

Relatório Final - MAC213

Projeto : Desenvolvimento de tutoriais de manuseio e análise de dados

Guilherme Yambanis Thomaz - 8041265

Objetivos

Com o intuito de fortalecer a transparência, a SME tem publicado e atualizado um conjunto de bases de dados referentes à educação municipal, variando de planilhas a microdados. O acesso a tais bases de dados pode ser, muitas vezes, limitado, devido à falta de conhecimento para manuseio das bases do público interessado (jornalistas, pesquisadores, integrantes de conselhos escolares). Frente a esse desafio, o Pátio Digital lançou o “ABC Dados”, que consiste na publicação de tutoriais que visam a orientar, numa linguagem didática e fluida, metodologias de manuseio das bases.

Tarefas

Após reunião realizada no dia 31 de agosto com Priscilla, na sede do Pátio Digital, definiu-se que os temas dos tutoriais seriam Excel e Python, e que o controle de horas seria feito através da ferramenta Toggl.

Os dez tutoriais a serem realizados ficaram igualmente divididos entre Excel e Python.

Os cinco tutoriais de cada plataforma foram estruturados da seguinte forma:

Tutoriais Excel

Tutoriais acessíveis, sobre o Excel, para manipulação de dados, em especial arquivos csv.

Tutorial 1

- O que é uma planilha?
- Morfologia básica
- Importando conjunto de dados

Tutorial 2

- Introdução a funções
- Função SOMA
- Função CONT.SE
- Função SE

- Função SOMASE

Tutorial 3

- Criando um gráfico de pizza simples
- Customizando o gráfico

Tutorial 4

- Google Sheets
- Importando csv
- Usando funções no Google Sheets
- Criando uma tabela dinâmica no Google Sheets
- Criando um gráfico a partir dos dados
- Função FREQUÊNCIA

Tutorial 5

- Criando uma tabela Dinâmica
- Manipulação e visualização dos dados

Tutoriais Python

Tutoriais criado na plataforma Jupyter Notebook.

Tópicos abordados:

Tutorial 1

- Noções elementares de Python

Tutorial 2

- Pacote Pandas

Tutorial 3 e 4

- Criando gráficos com Matplotlib

Tutorial 4

- Regex

Tutorial 5

- Definindo funções

Progresso e desafios

Excel

Definir inicialmente que os tutoriais seriam escritos em Markdown, por ser um formato de fácil exportação, seja para plataforma web, seja para pdf, além da possibilidade de inserção de links e gifs.

A construção e definição dos tutoriais tomou um certo tempo, porém, o maior desafio se mostrou na criação dos gifs, que demandaram diversas tentativas com ferramentas diferentes.

Além disso, após a parte técnica finalizada, escrever o texto de forma enxuta e didática também se mostrou algo desafiador. Grande parte da carga horária foi dedicada para refacção e reescrita, a fim de maior clareza e melhor didática.

Como o Excel depende da interação com a interface gráfica, foi necessário que o texto fosse bastante descritivo. Os gifs ajudaram muito nesse sentido, comunicando ações mais fáceis de serem traduzidas de maneira visual.

Python

Já para os tutoriais em Python, preferi o formato Notebook.

Iniciei os tutoriais usando a plataforma do Google, o Colab. Entretanto, após algum tempo, encontrei diversas dificuldades, como importações incompletas entre outras.

Isso tornou necessário migrar o projeto para o formato Jupyter Notebook. Como ele também é renderizado no próprio Github, não se mostrou uma perda muito grande nesse aspecto, além do ganho em robustez pela execução local do código.

A criação dos tutoriais em Python se mostrou mais desafiadora do ponto de vista técnico, exigindo muitas refacções e pesquisa sobre a linguagem.

Certas tarefas que propus nos tutoriais se mostraram mais complexas do que o previsto, mas isso permitiu não só que eu aprimorasse meus conhecimentos de Python, como também que eu criasse os tutoriais de um ponto de vista mais empático com o leitor, o que, acredito, melhorou a didática.

A junção do código com o texto em um só lugar, graças ao Jupyter Notebook, permitiu a criação de um tutorial muito mais intuitivo, já que o código e sua explicação estão lado a lado e o aluno pode executá-lo em tempo real.

Poster

O poster foi fixado no IME no dia 8 de novembro.

aluno responsável
GUILHERME YAMBANIS THOMAZ
supervisora
PRISCILLA CORRÊA, ASSESSORA TÉCNICA DA SME
disciplina
MAC 213 - Atividade Curricular em Comunidade
repositório
<https://github.com/yambanis/MAC213>

DESENVOLVIMENTO DE TUTORIAIS DE MANUSEIO E ANÁLISE DE DADOS

O QUE É O PÁTIO DIGITAL?

POLÍTICA DE GOVERNO ABERTO DA PREFEITURA
DE SÃO PAULO, ESTRUTURADA EM TRÊS EIXOS:

1. TRANSPARÊNCIA E DADOS ABERTOS
2. INOVAÇÃO TECNOLÓGICA
3. COLABORAÇÃO GOVERNO-SOCIEDADE

PROJETO

O PÁTIO DIGITAL LANÇOU O "ABC DADOS": TUTORIAIS QUE VISAM A ORIENTAR, NUMA LINGUAGEM DIDÁTICA E FLUIDA, METODOLOGIAS DE MANUSEIO DAS BASES DE DADOS REFERENTES À EDUCAÇÃO MUNICIPAL

DURANTE O SEMESTRE, FORAM DESENVOLVIDOS TUTORIAIS EM EXCEL E PYTHON, VISANDO A AUXILIAR, NO ACESSO E NA MANIPULAÇÃO DESSAS BASES DE DADOS, O PÚBLICO INTERESSADO QUE NÃO CONTA COM O CONHECIMENTO NECESSÁRIO PARA O SEU MANUSEIO

EXCEL

TUTORIAIS DESENVOLVIDOS EM
MARKDOWN, COM UTILIZAÇÃO
DE GIFS, PARA MELHOR DIDÁTICA

PYTHON

TUTORIAIS DESENVOLVIDOS EM
NOTEBOOK, PARA INCORPORAR
CÓDIGO E TEXTO EM UM ÚNICO
ARQUIVO

Resultado

O resultado final foram 10 tutoriais que servem para apresentar as duas plataformas, Python e Excel, de maneira amigável.

O intuito foi partir de um ponto inicial bastante básico, para que, independente do nível de conhecimento do leitor, fosse possível a utilização das ferramentas, sem uma barreira de entrada alta e intimidadora.

Os tutoriais iniciais são muito detalhados, sem pressupor nenhum conhecimento prévio.

Após os tutoriais iniciais de cada assunto, temas mais complexos começam a ser abordados de forma mais prática, o que foi possível graças à introdução aos *features* anteriormente.

É esperado que quem acompanhe todos os tutoriais seja capaz de sair de um total desconhecimento das ferramentas e possa criar análises de dados pertinentes com as técnicas adquiridas ou, ainda, possa utilizar esses tutoriais como ponto de partida para buscar informações mais específicas, já com certa familiaridade em relação às plataformas.

Acompanhamento

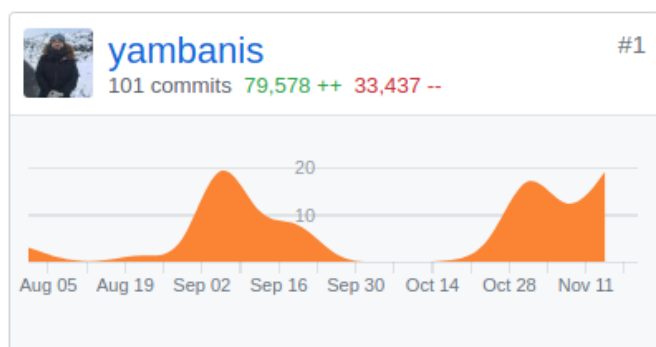
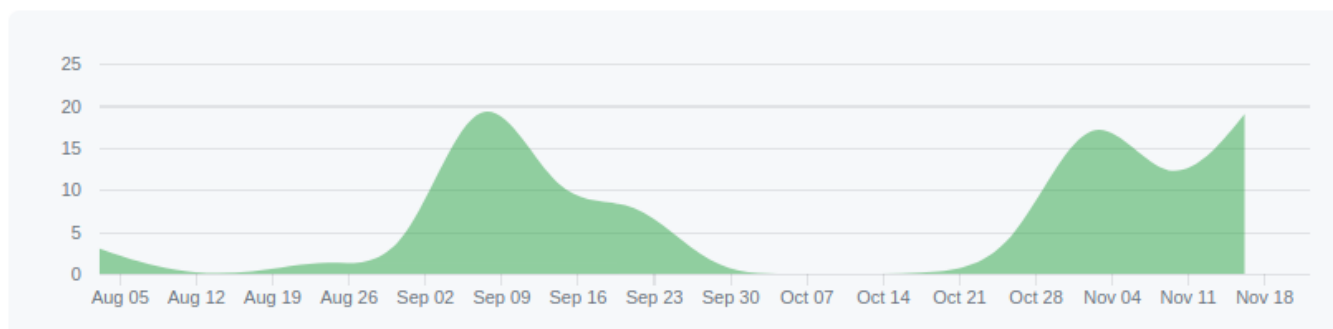
O acompanhamento se deu de duas formas principais:

- [Repositorio](#) aberto no Github, onde todas os arquivos foram salvos durante o semestre

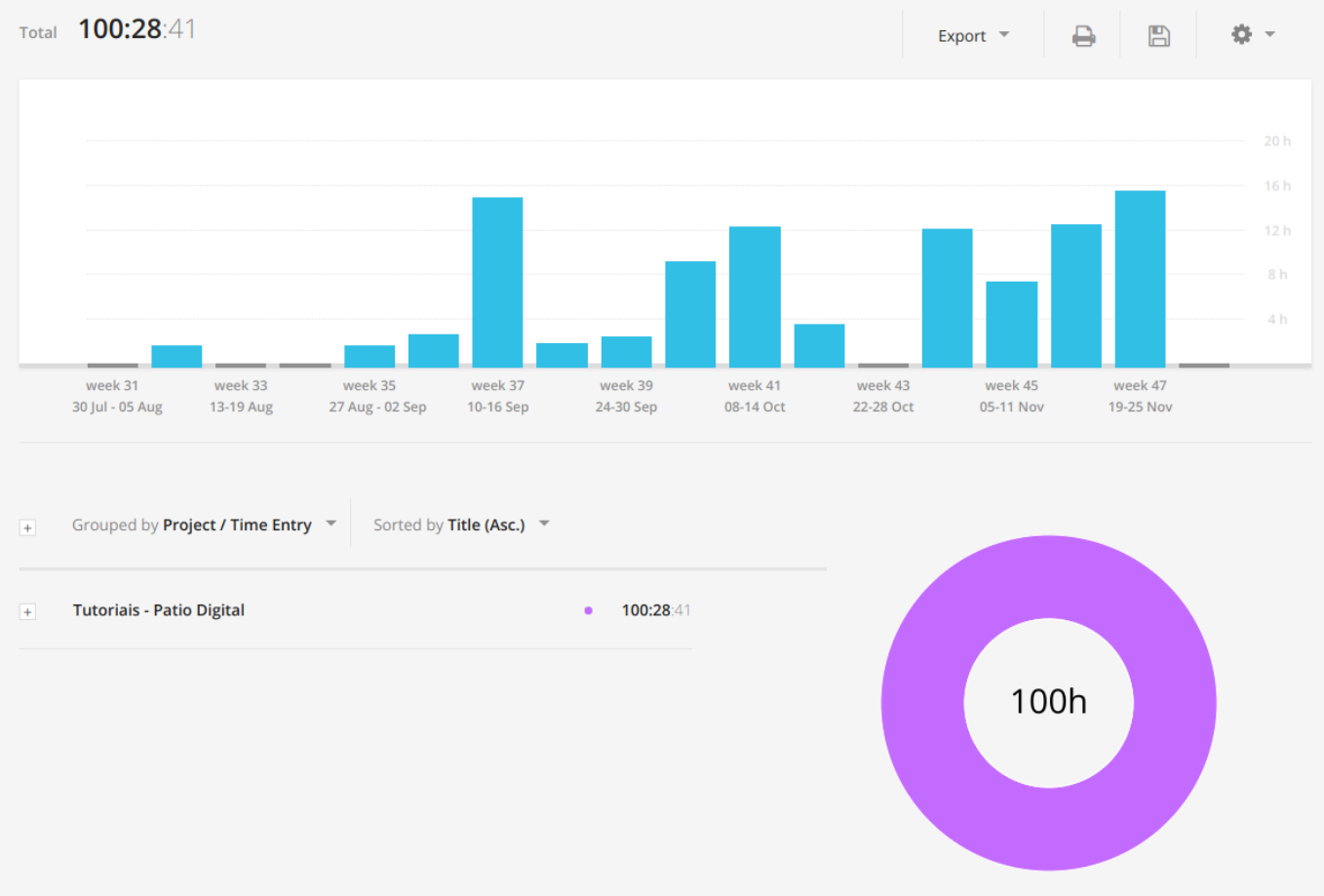
Aug 5, 2018 – Nov 24, 2018

Contributions: **Commits** ▾

Contributions to master, excluding merge commits

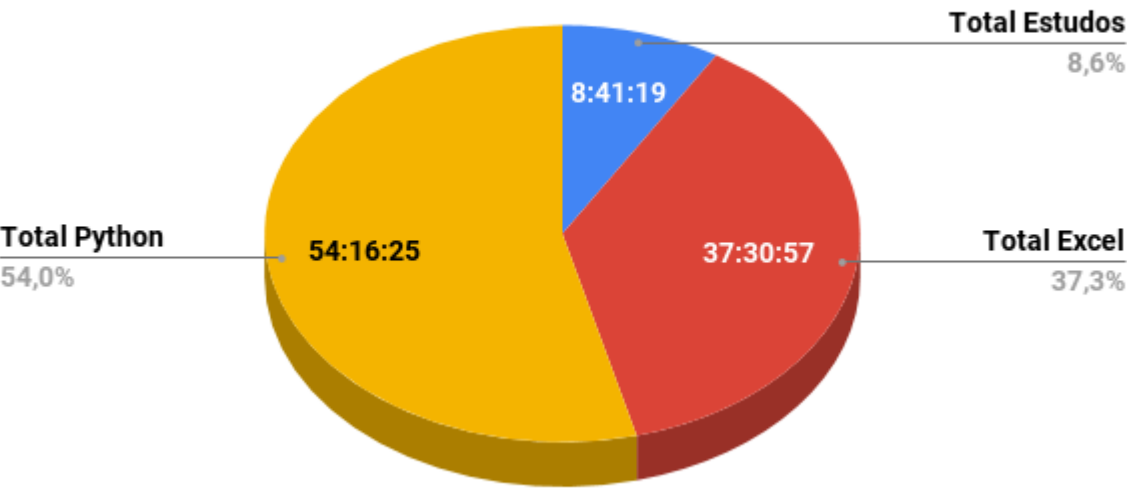


- Controle de horas realizado através da ferramenta Toggl



Horas despendidas no projeto

Duração das Atividades(24/11)



Horas despendidas (detalhado)

Estudos	Duração	Total Estudos	Excel	Duração	Total Excel	Python	Duração	Total Python
Criacao do Github, estudos iniciais	1:56:00	8:41:19	inicio tutorial 1	1:55:35	37:30:57	planejamento Python	0:55:28	54:16:25
Reuniao Patio Digital	2:00:00		Tutorial Excel	1:59:37		inicio tutorial Python	2:24:33	
Pesquisas iniciais - Topicos Excel	1:32:00		Tutorial funcoes	0:15:58		Continuacao tutorial Pandas	2:08:04	
Pesquisando Referencias - Tutoriais	1:29:00		Tutorial de formulas	2:16:33		Mais Pandas	2:12:00	
Pesquisando Topicos	1:44:19		tutorial GSheets	0:58:46		migrando para Jupyter	3:40:24	
			revisões	1:49:26		Python Md	4:12:17	
			Revisões	0:53:27		python	2:20:07	
			correcao orto	1:31:50		Tutorial Python - continuacao	2:17:48	
			checagem	0:29:59		Palestra Patio Digital	1:29:44	
			Refatorando	0:58:34		Cartaz	2:03:18	
			Refatorando	1:13:58		matplotlib	2:58:00	
			revisoes	0:57:23		Revisao Cartaz	1:05:07	
			arrumando gifs e tutorial funcoes	1:56:08		Python Grafico de barras	0:38:09	
			Conformando gifs	6:00:28		matplotlib	0:57:58	
			correcoes ortográficas e revisoes	4:53:55		Revisoes de texto	1:30:04	
			Revisao CountSe Excel	2:15:55		Modularizando Tutoriais	0:58:00	
			grafico de barras excel	1:17:40		revisao Python	0:29:19	
			Tutorial Tabela Dinamica	0:53:04		matplotlib Bar	0:00:10	
			Continuacao tutorial Tabela Dinamica	2:10:41		matplotlib Bar	3:13:50	
			Correcoes Finais	2:42:00		matplotlib Bar	2:16:21	
						Regex and Groupby	2:05:43	
						Modularizando Tutoriais	2:33:21	
						Correcoes Ortograficas	2:00:48	
						Mais correcoes	3:29:17	
						Revisao de Texto	2:49:31	
						Revisoes fianis Python	1:20:00	
						Exportar Python	0:21:27	
TOTAL GERAL	100:28:41					Novo Tutorial - funcoes	1:45:37	

Tarefas Realizadas (detalhado)

Irei detalhar as tarefas realizadas, utilizando as horas logadas na ferramenta Toggl. Algumas imagens meramente ilustrativas foram adicionadas, os tutoriais integrais podem ser visualizados no repositório, já que seria inviável colocá-los em pdf, principalmente pela impossibilidade da apresentação dos GIFs em PDF.

