

# Lab-04

Negar

2/10/2022

```
dictionary <- readr::read_csv(here::here("data", "green_dictionary.csv"))

## Rows: 36 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr (3): Item, Content, Options
## dbl (1): Keying
##
## 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.
green_data <- readr::read_csv(here::here("data", "green_data.csv"))

## Rows: 373 Columns: 37
## -- Column specification -----
## Delimiter: ","
## chr (1): id
## dbl (36): green1, green2, green3, green4, green5, comp1, comp2, comp3, comp4...
##
## 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.
```

## Question 1

```
dictionary |>
  summarize(dictionary)

## # A tibble: 36 x 4
##   Item   Content                                Options      Keying
##   <chr> <chr>                                <chr>      <dbl>
## 1 green1 Overall, I am regarded as an environmentally c~ 1-5; SD, D, NA~      1
## 2 green2 I am not particularly known for protecting the~ 1-5; SD, D, NA~     -1
## 3 green3 My friends know I enjoy nature.                1-5; SD, D, NA~      1
## 4 green4 I have a reputation for living in harmony with~ 1-5; SD, D, NA~      1
## 5 green5 Among people I know, I seem to be the one most~ 1-5; SD, D, NA~      1
## 6 comp1  I feel others' emotions.                        1-5; SD, D, NA~      1
## 7 comp2  I inquire about others' well-being.              1-5; SD, D, NA~      1
## 8 comp3  I sympathize with others' feelings.              1-5; SD, D, NA~      1
## 9 comp4  I take an interest in other people's lives.       1-5; SD, D, NA~      1
## 10 comp5 I like to do things for others.                  1-5; SD, D, NA~      1
## # ... with 26 more rows

green_data_changed <- green_data |>
  mutate(
```

```

    across(
      ~id,
      \ (x) na_if(x, -99)
    )
  )
reversed <- dictionary |>
  filter(Keying == -1) |>
  pull(Item)
green_data_reverse <- green_data_changed |>
  mutate(across(all_of(reversed),
    \ (x) recode(x, "1" = 5, "2" = 4, "3" = 3, "4" = 2, "5" = 1),
    .names = "{.col}_r"))
green_data_changed$student <- recode_factor(green_data_changed$student, '1' = "Non-student", '2' = "Student")

```

## Question 2

```

green_data_changed <- green_data_changed |>
  rowwise() |>
  mutate(
    Green_total = mean(c_across(green1:green5), na.rm = TRUE),
    Comp_total = mean(c_across(comp1:comp10), na.rm = TRUE),
    Intel_total = mean(c_across(intel1:intel10), na.rm = TRUE),
    Open_total = mean(c_across(open1:open10), na.rm = TRUE),
    .after = id
  ) |>
  ungroup()

```

## Question 3

```

green_data_pomp <- green_data_changed |>
  mutate(across(Green_total:Open_total,
    \ (x) datawizard::change_scale(x, range = c(1, 5), to = c(0, 100)),
    .names = "{.col}_pomp"), .after = Open_total)
green_data_pomp

```

```

## # A tibble: 373 x 45
##   id      Green_total Comp_total Intel_total Open_total Green_total_pomp
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 9099         3.6         2.7         2.4         3.3         65
## 2 6275         3         2.8         3.5         3.9         50
## 3 8116         3.6         3.2         3.2         3.3         65
## 4 8586         3.8         3         3.2         2.9         70
## 5 0406         3.4         3         3.1         3.3         60
## 6 5645         3.6         3.4         3         3.2         65
## 7 3788         1         3         3         3         0
## 8 8424         2.6         2.5         3.1         2.8         40
## 9 8450         3.2         2.8         2.8         3.44        55
## 10 0512         4         2.8         3.1         3         75
## # ... with 363 more rows, and 39 more variables: Comp_total_pomp <dbl>,
## #   Intel_total_pomp <dbl>, Open_total_pomp <dbl>, green1 <dbl>, green2 <dbl>,
## #   green3 <dbl>, green4 <dbl>, green5 <dbl>, comp1 <dbl>, comp2 <dbl>,
## #   comp3 <dbl>, comp4 <dbl>, comp5 <dbl>, comp6 <dbl>, comp7 <dbl>,

