Introduction - Purpose:

The idea of the notebook is to have all the documentation for the Samarth database work to clean up the dataset and include them for analysis and creating sample use-case in one place.

The TOC on th left can be used to guide one to relevant parts of the work.

Toggling the raw codes:

```
Out[13]:
```

Click here to toggle on/off the raw code.

Importing Libraries and connecting to SQL:

Connecting to SQL server

Enter password to connect to the Samarth Staging server :

```
········
```

```
Connected successfully
```

Data exploration:

Joining the required table to get required columns together :

Sample 10 rows of the table acquired for calculating student grades for assessments :

Out[37]:

	assessment_percent	assessment_grade	subm_grade_number	stud_grade_number	subject
0	-1.0	C	3	4	
1	-1.0	A	1	2	
2	-1.0	B	1	2	
3	-1.0	B	1	2	
4	-1.0	D	2	3	
5	-1.0	A	2	3	
6	-1.0	A	1	2	
7	-1.0	C	1	2	
8	-1.0	B	1	2	
9	-1.0	B	3	4	

Important columns :

1. Assessment_percent : Assesment grade in terms of percentage
2. Assessment_grade : Assessment grade in terms of grade (A,B,C,D,E) and Z (Absent)
3. Submission_grade_number : Grade number of the student when they submitted
4. Student_grade_number : Student current grade
5. Subject_desc : subject description
6. Subject_grade_number : Grade number obtained from Subject table (looks incorrect)
7. category : GEN,OBC, SC etc
8. is_cswn : Special needs of the students as defined by the state for providing benefits
9. submission_date : date when the submission was made
10. assessment_type : type of assessment - SA2, FA1 etc

Filters applied :

These filters are basic filters to remove test and inactive students/school:

- student.is_enabled = 't'
- school.is_active = 't'
- school.udise > 1111111111

We can see how many rows are being removed each of the filters (and all of the filters combined):

Out[44]:

	Filter_applied	count_rows	Percentage_filtered_out
0	None	11354429	0.00
1	student.is_enabled	6652839	41.41
2	school.is_active	11351087	0.03
3	school.udise	11352492	0.02
4	all	6650036	41.43

I'll save the above table with the filters as a temporary view.

Creating a columns for the grades :

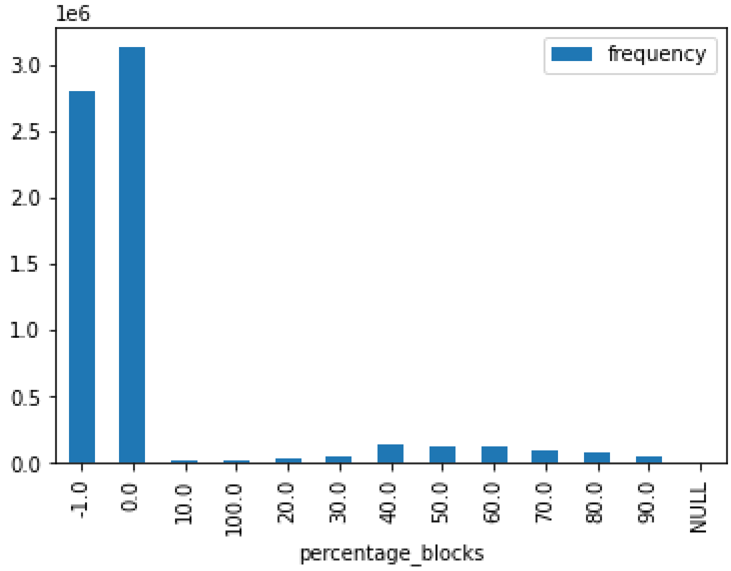
We have two columns for the grades - Assessment_percent : Assesment grade in terms of percentage Assessment_grade : Assessment grade in terms of grade (A,B,C,D,E) and Z (Absent)

We need to verify if these are provided separate from each other or if they are both available at the same time for the submissions

Plotting the assessment percentages :

