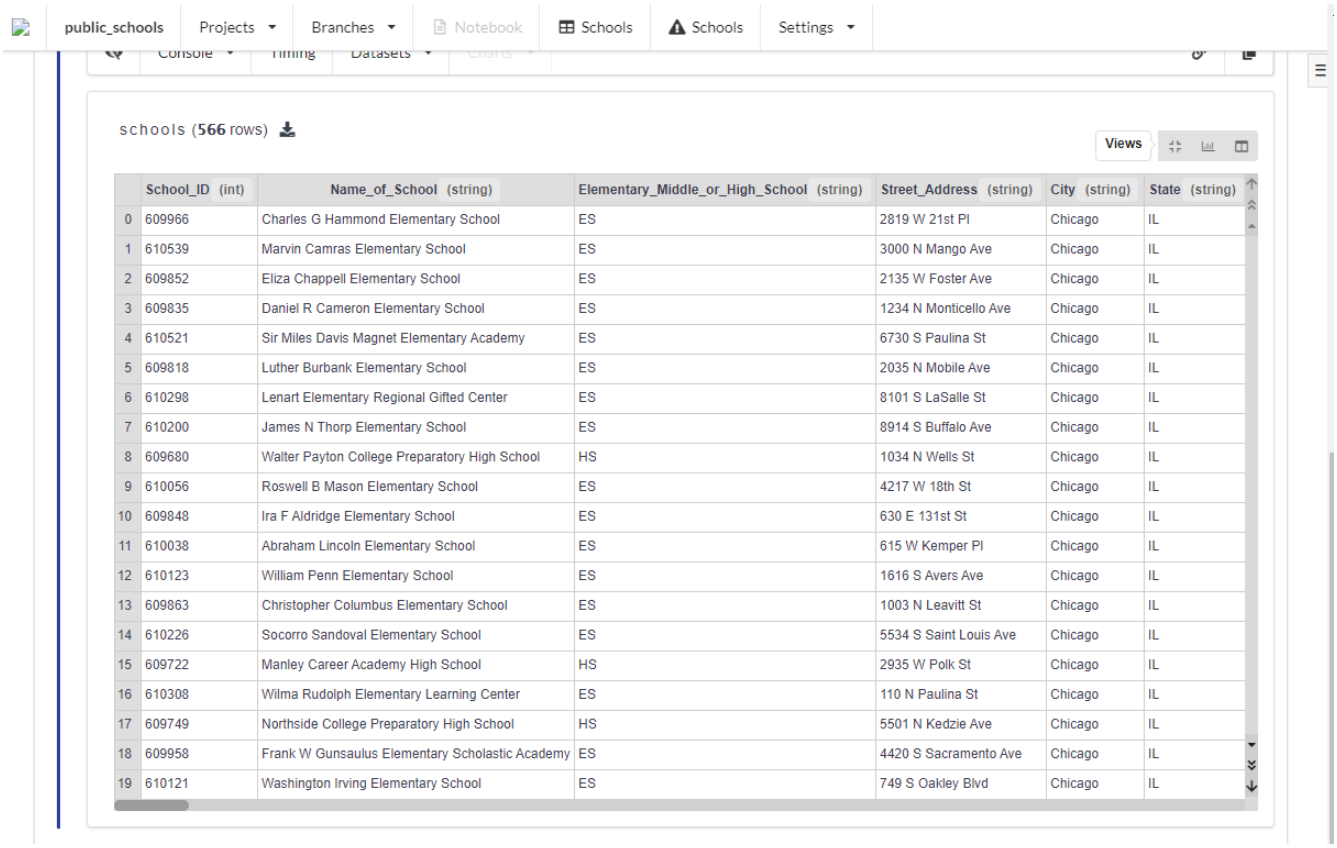
Vizier Assignment

Name:Mingli Tan CID:A20504621

- **Task 1**: load a dataset and take a screenshot of the result



	School_ID (int)	Name_of_School (string)	Elementary_Middle_or_High_School (string)	Street_Address (string)	City (string)	State (string)
0	609966	Charles G Hammond Elementary School	ES	2819 W 21st Pl	Chicago	IL
1	610539	Marvin Camras Elementary School	ES	3000 N Mango Ave	Chicago	IL
2	609852	Eliza Chappell Elementary School	ES	2135 W Foster Ave	Chicago	IL
3	609835	Daniel R Cameron Elementary School	ES	1234 N Monticello Ave	Chicago	IL
4	610521	Sir Miles Davis Magnet Elementary Academy	ES	6730 S Paulina St	Chicago	IL
5	609818	Luther Burbank Elementary School	ES	2035 N Mobile Ave	Chicago	IL
6	610298	Lenart Elementary Regional Gifted Center	ES	8101 S LaSalle St	Chicago	IL
7	610200	James N Thorp Elementary School	ES	8914 S Buffalo Ave	Chicago	IL
8	609680	Walter Payton College Preparatory High School	HS	1034 N Wells St	Chicago	IL
9	610056	Roswell B Mason Elementary School	ES	4217 W 18th St	Chicago	IL
10	609848	Ira F Aldridge Elementary School	ES	630 E 131st St	Chicago	IL
11	610038	Abraham Lincoln Elementary School	ES	615 W Kemper Pl	Chicago	IL
12	610123	William Penn Elementary School	ES	1616 S Avers Ave	Chicago	IL
13	609863	Christopher Columbus Elementary School	ES	1003 N Leavitt St	Chicago	IL
14	610226	Socorro Sandoval Elementary School	ES	5534 S Saint Louis Ave	Chicago	IL
