MANAGING PACKAGES AND READING IN FILES

Statistical Computing in R

Lecturer: Ivan Innocent Sekibenga

1. INSTALLATION AND LOADING OF PACKAGES.

Packages in R are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the library. R comes with a standard set of packages, but you can also install additional packages from CRAN (the Comprehensive R Archive Network) or other repositories.

Note: You need an internet connection to install packages from CRAN or other repositories.

Commonly used packages and their use in R

Some common packages include:`tidyverse` (data manipulation and visualization)which contains `dplyr` (data manipulation), `ggplot2` (data visualization), `readr` (data import), `tidyr` (data tidying or cleaning),`tibble`(dataframes plus fast and accurate data upload) and `forcats`(categorical variables). For more information on `tidyverse`, type the function help(tidyverse) into a code chunk.

Others include; `readxl`(read excel files),`lubridate` (date and time manipulation), `stringr` (string or text manipulation), `caret` (machine learning), `randomForest` (random forest algorithm), `shiny` (interactive web applications), `rmarkdown` (dynamic documents), `knitr` (report generation),`tidymodels`(machine learning).

Package installation

Packages can be installed from CRAN using the `install.packages()` function in a chunk or by using the RStudio interface.To use the RStudio interface, go to the "Packages" tab in the bottom right panel of RStudio, click on "Install", and then type the name of the package you want to install in the dialog box that appears. You can also select multiple packages to install at once by separating their names with commas.

1. Code packages

To install a package, you can use the `install.packages()` function. For example, to install the 'readxl" package, you would run the following command in your chunk:

```
install.packages("readxl")
```

You can also install multiple packages at once by passing a vector of package names to the `install.packages()` function in your code chunk:

```
install.packages(c("randomForest", "caret"))
```

2. Data packages

We can also install packages that hold data. We will be using data from Nate Silver's statistical analysis website—fivethirtyeight. The `fivethirtyeight` package includes 128 callable datasets. For a full list of datasets, follow this link: https://cran.r-project.org/web/packages/fivethirtyeight/vignettes/fivethirtyeight .html. To install this package, use the code below. Remember to wrap the package in quotes when you are installing it.

```
install.packages("fivethirtyeight")
```

R base also comes with its own inbuilt datasets. To see all inbuilt R datasets that are available to you as callable objects, run the function

`data()` in your code chunk:

To find more information about a specific dataset, you can use the `?` operator followed by the dataset name. For example, to get more information about the `mtcars` dataset, you would run:

```
?mtcars  # displays information about the mtcars dataset
```

```
## starting httpd help server ... done
```

The information is displayed in the Help pane of RStudio. It contains a description of the dataset, the format of the data, and the source of the data and examples of how to use the dataset.

2. LOADING INSTALLED PACKAGES (LIBRARIES)

After installing a package, you need to load it into your R session using the `library()` function. You only need to install a package once, but you need to load it into your R session every time you start a new R session.

You do not need to use quotes when calling packages you have already installed. For example, to load the `readxl` package, you would run the following command in your code chunk:

`library(readxl)`

The same applies to loading the `fivethirtyeight` package.

`library(fivethirtyeight)`

Examples of installing and loading package from fivethirtyeight and using inbuilt R datasets

1. Example on fivethirtyeight package

`install.packages("fivethirtyeight")` :installs the `fivethirtyeight` package.

```
library(fivethirtyeight) # loads the fivethirtyeight package
```

The `drinks` dataset is part of the `fivethirtyeight` package.

```
dr <- drinks   # loads the drinks dataset and assigns it to the object dr

head(dr) # displays first six rows in a tibble
```

```
##             country beer_servings spirit_servings wine_servings
## 1       Afghanistan             0               0             0
## 2           Albania            89             132            54
## 3           Algeria            25               0            14
## 4           Andorra           245             138           312
## 5            Angola           217              57            45
## 6 Antigua & Barbuda           102             128            45
##   total_litres_of_pure_alcohol
## 1                          0.0
## 2                          4.9
## 3                          0.7
## 4                         12.4
## 5                          5.9
## 6                          4.9
```

You can also use the `View()` function to view the dataset in a spreadsheet-like format.

