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

Previous Class

- Advanced Data Management & Graphics in R/RStudio
- Advanced Operations
  - Tidying
  - Binding
  - Appending
  - Merging
  - Long ↔ Wide
  - ………

Announcement

- Homework 1 due by midnight
- Midterm1
  - Next Tuesday (19th April 2022); Multiple choice quiz on canvas
  - Topics discussed until the end of the next class
  - 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

Today's class

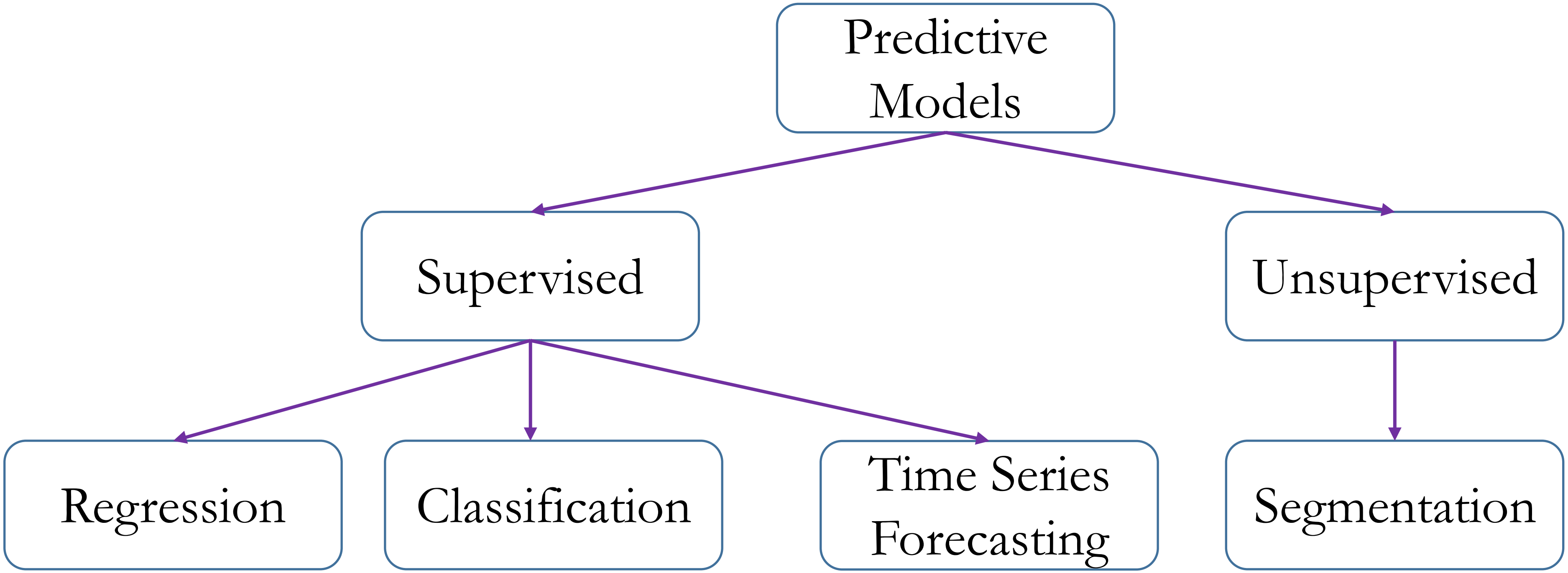
- Advanced operations – Handling missing values
- *k*-Nearest Neighbor (*k*-NN) as Classification
- Application of *k*-NN in R/RStudio and Inference

Handling Missing values

- Missing numeric/character data in R is represented by **NA**
- Missing values can lead to incorrect analysis
- Pay keen attention to missing values
- Actions
  - Delete observations
  - Replace with a value
- No correct action
- Depends on data, context, the extent to which it is a problem
- Make conscious action and support why you are doing it

Mandatory steps

- Open RStudio project
- Open “**data_mgmt2_code_complete.R**” file within RStudio present in the path “**oba_455_555_ddpm_r/rproject/d.data_mgmt2**”



Predictive Models

Supervised

Unsupervised

Regression

Classification

Time Series Forecasting

Segmentation

- ***k*-Nearest Neighbor**
- **Linear Regression**
- **Regression Trees**
- Neural Networks
- Ensembles
- ……

- ***k*-Nearest Neighbor**
- Naïve Bayes
- **Logistic Regression**
- **Classification Trees**
- Neural Networks
- Discriminant Analysis
- Ensembles
- ……

