R PROGRAMMING CONCEPTS

Sep 13-14, 2023

You should have R & RStudio installed
https://sites.northwestern.edu/researchcomputing/resources/r-and-rstudio/

Download materials from:
https://github.com/nuitrcs/R-programming-concepts-Chicago-2023

Northwestern | INFORMATION TECHNOLOGY

Research Computing and Data Services

Ritika Giri

Christina Maimone

Efrén Cruz Cortés

We're here to help after the workshops!

bit.ly/rcdsconsult

BYOD

- BYOD = Bring Your Own Data
- Small working groups, meet weekly
- https://sites.northwestern.edu/researchcomputing/events-and-programs/byod/

What are we doing today?

- Getting to know R and RStudio
- Learning concepts that you need for R (that also apply to other programming languages)

What are we doing?

Topics

R / Rstudio

File paths

Working directory

R scripts and Documents

Functions

Variables

Packages

Importing data

Data frames

Indexing data frames and vectors

Subsetting data

Missing values


TERMINOLOGY
Programming Languages, Consoles, and GUIs

GUIs

GUI: Graphical User Interface: Programs on your computer with buttons and other visual elements; you can interact with a mouse or touchscreen

Powerpoint, Word, SPSS, Zoom, internet browser, etc.

PROBLEM:

HOW DO WE TELL OTHERS WHAT WE DID?
HOW DO WE KEEP TRACK OF WHAT WE DID FOR OURSELVES?

Programming Languages

A system for giving instructions to a computer (writing code)

- *Specific words and characters in a specific order*

Programming Languages

A system for giving instructions to a computer (writing code)

- *Specific words and characters in a specific order*

You give these instructions to an **interpreter** or **compiler** which translates them into something your specific computer can use

- ***Interpreters*** *(R and Python): Give an instruction, get a result – directly executes the instructions*
- ***Compilers*** *(C++, Java): Write an entire program, then translate the entire program into computer-level instructions, and then run that program*

Console/Terminal/Shell/Command Line Interface

Simple computer program where we type *commands*, one *line* at a time (hence: **command line**)

They can run different interpreters: bash, zsh, PowerShell (Windows), R, Python

Type instructions at a prompt (**>** or **$** or **%** or **>>>** or **:** or others)

Used for basic operations and running other programs

Console/Terminal/Shell

Mac:

Applications > Terminal

In RStudio, "Terminal" tab (we'll see this later)

Language-specific consoles (like R)

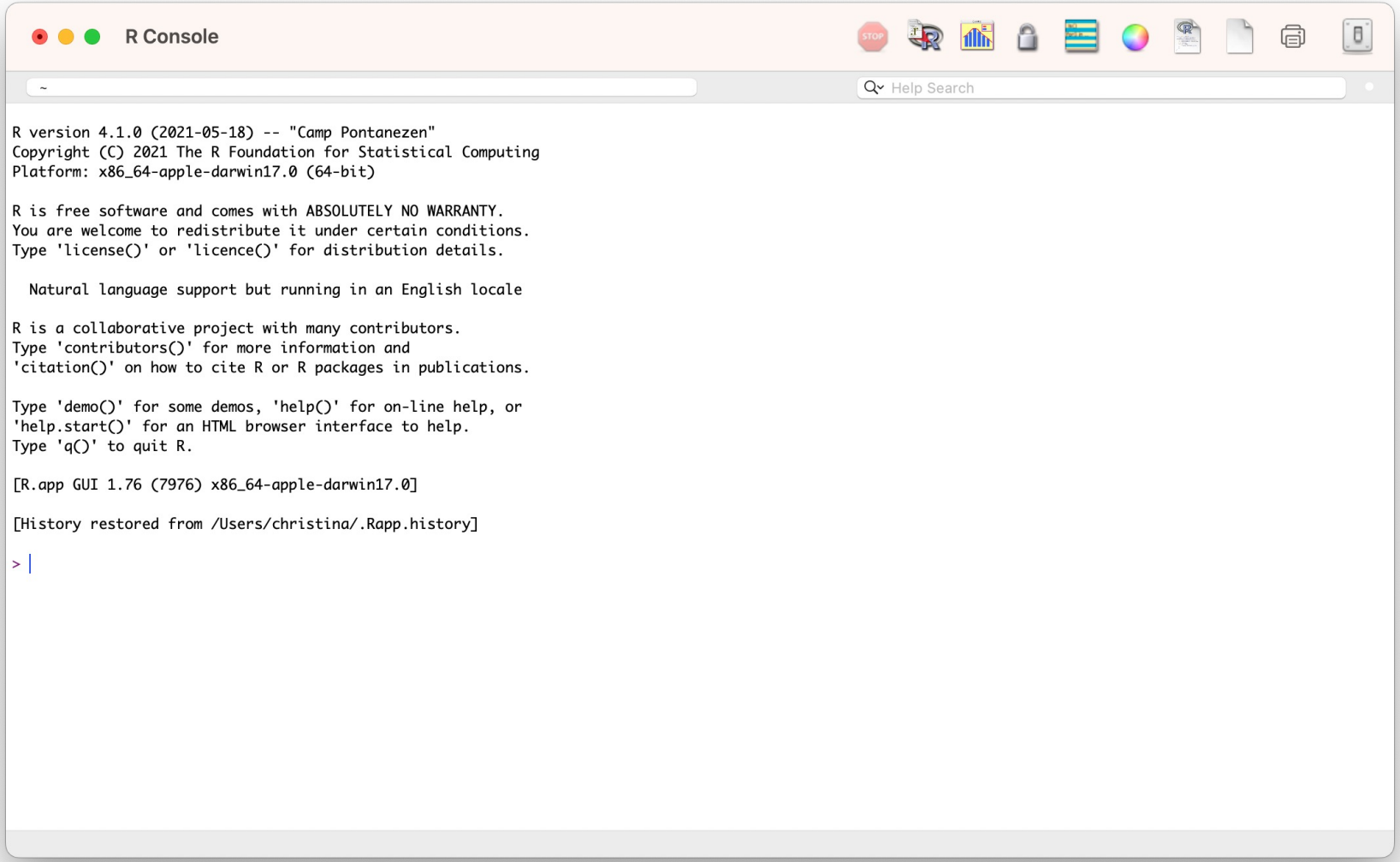
Windows:

Windows Console or Command Prompt (Start button, search for cmd)

PowerShell

Language-specific consoles (like R)


christina — -zsh — 90×25
Last login: Fri Jun 2 16:47:55 on ttys000
/Users/christina %


R Console
~
Help Search
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[R.app GUI 1.76 (7976) x86_64-apple-darwin17.0]
[History restored from /Users/christina/.Rapp.history]
>

BUT THAT'S SOMETIMES ANNOYING AND INCONVENIENT…

AND HOW DO WE RECORD WHAT WE DID?

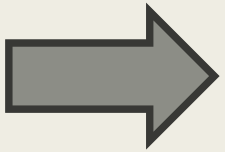
IDEs

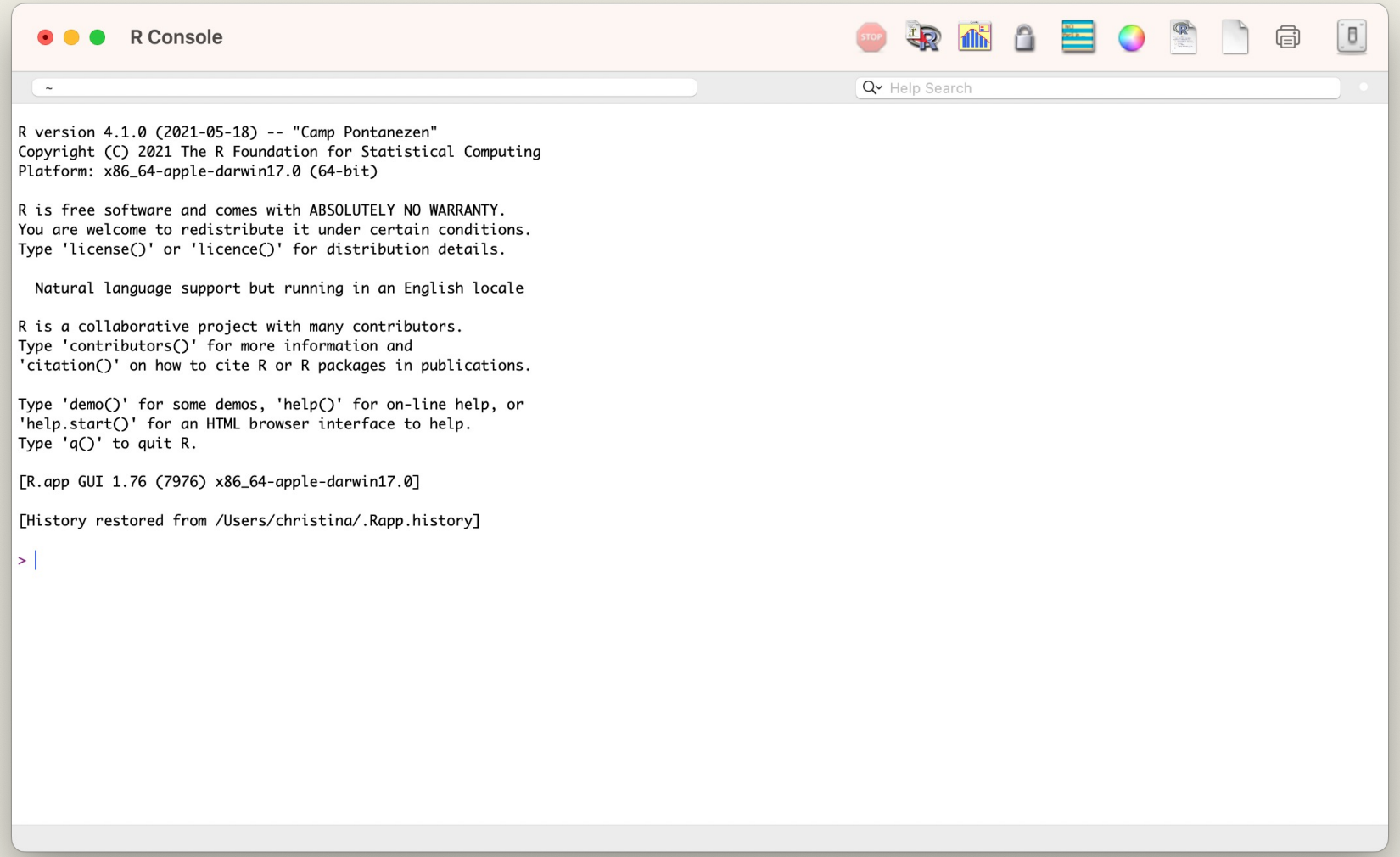
IDE: Integrated Development Environment (e.g. RStudio)**:** A GUI that helps make writing computer programs easier and interfaces with an interpreter or compiler

Usually includes:

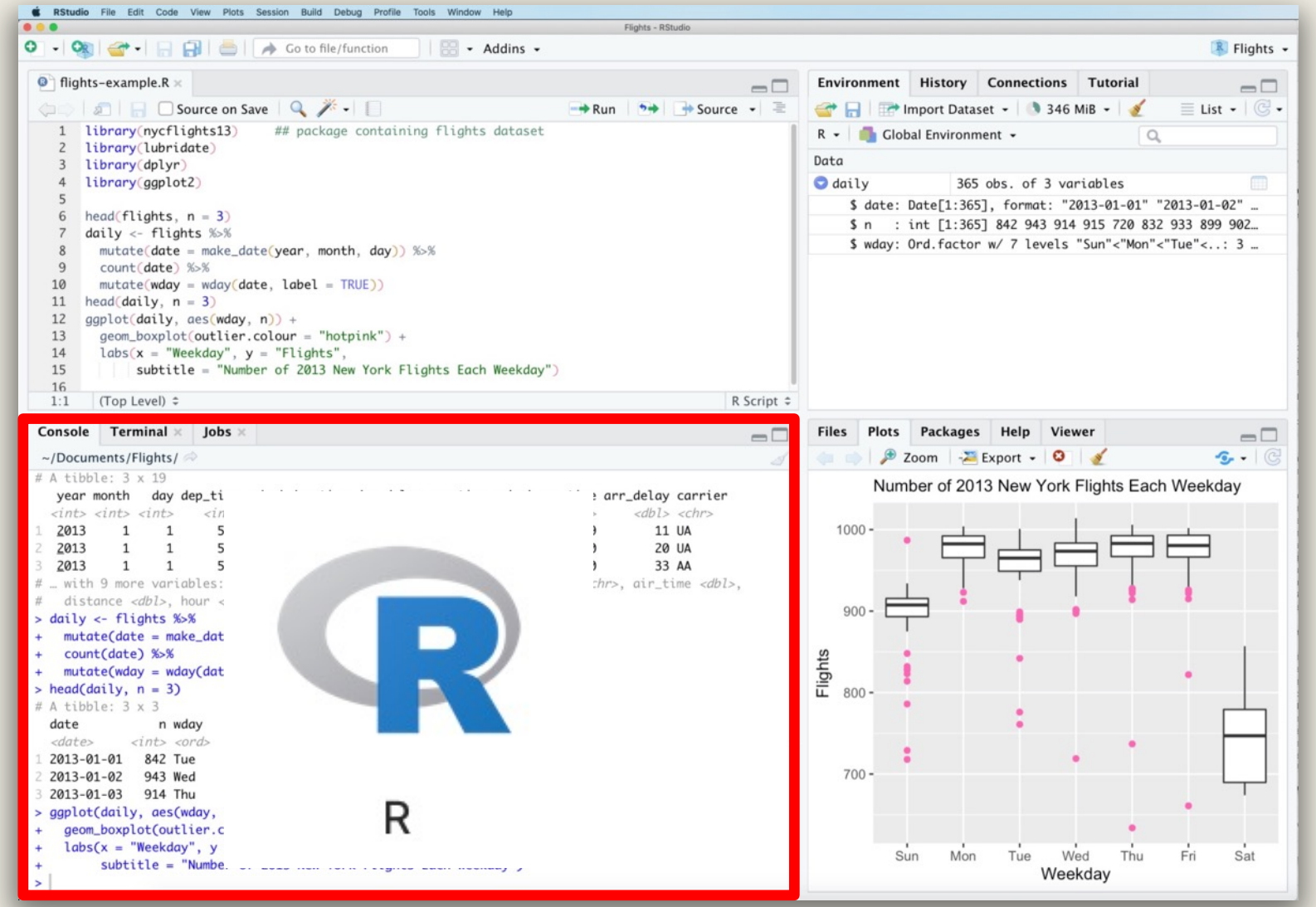
- File explorer
- Area for writing code/scripts
- Console for running interpreted languages
- General terminal window for your system


R




R Console
~
Help Search
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[R.app GUI 1.76 (7976) x86_64-apple-darwin17.0]
[History restored from /Users/christina/.Rapp.history]
>

RStudio


flights-example.R
library(nycflights13) ## package containing flights dataset
library(lubridate)
library(dplyr)
library(ggplot2)
head(flights, n = 3)
daily <- flights %>%
mutate(date = make_date(year, month, day)) %>%
count(date) %>%
mutate(wday = wday(date, label = TRUE))
head(daily, n = 3)
ggplot(daily, aes(wday, n)) +
geom_boxplot(outlier.colour = "hotpink") +
labs(x = "Weekday", y = "Flights",
subtitle = "Number of 2013 New York Flights Each Weekday")
Console
Terminal
Jobs
~/Documents/Flights/
Environment
History
Connections
Tutorial
Import Dataset
346 MiB
Global Environment
Data
daily
365 obs. of 3 variables
$ date: Date[1:365], format: "2013-01-01" "2013-01-02" …
$ n : int [1:365] 842 943 914 915 720 832 933 899 902…
$ wday: Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 3 …
Files
Plots
Packages
Help
Viewer
Zoom
Export
Number of 2013 New York Flights Each Weekday
Flights
1000
900
800
700
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekday

EVERYTHING THAT IS AN R COMMAND STILL HAS TO RUN THROUGH THE R INTERPRETER (IN THE R CONSOLE IN RSTUDIO)

Some buttons and menus write R code for us…

Yay!

