NYC_schools_perceptions

Bena

2023-05-19

Setting up

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0     ✓ purrr   1.0.1
## ✓ tibble  3.1.8     ✓ dplyr   1.1.0
## ✓ tidyr   1.3.0     ✓ stringr 1.5.0
## ✓ readr   2.1.3     ✓ forcats 1.0.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Import files

The data was downloaded from data.world & NYC education websites using the links below: https://data.world/dataquest/nyc-schools-data/workspace/file?filename=combined.csv (https://data.world/dataquest/nyc-schools-data/workspace/file?filename=combined.csv) & https://data.cityofnewyork.us/Education/2011-NYC-School-Survey/mnz3-dyi8 (https://data.cityofnewyork.us/Education/2011-NYC-School-Survey/mnz3-dyi8)

```
nyc_hs <- read_csv("combined.csv")
```

```
## Rows: 479 Columns: 30
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): DBN, school_name, boro
## dbl (27): Num of SAT Test Takers, SAT Critical Reading Avg. Score, SAT Math ...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
survey_gened <- read_tsv("masterfile11_gened_final.txt")
```

```
## Rows: 1646 Columns: 1942
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: "\t"
## chr    (5): dbn, bn, schoolname, studentssurveyed, schooltype
## dbl (1904): d75, highschool, rr_s, rr_t, rr_p, N_s, N_t, N_p, nr_s, nr_t, nr...
## lgl   (33): p_q1, p_q3d, p_q9, p_q10, p_q12aa, p_q12ab, p_q12ac, p_q12ad, p_...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
survey_d75 <- read_tsv("masterfile11_d75_final.txt")
```

```
## Rows: 56 Columns: 1773
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: "\t"
## chr    (5): dbn, bn, schoolname, studentssurveyed, schooltype
## dbl (1739): d75, highschool, rr_s, rr_t, rr_p, N_s, N_t, N_p, nr_s, nr_t, nr...
## lgl   (29): p_q5, p_q9, p_q13a, p_q13b, p_q13c, p_q13d, p_q14a, p_q14b, p_q1...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Get overview of data

```
head(nyc_hs)
```

```
## # A tibble: 6 × 30
##   DBN    schoo…¹ Num o…² SAT C…³ SAT M…⁴ SAT W…⁵ avg_s…⁶ AP Te…⁷ Total…⁸ Numbe…⁹
##   <chr>  <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 01M292 HENRY …      29     355     404     363    1122     2.5      NA      NA
## 2 01M448 UNIVER…      91     383     423     366    1172    39        49      10
## 3 01M450 EAST S…      70     377     402     370    1149    19        21      NA
## 4 01M458 FORSYT…       7     414     401     359    1174     2.5      NA      NA
## 5 01M509 MARTA …      44     390     433     384    1207     2.5      NA      NA
## 6 01M515 LOWER …     112     332     557     316    1205    24        26      24
## # … with 20 more variables: exams_per_student <dbl>, high_score_percent <dbl>,
## #   avg_class_size <dbl>, frl_percent <dbl>, total_enrollment <dbl>,
## #   ell_percent <dbl>, sped_percent <dbl>, selfcontained_num <dbl>,
## #   asian_per <dbl>, black_per <dbl>, hispanic_per <dbl>, white_per <dbl>,
## #   male_per <dbl>, female_per <dbl>, `Total Cohort` <dbl>,
## #   grads_percent <dbl>, dropout_percent <dbl>, boro <chr>, lat <dbl>,
## #   long <dbl>, and abbreviated variable names ¹school_name, …
```

```
head(survey_gened)
```

```
## # A tibble: 6 × 1,942
##   dbn    bn    schoolname    d75 stude…¹ highs…² schoo…³  rr_s  rr_t  rr_p   N_s
##   <chr>  <chr> <chr>       <dbl> <chr>     <dbl> <chr>   <dbl> <dbl> <dbl> <dbl>
## 1 01M015 M015  P.S. 015 R…     0 No            0 Elemen…    NA    88    60    NA
## 2 01M019 M019  P.S. 019 A…     0 No            0 Elemen…    NA   100    60    NA
## 3 01M020 M020  P.S. 020 A…     0 No            0 Elemen…    NA    88    73    NA
## 4 01M034 M034  P.S. 034 F…     0 Yes           0 Elemen…    89    73    50   145
## 5 01M063 M063  P.S. 063 W…     0 No            0 Elemen…    NA   100    60    NA
## 6 01M064 M064  P.S. 064 R…     0 No            0 Elemen…    NA    94    61    NA
## # … with 1,931 more variables: N_t <dbl>, N_p <dbl>, nr_s <dbl>, nr_t <dbl>,
## #   nr_p <dbl>, saf_p_11 <dbl>, com_p_11 <dbl>, eng_p_11 <dbl>, aca_p_11 <dbl>,
## #   saf_t_11 <dbl>, com_t_11 <dbl>, eng_t_11 <dbl>, aca_t_11 <dbl>,
## #   saf_s_11 <dbl>, com_s_11 <dbl>, eng_s_11 <dbl>, aca_s_11 <dbl>,
## #   saf_tot_11 <dbl>, com_tot_11 <dbl>, eng_tot_11 <dbl>, aca_tot_11 <dbl>,
## #   p_q2h <dbl>, p_q7a <dbl>, p_q7b <dbl>, p_q7c <dbl>, p_q7d <dbl>,
## #   p_q8a <dbl>, p_q8b <dbl>, p_q8c <dbl>, p_q8d <dbl>, p_q8e <dbl>, …
```

```
head(survey_d75)
```

```
## # A tibble: 6 × 1,773
##   dbn    bn    schoolname   d75 studen…¹ highs…² schoo…³  rr_s  rr_t  rr_p   N_s
##   <chr>  <chr> <chr>      <dbl> <chr>     <dbl> <chr>   <dbl> <dbl> <dbl> <dbl>
## 1 75K004 K004  P.S. K004      1 Yes           0 Distri…    38    90    72     8
## 2 75K036 K036  P.S. 36        1 Yes          NA Distri…    70    69    44    97
## 3 75K053 K053  P.S. K053      1 Yes          NA Distri…    94    97    53   131
## 4 75K077 K077  P.S. K077      1 Yes          NA Distri…    95    65    55    70
## 5 75K140 K140  P.S. K140      1 Yes           0 Distri…    77    70    42    69
## 6 75K141 K141  P.S. K141      1 Yes          NA Distri…    73    55    29    53
## # … with 1,762 more variables: N_t <dbl>, N_p <dbl>, nr_s <dbl>, nr_t <dbl>,
## #   nr_p <dbl>, saf_p_11 <dbl>, com_p_11 <dbl>, eng_p_11 <dbl>, aca_p_11 <dbl>,
## #   saf_t_11 <dbl>, com_t_11 <dbl>, eng_t_11 <dbl>, aca_t_11 <dbl>,
## #   saf_s_11 <dbl>, com_s_11 <dbl>, eng_s_11 <dbl>, aca_s_11 <dbl>,
## #   saf_tot_11 <dbl>, com_tot_11 <dbl>, eng_tot_11 <dbl>, aca_tot_11 <dbl>,
## #   p_q1c <dbl>, p_q10a <dbl>, p_q10b <dbl>, p_q10c <dbl>, p_q10d <dbl>,
## #   p_q10e <dbl>, p_q10f <dbl>, p_q11a <dbl>, p_q11b <dbl>, p_q11c <dbl>, …
```

```
dim(nyc_hs)
```

```
## [1] 479  30
```

NYC High School data has 479 rows & 30 columns

```
dim(survey_gened)
```

```
## [1] 1646 1942
```

general education survey data has 1646 rows & 1942 columns

```
dim(survey_d75)
```

```
## [1]   56 1773
```

district75 survey data has 56 rows & 1773 columns

A closer look at the data

What are the school categories?

