DS5011 Project Description

DS5011 Project: Behavioral Risk Factor Surveillance System Dashboard

Introduction

BRFSS, or the Behavioral Risk Factor Surveillance System, is the world's largest and longest-running telephone survey system for collecting health-related information from adults in the U.S.. It is a collaborative project between the Centers for Disease Control and Prevention (CDC) and all U.S. states, the District of Columbia, and U.S. territories. The survey gathers data on health behaviors, chronic conditions, and preventive care to monitor public health and inform policy.

What BRFSS collects

- **Health-related risk behaviors**: Such as tobacco use, alcohol consumption, and physical activity.
- **Chronic health conditions**: Data on conditions like heart disease, diabetes, and cancer.
- **Preventive care**: Information on access to and use of health services, including vaccinations and screenings.
- **Demographics and other factors**: Includes information on health care access, sleep patterns, and other demographic data.

How BRFSS is conducted

- **Surveys**: Telephone surveys are conducted monthly using landlines and cellular phones.
- **Data collection**: Each state conducts its own survey with the assistance of the CDC, and the data is aggregated at the state level.
- **Population**: The target population is the civilian noninstitutionalized adult population aged 18 and older.

- **Scale**: The survey system completes over 400,000 adult interviews each year across all participating states and territories.

Why BRFSS is important

- **Monitoring health**: It provides state-specific data that helps monitor public health trends and assess progress toward national health objectives.
- **Informing policy**: The data is used to design and evaluate public health programs and policies aimed at improving the health of the population.
- **Research**: It serves as a crucial data source for researchers studying health-related issues across the country.

About the Dataset

The BRFSS data repository is enormous and not available to the public. However, summary statistics are publicly available here in *.csv format at the relatively modest size of 1GB. Please download this summary data into your IDE as tibble `df` to explore it with this markdown file.

The *.csv file has 2763102 rows and 27 columns.

For each question asked in column `Question`, summary statistics include *percentage confidence intervals* for each possible response value in column `Response`. The left (right) endpoint of this interval is in `Confidence_limit_Low` (`Confidence_limit_High`). Most importantly for your project goals, these percentages are computed on grouped data. Grouping is done always using columns `Locationabbr` (state/territory) and `Year`. For this broadest grouping, columns `Break_Out` and `Break_Out_Category` take value "Overall". Further grouping is indicated in those columns, including by: age, household income, sex, race, and highest education level.

Note that the use of `ID` columns in large table provide *keys* to the values in other columns. Key matching is much more efficient than value matching. Here is the key:value list between `BreakOutCategoryID` and `Break_Out_Category`, for example

```
# A tibble: 6 x 3
# Groups:   BreakOutCategoryID [6]
  BreakOutCategoryID Break_Out_Category  count
  <chr>              <chr>               <int>
1 CAT1               Overall            116860
2 CAT2               Sex                221825
3 CAT3               Age Group          621025
4 CAT4               Race/Ethnicity     768868
5 CAT5               Education Attained 439384
6 CAT6               Household Income   595140
```

Survey questions are assigned both a `Topic` and a `Class`, which may help in organizing the questions into layers. For example, here is a possible layering scenario.

```
layerQ <- df |>
    select(Class,Topic,Question) |>
    distinct()
layerQ |>
    group_by(Class,Topic) |>
    summarize(numberQuestions=n()) |>
    print(n=75)
```

```
# A tibble: 66 x 3
# Groups:   Class [21]
   Class                        Topic                    numberQuestions
   <chr>                        <chr>                              <int>
 1 Alcohol Consumption          Alcohol Consumption                    1
 2 Alcohol Consumption          Binge Drinking                         1
 3 Alcohol Consumption          Heavy Drinking                         2
 4 Cholesterol Awareness        Cholesterol Checked                    2
 5 Cholesterol Awareness        Cholesterol High                       1
 6 Chronic Health Indicators    Arthritis                              4
 7 Chronic Health Indicators    Asthma                                 2
 8 Chronic Health Indicators    COPD                                   1
 9 Chronic Health Indicators    Cardiovascular Disease                 4
10 Chronic Health Indicators    Depression                             1
11 Chronic Health Indicators    Diabetes                               1
12 Chronic Health Indicators    Kidney                                 1
13 Chronic Health Indicators    Other Cancer                           1
14 Chronic Health Indicators    Skin Cancer                            1
15 Chronic Health Indicators    Vision                                 1
16 Colorectal Cancer Screening  Blood Stool Test                       4
17 Colorectal Cancer Screening  Colonoscopy                            1
18 Colorectal Cancer Screening  Sigmoidoscopy                          2
19 Colorectal Cancer Screening  USPSTF Recommendations                 2
20 Days of Poor Health          Healthy Days                           2
21 Demographics                 Age                                    1
22 Demographics                 Disability status                      7
23 Demographics                 Education                              1
24 Demographics                 Employment                             1
25 Demographics                 Hearing                                1
26 Demographics                 Income                                 1
27 Demographics                 Internet                               1
```

28	Demographics	Marital Status	1
29	Demographics	Number of Children	1
30	Demographics	Race	1
31	Demographics	Rent/Own Home	1
32	Demographics	Sex	1
33	Demographics	Veteran Status	1
34	E-Cigarette Use	E-Cigarette Use	5
35	Fruits and Vegetables	Fruit Consumption	1
36	Fruits and Vegetables	Vegetable Consumption	1
37	HIV-AIDS	HIV Test	1
38	Health Care Access/Coverage	Health Care Cost	2
39	Health Care Access/Coverage	Health Care Coverage	2
40	Health Care Access/Coverage	Last Checkup	1
41	Health Care Access/Coverage	Personal Care Provider	2
42	Health Care Access/Coverage	Under 65 Coverage	1
43	Health Status	Fair or Poor Health	1
44	Health Status	Overall Health	1
45	Hypertension Awareness	High Blood Pressure	1
46	Immunization	Flu Shot	1
47	Immunization	Pneumonia Vaccination	1
48	Immunization	Shingle Vaccination	1
49	Immunization	Tetanus Shot	1
50	Injury	Drink and Drive	1
51	Injury	Seatbelt Use	2
52	Lung Cancer Screening	Had CAT/CT Chest Scan	1
53	Oral Health	All Teeth Removed	1
54	Oral Health	Dental Visit	1
55	Oral Health	Teeth Removed	1
56	Overweight and Obesity (BMI)	BMI Categories	1
57	Physical Activity	Aerobic Activity	1
58	Physical Activity	Exercise	1
59	Physical Activity	Physical Activity Index	1
60	Physical Activity	Strength Activity	2
61	Prostate Cancer	PSA Test	1
62	Tobacco Use	Current Smoker Status	1
63	Tobacco Use	Smokeless Tobacco	1
64	Tobacco Use	Smoker Status	1
65	Women's Health	Mammogram	3
66	Women's Health	Pap Test	2

Finally, the survey questions themselves have IDs (sometimes several).

```
# A tibble: 228 x 4
```

```
   TopicId ClassId QuestionID Question
   <chr>   <chr>   <chr>      <chr>
 1 TOPIC01 CLASS15 _PAINDX3   Participated in 150 minutes or more of Aerobic Ph~
 2 TOPIC01 CLASS15 _PAINDX2   Participated in 150 minutes or more of Aerobic Ph~
 3 TOPIC01 CLASS15 _PAINDX1   Participated in 150 minutes or more of Aerobic Ph~
 4 TOPIC01 CLASS15 _PAINDEX   Participated in 150 minutes or more of Aerobic Ph~
 5 TOPIC02 CLASS05 AGE        What is your age?
 6 TOPIC03 CLASS01 DRNKANY6   Adults who have had at least one drink of alcohol~
 7 TOPIC03 CLASS01 DRNKANY5   Adults who have had at least one drink of alcohol~
 8 TOPIC04 CLASS13 _ALTETH3   Adults aged 65+ who have had all their natural te~
 9 TOPIC04 CLASS13 _ALTETH2   Adults aged 65+ who have had all their natural te~
10 TOPIC05 CLASS03 _DRDXAR2   Adults who have been told they have arthritis (va~
11 TOPIC05 CLASS03 _DRDXAR3   Adults who have been told they have arthritis (va~
12 TOPIC05 CLASS03 _DRDXAR1   Adults who have been told they have arthritis (va~
13 TOPIC05 CLASS03 _LMTACT3   Are you now limited in any way in any of your usu~
14 TOPIC05 CLASS03 _LMTACT2   Are you now limited in any way in any of your usu~
15 TOPIC05 CLASS03 _LMTACT1   Are you now limited in any way in any of your usu~
16 TOPIC05 CLASS03 _LMTWRK3   Do arthritis or joint symptoms now affect whether~
17 TOPIC05 CLASS03 _LMTWRK2   Do arthritis or joint symptoms now affect whether~
18 TOPIC05 CLASS03 _LMTWRK1   Do arthritis or joint symptoms now affect whether~
19 TOPIC05 CLASS03 _LMTSCL1   Does arthritis or joint symptoms interfere with y~
20 TOPIC06 CLASS03 _CASTHM1   Adults who have been told they currently have ast~
# i 208 more rows
```

