



# PROJETO FINAL DATA ANALYST



Centro para o Desenvolvimento  
de Competências Digitais



Reskilling 4  
Employment

# A NOSSA EQUIPA

MEMBROS DO PROJETO



TIAGO MARQUES



ANA CATARINA



GONÇALO REIS



JOANA AFONSO



CESAE DIGITAL

# NEW YORK TAXIS

CLIENTE WORTEN



# O PROJETO



Taxi & Limousine Commission

ESTE PROJETO É UMA ANÁLISE DETALHADA RELATIVA A VIAGENS DE TÁXI NA CIDADE DE NOVA IORQUE.

ABRANGENDO TRÊS TIPOS DE VEÍCULOS DISTINTOS AO LONGO DE QUATRO MESES.



# O PROJETO



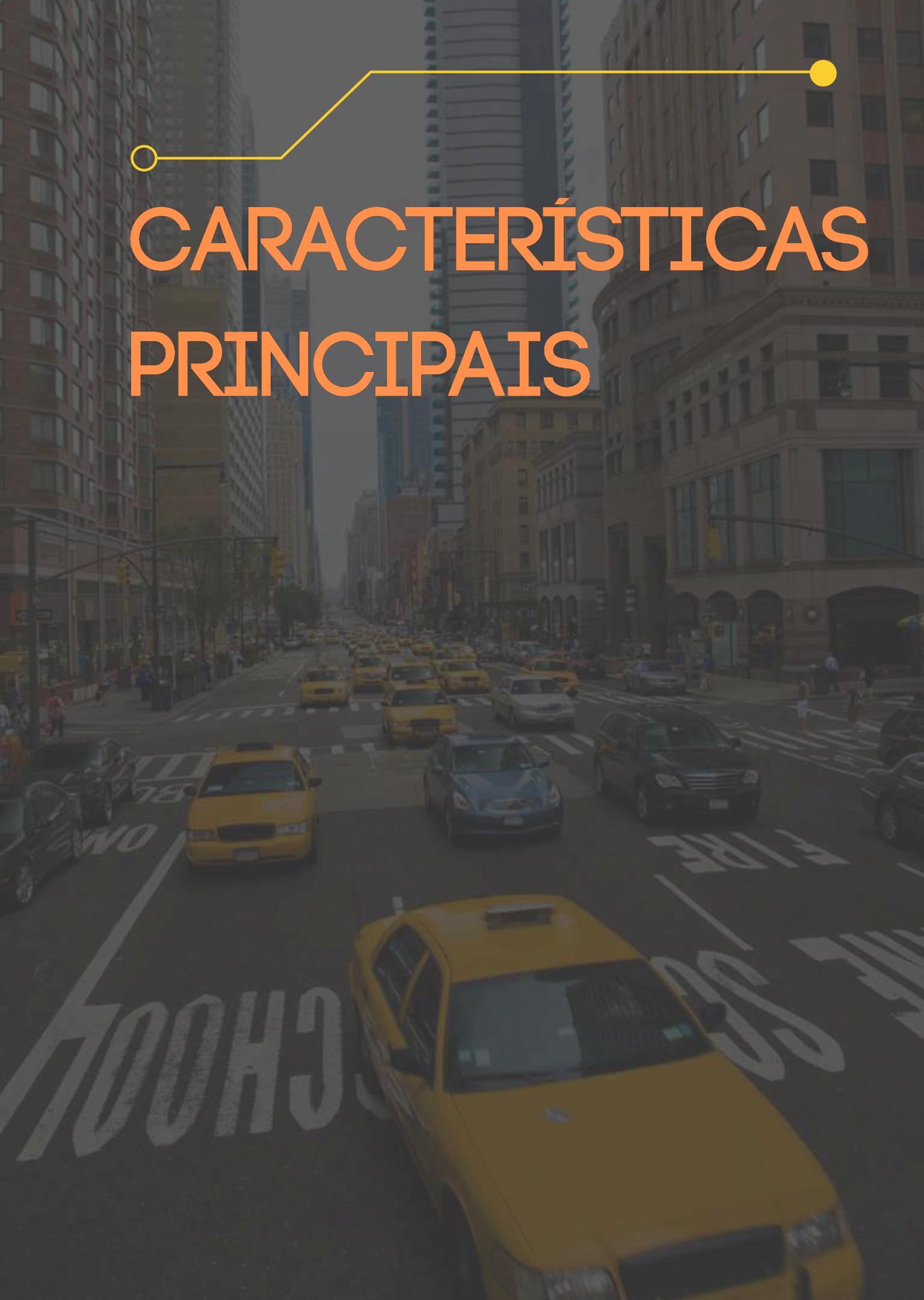
- TÍPICOS TÁXIS DE NOVA YORK.
- SERVIÇO MAIORITARIAMENTE NO CENTRO DA CIDADE.
- SERVIÇO DE CHAMADA NA RUA.
- MEDALLION (IDENTIFICAÇÃO) ÚNICA PARA CADA TÁXI.
- O PROPRIETÁRIO PODE SER O CONDUTOR OU LEASING DE AGÊNCIA.



- OS TÁXIS VERDES PRESTAM SERVIÇO PREVIAMENTE AGENDADO.
- SERVIÇO DE CHAMADA NA RUA.
- FUNCIONAM MAIORITARIAMENTE NOS DISTRITOS EXTERIORES.



