# CS544 Final Project

# Shibo Cao

# dataset: Ramen rating

### Part 1: Data processing

# import statistical packages

library(dplyr)

library(sampling)

library(stringr)

# set working directory

setwd("~/Desktop/544/CS544Final\_Cao")

# read file

ramen\_data <- read.csv("ramen-ratings.csv")

# eliminate all data with no information for Stars and Style

ramen\_data <- ramen\_data[ramen\_data$Stars != "Unrated",]

ramen\_data <- ramen\_data[ramen\_data$Style != "",]

# set all Star rating with two decimal places

ramen\_data$Stars <- as.character(ramen\_data$Stars)

for (i in 1: nrow(ramen\_data)) {

if (nchar(ramen\_data$Stars[i]) == 1){

ramen\_data$Stars[i] <- paste(ramen\_data$Stars[i], ".", "00", sep = "")

}

if (nchar(ramen\_data$Stars[i]) == 3){

ramen\_data$Stars[i] <- paste(ramen\_data$Star[i], "0", sep = "")

}

}

ramen\_data$Stars <- as.numeric(ramen\_data$Stars)

# In "Conutry" column, convert all "United States" to "USA"

row\_USA\_num <- which(ramen\_data$Country == "United States")

row\_USA <- ramen\_data[row\_USA\_num,]

row\_USA$Country <- "USA"

ramen\_data[row\_USA\_num,] <- row\_USA

### Part 2A: Categorical data analysis: origin country of product

# frequency table:

ramen\_freq <- table(ramen\_data$Country)

# sort out the table based on frequency with descending order

ramen\_freq\_ordered <- ramen\_data %>%

count(Country) %>%

arrange(desc(n))

ramen\_freq\_ordered <- as.data.frame(ramen\_freq\_ordered)

# Combine those categories with a frequency less than 100 into "Others"

# locate the index and calculate the sum

less\_than\_hundred\_index <- which(ramen\_freq\_ordered$n <= 100)

total\_freq\_others <- sum(ramen\_freq\_ordered$n[less\_than\_hundred\_index])

ramen\_freq\_ordered <- subset(ramen\_freq\_ordered, n > 100)

Others\_row <- data.frame(Country = "Others", n = total\_freq\_others)

ramen\_freq\_ordered <- rbind(ramen\_freq\_ordered, Others\_row)

> ramen\_freq\_ordered

Country n

1 Japan 352

2 USA 324

3 South Korea 307

4 Taiwan 223

5 Thailand 191

6 China 168

7 Malaysia 155

8 Hong Kong 137

9 Indonesia 126

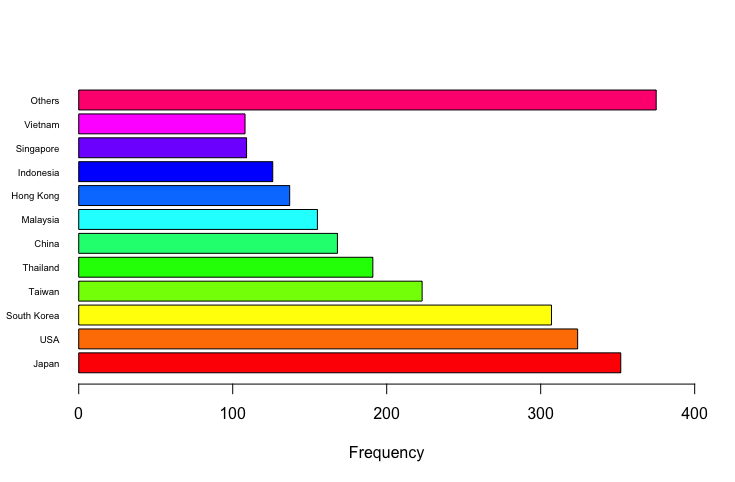
10 Singapore 109

11 Vietnam 108

12 Others 375

# bar plot:

barplot(as.vector(ramen\_freq\_ordered$n), horiz = TRUE, xlim = c(0,400), col = rainbow(nrow(ramen\_freq\_ordered)), xlab = "Frequency", names.arg = ramen\_freq\_ordered$Country, las = 1, cex.names=0.6)



### Part 2B: Numerical data analysis: Stars rating

# mean, median and mode of stars rating across all data

mean\_rating <- mean(ramen\_data$Stars, trim = 0.1)

> mean\_rating

[1] 3.765175

median\_rating <- median(ramen\_data$Stars)

