Tutorial on Data preparation with R

Import data into R

Read .csv file.

Take titanic data as an example:

```
titanic <- read.csv("/Users/byu/Desktop/Data/titanic-train.csv", na.string = c(“”))  #
Use the path to the depository where you save the csv file. “na.string” is used to
specify missing values.
or
titanic <- read.table("/Users/byu/Desktop/Data/titanic-train.csv", sep=",",
header=TRUE, na.string = c("")) # “sep” indicates the field separator character
```

Note: Other format data files may need additional package to import, for example:

Read .xlsx file

```
Install.packages(“xlsx”) # install the package
library(xlsx)
titanic=read.xlsx("/Users/byu/Desktop/Data/titanic-train.xlsx", 1) # “1” is the sheet
index
```

Examine data definitions

List the structure of the data

```
str(titanic)
# It will show the number of total observations (rows) and variables (columns), as
well as the name and type (e.g. integer, factor, numeric) of each variable.
```

Note: R treats nominal variables as factors and ordinal variables as ordered factors.

```
survived_factor=factor(titanic$Survived)
str(survived_factor)
pclass_ordered=ordered(titanic$Pclass)
str(pclass_ordered)
mons=c("Jan", "Jan", "Feb", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
"Oct", "Oct", "Nov", "Dec", "Dec")
table(mons)
mons_factor=factor(mons, levels=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec"), ordered=TRUE)
table(mons_factors)
Or
levels=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
"Dec")
```

```
mons_factor=factor(mons, levels=levels, ordered=TRUE)
Table(mons_factors)
```

Handle missing values

Deal with missing data

- Find out missing value

```
is.na(titanic)  # Returns True and False. True represents missing value
is.na(titanic$Age) # List missing value for specific attribute

complete.cases(titanic) # Returns a logical vector indicating which cases are
complete. True represents NON missing value

# list rows of data that do NOT have missing values
titanic[complete.cases(titanic),]   # The square brackets indicates the index of
selected data with format [row, column].
nrow(titanic[complete.cases(titanic),])

# list rows of data that have missing values
titanic[!complete.cases(titanic),]  # The exclamation mark means NOT
# cout how many missing values in a column
view(titanic)
length(which(!is.na(titanic$Age)))
```

- Estimate missing value

Taking attribute “age” for example, one way is to replace missing values with the average age.

```
titanic$Age[is.na(titanic$Age)] <- mean(titanic$Age, na.rm = TRUE)
```

- Ignore the Missing Value During Analysis

```
titanic <- titanic[complete.cases(titanic),]
or
titanic <- na.omit(titanic)
```

Descriptive statistics

For numerical attribute “Age”:

```
mean(titanic$Age)
median(titanic$Age)
freq=table(titanic$Age) # frequency distribution
table(titanic$Age)[which.max(table(titanic$Age))]  # mode
var(titanic$Age) # variance
sd(titanic$Age) # standard deviation
max(titanic$Age)
```

```
min(titanic$Age)
range <- max(titanic$Age) - min(titanic$Age)
qt <- quantile(titanic$Age, na.rm=TRUE) # quartile, remove missing values
IQR=qt[['75%']]-qt[['25%']] # Interquartile range
summary(titanic)
# It will show the count number of individual value for factor variables and
minimum, maximum, and mean value for numeric variables.
```

- Count the number for factors

```
table(titanic$Sex) # It will show the number of female and male respectively.
```

- Summary

```
summary(titanic)
# It will show the count number of individual value for factor variables and
minimum, maximum, and mean value for numeric variables.
```

Which class is the most common, 1, 2, or 3?

```
freq=table(titanic$Pclass) # frequency distribution
table(titanic$Pclass)[which.max(table(titanic$Pclass))]  # mode
```

Visualization

```
# Histogram
hist(titanic$Age) # Note: the variable must be numeric
# Boxplot
boxplot(titanic$Age)
qt = quantile(titanic$Age, na.rm=TRUE) # quartile, remove missing values
IQR=qt[['75%']]-qt[['25%']] # Interquartile range
# Scatterplot
plot(titanic$Age, titanic$Fare)
# Crosstab
titanic.tab=table(titanic$Sex, titanic$Survived)
# Pie chart
pie(table(titanic$Sex))

# Boxplot
boxplot(titanic$Age)
qt = quantile(titanic$Age, na.rm=TRUE) # quartile, remove missing values
IQR=qt[['75%']]-qt[['25%']] # Interquartile range
```

Data Aggregation

```
library(xlsx)
```

Sample data(inserted in the slide) are weekly product sales in retail stores.

```
sales <- read.xlsx("/Users/byu/Desktop/data/sales.xlsx",1)
attach(sales)
```

Aggregate rows

```
# How many products were sold each day in each region?
salesByRegion <- aggregate(cbind(Mon,Tue,Wed,Thu,Fri,Sat,Sun),by=list(Group.region=Region),FUN=sum) # Calculate the total for each region
View(salesByRegion)
# Note: by variables must be in a list (even if there is only one)
```

Aggregate rows and columns

```
# What were the average sales for each region during the weekend?
InWeekend <- rowSums(sales[,c(“Sat”,“Sun”)]) # Sum column “Sat” and “Sun” by each row
salesNew <- data.frame(sales,InWeekend) # Add new column into original data frame
salesInWeekend <-aggregate(InWeekend, by=list(Region), FUN=mean) # Calculate the mean for each region
detach(sales)
```

Data transformation

Discretization:

Take attribute “Age” in Titanic data as example

```
# discretize age into seven bins
age <- cut(titanic$Age, breaks = c(0,10,20,30,40,50,60,Inf),labels=c("child","teens","twenties","thirties","fourties","fifties","old"))
```

Log transformation

```
plot(titanic$Age, log(titanic$Age))
```

Calculating Z-score with R

Using the attribute “Age” in Titanic data

```
# function “scale”
scale(titanic$Age, center = TRUE, scale = TRUE)
```

Or

```
(titanic$Age-mean(titanic$Age, na.rm = TRUE))/sd(titanic$Age, na.rm = TRUE)
plot(titanic$Age, scale(titanic$Age, center = TRUE, scale = TRUE))
```

Min-max transformation with R

```
Min_max <- (titanic$Age-min(titanic$Age,na.rm=TRUE))/(max(titanic$Age,na.rm=TRUE)-min(titanic$Age,na.rm=TRUE))
plot(Min_max, titanic$Age)
```

Random Sampling

Assuming we want to pick 100 records from Titanic data randomly, we could use the function “sample”

```
sample <- titanic[sample(1:nrow(titanic), 100, replace=FALSE), ]
# “nrow” is a function of counting the total row number of a dataset
# replace = FALSE represents sampling without replacement, while TRUE represents sampling with replacement.
View(sample)
table(sample$Survived)
table(titanic$Survived)
```

Systematic sampling:

```
ss=titanic[titanic$PassengerId%%10==0,] # sample lines #10, 20, 30, …
Nrow(ss)
```

Or

```
ss=titanic[seq(1, nrow(titanic),10),] # sample lines #1, #11, #21, …
```

Or

```
ss=titanic[seq(0, nrow(titanic),10),] # sample lines #10, #20, …
```