Missing value handling in R

MISSING VALUE HANDLING IN R

Missing value handling in R

In this session

1. Replace missing values with the mean
2. Replace missing values with the median
3. Replace missing values with an interpolated estimate
4. Replace missing values with a constant
5. Replace missing values using imputation
6. Replace missing values with a missing rank
7. Replace missing values with a dummy
8. Replace missing values with 0
9. Create an indicator variable for "missing."
10. Replace missing values with a string
11. Add an indicator variable showing which strings are considered "missing."
12. Delete columns that are missing too many values to be useful
13. Delete rows that are missing critical values

Missing value handling in R

We need data that is:

- Relevant
- Connected
- Accurate
- Enough to work with

Missing value handling in R

Example of missing values dataset

Column 0	age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
Tony	48	27		1	5	shrimp		Pepper
Donald	67	25	86	10	2	beef		Jane
Henry	69	21	95	6	1	chicken	62	Janet
Janet	62	21	110	3	1	beef		Henry
Nick		17		4				
Bruce	37	14	63		1	veggie		NA
Steve	83		77	7	1	chicken		n/a
Clint	27	9	118	9		shrimp	3	None
Wanda	19	7	52	2	2	shrimp		empty
Natasha	26	4	162	5	3			_
Carol		3	127	11	1	veggie	1	""""
Mandy	44	2	68	8	1	chicken		null

Missing value handling in R

Too many missing data == Swiss cheese

Missing value handling in R

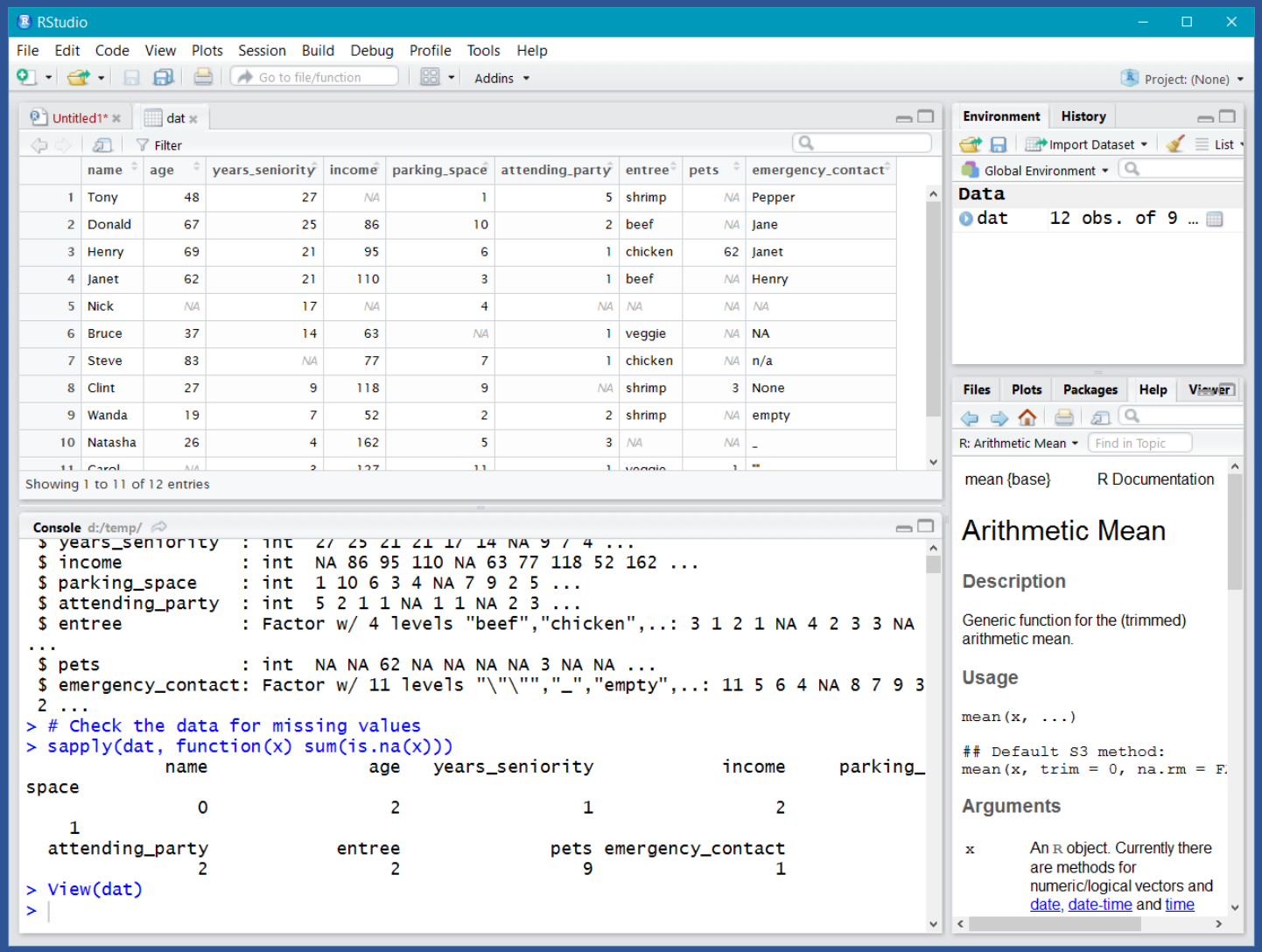
Example of missing values dataset CSV file