> median\_rating

[1] 3.75

mode\_rating <- which(table(ramen\_data$Stars) == max(table(ramen\_data$Stars)))

> mode\_rating

4

36

# distribution of data

range\_rating <- range(ramen\_data$Stars)

> range\_rating

[1] 0 5

variance\_rating <- var(ramen\_data$Stars)

> variance\_rating

[1] 1.031527

stdev\_rating <- sd(ramen\_data$Stars)

> stdev\_rating

[1] 1.015641

f <- fivenum(ramen\_data$Stars)

> f

[1] 0.00 3.25 3.75 4.25 5.00

Quantile\_range <- quantile(ramen\_data$Stars, c(0, 0.25, 0.5, 0.75, 1))

> Quantile\_range

0% 25% 50% 75% 100%

0.00 3.25 3.75 4.25 5.00

Interquar\_range <- IQR(ramen\_data$Stars)

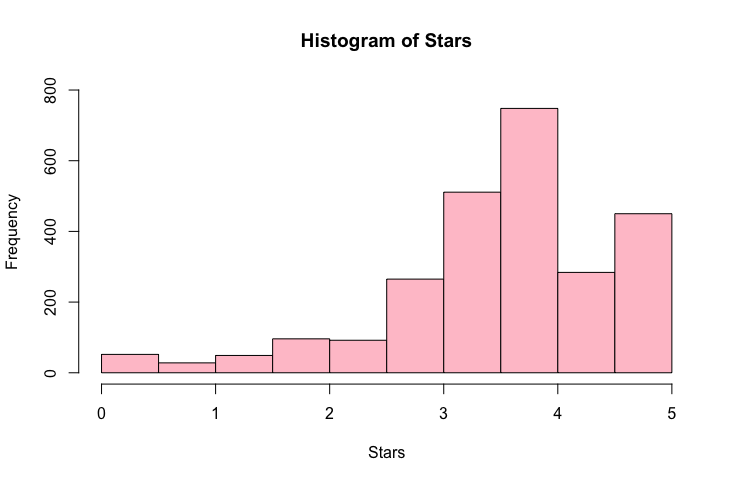
> Interquar\_range

[1] 1

# histogram and boxplot of Stars rating

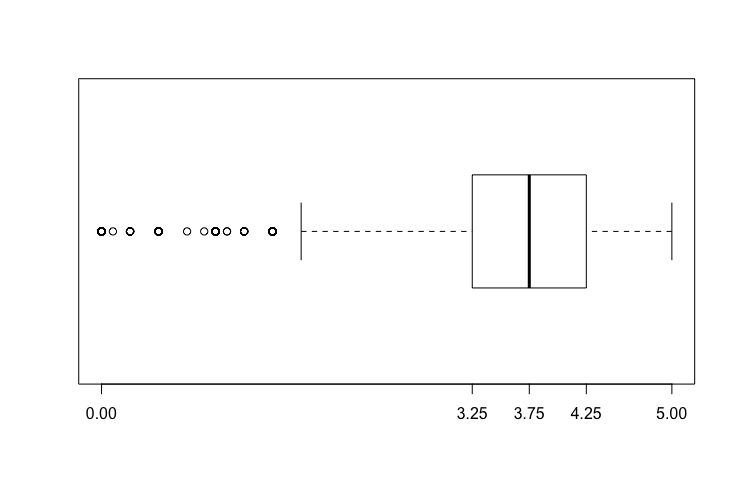
attach(ramen\_data)

hist\_stars <- hist(Stars, col = hcl(0), xlim = c(0,5), ylim = c(0, 800))



boxplot(Stars, horizontal = TRUE, xaxt = "n")

axis(side = 1, at= fivenum(Stars), labels = TRUE)



# outliers

outlier\_range <- c(f[2] - 1.5\*(f[4] - f[2]))

> outlier\_range

[1] 1.75

outlier\_num <- Stars[Stars <= outlier\_range]

> outlier\_num

[1] 1.00 0.25 1.50 1.50 0.00 1.00 0.00 1.50 0.00 0.25 1.00 0.50 1.00 1.00 0.00 1.00

[17] 1.00 1.00 0.00 1.00 0.00 1.75 0.00 1.50 0.00 0.00 0.00 1.50 1.00 0.25 0.50 1.50

[33] 0.75 0.25 0.00 1.50 0.00 0.00 0.50 1.50 0.25 0.00 0.00 1.00 1.25 1.50 0.25 0.00

[49] 0.00 0.25 1.00 1.50 1.75 1.50 1.00 1.75 0.50 0.50 1.75 0.00 1.50 1.75 1.50 1.25

