STAT 33B Workbook 5

YOUR NAME (YOUR SID)

Oct 1, 2020

This workbook is due **Oct 1, 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.

Apply Function Basics

Watch the "Apply Function Basics" lecture video.

For exercises that mention the dogs data, you can use either `dogs.rds` or `dogs_tibble.rds`. Both are on the bCourse.

Exercise 1

1. Suppose you call `sapply()` with a function that returns vectors. What class of object does `sapply()` return if all of the result vectors have the same length?

   For instance, what if the applied function returns a length-3 vector for each element?

   Hint: `rnorm()` and `class()` are examples of functions that return vectors.

2. Again suppose you call `sapply()` with a function that returns vectors. What class of object does `sapply()` return if the result vectors have the different lengths?

3. Suppose you call `sapply()` with a function that returns different types. What happens?
   Hint: `all.equal()` is one function that returns different types.

YOUR ANSWER GOES HERE:

```
dogs = readRDS("dogs.rds")
cols = c("weight", "height", "price")
class(sapply(dogs[cols], rnorm)) #Question1, the class returned is character
```

```
## [1] "matrix" "array"
```

```
class(sapply(dogs[cols], unique)) #Question2, the class returned is list
```

```
## [1] "list"
```

```
#class(sapply(dogs[cols], all.equal())) #Question3,it will throw an error
```

Exercise 2

1. Use `sapply()` and `is.numeric()` to identify all of the numeric columns in the dogs data frame.
2. Use `sapply()` and your result from part 1 to compute the range of every numeric column in the dogs data frame.

YOUR ANSWER GOES HERE:

```
sapply(dogs[cols], is.numeric)
```

```
## weight height  price
##   TRUE   TRUE   TRUE
```

```
sapply(dogs[cols], range, na.rm = TRUE)
```

```
##      weight height price
## [1,]      5      5   283
## [2,]    175     32  3460
```

The Split-Apply Strategy

Watch the "The Split-Apply Strategy" lecture video.

Exercise 3

Use the split-apply strategy to compute the minimum weight (ignoring missing values) for each size of dog.

YOUR ANSWER GOES HERE:

```
by_group = split(dogs$weight, dogs$size)
sapply(by_group, min, na.rm = TRUE)
```

```
##  large medium  small
##     55     16      5
```

Exercise 4

Use `tapply()` to compute a `summary()` of the weight column for each group (hound, herding, etc) of dog.