- OS VEÍCULOS FHV INCLUEM CARROS E LIMOUSINES DE LUXO.
- SERVIÇO UBER, JUNO, LYFT E VIA
- FORNECEM APENAS SERVIÇO PREVIAMENTE AGENDADO



# CARACTERÍSTICAS PRINCIPAIS



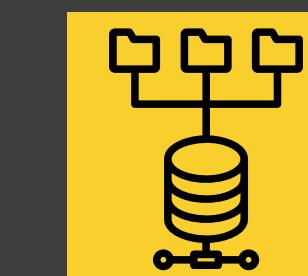
## OBJETIVO

Extrair insights valiosos e resposta aos desafios do cliente



## FORMATO DE DADOS

Apache Parquet



## FONTE DE DADOS

[www.nyc.gov](http://www.nyc.gov)



## TAMANHO DOS DADOS

10 GB

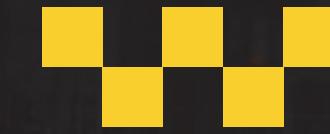


## CONTEXTO TEMPORAL

Jan 2015 & Jan Feb Mar 2016

# TAMANHO DO DATASET

10 GB

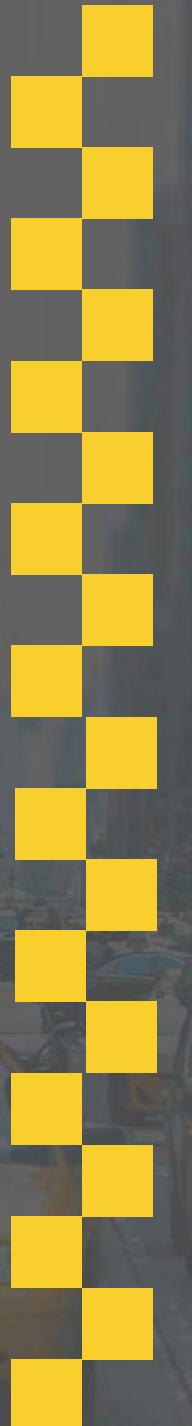


100M LINHAS



# DESAFIOS PROPOSTOS

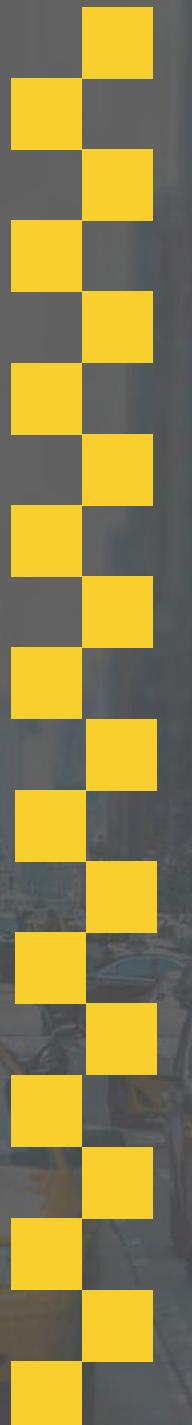
- ❖ Encontre a distribuição dos montantes das tarifas, ou seja, para cada montante A, o número de viagens que custam A.
- ❖ Encontre o número de viagens com um custo total inferior a \$10.
- ❖ Encontre a distribuição do número de passageiros.
- ❖ Encontre a receita total (para todos os táxis) por dia. A receita deve incluir o valor da tarifa, gorjetas, impostos, portagens, sobretaxas.



- ❖ Encontre o número total de viagens para cada táxi.
- ❖ Existe mais do que um registo para um determinado táxi ao mesmo tempo?
- ❖ Para cada táxi, qual é a percentagem de viagens sem coordenadas GPS, ou seja, todas as 4 coordenadas são registadas como 0's?
- ❖ Qual é a distância média percorrida por viagem?

# DESAFIOS PROPOSTOS

- ❖ Comparar viagens com base no tipo de medallion (Motorista Nomeado, Proprietário tem de conduzir).
- ❖ Listar os 10 melhores agentes por receita total.
- ❖ Qual é o número médio de viagens que podemos esperar esta semana?
- ❖ Qual é a tarifa média por viagem que esperamos cobrar?
- ❖ Determine o número de taxis diferentes utilizados por cada motorista



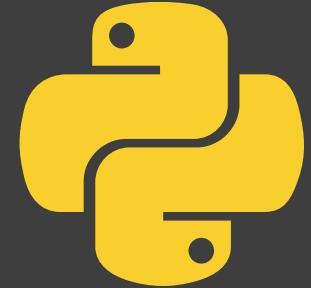
- ❖ Como é que se prevê que o volume de viagens se altere em relação à semana passada?
- ❖ Que dias da semana e horas do dia serão mais movimentados?
- ❖ Compare as viagens com base no tipo de veículo: WAV, HYB, CNG, LV1, DSE, NRML.
- ❖ Quais serão provavelmente os locais de recolha e entrega mais populares?

# PLANEAMENTO DO PROJETO

DEFINIÇÃO DAS DIFERENTES FASES DO PROJETO.



