

Lab1-torontodatavis-Asude

February 6, 2024

```
[1]: mydata = c(3,8,2,1)
```

```
[2]: mydata
```

```
1. 3 2. 8 3. 2 4. 1
```

```
[4]: library(tidyverse)
```

```
[5]: library(repr)
      library(viridis)
```

```
Loading required package: viridisLite
```

```
[6]: library(lubridate)
      library(zoo)
```

```
Attaching package: 'lubridate'
```

```
The following objects are masked from 'package:base':
```

```
date, intersect, setdiff, union
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':
```

```
as.Date, as.Date.numeric
```

```
[7]: getwd()
```

```
'/home/jovyan/work'
```

```
[8]: plot_1.df <- read_tsv("Lab1_scatterplot_set_1.txt")
```

Parsed with column specification:

```
cols(  
  dataset = col_character(),  
  x_values = col_double(),  
  y_values = col_double()  
)
```

```
[33]: x <- read_tsv("Lab1_scatterplot_set_1.txt")
```

Parsed with column specification:

```
cols(  
  dataset = col_character(),  
  x_values = col_double(),  
  y_values = col_double()  
)
```

```
[35]: x <- c(1,2,3,4)
```

```
[36]: stdev(x)
```

Error in stdev(x): could not find function "stdev"
Traceback:

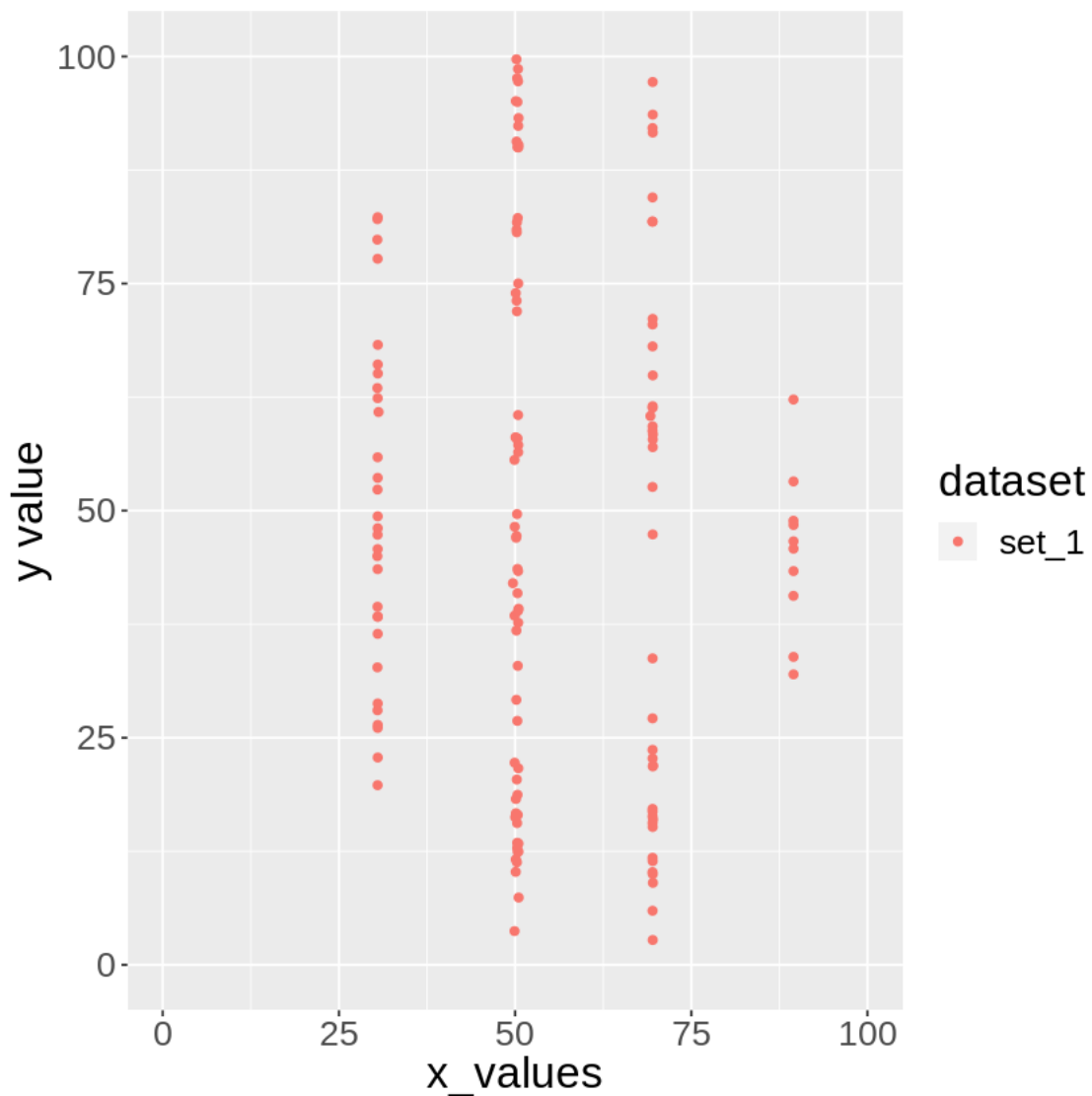
