Class Activity 10

Your name here

March 19 2024

Your Turn 1

```
students <- tibble(
  id = 1:24,
  grade = sample(c("9th", "10th", "11th"), 24, replace = TRUE),
  region = sample(c("North America", "Europe", "Asia",
                    "South America", "Middle East", "Africa"), 24, replace = TRUE),
  score = round(runif(24,50, 100))
)
```

a. Create a new column `grade_fac` by converting the grade column into a factor. Reorder the levels of `grade_fac` to be "9th", "10th", and "11th". Sort the dataset based on the `grade_fac` column.

Answer:

```
students_a <- students %>%
  mutate(grade_fac = factor(grade)) %>%
  mutate(grade_fac = fct_relevel(grade_fac, c("9th", "10th", "11th"))) %>%
  arrange(grade_fac)
print(students_a, n = 24)
# A tibble: 24 x 5
      id grade region        score grade_fac
   <int> <chr> <chr>         <dbl> <fct>
 1     1 9th   Asia             88 9th
 2     5 9th   North America    59 9th
 3     9 9th   Asia             95 9th
 4    11 9th   Middle East      77 9th
 5    13 9th   Middle East      52 9th
 6    14 9th   South America    67 9th
 7    22 9th   Europe           77 9th
 8    24 9th   South America    77 9th
 9     3 10th  Middle East      87 10th
10     4 10th  Africa           64 10th
11     8 10th  Africa           87 10th
12    12 10th  Africa           85 10th
```

```
13    16 10th  Middle East      95 10th
14    17 10th  Europe           84 10th
15    18 10th  Europe           85 10th
16    19 10th  Europe           83 10th
17     2 11th  North America    57 11th
18     6 11th  Africa          100 11th
19     7 11th  South America    54 11th
20    10 11th  South America    74 11th
21    15 11th  Europe           98 11th
22    20 11th  South America    57 11th
23    21 11th  Africa           85 11th
24    23 11th  South America    52 11th
```

b. Create a new column `region_fac` by converting the `region` column into a factor. Collapse the `region_fac` levels into two categories: "Male" and "Female". Count the number of students in each collapsed region category.

```
students_b <- students_a %>%
  mutate(region_fac = factor(region)) %>%
  mutate(region_collapsed = fct_collapse(region_fac,
                                         Americas = c("North America", "South America"),
                                         EMEA = c("Europe", "Middle East", "Africa"),
                                         Asia = "Asia")) %>%
  count(region_collapsed)
print(students_b)
# A tibble: 3 x 2
  region_collapsed     n
  <fct>            <int>
1 EMEA                14
2 Asia                 2
3 Americas             8
```

c. Create a new column `grade_infreq` that is a copy of the `grade_fac` column. Reorder the levels of `grade_infreq` based on their frequency in the dataset. Print the levels of `grade_infreq` to check the ordering.

```
students_c <- students_a %>%
  mutate(grade_infreq = grade_fac) %>%
  mutate(grade_infreq = fct_infreq(grade_infreq))

levels(students_c$grade_infreq)
[1] "9th"  "10th" "11th"
```

d. Create a new column `grade_lumped` by lumping the least frequent level of the `grade_fac` column into an 'Others' category.

Count the number of students in each of the categories of the `grade_lumped` column.

```
students_d <- students_a %>%
  mutate(grade_lumped = fct_lump(grade_fac, n = 1, other_level = "Others")) %>%
  count(grade_lumped)
students_d
# A tibble: 3 x 2
  grade_lumped     n
  <fct>         <int>
1 9th               8
2 10th              8
3 11th              8
```