CONVERSÃO DOS FICHEIROS  
PARQUET



IMPORTAÇÃO DOS FICHEIROS  
PARA SQL SERVER



REALIZAÇÃO DE QUERIES EM SQL



IMPORTAR QUERIES PARA POWER  
BI



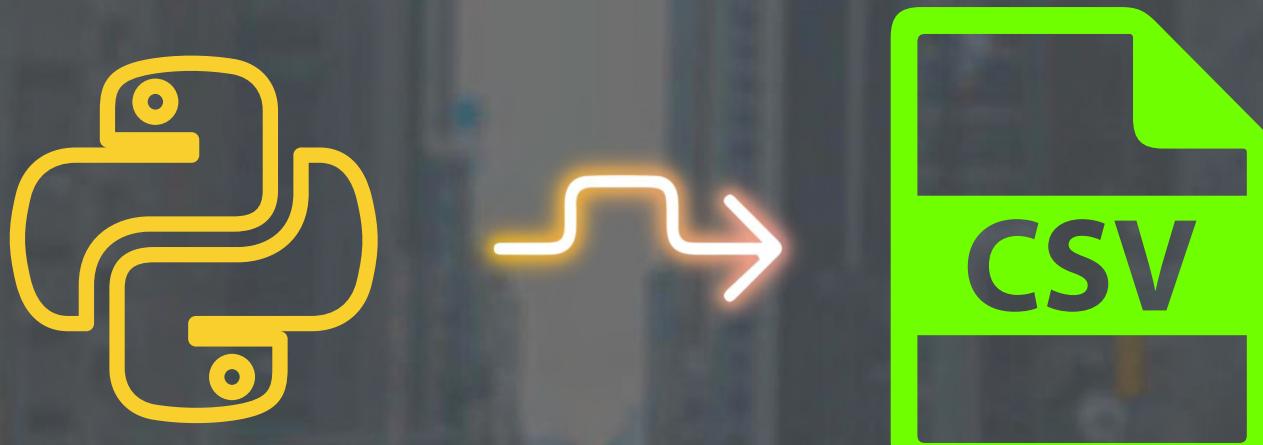
MODELAÇÃO, ANÁLISE E  
CRIAÇÃO DE DASHBOARD EM  
POWER BI



# IMPORTAÇÃO DOS DADOS |

CONVERÇÃO DE PARQUET PARA CSV COM PYTHON.

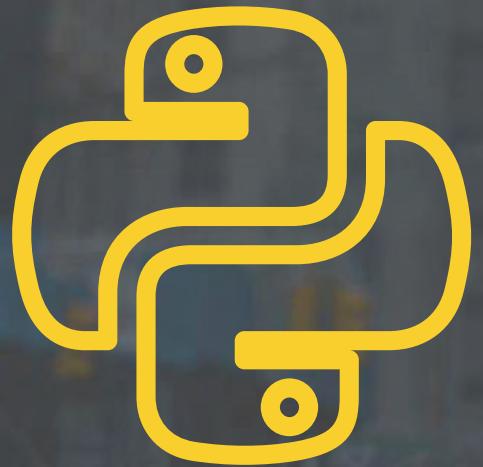
1



```
import pandas as pd\n\n# #converter yellow cabs\ndf = pd.read_parquet('yellow_tripdata_2015-01.parquet')\n\ndf.to_csv('yellow_tripdata_2015-01.csv')
```

# IMPORTAÇÃO DOS DADOS | 2

OBTENÇÃO DE INFORMAÇÕES SOBRE O DATASET COM PYTHON.



```
# Dicionário para armazenar informações sobre os datasets
datasets_info = {}

# Loop sobre os arquivos CSV
for file in csv_files:
    df = pd.read_csv(file)
    nas_per_column = df.isna().sum() # Retorna a quantidade de valores ausentes por coluna
    nas_columns = nas_per_column[nas_per_column > 0].index.tolist() # Obtém os nomes das colunas com valores ausentes
    datasets_info[file] = {
        'linhas': df.shape[0], # Quantidade de linhas
        'colunas': df.shape[1], # Quantidade de colunas
        'nas': df.isna().sum().sum(), # Soma total de valores ausentes
        'nas_colunas': nas_columns, # Nomes das colunas com valores ausentes
        'nomes_colunas': df.columns.tolist() # Nomes de todas as colunas
    }
```

# IMPORTAÇÃO DOS DADOS | 3

IMPORTAÇÃO DOS FICHEIROS CSV PARA SQL SERVER.



Import Wizard

This wizard allows you to specify details on how should import your data. Which import data format would you like?

Import type:

- DBase file (\*.dbf)
- Paradox file (\*.db)
- Text file (\*.txt)
- CSV file (\*.csv)
- Excel file (\*.xls; \*.xlsx)
- XML file (\*.xml)
- JSON file (\*.json)
- MS Access database (\*.mdb; \*.accdb)
- ODBC

Import Wizard

There are some guesses made on your table structure and you can make adjustments now.

Source Table: green\_tripdata\_2015-01

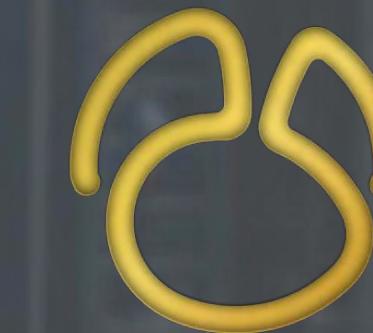
Target Table: green\_tripdata\_2015\_jan

	Source Field	Target Field	Type	Length	Scale	Primary Key
▶	f1	f1	int	0	0	🔑
✓	VendorID	VendorID	int	0	0	
✓	lpep_pickup	lpep_pickup	datetime	0	0	
✓	lpep_dropoff	lpep_dropoff	datetime	0	0	
✓	store_and_fw	store_and_fw	nvarchar	255	0	
✓	RatecodeID	RatecodeID	int	0	0	
✓	PULocationID	PULocationID	int	0	0	
✓	DOLocationID	DOLocationID	int	0	0	
✓	passenger_c	passenger_c	int	0	0	
✓	trip_distance	trip_distance	float	0	0	
✓	fare_amount	fare_amount	float	0	0	
✓	extra	extra	float	0	0	

<< < Back Next > >> Cancel

# IMPORTAÇÃO DOS DADOS | 4

ACESSO AOS DADOS POR MEIO DE APLICAÇÕES DE DATA MANAGEMENT.