```
[9]: str(plot_1.df)
```

```
spc_tbl_ [142 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)  
 $ dataset : chr [1:142] "set_1" "set_1" "set_1" "set_1" ...  
 $ x_values: num [1:142] 50.5 50.3 50.2 50.3 50.5 ...  
 $ y_values: num [1:142] 93.2 97.6 99.7 90 90 ...  
 - attr(*, "spec")=  
 .. cols(  
 ..   dataset = col_character(),  
 ..   x_values = col_double(),  
 ..   y_values = col_double()  
 .. )
```

```
[ ]:
```

```
[10]: plot_1.plot <- ggplot(plot_1.df)
```

```
[12]: plot_1.plot <- plot_1.plot +
aes(x = x_values, y = y_values, colour = dataset) + geom_point() +
theme(text = element_text(size = 20)) + # set text size guides(colour =
  ↳ guide_legend(title="Set 1")) + # Legend title xlab("x value") + # Set the
  ↳ x-axis label
ylab("y value") + # Set the y-axis label
xlim(0,100) + # specify the minimum and maximum x axis values
ylim(0,100) # ditto for the y axis
#display our plot
plot_1.plot
```



```
[13]: plot_2.df <- read_tsv("Lab1_scatterplot_set_2.txt")
```

```

Parsed with column specification:
cols(
  dataset = col_character(),
  x_values = col_double(),
  y_values = col_double()
)

