Excel and SQL Analysis - Notes

Welcome to my comprehensive notes on Excel and SQL analysis techniques. This collection of notes serves as a documentation of my data analysis journey, showcasing my proficiency in Excel and SQL.

About the Notes

In this document, I've meticulously documented the process and insights derived from my data analysis journey using Excel and SQL. With a focus on the Cylistic Case Study provided under the Google Data Analytics Professional certificate, these notes highlight my approach to:

- **Excel Analysis**: Learn how I harnessed the power of Excel to clean, transform, and visualize the data. From basic descriptive statistics to advanced functions, these notes cover it all.
- **SQL Techniques**: Dive into SQL commands used for data querying, transformations, and analysis. Witness how I employed SQL to unveil hidden patterns and trends within the dataset.

Highlights

- Detailed step-by-step instructions for replicating the analyses conducted in the project.
- Insights into data cleaning, data transformation, and visualization techniques using Excel.
- SQL commands and strategies for extracting valuable information from the dataset.

How to Use

Whether you're a beginner or an experienced data enthusiast, these notes provide valuable insights and practical knowledge to elevate your analytical skills. Feel free to explore the notes to enhance your understanding of Excel and SQL analysis.

Your feedback is invaluable as we continue to explore the realm of data analysis and extract insights that drive informed decisions together!

Dhupar

Abhay Dhupar

(Author)

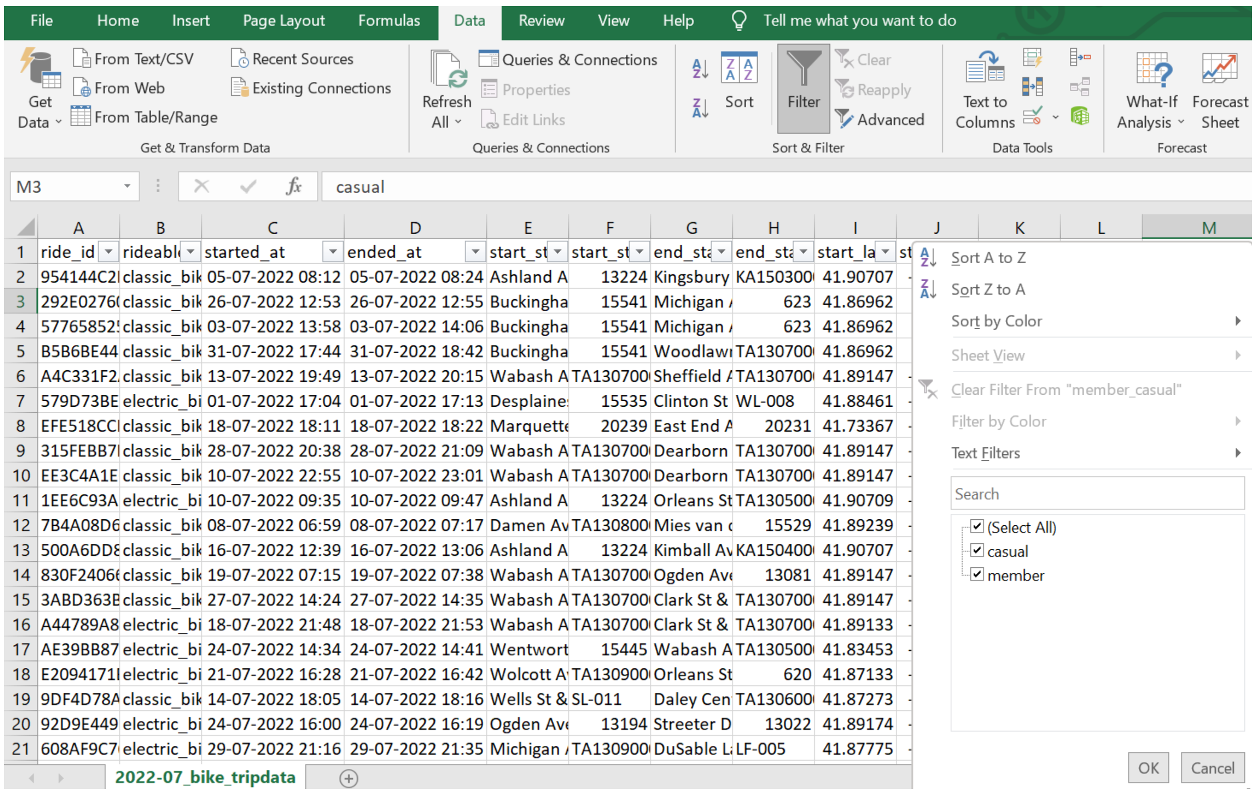
Excel

First we'll save the csv files to workbook type so that we don't have data loss.

Now analyzing consolidated data for year is difficult on excel as there are too many rows (more than 5 lakhs). So we'll analyze month wise.

Working on month data

First we'll use the filter option in the data tab to check if there's any wrong value in any field.



M3 casual

[✓] (Select All)
[✓] casual
[✓] member

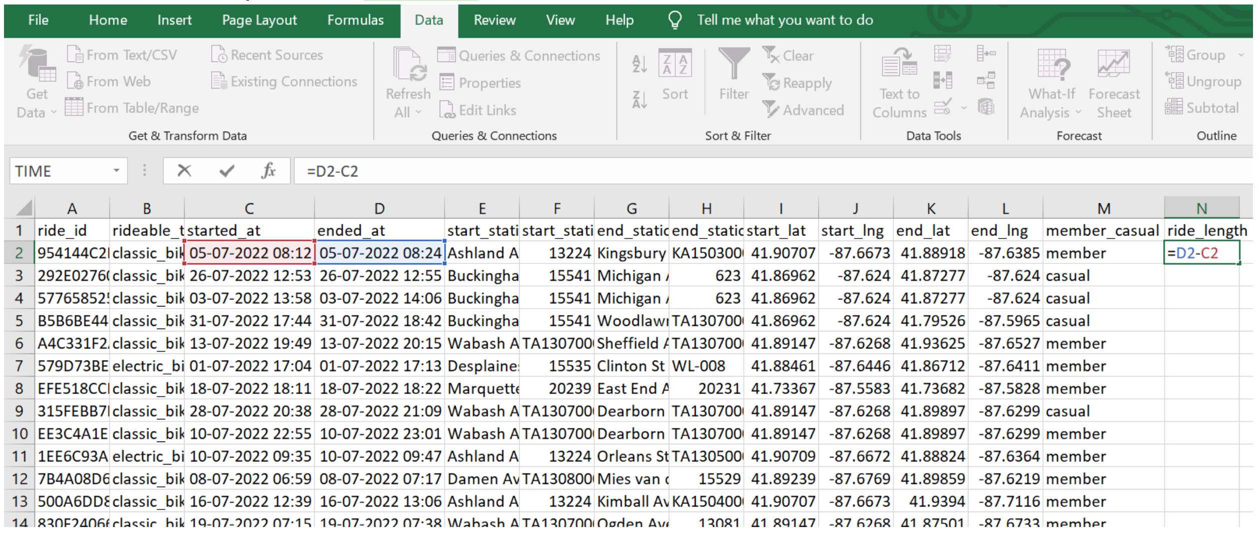
	A	B	C	D	E	F	G	H	I	J
1	ride_id	rideabl	started_at	ended_at	start_st	start_st	end_sta	end_sta	start_la	st
2	954144C2	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	KA150300	41.90707	-
3	292E0276	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan /	623	41.86962	
4	57765852	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan /	623	41.86962	
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlawı	TA130700	41.86962	
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield /	TA130700	41.89147	-
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine:	15535	Clinton St	WL-008	41.88461	-
8	EFE518CC	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End A	20231	41.73367	-
9	315FEBB7	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	TA130700	41.89147	-
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	TA130700	41.89147	-
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	TA130500	41.90709	-
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van (	15529	41.89239	-
13	500A6DD8	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	KA150400	41.90707	-
14	830F2406	classic_bik	19-07-2022 07:15	19-07-2022 07:38	Wabash A	TA130700	Ogden Ave	13081	41.89147	-
15	3ABD363B	classic_bik	27-07-2022 14:24	27-07-2022 14:35	Wabash A	TA130700	Clark St &	TA130700	41.89147	-
16	A44789A8	electric_bi	18-07-2022 21:48	18-07-2022 21:53	Wabash A	TA130700	Clark St &	TA130700	41.89133	-
17	AE39BB87	electric_bi	24-07-2022 14:34	24-07-2022 14:41	Wentwort	15445	Wabash A	TA130500	41.83453	-
18	E2094171	electric_bi	21-07-2022 16:28	21-07-2022 16:42	Wolcott A	TA130900	Orleans St	620	41.87133	-
19	9DF4D78A	classic_bik	14-07-2022 18:05	14-07-2022 18:16	Wells St &	SL-011	Daley Cen	TA130600	41.87273	-
20	92D9E449	electric_bi	24-07-2022 16:00	24-07-2022 16:19	Ogden Ave	13194	Streeter D	13022	41.89174	-
21	608AF9C7	electric_bi	29-07-2022 21:16	29-07-2022 21:35	Michigan /	TA130900	DuSable L	LF-005	41.87775	-

This way, we can check for any missing values and if there are any, we can remove it.

Add more columns for descriptive analysis

Here we will add two more columns **ride_length** and **day_of_week** to make our descriptive analysis more easy.

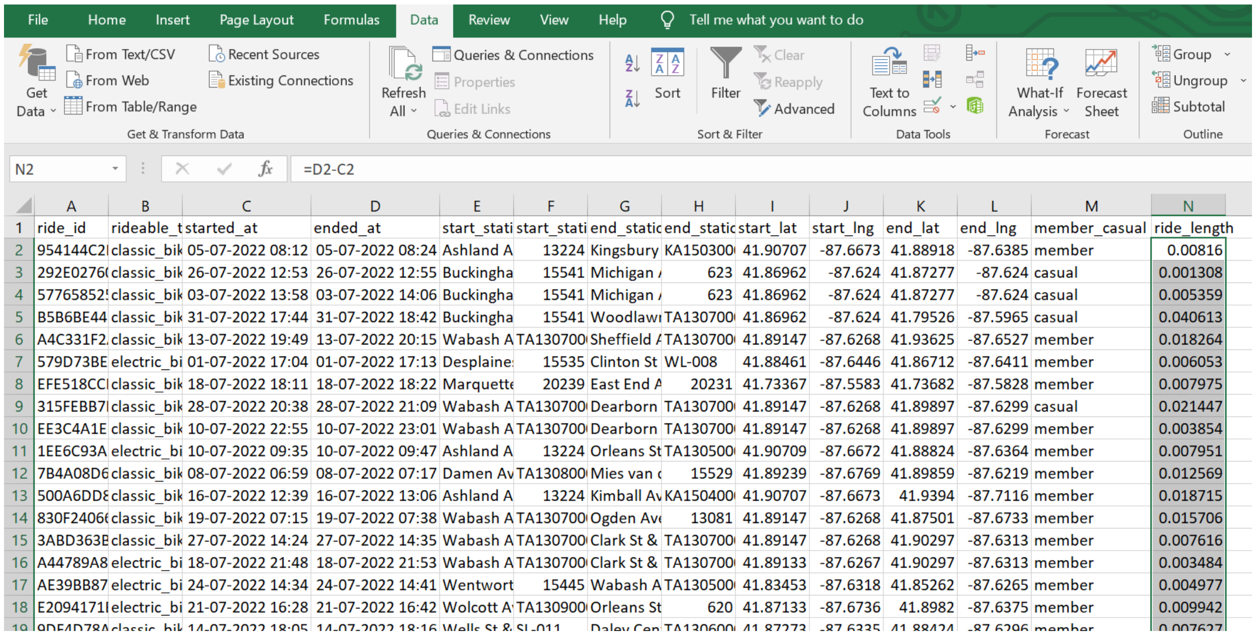
To calculate **ride_length,** we'll find the difference between columns **ended_at** and **started_at.** So we'll use the expression '=D2-C2' to find the difference.



TIME | =D2-C2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ride_id	rideable_t	started_at	ended_at	start_stati	start_stati	end_static	end_static	start_lat	start_lng	end_lat	end_lng	member_casual	ride_length
2	954144C2	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	KA150300	41.90707	-87.6673	41.88918	-87.6385	member	=D2-C2
3	292E0276	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan	623	41.86962	-87.624	41.87277	-87.624	casual	
4	57765852	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan	623	41.86962	-87.624	41.87277	-87.624	casual	
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlaw	TA130700	41.86962	-87.624	41.79526	-87.5965	casual	
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield	TA130700	41.89147	-87.6268	41.93625	-87.6527	member	
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine	15535	Clinton St	WL-008	41.88461	-87.6446	41.86712	-87.6411	member	
8	EFE518CC	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End	20231	41.73367	-87.5583	41.73682	-87.5828	member	
9	315FEBB7	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	casual	
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	member	
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	TA130500	41.90709	-87.6672	41.88824	-87.6364	member	
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van	15529	41.89239	-87.6769	41.89859	-87.6219	member	
13	500A6DD	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	KA150400	41.90707	-87.6673	41.9394	-87.7116	member	

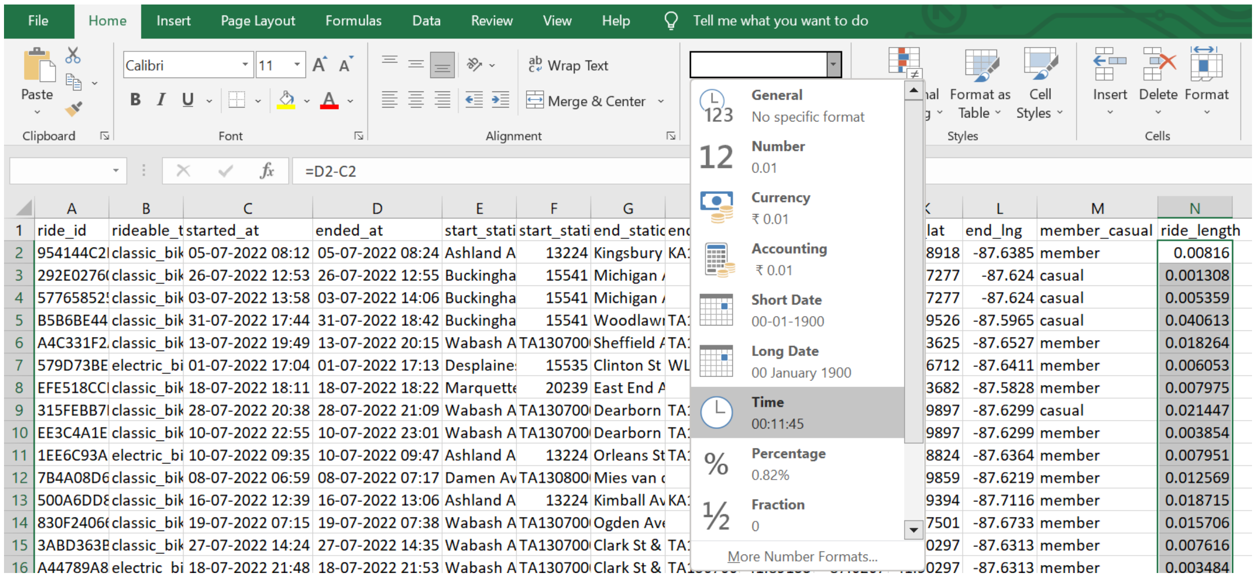
After than we can press enter and use the fill handle to apply for all rows. Or we can also select the rows column to fill and press CTRL+D to make the fill handle on column operation.



N2 | =D2-C2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ride_id	rideable_t	started_at	ended_at	start_stati	start_stati	end_static	end_static	start_lat	start_lng	end_lat	end_lng	member_casual	ride_length
2	954144C2	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	KA150300	41.90707	-87.6673	41.88918	-87.6385	member	0.00816
3	292E0276	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan	623	41.86962	-87.624	41.87277	-87.624	casual	0.001308
4	57765852	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan	623	41.86962	-87.624	41.87277	-87.624	casual	0.005359
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlaw	TA130700	41.86962	-87.624	41.79526	-87.5965	casual	0.040613
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield	TA130700	41.89147	-87.6268	41.93625	-87.6527	member	0.018264
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine	15535	Clinton St	WL-008	41.88461	-87.6446	41.86712	-87.6411	member	0.006053
8	EFE518CC	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End	20231	41.73367	-87.5583	41.73682	-87.5828	member	0.007975
9	315FEBB7	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	casual	0.021447
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	member	0.003854
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	TA130500	41.90709	-87.6672	41.88824	-87.6364	member	0.007951
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van	15529	41.89239	-87.6769	41.89859	-87.6219	member	0.012569
13	500A6DD	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	KA150400	41.90707	-87.6673	41.9394	-87.7116	member	0.018715
14	830F2406	classic_bik	19-07-2022 07:15	19-07-2022 07:38	Wabash A	TA130700	Ogden Ave	13081	41.89147	-87.6268	41.87501	-87.6733	member	0.015706
15	3ABD363B	classic_bik	27-07-2022 14:24	27-07-2022 14:35	Wabash A	TA130700	Clark St &	TA130700	41.89147	-87.6268	41.90297	-87.6313	member	0.007616
16	A44789A8	electric_bi	18-07-2022 21:48	18-07-2022 21:53	Wabash A	TA130700	Clark St &	TA130700	41.89133	-87.6267	41.90297	-87.6313	member	0.003484
17	AE39BB87	electric_bi	24-07-2022 14:34	24-07-2022 14:41	Wentwort	15445	Wabash A	TA130500	41.83453	-87.6318	41.85262	-87.6265	member	0.004977
18	E2094171	electric_bi	21-07-2022 16:28	21-07-2022 16:42	Wolcott A	TA130900	Orleans St	620	41.87133	-87.6736	41.8982	-87.6375	member	0.009942

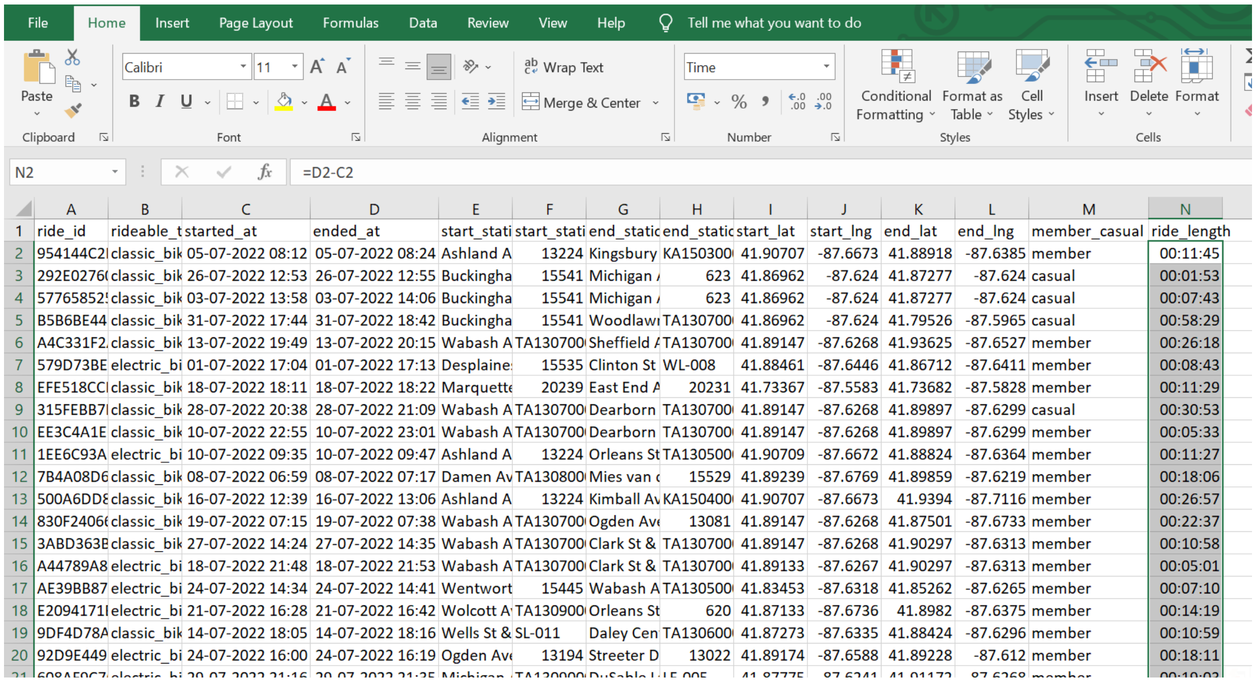
Now the data type we have for ride length is in decimal. So let’s convert it to time. For this first we’ll select the data and then go to Home>Number>Dropdown to select time.



