Midterm 1

Dan Aurell

2023-03-07

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.1 ✔ purrr 1.0.1  
## ✔ tibble 3.1.8 ✔ dplyr 1.1.0  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0  
## ✔ readr 2.1.4 ✔ forcats 1.0.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

## Question 1.

You are looking at a collaborator’s R code on github, and download the repository, and start exploring the code. The first line of the script is

setwd(“C:/Users/…”)

* What is the author of this code trying to do with the function setwd()?
* Please discuss what is wrong with this approach in terms of reproducibility.
* Where is the working directory of an R project?
* Explain the concept of relative file paths. Is the author of this code using relative file paths?

### Answer to Q1

Your collaborator is setting the working directory in terms of an absolute file path. Note that to set an absolute file path you starts all the way down at C:/ the root directory of the Windows hard drive. This would be reproducible on their machine, but not when I copy the R script to a directory on my own machine, since the absolute file path is not the same. An R Project solves this by letting you set relative file paths. When you open a script through an R Project, the working directory sets to the directory that the R Project is located in. You can now specify relative file paths - file paths relative to the working directory. For example, you can have a data folder containing a csv within the working directory and then just refer to read.csv(“data/file.csv”). The original author may be setting the working directory (using an absolute file path) in order to then use relative file paths, but we don’t know this from the prompt.

## Question 2.

What does the acronym FAIR stand for in the context of this class? Explain how R, GitHub, and other lecture concepts introduced in this course specifically help complete FAIR data principles.

### Answer to Q2

This stands for Findable, Accessible, Interpretable, Reusable

Doing analyses and graphing in R makes sure that the workflow is documented (Reusable) - when compared to a point-and-click interface. If the code is well commented, this also can explain which sections of code generated individual findings (Interpretable). GitHub lets you host code and data online (Findable, Accessible). If you add a DOI to your GitHub repository, this lets you link to the code from a publication of your own and even lets other people cite your data and code directly (Findable, Accessible). Using formats such as Markdown allows people to view your code and outputs in context without even having to download your code (Accessible, Interpretable). GitHub is a backup for your code, which is valuable. Finally, if people want to access your whole repository, GitHub lets them pull a version of the repository onto their own computers so they can recreate your work (Reusable).

## Question 3.

Explain the concept of R packages. What are R packages? Who writes R packages? What is the difference between installing and loading a package? Explain two ways to install and load packages into R.

### Answer to Q3

Packages are collections of R functions that someone else has written, that are available to download. Anyone can write an R package, but depending on where they are hosted they probably have to comply with some standards. For example, R packages are supposed to include help files. They may also include datasets and vignettes to show you how to use the package. You usually download R packages from CRAN, a network of servers that hosts R packages. When you use install.packages() this downloads them to your computer. When you use library(), this loads the packages into memory, making all the functions available to use during an R session. You can alternatively install a package in the “Packages” pane of RStudio by pressing “Install”; once a package is installed you can load a package by pointing and clicking to check the checkbox for that package.

## Question 4.

Explain the following concepts of ggplot and give examples of each concept using code and figures generated with ggplot using the data of your choosing.

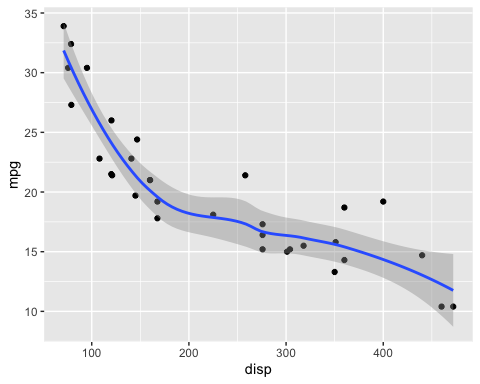
* Layering
* Scales
* Themes
* Facets

### Answer to Q4

Layering is the idea that a plot is built up from individual layers that either render something on top of the previous layers - or modifies other layers. For example, adding geom\_point() to a boxplot adds a new scatterplot layer on top of the boxplot,

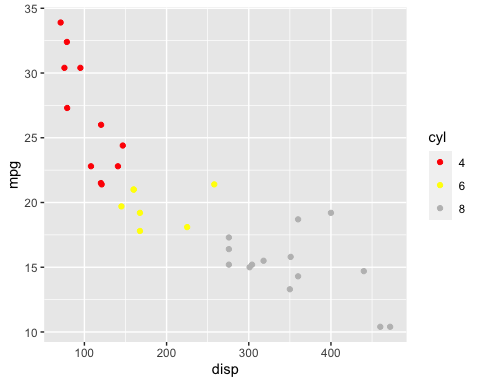
df <- mtcars  
df$cyl <- as.factor(df$cyl)  
  
df %>%   
 ggplot(aes(x = disp, y = mpg)) +  
 geom\_point() +  
 geom\_smooth() # Added layer