The output below shows that there are approximately 100 questions and some of the questions have multiple values of `QuestionID`.

```
# A tibble: 99 x 2
   Question                                                          countQIDs
   <chr>                                                                 <int>
 1 About how long has it been since you last visited a doctor for a r~         1
 2 Adults aged 18-64 who have any kind of health care coverage (varia~         3
 3 Adults aged 45-75 who have had a blood stool test within the past ~         1
 4 Adults aged 50+ who have ever had a sigmoidoscopy or colonoscopy (~         1
 5 Adults aged 50+ who have had a blood stool test within the past tw~         1
 6 Adults aged 50-75 who have had a blood stool test within the past ~         2
 7 Adults aged 65+ who have ever had a pneumonia vaccination (variabl~         2
 8 Adults aged 65+ who have had a flu shot within the past year (vari~         3
 9 Adults aged 65+ who have had all their natural teeth extracted (va~         2
10 Adults that have had any permanent teeth extracted (variable calcu~         2
11 Adults who are current e-cigarette users (variable calculated from~         1
12 Adults who are current e-cigarette users (variable calculated from~         1
```

```
13 Adults who are current smokers (variable calculated from one or mo~         1
14 Adults who are limited in any activities because of physical, ment~         1
15 Adults who had some form of health insurance (variable calculated ~         2
# i 84 more rows
```

Why is there a need for multiple values of `QuestionID` for the same question? Let's investigate this further for the `Question` matching "(Heavy drinkers).*(male).*(14 drinks)". Here is the count of each income breakout in the dataset for this question.

```
# A tibble: 12 x 3
# Groups:   BreakoutID [12]
   BreakoutID Break_Out          countBreakIDs
   <chr>      <chr>                      <int>
 1 INCOME01   Less than $15,000              6
 2 INCOME02   $15,000-$24,999                6
 3 INCOME03   $25,000-$34,999                6
 4 INCOME04   $35,000-$49,999                6
 5 INCOME05   $50,000-$99,999                6
 6 INCOME06   $100,000-$199,999              6
 7 INCOME07   $200,000+                      6
 8 INCOME1    Less than $15,000             12
 9 INCOME2    $15,000-$24,999               12
10 INCOME3    $25,000-$34,999               12
11 INCOME4    $35,000-$49,999               12
12 INCOME5    $50,000+                      12
```

A new income-level breakout was introduced in the last 3 years of the dataset, providing 7 instead of 5 categories. As a result of the new division, a new value of `QuestionID` was assigned. Care must be taken when combining results when a question has multiple values of `QuestionID`.

Project Goal

In this effort you are to produce a dashboard interface to the BRFSS dataset, allowing the user to select the survey question and the dashboard displays the various breakouts of the survey results for the question. Each subpanel of the dashboard should subset the results by only one criteria (age, for example). There should be a subpanel devoted to choosing the question. In addition, for each possible response, a:

- **confidence interval** overall in an **Overall** subpanel
- **confidence interval** versus gender in a **By_Gender** subpanel

- **confidence interval** versus age in a **By_Age_Group** subpanel
- **confidence interval** versus state/territory in a **By_Location** subpanel
- **confidence interval** versus level of education in a **By_Education** subpanel
- **confidence interval** versus income level in a **By_Income** subpanel
- **confidence interval** versus year in a **Temporal** subpanel

Grading metrics and design considerations

Function over fashion. The dashboard does not need to be fancy, and it must function accurately and repeatedly. Emphacize function, not fashion, in your work product. One more time: wrangle Dash Plotly or RShiny last, or not at all, but wrangle the workflow accuracy. Lastly, a pretty and inaccurate dashboard will receive a pretty low score. Most of the credit for this project will be given to accurate aggregation of the data, not the display of the aggregation. Do not overtweek the user interface! Below are some suggestions to consider when deploying your user interface.

Question selection. There are 100 questions, so the selection of a question should be layered. Consider a top-down layering of Class/Topic/Question, for example. Do not present the user with keys, only the value of the keys.

Subpanel Visualization. Please consider this important problem carefully. Subset the results *only* by one variable in each subpanel. For example, subset by year only in one subpanel, geography in another subpanel, age in a third subpanel, etc. This means that all other breakouts must be aggregated! More on this later.

State/territory subpanel. There are more than 50 regions to consider, so use a compact way to represent the geographical results, such as a map, or aggregating states into national regions, or showing highest, lowest and average geographical regions.

User adaptable subpanel complexity. Consider having, for each subpanel, a "more" feature and a "less" feature. This enables the user to adjust the granularity of each subpanel. **And,** provide only a few (5?) levels of granularity for each subpanel. Otherwise, you risk underwhelming or overwhelming your audience.

Proposed workflow

- **Consider a parquet-formatted dataset, based on observed speedup of information retrieval**.
- **Design the question selection tool**.
- **Design the aggregation workflow for each subpanel, beginning with the Overall subpanel**.
- **Design the Overall subpanel without "more" or "less" options**.

- **Augment the Overall subpanel with "more" or "less" options.**
- **Design the other subpanels without "more" or "less" options.**
- **Modify these subpanels to include "more" or "less" options.**

Aggregation Strategy for "Overall" Breakout

Most of the questions have a breakout group called "Overall". As the name suggests, this is the broadest aggregation so far in the dataset. However, percentages labeled "Overall" are still per-territory, per-year summary metrics and not true overall results! Here, we consider a method to aggregate and produce confidence limits for response percentages.

Note that the sample size is provided in column `Sample_Size` along with a percentage estimate in column `Data_value`. The percentage estimate and the sample size allow you to translate per-breakout percentage estimates to per-breakout person estimates prior to aggregation. Upon aggregation, the aggregate sample size can be used to determine an aggregate percentage confidence interval. If the possible responses to a questions do not vary over time or territory, then aggregation is straightforward. Let's consider that case first.

The chunk below illustrates how one could aggregate the per-territory per-year "Overall" value for `Break_Out` ("BO1" value for `BreakOutID`). However, if we "Run Cell" repeatedly, we see some problems with our aggregation. Try it!

```
    save_rows <- c(2150434,830046,131103)
    qidx<- sample(save_rows,size=1)
df |>
    select(Question,Break_Out,Response,ResponseID,
    Sample_Size,Data_value) |>
    filter(Question==Question[qidx]) |>
    filter(Break_Out=="Overall") |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    mutate(Sample_Size=persons*100/Data_value) |>
    group_by(ResponseID,Response) |>
    summarize(agg_ss=sum(Sample_Size),
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
        print(n=20)
```

```
# A tibble: 11 x 8
# Groups:   ResponseID [11]
   ResponseID Response         agg_ss agg_persons agg_percent agg_percent_sdev
   <chr>      <chr>             <dbl>       <dbl>       <dbl>             <dbl>
 1 RESP025    Emplyd          829310.      403874       48.7           0.0549
 2 RESP026    Self emplyd     952084.       79307        8.33          0.0283
 3 RESP027    No work< yr    4345159.      137203        3.16          0.00839
 4 RESP028    No work >yr    5004899.      138615        2.77          0.00734
 5 RESP029    Homemkr        1025198.       67172        6.55          0.0244
 6 RESP030    Student        2854783.      150163        5.26          0.0132
 7 RESP031    Retired        9426205.     1750896       18.6           0.0127
 8 RESP032    Unable to work 6332814.      411954        6.51          0.00980
 9 RESP137    Employed       4116750.     2015613       49.0           0.0246
10 RESP141    Homemaker      4500671.      251868        5.60          0.0108
11 RESP172    Self-employed  4674301.      424403        9.08          0.0133
# i 2 more variables: agg_low_ci_limit <dbl>, agg_high_ci_limit <dbl>
```

By a few trials of the above chunk, it may be seen that the response options changed over the course of this survey for some questions. Often, the changes are trivial, for example, "Emplyd" vs. "Employed", or "Homemkr" vs "Homemaker". In other cases, the response options have different refinement, for example, "$50,000+", vs the finer options "$50,000-99,999","$100,000-199,999", and "$200,000+". In all these cases, note that the value of `ResponseID` changes when the response option changes.

