

# Case study EDA & imputation

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER



**Meghan Kwartler**  
IT Consultant

# Taxi ride business problem

1. EU private equity firm seeking investment opportunity in US Transportation.
2. What is the **average fare per distance, ride count & total ride time** for each NYC borough on each day of the week?
3. Which **pickup locations** within the borough should be scheduled for each of the driver shifts?



# Essential EDA

- Distributed transactional datasets can contain impossible scenarios due to data collection calibration problems
  - Dates in future
  - End dates before start dates

```
SELECT *  
FROM CapitalBikeShare  
WHERE  
    StartDate > GetDate()  
    OR EndDate > GetDate()  
    OR StartDate > EndDate
```

# Data imputation

- Divide by zero error when calculating Avg Fare/TripDistance
- EDA uncovers hundreds of TaxiRide trip records with Trip Distance = 0
- Data Imputation methods to resolve
  - Mean
  - Hot Deck
  - Omission

# Mean imputation

- Replace missing value with mean
- Doesn't change the mean value
- Increases correlations with other columns

```
CREATE PROCEDURE dbo.ImputeDurMean
AS
BEGIN
DECLARE @AvgTripDuration AS float

SELECT @AvgTripDuration = AVG(Duration)
FROM CapitalBikeShare
WHERE Duration > 0

UPDATE CapitalBikeShare
SET Duration = @AvgTripDuration
WHERE Duration = 0
END;
```

# Hot Deck imputation

- Missing value set to randomly selected value
- TABLESAMPLE clause of FROM clause

```
CREATE FUNCTION dbo.GetDurHotDeck()
RETURNS decimal (18,4)
AS BEGIN
RETURN (SELECT TOP 1 Duration
FROM CapitalBikeShare
TABLESAMPLE (1000 rows)
WHERE Duration > 0)
END
SELECT
    StartDate,
    "TripDuration" = CASE WHEN Duration > 0 THEN Duration
                        ELSE dbo.GetDurHotDeck() END
FROM CapitalBikeShare;
```

```
SELECT
    DATENAME(weekday, StartDate) AS DayofWeek,
    AVG(Duration) AS 'AvgDuration'
FROM CapitalBikeShare
WHERE Duration > 0
GROUP BY DATENAME(weekday, StartDate)
ORDER BY AVG(Duration) desc
```

```
+-----+-----+
| DayofWeek | AvgDuration |
+-----+-----+
| Saturday  | 1476        |
| Sunday    | 1391        |
| Monday    | 979         |
| Friday    | 957         |
| Thursday  | 956         |
| Wednesday | 807         |
| Tuesday   | 763         |
+-----+-----+
```

# Your turn!

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER



# Case study UDFs

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER



**Meghan Kwartler**  
IT Consultant

# Taxi ride business problem

1. EU private equity firm seeking investment opportunity in US Transportation.
2. What is the **average fare per distance**, **ride count** & **total ride time** for each NYC borough on each day of the week?



# Taxi ride business problem

1. EU private equity firm seeking investment opportunity in US Transportation.
2. What is the **average fare per distance, ride count & total ride time** for each NYC borough on each day of the week?
3. Which **pickup locations** within the borough should be scheduled for each of the driver shifts?



# Conversion UDFs

```
CREATE FUNCTION dbo.ConvertMileToMeter (@miles numeric)
RETURNS numeric
AS
BEGIN
RETURN (SELECT @miles * 1609.34)
END
```

```
CREATE FUNCTION dbo.ConvertCurrency (@Currency numeric, @ExchangeRate numeric)
RETURNS numeric
AS
BEGIN
RETURN (SELECT @ExchangeRate * @Currency)
END
```

```
SELECT TripDistance as 'MileDistance',
dbo.ConvertMileToMeter (TripDistance) as 'MeterDistance',
FareAmount as 'FareUSD',
dbo.ConvertCurrency (FareAmount, '.78') as 'FareGBP'
FROM dbo.YellowTripData
```

MileDistance	MeterDistance	FareUSD	FareGBP
1.10	1609	6	6
0.02	0	52	52
0.50	1609	4	4
7.75	12875	22	22
0.80	1609	6	6
0.90	1609	7	7
1.76	3219	7	7
8.47	12875	24	24
2.40	3219	10.50	11
12.60	20921	60	60
0.90	1609	7	7

# Iterate

```
ALTER FUNCTION dbo.ConvertMileToMeter (  
    @miles numeric (18, 2)  
) RETURNS numeric (18, 2) as BEGIN RETURN (  
    SELECT @miles * 1609.34  
) END;
```

```
ALTER FUNCTION dbo.ConvertCurrency (  
    @Currency numeric (18, 2),  
    @ExchangeRate numeric(18, 2)  
) RETURNS numeric (18, 2) AS BEGIN RETURN (  
    SELECT @ExchangeRate * @Currency  
) END;
```

