# PATIENT21 ANALYTICS REPORT

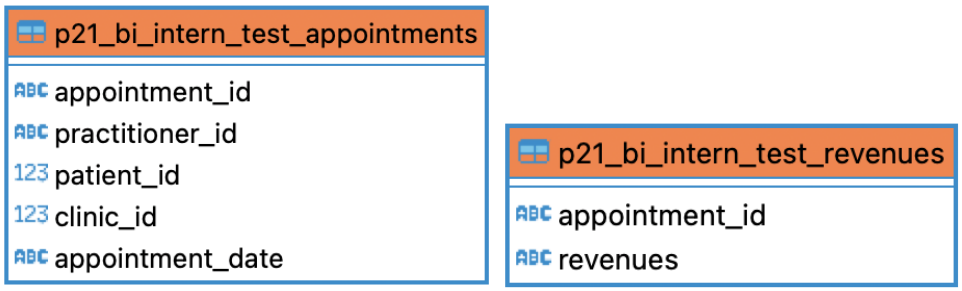
**By – Ayush Tankha (ESSEC Business School)**

## Introduction

I am provided with patient and revenue data from patient 21 data source. The task at hand requires me to analyze two csv files provided by the organization and estimate revenue for 2023 and unique patient acquisition.

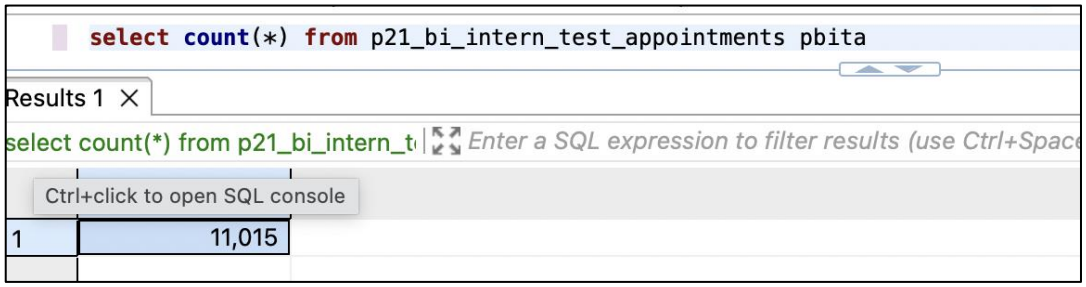
## Initial Observation

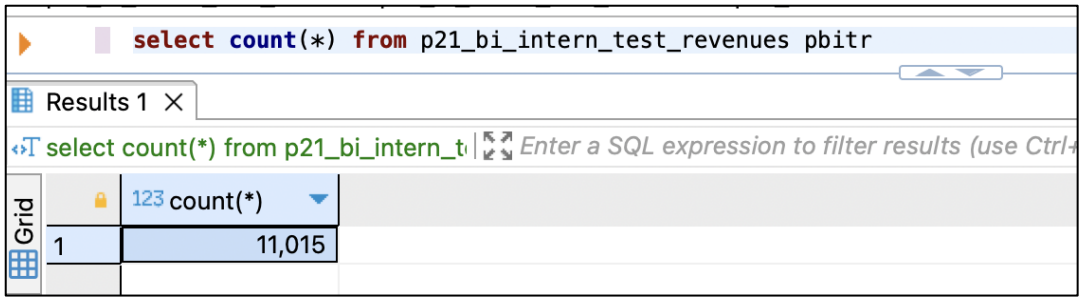
I started by ingesting the file into DBeaver for observing the schema of both the CSV files.




From above I noted that appointment_id is the **primary key** while it also acts as a **foreign key** to link both the tables.

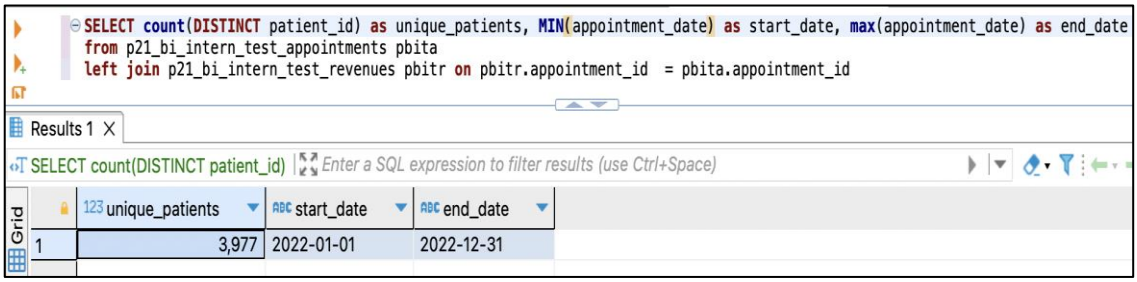
Before proceeding to join both the tables I checked the total number of rows to be conclude which type of join operator would be best suitable.



