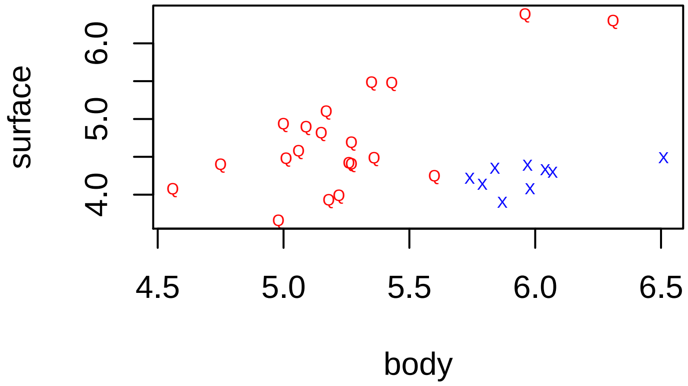
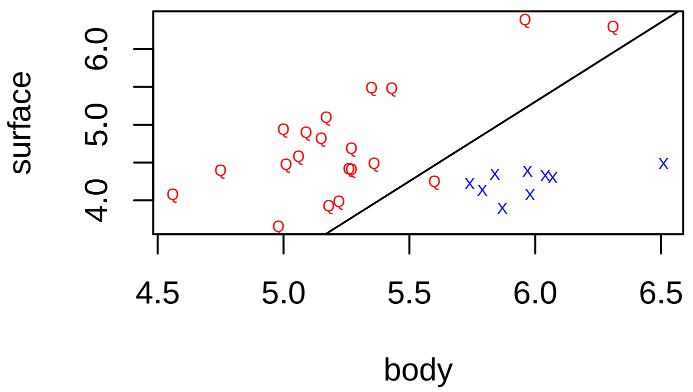
Zadanie 1.1

```
# 1.1

e <- read.table("http://www.ipipan.eu/~teisseyrep/TEACHING/DM/DANE/earthquake.txt",
    header = TRUE)
head(e)

##     popn body surface
## 1 equake 5.60    4.25
## 2 equake 5.18    3.93
## 3 equake 6.31    6.30
## 4 equake 5.36    4.49
## 5 equake 5.96    6.39
## 6 equake 5.26    4.42

attach(e)

# a)

n <- nrow(e)
ile <- table(popn)
ile

## popn
##  equake explosn
##      20       9

nq <- ile[1]
nx <- ile[2]

kol <- c(rep("red", nq), rep("blue", nx))
styl <- c(rep("Q", nq), rep("X", nx))

plot(body, surface, col = kol, pch = styl, cex = 0.5)
```



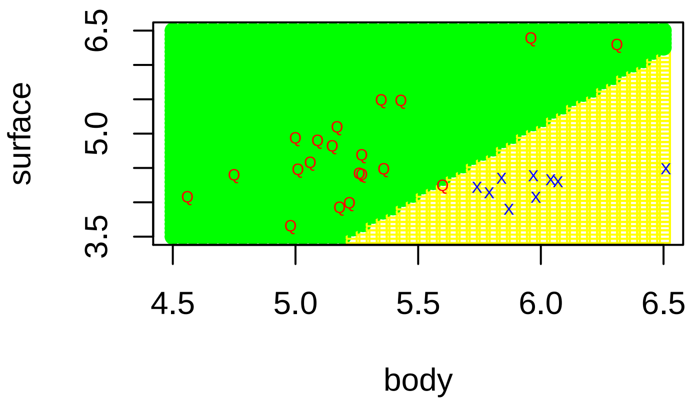

```
# b)

# macierze kowariancji w klasach:

eq <- e[popn == "equake", ]
ex <- e[popn == "explosn", ]

(cov_q <- cov(eq[, 2:3]))

##           body surface
## body    0.1478   0.200
## surface 0.2000   0.527
```

```r
(cov_x <- cov(ex[, 2:3]))

##             body surface
## body     0.05226 0.02346
## surface 0.02346 0.03253

# macierz kowariancji wewnatrzgrupowej:

(w <- ((nq - 1) * cov_q + (nx - 1) * cov_x)/(n - 2))

##            body surface
## body    0.1195  0.1477
## surface 0.1477  0.3805

# pierwszy wektor kanoniczny:

(sr_q <- apply(eq[, 2:3], 2, mean))

##    body surface
##   5.249   4.740

(sr_x <- apply(ex[, 2:3], 2, mean))

##    body surface
##   5.979   4.244

(a <- solve(w) %*% t(t(sr_x - sr_q)))

##            [,1]
## body    14.832
## surface -7.059

# prosta rozdzielajaca klasy:

z <- 1/2 * (sr_x + sr_q)

wsp_kier <- -a[1]/a[2]
wyr_woln <- t(a) %*% t(t(z))/a[2]

plot(body, surface, col = kol, pch = styl, cex = 0.5)
abline(wyr_woln, wsp_kier)
```




```r
# c)

library("MASS")

l <- lda(popn ~ ., data = e)
l$scaling  # pierwszy wektor kanoniczny (wyszedl troche inny niz
```

