Class09_Mini_Project

Monica Lin (PID: A15524235)

10/26/2021

1. Exploratory data analysis

Preparing the data

First, download and import our data. Use the `read.csv()` function to read the Comma-Separated Values file. Assign the result to an object called `wisc.df`.

```r
# Save your input data file into your Project directory
fna.data <- "WisconsinCancer.csv"

# Complete the following code to input the data and store as wisc.df
wisc.df <- read.csv(fna.data, row.names=1)
```

Examine your input data to ensure column names are set correctly (you can use the `View()` or `head()` functions here).

```r
head(wisc.df)
```

```
##          diagnosis radius_mean texture_mean perimeter_mean area_mean
## 842302           M       17.99        10.38         122.80    1001.0
## 842517           M       20.57        17.77         132.90    1326.0
## 84300903         M       19.69        21.25         130.00    1203.0
## 84348301         M       11.42        20.38          77.58     386.1
## 84358402         M       20.29        14.34         135.10    1297.0
## 843786           M       12.45        15.70          82.57     477.1
##          smoothness_mean compactness_mean concavity_mean concave.points_mean
## 842302           0.11840          0.27760         0.3001             0.14710
## 842517           0.08474          0.07864         0.0869             0.07017
## 84300903         0.10960          0.15990         0.1974             0.12790
## 84348301         0.14250          0.28390         0.2414             0.10520
## 84358402         0.10030          0.13280         0.1980             0.10430
## 843786           0.12780          0.17000         0.1578             0.08089
##          symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 842302          0.2419                0.07871    1.0950     0.9053        8.589
## 842517          0.1812                0.05667    0.5435     0.7339        3.398
## 84300903        0.2069                0.05999    0.7456     0.7869        4.585
## 84348301        0.2597                0.09744    0.4956     1.1560        3.445
## 84358402        0.1809                0.05883    0.7572     0.7813        5.438
## 843786          0.2087                0.07613    0.3345     0.8902        2.217
##          area_se smoothness_se compactness_se concavity_se concave.points_se
## 842302    153.40      0.006399        0.04904      0.05373           0.01587
## 842517     74.08      0.005225        0.01308      0.01860           0.01340
## 84300903   94.03      0.006150        0.04006      0.03832           0.02058
## 84348301   27.23      0.009110        0.07458      0.05661           0.01867
## 84358402   94.44      0.011490        0.02461      0.05688           0.01885
## 843786     27.19      0.007510        0.03345      0.03672           0.01137
##          symmetry_se fractal_dimension_se radius_worst texture_worst
## 842302       0.03003             0.006193        25.38         17.33
## 842517       0.01389             0.003532        24.99         23.41
## 84300903     0.02250             0.004571        23.57         25.53
## 84348301     0.05963             0.009208        14.91         26.50
## 84358402     0.01756             0.005115        22.54         16.67
## 843786       0.02165             0.005082        15.47         23.75
##          perimeter_worst area_worst smoothness_worst compactness_worst
## 842302            184.60     2019.0           0.1622            0.6656
## 842517            158.80     1956.0           0.1238            0.1866
## 84300903          152.50     1709.0           0.1444            0.4245
## 84348301           98.87      567.7           0.2098            0.8663
## 84358402          152.20     1575.0           0.1374            0.2050
## 843786            103.40      741.6           0.1791            0.5249
##          concavity_worst concave.points_worst symmetry_worst
## 842302            0.7119               0.2654         0.4601
## 842517            0.2416               0.1860         0.2750
## 84300903          0.4504               0.2430         0.3613
## 84348301          0.6869               0.2575         0.6638
## 84358402          0.4000               0.1625         0.2364
## 843786            0.5355               0.1741         0.3985
##          fractal_dimension_worst
## 842302                   0.11890
## 842517                   0.08902
## 84300903                 0.08758
## 84348301                 0.17300
## 84358402                 0.07678
## 843786                   0.12440
```

Note: the first column here `wisc.df$diagnosis` is a pathologist-provided expert diagnosis. We will not be using this for our unsupervised analysis as it essentially answers the question of which cell samples are malignant or benign.

To make sure we don't accidentally include this in our analysis, let's create a new data.frame that omits this first column.

```
# We can use -1 here to remove the first column
wisc.data <- wisc.df[,-1]
```

Finally, set up a separate new vector called `diagnosis` that contains the data from the diagnosis column of the original dataset. We will store this as a **factor** (useful for plotting) and use this later to check our results. Make the vector so that diagnsosi will be value 1 if malignant ("M") and 0 otherwise ("B", benign).

```
# Create diagnosis vector for later
diagnosis <- as.numeric(wisc.df$diagnosis == "M")
```

Exploratory data analysis

Explore the data you created before (`wisc.data` and `diagnosis`) to answer these questions:

> **Q1**. How many observations are in this dataset?

```
dim(wisc.data)
```

```
## [1] 569  30
```

There are 569 rows, i.e. 569 observations.

> **Q2**. How many of the observations have a malignant diagnosis?

```
sum(diagnosis)
```

```
## [1] 212
```

> **Q3**. How many variables/features in the data are suffixed with `_mean`?

```
length(grep(pattern = "_mean", x = colnames(wisc.data)))
```

```
## [1] 10
```

2. Principal Component Analysis

Performing PCA

The next step in our analysis is to perform PCA on `wisc.data`. Check first if the data need to be scaled before performing PCA. Recall two common reasons for scaling data include:

- The input variables use different units of measurement
- The input variables have significantly different variances