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



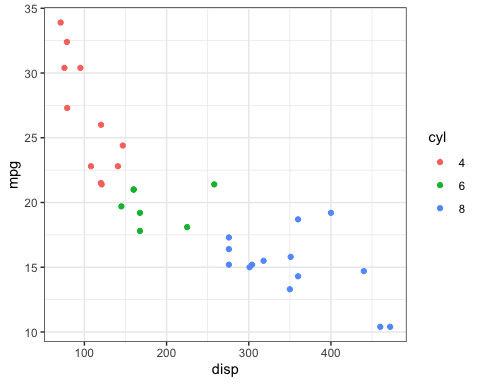
Once you specify that the color aesthetic is mapped to (say) the number of cylinders in an engine, you can use scales to specify HOW that mapping occurs. For example, you can set the levels of colour to correspond to the levels of a categorical value as below, or you can use scale\_continuous\_manual() to set endpoints of colour to line up with specific values of a continuous variable.

df %>%   
 ggplot(aes(x = disp, y = mpg, color = cyl)) +  
 geom\_point() +  
 scale\_discrete\_manual(values = c("red", "yellow", "grey"), aesthetics = "color") # Manually added a bad choice of color



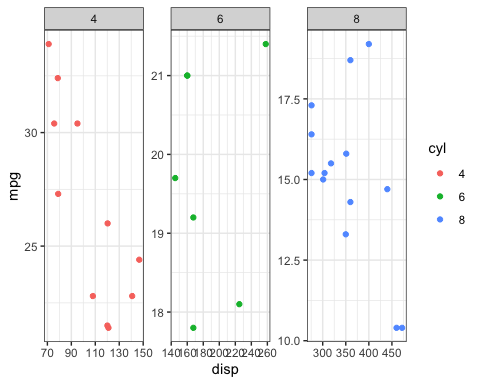
Themes modify aspects of the plotting in general. For example, adding theme\_classic() modifies the axes and plot background to be more minimalistic. Within theme() you can also set other global aspects of the plot, like font size.

df %>%   
 ggplot(aes(x = disp, y = mpg, color = cyl)) +  
 geom\_point() +  
 theme\_bw()



Facetting generates several similar graphs - just that each graph is a subset of the data. For example, if you facet by cylinder count using facet\_wrap(~cyl) ggplot makes a separate subplot for each number of cylinders

df %>%   
 ggplot(aes(x = disp, y = mpg, color = cyl)) +  
 geom\_point() +  
 theme\_bw() +  
 facet\_wrap(~cyl, scales = "free")



## Question 5.

Explain the differences and similarities between a vector, matrix, and dataframe. Demonstrate you know how to subset a dataframe in two ways using the built in dataset ‘ToothGrowth’ with the prompts below:

* Subset ToothGrowth to include rows such that supp is equal to VC
* Subset ToothGrowth to include rows such that supp is equal to VC and dose is equal to 0.5
* Subset ToothGrowth to include the values of len such that supp is equal to VC and dose is equal to 0.5

### Answer to Q5

A vector is a one-dimensional set of items. Matrices and dataframes are two-dimensional sets of items (i.e., they have both rows and columns). All the items in a matrix have to be of the same class (e.g., numeric), while a dataframe can have different classes.

tooth <- ToothGrowth  
  
# Classic subsetting  
tooth[tooth$supp=="VC",]

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5  
## 11 16.5 VC 1.0  
## 12 16.5 VC 1.0  
## 13 15.2 VC 1.0  
## 14 17.3 VC 1.0  
## 15 22.5 VC 1.0  
## 16 17.3 VC 1.0  
## 17 13.6 VC 1.0  
## 18 14.5 VC 1.0  
## 19 18.8 VC 1.0  
## 20 15.5 VC 1.0  
## 21 23.6 VC 2.0  
## 22 18.5 VC 2.0  
## 23 33.9 VC 2.0  
## 24 25.5 VC 2.0  
## 25 26.4 VC 2.0  
## 26 32.5 VC 2.0  
## 27 26.7 VC 2.0  
## 28 21.5 VC 2.0  
## 29 23.3 VC 2.0  
## 30 29.5 VC 2.0

tooth[tooth$supp=="VC" & tooth$dose==0.5,]

