Chapitre 5: Manipulating and analyzing datas

Armelle de le Court

10/05/2021

1: data manipulation using dplyr and tiydr

Using tidyverse package to read the data and avoid having to se stringAsFactors to False

```
library("tidyverse")
```

2: What are dplyr and tidyr?

We're reading our data with read_csv instead of read.csv

```
surveys <- read_csv("data/portal_data_joined.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   record_id = col_double(),
##   month = col_double(),
##   day = col_double(),
##   year = col_double(),
##   plot_id = col_double(),
##   species_id = col_character(),
##   sex = col_character(),
##   hindfoot_length = col_double(),
##   weight = col_double(),
##   genus = col_character(),
##   species = col_character(),
##   taxa = col_character(),
##   plot_type = col_character()
## )
```

```
str(surveys) # to inspect the data
```

some important functions from dplyr:

select(): subset columns

filter(): subset rows on conditions

mutate(): create new columns by using information from other columns

group_by() and summarize(): create summary statisitcs on grouped data

arrange(): sort results

count(): count discrete values

3: Selecting columns and filtering rows

use select() to select colums of a dataframe

```
select(surveys, plot_id, species_id,weight)
## to select all of the columns except certain ones
select(surveys,-record_id,-species_id)
```

use filter to choose rows based on a specifi criteria

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

4: Pipes

To do multiple functions at the same time, we can use pipes, which are %>% (shortcut : Cmd + maj + M)

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

Question : Using pipes, subset the surveys data to include animals collected before 1995 and retain only the columns year, sex, and weight.

```
surveys %>%
  filter(year>1995) %>%
  select(year,sex,weight)
```

5: Mutate

To create a new column base on values in existing columns.

```
surveys %>%
  mutate(weight_kg = weight/1000) %>%
  head() # use head() to see only the 1st 6 lines
```

```
## # 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
## # ... with 5 more variables: genus <chr>, species <chr>, taxa <chr>,
## #   plot_type <chr>, weight_kg <dbl>
```

To remove NAs, we need to use filter too

```
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
## # ... with 5 more variables: genus <chr>, species <chr>, taxa <chr>,
## #   plot_type <chr>, weight_kg <dbl>
```

Question: Create a new data frame from the surveys data that meets the following criteria: contains only the species_id column and a new column called hindfoot_half containing values that are half the hindfoot_length values. In this hindfoot_half column, there are no NAs and all values are less than 30.

```
surveys_hindfoot_half <- surveys %>%
  filter(!is.na(hindfoot_length)) %>%
  mutate(hindfoot_half = hindfoot_length / 2) %>%
  filter(hindfoot_half < 30) %>%
  select(species_id, hindfoot_half)
surveys_hindfoot_half
```

```
## # A tibble: 31,436 x 2
```

```
##    species_id hindfoot_half
##    <chr>              <dbl>
##  1 NL                  16
##  2 NL                  15.5
##  3 NL                  16
##  4 NL                  17
##  5 NL                  16
##  6 NL                  16.5
##  7 NL                  16
##  8 NL                  16
##  9 NL                  16.5
## 10 NL                  15
## # ... with 31,426 more rows
```

6: Split-apply-combine data analysis

group_by to split the data into groups, apply some analysis to each group and then combine the results.

```
surveys %>%
  group_by(sex) %>%
  head()
```

```
## # A tibble: 6 x 13
## # Groups:   sex [2]
##   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>
```

```
# here it doesnt perform any data processing
```

Summarize is being used with group_by, which collapses each group into a single-row summary of that group. So to compute the mean weight by sex :

```
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
```

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

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

```
## # A tibble: 92 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
## # ... with 82 more rows
```

When there are NaN, we need to remove them

```
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
## # ... with 54 more rows
```

We can print with the argument "n" specifying the nmbr of rows to display.