[65] 1.50 1.00 1.75 0.50 1.00 0.50 1.00 1.75 1.50 1.50 1.00 1.50 1.00 1.75 1.25 1.50

[81] 1.75 1.25 1.75 1.75 1.00 1.50 0.25 1.75 1.50 1.25 1.50 1.10 0.50 1.25 1.25 1.25

[97] 1.75 1.50 1.50 1.50 0.00 0.00 1.75 1.50 0.50 0.90 1.50 0.25 1.00 0.00 1.75 1.75

[113] 1.50 1.75 0.25 1.50 0.00 1.25 0.50 0.50 1.25 1.75 1.75 0.00 1.50 1.75 1.75 1.50

[129] 0.25 1.50 0.00 1.50 1.75 1.75 1.75 1.75 1.75 0.00 0.10 1.50 1.50 1.10 0.50 1.50

[145] 1.00 1.75 1.50 1.00 1.00 0.00 0.50 1.00 1.50 1.00 1.00 0.50

### Part 3: Bivariate analysis: Country and Style

country\_style\_table <- rbind(table(Country, Style))

> country\_style\_table

Bar Bowl Box Can Cup Pack Tray

Australia 0 0 0 0 0 17 5 0

Bangladesh 0 0 0 0 0 0 7 0

Brazil 0 0 0 0 0 2 3 0

Cambodia 0 0 0 0 0 0 5 0

Canada 0 0 8 0 0 17 16 0

China 0 0 45 0 0 16 98 9

Colombia 0 0 0 0 0 3 3 0

Dubai 0 0 0 0 0 0 3 0

Estonia 0 0 0 0 0 0 2 0

Fiji 0 0 0 0 0 0 4 0

Finland 0 0 0 0 0 0 3 0

Germany 0 0 0 0 0 11 16 0

Ghana 0 0 0 0 0 0 2 0

Holland 0 0 0 0 0 0 4 0

Hong Kong 0 0 30 0 0 38 67 2

Hungary 0 0 0 0 0 0 9 0

India 0 0 0 0 0 3 28 0

Indonesia 0 0 0 1 0 21 104 0

Japan 0 0 126 2 0 49 155 20

Malaysia 0 0 8 2 0 21 124 0

Mexico 0 0 0 0 0 15 10 0

Myanmar 0 0 0 0 0 3 11 0

Nepal 0 0 0 0 0 0 14 0

Netherlands 0 0 0 0 0 3 12 0

Nigeria 0 0 0 0 0 0 1 0

Pakistan 0 0 0 0 0 0 9 0

Philippines 0 0 10 0 0 4 33 0

Poland 0 0 0 0 0 0 4 0

Sarawak 0 0 0 0 0 0 3 0

Singapore 0 0 13 0 0 27 69 0

South Korea 0 0 68 0 0 40 181 18

Sweden 0 0 0 0 0 0 3 0

Taiwan 0 0 37 0 0 2 181 3

Thailand 0 0 44 0 0 48 97 2

UK 0 0 2 0 0 32 35 0

United States 0 0 0 0 0 0 0 0

USA 0 1 70 1 1 70 129 52

Vietnam 0 0 20 0 0 8 78 2

# distribution of percentage of each style of ramen different countries produce

country\_style\_prop\_table <- prop.table(country\_style\_table,1)

> country\_style\_prop\_table

Bar Bowl Box Can Cup Pack

Australia 0 0.00000000 0.00000000 0.000000000 0.00000000 0.77272727 0.2272727

Bangladesh 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Brazil 0 0.00000000 0.00000000 0.000000000 0.00000000 0.40000000 0.6000000

Cambodia 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Canada 0 0.00000000 0.19512195 0.000000000 0.00000000 0.41463415 0.3902439

China 0 0.00000000 0.26785714 0.000000000 0.00000000 0.09523810 0.5833333

Colombia 0 0.00000000 0.00000000 0.000000000 0.00000000 0.50000000 0.5000000

Dubai 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Estonia 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Fiji 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Finland 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Germany 0 0.00000000 0.00000000 0.000000000 0.00000000 0.40740741 0.5925926

Ghana 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Holland 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Hong Kong 0 0.00000000 0.21897810 0.000000000 0.00000000 0.27737226 0.4890511

