Real Time Traffic Project

Under Super Vision: DR. Amira Yousef

Ahmed Salama Soliman Hasan

Mohamed Nasr

Osama Ahmed Reda Hegazy

Sara Hisham Ahmed Mohamed

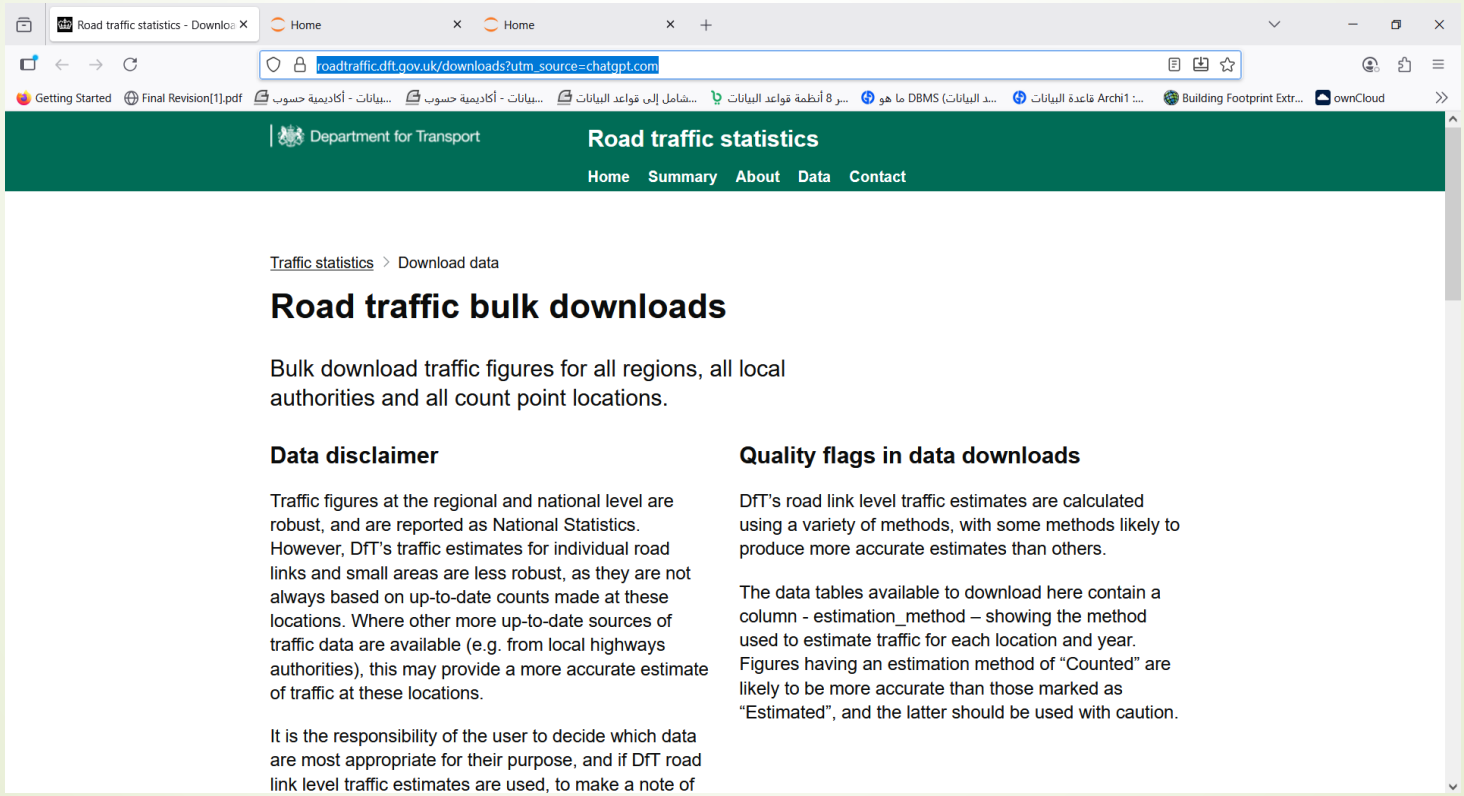
Sherif Gamal Kamed AbdElwahed

Yousef Ahmed Mohamed Ibrahiem

Zakaria Yehia Ahmed

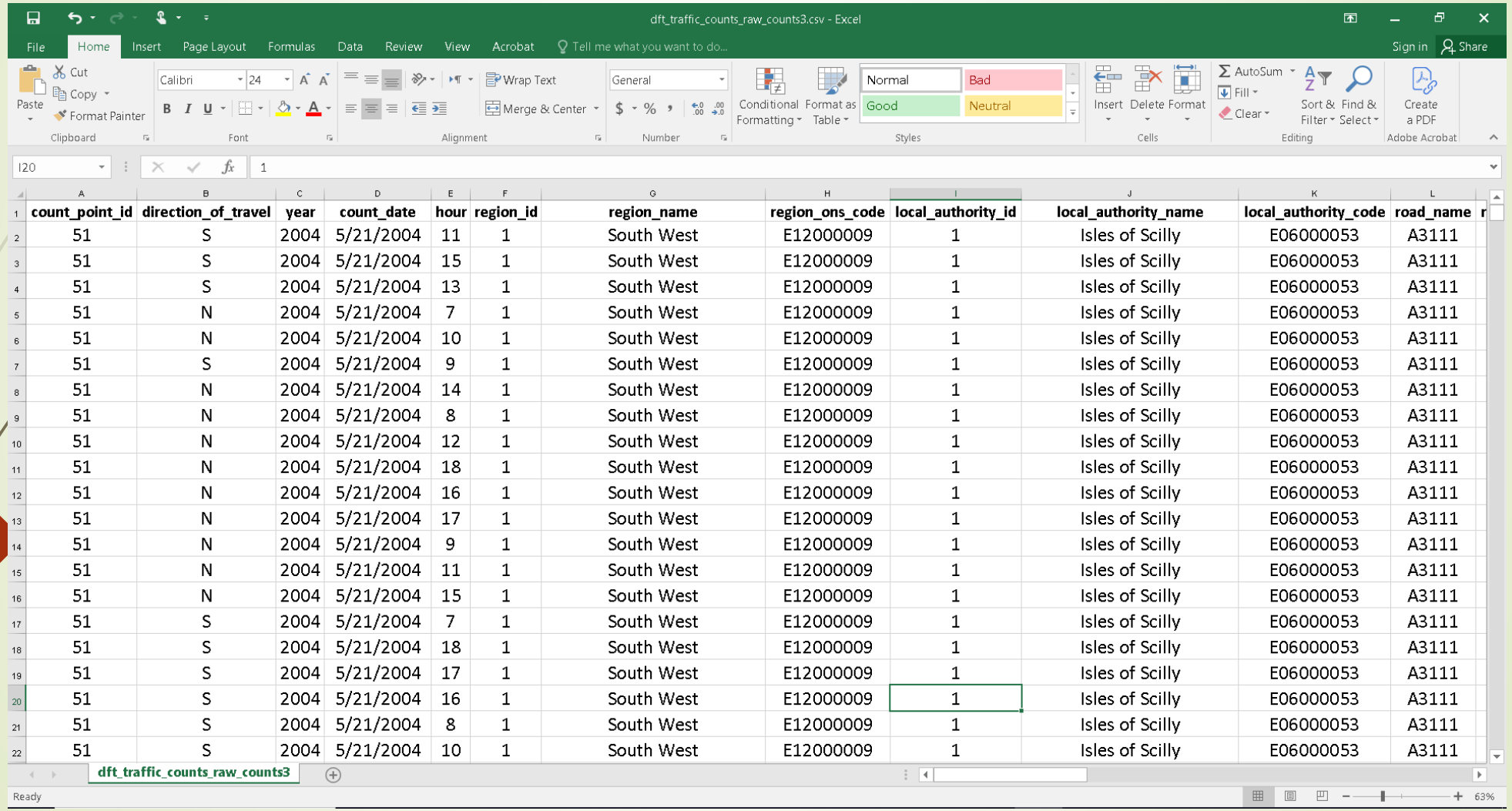
Data Source

https://roadtraffic.dft.gov.uk/downloads?utm_source=chatgpt.com



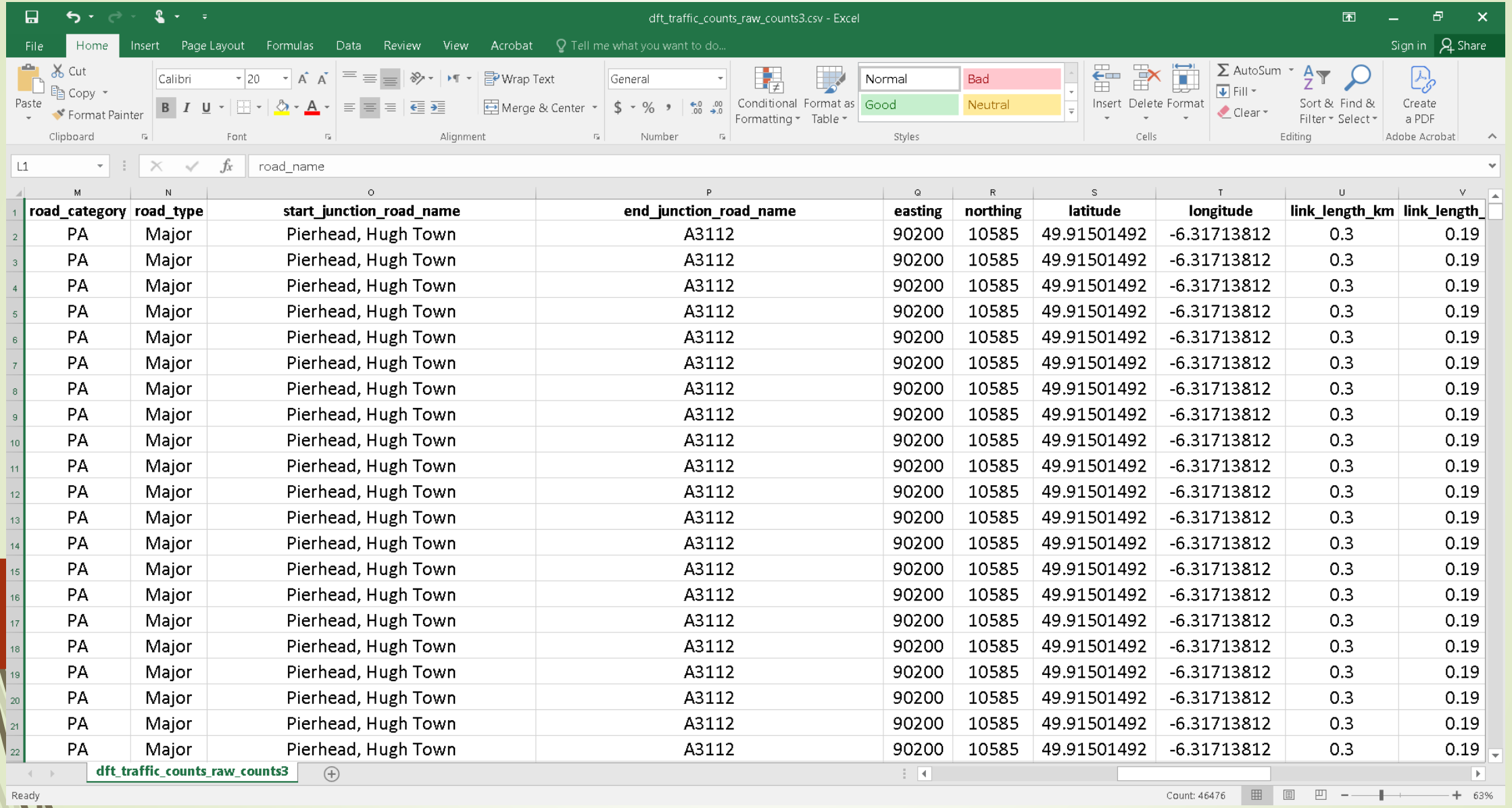
Data Before

(worked on 46476 out of 10,485,76records)



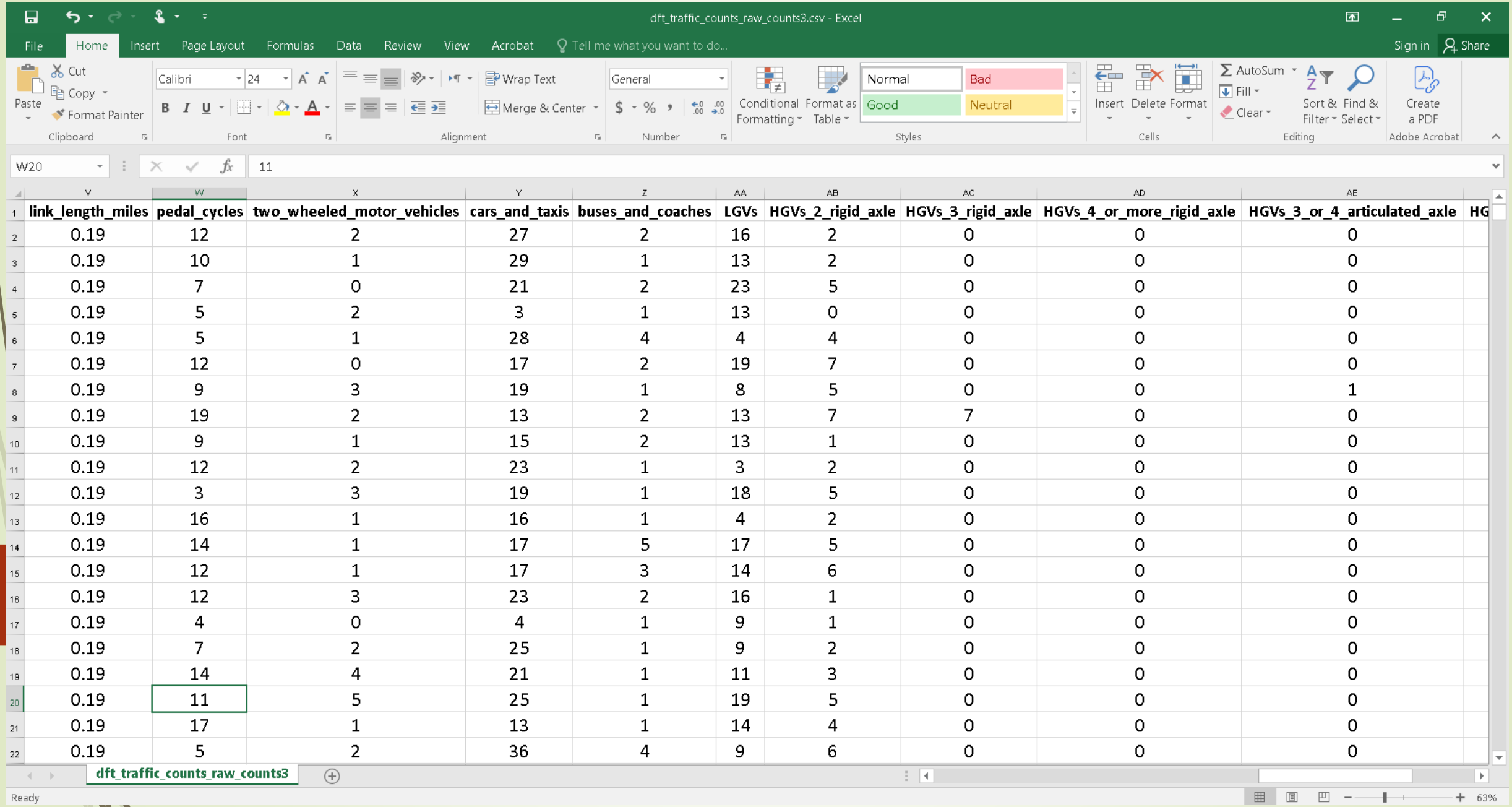
dft_traffic_counts_raw_counts3.csv - Excel

count_point_id	direction_of_travel	year	count_date	hour	region_id	region_name	region_ons_code	local_authority_id	local_authority_name	local_authority_code	road_name
51	S	2004	5/21/2004	11	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	15	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	13	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	7	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	10	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	9	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	14	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	8	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	12	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	18	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	16	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	17	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	9	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	11	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	N	2004	5/21/2004	15	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	7	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	18	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	17	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	16	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	8	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111
51	S	2004	5/21/2004	10	1	South West	E12000009	1	Isles of Scilly	E06000053	A3111



L1 road_name

	M	N	O	P	Q	R	S	T	U	V
1	road_category	road_type	start_junction_road_name	end_junction_road_name	easting	northing	latitude	longitude	link_length_km	link_length_
2	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
3	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
4	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
5	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
6	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
7	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
8	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
9	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
10	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
11	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
12	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
13	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
14	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
15	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
16	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
17	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
18	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
19	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
20	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
21	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19
22	PA	Major	Pierhead, Hugh Town	A3112	90200	10585	49.91501492	-6.31713812	0.3	0.19


dft_traffic_counts_raw_counts3.csv - Excel

File Home Insert Page Layout Formulas Data Review View Acrobat Tell me what you want to do... Sign in Share

Paste Cut Copy Format Painter | Calibri 24 | Wrap Text | General | Conditional Formatting | Format as Table | Normal Bad Good Neutral | Insert Delete Format | AutoSum Fill Clear | Sort & Filter | Find & Select | Create a PDF

