

Aula 01 | PosTech

Anotações sobre a primeira aula da PosTech FIAP ✨✨

<https://on.fiap.com.br/mod/conteudoshtml/view.php?id=307785&c=8729&sesskey=vCMoHFxpWh>

Temas abordados:

- Ambiente Google Colab
- Primeira leitura e manipulação de dados com Pandas
- Como ler dados vindo de uma fonte real?

☐ pip install matplotlib

☐ pip install pandas

```
import pandas as pd

dados = pd.read_csv('/content/A150850189_28_143_208.csv',
                    encoding='ISO-8859-1',
                    skiprows=3,
                    sep=';',
                    skipfooter=12,
                    thousands='.',
                    decimal=',')

dados.head()
dados.tail()
dados.info()
pd.options.display.float_format = '{:.2f}'.format
dados.mean()
```

Parte 1 | Aquisição e leitura de dados reais I

O DATASUS tem uma disponibilização de diversos bancos de dados de informações relativas à saúde.

<https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>

Dentro do DATASUS, tem uma área chamada “tabnet”, que vai disponibilizar os dados tabulados para nós.

- ☐ Entrar em “Assistência à saúde”
- ☐ Entrar em “Produção Hospitalar (SIH/SUS)”

▶ Indicadores de Saúde e Pactuações

▲ Assistência à Saúde

- Produção Hospitalar (SIH/SUS)
- Produção Ambulatorial (SIA/SUS)
- Imunizações – desde 1994
- Atenção Básica – Saúde da Família – de 1998 a 2015
- Vigilância Alimentar e Nutricional
- Conjunto Mínimo de Dados (CMD)

▶ Epidemiológicas e Morbidade

▶ Rede Assistencial

▶ Estatísticas Vitais

▶ Demográficas e Socioeconômicas

▶ Inquéritos e Pesquisas

▶ Saúde Suplementar (ANS)

▶ Informações Financeiras

▶ Estatísticas de acesso ao TABNET

Então, vai abrir uma tela em que tenho a opção de escolher **como** quero trabalhar com os meus dados, que são dados do mundo real, eles vão mudando o tempo todo.

A versão mais recente é a partir de 2008.

Nós queremos trabalhar com dados consolidados, não super detalhados.

Temos três opções: local de internação, local de resistência ou por gestão.

☐ Selecionar Dados **Consolidados** AIH (RD), por **local de internação**, a partir de **2008**.

Depois, em Abrangência Geográfica...

☐ Brasil por Região e Unidade de Federação

Produção Hospitalar (SIH/SUS)

Opção selecionada: Produção Hospitalar (SIH/SUS)

☒ Dados Consolidados AIH (RD), por local de internação, a partir de 2008

☐ Dados Consolidados AIH (RD), por local de internação, de 1992 a 2007

☐ Dados Consolidados AIH (RD), por local de residência, a partir de 2008

☐ Dados Consolidados AIH (RD), por local de residência, de 1995 a 2007

☐ Dados Consolidados AIH (RD), por gestor, a partir de 2008

☐ Dados Detalhados de AIH (SP), por local de internação, 2008 em diante

☐ Dados Detalhados de AIH (SP), por local de residência, 2008 em diante

☐ Dados Detalhados de AIH (SP), por gestor, de 2008 em diante

Abrangência Geográfica:

Brasil por Região e Unidade da Federação ▼

Na próxima página, temos: “**Procedimentos Hospitalares do SUS por local de internação no Brasil**” com as opções de Linha, Coluna e Conteúdo; “**Períodos Disponíveis**” com o range de datas disponíveis e “**Seleções Disponíveis**”.

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

Linha	Coluna	Conteúdo
Região	Não ativa	AIH aprovadas
Região/Unidade da Federação	Região	Internações
Unidade da Federação	Unidade da Federação	Valor total
Ano/mês processamento	Ano/mês processamento	Valor serviços hospitalares

PERÍODOS DISPONÍVEIS

Mar/2023
Fev/2023
Jan/2023
Dez/2022
Nov/2022
Out/2022

SELEÇÕES DISPONÍVEIS

- ☐ Região
- ☐ Unidade da Federação
- ☐ Caráter atendimento
- ☐ Procedimento
- ☐ Grupo procedimento
- ☐ Subgrupo proced.
- ☐ Forma organização
- ☐ Complexidade
- ☐ Financiamento
- ☐ Rubrica FAEC
- ☐ Regra contratual
- ☐ Natureza
- ☐ Regime
- ☐ Natureza jurídica
- ☐ Esfera jurídica
- ☐ Gestão

☐ Deixar tudo como padrão (default).

☐ Clicar em “Mostra”

- ☐ Complexidade
- ☐ Financiamento
- ☐ Rubrica FAEC
- ☐ Regra contratual
- ☐ Natureza
- ☐ Regime
- ☐ Natureza jurídica
- ☐ Esfera jurídica
- ☐ Gestão

☐ Ordenar pelos valores da coluna ☐ Exibir linhas zeradas

Formato ☒ Tabela com bordas ☐ Texto pré formatado ☐ Colunas separadas por ";;"

Mostra

Limpa

e: Ministério da Saúde - Sistema de Informações Hospitalares do SUS (SIH/SUS)

S:

1. Dados referentes aos últimos seis meses, sujeitos a atualização.
2. A partir do processamento de junho de 2012, houve mudança na classificação da natureza e esfera dos estabelecimentos. Com isso, temos que:
 - o Até maio de 2012 estas informações estão disponíveis como "Natureza" e "Esfera Administrativa".
 - o De junho de 2012 a outubro de 2015, estão disponíveis tanto como "Natureza" e "Esfera Administrativa", como "Natureza Jurídica" e "Esfera Jurídica".
 - o A partir de novembro de 2015, estão disponíveis como "Natureza Jurídica" e "Esfera Jurídica".

DATASUS

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

AIH aprovadas segundo Região
Período: Mar/2023

Região	AIH aprovadas
TOTAL	1.050.280
1 Região Norte	73.457
2 Região Nordeste	269.231
3 Região Sudeste	434.609
4 Região Sul	190.294
5 Região Centro-Oeste	82.689

Fonte: Ministério da Saúde - Sistema de Informações Hospitalares do SUS (SIH/SUS)

Notas:

1. Dados referentes aos últimos seis meses, sujeitos a atualização.
2. A partir do processamento de junho de 2012, houve mudança na classificação da natureza e esfera dos estabelecimentos. Com isso, temos que:
 - o Até maio de 2012 estas informações estão disponíveis como "Natureza" e "Esfera Administrativa".
 - o De junho de 2012 a outubro de 2015, estão disponíveis tanto como "Natureza" e "Esfera Administrativa", como "Natureza Jurídica" e "Esfera Jurídica".
 - o A partir de novembro de 2015, estão disponíveis como "Natureza Jurídica" e "Esfera Jurídica".

COPIA COMO .CSV

COPIA PARA TABWIN

MOSTRA COMO MAPA

MOSTRA COMO GRÁFICO

VOLTAR

☐ Clicar em "Voltar"

Para quem já está trabalhando com SQL, com linguagens em que a gente faz Querys, Excel, pesquisas, vocês vão começar a perceber a similaridade.

Linha

O primeiro ponto é na **LINHA**, em vez de ter uma linha que vai me trazer (quando eu trago, busco, seleciono), em vez de agrupar por REGIÃO (group by região), eu vou querer agrupar por **Região / Unidade da Federação**.

Ex: Região Norte e a unidade de federação Amazonas, por exemplo.

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

Linha	Coluna	Conteúdo
Região	Não ativa	AIH aprovadas
Região/Unidade da Federação	Região	Internações
Unidade da Federação	Unidade da Federação	Valor total
Ano/mês processamento	Ano/mês processamento	Valor serviços hospitalares

PERÍODOS DISPONÍVEIS

Mar/2023
Fev/2023
Jan/2023
Dez/2022
Nov/2022
Out/2022

TabNet Win32 3.0: Procedimento: x +

Não seguro | tabnet.datasus.gov.br/cgi/tabcgi.exe?sih/cnv/qjuf.def

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

AIH aprovadas segundo Região/Unidade da Federação
Período: Mar/2023

Região/Unidade da Federação	AIH aprovadas
TOTAL	1.050.280
Região Norte	73.457
.. Rondônia	8.463
.. Acre	5.266
.. Amazonas	16.125
.. Roraima	3.945
.. Pará	33.677
.. Amapá	5.474
.. Tocantins	507
Região Nordeste	269.231
.. Maranhão	31.003
.. Piauí	17.989
.. Ceará	44.436
.. Rio Grande do Norte	16.513
.. Paraíba	17.766
.. Pernambuco	51.837
.. Alagoas	11.821
.. Sergipe	9.758
.. Bahia	68.108
Região Sudeste	434.609
.. Minas Gerais	117.084
.. Espírito Santo	22.332
.. Rio de Janeiro	67.896
.. São Paulo	227.297
Região Sul	190.294
.. Paraná	77.211

Agora nós temos várias linhas, porque em vez de agrupar por região, agrupou por unidades federativas.

