Data Analysis with R

UCLA Stats Club / DataFest Planning Committee

2/6/2019

Installing and loading the Tidyverse

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

Applying the Tidyverse: the 2017 grades dataset

```
grades_2017 <- read_csv("grades_2017.csv")
names(grades_2017)
```

```
##  [1] "name"          "number"        "title"         "full_name"    
##  [5] "term_quarter"  "term_year"     "grade_Ap"      "grade_A"      
##  [9] "grade_Am"      "grade_Bp"      "grade_B"       "grade_Bm"     
## [13] "grade_Cp"      "grade_C"       "grade_Cm"      "grade_Dp"     
## [17] "grade_D"       "grade_Dm"      "grade_F"       "grade_no_pass"
## [21] "grade_pass"
```

We can use dplyr's 'mutate' to create a variable, 'total_students':

```
grades_2017 <- grades_2017 %>%
  mutate(total_students = rowSums(grades_2017[, 7:21]))
```

Creating a variable using separate()

A good way to begin working with a function, to make sure you're using it right, is to consult its documentation. What's great about RStudio is that it allows you to do so within itself. Type '?separate' into the console to see its documentation. This will help you understand why we frame the ***parameters*** of the function the way we do.

```
grades_2017 <- grades_2017 %>%
  separate(name, c("subject_area", NA),
           sep = "(M|C|CM|)[0-9]", remove = FALSE)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 3948 rows [1,
## 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
## 27, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 3 rows [8,
## 1503, 3016].
```

```
# the regex says "(M or C or CM or <nothing>) and [any digit (from 0-9, so any digit)]"
```

There may be a few classes numbered slightly differently, meaning the 'subject area' variable will be the same as the 'name' variable in their case (since no part of their name would have matched the regular expression). Let's check, using dplyr's 'filter' function:

```
grades_2017 %>% filter(subject_area == name)
```

```
## # A tibble: 3 x 23
##   name  subject_area number title full_name term_quarter term_year grade_Ap
##   <chr> <chr>        <chr>  <chr> <chr>     <chr>            <dbl>    <dbl>
## 1 AERO~ AERO ST A    A      Lead~ J.R. Lis~ FA                2017        0
## 2 AERO~ AERO ST A    A      Lead~ J.R. Lis~ SP                2017        0
## 3 AERO~ AERO ST A    A      Lead~ J.R. Lis~ WI                2017        0
## # ... with 15 more variables: grade_A <dbl>, grade_Am <dbl>,
## #   grade_Bp <dbl>, grade_B <dbl>, grade_Bm <dbl>, grade_Cp <dbl>,
## #   grade_C <dbl>, grade_Cm <dbl>, grade_Dp <dbl>, grade_D <dbl>,
## #   grade_Dm <dbl>, grade_F <dbl>, grade_no_pass <dbl>, grade_pass <dbl>,
## #   total_students <dbl>
```

```
# only one class fails to meet numbering conventions; let's take it out
grades_2017 <- grades_2017 %>% filter(subject_area != name)
# this command allows us to keep all the other rows.

grades_2017$subject_area <- trimws(grades_2017$subject_area)
# handy base function to trim white space
```

