Curso de Python do ZERO AO DS

com Meigarom do canal “Seja Um Data Scientist”

Instagram: @meigarom.datascience (Mais informações sobre o Curso)
LinkedIn: https://www.linkedin.com/in/meigarom/ (Contato Profissional)
Telegram: https://t.me/sejaumdatascientist (GRUPO DE

ESTUDOS)

Aula 03 - Transformação de Dados I - Básico

Agenda:

1. Novas perguntas de negócio.
2. Planejamento da Solução.

3. Estrutura de Dados.
4. Transformação de Dados.
5. Exercícios Práticos.

1. Novas perguntas de negócio.

1.1. Recapitulando o desafio: (https://sejaumdatascientist.com/os-5-projetos-de-data-science-que-fara-o-recrutador-olhar-para-voce/)

- EMPRESA: House Rocket

- MODELO DE NEGÓCIO: Compra casas com preço baixo e revendo com o preço mais alto.

- QUAL O DESAFIO: Encontrar bons negócios dentro do portfólio disponível, ou seja, encontrar casas com preço baixo, em ótima localização e que tenham um ótimo potencial de revenda por um preço mais alto.

1.2. Novas perguntas do

CEO para você:

1. Qual o número de imóveis por ano de construção?
2. Qual o menor número de quartos por ano de construção dos imóveis?
3. Qual o preço de compra mais alto por cada número de quartos?
4. Qual a soma de todos os preços de compra por cada número de quartos?
5. Qual a soma de todos

os preços de compra pelo número de quartos e banheiros?

6. Qual o tamanho médio das salas dos imóveis por ano de construção?

7. Qual o tamanho mediano das salas dos imóveis por ano de construção?

8. Qual o desvio-padrão do tamanho das salas dos imóveis por ano de construção?

9. Como é o crescimento

médio preços de compra dos imóveis, por dia e semana do ano?

10. Eu gostaria de olhar no mapa e conseguir identificar as casas com o maior preço.

2. Planejamento da solução:

2.1. Produto Final (O que eu vou entregar? Planilha, gráfico, modelo de ML, …)

- Email + 2 anexos:
    - Email: As respostas das perguntas.
        - Pergunta I Resposta
    - Anexo 01: Um dashboard com 3 gráfico.
    - Anexo 02: A foto de um mapa 2.0 em .html

2.2. Ferramenta (Qual ferramenta usar?)

- Python 3.8.0
- Jupyter Notebook

2.3. Processo (Como fazer?)

1. Qual o número de imóveis por ano de construção?

- Contar o número de ids por ano de construção

2. Qual o menor número de quartos por ano de construção dos imóveis?

- Filtrar todos os imóveis por ano de construção e selecionar o menor número de quartos.

Daniel Azevedo de Sousa - d.azevedo.s7@gmail.com - CPF: 027.074.713-30

3. Qual o preço de compra mais alto por cada número de quartos?

- Filtrar todos os imóveis por número de quarto e selecionar o maior preço.

4. Qual a soma de todos os preços de compra por cada número de quartos?

- Filtrar todos os imóveis por número de quarto e somar todos os

preços.

5. Qual a soma de todos os preços de compra pelo número de quartos e banheiros?

- Filtrar todos os imóveis por número de quarto e banheiro e somar todos os preços.

6. Qual o tamanho médio das salas dos imóveis por ano de construção?

- Filtrar todos os

imóveis por ano de construção e fazer a média do tamanho das salas.

7. Qual o tamanho mediano das salas dos imóveis por ano de construção?

- Filtrar todos os imóveis por ano de construção e calcular a mediana do tamanho das salas.

8. Qual o desvio-padrão

do tamanho das salas dos imóveis por ano de construção?

- Filtrar todos os imóveis por ano de construção e calcular o desvio-padrão do tamanho das salas.

9. Como é o crescimento médio preços de compra dos imóveis, por dia e semana do ano?

- Filtrar todos os imóveis por data e calcular

o preço médio.

- Procurar uma Biblioteca em Python que tenha uma Função que desenhe um gráfico de linha.

- Aprender a usar a função e desenhar um a variação do preço médio por dia e semana do ano.

10. Eu gostaria de olhar no mapa e conseguir identificar as casas com o maior preço.

- Modificar o mapa da entrega anterior fazendo com que o pontos tenham o tamanho dependente do preço.

