TODAY’S CLASS

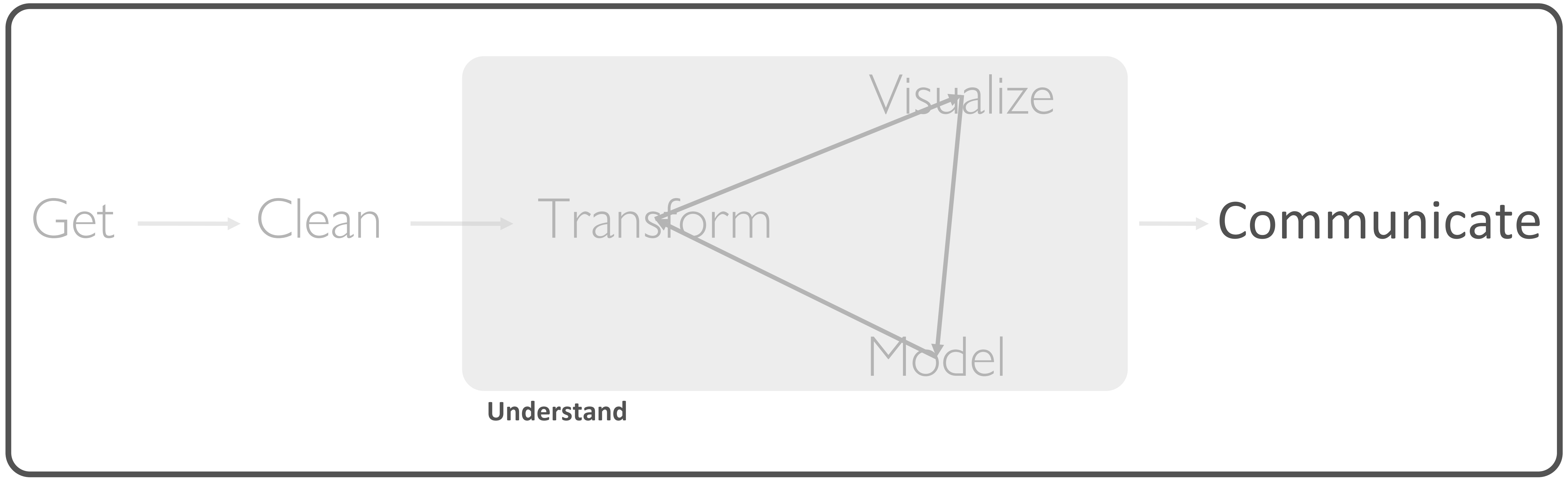
Part 1: Working in a Reproducible Environment:

- ❑ R Projects, R Markdown, R Notebooks
- ❑ Creating and editing R Markdown files

Part 2: Importing Data with Base R and the Tidyverse

Part 3: Coding Exercises (importing data to begin your midterm project)

WORKING IN A REPRODUCIBLE ENVIRONMENT



[†] A modified version of Hadley Wickham's analytic process

“Organization is what you do before you do something, so that when you do it, it is not all mixed up.”

- A. A. Milne

THE BASIC IDEA

Topic	Description
`R project`	?
`R Markdown`	?
`R Notebook`	?

THE BASIC IDEA

Topic	Description
`R project`	Conveniently organizes files pertaining to specific analytic projects
`R Markdown`	Allows user to combine prose, code, and metadata into one file to increase reproducibility and reporting capabilities
`R Notebook`	An R Markdown document that allows for independent and interactive execution of code chunks. Great for sharing.

All 3 have the goal of streamlining your workflow!

RSTUDIO PROJECTS

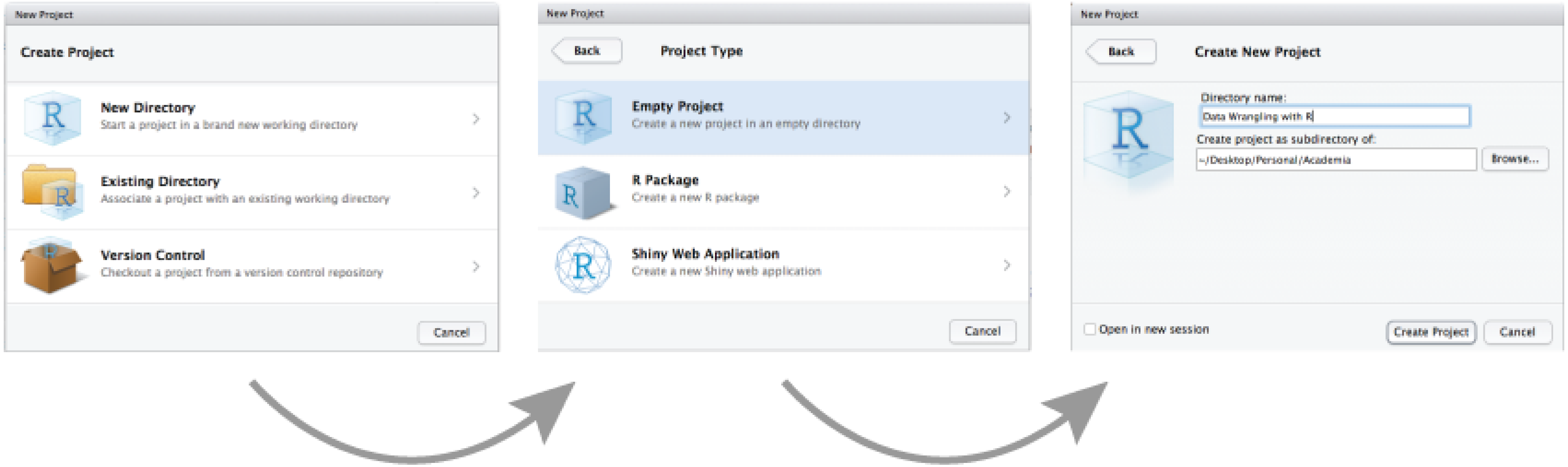
USING PROJECTS

RStudio projects make it straightforward to divide your work into multiple contexts, each with their own:

- working directory
- workspace
- history
- source documents

HOW TO CREATE A PROJECT

File >> New Project



Start a project in a brand new working directory

Associate a project with an existing working directory

Checkout a project from a version control repository

Create a new project in an empty directory

Create a new R package

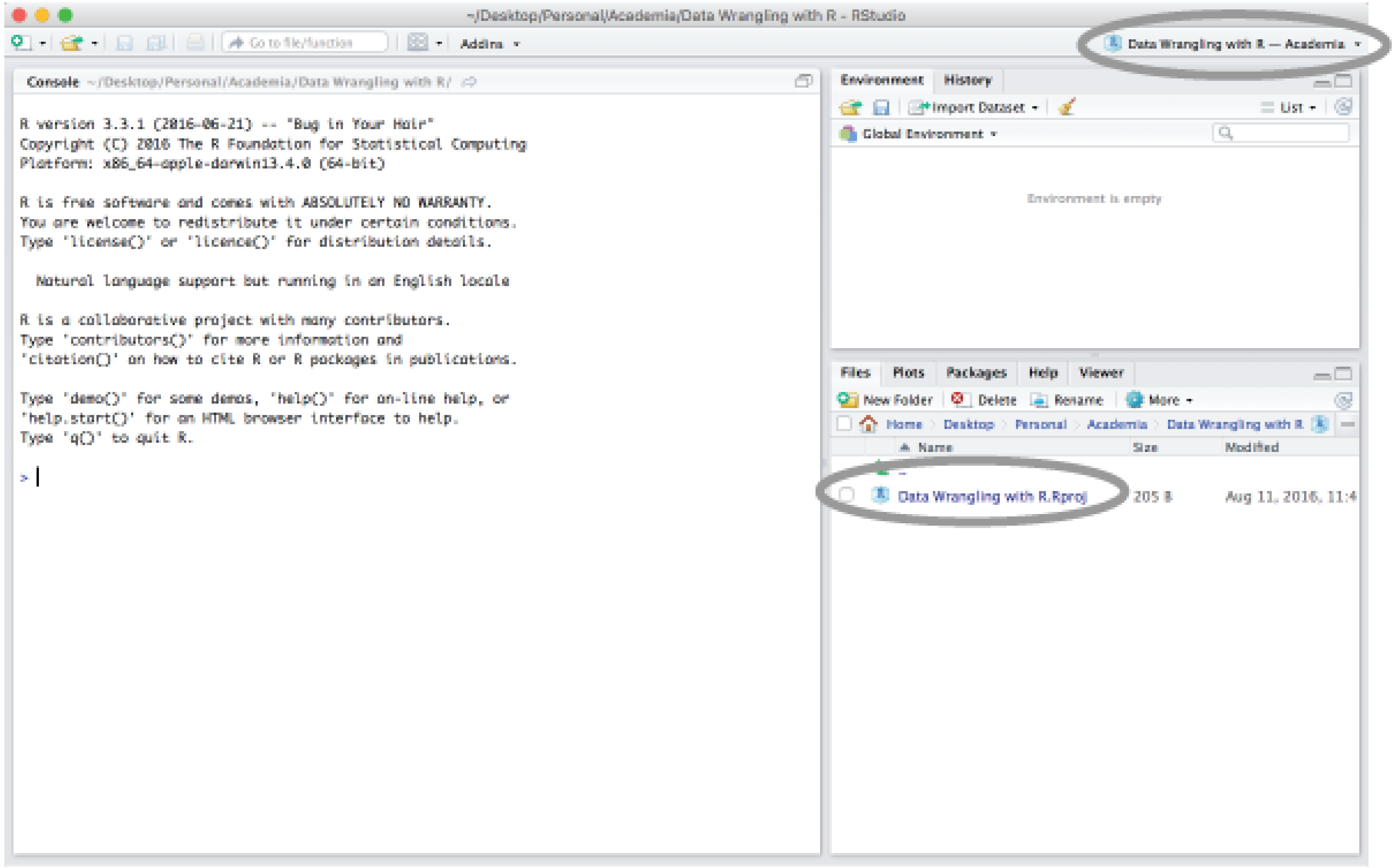
Create a new Shiny web application

Directory name: Data Wrangling with R

Create project as subdirectory of: ~/Desktop/Personal/Academia

☐ Open in new session

SO WHAT'S DIFFERENT?



```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (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.

> 
```

SO WHAT'S DIFFERENT?

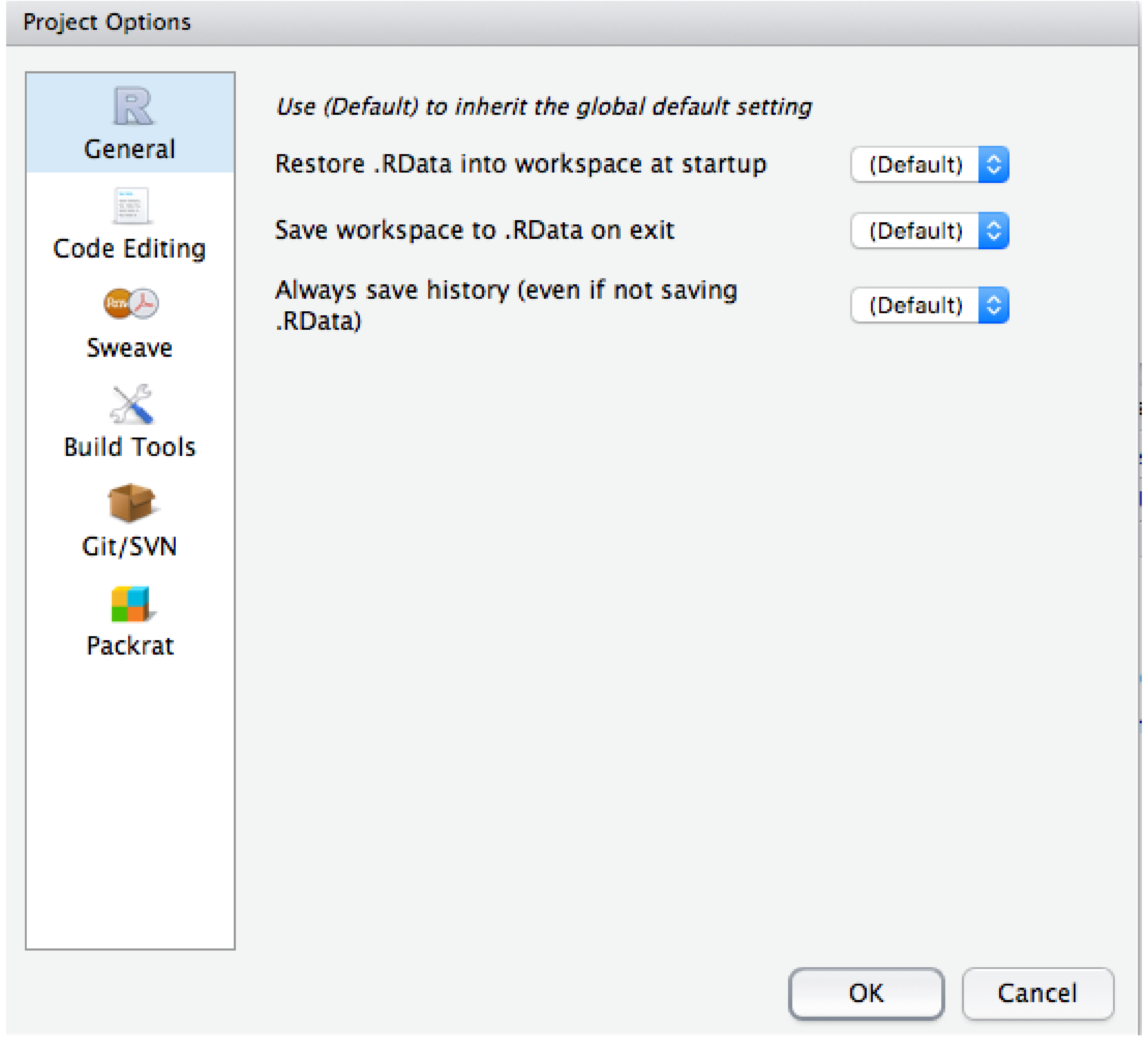
When a project is opened:

- A new R session is started
- The .Rprofile file in the project's main directory is sourced
- ~~The .RData file in the project's main directory is loaded~~
- The .Rhistory file in the project's main directory is loaded
- The current working directory is set to the project directory
- Previously edited source docs are restored to the editor tab
- Other RStudio settings (active tabs, splitter positions, etc) are restored

We can also work with multiple projects at one time

SET PROJECT OPTIONS

Tools >> Project Options



Use (Default) to inherit the global default setting

Restore .RData into workspace at startup (Default)

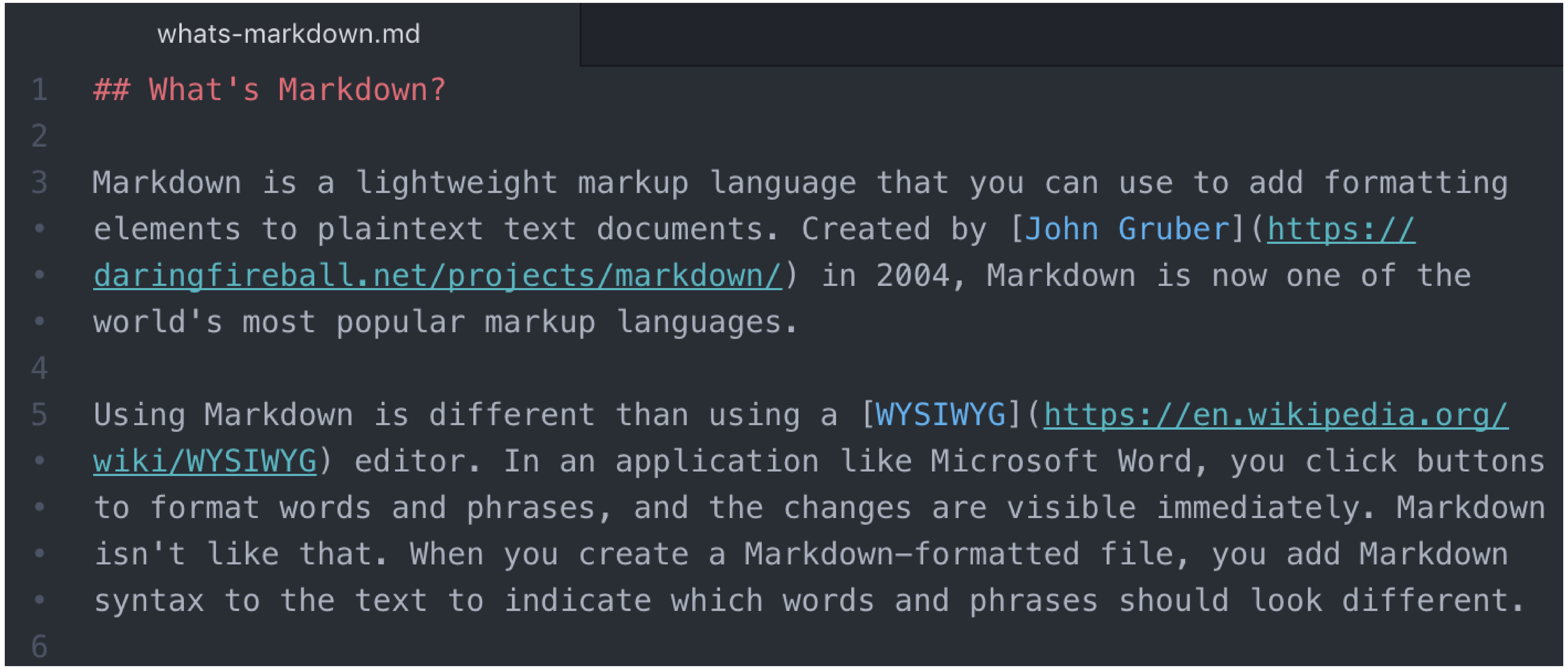
Save workspace to .RData on exit (Default)

Always save history (even if not saving .RData) (Default)

R MARKDOWN

WHAT IS MARKDOWN?

