## **True Car Listings 2017**

**วัตถุประสงค์** ของ Project นี้ เพื่อศึกษาและฝึกฝนโดยการนำ Data ในชีวิตจริงจาก Website ขายรถยนต์ Online ของประเทศสหัฐอเมริกามาออกแบบวิเคราะห์ ตั้งคำถาม และ กลั่นกรองข้อมูล เริ่มต้นด้วยการสร้าง Database ใน SQL วิเคราะห์ข้อมูลเบื้องต้นโดยใช้ Python และท้ายที่สุดจะนำ Project นี้ไปสร้าง dashbroad ผ่านทาง Tableau Public เพื่อช่วยในการ Visualize ข้อมูล โดยข้อมูลทั้งหมดเป็นข้อมูลที่มาจากการสั่งซื้อรถยนต์ Online ของคนใน USA ผ่านทางเว็ปไซต์ TrueCar ตั้งแต่อดีตมาจนถึงปี 2017 (แหล่งที่มาข้อมูล [www.TrueCar.com](http://www.truecar.com/))

## **Database & Table Creation**

**CREATE DATABASE** TrueCar**;**

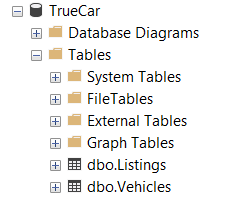
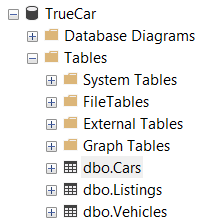
****

Table ที่ import เข้ามา ประกอบไปด้วย 2 Tables ดังนี้

**Table 1 :** Vehicles ประกอบไปด้วย ( **Id**, Make, Model, Year, VIN )

**SELECT TOP 5 \***

**FROM TrueCar..Vehicles;**

**Result :**



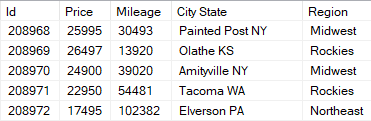
1. Id คือ ตัวระบุอัตโนมัติที่ถูกสร้างขึ้นมาเพื่อใช้เป็น Primary Key
2. Year คือ ปีที่ผลิตรถยนต์
3. VIN คือ รหัสระบุเฉพาะยานยนต์
4. Make คือ ชื่อยี่ห้อของรถยนต์
5. Model คือ ชื่อรุ่นโมเดลของรถยนต์

**Table 2 :** Listings ประกอบไปด้วย ( **Id**, Price, Mileage, City State, Region )

**SELECT TOP 5 \***

**FROM TrueCar..Listings;**

**Result :**



1. Price คือ ราคาที่จำหน่าย (หน่วย:ดอลลาร์)
2. Mileage คือ ระยะทางที่รถยนต์วิ่งมาแล้ว (หน่วย:ไมล์)
3. City State คือ เมืองที่จำหน่ายรถยนต์
4. Region คือ ภูมิภาคที่จำหน่ายรถยนต์