- **Regression-based**
- Smoothing methods
- ……

- **Clustering**
- ……

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

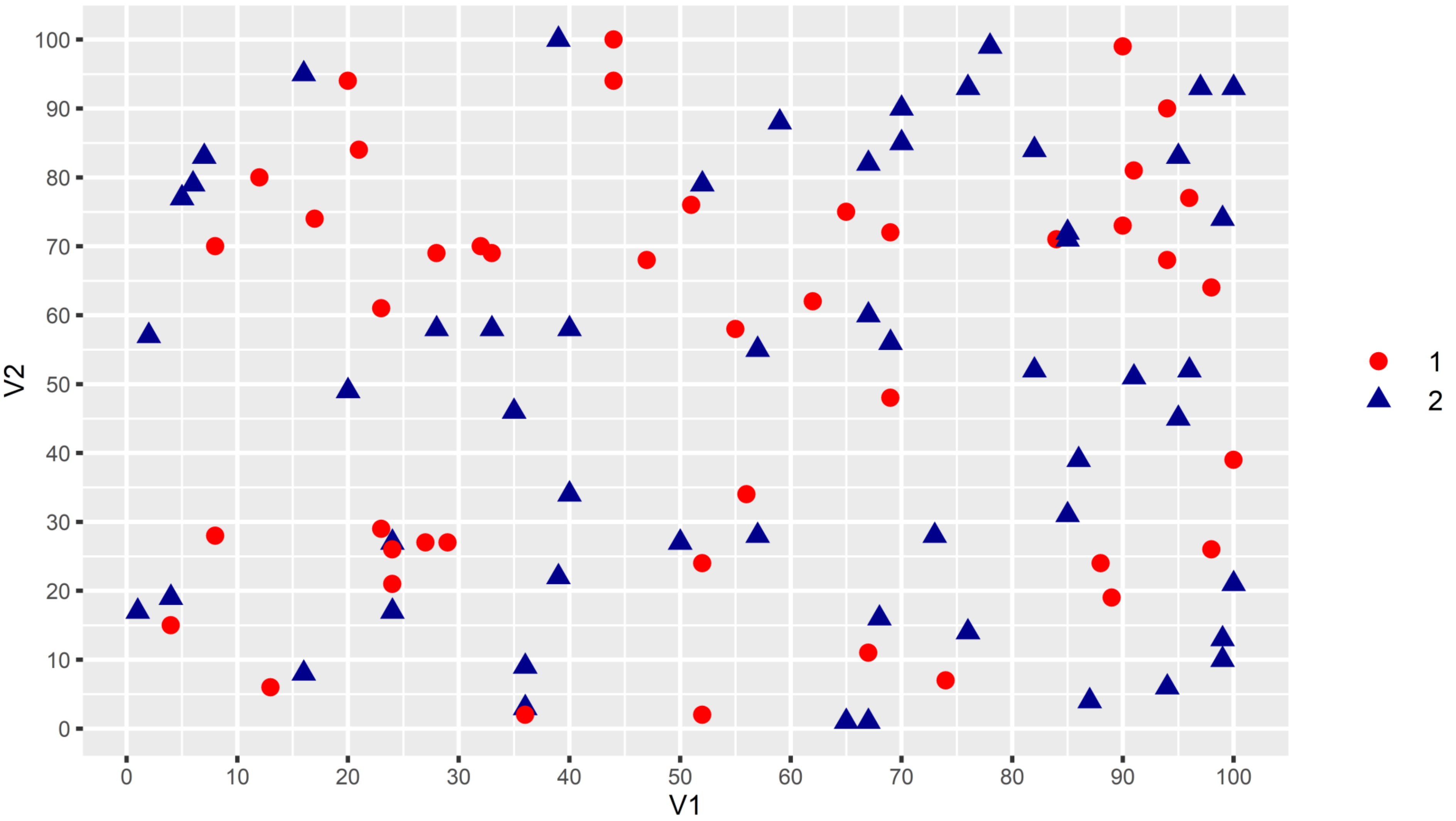
⋮

⋮

V1	**V2**	**Class**
60	60	?