YOUR ANSWER GOES HERE:

```
c(TRUE, FALSE) || c(FALSE, FALSE)
```

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

```
x = c(1, 2, 'hi')
class(dogs$size)
```

```
## [1] "factor"
```

```
class(dogs['size'])
```

```
## [1] "data.frame"
```

```
dogs[dogs$size == 'small',]
```

```
##                            breed        group datadog popularity_all
## 2                 Border Terrier      terrier    3.61             80
## 4                  Cairn Terrier      terrier    3.53             59
## 7                 Cocker Spaniel     sporting    3.30             27
## 8                       Papillon          toy    3.26             38
## 10              Shetland Sheepdog      herding    3.22             20
## 12                    Lhasa Apso non-sporting    3.21             62
## 13                  Affenpinscher          toy    3.20            139
## 14                     Dachshund        hound    3.19              9
## 15            Miniature Schnauzer      terrier    3.19             12
## 16                     Chihuahua          toy    3.15             14
## 17             Australian Terrier      terrier    3.11            121
## 20    West Highland White Terrier      terrier    3.08             35
## 21             Bedlington Terrier      terrier    3.07            134
## 23                   Bichon Frise non-sporting    3.03             39
## 26                Tibetan Spaniel non-sporting    3.02            114
## 28                        Maltese          toy    2.93             23
## 29                     Pomeranian          toy    2.93             17
## 30                       Shih Tzu          toy    2.93             11
## 32              Yorkshire Terrier          toy    2.85              5
## 35               Brussels Griffon          toy    2.80             77
## 38                         Beagle        hound    2.79              3
## 40                Tibetan Terrier non-sporting    2.75             86
```

```
## 45                Norfolk Terrier      terrier    2.71            120
## 49            English Toy Spaniel          toy    2.59            129
## 51  Cavalier King Charles Spaniel          toy    2.57             21
## 53                   Basset Hound        hound    2.54             41
## 55              Italian Greyhound          toy    2.49             65
## 58            Pembroke Welsh Corgi      herding    2.45             25
## 60          Dandie Dinmont Terrier      terrier    2.42            160
## 64               Scottish Terrier      terrier    2.27             54
## 70                     Pekingese          toy    2.05             64
## 90            American Eskimo Dog non-sporting      NA            116
## 105           Cardigan Welsh Corgi      herding      NA             81
## 106                Cesky Terrier      terrier      NA            106
## 107               Chinese Crested          toy      NA             61
## 118         Glen of Imaal Terrier      terrier      NA            158
## 122                     Havanese          toy      NA             31
## 128                Japanese Chin          toy      NA             78
## 132              Lakeland Terrier      terrier      NA            135
## 134                       Löwchen non-sporting      NA            152
## 135            Manchester Terrier      terrier      NA            119
## 136         Miniature Bull Terrier      terrier      NA            127
## 137             Miniature Pinscher          toy      NA             42
## 141           Norwegian Lundehund non-sporting      NA            170
## 142               Norwich Terrier      terrier      NA             89
## 145         Parson Russell Terrier      terrier      NA             97
## 146  Petit Basset Griffon Vendeen        hound      NA            131
## 153                   Schipperke non-sporting      NA            105
## 155              Sealyham Terrier      terrier      NA            163
## 156                     Shiba Inu non-sporting      NA             53
## 157                 Silky Terrier          toy      NA             85
## 158                  Skye Terrier      terrier      NA            164
## 159            Smooth Fox Terrier      terrier      NA            113
## 163                Sussex Spaniel     sporting      NA            161
## 164              Swedish Vallhund      herding      NA            153
## 166               Toy Fox Terrier          toy      NA            101
## 169                 Welsh Terrier      terrier      NA             99
## 170              Wire Fox Terrier      terrier      NA            100
##     popularity lifetime_cost intelligence_rank longevity ailments price
## 2           61         22638                30     14.00        0   833
## 4           48         21992                35     13.84        2   435
## 7           27         24330                20     12.50        2   465
## 8           33         21001                 8     13.00        5   740
## 10          20         21006                 6     12.53        5   465
## 12          50         22031                68     13.92        1   350
## 13          84         18333                37     11.42        0   510
## 14           9         20113                49     12.63        2   423
## 15          12         20087                12     11.81        2   715
## 16          14         26250                67     16.50        1   588
## 17          77         17892                34     11.05        0   640
## 20          32         20490                47     12.80        3   538
## 21          83         22107                40     13.51        2  1058
## 23          34         19735                45     12.21        0   693
## 26          73         25549                46     14.42        0  1050
## 28          23         19084                59     12.25        1   650
## 29          17         15792                23      9.67        1   670
```

```
## 30          11         21152                70     13.20        1   583
## 32           5         20701                27     12.60        4  1057
## 35          59         19551                59     12.00        0   833
## 38           3         19468                73     12.30        1   288
## 40          64         20336                62     12.31        0  1140
## 45          76         24308                56     13.07        0  2083
## 49          80         17521                45     10.10        0   925
## 51          21         18639                44     11.29        2  1017
## 53          36         18328                71     11.43        2   490
## 55          53         16463                60     10.02        0   800
## 58          25         23978                11     12.25        9   587
## 60          87         21633                62     12.17        0   925
## 64          45         17525                65     10.69        1   829
## 70          52         20565                74     11.56        1   885
## 90          NA            NA                NA        NA        0   560
## 105         NA            NA                26     12.70        0   828
## 106         NA            NA                NA      8.42       NA    NA
## 107         NA            NA                61     10.08       NA   538
## 118         NA            NA                NA     10.42       NA    NA
## 122         NA            NA                NA     10.25        0   830
## 128         NA            NA                62      9.25        0   513
## 132         NA            NA                62        NA        0  1093
## 134         NA            NA                NA     10.00        0    NA
## 135         NA            NA                32      9.32        0   720
## 136         NA            NA                NA      6.60        0  1740
## 137         NA            NA                37        NA        0   535
## 141         NA            NA                NA        NA       NA    NA
## 142         NA            NA                38        NA        0  1245
## 145         NA            NA                NA        NA        0   528
## 146         NA            NA                62     12.70        0   400
## 153         NA            NA                15     13.00        4   658
## 155         NA            NA                56     12.25        1    NA
## 156         NA            NA                NA      7.00        1   890
## 157         NA            NA                37     14.25        0   448
## 158         NA            NA                55     11.00        0   550
## 159         NA            NA                40     13.17        0   575
## 163         NA            NA                NA     11.17        0    NA
## 164         NA         22839                NA     14.17       NA   772
## 166         NA            NA                NA        NA       NA   460
## 169         NA            NA                53        NA        0   843
## 170         NA            NA                51     13.17        0   668
##     food_cost grooming   kids megarank_kids megarank  size weight height
## 2         324   weekly   high             2        1 small   13.5     NA
## 4         324   weekly   high             4        2 small   14.0  10.00
## 7         674   weekly   high             7        6 small   25.0  14.50
## 8         324   weekly medium             8       22 small     NA   9.50
## 10        405    daily   high            11        8 small   22.0  14.50
## 12        324   weekly   high            12        7 small   15.0  10.50
## 13        324   weekly medium            13       26 small     NA  10.25
## 14        324   weekly    low            14       54 small   24.0     NA
## 15        405   weekly medium            14       27 small   15.5  13.00
## 16        324   weekly    low            16       55 small    5.5   5.00
## 17        324   weekly medium            17       30 small     NA  10.50
## 20        324   weekly   high            20       10 small     NA  10.50
```

```
## 21        324   weekly medium          21       35 small   20.0     NA
## 23        324    daily   high          25       16 small     NA  10.50
## 26        466   weekly   high          26       14 small   12.0  10.00
## 28        270    daily    low          29       65 small    5.0   9.00
## 29        324   weekly medium          28       45 small    5.0     NA
## 30        324    daily   high          29       19 small   12.5   9.75
## 32        324    daily    low          32       69 small    5.5     NA
## 35        324   weekly    low          36       66 small    9.0     NA
## 38        324    daily   high          38       28 small     NA  14.00
## 40        324   weekly medium          40       51 small   24.0  15.50
## 45        466   weekly medium          45       53 small   12.0   9.50
## 49        405   weekly medium          48       56 small   11.0  10.00
## 51        324   weekly   high          50       35 small   15.5  12.50
## 53        324   weekly   high          53       39 small     NA  14.00
## 55        324   weekly    low          55       77 small     NA  14.00
## 58        674   weekly   high          58       43 small   26.0  11.00
## 60        466    daily   high          60       49 small   21.0   9.00
## 64        324    daily medium          64       71 small   20.0  10.00
## 70        466    daily medium          70       78 small   13.0     NA
## 90         NA     <NA>   <NA>          NA       NA small     NA  14.00
## 105        NA     <NA>   <NA>          NA       NA small   31.5  11.50
## 106        NA   weekly   high          NA       NA small   19.0  11.50
## 107        NA     <NA>   <NA>          NA       NA small     NA  12.00
## 118        NA     <NA>   <NA>          NA       NA small   35.0  13.25
## 122        NA     <NA>   <NA>          NA       NA small     NA   9.75
## 128        NA     <NA>   <NA>          NA       NA small     NA   9.50
## 132        NA   weekly   high          NA       NA small   17.0  14.50
## 134        NA     <NA>   <NA>          NA       NA small     NA  13.00
## 135        NA     <NA>   <NA>          NA       NA small   17.0     NA
## 136        NA     <NA>   <NA>          NA       NA small     NA  12.00
## 137        NA     <NA>   <NA>          NA       NA small     NA  11.25
## 141        NA     <NA>   <NA>          NA       NA small     NA  13.50
## 142        NA   weekly medium          NA       NA small   12.0  10.00
## 145        NA   weekly medium          NA       NA small   15.0  13.50
## 146        NA     <NA>   <NA>          NA       NA small     NA  14.00
## 153        NA     <NA>   <NA>          NA       NA small     NA  11.50
## 155        NA     <NA>   <NA>          NA       NA small   24.0  10.50
## 156        NA     <NA>   <NA>          NA       NA small   20.0  15.00
## 157        NA     <NA>   <NA>          NA       NA small   10.0   9.50
## 158        NA     <NA>   <NA>          NA       NA small   40.0   9.75
## 159        NA     <NA>   <NA>          NA       NA small   17.5  15.00
## 163        NA   weekly    low          NA       NA small   40.0  14.00
## 164        NA   weekly   high          NA       NA small     NA  12.50
## 166        NA     <NA>   <NA>          NA       NA small     NA  10.00
## 169        NA   weekly   high          NA       NA small   20.0  15.00
## 170        NA     <NA>   <NA>          NA       NA small   17.5  15.00
```

