List the sheets of an Excel file

Before you can start importing from Excel, you should find out which sheets are available in the workbook. You can use the excel_sheets() function for this.

You will find the Excel file latitude.xlsx in your working directory (type dir() to see it). This dataset gives information on the latitude of different countries for two different points in time (Source: Gapminder).

```
# Load the readxl package
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 3.2.4
```

```
# Find the names of both spreadsheets: sheets
sheets <- excel_sheets("latitude.xlsx")

# Print sheets
sheets
```

```
## [1] "1700" "1900"
```

```
# Find out the class of the sheets vector
class(sheets)
```

```
## [1] "character"
```

Import an Excel sheet

Now that you know the names of the sheets in the Excel file you want to import, it is time to import those sheets into R. You can do this with the read_excel() function. Have a look at this recipe:

- data <- read_excel("data.xlsx", sheet = "my_sheet")

This call simply imports the sheet with the name "my_sheet" from the "data.xlsx" file. You can also pass a number to the sheet argument; this will cause read_excel() to import the sheet with the given sheet number. sheet = 1 will import the first sheet, sheet = 2 will import the second sheet, and so on.

```
# Read the first sheet of latitude.xlsx: latitude_1
latitude_1 <- read_excel("latitude.xlsx", sheet=1)

# Read the second sheet of latitude.xlsx: latitude_2
latitude_2 <- read_excel("latitude.xlsx", sheet="1900")

# Put latitude_1 and latitude_2 in a list: lat_list

lat_list <- list(latitude_1, latitude_2)

# Display the structure of lat_list
str(lat_list)
```

```
## List of 2
##  $ :Classes 'tbl_df', 'tbl' and 'data.frame':    246 obs. of  2 variables:
##   ..$ country      : chr [1:246] "Afghanistan" "Akrotiri and Dhekelia" "Albania" "Algeria"
...
##   ..$ latitude_1700: num [1:246] 34.6 34.6 41.3 36.7 -14.3 ...
##  $ :Classes 'tbl_df', 'tbl' and 'data.frame':    246 obs. of  2 variables:
##   ..$ country      : chr [1:246] "Afghanistan" "Akrotiri and Dhekelia" "Albania" "Algeria"
...
##   ..$ latitude_1900: num [1:246] 34.6 34.6 41.3 36.7 -14.3 ...
```

Reading a workbook

In the previous exercise you generated a list of both Excel sheets that you imported. However, loading in every sheet manually and then merging them in a list can be quite tedious. There must be a way to elegantly automate this, right? lapply() comes to the rescue!

Have a look at the example code below:

- my_workbook <- lapply(excel_sheets("data.xlsx"), read_excel, path = "data.xlsx")

The read_excel() function is called multiple times on the "data.xlsx" file and each sheet is loaded in one after the other. The result is a list of data frames, each data frame representing one of the sheets in data.xlsx.

```
# Read all Excel sheets with lapply(): lat_list
lat_list <- lapply(excel_sheets("latitude.xlsx"), read_excel, path="latitude.xlsx")

# Display the structure of lat_list
str(lat_list)
```

```
## List of 2
##  $ :Classes 'tbl_df', 'tbl' and 'data.frame':    246 obs. of  2 variables:
##   ..$ country      : chr [1:246] "Afghanistan" "Akrotiri and Dhekelia" "Albania" "Algeria"
...
##   ..$ latitude_1700: num [1:246] 34.6 34.6 41.3 36.7 -14.3 ...
##  $ :Classes 'tbl_df', 'tbl' and 'data.frame':    246 obs. of  2 variables:
##   ..$ country      : chr [1:246] "Afghanistan" "Akrotiri and Dhekelia" "Albania" "Algeria"
...
##   ..$ latitude_1900: num [1:246] 34.6 34.6 41.3 36.7 -14.3 ...
```

The col_names argument

Apart from path and sheet, there are several other arguments you can specify in read_excel(). One of these arguments is called col_names.

By default it is TRUE, denoting that the first row in the Excel sheets denote the column names. If this is not the case, you can either set col_names to FALSE. In this case, R will choose column names for you. You can also choose to set col_names to a character vector with names for each column. It works exactly the same as in the readr package.