```

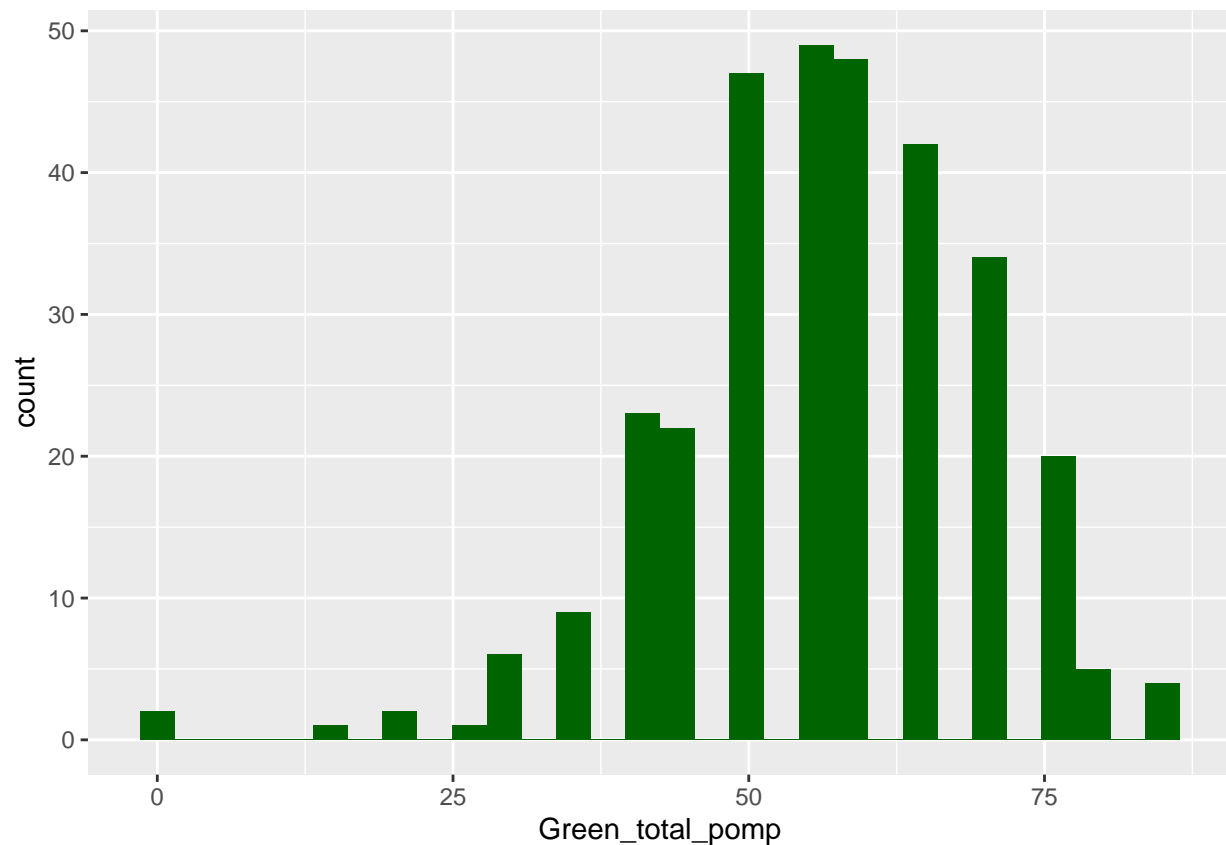
```
## #   comp8 <dbl>, comp9 <dbl>, comp10 <dbl>, intel1 <dbl>, intel2 <dbl>,
## #   intel3 <dbl>, intel4 <dbl>, intel5 <dbl>, intel6 <dbl>, intel7 <dbl>,
## #   intel8 <dbl>, intel9 <dbl>, intel10 <dbl>, open1 <dbl>, open2 <dbl>, ...
```

