

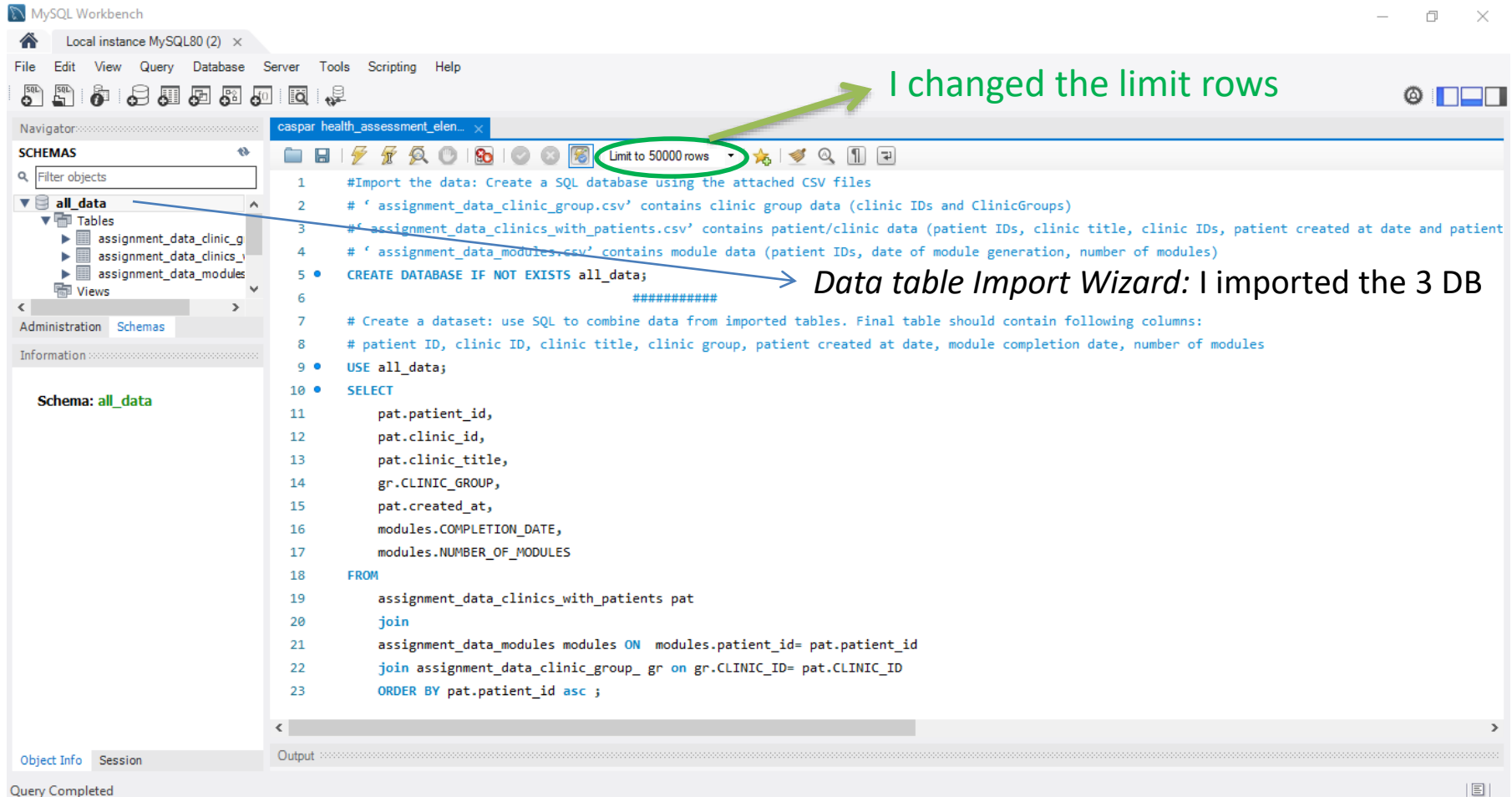
Elena Peña

Caspar Health

SQL and Tableau Assignment

SQL(1): Code

Difficulties: *Import wizard* couldn't read well the csv-files. There were problems with the encoding, so I went back to Excel reader and inside of "Save as": "Excel options" I changed the file type and "save files in web service". It did work!!



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'all_data' selected. The main editor shows a SQL query titled 'caspar health_assessment_elen...'. A green circle highlights the 'Limit to 50000 rows' button in the toolbar, with a green arrow pointing to it and the text 'I changed the limit rows'. A blue arrow points from the 'all_data' schema in the sidebar to the 'CREATE DATABASE IF NOT EXISTS all_data;' line in the query, with the text 'Data table Import Wizard: I imported the 3 DB' next to it. The SQL query is as follows:

```
1 #Import the data: Create a SQL database using the attached CSV files
2 # ' assignment_data_clinic_group.csv' contains clinic group data (clinic IDs and ClinicGroups)
3 # ' assignment_data_clinics_with_patients.csv' contains patient/clinic data (patient IDs, clinic title, clinic IDs, patient created at date and patient
4 # ' assignment_data_modules.csv' contains module data (patient IDs, date of module generation, number of modules)
5 • CREATE DATABASE IF NOT EXISTS all_data;
6 #####
7 # Create a dataset: use SQL to combine data from imported tables. Final table should contain following columns:
8 # patient ID, clinic ID, clinic title, clinic group, patient created at date, module completion date, number of modules
9 • USE all_data;
10 • SELECT
11     pat.patient_id,
12     pat.clinic_id,
13     pat.clinic_title,
14     gr.CLINIC_GROUP,
15     pat.created_at,
16     modules.COMPLETION_DATE,
17     modules.NUMBER_OF_MODULES
18 FROM
19     assignment_data_clinics_with_patients pat
20     join
21     assignment_data_modules modules ON modules.patient_id= pat.patient_id
22     join assignment_data_clinic_group gr on gr.CLINIC_ID= pat.CLINIC_ID
23 ORDER BY pat.patient_id asc ;
```

SQL(2): results

MySQL Workbench

Local instance MySQL80 (2) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

all_data

Tables

- assignment_data_clinic_group_
- assignment_data_clinics_with_patients
- assignment_data_modules

Views

Stored Procedures

Administration Schemas

Information

Schema: all_data

caspar health_assessment_elec...

