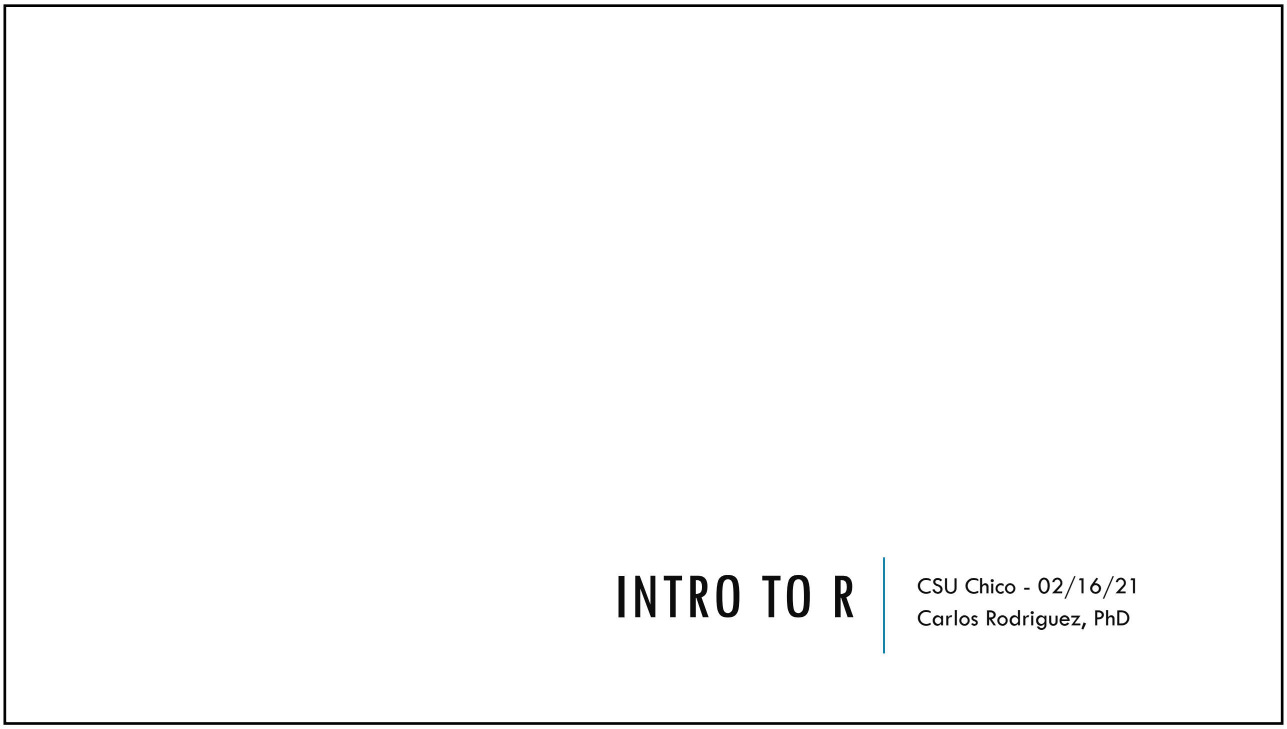
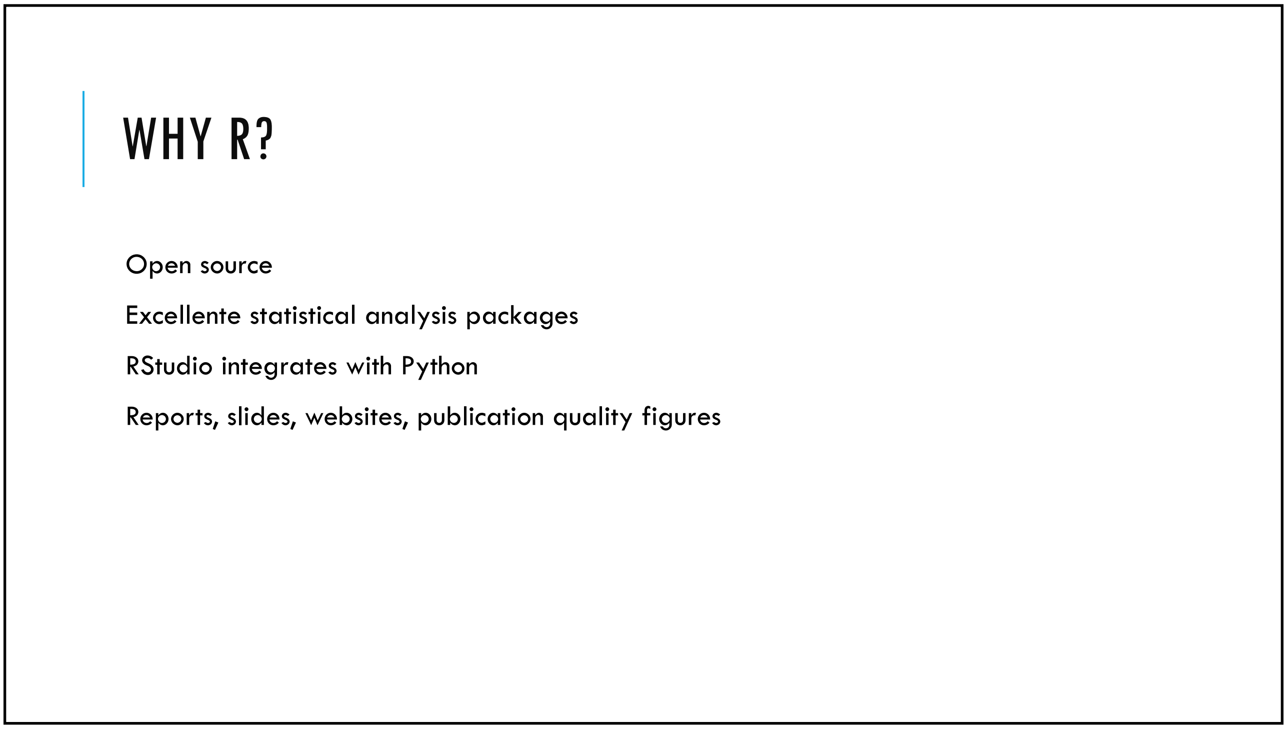


INTRO TO R

CSU Chico - 02/16/21
Carlos Rodriguez, PhD

1



WHY R?