Hungary 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

India 0 0.00000000 0.00000000 0.000000000 0.00000000 0.09677419 0.9032258

Indonesia 0 0.00000000 0.00000000 0.007936508 0.00000000 0.16666667 0.8253968

Japan 0 0.00000000 0.35795455 0.005681818 0.00000000 0.13920455 0.4403409

Malaysia 0 0.00000000 0.05161290 0.012903226 0.00000000 0.13548387 0.8000000

Mexico 0 0.00000000 0.00000000 0.000000000 0.00000000 0.60000000 0.4000000

Myanmar 0 0.00000000 0.00000000 0.000000000 0.00000000 0.21428571 0.7857143

Nepal 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Netherlands 0 0.00000000 0.00000000 0.000000000 0.00000000 0.20000000 0.8000000

Nigeria 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Pakistan 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Philippines 0 0.00000000 0.21276596 0.000000000 0.00000000 0.08510638 0.7021277

Poland 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Sarawak 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Singapore 0 0.00000000 0.11926606 0.000000000 0.00000000 0.24770642 0.6330275

South Korea 0 0.00000000 0.22149837 0.000000000 0.00000000 0.13029316 0.5895765

Sweden 0 0.00000000 0.00000000 0.000000000 0.00000000 0.00000000 1.0000000

Taiwan 0 0.00000000 0.16591928 0.000000000 0.00000000 0.00896861 0.8116592

Thailand 0 0.00000000 0.23036649 0.000000000 0.00000000 0.25130890 0.5078534

UK 0 0.00000000 0.02898551 0.000000000 0.00000000 0.46376812 0.5072464

United States NaN NaN NaN NaN NaN NaN NaN

USA 0 0.00308642 0.21604938 0.003086420 0.00308642 0.21604938 0.3981481

Vietnam 0 0.00000000 0.18518519 0.000000000 0.00000000 0.07407407 0.7222222

Tray

Australia 0.00000000

Bangladesh 0.00000000

Brazil 0.00000000

Cambodia 0.00000000

Canada 0.00000000

China 0.05357143

Colombia 0.00000000

Dubai 0.00000000

Estonia 0.00000000

Fiji 0.00000000

Finland 0.00000000

Germany 0.00000000

Ghana 0.00000000

Holland 0.00000000

Hong Kong 0.01459854

Hungary 0.00000000

India 0.00000000

Indonesia 0.00000000

Japan 0.05681818

Malaysia 0.00000000

Mexico 0.00000000

Myanmar 0.00000000

Nepal 0.00000000

Netherlands 0.00000000

Nigeria 0.00000000

Pakistan 0.00000000

Philippines 0.00000000

Poland 0.00000000

Sarawak 0.00000000

Singapore 0.00000000

South Korea 0.05863192

Sweden 0.00000000

Taiwan 0.01345291

Thailand 0.01047120

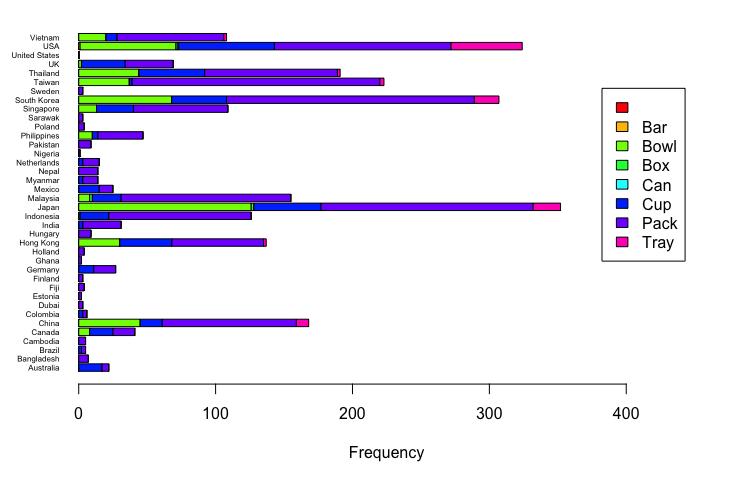
UK 0.00000000

United States NaN

USA 0.16049383

Vietnam 0.01851852

barplot(t(country\_style\_table), horiz = TRUE, col = rainbow(ncol(country\_style\_table)), xlab = "Frequency", xlim = c(0, 450), ylim = c(0,38), las = 1, cex.names=0.5, legend.text=TRUE)



### Part 4: Distribution of numerical data

# assume the Star rating follows normal distribution

# PDF

x <- seq(0,5, by = 0.1)

pdf <- dnorm(x, mean = mean\_rating, sd = stdev\_rating)

