Demonstration of R Markdown

JAS

## R Markdown Demonstration using Breast Imaging Data

Data Citation: This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. See Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. Cancer Letters 1994; 77:163-171.

### Step 1: Load Packages

We will be using two packages in addition to base R. Amelia is for missing data and will provide us with a plot of missingness. Caret is primarly from classification and regression trees, but has many features that are useful for data preprocessing. We will be using a function to partition data into training and testing.

#install.packages("caret")  
  
library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.8.1, built: 2022-11-18)  
## ## Copyright (C) 2005-2023 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.5.0   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()

### Step 2: Load Data into R Environment and Perform Data Cleaning

#### Illustrates the following:

1. How to load a flat text file
2. How to assign column names when none are provided
3. How to check variable types across the dataframe
4. How to recode missing indicators, change variable types and explore variable distributions
5. Create a quick plot to indicate missingness
6. Remove duplicates and missings
7. Create a quick and dirty plot to compare features across outcome levels

bc.data<-read.csv("data/breast-cancer-wisconsin.data.txt", header=FALSE)  
  
var.names<-c("id", "clump\_thickness", "uniformity\_csize", "uniformity\_cshape", "marg\_adhesion", "single\_ecell\_size", "bare\_nuclei", "b\_chromatin", "normal\_nucleoli", "mitoses", "outcome")  
  
colnames(bc.data)<-var.names  
str(bc.data)

## 'data.frame': 699 obs. of 11 variables:  
## $ id : int 1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...  
## $ clump\_thickness : int 5 5 3 6 4 8 1 2 2 4 ...  
## $ uniformity\_csize : int 1 4 1 8 1 10 1 1 1 2 ...  
## $ uniformity\_cshape: int 1 4 1 8 1 10 1 2 1 1 ...  
## $ marg\_adhesion : int 1 5 1 1 3 8 1 1 1 1 ...  
## $ single\_ecell\_size: int 2 7 2 3 2 7 2 2 2 2 ...  
## $ bare\_nuclei : chr "1" "10" "2" "4" ...  
## $ b\_chromatin : int 3 3 3 3 3 9 3 3 1 2 ...  
## $ normal\_nucleoli : int 1 2 1 7 1 7 1 1 1 1 ...  
## $ mitoses : int 1 1 1 1 1 1 1 1 5 1 ...  
## $ outcome : int 2 2 2 2 2 4 2 2 2 2 ...

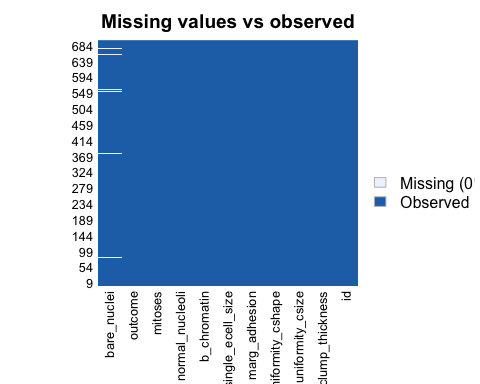
bc.data[bc.data=="?"]<-NA  
bc.data$bare\_nuclei<-as.numeric(bc.data$bare\_nuclei)  
  
bc.data$outcome<-as.factor(bc.data$outcome)  
levels(bc.data$outcome)<-c("Benign", "Malignant")  
str(bc.data)

## 'data.frame': 699 obs. of 11 variables:  
## $ id : int 1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...  
## $ clump\_thickness : int 5 5 3 6 4 8 1 2 2 4 ...  
## $ uniformity\_csize : int 1 4 1 8 1 10 1 1 1 2 ...  
## $ uniformity\_cshape: int 1 4 1 8 1 10 1 2 1 1 ...  
## $ marg\_adhesion : int 1 5 1 1 3 8 1 1 1 1 ...  
## $ single\_ecell\_size: int 2 7 2 3 2 7 2 2 2 2 ...  
## $ bare\_nuclei : num 1 10 2 4 1 10 10 1 1 1 ...  
## $ b\_chromatin : int 3 3 3 3 3 9 3 3 1 2 ...  
## $ normal\_nucleoli : int 1 2 1 7 1 7 1 1 1 1 ...  
## $ mitoses : int 1 1 1 1 1 1 1 1 5 1 ...  
## $ outcome : Factor w/ 2 levels "Benign","Malignant": 1 1 1 1 1 2 1 1 1 1 ...

