Data_Wrangling_Jacob_Chung_Joshua_Holt

```r
#install.packages("tidyverse")
## Load packages from library
library("tidyverse")
```

```
Warning: package 'dplyr' was built under R version 4.2.3

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.6
v forcats   1.0.1     v stringr   1.6.0
v ggplot2   4.0.1     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.2
v purrr     1.2.0
-- 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 becom
```

```r
library(here)
```

```
here() starts at /Users/jacobchung/coding_projects/Data_Wrangling_1_Rows_Columns_JC_JH
```

```r
fruits <- c("apple", "apple", "orange", "orange", "banana")
```

Start of code

```r
fruits <- c("apple", "apple", "orange", "orange", "banana")
unique(fruits)
```

```
[1] "apple"  "orange" "banana"
```

length

```
length(unique(fruits))
```

```
[1] 3
```

```
#create intermediate data objects
unique.fruits <- unique(fruits)
length(unique.fruits)
```

```
[1] 3
```

Piping

```
# for instance:
fruits %>% unique()
```

```
[1] "apple"  "orange" "banana"
```

```
# is the same as:
unique(fruits)
```

```
[1] "apple"  "orange" "banana"
```

```
# Piping!
fruits %>% unique() %>% length()
```

```
[1] 3
```

```
#Add lines
fruits %>%
  unique() %>%
  length()
```

```
[1] 3
```

Importing data

```r
# Import the data
cereal <- read_csv(here("cereal.csv"))
```

```
Rows: 77 Columns: 16
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): name, mfr, type
dbl (13): calories, protein, fat, sodium, fiber, carbo, sugars, potass, vita...

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.
```

```r
# Take a look at the first six rows
head(cereal)
```

```
# A tibble: 6 x 16
  name       mfr   type  calories protein   fat sodium fiber carbo sugars potass
  <chr>      <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 100% Bran  Nabi~ C           70       4     1    130  10     5        6    280
2 100% Natu~ Quak~ C          120       3     5     15   2     8        8    135
3 All-Bran   Kell~ C           70       4     1    260   9     7        5    320
4 All-Bran ~ Kell~ C           50       4     0    140  14     8        0    330
5 Almond De~ Rals~ C          110       2     2    200   1    14        8     -1
6 Apple Cin~ Gene~ C          110       2     2    180   1.5  10.5     10     70
# i 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>, cups <dbl>,
#   rating <dbl>
```

```r
# Start with the cereal dataframe
cereal %>%
  # This line extracts the column names from the dataframe input and creates a vector
  colnames() %>%
  # This line calculates the "length" (the number) of those names from the vector input
  length()
```

```
[1] 16
```

Select() columns

```
cereal %>%
  select(name, calories, fiber)
```

```
# A tibble: 77 x 3
   name                      calories fiber
   <chr>                        <dbl> <dbl>
 1 100% Bran                       70  10  
 2 100% Natural Bran              120   2  
 3 All-Bran                        70   9  
 4 All-Bran with Extra Fiber       50  14  
 5 Almond Delight                 110   1  
 6 Apple Cinnamon Cheerios        110   1.5
 7 Apple Jacks                    110   1  
 8 Basic 4                        130   2  
 9 Bran Chex                       90   4  
10 Bran Flakes                     90   5  
# i 67 more rows
```

```
cereal_fiber <- cereal %>%
  select(name, calories, fiber)

#Excluding columns
cereal %>%
  select(-name, -mfr)
```

```
# A tibble: 77 x 14
   type  calories protein   fat sodium fiber carbo sugars potass vitamins shelf
   <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl> <dbl>
 1 C           70       4     1    130  10     5        6    280       25     3
 2 C          120       3     5     15   2     8        8    135        0     3
 3 C           70       4     1    260   9     7        5    320       25     3
 4 C           50       4     0    140  14     8        0    330       25     3
 5 C          110       2     2    200   1    14        8     -1       25     3
 6 C          110       2     2    180   1.5  10.5     10     70       25     1
 7 C          110       2     0    125   1    11       14     30       25     2
 8 C          130       3     2    210   2    18        8    100       25     3
 9 C           90       2     1    200   4    15        6    125       25     1
10 C           90       3     0    210   5    13        5    190       25     3
# i 67 more rows
# i 3 more variables: weight <dbl>, cups <dbl>, rating <dbl>
```

