k-Nearest Neighbor (*k*-NN) as Regression

Previous class

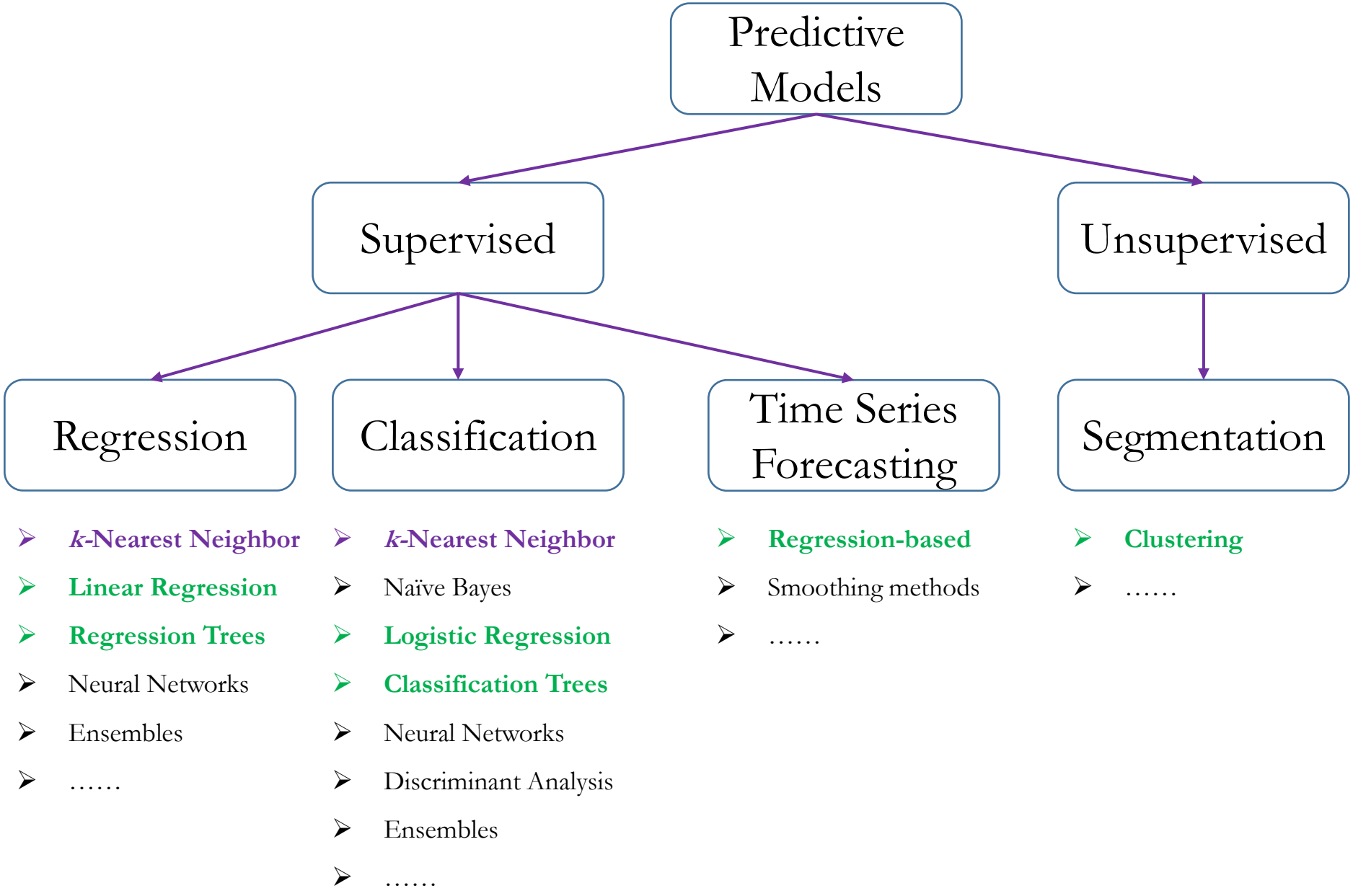
- *k*-Nearest Neighbor (*k*-NN) as Classification
- Application of *k*-NN in R/RStudio and Inference

Today's class

- *R*e-cap and application of *k*-NN as Classification
- *k*-Nearest Neighbor (*k*-NN) as Regression
- Application of *k*-NN in R/RStudio and Inference

Industry Speaker Session

- Topic:
  - Supply Chain Management in Footwear and Apparel Industry
- Mike Brewer, VP of Global Sourcing and Manufacturing, Nike
- Thursday, 14 April 2022 (Today), 6:00pm – 7:15pm
- 132 Lillis



Supervised Learning

- **Regression**
  - Goal is to predict a continuous numerical outcome
  - Predicting House price
  - Predicting patients' length of stay (LOS) in an outpatient department
  - Predicting Sales of a brick & mortar retail store based on traffic, labor ……
- **Classification**
  - Goal is to predict a categorical outcome
  - Two classes: Is the email spam or not spam?

    Is the tumor benign or malignant?

    Is the arriving patient high risk or low risk?
  - Multi-class: Classifying fruits into Apple, Orange, Banana based on shape, color…

    Classifying a new movie into one of the groups - PG, TV-14, G

k-NN

- Simple Machine Learning/Predictive algorithm
- Used for
  - Classification (of a categorical outcome)
  - Regression (of a numerical outcome)
- Method relies on finding "**similar**" observations in the data
- Referred as "**Neighbors.**"
- "**Neighbors**" are used to derive a prediction for a new observation

k-NN as Classification

- Identify ***k*** neighboring observations in the dataset that are similar to the new observation you wish to classify
- Assign the **predominant class** of neighbors to a new observation

1-NN as Classifier

- Identify ***1*** observation in the dataset that is **near** to the new observation you wish to classify
- Assign the class of neighboring observation to new observation
- Sample data with three variables V1, V2, Class

