

# Quarteto de Anscombe

Iniciamos importando as bibliotecas necessárias:

In [ ]:

```
# Import lybraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Vega lite for interactive plots
import altair as alt
from vega_datasets import data
import altair.vegalite.v3 as v3
```

Depois disso, definimos o conjunto de dados para a visualização em um **Dataframe** do **Pandas** (os dados já são conhecidos):

In [ ]:

```
# Set data of the anscombe quartet
data = pd.DataFrame({"x1": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
                    "y1": [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68],
                    "x2": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
                    "y2": [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74],
                    "x3": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
                    "y3": [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73],
                    "x4": [8, 8, 8, 8, 8, 8, 19, 8, 8, 8, 8],
                    "y4": [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89]})

data
```

Out[ ]:

	x1	y1	x2	y2	x3	y3	x4	y4
0	10	8.04	10	9.14	10	7.46	8	6.58
1	8	6.95	8	8.14	8	6.77	8	5.76
2	13	7.58	13	8.74	13	12.74	8	7.71
3	9	8.81	9	8.77	9	7.11	8	8.84
4	11	8.33	11	9.26	11	7.81	8	8.47
5	14	9.96	14	8.10	14	8.84	8	7.04
6	6	7.24	6	6.13	6	6.08	8	5.25
7	4	4.26	4	3.10	4	5.39	19	12.50
8	12	10.84	12	9.13	12	8.15	8	5.56
9	7	4.82	7	7.26	7	6.42	8	7.91
10	5	5.68	5	4.74	5	5.73	8	6.89

Com isso, podemos fazer os gráficos (objetos) utilizando o Altair:

In [ ]:

```
# Plot charts with altair and vegalite
chart = []
for i in range(1, 5):
    chart.append(
        alt.Chart(
            data
        ).mark_circle(
            size=60
        ).encode(
            alt.X(f"x{i}"),
            scale=alt.Scale(domain=(0, 20))),
            alt.Y(f"y{i}"),
            scale=alt.Scale(domain=(0,15))),
            tooltip=[f"x{i}", f"y{i}"]
        ).properties(
            title=f"Dataset {i}"
        ).interactive()
    )
```

Configurando os gráficos e plotando para ver o resultado, temos:

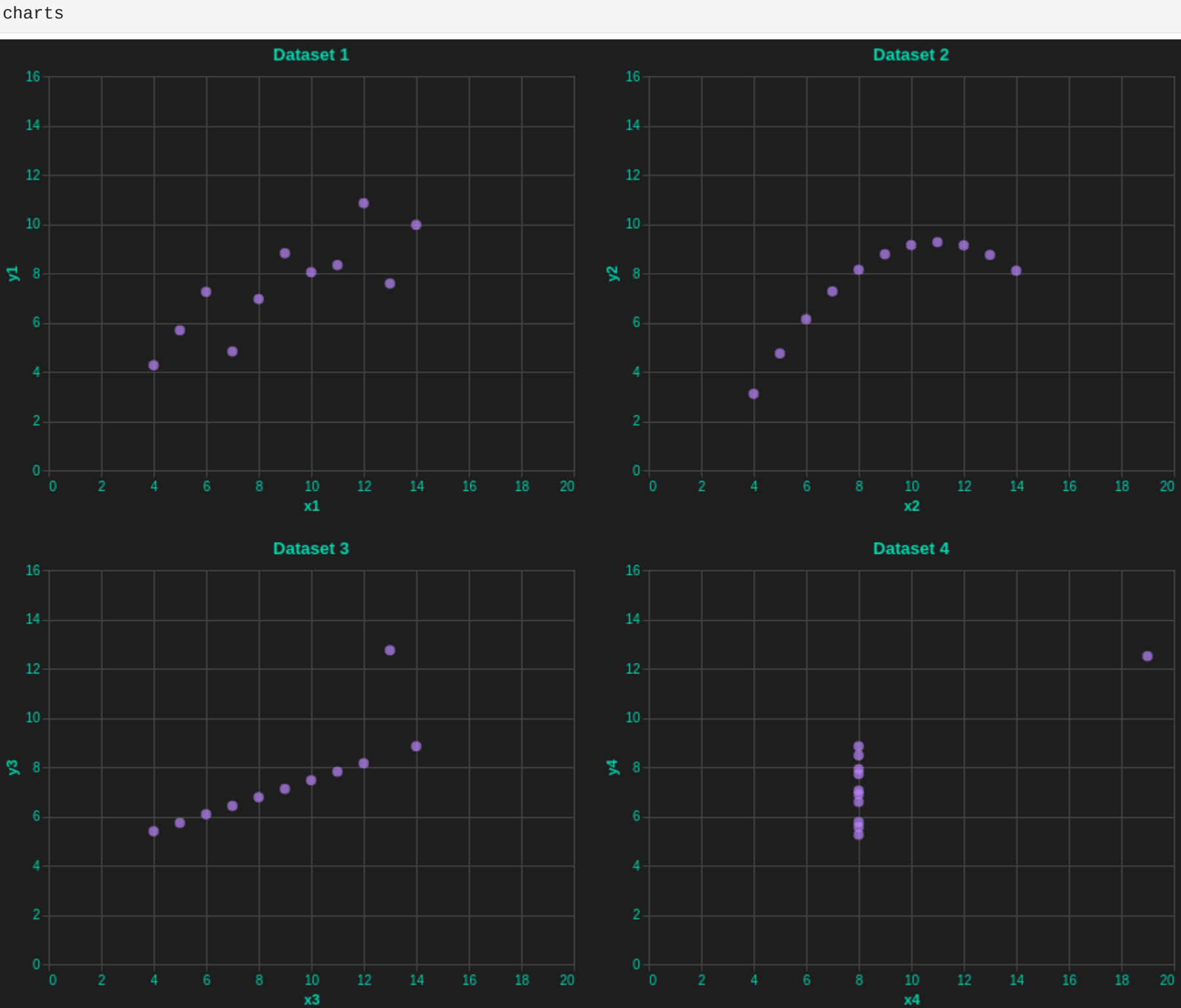
In [ ]:

```
# Concatenate the charts
charts = ((chart[0] | chart[1]) & (chart[2] | chart[3]))

# Configure the charts
charts = charts.configure_view(
    fill = '#121212'
)
charts = charts.configure(background = "#1E1E1E")
charts = charts.configure_mark(color = "#BB86FC")
charts = charts.configure_axis(
    gridColor = "#454545",
    domainColor = "#454545",
    tickColor = "#454545",
    labelColor = "#07D9B2",
    titleColor = "#07D9B2"
)
charts = charts.configure_title(color = "#07D9B2")

charts
```