=D2-C2

	A	B	C	D	E	F	G	L	M	N
1	ride_id	rideable_t	started_at	ended_at	start_stati	start_stati	end_static	end_lng	member_casual	ride_length
2	954144C2I	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	-87.6385	member	0.00816
3	292E0276(	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan /	-87.624	casual	0.001308
4	57765852!	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan /	-87.624	casual	0.005359
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlawı	-87.5965	casual	0.040613
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield /	-87.6527	member	0.018264
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine:	15535	Clinton St	-87.6411	member	0.006053
8	EFE518CCI	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End A	-87.5828	member	0.007975
9	315FEBB7I	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	-87.6299	casual	0.021447
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	-87.6299	member	0.003854
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	-87.6364	member	0.007951
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van (	-87.6219	member	0.012569
13	500A6DD8	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	-87.7116	member	0.018715
14	830F2406(	classic_bik	19-07-2022 07:15	19-07-2022 07:38	Wabash A	TA130700	Ogden Ave	-87.6733	member	0.015706
15	3ABD363E	classic_bik	27-07-2022 14:24	27-07-2022 14:35	Wabash A	TA130700	Clark St &	-87.6313	member	0.007616
16	A44789A8	electric_bi	18-07-2022 21:48	18-07-2022 21:53	Wabash A	TA130700	Clark St &	-87.6313	member	0.003484

This will update the data type of all values to be shown in time format (i.e. hh:mm:ss)

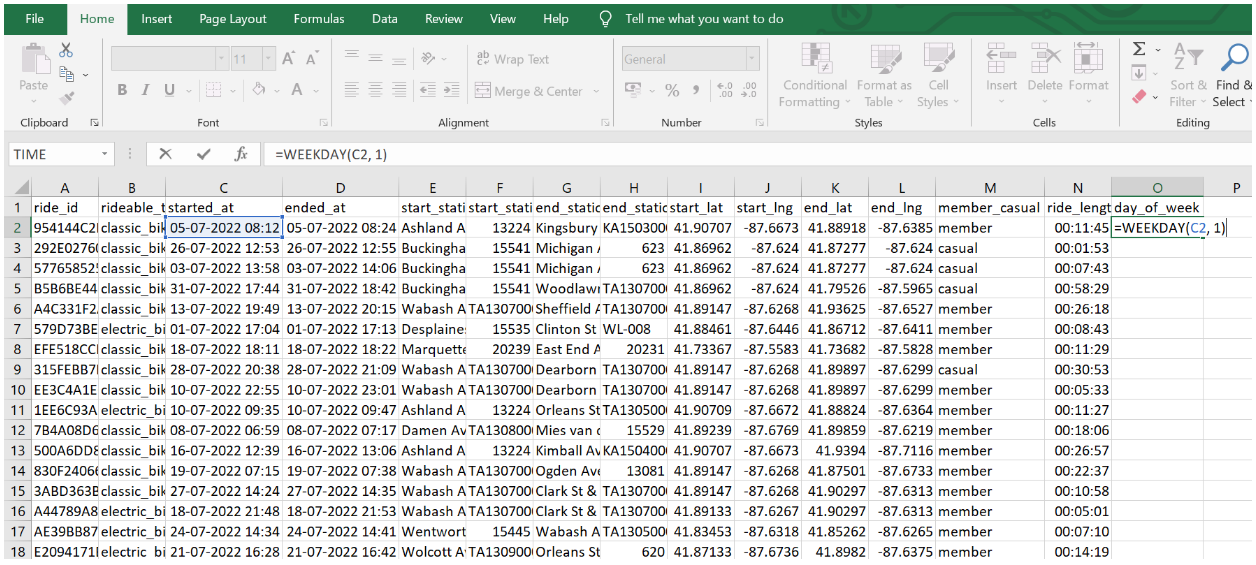


N2 =D2-C2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ride_id	rideable_t	started_at	ended_at	start_stati	start_stati	end_static	end_static	start_lat	start_lng	end_lat	end_lng	member_casual	ride_length
2	954144C2I	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	KA150300	41.90707	-87.6673	41.88918	-87.6385	member	00:11:45
3	292E0276(	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan /	623	41.86962	-87.624	41.87277	-87.624	casual	00:01:53
4	57765852!	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan /	623	41.86962	-87.624	41.87277	-87.624	casual	00:07:43
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlawı	TA130700	41.86962	-87.624	41.79526	-87.5965	casual	00:58:29
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield /	TA130700	41.89147	-87.6268	41.93625	-87.6527	member	00:26:18
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine:	15535	Clinton St	WL-008	41.88461	-87.6446	41.86712	-87.6411	member	00:08:43
8	EFE518CCI	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End A	20231	41.73367	-87.5583	41.73682	-87.5828	member	00:11:29
9	315FEBB7I	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	casual	00:30:53
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	member	00:05:33
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	TA130500	41.90709	-87.6672	41.88824	-87.6364	member	00:11:27
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van (	15529	41.89239	-87.6769	41.89859	-87.6219	member	00:18:06
13	500A6DD8	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	KA150400	41.90707	-87.6673	41.9394	-87.7116	member	00:26:57
14	830F2406(	classic_bik	19-07-2022 07:15	19-07-2022 07:38	Wabash A	TA130700	Ogden Ave	13081	41.89147	-87.6268	41.87501	-87.6733	member	00:22:37
15	3ABD363E	classic_bik	27-07-2022 14:24	27-07-2022 14:35	Wabash A	TA130700	Clark St &	TA130700	41.89147	-87.6268	41.90297	-87.6313	member	00:10:58
16	A44789A8	electric_bi	18-07-2022 21:48	18-07-2022 21:53	Wabash A	TA130700	Clark St &	TA130700	41.89133	-87.6267	41.90297	-87.6313	member	00:05:01
17	AE39BB87	electric_bi	24-07-2022 14:34	24-07-2022 14:41	Wentwort	15445	Wabash A	TA130500	41.83453	-87.6318	41.85262	-87.6265	member	00:07:10
18	E2094171I	electric_bi	21-07-2022 16:28	21-07-2022 16:42	Wolcott A	TA130900	Orleans St	620	41.87133	-87.6736	41.8982	-87.6375	member	00:14:19
19	9DF4D78A	classic_bik	14-07-2022 18:05	14-07-2022 18:16	Wells St &	SL-011	Daley Cen	TA130600	41.87273	-87.6335	41.88424	-87.6296	member	00:10:59
20	92D9E449	electric_bi	24-07-2022 16:00	24-07-2022 16:19	Ogden Ave	13194	Streeter D	13022	41.89174	-87.6588	41.89228	-87.612	member	00:18:11

Now let’s add the **day_of_week** column to our data.

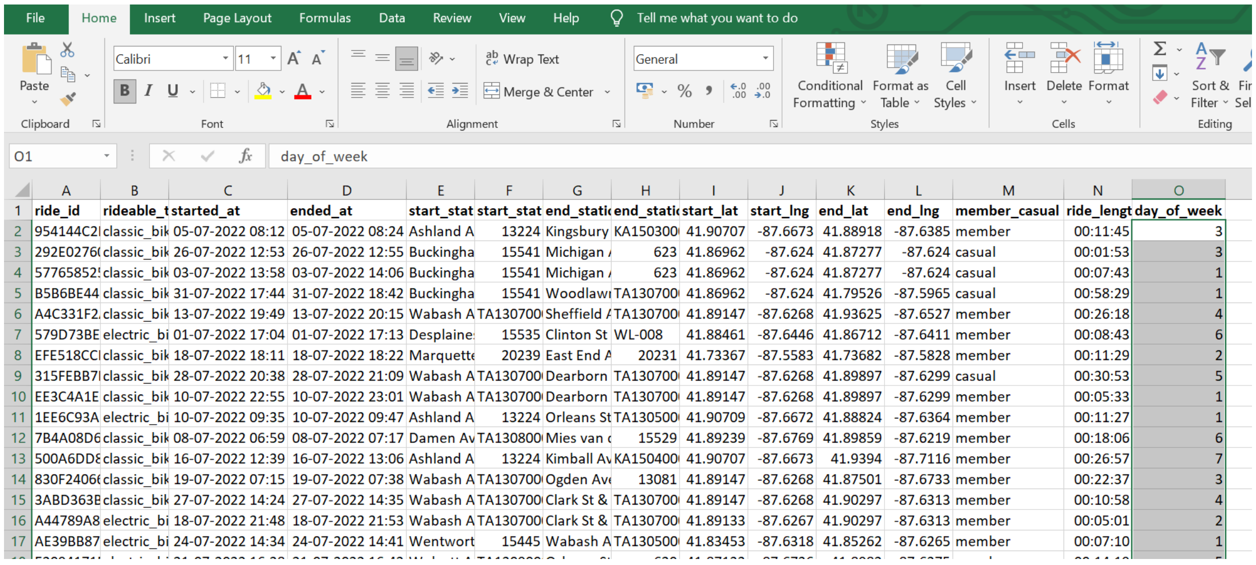
We’ll use the WEEKDAY() to do so. There are two argument we’ll give. FIrst will be the date from which we’ll find our weekday. In this case, we’ll use Column C, which is **started_at**
And then we’ll give the second argument as **1,** to indicate that the series as (1-Sunday, 2-Monday .. and so on)



TIME | =WEEKDAY(C2, 1)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ride_id	rideable_t	started_at	ended_at	start_stati	start_stati	end_static	end_static	start_lat	start_lng	end_lat	end_lng	member_casual	ride_lengt	day_of_week
2	954144C2	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	KA150300	41.90707	-87.6673	41.88918	-87.6385	member	00:11:45	=WEEKDAY(C2, 1)
3	292E0276	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan /	623	41.86962	-87.624	41.87277	-87.624	casual	00:01:53	
4	57765852	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan /	623	41.86962	-87.624	41.87277	-87.624	casual	00:07:43	
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlawi	TA130700	41.86962	-87.624	41.79526	-87.5965	casual	00:58:29	
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield /	TA130700	41.89147	-87.6268	41.93625	-87.6527	member	00:26:18	
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine	15535	Clinton St	WL-008	41.88461	-87.6446	41.86712	-87.6411	member	00:08:43	
8	EFE518CC	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End A	20231	41.73367	-87.5583	41.73682	-87.5828	member	00:11:29	
9	315FEBB7	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	casual	00:30:53	
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	member	00:05:33	
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	TA130500	41.90709	-87.6672	41.88824	-87.6364	member	00:11:27	
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van c	15529	41.89239	-87.6769	41.89859	-87.6219	member	00:18:06	
13	500A6DD8	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	KA150400	41.90707	-87.6673	41.9394	-87.7116	member	00:26:57	
14	830F2406	classic_bik	19-07-2022 07:15	19-07-2022 07:38	Wabash A	TA130700	Ogden Ave	13081	41.89147	-87.6268	41.87501	-87.6733	member	00:22:37	
15	3ABD363B	classic_bik	27-07-2022 14:24	27-07-2022 14:35	Wabash A	TA130700	Clark St &	TA130700	41.89147	-87.6268	41.90297	-87.6313	member	00:10:58	
16	A44789A8	electric_bi	18-07-2022 21:48	18-07-2022 21:53	Wabash A	TA130700	Clark St &	TA130700	41.89133	-87.6267	41.90297	-87.6313	member	00:05:01	
17	AE39BB87	electric_bi	24-07-2022 14:34	24-07-2022 14:41	Wentwort	15445	Wabash A	TA130500	41.83453	-87.6318	41.85262	-87.6265	member	00:07:10	
18	E2094171	electric_bi	21-07-2022 16:28	21-07-2022 16:42	Wolcott A	TA130900	Orleans St	620	41.87133	-87.6736	41.8982	-87.6375	member	00:14:19	

After that we’ll press enter and use the fill handle or select all and click CTRL + D to apply to all rows.

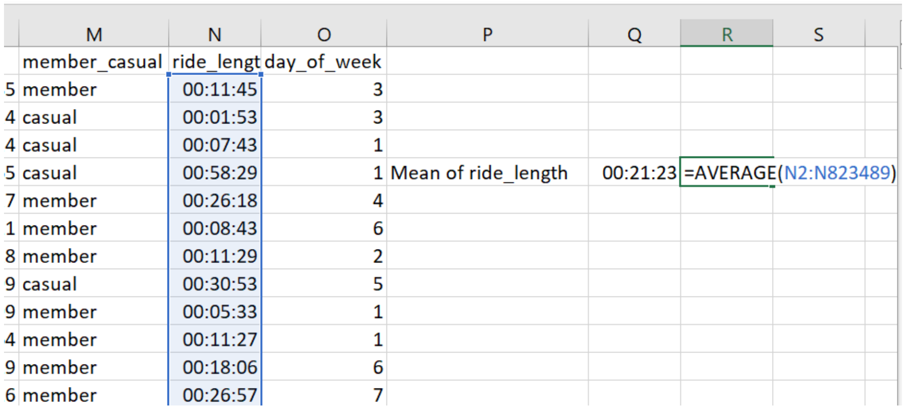


O1 | day_of_week

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ride_id	rideable_t	started_at	ended_at	start_stat	start_stat	end_static	end_static	start_lat	start_lng	end_lat	end_lng	member_casual	ride_lengt	day_of_week
2	954144C2	classic_bik	05-07-2022 08:12	05-07-2022 08:24	Ashland A	13224	Kingsbury	KA150300	41.90707	-87.6673	41.88918	-87.6385	member	00:11:45	3
3	292E0276	classic_bik	26-07-2022 12:53	26-07-2022 12:55	Buckingha	15541	Michigan /	623	41.86962	-87.624	41.87277	-87.624	casual	00:01:53	3
4	57765852	classic_bik	03-07-2022 13:58	03-07-2022 14:06	Buckingha	15541	Michigan /	623	41.86962	-87.624	41.87277	-87.624	casual	00:07:43	1
5	B5B6BE44	classic_bik	31-07-2022 17:44	31-07-2022 18:42	Buckingha	15541	Woodlawi	TA130700	41.86962	-87.624	41.79526	-87.5965	casual	00:58:29	1
6	A4C331F2.	classic_bik	13-07-2022 19:49	13-07-2022 20:15	Wabash A	TA130700	Sheffield /	TA130700	41.89147	-87.6268	41.93625	-87.6527	member	00:26:18	4
7	579D73BE	electric_bi	01-07-2022 17:04	01-07-2022 17:13	Desplaine	15535	Clinton St	WL-008	41.88461	-87.6446	41.86712	-87.6411	member	00:08:43	6
8	EFE518CC	classic_bik	18-07-2022 18:11	18-07-2022 18:22	Marquette	20239	East End A	20231	41.73367	-87.5583	41.73682	-87.5828	member	00:11:29	2
9	315FEBB7	classic_bik	28-07-2022 20:38	28-07-2022 21:09	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	casual	00:30:53	5
10	EE3C4A1E	classic_bik	10-07-2022 22:55	10-07-2022 23:01	Wabash A	TA130700	Dearborn	TA130700	41.89147	-87.6268	41.89897	-87.6299	member	00:05:33	1
11	1EE6C93A	electric_bi	10-07-2022 09:35	10-07-2022 09:47	Ashland A	13224	Orleans St	TA130500	41.90709	-87.6672	41.88824	-87.6364	member	00:11:27	1
12	7B4A08D6	classic_bik	08-07-2022 06:59	08-07-2022 07:17	Damen Av	TA130800	Mies van c	15529	41.89239	-87.6769	41.89859	-87.6219	member	00:18:06	6
13	500A6DD8	classic_bik	16-07-2022 12:39	16-07-2022 13:06	Ashland A	13224	Kimball Av	KA150400	41.90707	-87.6673	41.9394	-87.7116	member	00:26:57	7
14	830F2406	classic_bik	19-07-2022 07:15	19-07-2022 07:38	Wabash A	TA130700	Ogden Ave	13081	41.89147	-87.6268	41.87501	-87.6733	member	00:22:37	3
15	3ABD363B	classic_bik	27-07-2022 14:24	27-07-2022 14:35	Wabash A	TA130700	Clark St &	TA130700	41.89147	-87.6268	41.90297	-87.6313	member	00:10:58	4
16	A44789A8	electric_bi	18-07-2022 21:48	18-07-2022 21:53	Wabash A	TA130700	Clark St &	TA130700	41.89133	-87.6267	41.90297	-87.6313	member	00:05:01	2
17	AE39BB87	electric_bi	24-07-2022 14:34	24-07-2022 14:41	Wentwort	15445	Wabash A	TA130500	41.83453	-87.6318	41.85262	-87.6265	member	00:07:10	1

Doing some descriptive analysis

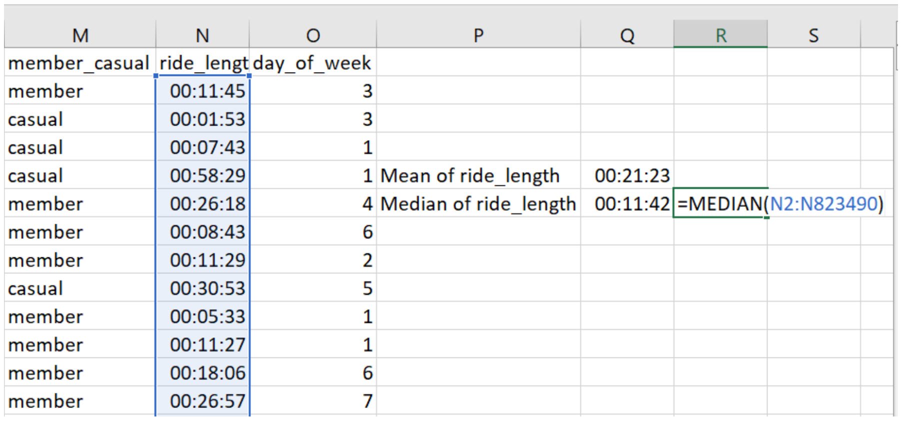
Now we can find **mean, median** of the **ride length** and **mode** of **day_of_week** to get some simple insights about our data.