Open source

Excellente statistical analysis packages

RStudio integrates with Python

Reports, slides, websites, publication quality figures

2

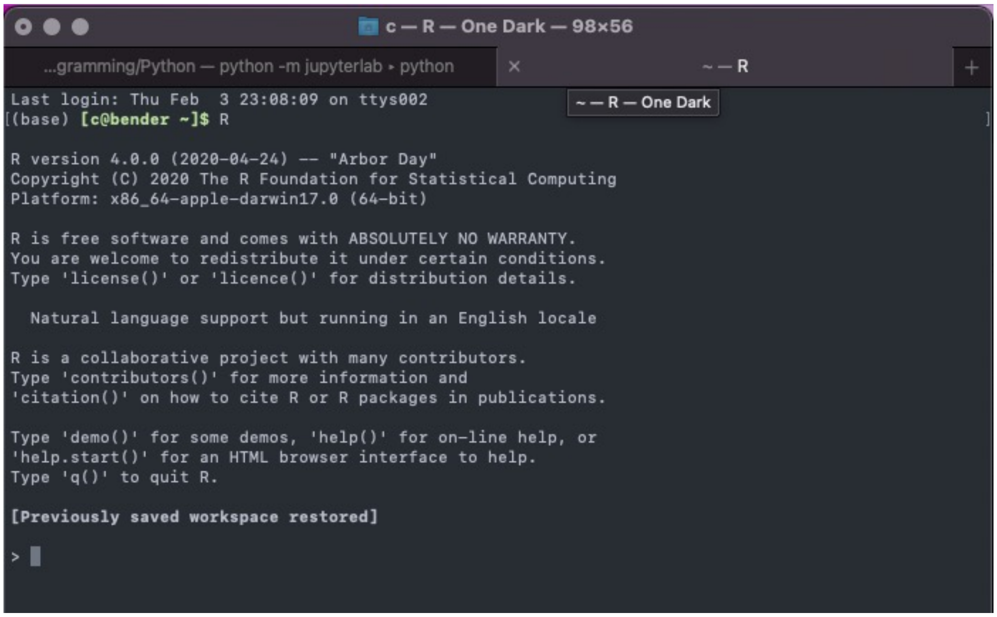
R VS RSTUDIO

R is a *programming language*

RStudio is the *integrated development environment (IDE)*

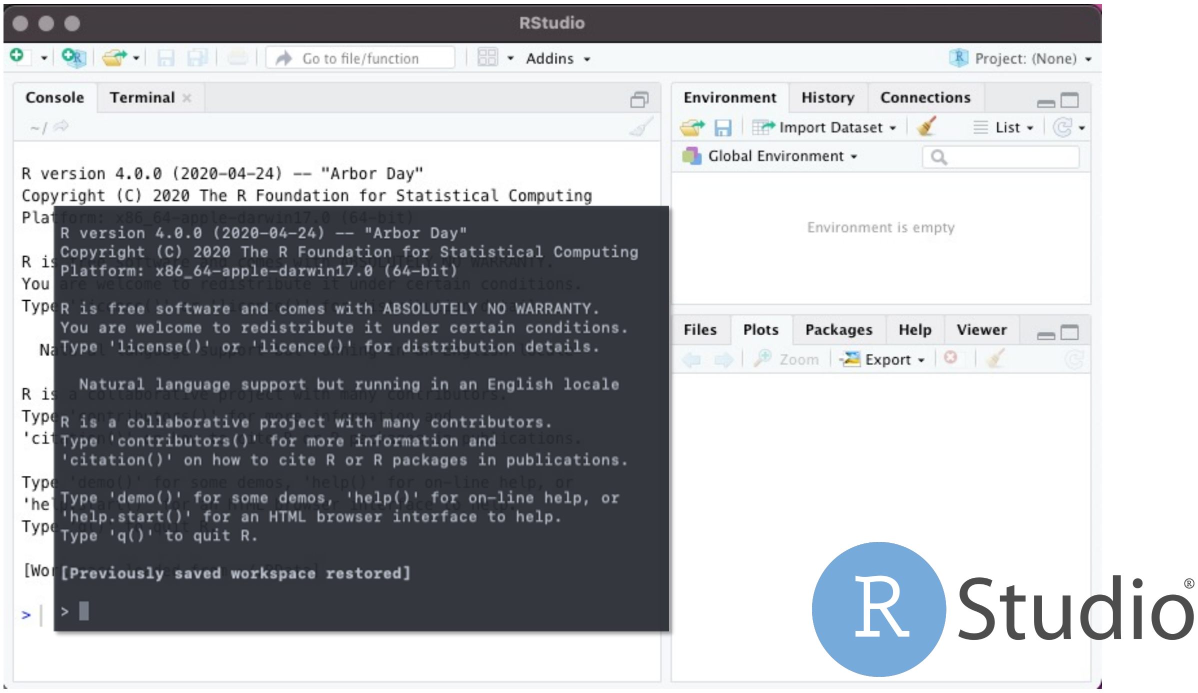
Installing R is a pre-requisite for installing RStudio

3



```
Last login: Thu Feb  3 23:08:09 on ttys002
[(base) [c@bender ~]$ R

R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 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.

[Previously saved workspace restored]

> 
```