Q1.1A:

```
cereal_sugar <- cereal %>%
  select(name, mfr, sugars)
```

1.4 rename

```
cereal_sugar %>%
  # Rename the mfr column to a more informative manufacturer
  rename(manufacturer = mfr)
```

```
# A tibble: 77 x 3
   name                      manufacturer   sugars
   <chr>                     <chr>           <dbl>
 1 100% Bran                 Nabisco             6
 2 100% Natural Bran         Quaker_Oats         8
 3 All-Bran                  Kelloggs            5
 4 All-Bran with Extra Fiber Kelloggs            0
 5 Almond Delight            Ralston_Purina      8
 6 Apple Cinnamon Cheerios   General_Mills      10
 7 Apple Jacks               Kelloggs           14
 8 Basic 4                   General_Mills       8
 9 Bran Chex                 Ralston_Purina      6
10 Bran Flakes               Post                5
# i 67 more rows
```

```
#multiple columns
cereal_sugar %>%
  rename(manufacturer = mfr,
         cereal_name = name)
```

```
# A tibble: 77 x 3
   cereal_name               manufacturer   sugars
   <chr>                     <chr>           <dbl>
 1 100% Bran                 Nabisco             6
 2 100% Natural Bran         Quaker_Oats         8
 3 All-Bran                  Kelloggs            5
 4 All-Bran with Extra Fiber Kelloggs            0
 5 Almond Delight            Ralston_Purina      8
 6 Apple Cinnamon Cheerios   General_Mills      10
 7 Apple Jacks               Kelloggs           14
 8 Basic 4                   General_Mills       8
 9 Bran Chex                 Ralston_Purina      6
10 Bran Flakes               Post                5
# i 67 more rows
```

1.5 relocate()

```
cereal_sugar %>%
  relocate(mfr, .before = name)
```

```
# A tibble: 77 x 3
   mfr            name                       sugars
   <chr>          <chr>                       <dbl>
 1 Nabisco        100% Bran                       6
 2 Quaker_Oats    100% Natural Bran               8
 3 Kelloggs       All-Bran                        5
 4 Kelloggs       All-Bran with Extra Fiber       0
 5 Ralston_Purina Almond Delight                  8
 6 General_Mills  Apple Cinnamon Cheerios        10
 7 Kelloggs       Apple Jacks                    14
 8 General_Mills  Basic 4                         8
 9 Ralston_Purina Bran Chex                       6
10 Post           Bran Flakes                     5
# i 67 more rows
```

1.6 filter()

```
cereal_sugar %>%
  filter(sugars > 12)
```

```
# A tibble: 9 x 3
  name                  mfr           sugars
  <chr>                 <chr>          <dbl>
1 Apple Jacks           Kelloggs          14
2 Cocoa Puffs           General_Mills     13
```

```
3 Count Chocula         General_Mills     13
4 Froot Loops           Kelloggs          13
5 Golden Crisp          Post              15
6 Mueslix Crispy Blend  Kelloggs          13
7 Post Nat. Raisin Bran Post              14
8 Smacks                Kelloggs          15
9 Total Raisin Bran     General_Mills     14
```

```
cereal_sugar %>%
  filter(mfr == "Kelloggs")
```

```
# A tibble: 23 x 3
   name                      mfr      sugars
   <chr>                     <chr>     <dbl>
 1 All-Bran                  Kelloggs      5
 2 All-Bran with Extra Fiber Kelloggs      0
 3 Apple Jacks               Kelloggs     14
 4 Corn Flakes               Kelloggs      2
 5 Corn Pops                 Kelloggs     12
 6 Cracklin' Oat Bran        Kelloggs      7
 7 Crispix                   Kelloggs      3
 8 Froot Loops               Kelloggs     13
 9 Frosted Flakes            Kelloggs     11
10 Frosted Mini-Wheats       Kelloggs      7
# i 13 more rows
```