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

```
  summarize(mean_weight = mean(weight)) %>%
  print(n = 5)
```

```
## `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.
## # ... with 59 more rows
```

we can summarize multiple variables at the same time. + arrange the results, to put the lighter species first

```
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
## # ... with 54 more rows
```

```
# in decreasing order we will use arrange(desc(x))
```

Counting

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

```
## # A tibble: 3 x 2
##   sex       n
```

```
##   <chr> <int>
## 1 F     15690
## 2 M     17348
## 3 <NA>   1748
```

```
# to count the number of rows of data for each sex, with an alphabetical order
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
## # ... with 71 more rows
```

```
# if we want an descending order of the count
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
## # ... with 71 more rows
```

Question:

```
#How many animals were caught in each plot_type surveyed?
surveys %>%
  count(plot_type)
```

```
## # A tibble: 5 x 2
##   plot_type                    n
##   <chr>                    <int>
## 1 Control                  15611
## 2 Long-term Krat Exclosure  5118
```

```
## 3 Rodent Exclosure           4233
## 4 Short-term Krat Exclosure  5906
## 5 Spectab exclosure          3918
```

```
#Use group_by() and summarize() to find the mean, min, and max hindfoot length for each species (using
```

```
surveys %>%
  filter(!is.na(hindfoot_length)) %>%
  group_by(species) %>%
  summarize(mean_hindfoot = mean(hindfoot_length),
            min_hindfoot = min(hindfoot_length),
            max_hindfoot = max(hindfoot_length),
            n=n())
```

```
## # A tibble: 22 x 5
##    species     mean_hindfoot min_hindfoot max_hindfoot     n
##    <chr>               <dbl>        <dbl>        <dbl> <int>
##  1 albigula             32.3           21           70  1074
##  2 baileyi              26.1            2           47  2864
##  3 eremicus             20.2           11           30  1212
##  4 flavus               15.6            7           38  1493
##  5 fulvescens           17.5           15           20    73
##  6 fulviventer          26.7           21           38    41
##  7 harrisi              33             31           35     2
##  8 hispidus             28.0           20           39   162
##  9 intermedius          22.2           20           23     9
## 10 leucogaster          20.5           12           39   920
## # ... with 12 more rows
```

```
#What was the heaviest animal measured in each year? Return the columns year, genus, species_id, and we
```

```
surveys %>%
  filter(!is.na(weight)) %>%
  group_by(year) %>%
  filter(weight==max(weight)) %>%
  select(year,genus,species_id,weight) %>%
  arrange(year)
```

```
## # A tibble: 27 x 4
## # Groups:   year [26]
##     year genus     species_id weight
##    <dbl> <chr>     <chr>       <dbl>
##  1  1977 Dipodomys DS            149
##  2  1978 Neotoma   NL            232
##  3  1978 Neotoma   NL            232
##  4  1979 Neotoma   NL            274
##  5  1980 Neotoma   NL            243
##  6  1981 Neotoma   NL            264
##  7  1982 Neotoma   NL            252
##  8  1983 Neotoma   NL            256
##  9  1984 Neotoma   NL            259
## 10  1985 Neotoma   NL            225
## # ... with 17 more rows
```

7: Reshaping data

Pivoting the data into a wider format

Pivot_wider takes three main arguments : the data to be transformed, the "names_from" col names whose values will become new col names, the "values_from" col names whose values will fill the new cols.

We will use that function to transform surveys to find the mean weight of each species in each plot over the entire survey period for example.

```
surveys_gw <- surveys %>%
  filter(!is.na(weight)) %>%
  group_by(genus, plot_id) %>%
  summarize(mean_weight = mean(weight))
```

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

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

```
## # A tibble: 24 x 11
##    plot_id Baiomys Chaetodipus Dipodomys Neotoma Onychomys Perognathus
##      <dbl>   <dbl>       <dbl>     <dbl>   <dbl>     <dbl>       <dbl>
##  1       1    7           22.2      60.2    156.      27.7        9.62
##  2       2    6           25.1      55.7    169.      26.9        6.95
##  3       3    8.61        24.6      52.0    158.      26.0        7.51
##  4       5    7.75        18.0      51.1    190.      27.0        8.66
##  5      18    9.5         26.8      61.4    149.      26.6        8.62
##  6      19    9.53        26.4      43.3    120       23.8        8.09
##  7      20    6           25.1      65.9    155.      25.2        8.14
##  8      21    6.67        28.2      42.7    138.      24.6        9.19
##  9       4   NA           23.0      57.5    164.      28.1        7.82
## 10       6   NA           24.9      58.6    180.      25.9        7.81
## # ... with 14 more rows, and 4 more variables: Peromyscus <dbl>,
## #   Reithrodontomys <dbl>, Sigmodon <dbl>, Spermophilus <dbl>
```