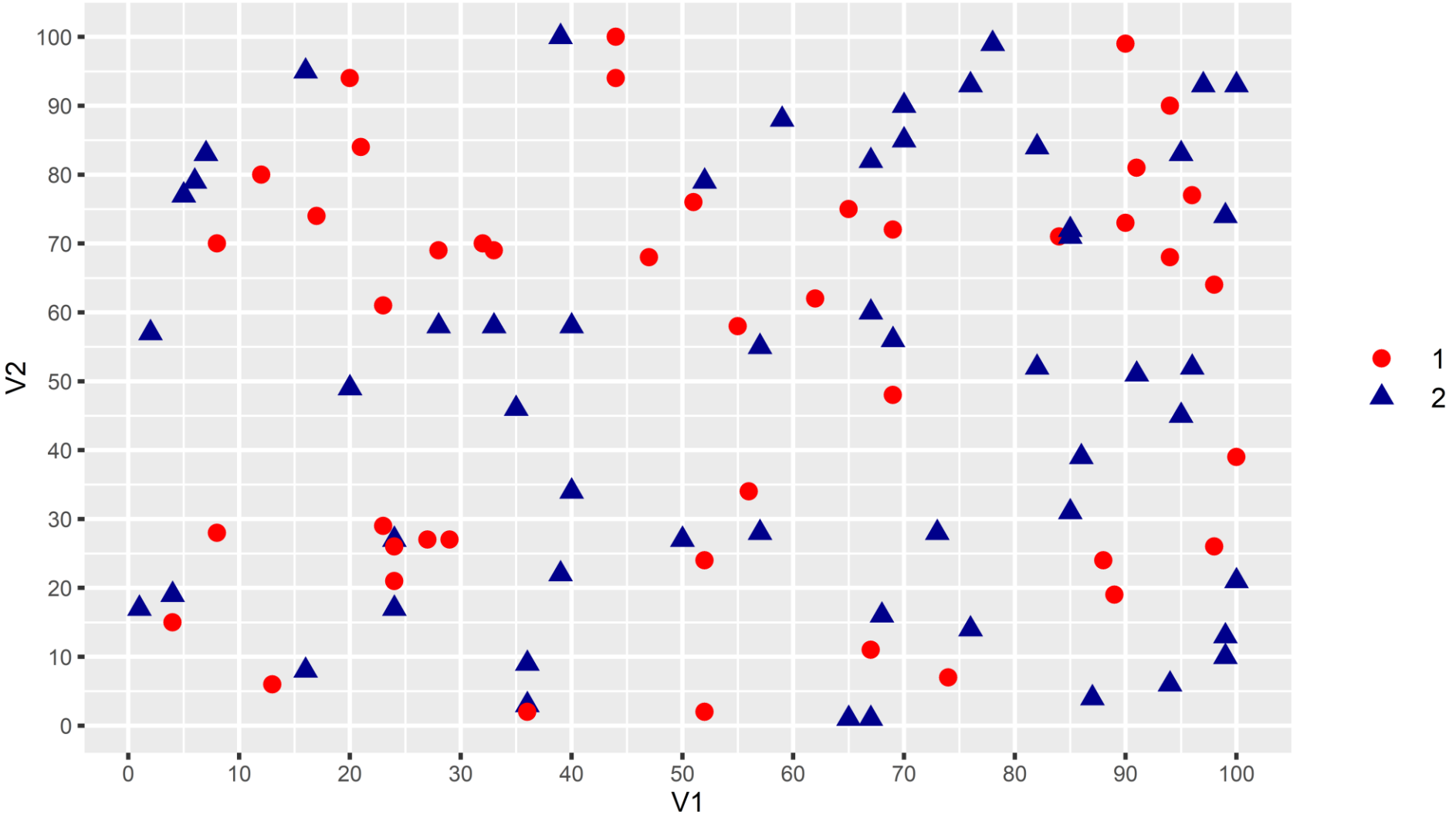
V1	V2	Class
64	94	1
18	70	2
24	9	1
46	20	2
72	91	2
66	1	1
12	11	1

⋮

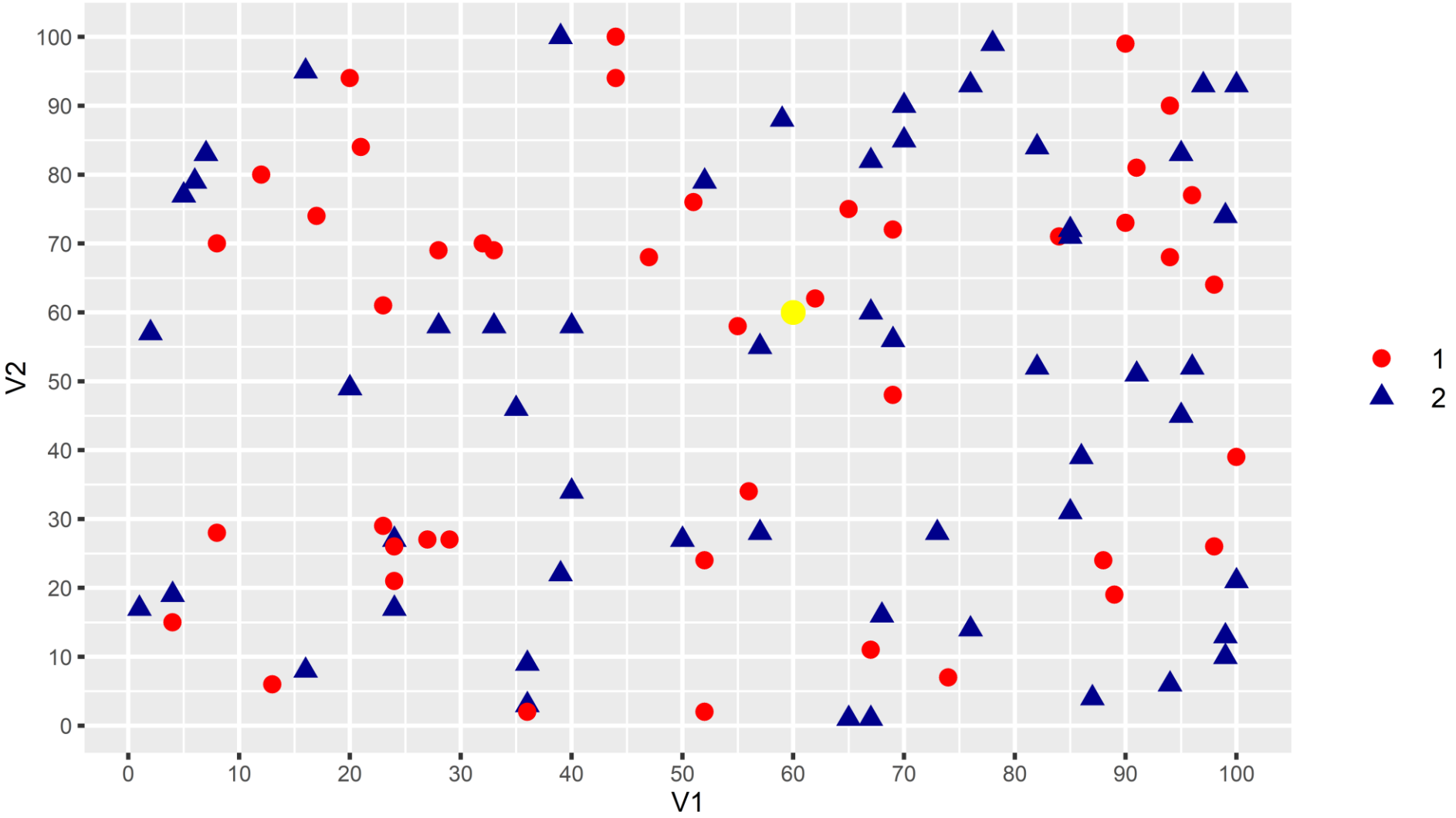
V1	V2	Class
60	60	?

