Egzamin

Hanna Hankus

22/01/2022

CEL

Zadaniem jest, wczytanie zbioru `wdbc.csv` zawierającego informacje o pacjentach chorujących na raka piersi. Należy zamodelować zmienną `Diagnosis` oznaczającą typ raka piersi (zmienna przyjmuje wartości: `M` = malignant, `B` = benign).

Po wczytaniu zbioru nalezy dokonać:

- wstępnej eksploracyjnej analizy danych
- wyczyszczenia danych jeśli zachodzi potrzeba (np. braki danych, outliery)
- zamodelowania `Diagnosis` klasyczną regresją logistyczną bez regularyzacji oraz dwoma dowolnymi metodami
- porównanie 3 modeli i wybór najlepszego - trzeba uzasadznić wybrane kryterium wyboru oraz metodykę.

```
wdbc <- read.csv("wdbc.csv", header = TRUE, sep=",")
head(wdbc)
```

```
##   ID.number Diagnosis    V1    V2     V3     V4      V5      V6     V7      V8
## 1    842302         M 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.3001 0.14710
## 2    842517         M 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.0869 0.07017
## 3  84300903         M 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.1974 0.12790
## 4  84348301         M 11.42 20.38  77.58  386.1 0.14250 0.28390 0.2414 0.10520
## 5  84358402         M 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.1980 0.10430
## 6    843786         M 12.45 15.70  82.57  477.1 0.12780 0.17000 0.1578 0.08089
##       V9     V10    V11    V12   V13    V14      V15     V16     V17     V18
## 1 0.2419 0.07871 1.0950 0.9053 8.589 153.40 0.006399 0.04904 0.05373 0.01587
## 2 0.1812 0.05667 0.5435 0.7339 3.398  74.08 0.005225 0.01308 0.01860 0.01340
## 3 0.2069 0.05999 0.7456 0.7869 4.585  94.03 0.006150 0.04006 0.03832 0.02058
## 4 0.2597 0.09744 0.4956 1.1560 3.445  27.23 0.009110 0.07458 0.05661 0.01867
## 5 0.1809 0.05883 0.7572 0.7813 5.438  94.44 0.011490 0.02461 0.05688 0.01885
## 6 0.2087 0.07613 0.3345 0.8902 2.217  27.19 0.007510 0.03345 0.03672 0.01137
##       V19      V20   V21   V22    V23    V24    V25    V26    V27    V28    V29
## 1 0.03003 0.006193 25.38 17.33 184.60 2019.0 0.1622 0.6656 0.7119 0.2654 0.4601
## 2 0.01389 0.003532 24.99 23.41 158.80 1956.0 0.1238 0.1866 0.2416 0.1860 0.2750
## 3 0.02250 0.004571 23.57 25.53 152.50 1709.0 0.1444 0.4245 0.4504 0.2430 0.3613
## 4 0.05963 0.009208 14.91 26.50  98.87  567.7 0.2098 0.8663 0.6869 0.2575 0.6638
## 5 0.01756 0.005115 22.54 16.67 152.20 1575.0 0.1374 0.2050 0.4000 0.1625 0.2364
## 6 0.02165 0.005082 15.47 23.75 103.40  741.6 0.1791 0.5249 0.5355 0.1741 0.3985
##       V30
## 1 0.11890
```

```
## 2 0.08902
## 3 0.08758
## 4 0.17300
## 5 0.07678
## 6 0.12440
```

```
str(wdbc)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ ID.number: int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 .
##  $ Diagnosis: chr  "M" "M" "M" "M" ...
##  $ V1       : num  18 20.6 19.7 11.4 20.3 ...
##  $ V2       : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ V3       : num  122.8 132.9 130 77.6 135.1 ...
##  $ V4       : num  1001 1326 1203 386 1297 ...
##  $ V5       : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ V6       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ V7       : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ V8       : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ V9       : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ V10      : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ V11      : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ V12      : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ V13      : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ V14      : num  153.4 74.1 94 27.2 94.4 ...
##  $ V15      : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ V16      : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ V17      : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ V18      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ V19      : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ V20      : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ V21      : num  25.4 25 23.6 14.9 22.5 ...
##  $ V22      : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ V23      : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ V24      : num  2019 1956 1709 568 1575 ...
##  $ V25      : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ V26      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ V27      : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ V28      : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ V29      : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ V30      : num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

- Zmienne `V1-V30` są zakodowane, dlatego summary() nie ma sensu, ponieważ nie można wysnuć wstępnych wniosków na temat m.in. punktów oddalonych.
- Zmienne `V1-V30` są numeryczne, więc ok, natomiast `Diagnosis` trzeba zmienić na factor

```
wdbc$Diagnosis <- as.factor(wdbc$Diagnosis)
summary(wdbc$Diagnosis)
```

```
##   B   M
## 357 212
```

- Wstępnie założono, że klasy są wystarczająco zbalansowane

Sprawdzenie czy każdy `ID.number` jest unikalny

```
length(unique(wdbc$ID.number))
```

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

Sprawdzenie **NA's**

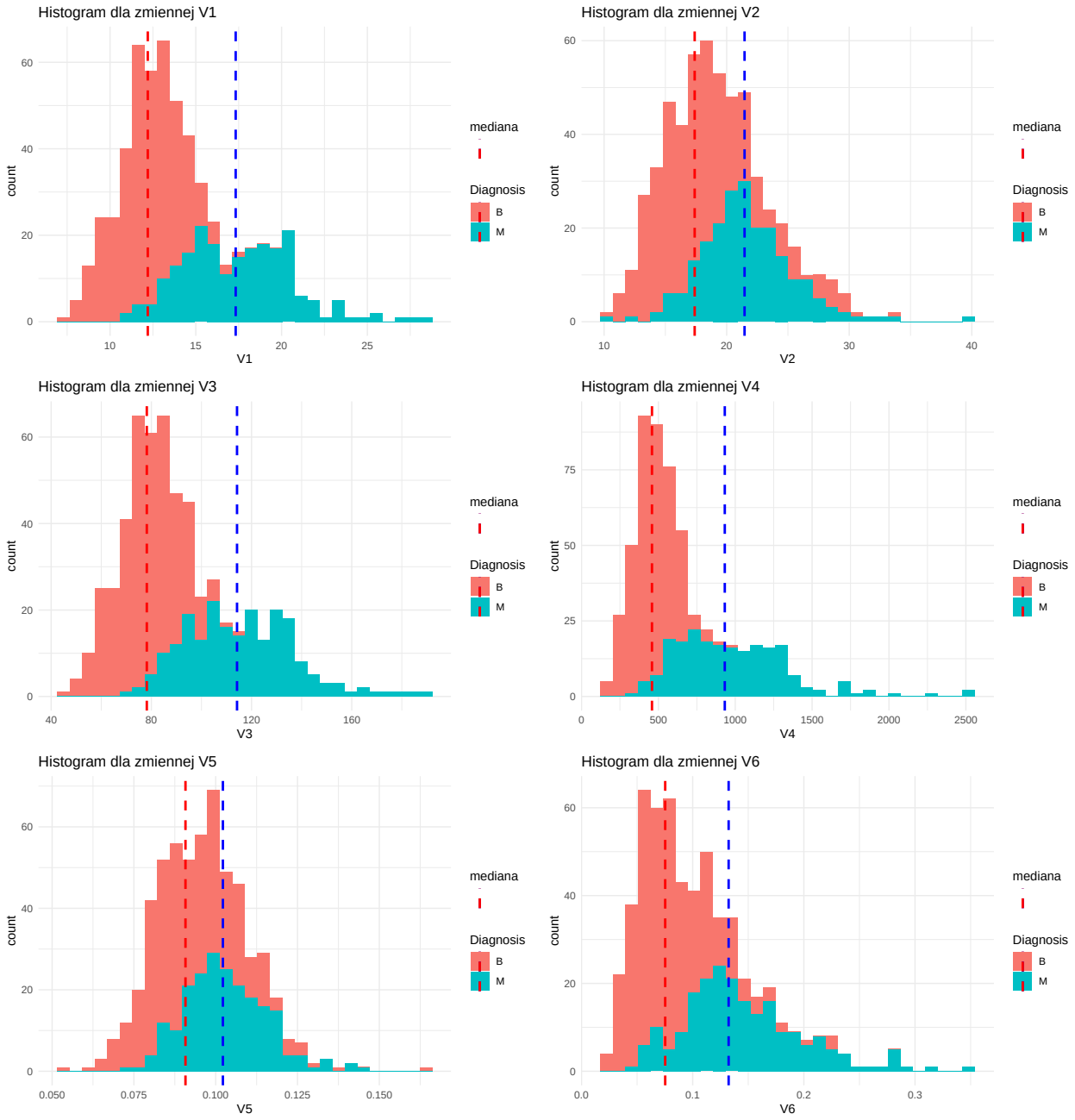
```
NA_check <- function(x){

   NA_df <- matrix(ncol=ncol(x), nrow=1)

  for(i in 1:ncol(x)) {
    NA_df[ , i] <- sum(is.na((x[ , i])))
  }

  NA_df <- data.frame(NA_df)
  colnames(NA_df) <- colnames(x)
  NA_df
}

NA_check(wdbc)
```

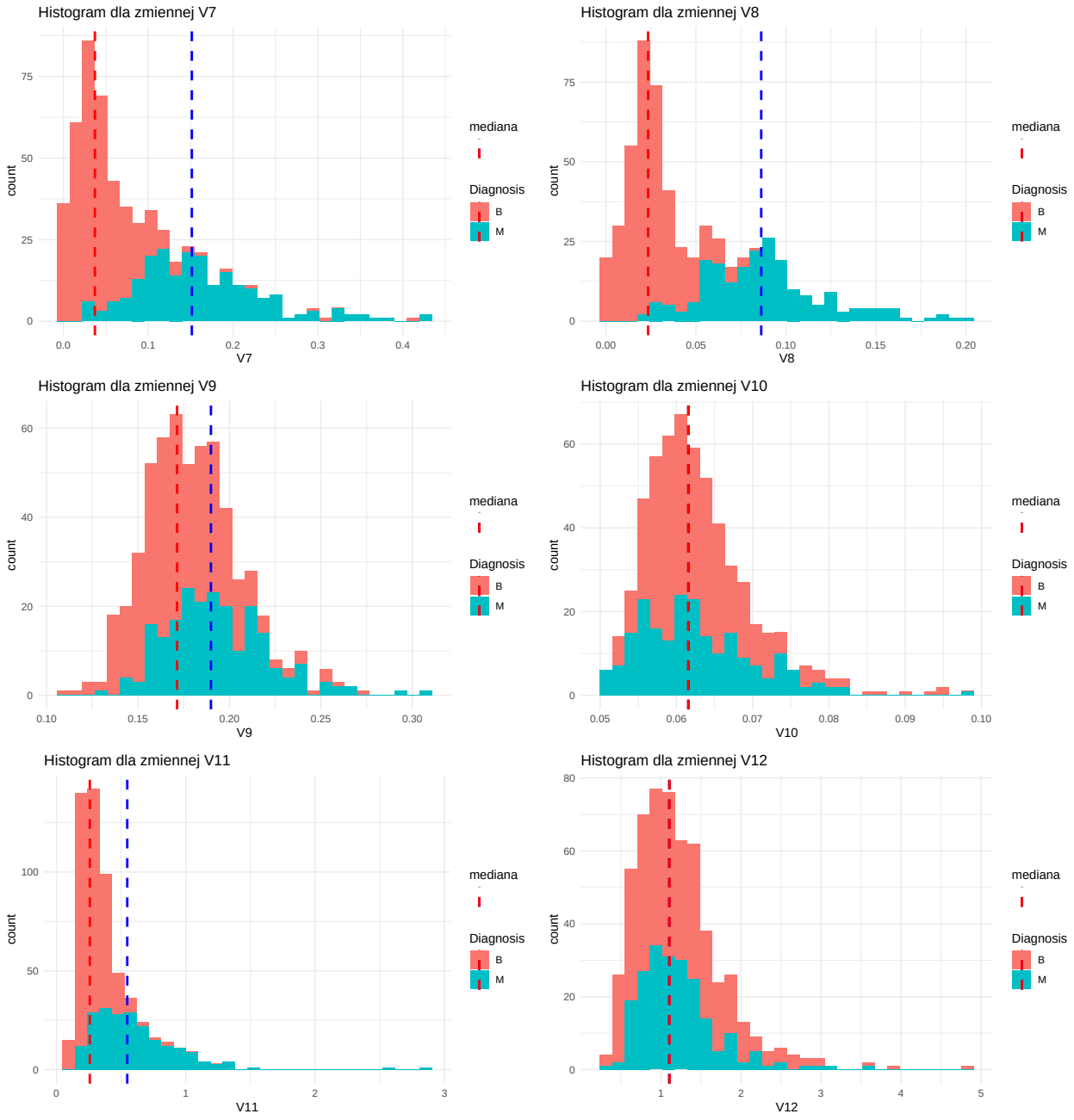
```
##   ID.number Diagnosis V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
## 1         0         0  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0
##   V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

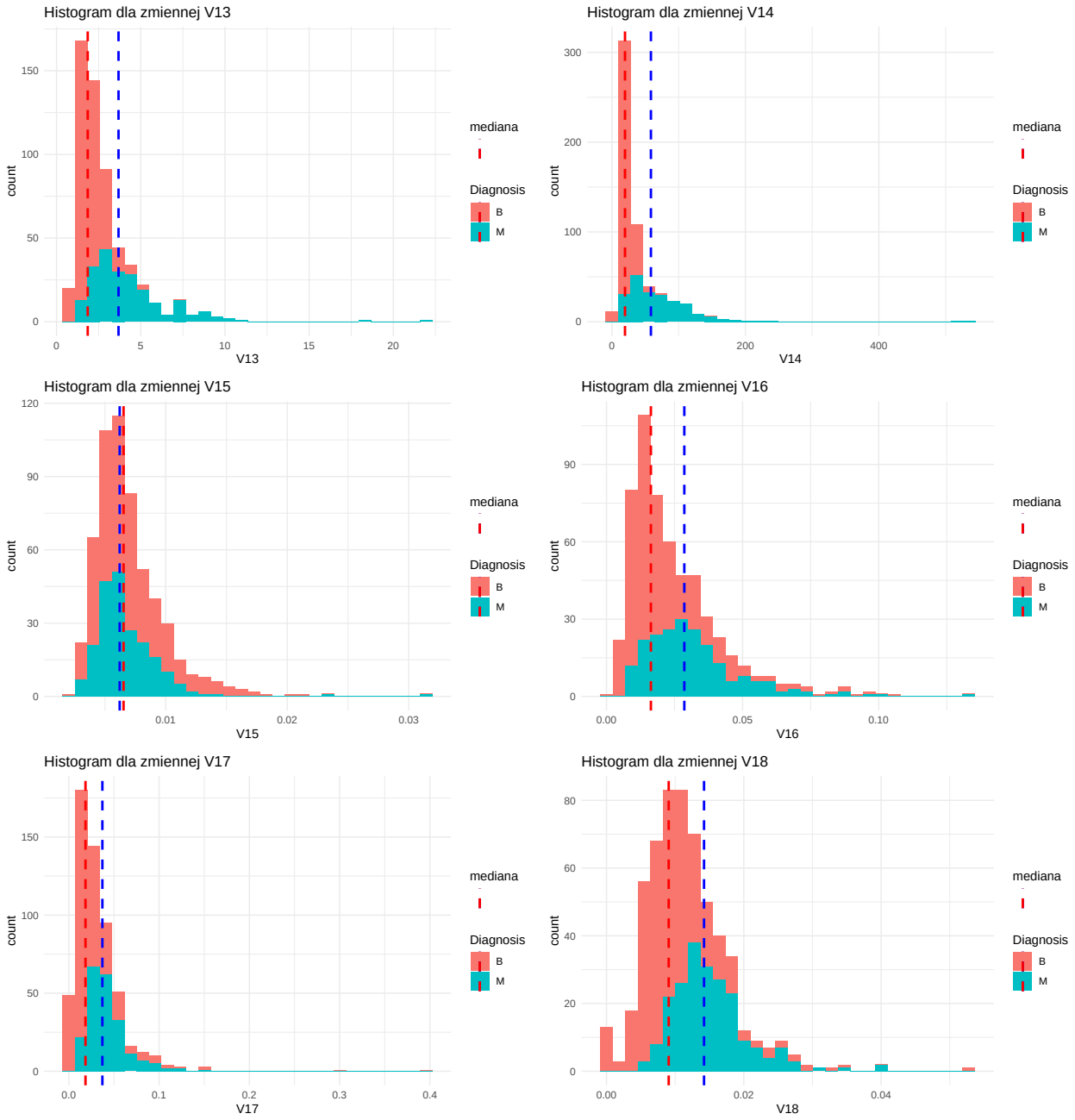
EDA

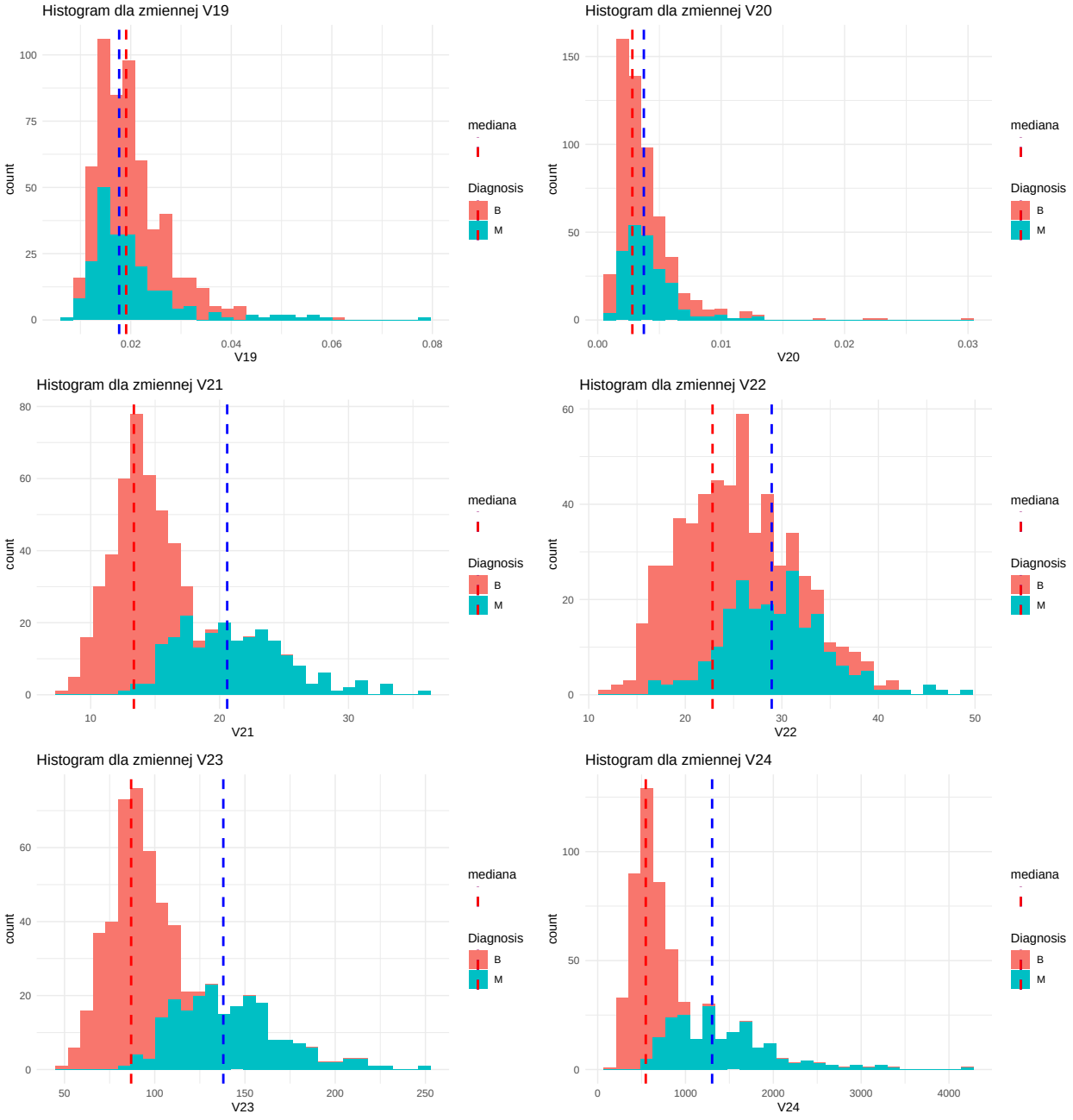
Predyktorów jest sporo, do tego są zakodowane, dlatego nie ma sensu wizualizować relacji 2 zmiennych i więcej. Natomiast histogram dla każdej zmiennej `V` z podziałem na zmienną celu `Diagnosis`, graficznie pokaże różnice między rozkładami wartości zmiennej `Diagnosis` (`B`, `M`) oraz nienormalizując oś Y można wstępnie ocenić czy występują **punkty oddalone**.

