w03b p2 Coding Expectations and Clean R Code

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

31 January, 2023

Contents

3.2.3.1 Coding Expectations in DSCI courses

- Congratulations! You're working on lab exercise 1 (LE1).

Shortly you'll receive your LE scores and feedback about their submissions.

This document outlines general expectations moving forth.

Some of you who took the DSCI course last semester

- may have seen a similar document.

Please read through this document carefully

- and if you have any questions, do message us on Slack.

3.2.3.1.1 General Expectations

- Please follow these points carefully and work on your lab exercises.

The LE grading will be done based on these points

To reiterate the points:-

- Remember that the knitted PDF/HTML shows the first 80 characters in the code as the output.
- And using Canvas grading, we grade on your pdf
  - Make sure you format your code chunks properly.
  - Please verify this before submission.
- Since we'll be grading your work, be sure to comment your code.
  - Suppose you use a function that is not common (for example, unite()),
  - explain what it does in a short comment
  - and which package you used.
- Best way is to define a function and package as follows
  - `ggplot2::ggsave`
  - This defines the package and function you are using
- It would also help me learn about new functions in the process.
- There are multiple approaches to the same question.
  - we'll take that into account while grading.
- Please answer directly under subquestions (indicated by -).
- If you want to answer everything in one place,
  - say which subquestion you are referring to.
  - For example: Answer to 'what's the highest ranked…': answer here.
- Please answer questions outside the code chunk.
  - Suppose, the question asks you to compare trends,
  - don't type it as a code comment.
- Be creative while you work on plots.
  - ggplot2 package has amazing functionalities.
  - Spend some time formatting plots to make them more understandable.
- For example,
  - adding a plot title,
  - axes labels (specify units wherever required),
  - changing theme specifications will earn full points.
- Include meaningful variable names.
- e.g. for the billboard lyrics dataset,
  - billboard_lyrics is too long and bbl is too short.

    - Find a balance between the two extremes like ‘songs’.

Your see our comments in your lab exercises when graded.

- If there are any questions on these aspects,
- don’t hesitate to contact us on Slack.

3.2.3.1.2 Slack

- From now on, please use Slack as your primary mode of communication with us,
    - especially for asking questions.

Don’t forget to tag us by using @name

- because it’s highly probable that your messages might get lost before the submission.

While asking questions, refer to the question number in your question.

Beaware of threads in the slack channel.

Ask new questions, outside prior hreads

- so that we won’t miss them.

Ask your questions in the class channel #dsci353-353m-453.

- This way others can also benefit and help with your answers.
- It’ll be a win-win situation!

If something has to be privately discussed with us or Prof. French,

- please contact by DM, on 1on1 message in slack

3.2.3.1.3 Office Hours

- We have two office hours in the week:-
    - Mondays and Wednesdays from 4-5 pm.

Please attend the office hours if you have questions and any other concerns.

Generally these sessions would be recorded.

3.2.3.1.4 Submission Time

- Typically lab exercises are expected to be submitted by 11:59 pm of the deadline day.

If you are facing any challenges or problems

- that prevent you from turning in the lab exercise on time,
- please contact us with an explanation of why
    - we’ll see if your request can be accommodated.

3.2.3.1.5 Git Commands

- **NOTE: NOT FOLLOWING THE GIT COMMANDS COULD LEAD TO SEVERAL PROBLEMS.**

I cannot emphasize enough why this is important.

Personal story (some of you might know about this): when I was a DSCI student, I wasn’t regular at git pushing and git pulling.

At one point, I lost all of my files from the ODS VDI and thankfully, it was possible to retrieve after spending a few hours from the CWRU Network.

All of this happened just before the midterm exam!

It was definitely scary.

Please don't land in such situations!

Git pushing and pulling regularly will save your work virtually and in events of your computer crashing with no particular reason, you can always clone your fork.

If you are not doing it already, please make it a habit.

3.2.3.1.6 Markov Data Science Cluster, vs ODS Win10 Desktop Versus Local Computer

- There are several reasons why we encourage you to work on the ODS VDI for a couple of lab exercises.

This is because the packages would be updated from time to time and it's easier to reach out to help@case.edu about the problems.

In fact, we can try diagnosing the issues from our Markov too.

In such cases, make sure you have the Git set up

- and do git pushing/pulling regularly.

If you ever face any issue, please contact us.

3.2.3.1.7 Coding Style

- Code styling is important!
  - As data scientists, part of our job is effectively communicating our process and results to a broader audience.

We expect your code to be

- compact,
- well-commented and
- effectively answers the question you are trying to solve.

There needs to be a whitespace between symbols and characters.

- `hits %>% filter(Artist=="madonna")` is bad style
- `hits %>% filter(Artist == "madonna")` is good style

There are multiple methods to solve a question

- and new approaches to solving a question would always be appreciated.

If you are using a new function that is not regularly used,

- please use package::function approach and say what it does.

Thus, going forward we will

- scrutinize the following code style things
- and points will be deducted accordingly.

NOTE

When looking at the `.Rmd` file you can notice something interesting.

Every sentence I write is ended by a **PERIOD** a **SPACE** and a **ENTER** key.

Although it creates a new line of code,

- when it is knitted they will all be in one block of text.

To make a separate paragraph

- requires a blank line

3.2.3.1.8 Code Chunks

- We will first be more closely looking at the outputs of code chunks.