Out[ ]:



Agora, fazemos um template HTML para inserir os gráficos:

In [ ]:

```
html_template = """
<!DOCTYPE html>
<html>
<head>
    <script src="https://cdn.jsdelivr.net/npm/vega@{vega_version}"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-lite@{vegalite_version}"></script>
    <script src="https://cdn.jsdelivr.net/npm/vega-embed@{vegaembed_version}"></script>
    <style>
        {style}
    </style>
</head>
<body>

<h1>Quarteto de Anscombe</h1>
<h2>Introdução</h2>
<p>{introduction}</p>

<div id="mybeautifulgraph"></div>

<br>

<h2>Considerações finais</h2>
<p>{comments}</p>

<br>

<h2>Observações</h2>
<p>{notes}</p>

<br>

<h2>Bibliografia</h2>
{bibliography}

<br>

<script type="text/javascript">
    vegaEmbed('#mybeautifulgraph', {spec1}).catch(console.error);
</script>
</body>
</html>
"""
```

Criamos uma folha de estilo:

In [ ]:

```
style = """ body {
    background-color: #121212;
    margin-left: 20%;
    margin-right: 20%;
    text-align: justify;
    font-family: Arial, Helvetica, sans-serif;
}
h1 {
    color: #07D9B2;
}
h2 {
    color: #F2E7FE;
}
p {
    color: #E2E2E2;
}"""
```

Anotamos todas as considerações (comentários textuais):

In [ ]:

```
introduction = """O Quarteto de Anscombe consiste em quatro conjuntos de dados nos quais têm as mesmas
estatísticas descritivas: média de x, média de y, variância de x, variância de y, correlação entre x e y,
coeficiente de determinação e regressão linear. Porém, como veremos mais abaixo, os dados não são iguais.
Esse é um famoso exemplo de como as estatísticas descritivas podem ser enganosas e por isso é importante
visualizar os dados antes de tirar conclusões precipitadas. Tendo dito isso, para exibir os dados,
utilizaremos a biblioteca Altair, que é uma biblioteca de visualização de dados baseada em Vega-Lite (a motivação
para o uso dessa biblioteca se encontra descrita nas considerações finais). Por sua vez, segue abaixo os gráficos
de dispersão com os conjuntos de dados do quarteto de Anscombe:"""
```