Pivoting data into a longer format

pivot_longer() takes four main arguments:the data to be transformed, the new names_to column we wish to create and populate with the current column names, the new values_to column we wish to create and populate with current values, the names of the columns to be used to populate the names_to and values_to variables (or to drop).

we are going to recreate surveys_gw from surveys_wide, for that we need to drop plot_id.

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

```
surveys_long
```

```
## # A tibble: 240 x 3
##    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
##  7       1 Peromyscus            22.2
##  8       1 Reithrodontomys       11.4
##  9       1 Sigmodon              NA
## 10       1 Spermophilus          NA
## # ... with 230 more rows
```

specification for what col to include with the ":" operator.

```
surveys_wide %>%
    pivot_longer(names_to = "genus",
                 values_to = "mean_weight",
                 Baiomys:Spermophilus)
```

```
## # A tibble: 240 x 3
##    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
##  7       1 Peromyscus            22.2
##  8       1 Reithrodontomys       11.4
##  9       1 Sigmodon              NA
## 10       1 Spermophilus          NA
## # ... with 230 more rows
```

To re-arraneg the data, we use arrange() function

```
surveys_wide %>%
    pivot_longer(names_to = "genus",
                 values_to = "mean_weight",
                 Baiomys:Spermophilus) %>%
    arrange(genus, plot_id)
```

```
## # A tibble: 240 x 3
##    plot_id genus   mean_weight
##      <dbl> <chr>         <dbl>
##  1       1 Baiomys        7
```

```
##  2       2 Baiomys        6
##  3       3 Baiomys        8.61
##  4       4 Baiomys       NA
##  5       5 Baiomys        7.75
##  6       6 Baiomys       NA
##  7       7 Baiomys       NA
##  8       8 Baiomys       NA
##  9       9 Baiomys       NA
## 10      10 Baiomys       NA
## # ... with 230 more rows
```

Questions

```
# Spread the surveys data frame with year as columns, plot_id as rows, and the number of genera per plo
rich_time <- surveys %>%
  group_by(plot_id, year) %>%
  summarize(n_genera = n_distinct(genus)) %>%
  pivot_wider(names_from = "year",
              values_from = "n_genera")
```