Limit to 50000 rows

```
17 modules.NUMBER_OF_MODULES
18 FROM
19 assignment_data_clinics_with_patients pat
20 join
21 assignment_data_modules modules ON modules.patient_id= pat.patient_id
22 join assignment_data_clinic_group_ gr on gr.CLINIC_ID= pat.CLINIC_ID
23 ORDER BY pat.patient_id asc ;
```

Result Grid

	patient_id	clinic_id	clinic_title	CLINIC_GROUP	created_at	COMPLETION_DATE	NUMBER_OF_MODULES
▶	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	18/10/2020	1
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	19/10/2020	1
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	24/10/2020	1
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	25/10/2020	1
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	28/10/2020	2
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	04/11/2020	2
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	11/11/2020	2
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	18/11/2020	2
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	25/11/2020	2
	3764	abc9	Rehab on the go	Re-Freshed	04/06/2018	02/12/2020	2

Result 2 x

Output

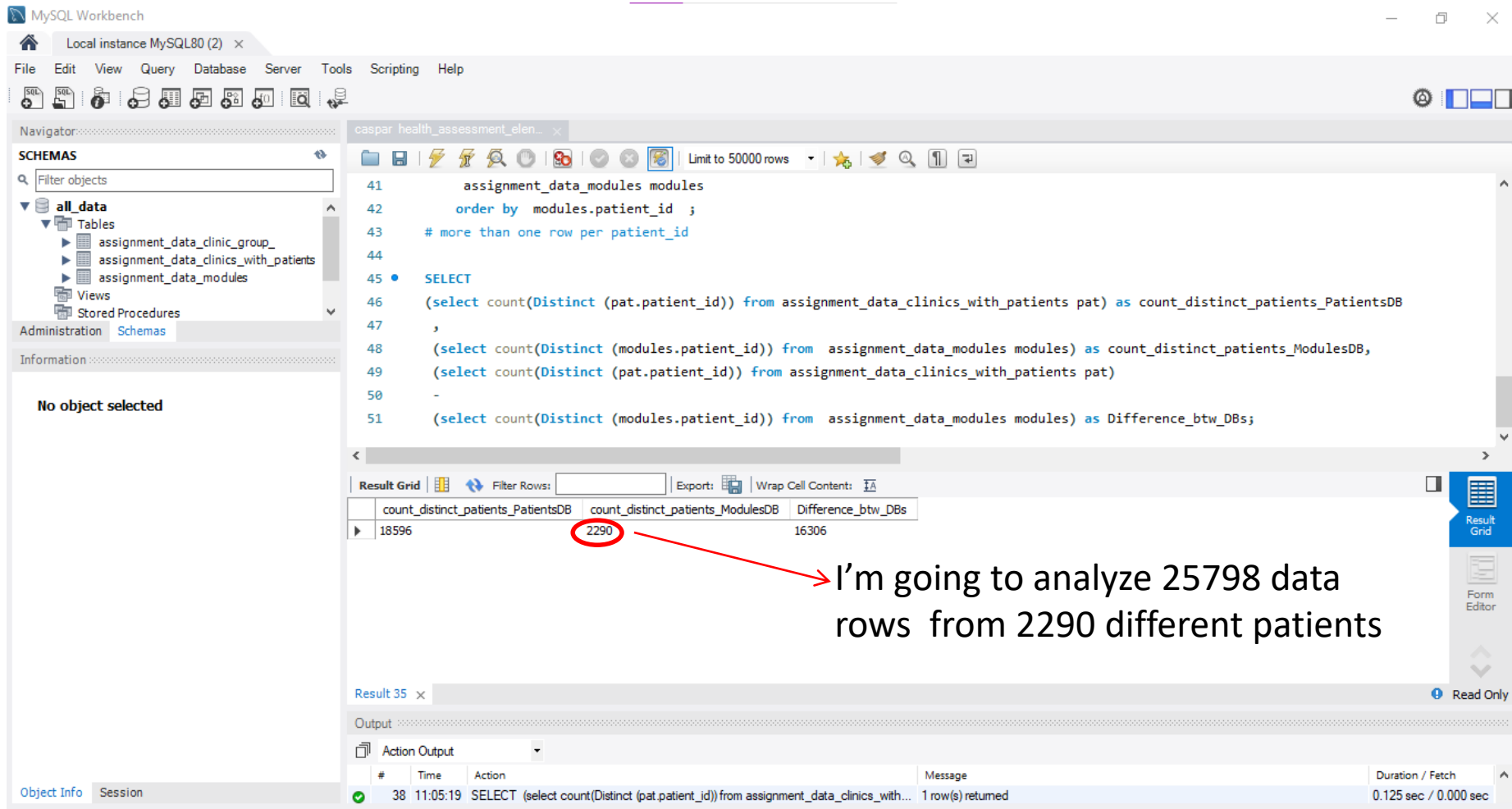
Action Output

#	Time	Action	Message	Duration / Fetch
✓ 27	06:55:57	USE all_data	0 row(s) affected	0.000 sec
✓ 28	06:55:57	SELECT pat.patient_id, pat.clinic_id, pat.clinic_title, gr.CLINIC...	25798 row(s) returned	0.312 sec / 0.079 sec

