EDUARDO VELLOSO

Processing

SENSOR DATA

MACHINE LEARNING 101

Fisher's Iris

Predicting Flower Type

SEPAL LENGTH
SEPAL WIDTH
PETAL LENGTH
PETAL WIDTH

SETOSA,
VERSICOLOUR, OR
VIRGINICA

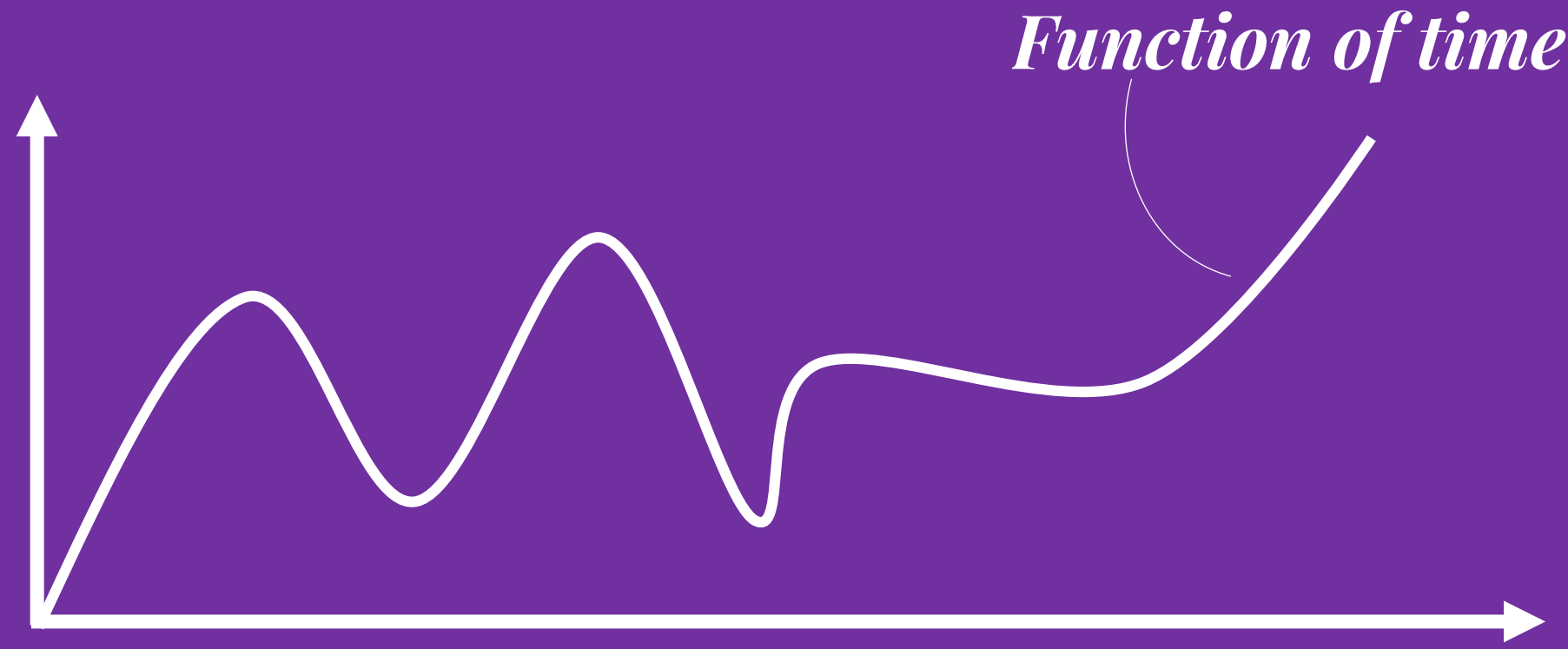
Predicting House Prices

Predicting House Prices

FLOOR SPACE
#ROOMS
LOT SIZE
CORNER?
DETACHED?

PRICE

SENSOR DATA

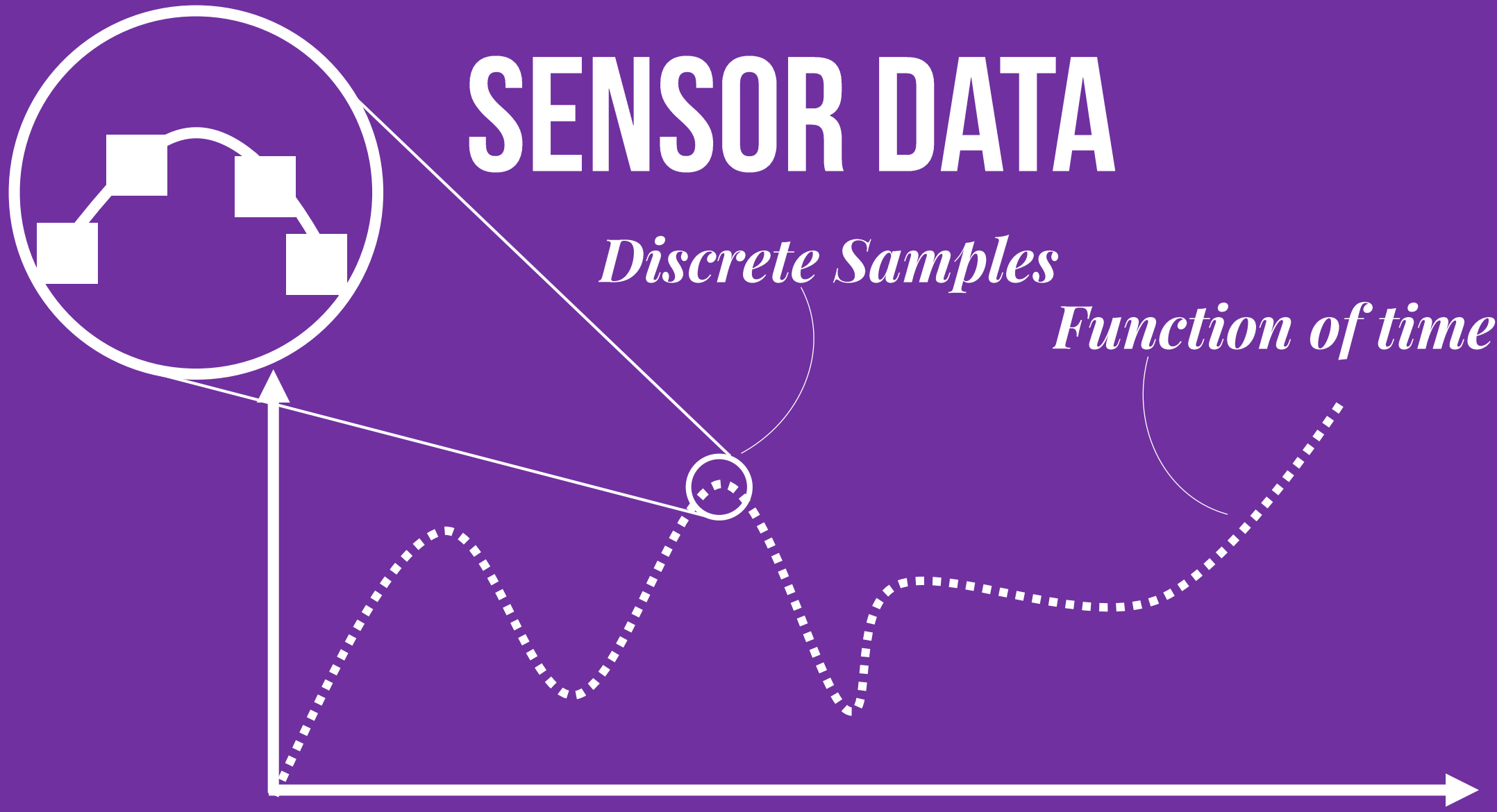


SENSOR DATA

Discrete Samples

Function of time

SENSOR DATA



SENSOR DATA

Discrete Samples

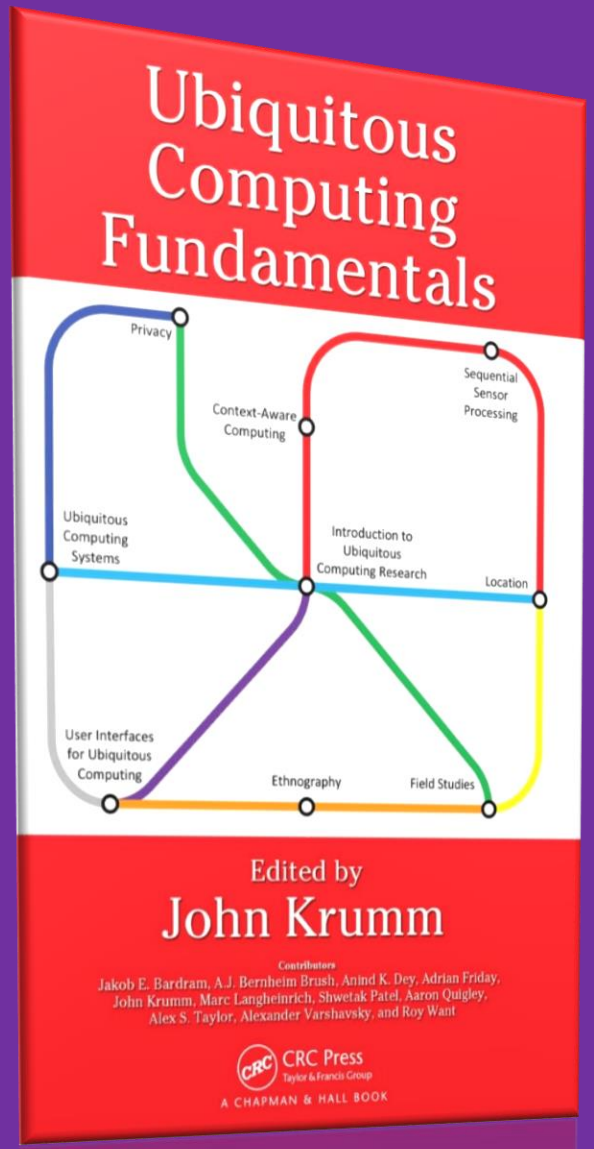
Function of time

Noisy

Mean and Median Filters
Kalman Filters
Particle Filters

JOHN KRUMM

Processing Sequential Sensor Data

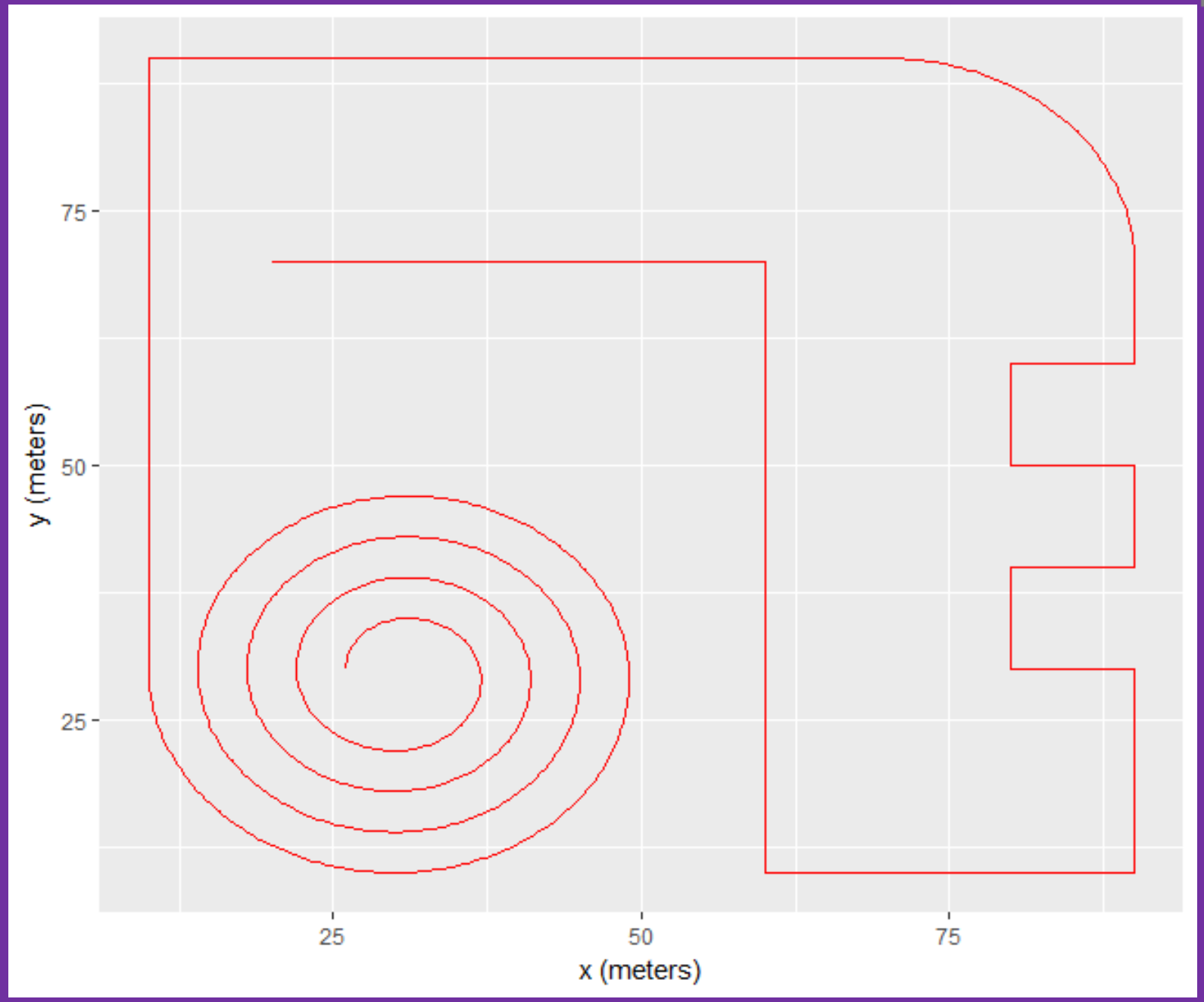

Ubiquitous Computing Fundamentals
Privacy
Context-Aware Computing
Sequential Sensor Processing
Ubiquitous Computing Systems
Introduction to Ubiquitous Computing Research
Location
User Interfaces for Ubiquitous Computing
Ethnography
Field Studies
Edited by
John Krumm
Contributors
Jakob E. Bardram, A.J. Bernheim Brush, Anind K. Dey, Adrian Friday, John Krumm, Marc Langheinrich, Shwetak Patel, Aaron Quigley, Alex S. Taylor, Alexander Varshavsky, and Roy Want
CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

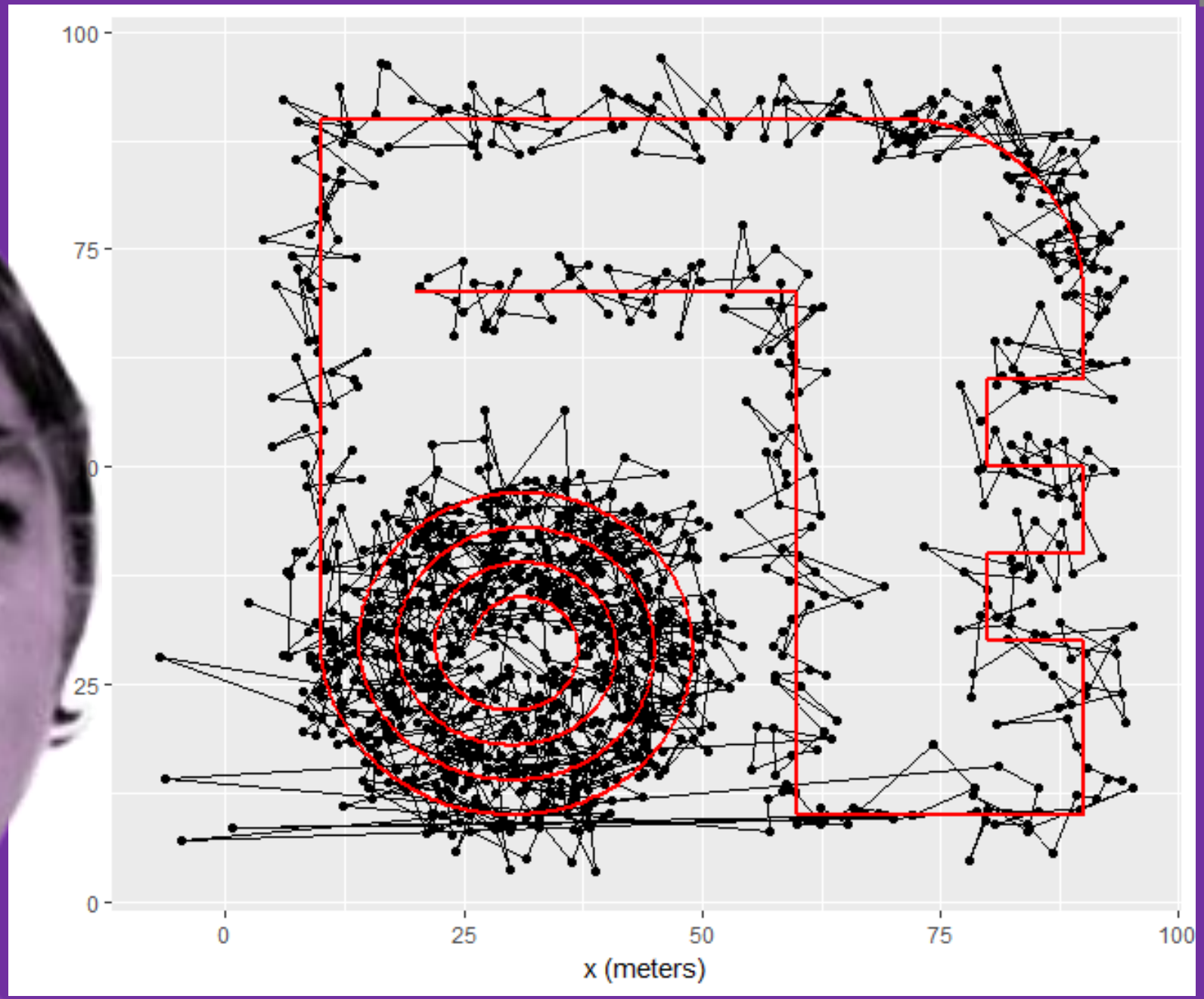
$$z_i = x_i$$

Measurement from sensor

Actual Value or Ground Truth

What we wanted




100
75
0
25
0
0
25
50
75
100
x (meters)


R
Studio

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Sensor Data Processing

sensorProcessing.Rmd | data_generation.R*

Source on Save | Run | Source

```
1  library(ggplot2)
2
3  walk <- function(path, direction, speed, samplingRate){
4    dt <- seq(path$t[nrow(path)]+1/samplingRate,path$t[nrow(path)]+1/samplingRate+speed,length.out = samplingRa
5    if(identical('up',direction)){
6      print('up')
7      dy <- seq(path$y[nrow(path)],path$y[nrow(path)]+speed,length.out = samplingRate)
8      dx <- rep(path$x[nrow(path)],length.out = samplingRate)
9    }else if(identical('down',direction)){
10     print('down')
11     dy <- seq(path$y[nrow(path)],path$y[nrow(path)]-speed,length.out = samplingRate)
12     dx <- rep(path$x[nrow(path)],length.out = samplingRate)
13   }else if(identical('left',direction)){
14     print('left')
15     dx <- seq(path$x[nrow(path)],path$x[nrow(path)]-speed,length.out = samplingRate)
16     dy <- rep(path$y[nrow(path)],length.out = samplingRate)
17   }else if(identical('right',direction)){
18     print('right')
19     dx <- seq(path$x[nrow(path)],path$x[nrow(path)]+speed,length.out = samplingRate)
20     dy <- rep(path$y[nrow(path)],length.out = samplingRate)
21   }
22   d
```

120:1 (Top Level) | R Script

Console | R Markdown

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/.RData]