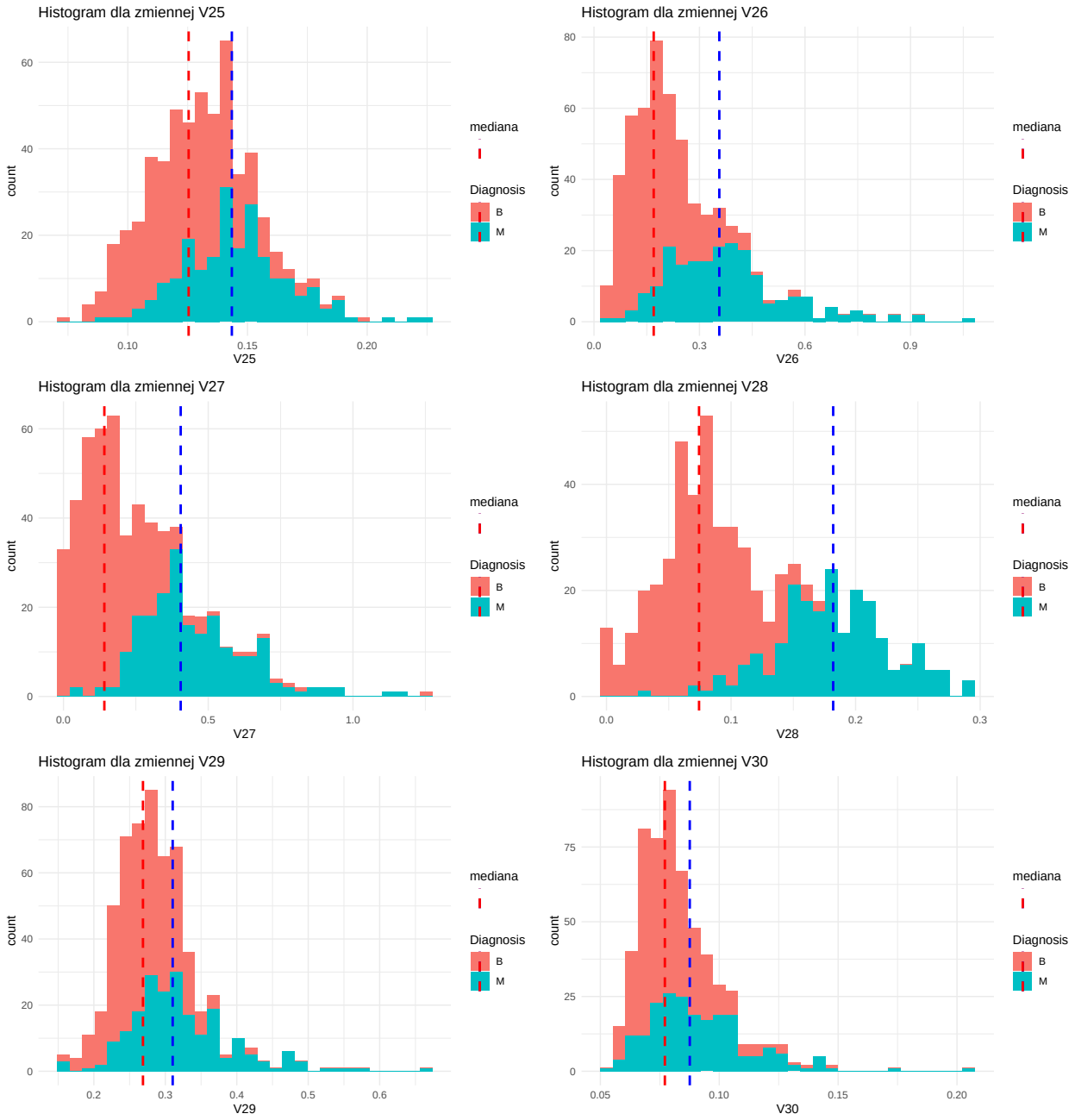
```
for (i in 3:ncol(wdbc)) {
  show(ggplot(wdbc, aes(x= wdbc[,i], fill=Diagnosis)) +
         geom_histogram()+
         ggtitle(paste("Histogram dla zmiennej V", i-2, sep="") )+
         xlab(paste("V", i-2, sep="")) +
         theme_minimal() +
         geom_vline(aes(xintercept=median(wdbc[,i][wdbc$Diagnosis == 'M'])), color='blue',
                    linetype='dashed', size=1, show_guide=T) +
         geom_vline(aes(xintercept=median(wdbc[,i][wdbc$Diagnosis == 'B'])), color='red',
                    linetype='dashed', size=1, show_guide=T) +
         scale_color_manual(name = "mediana", values = c(" " = "black")))
}
```


Histogram dla zmiennej V1
count
V1
mediana
Diagnosis
B
M
Histogram dla zmiennej V2
count
V2
mediana
Diagnosis
B
M
Histogram dla zmiennej V3
count
V3
mediana
Diagnosis
B
M
Histogram dla zmiennej V4
count
V4
mediana
Diagnosis
B
M
Histogram dla zmiennej V5
count
V5
mediana
Diagnosis
B
M
Histogram dla zmiennej V6
count
V6
mediana
Diagnosis
B
M


Histogram dla zmiennej V7
count
V7
mediana
Diagnosis
B
M
Histogram dla zmiennej V8
count
V8
mediana
Diagnosis
B
M
Histogram dla zmiennej V9
count
V9
mediana
Diagnosis
B
M
Histogram dla zmiennej V10
count
V10
mediana
Diagnosis
B
M
Histogram dla zmiennej V11
count
V11
mediana
Diagnosis
B
M
Histogram dla zmiennej V12
count
V12
mediana
Diagnosis
B
M


Histogram dla zmiennej V13
count
V13
mediana
Diagnosis
B
M
Histogram dla zmiennej V14
count
V14
mediana
Diagnosis
B
M
Histogram dla zmiennej V15
count
V15
mediana
Diagnosis
B
M
Histogram dla zmiennej V16
count
V16
mediana
Diagnosis
B
M
Histogram dla zmiennej V17
count
V17
mediana
Diagnosis
B
M
Histogram dla zmiennej V18
count
V18
mediana
Diagnosis
B
M


Histogram dla zmiennej V19
count
V19
mediana
Diagnosis
B
M
Histogram dla zmiennej V20
count
V20
mediana
Diagnosis
B
M
Histogram dla zmiennej V21
count
V21
mediana
Diagnosis
B
M
Histogram dla zmiennej V22
count
V22
mediana
Diagnosis
B
M
Histogram dla zmiennej V23
count
V23
mediana
Diagnosis
B
M
Histogram dla zmiennej V24
count
V24
mediana
Diagnosis
B
M



Na podstawie powyższych histogramów dla każdej zmiennej `V`, można zauważyć, że:

- występują punkty oddalone, co zostanie zweryfikowane jeszcze w dalszych krokach
- dla niektórych zmiennych widać wyraźne różnice w położeniach rozkładów (medianach), zwłaszcza dla `V1`, `V3`, `V4`, `V7`, `V8`, `V21`, `V23`, `V24`, `V26`, `V27`, `V28`, co może świasczyć o tym, że będą istotnymi predyktorami w modelu.

Sprawdzenie czy występuje **współliniowość** wśród zmiennych `V`

```
korelacja <- round(cor(wdbc[,3:ncol(wdbc)]), 2)
as.data.frame(ifelse(korelacja >= 0.7, korelacja, ""))
```

```
##       V1   V2   V3   V4   V5   V6   V7   V8  V9  V10  V11 V12  V13  V14 V15
## V1     1         1 0.99               0.82                       0.74
## V2           1
## V3     1         1 0.99          0.72 0.85                       0.74
## V4  0.99      0.99    1               0.82          0.73     0.73  0.8
## V5                          1
## V6                               1 0.88 0.83
## V7            0.72           0.88    1 0.92
## V8  0.82      0.85 0.82      0.83 0.92    1           0.7     0.71
## V9                                              1
## V10                                                 1
## V11                0.73                0.7             1     0.97 0.95
## V12                                                        1
## V13                0.73                0.71          0.97        1 0.94
## V14 0.74      0.74  0.8                                 0.95     0.94    1
## V15                                                                      1
## V16                          0.74
## V17
## V18
## V19
## V20
## V21 0.97      0.97 0.96               0.83          0.72      0.7 0.76
## V22      0.91
## V23 0.97      0.97 0.96          0.73 0.86          0.72     0.72 0.76
## V24 0.94      0.94 0.96               0.81          0.75     0.73 0.81
## V25                     0.81
## V26                          0.87 0.75
## V27                          0.82 0.88 0.75
## V28 0.74      0.77 0.72      0.82 0.86 0.91
## V29                                        0.7
## V30                                           0.77
##      V16  V17  V18 V19  V20  V21  V22  V23  V24  V25  V26  V27  V28 V29  V30
## V1                          0.97      0.97 0.94               0.74
## V2                               0.91
## V3                          0.97      0.97 0.94               0.77
## V4                          0.96      0.96 0.96               0.72
## V5                                             0.81
## V6  0.74                                            0.87 0.82 0.82
## V7                                     0.73          0.75 0.88 0.86
## V8                          0.83      0.86 0.81          0.75 0.91
## V9                                                                0.7
## V10                                                                   0.77
## V11                         0.72      0.72 0.75
## V12
## V13                          0.7      0.72 0.73
## V14                         0.76      0.76 0.81
## V15
## V16    1  0.8 0.74      0.8
## V17  0.8    1 0.77     0.73
## V18 0.74 0.77    1
## V19                 1
## V20  0.8 0.73            1
## V21                      1      0.99 0.98               0.79
## V22                           1
```

```
## V23                          0.99         1 0.98                0.82
## V24                          0.98      0.98    1                0.75
## V25                                                1
## V26                                                     1 0.89  0.8     0.81
## V27                                                  0.89    1 0.86
## V28                          0.79      0.82 0.75       0.8 0.86    1
## V29                                                                      1
## V30                                                  0.81                    1
```

- Z powyższej macierzy korelacji Pearsona wynika, że występuje silna współliniowość wśród większości zmiennych `V`.
- W dalszych krokach zredukuję liczbę zmiennych na podstawie współczynnika VIF

Szukanie **puntów oddalonych** z wykorzystaniem IQR dla założenia, że punkt x jest oddalony, gdy:

$$(x < Q_1 - 1.5 * IQR) \ \vee \ (x > Q_3 + 1.5 * IQR)$$

```
outliers <- function(x, s, e){

  # x = dataframe
  # s = index of first col to take
  # e = index of last column to take

  p <- x

  for(i in s:e){

    Q1 <- quantile(p[,i], 0.25, names = FALSE)
    Q3 <- quantile(p[,i], 0.75, names = FALSE)
    iqr <- IQR(p[,i])
    low <- Q1 - iqr*1.5
    up <- Q3 + iqr*1.5

    p[,i] <- ((p[,i] < low) | (p[,i] > up))
  }

  p <- p %>% mutate(outliers_numb = rowSums(p[,s:e]))
  x$outliers_numb <- p$outliers_numb

  tot <- sum(x$outliers_numb)
  totr <- nrow(x %>% filter(outliers_numb > 0))
  perc <- (tot*100)/(nrow(x)*ncol(x))
  percr <- (totr*100)/nrow(x)


  print(paste("Total number of outliers:", round(tot, 0)))
  print(paste("% of outliers:", round(perc, 2)))
  print(paste("Total number of rows with outliers:", round(totr, 0)))
  print(paste("% of rows with outliers:", round(percr, 2)))

  print("Rows with outliers:")
  print(x %>% filter(outliers_numb > 0))
```

```
  return(invisible(x))
  }
```

```
outliers(wdbc, s=3, e=ncol(wdbc))
```

```
## [1] "Total number of outliers: 608"
## [1] "% of outliers: 3.24"
## [1] "Total number of rows with outliers: 171"
## [1] "% of rows with outliers: 30.05"
## [1] "Rows with outliers:"
##     ID.number Diagnosis     V1    V2     V3     V4      V5      V6       V7
## 1      842302         M 17.990 10.38 122.80 1001.0 0.11840 0.27760 0.300100
## 2      842517         M 20.570 17.77 132.90 1326.0 0.08474 0.07864 0.086900
## 3    84300903         M 19.690 21.25 130.00 1203.0 0.10960 0.15990 0.197400
## 4    84348301         M 11.420 20.38  77.58  386.1 0.14250 0.28390 0.241400
## 5    84358402         M 20.290 14.34 135.10 1297.0 0.10030 0.13280 0.198000
## 6      843786         M 12.450 15.70  82.57  477.1 0.12780 0.17000 0.157800
## 7      844981         M 13.000 21.82  87.50  519.8 0.12730 0.19320 0.185900
## 8    84501001         M 12.460 24.04  83.97  475.9 0.11860 0.23960 0.227300
## 9      846226         M 19.170 24.80 132.40 1123.0 0.09740 0.24580 0.206500
## 10   84667401         M 13.730 22.61  93.60  578.3 0.11310 0.22930 0.212800
## 11   84799002         M 14.540 27.54  96.73  658.8 0.11390 0.15950 0.163900
## 12     849014         M 19.810 22.15 130.00 1260.0 0.09831 0.10270 0.147900
## 13    8511133         M 15.340 14.26 102.50  704.4 0.10730 0.21350 0.207700
## 14     851509         M 21.160 23.04 137.20 1404.0 0.09428 0.10220 0.109700
## 15     852552         M 16.650 21.38 110.00  904.6 0.11210 0.14570 0.152500
## 16     852631         M 17.140 16.40 116.00  912.7 0.11860 0.22760 0.222900
## 17     852763         M 14.580 21.53  97.41  644.8 0.10540 0.18680 0.142500
## 18     852781         M 18.610 20.25 122.10 1094.0 0.09440 0.10660 0.149000
## 19     853401         M 18.630 25.11 124.80 1088.0 0.10640 0.18870 0.231900
## 20     853612         M 11.840 18.70  77.93  440.6 0.11090 0.15160 0.121800
## 21     854002         M 19.270 26.47 127.90 1162.0 0.09401 0.17190 0.165700
## 22     854039         M 16.130 17.88 107.00  807.2 0.10400 0.15590 0.135400
## 23     854253         M 16.740 21.59 110.10  869.5 0.09610 0.13360 0.134800
## 24     855133         M 14.990 25.20  95.54  698.8 0.09387 0.05131 0.023980
## 25     855563         M 10.950 21.35  71.90  371.1 0.12270 0.12180 0.104400
## 26     855625         M 19.070 24.81 128.30 1104.0 0.09081 0.21900 0.210700
## 27     857392         M 18.220 18.70 120.30 1033.0 0.11480 0.14850 0.177200
## 28     857637         M 19.210 18.57 125.50 1152.0 0.10530 0.12670 0.132300
## 29     858970         B 10.170 14.88  64.55  311.9 0.11340 0.08061 0.010840
## 30     858986         M 14.250 22.15  96.42  645.7 0.10490 0.20080 0.213500
## 31     859196         B  9.173 13.86  59.20  260.9 0.07721 0.08751 0.059880
## 32     859471         B  9.029 17.33  58.79  250.5 0.10660 0.14130 0.313000
## 33     859575         M 18.940 21.31 123.60 1130.0 0.09009 0.10290 0.108000
## 34     859711         B  8.888 14.64  58.79  244.0 0.09783 0.15310 0.086060
## 35     859717         M 17.200 24.52 114.20  929.4 0.10710 0.18300 0.169200
## 36    8610629         B 13.530 10.94  87.91  559.2 0.12910 0.10470 0.068770
## 37    8610637         M 18.050 16.15 120.20 1006.0 0.10650 0.21460 0.168400
## 38    8610862         M 20.180 23.97 143.70 1245.0 0.12860 0.34540 0.375400
## 39    8611555         M 25.220 24.91 171.50 1878.0 0.10630 0.26650 0.333900
## 40    8611792         M 19.100 26.29 129.10 1132.0 0.12150 0.17910 0.193700
## 41      86208         M 20.260 23.03 132.40 1264.0 0.09078 0.13130 0.146500
## 42     863030         M 13.110 15.56  87.21  530.2 0.13980 0.17650 0.207100
```

```
## 43      86355         M 22.270 19.67 152.80 1509.0 0.13260 0.27680 0.426400
## 44     864033         B  9.777 16.99  62.50  290.2 0.10370 0.08404 0.043340
## 45      86408         B 12.630 20.76  82.15  480.4 0.09933 0.12090 0.106500
## 46      86409         B 14.260 19.65  97.83  629.9 0.07837 0.22330 0.300300
## 47     864726         B  8.950 15.76  58.74  245.2 0.09462 0.12430 0.092630
## 48     864877         M 15.780 22.91 105.70  782.6 0.11550 0.17520 0.213300
## 49     865128         M 17.950 20.01 114.20  982.0 0.08402 0.06722 0.072930
## 50      86517         M 18.660 17.12 121.40 1077.0 0.10540 0.11000 0.145700
## 51     865423         M 24.250 20.20 166.20 1761.0 0.14470 0.28670 0.426800
## 52     868223         B 11.710 16.67  74.72  423.6 0.10510 0.06095 0.035920
## 53     868826         M 14.950 17.57  96.85  678.1 0.11670 0.13050 0.153900
## 54     869476         B 11.900 14.65  78.11  432.8 0.11520 0.12960 0.037100
## 55     869691         M 11.800 16.58  78.99  432.0 0.10910 0.17000 0.165900
## 56   86973701         B 14.950 18.77  97.84  689.5 0.08138 0.11670 0.090500
## 57  871001501         B 13.000 20.78  83.51  519.4 0.11350 0.07589 0.031360
## 58  871001502         B  8.219 20.70  53.27  203.9 0.09405 0.13050 0.132100
## 59    8710441         B  9.731 15.34  63.78  300.2 0.10720 0.15990 0.410800
## 60    8711202         M 17.680 20.74 117.40  963.7 0.11150 0.16650 0.185500
## 61    8711803         M 19.190 15.94 126.30 1157.0 0.08694 0.11850 0.119300
## 62     871201         M 19.590 18.15 130.70 1214.0 0.11200 0.16660 0.250800
## 63    8712289         M 23.270 22.04 152.10 1686.0 0.08439 0.11450 0.132400
## 64    8712766         M 17.470 24.68 116.10  984.6 0.10490 0.16030 0.215900
## 65     871641         B 11.080 14.71  70.21  372.7 0.10060 0.05743 0.023630
## 66     872608         B  9.904 18.06  64.60  302.4 0.09699 0.12940 0.130700
## 67     873592         M 27.220 21.87 182.10 2250.0 0.10940 0.19140 0.287100
## 68     873593         M 21.090 26.57 142.70 1311.0 0.11410 0.28320 0.248700
## 69     874158         B 10.080 15.11  63.76  317.5 0.09267 0.04695 0.001597
## 70     874858         M 14.220 23.12  94.37  609.9 0.10750 0.24130 0.198100
## 71     875099         B  9.720 18.22  60.73  288.1 0.06950 0.02344 0.000000
## 72     875938         M 13.770 22.29  90.63  588.9 0.12000 0.12670 0.138500
## 73     877500         M 14.450 20.22  94.49  642.7 0.09872 0.12060 0.118000
## 74     878796         M 23.290 26.67 158.90 1685.0 0.11410 0.20840 0.352300
## 75      87880         M 13.810 23.75  91.56  597.8 0.13230 0.17680 0.155800
## 76  881046502         M 20.580 22.14 134.70 1290.0 0.09090 0.13480 0.164000
## 77    8810703         M 28.110 18.47 188.50 2499.0 0.11420 0.15160 0.320100
## 78  881094802         M 17.420 25.56 114.50  948.0 0.10060 0.11460 0.168200
## 79    8810955         M 14.190 23.81  92.87  610.7 0.09463 0.13060 0.111500
## 80    8811842         M 19.800 21.56 129.70 1230.0 0.09383 0.13060 0.127200
## 81   88119002         M 19.530 32.47 128.00 1223.0 0.08420 0.11300 0.114500
## 82     881861         M 12.830 22.33  85.26  503.2 0.10880 0.17990 0.169500
## 83   88203002         B 11.220 33.81  70.79  386.8 0.07780 0.03574 0.004967
## 84   88299702         M 23.210 26.97 153.50 1670.0 0.09509 0.16820 0.195000
## 85   88330202         M 17.460 39.28 113.40  920.6 0.09812 0.12980 0.141700
## 86     883852         B 11.300 18.19  73.93  389.4 0.09592 0.13250 0.154800
## 87     884437         B 10.480 19.86  66.72  337.7 0.10700 0.05971 0.048310
## 88     884948         M 20.940 23.56 138.90 1364.0 0.10070 0.16060 0.271200
## 89     885429         M 19.730 19.82 130.70 1206.0 0.10620 0.18490 0.241700
## 90     886226         M 19.450 19.33 126.50 1169.0 0.10350 0.11880 0.137900
## 91   88649001         M 19.550 28.77 133.60 1207.0 0.09260 0.20630 0.178400
## 92     886776         M 15.320 17.27 103.20  713.3 0.13350 0.22840 0.244800
## 93     887181         M 15.660 23.20 110.20  773.5 0.11090 0.31140 0.317600
## 94   88725602         M 15.530 33.56 103.70  744.9 0.10630 0.16390 0.175100
## 95     888570         M 17.290 22.13 114.40  947.8 0.08999 0.12730 0.096970
## 96   88995002         M 20.730 31.12 135.70 1419.0 0.09469 0.11430 0.136700
```

```
## 97    8910988         M 21.750 20.99 147.30 1491.0 0.09401 0.19610 0.219500
## 98    8910996         B  9.742 15.67  61.50  289.9 0.09037 0.04689 0.011030
## 99    8911164         B 11.890 17.36  76.20  435.6 0.12250 0.07210 0.059290
## 100   8913049         B 11.260 19.96  73.72  394.1 0.08020 0.11810 0.092740
## 101  89143602         B 14.410 19.73  96.03  651.0 0.08757 0.16760 0.136200
## 102    892438         M 19.530 18.90 129.50 1217.0 0.11500 0.16420 0.219700
## 103  89263202         M 20.090 23.86 134.70 1247.0 0.10800 0.18380 0.228300
## 104    894047         B  8.597 18.60  54.09  221.2 0.10740 0.05847 0.000000
## 105    894329         B  9.042 18.90  60.07  244.5 0.09968 0.19720 0.197500
## 106    895100         M 20.340 21.51 135.90 1264.0 0.11700 0.18750 0.256500
## 107    895633         M 16.260 21.88 107.50  826.8 0.11650 0.12830 0.179900
## 108    897132         B 11.220 19.86  71.94  387.3 0.10540 0.06779 0.005006
## 109  89742801         M 17.060 21.00 111.80  918.6 0.11190 0.10560 0.150800
## 110    897630         M 18.770 21.43 122.90 1092.0 0.09116 0.14020 0.106000
## 111     89812         M 23.510 24.27 155.10 1747.0 0.10690 0.12830 0.230800
## 112    898431         M 19.680 21.68 129.90 1194.0 0.09797 0.13390 0.186300
## 113    898677         B 10.260 14.71  66.20  321.6 0.09882 0.09159 0.035810
## 114    899667         M 15.750 19.22 107.10  758.6 0.12430 0.23640 0.291400
## 115    899987         M 25.730 17.46 174.20 2010.0 0.11490 0.23630 0.336800
## 116   9011494         M 20.200 26.83 133.70 1234.0 0.09905 0.16690 0.164100
## 117   9011971         M 21.710 17.25 140.90 1546.0 0.09384 0.08562 0.116800
## 118   9012000         M 22.010 21.90 147.20 1482.0 0.10630 0.19540 0.244800
## 119   9012315         M 16.350 23.29 109.00  840.4 0.09742 0.14970 0.181100
## 120   9012795         M 21.370 15.10 141.30 1386.0 0.10010 0.15150 0.193200
## 121    901288         M 20.640 17.35 134.80 1335.0 0.09446 0.10760 0.152700
## 122    901315         B 10.570 20.22  70.15  338.3 0.09073 0.16600 0.228000
## 123   9013838         M 11.080 18.83  73.30  361.6 0.12160 0.21540 0.168900
## 124    903011         B 11.270 15.50  73.38  392.0 0.08365 0.11140 0.100700
## 125     90312         M 19.550 23.21 128.90 1174.0 0.10100 0.13180 0.185600
## 126    903483         B  8.734 16.84  55.27  234.3 0.10390 0.07428 0.000000
## 127    903516         M 21.610 22.28 144.40 1407.0 0.11670 0.20870 0.281000
## 128  90439701         M 17.910 21.02 124.40  994.0 0.12300 0.25760 0.318900
## 129    905978         B  9.405 21.70  59.60  271.2 0.10440 0.06159 0.020470
## 130  90602302         M 15.500 21.08 102.90  803.1 0.11200 0.15710 0.152200
## 131    907145         B  9.742 19.12  61.93  289.7 0.10750 0.08333 0.008934
## 132    907914         M 14.900 22.53 102.10  685.0 0.09947 0.22250 0.273300
## 133    908445         M 18.820 21.97 123.70 1110.0 0.10180 0.13890 0.159400
## 134    909777         B 10.570 18.32  66.82  340.9 0.08142 0.04462 0.019930
## 135 911157302         M 21.100 20.52 138.10 1384.0 0.09684 0.11750 0.157200
## 136   9111596         B 11.870 21.54  76.83  432.0 0.06613 0.10640 0.087770
## 137   9112085         B 13.380 30.72  86.34  557.2 0.09245 0.07426 0.028190
## 138 911296201         M 17.080 27.15 111.20  930.9 0.09898 0.11100 0.100700
## 139 911296202         M 27.420 26.27 186.90 2501.0 0.10840 0.19880 0.363500
## 140   9113239         B 13.240 20.13  86.87  542.9 0.08284 0.12230 0.101000
## 141   9113538         M 17.600 23.33 119.00  980.5 0.09289 0.20040 0.213600
## 142    911366         B 11.620 18.18  76.38  408.8 0.11750 0.14830 0.102000
## 143   9113816         B 12.040 28.14  76.85  449.9 0.08752 0.06000 0.023670
## 144   9113846         B 12.270 29.97  77.42  465.4 0.07699 0.03398 0.000000
## 145    913063         B 12.450 16.41  82.85  476.7 0.09514 0.15110 0.154400
## 146    913535         M 16.690 20.20 107.10  857.6 0.07497 0.07112 0.036490
## 147    914062         M 18.010 20.56 118.40 1007.0 0.10010 0.12890 0.117000
## 148    914769         M 18.490 17.52 121.30 1068.0 0.10120 0.13170 0.149100
## 149    915143         M 23.090 19.83 152.10 1682.0 0.09342 0.12750 0.167600
## 150    915186         B  9.268 12.87  61.49  248.7 0.16340 0.22390 0.097300
```

```
## 151    915276         B  9.676 13.14  64.12  272.5 0.12550 0.22040 0.118800
## 152  91544002         B 11.060 17.12  71.25  366.5 0.11940 0.10710 0.040630
## 153    917092         B  9.295 13.90  59.96  257.8 0.13710 0.12250 0.033320
## 154  91762702         M 24.630 21.60 165.50 1841.0 0.10300 0.21060 0.231000
## 155    918192         B 13.940 13.17  90.31  594.2 0.12480 0.09755 0.101000
## 156  91930402         M 20.470 20.67 134.70 1299.0 0.09156 0.13130 0.152300
## 157    919555         M 20.550 20.86 137.80 1308.0 0.10460 0.17390 0.208500
## 158    919812         B 11.690 24.44  76.37  406.4 0.12360 0.15520 0.045150
## 159    921092         B  7.729 25.49  47.98  178.8 0.08098 0.04878 0.000000
## 160    921362         B  7.691 25.44  48.34  170.4 0.08668 0.11990 0.092520
## 161    924342         B  9.333 21.94  59.01  264.0 0.09240 0.05605 0.039960
## 162    924964         B 10.160 19.59  64.73  311.7 0.10030 0.07504 0.005025
## 163    925236         B  9.423 27.88  59.26  271.3 0.08123 0.04971 0.000000
## 164    925291         B 11.510 23.93  74.52  403.5 0.09261 0.10210 0.111200
## 165    925311         B 11.200 29.37  70.67  386.0 0.07449 0.03558 0.000000
## 166    925622         M 15.220 30.62 103.40  716.9 0.10480 0.20870 0.255000
## 167    926125         M 20.920 25.09 143.00 1347.0 0.10990 0.22360 0.317400
## 168    926424         M 21.560 22.39 142.00 1479.0 0.11100 0.11590 0.243900
## 169    926682         M 20.130 28.25 131.20 1261.0 0.09780 0.10340 0.144000
## 170    927241         M 20.600 29.33 140.10 1265.0 0.11780 0.27700 0.351400
## 171     92751         B  7.760 24.54  47.92  181.0 0.05263 0.04362 0.000000
##           V8     V9     V10    V11    V12    V13     V14      V15      V16
## 1   0.147100 0.2419 0.07871 1.0950 0.9053  8.589 153.400 0.006399 0.049040
## 2   0.070170 0.1812 0.05667 0.5435 0.7339  3.398  74.080 0.005225 0.013080
## 3   0.127900 0.2069 0.05999 0.7456 0.7869  4.585  94.030 0.006150 0.040060
## 4   0.105200 0.2597 0.09744 0.4956 1.1560  3.445  27.230 0.009110 0.074580
## 5   0.104300 0.1809 0.05883 0.7572 0.7813  5.438  94.440 0.011490 0.024610
## 6   0.080890 0.2087 0.07613 0.3345 0.8902  2.217  27.190 0.007510 0.033450
## 7   0.093530 0.2350 0.07389 0.3063 1.0020  2.406  24.320 0.005731 0.035020
## 8   0.085430 0.2030 0.08243 0.2976 1.5990  2.039  23.940 0.007149 0.072170
## 9   0.111800 0.2397 0.07800 0.9555 3.5680 11.070 116.200 0.003139 0.082970
## 10  0.080250 0.2069 0.07682 0.2121 1.1690  2.061  19.210 0.006429 0.059360
## 11  0.073640 0.2303 0.07077 0.3700 1.0330  2.879  32.550 0.005607 0.042400
## 12  0.094980 0.1582 0.05395 0.7582 1.0170  5.865 112.400 0.006494 0.018930
## 13  0.097560 0.2521 0.07032 0.4388 0.7096  3.384  44.910 0.006789 0.053280
## 14  0.086320 0.1769 0.05278 0.6917 1.1270  4.303  93.990 0.004728 0.012590
## 15  0.091700 0.1995 0.06330 0.8068 0.9017  5.455 102.600 0.006048 0.018820
## 16  0.140100 0.3040 0.07413 1.0460 0.9760  7.276 111.400 0.008029 0.037990
## 17  0.087830 0.2252 0.06924 0.2545 0.9832  2.110  21.050 0.004452 0.030550
## 18  0.077310 0.1697 0.05699 0.8529 1.8490  5.632  93.540 0.010750 0.027220
## 19  0.124400 0.2183 0.06197 0.8307 1.4660  5.574 105.000 0.006248 0.033740
## 20  0.051820 0.2301 0.07799 0.4825 1.0300  3.475  41.000 0.005551 0.034140
## 21  0.075930 0.1853 0.06261 0.5558 0.6062  3.528  68.170 0.005015 0.033180
## 22  0.077520 0.1998 0.06515 0.3340 0.6857  2.183  35.030 0.004185 0.028680
## 23  0.060180 0.1896 0.05656 0.4615 0.9197  3.008  45.190 0.005776 0.024990
## 24  0.028990 0.1565 0.05504 1.2140 2.1880  8.077 106.000 0.006883 0.010940
## 25  0.056690 0.1895 0.06870 0.2366 1.4280  1.822  16.970 0.008064 0.017640
## 26  0.099610 0.2310 0.06343 0.9811 1.6660  8.830 104.900 0.006548 0.100600
## 27  0.106000 0.2092 0.06310 0.8337 1.5930  4.877  98.810 0.003899 0.029610
## 28  0.089940 0.1917 0.05961 0.7275 1.1930  4.837 102.500 0.006458 0.023060
## 29  0.012900 0.2743 0.06960 0.5158 1.4410  3.312  34.620 0.007514 0.010990
## 30  0.086530 0.1949 0.07292 0.7036 1.2680  5.373  60.780 0.009407 0.070560
## 31  0.021800 0.2341 0.06963 0.4098 2.2650  2.608  23.520 0.008738 0.039380
## 32  0.043750 0.2111 0.08046 0.3274 1.1940  1.885  17.670 0.009549 0.086060
```