We can keep a full record of everything we do!

Don’t have to write down instructions like:

open this menu, pick this option, check the third button, etc.

where the steps might vary with software version, operating system, etc

Means we *can* do reproducible science and research

Does not mean that this happens automatically!

Interpreted Programming Languages

Interactive programming: Interact with the interpreter via a console or terminal at the command line; execute one command at a time

> For reproducible research, still want keep a record of what you do

Batch programming: Write a script with multiple instructions that can be re-run in order

> You are writing a reproducible record (but still need to pay attention to things like your data files)

Which of these are true?

A: RStudio is a programming language

B: R and RStudio are the same program

C: RStudio is an IDE

D: RStudio is a GUI

E: R automatically makes our code reproducible

F: All R commands have to run through the R console/interpreter

Open RStudio

Type some math equations in the console and get answers

> *Press enter at the end of a line to run the command*

Open the file script1.R in RStudio

File menu > Open File…

or

Navigate to the file in the bottom right Files window in RStudio and click on the file to open it

DATA TYPES & BOOLEAN OPERATORS

Head back to RStudio...

Which of these are true?

A: "\n" is an empty string

B: (TRUE & FALSE) | (FALSE | TRUE)

C: TRUE == 1

D: “Dog” == “dog”

E: "this is a valid character data'

F: The value false can be represented as: FALSE, F, or false


VARIABLES

Variables

3

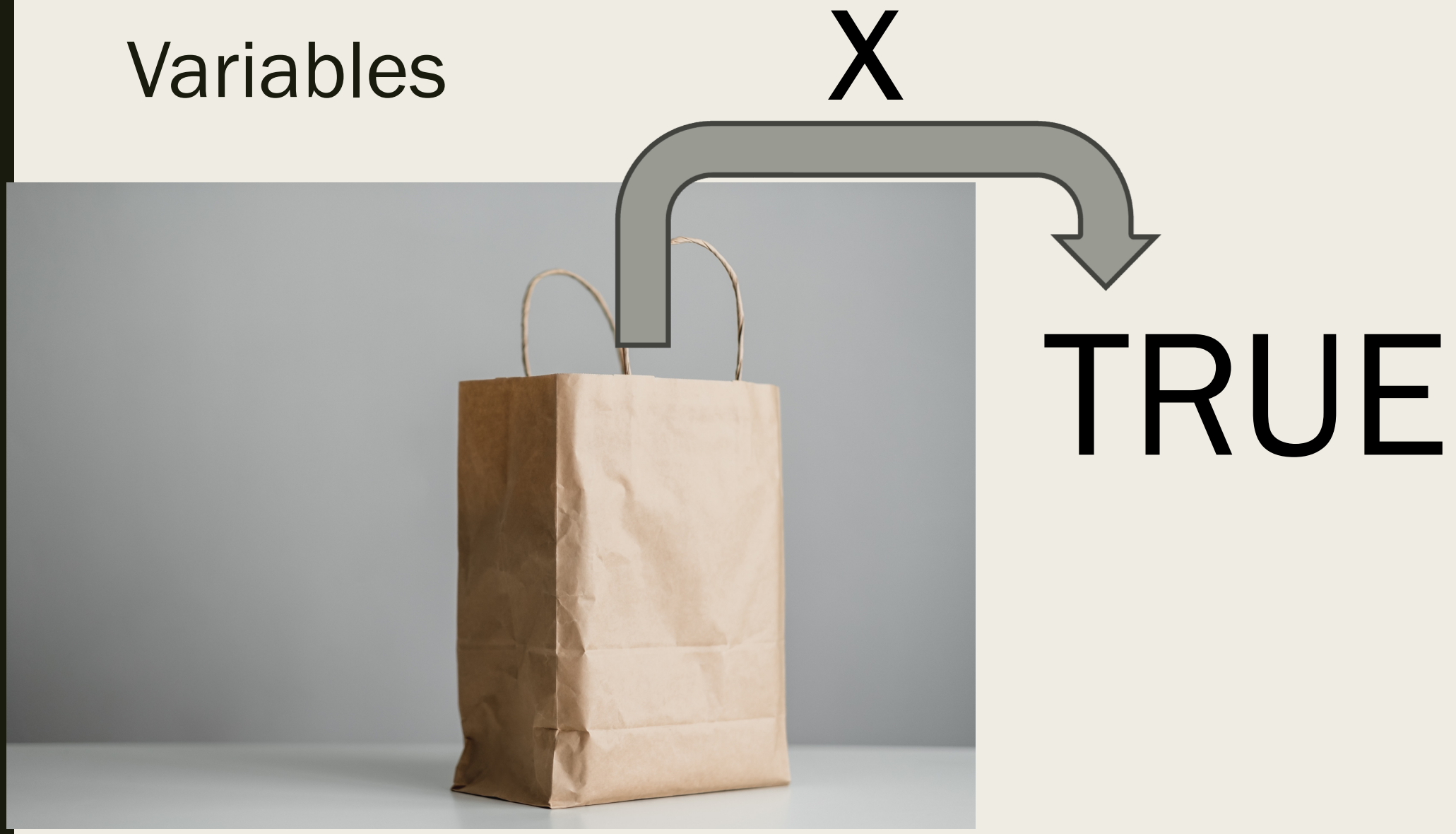
Variables

“dog”

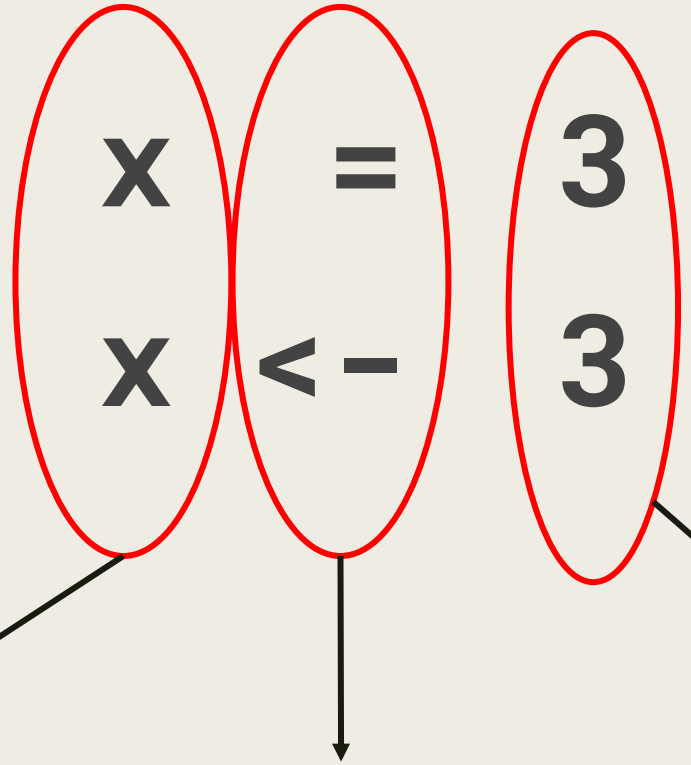
Variables

Variables

Variables



Variables



Variable name

"Left hand side"

Assignment operator

Value

"Right hand side"

Variables

```
x <- 3
```

```
<- is preferred in R
```

Variables

```
x <- 3 + 5

x
```

```
x is 8
```

Variables

```
x <- 3

x + 5

x
```

```
x is 3
```

Variables

```
x <- 3
x <- x + 5
x
```

```
x is 8
```

Variables

```
x <- 3
y <- 5
x <- x + y
x
```

```
x is 8
```

Variables

```
x <- 3
y <- 5
x <- x + y
y <- 7
x
```

x is 8


VECTORS AND LISTS

Storing Multiple Values Together

We combine them with c() into a vector

```
x <- c(2, 4, 7, 8)
```

Vectors may be called arrays or lists in other languages

Variables

c(2, 4, 7, 8)

Index the Values in a Vector

Indices in R start with 1

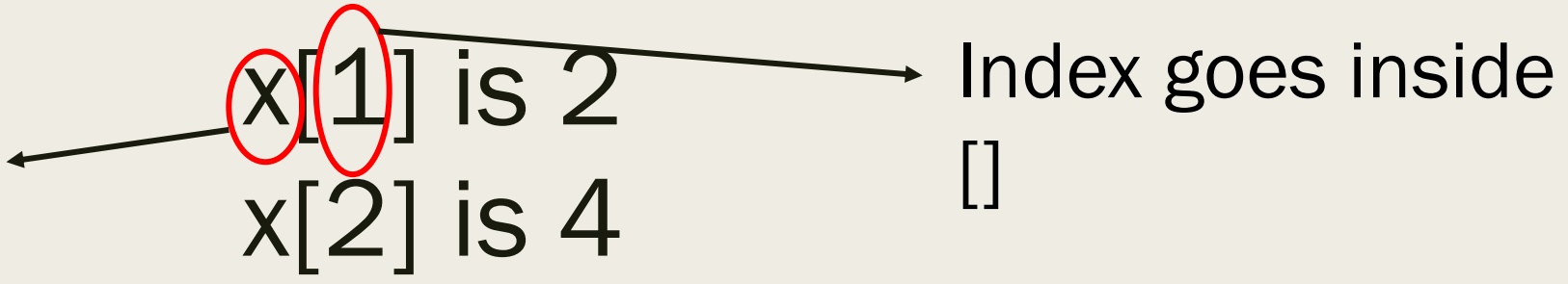
1 2 3 4

```
x <- c(2, 4, 7, 8)
```

Index the Values in a Vector

Indices in R start with 1

```
x <- c(2, 4, 7, 8)
```



Changing values in a vector

```
x <- c(2, 4, 7, 8)
x[2] <- 6
x
```

2, 6, 7, 8

Changing a variable

```
x <- c(2, 4, 7, 8)
x <- 6
x
```

6

R doesn’t care what’s in the variable “bag”!


BACK TO RSTUDIO…


FUNCTIONS

Functions

- Units of code
- Take input values (**arguments**)
- Can **return** an output value (e.g. a **return value**)
- R functions usually don't alter their input values (but they sometimes do!)

Function Definitions

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Function Definitions: Function Name

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Function Definitions: Parameters

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Function Definitions: Parameter Names

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Function Definitions: Default Values

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Function Definitions: Required Parameters

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Function Definitions:
Additional Undefined Parameters

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
        dec = ".", fill = TRUE, comment.char = "", ...)
```

Calling Functions: R: Named Arguments

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv(file="results.txt",
         header=TRUE,
         sep="\t")
```

Calling Functions: R: Named Arguments

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv(sep="\t”,
         file="results.txt”,
         header=TRUE)
```

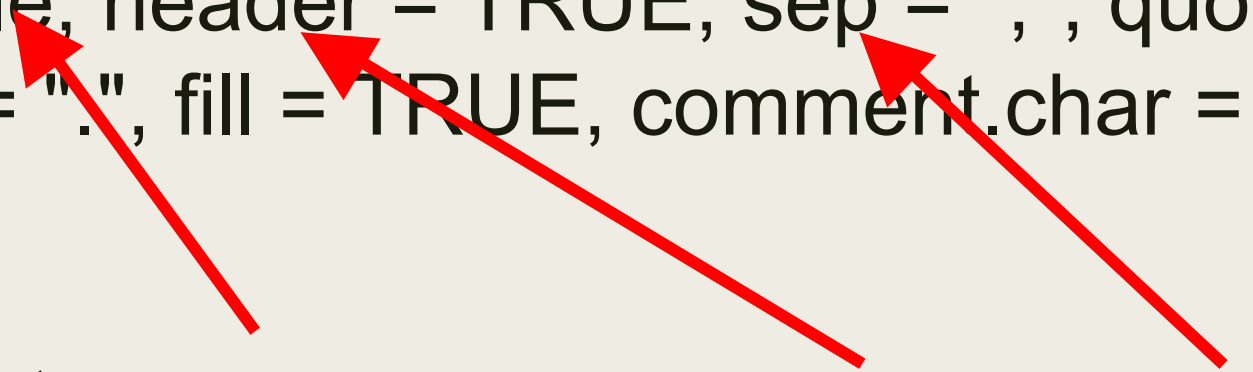
Calling Functions: R: Positional Arguments

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv("results.txt", TRUE, "\t")
```

Calling Functions: R: Positional Arguments

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)



```
read.csv("results.txt", TRUE, "\t")
```

Calling Functions: R: Positional and Named

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv(file="results.txt", TRUE, "\t")
```

Calling Functions: R: Positional and Named

1 2

read.csv(~~file~~, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

1 2

```
read.csv(~~file="results.txt"~~, TRUE, "\t")
```

Calling Functions: R: Positional and Named

1 2

read.csv(file, ~~header = TRUE~~, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

1 2

```
read.csv(header=TRUE, "results.txt", "\t")
```

Calling Functions: R

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Calling Functions: R: Which will work?

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv("results.txt", sep="\t")

read.csv("results.txt", "\t")

read.csv("results.txt", TRUE, "\t")
```

Calling Functions: R

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv("results.txt", sep="\t")
```

Calling Functions: R

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv("results.txt", sep="\t")

read.csv("results.txt", "\t")
```

Calling Functions: R

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv("results.txt",  sep="\t")

read.csv("results.txt",  "\t")
```

Calling Functions: R

read.csv(file, header = TRUE, sep = ",", quote = "\"",
dec = ".", fill = TRUE, comment.char = "", ...)

