Rebecca Oddone - 24/09/2021

Boxplot

The data in the file “test_2109.RData” concern 100 tumor patients. For each patient, the expression of two genes is given (columns “gene1” and “gene2”) and a binary variable describing whether the tumor is metastatic (0 = not metastatic; 1 = metastatic).

I load the file and check its content:

```
load("~/esercizi/test_2109.RData")
head(data)
```

```
##       gene1     gene2 metastatic
## 66 28.46414 10.566377          0
## 19 13.18606 17.284885          1
## 94 20.05122 12.100641          0
## 88 15.08214  9.087448          0
## 91 18.74204  8.372440          0
## 69 16.81824 10.459760          0
```

I check the class and the dimensions.

```
class(data)
```

```
## [1] "data.frame"
```

```
dim(data)
```

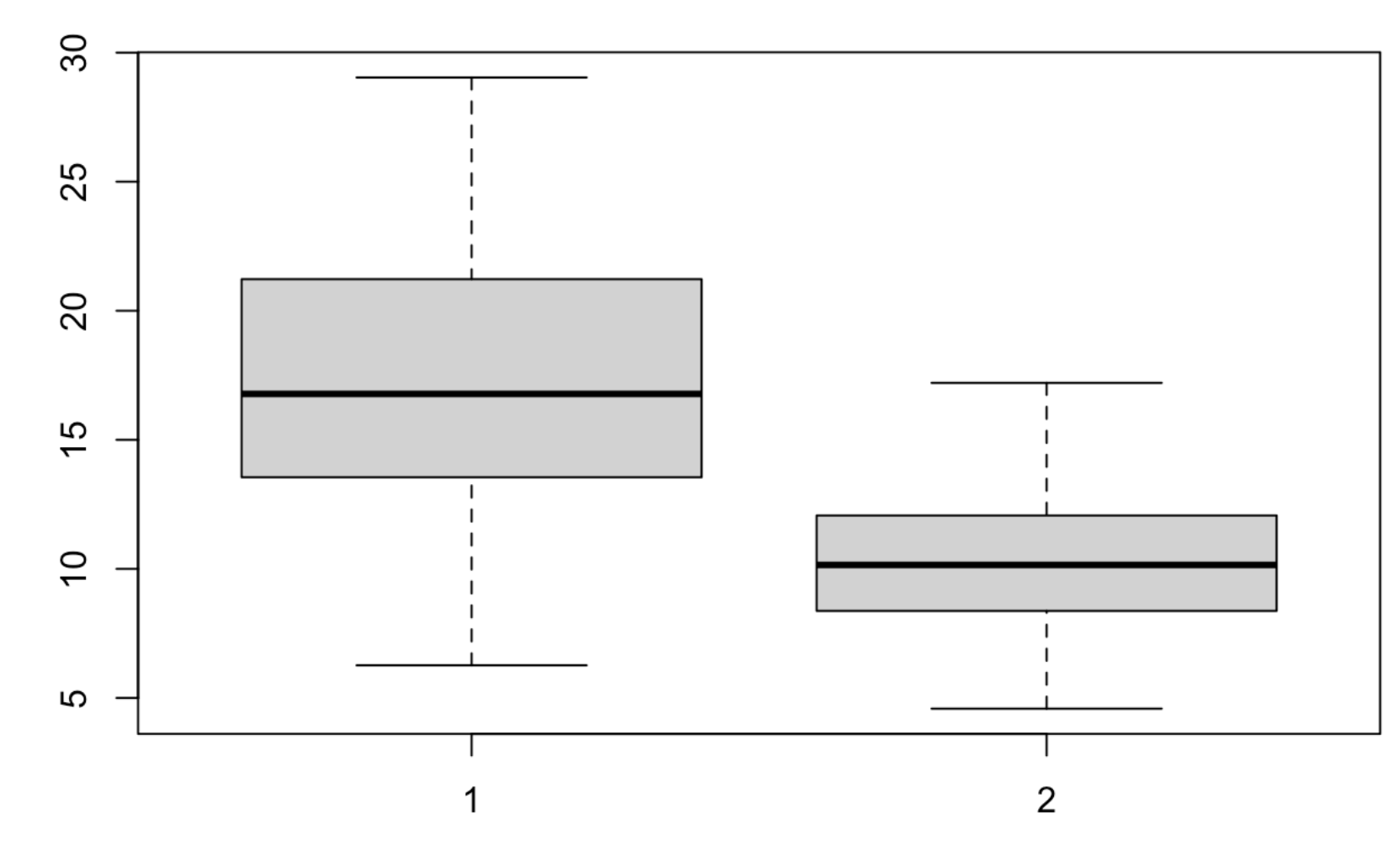
```
## [1] 100   3
```

I want to know graphically if the two genes have a different expression in metastatic and not metastatic tumors. So I divide the metastatic and not metastatic tumors and I check the expression level of the gene1 and gene2.

```
a <- data$metastatic == 0
b <- data$metastatic == 1
no_met <- data[a, ]
met <- data[b, ]
```

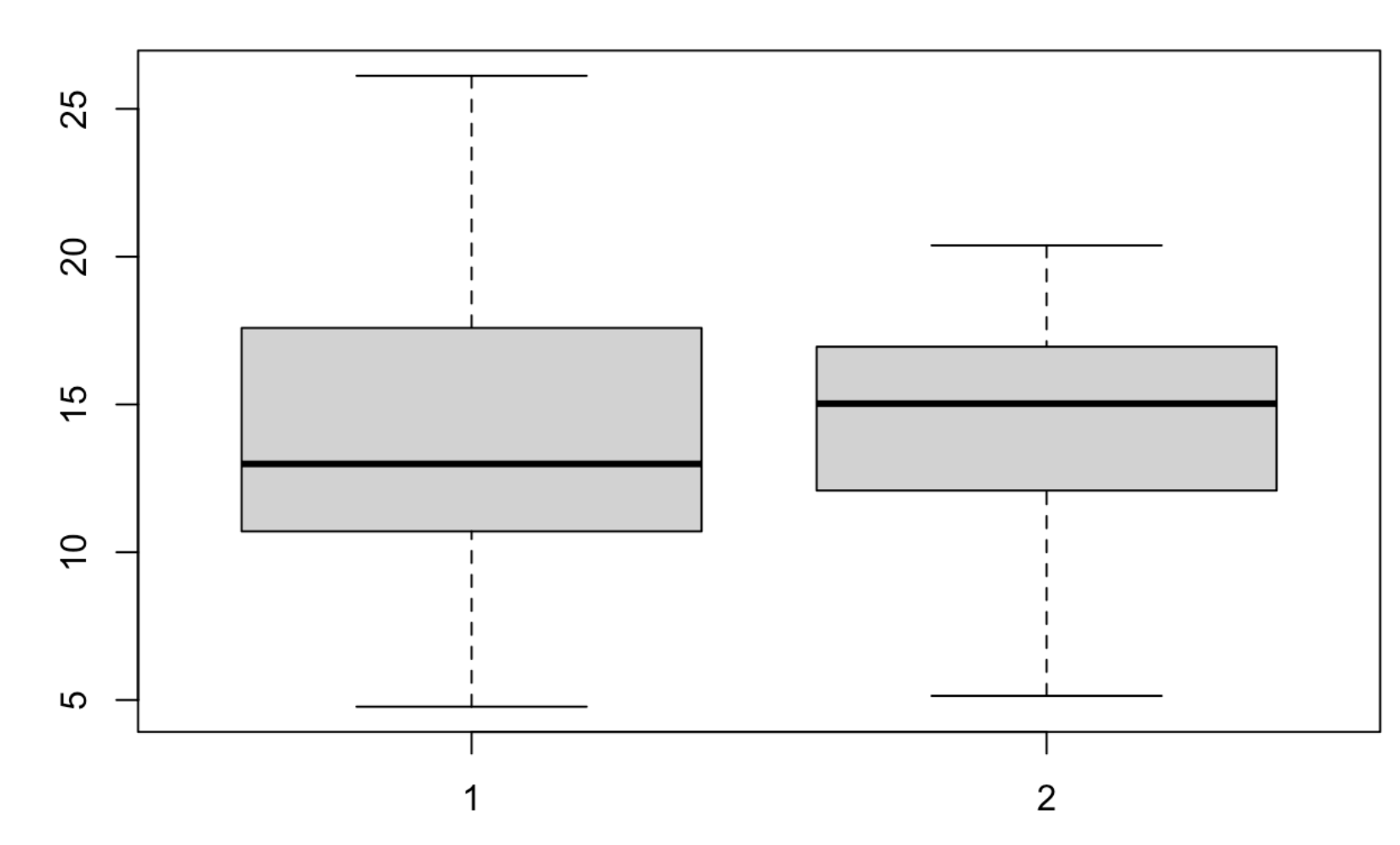
Gene1 and gene2 in not metastatic tumors.

```
boxplot(no_met$gene1, no_met$gene2)
```