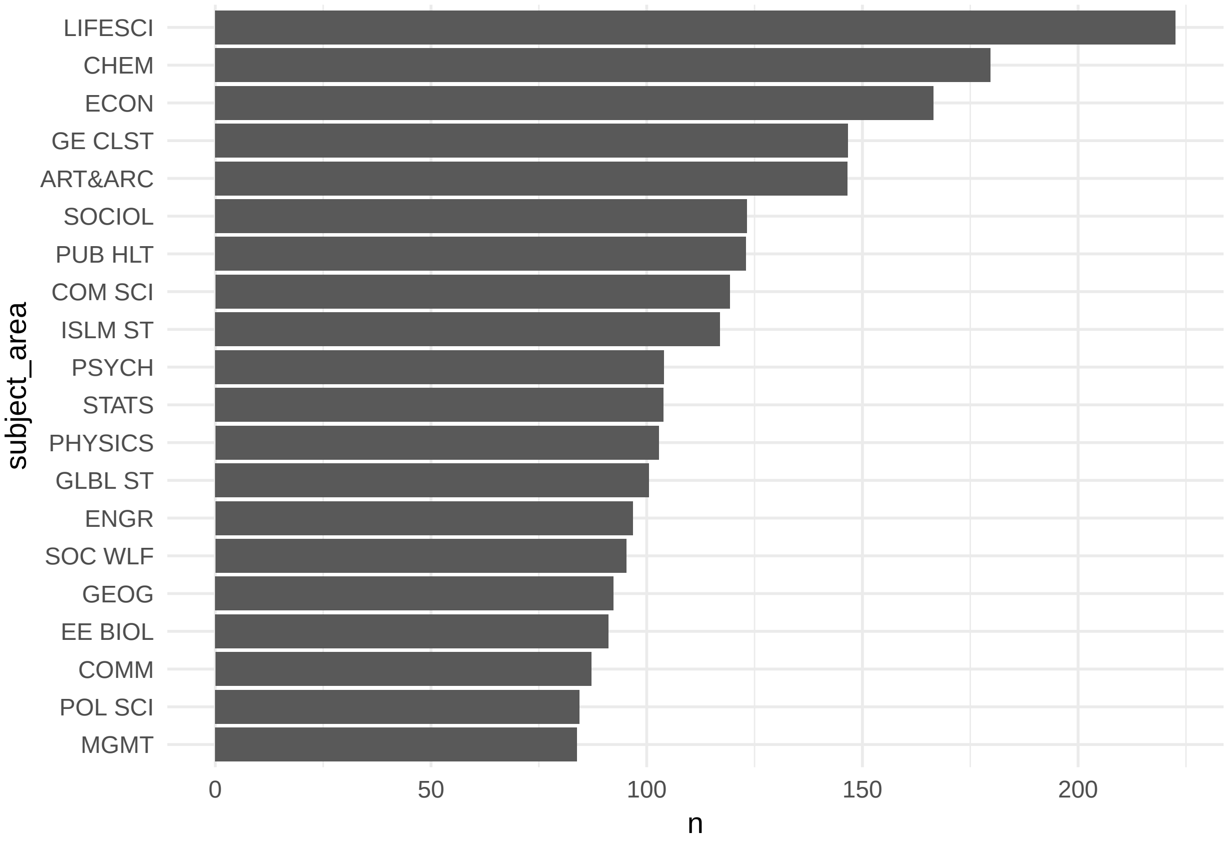
Using dplyr's group_by() and summarize()

```
largest_classes_subj <- grades_2017 %>%
  group_by(subject_area) %>%
  summarise(n = mean(total_students)) %>%
  # gives us the sum of class enrollments
  # grouped by subject area.
  # replacing 'sum' with 'mean' within summarise
  # gives mean enrollments by subject area
  arrange(desc(n)) %>%
  # arrange in descending order of n
  top_n(20)
  # select top 20 for further analysis

largest_classes_subj
```

```
## # A tibble: 20 x 2
##    subject_area     n
##    <chr>        <dbl>
##  1 LIFESCI       223.
##  2 CHEM          180.
##  3 ECON          167.
##  4 GE CLST       147.
##  5 ART&ARC       146.
##  6 SOCIOL        123.
##  7 PUB HLT       123
##  8 COM SCI       119.
##  9 ISLM ST       117
## 10 PSYCH         104.
## 11 STATS         104.
## 12 PHYSICS       103.
## 13 GLBL ST       100.
## 14 ENGR           96.8
## 15 SOC WLF        95.2
## 16 GEOG           92.3
## 17 EE BIOL        91.2
## 18 COMM           87.1
## 19 POL SCI        84.4
## 20 MGMT           83.9
```

Using ggplot2 to plot our findings:

```
largest_classes_subj %>%
  # DATASET
  ggplot(aes(x = subject_area, y = n)) +
  # ggplot() + AESTHETIC MAPPING (only axes in this case)
  geom_bar(stat = "identity") +
  # LAYER (barplot)
  # stat = 'identity' tells ggplot not to summarize data.
  scale_x_discrete(limits =
                     rev(largest_classes_subj$subject_area)) +
  # SCALES (orders plot from least to most)
  coord_flip() +
  # COORDINATE SYSTEMS (flipping coordinates)
  theme_minimal()
```