```
read.csv("results.txt", sep="\t")

read.csv("results.txt", "\t")

read.csv("results.txt", TRUE, "\t")
```

Calling Functions: Return Values

```
abs(-3)
```

```
Returns: 3
```

If we want to save the output:

```
x <- abs(-3)
```

Calling Functions: No Return Value

```
print("Hello World!")
```

>> Called for their side-effects (what they do)

```
x = print("Hello World!")
```

Calling Functions: Variable Arguments

```
x <- "Hello World!"

print(x)


file <- "results.csv"

read.csv(file = file)
```

BACK TO RSTUDIO...

Open script4.R

FILE SYSTEMS

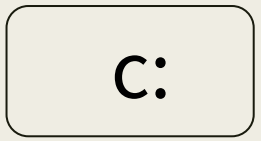
Where does R think it is?

File Paths

How do we tell R where our files are without using point and click interfaces?

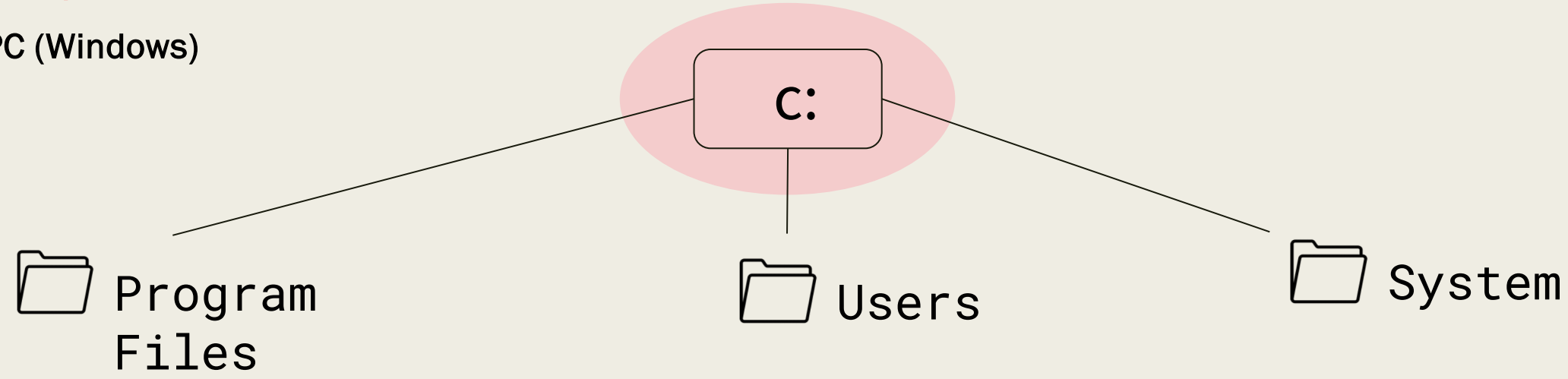