Gene1 and gene2 in metastatic tumors.

```
boxplot(met$gene1, met$gene2)
```



It seems that gene1 is highly expressed in not metastatic tumors while gene2 is highly expressed in metastatic tumors, however their variability seems to be high.

So now we can use regression methods to investigate the relationships among these variables, we can apply a linear regression or a logistic regression; they essentially give the same information, but a different interpretations.

Linear regression

I can use a linear regression to predict a numerical variable from a categorical variable. I want to know if the metastaticity (categorical variable) gives some information about the expression of gene1 and gene2.

Regression of metastaticy on the gene1

```
lreg_1 <- lm(data$gene1 ~ data$metastatic)
summary_gene1 <- summary(lreg_1)
summary_gene1
```

```
##
## Call:
## lm(formula = data$gene1 ~ data$metastatic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.947  -3.415  -0.730   3.598  12.079
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      17.2111     0.6768  25.431   <2e-16 ***
## data$metastatic  -3.1718     1.2356  -2.567   0.0118 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.662 on 98 degrees of freedom
## Multiple R-squared:  0.063,  Adjusted R-squared:  0.05344
## F-statistic: 6.589 on 1 and 98 DF,  p-value: 0.01177
```

Being metastatic decreases expression to the gene1, but this has low significance, in fact the P-value is about 0.01.

Regression of the metastaticity on the gene2

```
lreg_2 <- lm(data$gene2 ~ data$metastatic)
summary_gene2 <- summary(lreg_2)
summary_gene2
```