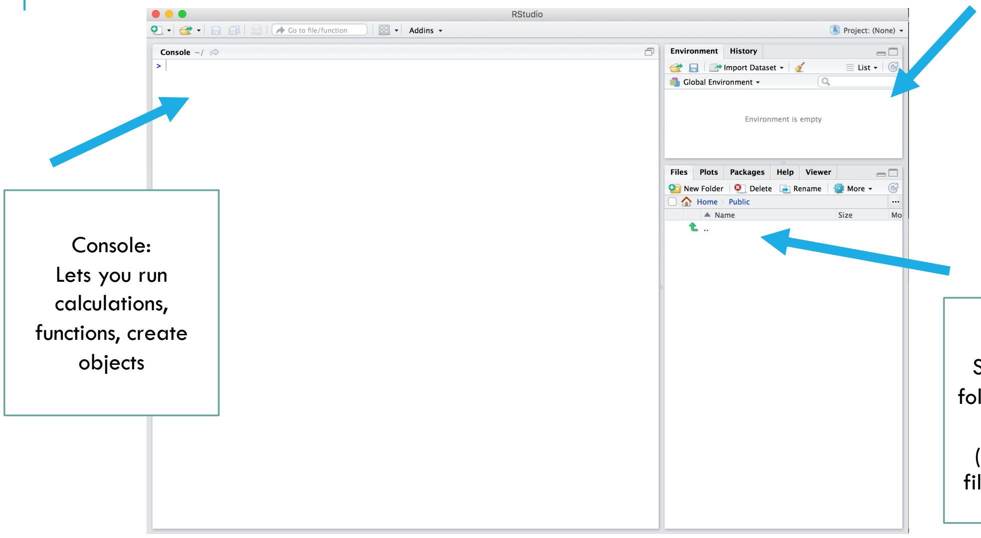
4



```
R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 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.

[Previously saved workspace restored]

> 
```

5

RSTUDIO BASICS



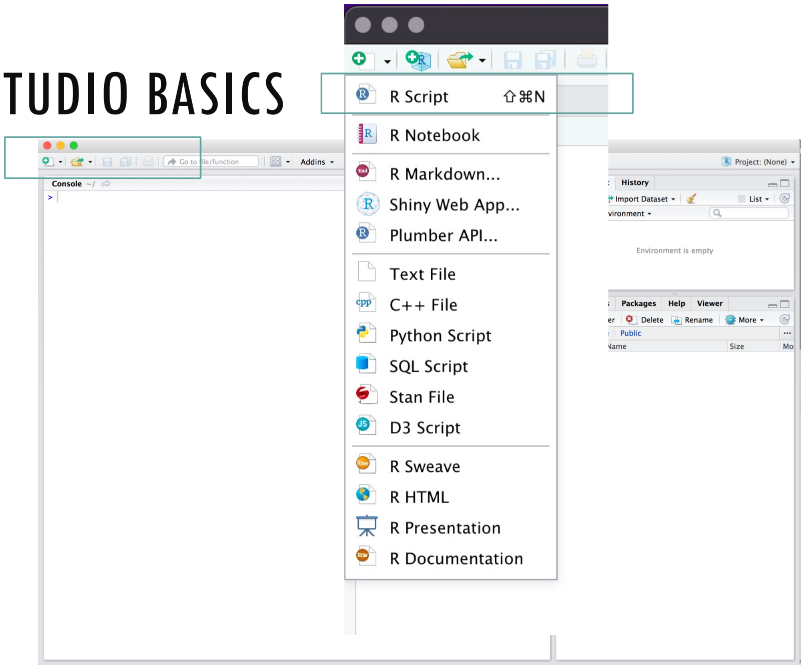
Environment:
Lets you see what objects or variables you have imported or created

Console:
Lets you run calculations, functions, create objects

Files pane:
Shows you which folder on your hard drive you're in (path) and what files are available

6

RSTUDIO BASICS



7

RSTUDIO BASICS

R scripts

- Lets you save all of your commands
- Reproducibility/Replicability

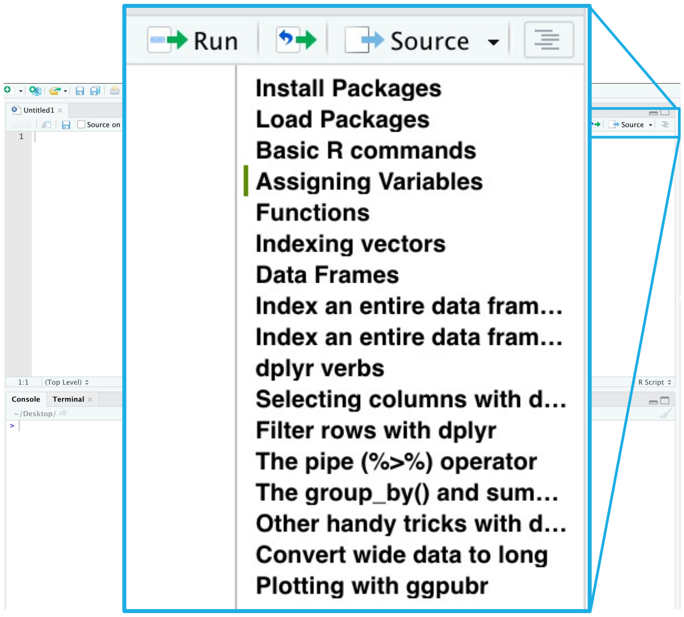
Scripts run each line in sequence

# sign let's you "comment out"

- Shift + Command + C (macOS)
- Shift + Cntrl + C (windows)

"# ----" lets you create sections to organize your code

Highlight and run each line or run the entire script (Source")



8

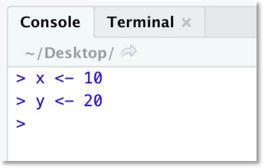
R BASICS: VARIABLE ASSIGNMENT

Variables

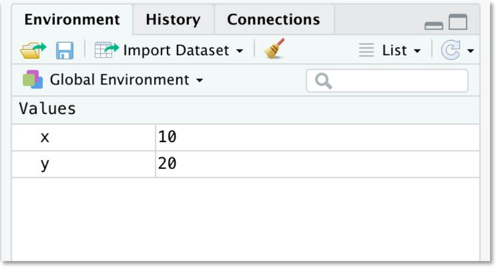
- assign the value 10 to the variable x
- x <- 10 (will then show up in your environment pane)
- y <- 20

