Introdução a Ciência de Dado

Professor: Alex Pereira

Comentários dos Exercícios 5.1 e 5.2

- Pivoteamento da Tabela do IDEB (municípios)
  - Quais colunas compõem a chave primária desta tabela?

ano	sigla_uf	id_municipio	rede	ensino	anos_escolares	taxa_aprovacao
2005	AC	1200013	estadual	fundamental	finais (6-9)	90.0
2005	AC	1200013	estadual	fundamental	iniciais (1-5)	80.4
2005	AC	1200013	municipal	fundamental	finais (6-9)	*null*
2005	AC	1200013	municipal	fundamental	iniciais (1-5)	66.3
2005	AC	1200013	publica	fundamental	finais (6-9)	89.0

- Média conjunta de notas de iniciais (1-5) e finais (6-9)
  - Qual o significado dessa média ?
    - ✓ Alternativas
      - Pivotar a coluna rede e a coluna anos_escolares
      - Filtrar (com cláusula where) os anos iniciais ou finais
        - Fazer os dois pivôts seguido de um join (no SQL ou no Pandas)
- Qual o efeito de Inner Join nesta query ?

Comentários dos Exercícios 5.1 e 5.2

- Pivot com group by

```sql
SELECT * FROM
( SELECT ano, id_municipio, rede, AVG(nota_saeb_media_padronizada) AS media_nota_saeb
FROM `basedosdados.br_inep_ideb.municipio` ideb
GROUP BY ano, id_municipio, rede )
PIVOT (SUM(media_nota_saeb) AS nota_saeb FOR rede IN ('municipal', 'estadual', 'federal', 'publica') ) AS ideb

```

Local de processamento: US

Resultados da consulta SALVAR RESULTADOS EXPLORAR DADOS

Consulta finalizada (tempo decorrido:3,0 s, bytes processados: 23,9 MB)

Informações do job | Resultados | JSON | Detalhes da execução

Linha	ano	id_municipio	nota_saeb_municipal	nota_saeb_estadual	nota_saeb_federal	nota_saeb_publica
1	2005	1200013	4.1680932	3.9275875	*null*	3.9728335
2	2005	1200054	3.794003	4.2849584	*null*	4.21941115
3	2005	1200104	4.433672	4.2395434	*null*	4.2812326
4	2005	1200138	3.693315	4.3400469	*null*	4.0166576

Comentários dos Exercícios 5.1 e 5.2

- Pivot sem group by

```sql
SELECT * FROM
(
    SELECT ano, id_municipio, rede, nota_saeb_media_padronizada
    FROM `basedosdados.br_inep_ideb.municipio` ideb
)
PIVOT (AVG(nota_saeb_media_padronizada) AS nota_saeb FOR rede IN ('municipal', 'estadual', 'federal', 'publica') )
```

Local de processamento: US

Resultados da consulta SALVAR RESULTADOS EXPLORAR DADOS

Consulta finalizada (tempo decorrido:2,5 s, bytes processados: 23,9 MB)

Informações do job | Resultados | JSON | Detalhes da execução

Linha	ano	id_municipio	nota_saeb_municipal	nota_saeb_estadual	nota_saeb_federal	nota_saeb_publica
1	2005	1200013	4.1680932	3.9275875	*null*	3.9728335
2	2005	1200054	3.794003	4.2849584	*null*	4.21941115
3	2005	1200104	4.433672	4.2395434	*null*	4.2812326
4	2005	1200138	3.693315	4.3400469	*null*	4.0166576
5	2005	1200179	4.2907338	4.3749299	*null*	4.31873845

Comentários dos Exercícios 5.1 e 5.2

- A turma conseguiu fazer o pivoteamento da tabela SAEB
  - Seguido do join entre as tabelas
    - ✓ É uma tarefa um pouco complexa
      - Mission accomplished! 👏
- Alguns colegas ainda dependem da ajuda
  - do professor/monitor para queries SQL
- Enunciado do Comentário do 5.2
  - 5.2) Elenque critérios a serem considerados para escolher entre a solução do exercício 5.1 ou a solução do exercício 5.2.
    - ✓ Principais tipos de respostas
      - Explicação da query;
      - Juízo de valor (qual alternativa é melhor);
      - Lista de critérios.

Comentários dos Exercícios 5.1 e 5.2

- Não há uma resposta exata/genérica para qual é a melhor solução
  - Depende das restrições, propósitos e recursos envolvidos.
- Alguns possíveis critérios a serem considerados
  - Custo operacional (Armazenamento no DW e Processamento na Nuvem)
    - ✓ BigQuery é um banco de dados Colunar
  - Tempo de execução
  - Facilidade de Manutenção / Simplicidade da solução
    - ✓ Impacto nos custos de RH e **riscos de continuidade dos projetos**
  - Domínio das tecnologias envolvidas pela equipe de engenheiros e cientistas de dados
- Desenvolva sua habilidade de síntese
  - Sem ela é difícil ponderar os trade-offs

Classifique essas sentenças nos respectivos critérios

- A 5.2 é mais simples, pois se divide o problema em partes menores, mais fáceis de se implementar, se compreender e, consequentemente, dar manutenção.
- A solução do pandas foi mais rápida (eficiente) e tornou a interpretação do código muito mais prática, sendo uma opção mais reprodutível. Por outro lado, a solução em SQL permite resolver o problema em uma única célula de código e poderia ser executada diretamente no SGBD. Por fim, a execução no colab tem o limite de memória, o que poderia ser um empecilho dependendo da base de dados.
- No caso do 5.1, cujo SQL com o merge/join já foi executado lá no BigQuery, temos somente uma transação, uma transferência de dados, de forma a reduzir tráfego, conexões com o servidor, etc. Foi obtido um único dataframe com 94.616 registros.
- Custo de armazenamento e processamento no servidor de Banco de Dados (Dependendo da criticidade desses custos, poderia optar-se por uma consulta mais geral, como no 5.2 - que retornasse um volume dados maior - ou uma consulta mais específica, como no 5.1)
- 5.2: Resultado mais inteligível, porém possui tempo de execução muito superior.
- Acredito que o domínio/experiência do cientista de dados em cada tecnologia (SQL ou Pandas/Python) também pode ser um fator importante. No meu caso, como já fui programador, tenho a percepção que o Pandas/Python oferece uma produtividade e um conjunto de recursos superiores a linguagem SQL.