`View(dr)` :viewing the dataset in a spreadsheet-like format:

2. Example on inbuilt R datasets

`LakeHuron` is a time series inbuilt R dataset of water Level of Lake Huron from 1875-1972

```r
lh <- LakeHuron # Loads the dataset and assigns it to object lh
lh              # displays the timeseries dataset
```

```
## Time Series:
## Start = 1875
## End = 1972
## Frequency = 1
##  [1] 580.38 581.86 580.97 580.80 579.79 580.39 580.42 580.82 581.40 581.32
## [11] 581.44 581.68 581.17 580.53 580.01 579.91 579.14 579.16 579.55 579.67
## [21] 578.44 578.24 579.10 579.09 579.35 578.82 579.32 579.01 579.00 579.80
## [31] 579.83 579.72 579.89 580.01 579.37 578.69 578.19 578.67 579.55 578.92
## [41] 578.09 579.37 580.13 580.14 579.51 579.24 578.66 578.86 578.05 577.79
## [51] 576.75 576.75 577.82 578.64 580.58 579.48 577.38 576.90 576.94 576.24
## [61] 576.84 576.85 576.90 577.79 578.18 577.51 577.23 578.42 579.61 579.05
## [71] 579.26 579.22 579.38 579.10 577.95 578.12 579.75 580.85 580.41 579.96
## [81] 579.61 578.76 578.18 577.21 577.13 579.10 578.25 577.91 576.89 575.96
## [91] 576.80 577.68 578.38 578.52 579.74 579.31 579.89 579.96
```

Note: The `View()` function does not work on time series datasets.

3. OBTAINING AND LOADING EXTERNAL DATASETS

1. Downloading external data from Kaggle. Visit https://www.kaggle.com/datasets to download datasets of your choice. You will need to create an account on Kaggle to download datasets.
2. After downloading the dataset, you can use the `read_csv()` function to read in the data into R. For example, if you downloaded a dataset called `data.csv`, you would use the following command in your code chunk:

`data <- read_csv("path/to/data.csv")` or

simply `data <- read_csv("data.csv")`

if the file is in your working directory. You can check your working directory using the `getwd()` function and set it using the `setwd()` function.

Loading different file formats

These examples assume that the files are in your working directory. If they are not, you will need to provide the full path to the file.

Importing sas files using the 'haven' package.

sas files are files with the extension `.sas7bdat`and are commonly used in statistical analysis. They can be imported into R using the `haven` package.They are created using the SAS software.

SAS stands for Statistical Analysis System and is a software suite used for advanced analytics, business intelligence, data management, and predictive analytics.

```r
library(haven) # load haven package
```

```r
sas_data <- read_sas("Public_library.sas7bdat") # read in sas data file

head(sas_data) # display first six rows of the sas dataset)
```

```
## # A tibble: 6 x 192
##   STABR FSCSKEY LIBID    LIBNAME ADDRESS CITY  ZIP   ZIP4  ADDRES_M CITY_M ZIP_M
##   <chr> <chr>   <chr>    <chr>   <chr>   <chr> <chr> <chr> <chr>    <chr>  <chr>
## 1 AK    AK0001  AK0001-~ ANCHOR~ 34020 ~ ANCH~ 99556 9150  P.O. BO~ ANCHO~ 99556
## 2 AK    AK0002  AK0002-~ ANCHOR~ 3600 D~ ANCH~ 99503 6055  3600 DE~ ANCHO~ 99503
## 3 AK    AK0003  AK0003-~ ANDERS~ 101 FI~ ANDE~ 99744 M     P.O. BO~ ANDER~ 99744
## 4 AK    AK0006  AK0006-~ KUSKOK~ 420 CH~ BETH~ 99559 M     P.O. BO~ BETHEL 99559
## 5 AK    AK0007  AK0007-~ BIG LA~ 3140 S~ WASI~ 99623 9663  P.O. BO~ BIG L~ 99652
## 6 AK    AK0008  AK0008-~ CANTWE~ 1 SCHO~ CANT~ 99729 M     P.O. BO~ CANTW~ 99729
## # i 181 more variables: ZIP4_M <chr>, CNTY <chr>, PHONE <chr>, C_RELATN <chr>,
## #   C_LEGBAS <chr>, C_ADMIN <chr>, C_FSCS <chr>, GEOCODE <chr>, LSABOUND <chr>,
## #   STARTDAT <chr>, ENDDATE <chr>, POPU_LSA <dbl>, F_POPLSA <chr>,
## #   POPU_UND <dbl>, CENTLIB <dbl>, F_CENLIB <chr>, BRANLIB <dbl>,
## #   F_BRLIB <chr>, BKMOB <dbl>, F_BKMOB <chr>, MASTER <dbl>, F_MASTER <chr>,
## #   LIBRARIA <dbl>, F_LIBRAR <chr>, OTHPAID <dbl>, F_OTHSTF <chr>,
## #   TOTSTAFF <dbl>, F_TOTSTF <chr>, LOCGVT <dbl>, F_LOCGVT <chr>, ...
```

