STAT 408: Week 4

Tidyverse Overview

2/8/2022

Cheat sheets!

Data wrangling cheat sheets

`dplyr`:

https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-transformation.pdf

`tidyr`:

https://raw.githubusercontent.com/rstudio/cheatsheets/main/tidyr.pdf

Animal survey data

Data

In much of these slides, we will use the animal species diversity data from Data Carpentry. Each row holds information for a single animal, and the columns represent:

Column	Description
record_id	Unique id for the observation
month	month of observation
day	day of observation
year	year of observation
plot_id	ID of a particular experimental plot of land

Data (cont)

Column	Description
species_id	2-letter code
sex	sex of animal (“M”, “F”)
hindfoot_length	length of the hindfoot in mm
weight	weight of the animal in grams
genus	genus of animal
species	species of animal
taxon	e.g. Rodent, Reptile, Bird, Rabbit
plot_type	type of plot

Tidyverse vs Base R

Reading in data

Base R: `read.csv()` (more generally, `read.table()`)

Tidyverse: `read_csv()` (more generally, `read_delim()`)

```
surveys <- read_csv("https://math.montana.edu/shancock/data/animal_survey.csv")
```

View the first few lines of the data…

```
head(surveys)
```

```
## # A tibble: 6 × 13
##   record_id month   day  year plot_id species_id sex   hindfoot_length weight
##       <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>           <dbl>  <dbl>
## 1         1     7    16  1977       2 NL         M                  32     NA
## 2        72     8    19  1977       2 NL         M                  31     NA
## 3       224     9    13  1977       2 NL         <NA>               NA     NA
## 4       266    10    16  1977       2 NL         <NA>               NA     NA
## 5       349    11    12  1977       2 NL         <NA>               NA     NA
## 6       363    11    12  1977       2 NL         <NA>               NA     NA
## # … with 4 more variables: genus <chr>, species <chr>, taxa <chr>,
## #   plot_type <chr>
```


Inspect the structure of the data…

```
str(surveys)
```

```
## spec_tbl_df [34,786 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ record_id      : num [1:34786] 1 72 224 266 349 363 435 506 588 661 ...
##  $ month          : num [1:34786] 7 8 9 10 11 11 12 1 2 3 ...
##  $ day            : num [1:34786] 16 19 13 16 12 12 10 8 18 11 ...
##  $ year           : num [1:34786] 1977 1977 1977 1977 1977 ...
##  $ plot_id        : num [1:34786] 2 2 2 2 2 2 2 2 2 2 ...
##  $ species_id     : chr [1:34786] "NL" "NL" "NL" "NL" ...
##  $ sex            : chr [1:34786] "M" "M" NA NA ...
##  $ hindfoot_length: num [1:34786] 32 31 NA NA NA NA NA NA NA NA ...
##  $ weight         : num [1:34786] NA NA NA NA NA NA NA NA 218 NA ...
##  $ genus          : chr [1:34786] "Neotoma" "Neotoma" "Neotoma" "Neotoma" ...
##  $ species        : chr [1:34786] "albigula" "albigula" "albigula" "albigula" ...
##  $ taxa           : chr [1:34786] "Rodent" "Rodent" "Rodent" "Rodent" ...
##  $ plot_type      : chr [1:34786] "Control" "Control" "Control" "Control" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   record_id = col_double(),
##   ..   month = col_double(),
##   ..   day = col_double(),
##   ..   year = col_double(),
##   ..   plot_id = col_double(),
##   ..   species_id = col_character(),
```