Custo Operacional; Tempo de Execução; Facilidade de Manutenção / Simplicidade da solução; Domínio das tecnologias envolvidas

Principais operações de manipulação de dados num BI

- Do menos complexo para o mais complexo
  - Filtro de Colunas
  - Filtro de Linhas
  - Join
  - Group By
  - Pivot com um único registro por grupo
  - Pivot com mais de um registro por grupo
- Domine essas operações no SQL e no Pandas
  - Treine com exercícios de recall (lembrar sem estímulo)
    - ✓ Em vez de reconhecer (olhar um exemplo pronto)

Revisão da Aula/Semana Anterior

- Foram apresentadas duas possíveis aplicações/utilidades
  - distintas das ferramentas de BI.
    - ✓ Quais são elas ?

Join com chave composta menor

- Adicionar ao modelo de dados o consumo de Energia por UF
  - basedosdados.br_mme_consumo_energia_eletrica.uf
- Seu modelo de dados tem granularidade por município
  - Se repetirmos o valor do consumo para cada registro
    - ✓ Não conseguiremos calcular o consumo do Brasil no Data Studio
      - A soma seria muito maior do que o valor real
- Como resolver ?

Tabela Fato

ano	sigla_uf	id_municipio	populacao	nome_municipio	pib
2002	RO	1100023	78039.0	Ariquemes	449592816.0
2003	RO	1100023	79680.0	Ariquemes	539636214.0
2004	RO	1100023	86901.0	Ariquemes	657193231.0
2005	RO	1100023	85031.0	Ariquemes	749021187.0
2006	RO	1100023	86924.0	Ariquemes	790696634.0

Tabela da Dimensão de Consumo de Energia (MWh)

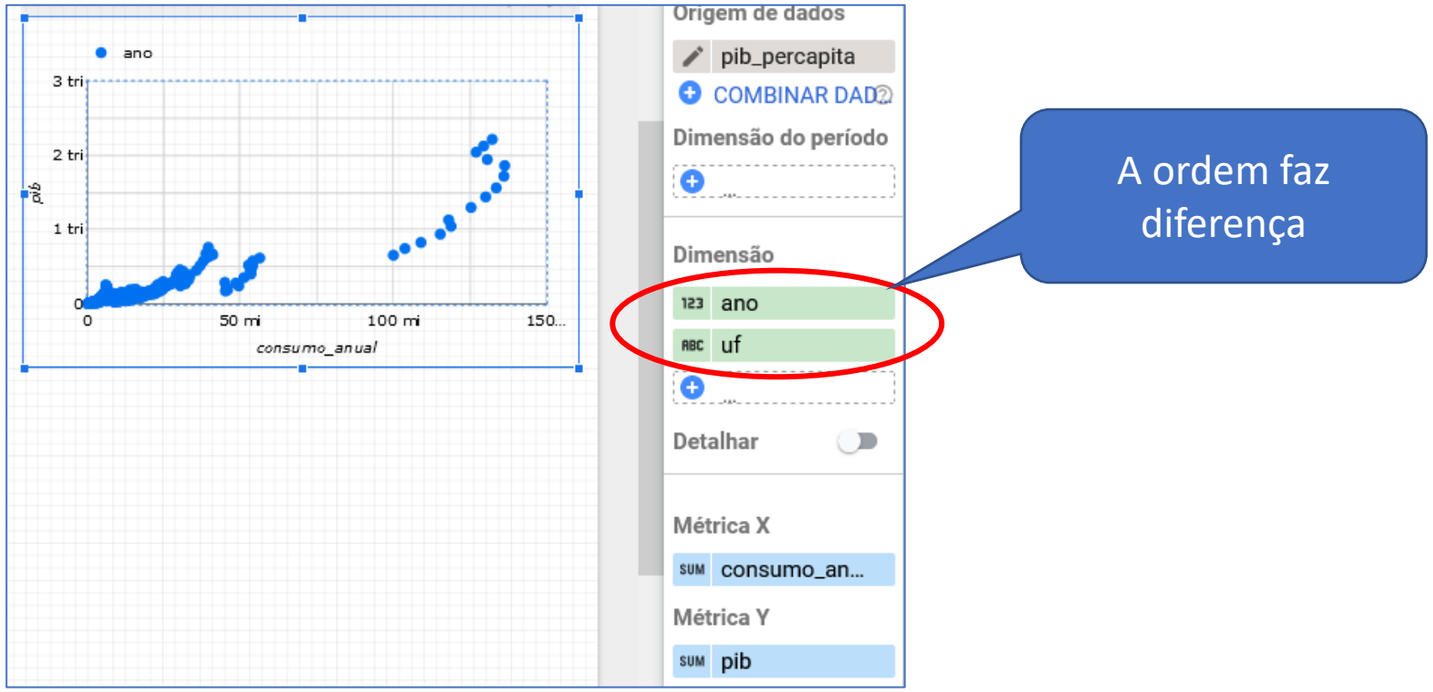
ano	mes	sigla_uf	tipo_consumo	consumo	numero_consumidores
2004	1	RO	Total	112812.0	*null*
2004	1	AC	Total	34840.05	*null*
2004	1	AM	Total	274773.0	*null*
2004	1	RR	Total	31695.63	*null*
2004	1	PA	Total	1011353.04	*null*

Join com chave composta menor

- Solução
  - Crie um dataframe com o consumo repetido para apenas um dos municípios
    - ✓ Não utilizar esta métrica como consumo de energia de municípios
  - Existe mais alguma solução ?