Every file has an absolute path, which starts with the root

PC (Windows)



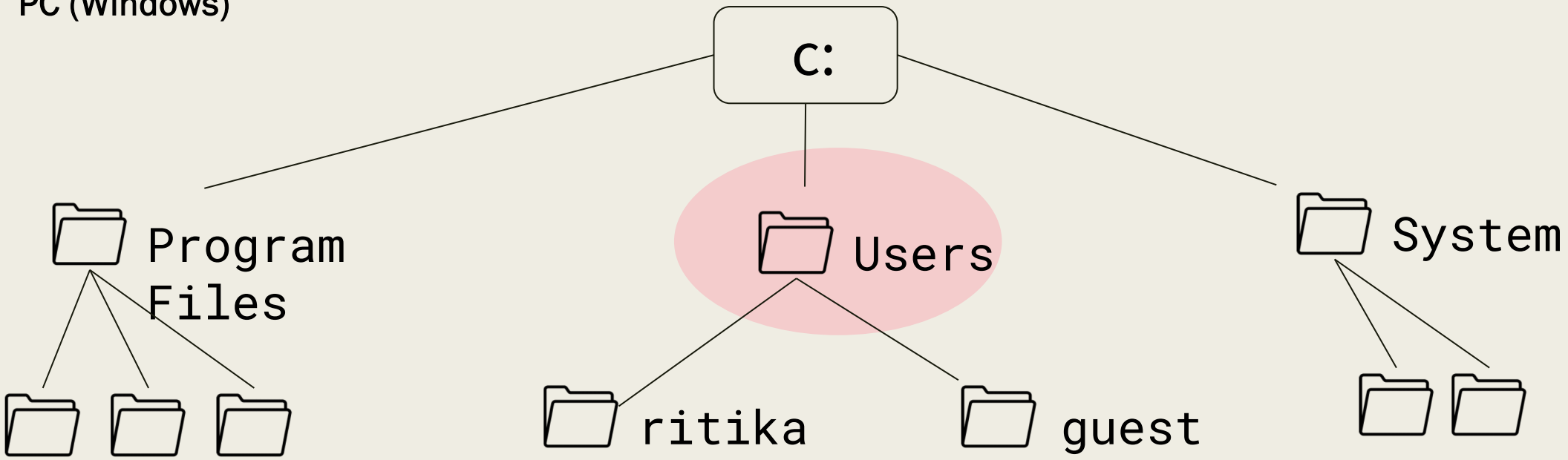
C:\

PC (Windows)



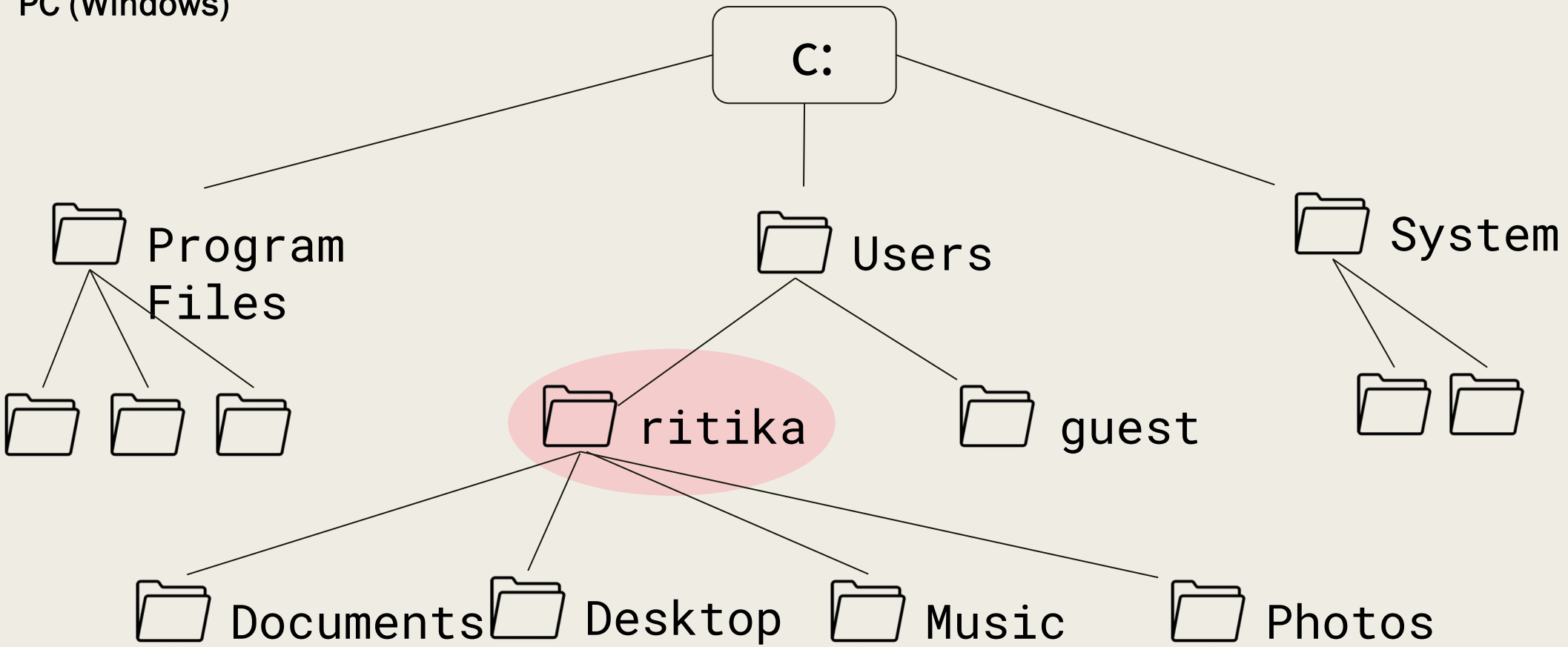
C:\Users\

PC (Windows)



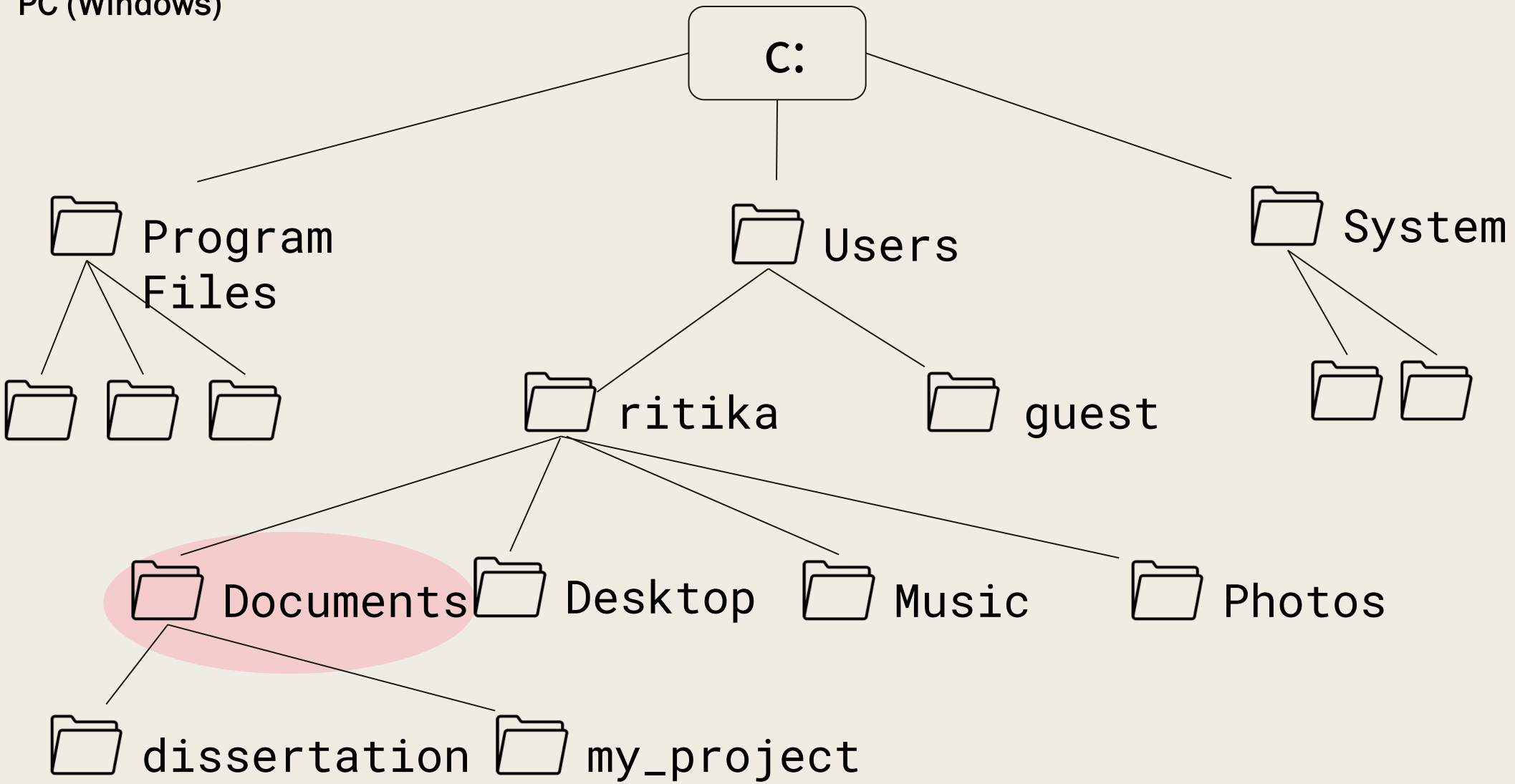
C:\Users\ritika\

PC (Windows)



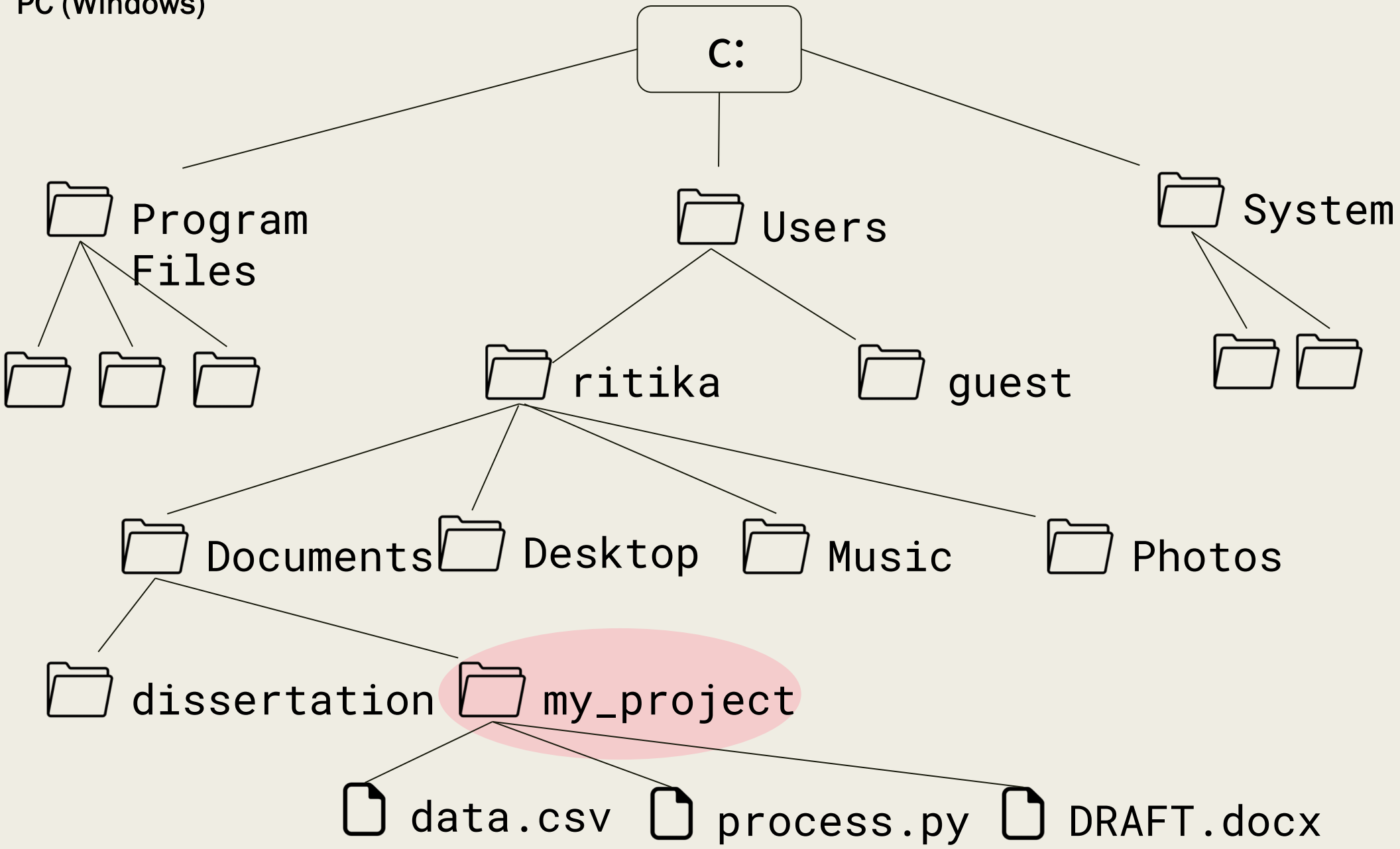
C:\Users\ritika\Documents\

PC (Windows)

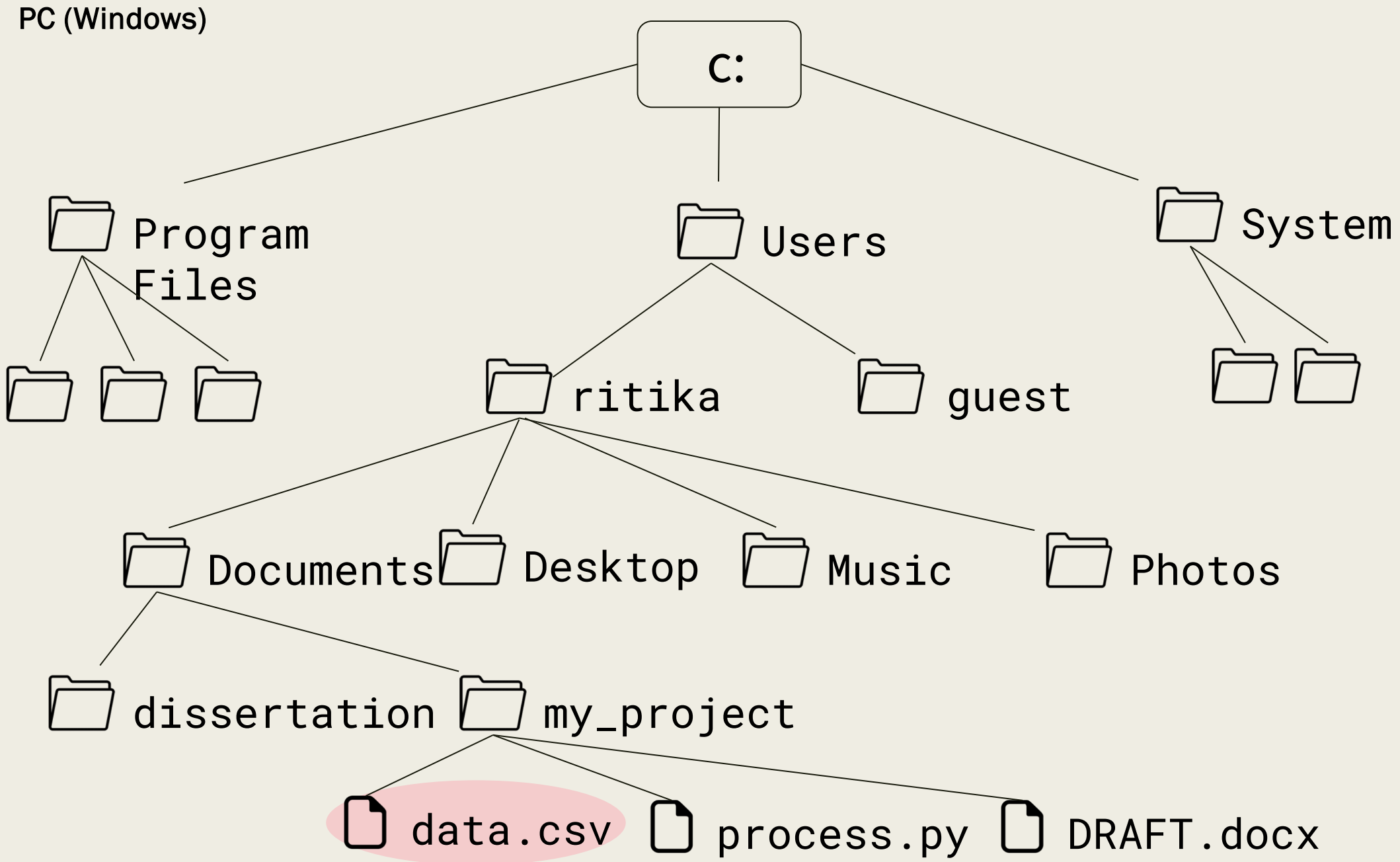


C:\Users\ritika\Documents\my_project

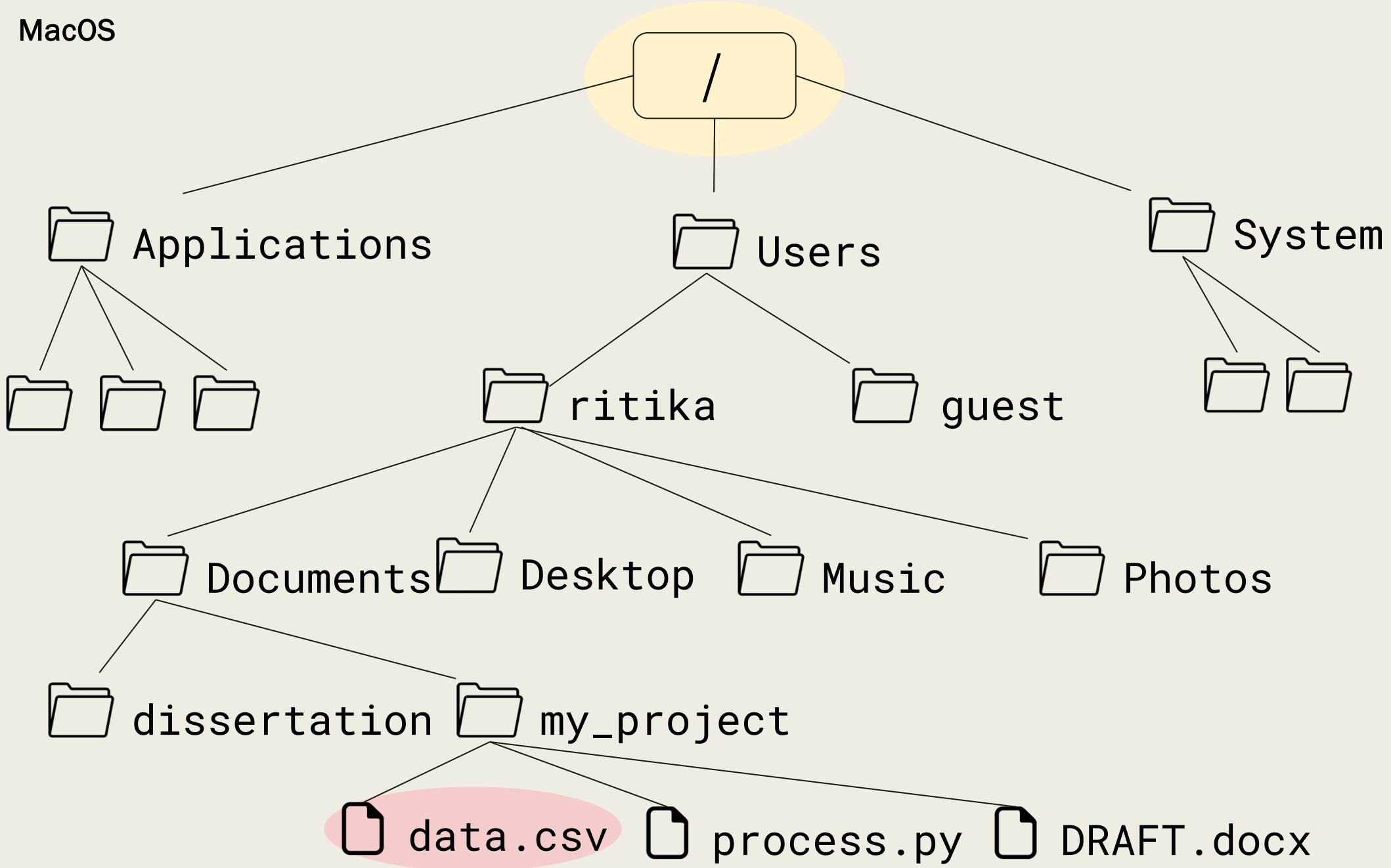
PC (Windows)



C:\Users\ritika\Documents\my_project\data.csv



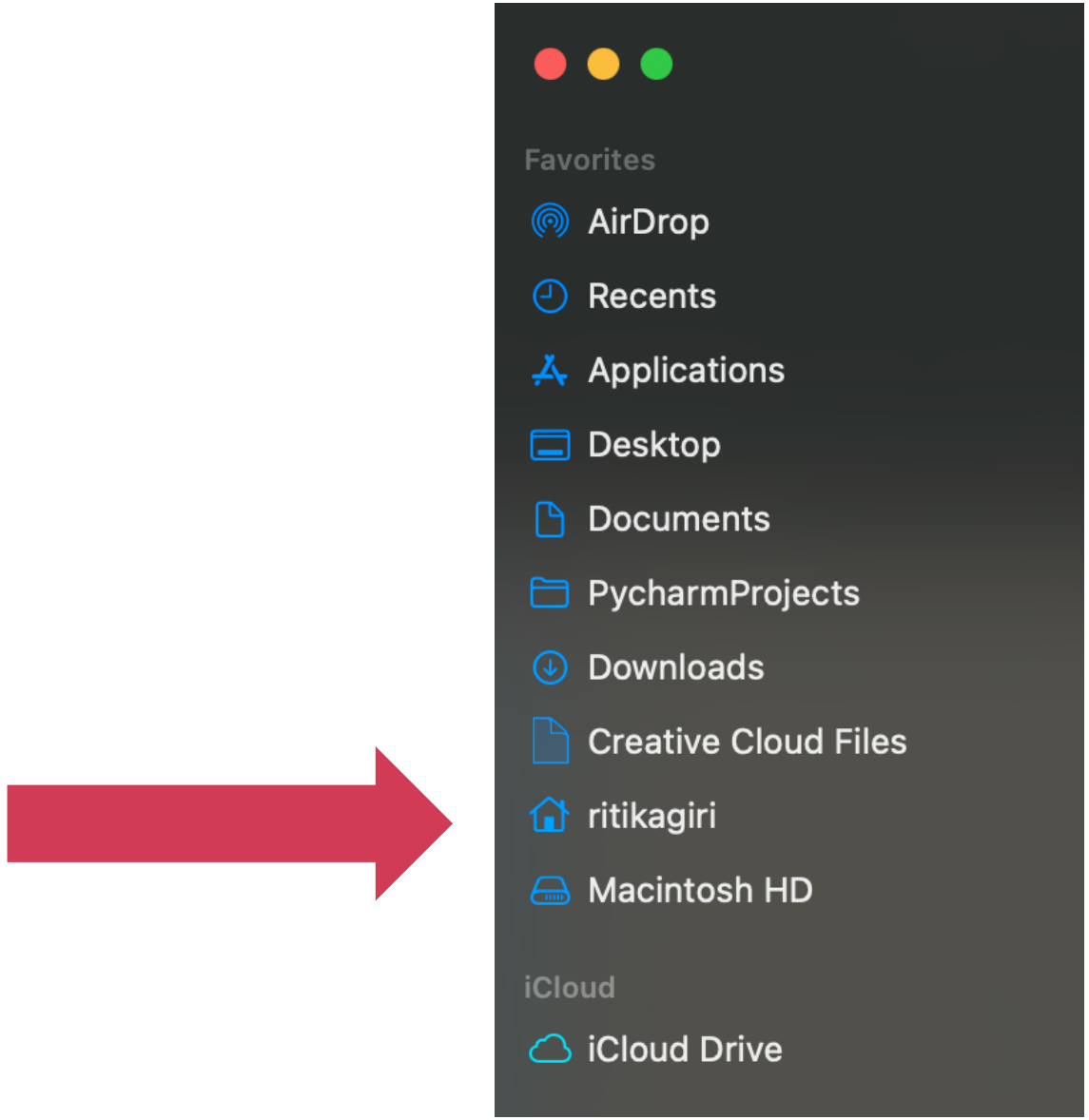
/Users/ritika/Documents/my_project/data.csv



Absolute Paths

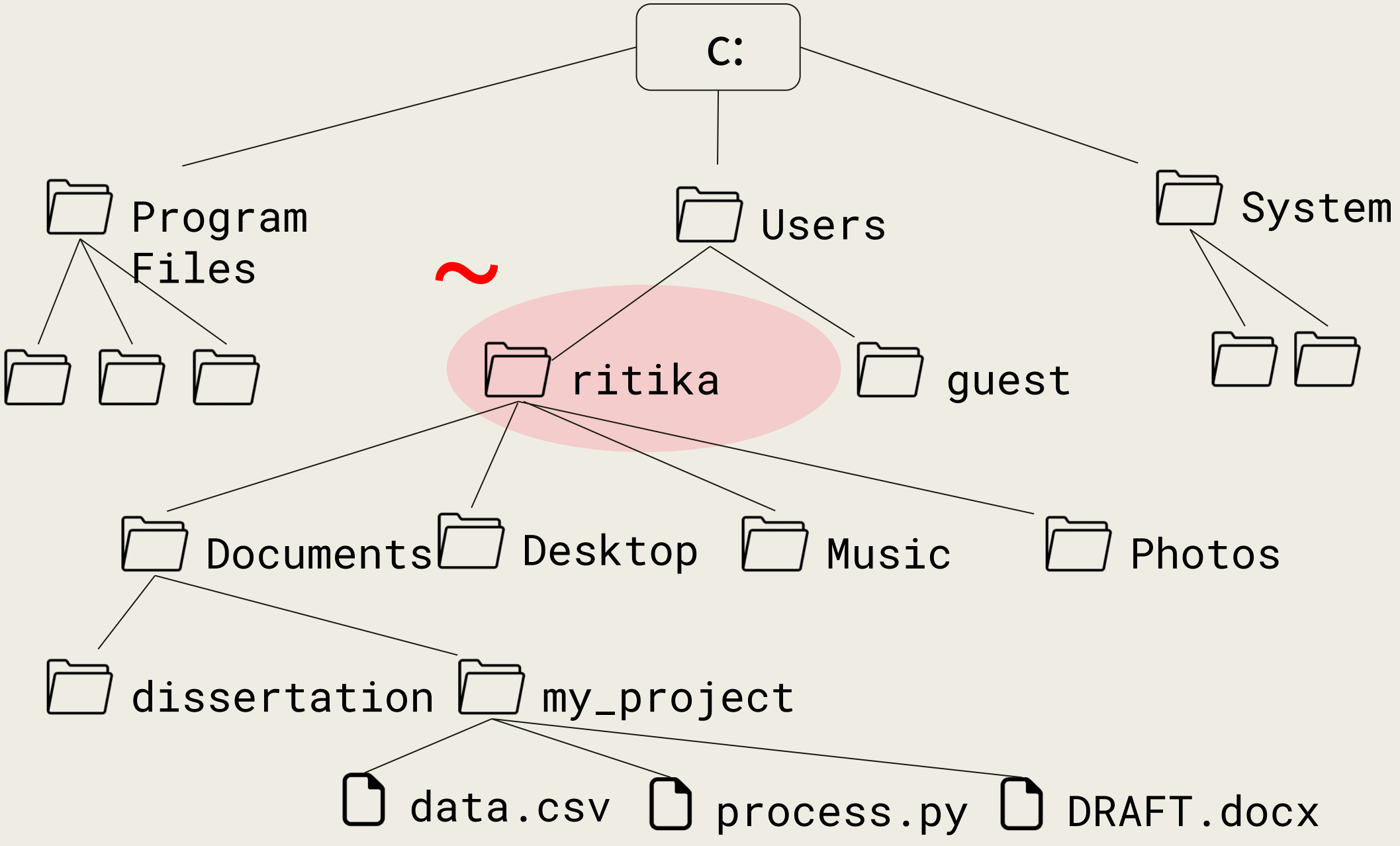
```
C:\Users\ritika\Documents\my_project\data.csv