Fact : ทั้ง 2 Tables มี row จำนวนทั้งหมดประมาณ 1.05M และ 9 unique columns

## **Data Cleaning with SQL Server**

ตรวจสอบ Null values ทั้ง 2 Tables โดยใช้ Inner Join เข้ามาช่วย

**SELECT \***

**FROM TrueCar..Listings l**

**INNER JOIN TrueCar..Vehicles v**

**ON l.Id = v.Id**

**WHERE coalesce(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) IS NULL;**

**Result :** ไม่มี Null Value

แยก Column **‘City State’** ออกเป็น Column **‘City’** และ **‘State’**

**ALTER TABLE TrueCar..Listings**

**ADD City VARCHAR(50),**

**State VARCHAR(5);**

**UPDATE TrueCar..Listings**

**SET City = SUBSTRING(TRIM([City State]), 1, LEN(TRIM([City State])) - 2),**

**State = SUBSTRING(TRIM([City State]), LEN(TRIM([City State])) - 2, LEN(TRIM([City State])));**

**SELECT TOP 5 [City State], City, State**

**FROM TrueCar..Listings;**

**Result :**



ตรวจสอบว่า Data มี row ซ้ำกันหรือไม่

**WITH RowNumber AS**

**(**

**SELECT l.id, Vin, Make, Model, Year, Mileage, City,**

**State, Region, Price,**

**ROW\_NUMBER() OVER(**

**PARTITION BY Vin, Make, Model, Year,**

**Mileage, City, State, Region, Price**

**ORDER BY l.Id) as NumberOfRow**

**FROM TrueCar..Listings l**

**INNER JOIN TrueCar..Vehicles v**

**ON l.Id = v.Id**

**)**

**SELECT \***

**FROM RowNumber**

**WHERE NumberOfRow > 1;**

**Result :** ไม่มี row ไหนที่มีข้อมูลซ้ำกัน

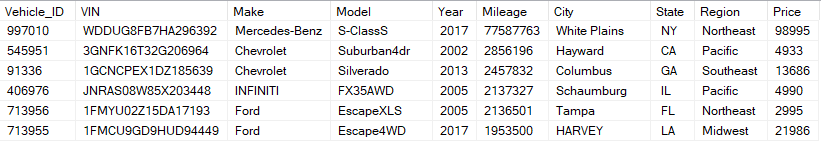
จัดการกับข้อมูลที่เป็น Outlier

หากเรียงลำดับข้อมูลโดยอิงจากค่า Mileage ในรูปแบบ Descending order จะมีอยู่ 1 row ที่มีค่า Mileage(ระยะทางการวิ่งของรถยนต์) สูงถึง 77.5 ล้านไมล์ หรือ ประมาณ 126 ล้านกิโลเมตร ซึ่งคิดว่าน่าจะเป็นข้อผิดพลาดจากการเก็บข้อมูล เนื่องจากข้อมูลนั้นมีระยะทางการวิ่งที่สูงมาก แต่รถยนต์เพิ่งถูกผลิตมาในปี 2017 จึงตัดสินใจเลือกที่จะตัดข้อมูล row นี้ออกไป

**SELECT TOP 5 \***

**FROM TrueCar..Cars**

**ORDER BY Mileage DESC**

****

**DELETE FROM TrueCar..Cars WHERE Mileage > 70000000**

## **Create New Table**

สร้าง Table ใหม่โดยการนำข้อมูลของ Table Vehicles และ Listings มารวมกัน เพื่อให้ง่ายต่อการ Execute query ในครั้งถัดไป รวมทั้งลบ data รถยนต์ที่ถูกผลิตในปี 2018 เนื่องจากไม่ทราบว่าข้อมูลที่เก็บมา เก็บเป็นระยะเวลานานแค่ไหนแล้วจึงคิดว่าจะส่งผลเสียต่อการ Visualize

**CREATE TABLE TrueCar..Cars (**

**Vehicle\_ID INT PRIMARY KEY NOT NULL,**

**VIN VARCHAR(20) UNIQUE NOT NULL,**

**Make VARCHAR(50) NOT NULL,**

**Model VARCHAR(100) NOT NULL,**

**Year SmallINT NOT NULL,**

**Mileage INT NOT NULL,**

**City VARCHAR(50) NOT NULL,**

**State VARCHAR(5) NOT NULL,**

**Region VARCHAR(50) NOT NULL,**

**Price INT NOT NULL**

**);**

**INSERT INTO TrueCar..Cars**

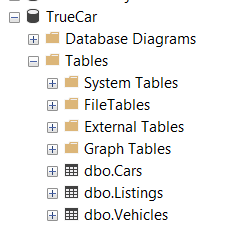
**SELECT l.id, Vin, Make, Model, Year, Mileage, City, State, Region, Price**

**FROM TrueCar..Listings l**

**INNER JOIN TrueCar..Vehicles v**

**ON l.Id = v.Id;**

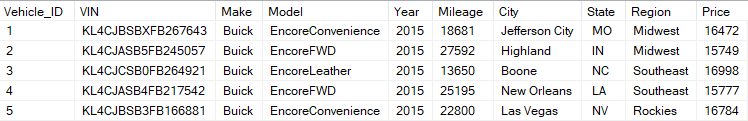
**DELETE FROM TrueCar..Cars WHERE Year = 2018**

