Data wrangling : Transforming (3/3)

Data wrangling is a task of great importance in data analysis. Data wrangling, is the process of importing, cleaning and transforming raw data into actionable information for analysis. It is a time-consuming process which is estimated to take about 60-80% of analyst's time. In this series we will go through this process. It will be a brief series with goal to craft the reader's skills on the data wrangling task. This is the third part of the series and it aims to cover the transforming of data used.This can include filtering, summarizing, and ordering your data by different means. This also includes combining various data sets, creating new variables, and many other manipulation tasks. At this post, we will go through a few more advanced transformation tasks on mtcars data set, in particular table manipulation.

Before proceeding, it might be helpful to look over the help pages for the iner_join, full__join, left_join, right_join, semi_join, anti_join, intersect, union, setdiff, bind_rows.

Moreover please load the following libraries and run the following link.

```
install.packages("dplyr")
library(dplyr)
```

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Create a new object named *car_inner* containing the observations that have matching values in both tables *mtcars* and *cars_table* using as key the variable ID.

Exercise 2

Create a new object named *car_left* containing all the observations from the left table (*mtcars*), and the matched records from the right table (*cars_table*) using as key the variable ID.

Learn more about Data Pre-Processing in the online course R Data Pre-Processing & Data Management – Shape your Data!. In this course you will learn how to:

- Work with popular libraries such as dplyr
- Learn about methods such as pipelines
- And much more

Exercise 3

Create a new object named *car_right* containing all the observations from the right table (*cars_table*), and the matched records from the right table (*mtcars*) using as key the variable ID.

Exercise 4

Create a new object named *car_full* containing all the observations when there is a match in either left (*cars_table*) or right (*mtcars*) table observation using as key the variable ID.

Exercise 5

Create a new object named *car_semi* containing all the observations from *mtcars* where there are matching values in

cars_table using as key the variable ID.

Exercise 6
Create a new object named *car_anti* containing all the observations from *mtcars* where there are not matching values in *cars_table* using as key the variable ID.

Exercise 7

Create a new object named *cars_inter* which contains rows that appear in both tables *mtcars* and *cars*.

Exercise 8

Create a new object named *cars_union* which contains rows appear in either tables *mtcars* and *cars*.

Exercise 9

Create a new object named *cars_diff* which contains rows appear in table *mtcars* and not *cars*.

Exercise 10

Append *mtcars* to *cars* and assign it at the object *car_rows*.

Data wrangling : Transforming (3/3) Solutions

Below are the solutions to these exercises on data transformation.

```
#####################  #  #  #  Exercise  1  #  #  #  #####################  cars_inner  <-  inner_join(mtcars,  cars_table,  by  =  'ID')  ;  cars_inner
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 2 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 3 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 4 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 5 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 6 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 7 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 8 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 9 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 10 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 11 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 12 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 Pontiac306 ## 13 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 14 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 15 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 16 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47 ## performance year ## 1 5.238095 2000 ## 2 4.078947 2007 ## 3 9.358289 2018 ## 4 5.801105 1999 ## 5 6.406250 2004 ## 6 10.404624 1985 ## 7 11.842105 1984 ## 8 20.673077 2009 ## 9 2.037037 2015 ## 10 4.511628 2006 ## 11 18.421053 2004 ## 12 9.114583 2003 ## 13 16.708861 1984 ## 14 8.883249 1998 ## 15 22.333333 2008 ## 16 5.093458 1998

```
##################### # # # Exercise 2 # # # ##################### cars_left <- left_join(mtcars, cars_table, by = 'ID'); cars_left
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda185 ## 2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet6 ## 5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 Duster128 ## 8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc-150 ## 9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc259 ## 10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 11 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 Merc336 ## 12 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 Merc259 ## 13 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 14 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 15 10.4 8 472.0 205 2.93

5.250 17.98 0 0 3 4 Cadillac109 ## 16 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 17 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 Chrysler92 ## 18 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 19 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 Honda207 ## 20 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 Toyota268 ## 21 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 22 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 Dodge212 ## 23 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 AMC148 ## 24 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 25 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 Pontiac306 ## 26 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 Fiat83 ## 27 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 Porsche289 ## 28 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 Lotus139 ## 29 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 30 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 31 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 32 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47 ## performance year ## 1 NA NA ## 2 5.238095 2000 ## 3 4.078947 2007 ## 4 NA NA ## 5 9.358289 2018 ## 6 5.801105 1999 ## 7 NA NA ## 8 NA NA ## 9 NA NA ## 10 6.406250 2004 ## 11 NA NA ## 12 NA NA ## 13 10.404624 1985 ## 14 11.842105 1984 ## 15 NA NA ## 16 20.673077 2009 ## 17 NA NA ## 18 2.037037 2015 ## 19 NA NA ## 20 NA NA ## 21 4.511628 2006 ## 22 NA NA ## 23 NA NA ## 24 18.421053 2004 ## 25 9.114583 2003 ## 26 NA NA ## 27 NA NA ## 28 NA NA ## 29 16.708861 1984 ## 30 8.883249 1998 ## 31 22.333333 2008 ## 32 5.093458 1998

```
#####################  #  #  #  Exercise  3  #  #  #  ##################### cars_right <- right_join(mtcars, cars_table, by = 'ID'); cars_right
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 2 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 3 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 4 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 5 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 6 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47 ## 7 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 8 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 9 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 10 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 11

15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 12 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 13 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 14 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 15 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 16 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 Pontiac306 ## performance year ## 1 4.511628 2006 ## 2 9.358289 2018 ## 3 8.883249 1998 ## 4 16.708861 1984 ## 5 5.801105 1999 ## 6 5.093458 1998 ## 7 4.078947 2007 ## 8 2.037037 2015 ## 9 11.842105 1984 ## 10 5.238095 2000 ## 11 22.333333 2008 ## 12 20.673077 2009 ## 13 10.404624 1985 ## 14 18.421053 2004 ## 15 6.406250 2004 ## 16 9.114583 2003

```
##################### # # # Exercise 4 # # # ##################### cars_full <- full_join(mtcars, cars_table, by = 'ID'); cars_full
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda185 ## 2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet6 ## 5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 Duster128 ## 8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc-150 ## 9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc259 ## 10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 11 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 Merc336 ## 12 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 Merc259 ## 13 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 14 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 15 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 Cadillac109 ## 16 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 17 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 Chrysler92 ## 18 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 19 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 Honda207 ## 20 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 Toyota268 ## 21 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 22 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 Dodge212 ## 23 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 AMC148 ## 24 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 25 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2

Pontiac306 ## 26 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 Fiat83 ## 27 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 Porsche289 ## 28 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 Lotus139 ## 29 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 30 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 31 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 32 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47 ## performance year ## 1 NA NA ## 2 5.238095 2000 ## 3 4.078947 2007 ## 4 NA NA ## 5 9.358289 2018 ## 6 5.801105 1999 ## 7 NA NA ## 8 NA NA ## 9 NA NA ## 10 6.406250 2004 ## 11 NA NA ## 12 NA NA ## 13 10.404624 1985 ## 14 11.842105 1984 ## 15 NA NA ## 16 20.673077 2009 ## 17 NA NA ## 18 2.037037 2015 ## 19 NA NA ## 20 NA NA ## 21 4.511628 2006 ## 22 NA NA ## 23 NA NA ## 24 18.421053 2004 ## 25 9.114583 2003 ## 26 NA NA ## 27 NA NA ## 28 NA NA ## 29 16.708861 1984 ## 30 8.883249 1998 ## 31 22.333333 2008 ## 32 5.093458 1998

```
#####################  #  #  #  Exercise  5  #  #  #  #####################  cars_semi  <-  semi_join(mtcars,  cars_table,  by  =  'ID');  cars_semi
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 2 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 3 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 4 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 5 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 6 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47 ## 7 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 8 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 9 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 10 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 11 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 12 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 13 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 14 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 15 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 16 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 Pontiac306

```
#####################  #  #  #  Exercise  6  #  #  #  #####################  cars_anti  <-  anti_join(mtcars,  cars_table,  by  =  'ID');  cars_anti
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 Lotus139 ## 2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 Porsche289 ## 3 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 Fiat83 ## 4 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 AMC148 ## 5 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 Dodge212 ## 6 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 Toyota268 ## 7 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 Honda207 ## 8 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 Chrysler92 ## 9 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 Cadillac109 ## 10 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 Merc336 ## 11 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc259 ## 12 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 Merc259 ## 13 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc-150 ## 14 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 Duster128 ## 15 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet6 ## 16 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda185

########## # # # Exercise 7 # # # ########## cars <- mtcars %>% sample_frac(size = 0.5, replace = FALSE) cars_inter <- intersect(mtcars, cars); cars_inter

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 Fiat83 ## 2 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 Cadillac109 ## 3 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 4 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 5 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 AMC148 ## 6 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 Honda207 ## 7 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 8 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 9 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 10 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 Merc259 ## 11 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 Chrysler92 ## 12 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda185 ## 13 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 14 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 15 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 16 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 Lotus139

