Data Wrangling Using Dplyr

TA Crystal

9/8/2020

Contents

Loading R packages

- dplyr: dataframe manipulation
- ggplot2: visualization

```
#install packages only if you have not already done so
list.of.packages <- c( "dplyr", "tidyverse")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

#library packages
```

```r
for (pkg in c("dplyr", "tidyverse")) {
  library(pkg, character.only = TRUE)
  }
```

```r
# read in data
load("surgery_data.RData")
```

I. Mutate Function

Example I.i: Change the label for a categorical variable

Instead of abrreviation for the gender, “F” and “M”, we want them to be “Female” and “Male”.

Before any data wrangling, we will always perform the 3-step procedure for the variable we are interested in:

1. check the data type (character or integer or others)
2. check whether there are NAs, how many NAs are there in the variable
3. what values are there in the variable. Use `table()` function for categorical variables, use `summary()` function for numerica variables.

```r
#step 1.
class(surgery_data$gender) #check the data type
```

```
## [1] "character"
```

```r
#step 2.
anyNA(surgery_data$gender) #check whether there are NA values
```

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

```r
table(is.na(surgery_data$gender)) #gives the count of NA values: 3
```

```
##
## FALSE  TRUE
## 31998     3
```

```r
#step 3.
table(surgery_data$gender) #check how many non-NA levels are there in the gender variable
```

```
##
##     F     M
## 17230 14768
```

```r
#overwirte gender variable
surgery_data <- surgery_data%>%
  mutate(gender = if_else(gender == "F", "Female",
                  if_else(gender == "M", "Male", "Unknown")))
```

Example I.ii: Group patients whose race, gender are NA into a seperate group

There are 480 patients who have NA values for race. We don't want to exclude these sample from our data, let's treat them as a seperate group called "Unknown"

```
#step 1.
class(surgery_data$race)
```

```
## [1] "character"
```

```
#step 2.
anyNA(surgery_data$race) #check whether there are NA values
```

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

```
table(is.na(surgery_data$race)) #gives the count of NA values: 3
```

```
##
## FALSE  TRUE
## 31521   480
```

```
#step 3.
table(surgery_data$race)
```

```
##
## African American        Caucasian            Other
##             3790            26488             1243
```

```
surgery_data <- surgery_data%>%
  mutate(race = if_else(is.na(race), "Unknown", race))%>%
  mutate(gender = if_else(is.na(gender), "Unknown", gender))

table(surgery_data$race)
```

```
##
## African American        Caucasian            Other          Unknown
##             3790            26488             1243              480
```

```
table(surgery_data$gender)
```

```
##
##  Female    Male Unknown
##   17230   14768       3
```

```
anyNA(surgery_data$race)
```

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

```
anyNA(surgery_data$gender)
```

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

Example I.iii: Create age groups from a numeric age variable

Currently age is a numeric variable, we want to form five age groups for analysis: less than 20, 20-40, 40-60, 60-80, 80+.

```
#step 1
class(surgery_data$age) #check the data type
```

```
## [1] "numeric"
```

```
#step 2
anyNA(surgery_data$age) #check whether there are NA values
```