Let's take a look at the following.

```
library(tidyverse)
```

```
## Error: package or namespace load failed for 'tidyverse' in loadNamespace(i, c(lib.loc, .libPaths()),
##  namespace 'vctrs' 0.5.1 is already loaded, but >= 0.5.2 is required
```

```
library(palmerpenguins)

df<-palmerpenguins::penguins

as.data.frame(df)
```

```
##       species    island bill_length_mm bill_depth_mm flipper_length_mm
## 1      Adelie Torgersen           39.1          18.7               181
## 2      Adelie Torgersen           39.5          17.4               186
## 3      Adelie Torgersen           40.3          18.0               195
## 4      Adelie Torgersen             NA            NA                NA
## 5      Adelie Torgersen           36.7          19.3               193
## 6      Adelie Torgersen           39.3          20.6               190
## 7      Adelie Torgersen           38.9          17.8               181
## 8      Adelie Torgersen           39.2          19.6               195
## 9      Adelie Torgersen           34.1          18.1               193
## 10     Adelie Torgersen           42.0          20.2               190
## 11     Adelie Torgersen           37.8          17.1               186
## 12     Adelie Torgersen           37.8          17.3               180
## 13     Adelie Torgersen           41.1          17.6               182
## 14     Adelie Torgersen           38.6          21.2               191
## 15     Adelie Torgersen           34.6          21.1               198
## 16     Adelie Torgersen           36.6          17.8               185
## 17     Adelie Torgersen           38.7          19.0               195
## 18     Adelie Torgersen           42.5          20.7               197
## 19     Adelie Torgersen           34.4          18.4               184
## 20     Adelie Torgersen           46.0          21.5               194
## 21     Adelie    Biscoe           37.8          18.3               174
## 22     Adelie    Biscoe           37.7          18.7               180
## 23     Adelie    Biscoe           35.9          19.2               189
## 24     Adelie    Biscoe           38.2          18.1               185
## 25     Adelie    Biscoe           38.8          17.2               180
## 26     Adelie    Biscoe           35.3          18.9               187
## 27     Adelie    Biscoe           40.6          18.6               183
## 28     Adelie    Biscoe           40.5          17.9               187
## 29     Adelie    Biscoe           37.9          18.6               172
## 30     Adelie    Biscoe           40.5          18.9               180
## 31     Adelie     Dream           39.5          16.7               178
## 32     Adelie     Dream           37.2          18.1               178
## 33     Adelie     Dream           39.5          17.8               188
## 34     Adelie     Dream           40.9          18.9               184
## 35     Adelie     Dream           36.4          17.0               195
## 36     Adelie     Dream           39.2          21.1               196
## 37     Adelie     Dream           38.8          20.0               190
## 38     Adelie     Dream           42.2          18.5               180
## 39     Adelie     Dream           37.6          19.3               181
## 40     Adelie     Dream           39.8          19.1               184
## 41     Adelie     Dream           36.5          18.0               182
## 42     Adelie     Dream           40.8          18.4               195
## 43     Adelie     Dream           36.0          18.5               186
## 44     Adelie     Dream           44.1          19.7               196
## 45     Adelie     Dream           37.0          16.9               185
## 46     Adelie     Dream           39.6          18.8               190
## 47     Adelie     Dream           41.1          19.0               182
## 48     Adelie     Dream           37.5          18.9               179
## 49     Adelie     Dream           36.0          17.9               190
## 50     Adelie     Dream           42.3          21.2               191
## 51     Adelie    Biscoe           39.6          17.7               186
## 52     Adelie    Biscoe           40.1          18.9               188
## 53     Adelie    Biscoe           35.0          17.9               190
## 54     Adelie    Biscoe           42.0          19.5               200
## 55     Adelie    Biscoe           34.5          18.1               187
## 56     Adelie    Biscoe           41.4          18.6               191
## 57     Adelie    Biscoe           39.0          17.5               186
## 58     Adelie    Biscoe           40.6          18.8               193
## 59     Adelie    Biscoe           36.5          16.6               181
## 60     Adelie    Biscoe           37.6          19.1               194
## 61     Adelie    Biscoe           35.7          16.9               185
## 62     Adelie    Biscoe           41.3          21.1               195
## 63     Adelie    Biscoe           37.6          17.0               185
## 64     Adelie    Biscoe           41.1          18.2               192
## 65     Adelie    Biscoe           36.4          17.1               184
## 66     Adelie    Biscoe           41.6          18.0               192
## 67     Adelie    Biscoe           35.5          16.2               195
## 68     Adelie    Biscoe           41.1          19.1               188
## 69     Adelie Torgersen           35.9          16.6               190
## 70     Adelie Torgersen           41.8          19.4               198
## 71     Adelie Torgersen           33.5          19.0               190
## 72     Adelie Torgersen           39.7          18.4               190
## 73     Adelie Torgersen           39.6          17.2               196
## 74     Adelie Torgersen           45.8          18.9               197
## 75     Adelie Torgersen           35.5          17.5               190
## 76     Adelie Torgersen           42.8          18.5               195
## 77     Adelie Torgersen           40.9          16.8               191
## 78     Adelie Torgersen           37.2          19.4               184
## 79     Adelie Torgersen           36.2          16.1               187
## 80     Adelie Torgersen           42.1          19.1               195
## 81     Adelie Torgersen           34.6          17.2               189
## 82     Adelie Torgersen           42.9          17.6               196
## 83     Adelie Torgersen           36.7          18.8               187
## 84     Adelie Torgersen           35.1          19.4               193
## 85     Adelie     Dream           37.3          17.8               191
## 86     Adelie     Dream           41.3          20.3               194
## 87     Adelie     Dream           36.3          19.5               190
## 88     Adelie     Dream           36.9          18.6               189
## 89     Adelie     Dream           38.3          19.2               189
## 90     Adelie     Dream           38.9          18.8               190
## 91     Adelie     Dream           35.7          18.0               202
## 92     Adelie     Dream           41.1          18.1               205
## 93     Adelie     Dream           34.0          17.1               185
## 94     Adelie     Dream           39.6          18.1               186
## 95     Adelie     Dream           36.2          17.3               187
## 96     Adelie     Dream           40.8          18.9               208
## 97     Adelie     Dream           38.1          18.6               190
## 98     Adelie     Dream           40.3          18.5               196
## 99     Adelie     Dream           33.1          16.1               178
## 100    Adelie     Dream           43.2          18.5               192
## 101    Adelie    Biscoe           35.0          17.9               192
## 102    Adelie    Biscoe           41.0          20.0               203
## 103    Adelie    Biscoe           37.7          16.0               183
## 104    Adelie    Biscoe           37.8          20.0               190
## 105    Adelie    Biscoe           37.9          18.6               193
## 106    Adelie    Biscoe           39.7          18.9               184
## 107    Adelie    Biscoe           38.6          17.2               199
## 108    Adelie    Biscoe           38.2          20.0               190
## 109    Adelie    Biscoe           38.1          17.0               181
## 110    Adelie    Biscoe           43.2          19.0               197
## 111    Adelie    Biscoe           38.1          16.5               198
## 112    Adelie    Biscoe           45.6          20.3               191
## 113    Adelie    Biscoe           39.7          17.7               193
## 114    Adelie    Biscoe           42.2          19.5               197
## 115    Adelie    Biscoe           39.6          20.7               191
## 116    Adelie    Biscoe           42.7          18.3               196
## 117    Adelie Torgersen           38.6          17.0               188
## 118    Adelie Torgersen           37.3          20.5               199
## 119    