☐ Clicar em “Voltar” porque ainda não é isso o que eu quero...

Conteúdo

- ☐ Mudar conteúdo para “**Valor Total**”
- ☐ Mudar a linha para “**Unidade da Federação**”

Vou descartar as regiões, vai ser apenas “Unidade da Federação”.

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

Linha	Coluna	Conteúdo
Região	Não ativa	AIH aprovadas
Região/Unidade da Federação	Região	Internações
Unidade da Federação	Unidade da Federação	Valor total
Ano/mês processamento	Ano/mês processamento	Valor serviços hospitalares

PERÍODOS DISPONÍVEIS

Mar/2023
Fev/2023
Jan/2023
Dez/2022
Nov/2022
Out/2022

SELEÇÕES DISPONÍVEIS

Valor total segundo Unidade da Federação
Período: Mar/2023

Unidade da Federação	Valor total
TOTAL	1.636.140.330,59
11 Rondônia	10.106.389,66
12 Acre	4.927.482,49
13 Amazonas	20.369.596,71
14 Roraima	3.803.846,21
15 Pará	42.871.236,38
16 Amapá	4.172.716,34
17 Tocantins	284.553,86
21 Maranhão	34.052.990,12
22 Piauí	21.376.310,93
23 Ceará	60.685.730,73
24 Rio Grande do Norte	26.508.039,08
25 Paraíba	25.517.986,11
26 Pernambuco	81.345.720,90
27 Alagoas	16.327.347,69
28 Sergipe	14.750.391,01
29 Bahia	88.938.352,90
31 Minas Gerais	207.885.850,88
32 Espírito Santo	39.104.818,79
33 Rio de Janeiro	103.199.259,99
35 São Paulo	371.658.829,80
41 Paraná	146.448.058,69
42 Santa Catarina	81.741.392,27
43 Rio Grande do Sul	111.030.086,45

Agora temos uma linha por “Unidade da Federação” e, na direita, temos a segunda coluna com o “Valor Total” em milhões de reais gastos na saúde.

☐ Clicar em “Voltar” porque ainda não é isso o que eu quero...

Colunas

Em vez de uma coluna, eu quero várias colunas.

☐ Clicar em “Ano / mês processamento”

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

Linha	Coluna	Conteúdo
Região	Unidade da Federação	AIH aprovadas
Região/Unidade da Federação	Ano/mês processamento	Internações
Unidade da Federação	Ano processamento	Valor total
Ano/mês processamento	Ano/mês atendimento	Valor serviços hospitalares

PERÍODOS DISPONÍVEIS

Nov/2021
 Out/2021
 Set/2021
 Ago/2021
 Jul/2021
 Jun/2021

Surpresa: Vai trazer basicamente 2021 / Setembro.

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

Valor total por Ano/mês processamento segundo Unidade da Federação
Período: Set/2021

Unidade da Federação	2021/Set	Total
TOTAL	1.848.026.855,82	1.848.026.855,82
11 Rondônia	12.115.394,39	12.115.394,39
12 Acre	4.260.017,06	4.260.017,06
13 Amazonas	21.120.609,03	21.120.609,03
14 Roraima	3.150.835,05	3.150.835,05
15 Pará	45.776.186,52	45.776.186,52
16 Amapá	5.036.697,91	5.036.697,91
17 Tocantins	11.170.484,72	11.170.484,72
21 Maranhão	40.122.547,01	40.122.547,01
22 Piauí	23.195.247,30	23.195.247,30
23 Ceará	81.201.363,06	81.201.363,06
24 Rio Grande do Norte	28.124.087,90	28.124.087,90
25 Paraíba	24.934.526,89	24.934.526,89
26 Pernambuco	85.587.490,39	85.587.490,39
27 Alagoas	21.964.123,03	21.964.123,03
28 Sergipe	12.144.593,36	12.144.593,36
29 Bahia	103.742.560,79	103.742.560,79
31 Minas Gerais	213.196.539,53	213.196.539,53
32 Espírito Santo	39.443.023,62	39.443.023,62
33 Rio de Janeiro	130.088.686,39	130.088.686,39

E é o mesmo valor o de mês de setembro e o total... Porque o ano de mês de processamento daquilo que foi processado em Setembro de 2021 é Setembro de 2021. :) kkkk portanto, temos que entender sobre domínio, sobre o que exatamente estamos trabalhando. ;) Isso é muito comum no dia-a-dia de cientistas de dados / análise de dados.

Obs: AIH → autorização de internação hospitalar

Precisamos entender o que significa cada escopo do dado quando estamos olhando para eles.

- ☐ Clicar em “Voltar” porque ainda não é isso o que eu quero...
- ☐ Fazer um range de “Setembro de 2021” até “Janeiro de 2021”

PROCEDIMENTOS HOSPITALARES DO SUS - POR LOCAL DE INTERNAÇÃO - BRASIL

Linha	Coluna	Conteúdo
Região	Unidade da Federação	AIH aprovadas
Região/Unidade da Federação	Ano/mês processamento	Internações
Unidade da Federação	Ano processamento	Valor total
Ano/mês processamento	Ano/mês atendimento	Valor serviços hospitalares

PERÍODOS DISPONÍVEIS

<div>Jun/2021</div> <div>Mai/2021</div> <div>Abr/2021</div> <div>Mar/2021</div> <div>Fev/2021</div> <div>Jan/2021</div>

Ele vai demorar um pouco mais porque antes era apenas uma seleção, agora seleciona o **WHERE** (onde) entre Janeiro e Setembro e está agrupando cada coluna em Janeiro e Setembro.

Valor total por Ano/mês processamento segundo Unidade da Federação
Período: Jan-Set/2021

Unidade da Federação	2021/Jan	2021/Fev	2021/Mar	2021/Abr	2021/Mai	2021/Jun	2021/Jul	2021/Ago	2021/Set
TOTAL	1.541.725.048,75	1.555.167.786,91	1.807.199.007,24	2.039.787.062,05	2.165.517.054,08	2.257.909.652,62	2.229.133.901,78	2.067.490.720,10	1.848.026.855,82
11 Rondônia	9.861.592,32	9.812.340,48	14.953.391,13	15.563.729,11	16.492.196,14	15.572.901,51	14.742.747,37	13.054.732,77	12.115.394,39
12 Acre	3.715.723,12	3.538.427,92	4.077.045,92	6.307.616,67	6.354.514,15	3.461.663,11	4.979.239,30	4.138.751,64	4.260.017,06
13 Amazonas	21.621.697,26	26.903.413,08	26.598.262,21	22.733.131,00	25.882.861,87	26.200.089,42	24.066.337,32	22.720.450,82	21.120.609,03
14 Roraima	3.266.928,47	3.519.773,73	4.437.161,10	5.171.370,92	4.266.254,94	3.103.979,88	3.743.330,65	3.781.921,16	3.150.835,05
15 Pará	38.842.407,79	39.270.803,77	40.701.470,83	46.662.714,56	55.352.070,66	48.295.002,33	46.336.237,30	45.126.860,59	45.776.186,52
16 Amapá	2.807.807,85	2.822.559,17	3.192.515,69	3.622.062,04	4.388.127,81	3.967.128,98	5.215.333,80	5.269.329,19	5.036.697,91
17 Tocantins	7.302.211,47	9.879.152,35	10.700.770,91	9.365.632,90	9.511.274,51	11.262.768,61	10.159.174,66	10.136.608,55	11.170.484,72
21 Maranhão	31.820.050,29	32.763.511,76	44.854.249,48	50.406.725,35	53.579.187,88	51.476.584,09	52.577.550,99	44.466.565,18	40.122.547,01
22 Piauí	22.694.558,18	20.989.694,74	21.620.638,14	27.410.286,55	30.173.706,38	31.135.971,40	29.451.460,79	28.954.296,01	23.195.247,30
23 Ceará	52.326.883,60	56.302.448,96	68.431.457,05	78.634.779,04	87.024.180,65	110.342.286,04	90.870.768,49	80.203.689,73	81.201.363,06
24 Rio Grande do Norte	28.075.981,70	26.593.976,81	28.485.021,87	35.952.723,38	36.887.747,96	36.785.336,80	36.055.398,64	32.939.814,29	28.124.087,90
25 Paraíba	19.552.670,55	21.541.485,83	26.411.037,52	29.593.654,84	34.740.964,07	34.259.202,92	29.846.704,89	28.354.953,45	24.934.526,89
26 Pernambuco	83.926.177,17	78.212.282,77	87.957.628,27	103.993.131,51	123.659.775,59	125.973.975,41	115.602.232,15	104.026.183,65	85.587.490,39
27 Alagoas	17.548.566,67	18.587.484,37	22.323.051,11	26.826.283,56	29.651.474,26	28.155.677,93	28.995.164,37	24.880.899,63	21.964.123,03
28 Sergipe	12.060.864,70	13.226.099,48	13.507.023,71	14.928.721,85	17.724.664,13	19.633.473,10	18.894.097,84	14.566.795,83	12.144.593,36
29 Bahia	80.041.951,06	82.385.385,11	102.012.674,19	115.565.887,57	115.553.383,46	129.336.764,01	133.185.386,00	110.548.470,15	103.742.560,79
31 Minas Gerais	185.696.085,75	189.117.380,31	214.175.000,28	243.308.338,31	254.115.402,21	268.787.737,44	261.663.469,73	245.948.544,23	213.196.539,53
32 Espírito Santo	41.075.770,15	42.552.074,65	48.960.723,62	56.417.132,95	55.162.721,74	52.214.129,02	55.588.132,78	45.641.859,94	39.443.023,62
33 Rio de Janeiro									

Mas que história é essa de ano / mês de processamento? tem também ano / mês atendimento. Quando eu sou atendido, não necessariamente é quando os dados são processados. Posso ser atendido hoje e, de repente, meus dados são processados daqui a 3 dias ou daqui a 1 mês, seja lá quando for.