Criação do Github, estudos iniciais 1:56:00

Após a troca inicial de emails com Priscilla, criei o repositório que armazenou o projeto durante todo o semestre. Priscilla me passou algumas referências de tutoriais, que estudei e iniciei algumas pesquisas sobre tópicos a serem abordados, bem como familiarizei-me com o banco de dados da SME.

Reunião Patio Digital 2:00:00

Reunião inicial no pátio Digital, em sua sede no Pacaembu. Nela, ficou definido o assunto dos tutoriais, que seria Excel e Python.

Pesquisas iniciais - Tópicos Excel 1:32:00

Pesquisei forma e conteúdo de cursos e tutoriais de Excel.

Pesquisando Referências - Tutoriais 1:29:00

Continuação da pesquisa anterior, após ambas as pesquisas, consegui definir o formato dos tutoriais, com o uso de GIFs. Grande parte da decisão foi devida a tutoriais já existentes no site do Pátio Digital.

Pesquisando Tópicos 1:44:19

Elenquei Quais seriam os tópicos mais interessantes para serem abordados. Fiz um RoadMap, dos tópicos a serem abordados nos 5 tutoriais

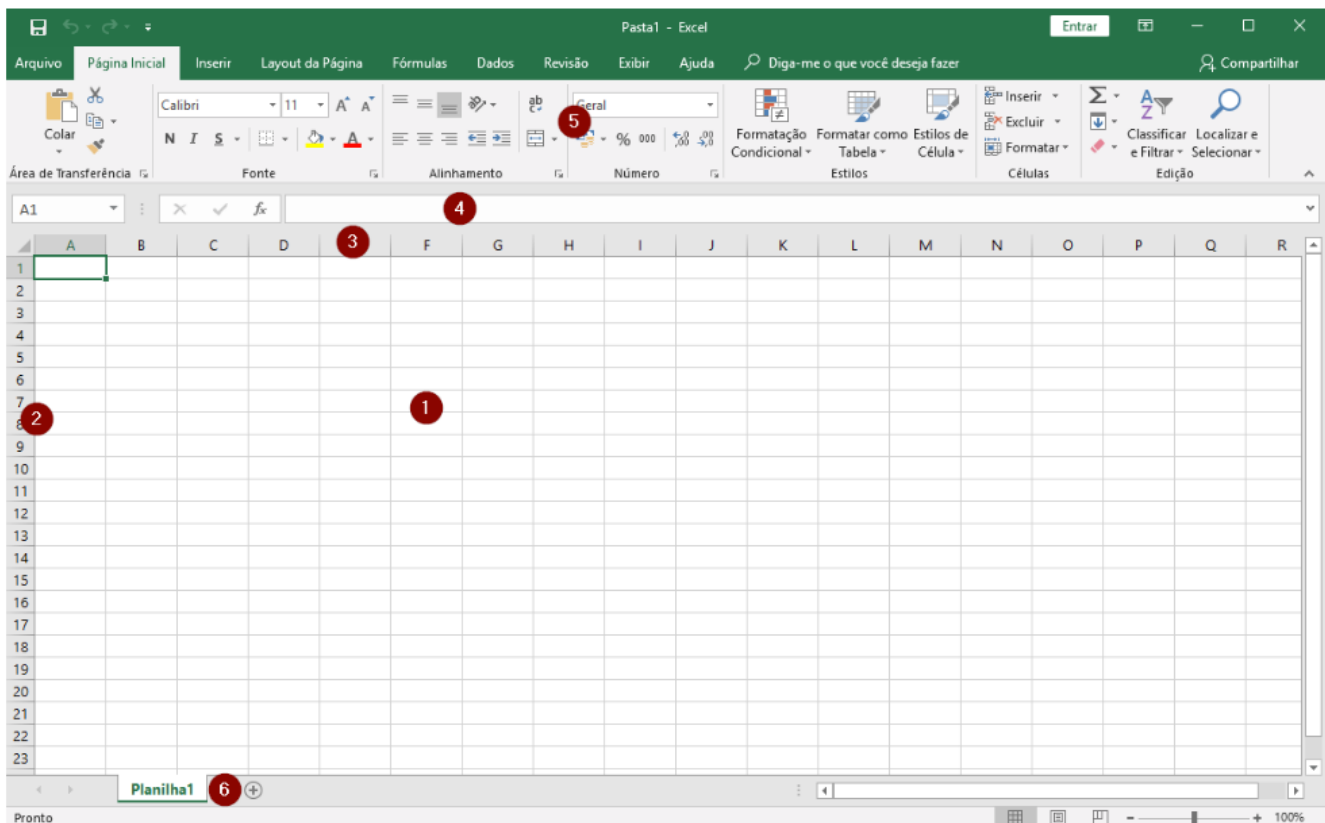
início tutorial 1 1:55:35

Dei início ao tutorial 1

O que é uma planilha?

A planilha - ou em inglês *spreadsheet* - é uma aplicação de software que tem como finalidade salvar e organizar dados em linhas e colunas.

Usaremos a versão 2016 do Excel neste tutorial, mas tudo deve ser equivalente para outras versões.



Alguns dos elementos presentes na tela inicial do Excel.

1. Células.
2. Linhas - indicadas por números.
3. Colunas - indicadas por letras.

Tutorial Excel 1:59:37

Continuação do Tutorial 1 e início do segundo tutorial

Fórmulas

Agora que já temos nossos dados importados, começaremos a extração de informações.

Lembre-se de que na [página](#) em que fizemos o download do CSV "Perfil dos educandos - Cor/raça, idade, sexo, necessidades educacionais especiais", temos o arquivo "Dicionário", que será importante para a compreensão das siglas presentes no nosso CSV.

Tópicos a serem abordados:

- Introdução a funções
- Função SOMA
- Função SE
- Função SOMASE

Introdução a funções

As funções no excel são uma das maneiras mais simples de automatizar processos, o que é muito útil, seja no tratamento de pequenos conjuntos de valores até tabelas bastante grandes.

Introduzindo valores

Para introduzir um valor no excel, basta clicar na célula em questão e digitar o valor que quisermos. Aqui, usamos o termo valor de forma genérica: pode ser uma quantia em dinheiro, uma data, um nome. O excel é capaz de tratar todos esses tipos de dados. Vejamos como informar o que está na célula do excel mais à frente.

	A	B	C	D	E	F	G	H
1	Valores							
2								
3								
4								
5								
6								

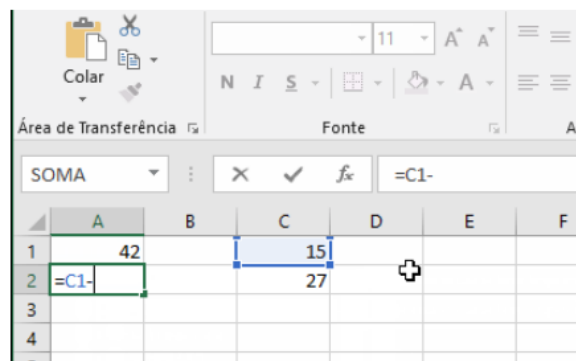
Tutorial funções 0:15:58

Dei início ao segundo tutorial

Fórmulas básicas

Podemos fazer operações básicas no excel usando os operadores conhecidos da matemática: +, -, * e /. No excel, fazemos referência às células pela letra de sua coluna, seguida do número de sua linha. Assim, podemos selecionar uma célula para guardar o resultado da operação entre outras células.

