

# Métodos Quantitativos

## Aula 03

Preparação de Dados

Roberto Massi de Oliveira  
Alex Borges Vieira

# Preparação de Dados

- Palavra-chave: Data Munging / Data wrangling

*“Most data scientists spend much of their time cleaning and formatting data. The rest spend most of their time complaining that there is no data available to do what they want to do.”*

SKIENA, S. S. *The Data Science Design Manual*. Springer, 2017.

# Linguagens para Data Science

- Python:
  - Linguagem atual, amplamente utilizada
  - Muitas bibliotecas para data science: Scipy, Numpy, etc
  - Linguagem interpretada, mais lenta que linguagens compiladas (há controvérsias)  
<https://guide.freecodecamp.org/computer-science/compiled-versus-interpreted-languages/>
- Pearl:
  - Era a linguagem em ascensão, até que foi engolida pelo python a partir do ano de 2008
  - Não possui um bom suporte para códigos orientados a objeto
  - Bibliotecas não tão completas quanto as de Python
- R:
  - Linguagem mais antiga, mas ainda muito utilizada
  - Bibliotecas mais completas no escopo da estatística
  - Mais recursos para análises e visualizações
  - Blocos de código em R podem ser utilizados em Python

# Linguagens para Data Science

- Matlab:
  - Linguagem desenvolvida para manipulação fácil e eficiente de matrizes
  - Muito utilizada em algoritmos de Machine Learning
  - Sistema proprietário, usualmente substituído pelo seu similar gratuito: GNU Octave
- Java e C/C++:
  - Linguagens importantes no contexto de Big Data
  - Linguagens fundamentais no contexto de Sistemas Distribuídos/Processamento Paralelo
- Mathematica/Wolfram Alpha
  - Boa linguagem para a solução de cálculos e problemas matemáticos
  - Adequada para pequenas análises ou simulações  
<https://www.wolframalpha.com>

# Padrões de Formatos de Dados

- Formatos de fácil análise computacional
  - Podem ser usados e reutilizados
- Formatos de fácil leitura humana
  - Podem ser abertos em softwares leitores de texto
- Podem ser usados por uma ampla variedade de softwares e sistemas
  - Podem ser lidos por aplicações/sistemas/softwares gratuitos

# Padrões de Formatos de Dados

- CSV (*Comma Separated Value*)
  - Formato extremamente simples e popular para troca de dados entre programas
  - Cada linha representa um registro com campos separados por vírgulas

H19					
	A	B	C	D	E
1	1	Ativo	nao	1	
2	11	Ativo Circ	nao	2	
3	11010001	Caixa	sim	3	
4	2	Passivo	nao	4	
5	22	Passivo Ci	nao	5	
6	22020002	Titulos a P	sim	6	
7	3	Receita	nao	7	
8	33	Receita Lj	nao	8	
9	33030003	Impostos	sim	9	
10	4	Despesa	nao	10	
11	44	Custos	nao	11	
12	44040004	µgua	sim	12	



modelo.csv - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
1;Ativo;nao;1
11;Ativo Circulante;nao;2
11010001;Caixa;sim;3
2;Passivo;nao;4
22;Passivo Ciculante;nao;5
22020002;Titulos a Pagar;sim;6
3;Receita;nao;7
33;Receita L;liquida;nao;8
33030003;Impostos;sim;9
4;Despesa;nao;10
44;Custos;nao;11
44040004;µgua;sim;12
```

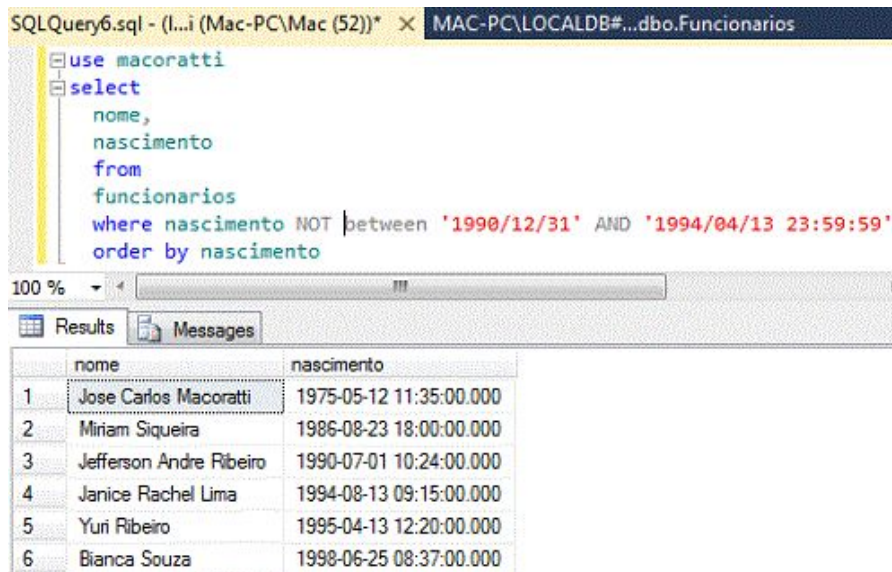
# Padrões de Formatos de Dados

- XML (eXtensible Markup Language)
  - Estruturado, mas não-tabular
  - Markups universais
  - Não serve para exibição, mas sim para transporte e compartilhamento