```
schooltype_gened <- survey_gened %>%
  pull(schooltype) %>%
  unique

print(schooltype_gened)
```

```
## [1] "Elementary School"                "Elementary / Middle School"
## [3] "Middle / High School"             "Middle School"
## [5] "High School"                      "Elementary / Middle / High School"
## [7] "Early Childhood School"           "YABC"
```

```
schooltype_d75 <- survey_d75 %>%
  pull(schooltype) %>%
  unique

print(schooltype_d75)
```

```
## [1] "District 75 Special Education"
```

School type in d75 data is “District 75 Special Education, which may refer to either high schools or elementary schools.

Since nyc_hs only contains data for NYC high schools, we’ll filter the school type for schooltype_gen We’ll also remove some columns & remain with those that are necessary to do the analysis. The removed columns show responses to each of the survey questions.

```
survey_gened_select <- survey_gened %>%
  filter(schooltype == "High School") %>%
  select(dbn:aca_tot_11)

View(survey_gened_select)
```

```
survey_d75_select <- survey_d75 %>%
  select(dbn:aca_tot_11)

View(survey_d75_select)
```

Remove the `bn` column since it's contained in `dbn` variable

```
survey_gened_select <- survey_gened_select %>%
  select(-bn)

survey_d75_select <- survey_d75_select %>%
  select(-bn)
```

Combine the 2 dataframes for surveys

```
survey_combined <- bind_rows(survey_gened_select, survey_d75_select)

View(survey_combined)
```

Alternatively

```
survey_combined <- survey_gened_select %>%
  bind_rows(survey_d75_select)
```

Since we're interested in relationships of survey_combined data with variables in the nyc_hs dataframe, it makes sense to join the survey data to nyc_hs using left_join(). This will retain only observations in the survey dataframe that correspond to observations in nyc_hs. Before joining, rename `dbn` so it matches `DBN` in nyc_hs

```
survey_combined <- survey_combined %>%
  rename(DBN = "dbn")
```

```
survey_nychs <- nyc_hs %>%
  left_join(survey_combined, by = "DBN")

View(survey_nychs)
```

Establish which correlations may be interesting to explore further

```
correlation_matrix <- survey_nychs %>%
  select(avg_sat_score, saf_p_11:aca_tot_11) %>%
  cor(use = "pairwise.complete.obs")

print(correlation_matrix)
```

```
##               avg_sat_score   saf_p_11   com_p_11  eng_p_11    aca_p_11
## avg_sat_score   1.000000000 0.08913424 -0.1139639 0.0094307 0.008276863
## saf_p_11        0.089134237 1.00000000  0.8029932 0.7984741 0.817219807
## com_p_11       -0.113963937 0.80299315  1.0000000 0.9291268 0.925565838
## eng_p_11        0.009430700 0.79847408  0.9291268 1.0000000 0.907884041
## aca_p_11        0.008276863 0.81721981  0.9255658 0.9078840 1.000000000
## saf_t_11        0.309144422 0.49046519  0.3149366 0.3392305 0.415400554
## com_t_11        0.107470334 0.23164622  0.2411942 0.2675333 0.285824766
## eng_t_11        0.056125707 0.31403333  0.3071861 0.3296697 0.373418577
## aca_t_11        0.141157527 0.36423106  0.3470251 0.3585188 0.419085881
## saf_s_11        0.277268115 0.73241477  0.5411576 0.5390565 0.593038790
## com_s_11        0.162302236 0.61691582  0.5687691 0.5450478 0.598092320
## eng_s_11        0.170346019 0.64749568  0.5913392 0.6075781 0.622737241
## aca_s_11        0.292587986 0.69197352  0.6146925 0.6236426 0.679837302
## saf_tot_11      0.276041045 0.82764669  0.6105585 0.6209748 0.682848168
## com_tot_11      0.089098561 0.64231997  0.6933924 0.6820356 0.711988528
## eng_tot_11      0.094794066 0.66370163  0.6714581 0.7101096 0.720034872
## aca_tot_11      0.173800408 0.70043795  0.7028261 0.7054661 0.789215088
##                saf_t_11  com_t_11   eng_t_11  aca_t_11  saf_s_11  com_s_11
## avg_sat_score 0.3091444 0.1074703 0.05612571 0.1411575 0.2772681 0.1623022
## saf_p_11      0.4904652 0.2316462 0.31403333 0.3642311 0.7324148 0.6169158
## com_p_11      0.3149366 0.2411942 0.30718612 0.3470251 0.5411576 0.5687691
## eng_p_11      0.3392305 0.2675333 0.32966966 0.3585188 0.5390565 0.5450478
## aca_p_11      0.4154006 0.2858248 0.37341858 0.4190859 0.5930388 0.5980923
## saf_t_11      1.0000000 0.7437526 0.79946285 0.8448628 0.5922487 0.3825288
## com_t_11      0.7437526 1.0000000 0.90329649 0.8960927 0.2527900 0.1683019
## eng_t_11      0.7994629 0.9032965 1.00000000 0.9498147 0.3678941 0.2887301
## aca_t_11      0.8448628 0.8960927 0.94981468 1.0000000 0.4186507 0.3239221
## saf_s_11      0.5922487 0.2527900 0.36789407 0.4186507 1.0000000 0.8227886
## com_s_11      0.3825288 0.1683019 0.28873011 0.3239221 0.8227886 1.0000000
## eng_s_11      0.4565140 0.2585602 0.35784213 0.3881175 0.8847919 0.8836172
## aca_s_11      0.4932720 0.2652035 0.35936021 0.4089070 0.8719545 0.8649279
## saf_tot_11    0.8435345 0.5074542 0.60846241 0.6660397 0.8908945 0.6907220
## com_tot_11    0.7363689 0.7997233 0.80356551 0.8236332 0.6412669 0.6575138
## eng_tot_11    0.7568738 0.7214690 0.83787593 0.8276065 0.7078967 0.6591503
## aca_tot_11    0.7817245 0.6943313 0.78680846 0.8445192 0.7142941 0.6597831
##                eng_s_11  aca_s_11 saf_tot_11 com_tot_11 eng_tot_11 aca_tot_11
## avg_sat_score 0.1703460 0.2925880  0.2760410 0.08909856 0.09479407  0.1738004
## saf_p_11      0.6474957 0.6919735  0.8276467 0.64231997 0.66370163  0.7004380
## com_p_11      0.5913392 0.6146925  0.6105585 0.69339241 0.67145808  0.7028261
## eng_p_11      0.6075781 0.6236426  0.6209748 0.68203559 0.71010958  0.7054661
## aca_p_11      0.6227372 0.6798373  0.6828482 0.71198853 0.72003487  0.7892151
## saf_t_11      0.4565140 0.4932720  0.8435345 0.73636889 0.75687379  0.7817245
## com_t_11      0.2585602 0.2652035  0.5074542 0.79972327 0.72146902  0.6943313
## eng_t_11      0.3578421 0.3593602  0.6084624 0.80356551 0.83787593  0.7868085
## aca_t_11      0.3881175 0.4089070  0.6660397 0.82363320 0.82760652  0.8445192
## saf_s_11      0.8847919 0.8719545  0.8908945 0.64126689 0.70789667  0.7142941
## com_s_11      0.8836172 0.8649279  0.6907220 0.65751381 0.65915031  0.6597831
## eng_s_11      1.0000000 0.9213798  0.7573247 0.68395614 0.75920793  0.7208915
## aca_s_11      0.9213798 1.0000000  0.7855468 0.69009358 0.73953389  0.7778084
## saf_tot_11    0.7573247 0.7855468  1.0000000 0.79013553 0.83403143  0.8601577
```

```
## com_tot_11    0.6839561 0.6900936  0.7901355 1.00000000 0.93995349  0.9317467
## eng_tot_11    0.7592079 0.7395339  0.8340314 0.93995349 1.00000000  0.9525649
## aca_tot_11    0.7208915 0.7778084  0.8601577 0.93174674 0.95256492  1.0000000
```

Convert the matrix to tibble so it's easier to work with