1. Para somar, escolha uma célula que guardará o resultado e digite =C1+C2, por exemplo, em seguida, pressione ENTER.
2. Para somar, escolha uma célula que guardará o resultado e digite =C1-C2, por exemplo, em seguida, pressione ENTER.
3. Para somar, escolha uma célula que guardará o resultado e digite =C1*C2, por exemplo, em seguida, pressione ENTER.
4. Para somar, escolha uma célula que guardará o resultado e digite =C1/C2, por exemplo, em seguida, pressione ENTER.



Algumas funções úteis

Função SOMA

Esta é uma das funções mais básicas e úteis do excel.

Utilizaremos esta [base de dados](#) para calcular o número de matriculados por distrito.

- Primeiro, adicionamos um cabeçalho, indicando que aquela coluna faz referência ao total de matrículas.
- Em seguida, digitamos =SOMA e selecionamos a função SOMA que o excel sugere, com o mouse.

Tutorial de fórmulas 2:16:33

Continuação do item anterior

Preencher

Agora, aplicaremos a mesma fórmula SOMA para todos os distritos.

- Primeiro, clicamos na **CAIXA DE NOME** no canto superior esquerdo e selecionamos o nome que acabamos de definir, Matrículas.
- Agora, selecionamos o símbolo referente ao comando preencher.
- Na caixa que se abriu, selecionamos "para baixo".

Pronto! Nossa fórmula de soma acabou de ser aplicada para todos os distritos!

The screenshot shows the Microsoft Excel interface with the 'Formulas' ribbon selected. A 'Novo Nome' (New Name) dialog box is open, displaying the name 'Matrículas' and the formula '=evolucaodemanda!\$J\$2'. The background spreadsheet shows a table with columns for districts (A) and various matriculation metrics (B-I). The formula bar at the top displays '=SOMA(B2:I2)'.

DISTRITO	MAT_CRE	MAT_PRE	MAT_FUN	MAT_FUN	MAT_EJA	MAT_EJA_II	MAT_ENS_MED	MAT_ED_PROF	Total de Matrículas	Total de Matriculados
AGUA RASA	573	1.929	1.344	1.376	219	137	0	0	5.578	
ALTO DE PINHEIROS	236	198	0	0	0	0	0	0		
ANHANGUERA	207	1.947	2.943	2.673	187	856	0	0		
ARICANDUVA	435	1.675	2.429	2.377	187	670	0	0		
ARTUR ALVIM	901	4.595	3.216	2.784	201	837	0	0		
BARRA FUNDA	220	335	0	0	0	0	0	0		
BELA VISTA	493	1.118	516	519	0	632	0	0		
BELEM	154	1.619	462	0	0	0	0	0		
BOM RETIRO	452	1.643	0	0	0	0	0	0		
BRAS	199	600	0	0	0	0	0	0		
BRASILANDIA	2.121	9.069	8.966	7.005	935					
BUTANTA	231	1.636	1.635	1.515	411					
CACHOEIRINHA	909	4.602	4.471	3.576	576					
CAMBUCI	131	786	0	0	316					
CAMPO BELO	336	782	1.492	1.350	210					
CAMPO GRANDE	218	2.310	3.486	3.305	468					
CAMPO LIMPO	1.286	9.673	9.669	8.696	1.193					
CANGAIBA	750	4.618	2.843	2.486	221					
CAPAO REDONDO	902	7.652	11.599	9.209	1.220					
CARRAO	506	1.745	1.068	1.028	0					
CASA VERDE	403	2.451	1.145	1.169	143					
CIDADE ADEMAR	981	6.134	4.935	4.132	749					
CIDADE DUTRA	2.421	6.612	6.624	6.381	433	2.600	0	0		
CIDADE LIDER	822	5.232	3.801	3.631	476	2.286	0	0		
CIDADE TIRADENTES	1.676	11.776	13.221	12.656	735	3.511	557	25		

tutorial GSheets 0:58:46

Decidi criar um tutorial sobre o Google Sheets, por ser um software livre e online

Alternativas ao Excel

Tópicos a serem abordados

- Google Sheets
- Importando .csv
- Usando funções no Google Sheets
- Criando uma tabela dinâmica no Google Sheets
- Criando um gráfico a partir dos dados
- Função FREQUÊNCIA

Google Sheets

O Google Sheets é uma ferramenta muito poderosa e, o melhor de tudo, grátis!

Ela pode ser acessada pelo navegador, editada por múltiplos usuários e os seus arquivos ficam salvos na nuvem, podendo ser acessados por qualquer computador, através da sua conta do Google.

Acessando o Google Sheets

- Acesse [o site do Google Sheets](#), logue com sua conta do Google.
- Crie uma nova planilha em branco

Importando .csv

- Faça o download do arquivo "[Instituições Credenciadas - Abr/2018](#)"
- Clique em Arquivo > Importar > Upload
- Arraste o arquivo osccredenciamentoabr18.csv da pasta de downloads para a janela do Google Sheets
- Aguarda a realização do upload
- Selecione **Inserir novas náminas**

revisões 1:49:26

A quantidade de texto dos tutoriais requereu diversas revisões, tanto gramática e ortografia, como forma e clareza. Por se tratarem de tutoriais pressupondo nível de conhecimento bastante básico, era extremamente necessária clareza nos comandos e explicações. Existiram Diversas revisões necessárias, seja por auto crítica ou por feedbacks fornecidas pela supervisora. Era imprescindível que não houvessem erros nos tutoriais para que os mesmo possam ser publicados.

Revisões 0:53:27

Vide revisões

correção orto 1:31:50

Vide revisões

checagem 0:29:59

Mudanças na função SOMA para ser relevante com a base de dados da SME, baseado em Feedback da supervisora

Refatorando 0:58:34

Organização das pastas do Github, modularizando os tutoriais e mudança dos links dos GIFS

Refatorando 1:13:58

Vide acima

revisões 0:57:23

Vide revisões

arrumando gifs e tutorial funções 1:56:08

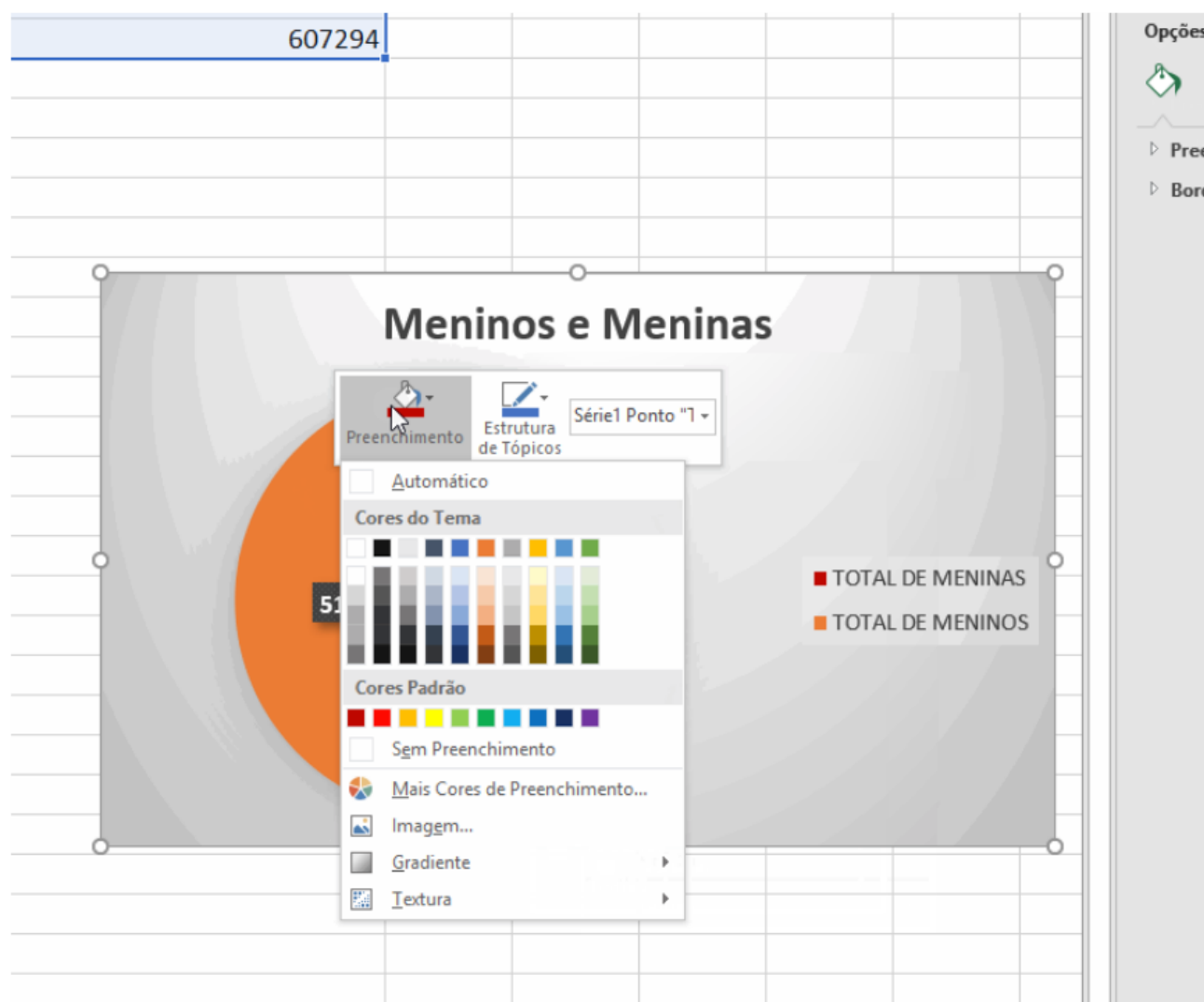
Os gifs dos tutoriais do Google Sheets continham informações pessoais em alguns frames. Tiveram que ser refeitos em sua totalidade

Conformando gifs 6:00:28

O programa que utilizei para gravar os gifs gerou problemas de compressão. Os gifs tiveram que ser refeitos para manter maior fidelidade.