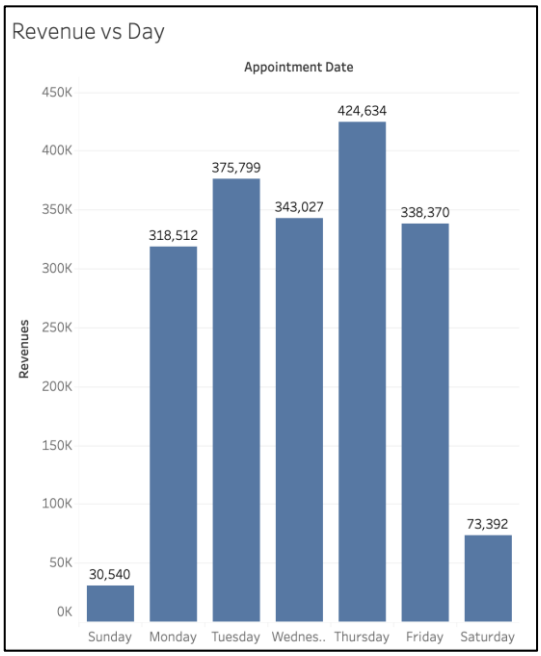



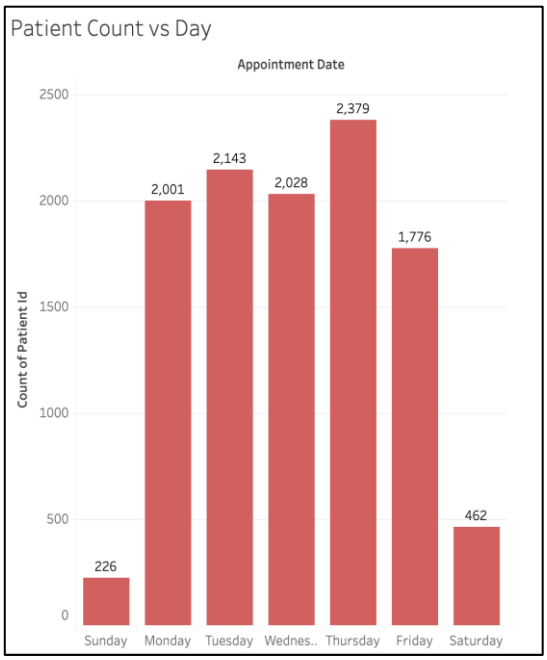
From above query outputs we can conclude that any type of join operator would suffice since both the tables have equal number rows and hence for each appointment_id in p21_by_intern_test_appointments table there is a corresponding appointment_id in p21_bi_intern_test_revenues table.

We observe a total number of **3977 unique patients** from a range of appointment dates from **1st Jan 2022 to 31st December 2022.**



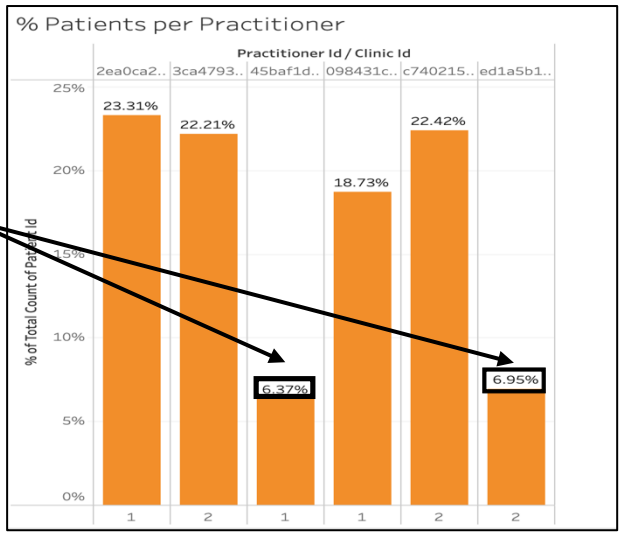

Now that we have the combined data, I visualized it using Tableau dashboard for better find any underlying patters from the data.




From above I observed that there is a significant drop in revenue during weekends, this was also supported by the fact that there was a drop in number of unique patients. This can also be since many clinics might be closed during the weekend.

On the right we can observe the % patients per practitioner, I observed that one practitioner in each clinic (1&2) has a significantly low number of patients. This can be explained due to that fact that these 2 practitioners attend the clinic during the weekend, and as we already observed from above analysis that there is a drop in footfall of patients. Concluding that practitioners with ID **"45baf1d2-e066-4f40-8fc5-c61c08d4af4c"** and **"ed1a5b13-3dc5-4212-b989-f59258b4410e"** sit during the weekend.

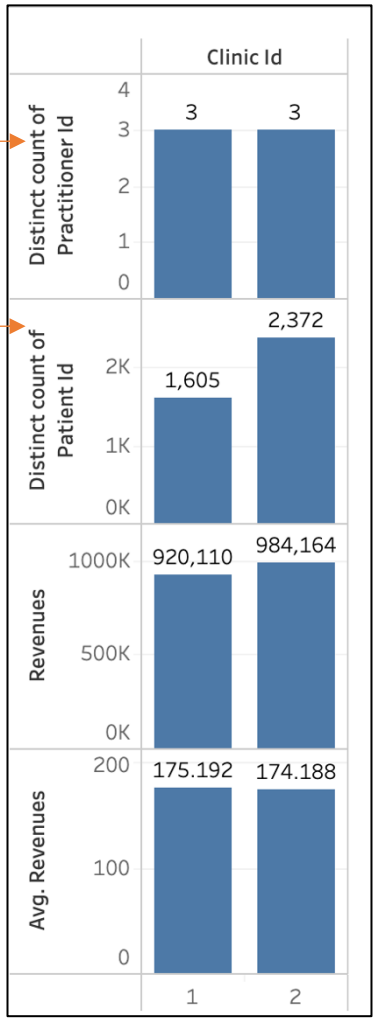



Moving forward I quickly took an overall look at some other metrics; in conclusion I made the following observations –