Adelie Torgersen           35.7          17.0               189
## 120    Adelie Torgersen           41.1          18.6               189
## 121    Adelie Torgersen           36.2          17.2               187
## 122    Adelie Torgersen           37.7          19.8               198
## 123    Adelie Torgersen           40.2          17.0               176
## 124    Adelie Torgersen           41.4          18.5               202
## 125    Adelie Torgersen           35.2          15.9               186
## 126    Adelie Torgersen           40.6          19.0               199
## 127    Adelie Torgersen           38.8          17.6               191
## 128    Adelie Torgersen           41.5          18.3               195
## 129    Adelie Torgersen           39.0          17.1               191
## 130    Adelie Torgersen           44.1          18.0               210
## 131    Adelie Torgersen           38.5          17.9               190
## 132    Adelie Torgersen           43.1          19.2               197
## 133    Adelie     Dream           36.8          18.5               193
## 134    Adelie     Dream           37.5          18.5               199
## 135    Adelie     Dream           38.1          17.6               187
## 136    Adelie     Dream           41.1          17.5               190
## 137    Adelie     Dream           35.6          17.5               191
## 138    Adelie     Dream           40.2          20.1               200
## 139    Adelie     Dream           37.0          16.5               185
## 140    Adelie     Dream           39.7          17.9               193
## 141    Adelie     Dream           40.2          17.1               193
## 142    Adelie     Dream           40.6          17.2               187
## 143    Adelie     Dream           32.1          15.5               188
## 144    Adelie     Dream           40.7          17.0               190
## 145    Adelie     Dream           37.3          16.8               192
## 146    Adelie     Dream           39.0          18.7               185
## 147    Adelie     Dream           39.2          18.6               190
## 148    Adelie     Dream           36.6          18.4               184
## 149    Adelie     Dream           36.0          17.8               195
## 150    Adelie     Dream           37.8          18.1               193
## 151    Adelie     Dream           36.0          17.1               187
## 152    Adelie     Dream           41.5          18.5               201
## 153    Gentoo    Biscoe           46.1          13.2               211
## 154    Gentoo    Biscoe           50.0          16.3               230
## 155    Gentoo    Biscoe           48.7          14.1               210
## 156    Gentoo    Biscoe           50.0          15.2               218
## 157    Gentoo    Biscoe           47.6          14.5               215
## 158    Gentoo    Biscoe           46.5          13.5               210
## 159    Gentoo    Biscoe           45.4          14.6               211
## 160    Gentoo    Biscoe           46.7          15.3               219
## 161    Gentoo    Biscoe           43.3          13.4               209
## 162    Gentoo    Biscoe           46.8          15.4               215
## 163    Gentoo    Biscoe           40.9          13.7               214
## 164    Gentoo    Biscoe           49.0          16.1               216
## 165    Gentoo    Biscoe           45.5          13.7               214
## 166    Gentoo    Biscoe           48.4          14.6               213
## 167    Gentoo    Biscoe           45.8          14.6               210
## 168    Gentoo    Biscoe           49.3          15.7               217
## 169    Gentoo    Biscoe           42.0          13.5               210
## 170    Gentoo    Biscoe           49.2          15.2               221
## 171    Gentoo    Biscoe           46.2          14.5               209
## 172    Gentoo    Biscoe           48.7          15.1               222
## 173    Gentoo    Biscoe           50.2          14.3               218
## 174    Gentoo    Biscoe           45.1          14.5               215
## 175    Gentoo    Biscoe           46.5          14.5               213
## 176    Gentoo    Biscoe           46.3          15.8               215
## 177    Gentoo    Biscoe           42.9          13.1               215
## 178    Gentoo    Biscoe           46.1          15.1               215
## 179    Gentoo    Biscoe           44.5          14.3               216
## 180    Gentoo    Biscoe           47.8          15.0               215
## 181    Gentoo    Biscoe           48.2          14.3               210
## 182    Gentoo    Biscoe           50.0          15.3               220
## 183    Gentoo    Biscoe           47.3          15.3               222
## 184    Gentoo    Biscoe           42.8          14.2               209
## 185    Gentoo    Biscoe           45.1          14.5               207
## 186    Gentoo    Biscoe           59.6          17.0               230
## 187    Gentoo    Biscoe           49.1          14.8               220
## 188    Gentoo    Biscoe           48.4          16.3               220
## 189    Gentoo    Biscoe           42.6          13.7               213
## 190    Gentoo    Biscoe           44.4          17.3               219
## 191    Gentoo    Biscoe           44.0          13.6               208
## 192    Gentoo    Biscoe           48.7          15.7               208
## 193    Gentoo    Biscoe           42.7          13.7               208
## 194    Gentoo    Biscoe           49.6          16.0               225
## 195    Gentoo    Biscoe           45.3          13.7               210
## 196    Gentoo    Biscoe           49.6          15.0               216
## 197    Gentoo    Biscoe           50.5          15.9               222
## 198    Gentoo    Biscoe           43.6          13.9               217
## 199    Gentoo    Biscoe           45.5          13.9               210
## 200    Gentoo    Biscoe           50.5          15.9               225
## 201    Gentoo    Biscoe           44.9          13.3               213
## 202    Gentoo    Biscoe           45.2          15.8               215
## 203    Gentoo    Biscoe           46.6          14.2               210
## 204    Gentoo    Biscoe           48.5          14.1               220
## 205    Gentoo    Biscoe           45.1          14.4               210
## 206    Gentoo    Biscoe           50.1          15.0               225
## 207    Gentoo    Biscoe           46.5          14.4               217
## 208    Gentoo    Biscoe           45.0          15.4               220
## 209    Gentoo    Biscoe           43.8          13.9               208
## 210    Gentoo    Biscoe           45.5          15.0               220
## 211    Gentoo    Biscoe           43.2          14.5               208
## 212    Gentoo    Biscoe           50.4          15.3               224
## 213    Gentoo    Biscoe           45.3          13.8               208
## 214    Gentoo    Biscoe           46.2          14.9               221
## 215    Gentoo    Biscoe           45.7          13.9               214
## 216    Gentoo    Biscoe           54.3          15.7               231
## 217    Gentoo    Biscoe           45.8          14.2               219
## 218    Gentoo    Biscoe           49.8          16.8               230
## 219    Gentoo    Biscoe           46.2          14.4               214
## 220    Gentoo    Biscoe           49.5          16.2               229
## 221    Gentoo    Biscoe           43.5          14.2               220
## 222    Gentoo    Biscoe           50.7          15.0               223
## 223    Gentoo    Biscoe           47.7          15.0               216
## 224    Gentoo    Biscoe           46.4          15.6               221
## 225    Gentoo    Biscoe           48.2          15.6               221
## 226    Gentoo    Biscoe           46.5          14.8               217
## 227    Gentoo    Biscoe           46.4          15.0               216
## 228    Gentoo    Biscoe           48.6          16.0               230
## 229    Gentoo    Biscoe           47.5          14.2               209
## 230    Gentoo    Biscoe           51.1          16.3               220
## 231    Gentoo    Biscoe           45.2          13.8               215
## 232    Gentoo    Biscoe           45.2          16.4               223
## 233    Gentoo    Biscoe           49.1          14.5               212
## 234    Gentoo    Biscoe           52.5          15.6               221
## 235    Gentoo    Biscoe           47.4          14.6               212
## 236    Gentoo    Biscoe           50.0          15.9               224
## 237    Gentoo    Biscoe           44.9          13.8               212
## 238    Gentoo    Biscoe           50.8          17.3               228
## 239    Gentoo    Biscoe           43.4          14.4               218
## 240    Gentoo    Biscoe           51.3          14.2               218
## 241    Gentoo    Biscoe           47.5          14.0               212
## 242    Gentoo    Biscoe           52.1          17.0               