	M	N	O	P	Q	R	S
	member_casual	ride_lengt	day_of_week				
5	member	00:11:45	3				
4	casual	00:01:53	3				
4	casual	00:07:43	1				
5	casual	00:58:29	1	Mean of ride_length	00:21:23	=AVERAGE(N2:N823489)	
7	member	00:26:18	4				
1	member	00:08:43	6				
8	member	00:11:29	2				
9	casual	00:30:53	5				
9	member	00:05:33	1				
4	member	00:11:27	1				
9	member	00:18:06	6				
6	member	00:26:57	7				

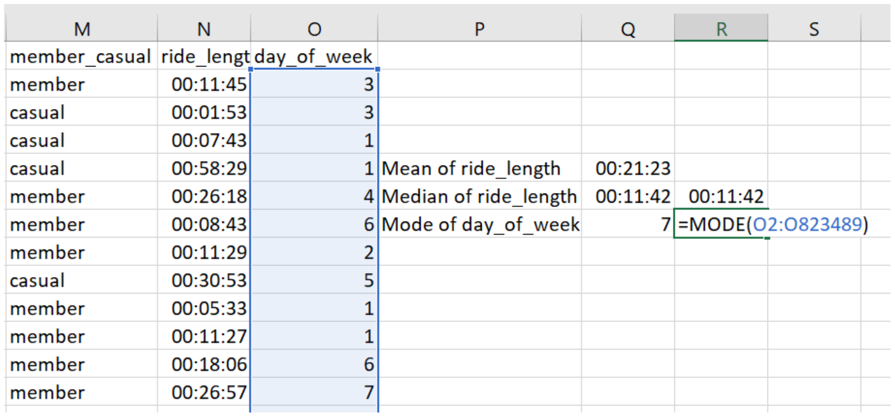
This way we'll find the average or mean.

To find median of ride_length



M	N	O	P	Q	R	S
member_casual	ride_lengt	day_of_week				
member	00:11:45	3				
casual	00:01:53	3				
casual	00:07:43	1				
casual	00:58:29	1	Mean of ride_length	00:21:23		
member	00:26:18	4	Median of ride_length	00:11:42	=MEDIAN(N2:N823490)	
member	00:08:43	6				
member	00:11:29	2				
casual	00:30:53	5				
member	00:05:33	1				
member	00:11:27	1				
member	00:18:06	6				
member	00:26:57	7				

To find mode of day_of_week

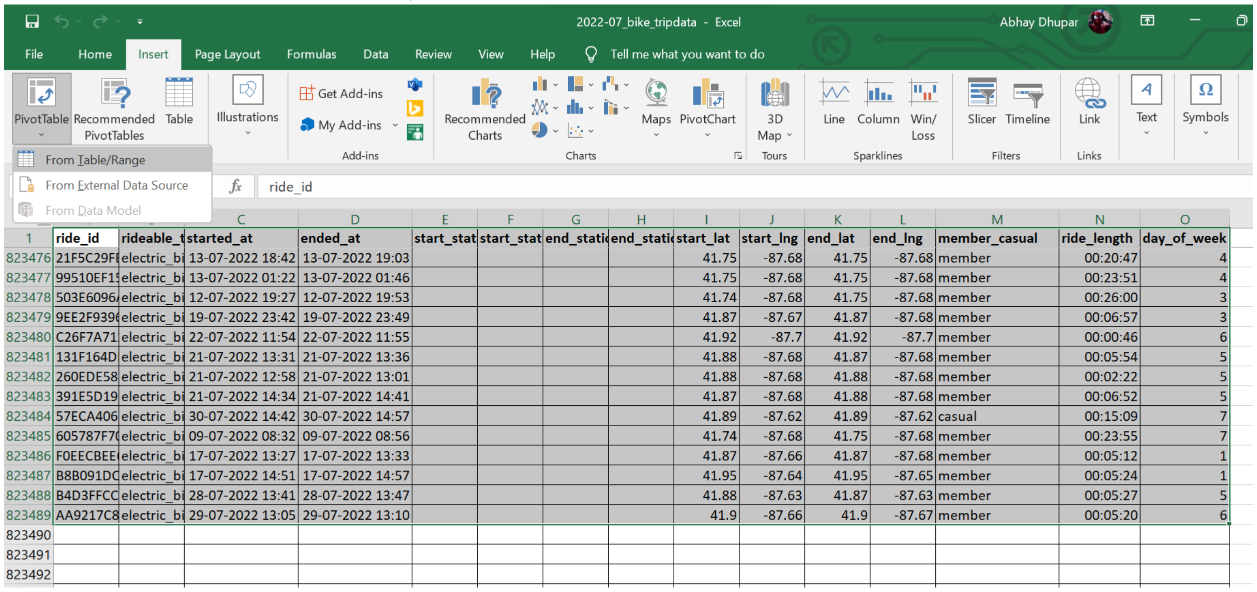


M	N	O	P	Q	R	S
member_casual	ride_lengt	day_of_week				
member	00:11:45	3				
casual	00:01:53	3				
casual	00:07:43	1				
casual	00:58:29	1	Mean of ride_length	00:21:23		
member	00:26:18	4	Median of ride_length	00:11:42	00:11:42	
member	00:08:43	6	Mode of day_of_week	7	=MODE(O2:O823489)	
member	00:11:29	2				
casual	00:30:53	5				
member	00:05:33	1				
member	00:11:27	1				
member	00:18:06	6				
member	00:26:57	7				

This is the frequency of **day_of_week** is 7.

Creating Pivot table and inserting visuals

Select all the rows and insert a pivot table on a blank sheet.

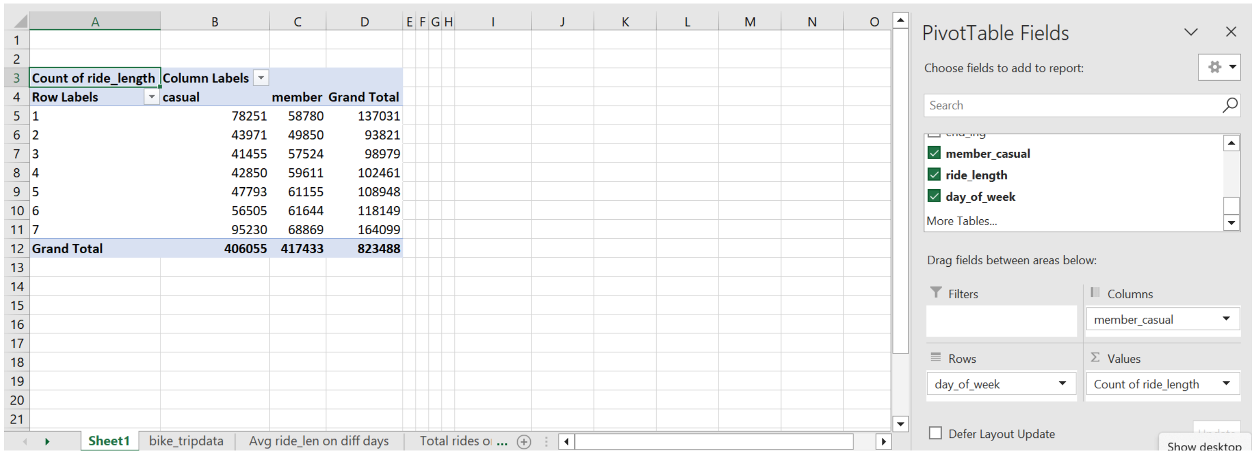


2022-07_bike_tripdata - Excel
Abhay Dhupar
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do
PivotTable Recommended PivotTables Table Illustrations Get Add-ins My Add-ins Add-ins Recommended Charts Charts Maps PivotChart 3D Map Tours Line Column Win/Loss Sparklines Slicer Timeline Filters Link Links Text Symbols
From Table/Range
From External Data Source
From Data Model

fx ride_id

	ride_id	rideable_t	started_at	ended_at	start_stat	start_stat	end_statio	end_statio	start_lat	start_lng	end_lat	end_lng	member_casual	ride_length	day_of_week
823476	21F5C29FE	electric_bi	13-07-2022 18:42	13-07-2022 19:03					41.75	-87.68	41.75	-87.68	member	00:20:47	4
823477	99510EF15	electric_bi	13-07-2022 01:22	13-07-2022 01:46					41.75	-87.68	41.75	-87.68	member	00:23:51	4
823478	503E6096/	electric_bi	12-07-2022 19:27	12-07-2022 19:53					41.74	-87.68	41.75	-87.68	member	00:26:00	3
823479	9EE2F939	electric_bi	19-07-2022 23:42	19-07-2022 23:49					41.87	-87.67	41.87	-87.68	member	00:06:57	3
823480	C26F7A71	electric_bi	22-07-2022 11:54	22-07-2022 11:55					41.92	-87.7	41.92	-87.7	member	00:00:46	6
823481	131F164D	electric_bi	21-07-2022 13:31	21-07-2022 13:36					41.88	-87.68	41.87	-87.68	member	00:05:54	5
823482	260EDE58	electric_bi	21-07-2022 12:58	21-07-2022 13:01					41.88	-87.68	41.88	-87.68	member	00:02:22	5
823483	391E5D19	electric_bi	21-07-2022 14:34	21-07-2022 14:41					41.87	-87.68	41.88	-87.68	member	00:06:52	5
823484	57ECA406	electric_bi	30-07-2022 14:42	30-07-2022 14:57					41.89	-87.62	41.89	-87.62	casual	00:15:09	7
823485	605787F7	electric_bi	09-07-2022 08:32	09-07-2022 08:56					41.74	-87.68	41.75	-87.68	member	00:23:55	7
823486	F0EECBEE	electric_bi	17-07-2022 13:27	17-07-2022 13:33					41.87	-87.66	41.87	-87.68	member	00:05:12	1
823487	B8B091DC	electric_bi	17-07-2022 14:51	17-07-2022 14:57					41.95	-87.64	41.95	-87.65	member	00:05:24	1
823488	B4D3FFCC	electric_bi	28-07-2022 13:41	28-07-2022 13:47					41.88	-87.63	41.87	-87.63	member	00:05:27	5
823489	AA9217C8	electric_bi	29-07-2022 13:05	29-07-2022 13:10					41.9	-87.66	41.9	-87.67	member	00:05:20	6
823490															
823491															
823492															

Now let's calculate the avg ride length on each weekday by each customer type.
For this, let's put day_of_week to column, and member_casual to rows. And ride length to values in the **pivot table fields pane.**



PivotTable Fields
Choose fields to add to report:
Search
More Tables...
Drag fields between areas below:
Defer Layout Update
Sheet1
bike_tripdata
Avg ride_len on diff days
Total rides o

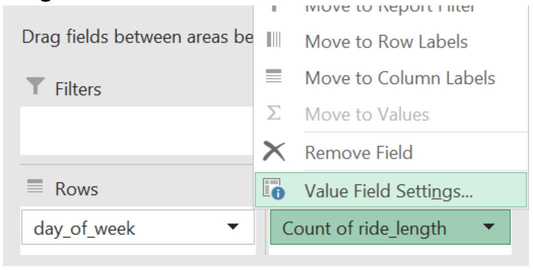
Count of ride_length	Column Labels		
Row Labels	casual	member	Grand Total
1	78251	58780	137031
2	43971	49850	93821
3	41455	57524	98979
4	42850	59611	102461
5	47793	61155	108948
6	56505	61644	118149
7	95230	68869	164099
Grand Total	406055	417433	823488

✓ member_casual
✓ ride_length
✓ day_of_week

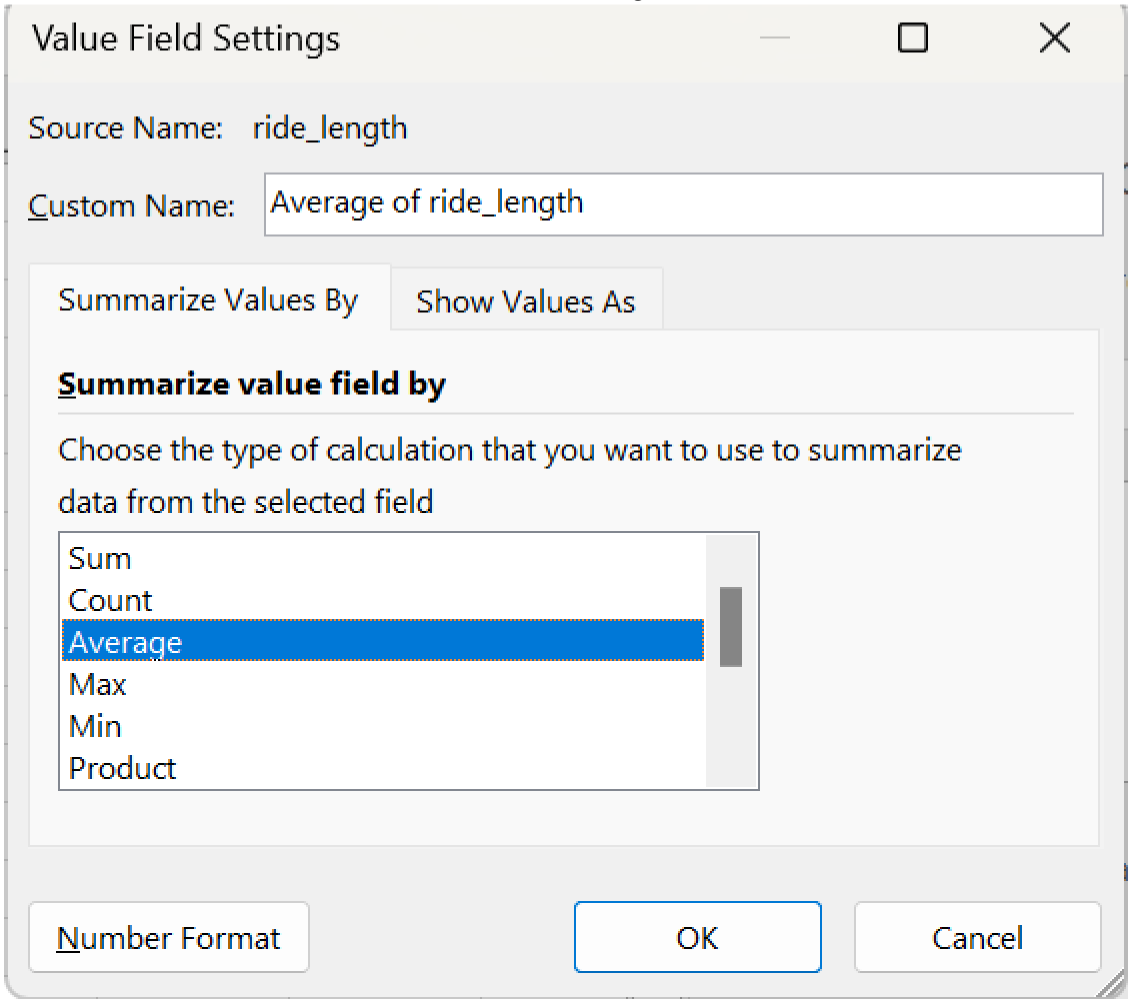
Filters:
Columns: member_casual
Rows: day_of_week
Values: Count of ride_length

This will generate a summary table in the sheets.

Since we want the average ride length, we'd go the the value field settings of the Count of ride length in the values section.



Now we'll set the Summarize value field to average.



Value Field Settings

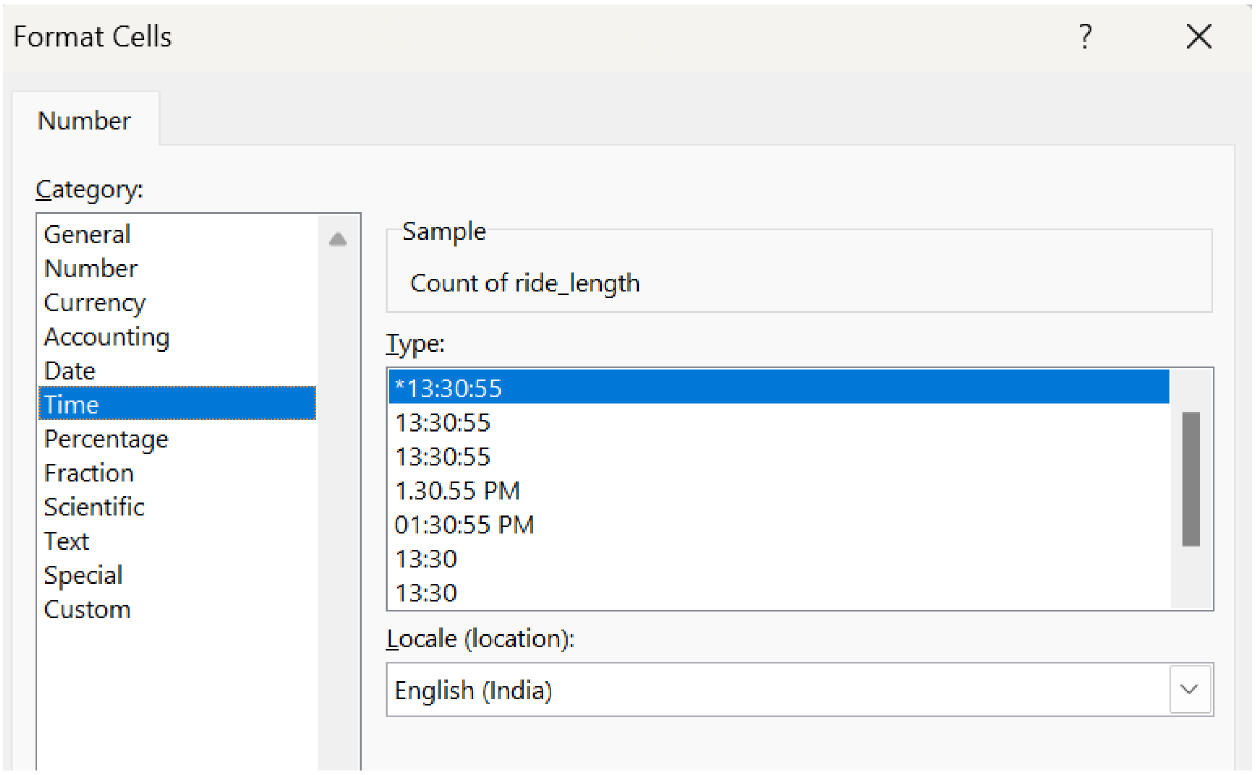
Source Name: ride_length

Custom Name: Average of ride_length

Summarize value field by

Choose the type of calculation that you want to use to summarize data from the selected field

Also we'll go to the number format section in the lower left and set it to time. This will change the type of our result to the following format 'hh/mm/ss'



Format Cells

Number

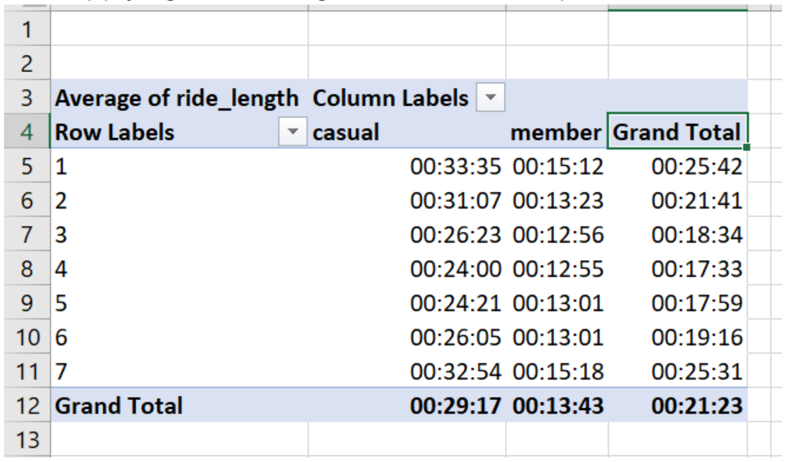
Category: General, Number, Currency, Accounting, Date, **Time**, Percentage, Fraction, Scientific, Text, Special, Custom

Sample: Count of ride_length

Type: ***13:30:55**, 13:30:55, 13:30:55, 1.30.55 PM, 01:30:55 PM, 13:30, 13:30

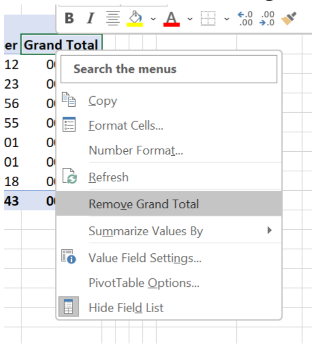
Locale (location): English (India)

After applying this, we'll get our desired output.

