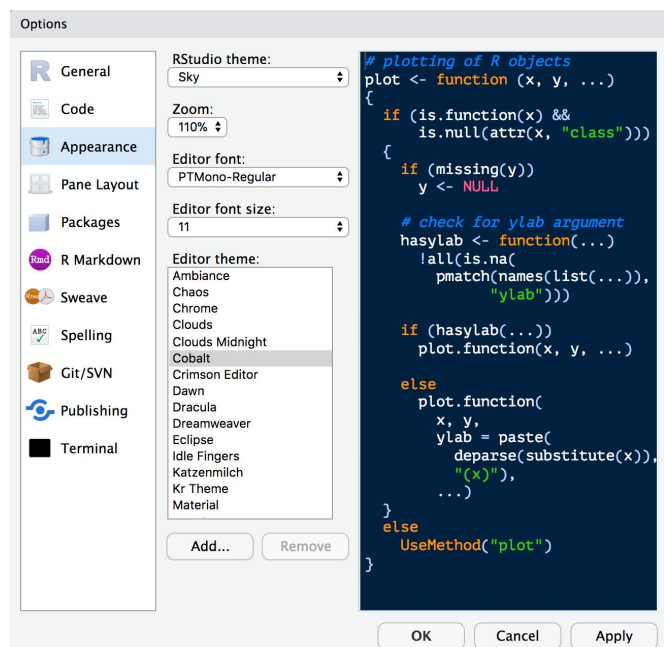


## GENERAL NOTES FOR R MARKDOWN

1. Every time you create a code chunk, add a title before it. It's up to you what size you want the header to be. I generally use 2 or 3 hashtags. Remember that one hashtag is the largest title size, and size decreases with additional hashtags
2. In addition to titling the code chunk in the text, it can be helpful to add a short name to the curly brackets `{r}`. You must put a space after the letter r, and I'd suggest a shorter description. For example, my first code chunk is called "Loading data" and then the chunk is named `{r data}`
3. It may be helpful to set a different theme with a dark background. That makes the code easier to look at. Go to [Rstudio] in the menu bar then click [Preferences] then click [Appearance]. I like "Cobalt" but you can use whatever theme you like.

```
### Loading data
```{r data}
```



## HOMEWORK #1

### 1. Prepare your document

- Create a new project file. Select "new directory" then "new project". Name your project and click "Browse" to choose where you want it located on your computer.
- Make a copy of the data I sent you, and relocate it within that folder.
- Create a new rmarkdown file. After you create it, save it within the same folder.
- You'll notice there is a lot of extra text in the default rmarkdown document. Feel free to check out the link mentioned (<http://rmarkdown.rstudio.com>) and check out the sample summary data and plot. Then delete everything after line 10.

### 2. Load your packages and data

- Create and title a new code chunk (do this every time you do something different so that your document remains organized). Install and load the appropriate packages. One new package that I want you to load is "readxl" to load an xlsx file.
- Load the raw data using the code "read\_xlsx()" instead of "read\_csv()". Remember that the data must be in the same folder as your project file in order for you to load it without specifying the path on the computer. View your raw data.
- Clean the variable names and give your data a short name that will be easy to type. I suggest using all lowercase. View your cleaned dataframe.

### 3. Create new dataframes

- Examine the revised variable names, so you can easily refer to relevant columns
- Create the following new dataframes that retain only certain columns:
  - 1) Only retain the columns for polygon ID, pin #, transect, and species code
  - 2) Only retain the columns for native/non-native, functional group, and layer
  - 3) Only retain the columns for data type, desired habitat, and scientific name
- Create the following new dataframes that filter for certain data:
  - 1) Only retain non-native species
  - 2) Only retain perennial forbs and grasses
  - 3) Only retain native species from the point-intercept data
  - 4) Only retain observations from pins less than or equal to 25

### 4. Create plots

- *Hint - you may need to adjust the first number in `scale_x_discrete` (highlighted in red below) to make the appropriate amount of space between the y-axis and the bar graph*
  - `scale_x_discrete(expand=c(0.5,0))`
- Create plots with X & Y labels and titles:
  - 1) Displaying the count of all species by layer.
  - 2) Displaying the count of non-native species by functional group. *Hint, you can use the dataframe you created earlier in step 3.*
  - 3) Displaying the count of native species from the point-intercept data by desired habitat. Change the color of the graph.
- If you want more practice customizing graphs, google how to:
  - Use different themes such as `theme_classic()` or `theme_minimal()` to change the overall look and feel of the graph
  - Change the font in the graph
  - Make x labels vertical or angled so they're not overlapping