CheatSheet - Data Wrangling with Tidyverse

IBM Developer
SKILLS NETWORK

Commands	**Syntax**	**Description**	**Example**
install package	`install.packages("packagename")`	`install.packages` is used to install the packages from the R library.	`install.packages("tidyverse")`
load package	`library(packagename)`	`library()` Load the package from R library.	`library(tidyverse)`

download.file	`download.file(url, destfile, method, quiet = FALSE, mode = "w",cacheOK = TRUE,headers = NULL, …)`	`download.file() to download the file locally using the download.file() function. url naming the URL of a resource to be downloaded.`
		`destfile a character string with the name where the downloaded file is saved.`
			`download.file(url, destfile = "lax_to_jfk.tar.gz")`

untar	`untar()`	`untar()` is used to extract files from a tar archive is done with untar function from the utils package.	`untar("lax_to_jfk.tar.gz")`
read_csv	`read_csv(file)`	`read_csv()` reads the csv file using readr package.	`read_csv("lax_to_jfk/lax_to_jfk.csv")`
Missing Values and Formatting			
is.na	`is.na(x)`	`is.na(x)` returns a vector of TRUE or FALSE depending if the according element in x is NA or not.	`is.na(c(1, na))` # FALSE TRUE
anyNA	`anyNA(x, recursive = FALSE)`	`anyNA()` returns TRUE if x contains any NAs and FALSE otherwise.	`anyNA(c(1, na))` # TRUE
sum	`sum(object)`	`sum()` is used to calculate sum.	`sum(is.na(carrierdelay))`

summarize	`summarize(X, by, FUN, …,stat.name=deparse(substitute(X)),type=c('variables','matrix'), subset=TRUE,keepcolnames=FALSE)`	`summarize()` function reduces a data frame to a summary of just one vector or value. X a vector or matrix capable of being operated on by the function specified as the FUN argument by one or more stratification variables. If a single variable, by may be a vector, otherwise it should be a list. FUN a function of a single vector argument, used to create the statistical summaries for summarize. FUN may compute any number of statistics.	`summarize(count = sum(is.na(carrierdelay)))`
map	`map(.x, .f, ...)`	`map()` functions transform their input by applying a function to each element and returning a vector the same length as the input.	`map(sub_airline, ~sum(is.na(.)))`
dim	`dim(object)`	`dim` returns the dimension of the matrix, array, or data frame.	`dim(sub_airline)`
drop_na	`drop_na(object)`	`drop_na()` drop rows containing missing values.	`drop_na(carrierdelay)`
replace_na	`replace_na(data, replace, ...)`	`replace_na` replace missing values. data A data frame or vector. replace If data is a data frame, a named list giving the value to replace NA with for each column. If data is a vector, a single value used for replacement.	`replace_na(list(carrierdelay = 0, weatherdelay = 0, nasdelay = 0, securitydelay = 0, lateaircraftdelay = 0))`
mean	`mean(x, na.rm)`	`mean()` calculate the arithmetic mean of the elements of the numeric vector passed to it as argument.	`mean(drop_na_rows$carrierdelay)`
mutate, mutate_all, mutate_if	`mutate(data, ...)`	`mutate` function in R (mutate, mutate_all and mutate_at) is used to create new variable or column to the dataframe in R.	`date_airline %>% select(year, month, day) %>% mutate_all(type.convert) %>% mutate_if(is.character, as.numeric)`
Data Normalization			
Simple scaling	`xnew=xold/xmax`	Simple scaling divides each value by the maximum value in a feature. The new range is between 0 and 1.	`sub_airline$arrdelay / max(sub_airline$arrdelay)`

Min-max	`xnew= (xold-xmax) / (xmax-xmin)`	Min-max subtracts the minimum value from the original and divides by the maximum minus the minimum. The minimum becomes 0 and the maximum becomes 1.	`(sub_airline$arrdelay - min(sub_airline$arrdelay)) /(max(sub_airline$arrdelay) - min(sub_airline$arrdelay))`
Z-score	`xnew= (xold - μ) / σ`	Standardization (Z-score) subtracts the mean (μ) of the feature and divides by the standard deviation (σ).	`(sub_airline$arrdelay - mean(sub_airline$arrdelay)) / sd(sub_airline$arrdelay)`
Binning Data			
ggplot	`ggplot(df, aes(x, y, other aesthetics))`	`ggplot` is a plotting package that makes it simple to create complex plots from data in a data frame.	`ggplot(data = sub_airline, mapping = aes(x = arrdelay)) + geom_histogram(bins = 100, color = "white", fill = "red")`
ntile	`ntile(data)`	`ntile()` function is used to divide the data into N bins there by providing ntile rank.	`sub_airline %>% mutate(quantile_rank = ntile(sub_airline$arrdelay,4))`
geom_histogram	`geom_histogram(*arguments)`	`geom_histogram()` function display the counts with bars.	`geom_histogram(bins = 4, color = "white", fill = "red")`
Indicator variable			
spread	`spread(data, key, value)`	`spread` a key-value pair across multiple columns * data is your dataframe of interest. * key is the column whose values will become variable names. * value is the column where values will fill in under the new variables created from key.	`sub_airline %>% spread(reporting_airline, arrdelay)`
slice	`slice(num1 : num5 )`	`slice()`looks at the specified rows.	`slice(1:5)`
factor	`factor(x)`	`factor()` function is used to encode a vector as a factor, If argument ordered is TRUE, the factor levels are assumed to be ordered.	`sub_airline %>% mutate(reporting_airline = factor(reporting_airline,labels = c("aa", "as", "dl", "ua", "b6", "pa (1)", "hp", "tw", "vx")))`

Author(s)

D.M. Naidu

Changelog

Date	Version	Changed by	Change Description
2020-08-11	1.0	D.M. Naidu	Initial Version