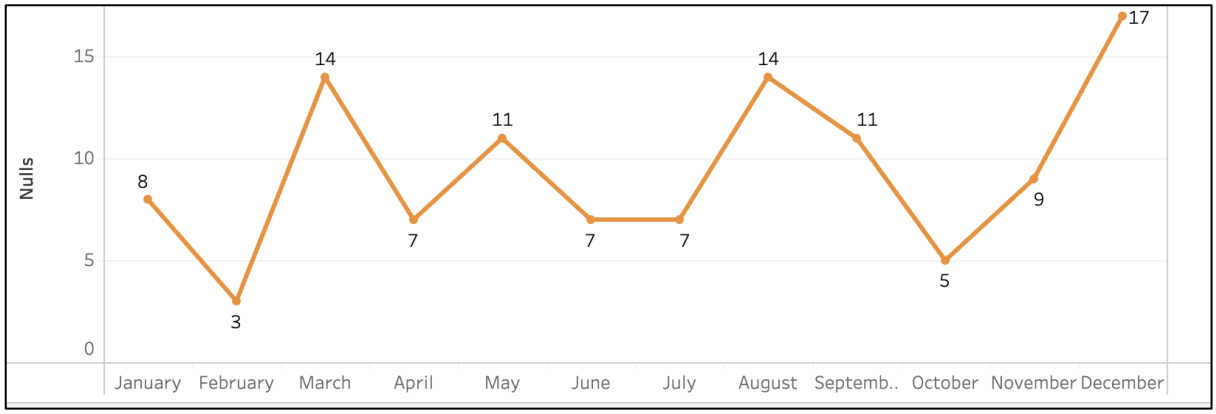
- Both the clinics have 3 distinct practitioners each.
- Clinic 2 has more unique patients visiting than Clinic 1, which is approximately 50% more.
- However, the total revenue and average revenue tell us a different story. The total revenue and average revenue being almost comparable we can conclude that the patients revisit more often to clinic 1 as compared to clinic 2.

This could have varying interpretations, some of them are below –

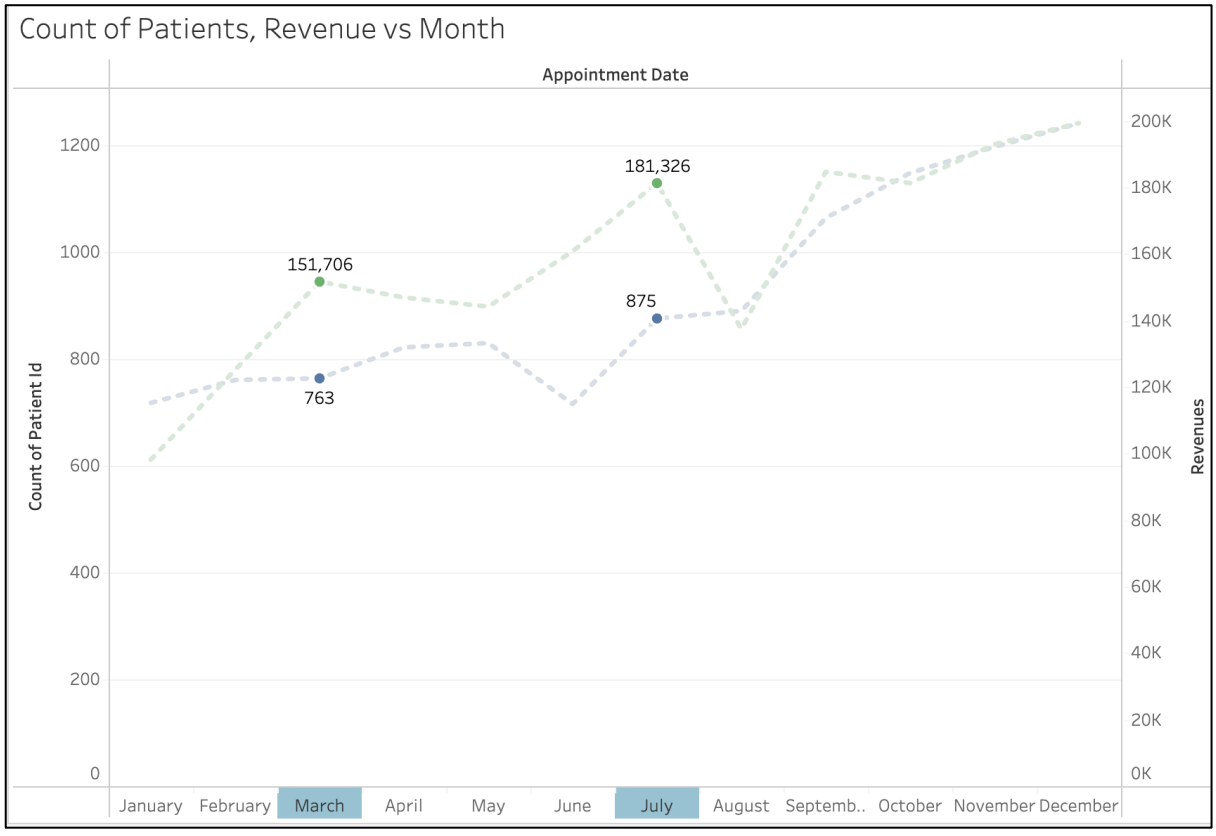
- Practitioners from Clinic 1 show better bonding with their patients.
- Clinic 1 could be dealing with **more patients going through a chronic disease** which requires repeat visits while Clinic 2 could be related to patients with a **more general diseases** like flu which may require less frequent visits for treatment.
- **Clinic 1 can be in outskirts of the city** where unique patients are less but frequent since the local areas have less clinics. However, **Clinic 2 on the other hand can be in major cities** where there can be high number of unique patients and since there are already a lot of clinics in major cities, patients can prefer to switch clinics easily, therefore less frequent visits.