>
```

Environment | History

Import Dataset | List

Global Environment

Environment is empty

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

Projects > 2017.2 - Mobile Computing > Teaching > 2017 > Sensor Data Processing > For students

Name	Size	Modified
..		
data_generation.R	4.9 KB	Aug 9, 2017, 10:04 AM
path.csv	88 KB	Jul 8, 2017, 1:13 PM
sensorProcessing.Rmd	9.6 KB	Aug 9, 2017, 10:08 AM
sensorProcessing.nb.html	784 KB	Aug 9, 2017, 10:08 AM
sensorProcessing.html	1.2 MB	Aug 9, 2017, 10:08 AM

WE ASSUME...

$$z_i = x_i$$

Measurement from sensor

Actual Value

WE ASSUME...

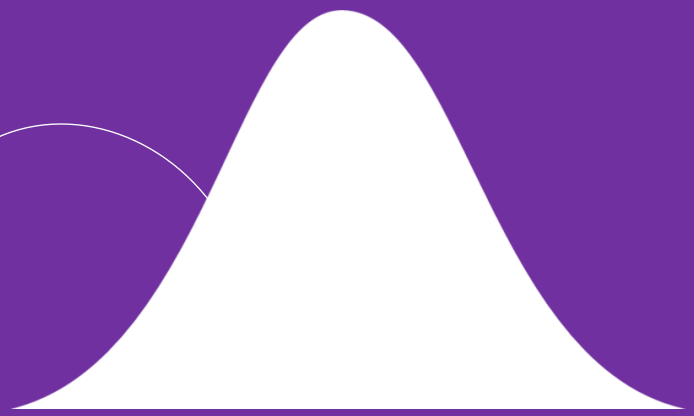
$$z_i = x_i + v_i$$

Measurement from noisy sensor

Actual Value

Sensor noise

$$v_i \sim N(0, \sigma)$$



Select font size T T T

If our sensor is a thermometer, how many dimensions does z_i have?

Allow Single Choice Only | Allow Multiple Choices | Shuffle Answers | Allow Retry | Limit Attempts

1

That's right. Because a thermometer only measures temperature, we only need one dimension to store this value

2

3

4

+ Add another answer

Preview

Select font size T T T

In the case of tracking GPS coordinates, how many dimensions we need for z_i?

Allow Single Choice Only | Allow Multiple Choices | Shuffle Answers | Allow Retry | Limit Attempts

1

2

To track GPS coordinates we to store two values: latiture and longitude, so z_i is a 2D vector in this case

3

4