```
<?xml version="1.0" encoding="UTF-8"?>
- <samples>
  - <sample>
    <collectionID>4f4e48c5e4b07f02db53f51c</collectionID>
    <sourceItemKey>114054</sourceItemKey>
    <title>Exploration no. 132000</title>
    - <alternateTitle>
      <title>Subsurface Document no. 57950</title>
      <title>Source ID EP-11</title>
      <title>Canton Barns</title>
    </alternateTitle>
    - <abstract>
      <![CDATA[Exploration no. 132000 is a test pit (depth 8 feet) described in Subsurface Document no. 57950. The document, a report
        titled "Canton Barns", was prepared by Associated Earth Sciences, Inc. on June 5, 2007 for a residential project.]]>
    </abstract>
    <dataType>Paper reports</dataType>
  </sample>
```

# Padrões de Formatos de Dados

- SQL (Structured Query Language)
  - Dados em tabelas
  - Banco de dados



The screenshot shows a SQL query editor window titled "SQLQuery6.sql - (I...i (Mac-PC\Mac (52)))" with a sub-tab "MAC-PC\LOCALDB#...dbo.Funcionarios". The query is as follows:

```
use macoratti
select
    nome,
    nascimento
from
    funcionarios
where nascimento NOT between '1990/12/31' AND '1994/04/13 23:59:59'
order by nascimento
```

Below the query editor, the "Results" tab is active, displaying a table with 6 rows and 2 columns: "nome" and "nascimento". The table data is as follows:

	nome	nascimento
1	Jose Carlos Macoratti	1975-05-12 11:35:00.000
2	Miriam Siqueira	1986-08-23 18:00:00.000
3	Jefferson Andre Ribeiro	1990-07-01 10:24:00.000
4	Janice Rachel Lima	1994-08-13 09:15:00.000
5	Yuri Ribeiro	1995-04-13 12:20:00.000
6	Bianca Souza	1998-06-25 08:37:00.000



# Padrões de Formatos de Dados

- JSON (JavaScript Object Notation)
  - Usado para transferir objetos de dados de um programa para outro
  - Se assemelha com uma struct
  - É composta por nomes de campos e seus conteúdos
  - Existem bibliotecas que interpretam entradas JSON em qualquer linguagem moderna

```
{
  "id": 1000501,
  "name": "Wilson Júnior",
  "city": {
    "name": "Rio de Janeiro",
    "state": "RJ"
  },
  "age": 24
}
```

# Padrões de Formatos de Dados

- Protocol Buffers (arquivos .proto)
  - Neutro em relação a linguagem/plataforma
  - Usado para a comunicação de pequenas quantidades de dados entre programas
  - Usado na comunicação entre máquinas do Google. Algo semelhante é usado no Facebook

```
// Person created with Protobuf
message Person {

    required string name = 1;
    required int32 id = 2;
    optional string email = 3;
    repeated PhoneNumber phone = 4;

    enum PhoneType {
        MOBILE = 0;
        HOME = 1;
        WORK = 2;
    }

    message PhoneNumber {
        required string number = 1;
        optional PhoneType type = 2 [default = HOME];
    }
}
```