"<-" assignment operator

- Mac OS shortcut: option + -
- Windows shortcut: Alt + -



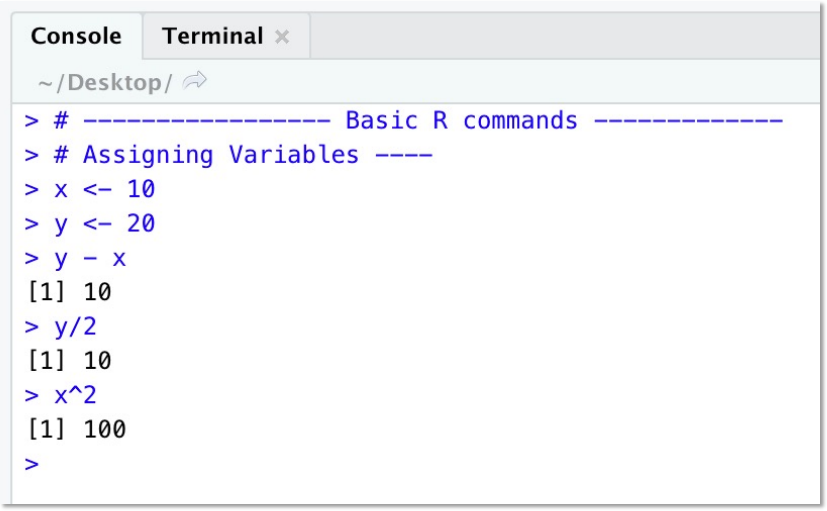
```
> x <- 10
> y <- 20
>
```



```
Values
  x          10
  y          20
```

10

R BASICS: MATHEMATICAL OPERATIONS



```
> # ----------------- Basic R commands -------------
> # Assigning Variables ----
> x <- 10
> y <- 20
> y - x
[1] 10
> y/2
[1] 10
> x^2
[1] 100
>
```

11

FUNCTIONS

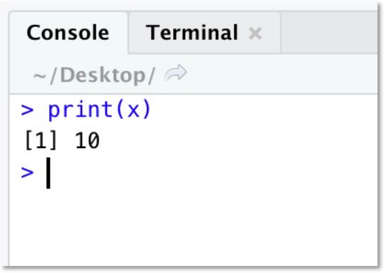
Functions/commands perform some type of operation or task

Some functions are “built-in”

print(x)

other built-ins

- sum()
- mean()
- sd()
- c()



```
> print(x)
[1] 10
>
```

12

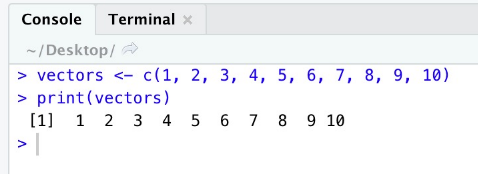
COMBINE: C()

c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

- combines the values

x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

- assigns the values to a variable/vector x
- mean(x)



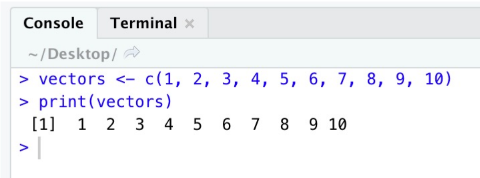
```
> vectors <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
> print(vectors)
 [1]  1  2  3  4  5  6  7  8  9 10
>
```

13

COMBINE: C()

y <- c(“one”, “two”, “three”)

- assigns the combination of strings to a variable y
- mean(y)



```
> vectors <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
> print(vectors)
 [1]  1  2  3  4  5  6  7  8  9 10
> 
```

14

DATA TYPES

numeric

- 1.89, 6.78

integer

- 1, 2, 3

character

- “one”, “two”, “three”, “four”, “five”

logical

- TRUE or FALSE

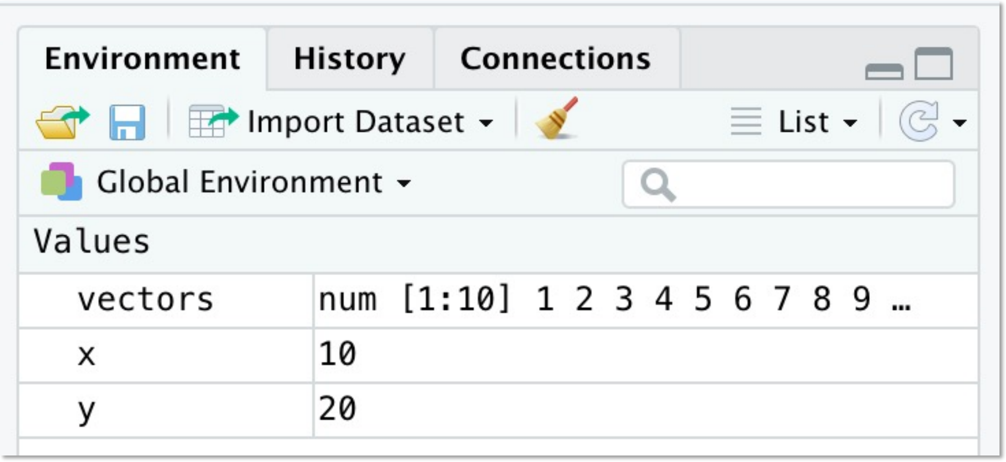
Data frames – analogous to a spread sheet in R

Models – from statistical analysis e.g. a linear regression model

User defined functions

15

DATA TYPES (NUMERIC)