#### Question 4

```
Avg_green_data_pomp <- green_data_pomp
Avg_green_data_pomp$student <- recode_factor(Avg_green_data_pomp$student, '1' = "Non-student", '2' = "S

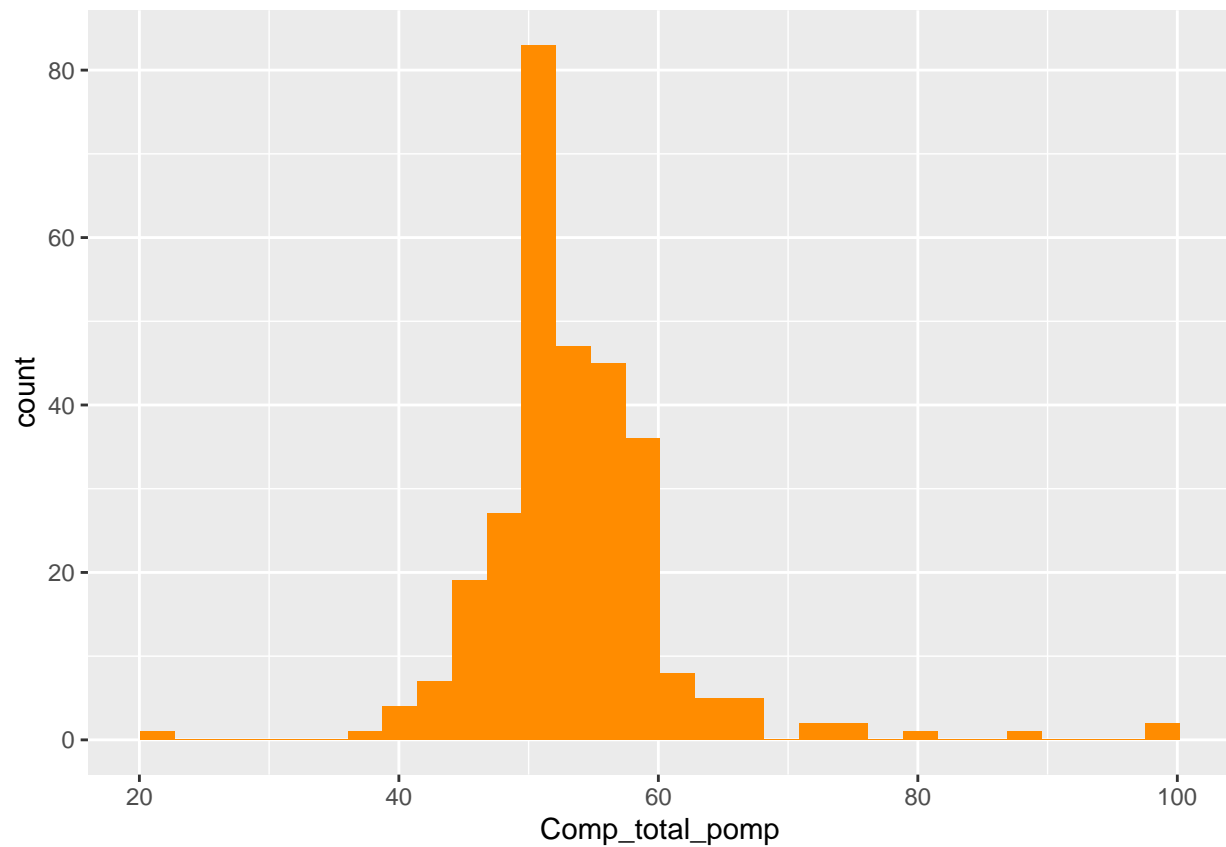
Avg_green_data_pomp |>
ggplot() +
aes(x = Green_total_pomp) +
  geom_histogram(fill = "darkgreen")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 58 rows containing non-finite values (stat_bin).
```



```
Avg_green_data_pomp |>
ggplot() +
aes(x = Comp_total_pomp) +
  geom_histogram(fill = "darkorange")
```

