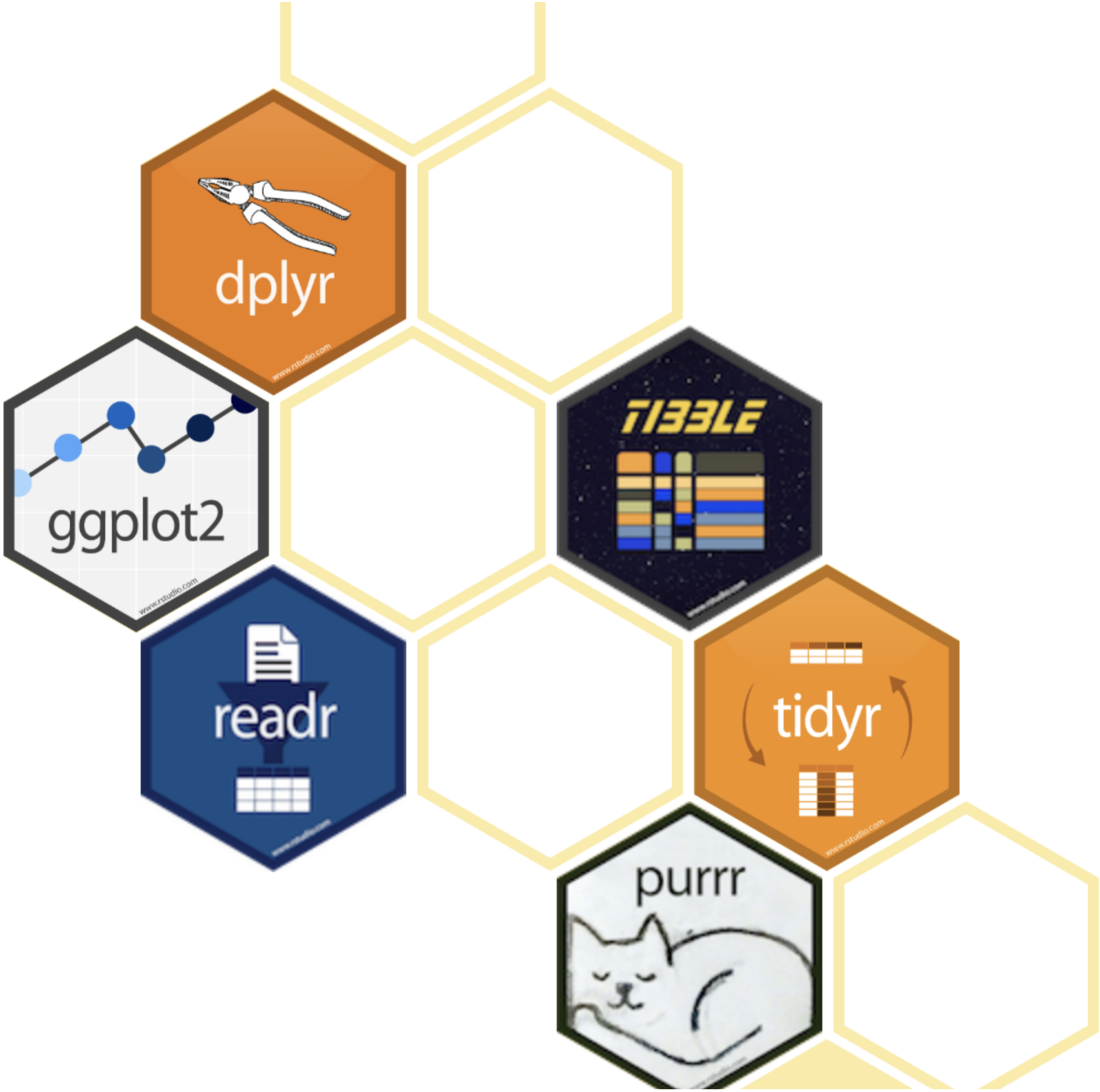
Introduction to the Tidyverse

Import, wrangle, model, and communicate data

2021-03-04


dplyr
www.rstudio.com
ggplot2
www.rstudio.com
TIBBLE
readr
tidyr
www.rstudio.com
purrr

Working with data in R

the tidyverse is a collection of friendly and consistent tools for data analysis and visualization.