########## # # # Exercise 8 # # # ########## cars_union <- union(mtcars, cars);

```
cars_union
```

```
## mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47 ## 2 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 3 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 4 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 5 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 Lotus139 ## 6 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 Porsche289 ## 7 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 Fiat83 ## 8 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 Pontiac306 ## 9 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 10 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 AMC148 ## 11 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 Dodge212 ## 12 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 13 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 Toyota268 ## 14 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 Honda207 ## 15 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 16 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 Chrysler92 ## 17 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 18 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 Cadillac109 ## 19 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 20 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 21 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 Merc259 ## 22 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 Merc336 ## 23 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 24 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc259 ## 25 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc-150 ## 26 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 Duster128 ## 27 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 28 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 29 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet6 ## 30 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 31 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 32 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda185
```

```
##################### # # # Exercise 9 # # # ##################### cars_dif <- setdiff(mtcars, cars); cars_dif
```

```
## mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet6 ## 2 14.3 8 360.0
```

245 3.21 3.570 15.84 0 0 3 4 Duster128 ## 3 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc-150 ## 4 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc259 ## 5 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 Merc336 ## 6 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 7 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 8 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 9 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 Toyota268 ## 10 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 Dodge212 ## 11 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 Pontiac306 ## 12 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 Porsche289 ## 13 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 Ford-111 ## 14 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 Ferrari269 ## 15 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 Maserati168 ## 16 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 Volvo47

```
##################### # # # Exercise 10 # # # ##################### car_rows <- bind_rows(mtcars, cars); car_rows
```

mpg cyl disp hp drat wt qsec vs am gear carb ID ## 1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 Mazda185 ## 2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 Mazda86 ## 3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 Datsun73 ## 4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 Hornet6 ## 5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 Hornet350 ## 6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 Valiant392 ## 7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 Duster128 ## 8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 Merc-150 ## 9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 Merc259 ## 10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 11 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 Merc336 ## 12 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 Merc259 ## 13 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 Merc172 ## 14 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 Merc175 ## 15 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 Cadillac109 ## 16 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 Lincoln219 ## 17 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 Chrysler92 ## 18 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 Fiat31 ## 19 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 Honda207 ## 20 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 Toyota268 ## 21 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 Toyota-47 ## 22 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2

```
Dodge212 ## 23 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2
AMC148 ## 24 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4
Camaro54 ## 25 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2
Pontiac306 ## 26 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1
Fiat83 ## 27 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2
Porsche289 ## 28 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2
Lotus139 ## 29 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4
Ford-111 ## 30 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6
Ferrari269 ## 31 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8
Maserati168 ## 32 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2
Volvo47 ## 33 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 Fiat83
## 34 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 Cadillac109 ##
35 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 Camaro54 ## 36
19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 Merc217 ## 37 15.2 8
304.0 150 3.15 3.435 17.30 0 0 3 2 AMC148 ## 38 30.4 4 75.7 52
4.93 1.615 18.52 1 1 4 2 Honda207 ## 39 21.0 6 160.0 110 3.90
2.875 17.02 0 1 4 4 Mazda86 ## 40 21.5 4 120.1 97 3.70 2.465
20.01 1 0 3 1 Toyota-47 ## 41 32.4 4 78.7 66 4.08 2.200 19.47
1 1 4 1 Fiat31 ## 42 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3
Merc259 ## 43 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4
Chrysler92 ## 44 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4
Mazda185 ## 45 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2
Hornet350 ## 46 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1
Datsun73 ## 47 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1
Valiant392 ## 48 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2
Lotus139
```

Data wrangling : Transforming (2/3) Solution

Below are the solutions to these exercises on data transformation.

```
#################### # # # Exercise 1 # # #
#################### mtcars$cyl <- as.factor(mtcars$cyl)
```

```
cars_cyl <- mtcars %>% group_by(cyl) ##################### # # # Exercise 2 # # # ##################### ungroup(cars_cyl)
```

```
## # A tibble: 32 x 11 ## mpg cyl disp hp drat wt qsec vs am gear carb ## * <dbl> <fctr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> ## 1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 ## 2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 ## 3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 ## 4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 ## 5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 ## 6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 ## 7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 ## 8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 ## 9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 ## 10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 ## # ... with 22 more rows
```

```
#OR cars_cyl %>% ungroup()
```

```
## # A tibble: 32 x 11 ## mpg cyl disp hp drat wt qsec vs am gear carb ## * <dbl> <fctr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> ## 1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 ## 2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 ## 3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 ## 4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 ## 5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 ## 6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 ## 7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 ## 8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 ## 9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 ## 10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 ## # ... with 22 more rows
```

```
##################### # # # Exercise 3 # # # ##################### mtcars %>% summary()
```

```
## mpg cyl disp hp drat ## Min. :10.40 4:11 Min. : 71.1 Min. : 52.0 Min. :2.760 ## 1st Qu.:15.43 6: 7 1st Qu.:120.8 1st Qu.: 96.5 1st Qu.:3.080 ## Median :19.20 8:14 Median :196.3 Median :123.0 Median :3.695 ## Mean :20.09 Mean :230.7 Mean :146.7 Mean :3.597 ## 3rd Qu.:22.80 3rd Qu.:326.0 3rd Qu.:180.0 3rd Qu.:3.920 ## Max. :33.90 Max. :472.0 Max. :335.0 Max. :4.930 ## wt qsec vs am ## Min. :1.513 Min. :14.50 Min. :0.0000 Min. :0.0000 ## 1st Qu.:2.581 1st Qu.:16.89 1st Qu.:0.0000 1st
```

Qu.:0.0000 ## Median :3.325 Median :17.71 Median :0.0000 Median :0.0000 ## Mean :3.217 Mean :17.85 Mean :0.4375 Mean :0.4062 ## 3rd Qu.:3.610 3rd Qu.:18.90 3rd Qu.:1.0000 3rd Qu.:1.0000 ## Max. :5.424 Max. :22.90 Max. :1.0000 Max. :1.0000 ## gear carb ## Min. :3.000 Min. :1.000 ## 1st Qu.:3.000 1st Qu.:2.000 ## Median :4.000 Median :2.000 ## Mean :3.688 Mean :2.812 ## 3rd Qu.:4.000 3rd Qu.:4.000 ## Max. :5.000 Max. :8.000

```
#####################  #  #  #  Exercise  4  #  #  #  ##################### mtcars %>% summarise(Min = min(hp), Quant_25 = quantile(hp, 0.25), Median = median(hp), Mean = mean(hp), Quant_75 = quantile(hp, 0.75), Max = max(hp), Std = sd(hp), Count = n())
```

Min Quant_25 Median Mean Quant_75 Max Std Count ## 1 52 96.5 123 146.6875 180 335 68.56287 32

```
#####################  #  #  #  Exercise  5  #  #  #  #####################  mtcars  %>%  group_by(cyl)  %>%  summarize(Mean = mean(hp), Std = sd(hp))
```

A tibble: 3 x 3 ## cyl Mean Std ## <fctr> <dbl> <dbl> ## 1 4 82.63636 20.93453 ## 2 6 122.28571 24.26049 ## 3 8 209.21429 50.97689

```
#####################  #  #  #  Exercise  6  #  #  #  ##################### mtcars %>% arrange(hp)
```

mpg cyl disp hp drat wt qsec vs am gear carb ## 1 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 ## 2 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 ## 3 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 ## 4 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 ## 5 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 ## 6 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 ## 7 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 ## 8 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 ## 9 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 ## 10 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 ## 11 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 ## 12 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 ## 13 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 ## 14 21.4 6 258.0

110 3.08 3.215 19.44 1 0 3 1 ## 15 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 ## 16 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 ## 17 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 ## 18 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 ## 19 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 ## 20 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 ## 21 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 ## 22 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 ## 23 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 ## 24 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 ## 25 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 ## 26 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 ## 27 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 ## 28 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 ## 29 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 ## 30 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 ## 31 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 ## 32 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8

```
##################### # # # Exercise 7 # # # ##################### mtcars %>% arrange(desc(hp))
```

mpg cyl disp hp drat wt qsec vs am gear carb ## 1 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 ## 2 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 ## 3 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 ## 4 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 ## 5 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 ## 6 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 ## 7 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 ## 8 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 ## 9 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 ## 10 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 ## 11 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 ## 12 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 ## 13 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 ## 14 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 ## 15 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 ## 16 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 ## 17 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 ## 18 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 ## 19 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 ## 20 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 ## 21 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 ## 22 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 ## 23 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 ## 24 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 ## 25 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 ## 26 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 ## 27 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 ## 28 32.4 4 78.7 66 4.08