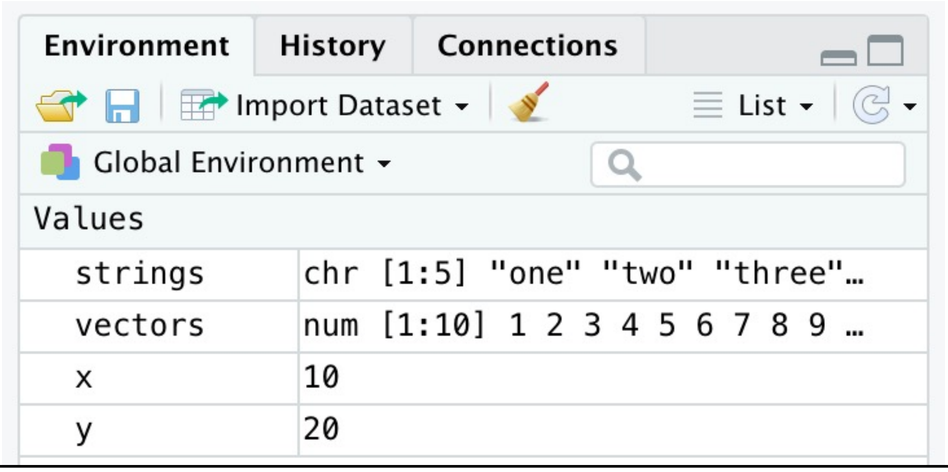


Values

vectors	num [1:10] 1 2 3 4 5 6 7 8 9 …
x	10
y	20

16

DATA TYPES (CHARACTER)

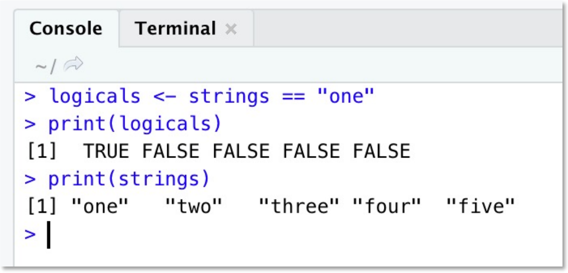


Values

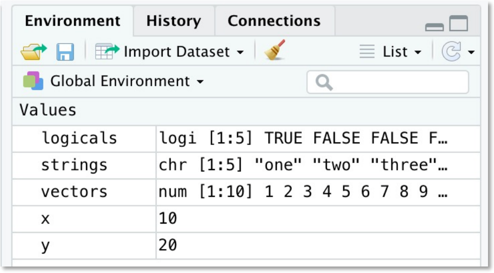
strings	chr [1:5] "one" "two" "three"…
vectors	num [1:10] 1 2 3 4 5 6 7 8 9 …
x	10
y	20

17

DATA TYPES (LOGICALS)



```
> logicals <- strings == "one"
> print(logicals)
[1]  TRUE FALSE FALSE FALSE FALSE
> print(strings)
[1] "one"   "two"   "three" "four"  "five"
> |
```



Values

logicals	logi [1:5] TRUE FALSE FALSE F…
strings	chr [1:5] "one" "two" "three"…
vectors	num [1:10] 1 2 3 4 5 6 7 8 9 …
x	10
y	20

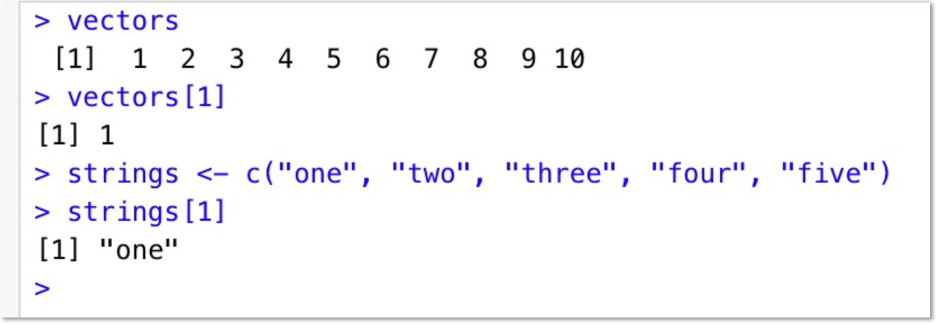
18

INDEXING

In R and MATLAB indexing starts at 1

- Python starts at 0

Use square brackets to index vectors



```
> vectors
 [1]  1  2  3  4  5  6  7  8  9 10
> vectors[1]
[1] 1
> strings <- c("one", "two", "three", "four", "five")
> strings[1]
[1] "one"
>
```

19

PACKAGES

Packages add additional functions to R

20

INSTALL AND LOAD PACKAGES

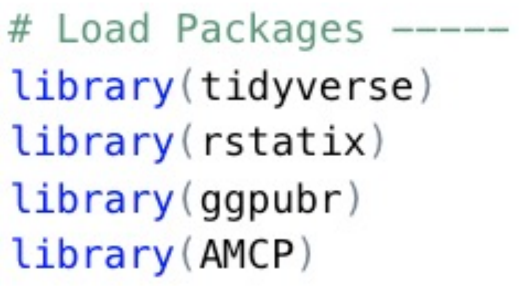
Install

- install.packages(“name of package”)example: install.packages(“AMCP”), notice there are quotes here

Load

- library(name of package)
- library(AMCP), notice there are no quotes here

The CRAN network features over 18,000 packages for R



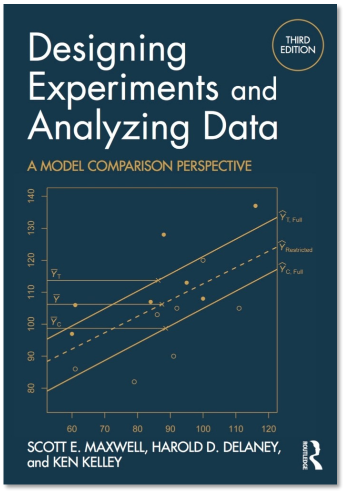
```
# Load Packages -----
library(tidyverse)
library(rstatix)
library(ggpubr)
library(AMCP)
```