In order to aggregate properly, we need to merge these responses. Without this merge, "Homemkr" and "Homemaker" responses will not be pooled. Fortunately, this may be done by a simple mapping of every value of `ResponseID` to a new one, reflecting the merge. I have drafted merge rules for responseID values in the function below. Please start with that function and modify it as necessary.

```
merge_ResponseID <- function(char_vec) {
    char_vec <- str_replace(char_vec,"RESP025","RESP137")
    char_vec <- str_replace(char_vec,"RESP026","RESP172")
    char_vec <- str_replace(char_vec,"RESP029","RESP141")
    char_vec <- str_replace(char_vec,"RESP230","RESP020")
    char_vec <- str_replace(char_vec,"RESP231","RESP020")
    char_vec <- str_replace(char_vec,"RESP232","RESP020")
    char_vec <- str_replace(char_vec,"RESP196","RESP199")
    char_vec <- str_replace(char_vec,"RESP197","RESP199")
    char_vec <- str_replace(char_vec,"RESP198","RESP199")
    char_vec <- str_replace(char_vec,"RESP199","RESP199")
    char_vec <- str_replace(char_vec,"RESP200","RESP008")
    char_vec <- str_replace(char_vec,"RESP194","RESP005")
```

```
    char_vec <- str_replace(char_vec,"RESP195","RESP006")
    return(char_vec)
}
```

After the `ResponseID` column is merged, the following function changes the `Response` column. Both functions should be updated in tandem.

```
merge_Response <- function(ResponseID,Response) {
idx <- str_detect(ResponseID,"RESP137")
Response[idx] <- "Employed"
idx <- str_detect(ResponseID,"RESP172")
Response[idx] <- "Self-employed"
idx <- str_detect(ResponseID,"RESP141")
Response[idx] <- "Homemaker"
idx <- str_detect(ResponseID,"RESP020")
Response[idx] <- "$50,000+"
idx <- str_detect(ResponseID,"RESP199")
Response[idx] <- "A/A Native, Asian,Other"
idx <- str_detect(ResponseID,"RESP008")
Response[idx] <- "Multiracial"
idx <- str_detect(ResponseID,"RESP005")
Response[idx] <- "White"
idx <- str_detect(ResponseID,"RESP006")
Response[idx] <- "Black"
# make all responses lower case
Response <- tolower(Response)
return(Response)
}
```

One of your tasks will be to map each one of these values of `ResponseID` to a (possible) new one, reflecting how the responses are merged, starting with the draft functions above. There will only be a small number of changes in this process, as most of the response options do not need to be merged.

Here the draft functions are applied to the same questions. Are there improvements?

```
    save_rows <- c(2150434,830046,131103)
    qidx<- sample(save_rows,size=1)
lDf <- df |>
    select(Question,Break_Out,Response,ResponseID,
    Sample_Size,Data_value) |>
    filter(Question==Question[qidx]) |>
```

```
    filter(Break_Out=="Overall")

    lDf$ResponseID <- merge_ResponseID(lDf$ResponseID)
    lDf$Response <- merge_Response(unlist(lDf$ResponseID),
    unlist(lDf$Response))

    lDf |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    mutate(Sample_Size=persons*100/Data_value) |>
    group_by(ResponseID,Response) |>
    summarize(agg_ss=sum(Sample_Size),
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
        print(n=20)
```

```
# A tibble: 6 x 8
# Groups:   ResponseID [6]
  ResponseID Response      agg_ss agg_persons agg_percent agg_percent_sdev
  <chr>      <chr>          <dbl>       <dbl>       <dbl>            <dbl>
1 RESP010    18-24 years 2646706.      333789        12.6           0.0204
2 RESP011    25-34 years 3556121.      607216        17.1           0.0200
3 RESP012    35-44 years 4456831.      727847        16.3           0.0175
4 RESP013    45-54 years 5686151.      945513        16.6           0.0156
5 RESP014    55-64 years 7465206.     1236977        16.6           0.0136
6 RESP015    65+ years   9930209.     2048338        20.6           0.0128
# i 2 more variables: agg_low_ci_limit <dbl>, agg_high_ci_limit <dbl>
```

General Aggregation Approach

To aggregate "Overall" breakouts, it is important to merge the response options, as shown above. Aggregating of "Overall" breakouts is relatively straightforward, however, because there can be no change in the **breakout categories** over year or territory. "Overall" doesn't change throughout this survey. This is not the case for other breakout categories, however. Let's see an example of a question, whose responses are merged, below. Note the difference in the number of breakouts in the "Household Income" breakout category.

```r
#my_Question <- "What is your race/ethnicity?"
#my_Question <- df$Question[sample(dim(df)[1],1)]
my_Question <- "Ever told you that you have a form of depression?"
lDf <- df |>
    filter(Question==my_Question) |>
    select(Question,QuestionID,Response,
   ResponseID,Break_Out, Break_Out_Category)

lDf$ResponseID <- merge_ResponseID(lDf$ResponseID)
    lDf$Response <- merge_Response(unlist(lDf$ResponseID),
    unlist(lDf$Response))

    lDf |>
    filter(Response=="yes") |>
    select(Question,QuestionID,Response,
   Break_Out, Break_Out_Category) |>
    distinct() |>
    group_by(Question,QuestionID,Break_Out_Category) |>
    summarize(countBreakOuts=n()) |>
    arrange(Break_Out_Category)
```

```
# A tibble: 12 x 4
# Groups:   Question, QuestionID [2]
   Question                          QuestionID Break_Out_Category countBreakOuts
   <chr>                             <chr>      <chr>                       <int>
 1 Ever told you that you have a f~ ADDEPEV2   Age Group                       6
 2 Ever told you that you have a f~ ADDEPEV3   Age Group                       6
 3 Ever told you that you have a f~ ADDEPEV2   Education Attained              4
 4 Ever told you that you have a f~ ADDEPEV3   Education Attained              4
 5 Ever told you that you have a f~ ADDEPEV2   Household Income                5
 6 Ever told you that you have a f~ ADDEPEV3   Household Income                8
 7 Ever told you that you have a f~ ADDEPEV2   Overall                         1
 8 Ever told you that you have a f~ ADDEPEV3   Overall                         1
 9 Ever told you that you have a f~ ADDEPEV2   Race/Ethnicity                  8
10 Ever told you that you have a f~ ADDEPEV3   Race/Ethnicity                  8
11 Ever told you that you have a f~ ADDEPEV2   Sex                             2
12 Ever told you that you have a f~ ADDEPEV3   Sex                             2
```