2.200 19.47 1 1 4 1 ## 29 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 ## 30 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 ## 31 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 ## 32 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2

```
##################### # # # Exercise 8 # # # ##################### cars_per <- mtcars %>% mutate (performance = hp/mpg) ##################### # # # Exercise 9 # # # ##################### cars_per %>% arrange(desc(performance)) %>% mutate(rank = cumsum(rep(1,nrow(cars_per))))
```

mpg cyl disp hp drat wt qsec vs am gear carb performance rank ## 1 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 22.333333 1 ## 2 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 20.673077 2 ## 3 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 19.711538 3 ## 4 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 18.421053 4 ## 5 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 17.132867 5 ## 6 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 16.708861 6 ## 7 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 15.646259 7 ## 8 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 11.842105 8 ## 9 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 10.975610 9 ## 10 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 10.404624 10 ## 11 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 9.868421 11 ## 12 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 9.677419 12 ## 13 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 9.358289 13 ## 14 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 9.114583 14 ## 15 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 8.883249 15 ## 16 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 6.910112 16 ## 17 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 6.406250 17 ## 18 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 5.801105 18 ## 19 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 5.238095 19 ## 20 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 5.238095 20 ## 21 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 5.140187 21 ## 22 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 5.093458 22 ## 23 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 4.511628 23 ## 24 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 4.166667 24 ## 25 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 4.078947 25 ## 26 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 3.717105 26 ## 27 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 3.500000 27 ## 28 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 2.540984 28 ## 29

```
27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 2.417582 29 ## 30 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 2.037037 30 ## 31 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 1.917404 31 ## 32 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 1.710526 32

##################### # # # Exercise 10 # # # ##################### iris %>% group_by(Species) %>% arrange(Sepal.Length) %>% summarize(Sepal.Length = mean(Sepal.Length), Sepal.Width = mean(Sepal.Width), Petal.Length = mean(Petal.Length), Petal.Width = mean(Petal.Width)) %>% mutate(Sepal.Density = Sepal.Length *Sepal.Width, Petal.Density = Sepal.Length * Petal.Width)

## # A tibble: 3 x 7 ## Species Sepal.Length Sepal.Width Petal.Length Petal.Width ## <fctr> <dbl> <dbl> <dbl> <dbl> ## 1 setosa 5.006 3.428 1.462 0.246 ## 2 versicolor 5.936 2.770 4.260 1.326 ## 3 virginica 6.588 2.974 5.552 2.026 ## # ... with 2 more variables: Sepal.Density <dbl>, Petal.Density <dbl>
```

Data wrangling : Transforming (2/3)

Data wrangling is a task of great importance in data analysis. Data wrangling, is the process of importing, cleaning and transforming raw data into actionable information for analysis. It is a time-consuming process which is estimated to take about 60-80% of analyst's time. In this series we will go through this process. It will be a brief series with goal to craft the reader's skills on the data wrangling task. This is the third part of the series

and it aims to cover the transforming of data used.This can include filtering, summarizing, and ordering your data by different means. This also includes combining various data sets, creating new variables, and many other manipulation tasks. At this post, we will go through a few more advanced transformation tasks on mtcars data set.

Before proceeding, it might be helpful to look over the help pages for the group_by, ungrpoup, summary, summarise, arrange, mutate, cumsum.

Moreover please load the following libraries.
install.packages("dplyr")
library(dplyr)

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Create a new object named *cars_cyl* and assign to it the *mtcars* data frame grouped by the variable *cyl*
Hint: be careful about the data type of the variable, in order to be used for grouping it has to be a factor.

Exercise 2

Remove the grouping from the object *cars_cyl*

Exercise 3

Print out the summary statistics of the *mtcars* data frame using the summary function and pipeline symbols %>%.

this course you will learn how to:

- Work with popular libraries such as dplyr
- Learn about methods such as pipelines
- And much more

Exercise 4

Make a more descriptive summary statistics output containing the 4 quantiles, the mean, the standard deviation and the count.

Exercise 5

Print out the average *hp* for every *cyl* category

Exercise 6

Print out the *mtcars* data frame sorted by *hp* (ascending oder)

Exercise 7

Print out the *mtcars* data frame sorted by *hp* (descending oder)

Exercise 8

Create a new object named *cars_per* containing the *mtcars* data frame along with a new variable called *performance* and calculated as performance = hp/mpg

Exercise 9

Print out the cars_per data frame, sorted by *performance* in descending order and create a new variable called *rank* indicating the rank of the cars in terms of performance.

Exercise 10

To wrap everything up, we will use the *iris* data set. Print out the mean of every variable for every *Species* and create two new variables called *Sepal.Density* and *Petal.Density* being

calculated as Sepal.Density = Sepal.Length Sepal.Width and Petal.Density = Sepal.Length Petal.Width respectively.

Data wrangling : Transforming (1/3)

Data wrangling is a task of great importance in data analysis. Data wrangling, is the process of importing, cleaning and transforming raw data into actionable information for analysis. It is a time-consuming process which is estimated to take about 60-80% of analyst's time. In this series we will go through this process. It will be a brief series with goal to craft the reader's skills on the data wrangling task. This is the third part of the series and it aims to cover the transforming of data used.This can include filtering, summarizing, and ordering your data by different means. This also includes combining various data sets, creating new variables, and many other manipulation tasks. At this post, we will go through the most basic tasks including slicing, and filtering on the famous mtcars data set.

Before proceeding, it might be helpful to look over the help pages for the select, rename, sample_frac, slice, distinct, filter, rownames, %in%.

Moreover please load the following libraries.

```
install.packages("dplyr")
library(dplyr)
```

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Print out the *hp* column using the select function.

Exercise 2

Print out the all **but** *hp* column using the select function.

Exercise 3

Print out the *mpg, hp, vs, am, gear* columns. Consider using the colon (:) symbol.

Exercise 4

Create the object **cars_m_h** containing the columns *mpg, hp* columns but let the column names be 'miles_per_gallon', and 'horse_power' respectively.

Exercise 5

Change the column names of **cars_m_h** from 'miles_per_gallon', and 'horse_power' to 'mpg' and 'hp' respectively.

Exercise 6

Print out a randomly half the observations of **cars_m_h**.
Hint : consider using the sample_frac function

Exercise 7

Create a cars_m_h_s object, containing from 10th to 35th row of **cars_m_h**.
Hint : Consider using the slice function.

Exercise 8

Print out the **cars_m_h_s** object without any duplicates.
Hint : Consider using the distinct function.

Exercise 9

Print out from **cars_m_h_s** object all the observations which have mpg>20 and hp>100.

Exercise 10

Select the ‘Lotus Europa’ car.

Data wrangling : Transforming (1/3) Solution

Below are the solutions to these exercises on data transoformation.

```
#####################  #  #  #  Exercise  1  #  #  #
#################### mtcars %>% select(hp)
```

```
## hp ## Mazda RX4 110 ## Mazda RX4 Wag 110 ## Datsun 710 93
## Hornet 4 Drive 110 ## Hornet Sportabout 175 ## Valiant 105
## Duster 360 245 ## Merc 240D 62 ## Merc 230 95 ## Merc 280
123 ## Merc 280C 123 ## Merc 450SE 180 ## Merc 450SL 180 ##
Merc 450SLC 180 ## Cadillac Fleetwood 205 ## Lincoln
Continental 215 ## Chrysler Imperial 230 ## Fiat 128 66 ##
Honda Civic 52 ## Toyota Corolla 65 ## Toyota Corona 97 ##
Dodge Challenger 150 ## AMC Javelin 150 ## Camaro Z28 245 ##
Pontiac Firebird 175 ## Fiat X1-9 66 ## Porsche 914-2 91 ##
Lotus Europa 113 ## Ford Pantera L 264 ## Ferrari Dino 175 ##
Maserati Bora 335 ## Volvo 142E 109
```

```
#####################  #  #  #  Exercise  2  #  #  #
```

```
#################### mtcars %>% select(-hp)
```

mpg cyl disp drat wt qsec vs am gear carb ## Mazda RX4 21.0 6 160.0 3.90 2.620 16.46 0 1 4 4 ## Mazda RX4 Wag 21.0 6 160.0 3.90 2.875 17.02 0 1 4 4 ## Datsun 710 22.8 4 108.0 3.85 2.320 18.61 1 1 4 1 ## Hornet 4 Drive 21.4 6 258.0 3.08 3.215 19.44 1 0 3 1 ## Hornet Sportabout 18.7 8 360.0 3.15 3.440 17.02 0 0 3 2 ## Valiant 18.1 6 225.0 2.76 3.460 20.22 1 0 3 1 ## Duster 360 14.3 8 360.0 3.21 3.570 15.84 0 0 3 4 ## Merc 240D 24.4 4 146.7 3.69 3.190 20.00 1 0 4 2 ## Merc 230 22.8 4 140.8 3.92 3.150 22.90 1 0 4 2 ## Merc 280 19.2 6 167.6 3.92 3.440 18.30 1 0 4 4 ## Merc 280C 17.8 6 167.6 3.92 3.440 18.90 1 0 4 4 ## Merc 450SE 16.4 8 275.8 3.07 4.070 17.40 0 0 3 3 ## Merc 450SL 17.3 8 275.8 3.07 3.730 17.60 0 0 3 3 ## Merc 450SLC 15.2 8 275.8 3.07 3.780 18.00 0 0 3 3 ## Cadillac Fleetwood 10.4 8 472.0 2.93 5.250 17.98 0 0 3 4 ## Lincoln Continental 10.4 8 460.0 3.00 5.424 17.82 0 0 3 4 ## Chrysler Imperial 14.7 8 440.0 3.23 5.345 17.42 0 0 3 4 ## Fiat 128 32.4 4 78.7 4.08 2.200 19.47 1 1 4 1 ## Honda Civic 30.4 4 75.7 4.93 1.615 18.52 1 1 4 2 ## Toyota Corolla 33.9 4 71.1 4.22 1.835 19.90 1 1 4 1 ## Toyota Corona 21.5 4 120.1 3.70 2.465 20.01 1 0 3 1 ## Dodge Challenger 15.5 8 318.0 2.76 3.520 16.87 0 0 3 2 ## AMC Javelin 15.2 8 304.0 3.15 3.435 17.30 0 0 3 2 ## Camaro Z28 13.3 8 350.0 3.73 3.840 15.41 0 0 3 4 ## Pontiac Firebird 19.2 8 400.0 3.08 3.845 17.05 0 0 3 2 ## Fiat X1-9 27.3 4 79.0 4.08 1.935 18.90 1 1 4 1 ## Porsche 914-2 26.0 4 120.3 4.43 2.140 16.70 0 1 5 2 ## Lotus Europa 30.4 4 95.1 3.77 1.513 16.90 1 1 5 2 ## Ford Pantera L 15.8 8 351.0 4.22 3.170 14.50 0 1 5 4 ## Ferrari Dino 19.7 6 145.0 3.62 2.770 15.50 0 1 5 6 ## Maserati Bora 15.0 8 301.0 3.54 3.570 14.60 0 1 5 8 ## Volvo 142E 21.4 4 121.0 4.11 2.780 18.60 1 1 4 2

```
#################### # # # Exercise 3 # # # #################### mtcars %>% select(mpg, hp, vs:gear)
```