Join com chave composta menor: Visualização

- Adicionar ao modelo de dados
  - Os dados do consumo de Energia por UF
    - ✓ Caderno colab
- Scatter Plot com os valores
  - do consumo de energia e do PIB dos Estados



Pivotar tabela usando a função CASE

- Calcular a quantidade de doses 1ª, 2ª, Única, Adicional e Reforço de vacina do COVID-19
  - Para cada UF, Semana, Imunizante
- Classificação das doses (do professor)
  - 1ª Dose
    - ✓ 1ª Dose, Dose, Dose Inicial
  - 2ª Dose
  - Reforço
    - ✓ Qualquer contendo a palavra Reforço
  - Adicional
    - ✓ Doses Adicional e 3ª Dose
  - Única

Linha	dose	Qtd
1	Única	224783
2	1º Reforço	33188
3	3º Reforço	2
4	Tratamento com dezessete doses	1
5	Revacinação	3
6	1ª Dose	154696905
7	2º Reforço	1879
8	Reforço	11565304
9	Dose Adicional	511211
10	Dose Inicial	1378
11	Tratamento com uma dose	2
12	1ª Dose Revacinação	759
13	2ª Dose Revacinação	862
14	Dose	4353953
15	2ª Dose	118663607
16	3ª Dose	308020

Diferença entre 3ª Dose (Adicional) e Reforço

Se entender, já está falando a língua dos nerds

Regex para detectar tipos de doses

- Usar a função REGEXP_CONTAINS(value, regexp) do BigQuery
  - REGEXP_CONTAINS(dose, regexp)
    - ✓ 1ª Dose
      - 1ª Dose, Dose, Dose Inicial
        - ' 1ª Dose$|^Dose$|Inicial'
    - ✓ '2ª Dose$'
    - ✓ Reforço
      - Qualquer contendo a palavra Reforço
        - 'Reforço'
    - ✓ Adicional
      - Doses Adicional e 3ª Dose
        - 'Dose Adicional|3ª Dose'
    - ✓ 'Única'
- Solução

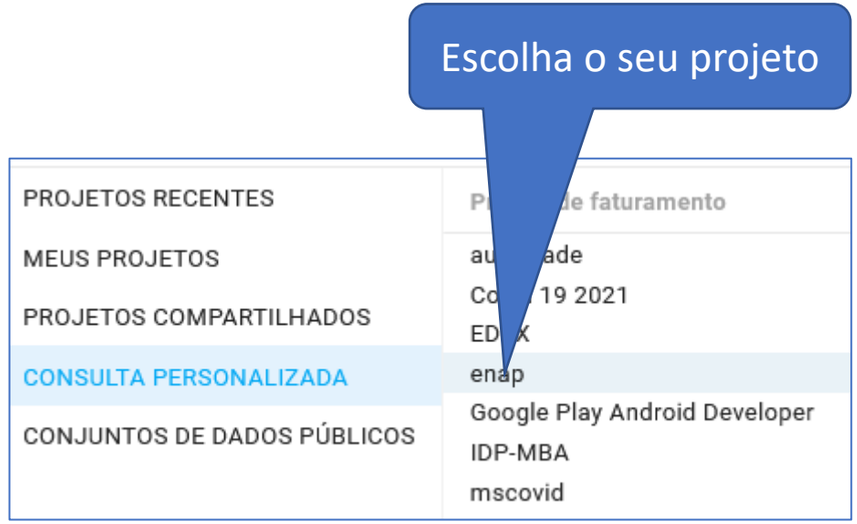
Linha	dose	Qtd
1	Única	224783
2	1º Reforço	33188
3	3º Reforço	2
4	Tratamento com dezessete doses	1
5	Revacinação	3
6	1ª Dose	154696905
7	2º Reforço	1879
8	Reforço	11565304
9	Dose Adicional	511211
10	Dose Inicial	1378
11	Tratamento com uma dose	2
12	1ª Dose Revacinação	759
13	2ª Dose Revacinação	862
14	Dose	4353953
15	2ª Dose	118663607
16	3ª Dose	308020

Custom query no Data Studio e BigQuery

- Simulação de projeção de demanda de 2ª e 3ª Dose
  - A partir de input do usuário
    - ✓ no Data Studio
- Custom Query com Parâmetro
  - na Documentação do BigQuery

Atividade 8.2 (5 min)