```
## 33  0.079510 0.1582 0.05461 0.7888 0.7975  5.486  96.050 0.004444 0.016520
## 34  0.028720 0.1902 0.08980 0.5262 0.8522  3.168  25.440 0.017210 0.093680
## 35  0.079440 0.1927 0.06487 0.5907 1.0410  3.705  69.470 0.005820 0.056160
## 36  0.065560 0.2403 0.06641 0.4101 1.0140  2.652  32.650 0.013400 0.028390
## 37  0.108000 0.2152 0.06673 0.9806 0.5505  6.311 134.800 0.007940 0.058390
## 38  0.160400 0.2906 0.08142 0.9317 1.8850  8.649 116.400 0.010380 0.068350
## 39  0.184500 0.1829 0.06782 0.8973 1.4740  7.382 120.000 0.008166 0.056930
## 40  0.146900 0.1634 0.07224 0.5190 2.9100  5.801  67.100 0.007545 0.060500
## 41  0.086830 0.2095 0.05649 0.7576 1.5090  4.554  87.870 0.006016 0.034820
## 42  0.096010 0.1925 0.07692 0.3908 0.9238  2.410  34.660 0.007162 0.029120
## 43  0.182300 0.2556 0.07039 1.2150 1.5450 10.050 170.000 0.006515 0.086680
## 44  0.017780 0.1584 0.07065 0.4030 1.4240  2.747  22.870 0.013850 0.029320
## 45  0.060210 0.1735 0.07070 0.3424 1.8030  2.711  20.480 0.012910 0.040420
## 46  0.077980 0.1704 0.07769 0.3628 1.4900  3.399  29.250 0.005298 0.074460
## 47  0.023080 0.1305 0.07163 0.3132 0.9789  3.280  16.940 0.018350 0.067600
## 48  0.094790 0.2096 0.07331 0.5520 1.0720  3.598  58.630 0.008699 0.039760
## 49  0.055960 0.2129 0.05025 0.5506 1.2140  3.357  54.040 0.004024 0.008422
## 50  0.086650 0.1966 0.06213 0.7128 1.5810  4.895  90.470 0.008102 0.021010
## 51  0.201200 0.2655 0.06877 1.5090 3.1200  9.807 233.000 0.023330 0.098060
## 52  0.026000 0.1339 0.05945 0.4489 2.5080  3.258  34.370 0.006578 0.013800
## 53  0.086240 0.1957 0.06216 1.2960 1.4520  8.419 101.900 0.010000 0.034800
## 54  0.030030 0.1995 0.07839 0.3962 0.6538  3.021  25.030 0.010170 0.047410
## 55  0.074150 0.2678 0.07371 0.3197 1.4260  2.281  24.720 0.005427 0.036330
## 56  0.035620 0.1744 0.06493 0.4220 1.9090  3.271  39.430 0.005790 0.048770
## 57  0.026450 0.2540 0.06087 0.4202 1.3220  2.873  34.780 0.007017 0.011420
## 58  0.021680 0.2222 0.08261 0.1935 1.9620  1.243  10.210 0.012430 0.054160
## 59  0.078570 0.2548 0.09296 0.8245 2.6640  4.073  49.850 0.010970 0.095860
## 60  0.105400 0.1971 0.06166 0.8113 1.4000  5.540  93.910 0.009037 0.049540
## 61  0.096670 0.1741 0.05176 1.0000 0.6336  6.971 119.300 0.009406 0.030550
## 62  0.128600 0.2027 0.06082 0.7364 1.0480  4.792  97.070 0.004057 0.022770
## 63  0.097020 0.1801 0.05553 0.6642 0.8561  4.603  97.850 0.004910 0.025440
## 64  0.104300 0.1538 0.06365 1.0880 1.4100  7.337 122.300 0.006174 0.036340
## 65  0.025830 0.1566 0.06669 0.2073 1.8050  1.377  19.080 0.014960 0.021210
## 66  0.037160 0.1669 0.08116 0.4311 2.2610  3.132  27.480 0.012860 0.088080
## 67  0.187800 0.1800 0.05770 0.8361 1.4810  5.820 128.700 0.004631 0.025370
## 68  0.149600 0.2395 0.07398 0.6298 0.7629  4.414  81.460 0.004253 0.047590
## 69  0.002404 0.1703 0.06048 0.4245 1.2680  2.680  26.430 0.014390 0.012000
## 70  0.066180 0.2384 0.07542 0.2860 2.1100  2.112  31.720 0.007970 0.135400
## 71  0.000000 0.1653 0.06447 0.3539 4.8850  2.230  21.690 0.001713 0.006736
## 72  0.065260 0.1834 0.06877 0.6191 2.1120  4.906  49.700 0.013800 0.033480
## 73  0.059800 0.1950 0.06466 0.2092 0.6509  1.446  19.420 0.004044 0.015970
## 74  0.162000 0.2200 0.06229 0.5539 1.5600  4.667  83.160 0.009327 0.051210
## 75  0.091760 0.2251 0.07421 0.5648 1.9300  3.909  52.720 0.008824 0.031080
## 76  0.095610 0.1765 0.05024 0.8601 1.4800  7.029 111.700 0.008124 0.036110
## 77  0.159500 0.1648 0.05525 2.8730 1.4760 21.980 525.600 0.013450 0.027720
## 78  0.065970 0.1308 0.05866 0.5296 1.6670  3.767  58.530 0.031130 0.085550
## 79  0.064620 0.2235 0.06433 0.4207 1.8450  3.534  31.000 0.010880 0.037100
## 80  0.086910 0.2094 0.05581 0.9553 1.1860  6.487 124.400 0.006804 0.031690
## 81  0.066370 0.1428 0.05313 0.7392 1.3210  4.722 109.900 0.005539 0.026440
## 82  0.068610 0.2123 0.07254 0.3061 1.0690  2.257  25.130 0.006983 0.038580
## 83  0.006434 0.1845 0.05828 0.2239 1.6470  1.489  15.460 0.004359 0.006813
## 84  0.123700 0.1909 0.06309 1.0580 0.9635  7.247 155.800 0.006428 0.028630
## 85  0.088110 0.1809 0.05966 0.5366 0.8561  3.002  49.000 0.004860 0.027850
## 86  0.028540 0.2054 0.07669 0.2428 1.6420  2.369  16.390 0.006663 0.059140
```

```
## 87  0.030700 0.1737 0.06440 0.3719 2.6120  2.517  23.220 0.016040 0.013860
## 88  0.131000 0.2205 0.05898 1.0040 0.8208  6.372 137.900 0.005283 0.039080
## 89  0.097400 0.1733 0.06697 0.7661 0.7800  4.115  92.810 0.008482 0.050570
## 90  0.085910 0.1776 0.05647 0.5959 0.6342  3.797  71.000 0.004649 0.018000
## 91  0.114400 0.1893 0.06232 0.8426 1.1990  7.158 106.400 0.006356 0.047650
## 92  0.124200 0.2398 0.07596 0.6592 1.0590  4.061  59.460 0.010150 0.045880
## 93  0.137700 0.2495 0.08104 1.2920 2.4540 10.120 138.500 0.012360 0.059950
## 94  0.083990 0.2091 0.06650 0.2419 1.2780  1.903  23.020 0.005345 0.025560
## 95  0.075070 0.2108 0.05464 0.8348 1.6330  6.146  90.940 0.006717 0.059810
## 96  0.086460 0.1769 0.05674 1.1720 1.6170  7.749 199.700 0.004551 0.014780
## 97  0.108800 0.1721 0.06194 1.1670 1.3520  8.867 156.800 0.005687 0.049600
## 98  0.014070 0.2081 0.06312 0.2684 1.4090  1.750  16.390 0.013800 0.010670
## 99  0.074040 0.2015 0.05875 0.6412 2.2930  4.021  48.840 0.014180 0.014890
## 100 0.055880 0.2595 0.06233 0.4866 1.9050  2.877  34.680 0.015740 0.082620
## 101 0.066020 0.1714 0.07192 0.8811 1.7700  4.360  77.110 0.007762 0.106400
## 102 0.106200 0.1792 0.06552 1.1110 1.1610  7.237 133.000 0.006056 0.032030
## 103 0.128000 0.2249 0.07469 1.0720 1.7430  7.804 130.800 0.007964 0.047320
## 104 0.000000 0.2163 0.07359 0.3368 2.7770  2.222  17.810 0.020750 0.014030
## 105 0.049080 0.2330 0.08743 0.4653 1.9110  3.769  24.200 0.009845 0.065900
## 106 0.150400 0.2569 0.06670 0.5702 1.0230  4.012  69.060 0.005485 0.024310
## 107 0.079810 0.1869 0.06532 0.5706 1.4570  2.961  57.720 0.010560 0.037560
## 108 0.007583 0.1940 0.06028 0.2976 1.9660  1.959  19.620 0.012890 0.011040
## 109 0.099340 0.1727 0.06071 0.8161 2.1290  6.076  87.170 0.006455 0.017970
## 110 0.060900 0.1953 0.06083 0.6422 1.5300  4.369  88.250 0.007548 0.038970
## 111 0.141000 0.1797 0.05506 1.0090 0.9245  6.462 164.100 0.006292 0.019710
## 112 0.110300 0.2082 0.05715 0.6226 2.2840  5.173  67.660 0.004756 0.033680
## 113 0.020370 0.1633 0.07005 0.3380 2.5090  2.394  19.330 0.017360 0.046710
## 114 0.124200 0.2375 0.07603 0.5204 1.3240  3.477  51.220 0.009329 0.065590
## 115 0.191300 0.1956 0.06121 0.9948 0.8509  7.222 153.100 0.006369 0.042430
## 116 0.126500 0.1875 0.06020 0.9761 1.8920  7.128 103.600 0.008439 0.046740
## 117 0.084650 0.1717 0.05054 1.2070 1.0510  7.733 224.100 0.005568 0.011120
## 118 0.150100 0.1824 0.06140 1.0080 0.6999  7.561 130.200 0.003978 0.028210
## 119 0.087730 0.2175 0.06218 0.4312 1.0220  2.972  45.500 0.005635 0.039170
## 120 0.125500 0.1973 0.06183 0.3414 1.3090  2.407  39.060 0.004426 0.026750
## 121 0.089410 0.1571 0.05478 0.6137 0.6575  4.119  77.020 0.006211 0.018950
## 122 0.059410 0.2188 0.08450 0.1115 1.2310  2.363   7.228 0.008499 0.076430
## 123 0.063670 0.2196 0.07950 0.2114 1.0270  1.719  13.990 0.007405 0.045490
## 124 0.027570 0.1810 0.07252 0.3305 1.0670  2.569  22.970 0.010380 0.066690
## 125 0.102100 0.1989 0.05884 0.6107 2.8360  5.383  70.100 0.011240 0.040970
## 126 0.000000 0.1985 0.07098 0.5169 2.0790  3.167  28.850 0.015820 0.019660
## 127 0.156200 0.2162 0.06606 0.6242 0.9209  4.158  80.990 0.005215 0.037260
## 128 0.119800 0.2113 0.07115 0.4030 0.7747  3.123  41.510 0.007159 0.037180
## 129 0.012570 0.2025 0.06601 0.4302 2.8780  2.759  25.170 0.014740 0.016740
## 130 0.084810 0.2085 0.06864 1.3700 1.2130  9.424 176.500 0.008198 0.038890
## 131 0.019670 0.2538 0.07029 0.6965 1.7470  4.607  43.520 0.013070 0.018850
## 132 0.097110 0.2041 0.06898 0.2530 0.8749  3.466  24.190 0.006965 0.062130
## 133 0.087440 0.1943 0.06132 0.8191 1.9310  4.493 103.900 0.008074 0.040880
## 134 0.011110 0.2372 0.05768 0.1818 2.5420  1.277  13.120 0.010720 0.013310
## 135 0.115500 0.1554 0.05661 0.6643 1.3610  4.542  81.890 0.005467 0.020750
## 136 0.023860 0.1349 0.06612 0.2560 1.5540  1.955  20.240 0.006854 0.060630
## 137 0.032640 0.1375 0.06016 0.3408 1.9240  2.287  28.930 0.005841 0.012460
## 138 0.064310 0.1793 0.06281 0.9291 1.1520  6.051 115.200 0.008740 0.022190
## 139 0.168900 0.2061 0.05623 2.5470 1.3060 18.650 542.200 0.007650 0.053740
## 140 0.028330 0.1601 0.06432 0.2810 0.8135  3.369  23.810 0.004929 0.066570
```

```
## 141 0.100200 0.1696 0.07369 0.9289 1.4650  5.801 104.900 0.006766 0.070250
## 142 0.055640 0.1957 0.07255 0.4101 1.7400  3.027  27.850 0.014590 0.032060
## 143 0.023770 0.1854 0.05698 0.6061 2.6430  4.099  44.960 0.007517 0.015550
## 144 0.000000 0.1701 0.05960 0.4455 3.6470  2.884  35.130 0.007339 0.008243
## 145 0.048460 0.2082 0.07325 0.3921 1.2070  5.004  30.190 0.007234 0.074710
## 146 0.023070 0.1846 0.05325 0.2473 0.5679  1.775  22.950 0.002667 0.014460
## 147 0.077620 0.2116 0.06077 0.7548 1.2880  5.353  89.740 0.007997 0.027000
## 148 0.091830 0.1832 0.06697 0.7923 1.0450  4.851  95.770 0.007974 0.032140
## 149 0.100300 0.1505 0.05484 1.2910 0.7452  9.635 180.200 0.005753 0.033560
## 150 0.052520 0.2378 0.09502 0.4076 1.0930  3.014  20.040 0.009783 0.045420
## 151 0.070380 0.2057 0.09575 0.2744 1.3900  1.787  17.670 0.021770 0.048880
## 152 0.042680 0.1954 0.07976 0.1779 1.0300  1.318  12.300 0.012620 0.023480
## 153 0.024210 0.2197 0.07696 0.3538 1.1300  2.388  19.630 0.015460 0.025400
## 154 0.147100 0.1991 0.06739 0.9915 0.9004  7.050 139.900 0.004989 0.032120
## 155 0.066150 0.1976 0.06457 0.5461 2.6350  4.091  44.740 0.010040 0.032470
## 156 0.101500 0.2166 0.05419 0.8336 1.7360  5.168 100.400 0.004938 0.030890
## 157 0.132200 0.2127 0.06251 0.6986 0.9901  4.706  87.780 0.004578 0.026160
## 158 0.045310 0.2131 0.07405 0.2957 1.9780  2.158  20.950 0.012880 0.034950
## 159 0.000000 0.1870 0.07285 0.3777 1.4620  2.492  19.140 0.012660 0.009692
## 160 0.013640 0.2037 0.07751 0.2196 1.4790  1.445  11.730 0.015470 0.064570
## 161 0.012820 0.1692 0.06576 0.3013 1.8790  2.121  17.860 0.010940 0.018340
## 162 0.011160 0.1791 0.06331 0.2441 2.0900  1.648  16.800 0.012910 0.022220
## 163 0.000000 0.1742 0.06059 0.5375 2.9270  3.618  29.110 0.011590 0.011240
## 164 0.041050 0.1388 0.06570 0.2388 2.9040  1.936  16.970 0.008200 0.029820
## 165 0.000000 0.1060 0.05502 0.3141 3.8960  2.041  22.810 0.007594 0.008878
## 166 0.094290 0.2128 0.07152 0.2602 1.2050  2.362  22.650 0.004625 0.048440
## 167 0.147400 0.2149 0.06879 0.9622 1.0260  8.758 118.800 0.006399 0.043100
## 168 0.138900 0.1726 0.05623 1.1760 1.2560  7.673 158.700 0.010300 0.028910
## 169 0.097910 0.1752 0.05533 0.7655 2.4630  5.203  99.040 0.005769 0.024230
## 170 0.152000 0.2397 0.07016 0.7260 1.5950  5.772  86.220 0.006522 0.061580
## 171 0.000000 0.1587 0.05884 0.3857 1.4280  2.548  19.150 0.007189 0.004660
##          V17      V18      V19      V20    V21   V22    V23    V24     V25
## 1   0.053730 0.015870 0.030030 0.006193 25.380 17.33 184.60 2019.0 0.16220
## 2   0.018600 0.013400 0.013890 0.003532 24.990 23.41 158.80 1956.0 0.12380
## 3   0.038320 0.020580 0.022500 0.004571 23.570 25.53 152.50 1709.0 0.14440
## 4   0.056610 0.018670 0.059630 0.009208 14.910 26.50  98.87  567.7 0.20980
## 5   0.056880 0.018850 0.017560 0.005115 22.540 16.67 152.20 1575.0 0.13740
## 6   0.036720 0.011370 0.021650 0.005082 15.470 23.75 103.40  741.6 0.17910
## 7   0.035530 0.012260 0.021430 0.003749 15.490 30.73 106.20  739.3 0.17030
## 8   0.077430 0.014320 0.017890 0.010080 15.090 40.68  97.65  711.4 0.18530
## 9   0.088900 0.040900 0.044840 0.012840 20.960 29.94 151.70 1332.0 0.10370
## 10  0.055010 0.016280 0.019610 0.008093 15.030 32.01 108.80  697.7 0.16510
## 11  0.047410 0.010900 0.018570 0.005466 17.460 37.13 124.10  943.2 0.16780
## 12  0.033910 0.015210 0.013560 0.001997 27.320 30.88 186.80 2398.0 0.15120
## 13  0.064460 0.022520 0.036720 0.004394 18.070 19.08 125.10  980.9 0.13900
## 14  0.017150 0.010380 0.010830 0.001987 29.170 35.59 188.00 2615.0 0.14010
## 15  0.027410 0.011300 0.014680 0.002801 26.460 31.56 177.00 2215.0 0.18050
## 16  0.037320 0.023970 0.023080 0.007444 22.250 21.40 152.40 1461.0 0.15450
## 17  0.026810 0.013520 0.014540 0.003711 17.620 33.21 122.40  896.9 0.15250
## 18  0.050810 0.019110 0.022930 0.004217 21.310 27.26 139.90 1403.0 0.13380
## 19  0.051960 0.011580 0.020070 0.004560 23.150 34.01 160.50 1670.0 0.14910
## 20  0.042050 0.010440 0.022730 0.005667 16.820 28.12 119.40  888.7 0.16370
## 21  0.034970 0.009643 0.015430 0.003896 24.150 30.90 161.40 1813.0 0.15090
## 22  0.026640 0.009067 0.017030 0.003817 20.210 27.26 132.70 1261.0 0.14460
```

```
## 23  0.036950 0.011950 0.027890 0.002665 20.010 29.02 133.50 1229.0 0.15630
## 24  0.018180 0.019170 0.007882 0.001754 14.990 25.20  95.54  698.8 0.09387
## 25  0.025950 0.010370 0.013570 0.003040 12.840 35.34  87.22  514.0 0.19090
## 26  0.097230 0.026380 0.053330 0.007646 24.090 33.17 177.40 1651.0 0.12470
## 27  0.028170 0.009222 0.026740 0.005126 20.600 24.13 135.10 1321.0 0.12800
## 28  0.029450 0.015380 0.018520 0.002608 26.140 28.14 170.10 2145.0 0.16240
## 29  0.007665 0.008193 0.041830 0.005953 11.020 17.45  69.86  368.6 0.12750
## 30  0.068990 0.018480 0.017000 0.006113 17.670 29.51 119.10  959.5 0.16400
## 31  0.043120 0.015600 0.041920 0.005822 10.010 19.23  65.59  310.1 0.09836
## 32  0.303800 0.033220 0.041970 0.009559 10.310 22.65  65.50  324.7 0.14820
## 33  0.022690 0.013700 0.013860 0.001698 24.860 26.58 165.90 1866.0 0.11930
## 34  0.056710 0.017660 0.025410 0.021930  9.733 15.67  62.56  284.4 0.12070
## 35  0.042520 0.011270 0.015270 0.006299 23.320 33.82 151.60 1681.0 0.15850
## 36  0.011620 0.008239 0.025720 0.006164 14.080 12.49  91.36  605.5 0.14510
## 37  0.046580 0.020700 0.025910 0.007054 22.390 18.91 150.10 1610.0 0.14780
## 38  0.109100 0.025930 0.078950 0.005987 23.370 31.72 170.30 1623.0 0.16390
## 39  0.057300 0.020300 0.010650 0.005893 30.000 33.62 211.70 2562.0 0.15730
## 40  0.021340 0.018430 0.030560 0.010390 20.330 32.72 141.30 1298.0 0.13920
## 41  0.042320 0.012690 0.026570 0.004411 24.220 31.59 156.10 1750.0 0.11900
## 42  0.054730 0.013880 0.015470 0.007098 16.310 22.40 106.40  827.2 0.18620
## 43  0.104000 0.024800 0.031120 0.005037 28.400 28.01 206.80 2360.0 0.17010
## 44  0.027220 0.010230 0.032810 0.004638 11.050 21.47  71.68  367.0 0.14670
## 45  0.051010 0.022950 0.021440 0.005891 13.330 25.47  89.00  527.4 0.12870
## 46  0.143500 0.022920 0.025660 0.012980 15.300 23.73 107.00  709.0 0.08949
## 47  0.092630 0.023080 0.023840 0.005601  9.414 17.07  63.34  270.0 0.11790
## 48  0.059500 0.013900 0.014950 0.005984 20.190 30.50 130.30 1272.0 0.18550
## 49  0.022910 0.009863 0.050140 0.001902 20.580 27.83 129.20 1261.0 0.10720
## 50  0.033420 0.016010 0.020450 0.004570 22.250 24.90 145.40 1549.0 0.15030
## 51  0.127800 0.018220 0.045470 0.009875 26.020 23.99 180.90 2073.0 0.16960
## 52  0.026620 0.013070 0.013590 0.003707 13.330 25.48  86.16  546.7 0.12710
## 53  0.065770 0.028010 0.051680 0.002887 18.550 21.43 121.40  971.4 0.14110
## 54  0.027890 0.011100 0.031270 0.009423 13.150 16.51  86.26  509.6 0.14240
## 55  0.046490 0.018430 0.056280 0.004635 13.740 26.38  91.93  591.7 0.13850
## 56  0.053030 0.015270 0.033560 0.009368 16.250 25.47 107.10  809.7 0.09970
## 57  0.019490 0.011530 0.029510 0.001533 14.160 24.11  90.82  616.7 0.12970
## 58  0.077530 0.010220 0.023090 0.011780  9.092 29.72  58.08  249.8 0.16300
## 59  0.396000 0.052790 0.035460 0.029840 11.020 19.49  71.04  380.5 0.12920
## 60  0.052060 0.018410 0.017780 0.004968 20.470 25.11 132.90 1302.0 0.14180
## 61  0.043440 0.027940 0.031560 0.003362 22.030 17.81 146.60 1495.0 0.11240
## 62  0.040290 0.013030 0.016860 0.003318 26.730 26.39 174.90 2232.0 0.14380
## 63  0.028220 0.016230 0.019560 0.003740 28.010 28.22 184.20 2403.0 0.12280
## 64  0.046440 0.015690 0.011450 0.005120 23.140 32.33 155.30 1660.0 0.13760
## 65  0.014530 0.015830 0.030820 0.004785 11.350 16.82  72.01  396.5 0.12160
## 66  0.119700 0.024600 0.038800 0.017920 11.260 24.39  73.07  390.2 0.13010
## 67  0.031090 0.012410 0.015750 0.002747 33.120 32.85 220.80 3216.0 0.14720
## 68  0.038720 0.015670 0.017980 0.005295 26.680 33.48 176.50 2089.0 0.14910
## 69  0.001597 0.002404 0.025380 0.003470 11.870 21.18  75.39  437.0 0.15210
## 70  0.116600 0.016660 0.051130 0.011720 15.740 37.18 106.40  762.4 0.15330
## 71  0.000000 0.000000 0.037990 0.001688  9.968 20.83  62.25  303.8 0.07117
## 72  0.046650 0.020600 0.026890 0.004306 16.390 34.01 111.60  806.9 0.17370
## 73  0.020000 0.007303 0.015220 0.001976 18.330 30.12 117.90 1044.0 0.15520
## 74  0.089580 0.024650 0.021750 0.005195 25.120 32.68 177.00 1986.0 0.15360
## 75  0.031120 0.012910 0.019980 0.004506 19.200 41.85 128.50 1153.0 0.22260
## 76  0.054890 0.027650 0.031760 0.002365 23.240 27.84 158.30 1656.0 0.11780
```

```