mpg hp vs am gear ## Mazda RX4 21.0 110 0 1 4 ## Mazda RX4 Wag 21.0 110 0 1 4 ## Datsun 710 22.8 93 1 1 4 ## Hornet 4 Drive 21.4 110 1 0 3 ## Hornet Sportabout 18.7 175 0 0 3 ## Valiant 18.1 105 1 0 3 ## Duster 360 14.3 245 0 0 3 ## Merc 240D 24.4 62 1 0 4 ## Merc 230 22.8 95 1 0 4 ## Merc 280 19.2 123 1 0 4 ## Merc 280C 17.8 123 1 0 4 ## Merc 450SE 16.4 180 0

0 3 ## Merc 450SL 17.3 180 0 0 3 ## Merc 450SLC 15.2 180 0 0 3 ## Cadillac Fleetwood 10.4 205 0 0 3 ## Lincoln Continental 10.4 215 0 0 3 ## Chrysler Imperial 14.7 230 0 0 3 ## Fiat 128 32.4 66 1 1 4 ## Honda Civic 30.4 52 1 1 4 ## Toyota Corolla 33.9 65 1 1 4 ## Toyota Corona 21.5 97 1 0 3 ## Dodge Challenger 15.5 150 0 0 3 ## AMC Javelin 15.2 150 0 0 3 ## Camaro Z28 13.3 245 0 0 3 ## Pontiac Firebird 19.2 175 0 0 3 ## Fiat X1-9 27.3 66 1 1 4 ## Porsche 914-2 26.0 91 0 1 5 ## Lotus Europa 30.4 113 1 1 5 ## Ford Pantera L 15.8 264 0 1 5 ## Ferrari Dino 19.7 175 0 1 5 ## Maserati Bora 15.0 335 0 1 5 ## Volvo 142E 21.4 109 1 1 4

```
#####################  #  #  #  Exercise  4  #  #  #  ##################### cars_m_h <- mtcars %>% select(miles_per_gallon = mpg, horse_power = hp) #####################  #  #  #  Exercise  5  #  #  #  ##################### cars_m_h <- cars_m_h %>% rename('mpg' = 'miles_per_gallon', 'hp'= 'horse_power') ##################### # # # Exercise 6 # # # ##################### cars_m_h %>% sample_frac(size = 0.5, replace = FALSE)
```

mpg hp ## Dodge Challenger 15.5 150 ## Toyota Corona 21.5 97 ## Cadillac Fleetwood 10.4 205 ## Ford Pantera L 15.8 264 ## Honda Civic 30.4 52 ## Chrysler Imperial 14.7 230 ## Camaro Z28 13.3 245 ## Duster 360 14.3 245 ## Volvo 142E 21.4 109 ## Merc 450SL 17.3 180 ## Fiat 128 32.4 66 ## Merc 280C 17.8 123 ## Merc 230 22.8 95 ## Lincoln Continental 10.4 215 ## Pontiac Firebird 19.2 175 ## AMC Javelin 15.2 150

```
#####################  #  #  #  Exercise  7  #  #  #  ##################### cars_m_h_s <- cars_m_h %>% slice(10:35) #####################  #  #  #  Exercise  8  #  #  #  ##################### cars_m_h_s %>% distinct()
```

A tibble: 23 x 2 ## mpg hp ## <dbl> <dbl> ## 1 19.2 123 ## 2 17.8 123 ## 3 16.4 180 ## 4 17.3 180 ## 5 15.2 180 ## 6 10.4 205 ## 7 10.4 215 ## 8 14.7 230 ## 9 32.4 66 ## 10 30.4 52 ## # ... with 13 more rows

```
#####################  #  #  #  Exercise  9  #  #  #
```

```
#################### cars_m_h_s %>% filter(mpg>20, hp >100)

## # A tibble: 2 x 2 ## mpg hp ## <dbl> <dbl> ## 1 30.4 113 ## 2 21.4 109

#####################  #  #  #  Exercise  10  #  #  # #################### mtcars %>% filter(rownames(mtcars) == "Lotus Europa")

## mpg cyl disp hp drat wt qsec vs am gear carb ## 1 30.4 4 95.1 113 3.77 1.513 16.9 1 1 5 2

#OR mtcars %>% filter(rownames(mtcars) %in% "Lotus Europa")

## mpg cyl disp hp drat wt qsec vs am gear carb ## 1 30.4 4 95.1 113 3.77 1.513 16.9 1 1 5 2
```

Data wrangling : Reshaping

Data wrangling is a task of great importance in data analysis. Data wrangling, is the process of importing, cleaning and transforming raw data into actionable information for analysis. It is a time-consuming process which is estimated to take about 60-80% of analyst’s time. In this series we will go through this process. It will be a brief series with goal to craft the reader’s skills on the data wrangling task. This is the second part of this series and it aims to cover the reshaping of data used to turn them into a tidy form. By tidy form, we mean that each feature forms a column and each observation forms a row.

Before proceeding, it might be helpful to look over the help pages for the spread, gather, unite, separate, replace_na, fill, extract_numeric.

Moreover please load the following libraries.

```
install.packages("magrittr")
library(magrittr)
install.packages("tidyr")
library(tidyr)
```

Please run the code below in order to load the data set:

```
data <- airquality[4:6]
```

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Print out the structure of the data frame.

Exercise 2

Let’s turn the data frame in a wider form, from above and and turn the Month variable into column headings and spread the Temp values across the months they are related to.

Exercise 3

Turn the wide (exercise 2) data frame into its initial format using the gather function, specify the columns you would like to gather by index number.

Exercise 4

Turn the wide (exercise 2) data frame into its initial format using the gather function, specify the columns you would like to gather by column name.

Learn more about Data Pre-Processing in the online course R Data Pre-Processing & Data Management – Shape your Data!. In this course you will learn how to:

- import data into R in several ways while also beeing able to identify a suitable import tool
- use SQL code within R
- And much more

Exercise 5

Turn the wide (exercise 2) data frame into its initial format using the gather function, specify the columns by using remaining column names(the ones you don't use for gathering).

Exercise 6

Unite the variables Day and Month to a new feature named Date with the format %d-%m .

Exercise 7

Create the data frame at its previous format (exercise 6). Separate the variable you have created before (Date) to Day, Month.

Exercise 8

Replace the missing values (NA) with 'Unknown'.

Exercise 9

Run the script below, so that you make a new feature year.

```
back2long_na$year <- rep(NA, nrow(back2long_na))

back2long_na$year[1] <- '2015'
back2long_na$year[as.integer(nrow(back2long_na)/3)] <- '2016'

back2long_na$year[as.integer(2*nrow(back2long_na)/3)]   <-
'2017'
```

You have noticed, that the new column has many values. Fill the NAs with the non-missing value write above it. (eg.the NA's that are below the '2016' and '2017' value assign it to '2016'.

Hint: use the fill function.

Exercise 10

Extract the numeric values from the Temp feature.

Hint: extract_numeric, this is a very important function when the variable we apply the function on is a character with 'noise', for example '$40' and you want to transform it to 40.

Data wrangling : Reshaping Solution

Below are the solutions to these exercises on data reshaping.

```
#####################  #  #  #  Exercise  1  #  #  ##################### str(data)
```

'data.frame': 153 obs. of 3 variables: ## $ Temp : int 67 72 74 62 56 66 65 59 61 69 ... ## $ Month: int 5 5 5 5 5 5 5 5 5 5 ... ## $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

```
#####################  #  #  #  Exercise  2  #  #  ##################### long2wide <- data %>% spread(Month, Temp); long2wide
```

Day 5 6 7 8 9 ## 1 1 67 78 84 81 91 ## 2 2 72 74 85 81 92 ## 3 3 74 67 81 82 93 ## 4 4 62 84 84 86 93 ## 5 5 56 85 83 85 87 ## 6 6 66 79 83 87 84 ## 7 7 65 82 88 89 80 ## 8 8 59 87 92

90 78 ## 9 9 61 90 92 90 75 ## 10 10 69 87 89 92 73 ## 11 11 74 93 82 86 81 ## 12 12 69 92 73 86 76 ## 13 13 66 82 81 82 77 ## 14 14 68 80 91 80 71 ## 15 15 58 79 80 79 71 ## 16 16 64 77 81 77 78 ## 17 17 66 72 82 79 67 ## 18 18 57 65 84 76 76 ## 19 19 68 73 87 78 68 ## 20 20 62 76 85 78 82 ## 21 21 59 77 74 77 64 ## 22 22 73 76 81 72 71 ## 23 23 61 76 82 75 81 ## 24 24 61 76 86 79 69 ## 25 25 57 75 85 81 63 ## 26 26 58 78 82 86 70 ## 27 27 57 73 86 88 77 ## 28 28 67 80 88 97 75 ## 29 29 81 77 86 94 76 ## 30 30 79 83 83 96 68 ## 31 31 76 NA 81 94 NA