# What about Shifts?

```
CREATE FUNCTION dbo.GetShift (@Hour int)
RETURNS int
AS
BEGIN
RETURN (CASE
    WHEN @Hour >= 0 AND @Hour < 9 THEN 1
    WHEN @Hour >= 9 AND @Hour < 18 THEN 2
    WHEN @Hour >= 18 AND @Hour < 24 THEN 3
END)
END;
```

# Test Shifts

```
SELECT
  DATENAME(hour, PickupDate) AS 'Hour',
  dbo.GetShift (
    DATENAME(hour, PickupDate)
  ) AS 'Shift'
FROM YellowTripData
GROUP BY DATENAME(hour, PickupDate)
ORDER BY
  dbo.GetShift (
    DATENAME(hour, PickupDate)
  )
```

Hour	Shift
3	1
6	1
4	1
2	1
0	1
8	1
5	1
1	1
7	1
9	2
11	2
15	2
14	2
...	...



# Your turn!

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER

# Formatting tools

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER



**Meghan Kwartler**  
IT Consultant

# Before formatting

```
SELECT
    DATENAME(weekday, StartDate) AS 'DayOfWeek',
    SUM(Duration) AS TotalDuration
FROM CapitalBikeShare
GROUP BY DATENAME(weekday, StartDate)
ORDER BY DATENAME(weekday, StartDate)
```

```
+-----+-----+
| DayOfWeek | TotalDuration |
+-----+-----+
| Friday    | 7264870       |
| Monday    | 6571322       |
| Saturday  | 13411642      |
| Sunday    | 8418226       |
| Thursday  | 8646359       |
| Tuesday   | 3788474       |
| Wednesday | 3525955       |
+-----+-----+
```

# Sort by logical weekday

```
SELECT DATENAME(weekday, StartDate) as 'DayOfWeek',  
SUM(Duration) as TotalDuration  
FROM CapitalBikeShare  
GROUP BY DATENAME(WEEKDAY, StartDate)  
ORDER BY  
    CASE WHEN Datename(WEEKDAY, StartDate) = 'Sunday' THEN 1  
         WHEN Datename(WEEKDAY, StartDate) = 'Monday' THEN 2  
         WHEN Datename(WEEKDAY, StartDate) = 'Tuesday' THEN 3  
         WHEN Datename(WEEKDAY, StartDate) = 'Wednesday' THEN 4  
         WHEN Datename(WEEKDAY, StartDate) = 'Thursday' THEN 5  
         WHEN Datename(WEEKDAY, StartDate) = 'Friday' THEN 6  
         WHEN Datename(WEEKDAY, StartDate) = 'Saturday' THEN 7  
END ASC;
```

```
+-----+-----+  
| DayOfWeek | TotalDuration |  
+-----+-----+  
| Sunday    | 8418226       |  
| Monday    | 6571322       |  
| Tuesday   | 3788474       |  
| Wednesday | 3525955       |  
| Thursday  | 8646359       |  
| Friday    | 7264870       |  
| Saturday  | 13411642      |  
+-----+-----+
```

```

SELECT TOP 5
FORMAT(CAST(StartDate as Date), 'd', 'de-de')
AS 'German Date',
FORMAT(CAST(StartDate as Date), 'd', 'en-us')
AS 'US Eng Date',
FORMAT(Sum(Duration), 'n', 'de-de')
AS 'German Duration',
FORMAT(SUM(Duration), 'n', 'en-us')
AS 'US Eng Duration',
FORMAT(SUM(Duration), '#,0.00')
AS 'Custom Numeric'
FROM CapitalBikeShare
GROUP BY CAST(StartDate as Date)

```

```

+-----+-----+-----+-----+-----+
| German Date | US Eng Date | German Duration | US Eng Duration | Custom Numeric |
+-----+-----+-----+-----+-----+
| 09.03.2018 | 3/9/2018    | 1.141.796,00    | 1,141,796.00    | 1,141,796.00    |
| 18.03.2018 | 3/18/2018   | 3.074.907,00    | 3,074,907.00    | 3,074,907.00    |
| 12.03.2018 | 3/12/2018   | 1.088.822,00    | 1,088,822.00    | 1,088,822.00    |
| 26.03.2018 | 3/26/2018   | 2.160.609,00    | 2,160,609.00    | 2,160,609.00    |
| 29.03.2018 | 3/29/2018   | 3.552.955,00    | 3,552,955.00    | 3,552,955.00    |
+-----+-----+-----+-----+-----+