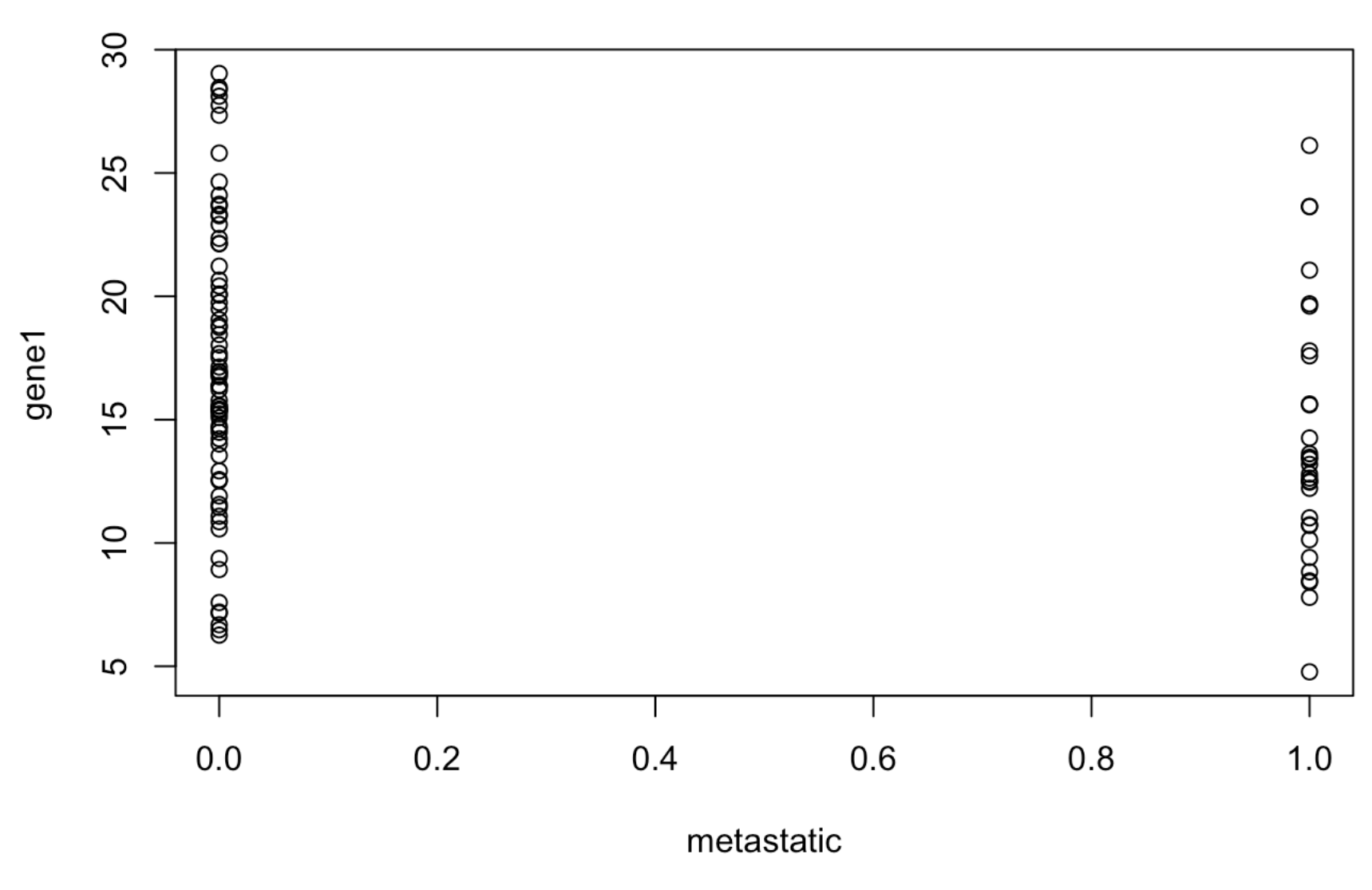
```
##
## Call:
## lm(formula = data$gene2 ~ data$metastatic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1886 -2.0078  0.0338  1.9329  6.8442
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.3606     0.3606  28.733  < 2e-16 ***
## data$metastatic   3.9692     0.6583   6.029 2.91e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.017 on 98 degrees of freedom
## Multiple R-squared:  0.2706, Adjusted R-squared:  0.2631
## F-statistic: 36.35 on 1 and 98 DF,  p-value: 2.912e-08
```

Being metastatic increases the expression of gene2 by about 3.7 with respect to being not metastatic. The standard error is about 0.66 and the P-value is 2.91e-08, so we are fairly confident that the effect is real.