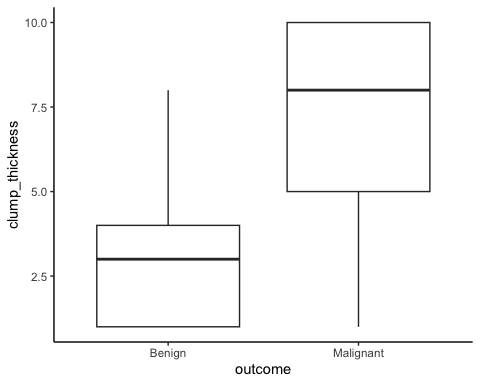
summary(bc.data)

## id clump\_thickness uniformity\_csize uniformity\_cshape  
## Min. : 61634 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 870688 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000   
## Median : 1171710 Median : 4.000 Median : 1.000 Median : 1.000   
## Mean : 1071704 Mean : 4.418 Mean : 3.134 Mean : 3.207   
## 3rd Qu.: 1238298 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000   
## Max. :13454352 Max. :10.000 Max. :10.000 Max. :10.000   
##   
## marg\_adhesion single\_ecell\_size bare\_nuclei b\_chromatin   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000   
## Median : 1.000 Median : 2.000 Median : 1.000 Median : 3.000   
## Mean : 2.807 Mean : 3.216 Mean : 3.545 Mean : 3.438   
## 3rd Qu.: 4.000 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## NA's :16   
## normal\_nucleoli mitoses outcome   
## Min. : 1.000 Min. : 1.000 Benign :458   
## 1st Qu.: 1.000 1st Qu.: 1.000 Malignant:241   
## Median : 1.000 Median : 1.000   
## Mean : 2.867 Mean : 1.589   
## 3rd Qu.: 4.000 3rd Qu.: 1.000   
## Max. :10.000 Max. :10.000   
##

missmap(bc.data, main = "Missing values vs observed")



#Remove missings  
bc.data<-na.omit(bc.data)  
  
#Remove duplicate IDs  
bc.data=bc.data %>% distinct(id, .keep\_all=TRUE)  
  
#Quick plot comparing clump thickness across Outcome Groups  
ggplot(bc.data, aes(y=clump\_thickness, x=outcome)) +   
 geom\_boxplot()+theme\_classic()



### Step 3: Construct logistic regression models to predict Malignancy  
\* Model 1: Include all features  
\* Model 2: Include only clump thickness  
  
```r  
model.1 <- glm(outcome ~ . ,family=binomial(link='logit'),data=bc.data)  
summary(model.1)

##   
## Call:  
## glm(formula = outcome ~ ., family = binomial(link = "logit"),   
## data = bc.data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.7126 -0.1108 -0.0484 0.0129 2.3074   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.096e+01 1.523e+00 -7.196 6.22e-13 \*\*\*  
## id 2.363e-07 4.349e-07 0.543 0.586888   
## clump\_thickness 6.082e-01 1.664e-01 3.655 0.000257 \*\*\*  
## uniformity\_csize 1.410e-01 2.577e-01 0.547 0.584166   
## uniformity\_cshape 2.117e-01 2.811e-01 0.753 0.451387   
## marg\_adhesion 3.897e-01 1.394e-01 2.795 0.005189 \*\*   
## single\_ecell\_size 5.678e-02 1.734e-01 0.327 0.743315   
## bare\_nuclei 4.792e-01 1.147e-01 4.178 2.94e-05 \*\*\*  
## b\_chromatin 3.837e-01 1.918e-01 2.000 0.045471 \*   
## normal\_nucleoli 2.106e-01 1.235e-01 1.706 0.088094 .   
## mitoses 6.915e-01 3.254e-01 2.125 0.033563 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 826.919 on 629 degrees of freedom  
## Residual deviance: 84.534 on 619 degrees of freedom  
## AIC: 106.53  
##   
## Number of Fisher Scoring iterations: 8

confint(model.1)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) -1.434913e+01 -7.972891e+00  
## id -1.568663e-06 7.009667e-07  
## clump\_thickness 3.099071e-01 9.731492e-01  
## uniformity\_csize -3.220270e-01 6.871021e-01  
## uniformity\_cshape -3.576323e-01 7.428257e-01  
## marg\_adhesion 1.277886e-01 6.865491e-01  
## single\_ecell\_size -2.967387e-01 3.946358e-01  
## bare\_nuclei 2.694766e-01 7.252172e-01  
## b\_chromatin 1.698120e-02 7.760055e-01  
## normal\_nucleoli -2.515750e-02 4.645328e-01  
## mitoses 4.035135e-02 1.252639e+00

model.2<-glm(outcome ~ clump\_thickness, family=binomial(link='logit'), data=bc.data)  
summary(model.2)

##   
## Call:  
## glm(formula = outcome ~ clump\_thickness, family = binomial(link = "logit"),   
## data = bc.data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2522 -0.4434 -0.1760 0.2564 2.8897   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.10461 0.39601 -12.89 <2e-16 \*\*\*  
## clump\_thickness 0.94480 0.07814 12.09 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 826.92 on 629 degrees of freedom  
## Residual deviance: 419.21 on 628 degrees of freedom  
## AIC: 423.21  
##   
## Number of Fisher Scoring iterations: 6

ci<-confint(model.2)

## Waiting for profiling to be done...

#Extract results--simple base stats way  
OR<-exp(model.2$coefficients[2])  
LL<-exp(ci[2,1])  
UL<-exp(ci[2,2])

```