########## # # # Exercise 3 # # # ########## back2long <- long2wide %>% gather(Month, Temp, 2:6); back2long

Day Month Temp ## 1 1 5 67 ## 2 2 5 72 ## 3 3 5 74 ## 4 4 5 62 ## 5 5 5 56 ## 6 6 5 66 ## 7 7 5 65 ## 8 8 5 59 ## 9 9 5 61 ## 10 10 5 69 ## 11 11 5 74 ## 12 12 5 69 ## 13 13 5 66 ## 14 14 5 68 ## 15 15 5 58 ## 16 16 5 64 ## 17 17 5 66 ## 18 18 5 57 ## 19 19 5 68 ## 20 20 5 62 ## 21 21 5 59 ## 22 22 5 73 ## 23 23 5 61 ## 24 24 5 61 ## 25 25 5 57 ## 26 26 5 58 ## 27 27 5 57 ## 28 28 5 67 ## 29 29 5 81 ## 30 30 5 79 ## 31 31 5 76 ## 32 1 6 78 ## 33 2 6 74 ## 34 3 6 67 ## 35 4 6 84 ## 36 5 6 85 ## 37 6 6 79 ## 38 7 6 82 ## 39 8 6 87 ## 40 9 6 90 ## 41 10 6 87 ## 42 11 6 93 ## 43 12 6 92 ## 44 13 6 82 ## 45 14 6 80 ## 46 15 6 79 ## 47 16 6 77 ## 48 17 6 72 ## 49 18 6 65 ## 50 19 6 73 ## 51 20 6 76 ## 52 21 6 77 ## 53 22 6 76 ## 54 23 6 76 ## 55 24 6 76 ## 56 25 6 75 ## 57 26 6 78 ## 58 27 6 73 ## 59 28 6 80 ## 60 29 6 77 ## 61 30 6 83 ## 62 31 6 NA ## 63 1 7 84 ## 64 2 7 85 ## 65 3 7 81 ## 66 4 7 84 ## 67 5 7 83 ## 68 6 7 83 ## 69 7 7 88 ## 70 8 7 92 ## 71 9 7 92 ## 72 10 7 89 ## 73 11 7 82 ## 74 12 7 73 ## 75 13 7 81 ## 76 14 7 91 ## 77 15 7 80 ## 78 16 7 81 ## 79 17 7 82 ## 80 18 7 84 ## 81 19 7 87 ## 82 20 7 85 ## 83 21 7 74 ## 84 22 7 81 ## 85 23 7 82 ## 86 24 7 86 ## 87 25 7 85 ## 88 26 7 82 ## 89 27 7 86 ## 90 28 7 88 ## 91 29 7 86 ## 92 30 7 83 ## 93 31 7 81 ## 94 1 8 81 ## 95 2 8 81 ## 96 3 8 82 ## 97 4 8 86 ## 98 5 8 85 ## 99 6 8 87 ## 100 7 8 89 ## 101 8 8 90 ## 102 9 8 90 ## 103 10 8 92 ## 104 11 8 86 ## 105 12 8 86 ## 106 13 8 82 ## 107 14 8 80 ## 108 15 8 79 ## 109 16 8 77 ## 110 17 8 79 ## 111 18 8 76 ## 112 19 8 78 ## 113 20 8 78 ## 114 21 8 77 ## 115 22 8 72 ## 116 23 8 75 ## 117 24 8 79 ## 118 25 8 81 ## 119 26 8 86

120 27 8 88 ## 121 28 8 97 ## 122 29 8 94 ## 123 30 8 96 ## 124 31 8 94 ## 125 1 9 91 ## 126 2 9 92 ## 127 3 9 93 ## 128 4 9 93 ## 129 5 9 87 ## 130 6 9 84 ## 131 7 9 80 ## 132 8 9 78 ## 133 9 9 75 ## 134 10 9 73 ## 135 11 9 81 ## 136 12 9 76 ## 137 13 9 77 ## 138 14 9 71 ## 139 15 9 71 ## 140 16 9 78 ## 141 17 9 67 ## 142 18 9 76 ## 143 19 9 68 ## 144 20 9 82 ## 145 21 9 64 ## 146 22 9 71 ## 147 23 9 81 ## 148 24 9 69 ## 149 25 9 63 ## 150 26 9 70 ## 151 27 9 77 ## 152 28 9 75 ## 153 29 9 76 ## 154 30 9 68 ## 155 31 9 NA

########## # # # Exercise 4 # # # ########## back2long <- long2wide %>% gather(Month, Temp, "5", "6", "7", "8", "9"); back2long

Day Month Temp ## 1 1 5 67 ## 2 2 5 72 ## 3 3 5 74 ## 4 4 5 62 ## 5 5 5 56 ## 6 6 5 66 ## 7 7 5 65 ## 8 8 5 59 ## 9 9 5 61 ## 10 10 5 69 ## 11 11 5 74 ## 12 12 5 69 ## 13 13 5 66 ## 14 14 5 68 ## 15 15 5 58 ## 16 16 5 64 ## 17 17 5 66 ## 18 18 5 57 ## 19 19 5 68 ## 20 20 5 62 ## 21 21 5 59 ## 22 22 5 73 ## 23 23 5 61 ## 24 24 5 61 ## 25 25 5 57 ## 26 26 5 58 ## 27 27 5 57 ## 28 28 5 67 ## 29 29 5 81 ## 30 30 5 79 ## 31 31 5 76 ## 32 1 6 78 ## 33 2 6 74 ## 34 3 6 67 ## 35 4 6 84 ## 36 5 6 85 ## 37 6 6 79 ## 38 7 6 82 ## 39 8 6 87 ## 40 9 6 90 ## 41 10 6 87 ## 42 11 6 93 ## 43 12 6 92 ## 44 13 6 82 ## 45 14 6 80 ## 46 15 6 79 ## 47 16 6 77 ## 48 17 6 72 ## 49 18 6 65 ## 50 19 6 73 ## 51 20 6 76 ## 52 21 6 77 ## 53 22 6 76 ## 54 23 6 76 ## 55 24 6 76 ## 56 25 6 75 ## 57 26 6 78 ## 58 27 6 73 ## 59 28 6 80 ## 60 29 6 77 ## 61 30 6 83 ## 62 31 6 NA ## 63 1 7 84 ## 64 2 7 85 ## 65 3 7 81 ## 66 4 7 84 ## 67 5 7 83 ## 68 6 7 83 ## 69 7 7 88 ## 70 8 7 92 ## 71 9 7 92 ## 72 10 7 89 ## 73 11 7 82 ## 74 12 7 73 ## 75 13 7 81 ## 76 14 7 91 ## 77 15 7 80 ## 78 16 7 81 ## 79 17 7 82 ## 80 18 7 84 ## 81 19 7 87 ## 82 20 7 85 ## 83 21 7 74 ## 84 22 7 81 ## 85 23 7 82 ## 86 24 7 86 ## 87 25 7 85 ## 88 26 7 82 ## 89 27 7 86 ## 90 28 7 88 ## 91 29 7 86 ## 92 30 7 83 ## 93 31 7 81 ## 94 1 8 81 ## 95 2 8 81 ## 96 3 8 82 ## 97 4 8 86 ## 98 5 8 85 ## 99 6 8 87 ## 100 7 8 89 ## 101 8 8 90 ## 102 9 8 90 ## 103 10 8 92 ## 104 11 8 86 ## 105 12 8 86 ## 106 13 8 82 ## 107 14 8 80 ## 108 15 8 79 ## 109 16 8 77 ## 110 17 8 79 ## 111 18 8 76 ## 112 19 8 78 ## 113 20 8 78 ## 114 21 8 77 ## 115 22 8 72 ## 116 23 8 75 ## 117 24 8 79 ## 118 25 8 81 ## 119 26 8 86

120 27 8 88 ## 121 28 8 97 ## 122 29 8 94 ## 123 30 8 96 ## 124 31 8 94 ## 125 1 9 91 ## 126 2 9 92 ## 127 3 9 93 ## 128 4 9 93 ## 129 5 9 87 ## 130 6 9 84 ## 131 7 9 80 ## 132 8 9 78 ## 133 9 9 75 ## 134 10 9 73 ## 135 11 9 81 ## 136 12 9 76 ## 137 13 9 77 ## 138 14 9 71 ## 139 15 9 71 ## 140 16 9 78 ## 141 17 9 67 ## 142 18 9 76 ## 143 19 9 68 ## 144 20 9 82 ## 145 21 9 64 ## 146 22 9 71 ## 147 23 9 81 ## 148 24 9 69 ## 149 25 9 63 ## 150 26 9 70 ## 151 27 9 77 ## 152 28 9 75 ## 153 29 9 76 ## 154 30 9 68 ## 155 31 9 NA

########## # # # Exercise 5 # # # ########## back2long <- long2wide %>% gather(Month, Temp, -Day); back2long