```
{"employees": [
    {"firstName": "John", "lastName": "Doe"},
    {"firstName": "Anna", "lastName": "Smith"},
    {"firstName": "Peter", "lastName": "Jones"}
]}
```

# Coleta de Dados

- Questões a serem respondidas:
  - Quem tem os dados que eu preciso?
  - Por que deixariam tais dados disponíveis?
  - Como posso obtê-los?
- Hunting:
  - Companhias e fonte de dados particulares:
    - Difícil obtenção, tanto por questões estratégicas quanto por questões de privacidade
    - Ex.: Google, Amazon, American Express, Facebook
    - Alguns dados são selecionados e liberados ao público, geralmente por marketing ou para acalmar os ânimos de quem tentaria obtê-los ilegalmente
    - Funcionários podem possuir acesso privilegiado a dados particulares de empresas

# Coleta de Dados

- Hunting:
  - Fontes de dados governamentais:
    - Costumam ser públicos, mas alguns são restritos por segurança ou estratégia
    - Ex.: Endereço, CPF e telefone de cidadãos
    - Preservação de privacidade é uma justificativa comum para a não divulgação
  - Fontes acadêmicas:
    - Pesquisas acadêmicas, colaborações interdisciplinares e interinstitucionais
    - Alguns periódicos pedem para que os dados sejam disponibilizados (reprodutibilidade)
    - Fonte de dados médicos, demográficos, históricos, artísticos, etc.
    - Google Scholar, palavras-chave: “Open Source”, “Data”
    - Geralmente alguém já validou os dados, facilitando o trabalho
  - Geração de dados: Você pode ser uma fonte de dados (extraídos dos seus trabalhos)

# Coleta de Dados

- Scraping:
  - “Raspagem” de informações e preparação das mesmas para análise computacional
  - Websites são escritos em linguagens conhecidas interpretadas por navegadores
  - Emulando um navegador, seu código pode baixar o conteúdo de qualquer site para análise
  - Baixar um conjunto de sites para análise está associado ao termo “spidering”
  - Programas de Scraping podem ser preparados para vasculhar e baixar conteúdos específicos
  - Uma das bibliotecas Python spider/scrapper: BeautifulSoup
  - Web crawling é um spidering avançado: todos os links de uma página são baixados
  - Cuidado com questões legais e políticas do alvo



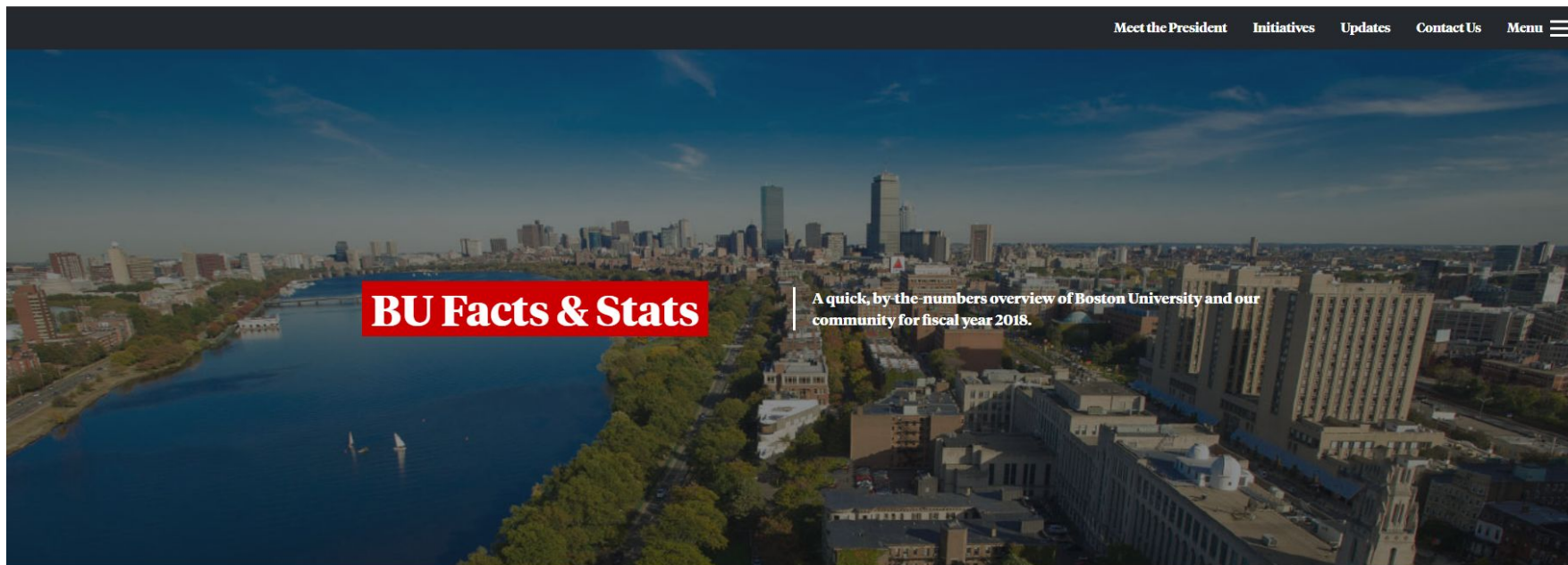
# Coleta de Dados

- Passos comuns de execução de Scraping
  1. Encontre uma URL de interesse
  2. “Importe” o HTML para o python através de bibliotecas como Requests e BeautifulSoup
  3. Use o recurso “Developer Tools” do navegador (F12 Chrome e Firefox, Ctrl+Shift+C Opera)
  4. Encontre as tags HTML que possuem as informações desejadas
  5. Utilize comandos das bibliotecas para extrair os dados a partir das tags desejadas
  6. (Opcional) Faça uso dos dados extraídos através de cálculos e análises
  7. (Opcional) Salve os dados extraídos ou resultados de análises em um formato conhecido

# Coleta de Dados

- Ex. (Scraping): <http://www.bu.edu/president/boston-university-facts-stats/>

Boston University Office of the President



# Coleta de Dados

- Ex. (Scraping): <http://www.bu.edu/president/boston-university-facts-stats/>

## Community

Student Body	34,262
Living Alumni	377,900+
Total Employees	10,182
Faculty	4,021
Non-Degree Students	2,232
Graduate & Professional Students	15,238
Undergraduate Students	16,792

## Campus

Classrooms	544
Buildings	310+
Laboratories	2,326
Libraries	21
Campus Area (acres)	134

## Academics

Study Abroad Programs	90+
Average Class Size	27
Faculty	4,021
Student/Faculty Ratio	10:1
Schools and Colleges	17
Programs of Study	300+



# Coleta de Dados

- Ex. (Scraping): <http://www.bu.edu/president/boston-university-facts-stats/>

