R1 Exercises: Basic Data Wrangling

Contents

Packages used in this notebook:

---

1 Create tibble friends

Create a tibble `friends` using the commands `as_tibble()`, `tibble()` and `tribble()`, respectively, with the following variables: `name` (Susan, Walter, Tim, Ann), `height` in cm (180, 185, 190, 172) and `weight` in kg (70, 85, 100, 75). Additionally add a variable `sex` with entries (Male and Female) that corresponds to the sex of the `name` entry.

1.1 Create `friends` using `as_tibble()` (1P)

```
## # A tibble: 4 x 4
##   name   height weight sex
##   <chr>   <dbl>  <dbl> <chr>
## 1 Susan     180     70 Female
## 2 Walter    185     85 Male
## 3 Tim       190    100 Male
## 4 Ann       172     75 Female
```

1.2 Create `friends` using `tibble()` (1P)

```
## # A tibble: 4 x 4
##   name   height weight sex
##   <chr>   <dbl>  <dbl> <chr>
## 1 Susan     180     70 Female
## 2 Walter    185     85 Male
## 3 Tim       190    100 Male
## 4 Ann       172     75 Female
```

1.3 Create `friends` using `tribble()` (1P)

```
## # A tibble: 4 x 4
##   name   height weight sex
##   <chr>   <dbl>  <dbl> <chr>
## 1 Susan     180     70 Female
## 2 Walter    185     85 Male
## 3 Tim       190    100 Male
## 4 Ann       172     75 Female
```

1.4 Tidy data format (1P)

Is the data in the tibble `friends` in the format of a tidy data set? Explain your reasoning.

2 Basic data manipulation

2.1 Transform variable `sex` into a factor (1P)

Change the variable `sex` in `friends` into factor (use command `as.factor()`).

```
## # A tibble: 4 x 4
##   name   height weight sex
##   <chr>   <dbl>  <dbl> <fct>
## 1 Susan     180     70 Female
## 2 Walter    185     85 Male
## 3 Tim       190    100 Male
## 4 Ann       172     75 Female
```

2.2 Sorting `friends` (1P)

Sort the data in `friends` such that Male entries come before Female entries, subsequently the names in ascending order, the height in ascending and finally the weight in descending order. Why does the result show the taller Male and the Female with less weight first?

```
## # A tibble: 4 x 4
##   name   height weight sex
##   <chr>   <dbl>  <dbl> <fct>
## 1 Tim       190    100 Male
## 2 Walter    185     85 Male
## 3 Ann       172     75 Female
## 4 Susan     180     70 Female
```

2.3 Add variable `bmi` (1P)

Add an additional variable `bmi` (body mass index) **after the variable `name`** to the friends data. The `bmi` entry is the weight of a person in kg divided through the squared height in meter of that person.

```
## # A tibble: 4 x 5
##   name     bmi height weight sex
##   <chr>  <dbl>  <dbl>  <dbl> <fct>
## 1 Susan   21.6    180     70 Female
## 2 Walter  24.8    185     85 Male
## 3 Tim     27.7    190    100 Male
## 4 Ann     25.4    172     75 Female
```

2.4 Add variable `overweight` (1P)

Add to `friends` before column 3 a variable `overweight` that is a factor with entry `yes` for persons with a `bmi` larger than 25 and `no` otherwise.

```
## # A tibble: 4 x 6
##   name     bmi overweight height weight sex
##   <chr>  <dbl> <fct>       <dbl>  <dbl> <fct>
## 1 Susan   21.6 no            180     70 Female
## 2 Walter  24.8 no            185     85 Male
## 3 Tim     27.7 yes           190    100 Male
## 4 Ann     25.4 yes           172     75 Female
```

2.5 Summarize friends (1P)

Summarize the `friends` data by showing the mean of the heights and weight.

```
## # A tibble: 1 x 2
##   mean_height mean_weight
##         <dbl>       <dbl>
## 1        182.        82.5
```

2.6 Summarize friends separated by `sex` (1P)

Summarize the `friends` data by showing the mean of the heights and weight separated by `sex`.

```
## # A tibble: 2 x 3
##   sex    mean_height mean_weight
##   <fct>        <dbl>       <dbl>
## 1 Female        176         72.5
## 2 Male          188.        92.5
```

2.7 Summarize `bmi` by `overweight` (1P)

Summarize the `bmi` in `friends` by showing the mean, min and max of `bmi` separated by overweight.

```
## # A tibble: 2 x 4
##   overweight bmi_mean bmi_max bmi_min
##   <fct>         <dbl>   <dbl>   <dbl>
## 1 no             23.2    24.8    21.6
## 2 yes            26.5    27.7    25.4
```

2.8 Summarize `bmi` separated by `sex` and `overweight` (1P)

Summarize the `bmi` in `friends` by showing the mean, min and max of `bmi` separated by `sex` and using the `%>%` operator.

```
## # A tibble: 4 x 5
## # Groups:   sex [2]
##   sex    overweight bmi_mean bmi_max bmi_min
##   <fct>  <fct>         <dbl>   <dbl>   <dbl>
## 1 Female no             21.6    21.6    21.6
## 2 Female yes            25.4    25.4    25.4
## 3 Male   no             24.8    24.8    24.8
## 4 Male   yes            27.7    27.7    27.7
```