15	609722	Manley Career Academy High School	HS	2935 W Polk St	Chicago	IL
16	610308	Wilma Rudolph Elementary Learning Center	ES	110 N Paulina St	Chicago	IL
17	609749	Northside College Preparatory High School	HS	5501 N Kedzie Ave	Chicago	IL
18	609958	Frank W Gunsaulus Elementary Scholastic Academy	ES	4420 S Sacramento Ave	Chicago	IL
19	610121	Washington Irving Elementary School	ES	749 S Oakley Blvd	Chicago	IL

Task 2: Select the detail view and look at the distributions of some columns. Then look at the column view and take a screenshot of the distribution for column `Teachers_Score`

schools (**566** rows)

Views

	School_ID (int)	Name_of_School (string)	Elementary_Middle_or_High_School (string)	Street_Address (string)
0	609966	Charles G Hammond Elementary School	ES	2819 W 21st Pl
1	610539	Marvin Camras Elementary School	ES	3000 N Mango Ave
2	609852	Eliza Chappell Elementary School	ES	2135 W Foster Ave
3	609835	Daniel R Cameron Elementary School	ES	1234 N Monticello Ave
4	610521	Sir Miles Davis Magnet Elementary Academy	ES	6730 S Paulina St
5	609818	Luther Burbank Elementary School	ES	2035 N Mobile Ave
6	610298	Lenart Elementary Regional Gifted Center	ES	8101 S LaSalle St
7	610200	James N Thorp Elementary School	ES	8914 S Buffalo Ave
8	609680	Walter Payton College Preparatory High School	HS	1034 N Wells St
9	610056	Roswell B Mason Elementary School	ES	4217 W 18th St

Console | Timing | Datasets | Charts

schools (**566** rows)