```
1 import requests
2 from urllib import request, response, error, parse
3 from urllib.request import urlopen
4 from bs4 import BeautifulSoup
```

```
1 url = "http://www.bu.edu/president/boston-university-facts-stats/"
2 html = urlopen(url)
3 soup = BeautifulSoup(html, "lxml")
4 title = soup.title
5 titleText = title.get_text()
6 print(titleText)
```

BU Facts & Stats | Office of the President

```
▼<head>
  <title>BU Facts & Stats | Office of the President</title>
```

# Coleta de Dados

- Ex. (Scraping):

Community	
Student Body	34,262
Living Alumni	377,900+
Total Employees	10,182
Faculty	4,021
Non-Degree Students	2,232
Graduate & Professional Students	15,238
Undergraduate Students	16,792



```
body id="top" class="page-template page-template-10-25 has-du-branding has-du-  
masterplate 1-default sidebarLocation-right page-template-facts-stats"> == $0  
<header class="masthead" role="banner">...</header>  
  <div class="wrapper">  
    <div class="content">  
      <div class="bannerContainer bannerContainer-windowWidth bannerType-  
image">...</div>  
      <div class="content-container">  
        <!-- Intro Banner -->  
        <article id="post-23">  
          <h1>  
            </h1>  
          <section class="facts-stats">...</section>  
          <section class="facts-categories">  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
            <div class="facts-wrapper">...</div>  
          </section>  
          <section class="news">...</section>  
        </article>  
        ::after  
      </div>  
    <!-- .content-container -->  
  </div>  
  <!-- .content -->  
  <!-- .wrapper -->  
</body>  
<!-- footer class="siteFooter has-info-links has-branding" role="contentinfo">...</footer>  
<script type="text/javascript">...</script>  
<script type="text/javascript" src="http://www.bu.edu/president/wp-content/  
themes/r-president/js/script.min.js?ver=1.0.1"></script>  
<script type="text/javascript" src="http://www.bu.edu/president/wp-  
include/js/wp-embed.min.js?ver=4.0.10"></script>
```



# Coleta de Dados

- Ex. (Scraping): observem as tags HTML

```
1 section = soup.find_all('section', class_='facts-categories')
2
3 for elemen in section:
4     wrappers = elemen.find_all('div')
5     for x in wrappers:
6         title = x.find('h5').get_text()
7         print(title)
8         detail = x.find_all('ul')
9         for row in detail:
10             for l in row.find_all('li'):
11                 text = l.find('p').get_text()
12                 value = l.find('span', class_='value').get_text()
13                 print(text + value)
14             print("-----")
```

```
Community
Student Body34,262
Living Alumni377,900+
Total Employees10,182
Faculty4,021
Non-Degree Students2,232
Graduate & Professional Students15,238
Undergraduate Students16,792
```

