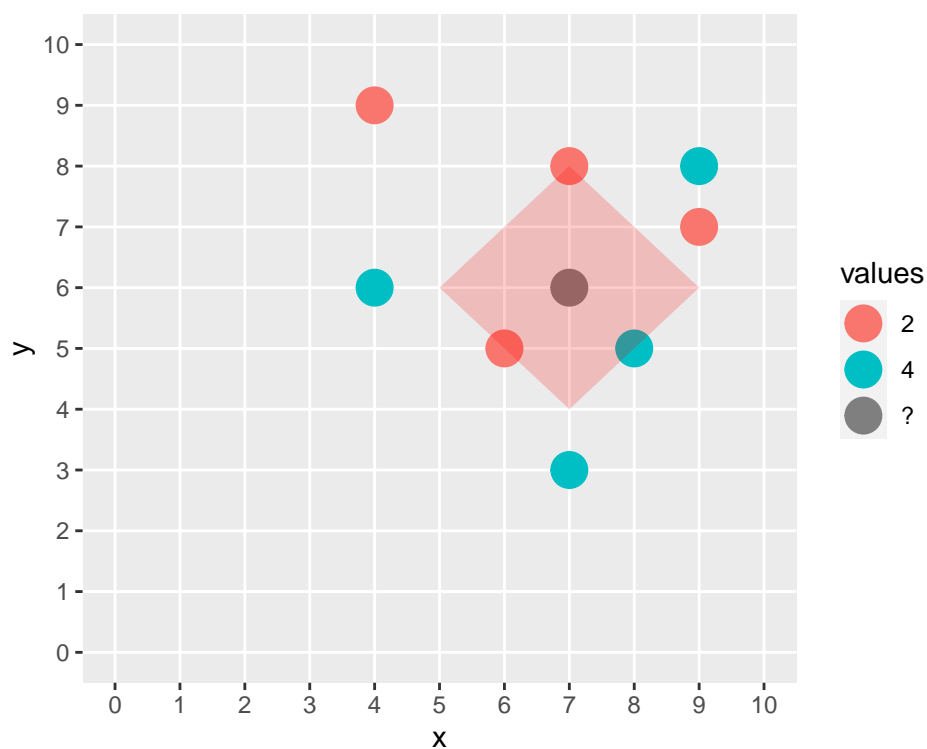


## Solution 1:

a)  $k = 3$ 

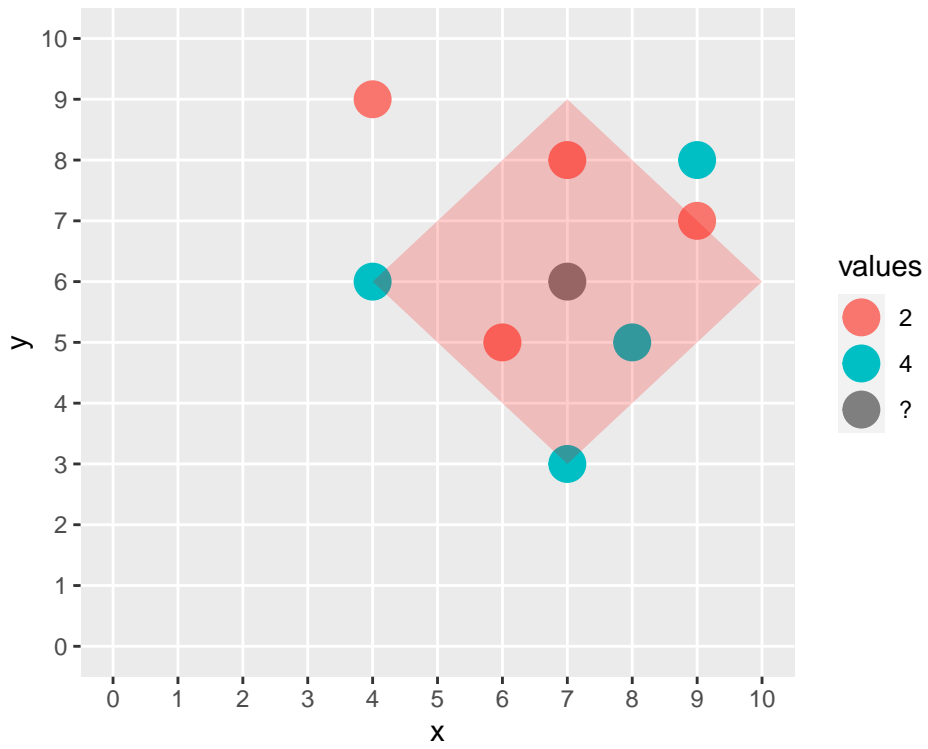
$$\hat{y} = \frac{2 + 2 + 4}{3} = \frac{8}{3} \approx 2.67$$