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5

tooth[tooth$supp=="VC" & tooth$dose==0.5,1]

## [1] 4.2 11.5 7.3 5.8 6.4 10.0 11.2 11.2 5.2 7.0

# Subsetting using dplyr verbs  
tooth %>%   
 filter(supp=="VC")

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5  
## 11 16.5 VC 1.0  
## 12 16.5 VC 1.0  
## 13 15.2 VC 1.0  
## 14 17.3 VC 1.0  
## 15 22.5 VC 1.0  
## 16 17.3 VC 1.0  
## 17 13.6 VC 1.0  
## 18 14.5 VC 1.0  
## 19 18.8 VC 1.0  
## 20 15.5 VC 1.0  
## 21 23.6 VC 2.0  
## 22 18.5 VC 2.0  
## 23 33.9 VC 2.0  
## 24 25.5 VC 2.0  
## 25 26.4 VC 2.0  
## 26 32.5 VC 2.0  
## 27 26.7 VC 2.0  
## 28 21.5 VC 2.0  
## 29 23.3 VC 2.0  
## 30 29.5 VC 2.0

tooth %>%   
 filter(supp=="VC", dose==0.5)

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5  
## 7 11.2 VC 0.5  
## 8 11.2 VC 0.5  
## 9 5.2 VC 0.5  
## 10 7.0 VC 0.5

tooth %>%   
 filter(supp=="VC", dose==0.5) %>%   
 select(len)

## len  
## 1 4.2  
## 2 11.5  
## 3 7.3  
## 4 5.8  
## 5 6.4  
## 6 10.0  
## 7 11.2  
## 8 11.2  
## 9 5.2  
## 10 7.0

## Question 6.

Create an R markdown version of your answer to question 4 and 5. Save the .Rmd file to your computer and render it as a word document (.docx), .html, and a .md file. Push these files to your github and paste your github url here.

### Answer to Q6

I have made this whole assignment in a version-controlled repository. These files are up my Github: <https://github.com/DanAurell/ENTM6820/tree/main/Midterm_1>

## Question 7.

What is the correct order of events to get your code on github through R studio? Explain each step from creation of a repository to pushing.

### Answer to Q7

Assuming you’ve already linked your RStudio to your GitHub account (a bit complicated)…

1. Create a repository (“repo”) on the GitHub website.
2. Copy the web address of the GitHub repo (with .git at the end) by pressing the green “Code” button on the GitHub repo page
3. In RStudio, make a New Project (select “Version Control” option) which checks out the project from a version control repository

* This will “clone” the repository which means that the contents of the GitHub repo (e.g., the Readme) will be downloaded into your local folder
* In addition, this creates an R Project in your local folder

1. Make some changes locally: For example, make a new R Markdown document
2. Then in the Git pane of RStudio, check the boxes by the file names whose changes you want to push
3. You press “commit” to somehow stage/package these changes and type in a description of the changes when prompted
4. Then you press “push” to update the GitHub repository to reflect your local changes

## Question 8.

After you have worked on a project for a while, you mistakenly delete a file on your github, while it still exists in your local repository (on your computer). Now when you try to push your code to github the push is rejected and gives the following error, “Updates were rejected because the remote contains work that you do not have locally.” How do you solve this error?

### Answer to Q8

First, you pull from GitHub so that your local folder contains all the files on GitHub. Then you should be able to push to GitHub.

## Question 9.

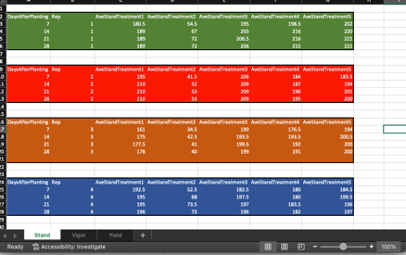
Explain the purpose of a Data Management Plan.

### Answer to Q9

The purpose of a data management plan is to codify the process of generating, managing, using, and archiving data during a research project. This lets you think through the automated, but most importantly, the manual processes that are potentially prone to errors or concerns about the integrity of the data. Some risks include editing data and no longer having access to something you know is the original dataset; losing data through computer failure; losing access to the data through password issues or expiry of institution credentials; etc. By writing a plan you can decide how to mitigate these risks.

## Question 10.

A colleague gives you data in an .xlsx file that looks like this:



Screenshot

Please discuss at least five things wrong with how these data are formatted that make it not reproducibility friendly. Then describe/show your colleague how the data should be formatted.