I also observed some null values in the revenue columns, so I plotted them below to look if the number is significant enough or not. From observation there are **113 null values** out of a total **11,015 (already calculated above**). This implies only a marginal **0.01% are null** hence it may not account for a huge significant difference.



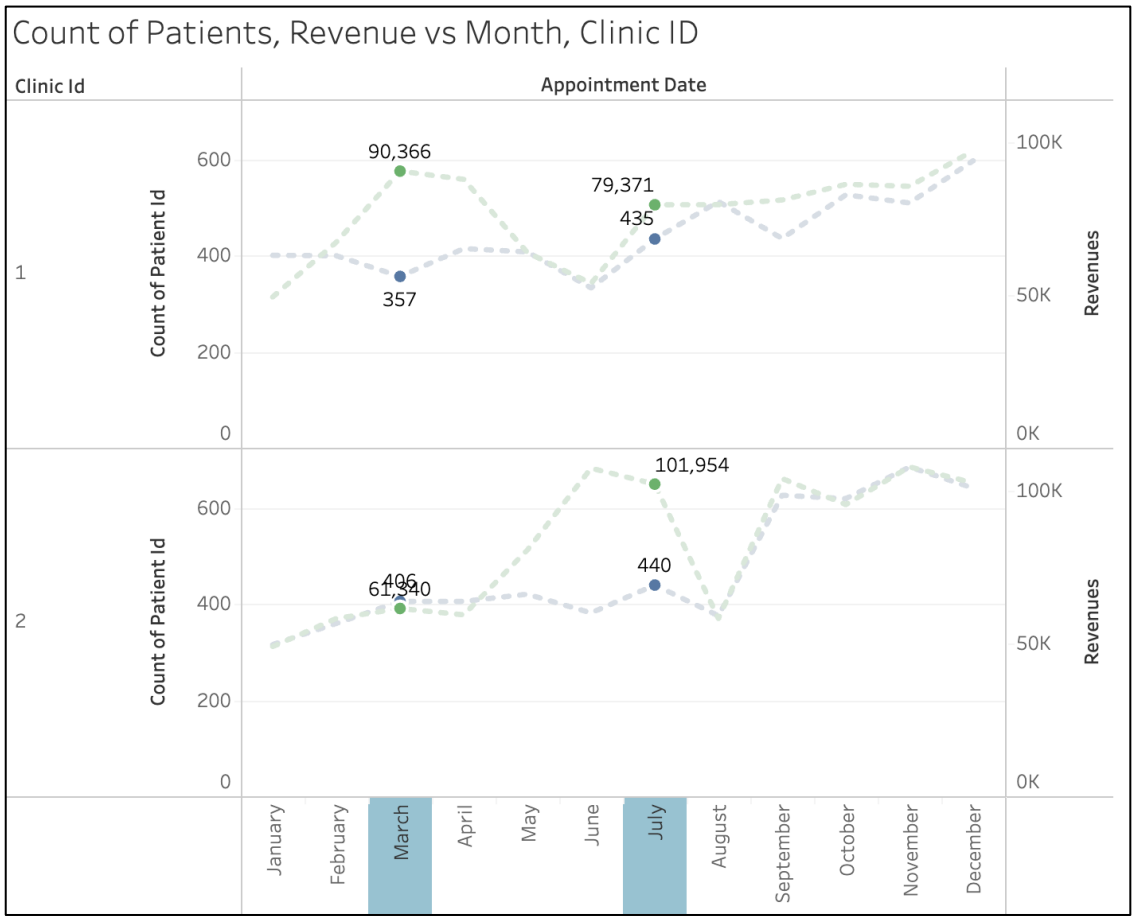

From below time series curve I can see that there is a huge gap between the number of patients and revenue incurred from those patients for the month of March and July.



I reached for the following conclusion for this huge gap –

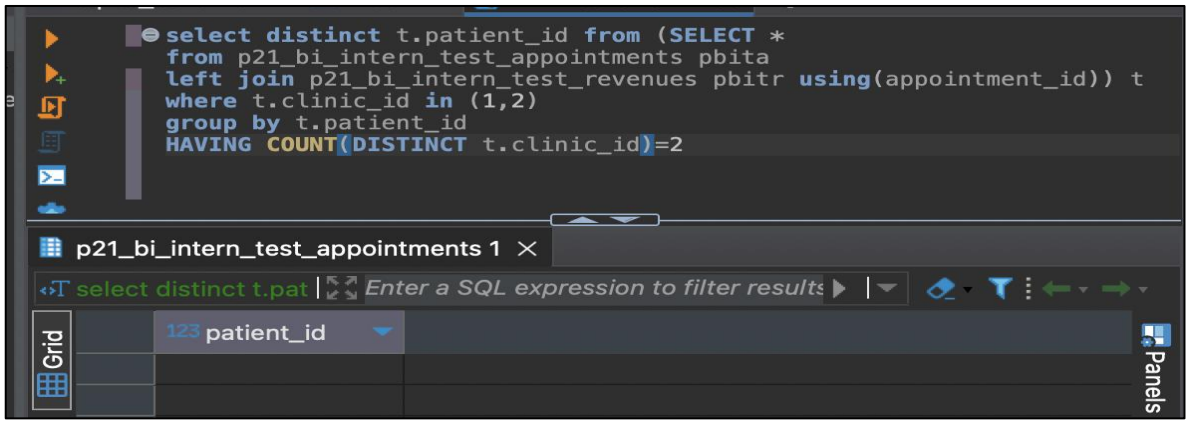
- Medical procedures involved during March and July tend to be expensive.
- Initiating two more clinics during this time can meet the additional demand expected in 2023.
- **Notice there is a sudden drop in revenue** even after each of these big gaps, this can explain because perhaps after a big procedure patient who revisit only go back for minor weekly/monthly checkups.
- Another reason for the drop in August can be due to vacation. **Holidays and festivities usually result in extreme or anomaly trends.**