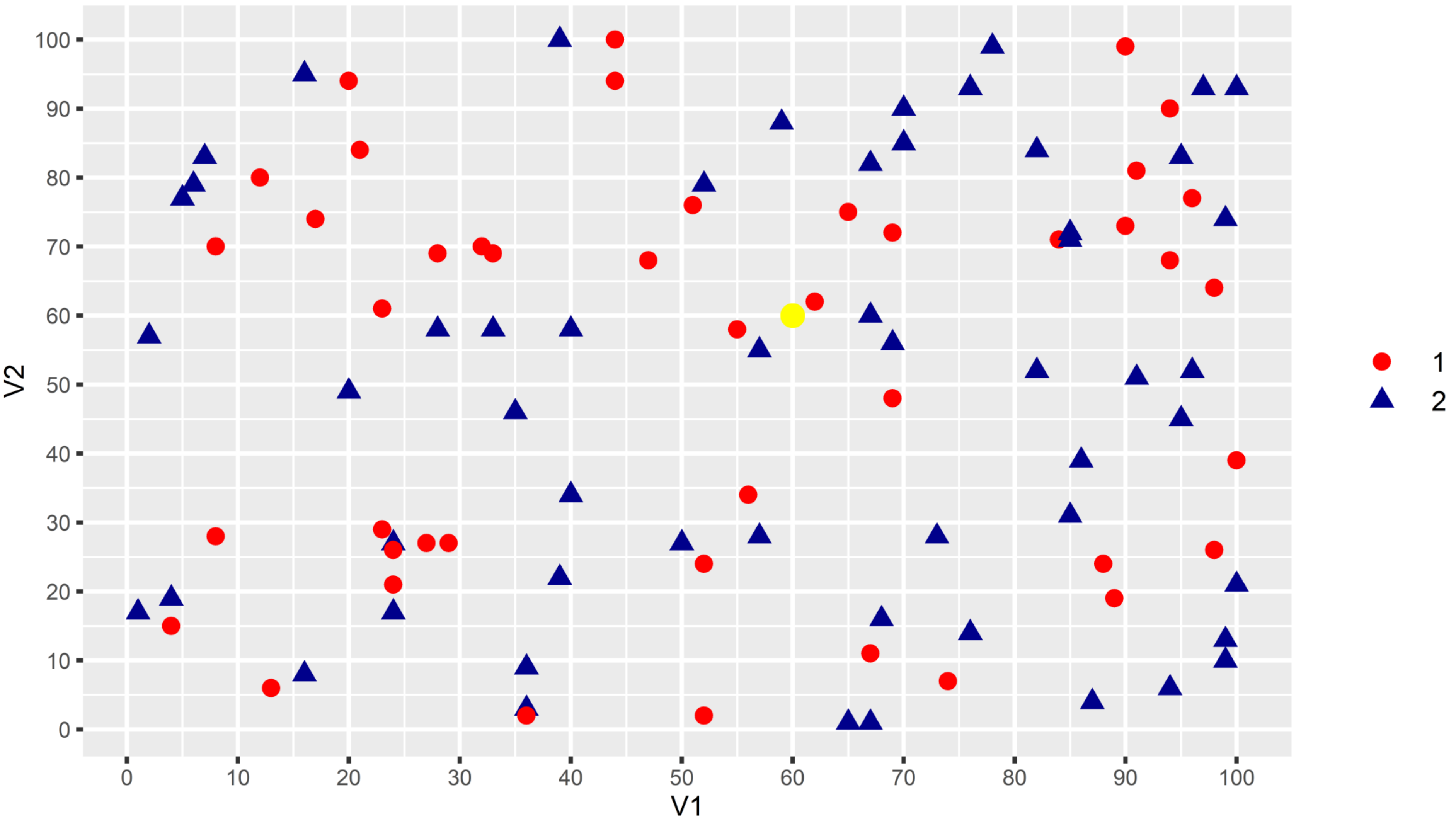
New observation

Scatter plot



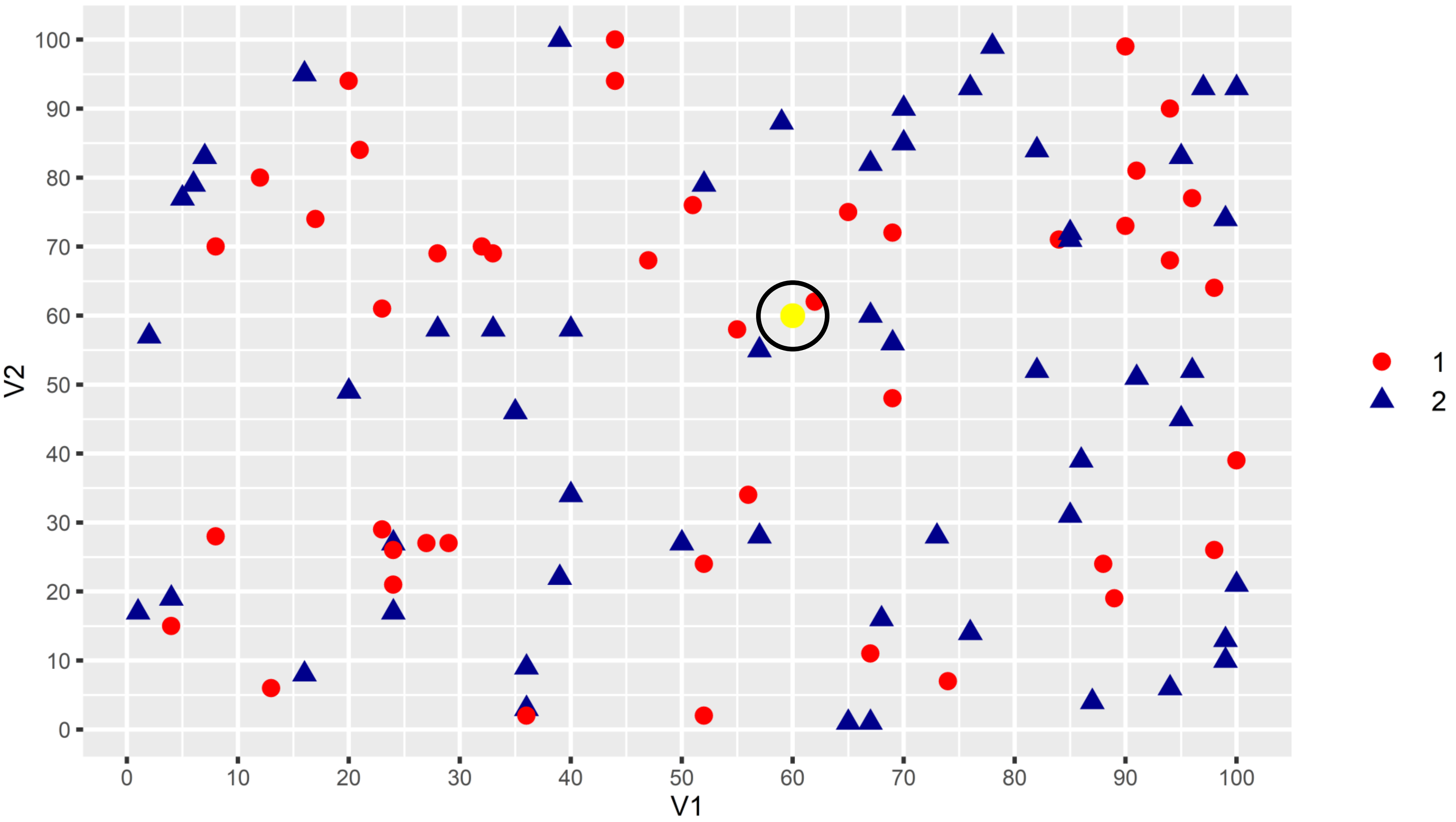
© Pradeep Pendem

New observation (yellow point)