### Answer to Q10

1. “Stand”, “Vigor”, and “Yield” are variables that should be in separate columns in a single worksheet
2. “Treatment” is a variable and should be a column header, with values of 1-5
3. “Rep” is a variable and should have its own column with values of 1-5 and the reps should not be separated with whitespace and new column headers between.
4. Color is unnecessary
5. You could include the raw data for “Stand” rather than the per (rep-and-treatment) average. This implies that there are also several measurements for each rep and treatment

Here is a simulated dataset that would follow [tidy data](https://r4ds.had.co.nz/tidy-data.html) principles:

data.frame(  
 Rep = c(rep(1, 60), rep(2, 60), rep(3, 60), rep(4, 60), rep(5, 60)),  
 Treatment = rep(c(rep(1, 12),rep(2, 12),rep(3, 12),rep(4, 12),rep(5, 12)), 5),  
 DaysAfterPlanting = rep(c(rep(7, 3),rep(14, 3),rep(21, 3),rep(28, 3)), 25),  
 TechRep = rep(c(1:3), 100),  
 Stand = rnorm(300, 200, 20),  
 Vigor = runif(300, 15, 100),  
 Yield = rnorm(300, 150, 15)  
)

## Rep Treatment DaysAfterPlanting TechRep Stand Vigor Yield  
## 1 1 1 7 1 185.6962 98.58005 135.8955  
## 2 1 1 7 2 209.4054 15.78010 157.7491  
## 3 1 1 7 3 190.0101 93.97732 142.9039  
## 4 1 1 14 1 195.9130 71.16985 173.1948  
## 5 1 1 14 2 187.2662 54.93376 162.4746  
## 6 1 1 14 3 228.0253 40.76277 160.6802  
## 7 1 1 21 1 214.3364 81.91885 158.6943  
## 8 1 1 21 2 157.2063 37.32619 139.5122  
## 9 1 1 21 3 204.3795 24.02125 138.0691  
## 10 1 1 28 1 217.3951 42.09239 152.3092  
## 11 1 1 28 2 205.0586 74.49852 131.3922  
## 12 1 1 28 3 206.4911 78.12627 149.6828  
## 13 1 2 7 1 194.1568 80.61647 155.0537  
## 14 1 2 7 2 217.6844 42.26841 163.1285  
## 15 1 2 7 3 217.7398 48.29060 155.3618  
## 16 1 2 14 1 256.0435 56.94379 154.5311  
## 17 1 2 14 2 246.4104 77.88924 120.1346  
## 18 1 2 14 3 191.4333 79.00342 113.6133  
## 19 1 2 21 1 166.5401 89.80861 146.2153  
## 20 1 2 21 2 237.7112 77.13387 169.6535  
## 21 1 2 21 3 176.2187 83.50020 140.0560  
## 22 1 2 28 1 220.6722 30.24946 171.3819  
## 23 1 2 28 2 188.9679 80.42332 180.0680  
## 24 1 2 28 3 203.9273 94.04009 188.3048  
## 25 1 3 7 1 182.7871 38.90312 141.4740  
## 26 1 3 7 2 180.5687 60.57963 182.1323  
## 27 1 3 7 3 185.4054 75.93392 149.2546  
## 28 1 3 14 1 185.8146 51.92897 162.9564  
## 29 1 3 14 2 206.2578 59.43668 162.4253  
## 30 1 3 14 3 178.1732 86.78027 166.4120  
## 31 1 3 21 1 199.6612 68.84226 174.8891  
## 32 1 3 21 2 221.8243 36.95422 165.4847  
## 33 1 3 21 3 176.4615 58.29764 145.8131  
## 34 1 3 28 1 169.1250 93.67193 140.5992  
## 35 1 3 28 2 199.0602 24.81069 152.4689  
## 36 1 3 28 3 198.4459 38.66856 162.5872  
## 37 1 4 7 1 242.3935 27.61867 128.0374  
## 38 1 4 7 2 143.2966 75.88351 159.2352  
## 39 1 4 7 3 178.8973 91.08485 162.3521  
## 40 1 4 14 1 219.9015 22.74502 122.2628  
## 41 1 4 14 2 178.4884 56.20131 