In [ ]:

```
comments = """Primeiramente, a intenção era utilizar a biblioteca Vega-Lite, mas ao entrar no site oficial da biblioteca
é recomendado o uso da biblioteca Altair e, caso seja desejado utilizar o Vega-Lite, ainda assim é
necessário importar o Altair, pois o Vega-Lite é encontrado como uma "sub biblioteca" da
Altair no Python. A motivação para utilizar o Vega-lite surgiu desde outra disciplina, de Análise Exploratória
de Dados e Visualização, onde ela foi mencionada como uma forma mais simplificada da biblioteca Vega (na qual é,
de certa forma, "mais completa" e permite mais personalização, porém é muito complexa de usar). Além disso,
era desejável tanto utilizar alguma biblioteca em Python (pela facilidade com a linguagem) quanto poder visualizar os
dados em uma página web, o que demandaria ser necessário exportar o gráfico de alguma forma que fosse compatível
com HTML. Dessa forma, a biblioteca Altair foi escolhida por unir essas duas características de maneira que conseguimos
exportar nossas visualizações facilmente em diversos formatos, inclusive mais de um que nos permite usá-las em páginas
web e, além disso, nos dá a possibilidade de implementar gráficos interativos de forma simplificada.
Ao trabalhar com a biblioteca, foi possível perceber que ela parece implementar bem a "gramática de gráficos", de forma
que vemos que de acordo com a sequência de modificação que fazemos na visualização, podemos ver uma sobreposição dos
recursos gráficos, de forma que podemos utilizar isso a nosso favor para construir com mais facilidade gráficos mais
complexos, com mais recursos, e coerentes. Por fim, a biblioteca é bem simples de usar e permite uma boa personalização
dos gráficos e, principalmente, tem uma boa documentação, desde o básico para começar a trabalhar com a biblioteca de
forma simples (inclusive permitindo implementar interatividade simples sem quase nenhum esforço), até recursos
"mais avançados", que abrem a possibilidade de personalização dos gráficos de forma mais detalhada."""
```

In [ ]:

```
notes = """O código para gerar os gráficos e esta página HTML foram feitos em um notebook python, que pode ser encontrado
em um documento separado. Tanto essa página HTML quanto o notebook python estão disponíveis no <a
href="https://github.com/BrunoFornaro/Visualiza-o-da-Infoma-o---Quarteto-de-Anscombe"
target="_blank">repositório do GitHub</a>."""
```

In [ ]:

```
bibliography = """
<p>ALTAIR. Vega-Altair: Declarative Visualization in Python. Disponível em: <a
href="https://altair-viz.github.io/index.html"
target="_blank">altair-viz.github.io</a>. Acessado em 27 de setembro de 2022.</p>

<p>WIKIPEDIA. Quarteto de Anscombe. Disponível em: <a
href="https://pt.wikipedia.org/wiki/Quarteto_de_Anscombe"
target="_blank">pt.wikipedia.org/wiki/Quarteto_de_Anscombe</a>. Acessado em 27 de setembro de 2022.</p>

<p>VANDERPLAS, Jake. Multiple Charts in one HTML. Disponível em: <a
href="https://github.com/altair-viz/altair/issues/1422"
target="_blank">github.com</a>. Acessado em 1 de outubro de 2022.</p>

<p>MATERIAL. Dark theme. Disponível em: <a
href="https://material.io/design/color/dark-theme.html"
target="_blank">material.io</a>. Acessado em 2 de outubro de 2022.</p>

<p>ADOBE. Adobe Color. Disponível em: <a
href="https://color.adobe.com/pt/create/image"
target="_blank">color.adobe.com/pt/create/image</a>. Acessado em 2 de outubro de 2022.</p>

<p>ALTAIR. Top-Level Chart Configuration. Disponível em: <a
href="https://altair-viz.github.io/user_guide/configuration.html#config-axis"
target="_blank">altair-viz.github.io</a>. Acessado em 2 de outubro de 2022.</p>

"""
```

Por fim, podemos unir tudo escrevendo um documento HMTL com as nossas variáveis criadas:

In [ ]:

```
with open('results.html', 'w') as f:
    f.write(html_template.format(
        vega_version = alt.VEGA_VERSION,
        vegalite_version = alt.VEGALITE_VERSION,
        vegaembed_version = alt.VEGAEMBED_VERSION,
        spec1 = charts.to_json(indent=None),
        style = style,
        introduction = introduction,
        comments = comments,
        notes = notes,
        bibliography = bibliography
    ))
```