Data manipulation II

KY Park

2018 3 29

Data structure

Vector - Q

- Make vector A1 (1, 2, 3, 4)
- What is the length of vector A1 ?
- Make vector (5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
- Make vector (1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3)
- Make vector (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)
- Make vector (1, 3, 5, 7, 9)

Vector - A

```
A1 <- c(1, 2, 3, 4)
A2 <- c("Math", "Science", "Science", "Music", "Math")
A3 <- c(1, 2, "Math", "Music")
length(A1)
```

```
## [1] 4
```

```
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
rep(5, 12)
```

```
##  [1] 5 5 5 5 5 5 5 5 5 5 5 5
```

```
rep(c(1, 2, 3), 6)
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
seq(0, 1, length = 11)
```

```
##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

```
seq(1, 9, by = 2)
```

```
## [1] 1 3 5 7 9
```

Factor - Q

- Make vector Gender (1, 2, 2, 2, 1, 2, 1, 2, 1, 1)
- Make Gender variable into categorical variable. 1 means male and 2 means female.
- What is the reference level of Gender?
- Change the reference level of Gender.

Factor - A

```
Gender <- c(1, 2, 2, 2, 1, 2, 1, 2, 1, 1)
Gender <- factor(Gender, levels = c(1, 2), labels = c("Male", "Female"))
levels(Gender)
```

```
## [1] "Male"   "Female"
```

```
Gender <- relevel(Gender, ref = "Female")
```

Matrix - Q

- Make 4x4 matrix Mat1 with 1:16
- Make 4x4 matrix Mat2 with 17:32
- Bind Mat1 and Mat2 by row.
- Bind Mat1 and Mat2 by column.

Matrix - A

```
Mat1 <- matrix(1:16, nrow = 4)
Mat1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
Mat2 <- matrix(17:32, nrow = 4, byrow = T)
Mat2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   17   18   19   20
## [2,]   21   22   23   24
## [3,]   25   26   27   28
## [4,]   29   30   31   32
```

```
rbind(Mat1, Mat2)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
## [5,]   17   18   19   20
## [6,]   21   22   23   24
## [7,]   25   26   27   28
## [8,]   29   30   31   32
```

```
cbind(Mat1, Mat2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    1    5    9   13   17   18   19   20
## [2,]    2    6   10   14   21   22   23   24
## [3,]    3    7   11   15   25   26   27   28
## [4,]    4    8   12   16   29   30   31   32
```

Data frame - Q1

```
intake.pre <- c(5260, 5470, 5640, 6180, 6390, 6515, 6805, 7515, 7515, 8230, 8770)
intake.post <- c(3910, 4220, 3885, 5160, 5645, 4680, 5265, 5975, 6790, 6990, 7335)
intake.race <- c("w", "w", "b", "h", "h", "w", "b", "w", "w", "h", "b")
```

- Make data frame DF1 including before = intake.pre, after = intake.post, and race = intake.race
- Show summary of DF1

Data frame - A1

```
DF1 <- data.frame(before = intake.pre, after = intake.post, race = intake.race)
DF1
```

```
##    before after race
## 1    5260  3910    w
## 2    5470  4220    w
## 3    5640  3885    b
## 4    6180  5160    h
## 5    6390  5645    h
## 6    6515  4680    w
## 7    6805  5265    b
## 8    7515  5975    w
## 9    7515  6790    w
## 10   8230  6990    h
## 11   8770  7335    b
```

```
summary(DF1)
```

```
##      before          after      race
##  Min.   :5260   Min.   :3885   b:3
##  1st Qu.:5910   1st Qu.:4450   h:3
##  Median :6515   Median :5265   w:5
##  Mean   :6754   Mean   :5441
##  3rd Qu.:7515   3rd Qu.:6382
##  Max.   :8770   Max.   :7335
```

Data frame - Q2

- Select rows with DF1$before > 7000
- Select rows with DF1$before > 7000 and show columns (before and after)
- Use **dplyr** package

Data frame - A2