You'll be working with the latitude_nonames.xlsx file. It contains the same data as latitude.xlsx but has no column names in the first row of the excel sheets.

```
# Import the the first Excel sheet of latitude_nonames.xlsx (R gives names): latitude_3
# Have R set the column names of the resulting data frame itself
```

```
# Have R set the column names of the resulting data frame itself:
latitude_3 <- read_excel("latitude_nonames.xlsx", sheet=1, col_names=FALSE)

# Import the the second Excel sheet of latitude_nonames.xlsx (specify col_names): latitude_4
# This time, specify the column names to be "country" and "latitude"
latitude_4 <- read_excel("latitude_nonames.xlsx", sheet=1, col_names=c("country", "latitude"))

# Print the summary of latitude_3
summary(latitude_3)
```

```
##       X0                  X1
##  Length:247         Min.   :-51.750
##  Class :character   1st Qu.:  2.557
##  Mode  :character   Median : 16.755
##                     Mean   : 18.026
##                     3rd Qu.: 39.051
##                     Max.   : 78.000
##                     NA's   :5
```

```
# Print the summary of latitude_4
summary(latitude_4)
```

```
##    country             latitude
##  Length:247         Min.   :-51.750
##  Class :character   1st Qu.:  2.557
##  Mode  :character   Median : 16.755
##                     Mean   : 18.026
##                     3rd Qu.: 39.051

##                     Max.   : 78.000
##                     NA's   :5
```

The skip argument

Another argument that can be very useful when reading in Excel files that are less tidy, is skip. With skip, you can tell R to ignore a specified number of rows inside the Excel sheets you're trying to pull data from. Have a look at this example:

- read_excel("data.xlsx", skip = 15)

In this case, the first 15 rows in the first sheet of "data.xlsx" are ignored. If the first row of this sheet contained the column names, this information will also be ignored by readxl. Make sure to set col_names to FALSE or manually specify column names in this case!

```
# Import the second sheet of latitude.xlsx, skipping the first 21 rows: latitude_sel
latitude_sel <- read_excel("latitude.xlsx", skip=21, col_names=FALSE)

# Select the first observation from latitude_sel: first
first <- latitude_sel[1, ]


# Print first
first
```

```
##       X0     X1
```

```
## 1 Belgium 50.837
```

Import a local file

In this part of the chapter you'll learn how to import .xls files of the format .xls using the gdata package. Similar to the readxl package, you can import single Excel sheets from Excel sheets to start your analysis in R.

You'll be working with the urbanpop.xls dataset; it contains urban population metrics for practically all countries in the world throughout time (Source: Gapminder). It contains three sheets for three different time periods. In each sheet, the first row contains the column names.

```
# Load the gdata package
#install.packages("gdata")
#library(gdata)

# Import the second sheet of urbanpop.xls: urban_pop
#urban_pop <- read.xls("urbanpop.xls", sheet="1967-1974")

# Print the first 11 observations using head()
#head(urban_pop, n=11)
```

Import a workbook

The functions you've learned about in the video provide a good starting point to experiment with XLConnect package.Typically, the first step will be to load a workbook in your R session with loadWorkbook(); this function will build a "bridge" between your Excel file and your R session.

In this and the following exercises, you will be working with latitude.xlsx, containing information on the latitude of different countries for two different points in time (Source: Gapminder).

```
# Load the XLConnect package
library(XLConnect)
```

```
## Warning: package 'XLConnect' was built under R version 3.2.5
```

```
## Loading required package: XLConnectJars
```

```
## Warning: package 'XLConnectJars' was built under R version 3.2.3
```

```
## XLConnect 0.2-11 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI, Apache Commons
##     Codec),
##   Stephen Colebourne [ctb, cph] (Joda-Time Java library)
```

```
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com
```

```
# Build connection to latitude.xlsx: my_book
my_book <- loadWorkbook("latitude.xlsx")

# Print out the class of my_book
class(my_book)
```

```
## [1] "workbook"
## attr(,"package")
## [1] "XLConnect"
```