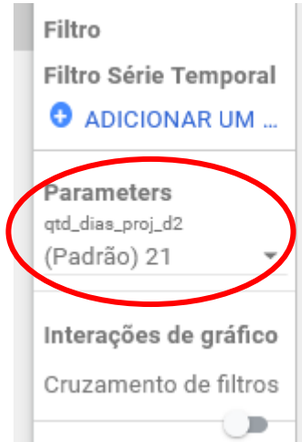
- Simulação de projeção de demanda de 2ª
  - A partir de input do usuário
    - ✓ no Data Studio e query no BigQuery
- Criar uma Consulta Personalizada
  - Escolha um projeto **SEU**




```
SELECT v.sigla_uf, v.vacina_apelido, v.semana, v.mes, v.qt_total, v.qt_D1, v.qt_D2, v.qt_Reforco, v.qt_Adicional, v.qt_Unica,
vp.qt_D2_Proj, vp.semana_proj, vp.sigla_uf_proj, vp. vacina_apelido_proj
FROM `enap-331414.enapdatasets.vacinacao` v
JOIN (
   SELECT sigla_uf as sigla_uf_proj, vacina_apelido as vacina_apelido_proj, qt_D1 as qt_D2_Proj, DATE_ADD(semana, INTERVAL
@qtd_dias_proj_d2 DAY) as semana_proj
   FROM `enap-331414.enapdatasets.vacinacao`
   ) as vp
ON v.sigla_uf=vp.sigla_uf_proj and v.vacina_apelido=vp.vacina_apelido_proj and vp.semana_proj=v.semana
order by v.sigla_uf, v.vacina, v.semana, vp.semana_proj, vp.sigla_uf_proj, vp.vacina_apelido_proj
```

Atividade 8.2

- Simulação de projeção de demanda de 2ª
  - A partir de input do usuário
    - ✓ no Data Studio e query no BigQuery
- Criar um gráfico de **Série Temporal**
  - Eixo x: data (**semana**)
    - ✓ Ajuste para o tipo semana ano
  - Eixo y: Projeção da 2ª dose (**qt_D2_Proj**)
  - Na métrica detalhada: nome da vacina (**vacina_apelido**)
  - Ordenação: pelo campo semana
    - ✓ Crescente
  - Ative a opção Cumulativo na aba estilos do gráfico
    - ✓ Para as 4 séries
- Teste vários valores para o parâmetro **qtd_dias_proj_d2**



Atividade 8.3 – Visualizar a projeção futura no gráfico (5 min)

- Simulação de projeção de demanda de 2ª
  - Mude a query para FULL OUTER JOIN
- Alterar a query para ficar assim

```
SELECT v.sigla_uf, v.vacina_apelido, v.semana, v.mes, v.qt_total, v.qt_D1, v.qt_D2, v.qt_Reforco, v.qt_Adicional, v.qt_Unica,
vp.qt_D2_Proj, vp.semana_proj, vp.sigla_uf_proj, vp. vacina_apelido_proj
FROM `enap-331414.enapdatasets.vacinacao` v
FULL OUTER JOIN (
   SELECT sigla_uf as sigla_uf_proj, vacina_apelido as vacina_apelido_proj, qt_D1 as qt_D2_Proj, DATE_ADD(semana,
INTERVAL @qtd_dias_proj_d2 DAY) as semana_proj
   FROM `enap-331414.enapdatasets.vacinacao`
   ) as vp ON v.sigla_uf=vp.sigla_uf_proj and v.vacina_apelido=vp.vacina_apelido_proj and vp.semana_proj=v.semana
order by v.sigla_uf, v.vacina, v.semana, vp.semana_proj, vp.sigla_uf_proj, vp.vacina_apelido_proj
```

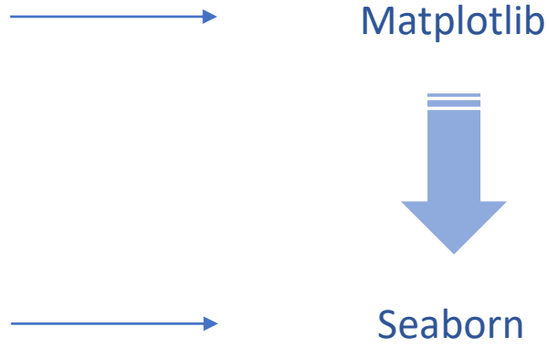
- Criar **2** (o da UF é opcional) campos calculado com as fórmula
  - IFNULL(semana, semana_proj)
  - IFNULL(vacina_apelido, vacina_apelido_proj)
  - IFNULL(sigla_uf, sigla_uf_proj) – não será usado na série temporal
- Adicionar os **2** campos ao gráfico de Série Temporal

Grammar of Graphics

- Gramática
  - um conjunto de regras que regem o uso de uma língua
- Grammar of Graphics (Leland Wilkinson)
  - é uma ferramenta que nos permite descrever concisamente os componentes de um gráfico;
  - útil para descrever e criar uma ampla gama de gráficos estatísticos.
- Mudança de mindset
  - Em vez de uma função para criar um gráfico de barras
    - ✓ Usa-se uma função para mapear (encoding) de variáveis nos eixos, exemplo:
      - x: variável categórica (ex.: empresas A, B e C)
      - y: variável contínua (ex.: faturamento)
    - ✓ mark: barra
  - Construir gráficos a partir de suas partes elementares (building blocks)

Vega e Vega-Lite

- São duas implementações de uma gramática de gráficos
  - Estendem a gramática de Wilkinson
    - ✓ adicionando interatividade
      - Open-source
        - Criado num laboratório da Universidade de Washington
    - ✓ Expressiveness: the quality of expressing somebody's thoughts and feelings
- Vega é uma implementação de baixo nível
  - Com mais opções e mais flexível
    - ✓ E mais difícil/demorado para aprender
- Vega-Lite é uma gramática de alto nível
  - Mais concisa, com valores padrão convenientes
    - ✓ Convertido para vega no momento da exibição
  - Biblioteca de gráficos disponível no Data Studio e no **Observable**