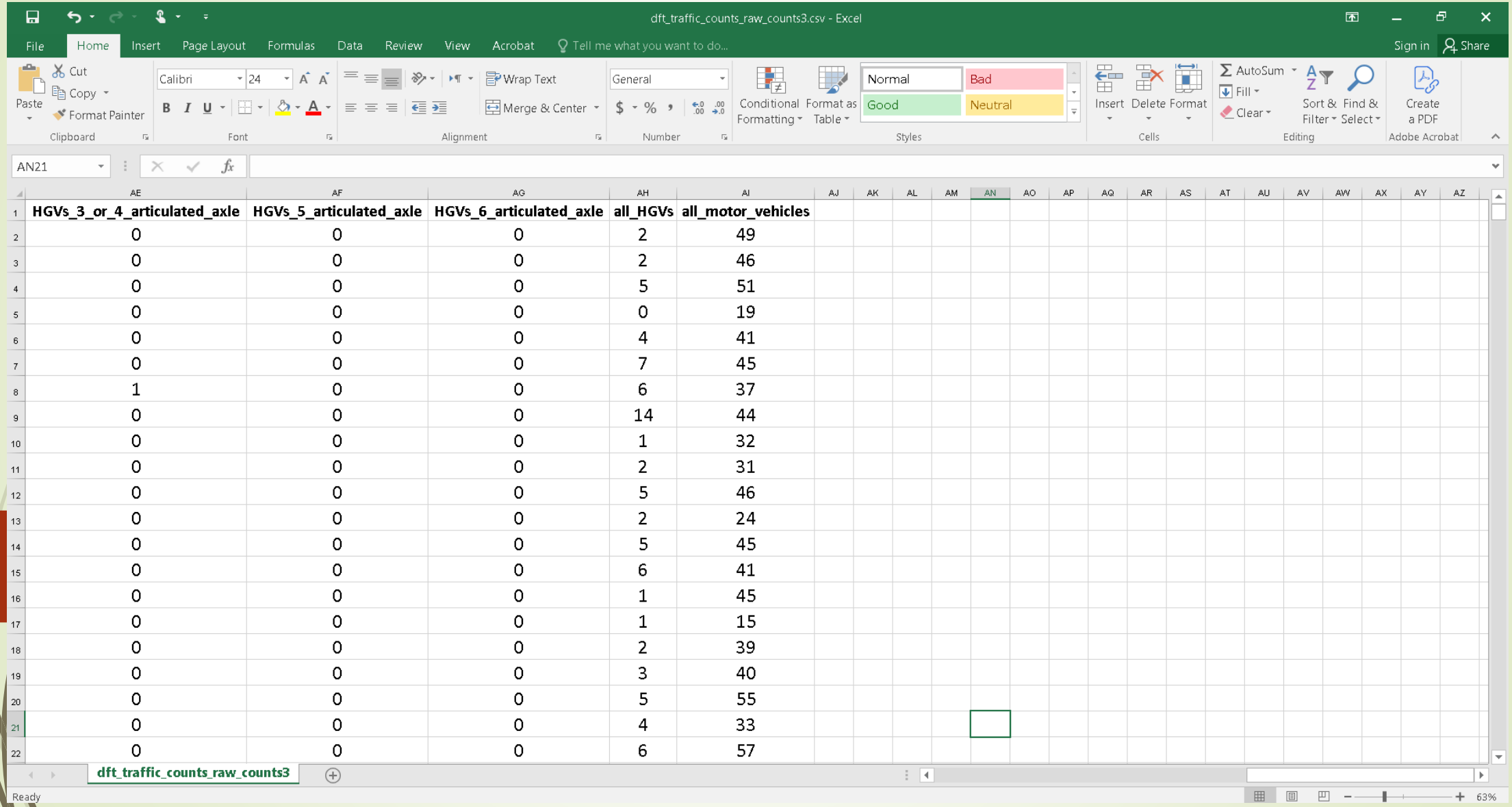
Clipboard Font Alignment Number Styles Cells Editing Adobe Acrobat

W20 | 11

	V	W	X	Y	Z	AA	AB	AC	AD	AE	
1	link_length_miles	pedal_cycles	two_wheeled_motor_vehicles	cars_and_taxis	buses_and_coaches	LGVs	HGVs_2_rigid_axle	HGVs_3_rigid_axle	HGVs_4_or_more_rigid_axle	HGVs_3_or_4_articulated_axle	HG
2	0.19	12	2	27	2	16	2	0	0	0	
3	0.19	10	1	29	1	13	2	0	0	0	
4	0.19	7	0	21	2	23	5	0	0	0	
5	0.19	5	2	3	1	13	0	0	0	0	
6	0.19	5	1	28	4	4	4	0	0	0	
7	0.19	12	0	17	2	19	7	0	0	0	
8	0.19	9	3	19	1	8	5	0	0	1	
9	0.19	19	2	13	2	13	7	7	0	0	
10	0.19	9	1	15	2	13	1	0	0	0	
11	0.19	12	2	23	1	3	2	0	0	0	
12	0.19	3	3	19	1	18	5	0	0	0	
13	0.19	16	1	16	1	4	2	0	0	0	
14	0.19	14	1	17	5	17	5	0	0	0	
15	0.19	12	1	17	3	14	6	0	0	0	
16	0.19	12	3	23	2	16	1	0	0	0	
17	0.19	4	0	4	1	9	1	0	0	0	
18	0.19	7	2	25	1	9	2	0	0	0	
19	0.19	14	4	21	1	11	3	0	0	0	
20	0.19	11	5	25	1	19	5	0	0	0	
21	0.19	17	1	13	1	14	4	0	0	0	
22	0.19	5	2	36	4	9	6	0	0	0	

dft_traffic_counts_raw_counts3

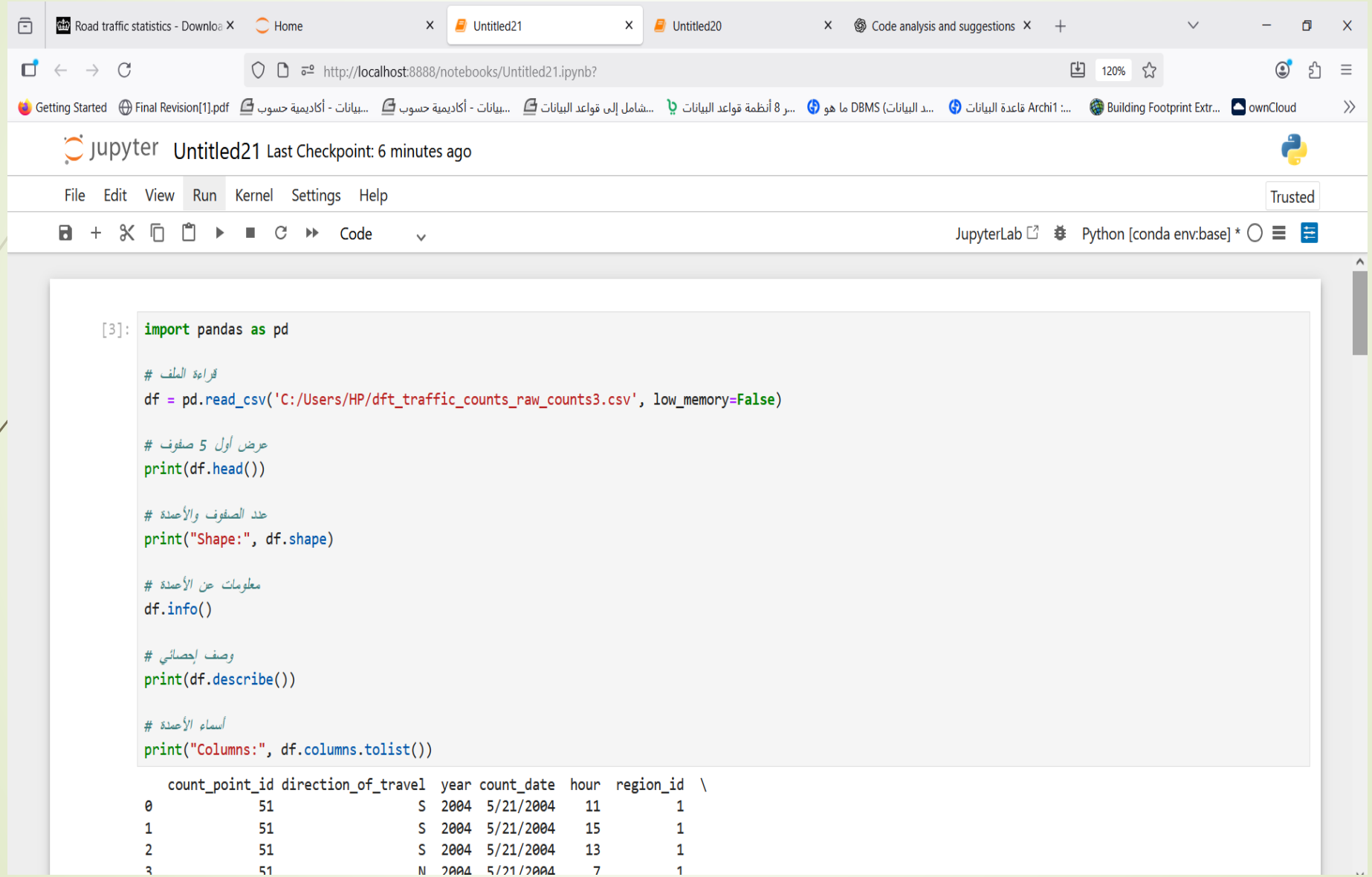
Ready 63%

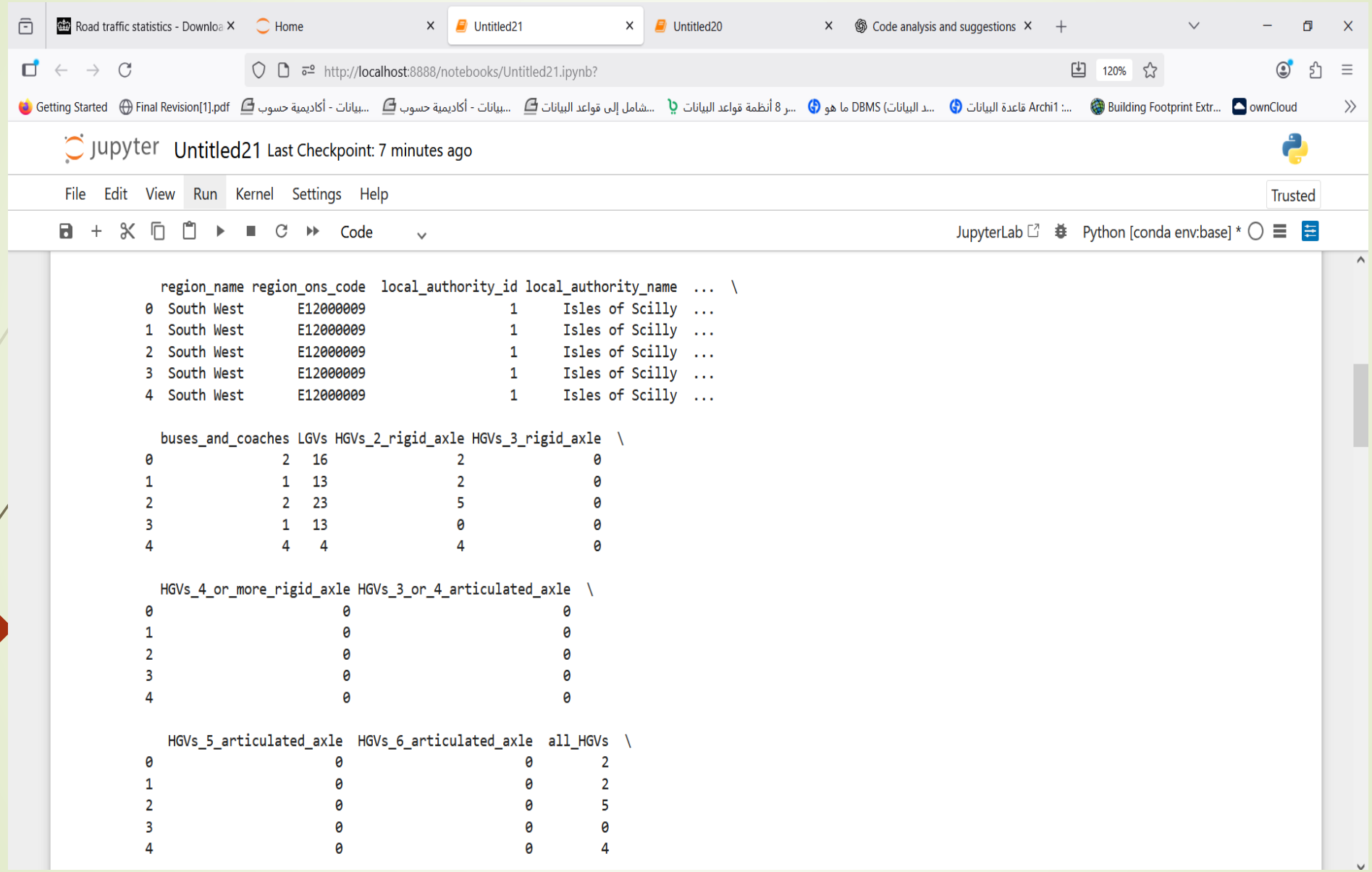


HGVs_3_or_4_articulated_axle	HGVs_5_articulated_axle	HGVs_6_articulated_axle	all_HGVs	all_motor_vehicles
0	0	0	2	49
0	0	0	2	46
0	0	0	5	51
0	0	0	0	19
0	0	0	4	41
0	0	0	7	45
1	0	0	6	37
0	0	0	14	44
0	0	0	1	32
0	0	0	2	31
0	0	0	5	46
0	0	0	2	24
0	0	0	5	45
0	0	0	6	41
0	0	0	1	45
0	0	0	1	15
0	0	0	2	39
0	0	0	3	40
0	0	0	5	55
0	0	0	4	33
0	0	0	6	57

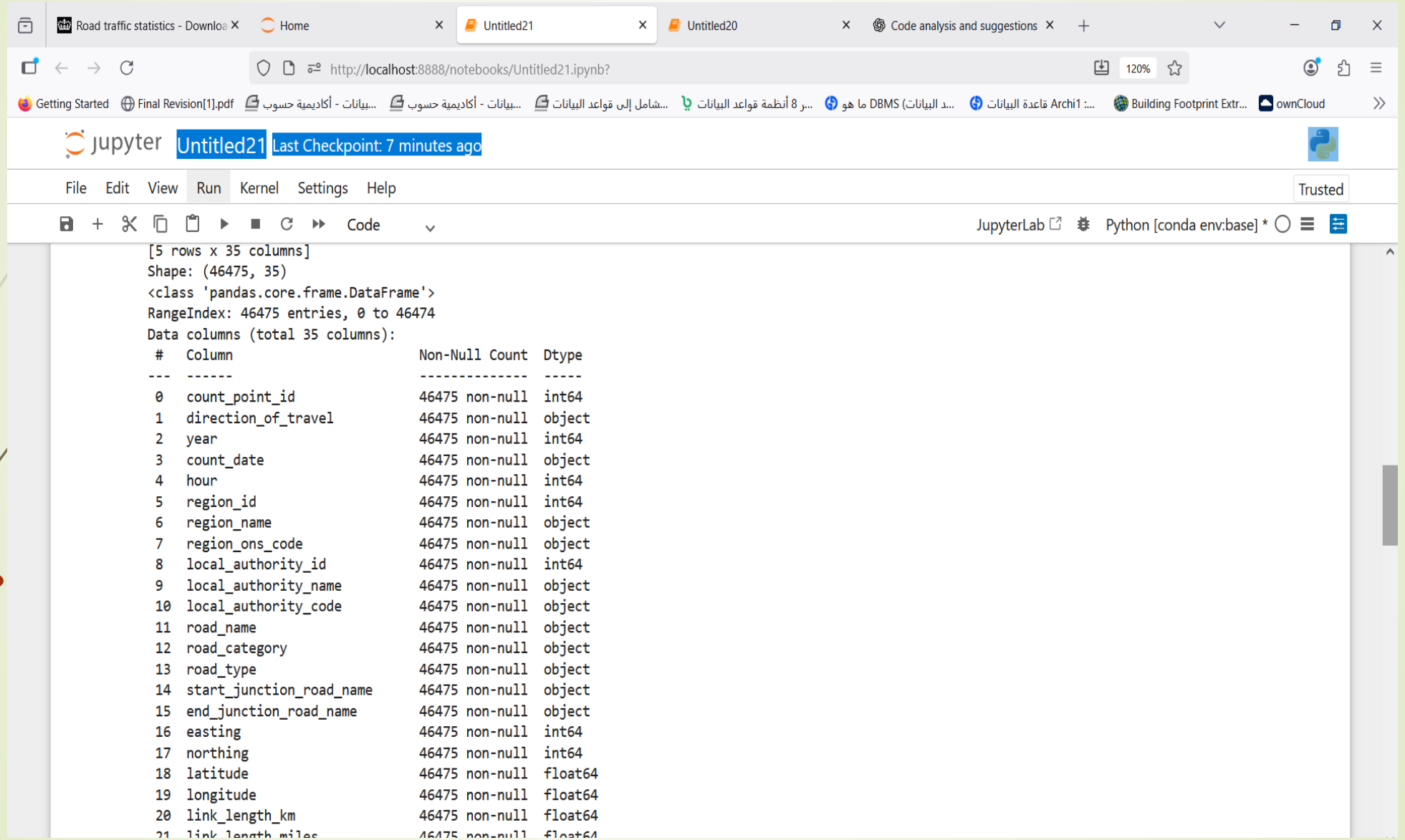
شرح معاني الحقول:(Columns)