Object Info Session

Query Completed

SQL(3): Data Exploration



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'assignment_data_clinic_group_', 'assignment_data_clinics_with_patients', and 'assignment_data_modules'. The main editor window shows a SQL query for 'caspar health_assessment_elen...'. The query is as follows:

```
41 assignment_data_modules modules
42 order by modules.patient_id ;
43 # more than one row per patient_id
44
45 • SELECT
46 (select count(Distinct (pat.patient_id)) from assignment_data_clinics_with_patients pat) as count_distinct_patients_PatientsDB
47 ,
48 (select count(Distinct (modules.patient_id)) from assignment_data_modules modules) as count_distinct_patients_ModulesDB,
49 (select count(Distinct (pat.patient_id)) from assignment_data_clinics_with_patients pat)
50 -
51 (select count(Distinct (modules.patient_id)) from assignment_data_modules modules) as Difference_bt看_DBs;
```

Below the query editor, the 'Result Grid' is visible, showing the results of the query. The grid has three columns: 'count_distinct_patients_PatientsDB', 'count_distinct_patients_ModulesDB', and 'Difference_bt看_DBs'. The first row shows values 18596, 2290, and 16306. A red circle highlights the value 2290, and a red arrow points from it to the text 'I'm going to analyze 25798 data rows from 2290 different patients'.

count_distinct_patients_PatientsDB	count_distinct_patients_ModulesDB	Difference_bt看_DBs
18596	2290	16306

Result 35 x

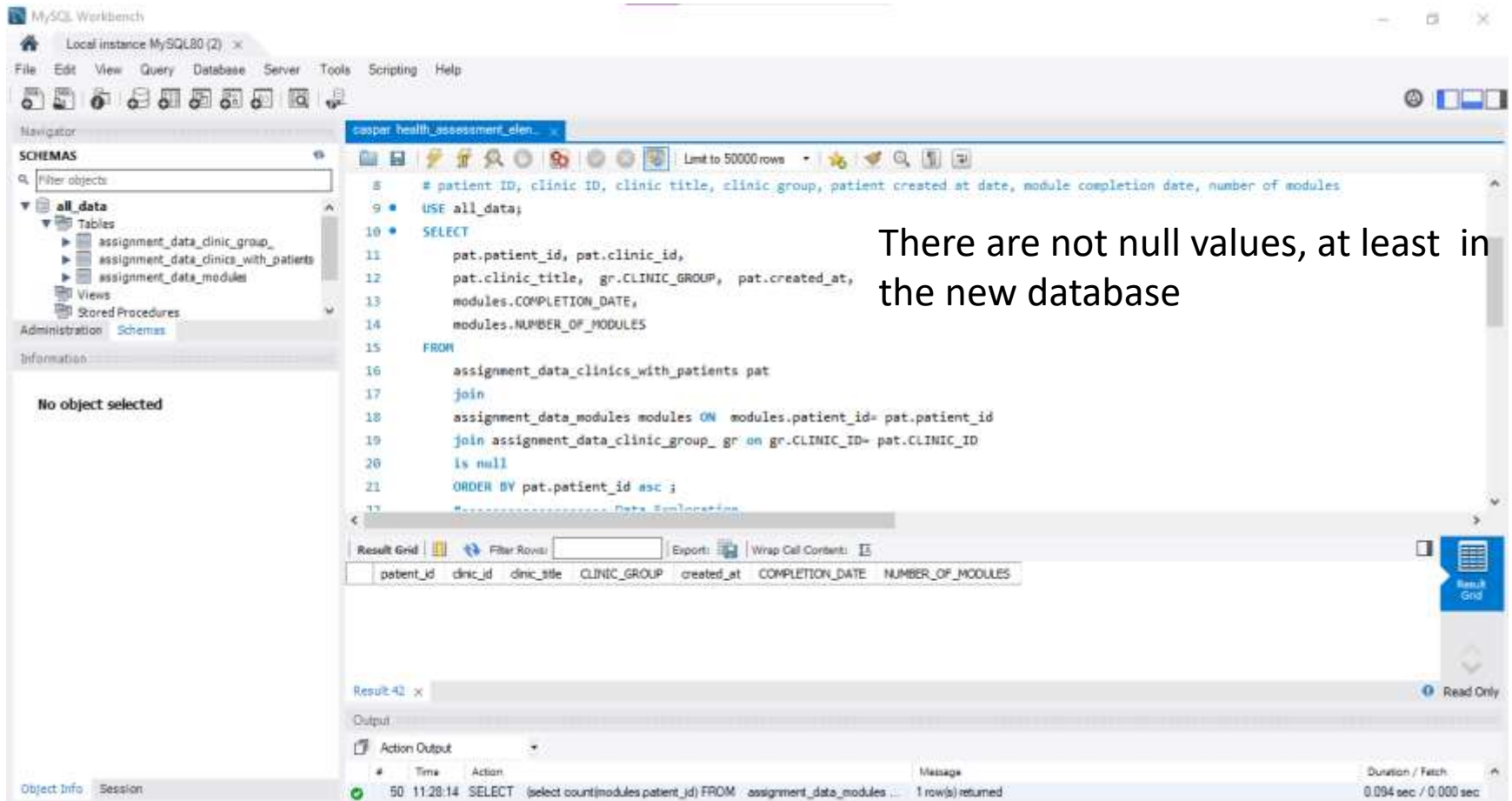
Output

Action Output

#	Time	Action	Message	Duration / Fetch
38	11:05:19	SELECT (select count(Distinct (pat.patient_id)) from assignment_data_clinics_with...	1 row(s) returned	0.125 sec / 0.000 sec

SQL(5): Data Exploration

Missing values



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a database named 'all_data' with several tables. The main editor displays a SQL query that joins three tables: 'assignment_data_clinics_with_patients', 'assignment_data_modules', and 'assignment_data_clinic_group_gr'. The query selects patient ID, clinic ID, clinic title, clinic group, patient created at date, module completion date, and number of modules. The results are ordered by patient ID. Below the query, the 'Result Grid' shows the first row of data. The status bar at the bottom indicates that 1 row(s) were returned in 0.094 seconds.