- *A markup language used to format plain text*
- *Used by*
  - *Facebook chat*
  - *Reddit*
  - *Skype*
  - *R*
  - *Python*
  - *And MANY others!*



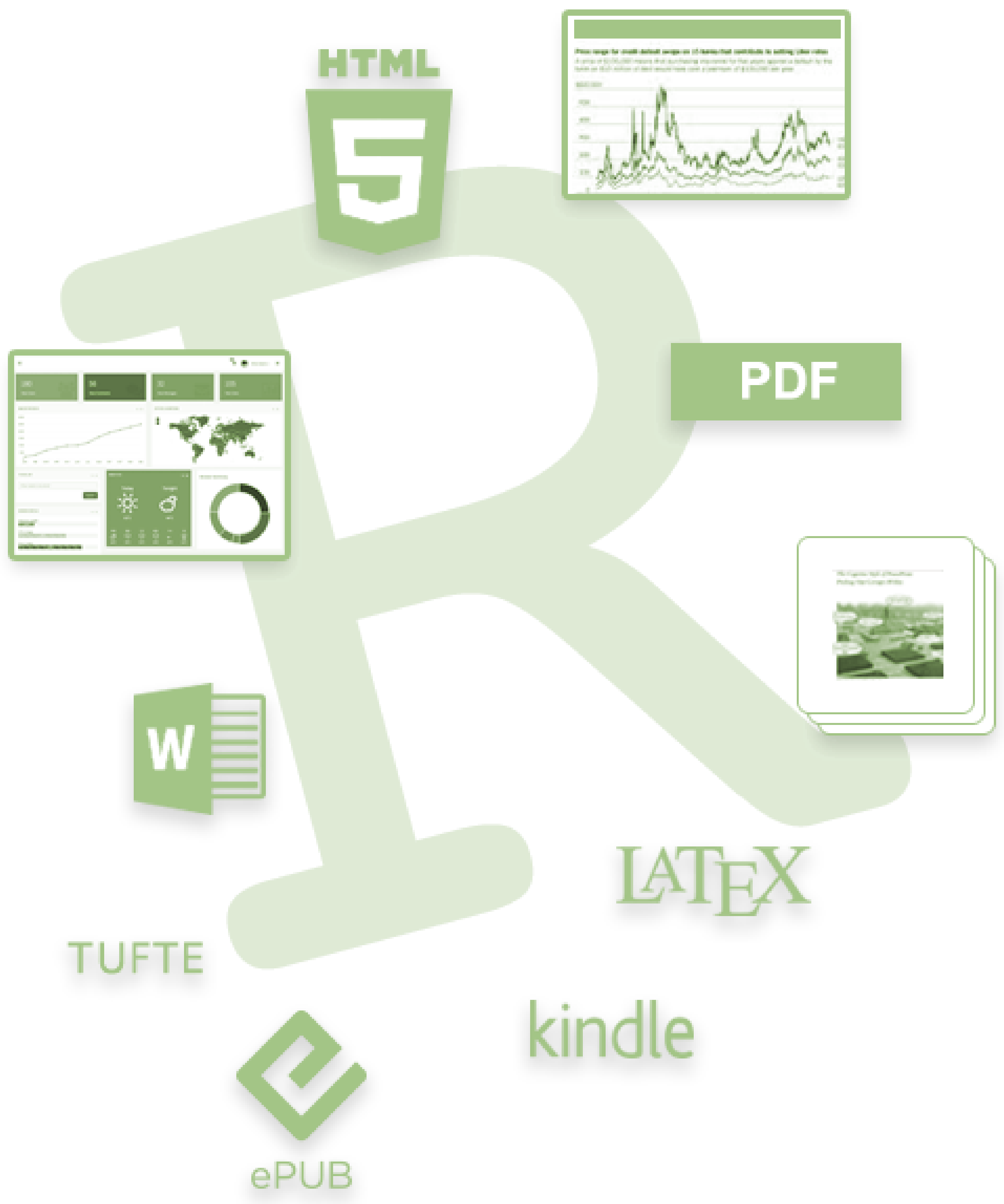
whats-markdown.md

```
## What's Markdown?

Markdown is a lightweight markup language that you can use to add formatting
elements to plaintext text documents. Created by [John Gruber](https://
daringfireball.net/projects/markdown/) in 2004, Markdown is now one of the
world's most popular markup languages.

Using Markdown is different than using a [WYSIWYG](https://en.wikipedia.org/
wiki/WYSIWYG) editor. In an application like Microsoft Word, you click buttons
to format words and phrases, and the changes are visible immediately. Markdown
isn't like that. When you create a Markdown-formatted file, you add Markdown
syntax to the text to indicate which words and phrases should look different.

```

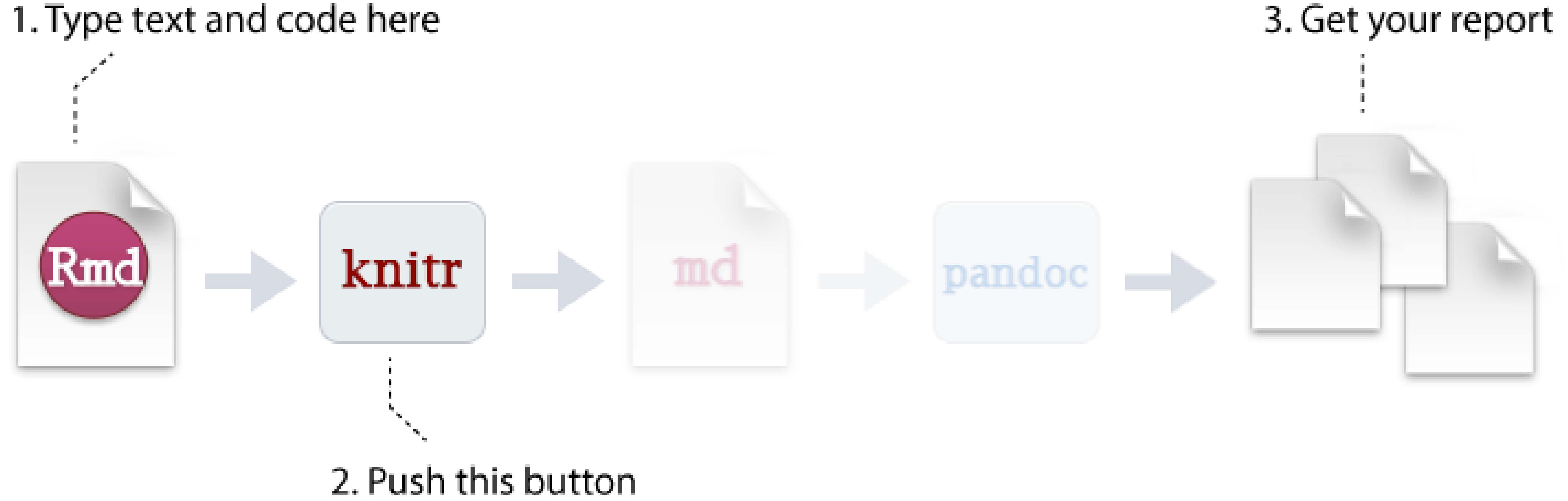
USING R MARKDOWN

R Markdown, a file format that leverages Markdown, allows you to turn your analyses into high quality documents, reports, presentations, and dashboards

- *HTML*
- *PDF*
- *LaTex*
- *ePUB*
- *etc.*



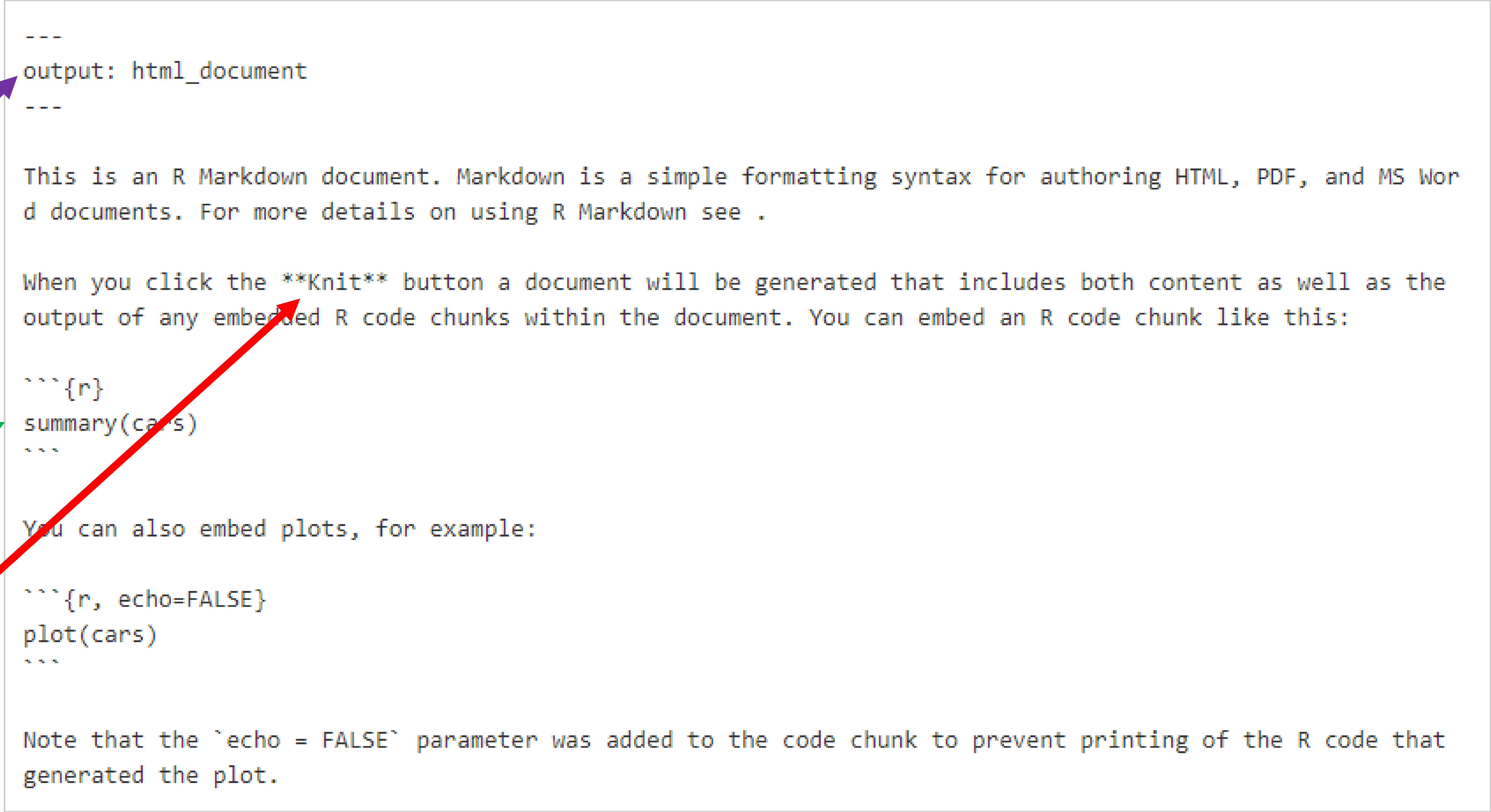
HOW DOES IT WORK?



WHAT DO R MARKDOWN FILES LOOK LIKE?

.Rmd (R Markdown files) typically contains three things:

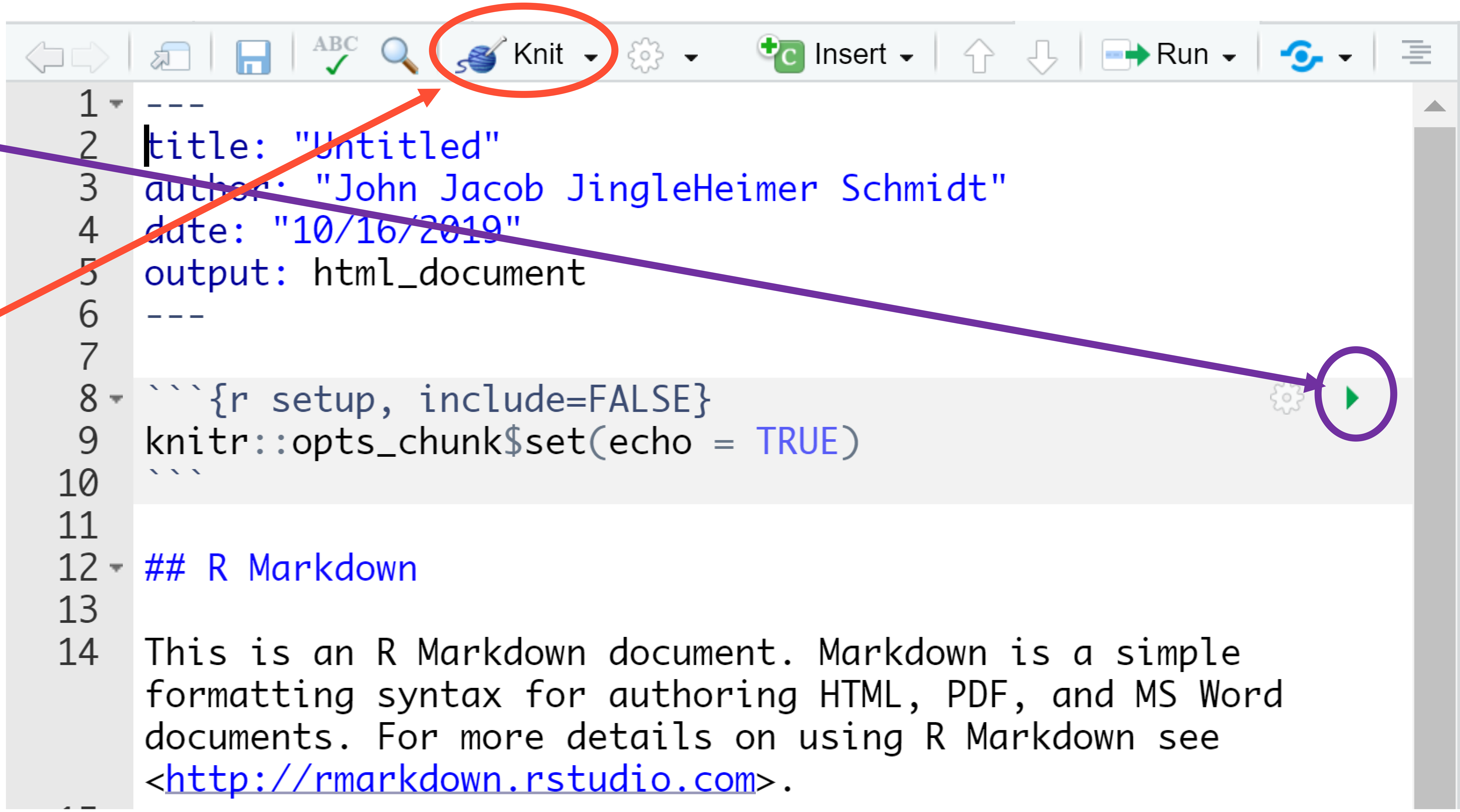
1. (optional) YAML header
2. Chunks of R code
3. Text mixed with simple formatting, leveraging Markdown



~~~
---
output: html_document
---

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see .

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r}
summary(cars)
```

You can also embed plots, for example:

```{r, echo=FALSE}
plot(cars)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
~~~

RUNNING CODE CHUNKS AND ENTIRE R MARKDOWN FILES

- Run individual code chunks with the green run icon
- Produce a complete report by pressing the Knit button
- “Knitting” a document means to execute all code chunks and produce a markdown document



~~~
---
title: "Untitled"
author: "John Jacob JingleHeimer Schmidt"
date: "10/16/2019"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple
formatting syntax for authoring HTML, PDF, and MS Word
documents. For more details on using R Markdown see
<http://rmarkdown.rstudio.com>.
~~~

Let’s Practice!

You’ll need three files open for this activity.

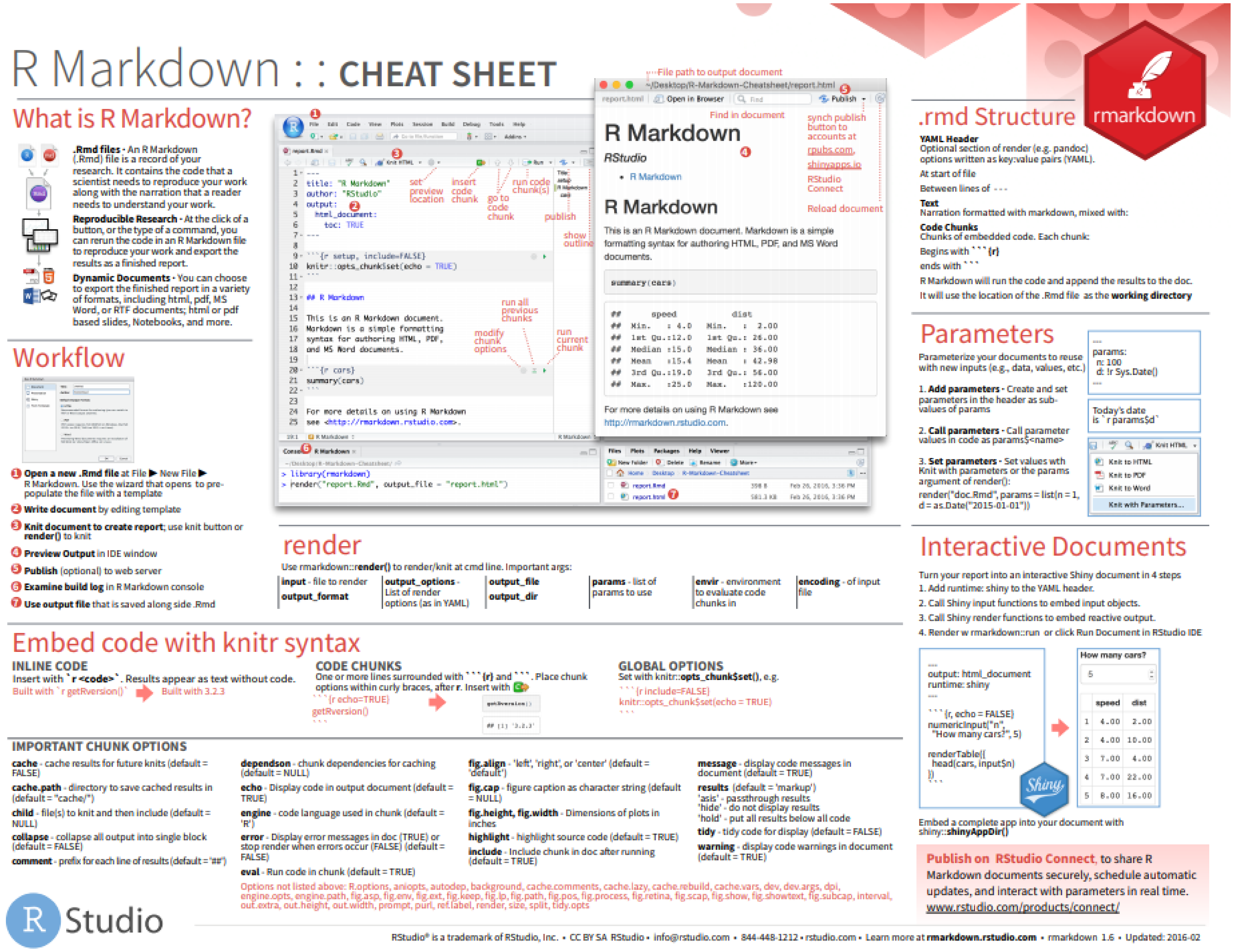
- Create a new .Rmd file: RMarkdown Demonstration Report.Rmd
- Open from today’s Week 2 folder that you downloaded:
  - RMarkdown Demonstration Text.docx
  - RMarkdown Demonstration Code.R

R MARKDOWN CHEAT SHEET!