+ Add another answer

Preview

$$z_i = x_i + v_i$$

noise

$$v_i = \begin{pmatrix} v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \sim \begin{pmatrix} N(0, \sigma) \\ N(0, \sigma) \end{pmatrix}$$

Normal Dist

Mean=0

sd=3

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

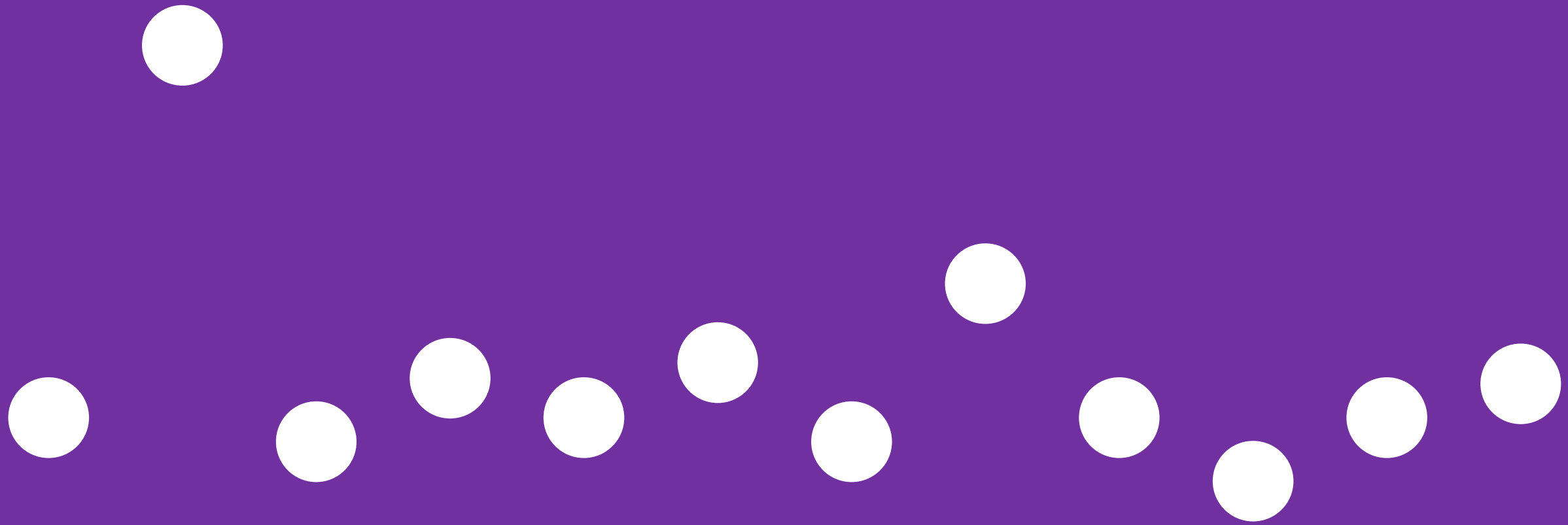
measurement

$$x_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

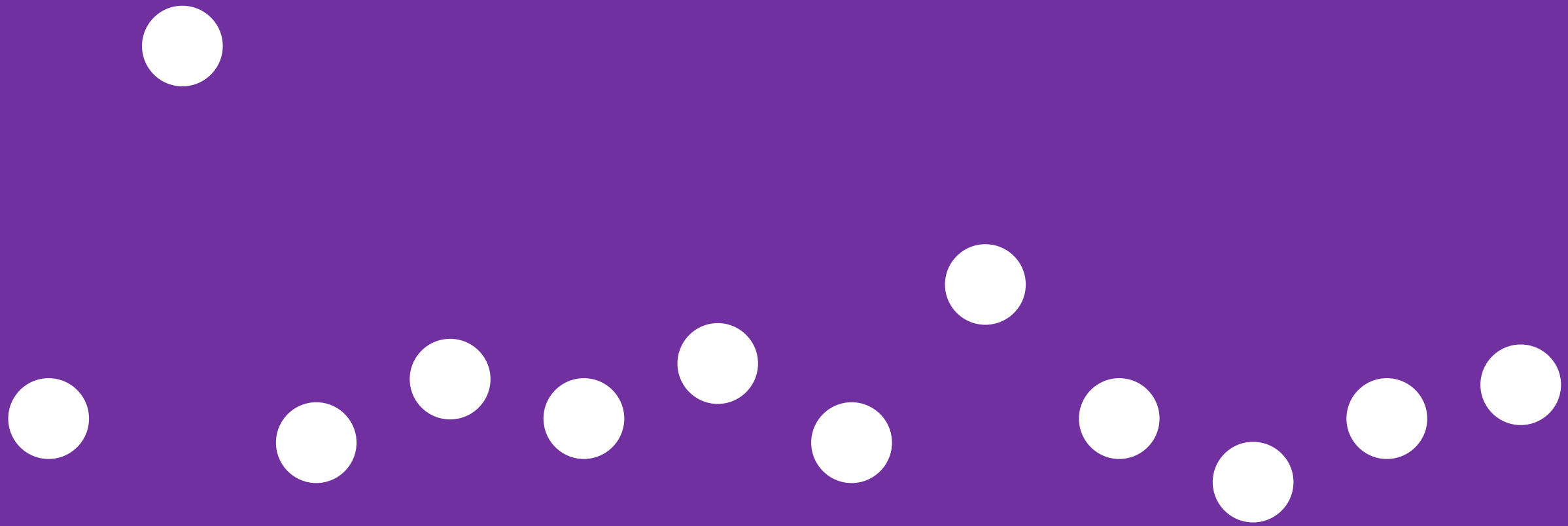
actual value

TRUTH

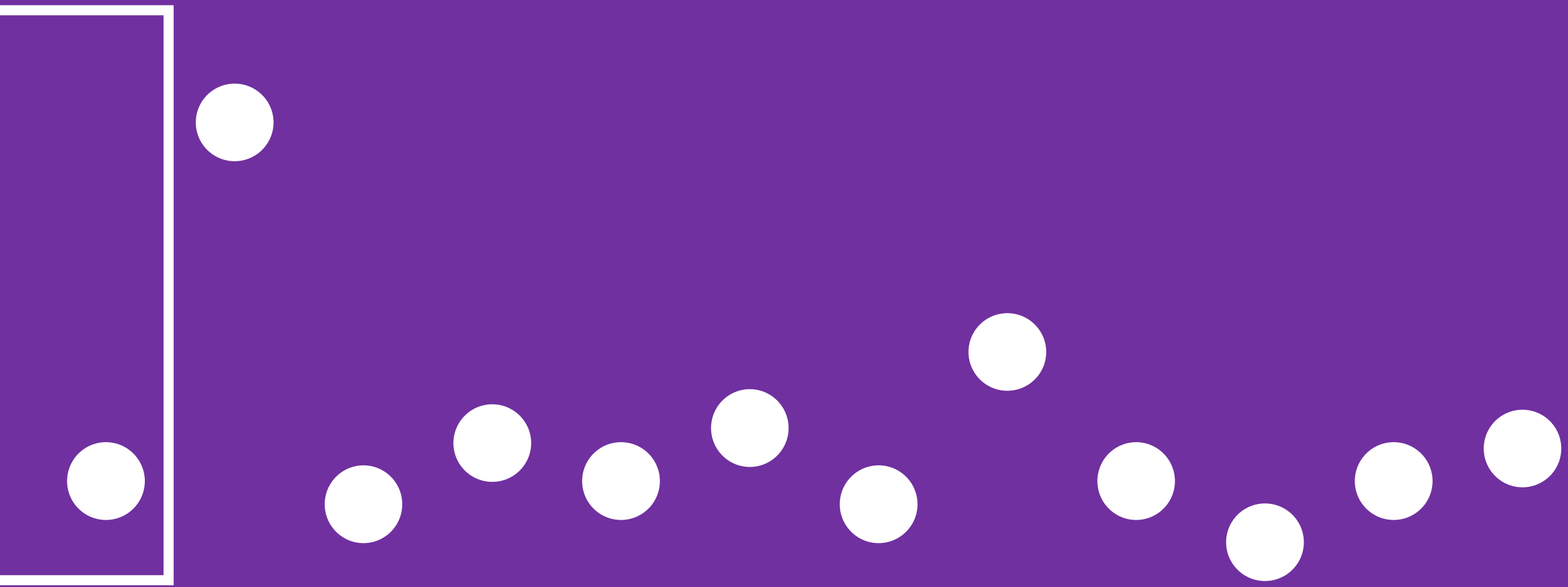
MEASUREMENTS



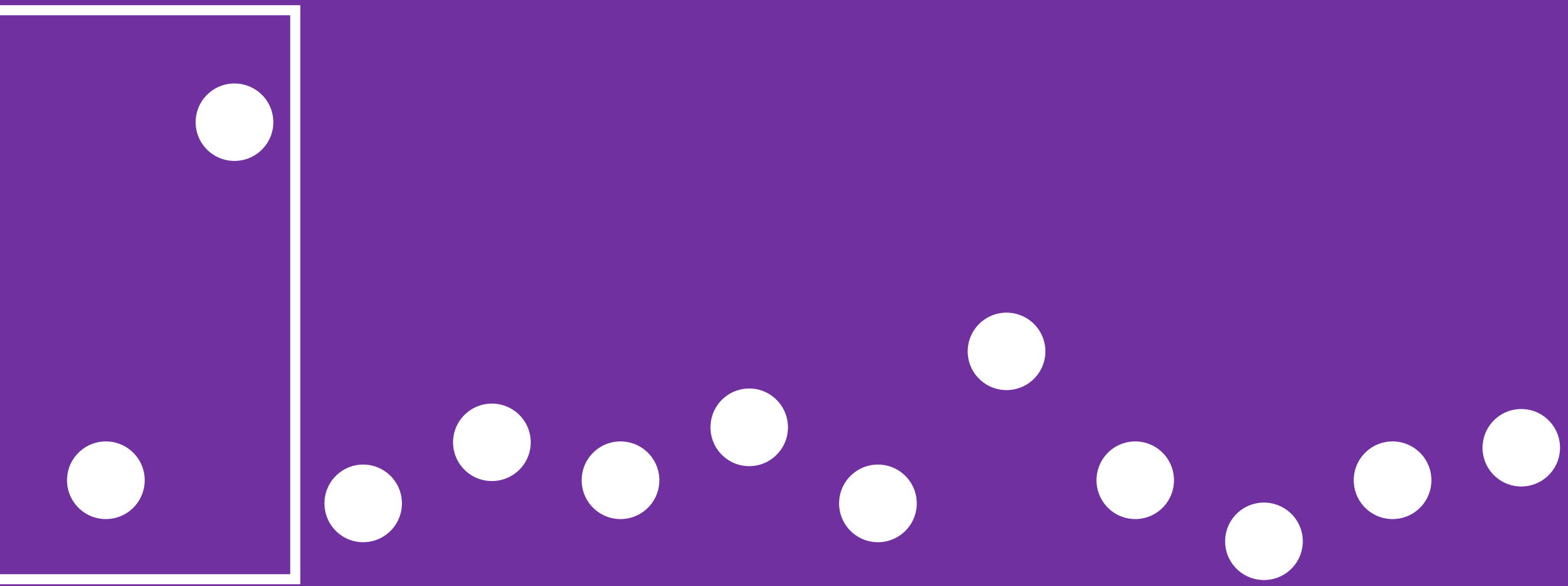
MEAN FILTER



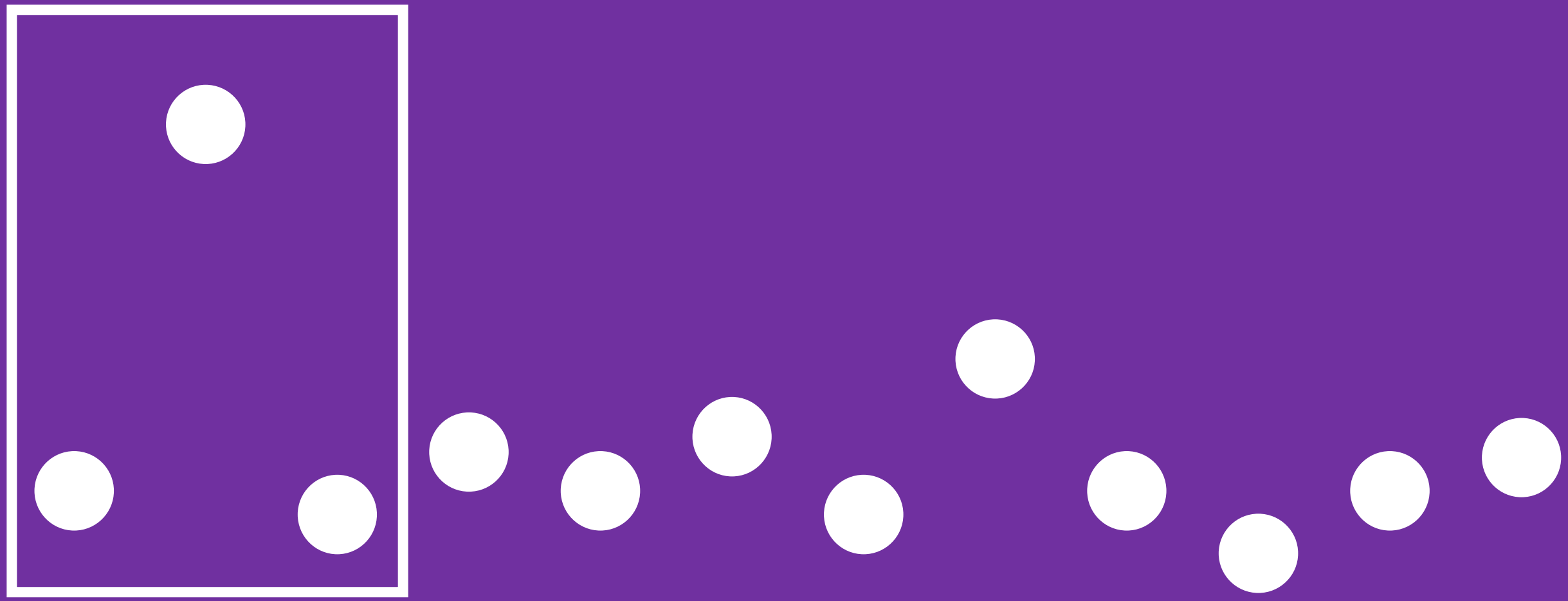
MEAN FILTER



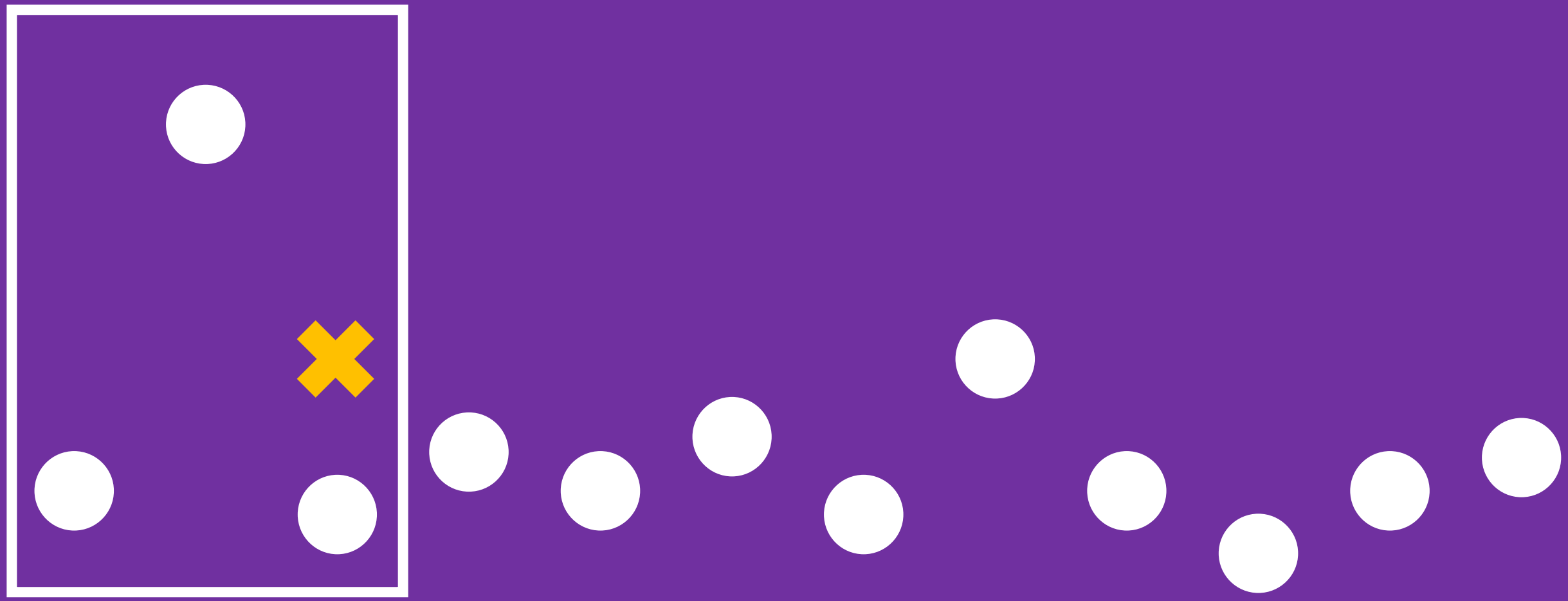
MEAN FILTER



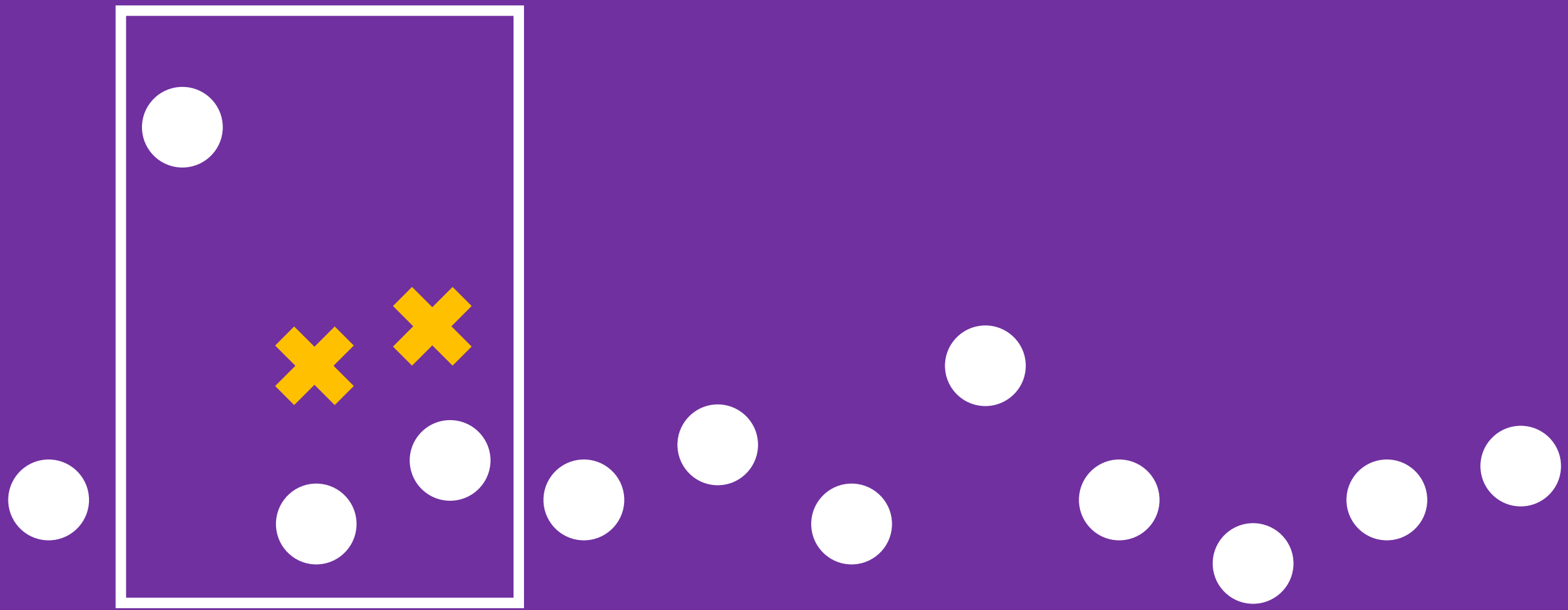
MEAN FILTER



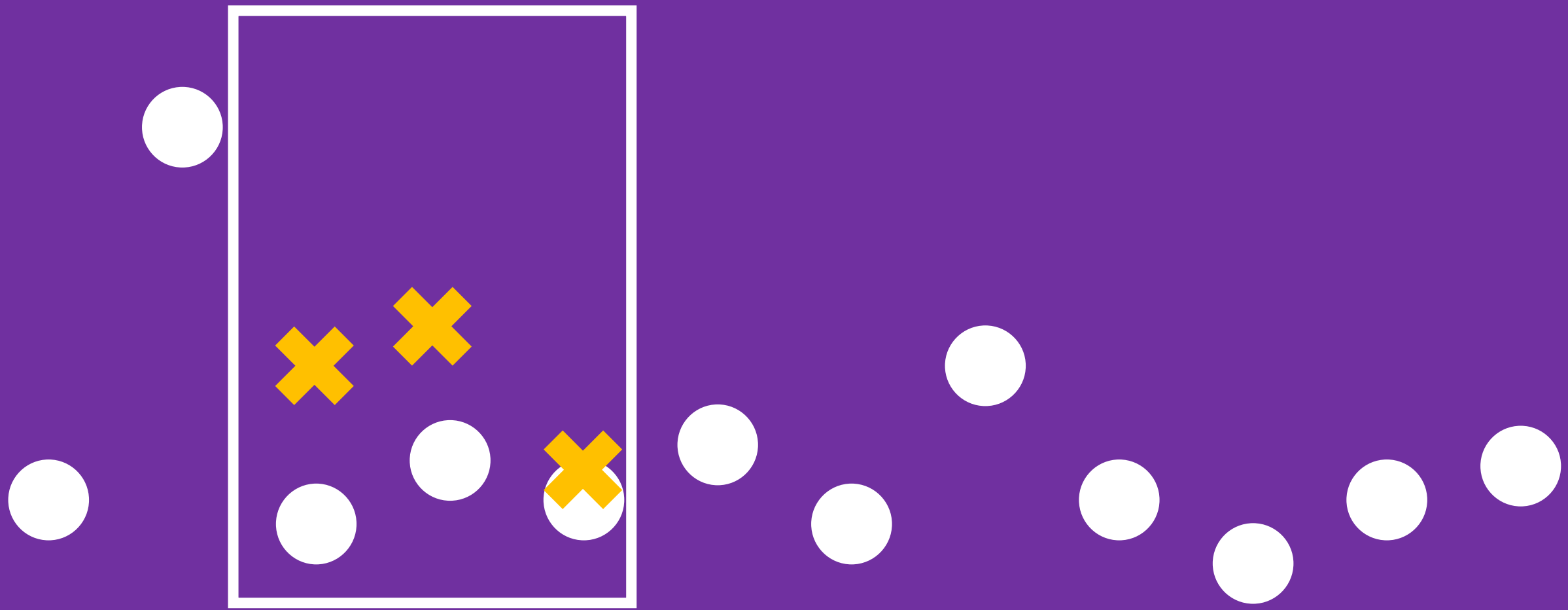
MEAN FILTER



MEAN FILTER

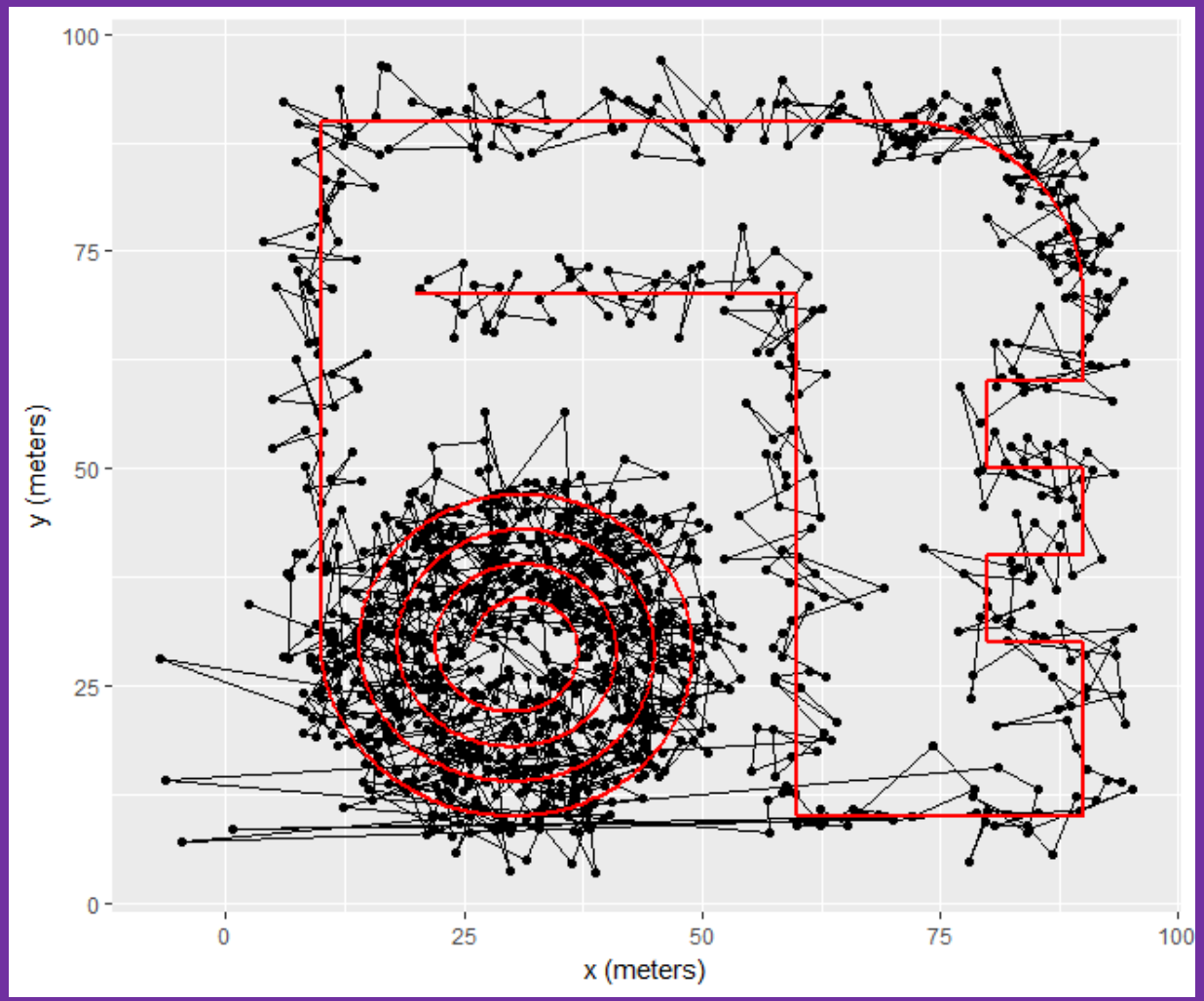


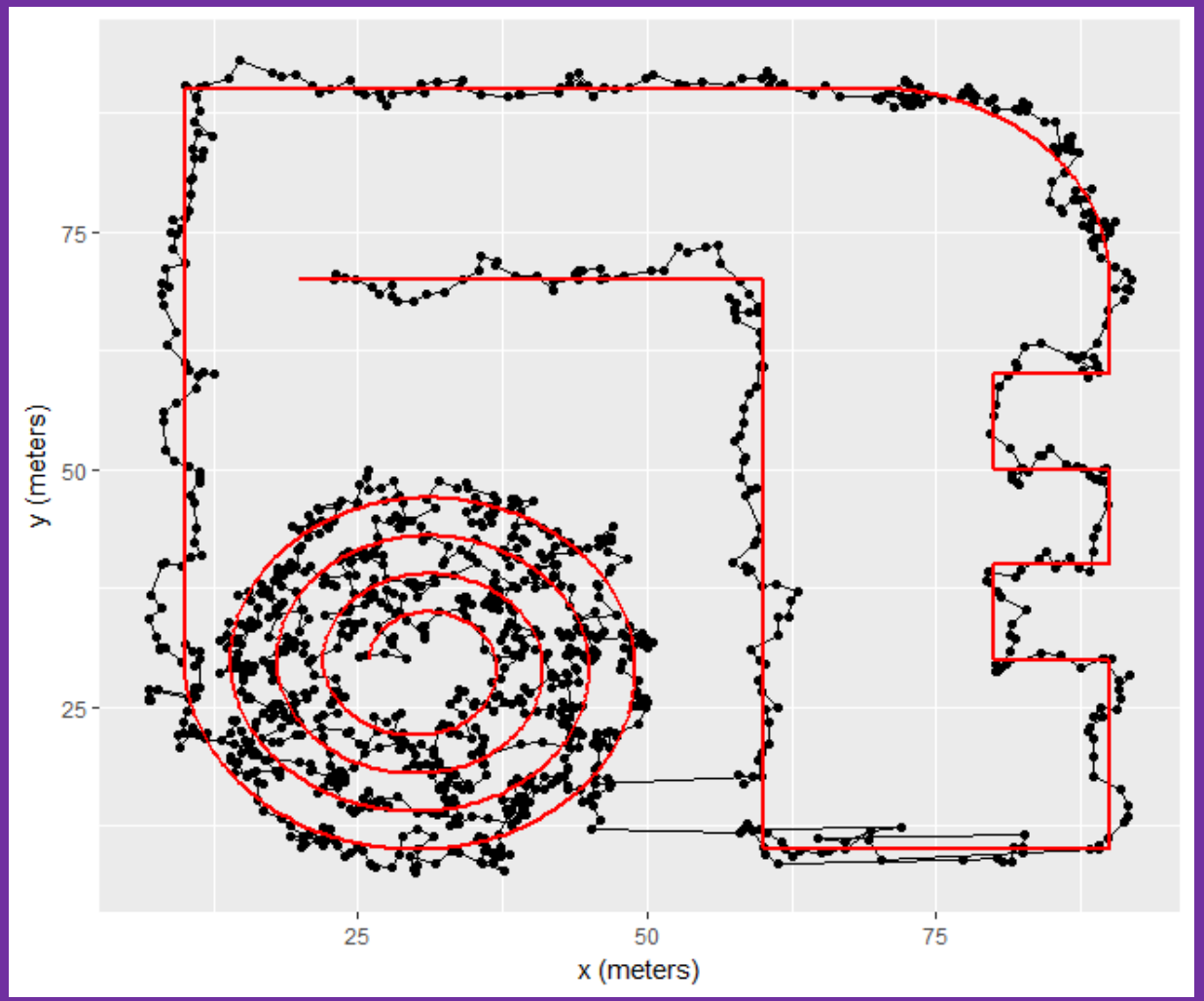
MEAN FILTER



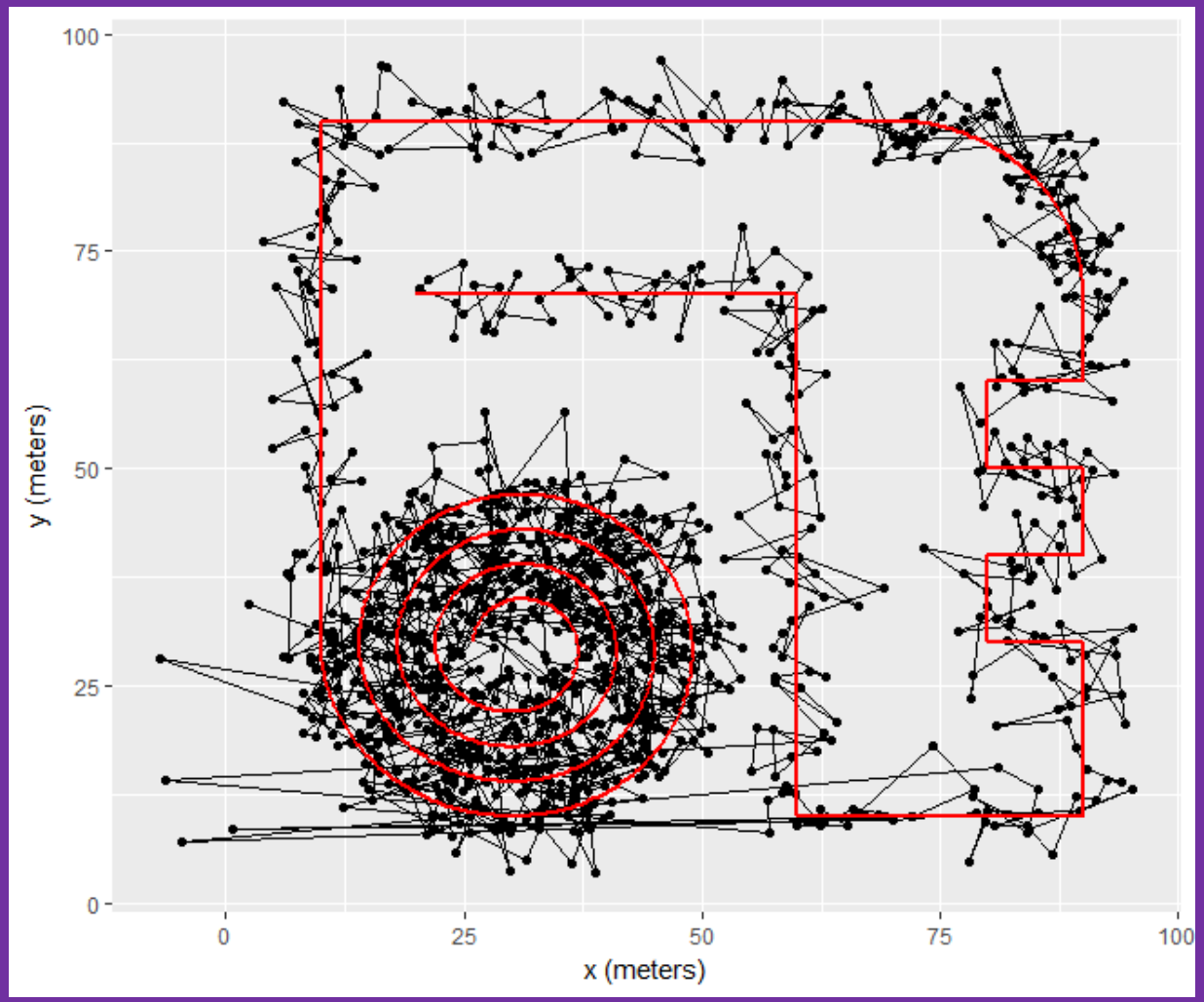
MEAN FILTER

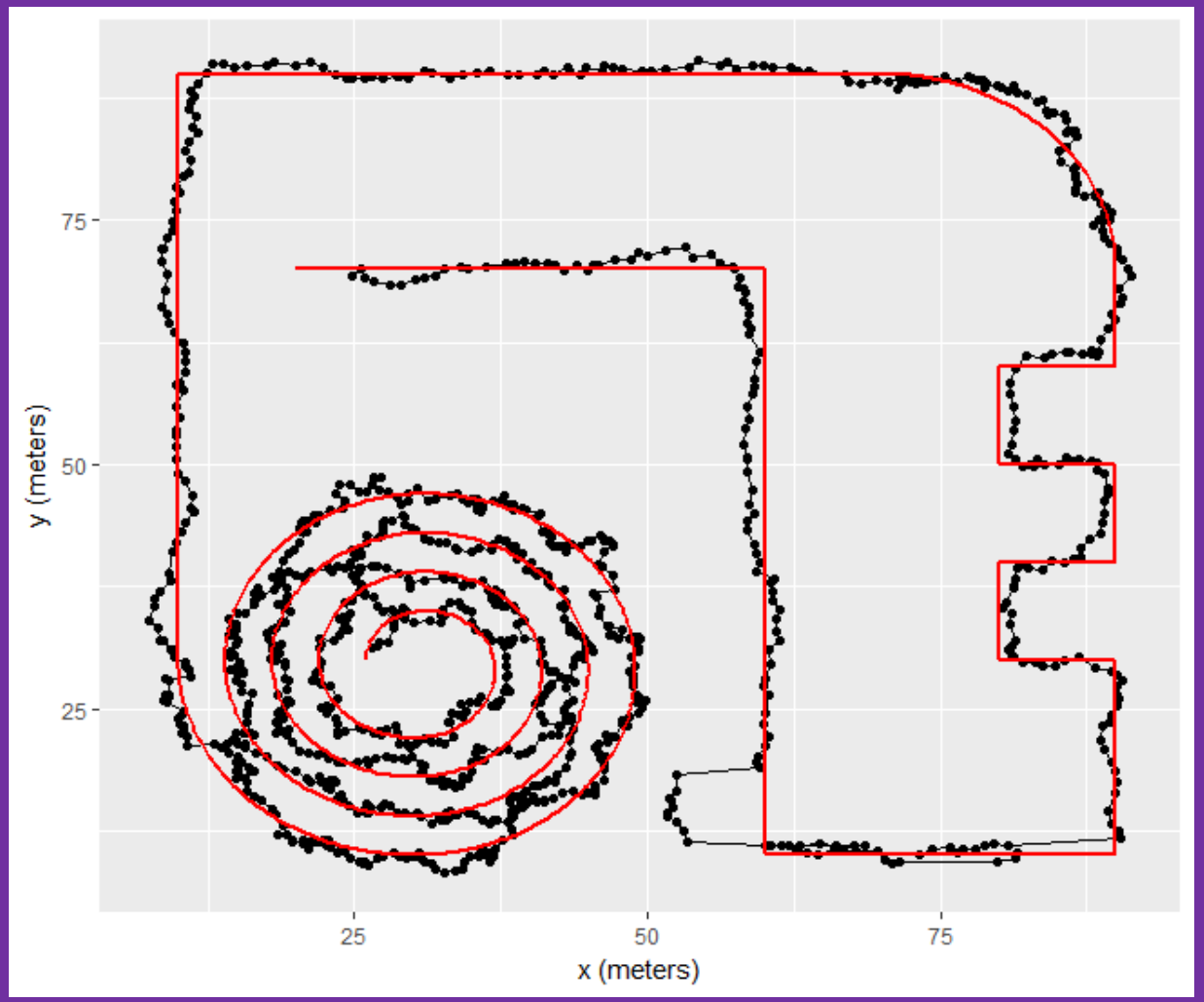
window = 5


100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)

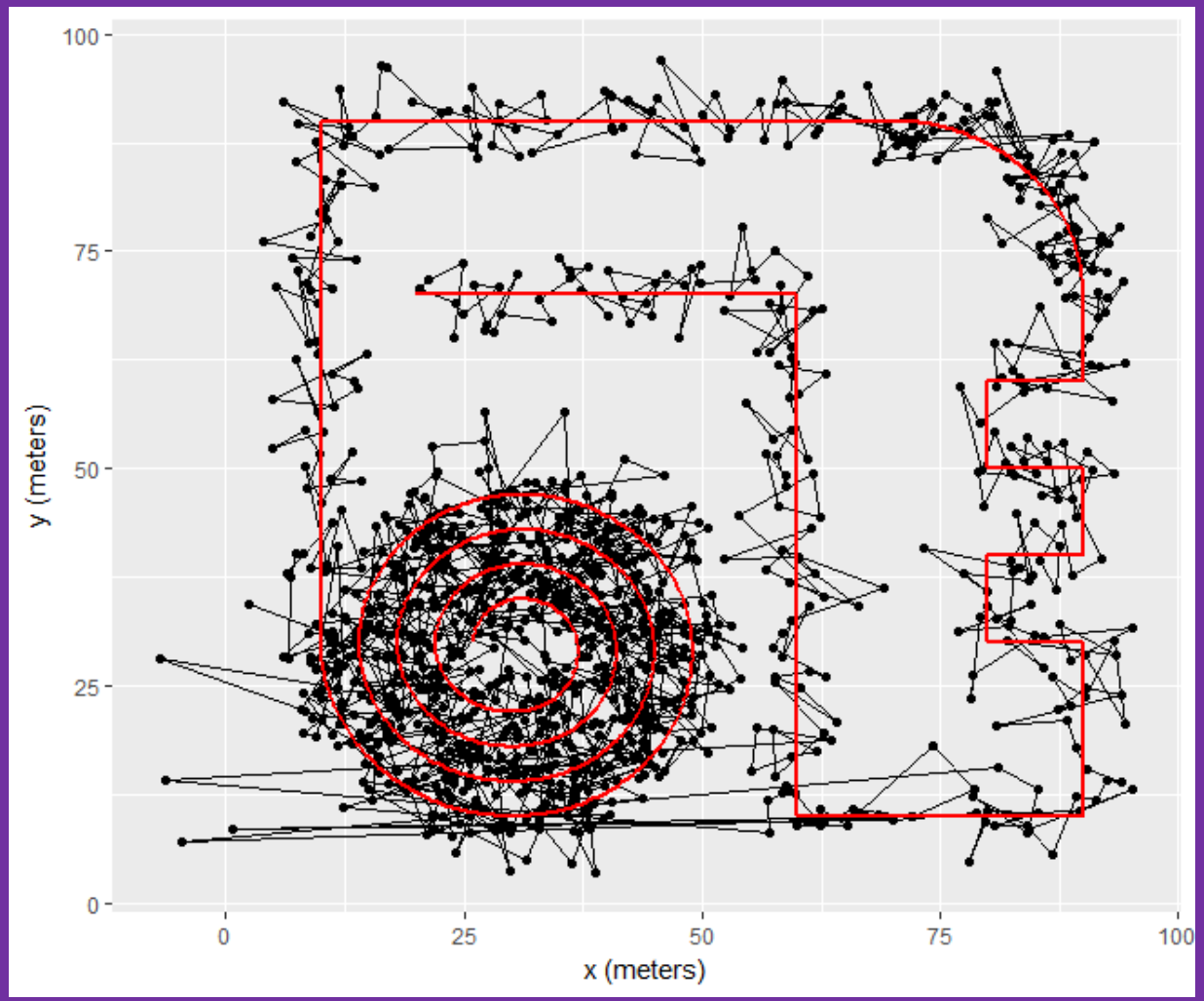

75
50
25
y (meters)
25
50
75
x (meters)

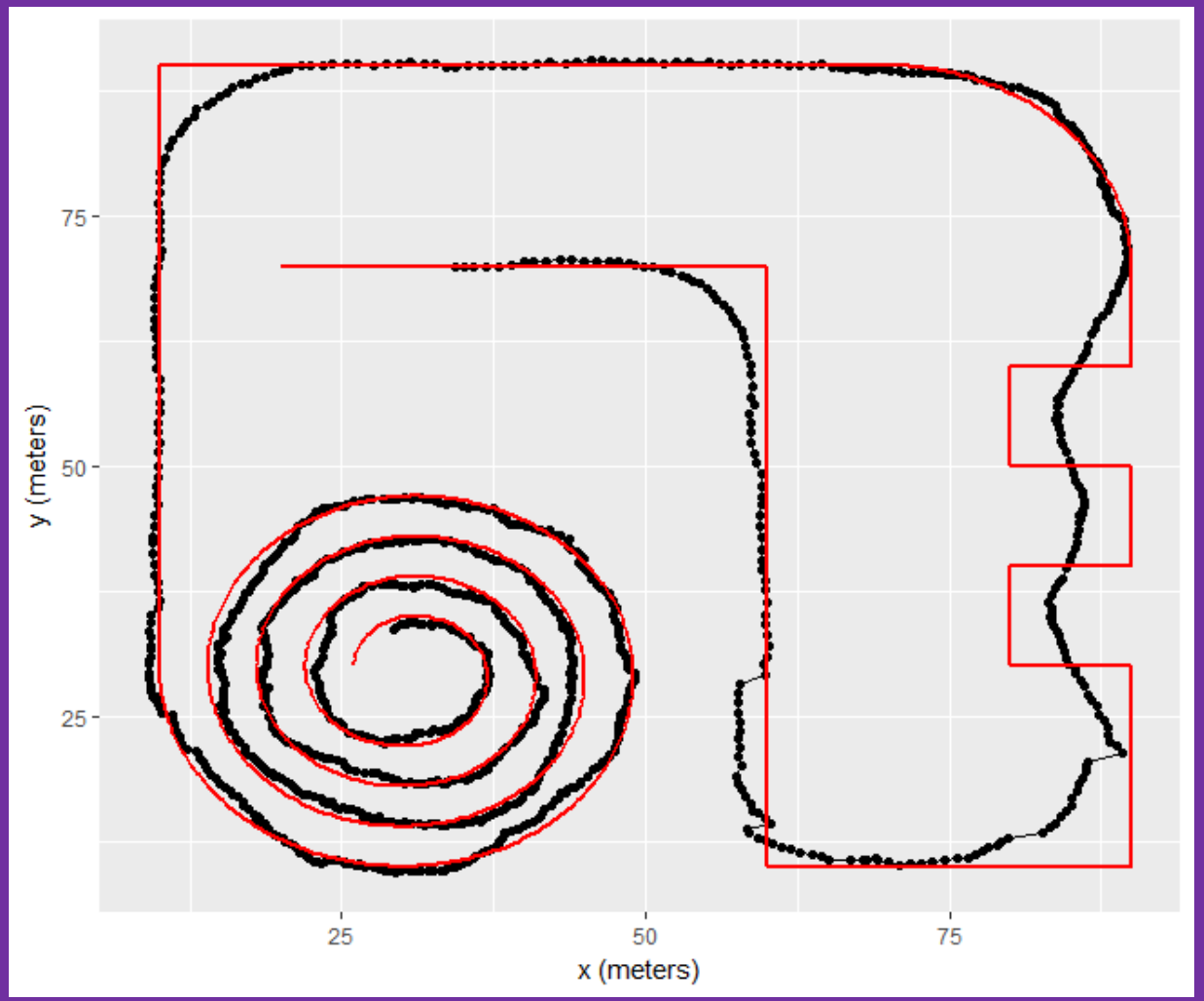
window = 10

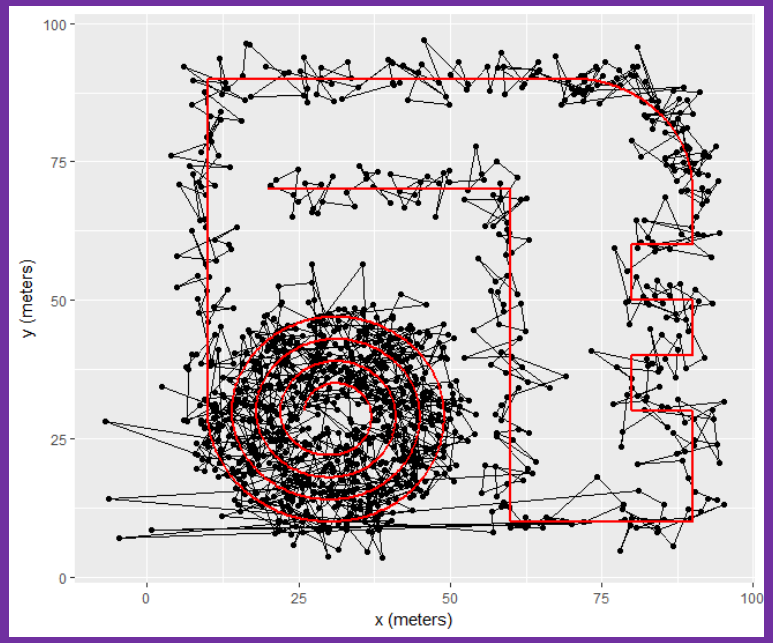

100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)

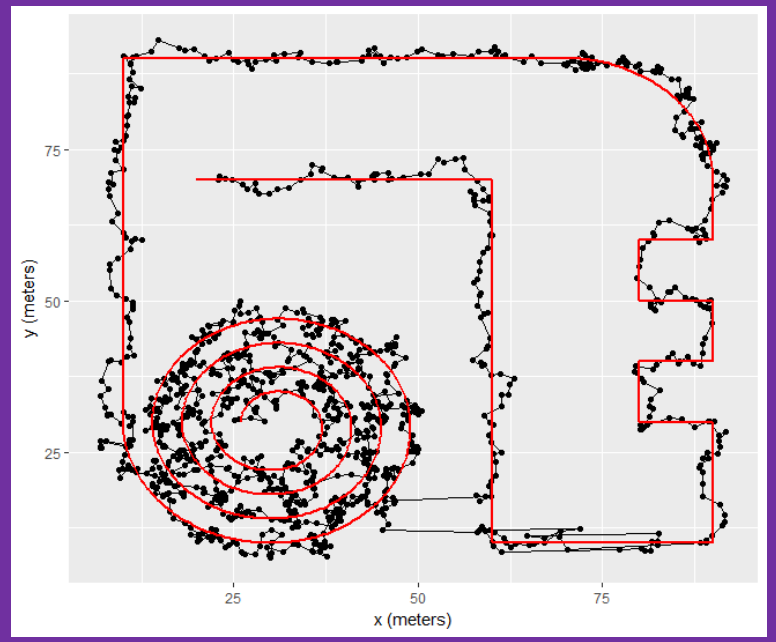

75
50
25
y (meters)
25
50
75
x (meters)

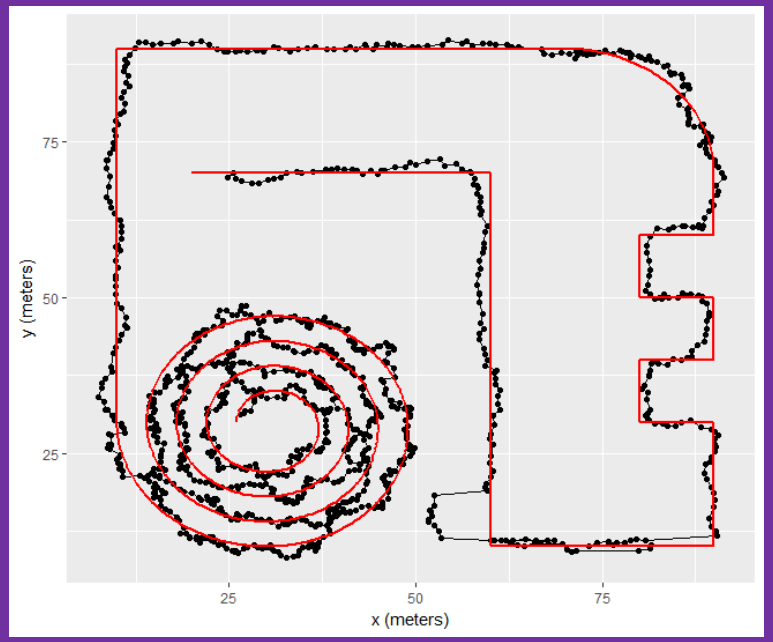
