Homework 03 Data Cleaning, Normal Distributions

Due by 11:59pm, Saturday, 2.10.24

S&DS 230/530/ENV 757

(1) More on List Manipulation *(18 points - 3 points each)*.

(1.1) Make an object called `myList` that contains the following elements (in order):

- The integers 1 through 10
- A matrix with the integers 1 through 25 that has five rows, filled by row
- A list that contains
  - The text "latte"
  - A vector with the text "taco" and "nan"
  - A vector with the integers 1 through 7

You should be able to make this object in a single line of code.

```
myList <- list(1:10, matrix(1:25, nrow = 5, byrow = TRUE),
list("latte", c("taco", "nan"), 1:7))
```

Use `[]`, `[[]]`, `[,]` notation to answer parts b) through f).

(1.2) Make an object called `ans1` that is the fourth row of the matrix contained in `myList`.

(1.3) Make an object called `ans2` that is the sum of the 5th column of the matrix contained in `myList`.

(1.4) Make an object called `ans3` that is the sum of EACH row of the matrix contained in `myList` *(use the `apply()` function or check out `rowSums()`)*.

(1.5) Make an object called `ans4` that is whichever single element of `myList` that you'd like to consume (yes, comedians, it has to be a food . . .).

(1.6) Make an object called `ans5` that is the third element of the third element of `myList` converted to characters.

Get the results of each of your objects you created above (i.e. get them to show up in your knitted file by typing their names or putting the code line that creates each object in parentheses).

```
#1.2
ans1 <- myList[[2]][4,]
ans1
```

```
## [1] 16 17 18 19 20
```

```
#1.3
ans2 <- sum(myList[[2]][,5])
ans2
```

```
## [1] 75
```

```
#1.4
ans3 <- rowSums(myList[[2]])
ans3
```

```
## [1]  15  40  65  90 115
```

```
#1.5
ans4 <- myList[[3]][[1]]
ans4
```

```
## [1] "latte"
```

```
#1.6
ans5 <- as.character(myList[[3]][[3]])
ans5
```

```
## [1] "1" "2" "3" "4" "5" "6" "7"
```

(2) Normal Quantile Plots and the Binomial Distribution *(20 points, 3 points each, part (2.5) is 5 points).*

You may recall from your Intro Statistics course that a binomial distribution looks like a normal distribution if np > 10 and n(1-p) > 10 (i.e. as long as the average number of successes and failures are both larger than 10). Recall that n is the number of trials, and p is the probability of success for each Bernoulli trial. *As an example, flip a coin 30 times, count the number of heads. n=30, p=.5, np = 15 > 10 and n(1-p) = 15 > 10, so the distribution should be approximately normal.*

You are going to make six normal quantile plots that simulate 100 random observations from binomial distributions with p = .2 and various values of n.

(2.1) Install the `car` package. This will allow you use the `qqPlot()` function. Load this package.

(2.2) Make a vector called `vec` that is powers of 10 for powers 0 through 5. The one caveat, is that you need to use the`**` operator which reads as ‘to the power of’ *(i.e. `2**3` is 8).*

(2.3) Use the `par()` function to set up your plot region to show 6 plots on a page. The par argument you want is `mfrow = c(2,3)` which sets your plot region to have 2 rows and 3 columns.

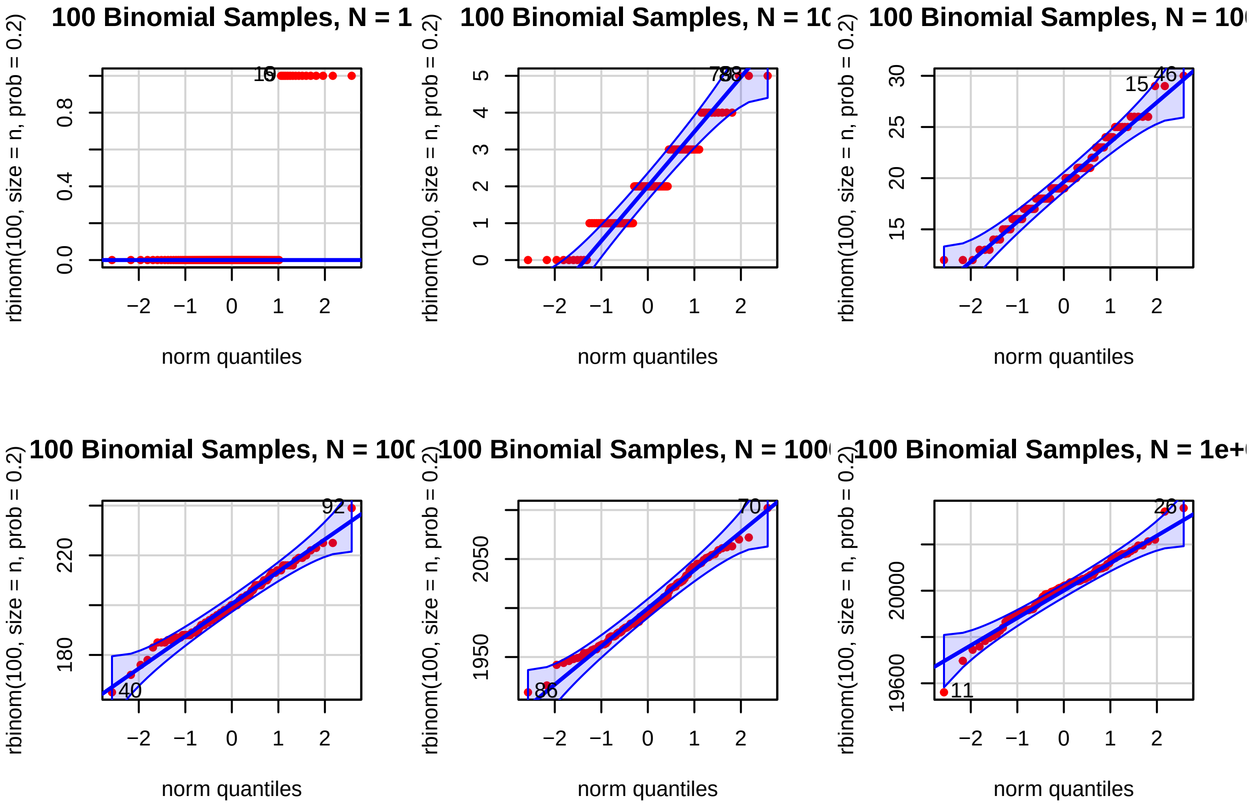
(2.4) Use the `rbinom()` function to generate 5 random binomial observations, each with 20 trials, and with p=0.8. You may need to type `?rbinom` to get the syntax for this function. Store the result in an object called `vec2`.