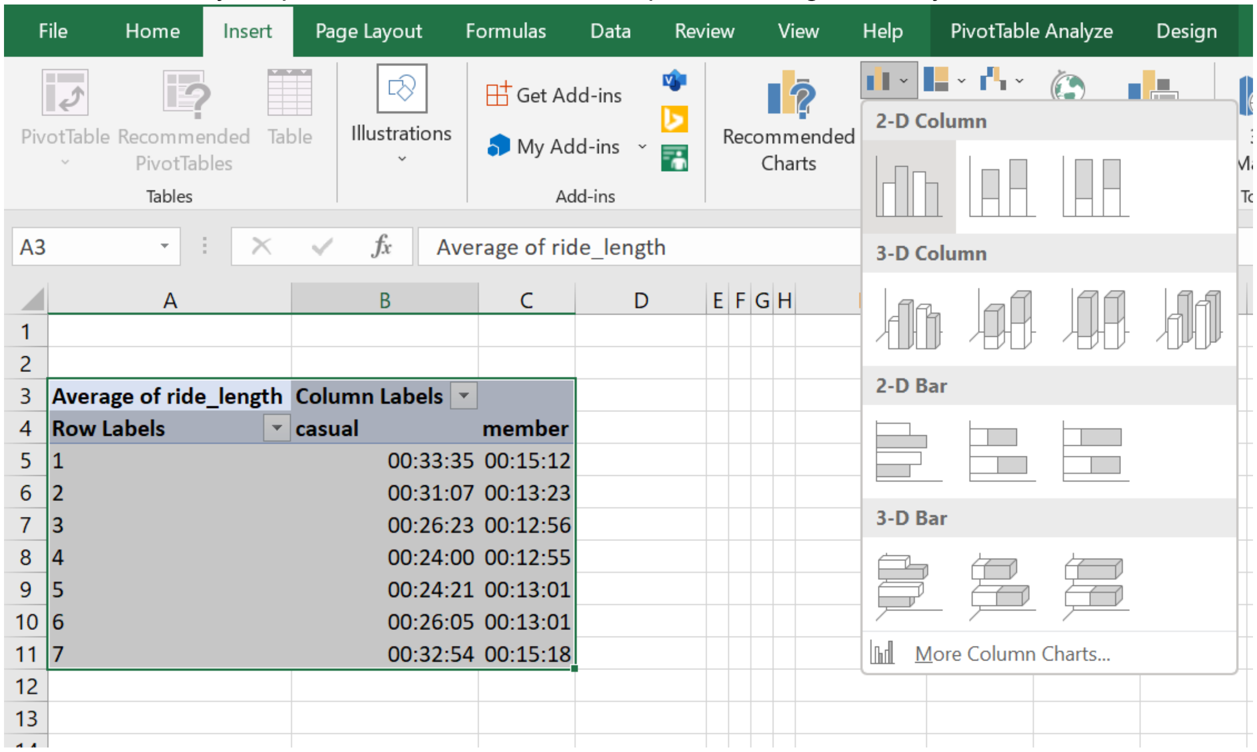


Average of ride_length	Column Labels		
Row Labels	casual	member	Grand Total
1	00:33:35	00:15:12	00:25:42
2	00:31:07	00:13:23	00:21:41
3	00:26:23	00:12:56	00:18:34
4	00:24:00	00:12:55	00:17:33
5	00:24:21	00:13:01	00:17:59
6	00:26:05	00:13:01	00:19:16
7	00:32:54	00:15:18	00:25:31
Grand Total	**00:29:17**	**00:13:43**	**00:21:23**

We can remove the grand totals by right clicking on the name and removing it.



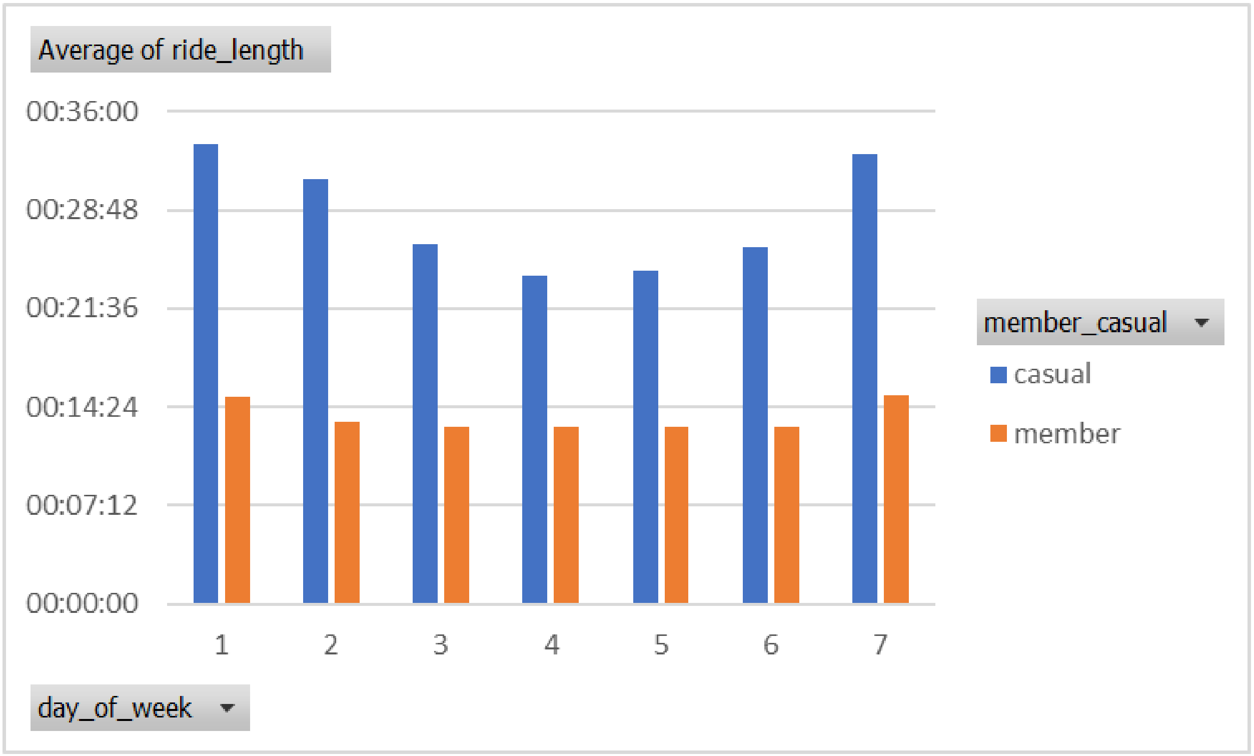
Now we can visualize it, select the table then follow the steps to choose the chart to visualize. You can choose your preferred chart. In this example, I'm using a side by side bar chart.



A3 | Average of ride_length

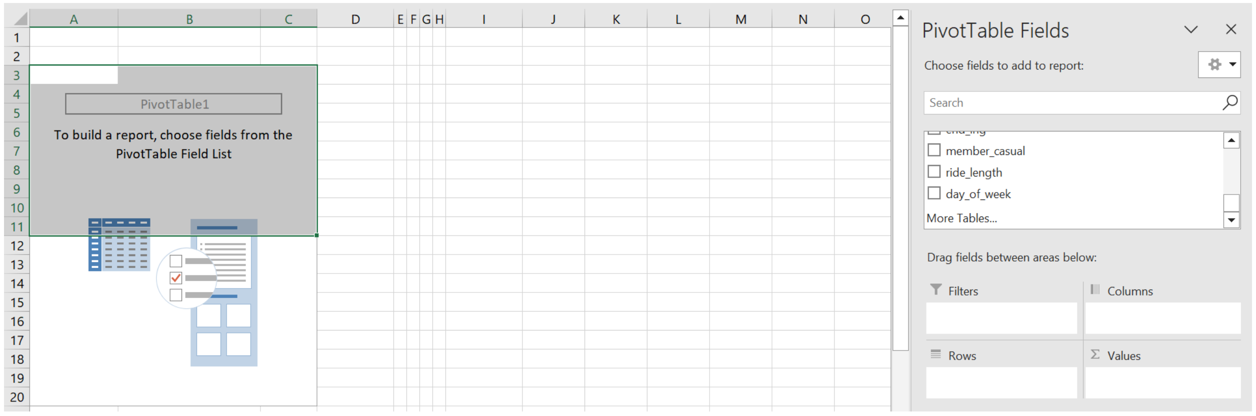
Average of ride_length	Column Labels	
Row Labels	casual	member
1	00:33:35	00:15:12
2	00:31:07	00:13:23
3	00:26:23	00:12:56
4	00:24:00	00:12:55
5	00:24:21	00:13:01
6	00:26:05	00:13:01
7	00:32:54	00:15:18

The results would be like this,



Let's now count the total rides by each member on each weekday.

Regenerate the pivot table insertion steps again,and reach upto this mark.


PivotTable1

To build a report, choose fields from the PivotTable Field List

PivotTable Fields

Choose fields to add to report:

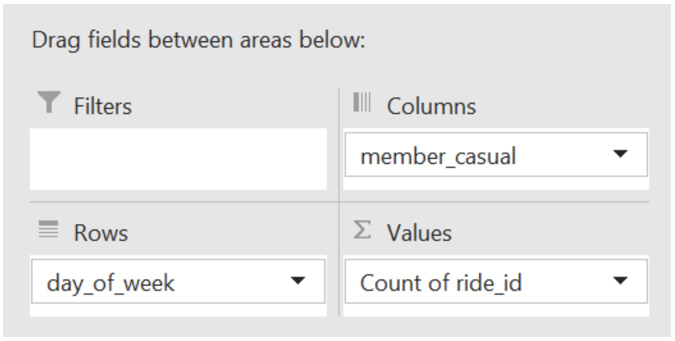
- ☐ member_casual
- ☐ ride_length
- ☐ day_of_week

More Tables...

Drag fields between areas below:

Filters | Columns | Rows | Values

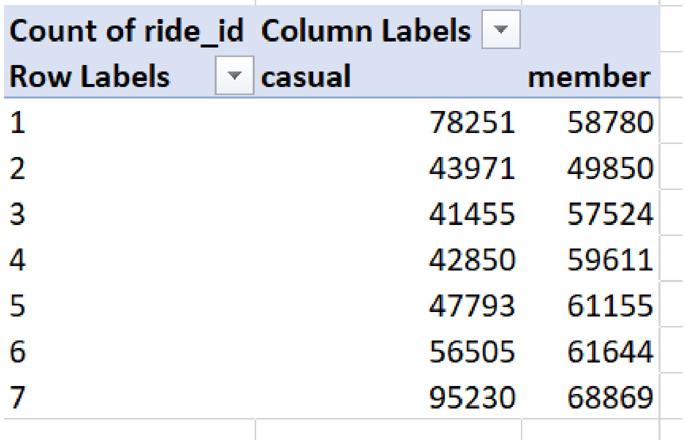
Now let's put the day_of_week and member_casual in the same places that we did before. This time rather than putting ride_length to values, we'd use **ride_id.**


Drag fields between areas below:

Filters	Columns
	member_casual

Rows	Values
day_of_week	Count of ride_id

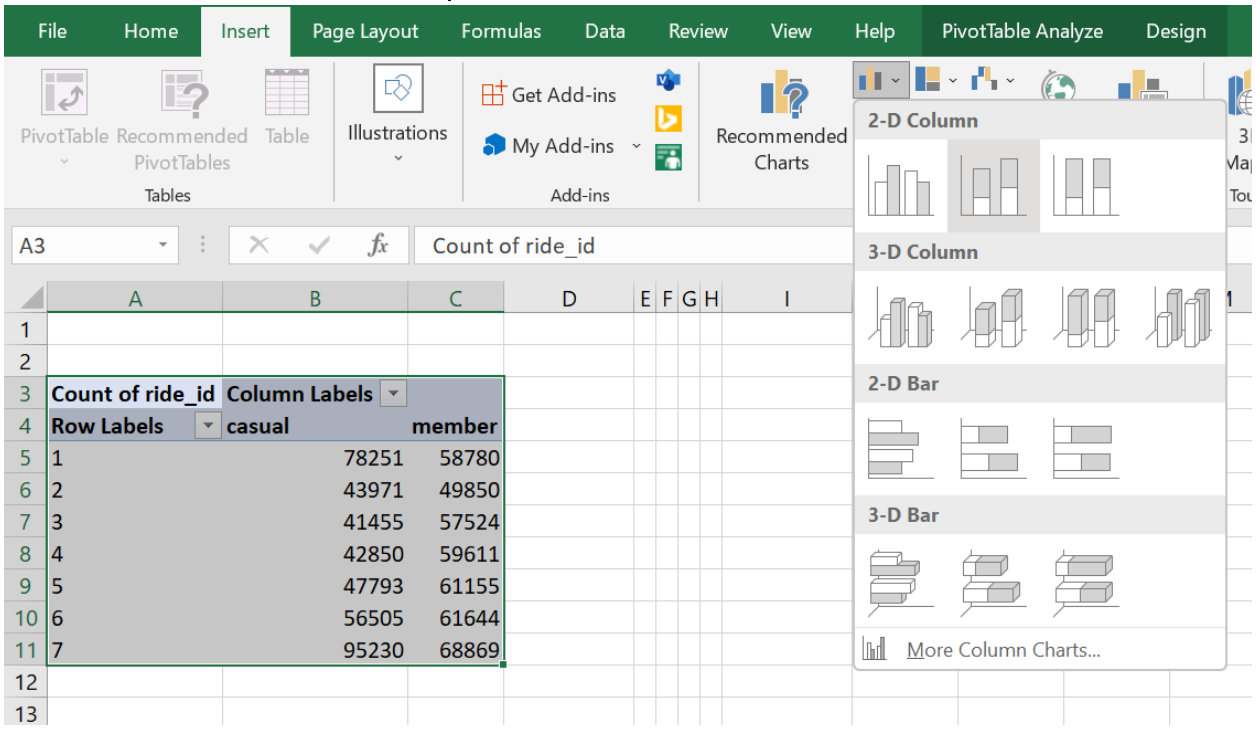
This ride_id is already set to count. And we'd get desired result.



Count of ride_id	Column Labels	
Row Labels	casual	member
1	78251	58780
2	43971	49850
3	41455	57524
4	42850	59611
5	47793	61155
6	56505	61644
7	95230	68869

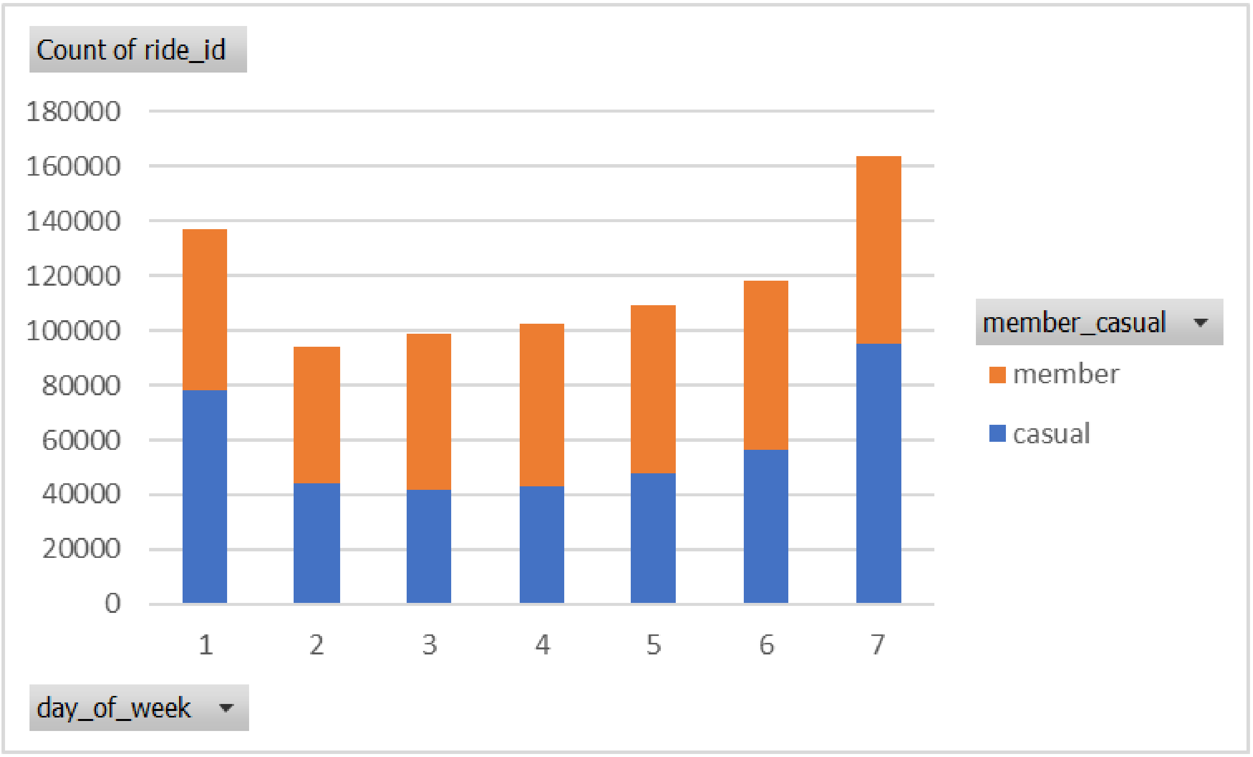
Now let's put the stacked bar chart to analyze this visually.
To get a stack bar chart, you can perform following operations

Select the table, and follow the steps.



