# Data Analysis with R

Saturday, January 28, 2023 22:23

# Intro to R

Tuesday, December 6, 2022 10:25

**R**: a programming language frequently used for statistical analysis, visualization, and other data analysis

- accessible
- data-centric
- open-source

**Integrated Development Environment (IDE)**: a software application that brings together all the tools you may want to use in a single place

## Programming fundamentals

**variable (R)** a representation of a value in R that can be stored for use later during programming (='objects'). a variable name should start with a letter and can also contain numbers and underscores

**comments start with '#'**

```
first variable <— "This is my variable"
```

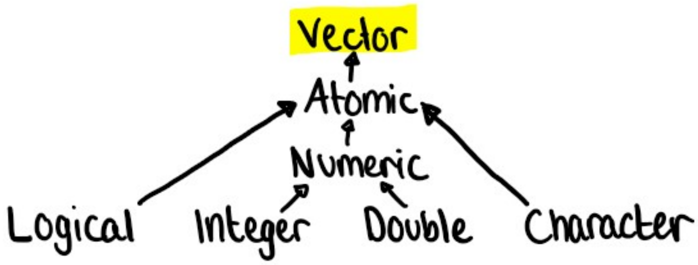
**Vector (R)** a group of data elements of the same type stored in a sequence in R `vec_1 <- c(13, 48.5, 71)`

**Pipe (R)** a tool in R for expressing a sequence of multiple operations. represented with "% >%"

```
ToothGrowth %>%
filter (dose ==0.5) %>%
arrange (len)
```

**Data Structures**: a format for organizing and storing data

- Vectors - atomic vectors & lists
- Data frames
- Matrices
- Arrays



create vectors using the c() function ("combine" function)

```
c(x,y,z)
```

```
typeof (vector)   -> "character";"integer" etc
length (vector)
is.logical (vector), is.integer(vector) etc
names(vector) <- c("a","b","c")
```

**Lists** their elements can be of any type:
dates, data frames, vectors, matrices, etc

```
list ("a", 1L, 1.5, TRUE)
```

can determine the structure of a list using the `str()` function

can name elements in a list

```
list('Chicago' = 1,'New York' = 2)
$ Chicago
[1]  1
```

## Dates & Times

using tidyverse & lubridate packages

```
install.packages ("tidyverse")
library(tidyverse)
library (lubridate)
```

### Types

- date ("2016-08-16")
- time within a day ("20:11:59 UTC")
- date-time ("2018-03-31  18:15:48 UTC")

**UTC**: Universal time coordinated

`today()` returns current date

`now()` returns current datetime

`today()` returns current date

`now()` returns current datetime

convert string tc date

`ymd("2021-01-20")` , also takes unquoted numbers

string to date-time

`mdy_hm("01/20/2021 08:01")`

switch between date-time and date

`as_date(now())`

## Other common data structures

**Data frame** a collection of columns

- columns should be named
- dataframes can include many different types of data
- elements in the same column should be of the same type

```
data.frame(x=c(1,2,3), y=c(1.5, 5.5, 7.5))

   x  y
1  1  1.5
2  2  5.5
3  3  7.5
```

**Files**

```
dir.create("destination_folder")
```

`file.create` , specify .txt, .csv etc

```
file.create("new_text_file.txt")
file.copy()
unlink("some_file.csv")
```

**Matrices** : a two-dimensional collection of data elements, contains both rows & columns, can only contain a single data type

```
matrix(c(3:8), nrow=2)
     [,1] [,2] [,3]
```

|  |  |  |  |
|---|---|---|---|
| [1,] | 3 | 5 | 7 |
| [2,] | 4 | 6 | 8 |

# Coding in R

Monday, December 26, 2022 15:51

**Logical Operators** return a logical data type such as TRUE or FALSE

- AND (&, &&)
- OR (|, ||)
- NOT (!)