(2.5) Write a loop that repeatedly creates a normal quantile plot for 100 random samples each from a binomial distribution with p=0.2 and n equal to the 6 values stored in `vec`. A few plot details : * Use the `qqPlot` function. * Make the graph points red solid dots (`pch = 19`). * Make the boundary lines blue (use `col.lines`) * Make a main graph title that pastes the text “100 Binomial Samples, N =” to the corresponding value from `vec`.

```
#2.1
library(car)
```

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

```
#2.2
vec <- 10**(0:5)
#2.3
par(mfrow = c(2,3))
#2.4
vec2 <- rbinom(5, 20, 0.8)
#2.5
for (n in vec) {
  qqPlot(rbinom(100, size = n, prob = 0.2), col = "red",
  main = paste("100 Binomial Samples, N =", n),
  pch = 19, col.lines = "blue")
}
```



(2.6) Take a look at the normal quantile plots. For what value of n do the graphs seem to be approximately normally distributed? Is this consistent with what you expect? Write two complete sentences to answer these questions.

For N being 100 and above, the graphs seem to be normally distributed, as there is a relatively smooth and continuous distribution rather than one that looks like a staircase for the ones below. Given p = 0.2 and np, n(1-p) needing to be greater than 10, I would expect anything above n=50 to have to look like a normal distribution, which is the case here

(3) Favorite food and Data Cleaning *(62 points. Parts 3.2 through 3.5, 2 pts each, other values listed below).*

This is data generated by former students of S&DS 230. I simply asked "What is your favorite food?". You can get the data HERE.

Your goal is similar to what we did with the question "What animal would you like to be?" in Class 5 : clean this variable, make a barplot, and discuss the results.

(3.1) *(1 pnt)* Read in the data to a new object called `food`.

(3.2) Get a sense of the dataset - dimensions, variable names, look at the first few rows.

(3.3) Convert `food` to a single vector that is just the first column (literally, replace `food` with `food$Food`).