3. As ferramentas para criar códigos em Python:

- IDEs (Interface Development Environment)
    - PyCharm

- VSCode
- Spyder
- JupyterLab

- Notebooks
    - Jupyter Notebook

A principal Vantagem e Desvantagem:

- IDE’s é sempre necessário transformar TODOS os comandos em linguagem de máquina, todas vez que você executa os arquivo que

contém o seu código.

- Notebook é necessário transformar apenas os comandos escolhidos em linguagem de máquina.

Os notebooks precisam de um "ambiente" para funcionar. Esse "ambiente" possui:

- Editor de Texto (Notebook)

- O interpretador do Python.

- As bibliotecas que você

está usando.

- O ambiente mais fácil para estudantes é o Anaconda

3.1. Instalando o Anaconda no Windows:

- 80% de vocês vão desistir nesse momento (“Não consigo”, “Tá dando erro”, “Não é pra mim”, ...)
- 10% de vocês vão seguir em frente (Um

problema superado + perto do objetivo)

- 10% de vocês não vão nem tentar (Aqueles que só assistem e não estudam)

- https://www.anaconda.com/distribution/#windows
https://www.linkedin.com/pulse/tutorial-pr%C3%A1tico-de-como-instalar-anaconda-para-gomes-de-lima

3.2. Extensões do Anaconda

- conda install -c conda-forge

jupyter_contrib_nbextensions

- Codefolding
- Collapsible Headings
- Code prettify
- Execute Time
- Hide input

4. As estruturas de Dados em Python.

- As 4 estruturas de dados mais usadas em Python são:

- Listas (Mostrarei na próxima aula, Aula 04)
- Dicionários
- Tuples (Mostrarei na próxima aula, Aula 04)
- Dataframes

4.1. Dicionários:

- Armazenam dados na forma de chave-valor e não aceitam duplicados.
- Os dados são armazenas com chave-valor (“nome”: valor)
- Precisam de um

nome.

==========

==========

====

Dentro do Jupyter Notebook

#

==========

==========

Estrutura de Dados - Dicionário

#

==========

==========

# o dicionário tem a seguinte forma:
dict = {‘chave01’: valor01,
‘chave02’: valor02,
‘chave03’: valor03,
‘chave04’: valor04 }
skirt = {'size': 'M', 'price': 139.90, 'color': 'black'}
skirt = {'size': 'M', 'price': 139.90, 'color': 'black',
‘launch_date’: ’2020-01-01’}
skirt = {'size': 'M', 'price':

```
139.90, 'color': ['black',
'red', 'white']}

# acesso aos valores -> via
chave
# skirt['size']
# skirt['color'][0]

# um dicionário vazio
# skirt = {}

# Adicionar valores
# skirt['category'] =
'bottom'
```

==========

==========

4.2. Dataframes:

- Armazenam dados na forma tabular com nomes nas linhas e colunas

- Precisam de um nome.

==========

==========

==========

Dentro do Jupyter

Notebook

```
#
==========
==========
# Estrutura de Dados - Dataframes
#
==========
==========
# Um dataframe vazio
# df = pd.DataFrame()
#
# Um dataframe a partir um dicionário
```

```
# data = {'size': ['P', 'M',
'G'], 'price': [139.90, 89.90,
29.90], 'color': ['black',
'red', 'white'] }
# df = pd.DataFrame( data )
```

==========

==========

====

5. Transformação de Dados:

- Agrupamento
- Operações

matemáticas

5.1. Agrupamento:

- Sequência de 3 tarefas:
Split, Apply, Combine
(Separa, Aplica, Combina)

==========

==========

====

Dentro do Jupyter Notebook

#

Daniel Azevedo de Sousa - d.azevedo.s7@gmail.com - CPF: 027.074.713-30

```
==========
# Agrupamento
#
==========
#
print( data[ data[ 'bedrooms' ] == 0 ].shape )
#
print( data[ data[ 'bedrooms' ] == 1 ].shape )
#
print( data[ data[ 'bedrooms' ] == 2 ].shape )
```

```
#
print( data[ data[ 'bedrooms' ] == 3 ].shape )
#
print( data[ data[ 'bedrooms' ] == 4 ].shape )
#
# df_grouped = data[ ['id', 'bedrooms'] ].groupby( 'bedrooms' )
#
# Inside of groupby
# for bedrooms, frame in df_grouped:
#     print( 'Number of
```

```
bedrooms:
{}'.format( bedrooms ) )
#     print( frame.shape,
end='\n\n' )
#
```

```
==========

==========

====
```

5.2. Operações:

- Com os dados agrupados, podemos realizar operações matemáticas:
    - Exemplos de

operações matemática:

- **Contagem.**
- **Mínimo.**
- **Máximo.**
- **Soma.**
- **Média.**
- **Mediana.**
- **Desvio Padrão.**

6. Executando o PROCESSO planejado:

1. Qual o número de

imóveis por ano de construção?

