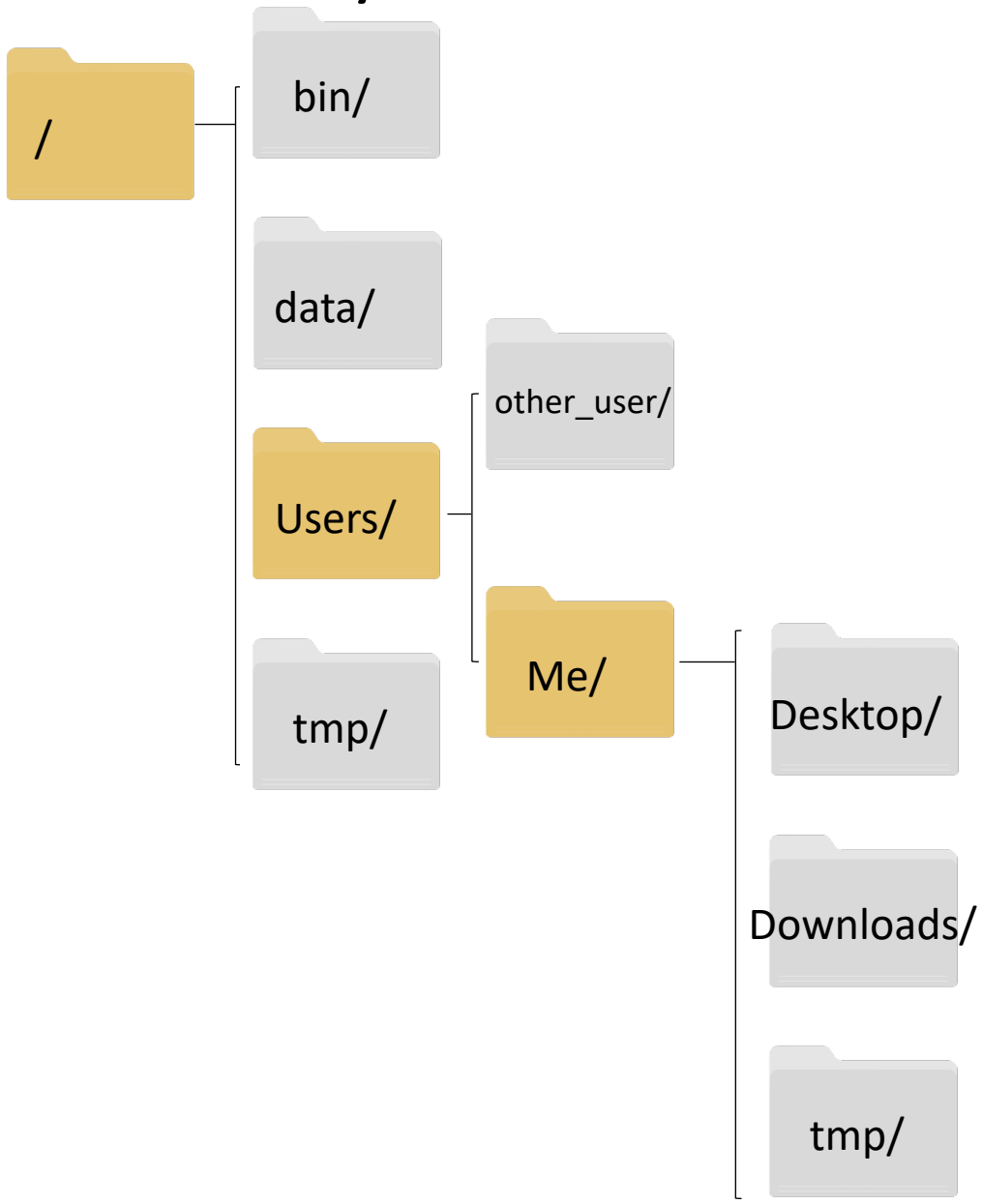
The Path:
/Users/Me/

Project Management

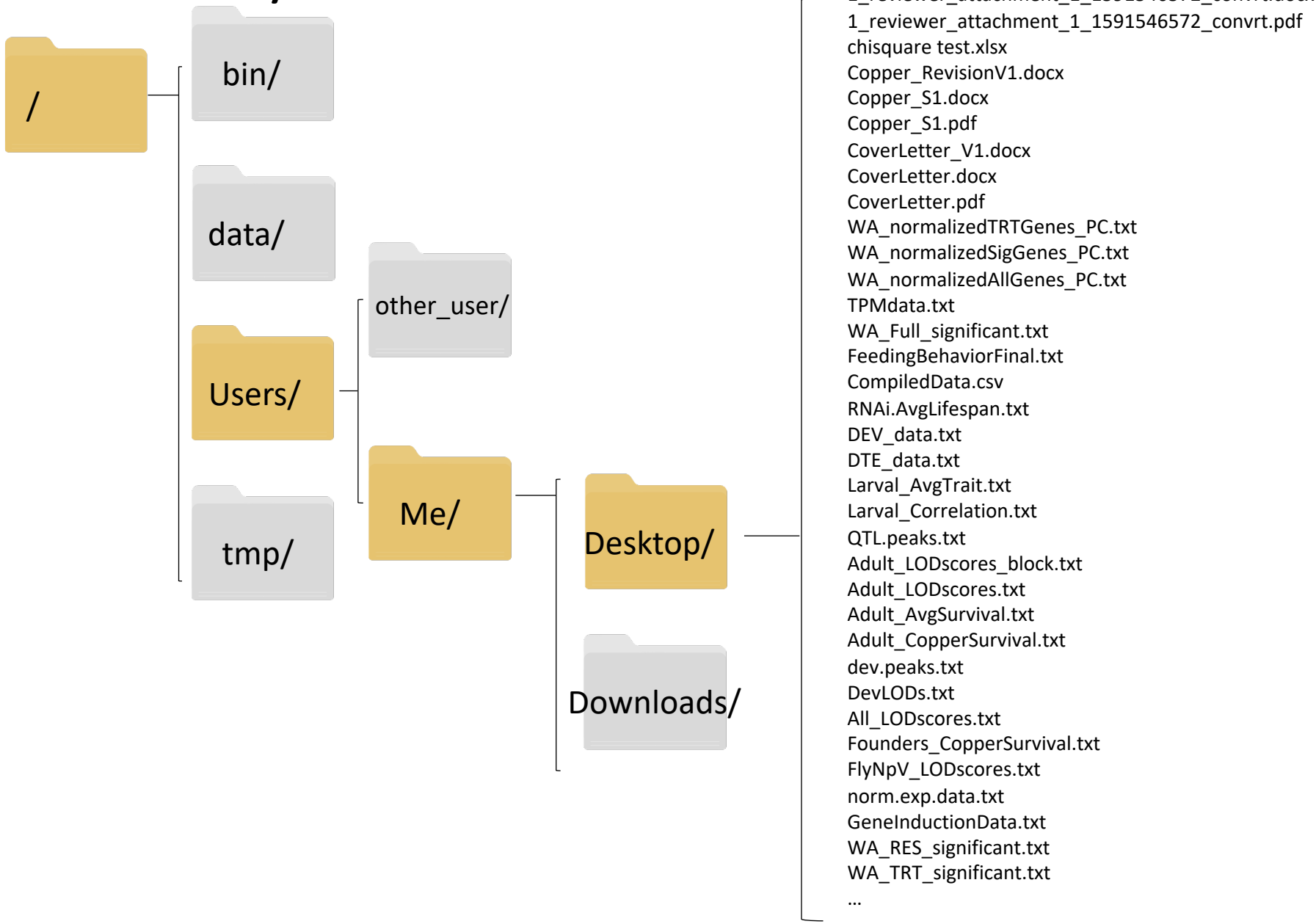
- Why? How do you organize your life (I mean data)?
  - Always treat data as “read-only” (and have a backup)
  - Multiple input files are often required
  - Multiple types of output files are generated

  - Stay organized
    - Keep directory names simple
    - Figure out a directory structure that works for your data analysis patterns.
    - Don’t bury our output files without a predictable path.

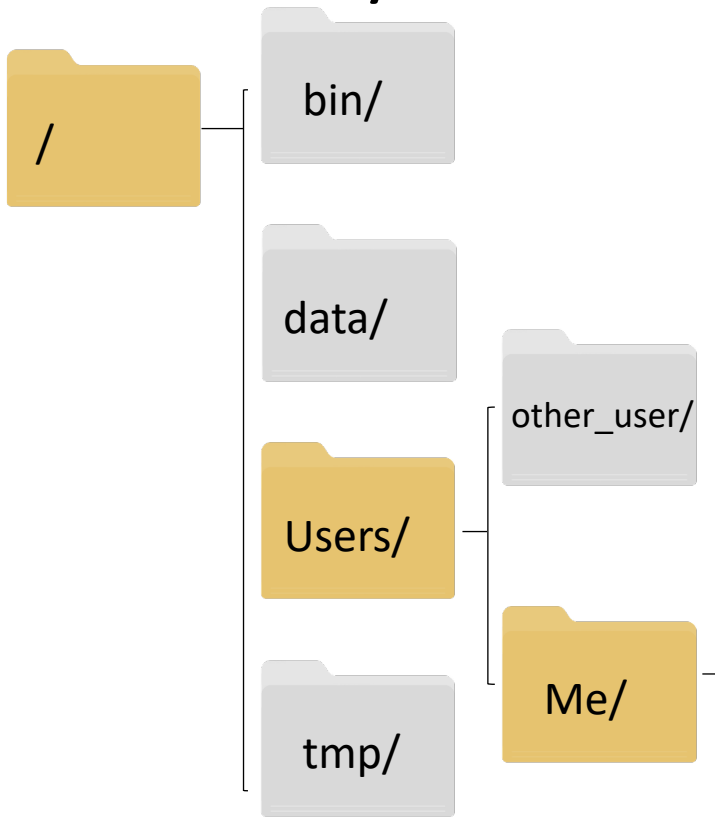
Directory Structure:



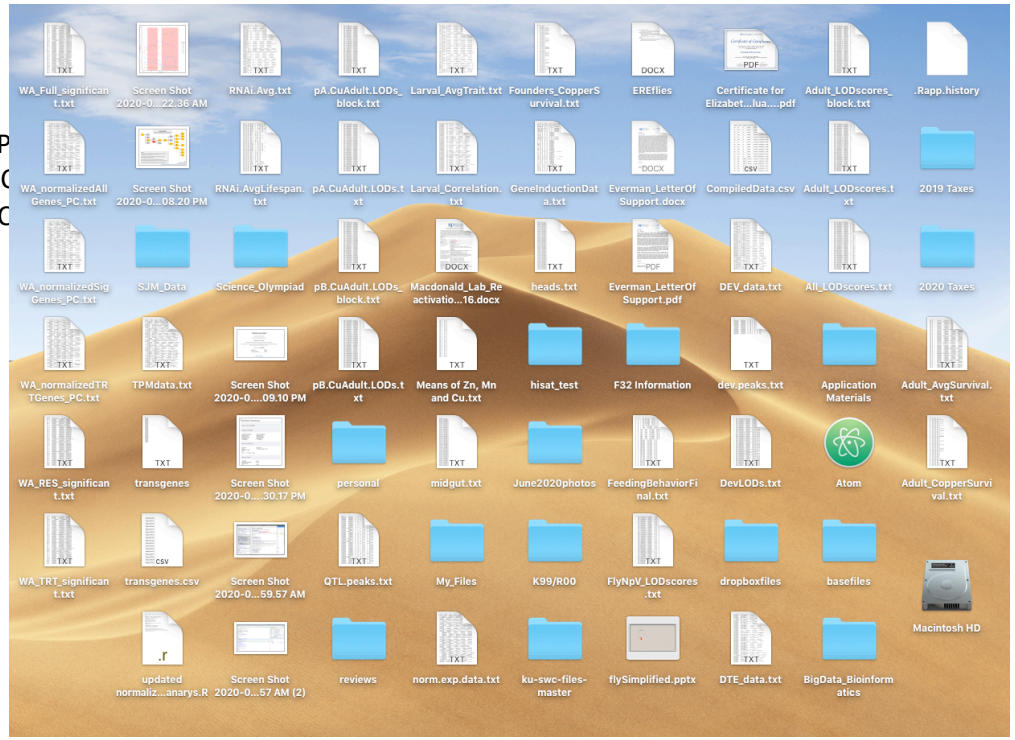
Directory Structure:



Directory Structure:

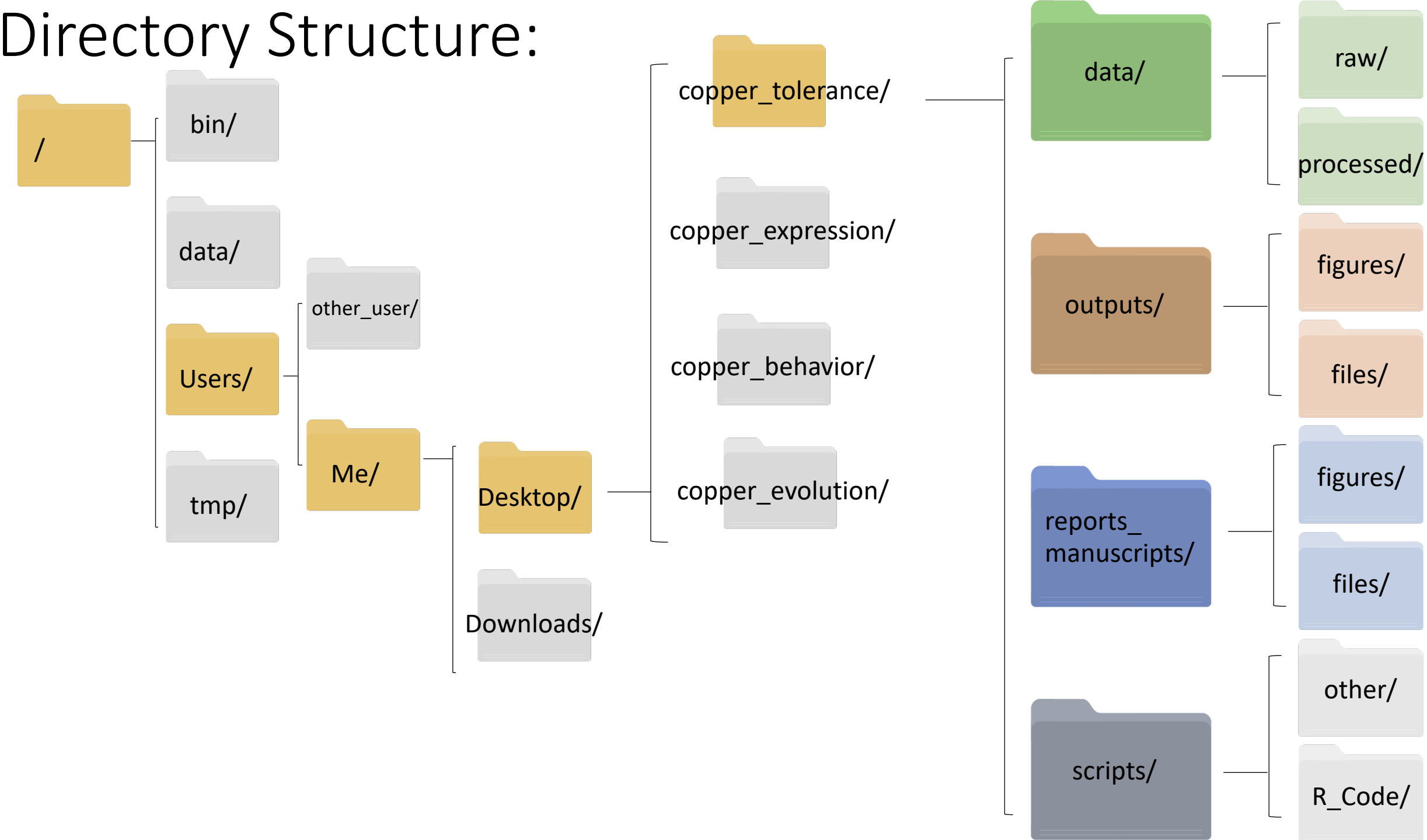


1_reviewer_attachment_1_1591546572_convrt.docx
1_reviewer_attachment_1_1591546572_convrt.pdf
chisquare test.xlsx
Copper_RevisionV1.docx
Copper_S1.docx
Copper_S1.pdf
CoverLetter_V1.docx
CoverLetter.docx
CoverLetter.pdf
WA_normalizedTRTGenes_P
WA_normalizedSigGenes_PC
WA_normalizedAllGenes_PC
TPMdata.txt
WA_Full_significant.txt
FeedingBehaviorFinal.txt
CompiledData.csv
RNAi.AvgLifespan.txt
DEV_data.txt
DTE_data.txt
Larval_AvgTrait.txt
Larval_Correlation.txt
QTL.peaks.txt
Adult_LODscores_block.txt
Adult_LODscores.txt
Adult_AvgSurvival.txt
Adult_CopperSurvival.txt
dev.peaks.txt
DevLODs.txt
All_LODscores.txt
Founders_CopperSurvival.txt
FlyNpV_LODscores.txt
norm.exp.data.txt
GeneInductionData.txt
WA_RES_significant.txt
WA_TRT_significant.txt
…
…

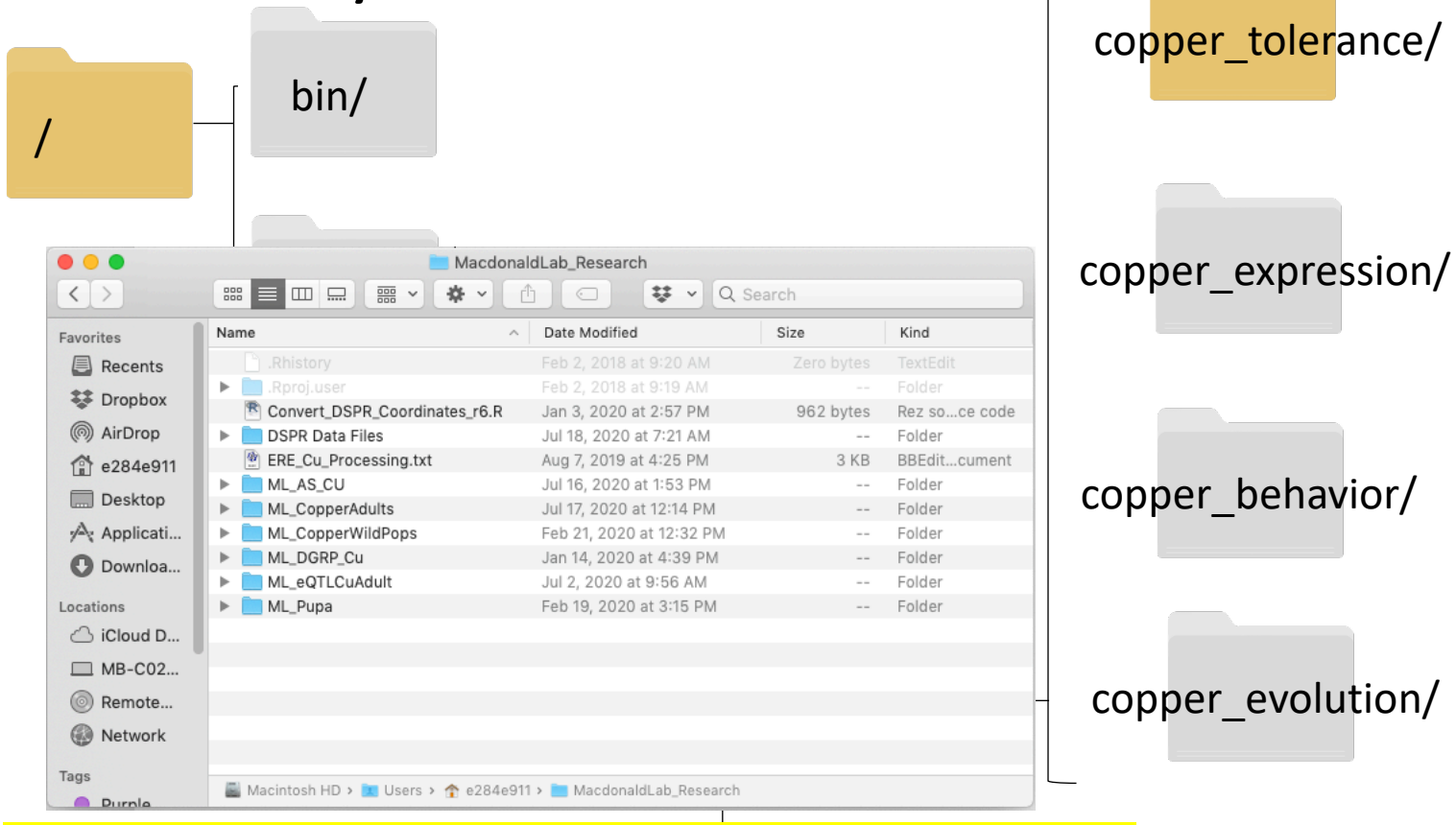


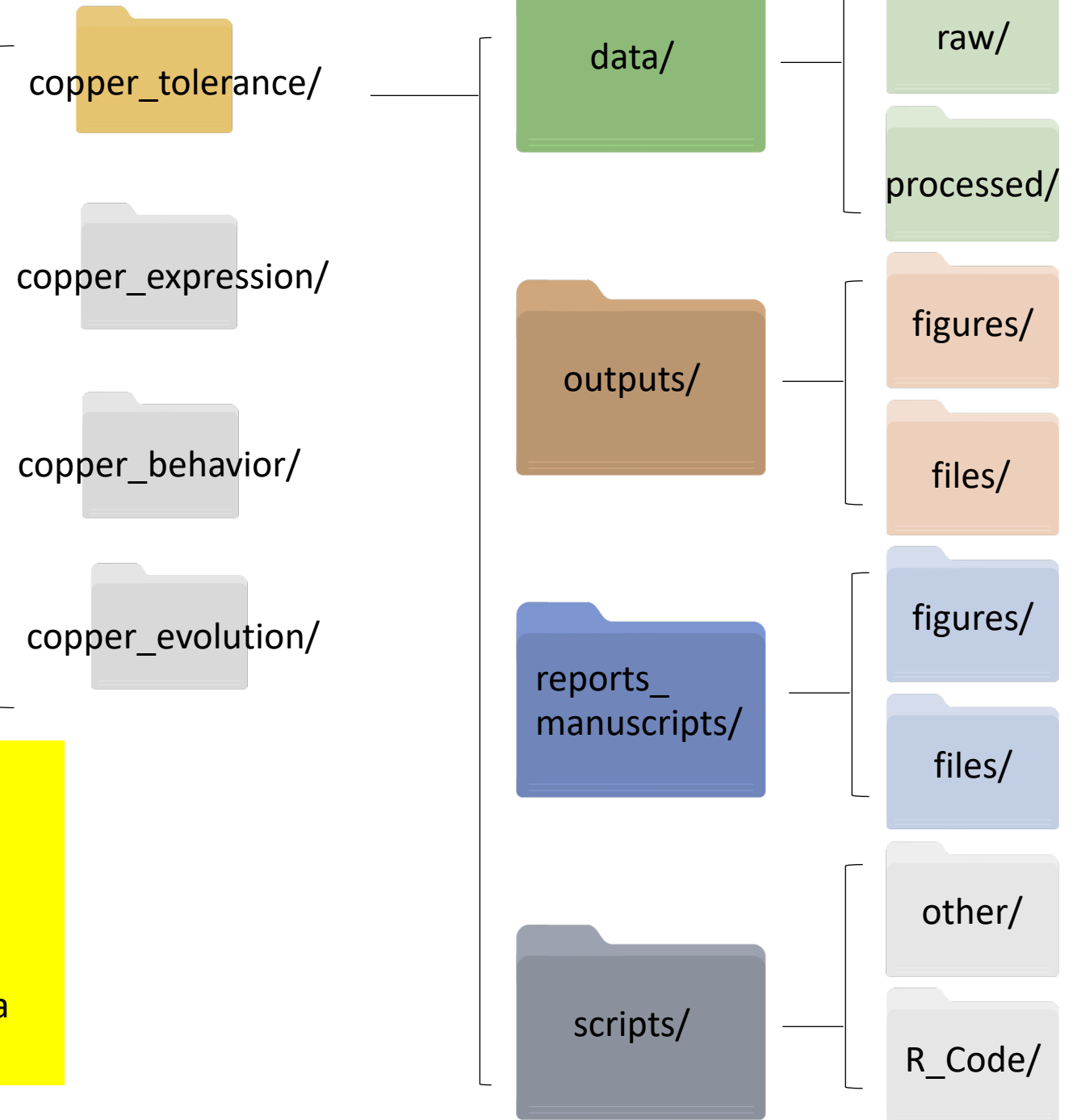
Not the best strategy…

Directory Structure:



Directory Structure:

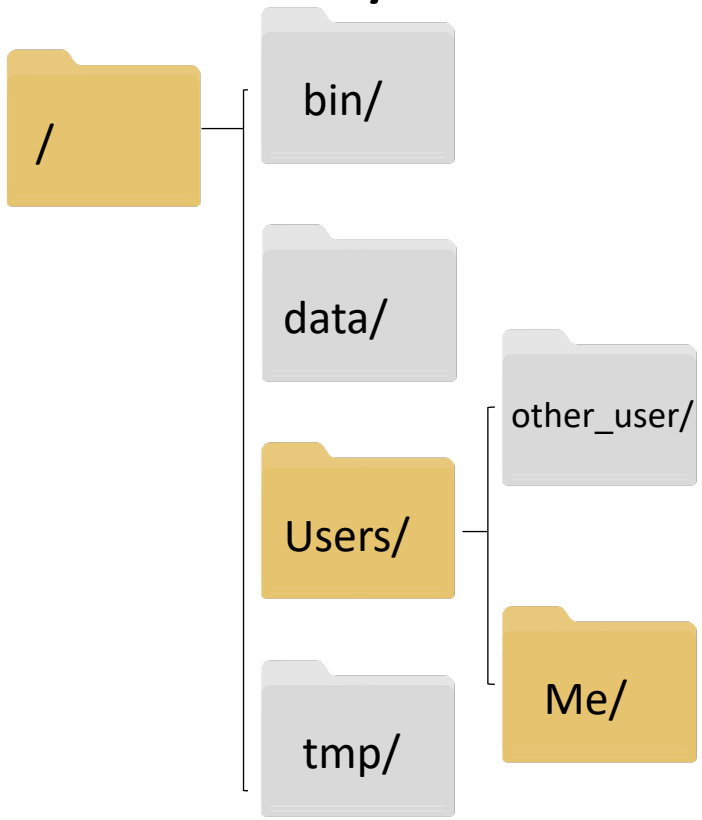




Tips:

- Use a template for project organization
- Commit to using upper/lower case
- Treat projects as whole units
  - E.g. copying one project directory moves everything you need for that project (code, data files, notes/ReadMe files)

Directory Structure:

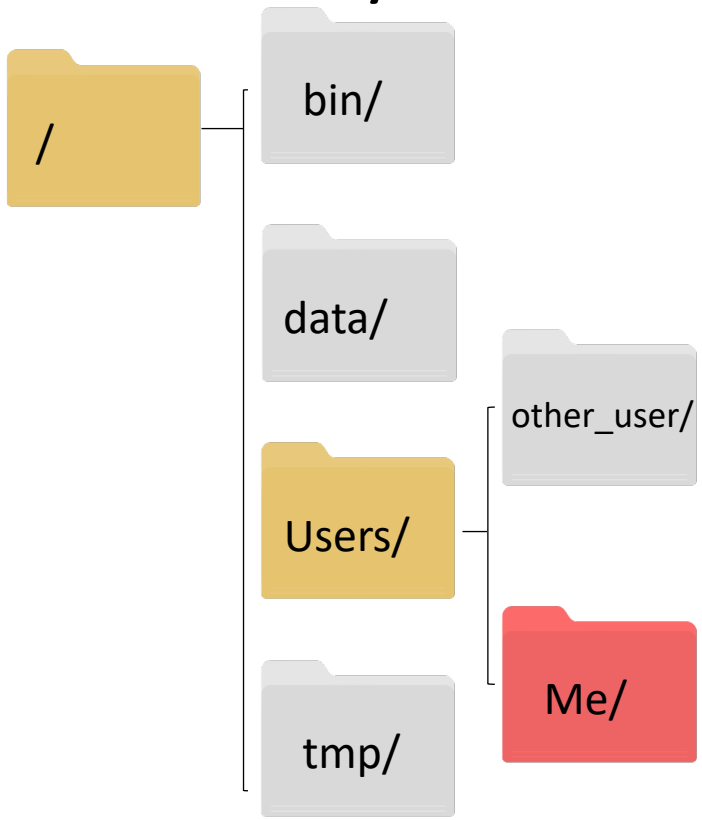


Do this:

- On your paper, copy this directory structure to use as a map.
- Open Terminal:
  - Mac:
    - Finder → Applications → Utilities → Terminal
  - PC:
- Put a marker (coin or other small object) on your map that shows where you are in your terminal (hint: look for the $ symbol).

- **Comment "done" in Zoom chat.**

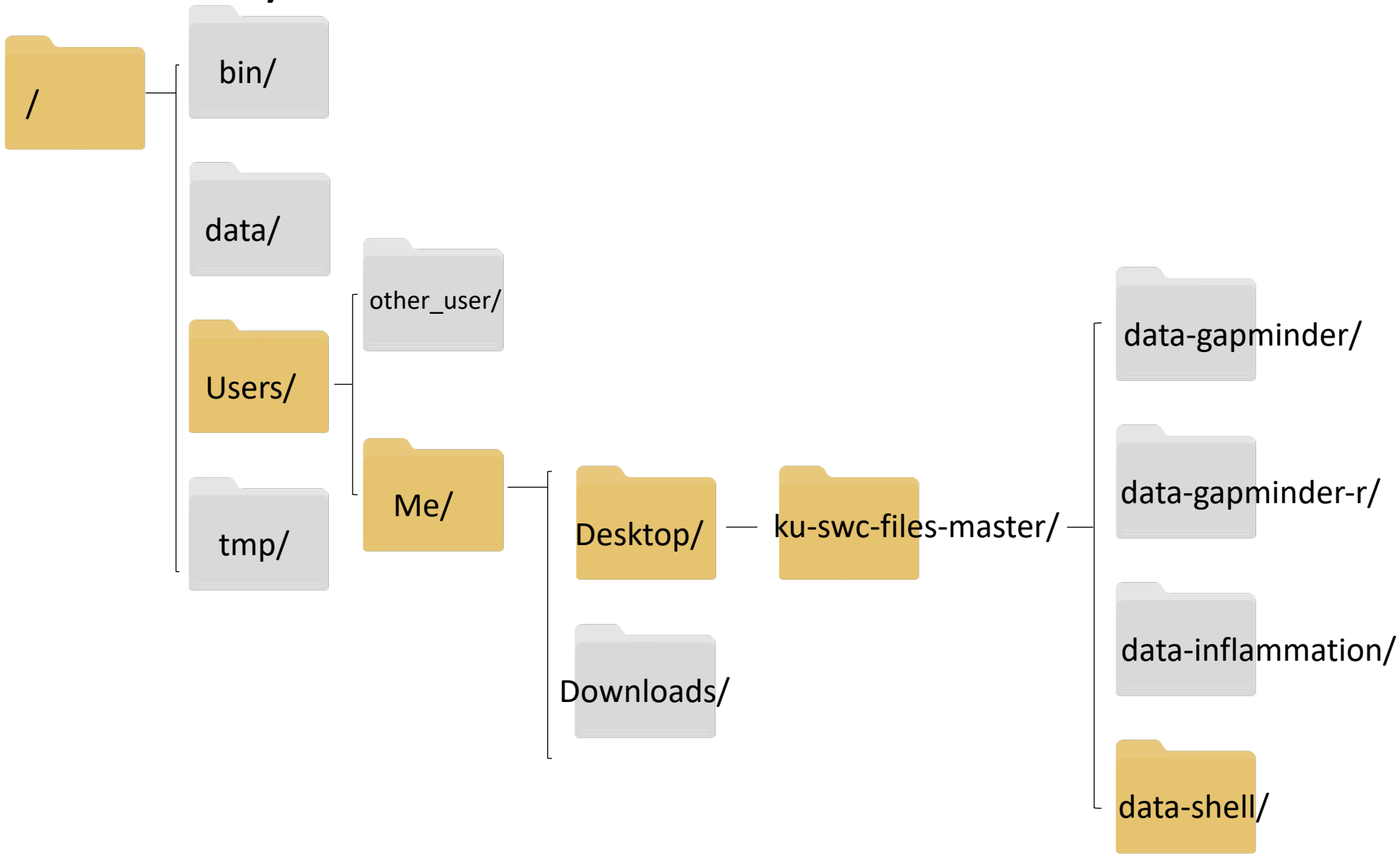
Directory Structure:



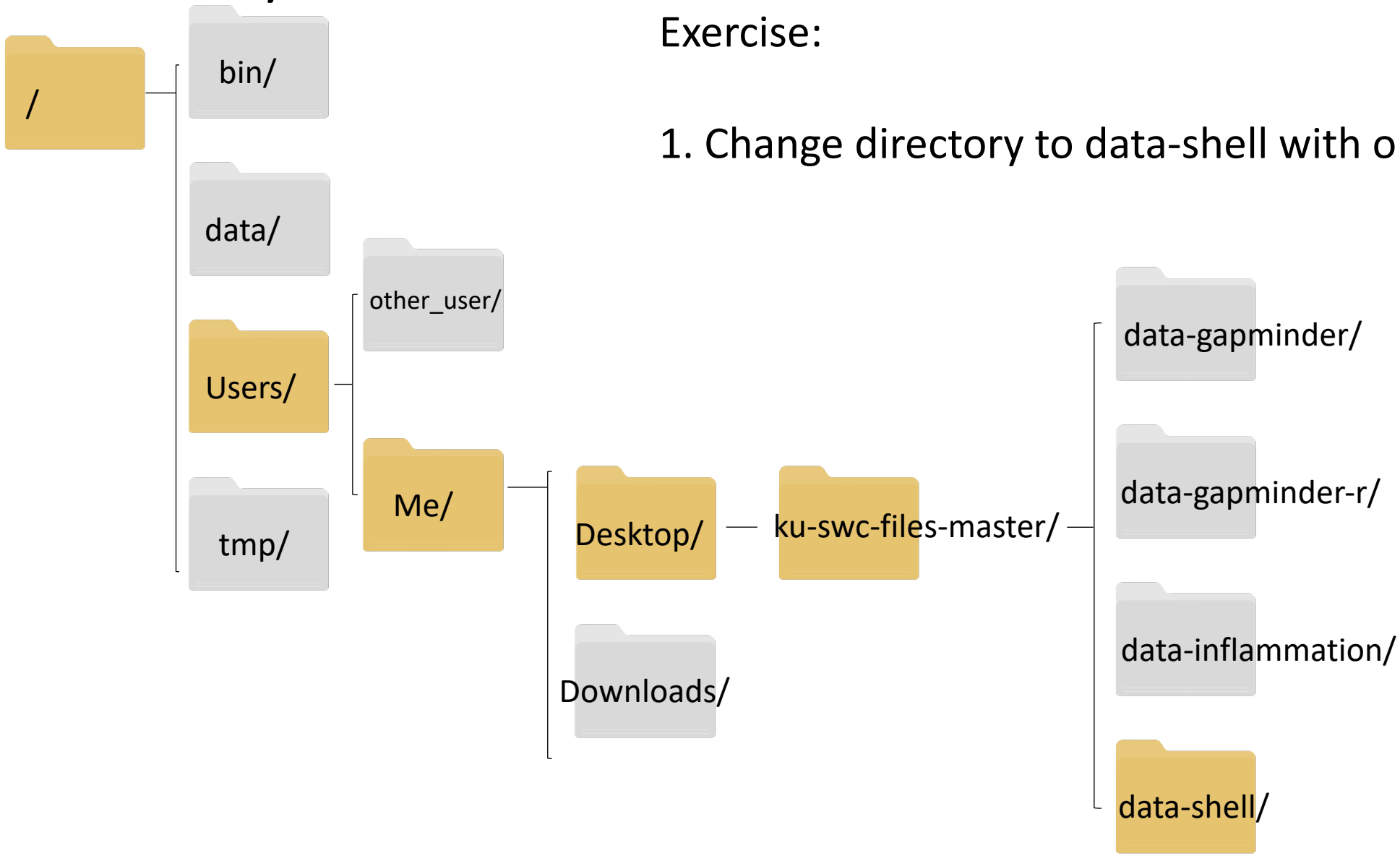
Exercise:

1. List the contents of the ku-swc-files-master directory from your current position

Directory Structure:



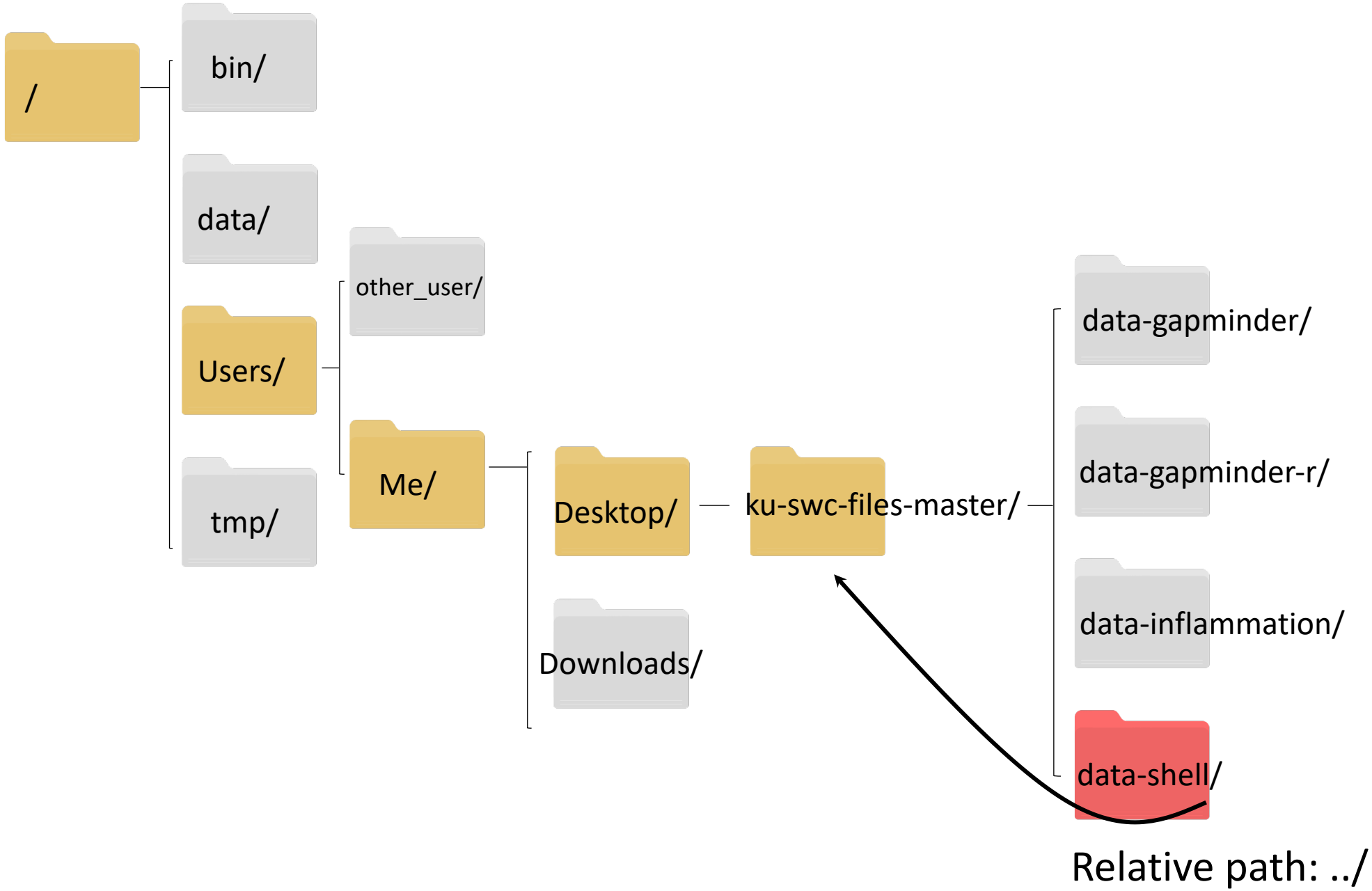
Directory Structure:



Exercise:

1. Change directory to data-shell with one line.

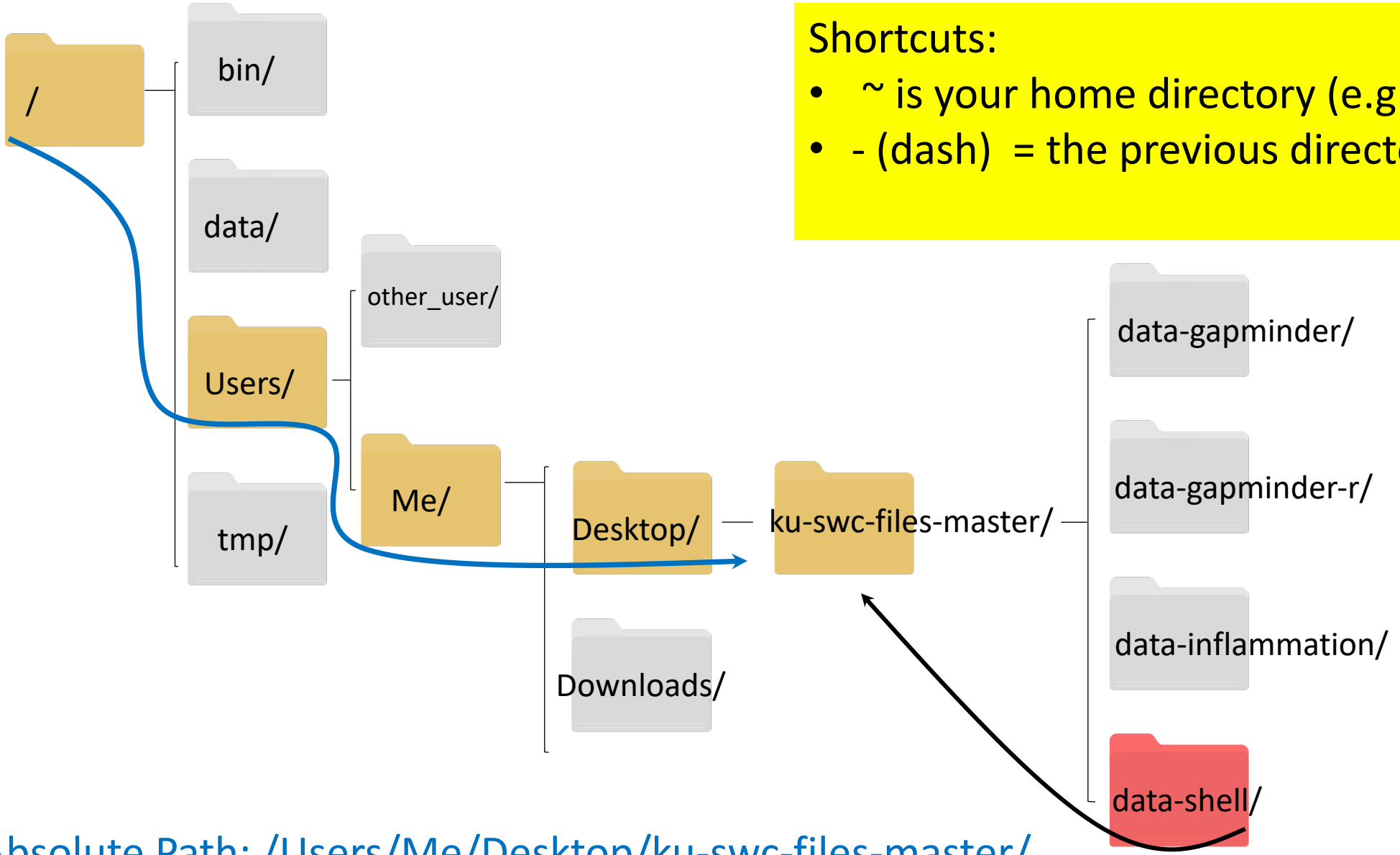
Relative vs Absolute Paths:



Relative vs Absolute Paths:

Shortcuts:

- ~ is your home directory (e.g. ~/ = /Users/Me/)
- - (dash) = the previous directory I was in.



Absolute Path: /Users/Me/Desktop/ku-swc-files-master/

Relative path: ../ku-swc-files-master/

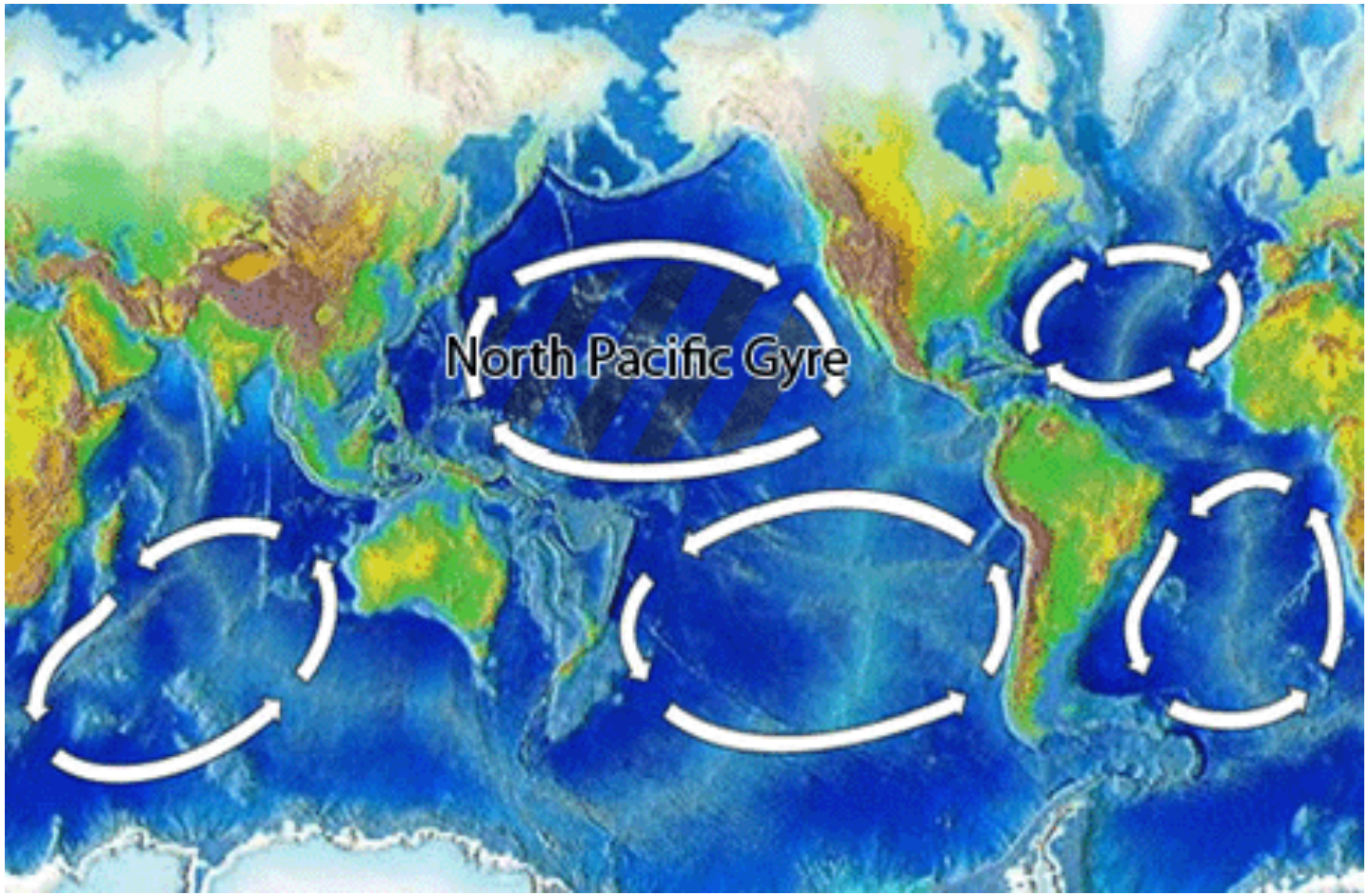
Exercise:

1. Starting from /Users/Me/Desktop/, which of the following could you use to navigate to your home directory?
   a) cd .
   b) cd /
   c) cd /home/Me
   d) cd ../..
   e) cd ~
   f) cd home
   g) cd ~/data/..
   h) cd
   i) cd ..