Q1.2A:

```
cereal_sugar %>%
  filter(mfr == "Kelloggs",
         sugars > 12)
```

```
# A tibble: 4 x 3
  name                 mfr      sugars
  <chr>                <chr>     <dbl>
1 Apple Jacks          Kelloggs     14
2 Froot Loops          Kelloggs     13
3 Mueslix Crispy Blend Kelloggs     13
4 Smacks               Kelloggs     15
```

Q1.3A:

```
cereal %>%
  select(name, fat, potass) %>%
  filter(potass < 30)
```

```
# A tibble: 8 x 3
  name                      fat potass
  <chr>                   <dbl>  <dbl>
1 Almond Delight              2     -1
2 Corn Chex                   0     25
3 Corn Pops                   0     20
4 Cream of Wheat (Quick)      0     -1
5 Frosted Flakes              0     25
6 Fruity Pebbles              1     25
7 Puffed Rice                 0     15
8 Trix                        1     25
```

Q1.4A:

```
cereal %>%
  filter(calories > 120) %>%
  select(fiber, fat, sodium)
```

```
# A tibble: 8 x 3
  fiber   fat sodium
  <dbl> <dbl>  <dbl>
1   2       2    210
2   2       1    170
3   3       3     95
4   3       3    150
5   3       2    150
6   3       2    220
7   1.5     2    170
8   4       1    190
```

arrange()

```
cereal %>%
  arrange(calories)
```

```
# A tibble: 77 x 16
   name      mfr   type  calories protein   fat sodium fiber carbo sugars potass
   <chr>     <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
 1 All-Bran~ Kell~ C           50       4     0    140    14     8      0    330
 2 Puffed R~ Quak~ C           50       1     0      0     0    13      0     15
 3 Puffed W~ Quak~ C           50       2     0      0     1    10      0     50
 4 100% Bran Nabi~ C           70       4     1    130    10     5      6    280
 5 All-Bran  Kell~ C           70       4     1    260     9     7      5    320
 6 Shredded~ Nabi~ C           80       2     0      0     3    16      0     95
 7 Bran Chex Rals~ C           90       2     1    200     4    15      6    125
 8 Bran Fla~ Post  C           90       3     0    210     5    13      5    190
 9 Nutri-gr~ Kell~ C           90       3     0    170     3    18      2     90
10 Raisin S~ Kell~ C           90       2     0      0     2    15      6    110
# i 67 more rows
# i 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>, cups <dbl>,
#   rating <dbl>
```

```
cereal %>%
  # The desc() reverses the order, making it highest to lowest
  arrange(desc(calories))
```

```
# A tibble: 77 x 16
   name      mfr   type  calories protein   fat sodium fiber carbo sugars potass
   <chr>     <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
 1 Mueslix ~ Kell~ C          160       3     2    150   3    17       13    160
 2 Muesli R~ Rals~ C          150       4     3     95   3    16       11    170
 3 Muesli R~ Rals~ C          150       4     3    150   3    16       11    170
 4 Just Rig~ Kell~ C          140       3     1    170   2    20        9     95
 5 Nutri-Gr~ Kell~ C          140       3     2    220   3    21        7    130
 6 Total Ra~ Gene~ C          140       3     1    190   4    15       14    230
 7 Basic 4   Gene~ C          130       3     2    210   2    18        8    100
 8 Oatmeal ~ Gene~ C          130       3     2    170   1.5  13.5     10    120
 9 100% Nat~ Quak~ C          120       3     5     15   2     8        8    135
10 Cap'n'Cr~ Quak~ C          120       1     2    220   0    12       12     35
# i 67 more rows
# i 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>, cups <dbl>,
#   rating <dbl>
```

```
cereal %>%
  arrange(mfr, calories)
```

```
# A tibble: 77 x 16
   name      mfr   type  calories protein   fat sodium fiber carbo sugars potass
   <chr>     <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
 1 Maypo     Amer~ H          100       4     1      0   0    16        3     95
 2 Crispy W~ Gene~ C          100       2     1    140   2    11       10    120
 3 Multi-Gr~ Gene~ C          100       2     1    220   2    15        6     90
 4 Raisin N~ Gene~ C          100       3     2    140   2.5  10.5      8    140
 5 Total Wh~ Gene~ C          100       3     1    200   3    16        3    110
 6 Wheaties  Gene~ C          100       3     1    200   3    17        3    110
 7 Apple Ci~ Gene~ C          110       2     2    180   1.5  10.5     10     70
 8 Cheerios  Gene~ C          110       6     2    290   2    17        1    105
 9 Clusters  Gene~ C          110       3     2    140   2    13        7    105
10 Cocoa Pu~ Gene~ C          110       1     1    180   0    12       13     55
# i 67 more rows
# i 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>, cups <dbl>,
#   rating <dbl>
```

