Freezer_Door

Lehninger

2025-01-13

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this: # Import the CSV as *CSV UTF-8

```
# save file as CSV UTF-8, otherwise, date extraction does not work.
freezer_door_csv <- read.csv("c:/Users/pottsau/Documents/Y18-2202-05_169043_EventLog_01_29_2025_08_34_3
head(freezer_door_csv)
```

```
##        Date.Time     Brief.Desc
## 1 8/2/2024 11:42 Flash Settings
## 2 8/3/2024 11:42 Flash Settings
## 3 8/4/2024 11:42 Flash Settings
## 4 8/5/2024 11:42 Flash Settings
## 5 8/6/2024 11:42 Flash Settings
## 6 8/7/2024 11:42 Flash Settings
##
## 1 Setpoint: -80 Offset: 0 Lifeguard: 94 Time Delay: 0 Warm Alarm: -60 Cold Alarm: -90 Ext Ambient: 3
## 2 Setpoint: -80 Offset: 0 Lifeguard: 94 Time Delay: 0 Warm Alarm: -60 Cold Alarm: -90 Ext Ambient: 3
## 3 Setpoint: -80 Offset: 0 Lifeguard: 94 Time Delay: 0 Warm Alarm: -60 Cold Alarm: -90 Ext Ambient: 3
## 4 Setpoint: -80 Offset: 0 Lifeguard: 94 Time Delay: 0 Warm Alarm: -60 Cold Alarm: -90 Ext Ambient: 3
## 5 Setpoint: -80 Offset: 0 Lifeguard: 94 Time Delay: 0 Warm Alarm: -60 Cold Alarm: -90 Ext Ambient: 3
## 6 Setpoint: -80 Offset: 0 Lifeguard: 94 Time Delay: 0 Warm Alarm: -60 Cold Alarm: -90 Ext Ambient: 3
```

Create a TIBBLE

```
freezer_door <- as_tibble(freezer_door_csv)

head(freezer_door) #see if the data loads in well
```

```
## # A tibble: 6 x 3
##   Date.Time      Brief.Desc     Detailed.Desc
##   <chr>          <chr>          <chr>
## 1 8/2/2024 11:42 Flash Settings Setpoint: -80 Offset: 0 Lifeguard: 94 Time Dela~
```

```
## 2 8/3/2024 11:42 Flash Settings Setpoint: -80 Offset: 0 Lifeguard: 94 Time Dela~
## 3 8/4/2024 11:42 Flash Settings Setpoint: -80 Offset: 0 Lifeguard: 94 Time Dela~
## 4 8/5/2024 11:42 Flash Settings Setpoint: -80 Offset: 0 Lifeguard: 94 Time Dela~
## 5 8/6/2024 11:42 Flash Settings Setpoint: -80 Offset: 0 Lifeguard: 94 Time Dela~
## 6 8/7/2024 11:42 Flash Settings Setpoint: -80 Offset: 0 Lifeguard: 94 Time Dela~
```

Separate the days from the times

```
freezer_door_datesep <- freezer_door %>% # make a new tibble with date, time, and Date.Time.
  transmute(Brief.Desc, # add new column and keep existing columns
    Date.Time = mdy_hm(Date.Time), # parse the date and time string
    date = as.Date(Date.Time), # extract the date
    time = format(Date.Time, "%H:%M") # extract the time in HH:MM format
  )

head(freezer_door_datesep) # visualize the tibble's top rows
```

```
## # A tibble: 6 x 4
##   Brief.Desc     Date.Time           date       time
##   <chr>          <dttm>              <date>     <chr>
## 1 Flash Settings 2024-08-02 11:42:00 2024-08-02 11:42
## 2 Flash Settings 2024-08-03 11:42:00 2024-08-03 11:42
## 3 Flash Settings 2024-08-04 11:42:00 2024-08-04 11:42
## 4 Flash Settings 2024-08-05 11:42:00 2024-08-05 11:42
## 5 Flash Settings 2024-08-06 11:42:00 2024-08-06 11:42
## 6 Flash Settings 2024-08-07 11:42:00 2024-08-07 11:42
```