Day Month Temp ## 1 1 5 67 ## 2 2 5 72 ## 3 3 5 74 ## 4 4 5 62 ## 5 5 5 56 ## 6 6 5 66 ## 7 7 5 65 ## 8 8 5 59 ## 9 9 5 61 ## 10 10 5 69 ## 11 11 5 74 ## 12 12 5 69 ## 13 13 5 66 ## 14 14 5 68 ## 15 15 5 58 ## 16 16 5 64 ## 17 17 5 66 ## 18 18 5 57 ## 19 19 5 68 ## 20 20 5 62 ## 21 21 5 59 ## 22 22 5 73 ## 23 23 5 61 ## 24 24 5 61 ## 25 25 5 57 ## 26 26 5 58 ## 27 27 5 57 ## 28 28 5 67 ## 29 29 5 81 ## 30 30 5 79 ## 31 31 5 76 ## 32 1 6 78 ## 33 2 6 74 ## 34 3 6 67 ## 35 4 6 84 ## 36 5 6 85 ## 37 6 6 79 ## 38 7 6 82 ## 39 8 6 87 ## 40 9 6 90 ## 41 10 6 87 ## 42 11 6 93 ## 43 12 6 92 ## 44 13 6 82 ## 45 14 6 80 ## 46 15 6 79 ## 47 16 6 77 ## 48 17 6 72 ## 49 18 6 65 ## 50 19 6 73 ## 51 20 6 76 ## 52 21 6 77 ## 53 22 6 76 ## 54 23 6 76 ## 55 24 6 76 ## 56 25 6 75 ## 57 26 6 78 ## 58 27 6 73 ## 59 28 6 80 ## 60 29 6 77 ## 61 30 6 83 ## 62 31 6 NA ## 63 1 7 84 ## 64 2 7 85 ## 65 3 7 81 ## 66 4 7 84 ## 67 5 7 83 ## 68 6 7 83 ## 69 7 7 88 ## 70 8 7 92 ## 71 9 7 92 ## 72 10 7 89 ## 73 11 7 82 ## 74 12 7 73 ## 75 13 7 81 ## 76 14 7 91 ## 77 15 7 80 ## 78 16 7 81 ## 79 17 7 82 ## 80 18 7 84 ## 81 19 7 87 ## 82 20 7 85 ## 83 21 7 74 ## 84 22 7 81 ## 85 23 7 82 ## 86 24 7 86 ## 87 25 7 85 ## 88 26 7 82 ## 89 27 7 86 ## 90 28 7 88 ## 91 29 7 86 ## 92 30 7 83 ## 93 31 7 81 ## 94 1 8 81 ## 95 2 8 81 ## 96 3 8 82 ## 97 4 8 86 ## 98 5 8 85 ## 99 6 8 87 ## 100 7 8 89 ## 101 8 8 90 ## 102 9 8 90 ## 103 10 8 92 ## 104 11 8 86 ## 105 12 8 86 ## 106 13 8 82 ## 107 14 8 80 ## 108 15 8 79 ## 109 16 8 77 ## 110 17 8 79 ## 111 18 8 76 ## 112 19 8 78 ## 113 20 8 78 ## 114 21 8 77 ## 115 22 8 72 ## 116 23 8 75 ## 117 24 8 79 ## 118 25 8 81 ## 119 26 8 86

120 27 8 88 ## 121 28 8 97 ## 122 29 8 94 ## 123 30 8 96 ## 124 31 8 94 ## 125 1 9 91 ## 126 2 9 92 ## 127 3 9 93 ## 128 4 9 93 ## 129 5 9 87 ## 130 6 9 84 ## 131 7 9 80 ## 132 8 9 78 ## 133 9 9 75 ## 134 10 9 73 ## 135 11 9 81 ## 136 12 9 76 ## 137 13 9 77 ## 138 14 9 71 ## 139 15 9 71 ## 140 16 9 78 ## 141 17 9 67 ## 142 18 9 76 ## 143 19 9 68 ## 144 20 9 82 ## 145 21 9 64 ## 146 22 9 71 ## 147 23 9 81 ## 148 24 9 69 ## 149 25 9 63 ## 150 26 9 70 ## 151 27 9 77 ## 152 28 9 75 ## 153 29 9 76 ## 154 30 9 68 ## 155 31 9 NA

```
##################### # # # Exercise 6 # # # #####################
back2long_unite <- back2long %>% unite(col = "Date", c(Day, Month), sep = "-"); back2long_unite
```

Date Temp ## 1 1-5 67 ## 2 2-5 72 ## 3 3-5 74 ## 4 4-5 62 ## 5 5-5 56 ## 6 6-5 66 ## 7 7-5 65 ## 8 8-5 59 ## 9 9-5 61 ## 10 10-5 69 ## 11 11-5 74 ## 12 12-5 69 ## 13 13-5 66 ## 14 14-5 68 ## 15 15-5 58 ## 16 16-5 64 ## 17 17-5 66 ## 18 18-5 57 ## 19 19-5 68 ## 20 20-5 62 ## 21 21-5 59 ## 22 22-5 73 ## 23 23-5 61 ## 24 24-5 61 ## 25 25-5 57 ## 26 26-5 58 ## 27 27-5 57 ## 28 28-5 67 ## 29 29-5 81 ## 30 30-5 79 ## 31 31-5 76 ## 32 1-6 78 ## 33 2-6 74 ## 34 3-6 67 ## 35 4-6 84 ## 36 5-6 85 ## 37 6-6 79 ## 38 7-6 82 ## 39 8-6 87 ## 40 9-6 90 ## 41 10-6 87 ## 42 11-6 93 ## 43 12-6 92 ## 44 13-6 82 ## 45 14-6 80 ## 46 15-6 79 ## 47 16-6 77 ## 48 17-6 72 ## 49 18-6 65 ## 50 19-6 73 ## 51 20-6 76 ## 52 21-6 77 ## 53 22-6 76 ## 54 23-6 76 ## 55 24-6 76 ## 56 25-6 75 ## 57 26-6 78 ## 58 27-6 73 ## 59 28-6 80 ## 60 29-6 77 ## 61 30-6 83 ## 62 31-6 NA ## 63 1-7 84 ## 64 2-7 85 ## 65 3-7 81 ## 66 4-7 84 ## 67 5-7 83 ## 68 6-7 83 ## 69 7-7 88 ## 70 8-7 92 ## 71 9-7 92 ## 72 10-7 89 ## 73 11-7 82 ## 74 12-7 73 ## 75 13-7 81 ## 76 14-7 91 ## 77 15-7 80 ## 78 16-7 81 ## 79 17-7 82 ## 80 18-7 84 ## 81 19-7 87 ## 82 20-7 85 ## 83 21-7 74 ## 84 22-7 81 ## 85 23-7 82 ## 86 24-7 86 ## 87 25-7 85 ## 88 26-7 82 ## 89 27-7 86 ## 90 28-7 88 ## 91 29-7 86 ## 92 30-7 83 ## 93 31-7 81 ## 94 1-8 81 ## 95 2-8 81 ## 96 3-8 82 ## 97 4-8 86 ## 98 5-8 85 ## 99 6-8 87 ## 100 7-8 89 ## 101 8-8 90 ## 102 9-8 90 ## 103 10-8 92 ## 104 11-8 86 ## 105 12-8 86 ## 106 13-8 82 ## 107 14-8 80 ## 108 15-8 79 ## 109 16-8 77 ## 110 17-8 79 ## 111 18-8 76 ## 112 19-8 78 ## 113 20-8 78 ## 114 21-8 77 ## 115 22-8 72 ## 116 23-8 75 ## 117 24-8 79 ## 118 25-8 81

119 26-8 86 ## 120 27-8 88 ## 121 28-8 97 ## 122 29-8 94 ## 123 30-8 96 ## 124 31-8 94 ## 125 1-9 91 ## 126 2-9 92 ## 127 3-9 93 ## 128 4-9 93 ## 129 5-9 87 ## 130 6-9 84 ## 131 7-9 80 ## 132 8-9 78 ## 133 9-9 75 ## 134 10-9 73 ## 135 11-9 81 ## 136 12-9 76 ## 137 13-9 77 ## 138 14-9 71 ## 139 15-9 71 ## 140 16-9 78 ## 141 17-9 67 ## 142 18-9 76 ## 143 19-9 68 ## 144 20-9 82 ## 145 21-9 64 ## 146 22-9 71 ## 147 23-9 81 ## 148 24-9 69 ## 149 25-9 63 ## 150 26-9 70 ## 151 27-9 77 ## 152 28-9 75 ## 153 29-9 76 ## 154 30-9 68 ## 155 31-9 NA

```
##################### # # # Exercise 7 # # # ##################### back2long_separate <- back2long_unite %>% separate(col = Date, into = c("Day", "Month")) ; back2long_separate
```