Vega e Vega-Lite

- Uma gramática facilita a geração automática de gráficos
  - E recomendação de gráficos
    - ✓ Voyager: ferramenta para geração automatizada de gráficos
      - Sugere várias visualizações para o mesmo conjunto de variáveis
- Possíveis aplicações no futuro
  - NLP (GPT-3) da OpenAI
    - ✓ Geração de código a partir de linguagem natural

```
"""
Table customers, columns = [CustomerId, FirstName, LastName, Company, Address, City, State,
Country, PostalCode, Phone, Fax, Email, SupportRepId, TotalSpend]

Create a MySQL query for all customers in Texas who have spent over five thousand dollars.
"""

query = "SELECT * FROM customers WHERE State = 'TX' AND TotalSpend > 5000"
```

```
~> find all files ending in .log in /var/log
🧠 Thinking...
<~ find /var/log -name "*.log"
```

Vega-Lite: Building blocks

- Data
  - Fonte dos dados (data source)
- Transform
  - Filtro, agregação, segmentação/categorização
- Mark
  - Elemento de representação gráfica (pontos, linhas, barras)
- Encoding (Mapeamento/codificação)
  - mapeamento entre dados e marks
- Scale
  - Funções que mapeiam dados em valores visuais (pixels)
- Guides
  - Eixos e legendas

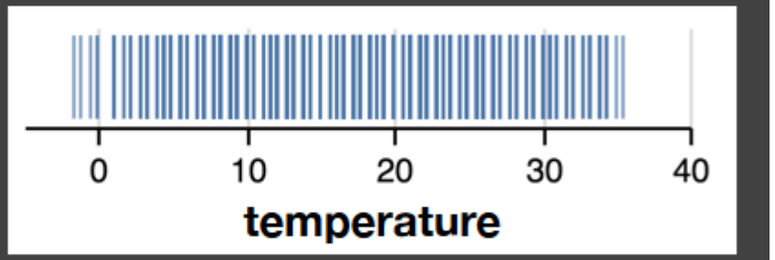
Exemplo com Vega-Lite

Abstract Data → Visual Representation

Weather Data for Seattle

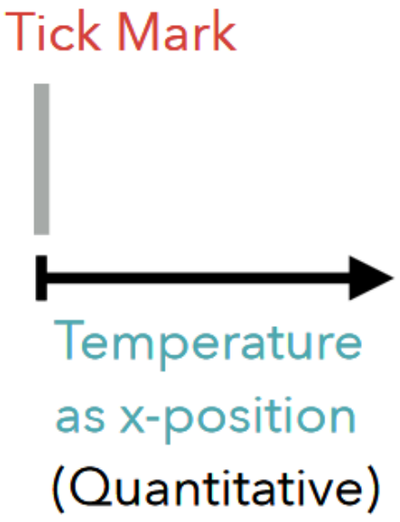
date	temperature	precipitation	weather
1/1	10.6	10.9	"rain"
1/2	11.7	0.8	"drizzle"
1/3	12.2	10.2	"rain"
…	…	…	…

Strip Plot of Temperature

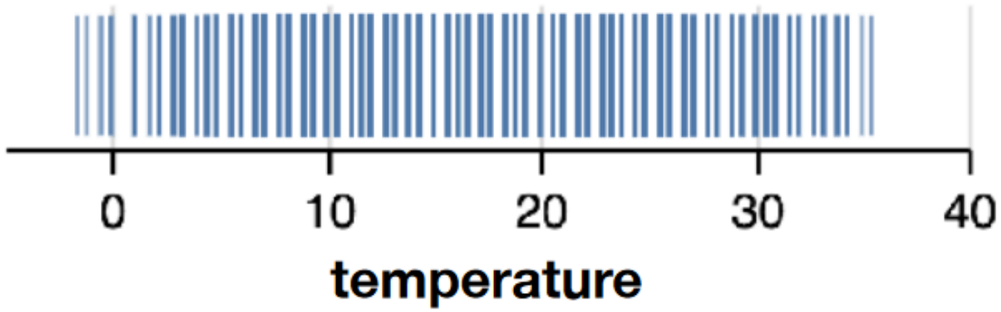


Fonte: https://www.domoritz.de/talks/VegaLite-OpenVisConf-2017.pdf

Strip Plot = (Tick with x=field)




```
{
  data: {url: "weather-seattle.json"},
  mark: "tick",
  encoding: {
    x: {
      field: "temperature",
      type: "quantitative"
    }
  }
}
```