```

```
[15]: str(plot_2.df)
```

```

spc_tbl_ [142 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ dataset : chr [1:142] "set_2" "set_2" "set_2" "set_2" ...
 $ x_values: num [1:142] 32.3 53.4 63.9 70.3 34.1 ...
 $ y_values: num [1:142] 61.4 26.2 30.8 82.5 45.7 ...
 - attr(*, "spec")=
 .. cols(
 ..   dataset = col_character(),
 ..   x_values = col_double(),
 ..   y_values = col_double()
 .. )

```

```
[16]: plot_2.plot <- ggplot(plot_2.df)
```

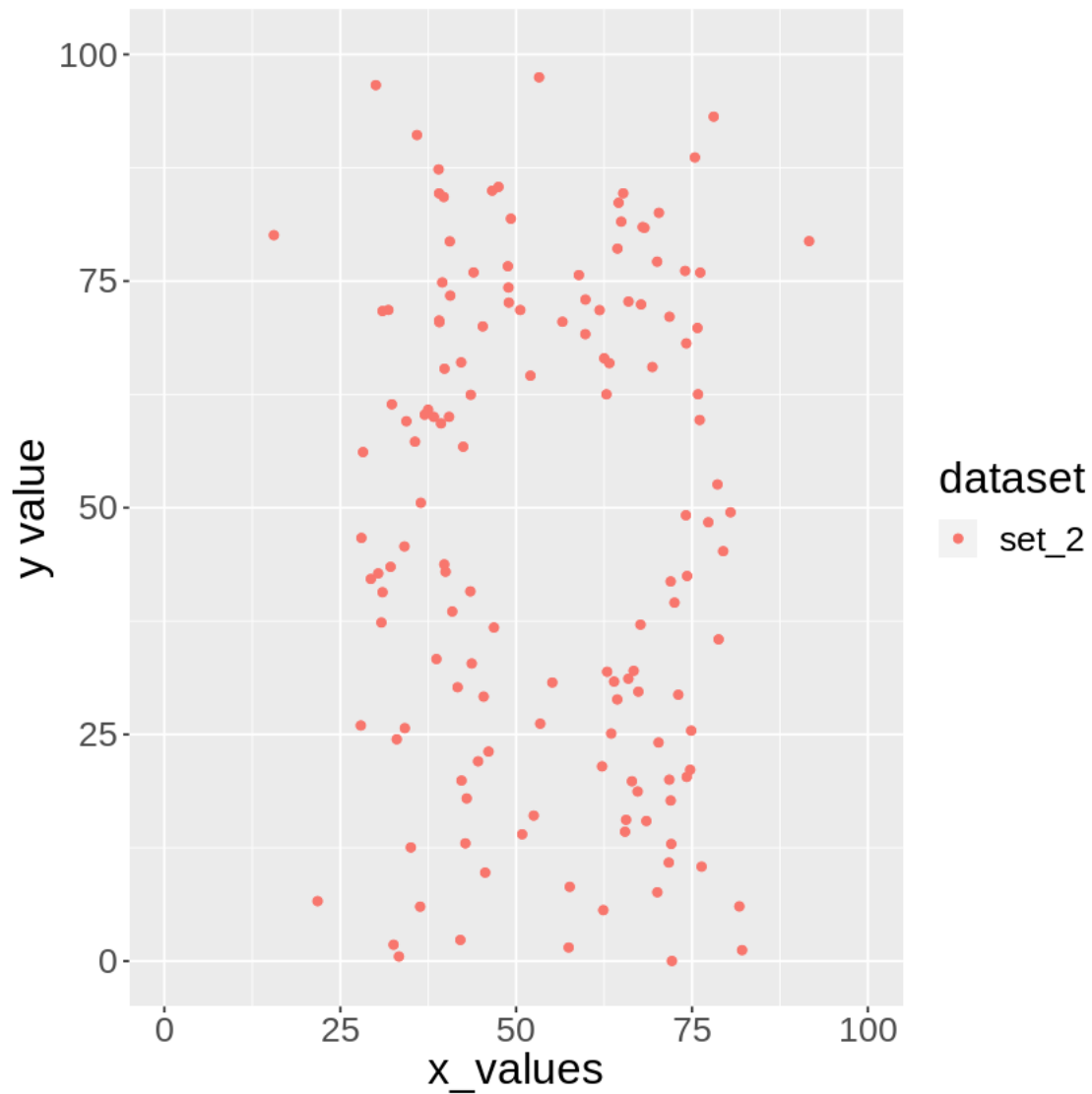
```

[18]: plot_2.plot <- plot_2.plot +
aes(x = x_values, y = y_values, colour = dataset) + geom_point() +
theme(text = element_text(size = 20)) + # set text size guides(colour =
  ↳guide_legend(title="Set 2")) + # Legend title xlab("x value") + # Set the
  ↳x-axis label
ylab("y value") + # Set the y-axis label
xlim(0,100) + # specify the minimum and maximum x axis values
ylim(0,100) # ditto for the y axis
#display our plot
plot_2.plot

```

Scale for 'x' is already present. Adding another scale for 'x', which will replace the existing scale.

Scale for 'y' is already present. Adding another scale for 'y', which will replace the existing scale.



```
[19]: plot_3.df <- read_tsv("Lab1_scatterplot_set_3.txt")
```

```
Parsed with column specification:
cols(
  dataset = col_character(),
  x_values = col_double(),
  y_values = col_double()
)
```

```
[20]: str(plot_3.df)
```

```
spc_tbl_ [142 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
```

```

$ dataset : chr [1:142] "set_3" "set_3" "set_3" "set_3" ...
$ x_values: num [1:142] 53.4 52.8 47.1 42.4 42.7 ...
$ y_values: num [1:142] 90.2 90.1 90.5 89.5 90.4 ...
- attr(*, "spec")=
  .. cols(
  ..   dataset = col_character(),
  ..   x_values = col_double(),
  ..   y_values = col_double()
  .. )