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

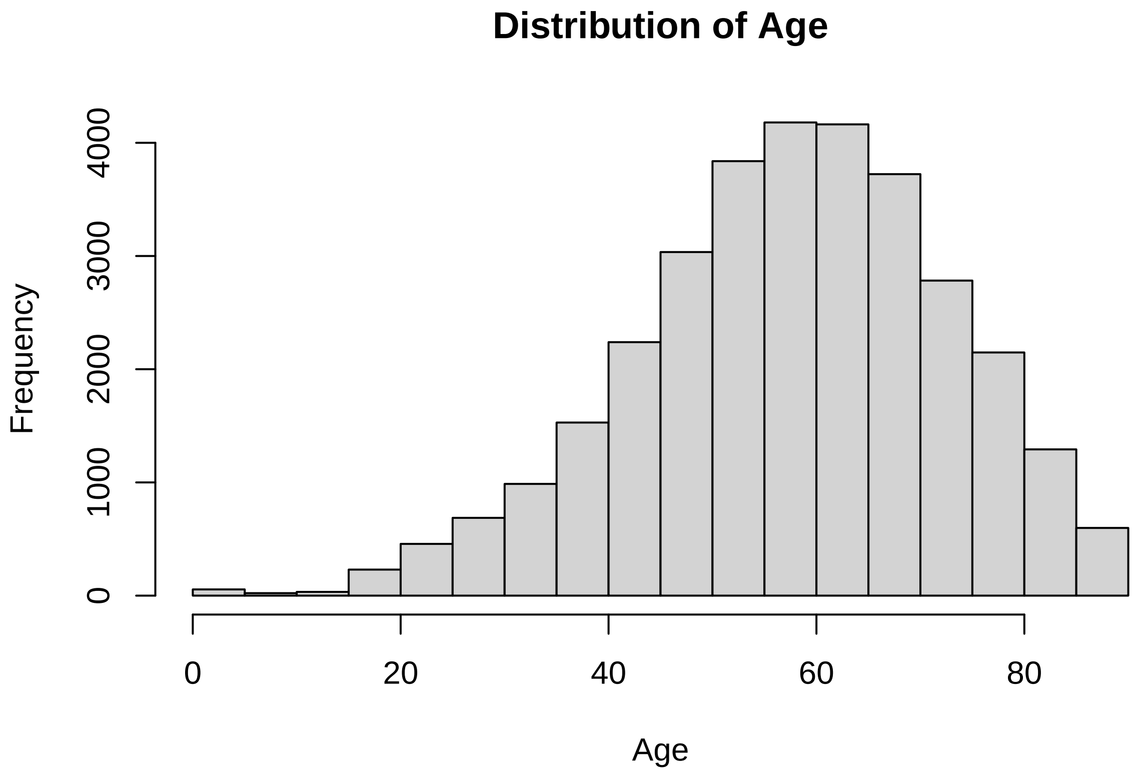
```
table(is.na(surgery_data$age)) #gives the count of NA values: 3
```

```
##
## FALSE  TRUE
## 31999     2
```

```
#step 3.
summary(surgery_data$age) #check the range of the variable
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00   48.20   58.60   57.66   68.30   90.00       2
```

```
hist(surgery_data$age,
     main = "Distribution of Age",
     xlab = "Age") #check the distribution of the variable, which helps us to seperate into groups
```



```
surgery_data <- surgery_data%>%
  mutate(age_group = if_else(age < 20, "less than 20",
                     if_else(age < 40, "20-40 yrs",
                     if_else(age < 60, "40-60 yrs",
                     if_else(age < 80, "60-80 yrs",
                             "80+")))))
table(surgery_data$age_group)
```

```
##
##    20-40 yrs    40-60 yrs    60-80 yrs          80+ less than 20
##         3628        13255        12857         1924          335
```

II. Select Function

Example II.i: Only keep variables of interest in the dataframe

```
surgery_data_subset <- surgery_data%>%
  select(age, gender, bmi, hour, race)
glimpse(surgery_data_subset)
```

```
## Rows: 32,001
## Columns: 5
## $ age    <dbl> 67.8, 39.5, 56.5, 71.0, 56.3, 57.7, 56.6, 64.2, 66.2, 20.1, ...
## $ gender <chr> "Male", "Female", "Female", "Male", "Male", "Female", "Male"...
## $ bmi    <dbl> 28.04, 37.85, 19.56, 32.22, 24.32, 40.30, 64.57, 43.20, 28.0...
## $ hour   <dbl> 9.03, 18.48, 7.88, 8.80, 12.20, 7.67, 9.53, 7.52, 16.35, 16....
## $ race   <chr> "Caucasian", "Caucasian", "Caucasian", "Caucasian", "African...
```

