Assessment 07 - Summarizing with dplyr

Gabriele Mineo - Harvard Data Science Professional

National Center for Health Statistics

To practice our dplyr skills we will be working with data from the survey collected by the United States National Center for Health Statistics (NCHS). This center has conducted a series of health and nutrition surveys since the 1960's.

Starting in 1999, about 5,000 individuals of all ages have been interviewed every year and then they complete the health examination component of the survey. Part of this dataset is made available via the NHANES package which can be loaded this way:

```
library(NHANES)
data(NHANES)
```

The NHANES data has many missing values. Remember that the main summarization function in R will return NA if any of the entries of the input vector is an NA. Here is an example:

```
library(dslabs)
data(na_example)
mean(na_example)
```

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

```
sd(na_example)
```

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

To ignore the NAs, we can use the na.rm argument:

```
mean(na_example, na.rm = TRUE)
```

```
## [1] 2.301754
```

```
sd(na_example, na.rm = TRUE)
```

```
## [1] 1.22338
```

Try running this code, then let us know you are ready to proceed with the analysis.

Blood pressure 1

Let's explore the NHANES data. We will be exploring blood pressure in this dataset.

First let's select a group to set the standard. We will use `20-29` year old females. Note that the category is coded with 20-29, with a space in front of the `20`! The `AgeDecade` is a categorical variable with these ages.

To know if someone is female, you can look at the `Gender` variable.

Instructions

- Filter the `NHANES` dataset so that only 20-29 year old females are included and assign this new data frame to the object `tab`.
- Use the pipe to apply the function `filter`, with the appropriate logicals, to `NHANES`.
- Remember that this age group is coded with `20-29`, which includes a space. You can use `head` to explore the `NHANES` table to construct the correct call to filter.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(NHANES)
data(NHANES)
tab <- NHANES %>% filter(AgeDecade == " 20-29" & Gender == "female")
```

Blood pressure 2

Now we will compute the average and standard deviation for the subgroup we defined in the previous exercise (20-29 year old females), which we will use reference for what is typical.

You will determine the average and standard deviation of systolic blood pressure, which are stored in the BPSysAve variable in the NHANES dataset.

Instructions

- Complete the line of code to save the average and standard deviation of systolic blood pressure as `average` and `standard_deviation` to a variable called `ref`.
- Use the `summarize` function after filtering for 20-29 year old females and connect the results using the pipe `%>%`. When doing this remember there are `NA`s in the data!

```
library(dplyr)
library(NHANES)
data(NHANES)
ref <- NHANES %>%
  filter(AgeDecade == " 20-29" & Gender == "female") %>%
  summarize(average = mean(BPSysAve, na.rm = TRUE),
            standard_deviation = sd(BPSysAve, na.rm=TRUE))
ref
```

```
## # A tibble: 1 x 2
##   average standard_deviation
##     <dbl>              <dbl>
## 1    108.               10.1
```

Summarizing averages

Now we will repeat the exercise and generate only the average blood pressure for 20-29 year old females. For this exercise, you should review how to use the place holder . in dplyr.

Instructions

- Modify the line of sample code to assign the average to a numeric variable called `ref_avg`.

```
library(dplyr)
library(NHANES)
```

```
data(NHANES)
ref_avg <- NHANES %>%
      filter(AgeDecade == " 20-29" & Gender == "female") %>%
      summarize(average = mean(BPSysAve, na.rm = TRUE)) %>%
      .$average
```

Min and max

Let's continue practicing by calculating two other data summaries: the minimum and the maximum.

Again we will do it for the `BPSysAve` variable and the group of 20-29 year old females.

Instructions

- Report the min and max values for the same group as in the previous exercises.
- Use `filter` and `summarize` connected by the pipe `%>%` again. The functions `min` and `max` can be used to get the values you want.
- Within `summarize`, save the min and max of systolic blood pressure as `min` and `max`.

```
library(NHANES)
data(NHANES)
NHANES %>%
      filter(AgeDecade == " 20-29"  & Gender == "female") %>%
      summarize(min = min(BPSysAve, na.rm = TRUE),
                max = max(BPSysAve, na.rm = TRUE))
```

```
## # A tibble: 1 x 2
##     min   max
##   <dbl> <dbl>
## 1    84   179
```

group_by

Now let's practice using the `group_by` function.

What we are about to do is a very common operation in data science: you will split a data table into groups and then compute summary statistics for each group.

We will compute the average and standard deviation of systolic blood pressure for females for each age group separately. Remember that the age groups are contained in `AgeDecade`.

Instructions

- Use the functions `filter, group_by, summarize`, and the pipe `%>%` to compute the average and standard deviation of systolic blood pressure for females for each age group separately.
- Within `summarize`, save the average and standard deviation of systolic blood pressure (`BPSysAve`) as `average` and `standard_deviation`.

