

# BMI Variable Help

BSTA 512/612

[Link to github page for qmd file](#)

**Loading the needed packages:**

```
library(tidyverse)
library(gtsummary)
library(here)
if(!require(lubridate)) { install.packages("lubridate"); library(lubridate) }
```

**Loading my IAT dataset (as it's Rda file):**

```
load(file = here("../TA_files/Project/data/IAT_data.rda"))
```

**Selecting the variables that I want to look at:**

```
iat_prep = iat_2021_raw %>%
  select(IAT_score = D_biep.Thin_Good_all,
         att7, iam_001, identfat_001,
         myweight_002, myheight_002,
         identthin_001, controlother_001,
         controlyou_001, mostpref_001,
         important_001,
         birthmonth, birthyear, month, year,
         raceomb_002, raceombmulti, ethnicityomb,
         edu, edu_14,
         genderIdentity,
         birthSex)
```

## Self-reported BMI

I started investigating the BMI because I was curious how the paper [elran-barak2018] used it and just wanted to check reproducibility. There are a few issues with the self-reported BMI that immediately stuck out:

- Components of BMI (weight and height) were self-reported
  - People told they are underweight often add pounds (REFERENCE)
  - People told they are overweight often subtract pounds (REFERENCE)
- Raw data from weight and height are categorical. This is according to the codebook associated with this dataset. Please find your codebook file named `Weight_IAT_public_2021_codebook.csv`. You can find the value names for `myweight_002` and `myheight_002`.

For example, in the weight variable,

- most categories identify a lower limit to the weight in the group. One example group is weight is greater than or equal to 200 pounds and less than 205 pounds (labelled as “200 lb :: 91 kg”).
- the first category for weight is “below 50lb:: 23kg” with 258 observations
- the last category for weight is “above 440lb:: above 200kg” with 295 observations
  - \* While the 5 groups of weight leading up the last category have 33, 28, 34, 20, and 89 observations, respectively.
- My intention here is not the question anyone’s weight, but keep in mind that surveys sometimes have people selecting the first or last option because they are not taking the survey seriously

## My exact steps

1. I wanted to get a table of the counts within each weight group. I used the `gt` package to make a table of what I thought was a categorical variable. It looks like R interprets the numbered categories as numbers.

```
iat_prep %>%  
  dplyr::select(myweight_002) %>%  
  tbl_summary()
```

2. I will first check the class of the variable to make sure R is doing what I think it’s doing.

Characteristic	N = 465,886 <sup>1</sup>
myweight_002	23 (18, 29)
Unknown	141,326

<sup>1</sup>Median (Q1, Q3)

```
class(iat_prep$myweight_002)
```

```
[1] "integer"
```

So R is interpreting the values as integers. I will need to make them categories to view them through `gt` commands.

- Let's make it a category:

```
iat_prep2 = iat_prep %>%
  mutate(myweight = as.factor(myweight_002))
```

- Now we make the table:

```
iat_prep2 %>%
  dplyr::select(myweight) %>%
  tbl_summary()
```

- The table is really long, so a histogram would work much better to visualize how many observations are in each category:

```
ggplot(data = iat_prep, aes(x = myweight_002)) +
  geom_histogram() +
  geom_vline(aes(xintercept = mean(iat_prep$myweight_002,
                                   na.rm = T)),
             color = "red", linewidth = 2)
```

Warning: Use of `iat\_prep\$myweight\_002` is discouraged.  
i Use `myweight\_002` instead.

Warning: Removed 141326 rows containing non-finite outside the scale range  
(`stat\_bin()`).

Characteristic	N = 465,886 <sup>1</sup>
myweight	
1	258 (<0.1%)
2	257 (<0.1%)
3	329 (0.1%)
4	363 (0.1%)
5	379 (0.1%)
6	329 (0.1%)
7	327 (0.1%)
8	360 (0.1%)
9	589 (0.2%)
10	1,002 (0.3%)
11	2,180 (0.7%)
12	3,766 (1.2%)
13	6,175 (1.9%)
14	9,038 (2.8%)
15	12,068 (3.7%)
16	15,598 (4.8%)
17	16,007 (4.9%)
18	17,518 (5.4%)
19	19,093 (5.9%)
20	17,794 (5.5%)
21	15,599 (4.8%)
22	16,636 (5.1%)
23	14,854 (4.6%)
24	14,643 (4.5%)
25	13,510 (4.2%)
26	12,778 (3.9%)
27	12,243 (3.8%)
28	11,498 (3.5%)
29	9,414 (2.9%)
30	9,099 (2.8%)
31	7,274 (2.2%)
32	8,775 (2.7%)
33	4,691 (1.4%)
34	5,411 (1.7%)
35	4,595 (1.4%)
36	5,659 (1.7%)
37	3,494 (1.1%)
38	3,938 (1.2%)
39	2,489 (0.8%)
40	2,932 (0.9%)
41	1,941 (0.6%)
42	3,197 (1.0%)
43	1,244 (0.4%)
44	1,794 (0.6%)
45	1,442 (0.4%)
46	1,322 (0.4%)
47	1,251 (0.4%)
48	1,238 (0.4%)