1.8 mutate()

```
cereal_carbs <- cereal %>%
  # Select just the name, carbs, and sugars columns, and store it as a new data object calle
  select(name, carbo, sugars)

cereal_carbs %>%
  # in this case, the new sugars_total column is going to be 2 times the sugars column
  mutate(sugars_total = sugars*2)
```

```
# A tibble: 77 x 4
   name                      carbo sugars sugars_total
   <chr>                     <dbl>  <dbl>        <dbl>
 1 100% Bran                   5        6           12
 2 100% Natural Bran           8        8           16
 3 All-Bran                    7        5           10
 4 All-Bran with Extra Fiber   8        0             0
 5 Almond Delight             14        8            16
 6 Apple Cinnamon Cheerios    10.5     10            20
 7 Apple Jacks                11       14            28
 8 Basic 4                    18        8            16
 9 Bran Chex                  15        6            12
10 Bran Flakes                13        5            10
# i 67 more rows
```

```
cereal_carbs %>%
  mutate(sugars = sugars*2)
```

```
# A tibble: 77 x 3
   name                      carbo sugars
   <chr>                     <dbl>  <dbl>
 1 100% Bran                   5       12
 2 100% Natural Bran           8       16
 3 All-Bran                    7       10
 4 All-Bran with Extra Fiber   8        0
 5 Almond Delight             14       16
 6 Apple Cinnamon Cheerios    10.5     20
 7 Apple Jacks                11       28
 8 Basic 4                    18       16
 9 Bran Chex                  15       12
10 Bran Flakes                13       10
# i 67 more rows
```

```
# Check that the original cereal_carbs data is unaltered
cereal_carbs
```

```
# A tibble: 77 x 3
   name                      carbo sugars
   <chr>                     <dbl>  <dbl>
 1 100% Bran                   5        6
 2 100% Natural Bran           8        8
 3 All-Bran                    7        5
 4 All-Bran with Extra Fiber   8        0
 5 Almond Delight             14        8
 6 Apple Cinnamon Cheerios    10.5     10
 7 Apple Jacks                11       14
 8 Basic 4                    18        8
 9 Bran Chex                     15         6
10 Bran Flakes                   13         5
# i 67 more rows
```

```r
cereal_carbs %>%
  mutate(sugars_with_milk = sugars + 5)
```

```
# A tibble: 77 x 4
   name                        carbo sugars sugars_with_milk
   <chr>                       <dbl>  <dbl>            <dbl>
 1 100% Bran                     5        6               11
 2 100% Natural Bran             8        8               13
 3 All-Bran                      7        5               10
 4 All-Bran with Extra Fiber     8        0                5
 5 Almond Delight               14        8               13
 6 Apple Cinnamon Cheerios      10.5     10               15
 7 Apple Jacks                  11       14               19
 8 Basic 4                      18        8               13
 9 Bran Chex                    15        6               11
10 Bran Flakes                  13        5               10
# i 67 more rows
```

```r
cereal_carbs %>%
  mutate(total_carbs = carbo + sugars) %>%
  mutate(sugars_with_milk = sugars + 5)
```

```
# A tibble: 77 x 5
   name                        carbo sugars total_carbs sugars_with_milk
   <chr>                       <dbl>  <dbl>       <dbl>            <dbl>
 1 100% Bran                     5        6        11                 11
 2 100% Natural Bran             8        8        16                 13
 3 All-Bran                      7        5        12                 10
 4 All-Bran with Extra Fiber     8        0         8                  5
 5 Almond Delight               14        8        22                 13
 6 Apple Cinnamon Cheerios      10.5     10        20.5               15
 7 Apple Jacks                  11       14        25                 19
 8 Basic 4                      18        8        26                 13
 9 Bran Chex                    15        6        21                 11
10 Bran Flakes                  13        5        18                 10
# i 67 more rows
```

```
cereal_carbs %>%
  mutate(total_carbs = carbo + sugars,
         sugars_with_milk = sugars + 5)
```

```
# A tibble: 77 x 5
   name                      carbo sugars total_carbs sugars_with_milk
   <chr>                     <dbl>  <dbl>       <dbl>            <dbl>
 1 100% Bran                   5        6        11                 11
 2 100% Natural Bran           8        8        16                 13
 3 All-Bran                    7        5        12                 10
 4 All-Bran with Extra Fiber   8        0         8                  5
 5 Almond Delight             14        8        22                 13
 6 Apple Cinnamon Cheerios    10.5     10        20.5               15
 7 Apple Jacks                11       14        25                 19
 8 Basic 4                    18        8        26                 13
 9 Bran Chex                  15        6        21                 11
10 Bran Flakes                13        5        18                 10
# i 67 more rows
```