Finalmente, podemos escolher as cores que serão utilizadas em nossos setores.

- Clique com o botão direito sobre um setor, selecione a opção preenchimento e escolha a cor desejada



correções ortográficas e revisões 4:53:55

Vide revisões

Revisão CountSe Excel 2:15:55

Função COUNTSE estava conceitualmente confusa, atrapalhando aprendizado.

Conforme aparece na tela, o comando CONT.SE recebe dois argumentos:

```
=CONT.SE (Intervalo; Critérios)  
=CONT.SE (Onde você quer procurar?; O que você quer procurar?)
```

Queremos procurar quantas vezes "Brooklin Paulista" está presente na coluna "Bairro da planilha. Então, vamos inserir esses dois argumentos na nossa fórmula.

Depois de inserir a CONT.SE na célula, selecionaremos agora os argumentos.

Primeiro, selecionaremos as células que nos interessam. Nesse caso, as informações estão na coluna 'Bairro'. Queremos selecionar a coluna em sua totalidade.

- Vamos clicar na planilha "osccredenciamentoabr18" e clicar sobre o "H" que marca a coluna.

Automaticamente, a fórmula que estamos criando, escrita na barra de fórmula, já foi preenchida com o argumento "'osccredenciamentoabr18 (1)'!H:H", que indica que desejamos analisar a coluna H da planilha "osccredenciamentoabr18" em sua totalidade.

Agora, precisamos preencher o segundo argumento da função, que, nesse caso, é o que queremos contar.

- Clicamos na barra de fórmulas e adicionamos um ponto e vírgula.

O ; indica que vamos agora inserir o próximo argumento. Nesse caso, queremos contar o número de ocorrências de "BROOKLIN PAULISTA"

- Inserimos BROOKLIN PAULISTA aspas, assim o excel contará todas as ocorrências dessa frase na coluna H.
- Colocamos então um fecha parênteses no fim da fórmula e damos enter.

O valor que aparece, 4, mostra que existem 4 ocorrências de "BROOKLIN PAULISTA" na coluna Bairro.

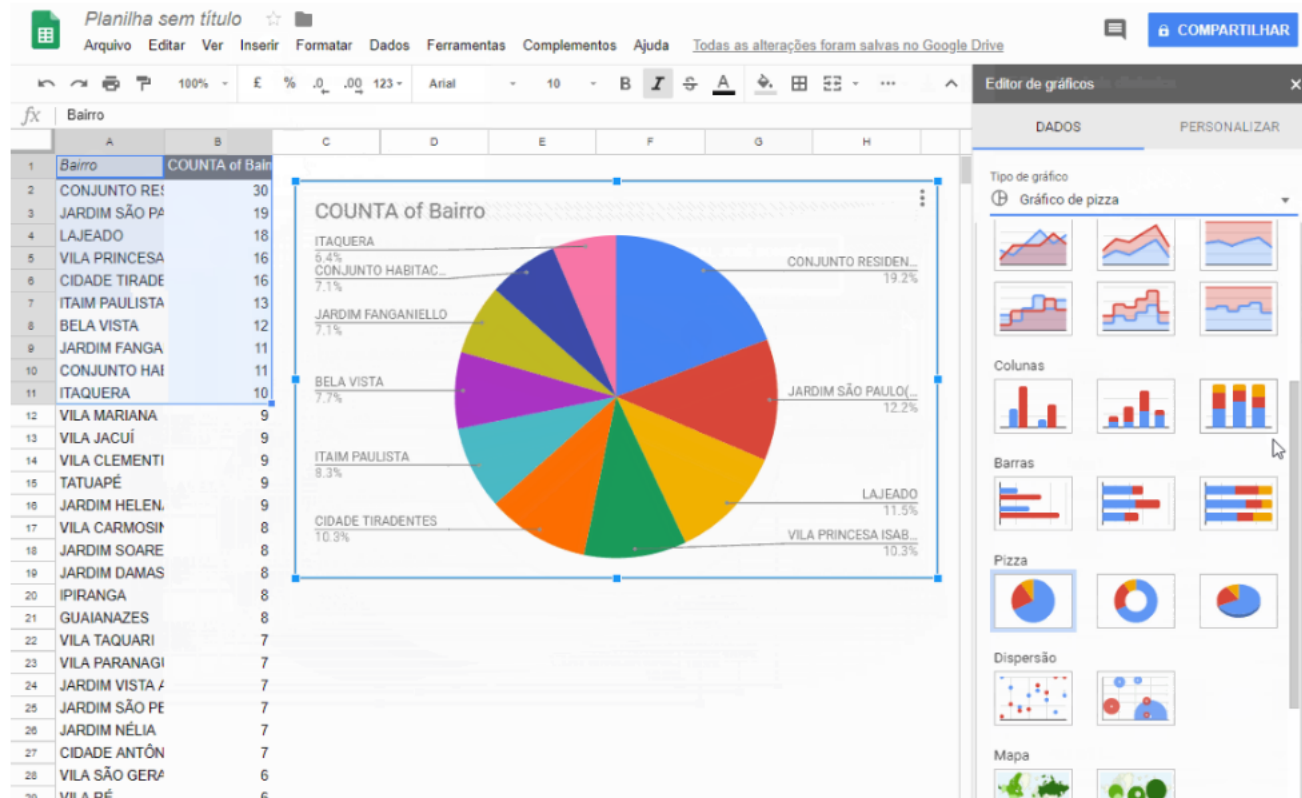
gráfico de barras excel 1:17:40

Continuação e melhoria do tutorial sobre gráficos

Criando um gráfico a partir dos dados

Criaremos um gráfico, mostrando quais bairros têm, pelo menos, 10 organizações cadastradas ou mais.

- Clique sobre a célula A1 e, sem soltar o clique, arraste até a célula B11.
- Clique em Inserir > Gráfico
- Em "Tipo de gráfico", selecione Gráfico de colunas
- Clique na caixa "Alternar linhas/colunas"
- Assim, obteremos um gráfico com os bairros que têm mais de 10 organizações cadastradas e suas quantidades correspondentes



Tutorial Tabela Dinâmica 0:53:04

Tutorial sobre Tabela Dinâmica

Tabela Dinâmica

Vamos importar a base de dados [sobre Rendimento e movimento escolar](#).

