

Homework_1_RK

Rachel Kenny

3/27/2020

Load packages

```
library(tidyverse)

## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2019c.1.0/
## zoneinfo/America/New_York'

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.0      v purrr  0.3.3
## v tibble  2.1.3      v dplyr  0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## Warning: package 'readr' was built under R version 3.4.4
## Warning: package 'stringr' was built under R version 3.4.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(rmarkdown)
library(readxl)

## Warning: package 'readxl' was built under R version 3.4.4

library(janitor)

## Warning: package 'janitor' was built under R version 3.4.4
```

Load data

```
ac_data_raw <- read_xlsx("OCWR_AC_2019_Data.xlsx")
# View(ac_data_raw)

ac_data <- ac_data_raw %>%
  clean_names()
# View(ac_data)

# Review the updated variable names
names(ac_data)

## [1] "desired_habitat" "polygon_id"      "transect"
## [4] "pin_number"      "data_type"       "species_code"
```

```
## [7] "scientific_name" "native_non_native" "functional_group"
## [10] "layer"
```

Create dataframes using select

```
# Create dataframe 1
df1 <- ac_data %>%
  select(polygon_id, pin_number, transect, species_code)

# Create dataframe 2
df2 <- ac_data %>%
  select(native_non_native, functional_group, layer)

# Create dataframe 3
df3 <- ac_data %>%
  select(data_type, desired_habitat, scientific_name)
```

Create dataframes using filter

```
# Create dataframe 1
df_1 <- ac_data %>%
  filter(native_non_native=="Non-Native")

# Create dataframe 2
# Option 1 - the "/" symbol means or
df_2 <- ac_data %>%
  filter(functional_group=="Perennial Forb - Other" | functional_group=="Perennial Forb - Late Flowering")

# Option 2 - %in% indicates all the following are included. This is the cleaner option.
df_2 <- ac_data %>%
  filter(functional_group %in% c("Perennial Forb - Other", "Perennial Forb - Late Flowering", "Perennial Forb - Early Flowering"))

# Create dataframe 3
# Option 1 - use the pipe operator to complete two filters
df_3 <- ac_data %>%
  filter(native_non_native=="Native") %>%
  filter(data_type=="T.PI")

# Option 2 - use the [ & ] symbol to complete two filters. This is the cleaner option.
df_3 <- ac_data %>%
  filter(native_non_native=="Native" & data_type=="T.PI")

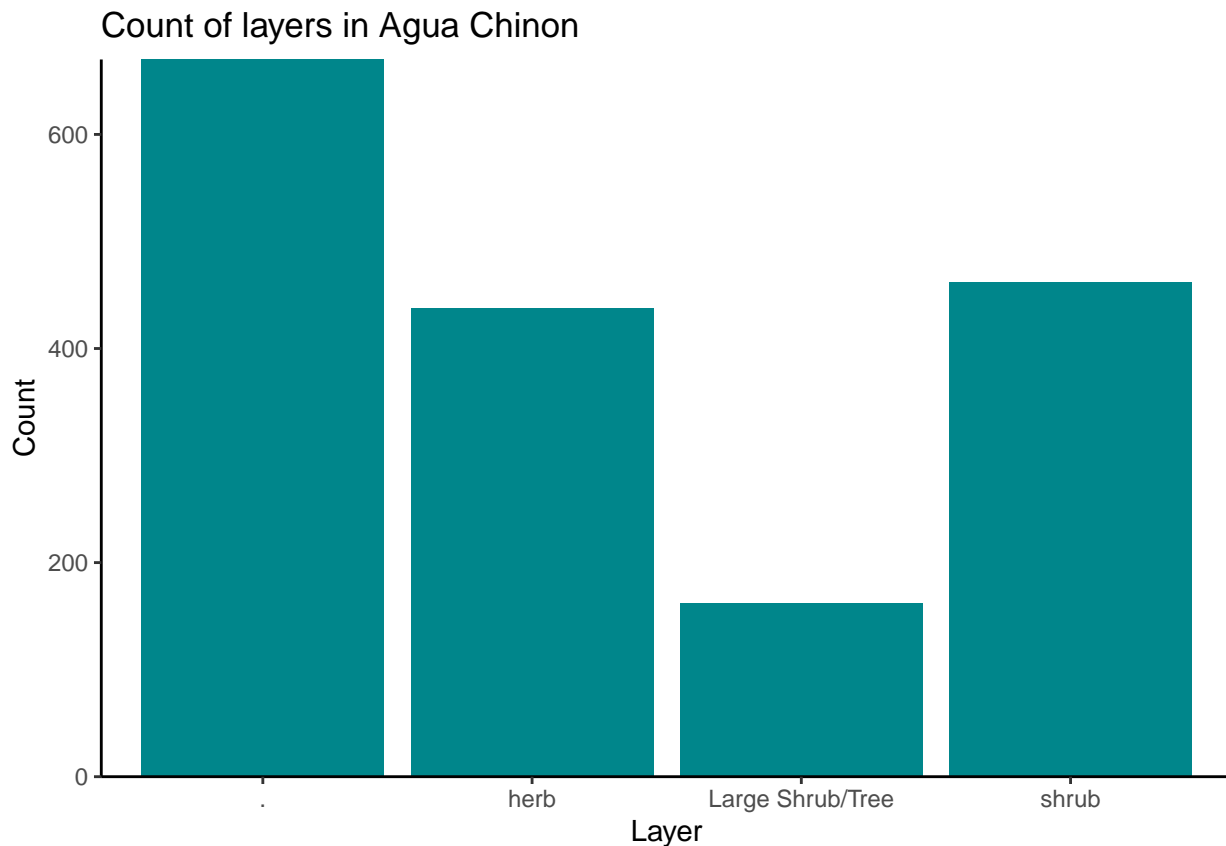
# Create dataframe 4
df_4 <- ac_data %>%
  filter(pin_number <= 25)
```