List and read Excel sheets

Just as readxl and gdata, you can use XLConnect to import data from Excel file into R. To list the sheets in an Excel file, use getSheets(). To actually import data from a sheet, you can use readWorksheet(). Both functions require an XLConnect workbook object as the first argument.

```
# Build connection to latitude.xlsx
library(XLConnect)
my_book <- loadWorkbook("latitude.xlsx")

# List the sheets in my_book
getSheets(my_book)
```

```
## [1] "1700" "1900"
```

```
# Import the second sheet in my_book
readWorksheet(my_book, sheet=2)
```

```
##                           country latitude_1900
## 1                     Afghanistan    34.5650000
## 2             Akrotiri and Dhekelia    34.6166667
## 3                         Albania    41.3120000
## 4                         Algeria    36.7200000
## 5                  American Samoa   -14.3070000
## 6                         Andorra    42.5460000
## 7                          Angola    -8.8430000
## 8                        Anguilla    18.2500000
## 9              Antigua and Barbuda    17.0720000
## 10                       Argentina   -36.6760000
## 11                         Armenia    40.2540000
## 12                           Aruba    12.5130000
## 13                       Australia   -32.2190000
## 14                         Austria    48.2310000
## 15                      Azerbaijan    40.3520000
## 16                         Bahamas    24.7000000
## 17                         Bahrain    26.0240000
```

```
## 17                          Bahrain   26.0240000
## 18                       Bangladesh   23.8800000
## 19                         Barbados   13.1790000
## 20                          Belarus   53.5470900
## 21                          Belgium   50.8370000
## 22                           Belize   17.8430000
## 23                            Benin    6.3640000
## 24                          Bermuda   32.2170000
## 25                           Bhutan   27.4790000
## 26                          Bolivia  -15.1900000
## 27          Bosnia and Herzegovina   44.1750100
## 28                         Botswana  -21.5360000
## 29                           Brazil  -19.5570000
## 30          British Virgin Islands   18.5000000
## 31                           Brunei    4.5010000
## 32                         Bulgaria   42.0730000
## 33                     Burkina Faso   12.0490000
## 34                          Burundi   -3.3650000
## 35                         Cambodia   12.0260000
## 36                         Cameroon   10.7300000
## 37                           Canada   43.7270000
## 38                       Cape Verde   15.0910000
## 39                   Cayman Islands   19.3190000
## 40             Central African Rep.    4.3310000
## 41                             Chad   10.3770000
## 42                   Channel Islands   49.2170000
## 43                             Chile  -33.5540000
## 44                             China   29.5610000
## 45                 Christmas Island  -10.5000000
## 46                     Cocos Island  -12.5000000
## 47                          Colombia    4.7880000
## 48                          Comoros  -11.6710000
## 49                Congo, Dem. Rep.   -2.9202760
## 50                     Congo, Rep.   -3.6840000
## 51                     Cook Islands  -21.2333333
## 52                       Costa Rica    9.9410000

## 53                    Cote d'Ivoire    5.4960000
## 54                          Croatia   45.1100800
## 55                             Cuba   23.0840000
## 56                           Cyprus   35.0810000
## 57                       Czech Rep.   49.7792900
## 58                          Denmark   55.7180000
## 59                         Djibouti   11.5050000
## 60                         Dominica   15.4330000
## 61                   Dominican Rep.   18.5610000
## 62                     East Germany           NA
## 63                          Ecuador   -2.0620000
## 64                            Egypt   29.9960000
## 65                      El Salvador   13.7750000
## 66                  Equatorial Guinea    2.3260000
## 67                          Eritrea   15.3120000
## 68                          Estonia   58.6850000
## 69                         Ethiopia    9.0070000
## 70                     Faeroe Islands   62.0000000
## 71       Falkland Islands (Malvinas)  -51.7500000
## 72                             Fiji  -17.8270000
## 73                          Finland   60.2120000
## 74                           France   48.8570000
```

```