> pdf

[1] 0.0004072488 0.0005838167 0.0008288633 0.0011654109 0.0016228003 0.0022379010 0.0030563736

[8] 0.0041339175 0.0055374135 0.0073458474 0.0096508739 0.0125568633 0.0161802596 0.0206480763

[15] 0.0260953723 0.0326615800 0.0404856087 0.0496997206 0.0604222666 0.0727494759 0.0867466109

[22] 0.1024389114 0.1198028594 0.1387583747 0.1591625965 0.1808059082 0.2034108015 0.2266340686

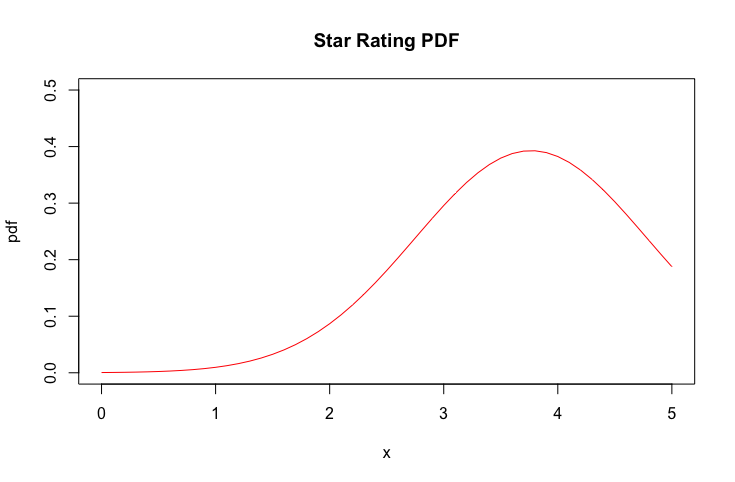
[29] 0.2500726353 0.2732731337 0.2957450568 0.3169770672 0.3364557712 0.3536860351 0.3682117422

[36] 0.3796357880 0.3876380942 0.3919905020 0.3925675789 0.3893526241 0.3824384712 0.3720230354

[43] 0.3583999013 0.3419445780 0.3230973182 0.3023435971 0.2801934568 0.2571609336 0.2337447091

[50] 0.2104109687 0.1875792320

plot(x, pdf, type = "l", col = "red", xlim = c(0,5), ylim = c(0,0.5), main = "Star Rating PDF" )



# CDF

x <- seq(0,5, by = 0.1)

cdf <- pnorm(x, mean = mean\_rating, sd = stdev\_rating)

> cdf

[1] 0.0001047860 0.0001538500 0.0002238262 0.0003226657 0.0004609281 0.0006524730 0.0009152762

[8] 0.0012723710 0.0017529112 0.0023933375 0.0032386196 0.0043435308 0.0057738939 0.0076077210

[15] 0.0099361535 0.0128640930 0.0165104041 0.0210075661 0.0265006538 0.0331455426 0.0411062577

[22] 0.0505514246 0.0616498246 0.0745651188 0.0894498643 0.1064390169 0.1256431722 0.1471418567

[29] 0.1709772189 0.1971484898 0.2256075833 0.2562561733 0.2889445329 0.3234723361 0.3595915222

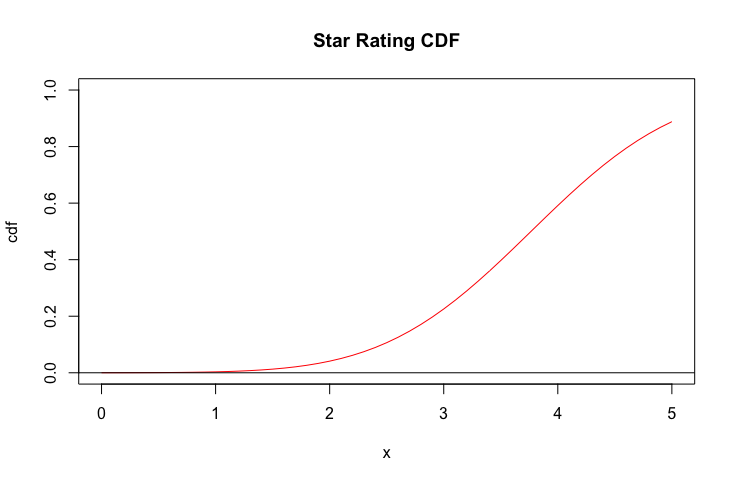
[36] 0.3970112098 0.4354045192 0.4744170506 0.5136766537 0.5528040429 0.5914237545 0.6291749208