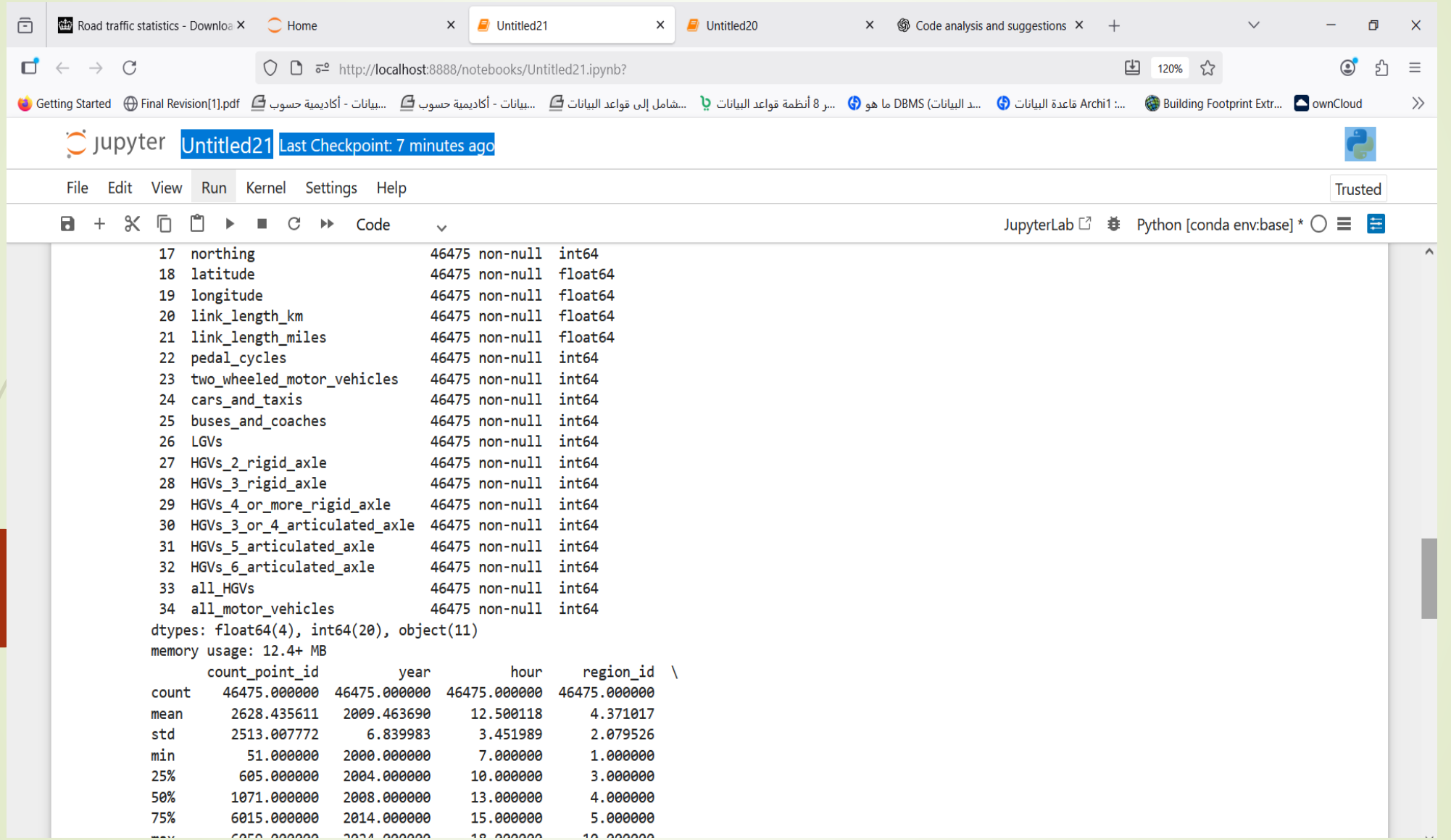
اسم الحقل	المعنى
count_point_id	رقم تعريف نقطة العدّ (مكان تم فيه عدّ المركبات).
direction_of_travel	اتجاه السفر (زي شمال، جنوب، شرق، غرب).
year	سنة تسجيل البيانات.
count_date	التاريخ الفعلي للعدّ (يوم العدّ).
hour	الساعة (الوقت اللي تم فيه العدّ – غالبًا من 0 إلى 23).
region_id	رقم تعريف المنطقة (رقم داخلي يستخدم لتحديد المنطقة).
region_name	اسم المنطقة) زي "East Midlands" أو"London").
region_ons_code	كود المنطقة حسب مكتب الإحصاء الوطني) ONS للاستخدام الإحصائي.(
local_authority_id	رقم تعريف السلطة المحلية (بلدية أو مجلس محلي).
local_authority_name	اسم السلطة المحلية) زي."Leeds City Council“
local_authority_code	كود السلطة المحلية (رمز مختصر).
road_name	اسم الطريق) زيA1 ، .M25).
road_category	فئة الطريق (زي طريق سريع، طريق رئيسي، محلي، الخ).
road_type	نوع الطريق (قد يكون مزيد من التفصيل عن الفئة – مفرد، مزدوج، الخ).
start_junction_road_name	اسم الطريق عند بداية المفرق أو التقاطع.
end_junction_road_name	اسم الطريق عند نهاية المفرق أو التقاطع.
easting	الإحداثي الشرقي (نظام الإحداثيات البريطانية).
northing	الإحداثي الشمالي (نظام الإحداثيات البريطانية).
latitude	دائرة العرض (إحداثيات جغرافية).
longitude	خط الطول (إحداثيات جغرافية).
link_length_km	طول المقطع الطرقي بالكيلومترات.
link_length_miles	طول المقطع الطرقي بالأميال.
pedal_cycles	عدد الدراجات الهوائية.
two_wheeled_motor_vehicles	عدد الدراجات النارية (سكوتر، موتوسيكل).
cars_and_taxis	عدد السيارات العادية والتاكسي.
buses_and_coaches	عدد الحافلات.
LGVs	سيارات النقل الخفيف – (Light Goods Vehicles) زي الفانات الصغيرة.
HGVs_2_rigid_axle	شاحنات ثقيلة بعدد 2 محور صلب.
HGVs_3_rigid_axle	شاحنات ثقيلة بعدد 3 محاور صلبة.
HGVs_4_or_more_rigid_axle	شاحنات ثقيلة بعدد 4 أو أكثر من المحاور الصلبة.
HGVs_3_or_4_articulated_axle	شاحنات ثقيلة مفصلية بعدد 3 أو 4 محاور.
HGVs_5_articulated_axle	شاحنات ثقيلة مفصلية بعدد 5 محاور.
HGVs_6_articulated_axle	شاحنات ثقيلة مفصلية بعدد 6 محاور أو أكثر.
all_HGVs	إجمالي الشاحنات الثقيلة.(HGVs)
all_motor_vehicles	إجمالي جميع المركبات الآلية (من موتوسيكلات لحد الشاحنات).

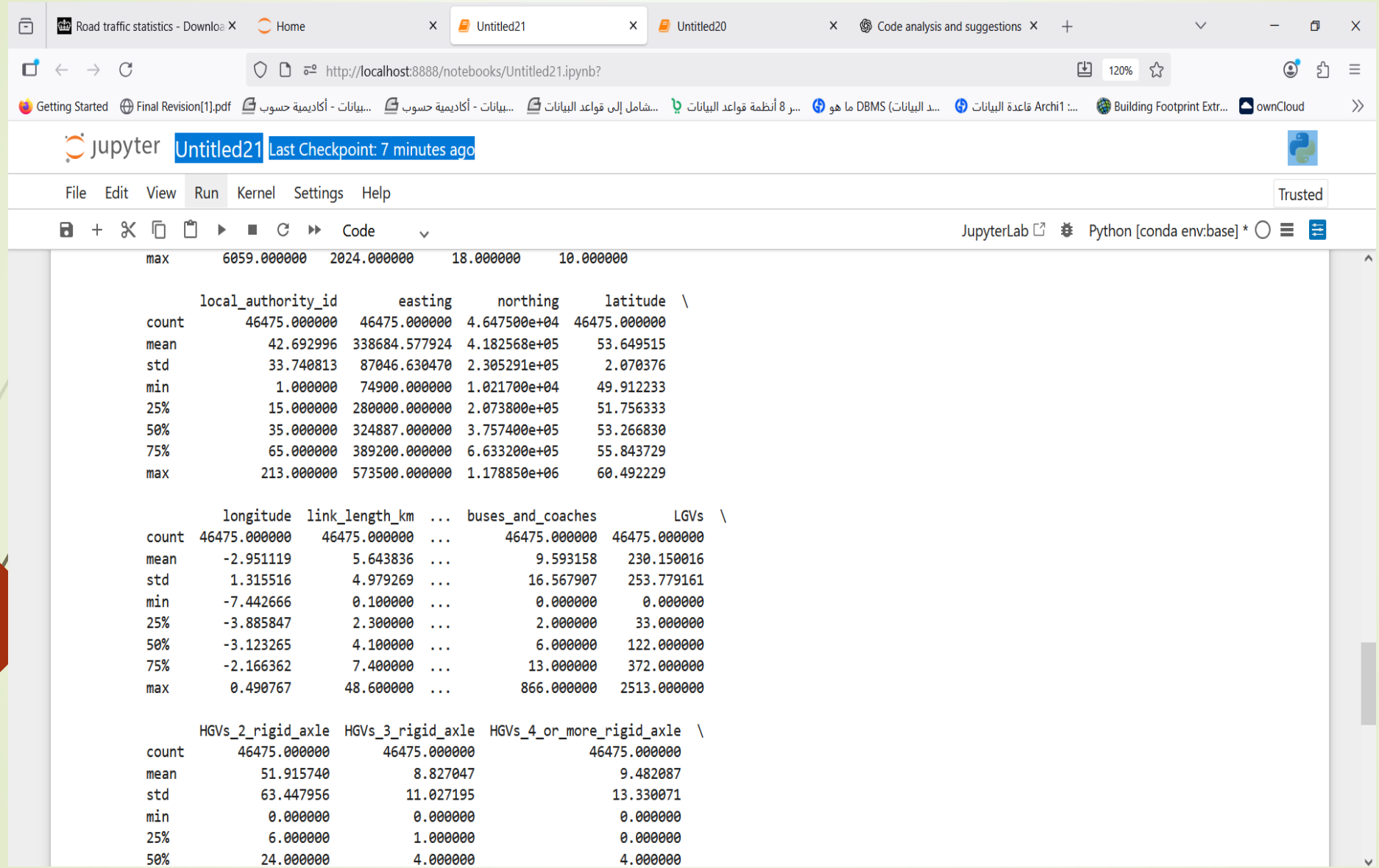

Road traffic statistics - Downloa
Home
Untitled21
Untitled20
Code analysis and suggestions
http://localhost:8888/notebooks/Untitled21.ipynb?
120%
Getting Started
Final Revision[1].pdf
jupyter
Untitled21 Last Checkpoint: 6 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
[3]:
import pandas as pd
# قراءة الملف
df = pd.read_csv('C:/Users/HP/dft_traffic_counts_raw_counts3.csv', low_memory=False)
# عرض أول 5 صفوف
print(df.head())
# عدد الصفوف والأعمدة
print("Shape:", df.shape)
# معلومات عن الأعمدة
df.info()
# وصف إحصائي
print(df.describe())
# أسماء الأعمدة
print("Columns:", df.columns.tolist())
count_point_id direction_of_travel year count_date hour region_id \
0 51 S 2004 5/21/2004 11 1
1 51 S 2004 5/21/2004 15 1
2 51 S 2004 5/21/2004 13 1
3 51 N 2004 5/21/2004 7 1



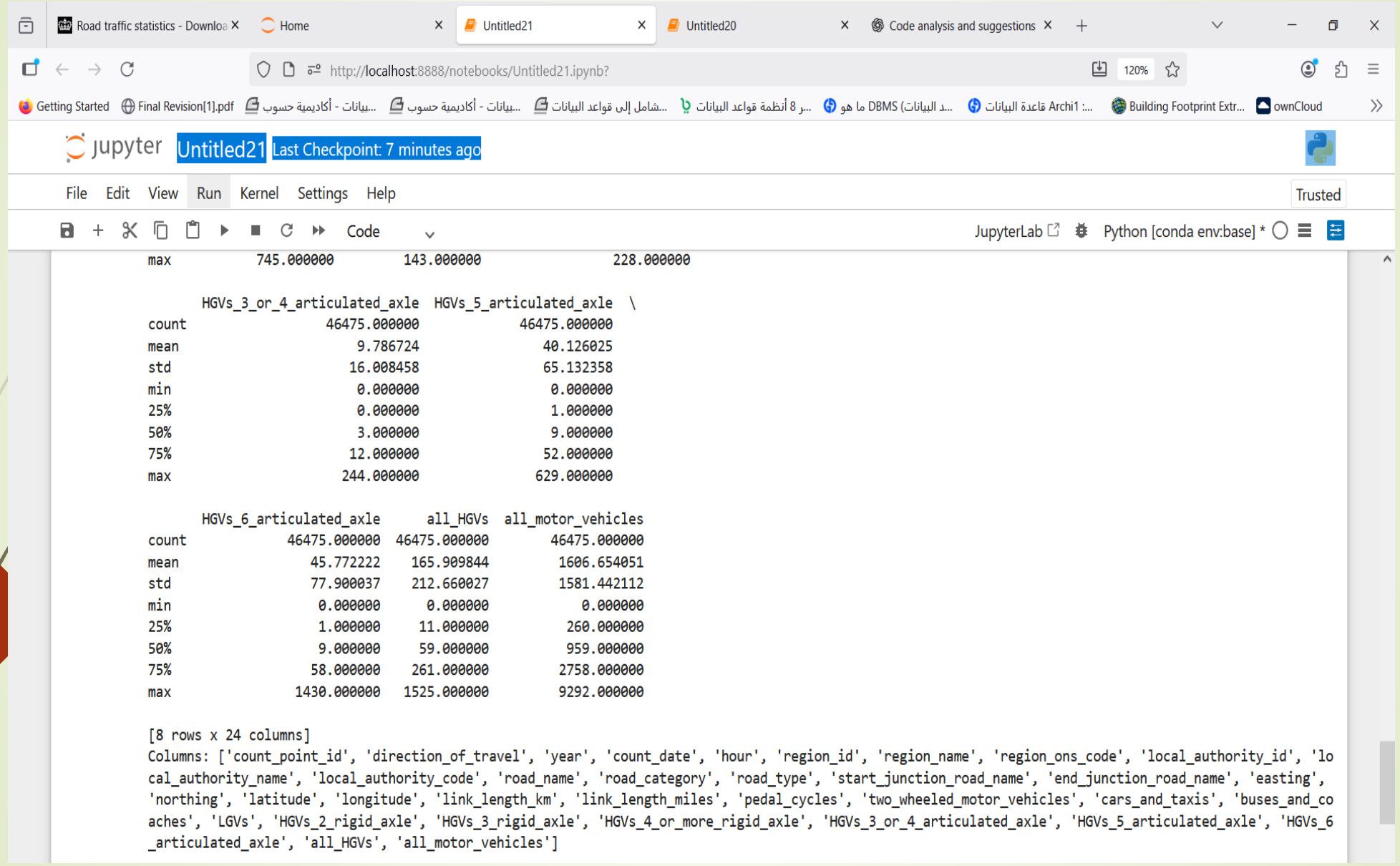

```
  region_name region_ons_code  local_authority_id local_authority_name  ...  \
0  South West       E12000009                   1      Isles of Scilly  ...
1  South West       E12000009                   1      Isles of Scilly  ...
2  South West       E12000009                   1      Isles of Scilly  ...
3  South West       E12000009                   1      Isles of Scilly  ...
4  South West       E12000009                   1      Isles of Scilly  ...

  buses_and_coaches  LGVs  HGVs_2_rigid_axle  HGVs_3_rigid_axle  \
0                 2    16                  2                  0
1                 1    13                  2                  0
2                 2    23                  5                  0
3                 1    13                  0                  0
4                 4     4                  4                  0

  HGVs_4_or_more_rigid_axle  HGVs_3_or_4_articulated_axle  \
0                         0                             0
1                         0                             0
2                         0                             0
3                         0                             0
4                         0                             0

   HGVs_5_articulated_axle  HGVs_6_articulated_axle  all_HGVs  \
0                        0                        0         2
1                        0                        0         2
2                        0                        0         5
3                        0                        0         0
4                        0                        0         4
```


Road traffic statistics - Downloa
Home
Untitled21
Untitled20
Code analysis and suggestions
http://localhost:8888/notebooks/Untitled21.ipynb?
120%
Getting Started
Final Revision[1].pdf
jupyter
Untitled21
Last Checkpoint: 7 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
[5 rows x 35 columns]
Shape: (46475, 35)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46475 entries, 0 to 46474
Data columns (total 35 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 count_point_id 46475 non-null int64
1 direction_of_travel 46475 non-null object
2 year 46475 non-null int64
3 count_date 46475 non-null object
4 hour 46475 non-null int64
5 region_id 46475 non-null int64
6 region_name 46475 non-null object
7 region_ons_code 46475 non-null object
8 local_authority_id 46475 non-null int64
9 local_authority_name 46475 non-null object
10 local_authority_code 46475 non-null object
11 road_name 46475 non-null object
12 road_category 46475 non-null object
13 road_type 46475 non-null object
14 start_junction_road_name 46475 non-null object
15 end_junction_road_name 46475 non-null object
16 easting 46475 non-null int64
17 northing 46475 non-null int64
18 latitude 46475 non-null float64
19 longitude 46475 non-null float64
20 link_length_km 46475 non-null float64


Road traffic statistics - Downloa
Home
Untitled21
Untitled20
Code analysis and suggestions
http://localhost:8888/notebooks/Untitled21.ipynb?
120%
Getting Started
Final Revision[1].pdf
jupyter
Untitled21
Last Checkpoint: 7 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
17 northing 46475 non-null int64
18 latitude 46475 non-null float64
19 longitude 46475 non-null float64
20 link_length_km 46475 non-null float64
21 link_length_miles 46475 non-null float64
22 pedal_cycles 46475 non-null int64
23 two_wheeled_motor_vehicles 46475 non-null int64
24 cars_and_taxis 46475 non-null int64
25 buses_and_coaches 46475 non-null int64
26 LGVs 46475 non-null int64
27 HGVs_2_rigid_axle 46475 non-null int64
28 HGVs_3_rigid_axle 46475 non-null int64
29 HGVs_4_or_more_rigid_axle 46475 non-null int64
30 HGVs_3_or_4_articulated_axle 46475 non-null int64
31 HGVs_5_articulated_axle 46475 non-null int64
32 HGVs_6_articulated_axle 46475 non-null int64
33 all_HGVs 46475 non-null int64
34 all_motor_vehicles 46475 non-null int64
dtypes: float64(4), int64(20), object(11)
memory usage: 12.4+ MB
count_point_id year hour region_id \
count 46475.000000 46475.000000 46475.000000 46475.000000
mean 2628.435611 2009.463690 12.500118 4.371017
std 2513.007772 6.839983 3.451989 2.079526
min 51.000000 2000.000000 7.000000 1.000000
25% 605.000000 2004.000000 10.000000 3.000000
50% 1071.000000 2008.000000 13.000000 4.000000
75% 6015.000000 2014.000000 15.000000 5.000000




```
max       6059.000000   2024.000000     18.000000     10.000000

       local_authority_id       easting      northing      latitude  \
count        46475.000000  46475.000000  4.647500e+04  46475.000000
mean            42.692996 338684.577924  4.182568e+05     53.649515
std             33.740813  87046.630470  2.305291e+05      2.070376
min              1.000000  74900.000000  1.021700e+04     49.912233
25%             15.000000 280000.000000  2.073800e+05     51.756333
50%             35.000000 324887.000000  3.757400e+05     53.266830
75%             65.000000 389200.000000  6.633200e+05     55.843729
max            213.000000 573500.000000  1.178850e+06     60.492229

          longitude  link_length_km  ...  buses_and_coaches          LGVs  \
count  46475.000000    46475.000000  ...       46475.000000  46475.000000
mean      -2.951119        5.643836  ...           9.593158    230.150016
std        1.315516        4.979269  ...          16.567907    253.779161
min       -7.442666        0.100000  ...           0.000000      0.000000
25%       -3.885847        2.300000  ...           2.000000     33.000000
50%       -3.123265        4.100000  ...           6.000000    122.000000
75%       -2.166362        7.400000  ...          13.000000    372.000000
max        0.490767       48.600000  ...         866.000000   2513.000000

       HGVs_2_rigid_axle  HGVs_3_rigid_axle  HGVs_4_or_more_rigid_axle  \
count       46475.000000       46475.000000                46475.000000
mean           51.915740           8.827047                    9.482087
std            63.447956          11.027195                   13.330071
min             0.000000           0.000000                    0.000000
25%             6.000000           1.000000                    0.000000
50%            24.000000           4.000000                    4.000000
```




```
max          745.000000        143.000000               228.000000

       HGVs_3_or_4_articulated_axle  HGVs_5_articulated_axle  \
count                  46475.000000             46475.000000
mean                       9.786724                40.126025
std                       16.008458                65.132358
min                        0.000000                 0.000000
25%                        0.000000                 1.000000
50%                        3.000000                 9.000000
75%                       12.000000                52.000000
max                      244.000000               629.000000

       HGVs_6_articulated_axle      all_HGVs  all_motor_vehicles
count             46475.000000  46475.000000        46475.000000
mean                 45.772222    165.909844         1606.654051
std                  77.900037    212.660027         1581.442112
min                   0.000000      0.000000            0.000000
25%                   1.000000     11.000000          260.000000
50%                   9.000000     59.000000          959.000000
75%                  58.000000    261.000000         2758.000000
max                1430.000000   1525.000000         9292.000000

[8 rows x 24 columns]
Columns: ['count_point_id', 'direction_of_travel', 'year', 'count_date', 'hour', 'region_id', 'region_name', 'region_ons_code', 'local_authority_id', 'lo
cal_authority_name', 'local_authority_code', 'road_name', 'road_category', 'road_type', 'start_junction_road_name', 'end_junction_road_name', 'easting',
'northing', 'latitude', 'longitude', 'link_length_km', 'link_length_miles', 'pedal_cycles', 'two_wheeled_motor_vehicles', 'cars_and_taxis', 'buses_and_co
aches', 'LGVs', 'HGVs_2_rigid_axle', 'HGVs_3_rigid_axle', 'HGVs_4_or_more_rigid_axle', 'HGVs_3_or_4_articulated_axle', 'HGVs_5_articulated_axle', 'HGVs_6
_articulated_axle', 'all_HGVs', 'all_motor_vehicles']
```