Group and Count the days with Door Closing Events

```
door_closings <- freezer_door_datesep %>% # parsing the tibble and making a  new tibble
  filter(date > ymd("2024-12-01")) %>% # filter for older dates before our project commenced
  select(date, Brief.Desc) %>% # selecting interested columns
  group_by(date) %>% # grouping the days together
  filter(Brief.Desc == "Door Close Event") %>% # filtering for the door closing event
  summarize(DoorClosings = n()) # counting the number of days with door closing events

door_closings
```

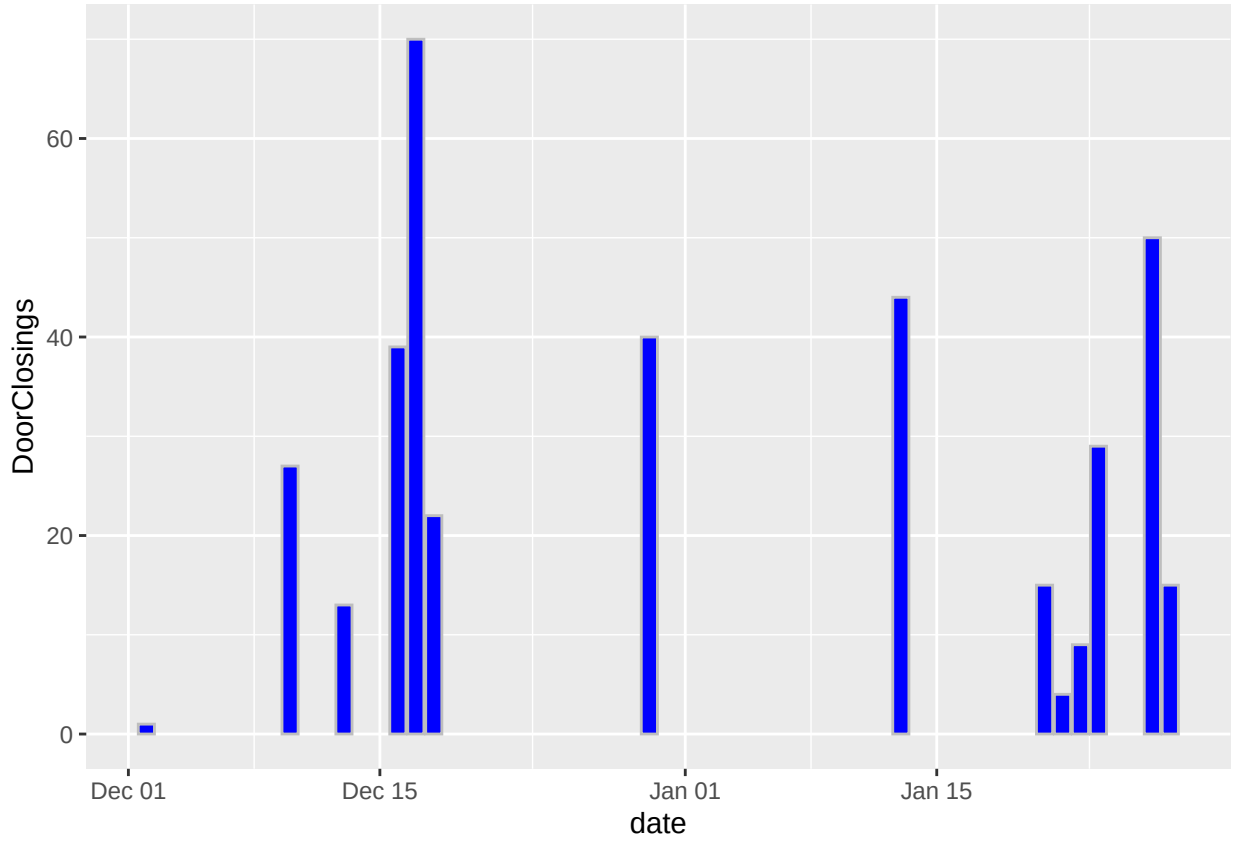
```
## # A tibble: 14 x 2
##    date       DoorClosings
##    <date>            <int>
##  1 2024-12-02            1
##  2 2024-12-10           27
##  3 2024-12-13           13
##  4 2024-12-16           39
##  5 2024-12-17           70
##  6 2024-12-18           22
##  7 2024-12-30           40
```

```
##  8 2025-01-13           44
##  9 2025-01-21           15
## 10 2025-01-22            4
## 11 2025-01-23            9
## 12 2025-01-24           29
## 13 2025-01-27           50
## 14 2025-01-28           15
```