## 75                    French Guiana     3.9880000
## 76                  French Polynesia   -17.6660000
## 77                             Gabon     0.3720000
## 78                            Gambia    13.2570000
## 79                           Georgia    42.0340000
## 80                           Germany    48.1610000
## 81                             Ghana     6.6940000
## 82                         Gibraltar    36.1333333
## 83                            Greece    38.0580000
## 84                         Greenland    64.2170000
## 85                           Grenada    12.1150000
## 86                        Guadeloupe    16.1630000
## 87                              Guam    13.4400000
## 88                         Guatemala    14.6220000
## 89                          Guernsey    49.4666667
## 90                            Guinea    11.6710000
## 91                     Guinea-Bissau    12.2620000
## 92                            Guyana     5.7610000
## 93                             Haiti    18.9320000
## 94                          Holy See    41.9000000
## 95                          Honduras    14.1940000
## 96                 Hong Kong, China    22.7040000
## 97                           Hungary    47.4190000
## 98                           Iceland    63.8920000
## 99                             India    25.2740000
## 100                        Indonesia    -6.5620000
## 101                             Iran    35.3800000
## 102                             Iraq    33.3140000
## 103                          Ireland    54.6110000
## 104                        Isle of Man    54.2250000
## 105                           Israel    32.0840000
## 106                            Italy    45.4150000
## 107                          Jamaica    18.0550000
## 108                            Japan    35.7120000
## 109                           Jersey    49.2500000
## 110                           Jordan    31.6020000
## 111                       Kazakhstan    44.3080000
## 112                            Kenya    -0.5130000
## 113                         Kiribati     1.8470000
## 114                 Korea, Dem. Rep.    39.5270000
## 115                      Korea, Rep.    37.5530000
## 116                           Kosovo    42.5833333
## 117                           Kuwait    29.3260000
## 118                       Kyrgyzstan    41.4894000
## 119                             Laos    16.5470000
## 120                           Latvia    56.8580000
## 121                          Lebanon    34.1110000
## 122                          Lesotho   -29.5950000
## 123                          Liberia     6.3850000
## 124                            Libya    32.6070000
## 125                    Liechtenstein    47.1410000
## 126                        Lithuania    55.3180000
## 127                       Luxembourg    49.7800000
## 128                     Macao, China    22.5060000
## 129                   Macedonia, FYR    41.5738100
## 130                       Madagascar   -18.9580000
## 131                           Malawi   -15.8110000
## 132                         Malaysia     3.2690000
```

```
## 133                         Maldives     3.2500000
## 134                             Mali    12.5080000
## 135                            Malta    35.8870000
## 136                  Marshall Islands     9.0000000
## 137                         Martinique    14.6540000
## 138                         Mauritania    17.9250000
## 139                          Mauritius   -20.2320000
## 140                            Mayotte   -12.8333333
## 141                             Mexico    16.7590000
## 142             Micronesia, Fed. Sts.     7.3570000
## 143                            Moldova    47.1660000
## 144                             Monaco    43.7460000
## 145                           Mongolia    47.4930000
## 146                         Montenegro    42.7889900
## 147                         Montserrat    16.7500000
## 148                            Morocco    33.5930000
## 149                         Mozambique   -18.4990000
## 150                            Myanmar    17.6790000
## 151                            Namibia   -17.9790000
## 152                              Nauru    -0.5333333
## 153                              Nepal    27.7120000
## 154                        Netherlands    51.8740000
## 155              Netherlands Antilles    12.1920000
## 156                      New Caledonia   -21.3290000
## 157                        New Zealand   -36.8920000
## 158                          Nicaragua    12.2110000
## 159                              Niger    13.8760000
## 160                            Nigeria     6.5430000
## 161                               Niue   -19.0333333
## 162                     Norfolk Island   -29.0333333
## 163          Northern Mariana Islands    15.1780000
## 164                             Norway    59.9770000
## 165                               Oman    20.4450000
## 166                           Pakistan    31.1730000
## 167                              Palau     7.5000000
## 168                             Panama     9.2060000
## 169                  Papua New Guinea    -6.6000000
## 170                           Paraguay   -25.5830000
## 171                               Peru   -11.7940000
## 172                        Philippines    13.9220000
## 173                           Pitcairn   -25.0666667
## 174                             Poland    50.2440000
## 175                           Portugal    38.8160000
## 176                        Puerto Rico    18.2250000
## 177                              Qatar    25.3090000
## 178                            Reunion   -20.9540000
## 179                            Romania    44.5260000
## 180                             Russia    55.6750000
## 181                             Rwanda    -2.0320000
## 182                  Saint Barthélemy    17.9000000
## 183                      Saint Helena   -15.9500000
## 184              Saint Kitts and Nevis    17.3270000
## 185                        Saint Lucia    13.8980000
## 186                       Saint Martin    18.0833333
## 187 Saint Vincent and the Grenadines    13.2540000
## 188          Saint-Pierre-et-Miquelon    46.8333333
## 189                              Samoa   -13.6330000
## 190                         San Marino    43.7666667
```

```
## 191            Sao Tome and Principe     1.0000000
## 192                     Saudi Arabia    23.0690000
## 193                          Senegal    14.7720000
## 194                           Serbia    44.0467300
## 195                       Seychelles    -4.6640000
## 196                     Sierra Leone     8.7010000
## 197                        Singapore     1.3550000
## 198                   Slovak Republic    48.7853800
## 199                          Slovenia    46.1279000
## 200                  Solomon Islands    -9.6250000
## 201                          Somalia    10.6330000
## 202                     South Africa   -29.1300000
## 203                            Spain    37.3980000
## 204                        Sri Lanka     6.8680000
## 205                            Sudan    14.0430000
## 206                         Suriname     5.6050000
## 207                         Svalbard    78.0000000
## 208                        Swaziland   -26.5450000
## 209                           Sweden    59.2780000
## 210                      Switzerland    47.4080000
## 211                            Syria    33.4580000
## 212                           Taiwan    23.6448500
## 213                       Tajikistan    37.8060000
## 214                         Tanzania    -2.1540000
## 215                         Thailand    13.7700000
## 216                      Timor-Leste    -8.8333333
## 217                             Togo     6.1940000
## 218                          Tokelau    -9.0000000
## 219                            Tonga   -21.1730000
## 220              Trinidad and Tobago    10.4180000
## 221                          Tunisia    36.8160000
## 222                           Turkey    41.2020000
## 223                     Turkmenistan    39.1293800
## 224          Turks and Caicos Islands    21.7500000
## 225                           Tuvalu    -8.0000000
## 226                           Uganda     0.2280000
## 227                          Ukraine    50.2810000
## 228              United Arab Emirates    23.3900000
## 229                    United Kingdom    51.5100000
## 230                     United States    34.3600000
## 231                           Uruguay   -34.8220000
## 232                              USSR           NA
## 233                        Uzbekistan    41.2720000
## 234                  Wallis et Futuna   -13.3000000
## 235                           Vanuatu   -15.2330000
## 236                         Venezuela     9.8430000
## 237                West Bank and Gaza    31.4166667
## 238                      West Germany           NA
## 239                    Western Sahara    24.6191300
## 240                           Vietnam    10.7980000
## 241              Virgin Islands (U.S.)    17.7360000
## 242                        Yemen, Rep.    15.2280000
## 243                         Yugoslavia           NA
## 244                             Zambia   -12.9420000
## 245                           Zimbabwe   -17.8760000
## 246                              Åland    60.0000000
```

Add and populate worksheets. Where readxl and gdata were only able to import Excel data, XLConnect's approach of providing an actual interface to an Excel file makes it able to edit your Excel files from inside R. In this exercise, you'll create a new sheet, populate the sheet with data, and save the results in a new Excel file..

```
# Build connection to latitude.xlsx
library(XLConnect)
my_book <- loadWorkbook("latitude.xlsx")

# Create data frame: summ
dims1 <- dim(readWorksheet(my_book, 1))
dims2 <- dim(readWorksheet(my_book, 2))
summ <- data.frame(sheets = getSheets(my_book),
                   nrows = c(dims1[1], dims2[1]),
                   ncols = c(dims1[2], dims2[2]))

# Add a worksheet to my_book, named "data_summary"
createSheet(my_book, name="data_summary")

# Populate "data_summary" with summ
writeWorksheet(my_book, summ, sheet="data_summary")

# Save workbook as latitude_with_summ.xlsx
saveWorkbook(my_book, "latitude_with_summ.xlsx")
```