

Data Manipulation

```
## you can set your working directory in this chunk

#setwd("/Users/namigabbasov/Desktop/R-Data-Carpentry")
```

```
## libraries
```

```
#install.packages("tidyverse")
#install.packages("magrittr")
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.1
v purrr      1.0.2
```

```
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(magrittr)
```

Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

```
set_names
```

The following object is masked from 'package:tidyr':

extract

```
## import data
surveys<-read_csv("https://raw.githubusercontent.com/UnitForDataScience/RWorkshop/main/porta
```

Rows: 34786 Columns: 13

-- Column specification -----

Delimiter: ","

chr (6): species_id, sex, genus, species, taxa, plot_type

dbl (7): record_id, month, day, year, plot_id, hindfoot_length, weight

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.

Data manipulation using dplyr

We're going to learn some of the most common dplyr functions:

select(): subset columns filter(): subset rows on conditions mutate(): create new columns by using information from other columns group_by() and summarize(): create summary statistics on grouped data arrange(): sort results count(): count discrete values

Selecting columns

```
select(surveys, plot_id, species_id, weight)
```

A tibble: 34,786 x 3

	plot_id	species_id	weight
	<dbl>	<chr>	<dbl>
1	2	NL	NA
2	2	NL	NA
3	2	NL	NA
4	2	NL	NA
5	2	NL	NA
6	2	NL	NA
7	2	NL	NA

```

      8      2 NL      NA
      9      2 NL     218
     10      2 NL      NA
# i 34,776 more rows

```

Filtering rows

```
select(surveys, -record_id, -species_id)
```

```
# A tibble: 34,786 x 11
```

	month	day	year	plot_id	sex	hindfoot_length	weight	genus	species	taxa
	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<chr>	<chr>	<chr>
1	7	16	1977	2	M	32	NA	Neotoma	albigula	Rode~
2	8	19	1977	2	M	31	NA	Neotoma	albigula	Rode~
3	9	13	1977	2	<NA>	NA	NA	Neotoma	albigula	Rode~
4	10	16	1977	2	<NA>	NA	NA	Neotoma	albigula	Rode~
5	11	12	1977	2	<NA>	NA	NA	Neotoma	albigula	Rode~
6	11	12	1977	2	<NA>	NA	NA	Neotoma	albigula	Rode~
7	12	10	1977	2	<NA>	NA	NA	Neotoma	albigula	Rode~
8	1	8	1978	2	<NA>	NA	NA	Neotoma	albigula	Rode~
9	2	18	1978	2	M	NA	218	Neotoma	albigula	Rode~
10	3	11	1978	2	<NA>	NA	NA	Neotoma	albigula	Rode~

```

# i 34,776 more rows
# i 1 more variable: plot_type <chr>

```

```
filter(surveys, year == 1995)
```

```
# A tibble: 1,180 x 13
```

	record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>
1	22314	6	7	1995	2	NL	M	34	NA
2	22728	9	23	1995	2	NL	F	32	165
3	22899	10	28	1995	2	NL	F	32	171
4	23032	12	2	1995	2	NL	F	33	NA
5	22003	1	11	1995	2	DM	M	37	41
6	22042	2	4	1995	2	DM	F	36	45
7	22044	2	4	1995	2	DM	M	37	46
8	22105	3	4	1995	2	DM	F	37	49
9	22109	3	4	1995	2	DM	M	37	46

```

10      22168      4      1 1995      2 DM      M      36      48
# i 1,170 more rows
# i 4 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>

```

Two or More Functions at the Same Time: Pipes

```

## Intermediate steps

surveys2 <- filter(surveys, weight < 5)
surveys_sml <- select(surveys2, species_id, sex, weight)

## nested functions
surveys_sml <- select(filter(surveys, weight < 5), species_id, sex, weight)

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

```

```

# A tibble: 17 x 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

```

```
## Alternative New Pipe Operator
surveys %>%
  filter(weight < 5) %>%
  select(species_id, sex, weight)
```

```
# A tibble: 17 x 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
```

```
## Assign it to a new object(surveys_sml)
```

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

surveys_sml
```

```
# A tibble: 17 x 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

Mutate: Make new variables

```
surveys %>%
  mutate(weight_kg = weight / 1000)
```