To get started, here is a list of the breakouts, with `BreakoutID` and `BreakOutCategoryID` included.

```
# A tibble: 58 x 4
```

```
# Groups:   BreakOutCategoryID, BreakoutID [58]
   BreakOutCategoryID BreakoutID Break_Out                                  count
   <chr>              <chr>      <chr>                                      <int>
 1 CAT1               BO1        Overall                                   116860
 2 CAT2               SEX1       Male                                      110490
 3 CAT2               SEX2       Female                                    111335
 4 CAT3               AGE01      18-24                                      99515
 5 CAT3               AGE02      25-34                                      99773
 6 CAT3               AGE03      35-44                                      99848
 7 CAT3               AGE04      45-54                                     100186
 8 CAT3               AGE05      55-64                                     100164
 9 CAT3               AGE06      40-49                                       1172
10 CAT3               AGE07      50-59                                       4262
11 CAT3               AGE08      60-64                                       2020
12 CAT3               AGE09      65+                                       100435
13 CAT3               AGE10      65-74                                       3396
14 CAT3               AGE11      75+                                         3396
15 CAT3               AGE12      21-30                                        304
16 CAT3               AGE13      31-40                                        304
17 CAT3               AGE14      41-50                                        304
18 CAT3               AGE15      51-60                                        304
19 CAT3               AGE16      61-65                                        304
20 CAT3               AGE17      60-69                                       2242
21 CAT3               AGE18      70-74                                        534
22 CAT3               AGE19      70-75                                       1600
23 CAT3               AGE20      21-25                                        106
24 CAT3               AGE21      26-35                                        106
25 CAT3               AGE22      36-45                                        106
26 CAT3               AGE23      46-55                                        106
27 CAT3               AGE24      56-65                                        106
28 CAT3               AGE25      65-75                                        324
29 CAT3               AGE26      70-80                                        108
30 CAT4               RACE01     White, non-Hispanic                        77207
31 CAT4               RACE02     Black, non-Hispanic                        77207
32 CAT4               RACE03     American Indian or Alaskan Native, non-~   77207
33 CAT4               RACE04     Asian, non-Hispanic                        77207
34 CAT4               RACE05     Native Hawaiian or other Pacific Island~   77207
35 CAT4               RACE06     Other, non-Hispanic                        77207
36 CAT4               RACE07     Multiracial, non-Hispanic                  77207
37 CAT4               RACE08     Hispanic                                   77207
38 CAT4               RACE1      White, non-Hispanic                        30504
39 CAT4               RACE2      Black, non-Hispanic                        29950
40 CAT4               RACE3      Hispanic                                   30366
```

```
41 CAT4              RACE4      Other, non-Hispanic                30364
42 CAT4              RACE5      Multiracial, non-Hispanic          30028
43 CAT5              EDUCA1     Less than H.S.                    109834
44 CAT5              EDUCA2     H.S. or G.E.D.                    109850
45 CAT5              EDUCA3     Some post-H.S.                    109848
46 CAT5              EDUCA4     College graduate                  109852
47 CAT6              INCOME01   Less than $15,000                  25494
48 CAT6              INCOME02   $15,000-$24,999                    25494
49 CAT6              INCOME03   $25,000-$34,999                    25494
50 CAT6              INCOME04   $35,000-$49,999                    25494
51 CAT6              INCOME05   $50,000-$99,999                    25494
52 CAT6              INCOME06   $100,000-$199,999                  25494
53 CAT6              INCOME07   $200,000+                          25494
54 CAT6              INCOME1    Less than $15,000                  83342
55 CAT6              INCOME2    $15,000-$24,999                    83346
56 CAT6              INCOME3    $25,000-$34,999                    83329
57 CAT6              INCOME4    $35,000-$49,999                    83323
58 CAT6              INCOME5    $50,000+                           83342
```

Let's begin with "CAT6", or income breakout. Clearly, "INCOME05", "INCOME06", and "INCOME07" are a refinement of "INCOME5". For "CAT4", or race, we also see a refinement, ("RACE03","RACE04","RACE05", and "RACE06" being a refinement of "RACE4"). Finally, for "CAT3", or age, we see two breakout groups (AGE01 through AGE11, and AGE12 through AGE26). Some of these values of `BreakoutID` have full or partial overlap, while others reflect a refinement.

One possible approach to merging the breakout categories mimics that for response options: first change the `BreakoutID` values, then change the `Break_Out` values. The following are draft functions which attempt this.

```
merge_BreakoutID <- function(char_vec) {
    char_vec <- str_replace(char_vec,"INCOME01","INCOME1")
    char_vec <- str_replace(char_vec,"INCOME02","INCOME2")
    char_vec <- str_replace(char_vec,"INCOME03","INCOME3")
    char_vec <- str_replace(char_vec,"INCOME04","INCOME4")
    char_vec <- str_replace(char_vec,"INCOME05","INCOME5")
    char_vec <- str_replace(char_vec,"INCOME06","INCOME5")
    char_vec <- str_replace(char_vec,"INCOME07","INCOME5")
    char_vec <- str_replace(char_vec,"RACE01","RACE1")
    char_vec <- str_replace(char_vec,"RACE02","RACE2")
    char_vec <- str_replace(char_vec,"RACE08","RACE3")
    char_vec <- str_replace(char_vec,"RACE04","RACE4")
    char_vec <- str_replace(char_vec,"RACE05","RACE4")
```

```
    char_vec <- str_replace(char_vec,"RACE06","RACE4")
    char_vec <- str_replace(char_vec,"RACE03","RACE4")
    char_vec <- str_replace(char_vec,"RACE07","RACE5")
return(char_vec)
}
```

```
merge_Break_Out <- function(BreakoutID,Break_Out) {
idx <- str_detect(BreakoutID,"INCOME5")
Break_Out[idx] <- "$50,000+"
idx <- str_detect(BreakoutID,"RACE1")
Break_Out[idx] <- "White"
idx <- str_detect(BreakoutID,"RACE2")
Break_Out[idx] <- "Black"
idx <- str_detect(BreakoutID,"RACE3")
Break_Out[idx] <- "Hispanic"
idx <- str_detect(BreakoutID,"RACE4")
Break_Out[idx] <- "A/A Native, Asian,Other"
idx <- str_detect(BreakoutID,"RACE5")
Break_Out[idx] <- "Multiracial"
return(Break_Out)
}
```

Let's apply these functions to merge values in `BreakoutID` and `Break_Out` for the same question.

```
#my_Question <- "What is your race/ethnicity?"
#my_Question <- df$Question[sample(dim(df)[1],1)]
my_Question <- "Ever told you that you have a form of depression?"
lDf <- df |>
    filter(Question==my_Question) |>
    select(Question,QuestionID,Response,
   ResponseID,Break_Out, BreakoutID,Break_Out_Category)

lDf$ResponseID <- merge_ResponseID(lDf$ResponseID)
    lDf$Response <- merge_Response(unlist(lDf$ResponseID),
    unlist(lDf$Response))
lDf$BreakoutID <- merge_BreakoutID(lDf$BreakoutID)
    lDf$Break_Out <- merge_Break_Out(unlist(lDf$BreakoutID),
    unlist(lDf$Break_Out))

lDf |>
    filter(Response=="yes") |>
```

```
    select(Question,QuestionID,Response,
   Break_Out, Break_Out_Category) |>
    distinct() |>
    group_by(Question,QuestionID,Break_Out_Category) |>
    summarize(countBreakOuts=n()) |>
    arrange(Break_Out_Category)
```

```
# A tibble: 12 x 4
# Groups:   Question, QuestionID [2]
   Question                           QuestionID Break_Out_Category countBreakOuts
   <chr>                              <chr>      <chr>                       <int>
 1 Ever told you that you have a f~ ADDEPEV2   Age Group                       6
 2 Ever told you that you have a f~ ADDEPEV3   Age Group                       6
 3 Ever told you that you have a f~ ADDEPEV2   Education Attained              4
 4 Ever told you that you have a f~ ADDEPEV3   Education Attained              4
 5 Ever told you that you have a f~ ADDEPEV2   Household Income                5
 6 Ever told you that you have a f~ ADDEPEV3   Household Income                5
 7 Ever told you that you have a f~ ADDEPEV2   Overall                         1
 8 Ever told you that you have a f~ ADDEPEV3   Overall                         1
 9 Ever told you that you have a f~ ADDEPEV2   Race/Ethnicity                  5
10 Ever told you that you have a f~ ADDEPEV3   Race/Ethnicity                  5
11 Ever told you that you have a f~ ADDEPEV2   Sex                             2
12 Ever told you that you have a f~ ADDEPEV3   Sex                             2
```

As with response options, a complete aggregation will require that a mapping of BreakoutID values, reflected above, to a merged BreakoutID. **One of your responsibilities is to merge values of BreakoutID, using your best judgement**. The draft functions above serve as starting points. Such a mapping will allow aggregation of responses to the same questions across different versions of breakout category.