- Contar o número de ids dos imóveis por ano de construção

Dentro do Jupyter Notebook

```
# Count
#df3[['id',
'yr_built']].groupby( 'yr_built' ).count()
```

==========

Daniel Azevedo de Sousa - d.azevedo.s7@gmail.com - CPF: 027.074.713-30

==========

==========

2. Qual o menor número de quartos por ano de construção dos imóveis?

- Filtrar todos os imóveis por ano de construção e selecionar o menor número de quartos.

Dentro do Jupyter Notebook

Min

```
df3[['bedrooms', 'yr_built']].groupby( 'yr_built' ).min()
```

==========

==========

====

3. Qual o preço de compra mais altos por cada número de quartos?

- Filtrar todos os imóveis por número de quarto e selecionar o maior preço.

Dentro do Jupyter Notebook

```
# Max
#df3[['price',
'bedrooms']].groupby( 'bed
rooms' ).max()
```

==========

==========

====

4. Qual a soma de todos

os preços de compra por cada número de quartos?

- Filtrar todos os imóveis por número de quarto e somar todos os preços.

Dentro do Jupyter Notebook

```
# Soma
#df3[['price', 'bedrooms']].groupby( 'bedrooms' ).sum()
```

==========

==========

====

5. Qual a soma de todos os preços de compra pelo número de quartos e banheiros?

- Filtrar todos os imóveis por número de quarto e banheiro e somar todos os preços.

Dentro do Jupyter

Notebook

```
#df3[['price', 'bedrooms',
'bathrooms']].groupby( ['be
drooms',
'bathrooms'] ).sum()
#df3[['price', 'bedrooms',
'bathrooms']].groupby( ['ba
throoms',
'bedrooms'] ).sum()
```

==========

==========

====

6. Qual o tamanho médio

das salas dos imóveis por ano de construção?

- Filtrar todos os imóveis por ano de construção e fazer a média do tamanho das salas.

Dentro do Jupyter Notebook

```
# Media
#df3[['sqft_living', 'yr_built']].groupby( 'yr_built' ).mean()
#df3[['sqft_living',
```

```
'bedrooms',
'yr_built']].groupby( ['yr_bu
ilt', 'bedrooms'] ).mean()
```

==========

==========

====

7. Qual o tamanho mediano das salas dos imóveis por ano de construção?

- Filtrar todos os imóveis por ano de construção e calcular a

mediana do tamanho das salas.

Dentro do Jupyter Notebook

```
# Mediana
#df3[['sqft_living', 'yr_built']].groupby( 'yr_built' ).median()
#df3[['sqft_living', 'bedrooms', 'yr_built']].groupby( ['yr_built', 'bedrooms'] ).median()
```

```
# -------- Bonus --------
#df3[['sqft_living',
'bedrooms',
'yr_built']].groupby( ['yr_bu
ilt',
'bedrooms'] ).agg( ['max',
'min', 'mean', 'median'] )
```

==========

==========

=====

8. Qual o desvio-padrão do tamanho das salas dos

imóveis por ano de construção?

- Filtrar todos os imóveis por ano de construção e calcular o desvio-padrão do tamanho das salas.

Dentro do Jupyter Notebook

```
# Desvio Padrao
#df3[['sqft_living', 'yr_built']].groupby( 'yr_built' ).std()
```

```
#df3[['sqft_living',
'bedrooms',
'yr_built']].groupby( ['yr_bu
ilt', 'bedrooms'] ).std()
```

==========

==========

====

9. Como é o crescimento médio preços de compra dos imóveis, por dia, mês e ano?

- Filtrar todos os imóveis por data e calcular

o preço médio.

- Procurar uma Biblioteca em Python que tenha uma Função que desenhe um gráfico de linha.

- Aprender a usar a função e desenhar um a variação do preço médio por dia e semana do ano.