151.1984  
## 42 1 4 14 3 180.6622 41.99424 154.5612  
## 43 1 4 21 1 215.6772 97.91638 126.6015  
## 44 1 4 21 2 204.9932 73.56109 167.8142  
## 45 1 4 21 3 125.3885 75.71929 117.4039  
## 46 1 4 28 1 187.7314 58.43941 155.0504  
## 47 1 4 28 2 188.8513 37.79708 167.5929  
## 48 1 4 28 3 216.0575 39.08768 154.8994  
## 49 1 5 7 1 194.1951 23.16171 134.2408  
## 50 1 5 7 2 212.9137 17.97805 155.0352  
## 51 1 5 7 3 211.4921 56.66450 158.9236  
## 52 1 5 14 1 191.1224 21.07739 147.8440  
## 53 1 5 14 2 190.4620 32.15721 125.6742  
## 54 1 5 14 3 199.9839 67.86635 154.5522  
## 55 1 5 21 1 194.1336 62.39370 145.3907  
## 56 1 5 21 2 198.9576 77.90896 163.5220  
## 57 1 5 21 3 242.6024 88.62470 151.3815  
## 58 1 5 28 1 197.8522 62.06147 125.2741  
## 59 1 5 28 2 226.4209 63.88563 182.7694  
## 60 1 5 28 3 186.4992 99.81502 143.3597  
## 61 2 1 7 1 210.6348 17.86245 169.3807  
## 62 2 1 7 2 195.0197 35.70802 140.7215  
## 63 2 1 7 3 220.1109 49.10727 161.1459  
## 64 2 1 14 1 181.0462 66.24908 151.7312  
## 65 2 1 14 2 199.2559 71.12706 164.8213  
## 66 2 1 14 3 190.6997 35.95111 153.9472  
## 67 2 1 21 1 205.2931 68.89727 155.1822  
## 68 2 1 21 2 225.5933 42.92344 170.2149  
## 69 2 1 21 3 248.3389 31.86377 170.5580  
## 70 2 1 28 1 215.1049 53.42338 143.9521  
## 71 2 1 28 2 191.4456 27.18675 159.1141  
## 72 2 1 28 3 194.3676 96.76697 153.9187  
## 73 2 2 7 1 181.4458 79.17191 152.6482  
## 74 2 2 7 2 180.0815 97.83718 161.5423  
## 75 2 2 7 3 159.3804 69.56778 163.0135  
## 76 2 2 14 1 188.7119 25.93097 156.3166  
## 77 2 2 14 2 198.0955 47.62080 150.0232  
## 78 2 2 14 3 246.0287 74.01931 156.1643  
## 79 2 2 21 1 216.1550 64.18373 142.4416  
## 80 2 2 21 2 212.1922 74.04115 150.0454  
## 81 2 2 21 3 229.6139 78.89225 133.2295  
## 82 2 2 28 1 194.1493 15.36168 165.1025  
## 83 2 2 28 2 165.3693 47.26168 165.9725  
## 84 2 2 28 3 179.2233 94.91543 112.5422  
## 85 2 3 7 1 219.0233 41.42386 127.7929  
## 86 2 3 7 2 173.2197 75.26748 157.6434  
## 87 2 3 7 3 210.0101 37.73943 117.8428  
## 88 2 3 14 1 162.7346 68.77139 146.2272  
## 89 2 3 14 2 182.9788 94.50364 124.9684  
## 90 2 3 14 3 169.7430 46.24518 133.1743  
## 91 2 3 21 1 190.8511 54.90304 143.3417  
## 92 2 3 21 2 190.8373 39.41720 148.2060  
## 93 2 3 21 3 195.4875 16.03695 168.0169  
## 94 2 3 28 1 201.7959 40.79353 172.2061  
## 95 2 3 28 2 199.6574 64.63586 161.9284  
## 96 2 3 28 3 198.6849 78.69078 155.8865  
## 97 2 4 7 1 201.6224 64.88089 164.7965  
## 98 2 4 7 2 179.2871 99.61347 149.2068  
## 99 2 4 7 3 216.8155 17.61131 125.3272  
## 100 2 4 14 1 205.2941 88.41431 159.5467  
## 101 2 4 14 2 218.9039 26.04360 174.8451  
## 102 2 4 14 3 224.7627 60.17190 163.4427  
## 103 2 4 21 1 201.2973 38.46830 137.2059  
## 104 2 4 21 2 206.4075 88.46198 142.6149  
## 105 2 4 21 3 199.2613 73.85897 122.3431  
## 106 2 4 28 1 198.8925 68.86465 142.4549  
## 107 2 4 28 2 191.7556 