Pre-Beta Prototype Dashboard

In this section we consider a workflow, from choosing a question, to summarizing the results.

```
# prototype_workflow_choose_question
# interactive ONLY - prompt user for class
#option <- unique(layerQ$Class)
#idx <- utils::menu(option,title="Which class?")
#my_class <- option[idx]
#slayerQ <- layerQ |>
```

```
#    filter(Class==my_class)
## prompt user for topic
#option <- unique(slayerQ$Topic)
#idx <- utils::menu(option,title="Which topic?")
#my_topic <- option[idx]
#vslayerQ <- slayerQ |>
#    filter(Topic==my_topic)
## prompt user for question
#option <- unique(vslayerQ$Question)
#idx <- utils::menu(option,title="Which question?")
#my_q <- option[idx]
# from RDS file - question stored there
my_q <- readRDS("my_q.RDS")
```

```
qDf <- df |>
    filter(Question==my_q) |>
    filter(!(Locationabbr == "US")) |>
    filter(!(Locationabbr == "UW"))
# merges
qDf$ResponseID <- merge_ResponseID(qDf$ResponseID)
    qDf$Response <- merge_Response(unlist(qDf$ResponseID),
    unlist(qDf$Response))
qDf$BreakoutID <- merge_BreakoutID(qDf$BreakoutID)
    qDf$Break_Out <- merge_Break_Out(unlist(qDf$BreakoutID),
    unlist(qDf$Break_Out))
```

```
# comment out if next chunk is used
(my_q <- str_extract(my_q,"[^(]+"))
```

```
[1] "Days when physical health status not good "
```

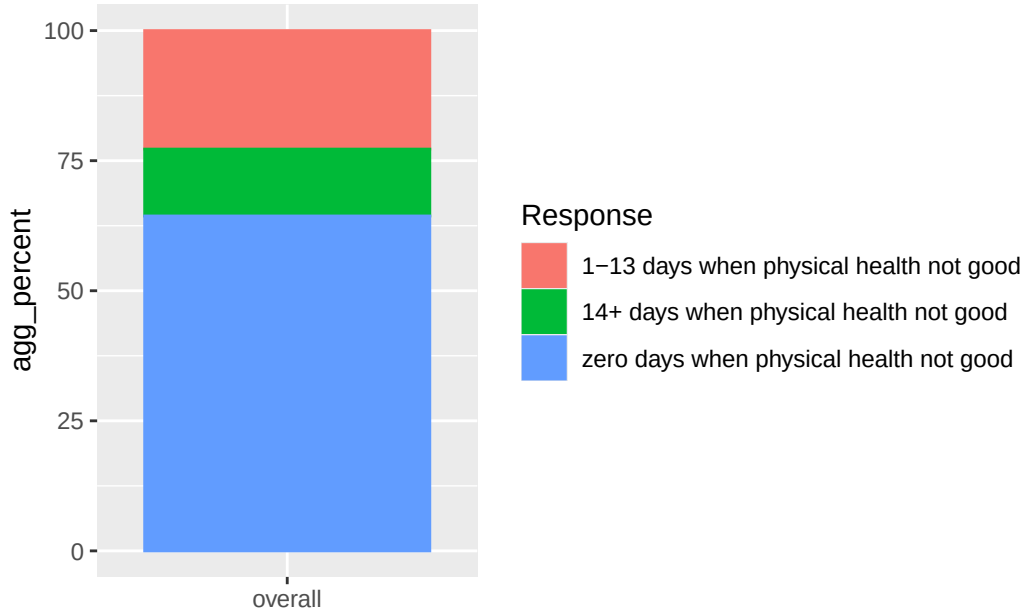
```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT1") |>
    select(Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Response) |>
    summarize(agg_persons=sum(persons))

plotDf <- plotDf |>
    mutate(agg_ss= sum(agg_persons)) |>
```

```
    mutate(agg_percent=agg_persons*100/agg_ss) |>
    mutate( agg_percent_sdev=sqrt(
        agg_percent*(100-agg_percent)/agg_ss)) |>
    mutate(agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev) |>
    mutate(agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
    select(-c(agg_persons,agg_ss,agg_percent_sdev))
plotDf |>
        select(-agg_percent) |>
        print(n=20)
```

```
# A tibble: 3 x 3
  Response                                  agg_low_ci_limit agg_high_ci_limit
  <chr>                                                <dbl>              <dbl>
1 1-13 days when physical health not good               22.7               22.8
2 14+ days when physical health not good                12.8               12.9
3 zero days when physical health not good               64.3               64.4
```

```
    plotDf |>
        ggplot( aes(x="overall",y=agg_percent,
    fill=Response,color=Response,position="fill")) +
         geom_col() + xlab(my_q)
```



Days when physical health status not good

```
## summary by overall
#(my_q <- str_extract(my_q,"[^(]+"))
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT1") |>
#    select(Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons/Data_value*100)|>
##ssSum <- plotDf |>
##    summarize(sum(persons))
##ssSum <- unlist(ssSum)
##numResponseOptions <- length(unique(plotDf$Response))
##ssVec <- rep(ssSum,each=numResponseOptions)
##plotDf |>
#    group_by(Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#    select(-c(agg_persons,agg_percent_sdev))
#
```

```
#    plotDf |>
#        select(-agg_percent) |>
#        print(n=20)
#    plotDf |>
#        ggplot( aes(x="overall",y=agg_percent,
#    fill=Response,color=Response,position="fill")) +
#         geom_col() + xlab(my_q)
```

```
# comment out if next chunk is used
# summary by territory
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT1") |>
    select(Locationabbr,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Locationabbr) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf <- plotDf |>
    group_by(Locationabbr,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
```

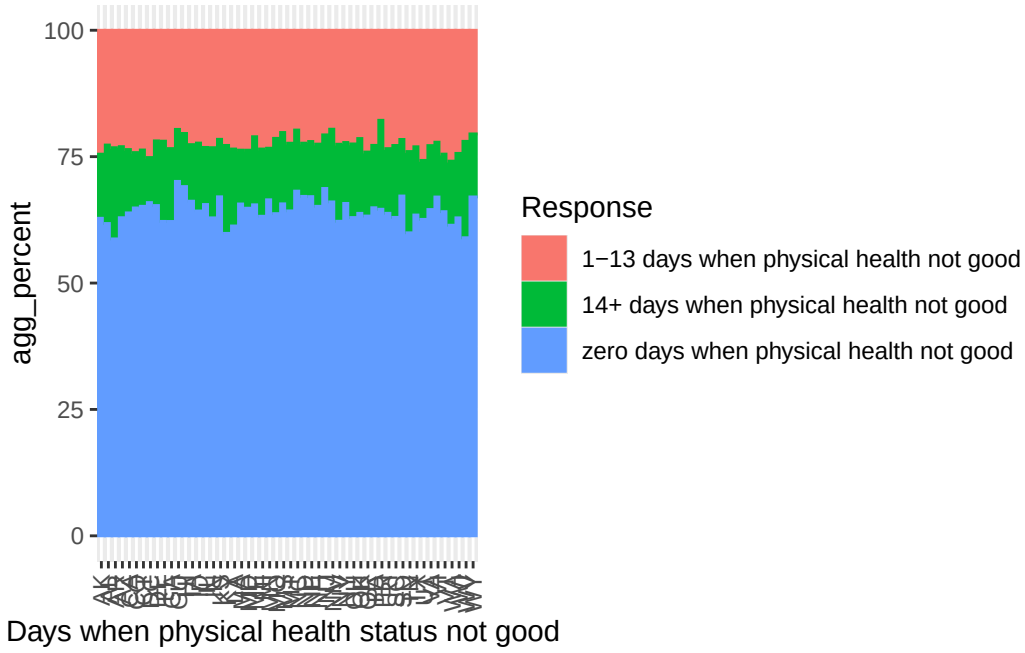
```
    agg_percent + 2*agg_percent_sdev) |>
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=20)
```

```
# A tibble: 162 x 4
# Groups:   Locationabbr, Response [162]
   Locationabbr Response                       agg_low_ci_limit agg_high_ci_limit
   <chr>        <chr>                                     <dbl>              <dbl>
 1 AK           1-13 days when physical heal~              23.9               25.1
 2 AK           14+ days when physical healt~              12.2               13.1
 3 AK           zero days when physical heal~              62.2               63.5
 4 AL           1-13 days when physical heal~              22.2               23.2
 5 AL           14+ days when physical healt~              15.1               16.0
 6 AL           zero days when physical heal~              61.2               62.4
 7 AR           1-13 days when physical heal~              22.7               23.8
 8 AR           14+ days when physical healt~              17.6               18.5
 9 AR           zero days when physical heal~              58.1               59.3
10 AZ           1-13 days when physical heal~              22.6               23.4
11 AZ           14+ days when physical healt~              13.7               14.4
12 AZ           zero days when physical heal~              62.5               63.4
13 CA           1-13 days when physical heal~              23.2               24.0
14 CA           14+ days when physical healt~              12.2               12.8
15 CA           zero days when physical heal~              63.4               64.3
16 CO           1-13 days when physical heal~              23.8               24.6
17 CO           14+ days when physical healt~              10.7               11.2
18 CO           zero days when physical heal~              64.4               65.3
19 CT           1-13 days when physical heal~              23.3               24.1
20 CT           14+ days when physical healt~              10.8               11.4
# i 142 more rows
```

```
plotDf |>
    ggplot( aes(x=Locationabbr,
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
## summary by territory
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT1") |>
#    select(Locationabbr,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Locationabbr,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
```

```
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
#    plotDf |>
#        select(-agg_percent) |>
#        print(n=10)
#    plotDf |>
#        ggplot( aes(x=Locationabbr,
#        y=agg_percent,
#        fill=Response,color=Response,position="fill")) +
#             geom_col() + xlab(my_q) +
#             theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
#(my_q)
```

```
# comment out if next chunk is used
# summary by year
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT1") |>
    select(Year,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Year) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf <- plotDf |>
    group_by(Year,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
```

```
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=100)
```

```
# A tibble: 15 x 4
# Groups:   Year, Response [15]
    Year Response                                agg_low_ci_limit agg_high_ci_limit
   <dbl> <chr>                                              <dbl>              <dbl>
 1  2019 1-13 days when physical health not ~              23.8               24.0
 2  2019 14+ days when physical health not g~              14.1               14.3
 3  2019 zero days when physical health not ~              61.8               62.1
 4  2020 1-13 days when physical health not ~              18.1               18.3
 5  2020 14+ days when physical health not g~              10.8               11.0
 6  2020 zero days when physical health not ~              70.7               71.0
 7  2021 1-13 days when physical health not ~              20.9               21.2
 8  2021 14+ days when physical health not g~              11.8               12.0
 9  2021 zero days when physical health not ~              66.9               67.2
10  2022 1-13 days when physical health not ~              24.8               25.1
11  2022 14+ days when physical health not g~              13.3               13.5
12  2022 zero days when physical health not ~              61.5               61.8
13  2023 1-13 days when physical health not ~              25.3               25.6
14  2023 14+ days when physical health not g~              13.7               13.9
15  2023 zero days when physical health not ~              60.6               60.9
```

```
plotDf |>
    ggplot( aes(x=Year,
```

```
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

Days when physical health status not good