New observation

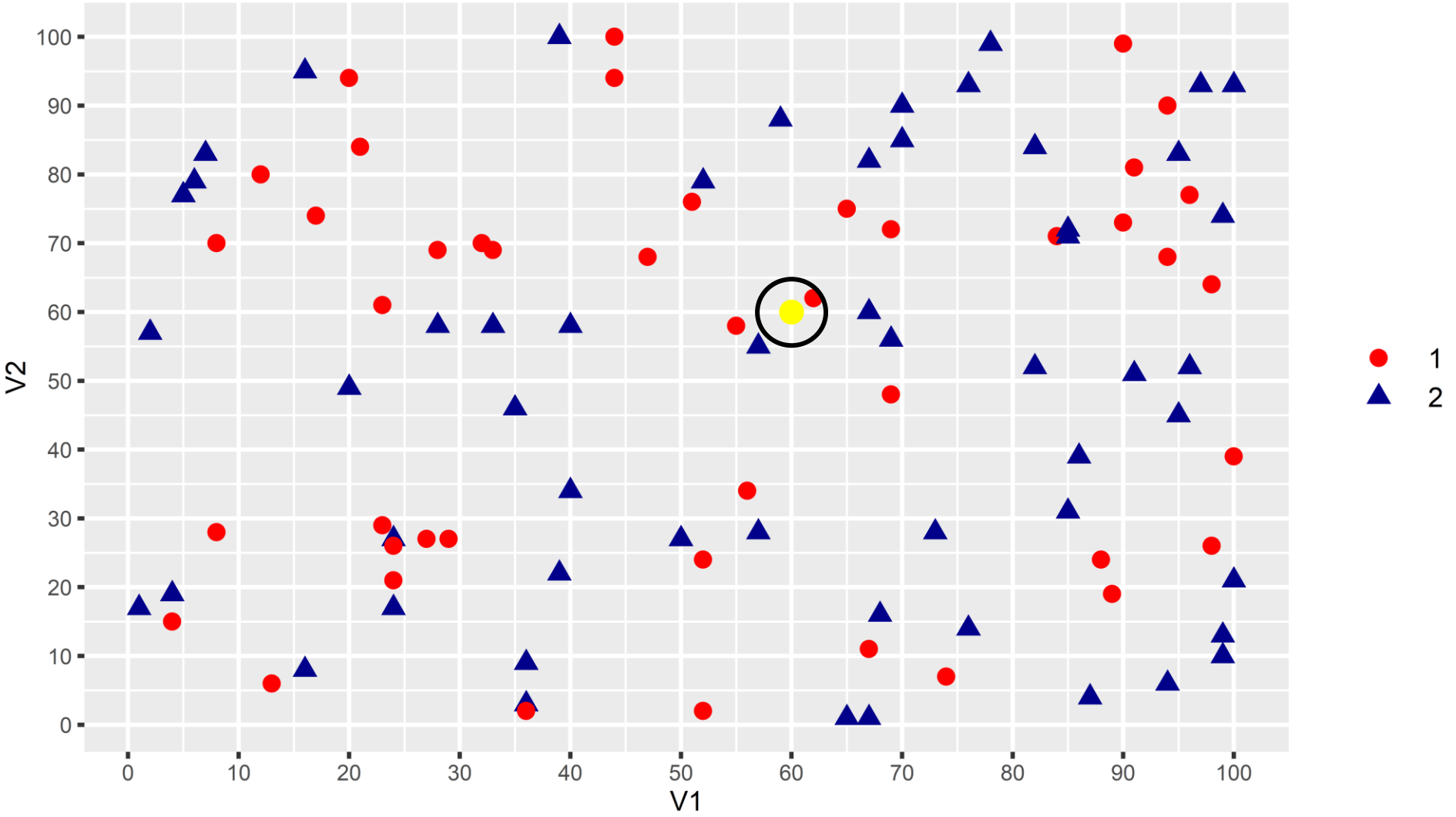
Scatter plot



New observation (yellow point)