```
-----
Campus
Classrooms544
Buildings310+
Laboratories2,326
Libraries21
Campus Area (acres)134
-----
```

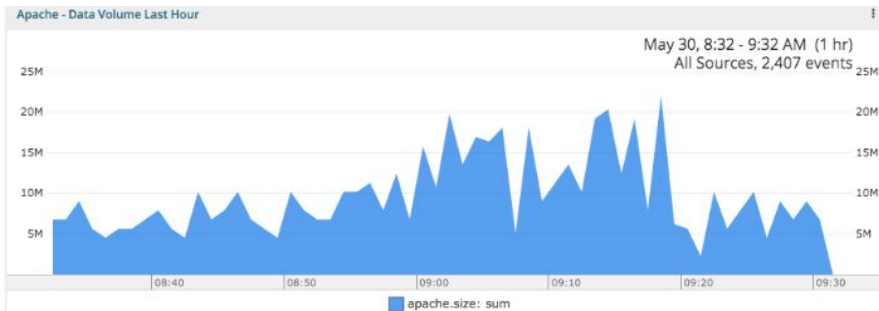
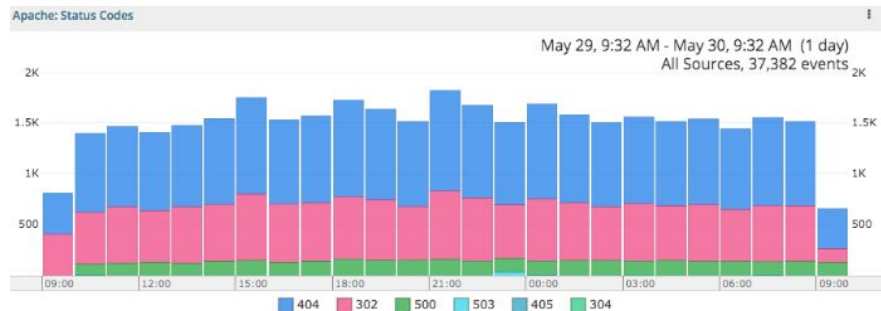
```
▶<section class="facts-stats">...</section>
▼<section class="facts-categories">
  ▼<div class="facts-wrapper"> == $0
    ▼<h5>
      ::before
      "Community"
      ::after
    </h5>
    ▼<ul class="row list">
      ▼<li class="list-item">
        <p class="text">Student Body</p>
        <span class="value">34,262</span>
      </li>
```

# Coleta de Dados

- Ex. (Scraping):
  - Esse exemplo foi extraído e modificado de:  
<https://levelup.gitconnected.com/quick-web-scraping-with-python-beautiful-soup-4dde18468f1f>
  - Código Fonte: <https://drive.google.com/open?id=1jzovVRXNldgzWk1BvjksH9UV9sNgsZd4>
  - Explore o código:
    - Espalhem comandos “print” para entenderem melhor cada passo
    - Testem outros comandos do BeautifulSoup
    - Entrem no site alvo do scraping e usem o “*Developer Tools*” do seu navegador para visualizar outras tags HTML do site. Extraiam outros conteúdos do mesmo para se acostumarem com a ferramenta: <http://www.bu.edu/president/boston-university-facts-stats/>

# Coleta de Dados

- Logging:
  - Funcionários e alunos possuem acesso interno a dados de empresas/laboratórios/sites
  - Você pode ser dono e origem dos dados (ex.: sites próprios, uso de câmeras e sensores)
  - Palavras-chave: Weblogs, sensing devices, Internet of Things (IoT)
  - Para o design de qualquer sistema para Logging:
    - O sistema deve exigir o mínimo de manutenções
    - Guardar o máximo de informações possíveis a cada acesso
    - Usar formatos apropriados para armazenamento e compartilhamento de dados



# Coleta de Dados

- Ex. (Logging):
  - Opções Gratuitas, como o Grafana, podem ser empregadas para facilitar análises de Logging





# Coleta de Dados

- Ex. (Logging): Logs do Netlab <https://drive.google.com/open?id=1cTIDeFjWLftjF7ZGjDKbqh7Clte7eUDk>
  - Pode ser lido como data frame Pandas, devendo ser feitos ajustes de formatação (e.g., eliminação de colchetes, separação de datas num campo data, etc.)
  - Estatísticas do log podem ser estimadas (e.g., média de duração dos acessos)