```
correlation_tibble <- correlation_matrix %>%
  as_tibble(rownames = "variable")

print(correlation_tibble)
```

```
## # A tibble: 17 × 18
##    variable     avg_sa…¹ saf_p…² com_p…³ eng_p…⁴ aca_p…⁵ saf_t…⁶ com_t…⁷ eng_t…⁸
##    <chr>           <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 avg_sat_sco…  1        0.0891  -0.114 0.00943 0.00828   0.309   0.107  0.0561
##  2 saf_p_11      0.0891   1        0.803 0.798   0.817     0.490   0.232  0.314
##  3 com_p_11     -0.114    0.803    1     0.929   0.926     0.315   0.241  0.307
##  4 eng_p_11      0.00943  0.798    0.929 1       0.908     0.339   0.268  0.330
##  5 aca_p_11      0.00828  0.817    0.926 0.908   1         0.415   0.286  0.373
##  6 saf_t_11      0.309    0.490    0.315 0.339   0.415     1       0.744  0.799
##  7 com_t_11      0.107    0.232    0.241 0.268   0.286     0.744   1      0.903
##  8 eng_t_11      0.0561   0.314    0.307 0.330   0.373     0.799   0.903  1
##  9 aca_t_11      0.141    0.364    0.347 0.359   0.419     0.845   0.896  0.950
## 10 saf_s_11      0.277    0.732    0.541 0.539   0.593     0.592   0.253  0.368
## 11 com_s_11      0.162    0.617    0.569 0.545   0.598     0.383   0.168  0.289
## 12 eng_s_11      0.170    0.647    0.591 0.608   0.623     0.457   0.259  0.358
## 13 aca_s_11      0.293    0.692    0.615 0.624   0.680     0.493   0.265  0.359
## 14 saf_tot_11    0.276    0.828    0.611 0.621   0.683     0.844   0.507  0.608
## 15 com_tot_11    0.0891   0.642    0.693 0.682   0.712     0.736   0.800  0.804
## 16 eng_tot_11    0.0948   0.664    0.671 0.710   0.720     0.757   0.721  0.838
## 17 aca_tot_11    0.174    0.700    0.703 0.705   0.789     0.782   0.694  0.787
## # … with 9 more variables: aca_t_11 <dbl>, saf_s_11 <dbl>, com_s_11 <dbl>,
## #   eng_s_11 <dbl>, aca_s_11 <dbl>, saf_tot_11 <dbl>, com_tot_11 <dbl>,
## #   eng_tot_11 <dbl>, aca_tot_11 <dbl>, and abbreviated variable names
## #   ¹avg_sat_score, ²saf_p_11, ³com_p_11, ⁴eng_p_11, ⁵aca_p_11, ⁶saf_t_11,
## #   ⁷com_t_11, ⁸eng_t_11
```

Select variables of interest moderate to strong relationships have a correlation coefficient, Pearson's r value <0.25 or >-0.25

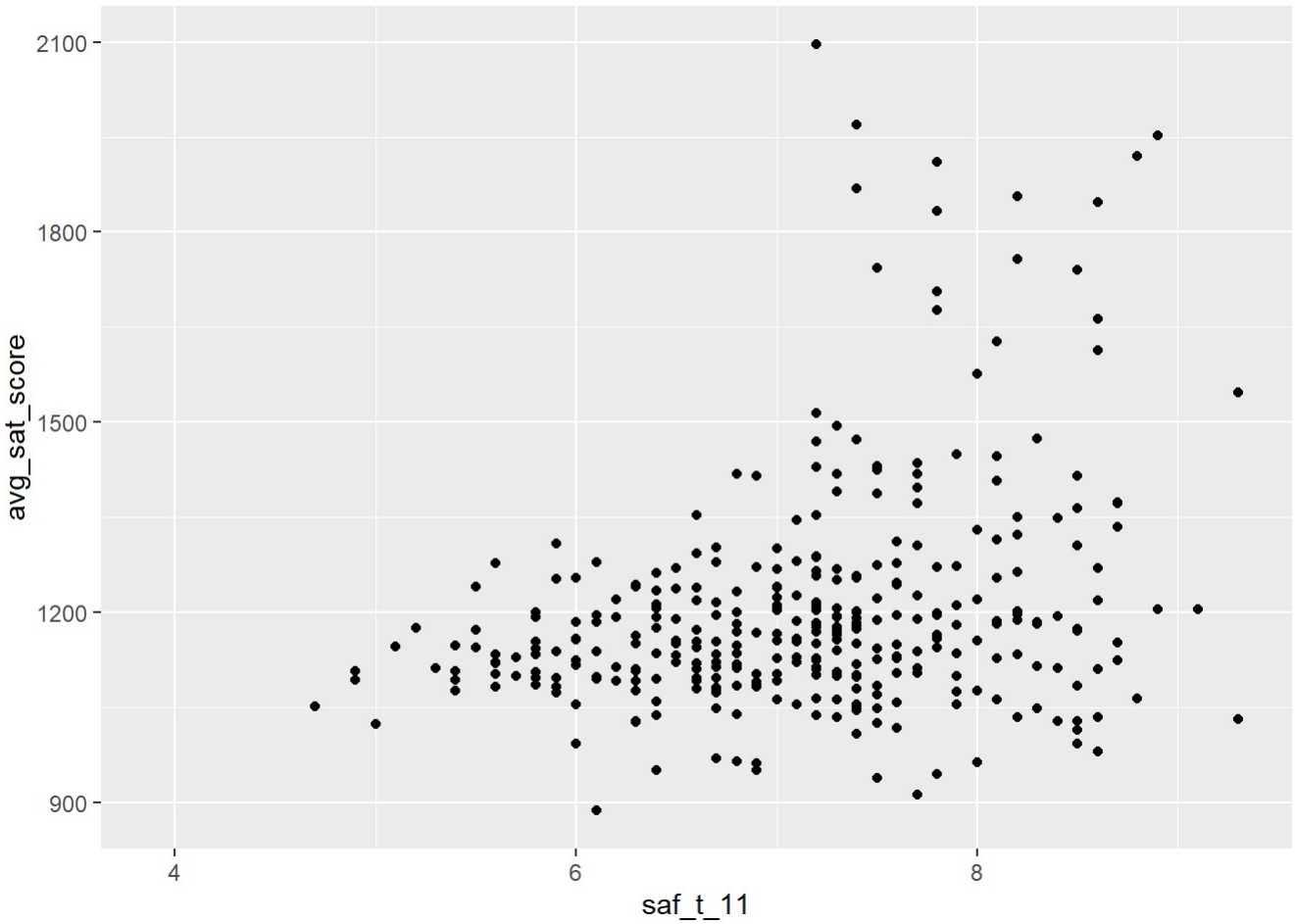
```
strong_cors <- correlation_tibble %>%
  select(variable, avg_sat_score) %>%
  filter(avg_sat_score > 0.25 | avg_sat_score < -0.25)