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

```
head(rich_time)
```

```
## # A tibble: 6 x 27
## # Groups:   plot_id [6]
##   plot_id `1977` `1978` `1979` `1980` `1981` `1982` `1983` `1984` `1985` `1986`
##     <dbl>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
## 1       1      2      3      4      7      5      6      7      6      4      3
## 2       2      6      6      6      8      5      9      9      9      6      4
## 3       3      5      6      4      6      6      8     10     11      7      6
## 4       4      4      4      3      4      5      4      6      3      4      3
## 5       5      4      3      2      5      4      6      7      7      3      1
## 6       6      3      4      3      4      5      9      9      7      5      6
## # ... with 16 more variables: 1987 <int>, 1988 <int>, 1989 <int>, 1990 <int>,
## #   1991 <int>, 1992 <int>, 1993 <int>, 1994 <int>, 1995 <int>, 1996 <int>,
## #   1997 <int>, 1998 <int>, 1999 <int>, 2000 <int>, 2001 <int>, 2002 <int>
```

```
#data frame and transform it with pivot_longer() so each row is a unique plot_id by year combination

rich_time %>%
  pivot_longer(names_to = "year",
               values_to = "n_genera",
               -plot_id)
```

```
## # A tibble: 624 x 3
## # Groups:   plot_id [24]
##    plot_id year  n_genera
##      <dbl> <chr>    <int>
##  1       1 1977         2
##  2       1 1978         3
##  3       1 1979         4
##  4       1 1980         7
##  5       1 1981         5
##  6       1 1982         6
```

```
##  7       1 1983         7
##  8       1 1984         6
##  9       1 1985         4
## 10       1 1986         3
## # ... with 614 more rows
```

```
#use pivot_longer() to create a dataset where we have a key column called measurement and a value colum
surveys_long <-
  surveys %>%
  pivot_longer(
    names_to = "measurement",
    values_to = "value",
    c(hindfoot_length, weight))
surveys_long
```

```
## # A tibble: 69,572 x 13
##    record_id month   day  year plot_id species_id sex   genus   species  taxa
##        <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr> <chr>   <chr>    <chr>
##  1         1     7    16  1977       2 NL         M     Neotoma albigula Rodent
##  2         1     7    16  1977       2 NL         M     Neotoma albigula Rodent
##  3        72     8    19  1977       2 NL         M     Neotoma albigula Rodent
##  4        72     8    19  1977       2 NL         M     Neotoma albigula Rodent
##  5       224     9    13  1977       2 NL         <NA>  Neotoma albigula Rodent
##  6       224     9    13  1977       2 NL         <NA>  Neotoma albigula Rodent
##  7       266    10    16  1977       2 NL         <NA>  Neotoma albigula Rodent
##  8       266    10    16  1977       2 NL         <NA>  Neotoma albigula Rodent
##  9       349    11    12  1977       2 NL         <NA>  Neotoma albigula Rodent
## 10       349    11    12  1977       2 NL         <NA>  Neotoma albigula Rodent
## # ... with 69,562 more rows, and 3 more variables: plot_type <chr>,
## #   measurement <chr>, value <dbl>
```

```
#calculate the average of each measurement in each year for each different plot_type.Then use pivot_wid

surveys_long %>%
  group_by(year, measurement, plot_type) %>%
  summarize(mean_value = mean(value, na.rm=TRUE)) %>%
  pivot_wider(names_from = "measurement",
              values_from = "mean_value")
```

```
## `summarise()` has grouped output by 'year', 'measurement'. You can override using the `.groups` argu
```

```
## # A tibble: 130 x 4
## # Groups:   year [26]
##     year plot_type                  hindfoot_length weight
##    <dbl> <chr>                                <dbl>  <dbl>
##  1  1977 Control                               36.1   50.4
##  2  1977 Long-term Krat Exclosure              33.7   34.8
##  3  1977 Rodent Exclosure                      39.1   48.2
##  4  1977 Short-term Krat Exclosure             35.8   41.3
##  5  1977 Spectab exclosure                     37.2   47.1
##  6  1978 Control                               38.1   70.8
##  7  1978 Long-term Krat Exclosure              22.6   35.9
##  8  1978 Rodent Exclosure                      37.8   67.3
##  9  1978 Short-term Krat Exclosure             36.9   63.8
## 10  1978 Spectab exclosure                     42.3   80.1
## # ... with 120 more rows
```

8: Exporting 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
surveys_complete
```

```
## # A tibble: 30,676 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       845     5     6  1978       2 NL         M                  32    204
##  2      1164     8     5  1978       2 NL         M                  34    199
##  3      1261     9     4  1978       2 NL         M                  32    197
##  4      1756     4    29  1979       2 NL         M                  33    166
##  5      1818     5    30  1979       2 NL         M                  32    184
##  6      1882     7     4  1979       2 NL         M                  32    206
##  7      2133    10    25  1979       2 NL         F                  33    274
##  8      2184    11    17  1979       2 NL         F                  30    186
##  9      2406     1    16  1980       2 NL         F                  33    184
## 10      3000     5    18  1980       2 NL         F                  31     87
## # ... with 30,666 more rows, and 4 more variables: genus <chr>, species <chr>,
## #   taxa <chr>, plot_type <chr>
```

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

```
## # A tibble: 14 x 2
##    species_id     n
##    <chr>      <int>
##  1 DM          9727
##  2 DO          2790
##  3 DS          2023
##  4 NL          1045
##  5 OL           905
##  6 OT          2081
##  7 PB          2803
##  8 PE          1198
##  9 PF          1469
## 10 PM           835
## 11 PP          2969
## 12 RF            73
## 13 RM          2417
## 14 SH           128
```

```
## Only keep the most common species
surveys_complete <- surveys_complete %>%
  filter(species_id %in% species_counts$species_id)
surveys_complete
```