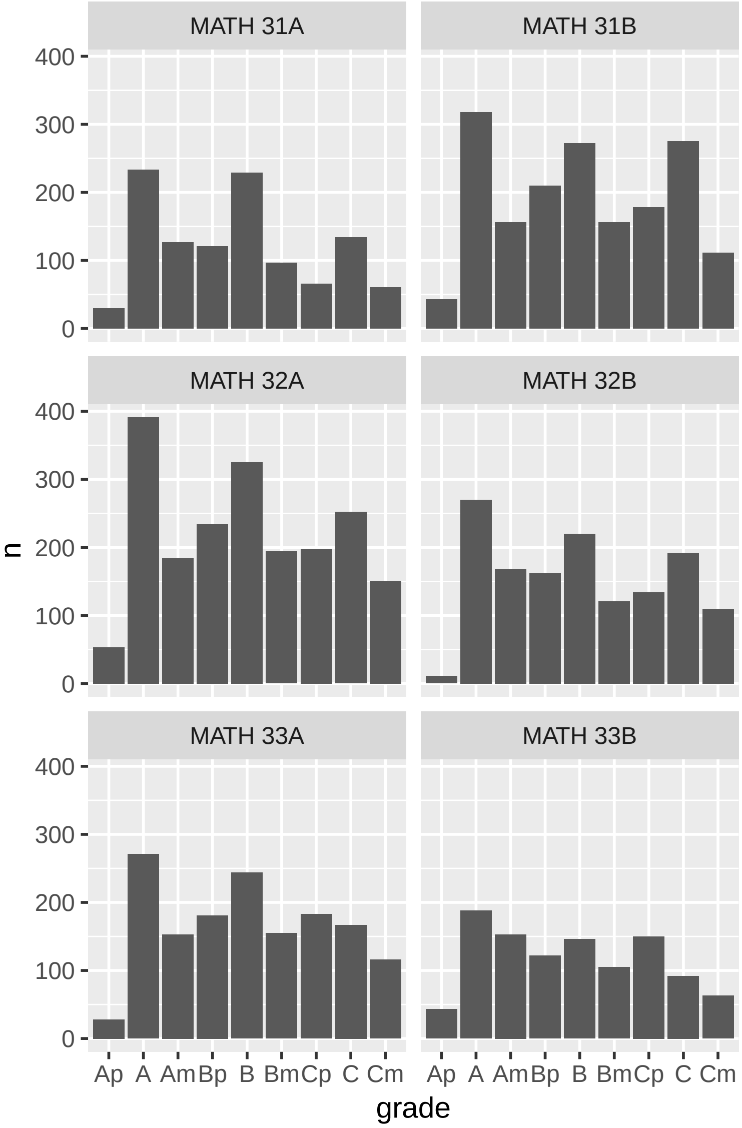


```
  # changing the plot's look
```

Grade distributions in Math lower-division classes:

```
math_lowerdivs <- which(str_detect(grades_2017$name, "MATH 3[123][AB]"))
# use str_detect() to detect the presence of the regex in a row
# wrap the which function around it to get the indices of those rows
# the regex says "MATH 3[either 1, 2 or 3][either A or B]"
math_lowerdivs <- grades_2017[math_lowerdivs, ]
# select only those rows
math_lowerdivs$name <- str_sub(math_lowerdivs$name, start = 1, end = 8)
# use str_sub() to obtain substrings to omit whether a class is honors
math_lowerdivs <- math_lowerdivs %>%
  gather("grade", "num_students",
         grade_Ap:grade_Cm)
# use tidyr's gather to make the wide data long
math_lowerdivs <- math_lowerdivs %>%
  group_by(name, grade) %>%
  summarise(n = sum(num_students))
# group by both name and grade to find
# the number of students per grade per class

math_lowerdivs$grade <- str_sub(math_lowerdivs$grade, start = 7)
# omit "grade_" from each grade level to just get A, Ap, Am, etc.
```

```
math_lowerdivs %>%
  ggplot(aes(x = grade, y = n)) +
  # ggplot() + AESTHETIC MAPPING (only axes)
  geom_bar(stat = "identity") +
  # LAYER (barplot)
  # stat = 'identity' tells ggplot not to summarize data.
  scale_x_discrete(limits =
    c("Ap", "A", "Am", "Bp", "B", "Bm", "Cp", "C", "Cm")) +
  # SCALES (orders plot as desired)
  facet_wrap(~name, nrow = 3)
```



```
  # FACET (splits plot based on a variable)
```

Some more ggplot2 with the cereal dataset:

But first, let's clean it up a bit.

```r
cereal <- read_csv("cereal.csv")
```

```
## Parsed with column specification:
## cols(
##   name = col_character(),
##   mfr = col_character(),
##   type = col_character(),
##   calories = col_double(),
##   protein = col_double(),
##   fat = col_double(),
##   sodium = col_double(),
##   fiber = col_double(),
##   carbo = col_double(),
##   sugars = col_double(),
##   potass = col_double(),
##   weight = col_double()
## )
```

```r
cereal %>%
  group_by(mfr) %>%
  count()
```