print(strong_cors)
```

```
## # A tibble: 5 × 2
##   variable      avg_sat_score
##   <chr>                 <dbl>
## 1 avg_sat_score         1
## 2 saf_t_11              0.309
## 3 saf_s_11              0.277
## 4 aca_s_11              0.293
## 5 saf_tot_11            0.276
```

Visualize for closer examination

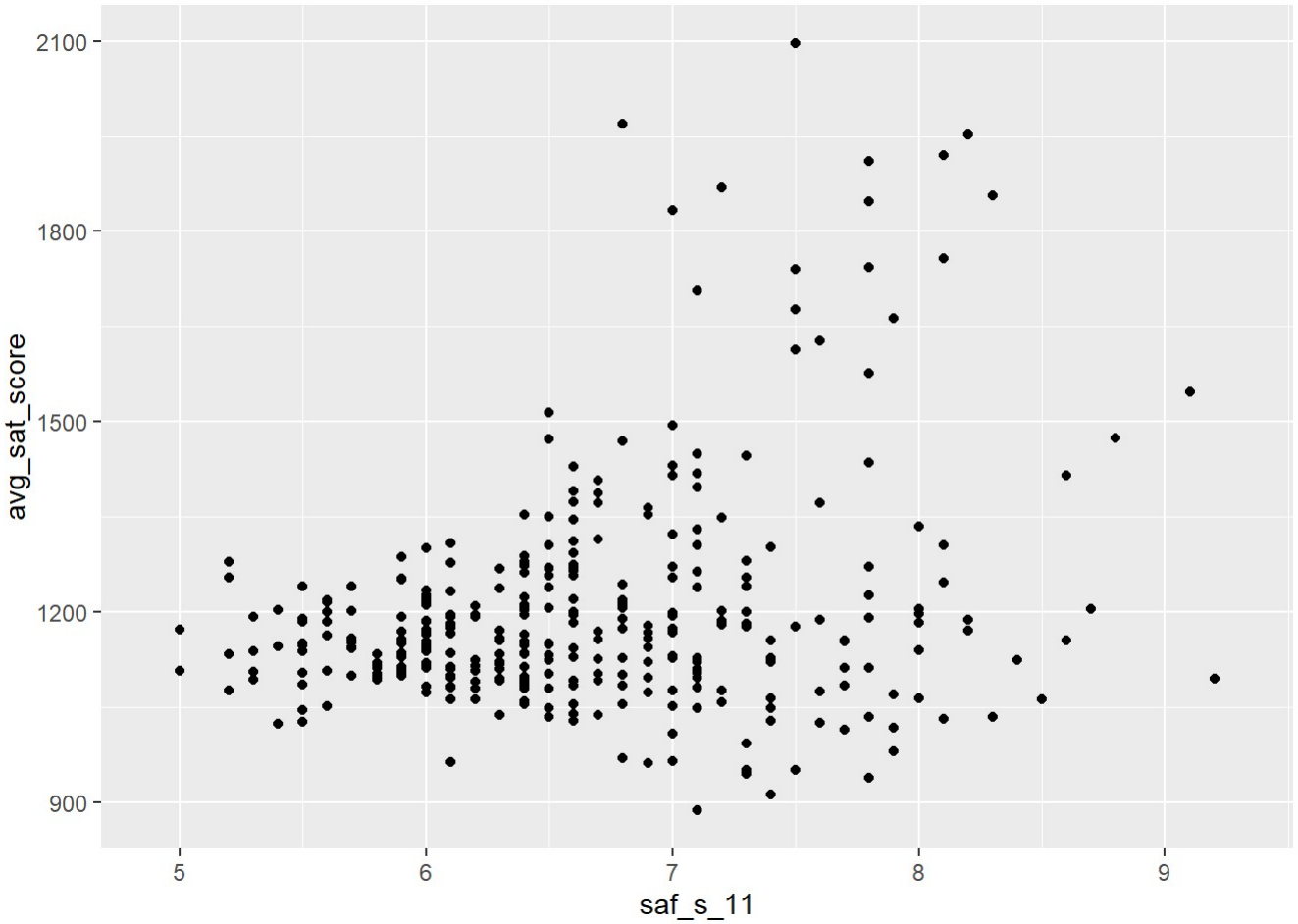
```
ggplot(data = survey_nychs,
       aes(x = saf_t_11, y = avg_sat_score)) +
       geom_point()
```

```
## Warning: Removed 137 rows containing missing values (`geom_point()`).
```



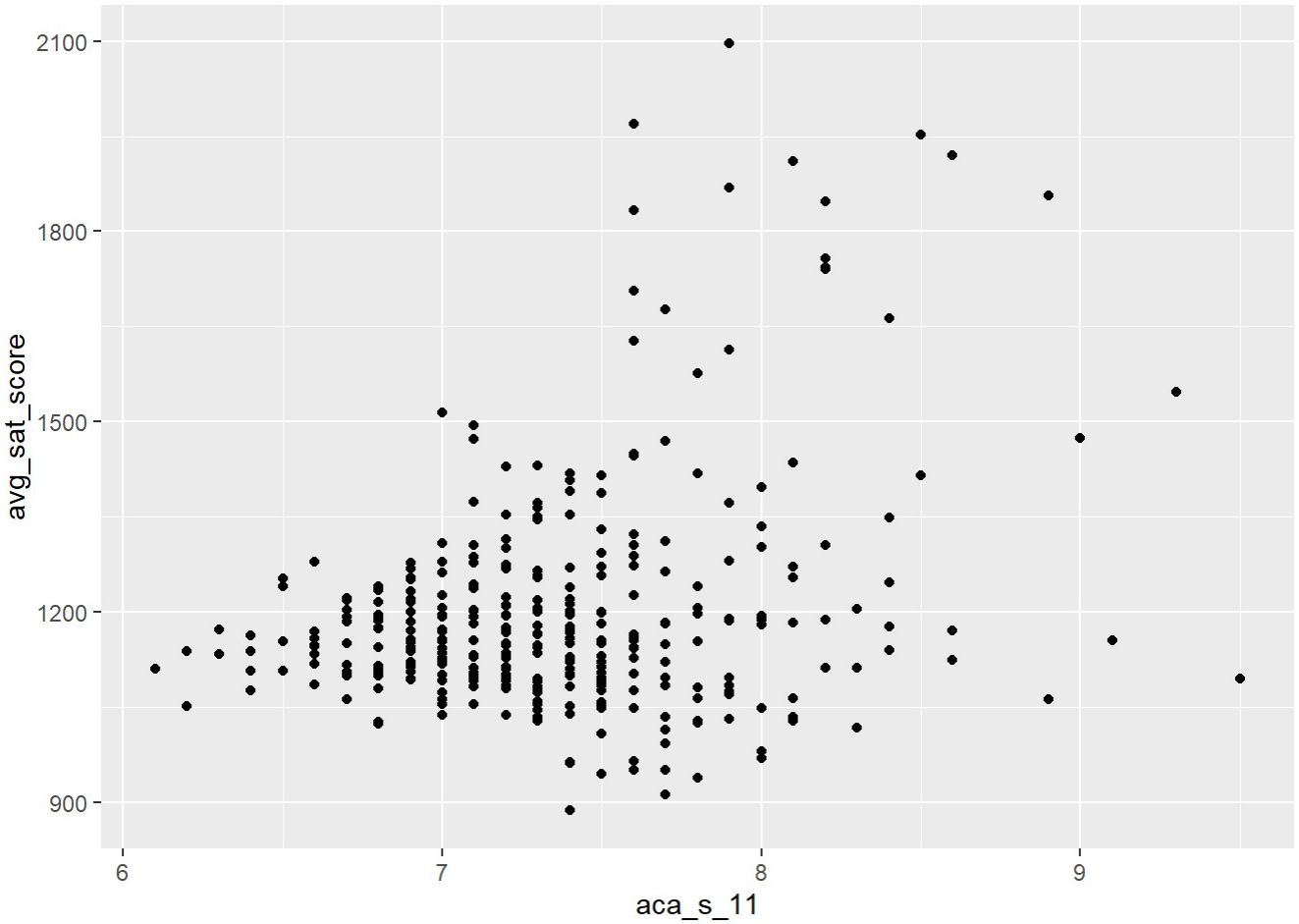

```
ggplot(data = survey_nychs,
       aes(x = saf_s_11, y = avg_sat_score)) +
  geom_point()
```