Importing stata files using the ‘haven‘ package.

Stata files are files with the extension `.dta` and are commonly used in statistical analysis. They can be imported into R using the `haven` package. Stata is a software package used for data management, statistical analysis, and graphics.

```r
library(haven) # load haven package

stata_data <- read_dta("household.dta") # read in stata data file

head(stata_data) # display first six rows of the stata dataset
```

```
## # A tibble: 6 x 326
##   hhid  hvidx hv000 hv001 hv002 hv003 hv004  hv005 hv006 hv007 hv008 hv009 hv010
##   <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 "   ~     1 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 2 "   ~     2 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 3 "   ~     3 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 4 "   ~     4 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 5 "   ~     5 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 6 "   ~     6 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## # i 313 more variables: hv011 <dbl>, hv012 <dbl>, hv013 <dbl>, hv014 <dbl>,
## #   hv015 <dbl+lbl>, hv016 <dbl>, hv017 <dbl>, hv018 <dbl>, hv019 <dbl>,
## #   hv020 <dbl+lbl>, hv021 <dbl>, hv022 <dbl>, hv023 <dbl+lbl>,
## #   hv024 <dbl+lbl>, hv025 <dbl+lbl>, hv026 <dbl+lbl>, hv027 <dbl+lbl>,
## #   hv028 <dbl>, hv030 <dbl>, hv031 <dbl>, hv032 <dbl>, hv035 <dbl>,
## #   hv040 <dbl>, hv041 <dbl>, hv042 <dbl+lbl>, hv044 <dbl+lbl>,
## #   hv201 <dbl+lbl>, hv202 <dbl+lbl>, hv204 <dbl+lbl>, hv205 <dbl+lbl>, ...
```

Importing csv files using the ‘readr‘ package.

csv files are files with the extension `.csv` and are commonly used for storing tabular data. They can be imported into R using the `readr` package. csv stands for Comma Separated Values.

```r
library(readr) # load readr package

data <- read_csv("Financial.csv") # read in csv data file
```

```
## Rows: 351 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (16): Account, Businees Unit, Currency, Scenario, Jan, Feb, Mar, Apr, Ma...
## dbl  (1): Year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(data) # display first six rows of the csv dataset
```

```
## # A tibble: 6 x 17
##   Account  `Businees Unit` Currency  Year Scenario Jan   Feb   Mar   Apr   May  
##   <chr>    <chr>           <chr>    <dbl> <chr>    <chr> <chr> <chr> <chr> <chr>
## 1 Sales    Software        USD       2012 Actuals  $90,~ $82,~ $72,~ $52,~ $77,~
## 2 Cost of~ Software        USD       2012 Actuals  ($41~ ($40~ ($30~ ($21~ ($37~
## 3 Commiss~ Software        USD       2012 Actuals  ($4,~ ($3,~ ($3,~ ($2,~ ($3,~
## 4 Payroll~ Software        USD       2012 Actuals  ($9,~ ($9,~ ($8,~ ($6,~ ($8,~
## 5 Travel ~ Software        USD       2012 Actuals  ($95~ ($83~ ($87~ ($62~ ($91~
## 6 R&D Exp~ Software        USD       2012 Actuals  ($4,~ ($3,~ ($3,~ ($2,~ ($3,~
## # i 7 more variables: Jun <chr>, Jul <chr>, Aug <chr>, Sep <chr>, Oct <chr>,
## #   Nov <chr>, Dec <chr>
```