Working with data in R

the tidyverse is a collection of friendly and consistent tools for data analysis and visualization.

They live as, R packages, each of which does one thing well.

library(tidyverse) will load the core packages:

ggplot2, for data visualisation.

dplyr, for data manipulation.

tidyr, for data tidying.

readr, for data import.

purrr, for functional programming.

tibble, for tibbles, a modern re-imagining of data frames.

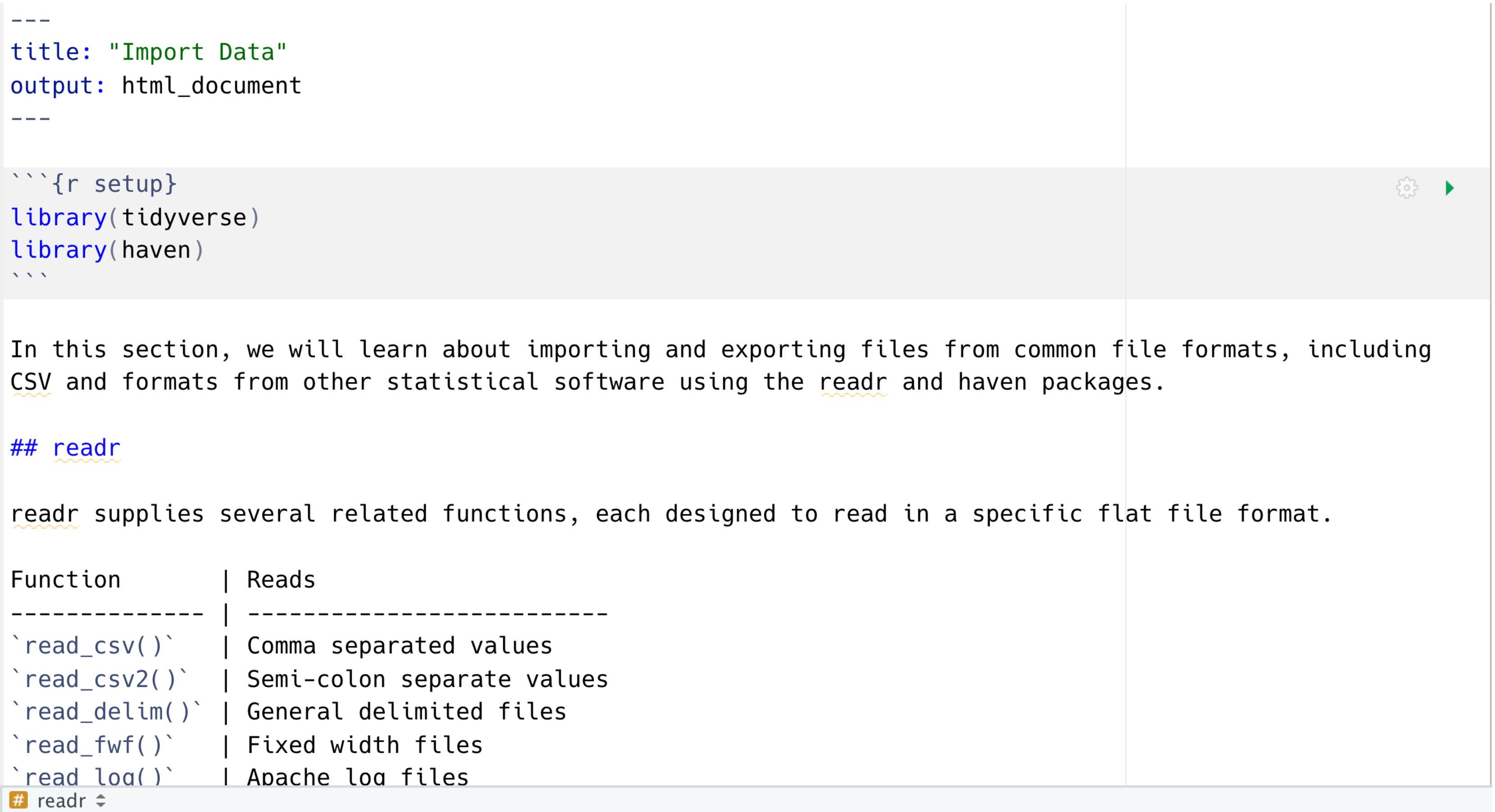
stringr, for strings.

forcats, for factors.