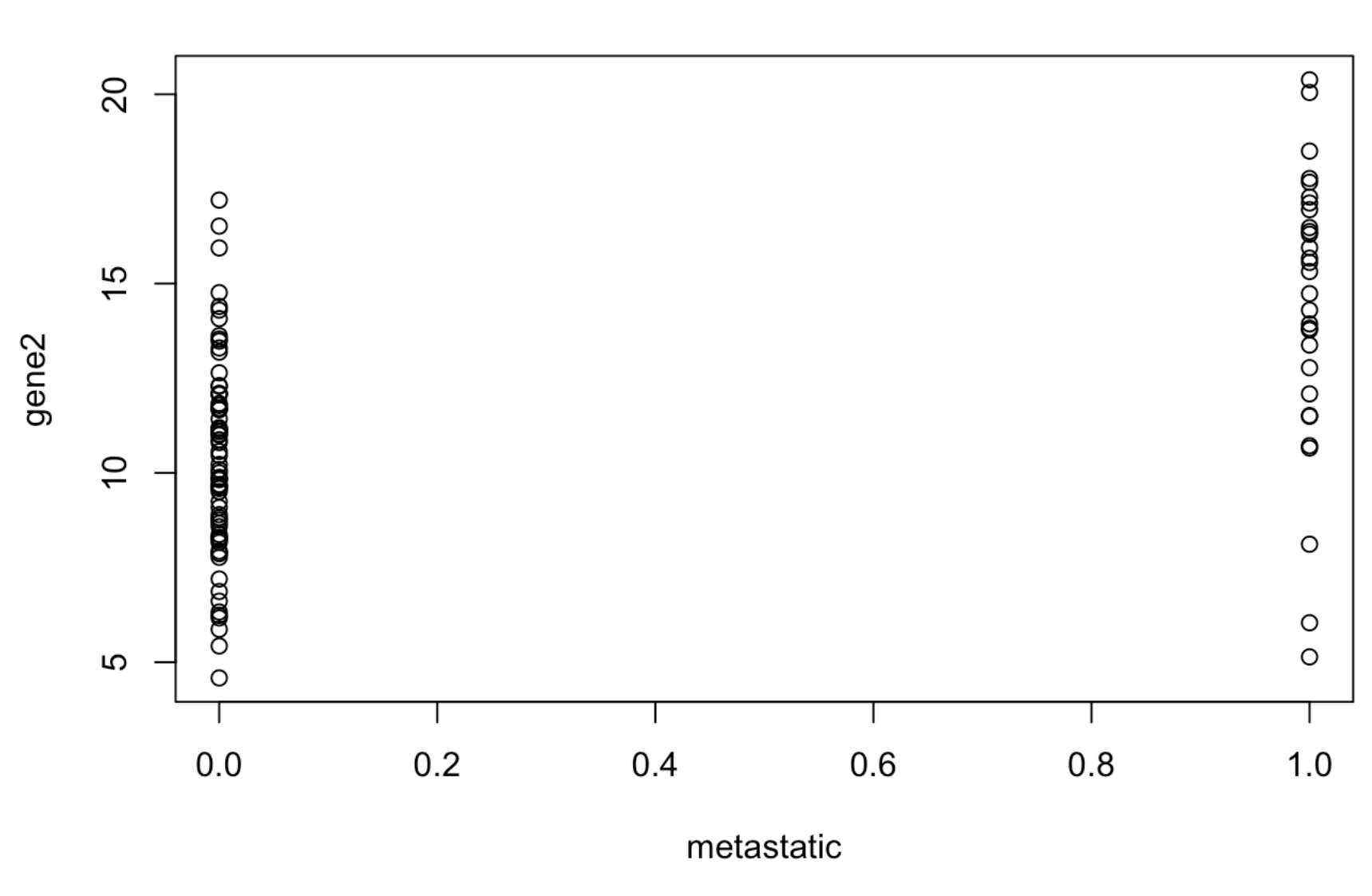
Logistic regression

We can also use logistic regression to predict a categorical variable from a numerical one. I want to understand if knowing the expression of gene1 or gene2, I can make a prediction of the metastaticity.

```
plot(data$gene1 ~ data$metastatic, xlab = "metastatic", ylab = "gene1")
```



```
plot(data$gene2 ~ data$metastatic, xlab = "metastatic", ylab = "gene2")
```



Regression of the gene1 on the metastaticity.

```
loreg_1 <- glm(data$metastatic ~ data$gene1, family = "binomial")
summary(loreg_1)
```

```
##
## Call:
## glm(formula = data$metastatic ~ data$gene1, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2288  -0.8534  -0.6838   1.2468   2.0384
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.77103    0.67554   1.141   0.2537
## data$gene1  -0.10394    0.04268  -2.435   0.0149 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 122.17  on 99  degrees of freedom
## Residual deviance: 115.54  on 98  degrees of freedom
## AIC: 119.54
##
## Number of Fisher Scoring iterations: 4
```

A decrease of the expression of gene1 has a positive effect on the probability that the tumor is metastatic. The uncertainty on this is about 0.04 and the P-value has a low significance.