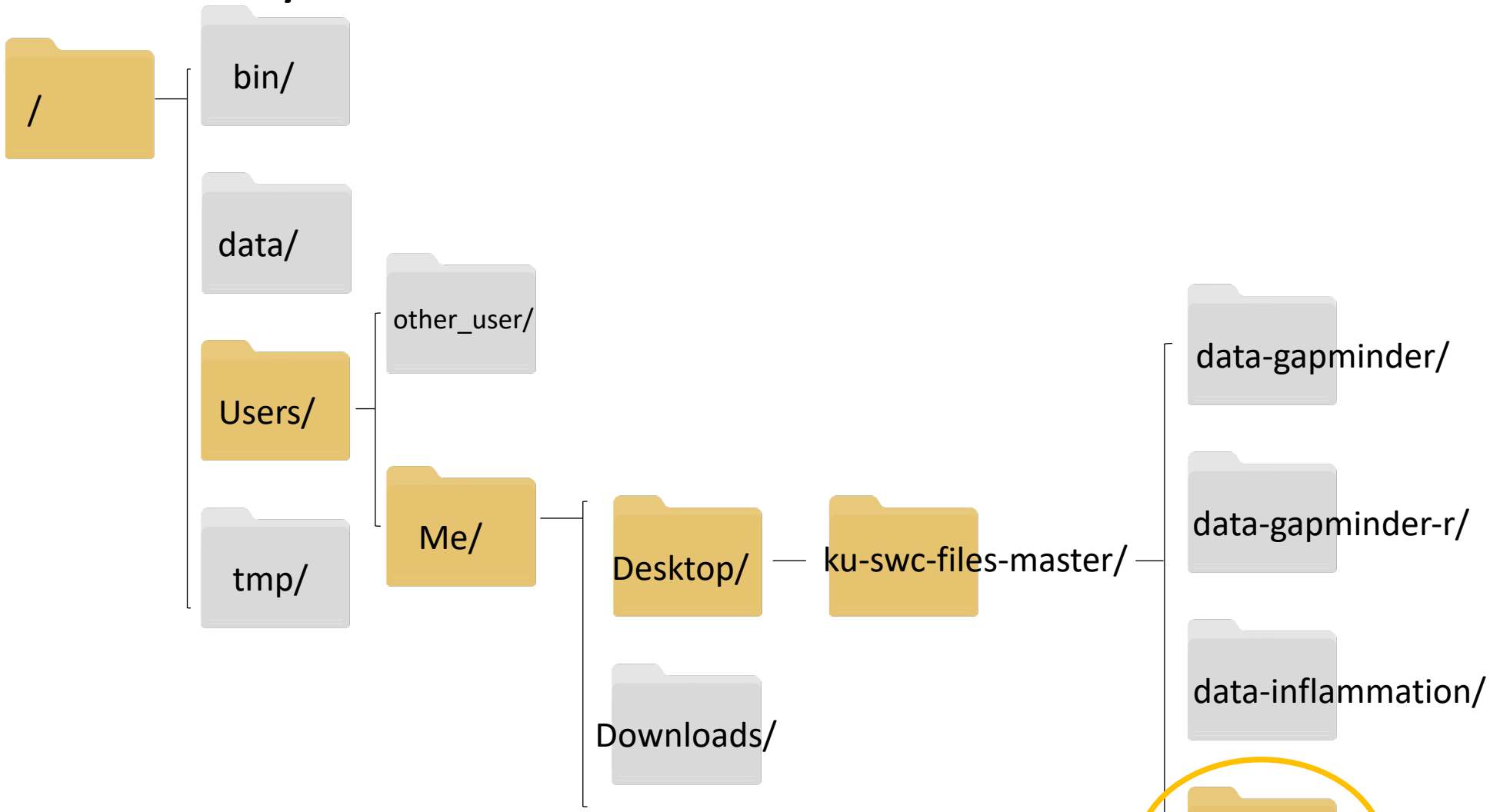
Exercise:

1. Starting from /Users/Me/Desktop/, which of the following could you use to navigate to your home directory?
   a) cd .
   b) cd /
   c) cd /home/Me
   d) cd ../..
   e) cd ~
   f) cd home
   g) cd ~/data/..
   h) cd
   i) cd ..

Case Study:

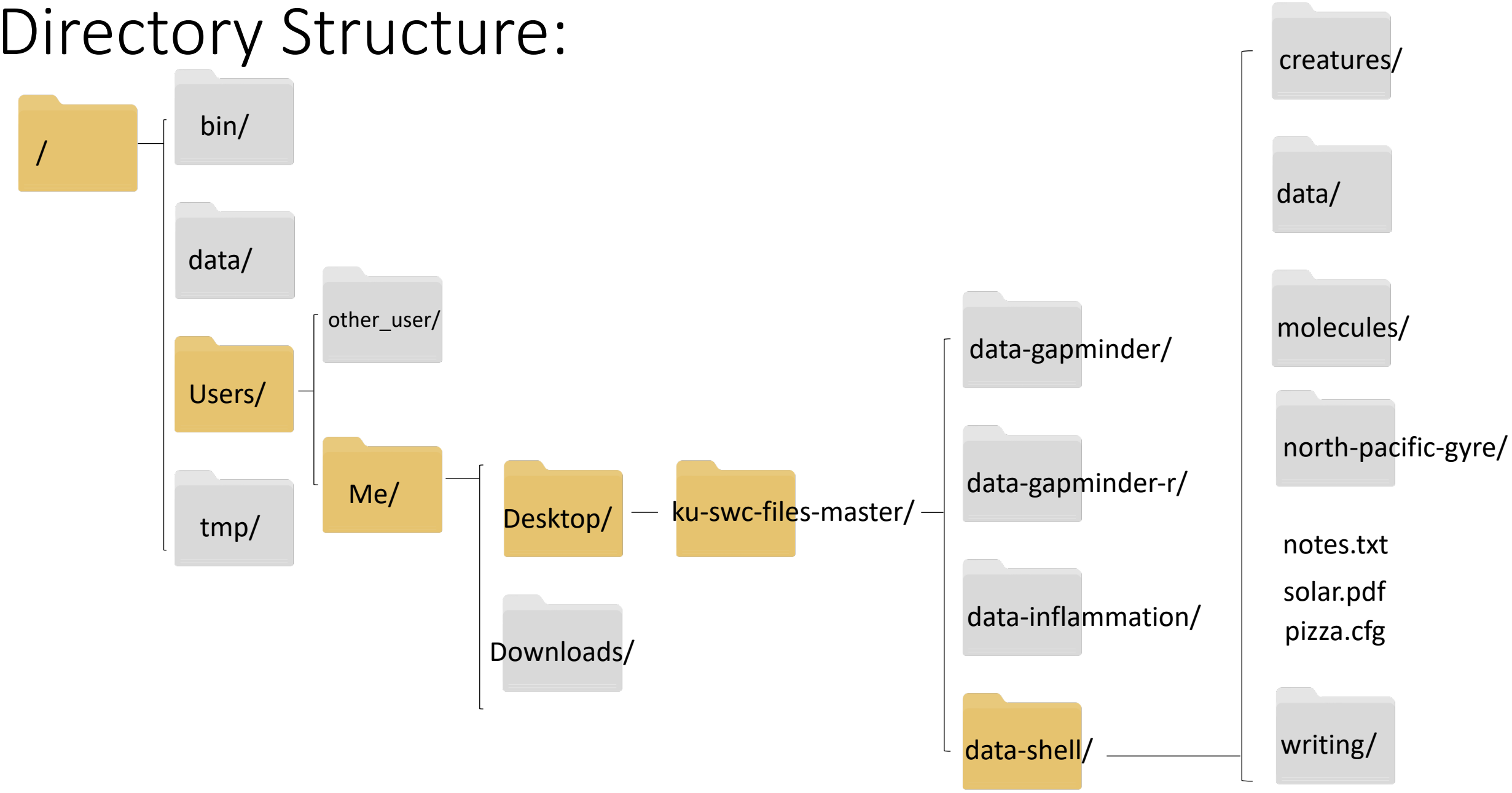


Directory Structure:

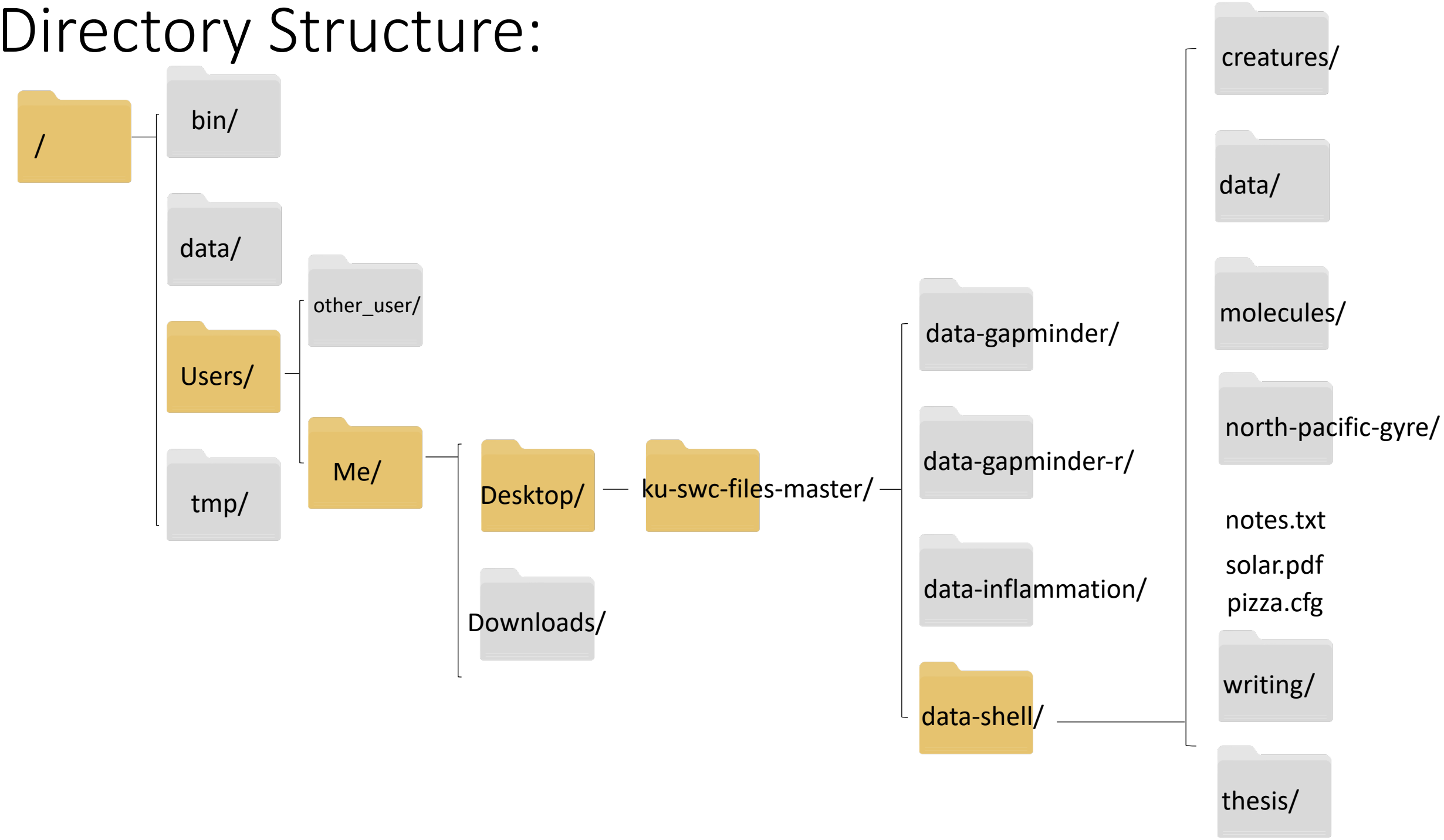


1. Navigate to data-shell
2. ls -F

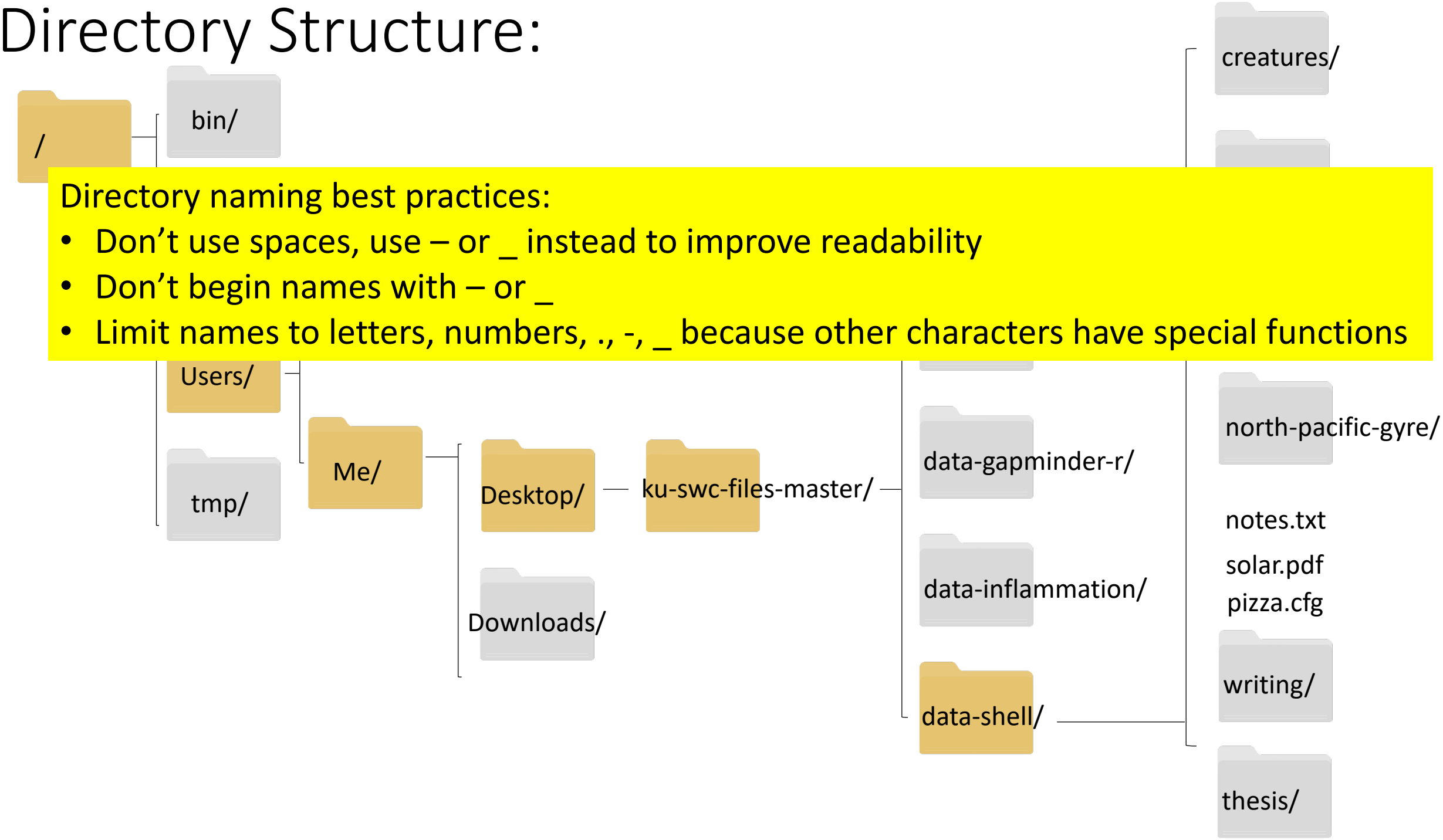
Directory Structure:



Directory Structure:



Directory Structure:



Directory naming best practices:

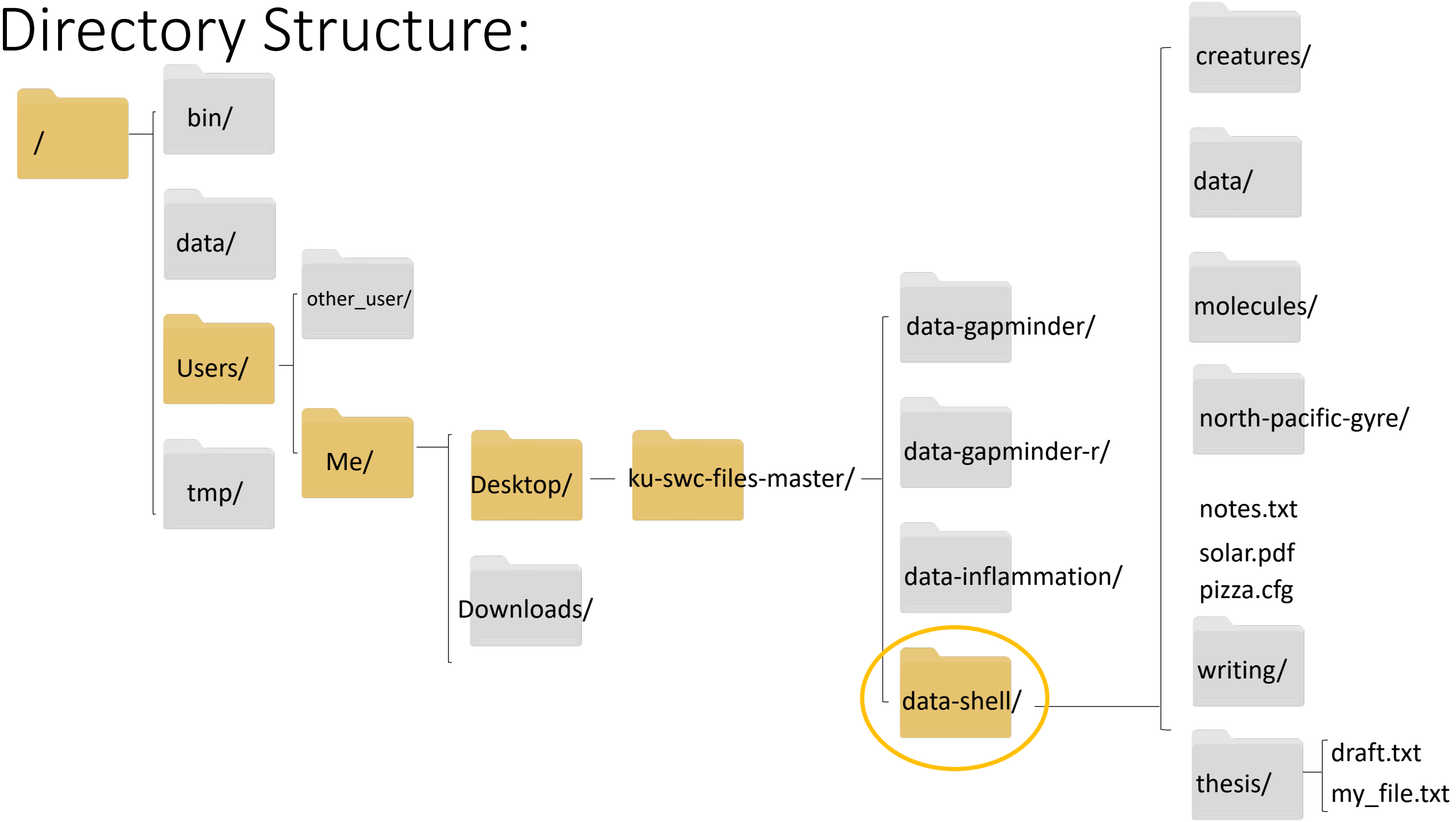
- Don’t use spaces, use – or _ instead to improve readability
- Don’t begin names with – or _
- Limit names to letters, numbers, ., -, _ because other characters have special functions

File Extensions:

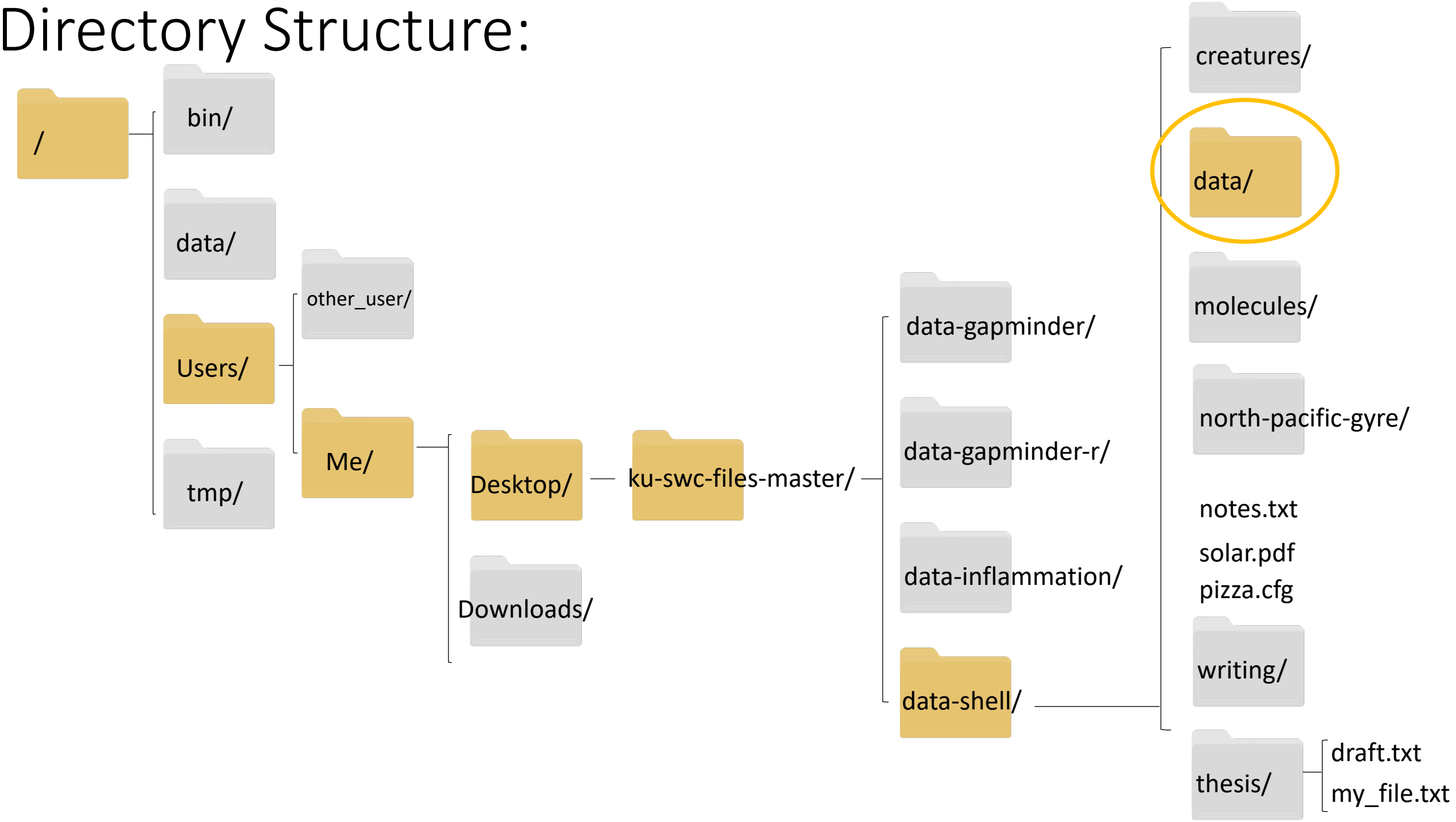
- Conventions exist that help programs and us differentiate between types of files
- Nothing prevents you from designating something as .mp3, but this doesn't automatically make it an audio recording.

https://www.howtogeek.com/wp-content/uploads/2018/06/icon-2488093_640.png

Directory Structure:



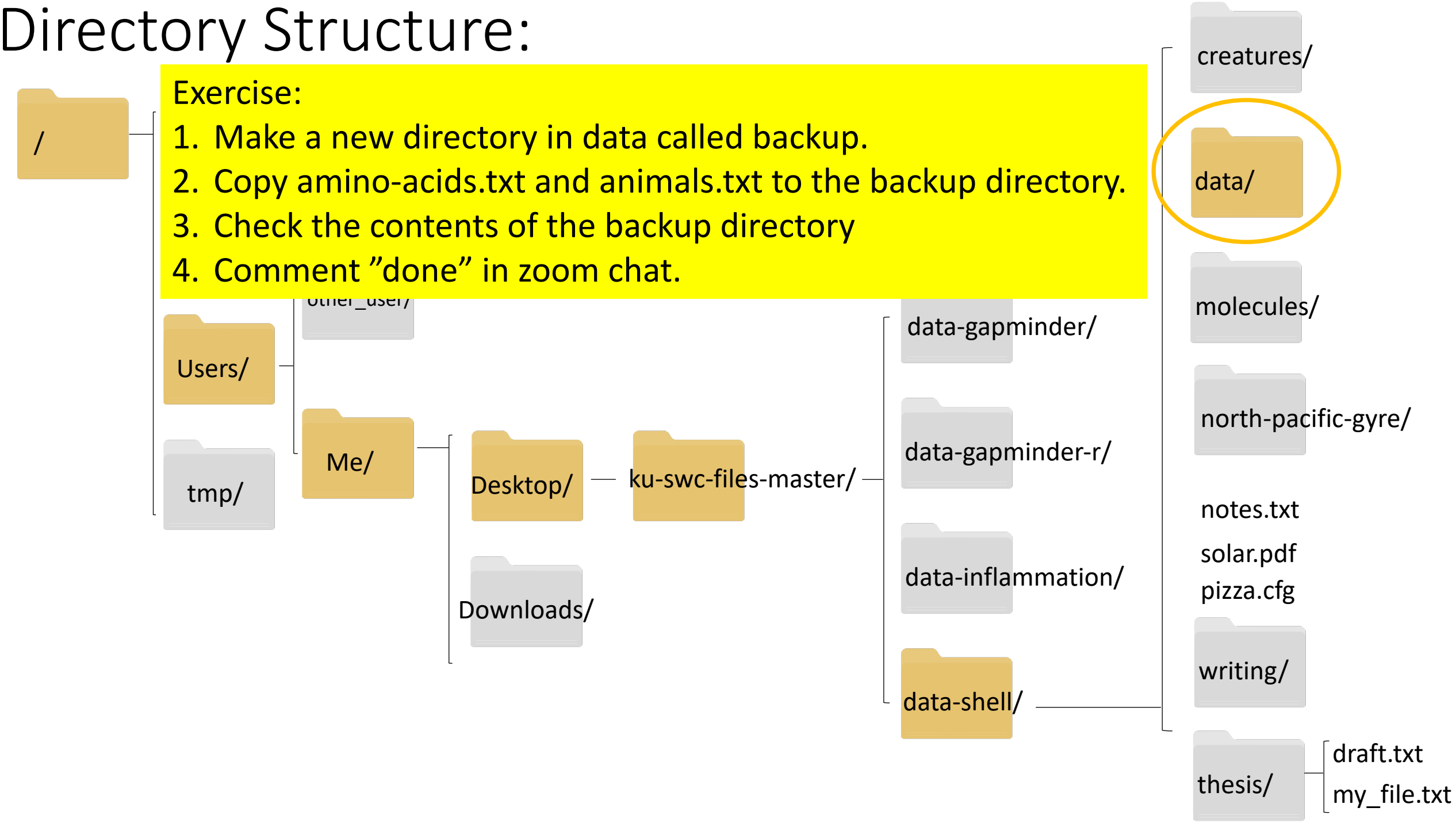
Directory Structure:



Directory Structure:

Exercise:

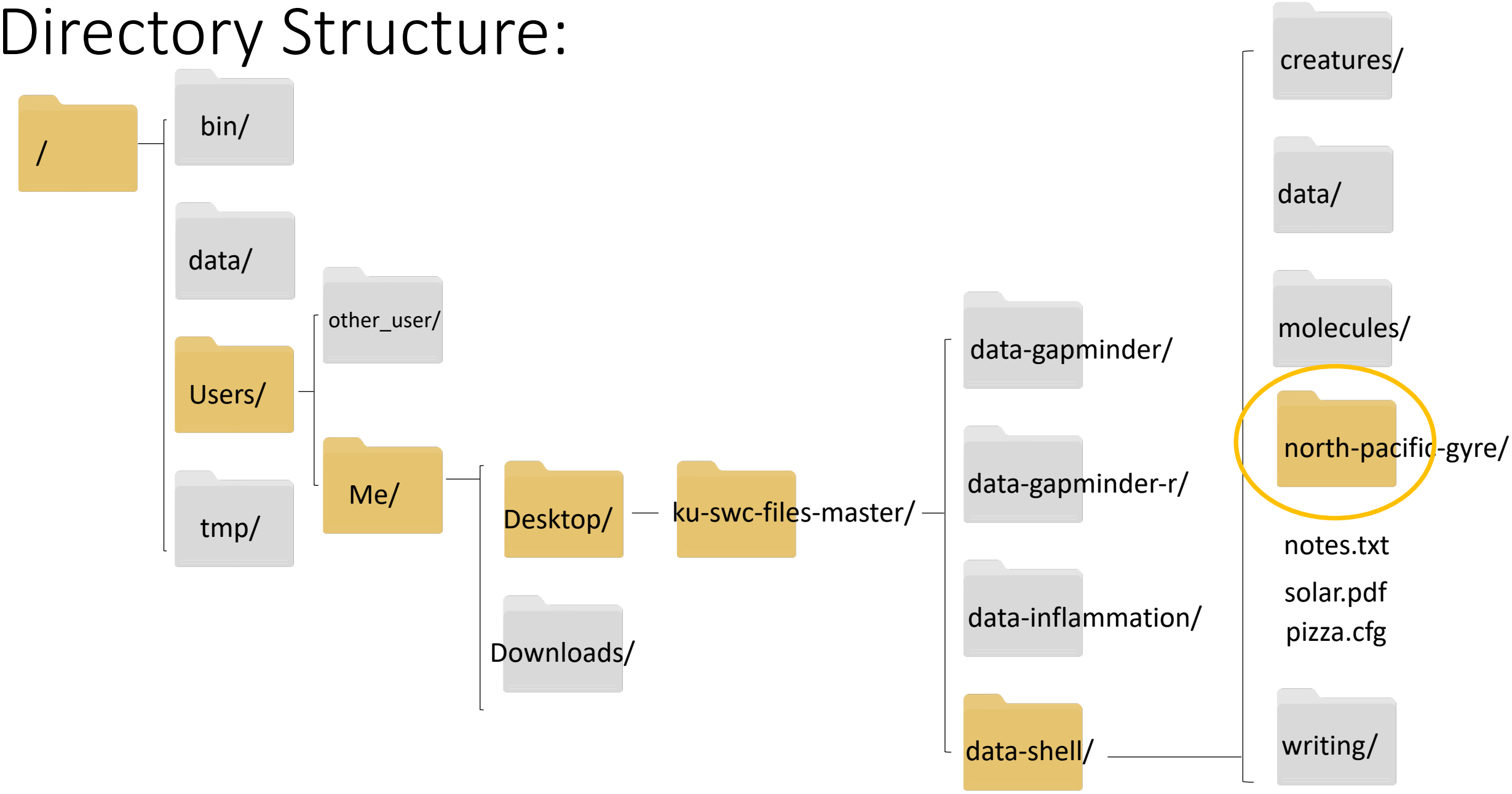
1. Make a new directory in data called backup.
2. Copy amino-acids.txt and animals.txt to the backup directory.
3. Check the contents of the backup directory
4. Comment ”done” in zoom chat.



Exercise:

1. In the molecules directory, create fructose.dat and sucrose.dat files.
2. In the molecules directory, create analyzed and raw directories.
3. Use a wildcard to move fructose.dat and sucrose.dat to the to the analyzed directory and copy all files that end in .pdb to the raw directory.
4. Comment “done” in zoom chat.

Directory Structure:



Loop syntax:

```
for thing in list_of_things
do
        operation_using $thing
done
```

- Uses an indexing system to move through each element in a list

thing = i
List_of_things = [2, 3, 4, 5]
Operation = “add 2”
Expected result: when i = 2, the answer is 4; when i = 3, the answer is 5…

More ways to use variable names:

HR_21006_Cu_S1_R1_001.fastq.gz
HR_21006_NA_S2_R1_001.fastq.gz
HR_21012_Cu_S5_R1_001.fastq.gz
HR_21012_NA_S6_R1_001.fastq.gz
HR_21026_Cu_S87_R1_001.fastq.gz
HR_21026_NA_S10_R1_001.fastq.gz
HR_21042_Cu_S13_R1_001.fastq.gz
HR_21042_NA_S14_R1_001.fastq.gz
HR_21050_Cu_S17_R1_001.fastq.gz
HR_21050_NA_S18_R1_001.fastq.gz
HR_21060_Cu_S21_R1_001.fastq.gz
HR_21060_NA_S22_R1_001.fastq.gz
HR_21064_Cu_S25_R1_001.fastq.gz
HR_21064_NA_S26_R1_001.fastq.gz
HR_21074_Cu_S29_R1_001.fastq.gz
HR_21074_NA_S30_R1_001.fastq.gz

${sample}_R1_001.fastq.gz

Try this: what is the output?

```
$ for datafile in *.pdb
> do
> ls *.pdb
> done
```

Try this: what is the output?

```
$ for datafile in *.pdb
> do
> ls *.pdb
> done
```

Now try this:

```
$ for datafile in *.pdb
> do
> ls $datafile
> done
```

Try this: what is the output?

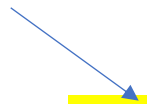
```
$ for datafile in *.pdb
> do
> ls *.pdb
> done
```

Now try this:

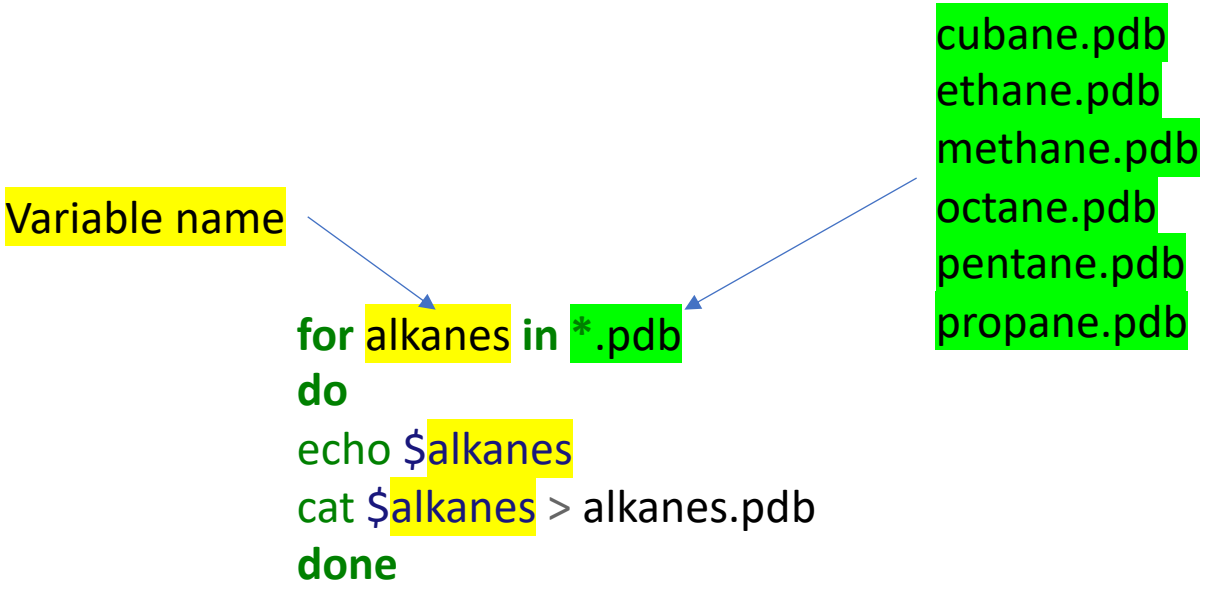
```
$ for datafile in *.pdb
> do
> ls $datafile
> done
```