## 77  0.063890 0.014070 0.047830 0.004476 28.110 18.47 188.50 2499.0 0.11420
## 78  0.143800 0.039270 0.021750 0.012560 18.070 28.07 120.40 1021.0 0.12430
## 79  0.036880 0.016270 0.044990 0.004768 16.860 34.85 115.00  811.3 0.15590
## 80  0.034460 0.017120 0.018970 0.004045 25.730 28.64 170.30 2009.0 0.13530
## 81  0.026640 0.010780 0.013320 0.002256 27.900 45.41 180.20 2477.0 0.14080
## 82  0.046830 0.014990 0.016800 0.005617 15.200 30.15 105.30  706.0 0.17770
## 83  0.003223 0.003419 0.019160 0.002534 12.360 41.78  78.44  470.9 0.09994
## 84  0.044970 0.017160 0.015900 0.003053 31.010 34.51 206.00 2944.0 0.14810
## 85  0.026020 0.013740 0.012260 0.002759 22.510 44.87 141.20 1408.0 0.13650
## 86  0.088800 0.013140 0.019950 0.008675 12.580 27.96  87.16  472.9 0.13470
## 87  0.018650 0.011330 0.034760 0.003560 11.480 29.46  73.68  402.8 0.15150
## 88  0.095180 0.018640 0.024010 0.005002 25.580 27.00 165.30 2010.0 0.12110
## 89  0.068000 0.019710 0.014670 0.007259 25.280 25.59 159.80 1933.0 0.17100
## 90  0.027490 0.012670 0.013650 0.002550 25.700 24.57 163.10 1972.0 0.14970
## 91  0.038630 0.015190 0.019360 0.005252 25.050 36.27 178.60 1926.0 0.12810
## 92  0.049830 0.021270 0.018840 0.008660 17.730 22.66 119.80  928.8 0.17650
## 93  0.082320 0.030240 0.023370 0.006042 19.850 31.64 143.70 1226.0 0.15040
## 94  0.028890 0.010220 0.009947 0.003359 18.490 49.54 126.30 1035.0 0.18830
## 95  0.046380 0.021490 0.027470 0.005838 20.390 27.24 137.90 1295.0 0.11340
## 96  0.021430 0.009280 0.013670 0.002299 32.490 47.16 214.00 3432.0 0.14010
## 97  0.063290 0.015610 0.019240 0.004614 28.190 28.18 195.90 2384.0 0.12720
## 98  0.008347 0.009472 0.017980 0.004261 10.750 20.88  68.09  355.2 0.14670
## 99  0.012670 0.019100 0.026780 0.003002 12.400 18.99  79.46  472.4 0.13590
## 100 0.080990 0.034870 0.034180 0.006517 11.860 22.33  78.27  437.6 0.10280
## 101 0.099600 0.027710 0.040770 0.022860 15.770 22.13 101.70  767.3 0.09983
## 102 0.056380 0.017330 0.018840 0.004787 25.930 26.24 171.10 2053.0 0.14950
## 103 0.076490 0.019360 0.027360 0.005928 23.680 29.43 158.80 1696.0 0.13470
## 104 0.000000 0.000000 0.061460 0.006820  8.952 22.44  56.65  240.1 0.13470
## 105 0.102700 0.025270 0.034910 0.007877 10.060 23.40  68.62  297.1 0.12210
## 106 0.031900 0.013690 0.027680 0.003345 25.300 31.86 171.10 1938.0 0.15920
## 107 0.058390 0.011860 0.040220 0.006187 17.730 25.21 113.70  975.2 0.14260
## 108 0.003297 0.004967 0.042430 0.001963 11.980 25.78  76.91  436.1 0.14240
## 109 0.045020 0.017440 0.018290 0.003733 20.990 33.15 143.20 1362.0 0.14490
## 110 0.039140 0.018160 0.021680 0.004445 24.540 34.37 161.10 1873.0 0.14980
## 111 0.035820 0.013010 0.014790 0.003118 30.670 30.73 202.40 2906.0 0.15150
## 112 0.043450 0.018060 0.037560 0.003288 22.750 34.66 157.60 1540.0 0.12180
## 113 0.026110 0.012960 0.036750 0.006758 10.880 19.48  70.89  357.1 0.13600
## 114 0.099530 0.022830 0.055430 0.007330 17.360 24.17 119.40  915.3 0.15500
## 115 0.042660 0.015080 0.023350 0.003385 33.130 23.58 229.30 3234.0 0.15300
## 116 0.059040 0.025360 0.037100 0.004286 24.190 33.81 160.00 1671.0 0.12780
## 117 0.020960 0.011970 0.012630 0.001803 30.750 26.44 199.50 3143.0 0.13630
## 118 0.035760 0.014710 0.015180 0.003796 27.660 25.80 195.00 2227.0 0.12940
## 119 0.060720 0.016560 0.031970 0.004085 19.380 31.03 129.30 1165.0 0.14150
## 120 0.034370 0.013430 0.016750 0.004367 22.690 21.84 152.10 1535.0 0.11920
## 121 0.026810 0.012320 0.012760 0.001711 25.370 23.17 166.80 1946.0 0.15620
## 122 0.153500 0.029190 0.016170 0.012200 10.850 22.82  76.51  351.9 0.11430
## 123 0.045880 0.013390 0.017380 0.004435 13.240 32.82  91.76  508.1 0.21840
## 124 0.094720 0.020470 0.012190 0.012330 12.040 18.93  79.73  450.0 0.11020
## 125 0.074690 0.034410 0.027680 0.006240 20.820 30.44 142.00 1313.0 0.12510
## 126 0.000000 0.000000 0.018650 0.006736 10.170 22.80  64.01  317.0 0.14600
## 127 0.047180 0.012880 0.020450 0.004028 26.230 28.74 172.00 2081.0 0.15020
## 128 0.061650 0.010510 0.015910 0.005099 20.800 27.78 149.60 1304.0 0.18730
## 129 0.013670 0.008674 0.030440 0.004590 10.850 31.24  68.73  359.4 0.15260
## 130 0.044930 0.021390 0.020180 0.005815 23.170 27.65 157.10 1748.0 0.15170
```

```
## 131 0.006021 0.010520 0.031000 0.004225 11.210 23.17  71.79  380.9 0.13980
## 132 0.079260 0.022340 0.014990 0.005784 16.350 27.57 125.40  832.7 0.14190
## 133 0.053210 0.018340 0.023830 0.004515 22.660 30.93 145.30 1603.0 0.13900
## 134 0.019930 0.011110 0.017170 0.004492 10.940 23.31  69.35  366.3 0.09794
## 135 0.031850 0.014660 0.010290 0.002205 25.680 32.07 168.20 2022.0 0.13680
## 136 0.066630 0.015530 0.023540 0.008925 12.790 28.18  83.51  507.2 0.09457
## 137 0.007936 0.009128 0.015640 0.002985 15.050 41.61  96.69  705.6 0.11720
## 138 0.027210 0.014580 0.020450 0.004417 22.960 34.49 152.10 1648.0 0.16000
## 139 0.080550 0.025980 0.016970 0.004558 36.040 31.37 251.20 4254.0 0.13570
## 140 0.076830 0.013680 0.015260 0.008133 15.440 25.50 115.00  733.5 0.12010
## 141 0.065910 0.023110 0.016730 0.011300 21.570 28.87 143.60 1437.0 0.12070
## 142 0.049610 0.018410 0.018070 0.005217 13.360 25.40  88.14  528.1 0.17800
## 143 0.014650 0.011830 0.020470 0.003883 13.600 33.33  87.24  567.6 0.10410
## 144 0.000000 0.000000 0.031410 0.003136 13.450 38.05  85.08  558.9 0.09422
## 145 0.111400 0.027210 0.032320 0.009627 13.780 21.03  97.82  580.6 0.11750
## 146 0.014230 0.005297 0.019610 0.001700 19.180 26.56 127.30 1084.0 0.10090
## 147 0.037370 0.016480 0.028970 0.003996 21.530 26.06 143.40 1426.0 0.13090
## 148 0.044350 0.015730 0.016170 0.005255 22.750 22.88 146.40 1600.0 0.14120
## 149 0.039760 0.021560 0.022010 0.002897 30.790 23.87 211.50 2782.0 0.11990
## 150 0.034830 0.021880 0.025420 0.010450 10.280 16.38  69.05  300.2 0.19020
## 151 0.051890 0.014500 0.026320 0.011480 10.600 18.04  69.47  328.1 0.20060
## 152 0.018000 0.012850 0.022200 0.008313 11.690 20.74  76.08  411.1 0.16620
## 153 0.021970 0.015800 0.039970 0.003901 10.570 17.84  67.84  326.6 0.18500
## 154 0.035710 0.015970 0.018790 0.004760 29.920 26.93 205.70 2642.0 0.13420
## 155 0.047630 0.028530 0.017150 0.005528 14.620 15.38  94.52  653.3 0.13940
## 156 0.040930 0.016990 0.028160 0.002719 23.230 27.15 152.00 1645.0 0.10970
## 157 0.040050 0.014210 0.019480 0.002689 24.300 25.48 160.20 1809.0 0.12680
## 158 0.018650 0.017660 0.015600 0.005824 12.980 32.19  86.12  487.7 0.17680
## 159 0.000000 0.000000 0.028820 0.006872  9.077 30.92  57.17  248.0 0.12560
## 160 0.092520 0.013640 0.021050 0.007551  8.678 31.89  54.49  223.6 0.15960
## 161 0.039960 0.012820 0.037590 0.004623  9.845 25.05  62.86  295.8 0.11030
## 162 0.004174 0.007082 0.025720 0.002278 10.650 22.88  67.88  347.3 0.12650
## 163 0.000000 0.000000 0.030040 0.003324 10.490 34.24  66.50  330.6 0.10730
## 164 0.057380 0.012670 0.014880 0.004738 12.480 37.16  82.28  474.2 0.12980
## 165 0.000000 0.000000 0.019890 0.001773 11.920 38.30  75.19  439.6 0.09267
## 166 0.073590 0.016080 0.021370 0.006142 17.520 42.79 128.70  915.0 0.14170
## 167 0.078450 0.026240 0.020570 0.006213 24.290 29.41 179.10 1819.0 0.14070
## 168 0.051980 0.024540 0.011140 0.004239 25.450 26.40 166.10 2027.0 0.14100
## 169 0.039500 0.016780 0.018980 0.002498 23.690 38.25 155.00 1731.0 0.11660
## 170 0.071170 0.016640 0.023240 0.006185 25.740 39.42 184.60 1821.0 0.16500
## 171 0.000000 0.000000 0.026760 0.002783  9.456 30.37  59.16  268.6 0.08996
##         V26     V27     V28    V29     V30 outliers_numb
## 1   0.66560 0.71190 0.26540 0.4601 0.11890             8
## 2   0.18660 0.24160 0.18600 0.2750 0.08902             1
## 3   0.42450 0.45040 0.24300 0.3613 0.08758             1
## 4   0.86630 0.68690 0.25750 0.6638 0.17300            11
## 5   0.20500 0.40000 0.16250 0.2364 0.07678             1
## 6   0.52490 0.53550 0.17410 0.3985 0.12440             1
## 7   0.54010 0.53900 0.20600 0.4378 0.10720             1
## 8   1.05800 1.10500 0.22100 0.4366 0.20750             8
## 9   0.39030 0.36390 0.17670 0.3176 0.10230            10
## 10  0.77250 0.69430 0.22080 0.3596 0.14310             4
## 11  0.65770 0.70260 0.17120 0.4218 0.13410             3
## 12  0.31500 0.53720 0.23880 0.2768 0.07615             2
```

```
## 13  0.59540 0.63050 0.23930 0.4667 0.09946            3
## 14  0.26000 0.31550 0.20090 0.2822 0.07526            5
## 15  0.35780 0.46950 0.20950 0.3613 0.09564            2
## 16  0.39490 0.38530 0.25500 0.4066 0.10590            4
## 17  0.66430 0.55390 0.27010 0.4264 0.12750            3
## 18  0.21170 0.34460 0.14900 0.2341 0.07421            2
## 19  0.42570 0.61330 0.18480 0.3444 0.09782            1
## 20  0.57750 0.69560 0.15460 0.4761 0.14020            2
## 21  0.65900 0.60910 0.17850 0.3672 0.11230            1
## 22  0.58040 0.52740 0.18640 0.4270 0.12330            2
## 23  0.38350 0.54090 0.18130 0.4863 0.08633            1
## 24  0.05131 0.02398 0.02899 0.1565 0.05504            3
## 25  0.26980 0.40230 0.14240 0.2964 0.09606            1
## 26  0.74440 0.72420 0.24930 0.4670 0.10380            9
## 27  0.22970 0.26230 0.13250 0.3021 0.07987            1
## 28  0.35110 0.38790 0.20910 0.3537 0.08294            2
## 29  0.09866 0.02168 0.02579 0.3557 0.08020            2
## 30  0.62470 0.69220 0.17850 0.2844 0.11320            1
## 31  0.16780 0.13970 0.05087 0.3282 0.08490            1
## 32  0.43650 1.25200 0.17500 0.4228 0.11750            9
## 33  0.23360 0.26870 0.17890 0.2551 0.06589            1
## 34  0.24360 0.14340 0.04786 0.2254 0.10840            4
## 35  0.73940 0.65660 0.18990 0.3313 0.13390            2
## 36  0.13790 0.08539 0.07407 0.2710 0.07191            1
## 37  0.56340 0.37860 0.21020 0.3751 0.11080            3
## 38  0.61640 0.76810 0.25080 0.5440 0.09964           13
## 39  0.60760 0.64760 0.28670 0.2355 0.10510           12
## 40  0.28170 0.24320 0.18410 0.2311 0.09203            2
## 41  0.35390 0.40980 0.15730 0.3689 0.08368            1
## 42  0.40990 0.63760 0.19860 0.3147 0.14050            2
## 43  0.69970 0.96080 0.29100 0.4055 0.09789           17
## 44  0.17650 0.13000 0.05334 0.2533 0.08468            1
## 45  0.22500 0.22160 0.11050 0.2226 0.08486            1
## 46  0.41930 0.67830 0.15050 0.2398 0.10820            4
## 47  0.18790 0.15440 0.03846 0.1652 0.07722            3
## 48  0.49250 0.73560 0.20340 0.3274 0.12520            1
## 49  0.12020 0.22490 0.11850 0.4882 0.06111            2
## 50  0.22910 0.32720 0.16740 0.2894 0.08456            1
## 51  0.42440 0.58030 0.22480 0.3222 0.08009           18
## 52  0.10280 0.10460 0.06968 0.1712 0.07343            1
## 53  0.21640 0.33550 0.16670 0.3414 0.07147            5
## 54  0.25170 0.09420 0.06042 0.2727 0.10360            1
## 55  0.40920 0.45040 0.18650 0.5774 0.10300            3
## 56  0.25210 0.25000 0.08405 0.2852 0.09218            1
## 57  0.11050 0.08112 0.06296 0.3196 0.06435            1
## 58  0.43100 0.53810 0.07879 0.3322 0.14860            3
## 59  0.27720 0.82160 0.15710 0.3108 0.12590           10
## 60  0.34980 0.35830 0.15150 0.2463 0.07738            1
## 61  0.20160 0.22640 0.17770 0.2443 0.06251            4
## 62  0.38460 0.68100 0.22470 0.3643 0.09223            2
## 63  0.35830 0.39480 0.23460 0.3589 0.09187            6
## 64  0.38300 0.48900 0.17210 0.2160 0.09300            3
## 65  0.08240 0.03938 0.04306 0.1902 0.07313            1
## 66  0.29500 0.34860 0.09910 0.2614 0.11620            6
```

```
## 67  0.40340 0.53400 0.26880 0.2856 0.08082            9
## 68  0.75840 0.67800 0.29030 0.4098 0.12840            4
## 69  0.10190 0.00692 0.01042 0.2933 0.07697            1
## 70  0.93270 0.84880 0.17720 0.5166 0.14460            9
## 71  0.02729 0.00000 0.00000 0.1909 0.06559            3
## 72  0.31220 0.38090 0.16730 0.3080 0.09333            1
## 73  0.40560 0.49670 0.18380 0.4753 0.10130            1
## 74  0.41670 0.78920 0.27330 0.3198 0.08762            8
## 75  0.52090 0.46460 0.20130 0.4432 0.10860            2
## 76  0.29200 0.38610 0.19200 0.2909 0.05865            4
## 77  0.15160 0.32010 0.15950 0.1648 0.05525           13
## 78  0.17930 0.28030 0.10990 0.1603 0.06818            5
## 79  0.40590 0.37440 0.17720 0.4724 0.10260            2
## 80  0.32350 0.36170 0.18200 0.3070 0.08255            4
## 81  0.40970 0.39950 0.16250 0.2713 0.07568            5
## 82  0.53430 0.62820 0.19770 0.3407 0.12430            1
## 83  0.06885 0.02318 0.03002 0.2911 0.07307            1
## 84  0.41260 0.58200 0.25930 0.3103 0.08677            9
## 85  0.37350 0.32410 0.20660 0.2853 0.08496            2
## 86  0.48480 0.74360 0.12180 0.3308 0.12970            3
## 87  0.10260 0.11810 0.06736 0.2883 0.07748            2
## 88  0.31720 0.69910 0.21050 0.3126 0.07849            6
## 89  0.59550 0.84890 0.25070 0.2749 0.12970            3
## 90  0.31610 0.43170 0.19990 0.3379 0.08950            1
## 91  0.53290 0.42510 0.19410 0.2818 0.10050            2
## 92  0.45030 0.44290 0.22290 0.3258 0.11910            1
## 93  0.51720 0.61810 0.24620 0.3277 0.10190            9
## 94  0.55640 0.57030 0.20140 0.3512 0.12040            2
## 95  0.28670 0.22980 0.15280 0.3067 0.07484            2
## 96  0.26440 0.34420 0.16590 0.2868 0.08218            9
## 97  0.47250 0.58070 0.18410 0.2833 0.08858            7
## 98  0.09370 0.04043 0.05159 0.2841 0.08175            1
## 99  0.08368 0.07153 0.08946 0.2220 0.06033            1
## 100 0.18430 0.15460 0.09314 0.2955 0.07009            4
## 101 0.24720 0.22200 0.10210 0.2272 0.08799            6
## 102 0.41160 0.61210 0.19800 0.2968 0.09929            4
## 103 0.33910 0.49320 0.19230 0.3294 0.09469            3
## 104 0.07767 0.00000 0.00000 0.3142 0.08116            3
## 105 0.37480 0.46090 0.11450 0.3135 0.10550            3
## 106 0.44920 0.53440 0.26850 0.5558 0.10240            3
## 107 0.21160 0.33440 0.10470 0.2736 0.07953            1
## 108 0.09669 0.01335 0.02022 0.3292 0.06522            2
## 109 0.20530 0.39200 0.18270 0.2623 0.07599            2
## 110 0.48270 0.46340 0.20480 0.3679 0.09870            1
## 111 0.26780 0.48190 0.20890 0.2593 0.07738            9
## 112 0.34580 0.47340 0.22550 0.4045 0.07918            1
## 113 0.16360 0.07162 0.04074 0.2434 0.08488            3
## 114 0.50460 0.68720 0.21350 0.4245 0.10500            6
## 115 0.59370 0.64510 0.27560 0.3690 0.08815           12
## 116 0.34160 0.37030 0.21520 0.3271 0.07632            5
## 117 0.16280 0.28610 0.18200 0.2510 0.06494            7
## 118 0.38850 0.47560 0.24320 0.2741 0.08574            8
## 119 0.46650 0.70870 0.22480 0.4824 0.09614            1
## 120 0.28400 0.40240 0.19660 0.2730 0.08666            1
```

```
## 121 0.30550 0.41590 0.21120 0.2689 0.07055            2
## 122 0.36190 0.60300 0.14650 0.2597 0.12000            5
## 123 0.93790 0.84020 0.25240 0.4154 0.14030            5
## 124 0.28090 0.30210 0.08272 0.2157 0.10430            3
## 125 0.24140 0.38290 0.18250 0.2576 0.07602            2
## 126 0.13100 0.00000 0.00000 0.2445 0.08865            1
## 127 0.57170 0.70530 0.24220 0.3828 0.10070            3
## 128 0.59170 0.90340 0.19640 0.3245 0.11980            3
## 129 0.11930 0.06141 0.03770 0.2872 0.08304            2
## 130 0.40020 0.42110 0.21340 0.3003 0.10480            3
## 131 0.13520 0.02085 0.04589 0.3196 0.08009            2
## 132 0.70900 0.90190 0.24750 0.2866 0.11550            3
## 133 0.34630 0.39120 0.17080 0.3007 0.08314            1
## 134 0.06542 0.03986 0.02222 0.2699 0.06736            1
## 135 0.31010 0.43990 0.22800 0.2268 0.07425            2
## 136 0.33990 0.32180 0.08750 0.2305 0.09952            1
## 137 0.14210 0.07003 0.07763 0.2196 0.07675            1
## 138 0.24440 0.26390 0.15550 0.3010 0.09060            3
## 139 0.42560 0.68330 0.26250 0.2641 0.07427           12
## 140 0.56460 0.65560 0.13570 0.2845 0.12490            3
## 141 0.47850 0.51650 0.19960 0.2301 0.12240            4
## 142 0.28780 0.31860 0.14160 0.2660 0.09270            1
## 143 0.09726 0.05524 0.05547 0.2404 0.06639            1
## 144 0.05213 0.00000 0.00000 0.2409 0.06743            1
## 145 0.40610 0.48960 0.13420 0.3231 0.10340            4
## 146 0.29200 0.24770 0.08737 0.4677 0.07623            1
## 147 0.23270 0.25440 0.14890 0.3251 0.07625            1
## 148 0.30890 0.35330 0.16630 0.2510 0.09445            1
## 149 0.36250 0.37940 0.22640 0.2908 0.07277            9
## 150 0.34410 0.20990 0.10250 0.3038 0.12520            5
## 151 0.36630 0.29130 0.10750 0.2848 0.13640            5
## 152 0.20310 0.12560 0.09514 0.2780 0.11680            3
## 153 0.20970 0.09996 0.07262 0.3681 0.08982            3
## 154 0.41880 0.46580 0.24750 0.3157 0.09671            9
## 155 0.13640 0.15590 0.10150 0.2160 0.07253            2
## 156 0.25340 0.30920 0.16130 0.3220 0.06386            1
## 157 0.31350 0.44330 0.21480 0.3077 0.07569            1
## 158 0.32510 0.13950 0.13080 0.2803 0.09970            1
## 159 0.08340 0.00000 0.00000 0.3058 0.09938            1
## 160 0.30640 0.33930 0.05000 0.2790 0.10660            3
## 161 0.08298 0.07993 0.02564 0.2435 0.07393            1
## 162 0.12000 0.01005 0.02232 0.2262 0.06742            1
## 163 0.07158 0.00000 0.00000 0.2475 0.06969            1
## 164 0.25170 0.36300 0.09653 0.2112 0.08732            1
## 165 0.05494 0.00000 0.00000 0.1566 0.05905            2
## 166 0.79170 1.17000 0.23560 0.4089 0.14090            5
## 167 0.41860 0.65990 0.25420 0.2929 0.09873            6
## 168 0.21130 0.41070 0.22160 0.2060 0.07115            5
## 169 0.19220 0.32150 0.16280 0.2572 0.06637            2
## 170 0.86810 0.93870 0.26500 0.4087 0.12400            7
## 171 0.06444 0.00000 0.00000 0.2871 0.07039            1
```