Tem pessoas, por exemplo, que precisam usar esses dois dados, então a plataforma fornece tudo isso pra gente e é nosso dever aqui, fazendo essa análise, entender o domínio e os porquês. Quando encontrarmos algo do gênero, pergunta! Procurar na documentação do sistema que tem explicação das colunas, dos valores e pergunte para as pessoas ao meu redor.

☐ Clicar em “Voltar” porque ainda não é isso o que eu quero...

☐ Fazer um range completo, escolher tudo!

Vai demorar muito, mas vai gerar uma tabela completa. 😊

Valor total por Ano/mês processamento segundo Unidade da Federação
Período: Jan/2008-Mar/2023

Unidade da Federação	2008/Jan	2008/Fev	2008/Mar	2008/Abr	2008/Mai	2008/Jun	2008/Jul	2008/Ago	2008/Set	2008/Out
TOTAL	408.424.859,23	613.174.269,21	664.043.058,38	686.662.861,42	715.803.639,49	752.181.563,64	764.990.284,76	750.608.648,28	752.962.642,15	733.019.065,83
11 Rondônia	1.388.528,39	2.931.283,42	1.541.682,52	1.525.314,96	1.645.953,84	1.406.150,68	3.065.279,01	3.231.494,61	3.117.178,63	3.068.430,44
12 Acre	902.416,00	1.497.206,26	1.794.028,48	1.730.469,42	1.819.443,92	1.828.496,00	2.511.754,59	2.089.107,14	2.275.708,53	1.923.886,57
13 Amazonas	4.735.529,42	7.118.990,57	8.196.635,49	8.259.378,42	7.831.399,90	8.477.273,62	9.368.858,72	9.352.532,70	9.363.099,35	8.618.318,83
14 Roraima	657.889,53	777.939,31	718.688,03	839.994,39	862.347,96	832.440,66	996.693,09	894.271,18	910.424,17	1.191.770,32
15 Pará	18.864.744,11	19.553.758,20	21.937.342,70	20.842.829,69	23.249.952,88	23.240.687,56	24.002.223,56	23.341.218,03	25.172.261,32	24.011.712,48
16 Amapá	1.120.973,34	1.268.740,33	974.651,24	1.141.137,32	866.346,18	1.440.978,95	1.426.659,48	1.475.551,38	1.651.341,43	1.866.748,58
17 Tocantins	3.441.592,69	4.985.688,85	6.205.612,60	4.984.482,09	5.511.002,46	4.927.934,22	5.445.602,28	5.413.710,38	5.596.669,50	4.658.414,01
21 Maranhão	10.723.506,39	13.559.679,41	13.985.142,36	14.809.416,46	14.505.727,38	14.908.091,32	15.784.899,40	15.195.312,89	14.951.277,75	16.072.219,28
22 Piauí	7.985.142,43	10.079.928,25	10.298.372,45	10.775.976,33	10.398.360,88	10.528.253,88	11.040.107,92	10.549.176,40	10.899.198,97	11.715.943,38
23 Ceará	15.539.821,90	25.735.101,55	28.724.625,75	30.676.457,06	30.604.453,47	31.272.395,18	30.925.391,36	27.851.438,72	30.362.719,90	27.657.898,74
24 Rio Grande do Norte	5.947.814,08	9.509.755,83	10.352.238,59	10.855.108,50	11.160.168,19	10.475.071,38	11.386.349,96	10.640.817,47	11.336.030,28	10.606.465,07
25 Paraíba	10.723.002,45	11.386.274,47	12.053.643,87	12.653.528,76	12.215.107,11	12.256.928,49	12.755.413,77	12.872.578,37	12.558.296,11	12.768.727,85
26 Pernambuco	14.393.949,89	24.181.400,95	29.991.302,71	34.806.197,97	32.147.987,35	32.239.094,89	32.685.640,76	35.457.038,66	33.668.261,46	30.579.747,51
27 Alagoas	6.243.726,32	8.376.622,16	8.408.336,22	9.090.794,82	9.338.771,45	10.068.210,08	9.887.061,61	9.373.432,00	9.563.580,50	9.805.242,71
28 Sergipe	2.780.750,11	3.676.483,16	5.051.473,61	5.046.063,36	4.741.151,56	5.401.259,63	5.521.904,44	5.238.398,50	5.187.912,19	4.584.068,70
29 Bahia	21.211.522,47	28.189.304,95	34.181.483,52	39.026.934,13	42.949.566,82	47.025.972,04	47.481.695,17	46.931.248,28	44.883.379,31	41.920.011,46
31 Minas Gerais	45.671.659,76	70.194.037,10	79.231.336,95	77.136.537,58	80.554.440,63	94.758.522,83	85.360.064,11	84.722.528,10	84.601.757,95	85.559.789,46
32 Espírito Santo	5.456.723,98	9.102.212,86	10.249.595,97	11.253.833,56	10.198.619,17	13.364.258,29	13.035.374,08	12.588.595,77	12.727.433,30	12.656.039,16

☐ Abrir uma nova aba e selecionar “Ano / mês atendimento” em vez de processamento, selecionando Ctrl + A também em “Períodos Disponíveis”.

Valor total por Ano/mês atendimento segundo Unidade da Federação
Período: Jan/2008-Abr/2023

Unidade da Federação	1992/Mar	1992/Abr	1993/Mai	1993/Dez	1994/Jan	1994/Fev	1994/Mai	1994/Ago	1994/Nov	1995/Jan	1995/Mar	1995/Jun	1995/Ago
TOTAL	40.243,16	11.449,99	34.583,77	68.556,52	39.809,34	19.904,67	19.904,67	34.532,40	23.374,12	19.904,67	24.509,71	19.442,14	10.536,67
11 Rondônia	-	-	-	-	-	-	-	-	-	-	-	-	-
12 Acre	-	-	-	-	-	-	-	-	-	-	-	-	-
13 Amazonas	-	-	-	-	-	-	-	-	-	-	-	-	-
14 Roraima	-	-	-	-	-	-	-	-	-	-	-	-	-
15 Pará	-	-	-	-	-	-	-	-	-	-	-	-	-
16 Amapá	-	-	-	-	-	-	-	-	-	-	-	-	-
17 Tocantins	-	-	-	-	-	-	-	-	-	-	-	-	-
21 Maranhão	-	-	-	-	-	-	-	-	-	-	-	-	-
22 Piauí	-	-	-	-	-	-	-	-	-	-	-	-	-
23 Ceará	-	-	-	-	-	-	-	-	-	-	-	-	-
24 Rio Grande do Norte	-	-	-	-	-	-	-	-	-	-	24.509,71	-	-
25 Paraíba	-	-	-	-	-	-	-	-	-	-	-	-	-
26 Pernambuco	-	-	-	-	-	-	-	-	-	-	-	-	10.536,67
27 Alagoas	-	-	-	-	-	-	-	-	-	-	-	-	-
28 Sergipe	-	-	-	-	-	-	-	-	-	-	-	-	-
29 Bahia	-	-	-	-	-	-	-	-	-	-	-	-	-
31 Minas Gerais	-	-	-	-	-	-	-	-	-	-	-	-	-

Mas não quero que sempre demore toda vez que utilizo essas tabelas, então quero baixar isso como um arquivo...

Se descer na página, dá pra ver a opção “Copia como .csv”, que baixa um arquivo CSV para mim.

☐ Clicar em “Copia como .csv” em ambas as tabelas, ok?