Day Month Temp ## 1 1 5 67 ## 2 2 5 72 ## 3 3 5 74 ## 4 4 5 62 ## 5 5 5 56 ## 6 6 5 66 ## 7 7 5 65 ## 8 8 5 59 ## 9 9 5 61 ## 10 10 5 69 ## 11 11 5 74 ## 12 12 5 69 ## 13 13 5 66 ## 14 14 5 68 ## 15 15 5 58 ## 16 16 5 64 ## 17 17 5 66 ## 18 18 5 57 ## 19 19 5 68 ## 20 20 5 62 ## 21 21 5 59 ## 22 22 5 73 ## 23 23 5 61 ## 24 24 5 61 ## 25 25 5 57 ## 26 26 5 58 ## 27 27 5 57 ## 28 28 5 67 ## 29 29 5 81 ## 30 30 5 79 ## 31 31 5 76 ## 32 1 6 78 ## 33 2 6 74 ## 34 3 6 67 ## 35 4 6 84 ## 36 5 6 85 ## 37 6 6 79 ## 38 7 6 82 ## 39 8 6 87 ## 40 9 6 90 ## 41 10 6 87 ## 42 11 6 93 ## 43 12 6 92 ## 44 13 6 82 ## 45 14 6 80 ## 46 15 6 79 ## 47 16 6 77 ## 48 17 6 72 ## 49 18 6 65 ## 50 19 6 73 ## 51 20 6 76 ## 52 21 6 77 ## 53 22 6 76 ## 54 23 6 76 ## 55 24 6 76 ## 56 25 6 75 ## 57 26 6 78 ## 58 27 6 73 ## 59 28 6 80 ## 60 29 6 77 ## 61 30 6 83 ## 62 31 6 NA ## 63 1 7 84 ## 64 2 7 85 ## 65 3 7 81 ## 66 4 7 84 ## 67 5 7 83 ## 68 6 7 83 ## 69 7 7 88 ## 70 8 7 92 ## 71 9 7 92 ## 72 10 7 89 ## 73 11 7 82 ## 74 12 7 73 ## 75 13 7 81 ## 76 14 7 91 ## 77 15 7 80 ## 78 16 7 81 ## 79 17 7 82 ## 80 18 7 84 ## 81 19 7 87 ## 82 20 7 85 ## 83 21 7 74 ## 84 22 7 81 ## 85 23 7 82 ## 86 24 7 86 ## 87 25 7 85 ## 88 26 7 82 ## 89 27 7 86 ## 90 28 7 88 ## 91 29 7 86 ## 92 30 7 83 ## 93 31 7 81 ## 94 1 8 81 ## 95 2 8 81 ## 96 3 8 82 ## 97 4 8 86 ## 98 5 8 85 ## 99 6 8 87 ## 100 7 8 89 ## 101 8 8 90 ## 102 9 8 90 ## 103 10 8 92 ## 104 11 8 86 ## 105 12 8 86 ## 106 13 8 82 ## 107 14 8 80 ## 108 15 8 79 ## 109 16 8 77 ## 110 17 8 79 ## 111 18 8 76 ## 112 19 8 78 ## 113 20 8 78 ## 114 21 8 77 ## 115 22 8 72

116 23 8 75 ## 117 24 8 79 ## 118 25 8 81 ## 119 26 8 86 ## 120 27 8 88 ## 121 28 8 97 ## 122 29 8 94 ## 123 30 8 96 ## 124 31 8 94 ## 125 1 9 91 ## 126 2 9 92 ## 127 3 9 93 ## 128 4 9 93 ## 129 5 9 87 ## 130 6 9 84 ## 131 7 9 80 ## 132 8 9 78 ## 133 9 9 75 ## 134 10 9 73 ## 135 11 9 81 ## 136 12 9 76 ## 137 13 9 77 ## 138 14 9 71 ## 139 15 9 71 ## 140 16 9 78 ## 141 17 9 67 ## 142 18 9 76 ## 143 19 9 68 ## 144 20 9 82 ## 145 21 9 64 ## 146 22 9 71 ## 147 23 9 81 ## 148 24 9 69 ## 149 25 9 63 ## 150 26 9 70 ## 151 27 9 77 ## 152 28 9 75 ## 153 29 9 76 ## 154 30 9 68 ## 155 31 9 NA

```
##################### # # # Exercise 8 # # # ##################### back2long_na <- back2long %>% replace_na(replace = list(Temp = "unknown")) ; back2long_na
```

Day Month Temp ## 1 1 5 67 ## 2 2 5 72 ## 3 3 5 74 ## 4 4 5 62 ## 5 5 5 56 ## 6 6 5 66 ## 7 7 5 65 ## 8 8 5 59 ## 9 9 5 61 ## 10 10 5 69 ## 11 11 5 74 ## 12 12 5 69 ## 13 13 5 66 ## 14 14 5 68 ## 15 15 5 58 ## 16 16 5 64 ## 17 17 5 66 ## 18 18 5 57 ## 19 19 5 68 ## 20 20 5 62 ## 21 21 5 59 ## 22 22 5 73 ## 23 23 5 61 ## 24 24 5 61 ## 25 25 5 57 ## 26 26 5 58 ## 27 27 5 57 ## 28 28 5 67 ## 29 29 5 81 ## 30 30 5 79 ## 31 31 5 76 ## 32 1 6 78 ## 33 2 6 74 ## 34 3 6 67 ## 35 4 6 84 ## 36 5 6 85 ## 37 6 6 79 ## 38 7 6 82 ## 39 8 6 87 ## 40 9 6 90 ## 41 10 6 87 ## 42 11 6 93 ## 43 12 6 92 ## 44 13 6 82 ## 45 14 6 80 ## 46 15 6 79 ## 47 16 6 77 ## 48 17 6 72 ## 49 18 6 65 ## 50 19 6 73 ## 51 20 6 76 ## 52 21 6 77 ## 53 22 6 76 ## 54 23 6 76 ## 55 24 6 76 ## 56 25 6 75 ## 57 26 6 78 ## 58 27 6 73 ## 59 28 6 80 ## 60 29 6 77 ## 61 30 6 83 ## 62 31 6 unknown ## 63 1 7 84 ## 64 2 7 85 ## 65 3 7 81 ## 66 4 7 84 ## 67 5 7 83 ## 68 6 7 83 ## 69 7 7 88 ## 70 8 7 92 ## 71 9 7 92 ## 72 10 7 89 ## 73 11 7 82 ## 74 12 7 73 ## 75 13 7 81 ## 76 14 7 91 ## 77 15 7 80 ## 78 16 7 81 ## 79 17 7 82 ## 80 18 7 84 ## 81 19 7 87 ## 82 20 7 85 ## 83 21 7 74 ## 84 22 7 81 ## 85 23 7 82 ## 86 24 7 86 ## 87 25 7 85 ## 88 26 7 82 ## 89 27 7 86 ## 90 28 7 88 ## 91 29 7 86 ## 92 30 7 83 ## 93 31 7 81 ## 94 1 8 81 ## 95 2 8 81 ## 96 3 8 82 ## 97 4 8 86 ## 98 5 8 85 ## 99 6 8 87 ## 100 7 8 89 ## 101 8 8 90 ## 102 9 8 90 ## 103 10 8 92 ## 104 11 8 86 ## 105 12 8 86 ## 106 13 8 82 ## 107 14 8 80 ## 108 15 8 79 ## 109 16 8 77 ## 110 17 8 79 ## 111 18 8 76 ## 112 19 8 78 ## 113 20 8 78 ## 114 21 8 77 ## 115 22 8 72

116 23 8 75 ## 117 24 8 79 ## 118 25 8 81 ## 119 26 8 86 ## 120 27 8 88 ## 121 28 8 97 ## 122 29 8 94 ## 123 30 8 96 ## 124 31 8 94 ## 125 1 9 91 ## 126 2 9 92 ## 127 3 9 93 ## 128 4 9 93 ## 129 5 9 87 ## 130 6 9 84 ## 131 7 9 80 ## 132 8 9 78 ## 133 9 9 75 ## 134 10 9 73 ## 135 11 9 81 ## 136 12 9 76 ## 137 13 9 77 ## 138 14 9 71 ## 139 15 9 71 ## 140 16 9 78 ## 141 17 9 67 ## 142 18 9 76 ## 143 19 9 68 ## 144 20 9 82 ## 145 21 9 64 ## 146 22 9 71 ## 147 23 9 81 ## 148 24 9 69 ## 149 25 9 63 ## 150 26 9 70 ## 151 27 9 77 ## 152 28 9 75 ## 153 29 9 76 ## 154 30 9 68 ## 155 31 9 unknown

```
##################### # # # Exercise 9 # # # ##################### back2long_na <- back2long_na %>% fill(year) ; back2long_na
```