Navicat  
Premium



Microsoft®  
SQL Server  
Management



# LIMPEZA DOS DADOS | 5

LIMPEZA DE NULOS E VAZIOS NO DATASET.



```
-- Eliminar linhas dos yellow com total_amount <= 0

DELETE FROM [WortenTaxi].[dbo].[yellow_tripdata_15_jan]
WHERE total_amount <= 0;

DELETE FROM [WortenTaxi].[dbo].[yellow_tripdata_16_jan]
WHERE total_amount <= 0;

DELETE FROM [WortenTaxi].[dbo].[yellow_tripdata_16_fev]
WHERE total_amount <= 0;

DELETE FROM [WortenTaxi].[dbo].[yellow_tripdata_16_mar]
WHERE total_amount <= 0;
```

# ORGANIZAÇÃO DO DATASET

| 6

## CRIAÇÃO DE VIEWS

```
1 CREATE VIEW [Total Amount_e_Date] AS
2 SELECT
3     *
4 FROM
5     [WortenTaxi].[dbo].[Total Amount Green]
UNION ALL
6 SELECT
7     *
8 FROM
9     [WortenTaxi].[dbo].[Total Amount Yellow];
10 SELECT COUNT
11     ( total_amount )
12 FROM
13     [WortenTaxi].[dbo].[Total Amount_e_Date];
```

```
CREATE VIEW [Yellow_dates] AS
SELECT
    [Taxi_Cond_ID],
    [total_amount],
    [tpep_pickup_datetime],
    [tpep_dropoff_datetime],
    [trip_distance],
    [PUlocationID],
    [DOlocationID],
    [tip_amount],
    [payment_type],
    [fare_amount],
    [passenger_count]
FROM
    [WortenTaxi].[dbo].[yellow_tripdata_15_jan]
UNION ALL
SELECT
    [Taxi_Cond_ID],
    [total_amount],
    [tpep_pickup_datetime],
    [tpep_dropoff_datetime],
    [trip_distance],
    [PUlocationID],
    [DOlocationID],
    [tip_amount],
    [payment_type],
    [fare_amount],
    [passenger_count]
FROM
    [WortenTaxi].[dbo].[yellow_tripdata_16_jan]
UNION ALL
SELECT
    [Taxi_Cond_ID],
    [total_amount],
    [tpep_pickup_datetime],
    [tpep_dropoff_datetime],
    [trip_distance],
    [PUlocationID],
    [DOlocationID],
    [tip_amount],
    [payment_type],
    [fare_amount],
    [passenger_count]
FROM
    [WortenTaxi].[dbo].[yellow_tripdata_16_fev]
UNION ALL
SELECT
    [Taxi_Cond_ID],
    [total_amount],
    [tpep_pickup_datetime],
    [tpep_dropoff_datetime],
    [trip_distance],
    [PUlocationID],
    [DOlocationID],
    [tip_amount],
    [payment_type],
    [fare_amount],
    [passenger_count]
FROM
    [WortenTaxi].[dbo].[yellow_tripdata_16_mar];
```

# ANÁLISE DO DATASET |

RESPONDER ÀS QUESTÕES POSSÍVEIS.

6

COLUMN_NAME	DATA_TYPE
airport_fee	varchar
congestion_surcharge	varchar
DOLocationID	int
extra	double
f1	int
fare_amount	double
improvement_surcharge	double
mta_tax	double
passenger_count	int
payment_type	int
PULocationID	int
RatecodeID	int
store_and_fwd_flag	varchar
tip_amount	double
tolls_amount	double
total_amount	double
tpep_dropoff_datetime	datetime
tpep_pickup_datetime	datetime
trip_distance	double
VendorID	int

21 rows in set (0.02 sec)

AS PERGUNTAS A QUE  
TINHAMOS RESPOSTA

10 / 17

DADOS QUE  
PRECISAVAMOS

Condutores  
Medallion  
taxis  
agentes  
tipo taxi

# ANÁLISE DO DATASET |

6

PESQUISA DE FONTES DE DADOS EXTERNAS PARA CONSEGUIR RESPONDER A TODAS AS QUESTÕES DO CLIENTE.

COLUMN_NAME	DATA_TYPE
Agent Address	varchar
Agent Name	varchar
Agent Number	int
Agent Telephone Number	varchar
Agent Website Address	varchar
Current Status	varchar
DMV License Plate Number	varchar
ExpirationDate	varchar
Last Date Updated	varchar
Last Time Updated	time
License Number	varchar
Medallion Type	varchar
Model Year	int
Name	varchar
Vehicle Type	varchar
Vehicle VIN Number	varchar

16 rows in set (0.01 sec)

ONDE FOMOS BUSCAR  
OS DADOS EXTRA?



[NYC OPEN DATA](#)

# ANÁLISE DO DATASET |

6

DIVISÃO DA TABELA MEDALLIONS EM DUAS DIFERENTES.

COLUMN_NAME	DATA_TYPE
Agent Address	varchar
Agent Name	varchar
Agent Number	int
Agent Telephone Number	varchar
Agent Website Address	varchar
Current Status	varchar
DMV License Plate Number	varchar
ExpirationDate	varchar
Last Date Updated	varchar
Last Time Updated	time
License Number	varchar
Medallion Type	varchar
Model Year	int
Name	varchar
Vehicle Type	varchar
Vehicle VIN Number	varchar

16 rows in set (0.01 sec)

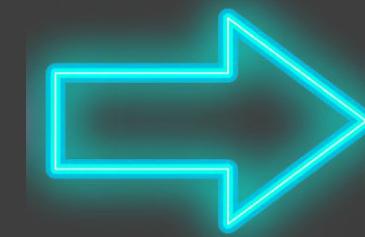


TABELA TAXIS  
DISTINTOS

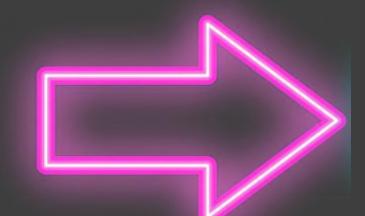


TABELA CONDUTORES  
DISTINTOS

# ANÁLISE DO DATASET |

6

TABELA TAXIS DISTINTOS E CONDUTORES DISTINTOS.

