NZSSN Courses: Introduction to R

Session 3 –Data Manipulation

Statistical Consulting Centre

consulting@stat.auckland.ac.nz
The Department of Statistics
The University of Auckland

1 March, 2017

SCIENCE
DEPARTMENT OF STATISTICS

Data types

Everything in R is a vector (but some have only one element).

1. Numeric (same as double), or integer
   E.g. `Age`
2. String (same as character), or categorical
   E.g. `Ethnicity`, `Gender`, `Q1 - Q8`.
3. Logical: `TRUE` or `FALSE`, e.g.

```
1 == 1
```

```
[1] TRUE
```

```
2 <= 0
```

```
[1] FALSE
```

```
3 != 2
```

```
[1] TRUE
```

Missing values

```
table(issp.df$Gender)


     Female        Male NA, refused
        607         418          22
```

"NA, refused" means the information is unavailable. R's `is.na()` tests for missing values. Let's try this for the 9th participant who had the value "NA, refused" Gender.

```
issp.df[9, "Gender"]

[1] "NA, refused"

is.na(issp.df[9, "Gender"])

[1] FALSE
```

Missing values

- R reserves the object NA (Not Available) for elements of a vector that are missing or unavailable.
- Use of is.na() to search for missing values requires that they are recorded as NA.
- na will not do because R is case sensitive!

```
gender.na <- which(issp.df$Gender == "NA, refused")
gender.na

 [1]    9   31   49   72   79   98  141  226  269
[10]  271  377  382  522  538  540  705  759  760
[19]  829  881 1025 1035
```

Missing values

```
issp.df$Gender[gender.na]

 [1] "NA, refused" "NA, refused" "NA, refused"
 [4] "NA, refused" "NA, refused" "NA, refused"
 [7] "NA, refused" "NA, refused" "NA, refused"
[10] "NA, refused" "NA, refused" "NA, refused"
[13] "NA, refused" "NA, refused" "NA, refused"
[16] "NA, refused" "NA, refused" "NA, refused"
[19] "NA, refused" "NA, refused" "NA, refused"
[22] "NA, refused"
```

Missing values

Fix this:

```
issp.df$Gender[gender.na] <- NA
issp.df$Gender[gender.na]

 [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[16] NA NA NA NA NA NA NA

gender.missing <- is.na(issp.df$Gender[gender.na])
gender.missing

 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[10] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[19] TRUE TRUE TRUE TRUE
```

Missing values

How many cases of Gender are missing?

```
# How many are missing?
sum(gender.missing)

[1] 22
```

Missing values

The default option of `table()` ignores `NA`s when constructing frequency tables. Now that all occurrences of `"NA, refused"` have been replaced with `NA`, missing values will no longer be shown in frequency tables constructed using `table()`. If you still want to see how many `NA`s in the frequency tables, you can change the `useNA` arguement to `"always"` in `table()`.

```
table(issp.df$Gender)
```

```
Female   Male
   607    418
```

```
table(issp.df$Gender, useNA = "always")
```

```
Female   Male   <NA>
   607    418     22
```

Missing values

```
with(issp.df, table(Q6, Gender))
```

```
                        Gender
Q6                       Female Male
  almost always wrong        36   13
  always wrong              109   63
  cant choose, dk            61   27
  na, refused                 6    0
  not wrong at all          317  259
  only sometimes wrong       78   56
```

"na, refused" should be treated as missing values.

Missing values

```
# Case (row) numbers for which Q6 contains "na, refused"
Q6.na <- which(issp.df$Q6 == "na, refused")
Q6.na

[1]   9  25  96 141 267 383 518 538 940

issp.df$Q6[Q6.na]

[1] "na, refused" "na, refused" "na, refused"
[4] "na, refused" "na, refused" "na, refused"
[7] "na, refused" "na, refused" "na, refused"
```

Missing values

```
# Set cases with "na, refused" in Q6 to NA
issp.df$Q6[Q6.na] <- NA

# Re-print the table
with(issp.df, table(Q6, Gender))
```

```
                        Gender
Q6                       Female Male
  almost always wrong        36   13
  always wrong              109   63
  cant choose, dk            61   27
  not wrong at all          317  259
  only sometimes wrong       78   56
```

factor

What is a factor?

A variable which takes either qualitative values, ordinal values or a discrete set of quantitative values. The values of a factor are called its levels.