230
## 243    Gentoo    Biscoe           47.5          15.0               218
## 244    Gentoo    Biscoe           52.2          17.1               228
## 245    Gentoo    Biscoe           45.5          14.5               212
## 246    Gentoo    Biscoe           49.5          16.1               224
## 247    Gentoo    Biscoe           44.5          14.7               214
## 248    Gentoo    Biscoe           50.8          15.7               226
## 249    Gentoo    Biscoe           49.4          15.8               216
## 250    Gentoo    Biscoe           46.9          14.6               222
## 251    Gentoo    Biscoe           48.4          14.4               203
## 252    Gentoo    Biscoe           51.1          16.5               225
## 253    Gentoo    Biscoe           48.5          15.0               219
## 254    Gentoo    Biscoe           55.9          17.0               228
## 255    Gentoo    Biscoe           47.2          15.5               215
## 256    Gentoo    Biscoe           49.1          15.0               228
## 257    Gentoo    Biscoe           47.3          13.8               216
## 258    Gentoo    Biscoe           46.8          16.1               215
## 259    Gentoo    Biscoe           41.7          14.7               210
## 260    Gentoo    Biscoe           53.4          15.8               219
## 261    Gentoo    Biscoe           43.3          14.0               208
## 262    Gentoo    Biscoe           48.1          15.1               209
## 263    Gentoo    Biscoe           50.5          15.2               216
## 264    Gentoo    Biscoe           49.8          15.9               229
## 265    Gentoo    Biscoe           43.5          15.2               213
## 266    Gentoo    Biscoe           51.5          16.3               230
## 267    Gentoo    Biscoe           46.2          14.1               217
## 268    Gentoo    Biscoe           55.1          16.0               230
## 269    Gentoo    Biscoe           44.5          15.7               217
## 270    Gentoo    Biscoe           48.8          16.2               222
## 271    Gentoo    Biscoe           47.2          13.7               214
## 272    Gentoo    Biscoe             NA            NA                NA
## 273    Gentoo    Biscoe           46.8          14.3               215
## 274    Gentoo    Biscoe           50.4          15.7               222
## 275    Gentoo    Biscoe           45.2          14.8               212
## 276    Gentoo    Biscoe           49.9          16.1               213
## 277 Chinstrap     Dream           46.5          17.9               192
## 278 Chinstrap     Dream           50.0          19.5               196
## 279 Chinstrap     Dream           51.3          19.2               193
## 280 Chinstrap     Dream           45.4          18.7               188
## 281 Chinstrap     Dream           52.7          19.8               197
## 282 Chinstrap     Dream           45.2          17.8               198
## 283 Chinstrap     Dream           46.1          18.2               178
## 284 Chinstrap     Dream           51.3          18.2               197
## 285 Chinstrap     Dream           46.0          18.9               195
## 286 Chinstrap     Dream           51.3          19.9               198
## 287 Chinstrap     Dream           46.6          17.8               193
## 288 Chinstrap     Dream           51.7          20.3               194
## 289 Chinstrap     Dream           47.0          17.3               185
## 290 Chinstrap     Dream           52.0          18.1               201
## 291 Chinstrap     Dream           45.9          17.1               190
## 292 Chinstrap     Dream           50.5          19.6               201
## 293 Chinstrap     Dream           50.3          20.0               197
## 294 Chinstrap     Dream           58.0          17.8               181
## 295 Chinstrap     Dream           46.4          18.6               190
## 296 Chinstrap     Dream           49.2          18.2               195
## 297 Chinstrap     Dream           42.4          17.3               181
## 298 Chinstrap     Dream           48.5          17.5               191
## 299 Chinstrap     Dream           43.2          16.6               187
## 300 Chinstrap     Dream           50.6          19.4               193
## 301 Chinstrap     Dream           46.7          17.9               195
## 302 Chinstrap     Dream           52.0          19.0               197
## 303 Chinstrap     Dream           50.5          18.4               200
## 304 Chinstrap     Dream           49.5          19.0               200
## 305 Chinstrap     Dream           46.4          17.8               191
## 306 Chinstrap     Dream           52.8          20.0               205
## 307 Chinstrap     Dream           40.9          16.6               187
## 308 Chinstrap     Dream           54.2          20.8               201
## 309 Chinstrap     Dream           42.5          16.7               187
## 310 Chinstrap     Dream           51.0          18.8               203
## 311 Chinstrap     Dream           49.7          18.6               195
## 312 Chinstrap     Dream           47.5          16.8               199
## 313 Chinstrap     Dream           47.6          18.3               195
## 314 Chinstrap     Dream           52.0          20.7               210
## 315 Chinstrap     Dream           46.9          16.6               192
## 316 Chinstrap     Dream           53.5          19.9               205
## 317 Chinstrap     Dream           49.0          19.5               210
## 318 Chinstrap     Dream           46.2          17.5               187
## 319 Chinstrap     Dream           50.9          19.1               196
## 320 Chinstrap     Dream           45.5          17.0               196
## 321 Chinstrap     Dream           50.9          17.9               196
## 322 Chinstrap     Dream           50.8          18.5               201
## 323 Chinstrap     Dream           50.1          17.9               190
## 324 Chinstrap     Dream           49.0          19.6               212
## 325 Chinstrap     Dream           51.5          18.7               187
## 326 Chinstrap     Dream           49.8          17.3               198
## 327 Chinstrap     Dream           48.1          16.4               199
## 328 Chinstrap     Dream           51.4          19.0               201
## 329 Chinstrap     Dream           45.7          17.3               193
## 330 Chinstrap     Dream           50.7          19.7               203
## 331 Chinstrap     Dream           42.5          17.3               187
## 332 Chinstrap     Dream           52.2          18.8               197
## 333 Chinstrap     Dream           45.2          16.6               191
## 334 Chinstrap     Dream           49.3          19.9               203
## 335 Chinstrap     Dream           50.2          18.8               202
## 336 Chinstrap     Dream           45.6          19.4               194
## 337 Chinstrap     Dream           51.9          19.5               206
## 338 Chinstrap     Dream           46.8          16.5               189
## 339 Chinstrap     Dream           45.7          17.0               195
## 340 Chinstrap     Dream           55.8          19.8               207
## 341 Chinstrap     Dream           43.5          18.1               202
## 342 Chinstrap     Dream           49.6          18.2               193
## 343 Chinstrap     Dream           50.8          19.0               210
## 344 Chinstrap     Dream           50.2          18.7               198
##     body_mass_g    sex year
## 1          3750   male 2007
## 2          3800 female 2007
## 3          3250 female 2007
## 4            NA   <NA> 2007
## 5          3450 female 2007
## 6          3650   male 2007
## 7          3625 female 2007
## 8          4675   male 2007
## 9          3475   <NA> 2007
## 10         4250   <NA> 2007
## 11         3300   <NA> 2007
## 12         3700   <NA> 2007
## 13         3200 female 2007
## 14         3800   male 2007
## 15         4400   male 2007
## 16         3700 female 2007
## 17         3450 female 2007
## 18         4500   male 2007
## 19         3325 female 2007
## 20         4200   male 2007
## 21         3400 female 2007
## 22         3600   male 2007
## 23         3800 female 2007
## 24         3950   male 2007
## 25         3800   male 2007
## 26         3800 female 2007
## 27         3550   male 2007
## 28         3200 female 2007
## 29         3150 female 2007
## 30         3950   male 2007
## 31         3250 female 2007
## 32         3900   male 2007
## 33         3300 female 2007
## 34         3900   male 2007
## 35         3325 female 2007
## 36         4150   male 2007
## 37         3950   male 2007
## 38         3550 female 2007
## 39         3300 female 2007
## 40         4650   male 2007
## 41         3150 female 2007
## 42         3900   male 2007
## 43         3100 female 2007
## 44         4400   male 2007
## 45         3000 female 2007
## 46         4600   male 2007
## 47         3425   male 2007
## 48         2975   <NA> 2007
## 49         3450 female 2007
## 50         4150   male 2007
## 51         3500 