```

```
[21]: plot_3.plot <- ggplot(plot_3.df)
```

```

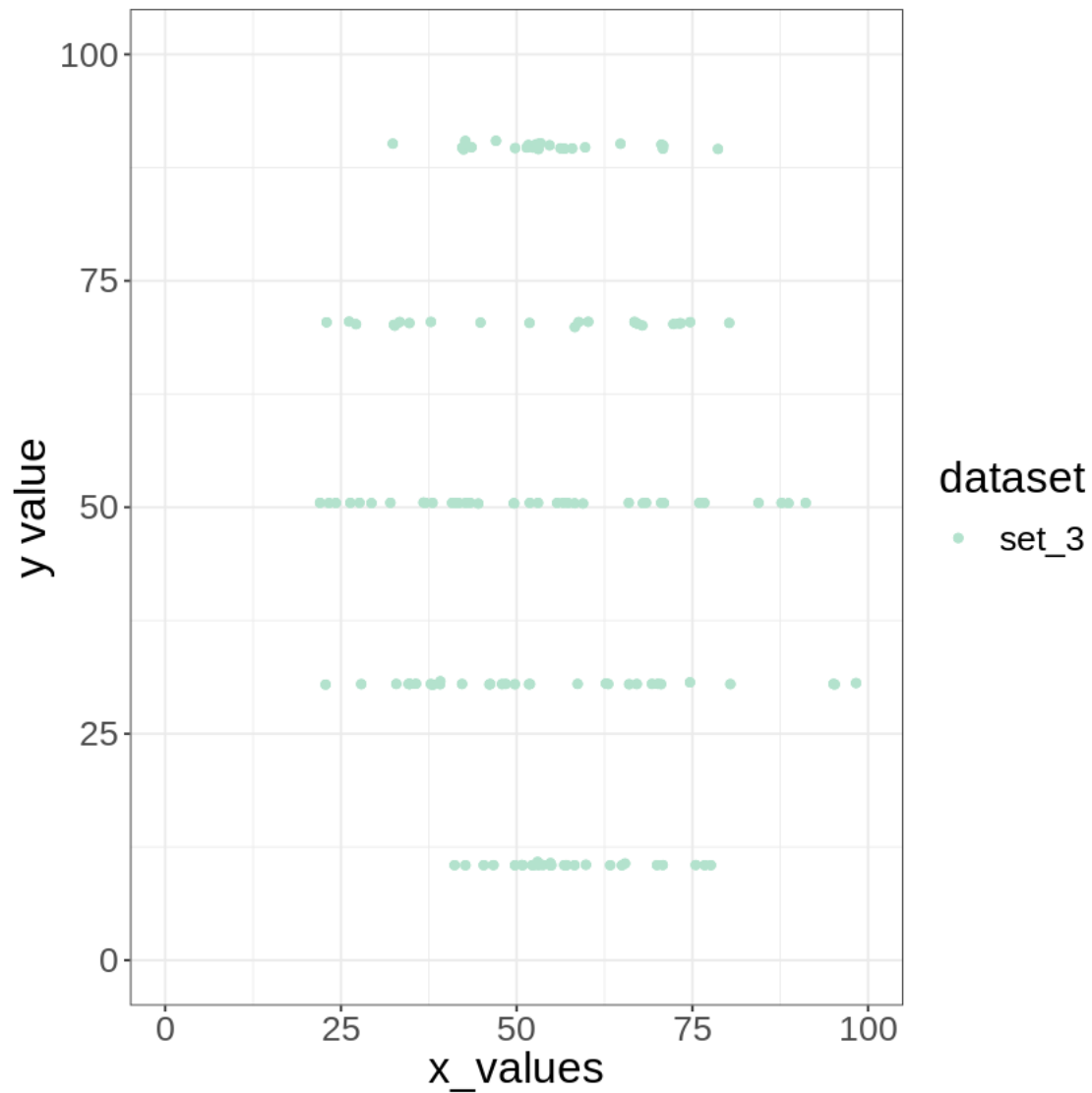
[24]: plot_3.plot <- plot_3.plot +
aes(x = x_values, y = y_values, colour = dataset) +
geom_point() +
theme_bw() +
theme(text = element_text(size = 20)) + # set text size guides(colour =
  ↳ guide_legend(title="Set 3")) + # Legend title xlab("x value") + # Set the
  ↳ x-axis label
ylab("y value") + # Set the y-axis label
xlim(0,100) + # specify the minimum and maximum x axis values
ylim(0,100) # ditto for the y axis
plot_3.plot <- plot_3.plot + scale_colour_brewer(palette = "Pastel2")
#display our plot
plot_3.plot

```

Scale for 'x' is already present. Adding another scale for 'x', which will replace the existing scale.

Scale for 'y' is already present. Adding another scale for 'y', which will replace the existing scale.

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



```
[25]: plot_4.df <- read_tsv("Lab1_scatterplot_set_4.txt")
```

```
Parsed with column specification:
cols(
  dataset = col_character(),
  x_values = col_double(),
  y_values = col_double()
)
```

```
[26]: str(plot_4.df)
```

```
spc_tbl_ [142 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
```

```

$ dataset : chr [1:142] "set_4" "set_4" "set_4" "set_4" ...
$ x_values: num [1:142] 55.4 51.5 46.2 42.8 40.8 ...
$ y_values: num [1:142] 97.2 96 94.5 91.4 88.3 ...
- attr(*, "spec")=
  .. cols(
  ..   dataset = col_character(),
  ..   x_values = col_double(),
  ..   y_values = col_double()
  .. )