[43] 0.6657213468 0.7007604225 0.7340304808 0.7653163102 0.7944526487 0.8213256062 0.8458720819

[50] 0.8680773498 0.8879710733

plot(x, cdf, type = "l", col = "red", xlim = c(0,5), ylim = c(0,1), main = "Star Rating CDF" )

abline(h = 0)



### Part 5: Central Limit theorem

# if sample size = 50, sampling 1000 times

samples <- 1000

sample.size <- 50

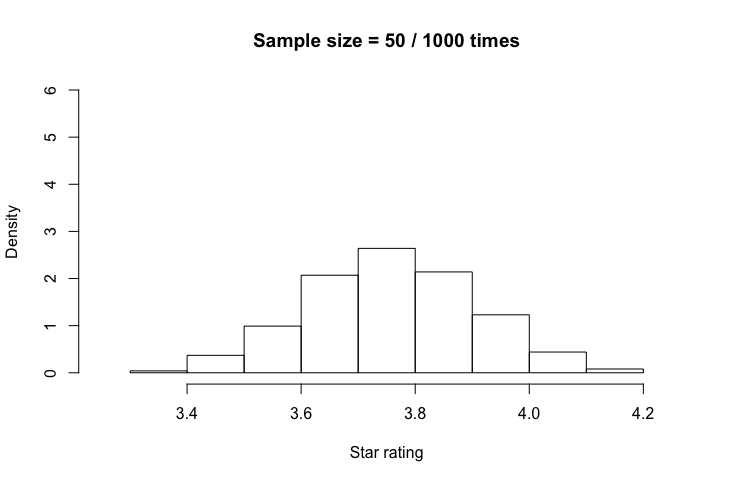
xbar <- numeric(samples)

for (i in 1: samples){

xbar[i] <- mean(rnorm(sample.size, mean = mean\_rating, sd = stdev\_rating))

}

hist(xbar, prob = TRUE, xlim = c(3.25,4.25), breaks = 10, ylim = c(0,6), xlab = "Star rating", main = "Sample size = 50 / 1000 times")



# if sample size = 100, sampling 1000 times

samples <- 1000

sample.size <- 100

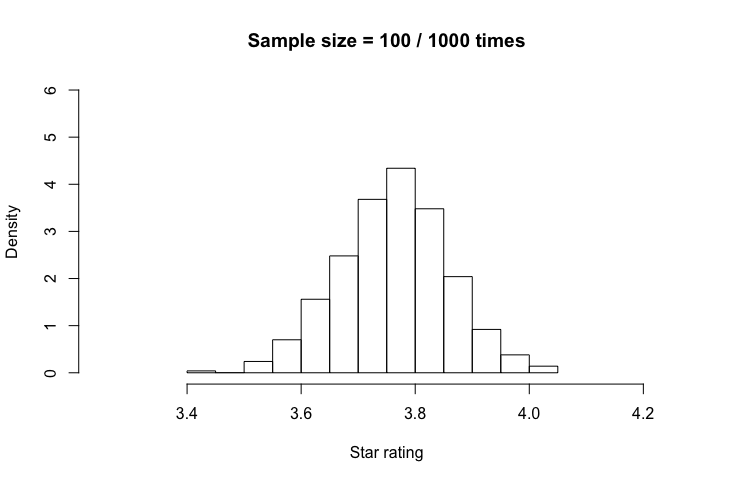
xbar <- numeric(samples)

for (i in 1: samples){

xbar[i] <- mean(rnorm(sample.size, mean = mean\_rating, sd = stdev\_rating))

}

hist(xbar, prob = TRUE, xlim = c(3.25,4.25), breaks = 10, ylim = c(0,6), xlab = "Star rating", main = "Sample size = 100 / 1000 times")



# if sample size = 200, sampling 1000 times

samples <- 1000

sample.size <- 200

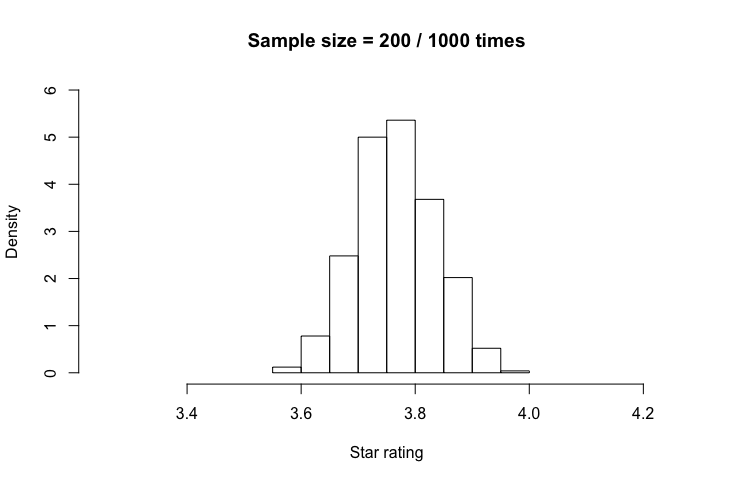
xbar <- numeric(samples)