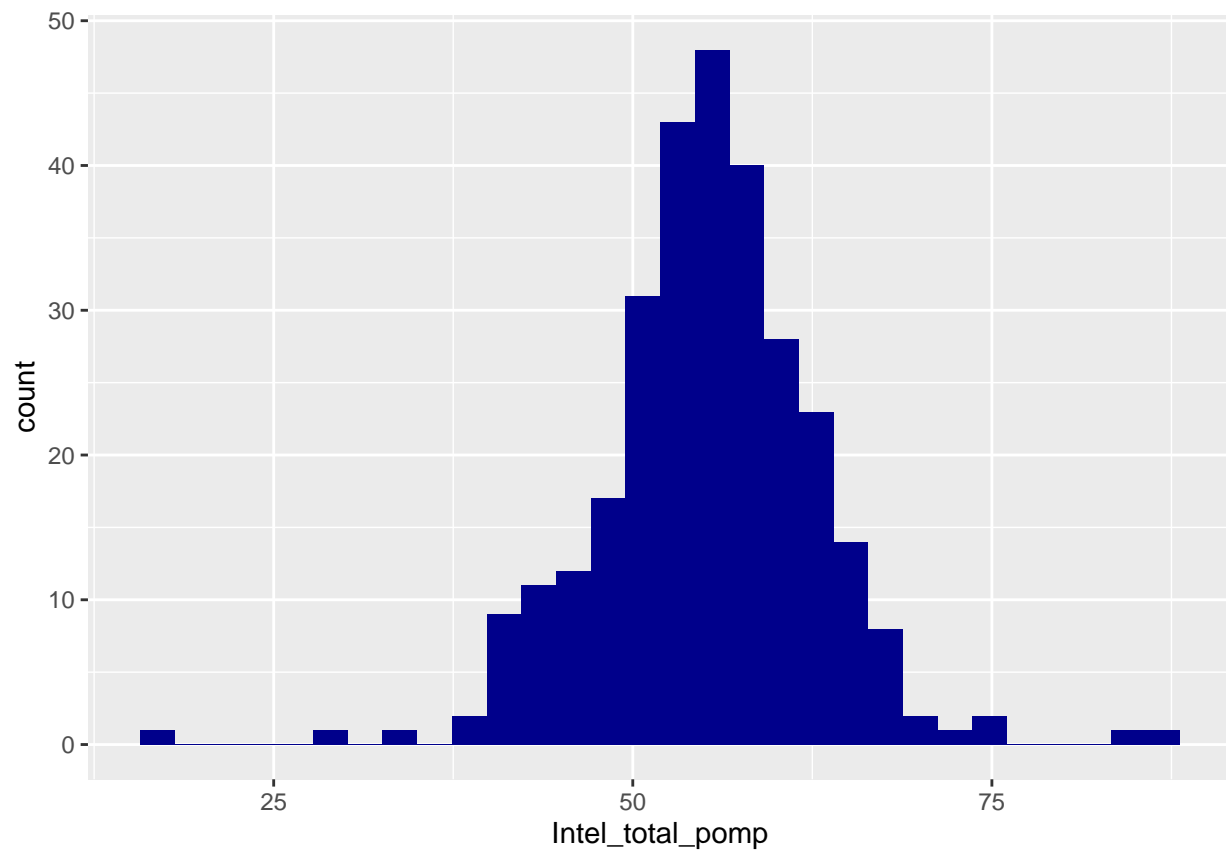
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



```
Avg_green_data_pomp |>  
ggplot() +  
aes(x = Intel_total_pomp) +  
  geom_histogram(fill = "darkblue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

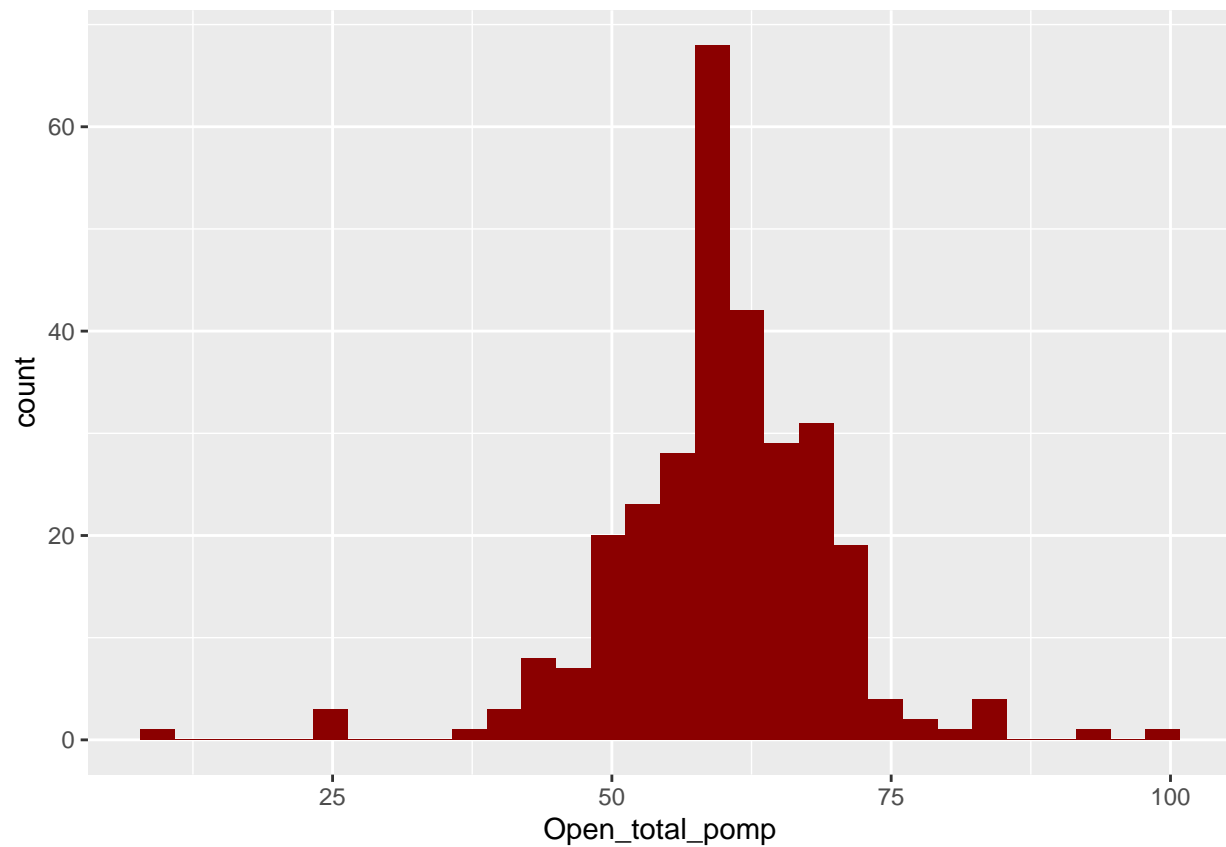