Regression of the gene2 on the metastaticity.

```
loreg_2 <- glm(data$metastatic ~ data$gene2, family = "binomial")
summary(loreg_2)
```

```
##
## Call:
## glm(formula = data$metastatic ~ data$gene2, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6931  -0.6921  -0.4345   0.6904   2.7554
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.87639    1.18956  -4.940 7.81e-07 ***
## data$gene2   0.40903    0.09099   4.495 6.94e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 122.173  on 99  degrees of freedom
## Residual deviance:  92.388  on 98  degrees of freedom
## AIC: 96.388
##
## Number of Fisher Scoring iterations: 5
```

An increase expression of gene2 contributes to the probability that the tumor is metastatic rather that not metastatic. The uncertainty is about 0.09 and the P-value is 6.94e-06, so we are fairly confident that the effect is real.

Multivariable regression

```
loreg_all <- glm(data$metastatic ~ data$gene1 + data$gene2, family = "binomial")
sum_loreg_all <- summary(loreg_all)
sum_loreg_all
```

```
##
## Call:
## glm(formula = data$metastatic ~ data$gene1 + data$gene2, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7449  -0.6981  -0.4164   0.6796   2.8236
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.20424    2.04800  -3.518 0.000435 ***
## data$gene1   0.04462    0.05356   0.833 0.404789
## data$gene2   0.45939    0.11250   4.083 4.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 122.173  on 99  degrees of freedom
## Residual deviance:  91.696  on 97  degrees of freedom
## AIC: 97.696
##
## Number of Fisher Scoring iterations: 5
```

I perform the multivariable regression because I can postulate that the dependent variable (metastaticity) depends on more indipendent variables (genes). However only gene2 is significant, instead gene1 is not significant. The interesting point is that if we mantein fixed the gene1 expression, the increasing of gene2 expression contributes to the probability that the tumor is metastatic.