2.9 Add `mean` of `bmi` to the `friends` data (1P)

Add the mean of the `bmi` of all friends permanently to the `friends` data right after the `bmi` variable.

```
## # A tibble: 4 x 7
##   name     bmi bmi_mean overweight height weight sex
##   <chr>  <dbl>    <dbl> <fct>       <dbl>  <dbl> <fct>
## 1 Susan   21.6     24.9 no            180     70 Female
## 2 Walter  24.8     24.9 no            185     85 Male
## 3 Tim     27.7     24.9 yes           190    100 Male
## 4 Ann     25.4     24.9 yes           172     75 Female
```

2.10 Filter on two rows (1P)

Filter all friends with a height between 172 and 180 cm OR a having a weight exceeding 90 kg.

```
## # A tibble: 3 x 7
##   name    bmi bmi_mean overweight height weight sex
##   <chr> <dbl>    <dbl> <fct>       <dbl>  <dbl> <fct>
## 1 Susan  21.6     24.9 no            180     70 Female
## 2 Tim    27.7     24.9 yes           190    100 Male
## 3 Ann    25.4     24.9 yes           172     75 Female
```

2.11 Filter data on bmi (1P)

Show only those entries in `friends` that have a `bmi` larger than the average `bmi` of all entries in `friends`.

```
## # A tibble: 2 x 7
##   name    bmi bmi_mean overweight height weight sex
##   <chr> <dbl>    <dbl> <fct>       <dbl>  <dbl> <fct>
## 1 Tim    27.7     24.9 yes           190    100 Male
## 2 Ann    25.4     24.9 yes           172     75 Female
```

2.12 Select data on `bmi` (1P)

Show only the names of the persons in `friends` that have a `bmi` larger than the average of the `bmi`

```
## # A tibble: 2 x 1
##   name
##   <chr>
## 1 Tim
## 2 Ann
```

2.13 Select data on `bmi` and show it as a vector (1P)

Show the names of the persons in `friends` that have a `bmi` larger than the average of the `bmi` as a vector (Hint: a tibble is still a data frame and a data frame is a list, so you can extract the names form a tibble the way you would extract it from a list).

```
## [1] "Tim" "Ann"
```

3 The state.x77 data

3.1 Transform the data (4P)

In the state.x77 data, create a new variable Risk with the values high (Murder > 10), low (Murder < 4) and average.

- Show how to create this new variable Risk with ifelse() and with case_when()
- Transform Area into square kilometers. Replace the old variable. One square miles is equals to 2.58998811 square kilometers.
- Remove the variable Frost

Using ifelse():

```
## # A tibble: 50 x 8
##    Population Income Illiteracy ‘Life Exp‘ Murder ‘HS Grad‘     Area Risk
##         <dbl>  <dbl>      <dbl>      <dbl>  <dbl>     <dbl>    <dbl> <chr>
##  1       3615   3624        2.1       69.0   15.1      41.3  131333. high
##  2        365   6315        1.5       69.3   11.3      66.7 1467052. high
##  3       2212   4530        1.8       70.6    7.8      58.1  293749. average
##  4       2110   3378        1.9       70.7   10.1      39.9  134537. high
##  5      21198   5114        1.1       71.7   10.3      62.6  404973. high
##  6       2541   4884        0.7       72.1    6.8      63.9  268753. average
##  7       3100   5348        1.1       72.5    3.1      56     12593. low
##  8        579   4809        0.9       70.1    6.2      54.6    5133. average
##  9       8277   4815        1.3       70.7   10.7      52.6  140092. high
## 10       4931   4091        2         68.5   13.9      40.6  150408. high
## # ... with 40 more rows
```

Using case_when():

```
## # A tibble: 50 x 8
##    Population Income Illiteracy ‘Life Exp‘ Murder ‘HS Grad‘     Area Risk
##         <dbl>  <dbl>      <dbl>      <dbl>  <dbl>     <dbl>    <dbl> <chr>
##  1       3615   3624        2.1       69.0   15.1      41.3  131333. high
##  2        365   6315        1.5       69.3   11.3      66.7 1467052. high
##  3       2212   4530        1.8       70.6    7.8      58.1  293749. average
##  4       2110   3378        1.9       70.7   10.1      39.9  134537. high
##  5      21198   5114        1.1       71.7   10.3      62.6  404973. high
##  6       2541   4884        0.7       72.1    6.8      63.9  268753. average
##  7       3100   5348        1.1       72.5    3.1      56     12593. low
##  8        579   4809        0.9       70.1    6.2      54.6    5133. average
##  9       8277   4815        1.3       70.7   10.7      52.6  140092. high
## 10       4931   4091        2         68.5   13.9      40.6  150408. high
## # ... with 40 more rows
```

3.2 Transforming and Summarizing (4P)

Use the state.x77 data with the added Risk variable from above. For each risk group, compute mean, median, minimum, maximum income and count. Filter out the group with highest average income.

```
## # A tibble: 3 x 6
##   Risk     mean median   min   max     N
##   <chr>   <dbl>  <dbl> <dbl> <dbl> <int>
## 1 average 4477.  4546.  3601  5299    22
## 2 high    4301.  4091   3098  6315    17
## 3 low     4561.  4558   3694  5348    11
```

```
## # A tibble: 1 x 6
##   Risk   mean median   min   max     N
##   <chr> <dbl>  <dbl> <dbl> <dbl> <int>
## 1 low   4561.   4558  3694  5348    11
```