(3.4) Show the sorted unique values of `food`. Calculate how many unique values exist in `food`.

```
#3.1
food <- read.csv("Food_230.csv")
#3.2
dim(food)
```

```
## [1] 317   1
```

```r
names(food)
```

```
## [1] "Food"
```

```r
head(food)
```

```
##              Food
## 1  french fries.
## 2         Chicken
## 3 butter chicken
## 4          Ginger
## 5      Dal Bhaat
## 6      Thai food
```

```r
#3.3
food <- food$Food
#3.4
sort(unique(food))
```

```
##   [1] "Albanian Food "
##   [2] "all kinds of delicious food"
##   [3] "amaretto dark chocolate"
##   [4] "any type of cheese"
##   [5] "Arepa"
##   [6] "Artichoke"
##   [7] "Asian cuisine"
##   [8] "asian food"
##   [9] "Asian food"
##  [10] "Bagel"
##  [11] "baguettes"
##  [12] "Bananas"
##  [13] "Blue Point Oyster"
##  [14] "brazilian chorizo pizzas"
##  [15] "Brazilian food."
##  [16] "Bread"
##  [17] "Burgers"
##  [18] "burritos"
##  [19] "Burritos"
##  [20] "butter chicken"
##  [21] "Cajun Fries"
##  [22] "cake"
##  [23] "Ceviche"
##  [24] "cheese"
##  [25] "Cheese"
##  [26] "Cheeseburgers"
##  [27] "cheez its"
##  [28] "chicken"
##  [29] "Chicken"
##  [30] "Chicken Malai Kabab"
##  [31] "Chicken parmesan and penne alla vodka"
##  [32] "chicken tenders"
##  [33] "Chicken Tikka and Naan"
##  [34] "Chicken Tikka Masala"
##  [35] "Chicken wings"
## [36] "chinese"
## [37] "Chinese food"
## [38] "Chinese Food"
## [39] "Chinese food "
## [40] "Chinese food is my favorite."
## [41] "chinese hotpot"
## [42] "Chipotle"
## [43] "chocolate"
## [44] "Chocolate\n\n \n\nA cat"
## [45] "chocolate chip cookies"
## [46] "Cinnamon Rolls"
## [47] "comfort food"
## [48] "Cookies"
## [49] "Corn"
## [50] "Cottage Pie"
## [51] "Crepes"
## [52] "Curries of all sorts"
## [53] "Curry"
## [54] "Curry vindaloo"
## [55] "Dal Bhaat"
## [56] "Dandan noodles"
## [57] "Delicious "
## [58] "Dessert"
## [59] "Donuts!"
## [60] "Dried mangos"
## [61] "dumplings"
## [62] "Empanadas"
## [63] "enchiladas"
## [64] "Escargot in garlic butter"
## [65] "Ethiopian Cuisine"
## [66] "farofa"
## [67] "Farofa "
## [68] "Fish and chips"
## [69] "Fish tacos"
## [70] "Five guys' fries"
## [71] "Flautas"
## [72] "Freeze Dried"
## [73] "french fries"
## [74] "french fries."
## [75] "French onion soup"
## [76] "fried chicken"
## [77] "Fried chicken"
## [78] "Fried fish"
## [79] "Fried okra"
## [80] "Fried Rice"
## [81] "fruit"
## [82] "Fruit"
## [83] "Fruits"
## [84] "Ginger"
## [85] "Gizzard (chicken) and vegetables (greens), Igbo dish"
## [86] "gnocchi"
## [87] "Good pizza"
## [88] "Grilled chicken breast"
## [89] "guacamole"
##  [90] "guacamole "
##  [91] "Gyros"
##  [92] "ham"
##  [93] "Hamburgers"
##  [94] "Hibachi"
##  [95] "Hot dog"
##  [96] "hot pot"
##  [97] "Hot pot"
##  [98] "hotpot"
##  [99] "I love Asian food, especially Chinese food, Thai food, and Sushi. \n\n "
## [100] "I love Korean cuisine."
## [101] "i love mexican food"
## [102] "ice cream"
## [103] "Ice cream"
## [104] "Ice Cream"
## [105] "ICE CREAM"
## [106] "Ice cream!!!"
## [107] "Ice cream. "
## [108] "indian"
## [109] "Indian"
## [110] "Indian food"
## [111] "Indian Food"
## [112] "Indian Food "
## [113] "Indian food is great."
## [114] "Indonesian food"
## [115] "Instant ramen"
## [116] "italian"
## [117] "Italian"
## [118] "italian food"
## [119] "Italian food"
## [120] "Italian Food"
## [121] "italian food - pasta"
## [122] "Italian food!"
## [123] "Japanese "
## [124] "Japanese food"
## [125] "Jollof Rice and Chicken"
## [126] "Jollof Rice with chicken "
## [127] "Kelewele, it's an African dish"
## [128] "Korean"
## [129] "Korean Barbeque"
## [130] "Korean BBQ"
## [131] "korean bbq!"
## [132] "korean bbq!!"
## [133] "korean food"
## [134] "Korean food"
## [135] "Korean Food"
## [136] "korean food "
## [137] "kosher Steak "
## [138] "lahmacun (turkish pizza)"
## [139] "Lasagna"
## [140] "Lasagna."
## [141] "Lasagne"
## [142] "Lebanese Food"
## [143] "mac and cheese"
## [144] "Macaroni and Cheese"
## [145] "mango"
## [146] "Meat"
## [147] "mediterranean "
## [148] "Mediterranean "
## [149] "Mediterranean food "
## [150] "Mexican"
## [151] "Mexican food"
## [152] "Mexican Food"
## [153] "Multigrain pancake"
## [154] "My favorite kind of food is pastries."
## [155] "nectarines"
## [156] "noodles"
## [157] "Noodles"
## [158] "noodles with broth"
## [159] "Noodles."
## [160] "palak paneer"
## [161] "pasta"
## [162] "Pasta"
## [163] "Patty Melt"
## [164] "Persian and Mediterranean"
## [165] "Pho"
## [166] "pizza"
## [167] "Pizza"
## [168] "Pizza!!!"
## [169] "platanos"
## [170] "Poke Bowl"
## [171] "Postickers"
## [172] "probably either casual diner fare, Italian, or shellfish"
## [173] "Puerto Rican food"
## [174] "ramen"
## [175] "Ramen"
## [176] "ramen\n\n "
## [177] "Ribs"
## [178] "rice with curry"
## [179] "Roast dinner "
## [180] "salmon"
## [181] "Salmon"
## [182] "salty"
## [183] "seafood"
## [184] "shahi paneer"
## [185] "sharp cheddar cheese"
## [186] "Shrimp curry"
## [187] "soup"
## [188] "Soup"
## [189] "South Asian Rice Dishes for example Biryani, Pulao, Mandi, Tahri "
## [190] "Spaghetti "
## [191] "Spicy"
## [192] "Spicy and flavorful food"
## [193] "spicy food "
## [194] "Spicy Hotpot"
## [195] "spicy tofu"
## [196] "steak"
## [197] "Steak"
## [198] "strawberries"
## [199] "Strawberries  "
## [200] "sundried tomatoes"
## [201] "sushi"
## [202] "Sushi"
## [203] "Sushi with an overwhelming amount of raw salmon"
## [204] "Swedish meatballs"
## [205] "Sweet chili Doritos"
## [206] "sweet potato"
## [207] "Tagine"
## [208] "Thai"
## [209] "thai food"
## [210] "Thai food"
## [211] "Thai Food"
## [212] "Thai food "
## [213] "Thai food is my favorite kind of food "
## [214] "Thai specifically beef pad see-ew and some rice."
## [215] "Tofu"
## [216] "udon noodle"
## [217] "Vietnamese food"
## [218] "Vietnamese spring rolls"
```

```
length(unique(food))
```

```
## [1] 218
```

(3.5) Write a couple of sentences about what data cleaning issues you notice amoung the unique values of `food`.

There are some repeats with different capitalizations and some foods are have different formats, such as being plural or having a modifier. There are also some foods that are not actually foods, such as "delicious" and "anything with cheese". Some of the foods also contain multiple food items or are cuisines and not foods

(3.6) Cleaning Part I *(8 pts)*: Clean the data using the following steps (in order): Before proceeding to data cleaning, a quick reminder example of how to remove text before or after a particular word using the regular expression `.*` Note that `.` stands for 'any character' (other than a new line), and `*` stands for '0 or more times'.