```
## Warning: Removed 139 rows containing missing values (`geom_point()`).
```



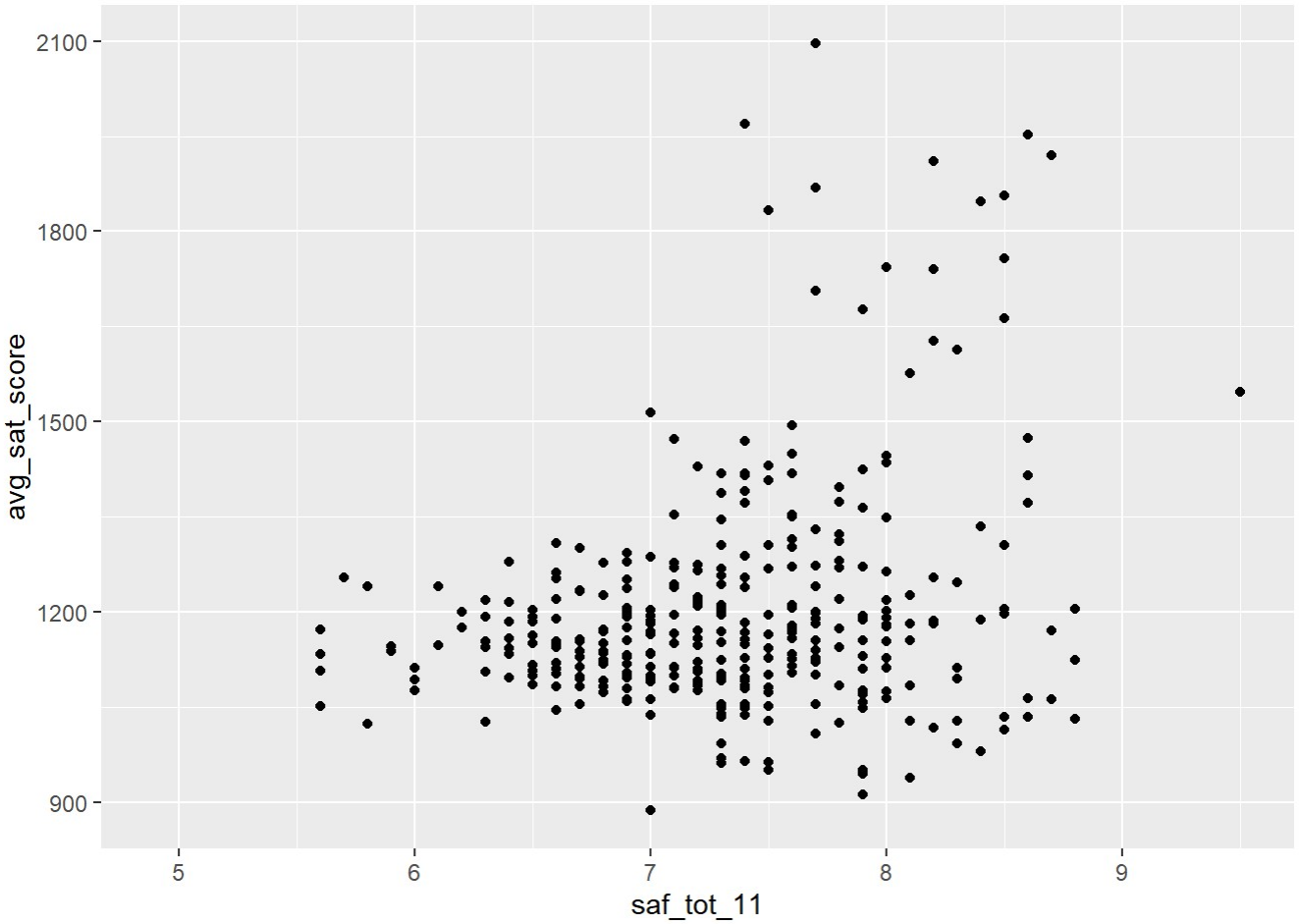

```
ggplot(data = survey_nychs,
       aes(x = aca_s_11, y = avg_sat_score)) +
  geom_point()
```

```
## Warning: Removed 139 rows containing missing values (`geom_point()`).
```




```
ggplot(data = survey_nychs,
       aes(x = saf_tot_11, y = avg_sat_score)) +
  geom_point()
```

```
## Warning: Removed 137 rows containing missing values (`geom_point()`).
```



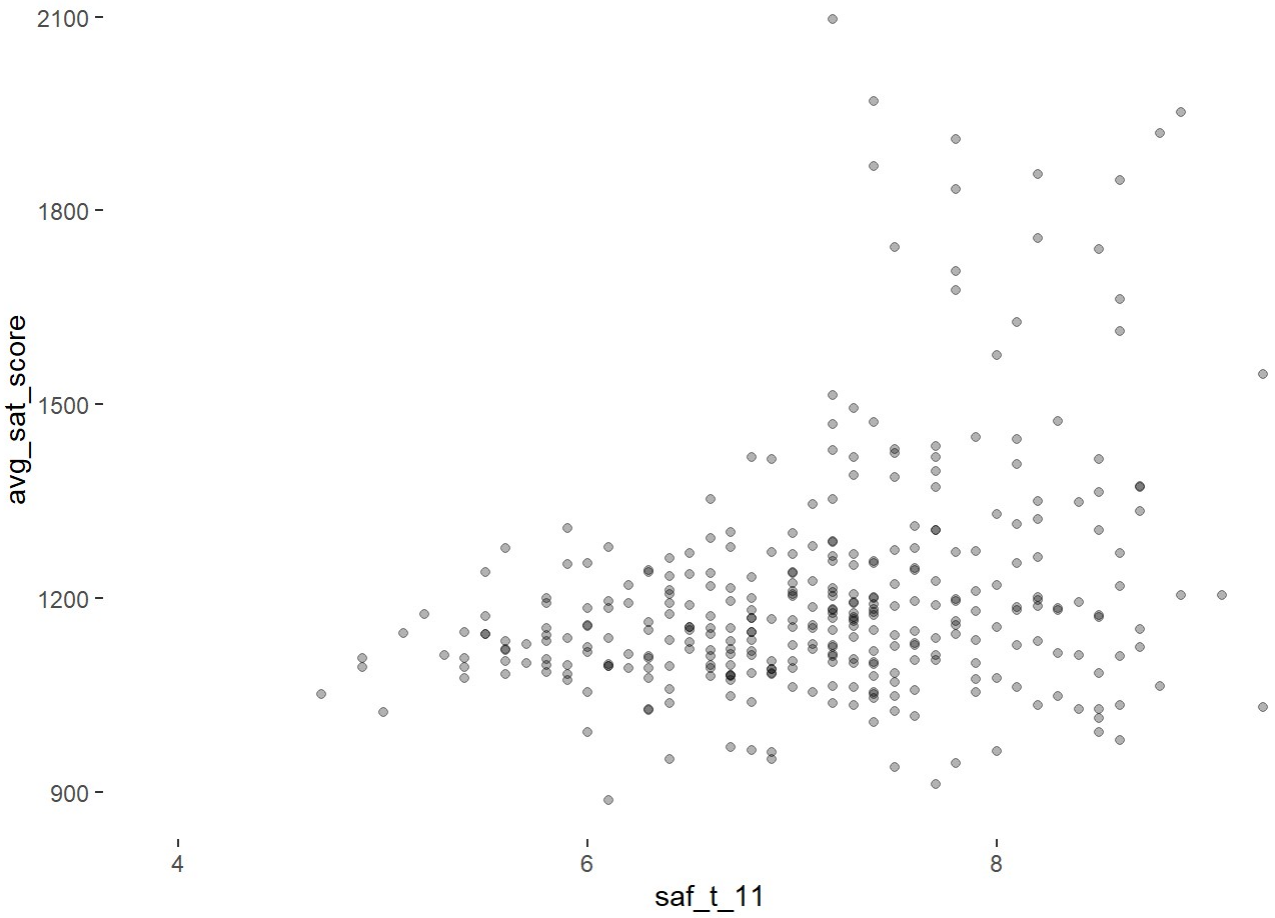
Alternatively, we can iterate using function so we don't have to write 4 different code chunks