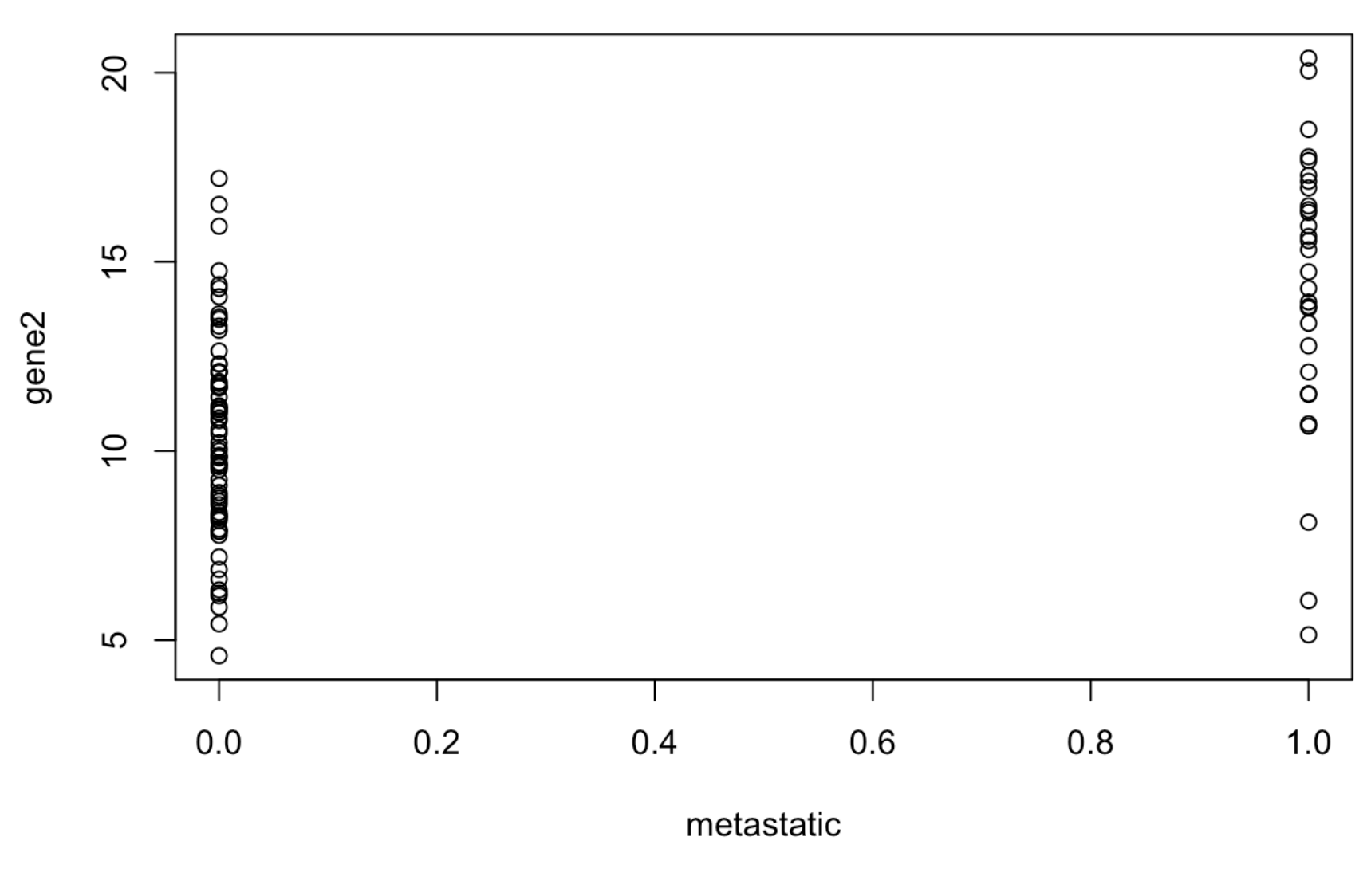
Training and testing sets

The overfitting problem can be tackled by cross-validation. The basic idea consists of dividing the data into two sets: the training set, used to build the model, and the testing set, used to validate the model.

I decide to use only gene2 because it is the most significant in relation to metastaticity.

```
data_2 <- data[1:100, 2:3]

plot(data_2$gene2 ~ data_2$metastatic, xlab = "metastatic", ylab = "gene2")
```



I take a part of the data (testset) and the model is developed using the other observations (trainset). The predictive power is evaluated on the testing set. So fitting the accidental details of the training set will not improve the performance and these are not replicated in the testing set.

```
testset <- lm(metastatic ~ gene2, data = data_2[1:30, ])
trainset <- predict(testset, newdata = data_2[31:100, ])
summary(testset)$r.squared
```

```
## [1] 0.3545656
```

```
r2 <- function(y, y_pred) 1-sum((y-y_pred)^2)/sum((y-mean(y))^2)
r2(data_2[1:30, ]$metastatic, testset$fitted.values)
```

```
## [1] 0.3545656
```

```
r2(data_2[31:100,]$metastatic, trainset)
```

```
## [1] 0.1686368
```

I can consider adding gene1, but this model is not better.

```
data_3 <- data[1:100, ]
testset_tot <- lm(metastatic ~ gene1 + gene2, data = data_3[1:30, ])
trainset_tot <- predict(testset_tot, newdata = data.frame(data_3[31:100, ]))
summary(testset_tot)$r.squared
```

```
## [1] 0.354631
```

```
r2(data_3[1:30, ]$metastatic, testset_tot$fitted.values)
```

```
## [1] 0.354631
```

```
r2(data_3[31:100, ]$metastatic, trainset_tot)
```

```
## [1] 0.1677301
```