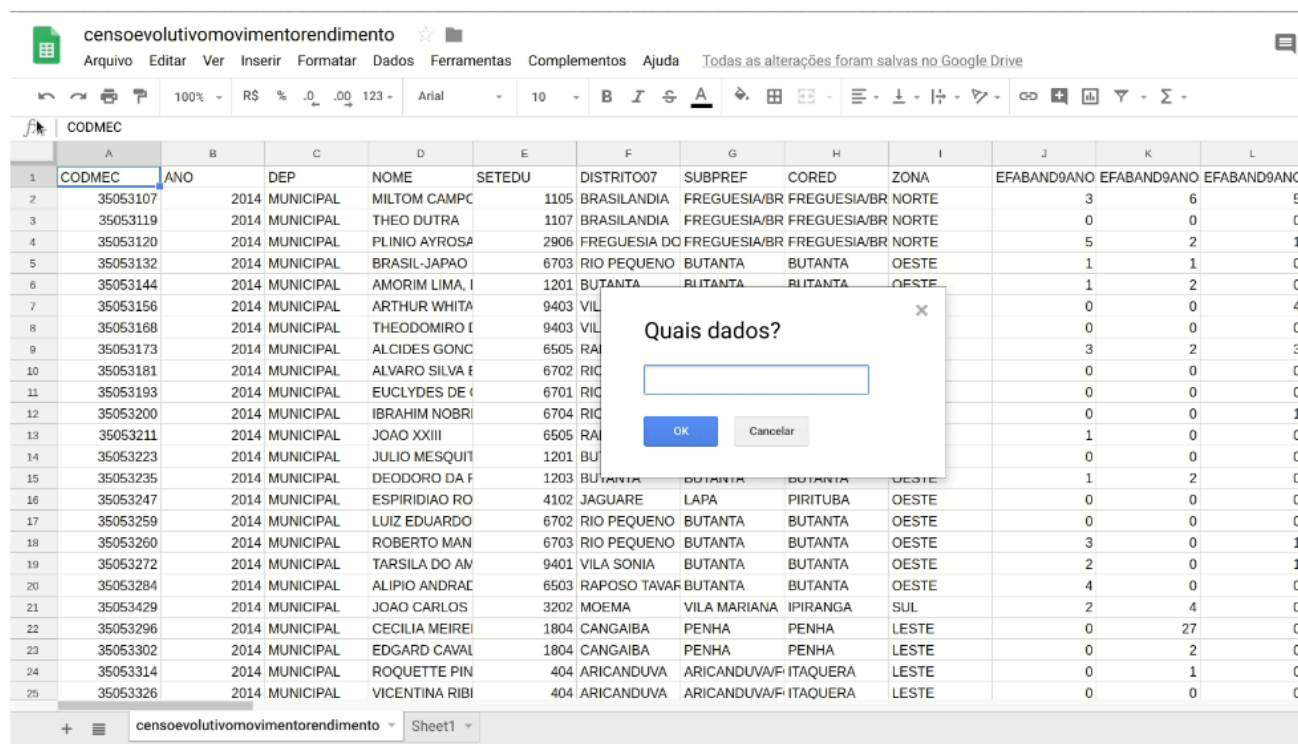
Começamos abrindo um novo arquivo, selecionando *Arquivo -> Importar*.

Selecionamos Upload e realizamos a transferência do arquivo "censoevolitivomovimentorendimento.csv".

No campo "Tipo de Separador", escolher Personalizado e digitar o símbolo de ponto e vírgula.

Agora, em uma nova planilha, selecionamos *DADOS -> Tabela Dinâmica*

No campo "Quais dados?" selecionamos a planilha "censoevolitivorendimento" em sua totalidade, selecionando o quadrado cinza no canto superior esquerdo das células.



The screenshot shows a Google Sheets interface with a spreadsheet titled "censoevolitivomovimentorendimento". A dialog box titled "Quais dados?" is open, showing a selection of the entire data range. The spreadsheet data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L
	CODMEC	ANO	DEP	NOME	SETEDU	DISTRITO07	SUBPREF	CORED	ZONA	EFABAND9ANO	EFABAND9ANO	EFABAND9ANO
1	35053107	2014	MUNICIPAL	MILTON CAMPO	1105	BRASILANDIA	FREGUESIA/BR	FREGUESIA/BR	NORTE	3	6	5
2	35053119	2014	MUNICIPAL	THEO DUTRA	1107	BRASILANDIA	FREGUESIA/BR	FREGUESIA/BR	NORTE	0	0	0
3	35053120	2014	MUNICIPAL	PLINIO AYROSA	2906	FREGUESIA DO	FREGUESIA/BR	FREGUESIA/BR	NORTE	5	2	1
4	35053132	2014	MUNICIPAL	BRASIL-JAPAO	6703	RIO PEQUENO	BUTANTA	BUTANTA	OESTE	1	1	0
5	35053144	2014	MUNICIPAL	AMORIM LIMA, I	1201	BUTANTA	BUTANTA	BUTANTA	OESTE	1	2	0
6	35053156	2014	MUNICIPAL	ARTHUR WHITA	9403	VILA				0	0	4
7	35053168	2014	MUNICIPAL	THEODOMIRO I	9403	VILA				0	0	0
8	35053173	2014	MUNICIPAL	ALCIDES GONC	6505	RAPOSO				3	2	3
9	35053181	2014	MUNICIPAL	ALVARO SILVA E	6702	RIO				0	0	0
10	35053193	2014	MUNICIPAL	EUCLYDES DE S	6701	RIO				0	0	0
11	35053200	2014	MUNICIPAL	IBRAHIM NOBRE	6704	RIO				0	0	1
12	35053211	2014	MUNICIPAL	JOAO XXIII	6505	RAPOSO				1	0	0
13	35053223	2014	MUNICIPAL	JULIO MESQUITA	1201	BUTANTA				0	0	0
14	35053235	2014	MUNICIPAL	DEODORO DA F	1203	BUTANTA	BUTANTA	BUTANTA	OESTE	1	2	0
15	35053247	2014	MUNICIPAL	ESPIRIDIAO RO	4102	JAGUARE	LAPA	PIRITUBA	OESTE	0	0	0
16	35053259	2014	MUNICIPAL	LUIZ EDUARDO	6702	RIO PEQUENO	BUTANTA	BUTANTA	OESTE	0	0	0
17	35053260	2014	MUNICIPAL	ROBERTO MAN	6703	RIO PEQUENO	BUTANTA	BUTANTA	OESTE	3	0	1
18	35053272	2014	MUNICIPAL	TARSILA DO AM	9401	VILA SONIA	BUTANTA	BUTANTA	OESTE	2	0	1
19	35053284	2014	MUNICIPAL	ALIPIO ANDRAE	6503	RAPOSO TAVAR	BUTANTA	BUTANTA	OESTE	4	0	0
20	35053429	2014	MUNICIPAL	JOAO CARLOS	3202	MOEMA	VILA MARIANA	IPIRANGA	SUL	2	4	0
21	35053296	2014	MUNICIPAL	CECILIA MEIRE	1804	CANGAIBA	PENHA	PENHA	LESTE	0	27	0
22	35053302	2014	MUNICIPAL	EDGARD CAVAL	1804	CANGAIBA	PENHA	PENHA	LESTE	0	2	0
23	35053314	2014	MUNICIPAL	ROQUETTE PIN	404	ARICANDUVA	ARICANDUVA/F	ITAQUERA	LESTE	0	1	0
24	35053326	2014	MUNICIPAL	VICENTINA RIBI	404	ARICANDUVA	ARICANDUVA/F	ITAQUERA	LESTE	0	0	0

Continuação tutorial Tabela Dinâmica 2:10:41

Vide acima

Correções Finais 2:42:00

Últimas correções e checagens para entrega dos tutoriais finalizados

planejamento Python 0:55:28

Elenquei quais informações seriam as mais interessantes para os tutoriais. Decidi que os tutoriais seriam feitos na plataforma Google Colab e que abrange conceitos básicos, Pandas e Matplotlib

início tutorial Python 2:24:33

Dei início ao tutorial de Python, que cobre conceitos básicos da linguagem de programação. Neste estágio o projeto estava sendo feito no Google Colab e em um arquivo Notebook único.

O que é Python?

[Python](#) é uma das linguagens de programação mais utilizadas atualmente. Ele é conhecida por ser de fácil legibilidade.

Este tutorial não tem por objetivo ser um curso completo de Python. Iremos utilizá-lo apenas como uma ferramenta para analisarmos dados e cobrir apenas aspectos que forem se mostrando necessários.

Existem muitos [materiais excelentes na internet](#) e, se você se interessar mais por Python, recomendamos que consulte alguns deles.

Variáveis

Programar em Python pode ser bastante simples.

Vamos começar declarando uma variável.

[Variáveis](#) é um nome que, em computação, refere-se a um local onde iremos guardar informações.

Usando algumas informações que coletamos [neste tutorial](#), vamos criar uma variável para guardar o número de matriculados no distrito de Água Rasa.

```
In [1]: aguaRasa = 0
```

Pronto! Sua variável já está criada! Simples, não?

Mas nossa variável `aguaRasa` ainda não possui nenhuma informação. Para guardar dados em uma variável, usamos o comando de [atribuição](#), usando o símbolo de igual.

```
In [2]: aguaRasa = 5578
```

Dessa forma, nossa variável `aguaRasa` agora guarda dentro de si o valor 5578.

Funções

Utilizando [funções](#), somos capazes de realizar tarefas que alguém já programou previamente, sem saber como elas funcionam!