window = 30

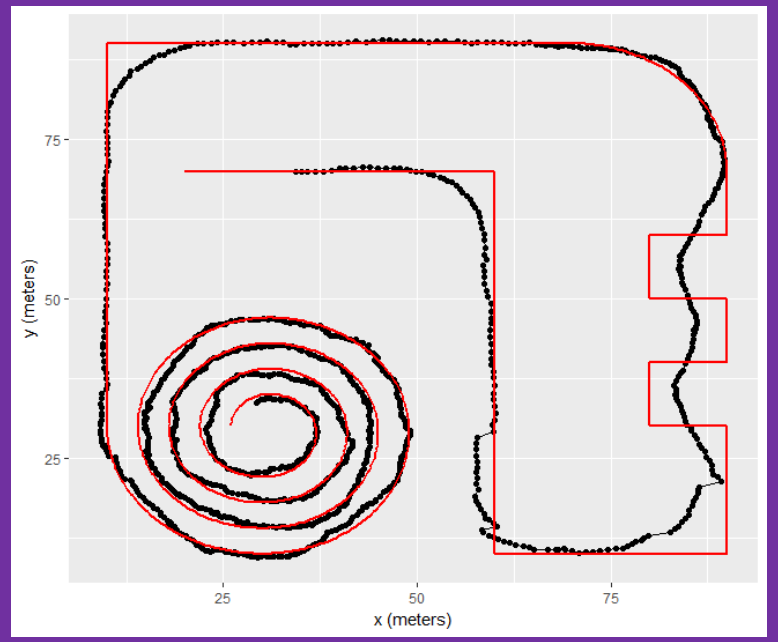

100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)


75
50
25
y (meters)
25
50
75
x (meters)


100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)

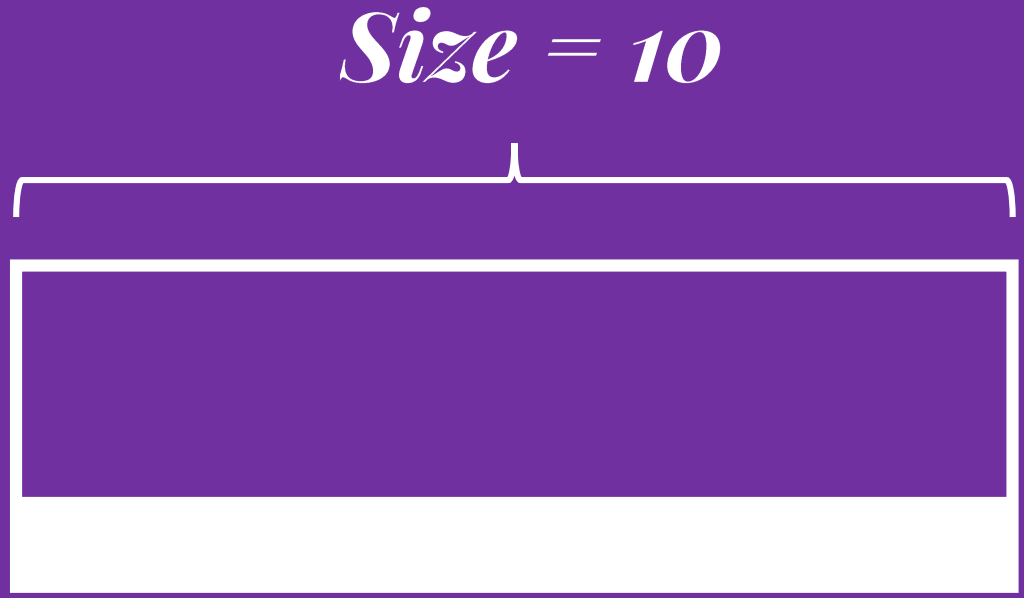

75
50
25
y (meters)
25
50
75
x (meters)


75
50
25
y (meters)
25
50
75
x (meters)


75
50
25
y (meters)
25
50
75
x (meters)

BASIC

Uniform Mean



Select font size T T T

If the window size is 10, what is the weight that each value will have?

Allow Single Choice Only | Allow Multiple Choices | Shuffle Answers | Allow Retry | Limit Attempts

0.01

0.5

0.1

That is right. The weight of each element in the window will be the inverse of the window size.

10

+ Add another answer

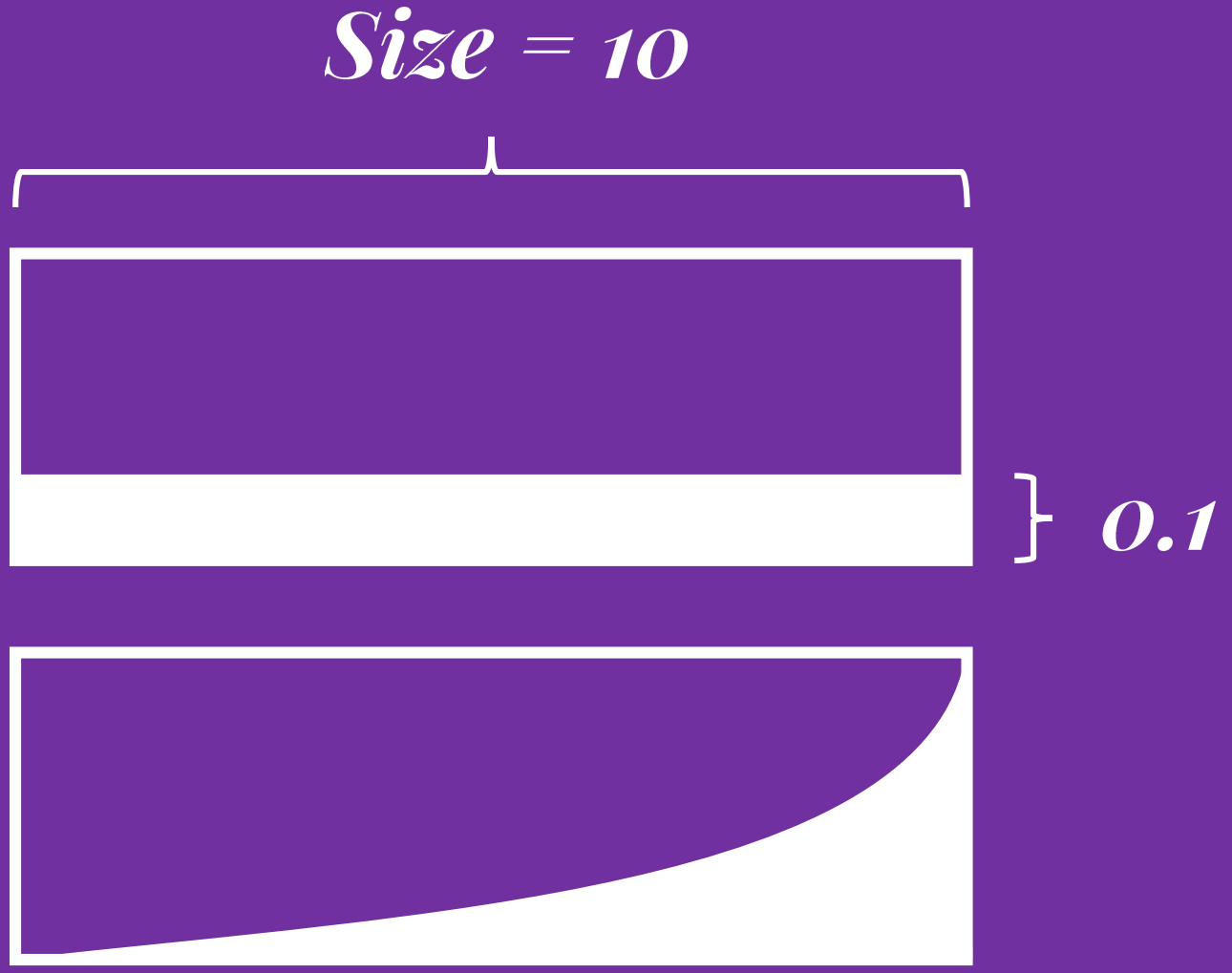
Preview

BASIC

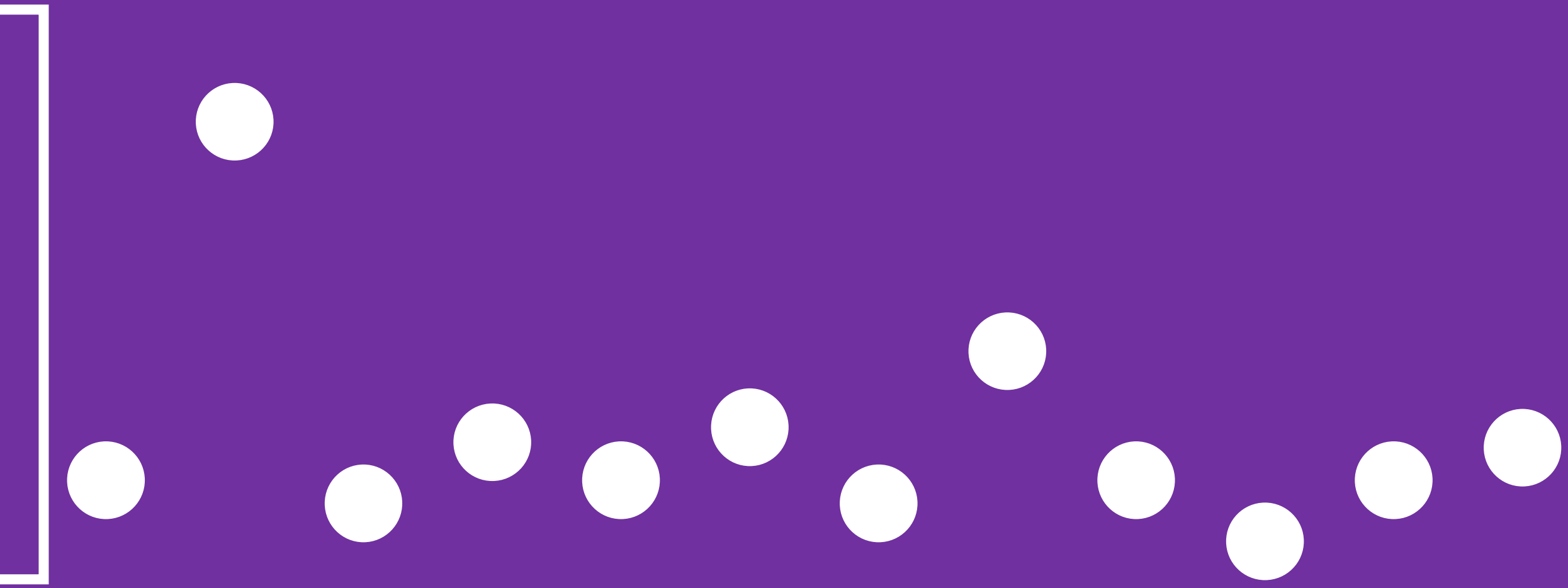
Uniform Mean

VARIATION

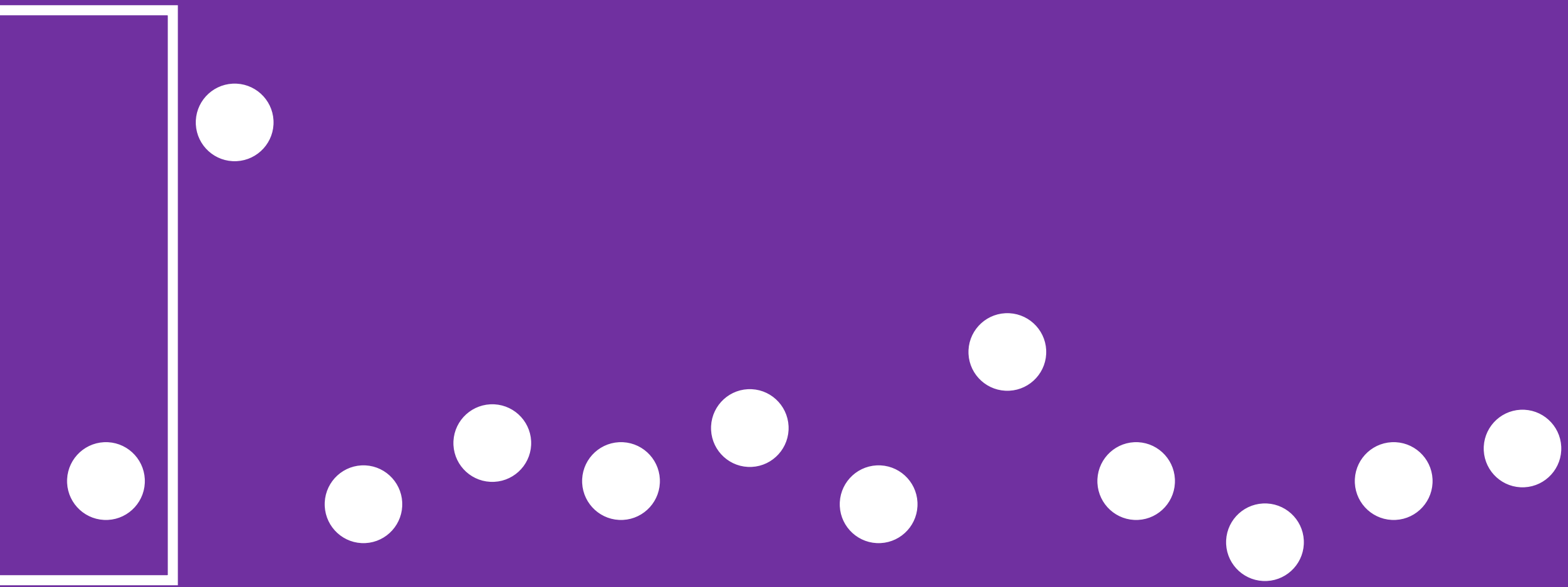
Weighted Mean



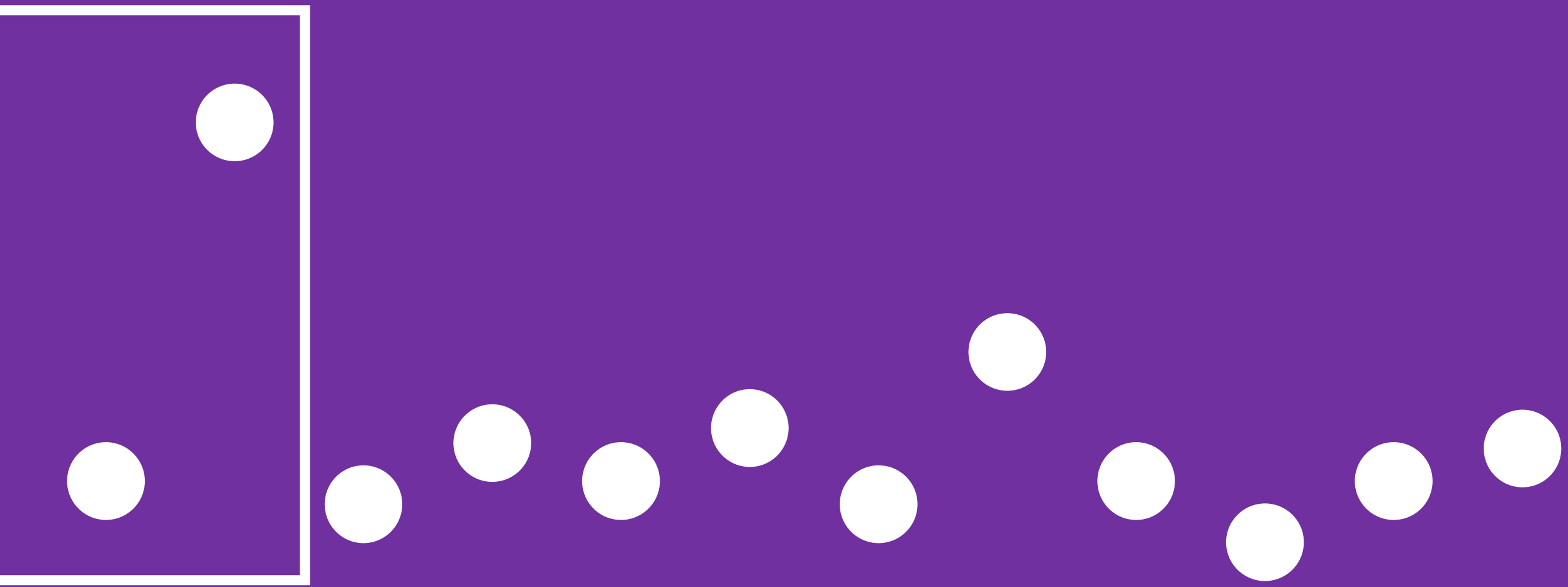
MEDIAN FILTER



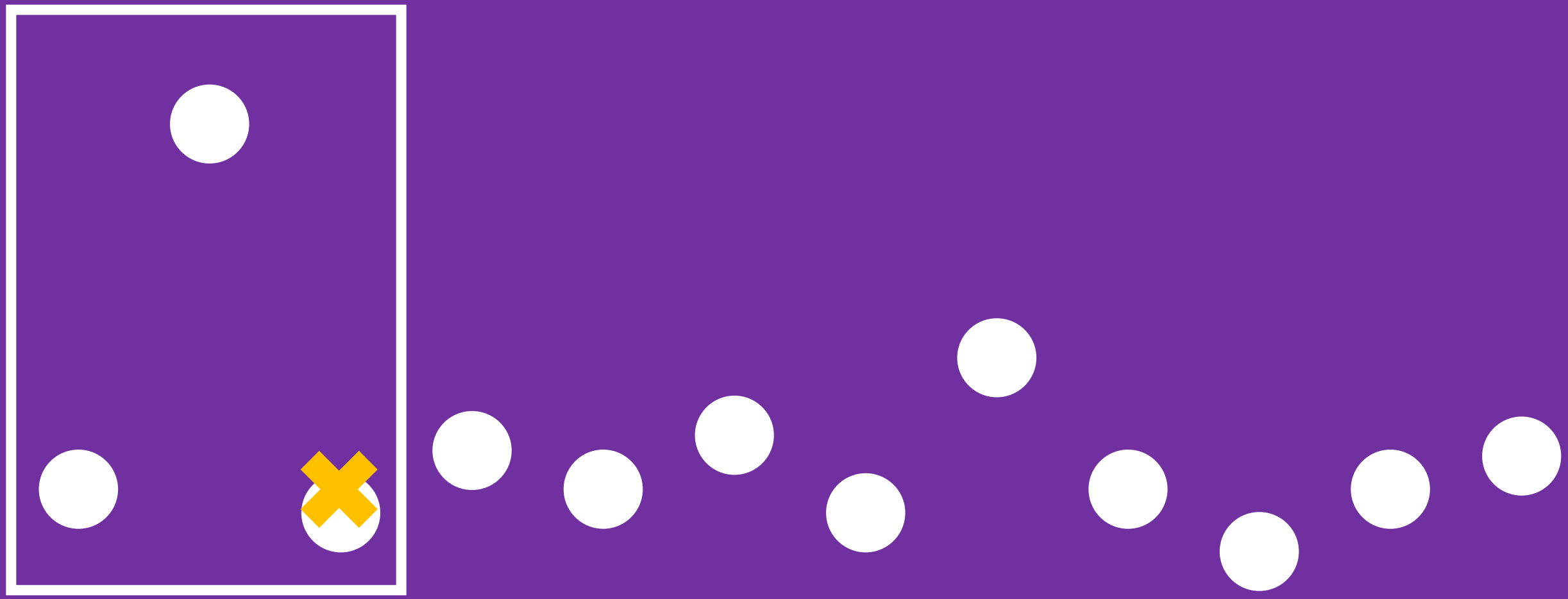
MEDIAN FILTER



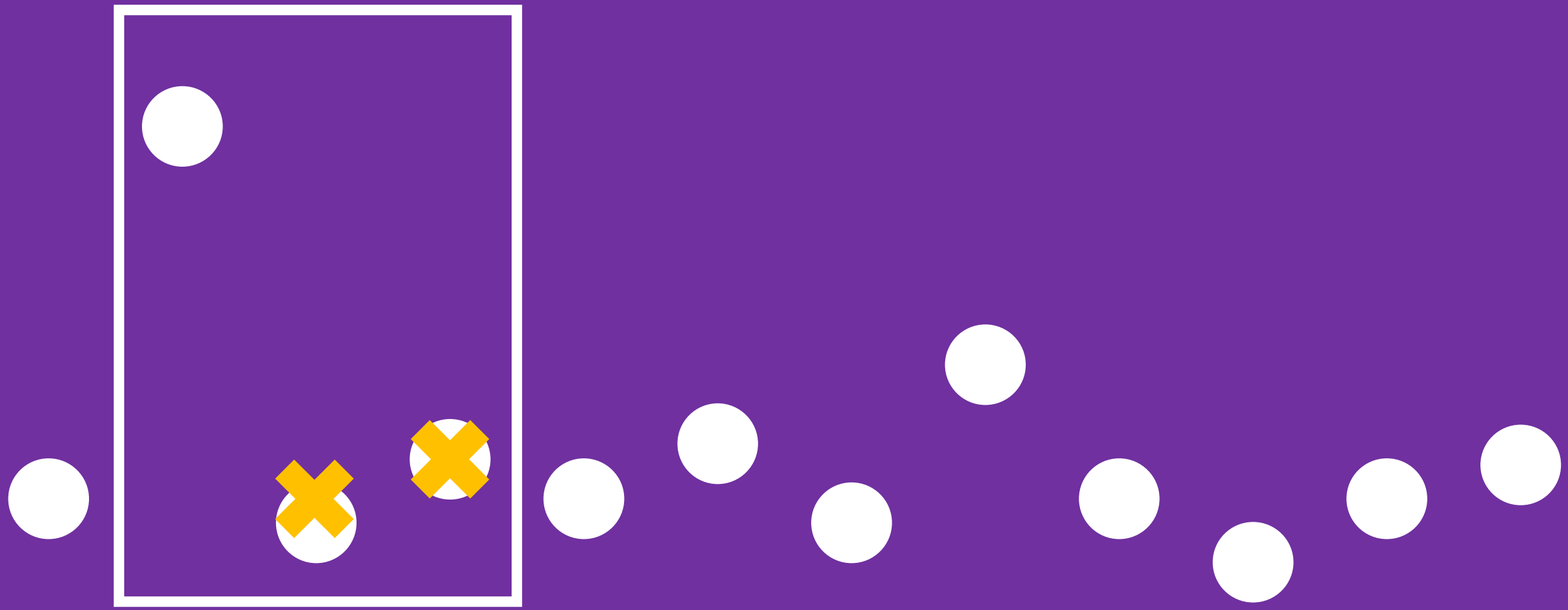
MEDIAN FILTER



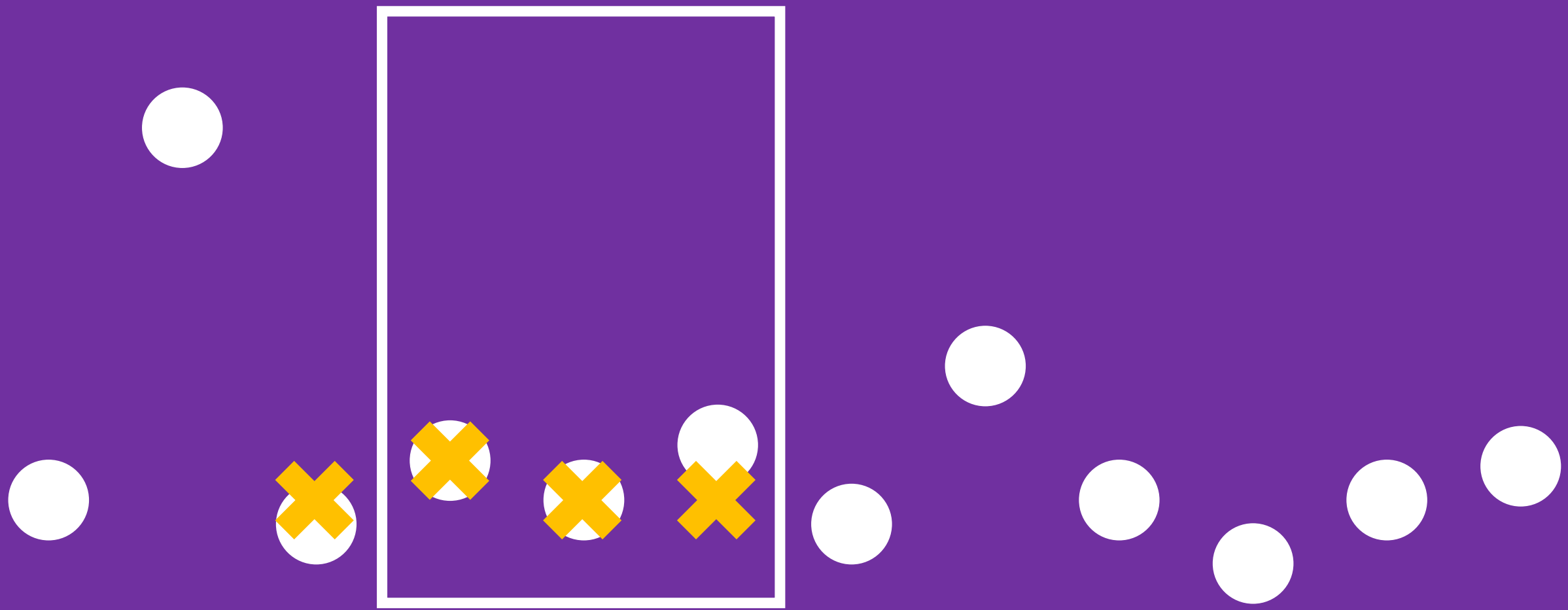
MEDIAN FILTER



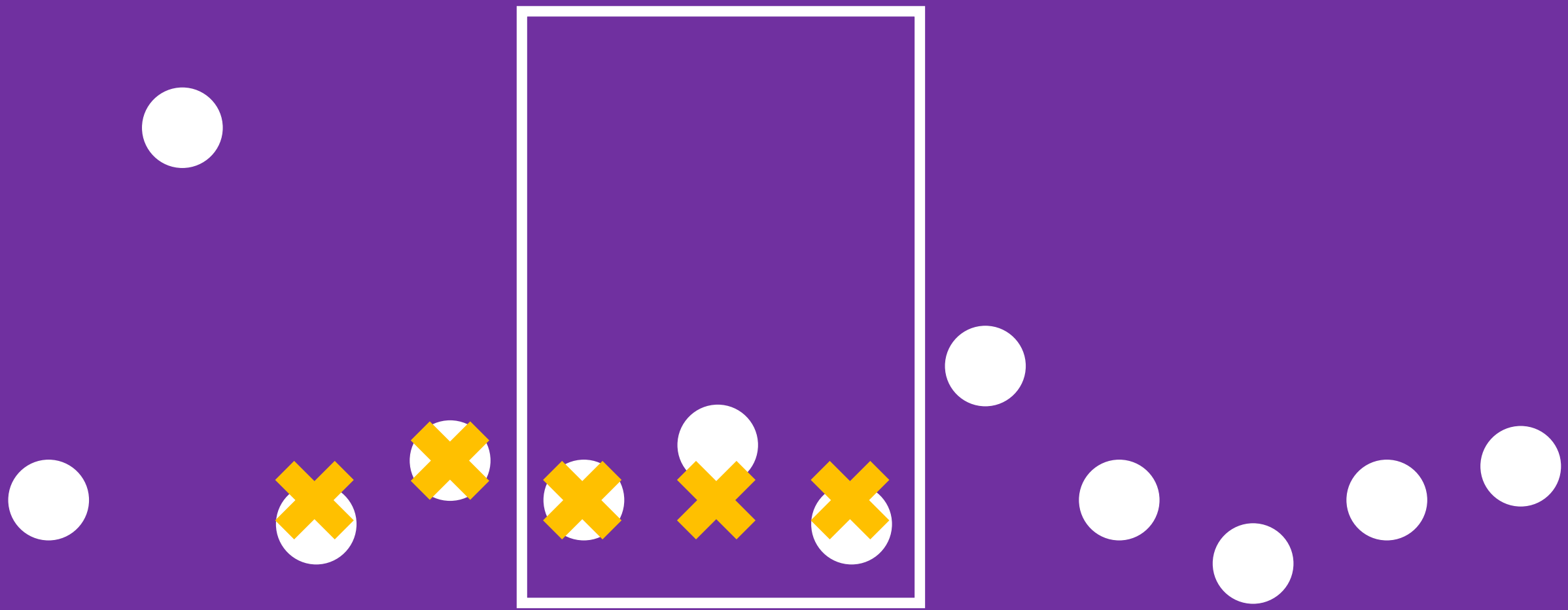
MEDIAN FILTER



MEDIAN FILTER



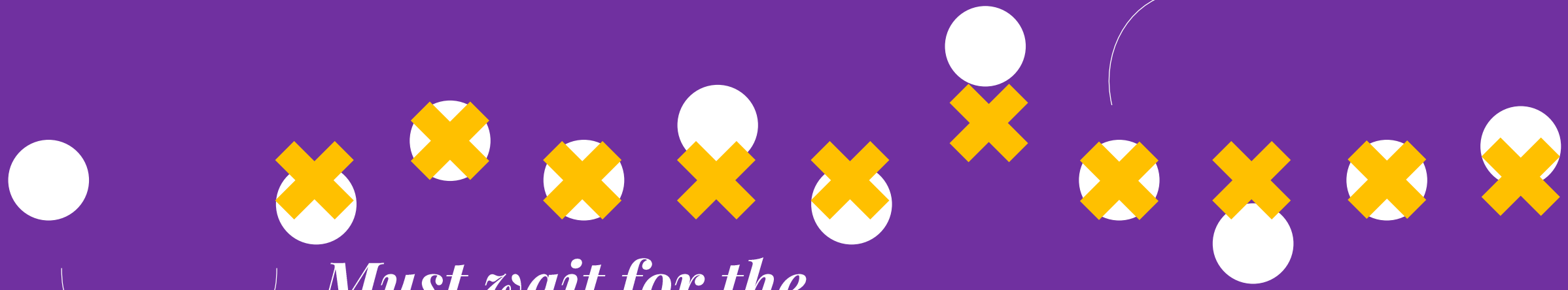
MEDIAN FILTER



MEDIAN FILTER

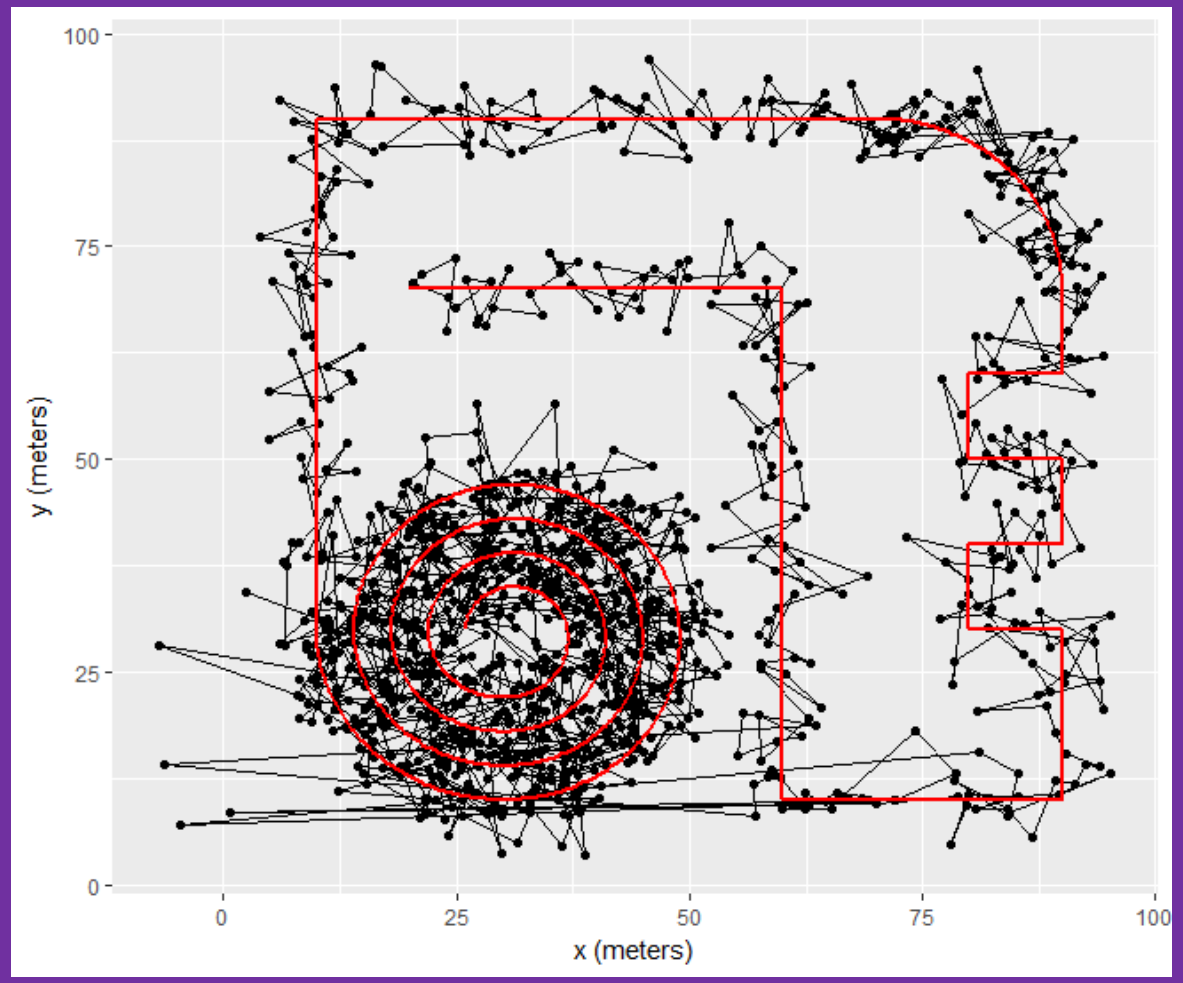
Less susceptible to outliers

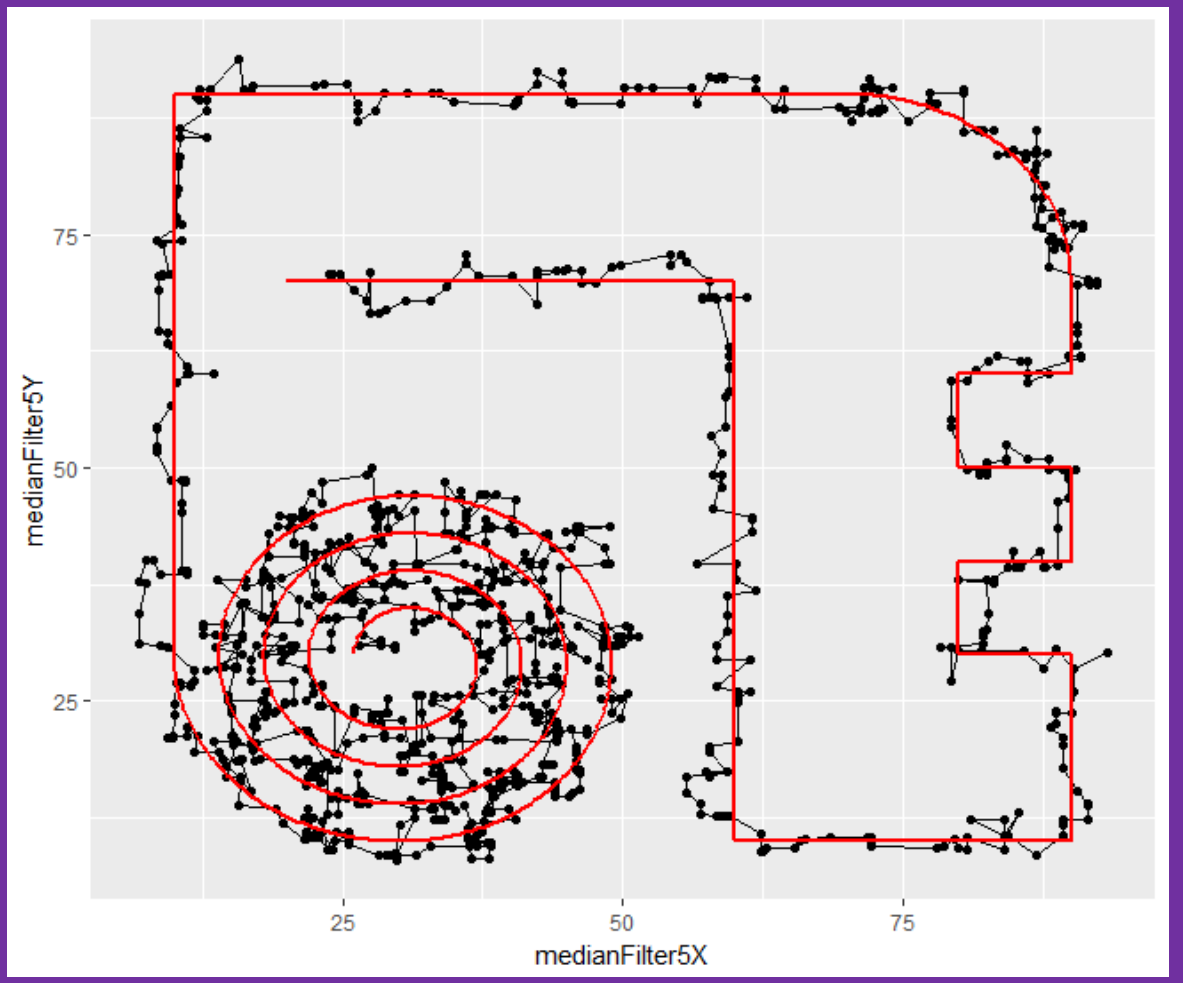