A tibble: 34,786 x 14

	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
7	435	12	10	1977	2	NL	<NA>	NA	NA
8	506	1	8	1978	2	NL	<NA>	NA	NA
9	588	2	18	1978	2	NL	M	NA	218
10	661	3	11	1978	2	NL	<NA>	NA	NA

i 34,776 more rows

i 5 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>,

weight_kg <dbl>

create two columns at the same time

```
surveys %>%
  mutate(weight_kg = weight / 1000,
         weight_lb = weight_kg * 2.2)
```

```
# A tibble: 34,786 x 15
  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
7       435    12    10  1977     2 NL        <NA>      NA     NA
8       506     1     8  1978     2 NL        <NA>      NA     NA
9       588     2    18  1978     2 NL         M          NA    218
10      661     3    11  1978     2 NL        <NA>      NA     NA
# i 34,776 more rows
# i 6 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>,
#   weight_kg <dbl>, weight_lb <dbl>
```

```
## you can add other functions to Pipe as well
surveys %>%
  mutate(weight_kg = weight / 1000) %>%
  head()
```

```
# A tibble: 6 x 14
  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
# i 5 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>,
#   weight_kg <dbl>
```

```
## We can also filter missing observations before making a new column
surveys %>%
  filter(!is.na(weight)) %>%
  mutate(weight_kg = weight / 1000) %>%
  head()
```

```
# A tibble: 6 x 14
```

	record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>
1	588	2	18	1978	2	NL	M	NA	218
2	845	5	6	1978	2	NL	M	32	204
3	990	6	9	1978	2	NL	M	NA	200
4	1164	8	5	1978	2	NL	M	34	199
5	1261	9	4	1978	2	NL	M	32	197
6	1453	11	5	1978	2	NL	M	NA	218

i 5 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>,
weight_kg <dbl>

Operations on Groups: The `group_by()` and `summarize()` functions

```
## We group the data by "sex" column
surveys %>%
  group_by(sex) %>%
  summarize(mean_weight = mean(weight, na.rm = TRUE))
```

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

```
## You can also group by multiple columns
surveys %>%
  group_by(sex, species_id) %>%
  summarize(mean_weight = mean(weight, na.rm = TRUE)) %>%
  tail()
```

``summarise()`` has grouped output by 'sex'. You can override using the ``groups`` argument.

```
# A tibble: 6 x 3
# Groups:   sex [1]
  sex    species_id mean_weight
  <chr> <chr>          <dbl>
1 <NA> SU             NaN
```



```

2 <NA>  UL          NaN
3 <NA>  UP          NaN
4 <NA>  UR          NaN
5 <NA>  US          NaN
6 <NA>  ZL          NaN

```

```

## Filter out NAs before grouping
surveys %>%
  filter(!is.na(weight)) %>%
  group_by(sex, species_id) %>%
  summarize(mean_weight = mean(weight))

```

`summarise()` has grouped output by 'sex'. You can override using the `.groups` argument.

```

# A tibble: 64 x 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
# i 54 more rows

```

```

## Add print function to see the first 15 observations
surveys %>%
  filter(!is.na(weight)) %>%
  group_by(sex, species_id) %>%
  summarize(mean_weight = mean(weight)) %>%
  print(n = 15)

```

`summarise()` has grouped output by 'sex'. You can override using the `.groups` argument.

```
# A tibble: 64 x 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
11 F   PF           7.97
12 F   PH          30.8
13 F   PL          19.3
14 F   PM          22.1
15 F   PP          17.2
# i 49 more rows
```

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

`summarise()` has grouped output by 'sex'. You can override using the `.groups` argument.

```
# A tibble: 64 x 4
# Groups:   sex [3]
  sex  species_id mean_weight min_weight
<chr> <chr>         <dbl>         <dbl>
1 F    BA           9.16             6
2 F    DM          41.6            10
3 F    DO          48.5            12
4 F    DS         118.            45
5 F    NL         154.            32
6 F    OL          31.1            10
7 F    OT          24.8             5
```

```

8 F      OX      21      20
9 F      PB      30.2    12
10 F     PE      22.8    11
# i 54 more rows

```

arrange(): sort results

```

surveys|>
  arrange(hindfoot_length)

```

```

# A tibble: 34,786 x 13
  record_id month   day  year plot_id species_id sex  hindfoot_length weight
    <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>      <dbl>   <dbl>
1    31400     9    30  2000     19 PB        M          2     30
2    10067     3    16  1985     19 RM        M          6     16
3    19567     1     8  1992     19 BA        M          6      8
4    19015     9     9  1991     19 BA        F          7     11
5    21036     8    19  1993     21 PF        F          7      6
6    31457     9    31  2000     6  RM        M          8      8
7    19191    10    11  1991    13  PF        F          8     15
8     5801     4    29  1982     7  RM        <NA>       8     16
9    33647     3    14  2002     3  PF        M          9      8
10   20562    12    22  1992     5  RM        F          9     17
# i 34,776 more rows
# i 4 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>