© Pradeep Pendem

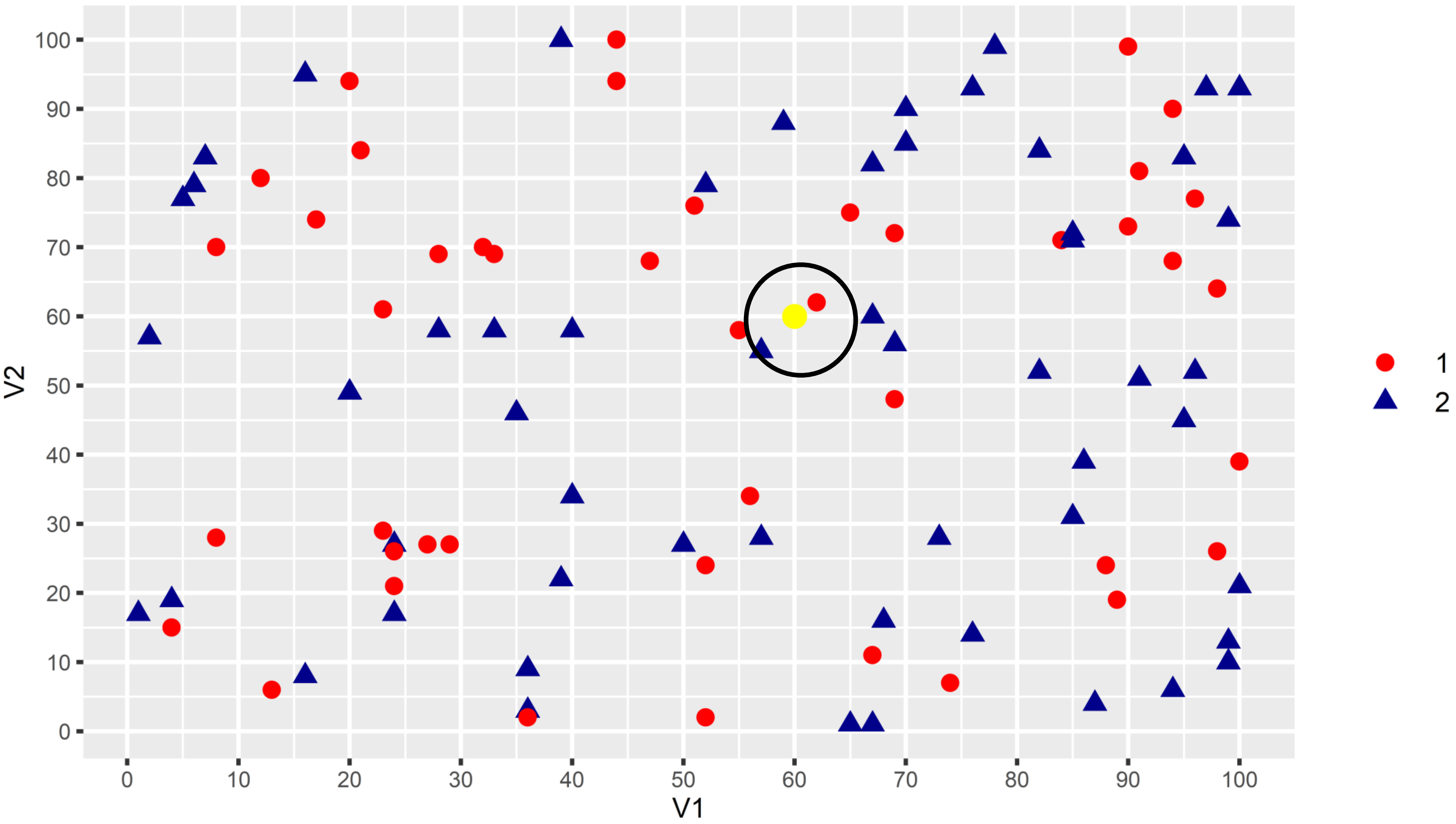
1-NN as Classifier



New observation prediction = **Class 1**

© Pradeep Pendem

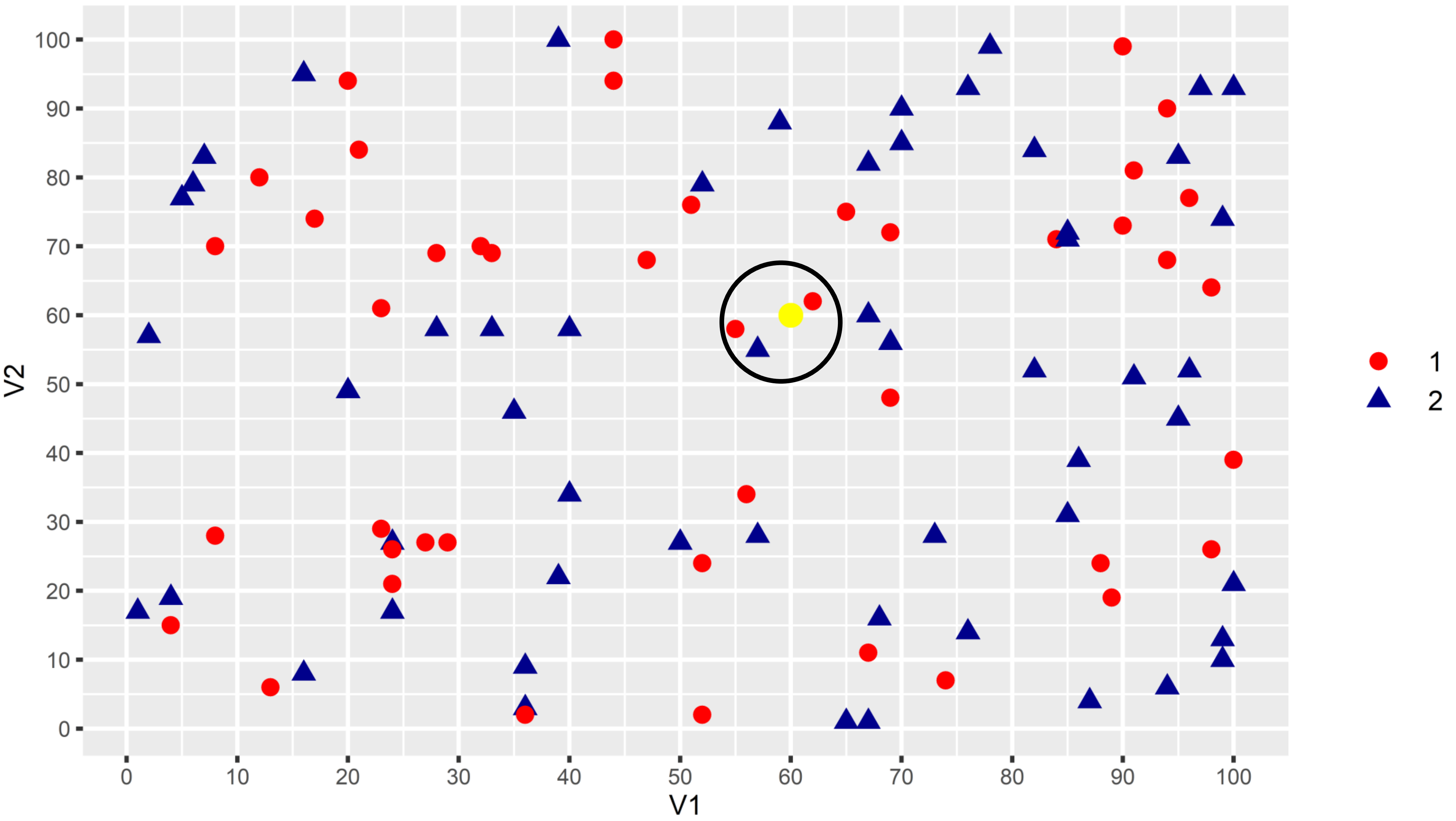
2-NN as Classifier



New observation prediction = **Tie**

© Pradeep Pendem

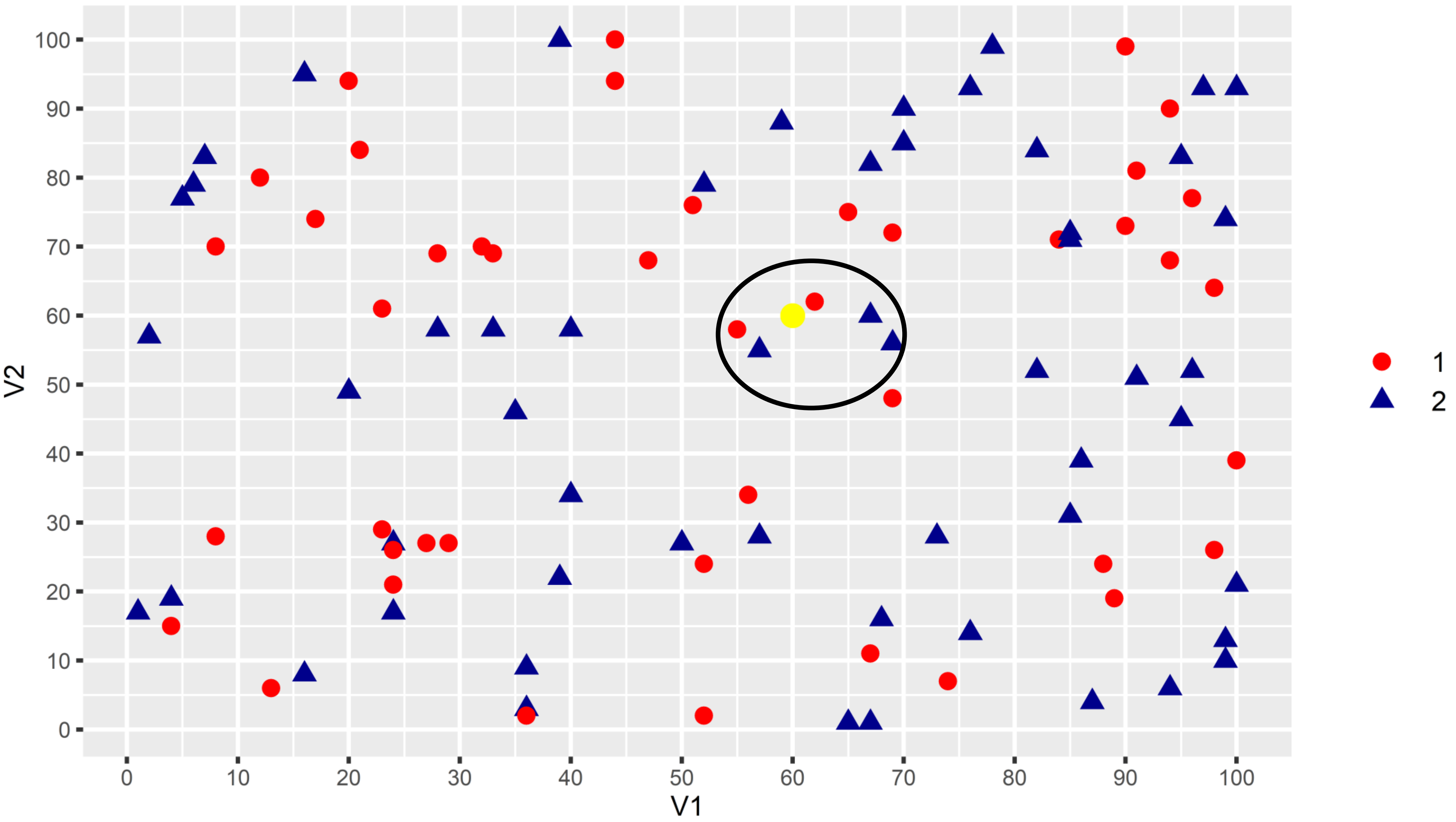