  /Users/ritika/Documents/my_project/data.csv
```

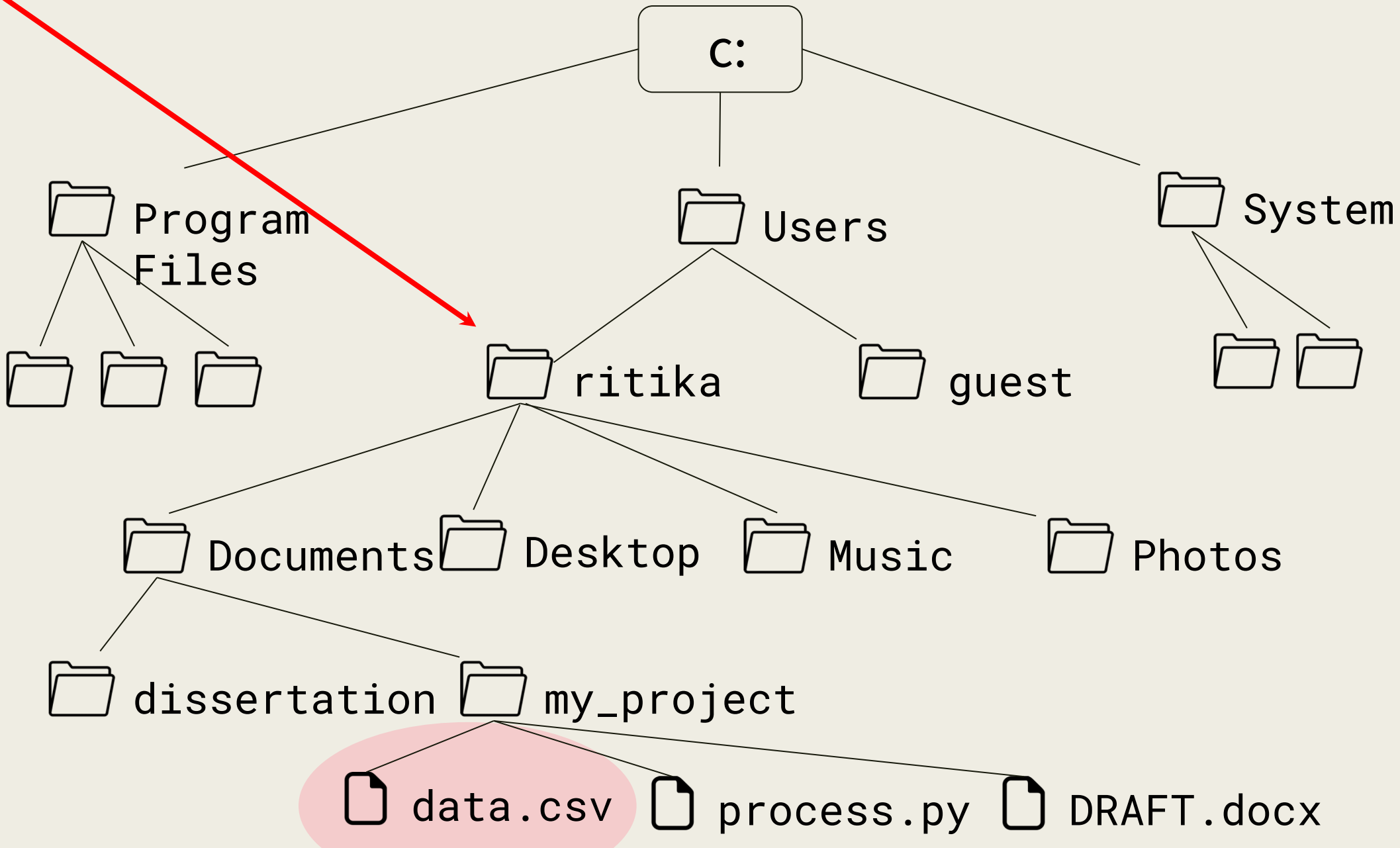


Home Directory

Home Directory: C:\Users\ritika = ~



~/Documents/my_project/data.csv



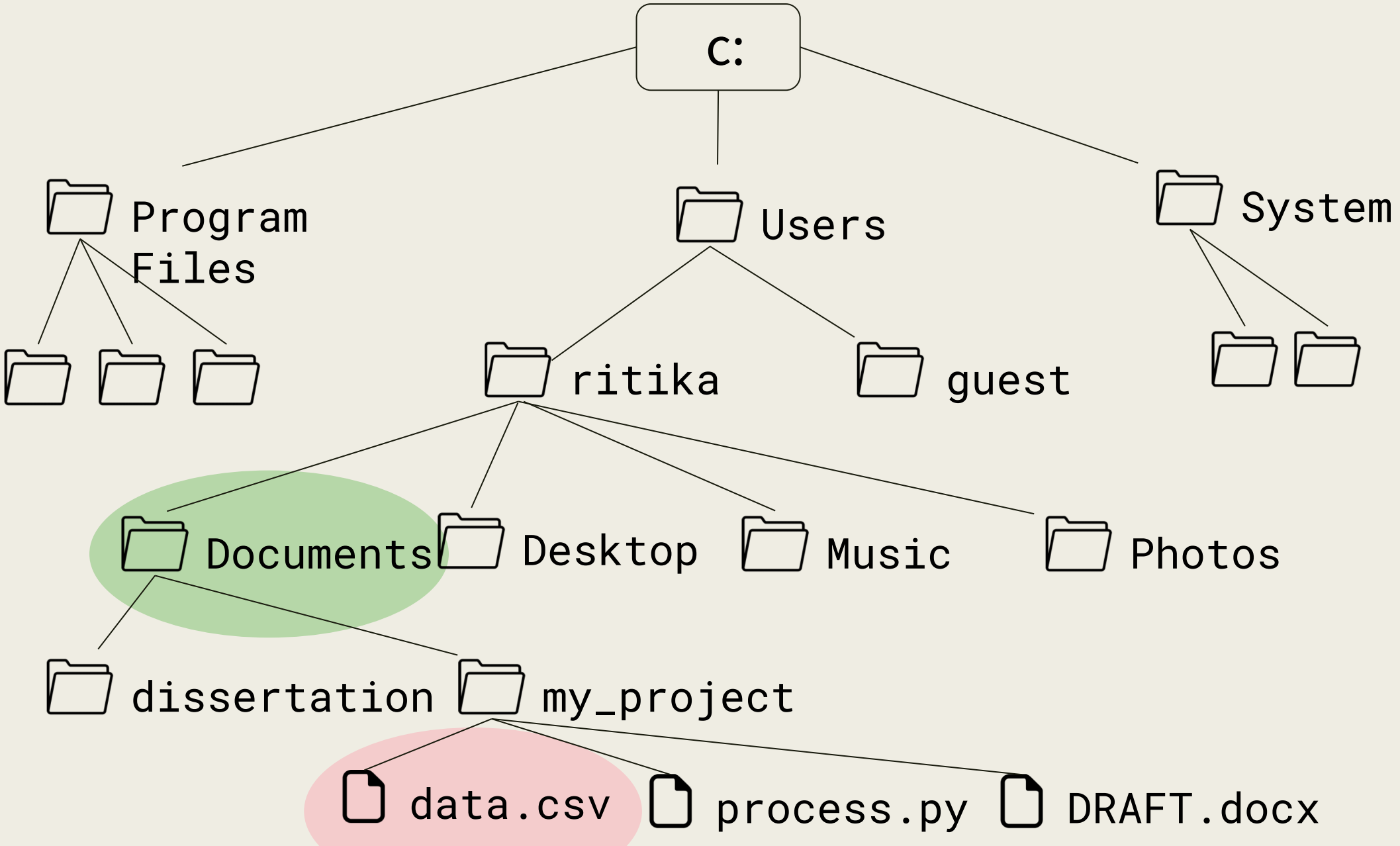
Working Directory

A directory in the filesystem associated with a running process or program

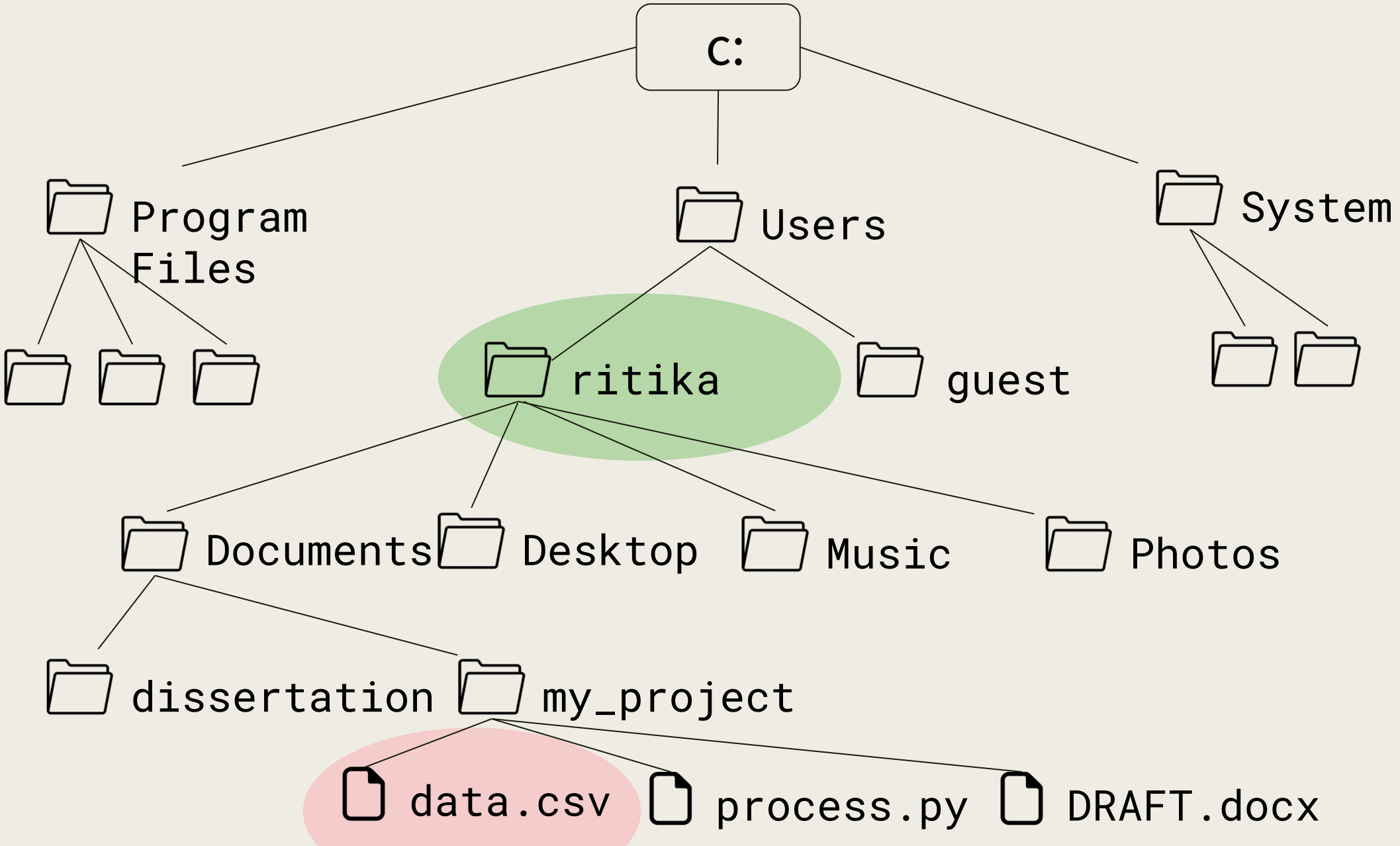
>> Where the computer starts when looking for files

>> Relative File Paths

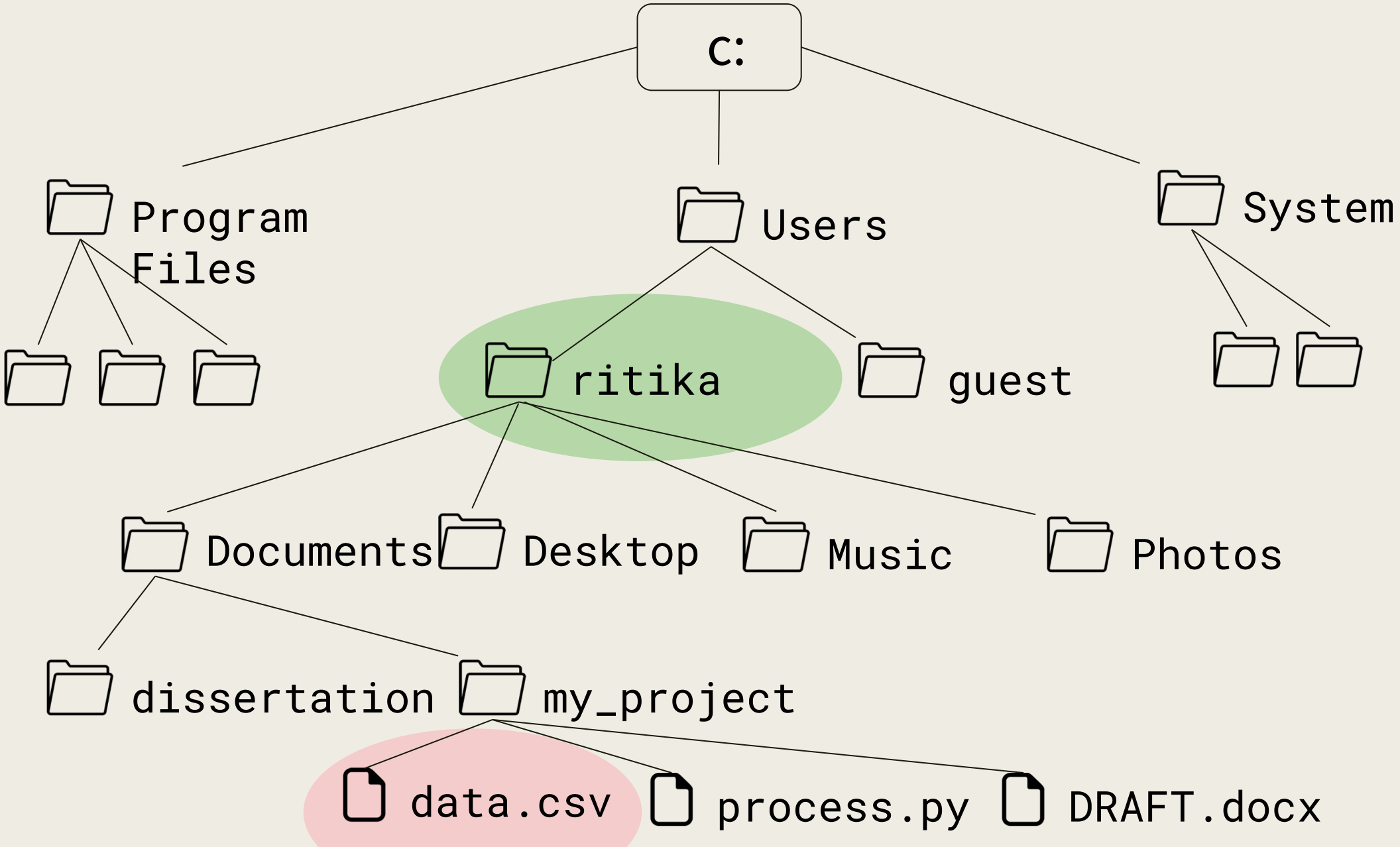
Relative Path from Documents: my_project/data.csv



From ritika:

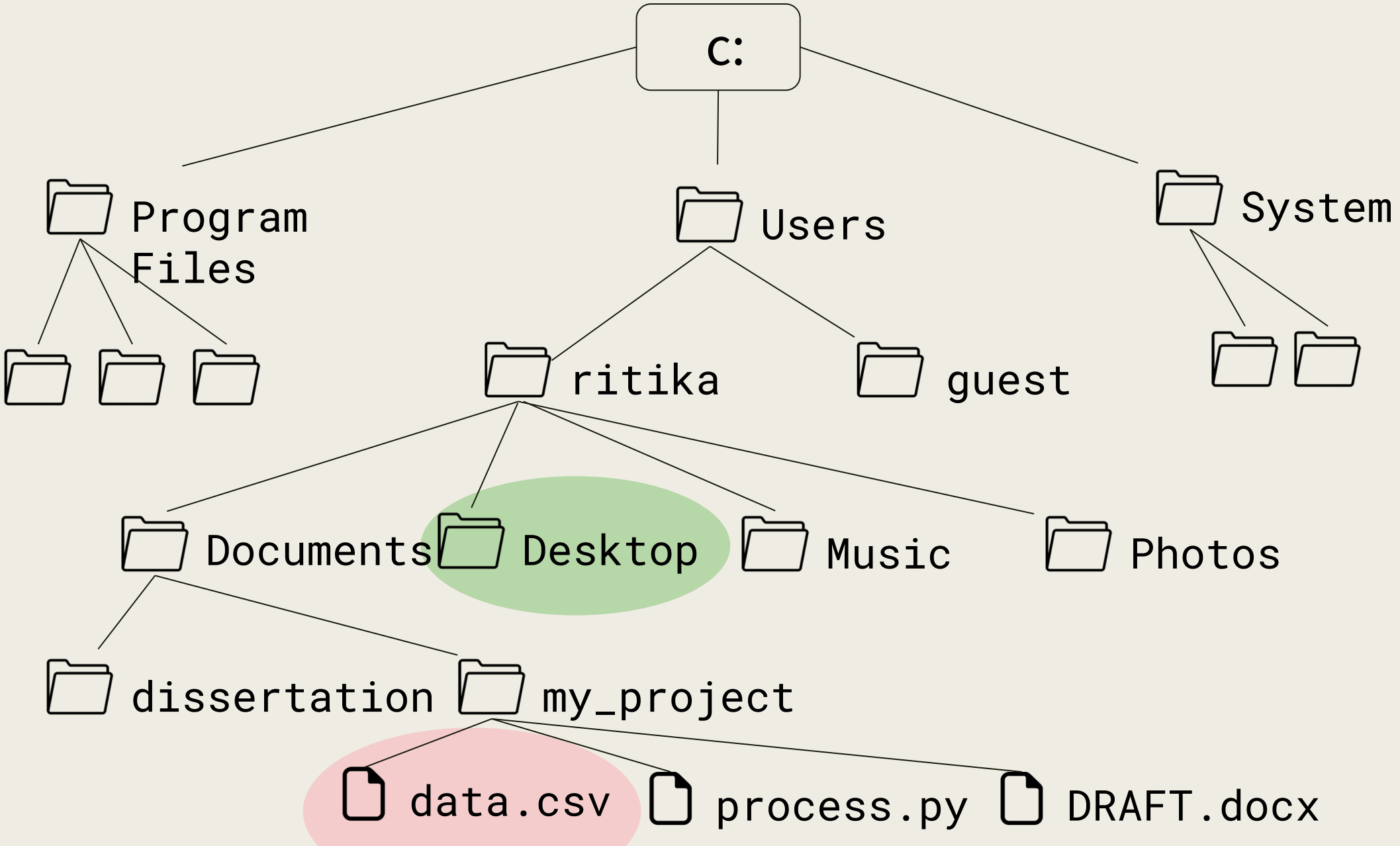

C:
Program Files
Users
System
ritika
guest
Documents
Desktop
Music
Photos
dissertation
my_project
data.csv
process.py
DRAFT.docx

From ritika: Documents/my_project/data.csv

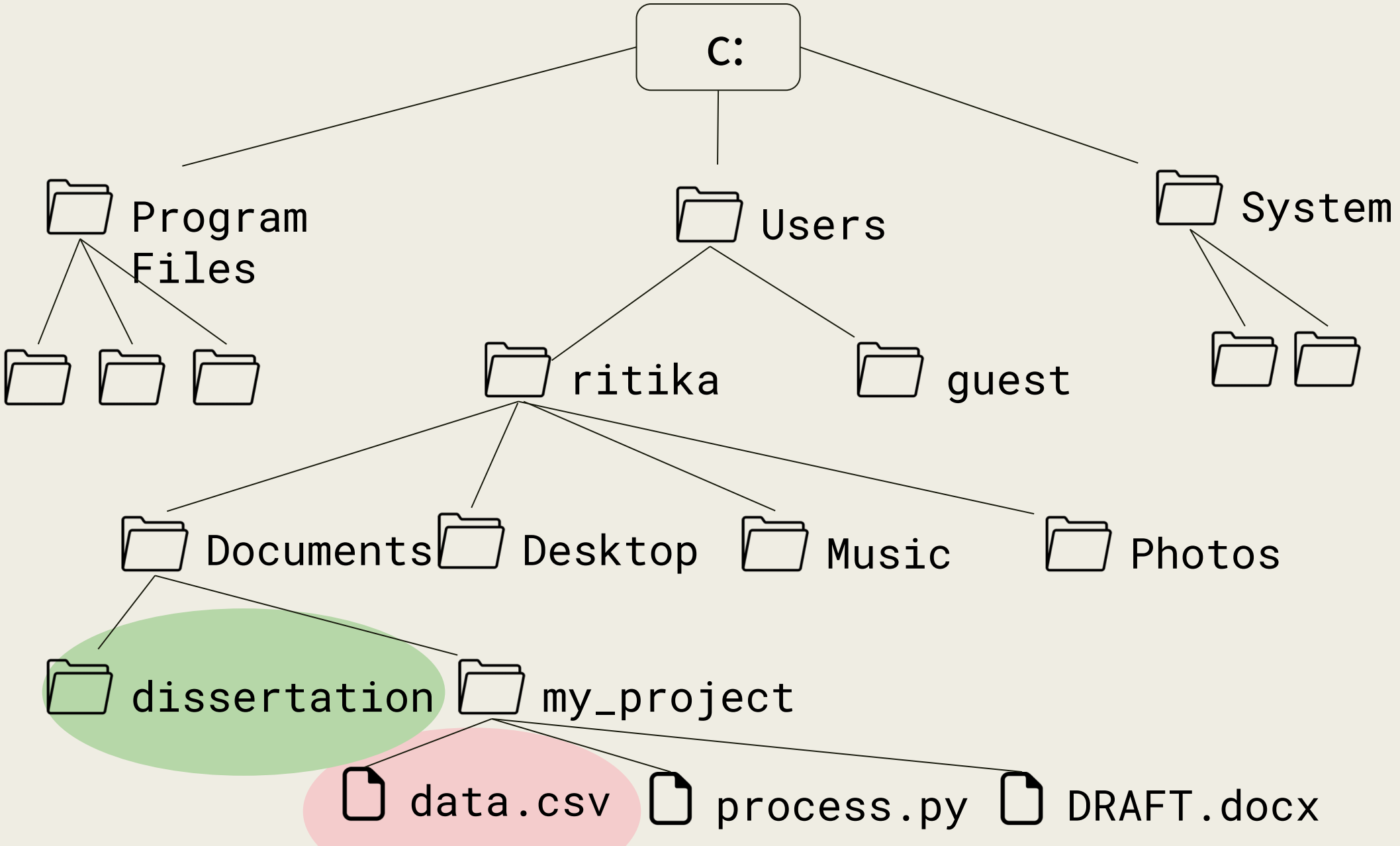


.. MEANS GO UP A LEVEL

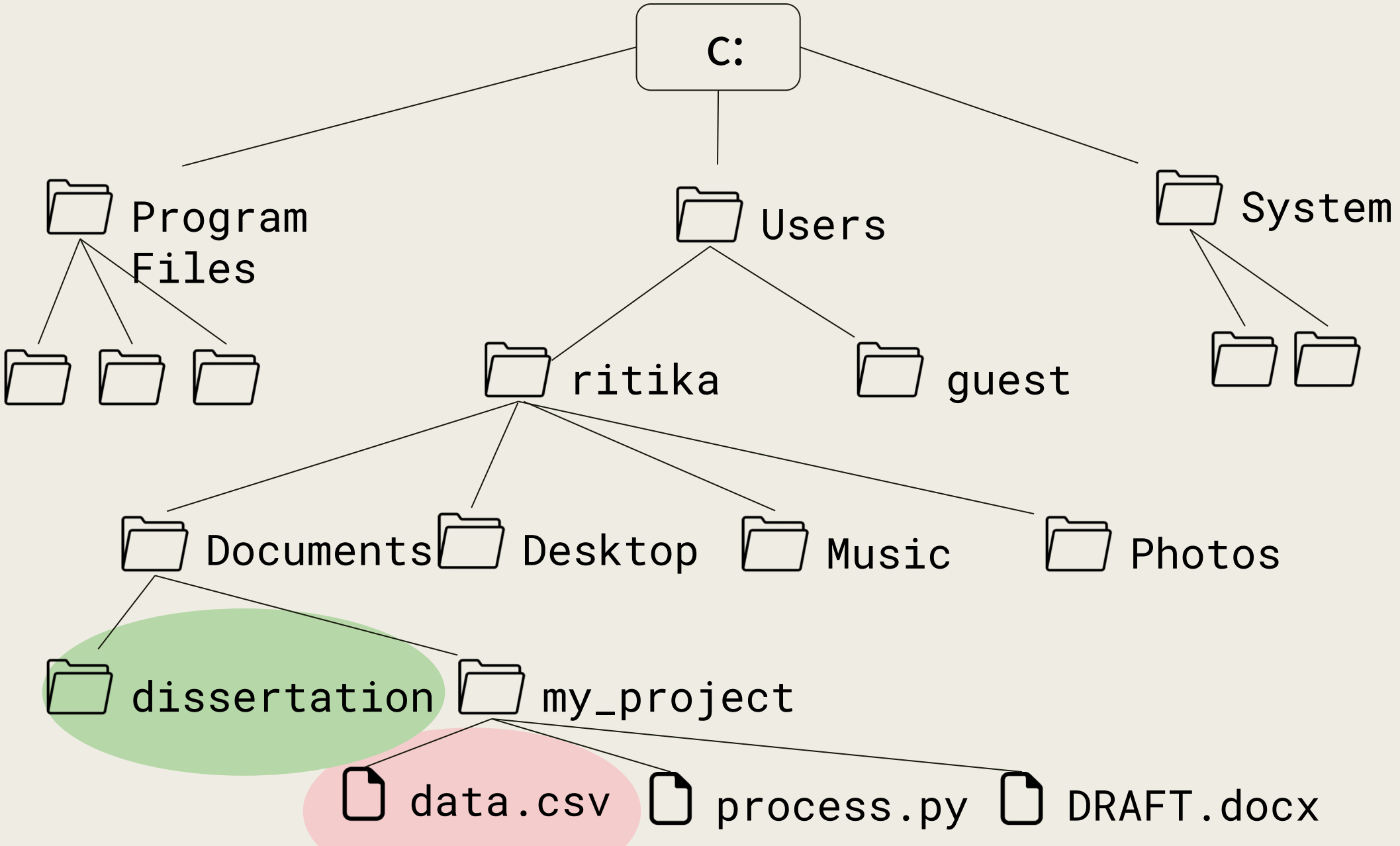
From Desktop: ../Documents/my_project/data.csv



From dissertation: ../Documents/my_project/data.csv



From dissertation: ../my_project/data.csv



But what is the Working Directory for R?

- Default: home directory
- RStudio Projects: the folder you associate with the project

>> You can set or change the working directory

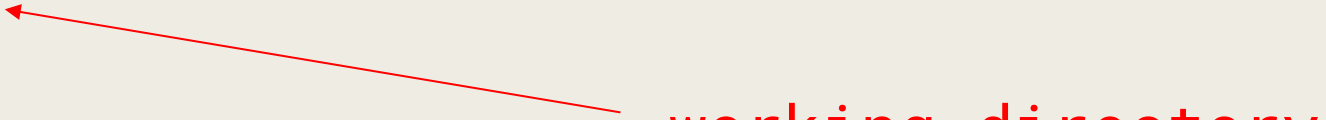
Let's go to R for a minute…