missing_values.csv

	A	B	C	D	E	F	G	H	I
1		age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
2	Tony	48	27		1	5	shrimp		Pepper
3	Donald	67	25	86	10	2	beef		Jane
4	Henry	69	21	95	6	1	chicken	62	Janet
5	Janet	62	21	110	3	1	beef		Henry
6	Nick		17		4				
7	Bruce	37	14	63		1	veggie		NA
8	Steve	83		77	7	1	chicken		n/a
9	Clint	27	9	118	9		shrimp	3	None
10	Wanda	19	7	52	2	2	shrimp		empty
11	Natasha	26	4	162	5	3			_
12	Carol		3	127	11	1	veggie	1	""
13	Mandy	44	2	68	8	1	chicken		null

Missing value handling in R

R Studio



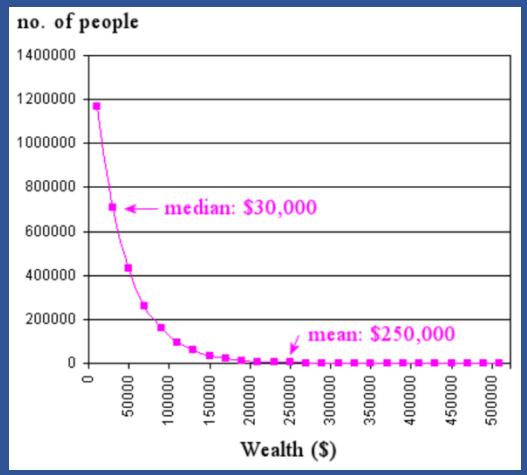
Missing value handling in R

General commands

```
rm(list = ls()) # clear work space
setwd("d:/temp") # set current work directory
sessionInfo() # get session information
installed.packages() # list installed packages
# import data file
dat <- read.csv("missing_values.csv", na.strings = "")
str(dat) # show data frame structure
# Check the data for missing values
sapply(dat, function(x) sum(is.na(x)))
```

Missing value handling in R

Replace missing values with the mean



Sample Mean	Population Mean
$\bar{x} = \frac{\Sigma x}{n}$	$\mu = \frac{\Sigma x}{N}$

where Σx is sum of all data values

N is number of data items in population

n is number of data items in sample

```r
# Replace missing values with the mean
# column = age
# Missing values type = distributed
# Formal name =  Missing Completely at Random (MCAR)
rm(list = ls()) # clear work space
setwd("d:/temp") # set current work directory
# import data file
dat <- read.csv("missing_values.csv", na.strings = "")
dat$age.mean <- ifelse(is.na(dat$age),
                       mean(dat$age, na.rm = TRUE),
                       dat$age)
```

Missing value handling in R

Replace missing values with the median

$$Median = l + \frac{h}{f}\left(\frac{N}{2} - c\right)$$

Where:

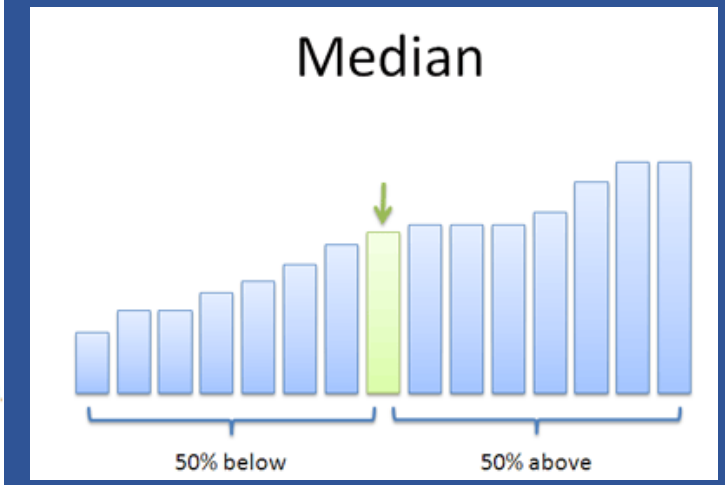
l = lower class boundary of the median class

h = Size of the median class interval

f = Frequency corresponding to the median class

N = Total number of observations i.e. sum of the frequencies

c = Cumulative frequency preceding median class.

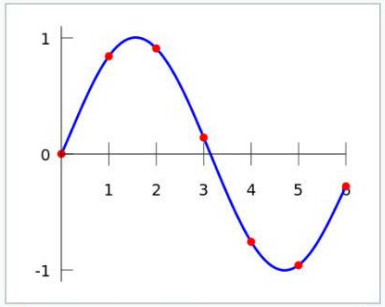



```
# Replace missing values with the median
# column = age
# • Another justifiable way to handle missing-at-random data
rm(list = ls()) # clear work space
setwd("d:/temp") # set current work directory
# import data file
dat <- read.csv("missing_values.csv", na.strings = "")
dat$age.mean <- ifelse(is.na(dat$age),
                       median(dat$age, na.rm = TRUE), dat$age)
```

Missing value handling in R

Replace missing values with an interpolated estimate

```
# Replace missing values with an interpolated estimate
# column = years_seniority
# the values in this column, years seniority, is ordered,
# greatest to least. This structure can be exploited by
# interpolating the missing value. This approach is very effective
# when it is appropriate, usually with time - series data.
rm(list = ls()) # clear work space
setwd("d:/temp") # set current work directory
# import data file
dat <- read.csv("missing_values.csv", na.strings = "")
dat$senior <- ifelse(is.na(dat$years_seniority), 11.5,
                     dat$years_seniority)
```



Plot of the data with polynomial interpolation applied

Missing value handling in R

Replace missing values with a constant

```
# Replace missing values with a constant
# column = income
# Missing values are Missing Not at Random(MNAR)
# Those with very high incomes preferred not to state them
# Make a reasonable guess for what "high" means and fill
# in the blanks. It will still be inaccurate, but better than blank

rm(list = ls()) # clear work space
setwd("d:/temp") # set current work directory
# import data file
dat <- read.csv("missing_values.csv", na.strings = "")
dat$income4 <- ifelse(is.na(dat$income), 250, dat$income)
```

Missing value handling in R

Replace missing values using imputation MICE

```
# Column = years_seniority, age, income
# Method = multivariate imputation by chained equations(MICE)
# MICE is the method of choice for complex incomplete data
# Good for missing data in more than one variable
# powerful when features are somewhat related

rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
# A fast, consistent tool for working with data frame
install.packages("dplyr")
library(dplyr) # attach packages
# Make new dataframe with column years_seniority,
# age, income with column type = numeric
dat <- dat %>% # make new 3 columns with type numberic
        mutate(
            senior1 = as.numeric(years_seniority),
            age1 = as.numeric(age),
            income1 = as.numeric(income)
            )
```