female 2008
## 52         4300   male 2008
## 53         3450 female 2008
## 54         4050   male 2008
## 55         2900 female 2008
## 56         3700   male 2008
## 57         3550 female 2008
## 58         3800   male 2008
## 59         2850 female 2008
## 60         3750   male 2008
## 61         3150 female 2008
## 62         4400   male 2008
## 63         3600 female 2008
## 64         4050   male 2008
## 65         2850 female 2008
## 66         3950   male 2008
## 67         3350 female 2008
## 68         4100   male 2008
## 69         3050 female 2008
## 70         4450   male 2008
## 71         3600 female 2008
## 72         3900   male 2008
## 73         3550 female 2008
## 74         4150   male 2008
## 75         3700 female 2008
## 76         4250   male 2008
## 77         3700 female 2008
## 78         3900   male 2008
## 79         3550 female 2008
## 80         4000   male 2008
## 81         3200 female 2008
## 82         4700   male 2008
## 83         3800 female 2008
## 84         4200   male 2008
## 85         3350 female 2008
## 86         3550   male 2008
## 87         3800   male 2008
## 88         3500 female 2008
## 89         3950   male 2008
## 90         3600 female 2008
## 91         3550 female 2008
## 92         4300   male 2008
## 93         3400 female 2008
## 94         4450   male 2008
## 95         3300 female 2008
## 96         4300   male 2008
## 97         3700 female 2008
## 98         4350   male 2008
## 99         2900 female 2008
## 100        4100   male 2008
## 101        3725 female 2009
## 102        4725   male 2009
## 103        3075 female 2009
## 104        4250   male 2009
## 105        2925 female 2009
## 106        3550   male 2009
## 107        3750 female 2009
## 108        3900   male 2009
## 109        3175 female 2009
## 110        4775   male 2009
## 111        3825 female 2009
## 112        4600   male 2009
## 113        3200 female 2009
## 114        4275   male 2009
## 115        3900 female 2009
## 116        4075   male 2009
## 117        2900 female 2009
## 118        3775   male 2009
## 119        3350 female 2009
## 120        3325   male 2009
## 121        3150 female 2009
## 122        3500   male 2009
## 123        3450 female 2009
## 124        3875   male 2009
## 125        3050 female 2009
## 126        4000   male 2009
## 127        3275 female 2009
## 128        4300   male 2009
## 129        3050 female 2009
## 130        4000   male 2009
## 131        3325 female 2009
## 132        3500   male 2009
## 133        3500 female 2009
## 134        4475   male 2009
## 135        3425 female 2009
## 136        3900   male 2009
## 137        3175 female 2009
## 138        3975   male 2009
## 139        3400 female 2009
## 140        4250   male 2009
## 141        3400 female 2009
## 142        3475   male 2009
## 143        3050 female 2009
## 144        3725   male 2009
## 145        3000 female 2009
## 146        3650   male 2009
## 147        4250   male 2009
## 148        3475 female 2009
## 149        3450 female 2009
## 150        3750   male 2009
## 151        3700 female 2009
## 152        4000   male 2009
## 153        4500 female 2007
## 154        5700   male 2007
## 155        4450 female 2007
## 156        5700   male 2007
## 157        5400   male 2007
## 158        4550 female 2007
## 159        4800 female 2007
## 160        5200   male 2007
## 161        4400 female 2007
## 162        5150   male 2007
## 163        4650 female 2007
## 164        5550   male 2007
## 165        4650 female 2007
## 166        5850   male 2007
## 167        4200 female 2007
## 168        5850   male 2007
## 169        4150 female 2007
## 170        6300   male 2007
## 171        4800 female 2007
## 172        5350   male 2007
## 173        5700   male 2007
## 174        5000 female 2007
## 175        4400 female 2007
## 176        5050   male 2007
## 177        5000 female 2007
## 178        5100   male 2007
## 179        4100   <NA> 2007
## 180        5650   male 2007
## 181        4600 female 2007
## 182        5550   male 2007
## 183        5250   male 2007
## 184        4700 female 2007
## 185        5050 female 2007
## 186        6050   male 2007
## 187        5150 female 2008
## 188        5400   male 2008
## 189        4950 female 2008
## 190        5250   male 2008
## 191        4350 female 2008
## 192        5350   male 2008
## 193        3950 female 2008
## 194        5700   male 2008
## 195        4300 female 2008
## 196        4750   male 2008
## 197        5550   male 2008
## 198        4900 female 2008
## 199        4200 female 2008
## 200        5400   male 2008
## 201        5100 female 2008
## 202        5300   male 2008
## 203        4850 female 2008
## 204        5300   male 2008
## 205        4400 female 2008
## 206        5000   male 2008
## 207        4900 female 2008
## 208        5050   male 2008
## 209        4300 female 2008
## 210        5000   male 2008
## 211        4450 female 2008
## 212        5550   male 2008
## 213        4200 female 2008
## 214        5300   male 2008
## 215        4400 female 2008
## 216        5650   male 2008
## 217        4700 female 2008
## 218        5700   male 2008
## 219        4650   <NA> 2008
## 220        5800   male 2008
## 221        4700 female 2008
## 222        5550   male 2008
## 223        4750 female 2008
## 224        5000   male 2008
## 225        5100   male 2008
## 226        5200 female 2008
## 227        4700 female 2008
## 228        5800   male 2008
## 229        4600 female 2008
## 230        6000   male 2008
## 231        4750 female 2008
## 232        5950   male 2008
## 233        4625 female 2009
## 234        5450   male 2009
## 235        4725 female 2009
## 236        5350   male 2009
## 237        4750 female 2009
## 238        5600   male 2009
## 239        4600 female 2009
## 240        5300   male 2009
## 241        4875 female 2009
## 242        5550   male 2009
## 243        4950 female 2009
## 244        5400   male 2009
## 245        4750 female 2009
## 246        5650   male 2009
## 247        4850 female 2009
## 248        5200   male 2009
## 249        4925   male 2009
## 250        4875 female 2009
## 251        4625 female 2009
## 252        5250   male 2009
## 253        4850 female 2009
## 254        5600   male 2009
## 255        4975 female 2009
## 256        5500   male 2009
## 257        4725   <NA> 2009
## 258        5500   male 2009
## 259        4700 female 2009
## 260        5500   male 2009
## 261        4575 female 2009
## 262        5500   male 2009
## 263        5000 female 2009
## 264        5950   male 2009
## 265        4650 female 2009
## 266        5500   male 2009
## 267        4375 female 2009
## 268        5850   male 2009
## 269        4875   <NA> 2009
## 270        6000   male 2009
## 271        4925 female 2009
## 272          NA   <NA> 2009
## 273        4850 female 2009
## 274        5750   male 2009
## 275        5200 female 2009
## 276        5400   male 2009
## 277        3500 female 2007
## 278        3900   male 2007
## 279        3650   male 2007
## 280        3525 female 2007
## 281        3725   male 2007
## 282        3950 female 2007
## 283        3250 female 2007
## 284        3750   male 2007
## 285        4150 female 2007
## 286         3700   male 2007
## 287         3800 female 2007
## 288         3775   male 2007
## 289         3700 female 2007
## 290         4050   male 2007
## 291         3575 female 2007
## 292         4050   male 2007
## 293         3300   male 2007
## 294         3700 female 2007
## 295         3450 female 2007
## 296         4400   male 2007
## 297         3600 female 2007
## 298         3400   male 2007
## 299         2900 female 2007
## 300         3800   male 2007
## 301         3300 female 2007
## 302         4150   male 2007
## 303         3400 female 2008
## 304         3800   male 2008
## 305         3700 female 2008
## 306         4550   male 2008
## 307         3200 female 2008
## 308         4300   male 2008
## 309         3350 female 2008
## 310         4100   male 2008
## 311         3600   male 2008
## 312         3900 female 2008
## 313         3850 female 2008
## 314         4800   male 2008
## 315         2700 female 2008
## 316         4500   male 2008
## 317         3950   male 2008
## 318         3650 female 2008
## 319         3550   male 2008
## 320         3500 female 2008
## 321         3675 female 2009
## 322         4450   male 2009
## 323         3400 female 2009
## 324         4300   male 2009
## 325         3250   male 2009
## 326         3675 female 2009
## 327         3325 female 2009
## 328         3950   male 2009
## 329         3600 female 2009
## 330         4050   male 2009
## 331         3350 female 2009
## 332         3450   male 2009
## 333         3250 female 2009
## 334         4050   male 2009
## 335         3800   male 2009
## 336         3525 female 2009
## 337         3950   male 2009
## 338         3650 female 2009
## 339         3650 female 2009
## 340        4000   male 2009
## 341        3400 female 2009
## 342        3775   male 2009
## 343        4100   male 2009
## 344        3775 female 2009
```