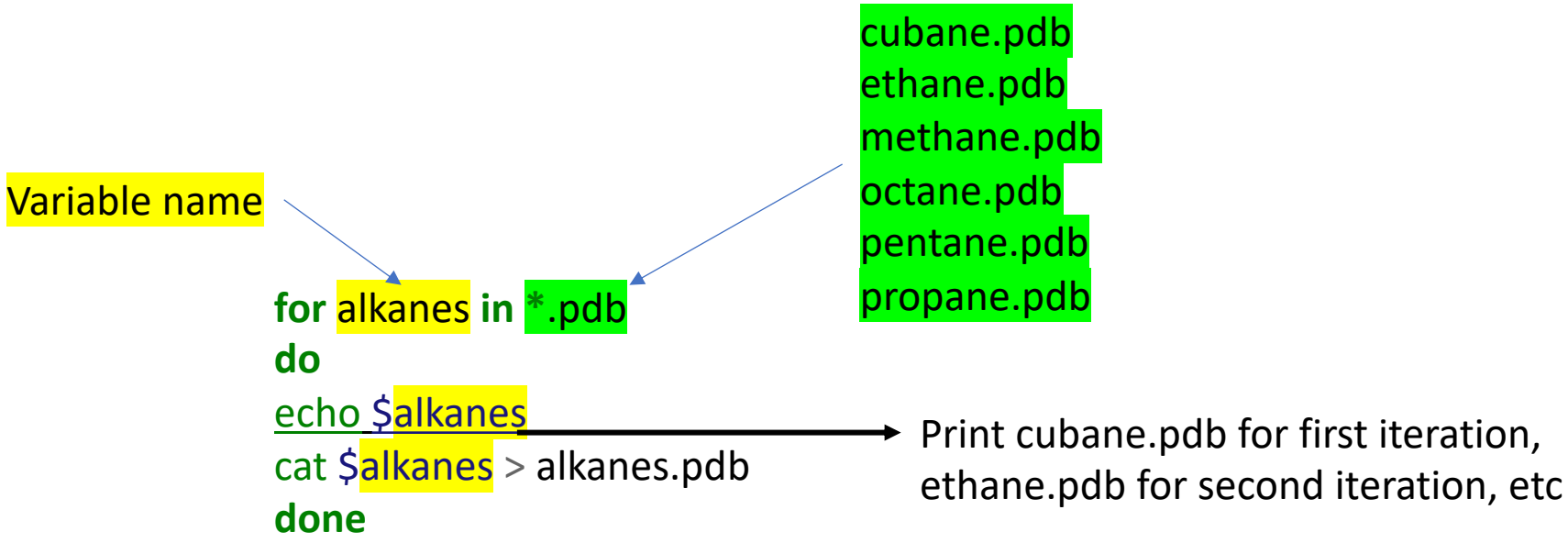
```
for alkanes in *.pdb
do
echo $alkanes
cat $alkanes > alkanes.pdb
done
```

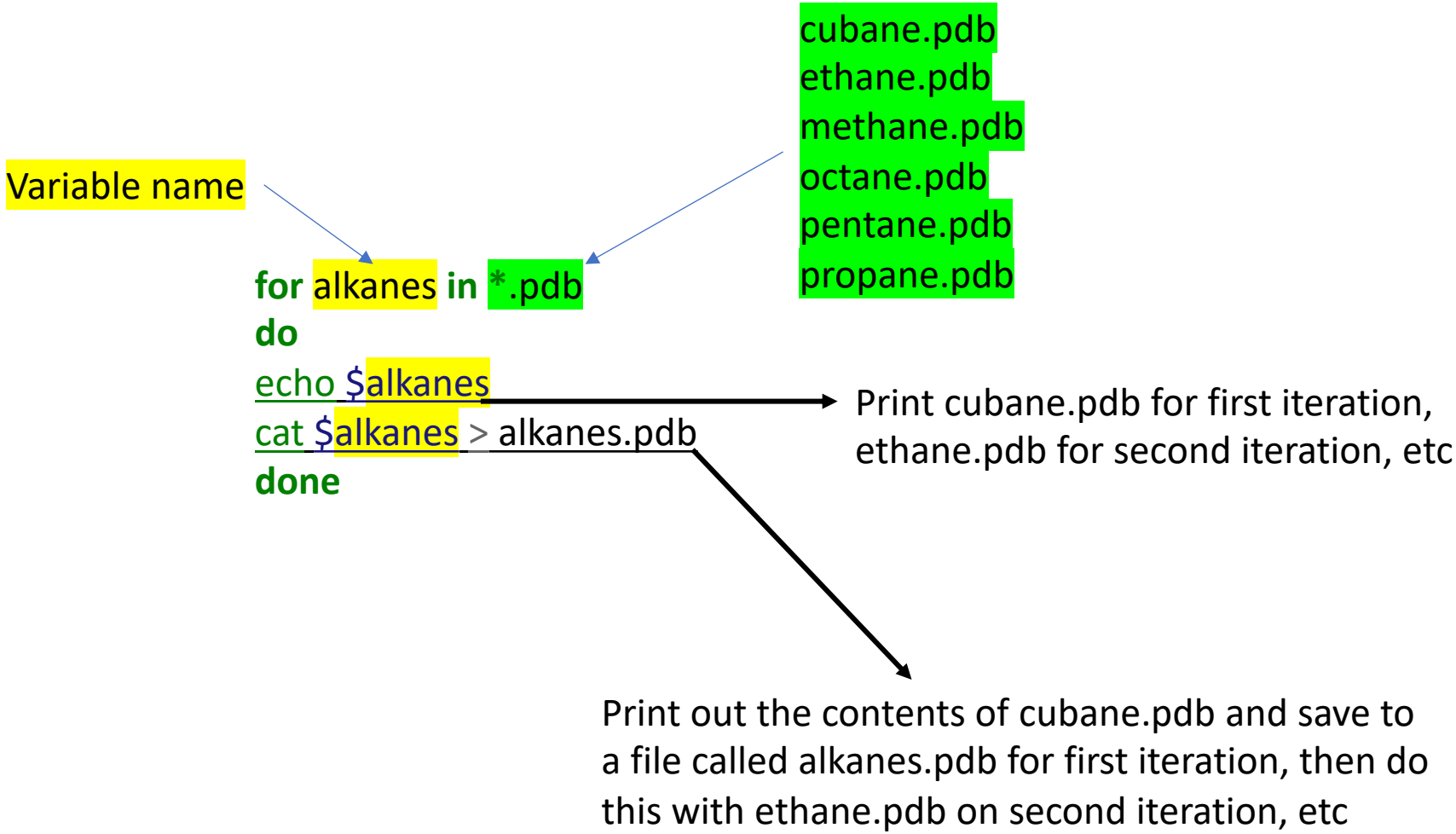
Variable name



```
for alkanes in *.pdb
do
echo $alkanes
cat $alkanes > alkanes.pdb
done
```


Variable name
cubane.pdb
ethane.pdb
methane.pdb
octane.pdb
pentane.pdb
propane.pdb
for alkanes in *.pdb
do
echo $alkanes
cat $alkanes > alkanes.pdb
done

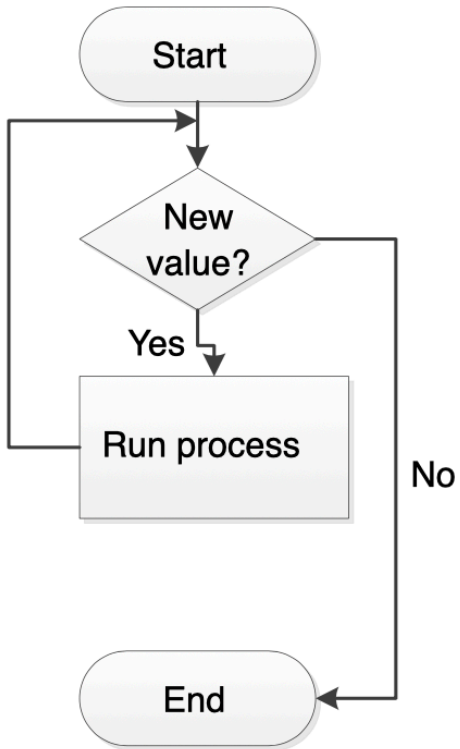

Variable name
cubane.pdb
ethane.pdb
methane.pdb
octane.pdb
pentane.pdb
propane.pdb
for alkanes in *.pdb
do
echo $alkanes
cat $alkanes > alkanes.pdb
done
Print cubane.pdb for first iteration,
ethane.pdb for second iteration, etc


cubane.pdb
ethane.pdb
methane.pdb
octane.pdb
pentane.pdb
propane.pdb
Variable name
for alkanes in *.pdb
do
echo $alkanes
cat $alkanes > alkanes.pdb
done
Print cubane.pdb for first iteration, ethane.pdb for second iteration, etc
Print out the contents of cubane.pdb and save to a file called alkanes.pdb for first iteration, then do this with ethane.pdb on second iteration, etc

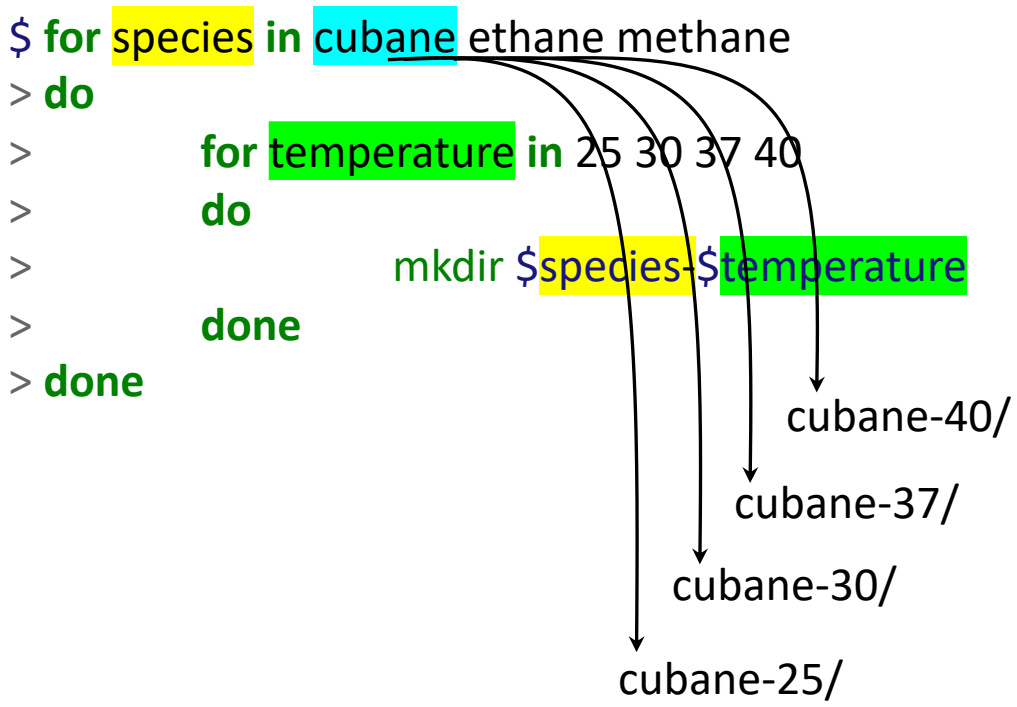
Shell script

```
for filename in *.dat
do
 echo cp $filename original-$filename
done
```

Phase

Process Flow	Variable value	Process	Stdout
	basilisk.dat	echo	cp basilisk.dat original-basilisk.dat
	minotaur.dat	echo	cp minotaur.dat original-minotaur.dat
	unicorn.dat	echo	cp unicorn.dat original-unicorn.dat

```
$ for species in cubane ethane methane
> do
>         for temperature in 25 30 37 40
>         do
>                 mkdir $species-$temperature
>         done
> done
```


$ for species in cubane ethane methane
> do
> for temperature in 25 30 37 40
> do
> mkdir $species-$temperature
> done
> done
cubane-40/
cubane-37/
cubane-30/
cubane-25/