Jak widać na podsumowaniu:

- 30% obserwacji zawiera conajmniej 1 punkt oddalony
- niektóre obserwacje zawierają nawet ponad 10 punktów oddalonych

Sprawdzenie czy dla mniej rygorystycznego warunku (poniżej) znacznie się zmieni liczba punktów oddalonych

$$(x < Q_1 - 2 * IQR) \ \vee \ (x > Q_3 + 2 * IQR)$$

```
## [1] "Total number of outliers: 361"
## [1] "% of outliers: 1.92"
## [1] "Total number of rows with outliers: 115"
## [1] "% of rows with outliers: 20.21"
```

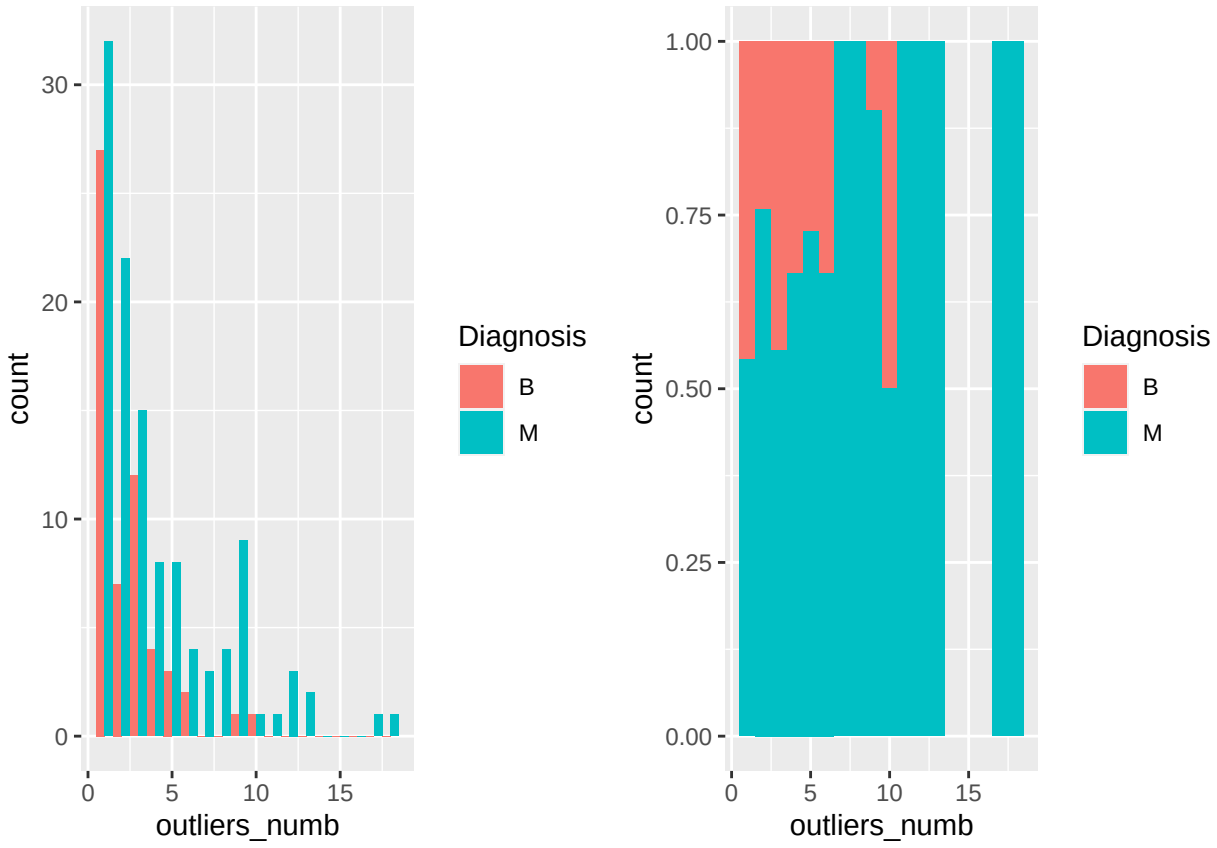
- Dla znacznego zwiększenia zakresu występowania punktów oddalonych, nadal 20% obserwacji się klasyfikuje jako outlier

Sprawdzenie czy duża liczba punktów oddalonych jest skorelowana ze zmienną celu

```
wdbc <- outliers(wdbc, 3, ncol(wdbc))
```

```
## [1] "Total number of outliers: 608"
## [1] "% of outliers: 3.24"
## [1] "Total number of rows with outliers: 171"
## [1] "% of rows with outliers: 30.05"
```

```
temp <- wdbc %>% filter(wdbc$outliers_numb > 0)
dod <- ggplot(temp, aes(outliers_numb, fill = Diagnosis)) +
  geom_histogram(binwidth = 1,position = 'dodge')
fil <- ggplot(temp, aes(outliers_numb, fill = Diagnosis)) +
  geom_histogram(binwidth = 1,position = 'fill')
grid.arrange(dod, fil, ncol=2)
```



- Wynika z tego, że większość outlierów związana jest z mniej liczną grupą zmiennej `Diagnosis` czyli `M`, co może być nieprzypadkowe
- Można założyć, że dla więcej niż 7 outlierów w obserwacji zmienna przymuje wartość `M`

Utworzenie dodatkowej zmiennej `many_outl` z oznaczeniem dla obserwacji, w których występuje >7 outlierów

```
wdbc$many_outl[wdbc$outliers_numb > 7] <- TRUE
wdbc$many_outl[wdbc$outliers_numb <= 7] <- FALSE
```

Ze względu na dużą ilość outlierów, zastąpię ich wartości medianą danej zmiennej `V` zamiast je usuwać

Data Cleaning

Punkty oddalone

Utworzenie funkcji do nadpisywania outlierów w data frame'ach

```
outliers_deal <- function(x, s, e, f){

  # x = dataframe
  # s = index of first col to take
  # e = index of last column to take
  # f =  method to replace outliers (mean, median, mode)
```

```
  for(i in s:e){

     val <- f(x[,i])
    Q1 <- quantile(x[,i], 0.25, names = FALSE)
    Q3 <- quantile(x[,i], 0.75, names = FALSE)
    iqr <- IQR(x[,i])
    low <- Q1 - iqr*1.5
    up <- Q3 + iqr*1.5

    x[,i] <- ifelse((x[,i] < low) | (x[,i] > up), val, x[,i])
  }
  return(invisible(x))
}
```

Sprawdzenie funkcji

```
w <- data.frame(col1 = c(1, 2, 3, 4, 5, 90, 6),
                col2 = c(13000, 6, 13000, 18000, 13000, 12000, 90000),
                col3 = c(1, 899, 5, 4, 3, 8, 6))
w
```

```
##   col1  col2 col3
## 1    1 13000    1
## 2    2     6  899
## 3    3 13000    5
## 4    4 18000    4
## 5    5 13000    3
## 6   90 12000    8
## 7    6 90000    6
```

```
w <- outliers_deal(w, 1, 3, median)
w
```

```
##   col1  col2 col3
## 1    1 13000    1
## 2    2 13000    5
## 3    3 13000    5
## 4    4 18000    4
## 5    5 13000    3
## 6    4 12000    8
## 7    6 13000    6
```

Utworzenie nowego dataframe z zastąpionymi punktami oddalonymi i sprawdzenie występowania "nowych" outlierów

```
wdbc1 <- outliers_deal(wdbc, 3, ncol(wdbc)-2, median)
outliers(wdbc1, 3, ncol(wdbc)-2)
```

```
## [1] "Total number of outliers: 373"
## [1] "% of outliers: 1.93"
## [1] "Total number of rows with outliers: 179"
## [1] "% of rows with outliers: 31.46"
```

- Zastąpienie punktów oddalonych medianą spowodowało zaklasyfikowanie "nowych" punktów jako outliery

Porównanie oryginalnych danych z zastąpionymi przez medianę dla wybranych zmiennych `V`

```
identical(wdbc1, wdbc)
```

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

```
summary(wdbc[,7:10])
```

```
##        V5                V6                V7                V8
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
```

```
summary(wdbc1[,7:10])
```

```
##        V5                V6                V7                V8
##  Min.   :0.06251   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08641   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09600   Mean   :0.09940   Mean   :0.07995   Mean   :0.04643
##  3rd Qu.:0.10490   3rd Qu.:0.12750   3rd Qu.:0.11680   3rd Qu.:0.06847
##  Max.   :0.13350   Max.   :0.22840   Max.   :0.28100   Max.   :0.15200
```

- Redukcja outlierów wyraźnie zmniejszyła wartości maksymalne zmiennych
- W ten sposób przygotowane dane zostaną zastosowane do modelowania w drugiej części

Współliniowość

Stopniowa redukcja współliniowych zmiennych `V` poprzez wyznaczanie kolejnych współczynników VIF i usuwanie zmiennych dla których jest spełniony warunek:

$$VIF > 5$$

```
VIF_max <- 5
wdbc1_V <- wdbc1 %>% select(3:32)
#nie wiem jak zrobić działającą pętle (?)
```

```
s <- lm(formula = V1 ~ . , data = wdbc1_V) %>% summary()
vif_z <- 1 / (1 - s$r.squared)
ifelse(vif_z > VIF_max, print(c(vif_z, "VIF > 5, zmienna wspołliniowa !")), vif_z)
```

```
## [1] "77.9526138618261"                "VIF > 5, zmienna wspołliniowa !"
```

```
## [1] "77.9526138618261"
```

```
## [1] "7.07048591532057"               "VIF > 5, zmienna współliniowa !"

## [1] "7.07048591532057"

## [1] "12.9000092396509"               "VIF > 5, zmienna współliniowa !"

## [1] "12.9000092396509"

## [1] "9.85111003022877"               "VIF > 5, zmienna współliniowa !"

## [1] "9.85111003022877"

## [1] 4.191865

## [1] "6.20050876327581"               "VIF > 5, zmienna współliniowa !"

## [1] "6.20050876327581"

## [1] "6.68195482228325"               "VIF > 5, zmienna współliniowa !"

## [1] "6.68195482228325"

## [1] "6.22790922628577"               "VIF > 5, zmienna współliniowa !"

## [1] "6.22790922628577"

## [1] 2.215314

## [1] 3.391566

## [1] "6.85855458757022"               "VIF > 5, zmienna współliniowa !"

## [1] "6.85855458757022"

## [1] 2.392891

## [1] 2.926042

## [1] 2.507827

## [1] 1.99997

## [1] 4.241589

## [1] 3.968189

## [1] 3.525691
```

```
## [1] 2.067413

## [1] 2.973834

## [1] "29.6337719904128"                "VIF > 5, zmienna współliniowa !"

## [1] "29.6337719904128"

## [1] 2.234111

## [1] "6.74261423791062"                "VIF > 5, zmienna współliniowa !"

## [1] "6.74261423791062"

## [1] 2.466394

## [1] 4.370501

## [1] "5.91978658000078"                "VIF > 5, zmienna współliniowa !"

## [1] "5.91978658000078"

## [1] "5.06611335850456"                "VIF > 5, zmienna współliniowa !"

## [1] "5.06611335850456"

## [1] "5.9395648113425"                 "VIF > 5, zmienna współliniowa !"

## [1] "5.9395648113425"

## [1] 2.704929

## [1] 2.820339
```

Ostateczna postać data frame'a `wdbc1` przygotowana do modelowania:

```
colnames(wdbc1_V)
```

```
##  [1] "V5"  "V9"  "V10" "V12" "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20"
## [13] "V22" "V24" "V25" "V29" "V30"
```

```
wdbc1 <- wdbc1 %>% select(2, 7,11,12,14:22,24,26,27,31:34)
head(wdbc1)
```

```
##   Diagnosis      V5     V9     V10    V12   V13   V14      V15     V16     V17
## 1         M 0.11840 0.2419 0.07871 0.9053 2.287 24.53 0.006399 0.04904 0.05373
## 2         M 0.08474 0.1812 0.05667 0.7339 3.398 74.08 0.005225 0.01308 0.01860
## 3         M 0.10960 0.2069 0.05999 0.7869 4.585 24.53 0.006150 0.04006 0.03832
## 4         M 0.09587 0.1792 0.06154 1.1560 3.445 27.23 0.009110 0.02045 0.05661
## 5         M 0.10030 0.1809 0.05883 0.7813 5.438 24.53 0.011490 0.02461 0.05688
## 6         M 0.12780 0.2087 0.07613 0.8902 2.217 27.19 0.007510 0.03345 0.03672
##       V18     V19      V20   V22    V24    V25    V29     V30 outliers_numb
## 1 0.01587 0.03003 0.006193 17.33  686.5 0.1622 0.2822 0.11890             8
## 2 0.01340 0.01389 0.003532 23.41  686.5 0.1238 0.2750 0.08902             1
## 3 0.02058 0.02250 0.004571 25.53 1709.0 0.1444 0.3613 0.08758             1
## 4 0.01867 0.01873 0.003187 26.50  567.7 0.1313 0.2822 0.08004            11
## 5 0.01885 0.01756 0.005115 16.67 1575.0 0.1374 0.2364 0.07678             1
## 6 0.01137 0.02165 0.005082 23.75  741.6 0.1791 0.3985 0.08004             1
##   many_outl
## 1      TRUE
## 2     FALSE
## 3     FALSE
## 4      TRUE
## 5     FALSE
## 6     FALSE
```

Modelowanie

W ramach pracy zbudowane zostaną 3 modele:

- model 1: klasyczną regresją logistyczną bez regularyzacji
- model 2: regresją krokową
- model 3: regresją logistyczną z regularyzacją Lasso

Modele zostaną porównane na podstawie metryk:

- **Accuracy**, ponieważ jest bardzo intuicyjna, mówi o stosunku poprawnie sklasyfikowanych obserwacji do ilości wszystkich klasyfikacji oraz w swojej pracy założyłam, że klasy nie są niezbalansowane
- **F1**, która jest średnią harmoniczną z wartości **recall** oraz **precision** i uwzględnia je równomiernie.