```
## # A tibble: 30,463 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       845     5     6  1978       2 NL         M                  32    204
##  2      1164     8     5  1978       2 NL         M                  34    199
##  3      1261     9     4  1978       2 NL         M                  32    197
##  4      1756     4    29  1979       2 NL         M                  33    166
##  5      1818     5    30  1979       2 NL         M                  32    184
##  6      1882     7     4  1979       2 NL         M                  32    206
##  7      2133    10    25  1979       2 NL         F                  33    274
##  8      2184    11    17  1979       2 NL         F                  30    186
##  9      2406     1    16  1980       2 NL         F                  33    184
## 10      3000     5    18  1980       2 NL         F                  31     87
## # ... with 30,453 more rows, and 4 more variables: genus <chr>, species <chr>,
## #   taxa <chr>, plot_type <chr>
```

```
write_csv(surveys_complete, path = "data_output/surveys_complete.csv")
```

```
## Warning: The `path` argument of `write_csv()` is deprecated as of readr 1.4.0.
## Please use the `file` argument instead.
```

9: Additional exercises

Question 1

We are going to re-analyse beer consumption in 48 individuals using dplyr. The data are available in the rWSBIM1207 package. The data illustrated the fictive beer consumption in litres per year at different age according to gender and employment.

Directly load the data by typing

```
library(rWSBIM1207)
data(beers)
```

Remove observations with missing values.

```
beers_no_na<- beers %>%
  filter(!is.na(beers))
```

Using the Year, Month and Year columns, create a new column Date using dplyr::mutate and lubridate::ymd. What is the class of Date ?

```
library(lubridate)
library(dplyr)
library(tidyverse)
beers_no_na %>%
  mutate(Date = ymd(paste(Year,Month,Day)))
```

Create a new table, containing observations for women older than 35 years old, employed, and select all columns except Day, Month and Year, and order in descending value of consumption of beers.

```
new_obs_beers<- beers_no_na %>%
  filter(Gender=="Female",Age>35,Work=="Employed") %>%
  select(-Day,-Month,-Year) %>%
  arrange(desc(Consumption))
```

Export the new table to a csv file.

```
write_csv(new_obs_beers, path = "data_output/new_obs_beers.csv")
```

Beer consumption analysis:

Does employment status have an impact on beer consumption?

```
beers_no_na %>%
  group_by(Work) %>%
  summarize(Mean_Consumption= mean(Consumption,na.rm=TRUE))
```

```
## # A tibble: 2 x 2
##   Work       Mean_Consumption
##   <fct>                 <dbl>
## 1 Employed               172.
## 2 Unemployed             189
```

Do men drink more than women?

```
beers_no_na %>%
  group_by(Gender) %>%
  summarize(mean_cons = mean(Consumption, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   Gender mean_cons
##   <fct>      <dbl>
## 1 Female      165.
## 2 Male        195.
```

Does employment status have an influence on beer consumption according to gender?

```
beers_no_na %>%
  group_by(Gender,Work) %>%
  summarize(mean_cons = mean(Consumption, na.rm = TRUE))
```

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

```
## # A tibble: 4 x 3
## # Groups:   Gender [2]
```

```
##   Gender Work       mean_cons
##   <fct>  <fct>          <dbl>
## 1 Female Employed        178.
## 2 Female Unemployed      151.
## 3 Male   Employed        165.
## 4 Male   Unemployed      224
```

Question 2

The Cancer Genome Atlas (TCGA) is a large scale effort that has collected high throughput biology data from hundreds of patients samples. In this exercise, we are going to analyse the clinical variables recorded for a subset of the patients.

Using the clinical1.csv() function from rWSBIM1207, find the path the clinical1.csv file and read it to produce a data.frame named clinical.

```
library("rWSBIM1207")
clinical1<- read_csv(clinical1.csv())
clinical<- data_frame(clinical1)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
```

```
class(clinical)
```

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

Create a smaller data frame called clinical_mini containing only the columns corresponding to patientID, gender, age_at_diagnosis, smoking_history, number_pack_years_smoked, year_of_tobacco_smoking_onset, and stopped_smoking_year.