Exercise 5

The `aggregate()` function also implements the split-apply strategy, but returns the results as a data frame.

Use `aggregate()` to compute the maximum weight (ignoring missing values) for each group of dog.

Hint: The `by` parameter in `aggregate()` expects a list or data frame, so use `[` to select columns for `by` rather

than `$` or `[[`.

YOUR ANSWER GOES HERE:

```
aggregate(dogs['weight'], by = list(dogs$group), FUN = max, na.rm = TRUE)
```

```
##        Group.1 weight
## 1      herding   62.5
## 2        hound   97.5
## 3 non-sporting   52.5
## 4     sporting   70.0
## 5      terrier   60.0
## 6          toy   16.0
## 7      working  175.0
```

Exercise 6

Like `table()`, the `tapply()` function can use multiple categorical features to cross-tabulate results.

Use `tapply()` to compute the median price (ignoring missing values) for dogs, grouped by both size and grooming.

Hint: see the `tapply()` documentation for the `INDEX` parameter.

YOUR ANSWER GOES HERE:

```
tapply(dogs$price, list(dogs$size, dogs$grooming), median, na.rm = TRUE)
```

```
##        daily weekly monthly
## large  842.5   1040      NA
## medium 832.0    810     650
## small  693.0    740      NA
```

Even More Apply Functions

Watch the "Even More Apply Functions" lecture video.

Exercise 7

Translate your code from Exercise 1, Part 1 to use `vapply()` rather than `sapply()`.

YOUR ANSWER GOES HERE:

```
cols = c("weight", "height", "price")
vapply(dogs[cols], mean, 5100, na.rm = TRUE)
```

```
##    weight    height     price
##  44.97093  19.08962 876.81507
```

Choosing an Apply Function

Watch the "Choosing an Apply Function" lecture video.

No exercises for this section.

Conditional Expressions

Watch the "Choosing an Apply Function" lecture video.

No exercises for this section.

The `switch()` Function

Watch the "The switch() Function" lecture video.

No exercises for this section.

The Congruent Vectors Strategy

Watch the "The Congruent Vectors Strategy" lecture video.

No exercises for this section.