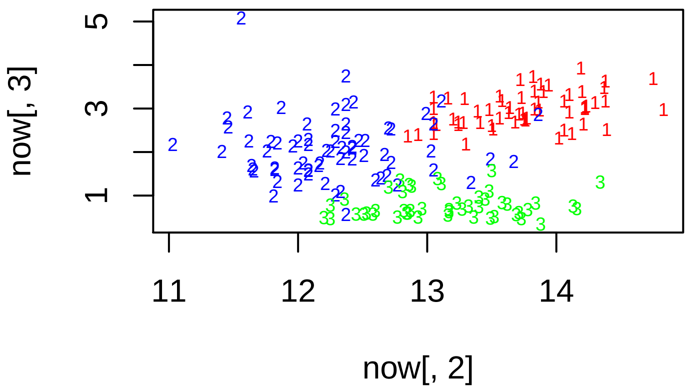
```
##            LD1
## body     3.919
## surface -1.865

# liczylismy recznie, ale wystarczy by byl proporcjonalny)

# d)

x_siat <- seq(4.5, 6.5, length.out = 50)
y_siat <- seq(3.5, 6.5, length.out = 50)
siatka <- expand.grid(body = x_siat, surface = y_siat)

l_pred <- predict(l, siatka)$class

kol2 <- ifelse(l_pred == "equake", "green", "yellow")
styl2 <- ifelse(l_pred == "equake", 19, 22)
plot(siatka, pch = styl2, col = kol2)
text(e$body, e$surface, styl, col = kol, cex = 0.5)
```




```
# e)

zm_kanon <- t(a) %*% t(e[, 2:3])
prog <- t(a) %*% z

pr <- ifelse(as.numeric(zm_kanon) - prog > 0, "explosn", "equake")
pr == as.character(e$popn)  # widac, ze tylko pierwsza zle sklasyfikowana

##  [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

# tabela reklasyfikacji

pred <- predict(l, newdata = e)$class
table(popn, pred)

##          pred
## popn      equake explosn
##   equake      19       1
##   explosn      0       9

# f)

x0 <- as.data.frame(t(c(body = 6, surface = 4)))
predict(l, x0)$class  # przypisujemy do klasy explosn

## [1] explosn
## Levels: equake explosn
```

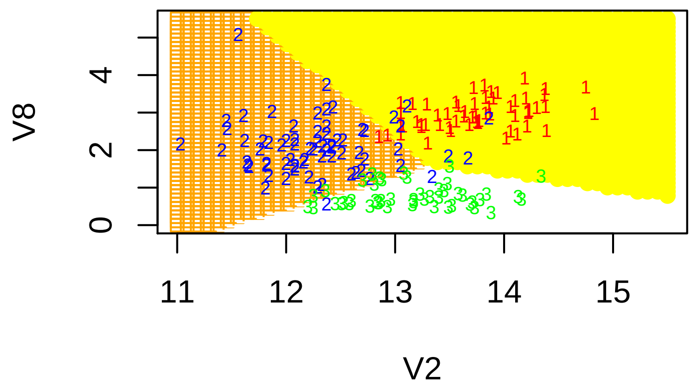
Zadanie 1.3

```
# 1.3

w <- read.table("http://www.ipipan.eu/~teisseyrep/TEACHING/DM/DANE/wine.data",
    sep = ",")
head(w)

##   V1    V2   V3   V4   V5  V6   V7   V8   V9  V10  V11  V12  V13  V14
## 1  1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92 1065
## 2  1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40 1050
## 3  1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17 1185
## 4  1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45 1480
## 5  1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93  735
## 6  1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85 1450

attach(w)

# a)

now <- w[, c(1, 2, 8)]
n1 <- length(which(now[, 1] == 1))
n2 <- length(which(now[, 1] == 2))
n3 <- length(which(now[, 1] == 3))
n <- nrow(now)

kol <- c(rep("red", n1), rep("blue", n2), rep("green", n3))
styl <- c(rep("1", n1), rep("2", n2), rep("3", n3))

plot(now[, 2], now[, 3], col = kol, pch = styl, cex = 0.6)
```