WHY NOT JUST USE ABSOLUTE PATHS?

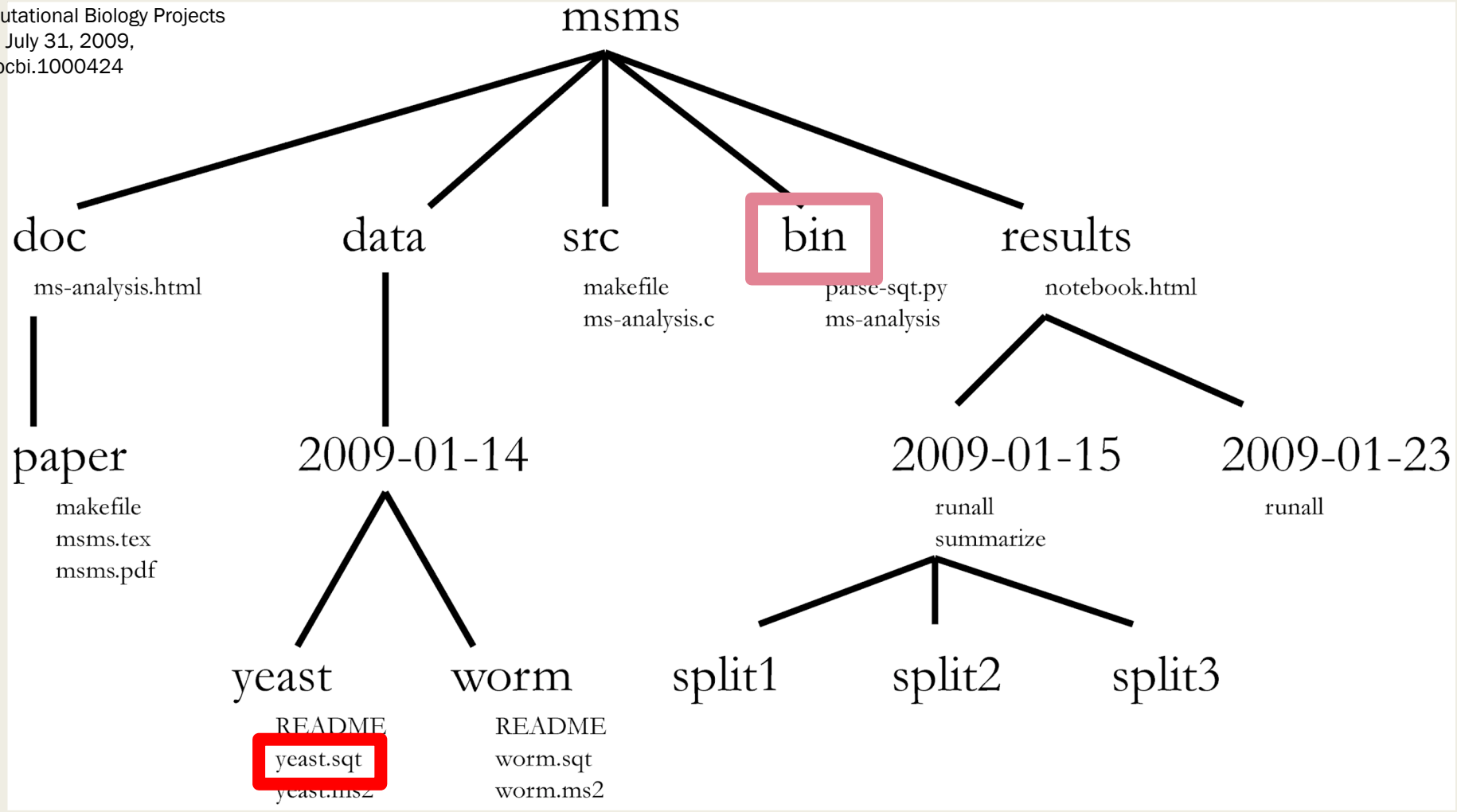
Store Project Files Together

project_directory

- code
  - *process_data.r*
  - *make_graphs.r*
  - *estimate_model.r*
- data
  - *people.csv*
  - *exam1.csv*
- output
  - *impact_plot.pdf*
- reports
  - *chapter1.docx*



A Quick Guide to Organizing Computational Biology Projects
William Stafford Noble, Published: July 31, 2009,
https://doi.org/10.1371/journal.pcbi.1000424



A) msms/data/2009-01-14/yeast/yeast.sqt
B) ../data/2009-01-14/yeast/yeast.sqt
C) ~/data/2009-01-14/yeast/yeast.sqt

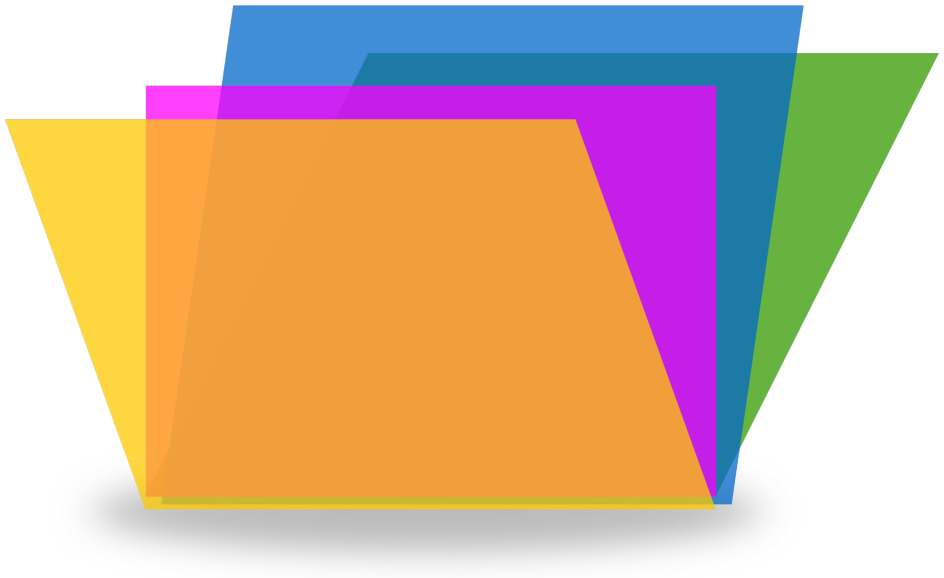
BACK TO RSTUDIO...

Open script5.R

WORKING WITH FILES

Reading and writing, Types

File Access Modes



Read: Open a file to get the contents

Write: Open a file, empty it!!!, and then write new data

Append: Open a file and write new data to the end of it

File Types

Text: Restricted set of characters
>> Data can be opened and viewed directly

Binary: Custom data
>> Needs a program to interpret and display the data

Plain Text Files

NO FORMATTING
NO IMAGES

Common extensions: .txt, .tab, .csv

Also plain text:

Data files: XML, JSON

Markup: HTML, Markdown (.md), LaTeX (.tex)

Code: R scripts (.r), Python scripts (.py)