Further exploring this curve for the respective clinics, I obtained the following graph –




From above graph I realized that **Clinic Id 1 has the most potential for revenue per count of patients during the month of March** whereas **For Clinic Id 2 the peak comes in July.**

Before proceeding to make assumptions for our model we also make an interesting observation by running the following SQL query -




There were **no patients that visited both clinics** 1 and 2, this only **implies that the clinics don't compete internally.**

This leads to the following assumptions that are taken into account for modelling.

**Basic Assumptions for Modelling**

- Assuming Clinic Id 1 represents clinic in city outskirts. **(reason already mentioned previously in my analysis)** (**we will call this Type 1 clinic)**.
- Assuming Clinic Id 2 represents clinic in central city. **(reason already mentioned previously in my analysis)** (**we will call this Type 2 clinic).**
- I also assume that the company will try strategizing in a way that **maximize profits.**
- Assuming that the two new clinics being opened in 2023 will be of similar nature to Clinic Id 1 (Type 1) or to Clinic Id 2 (Type 2).

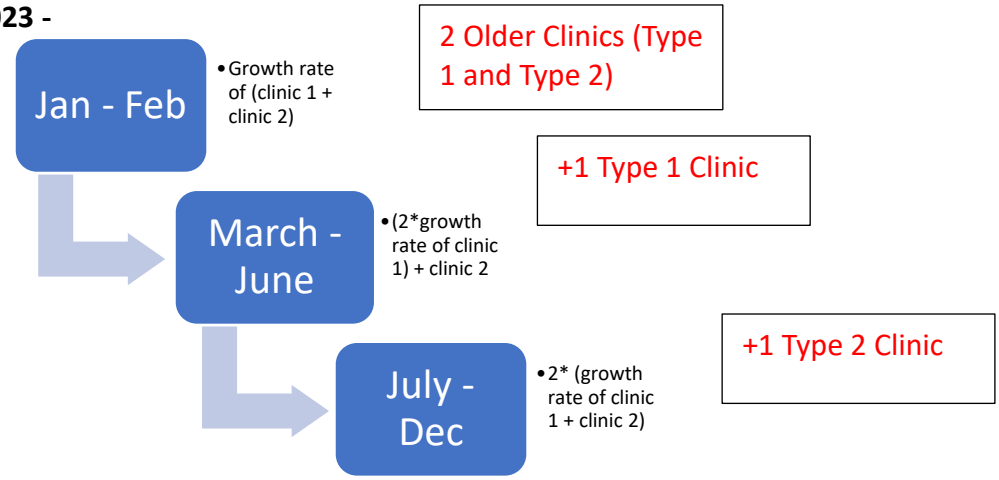
Note – **Type 1 means clinic in outskirts region, Type 2 means clinic in central region**

**For more clarity refer to below table –**

| Type 1 | Type 2 |
|---|---|
| Outskirts Region | Central Region |
| Less unique patients | More unique patients |
| More revisiting patients | Less revisiting patients |

From above assumptions we can conclude that the best way to strategize for maximum revenue generation would be **opening a Type 1 clinic in March and Type 2 clinic in July. (This is due to the observed gaps explained in the previous analysis)**

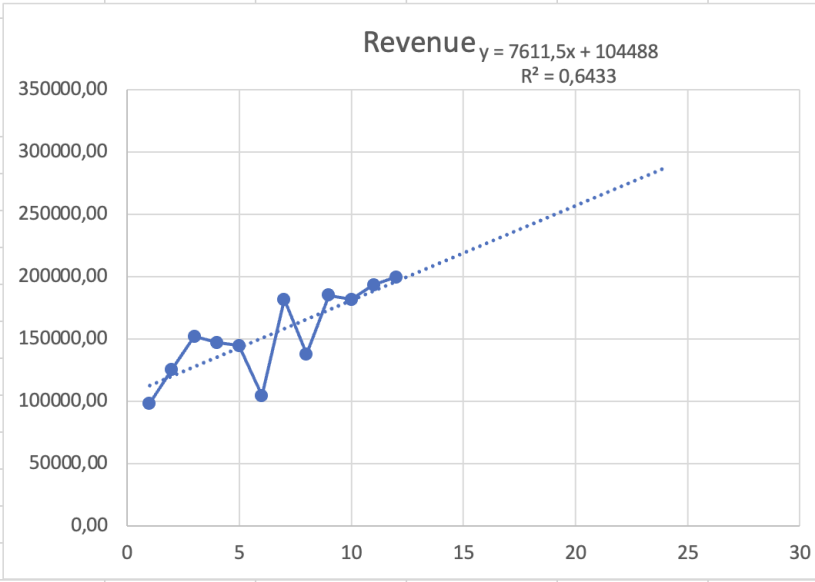
**From all the above analysis I want to now present the structure of my real scenario that will play out in 2023 -**