83.09817 150.8941  
## 108 2 4 28 3 207.9502 54.52656 178.7464  
## 109 2 5 7 1 173.6852 22.95985 159.0974  
## 110 2 5 7 2 176.3579 93.96477 161.3848  
## 111 2 5 7 3 189.4278 70.67755 155.8996  
## 112 2 5 14 1 204.9276 79.93991 133.6701  
## 113 2 5 14 2 235.7093 87.62670 155.9871  
## 114 2 5 14 3 219.9283 47.94678 146.6204  
## 115 2 5 21 1 238.7653 63.52521 135.9730  
## 116 2 5 21 2 174.1337 34.34886 148.8840  
## 117 2 5 21 3 195.4681 97.95302 132.3667  
## 118 2 5 28 1 202.6979 29.92255 149.7321  
## 119 2 5 28 2 203.4913 31.09418 122.3447  
## 120 2 5 28 3 208.3724 34.46171 155.7395  
## 121 3 1 7 1 211.9386 67.86122 133.8630  
## 122 3 1 7 2 228.6189 22.99439 162.8778  
## 123 3 1 7 3 209.5862 99.16503 157.6877  
## 124 3 1 14 1 168.4878 87.05095 131.2081  
## 125 3 1 14 2 175.6489 90.64546 141.0510  
## 126 3 1 14 3 230.1831 34.97667 160.3381  
## 127 3 1 21 1 211.8434 75.97374 160.0833  
## 128 3 1 21 2 225.0822 70.83881 138.0302  
## 129 3 1 21 3 204.8549 85.68696 142.8807  
## 130 3 1 28 1 219.2347 66.89194 134.9511  
## 131 3 1 28 2 183.6156 74.31960 163.6383  
## 132 3 1 28 3 180.7113 88.80578 131.9359  
## 133 3 2 7 1 194.8407 30.15804 171.5243  
## 134 3 2 7 2 209.5532 39.27138 125.3147  
## 135 3 2 7 3 165.0279 97.25256 161.2403  
## 136 3 2 14 1 186.7401 95.00541 134.6322  
## 137 3 2 14 2 208.6213 21.39712 124.7490  
## 138 3 2 14 3 210.0690 22.96652 153.8587  
## 139 3 2 21 1 199.2229 64.47065 151.8384  
## 140 3 2 21 2 201.6632 30.27402 144.1915  
## 141 3 2 21 3 220.0238 88.95044 138.1641  
## 142 3 2 28 1 185.9465 73.71483 139.8185  
## 143 3 2 28 2 217.7148 44.25535 157.3821  
## 144 3 2 28 3 191.5020 45.30102 128.9034  
## 145 3 3 7 1 207.8970 89.87138 153.4253  
## 146 3 3 7 2 215.2907 78.04373 172.0909  
## 147 3 3 7 3 224.2049 71.42244 134.7762  
## 148 3 3 14 1 212.2028 93.93715 128.5616  
## 149 3 3 14 2 181.0239 52.60008 136.0711  
## 150 3 3 14 3 211.1153 28.44076 143.2121  
## 151 3 3 21 1 165.7052 75.40610 158.2575  
## 152 3 3 21 2 205.2503 32.64413 149.2312  
## 153 3 3 21 3 212.1109 74.66885 151.3837  
## 154 3 3 28 1 211.2551 49.64437 134.0283  
## 155 3 3 28 2 195.5297 84.32938 156.4756  
## 156 3 3 28 3 190.3306 78.24384 149.5816  
## 157 3 4 7 1 210.9272 43.03114 152.2676  
## 158 3 4 7 2 192.3411 65.05554 169.1430  
## 159 3 4 7 3 182.9806 91.07047 149.0289  
## 160 3 4 14 1 182.6072 30.22553 140.2068  
## 161 3 4 14 2 213.2696 79.18769 153.5502  
## 162 3 4 14 3 184.0047 19.26523 165.5577  
## 163 3 4 21 1 205.6721 93.56657 171.0774  
## 164 3 4 21 2 207.5199 53.83240 134.1837  
## 165 3 4 21 3 183.2253 29.00382 166.4267  
## 166 3 4 28 1 207.1683 56.66325 138.9922  
## 167 3 4 28 2 163.9405 85.40706 123.8408  
## 168 3 4 28 3 192.0196 52.22775 157.5339  
## 169 3 5 7 1 179.8847 62.67023 138.6411  
## 170 3 5 7 2 178.0338 21.67621 113.2354  
## 171 3 5 7 3 202.2552 83.04988 140.0776  
## 