Factors and Dates Assignment

Kim Roth & Johnathan Clementi

4/14/2020

These are exercises based on the exercises in R for Data Science by Wickham and Grolemund.

The website where the book appears is (and will always be) **free to use**, and is licensed under the [Creative Commons Attribution-NonCommercial-NoDerivs 3.0](http://creativecommons.org/licenses/by-nc-nd/3.0/us/) License.

library(tidyverse)

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

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

gss\_cat #our data set for the first part. See the book for details

## # A tibble: 21,483 x 9  
## year marital age race rincome partyid relig denom tvhours  
## <int> <fct> <int> <fct> <fct> <fct> <fct> <fct> <int>  
## 1 2000 Never ma~ 26 White $8000 to ~ Ind,near r~ Protesta~ Souther~ 12  
## 2 2000 Divorced 48 White $8000 to ~ Not str re~ Protesta~ Baptist~ NA  
## 3 2000 Widowed 67 White Not appli~ Independent Protesta~ No deno~ 2  
## 4 2000 Never ma~ 39 White Not appli~ Ind,near r~ Orthodox~ Not app~ 4  
## 5 2000 Divorced 25 White Not appli~ Not str de~ None Not app~ 1  
## 6 2000 Married 25 White $20000 - ~ Strong dem~ Protesta~ Souther~ NA  
## 7 2000 Never ma~ 36 White $25000 or~ Not str re~ Christian Not app~ 3  
## 8 2000 Divorced 44 White $7000 to ~ Ind,near d~ Protesta~ Luthera~ NA  
## 9 2000 Married 44 White $25000 or~ Not str de~ Protesta~ Other 0  
## 10 2000 Married 47 White $25000 or~ Strong rep~ Protesta~ Souther~ 3  
## # ... with 21,473 more rows

gssCopy = gss\_cat # Make copy for edititing

#Factors

1. Explore the distribution of partyid (reported political party preference). What is the default order of factors? Make a plot with reordered factors. Describe what your best graph tells you about partid. 1A. Default order: No answer, Don’t Know, Other party, Strong republican, Not str republican, Ind, near rep, Independent, Ind, near dem, Not str democrat, Strong democrat

count(gss\_cat, partyid)

## # A tibble: 10 x 2  
## partyid n  
## <fct> <int>  
## 1 No answer 154  
## 2 Don't know 1  
## 3 Other party 393  
## 4 Strong republican 2314  
## 5 Not str republican 3032  
## 6 Ind,near rep 1791  
## 7 Independent 4119  
## 8 Ind,near dem 2499  
## 9 Not str democrat 3690  
## 10 Strong democrat 3490

gssCopy = mutate(gssCopy, partyid = factor(partyid, levels = c("Strong democrat", "Not str democrat", "Ind,near dem", "Independent", "Ind,near rep", "Not str republican", "Strong republican", "Other party", "Don't know", "No answer")))  
  
count(gssCopy, partyid)

## # A tibble: 10 x 2  
## partyid n  
## <fct> <int>  
## 1 Strong democrat 3490  
## 2 Not str democrat 3690  
## 3 Ind,near dem 2499  
## 4 Independent 4119  
## 5 Ind,near rep 1791  
## 6 Not str republican 3032  
## 7 Strong republican 2314  
## 8 Other party 393  
## 9 Don't know 1  
## 10 No answer 154

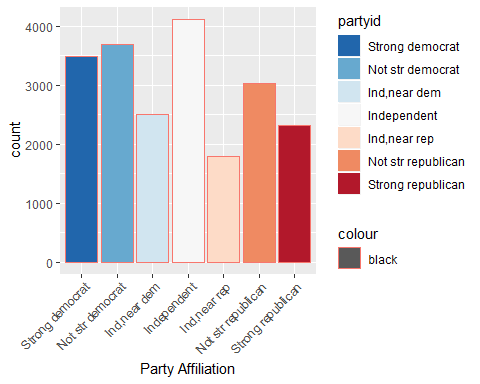
Load ColorBrewer

### Load the package or install if not present  
if (!require("RColorBrewer")) {  
install.packages("RColorBrewer")  
library(RColorBrewer)  
}

## Loading required package: RColorBrewer

1A(pt2). This figure shows the distribution of political party affiliation amoung people sampled in the GSS. There seem to be more people that identify as Democrat (and not strong democrat) than identify as Republican (or not strong republican). I filtered out 3rd party, Don’t know, and didn’t answer data to get the color ramp to fit well with the Democrat - Republican (Left - Right) spread.