```

```

SELECT DATENAME(weekday, StartDate)
AS 'DayOfWeek',
FORMAT(SUM(Duration), '#,0.00')
AS 'TotalDuration'
FROM CapitalBikeShare
GROUP BY DATENAME(WEEKDAY, StartDate)
ORDER BY
    CASE
        WHEN Datename(WEEKDAY, StartDate) = 'Sunday' THEN 1
        WHEN Datename(WEEKDAY, StartDate) = 'Monday' THEN 2
        WHEN Datename(WEEKDAY, StartDate) = 'Tuesday' THEN 3
        WHEN Datename(WEEKDAY, StartDate) = 'Wednesday' THEN 4
        WHEN Datename(WEEKDAY, StartDate) = 'Thursday' THEN 5
        WHEN Datename(WEEKDAY, StartDate) = 'Friday' THEN 6
        WHEN Datename(WEEKDAY, StartDate) = 'Saturday' THEN 7
    END ASC

```

```

+-----+-----+
| DayOfWeek | TotalDuration |
+-----+-----+
| Sunday    | 8,418,226.00  |
| Monday    | 6,571,322.00  |
| Tuesday    | 3,788,474.00  |
| Wednesday | 3,525,955.00  |
| Thursday   | 8,646,359.00  |
| Friday     | 7,264,870.00  |
| Saturday   | 13,411,642.00 |
+-----+-----+

```

# Your turn!

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER

# Case study stored procedures

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER



**Meghan Kwartler**  
IT Consultant



# Taxi ride business problem

1. EU private equity firm seeking investment opportunity in US Transportation.
2. What is the **average fare per distance, ride count & total ride time** for each NYC borough on each day of the week?
3. Which **pickup locations** within the borough should be scheduled for each of the driver shifts?



# Evolution of an SP

--Query detail level with UDFs

**SELECT**

DATENAME (weekday, PickupDate) **as** 'Weekday',

PickupDate,

DropOffDate,

TotalAmount,

TripDistance,

dbo.ConvertDollar(TotalAmount, .88)/ dbo.ConvertMileToKm(TripDistance) **as** 'EuroFarePerKM',

DATEDIFF(**SECOND**, PickupDate, DropOffDate)/ 60 **as** 'TotalRideMin'

**FROM** YellowTripData

**WHERE** TripDistance > 0

```

SELECT DATENAME(weekday, PickupDate) as 'Weekday',
       Zone.Borough as 'PickupBorough',
       AVG(dbo.ConvertDollar(TotalAmount, .77)/
       dbo.ConvertMiletoKM(TripDistance)))
       AS 'AvgFarePerKM',
       COUNT (ID) as 'RideCount',
       SUM(DATEDIFF(SECOND, PickupDate, DropOffDate)/60) as 'TotalRideMin'
FROM YellowTripData
INNER JOIN TaxiZoneLookup AS Zone
ON PULocationID = Zone.LocationID
WHERE dbo.ConvertMiletoKM(TripDistance) > 0
GROUP BY DATENAME(WEEKDAY, PickupDate), Zone.Borough
ORDER BY CASE
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Monday' THEN 1
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Tuesday' THEN 2
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Wednesday' THEN 3
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Thursday' THEN 4
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Friday' THEN 5
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Saturday' THEN 6
    WHEN DATENAME(WEEKDAY, PickupDate) = 'Sunday' THEN 7
END,
AVG(dbo.ConvertDollar(TotalAmount, .77)/dbo.ConvertMiletoKM(TripDistance)) DESC;

```

# "Last" step

```
CREATE OR ALTER PROCEDURE dbo.cuspPickupZoneShiftStats
@Borough nvarchar(30)
AS
BEGIN
.....
END
```

```
DROP PROCEDURE IF EXISTS dbo.cuspPickupZoneShiftStats
GO
CREATE PROCEDURE dbo.cuspPickupZoneShiftStats
@Borough nvarchar(30)
AS
BEGIN
....
END
```

# Your turn!

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER

# Congratulations!

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER



**Meghan Kwartler**  
IT Consultant

# Continued practice

- Download
  - CapitalBikeShare
  - YellowTripTaxi

# Accomplishments

- EDA for distributed transactional data
- CONVERT(), CAST(), DATEDIFF(), DATENAME(), DATEADD(), DATEPART(), GETDATE()
- DECLARE & SET scalar variables
- DECLARE & INSERT INTO table variables
- CREATE, EXEC, UPDATE, DROP user defined functions (scalar, ITVF, MSTVF)
- CREATE, EXEC, UPDATE, DROP stored procedures
- Solved real world business problems



# Next Steps

- Error Handling in stored procedures with TRY CATCH THROW
- FORMAT()
- Data imputation
- SQL for Exploratory Data Analysis
- Intermediate SQL Server
- Intermediate SQL

# Good Luck!

WRITING FUNCTIONS AND STORED PROCEDURES IN SQL SERVER