```
create_scatter <- function(x, y) {
  ggplot(data = survey_nychs) +
    aes_string(x = x, y = y) +
    geom_point(alpha = 0.3) +
    theme(panel.background = element_rect(fill = "white"))
}
x_var <- strong_cors$variable[2:5]
y_var <- "avg_sat_score"

map2(x_var, y_var, create_scatter)
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation ideoms with `aes()`
```

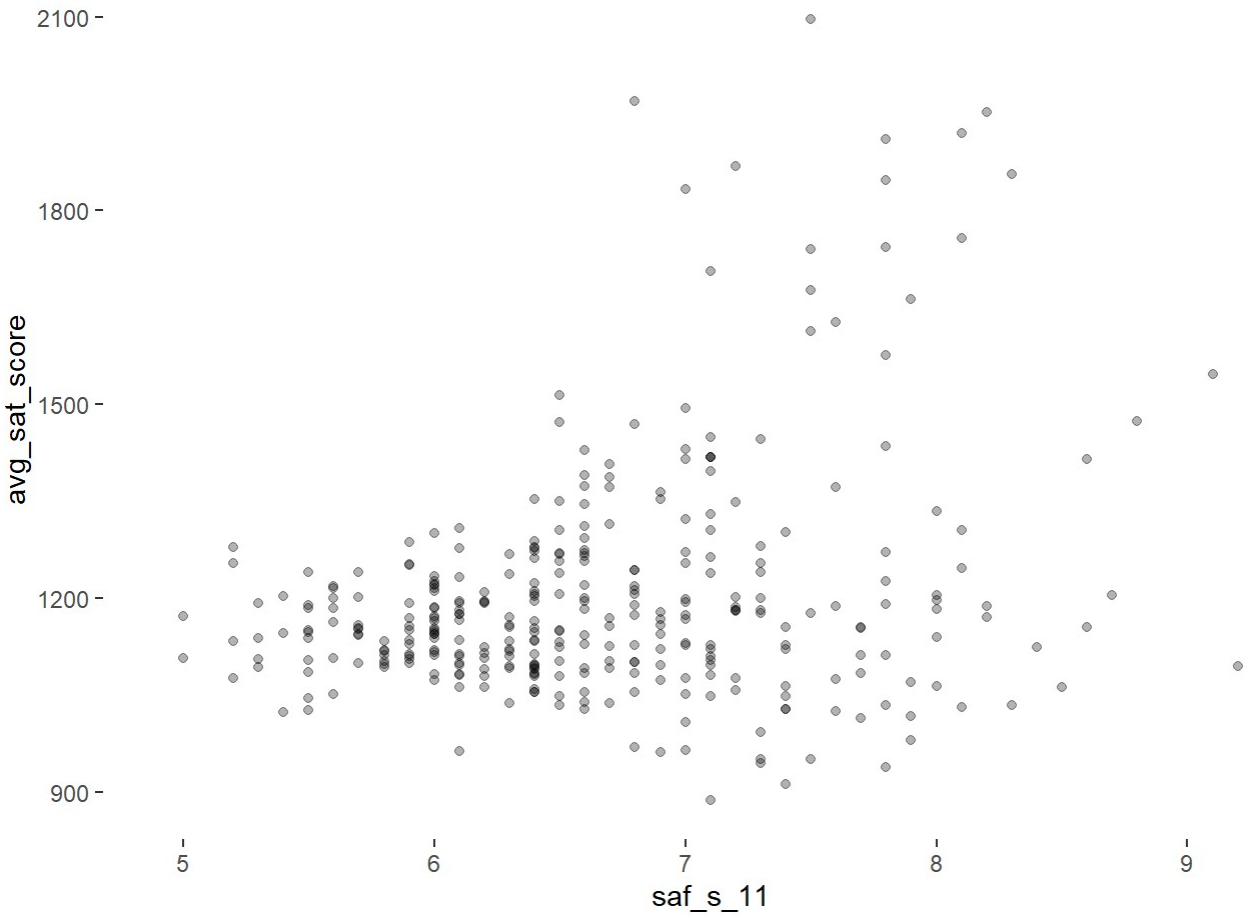
```
## [[1]]
```

```
## Warning: Removed 137 rows containing missing values (`geom_point()`).
```



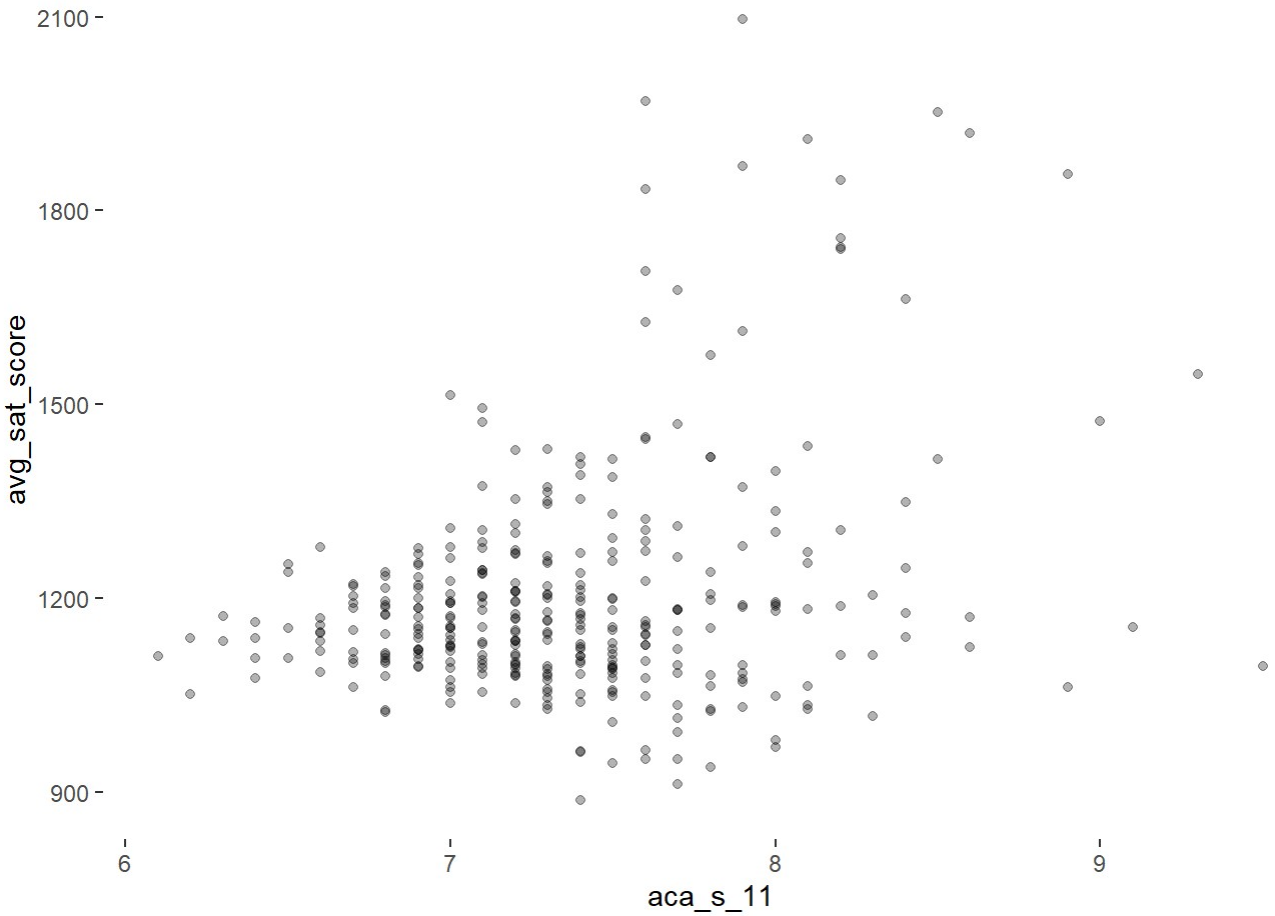