```
# R Workshops

This repository is a clearing house for resources for individual R workshops from [Research Computing Services](http://www.it.northwestern.

# Summer 2023

[Chicago Sept 5-7](https://github.com/nuitrcs/R-bootcamp-summer-Chicago-2023)

[R Programming Concepts for Complete Beginners](https://github.com/nuitrcs/programmingConceptsR)

[R Introductory Bootcamp](https://github.com/nuitrcs/R-Bootcamp-summer-2023)

[R Intro to Tidyverse]()https://github.com/nuitrcs/R-intro-tidyverse-2023

[R Intermediate Tidyverse](https://github.com/nuitrcs/R-intermediate-tidyverse-2023)

# Past Workshops

[R Fundamentals Series](https://github.com/nuitrcs/r-intro-series): the most recently taught materials

[`ggplot2`](https://github.com/nuitrcs/r-ggplot2-april2020)

[Tidyverse](https://github.com/nuitrcs/r-tidyverse)

[Databases](https://github.com/nuitrcs/databases_workshop/tree/master/r): Information on how to connect to databases from R is part of the
that repository for more details.

[R Shiny](https://github.com/nuitrcs/rshiny)

[R Markdown](https://github.com/nuitrcs/rmarkdown_workshop): this one is a little older than the others, but the material should still be r

[Webscraping with rvest](https://github.com/turnerdan/rvest_tutorial)

[Statistical Models](https://github.com/aarcher07/stats_models)
```

https://raw.githubusercontent.com/nuitrcs/rworkshops/master/README.md

```
ATOM    324  NE2 GLN A  39     -26.856  16.418  34.025  1.00 32.16           N
ATOM    325  N  AARG A  40     -30.544  11.873  35.027  0.50 40.05           N
ATOM    326  CA AARG A  40     -31.291  11.740  36.269  0.50 41.18           C
ATOM    327  C  AARG A  40     -30.927  12.911  37.168  0.50 42.04           C
ATOM    328  O  AARG A  40     -29.815  13.433  37.066  0.50 41.86           O
ATOM    329  CB AARG A  40     -30.938  10.439  36.985  0.50 41.31           C
ATOM    330  CG AARG A  40     -31.407   9.172  36.309  0.50 41.88           C
ATOM    331  CD AARG A  40     -30.774   7.994  37.009  0.50 43.62           C
ATOM    332  NE AARG A  40     -29.355   8.245  37.268  0.50 44.98           N
ATOM    333  CZ AARG A  40     -28.608   7.539  38.113  0.50 45.52           C
ATOM    334  NH1AARG A  40     -29.138   6.528  38.787  0.50 44.68           N
ATOM    335  NH2AARG A  40     -27.329   7.844  38.281  0.50 46.16           N
```

File: **molecule.pdb**