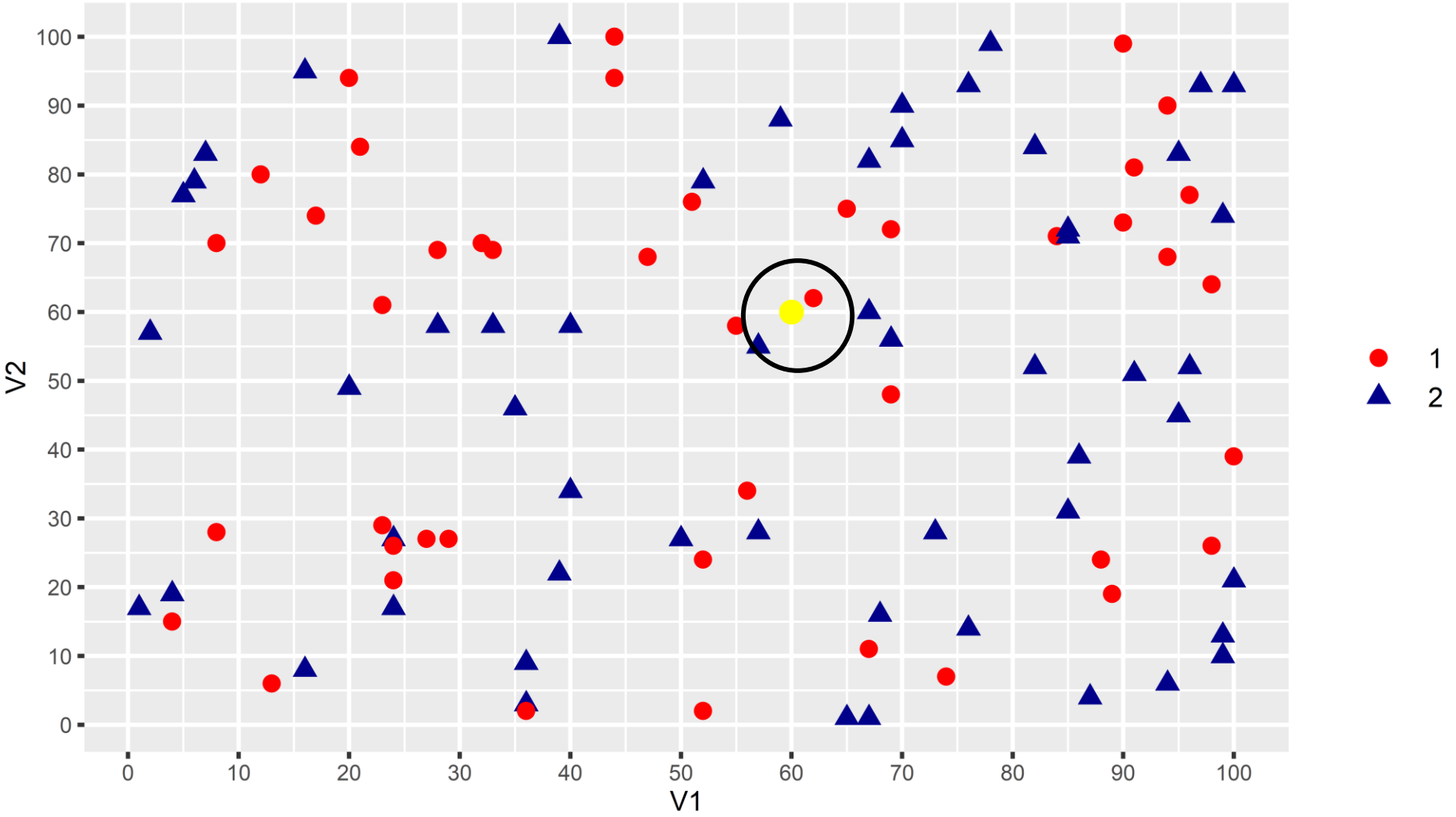


1-NN as Classifier



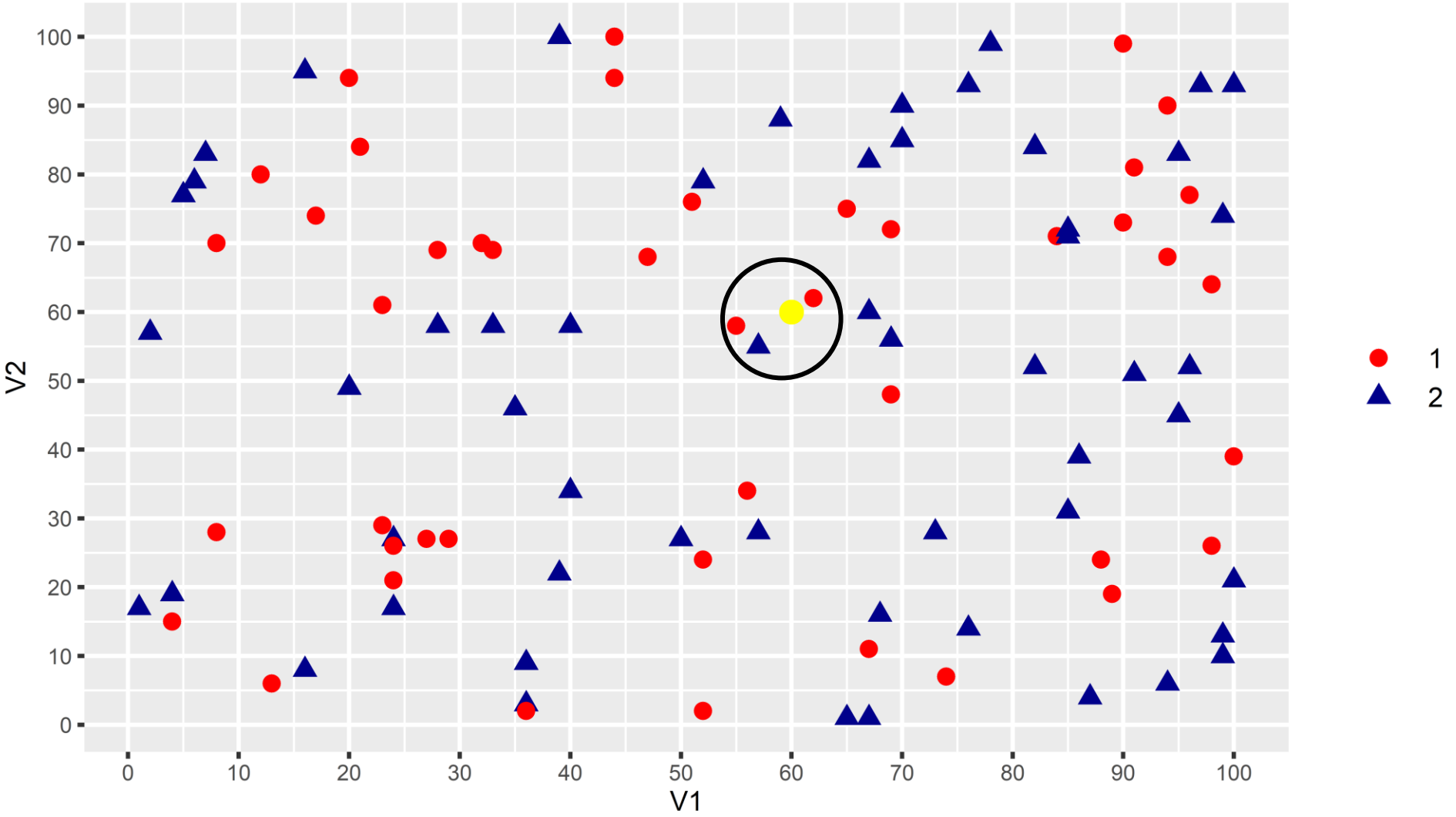
New observation prediction = **Class 1**