III. Filter Function

Example III.i: Identify only African American patients

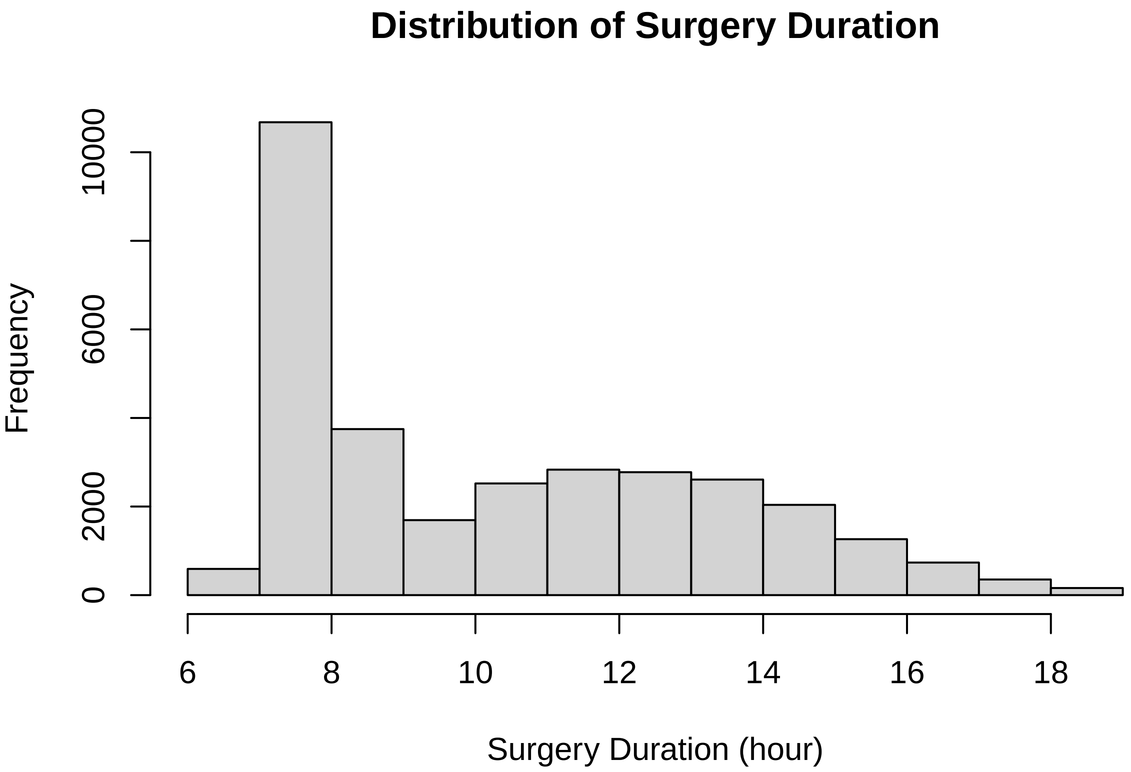
```
table(surgery_data$race)
```

```
##
## African American        Caucasian            Other          Unknown
##             3790            26488             1243              480
```

```
surgery_data_AfricanAmerican <- surgery_data%>%
  filter(race == "African American")
```

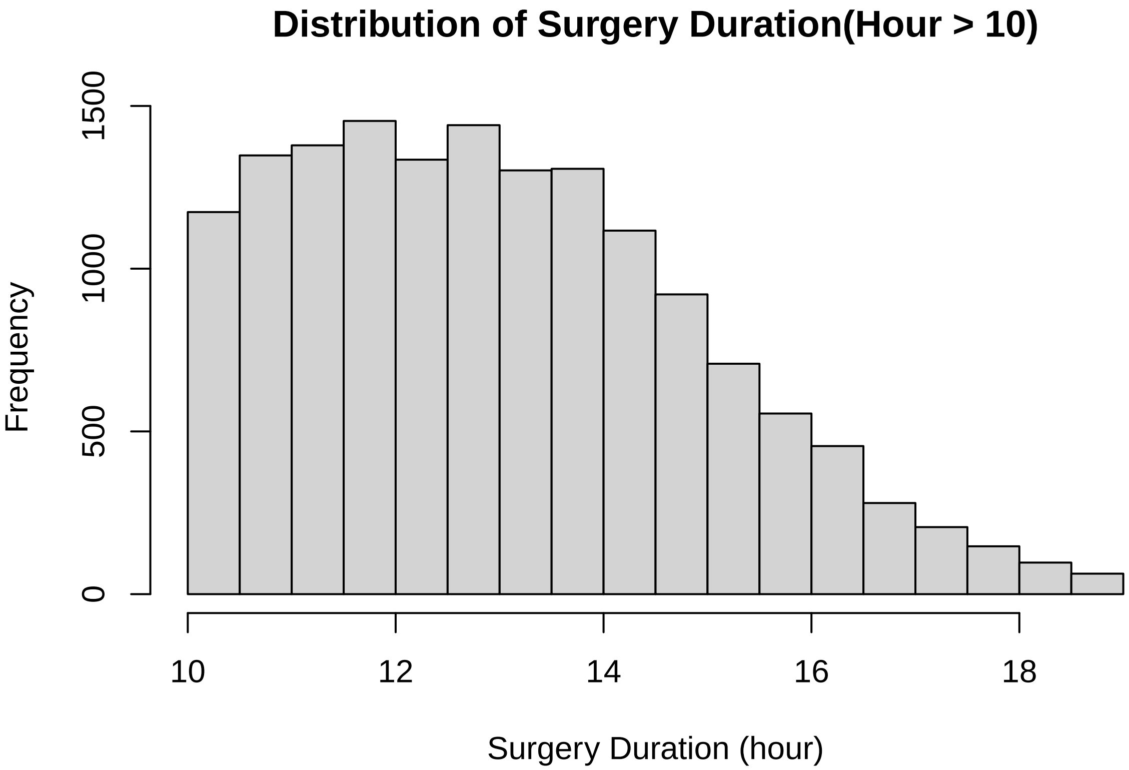
Example III.ii: Identify patients who's surgery time is longer than 10 hours

```
hist(surgery_data$hour,
     main = "Distribution of Surgery Duration",
     xlab = "Surgery Duration (hour)")
```



```
surgery_data_10hr<- surgery_data%>%
  filter(hour > 10)

hist(surgery_data_10hr$hour,
     main = "Distribution of Surgery Duration(Hour > 10)",
     xlab = "Surgery Duration (hour)")
```