Does not make up data



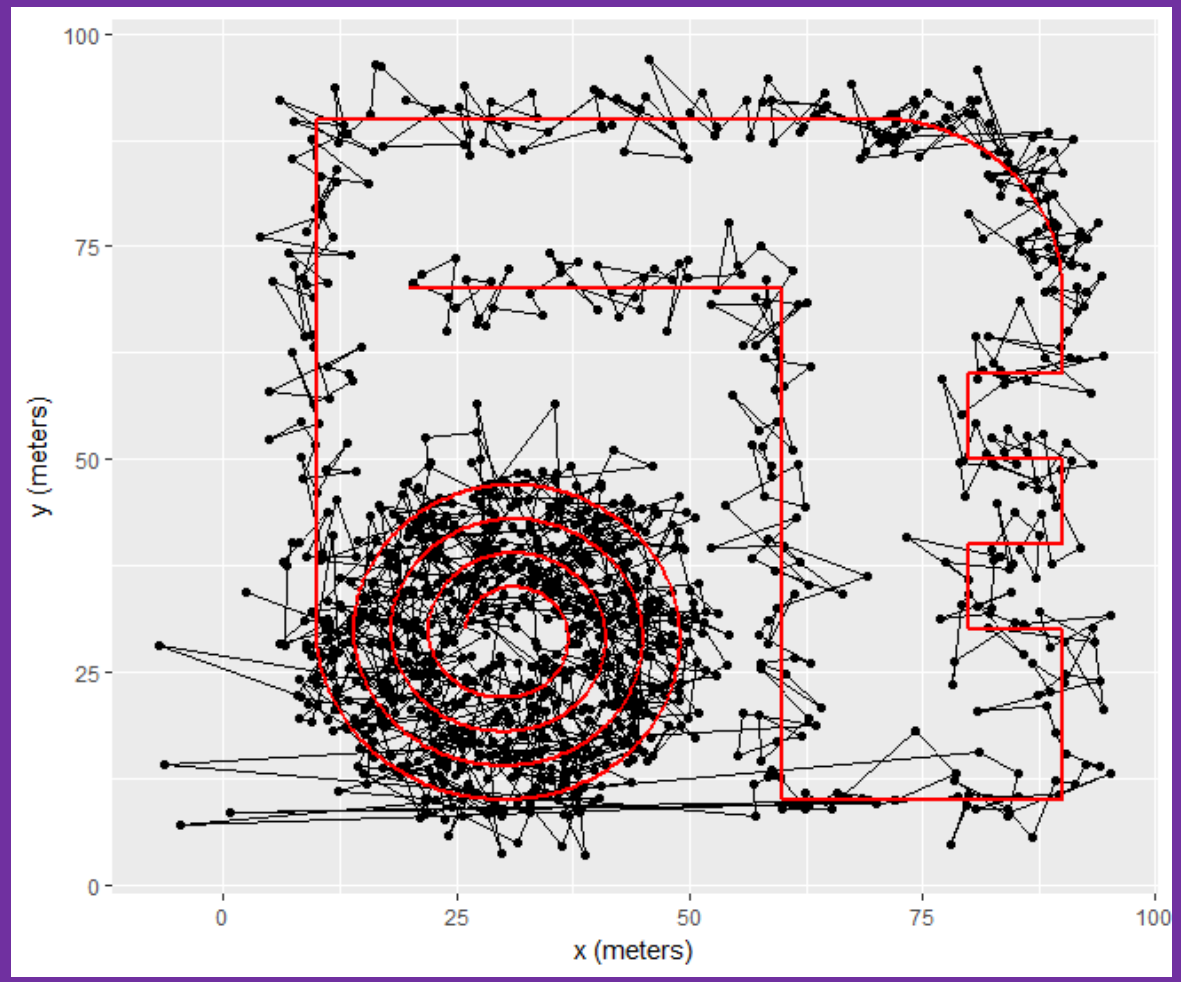
Must wait for the window to fill

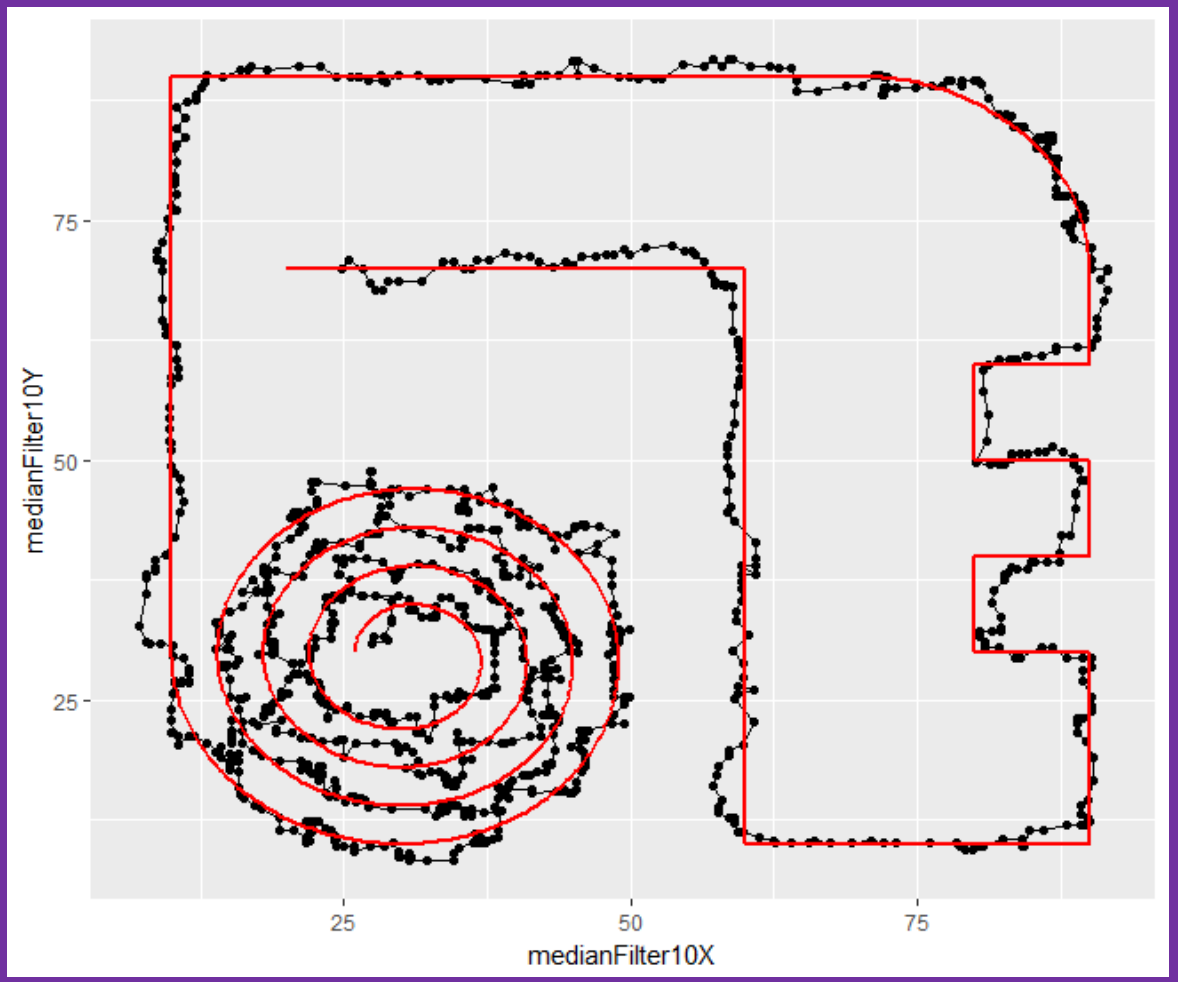
window = 5


100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)

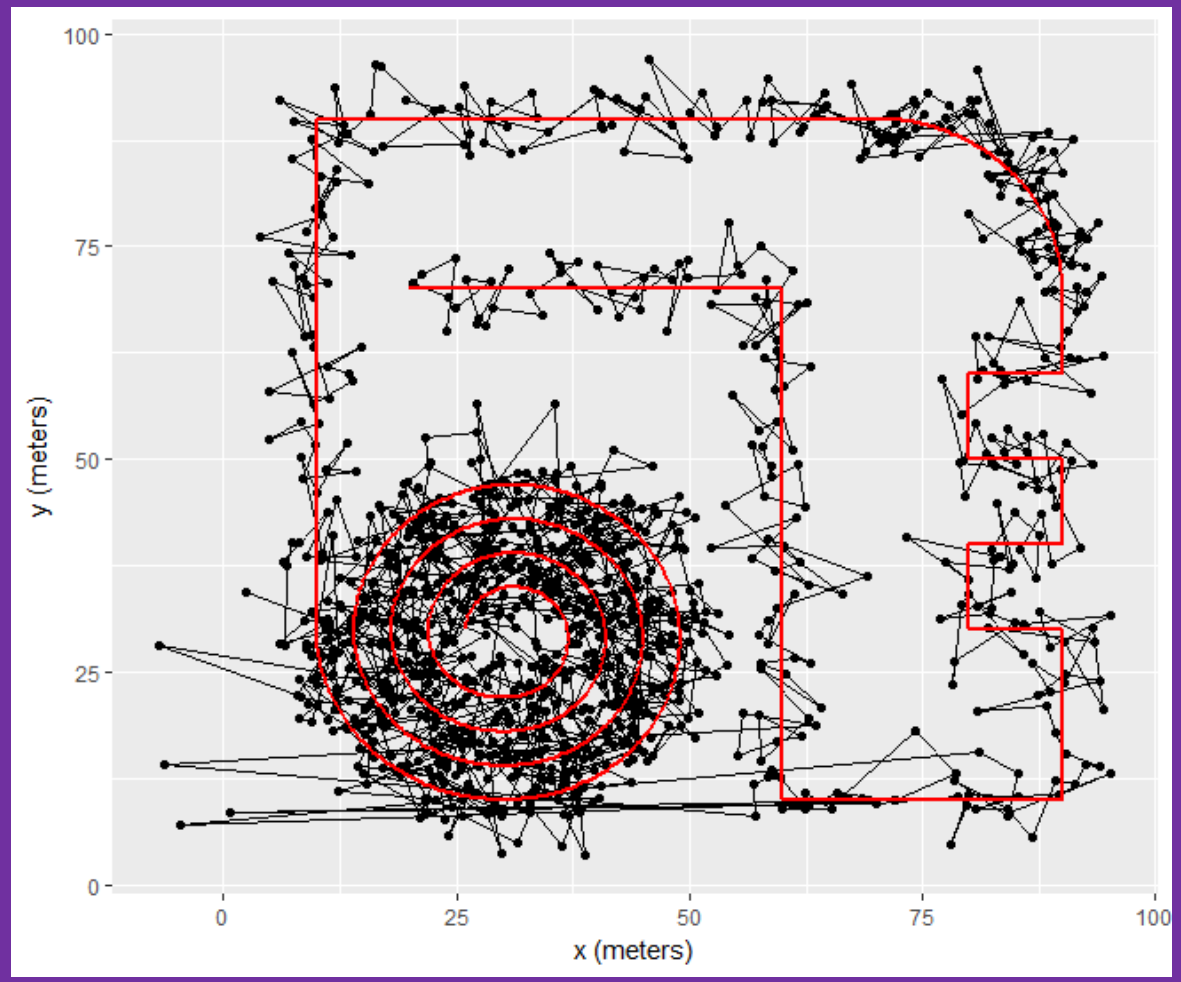

75
50
25
medianFilter5Y
25
50
75
medianFilter5X

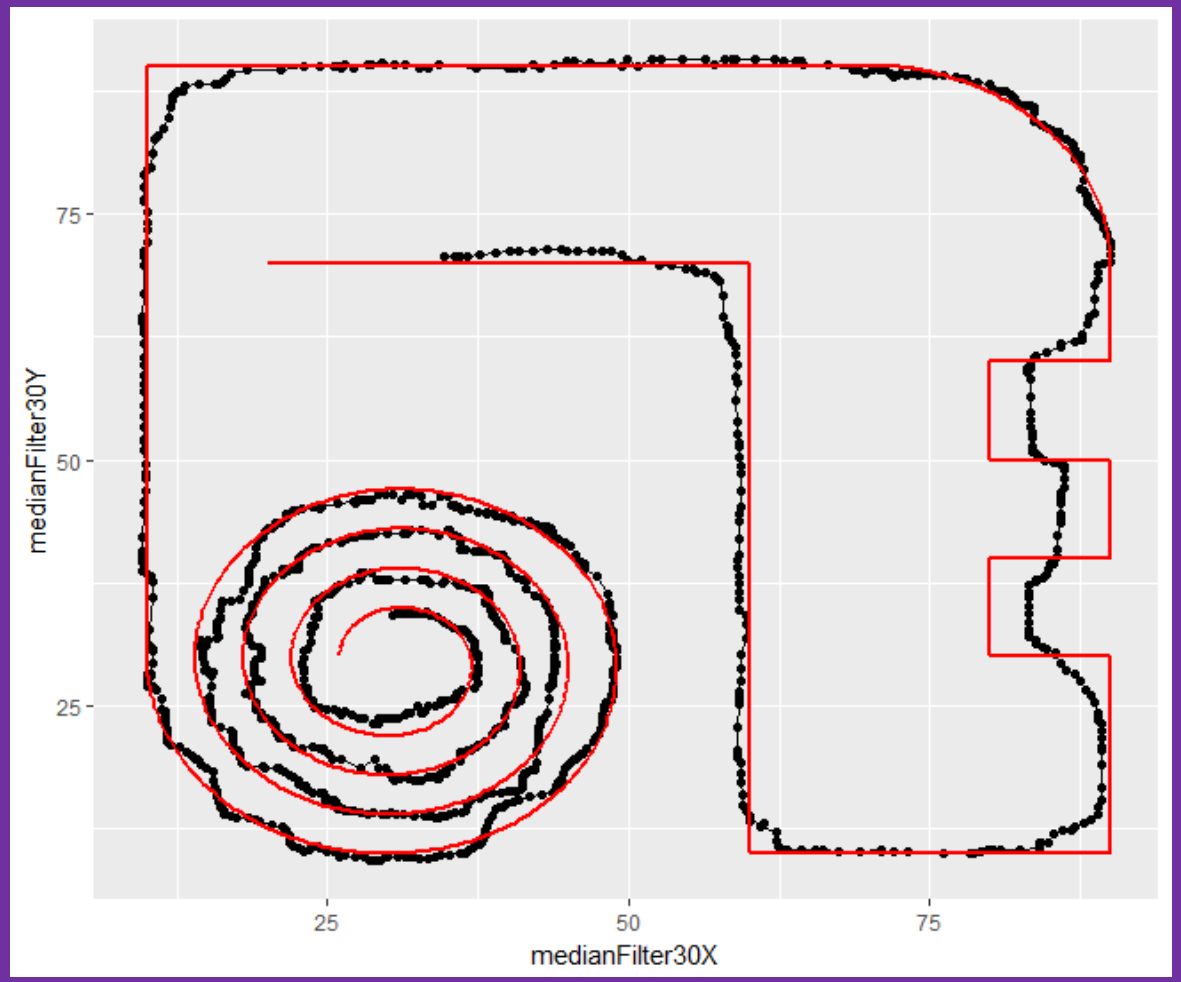
window = 10

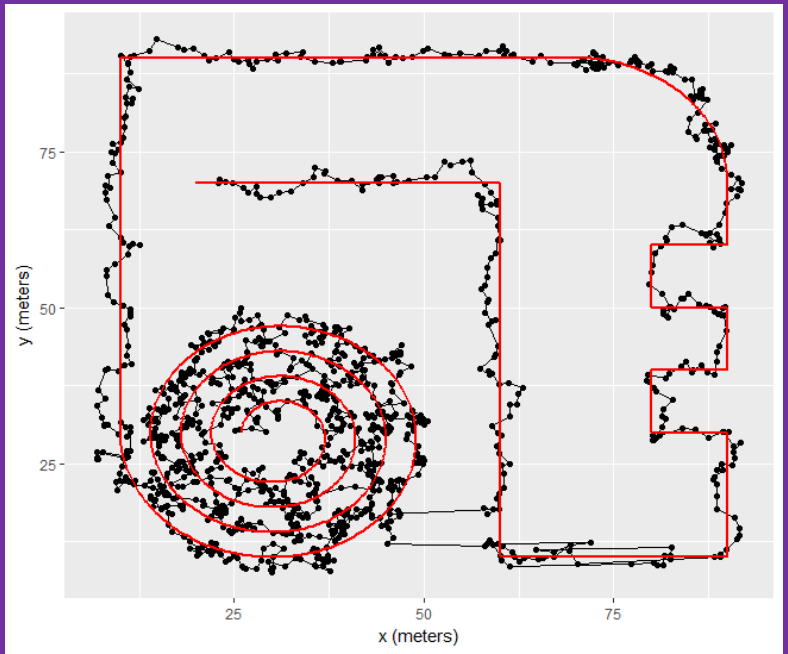

100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)

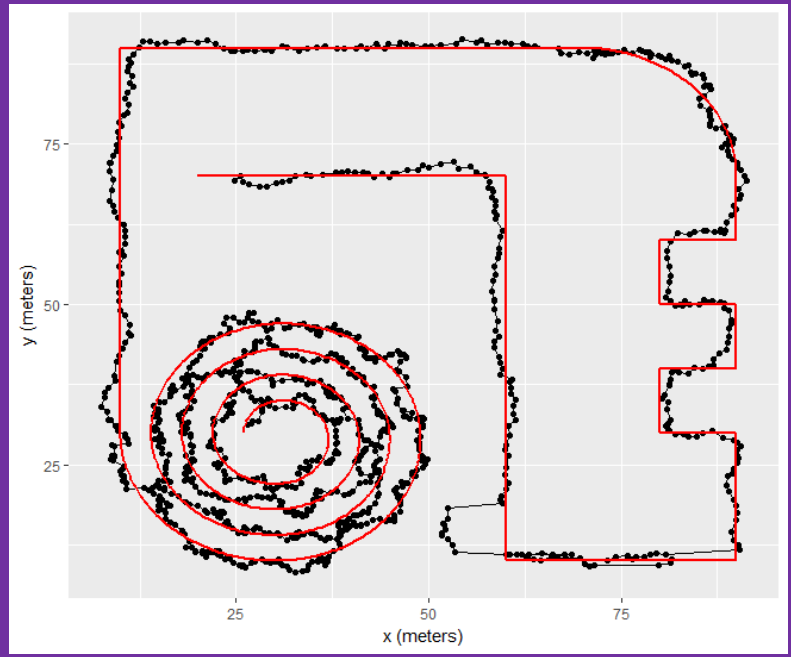

75
50
25
medianFilter10Y
25
50
75
medianFilter10X

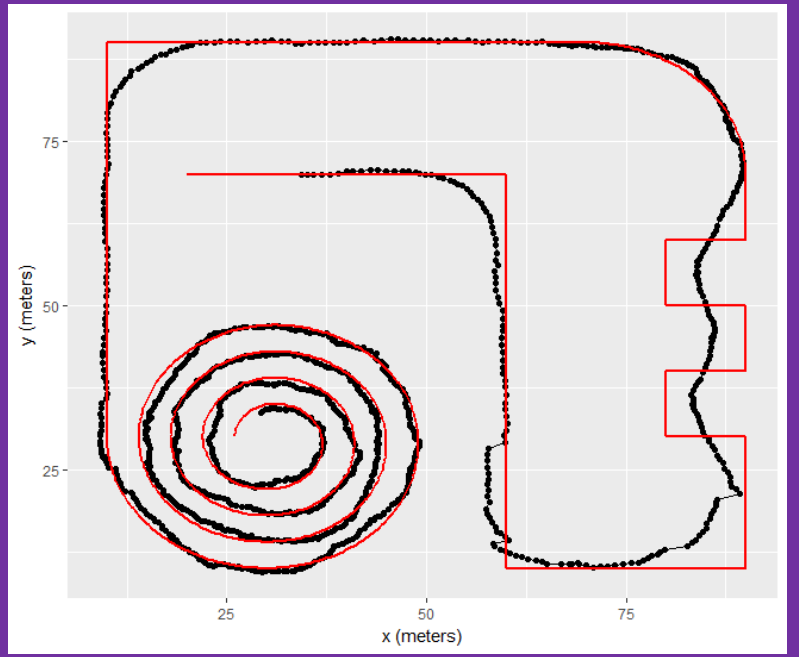
window = 30

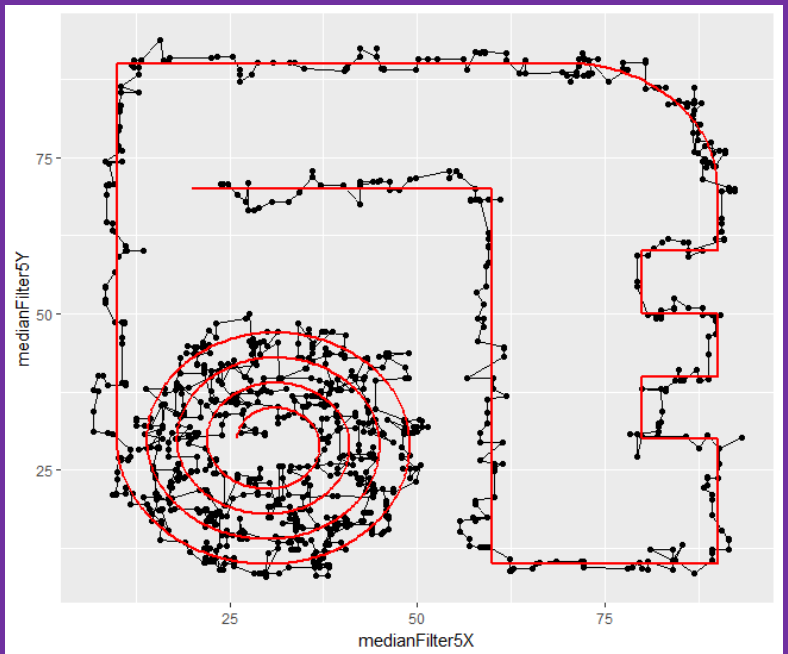

100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)


75
50
25
medianFilter30Y
25
50
75
medianFilter30X

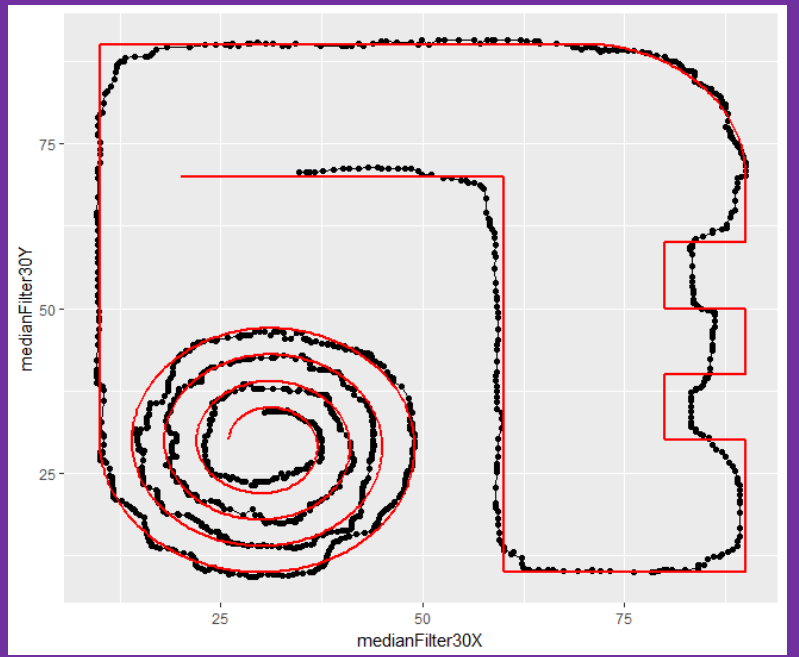

y (meters)
x (meters)
75
50
25
25
50
75


y (meters)
x (meters)
75
50
25
25
50
75

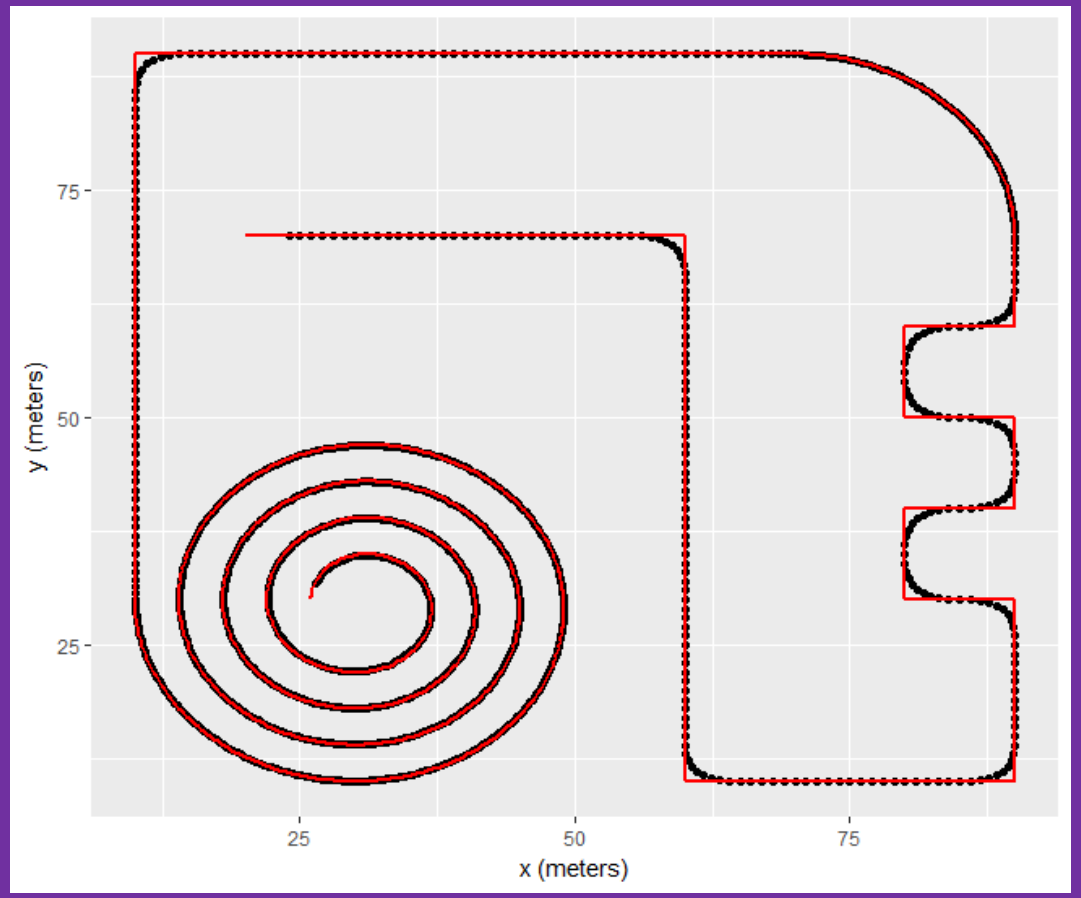

y (meters)
x (meters)
75
50
25
25
50
75


medianFilter5Y
medianFilter5X
75
50
25
25
50
75

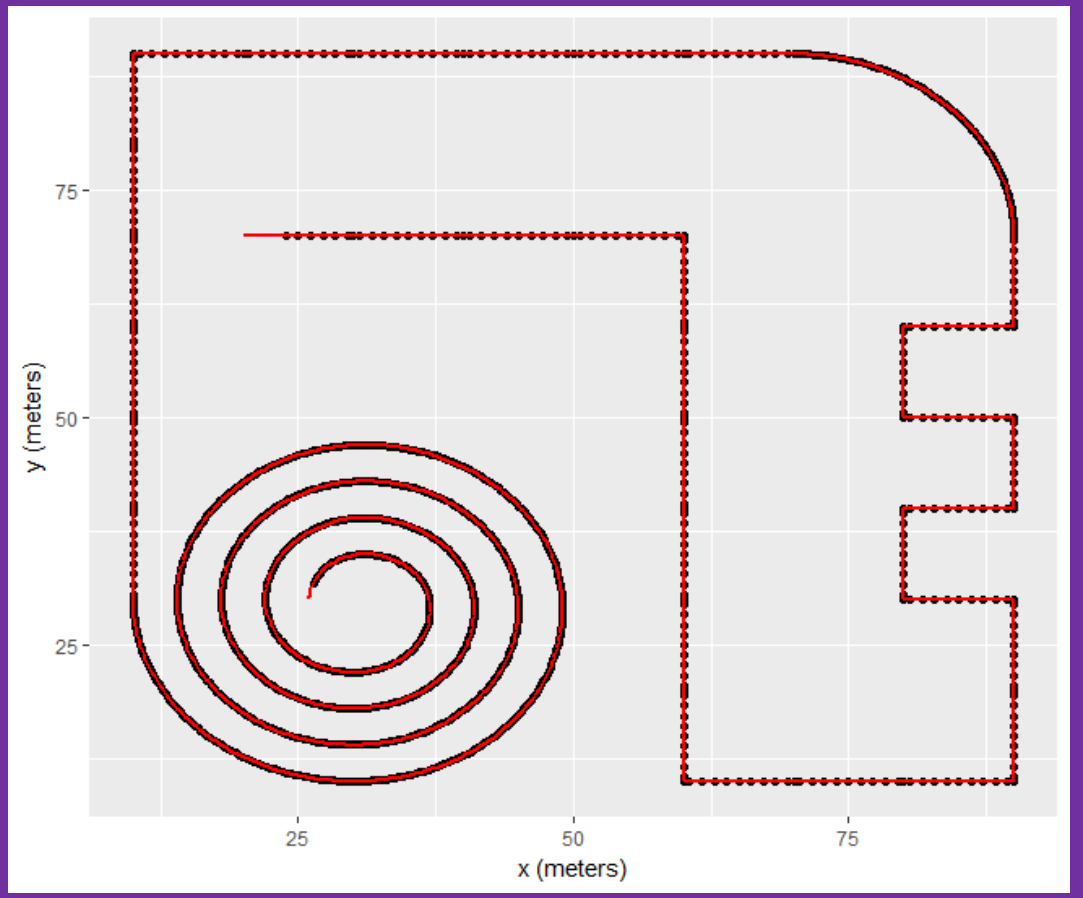

medianFilter10Y
medianFilter10X
75
50
25
25
50
75


medianFilter30Y
medianFilter30X
75
50
25
25
50
75

MEAN


75
50
25
y (meters)
25
50
75
x (meters)

MEDIAN


75
50
25
y (meters)
25
50
75
x (meters)

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Sensor Data Processing

sensorProcessing.Rmd | data_generation.R*

Knit | Insert | Run

