

# Anscombe's Quartet

Anscombe's Quartet Example Ref.: [Anscombe's Quartet Wiki](#)

## Julia Program to Visualize Anscombe's Quartet

### Creating a dataframe and reading data from a csv file

*the pipe symbol in Julia `|>` is used to send the output of one function to the input of another, in this case the data read from the CSV file by the CSV library will be sent to the DataFrame library.*

```
• using DataFrames, CSV
```

df =	x1	x2	x3	x4	y1	y2	y3	y4
<b>1</b>	10	10	10	8	8.04	9.14	7.46	6.58
<b>2</b>	8	8	8	8	6.95	8.14	6.77	5.76
<b>3</b>	13	13	13	8	7.58	8.74	12.74	7.71
<b>4</b>	9	9	9	8	8.81	8.77	7.11	8.84
<b>5</b>	11	11	11	8	8.33	9.26	7.81	8.47
<b>6</b>	14	14	14	8	9.96	8.1	8.84	7.04
<b>7</b>	6	6	6	8	7.24	6.13	6.08	5.25
<b>8</b>	4	4	4	19	4.26	3.1	5.39	12.5
<b>9</b>	12	12	12	8	10.84	9.13	8.15	5.56
<b>10</b>	7	7	7	8	4.82	7.26	6.42	7.91
<b>11</b>	5	5	5	8	5.68	4.74	5.73	6.89

```
• df = CSV.File("anscombe.csv") |> DataFrame
```

Find global extrema for rows and columns to use for plot x and y limits

first select the 4 columns in *df* with *x* values and convert them to a matrix

then input that matrix to the `extrema()` function to find *x* max and *x* min values

for better plot limits, subtract 1 from the minimum and add one to the maximum

`Plot()` expects the limits as an array so convert them to one using `collect()` function

repeat above steps for the *y* limits

```
xlim = [3, 20]
```

- `xlim = collect(extrema(Matrix(select(df,1:4))) .+ (-1, 1))`

```
ylim = [2.1, 13.74]
```

- `ylim = collect(extrema(Matrix(select(df,5:8))) .+ (-1, 1))`

## Using `Plots()` library to plot the quartet of graphs

- `using Plots`

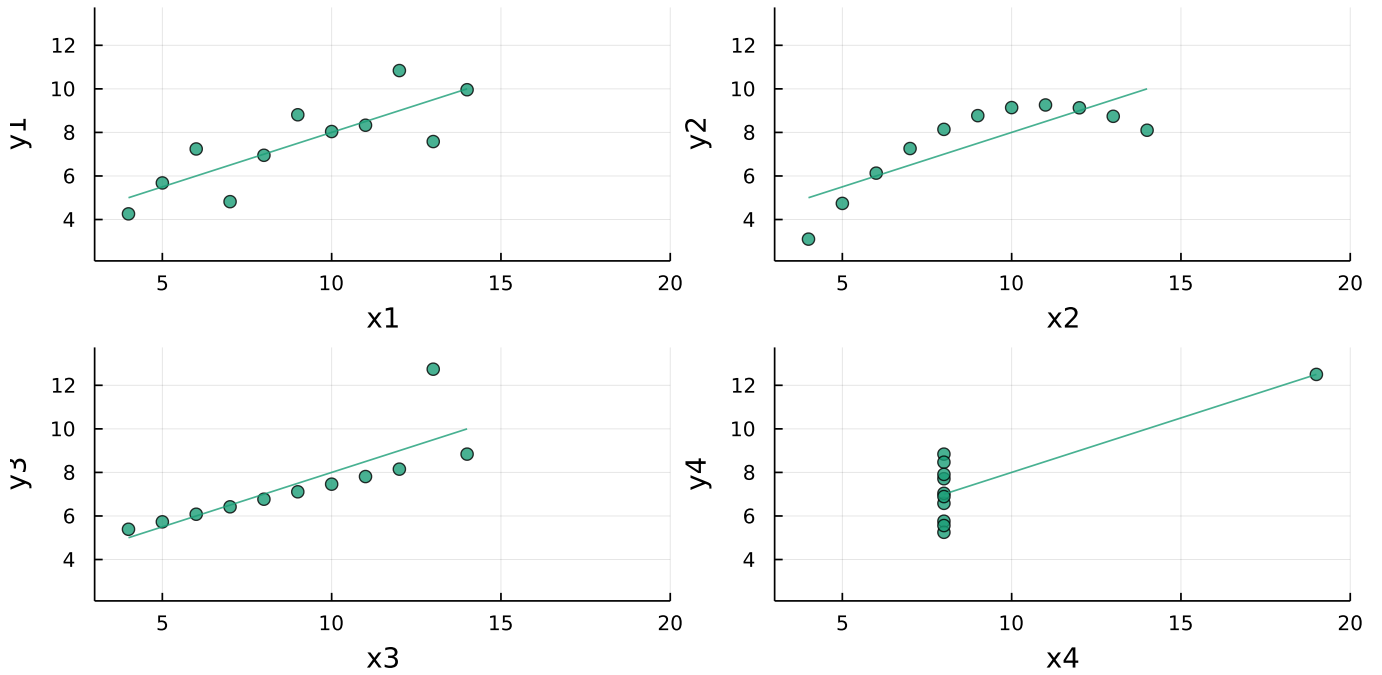
```
plot_array = []
```

- `# create general array to hold the 4 arrays to be plotted`
- `plot_array = Any[]`

`DataType`

- `typeof(Any)`

- `for i in 1:4`
- `x_vals = df[:,i] # select all x values in col 1:4`
- `y_vals = df[:,i+4] # select all y values in col 5:8`
- `p = plot(x_vals, y_vals,`
- `seriestype = :scatter,`
- `xlabel = "x$i", ylabel = "y$i",`
- `xlims = xlim, ylims = ylim,`
- `smooth = true, # adds regression line`
- `opacity = 0.8, palette = :Dark2_5,`
- `)`
- `push!(plot_array, p)`
- `end`



- *# the dot dot dot is the splatting operator to pass the 4 arrays to Plot() separately*
- `plot(plot_array..., layout = (2,2), legend = false, size = (800,400))`

### Basic descriptive statistics - misleading!!

	variable	mean	std	min	q25	median	q75	max	nunique	nmissir
1	:x1	9.0	3.31662	4	6.5	9.0	11.5	14	nothing	0
2	:x2	9.0	3.31662	4	6.5	9.0	11.5	14	nothing	0
3	:x3	9.0	3.31662	4	6.5	9.0	11.5	14	nothing	0
4	:x4	9.0	3.31662	8	8.0	8.0	8.0	19	nothing	0
5	:y1	7.50091	2.03157	4.26	6.315	7.58	8.57	10.84	nothing	0
6	:y2	7.50091	2.03166	3.1	6.695	8.14	8.95	9.26	nothing	0
7	:y3	7.5	2.03042	5.39	6.25	7.11	7.98	12.74	nothing	0
8	:y4	7.50091	2.03058	5.25	6.17	7.04	8.19	12.5	nothing	0

- `describe(df, :detailed)` *# the colon indicates a symbol, here used to pass a parameter*