- RStudio creates cheat sheets to help you remember everything!
- The R Markdown cheat sheet will help you today!
- To find RStudio's cheat sheets, you can:

❑ Google it

❑ In RStudio, go to Help→Cheatsheets



R Markdown : : CHEAT SHEET

What is R Markdown?

.Rmd files · An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

Reproducible Research · At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents · You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow

1. **Open a new .Rmd file** at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template
2. **Write document** by editing template
3. **Knit document to create report**; use knit button or **render()** to knit
4. **Preview Output** in IDE window
5. **Publish** (optional) to web server
6. **Examine build log** in R Markdown console
7. **Use output file** that is saved along side .Rmd

~~~
---
title: "R Markdown"
author: "RStudio"
output:
  html_document:
    toc: TRUE
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

This is an R Markdown document.
Markdown is a simple formatting
syntax for authoring HTML, PDF,
and MS Word documents.

```{r cars}
summary(cars)
```

For more details on using R Markdown
see <http://rmarkdown.rstudio.com>.
```

```
> library(rmarkdown)
> render("report.Rmd", output_file = "report.html")
```


File
Edit
Code
View
Plots
Session
Build
Debug
Tools
Help
Addins
Knit HTML
Run
set preview location
insert code chunk
go to code chunk
run code chunk(s)
publish
show outline
run all previous chunks
modify chunk options
run current chunk
Console
R Markdown
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Desktop
R-Markdown-Cheatsheet
report.Rmd
report.html
File path to output document
~/Desktop/R-Markdown-Cheatsheet/report.html
report.html
Open in Browser
Find
Publish
Find in document
synch publish button to accounts at rpubs.com, shinyapps.io RStudio Connect
Reload document

R Markdown

RStudio

- R Markdown

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

For more details on using R Markdown see http://rmarkdown.rstudio.com.

render

Use rmarkdown::**render()** to render/knit at cmd line. Important args:

input - file to render

output_format

output_options - List of render options (as in YAML)

output_file

output_dir

params - list of params to use

envir - environment to evaluate code chunks in

encoding - of input file

Embed code with knitr syntax

INLINE CODE

Insert with `` `r <code>` ``. Results appear as text without code.

Built with `` `r getRversion()` `` → Built with 3.2.3

CODE CHUNKS

One or more lines surrounded with ` ```{r} ` and ` ``` `. Place chunk options within curly braces, after **r**. Insert with

````
```{r echo=TRUE}
getRversion()
```
~~~

```
getRversion()
## [1] '3.2.3'
```

### GLOBAL OPTIONS

Set with knitr::**opts_chunk$set()**, e.g.

~~~
```{r include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
~~~

### IMPORTANT CHUNK OPTIONS

**cache** - cache results for future knits (default = FALSE)

**cache.path** - directory to save cached results in (default = "cache/")

**child** - file(s) to knit and then include (default = NULL)

**collapse** - collapse all output into single block (default = FALSE)

**comment** - prefix for each line of results (default = '##')

**dependson** - chunk dependencies for caching (default = NULL)

**echo** - Display code in output document (default = TRUE)

**engine** - code language used in chunk (default = 'R')

**error** - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)

**eval** - Run code in chunk (default = TRUE)

**fig.align** - 'left', 'right', or 'center' (default = 'default')

**fig.cap** - figure caption as character string (default = NULL)

**fig.height, fig.width** - Dimensions of plots in inches

**highlight** - highlight source code (default = TRUE)

**include** - Include chunk in doc after running (default = TRUE)

**message** - display code messages in document (default = TRUE)

**results** (default = 'markup')
'asis' - passthrough results
'hide' - do not display results
'hold' - put all results below all code

**tidy** - tidy code for display (default = FALSE)

**warning** - display code warnings in document (default = TRUE)

Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts

## .rmd Structure

**YAML Header**
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).

At start of file

Between lines of - - -

**Text**
Narration formatted with markdown, mixed with:

**Code Chunks**
Chunks of embedded code. Each chunk:

Begins with ` ```{r} `

ends with ` ``` `

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the **working directory**

## Parameters

Parameterize your documents to reuse with new inputs (e.g., data, values, etc.)

1. **Add parameters** - Create and set parameters in the header as sub-values of params
2. **Call parameters** - Call parameter values in code as params$<name>
3. **Set parameters** - Set values wth Knit with parameters or the params argument of render():
render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01"))

```
---
params:
  n: 100
  d: !r Sys.Date()
---
```

Today's date is `` `r params$d` ``


Knit HTML
Knit to HTML
Knit to PDF
Knit to Word
Knit with Parameters...


## Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

1. Add runtime: shiny to the YAML header.
2. Call Shiny input functions to embed input objects.
3. Call Shiny render functions to embed reactive output.
4. Render w rmarkdown::run or click Run Document in RStudio IDE

~~~
---
output: html_document
runtime: shiny
---

```{r, echo = FALSE}
numericInput("n",
  "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```
````

How many cars?

5

	speed	dist
1	4.00	2.00
2	4.00	10.00
3	7.00	4.00
4	7.00	22.00
5	8.00	16.00

Embed a complete app into your document with shiny::**shinyAppDir()**

Publish on RStudio Connect, to share R Markdown documents securely, schedule automatic updates, and interact with parameters in real time. www.rstudio.com/products/connect/

RStudio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **rmarkdown.rstudio.com** • rmarkdown 1.6 • Updated: 2016-02
````

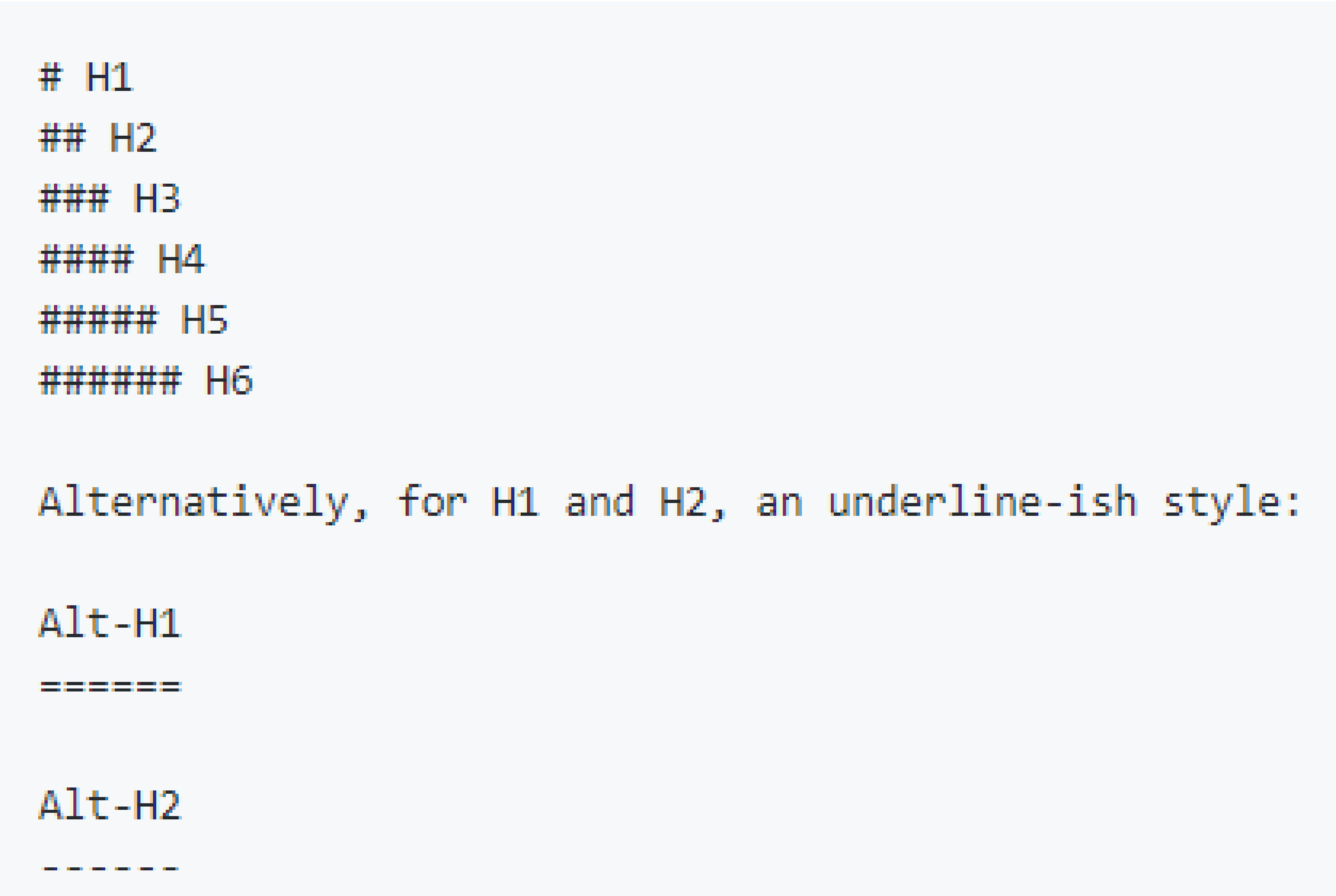
# HOW TO: FORMAT PLAIN TEXT

- Place one asterisk or one underscore on each side of a word/phrase to *italicize* it
- Place two asterisks or two underscores on each side of a word/phrase to **bold** it
- Place backticks around an R command to create inline code (or around a word to monospace it)
- End a word/phrase with ^2^ for superscript or ~2~ for subscript

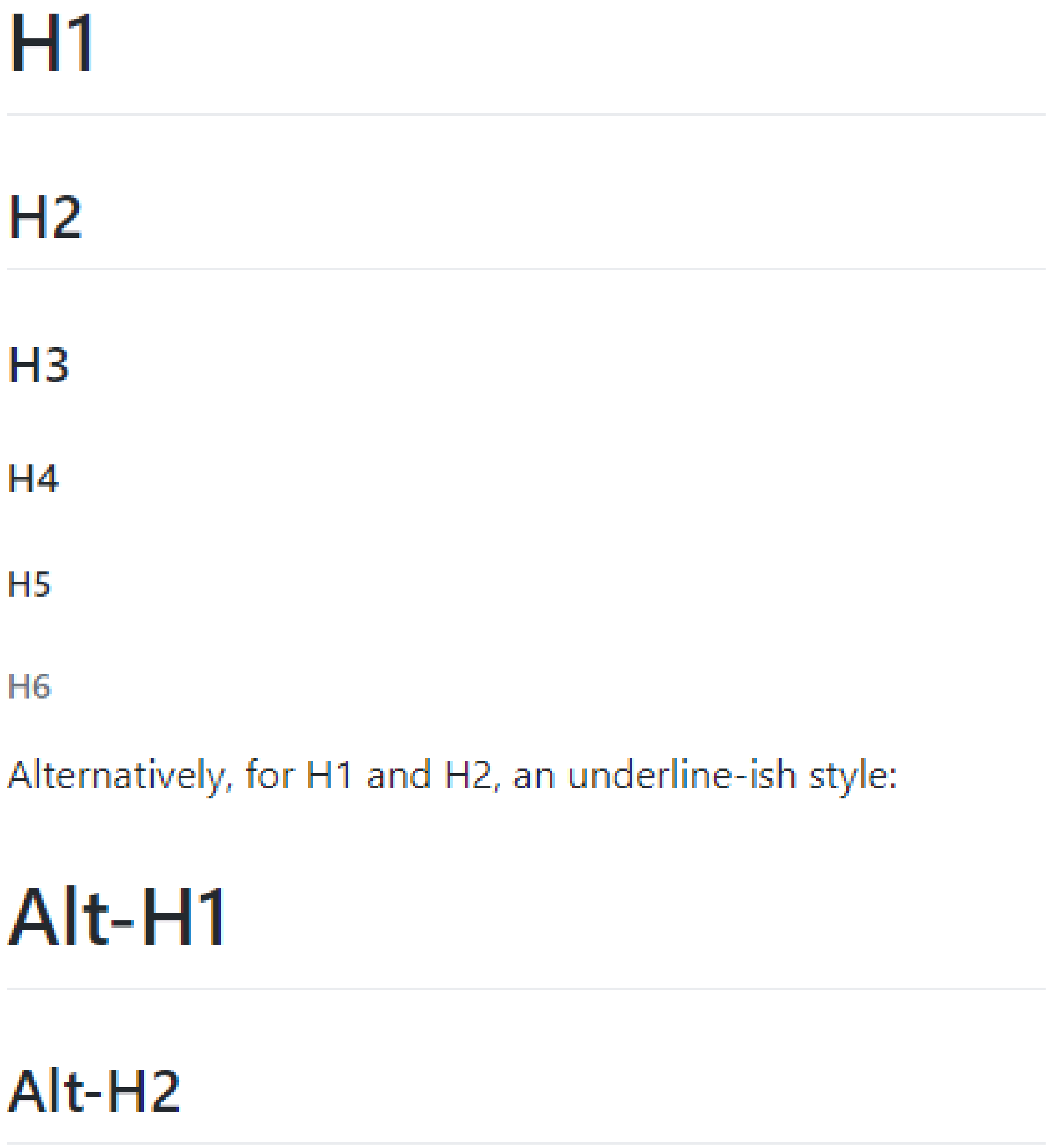
```
# italicize a word/phrase
*italic*
_this phrase is in italics_

# bold a word
**hey I'm bold and awesome**
__bold__

# inline code chunks with backticks
`package_name` # this appears in monospaced font
`nrow(iris)` # this will show a number after knitting

# superscript and subscript
superscript^2^
subscript~2~
```

# HOW TO: FORMAT HEADERS

The number of # signs dictate how large the header is



```
# H1
## H2
### H3
#### H4
##### H5
###### H6

Alternatively, for H1 and H2, an underline-ish style:

Alt-H1
======

Alt-H2
------
```



H1

H2

H3

H4

H5

H6

Alternatively, for H1 and H2, an underline-ish style:

Alt-H1

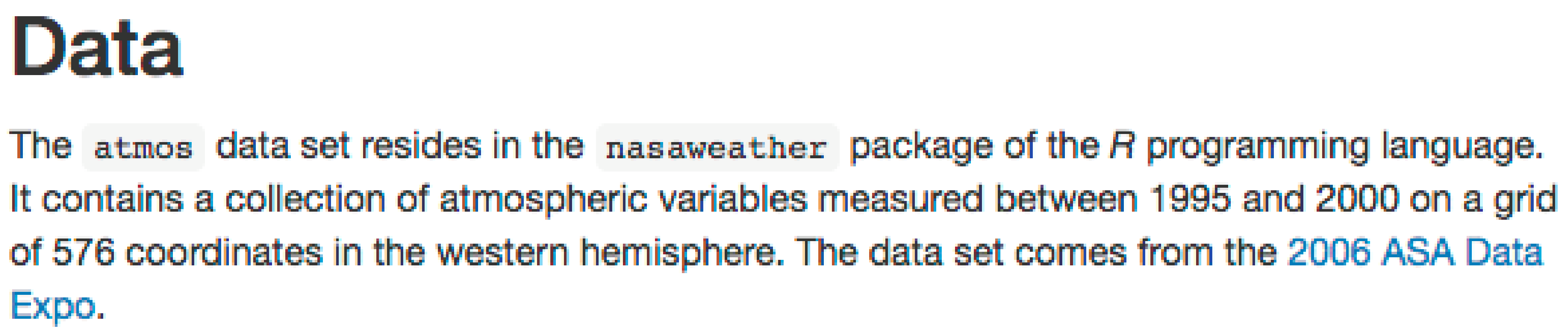
Alt-H2

# HOW TO: LINKS AND IMAGES