Z punktu widzenia analizowanego problemu, czyli klasyfikacji rodzaju nowotworu pacjenta, metryka **recall** jest ważna ze względu na to, że wskazuje ilu pacjentów z odmianą złośliwą zostało błędnie sklasyfikowanych, natomiast jednocześnie **precision** wskazuje dokładność klasyfikacji i uwzględnia przypadki osób z łagodnym nowotworem, które błędnie zostały uznane za chorych na odmianę złośliwą

Proces modelowania obejmuje następujące kroki:

- Podział zbioru na **zbiór testowy** oraz **zbiór treningowy** poprzez utworzenie 10 fold'ów
- Utworzenie podstawowego modelu
- Redukcja liczby predyktorów do jedynie istotnych na podstawie **p-value**
- Sprawdzenie separacji klas dla modelu poprzez ich wizualizację na histogramie
- Określenie jakości modelu poprzez wznaczenie średnich współczynników **Accuracy** oraz **F1** w walidacji krzyżowej
- Te same kroki dla modelu 2 i 3
- Porównanie metryk między modelami i wybór najlepszego

```
wdbc1 <- wdbc1 %>% mutate(cv_fold = (row_number() - 1) %% 10)
model1_ACC <- c()
model1_F1 <- c()
model2_ACC <- c()
model2_F1 <- c()
model3_ACC <- c()
model3_F1 <- c()

models_summary <- data.frame(matrix(ncol = 3, nrow = 0))
colnames(models_summary) <- c("model", "ACC", "F1")
```

Model 1: regresja logistyczna

Stworzenie podstawowego modelu i sprawdzenie istotności zmiennych

```
train <- wdbc1 %>% filter(cv_fold != 0) %>% select(-cv_fold)
test <- wdbc1 %>% filter(cv_fold == 0) %>% select(-cv_fold)
model1 <- glm(data = train, formula = Diagnosis ~ ., family = binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
print(summary(model1))
```

```
##
## Call:
## glm(formula = Diagnosis ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8685  -0.0195  -0.0008   0.0006   4.0838
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.621e+01  1.265e+01  -4.444 8.84e-06 ***
## V5             1.064e+02  5.712e+01   1.863  0.06242 .
## V9             2.053e+01  2.804e+01   0.732  0.46407
## V10            6.329e+00  1.023e+02   0.062  0.95068
## V12           -2.452e+00  1.251e+00  -1.960  0.04994 *
## V13            2.236e-01  7.374e-01   0.303  0.76174
## V14            9.909e-02  5.361e-02   1.848  0.06456 .
## V15            5.588e+02  3.421e+02   1.633  0.10239
## V16           -4.028e+01  6.948e+01  -0.580  0.56216
## V17            3.302e+01  3.433e+01   0.962  0.33609
## V18            3.517e+02  1.445e+02   2.434  0.01493 *
## V19           -3.285e+02  1.349e+02  -2.435  0.01487 *
## V20           -9.115e+02  6.165e+02  -1.478  0.13930
## V22            5.147e-01  1.284e-01   4.009 6.09e-05 ***
## V24            1.929e-02  4.353e-03   4.433 9.31e-06 ***
## V25            2.643e+01  3.610e+01   0.732  0.46413
## V29            1.810e+01  1.705e+01   1.061  0.28846
## V30            4.506e+01  3.975e+01   1.134  0.25696
## outliers_numb  1.189e+00  4.005e-01   2.969  0.00299 **
## many_outlTRUE -3.317e+00  3.901e+00  -0.850  0.39510
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 678.454  on 511  degrees of freedom
## Residual deviance:  66.142  on 492  degrees of freedom
## AIC: 106.14
##
## Number of Fisher Scoring iterations: 10
```

- Zmienne V5, V9, V10, V13-17, V20, V25, V29, V30 i many_outl są nieistotne, więc będą pojedynczo usuwane z modelu od najmniejszej istotności

```
train <- wdbc1 %>% filter(cv_fold != 0) %>% select(-cv_fold)
test <- wdbc1 %>% filter(cv_fold == 0) %>% select(-cv_fold)
model1 <- glm(data = train, formula = Diagnosis ~ V5 + V9 + V12 + V13 + V14 + V15 + V16 + V17 + V18 + V
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
print(summary(model1))
```

```
##
## Call:
## glm(formula = Diagnosis ~ V5 + V9 + V12 + V13 + V14 + V15 + V16 +
##     V17 + V18 + V19 + V20 + V22 + V24 + V25 + V29 + V30 + outliers_numb +
##     many_outl, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8593  -0.0196  -0.0008   0.0006   4.0847
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.589e+01  1.146e+01  -4.876 1.08e-06 ***
## V5             1.074e+02  5.501e+01   1.952  0.05098 .
## V9             2.026e+01  2.768e+01   0.732  0.46410
## V12           -2.448e+00  1.248e+00  -1.961  0.04992 *
## V13            2.207e-01  7.380e-01   0.299  0.76494
## V14            9.884e-02  5.366e-02   1.842  0.06547 .
## V15            5.536e+02  3.317e+02   1.669  0.09514 .
## V16           -4.066e+01  6.941e+01  -0.586  0.55803
## V17            3.328e+01  3.406e+01   0.977  0.32854
## V18            3.515e+02  1.445e+02   2.432  0.01502 *
## V19           -3.267e+02  1.317e+02  -2.481  0.01311 *
## V20           -8.970e+02  5.685e+02  -1.578  0.11462
## V22            5.137e-01  1.269e-01   4.046 5.20e-05 ***
## V24            1.925e-02  4.276e-03   4.501 6.77e-06 ***
## V25            2.650e+01  3.599e+01   0.736  0.46157
## V29            1.804e+01  1.698e+01   1.062  0.28809
## V30            4.586e+01  3.751e+01   1.223  0.22144
## outliers_numb  1.186e+00  3.970e-01   2.988  0.00281 **
## many_outlTRUE -3.341e+00  3.877e+00  -0.862  0.38884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 678.454  on 511  degrees of freedom
## Residual deviance:  66.146  on 493  degrees of freedom
## AIC: 104.15
##
## Number of Fisher Scoring iterations: 10
```

itd...

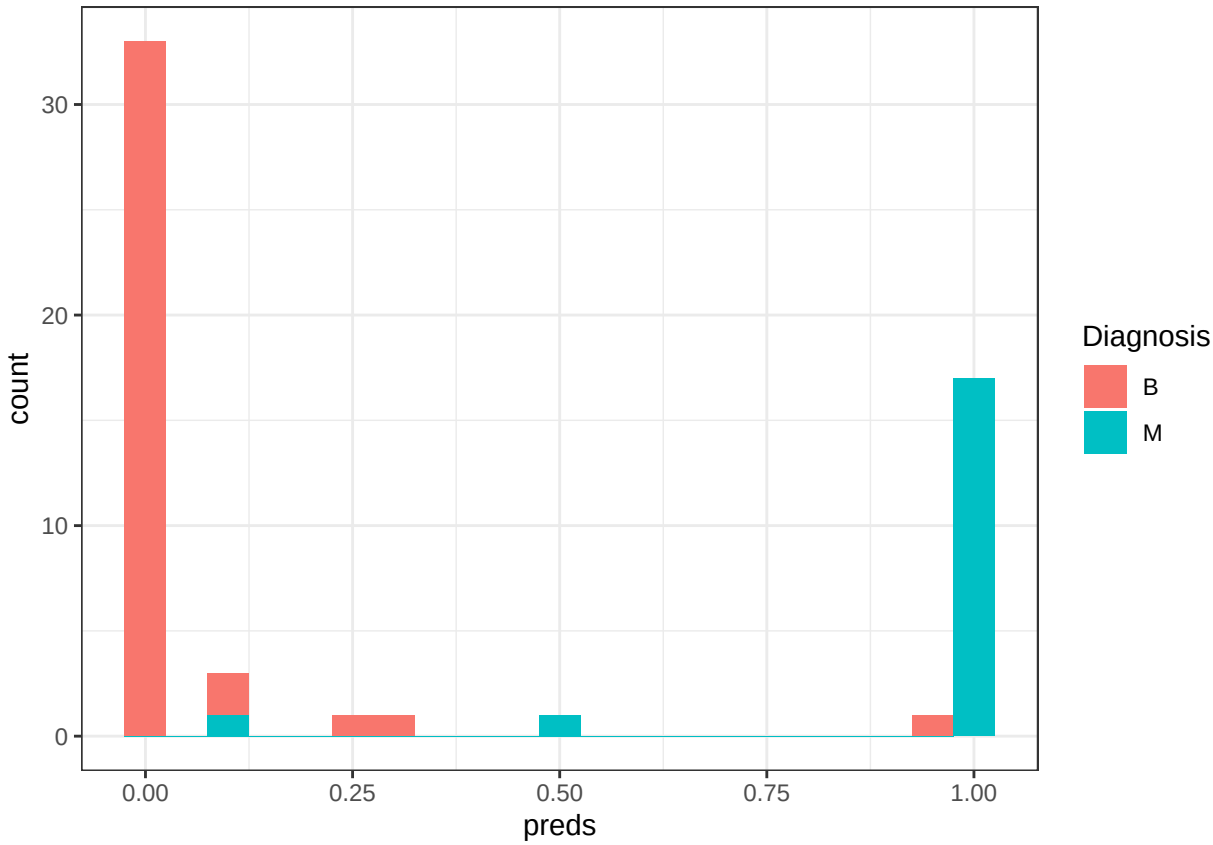
```
train <- wdbc1 %>% filter(cv_fold != 0) %>% select(-cv_fold)
test <- wdbc1 %>% filter(cv_fold == 0) %>% select(-cv_fold)
model1 <- glm(data = train, formula = Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 + V22 + V24 + V29 + c
print(summary(model1))
```

```
##
## Call:
## glm(formula = Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 +
##     V22 + V24 + V29 + outliers_numb, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6032  -0.0298  -0.0019   0.0014   3.9849
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.505e+01  7.584e+00  -5.940 2.85e-09 ***
## V5             9.049e+01  3.267e+01   2.769 0.005617 **
## V12           -2.206e+00  1.009e+00  -2.186 0.028801 *
## V14            9.018e-02  3.832e-02   2.353 0.018606 *
## V15            5.174e+02  2.516e+02   2.056 0.039759 *
## V18            3.600e+02  1.091e+02   3.299 0.000972 ***
## V19           -3.848e+02  1.059e+02  -3.635 0.000278 ***
## V22            4.402e-01  9.400e-02   4.684 2.82e-06 ***
## V24            1.725e-02  3.387e-03   5.092 3.55e-07 ***
## V29            3.271e+01  1.022e+01   3.201 0.001371 **
## outliers_numb  9.848e-01  2.223e-01   4.430 9.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 678.454  on 511  degrees of freedom
## Residual deviance:  73.411  on 501  degrees of freedom
## AIC: 95.411
##
## Number of Fisher Scoring iterations: 9
```

W modelu pozostały jedynie istotne zmienne, jednak z jakiegoś powodu pojawia się ostrtzeżenie "Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred".

Wizualizacja separacji grup dla modelu w celu jej oceny i dobrania cutoffu

```
model1_pred <- predict(model1, test, type = "response") %>% bind_cols(test %>% select(Diagnosis), preds
ggplot(model1_pred, aes(x = preds, fill = Diagnosis)) + geom_histogram(binwidth = 0.05) + theme_bw()
```



Separacja grup zmiennej celu jest podejrzanie bardzo wyraźna. Cutoff pozstanie = 0.5.

Cross-validation

Poprzez walidację krzyżową **modelu 1** wyznaczono Accuracy oraz F1

```
for (fold in 0:9) {
  train <- wdbc1 %>% filter(cv_fold != fold) %>% select(-cv_fold)
  test <- wdbc1 %>% filter(cv_fold == fold) %>% select(-cv_fold)
  model1 <- glm(data = train, formula = Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 + V22 + V24 + V29 +
  model1_pred <- predict(model1, test, type = "response") %>% bind_cols(test %>% select(Diagnosis), pred
  cut05 <- model1_pred %>% mutate(predicted = ifelse(preds >= 0.5, 'M', 'B')) %>%
    select(-preds) %>% select(predicted, Diagnosis) %>%
    mutate_all(list(~ factor(., levels = c('M', 'B')))) %>% table()
  model1_ACC[fold] <- sum(diag(cut05)) / sum(cut05)
  pre <- cut05[1, 1] / sum(cut05[1, ])
  rec <- cut05[1, 1] / sum(cut05[, 1])
  model1_F1[fold] <- 2 * pre * rec / (pre + rec)
}

model_1_summary <- list("model 1", mean(model1_ACC), mean(model1_F1))
models_summary[1,] <- model_1_summary
```

```
models_summary
```

```
##     model       ACC       F1
```

```
## 1 model 1 0.9648427 0.951722
```

Model 2: regresja krokowa

Stworzenie modelu regresji krokowej, która dobierze model poprzez minimalizację współczynnika Akaike'go (AIC)

```
train <- wdbc1 %>% filter(cv_fold != 0) %>% select(-cv_fold)
test <- wdbc1 %>% filter(cv_fold == 0) %>% select(-cv_fold)

null_model <- glm(data = train, formula = Diagnosis ~ 1, family = binomial(link = "logit"))
full_model <- glm(data = train, formula = Diagnosis ~ ., family = binomial(link = "logit"))
model2 <- step(full_model, scope = list(lower = null_model, upper = full_model),
direction = "backward")
```

```
## Start:  AIC=106.14
## Diagnosis ~ V5 + V9 + V10 + V12 + V13 + V14 + V15 + V16 + V17 +
##     V18 + V19 + V20 + V22 + V24 + V25 + V29 + V30 + outliers_numb +
##     many_outl
##
##                 Df Deviance    AIC
## - V10            1   66.146 104.15
## - V13            1   66.237 104.24
## - V16            1   66.495 104.50
## - V25            1   66.675 104.67
## - V9             1   66.695 104.69
## - many_outl      1   66.857 104.86
## - V17            1   67.142 105.14
## - V29            1   67.316 105.32
## - V30            1   67.382 105.38
## <none>               66.142 106.14
## - V20            1   68.447 106.45
## - V15            1   69.106 107.11
## - V5             1   70.088 108.09
## - V12            1   70.359 108.36
## - V14            1   70.375 108.38
## - V19            1   72.718 110.72
## - V18            1   75.145 113.14
## - outliers_numb  1   84.951 122.95
## - V22            1   98.260 136.26
## - V24            1  141.144 179.14
##
## Step:  AIC=104.15
## Diagnosis ~ V5 + V9 + V12 + V13 + V14 + V15 + V16 + V17 + V18 +
##     V19 + V20 + V22 + V24 + V25 + V29 + V30 + outliers_numb +
##     many_outl
##
##                 Df Deviance    AIC
## - V13            1   66.239 102.24
## - V16            1   66.504 102.50
## - V25            1   66.686 102.69
## - V9             1   66.697 102.70
```

```
## - many_outl      1   66.883 102.88
## - V17            1   67.181 103.18
## - V29            1   67.317 103.32
## - V30            1   67.613 103.61
## <none>               66.146 104.15
## - V20            1   68.642 104.64
## - V15            1   69.280 105.28
## - V12            1   70.361 106.36
## - V14            1   70.469 106.47
## - V5             1   70.870 106.87
## - V19            1   73.263 109.26
## - V18            1   75.204 111.20
## - outliers_numb  1   86.139 122.14
## - V22            1   98.409 134.41
## - V24            1  151.529 187.53
##
## Step:  AIC=102.24
## Diagnosis ~ V5 + V9 + V12 + V14 + V15 + V16 + V17 + V18 + V19 +
##     V20 + V22 + V24 + V25 + V29 + V30 + outliers_numb + many_outl
##
##                 Df Deviance    AIC
## - V16            1   66.513 100.51
## - V9             1   66.751 100.75
## - V25            1   66.781 100.78
## - V17            1   67.306 101.31
## - many_outl      1   67.395 101.39
## - V29            1   67.504 101.50
## - V30            1   67.627 101.63
## <none>               66.239 102.24
## - V20            1   68.878 102.88
## - V15            1   69.354 103.35
## - V12            1   70.367 104.37
## - V5             1   70.979 104.98
## - V19            1   73.281 107.28
## - V18            1   75.227 109.23
## - V14            1   76.980 110.98
## - outliers_numb  1   90.180 124.18
## - V22            1   98.589 132.59
## - V24            1  162.700 196.70
##
## Step:  AIC=100.51
## Diagnosis ~ V5 + V9 + V12 + V14 + V15 + V17 + V18 + V19 + V20 +
##     V22 + V24 + V25 + V29 + V30 + outliers_numb + many_outl
##
##                 Df Deviance     AIC
## - V9             1   67.210  99.210
## - V25            1   67.242  99.242
## - V17            1   67.336  99.336
## - V29            1   67.589  99.589
## - many_outl      1   67.831  99.831
## - V30            1   67.839  99.839
## <none>               66.513 100.513
## - V15            1   69.818 101.818
## - V5             1   71.140 103.140
```

```
## - V12            1   71.281 103.281
## - V20            1   71.991 103.991
## - V19            1   74.433 106.433
## - V18            1   75.244 107.244
## - V14            1   77.300 109.300
## - outliers_numb  1   91.332 123.332
## - V22            1   99.756 131.756
## - V24            1  164.716 196.716
##
## Step:  AIC=99.21
## Diagnosis ~ V5 + V12 + V14 + V15 + V17 + V18 + V19 + V20 + V22 +
##     V24 + V25 + V29 + V30 + outliers_numb + many_outl
##
##                 Df Deviance     AIC
## - V25            1   67.682  97.682
## - V17            1   67.843  97.843
## - many_outl      1   68.031  98.031
## - V30            1   68.206  98.206
## <none>               67.210  99.210
## - V15            1   70.364 100.364
## - V29            1   71.210 101.210
## - V12            1   71.324 101.324
## - V20            1   72.001 102.001
## - V5             1   73.013 103.013
## - V19            1   75.172 105.172
## - V18            1   76.002 106.002
## - V14            1   77.816 107.816
## - outliers_numb  1   91.444 121.444
## - V22            1   99.989 129.989
## - V24            1  165.675 195.675
##
## Step:  AIC=97.68
## Diagnosis ~ V5 + V12 + V14 + V15 + V17 + V18 + V19 + V20 + V22 +
##     V24 + V29 + V30 + outliers_numb + many_outl
##
##                 Df Deviance     AIC
## - V17            1   68.250  96.250
## - V30            1   68.860  96.860
## - many_outl      1   68.984  96.984
## <none>               67.682  97.682
## - V20            1   72.379 100.379
## - V12            1   72.578 100.578
## - V15            1   73.951 101.951
## - V18            1   76.286 104.286
## - V29            1   76.835 104.835
## - V14            1   78.797 106.797
## - V5             1   82.211 110.211
## - V19            1   83.888 111.888
## - outliers_numb  1   93.660 121.660
## - V22            1  102.470 130.470
## - V24            1  166.974 194.974
##
## Step:  AIC=96.25
## Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 + V20 + V22 + V24 +
```

```
##     V29 + V30 + outliers_numb + many_outl
##
##                 Df Deviance     AIC
## - V30            1   69.105  95.105
## - many_outl      1   69.519  95.519
## <none>               68.250  96.250
## - V20            1   72.405  98.405
## - V12            1   73.685  99.685
## - V15            1   75.163 101.163
## - V29            1   78.807 104.807
## - V14            1   79.001 105.001
## - V18            1   79.675 105.675
## - V5             1   82.400 108.400
## - V19            1   84.651 110.651
## - outliers_numb  1   96.816 122.816
## - V22            1  104.923 130.923
## - V24            1  174.736 200.736
##
## Step:  AIC=95.11
## Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 + V20 + V22 + V24 +
##     V29 + outliers_numb + many_outl
##
##                 Df Deviance     AIC
## - many_outl      1   70.178  94.178
## <none>               69.105  95.105
## - V20            1   72.412  96.412
## - V12            1   75.117  99.117
## - V15            1   75.190  99.190
## - V14            1   79.103 103.103
## - V18            1   81.364 105.364
## - V5             1   83.074 107.074
## - V29            1   83.726 107.726
## - V19            1   86.166 110.166
## - outliers_numb  1   96.913 120.913
## - V22            1  107.254 131.254
## - V24            1  179.991 203.991
##
## Step:  AIC=94.18
## Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 + V20 + V22 + V24 +
##     V29 + outliers_numb
##
##                 Df Deviance     AIC
## <none>               70.178  94.178
## - V20            1   73.411  95.411
## - V15            1   75.272  97.272
## - V12            1   75.852  97.852
## - V14            1   80.484 102.484
## - V5             1   83.530 105.530
## - V29            1   84.921 106.921
## - V18            1   87.079 109.079
## - V19            1   89.566 111.566
## - V22            1  108.579 130.579
## - outliers_numb  1  121.334 143.334
## - V24            1  182.317 204.317
```