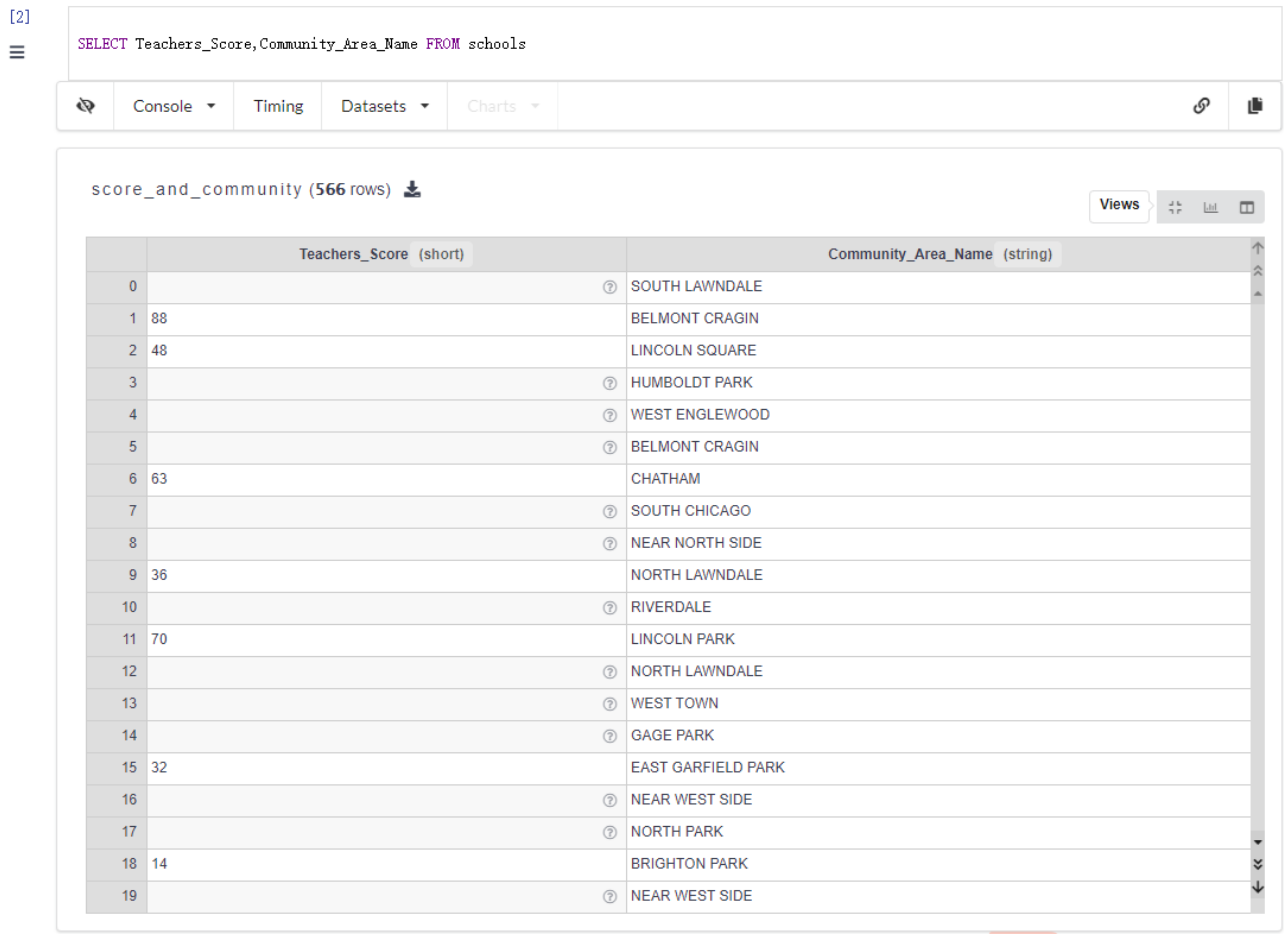
Views

		Null Values	0
Teachers_Score	short	Maximum	99
		Minimum	6
		Mean	49.05
		Std Deviation	17.85
		Sum	14470.00
		Unique Values	74
		Total Values	295
		Null Values	271
Parent_Engagement_Icon	string	Unique Values	4
		Total Values	566
		Null Values	0

Cell Annotations

Comment
Could not cast 'NDA' to ShortType (in schools)

- **Task 4**: Create a SQL cell and write a query that returns columns `Teachers_Score` and `Community_Area_Name`. SQL results can be stored as new datasets in Vizier. Call the result dataset `score_and_community`. And take a screenshot of the result.



[2]

```
SELECT Teachers_Score,Community_Area_Name FROM schools
```

Console | Timing | Datasets | Charts

score_and_community (566 rows)

Views

	Teachers_Score (short)	Community_Area_Name (string)
0		SOUTH LAWNDALE
1	88	BELMONT CRAGIN
2	48	LINCOLN SQUARE
3		HUMBOLDT PARK
4		WEST ENGLEWOOD
5		BELMONT CRAGIN
6	63	CHATHAM
7		SOUTH CHICAGO
8		NEAR NORTH SIDE
9	36	NORTH LAWNDALE
10		RIVERDALE
11	70	LINCOLN PARK
12		NORTH LAWNDALE
13		WEST TOWN
14		GAGE PARK
15	32	EAST GARFIELD PARK
16		NEAR WEST SIDE
17		NORTH PARK
18	14	BRIGHTON PARK
19		NEAR WEST SIDE

- **Task 5**: Create a SQL cell and write a query over the over the `score_and_community` dataset that computes the result as described above. Call the result dataset `community_teacher_scores`. And take a screenshot of the result.