```
%PDF-1.5
%–‘≈ÿ
13 0 obj
<<
/Length 1063
/Filter /FlateDecode
>>
stream
x⁄≠VYo‹6¿~¯ØPÛD¿¿#¿¿•æ5i¿¥pÉ†›¢¿u¿–¿Ω+D+.t‘ıøÔêC≠•µú¿Õæ,Ø¯Ê˙fFØwWØfiÚ4bí
ôÛhw¿)¿)ñ“T®hWG¿ë¿ı®„D®åÏb¡»√  6åò!˛{¯À´∑,èJZÊ<w¢iîpEUY†‰¿{‹⁄¿û¿AjD¿íå1/<
˛ôòë¿èõÆjß⁄¿|è»ú-ê¿ßLÂQ¬¿-Ö@¯¿Œ“¿#‡©¿¶lÒè^[Ø»∂Fw¿∏Ixù0FÀ,˙
‡ü;Á«h^¶˘¶∏Ô¶£ÈõÍõbæ9Ë^W#Z¿•¥,
∏Ç¨ªdÁêÏ,Öî¿/√¿y)…?qñ¿›7˙∂=g´$’|}¿ÚË7¶¿¿†ƒØ?,~xb¨M±∫°q"≥å˛y0¿Ø¿¬pß’°^£©Òx¥¯?´m˛´1NòÁ¿$·•˜
çÕ~9üÚ(+¿öe¿}ª¥HÃΔW^h¿<Iëå†Üë¿◊_"ΔË¿Ÿí=F6+RÍj )˘’¿£{õìVw˚IÔQ¿B€¿fi∑å¿¥s¿=«ª!TI’Ë¿OV≈¿¿Ü©r¡8
¿°Pò\,È¿å,JQ)ãïeÔ˛∏æfi™ÅÕ/"e4/ÚïòÌ?°G0AãT≠ıÿŒ|Nè(h*≥ïÿx–!pMW7ï¿Cl∫¿5‰g;¿G%Yí›¡”Ù´"®{≥f¿‰./
hûghC›‹§Bòfi`ÕcŒÔz{DÜ¿õa¿ùMΣ«¨Ÿ$|¿Ī∂} ÀMö•ÜÓÈV˛JA%¿Á^◆√VißTÊÁ“¿∞@æÑß¿t%◊MÚ±ÙWm¿¨
}wÊü ¿!§‡fi}Ëÿzxé˘6Ã]fiH¿?˚µ¿*0¿
∑%¿V¨>ˆ¿≠≠¿í§Tdeò#%æfi≈¿òs;°Dµ√IWffiËÆΔEgÓq–6ùAk*ÁŒÏ¿¿æKµ_ò¿ıÒ‘Œ¿lÔûsv¿≤ï¥`±è:¿YaBv[π¿T¿|Œ]3
€¥°˚©ö$¿Øg (¿˘Ø"?¿¿©/EJ9¿Å7gj¿<I¿¿≠˘±E63e⁄”AáÓ8¿Á[°∂}mzflÈ¿π∂ÆW›c{¿¿÷CÄmÕ∏ë©œTflt
≈|¿'f8ØôCcà¿1¿ãŒôø(}…
 3Õ˛_¢ü˘÷b±ƒ\=Ø9ÿo¿DWœl¿ÒÇiâãSo¯Ω>ûßH°¿Ê§@0.ü§#'Gg¡‰∫]ËêãaÏΔBfl€©´qÑ∫K◊¯∞i~xÊÁCï∫o¿/Ú[hè]
Äx˘0∫ ⁄Œ¿ü"¿ÿ“¿Ñƒ¶êì⁄NÓK·íV8èEQPëóOÁÒE›1¿æ0¿_8¿ß”`jßú)r;ç¯¿¢¯Ä∑¿¿¿Σ”#¿Øü¶
í°o¿v¿Ú¿¿m¿¿¿k¿¿^>¥aã.¿»B;w¿'C3å¿?>æ˚ƒ¿Ñ°  ◆Yä¿}}1L∑/6ÍîóT*Ò%u∫
Î•N¿CCŒ¿N”‡Y≤•/S4¿Á∑ûoW?ÌÆ˛¿¥¿
endstream
endobj
27 0 obj
<<
```

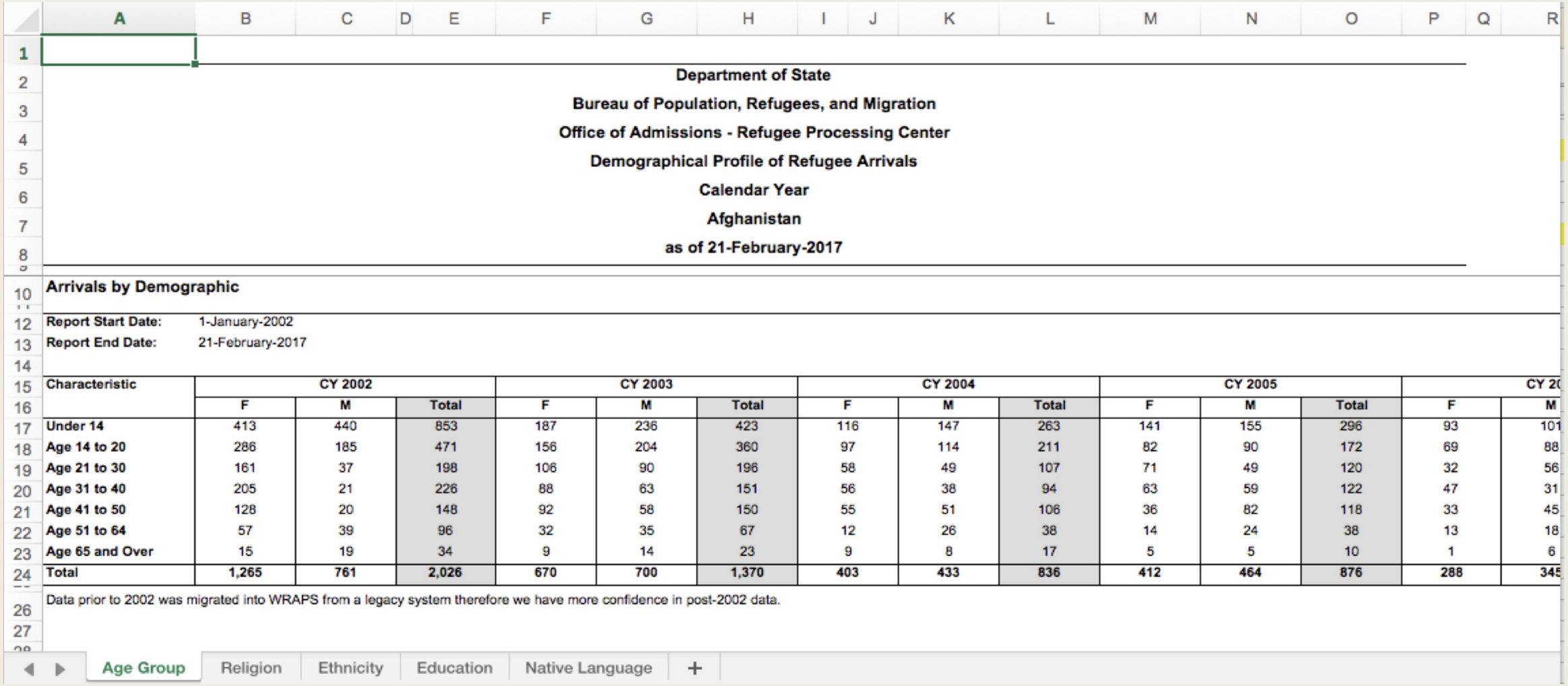
????JFIF??C

!"$"$??gg"??

???}!1AQa"q2??#B??R??$3br?
%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz??
????????

???w!1AQaq"2B????   #3R?br?
$4?%?&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz???
??????????
????co?Ͼ??n?jc???e??,?;???Dg|???h5   X?+lr?TNXs\??   ??h??l?˥???kꝁ?2?⌐f,KɣT?V
?*?\n??6?^???ɹł K??????U
X?q?.?_lp?Y???Ic?%J??M?eɎv?)??C??/#Z?[????mx_Ē??&bT+9P揄 ??IGs??#?s?$??Lz??/?=?@??=k~???ls???T??-D??皅 ?d℘??Z;Av9??
ZɯPT??k???T?OЏ??ў???ᵽ?al?V.??? t??x&t??+J??p?f2k)C??U?3?ӛvvd?X??????E?4G????[)Bz?}??PG?^?U??¡?<?Hk??A?Q?4u?2R???
KAX??;???qN??????I?B-X?Ƭ⍙h
(?[⊙1E??+7+??2i?碉 ?ɸ?I1
OZ2G5IXre?q??Y[???S??n?.=)?Ÿ?r|?3?R??F9?Ж'?O???W??n?n~??ɤ?yX?????r\?F?q???Ꝩ?K?]-"H?цZ??ca
O???gpA?mG`??a??x??`?X?^%*r?MM'cᶉk?Y???y??ϴ?Pm*rk?P?lL]?{yUA???b?+?3??+kK?x??L???s?ЧGWe`??"?:??Q):???3[Fź??.!r??
?fkzBIvAPA>????X?e?h\U??p?$#&??~K?Ã¿??????V:{??b
???U?.?B=h?%&??dhGtX???\~?`?;J?w?+?C?|??????m? ?{P??vrBv??_DB1????_H
?r????h??z?t???꞉)??l?qɭlYW??g]B?H<b?Z?9+]N?
???ц?P?????F,?{?i?#?x??w??5???&Ui<zTdʒE?:ɥ??B$?????{Z?/??
?ɹ?z?g

Is this a plain text file?



Department of State

Bureau of Population, Refugees, and Migration

Office of Admissions - Refugee Processing Center

Demographical Profile of Refugee Arrivals

Calendar Year

Afghanistan

as of 21-February-2017

Arrivals by Demographic

Report Start Date: 1-January-2002

Report End Date: 21-February-2017

Characteristic	CY 2002			CY 2003			CY 2004			CY 2005			CY 20	
	F	M	Total	F	M	Total	F	M	Total	F	M	Total	F	M
Under 14	413	440	853	187	236	423	116	147	263	141	155	296	93	101
Age 14 to 20	286	185	471	156	204	360	97	114	211	82	90	172	69	88
Age 21 to 30	161	37	198	106	90	196	58	49	107	71	49	120	32	56
Age 31 to 40	205	21	226	88	63	151	56	38	94	63	59	122	47	31
Age 41 to 50	128	20	148	92	58	150	55	51	106	36	82	118	33	45
Age 51 to 64	57	39	96	32	35	67	12	26	38	14	24	38	13	18
Age 65 and Over	15	19	34	9	14	23	9	8	17	5	5	10	1	6
Total	**1,265**	**761**	**2,026**	**670**	**700**	**1,370**	**403**	**433**	**836**	**412**	**464**	**876**	**288**	**345**

Data prior to 2002 was migrated into WRAPS from a legacy system therefore we have more confidence in post-2002 data.

Age Group | Religion | Ethnicity | Education | Native Language | +

,,,
Department of State,,,
"Bureau of Population, Refugees, and
Migration",,,
Office of Admissions - Refugee Processing
Center,,,
Demographical Profile of Refugee Arrivals,,,
Calendar Year,,,
Afghanistan,,,
as of 21-February-2017,,,
,,,
Arrivals by Demographic,,,
,,,
Report Start Date:,1-January-2002,,
Report End Date:,21-February-2017,,
,,,
Characteristic,CY 2002,,,,CY 2003,,,CY 2004,,,,CY 2005,,,CY 2006,,,,CY 2007,,,CY 2008,,,CY
2009,,,CY 2010,,,CY 2011,,,CY 2012,,,CY 2013,,,CY 2014,,,CY 2015,,,CY 2016,,,CY 2017,,,Cumulative
Total,%
,F ,M ,Total,,F ,M ,Total,F ,,M ,Total,F ,M ,Total,F ,,M ,Total,F ,M ,Total,F ,M ,Total,F ,M
,Total,F ,M ,Total,F ,M ,Total,F ,M ,Total,F ,M ,Total,F ,M ,Total,F ,M ,Total,F ,M ,Total,F ,M
,Total,,Total
Under
14,413,440,853,,187,236,423,116,,147,263,141,155,296,93,,101,194,65,55,120,78,85,163,55,51,106,75,7
8,153,44,61,105,83,96,179,84,106,190,102,100,202,145,190,335,464,447,911,54,76,130,"4,623",31.96%
Age 14 to
20,286,185,471,,156,204,360,97,,114,211,82,90,172,69,,88,157,34,55,89,63,66,129,29,43,72,48,68,116,
47,38,85,56,79,135,63,92,155,97,87,184,120,197,317,311,385,696,36,58,94,"3,443",23.80%
Age 21 to
30,161,37,198,,106,90,196,58,,49,107,71,49,120,32,,56,88,39,24,63,50,42,92,24,32,56,45,67,112,46,37
,83,41,48,89,66,64,130,82,52,134,119,91,210,303,277,580,39,25,64,"2,322",16.05%
Age 31 to

Plain Text Editors

RStudio lets you write (or open) plain text files (so do other IDEs)

Stand alone options:

- Mac: BBEdit
- Windows: Notepad ++
- Either: Sublime Text

Usually avoid built-in options: Notepad, TextEdit, WordPad

R MARKDOWN/QUARTO

Back to RStudio...


DATA FRAMES
Rectangles of Data

Rectangles of Data

Rows: Observations

Person (or mouse, worm, etc.)
Country
Year
Run/trial of an experiment
Chemical
Sample

Columns: Variables

Measurements

Grouping/identification variables:

> trial/sample

> condition

> label for the observation (country name)

Each ID variable in its own column

The $ operator

Access columns by names as:

```
data$column
```

Indexing

Access rows or columns by indexing as:

```
data [ rows , columns ]
```

Indexing : by POSITION

Provide the positions of the rows/cols you want:

```
data [ 1:10 , 2:5 ]
data [      , 2:5 ]
data [ 1:10 ,     ]
data [c(1,4) , c(3,5)]
```

Indexing : by POSITION

Provide the positions of the rows/cols you don't want:

```
data [-1:-10, ]
```

Indexing : by NAME

Provide the names of the columns you want:

```
data [ , "school" ]
data [ , c("school", "year") ]
```

Indexing : by CONDITION

Filter rows by the conditions you want:

```
data[data$column == value, ]
```

Indexing : by CONDITION

Filter rows by the conditions you want:

```
data[data$column == value, ]
data[data$column > value, ]
data[data$column < value, ]
```

Indexing : by multiple CONDITIONs

Filter rows by combining conditions you want:

AND:

```
data[data$col1 >= x & data$col2 >= y , ]
```

OR:

```
data[data$col1 >= x | data$col2 >= y , ]
```

Missing values NA

Missing values are represented by a special value NA

```
is.na(x) - returns TRUE or FALSE
```

BACK TO RSTUDIO...

Open script6.R


WORKFLOW
Organizing coding projects..

Data Analysis Workflow