172 3 5 14 1 204.1655 91.21798 161.3751  
## 173 3 5 14 2 197.8153 73.46960 133.7633  
## 174 3 5 14 3 177.9741 19.30718 165.1708  
## 175 3 5 21 1 178.8867 25.00800 134.8445  
## 176 3 5 21 2 232.2885 23.58165 147.6786  
## 177 3 5 21 3 202.9656 83.50027 144.3304  
## 178 3 5 28 1 199.9668 54.99879 144.9900  
## 179 3 5 28 2 175.7909 75.78303 136.4803  
## 180 3 5 28 3 196.7169 44.79776 141.1150  
## 181 4 1 7 1 176.1078 75.08242 159.7930  
## 182 4 1 7 2 183.6163 85.83778 159.6740  
## 183 4 1 7 3 182.8743 16.13457 128.3311  
## 184 4 1 14 1 186.7735 57.77123 173.1907  
## 185 4 1 14 2 197.8409 28.14804 133.9371  
## 186 4 1 14 3 223.5096 26.10671 138.9002  
## 187 4 1 21 1 199.1279 56.29533 123.8238  
## 188 4 1 21 2 190.4068 68.33994 157.0937  
## 189 4 1 21 3 183.5101 66.56198 161.6577  
## 190 4 1 28 1 183.1305 31.82465 138.3614  
## 191 4 1 28 2 219.7824 92.92887 160.5294  
## 192 4 1 28 3 191.3354 33.82221 129.6342  
## 193 4 2 7 1 181.2974 74.83793 156.5583  
## 194 4 2 7 2 183.3328 33.97993 138.2861  
## 195 4 2 7 3 179.4315 30.38919 137.3704  
## 196 4 2 14 1 196.2531 73.91951 147.5861  
## 197 4 2 14 2 168.7092 20.78303 145.6522  
## 198 4 2 14 3 210.6343 76.19264 152.6001  
## 199 4 2 21 1 179.7267 85.16023 158.3410  
## 200 4 2 21 2 182.6576 30.59271 152.9042  
## 201 4 2 21 3 195.5256 97.39898 170.3636  
## 202 4 2 28 1 212.5447 96.25234 152.4584  
## 203 4 2 28 2 210.5167 81.28464 165.5217  
## 204 4 2 28 3 186.6626 34.21045 164.4393  
## 205 4 3 7 1 202.0477 55.07171 162.9186  
## 206 4 3 7 2 181.5011 49.81205 140.0840  
## 207 4 3 7 3 166.6446 71.05756 163.8639  
## 208 4 3 14 1 224.0358 15.59528 148.0358  
## 209 4 3 14 2 205.0720 35.24326 129.2180  
## 210 4 3 14 3 182.9356 61.02463 135.8380  
## 211 4 3 21 1 168.3208 80.37498 165.0325  
## 212 4 3 21 2 213.8078 95.50229 134.6602  
## 213 4 3 21 3 206.2385 73.75418 150.1325  
## 214 4 3 28 1 163.3196 99.13856 169.2266  
## 215 4 3 28 2 175.3313 29.65986 138.1546  
## 216 4 3 28 3 196.9914 31.02753 134.6033  
## 217 4 4 7 1 200.8662 89.79074 153.7396  
## 218 4 4 7 2 198.2220 23.75668 157.5609  
## 219 4 4 7 3 194.9959 53.78508 145.2605  
## 220 4 4 14 1 184.5933 17.41157 153.4038  
## 221 4 4 14 2 187.4725 84.64203 164.7954  
## 222 4 4 14 3 205.0366 31.10346 135.3263  
## 223 4 4 21 1 213.7802 18.44142 144.3353  
## 224 4 4 21 2 151.2307 74.14544 145.7129  
## 225 4 4 21 3 184.6035 66.66757 144.7062  
## 226 4 4 28 1 184.2677 18.53116 165.9033  
## 227 4 4 28 2 198.0367 70.63763 162.7714  
## 228 4 4 28 3 197.3324 66.01305 141.3345  
## 229 4 5 7 1 166.6810 31.59153 155.8745  
## 230 4 5 7 2 194.9469 92.41252 143.9371  
## 231 4 5 7 3 215.6775 97.27456 173.2913  
## 232 4 5 14 1 169.9806 84.76872 143.3572  
## 233 4 5 14 2 170.6676 46.11641 144.2301  
## 234 4 5 14 3 179.5180 17.54604 150.8841  
## 235 4 5 21 1 191.4444 78.03399 184.9274  
## 236 4 5 21 2 228.5983 