3-NN as Classifier



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

© Pradeep Pendem

5-NN as Classifier



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

© Pradeep Pendem

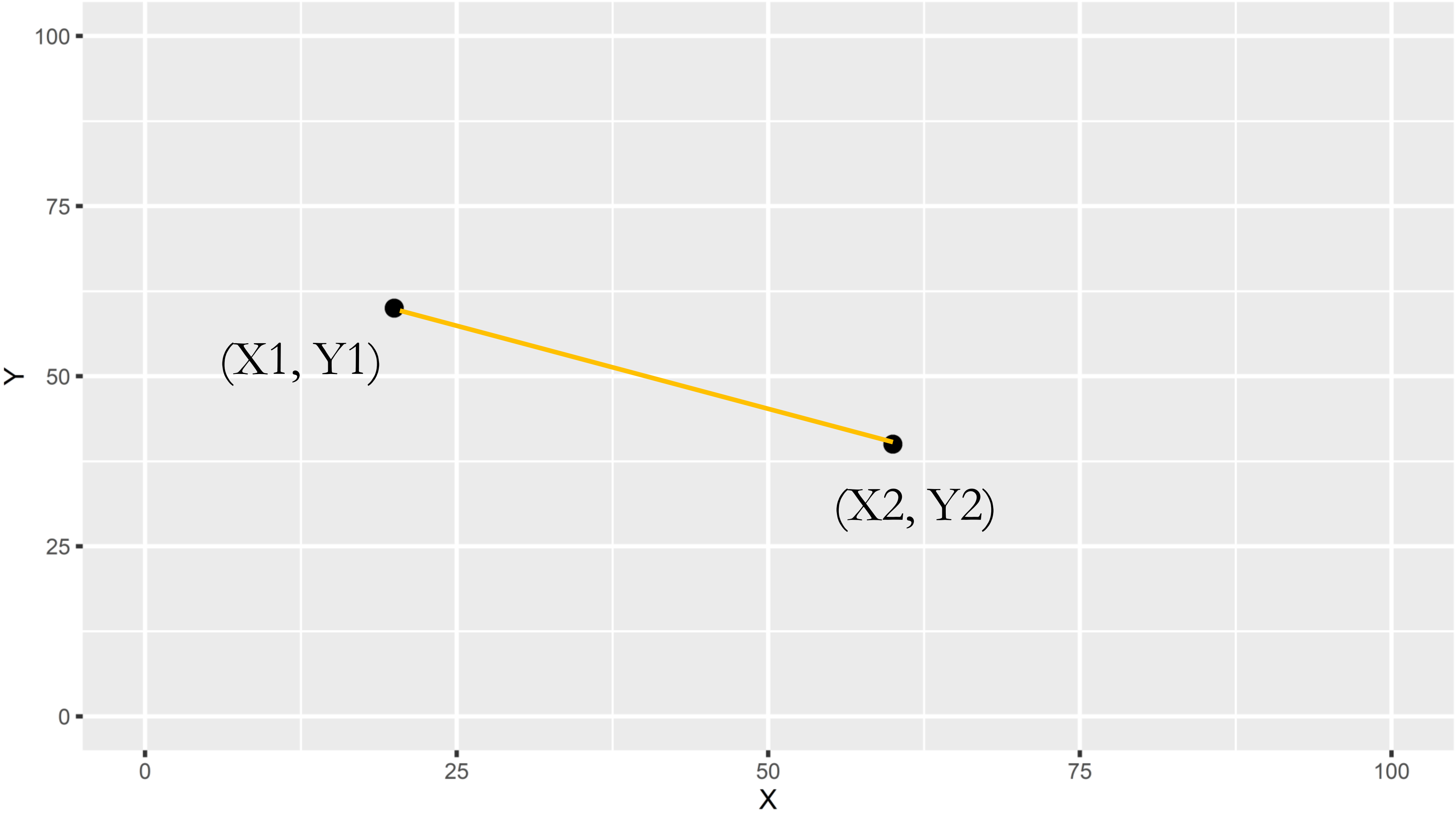
k-NN as Classifier

- Neighbors
  - **Nearest** to the new observation
  - What do you mean by **Nearest**? Distance?
  - **Euclidean distance:** Computationally cheap and most popular
- Other distance measures
  - Bregman divergence
  - Mahalonobis distance
  - Bhattacharya distance
  - Hellinger distance
  - Manhattan distance