Estado	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
41 Paraná	38.266.347,46	55.384.670,52	53.801.318,85	52.460.306,69	54.419.620,71	57.811.688,48	58.094.241,83	56.982.774,90	57.549.082,26	57.121.887,41	
42 Santa Catarina	16.504.913,45	25.651.429,11	26.743.524,89	27.181.488,26	26.845.308,18	29.909.762,04	30.789.135,84	31.250.854,33	30.019.676,93	31.957.603,59	
43 Rio Grande do Sul	30.931.325,46	57.363.420,01	51.547.794,07	56.673.338,49	48.397.733,07	52.413.429,62	56.636.376,98	56.113.054,70	55.356.224,30	53.637.713,13	
50 Mato Grosso do Sul	5.377.392,11	7.002.472,61	8.423.933,19	8.797.487,68	9.770.421,80	10.905.392,18	10.249.299,82	9.671.553,45	10.218.511,05	9.547.923,55	
51 Mato Grosso	4.795.299,01	8.684.882,23	7.886.243,14	8.794.109,79	9.231.645,40	10.221.763,77	9.163.314,28	9.243.148,81	9.052.354,89	9.677.093,95	
52 Goiás	15.310.637,29	22.258.668,68	21.442.515,75	20.921.212,45	21.615.970,67	18.270.043,87	20.540.517,13	20.650.730,67	20.868.443,56	20.621.423,04	
53 Distrito Federal	4.745.919,19	7.632.480,67	9.647.203,26	11.098.732,99	12.891.719,72	12.850.125,67	12.565.646,74	13.710.857,89	13.165.106,20	15.048.607,23	

Fonte: Ministério da Saúde - Sistema de Informações Hospitalares do SUS (SIH/SUS)

Notas:

1. Dados referentes aos últimos seis meses, sujeitos a atualização.
2. A partir do processamento de junho de 2012, houve mudança na classificação da natureza e esfera dos estabelecimentos. Com isso, temos que:
 - o Até maio de 2012 estas informações estão disponíveis como "Natureza" e "Esfera Administrativa".
 - o De junho de 2012 a outubro de 2015, estão disponíveis tanto como "Natureza" e "Esfera Administrativa", como "Natureza Jurídica" e "Esfera Jurídica".
 - o A partir de novembro de 2015, estão disponíveis como "Natureza Jurídica" e "Esfera Jurídica".

Legenda:

- Dado numérico igual a 0 não resultante de arredondamento.
- 0; 0,0 - Dado numérico igual a 0 resultante de arredondamento de um dado originalmente positivo.

COPIA COMO .CSV

COPIA PARA TABWIN

MOSTRA COMO MAPA

MOSTRA COMO GRÁFICO

VOLTAR

DESAFIO: Em uma das tabelas, temos dados desde 1992, na outra temos dados desde 2008. Por quê?

Por que tem dados vazios nos datasets até 2007?

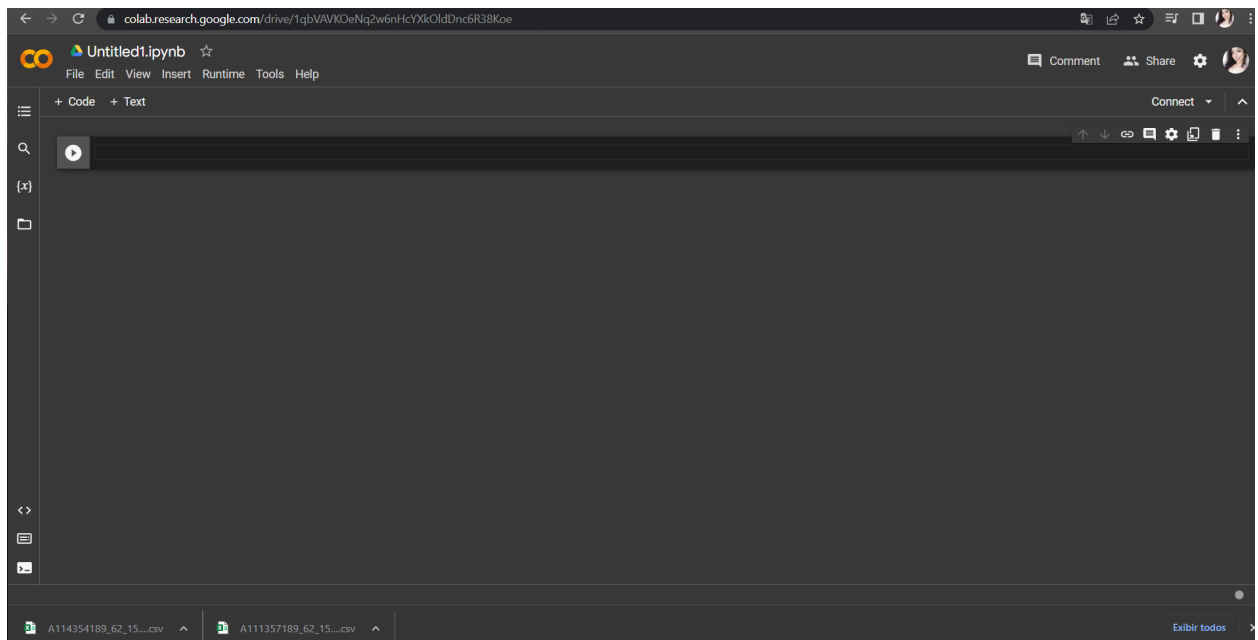
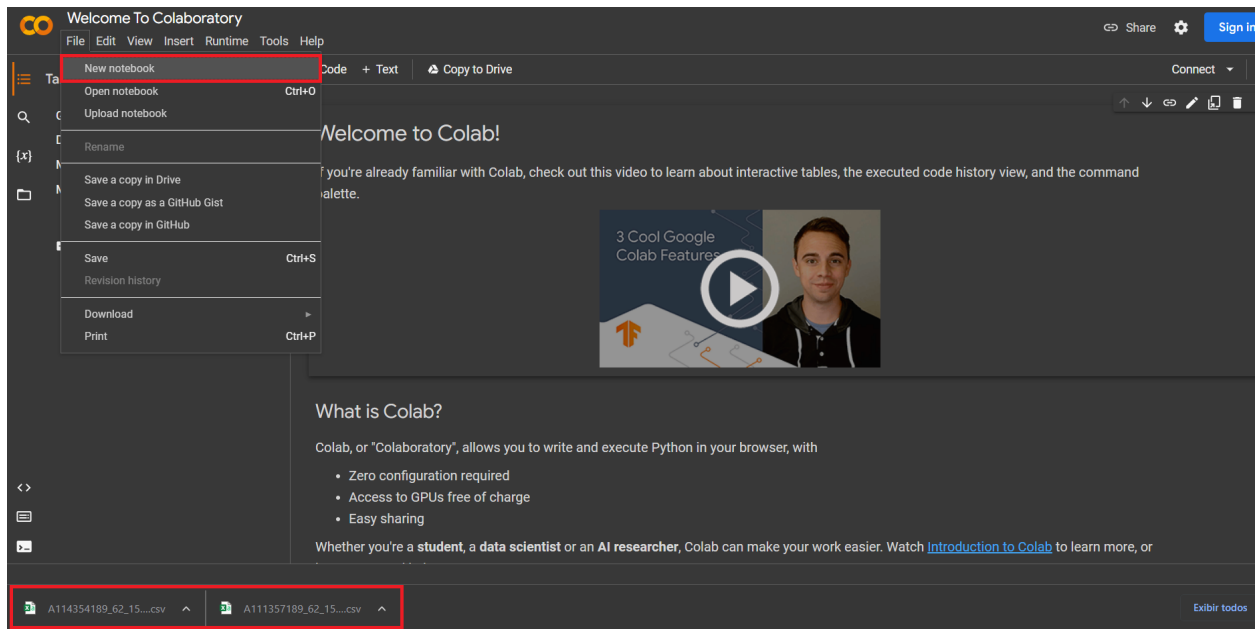
Agora, queremos trazer esses dados para algum lugar em que vou analisá-los...

O que vamos utilizar? Python.

☐ Utilizar o colab.research.google.com, que é um python nas máquinas do Google.

É uma versão online do Jupyter Notebook, que é um **caderno**. Se é um caderno,

- ☐ Dentro do Colab, clicar em “File” e escolher “New notebook”



Ele abre uma célula com um número 1. Como é uma célula, posso escrever o que eu quiser ali.