Write Tibble to CSV

```
write.csv(door_closings, "c:/Users/pottsau/Documents/DoorClosings_as_of_2025_01_27.csv", row.names = FA
```

```
door_closings_bar_graph <- ggplot(data = door_closings, mapping = aes(x = date, y = DoorClosings)) +
  geom_col(fill = "blue", color = "grey") #+
  #labs(title = "Count of Freezer Door Closes",
   #    x = "Days When the Door is Closed",
   #    y = "Number of Closes") +
  #theme_light()

door_closings_bar_graph
```



Import percent complete CSV UTF-8 for overlaying onto Bar graph

```
# save file as CSV UTF-8, otherwise, date extraction does not work.
percent_complete_csv <- read.csv("c:/Users/pottsau/Documents/Percent_Complete Prime_Air_2025.csv") #loa
percent_complete <- as.tibble(percent_complete_csv)
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` instead.
## i The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
percent_complete
```

```
## # A tibble: 5 x 8
##   PercentDate PercentFound PercentMissing X     X.1   X.2   X.3   X.4
##   <chr>              <int>          <int> <lgl> <lgl> <lgl> <lgl> <lgl>
## 1 1/5/2025               5             95 NA    NA    NA    NA    NA
## 2 1/13/2025             10             90 NA    NA    NA    NA    NA
## 3 1/17/2025             19             81 NA    NA    NA    NA    NA
## 4 1/27/2025             32             68 NA    NA    NA    NA    NA
## 5 1/29/2025             45             55 NA    NA    NA    NA    NA
```

```
percent_complete_date <- percent_complete %>% # make a new tibble with date, time, and date.
  mutate(PercentFound, PercentMissing, # add new column and keep existing columns
    PercentDate = mdy(PercentDate)) # parse the chr string as date



head(percent_complete_date) #visualize the new tibble
```

```
## # A tibble: 5 x 8
##   PercentDate PercentFound PercentMissing X     X.1   X.2   X.3   X.4
##   <date>             <int>          <int> <lgl> <lgl> <lgl> <lgl> <lgl>
## 1 2025-01-05             5             95 NA    NA    NA    NA    NA
## 2 2025-01-13            10             90 NA    NA    NA    NA    NA
## 3 2025-01-17            19             81 NA    NA    NA    NA    NA
## 4 2025-01-27            32             68 NA    NA    NA    NA    NA
## 5 2025-01-29            45             55 NA    NA    NA    NA    NA
```

Generate line graphs of the percent found/missing

```
#make a line graph of the percent found in RED
percent_found_line <- #ggplot(data = percent_complete) +
  geom_line(data = percent_complete_date, aes(x = PercentDate, y = PercentFound, color = "red", group =

percent_found_line
```

```
## mapping: x = ~PercentDate, y = ~PercentFound, colour = red, group = 1
```