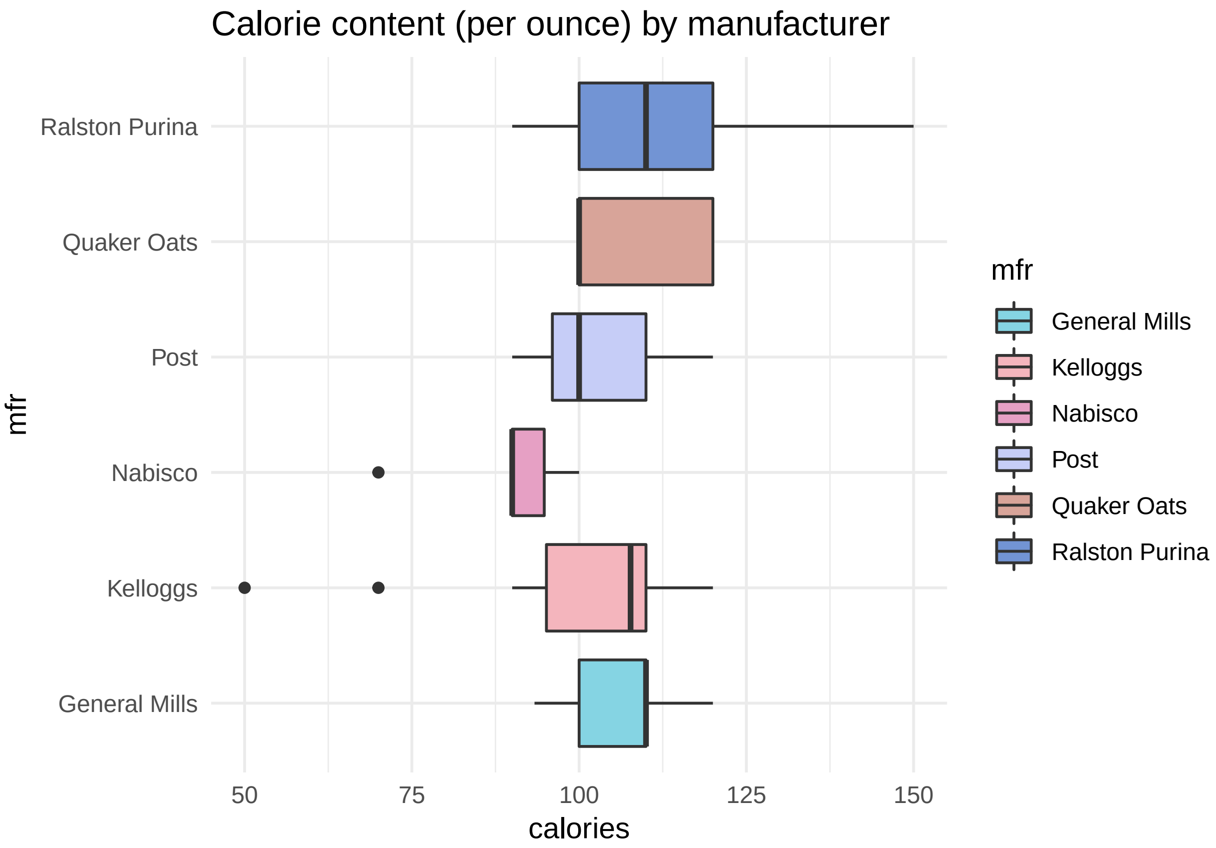
```
## # A tibble: 6 x 2
## # Groups:   mfr [6]
##   mfr       n
##   <chr> <int>
## 1 G        22
## 2 K        23
## 3 N         6
## 4 P         9
## 5 Q         8
## 6 R         8
```

```r
# these variables are pretty meaningless the way they're named.
# let's use str_replace_all() to rename.

# there are some -1s; not a valid value and will mess with calculations
# replace them with NA, using dplyr's na_if()
cereal <- na_if(cereal, -1)

cereal$mfr <- str_replace_all(cereal$mfr, "G", "General Mills")
cereal$mfr <- str_replace_all(cereal$mfr, "K", "Kelloggs")
cereal$mfr <- str_replace_all(cereal$mfr, "N", "Nabisco")
cereal$mfr <- str_replace_all(cereal$mfr, "P", "Post")
cereal$mfr <- str_replace_all(cereal$mfr, "Q", "Quaker Oats")
cereal$mfr <- str_replace_all(cereal$mfr, "R", "Ralston Purina")
```