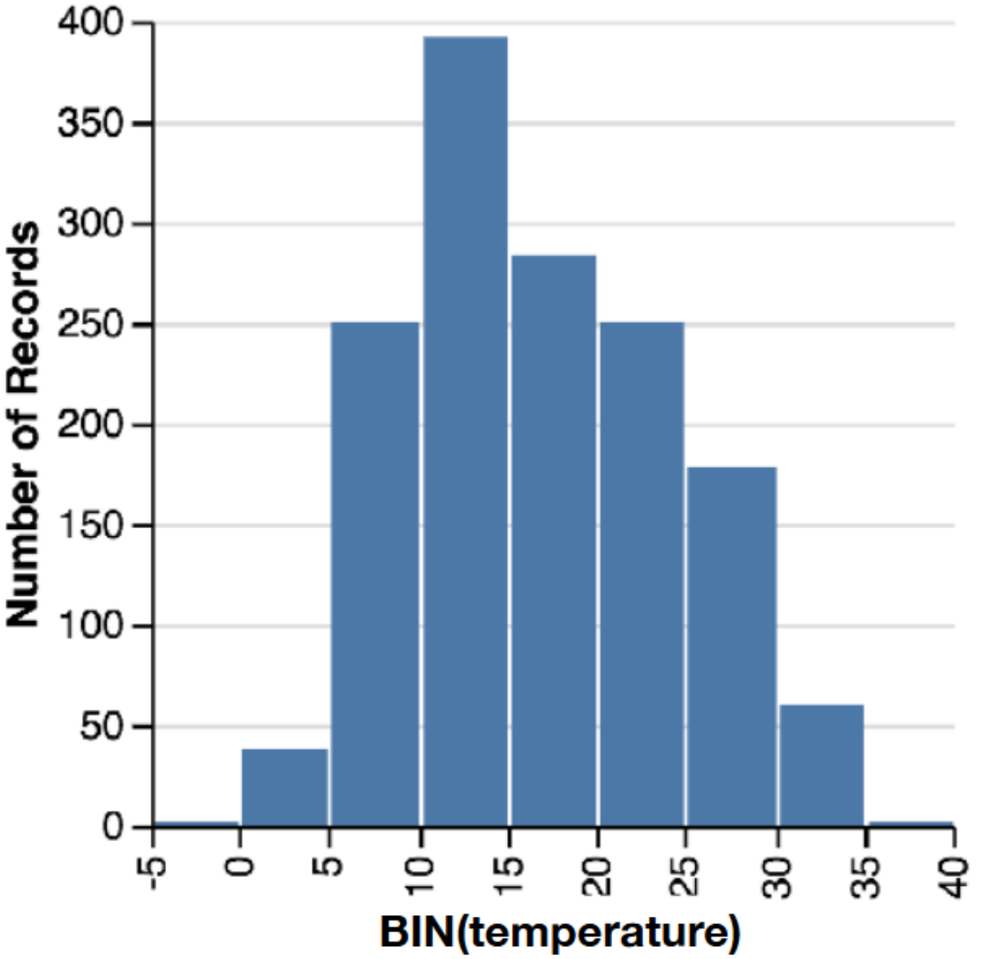


Vega-Lite is portable JSON specification

Fonte: https://www.domoritz.de/talks/VegaLite-OpenVisConf-2017.pdf

Histogram = (Bar with x=binned field, y=count)

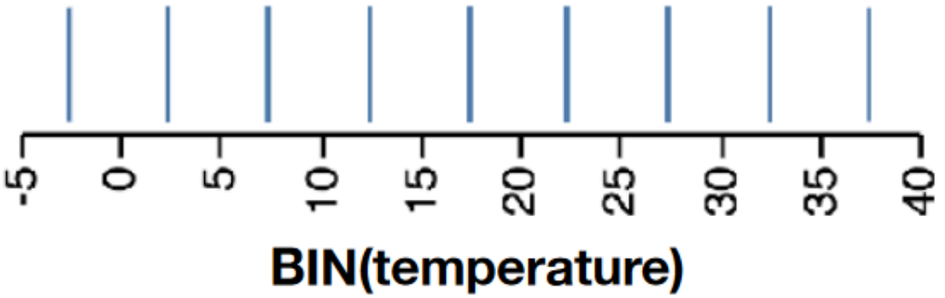
Goal



Fonte: https://www.domoritz.de/talks/VegaLite-OpenVisConf-2017.pdf

Histogram = (Bar with x=binned field, y=count)

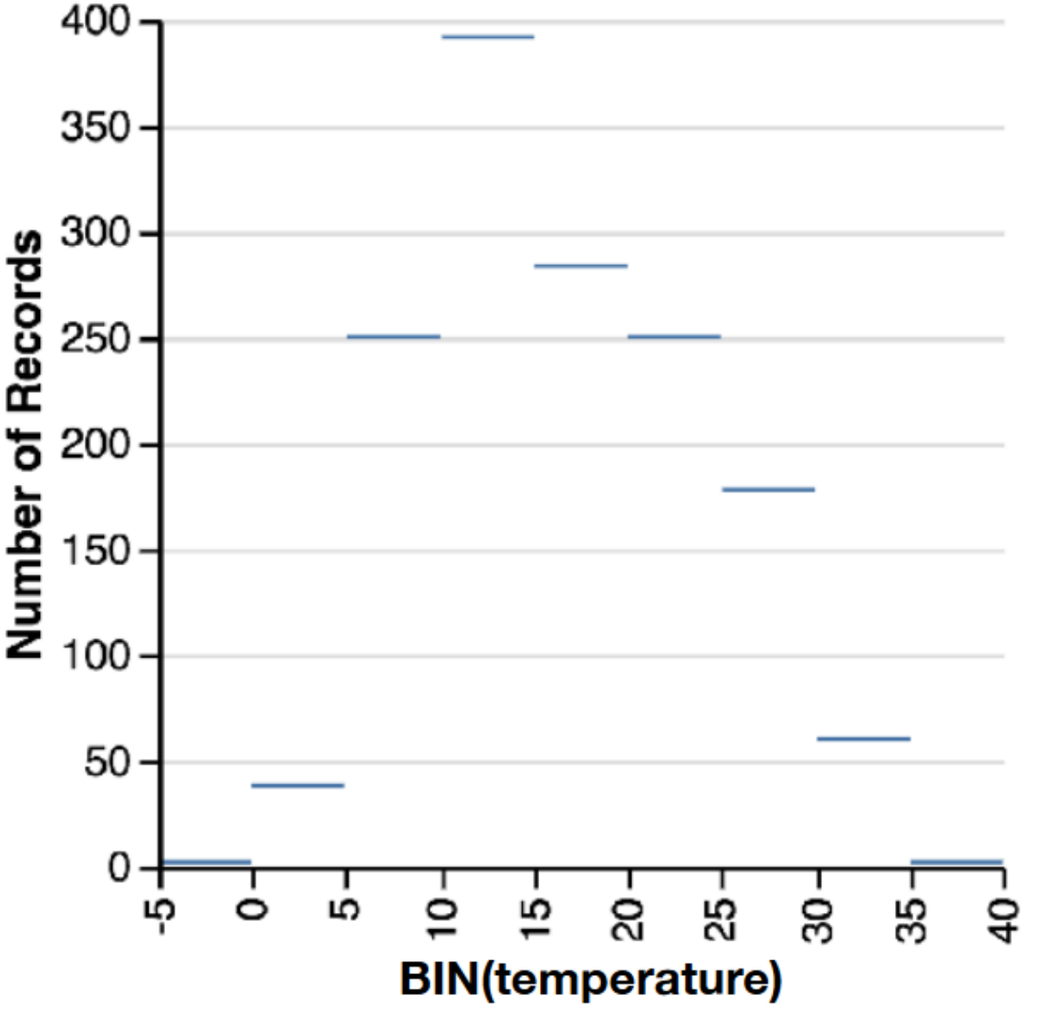
```
{
  data: {url: "weather-seattle.json"},
  mark: "tick",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    }
  }
}
```



