MSBA7002_Tutorial_1

Yutao DENG

2023-10-30

Contents

1 Introduction of R

1.1 R Markdown

1.1.1 Create a r chunk

by click or typing mac: option + command + I windows: Ctrl+Alt+I

1.1.2 Markdown Language

Try equations inline and aligned form

$$z = x_1^2 + x_2^2$$

$$\begin{aligned} x + 3 &= y + 3 \\ x &= y \end{aligned}$$

1.2 Numeric and string objects

```r
# store an object
scalar_x = 2
# print this object
scalar_x
# store and print an object
(scalar_x = 3)
# store a string object
str_x = "Hello"
# print a string
cat("Hello",str_x)
```

1.3 Vectors, Matrices and Dataframes

```r
# Vectors
vector_x = c(168, 177, 177, 177, 178, 172, 165, 171, 178, 170)
# Print the second component
vector_x[2]
# Print the second, the 3rd, the 4th and 5th component
vector_x[2:5]
# Define a vector as a sequence (1 to 10)
(obs = 1:10)
```

```r
# Matrices
# Create a matrix using 1:9
matrix_x = matrix(
  (1:9),
  nrow = 3,
  ncol = 3,
  byrow = TRUE
)

# Print the matrix
matrix_x
```

```r
# Accessing first and second row
matrix_x[1:2, ]
```

1.4 Defining functions and Control flow

```r
# Create f(x) = a/b, where b=2 by default
example_function = function(a, b=2) {
r=a/b
return(r)
}
# set a=b=1
example_function(a=1,b=1)

example_function(1,1)

# only set a=1
example_function(a=1)

# only set b=1
# example_function(b=1)
```

```r
# Control flows:
# If-else condition

# With one condition to check
a = 1
if(a == 1) {
  print(1)
} else {
  print(0)
}

# With multiple conditions
if(a == 1) {
  print(1)
} else if( a == 2) {
  print(2)
} else {
  print(0)
}

# Loops
dice <- c(1, 2, 3, 4, 5, 6)

for (x in dice) {
  print(x)
}
```

1.5 Others

1.5.1 Packages installation and importing

```r
# install.packages("ggplot2")
library(ggplot2)
```

1.5.2 Search for guide

```r
# local search
# ?lm() # linear model
# ? : local searching
```

```r
# global search
# ??glmnet() # lasso regression

## ?? global searching
## ?? is time-consuming
```

2 EDA

```r
# Load the smartphone data
sp <- read.csv("data/smartphone.dat")
```

```r
# Show the dimensions, colnames, structure
dim(sp)
colnames(sp)
str(sp)
summary(sp)
```

```r
# Preview the first and last two lines of the data
head(sp)
head(sp,2)
tail(sp,2)
```

```r
# Check missing values in each roll
apply(sp, 2, function(x) sum(is.na(x))) #2: column; #1: row

# Chek missing values in each column
apply(sp, 1, function(x) sum(is.na(x))) #2: column; #1: row

# Drop the missing values
sp <- na.omit(sp)
```

3 Linear Model

3.1 Linear Regression

```r
# With intercept
lm_fit <- lm(Rating ~ Age + Income + Group, data = sp)
summary(lm_fit)
```

```r
# Without intercept
lm_fit <- lm(Rating ~ Age + Income + Group - 1, data = sp)
summary(lm_fit)
```

```r
# use all the columns
lm_fit <- lm(Rating ~ ., data = sp)
summary(lm_fit)
```

4 Model Selection

4.1 ANOVA

4.1.1 Type-I

```r
colnames(sp)
```

```r
# fit the linear models
# order: Income -> Age -> Group
## order: prior knowledge from life experience ,literature, common knowledge
lm_fit_1.1 <- lm(Rating ~ Income, sp)
lm_fit_1.2 <- lm(Rating ~ Income + Age, sp)
lm_fit_1.3 <- lm(Rating ~ Income + Age + Group, sp)
```

```r
# apply anova in the order
## Test coeffcient is significant
## H0: beta-hat is 0
## Reject H0

## Test SSR ratio
### H0: the ratio is 1
### Reject H0

# p-value < 0.05 => longer model
# p-value > 0.05 => shorter model
anova(lm_fit_1.1, lm_fit_1.2)
cat("--------------------------------------------------------\n\n")

anova(lm_fit_1.2, lm_fit_1.3)
cat("--------------------------------------------------------\n\n")

summary(lm_fit_1.2)
```

4.1.2 Type-II

```r
# install.package("car") # if you did not install it before
library(car)
```

```
## Loading required package: carData
```

```r
# fit a lm with all variables
lm_fit2 <- lm(Rating ~ ., sp)

# apply Anova in library(car)
# p < 0.05(**) you should not delete it. 0.01/0.05/0.10
# p > 0.05(**) you are recommended to delete it
Anova(lm_fit2)

# Show the final model
```

```r
lm_fit2_final <- lm(Rating ~ Age + Income, sp)
summary(lm_fit2_final)
```

4.2 Plot(Basic and ggplot2)

4.2.1 Basic plot

```r
# Plot the income and residual from the previous model
plot(sp$Income, lm_fit2_final$residuals)
```

```r
# Change the size of figure and show two plots together
par(mfcol=c(2,1), mar = c(2, 4, 2, 1))
plot(sp$Income, lm_fit2_final$residuals)
plot(sp$Income, lm_fit2_final$residuals)
```