BISON!

```
#install.packages("lterdatasampler")
library(lterdatasampler)
```

Q2.1A

```
knz_bison
```

```
# A tibble: 8,325 x 8
   data_code rec_year rec_month rec_day animal_code animal_sex animal_weight
   <chr>        <dbl>     <dbl>   <dbl> <chr>       <chr>              <dbl>
 1 CBH01         1994        11       8 813         F                    890
 2 CBH01         1994        11       8 834         F                   1074
 3 CBH01         1994        11       8 B-301       F                   1060
 4 CBH01         1994        11       8 B-402       F                    989
 5 CBH01         1994        11       8 B-403       F                   1062
 6 CBH01         1994        11       8 B-502       F                    978
 7 CBH01         1994        11       8 B-503       F                   1068
 8 CBH01         1994        11       8 B-504       F                   1024
 9 CBH01         1994        11       8 B-601       F                    978
10 CBH01         1994        11       8 B-602       F                   1188
# i 8,315 more rows
# i 1 more variable: animal_yob <dbl>
```

calculating the age of the bison

```
bison_stats <- knz_bison %>%
  mutate(bison_age = rec_year - animal_yob,
         bison_weight_kg = animal_weight * 0.453592)
bison_stats
```

```
# A tibble: 8,325 x 10
   data_code rec_year rec_month rec_day animal_code animal_sex animal_weight
   <chr>        <dbl>     <dbl>   <dbl> <chr>       <chr>              <dbl>
 1 CBH01         1994        11       8 813         F                    890
 2 CBH01         1994        11       8 834         F                   1074
 3 CBH01         1994        11       8 B-301       F                   1060
 4 CBH01         1994        11       8 B-402       F                    989
 5 CBH01         1994        11       8 B-403       F                   1062
 6 CBH01         1994        11       8 B-502       F                    978
 7 CBH01         1994        11       8 B-503       F                   1068
 8 CBH01         1994        11       8 B-504       F                   1024
 9 CBH01         1994        11       8 B-601       F                    978
10 CBH01         1994        11       8 B-602       F                   1188
# i 8,315 more rows
# i 3 more variables: animal_yob <dbl>, bison_age <dbl>, bison_weight_kg <dbl>
```