```
---
title: "Sensor Data Processing"
output:
  html_document: default
  html_notebook: default
---

This lecture was created by Eduardo Velloso based on the chapter "Processing Sequential Sensor Data" by John Krumm.


# The Challenge

The example here focuses on a tracking challenge. Imagine you have a user who is walking around and we are recording their GPS coordinates. We would like to get the best approximation possible for their actual trajectory despite the fact that the sensor data is noisy.

To explore this problem, I generated this trajectory and simulated the sensor measurements by adding a Gaussian noise drawn from a normal distribution with mean zero and standard deviation of 3m. I also added ten random outliers drawn from a normal distribution with sd = 15m.

```{r}
```

20:17 | Chunk 1 | R Markdown

Console | R Markdown

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/.RData]

> ?rnorm
>
```

Environment | History

Import Dataset | List

Global Environment

Environment is empty

Files | Plots | Packages | Help | Viewer

R: The Normal Distribution | Find in Topic

Normal {stats} R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
```

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Sensor Data Processing

sensorProcessing.Rmd | data_generation.R*

Knit | Insert | Run

```
Notice the effect of the window size on the outliers on the second corner of the path.

39
40 ```{r}
41 # MEAN FILTER window = 5
42 path$meanFilter5X <- rollapply(path$noisyX, 5, mean, fill = "extend")
43 path$meanFilter5Y <- rollapply(path$noisyY, 5, mean, fill = "extend")
44
45 ggplot(path, aes(x=meanFilter5X,
   y=meanFilter5Y))+geom_path()+geom_point()+geom_path(aes(x=x,y=y),col='red',size=1)+xlab('x
   (meters)')+ylab('y (meters)')+ggtitle('Mean Filter with a window size = 5 samples')
   ```
46
47
48 ```{r}
49 # MEAN FILTER window = 10
50 path$meanFilter10X <- rollapply(path$noisyX, 10, mean, fill = "extend")
51 path$meanFilter10Y <- rollapply(path$noisyY, 10, mean, fill = "extend")
52
53 ggplot(path, aes(x=meanFilter10X,
   y=meanFilter10Y))+geom_path()+geom_point()+geom_path(aes(x=x,y=y),col='red',size=1)+xlab('x
   (meters)')+ylab('y (meters)')+ggtitle('Mean Filter with a window size = 10 samples')
   ```
54
55
```

42:31 | Chunk 2 | R Markdown

Console | R Markdown

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/.RData]

