STAT 33A Workbook 7

CJ HINES (3034590053)

Oct 15, 2020

This workbook is due **Oct 15, 2020** by 11:59pm PT.

The workbook is organized into sections that correspond to the lecture videos for the week. Watch a video, then do the corresponding exercises *before* moving on to the next video.

Workbooks are graded for completeness, so as long as you make a clear effort to solve each problem, you'll get full credit. That said, make sure you understand the concepts here, because they're likely to reappear in homeworks, quizzes, and later lectures.

As you work, write your answers in this notebook. Answer questions with complete sentences, and put code in code chunks. You can make as many new code chunks as you like.

In the notebook, you can run the line of code where the cursor is by pressing `Ctrl` + `Enter` on Windows or `Cmd` + `Enter` on Mac OS X. You can run an entire code chunk by clicking on the green arrow in the upper right corner of the code chunk.

Please do not delete the exercises already in this notebook, because it may interfere with our grading tools.

You need to submit your work in two places:

- Submit this Rmd file with your edits on bCourses.
- Knit and submit the generated PDF file on Gradescope.

If you have any last-minute trouble knitting, **DON'T PANIC**. Submit your Rmd file on time and follow up in office hours or on Piazza to sort out the PDF.

dplyr Overview

Watch the "dplyr Overview" lecture video.

No exercises for this section.

Subsets with dplyr

Watch the "Subsets with dplyr" lecture video.

Exercise 1

Use dplyr and the dogs data to compute each of the following subsets:

1. Rows 10-30 only

2. All rows except row 51
3. All columns except `popularity_all` and `popularity`
4. Rows 1-10 with only the `breed`, `weight`, and `height` columns

You do not need to print out these subsets, just show us the code to compute them.

YOUR ANSWER GOES HERE:

```
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
```

```
dogs = readRDS("dogs.rds")
```

1. Rows 10-30 only

```
#slice(dogs, seq(10,30))
tentothirty = slice(dogs, 10:30)
```

2. All rows except row 51

```
no51 = slice(dogs, -51)
```

3. All columns except `popularity_all` and `popularity`

```
nopop = select(dogs, -popularity_all, -popularity)
```

4. Rows 1-10 with only the `breed`, `weight`, and `height` columns

```
dogsbwh = select(dogs, breed, weight, height)
sldogsbwh = slice(dogsbwh, 1:10)
```

Exercise 2

Use dplyr to show that there are no duplicated rows in the dogs data.

Explain your reasoning.

YOUR ANSWER GOES HERE:

```
howmany = nrow(dogs)
distincthowmany = nrow(distinct(dogs))
howmany == distincthowmany
```

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

`howmany` are the number of rows in dogs. `distincthowmany` are the number of distinct (or not duplicated) rows in dogs. The numbers are equal.

Base R versus dplyr

Watch the "Base R versus dplyr" lecture video.

No exercises for this section.

Transformations with dplyr

Watch the "Transformations with dplyr" lecture video.

Exercise 3

Workbook 4, Exericse 6 asked you to use base R and the dogs data to compute:

1. The mean and median of the longevity column (ignoring missing values).
2. The subset that contains rows 10-20 of the height, weight, and longevity columns.
3. The number of dog breeds with weight greater than 42.
4. The subset of large dogs that require daily grooming.

For each of these, show the code to compute the result:

1. Using base R
2. Using dplyr

YOUR ANSWER GOES HERE:

BASE R

1. The mean and median of the longevity column (ignoring missing values).

```
mean(dogs$longevity, na.rm = TRUE)
```

```
## [1] 10.95674
```

```
median(dogs$longevity, na.rm = TRUE)
```

```
## [1] 11.29
```

2. The subset that contains rows 10-20 of the height, weight, and longevity columns.

```
rh = dogs$height[10:20]
rw = dogs$weight[10:20]
rl = dogs$longevity[10:20]
```

3. The number of dog breeds with weight greater than 42.

```
heavy = subset(dogs, weight > 42)
length(heavy$breed)
```

```
## [1] 37
```

4. The subset of large dogs that require daily grooming.

```
biggroom = subset(dogs, size == "large" & grooming == "daily")
```

DPLYR

1. The mean and median of the longevity column (ignoring missing values).

```
summarize(dogs, mean(longevity, na.rm = TRUE))
```

```
##   mean(longevity, na.rm = TRUE)
## 1                      10.95674
```

```
summarize(dogs, median(longevity, na.rm = TRUE))
```

```
##   median(longevity, na.rm = TRUE)
## 1                           11.29
```

