



48%

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



15.3k xp

a

03

Investigando a sazonalidade

Transcrição

A Chocolura possui diversas lojas espalhadas pelo país, cada uma com seus gerentes, suas regras de negócios e seus vendedores.

Uma loja em especial chamou muito a atenção dos diretores da empresa pelas vendas diárias nos meses de Outubro e Novembro de 2018. Tentaremos analisar o que terá acontecido.

Criaremos uma célula do tipo texto clicando o botão "Text" e digitaremos "#Chocolura - Vendas diárias (Outubro e Novembro)".

Utilizaremos o arquivo "vendas_por_dia.csv" em "Files".

Como tem sido comum em nossas aulas, será necessário carregá-lo e atribuí-lo à variável "vendas_por_dia" para ler. `vendas_por_dia = pd.read_csv('vendas_por_dia.csv')` Veremos os 5 primeiros registros escrevendo `vendas_por_dia.head()`.

Teremos uma coluna para o dia e outra para a quantidade de vendas referente a ele dessa loja.

Vamos conferir com prints se há algum dado nulo e quantos são, além do número de linhas.



48%

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



15.3k xp

a

```
print ('Quantidade de linhas:', vendas_por_dia.shape)
print ('Quantidade de dados nulos:', vendas_por_dia.isna().sum().sum())
```

COPIAR CÓDIGO

Serão 61 linhas e 2 colunas, ou seja, os meses inteiros de Outubro e Novembro. Não teremos dados nulos e poderemos continuar nossa análise.

Conferiremos os tipos com `vendas_por_dia.dtypes`. O dia estará tipado como *object* do pandas, então, vamos transformá-lo em *datetime64* por meio de `vendas_por_dia['dia'] = pd.to_datetime(vendas_por_dia['dia'])`.

Com `vendas_por_dia.dtypes` novamente, fazemos a conferência de que o tipo do dia foi alterado corretamente para *datetime64*.

Faremos, então, um `diff` do aumento e das vendas para descobrir a aceleração.

```
vendas_por_dia['aumento'] = vendas_por_dia['vendas'].diff()
vendas_por_dia['aceleracao'] = vendas_por_dia['aumento'].diff()
```

COPIAR CÓDIGO

Nos certificando de que o procedimento foi bem sucedido, escreveremos `vendas_por_dia.head()`. Teremos as colunas de aumento e aceleração agora. O aumento no dia 2 será -2.0, levando em conta que no dia 1 as vendas foram 50 e no dia 2, 48. Então, $48 - 50 = -2$, não tendo havido aceleração. No dia



48%

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODOS
NOTURNO



15.3k xp



3, as vendas foram 56. 56 menos as 48 vendas do dia 2 será um aumento de 8. A aceleração será $8 - (-2) = 10$, e assim por diante.

Agora, faremos o plot de comparação dos três gráficos. Escrevendo "plot_" e dando um "Tab" o programa já completará com a comparação e os parâmetros "plot_comparacao (x, y1,, y2, y3, dataset, titulo)". Nós os definiremos como:

```
plot_comparacao('dia', 'vendas', 'aumento', 'aceleracao', vendas_por_dia, 'Análise de vendas de Outubro e Novembro - Choccolura') .
```

Executaremos e os gráficos serão plotados. Vamos analisá-los e serão bem diferentes do que estamos acostumados. As vendas terão um padrão sazonal mais uma vez, pois estão repetidamente aumentando e diminuindo na sequência, mas diferentemente do gráfico da aula anterior, isso não ocorrerá especificamente em duas datas do comemorativas. Teremos que explorar por qual motivo isso estará acontecendo.

No aumento também haverá o padrão repetitivo de crescimento significativo e queda. Até mesmo a aceleração carregará o reflexo desse padrão.

Criaremos uma nova célula de tipo texto para começar a análise desses gráficos. Clicaremos no botão "Text" da barra superior do Colab e digitaremos na célula " ****Analisando a sazonalidade**** ".