Example : Find " have ", delete this AND everything preceeding or following:

```
#Deletes " have " and everything preceeding
gsub(".* have ", "", "Cats have personality")

#Deletes " have" and everything following
gsub(" have .*", "", "Cats have personality")
```

(3.6) Cleaning Part I *(8 pts)*: Clean the data using the following steps (in order):

- Convert data to lower case
- Find " or " and remove this and anything that follows.
- Find " and " and remove this and anything that follows.
- Find " food" and remove this AND anything that follows.
- Find " cuisine" and remove this AND anything that follows.
- Remove all special characters and punctuation.
- Remove trailing spaces at the end of text (use the `trimws()` function)

At each step, you'll probably want to check what unique values of `food` are left to make sure your functions are working correctly. By the time you finish, you should have 156 unique levels.

Your final two lines of code should again show the sorted unique values of food and the current number of unique values.

```r
#3.6
food <- tolower(food)
food <- gsub(" or .*", "", food)
food <- gsub(" and .*", "", food)
food <- gsub(" food.*", "", food)
food <- gsub(" cuisine.*", "", food)
food <- gsub("[^0-9A-Za-z///' ]", "", food)
food <- trimws(food)
sort(unique(food))
```

```
##   [1] "albanian"
##   [2] "all kinds of delicious"
##   [3] "amaretto dark chocolate"
##   [4] "any type of cheese"
##   [5] "arepa"
##   [6] "artichoke"
##   [7] "asian"
##   [8] "bagel"
##   [9] "baguettes"
##  [10] "bananas"
##  [11] "blue point oyster"
##  [12] "brazilian"
##  [13] "brazilian chorizo pizzas"
##  [14] "bread"
##  [15] "burgers"
##  [16] "burritos"
##  [17] "butter chicken"
##  [18] "cajun fries"
##  [19] "cake"
##  [20] "ceviche"
##  [21] "cheese"
##  [22] "cheeseburgers"
##  [23] "cheez its"
##  [24] "chicken"
##  [25] "chicken malai kabab"
##  [26] "chicken parmesan"
##  [27] "chicken tenders"
##  [28] "chicken tikka"
##  [29] "chicken tikka masala"
##  [30] "chicken wings"
##  [31] "chinese"
##  [32] "chinese hotpot"
##  [33] "chipotle"
##  [34] "chocolate"
##  [35] "chocolate chip cookies"
##  [36] "chocolatea cat"
##  [37] "cinnamon rolls"
##  [38] "comfort"
##  [39] "cookies"
##  [40] "corn"
##  [41] "cottage pie"
##  [42] "crepes"
##  [43] "curries of all sorts"
##  [44] "curry"
##  [45] "curry vindaloo"
##  [46] "dal bhaat"
##  [47] "dandan noodles"
##  [48] "delicious"
##  [49] "dessert"
##  [50] "donuts"
##  [51] "dried mangos"
##  [52] "dumplings"
##  [53] "empanadas"
##  [54] "enchiladas"
##  [55] "escargot in garlic butter"
##  [56] "ethiopian"
##  [57] "farofa"
##  [58] "fish"
##  [59] "fish tacos"
##  [60] "five guys' fries"
##  [61] "flautas"
##  [62] "freeze dried"
##  [63] "french fries"
##  [64] "french onion soup"
##  [65] "fried chicken"
##  [66] "fried fish"
##  [67] "fried okra"
##  [68] "fried rice"
##  [69] "fruit"
##  [70] "fruits"
##  [71] "ginger"
##  [72] "gizzard chicken"
##  [73] "gnocchi"
##  [74] "good pizza"
##  [75] "grilled chicken breast"
##  [76] "guacamole"
##  [77] "gyros"
##  [78] "ham"
##  [79] "hamburgers"
##  [80] "hibachi"
##  [81] "hot dog"
##  [82] "hot pot"
##  [83] "hotpot"
##  [84] "i love asian"
##  [85] "i love korean"
##  [86] "i love mexican"
##  [87] "ice cream"
##  [88] "indian"
##  [89] "indonesian"
##  [90] "instant ramen"
##  [91] "italian"
##  [92] "japanese"
##  [93] "jollof rice"
##  [94] "jollof rice with chicken"
##  [95] "kelewele it's an african dish"
##  [96] "korean"
##  [97] "korean barbeque"
##  [98] "korean bbq"
##  [99] "kosher steak"
## [100] "lahmacun turkish pizza"
## [101] "lasagna"
## [102] "lasagne"
## [103] "lebanese"
## [104] "mac"
## [105] "macaroni"
## [106] "mango"
## [107] "meat"
## [108] "mediterranean"
## [109] "mexican"
## [110] "multigrain pancake"
## [111] "my favorite kind of"
## [112] "nectarines"
## [113] "noodles"
## [114] "noodles with broth"
## [115] "palak paneer"
## [116] "pasta"
## [117] "patty melt"
## [118] "persian"
## [119] "pho"
## [120] "pizza"
## [121] "platanos"
## [122] "poke bowl"
## [123] "postickers"
## [124] "probably either casual diner fare italian"
## [125] "puerto rican"
## [126] "ramen"
## [127] "ribs"
## [128] "rice with curry"
## [129] "roast dinner"
## [130] "salmon"
## [131] "salty"
## [132] "seafood"
## [133] "shahi paneer"
## [134] "sharp cheddar cheese"
## [135] "shrimp curry"
## [136] "soup"
## [137] "south asian rice dishes for example biryani pulao mandi tahri"
## [138] "spaghetti"
## [139] "spicy"
## [140] "spicy hotpot"
## [141] "spicy tofu"
## [142] "steak"
## [143] "strawberries"
## [144] "sundried tomatoes"
## [145] "sushi"
## [146] "sushi with an overwhelming amount of raw salmon"
## [147] "swedish meatballs"
## [148] "sweet chili doritos"
## [149] "sweet potato"
## [150] "tagine"
## [151] "thai"
## [152] "thai specifically beef pad seeew"
## [153] "tofu"
## [154] "udon noodle"
## [155] "vietnamese"
## [156] "vietnamese spring rolls"
```

```r
length(unique(food))
```

```
## [1] 156
```