## PART 1 – REVENUE FORECAST 2023

First we try to forecast for the months of "Jan-Feb" , for this I calculated the sum of revenue accumulated for each month and plotted a line plot in excel.

| Month | Revenue |
|---|---|
| 1 | 98149,78 |
| 2 | 125227,64 |
| 3 | 151705,88 |
| 4 | 146936,54 |
| 5 | 144192,08 |
| 6 | 104117,89 |
| 7 | 181325,57 |
| 8 | 137464,57 |
| 9 | 184714,88 |
| 10 | 181327,73 |
| 11 | 193002,82 |
| 12 | 199388,97 |




From above, I tried to fit a linear trend line for simplicity and got an **$R^2$ value = 0.6433** showing an average confidence in our trendline. The trendline forecasts in the future up until 24 months (end of Dec 2023)

Since we now have the equation for regression, we easily calculate the revenue generated in Jan and Feb for 2 clinics and then simply double the revenue it to get the revenues for 4 clinics.

| Month | Single Forecasted Growth Rate (2 clinics) | Double Forecasted Growth Rate (4 clinics) |
|---|---|---|
| 13 | 203437,5 | Not Required |
| 14 | 211049 | Not Required |
| 19 | 249106,5 | 498213 |
| 20 | 256718 | 513436 |
| 21 | 264329,5 | 528659 |
| 22 | 271941 | 543882 |
| 23 | 279552,5 | 559105 |
| 24 | 287164 | 574328 |
| | | |
| Total | 414486,5 | 3217623 |
| | Jan - Feb | July - Dec |

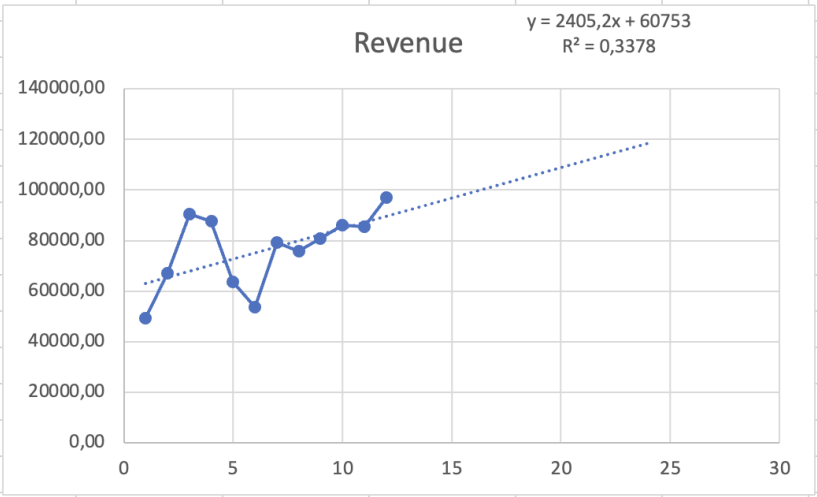
So above **month 13 , 14 correspond to Jan and Feb** with 2 clinics (Type 1 and Type 2) **Month 19,20,21,22,23,24 correspond to months from July to December**. Since we have 4 clinics and we already assumed their behavior to be same (2*Type 1 and 2*Type 2), so we simply double the revenue.

Further we add the corresponding values highlighted in yellow above to get the total in each corresponding periods of time.

Now comes the tricky part, to calculate the revenue from March to June where we have 2*Type 1 clinics and 1*Type 2.

So for this we separate the data of Type 1 clinic individually and forecast its trend line, we repeat the same process but this time we get a **new regression line for the single Type 1** clinic. Now we just simply add the revenue from this forecast of Type 1 to our already forecasted Revenues for the same period for Type 1 and Type 2.

| Month | Revenue |
|---|---|
| 1 | 49252,01 |
| 2 | 67169,45 |
| 3 | 90365,66 |
| 4 | 87678,27 |
| 5 | 63757,20 |
| 6 | 53689,00 |
| 7 | 79371,47 |
| 8 | 75943,83 |
| 9 | 80940,53 |
| 10 | 86047,62 |
| 11 | 85419,24 |
| 12 | 97008,34 |




***Note here $R^2$ value is low indicating less confidence in our forecast this time**

Keep in mind we have already calculated the regression line for the combined Type 1 and Type 2, we are just simple finding the new regression line for the single Type 1 and then adding the results.

| Month | Revenue |
|---|---|
| 15 | 96831 |
| 16 | 99236,2 |
| 17 | 101641,4 |
| 18 | 104046,6 |
| Total | 401755,2 |
| | Mar - June |