[3]

```
select Community_Area_Name,avg(Teachers_Score) as average_teachers_score from score_and_community group by Community_Area_Name
```

Console | Timing | Datasets | Charts

community_teacher_scores (77 rows)

Views

	Community_Area_Name (string)	average_teachers_score (real)
0	BRIGHTON PARK	50.2
1	LINCOLN PARK	51.8
2	MONTCLARE	45
3	HERMOSA	42
4	LOOP	
5	BELMONT CRAGIN	45.5
6	ROSELAND	45.714285714285715
7	OAKLAND	
8	WASHINGTON PARK	43.5
9	GREATER GRAND CROSSING	40.333333333333336
10	AUBURN GRESHAM	49.57142857142857
11	SOUTH SHORE	52.6
12	SOUTH LAWNDALE	49.15384615384615
13	GARFIELD RIDGE	50.666666666666664
14	LINCOLN SQUARE	43.333333333333336
15	EDISON PARK	71.5
16	DOUGLAS	54.5
17	FULLER PARK	28
18	IRVING PARK	43
19	WEST ELSDON	39.5

- **Task 6**: Create a line chart of the aggregation result by creating a plot cell and take a screenshot of the result.

[4]

```
CREATE Line Chart without Points avrage_teacher_score_chart FOR "community_teacher_scores"
```

