Vemo – Actionable Insights

1) How many trips are completed each day of the week?

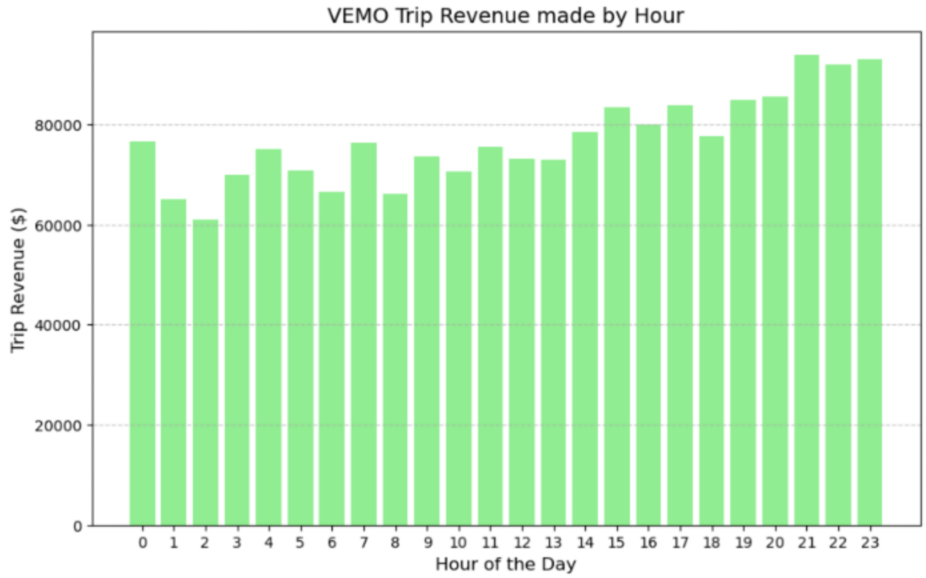
Analysing and refining the trips data to figure out the number of trips completed per day within the business week coming from 22/08/2022 05:00:00 to 29/08/2022 05:00:00, we can determine that next findings under pandas framework:

			id_week
trip_status	**completed**	**failed**	**fare_split**
Formatted_Day			
22 - Monday	2121.0	NaN	1.0
23 - Tuesday	2602.0	NaN	3.0
24 - Wednesday	2661.0	NaN	3.0
25 - Thursday	2486.0	NaN	3.0
26 - Friday	2475.0	1.0	4.0
27 - Saturday	2782.0	1.0	3.0
28 - Sunday	2647.0	NaN	4.0
29 - Monday	485.0	NaN	1.0

Based upon this data- frame we setup the next time series for trips completed: Monday-22-2,121, Tuesday-23-2,602, Wednesday-24-2,661, Thursday-25-2,486, Friday-26- 2,475, Saturday-27-2,782, Sunday-28-2,647 and Monday-29- 485 (for this data point we stand out that due to a cut in the current business week to start tracking another one new business week, we cannot see the total of data points displayed for this day, however, this is not bad it's just the way the data architecture of the company works out.). Moreover, if we sum up the total data points or trips completed for the business week mentioned **we got finally 18,259 trips completed**.

2) Which hour of the day generates the most revenue?

Was interesting apply different frameworks to localize the hour that generates the most revenues. If you guys open up the notebook where it's displayed all the methodology to solve this question, you'll see a heat-map that illustrates the hour that contains most of red-orange squares, same that graphically explain why broadly specking **every night at 21:00 PM, it's the time that generates the vast majority of the revenues (93,917 MXN approximately).** However, here you can watch a nice plot that points out this situation:



3) For the driver ID: 673 determine the mean time between trips (from one trip's drop_datetime, to the next trip's request_datetime).

This was one of the most challenging questions, given that I had to use different time codes to extract the difference between both dates getting like result the waiting time for the next driver request. Therefore we can securely establish that the average waiting time for the **ID-673 driver it's 9 minutes and 16 seconds**. This is a very critical KPI to tracking timing performance, because the lower the waiting time to request a driver the higher the revenues would be for the company due to a more dynamic and scalable service.

4) If you had to fire one driver, who would it be? Indicate the Id_driver and the reasons that support your decision.

Yes, implementing tough decisions can be critical for business performance, as long as they are supported by reliable data. Following the same approach as in the previous question, I developed a matrix to measure and extract the waiting time for each driver before getting a new ride.

The findings were astonishing. The data reveals that the **average waiting time for a new ride is 12 minutes and 59 seconds**. Expanding this analysis, I sorted the data in descending order to identify the driver with the highest waiting time—an inefficiency that significantly impacts productivity.

In this case, **Driver ID 970 emerged as the least efficient driver**, with an excessive waiting time of **59 minutes and 10 seconds**—almost an entire hour. This decision is not only backed by numerical evidence but also by a comparative analysis: the penultimate driver in the ranking is **12 minutes more efficient**, making Driver ID 970 **25% less productive**.

Notably, for the last five drivers (excluding Driver ID 970), the waiting time gap between them is less than three minutes, highlighting a relatively stable performance among them.

	id_driver	mean_time_between_trips
0	970	00:59:10
1	61	00:47:11
2	789	00:46:20
3	672	00:43:11
4	880	00:42:32
5	721	00:42:00
6	598	00:40:40
7	754	00:39:07

With this data-driven approach, the rationale behind my decision becomes clear.

Bonus track

Do you find any correlation between the parameters in the dataset that have caught
your eye?

Business Insights from Data Analysis: Maximizing Revenue & Efficiency

To better understand what drives revenue in VEMO ride platform, I analyzed the relationship between trip distance, time efficiency, and earnings. My goal was to uncover insights that could help optimize pricing, driver incentives, and operational efficiency.

Key Findings & Actionable Strategies

Longer trips generate more revenue

The data shows that the more kilometers a driver covers, the higher their earnings. This is a strong and proven relationship, meaning drivers and the platform should incentivize longer rides (e.g., through dynamic pricing, bonus structures, or targeted promotions for longer routes).

Time efficiency impacts earnings—but less significantly

• While making more trips per hour contributes to revenue, its effect is much smaller compared to trip distance.

• This suggests that maximizing trip volume isn’t as profitable as optimizing trip length.

• Operational improvements like reducing waiting time between rides and strategically positioning drivers in high-demand areas can help boost efficiency.

Better forecasting for pricing & incentives

We tested two predictive models:

1. A simple model based on trip distance, which explains almost **98% of revenue variation.**

2. A more refined model including trip distance and trips per hour, which still performs strongly at **94.6% accuracy.**

The second model helps us understand how changes in ride frequency and distance impact overall revenue, making it useful for adjusting commission structures and setting driver incentives.

Why this matters?

• Using a log-log model, we found that **a 1% increase in trip distance boosts revenue by 0.96%, while a 1% increase in trips per hour raises earnings by only 0.28%**.

• This confirms that drivers should focus on **optimizing trip length** rather than rushing to complete more rides.

Final Conclusion:

The key to higher earnings lies in optimizing trip distance rather than maximizing trip count—helping drivers, platforms, and pricing strategies align for maximum profitability.