Here, I have forced the knitted file to express the entire dataset, a full 79 pages!

Additionally when tidyverse was libraried into R

- it expressed a bunch of stuff we may not want in our final report!

We do not want to see either of these examples

- in a submitted assignment anymore!

If you need to print a data frame or value make sure that it doesnt take up pages.

We suggest you use

- `dplyr::glimpse()`
- or `dplyr::tibble()`
- which both natively suppress `base::print()` statements to just a few rows.

Notice that **I am declaring the namespace** in my code,

- there a thousands of packages
- and people aren’t that creative
  - so it is often necessary what package’s function you are using!

We can suppress WARNINGS and MESSAGES in our finished report

- by adding it to the beginning of our code blocks like so:

- “‘`{r, warning = FALSE, message = FALSE}`

```
library(tidyverse)
```

```
## Error: package or namespace load failed for 'tidyverse' in loadNamespace(i, c(lib.loc, .libPaths()),
##  namespace 'vctrs' 0.5.1 is already loaded, but >= 0.5.2 is required
```

```
library(palmerpenguins)
df <- palmerpenguins::penguins

df %>%
  dplyr::group_by(year) %>%
  dplyr::tally()
```

```
## Error in df %>% dplyr::group_by(year) %>% dplyr::tally(): could not find function "%>%"
```

```
##Print Statements should go at the end of a code block
dplyr::tibble(df)
```

```
## # A tibble: 344 x 8
##    species island    bill_length_mm bill_depth_mm flipper_~1 body_~2 sex    year
##    <fct>   <fct>              <dbl>         <dbl>      <int>   <int> <fct> <int>
##  1 Adelie  Torgersen           39.1          18.7        181    3750 male   2007
##  2 Adelie  Torgersen           39.5          17.4        186    3800 fema~  2007
##  3 Adelie  Torgersen           40.3          18          195    3250 fema~  2007
##  4 Adelie  Torgersen           NA            NA           NA      NA <NA>   2007
##  5 Adelie  Torgersen           36.7          19.3        193    3450 fema~  2007
##  6 Adelie  Torgersen           39.3          20.6        190    3650 male   2007
##  7 Adelie  Torgersen           38.9          17.8        181    3625 fema~  2007
##  8 Adelie  Torgersen           39.2          19.6        195    4675 male   2007
##  9 Adelie  Torgersen           34.1          18.1        193    3475 <NA>   2007
## 10 Adelie  Torgersen           42            20.2        190    4250 <NA>   2007
## # ... with 334 more rows, and abbreviated variable names 1: flipper_length_mm,
## #   2: body_mass_g
```

```
dplyr::glimpse(df)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

3.2.3.1.9 Comments

- In order for us to better understand our own work and to better to be able to share knowledge is it important to comment your code!

Even with the most strigent of commenting,

- I myself have returned to scripts written with confusion and fustration.

Therefore, from now on **YOU MUST BE ACTIVELY COMMENTING YOUR CODE**.

We will also be checking your adherence to the 80 character line limit.

A comment is not helpful if it gets cut off halfway through explaining your work.

NOTE: DO NOT WRITE THE TEXT-BASED ANSWER AS A COMMENT!

3.2.3.1.10 Questions and Answering Style

- One word answers will now not receive full credit.

We want you to fully explain your reasoning

- and how the data/analysis leads you to these conclusions.

This will enable us to give partial credit where possible.

Make sure to show the relative code outputs to the questions asked.

Ask yourself if your report can be read by anyone!

**NOTE: DO NOT WRITE A PARAGRAPH FOR AN ANSWER.*

It would be easier if you have a text answer

- right below the question/sub-question.

If you decide to answer everything in one place,

- make sure you are referring to the correct sub-question.

Please double check before you submit.

NOTE

The code blocks and comments in the assigment R Markdown files

- are merely a suggestion, or guidance, of what you can do.

The comments are there to guide your process.

However, there are many approaches to problems

- as mentioned before so your method
- may not always match and that's okay!

3.2.3.1.11 Plots

- Plots now will also have stricter requirements.

You will no longer receive full credit for the base output

- of the ggplot2:ggplot() function.
- or for using the base `plot` command

Experiment with different aspects of the ggplot2 package

- and make your plots more colorful and easy to follow.

The expectations of a good plot include:

- adding a plot title,
- axes labels (with units, if any),
- changing theme specifications,
- adding a meaningful axes scale
- and using good color schemes with appropriate legends.

An example of a professional plot is included below.

NOTE

We do not mean to make you think

- that the more complicated your plot is,
- the better.

Visuals should be purposeful.

If your visuals are too complicated

- it can become distracting.

With that being said,

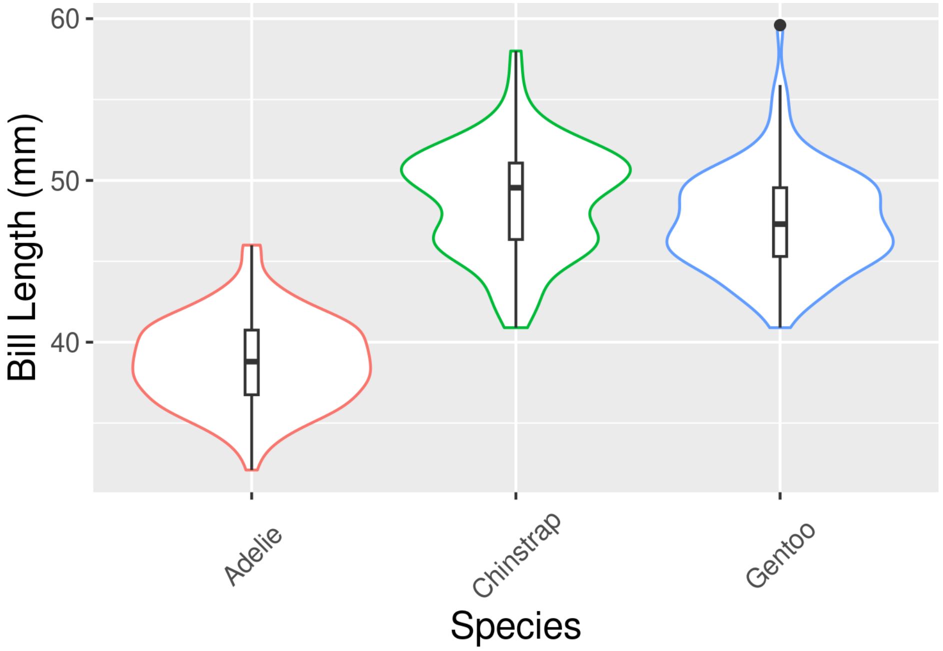
- you are highly encouraged to reuse past work for future submissions.

Infact, you can start collecting useful code snippets in

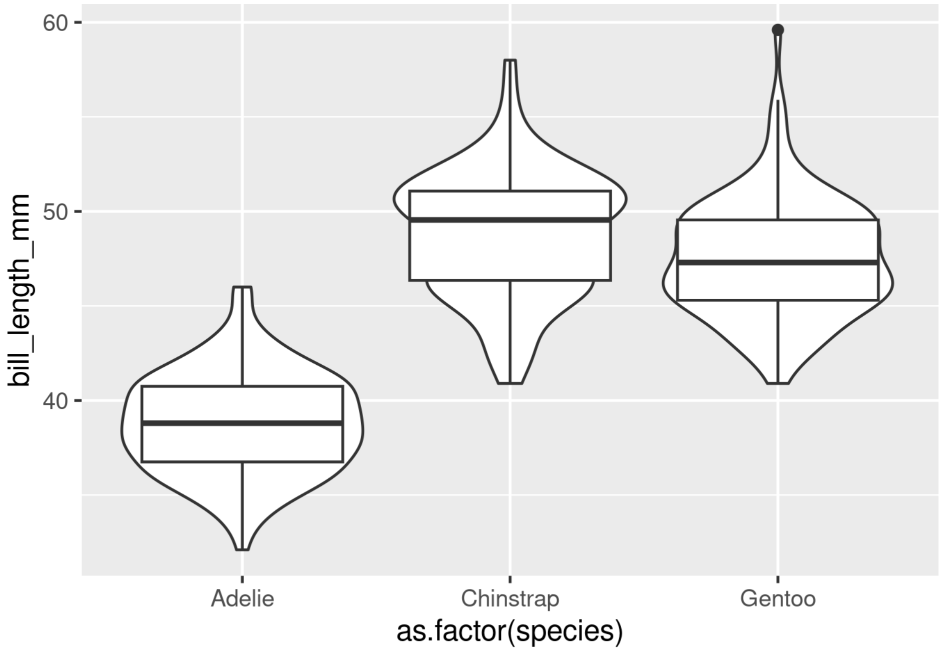
- either, .R scripts or better yet, R Markdown files.