Summarize the variables…

```
summary(surveys)
```

```
##    record_id         month             day            year         plot_id
##  Min.   :    1   Min.   : 1.000   Min.   : 1.0   Min.   :1977   Min.   : 1.00
##  1st Qu.: 8964   1st Qu.: 4.000   1st Qu.: 9.0   1st Qu.:1984   1st Qu.: 5.00
##  Median :17762   Median : 6.000   Median :16.0   Median :1990   Median :11.00
##  Mean   :17804   Mean   : 6.474   Mean   :16.1   Mean   :1990   Mean   :11.34
##  3rd Qu.:26655   3rd Qu.:10.000   3rd Qu.:23.0   3rd Qu.:1997   3rd Qu.:17.00
##  Max.   :35548   Max.   :12.000   Max.   :31.0   Max.   :2002   Max.   :24.00
##
##   species_id            sex            hindfoot_length     weight
##  Length:34786       Length:34786       Min.   : 2.00   Min.   :  4.00
##  Class :character   Class :character   1st Qu.:21.00   1st Qu.: 20.00
##  Mode  :character   Mode  :character   Median :32.00   Median : 37.00
##                                        Mean   :29.29   Mean   : 42.67
##                                        3rd Qu.:36.00   3rd Qu.: 48.00
##                                        Max.   :70.00   Max.   :280.00
##                                        NA's   :3348    NA's   :2503
##     genus              species              taxa            plot_type
##  Length:34786       Length:34786       Length:34786       Length:34786
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```


Data frames

Base R: `data.frame()`

Tidyverse: `tibble()`

```
tibble1 <- tibble(x = 1:3, y = c('a','b','c'))
tibble1
```

```
## # A tibble: 3 × 2
##       x y
##   <int> <chr>
## 1     1 a
## 2     2 b
## 3     3 c
```

Tibbles

- The tibble includes the type of each vector, and only prints a certain number of rows/columns
- The `read_csv()` function creates a tibble rather than a `data.frame` object

```
is_tibble(surveys)
```

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


Subsetting

Base R: `[]`, `$`, `subset()`

Tidyverse: From the `dplyr` package,

- `filter()` will subset rows
- `select()` will subset columns

```
surveys %>% filter(weight < 5) %>% select(species_id, sex, weight)
```

```
## # A tibble: 17 × 3
##    species_id sex   weight
##    <chr>      <chr>  <dbl>
##  1 PF         F          4
##  2 PF         F          4
##  3 PF         M          4
##  4 RM         F          4
##  5 RM         M          4
##  6 PF         <NA>       4
##  7 PP         M          4
##  8 RM         M          4
##  9 RM         M          4
## 10 RM         M          4
## 11 PF         M          4
## 12 PF         F          4
## 13 RM         M          4
## 14 RM         M          4
## 15 RM         F          4
## 16 RM         M          4
## 17 RM         M          4
```


Re-ordering

Base R: `sort()`

Tidyverse: `arrange()`, `top_n()`

- use `arrange(desc())` for descending order

```
surveys %>% select(species_id, sex, weight) %>% arrange(weight) %>% head()
```

```
## # A tibble: 6 × 3
##   species_id sex   weight
##   <chr>      <chr>  <dbl>
## 1 PF         F          4
## 2 PF         F          4
## 3 PF         M          4
## 4 RM         F          4
## 5 RM         M          4
## 6 PF         <NA>       4
```



```
surveys  %>%
  select(species, weight, hindfoot_length) %>%
  top_n(n = 10, hindfoot_length) %>%
  arrange(desc(hindfoot_length))
```

```
## # A tibble: 19 × 3
##    species     weight hindfoot_length
##    <chr>        <dbl>           <dbl>
##  1 albigula        NA              70
##  2 ordii           35              64
##  3 ordii           51              58
##  4 spectabilis    123              58
##  5 spectabilis    136              57
##  6 spectabilis    156              57
##  7 spectabilis    143              56
##  8 spectabilis    148              55
##  9 spectabilis    140              55
## 10 spectabilis    104              55
## 11 spectabilis    144              55
## 12 spectabilis    136              55
## 13 spectabilis    168              55
## 14 spectabilis    142              55
## 15 spectabilis    154              55
## 16 spectabilis    128              55
## 17 spectabilis    142              55
```

Summarizing

Many data analysis tasks can be approached using the *split-apply-combine* paradigm: split the data into groups, apply some analysis to each group, and then combine the results.

Base R: `apply()`, `tapply()`, `lapply()`, etc.

Tidyverse:

- `group_by()` changes the scope of a function from operating on the entire data set to operating on it group-by-group, then
- `summarize()` calculates summary statistics like means and standard deviations


```
surveys %>%
  group_by(sex) %>%
  summarize(mean_weight = mean(weight, na.rm = TRUE))
```

```
## # A tibble: 3 × 2
##   sex   mean_weight
##   <chr>       <dbl>
## 1 F            42.2
## 2 M            43.0
## 3 <NA>         64.7
```

You can group by more than one variable…