Examples of factors:

- `Gender` with 2 *qualitative* levels: `Male` and `Female`.
- `Education` with 6 *ordinal* levels: `None` $<$ `"Primary compl` $<$ `Incpl secondary` $<$ `Secondary compl` $<$ `Incpl university` $<$ `University degree`.
- `Income` has 9 *quantitative* levels when the mid-values of the income ranges are used: 5000, 12500, 17500, 22500, 27500, 35000, 45000, 60000 and 85000.

```
levels(factor(issp.df$Q6))

[1] "almost always wrong"  "always wrong"
[3] "cant choose, dk"      "not wrong at all"
[5] "only sometimes wrong"
```

Factor

- R stores two *additional* pieces of information for each factor: (1) the unique set of levels and (2) an integer value, assigned by R, for each unique level.
- The integer values are assigned to factor levels so that they have an order associated with them.
- By default, the unique levels are assigned the values 1, 2,..., according to ascending alphabetical order. This is not always appropriate!
- Example: Consider the factor `issp.df$Q6`

Level	Default order
almost always wrong	1
always wrong	2
cant choose, dk	3
only sometimes wrong	4
not wrong at all	5

Factor

Specify order using `levels` argument, i.e.

```
issp.df$Q6 <- factor(issp.df$Q6,
                     levels = c("always wrong",
                                "almost always wrong",
                                "only sometimes wrong",
                                "not wrong at all",
                                "cant choose, dk"))
with(issp.df, table(Q6, Gender))
```

```
                        Gender
Q6                       Female Male
  always wrong              109   63
  almost always wrong        36   13
  only sometimes wrong       78   56
  not wrong at all          317  259
  cant choose, dk            61   27
```

Repeat for Q7 and Q8

```
Q7.na <- which(issp.df$Q7 == "na, refused")
issp.df$Q7[Q7.na] <- NA
issp.df$Q7 = factor(issp.df$Q7, levels = c("always wrong",
            "almost always wrong", "only sometimes wrong",
            "not wrong at all", "cant choose, dk"))
with(issp.df, table(Q7, Gender))
```

```
                       Gender
Q7                      Female Male
  always wrong             417  258
  almost always wrong       97   78
  only sometimes wrong      46   30
  not wrong at all          13   23
  cant choose, dk           27   29
```

Repeat for Q7 and Q8

```
Q8.na <- which(issp.df$Q8 == "na, refused")
issp.df$Q8[Q8.na] <- NA
issp.df$Q8 = factor(issp.df$Q8, levels = c("always wrong",
            "almost always wrong", "only sometimes wrong",
            "not wrong at all", "cant choose, dk"))
with(issp.df, table(Q8, Gender))
```

```
                        Gender
Q8                       Female Male
  always wrong              456  283
  almost always wrong        89   83
  only sometimes wrong       29   30
  not wrong at all            4    9
  cant choose, dk            24   13
```

Q1 to Q4

```
with(issp.df, table(Q1, Gender))
```

```
                        Gender
Q1                       Female Male
  agree                     162  131
  cant choose, dk            15    5
  disagree                  157   96
  na, refused                12    2
  neither agree nor dis     188  139
  strongly agree             62   35
  strongly disagree          11   10
```

Tidy up Q1

This time we will treat both "na, refused" and "cant choose, dk" as missing values.

```
Q1.na <- which(issp.df$Q1 == "na, refused" |
               issp.df$Q1 == "can't choose, dk")
issp.df$Q1[Q1.na] <- NA
```

The pipe symbol '|' is read as 'or'.

Convert `Q1` to a `factor` with appropriately ordered levels, i.e.

```
issp.df$Q1 <- factor(issp.df$Q1,
                     levels = c("strongly agree",
                    "agree", "neither agree nor dis",
                    "disagree", "strongly disagree"))
```

Always check whether R has done the right thing!

```
with(issp.df, table(Q1, Gender))
```

```
                          Gender
Q1                         Female Male
  strongly agree               62   35
  agree                       162  131
  neither agree nor dis       188  139
  disagree                    157   96
  strongly disagree            11   10
```

Binning ages into age groups

Sometimes we are interested in examining responses by age group. The `ifelse()` function provides a quick way of doing binning ages into age groups, i.e.

```
ifelse(test, yes, no)
```

- `test`: a logical test.
- `yes`, what happens if the test is True.
- `no`, what happens if the test is False.

ifelse()

Example. Consider the ages of the first 10 participants.

```
test1 <- issp.df$Age[1:10]
test1

 [1] 56 45 38 33 37 27 43 24 NA 22

test1 < 30

 [1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
 [8]  TRUE    NA  TRUE

ifelse(test1<=30, "Under 30", "Over 31")

 [1] "Over 31"  "Over 31"  "Over 31"  "Over 31"
 [5] "Over 31"  "Under 30" "Over 31"  "Under 30"
 [9] NA         "Under 30"
```

ifelse()

What about *three* age groups?