IV. Summarize Function

Example IV.i: Identify the average surgery hour for each race group

```
surgery_data%>%
  group_by(race)%>%
  summarize(count = n(),
            hour_mean = mean(hour),
           hour_median= median(hour),
           hour_sd = sd(hour))%>%
  mutate(perc = count/sum(count) * 100)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 6
##   race             count hour_mean hour_median hour_sd  perc
##   <chr>            <int>     <dbl>       <dbl>   <dbl> <dbl>
## 1 African American  3790      10.6       10.1     2.98 11.8
## 2 Caucasian        26488      10.4        9.6     2.91 82.8
## 3 Other             1243      10.3        9.28    2.94  3.88
## 4 Unknown            480      10.5        9.45    2.91  1.50
```

Example IV.ii: Further investigate within each race, what's the average surgery hour for different asa statis

```
table <- surgery_data%>%
  mutate(asa_status = if_else(is.na(asa_status), "Unknown", asa_status))%>%
  group_by(race, asa_status)%>%
  summarize(count = n(),
            hour_mean = mean(hour),
            hour_median= median(hour),
            hour_sd = sd(hour))%>%
  filter(count > 5)
```

```
## `summarise()` regrouping output by 'race' (override with `.groups` argument)
```

```
table
```

```
## # A tibble: 13 x 6
## # Groups:   race [4]
##    race             asa_status count hour_mean hour_median hour_sd
##    <chr>            <chr>      <int>     <dbl>       <dbl>   <dbl>
##  1 African American I-II        1839      10.5        9.83    3.01
##  2 African American III         1785      10.6       10.2     2.94
##  3 African American IV-VI        165      11.1       11.1     2.93
##  4 Caucasian        I-II       14443      10.2        9.22    2.87
##  5 Caucasian        III        11201      10.5        9.87    2.92
##  6 Caucasian        IV-VI        837      11.1       11.0     3.04
##  7 Caucasian        Unknown        7      11.1       13.0     3.38
##  8 Other            I-II         718      10.3        8.87    2.98
##  9 Other            III          492      10.3        9.30    2.86
## 10 Other            IV-VI         33      11.5       11.0     3.18
## 11 Unknown          I-II         261      10.4        8.92    2.96
## 12 Unknown          III          199      10.4        9.58    2.85
## 13 Unknown          IV-VI         20      11.4       12.0     2.82
```

V. Arrange Function

Arrange the median surgery hour in race+asa status group in descending order

```
table%>%
  arrange(-hour_median)
```

```
## # A tibble: 13 x 6
## # Groups:   race [4]
##    race             asa_status count hour_mean hour_median hour_sd
##    <chr>            <chr>      <int>     <dbl>       <dbl>   <dbl>
##  1 Caucasian        Unknown        7      11.1       13.0     3.38
##  2 Unknown          IV-VI         20      11.4       12.0     2.82
##  3 African American IV-VI        165      11.1       11.1     2.93
##  4 Other            IV-VI         33      11.5       11.0     3.18
##  5 Caucasian        IV-VI        837      11.1       11.0     3.04
##  6 African American III         1785      10.6       10.2     2.94
##  7 Caucasian        III        11201      10.5       9.87    2.92
##  8 African American I-II        1839      10.5       9.83    3.01
##  9 Unknown          III          199      10.4       9.58    2.85
## 10 Other            III          492      10.3       9.30    2.86
## 11 Caucasian        I-II       14443      10.2       9.22    2.87
## 12 Unknown          I-II         261      10.4       8.92    2.96
## 13 Other            I-II         718      10.3       8.87    2.98
```