Assignment 1 - Introduction and Wrangling

Abrar Faruque

23Sept2025

Set-Up

The dataset can be accessed here: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

The phenotype attributions are as follows:

1) ID number
2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

Here is some code to read in the dataset

bCancer <- read.csv(here("3_csv", "breastCancer.csv"), header=T)

Questions

1. When installing a package with the `install.packages()` function, the incoming package name needs to be enclosed in quotations. Why is this? [1]

```
# 1.
# .packages function argument takes string, if package was not enclosed with quotations
# then package would appear as a variable with no assigned value/ error not found
```

2. Create a vector object with a numeric value, a character value, and a logical value. What data type is this vector and why? [1]

```r
# 2.
myVector = c (1:3,"hello",TRUE)
summary(myVector) #vector type is 'character'
```

```
##    Length     Class      Mode
##         5 character character
```

```r
# Vector data type is 'character' because of the character value store in vector; this
# changes all other data types to character as well.
```

3. Create a new object called `recent_fruits`.

a) Create 5 rows with the rownames as different fruits and 3 columns named "colour", "shape", and "taste". Describe the fruit's colour and shape in one word, rank the taste out of 10. Fill out the table! [0.5]
b) Calculate the mean taste rank from your previous table. [0.5]
c) c) Write a function called "middle_mean" that removes the largest and smallest taste rating, and then calculates the mean. Use the function on your dataset. *Hint*: you may need any of these functions: min(), max(), nrow(), length(). [0.75]

```r
#3a
recent_fruits<- data.frame( colour <- c("Red","Green","Purple","Yellow","Yellow"),
                            shape <- c("Spherical","Bell-shaped","Spherical","Elongated","Spheroid"),
                            taste <- c(5,6,8,9,2)

)
rownames(recent_fruits) <- c("Apple", "Pear", "Grape","Banana","Lemon")
print(recent_fruits)
```

```
##        colour....c..Red....Green....Purple....Yellow....Yellow..
## Apple                                                        Red
## Pear                                                       Green
## Grape                                                     Purple
## Banana                                                    Yellow
## Lemon                                                     Yellow
##        shape....c..Spherical....Bell.shaped....Spherical....Elongated...
## Apple                                                          Spherical
## Pear                                                         Bell-shaped
## Grape                                                          Spherical
## Banana                                                         Elongated
## Lemon                                                           Spheroid
##        taste....c.5..6..8..9..2.
## Apple                          5
## Pear                           6
## Grape                          8
## Banana                         9
## Lemon                          2
```

```r
#3b
print(mean(recent_fruits$taste))
```

```
## [1] 6
```

```
#3c
middle_mean <- function(x){
  values <- x[-c(which(x==max(x)),which(x==min(x)))]
  return (mean(values))
}
print(middle_mean(recent_fruits$taste))
```

```
## [1] 6.333333
```

4. Use the bCancer dataset.

a) Subset the samples with a mean radius less than 20 into a new object called `large_tumour`. [0.5]
b) From the `large_tumour` object, subset for the columns describing the `id`, `diagnosis`, `radius_mean`, `texture_mean`, `smoothness_mean`, and `compactness_mean`. Reshape the data into a longer format so that there are only four columns: `id`, `diagnosis`, `variable`, and `value`. Save this into a new object called `large_tumour_long`. [0.5]
c) Group your data by diagnosis and variable, and calculate the mean, median, max, min, standard deviation, and count the number of observations. Save this into a table object called `large_tumour_summary`. *Hint*: first, figure out what type of data format you may need (`large_tumour_long` or `large_tumour`), and then check that your data columns are the correct class. [0.5]
d) Which patients have the 5 largest `radius_mean`, and what is their diagnosis? Which patients have the 5 smallest `radius_mean`, and what is their diagnosis? What does this trend potentially tell you about the data? *Hint*: there are multiple measurements for each patient, so use mean() to average these measurements. [0.75]

```
# CODE CELL 4
setwd("C:/Users/Delwar/Desktop/Humber/BINF5003- Data Mining, Modeling, and Biostatistcs/Labs")
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.2
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
breastCancer <- read.csv("breastCancer.csv", header = TRUE)

#4a)
large_tumour = breastCancer[breastCancer$radius_mean<20,]

#4b)
large_tumour_long <- large_tumour[,c("id", "diagnosis", "radius_mean", "texture_mean",
                                     "smoothness_mean", "compactness_mean")]
large_tumour_long <- large_tumour_long %>%
  pivot_longer(cols = "radius_mean":"compactness_mean",
```

```
               names_to = "variable",
               values_to = "value")

#4c)
large_tumour_summary<-large_tumour_long %>%
  group_by(diagnosis, variable) %>% #"Group your data by diagnosis and variable"
  summarise(mean = mean(value),
            median = median(value),
            max = max(value),
            min = min(value),
            standard_deviation = sd(value),
            length = length(value)
            )
```

```
## `summarise()` has grouped output by 'diagnosis'. You can override using the
## `.groups` argument.
```

```
print(large_tumour_summary)#there are 8 observations
```

```
## # A tibble: 8 x 8
## # Groups:   diagnosis [2]
##   diagnosis variable       mean  median    max     min standard_deviation length
##   <chr>     <chr>         <dbl>   <dbl>  <dbl>   <dbl>              <dbl>  <int>
## 1 B         compactnes~  0.0801  0.0753  0.224  0.0194             0.0337    357
## 2 B         radius_mean 12.1    12.2    17.8    6.98               1.78      357
## 3 B         smoothness~  0.0925  0.0908  0.163  0.0526             0.0134    357
## 4 B         texture_me~ 17.9    17.4    33.8    9.71               4.00      357
## 5 M         compactnes~  0.139   0.131   0.311  0.0460             0.0497    167
## 6 M         radius_mean 16.3    16.2    19.9   11.0                2.26      167
## 7 M         smoothness~  0.103   0.102   0.142  0.0737             0.0125    167
## 8 M         texture_me~ 21.4    21.2    39.3   10.4                3.78      167
```

```
#4d)
Largest <- breastCancer[,c("id", "diagnosis", "radius_mean")] %>%
  slice_max(radius_mean,n=5)
Smallest <- breastCancer[,c("id", "diagnosis", "radius_mean")] %>%
  slice_min(radius_mean,n=5)
print(Largest) #the tumors with largest mean radius are all "M" (Malignant) diagnosis
```

```
##          id diagnosis radius_mean
## 1   8810703         M       28.11
## 2 911296202         M       27.42
## 3    873592         M       27.22
## 4    899987         M       25.73
## 5   8611555         M       25.22
```

```
print(Smallest) #the tumors with smallest mean radius are all "B" (Benign) diagnosis
```

```
##         id diagnosis radius_mean
## 1   862722         B       6.981
## 2   921362         B       7.691
```

```
## 3   921092         B       7.729
## 4    92751         B       7.760
## 5 85713702         B       8.196
```