Count of ride_id	Column Labels	
Row Labels	casual	member
1	78251	58780
2	43971	49850
3	41455	57524
4	42850	59611
5	47793	61155
6	56505	61644
7	95230	68869

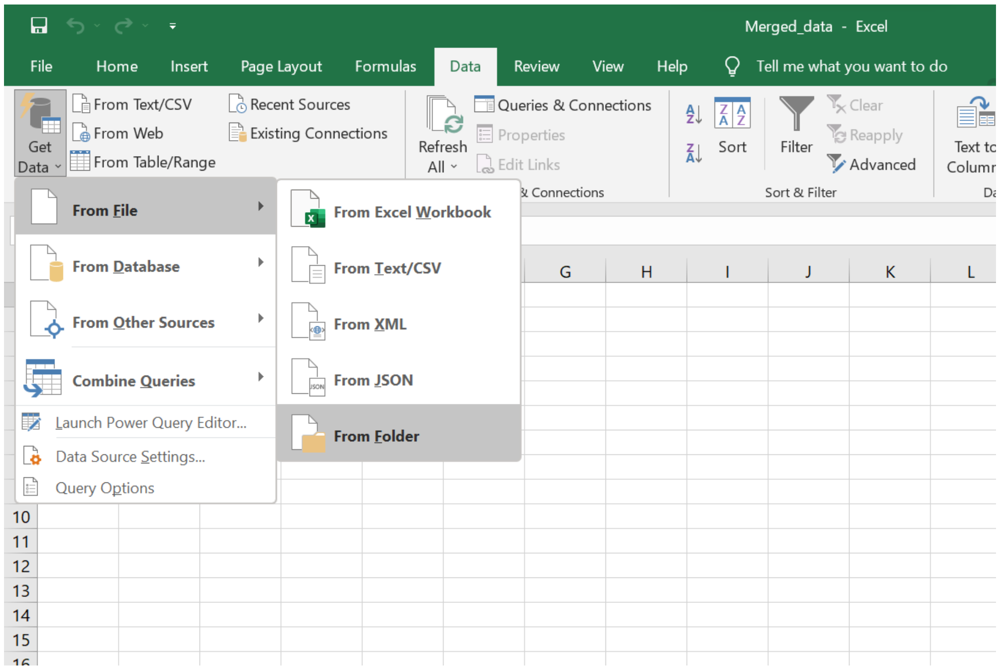
We'd get this



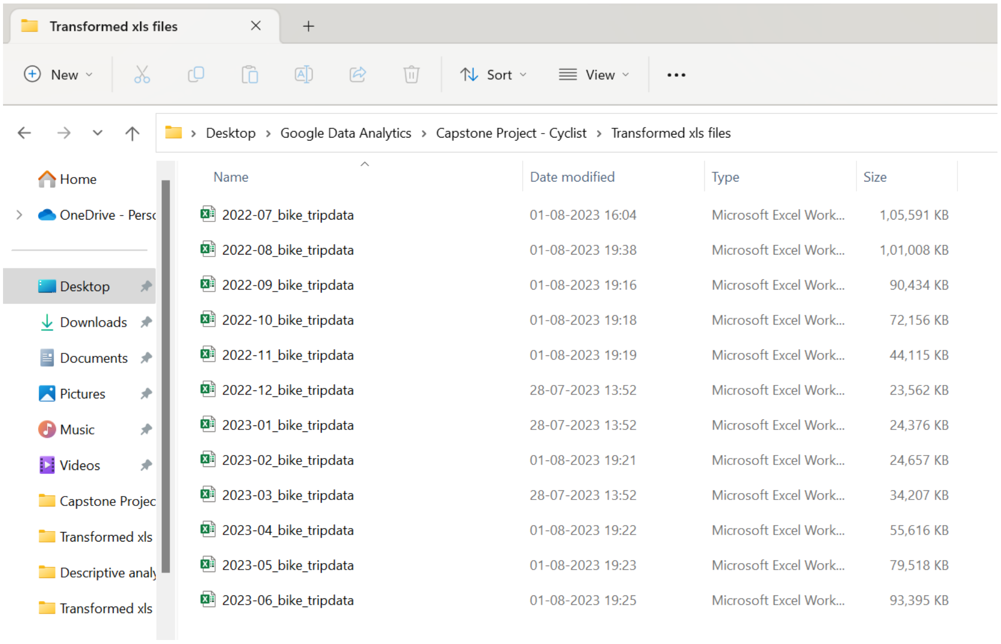
We can analyze with other months data like this, and see the trends for each month.

Working on Year data

We can easily merge the data by following these steps:

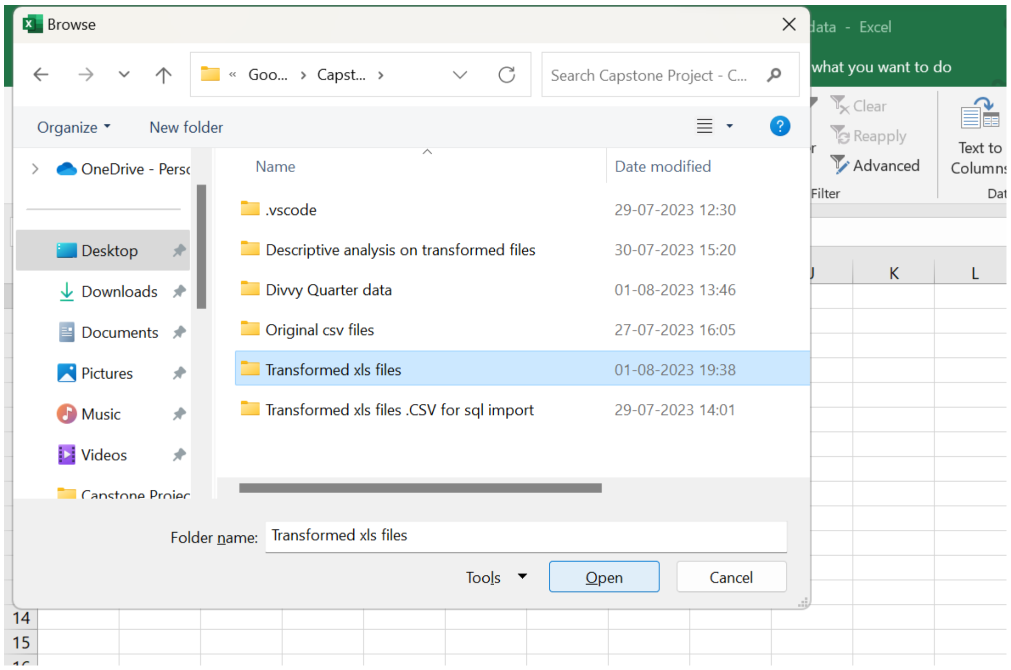


Now we can put all our files in a folder to give the location of the folder.



Name	Date modified	Type	Size
2022-07_bike_tripdata	01-08-2023 16:04	Microsoft Excel Work...	1,05,591 KB
2022-08_bike_tripdata	01-08-2023 19:38	Microsoft Excel Work...	1,01,008 KB
2022-09_bike_tripdata	01-08-2023 19:16	Microsoft Excel Work...	90,434 KB
2022-10_bike_tripdata	01-08-2023 19:18	Microsoft Excel Work...	72,156 KB
2022-11_bike_tripdata	01-08-2023 19:19	Microsoft Excel Work...	44,115 KB
2022-12_bike_tripdata	28-07-2023 13:52	Microsoft Excel Work...	23,562 KB
2023-01_bike_tripdata	28-07-2023 13:52	Microsoft Excel Work...	24,376 KB
2023-02_bike_tripdata	01-08-2023 19:21	Microsoft Excel Work...	24,657 KB
2023-03_bike_tripdata	28-07-2023 13:52	Microsoft Excel Work...	34,207 KB
2023-04_bike_tripdata	01-08-2023 19:22	Microsoft Excel Work...	55,616 KB
2023-05_bike_tripdata	01-08-2023 19:23	Microsoft Excel Work...	79,518 KB
2023-06_bike_tripdata	01-08-2023 19:25	Microsoft Excel Work...	93,395 KB

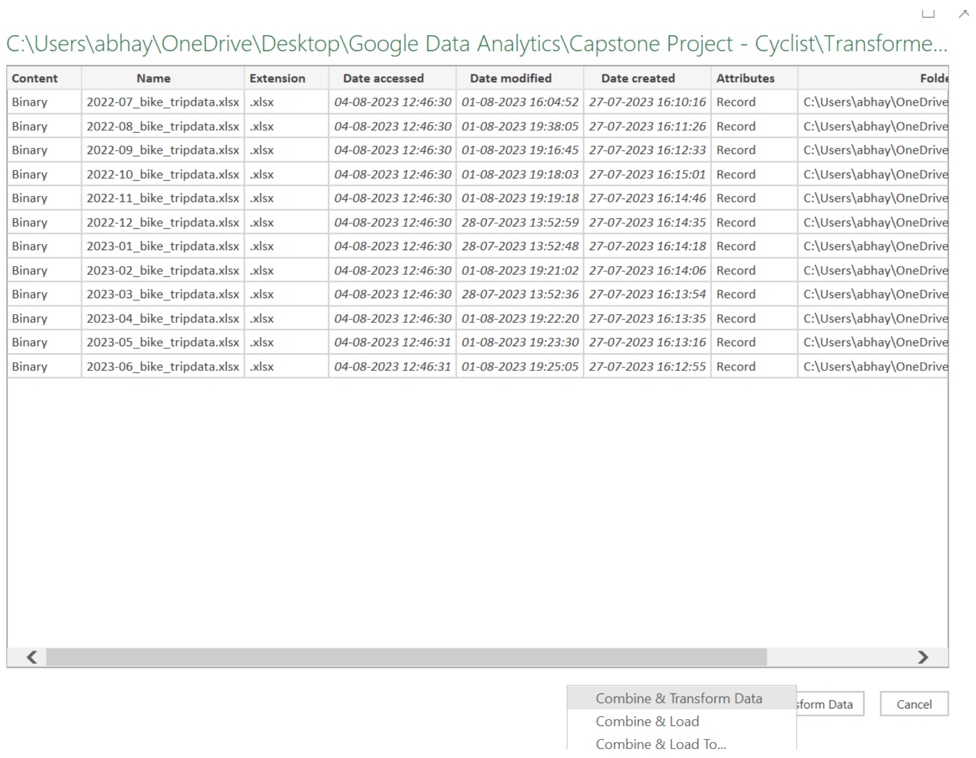
And now we can select the folder where all our files resides.



Name	Date modified
.vscode	29-07-2023 12:30
Descriptive analysis on transformed files	30-07-2023 15:20
Divvy Quarter data	01-08-2023 13:46
Original csv files	27-07-2023 16:05
Transformed xls files	01-08-2023 19:38
Transformed xls files .CSV for sql import	29-07-2023 14:01

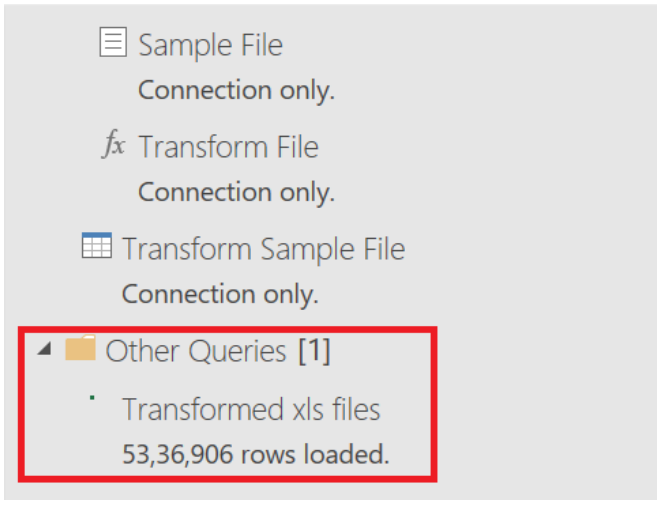
Folder name: Transformed xls files

If the data is clean and does not contain any error or change in number of columns, we can directly do **Combine and load,** But if we think that the data needs some cleanliness and change of data types, we can select **Combine and transform.** This will open the power query editor and we can perform the operations.


C:\Users\abhay\OneDrive\Desktop\Google Data Analytics\Capstone Project - Cyclist\Transforme...

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folde
Binary	2022-07_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 16:04:52	27-07-2023 16:10:16	Record	C:\Users\abhay\OneDrive
Binary	2022-08_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 19:38:05	27-07-2023 16:11:26	Record	C:\Users\abhay\OneDrive
Binary	2022-09_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 19:16:45	27-07-2023 16:12:33	Record	C:\Users\abhay\OneDrive
Binary	2022-10_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 19:18:03	27-07-2023 16:15:01	Record	C:\Users\abhay\OneDrive
Binary	2022-11_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 19:19:18	27-07-2023 16:14:46	Record	C:\Users\abhay\OneDrive
Binary	2022-12_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	28-07-2023 13:52:59	27-07-2023 16:14:35	Record	C:\Users\abhay\OneDrive
Binary	2023-01_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	28-07-2023 13:52:48	27-07-2023 16:14:18	Record	C:\Users\abhay\OneDrive
Binary	2023-02_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 19:21:02	27-07-2023 16:14:06	Record	C:\Users\abhay\OneDrive
Binary	2023-03_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	28-07-2023 13:52:36	27-07-2023 16:13:54	Record	C:\Users\abhay\OneDrive
Binary	2023-04_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:30	01-08-2023 19:22:20	27-07-2023 16:13:35	Record	C:\Users\abhay\OneDrive
Binary	2023-05_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:31	01-08-2023 19:23:30	27-07-2023 16:13:16	Record	C:\Users\abhay\OneDrive
Binary	2023-06_bike_tripdata.xlsx	.xlsx	04-08-2023 12:46:31	01-08-2023 19:25:05	27-07-2023 16:12:55	Record	C:\Users\abhay\OneDrive

Combining the data may take time. And analyzing it will also make a lot of time, as excel is suitable for less data and processing and analysis with big data is not possible with excel.

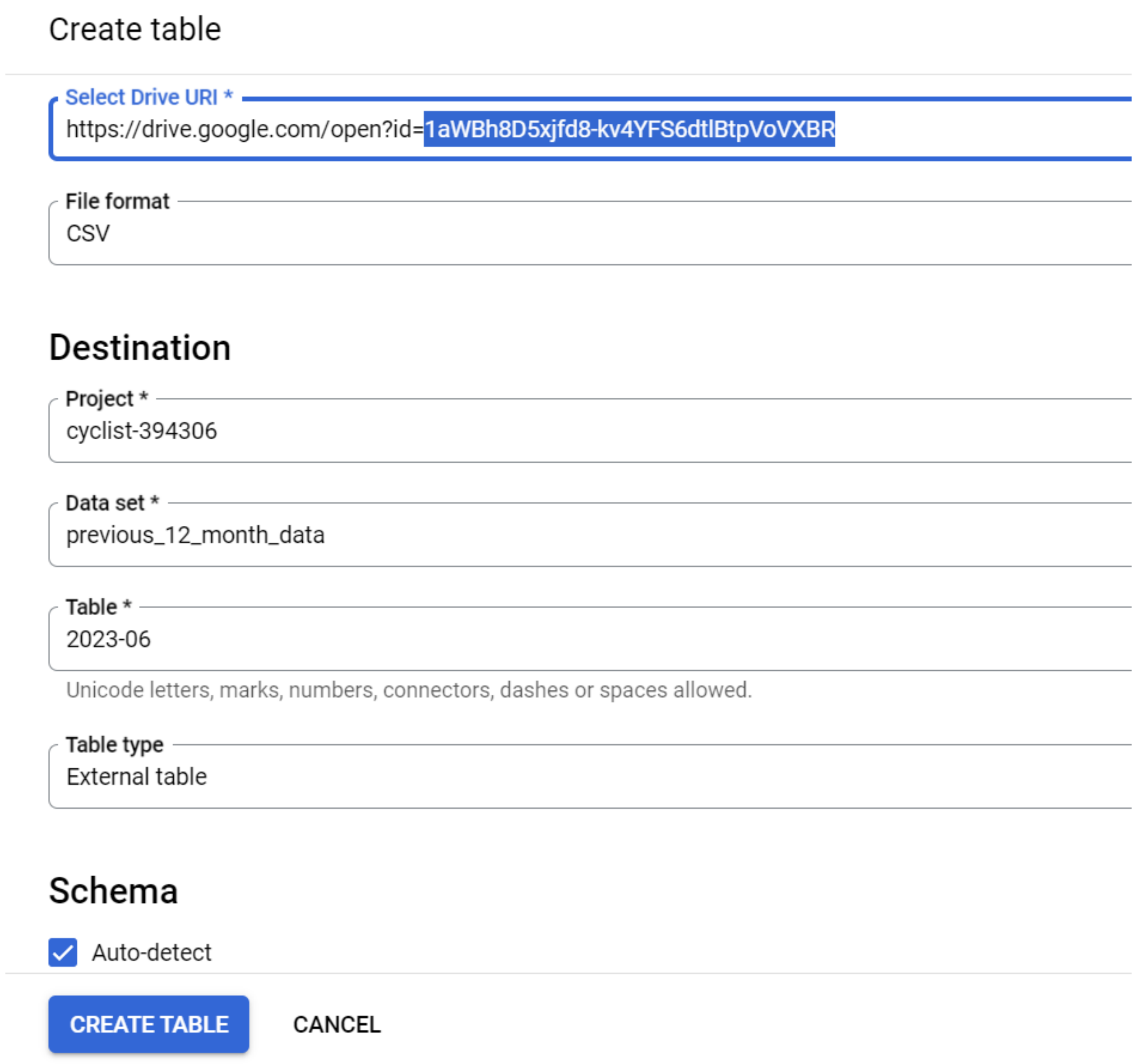


Sample File
Connection only.
Transform File
Connection only.
Transform Sample File
Connection only.
Other Queries [1]
Transformed xls files
53,36,906 rows loaded.

Now that the data is so big (about 5 million rows), it can't be proceed in a Excel file.
So now we'll switch to BigQuery

BigQuery

First I uploaded my files on the dataset. Greater than 100 mb is not available with the upload option.
So I rather use the drive option. I uploaded it on drive and use the link to give table.



