Plot_ly function for a histogram

This lab is preconfigured to include all dependencies (libraries, packages, and datasets) you'll need to complete your work in RStudio. You can practice, run test cases, and work on assignments from your browser.

Assignment Overview

In this lab, you are required to use the skills learned from this module to create a R Markdown which includes the usage of plot_ly() for a histogram.

Instructions

1. Please ensure the Crop_Range_GOES0901_CountJday.csv dataset is in your working directory.
2. Please create a R Markdown file, set the output format as html_document or html_document2.
3. Please use the data provided to conduct a Plot_ly function for a histogram based on the step mentioned below.

- Steps:
    - Install and load required packages.
    - Load and inspect the dataset.
    - Plot a histogram for the variable "FDCount" using plot_ly().

4. Please name your file as Yourname_plot_ly_histogram.RMD and save your .Rmd file in your lab.
5. Please knit your R Markdown file into a HTML file with the codes and graphs displayed in your file.

Important Reminder on Knit in this In-Browser RStudio option for this lab

This lab is hosted in an iframe that facilitates lab management features but consequently will prevent Knitting to HTML or Preview Notebook working by default. However, you can still Knit your files in lab by taking the following steps: - Step 1: Go to the "Help" icon in your lab toolbar (top right corner). - Step 2: Select the "Switch Back to the Old Experience" hyperlink (right click select if you'd like to keep both the submit and knit windows open) - Step 3: Knit your files to HTML or Preview Notebook. You should now be able to load and preview them in your lab appropriately.

More details can be found in the RStudio Lab - In-Browser Option Reading : https://www.coursera.org/lea (https://www.coursera.org/lea) rn/ball-state-university-data-visualization/supplement/E9jjS/rstudio-lab-in-browser-option

```
# Load the libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(plotly)
```

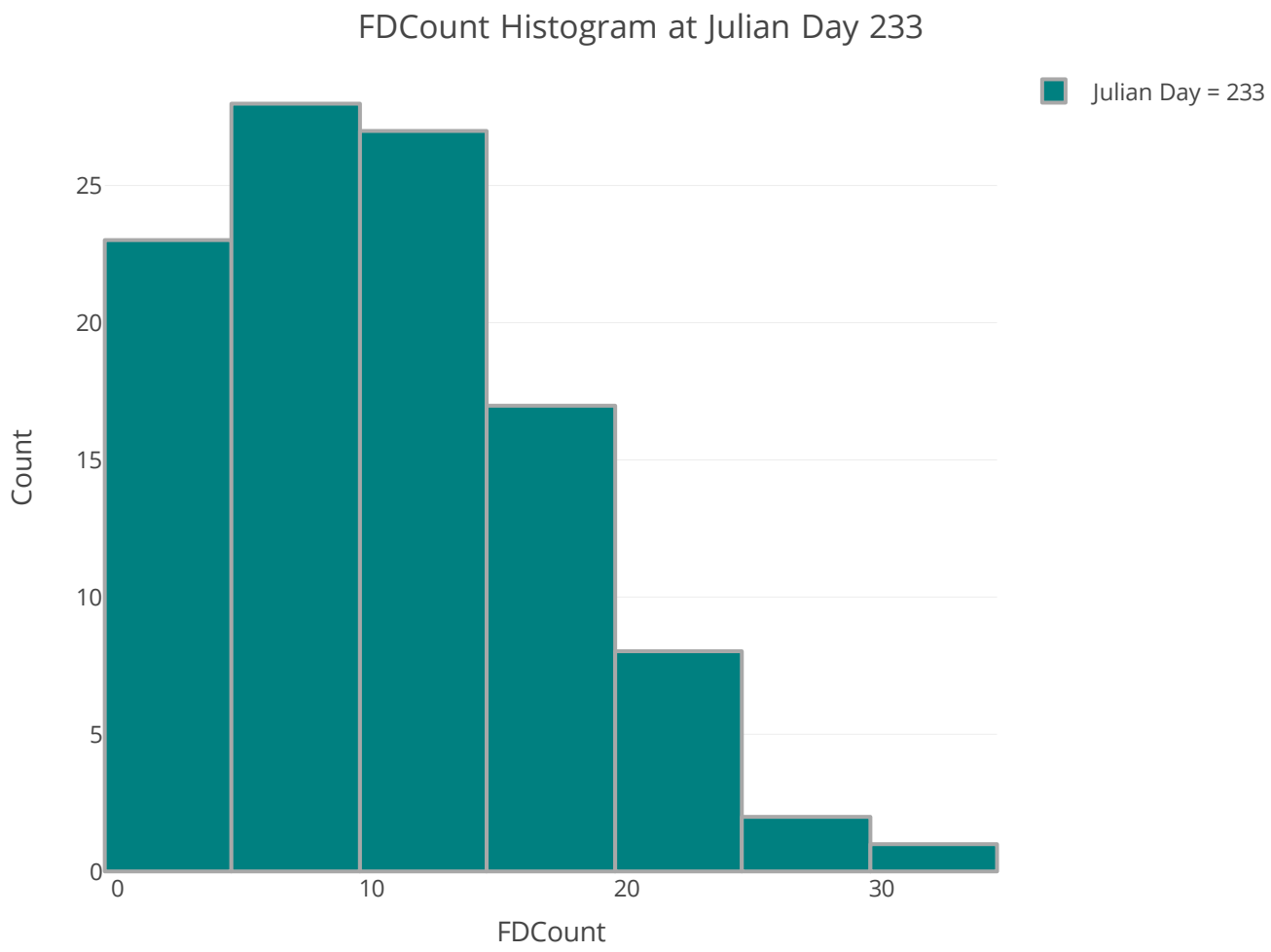
```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(readxl)
```

```
#Set your own t working directory
# setwd("./home/studio/")
##################Read csv and shape file into R
# learners will have this data loaded
land =  readxl::read_xlsx('Crop_Range_GOES0901_CountJday.xlsx')
```

0.2 Create a histogram using plot_ly()

```
# histogram
plot_ly(land, x = ~FDCount,nbinsx = 10) %>%
      filter(jday %in% 233) %>%
      add_histogram(marker = list(color = "teal",
                                  line = list(color = "darkgray",
                                              width = 2)),
                    name = "Julian Day = 233")%>%
      layout(title="FDCount Histogram at Julian Day 233",
             yaxis=list(title = "Count"),
             xaxis=list(title = "FDCount")) %>%
      layout(showlegend = TRUE)
```




```
# Another way
plot_ly(land, x = ~FDCount,nbinsx = 10, type
      ="histogram") %>%
      filter(jday %in% 233) %>%
      layout(title="FDCount Histogram at Julian
      Day 233",
             yaxis=list(title = "Count"),
             xaxis=list(title = "FDCount")) %>%
      layout(showlegend = TRUE)
```