Investigate the csv dataset

```r
class(data) # class of the csv dataset
```

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

```r
names(data) # variable names of the csv dataset
```

```
##  [1] "Account"       "Businees Unit" "Currency"      "Year"         
##  [5] "Scenario"      "Jan"           "Feb"           "Mar"          
##  [9] "Apr"           "May"           "Jun"           "Jul"          
## [13] "Aug"           "Sep"           "Oct"           "Nov"          
## [17] "Dec"
```

```r
dim(data) # dimensions of the csv dataset
```

```
## [1] 351  17
```

Importing excel files using the ‘readxl‘ package.

Excel files are files with the extension `.xlsx` or `.xls` and are commonly used for storing tabular data. They can be imported into R using the `readxl` package. Excel is a spreadsheet software developed by Microsoft.

```r
library(readxl) # load readxl package

excel_data <- read_excel("Employees.xlsx") # read in excel data file

head(excel_data) # display first six rows of the excel dataset
```

```
## # A tibble: 6 x 14
##   EEID   `Full Name`     `Job Title` Department `Business Unit` Gender Ethnicity
##   <chr>  <chr>           <chr>       <chr>      <chr>           <chr>  <chr>
## 1 E02387 Emily Davis     Sr. Manger  IT         Research & Dev~ Female Black
## 2 E04105 Theodore Dinh   Technical ~ IT         Manufacturing   Male   Asian
## 3 E02572 Luna Sanders    Director    Finance    Speciality Pro~ Female Caucasian
## 4 E02832 Penelope Jordan Computer S~ IT         Manufacturing   Female Caucasian
## 5 E01639 Austin Vo       Sr. Analyst Finance    Manufacturing   Male   Asian
## 6 E00644 Joshua Gupta    Account Re~ Sales      Corporate       Male   Asian
## # i 7 more variables: Age <dbl>, `Hire Date` <dttm>, `Annual Salary` <dbl>,
## #   `Bonus %` <dbl>, Country <chr>, City <chr>, `Exit Date` <dttm>
```

Importing sav files using the ‘haven‘ package.

sav files are files with the extension `.sav` and are commonly used in statistical analysis. They can be imported into R using the `haven` package. SPSS stands for Statistical Package for the Social Sciences and is a software package used for statistical analysis.

```r
library(haven) # load haven package

sav_data <- read_sav("household.sav") # read in sav data file

head(sav_data) # display first six rows of the sav dataset
```

```
## # A tibble: 6 x 326
##   HHID  HVIDX HV000 HV001 HV002 HV003 HV004  HV005 HV006 HV007 HV008 HV009 HV010
##   <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 "   ~     1 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 2 "   ~     2 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 3 "   ~     3 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 4 "   ~     4 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 5 "   ~     5 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## 6 "   ~     6 ZZ6       1     1 2         1 1.05e6     6  2015  1386     6     1
## # i 313 more variables: HV011 <dbl>, HV012 <dbl>, HV013 <dbl>, HV014 <dbl>,
## #   HV015 <dbl+lbl>, HV016 <dbl>, HV017 <dbl>, HV018 <dbl>, HV019 <dbl>,
## #   HV020 <dbl+lbl>, HV021 <dbl>, HV022 <dbl>, HV023 <dbl+lbl>,
## #   HV024 <dbl+lbl>, HV025 <dbl+lbl>, HV026 <dbl+lbl>, HV027 <dbl+lbl>,
## #   HV028 <dbl>, HV030 <dbl>, HV031 <dbl>, HV032 <dbl>, HV035 <dbl>,
## #   HV040 <dbl>, HV041 <dbl>, HV042 <dbl+lbl>, HV044 <dbl+lbl>,
## #   HV201 <dbl+lbl>, HV202 <dbl+lbl>, HV204 <dbl+lbl>, HV205 <dbl+lbl>, ...
```

Practical Exercise.

Import text files using the readr package. The text file is called `city_temperature.txt`. After importing, investigate the dataset using the functions `class()`, `names()`, and `dim()`. The code to use is provided below.

```r
library(readr) # load readr package
text_data <- read_tsv("city_temperature.txt") # read in text data file
```