⋮

⋮

Euclidean Distance



$$\sqrt{(X2 - X1)^2 + (Y2 - Y1)^2}$$

Euclidean Distance

- Numbers must be same unit/unit free
- In practice, numbers have units
- Example: Toyota Corolla
  - **price** variable is in **euro**
  - **km** variable is in **kilometers**
- Distance computation should be unit free
- X1, X2, Y1, Y2 all must be unit free/same unit
- Solution
  - Standardization/Normalization

Standardization/Normalization

- Transformation of data
- Subtract mean from each observation
- Divide the result by standard deviation

X
64
18
24
46
72

- **m = mean(c(64, 18, 24, 46, 72))**
- **s = sd(c(64, 18, 24, 46, 72))**
- X_norm = (X-m)/s

X_norm
0.8076
-1.1273
-0.8749
0.0505
1.1441

- Mean of normalized data is 0
- Standard deviation of normalized data is 1

Data on Riding Mowers

- Riding-mower manufacturer would like to find a way of classifying families in a city into an **owner** or **non-owner**
- Attributes
  - Income: Income of the household in thousand of dollars
  - Lot Size: Lot size in thousand of square foot
  - Ownership: Owner or Non-owner

Income	Lot_Size	Ownership
60	18.4	Owner
85.5	16.8	Owner
64.8	21.6	Owner
61.5	20.8	Owner

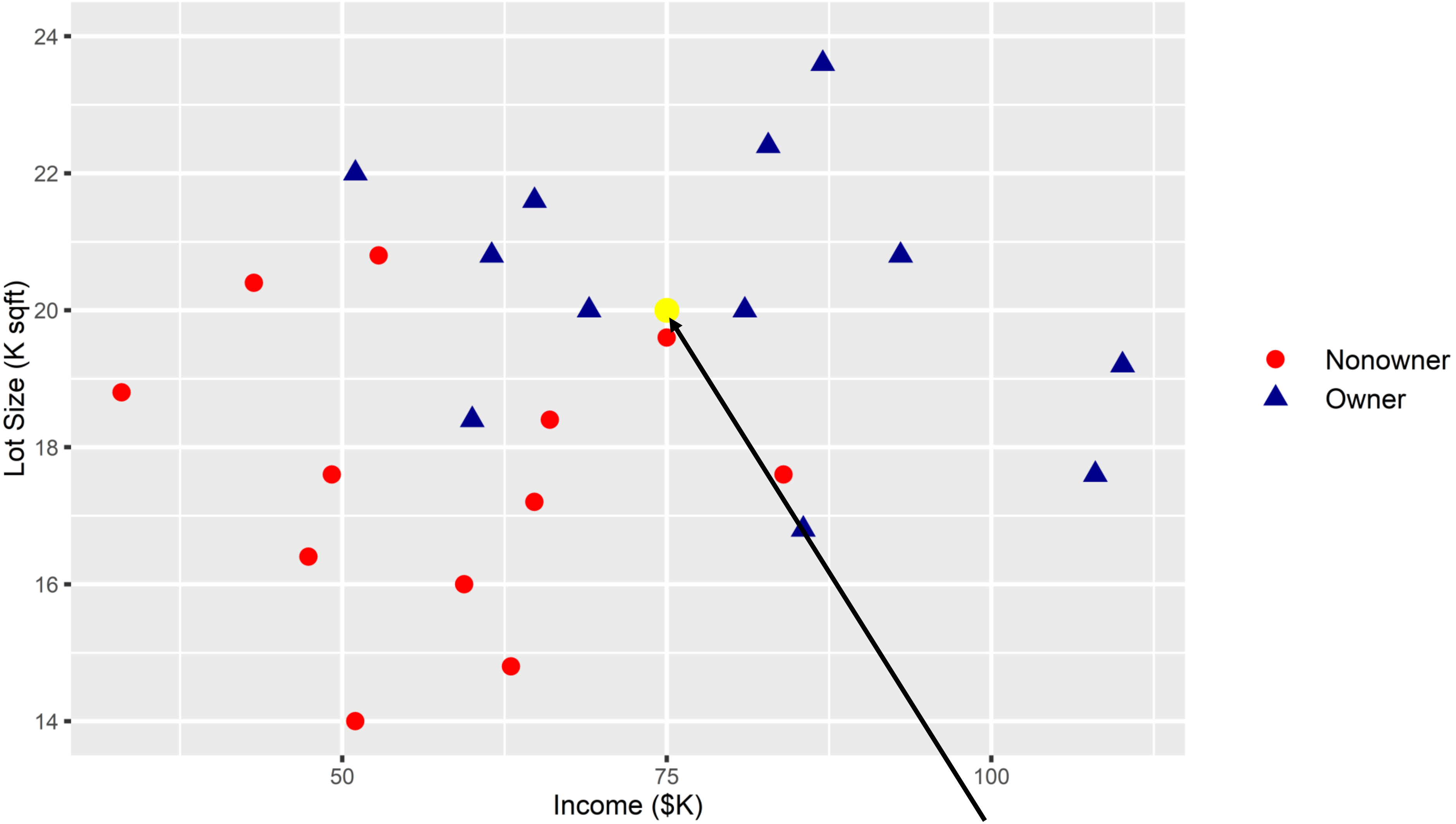
⋮

⋮

Income	Lot_Size	Ownership
75	20	?