Nest one `ifelse()` inside another. E.g.

```
test1

 [1] 56 45 38 33 37 27 43 24 NA 22

ifelse(test1<=30, "Under 30",
       ifelse(test1 <= 40, "31 to 40", "Over 40"))

 [1] "Over 40"  "Over 40"  "31 to 40" "31 to 40"
 [5] "31 to 40" "Under 30" "Over 40"  "Under 30"
 [9] NA         "Under 30"
```

ifelse()

Now, assign each of the 1047 respondents to one of three age groups: "Under 35", "36 to 60" and "Over 61".

```
issp.df$age.group <-
  ifelse(issp.df$Age<=35, "Under 35",
         ifelse(issp.df$Age <= 60, "36 to 60", "Over 61"))
table(issp.df$age.group)
```

```
36 to 60  Over 61 Under 35
     470      218      333
```

ifelse()

Convert age.group to a factor with levels in ascending order.

```
issp.df$age.group <- factor(issp.df$age.group,
                            levels = c("Under 35",
                                       "36 to 60",
                                       "Over 61"))
table(issp.df$age.group)
```

```
Under 35 36 to 60  Over 61
     333      470      218
```

Using ifelse() to tidy up Q2 – Q4

Use ifelse() to replace "na, refused" and "cant choose, dk" with NAs.

```
issp.df$Q2[7:12]
```

```
[1] "neither agree nor dis"
[2] "strongly disagree"
[3] "na, refused"
[4] "strongly disagree"
[5] "disagree"
[6] "agree"
```

```
# Create logical (TRUE/FALSE) variable of cases
# to replace with NA
make.na <- (issp.df$Q2 == "cant choose, dk" |
            issp.df$Q2 == "na, refused")
make.na[7:12]
```

```
[1] FALSE FALSE  TRUE FALSE FALSE FALSE
```

Using ifelse() to tidy up Q2 – Q4

```
issp.df$Q2 <- ifelse(make.na, NA, issp.df$Q2)
issp.df$Q2[7:12]

[1] "neither agree nor dis"
[2] "strongly disagree"
[3] NA
[4] "strongly disagree"
[5] "disagree"
[6] "agree"
```

Translation: If `make.na` is `TRUE` (i.e. cases for which `Q2` is "cant choose, dk" or "na, refused"), then replace that case in `Q2` with `NA`, otherwise leave it as is.

Using ifelse() to tidy up Q2 – Q4

```
issp.df$Q2 <- factor(issp.df$Q2,
                     levels = c("strongly agree",
                     "agree", "neither agree nor dis",
                     "disagree", "strongly disagree"))
with(issp.df, table(Q2, age.group))
```

```
                          age.group
Q2                         Under 35 36 to 60 Over 61
  strongly agree                  7       12       21
  agree                          23       87       83
  neither agree nor dis          46       81       41
  disagree                      134      192       56
  strongly disagree             119       88        6
```

Using ifelse() to tidy up Q2 – Q4

```
issp.df$Q3 <- ifelse(issp.df$Q3 == "cant choose, dk" |
                     issp.df$Q3 == "na, refused",
                     NA, issp.df$Q3)
issp.df$Q3 <- factor(issp.df$Q3,
                     levels = c("strongly agree",
                     "agree", "neither agree nor dis",
                     "disagree", "strongly disagree"))
issp.df$Q4 <- ifelse(issp.df$Q4 == "cant choose, dk" |
                     issp.df$Q4 == "na, refused",
                     NA, issp.df$Q4)
issp.df$Q4 <- factor(issp.df$Q4,
                     levels = c("strongly agree",
                     "agree", "neither agree nor dis",
                     "disagree", "strongly disagree"))
```

Using ifelse() to tidy up Q2 – Q4

```
with(issp.df, table(Q3, age.group))
```

```
                          age.group
Q3                         Under 35 36 to 60 Over 61
  strongly agree                  8        11       8
  agree                          16        54      63
  neither agree nor dis          27        75      45
  disagree                      164       227      83
  strongly disagree             116        93       9
```

Using ifelse() to tidy up Q2 – Q4

```
with(issp.df, table(Q4, age.group))
```

```
                          age.group
Q4                         Under 35 36 to 60 Over 61
  strongly agree                 25       54      30
  agree                         179      304     139
  neither agree nor dis          65       55      21
  disagree                       52       49      21
  strongly disagree               7        2       2
```

Summary

- Missing values
- Factor
- Subsetting vectors and datasets
- `ifelse()`