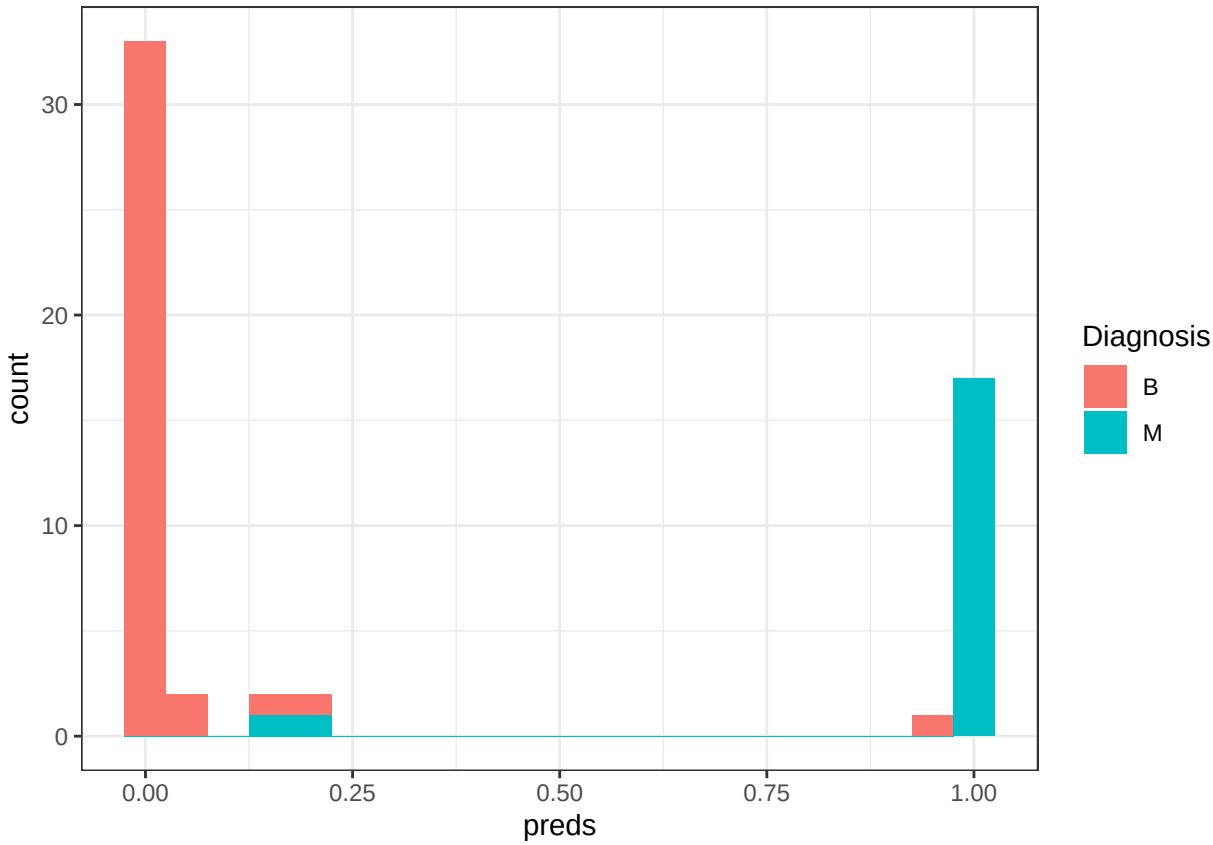
```
summary(model2)
```

```
##
## Call:
## glm(formula = Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 +
##     V20 + V22 + V24 + V29 + outliers_numb, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7589  -0.0259  -0.0010   0.0009   4.0546
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.943e+01  8.622e+00  -5.733 9.88e-09 ***
## V5             1.162e+02  3.753e+01   3.097 0.001953 **
## V12           -2.484e+00  1.073e+00  -2.315 0.020619 *
## V14            9.733e-02  3.813e-02   2.553 0.010694 *
## V15            5.418e+02  2.546e+02   2.128 0.033344 *
## V18            4.389e+02  1.250e+02   3.511 0.000447 ***
## V19           -3.964e+02  1.078e+02  -3.677 0.000236 ***
## V20           -5.832e+02  3.274e+02  -1.781 0.074889 .
## V22            4.982e-01  1.108e-01   4.498 6.87e-06 ***
## V24            1.771e-02  3.388e-03   5.226 1.74e-07 ***
## V29            3.615e+01  1.074e+01   3.365 0.000765 ***
## outliers_numb  1.032e+00  2.262e-01   4.564 5.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 678.454  on 511  degrees of freedom
## Residual deviance:  70.178  on 500  degrees of freedom
## AIC: 94.178
##
## Number of Fisher Scoring iterations: 10
```

W wyniku regresji krokowej, zostały dobrane następujące predyktory: `V5`, `V12`, `V14`, `V15`, `V18`, `V19`, `V20`, `V22`, `V24`, `V29`, `outliers_numb`, z czego zmienna `V20` jest nieistotna na podstawie p-value

Wizualizacja separacji grup dla modelu w celu jej oceny i dobrania cutoffu

```
model2_pred <- predict(model2, test, type = "response") %>% bind_cols(test %>% select(Diagnosis), preds
ggplot(model2_pred, aes(x = preds, fill = Diagnosis)) + geom_histogram(binwidth = 0.05) + theme_bw()
```



Separacja grup zmiennej celu dla drugiego modelu jest również bardzo wyraźna. Założę cutoff = 0.5.

Cross-validation

Poprzez walidację krzyżową **modelu 2** wyznaczono Accuracy oraz F1

```
for (fold in 0:9) {
  train <- wdbc1 %>% filter(cv_fold != fold) %>% select(-cv_fold)
  test <- wdbc1 %>% filter(cv_fold == fold) %>% select(-cv_fold)
  model2 <- glm(formula = Diagnosis ~ V5 + V12 + V14 + V15 + V18 + V19 +
    V20 + V22 + V24 + V29 + outliers_numb, family = binomial(link = "logit"), data = train)
  model2_pred <- predict(model2, test, type = "response") %>% bind_cols(test %>% select(Diagnosis), pre
  cut05 <- model2_pred %>% mutate(predicted = ifelse(preds >= 0.5, 'M', 'B')) %>%
    select(-preds) %>% select(predicted, Diagnosis) %>%
    mutate_all(list(~ factor(., levels = c('M', 'B')))) %>% table()
  model2_ACC[fold] <- sum(diag(cut05)) / sum(cut05)
  pre <- cut05[1, 1] / sum(cut05[1, ])
  rec <- cut05[1, 1] / sum(cut05[, 1])
  model2_F1[fold] <- 2 * pre * rec / (pre + rec)
}

model_2_summary <- list("model 2", mean(model2_ACC), mean(model2_F1))
models_summary[2,] <- model_2_summary
```

```
models_summary
```

```
##     model       ACC        F1
## 1 model 1 0.9648427 0.9517220
## 2 model 2 0.9667920 0.9533436
```

Model otrzymany metodą regresji krokowej ma nieco wyższe wartości dla obu metryk oraz przy stosowaniu tego modelu również pojawia się komunikat ostrzegawczy "Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred". Być może świadczy on o wciąż obecnych outlierach w danych. (???)

Model 3: regresja z regularyzacją LASSO z wykorzystaniem pakietu `glmnet`

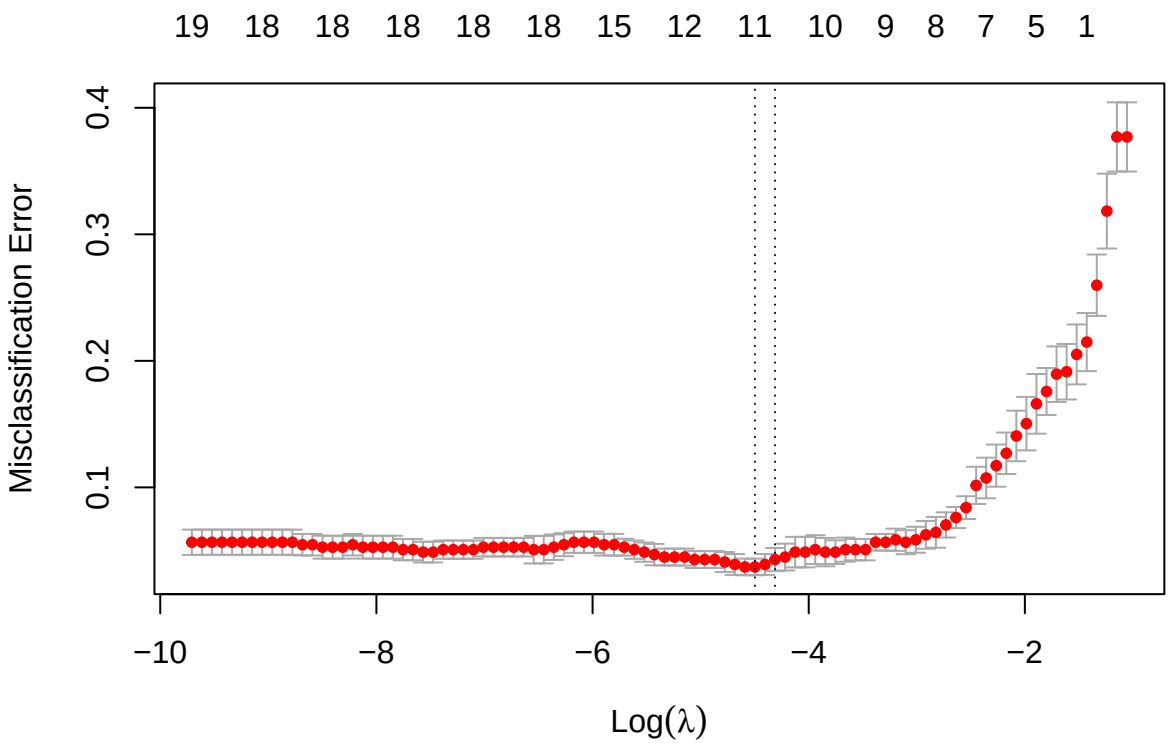
Utworzenie zbiorów treningowych i testowych dla predyktorów (`X`) i zmiennej celu (`Y`)

```
Y_train <- wdbc1 %>% filter(cv_fold != 0) %>% select(Diagnosis)
X_train <- wdbc1 %>% filter(cv_fold != 0) %>% select(-cv_fold) %>% select(-Diagnosis)
X_train <- as.matrix(X_train)
Y_test <- wdbc1 %>% filter(cv_fold == 0) %>% select(Diagnosis)
X_test <- wdbc1 %>% filter(cv_fold == 0) %>% select(-cv_fold) %>% select(-Diagnosis)
X_test <- as.matrix(X_test)
```

Walidacja krzyżowa dla znalezienia optymalnej wartości hiperparametru lambda, determinująca siłę penalizacji modelu

Współczynnik **alpha** wynosi 1 dla Lasso. `type.measure = "class"` oznacza, że algorytm będzie korzystał z **misclassification error** czyli 1-ACC

```
set.seed(1234)
model3_cv = cv.glmnet(x= X_train, y=Y_train$Diagnosis, family = "binomial", type.measure = "class", alp
plot(model3_cv)
```



Na wykresie zostały oznaczone przerywaną linią dwie wartości lambda:

- `lambda.min`, która oznacza minimalny średni błąd sprawdzany krzyżowo,
- `lambda.1se`, która oznacza wartość, dla której błąd walidacji mieści się w zakresie jednego błędu standardowego od minimum

Rzut okiem na współczynniki dla lambda.min

```
coef(model3_cv, s=model3_cv$lambda.min)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)   -19.391526456
## V5              7.829412467
## V9              .
## V10             .
## V12             .
## V13             0.075182332
## V14             0.032591799
## V15             .
## V16             .
## V17             4.635821202
## V18            96.958603648
## V19           -58.520139856
## V20             .
```

```
## V22             0.146941146
## V24             0.007497473
## V25            33.717524077
## V29             8.087746279
## V30             .
## outliers_numb   0.401100242
## many_outl       .
```

Rzut okiem na współczynniki dla lambda.1se

```
coef(model3_cv, s=model3_cv$lambda.1se)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)   -18.116833495
## V5              5.168712891
## V9              .
## V10             .
## V12             .
## V13             0.078652947
## V14             0.029862808
## V15             .
## V16             .
## V17             4.097295837
## V18            86.617625454
## V19           -45.947847979
## V20             .
## V22             0.135720977
## V24             0.007054802
## V25            33.428358949
## V29             6.888109966
## V30             .
## outliers_numb   0.374957452
## many_outl       .
```

Nie jestem pewna czy użyć `lambda.min` czy `lambda.1se`, dlatego dla każdej z osobna zrobię predykcje i porównam metryki

Rzut okiem na klasyfikacje dla `lambda.min`

```
model3_pred_lmin<-predict(model3_cv, newx = X_test, s = "lambda.min", standardize = TRUE, type="class")
head(model3_pred_lmin)
```

```
##      lambda.min
## [1,] "M"
## [2,] "M"
## [3,] "B"
## [4,] "M"
## [5,] "B"
## [6,] "B"
```

Dodanie kolumny z `Diagnosis` ze zbioru testowego zmiennej celu `Y_test` oraz utworzenie macierzy pomyłek i wyznaczenie współczynników **Accuracy** oraz **F1**

```
model3_pred_lmin <- as.data.frame(model3_pred_lmin)
colnames(model3_pred_lmin) <- "predicted"
model3_pred_lmin$Diagnosis <- Y_test$Diagnosis

confmatrix_model3 <- model3_pred_lmin %>% mutate_all(list(~ factor(., levels = c('M', 'B')))) %>% table
  model3_ACC <- sum(diag(confmatrix_model3)) / sum(confmatrix_model3)
  pre <- confmatrix_model3[1, 1] / sum(confmatrix_model3[1, ])
  rec <- confmatrix_model3[1, 1] / sum(confmatrix_model3[, 1])
  model3_F1 <- 2 * pre * rec / (pre + rec)

model_3_summary <- list("model 3 z lambda.min", model3_ACC, model3_F1)
models_summary[3,] <- model_3_summary
```

Klasyfikacja dla `lambda.1se`

```
model3_pred_lse<-predict(model3_cv, newx = X_test, s = "lambda.1se", standardize = TRUE, type="class")

model3_pred_lse <- as.data.frame(model3_pred_lse)
colnames(model3_pred_lse) <- "predicted"
model3_pred_lse$Diagnosis <- Y_test$Diagnosis

confmatrix_model3 <- model3_pred_lse %>% mutate_all(list(~ factor(., levels = c('M', 'B')))) %>% table(
  model3_ACC <- sum(diag(confmatrix_model3)) / sum(confmatrix_model3)
  pre <- confmatrix_model3[1, 1] / sum(confmatrix_model3[1, ])
  rec <- confmatrix_model3[1, 1] / sum(confmatrix_model3[, 1])
  model3_F1 <- 2 * pre * rec / (pre + rec)

model_3_summary <- list("model 3 z lambda.1se", model3_ACC, model3_F1)
models_summary[4,] <- model_3_summary
```

```
models_summary
```

```
##                   model       ACC        F1
## 1               model 1 0.9648427 0.9517220
## 2               model 2 0.9667920 0.9533436
## 3 model 3 z lambda.min 0.9649123 0.9473684
## 4 model 3 z lambda.1se 0.9649123 0.9473684
```

Dla analizowanego przypadku, okazuje się, że nie ma różnicy pomiędzy modelem z zastosowaniem `lambda.min` i `lambda.1se`. Co ciekawe, dla modelu z regularyzacją Lasso otrzymano nieco gorsze metryki `Accuracy` i `F1` niż dla regresji krokowej.

WNIOSKI

W niniejszej pracy zostały utworzone 3 modele bazujące na regresji logistycznej, z czego każdy inną metodą.

W pierwszej części przeprowadzono eksploracyjną analizę danych w celu sprawdzenia czy klasy w zmiennej celu są względnie zbalansowane, czy w zbiorze nie ma brakujących danych oraz czy pojawiają się outliery. Następnie "oczyszczono" dane poprzez dodanie kolumny z liczbą outlierów w każdej obserwacji oraz zastępując punkty oddalone medianami dla każdej zmiennej `V`. Nie usunięto punktów oddalonych, ponieważ

występowały one w około 30% obserwacji, byłaby to olbrzymia strata informacji oraz w przypadku analizowanego zagadnienia, tego typu anomalie są uzasadnione. Pod koniec, sprawdzono współliniowość Wśród zmiennych `V` i na podstawie współczynnika `VIF` zredukowano liczbę tych predyktorów z 30 do 17 (nie licząc zmiennych dodanych w trakcie analizy).

W drugiej części, zaproponowano klasyczny model regresji logistycznej, w którym optymalizowano liczbę predyktorów poprzez stopniowe usuwanie zmiennych o najmniejszej istotności. W ten sposób pozostawiono jedynie predyktory o p-value>0.05. Histogram dla tego modelu pokazał, że separacja klas zmiennej celu jest bardzo wyraźna i gdyby nie fakt, że przeprowadzono walidację krzyżową, możnaby podejrzewać, że doszło do overfittingu. Jednak ze względu na przeprowadzoną cross-validation, zakładam, że model jest stabilny i wyznaczone metryki (ACC = 0.9648, F1 = 0.9517) nie są zawyżone i wskazują na wysoką dokładność klasyfikacji modelu. Drugi model zbudowano z wykorzystaniem regresji krokowej, która stopniowo usuwała predyktory z "pełnego" modelu i minimalizując współczynnik Akaike'go (AIC) znalazła optymalne zmienne dla klasyfikatora. W walidacji krzyżowej, wyznaczono metryki ACC = 0.9668 i F1 = 0.9533, co świadczy o większej nieco dokładności modelu. Ostatni model zbudowano wykorzystując regularyzację Lasso. W walidacji krzyżowej znaleziono optymalne wartości `lambda` oraz porównano dwie predykcje z wykorzystaniem `lambda.min` oraz `lambda.1se`, ponieważ nie było pewności który współczynnik wykorzystać. Dla obu metryki były takie same i wynosiły: ACC = 0.9649 i F1 = 0.9474. Są to wartości nieco gorsze niż dla modelu 2. Teoretycznie model z regularyzacją powinien być najdokładniejszy. Otrzymane wyniki mogą wynikać z faktu, że zbudowano go na "przerobionych" danych. Być może gdyby został zbudowany na danych surowych bez wcześniejszej manipulacji, byłby lepiej dopasowany. Być może po drodze także zostały popełnione błędy wynikające z faktu, że korzystano poraz pierwszy z pakietu `glmnet` oraz samo modelowanie z regularyzacją Lasso nie jest jeszcze dla mnie tak oczywiste jak poprzednie metody.

Podsumowując, wybrano model 2 z regresją krokową jako **najlepszy** klasyfikator, ponieważ ma najwyższe wybrane metryki oraz jest dla mniej najbardziej zrozumiały. Nawet gdyby model 3 z regularyzacją Lasso miał najkorzystniejsze metryki, nie zastosowałabym go jako klasyfikator, ponieważ mam wątpliwości co do jego budowy i działania.