Create table

Select Drive URI *
https://drive.google.com/open?id=1aWBh8D5xjfd8-kv4YFS6dtlBtpVoVXBR

File format
CSV

Destination

Project *
cyclist-394306

Data set *
previous_12_month_data

Table *
2023-06
Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

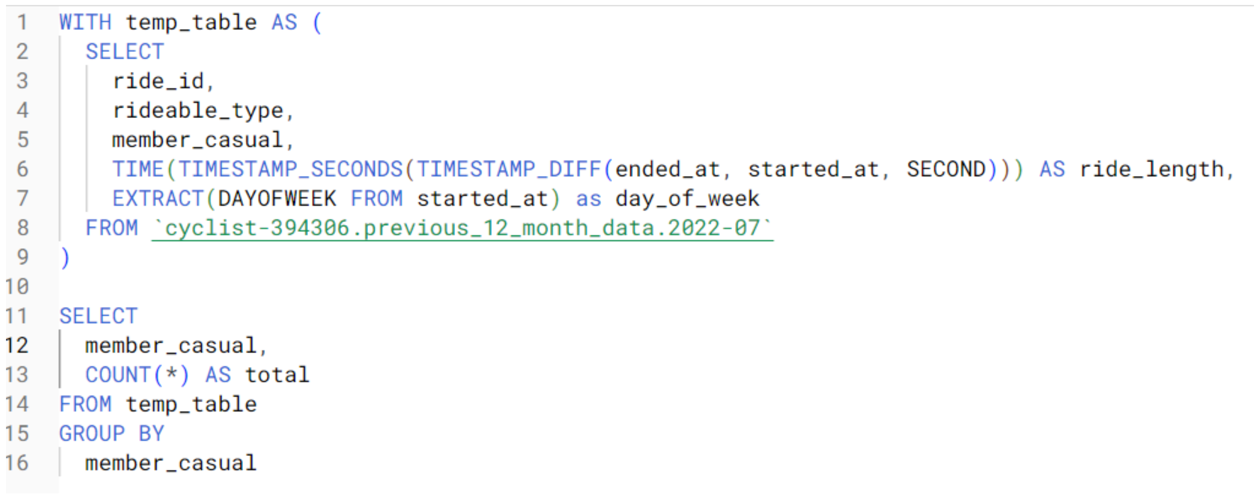
Table type
External table

Schema

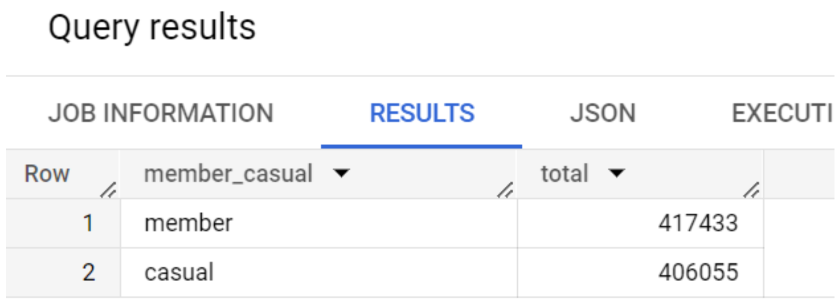
☑ Auto-detect

Working on Month data

In this below screenshot, I created a temp_table to store the following results from the query I'll use again and again to analyze from.
Here I added two columns, one for **ride_length** and the other for **day_of_week.**
Then I counted the total people grouped my their member_casual.



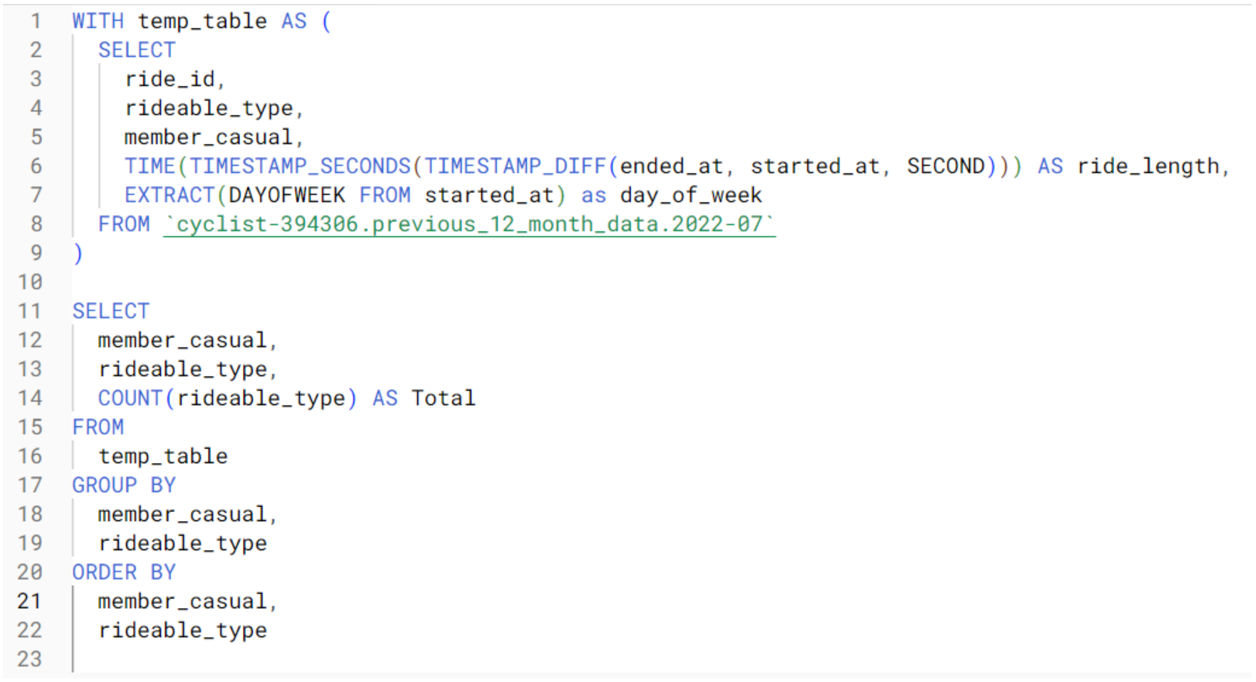
```
WITH temp_table AS (
  SELECT
    ride_id,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM `cyclist-394306.previous_12_month_data.2022-07`
)

SELECT
  member_casual,
  COUNT(*) AS total
FROM temp_table
GROUP BY
  member_casual
```



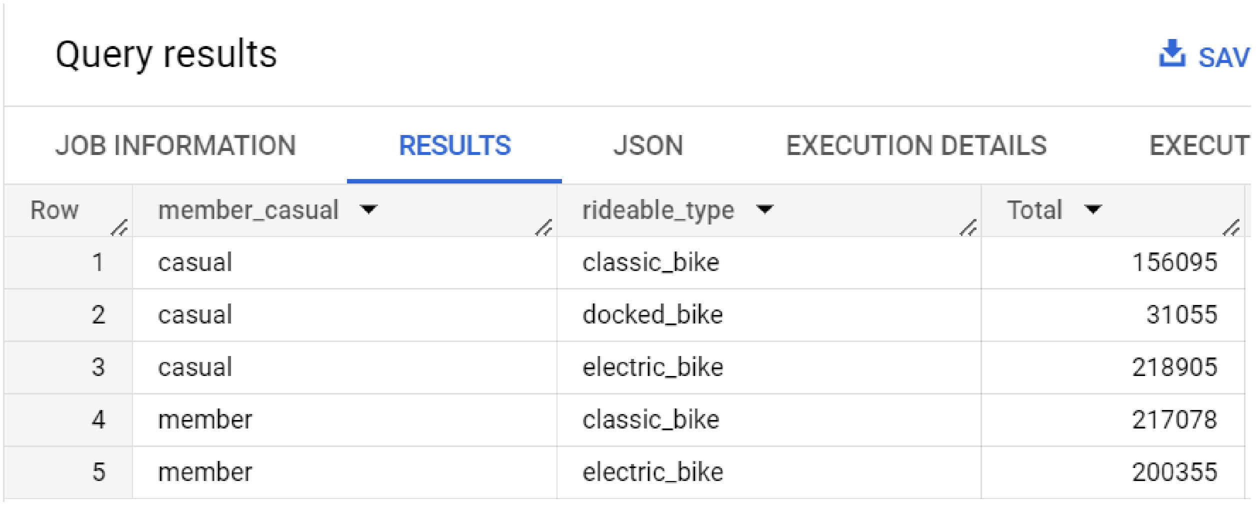
Row	member_casual	total
1	member	417433
2	casual	406055

Total rideable_type each member_casual prefer

Now let's calculate the total number of rideable_type each member_casual prefer.



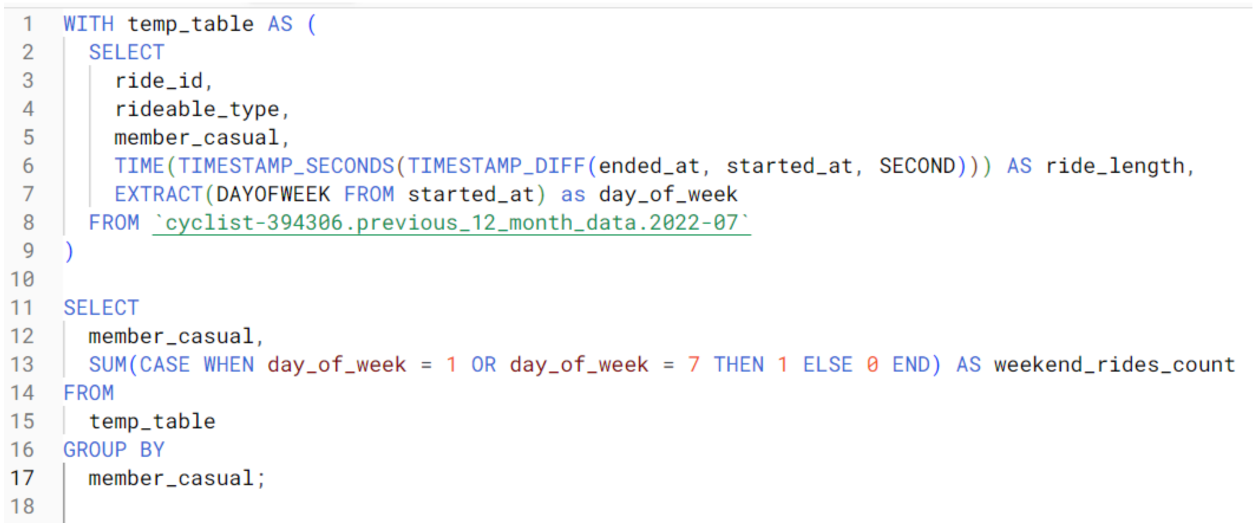
```
WITH temp_table AS (
  SELECT
    ride_id,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM `cyclist-394306.previous_12_month_data.2022-07`
)

SELECT
  member_casual,
  rideable_type,
  COUNT(rideable_type) AS Total
FROM
  temp_table
GROUP BY
  member_casual,
  rideable_type
ORDER BY
  member_casual,
  rideable_type

```



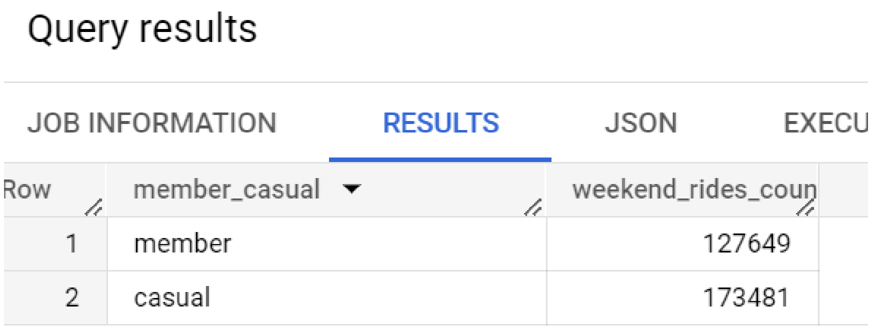
Row	member_casual	rideable_type	Total
1	casual	classic_bike	156095
2	casual	docked_bike	31055
3	casual	electric_bike	218905
4	member	classic_bike	217078
5	member	electric_bike	200355

Total rides by each member_casual on weekends.

Now let's calculate the total rides count by each member_casual on weekends.



```
WITH temp_table AS (
  SELECT
    ride_id,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM `cyclist-394306.previous_12_month_data.2022-07`
)

SELECT
  member_casual,
  SUM(CASE WHEN day_of_week = 1 OR day_of_week = 7 THEN 1 ELSE 0 END) AS weekend_rides_count
FROM
  temp_table
GROUP BY
  member_casual;

```



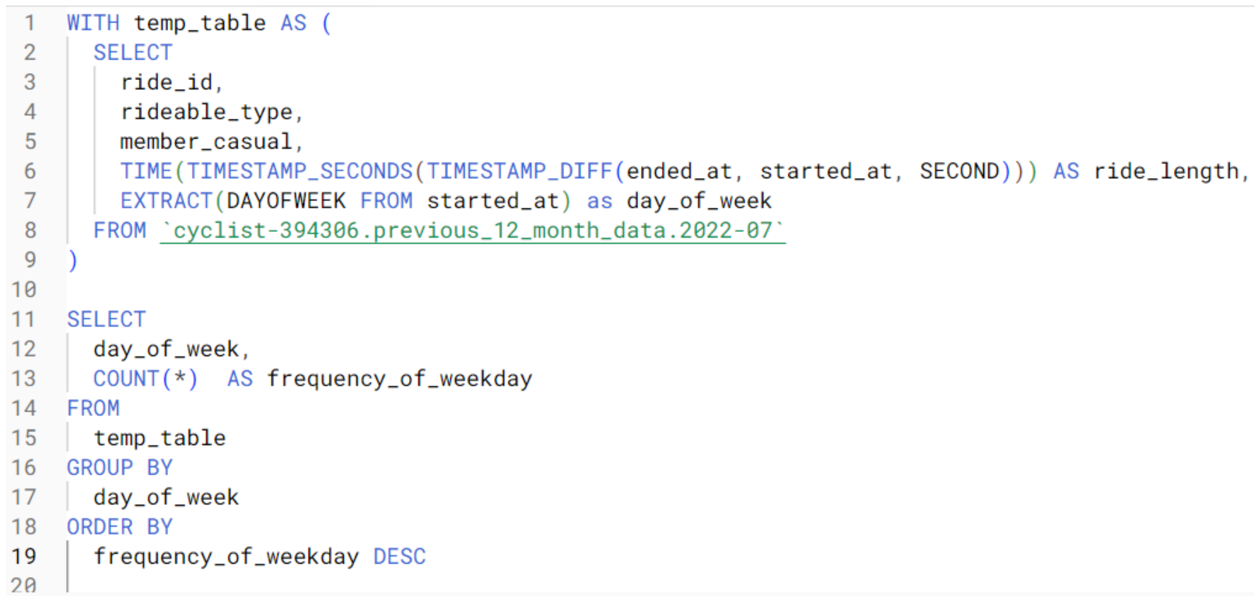
Query results

JOB INFORMATION RESULTS JSON EXECU

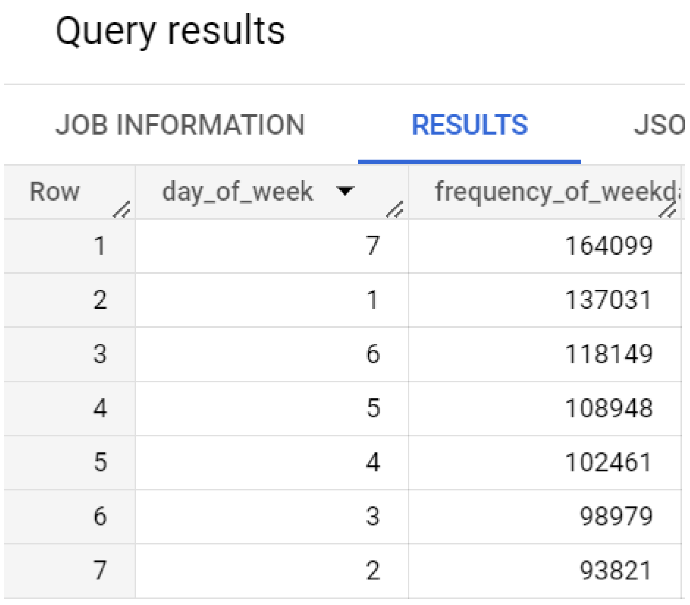
Row	member_casual	weekend_rides_coun
1	member	127649
2	casual	173481

Total No. of rides on each weekday

Now let us calculate the number of rides on each weekday and find out what weekday has the highest number of rides.



```sql
WITH temp_table AS (
  SELECT
    ride_id,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM `cyclist-394306.previous_12_month_data.2022-07`
)

SELECT
  day_of_week,
  COUNT(*)  AS frequency_of_weekday
FROM
  temp_table
GROUP BY
  day_of_week
ORDER BY
  frequency_of_weekday DESC

```


Query results

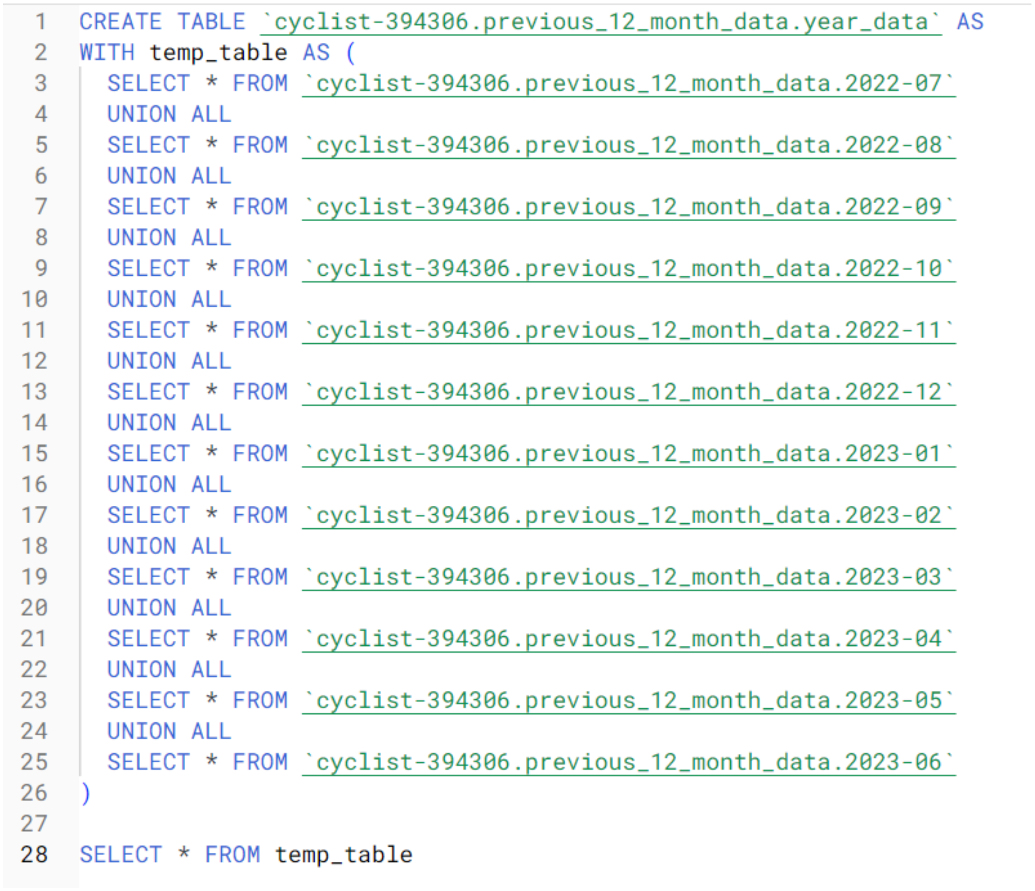
Row	day_of_week	frequency_of_weekd
1	7	164099
2	1	137031
3	6	118149
4	5	108948
5	4	102461
6	3	98979
7	2	93821

Now we can conclude that weekday 7 or Saturday has the highest number of rides.