(3.7) Cleaning Part II *(10 pts)*: A few quick random cleaning items:

Clean up the following types of food (in order) - one line of code per type of food. In each case, deal with misspellings, modifiers ("shrimp curry" vs just "curry"), two words ('hot pot' instead of 'hotpot'), plurals, etc.

- hotpot
- curry
- lasagna
- noodles
- cookies
- chocolate
- cheese
- steak
- sushi
- fries (french, cajun, five guys' all just call 'fries')
- ramen
- tofu
- burgers (of any kind)
- soup
- anything containing 'delicious' just call 'delicious'

When you're finished, you should have 130 unique values.

Your final two lines of code should again show the sorted unique values of food and the current number of unique values.

```r
#3.7
food <- gsub(".*(hot pot|hotpot).*", "hotpot", food)
food <- gsub(".*(curry|curries).*", "curry", food)
food <- gsub(".*(lasagna|lasagnas|lasagne).*", "lasagna", food)
food <- gsub(".*(noodles|noodle).*", "noodles", food)
food <- gsub(".*(cookies|cookie).*", "cookies", food)
food <- gsub(".*(chocolate|chocolates).*", "chocolate", food)
food <- gsub(".*cheese.*", "cheese", food)
food <- gsub(".*steak.*", "steak", food)
food <- gsub(".*sushi.*", "sushi", food)
food <- gsub(".*fries.*", "fries", food)
food <- gsub(".*ramen.*", "ramen", food)
food <- gsub(".*tofu.*", "tofu", food)
food <- gsub(".*(burgers|burger).*", "burgers", food)
food <- gsub(".*soup.*", "soup", food)
food <- gsub(".*delicious.*", "delicious", food)
sort(unique(food))
```

```
##   [1] "albanian"
##   [2] "arepa"
##   [3] "artichoke"
##   [4] "asian"
##   [5] "bagel"
##   [6] "baguettes"
##   [7] "bananas"
##   [8] "blue point oyster"
##   [9] "brazilian"
##  [10] "brazilian chorizo pizzas"
##  [11] "bread"
##  [12] "burgers"
##  [13] "burritos"
##  [14] "butter chicken"
##  [15] "cake"
##  [16] "ceviche"
##  [17] "cheese"
##  [18] "cheez its"
##  [19] "chicken"
##  [20] "chicken malai kabab"
##  [21] "chicken parmesan"
##  [22] "chicken tenders"
##  [23] "chicken tikka"
##  [24] "chicken tikka masala"
##  [25] "chicken wings"
##  [26] "chinese"
##  [27] "chipotle"
##  [28] "chocolate"
##  [29] "cinnamon rolls"
##  [30] "comfort"
##  [31] "cookies"
##  [32] "corn"
##  [33] "cottage pie"
##  [34] "crepes"
##  [35] "curry"
##  [36] "dal bhaat"
##  [37] "delicious"
##  [38] "dessert"
##  [39] "donuts"
##  [40] "dried mangos"
##  [41] "dumplings"
##  [42] "empanadas"
##  [43] "enchiladas"
##  [44] "escargot in garlic butter"
##  [45] "ethiopian"
##  [46] "farofa"
##  [47] "fish"
##  [48] "fish tacos"
##  [49] "flautas"
##  [50] "freeze dried"
##  [51] "fried chicken"
##  [52] "fried fish"
##  [53] "fried okra"
##  [54] "fried rice"
##  [55] "fries"
##  [56] "fruit"
##  [57] "fruits"
##  [58] "ginger"
##  [59] "gizzard chicken"
##  [60] "gnocchi"
##  [61] "good pizza"
##  [62] "grilled chicken breast"
##  [63] "guacamole"
##  [64] "gyros"
##  [65] "ham"
##  [66] "hibachi"
##  [67] "hot dog"
##  [68] "hotpot"
##  [69] "i love asian"
##  [70] "i love korean"
##  [71] "i love mexican"
##  [72] "ice cream"
##  [73] "indian"
##  [74] "indonesian"
##  [75] "italian"
##  [76] "japanese"
##  [77] "jollof rice"
##  [78] "jollof rice with chicken"
##  [79] "kelewele it's an african dish"
##  [80] "korean"
##  [81] "korean barbeque"
##  [82] "korean bbq"
##  [83] "lahmacun turkish pizza"
##  [84] "lasagna"
##  [85] "lebanese"
##  [86] "mac"
##  [87] "macaroni"
##  [88] "mango"
##  [89] "meat"
##  [90] "mediterranean"
##  [91] "mexican"
##  [92] "multigrain pancake"
##  [93] "my favorite kind of"
##  [94] "nectarines"
##  [95] "noodles"
##  [96] "palak paneer"
##  [97] "pasta"
##  [98] "patty melt"
##  [99] "persian"
## [100] "pho"
## [101] "pizza"
## [102] "platanos"
## [103] "poke bowl"
## [104] "postickers"
## [105] "probably either casual diner fare italian"
## [106] "puerto rican"
## [107] "ramen"
## [108] "ribs"
## [109] "roast dinner"
## [110] "salmon"
## [111] "salty"
## [112] "seafood"
## [113] "shahi paneer"
## [114] "soup"
## [115] "south asian rice dishes for example biryani pulao mandi tahri"
## [116] "spaghetti"
## [117] "spicy"
## [118] "steak"
## [119] "strawberries"
## [120] "sundried tomatoes"
## [121] "sushi"
## [122] "swedish meatballs"
## [123] "sweet chili doritos"
## [124] "sweet potato"
## [125] "tagine"
## [126] "thai"
## [127] "thai specifically beef pad seeew"
## [128] "tofu"
## [129] "vietnamese"
## [130] "vietnamese spring rolls"
```

```
length(unique(food))
```

```
## [1] 130
```