21

AMCP: A MODEL COMPARISON PERSPECTIVE

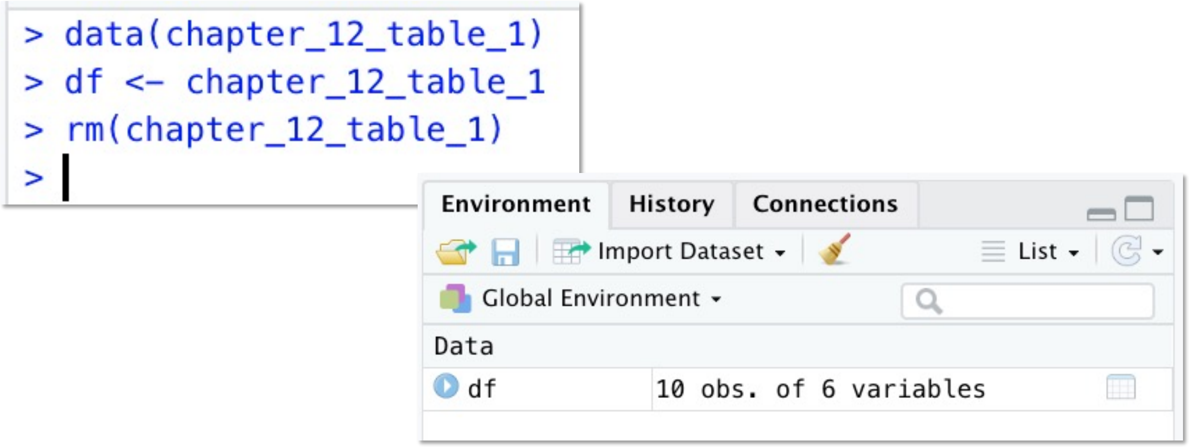
ANOVA Bible

- Between subjects designs
  - Designs with covariates (ANCOVA)
  - Designs with nested or random factors
- Within subjects designs
- Mixed effect models
- Online materials include R code


Designing Experiments and Analyzing Data
THIRD EDITION
A MODEL COMPARISON PERSPECTIVE
SCOTT E. MAXWELL, HAROLD D. DELANEY, and KEN KELLEY

22

DATAFRAMES



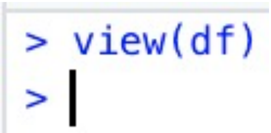
```
> data(chapter_12_table_1)
> df <- chapter_12_table_1
> rm(chapter_12_table_1)
> 
```

Data

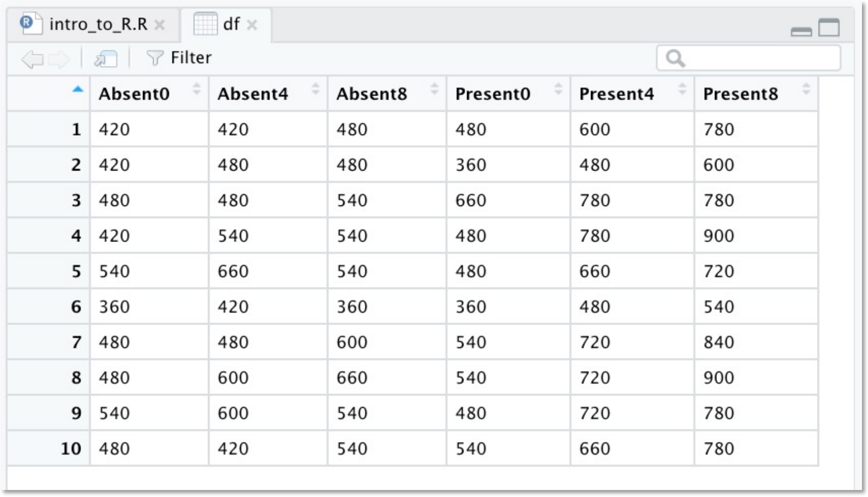
df 10 obs. of 6 variables

25

VIEWING DATAFRAMES: VIEW()



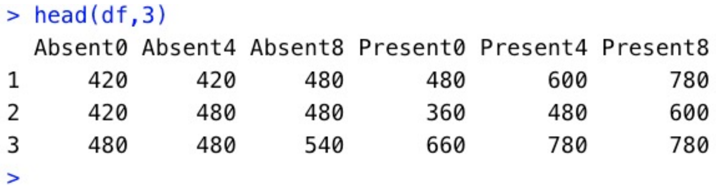
```
> view(df)
> |
```



	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

26

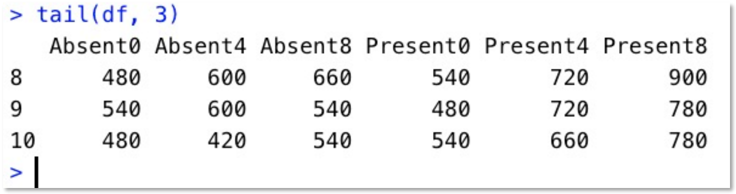
VIEWING DATAFRAMES: HEAD()



```
> head(df,3)
  Absent0 Absent4 Absent8 Present0 Present4 Present8
1     420     420     480      480      600      780
2     420     480     480      360      480      600
3     480     480     540      660      780      780
>
```