```

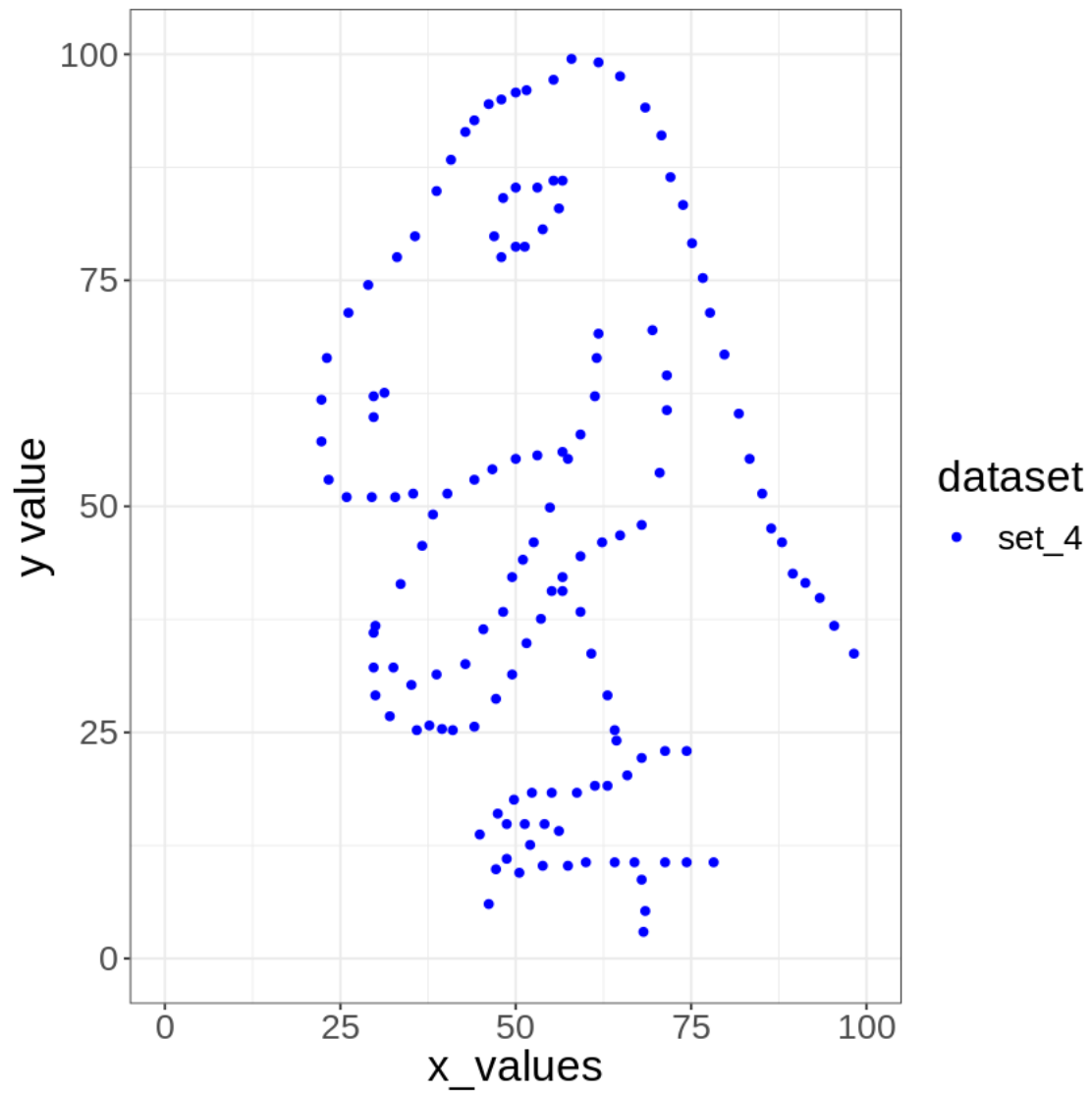
```
[27]: plot_4.plot <- ggplot(plot_4.df)
```

```

[28]: plot_4.plot <- plot_4.plot +
aes(x = x_values, y = y_values, colour = dataset) +
geom_point() +
theme_bw() +
theme(text = element_text(size = 20)) + # set text size guides(colour =
  ↳guide_legend(title="Set 4")) + # Legend title xlab("x value") + # Set the
  ↳x-axis label
ylab("y value") + # Set the y-axis label
xlim(0,100) + # specify the minimum and maximum x axis values
ylim(0,100) # ditto for the y axis

plot_4.plot <- plot_4.plot + scale_colour_manual(values= c("blue1", "blue2",
  ↳"blue3", "blue4"))
#display our plot
plot_4.plot

```

```
[32]: ggsave(plot = plot_4.plot, filename = "Set_4.png", scale=2, device = "png",  
↪units = c("cm"))
```

Saving 33.9 x 33.9 cm image

```
[ ]:
```