```
clinical_mini <- clinical %>%
  select(patientID,gender,age_at_diagnosis,smoking_history,
         number_pack_years_smoked,year_of_tobacco_smoking_onset,stopped_smoking_year)
```

Calculate the number of males and females in the cohort.

```
clinical_mini %>%
  group_by(gender) %>%
  count(gender)
```

```
## # A tibble: 2 x 2
## # Groups:   gender [2]
##   gender     n
##   <chr>  <int>
## 1 female   277
## 2 male     239
```

Create a new variable years_at_diagnosis corresponding to the age at diagnosis converted from days into years.

```
clinical_mini2 <- clinical_mini %>%
  mutate(years_at_diagnosis = age_at_diagnosis/365)
head(clinical_mini2)
```

```
## # A tibble: 6 x 8
##   patientID  gender age_at_diagnosis smoking_history          number_pack_years~
##   <chr>      <chr>             <dbl> <chr>                                <dbl>
## 1 TCGA-05-4~ male              24532 current reformed smoker~                52
## 2 TCGA-05-4~ male              24868 current reformed smoker~                62
## 3 TCGA-05-4~ male              24411 current reformed smoker~                20
## 4 TCGA-05-4~ male              25660 current reformed smoker~                43
## 5 TCGA-05-4~ female            21430 current reformed smoker~                15
## 6 TCGA-05-4~ male              27971 current reformed smoker~                NA
## # ... with 3 more variables: year_of_tobacco_smoking_onset <dbl>,
## #   stopped_smoking_year <dbl>, years_at_diagnosis <dbl>
```

###Calculate the mean and median age at diagnosis (in years). Pay attention to missing values!

```
clinical_mini2 %>%
  summarize(meanyear = mean(years_at_diagnosis, na.rm=TRUE),
            medianyear = median(years_at_diagnosis,na.rm=TRUE))
```

```
## # A tibble: 1 x 2
##   meanyear medianyear
##      <dbl>      <dbl>
## 1     65.8       66.9
```

Calculate the mean and median age at diagnosis for males and females.

```
clinical_mini2 %>%
  group_by(gender) %>%
  summarize(meanyear = mean(years_at_diagnosis, na.rm=TRUE),
            medianyear = median(years_at_diagnosis,na.rm=TRUE))
```

```
## # A tibble: 2 x 3
##   gender meanyear medianyear
##   <chr>     <dbl>      <dbl>
## 1 female     65.8       66.4
## 2 male       65.9       66.9
```

How many patient were diagnosed before 50 years?

```
clinical_mini2 %>%
  filter(years_at_diagnosis<50) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    32
```

Compare the mean age at diagnosis between current smoker and lifelong non-smoker.

```
clinical_mini2 %>%
  group_by(smoking_history) %>%
  summarize(meanAge = mean(years_at_diagnosis, na.rm = TRUE))
```

```
## # A tibble: 6 x 2
##   smoking_history                                    meanAge
##   <chr>                                                <dbl>
## 1 current reformed smoker for < or = 15 years           64.3
## 2 current reformed smoker for > 15 years                71.1
## 3 current reformed smoker, duration not specified       58.7
## 4 current smoker                                        61.9
## 5 lifelong non-smoker                                   66.1
## 6 <NA>                                                  68.7
```

Select patients who stopped smoking more than 15 years ago and calculate the number of smoking years for these cases. Display only cases for which you were able to calculate the data.

```
clinical %>%
  filter(stopped_smoking_year<2006, !is.na(stopped_smoking_year)) %>%
  mutate(number_of_smoking_years = stopped_smoking_year-year_of_tobacco_smoking_onset) %>%
  filter(!is.na(number_of_smoking_years))
```