```
library(ggplot2)

df <- palmerpenguins::penguins

mytheme <- theme(
  text = element_text(size = rel(3.5)),
  strip.text.y = element_text(size = rel(3.5)),
  strip.text.x = element_text(size = rel(3.5))
)
#sets a relative size for all labels
#Strip text is used when plots are faceted
```

```
mytheme2 <-
  theme(
    axis.text.x = element_text(
      size = 10,
      angle = 45,

      vjust = 0.5
    ),
    axis.text.y = element_text(size = 10),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14)
  )
#sets individual sizes
#Angle determines the angle of the text


ggplot(df) + #calls df
  geom_violin(aes(
    x = as.factor(species),
    y = bill_length_mm,
    color = as.factor(species)
  )) +
  #creates vioplot
  geom_boxplot(aes(x = as.factor(species),
                   y = bill_length_mm),
               width = .05) +
  #Notice I did not add a color here
  #I think it looks bad
  #creates boxplot narrow
  labs(x = 'Species',
       y = 'Bill Length (mm)') + #x & y labs
  mytheme2 +
  #12pt 45degreex .5vjustx
  theme(legend.position = 'none')
```



```
## Versus
##
##
ggplot(df) +
  geom_violin(aes(x = as.factor(species),
                  y = bill_length_mm)) +
  geom_boxplot(aes(x = as.factor(species),
                   y = bill_length_mm))
```



3.2.3.1.12 Grading

- With this information in mind,

    - We would go through each and every code block
    - and read through your comments as well as your answers.

Your score is reflective of your performance on the lab exercise.

Please be mindful that the grading is done on a curve,

- given the diversity of the students in the course.

That way, an A does not mean 90+ points on the course!

3.2.3.1.13 Shortcuts in RStudio

- To view the document outline and what all questions to answer: Ctrl + Shift + O

To reformat code: Ctrl + Shift + A

Please make sure you use these shortcuts each time you work on the code.

3.2.3.1.14 Final Thoughts

- The best way to learn this course is not only by learning on your own
    - but also by talking to your peers
    - and communicating with the professor as well as the TA.

While discussing ideas with your peers is highly encouraged,

- plagiarism will not be tolerated.

We want you to make the best use of this course

- by working sincerely
- and being open to learning new concepts.

As data modeling concepts can be applied to any industry you can think of,

- please make sure you are paying attention to the coursework
- and learning as much as you can.

Please contact us if you face any difficulty in the course

- and we'd be more than happy to help you out
- (this encompasses a range of problems from course difficulty to RStudio issues).

We will be on Slack most of the times

- and have regular office hours.

Happy learning and hope you all do well!

3.2.3.2 Clean R Code Is Critical

- Over many years of experience delivering successful projects,
    - There is one common element across all these projects

A clean, readable, and concise code base

- is the key to effective collaboration
- and provides the highest quality value to the client.

3.2.3.2.1 Code Review

- Code review is a crucial part
  - of maintaining a high-quality code process.
  - It is also a great way to
    - share best practices
    - and distribute knowledge among team members.
  - Code review as a must for every project.
    - Lets review best practices
    - recommended for all data science teams.

Having a well-established code review process does not change the fact

- that the data scientist is responsible for
  - writing good, clean code!
- Pointing out all of the code's basic mistakes
  - is painful, time-consuming,
- And distracts reviewers from going deep
  - into code logic
  - or improving the code's effectiveness.

Poorly written code can also harm team morale

- code reviewers are frustrated
  - while code creators might feel offended by a huge number of comments.

That is why before sending the code to review,

- developers need to make sure that the code is as "clean" as possible.

Also, note that there is not always a code reviewer that can come to the rescue.

- Sometimes you are on your own in a project.
- Even though you think the code is ok for you now,
  - consider rereading it in a few months
  - you want it to be clear to avoid wasting your own time later on.

Lets summarize

- the most common mistakes to avoid
- and outline best practices to follow
  - in programming in general.
- Follow these tips to speed up the code review iteration process
  - and be a better data scientist

3.2.3.2.2 Avoid Comments with Comments

- Adding comments to the code is a crucial developer skill.
  - However, a more critical and harder to master skill
    - is knowing when not to add comments.
  - Writing good comments is more of an art than a science.
  - It requires a lot of experience,
    - and you can write entire book chapters about it
    - (e.g., Clean Code: A Handbook of Agile Software Craftsmanship.

There are few simple rules that you should follow,

- to avoid comments about your comments:

The comments should add external knowledge to the reader:

- if they're explaining what is happening in the code itself,
  - it is a red flag that the code is not clean
  - and needs to be refactored.

3.2.3.2.3 Code Refactoring

- What is **code refactoring**
  - code refactoring is the process of restructuring existing computer code
    - changing the factoring
    - without changing its external behavior.
  - Refactoring is intended to improve
    - the design,
    - structure,
    - and/or implementation of the software
    - (its non-functional attributes),
  - while preserving its functionality.

Potential advantages of refactoring may include

- improved code readability
  - and reduced complexity;
- these can improve the source code's
  - maintainability
- and create a
  - simpler,
  - cleaner,
  - or more expressive internal architecture
  - or object model to improve extensibility.
- Another potential goal for refactoring is improved performance;
  - software engineers face an ongoing challenge
  - to write programs that perform faster
  - or use less memory.

3.2.3.2.4 Comments

- So in your code comments, if some hack was used,
  - then comments might be used to explain what is going on.
  - Comment required business logic
    - or exceptions added on purpose.
  - Try to think of what can be surprising to the future reader
    - and preempt their confusion.
  - Write only crucial comments!
    - Your comments should not be a dictionary of easily searchable information.

In general, comments are distracting

- and do not explain logic as well as the code does.

For example, I recently saw a comment like this in the code:

- `trimws(.) # this function trims leading/trailing white spaces`
  - This comment is is redundant.
- If the reader does not know what function trimws is doing,
  - it can be easily checked.

- A more robust comment here can be helpful,
  - e.g.: `trimws(.) # TODO(Marcin Dubel): Trimming white spaces is crucial here due to database entries inconsistency; data needs to be cleaned.`

3.2.3.2.5 Use `roxygen2` inline documentation

- When writing functions in R, I recommend using {`roxygen2`} comments
  - even if you are not writing a package.