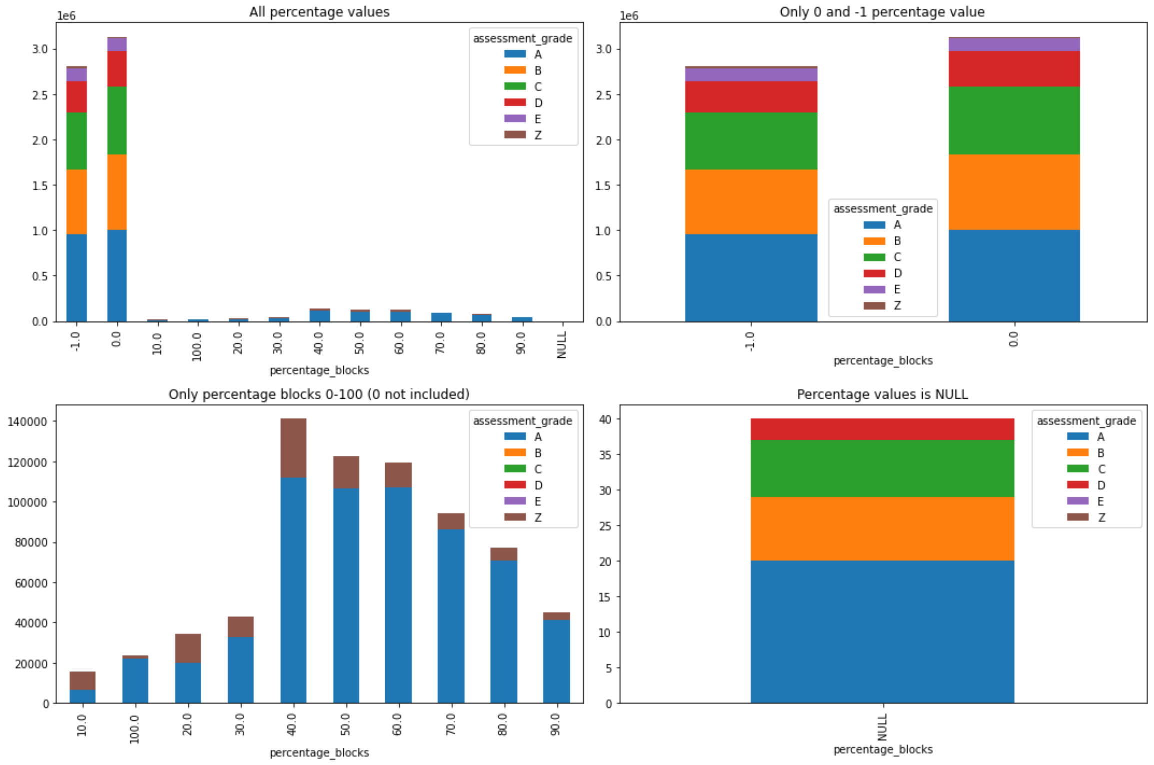
Plotting a histogram of the assessment percentage column , taking 0,-1 and NULL as separate values and considering the rest in blocks of 10

So, 0-10 is considered as 10, 10-20 as 20, 20-30 as 30 and so on:



We can see that most of the values are in 0,-1 while the rest are distributed amongst the 10 blocks. We can look at what are the grades provided for these values, specifically what are the grades when it is in (0,-1 NULL) and what are the grades when assessment_percentage is not in these values i.e. it is in one of the blocks

We plot all the Grades present for each of the percentage values:



From looking at the grade distribution for the percentage values, we can say that when percentage value is 0,-1 or NULL, the grades are provided i.e. we must look at assessment_grade which will have values A,B,C,D,E or Z(if absent)

In other cases ,we must consider the assessment_percentage and not consider the grades; when the percentage is provided, then the grades are either A or Z

Connecting assessment_percentage and assessment_grade:

Looking at the how grades are allocated for assessment percentage in historical work:

Out[184]:

	assessment_grade	assessment_pct_bin	assessment_pct_avg
0	A	80-100	90.0
1	B	65-79	72.0
2	C	50-64	57.0
3	D	35-49	42.0
4	E	1-34	17.0
5	Z	None	NaN

We can see the bins decided for each grade. Assessment percentage average is the average of the minimum and maximum of the range provided for the assessment percent bins for each grade

We can allocate the grades for the percentage scores in our data and look at what the actual average percentage is for those buckets :

Out[188]:

	assessment_grade	assessment_pct_bin	assessment_pct_avg	mean_perc_from_data	medi
0	A	80-100	90.0	87.41	
1	B	65-79	72.0	71.35	
2	C	50-64	57.0	56.85	
3	D	35-49	42.0	41.96	
4	E	1-34	17.0	24.54	
5	Z	None	NaN	NaN	

As we can see, the mean/median of percentage obtained by the students is close to the approximate average of the bins considered except for Grade A and E

For grade A, its lower than the middle of the bin: 87 instead of 90

For grade Z, its higher than the middle of the bin: 25.5 instead of 17

We can assign percentage for each of the grade to have a column with percentages for all rows :

WHEN assessment_grade = 'A' THEN 87

WHEN assessment_grade = 'B' THEN 72

WHEN assessment_grade = 'C' THEN 57

WHEN assessment_grade = 'D' THEN 42

WHEN assessment_grade = 'E' THEN 25.5

WHEN assessment_grade = 'Z' THEN NULL

Similarly, we can assign grades for the percentage values to have a column with grades for the column

We can assign grades based on the buckets pre-defined earlier, i.e.:

WHEN assessment_percent BETWEEN 80 AND 100 THEN 'A'

WHEN assessment_percent BETWEEN 65 AND 80 THEN 'B'

WHEN assessment_percent BETWEEN 50 AND 65 THEN 'C'

WHEN assessment_percent BETWEEN 35 AND 50 THEN 'D'

WHEN assessment_percent BETWEEN 0 AND 35 THEN 'E'

I created a view including all the above columns into the table.