```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



```
Avg_green_data_pomp |>  
ggplot() +  
aes(x = Open_total_pomp) +  
  geom_histogram(fill = "darkred")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

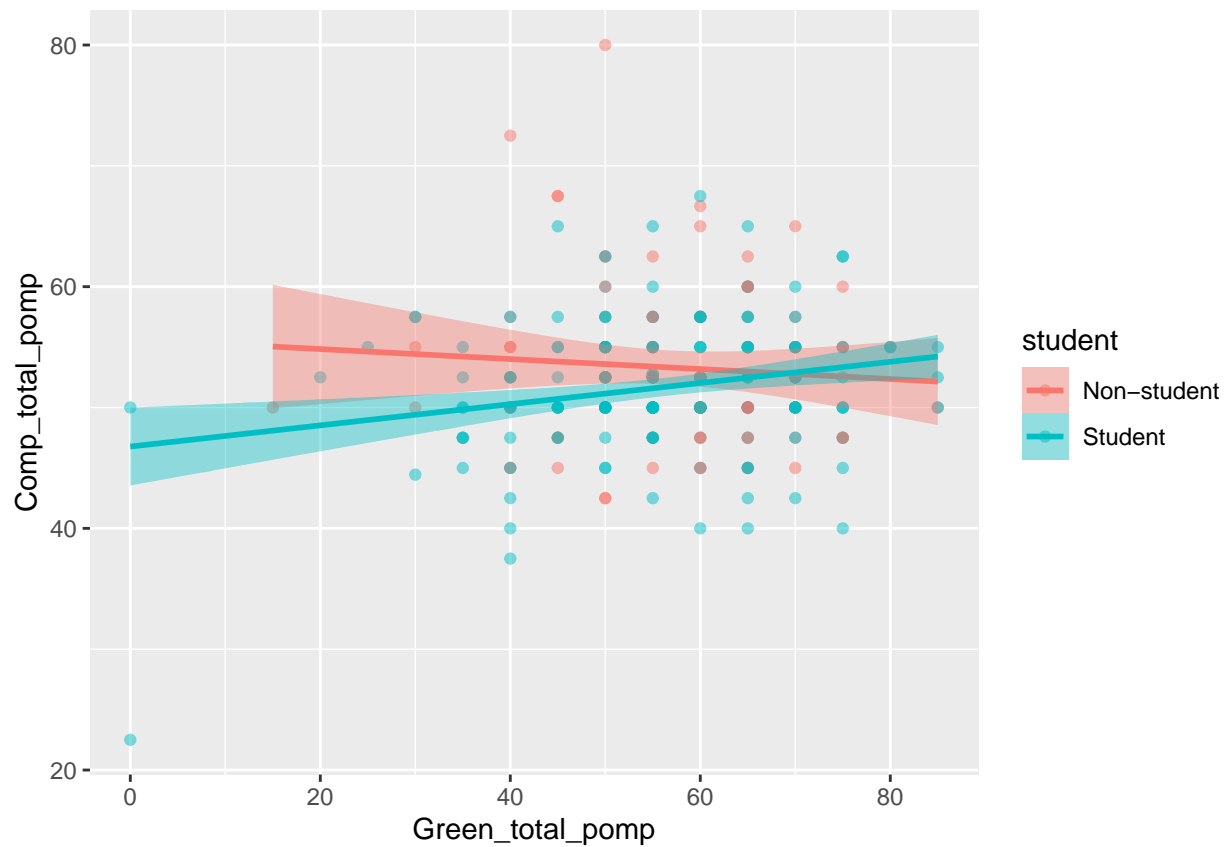
```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