(3.8) Cleaning Part III *(8 pts)*: Cleaning types of cuisine.

Clean up the following types of cuisine (in order) - in this case, you'll want to make a vector called `searchvec` that contains the types of cuisine. Then create a loop following the example in Class 5 to replace all the modifiers for each cuisine type so that you ultimately end up with cleaned up versions of each cuisine type. Use not more than 5 lines of code.

The cuisine types (in order) are * asian * chinese * vietnamese * italian * indian * thai * mexican * brazilian * korean

(there are other types of cuisine, but they don't require cleaning).

When you're finished, you should have 120 unique values.

Your final two lines of code should again show the sorted unique values of food and the current number of unique values.

```
#3.8
searchvec <- c("asian", "chinese", "vietnamese", "italian",
"indian", "thai", "mexican", "brazilian", "korean")
for (i in searchvec) {
  food <- gsub(paste(".*", i, ".*", sep = ""), i, food)
}
sort(unique(food))
```

```
##   [1] "albanian"                              "arepa"
##   [3] "artichoke"                             "asian"
##   [5] "bagel"                                 "baguettes"
##   [7] "bananas"                               "blue point oyster"
##   [9] "brazilian"                             "bread"
##  [11] "burgers"                               "burritos"
##  [13] "butter chicken"                        "cake"
##  [15] "ceviche"                               "cheese"
##  [17] "cheez its"                     "chicken"
##  [19] "chicken malai kabab"           "chicken parmesan"
##  [21] "chicken tenders"               "chicken tikka"
##  [23] "chicken tikka masala"          "chicken wings"
##  [25] "chinese"                       "chipotle"
##  [27] "chocolate"                     "cinnamon rolls"
##  [29] "comfort"                       "cookies"
##  [31] "corn"                          "cottage pie"
##  [33] "crepes"                        "curry"
##  [35] "dal bhaat"                     "delicious"
##  [37] "dessert"                       "donuts"
##  [39] "dried mangos"                  "dumplings"
##  [41] "empanadas"                     "enchiladas"
##  [43] "escargot in garlic butter"     "ethiopian"
##  [45] "farofa"                        "fish"
##  [47] "fish tacos"                    "flautas"
##  [49] "freeze dried"                  "fried chicken"
##  [51] "fried fish"                    "fried okra"
##  [53] "fried rice"                    "fries"
##  [55] "fruit"                         "fruits"
##  [57] "ginger"                        "gizzard chicken"
##  [59] "gnocchi"                       "good pizza"
##  [61] "grilled chicken breast"        "guacamole"
##  [63] "gyros"                         "ham"
##  [65] "hibachi"                       "hot dog"
##  [67] "hotpot"                        "ice cream"
##  [69] "indian"                        "indonesian"
##  [71] "italian"                       "japanese"
##  [73] "jollof rice"                   "jollof rice with chicken"
##  [75] "kelewele it's an african dish" "korean"
##  [77] "lahmacun turkish pizza"        "lasagna"
##  [79] "lebanese"                      "mac"
##  [81] "macaroni"                      "mango"
##  [83] "meat"                          "mediterranean"
##  [85] "mexican"                       "multigrain pancake"
##  [87] "my favorite kind of"           "nectarines"
##  [89] "noodles"                       "palak paneer"
##  [91] "pasta"                         "patty melt"
##  [93] "persian"                       "pho"
##  [95] "pizza"                         "platanos"
##  [97] "poke bowl"                     "postickers"
##  [99] "puerto rican"                  "ramen"
## [101] "ribs"                          "roast dinner"
## [103] "salmon"                        "salty"
## [105] "seafood"                       "shahi paneer"
## [107] "soup"                          "spaghetti"
## [109] "spicy"                         "steak"
## [111] "strawberries"                  "sundried tomatoes"
## [113] "sushi"                         "swedish meatballs"
## [115] "sweet chili doritos"           "sweet potato"
## [117] "tagine"                        "thai"
## [119] "tofu"                          "vietnamese"
```

```
length(unique(food))
```

```
## [1] 120
```

(3.9) *(15 pts)* Following the example from Class 05, display a dataframe of the sorted tabular results of `food` to see how many individuals prefer each kind of food.

From here on, the decisions of how to clean and combine categories are yours! Any food that currently has a count of 3 or more should remain (you can add to these categories - for example, you could add 'lasagna' to 'italian' or to 'pasta'). All other levels should be recoded or incorporated into a 'miscellaneous' food category. Points awarded based on thoughtfulness, effort, and quality/preciseness of your code.

Include your code below, and add comments where appropriate to describe the choices you make. You should have no more than 40 levels by the time you finish.