TABELA TAXIS  
DISTINTOS

COLUMN_NAME	DATA_TYPE
Agent Name	varchar
Agent Number	int
Taxi_ID	int
Vehicle Type	varchar
Vehicle VIN Number	varchar

TABELA CONDUTORES  
DISTINTOS

COLUMN_NAME	DATA_TYPE
Condutores_ID	int
License Number	varchar
Medallion Type	varchar
Name	varchar

# ANÁLISE DO DATASET

6

UNIÃO DAS DUAS TABELAS E CRIAÇÃO DA TABELA TAXI/COND ID.

TABELA TAXIS DISTINTOS

COLUMN_NAME	DATA_TYPE
Agent Name	varchar
Agent Number	int
Taxi_ID	int
Vehicle Type	varchar
Vehicle VIN Number	varchar

TABELA CONDUTORES  
DISTINTOS

COLUMN_NAME	DATA_TYPE
Condutores_ID	int
License Number	varchar
Medallion Type	varchar
Name	varchar



TABELA TAXI COND ID

COLUMN_NAME	DATA_TYPE
Associated License Number	varchar
Taxi_Cond_ID	int
Vehicle VIN Number	varchar

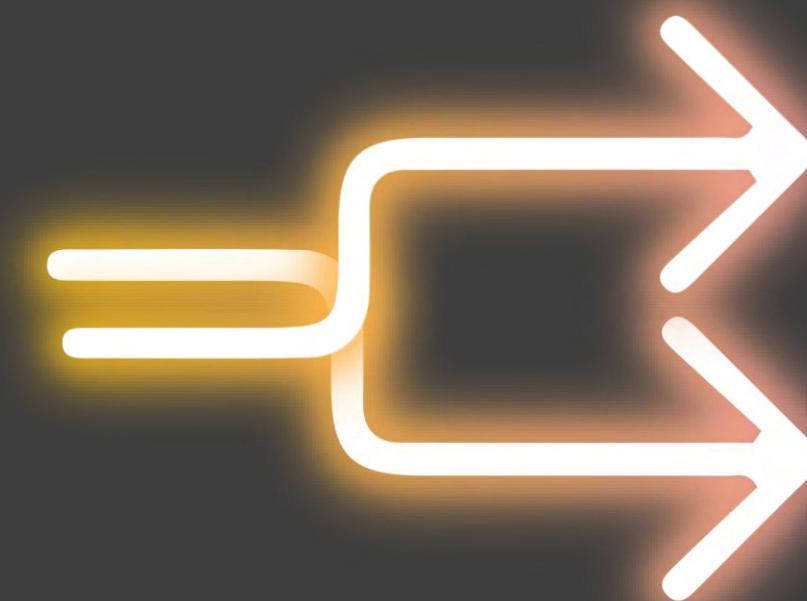
# ANÁLISE DO DATASET

6

ATRIBUIÇÃO DO TAXI\_COND\_ID ALEATÓRIAMENTE AO DATASET ORIGINAL.

TABELA TRIP ID

COLUMN_NAME	DATA_TYPE
Associated License Number	varchar
<b>Taxi_Cond_ID</b>	<b>int</b>
Vehicle VIN Number	varchar



COLUMN_NAME	DATA_TYPE
DOLocationID	int
fare_amount	double
passenger_count	int
payment_type	int
PULocationID	int
Taxi_Cond_ID	int
tip_amount	double
total_amount	double
tpep_dropoff_datetime	datetime
tpep_pickup_datetime	datetime
trip_distance	double