### Question 5

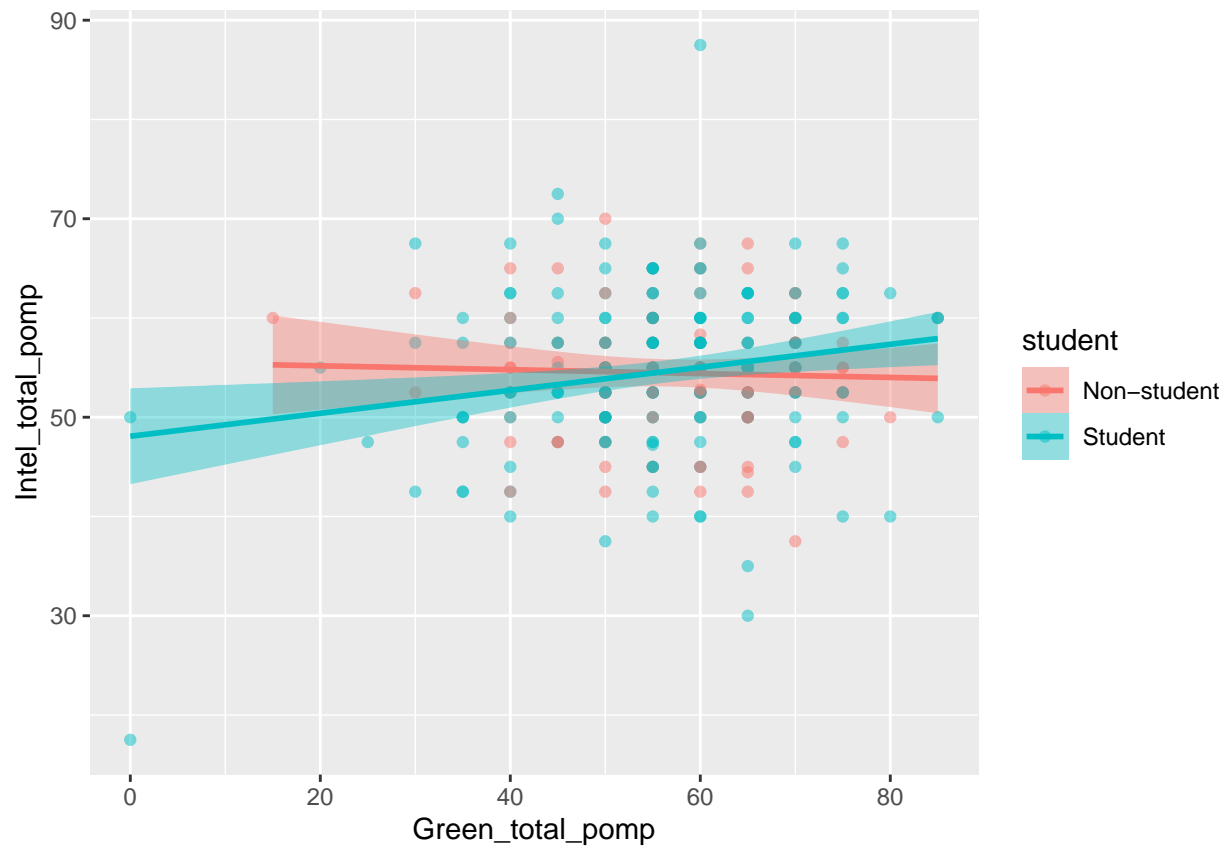
```
Avg_green_data_pomp |>
  filter(!is.na(student)) |>
  ggplot() +
  aes(x = Green_total_pomp, y = Comp_total_pomp,
      group = student,
      color = student,
      fill = student) +
  geom_point(alpha = .5) +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
Avg_green_data_pomp |>
  filter(!is.na(student)) |>
  ggplot() +
    aes(x = Green_total_pomp, y = Intel_total_pomp,
        group = student,
        color = student,
        fill = student) +
    geom_point(alpha = .5) +
    geom_smooth(method = "lm")
```