```
DF1[DF1$before > 7000, ]
```

```
##    before after race
## 8    7515  5975    w
## 9    7515  6790    w
## 10   8230  6990    h
## 11   8770  7335    b
```

```
DF1[DF1$before > 7000, c("before", "after")]
```

```
##    before after
## 8    7515  5975
## 9    7515  6790
## 10   8230  6990
## 11   8770  7335
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
DF1 %>% filter(before > 7000)
```

```
##   before after race
## 1   7515  5975    w
## 2   7515  6790    w
## 3   8230  6990    h
## 4   8770  7335    b
```

```
DF1 %>% filter(before > 7000) %>% select(before, after)
```

```
##   before after
## 1   7515  5975
## 2   7515  6790
## 3   8230  6990
## 4   8770  7335
```

Missing Values

Missing values - Q1

- Make variable B1 (1, 2, 3, 4, NA, 6, 7, NA)
- Can you check whether the variable B1 include NA?
- Make varialbe B2 (1, 2, 3, 4, 999, 6, 7, 999). Show summary of B2.
- 999 means missing values. Change 999 into NA.
- Show summary of B2
- What is the sum and mean of B2?

Missing values - A1

```
B1 <- c(1, 2, 3, 4, NA, 6, 7, NA)
is.na(B1)
```

```
## [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```
B2 <- c(1, 2, 3, 4, 999, 6, 7, 999)
summary(B2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.75    5.00  252.62  255.00  999.00
```

```
B2[B2 == 999] <- NA
summary(B2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   2.250   3.500   3.833   5.500   7.000       2
```

```
sum(B2, na.rm = T)
```

Missing values - A2

```
DF1
```

```
##    before after race
## 1    5260  3910    w
## 2    5470  4220    w
## 3    5640  3885    b
## 4    6180  5160    h
## 5    6390  5645    h
## 6    6515  4680    w
## 7    6805  5265    b
## 8    7515  5975    w
## 9    7515  6790    w
## 10   8230  6990    h
## 11   8770  7335    b
```

```
DF1$before[c(3, 5)] <- NA
DF1$after[c(1, 4)] <- NA
```

- Remove rows with missing values from DF1
- Remove rows with missing values from DF1$before

Missing values - Q2

```
na.omit(DF1)
```

```
##    before after race
## 2    5470  4220    w
## 6    6515  4680    w
## 7    6805  5265    b
## 8    7515  5975    w
## 9    7515  6790    w
## 10   8230  6990    h
## 11   8770  7335    b
```

```
complete.cases(DF1)
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
DF1 %>% select(before, race) %>% na.omit
```

```
##    before race
## 1    5260    w
## 2    5470    w
## 4    6180    h
## 6    6515    w
## 7    6805    b
## 8    7515    w
## 9    7515    w
## 10   8230    h
## 11   8770    b
```

```
DF1[!is.na(DF1$before),]
```

```
##    before after race
## 1    5260    NA    w
## 2    5470  4220    w
## 4    6180    NA    h
## 6    6515  4680    w
## 7    6805  5265    b
## 8    7515  5975    w
## 9    7515  6790    w
## 10   8230  6990    h
## 11   8770  7335    b
```

tidyr

Wide and long format

- Wide format is good for human to read
  - Row has all information of one subject.
  - Column has levels of factors
- Long format is used in almost all statiscal function.
  - Column has all levels of factor

```
##   id control treatment
## 1  1       3         5
## 2  2       4         6
## 3  3       3         4
## 4  4       2         3
```

```
##   id condition score
## 1  1   control     3
## 2  2   control     4
## 3  3   control     3
## 4  4   control     2
## 5  1 treatment     5
## 6  2 treatment     6
## 7  3 treatment     4
## 8  4 treatment     3
```

From wide to long format