```
201.179.162.179 - - [17/Sep/2019:06:30:49 -0300] "teSubmit=Save" 400 0 "-" "-"
201.179.162.179 - - [17/Sep/2019:06:30:49 -0300] "POST /cgi-bin/ViewLog.asp HTTP/1.1" 404 0
 "-" "Ankit"
80.95.44.9 - - [17/Sep/2019:06:31:55 -0300] "GET / HTTP/1.1" 200 12101
"http://netlab.ice.ufjf.br/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"
50.31.26.18 - - [17/Sep/2019:06:32:14 -0300] "GET /wp-login.php HTTP/1.1" 200 1514 "-"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
50.31.26.18 - - [17/Sep/2019:06:32:14 -0300] "POST /wp-login.php HTTP/1.1" 200 1897 "-"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
50.31.26.18 - - [17/Sep/2019:06:32:15 -0300] "POST /xmlrpc.php HTTP/1.1" 200 420 "-"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
37.115.205.210 - - [17/Sep/2019:06:33:36 -0300] "GET /index.php/2016/09/29/ola-mundo/
HTTP/1.0" 404 10566 "http://netlab.ice.ufjf.br/index.php/2016/09/29/ola-mundo/"
"Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87
Safari/537.36"
```



# Limpeza de Dados

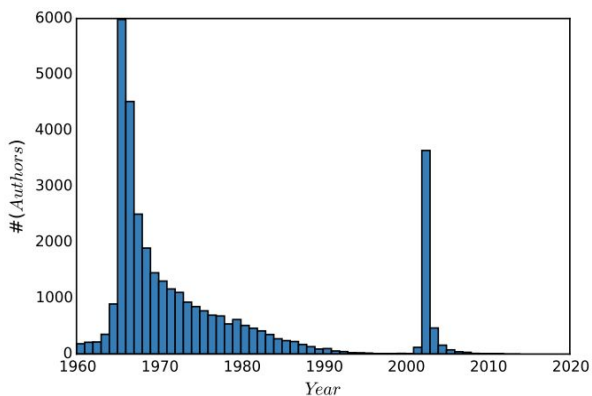
- Processo que antecede a análise e sucede o backup dos dados
- Artefatos de processamento:
  - erros sistemáticos oriundos do processamento da informação bruta
  - se detectados, podem ser corrigidos
  - sempre desconfie de algo inesperado/surpreendente

*“Surprising observations are what data scientists live for. Indeed, such insights are the primary reason we do what we do. But in my experience, most surprises turn out to be artifacts, so we must look at them skeptically”*

*SKIENA, S. S. The Data Science Design Manual. Springer, 2017.*

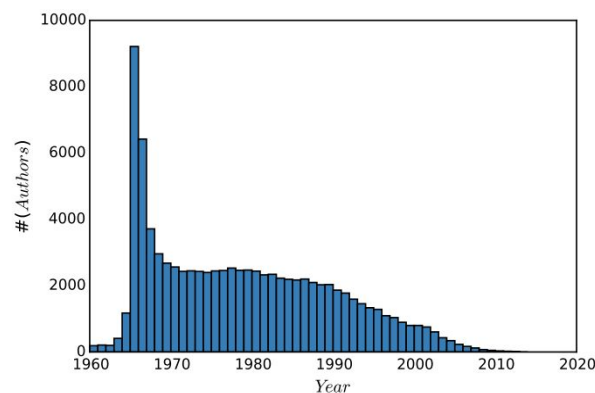
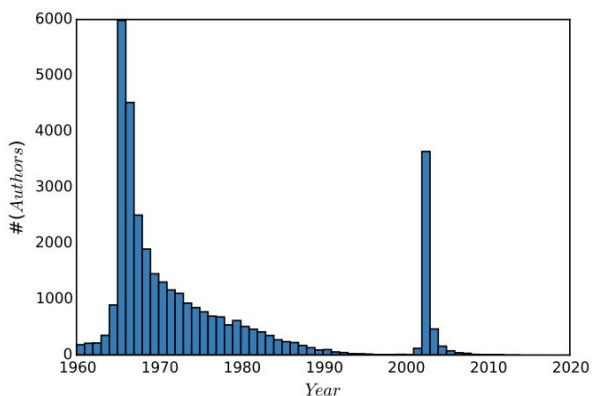
# Limpeza de Dados

- Detecção de artefatos: revisão dos dados, análise comportamental (curvas)
- A figura abaixo mostra quantos autores apareceram num dado periódico pela primeira vez em cada ano. Conseguem suspeitar de artefatos?



# Limpeza de Dados

- Detecção de artefatos: revisão dos dados, análise comportamental
- A figura abaixo mostra quantos autores apareceram num dado periódico pela primeira vez em cada ano. Conseguem suspeitar de artefatos?
  - Páginas 70 e 71 de “The Data Science Design Manual. Springer, 2017”.
  - Resumindo: mudança de registros/mudança de padrão de nomes nos registros.



# Limpeza de Dados

- Questões de compatibilidade:
  - Não faz sentido comparar coisas totalmente diferentes (e.g., unidades diferentes)
  - Lembrar, porém, do Coeficiente de Variação
  - Problemas que, geralmente, ocorrem quando conjuntos de dados distintos são mesclados

*“Review the meaning of each of the fields in any data set you work with. If you do not understand what’s in there down to the units of measurement, there is no sensible way you can use it.”*

*SKIENA, S. S. The Data Science Design Manual. Springer, 2017.*

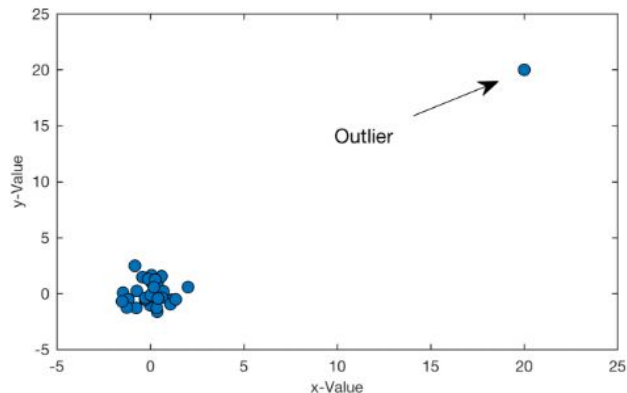
# Limpeza de Dados

- Solução para problemas de compatibilidade:
  - Conversão de unidades (e.g., passar todos os pesos para grama)
  - Conversão de representação numérica (e.g., evitar guardar números em strings)
  - Unificação de nomes (e.g., usar um padrão: last name/middle name/first name)
  - Unificação de datas (e.g., usar o formato americano YY/MM/DD)
  - Unificação financeira (e.g., conversão de todos os valores para dólar)

# Limpeza de Dados

- Detecção de Outliers:

- Valores que se distanciam consideravelmente dos demais numa dada amostra
- Geralmente causado por erros na entrada de dados ou no scraping
- Pode ser causado por questões de incompatibilidade
- Consequências: distorce média, variância, desvio padrão. Moda e mediana não sofrem muito
- Facilmente detectáveis, podem ser excluídos, mas seu motivo deve ser avaliado



# Limpeza de Dados

- Lidar com valores faltantes:
  - Observar campos que podem estar ausentes ou incompletos
  - Solução: pesquisar os melhores valores para preenchê-los
    - Ex.: O que fazer com um questionário de pesquisa deixado em branco ou preenchido com um outlier?

*“Separately maintain both the raw data and its cleaned version. The raw data is the ground truth, and must be preserved intact for future analysis. The cleaned data may be improved using imputation to fill in missing values. But keep raw data distinct from cleaned, so we can investigate different approaches to guessing.”*

SKIENA, S. S. *The Data Science Design Manual*. Springer, 2017.

# Limpeza de Dados

- Lidar com valores faltantes:

- Atribuição baseada em heurística:
  - Adivinhação coerente embasada no conhecimento sobre o assunto
- Atribuição baseada em média:
  - Conhecendo-se a média, escolhe-se um valor aproximado
- Atribuição baseada em valor aleatório:
  - Repetir um valor aleatório dentre os que já apareceram para aquela variável
- Atribuição baseada no vizinho mais próximo:
  - Quando a variância entre os elementos é pequena, é uma solução interessante
- Atribuição baseada por interpolação
  - Métodos de regressão para predição (e.g., regressão linear; serão vistos futuramente)

Variable

1	2	3
3	-	1
5	9	10
1	3	-
9	0	4
-	-	6
3	4	9
8	-	-
-	3	2
1	-	-
8	10	9



# Crowdsourcing

*“Crowdsourcing harnesses the insights and labor from large numbers of people towards a common goal. It exploits the wisdom of crowds, that the collective knowledge of a group of people might well be greater than that of the smartest individual among them.”*

SKIENA, S. S. *The Data Science Design Manual*. Springer, 2017.



# Crowdsourcing

- Ex. 01: Quantas moedas existem no pote?
  - Respostas de um primeiro grupo:

537, 556, 600, 636, 1200, 1250, 2350, 3000, 5000, 11,000, 15,000

**Mediana: 1250      Média: 3759      Valor Correto: 1879**

Conclusões: mediana mais próxima do que qualquer palpite isolado; grande média devido à presença de outliers



# Crowdsourcing

- Ex. 01: Quantas moedas existem no pote?
  - Respostas de um primeiro grupo:

537, 556, 600, 636, 1200, 1250, 2350, 3000, 5000, 11,000, 15,000

**Mediana: 1250      Média: 3759      Valor Correto: 1879**

Conclusões: mediana mais próxima do que qualquer palpite isolado; grande média devido à presença de outliers

- Respostas do segundo grupo, após observarem as do primeiro:

750, 750, 1000, 1000, 1000, 1250, 1400, 1770, 1800, 3500, 4000, 5000

**Mediana: 1325      Média: 1925      Valor Correto: 1879**

Conclusões: média e mediana próximas do valor correto; remoção de outliers observados nas respostas do primeiro grupo.



# Crowdsourcing

- Quando dados extraídos de grupos são relevantes?
  - Quando as opiniões são independentes
    - Muitas vezes, dados enviesados atrapalham o experimento
  - Quando o grupo é composto por pessoas com diferentes conhecimentos e metodologias
    - Um comitê composto por pessoas com opiniões semelhantes não “enriquece” o debate