There are not null values, at least in the new database

```
8  # patient ID, clinic ID, clinic title, clinic group, patient created at date, module completion date, number of modules
9  USE all_data;
10 SELECT
11     pat.patient_id, pat.clinic_id,
12     pat.clinic_title, gr.CLINIC_GROUP, pat.created_at,
13     modules.COMPLETION_DATE,
14     modules.NUMBER_OF_MODULES
15 FROM
16     assignment_data_clinics_with_patients pat
17 JOIN
18     assignment_data_modules modules ON modules.patient_id= pat.patient_id
19 JOIN assignment_data_clinic_group_gr on gr.CLINIC_ID= pat.CLINIC_ID
20 IS NULL
21 ORDER BY pat.patient_id asc ;
22
```

patient_id	clinic_id	clinic_title	CLINIC_GROUP	created_at	COMPLETION_DATE	NUMBER_OF_MODULES

Result 42 x

Output

Action Output

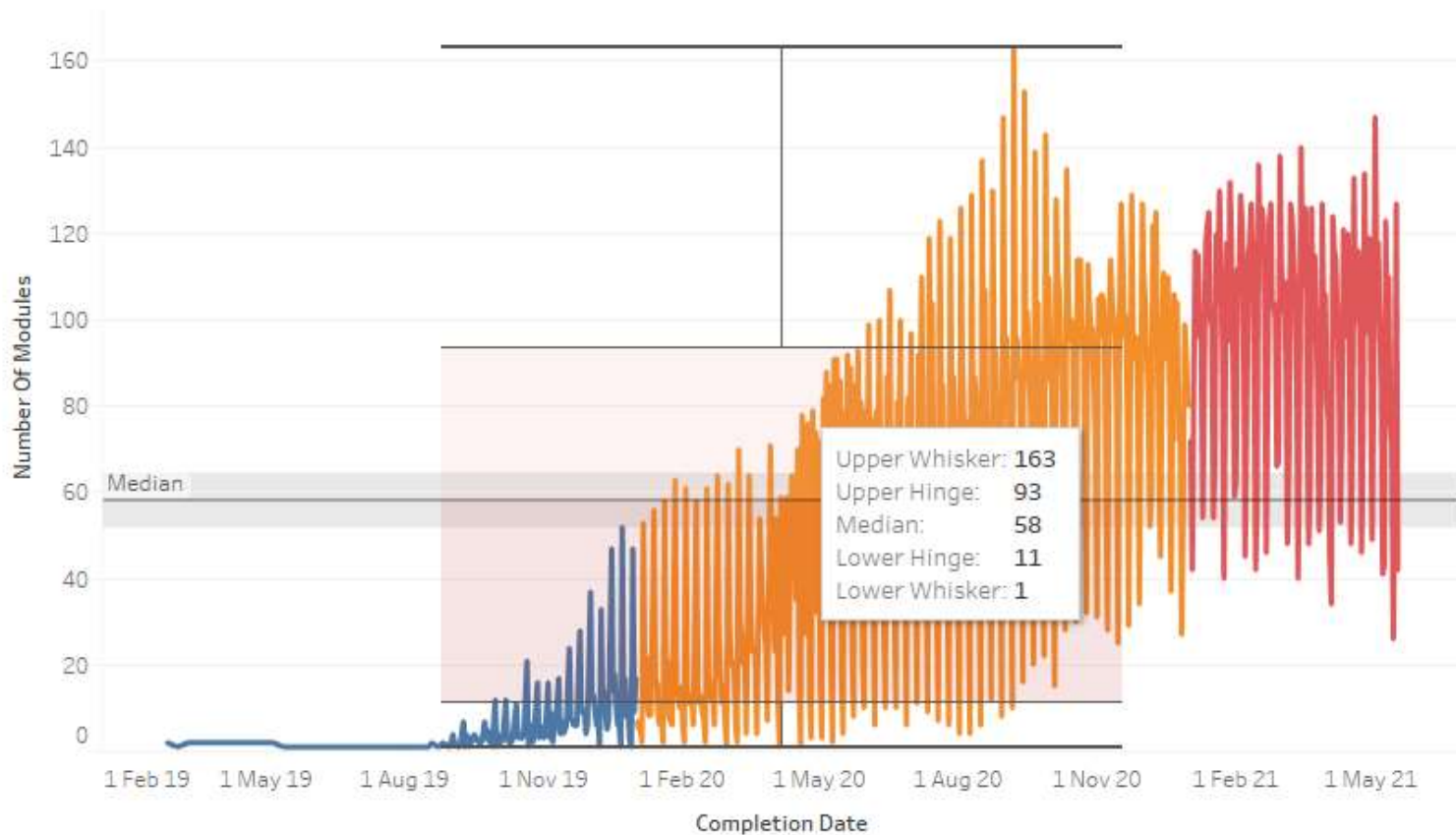
Time Action Message Duration / Fetch

50 11:28:14 SELECT (select count(modules.patient_id) FROM assignment_data_modules ...) 1 row(s) returned 0.094 sec / 0.000 sec

Tableau

Outliers: number of completed modules

Outliers-Completed Modules



MONTH(Completion D...