- قراءة البيانات وتنظيفها
- تحويل الأعمدة الزمنية
- التحقق من القيم الشاذة
- هندسة خصائص جديدة ((feature engineering
- تصدير البيانات إلى قاعدة بيانات SQL
- إنشاء GeoDataFrameوتصدير البيانات إلى GeoJSONوShapefile
- تقرير جودة البيانات

```
# 1. Initial Setup and Data Loading
# -------------------------------------
import pandas as pd
import numpy as np
import geopandas as gpd
from sqlalchemy import create_engine
from sqlalchemy.engine import URL

# Load the CSV data
df = pd.read_csv('C:/Users/HP/dft_traffic_counts_raw_counts3.csv',
low_memory=False)
print(f"Dataset shape: {df.shape}")
print(df.info())
```

```
Dataset shape: (46475, 35)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46475 entries, 0 to 46474
Data columns (total 35 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   count_point_id                  46475 non-null  int64
 1   direction_of_travel             46475 non-null  object
 2   year                            46475 non-null  int64
 3   count_date                      46475 non-null  object
 4   hour                            46475 non-null  int64
 5   region_id                       46475 non-null  int64
 6   region_name                     46475 non-null  object
 7   region_ons_code                 46475 non-null  object
 8   local_authority_id              46475 non-null  int64
 9   local_authority_name            46475 non-null  object
 10  local_authority_code            46475 non-null  object
 11  road_name                       46475 non-null  object
 12  road_category                   46475 non-null  object
 13  road_type                       46475 non-null  object
 14  start_junction_road_name        46475 non-null  object
 15  end_junction_road_name          46475 non-null  object
 16  easting                         46475 non-null  int64
 17  northing                        46475 non-null  int64
 18  latitude                        46475 non-null  float64
 19  longitude                       46475 non-null  float64
 20  link_length_km                  46475 non-null  float64
 21  link_length_miles               46475 non-null  float64
 22  pedal_cycles                    46475 non-null  int64
 23  two_wheeled_motor_vehicles      46475 non-null  int64
 24  cars_and_taxis                  46475 non-null  int64
 25  buses_and_coaches               46475 non-null  int64
 26  LGVs                            46475 non-null  int64
 27  HGVs_2_rigid_axle               46475 non-null  int64
 28  HGVs_3_rigid_axle               46475 non-null  int64
 29  HGVs_4_or_more_rigid_axle       46475 non-null  int64
 30  HGVs_3_or_4_articulated_axle    46475 non-null  int64
 31  HGVs_5_articulated_axle         46475 non-null  int64
 32  HGVs_6_articulated_axle         46475 non-null  int64
 33  all_HGVs                        46475 non-null  int64
 34  all_motor_vehicles              46475 non-null  int64
dtypes: float64(4), int64(20), object(11)
memory usage: 12.4+ MB
None
```

```
# ------------------------------------
# 2. Data Exploration and Cleaning
# ------------------------------------
```

```
# Check for missing values
missing_values = df.isnull().sum()
print("\nMissing values per column:")
print(missing_values[missing_values > 0])
```

```
Missing values per column:
Series([], dtype: int64)
```

```
# Handle missing values
df.fillna({
    'link_length_km': 0,
    'link_length_miles': 0,
    'pedal_cycles': 0,
}, inplace=True)
```

```
Missing values per column:
Series([], dtype: int64)
```

```
# Convert count_date to datetime (auto format)
df['count_date'] = pd.to_datetime(df['count_date'], errors='coerce', dayfirst=False)

# Check data types
print("\nData types after conversion:")
print(df.dtypes)
```

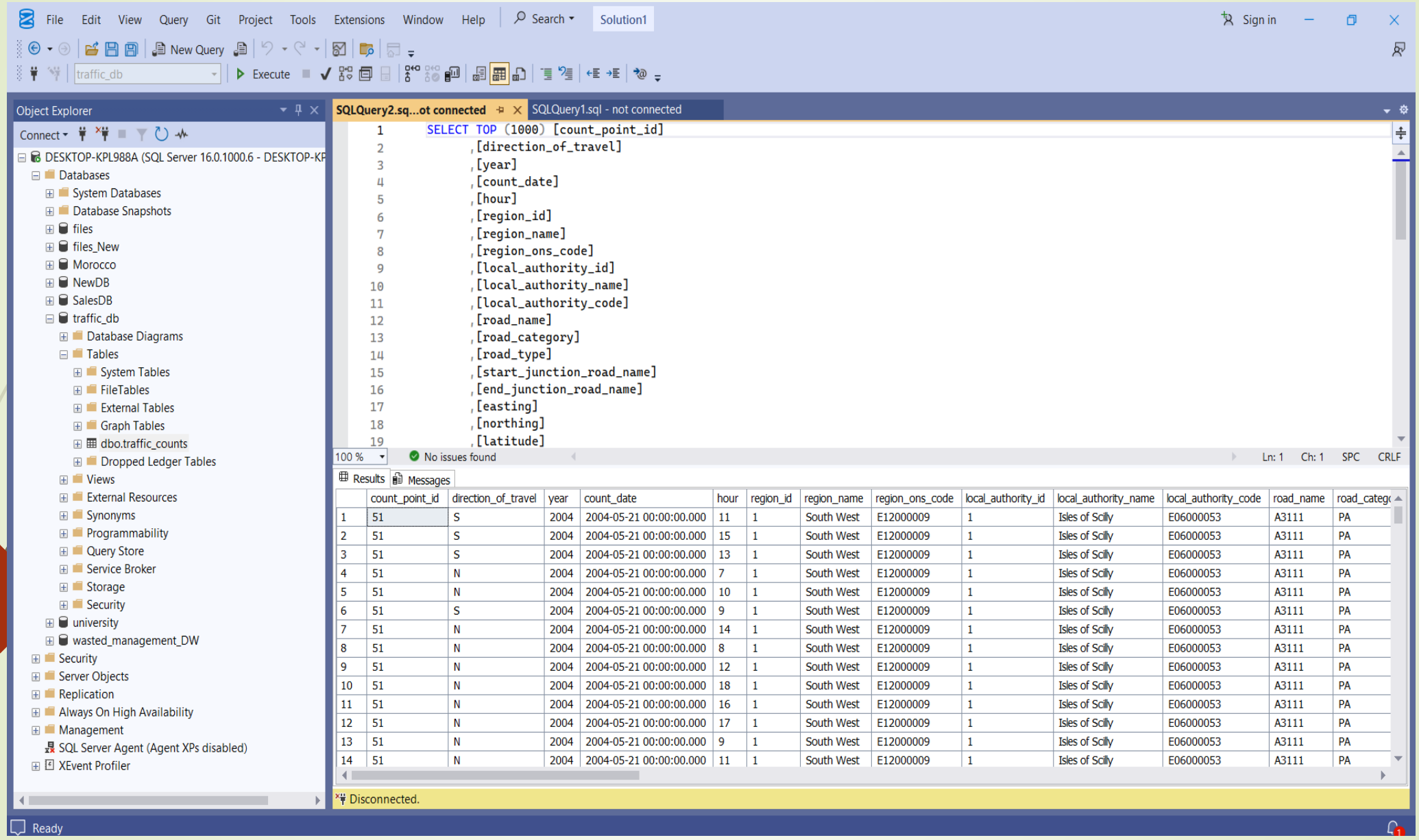
```
Data types after conversion:
count_point_id                      int64
direction_of_travel                object
year                                int64
count_date                 datetime64[ns]
hour                                int64
region_id                           int64
region_name                        object
region_ons_code                    object
local_authority_id                  int64
local_authority_name               object
local_authority_code               object
road_name                          object
road_category                      object
road_type                          object
start_junction_road_name           object
end_junction_road_name             object
easting                             int64
northing                            int64
latitude                          float64
longitude                         float64
link_length_km                    float64
link_length_miles                 float64
pedal_cycles                        int64
two_wheeled_motor_vehicles          int64
cars_and_taxis                      int64
buses_and_coaches                   int64
LGVs                                int64
HGVs_2_rigid_axle                   int64
HGVs_3_rigid_axle                   int64
HGVs_4_or_more_rigid_axle           int64
HGVs_3_or_4_articulated_axle        int64
HGVs_5_articulated_axle             int64
HGVs_6_articulated_axle             int64
all_HGVs                            int64
all_motor_vehicles                  int64
dtype: object
```

```
# ----------------------------------------

# 3. Data Validation and Quality Checks

# ----------------------------------------

numeric_columns = ['hour', 'link_length_km', 'pedal_cycles', 'cars_and_taxis',
'all_motor_vehicles']

for col in numeric_columns:

    if col in df.columns:

        print(f"{col} range: {df[col].min()} - {df[col].max()}")


print("\nRoad categories:", df['road_category'].unique())

print("Road types:", df['road_type'].unique())


valid_coords = (df['latitude'].between(-90, 90)) & (df['longitude'].between(-180, 180))

print(f"Invalid coordinates: {len(df[~valid_coords])}")


hour range: 7 - 18
link_length_km range: 0.1 - 48.6
pedal_cycles range: 0 - 219
cars_and_taxis range: 0 - 7642
all_motor_vehicles range: 0 - 9292

Road categories: ['PA' 'TM' 'TA']
Road types: ['Major']
Invalid coordinates: 0
```

```
# ----------------------------------------
# 4. Data Transformation and Feature Engineering
# ----------------------------------------
```

```
df['total_vehicles'] = (
    df['pedal_cycles'] +
    df['two_wheeled_motor_vehicles'] +
    df['cars_and_taxis'] +
    df['buses_and_coaches'] +
    df['LGVs'] +
    df['all_HGVs']
)

df['vehicle_density'] = df['total_vehicles'] / df['link_length_km'].replace(0, 0.001)

df['day_of_week'] = df['count_date'].dt.day_name()
df['month'] = df['count_date'].dt.month
df['season'] = df['month'].apply(lambda x: 'Winter' if x in [12, 1, 2] else
                                 'Spring' if x in [3, 4, 5] else
                                 'Summer' if x in [6, 7, 8] else
                                 'Autumn')

df['hgv_percentage'] = df['all_HGVs'] / df['total_vehicles'].replace(0, 1)
df['bicycle_percentage'] = df['pedal_cycles'] / df['total_vehicles'].replace(0, 1)
```

```
# ----------------------------------------
# 5. Save to Parquet (optional)
# ----------------------------------------
df.to_parquet('traffic_data_processed.parquet', engine='pyarrow')

# ----------------------------------------
from sqlalchemy import create_engine
from sqlalchemy.engine import URL

connection_url = URL.create(
    "mssql+pyodbc",
    host="localhost",
    database="traffic_db",
    query={
        "driver": "ODBC Driver 17 for SQL Server",
        "Trusted_Connection": "yes"
    }
)

engine = create_engine(connection_url)

df.to_sql('traffic_counts', engine, if_exists='replace', index=False)
```


File Edit View Query Git Project Tools Extensions Window Help
Search
Solution1
Sign in
New Query
traffic_db
Execute
Object Explorer
Connect
DESKTOP-KPL988A (SQL Server 16.0.1000.6 - DESKTOP-KP
Databases
System Databases
Database Snapshots
files
files_New
Morocco
NewDB
SalesDB
traffic_db
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.traffic_counts
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
university
wasted_management_DW
Security
Server Objects
Replication
Always On High Availability
Management
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
SQLQuery2.sq...ot connected
SQLQuery1.sql - not connected
SELECT TOP (1000) [count_point_id]
,[direction_of_travel]
,[year]
,[count_date]
,[hour]
,[region_id]
,[region_name]
,[region_ons_code]
,[local_authority_id]
,[local_authority_name]
,[local_authority_code]
,[road_name]
,[road_category]
,[road_type]
,[start_junction_road_name]
,[end_junction_road_name]
,[easting]
,[northing]
,[latitude]
100 %
No issues found
Ln: 1 Ch: 1 SPC CRLF
Results
Messages
count_point_id direction_of_travel year count_date hour region_id region_name region_ons_code local_authority_id local_authority_name local_authority_code road_name road_categ
1 51 S 2004 2004-05-21 00:00:00.000 11 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
2 51 S 2004 2004-05-21 00:00:00.000 15 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
3 51 S 2004 2004-05-21 00:00:00.000 13 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
4 51 N 2004 2004-05-21 00:00:00.000 7 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
5 51 N 2004 2004-05-21 00:00:00.000 10 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
6 51 S 2004 2004-05-21 00:00:00.000 9 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
7 51 N 2004 2004-05-21 00:00:00.000 14 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
8 51 N 2004 2004-05-21 00:00:00.000 8 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
9 51 N 2004 2004-05-21 00:00:00.000 12 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
10 51 N 2004 2004-05-21 00:00:00.000 18 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
11 51 N 2004 2004-05-21 00:00:00.000 16 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
12 51 N 2004 2004-05-21 00:00:00.000 17 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
13 51 N 2004 2004-05-21 00:00:00.000 9 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
14 51 N 2004 2004-05-21 00:00:00.000 11 1 South West E12000009 1 Isles of Scilly E06000053 A3111 PA
Disconnected.
Ready

----

7. Aggregate Data for Reporting

```
daily_totals = df.groupby(['count_date', 'region_name', 'local_authority_name']).agg({
    'total_vehicles': 'sum',
    'pedal_cycles': 'sum',
    'all_HGVs': 'sum'
}).reset_index()
```

```
     count_date    region_name local_authority_name  total_vehicles  \
0    2000-03-17       Scotland         East Lothian           13320
1    2000-03-17          Wales              Cardiff           17916
2    2000-03-21     North West           Warrington           72633
3    2000-03-22  East Midlands     Northamptonshire            9259
4    2000-03-22       Scotland    North Lanarkshire            7475
...         ...            ...                  ...             ...
1936 2024-09-18       Scotland     Shetland Islands            3295
1937 2024-09-18          Wales              Gwynedd           10587
1938 2024-09-26     North West            Stockport           30383
1939 2024-10-01          Wales         Denbighshire           32774
1940 2024-10-08     North West           Cumberland           38656

      pedal_cycles  all_HGVs
0               62       418
1               35       526
2                0     11003
3                1      1179
4               17       439
...            ...       ...
1936            13        43
1937             8       239
1938           143       627
1939             0      2998
1940             1      7344

[1941 rows x 6 columns]
```

```
# ----------------------------------------
# 8. Save GeoJSON (Optional for Spatial Analysis)
# ----------------------------------------
gdf = gpd.GeoDataFrame(
    df,
    geometry=gpd.points_from_xy(df.longitude, df.latitude),
    crs="EPSG:4326"
)

gdf.to_file("traffic_points.geojson", driver='GeoJSON')

# ---------------------------------------- Save to Shape File:
import geopandas as gpd

# إنشاء GeoDataFrame من بيانات df
gdf = gpd.GeoDataFrame(
    df,
    geometry=gpd.points_from_xy(df.longitude, df.latitude),
    crs="EPSG:4326"  # WGS 84
)

# حفظ البيانات كملف Shapefile
gdf.to_file("traffic_points_shapefile.shp", driver='ESRI Shapefile')
```

```
# ------------------------------------

# 9. Data Quality Report

# ------------------------------------

def check_data_quality(df):
    checks = {
        'total_rows': len(df),
        'missing_coordinates': df['latitude'].isnull().sum() + df['longitude'].isnull().sum(),
        'negative_traffic_counts': (df[['pedal_cycles', 'cars_and_taxis', 'all_HGVs']] < 0).any().any(),
        'duplicate_records': df.duplicated().sum()
    }
    return checks

quality_report = check_data_quality(df)

print("\nData Quality Report:")
for check, result in quality_report.items():
    print(f"{check}: {result}")
```

قيم غير منطقية(Logical Errors)

العمود	التحقق
hour	هل القيم بين 0 و 23 فقط؟
total_vehicles	هل في قيم سالبة أو أكبر من مليون؟
link_length_km	هل يوجد قيم سالبة؟
vehicle_density	هل القيمة منطقية؟ مثلًا مش 100,000 مركبة/كم
bicycle_percentage أوhgv_percentage	هل أكثر من 1 أو أقل من 0؟ (نسب غير منطقية)

```
print("Invalid hours:", df[~df['hour'].between(0, 23)].shape[0])

print("Negative total_vehicles:", (df['total_vehicles'] < 0).sum())

print("Suspiciously high density (> 10,000):", (df['vehicle_density'] > 10000).sum())

print("Out-of-range percentages:", ((df['hgv_percentage'] > 1) | (df['hgv_percentage'] <
0)).sum())
```