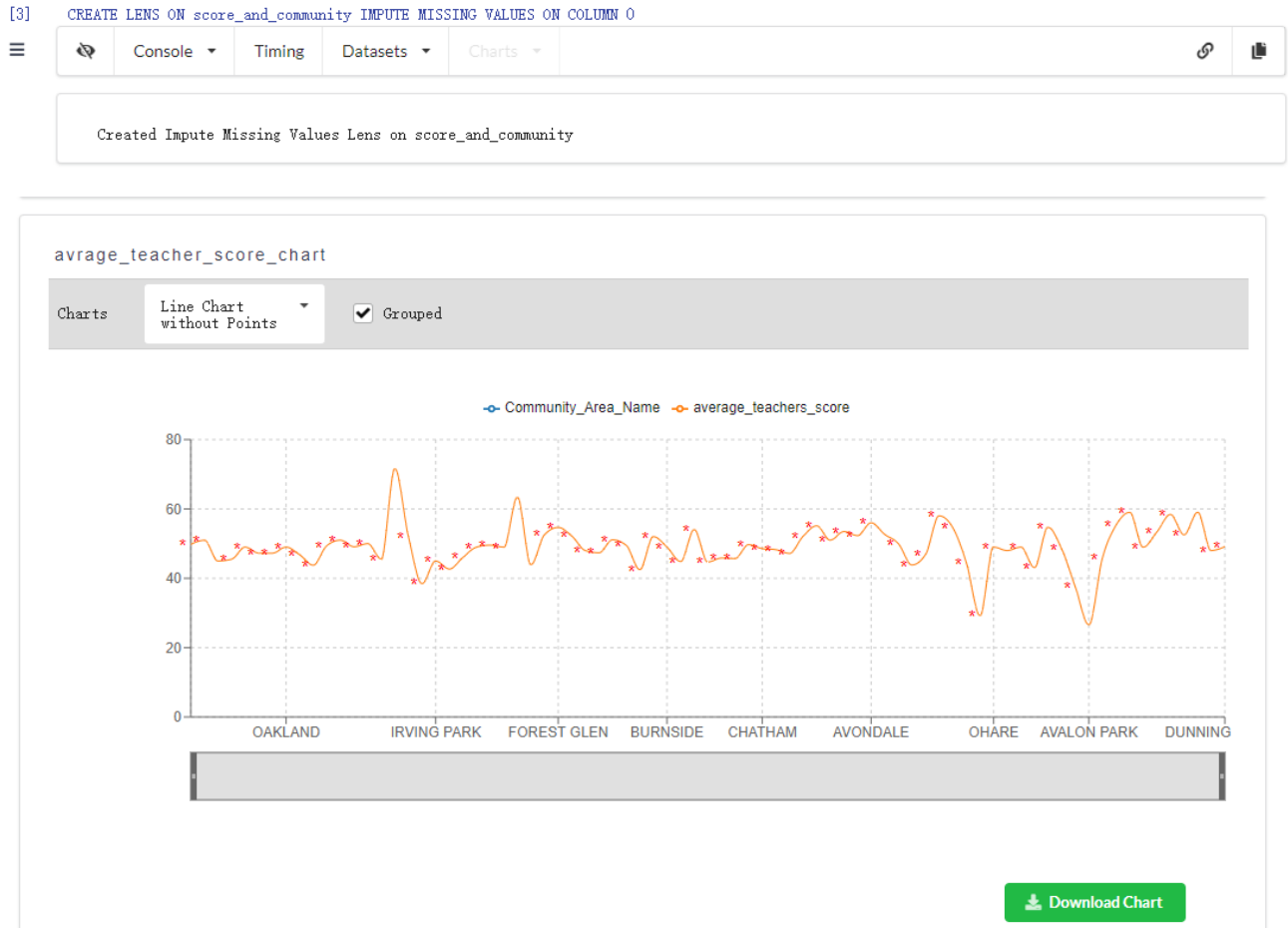
Console | Timing | Datasets | Charts

avrage_teacher_score_chart

Charts: Line Chart without Points | Grouped

- **Task 7**: Insert a new cell above the SQL cell that computes the average teacher scores (notebooks in Vizier are executed top down) by pressing the three bars below the cell number. Select *"Impute Missing*

Values", select the `score_and_community` dataset and `Teachers_Score` as the column to be imputed, and select mean as the imputation method and take a screenshot of the updated line chart.



- **Task 8** Create a Python cell at the end of the notebook and create a function called `print_avg_teachers` that uses Vizier's API to get a handle for this dataset and print all values of the `avg_teacher_score` column. *Hint: use the "Show Code Examples" buttom to see example Vizier API usage and see here for the API documention*. Then use `vizierdb.export_module` to export the function. Then create a second Python cell and use `vizierdb.get_model("print_avg_teachers")` for importing the function and then call it. Take a screenshot of the result.

[6]

```
# Get read-only pandas dataframe object for dataset with given name.
# df = vizierdb.get_data_frame('community_teacher_scores')
def print_avg_teachers():
    ds = vizierdb.get_dataset('community_teacher_scores')
# Iterate over list of dataset rows and print cell value.
# Reference column by name, label ('A', 'B', ...), or
# column index (0, 1, ...).
    for row in ds.rows:
        print(row.get_value('average_teachers_score'))

# Get object for dataset with given name.

# Export a variable, a function or a class for use in subsequent cells
vizierdb.export_module(
    print_avg_teachers
)
#  Use it in a subsequent like normal: add_numbers(1,2)
print_avg_teachers()
```

Console | Timing | Datasets | Charts

```
49.75
51
45
45.5
49
47.25
47.23076923076923
49
46.8
43.8
49.4
51
49.09090909090909
50
45.6
71.5
52
```

[7]

```
print_avg_teachers()
```

Console | Timing | Datasets | Charts

```
49.75
51
45
45.5
49
47.25
47.23076923076923
49
46.8
43.8
49.4
51
49.09090909090909
50
45.6
71.5
52
38.5
45
42.666666666666664
46
49
49.56521739130435
49
63.25
44
52.57142857142857
54.666666666666664
52.2
48
```

- **Task 9** Create another Python cell and use Vizier's API to access the dataset `community_teacher_scores` as a DataFrame, then filter out rows where the `avg_teacher_score` is larger than or equal to `30.0` and then print the remaining rows and take a screenshot.

[8]

```
# Get read-only pandas dataframe object for dataset with given name.
df = vizierdb.get_data_frame('community_teacher_scores')
# print(df)
df = df[df['average_teachers_score']>= 30]
print(df['average_teachers_score'])
```

Console | Timing | Datasets | Charts

```
0     49.750000
1     51.000000
2     45.000000
3     45.500000
4     49.000000
        ...
72    58.400000
73    52.571429
74    59.000000
75    48.000000
76    49.166667
Name: average_teachers_score, Length: 75, dtype: float64
```