- ☒ (All)
- ☒ January
- ☒ February
- ☒ March
- ☒ April
- ☒ May
- ☒ June
- ☒ July
- ☒ August
- ☒ September
- ☒ October
- ☒ November
- ☒ December

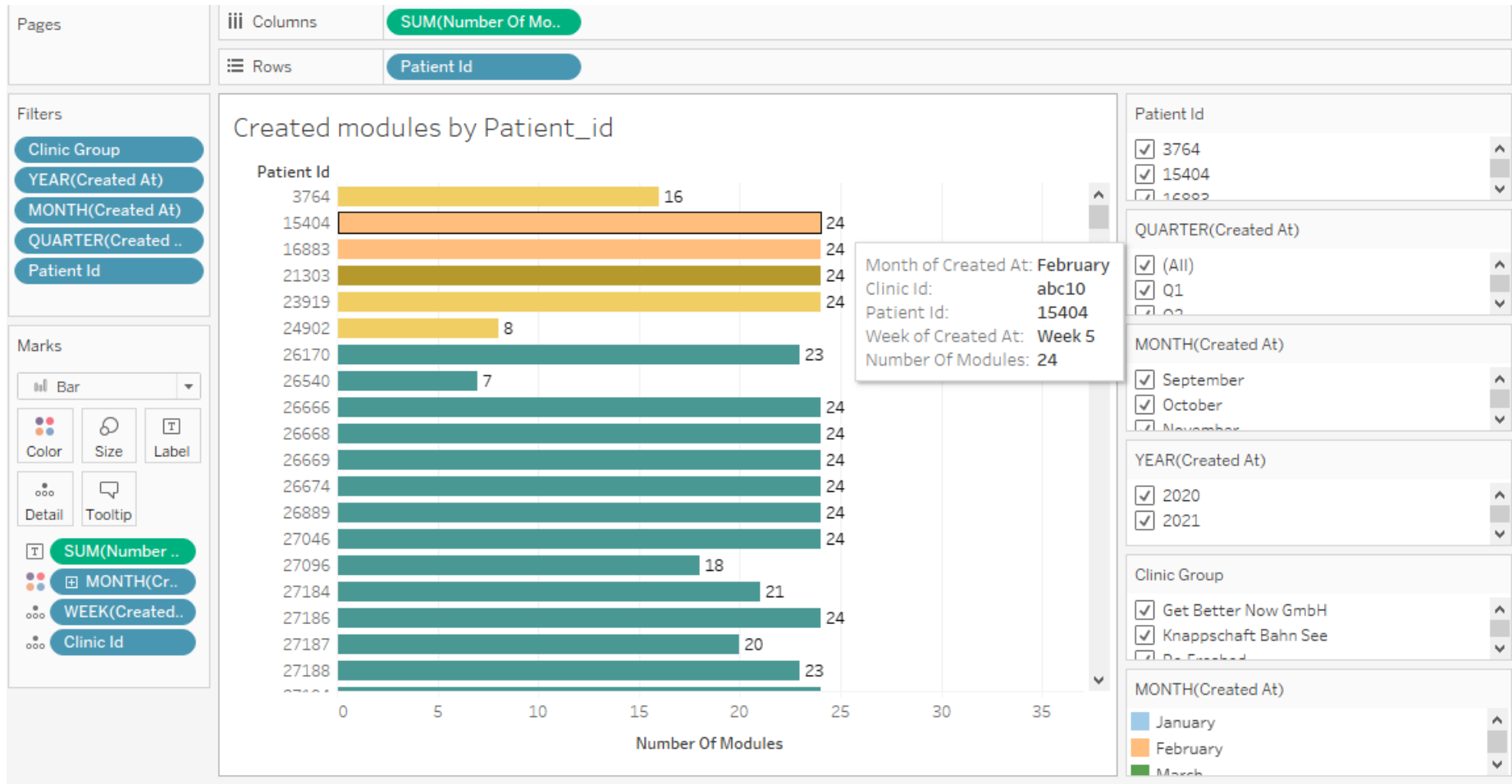
YEAR(Created At)

- ☒ (All)
- ☒ 2018
- ☒ 2019
- ☒ 2020
- ☒ 2021

YEAR(Completion Date)

