Week_5_Data_cleaning

Cici

2019.12.26

LO:

1. Describe the features of tidy data.
2. Use different data types and data structures in R.
3. Understand the steps of data cleaning.
4. Clean a real-world dataset according to tidy/data cleaning principles.

Notes:

- Working directory

absolute & relative.

- Four steps of data cleaning. (PPT 13/18)

1. Screening.
2. Diagnosis.
3. Treatment.
4. Documenting.

Functions:

- To examine features of vectors and other objects.

class()

typeof()

length()

attributes()

- Objects can have attributes. Attributes are part of the object.

names()

dimnames()

dim()

class()

attributes()

- Others during data cleaning.

is.na()

apply(is.na(data), 2, which) # to find row & col of NA.

apply(X, MARGIN, FUN, . . .) #MARGIN: 1 indicates row, 2 indicates column, c(1,2) indicates row and column.

data.noNA <- data[complete.cases(data),] # to remove NA-containing rows.

complete.cases() #return a logical vector indicating which cases are complete, i.e., have no missing values.

duplicated(x, incomparables = FALSE, fromLast = FALSE, . . .) # to find whether there is duplicated rows in the dataset. # duplciated() will only give you the duplicated rows, but not the original rows, so we need the next line to get the originals. # fromLast: logical indicating if duplication should be considered from the reverse side, i.e., the last (or rightmost) of identical elements would correspond to duplicated = FALSE.

frw.idx <- which(duplicated(data.noNA))

rvs.idx <- which(duplicated(data.noNA,fromLast = TRUE))

data.noNA[c(frw.idx,rvs.idx),]

read.csv(". . . ",na.strings = c("NA",""))

drop_na() # from tidyr package.

anyNA() #return TRUE or FALSE.

as.POSIXct(x,tz = "America/Chicago",format("%m/%d/%Y %H:%M:%S")) #tz: time zone

gsub(pattern, replacement, x, ignore.case = FALSE, perl = FALSE, fixed = FALSE, useBytes = FALSE) #perform replacement of the first and all matches respectively.

separate(data, col, into, sep = ". . . ", removw = TRUE, convert = FALSE, extra = "warn", fill = "warn", . . .)

```
data.noNA$LOCATION <- gsub("[()]","", data.noNA$LOCATION, perl = T)

data.final <- separate(data.noNA,LOCATION, into = c("LATITUDE", "LONGITUDE"), sep = ",", remove = F, fi
```

gather(data, key = "key", value = "value",. . . , na.rm = FALSE, convert = FALSE, factor_key = FALSE) # in tidyr package. # combine multiple columns and collapses into key-value pairs.

spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL) #in tidyr #spread a key-value pair across multiple columns.

summary()

facet_grid(rows = NULL, cols = NULL, scales = "fixed", space = "fixed", shrink = TRUE, labeller = "label_value", as.table = TRUE, switch = NULL, drop = TRUE, margins = FALSE, facets = NULL) #separate data points by their groups # ggplot2 package

p=ggplot(plot.df,aes(x=carat,y=price,color=clarity))+geom_point()+facet_grid(clarity~.)