74.77338 132.2750  
## 237 4 5 21 3 236.9520 20.10880 179.6382  
## 238 4 5 28 1 196.6213 42.86290 174.2765  
## 239 4 5 28 2 210.3746 55.49032 154.2762  
## 240 4 5 28 3 195.9978 82.78345 148.7505  
## 241 5 1 7 1 173.7985 25.48961 133.3109  
## 242 5 1 7 2 201.0728 41.12119 181.5726  
## 243 5 1 7 3 239.6264 84.73573 149.1429  
## 244 5 1 14 1 173.1539 24.80070 165.5939  
## 245 5 1 14 2 201.5582 58.99035 156.9753  
## 246 5 1 14 3 201.6668 83.50301 171.4875  
## 247 5 1 21 1 172.7840 86.27502 154.1324  
## 248 5 1 21 2 187.2866 47.40720 177.6913  
## 249 5 1 21 3 186.2470 84.36839 161.9336  
## 250 5 1 28 1 210.2502 35.75681 115.1307  
## 251 5 1 28 2 188.2304 18.36044 143.6110  
## 252 5 1 28 3 202.8154 24.00313 141.2183  
## 253 5 2 7 1 211.1675 30.08108 133.9421  
## 254 5 2 7 2 200.2923 72.27833 152.1352  
## 255 5 2 7 3 237.9650 54.30914 132.5030  
## 256 5 2 14 1 183.5091 43.99004 173.0055  
## 257 5 2 14 2 192.2992 80.35226 138.8951  
## 258 5 2 14 3 184.0132 60.04264 146.4188  
## 259 5 2 21 1 239.1061 57.32860 121.8858  
## 260 5 2 21 2 173.6221 88.65345 160.1443  
## 261 5 2 21 3 176.3772 58.07876 157.0123  
## 262 5 2 28 1 224.1229 33.69737 153.0148  
## 263 5 2 28 2 212.3549 74.57579 119.3630  
## 264 5 2 28 3 210.6226 96.81008 147.9701  
## 265 5 3 7 1 177.4965 71.48001 161.7911  
## 266 5 3 7 2 218.4847 84.75655 142.8314  
## 267 5 3 7 3 215.3399 61.63246 176.8809  
## 268 5 3 14 1 200.4631 28.05733 160.4510  
## 269 5 3 14 2 228.3507 34.07752 151.1308  
## 270 5 3 14 3 196.6298 98.91655 140.8185  
## 271 5 3 21 1 214.5644 83.13413 150.9102  
## 272 5 3 21 2 190.7848 77.28604 119.4103  
## 273 5 3 21 3 195.7770 78.94218 141.2598  
## 274 5 3 28 1 197.0142 63.02106 138.4525  
## 275 5 3 28 2 217.9896 20.15788 128.9212  
## 276 5 3 28 3 191.8066 46.19318 129.3196  
## 277 5 4 7 1 209.0243 30.57186 153.9044  
## 278 5 4 7 2 188.5493 62.71688 132.7792  
## 279 5 4 7 3 202.8765 67.49211 169.9434  
## 280 5 4 14 1 199.9591 58.35407 130.7941  
## 281 5 4 14 2 191.8636 56.29407 175.3823  
## 282 5 4 14 3 244.3684 46.29878 150.5984  
## 283 5 4 21 1 208.0861 32.06057 141.4743  
## 284 5 4 21 2 198.3786 70.61375 153.6715  
## 285 5 4 21 3 223.6406 18.83649 142.6763  
## 286 5 4 28 1 191.5107 44.43620 154.8965  
## 287 5 4 28 2 190.6564 89.75353 132.9059  
## 288 5 4 28 3 189.6338 29.05182 148.9332  
## 289 5 5 7 1 200.0971 89.81863 145.4909  
## 290 5 5 7 2 173.9726 94.85835 174.3520  
## 291 5 5 7 3 196.2865 18.29695 142.5501  
## 292 5 5 14 1 219.6705 42.63561 136.7062  
## 293 5 5 14 2 225.7137 75.61433 142.9469  
## 294 5 5 14 3 182.8641 84.16426 164.5623  
## 295 5 5 21 1 184.2217 56.28552 155.0820  
## 296 5 5 21 2 189.2600 75.88011 159.4957  
## 297 5 5 21 3 182.7276 97.52830 155.7213  
## 298 5 5 28 1 171.7147 23.57051 151.5928  
## 299 5 5 28 2 188.9540 62.34774 135.5944  
## 300 5 5 28 3 228.1529 54.23478 137.0418