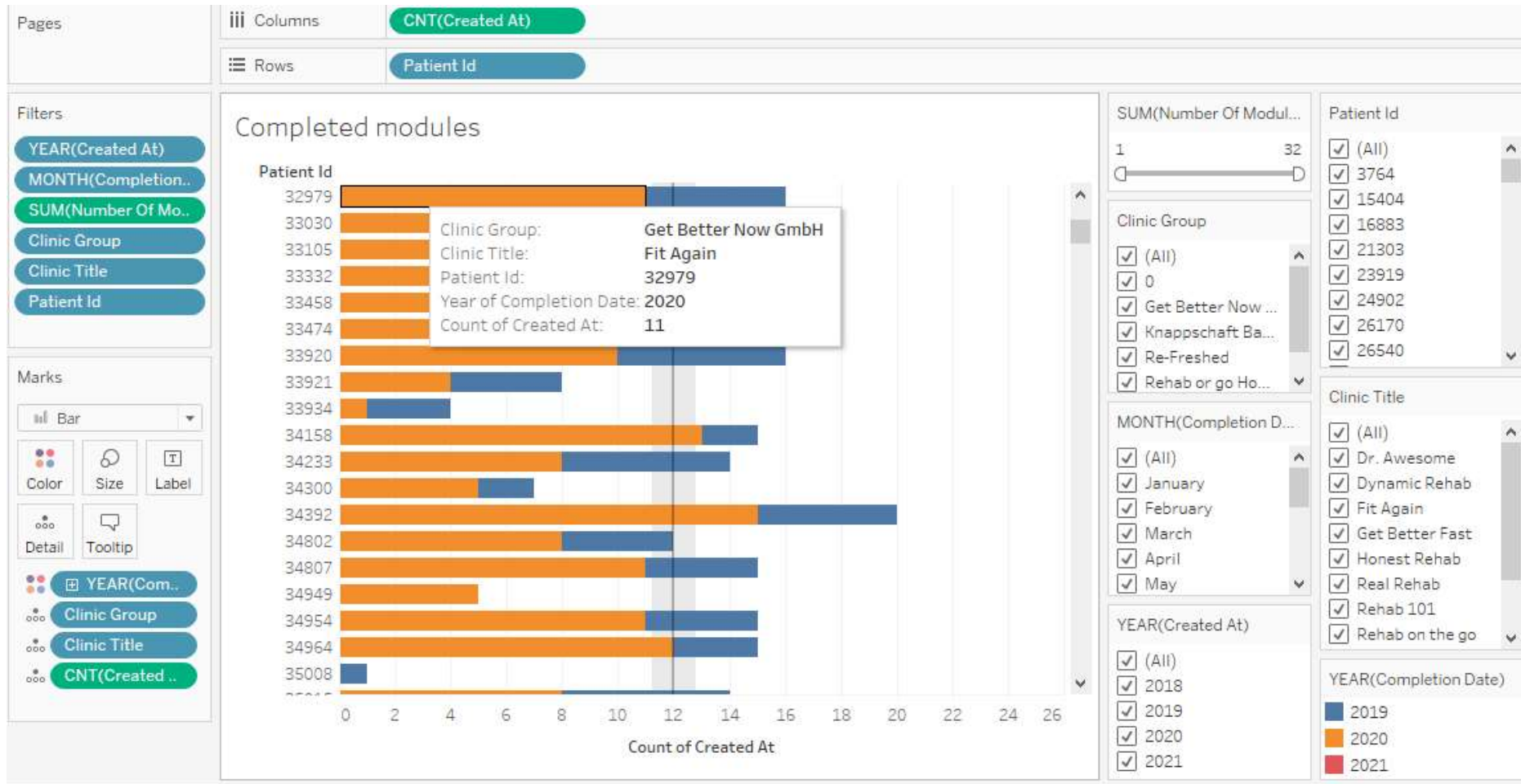
- ☒ 2019
- ☒ 2020
- ☒ 2021

Visualizations: Created Modules



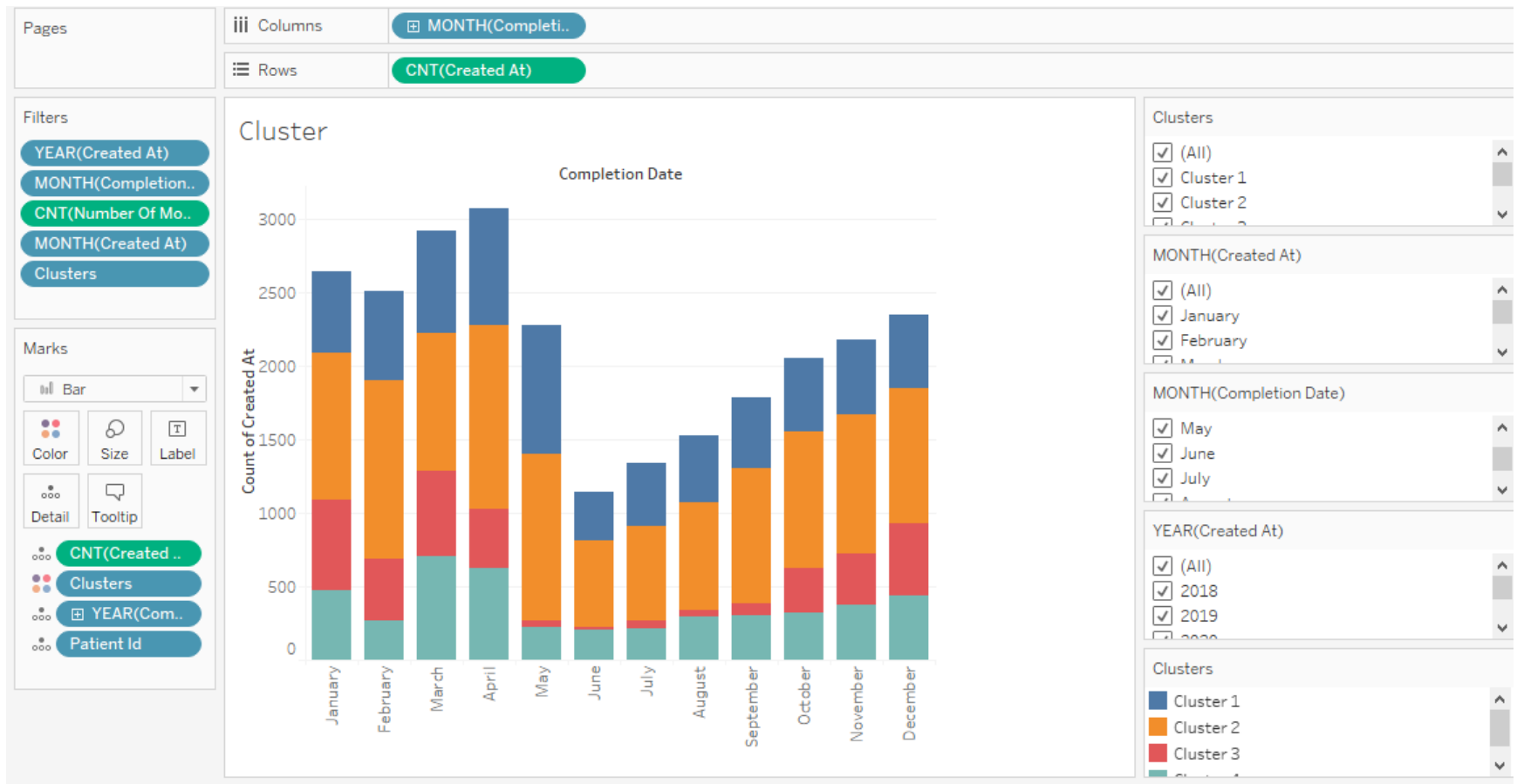
Visualizations:

Completed Modules vs created modules

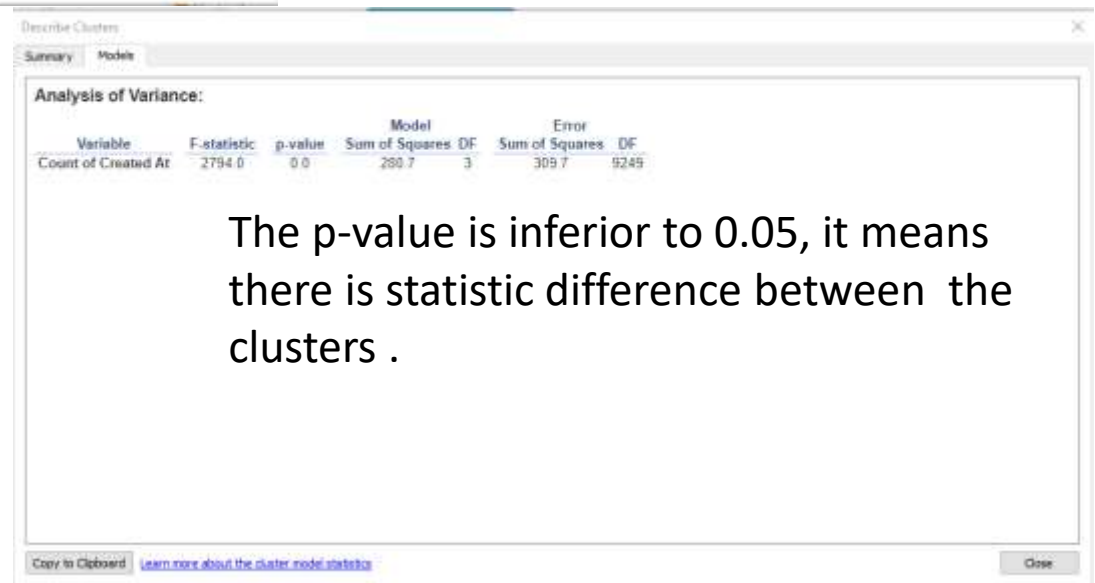
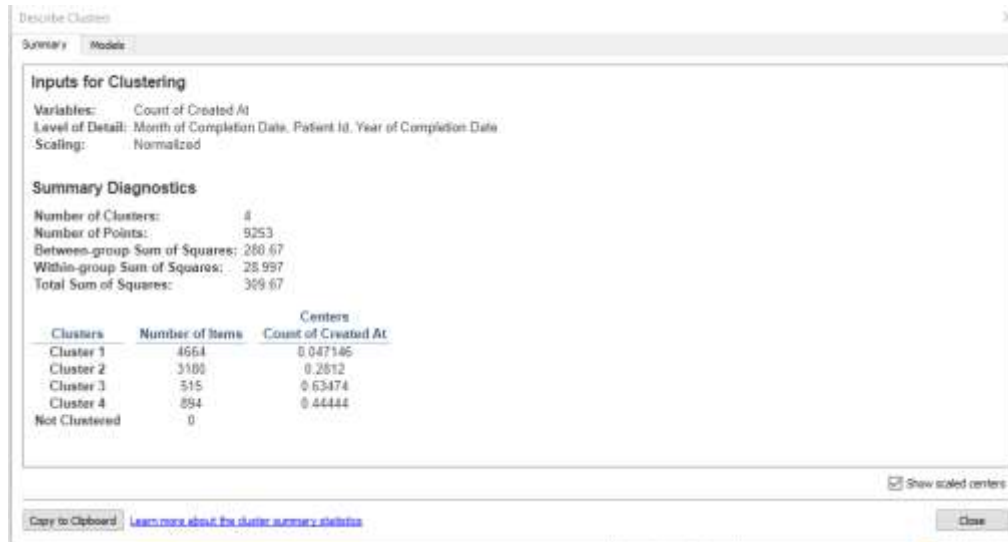


Median value with 95% CI=12

Clustering(1)