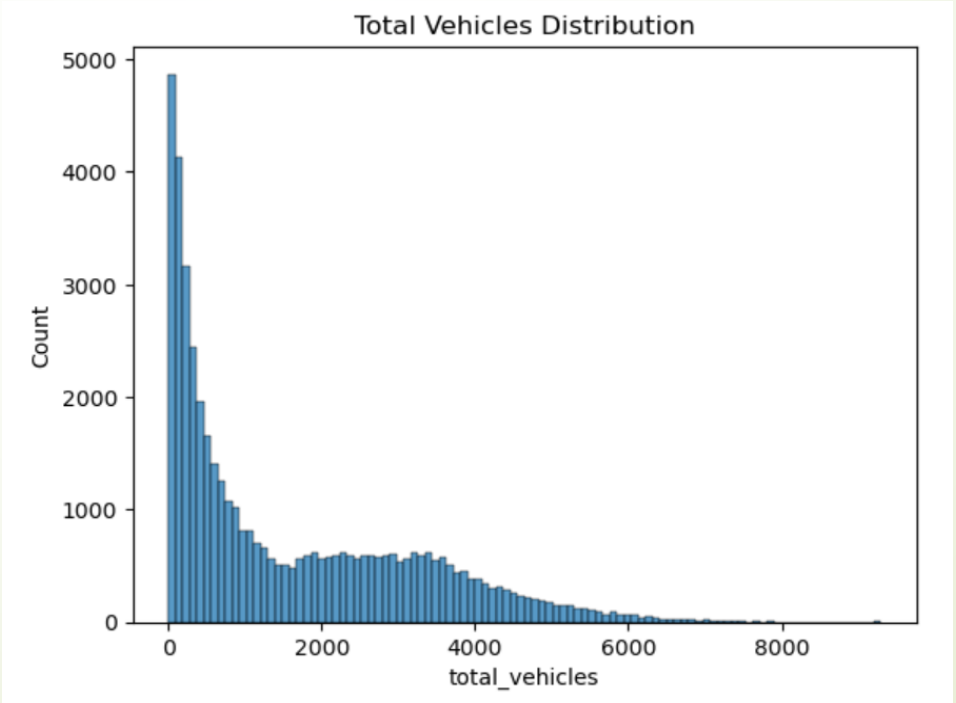
```
Invalid hours: 0
Negative total_vehicles: 0
Suspiciously high density (> 10,000): 0
Out-of-range percentages: 0
```

2. توزيع البيانات

لو البيانات فيها skewميل قوي أو outliersكثير، ممكن يؤثر على التحليل الإحصائي أو النمذجة.

📌استخدم هذا الكود لرؤية التوزيع:

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

sns.histplot(df['total_vehicles'], bins=100)
plt.title("Total Vehicles Distribution")
plt.show()
```



3.تواريخ غير منطقية

- هل عندك تواريخ في المستقبل؟
- هل بعض السجلات تواريخها ناقصة(NaT) ؟

```
from datetime import datetime


print("Future dates:", df[df['count_date'] > datetime.today()].shape[0])

print("Missing dates:", df['count_date'].isnull().sum())
```

```
Future dates: 0
Missing dates: 0
```

4. نقاط جغرافية خارج بريطانيا؟ إذا البيانات من UKمثل DfT

- تأكد إن latitudeو longitudeداخل نطاق المملكة المتحدة

📌تقريبًا:

- Lat: 49إلى 61
- Lon: -9إلى 2

```
uk_bounds = df[
    (df['latitude'].between(49, 61)) &
    (df['longitude'].between(-9, 2))
]
print("Outside UK bounds:", df.shape[0] - uk_bounds.shape[0])
```

```
Outside UK bounds: 0
```

5.تكرار غير متوقع في الـ ID أو المواقع

هل عندك تكرار في count_point_id + count_date + hour ؟ المفترض تكون فريدة.

```
duplicate_keys = df.duplicated(subset=['count_point_id', 'count_date', 'hour']).sum()
print("Duplicated count_point_id + count_date + hour:", duplicate_keys)
```

```
Duplicated count_point_id + count_date + hour: 23078
```

6.التناسق بين الأعمدة

مثلًا total_vehicles :هل يساوي فعليًا مجموع مكوناته؟

```
df['calc_total'] = (
    df['pedal_cycles'] +
    df['two_wheeled_motor_vehicles'] +
    df['cars_and_taxis'] +
    df['buses_and_coaches'] +
    df['LGVs'] +
    df['all_HGVs']
(
print("Mismatch in total_vehicles:", (df['total_vehicles'] != df['calc_total']).sum())
```

```
Mismatch in total_vehicles: 0
```

7. التغطية الزمنية للبيانات

كم عدد الأيام المختلفة في البيانات؟

هل التواريخ متصلة؟ هل عندك فجوات؟

```
date_range = df['count_date'].dropna().sort_values().unique()
print(f"Total distinct dates: {len(date_range)}")
print(f"From {date_range.min()} to {date_range.max()}")
```

```
Total distinct dates: 1269
From 2000-03-17 00:00:00 to 2024-10-08 00:00:00
```

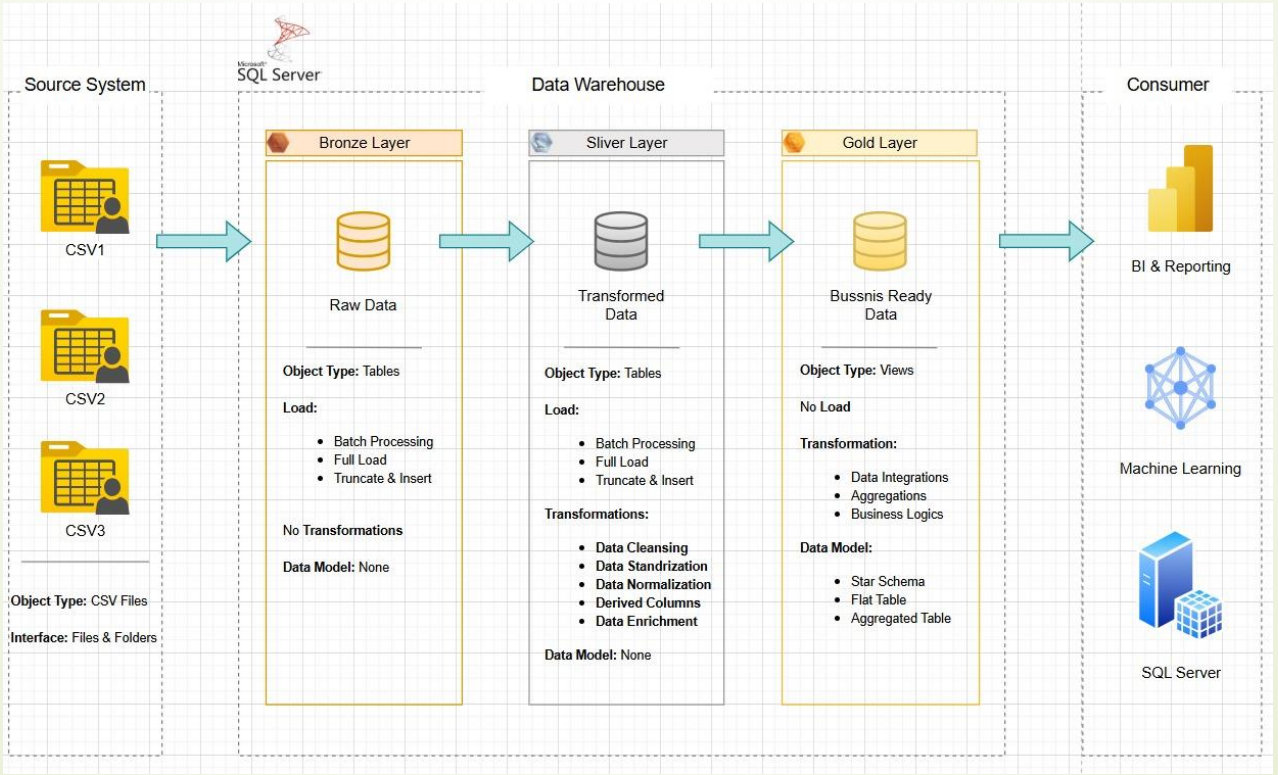
8.تحليل المناطق الفارغة(spatial gaps)

هل هناك مناطق بلا بيانات إطلاقًا؟

كم عدد النقاط الجغرافية الفريدة؟

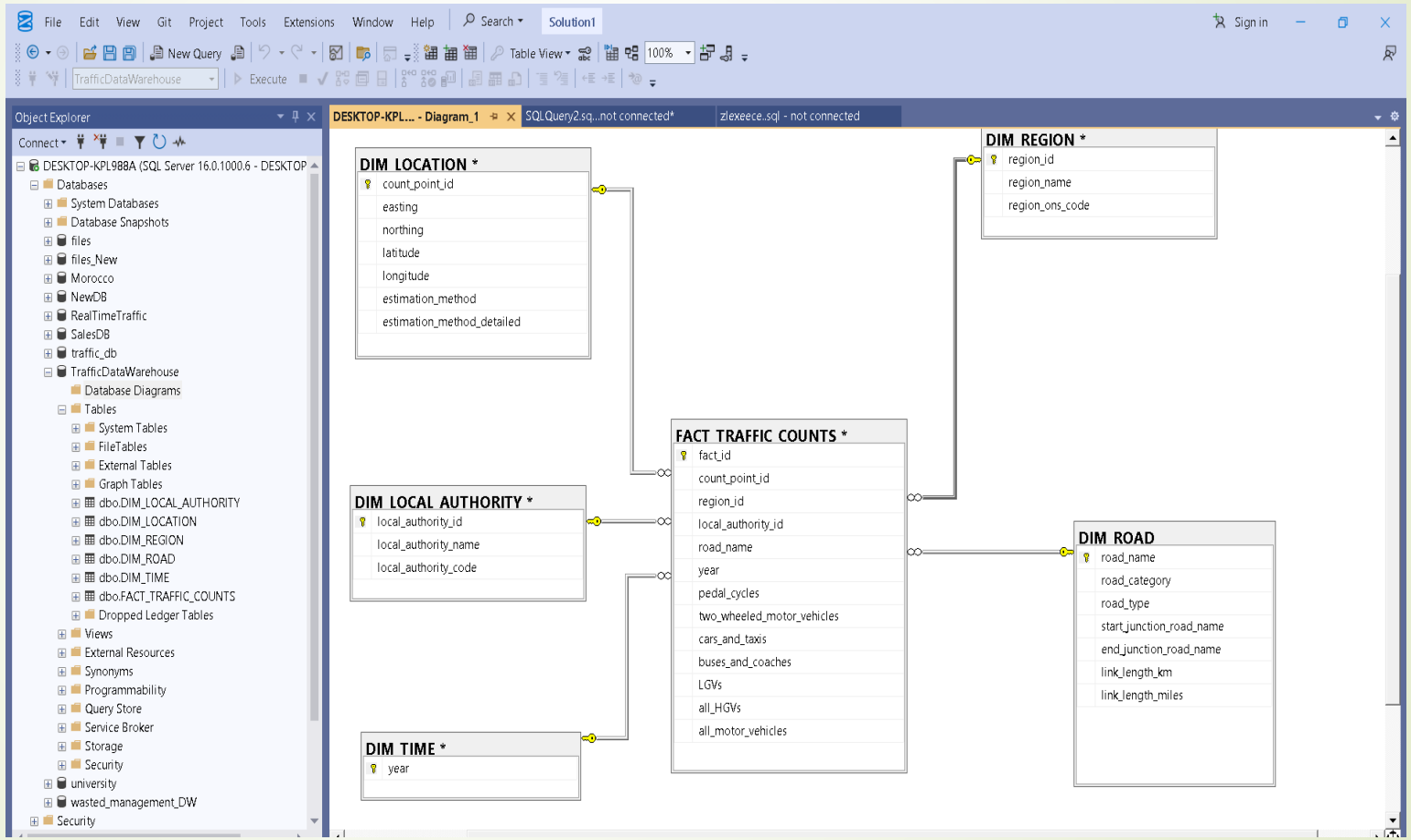
```
print("Unique count locations:", df[['latitude', 'longitude']].drop_duplicates().shape[0])
```

```
Unique count locations: 467
```


Source System
SQL Server
Data Warehouse
Consumer
CSV1
CSV2
CSV3
Object Type: CSV Files
Interface: Files & Folders
Bronze Layer
Raw Data
Object Type: Tables
Load:
Batch Processing
Full Load
Truncate & Insert
No Transformations
Data Model: None
Sliver Layer
Transformed Data
Object Type: Tables
Load:
Batch Processing
Full Load
Truncate & Insert
Transformations:
Data Cleansing
Data Standrization
Data Normalization
Derived Columns
Data Enrichment
Data Model: None
Gold Layer
Bussnis Ready Data
Object Type: Views
No Load
Transformation:
Data Integrations
Aggregations
Business Logics
Data Model:
Star Schema
Flat Table
Aggregated Table
BI & Reporting
Machine Learning
SQL Server

SQL Server

Data After Restructure (Star Schema)



Cleaning Data:

```
import pandas as pd


# 🗁 1. تحميل الملف الأصلي
file_path = "dft_traffic_counts_raw_counts3.csv"  # ضع اسم ملفك هنا
df = pd.read_csv(file_path)


# ✓ 2. إزالة الصفوف المكررة
df = df.drop_duplicates()


# ✓ 3. إنشاء جداول الأبعاد (Dimensions)


# DIM_REGION
dim_region = df[['region_id', 'region_name',
'region_ons_code']].drop_duplicates().reset_index(drop=True)
```

```
# DIM_LOCAL_AUTHORITY

dim_local_authority = df[['local_authority_id', 'local_authority_name',
'local_authority_code']].drop_duplicates().reset_index(drop=True)


# DIM_LOCATION

dim_location = df[['count_point_id', 'easting', 'northing', 'latitude',
'longitude']].drop_duplicates().reset_index(drop=True)

dim_location['estimation_method'] = 'N/A'

dim_location['estimation_method_detailed'] = 'N/A'


# DIM_ROAD

dim_road = df[['road_name', 'road_category', 'road_type', 'start_junction_road_name',

            'end_junction_road_name', 'link_length_km',
'link_length_miles']].drop_duplicates().reset_index(drop=True)


# DIM_TIME

dim_time = df[['year']].drop_duplicates().reset_index(drop=True)
```

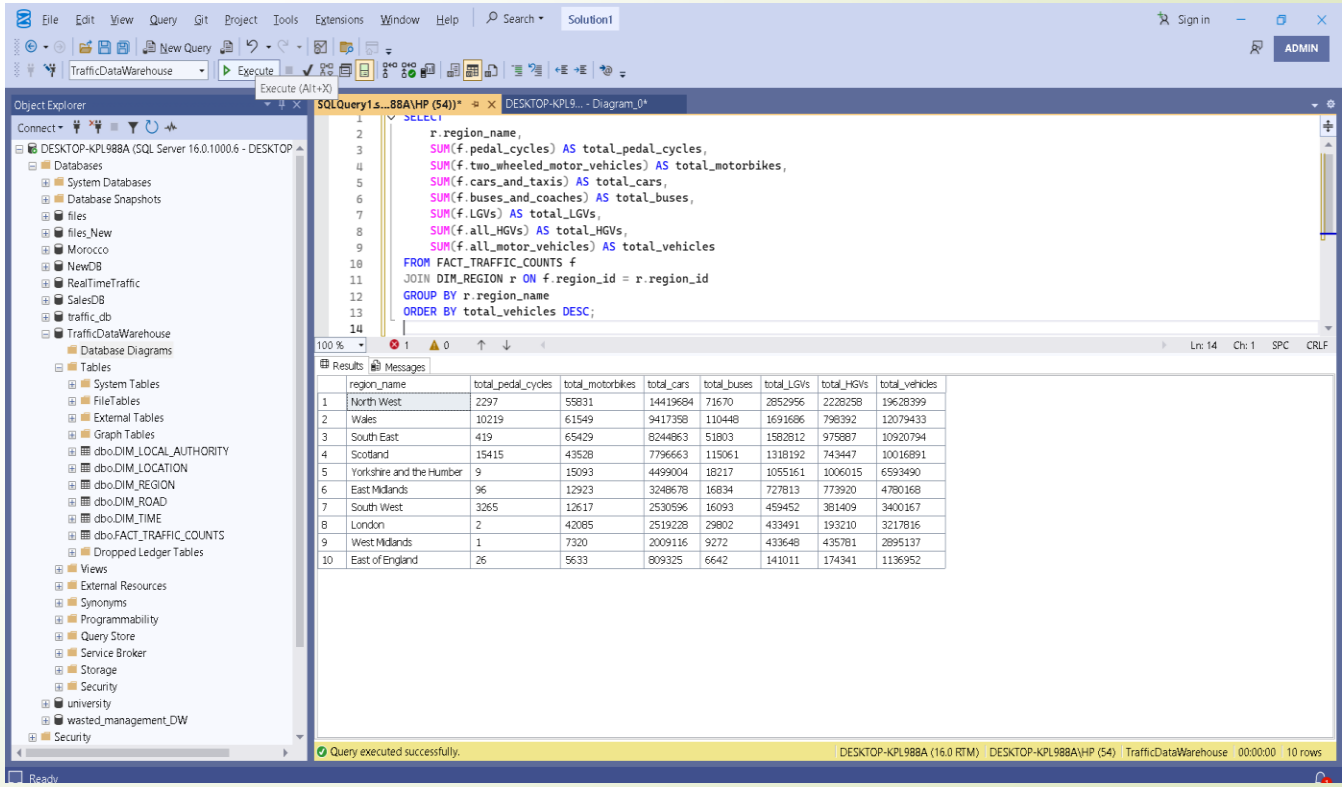
```
# ✅ 4. إنشاء جدول الوقائع (Fact)
fact_traffic_counts = df[['count_point_id', 'region_id', 'local_authority_id', 'road_name', 'year',
                          'pedal_cycles', 'two_wheeled_motor_vehicles', 'cars_and_taxis',
                          'buses_and_coaches', 'LGVs', 'all_HGVs', 'all_motor_vehicles']].copy()