Your Turn 2

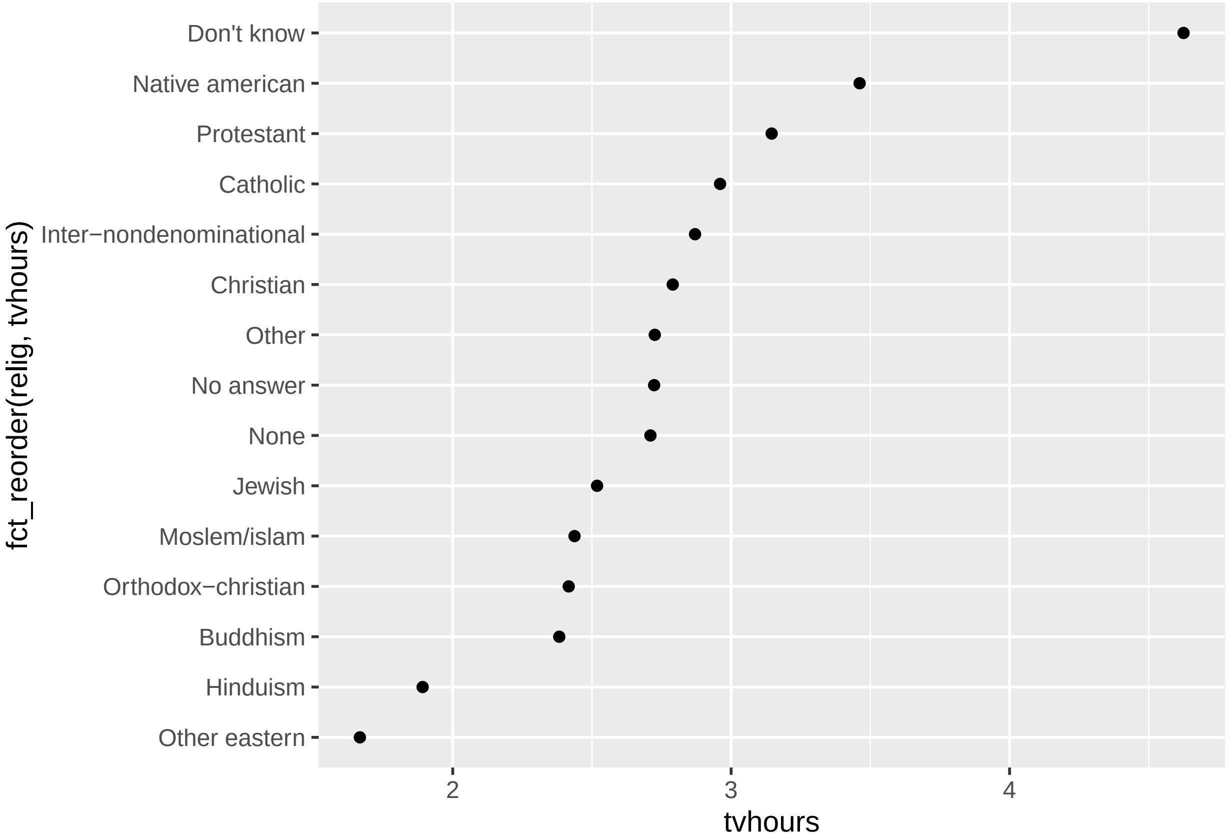
Lets import the `gss_cat` dataset from the `forcats` library. This datast contains a sample of categorical variables from the General Social survey.

```
# import gss_cat dataset from forcats library
forcats::gss_cat
# 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 married    26 White $8000 to 9999  Ind,near ~ Prot~ Sout~      12
 2  2000 Divorced         48 White $8000 to 9999  Not str r~ Prot~ Bapt~      NA
 3  2000 Widowed          67 White Not applicable Independe~ Prot~ No d~       2
 4  2000 Never married    39 White Not applicable Ind,near ~ Orth~ Not ~       4
 5  2000 Divorced         25 White Not applicable Not str d~ None  Not ~       1
 6  2000 Married          25 White $20000 - 24999 Strong de~ Prot~ Sout~      NA
 7  2000 Never married    36 White $25000 or more Not str r~ Chri~ Not ~       3
 8  2000 Divorced         44 White $7000 to 7999  Ind,near ~ Prot~ Luth~      NA
 9  2000 Married          44 White $25000 or more Not str d~ Prot~ Other       0
10  2000 Married          47 White $25000 or more Strong re~ Prot~ Sout~       3
# i 21,473 more rows
```

Use `gss_cat` to answer the following questions.