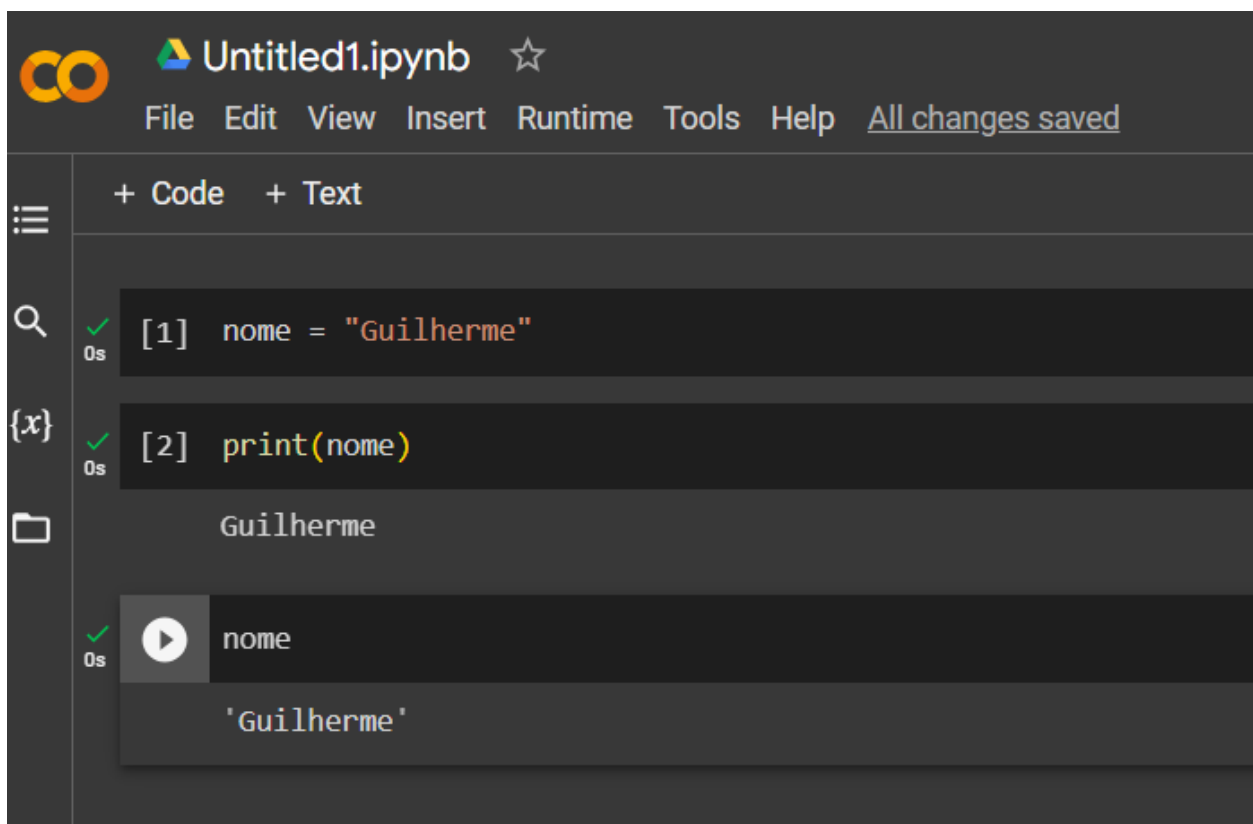
É um notebook de python, precisa ser escrito em python e depois dar um “play”.

Em uma célula declaro uma variável nome que recebe o valor de string de “Guilherme”;

Na próxima, dou um print na variável nome;

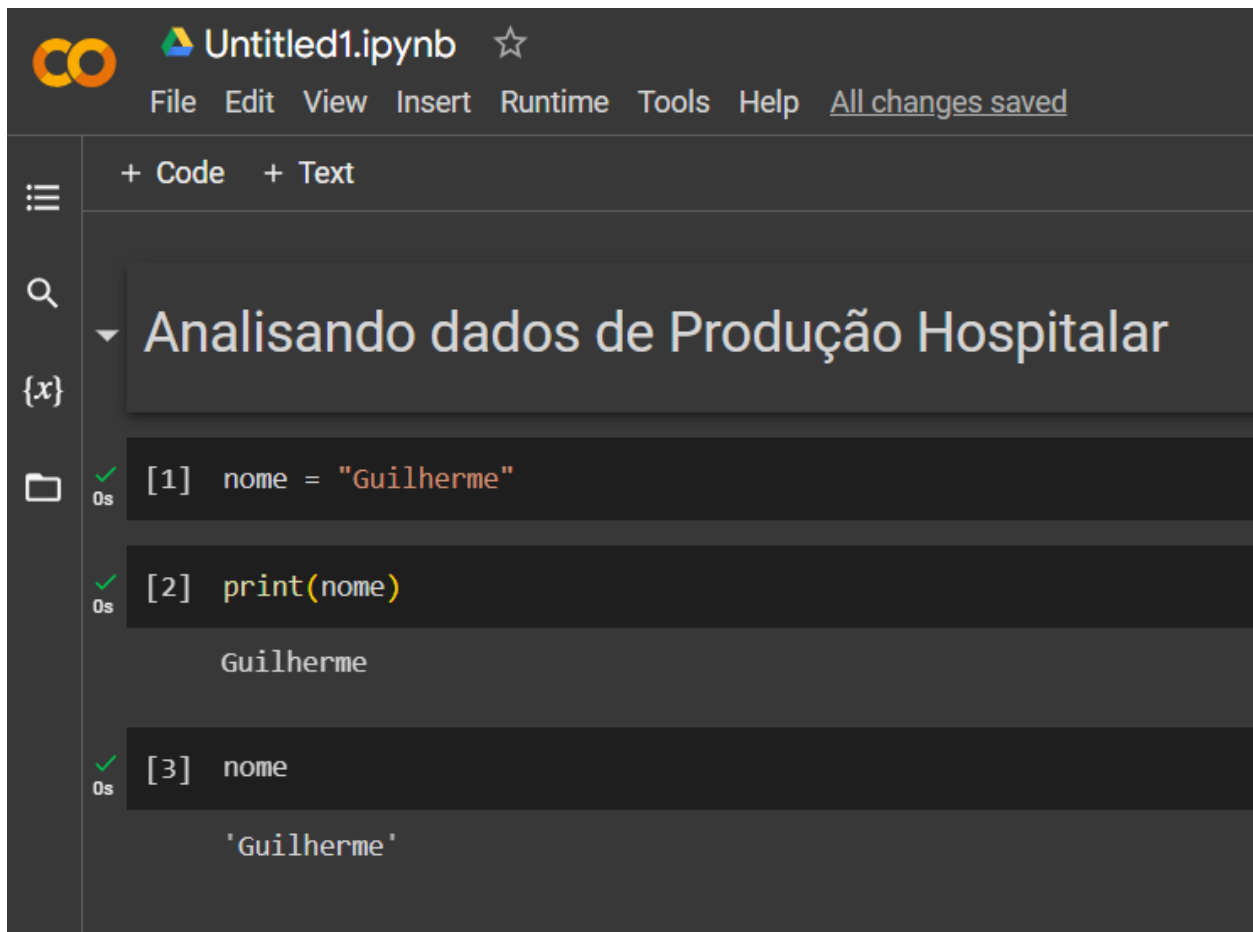
Se escrevo apenas nome em uma célula e dou play, retorna ‘Guilherme’.

O play é o mesmo que dar **Shift + Enter**!



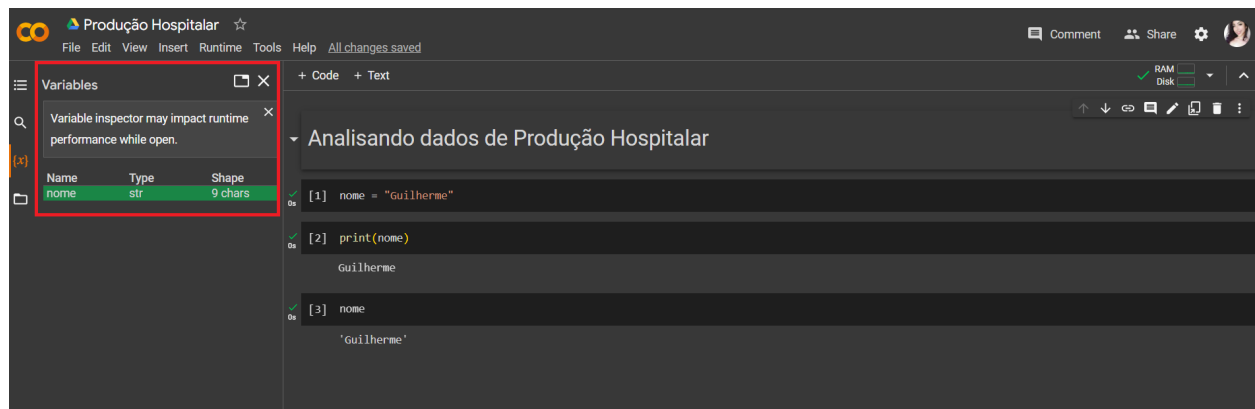
No botão “+ Code” → colocar mais código;

No botão “+ Text” → coloca um código texto, posso subir com uma setinha pra colocar o texto para cima também;



