Data Preprocessing Cheat Sheet in R

The questions we need to ask before everything getting started:
What is the goal?
What kind of plot do we want to create? …
package we use here: **tidyverse**
install.packages("tidyverse")
library(tidyverse)

Import Data

read_csv("mydata.csv") reads comma delimited files
read_csv2("mydata.csv") read semi-colon delimited files
read_delim("mydata.txt") reads files with any delimiter
read_tsv("mydata.tsv")/read_table("mydata.tsv") reads tab delimited files read fixed width files
read_fwf("mydata.tsv") reads fixed width files

Check Data

a. First n rows of data
head(mydata, n)
b. Descriptive statistics of each column of data, including min, max, mean, median, 1st and 3rd quantile
summary(mydata)
If you want more information such as sum, variance, standard deviation, etc.
install.packages("pastecs")
library(pastecs)
stat.desc(iris)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
nbr.val	150.00000000	150.00000000	150.0000000	150.00000000	NA
nbr.null	0.00000000	0.00000000	0.0000000	0.00000000	NA
nbr.na	0.00000000	0.00000000	0.0000000	0.00000000	NA
min	4.30000000	2.00000000	1.0000000	0.10000000	NA
max	7.90000000	4.40000000	6.9000000	2.50000000	NA
range	3.60000000	2.40000000	5.9000000	2.40000000	NA
sum	876.50000000	458.60000000	563.7000000	179.90000000	NA
median	5.80000000	3.00000000	4.3500000	1.30000000	NA
mean	5.84333333	3.05733333	3.7580000	1.19933333	NA
SE.mean	0.06761132	0.03558833	0.1441360	0.06223645	NA
CI.mean.0.95	0.13360085	0.07032302	0.2848146	0.12298004	NA
var	0.68569351	0.18997942	3.1162779	0.58100626	NA
std.dev	0.82806613	0.43586628	1.7652982	0.76223767	NA
coef.var	0.14171126	0.14256420	0.4697441	0.63555114	NA

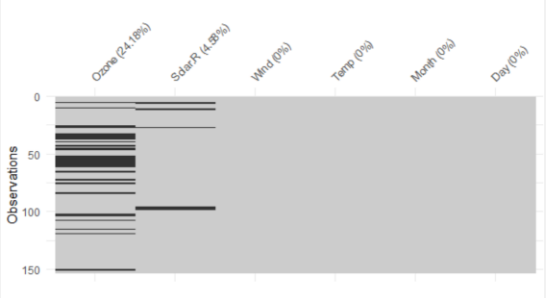
Missing Values

Check if the entry is missing

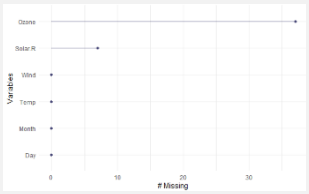
Check if each entry is missing
is.na(myata)
The number of missing values(NAs) in data
sum(is.na(mydata))

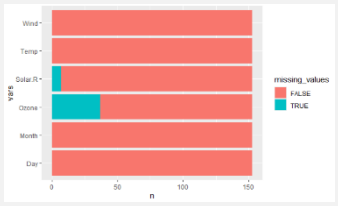
Visualize missing values

a. **vis_miss(mydata)**
install.packages("naniar")
library(naniar)
an at-a-glance ggplot showing percentage of missing values
need to install package naniar



b. **gg_miss_var(mydata)**
a ggplot showing the number of missings in each variable
c. **barplot**
a barplot showing missing values for each variable





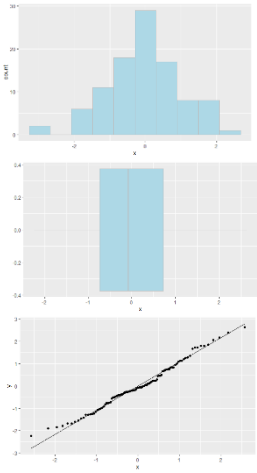
Drop, Replace or Fill in Missing Values

a. **drop_na(mydata)**
simply drop all rows and columns containing NAs
b. **fill(mydata, ..., .direction = c("down", "up", "downup", "updown"))**
fill NAs with previous value, next value, first down and then up, first up and then down respectively
c. **replace_na(mydata, replace=list(x=replace_value), ...)**
replace NAs with replace_values

Distributions

a. histogram
ggplot(data=mydata, aes(x=x)) + geom_histogram(bins=10, fill="lightblue", color="grey")
b. boxplot
ggplot(data=mydata, aes(x=x)) + geom_boxplot(fill="lightblue", color="grey")

Data is not always normal but if check normal:
1. histogram(see a.)
2. qqplot
if the points follow a straight line, then mydata is normally distributed
can also use shapiro.test() and ks.test()



Constraints

Data Type: if there are different data types in the same column...
a. check data type
typeof(value)
b. change type
as.character(value) / as.numeric(value) / as.integer(value) / as.Date(value, format=...)
Data Range/Reality Constraints
if there are reality limitation such as weight always being greater than 0 lbs, then data points containing negative weight might not make sense...
mydata %>% filter(weight >= 0) keeps data of weight >= 0
Uniqueness
a. check duplications
duplicated(mydata) / mydata[duplicated(mydata)]
b. remove duplicated terms
mydata[!duplicated(mydata)] / unique(mydata) / distinct(mydata)

Reshape

a. Pivot data from wide to long
1. **gather(data, key, value, ..., na.rm = FALSE, convert = FALSE)**
2. **pivot_longer(data, cols, names_to = "name", values_to="value", ...)**
b. Pivot data from long to wide
1. **spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE)**
2. **pivot_wider(data, names_from=name, values_from=value, ...)**

Reference
Nicholas Tierney, Gallery of Missing Data Visualisations, https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html
How to Test for Normality in R (4 Methods), https://www.statology.org/test-for-normality-in-r/