Working on Year data

Combining each month data.

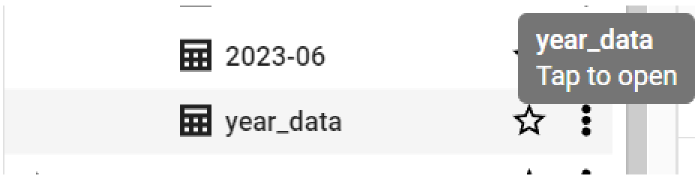
Now we will work on yearly data.
But we only have data in separate tables. So let's create a new table for year data.



```sql
CREATE TABLE `cyclist-394306.previous_12_month_data.year_data` AS
WITH temp_table AS (
  SELECT * FROM `cyclist-394306.previous_12_month_data.2022-07`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2022-08`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2022-09`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2022-10`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2022-11`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2022-12`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2023-01`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2023-02`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2023-03`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2023-04`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2023-05`
  UNION ALL
  SELECT * FROM `cyclist-394306.previous_12_month_data.2023-06`
)

SELECT * FROM temp_table
```

The UNION ALL will combine all rows from the different tables.
And the SELECT * FROM temp_table with flash the data then it will be stored in the table.

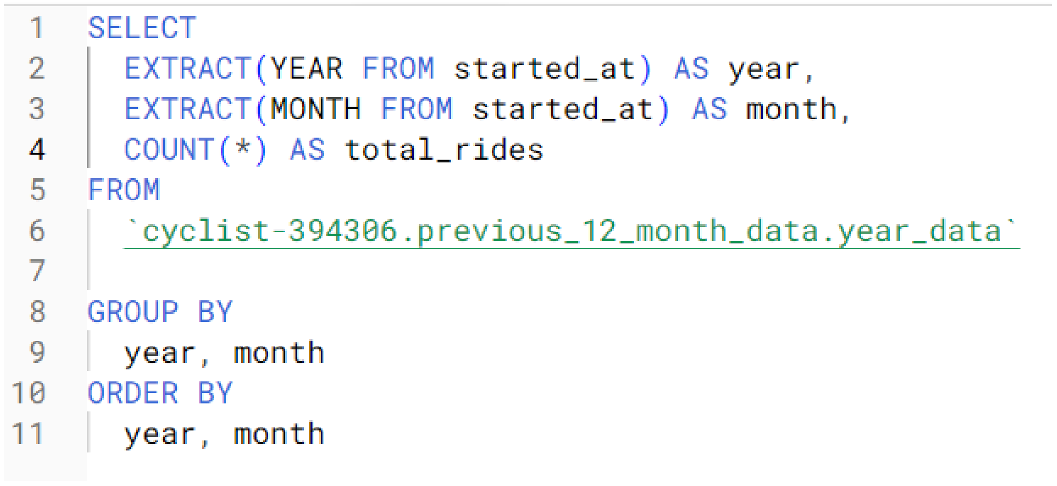
Then table will be created at the specified location



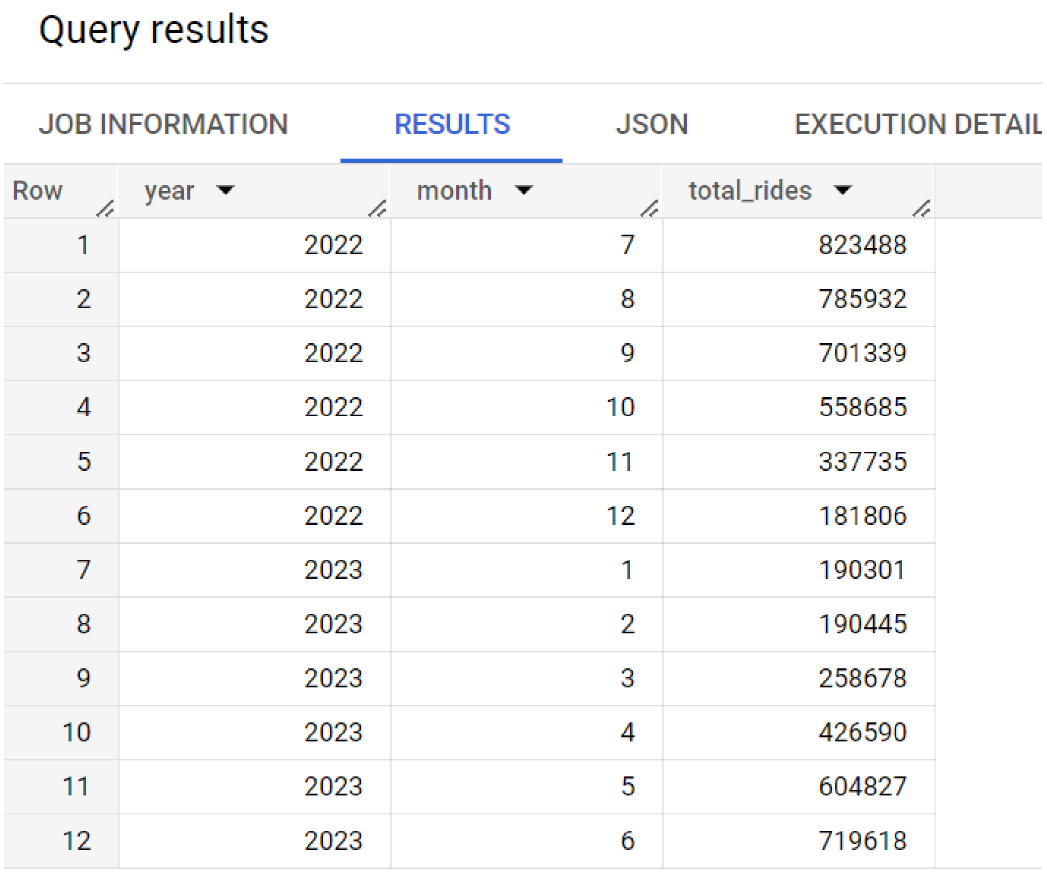
This will make query faster. Now we can use this data for yearly data analysis.

Total rides each month

Let's analyze total rides done in each month.



```sql
SELECT
  EXTRACT(YEAR FROM started_at) AS year,
  EXTRACT(MONTH FROM started_at) AS month,
  COUNT(*) AS total_rides
FROM
  `cyclist-394306.previous_12_month_data.year_data`

GROUP BY
  year, month
ORDER BY
  year, month
```

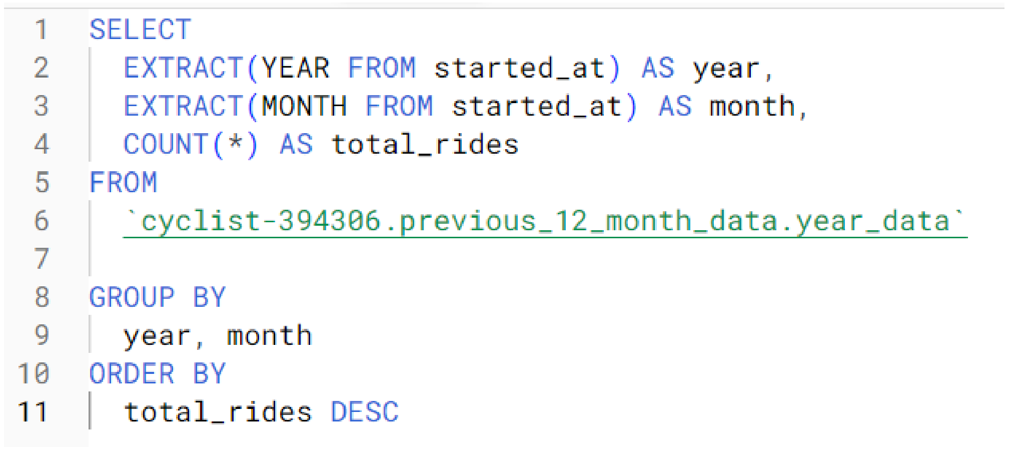


Query results

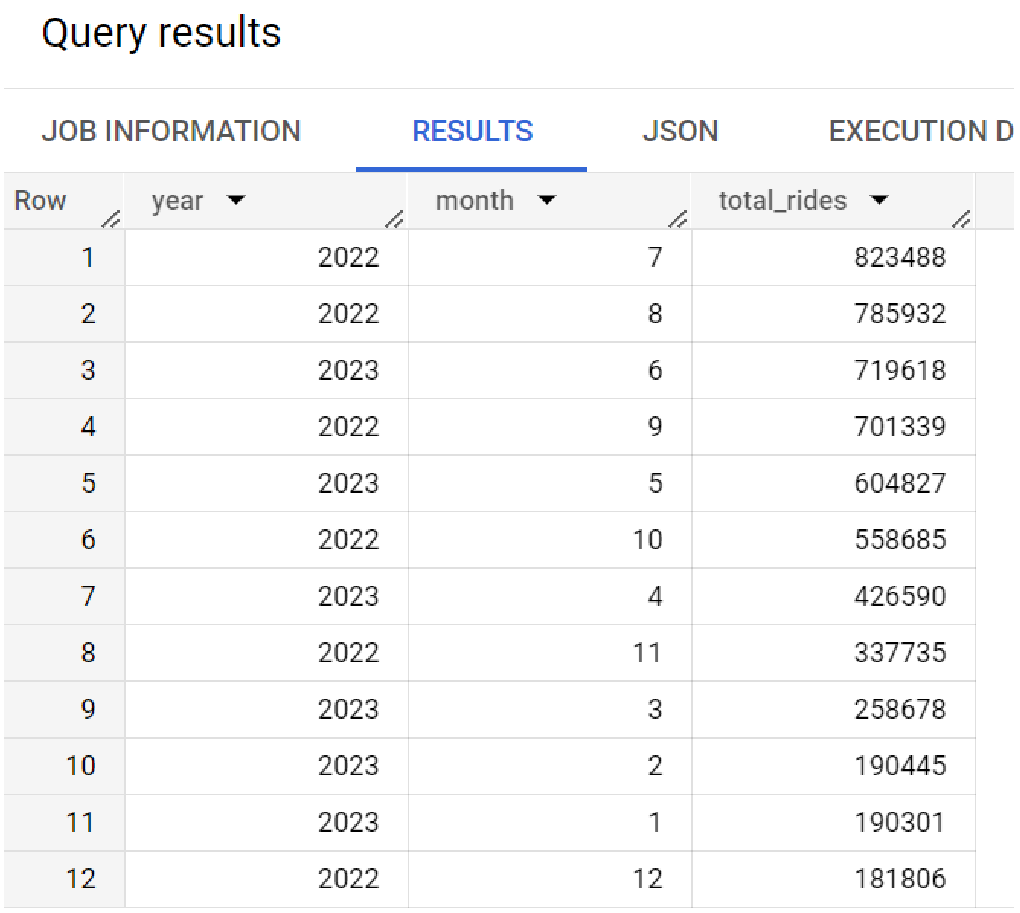
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

Row	year	month	total_rides
1	2022	7	823488
2	2022	8	785932
3	2022	9	701339
4	2022	10	558685
5	2022	11	337735
6	2022	12	181806
7	2023	1	190301
8	2023	2	190445
9	2023	3	258678
10	2023	4	426590
11	2023	5	604827
12	2023	6	719618

Let’s order the above results in descending to know which month has the highest number of rides.



```
SELECT
  EXTRACT(YEAR FROM started_at) AS year,
  EXTRACT(MONTH FROM started_at) AS month,
  COUNT(*) AS total_rides
FROM
  `cyclist-394306.previous_12_month_data.year_data`

GROUP BY
  year, month
ORDER BY
  total_rides DESC
```



Query results

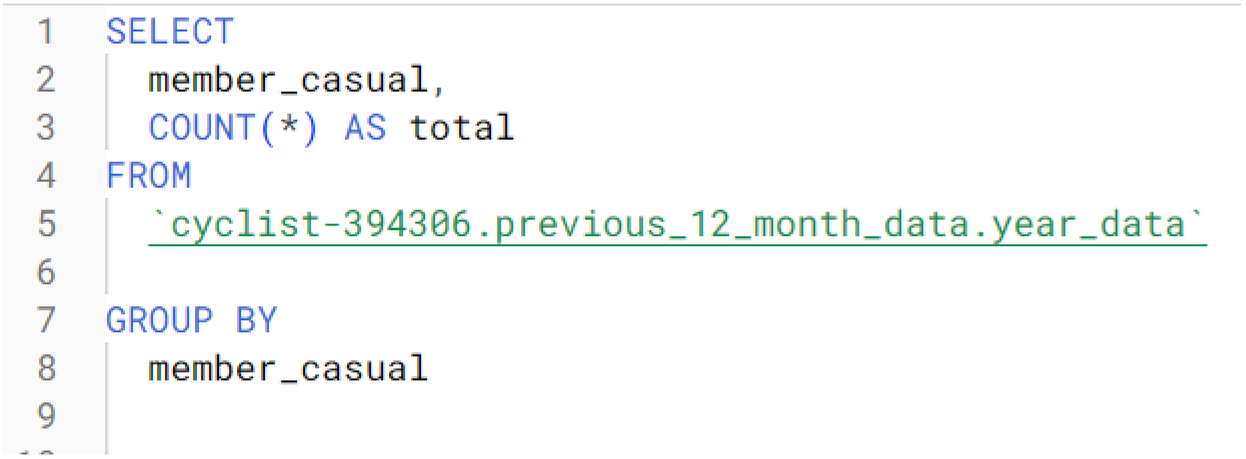
JOB INFORMATION RESULTS JSON EXECUTION D

Row	year	month	total_rides
1	2022	7	823488
2	2022	8	785932
3	2023	6	719618
4	2022	9	701339
5	2023	5	604827
6	2022	10	558685
7	2023	4	426590
8	2022	11	337735
9	2023	3	258678
10	2023	2	190445
11	2023	1	190301
12	2022	12	181806

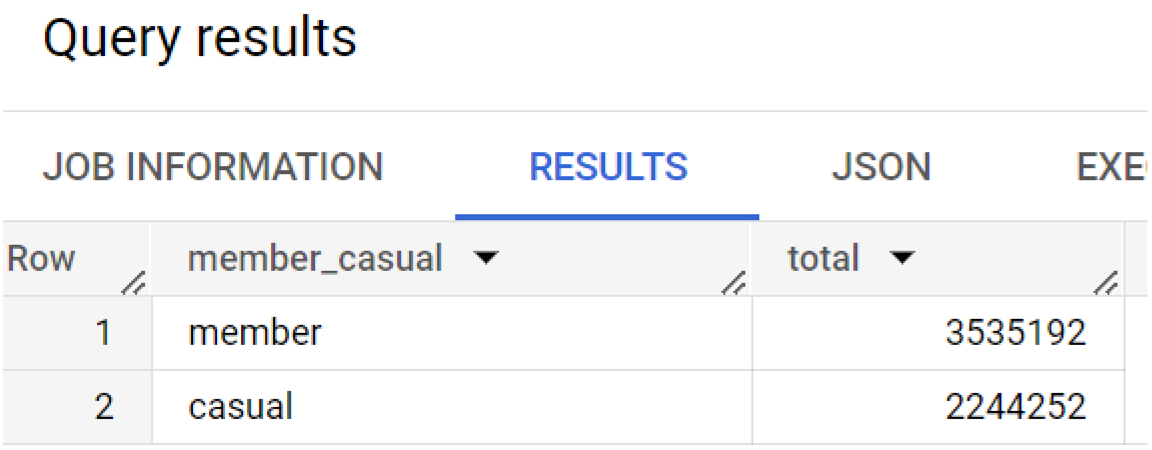
This data concludes that July, 2022 has highest number of rides.
While December, 2022 accounts for lowest number of rides.

Total no. of each member_casual in an year data

Now let's count the total members or casual customer in an year



```
SELECT
  member_casual,
  COUNT(*) AS total
FROM
  `cyclist-394306.previous_12_month_data.year_data`

GROUP BY
  member_casual

```



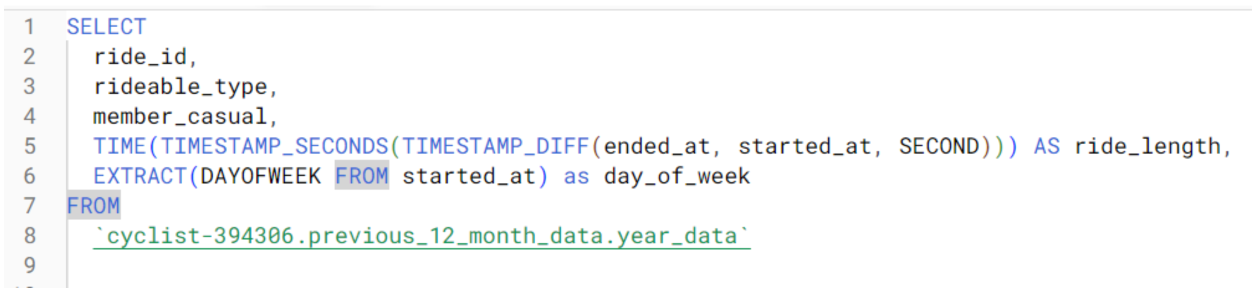
Query results

Row	member_casual	total
1	member	3535192
2	casual	2244252

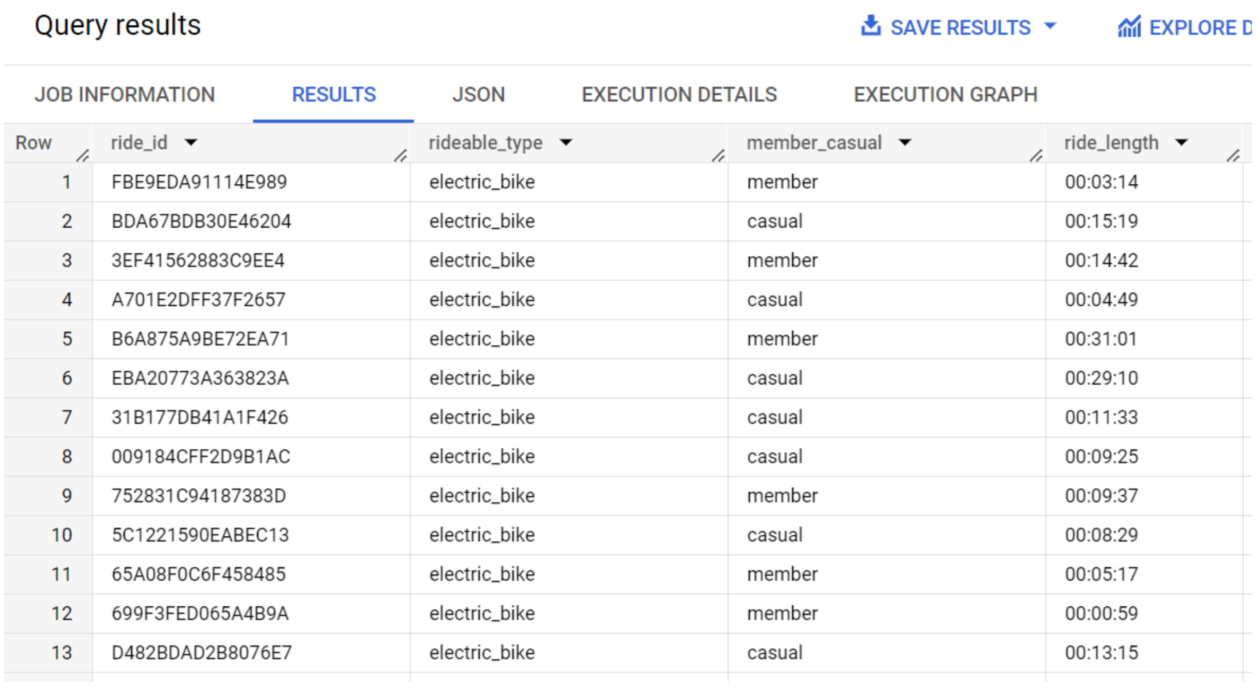
This data shows that there are more members riding bikes than the casual members.