**Conditional Statements** : a declaration that if a certain condition holds, then a certain event must take place

- if()
- else()
- elseif()

```
if (condition) {
  expr1
 } else if (condition2) {
 expr2
 } else {
 expr3
 }
```

## Tidyverse tour

packages

- ggplot2
- tidyr
- readr
- dplyr
- tibble
- purrr
- stringr
- forcats

**ggplot2** create a variety of data viz by applying different visual properties to the data variables in R

**tidyr** used for data cleaning to make tidy data

**tidyr** used for data cleaning to make tidy data

**readr** import data, `read_csv()`

**dplyr** consistent set of functions that help complete some common data manipulation tasks

**Factors (R)** : store categorical data in R where the data values are limited and usually based on a finite group like country or year

## Working with pipes

**Nested function** · a function that is completely contained within another function

```
arrange(filter(ToothGrowth, dose == 0.5), len)
```

Pipe:

```
filtered_toothgrowth <- ToothGrowth %>%
   filter(dose==0.5) %>%
   arrange(len)
```

When using pipes·

- add pipe operator at the end of each line of the piped operation except the last one
- check code after you've programmed your pipe
- revisit piped operations to check for parts of your code to fix

# Data in R

Tuesday, December 27, 2022 12:23

## R data frames

- columns should be named
- data stored can be many different types
- each column should contain the same number of data items

**Tibbles**

- never change the data types of your inputs
- never change the names of your variables
- never create row names

**Tidy data (R)** a way of standardizing the organization of data within R

- variables are organized into columns
- observations are organized into rows
- each value must have its own cell

## Working with dataframes

```
library(ggplot2)
data("diamonds")
View(diamonds)
head(diamonds)
```

`mutate()` → part of the tidyverse

**readr package**: part of the tidyverse, great for reading rectangular data, much faster than base R, produce tibbles

`read_csv(), read_fwf(), read_table(), read_log() etc`

**readxl** : for transferring data from Excel to R, need to load readxl separately: library(readxl)

## Cleaning data

packages : here, skimr, janitor, dplyr

skim_without_charts(), select(), rename(), rename_with()

## Organize data

packages: tidyverse

arrange(), group_by(), drop_na(), summarize(), filter()

## Transform data

separate(), unite(), mutate()

## Biased data

package : SimDesign

bias( )

→ close to 0 means no/little bias

# Visualization in R

Tuesday, December 27, 2022 19:42

**Popular packages:**

- ggplot2
- Plotly
- Lattice
- Highcharter
- gganimate
- RGL
- Dygraphs
- Leaflet
- Patchwork
- ggridges

gg stands for 'grammar of graphics'

**benefits of ggplot2:**

- create different types of plots
- customize the look and feel of plots
- create high quality visuals

**core topics:**

- **aesthetic**: a visual property of an object in your plot
- **geom**: the geometric object used to represent your data
- **facets**: let you display smaller groups, or subsets, of your data
- **labels and annotations**: let you customize your plot

- **aesthetics for points:**
  - x
  - Y
  - Color
  - Shape
  - size
  - Alpha
- **geom functions:**
  - geom_point
  - geom_bar
  - geom_line
  - geom_smooth method= "loess" or "gam"
    - "loess": <1000 data pts
    - "gam": >1000 data pts
  - geom_jitter
- **facet functions:**
  - facet_wrap()

- facet_wrap()
- facet_grid()

- misc
  - to add a title to a chart, use a label function, title = Title
  - to highight underperforming values, use an aesthetics function:
    col = ifelse (x<2, 'blue', 'yellow')
  - to label axes, use aesthetic function:
    aes (x= average price (USD), y= Product)

## Annotation layer (labels)

- +labs(title=" ", subtitle=" ")
  for title & subtitle, captions etc (outside data grid)
- annotate data (inside data grid)
  annotate("text", x= ,y= , label=" ")

## Saving Visualizations

-> Export option in plot window

- ggsave()
- png() ··· dev.off()
- pdf() ··· dev.off()

# Documentation and Reports

Wednesday, December 28, 2022 21:48

**R Markdown** : a file format for making dynamic documents
with R

**Markdown** : a syntax for formatting plain text files

- to italicize add _underscore_ or *asterisk*

**R Notebook** : lets users run your code and show the graphs and show
charts that visualize your code

## R Markdown

> lets you convert to HTML, PDF, Word documents
> slide presentation
> Dashboard

## Other Notebook Options

- Jupyter
- Kaggle
- Google Colab

**YAML** a language for data that translates it so it's readable

## Some other syntax :

- bullet points : *
  *
  *
- headers : ## Heading
- link : [click here](http://url)
- images : ![caption](image url)

## Code Chunks

**Delimiter** : a character that indicates the beginning or end
of a data item

code chunk delimiter:

~~~
```{r label for code chunk}
 . . .
```
~~~