Display a dataframe of the sorted tabular results of `food` to see how many individuals prefer each kind of food AGAIN after you've finished your coding.

```
#3.9
foodframe <- data.frame(sort(table(food), decreasing = T))
foodframe
```

```
##                               food Freq
## 1                          sushi   24
## 2                          pizza   17
## 3                         korean   14
## 4                           thai   14
## 5                        chinese   13
## 6                      ice cream   10
## 7                        mexican   10
## 8                         indian    9
## 9                        italian    9
## 10                         ramen    9
## 11                       noodles    8
## 12                         pasta    8
## 13                         steak    8
## 14                     chocolate    6
## 15                        hotpot    6
## 16                         asian    5
## 17                        cheese    5
## 18                         curry    5
## 19                          soup    5
## 20                         fries    4
## 21                       burgers    3
## 22                         fruit    3
## 23                      japanese    3
## 24                       lasagna    3
## 25                 mediterranean    3
## 26                           pho    3
## 27                         spicy    3
## 28                     brazilian    2
## 29                         bread    2
## 30                      burritos    2
## 31                       chicken    2
## 32                 chicken wings    2
## 33                       cookies    2
## 34                     delicious    2
## 35                         farofa    2
## 36                  fried chicken    2
## 37                      guacamole    2
## 38                         salmon    2
## 39                   strawberries    2
## 40                           tofu    2
## 41                     vietnamese    2
## 42                       albanian    1
## 43                          arepa    1
## 44                      artichoke    1
## 45                          bagel    1
## 46                      baguettes    1
## 47                        bananas    1
## 48              blue point oyster    1
## 49                 butter chicken    1
## 50                           cake    1
## 51                        ceviche    1
## 52                      cheez its    1
## 53            chicken malai kabab    1
## 54               chicken parmesan    1
## 55                chicken tenders    1
## 56                  chicken tikka    1
## 57           chicken tikka masala    1
## 58                       chipotle    1
## 59                 cinnamon rolls    1
## 60                        comfort    1
## 61                           corn    1
## 62                    cottage pie    1
## 63                         crepes    1
## 64                      dal bhaat    1
## 65                        dessert    1
## 66                         donuts    1
## 67                   dried mangos    1
## 68                      dumplings    1
## 69                      empanadas    1
## 70                     enchiladas    1
## 71      escargot in garlic butter    1
## 72                      ethiopian    1
## 73                           fish    1
## 74                     fish tacos    1
## 75                        flautas    1
## 76                   freeze dried    1
## 77                     fried fish    1
## 78                     fried okra    1
## 79                     fried rice    1
## 80                         fruits    1
## 81                         ginger    1
## 82                gizzard chicken    1
## 83                        gnocchi    1
## 84                     good pizza    1
## 85         grilled chicken breast    1
## 86                          gyros    1
## 87                            ham    1
## 88                        hibachi    1
## 89                        hot dog    1
## 90                     indonesian    1
## 91                    jollof rice    1
## 92       jollof rice with chicken    1
## 93  kelewele it's an african dish    1
## 94         lahmacun turkish pizza    1
## 95                       lebanese    1
## 96                            mac    1
## 97                       macaroni    1
## 98                          mango    1
## 99                           meat    1
## 100            multigrain pancake    1
## 101           my favorite kind of    1
## 102                    nectarines    1
## 103                  palak paneer    1
## 104                    patty melt    1
## 105                       persian    1
## 106                      platanos    1
## 107                     poke bowl    1
## 108                    postickers    1
## 109                  puerto rican    1
## 110                          ribs    1
## 111                  roast dinner    1
## 112                         salty    1
## 113                       seafood    1
## 114                  shahi paneer    1
## 115                     spaghetti    1
## 116             sundried tomatoes    1
## 117             swedish meatballs    1
## 118           sweet chili doritos    1
## 119                  sweet potato    1
## 120                        tagine    1
```

```
food_counts <- table(food)
popular_food <- names(food_counts[food_counts >= 3])
#sorting the chicken
food <- gsub(".*chicken.*", "chicken", food)
#grouping fried food
food <- gsub(".*fried.*", "fried food", food)
#sorting asian food
food <- gsub(".*(paneer|rice|postickers|hibachi|indonesian|turkish|vietnamese|
dumplings|dal bhaat|tofu|persian|korean|japanese|indian).*", "asian", food)
#sorting african food
food <- gsub(".*(rice|african|ethiopian|tagine).*", "african", food)
#sorting american food
food <- gsub(".*(bbq|american|mac|hot dogs|chili|cornbread|pulled pork).*",
"american", food)
#sorting european food
food <- gsub(".*(albanian|baguettes|cottage|crepes|lebanese|escargot|spaghetti|
swedish).*", "european", food)
#sorting south american food
food <- gsub(".*(empanadas|guacamole|taco|farofa|flautas|
arepa|brazil|ceviche|rico|mexi).*", "south american", food)
#sorting seafood
```

```
food <- gsub(".*(salmon|oyster|fish|seafood).*", "seafood", food)
#sorting veggies
food <- gsub(".*(tomato|spotato|ginger|artichoke|corn).*", "veggies", food)
#sorting fruit
food <- gsub(".*(fruit|strawberries|bananas|mango|nectarines).*", "fruit", food)
#sorting dessert
food <- gsub(".*(cookie|cake|cinnamon|creepe|donut|pancake|chocolate).*",
"dessert", food)
#sorting italian
food <- gsub("gnocchi|spaghetti|lasagna|macaroni", "italian", food)
#sorting american foods
food <- gsub(".*(fried|steak|chipotle|dog|doritos|bowl|ribs|roast|patty|ham).*",
 "american", food)
#everything not sorted is miscellaneous
categories <- c("chicken", "fried food", "asian", "african", "american",
 "european", "south american", "seafood",
 "veggies", "fruit", "dessert", "italian")

#setting everything that is not popular food or a category to miscellaneous
miscellaneous_food <- setdiff(unique(food), c(popular_food, categories))
food[food %in% miscellaneous_food] <- "miscellaneous"
foodframe <- data.frame(sort(table(food), decreasing = T))
foodframe
```