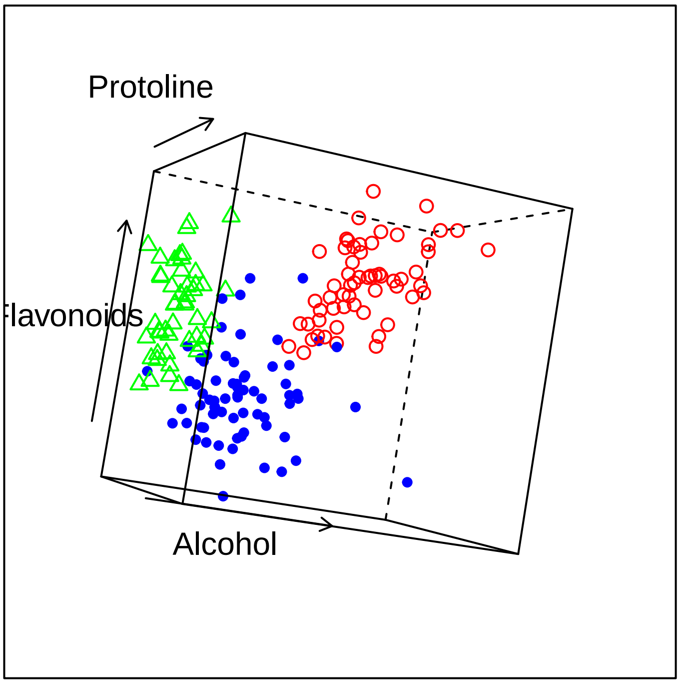



```
# b)

s <- sample(1:n, n/2)
tren <- w[s, ]
test <- w[-s, ]

l <- lda(V1 ~ V2 + V8, data = tren)
l_pred <- predict(l, newdata = test)$class
(t <- table(test$V1, l_pred))

##    l_pred
##      1  2  3
##   1 24  0  0
##   2  5 28  0
##   3  0  3 29
```

```r
# procent poprawnej klasyfikacji:

sum(diag(t))/nrow(test) * 100

## [1] 91.01

# c)

x_siat <- seq(11, 15.5, length.out = 50)
y_siat <- seq(0, 5.5, length.out = 50)
siatka <- expand.grid(V2 = x_siat, V8 = y_siat)

l_pred <- predict(l, siatka)$class

kol2 <- ifelse(l_pred == "1", "yellow", ifelse(l_pred == "2", "orange", "white"))
styl2 <- ifelse(l_pred == "1", 19, ifelse(l_pred == "2", 22, 24))
plot(siatka, pch = styl2, col = kol2)
points(now[, 2], now[, 3], col = kol, pch = styl, cex = 0.6)
```




```r
# d)

library("lattice")

l3 <- lda(V1 ~ V2 + V8 + V14, data = tren)
l3_pred <- predict(l3, newdata = test)$class
(t <- table(test$V1, l3_pred))

##    l3_pred
##      1  2  3
##   1 24  0  0
##   2  1 31  1
##   3  0  3 29

sum(diag(t))/nrow(test) * 100  # wiecej niz w poprzednim modelu

## [1] 94.38

# e)

kol <- c(rep("red", n1), rep("blue", n2), rep("green", n3))

cloud(w$V14 ~ w$V8 * w$V2, screen = c(z = -10, x = 0, y = 20), xlab = "Alcohol",
    ylab = "Flavonoids", zlab = "Protoline", col = kol, pch = c(rep(1, n1),
        rep(20, n2), rep(24, n3)))
```



Zadanie 1.4

```
# 1.4

# a)

Y1 <- c(rep(1, n1), rep(0, n2 + n3))
Y2 <- c(rep(0, n1), rep(1, n2), rep(0, n3))
Y3 <- c(rep(0, n1 + n2), rep(1, n3))

w.lm1 <- lm(Y1 ~ V2 + V8, data = w)
w.lm2 <- lm(Y2 ~ V2 + V8, data = w)
w.lm3 <- lm(Y3 ~ V2 + V8, data = w)

# b)

p1 <- predict(w.lm1, newdata = w)
p2 <- predict(w.lm2, newdata = w)
p3 <- predict(w.lm3, newdata = w)

razem <- cbind(p1, p2, p3)

p <- numeric(n)
for (i in 1:n) {
    p[i] <- which(razem[i, ] == max(razem[i, ]))
}

(t <- table(w$V1, p))

##    p
##       1  2  3
##   1 57  2  0
##   2  5 61  5
##   3  0  2 46

sum(diag(t))/n * 100

## [1] 92.13
```