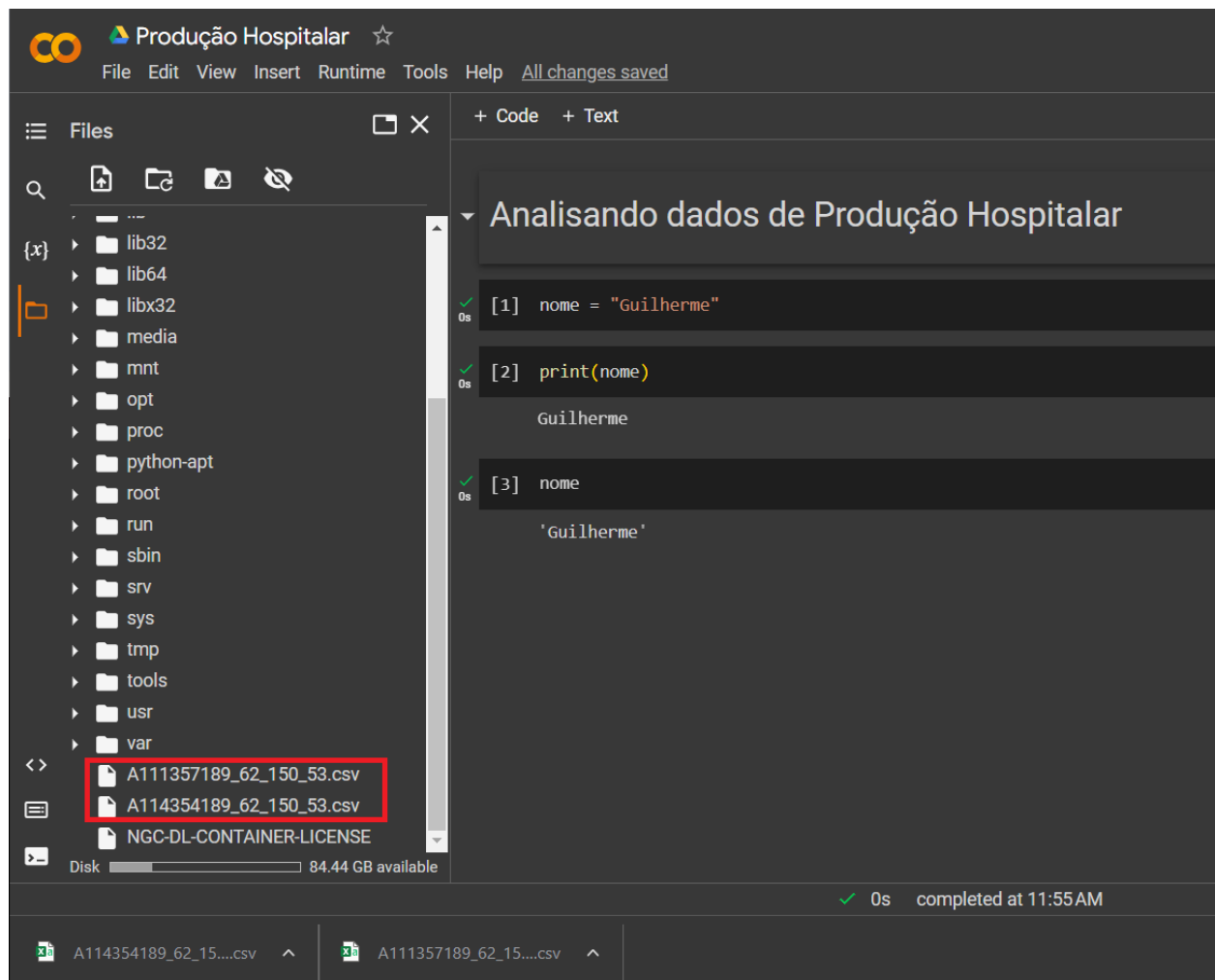
- ☐ Alterar o nome para “Produção Hospitalar”
- ☐ Salvar o arquivo

Essa aba aqui de Variáveis é muito importante, é legal pra visualizar todas as variáveis declaradas ao longo do código... É literalmente para debugar as coisas.

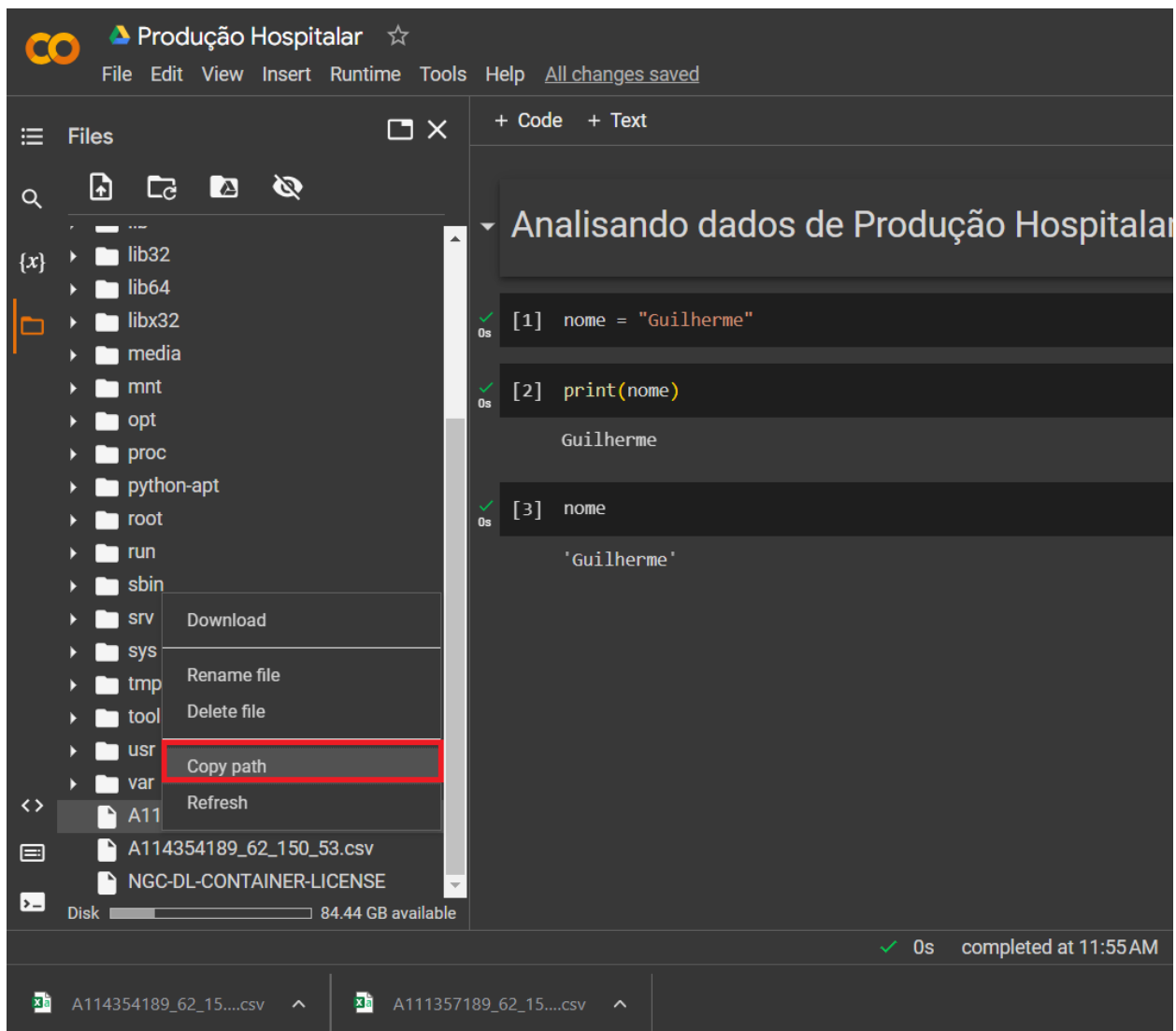


Agora, quero abrir aquele arquivo .csv.

- ☐ Clicar em “files”
- ☐ Arrastar o arquivo que eu baixei e colocar ali na parte de Files, arrastando mesmo. No meu caso não deu certo, então eu procuro o arquivo.



- ☐ Clicar no “menu de hamburguer” do lado do arquivo
- ☐ Clicar em “Copy path”, que copia o caminho para esse arquivo



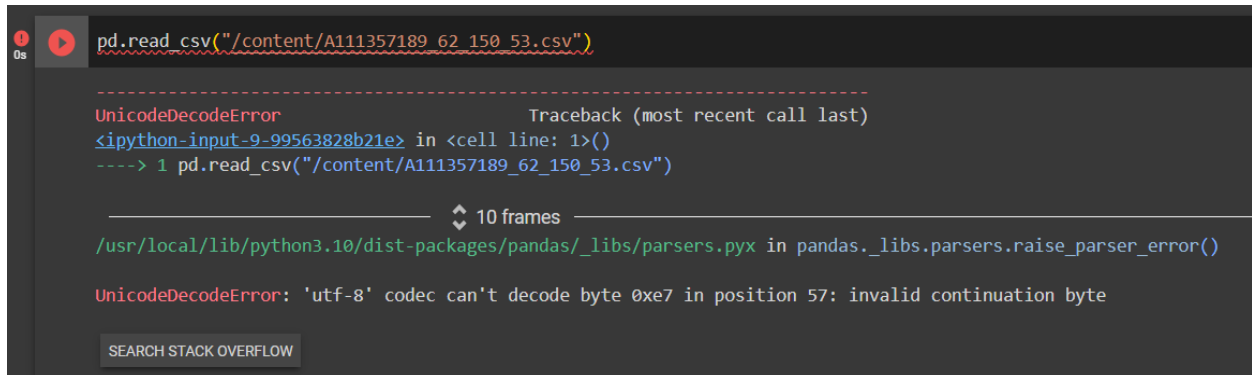
Voltamos para o nosso notebook. Como abrir um arquivo em código? Vamos usar uma biblioteca do Python...