**Now adding all revenues, we get the final forecast revenue for 2023 as**

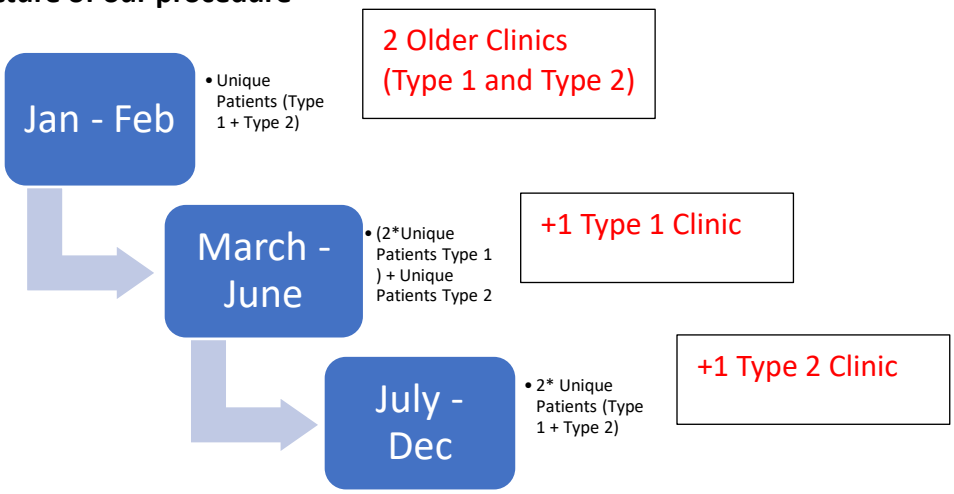
**414,486.5 + 401,755.2 + 3,217,623 = 4,033,864.7**
**(Jan-Feb) + (Mar -June) + (July – Dec)**

**Forecast**

## PART 2 – UNIQUE PATIENTS FORECAST 2023

Since we already know that all unique patients go only to one of the 2 clinics, we are going to model the forecast for Type 1 and Type 2 clinics separately and then add them over their corresponding overlapping time frames.

**Basic skeletal structure of our procedure –**




**I extract data with number of unique patients visiting the clinics for each day for Clinic 1 and Clinic 2 using SQL.**

```sql
SELECT
    first_visit_date,
    COUNT(patient_id) AS unique_patient_count
FROM
    (SELECT
        MIN(appointment_date) AS first_visit_date,
        patient_id
     FROM
        (SELECT *
from p21_bi_intern_test_appointments pbita
left join p21_bi_intern_test_revenues pbitr using(appointment_id)
where clinic_id = 1
)
     GROUP BY
        patient_id) AS first_visits
GROUP BY
    first_visit_date
ORDER BY
    first_visit_date;
```

Results 1

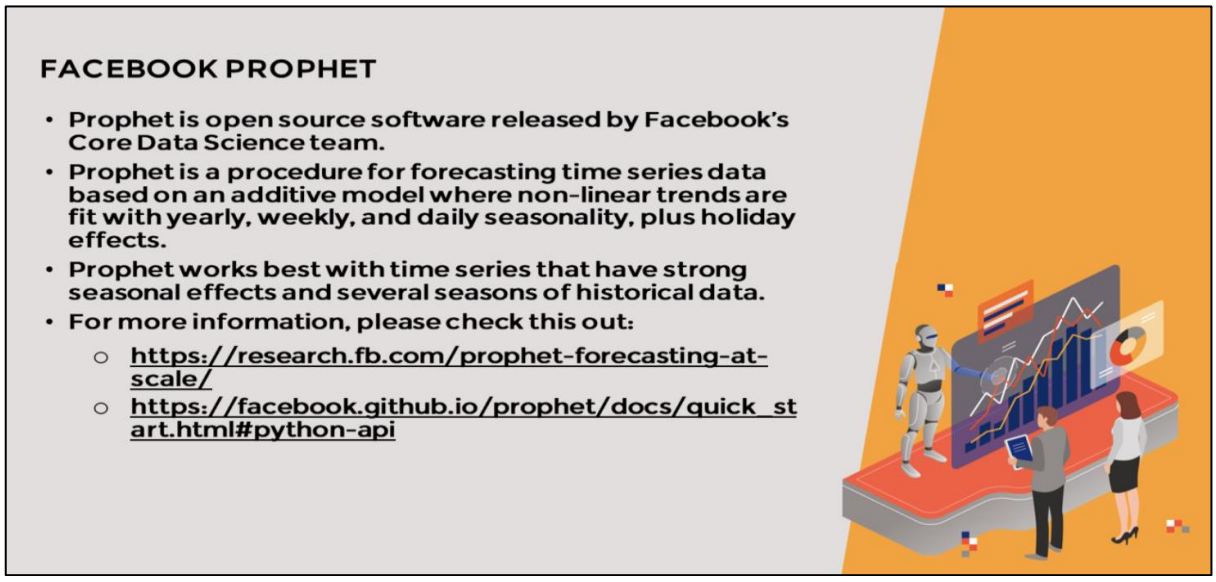
SELECT first_visit_date, COUNT(pati | Enter a SQL expression to filter results (use Ctrl+Sp

| | first_visit_date | unique_patient_count |
|---|---|---|
| 1 | 2022-01-01 | 31 |
| 2 | 2022-01-06 | 11 |
| 3 | 2022-01-07 | 11 |
| 4 | 2022-01-08 | 12 |
| 5 | 2022-01-09 | 10 |

***Note – I just change the value of clinic_id = 2 above, to get the other dataset for forecast**

Now that I have obtained both the datasets for Type 1 and Type 2 clinics for unique patient visits for each day, I proceed to use a forecasting model. This time however I will try a new method for forecast which will be using Facebook Prophet.