```
# show a URL as a hyperlink
<http://whatacoolexample.com>


# show a hyperlink with a linked phrase
[OMG this text will show instead of the URL](http://whatacoolexample.com)


# caption for a picture
![cute dog pic](/file/location/cutedogpic.png)
```

# YOUR TURN!

From the Word document, copy all text beginning with *# Data* and ending with *"…2006 ASA Data Expo"* and paste the text into your .Rmd file.



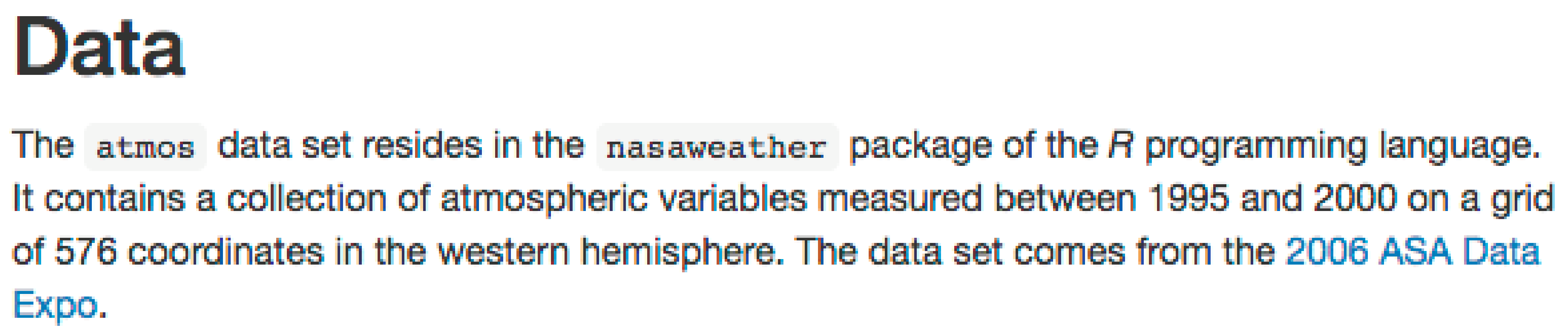
## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

1. Turn the line that begins with "Data" into a second level header.
2. Change the words atmos and nasaweather into a monospaced font suitable for code snippets.
3. Make the letter R italicized.
4. Change "2006 ASA Data Expo" to a link that points to http://stat-computing.org/dataexpo/2006

# SOLUTION

1. Turn the line that begins with "Data" into a second level header.



**Data**

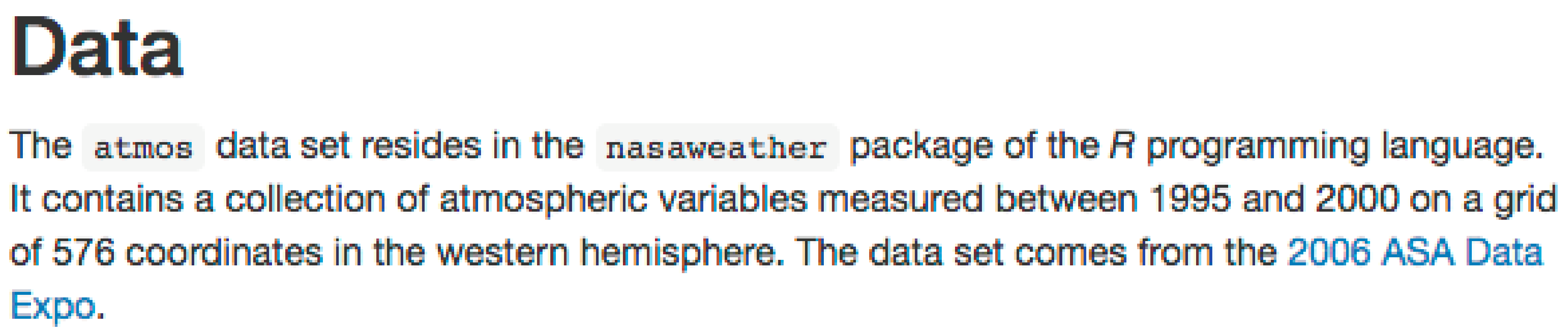
The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

```
## Data
```

# SOLUTION

2. Change the words atmos and nasaweather into a monospaced font suitable for code snippets.

The `atmos` data set resides in the `nasaweather` package



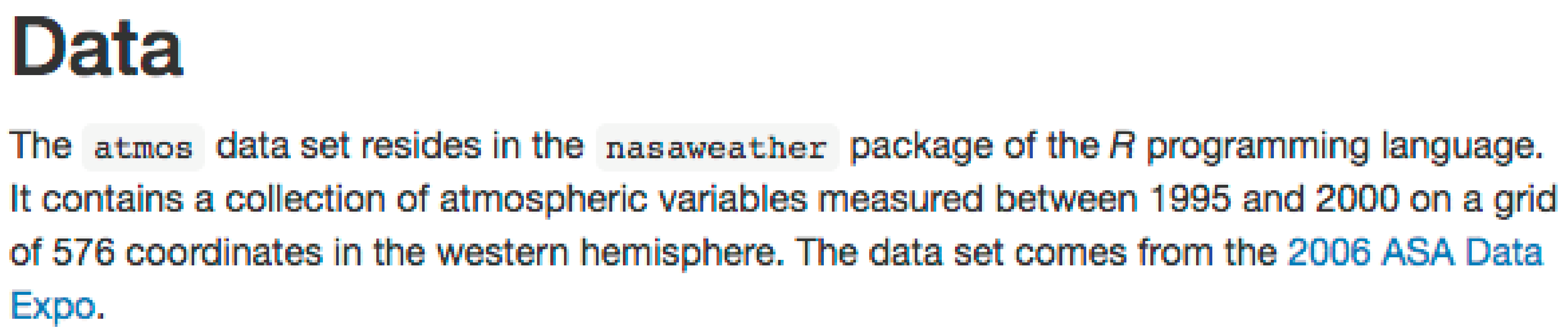
## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

# SOLUTION

3. Make the letter R italicized.

```
*R*
```

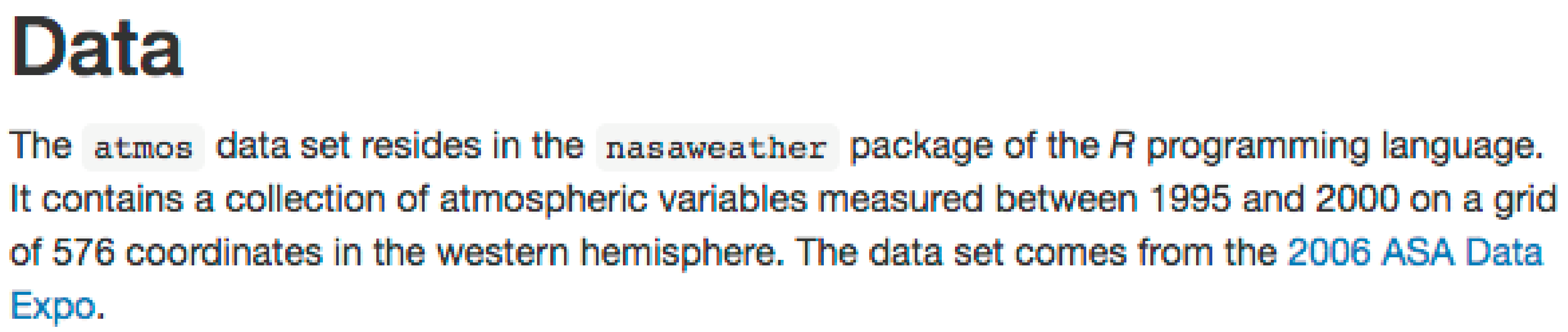


## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

# SOLUTION

4. Change "2006 ASA Data Expo" to a link that points to http://stat-computing.org/dataexpo/2006



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

```
The data set comes from the [2006 ASA Data Expo](http://stat-computing.org/dataexpo/2006).
```

# HOW TO: LISTS

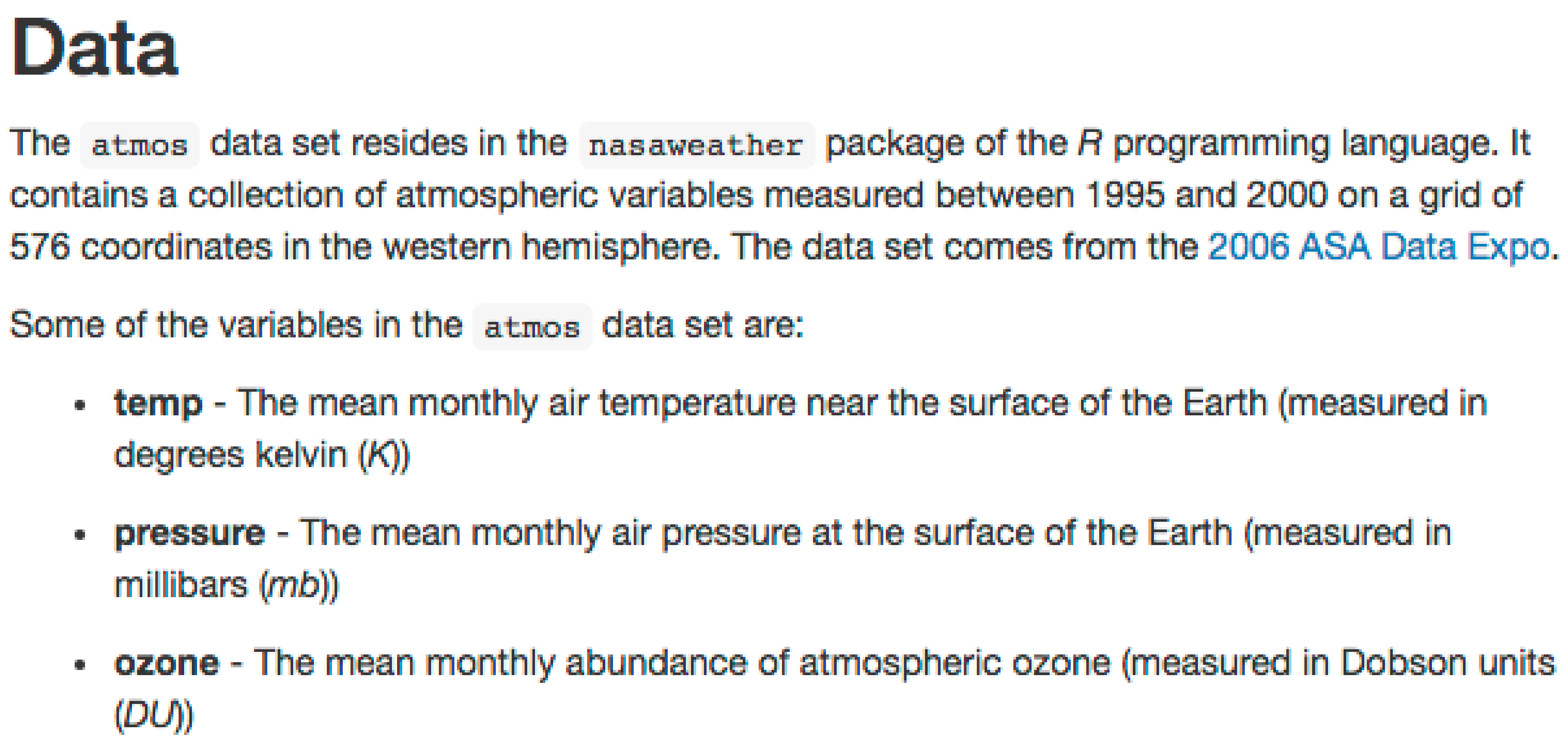
```
# numbered lists (don’t forget the space after the dot)
1. One
2. Two
3. Three

# bulleted list with sub points (two spaces in front for sub points)
# can use *, -, or + symbol for bullet points
* eggs
* milk
  - skim
  - full fat
* bread
* (chocolate to eat on way home so bae doesn’t eat it)

# can mix types
1. it’s
2. a
  + hi
  + there
3. mixed list
```

# YOUR TURN!

5. Turn the text into a bulleted list with 3 bullets: temp, pressure, ozone.

6. Make temp, pressure, ozone **bold** at the start of each entry.

7. Make *K*, *mb*, and *DU* italicized at the end of each entry.



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

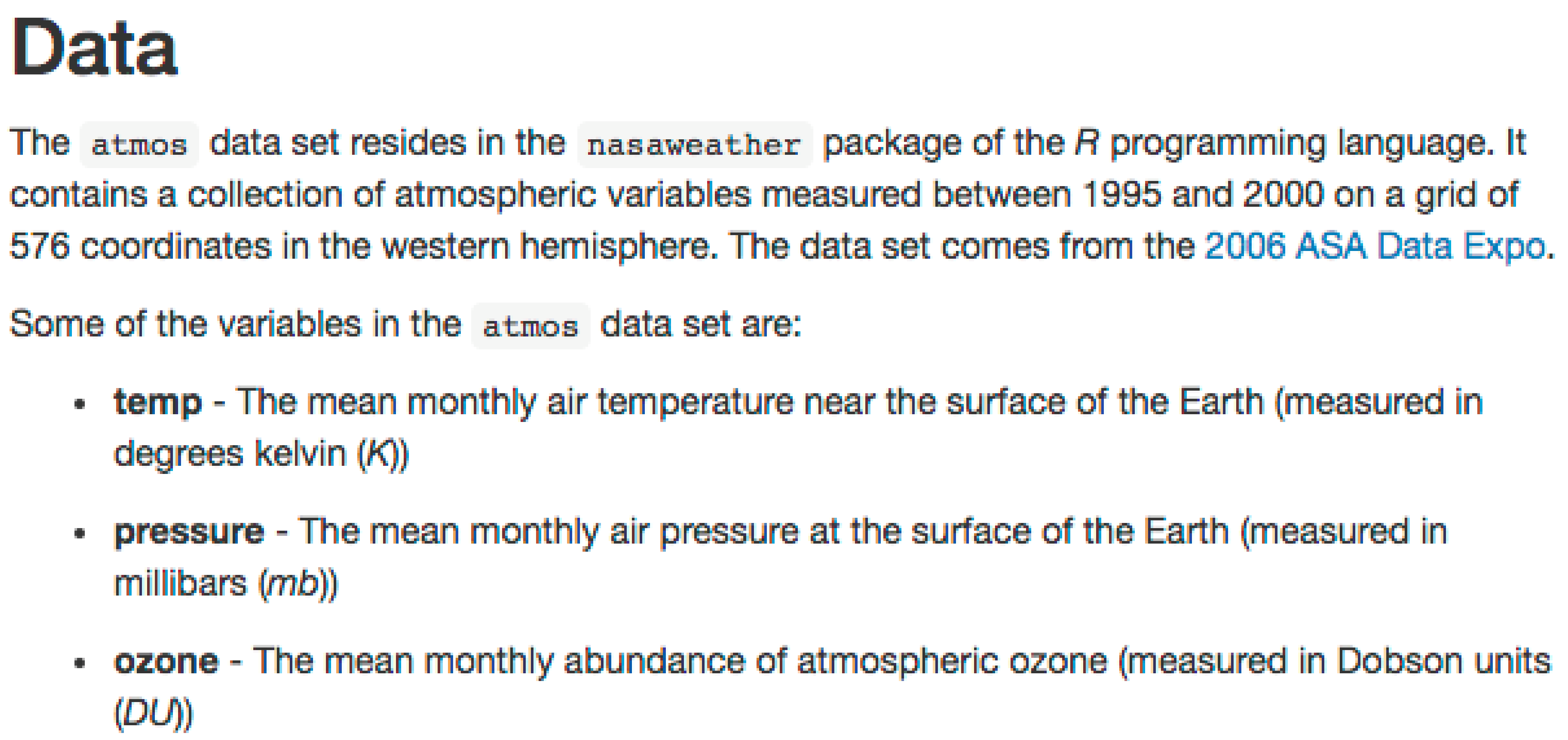
# SOLUTION

5. Turn the text into a bulleted list with 3 bullets: temp, pressure, ozone.

* temp –

* pressure –

* ozone –



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

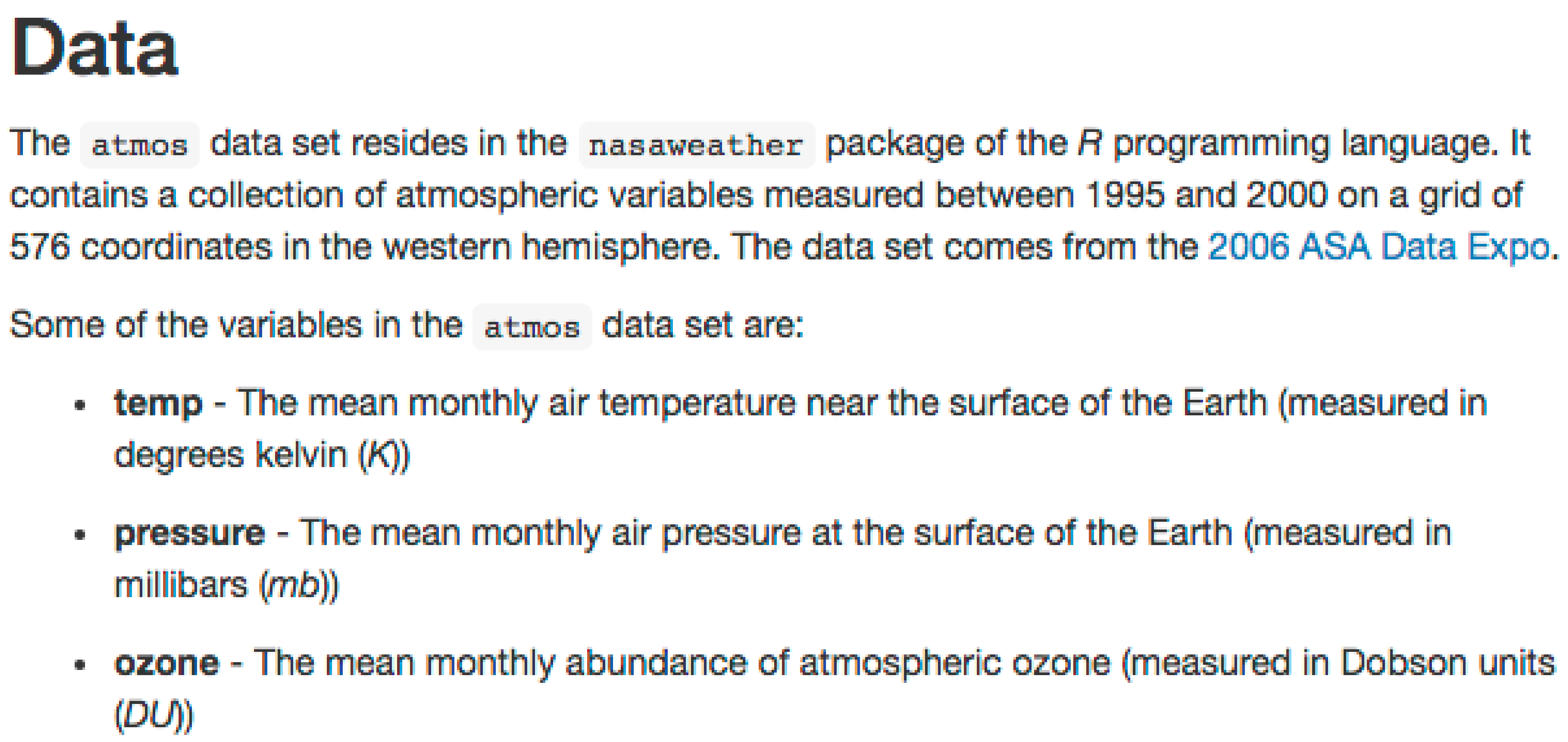
Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

# SOLUTION

6. Make temp, pressure, ozone **bold** at the start of each entry.