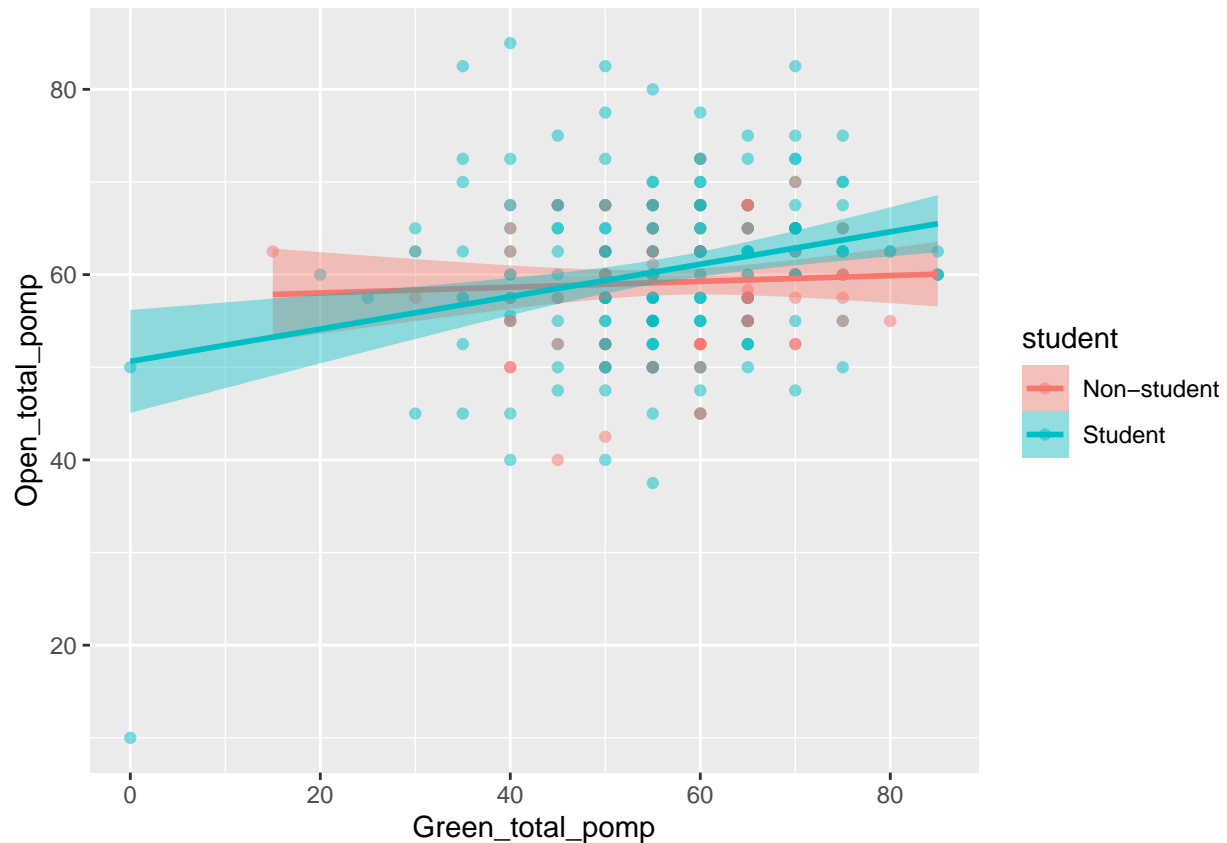
```
## `geom_smooth()` using formula 'y ~ x'
```



```
Avg_green_data_pomp |>
  filter(!is.na(student)) |>
  ggplot() +
  aes(x = Green_total_pomp, y = Open_total_pomp,
      group = student,
      color = student,
      fill = student) +
  geom_point(alpha = .5) +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

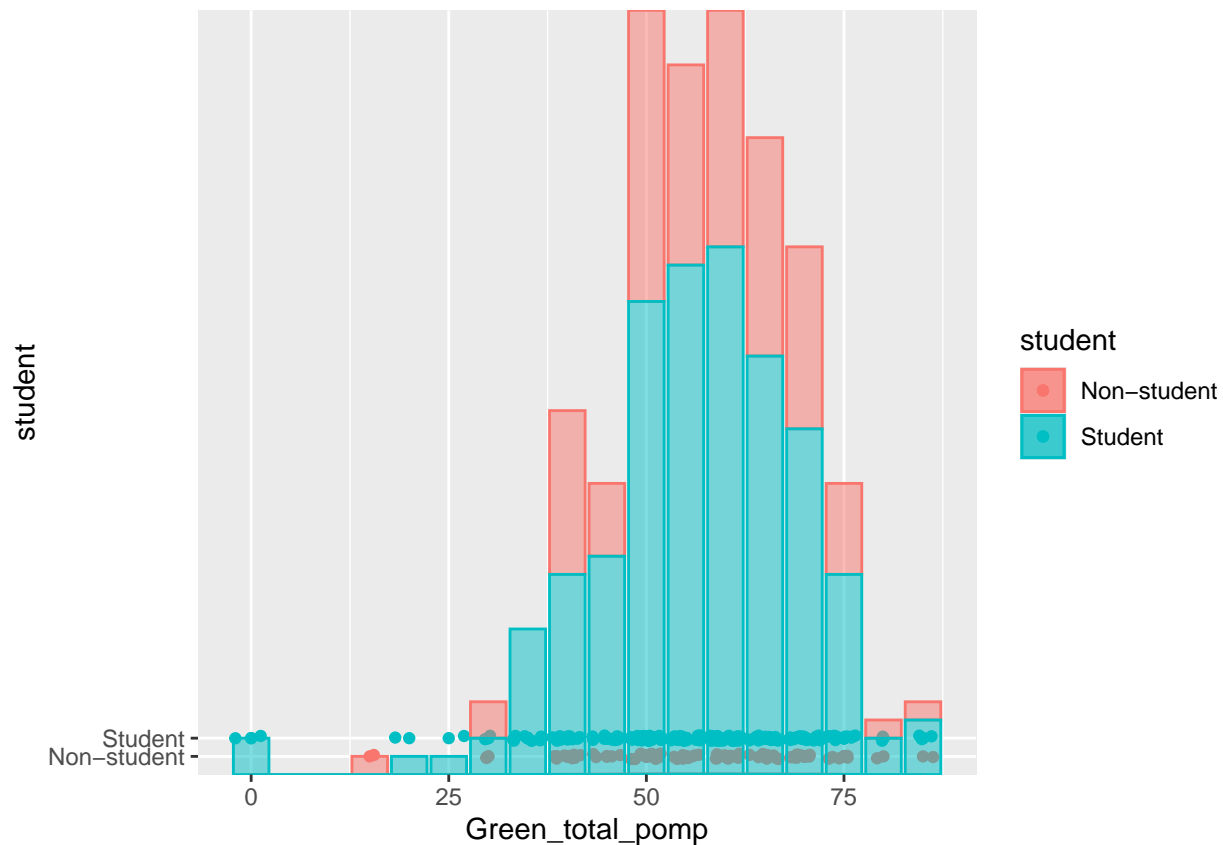




In all three plots, the non-student group showed higher scores at first, and then the scores for the student group went higher than the non-student group.

### Question 6