## UNDERSTANDING FACEBOOK PROPHET

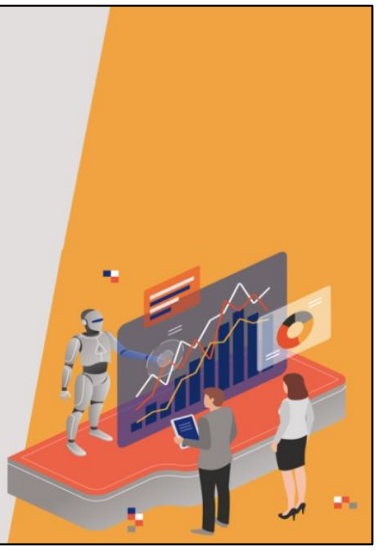


## FACEBOOK PROPHET

- Prophet implements an additive regression model with four elements:
  - A piecewise linear, Prophet automatically picks up change points in the data and identifies any change in trends.
  - A yearly seasonal component modeled using Fourier series.
  - A weekly seasonal component.
  - A holiday list that can be manually provided.
- Additive Regression model takes the form:

$$Y = \beta_0 + \sum_{j=1}^{p} f_j(X_j) + \epsilon$$

- The functions $f_j(x_j)$ are unknown smoothing functions fit from the data
- Reference: https://research.fb.com/prophet-forecasting-at-scale/



## FACEBOOK PROPHET

### ACCURATE AND FAST

- Facebook teams uses Prophet for accurate forecasting and planning.
- Prophet can generate results in seconds.

### AUTOMATIC

- No need to perform data preprocessing.
- Prophet works with missing data with several outliers.

### DOMAIN KNOWLEDGE INTEGRATION

- Users can tweak forecast by manually adding domain specific knowledge.

```
] future = model_1.make_future_dataframe(periods = 365)
  forecast = model_1.predict(future)
  figure = model_1.plot(forecast, xlabel = 'Date', ylabel='Unique Patients')
  figure2 =  model_1.plot_components(forecast)
```

From above I set the periods of days for prediction to be 365 since we want to predict for the whole year of 2023 and then forecasted on each of Type 1 clinic and Type 2 clinic data extracted before by SQL. I obtained the following forecasted data.

forecast_1

| | ds | yhat |
|---|---|---|
| 247 | 2023-01-01 | 4.694027 |
| 248 | 2023-01-02 | 6.307115 |
| 249 | 2023-01-03 | 6.327044 |
| 250 | 2023-01-04 | 5.470821 |
| 251 | 2023-01-05 | 5.832664 |
| ... | ... | ... |
| 607 | 2023-12-27 | 4.865583 |
| 608 | 2023-12-28 | 5.227426 |
| 609 | 2023-12-29 | 4.039445 |
| 610 | 2023-12-30 | 5.245410 |
| 611 | 2023-12-31 | 4.076921 |

365 rows × 2 columns

This forecast is for Type 1 clinic data however **I repeated the same steps to obtain the forecast for Type 2 clinic data.** Further I will **round off the forecasted value** to nearest integer since the unique count of patients cannot be in decimals.

After rounding off the values and adding them to find the total I got the following sum (for 365 days), I repeated the same process for Type 2 and both results are below –

```
sum(forecast_1['yhat'])
```

```
1913.0
```

Type 1

```
sum(forecast_2['yhat'])
```

```
6158.0
```

Type 2

Now we just extract the data according to our initial skeletal structure and combine the results.

On the next page are 3 python functions I created to extract the sum of unique patients from the following 3 periods –

- **Jan to Feb (One Type 1 + One Type 2)**
- **Mar to June (Two Type 1 + One Type 2)**
- **July to Dec (Two Type 1 + Two Type 2)**

Forecast

**We obtain a final forecast of 13249 new unique patients in 2023.**
**(Refer to code and outputs below for better understanding)**

Jan to Feb

```
def jan_to_feb(df):
  df['ds'] = pd.to_datetime(df['ds'])

# Filter the DataFrame for January and February
  jan_feb_df = df[df['ds'].dt.month.isin([1, 2])]

# Calculate the sum of the 'yhat' column for these months
  total_sum = jan_feb_df['yhat'].sum()

# Print or return the result
  return total_sum

period_1 = jan_to_feb(forecast_1)+jan_to_feb(forecast_2)
```

```
[71] print(f'total unique patients in jan-feb = {period_1}')

total unique patients in jan-feb = 1055.0
```

Mar to June

```
def mar_to_june(df):
  df['ds'] = pd.to_datetime(df['ds'])

# Filter the DataFrame for March to June
  mar_june_df = df[df['ds'].dt.month.isin([3, 4,5,6])]

# Calculate the sum of the 'yhat' column for these months
  total_sum = mar_june_df['yhat'].sum()

# Print or return the result
  return total_sum

period_2 =2* mar_to_june(forecast_1)+mar_to_june(forecast_2)
```

```
[73] print(f'total unique patients in mar - june = {period_2}')

total unique patients in mar - june = 3152.0
```

Sum = 1 year of 2023

```
total forecasted unique patients for 2023 = 13249.0
```

Jul to Dec

```
[74] def jul_to_dec(df):
  df['ds'] = pd.to_datetime(df['ds'])

# Filter the DataFrame for July to December
  jul_dec_df = df[df['ds'].dt.month.isin([7,8,9,10,11,12])]

# Calculate the sum of the 'yhat' column for these months
  total_sum = jul_dec_df['yhat'].sum()

# Print or return the result
  return total_sum

period_3 = 2* jul_to_dec(forecast_1)+ 2*jul_to_dec(forecast_2)
```

```
[75] print(f'total unique patients in mar - june = {period_3}')

total unique patients in mar - june = 9042.0
```