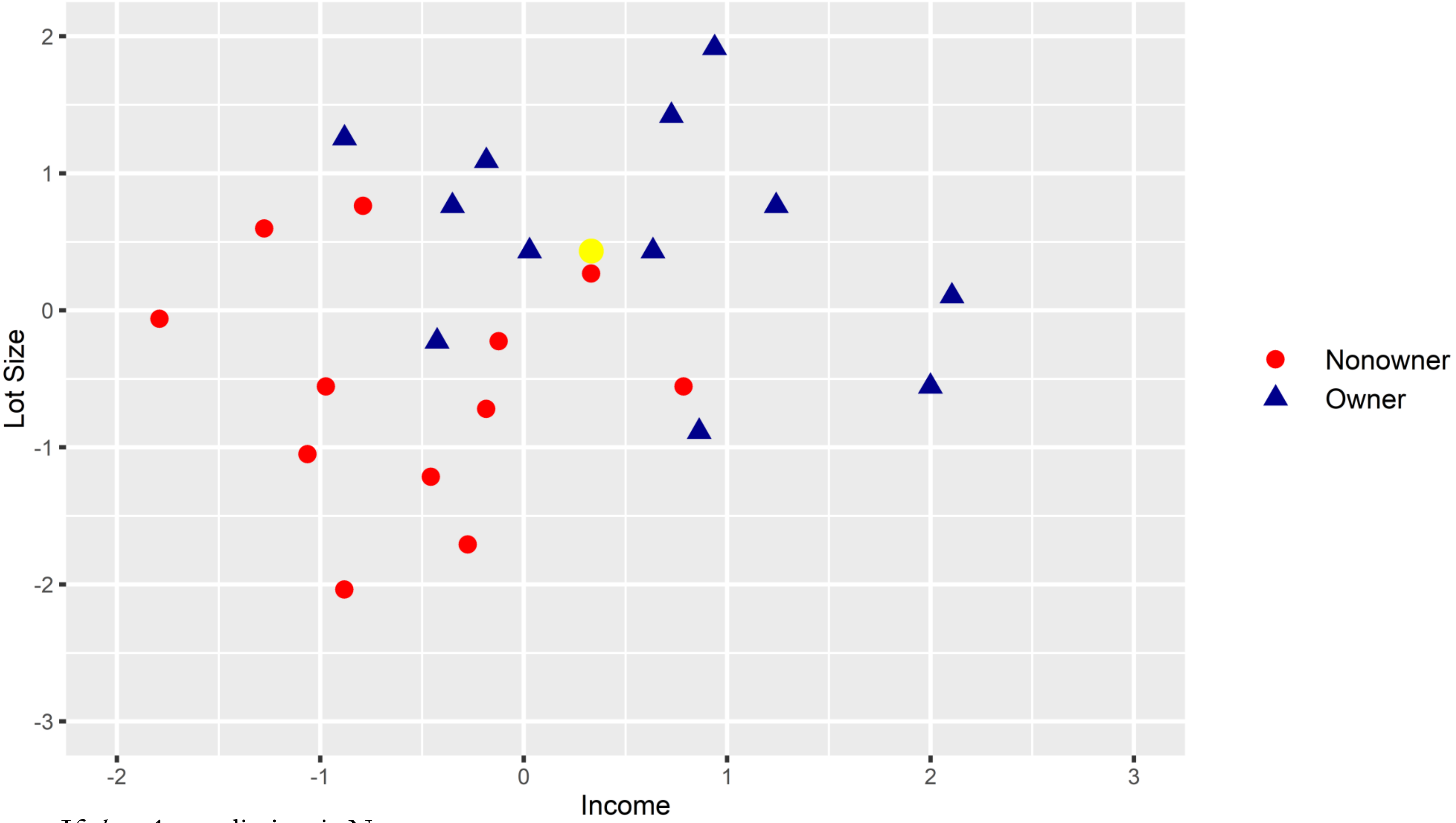
New observation

Scatter plot of entire data



Predict
Owner (blue triangle) or Nonowner (red circle)

Scatter plot of entire data post normalization



- If k = 1, prediction is Nonowner
- If k = 2, prediction in theory "tie" (algorithm gives "Nonowner" randomly)
- If k = 3, prediction is Owner
- If k = 4, prediction is Owner
- If k = 5, prediction is Owner

Today’s class mandatory steps

- Create a folder name “**e.knn_classifer**” within the folder

  “**oba_455_555_ddpm_r/rproject**”
- Download “**knn_classifier_code.R**”, and all **csv** files from canvas
- Place all downloaded files in

  “**oba_455_555_ddpm_r/rproject / e.knn_classifer**”
- Open RStudio project
- Open “**knn_classifier_code.R**” file within RStudio

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

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* = 20/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 2-3 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

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

Thank You