```

```

## Let's make it in descending order
surveys|>
  arrange(desc(hindfoot_length))

```

```

# A tibble: 34,786 x 13
  record_id month   day  year plot_id species_id sex  hindfoot_length weight
    <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>      <dbl>   <dbl>
1    10574     7    23  1985    12  NL        <NA>       70     NA
2    30425     3     4  2000     1  DO        F         64     35
3    22049     2     4  1995     2  DO        M         58     51
4     1694     3    31  1979     8  DS        F         58    123
5     4449     5     4  1981     2  DS        F         57    136
6     3784     1    12  1981     1  DS        M         57    156
7    12973     7     1  1987     8  DS        M         56    143

```

```

      8      3958      3      9 1981      2 DS      M      55      148
      9      4514      6      4 1981      2 DS      F      55      140
     10     17998     11     10 1990     22 DS      F      55      104
# i 34,776 more rows
# i 4 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>

```

```

## we can also arrange after other data manipulation operations
surveys %>%
  filter(!is.na(weight)) %>%
  group_by(sex, species_id) %>%
  summarize(mean_weight = mean(weight),
            min_weight = min(weight)) %>%
  arrange(min_weight)

```

`summarise()` has grouped output by 'sex'. You can override using the `.groups` argument.

```

# A tibble: 64 x 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
# i 54 more rows

```

Counting

```

## let's count how many categories we have in "sex" variable
surveys %>%
  count(sex)

```

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

```
## grouping by "sex" and summarizing will give the same result.
surveys %>%
  group_by(sex) %>%
  summarize(count = n())
```

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

```
## The sort = TRUE argument sorts the resulting counts in descending order.
surveys %>%
  count(sex, sort = TRUE)
```

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

```
## count combination of factors
surveys %>%
  count(sex, species)
```

```
# A tibble: 81 x 3
  sex  species      n
  <chr> <chr>    <int>
1 F    albigula    675
2 F    baileyi   1646
3 F    eremicus    579
```

```

4 F      flavus      757
5 F      fulvescens   57
6 F      fulviventer  17
7 F      hispidus    99
8 F      leucogaster  475
9 F      leucopus    16
10 F     maniculatus  382
# i 71 more rows

```

```

## arrange in descending order for species(alphabetically)
surveys %>%
  count(sex, species) %>%
  arrange(species, desc(n))

```

```

# A tibble: 81 x 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
# i 71 more rows

```

```

## arrange in descending order for n
surveys %>%
  count(sex, species, sort = TRUE)

```

```

# A tibble: 81 x 3
  sex  species      n
  <chr> <chr>    <int>
1 M    merriami   5969
2 F    merriami   4554
3 M    ordii     1707
4 F    baileyi   1646
5 F    penicillatus 1608

```

```

6 M      penicillatus  1468
7 M      megalotis    1349
8 F      ordii        1308
9 M      spectabilis  1270
10 M     baileyi      1216
# i 71 more rows

```

Reshaping with pivot_wider and pivot_longer

1. Each variable has its own column
2. Each observation has its own row
3. Each value must have its own cell
4. Each type of observational unit forms a table

```

## First, let's make a new dataframe from our "Surveys" dataset
surveys_gw <- surveys %>%
  filter(!is.na(weight)) %>%
  group_by(plot_id, genus) %>%
  summarize(mean_weight = mean(weight))

```

`summarise()` has grouped output by 'plot_id'. You can override using the `.groups` argument.

```

## check variables
str(surveys_gw)

```

```

gropd_df [196 x 3] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ plot_id      : num [1:196] 1 1 1 1 1 1 1 1 2 2 ...
 $ genus        : chr [1:196] "Baiomys" "Chaetodipus" "Dipodomys" "Neotoma" ...
 $ mean_weight: num [1:196] 7 22.2 60.2 156.2 27.7 ...
 - attr(*, "groups")= tibble [24 x 2] (S3: tbl_df/tbl/data.frame)
 ..$ plot_id: num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
 ..$ .rows  : list<int> [1:24]
 .. ..$ : int [1:8] 1 2 3 4 5 6 7 8
 .. ..$ : int [1:9] 9 10 11 12 13 14 15 16 17
 .. ..$ : int [1:9] 18 19 20 21 22 23 24 25 26
 .. ..$ : int [1:8] 27 28 29 30 31 32 33 34
 .. ..$ : int [1:9] 35 36 37 38 39 40 41 42 43
 .. ..$ : int [1:8] 44 45 46 47 48 49 50 51
 .. ..$ : int [1:7] 52 53 54 55 56 57 58

