filtering

Mick McQuaid

2020-02-01

Initialization

```
library(tidyverse)
library(data.table)
nyc311<-fread("mini311.csv")
names(nyc311)<-names(nyc311) %>%
  stringr::str_replace_all("\\s", ".")
```

Creating the main data frame

The following chunk creates everything we need for a stacked bar chart. There is one problem, though. If we try to sort the bars by length, we'll fail because the complaints are per borough. We need a column with total complaints for all boroughs.

```
df1 <- nyc311 %>%
    group_by(Complaint.Type,Borough) %>%
    subset(select=c(Complaint.Type,Borough)) %>%
    summarize(Complaints = n()) %>%
    filter(Complaints > 100)
```

Joining data frames

The quickest way (not perhaps the most elegant) to get a column of total complaints for all boroughs is to group by complaint type, add the complaints per complaint type, then join the result to the data frame created above. Warning: only do `full_join()` with very small data frames or you will run out of memory before the join is completed.

```
df2 <- df1 %>%
    group_by(Complaint.Type) %>%
    summarize(totComplaints=sum(Complaints))

df3<-full_join(df1,df2)
```

```
## Joining, by = "Complaint.Type"
```

Creating the barchart

Now we can create the barchart and order the bars by the `totComplaints` column, where there is an identical entry for each `Complaint.Type/Borough` pair. After you run this code, look at `df1`, `df2`, and `df3` if you're at all confused by what is going on.

```
df3 %>%
    ggplot( aes(x=reorder(Complaint.Type,totComplaints), y=Complaints, fill=Borough)) +
    xlab("Category") +
    geom_bar(stat="identity") +
```

```
  coord_flip() +
  ggtitle("Top Complaints by Type and Borough")
```