Usando Pandas! \o/

`import pandas as pd` + Shift + Enter para rodar

`pd.read_csv("<colar aqui o link do path>")`

Não vai funcionar. Por quê? Porque o erro mais comum é de Encode, que é o formato no qual um arquivo é escrito. Tudo aquilo é binário sim, mas precisamos de um formato de código! Esse é o encoding.



```
pd.read_csv("/content/A111357189_62_150_53.csv")

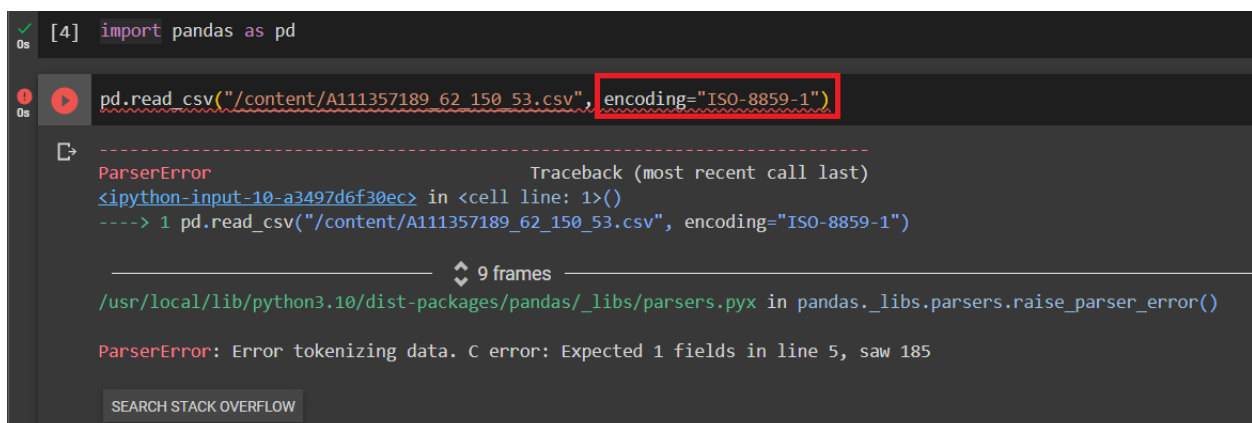
-----
UnicodeDecodeError                                Traceback (most recent call last)
<ipython-input-9-99563828b21e> in <cell line: 1>()
----> 1 pd.read_csv("/content/A111357189_62_150_53.csv")

-----
10 frames
/usr/local/lib/python3.10/dist-packages/pandas/_libs/parsers.pyx in pandas._libs.parsers.raise_parser_error()

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe7 in position 57: invalid continuation byte
```

Existe mais de um porque, historicamente, nos EUA não precisava de um Ã, na coreia não precisava do Ç, cada um tinha seu próprio padrão! O código está tentando criar um padrão uniforme, que é o 'utf-8' e não foi capaz de decodificar um padrão ali.

Mesmo colocando encoding, aparece outro erro apontando para a linha 5 da tabela.



```
[4] import pandas as pd

pd.read_csv("/content/A111357189_62_150_53.csv", encoding="ISO-8859-1")

-----
ParserError                                Traceback (most recent call last)
<ipython-input-10-a3497d6f30ec> in <cell line: 1>()
----> 1 pd.read_csv("/content/A111357189_62_150_53.csv", encoding="ISO-8859-1")

-----
9 frames
/usr/local/lib/python3.10/dist-packages/pandas/_libs/parsers.pyx in pandas._libs.parsers.raise_parser_error()

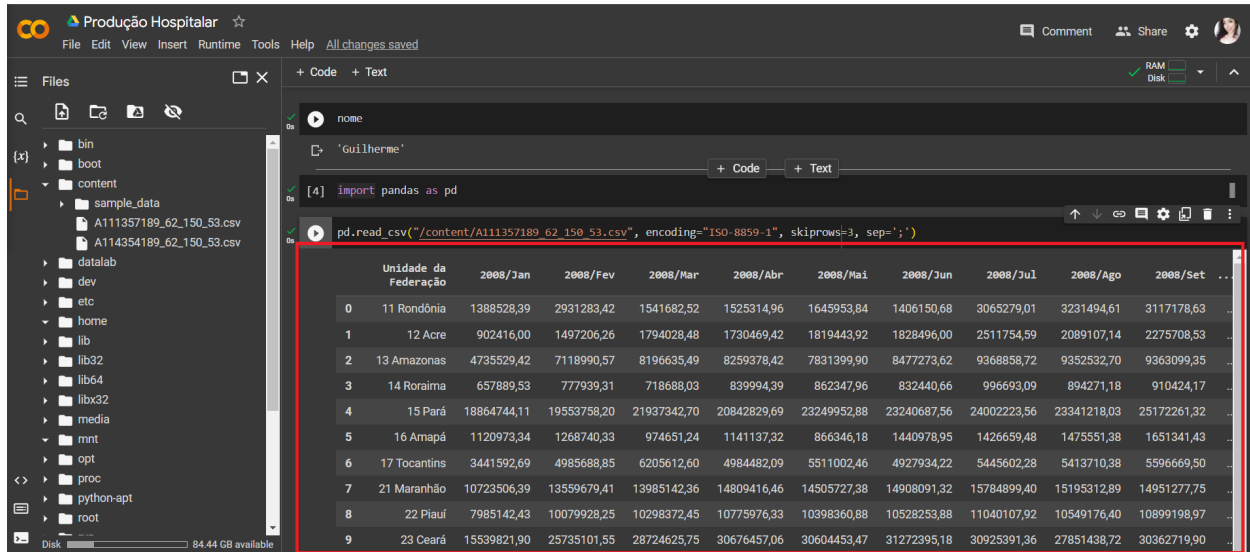
ParserError: Error tokenizing data. C error: Expected 1 fields in line 5, saw 185
```

☐ Abrir o arquivo da tabela em formato .txt

Notamos que os valores foram separados com ; e não com ,!

☐ Voltar no código e escrever `sep=';'`

Pronto, carregou muita coisa! 😊 Mas tem umas 11 linhas ali no final do documento que não quero!



The screenshot shows a Jupyter Notebook environment with a file explorer on the left and a code editor on the right. The code editor contains the following code:

```
nome  
'Guilherme'  
  
[4] import pandas as pd  
  
pd.read_csv("/content/A111357189_62_150_53.csv", encoding="ISO-8859-1", skiprows=3, sep=';')
```

The output of the code is a pandas DataFrame with the following structure:

	Unidade da Federação	2008/Jan	2008/Fev	2008/Mar	2008/Abr	2008/Mai	2008/Jun	2008/Jul	2008/Ago	2008/Set
0	11 Rondônia	1388528,39	2931283,42	1541682,52	1525314,96	1645953,84	1406150,68	3065279,01	3231494,61	3117178,63
1	12 Acre	902416,00	1497206,26	1794028,48	1730469,42	1819443,92	1828496,00	2511754,59	2089107,14	2275708,53
2	13 Amazonas	4735529,42	7118990,57	8196635,49	8259378,42	7831399,90	8477273,62	9368858,72	9352532,70	9363099,35
3	14 Roraima	657889,53	777939,31	718688,03	839994,39	862347,96	832440,66	996693,09	894271,18	910424,17
4	15 Pará	18864744,11	19553758,20	21937342,70	20842829,69	23249952,88	23240687,56	24002223,56	23341218,03	25172261,32
5	16 Amapá	1120973,34	1268740,33	974651,24	1141137,32	866346,18	1440978,95	1426659,48	1475551,38	1651341,43
6	17 Tocantins	3441592,69	4985688,85	6205612,60	4984482,09	5511002,46	4927934,22	5445602,28	5413710,38	5596669,50
7	21 Maranhão	10723506,39	13559679,41	13985142,36	14809416,46	14505727,38	14908091,32	15784899,40	15195312,89	14951277,75
8	22 Piauí	7985142,43	10079928,25	10298372,45	10775976,33	10398360,88	10528253,88	11040107,92	10549176,40	10899198,97
9	23 Ceará	15539821,90	25735101,55	28724625,75	30676457,06	30604453,47	31272395,18	30925391,36	27851438,72	30362719,90

☐ Colocar `skipfooter=12` porque quero ignorar a linha “Total”

☐ Agora, vou colocar essa tabela inteira dentro de uma variável chamada “dados”

Se eu compilar, não vai me devolver nada como output.

☐ `dados.head()` para imprimir os 5 primeiros elementos, é um dataframe do pandas;

☐ `dados.tail()` para imprimir os 5 últimos;

Se eu quiser saber a média de gastos por mês?

☐ `dados.mean()` para imprimir a média;

Aqui ele não devolve nada... pegadinha: o padrão do readcsv é que, para os milhares (thousands), ele separa com vírgula mas, no nosso caso, a gente quer com ponto. O decimal, eles separam com ponto e nós queremos com vírgula... é o contrário.