2-NN as Classifier



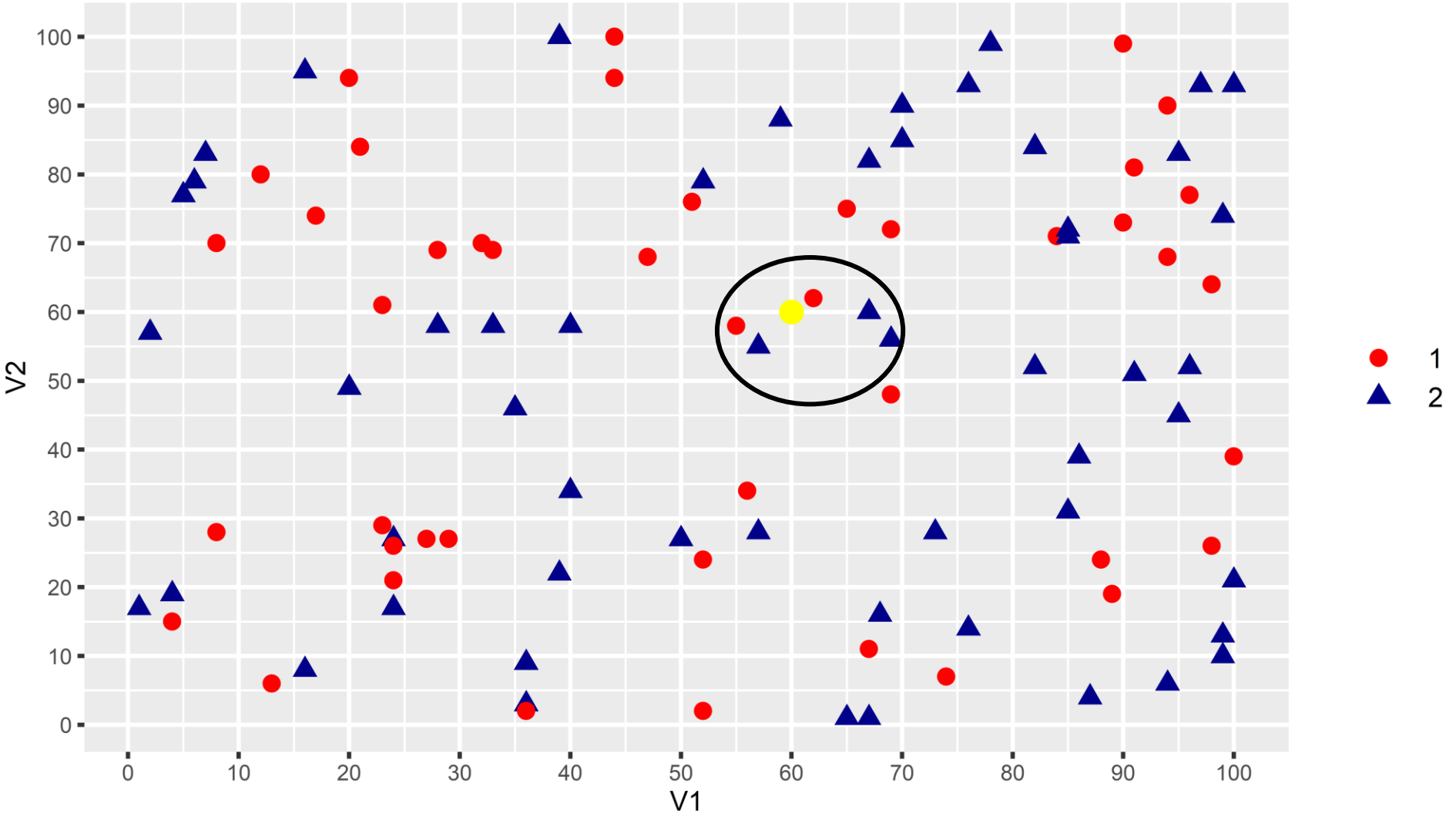
New observation prediction = **Tie**

3-NN as Classifier



New observation prediction = **Class 1 (predominant)**

5-NN as Classifier



New observation prediction = **Class 2 (predominant)**

k-NN as classification model in R

- Step 1: Main data
  - Standardize the numeric input variables
  - Convert input character variables into dummy (binary) variables
- Step 2: Pick only standardized input numeric & dummy variables in main data
  - **Standardized main data**
- Step 3: New data – prediction of interest
  - Standardize the numeric input variables
  - Convert input character variables into dummy variables
- Step 4: Pick only standardized input numeric & dummy variables in new data
  - **Standardized new data**
- Step 5: Track the output variable in the main data
  - **Main data output**
- Step 6: Execute the function "**knn**" to predict for new observation

k-NN as Regression

- Identify ***k*** observations in the dataset that are similar to the new observation you wish to predict
- Take the **average** of ***k*** observations as a prediction for new observation

1-NN as Regression

- Identify ***1*** observation in the dataset that is near to the new observation you wish to predict
- Assign the observation to new observation as a prediction
- Sample data with two variables X, Y

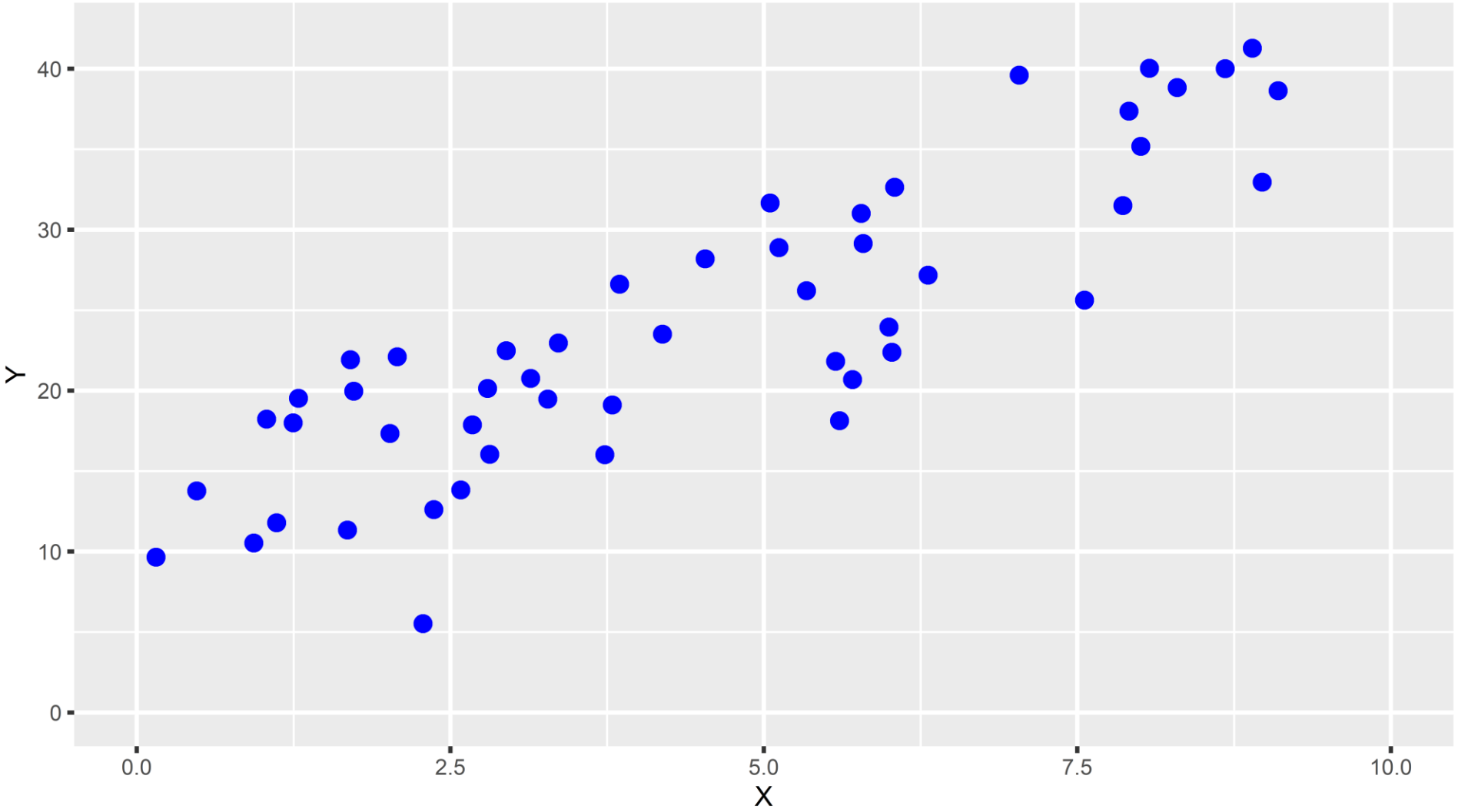
X	Y
2.573	18.887
9.667	46.964
6.619	29.495
1.150	10.620
2.271	14.267
2.472	13.381