# إضافة مفتاح أساسي (fact_id)
fact_traffic_counts.insert(0, 'fact_id', range(1, len(fact_traffic_counts) + 1))

# ✅ 5. حفظ الجداول النظيفة
dim_region.to_csv("DIM_REGION.csv", index=False)
dim_local_authority.to_csv("DIM_LOCAL_AUTHORITY.csv", index=False)
dim_location.to_csv("DIM_LOCATION.csv", index=False)
dim_road.to_csv("DIM_ROAD.csv", index=False)
dim_time.to_csv("DIM_TIME.csv", index=False)
fact_traffic_counts.to_csv("FACT_TRAFFIC_COUNTS.csv", index=False)

print("✅ تم إنشاء الملفات بنجاح:")
print("- DIM_REGION.csv")
print("- DIM_LOCAL_AUTHORITY.csv")
print("- DIM_LOCATION.csv")
print("- DIM_ROAD.csv")
print("- DIM_TIME.csv")
print("- FACT_TRAFFIC_COUNTS.csv")
```

Data Analysis:

🚗1️⃣Traffic by Region

تحليل إجمالي عدد المركبات لكل إقليم (Region):

```
USE TrafficDataWarehouse;
GO

SELECT
    r.region_name,
    SUM(f.pedal_cycles) AS total_pedal_cycles,
    SUM(f.two_wheeled_motor_vehicles) AS total_motorbikes,
    SUM(f.cars_and_taxis) AS total_cars,
    SUM(f.buses_and_coaches) AS total_buses,
    SUM(f.LGVs) AS total_LGVs,
    SUM(f.all_HGVs) AS total_HGVs,
    SUM(f.all_motor_vehicles) AS total_vehicles
FROM FACT_TRAFFIC_COUNTS f
JOIN DIM_REGION r ON f.region_id = r.region_id
GROUP BY r.region_name
ORDER BY total_vehicles DESC;
```

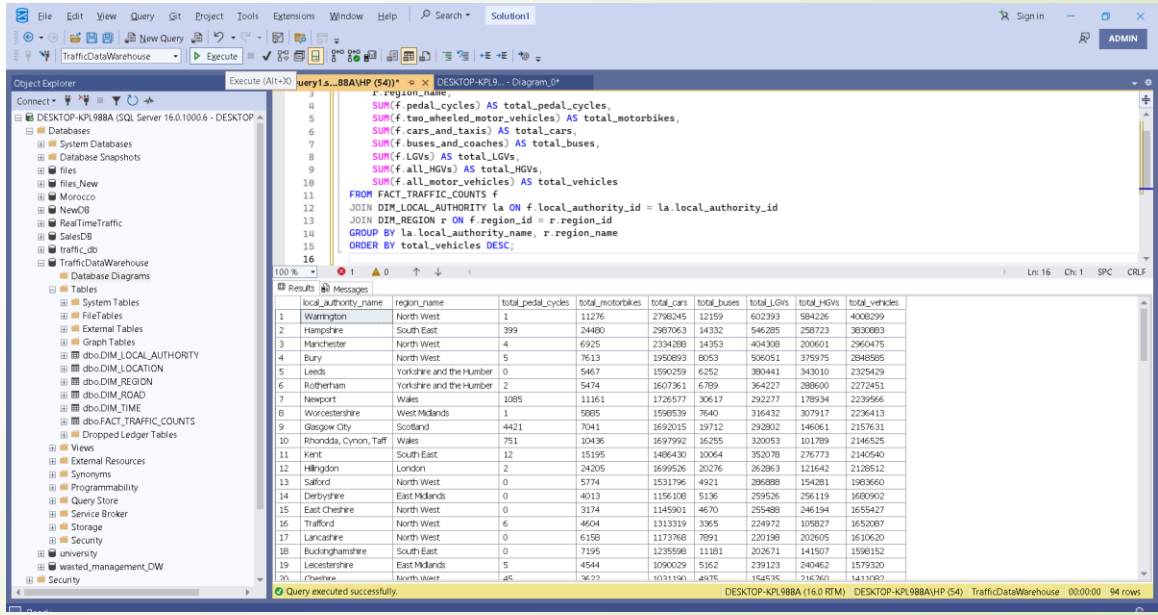


	region_name	total_pedal_cycles	total_motorbikes	total_cars	total_buses	total_LGVs	total_HGVs	total_vehicles
1	North West	2297	55831	14419684	71670	2852956	2228258	19628399
2	Wales	10219	61549	9417358	110448	1691686	798392	12079433
3	South East	419	65429	8244863	51803	1582812	975887	10920794
4	Scotland	15415	43528	7796663	115061	1318192	743447	10016891
5	Yorkshire and the Humber	9	15093	4499004	18217	1055161	1006015	6593490
6	East Midlands	96	12923	3248678	16834	727813	773920	4780168
7	South West	3265	12617	2530596	16093	459452	381409	3400167
8	London	2	42085	2519228	29802	433491	193210	3217816
9	West Midlands	1	7320	2009116	9272	433648	435781	2895137
10	East of England	26	5633	809325	6642	141011	174341	1136952

🏙️2️⃣Traffic by Local Authority

تحليل إجمالي المركبات لكل سلطة محلية (Local Authority):

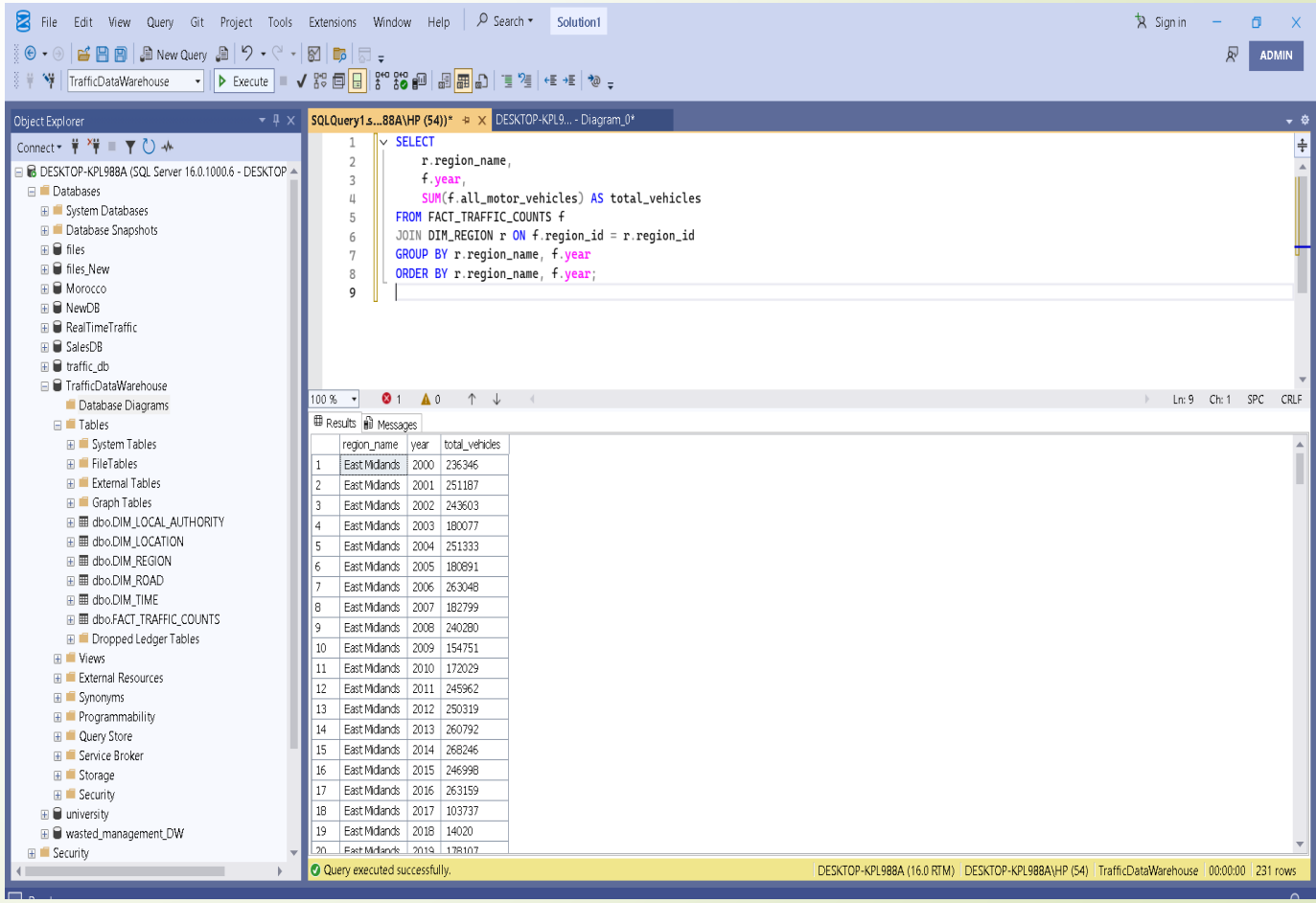
```
SELECT
    la.local_authority_name,
    r.region_name,
    SUM(f.pedal_cycles) AS total_pedal_cycles,
    SUM(f.two_wheeled_motor_vehicles) AS total_motorbikes,
    SUM(f.cars_and_taxis) AS total_cars,
    SUM(f.buses_and_coaches) AS total_buses,
    SUM(f.LGVs) AS total_LGVs,
    SUM(f.all_HGVs) AS total_HGVs,
    SUM(f.all_motor_vehicles) AS total_vehicles
FROM FACT_TRAFFIC_COUNTS f
JOIN DIM_LOCAL_AUTHORITY la ON f.local_authority_id = la.local_authority_id
JOIN DIM_REGION r ON f.region_id = r.region_id
GROUP BY la.local_authority_name, r.region_name
ORDER BY total_vehicles DESC;
```



📈📅Traffic Trend by Region (Over Time)

إذا تريد التحليل عبر السنوات:

```
SELECT
   r.region_name,
   f.year,
   SUM(f.all_motor_vehicles) AS total_vehicles
FROM FACT_TRAFFIC_COUNTS f
JOIN DIM_REGION r ON f.region_id = r.region_id
GROUP BY r.region_name, f.year
ORDER BY r.region_name, f.year;
```



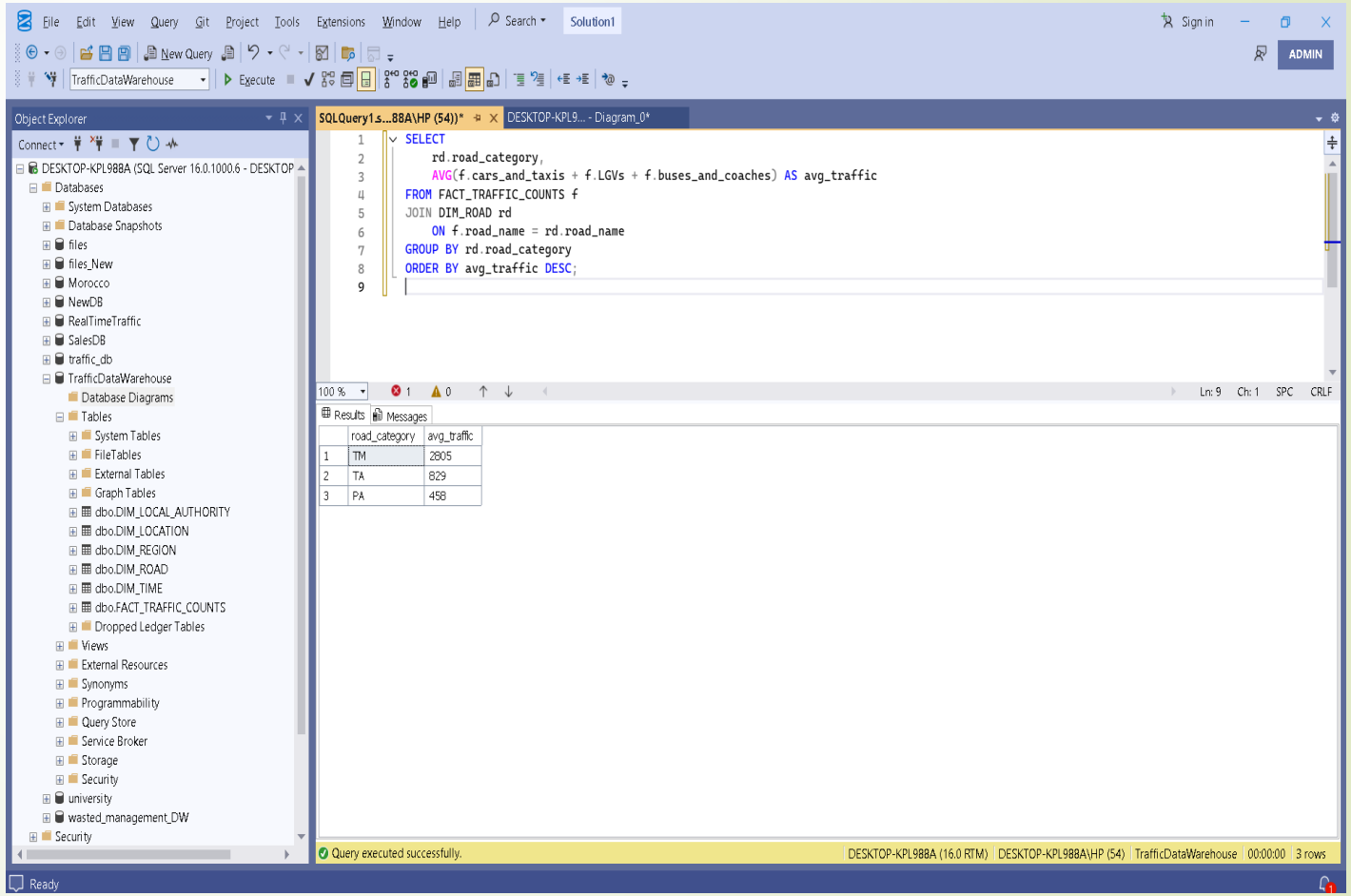
يمكنك بعدها رسمها بيانياً في Power BI أو Python Matplotlib أو Plotly.

الهيكل لديك حاليًا

- جدول الوقائع: FACT_TRAFFIC_COUNTS
- جدول الطرق: DIM_ROAD
- العلاقة:
- FACT_TRAFFIC_COUNTS.road_name → DIM_ROAD.road_name

✓ الاستعلام الصحيح في SQL Server

```
SELECT
   rd.road_category,
   AVG(f.cars_and_taxis + f.LGVs + f.buses_and_coaches) AS avg_traffic
FROM FACT_TRAFFIC_COUNTS f
JOIN DIM_ROAD rd
   ON f.road_name = rd.road_name
GROUP BY rd.road_category
ORDER BY avg_traffic DESC;
```



توضيح الفكرة

- هذا التحليل يحسب متوسط حجم الحركة المرورية لكل فئة طريق.
- تم جمع ثلاث فئات من المركبات:

cars_and_taxis + LGVs + buses_and_coaches

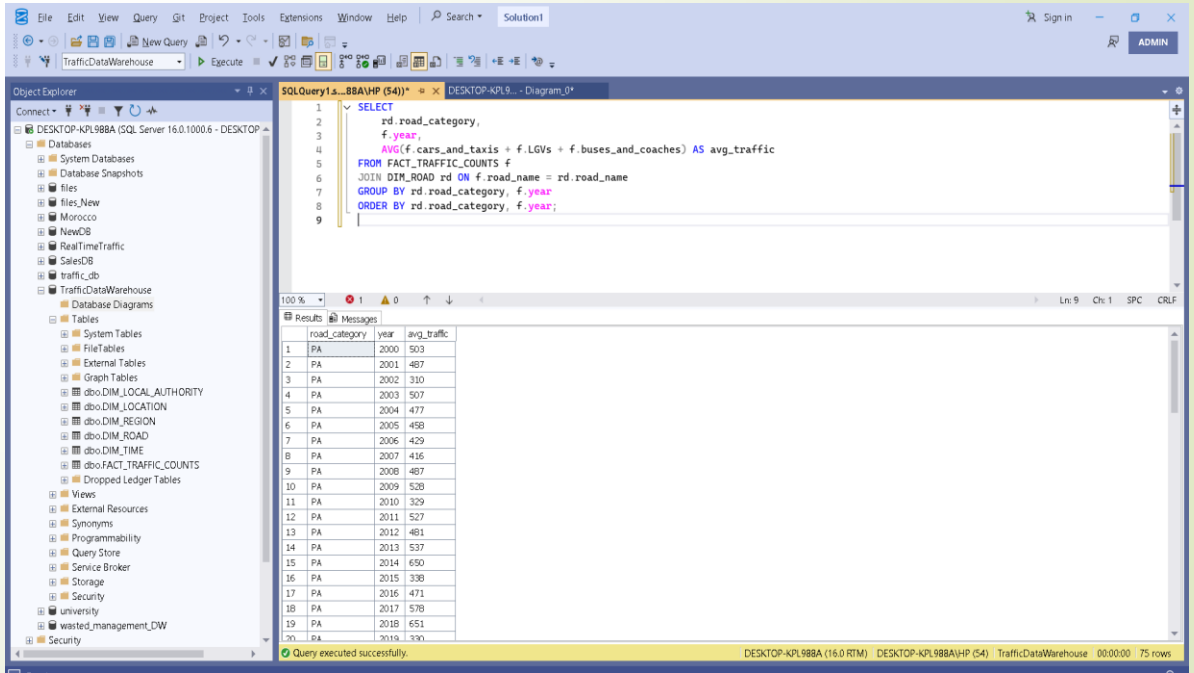
- يمكن تعديل المعادلة لتشمل فئات إضافية مثل HGVsأو pedal_cycles.

⚙️تحليل أكثر تقدمًا

يمكنك توسيع التحليل بإضافة سنة أو منطقة لفهم التغير الزمني أو الجغرافي:

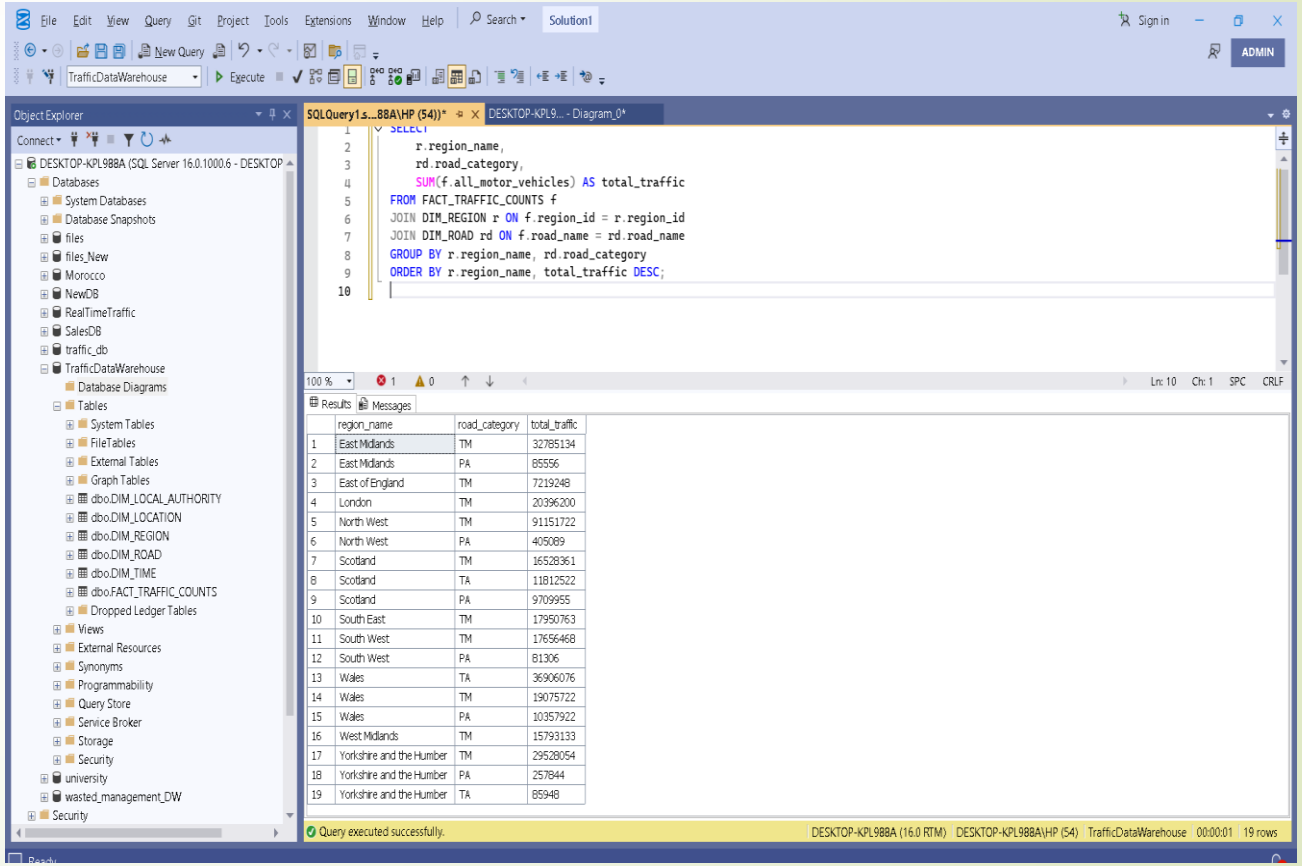
🗓️تحليل فئة الطريق عبر السنوات:

```
SELECT
    rd.road_category,
    f.year,
    AVG(f.cars_and_taxis + f.LGVs + f.buses_and_coaches) AS avg_traffic
FROM FACT_TRAFFIC_COUNTS f
JOIN DIM_ROAD rd ON f.road_name = rd.road_name
GROUP BY rd.road_category, f.year
ORDER BY rd.road_category, f.year;
```



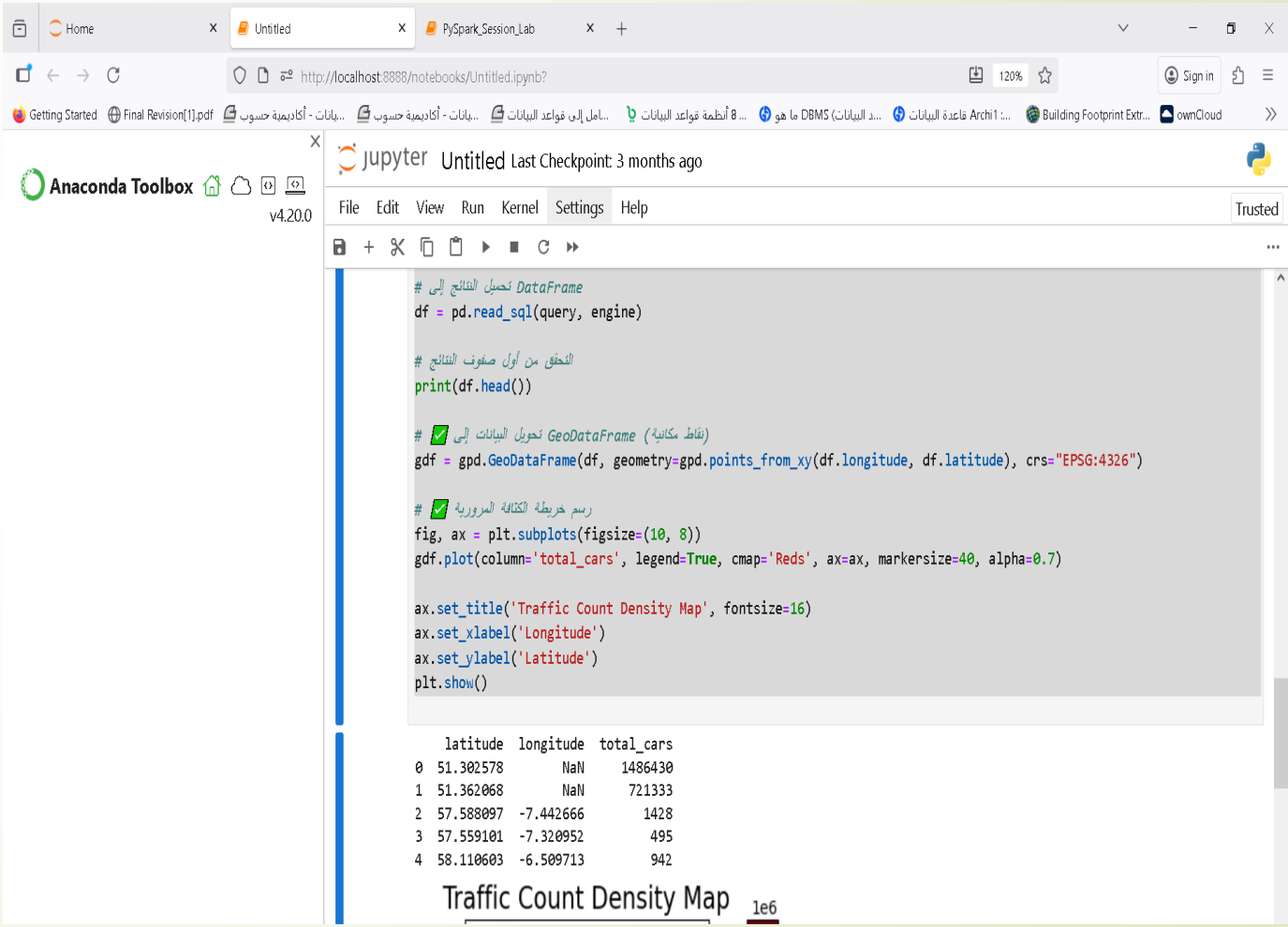
تحليل فئة الطريق حسب الإقليم:

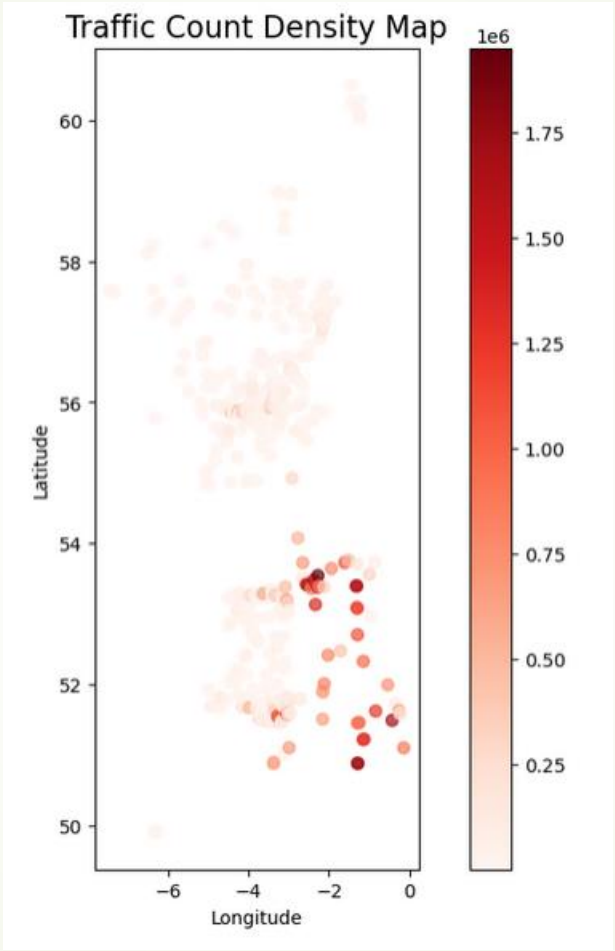
```
SELECT
    r.region_name,
    rd.road_category,
    SUM(f.all_motor_vehicles) AS total_traffic
FROM FACT_TRAFFIC_COUNTS f
JOIN DIM_REGION r ON f.region_id = r.region_id
JOIN DIM_ROAD rd ON f.road_name = rd.road_name
GROUP BY r.region_name, rd.road_category
ORDER BY r.region_name, total_traffic DESC;
```



Plot:

```
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import create_engine

# إعداد معلومات الاتصال
server = 'localhost'       # اسم السيرفر المحلي
database = 'traffic_db'    # اسم قاعدة البيانات

# ✅ الاتصال بدون مصادقة (No username/password)
# إذا كان السيرفر لا يتطلب تسجيل دخول (يُسمح بالوصول العام)
conn_str = f"mssql+pyodbc://{server}/{database}?driver=ODBC+Driver+17+for+SQL+Server"

# إنشاء المحرك (engine)
engine = create_engine(conn_str)

# ✅ الاستعلام: جمع عدد السيارات حسب الإحداثيات
query = """
SELECT
    cp.latitude,
    cp.longitude,
    SUM(tc.cars_and_taxis) AS total_cars
FROM
    [TrafficDataWarehouse].[dbo].[DIM_LOCATION] cp
JOIN
    [TrafficDataWarehouse].[dbo].[FACT_TRAFFIC_COUNTS] tc ON cp.count_point_id =
tc.count_point_id
GROUP BY
    cp.latitude, cp.longitude
```




Traffic Count Density Map
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
60
58
56
54
52
50
Latitude
−6
−4
−2
0
Longitude

Data Analysis – Date time:

🎯الهدف من التحليل

تحليل بيانات الحركة المرورية Traffic Countsحسب الوقت، لفهم الأنماط الزمنية مثل:

- توزيع عدد المركبات حسب السنة / الشهر / اليوم / الساعة
- تحديد ساعات الذروة Peak Hours
- حساب إجمالي وعدد متوسط المركبات في أوقات مختلفة
- معرفة الاتجاهات الزمنية (يومي، أسبوعي، سنوي)

---

1. تحويل التاريخ والساعة إلى حقل تاريخ/وقت موحد

إذا كان لديك عمود count_dateتاريخ فقط وعمود hourرقم الساعة، يمكنك تكوين عمود جديد يمثل الوقت الكامل:

```
ALTER TABLE FACT_TRAFFIC_COUNTS1
ADD count_datetime AS
    DATEADD(HOUR, hour, CAST(count_date AS DATETIME));
```

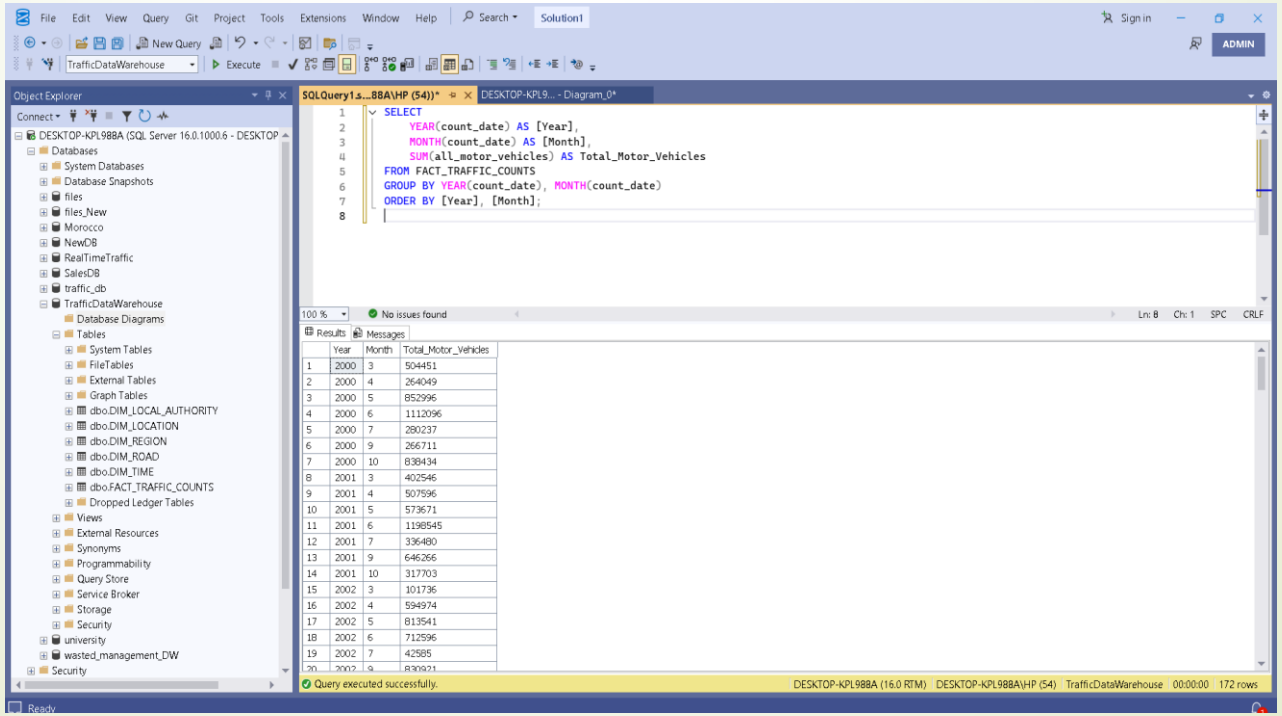
الآن أصبح لديك عمود count_datetimeيمثل التاريخ والساعة معًا، يمكن استخدامه للتحليل الزمني.

---

2. 📊تحليل إجمالي المركبات حسب السنة والشهر

```
SELECT
    YEAR(count_date) AS [Year],
    MONTH(count_date) AS [Month],
    SUM(all_motor_vehicles) AS Total_Motor_Vehicles
FROM FACT_TRAFFIC_COUNTS
GROUP BY YEAR(count_date), MONTH(count_date)
ORDER BY [Year], [Month];
```

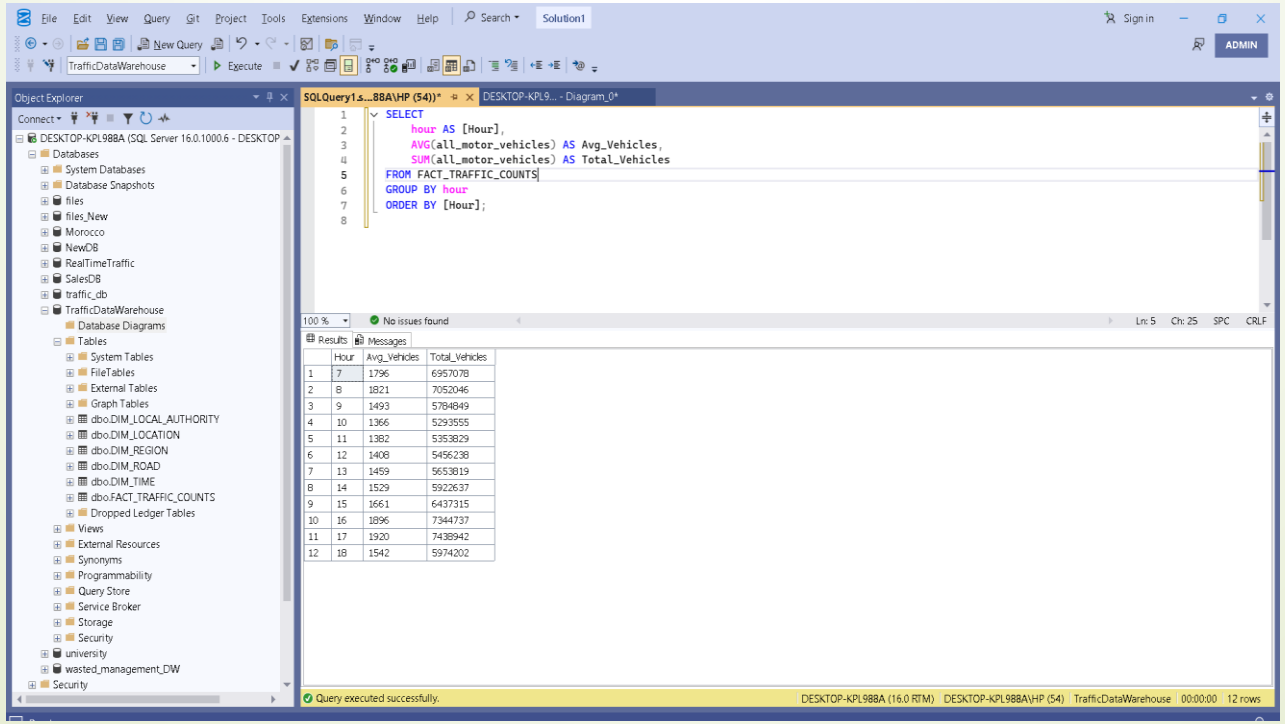
📈هذا التحليل يوضح الاتجاه العام لحركة المرور على مدار السنة.



3. 🕐تحليل عدد المركبات حسب الساعة Peak Hour Analysis