```
* **temp** -

* **pressure** -

* **ozone** -
```



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

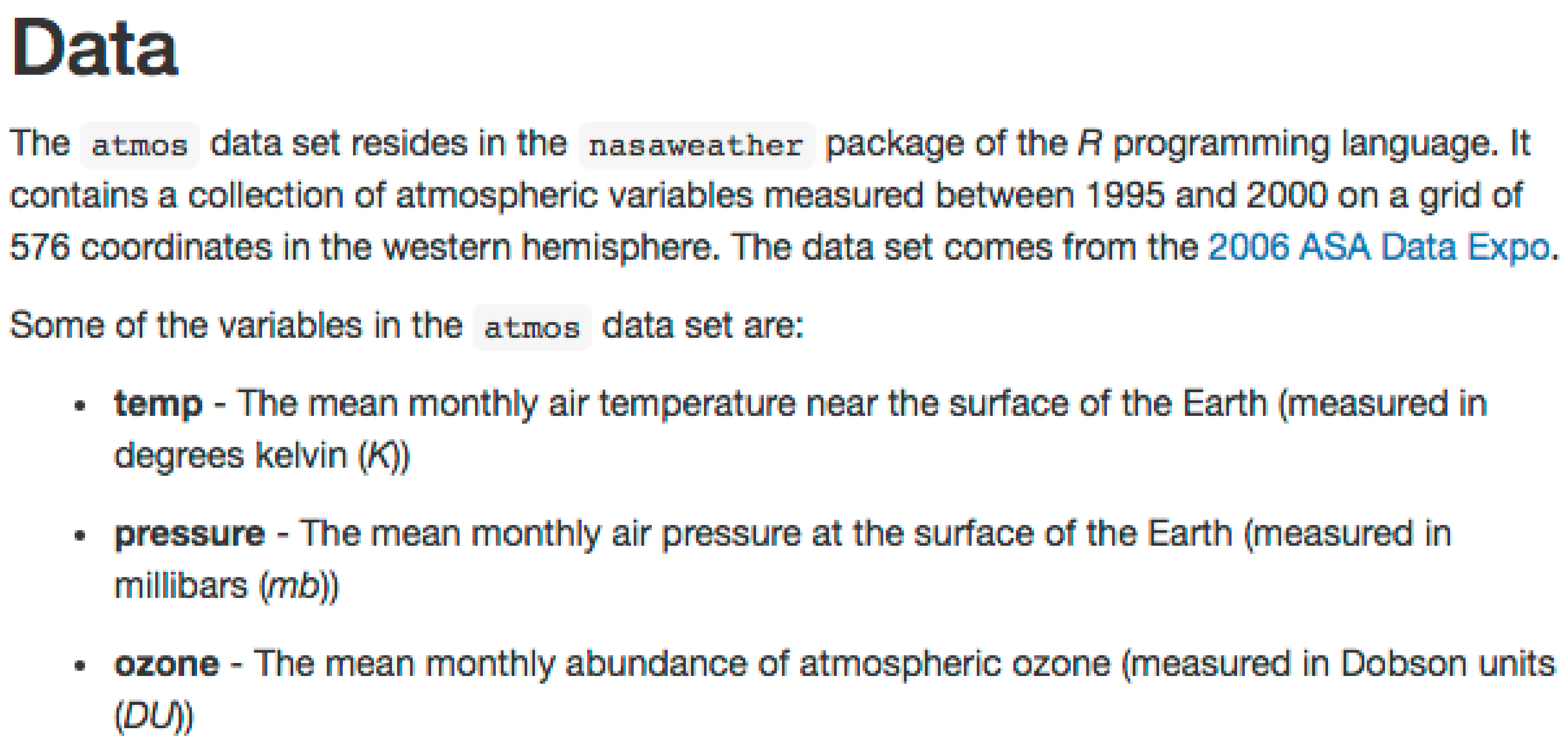
# SOLUTION

7. Make *K*, *mb*, and *DU* italicized at the end of each entry.

*K*

*mb*

*DU*



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

# HOW TO: FORMULAS

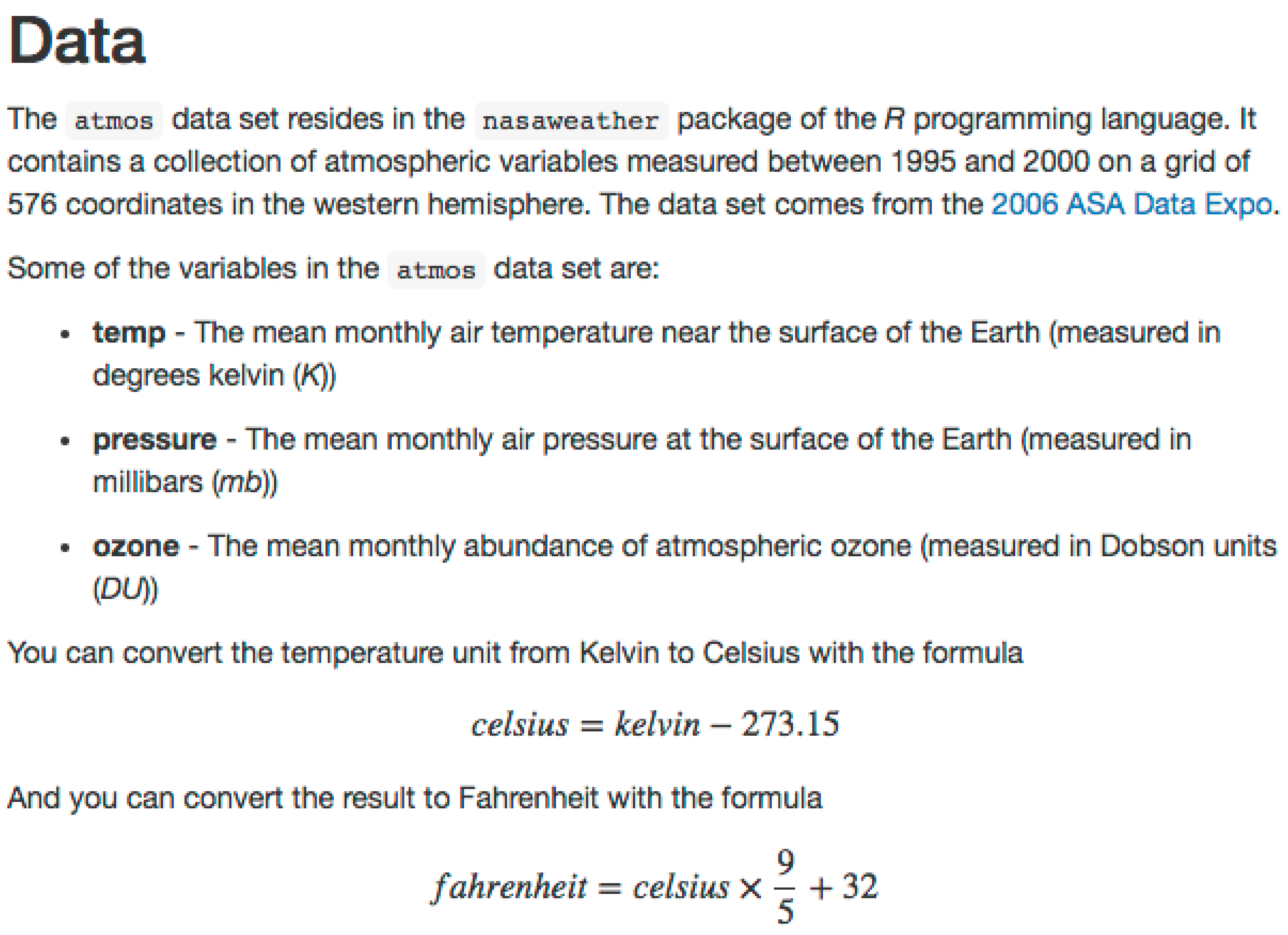
- Multiple markdown syntaxes exist to place inline formulas and to display formulas (centered on page on its own line)
- LaTeX formulae:
  - Surround formula with one $ sign to create an inline formula
  - Surround formula with two $ signs to display a formula

```
# create inline formula
$\forall x \in X, \quad \exists y \leq \epsilon$

# create displayed formula
$$\cos (2\theta) = \cos^2 \theta - \sin^2 \theta$$
```

# YOUR TURN!

8. Create a function that illustrates the conversion from Kelvin to degrees Celsius:

$$celsius = kelvin - 273.15$$



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

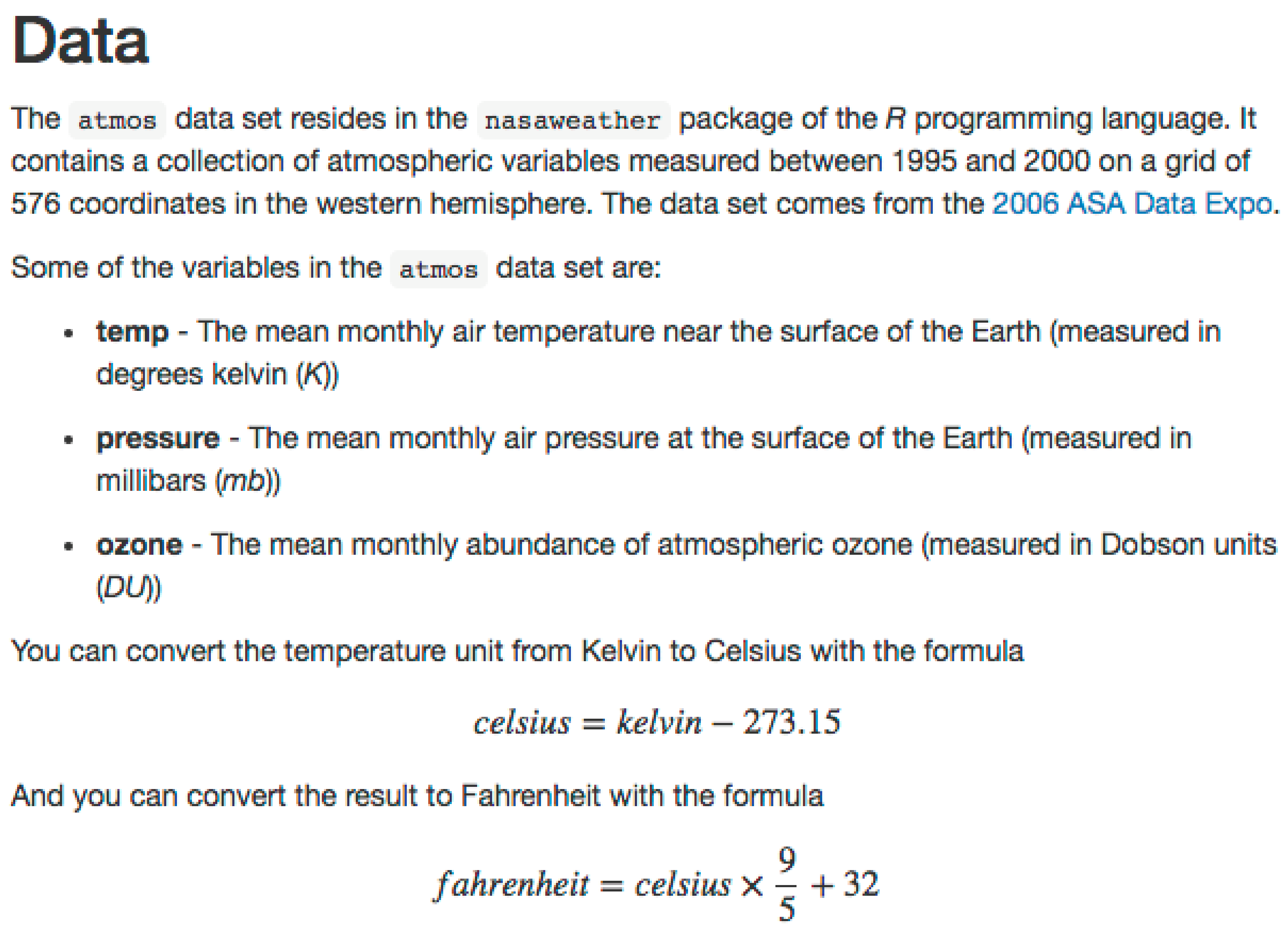
$$fahrenheit = celsius \times \frac{9}{5} + 32$$

# SOLUTION

8. Create a function that illustrates the conversion from Kelvin to degrees Celsius:

$$celsius = kelvin - 273.15$$

*$$celsius = kelvin - 273.15$$*



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

# HOW TO: WHAT IS A CODE CHUNK?

- A code chunk runs code (this is a markdown principle in general, not just for R Markdown)
- Necessary to run code in an R Markdown file
- Three ways to insert code chunk:
  - ✓ Keyboard shortcut!  Ctrl + Alt + I
  - ✓ "Insert" button in the toolbar
  - ✓ Manually typing ```{r}  ```

~~~
# code chunk example
# What will this code chunk do?
```{r}
summary(cars)
```
~~~

# HOW TO: CODE CHUNK NAMES

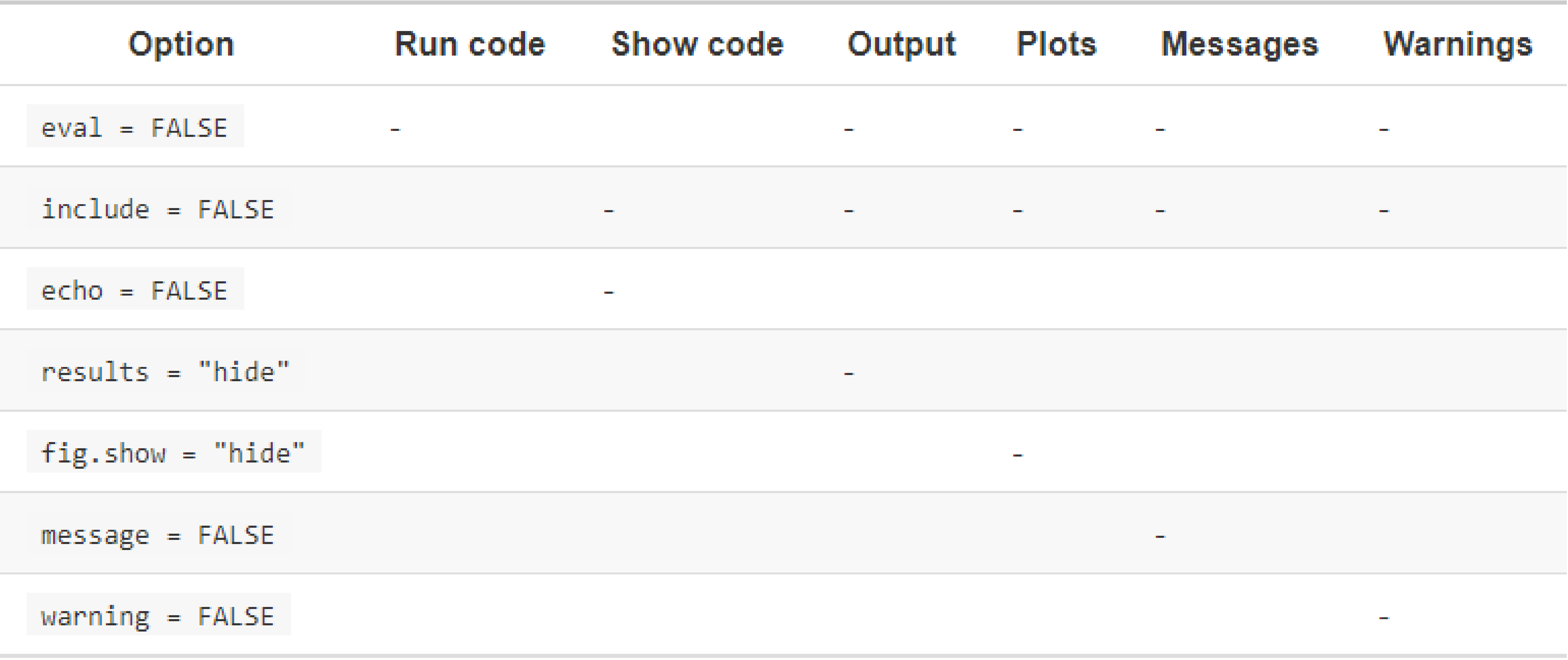
- Can give code chunk an optional name
- Advantages of naming code chunks:
    - Navigate to specific code chunks easily with the drop-down code navigator in the bottom-left of the script editor
    - Graphs made from chunks will be easier to use elsewhere
    - Can cache chunks to knitting time later
- Place code chunk name inside {} after the letter r