for (i in 1: samples){

xbar[i] <- mean(rnorm(sample.size, mean = mean\_rating, sd = stdev\_rating))

}

hist(xbar, prob = TRUE, xlim = c(3.25,4.25), breaks = 10, ylim = c(0,6), xlab = "Star rating", main = "Sample size = 200 / 1000 times")



### Part 6: Sampling method

# Simple random sampling (100 sample without replacement)

s <- srswor(100, nrow(ramen\_data))

sample\_1 <- ramen\_data[s != 0,]

# Systematic sampling (100)

N <- nrow(ramen\_data)

n <- 100

k <- ceiling(N / n)

r <- sample(k,1)

s <- seq(r, by = k, length = n)

sample\_2 <- ramen\_data[s,]

# Unequal stratified sampling based on Style

# order data

index <- order(ramen\_data$Style)

style\_ordered\_data <- ramen\_data[index,]

# Stratified sample

freq\_style <- table(style\_ordered\_data$Style)

str\_size <- round(100 \* freq\_style / sum(freq\_style))

# Based on the str\_size, we do not draw any sample from some of the categories

# therefore we drop them by create a new subset

subset <- ramen\_data[!ramen\_data$Style %in% c("Bar", "Box", "Can"), ]

subset <- subset[order(subset$Style),]

freq\_style\_subset <- table(subset$Style)

freq\_style\_subset <- freq\_style\_subset[freq\_style\_subset != 0]

str\_size\_subset <- str\_size[str\_size != 0]

sample\_3 <- getdata(style\_ordered\_data,strata(subset, stratanames = c("Style"), size = str\_size\_subset, method = "srswor", description = TRUE))

# Cluster sampling

# Choose 3 out of 6 styles

cl <- cluster(ramen\_data, c("Style"), size = 3, method = "srswor")

first\_cluster <- getdata(ramen\_data, cl)

# 2nd step sampling: simple random sampling of 100 samples

s <- srswor(100, nrow(first\_cluster))

sample\_4 <- first\_cluster[s != 0,]

# Compare mean & sd of each sample

mean(sample\_1$Stars, na.rm = TRUE)

> mean(sample\_1$Stars, na.rm = TRUE)

[1] 3.807

sd(sample\_1$Stars, na.rm = TRUE)

> sd(sample\_1$Stars, na.rm = TRUE)

[1] 0.9384605

mean(sample\_2$Stars, na.rm = TRUE)

> mean(sample\_2$Stars, na.rm = TRUE)

[1] 3.573232

sd(sample\_2$Stars, na.rm = TRUE)

> sd(sample\_2$Stars, na.rm = TRUE)

[1] 1.137898

mean(sample\_3$Stars, na.rm = TRUE)

> mean(sample\_3$Stars, na.rm = TRUE)

[1] 3.70202

sd(sample\_3$Stars, na.rm = TRUE)

> sd(sample\_3$Stars, na.rm = TRUE)

[1] 0.8253917

mean(sample\_4$Stars, na.rm = TRUE)

> mean(sample\_4$Stars, na.rm = TRUE)

[1] 3.5205

sd(sample\_4$Stars, na.rm = TRUE)

> sd(sample\_4$Stars, na.rm = TRUE)

[1] 1.235667

### Part 7: Implementing features not mentioned in the specification

# create a subset with data of top\_ten rating 2012-2016

all\_top\_ten <- ramen\_data[(ramen\_data$Top.Ten != "" & ramen\_data$Top.Ten != "\n"),]

all\_top\_ten <- all\_top\_ten[order(all\_top\_ten$Top.Ten),]

# add year to the subset

yr <- rep(c(2012,2013,2014,2015,2016), c(9,7,8,7,6))

all\_top\_ten <- cbind(all\_top\_ten, yr)

# make the subset as a tibble

all\_top\_ten <- as\_tibble(all\_top\_ten)