```

```

.. ..$ : int [1:7] 59 60 61 62 63 64 65
.. ..$ : int [1:8] 66 67 68 69 70 71 72 73
.. ..$ : int [1:7] 74 75 76 77 78 79 80
.. ..$ : int [1:8] 81 82 83 84 85 86 87 88
.. ..$ : int [1:8] 89 90 91 92 93 94 95 96
.. ..$ : int [1:8] 97 98 99 100 101 102 103 104
.. ..$ : int [1:8] 105 106 107 108 109 110 111 112
.. ..$ : int [1:8] 113 114 115 116 117 118 119 120
.. ..$ : int [1:7] 121 122 123 124 125 126 127
.. ..$ : int [1:8] 128 129 130 131 132 133 134 135
.. ..$ : int [1:9] 136 137 138 139 140 141 142 143 144
.. ..$ : int [1:9] 145 146 147 148 149 150 151 152 153
.. ..$ : int [1:10] 154 155 156 157 158 159 160 161 162 163
.. ..$ : int [1:9] 164 165 166 167 168 169 170 171 172
.. ..$ : int [1:8] 173 174 175 176 177 178 179 180
.. ..$ : int [1:8] 181 182 183 184 185 186 187 188
.. ..$ : int [1:8] 189 190 191 192 193 194 195 196
.. ..@ ptype: int(0)
..- attr(*, ".drop")= logi TRUE

```

Pivot_wider

```

surveys_wide <- surveys_gw %>%
  pivot_wider(names_from = genus, values_from = mean_weight)

str(surveys_wide)

```

```

gropd_df [24 x 11] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ plot_id      : num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
 $ Baiomys      : num [1:24] 7 6 8.61 NA 7.75 ...
 $ Chaetodipus  : num [1:24] 22.2 25.1 24.6 23 18 ...
 $ Dipodomys    : num [1:24] 60.2 55.7 52 57.5 51.1 ...
 $ Neotoma      : num [1:24] 156 169 158 164 190 ...
 $ Onychomys    : num [1:24] 27.7 26.9 26 28.1 27 ...
 $ Perognathus  : num [1:24] 9.62 6.95 7.51 7.82 8.66 ...
 $ Peromyscus   : num [1:24] 22.2 22.3 21.4 22.6 21.2 ...
 $ Reithrodontomys: num [1:24] 11.4 10.7 10.5 10.3 11.2 ...
 $ Sigmodon     : num [1:24] NA 70.9 65.6 82 82.7 ...
 $ Spermophilus : num [1:24] NA NA NA NA NA NA NA NA NA ...
 - attr(*, "groups")= tibble [24 x 2] (S3: tbl_df/tbl/data.frame)

```



```

..$ plot_id: num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
..$ .rows : list<int> [1:24]
.. ..$ : int 1
.. ..$ : int 2
.. ..$ : int 3
.. ..$ : int 4
.. ..$ : int 5
.. ..$ : int 6
.. ..$ : int 7
.. ..$ : int 8
.. ..$ : int 9
.. ..$ : int 10
.. ..$ : int 11
.. ..$ : int 12
.. ..$ : int 13
.. ..$ : int 14
.. ..$ : int 15
.. ..$ : int 16
.. ..$ : int 17
.. ..$ : int 18
.. ..$ : int 19
.. ..$ : int 20
.. ..$ : int 21
.. ..$ : int 22
.. ..$ : int 23
.. ..$ : int 24
.. ..@ ptype: int(0)
..- attr(*, ".drop")= logi TRUE

```

```

## fill in missing observations
surveys_gw %>%
  pivot_wider(names_from = genus, values_from = mean_weight, values_fill = 0) %>%
  head()

```

```

# A tibble: 6 x 11
# Groups:   plot_id [6]
  plot_id Baiomys Chaetodipus Dipodomys Neotoma Onychomys Perognathus Peromyscus
    <dbl>   <dbl>      <dbl>      <dbl>   <dbl>      <dbl>      <dbl>      <dbl>
1       1       7      22.2      60.2    156.      27.7       9.62     22.2
2       2       6      25.1      55.7    169.      26.9       6.95     22.3
3       3      8.61     24.6      52.0    158.      26.0       7.51     21.4
4       4       0      23.0      57.5    164.      28.1       7.82     22.6