This course is hands on!

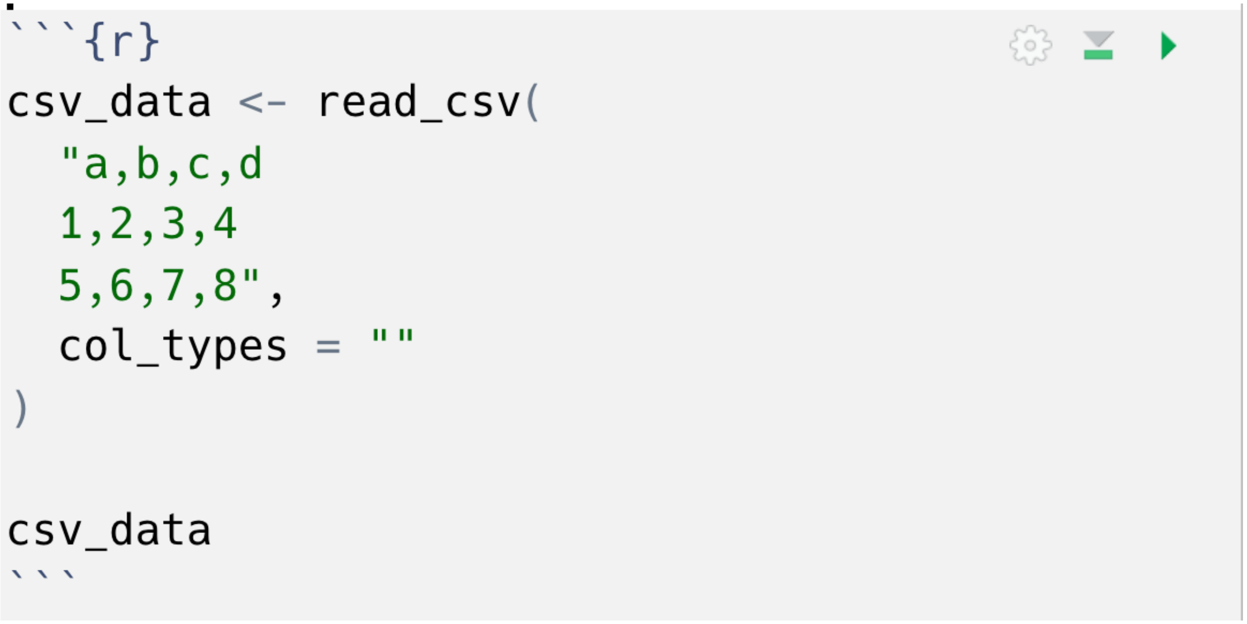
Each section has an exercises file: exercises.Rmd