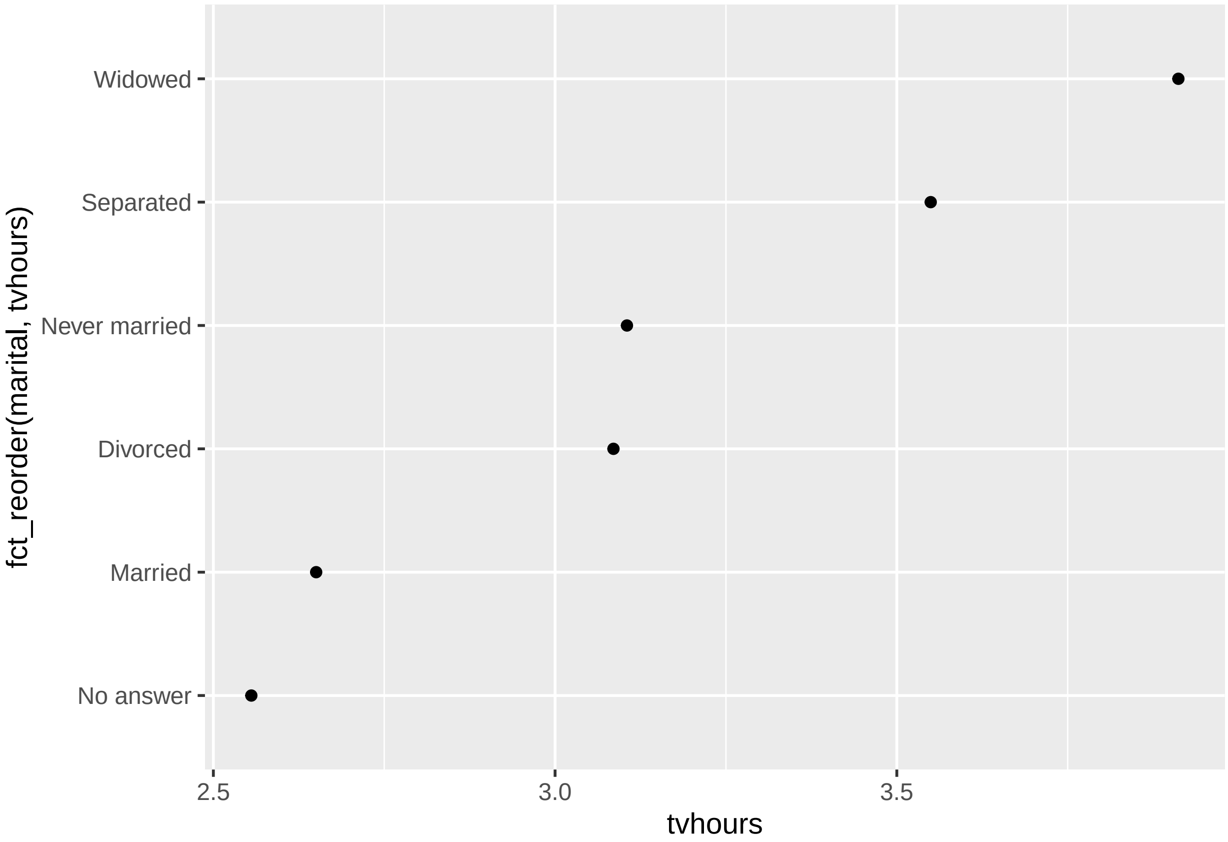
a. Which religions watch the least TV?

```
# your r-code

gss_cat %>%
  drop_na(tvhours) %>%
  group_by(relig) %>%
  summarize(tvhours = mean(tvhours)) %>%
  ggplot(aes(tvhours, fct_reorder(relig, tvhours))) +
    geom_point()
```



b. Do married people watch more or less TV than single people?

```
# your r-code

gss_cat %>%
  drop_na(tvhours) %>%
  group_by(marital) %>%
  summarize(tvhours = mean(tvhours)) %>%
  ggplot(aes(tvhours, fct_reorder(marital, tvhours))) +
    geom_point()
```



c. Collapse the `marital` variable to have levels `Married`, `not_married`, and `No answer` .Include `"Never married"`, `"Divorced"`, and “`Widowed"` in `not_married`

```
# your r-code

gss_cat %>%
  drop_na(tvhours) %>%
  select(marital, tvhours) %>%
  mutate(
    maritalStatus =
      fct_collapse(
        marital,
        Married = c("Married",
                         "Separated"),
        not_married = c("Never married",
                    "Divorced",
                    "Widowed"))
  )
# A tibble: 11,337 x 3
   marital       tvhours maritalStatus
   <fct>           <int> <fct>
 1 Never married      12 not_married
 2 Widowed             2 not_married
 3 Never married       4 not_married
 4 Divorced            1 not_married
 5 Never married       3 not_married
 6 Married             0 Married
 7 Married             3 Married
 8 Married              2 Married
 9 Married              1 Married
10 Divorced             1 not_married
# i 11,327 more rows
```