⋮

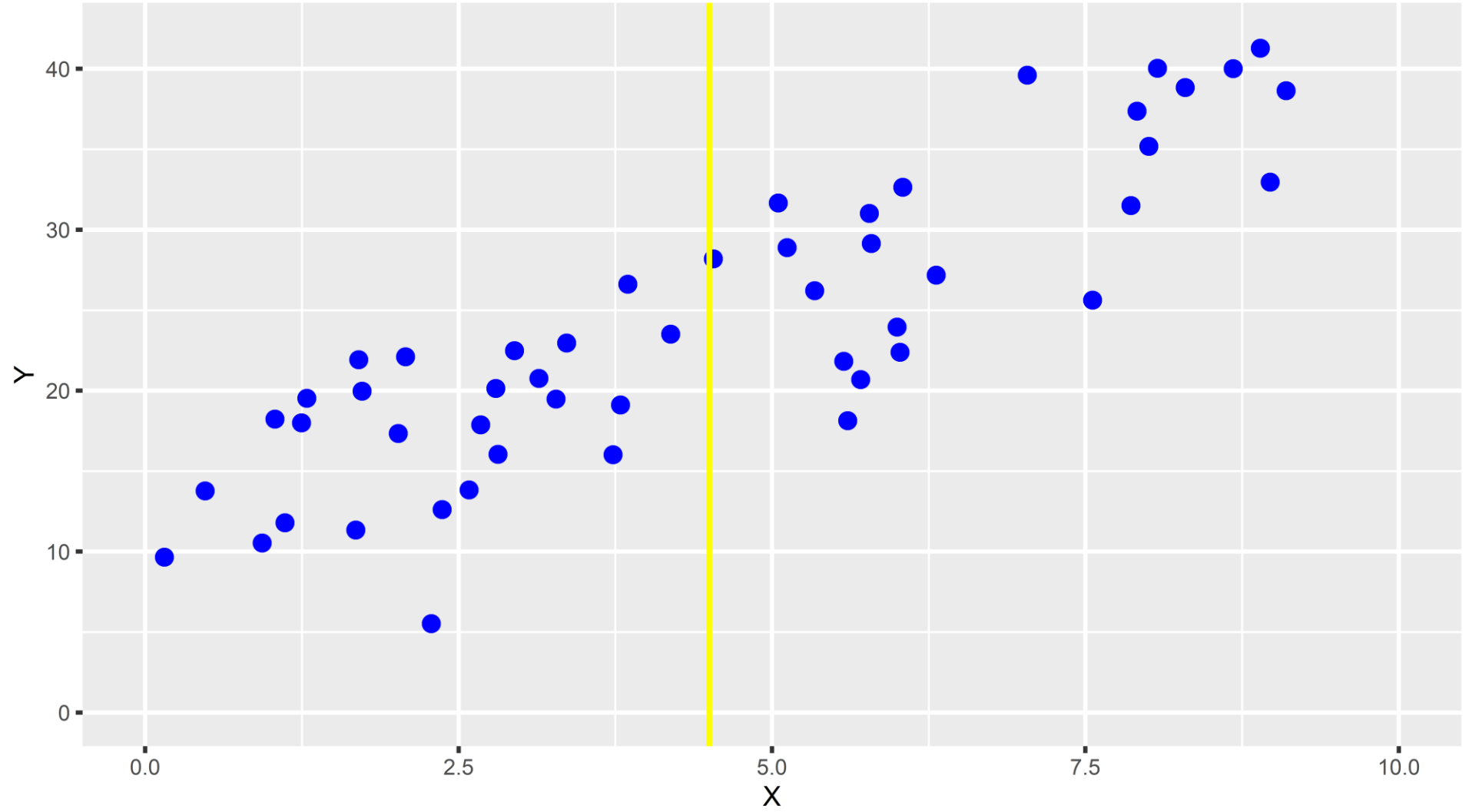
X	Y
4.5	?

New observation

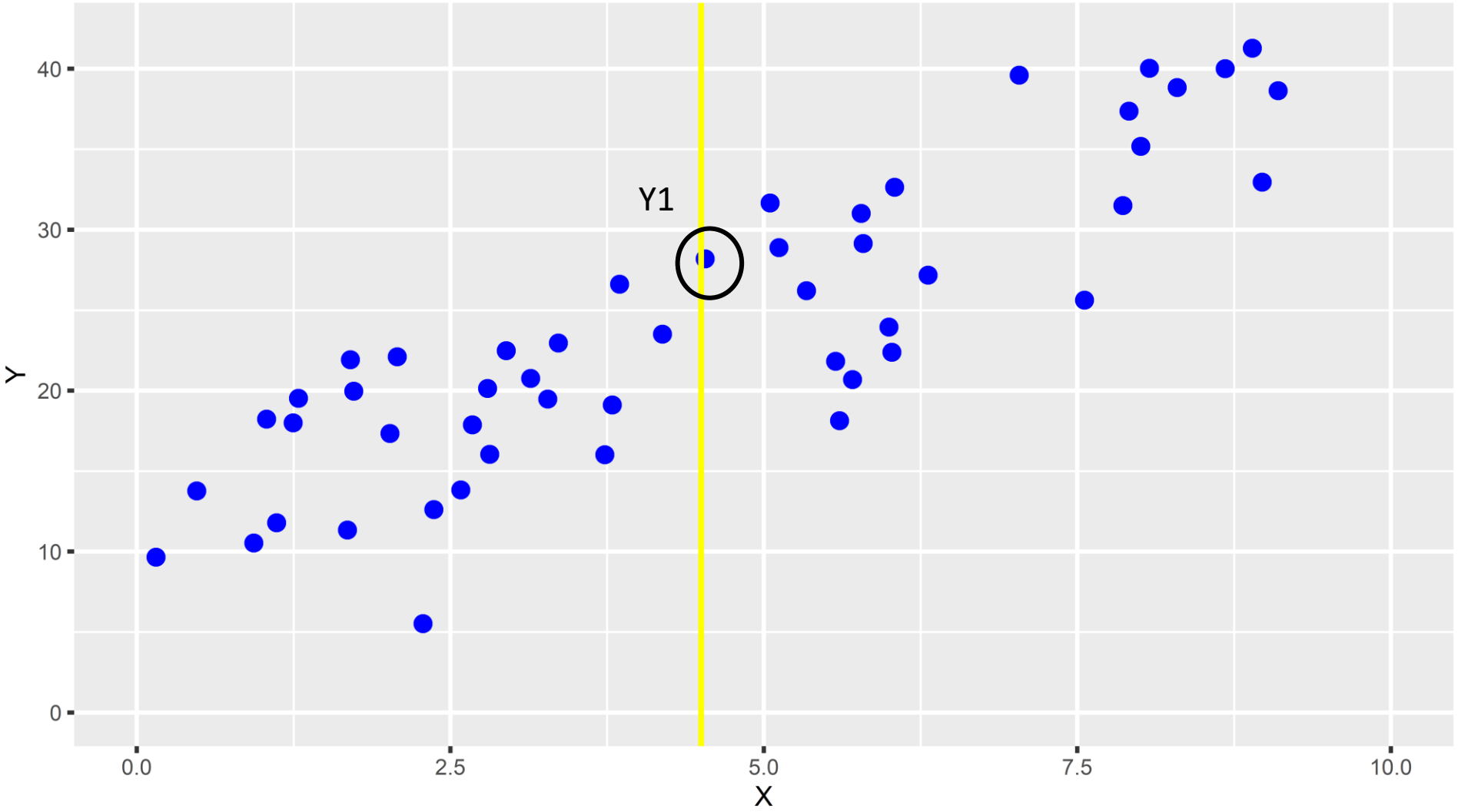
Scatter plot



New observation (yellow line)