Vamos agora imprimir (mostrar na tela) o valor que guardamos na nossa variável `aguaRasa`. Muitas coisas estão envolvidas neste processo, como acesso à memória, controle de entrada e saída, mas, utilizando a função **`print()`**, não é necessário que saibamos fazer nenhuma dessas coisas. Basta escrever:

```
In [3]: print(aguaRasa)
```

```
5578
```

Continuação tutorial Pandas 2:08:04

Início do tutorial de Pandas.

Função sum

Utilizando a função `sum` do Pandas Data frame, conseguimos somar todos os valores. Vamos supor que desejamos descobrir quantos alunos participam da atividade "CLUBE MATEMATIC".

Para isso, vamos criar uma variável chamada `mathClub`. Vamos atribuir a ela a soma das QTDE, de todas as linhas que tiveram "CLUBE MATEMATIC" como descrição da série. Para isso, escrevemos:

```
In [6]: mathClub = df[df["DESCSERIE"]=="CLUBE MATEMATIC"].sum()["QTDE"]  
mathClub
```

Out[6]: 2963

Assim, conseguimos obter o valor total de matriculados em "CLUBE MATEMATIC". Se, no código acima, trocarmos "CLUBE MATEMATIC" por outro argumento, seremos capazes de descobrir a quantidade de alunos de qualquer outra atividade que conste em nosso Data frame.

Criando uma pivot table

Vamos agora usar a função `pivot_table` para criar uma tabela.

Vamos criar uma tabela que agrupe todas as quantidades de meninas(F) e meninos(M), indexadas pela descrição da atividade(DESCSERIE).

Para tal, iremos criar uma variável `pivotSexo` e chamar `pivot_table` sobre nosso dataframe `df`.

```
In [7]: pivotSexo = df.pivot_table(index = 'DESCSERIE', columns = 'SEXO', aggfunc=sum)  
pivotSexo
```

Out[7]:

	QTDE	
SEXO	F	M
DESCSERIE		
INF II ESP FI	11.0	24.0
1A.C.NORMAL	42.0	4.0
1A.SER.E.M.	522.0	566.0
1ª EJA MODULAR	233.0	167.0
2A.C.NORMAL	27.0	NaN

Mais Pandas 2:12:00

Continuando tutorial de Pandas. Encontrei diversas dificuldades nesse ponto. Em parte devido a conhecimento insuficiente sobre a biblioteca, o que requereu estudo, mas esbarrei em limitações da plataforma Google colab, o que nos levou a....

MAX e IDMAX

Suponha agora que desejamos descobrir quais atividades têm a maior diferença entre meninos e meninas inscritos. Para isso, temos duas funções muito úteis, [idxmax](#) e [max](#).

IDMAX retorna o índice com maior valor. No nosso caso, iremos usá-la para retornar o nome da atividade com o maior valor definido.

Já MAX retorna o valor propriamente dito.

```
In [12]: print("Atividade com maior diferença absoluta entre meninos e meninas : ")
print((pivotSexo['QTDE']['M'] - pivotSexo['QTDE']['F']).idxmax())

print("\nAtividade com maior diferença absoluta entre meninas e meninos : ")
print((pivotSexo['QTDE']['F'] - pivotSexo['QTDE']['M']).idxmax())

print("\nAtividade com maior diferença relativa entre meninos e meninas : ")
print(((pivotSexo['QTDE']['M'] - pivotSexo['QTDE']['F'])/(pivotSexo['QTDE']['M']+pivotSexo['QTDE']['F'])).idxmax())
print("Razão de meninos para meninas : ")
print(((pivotSexo['QTDE']['M'] - pivotSexo['QTDE']['F'])/(pivotSexo['QTDE']['M']+pivotSexo['QTDE']['F'])).max())

print("\nAtividade com maior diferença relativa entre meninas e meninos : ")
print(((pivotSexo['QTDE']['F'] - pivotSexo['QTDE']['M'])/(pivotSexo['QTDE']['M']+pivotSexo['QTDE']['F'])).idxmax())
```

Atividade com maior diferença absoluta entre meninos e meninas :
REC PAR PORTUGU

Atividade com maior diferença absoluta entre meninas e meninos :
ESP - GINASTIC

Atividade com maior diferença relativa entre meninos e meninas :
FUT FUTSAL MASC
Razão de meninos para meninas :
0.8980582524271845

Atividade com maior diferença relativa entre meninas e meninos :
OFICINA

```
In [13]: print(pivotSexo.sort_values(by = [('QTDE', 'F')], ascending = False))
```

	QTDE	
SEXO	F	M
DESCSERIE		
MINI - GRUPO II	49427.0	52624.0
INFANTIL II 6H	48966.0	51449.0

migrando para Jupyter 3:40:24

Decidi abandonar o Google Colab e migrar o tutorial para o Jupyter Notebook. Isso requereu que grandes partes do tutorial fossem reescritas, bem como a cópia dos trechos de código para a nova plataforma.

Python Md 4:12:17

Iniciei um Markdown do tutorial, para ajudar na migração para o Jupyter Notebook. Estudei também a utilização de GIFs, como nos tutoriais de Excel, mas acabei abandonando ambas as ideias e deixando tudo unicamente no Jupyter.

python 2:20:07

Finalmente com todo o tutorial migrado, foi possível dar continuidade às atividades com o Pandas.

Tutorial Python - continuação 2:17:48

Continuação do pacote Pandas

Palestra Pátio Digital 1:29:44

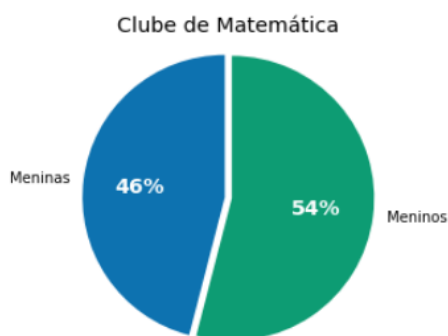
Palestra oferecida pelo TECs com a Fernanda do Pátio Digital, que discursou sobre tópicos relevantes ao Python e as bases de dados. Forneceram novas ideias para os tutoriais.

matplotlib 2:58:00

Início dos tutoriais sobre o Matplotlib, com foco em gráfico de Pizza

Com mais algumas modificações, já temos uma visualização muito melhor do nosso conjunto de dados:

```
In [10]: fig = plt.pie(mathClubSexo, labels = ['Meninas', 'Meninos'], autopct='%1.1f%%',
startangle=90, explode = [0.0, 0.05])
plt.axis('equal') #Fazer o grafico ser um circulo perfeito
plt.title('Clube de Matemática', fontsize = 'x-large') #adicionando o título
patches, texts, autotexts = fig
plt.setp(autotexts, size='x-large', color = 'w', weight = 'bold')
plt.savefig('Exemplos/Clube de Matemática.png', dpi = 300)
```



Explicando as modificações realizadas:

- labels = ['Meninas', 'Meninos'] => adiciona as legendas aos setores
- autopct='%1.1f%%' => adiciona as porcentagens dentro dos setores
- startangle=90 => determina o ângulo inicial para divisão dos setores
- explode = [0.0, 0.05] => cria a "quebra" entre as parte dos setores do gráfico
- plt.setp(autotexts, size='x-large', color = 'w', weight = 'bold') => modifica a fonte das porcentagens
- plt.savefig('Exemplos/Clube de Matemática.png', dpi = 300) => salva a figura no destino especificado

Python Gráfico de barras 0:38:09

Início do tutorial sobre gráfico de barras no Matplotlib

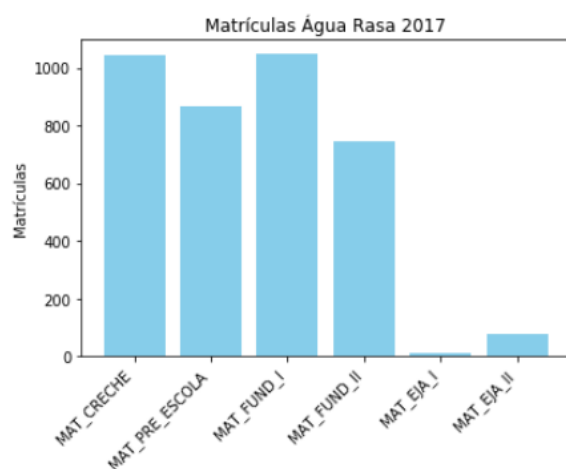
```
In [5]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

#Criar um vetor de indices
ind = np.arange(len(aguaRasa2017.index[1:-1]))
#Largura das barras
width = 0.35

fig, ax = plt.subplots()
rects1 = ax.bar(ind,height = aguaRasa2017.values[1:-1], color='SkyBlue')

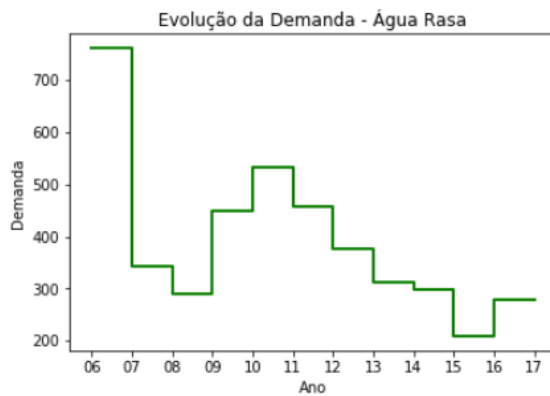
ax.set_ylabel('Matrículas')
ax.set_title('Matrículas Água Rasa 2017')
ax.set_xticks(ind)
ax.set_xticklabels(aguaRasa2017.index[1:-1], rotation = 45, ha = 'right')
plt.savefig('Exemplos/Matrículas Água Rasa 2017.png', dpi = 300, bbox_inches='tight')

plt.show()
```