27

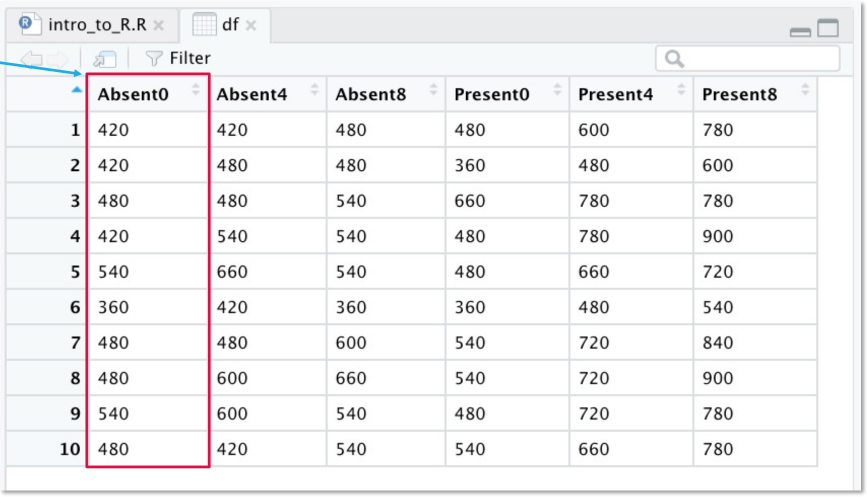
VIEWING DATAFRAMES: TAIL()



```
> tail(df, 3)
   Absent0 Absent4 Absent8 Present0 Present4 Present8
8      480     600     660      540      720      900
9      540     600     540      480      720      780
10     480     420     540      540      660      780
> 
```

28

INDEXING DATAFRAMES: $COL



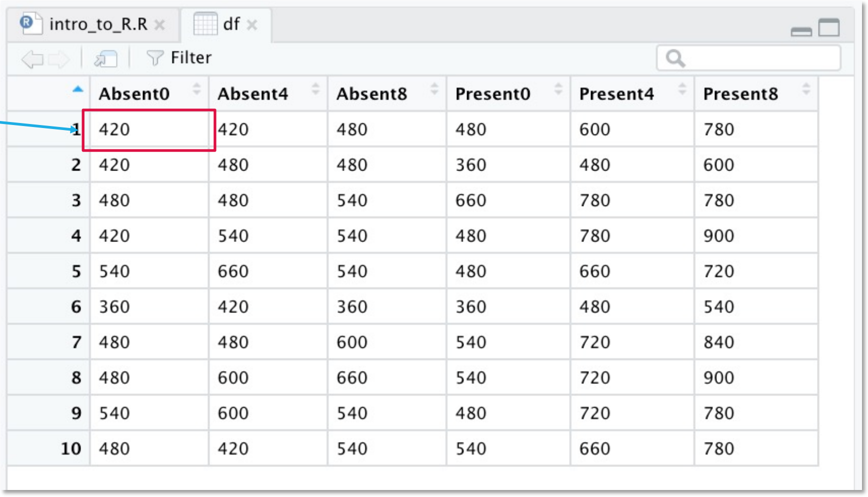
df$Absent0

	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

29

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,1]
df[1,3]
df[10,6]
df[,5]
```

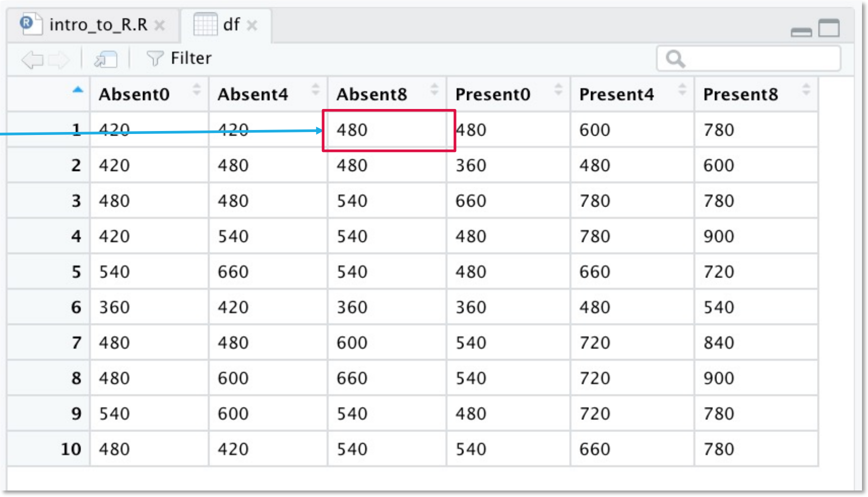


	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

30

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,1]
df[1,3]
df[10,6]
df[,5]
```