> ?rnorm
>
```

Environment | History

Import Dataset | List

Global Environment

Environment is empty

Files | Plots | Packages | Help | Viewer

R: The Normal Distribution | Find in Topic

Normal {stats} R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

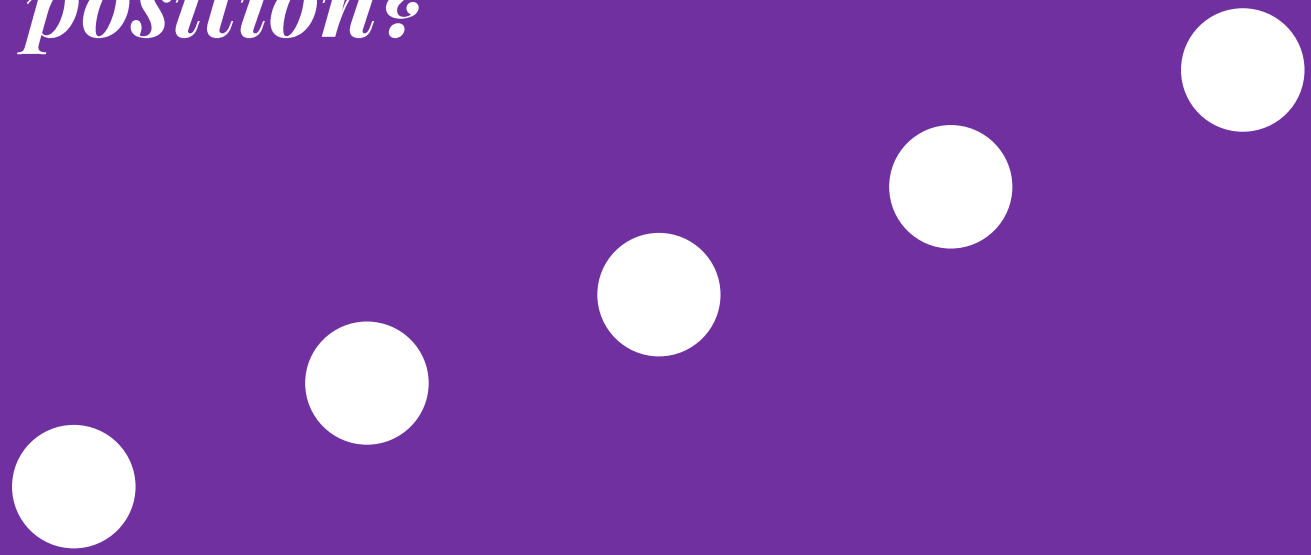
MEAN/MEDIAN FILTER

+ *Easy to implement*
+ *Efficient*
+ *Great cost-benefit*

− *Laggy*

− *No dynamic model*

Where is the next position?

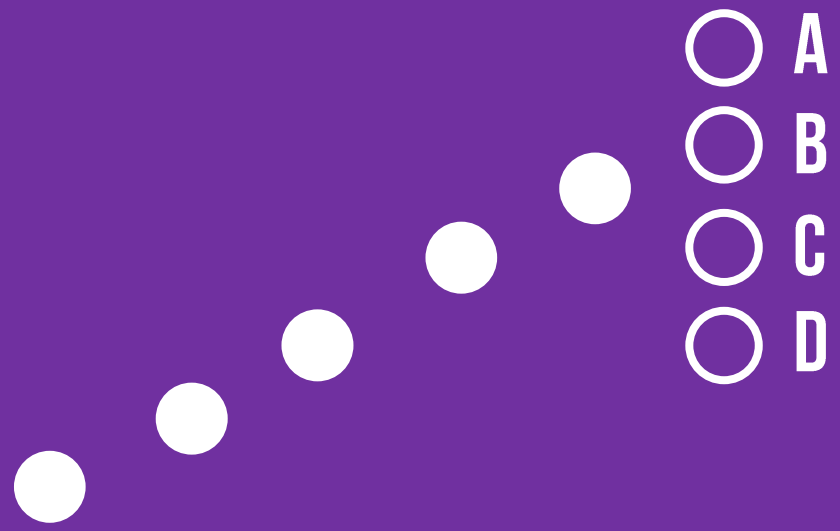


A

B

C

D



Select font size T T T

What is the most likely next position of the object?

Allow Single Choice Only Allow Multiple Choices

Shuffle Answers Allow Retry Limit Attempts

A

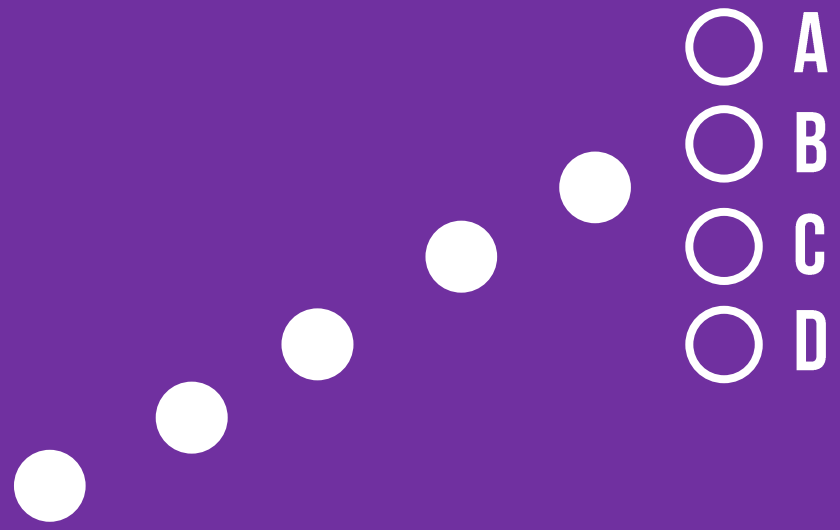
B

C

D

+ Add another answer

Preview



Select font size T T T

Where would a median filter (N=5) predict the object to be next?

Allow Single Choice Only Allow Multiple Choices

Shuffle Answers Allow Retry Limit Attempts

A

B

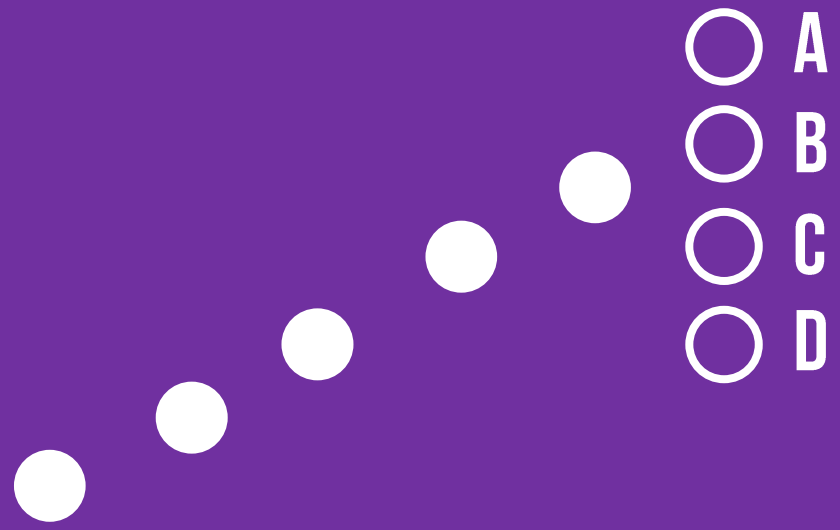
C

D

+ Add another answer

Preview

Terms | Privacy & Cookies



Select font size T T T

Where would a weighted mean filter (N=5) predict the object to be next?

◉ Allow Single Choice Only ○ Allow Multiple Choices

☐ Shuffle Answers ☑ Allow Retry ☐ Limit Attempts

A

B

C

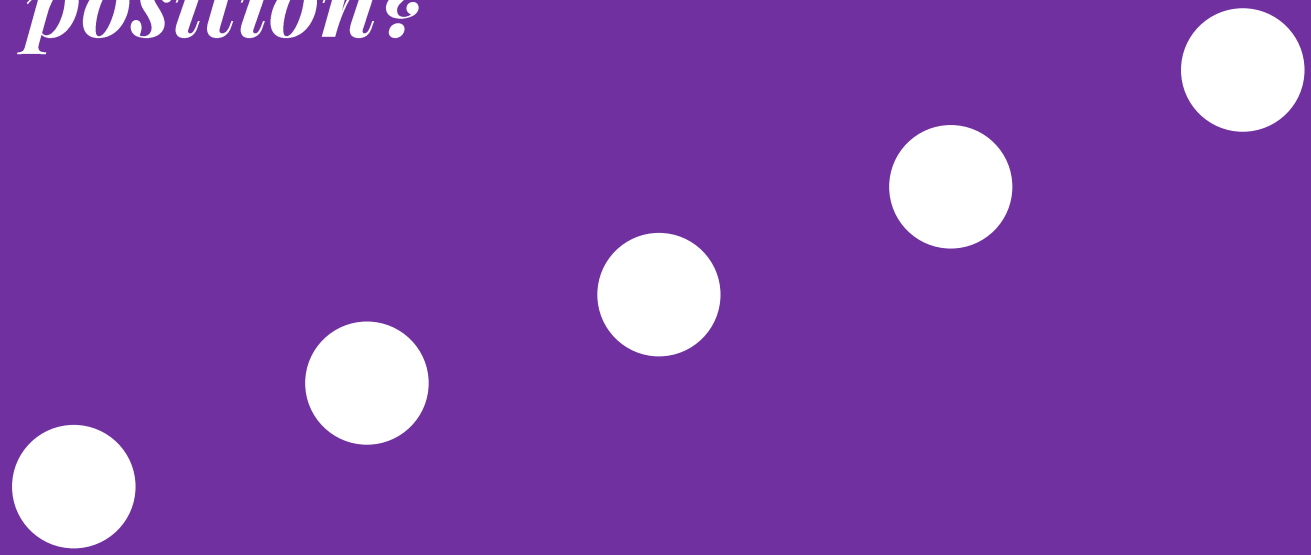
D

+ Add another answer

Preview

Terms | Privacy & Cookies

Where is the next position?



A

B

C

D

ASSUMPTIONS

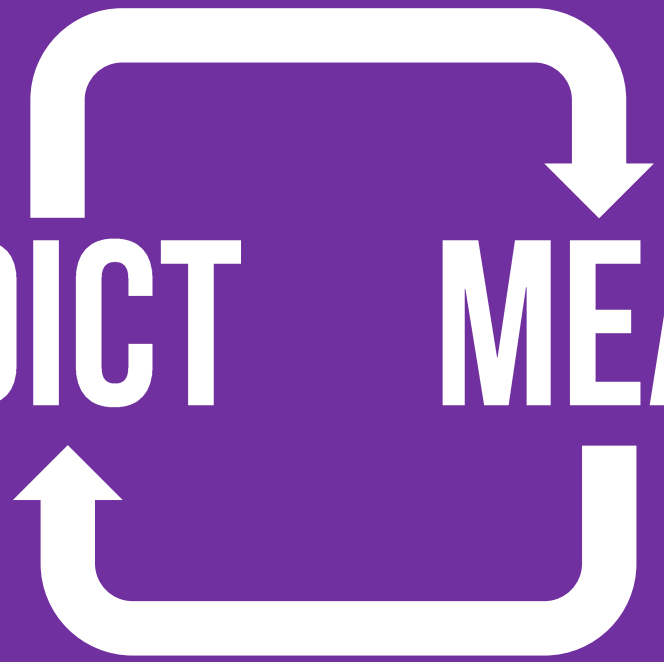
MEAN/MEDIAN FILTER

Trajectory is smooth

KALMAN FILTER

Underlying velocity

KALMAN FILTER



Assumptions about behaviour!

$$z_i = x_i + v_i$$

noise

$$v_i = \begin{pmatrix} v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \sim \begin{pmatrix} N(0, \sigma) \\ N(0, \sigma) \end{pmatrix}$$

Normal Dist

Mean=0

sd=3

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

measurement

$$x_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

state

measurement

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

state

$$x_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

measurement

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

state

$$x_i = \begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix}$$

x_i, y_i — *positions*

$v_i^{(x)}, v_i^{(y)}$ — *velocities*

Variables that are not measured directly

measurement

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

state

$$x_i = \begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix}$$

From state to measurement

$$\boldsymbol{z_i = x_i + v_i}$$

measurement

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

state

$$x_i = \begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix}$$

$$H = \begin{bmatrix} & \end{bmatrix}$$

From state to measurement

$$\boldsymbol{z_i = H\, x_i + v_i}$$

measurement

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

state

$$x_i = \begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix}$$

From state to measurement

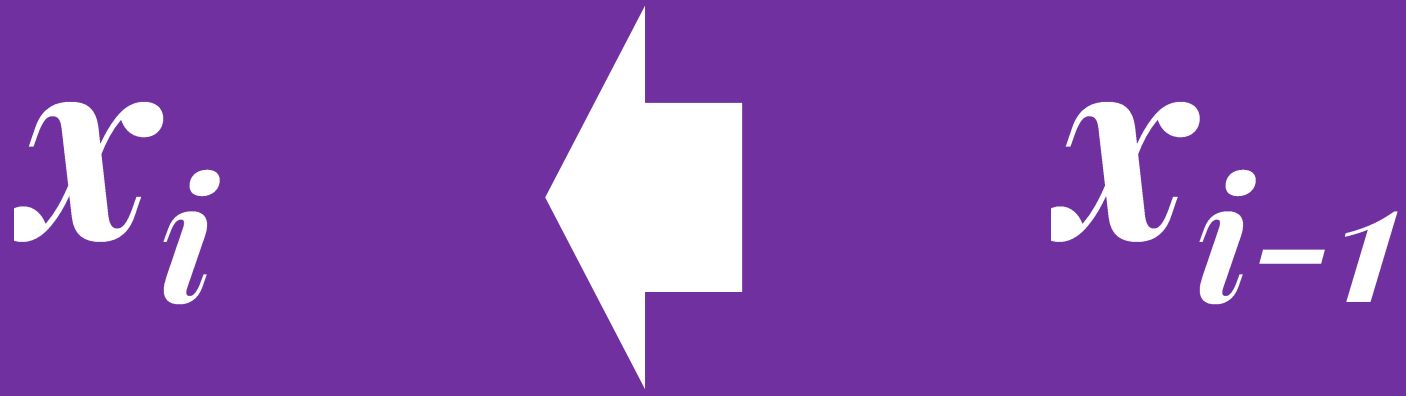
$$\boldsymbol{z_i = H\,x_i + v_i}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$v_i \sim N(0, \boldsymbol{R_i})$$

$$R_i = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

MODELLING THE DYNAMICS



Select font size T T T

What are the dimensions of vectors x_i and x_i-1?

Allow Single Choice Only | Allow Multiple Choices | Shuffle Answers | Allow Retry | Limit Attempts

1x4

2x1

2x4

4x1

That is right. The vector will have 4 rows and 1 column

4x2

4x4

+ Add another answer

Preview

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \Leftarrow \left[\quad \right] \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{bmatrix} \quad \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

$$x_i = x_{i-1} + v_{i-1}\Delta t$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \Leftarrow \begin{bmatrix} \quad \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

$$x_i = x_{i-1} + v_{i-1}\Delta t$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ & & & \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

$$x_i = x_{i-1} + v_{i-1}\Delta t$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ & & & \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

$$x_i = x_{i-1} + v_{i-1}\Delta t$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ & & & \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

$$x_i = x_{i-1} + v_{i-1}\Delta t$$

$$v_i = v_{i-1}$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix}$$

$$x_i = x_{i-1} + v_{i-1}\Delta t$$

$$v_i = v_{i-1}$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \leftarrow \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix} + \textit{Noise}$$

MODELLING THE DYNAMICS

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \Leftarrow \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\phi_{i-1}} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ N(0, \sigma_s) \\ N(0, \sigma_s) \end{pmatrix}}_{w_{i-1}}$$

$$\begin{pmatrix} x_i \\ y_i \\ v_i^{(x)} \\ v_i^{(y)} \end{pmatrix} \Leftarrow \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\phi_{i-1}} \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ v_{i-1}^{(x)} \\ v_{i-1}^{(y)} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ N(0,\sigma_s) \\ N(0,\sigma_s) \end{pmatrix}}_{w_{i-1}}$$

$$w_{i-1} \sim N(0, Qi)$$

$$Qi = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & 0 \\ 0 & 0 & 0 & \sigma_s^2 \end{bmatrix}$$

ALL TOGETHER

Measurement matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Dynamics of states

$$\phi_{i-1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Measurement noise

$$v_i \sim N(0, \boldsymbol{R}_i)$$

Dynamic noise

$$w_{i-1} \sim N(0, Qi)$$

ALL TOGETHER

Initial state estimate

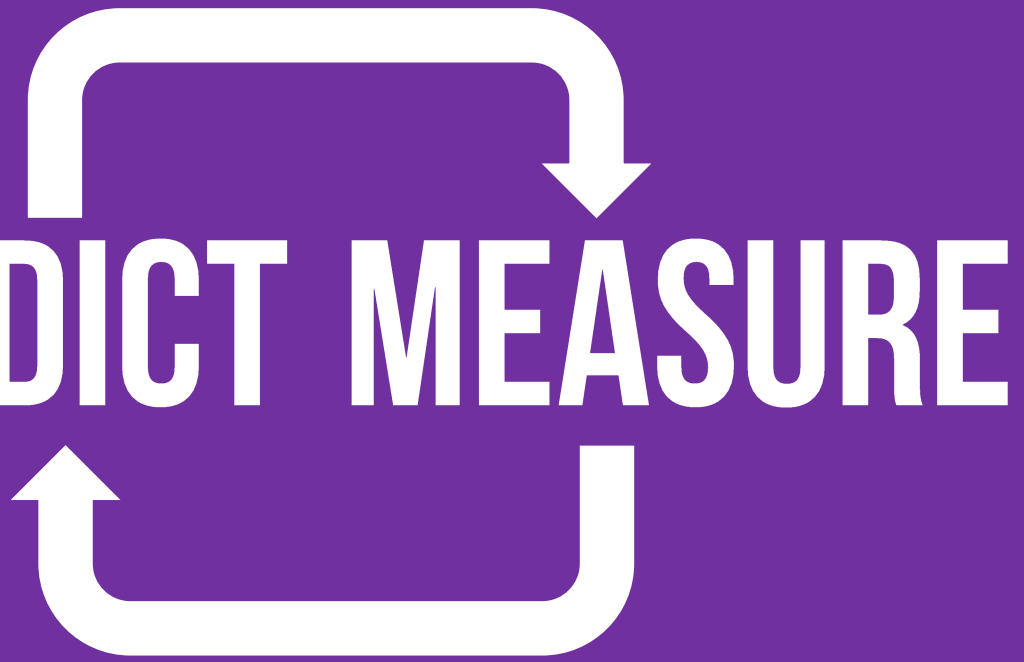
$$x_0 = \begin{pmatrix} x_0 \\ y_0 \\ v_0^{(x)} \\ v_0^{(y)} \end{pmatrix} = \underbrace{\begin{pmatrix} z_0^{(x)} \\ z_0^{(y)} \\ 0 \\ 0 \end{pmatrix}}$$

e.g. First measurement of position and zero velocity

Initial estimate of state error covariance

$$P_0 = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & 0 \\ 0 & 0 & 0 & \sigma_s^2 \end{bmatrix}$$

THE ALGORITHM



STEP 1: PREDICT

Extrapolate the state:

$$x_i^{predicted} = \phi_{i-1}\, x_{i-1}^{corrected}$$

Extrapolate the state error covariance:

$$P_i^{predicted} = \phi_{i-1}\, P_{i-1}^{corrected}\, \phi_{i-1}^{T} + Q_{i-1}$$

STEP 2: MEASURE

Compute the Kalman gain:

$$K_i = \frac{P_i^{predicted} H_i^T}{H_i P_i^{predicted} H_i^T + R_i}$$

THE KALMAN GAIN

Uncertainty propagated by the model

$$K_i = \frac{P_i^{predicted} H_i^T}{H_i P_i^{predicted} H_i^T + R_i} = H_i^{-1} \frac{H_i P_i^{predicted} H_i^T}{H_i P_i^{predicted} H_i^T + R_i}$$

Uncertainty from the measurement

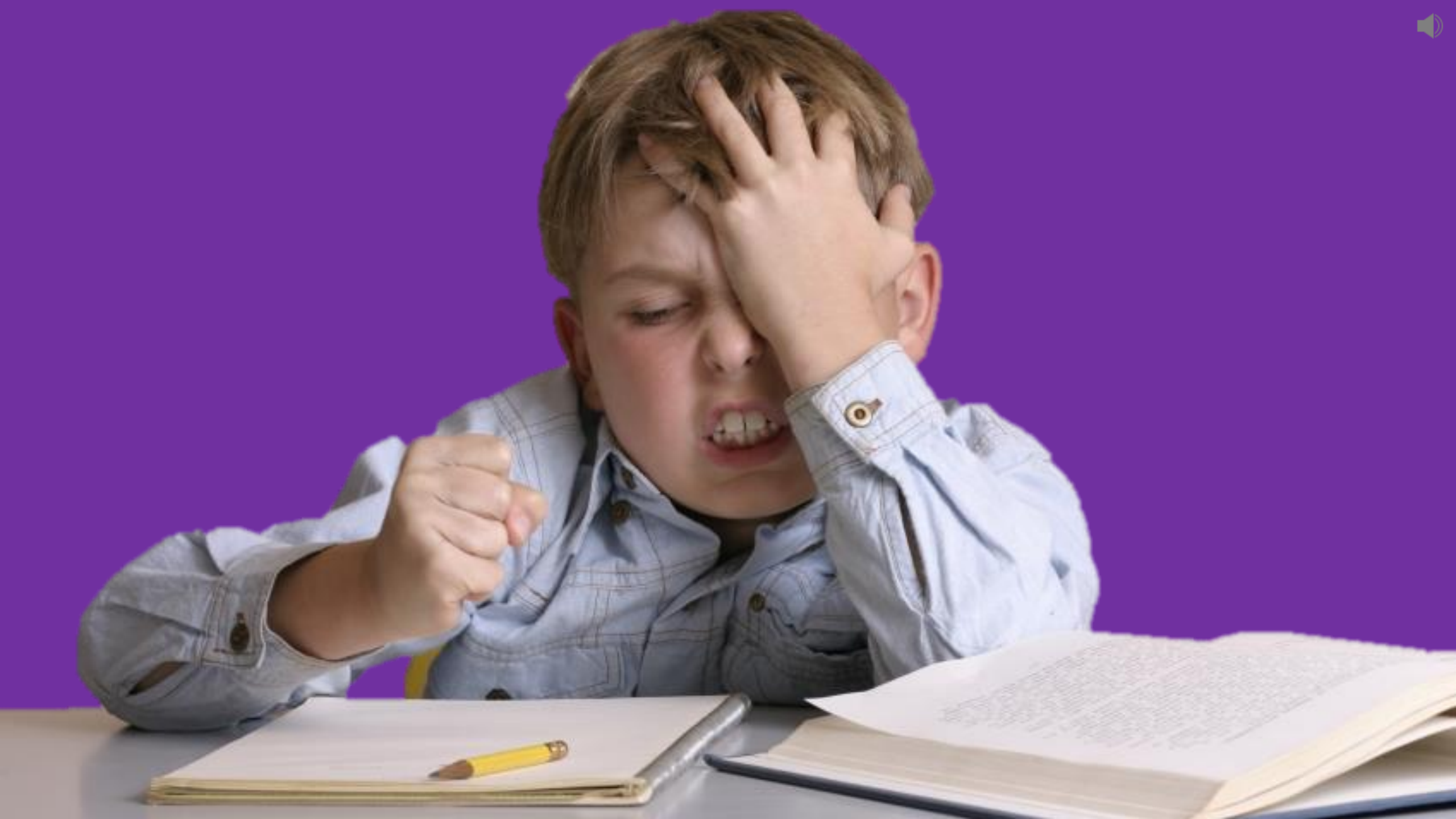
STEP 2: MEASURE

Update extrapolations with new measurements:

$$x_i^{corrected} = x_i^{predicted} + K_i\left(z_i - H_i x_i^{predicted}\right)$$

$$P_i^{corrected} = (I - K_i H_i) P_i^{predicted}$$

Identity



D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Sensor Data Processing

sensorProcessing.Rmd | data_generation.R*

Knit | Insert | Run

```
# Kalman Filter

The Kalman Filter lets you build into the filtering a model of the dynamics of the system.
```{r}
kalman <- function(path, varPos, varVel){
  dt <- 1 # time difference between measurements. In our case this is constant, but it could be variable in async systems
  H <- matrix(c(1,0,0,1,0,0,0,0),nrow=2) # Measurement matrix: simply converts form the x,y,vx,vy to x,y
  R <- matrix(c(varPos,0,0,varPos),nrow=2) # noise covariance matrix
  phi <- matrix(c(1,0,0,0,0,1,0,0,dt,0,1,0,0,dt,0,1),nrow=4) # system matrix, predicts state xi from state xi-1
  Q <- matrix(c(0,0,0,0,0,0,0,0,0,0,varVel,0,0,0,0,varVel), nrow=4) # Dynamic noise
  x <- matrix(c(path$noisyX[1],path$noisyY[1],0,0)) # initial state estimate
  P <- diag(c(varPos,varPos,varVel,varVel)) # initial estimate of state error covariance

  path$kalmanX <- rep(0,nrow(path))
  path$kalmanY <- rep(0,nrow(path))
  path$kalmanX[1] <- x[1,1]
  path$kalmanY[1] <- x[2,1]

  for (t in seq(2,nrow(path))){
    # PREDICT
```

95:1 Median Filter | R Markdown

Console | R Markdown

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/.RData]

> ?rnorm
>
```

Environment | History

Import Dataset | List

Global Environment

Environment is empty

Files | Plots | Packages | Help | Viewer

R: The Normal Distribution | Find in Topic

Normal {stats} R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
```

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Sensor Data Processing

sensorProcessing.Rmd | P | x | path | Q | phi | R | H | data_generation.R*

Knit | Insert | Run

```
121 ```{r}
122 kalman <- function(path, varPos, varVel){
123   dt <- 1 # time difference between measurements. In our case this is constant, but it could be variable in
    async systems
124   H <- matrix(c(1,0,0,1,0,0,0,0),nrow=2) # Measurement matrix: simply converts form the x,y,vx,vy to x,y
125   R <- matrix(c(varPos,0,0,varPos),nrow=2) # noise covariance matrix
126   phi <- matrix(c(1,0,0,0,0,1,0,0,dt,0,1,0,0,dt,0,1),nrow=4) # system matrix, predicts state xi from state
    xi-1
127   Q <- matrix(c(0,0,0,0,0,0,0,0,0,0,varVel,0,0,0,0,varVel), nrow=4) # Dynamic noise
128   x <- matrix(c(path$noisyX[1],path$noisyY[1],0,0)) # initial state estimate
129   P <- diag(c(varPos,varPos,varVel,varVel)) # initial estimate of state error covariance
130
131   path$kalmanX <- rep(0,nrow(path))
132   path$kalmanY <- rep(0,nrow(path))
133   path$kalmanX[1] <- x[1,1]
134   path$kalmanY[1] <- x[2,1]
135
136   for (t in seq(2,nrow(path))){
137     # PREDICT
138
139     # Extrapolate the state
140     predictedX <- phi %*% x
```

134:28 | kalman(path, varPos, varVel) | R Markdown

Console | R Markdown

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/

```
Error in matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, varVel, 0, 0, 0, 0, varVel),  :
  object 'varVel' not found
> varVel <- 1e-3
>   Q <- matrix(c(0,0,0,0,0,0,0,0,0,0,varVel,0,0,0,0,varVel), nrow=4) # Dynamic noise
> View(Q)
>   x <- matrix(c(path$noisyX[1],path$noisyY[1],0,0)) # initial state estimate
Error in matrix(c(path$noisyX[1], path$noisyY[1], 0, 0)) :
  object 'path' not found
> path <- read.csv('path.csv')
>   x <- matrix(c(path$noisyX[1],path$noisyY[1],0,0)) # initial state estimate
> View(x)
> View(path)
> P <- diag(c(varPos,varPos,varVel,varVel))
> View(P)
>
```

Environment | History

Import Dataset | List

Global Environment

Data

H	num [1:2, 1:4] 1 0 0 1 0 0 0 0
P	num [1:4, 1:4] 9 0 0 0 0 9 0 0 0 0 ...
path	1232 obs. of 6 variables
phi	num [1:4, 1:4] 1 0 0 0 0 1 0 0 1 0 ...
Q	num [1:4, 1:4] 0 0 0 0 0 0 0 0 0 0 ...
R	num [1:2, 1:2] 9 0 0 9
x	num [1:4, 1] 23.8 65.1 0 0

Values

dt	1
varPos	9
varVel	0.001

Files | Plots | Packages | Help | Viewer

R: The Normal Distribution | Find in Topic

Normal {stats} R Documentation

The Normal Distribution

Description

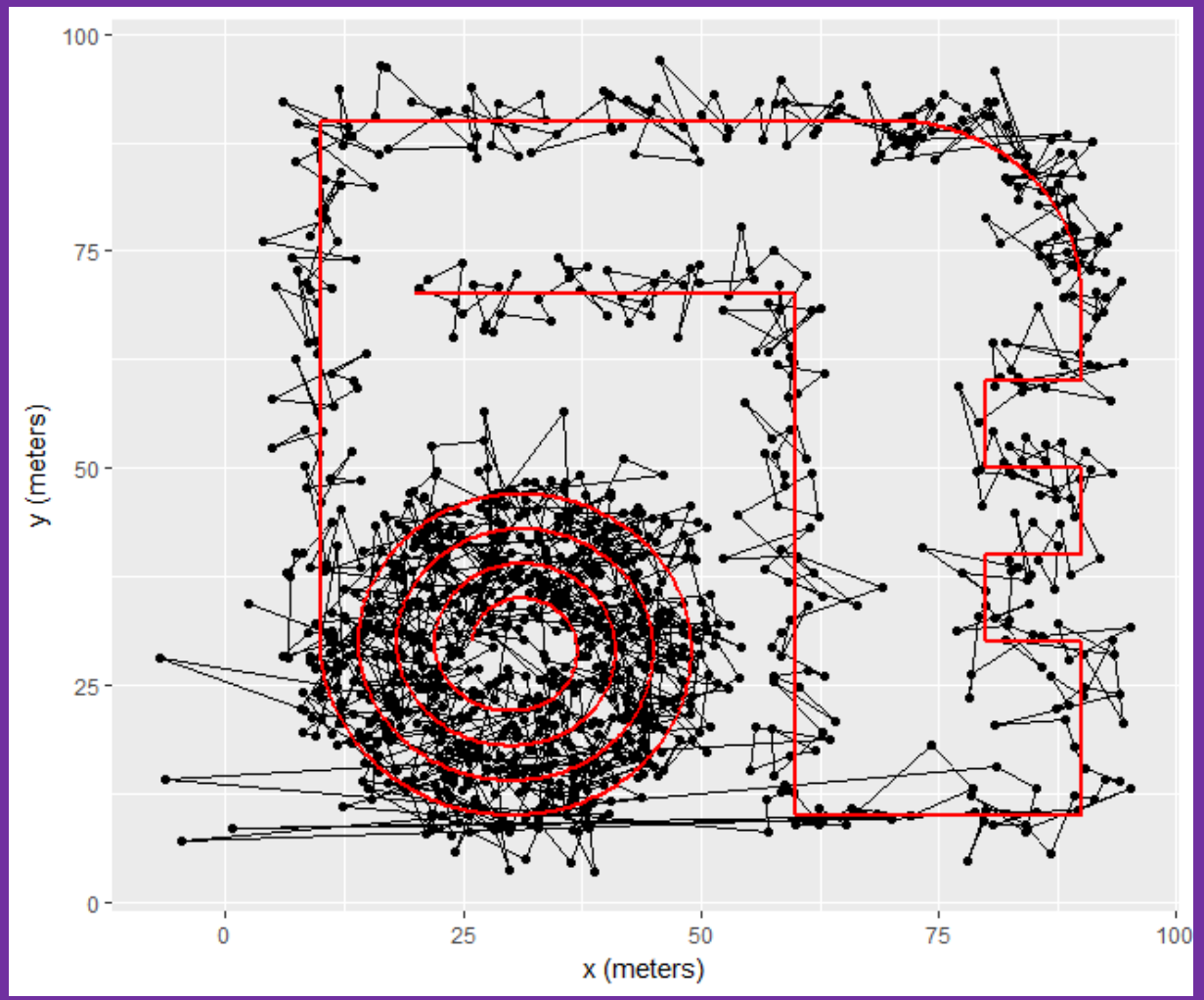
Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