```
##
## [[2]]
```

```
## Warning: Removed 139 rows containing missing values (`geom_point()`).
```



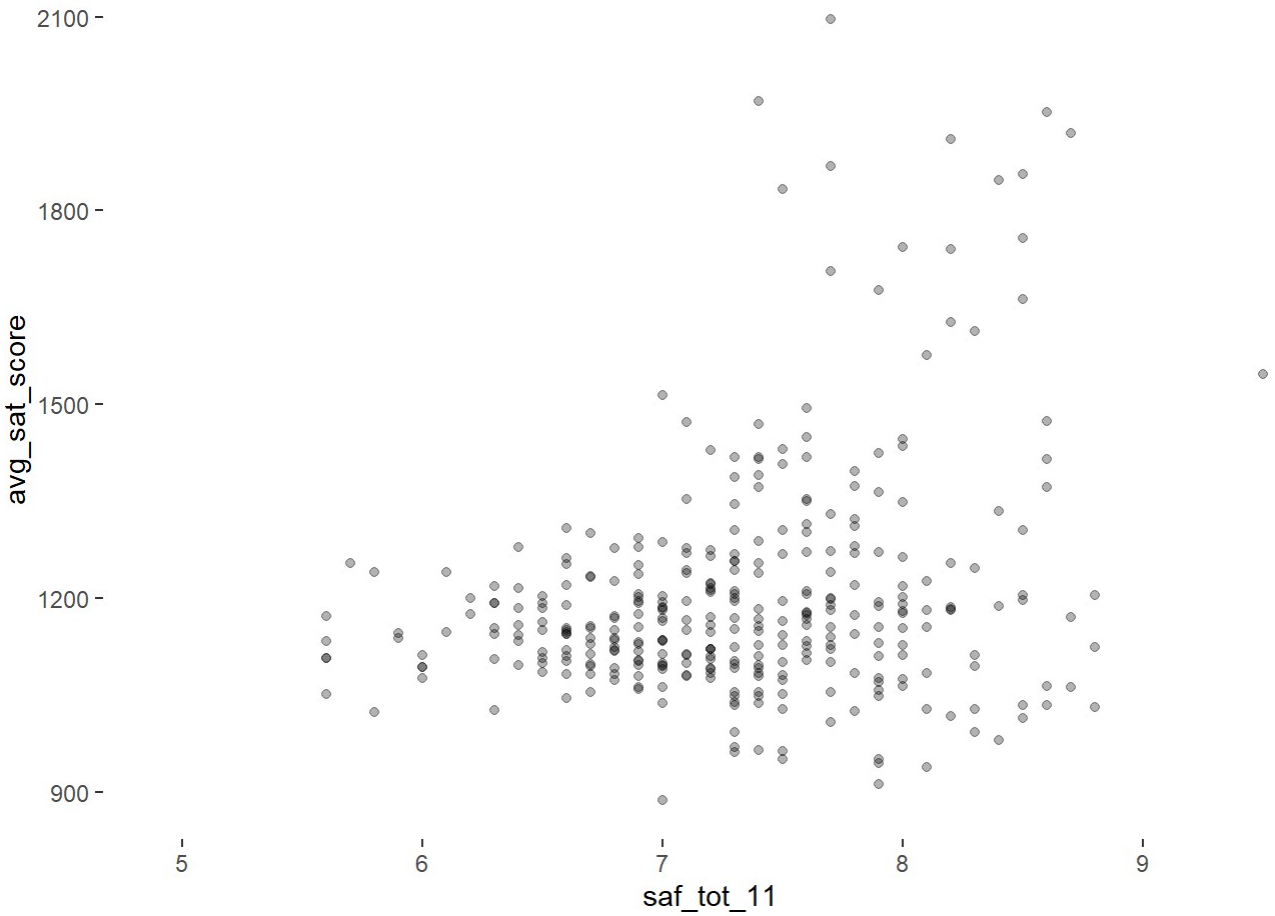

```
##
## [[3]]
```

```
## Warning: Removed 139 rows containing missing values (`geom_point()`).
```




```
##
## [[4]]
```

```
## Warning: Removed 137 rows containing missing values (`geom_point()`).
```




```
print(x_var)
```

```
## [1] "saf_t_11"   "saf_s_11"   "aca_s_11"   "saf_tot_11"
```

The avg_sat_score has a positive relationship with all 4 variables of interest i.e: “saf_t_11”, “saf_s_11”, “aca_s_11”, “saf_tot_11”
In other words, the sat_score increased the more Safety and Respect score (for teachers) increased. Same case applied to students and all respondent groups. The sat_score increased the more Academic expectations score (for students) increased.

Reshape the data so it’s more informative

To make it easier to see responses for all metrics by all groups, we’ll reshape the data from wide to long data

```
survey_nychs_long <- survey_nychs %>%
  pivot_longer(
    cols = c(saf_p_11:aca_tot_11),       #stacks the perception columns(survey questions) tog
ether
    names_to = "survey_question",
    values_to = "score"
  )

View(survey_nychs_long)
```

Create 2 new variables from the survey_question column i.e. metric & respondent

```
survey_nychs_long <- survey_nychs_long %>%
  mutate(
    metric = str_sub(survey_question, 1, 3),       #str_sub() function extracted infor from t
he survey_question variable
    respondent = str_sub(survey_question, 4, 6)    # start & end characters could have been
5, 5 but there are 2 t's for teacher & total
  )
```

Replace the values in metric & respondent columns with more meaningful names

```
survey_nychs_long <- survey_nychs_long %>%
  mutate(
    metric = case_when(
      metric == "saf" ~ "safety & respect",
      metric == "com" ~ "communication",
      metric == "eng" ~ "engagement",
      metric == "aca" ~ "academic expectations"
    )
  )
```

```
survey_nychs_long <- survey_nychs_long %>%
  mutate(
    respondent = case_when(
      respondent == "_p_" ~ "parent",
      respondent == "_t_" ~ "teacher",
      respondent == "_s_" ~ "student",
      respondent == "_to" ~ "total"
    )
  )
```

Alternatively, if_else() can be used in place of case_when(). Restore data in original long format then use if_else()

```
survey_nychs_long <- survey_nychs %>%
  pivot_longer(
    cols = c(saf_p_11:aca_tot_11),       #stacks the perception columns(survey questions) tog
ether
    names_to = "survey_question",
    values_to = "score"
  )
```