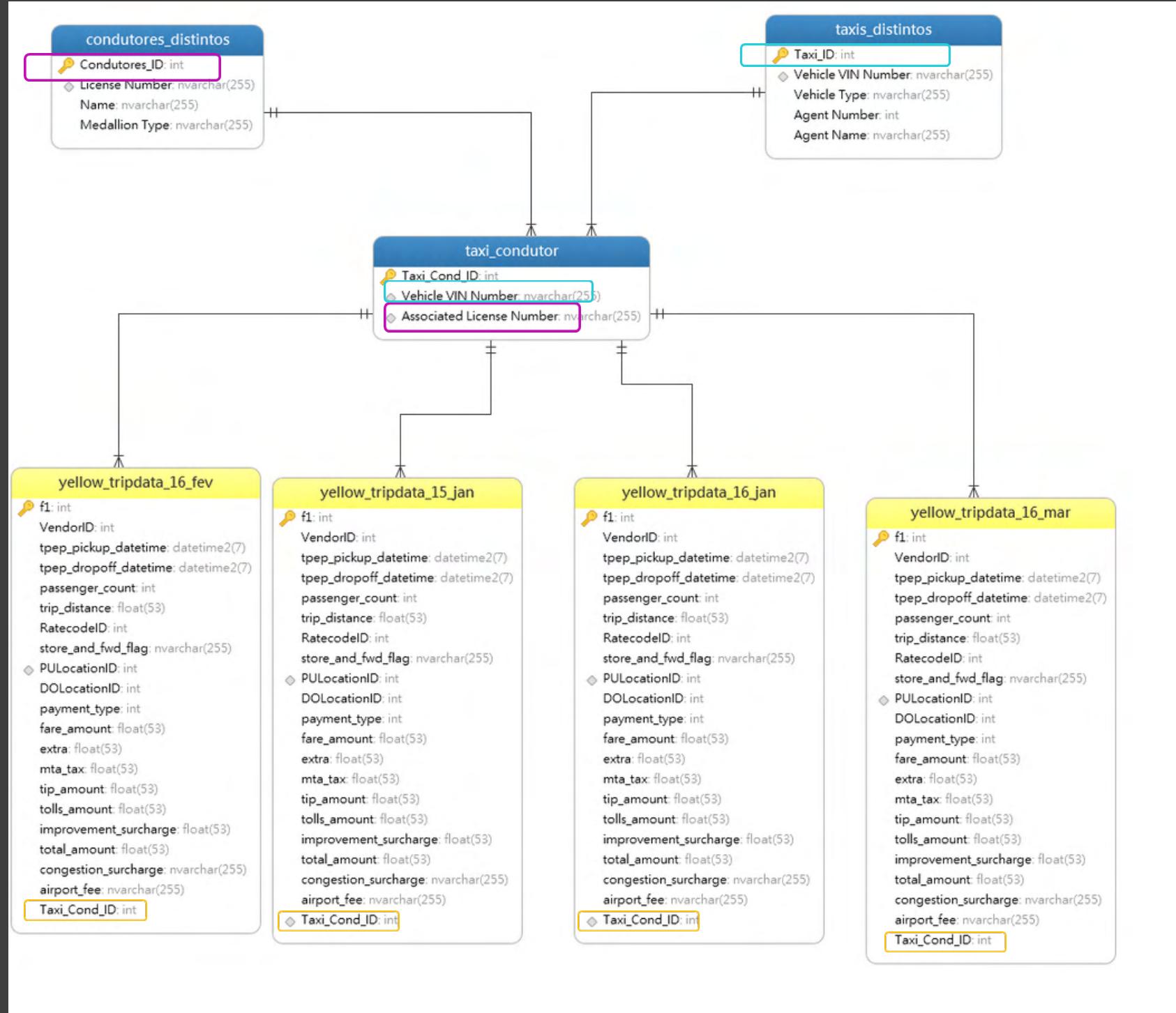
11 rows in set (0.00 sec)

```
ALTER TABLE `yellow_tripdata_16_mar` ADD COLUMN
`TaxiID`UPDATE `yellow_tripdata_16_mar`
INT;
SET Taxi_Cond_ID = FLOOR( RAND() * 24858 ) + 1;
```

# ANÁLISE DO DATASET

6

MODELO RELACIONAL ENTRE TABELAS E LIGAÇÃO AO DATASET ORIGINAL.



# ANÁLISE DO DATASET

6

RESPOSTA A TODAS AS QUESTÕES.

COLUMN_NAME	DATA_TYPE
DOLocationID	int
fare_amount	double
passenger_count	int
payment_type	int
PULocationID	int
Taxi_Cond_ID	int
tip_amount	double
total_amount	double
tpep_dropoff_datetime	datetime
tpep_pickup_datetime	datetime
trip_distance	double

11 rows in set (0.00 sec)



AS PERGUNTAS A QUE  
TINHAMOS RESPOSTA

17 / 17

# VISUALIZAÇÃO DE DADOS |

LIGAÇÃO DE SQL SERVER COM POWER BI.