Fonte: https://www.domoritz.de/talks/VegaLite-OpenVisConf-2017.pdf

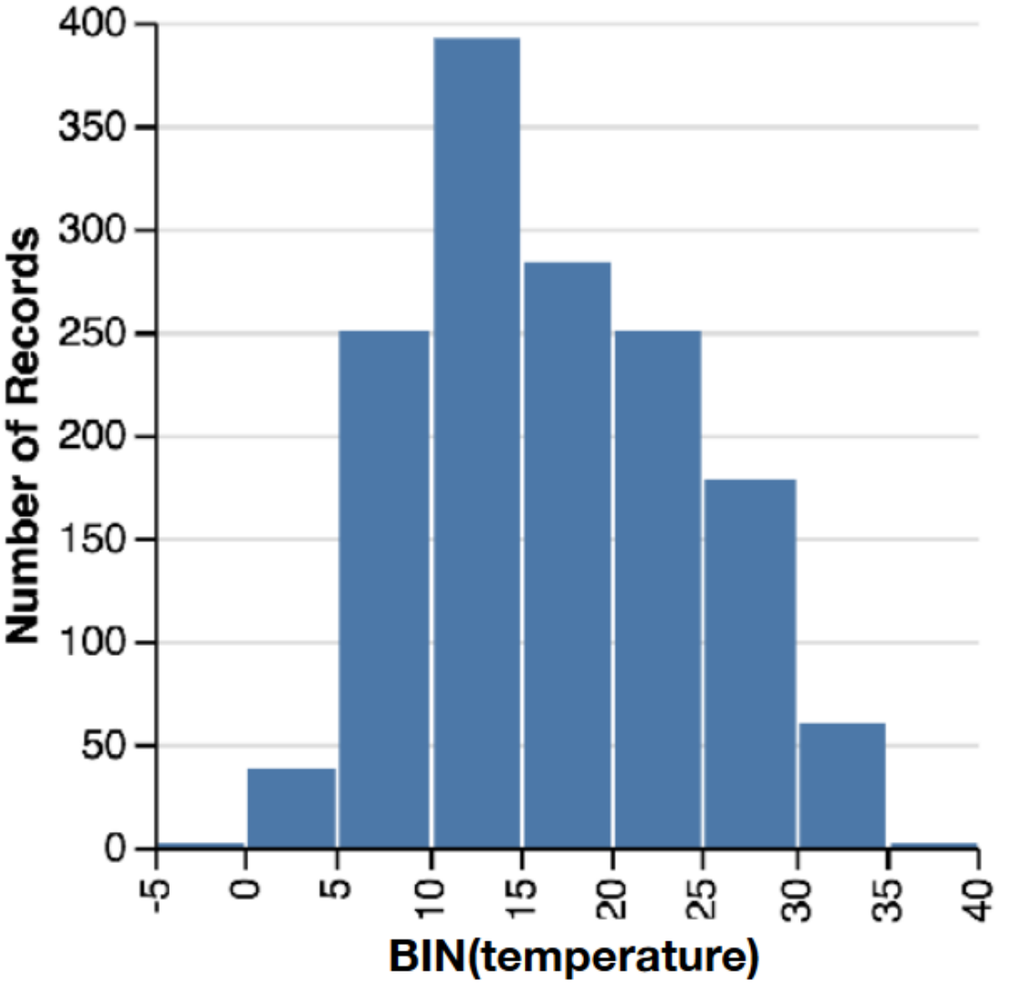
Histogram = (Bar with x=binned field, y=count)

```
{
  data: {url: "weather-seattle.json"},
  mark: "tick",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    }
  }
}
```



Histogram = (Bar with x=binned field, y=count)