```
survey_nychs_long <- survey_nychs_long %>%
  mutate(
    metric = str_sub(survey_question, 1, 3),       #str_sub() function extracted infor from t
he survey_question variable
    respondent = str_sub(survey_question, 4, 6)    # start & end characters could have been
5, 5 but there are 2 t's for teacher & total
  )
```

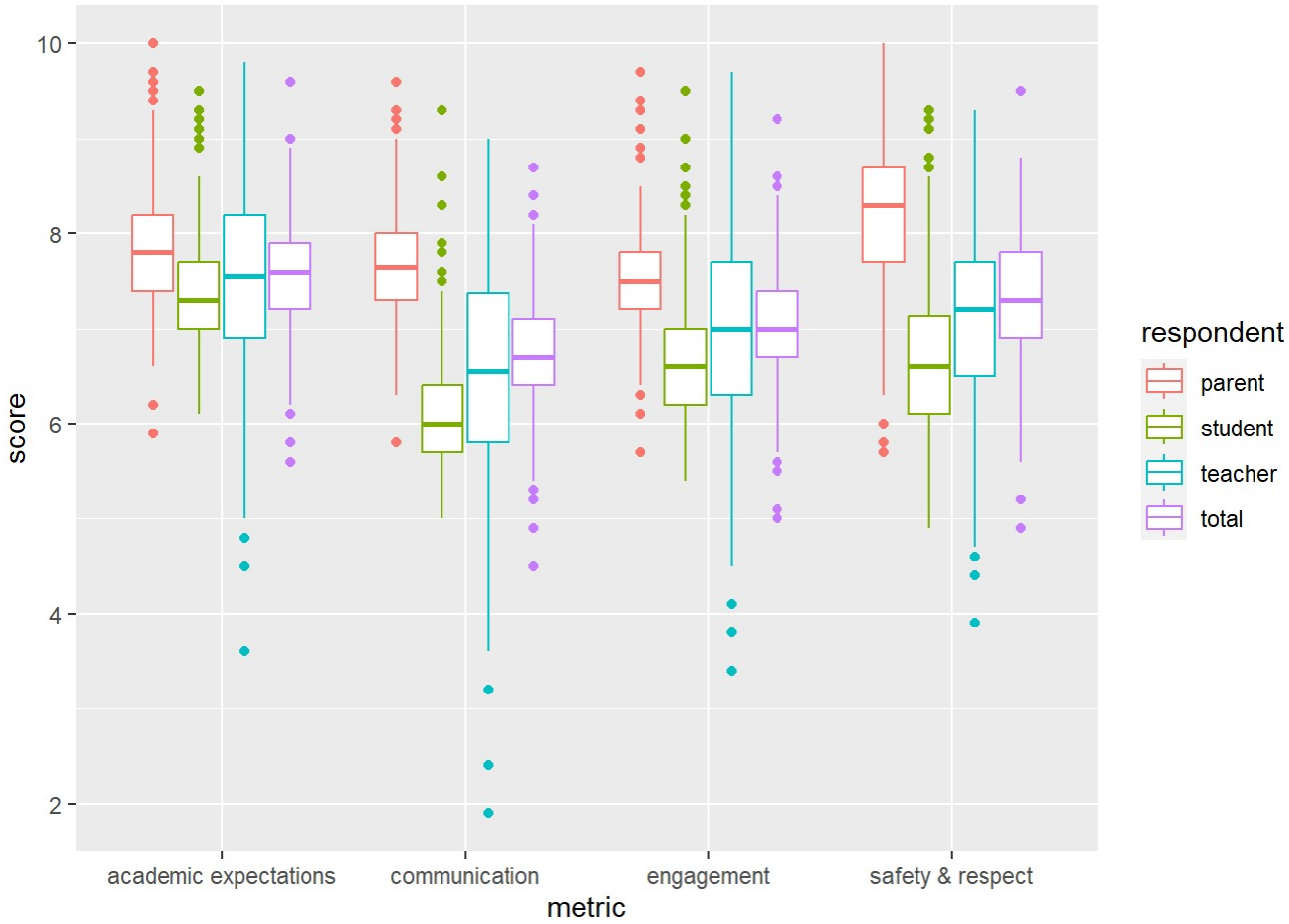
```
survey_nychs_long <- survey_nychs_long %>%
  mutate(metric = ifelse(metric  == "saf", "safety & respect",
                         ifelse(metric == "com", "communication",
                                ifelse(metric == "eng", "engagement",
                                       ifelse(metric == "aca", "academic expectations", "
NA")))))
```

```
survey_nychs_long <- survey_nychs_long %>%
  mutate(respondent = ifelse(respondent  == "_p_", "parent",
                             ifelse(respondent == "_t_", "teacher",
                                    ifelse(respondent == "_s_", "student",
                                           ifelse(respondent == "_to", "total", "NA")))))
```

Now it's manageable to visualize the perceptions

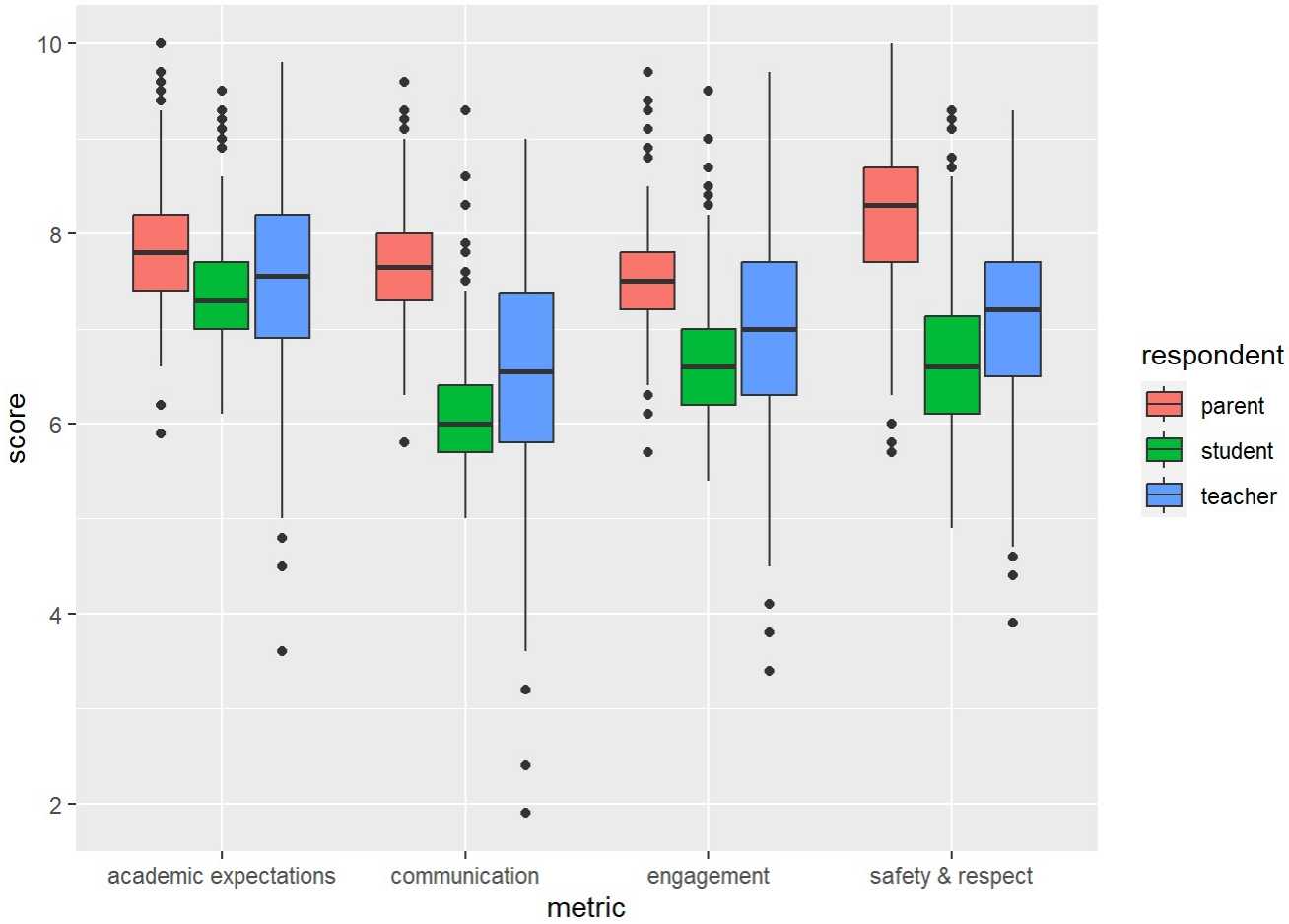
```
ggplot(data = survey_nychs_long,                    #visualizes all responses
       aes(x = metric, y = score, color = respondent)) +
       geom_boxplot()
```

```
## Warning: Removed 1688 rows containing non-finite values (`stat_boxplot()`).
```




```
survey_nychs_long %>%
  filter(respondent != "total") %>%
  ggplot(aes(x = metric, y = score, fill = respondent)) +
  geom_boxplot()
```

```
## Warning: Removed 1268 rows containing non-finite values (`stat_boxplot()`).
```



From the visualization created with code above, it was noted that:

- Scores for teachers were more spread out than those for students and parents
- The larger & smaller scores are equally spread out for all metrics across all groups except safety & respect(among parents) and academic expectations (among students)
- For academic expectations(among students), those that are smaller than medium value are close together.
- For safety & respect(among parents), those that are higher than medium value are closer together.

Other codes

```
HS <- survey_combined %>%   #checking unique entries for highschool column
  pull(highschool) %>%
  unique

print(HS)
```

```
## [1] NA  0
```

```
surveyed <- survey_combined %>%   #checking unique entries for studentssurveyed variable
  pull(studentssurveyed) %>%
  unique

print(surveyed)
```

```
## [1] "Yes" "No"
```

Investigate those which were not surveyed

```
not_surveyed <- survey_combined %>%
  filter(studentssurveyed == "No")

View(not_surveyed)  #Just 1 obs.
```