```
ageGroup <- c("<20", "20 - 29", "30 - 39", "40 - 49", ">50")
Amethod <- c(7, 8, 9, 10, 11)
Bmethod <- c(9, 8, 10, 11, 12)
Cmethod <- c(10, 11, 12, 13, 14)
DF4 <- data.frame(ageGroup, Amethod, Bmethod, Cmethod)
DF4
```

```
##   ageGroup Amethod Bmethod Cmethod
## 1      <20       7       9      10
## 2  20 - 29       8       8      11
## 3  30 - 39       9      10      12
## 4  40 - 49      10      11      13
## 5      >50      11      12      14
```

- Compare scores among the groups

tidyr::gather

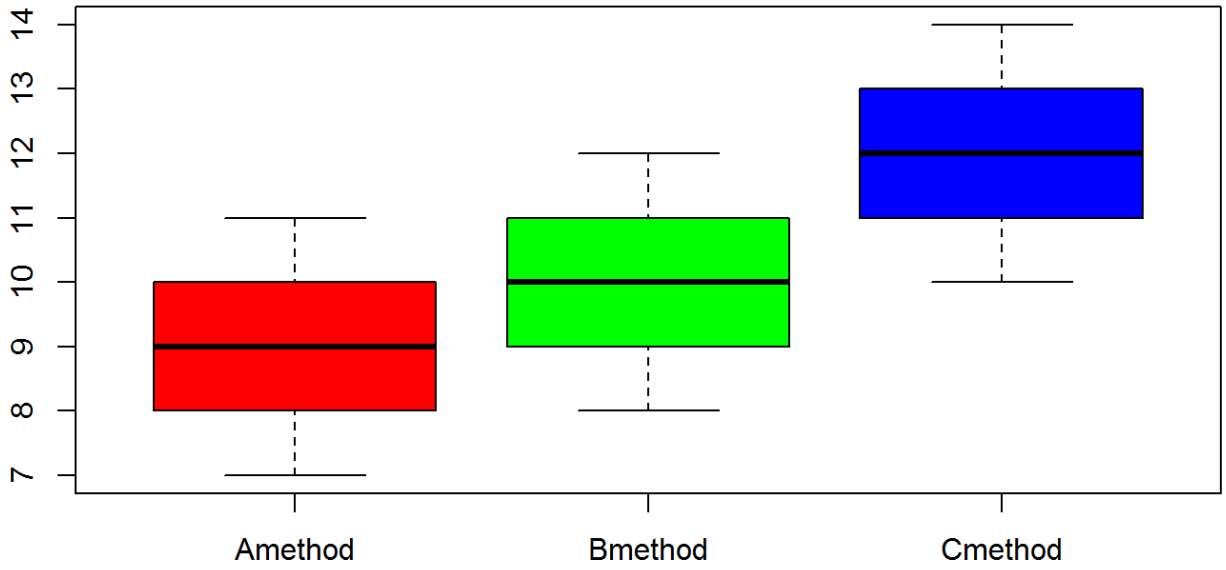
```
# install.packages("tidyr")
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.3
```

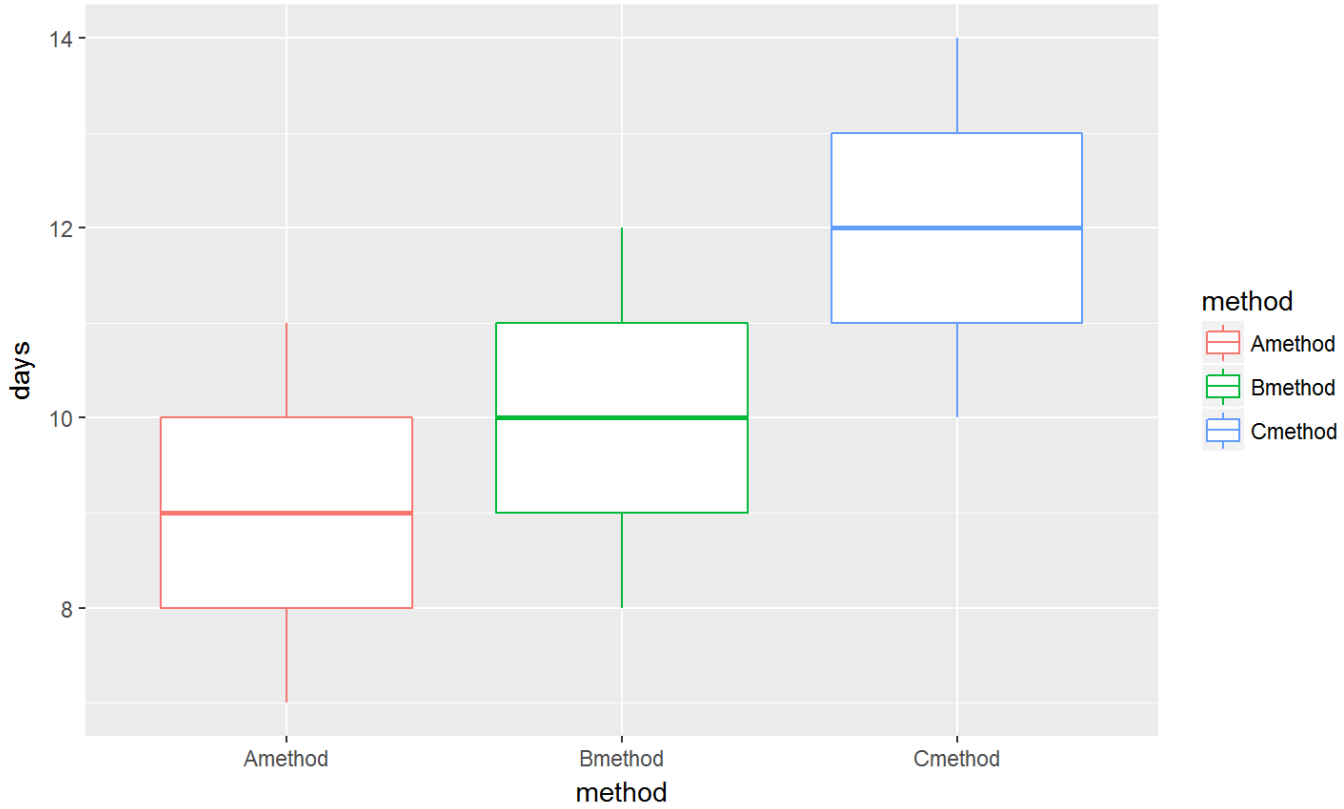
```
DF4_long <-  gather(DF4, method, days, Amethod:Cmethod)
DF4_long
```

```
##    ageGroup  method days
## 1       <20 Amethod    7
## 2    20 - 29 Amethod    8
## 3    30 - 39 Amethod    9
## 4    40 - 49 Amethod   10
## 5       >50 Amethod   11
## 6       <20 Bmethod    9
## 7    20 - 29 Bmethod    8
## 8    30 - 39 Bmethod   10
## 9    40 - 49 Bmethod   11
## 10      >50 Bmethod   12
```

```
boxplot(DF4_long$days ~ DF4_long$method, col = rainbow(3))
```



```
library(ggplot2)
ggplot(DF4_long, aes(x = method, y = days, col = method)) + geom_boxplot()
```



```
result_aov <- aov(days ~ method, data = DF4_long)
summary(result_aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## method       2  23.33   11.67   4.667 0.0317 *
## Residuals   12  30.00    2.50
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(result_aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = days ~ method, data = DF4_long)
##
## $method
##                 diff        lwr      upr     p adj
## Bmethod-Amethod    1 -1.6678637 3.667864 0.5907706
## Cmethod-Amethod    3  0.3321363 5.667864 0.0277219
## Cmethod-Bmethod    2 -0.6678637 4.667864 0.1545800
```