```
library(dplyr)
library(NHANES)
data(NHANES)
NHANES %>%
      filter(Gender == "female") %>%
      group_by(AgeDecade) %>%
      summarize(average = mean(BPSysAve, na.rm = TRUE),
                standard_deviation = sd(BPSysAve, na.rm=TRUE))
```

```
## # A tibble: 9 x 3
##   AgeDecade average standard_deviation
##   <fct>       <dbl>              <dbl>
## 1 " 0-9"      100.0               9.07
## 2 " 10-19"    104.                9.46
## 3 " 20-29"    108.               10.1
## 4 " 30-39"    111.               12.3
## 5 " 40-49"    115.               14.5
## 6 " 50-59"    122.               16.2
## 7 " 60-69"    127.               17.1
## 8 " 70+"      134.               19.8
## 9 <NA>        142.               22.9
```

group_by example 2

Now let's practice using `group_by` some more. We are going to repeat the previous exercise of calculating the average and standard deviation of systolic blood pressure, but for males instead of females.

This time we will not provide much sample code. You are on your own!

Instructions

Calculate the average and standard deviation of systolic blood pressure for males for each age group separately using the same methods as in the previous exercise.

```
library(dplyr)
library(NHANES)
data(NHANES)
NHANES %>%
      filter(Gender == "male") %>%
      group_by(AgeDecade) %>%
      summarize(average = mean(BPSysAve, na.rm = TRUE),
                standard_deviation = sd(BPSysAve, na.rm=TRUE))
```

```
## # A tibble: 9 x 3
##   AgeDecade average standard_deviation
##   <fct>       <dbl>              <dbl>
## 1 " 0-9"       97.4               8.32
## 2 " 10-19"    110.               11.2
## 3 " 20-29"    118.               11.3
## 4 " 30-39"    119.               12.3
## 5 " 40-49"    121.               14.0
## 6 " 50-59"    126.               17.8
## 7 " 60-69"    127.               17.5
## 8 " 70+"      130.               18.7
## 9 <NA>        136.               23.5
```

group_by example 3

We can actually combine both of these summaries into a single line of code. This is because `group_by` permits us to group by more than one variable.

We can use `group_by(AgeDecade, Gender)` to group by both age decades and gender.

Instructions

- Create a single summary table for the average and standard deviation of systolic blood pressure using `group_by(AgeDecade, Gender)`.
- Note that we no longer have to `filter`!
- Your code within `summarize` should remain the same as in the previous exercises.

```
library(dplyr)
library(NHANES)
data(NHANES)
NHANES %>%
      group_by(AgeDecade, Gender) %>%
      summarize(average = mean(BPSysAve, na.rm = TRUE),
                standard_deviation = sd(BPSysAve, na.rm=TRUE))
```

```
## # A tibble: 18 x 4
## # Groups:   AgeDecade [?]
##    AgeDecade Gender average standard_deviation
##    <fct>     <fct>    <dbl>              <dbl>
##  1 " 0-9"    female   100.0               9.07
##  2 " 0-9"    male      97.4               8.32
##  3 " 10-19"  female   104.                9.46
##  4 " 10-19"  male     110.               11.2
##  5 " 20-29"  female   108.               10.1
##  6 " 20-29"  male     118.               11.3
##  7 " 30-39"  female   111.               12.3
##  8 " 30-39"  male     119.               12.3
##  9 " 40-49"  female   115.               14.5
## 10 " 40-49"  male     121.               14.0
## 11 " 50-59"  female   122.               16.2
## 12 " 50-59"  male     126.               17.8
## 13 " 60-69"  female   127.               17.1
## 14 " 60-69"  male     127.               17.5
## 15 " 70+"    female   134.               19.8
## 16 " 70+"    male     130.               18.7
## 17 <NA>      female   142.               22.9
## 18 <NA>      male     136.               23.5
```

Arrange

Now we are going to explore differences in systolic blood pressure across races, as reported in the `Race1` variable.

We will learn to use the `arrange` function to order the outcome acording to one variable.

Note that this function can be used to order any table by a given outcome. Here is an example that arranges by systolic blood pressure.

`NHANES %>% arrange(BPSysAve)`

If we want it in descending order we can use the `desc` function like this:

`NHANES %>% arrange(desc(BPSysAve))`

In this example, we will compare systolic blood pressure across values of the `Race1` variable for males between the ages of 40-49.

Instructions

- Compute the average and standard deviation for each value of `Race1` for males in the age decade 40-49.

- Order the resulting table from lowest to highest average systolic blood pressure.
- Use the functions `filter, group_by, summarize, arrange`, and the pipe `%>%` to do this in one line of code.
- Within summarize, save the average and standard deviation of systolic blood pressure as `average` and `standard_deviation`.

```
library(dplyr)
library(NHANES)
data(NHANES)
NHANES %>%
      filter(Gender == "male" & AgeDecade==" 40-49") %>%
      group_by(Race1) %>%
      summarize(average = mean(BPSysAve, na.rm = TRUE),
                standard_deviation = sd(BPSysAve, na.rm=TRUE)) %>%
      arrange(average)
```

```
## # A tibble: 5 x 3
##   Race1    average standard_deviation
##   <fct>      <dbl>              <dbl>
## 1 White       120.               13.4
## 2 Other       120.               16.2
## 3 Hispanic    122.               11.1
## 4 Mexican     122.               13.9
## 5 Black       126.               17.1
```