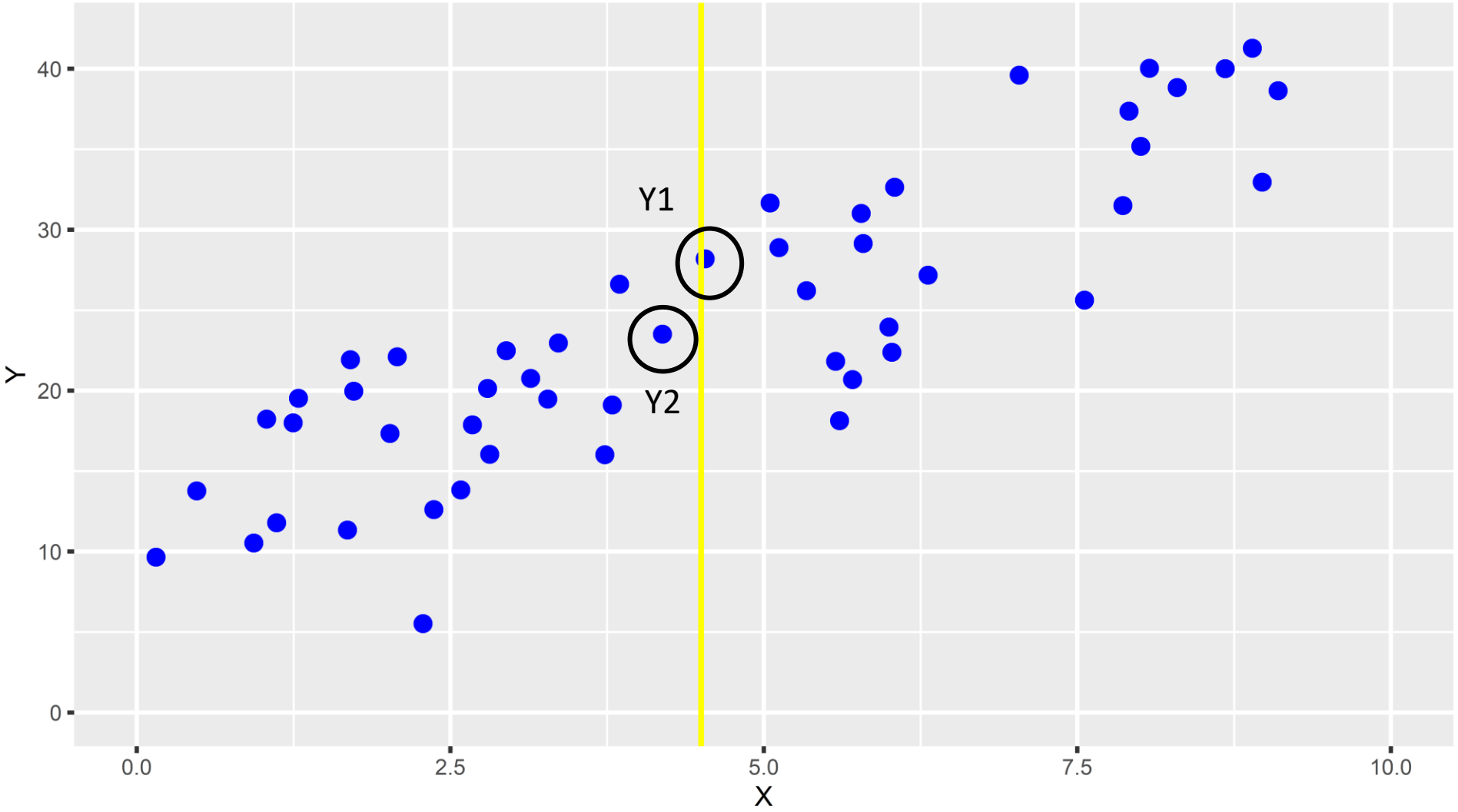


1-NN as Regression



New observation prediction = **Y1**

2-NN as Regression



$$\text{New observation prediction} = \frac{Y1+Y2}{2}$$

Data on used Toyota Corolla cars

- Data on sales of used cars in the Netherlands, late summer 2004
- Attributes
  - model: Toyota car model
  - price: offer price in euros
  - age_08_04: age in months as of August 2004
  - mfg_month : manufacturing month (1,2,3……12)
  - mfg_year: manufacturing year
  - km: accumulated kilometers on the odometer
  - fuel_type : fuel type (petrol, diesel, cng)
  - hp: horsepower

⋮

Data on used Toyota Corolla cars

- Selected variables
  - Price, age, and km

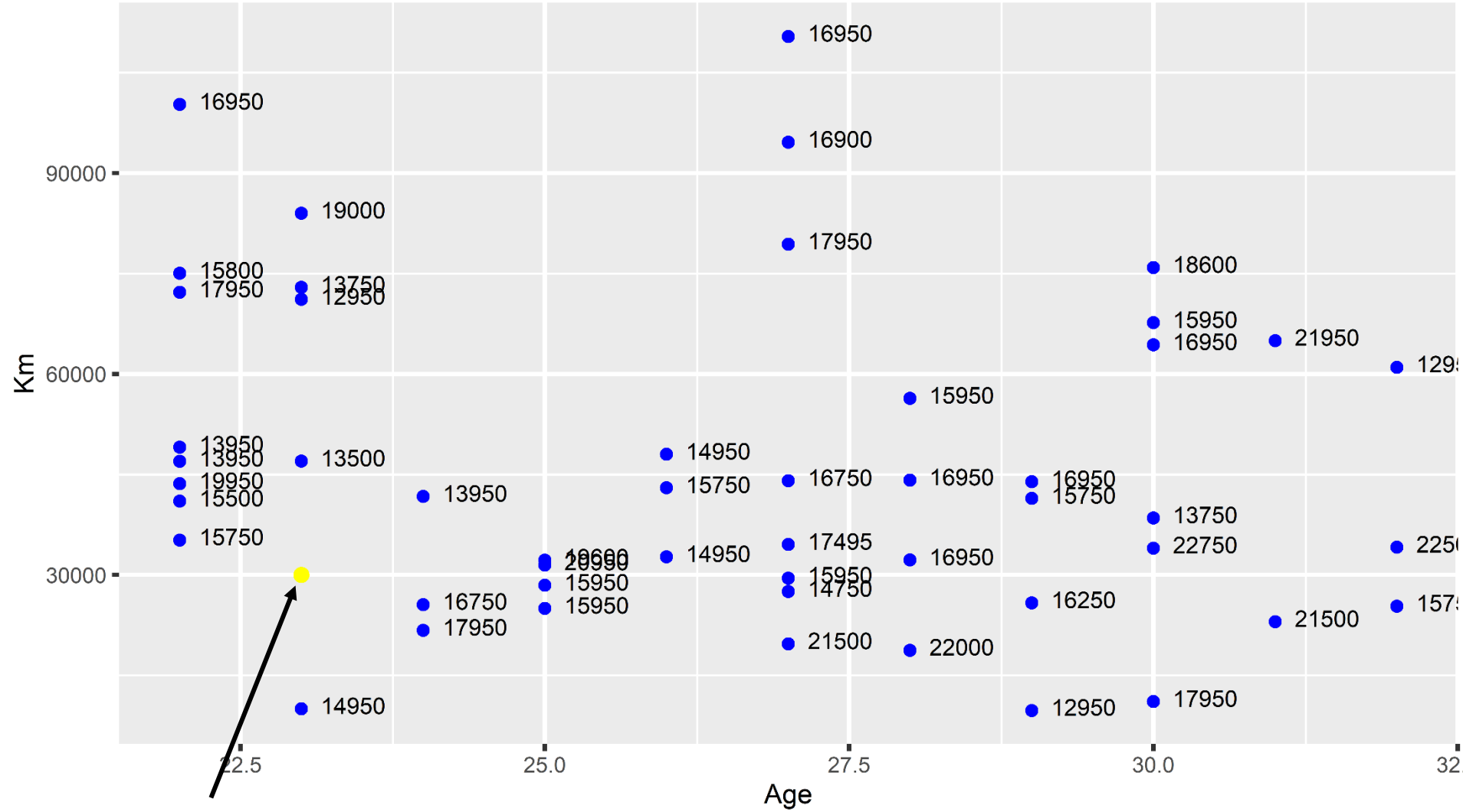
age_08_04	km	price
23	46,986	13,500
23	72,937	13,750
24	41,711	13,950
26	48,000	14,950
30	38,500	13,750
32	61,000	12,950
27	94,612	16,900
30	75,889	18,600
27	19,700	21,500

⋮

age_08_04	km	price
23	30,000	?