```
##summary by year
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT1") |>
#    select(Year,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Year,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
#    plotDf |>
```

```
#        select(-agg_percent) |>
#        print(n=20)
#    plotDf |>
#        ggplot( aes(x=Year,y=agg_percent,
#        fill=Response,color=Response,position="fill")) +
#             geom_col() + xlab(my_q)
```

```
# comment out if next chunk is used
# summary by gender
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT2") |>
    select(Break_Out,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Break_Out) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf <- plotDf |>
    group_by(Break_Out,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
```

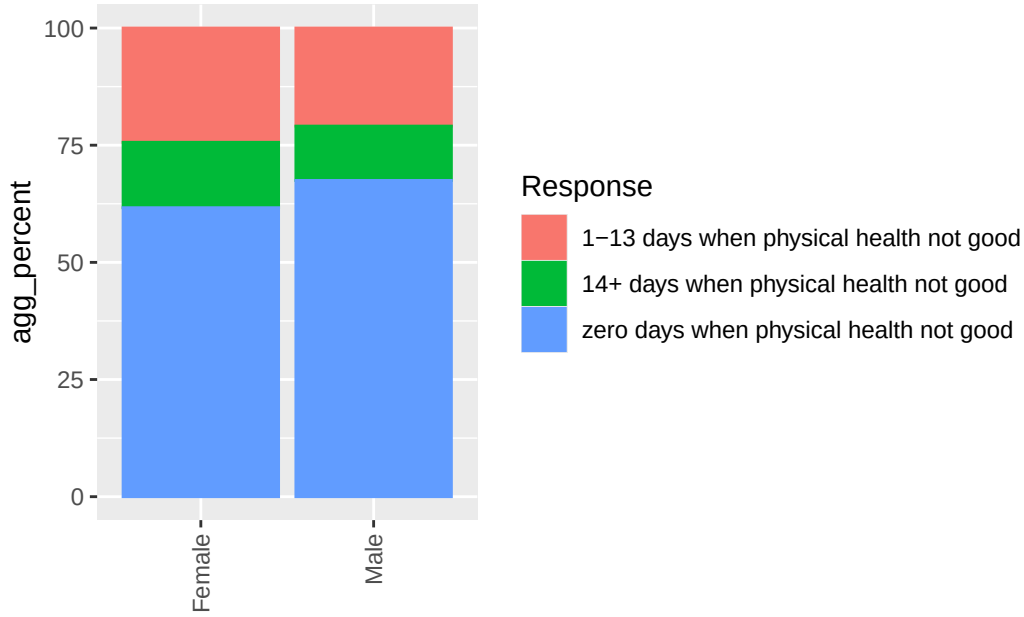
```
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=100)
```

```
# A tibble: 6 x 4
# Groups:   Break_Out, Response [6]
  Break_Out Response                              agg_low_ci_limit agg_high_ci_limit
  <chr>     <chr>                                            <dbl>              <dbl>
1 Female    1-13 days when physical health n~                 24.3               24.5
2 Female    14+ days when physical health no~                 13.9               14.0
3 Female    zero days when physical health n~                 61.6               61.8
4 Male      1-13 days when physical health n~                 20.8               21.0
5 Male      14+ days when physical health no~                 11.5               11.7
6 Male      zero days when physical health n~                 67.4               67.6
```

```
plotDf |>
    ggplot( aes(x=Break_Out,
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Days when physical health status not good

```
## summary by gender
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT2") |>
#    select(Break_Out,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Break_Out,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
#    plotDf |>
#        select(-agg_percent) |>
#        print(n=20)
#
#    plotDf |>
#        ggplot( aes(x=Break_Out,
#        y=agg_percent,
```

```
#        fill=Response,color=Response,position="fill")) +
#             geom_col() + xlab(my_q)
```

```
# comment out if next chunk is used
# summary by age
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT3") |>
    select(Break_Out,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Break_Out) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

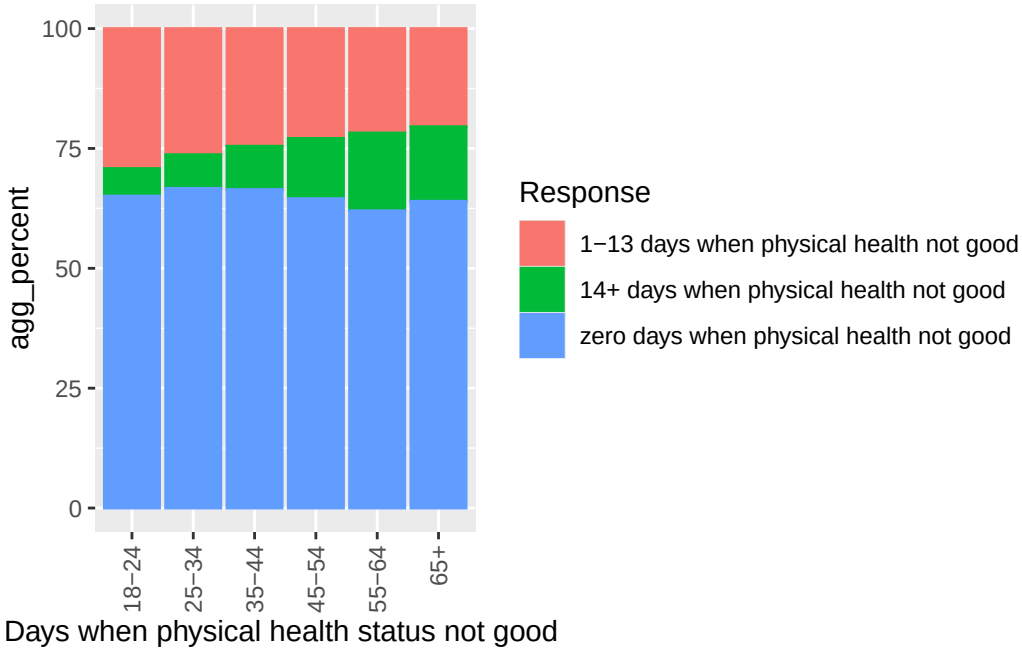
```
plotDf <- plotDf |>
    group_by(Break_Out,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=100)
```