Da mesma forma que no gráfico anterior pesquisamos se houve alguma data comemorativa nos meses em que as vendas foram maiores, vamos avaliar se houve datas específicas primeiro em Outubro de 2018. O dia 1° foi uma segunda-feira, dia 2 terça. Podemos estipular uma nova coluna para tentar descobrir o que estará acontecendo com as vendas por meio de uma análise de acordo com dias da semana.



48%

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



15.3k xp

a

Para escrever o código que gera a coluna nova, pegaremos nosso DataFrame e a coluna do dia, e precisaremos saber qual será o dia da semana relacionado a esse dia do Data. Existe uma função para isso, a `dt`.

```
vendas_por_dia['dia_da_semana'] = vendas_por_dia['dia'].dt.day_name(),
```

Para conferir, digitaremos:

```
vendas_por_dia.head()
```

Terá sido acrescentada essa coluna de dias da semana, relacionados aos 5 primeiros registros, pelo que poderemos observar. Podemos passar 7 dias para ter certeza de que foi uma semana

```
vendas_por_dia.head(7)
```

Porém, os dados estarão todos em inglês (Monday, Tuesday, Wednesday, etc.). Vamos simplificar deixando-os em Português. Pegaremos todos os dados de dias da semana e usaremos o `unique`, que nos retornará todos os valores únicos quanto aos dias da semana.

```
vendas_por_dia['dia_da_semana'].unique()
```

COPIAR CÓDIGO

Será devolvido: array (['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], dtype=object).

Então, criaremos um dicionário próprio com esses valores. Geraremos uma variável chamada `dias_traduzidos`. Daremos "Ctrl C + Ctrl V" para copiar os dias em Inglês, e colocaremos dois pontos (:) para passarmos traduções.



48%

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



15.3k xp

a

```
dias_traduzidos = {'Monday': 'Segunda', 'Tuesday': 'Terça', 'Wednesday': 'Quarta',  
'Thursday': 'Quinta', 'Friday': 'Sexta', 'Saturday': 'Sábado', 'Sunday': 'Domingo'}
```

Agora, precisaremos mapear esses dias da semana pelos dias da semana traduzidos. Na sequência já visualizaremos por 2 semanas se a alteração foi efetuada corretamente, passando 14 dias para o `head`.

```
vendas_por_dia['dia_da_semana'] = vendas_por_dia['dia_da_semana'].map(dias_tradu  
vendas_por_dia.head(14)
```

COPIAR CÓDIGO



Podemos observar que nos sábados e domingos e no feriado do dia 12 de Outubro as vendas foram bem menores do que em comparação com os dias da semana. Isso pode significar que a empresa durante esses dias não abra durante todo o horário comercial, ou não disponibilize todos os seus vendedores, porque há vendas, mas elas são diminutas.

Para conseguir afirmar categoricamente se as vendas sempre são menores aos finais-de-semana, vamos agrupar e obter médias dos valores de cada dia da semana. Faremos isso não só para as médias das vendas, como também para o aumento e a aceleração.

Criaremos uma célula de tipo texto clicando no botão "Text", escreveremos uma hashtag (#) e o título "Agrupando os dias". Agruparemos os dias da semana e calcularemos as médias. Antes, criaremos uma variável, `vendas_agrupadas`. `groupby` servirá para agrupar e `mean` para a média. O `round` será responsável por arredondar os números obtidos.



48%

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



15.3k xp

a

```
vendas_agrupadas = vendas_por_dia.groupby('dia_da_semana')['vendas', 'aumento',  
'aceleracao'].mean().round()
```

Executaremos esse código No Domingo, a média das vendas será 19.0 e do aumento, 3.0. No sábado, a média das vendas será 16 e a do aumento -28.0. Então sábado de fato será o dia mais fraco para as vendas. Nos demais dias, exceto por Sexta (45.0), Sábado e Domingo, a média das vendas será sempre maior do que 50.

Então, podemos concluir analisando a sazonalidade que as quedas representadas no gráfico serão referentes aos finais-de-semana. Por alguma regra dos negócios da empresa, esses dias, contando com feriados, contaram bastante para a tendência que nosso gráfico apresentará.