2. The subset that contains rows 10-20 of the height, weight, and longevity columns.

```
tentwenty = slice(dogs, 10:20)
select(tentwenty, height, weight, longevity)
```

```
##    height weight longevity
## 1   14.50   22.0     12.53
## 2   21.75   47.5     12.58
## 3   10.50   15.0     13.92
## 4   10.25     NA     11.42
## 5      NA   24.0     12.63
## 6   13.00   15.5     11.81
## 7    5.00    5.5     16.50
## 8   10.50     NA     11.05
## 9   20.00     NA     12.87
## 10  19.50   45.0     12.54
## 11  10.50     NA     12.80
```

3. The number of dog breeds with weight greater than 42.

```
count(filter(dogs, weight > 42))
```

```
##    n
## 1 37
```

4. The subset of large dogs that require daily grooming.

```
filt = filter(dogs, size == "large", grooming == "daily")
```

Exercise 4

Use dplyr and the dogs data to determine which 3 dogs cost the most.

Your answer to this exercise should be a data frame with 3 rows.

YOUR ANSWER GOES HERE:

```
dogs = readRDS("dogs.rds")
filter(dogs, lifetime_cost > 25600)
```

```
##                        breed    group datadog popularity_all popularity
## 1                  Chihuahua      toy    3.15             14         14
## 2 German Shorthaired Pointer sporting    3.03             15         15
## 3            Giant Schnauzer  working    2.38             95         70
##   lifetime_cost intelligence_rank longevity ailments price food_cost grooming
## 1         26250                67     16.50        1   588       324   weekly
## 2         25842                17     11.46        1   545       971   weekly
## 3         26686                28     10.00        1   810      1349    daily
##     kids megarank_kids megarank  size weight height
## 1    low            16       55 small    5.5    5.0
## 2   high            23       12 large   62.5   24.0
## 3 medium            62       67 large   77.5   25.5
```

Exercise 5

Use dplyr to answer each of the following:

1. On average, which `group` of dog has the highest lifetime cost? Which has the lowest?
2. How many dogs are there for each possible combination of `size` and `grooming`?
3. For each `group` of dog, what's the shortest lifespan? You should have one result per group here. For each `group` of dog, what's the longest lifespan?
4. Do popular dogs tend to be more expensive? Use any columns that seem appropriate; you can also use ggplot2 if you like.

YOUR ANSWER GOES HERE:

1. On average, which `group` of dog has the highest lifetime cost? Which has the lowest? On average, herding dogs have the highgest lifetime costs while working dogs have the lowest.

```
bygroup = group_by(dogs, group)
summarize(bygroup, mean(lifetime_cost, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 7 x 2
##   group        `mean(lifetime_cost, na.rm = TRUE)`
##   <fct>                                      <dbl>
## 1 herding                                   20692.
## 2 hound                                     19366.
## 3 non-sporting                              19316.
## 4 sporting                                  20299.
## 5 terrier                                   20504.
## 6 toy                                       19506.
## 7 working                                   19165.
```

2. How many dogs are there for each possible combination of `size` and `grooming`?

```
count(dogs, size, grooming)
```

```
##      size grooming  n
## 1   large    daily  6
## 2   large   weekly 30
## 3   large     <NA> 18
## 4  medium    daily  8
## 5  medium   weekly 29
## 6  medium  monthly  1
## 7  medium     <NA> 22
## 8   small    daily  9
## 9   small   weekly 29
## 10  small     <NA> 20
```

3. For each `group` of dog, what's the shortest lifespan? You should have one result per group here. For each `group` of dog, what's the longest lifespan?

Shortest:

```
summarize(bygroup, min(longevity, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 7 x 2
##   group        `min(longevity, na.rm = TRUE)`
##   <fct>                                 <dbl>
## 1 herding                                7.33
## 2 hound                                  6.75
## 3 non-sporting                           6.29
## 4 sporting                               6.5
## 5 terrier                                6.6
## 6 toy                                    9.25
## 7 working                                6.5
```