```
## # A tibble: 130 x 16
##    patientID tumor_tissue_si~ gender age_at_diagnosis vital_status days_to_death
##    <chr>     <chr>            <chr>             <dbl>        <dbl>         <dbl>
##  1 TCGA-05-~ lung             male              24411            0            NA
##  2 TCGA-05-~ lung             female            21430            0            NA
##  3 TCGA-05-~ lung             male              28094            1           303
##  4 TCGA-05-~ lung             female            27241            0            NA
##  5 TCGA-05-~ lung             male              30194            0            NA
##  6 TCGA-05-~ lung             male              24472            0            NA
##  7 TCGA-05-~ lung             male              23863            0            NA
##  8 TCGA-05-~ lung             male              21061            0            NA
##  9 TCGA-44-~ lung             female            23854            0            NA
## 10 TCGA-44-~ lung             female            23808            0            NA
## # ... with 120 more rows, and 10 more variables: days_to_last_followup <dbl>,
## #   pathologic_stage <chr>, pathology_T_stage <chr>, pathology_N_stage <chr>,
## #   pathology_M_stage <chr>, smoking_history <chr>,
## #   number_pack_years_smoked <dbl>, year_of_tobacco_smoking_onset <dbl>,
## #   stopped_smoking_year <dbl>, number_of_smoking_years <dbl>
```

How many of them smoked less than 5 years?

```
clinical %>%
  filter(stopped_smoking_year<2006, !is.na(stopped_smoking_year)) %>%
  mutate(number_of_smoking_years = stopped_smoking_year-year_of_tobacco_smoking_onset) %>%
  filter(!is.na(number_of_smoking_years)) %>%
```

```
  filter(number_of_smoking_years<5) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     4
```

Try to recreate the following table, reporting the number of smokers and lifelong-non smoker between males and females. Note: the layout can be different.

```
clinictable <- clinical%>%
  select(smoking_history, gender)%>%
  filter(smoking_history %in% c("current smoker", "lifelong non-smoker"))
table(clinictable)
```

```
##                       gender
## smoking_history       female male
##   current smoker          51   69
##   lifelong non-smoker     55   20
```

Question 3

Using the interroA.csv() function from the rWSBIM1207 package to get the path to the spreadsheet file, read the data into R using the read_csv function. This data is in the wide format, with the results of each test stored as a separate column.

Using the appropriate pivot function, convert the data into a long table with a column interro informing which test that line refers to and a column result with the student's mark.

```
interroA <- read_csv(interroA.csv())
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   id = col_character(),
##   height = col_double(),
##   gender = col_character(),
##   X = col_double(),
##   interro1 = col_double(),
##   interro2 = col_double(),
##   interro3 = col_double(),
##   interro4 = col_double()
## )
```

```
longer <- interroA %>%
  pivot_longer(cols = c(interro1, interro2, interro3, interro4),
               names_to= "interros", values_to = "results")
longer
```

```
## # A tibble: 400 x 6
##    id     height gender     X interros results
##    <chr>   <dbl> <chr>  <dbl> <chr>      <dbl>
```

```
##  1 A74890    168 M      1.43  interro1      16
##  2 A74890    168 M      1.43  interro2      18
##  3 A74890    168 M      1.43  interro3       7
##  4 A74890    168 M      1.43  interro4      10
##  5 A85494    167 M      1.05  interro1      15
##  6 A85494    167 M      1.05  interro2      18
##  7 A85494    167 M      1.05  interro3      13
##  8 A85494    167 M      1.05  interro4      NA
##  9 A51820    166 M      0.435 interro1       4
## 10 A51820    166 M      0.435 interro2      10
## # ... with 390 more rows
```

```
interro_long<- interroA %>%
  pivot_longer(names_to = "interro",
               values_to = "res",
               c(interro1, interro2, interro3, interro4))
interro_long
```

```
## # A tibble: 400 x 6
##    id     height gender     X interro    res
##    <chr>   <dbl> <chr>  <dbl> <chr>    <dbl>
##  1 A74890    168 M      1.43  interro1    16
##  2 A74890    168 M      1.43  interro2    18
##  3 A74890    168 M      1.43  interro3     7
##  4 A74890    168 M      1.43  interro4    10
##  5 A85494    167 M      1.05  interro1    15
##  6 A85494    167 M      1.05  interro2    18
##  7 A85494    167 M      1.05  interro3    13
##  8 A85494    167 M      1.05  interro4    NA
##  9 A51820    166 M      0.435 interro1     4
## 10 A51820    166 M      0.435 interro2    10
## # ... with 390 more rows
```