```
# A tibble: 18 x 4
# Groups:   Break_Out, Response [18]
   Break_Out Response                          agg_low_ci_limit agg_high_ci_limit
   <chr>     <chr>                                        <dbl>             <dbl>
 1 18-24     1-13 days when physical health ~            28.9              29.5
 2 18-24     14+ days when physical health n~             5.62              5.88
 3 18-24     zero days when physical health ~            64.8              65.3
 4 25-34     1-13 days when physical health ~            26.1              26.5
 5 25-34     14+ days when physical health n~             6.92              7.13
 6 25-34     zero days when physical health ~            66.5              66.9
 7 35-44     1-13 days when physical health ~            24.4              24.7
 8 35-44     14+ days when physical health n~             8.91              9.13
 9 35-44     zero days when physical health ~            66.3              66.6
10 45-54     1-13 days when physical health ~            22.8              23.1
11 45-54     14+ days when physical health n~            12.4              12.7
12 45-54     zero days when physical health ~            64.4              64.7
13 55-64     1-13 days when physical health ~            21.6              21.9
14 55-64     14+ days when physical health n~            16.1              16.3
15 55-64     zero days when physical health ~            61.9              62.2
16 65+       1-13 days when physical health ~            20.4              20.6
17 65+       14+ days when physical health n~            15.5              15.6
18 65+       zero days when physical health ~            63.9              64.1
```

```
plotDf |>
    ggplot( aes(x=Break_Out,
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
## summary by age group
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT3") |>
#    select(Break_Out,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Break_Out,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
# plotDf |>
#    select(-agg_percent) |>
#    print(n=20)
# plotDf |>
#    ggplot( aes(x=Break_Out,
#    y=agg_percent,
#    fill=Response,color=Response,position="fill")) +
```

```
#         geom_col() + xlab(my_q)
```

```
# comment out if next chunk is used
# summary by race
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT4") |>
    select(Break_Out,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Break_Out) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf <- plotDf |>
    group_by(Break_Out,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

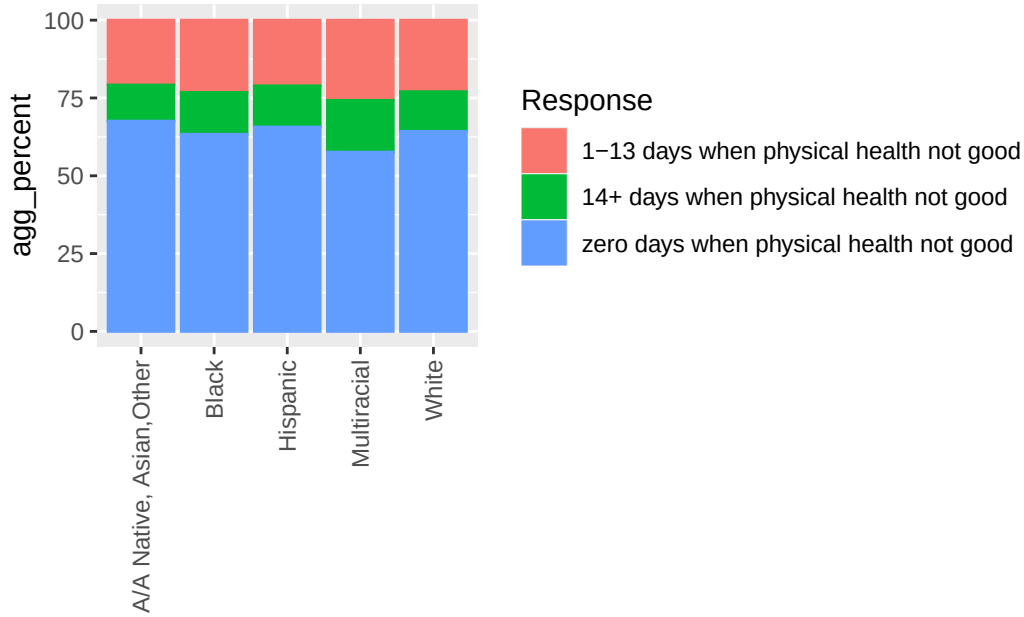
```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
```

```
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=100)
```

```
# A tibble: 15 x 4
# Groups:   Break_Out, Response [15]
   Break_Out                Response           agg_low_ci_limit agg_high_ci_limit
   <chr>                    <chr>                         <dbl>              <dbl>
 1 A/A Native, Asian,Other 1-13 days when ph~             20.6               21.1
 2 A/A Native, Asian,Other 14+ days when phy~             11.4               11.8
 3 A/A Native, Asian,Other zero days when ph~             67.3               67.8
 4 Black                   1-13 days when ph~             23.0               23.4
 5 Black                   14+ days when phy~             13.3               13.6
 6 Black                   zero days when ph~             63.1               63.6
 7 Hispanic                1-13 days when ph~             20.9               21.3
 8 Hispanic                14+ days when phy~             13.1               13.4
 9 Hispanic                zero days when ph~             65.4               65.9
10 Multiracial             1-13 days when ph~             25.4               26.2
11 Multiracial             14+ days when phy~             16.3               17.0
12 Multiracial             zero days when ph~             57.1               58.1
13 White                   1-13 days when ph~             22.9               23.1
14 White                   14+ days when phy~             12.7               12.8
15 White                   zero days when ph~             64.2               64.3
```

```
plotDf |>
    ggplot( aes(x=Break_Out,
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Days when physical health status not good

```
## summary by race
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT4") |>
#    select(Break_Out,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Break_Out,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
#plotDf |>
#    select(-agg_percent) |>
#    print(n=20)
#
#plotDf |>
#    ggplot( aes(x=Break_Out,y=agg_percent,
#    fill=Response,color=Response,position="fill")) +
```

```
#         geom_col() + xlab(my_q)
```

```
# comment out if next chunk is used
# summary by education level
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT5") |>
    select(Break_Out,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Break_Out) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf <- plotDf |>
    group_by(Break_Out,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