exercises.Rmd



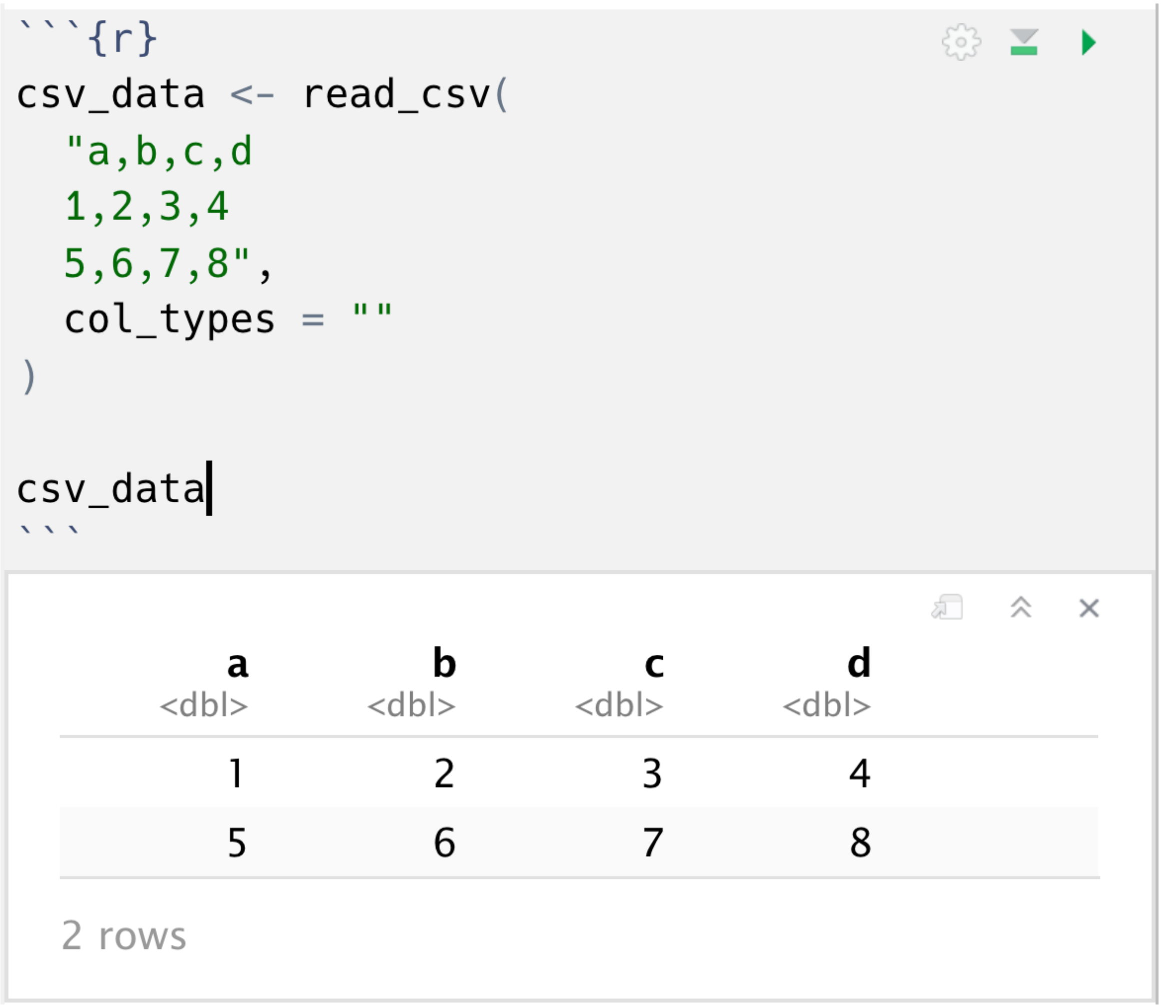
~~~
---
title: "Import Data"
output: html_document
---

```{r setup}
library(tidyverse)
library(haven)
```

In this section, we will learn about importing and exporting files from common file formats, including
CSV and formats from other statistical software using the readr and haven packages.

## readr

readr supplies several related functions, each designed to read in a specific flat file format.

Function       | Reads
-------------- | --------------------------
`read_csv()`   | Comma separated values
`read_csv2()`  | Semi-colon separate values
`read_delim()` | General delimited files
`read_fwf()`   | Fixed width files
`read_log()`   | Apache log files
~~~

code chunks



~~~
```{r}
csv_data <- read_csv(
  "a,b,c,d
  1,2,3,4
  5,6,7,8",
  col_types = ""
)