Adding two more columns to perform more descriptive analysis

w
Now let's create two more columns **ride_length** and **day_of_week** to do more descriptive analysis



```
SELECT
  ride_id,
  rideable_type,
  member_casual,
  TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
  EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
FROM
  `cyclist-394306.previous_12_month_data.year_data`
```

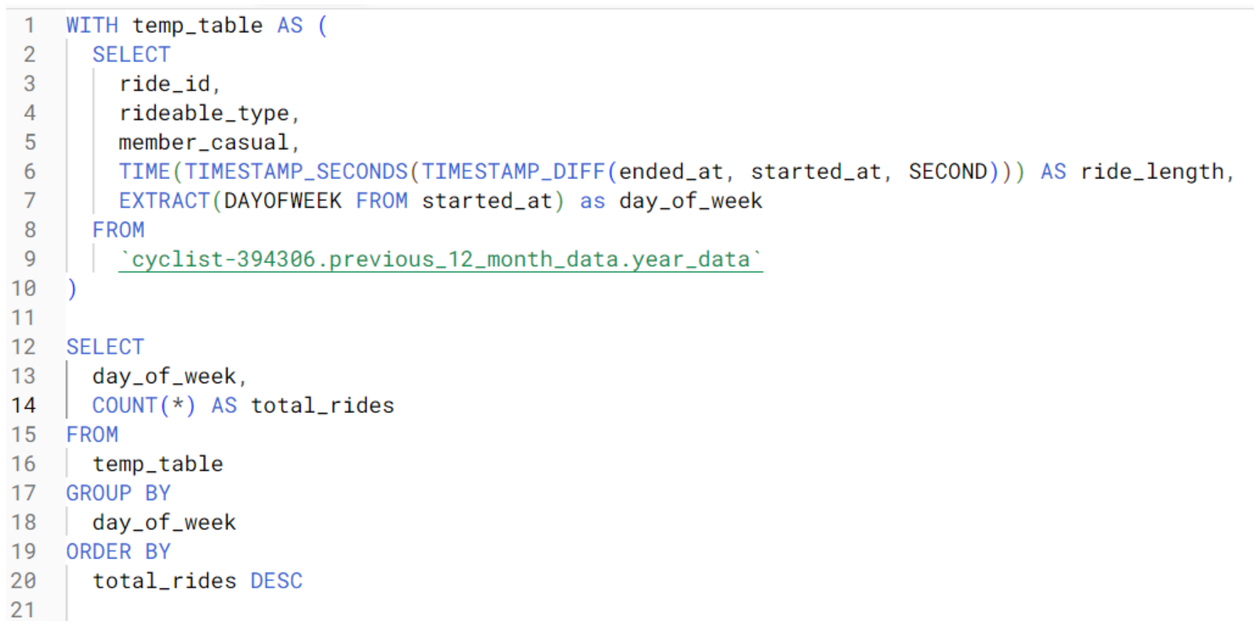


Row	ride_id	rideable_type	member_casual	ride_length
1	FBE9EDA91114E989	electric_bike	member	00:03:14
2	BDA67BDB30E46204	electric_bike	casual	00:15:19
3	3EF41562883C9EE4	electric_bike	member	00:14:42
4	A701E2DFF37F2657	electric_bike	casual	00:04:49
5	B6A875A9BE72EA71	electric_bike	member	00:31:01
6	EBA20773A363823A	electric_bike	casual	00:29:10
7	31B177DB41A1F426	electric_bike	casual	00:11:33
8	009184CFF2D9B1AC	electric_bike	casual	00:09:25
9	752831C94187383D	electric_bike	member	00:09:37
10	5C1221590EABEC13	electric_bike	casual	00:08:29
11	65A08F0C6F458485	electric_bike	member	00:05:17
12	699F3FED065A4B9A	electric_bike	member	00:00:59
13	D482BDAD2B8076E7	electric_bike	casual	00:13:15

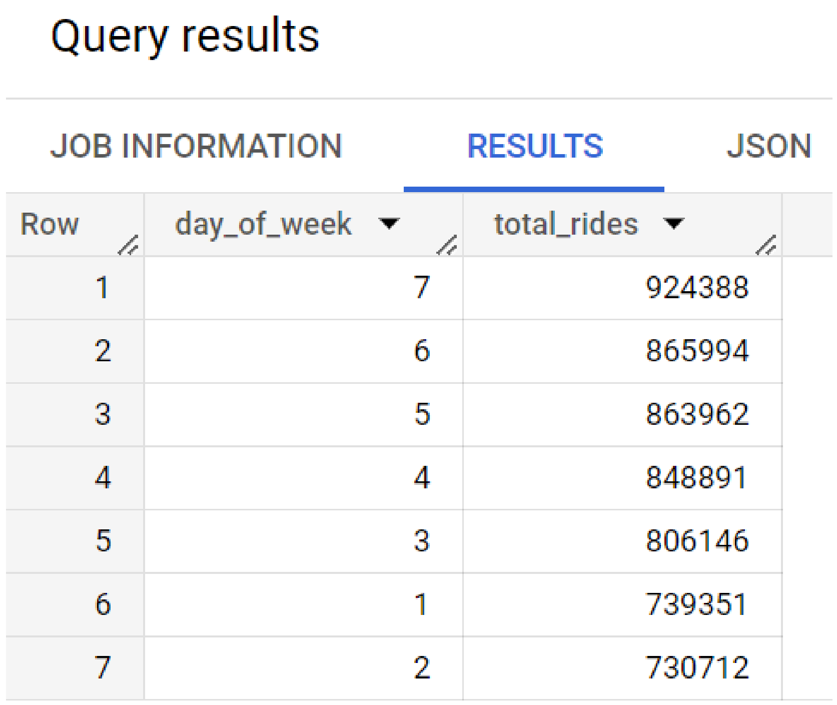
Now we can make this addition of two more variable as temp_table so we don't need to write it again and again for our analysis.

Most preferred day for riding

Now let’s find out which day is most preferred for riding bikes in an year data.



```sql
WITH temp_table AS (
  SELECT
    ride_id,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM
    `cyclist-394306.previous_12_month_data.year_data`
)

SELECT
  day_of_week,
  COUNT(*) AS total_rides
FROM
  temp_table
GROUP BY
  day_of_week
ORDER BY
  total_rides DESC

```

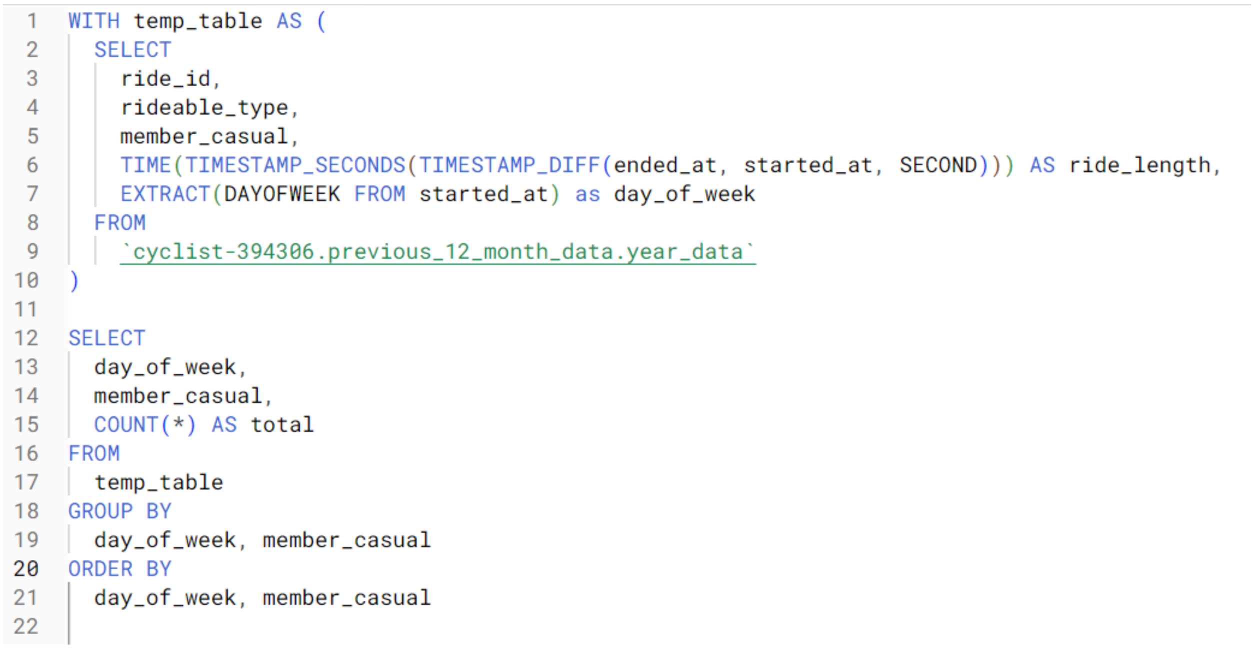


Query results

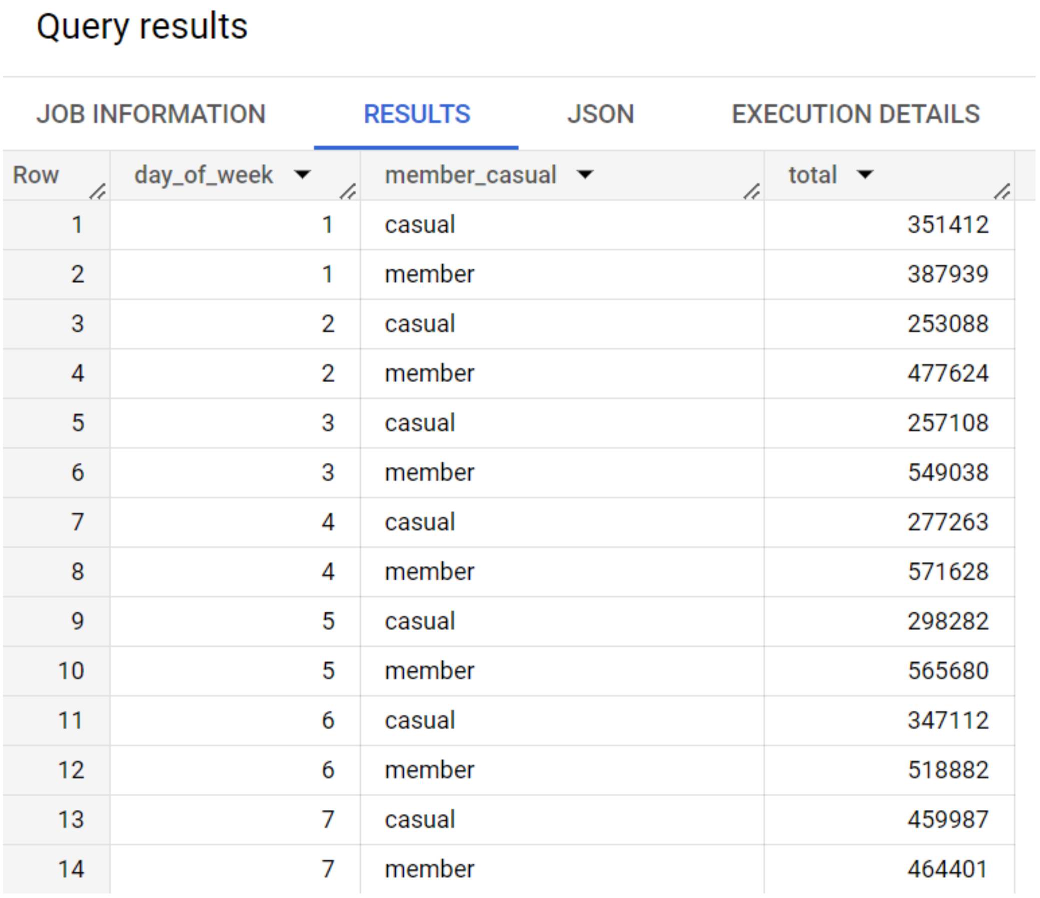
Row	day_of_week	total_rides
1	7	924388
2	6	865994
3	5	863962
4	4	848891
5	3	806146
6	1	739351
7	2	730712

This data reflects that Saturday is the most preferred day.
Now let’s find out total count of different member_casual on each day.

Find out total member_casual rides each weekday



```
WITH temp_table AS (
  SELECT
    ride_id,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM
    `cyclist-394306.previous_12_month_data.year_data`
)

SELECT
  day_of_week,
  member_casual,
  COUNT(*) AS total
FROM
  temp_table
GROUP BY
  day_of_week, member_casual
ORDER BY
  day_of_week, member_casual

```



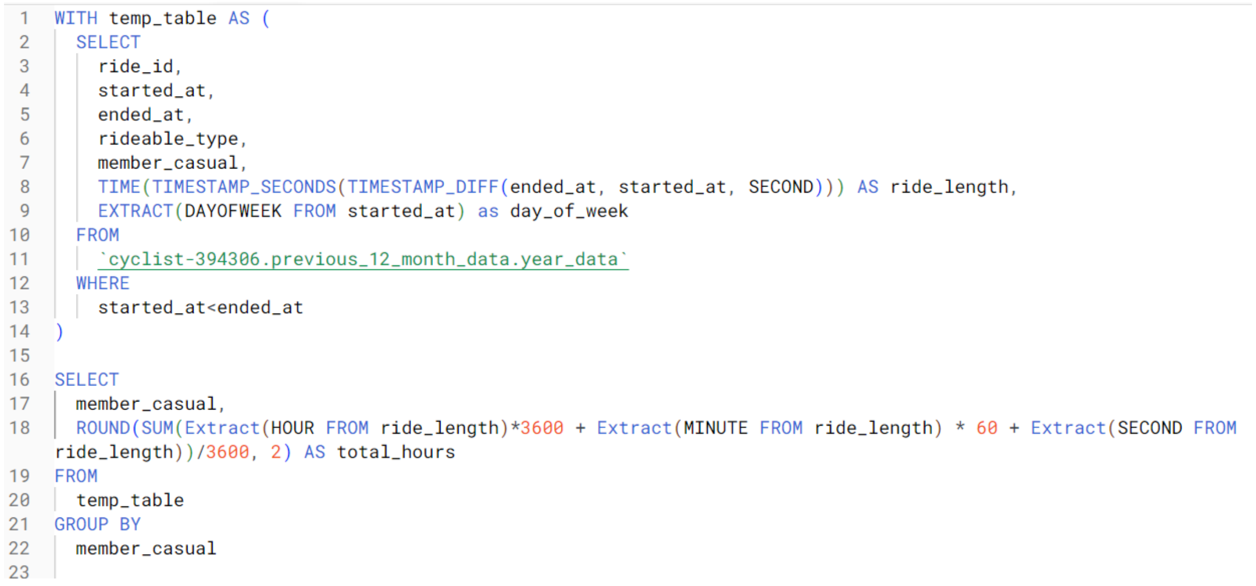
Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

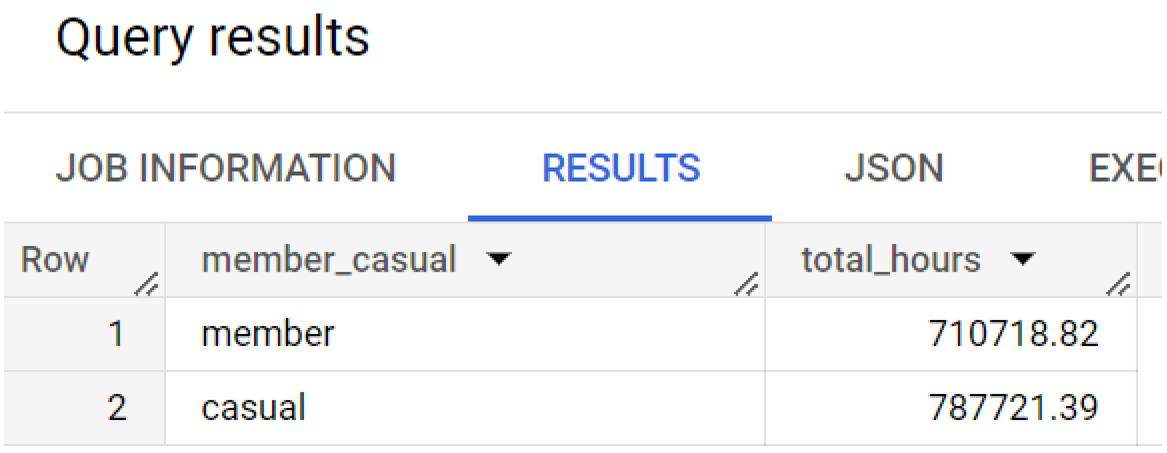
Row	day_of_week	member_casual	total
1	1	casual	351412
2	1	member	387939
3	2	casual	253088
4	2	member	477624
5	3	casual	257108
6	3	member	549038
7	4	casual	277263
8	4	member	571628
9	5	casual	298282
10	5	member	565680
11	6	casual	347112
12	6	member	518882
13	7	casual	459987
14	7	member	464401

Total Ride length in hours each member has in the year

Now let's find out how much ride_length in total each member has.



```
WITH temp_table AS (
  SELECT
    ride_id,
    started_at,
    ended_at,
    rideable_type,
    member_casual,
    TIME(TIMESTAMP_SECONDS(TIMESTAMP_DIFF(ended_at, started_at, SECOND))) AS ride_length,
    EXTRACT(DAYOFWEEK FROM started_at) as day_of_week
  FROM
    `cyclist-394306.previous_12_month_data.year_data`
  WHERE
    started_at<ended_at
)

SELECT
  member_casual,
  ROUND(SUM(Extract(HOUR FROM ride_length)*3600 + Extract(MINUTE FROM ride_length) * 60 + Extract(SECOND FROM 
ride_length))/3600, 2) AS total_hours
FROM
  temp_table
GROUP BY
  member_casual
```



Query results

JOB INFORMATION | RESULTS | JSON | EXE

Row	member_casual	total_hours
1	member	710718.82
2	casual	787721.39

This data shows that even the casual customers are less, they use the bikes the most.

The following analysis done give a good understanding of the data and some of the patterns in the data. Now we'll switch to Tableau for visualization.