Longest:

```
summarize(bygroup, max(longevity, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

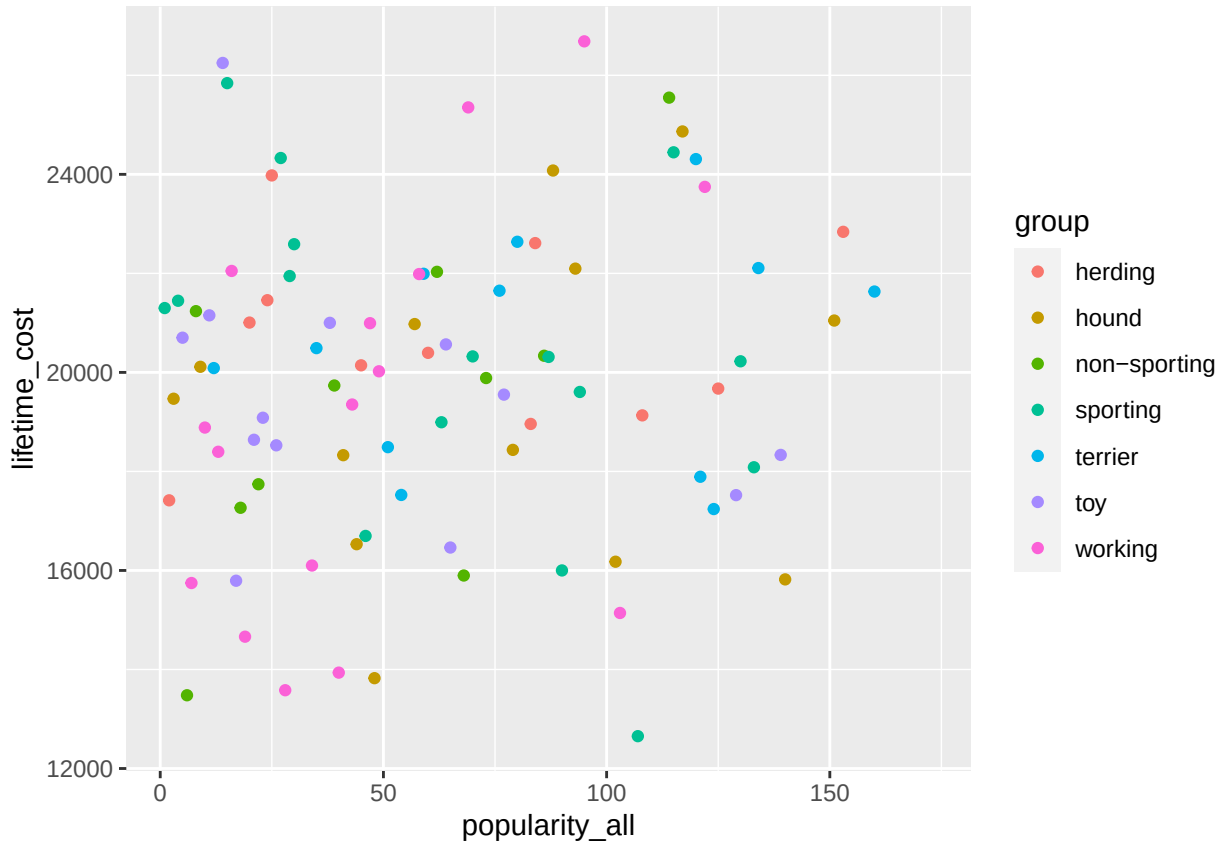
```
## # A tibble: 7 x 2
##   group        `max(longevity, na.rm = TRUE)`
##   <fct>                                 <dbl>
## 1 herding                                14.7
## 2 hound                                  13.6
## 3 non-sporting                           14.4
## 4 sporting                               12.9
## 5 terrier                                14
## 6 toy                                    16.5
## 7 working                                12.6
```

4. Do popular dogs tend to be more expensive? Use any columns that seem appropriate; you can also use ggplot2 if you like.

   No, popularity is not related to lifetime_cost.

```
library(ggplot2)
ggplot(dogs, aes(popularity_all, lifetime_cost, color = group)) + geom_point()
```

```
## Warning: Removed 81 rows containing missing values (geom_point).
```



The Pipe Operator

Watch the "The Pipe Operator" lecture video.

Exercise 6

Read the documentation for the pipe operator at https://magrittr.tidyverse.org/reference/pipe.html.

How do you pass the left-hand operand as the *second* argument to the right-hand side?

Give an example of computing the logarithm of 3, base 10, where you use the pipe to pass `10` as the argument for `base`.

YOUR ANSWER GOES HERE:

```
library("dplyr")
3 %>% log(10)
```

```
## [1] 0.4771213
```