```
SELECT
    hour AS [Hour],
    AVG(all_motor_vehicles) AS Avg_Vehicles,
    SUM(all_motor_vehicles) AS Total_Vehicles
FROM FACT_TRAFFIC_COUNTS
GROUP BY hour
ORDER BY [Hour];
```

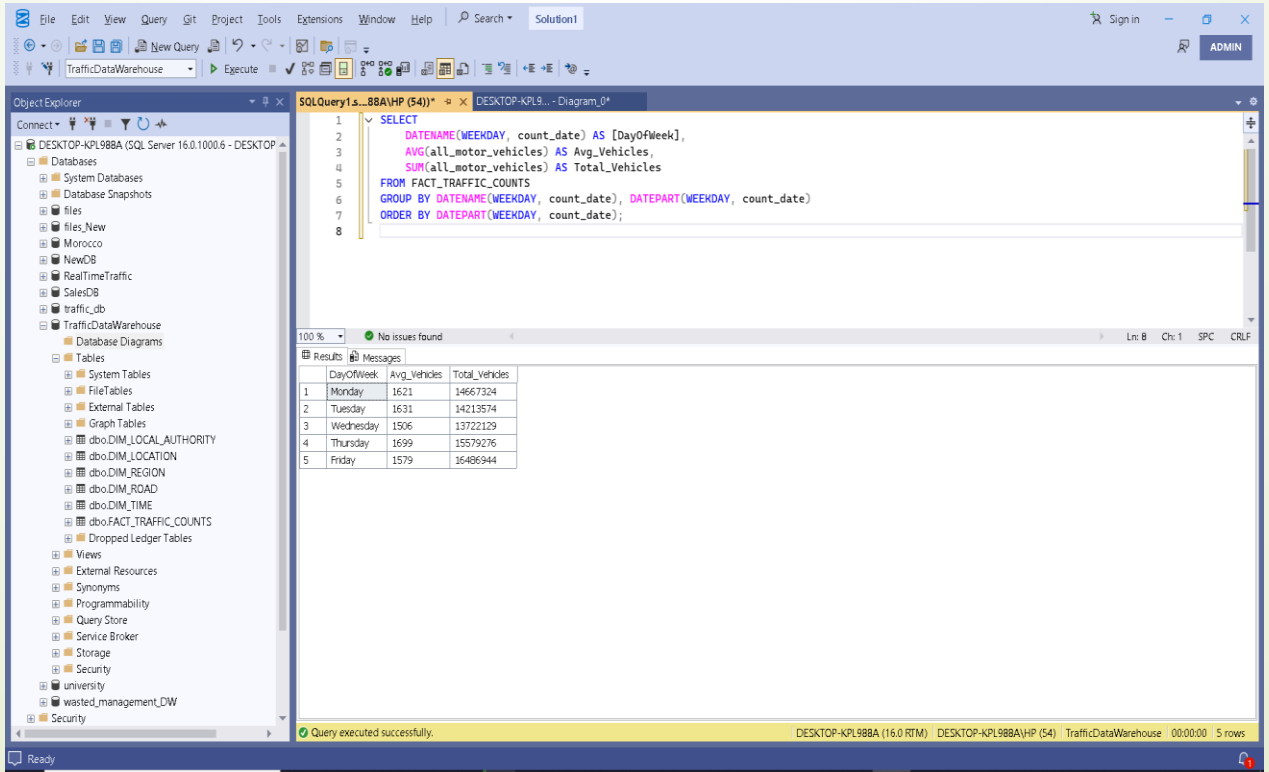
📊هذا الاستعلام يُظهر متوسط وعدد المركبات في كل ساعة لتحديد أوقات الذروة.



4. 🧮تحليل حسب اليوم في الأسبوع

```
SELECT
    DATENAME(WEEKDAY, count_date) AS [DayOfWeek],
    AVG(all_motor_vehicles) AS Avg_Vehicles,
    SUM(all_motor_vehicles) AS Total_Vehicles
FROM FACT_TRAFFIC_COUNTS
GROUP BY DATENAME(WEEKDAY, count_date),
DATEPART(WEEKDAY, count_date)
ORDER BY DATEPART(WEEKDAY, count_date);
```

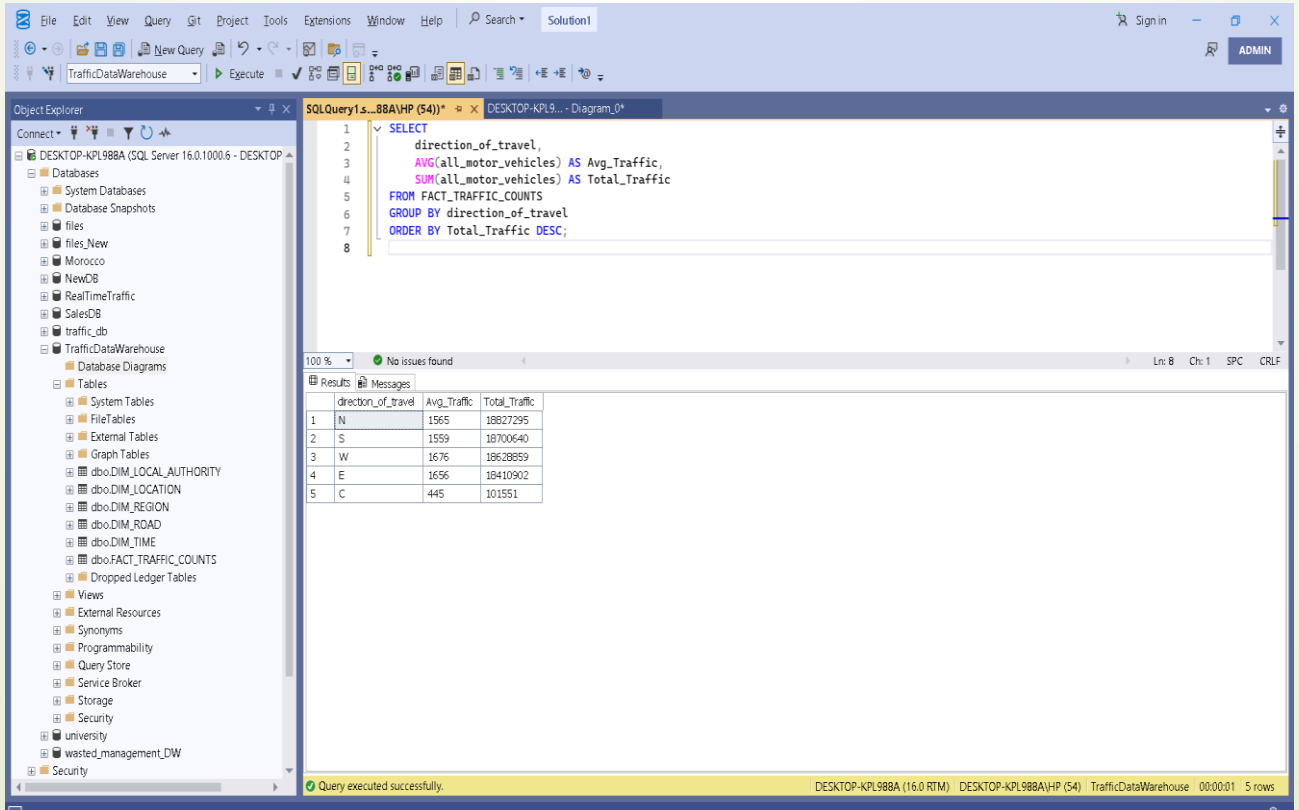
🗓يساعد على معرفة أكثر الأيام ازدحامًا في الأسبوع.



5. 🚗تحليل الاتجاه (اتجاه السير)

```
SELECT
    direction_of_travel,
    AVG(all_motor_vehicles) AS Avg_Traffic,
    SUM(all_motor_vehicles) AS Total_Traffic
FROM FACT_TRAFFIC_COUNTS
GROUP BY direction_of_travel
ORDER BY Total_Traffic DESC;
```

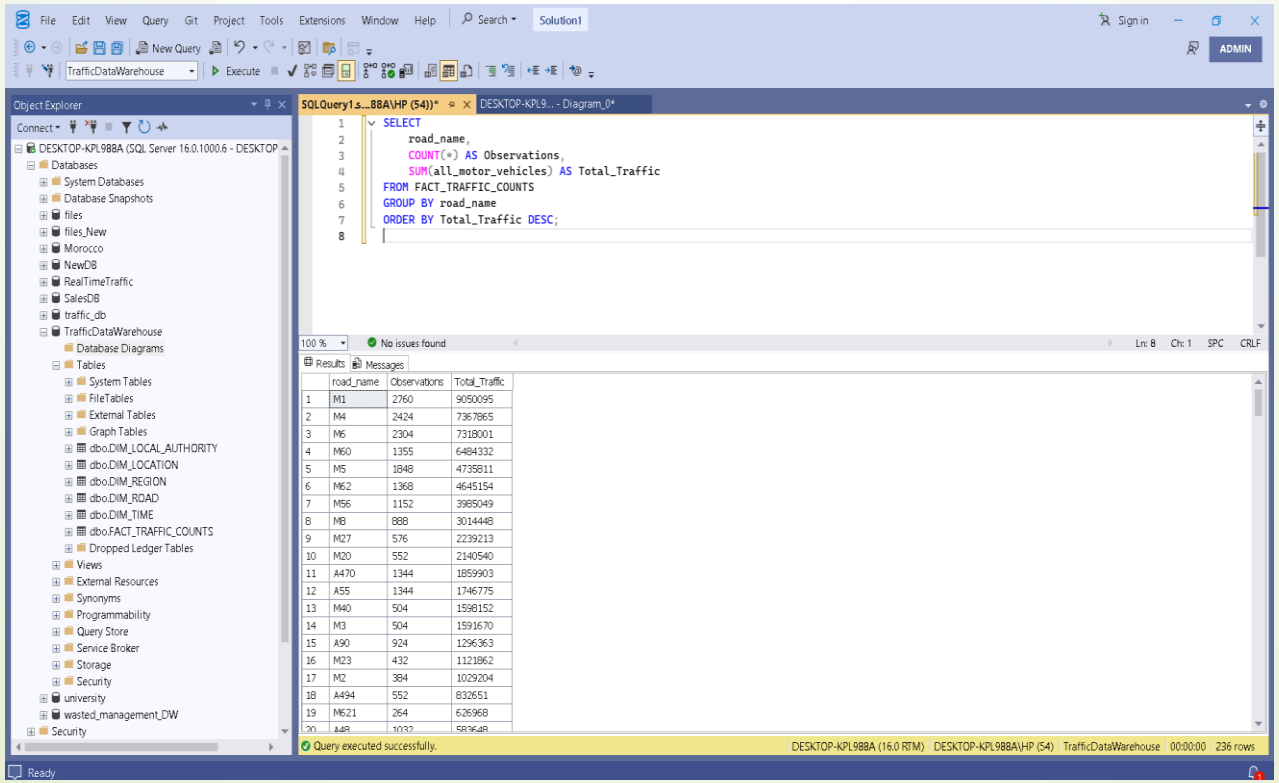
مفيد لتحديد الاتجاهات ذات الكثافة المرورية الأعلى.



6. تحليل الطريق Road Name

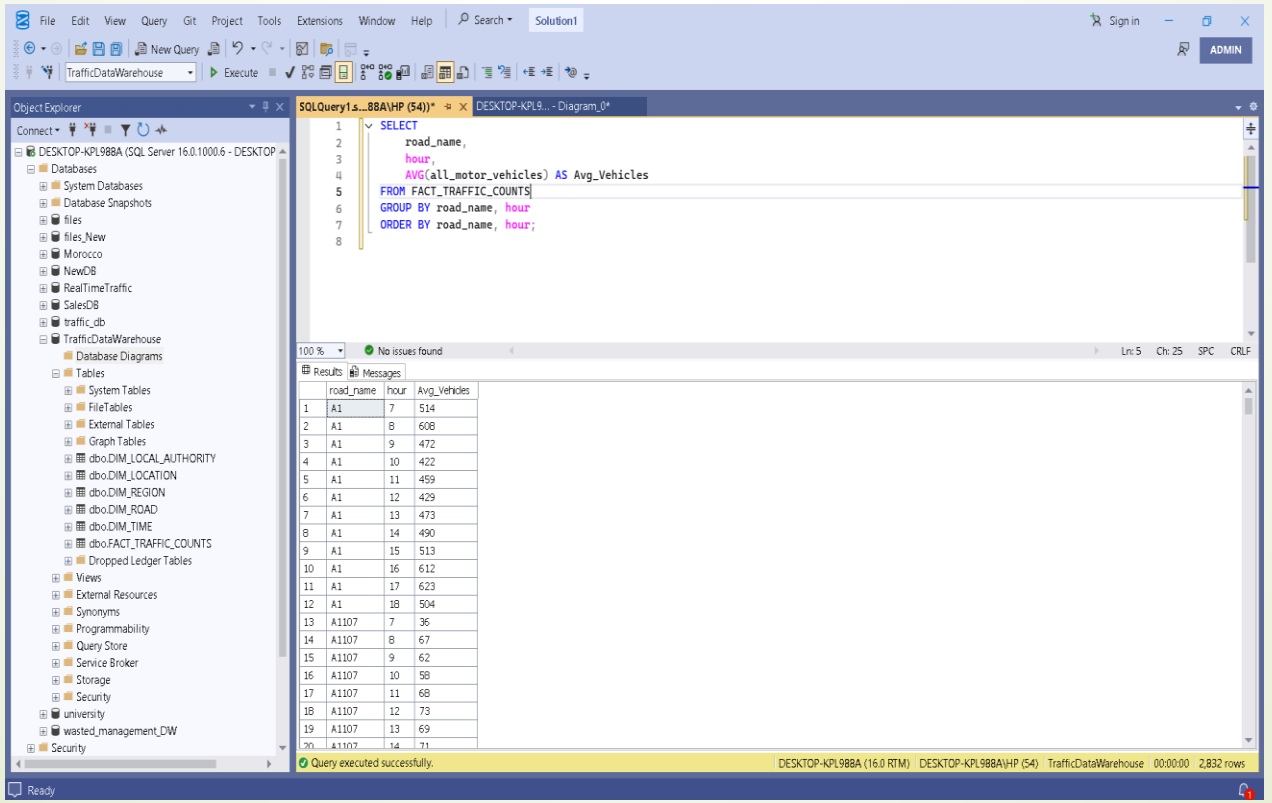
```
SELECT
    road_name,
    COUNT(*) AS Observations,
    SUM(all_motor_vehicles) AS Total_Traffic
FROM FACT_TRAFFIC_COUNTS
GROUP BY road_name
ORDER BY Total_Traffic DESC;
```

هذا التحليل يظهر أكثر الطرق ازدحامًا بالمرور.



7. 📈دمج التحليل الزمني والمكاني
مثلاً — حركة المرور لكل طريق حسب الساعة:

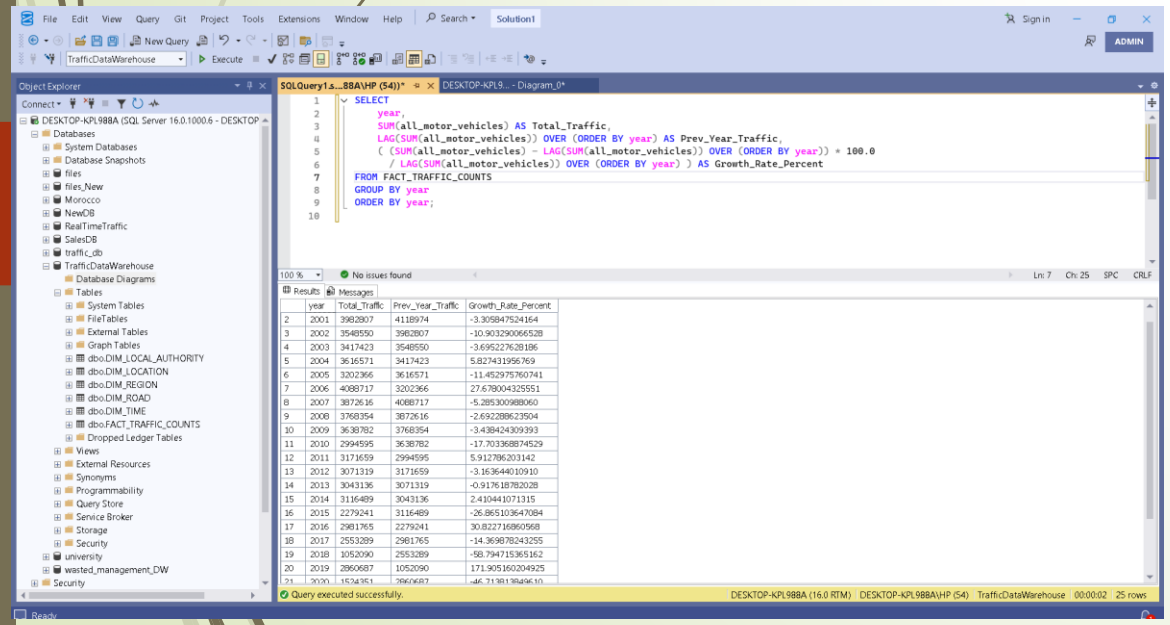
```
SELECT
    road_name,
    hour,
    AVG(all_motor_vehicles) AS Avg_Vehicles
FROM FACT_TRAFFIC_COUNTS
GROUP BY road_name, hour
ORDER BY road_name, hour;
```



8. تحليل معدل النمو السنوي

```
SELECT
    year,
    SUM(all_motor_vehicles) AS Total_Traffic,
    LAG(SUM(all_motor_vehicles)) OVER (ORDER BY
year) AS Prev_Year_Traffic,
    ( (SUM(all_motor_vehicles) -
LAG(SUM(all_motor_vehicles)) OVER (ORDER BY
year)) * 100.0
     / LAG(SUM(all_motor_vehicles)) OVER (ORDER BY
year) ) AS Growth_Rate_Percent
FROM FACT_TRAFFIC_COUNTS
GROUP BY year
ORDER BY year;
```

📊يوضح معدل النمو أو التغير السنوي في عدد المركبات.



The beneficiaries of the "Real Time Traffic Project" include:

1. **Commuters and Drivers** – who can access up-to-date traffic information, avoid congestion, and reduce travel time.
2. **Public Transportation Authorities** – who can optimize routes, schedules, and fleet management using real-time data.
3. **Emergency Services** – such as police, ambulance, and fire departments, enabling faster response times and route planning.
4. **Urban Planners and Decision Makers** – who can use traffic data to improve infrastructure planning and traffic flow management.
5. **Logistics and Delivery Companies** – who can enhance operational efficiency and reduce fuel consumption through route optimization.
6. **Environmental Agencies** – benefiting from reduced vehicle emissions due to improved traffic flow.
7. **Citizens and Local Communities** – who experience safer roads, less congestion, and better air quality.


Thank
you! :)