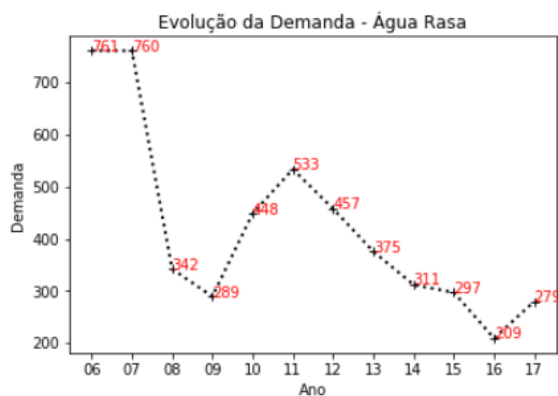


matplotlib 0:57:58

Continuação do tutorial sobre gráfico de barras no MatPlotLib



```
In [12]: fig, ax = plt.subplots()
ax.set_ylabel('Demanda')
ax.set_xlabel('Ano')
ax.set_title('Evolução da Demanda - Água Rasa')
line, = ax.plot(aguaRasaAno)
plt.setp(line, color='black', linewidth=2.0, ls=':', marker='+')
for i,j in zip(aguaRasaAno.index.values, list(aguaRasaAno['Demanda'])):
    ax.annotate(str(int(j)),xy=(i, j), color='r')
plt.savefig('Exemplos/Evolução da Demanda - Água Rasa 3', dpi = 300)
```



Revisões de texto 1:30:04

Revisões de Texto

Modularizando Tutoriais 0:58:00

Reestruturei o github como um todo

revisão Python 0:29:19

Revisões de texto, ortografia e gramática

matplotlib Bar 3:13:50

Continuação do tutorial sobre gráfico no Matplotlib

matplotlib Bar 2:16:21

Continuação do tutorial sobre gráfico no Matplotlib

Regex and Groupby 2:05:43

Tutorial específico sobre Regex e Groupby

Mais um exemplo com Regex

No exemplo do gráfico de barras, utilizamos as informações contidas no csv para criar os nomes no eixo X. Mas os nomes não estão num formato muito agradável. Recuperando esse mesmo exemplo, podemos usar Regex para modificar os textos, a fim de ficarem mais agradáveis:

```
In [13]: aguaRasa2017.index = aguaRasa2017.index.str.replace( " ", ' ' )
         aguaRasa2017.index = aguaRasa2017.index.str.replace( "MAT", ' ' )
         aguaRasa2017.index = aguaRasa2017.index.str.strip()
```

```
aguaRasa2017
```

```
Out[13]: DISTRITO      AGUA RASA
         CRECHE        1045
         PRE ESCOLA     866
         FUND I        1047
         FUND II       745
         EJA I         12
         EJA II        78
         MES ANO REF    dez-17
         Name: 46, dtype: object
```

```
In [14]: #Criar um vetor de indices
         ind = np.arange(len(aguaRasa2017.index[1:-1]))
         #Largura das barras
         width = 0.35

         fig, ax = plt.subplots()
         rectsl = ax.bar(ind,height = aguaRasa2017.values[1:-1], color='SkyBlue')

         ax.set_ylabel('Matrículas')
         ax.set_xlabel('Séries escolares ')
         ax.set_title('Matrículas Água Rasa 2017')
         ax.set_xticks(ind)
         ax.set_xticklabels(aguaRasa2017.index[1:-1], rotation = 45, ha = 'right')
         plt.savefig('Exemplos/Matrículas Água Rasa 2017 - 2.png', dpi = 300, bbox_inches='tight')

         plt.show()
```

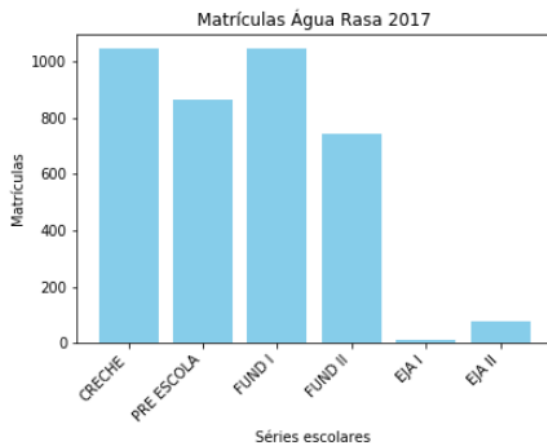
Matrículas Água Rasa 2017


```
In [14]: #Criar um vetor de indices
ind = np.arange(len(aguaRasa2017.index[1:-1]))
#Largura das barras
width = 0.35

fig, ax = plt.subplots()
rects1 = ax.bar(ind,height = aguaRasa2017.values[1:-1], color='SkyBlue')

ax.set_ylabel('Matrículas')
ax.set_xlabel('Séries escolares ')
ax.set_title('Matrículas Água Rasa 2017')
ax.set_xticks(ind)
ax.set_xticklabels(aguaRasa2017.index[1:-1], rotation = 45, ha = 'right')
plt.savefig('Exemplos/Matrículas Água Rasa 2017 - 2.png', dpi = 300, bbox_inches='tight')

plt.show()
```



Modularizando Tutoriais 2:33:21

Neste ponto, todos os tutoriais estavam em um único arquivo do Jupyter Notebook. Separei em tópicos e colapse blocos de código, reestruturando o tutorial como um todo.

Correções Ortográficas 2:00:48

Vide Revisões

Mais correctos 3:29:17

Vide Revisões

Revisão de Texto 2:49:31

Vide Revisões

Revisões finais Python 1:20:00

Vide Revisões

Exportar Python 0:21:27

Tutorial sobre exportar para CSV e Excel

Novo Tutorial - funções 1:45:37

Tutorial sobre definições de funções

Funções

Uma função é um bloco de código que apenas roda quando é chamada. Já usamos uma função, que vem definida por padrão no Python, o print. Mas podemos também criar nossas próprias funções, para realizarmos tarefas complexas, sem ter que reescrever muitas vezes o mesmo trecho de código.

In [1]: *#Exemplo de funcao*

```
def imprime(valor):  
    print("valor digitado: " + str(valor))  
    print("Dobro do valor digitado: " + str(valor*2))  
  
imprime(5)
```

valor digitado: 5
Dobro do valor digitado: 10

Vamos importar a base de dados [sobre Rendimento e movimento escolar](#) e verificar dados sobre alunos aprovados, repovados e evadidos no ensino fundamental:

In [2]: **import pandas as pd**

```
#Lista de colunas que desejamos importar  
cols = ["ANO", "TOTALEF_ABAND", "TOTALEF_APROV", "TOTALEF_REPROV"]  
  
df = pd.read_csv('CSV/censoevolitivomovimentorendimento.csv',  
                 sep = ";", encoding = 'latin-1', header = 0, usecols=cols)  
  
#Utilizar apenas linhas cujo valor da coluna ano e 2015  
df = df[df.ANO == 2015]  
  
df['TOTAL'] = df['TOTALEF_ABAND'] + df['TOTALEF_APROV'] + df['TOTALEF_REPROV']  
  
#Utilizar apenas linhas cujo valor da coluna Total e diferente de 0  
df = df[df.TOTAL != 0].reset_index(drop=True)  
  
df
```

Out[2]:

	ANO	TOTALEF_ABAND	TOTALEF_APROV	TOTALEF_REPROV	TOTAL
0	2015	4	557	59	620
1	2015	6	338	22	366
2	2015	7	438	19	464
3	2015	3	746	17	766