csv_data
```
~~~

running code chunks

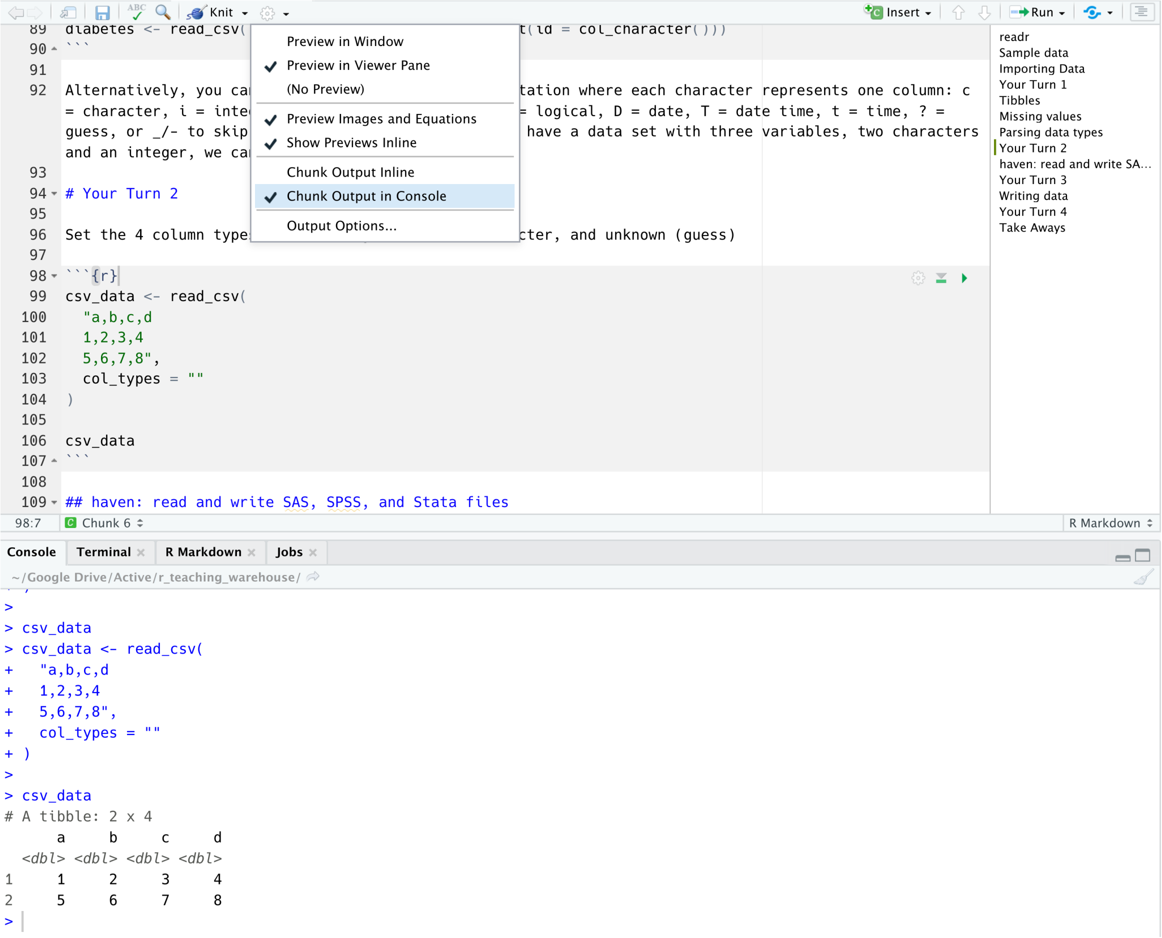


~~~
```{r}
csv_data <- read_csv(
  "a,b,c,d
  1,2,3,4
  5,6,7,8",
  col_types = ""
)

csv_data
```
~~~

a	b	c	d
<dbl>	<dbl>	<dbl>	<dbl>
1	2	3	4
5	6	7	8

2 rows

outputting to the console



```
diabetes <- read_csv(                              t(id = col_character()))
```

~~~
Alternatively, you ca                              tation where each character represents one column: c
= character, i = inte                              = logical, D = date, T = date time, t = time, ? =
guess, or _/- to skip                               have a data set with three variables, two characters
and an integer, we ca

# Your Turn 2

Set the 4 column type                              cter, and unknown (guess)

```{r}
csv_data <- read_csv(
  "a,b,c,d
  1,2,3,4
  5,6,7,8",
  col_types = ""
)

csv_data
```

## haven: read and write SAS, SPSS, and Stata files
~~~

```
>
> csv_data
> csv_data <- read_csv(
+   "a,b,c,d
+   1,2,3,4
+   5,6,7,8",
+   col_types = ""
+ )
>
> csv_data
# A tibble: 2 x 4
      a     b     c     d
  <dbl> <dbl> <dbl> <dbl>
1     1     2     3     4
2     5     6     7     8
>
```

Project contents

```
├── 01-dplyr_5verbs
|   ├── cheatsheet_dplyr_5verbs.pdf
|   ├── diabetes.csv
|   ├── exercises.Rmd
|   ├── slides.pdf
```

Let's head to https://rstudio.cloud/