```
ggplot(na.omit(Avg_green_data_pomp)) +
  geom_jitter(aes(x = Green_total_pomp, y = student,
    fill = student,
    color = student), height = .15) +
  geom_point(aes(x = Green_total_pomp, y = student,
    fill = student,
    color = student)) +
  geom_bar(aes(x = Green_total_pomp,
    fill = student,
    color = student), alpha = 0.5) +
  geom_density(aes(x = Green_total_pomp,
    fill = student,
    color = student), alpha = 0.5)
```



Overall, non-student group had higher environmental awareness scores compared to student group.

## Question 7

```
table <- Avg_green_data_pomp |>
  na.omit() |>
  group_by(student) |>
  summarize(across(c(Green_total_pomp, Comp_total_pomp, Intel_total_pomp, Open_total_pomp),
    list(
      Mean = ~ mean(.x, na.rm = T),
      SDs = ~ sd(.x, na.rm = T),
      Medians = ~ median(.x, na.rm = T),
      Minima = ~ min(.x, na.rm = T),
      Maxima = ~ max(.x, na.rm = T)
    )))
table
```

```
## # A tibble: 2 x 21
##   student Green_total_pomp~ Green_total_pomp~ Green_total_pomp~ Green_total_pomp~
##   <fct>      <dbl>          <dbl>          <dbl>          <dbl>
## 1 Non-stu~      56.7          12.3           55            15
## 2 Student      56.2          13.5           55             0
## # ... with 16 more variables: Green_total_pomp_Maxima <dbl>,
## #   Comp_total_pomp_Mean <dbl>, Comp_total_pomp_SDs <dbl>,
## #   Comp_total_pomp_Medians <dbl>, Comp_total_pomp_Minima <dbl>,
## #   Comp_total_pomp_Maxima <dbl>, Intel_total_pomp_Mean <dbl>,
## #   Intel_total_pomp_SDs <dbl>, Intel_total_pomp_Medians <dbl>,
```

```
## # Intel_total_pomp_Minima <dbl>, Intel_total_pomp_Maxima <dbl>,  
## # Open_total_pomp_Mean <dbl>, Open_total_pomp_SDs <dbl>, ...
```