~~~
# code chunk example
# What will this code chunk do?
```{r cars-summary}
summary(cars)
```
~~~

# HOW TO: CODE CHUNK OPTIONS

- Customize code chunks with options that are placed in the chunk header
- There are over 60 options!  The most important options are below.

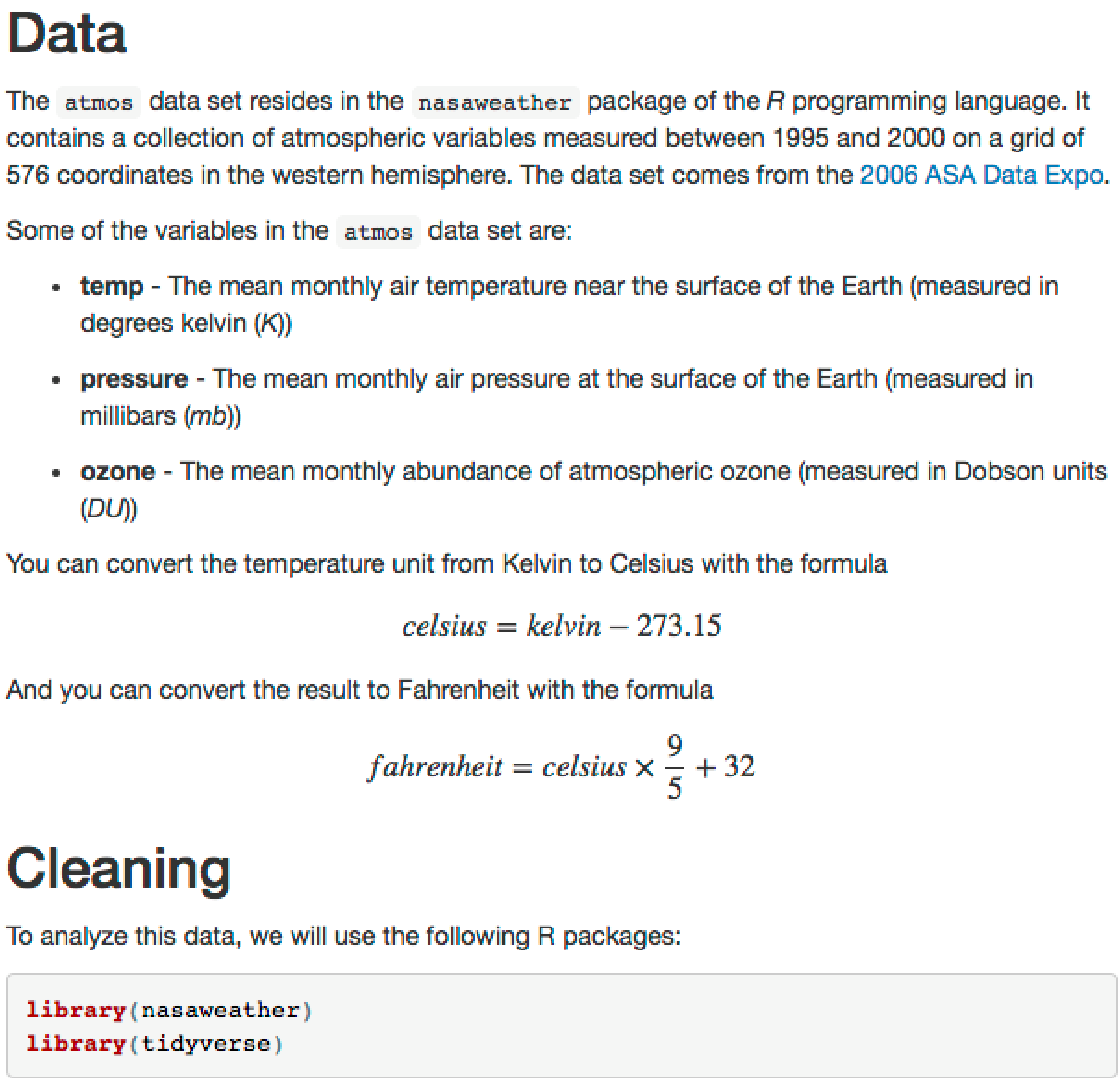
~~~
# code chunk example with option
# What will this code chunk option do?
```{r cars-summary, echo = FALSE, warning = FALSE}
summary(cars)
```
~~~



| Option | Run code | Show code | Output | Plots | Messages | Warnings |
|---|---|---|---|---|---|---|
| `eval = FALSE` | - |  | - | - | - | - |
| `include = FALSE` |  | - | - | - | - | - |
| `echo = FALSE` |  | - |  |  |  |  |
| `results = "hide"` |  |  | - |  |  |  |
| `fig.show = "hide"` |  |  |  | - |  |  |
| `message = FALSE` |  |  |  |  | - |  |
| `warning = FALSE` |  |  |  |  |  | - |

# YOUR TURN!

9. Add code chunk 1 (R packages)
   - Set message=FALSE so messages are not produced in your report when you load the packages
10. Add code chunk 2 (year being analyzed)
   - Set echo=FALSE so this code runs but is not visible to the reader



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

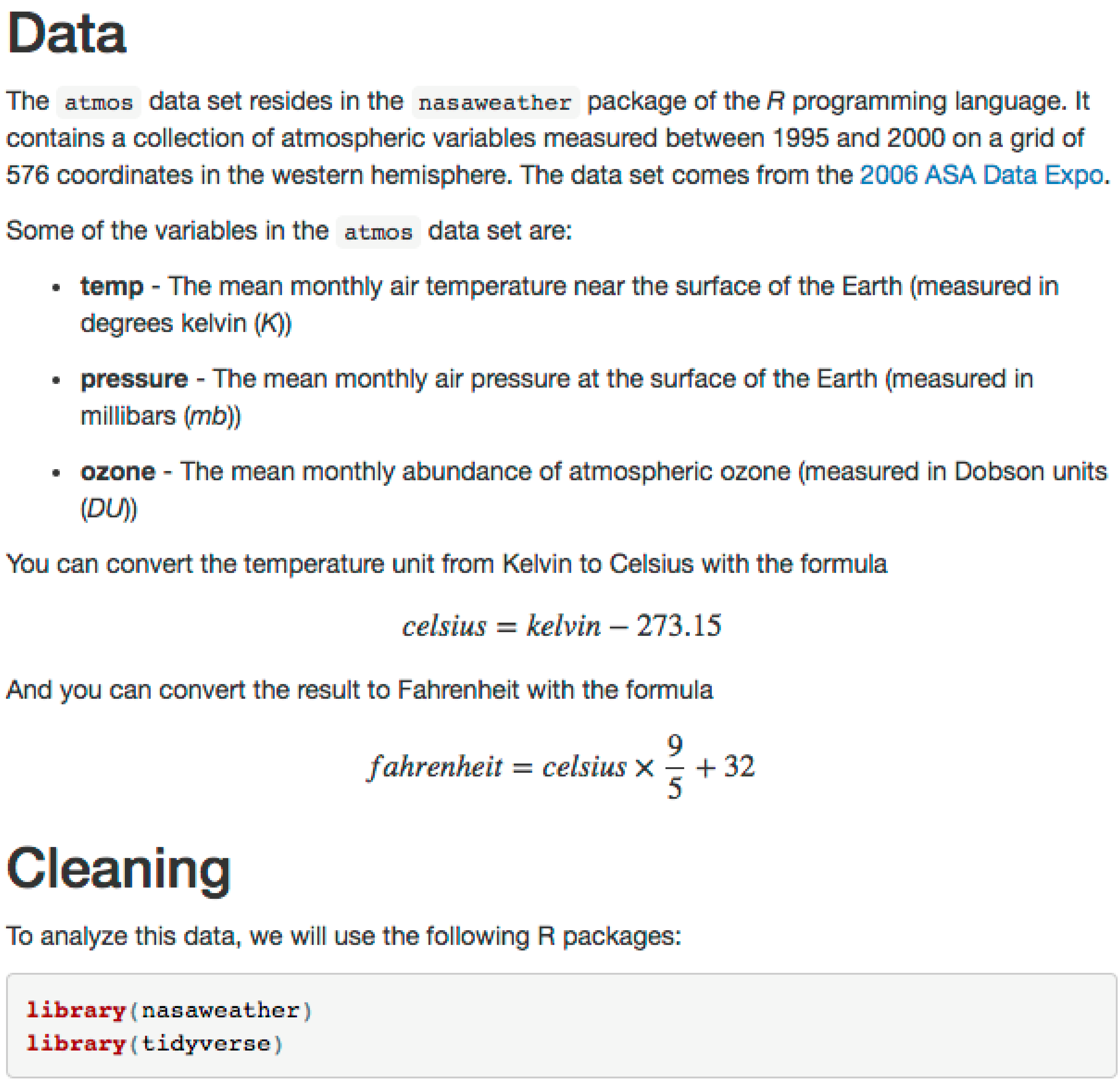
```
library(nasaweather)
library(tidyverse)
```

# SOLUTION

9. Add code chunk 1 (R packages)

- Set message=FALSE so messages are not produced in your report when you load the packages

~~~
```{r message = FALSE}
library(nasaweather)
library(tidyverse)
```
~~~



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

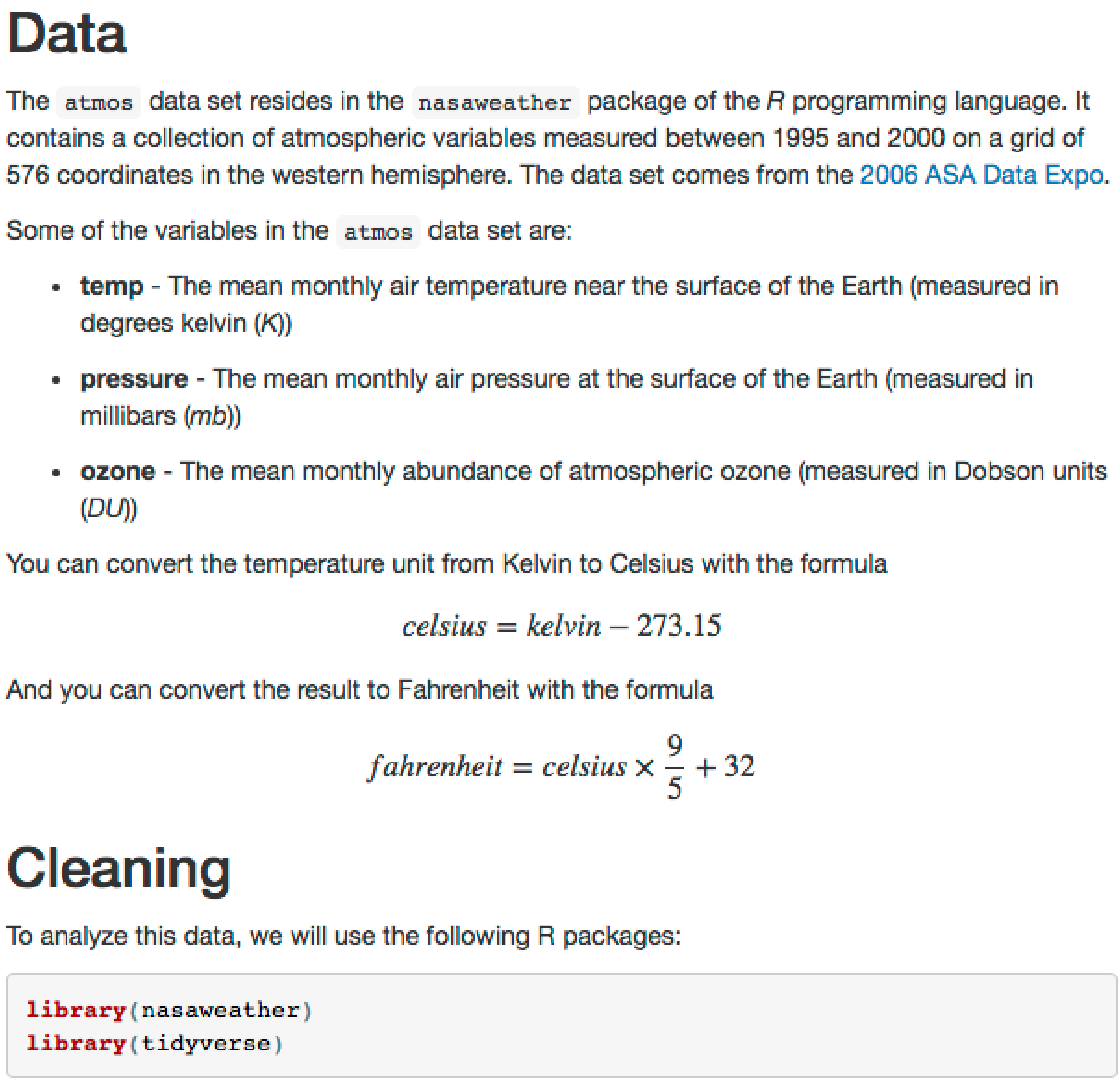
```
library(nasaweather)
library(tidyverse)
```

# SOLUTION

10. Add code chunk 2 (year being analyzed)

- Set echo=FALSE so this code runs but is not visible to the reader

~~~
```{r echo = FALSE}
year <- 1995
```
~~~



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

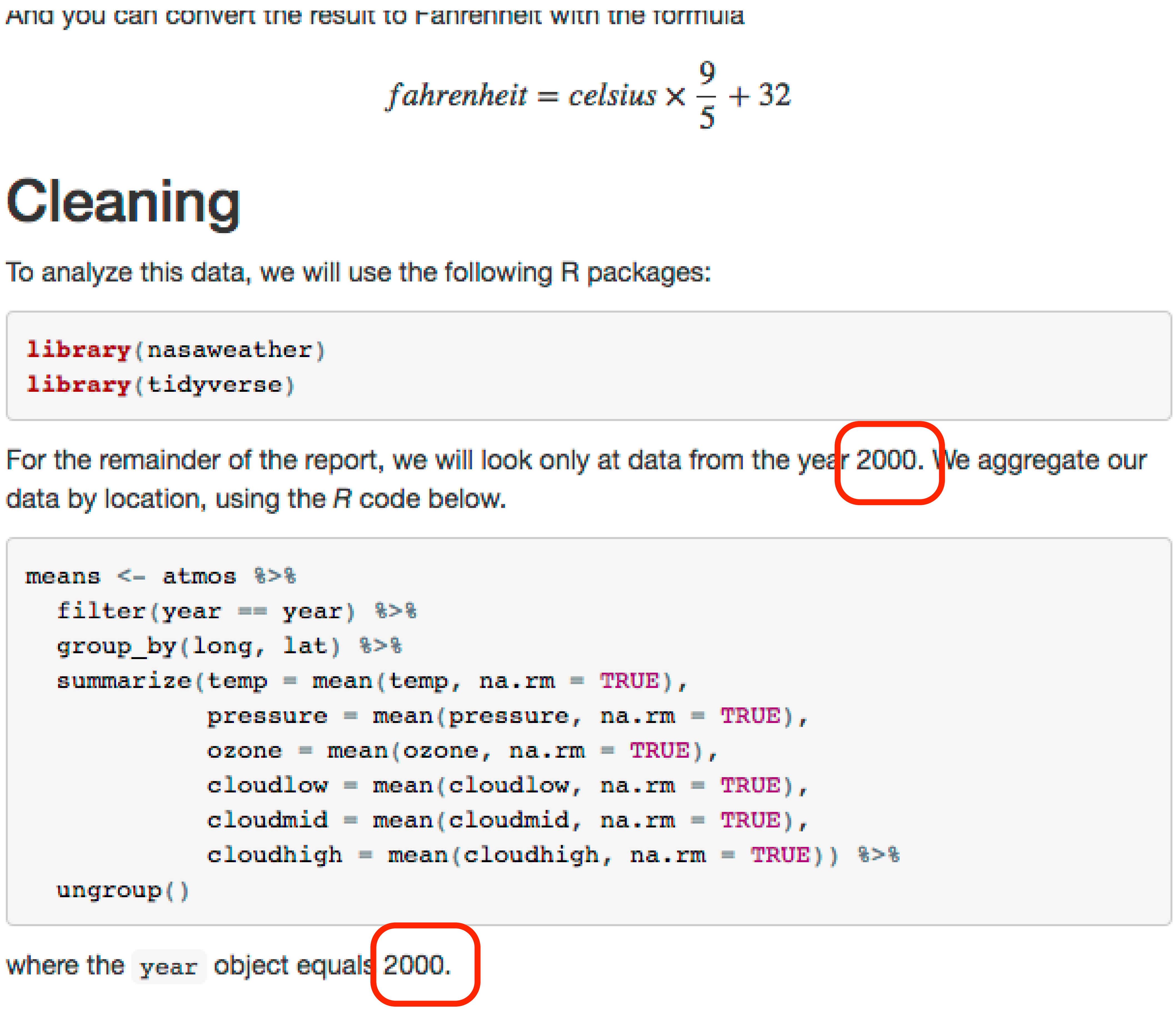
```
library(nasaweather)
library(tidyverse)
```

# YOUR TURN!

11. Add an inline code chunk to reference the year being analyzed (code chunk 3 & 5)

12. Add code chunk 4 and make it visible to the reader

13. Knit to HTML, change the year being analyzed to 2000 and re-knit document. Note how the inline code chunks change.



And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