ggplot(data = filter(gssCopy, partyid != "No answer" & partyid != "Don't know" & partyid != "Other party")) +  
 geom\_bar(mapping = aes(x = partyid, fill = partyid, color = 'black')) +  
 scale\_fill\_brewer(palette = "RdBu", direction = -1) +  
 labs(x = "Party Affiliation") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



2.What is the most common denom in this survey? What’s the most common marital? 2A. Most common denom is “Not applicable”; most common marital is “Married”

denomCount = count(gssCopy, denom)  
denomCount = arrange(denomCount, desc(n))  
denomCount

## # A tibble: 30 x 2  
## denom n  
## <fct> <int>  
## 1 Not applicable 10072  
## 2 Other 2534  
## 3 No denomination 1683  
## 4 Southern baptist 1536  
## 5 Baptist-dk which 1457  
## 6 United methodist 1067  
## 7 Episcopal 397  
## 8 Lutheran-dk which 267  
## 9 Presbyterian-dk wh 244  
## 10 Methodist-dk which 239  
## # ... with 20 more rows

maritalCount = count(gssCopy, marital)  
maritalCount = arrange(maritalCount, desc(n))  
maritalCount

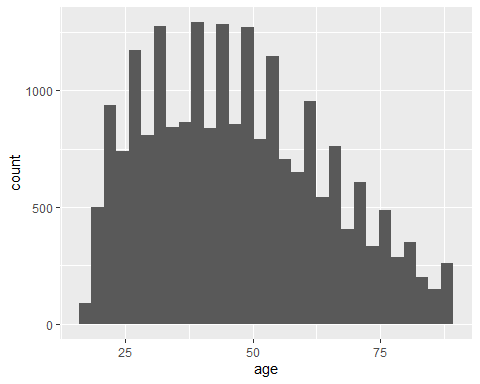
## # A tibble: 6 x 2  
## marital n  
## <fct> <int>  
## 1 Married 10117  
## 2 Never married 5416  
## 3 Divorced 3383  
## 4 Widowed 1807  
## 5 Separated 743  
## 6 No answer 17

1. Examine the distribution of age. Are there any surprises about the distribution? Would you prefer to use the mean or the median to summarize age? 3A. The distribution is strange - there seem to be short boom and busts between age classes, that almost make the figure look like a hand with too many fingers. I’m not sure what would cause this. I’d prefer the median over the mean: the 1st and 3rd Standard deviations are equal distance from the median (13), whereas they are different values when using the mean. Additionally, the median is a whole number (age is traditionally measured in whole years).

ggplot(data = gssCopy) +  
 geom\_histogram(mapping = aes(x = age))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 76 rows containing non-finite values (stat\_bin).



summary(gssCopy$age, na.rm = TRUE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 18.00 33.00 46.00 47.18 59.00 89.00 76

1. For rincome, what order are the factors in? Is this a useful order? 4A. Order of factors: No answer, Don’t know, Refused, $25000 or more, $20000 - 24999, $15000 - 19999, $10000 - 14999, $8000 to 9999, $7000 to 7999, $6000 to 6999, $5000 to 5999, $4000 to 4999, $3000 to 3999, $1000 to 2999, Lt $1000, Not applicable

The order is useful for all of the numeric ranges. However, the first three factors: “No answer, Don’t know, Refused” should be moved to the bottom of the factor list.

count(gssCopy, rincome)

## # A tibble: 16 x 2  
## rincome n  
## <fct> <int>  
## 1 No answer 183  
## 2 Don't know 267  
## 3 Refused 975  
## 4 $25000 or more 7363  
## 5 $20000 - 24999 1283  
## 6 $15000 - 19999 1048  
## 7 $10000 - 14999 1168  
## 8 $8000 to 9999 340  
## 9 $7000 to 7999 188  
## 10 $6000 to 6999 215  
## 11 $5000 to 5999 227  
## 12 $4000 to 4999 226  
## 13 $3000 to 3999 276  
## 14 $1000 to 2999 395  
## 15 Lt $1000 286  
## 16 Not applicable 7043

1. For each factor in gss\_cat identify whether the order of the levels is arbitrary or principled. 5A. marital: arbitrary race: principled rincome: principled partyid: principled relig: arbitrary denom: principled

count(gssCopy, marital)  
count(gssCopy, race)  
count(gssCopy, rincome)  
count(gssCopy, partyid)  
count(gssCopy, relig)  
count(gssCopy, denom)