Dentro do Jupyter Notebook - ORIGINAL

```
# First Graph
df['year'] =
pd.to_datetime( df['date'] ).
dt.year
by_year = df[['id',
'year']].groupby( 'year' ).me
an().reset_index()
plt.figure( figsize=(20,10))
plt.bar( by_year['year'],
by_year['id'] )

# Second Graph
df['day'] =
pd.to_datetime( df['date'] )
by_day = df[['id',
```

```
'day']].groupby( 'day' ).mea
n().reset_index()
plt.figure( figsize=(20,10))
plt.plot( by_day['day'],
by_day['id'] )

# Thrid Graph
df['year_week'] =
pd.to_datetime( df['date'] ).
dt.strftime( '%Y-%U')
by_week_of_year = df[['id',
'year_week']].groupby( 'yea
r_week' ).mean().reset_ind
ex()
plt.figure( figsize=(20,10))
```

```
plt.plot( by_week_of_year['
year_week'],
by_week_of_year['id'] )
plt.xticks( rotation=60 );
```

DASHBOARD

```
from matplotlib import
pyplot as plt
from matplotlib import
gridspec
fig = plt.figure( figsize=(24,
12) )
specs =
gridspec.GridSpec( ncols=
```

```
2, nrows=2, figure=fig )

ax1 =
fig.add_subplot( specs[0, :
] ) # First Row
ax2 =
fig.add_subplot( specs[1,
0] ) # First Row First
Column
ax3 =
fig.add_subplot( specs[1,
1] ) # Second Row First
Column

# Frist Graph
```

```
df['year'] =
pd.to_datetime( df['date'] ).
dt.year
by_year = df[['id',
'year']].groupby( 'year' ).su
m().reset_index()
ax1.bar( by_year['year'],
by_year['id'] )
ax1.set_title( "Title: Sum
Price by Year" )


# Second Graph
df['day'] =
pd.to_datetime( df['date'] )
```

```
by_day = df[['id',
'day']].groupby( 'day' ).mea
n().reset_index()
ax2.plot( by_day['day'],
by_day['id'] )
ax2.set_title( "Title:
Average Price by Day" )

# Thrid Graph
df['year_week'] =
pd.to_datetime( df['date'] ).
dt.strftime( '%Y-%U')
by_week_of_year = df[['id',
'year_week']].groupby( 'yea
r_week' ).mean().reset_ind
```

```
ex()
ax3.plot( by_week_of_year
['year_week'],
by_week_of_year['id'] )
plt.xticks( rotation=60 );
```

==========

==========

====

10. Eu gostaria de olhar no mapa e conseguir identificar as casas com o maior preço.

- Modificar o mapa da

entrega anterior fazendo com que o pontos tenham o tamanho dependente do preço.

Dentro do Jupyter Notebook

```
import plotly.express as px
houses = df3[['id', 'lat', 'long', 'price']]
fig = px.scatter_mapbox( houses,
                        lat="lat",
```

```
                        lon="long",

size="price",

color_continuous_scale=p
x.colors.cyclical.IceFire,

size_max=15,
                        zoom=10)

fig.update_layout(mapbox
_style="open-street-map")
fig.update_layout(height=6
00,
margin={"r":0,"t":0,"l":0,"b
```

```
":0})
fig.show()

mapa.write_html( ‘datasets
/
mapa_house_rocket.html’ )
```

==========

==========

====

7. Exercícios:

Novas perguntas do CEO

para você:

1. Crie uma nova coluna chamada: “dormitory_type”

    - Se o valor da coluna “bedrooms” for igual à 1 => ‘studio’

    - Se o valor da coluna “bedrooms” for igual a 2 => ‘apartament’

    - Se o valor da coluna “bedrooms” for maior que 2 => ‘house’

2. Faça um gráfico de

Daniel Azevedo de Sousa - d.azevedo.s7@gmail.com - CPF: 027.074.713-30

barras que represente a soma dos preços pelo número de quartos.

3. Faça um gráfico de linhas que represente a média dos preços pelo ano construção dos imóveis.

4. Faça um gráfico de barras que represente a média dos preços pelo tipo dos dormitórios.

5. Faça um gráfico de

linha que mostre a evolução da média dos preços pelo ano da reforma dos imóveis, a partir do ano de 1930.

6. Faça um tabela que mostre a média dos preços por ano de construção e tipo de dormitórios dos imóveis.

7. Crie um Dashboard com os gráficos das

questões 02, 03, 04
(Dashboard: 1 Linha e 2 colunas)

8. Crie um Dashboard com os gráficos das perguntas 02, 04
(Dashboard: 2 colunas)

9. Crie um Dashboard com os gráficos das perguntas 03, 05
(Dashboard: 2 Linhas)

10. Faça um gráfico com

o tamanho dos pontos sendo igual ao tamanho da sala de estar