Day Month Temp year ## 1 1 5 67 2015 ## 2 2 5 72 2015 ## 3 3 5 74 2015 ## 4 4 5 62 2015 ## 5 5 5 56 2015 ## 6 6 5 66 2015 ## 7 7 5 65 2015 ## 8 8 5 59 2015 ## 9 9 5 61 2015 ## 10 10 5 69 2015 ## 11 11 5 74 2015 ## 12 12 5 69 2015 ## 13 13 5 66 2015 ## 14 14 5 68 2015 ## 15 15 5 58 2015 ## 16 16 5 64 2015 ## 17 17 5 66 2015 ## 18 18 5 57 2015 ## 19 19 5 68 2015 ## 20 20 5 62 2015 ## 21 21 5 59 2015 ## 22 22 5 73 2015 ## 23 23 5 61 2015 ## 24 24 5 61 2015 ## 25 25 5 57 2015 ## 26 26 5 58 2015 ## 27 27 5 57 2015 ## 28 28 5 67 2015 ## 29 29 5 81 2015 ## 30 30 5 79 2015 ## 31 31 5 76 2015 ## 32 1 6 78 2015 ## 33 2 6 74 2015 ## 34 3 6 67 2015 ## 35 4 6 84 2015 ## 36 5 6 85 2015 ## 37 6 6 79 2015 ## 38 7 6 82 2015 ## 39 8 6 87 2015 ## 40 9 6 90 2015 ## 41 10 6 87 2015 ## 42 11 6 93 2015 ## 43 12 6 92 2015 ## 44 13 6 82 2015 ## 45 14 6 80 2015 ## 46 15 6 79 2015 ## 47 16 6 77 2015 ## 48 17 6 72 2015 ## 49 18 6 65 2015 ## 50 19 6 73 2015 ## 51 20 6 76 2016 ## 52 21 6 77 2016 ## 53 22 6 76 2016 ## 54 23 6 76 2016 ## 55 24 6 76 2016 ## 56 25 6 75 2016 ## 57 26 6 78 2016 ## 58 27 6 73 2016 ## 59 28 6 80 2016 ## 60 29 6 77 2016 ## 61 30 6 83 2016 ## 62 31 6 unknown 2016 ## 63 1 7 84 2016 ## 64 2 7 85 2016 ## 65 3 7 81 2016 ## 66 4 7 84 2016 ## 67 5 7 83 2016 ## 68 6 7 83 2016 ## 69 7 7 88 2016 ## 70 8 7 92 2016 ## 71 9 7 92 2016 ## 72 10 7 89 2016 ## 73 11 7 82 2016 ## 74 12 7 73 2016 ## 75 13 7 81 2016 ## 76 14 7 91 2016 ## 77 15 7 80 2016 ## 78 16 7 81 2016 ## 79 17 7 82 2016 ## 80 18 7 84 2016 ## 81 19 7 87 2016 ## 82 20 7 85 2016 ## 83 21 7 74 2016 ## 84 22 7 81 2016 ## 85 23 7 82 2016

```
## 86 24 7 86 2016 ## 87 25 7 85 2016 ## 88 26 7 82 2016 ## 89
27 7 86 2016 ## 90 28 7 88 2016 ## 91 29 7 86 2016 ## 92 30 7
83 2016 ## 93 31 7 81 2016 ## 94 1 8 81 2016 ## 95 2 8 81 2016
## 96 3 8 82 2016 ## 97 4 8 86 2016 ## 98 5 8 85 2016 ## 99 6
8 87 2016 ## 100 7 8 89 2016 ## 101 8 8 90 2016 ## 102 9 8 90
2016 ## 103 10 8 92 2017 ## 104 11 8 86 2017 ## 105 12 8 86
2017 ## 106 13 8 82 2017 ## 107 14 8 80 2017 ## 108 15 8 79
2017 ## 109 16 8 77 2017 ## 110 17 8 79 2017 ## 111 18 8 76
2017 ## 112 19 8 78 2017 ## 113 20 8 78 2017 ## 114 21 8 77
2017 ## 115 22 8 72 2017 ## 116 23 8 75 2017 ## 117 24 8 79
2017 ## 118 25 8 81 2017 ## 119 26 8 86 2017 ## 120 27 8 88
2017 ## 121 28 8 97 2017 ## 122 29 8 94 2017 ## 123 30 8 96
2017 ## 124 31 8 94 2017 ## 125 1 9 91 2017 ## 126 2 9 92 2017
## 127 3 9 93 2017 ## 128 4 9 93 2017 ## 129 5 9 87 2017 ##
130 6 9 84 2017 ## 131 7 9 80 2017 ## 132 8 9 78 2017 ## 133 9
9 75 2017 ## 134 10 9 73 2017 ## 135 11 9 81 2017 ## 136 12 9
76 2017 ## 137 13 9 77 2017 ## 138 14 9 71 2017 ## 139 15 9 71
2017 ## 140 16 9 78 2017 ## 141 17 9 67 2017 ## 142 18 9 76
2017 ## 143 19 9 68 2017 ## 144 20 9 82 2017 ## 145 21 9 64
2017 ## 146 22 9 71 2017 ## 147 23 9 81 2017 ## 148 24 9 69
2017 ## 149 25 9 63 2017 ## 150 26 9 70 2017 ## 151 27 9 77
2017 ## 152 28 9 75 2017 ## 153 29 9 76 2017 ## 154 30 9 68
2017 ## 155 31 9 unknown 2017

##################### # # # Exercise 10 # # #
#################### back2long_na %$% extract_numeric(Temp)

## [1] 67 72 74 62 56 66 65 59 61 69 74 69 66 68 58 64 66 57
68 62 59 73 61 ## [24] 61 57 58 57 67 81 79 76 78 74 67 84 85
79 82 87 90 87 93 92 82 80 79 ## [47] 77 72 65 73 76 77 76 76
76 75 78 73 80 77 83 NA 84 85 81 84 83 83 88 ## [70] 92 92 89
82 73 81 91 80 81 82 84 87 85 74 81 82 86 85 82 86 88 86 83 ##
[93] 81 81 81 82 86 85 87 89 90 90 92 86 86 82 80 79 77 79 76
78 78 77 72 ## [116] 75 79 81 86 88 97 94 96 94 91 92 93 93 87
84 80 78 75 73 81 76 77 71 ## [139] 71 78 67 76 68 82 64 71 81
69 63 70 77 75 76 68 NA
```

Data wrangling : I/O (Part-2)-Solutions

Below are the solutions to these exercises on data importing and exporting.

```
#####################  #  #  #  Exercise  1  #  #  #  ###################### url <- getURL("https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-1/data.csv") ###################### # # # Exercise 2 # # # ###################### csv_file <- read.csv(text = url) ###################### # # # Exercise 3 # # #  ######################  url <- getURL("https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-2/data.txt") txt_file <- read.table(text = url) ######################  #  #  #  Exercise  4  #  #  #  ###################### url <- getURL("https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-2/data.json") json_file <- fromJSON(url) ######################  #  #  #  Exercise  5  #  #  #  ###################### url <- getURL("https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-2/data.xml") xml_file <- ldply(xmlToList(url), data.frame)
```

```
#####################  #  #  #  Exercise  6  #  #  #  ###################### url <- read_html("http://www.worldatlas.com/articles/largest-cities-in-europe-by-population.html") ###################### # # # Exercise 7 # # # ###################### tbls <- html_nodes(url, "table") ###################### # # # Exercise 8 # # # ###################### tbls_read <- url %>% html_nodes("table") %>% html_table(fill = TRUE) ###################### # # # Exercise  9  #  #  #  ######################  url  <- "http://www.worldatlas.com/articles/largest-cities-in-europe-by-population.html"  tbls_xml  <-  readHTMLTable(url) ######################  #  #  #  Exercise  10  #  #  #  ###################### df_pop <- htmltab(doc = url, which = "//th[text() = 'Rank']/ancestor::table")
```

Data wrangling : I/O (Part-2)

Data wrangling is a task of great importance in data analysis. Data wrangling, is the process of importing, cleaning and transforming raw data into actionable information for analysis. It is a time-consuming process which is estimated to take about 60-80% of analyst's time. In this series we will go through this process. It will be a brief series with goal to craft the reader's skills on the data wrangling task. This is the first part of this series and it aims to cover the importing of data from the web. In many cases, downloading data in order to process them can be time consuming, therefore being able to import the data straight from the web is a 'nice-to-have' skill. Moreover, data isn't always not saved in structured files, but they are on the web in forms of text and tables, in this set of exercise we will go through the latter case. In case you want me to go through the former case as well, please let me know at the comment section.

Before proceeding, it might be helpful to look over the help pages for the getURL, fromJSON, ldply, xmlToList, read_html, html_nodes, html_table, readHTMLTable, htmltab.

Moreover please load the following libraries.

```
install.packages("RCurl")
library(RCurl)
install.packages("rjson")
library(rjson)
install.packages("XML")
library(XML)
```

```
install.packages("plyr")
library(plyr)
install.packages("rvest")
library(rvest)
install.packages("htmltab")
library(htmltab)
```

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Exercise 1

Retrieve the source of the web page "https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-1/data.csv" and assign it to the object "url"

Exercise 2

Read the csv file and assign it to the "csv_file" object.

Exercise 3

Do the same as exercise 1, but with the url: "https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-2/data.txt" and then assign it to the "txt_file" object.
Note: it is a txt file, so you should use the adequate function in order to import it.

Exercise 4

Do the same as exercise 1, but with the url: "https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-2/data.json" and then assign it to the "json_file" object.
Note: it is a json file, so you should use the adequate function in order to import it.

Learn more about Data Pre-Processing in the online course R Data Pre-Processing & Data Management – Shape your Data!. In this course you will learn how to:

- import data into R in several ways while also beeing able to identify a suitable import tool
- use SQL code within R
- And much more

Exercise 5

Do the same as exercise 1, but with the url: "https://raw.githubusercontent.com/VasTsak/r-exercises-dw/master/part-2/data.xml" and then assign it to the "xml_file" object.
Note: it is a xml file, so you should use the adequate function in order to import it.

Exercise 6

We will go through web scraping now. Read the html file "http://www.worldatlas.com/articles/largest-cities-in-europe-by-population.html" and assign it to the object "url".
hint: consider using read_html

Exercise 7

Select the "table" nodes from the html document you retrieved before.
hint: consider using html_nodes

Exercise 8

Convert the node you retrieved at exercise 7, to an actionable list for processing.
hint: consider using html_table

Exercise 9

Let’s go to a faster and more straight forward function, retrieve the html document like you did at exercise 6 and make it an actionable list using the function readHTMLTable.

Exercise 10

This may be a bit tricky, but give it a try. Retrieve the html document like you did at exercise 6 and make it an actionable data frame using the function htmltab.