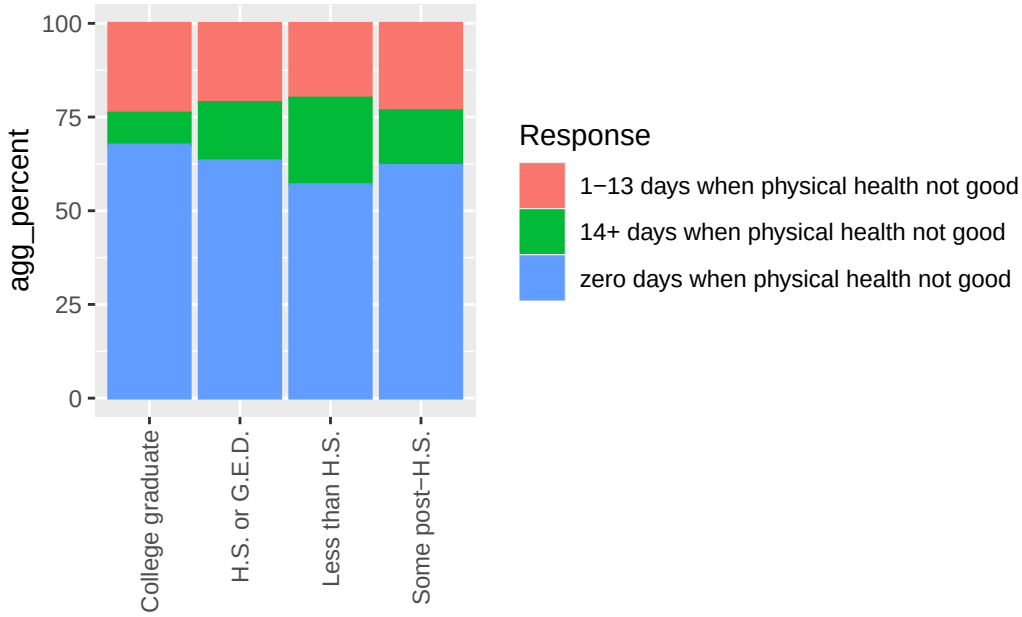
```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
```

```
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=100)
```

```
# A tibble: 12 x 4
# Groups:   Break_Out, Response [12]
   Break_Out        Response                  agg_low_ci_limit agg_high_ci_limit
   <chr>            <chr>                                <dbl>              <dbl>
 1 College graduate 1-13 days when physical ~            23.8               24.0
 2 College graduate 14+ days when physical h~             8.44               8.56
 3 College graduate zero days when physical ~            67.5               67.7
 4 H.S. or G.E.D.   1-13 days when physical ~            21.0               21.2
 5 H.S. or G.E.D.   14+ days when physical h~            15.5               15.7
 6 H.S. or G.E.D.   zero days when physical ~            63.2               63.5
 7 Less than H.S.   1-13 days when physical ~            19.7               20.1
 8 Less than H.S.   14+ days when physical h~            22.9               23.4
 9 Less than H.S.   zero days when physical ~            56.7               57.2
10 Some post-H.S.   1-13 days when physical ~            23.2               23.4
11 Some post-H.S.   14+ days when physical h~            14.5               14.7
12 Some post-H.S.   zero days when physical ~            62.0               62.2
```

```
plotDf |>
    ggplot( aes(x=Break_Out,
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
## summary by education level
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT5") |>
#    select(Break_Out,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Break_Out,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
#plotDf |>
#    select(-agg_percent) |>
#    print(n=20)
#plotDf |>
#    ggplot( aes(x=Break_Out,
#    y=agg_percent,
#    fill=Response,color=Response,position="fill")) +
```

```
#         geom_col() + xlab(my_q)
```

```
# comment out if next chunk is used
# summary by income
(my_q)
```

```
[1] "Days when physical health status not good "
```

```
plotDf <- qDf |>
    filter(BreakOutCategoryID=="CAT6") |>
    select(Break_Out,Response,Sample_Size) |>
    na.omit() |>
    rename(persons=Sample_Size) |>
    group_by(Break_Out) |>
    summarize(Response=Response,
    persons=persons,
    agg_ss=sum(persons))
```

```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf <- plotDf |>
    group_by(Break_Out,Response) |>
    summarize(agg_ss=agg_ss,
    agg_persons=sum(persons),
    agg_percent=agg_persons*100/agg_ss,
    agg_percent_sdev=
    sqrt(agg_percent*(100-agg_percent)/agg_ss),
    agg_low_ci_limit=
    agg_percent - 2*agg_percent_sdev,
    agg_high_ci_limit=
    agg_percent + 2*agg_percent_sdev) |>
    distinct() |>
    select(-c(agg_persons,agg_percent_sdev,agg_ss))
```

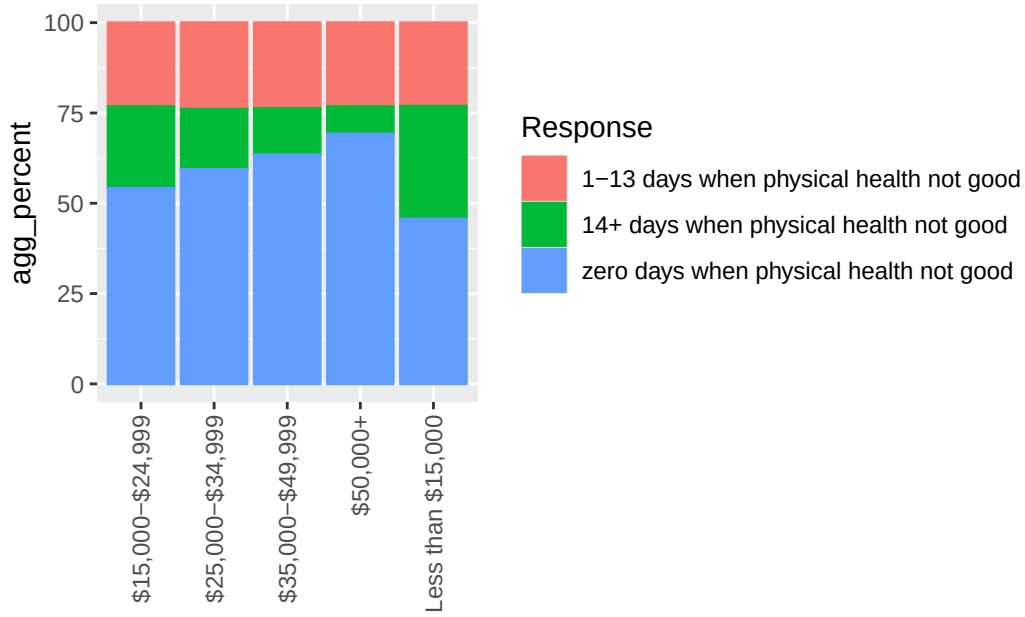
```
Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
```

```
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, remember that `reframe()`
  always returns an ungrouped data frame and adjust accordingly.
```

```
plotDf |>
        select(-agg_percent) |>
        print(n=100)
```

```
# A tibble: 15 x 4
# Groups:   Break_Out, Response [15]
   Break_Out         Response                  agg_low_ci_limit agg_high_ci_limit
   <chr>             <chr>                                <dbl>              <dbl>
 1 $15,000-$24,999   1-13 days when physical~             23.0               23.4
 2 $15,000-$24,999   14+ days when physical ~             22.4               22.8
 3 $15,000-$24,999   zero days when physical~             53.9               54.4
 4 $25,000-$34,999   1-13 days when physical~             23.8               24.2
 5 $25,000-$34,999   14+ days when physical ~             16.5               16.8
 6 $25,000-$34,999   zero days when physical~             59.2               59.6
 7 $35,000-$49,999   1-13 days when physical~             23.6               23.9
 8 $35,000-$49,999   14+ days when physical ~             12.7               13.0
 9 $35,000-$49,999   zero days when physical~             63.2               63.6
10 $50,000+          1-13 days when physical~             23.1               23.3
11 $50,000+          14+ days when physical ~              7.53               7.64
12 $50,000+          zero days when physical~             69.1               69.3
13 Less than $15,000 1-13 days when physical~             22.9               23.4
14 Less than $15,000 14+ days when physical ~             31.0               31.5
15 Less than $15,000 zero days when physical~             45.4               45.9
```

```
plotDf |>
    ggplot( aes(x=Break_Out,
    y=agg_percent,
    fill=Response,color=Response,position="fill")) +
            geom_col() + xlab(my_q) +
            theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
## summary by income
#(my_q)
#plotDf <- qDf |>
#    filter(BreakOutCategoryID=="CAT6") |>
#    select(Break_Out,Response,Sample_Size,Data_value) |>
#    na.omit() |>
#    rename(persons=Sample_Size) |>
#    mutate(Sample_Size=persons*100/Data_value) |>
#    group_by(Break_Out,Response) |>
#    summarize(agg_ss=sum(Sample_Size),
#    agg_persons=sum(persons),
#    agg_percent=agg_persons*100/agg_ss,
#    agg_percent_sdev=sqrt(agg_percent*(100-agg_percent)/agg_ss),
#    agg_low_ci_limit=agg_percent - 2*agg_percent_sdev,
#    agg_high_ci_limit=agg_percent + 2*agg_percent_sdev) |>
#        select(-c(agg_ss,agg_persons,agg_percent_sdev))
#plotDf |>
#    select(-agg_percent) |>
#    print(n=20)
#
#plotDf |>
#    ggplot( aes(x=Break_Out,
#    y=agg_percent,
```

```
#    fill=Response,color=Response,position="fill")) +
#         geom_col() + xlab(my_q)
```

Project Guidelines

- **IDE/Language** Any. Free. Publicly available.
- **team size** 1. 2. or 3. period.
- **deliverables**
  - team members listing (due 11/10/2025)
  - link-shared 10-minute recorded presentation (due 12/3/2025). *Walk-through dashboard demo. Equal time for each team member.*