A boxplot:

```r
personal_palette <- c("#85D4E3", "#F4B5BD", "#E6A0C4",
                      "#C6CDF7", "#D8A499", "#7294D4")
# colors taken from the wesanderson R color palette package
```

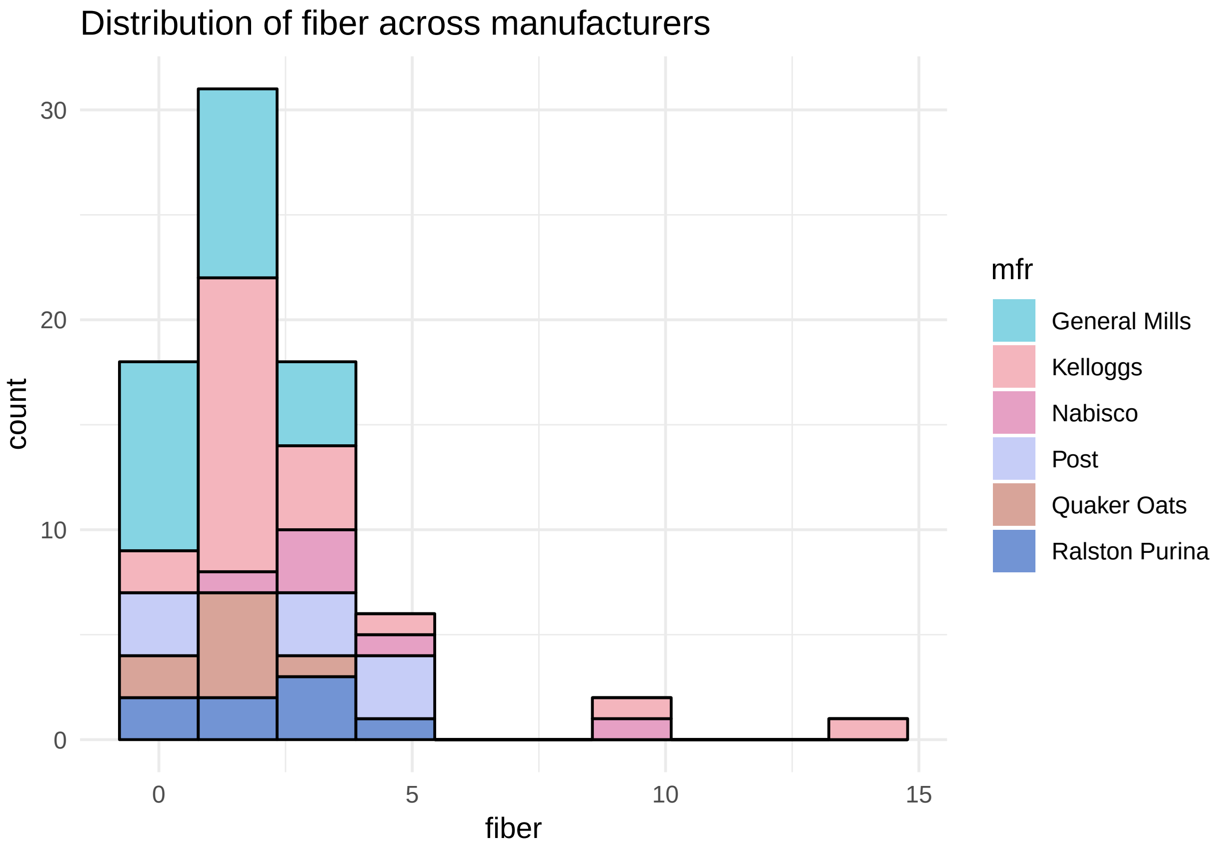
```
cereal %>%
  ggplot(aes(x = mfr, y = calories, fill = mfr)) +
  geom_boxplot() +
  scale_fill_manual(values = personal_palette) +
  coord_flip() +
  theme_minimal() +
  ggtitle("Calorie content (per ounce) by manufacturer")
```



```
  # ggtitle allows you title and subtitle graphs
```

A histogram:

```
cereal %>% ggplot(aes(x = fiber, fill = mfr, color = I("black"))) + geom_histogram(bins = 10) +
  scale_fill_manual(values = personal_palette) + theme_minimal() +
  ggtitle("Distribution of fiber across manufacturers") # don't worry about I("black")
```


Distribution of fiber across manufacturers
count
30
20
10
0
0
5
10
15
fiber
mfr
General Mills
Kelloggs
Nabisco
Post
Quaker Oats
Ralston Purina