  - Quando o tema não exige conhecimento especializado
    - É mais difícil confiar em respostas de grupos aleatórios para temas complexos
  - Quando opiniões podem ser razoavelmente agregadas
    - Questões muito genéricas podem levar a respostas excessivamente variadas

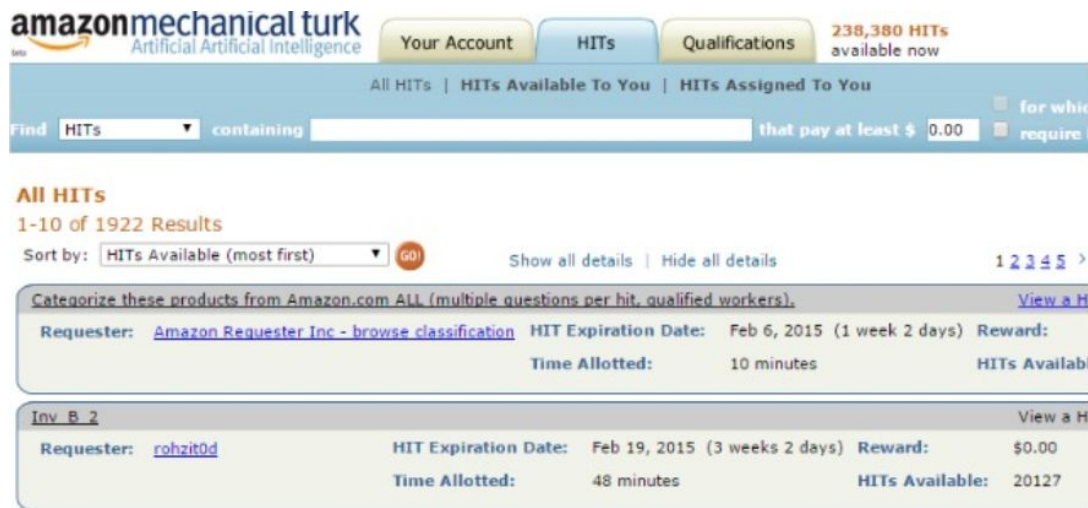
# Crowdsourcing

- Mecanismos de agregação:
  - Uso de distribuições estatísticas (assunto de aulas futuras)
  - Uso de dados como média e mediana (mediana é mais recomendado por causa de outliers), de preferência associados a medidas de dispersão (e.g., desvio padrão)
  - Remoção de outliers
  - Votação é um mecanismo popular de agregação estatisticamente validado
    - Condorcet Jury Theorem: a probabilidade  $P(n)$  do resultado de uma votação estar correto tende a 1 à medida em que o número  $n$  de votantes aumenta, mesmo em disputas acirradas onde  $p = 51\%$  é a porcentagem de votos do grupo vencedor

$$P(n) = \sum_{i=(n+1)/2}^n \binom{n}{i} p^i (1-p)^{n-i}$$

# Crowdsourcing

- Serviços de Crowdsourcing:
  - Serviços como Amazon Turk e CrowdFlower intermediam uma relação paga entre pesquisadores e pessoas para executar serviços não-automatizáveis
    - Ex.: prover dados para aprendizado de máquina, experimentos psicológicos/econômicos



The screenshot displays the Amazon Mechanical Turk dashboard. At the top, the logo "amazonmechanical turk" is visible, along with navigation tabs for "Your Account", "HITS", and "Qualifications". A status bar indicates "238,380 HITS available now". Below this, a search bar allows filtering by "HITS" containing a specific term, with a minimum payment of \$0.00. The main section, titled "All HITS", shows "1-10 of 1922 Results". A dropdown menu is set to "Sort by: HITS Available (most first)". Two HIT listings are visible:

Categorize these products from Amazon.com ALL (multiple questions per hit, qualified workers).			
Requester:	<a href="#">Amazon Requester Inc - browse classification</a>	HIT Expiration Date:	Feb 6, 2015 (1 week 2 days)
		Reward:	
		Time Allotted:	10 minutes
		HITS Available:	

Inv B 2			
Requester:	<a href="#">rohjit0d</a>	HIT Expiration Date:	Feb 19, 2015 (3 weeks 2 days)
		Reward:	\$0.00
		Time Allotted:	48 minutes
		HITS Available:	20127

# Crowdsourcing

- Gamificação, uma alternativa aos serviços pagos:
  - “Jogos” com propósito de coleta de dados
    - CAPTCHA: além de garantir acessos, respostas são mapeadas para melhorar a digitalização de documentos arquivísticos
    - Testes de QI em jogos e aplicativos: fontes de dados psicológicos/psiquiátricos
    - Jogos do tipo “que personagem você seria”, comuns no Facebook



# AAG02 Tarefa em Dupla

- Utilizar Spidering/Scraping ou Logging para coletar dados a serem passados como parâmetro para uma das funções feitas na AAG01. Em outras palavras, os dados coletados deverão ser “limpados” e passados como parâmetro para a geração de uma CDF **ou** PMF.
- Regras:
  1. Deve ser feito obrigatoriamente no Jupyter Notebook
  2. Markdowns com comandos LaTeX para documentar código e explicar resultados
  3. Deve ser entregue nos formatos .pdf e .ipynb (código fonte+markdowns)
  4. O dados **dados brutos** obtidos por spidering/scraping ou logging devem ser salvos separadamente do **dado tratado**, caso alguma limpeza tenha que ser realizada
  5. Os dados devem ser entregues em anexo