```
library(nasaweather)
library(tidyverse)
```

For the remainder of the report, we will look only at data from the year 2000. We aggregate our data by location, using the *R* code below.

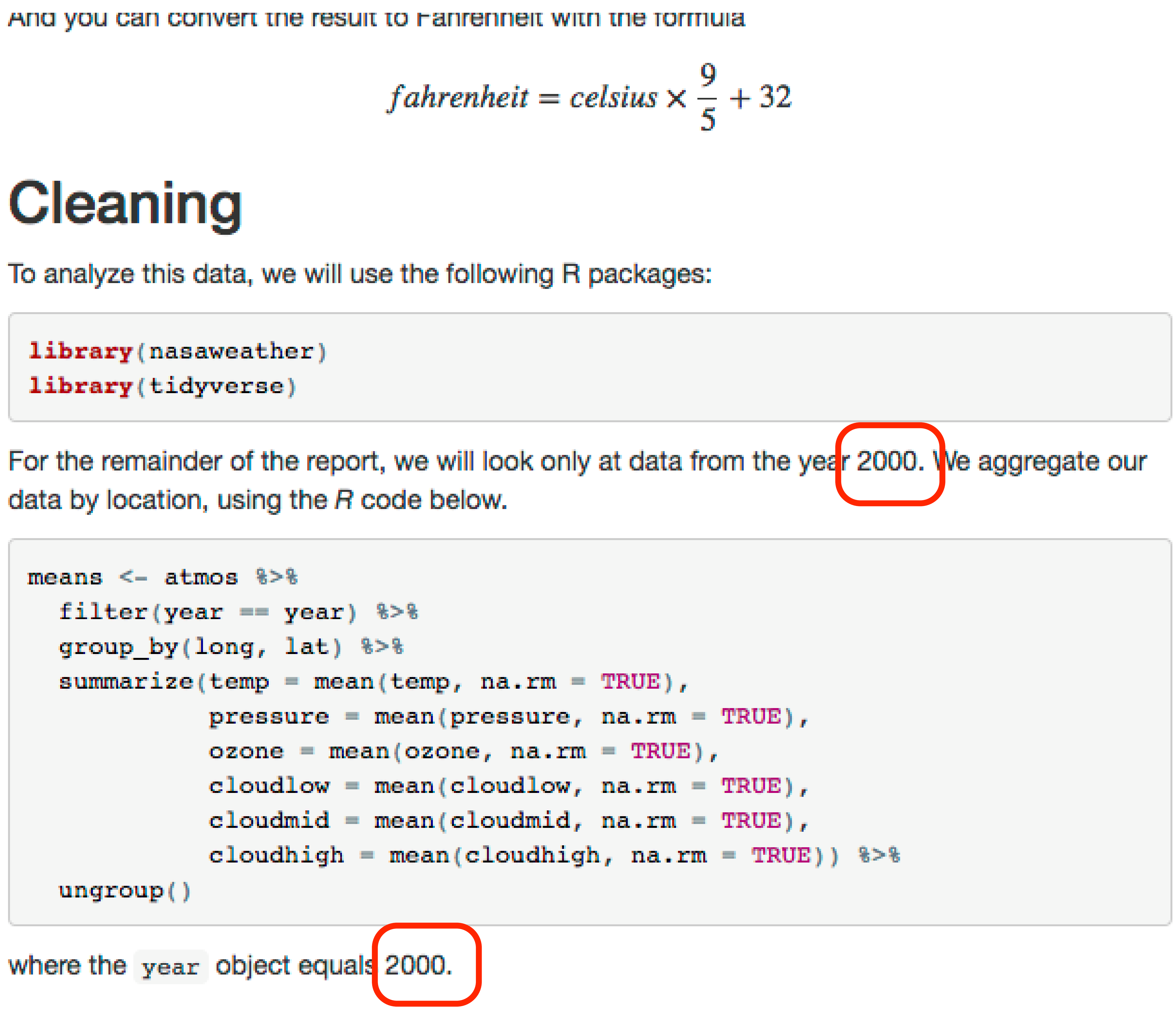
```
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```

where the `year` object equals 2000.

# SOLUTION

11. Add an inline code chunk to reference the year being analyzed (code chunk 3 & 5).

data from the year `r year = 2000` `r year`



And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

```
library(nasaweather)
library(tidyverse)
```

For the remainder of the report, we will look only at data from the year 2000. We aggregate our data by location, using the *R* code below.

```
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```

where the `year` object equals 2000.

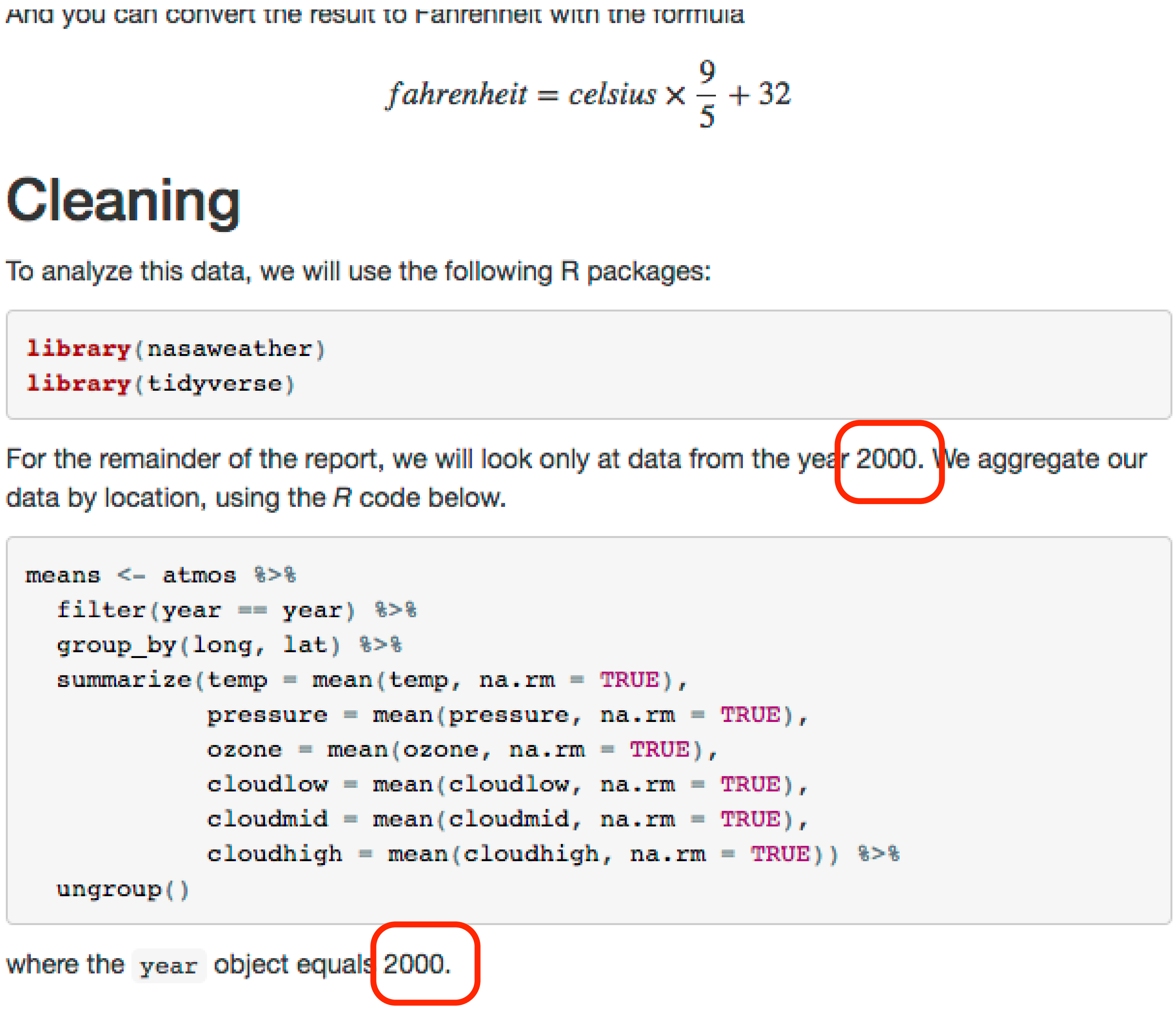
# SOLUTION

12. Add code chunk 4 and make it visible to the reader.

~~~
```{r message = FALSE, warning = FALSE}
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```
~~~

# SOLUTION

13. Knit to HTML, change the year being analyzed to 2000 and re-knit document.  Note how the inline code chunks change.



And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

```
library(nasaweather)
library(tidyverse)
```

For the remainder of the report, we will look only at data from the year 2000. We aggregate our data by location, using the *R* code below.

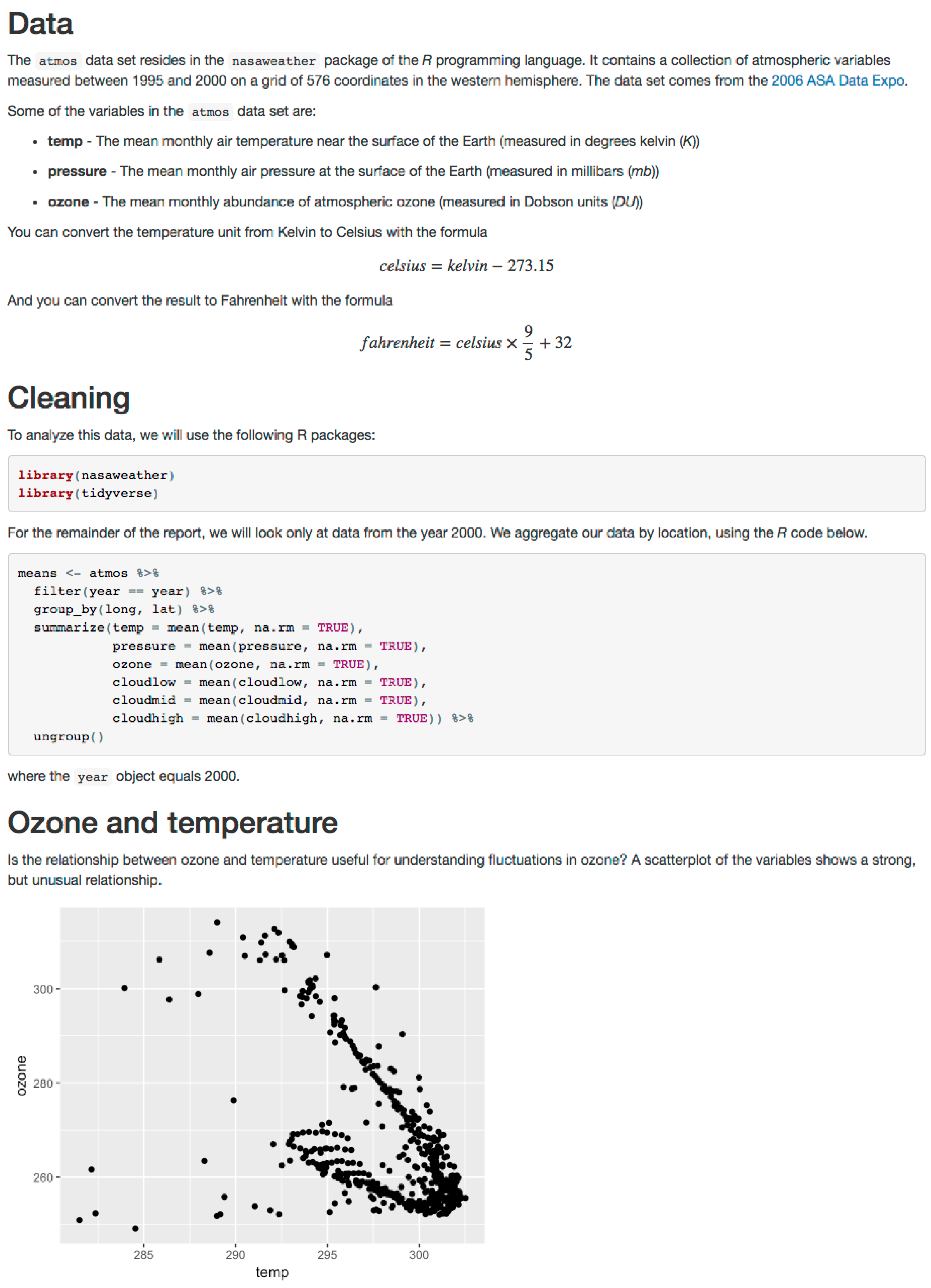
```
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```

where the `year` object equals 2000.

# YOUR TURN!

14. Add text and code chunk 6 so that the plot output is provided. Set the code chunk options to:

- echo = FALSE
- fig.height = 4
- fig.width = 5
- fig.align = ‘center’



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

```
library(nasaweather)
library(tidyverse)
```

For the remainder of the report, we will look only at data from the year 2000. We aggregate our data by location, using the *R* code below.

```
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```

where the `year` object equals 2000.

## Ozone and temperature

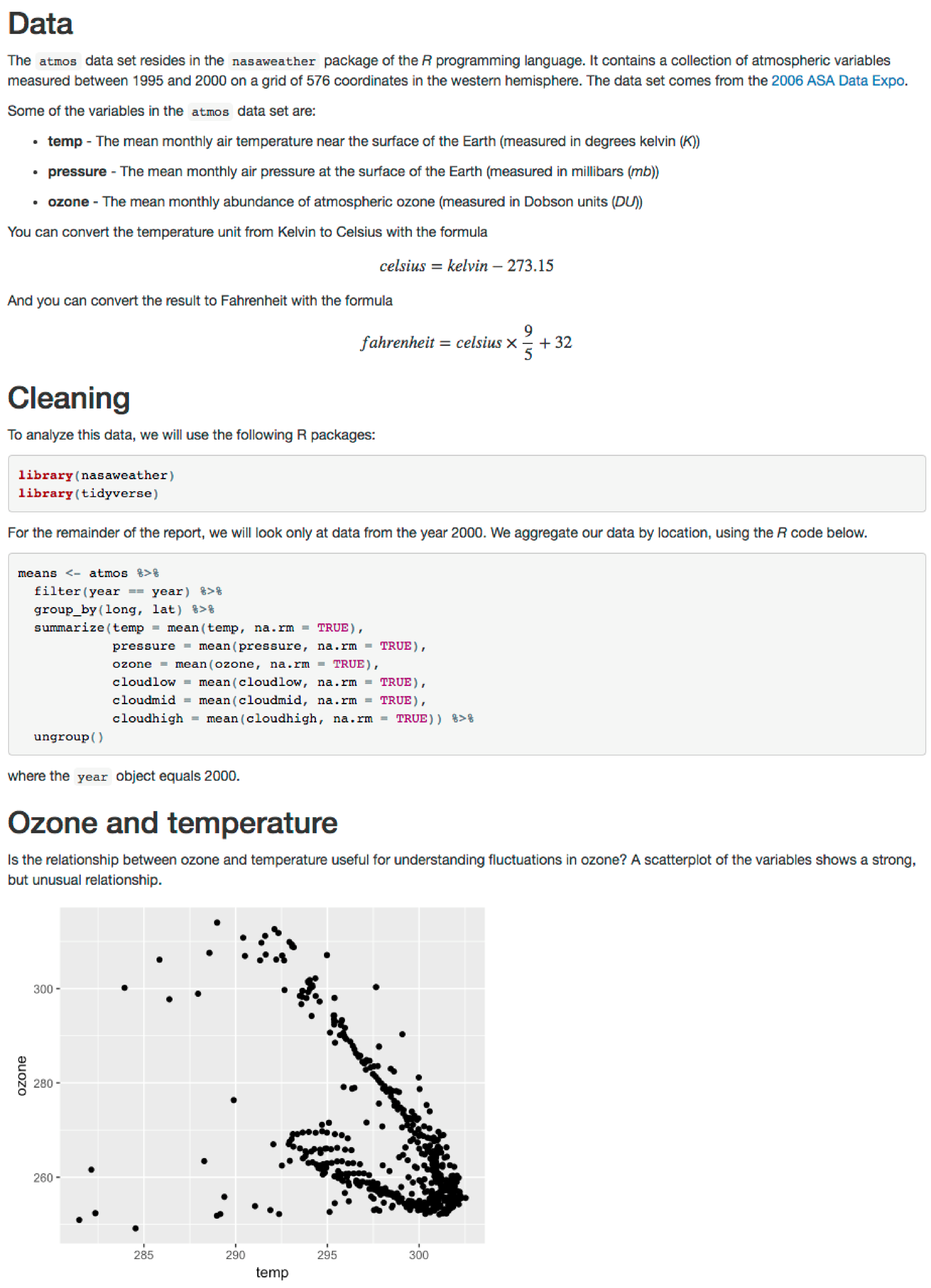
Is the relationship between ozone and temperature useful for understanding fluctuations in ozone? A scatterplot of the variables shows a strong, but unusual relationship.