```

```

5      5      7.75      18.0      51.1      190.      27.0      8.66      21.2
6      6      0      24.9      58.6      180.      25.9      7.81      21.8
# i 3 more variables: Reithrodontomys <dbl>, Sigmodon <dbl>, Spermophilus <dbl>

```

Pivot_longer

```

surveys_long <- surveys_wide %>%
  pivot_longer(names_to = "genus", values_to = "mean_weight", cols = -plot_id)

str(surveys_long)

```

```

gropd_df [240 x 3] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ plot_id      : num [1:240] 1 1 1 1 1 1 1 1 1 1 ...
 $ genus        : chr [1:240] "Baiomys" "Chaetodipus" "Dipodomys" "Neotoma" ...
 $ mean_weight  : num [1:240] 7 22.2 60.2 156.2 27.7 ...
 - attr(*, "groups")= tibble [24 x 2] (S3: tbl_df/tbl/data.frame)
 ..$ plot_id: num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
 ..$ .rows  : list<int> [1:24]
 .. ..$ : int [1:10] 1 2 3 4 5 6 7 8 9 10
 .. ..$ : int [1:10] 11 12 13 14 15 16 17 18 19 20
 .. ..$ : int [1:10] 21 22 23 24 25 26 27 28 29 30
 .. ..$ : int [1:10] 31 32 33 34 35 36 37 38 39 40
 .. ..$ : int [1:10] 41 42 43 44 45 46 47 48 49 50
 .. ..$ : int [1:10] 51 52 53 54 55 56 57 58 59 60
 .. ..$ : int [1:10] 61 62 63 64 65 66 67 68 69 70
 .. ..$ : int [1:10] 71 72 73 74 75 76 77 78 79 80
 .. ..$ : int [1:10] 81 82 83 84 85 86 87 88 89 90
 .. ..$ : int [1:10] 91 92 93 94 95 96 97 98 99 100
 .. ..$ : int [1:10] 101 102 103 104 105 106 107 108 109 110
 .. ..$ : int [1:10] 111 112 113 114 115 116 117 118 119 120
 .. ..$ : int [1:10] 121 122 123 124 125 126 127 128 129 130
 .. ..$ : int [1:10] 131 132 133 134 135 136 137 138 139 140
 .. ..$ : int [1:10] 141 142 143 144 145 146 147 148 149 150
 .. ..$ : int [1:10] 151 152 153 154 155 156 157 158 159 160
 .. ..$ : int [1:10] 161 162 163 164 165 166 167 168 169 170
 .. ..$ : int [1:10] 171 172 173 174 175 176 177 178 179 180
 .. ..$ : int [1:10] 181 182 183 184 185 186 187 188 189 190
 .. ..$ : int [1:10] 191 192 193 194 195 196 197 198 199 200
 .. ..$ : int [1:10] 201 202 203 204 205 206 207 208 209 210
 .. ..$ : int [1:10] 211 212 213 214 215 216 217 218 219 220

```

```
.. ..$ : int [1:10] 221 222 223 224 225 226 227 228 229 230
.. ..$ : int [1:10] 231 232 233 234 235 236 237 238 239 240
.. ..@ ptype: int(0)
..- attr(*, ".drop")= logi TRUE
```

```
## look at first observations
```

```
surveys_wide %>%
  pivot_longer(names_to = "genus", values_to = "mean_weight", cols = -plot_id) %>%
  head()
```

```
# A tibble: 6 x 3
# Groups:   plot_id [1]
  plot_id genus      mean_weight
  <dbl> <chr>          <dbl>
1      1 Baiomys          7
2      1 Chaetodipus    22.2
3      1 Dipodomys     60.2
4      1 Neotoma      156.
5      1 Onychomys     27.7
6      1 Perognathus    9.62
```

Exporting data

```
## Let's make a new data
```

```
surveys_complete <- surveys %>%
  filter(!is.na(weight),           # remove missing weight
         !is.na(hindfoot_length), # remove missing hindfoot_length
         !is.na(sex))              # remove missing sex
```

```
## Extract the most common species_id
species_counts <- surveys_complete %>%
  count(species_id) %>%
  filter(n >= 50)
```

```
## Only keep the most common species
```

```
surveys_complete <- surveys_complete %>%  
  filter(species_id %in% species_counts$species_id)  
  
# Export the data final data  
write_csv(surveys_complete, file = "surveys_complete.csv")  
  
## export it as excel  
  
#install.packages("writexl")  
  
library(writexl)  
write_xlsx(x= surveys_complete, path="surveys_complete.xlsx", col_names=TRUE)
```