1. How could you collapse partyid into a small set of categories? Collapse partyid into a small set of categories. 6A. I would collapse into 4 categories: Democrat, Independent, Republican, Other

gssCopy = gssCopy %>%  
 mutate(partyid = fct\_collapse(partyid,  
 Democrat = c("Strong democrat", "Not str democrat"),  
 Independent = c("Ind,near dem", "Independent", "Ind,near rep"),  
 Republican = c("Strong republican", "Not str republican"),  
 Other = c("No answer", "Don't know", "Other party")  
 ))  
  
count(gssCopy, partyid)

## # A tibble: 4 x 2  
## partyid n  
## <fct> <int>  
## 1 Democrat 7180  
## 2 Independent 8409  
## 3 Republican 5346  
## 4 Other 548

1. Use fct\_lump to collapse marital. What got lumped together and why? 7A.No answer, Seperated, and Widowed were all lumped together because they were the three factors with the lowest counts (least common)

count(gssCopy, marital)

## # A tibble: 6 x 2  
## marital n  
## <fct> <int>  
## 1 No answer 17  
## 2 Never married 5416  
## 3 Separated 743  
## 4 Divorced 3383  
## 5 Widowed 1807  
## 6 Married 10117

gssCopy = gssCopy %>%  
 mutate(marital = fct\_lump(marital))  
  
count(gssCopy, marital)

## # A tibble: 4 x 2  
## marital n  
## <fct> <int>  
## 1 Never married 5416  
## 2 Divorced 3383  
## 3 Married 10117  
## 4 Other 2567

#Dates

library(lubridate) #unlike most of the tidyverse libraries, this must be called seperately.

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

1. What happens if you parse a string that contains invalid dates? 8A. Lubridate throws an failed to parse error

* ymd(c("2020-01-10", "2020-13-41", "date"))

1. Use the appropriate lubridate function to parse each of the following dates or date-times:

* d1 <- "Apr 14, 2020"  
  d2 <- "2001-21-02"  
  d3 <- c("14-01-2021", "15-02-2021")  
  d4 <- "01/31/2017 08:01 EST"  
  d5 <- "2022-12-31 23:59:59"
* 9A.

d1=mdy(d1)  
d2=ydm(d2)  
d3=dmy(d3)  
d4=mdy\_hm(d4, tz = "America/New\_York")  
d5=ymd\_hms(d5)  
d1

## [1] "2020-04-14"

d2

## [1] "2001-02-21"

d3

## [1] "2021-01-14" "2021-02-15"

d4

## [1] "2017-01-31 08:01:00 EST"

d5

## [1] "2022-12-31 23:59:59 UTC"

1. Figure out how old you are in years exactly using R. 10A.

JMCBirthday = ymd\_hm("1996-05-12 16:11 EST", tz = "America/New\_York")  
JMCAge = now() - JMCBirthday  
as.duration(JMCAge)

## [1] "754618281.025277s (~23.91 years)"

1. Why is there months() but no dmonths()? 11A. The smallest unit of time in a duration is the second. For each d\*unit (second, minute, hour, etc), there is a known number of seconds in each of those units. Because months are variable in their length, the duration in seconds for each month is not constant.
2. Explain days(overnight \* 1) to someone who has just started learning R. How does it work? 12A. To account for periods that exist in multiple days (i.e. a person begins traveling on Sunday and arrives at their destination on Monday), we must tell R which direction to calculate the time difference, so it does not result in a negative time value (when departure time is greater than arrival time).
3. Create a vector of dates giving the first day of every month in the current year. 13A.

firstDay = floor\_date(today(), "year")  
firstDayList = seq.Date(firstDay, (firstDay + years(1)), "month")  
firstDayList

## [1] "2020-01-01" "2020-02-01" "2020-03-01" "2020-04-01" "2020-05-01"  
## [6] "2020-06-01" "2020-07-01" "2020-08-01" "2020-09-01" "2020-10-01"  
## [11] "2020-11-01" "2020-12-01" "2021-01-01"

1. Why can’t (today() %--% (today() + years(1)) / months(1)) work? 14A. This is only counting the number of first day of every month (12), it does not create the date and append it to a list.
2. Give at least one example that makes working with time zones difficult. 15A. There are some locations which fall in the same time zone but have different policies on observing day-light-savings time, time calculations may need extra care in those cases.