7



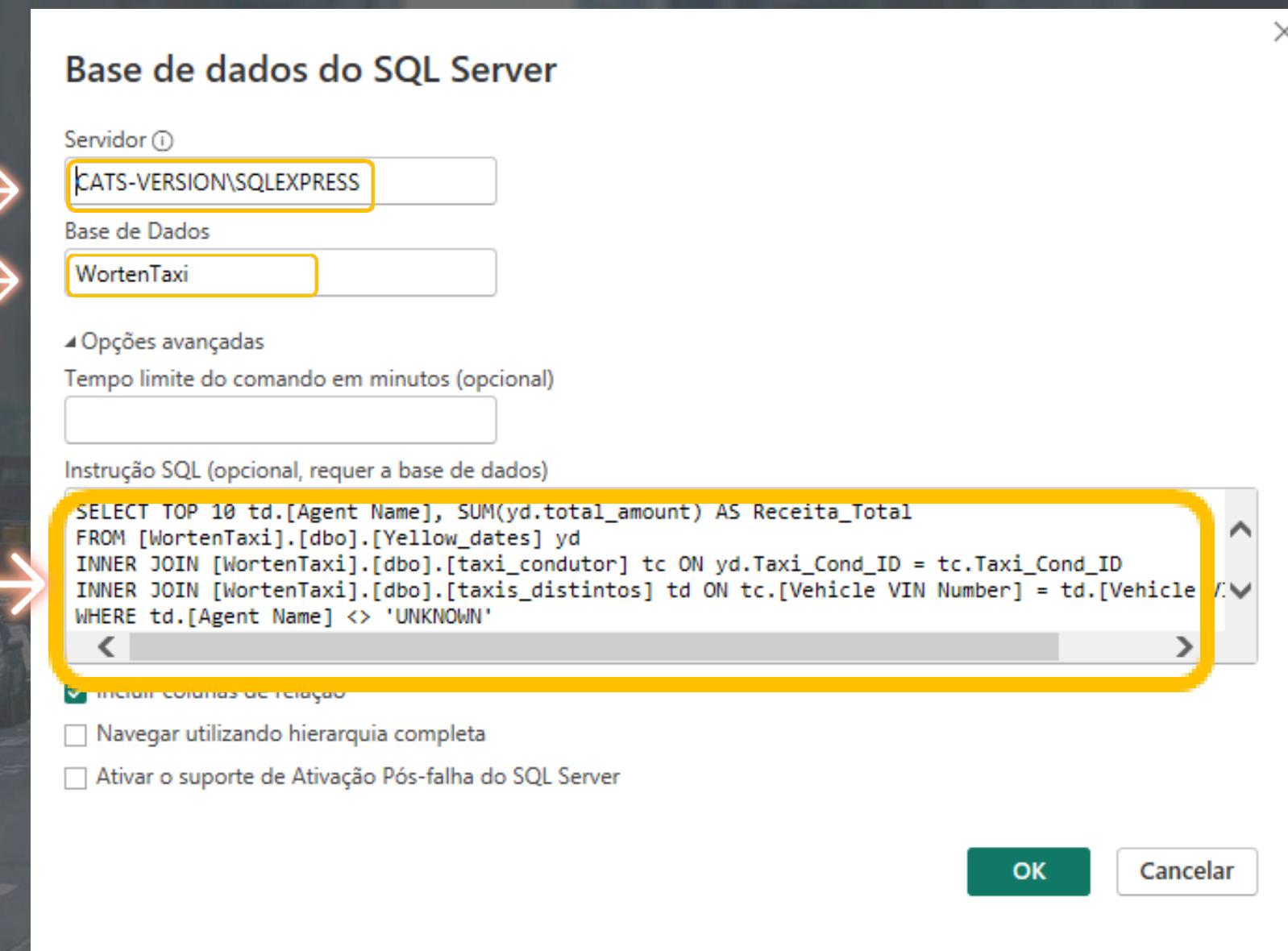
Power BI



# VISUALIZAÇÃO DE DADOS |

8

CRIAÇÃO DE QUERIES SQL DENTRO DE POWER BI.



# VISUALIZAÇÃO DE DADOS | 9

CRIAÇÃO DE VISUALIZAÇÕES PRETENDIDAS EM POWER BI.

- > 10 melhores agentes
- > Abaixo de 10
- > carros\_por\_taxistas
- > Contagem\_Por\_Tipo\_FHV
- > Distancia\_Média\_Green
- > Distancia\_Média\_Yellow
- > Distribuicao\_Montantes
- > DO\_Populares\_Green
- > DO\_Populares\_Yellow
- > Gorjetas\_Distribuicao\_Green
- > Gorjetas\_Distribuicao\_Yellow
- > Hora\_Semana\_Viagens\_FHV
- > Hora\_Semana\_Viagens\_Green
- > Hora\_Semana\_Viagens\_Yellow
- > Hora\_Viagens\_Yellow
- > Horas\_Viagens\_FHV
- > Horas\_Viagens\_Green
- > Lucro\_Por\_Zona\_Green
- > Lucro\_Por\_Zona\_Yellow
- > Média\_Viagens\_FHV
- > Média\_Viagens\_FHV\_Dias\_Semana
- > Média\_Viagens\_Green

Agent Name	Receita_Total
ALL TAXI MANAGEMENT INC	51635316,7699991
TAXIFLEET MANAGEMENT LLC	45030712,3199992
WOODSIDE MANAGEMENT INC.	30643077,4499995
GOTHAM YELLOW LLC	25694676,3299996
QUEENS MEDALLION LEASING INC.	25673571,5099996
TEAM SYSTEMS CORP	21010959,8599996
WHITE AND BLUE GROUP CORP.	20351403,1799996
MC GUINNESS MANAGEMENT CORP	19703004,0599997
S & R MEDALLION CORP	17253426,0499997
ARTHUR CAB LEASING CORP.	16011494,4399997

# VISUALIZAÇÃO DE DADOS

10

CRIAÇÃO DE CÓDIGO PYTHON DENTRO DO POWER BI PARA HEATMAP  
E HISTOGRAMA.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Supondo que 'dataset' seja o DataFrame que contém os dados

yellow = dataset.pivot_table(values='Num_Viagens_Yellow', index='Dia_Semana', columns='Hora')

# Definindo o tamanho da figura
plt.figure(figsize=(30, 10)) # Ajuste os valores de largura e altura conforme necessário

heatmap = sns.heatmap(yellow, cmap='YlOrRd', linecolor='white', linewidths=1)

# Adicionar título e rótulos aos eixos com tamanho de fonte aumentado
plt.ylabel('Número de Viagens', color='white', fontweight='bold', fontsize=40) # Set y-axis label color to white and bold
plt.xlabel('Hora', color='white', fontweight='bold', fontsize=40) # Set x-axis label color to white and bold