# SOLUTION

14. Add text and code chunk 6 so that the plot output is provided. Set the code chunk options to:

- echo = FALSE
- fig.height = 4
- fig.width = 5
- fig.align = ‘center’

~~~
```{r echo = FALSE, fig.height = 4, fig.width = 5, fig.align = "center"}
ggplot(data = means, aes(x = temp, y = ozone)) +
  geom_point()
```
~~~



## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

## Cleaning

To analyze this data, we will use the following R packages:

```
library(nasaweather)
library(tidyverse)
```

For the remainder of the report, we will look only at data from the year 2000. We aggregate our data by location, using the *R* code below.

```
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```

where the `year` object equals 2000.

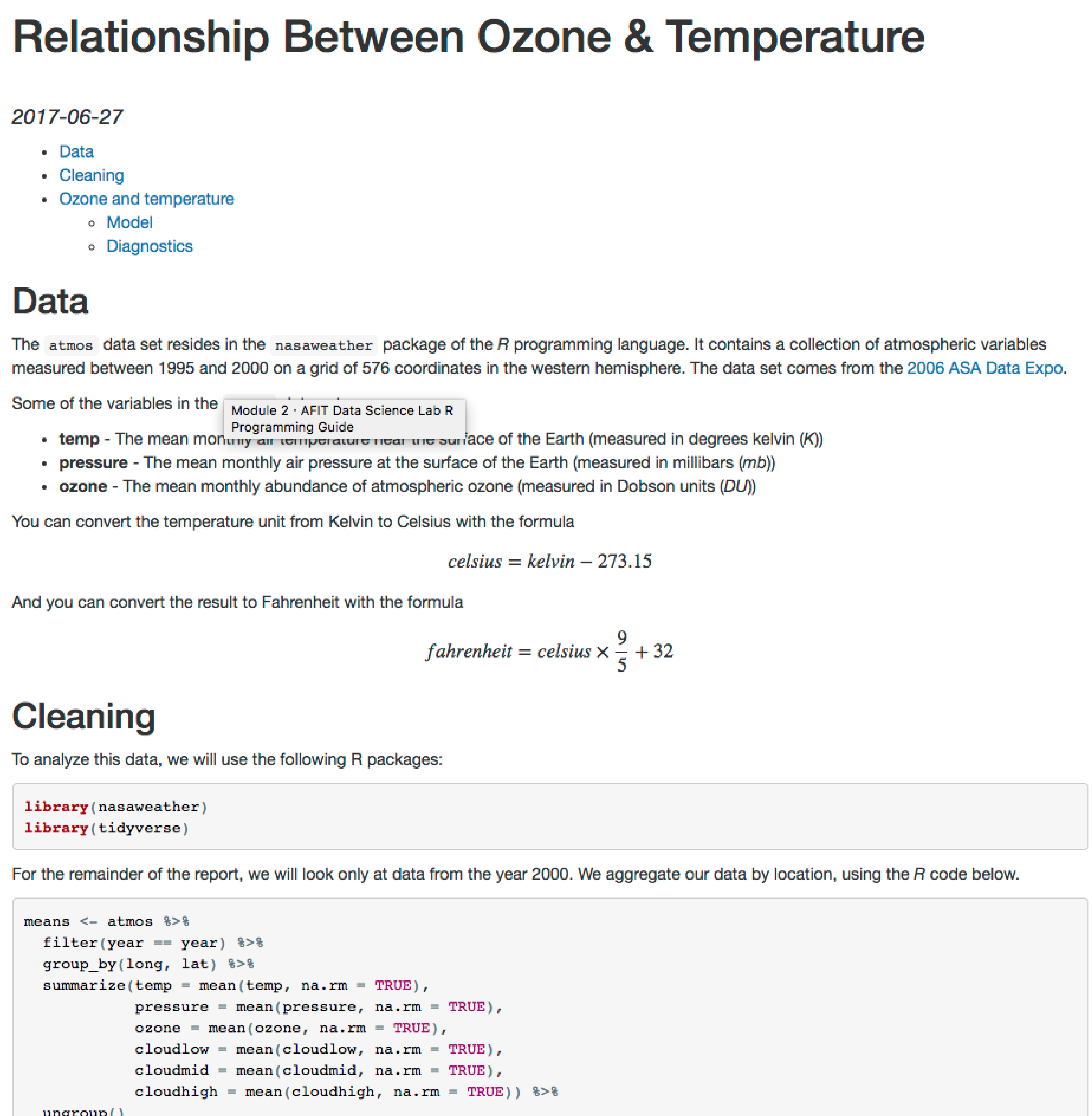
## Ozone and temperature

Is the relationship between ozone and temperature useful for understanding fluctuations in ozone? A scatterplot of the variables shows a strong, but unusual relationship.

# YAML

15. Add remaining text and code chunks
16. Change the YAML header to:

```
---
title: "Relationship Between Ozone & Temperature"
author: "Justin Jodrey"
date: "`r Sys.Date()`"
output:
  html_document:
    toc: true
---
```



## Relationship Between Ozone & Temperature

*2017-06-27*

- Data
- Cleaning
- Ozone and temperature
  - Model
  - Diagnostics

### Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the

Module 2 · AFIT Data Science Lab R Programming Guide

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

And you can convert the result to Fahrenheit with the formula

$$fahrenheit = celsius \times \frac{9}{5} + 32$$

### Cleaning

To analyze this data, we will use the following R packages:

```
library(nasaweather)
library(tidyverse)
```

For the remainder of the report, we will look only at data from the year 2000. We aggregate our data by location, using the *R* code below.

```
means <- atmos %>%
  filter(year == year) %>%
  group_by(long, lat) %>%
  summarize(temp = mean(temp, na.rm = TRUE),
            pressure = mean(pressure, na.rm = TRUE),
            ozone = mean(ozone, na.rm = TRUE),
            cloudlow = mean(cloudlow, na.rm = TRUE),
            cloudmid = mean(cloudmid, na.rm = TRUE),
            cloudhigh = mean(cloudhigh, na.rm = TRUE)) %>%
  ungroup()
```

# DIFFERENT OUTPUTS

17. Render as an HTML, PDF, and Word document

18. Change the YAML header to:

```
---
title: "Relationship Between Ozone & Temperature"
author: "Justin Jodrey"
date: "`r Sys.Date()`"
output: slidy_presentation
---
```

Default is to generate slides based on 2nd level headings but you can add *** anywhere and it will create a new slide there.

# HOW TO: FORMAT DATA TABLES

- Use the knitr::kable() function to display a table with additional formatting

- Look at documentation for LOTS of options

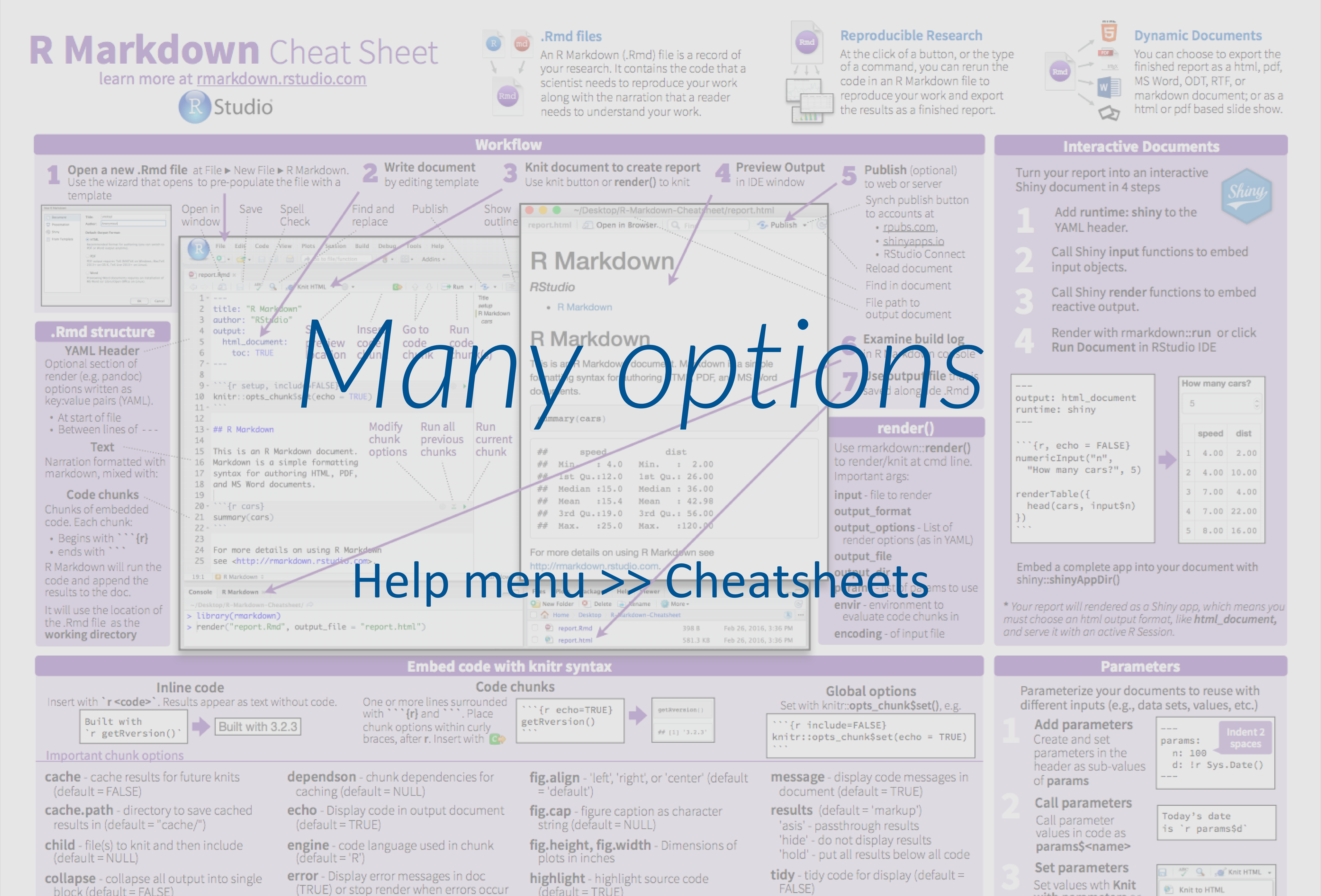
~~~
# code chunk example
# What will this code chunk do?
```{r echo = FALSE}
knitr::kable(
  airquality[1:10, ],
  caption = "What a cool table format!"
)
```
~~~

# HOW TO: CACHE RESULTS

- Use the code chunk option cache = TRUE to cache results, meaning save chunk output for future use
- VERY handy if code take a long time and you need to update/change code
- Can use the dependson = … option to run different code without reloading a massive data set

~~~~
# code chunk example
# What will this code chunk do?
```{r echo = FALSE, cache = TRUE}
data <- read.csv("some_huge_file.csv")
data
```
~~~~

# Many options



# R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

RStudio

## .Rmd files

An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

## Reproducible Research

At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

## Dynamic Documents

You can choose to export the finished report as a html, pdf, MS Word, ODT, RTF, or markdown document; or as a html or pdf based slide show.

## Workflow

1 **Open a new .Rmd file** at File ▸ New File ▸ R Markdown. Use the wizard that opens to pre-populate the file with a template

2 **Write document** by editing template

3 **Knit document to create report** Use knit button or render() to knit

4 **Preview Output** in IDE window

5 **Publish** (optional) to web or server

6 **Examine build log** in R Markdown console

7 **Use output file** that is saved along side .Rmd

~~~
---
title: "R Markdown"
author: "RStudio"
output:
  html_document:
    toc: TRUE
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

## R Markdown

This is an R Markdown document.
Markdown is a simple formatting
syntax for authoring HTML, PDF,
and MS Word documents.

```{r cars}
summary(cars)
```

For more details on using R Markdown
see <http://rmarkdown.rstudio.com>.
```

```
> library(rmarkdown)
> render("report.Rmd", output_file = "report.html")
```

~/Desktop/R-Markdown-Cheatsheet/report.html

R Markdown

*RStudio*

- R Markdown

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

For more details on using R Markdown see http://rmarkdown.rstudio.com.

| | | |
|---|---|---|
| report.Rmd | 398 B | Feb 26, 2016, 3:36 PM |
| report.html | 581.3 KB | Feb 26, 2016, 3:36 PM |

## .Rmd structure

**YAML Header**

Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).

- At start of file
- Between lines of - - -

**Text**

Narration formatted with markdown, mixed with:

**Code chunks**

Chunks of embedded code. Each chunk:

- Begins with ```{r}
- ends with ```

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the **working directory**

## render()

Use rmarkdown::**render()** to render/knit at cmd line. Important args:

**input** - file to render

**output_format**

**output_options** - List of render options (as in YAML)

**output_file**

**output_dir**

**params** - list of params to use

**envir** - environment to evaluate code chunks in

**encoding** - of input file

## Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

1 Add **runtime: shiny** to the YAML header.

2 Call Shiny **input** functions to embed input objects.

3 Call Shiny **render** functions to embed reactive output.

4 Render with rmarkdown::run or click **Run Document** in RStudio IDE

```
---
output: html_document
runtime: shiny
---

```{r, echo = FALSE}
numericInput("n",
  "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```
```

How many cars?

5

| | speed | dist |
|---|---|---|
| 1 | 4.00 | 2.00 |
| 2 | 4.00 | 10.00 |
| 3 | 7.00 | 4.00 |
| 4 | 7.00 | 22.00 |
| 5 | 8.00 | 16.00 |

Embed a complete app into your document with shiny::shinyAppDir()

\* *Your report will rendered as a Shiny app, which means you must choose an html output format, like **html_document**, and serve it with an active R Session.*

## Embed code with knitr syntax

**Inline code**

Insert with `r <code>`. Results appear as text without code.

```
Built with
`r getRversion()`
```

Built with 3.2.3

**Code chunks**

One or more lines surrounded with ```{r} and ```. Place chunk options within curly braces, after r. Insert with

```
```{r echo=TRUE}
getRversion()
```
```

```
getRversion()
## [1] '3.2.3'
```

**Global options**

Set with knitr::**opts_chunk$set()**, e.g.

```
```{r include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

Important chunk options

**cache** - cache results for future knits (default = FALSE)

**cache.path** - directory to save cached results in (default = "cache/")

**child** - file(s) to knit and then include (default = NULL)

**collapse** - collapse all output into single block (default = FALSE)

**dependson** - chunk dependencies for caching (default = NULL)

**echo** - Display code in output document (default = TRUE)

**engine** - code language used in chunk (default = 'R')

**error** - Display error messages in doc (TRUE) or stop render when errors occur

**fig.align** - 'left', 'right', or 'center' (default = 'default')

**fig.cap** - figure caption as character string (default = NULL)

**fig.height, fig.width** - Dimensions of plots in inches

**highlight** - highlight source code (default = TRUE)

**message** - display code messages in document (default = TRUE)

**results** (default = 'markup')
'asis' - passthrough results
'hide' - do not display results
'hold' - put all results below all code

**tidy** - tidy code for display (default = FALSE)

## Parameters

Parameterize your documents to reuse with different inputs (e.g., data sets, values, etc.)

1 **Add parameters**
Create and set parameters in the header as sub-values of **params**

```
---
params:
  n: 100
  d: !r Sys.Date()
---
```

Indent 2 spaces

2 **Call parameters**
Call parameter values in code as **params$<name>**

```
Today's date
is `r params$d`
```

3 **Set parameters**
Set values wth **Knit**

## Help menu >> Cheatsheets
```

# R NOTEBOOK

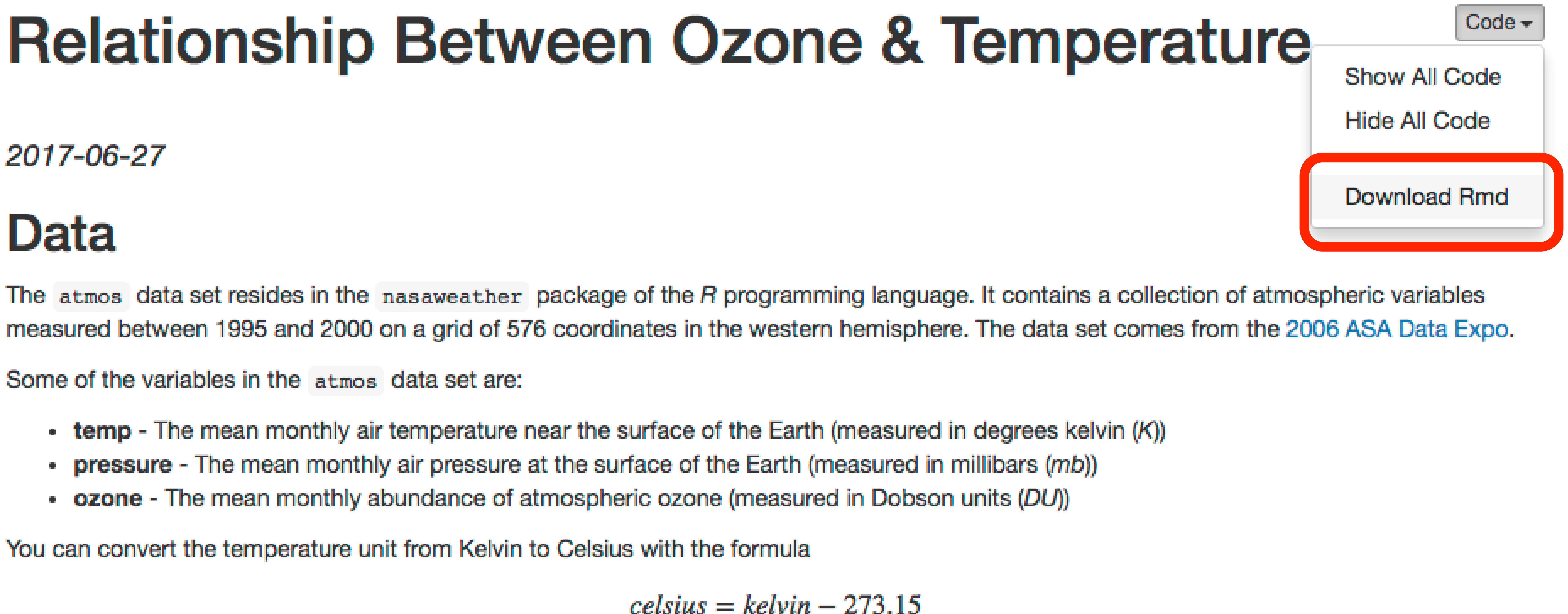
# WHY SO SPECIAL?

*An R Markdown document that allows for independent and interactive execution of code chunks.*

1. *Creates greater interactivity while creating your document*
2. *Easy to share your notebook directly*
3. *When complete, knit to the publication format desired*

# DEMO

- Open *“RMarkdown Demonstration Notebook.nb.html”* file.
- Download Rmd.
- Open Rmd in your RStudio, make changes, knit, and send back to your collaborator.



# Relationship Between Ozone & Temperature

*2017-06-27*

## Data

The `atmos` data set resides in the `nasaweather` package of the *R* programming language. It contains a collection of atmospheric variables measured between 1995 and 2000 on a grid of 576 coordinates in the western hemisphere. The data set comes from the 2006 ASA Data Expo.

Some of the variables in the `atmos` data set are:

- **temp** - The mean monthly air temperature near the surface of the Earth (measured in degrees kelvin (*K*))
- **pressure** - The mean monthly air pressure at the surface of the Earth (measured in millibars (*mb*))
- **ozone** - The mean monthly abundance of atmospheric ozone (measured in Dobson units (*DU*))

You can convert the temperature unit from Kelvin to Celsius with the formula

$$celsius = kelvin - 273.15$$

# WHAT TO REMEMBER

# WHAT YOU SHOULD KNOW

| Topic | Description |
|---|---|
| `R project` | Conveniently organizes files pertaining to specific analytic projects |
| `R Markdown` | Allows user to combine prose, code, and metadata into one file for to increase reproducibility and reporting capabilities |
| `R Notebook` | An R Markdown document that allows for independent and interactive execution of code chunks. |