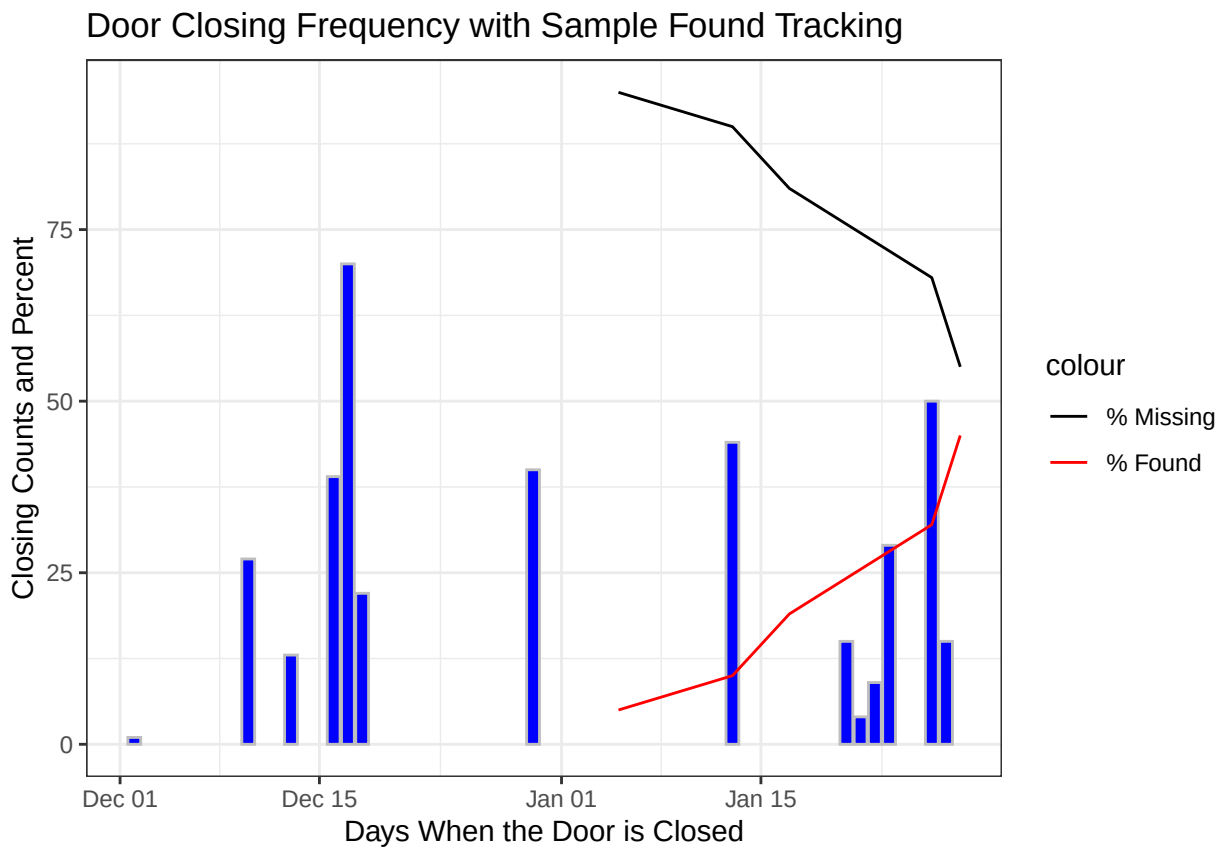
```
## geom_line: na.rm = FALSE, orientation = NA
## stat_identity: na.rm = FALSE
## position_identity
```

```
#make a 2nd line graph in the percent missing in BLACK
percent_missing_line <- #ggplot(data = percent_complete) +
  geom_line(data = percent_complete_date, aes(x = PercentDate, y = PercentMissing, color = "black", grou

percent_missing_line
```

```
## mapping: x = ~PercentDate, y = ~PercentMissing, colour = black, group = 1
## geom_line: na.rm = FALSE, orientation = NA
## stat_identity: na.rm = FALSE
## position_identity
```

layer the 3 graphs together for an neat overlay

```
bar_and_lines <- door_closings_bar_graph +
  percent_found_line +
  percent_missing_line +
  scale_color_manual(values = c("red" = "red", "black" = "black"), labels = c("% Missing", "% Found")) +
  labs(title = "Door Closing Frequency with Sample Found Tracking",
       x = "Days When the Door is Closed",
       y = "Closing Counts and Percent") +
  theme_bw()

print(bar_and_lines)
```


Door Closing Frequency with Sample Found Tracking
Closing Counts and Percent
75
50
25
0
Dec 01
Dec 15
Jan 01
Jan 15
Days When the Door is Closed
colour
% Missing
% Found