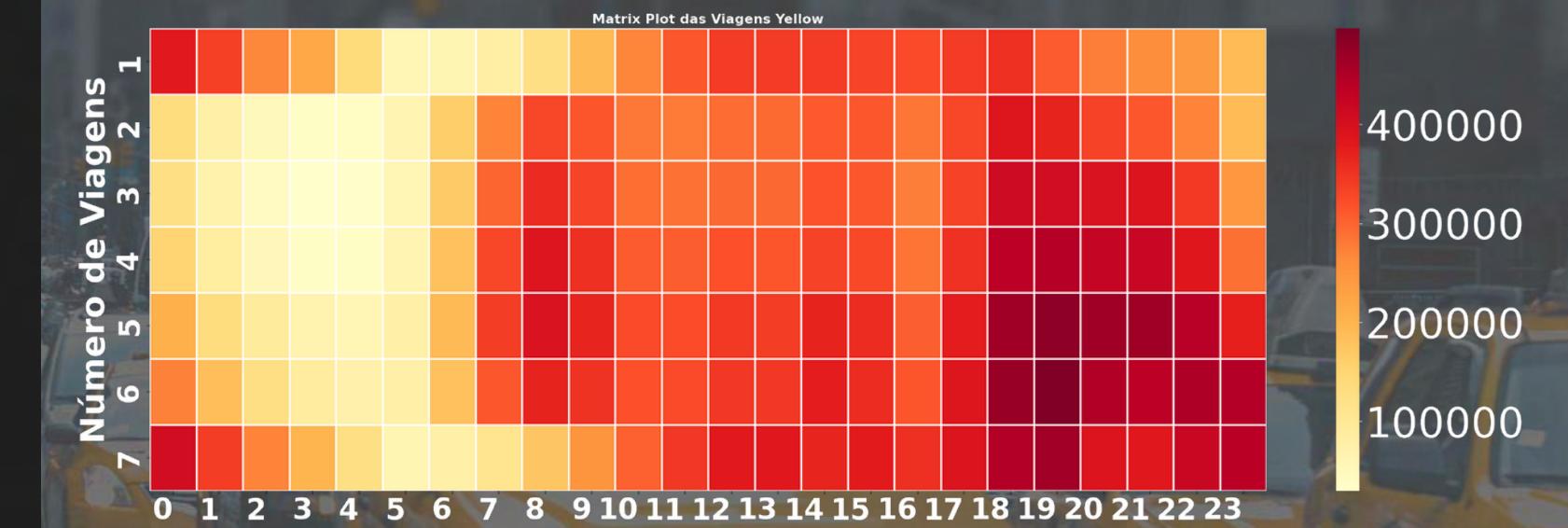
# Definir a cor do texto nos rótulos dos eixos e aumentar o tamanho da fonte
plt.xticks(color='white', fontweight='bold', ha='right', fontsize=35)
plt.yticks(color='white', fontweight='bold', ha='right', fontsize=35)

# Configurar a cor do texto da legenda da escala de cor (colorbar) e aumentar o tamanho da fonte
colorbar = heatmap.collections[0].colorbar
colorbar.set_label('', color='white', fontweight='bold', fontsize=50)
colorbar.ax.yaxis.set_tick_params(color='white', labelcolor='white', labelsize=50)

# Salvar o heatmap com fundo transparente
plt.savefig('plot.png', transparent=True)

plt.title('Matrix Plot das Viagens Yellow', color='white', fontweight='bold', fontsize=16)
plt.show()

# Epilog - Auto Generated #
os.chdir(u'C:/Users/ana_c/PythonEditorWrapper_f58b672b-3725-46d1-aa36-839e174aca99')
```



# VISUALIZAÇÃO DE DADOS |

11

CRIAÇÃO DE CÓDIGO R EM POWER BI PARA PREVISÃO DE PRÓXIMOS 14 DIAS.

```
#-----Regressão Linear
lm_model <- lm(trips ~ pickup_date, data = yellow_data)
prediction_range <- seq(min(yellow_data$pickup_date), max(yellow_data$pickup_date) + 14, by = "day")

forecast_values_lm <- predict(lm_model, newdata = data.frame(pickup_date = prediction_range))
forecast_df_lm <- data.frame(date = prediction_range, forecast_lm = forecast_values_lm)

par(bg = "transparent")
plot(yellow_data$pickup_date, yellow_data$trips, type = "l", col = "yellow",
      xlab = "Date", ylab = "Número de Viagens",
      main = "Volume de Viagens de Janeiro a Março e Previsão para Abril",
      lwd = 2,          # Define a largura da linha
      lty = 1,          # Define o tipo de linha
      xlim = c(min(yellow_data$pickup_date), max(forecast_df$date)), # Define os limites do eixo x
      ylim = c(0, max(yellow_data$trips, na.rm = TRUE) * 1.1),       # Define os limites do eixo y
      col.axis = "white",    # Define a cor dos rótulos dos eixos
      col.lab = "white",    # Define a cor dos rótulos dos eixos
      col.main = "white"
)

polygon(c(yellow_data$pickup_date, rev(yellow_data$pickup_date)),
        c(yellow_data$trips, rep(0, length(yellow_data$trips))),
        col = rgb(1, 0.7, 0, alpha = 0.2), border = NA)

polygon(c(forecast_df$date, rev(forecast_df$date)),
        c(forecast_df$forecast_arima, rep(0, length(forecast_df$forecast_arima))),
        col = rgb(1, 0.9, 0, alpha = 0.2), border = NA)

lines(forecast_df$date, forecast_df$forecast_arima, col = "orange") # Laranja
lines(forecast_df_lm$date, forecast_df_lm$forecast_lm, col = "red") # Vermelho

xlabel <- c(as.character(yellow_data$pickup_date[nrow(yellow_data)]), as.character(forecast_df$date))
axis(1, at = c(1:length(xlabel)), labels = xlabel, col.axis = "white")

legend("bottomright", legend = c("Observado", "Previsão (ARIMA)", "Previsão (Regressão Linear)"),
      col = c("yellow", "orange", "red"), lty = c(1, 1, 1), text.col = "white")
```



```
16  #-- Modelo Arima
17  yellow_data <- dataset
18  yellow_data$pickup_date <- as.Date(yellow_data$pickup_date)
19
20
21  arima_model <- auto.arima(yellow_data$trips)
22
23  forecast_next_14_days <- forecast(arima_model, h = 14)
24
25  forecast_means <- rep(mean(forecast_next_14_days$`mean`), 14)
26
27  next_14_days_dates <- seq(max(yellow_data$pickup_date) + 1, length.out = 14, by = "day")
28
29  forecast_df <- data.frame(date = next_14_days_dates, forecast_arima = forecast_means)
```



CESAE DIGITAL

# NEW YORK TAXIS

CLIENTE WORTEN