Q2.2A:

After looking at the data set, we came up with the research question, "Is there a case of sexual dimorphism in the bison based on the relationship between bison age and weight in the data provided?"

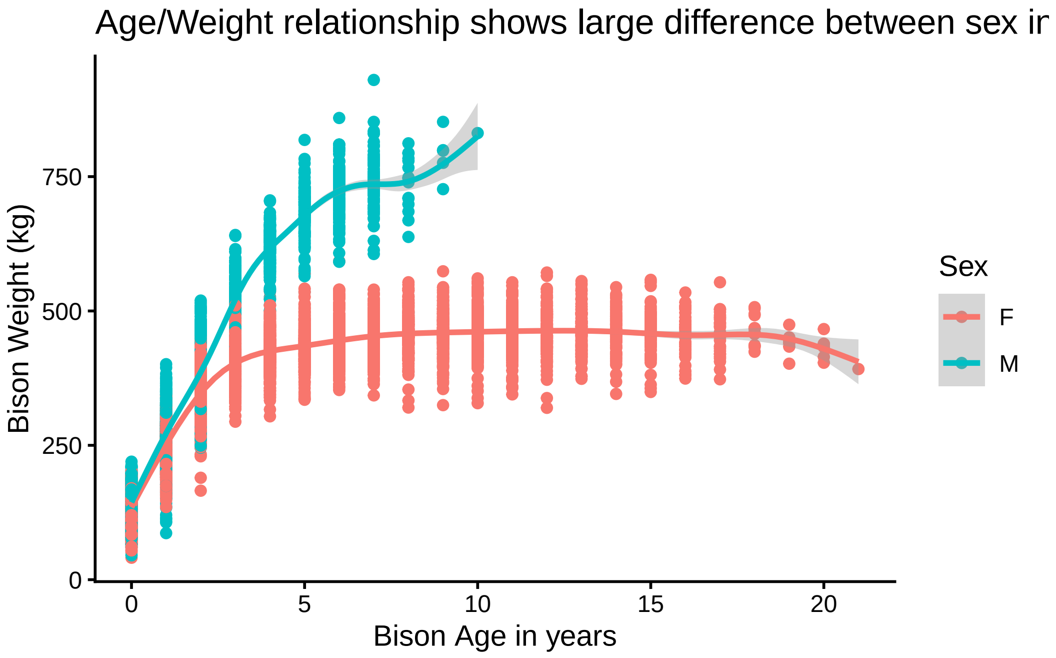
Q2.3A:

```
bison_graph <- ggplot(bison_stats, aes(x = bison_age, y = bison_weight_kg, color = animal_se
  geom_point() +
  geom_smooth() +
  labs(x = "Bison Age in years",
        y = "Bison Weight (kg)",
        title = "Age/Weight relationship shows large difference between sex in American Biso
  theme(plot.title = element_text(size = 10)) +
  theme_classic()
bison_graph
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
Warning: Removed 252 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 252 rows containing missing values or values outside the scale range
(`geom_point()`).
```



```r
ggsave("Age_Weight_Relationship_DataWrangling.png")
```

```
Saving 5.5 x 3.5 in image
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
Warning: Removed 252 rows containing non-finite outside the scale range
(`stat_smooth()`).
Removed 252 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Our sentences

After looking at our data, we noticed that there was a large dimorphism in weight between the sexes. A follow up research question is why do males disappear around age 10? Is it because of the way they were marking and recording the bison, or is there an ecological reason?