Missing value handling in R

Replace missing values using imputation MICE

```
# Replace missing values using imputation MICE
keep <- c("senior1", "age1", "income1")
#drop all columns but keep 3 col
dat <- dat[, keep, drop = FALSE]
install.packages('mice')        # standard command
library(mice)                   # standard command
init = mice(dat, maxit = 0)     # standard command
meth = init$method              # standard command
predM = init$predictorMatrix
# Bayesian linear regression (การวิเคราะห์ถดถอยเชิงเส้นแบบเบยส์)
meth[c("senior1")] = "norm"
Predictive mean matching
meth[c("age1")] = "pmm" #
meth[c("income1")] = "pmm"
# Replace missing values using imputation MICE
set.seed(103) # seed for pseudo random number generator
imputed = mice(dat, method = meth, predictorMatrix = predM, m = 5)
imputed <- complete(imputed)
```

Missing value handling in R

Replace missing values with a missing rank

```
# Replace missing values with a missing rank
# Column = parking_space
# Our knowledge of how parking spaces are numbered,
# let us make a guess here
# All the space numbers from 1 - 11
# Missing one might be 12
rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat$park <- ifelse(is.na(dat$parking_space), 12,
                   dat$parking_space)
```

Missing value handling in R

Replace missing values with a dummy

```
# eplace missing values with a dummy
# Column = parking_space
# Filling in a dummy value
# Clearly different from actual values
# Such as a negative rank
# Used to indicate that the feature is not applicable

rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat$park <- ifelse(is.na(dat$parking_space), -99,
                   dat$parking_space)
```

Missing value handling in R

Replace missing values with 0

```
# eplace missing values with 0
# Column = attending_party
# A missing numerical value can mean zero.
# In the case of an RSVP, invitees who are not planning
# to attend sometimes neglect to respond, but guests
# planning to attend are more likely to.
# In this case, filling in missing blanks with a zero is reasonable
rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat$party <- ifelse(is.na(dat$attending_party), 0, dat$attending_party)
```

Missing value handling in R

Create an indicator variable for "missing"

```
# Create an indicator variable for "missing"
# Column = pets
# Replacing missing values requires making assumptions.
# Whenever your confidence in those assumptions is low,
# it is safer to also create a true / false feature
# indicating that the value was missing.
# This allows many algorithms to learn to weight those differently.
rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat$pet1 <- ifelse(is.na(dat$pet), 0, dat$pets)
dat$pet2 <- complete.cases(dat$pets)
```

Missing value handling in R

Replace missing values with a string

```
# Replace missing values with a string
# Column = emergency_contact
# Replace NA with “no”

rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat$emer <- ifelse(is.na(dat$emergency_contact), 'no',
                   dat$emergency_contact)
```

Missing value handling in R

Add an indicator variable showing which strings are considered "missing."

```
# Add an indicator variable showing which strings
# are considered "missing."
# Column = emergency_contact
# There are lots of ways to communicate the concept
# of "missing" in a string
# Replace no, NA, n / a, None, _,"""",empty,null with 0
# Otherwise = 1
rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat$emer <- ifelse(dat[9] == 'NA' |
      is.na(dat[9]) | dat[9] == 'n/a' |
      dat[9] == 'None' | dat[9] == 'empty' |
      dat[9] == '_' | dat[9] == '""' | dat[9] == 'null', 0, 1)
```

Missing value handling in R

Delete columns that are missing too many values to be useful

```
# Column = pets
# Is a feature that missing too many values
# Not enough information available to make reasonable
# assumptions about how to replace the missing values
# Best policy = delete the column entirely.

rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat <- dat[, !(names(dat) %in% 'pets')]
```

Missing value handling in R

Delete rows that are missing critical values

```
# Delete rows that are missing critical values
# Rows that are missing important features can be deleted.
# This is particularly useful when you have the luxury of
# hand - picking high - quality data
# such as when training a model
rm(list = ls())
dat <- read.csv("missing_values.csv", na.strings = "")
dat <- dat[complete.cases(dat),]
```

Missing value handling in R

More information on Missing value handling in R

Bayesian linear regression analysis without tears (R)

https://www.r-bloggers.com/bayesian-linear-regression-analysis-without-tears-r/

Source code

https://github.com/laploy/ML/blob/master/Missing%20R%20Script.zip