Plots

```
# Note the use of different themes in the code (theme_classic, theme_minimal, theme_grey)

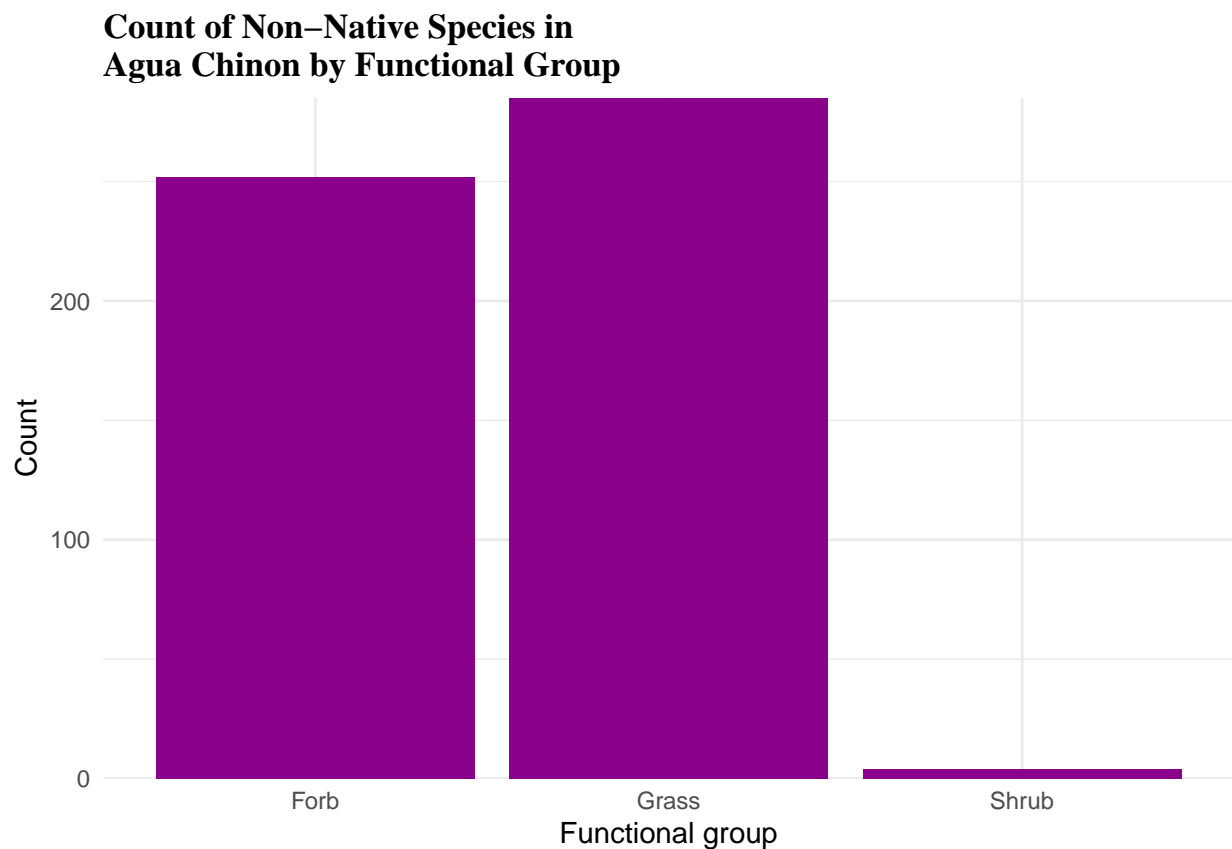
# Create plot 1
plot1 <- ggplot(ac_data, aes(layer)) +
  geom_bar(fill="turquoise4") +
  xlab("Layer") +
  ylab("Count") +
  ggtitle("Count of layers in Agua Chinon") +
  scale_x_discrete(expand=c(0.2,0)) +
  scale_y_continuous(expand=c(0,0)) +
  theme_classic()
```

plot1



```
# Create plot 2 - Note the use of [\n] to create a line break in the title, the use of theme() to make
plot2 <- ggplot(df_1, aes(functional_group)) +
  geom_bar(fill="magenta4") +
  xlab("Functional group") +
  ylab("Count") +
  ggtitle("Count of Non-Native Species in \nAgua Chinon by Functional Group") +
  scale_x_discrete(expand=c(0.3,0)) +
  scale_y_continuous(expand=c(0,0)) +
  theme_minimal() +
  theme(plot.title=element_text(face="bold", family = "serif"))
```

plot2



```
# Create plot 3 - Note the angling of the labels in the theme() code, the bold face of the axis titles,
plot3 <- ggplot(df_3, aes(desired_habitat)) +
  geom_bar(fill="gold3") +
  xlab("Desired habitat") +
  ylab("Count") +
  ggtitle("Count of Native Species in \nAgua Chinon by Desired Habitat") +
  scale_x_discrete(expand=c(0.15,0), labels = c("Elderberry\nScrubland", "Mulefat\nScrub", "Mulefat Scrub")) +
  scale_y_continuous(expand=c(0,0), limits = c(0,300)) +
  theme_grey() +
  theme(axis.text.x = element_text(angle = -45, hjust = 0), plot.title=element_text(face="bold"), axis.title.x=element_text(face="bold"))

plot3
```