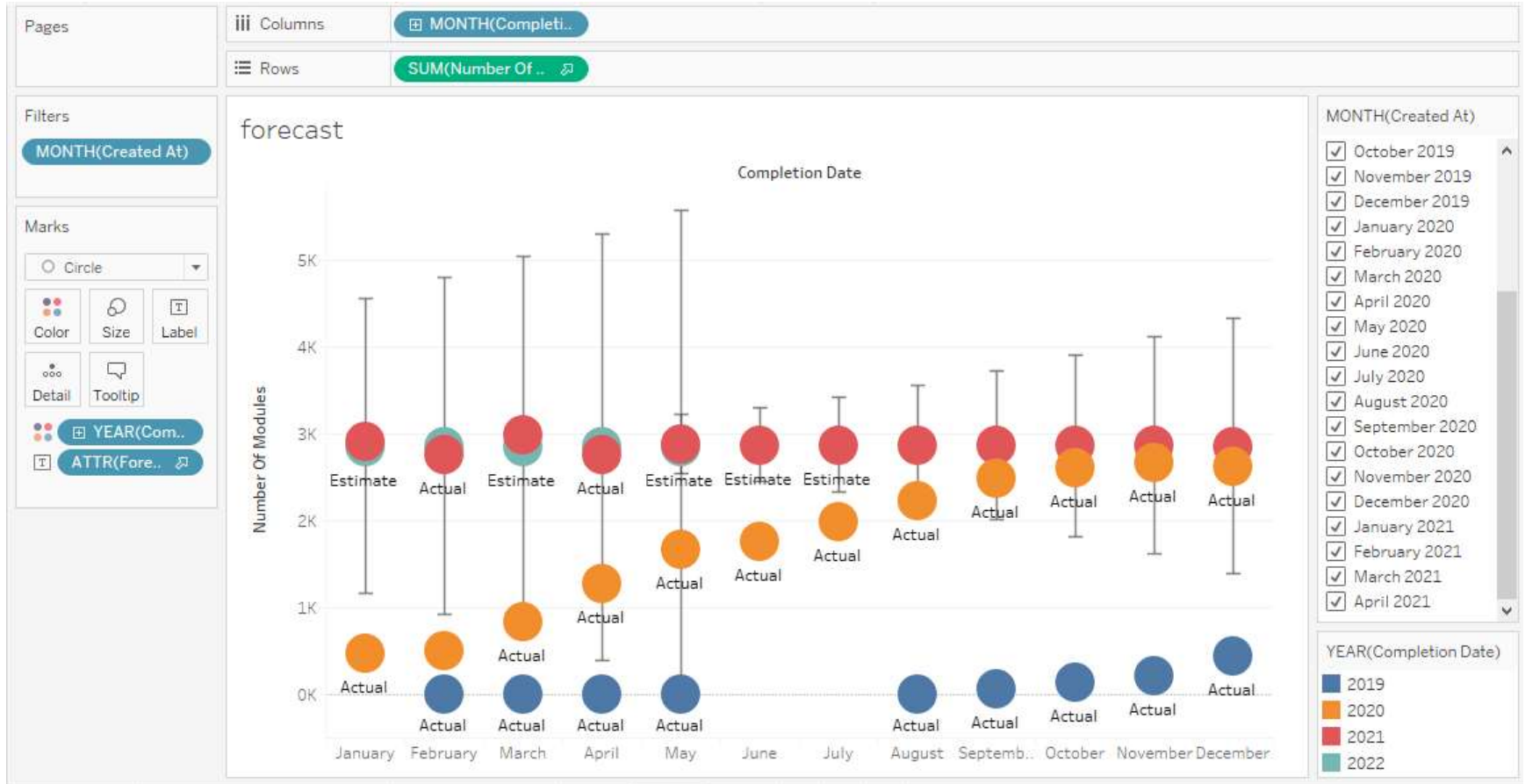


Clustering(2)



The p-value is inferior to 0.05, it means there is statistic difference between the clusters .

Forecasting (1)



Forecasting (2)

Forecast Options

Forecast Length

☒ Automatic Next 13 months

☐ Exactly 1 Years

☐ Until 1 Years

Source Data

Aggregate by: Automatic (Months)

Ignore last: 1 Months

☐ Fill in missing values with zeroes

Forecast Model

Automatic

Automatically selects an exponential smoothing model for data that may have a trend and may have a seasonal pattern.

☒ Show prediction intervals 95%

Currently using source data from February 2019 to April 2021 to create a forecast through May 2022. Looking for potential seasonal patterns every 12 Months.

[Learn more about forecast options](#)

OK

Describe Forecast

Summary Models

Options Used to Create Forecasts

Time series: Month of Completion Date
Measures: Sum of Number Of Modules
Forecast forward: 13 months (May 2021 – May 2022)
Forecast based on: February 2019 – April 2021
Ignore last: 1 month (May 2021)
Seasonal pattern: None (Searched for a seasonal pattern recurring every 12 Months)

Sum of Number Of Modules

Initial	Change From Initial	Seasonal Effect		Contribution		Quality
May 2021	May 2021 – May 2022	High	Low	Trend	Season	
2,880 ± 11.8%	-1.1%	None		100.0%	0.0%	Poor

☒ Show values as percentages

Copy to Clipboard [Learn more about the forecast summary](#) Close

Trend and Season depend very strong on the number of months (or other time of unit) in order to predict time series (how many modules will be done in 12 months)

There isn't a constant seasonality across the decomposition. The magnitude of the seasonal pattern in the data depends on the magnitude of the data, so therefore It's needed to have more data over time in order to calculate the seasonality

Describe Forecast

Summary Models

All forecasts were computed using exponential smoothing.

Sum of Number Of Modules

Model			Quality Metrics					Smoothing Coefficients		
Level	Trend	Season	RMSE	MAE	MASE	MAPE	AIC	Alpha	Beta	Gamma
Additive	Additive	None	174	145	0.93	461.4%	268	0.500	0.500	0.000

In Seasonality you can predict several years, for example, from 1950 to 1990. Meanwhile additive could be an example times series from one year to another

In the additive model, the behavior is linear where changes over time are consistently made by the same amount, like a linear trend.

Copy to Clipboard [Learn more about the forecast models](#) Close