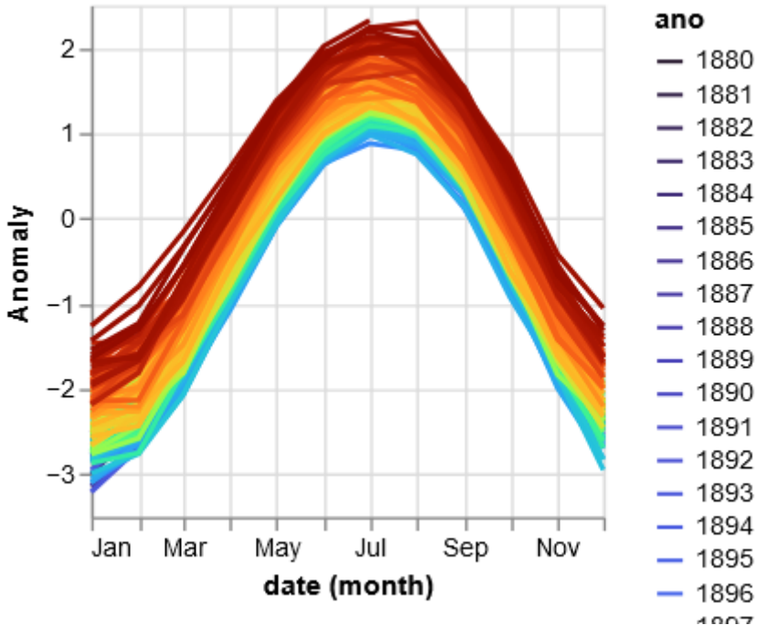
```
{
  data: {url: "weather-seattle.json"},
  mark: "bar",
  encoding: {
    x: {
      bin: true,
      field: "temperature",
      type: "quantitative"
    },
    y: {
      aggregate: "count",
      type: "quantitative"
    }
  }
}
```



Fonte: https://www.domoritz.de/talks/VegaLite-OpenVisConf-2017.pdf

Gráfico de Linha da Anomalia de Temperaturas

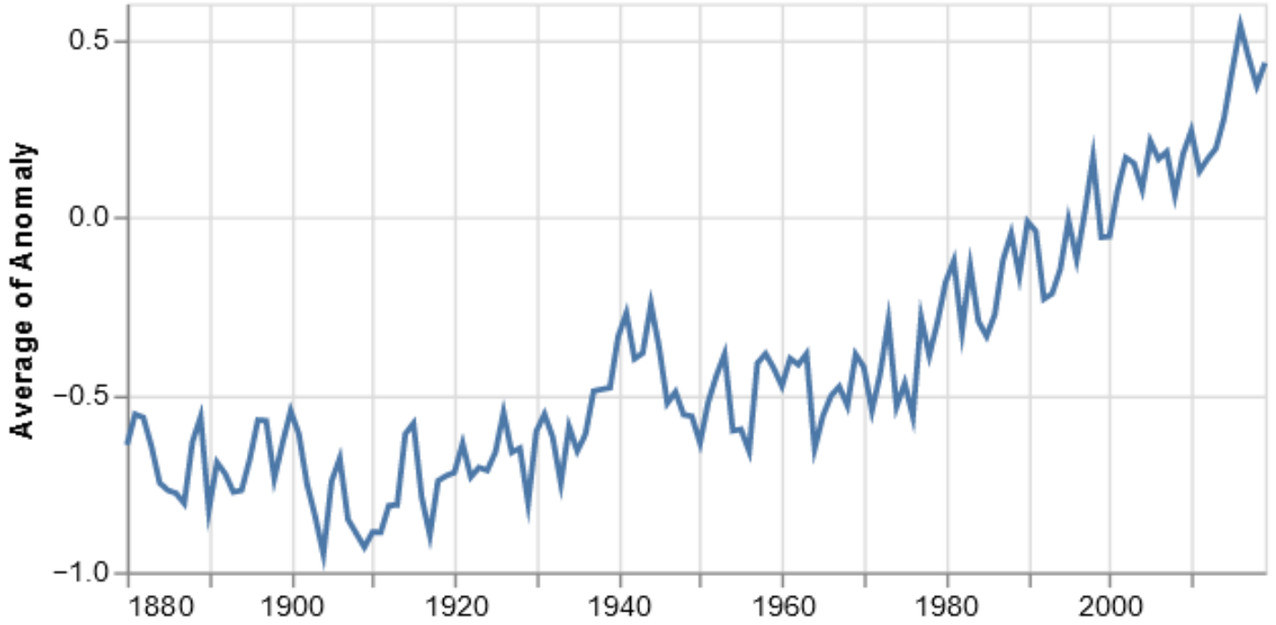
```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "data": {
    "url":
"https://raw.githubusercontent.com/alexlopespereira/enapespcd2021/main/data/originais/aquecimento_global/global_temperature_anomalies_tratado.csv",
    "format": {"type": "csv"}
  },
  "mark": "line",
  "encoding": {
    "x": {"timeUnit": "month", "field": "data1"},
    "y": {"field": "Anomaly", "type": "quantitative"},
    "color": {"field": "ano", "type": "ordinal", "scale": {"scheme":
"turbo"}}
  }
}
```



- **Não** copie o código do slide. Os códigos estão aqui.
- Colab usado para gerar esses dados
- Qual seria o equivalente no Seaborn e Data Studio

Atividade 8.3 (até o final da aula)

- Usando o editor do vega-lite
  - Altere o código do exemplo anterior para criar um
    - ✓ Gráfico de linha
      - x: anos
      - y: média da anomalia da temperatura



Exercício 8.1

- Escolha um tema e seus respectivos dados, à sua conveniência
- Faça um relatório no Google Data Studio
  - No formato de uma história
    - ✓ No mesmo layout do relatório deste vídeo, com gráficos dispostos verticalmente
      - Numa sequência que ajuda a contar uma história
- Seu relatório/história deve conter
  - Pelo menos 3 gráficos ou tabelas
    - ✓ E para cada gráfico/tabela pelo menos 1 comentário/anotação
- Use a metodologia de ETL, DW e Ferramenta de BI
  - apresentada no curso
    - ✓ Hospede seu modelo de dados no BigQuery
- Informe aqui o link para o seu caderno colab e o link público do seu dashboard