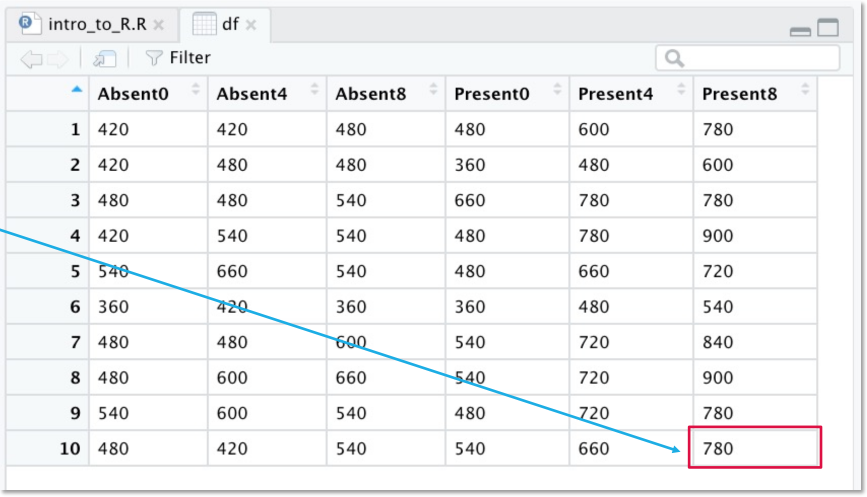


	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

31

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,1]
df[1,3]
df[10,6]
df[,5]
```

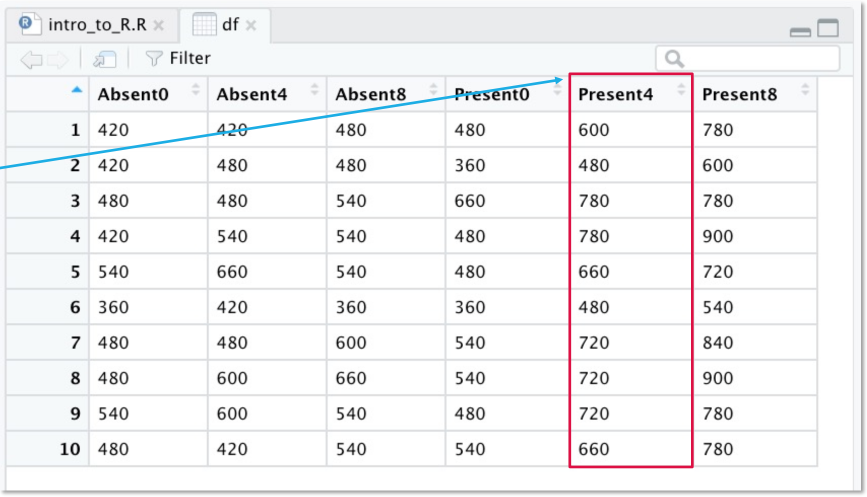


	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

32

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,1]
df[1,3]
df[10,6]
df[,5]
```

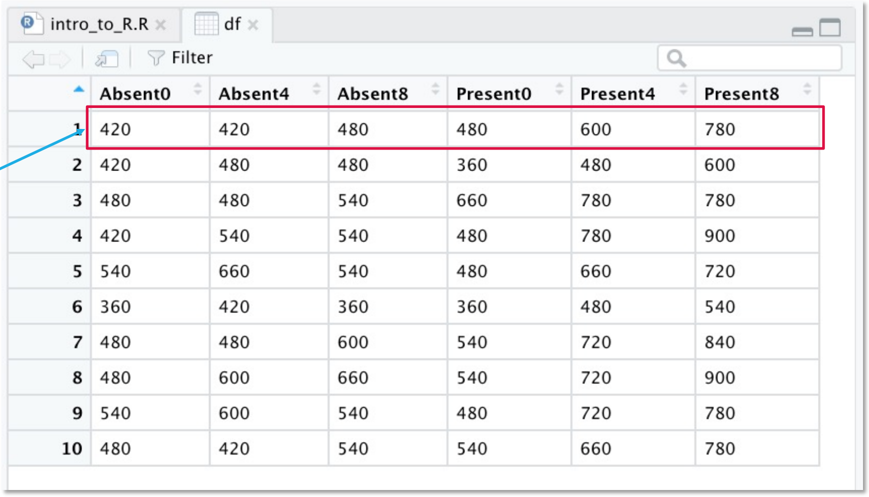


	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

33

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,]
df[1:3,]

df[1:3, 1:3]
```

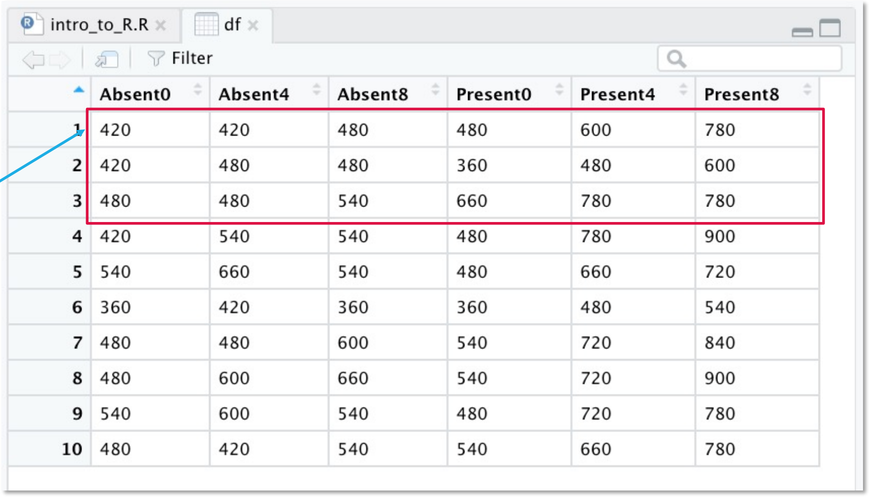


	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

34

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,]
df[1:3,]

df[1:3, 1:3]
```

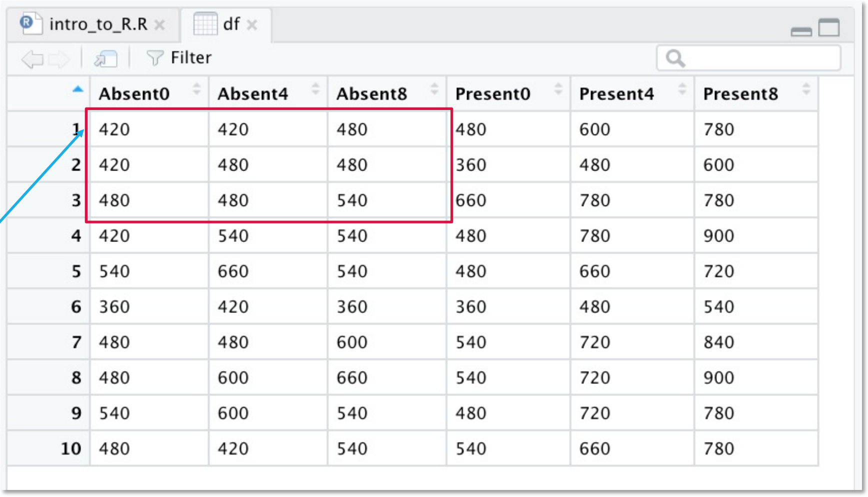


	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

35

INDEXING DATAFRAMES: [ROW, COL]

```
df[1,]
df[1:3,]

df[1:3, 1:3]
```



	Absent0	Absent4	Absent8	Present0	Present4	Present8
1	420	420	480	480	600	780
2	420	480	480	360	480	600
3	480	480	540	660	780	780
4	420	540	540	480	780	900
5	540	660	540	480	660	720
6	360	420	360	360	480	540
7	480	480	600	540	720	840
8	480	600	660	540	720	900
9	540	600	540	480	720	780
10	480	420	540	540	660	780

36