```
surveys %>%
  filter(!is.na(weight)) %>%
  group_by(sex, species_id) %>%
  summarize(mean_weight = mean(weight))
```

```
## # A tibble: 64 × 3
## # Groups:   sex [3]
##    sex   species_id mean_weight
##    <chr> <chr>            <dbl>
##  1 F     BA                9.16
##  2 F     DM               41.6
##  3 F     DO               48.5
##  4 F     DS              118.
##  5 F     NL              154.
##  6 F     OL               31.1
##  7 F     OT               24.8
##  8 F     OX               21
##  9 F     PB               30.2
## 10 F     PE               22.8
## # … with 54 more rows
```


And you can summarize with more than one statistic…

```
surveys %>%
  filter(!is.na(weight)) %>%
  group_by(sex, species_id) %>%
  summarize(mean_weight = mean(weight),
            min_weight = min(weight)) %>%
  arrange(min_weight)
```

```
## # A tibble: 64 × 4
## # Groups:   sex [3]
##    sex   species_id mean_weight min_weight
##    <chr> <chr>            <dbl>      <dbl>
##  1 F     PF                7.97          4
##  2 F     RM               11.1           4
##  3 M     PF                7.89          4
##  4 M     PP               17.2           4
##  5 M     RM               10.1           4
##  6 <NA>  PF                6             4
##  7 F     OT               24.8           5
##  8 F     PP               17.2           5
##  9 F     BA                9.16          6
## 10 M     BA                7.36          6
## # … with 54 more rows
```

Exercise

Repeat this exercise from Week 2 but now using the `tidyverse` for data import and subsetting:

Read in the `Seattle` data set:

```
Seattle <- read_csv(
  'http://math.montana.edu/ahoegh/teaching/stat408/datasets/SeattleHousing.csv')
```

1. Create a new data frame that only includes houses worth more than $1,000,000.
2. From this new data frame, what is the average living square footage (`sqft_living`) of houses?

Solution in base R

```
expensive.houses <- subset(Seattle, price > 1000000) # or
expensive.houses <- Seattle[Seattle$price > 1000000,]
mean(expensive.houses$sqft_living)
```

```
## [1] 3890.065
```


Solution in tidyverse

```
# enter code here
```

More features of the tidyverse

Data transformation

- `mutate()` creates new columns (variables) using information from other columns
- `rename()` renames columns (variables)

```
surveys %>%
  filter(!is.na(weight)) %>%
  select(species, weight) %>%
  mutate(weight_kg = weight / 1000) %>%
  head()
```

```
## # A tibble: 6 × 3
##   species  weight weight_kg
##   <chr>     <dbl>     <dbl>
## 1 albigula    218     0.218
## 2 albigula    204     0.204
## 3 albigula    200     0.2
## 4 albigula    199     0.199
## 5 albigula    197     0.197
## 6 albigula    218     0.218
```


Counting

We often want to know the numbers of observations in a particular group.

```
surveys %>% count(sex)
```

```
## # A tibble: 3 × 2
##   sex       n
##   <chr> <int>
## 1 F     15690
## 2 M     17348
## 3 <NA>   1748
```

… is short-hand for

```
surveys %>% group_by(sex) %>% summarize(count = n())
```

```
surveys %>%
  count(sex, species) %>%
  arrange(species, desc(n))
```

```
## # A tibble: 81 × 3
##    sex   species            n
##    <chr> <chr>          <int>
##  1 F     albigula         675
##  2 M     albigula         502
##  3 <NA>  albigula          75
##  4 <NA>  audubonii          75
##  5 F     baileyi         1646
##  6 M     baileyi         1216
##  7 <NA>  baileyi           29
##  8 <NA>  bilineata        303
##  9 <NA>  brunneicapillus   50
## 10 <NA>  chlorurus         39
## # … with 71 more rows
```


Exercise

Exercise

1. How many animals were caught in each `plot_type` surveyed?
2. Use `group_by()` and `summarize()` to find the mean, min, and max hindfoot length for each species (using `species_id`). Also add the number of observations.
3. What was the heaviest animal measured in each year? Return the columns `year`, `genus`, `species_id`, and `weight`.