4.2.2 ggplot2

```r
# install.packages("ggplot2")
library(ggplot2)
```

```r
# create a data frame with two columns (Income, group, residual from lm_fit2_final)
sp_res <- data.frame(Income = sp$Income, Group=sp$Group, resid = lm_fit2_final$residuals)

# show the first 5 samples
head(sp_res, 5)
```

```r
# aes: aesthetic
ggplot(sp_res, aes(x = Income, y = resid, color = Group)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red")
```

4.3 dplyr

```r
# install.packages("dplyr")
library(dplyr) ## %>%
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# colnames(sp) # without dplyr
sp %>%
  colnames()
```

```r
# sum(is.na(sp)) # without dplyr
sp %>%
  is.na() %>%
  sum()
```

```r
# select from sp data where Group is Med
sp %>%
  filter(Group=='Med')

# select from sp data where Group is Med or Low
sp %>% filter(Group %in% c('Med', 'Low'))
```

```r
# select some column by name
sp %>% select(c('Age','Group'))

# select some column by index
sp %>% select(c(1,2,4))

# use select and filter at the same time
sp %>% select(c(1,2,4)) %>% filter(Group %in% c('Med', 'Low'))
```

```r
# gourp by Group
grp_sp <- sp %>% group_by(Group)
sp
grp_sp
grp_sp %>% summarise(sum(Income))
```

```r
sp_res %>%
  ggplot(aes(x = Income, y = resid, color = Group)) +
  geom_point()
```

```r
# colnames(sp)[1] <- "test"
sp <- sp %>% rename(test = Age)
colnames(sp)
```

4.4 Regularization

```r
# Now use the cars04.dat
data.car0 <- read.csv("data/cars04.dat")
```

```r
# Check the missing value
data.car0 %>% apply(2, function(x) sum(is.na(x)))

# Follow the file Regularization_Car04 in class (Find and copy them here)
data.car <- data.car0[,-c(9,10,12,22,23,25)]
data.car$GPHM <- 100/data.car$MPG.City
data.car <- data.car[,-1]
data.car <- data.car[,-c(18,19)]
data.car$Model.Year <- data.car$Model.Year %>% factor()

data.car <- na.omit(data.car)
sum(is.na(data.car))

# Check the data agian
str(data.car)
```

4.4.1 Categorical variable

4.4.2 model.matrix()

explain onehot transformation

```r
# Recall that lm() will automatically create dummies for the categorical variable
# We need to do one-hot transformation by ourselves in other models.
# Example: Sex: F or M, then we can set F as 1 and M as 0.
# Q: How many dummies are needed for a categorical variable that takes n values?

# Use model.matrix to apply one-hot
x <- model.matrix(GPHM~. , data.car)[,-1] # -1: exclude the intercept
y <- data.car$GPHM
colnames(x)
```

4.4.3 Lasso & Ridge

```r
# lasso/ridge regression
#install.packages("glmnet")
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
# alpha = 1; lasso regression
# alpha = 0; ridge regreesion
fit.lasso0 <- glmnet(x = x, y = y, alpha = 1, lambda = 0.01)
fit.lasso0$beta #. : 0
```

4.4.4 Cross-Validation

```r
# Remember to set random seed to make sure that you can replicate your results
set.seed(10)

# how to choose lambda? which lambda is the best? <= cross-validation
cv.lasso <- cv.glmnet(x = x, y = y, alpha = 1)
cv.lasso$lambda.min # lambda will achieve minimal CV SSE

# cv -> train-test dataset split (random) -> set.seed

# You could specify the range of lambda you would like to search
# cv.lasso <- cv.glmnet(x = x, y = y, alpha = 1, lambda = 10^seq(-5,0,by = 0.05))
```

```r
# Now use the lambda from cv to fit a new lasso
fit.lasso1 <- glmnet(x = x, y = y, alpha = 1, lambda = cv.lasso$lambda.min)
fit.lasso1$beta

# select the X's with nonzero coefficients
which(fit.lasso1$beta != 0) # the index of beta estimation which is not 0
rownames(fit.lasso1$beta)[which(fit.lasso1$beta != 0)]
```

```r
# construct a new data using variables selected by lasso
sel_x = rownames(fit.lasso1$beta)[which(fit.lasso1$beta != 0)]
data.car1 <- data.frame(GPHM = y, x[,sel_x])
```

```r
# Fit another lm using the new data
lm_fit_lasso <- lm(GPHM ~ ., data.car1)
summary(lm_fit_lasso)
```

4.5 Subset selection

4.5.1 Best subset selection

```r
## lasso regression -> variable selection
## while, unfortunately, still some coefficients are insignificant
## we adopt best subset selection to select variables further.

#install.packages("leaps")
library(leaps)
```

```r
# Fit a best subset selection model using regsubsets
# nvmax: maximal of number of X
fit.exh <- regsubsets(GPHM ~ ., data.car1, method = "exhaustive", nvmax = dim(data.car1)[2], intercept =
summary(fit.exh)
## matrix above tells us which X's should be kept given the number of X's kept
```

```r
# what is the best number of X's we should keep
# Use BIC as the criteria
f.e <- summary(fit.exh)
f.e$bic

# which is the minimum
which.min(f.e$bic)

# get the subset with minimal BIC
f.e$which[9,]
sel_x2 = (names(f.e$which[9,][f.e$which[9,]][-1]))

# create a new data with the best subset
data.car2 <- data.frame(GPHM = y, data.car1[,sel_x2])
```

```r
# use lm to fit the final model with best subset
ss_fit_final <- lm(GPHM ~ ., data.car2)
summary(ss_fit_final)
```

4.5.2 Forward, backward

```r
regsubsets(GPHM~., data.car1, method = "backward", nvmax = 15) %>% summary()
```