> all\_top\_ten

# A tibble: 37 x 8

Review.. Brand Variety Style Country Stars Top.Ten yr

\* <int> <fct> <fct> <fct> <fct> <dbl> <fct> <dbl>

1 105 Indomie Special Fried Curly Noodle Pack Indonesia 5 2012 #1 2012

2 608 Koka Spicy Black Pepper Pack Singapore 5 2012 #10 2012

3 47 Indomie Mi Goreng Jumbo Barbecue Chicken Pack Indonesia 5 2012 #2 2012

4 392 Nissin Yakisoba Noodles Karashi Tray Japan 5 2012 #3 2012

5 13 Sapporo Ichiban Chow Mein Pack Japan 5 2012 #4 2012

6 434 Mi Sedaap Kari Spesial Pack Indonesia 4.5 2012 #5 2012

7 391 Myojo Hyoubanya No Chukasoba Oriental Pack Japan 4.25 2012 #6 2012

8 578 Nongshim Shin Ramyun Black Pack South Korea 4.75 2012 #7 2012

9 285 Doll Artificial Chicken Pack Hong Kong 4.5 2012 #9 2012

10 992 Prima Taste Singapore Laksa La Mian Pack Singapore 5 2013 #1 2013

# ... with 27 more rows

# 2012-2016 Top 10 ramen country of origin

yr\_country\_count <- all\_top\_ten %>% group\_by(yr,Country) %>% summarise(count = n())

> yr\_country\_count

# A tibble: 24 x 3

# Groups: yr [?]

yr Country count

<dbl> <fct> <int>

1 2012 Hong Kong 1

2 2012 Indonesia 3

3 2012 Japan 3

4 2012 Singapore 1

5 2012 South Korea 1

6 2013 Indonesia 1

7 2013 Japan 1

8 2013 Singapore 2

9 2013 South Korea 1

10 2013 Thailand 1

# ... with 14 more rows

sum\_2012 <- filter(yr\_country\_count, yr == 2012)

arrange(sum\_2012, desc(count))

> arrange(sum\_2012, desc(count))

# A tibble: 5 x 3

# Groups: yr [1]

yr Country count

<dbl> <fct> <int>

1 2012 Indonesia 3

2 2012 Japan 3

3 2012 Hong Kong 1

4 2012 Singapore 1

5 2012 South Korea 1

sum\_2013 <- filter(yr\_country\_count, yr == 2013)

arrange(sum\_2013, desc(count))

> arrange(sum\_2013, desc(count))

# A tibble: 6 x 3

# Groups: yr [1]

yr Country count

<dbl> <fct> <int>

1 2013 Singapore 2

2 2013 Indonesia 1

3 2013 Japan 1

4 2013 South Korea 1

5 2013 Thailand 1

6 2013 USA 1

sum\_2014 <- filter(yr\_country\_count, yr == 2014)

arrange(sum\_2014, desc(count))

> arrange(sum\_2014, desc(count))

# A tibble: 5 x 3

# Groups: yr [1]

yr Country count

<dbl> <fct> <int>

1 2014 South Korea 3

2 2014 Malaysia 2

3 2014 Japan 1

4 2014 Singapore 1

5 2014 Thailand 1

sum\_2015 <- filter(yr\_country\_count, yr == 2015)

arrange(sum\_2015, desc(count))

> arrange(sum\_2015, desc(count))

# A tibble: 4 x 3

# Groups: yr [1]

yr Country count

<dbl> <fct> <int>

1 2015 Malaysia 4

2 2015 Japan 1

3 2015 Taiwan 1

4 2015 Thailand 1

sum\_2016 <- filter(yr\_country\_count, yr == 2016)

arrange(sum\_2016, desc(count))

> arrange(sum\_2016, desc(count))

# A tibble: 4 x 3

# Groups: yr [1]

yr Country count

<dbl> <fct> <int>

1 2016 Singapore 3

2 2016 China 1

3 2016 Myanmar 1

4 2016 Taiwan 1

# all country of origin

Country\_count <- all\_top\_ten %>% group\_by(Country) %>% summarise(count = n())

arrange(Country\_count, desc(count))

> arrange(Country\_count, desc(count))

# A tibble: 11 x 2

Country count

<fct> <int>

1 Singapore 7

2 Japan 6

3 Malaysia 6

4 South Korea 5

5 Indonesia 4

6 Thailand 3

7 Taiwan 2

8 China 1

9 Hong Kong 1

10 Myanmar 1

11 USA 1