☐ No código, colocar: `thousands="."`

☐ Colocar também `decimal=","`

☐ Rodar novamente o `.mean()`

☐

```
dados.mean()
<ipython-input-21-99af3ba13e28>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will
dados.mean()
2007/Ago    5.881232e+05
2007/Set    1.519994e+06
2007/Out    4.706029e+06
2007/Nov    1.064061e+07
2007/Dez    2.379309e+07
...
2023/Jan    1.113021e+08
2023/Fev    9.016478e+07
2023/Mar    8.792502e+07
2023/Abr    4.053921e+07
Total      1.539976e+10
Length: 190, dtype: float64
```

Pronto, foi gerada a média por mês. 😊

Se eu quiser ver informações?

☐ `dados.info()` para ver uma visão geral de como o dataframe fala com a gente;

```
dados.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Columns: 312 entries, Unidade da Federação to Total
dtypes: float64(190), object(122)
memory usage: 85.4+ KB
```

Aqui, temos 34 entradas de 35 de linhas;

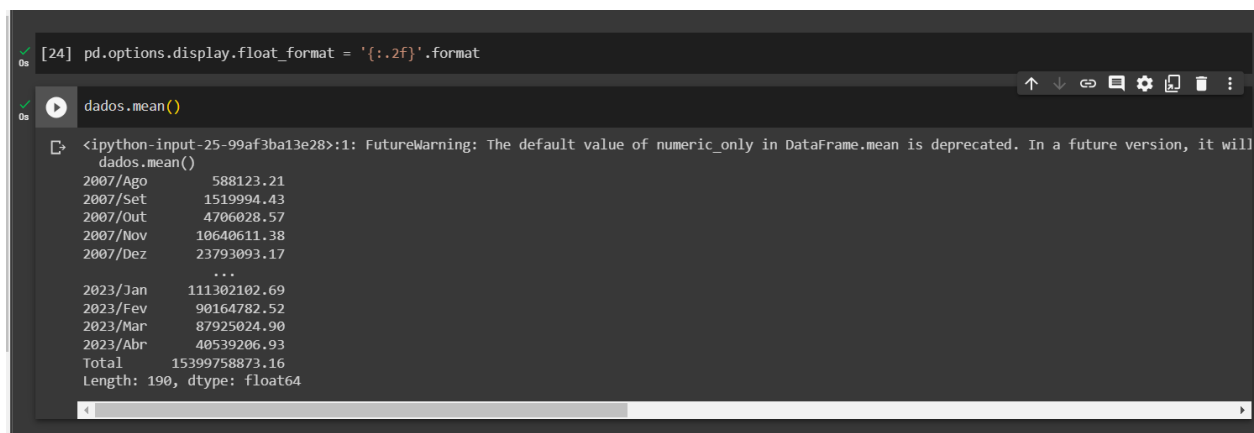
Temos 312 entradas de colunas;

Temos tipos de ponto flutuante (190) e algum tipo de objeto (122) também;

Vamos agora trabalhar com uma opção **global** do pandas para formatação...

☐ `pd.options.display.float_format = '{:,.2f}'.format` → essa é a formatação padrão do python

☐ `dados.mean()`



```
[24] pd.options.display.float_format = '{:,.2f}'.format

dados.mean()

<ipython-input-25-99af3ba13e28>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will
dados.mean()
2007/Ago      588123.21
2007/Set      1519994.43
2007/Out      4706028.57
2007/Nov      10640611.38
2007/Dez      23793093.17
...
2023/Jan      111302102.69
2023/Fev      90164782.52
2023/Mar      87925024.90
2023/Abr      40539206.93
Total        15399758873.16
Length: 190, dtype: float64
```

Aqui estamos vendo a média por mês, mas usa o ponto para formatar.

Nota-se que ao longo do vídeo obtivemos valores diferentes porque essa tabela foi mudando ao longo do tempo, temos um delay aqui de informações. 😊

Desafio: olhar apenas a coluna mais recente de todas, não o total, mas sim a coluna mais recente. Como fazer isso?

	2022/Oct	2022/Nov	2022/Dec	2023/Jan	2023/Fev	2023/Mar	2023/Abr	Total
1	1.618.556.847,70	1.542.830.354,94	1.501.800.988,03	1.558.229.437,70	1.262.306.955,25	1.230.950.348,65	567.548.897,04	215.596.624.224,21
2	10.026.702,19	10.061.542,23	9.921.628,37	9.774.862,91	6.939.913,08	6.121.142,34	2.809.110,10	1.286.713.207,96
3	4.040.340,45	4.028.748,15	3.990.231,61	3.915.217,49	3.301.011,42	3.384.939,31	1.401.760,07	150.167.864,59
4	20.121.900,40	19.374.330,26	18.194.222,38	17.421.962,57	15.268.586,91	13.108.969,95	4.407.011,87	2.431.109.386,08
5	4.409.762,46	4.391.756,17	3.307.092,09	3.335.101,99	2.449.009,26	2.247.863,29	836.884,29	421.459.020,87
6	45.498.400,27	45.863.296,96	44.721.007,44	47.007.463,78	37.790.060,31	34.611.486,72	14.416.398,87	5.861.270.009,50
7	3.321.503,57	3.032.253,40	2.836.766,19	3.016.405,19	2.758.034,08	3.631.842,52	1.823.184,04	397.186.995,74
8	9.088.239,46	8.386.412,31	7.805.215,89	4.385.841,27	3.367.889,15	2.366.899,55	226.111,49	1.277.421.198,05
9	38.614.290,55	35.773.302,32	35.967.999,90	36.918.388,00	27.682.496,87	23.619.300,84	9.437.097,82	4.875.091.790,27
10	20.546.181,25	19.886.523,30	18.461.093,41	19.057.216,16	16.248.263,09	17.244.416,46	9.105.939,36	2.845.118.452,79
11	61.606.686,16	60.197.355,94	57.463.183,93	61.220.069,28	46.249.799,42	42.097.657,36	17.094.223,01	8.268.976.821,23
12	26.695.976,44	24.582.851,45	21.830.149,96	24.576.886,61	19.826.243,68	20.996.114,33	8.056.974,26	3.374.722.049,08
13	23.859.301,89	23.707.214,71	22.702.777,24	23.097.438,24	20.166.297,58	20.542.029,04	10.457.707,79	3.252.791.816,51
14	79.993.349,57	75.866.212,48	74.506.276,64	77.265.340,54	67.763.120,05	70.513.105,61	34.576.781,98	10.779.456.862,94
15	18.567.099,98	17.581.930,80	17.192.743,09	16.333.409,94	13.129.087,83	13.395.964,04	6.570.640,67	2.735.240.932,12
16	13.651.567,43	13.280.397,80	13.651.176,71	13.267.054,42	8.201.382,08	5.086.881,34	2.274.803,87	1.562.303.965,21
17	91.138.952,91	86.310.337,05	85.454.981,47	84.973.088,82	67.154.216,25	62.190.799,05	30.716.940,30	12.203.298.024,05
18	198.247.020,98	194.245.346,43	189.208.410,36	190.744.910,08	152.293.894,79	152.145.233,50	66.724.936,31	25.619.967.806,11
19	38.695.224,83	35.714.674,03	36.114.490,61	36.044.499,92	30.158.826,25	30.091.665,76	13.025.442,16	4.410.880.310,69
20	106.996.338,55	102.863.236,39	98.384.071,29	104.234.549,24	78.988.850,46	77.746.696,70	34.715.797,10	13.168.898.651,80
21	258.478.084,13	243.431.103,77	230.839.787,20	250.705.475,66	293.918.299,77	294.741.277,18	152.211.025,08	50.975.927.460,49
22	140.275.716,28	137.011.330,55	129.061.875,96	133.886.009,41	109.507.706,20	113.834.016,08	51.134.228,39	18.384.286.981,08
23	81.636.655,45	76.754.091,23	72.936.611,52	77.798.266,97	66.303.124,06	58.108.952,29	24.507.952,14	9.958.148.895,27
24	108.954.117,24	106.363.813,89	99.999.202,88	104.126.112,74	83.687.243,14	81.288.000,01	38.614.776,79	15.695.778.721,48
25	25.110.578,25	25.995.928,39	22.781.076,34	21.723.670,65	13.895.690,55	10.135.943,20	3.464.911,18	2.977.545.547,46
26	20.271.425,43	20.133.488,72	19.316.247,01	20.174.211,64	15.567.132,19	13.619.328,17	6.040.972,96	2.776.984.815,31

Para pegar a penúltima coluna de uma tabela utilizando Python e Pandas, você pode utilizar a indexação negativa para acessar a coluna desejada. Aqui está um exemplo:

```
penultima_coluna = dados.iloc[:, -2]
print(penultima_coluna)
```

Nesse exemplo, **dados** representa o DataFrame que contém a tabela de dados carregada a partir do arquivo CSV.

→ Não tem como procurar como date, o Index é em string.