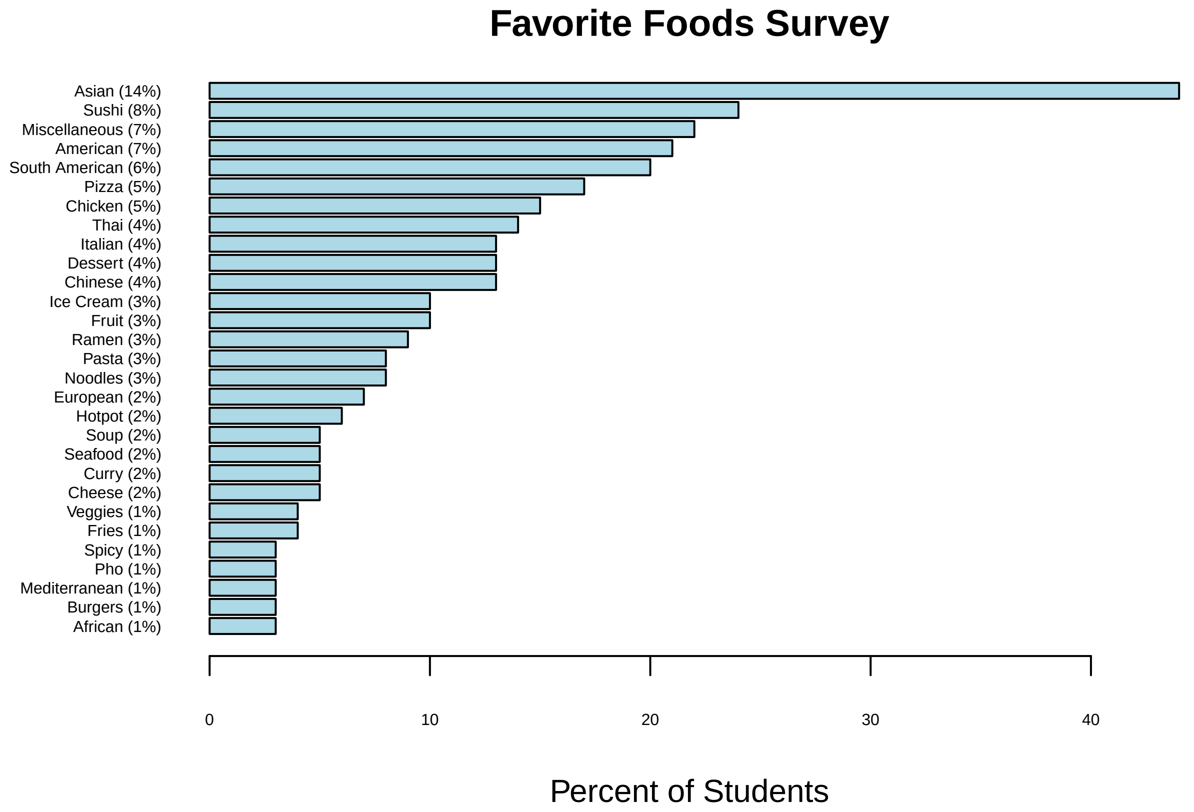
```
##              food Freq
## 1           asian   44
## 2           sushi   24
## 3   miscellaneous   22
## 4        american   21
## 5  south american   20
## 6           pizza   17
## 7         chicken   15
## 8            thai   14
## 9         chinese   13
## 10        dessert   13
## 11        italian   13
## 12          fruit   10
## 13      ice cream   10
## 14          ramen    9
## 15        noodles    8
## 16          pasta    8
## 17       european    7
## 18         hotpot    6
## 19         cheese    5
## 20          curry    5
## 21        seafood    5
## 22           soup    5
## 23          fries    4
## 24        veggies    4
## 25        african    3
## 26        burgers    3
## 27  mediterranean    3
## 28            pho    3
## 29          spicy    3
```

(3.10) *(8 pts)* Final steps and a plot: You'll want to CAREFULLY follow the example in the code at the end of Class 05.

- Use the `toTitleCase()` function from the package `tools` to convert food to title case.
- Make an object called `finaltab` that is a table of your final vector `food`.
- Calculate percents, rounded to the nearest integer, for each food type. Save this as an object called `percents`.
- Change the names of `finaltab` to include a space and then the percents followed by a "%" in curved parentheses.
- Make a horizontal barplot of your final plot. Choose a nice bar color, adjust the left margins as necessary, give a main title and label the horizontal axis.

```
#3.10
library(tools)
food <- toTitleCase(food)
finaltab <- table(food)
percents <- round(finaltab/sum(finaltab)*100)
names(finaltab) <- paste(names(finaltab), " (", percents, "%)", sep = "")
par(mar = c(5, 6, 2, 1))
par(cex.axis = .5)

barplot(sort(finaltab), horiz = T, col = "lightblue",
las = 1, main = "Favorite Foods Survey", xlab = "Percent of Students")
```



(3.11) *(3 pts)* In no more than three sentences, discuss your process and results. Be sure to mention how many unique values of 'food' you started and ended with. Any surprises?

I started with 218 unique values, and eventually came down to 29. The process started with removing duplicates *by capitalizations and using some regular expressions for common foods. From there, I group some foods into the groups given and then based on their geographical cuisine group and other types. I was suprised at the amount of data and different modifiers I was able to filter out, and also by the amount of people who prefered asian food, specifically sushi.*