$$\hat{y}_{\text{weighted}} = \frac{\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 4}{\frac{3}{2}} = \frac{8}{3} \approx 2.67$$

b)  $k = 5$ 

$$\hat{y} = \frac{2 + 2 + 2 + 4 + 4 + 4}{6} = 3$$

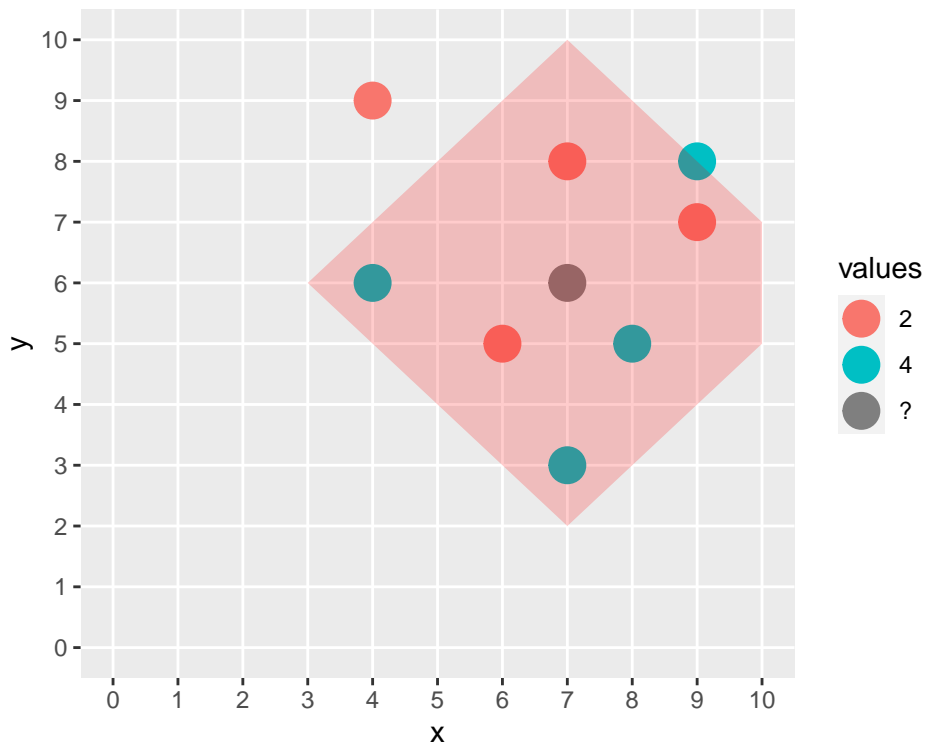
$$\hat{y}_{\text{weighted}} = \frac{\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 2 + \frac{1}{3} \cdot 2 + \frac{1}{2} \cdot 4 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 4}{\frac{5}{2}} = \frac{44}{15} \approx 2.93$$



c)  $k = 7$

$$\hat{y} = \frac{2 + 2 + 2 + 4 + 4 + 4 + 4}{7} = \frac{22}{7} \approx 3.14$$

$$\hat{y}_{\text{weighted}} = \frac{\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 2 + \frac{1}{3} \cdot 2 + \frac{1}{2} \cdot 4 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 4 + \frac{1}{4} \cdot 4}{\frac{11}{4}} = \frac{100}{33} \approx 3.03$$



## Solution 2:

a) Learning consists of *representation* (hypothesis space), *evaluation* (risk) and *optimization*.

A learner in mlr3 can be thought of as the implementation of these components, since

- a representation of the associated model learnt from the data by using the implemented optimization is stored in such a learner object,
- its performance measures can be accessed afterwards.

```
b) library(mlr3)
library(mlr3learners)

# show all available learners
mlr_learners$keys()

## [1] "classif.cv_glmnet" "classif.debug" "classif.featureless"
## [4] "classif.glmnet" "classif.kknn" "classif.lda"
## [7] "classif.log_reg" "classif.multinom" "classif.naive_bayes"
## [10] "classif.nnet" "classif.qda" "classif.ranger"
## [13] "classif.rpart" "classif.svm" "classif.xgboost"
## [16] "regr.cv_glmnet" "regr.featureless" "regr.glmnet"
## [19] "regr.kknn" "regr.km" "regr.lm"
## [22] "regr.ranger" "regr.rpart" "regr.svm"
## [25] "regr.xgboost" "surv.cv_glmnet" "surv.glmnet"
## [28] "surv.ranger" "surv.xgboost"

# see settings for a specific learner, e.g., for a regression tree
rpart_learner <- lrn("regr.rpart")
print(rpart_learner)

## <LearnerRegrRpart:regr.rpart>
## * Model: -
## * Parameters: xval=0
## * Packages: rpart
## * Predict Type: response
## * Feature types: logical, integer, numeric, factor, ordered
## * Properties: importance, missings, selected_features, weights
```

## Solution 3:

See R code