****

**SELECT TOP 5 \***

**FROM TrueCar..Cars**

**Result :**

****

## หลังจาก clean data และลบ row ที่ไม่จำเป็นออก จะเหลือ data อยู่ทั้งหมด 1,047,154 rows และ 10 unique columns

## **Basic Visualization**

1. จำนวนรถยนต์ที่ถูกจำหน่ายในแต่ละภูมิภาคและราคาเฉลี่ย

**CREATE VIEW Per as**

**SELECT Region, COUNT(CONVERT(NUMERIC, Vehicle\_ID)) as TotalCarsPerRegion,**

**AVG(CONVERT(BIGINT, price)) as AveragePrice,**

**SUM(CONVERT(NUMERIC, count(Vehicle\_ID))) OVER () as NumberOfCars**

**FROM TrueCar..Cars**

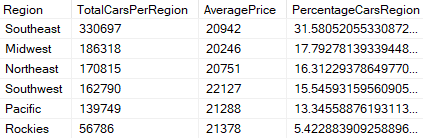
**GROUP BY Region**

**SELECT Region, TotalCarsPerRegion, AveragePrice, (TotalCarsPerRegion/NumberOfCars)\*100 as PercentageCarsRegion**

**FROM Per**

**ORDER BY PercentageCarsRegion DESC;**

**Result :**



* Southeast มีจำนวนรถยนต์ที่ถูกจำหน่ายมากที่สุด( 31.58% ของพื้นที่ใน USA ทั้งหมด) อาจเป็นเพราะมีราคาขายเฉลี่ยต่ำเป็นอันดับ 3 หรืออาจเกิดจากปัจจัยอื่นๆ เช่น จำนวนประชากร ความจำเป็นในการใช้รถยนต์ เป็นต้น

1. Luxury Cars (รถยนต์ที่จำหน่ายแพงกว่า $100,000) ที่ถูกจำหน่ายในแต่ละภูมิภาค

**CREATE VIEW LuxuryPer as**

**SELECT Region, COUNT(CONVERT(NUMERIC, Vehicle\_ID)) as TotalCarsPerRegion,**

**AVG(CONVERT(BIGINT, price)) as AveragePrice,**

**SUM(CONVERT(NUMERIC, count(Vehicle\_ID))) OVER () as NumberOfCars**

**FROM TrueCar..Cars**

**WHERE Price > 100000**

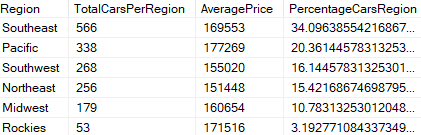
**GROUP BY Region**

**SELECT Region, TotalCarsPerRegion, AveragePrice, (TotalCarsPerRegion/NumberOfCars)\*100 as PercentageCarsRegion**

**FROM LuxuryPer**

**ORDER BY PercentageCarsRegion DESC;**

**Result :**

****

* เปอร์เซนต์รถยนต์ Luxury ที่ถูกจำหน่ายในแต่ละภูมิภาคค่อนข้างใกล้เคียงกับเปอร์เซนต์การจำหน่ายรถยนต์ทั้งหมด

1. รถยี่ห้ออะไรได้รับความนิยมมากที่สุด

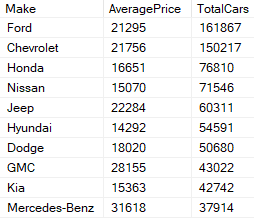
**SELECT Make, AVG(CONVERT(BIGINT, price)) as AveragePrice, COUNT(Vehicle\_ID) as TotalCars**

**FROM TrueCar..Cars**

**GROUP BY Make**

**ORDER BY TotalCars DESC;**

**Result :**



* Ford ได้รับความนิยมมากที่สุด รองมาด้วย Chevrolet เป็นอับดับ 2 และ Honda เป็นอันดับ 3

1. ค่า Median ของราคารถยนต์ในแต่ละยี่ห้อแตกต่างจากค่าเฉลี่ยมากหรือไม่

**SELECT a.Make, MAX(B.Median) as MedianPrice, COUNT(a.Vehicle\_ID) as TotalCar**

**FROM TrueCar..Cars as a**

**INNER JOIN (SELECT DISTINCT Make, PERCENTILE\_DISC(0.5) WITHIN GROUP (ORDER BY Price)**

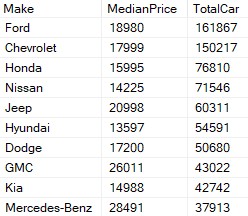
**OVER (PARTITION BY Make) as Median FROM TrueCar..Cars) as b**

**ON a.Make = b.Make**

**GROUP BY a.Make**

**ORDER BY TotalCar DESC;**

**Result:**

****

* ค่า Median ของราคาค่อนข้างต่างจากค่าเฉลี่ยของราคา อาจเป็นเพราะข้อมูลยังคงมี outlier จำนวนมาก ซึ่งจะเก็บไปตรวจสอบและวิเคราะห์ใน Python ต่อไป