New observation

Plot of toyota corolla data



Predict
Price (continuous)

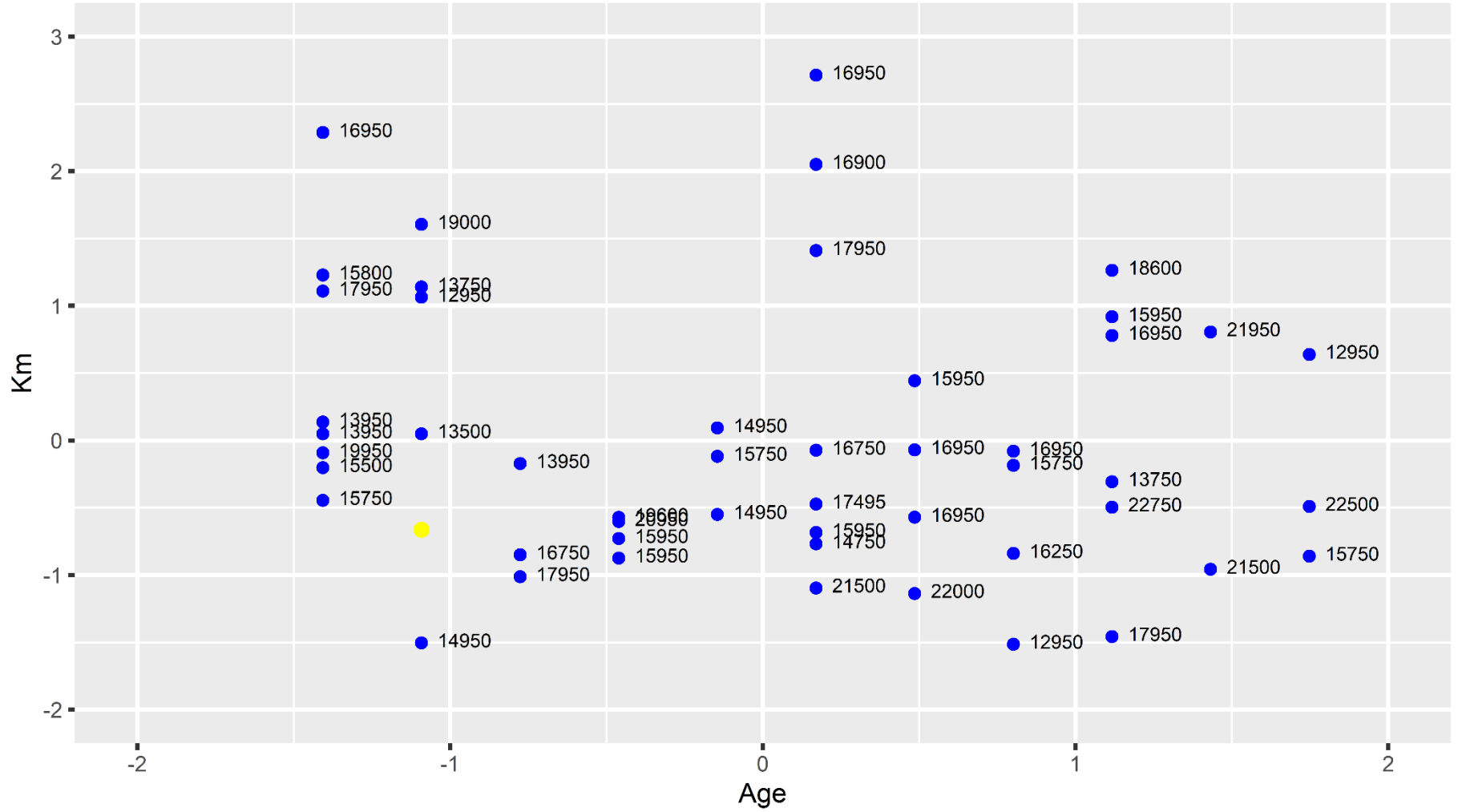
Today's class mandatory steps

- Create a folder name "**f.knn_regression**" within the folder "**oba_455_555_ddpm_r/rproject**"
- Download "**knn_regression_code.R**", and all **csv** files from canvas
- Place all downloaded files in "**oba_455_555_ddpm_r/rproject /f.knn_regression**"
- Open RStudio project
- Open "**knn_regression_code.R**" file within RStudio

k-NN as regression model in R

- Step 1: Main data
  - Standardize the numeric input variables
  - Convert input character variables into dummy (binary) variables
- Step 2: Pick only standardized input numeric & dummy variables in main data
  - **Standardized main data**
- Step 3: New data – prediction of interest
  - Standardize the numeric input variables
  - Convert input character variables into dummy variables
- Step 4: Pick only standardized input numeric & dummy variables in new data
  - **Standardized new data**
- Step 5: Track the output variable in the main data
  - **Main data output**
- Step 6: Execute the function "**knn.reg**" to predict for new observation

Plot of toyota data post normalization



- If $k = 1$, prediction is 16,750
- If $k = 2$, prediction in 16,250 $(= \frac{16750+15750}{2})$
- If $k = 3$, prediction is 16,816.67 $(= \frac{16750+15750+17950}{3})$

Choosing *k*

- Too Low (E.g., $k = 1$)
  - We may be fitting noise in the data
  - Ignoring a lot of information
  - Overfitting
- Too High (E.g., k = number of observations in the data)
  - Loss of ability to capture local structure of the data
  - Underfitting
- Balance between overfitting and underfitting
- How to achieve balance?
- How to choose *k*?
  - Best Classification/Regression (Prediction) performance
  - We will discuss this is more scientifically 2 classes from now

(Dis)Advantages of *k*-NN

- Simplicity and lack of parametric assumptions
- Time taken to find nearest neighbors in large datasets can be unaffordable
  - Reduce time taken to compute distance by using **dimension reduction** techniques
  - Sophisticated data structures such as **search trees** to speed up identifying the nearest neighbor
- Number of observations required increases exponentially with the number of variables/predictors in the data
  - E.g., in *k*-NN as a classifier for ridge mowers data, we have two variables – Income, Lot Size
- Lazy learner
  - For every prediction, the algorithm computes distances for all the data points

Next class

- Midterm1
  - Next Tuesday (19th April 2022); Multiple choice quiz on canvas
  - Topics discussed until today
  - Open book
  - Conceptual knowledge
  - Identifying the appropriateness of different techniques for different business problems/scenarios
  - Identifying strengths and shortcomings of the techniques
  - Interpret results of analyses
  - Code errors, output

Thank You