```
library(roxygen2)
?roxygen2
```

`roxygen2` is a package used for building R packages

- Generate your
  - Rd documentation,
  - 'NAMESPACE' file,
  - and collation field
- using specially formatted comments.
- Writing documentation in-line with code
  - makes it easier to keep your documentation up-to-date
  - as your requirements change.
- 'Roxygen2' is inspired by the 'Doxygen' system for C++.
- Python3 has `[sphinx](https://en.wikipedia.org/wiki/Sphinx_(documentation_generator))`

`roxygen2` is an excellent tool for organizing the knowledge about

- the function goal,
  - parameters,
  - and output.

3.2.3.2.6 More on commenting

- Only write comments (as well as all parts of code) in English.
  - Making it understandable to all readers
    - might save you encoding issues that can appear
    - if you use special characters from your native language.

In case some code needs to be refactored/modified in the future,

- mark it with the # TODO comment.

Also, add some information

- to identify you as the author of this comment
  - (to contact in case details are needed)
- and a brief explanation of
  - why the following code is marked as TODO
  - and not modified right away.

Never leave commented-out code un-commented!

- It is ok to keep some parts for the future
  - or turn them off for a while,
  - but always mark the reason for this action.

Remember that the comments will stay in the code.

- If there is something that you would like to tell your reviewer,
  - but only once,

- add a comment to the Pull (Merge) Request
    - and not to the code itself.

Example: I recently saw removing part of the code with a comment like:

- "Removed as the logic changed."
- Ok, good to know,
    - but later that comment in the code looks odd and is redundant,
    - as the reader no longer sees the removed code.

3.2.3.2.7 Strings

- A common problem related to texts
    - is the readability of string concatenations.
    - What one encounters a lot
        * is an overuse of the paste function.
    - Don't get me wrong;
        * it is a great function when your string is simple,
        * e.g. `paste("My name is", my_name)`,
    - but for more complicated forms, it is hard to read:

```
paste("My name is", my_name, "and I live in", my_city, "developing in",
      language, "for over", years_of_coding)
```

A better solution is to use

- `sprintf` functions
- or `glue`, e.g.

```
glue("My name is {my_name} and I live in {my_city} developing in {language}
     for over {years_of_coding}")
```

Isn't it clearer

- without all those commas
- and quotation marks?

When dealing with many code blocks,

- it would be great to extract them to separate locations,
    - e.g., to a .yml file.
- It makes both code and text blocks
    - easier to read and maintain.

The last tip related to texts:

- one of the debugging techniques,
    - often used in Shiny applications,
    - is adding `print()` statements.
- Double-check whether the prints are not left in the code
    - this can be quite embarrassing during code review!

3.2.3.2.8 Loops

- Loops are
    - one of the programming building blocks
    - and are a very powerful tool.

Nevertheless, they can be computationally heavy

- and thus need to be used carefully.

The rule of thumb that you should follow is:

- always double-check if looping is a good option.

It is hardly a case that

- you need to loop over rows in data.frame:
  - there should be a {`dplyr`} function
  - to deal with the problem more efficiently.

Another common source of issues is

- looping over elements
  - using the length of the object,
  - e.g. `for(i in 1:length(x)) ....` But what if the length of x is zero!
  - Yes, the loop will go another way
  - for iterator values 1, 0.
- That is probably not your plan.
  - Using `seq_along` or `seq_len` functions
  - are much safer.

Also, remember about the `apply` family of functions for looping.

- They are great
  - (not to mention {the `purrr` package} solutions)!
- Note that using `sapply`
  - might be commented by the reviewer as not stable
  - because this function chooses the type of the output itself!
- So sometimes it will be
  - a list,
  - sometimes a vector.
- Using vapply is safer,
  - as the programmer defines the expected output class.

3.2.3.2.9 Code Sharing

- Even if you are working alone,
  - you probably would like your program
    - to run correctly on other machines.
  - And how crucial it is
    - when you are sharing the code with the team!
  - To achieve this,
    - **never use absolute paths in your code**,
    - e.g. `/home/marcin/my_files/old_projects/september/project_name/file.txt`.
  - It won't be accessible for others.
    - Note that any violation of folder structure will crash the code.

As you should already **have an Rproject for all coding work**,

- you need to use paths related to the particular Rproject
  - in this case; it will be `./file.txt`.
- What is more, **one would suggest keeping all the paths**
  - as variables in a single place
- so that renaming a file requires one change in code,
  - not, e.g., twenty in six different files.

Sometimes your software needs to use some credentials or tokens,

- e.g., to a database or private repositories.

    - or an external API, like Google Maps
- You should never commit such secrets to the git repository!
    - Even if the entries are the same among the team.
- Usually, the good practice is to keep such values
    - in .Renviron file as environmental variables
    - that are loaded on start
    - and the file itself is ignored in the repo.
    - You can read more about .Renviron here.
- Or use the `keyring` package
    - It stores tokens or credentials
    - And exists for both R and Python3

3.2.3.2.10 Good Programming Practices

- Finally, let's focus on how you can improve your code.

    - First of all,
        - your code should be easily understandable and clean
    - even if you are working alone,
        - when you come back to code after a while,
        - it will make your life easier!

Use specific variable names,

- even if they seem to be lengthy
- the rule of thumb is that you should be able to guess
    - what is inside just by reading the name,
    - so `table_cases_per_country` is ok,
    - but `tbl1` is not.
- Avoid abbreviations.
    - Lengthy is preferable to vague.
- Keep consistent style for object names
    - (like camelCase or snake_case)
    - as agreed among your team members.

Do NOT abbreviate logical values

- such as `T` for `TRUE`
    - and `F` for `FALSE`
- the code will work,
    - but `T` and `F` are regular objects
    - that can be overwritten
- while `TRUE` and `FALSE` are special values
    - as defined in R.

Do not compare logical values using equations,

- like `if(my_logical == TRUE)`.
- If you can compare to `TRUE`,
    - it means your value is already logical,
    - so `if(my_logical)` is enough!
- If you want to double-check
    - that the value is `TRUE` indeed
    - (and not, e.g., NA),
    - you can use the isTRUE() function.

Make sure that your logic statements are correct.

- Check if you understand the difference in R

    - between single and double logical operators!

Good spacing is crucial for readability.

- Make sure that the rules are the same
    - and agreed upon in the team.
- It will make it easier to follow each other's code.
- The simplest solution is to stand on the shoulders of giants
    - and follow the tidyverse style guide.
    - Its the same as the Google R style guide.

However, checking the style in every line

- during the review is quite inefficient,
- so make sure to introduce `lintr` and `styler`
    - in your development workflow
- Our use the code diagnostics in Rstudio
- This can be lifesaving!

Recently we found an error in some legacy code

- that would have been automatically recognized by `lintr`:

```
sum_of_values <- first_element
  + second_element
```

This does not return the sum of the elements

- as the author was expecting.

Speaking of variable names

- this is known to be one of the hardest things in programming.
- Thus avoid it when it is unnecessary.

Note that R functions return, by default,

- the last created element,
- so you can easily replace that:

```
sum_elements <- function(first, second) {
  my_redundant_variable_name <- sum(first, second)
  return(my_redundant_variable_name)
}
```

With something shorter

- (and simpler,
    - you don't need to think about names):

```
sum_elements <- function(first, second) {
  sum(first, second)
}
```

On the other hand, please DO use additional variables

- anytime you repeat some function call or calculation!
- It will make it computationally more effective
    - and easier to be modified in the future.

Remember to keep your code DRY

- don't repeat yourself.
- If you copy-paste some code,

- think twice whether it
    - shouldn't be saved to a variable,
    - done in a loop,
    - or moved to a function.

3.2.3.2.11 Conclusion

- And there you have it
    - five strategies to write clean R code
        - and leave your code reviewer commentless.
    - These five alone will ensure you're writing great-quality code
        - that is easy to understand,
        - even years down the road.

3.2.3.3 Links

- Marcel Dubel, Clean Code
- Clean Code: A Handbook of Agile Software Craftsmanship