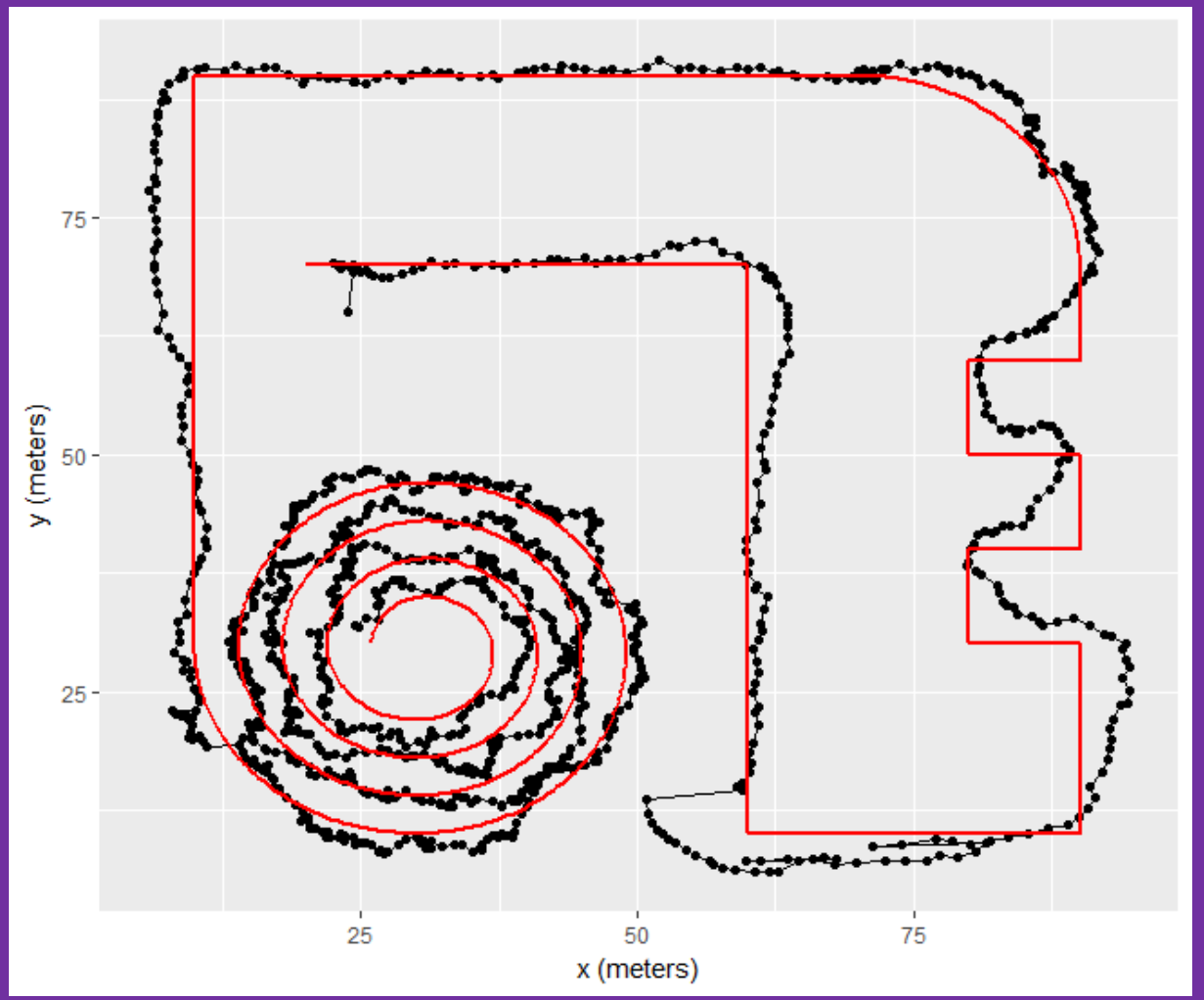
Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).


100
75
50
25
0
y (meters)
0
25
50
75
100
x (meters)


75
50
25
y (meters)
25
50
75
x (meters)

KALMAN FILTER

+ *Dynamic model of the system*
+ *No lag*
+ *Tunable trade-off between model and measurement*
+ *Uncertainty estimate*
− *Parameters are not intuitive*
− *Overshooting*

KALMAN

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayesian

Gaussian

Linear

Online

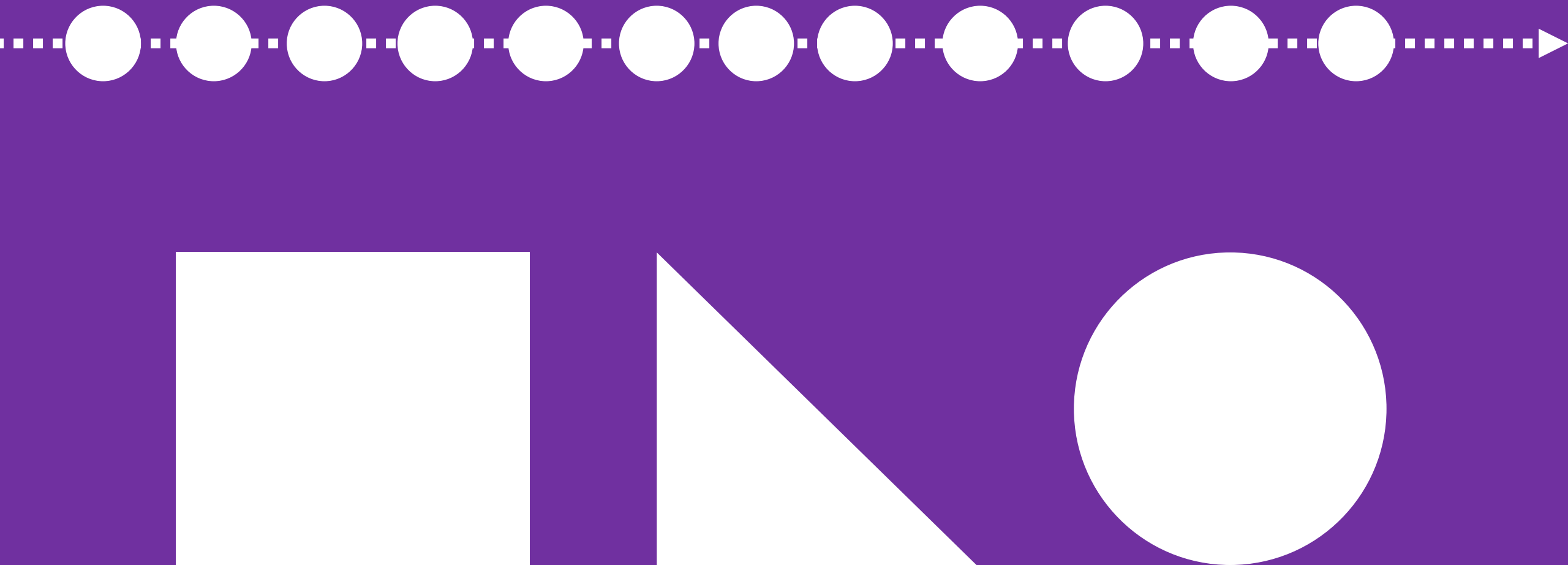
What if this is not the case?

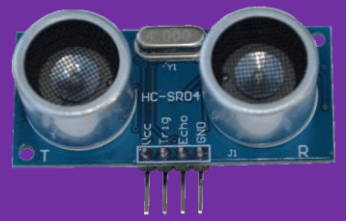
Kalman

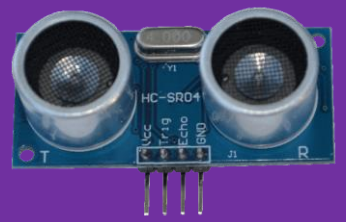
Particle

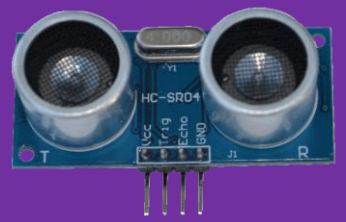
KALMAN	PARTICLE
Bayesian	*Bayesian*
Gaussian	*Gaussian or not*
Linear	*Linear or not*
Online	*Online*

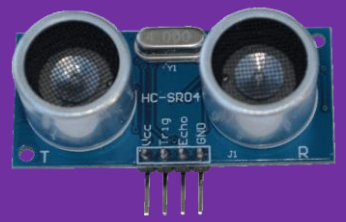

CLEAN

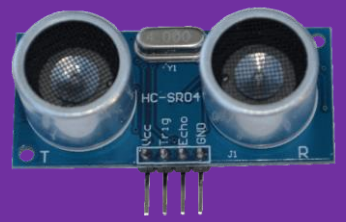


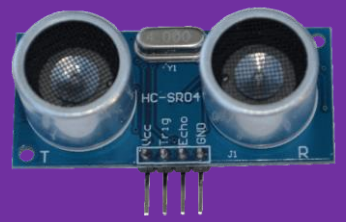

HC-SR04
Vcc
Trig
Echo
GND
T
R

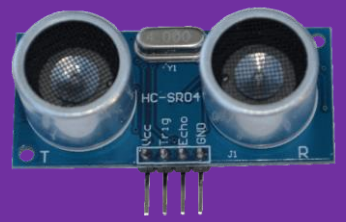

HC-SR04
Vcc
Trig
Echo
GND
T
R
J1

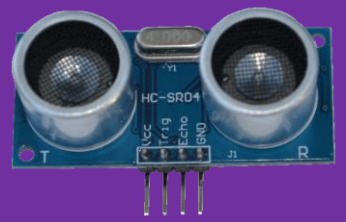

HC-SR04
Vcc
Trig
Echo
GND
T
R
J1
Y1

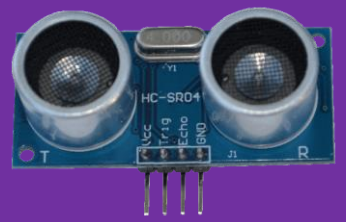

HC-SR04
Vcc
Trig
Echo
GND
J1
T
R


HC-SR04
Vcc
Trig
Echo
GND
T
R

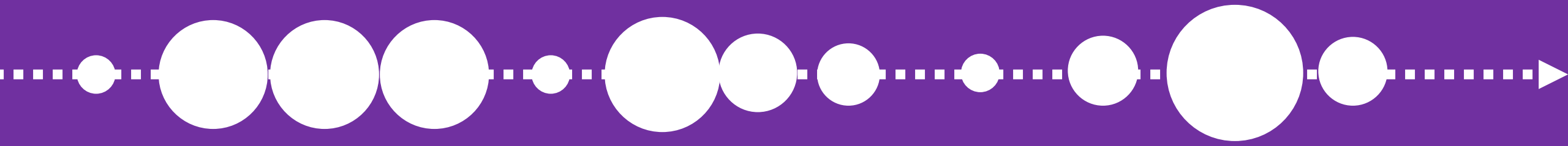

HC-SR04
Vcc
Trig
Echo
GND
T
R


HC-SR04
Vcc
Trig
Echo
GND
T
R


HC-SR04
Vcc
Trig
Echo
GND
T
R

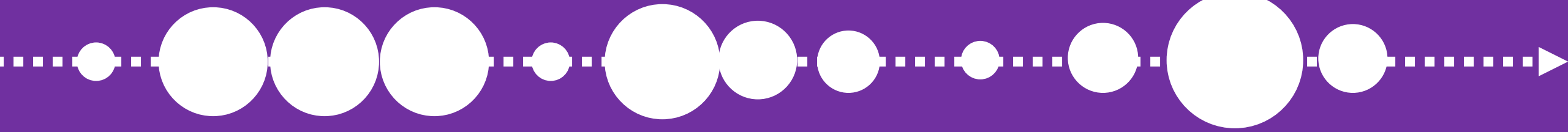

HC-SR04
Vcc
Trig
Echo
GND
J1
T
R

Importance Weights



Importance Weights

Resample



Importance Weights

Resample

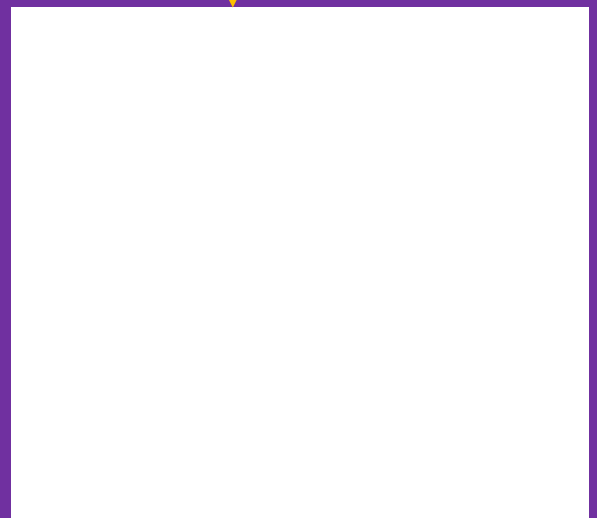
The robot moves

The robot moves

The robot moves

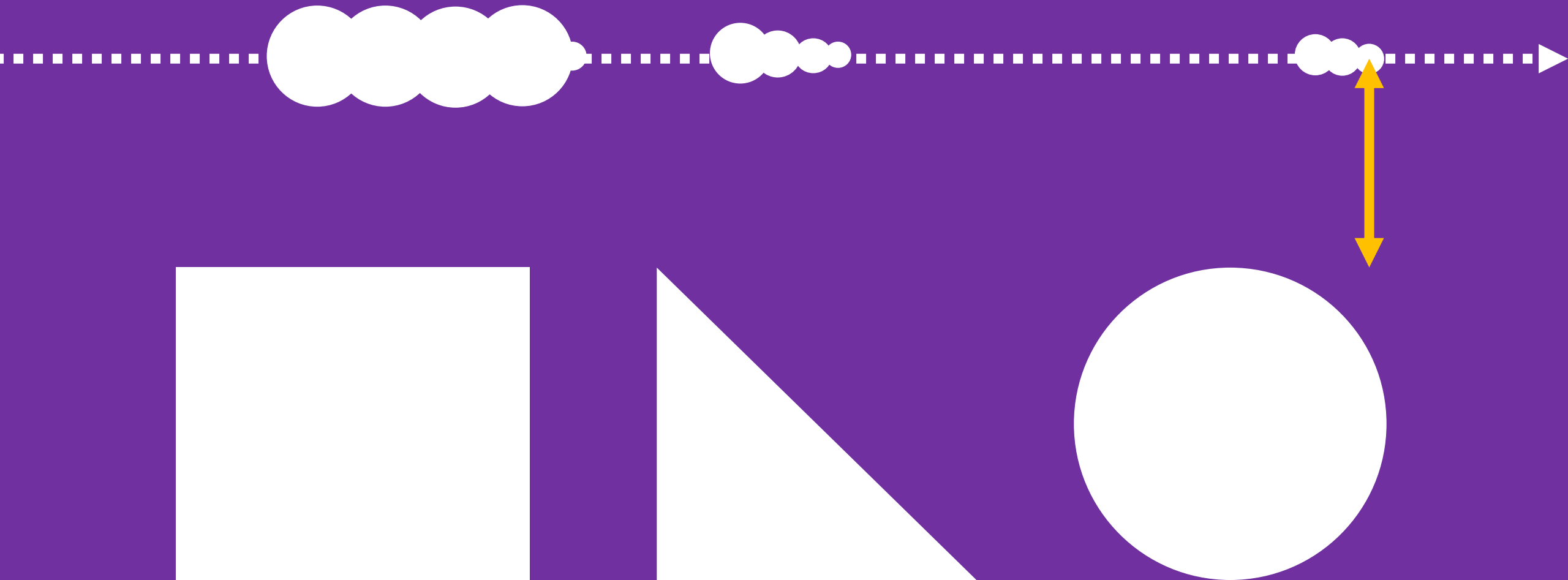
The robot moves

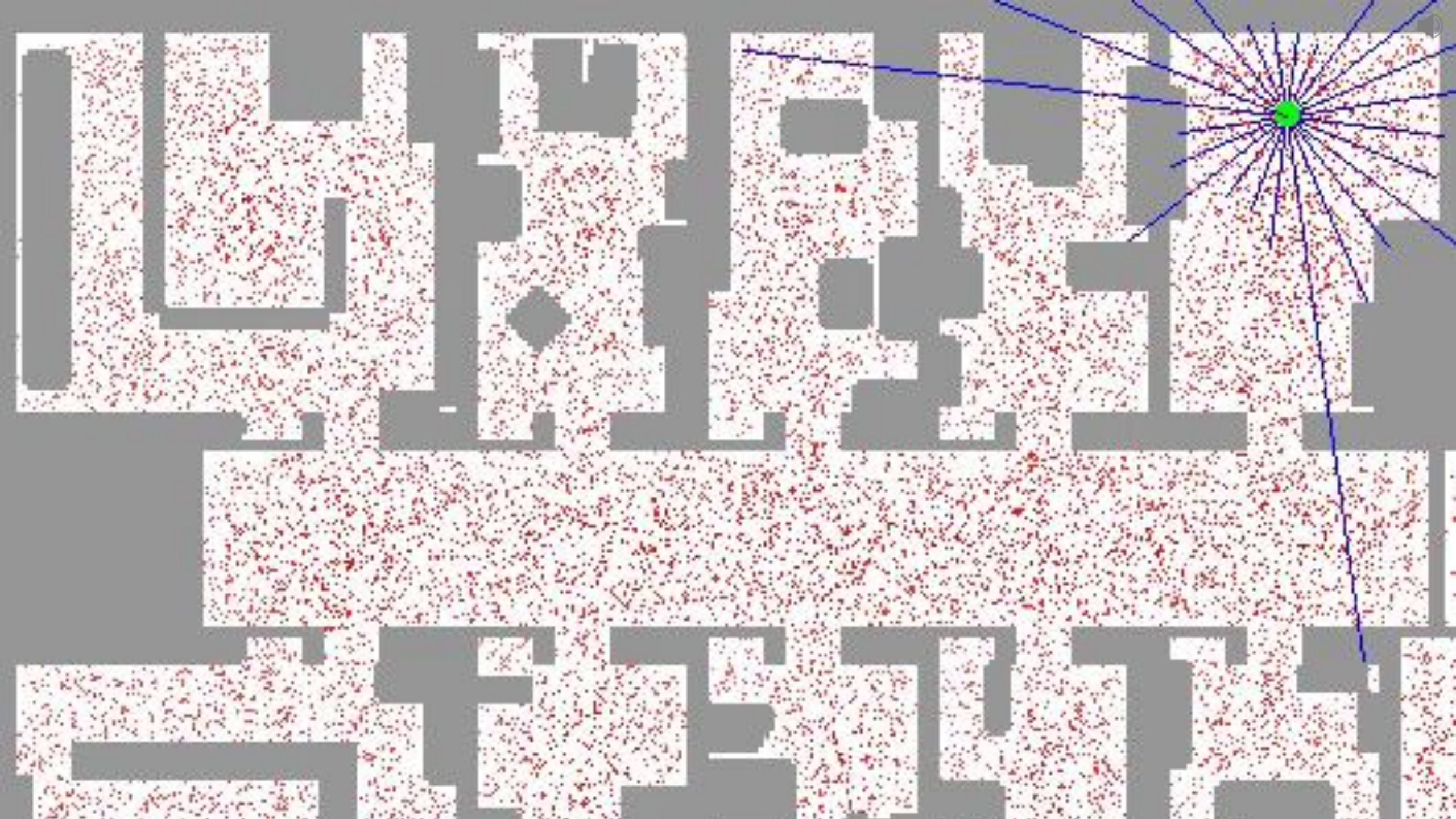
The robot moves



The robot moves

Rinse and Repeat!





Generate Hypotheses

Compute Weights

Resample

measurement

$$z_i = \begin{pmatrix} z_i^{(x)} \\ z_i^{(y)} \end{pmatrix}$$

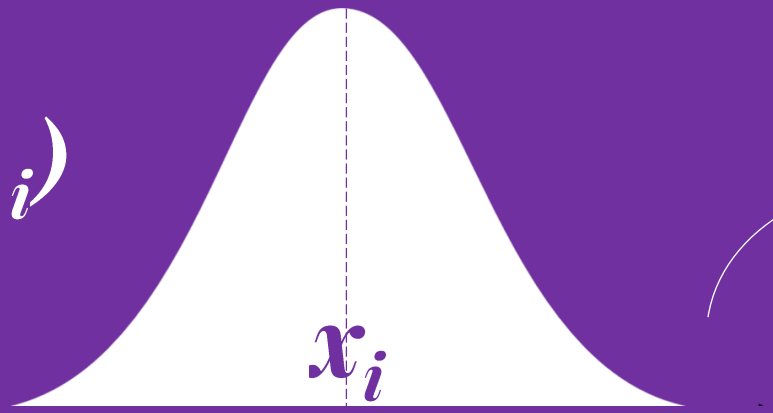
state

$$x_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

could also contain more vars

Probability distribution of measurement given actual value

$p(z_i|x_i)$



We use Gaussian, but it can be other

Probabilistic Dynamics

Again, we use Gaussian, but it can be other

TRACKING EXAMPLE

Measurements

State

$p(z_i|x_i)$

$p(x_i|x_{i-1})$

How would you model the tracking example in a particle filter? What would the measurement and state vectors be? Which weighting function would you use for assigning resampling weights? Which function would you use for modelling the dynamics?

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Sensor Data Processing

sensorProcessing.Rmd | z | particleFilerFromStackOverflow.R | P | x | path | Q | phi | R | H | data_generation.R*

Knit | Insert | Run

```
173 In a particle filter, each particle corresponds to a random hypothesis for the trajectory. We use the
    likelihood as a weight for how good is this hypothesis and resample. Each resampling step improves our
    guess. Note how the number of particles strongly influences the computation time and the quality of the
    output.
174
175 ```{r, warning=FALSE}
176 # Begin filter
177 N <- 200 # number of particles
178 tau <- nrow(path) # number of samples
179 sdPos = 3 # standard deviation of the position estimation
180 x.pf <- array(rep(NA, tau * N * 2), c(tau, 2, N)) # 3D array that stores the particles
181 z <- matrix(c(path$noisyX[1], path$noisyY[1])) # measurement matrix
182
183 # 1. Initialize 3D array N particles x 2 coordinates x tau points in time
184 x.pf[1,1,] <- array(rnorm(N, z[1], sdPos))
185 x.pf[1,2,] <- array(rnorm(N, z[2], sdPos))
186
187 for (t in 2:tau) {
188   # 2. Importance sampling step
189   # we don't know where we are going so we take random guesses
190   x.pf[t,, ] <- x.pf[t-1,,] + rnorm(N,mean = 0, sd = 3)
191
```

182:1 | Chunk 11 | R Markdown

Console | R Markdown

D:/Dropbox/Projects/2017.2 - Mobile Computing/Teaching/2017/Sensor Data Processing/

```
+   path$particleY[t] <- mean(x.pf[t,2,])
+
+   # 3. Resampling step
+   # use the importance weights to sample good estimates more often
+   s <- sample(1:N, size=N, replace=TRUE, prob=w)
+   x.pf <- x.pf[,, s]
+ }
> ggplot(path, aes(x=particleX, y=particleY))+geom_path()+geom_point()+geom_path(aes(x=x,y=y),col='red',size=1)+xl
ab('x (meters)')+ylab('y (meters)')+ggtitle(paste('Particle Filter with', N, 'particles'))
>
> N <- 200 # number of particles
> tau <- nrow(path) # number of samples
> sdPos = 3 # standard deviation of the position estimation
> x.pf <- array(rep(NA, tau * N * 2), c(tau, 2, N)) # 3D array that stores the particles
>
```

Environment | History

Import Dataset | List

Global Environment

s	int [1:1000] 140 362 372 689 881 778…
sdPos	3
t	1232L
tau	1232L
varPos	9
varVel	0.001
w	num [1:1000] 7.20e-11 3.25e-04 8.75e…
w.tilde	num [1:1000] 1.69e-09 7.60e-03 2.05e…
w.tildeX	num [1:1000] 7.98e-10 1.52e-02 1.29e…
w.tildeY	num [1:1000] 2.57e-09 1.88e-09 4.10e…
x.pf	Large array (492800 elements, 1.9 Mb)

Files | Plots | Packages | Help | Viewer

Zoom | Export

PARTICLE FILTER

+ *General*

+ *Continuous or Discrete vars*

+ *Great results*

− *Lots of memory*

− *Very slow*

Challenges in sensor data

Particle Filter

RECAP

Mean Filter

Kalman Filter

Median Filter

R implementations

Weighted Mean Filter

Select font size T T T

How did you feel about this lecture?

Allow Single Choice Only Allow Multiple Choices

Loved it!

Liked it

Neither liked it nor disliked it

Disliked it

Hated it!

+ Add another answer

Preview

Select font size T T T

What did you like and dislike about this lecture? How can I improve it?

Preview