Grade number issues :

Looking again at a sample 10 rows:

We see there are 3 columns for the class (grade_number - columns 5,6,7):

- Submission_grade_number: This is the grade for which the assessment is designed for (from student submission table)
- Stud_grade_number : This is the grade where the student is studying in currently (from the student table)
- Subject_grade_number : This is the grade for which the assessment is designed for (from the subject table)

We see that in the sample, these columns have different values. We expect the subm_grade_number and student_grade_number to have different value based on when the student has submitted the assesment

The current academic year is 2021-2022 and I created a yr_diff column (Stud_grade_number - Submission_grade_number) to calculate how many years ago the student submitted the assesments

The subject grade number values however seem to be unreliable

```
Out[218]:
```

	assessment_percent	assessment_grade	assessment_grade_all	assessment_percent_all	su
0	-1.0	D	D	42.0	
1	-1.0	D	D	42.0	
2	-1.0	B	B	72.0	
3	-1.0	B	B	72.0	
4	-1.0	A	A	87.5	
5	-1.0	A	A	87.5	
6	-1.0	A	A	87.5	
7	-1.0	A	A	87.5	
8	-1.0	B	B	72.0	
9	-1.0	D	D	42.0	

10 rows × 21 columns

We can check what are the yr_diff values present for each academic year :

Out[211]:

	submission_academic_year	yr_diff	count
8	2020-2021	1	5942800
3	2019-2020	2	535438
12	2020-2021	0	98828
18	2019-2020	1	67669
14	2020-2021	2	3397
0	2019-2020	0	983
19	2019-2020	3	276
16	2020-2021	-1	190
10	2020-2021	3	140
5	2020-2021	4	113
6	2020-2021	-4	59
1	2020-2021	-2	34
4	2020-2021	6	32
17	2020-2021	5	24
15	2019-2020	4	19
2	2019-2020	-3	12
9	2019-2020	-1	9
7	2019-2020	5	5
11	2020-2021	-3	4
13	2019-2020	-2	4

We see that there are various year_diff values for each submission_academic_year pointing to some mimatch between the grades_columns

Same submission year but different grades for same student:

For the same academic year, the same student is submitting the assessments for the same subjects of two differnt grades. Also, he is getting same grade in them.

Out[265]:

	assessment_percent	assessment_grade	assessment_grade_all	assessment_percent_all	s
0	-1.0	B	B	72.0	
1	-1.0	B	B	72.0	
2	-1.0	B	B	72.0	
3	-1.0	B	B	72.0	
4	-1.0	B	B	72.0	
5	-1.0	B	B	72.0	
6	-1.0	A	A	87.5	
7	-1.0	A	A	87.5	
8	-1.0	A	A	87.5	
9	-1.0	A	A	87.5	
10	-1.0	A	A	87.5	

Different grades for same student subject assessment submission :

For the same assessment and grade(class) , the student is having different rows with different marks.

Out[224]:

	student_id	assessment_id	subject_id	subject_desc	stud_grade_number	assessment_gr
0	879	16	2	English	8	
1	879	16	6	English	8	
2	882	16	2	English	8	
3	882	16	6	English	8	
4	884	16	1	Hindi	8	
5	884	16	5	Hindi	8	
6	886	16	1	Hindi	8	
7	886	16	5	Hindi	8	
8	890	16	2	English	8	
9	890	16	6	English	8	
10	974	16	1	Hindi	5	

We see that the subject id is differing for these cases. There is some subject id mapping issue as the same subject for the same class is getting mapped to different ids and getting different scores in the assessments

Out[7]:

	count_assessment	count
0	1	6439793
2	2	63319
1	3	9

Very rarely, do we have the issue of multiple grades for the same subject student year:

Out[80]:

```
0.01
```

Creating table at required level:

We finally want a table at the student_id, grade, assesment, subject level with scores for it.
However there is duplication happening at this level with most of it cause by the above subject id issue:

Looking at how much of the current data follows the correct level without duplication:

```
Percentage of the view that is not duplicated at all in the tables:  98.75
94862569297
```

We can also calculate the proportion of this that is caused due to incorrect subject mapping.

Out[76]:

```
0.9433327604409635
```

To summarize, the duplication issue affects only 1.24 % of the rows. 94% of this duplication is caused by different subject ids being mapped to same subject description

Out[15]:

	count_assessment	count_rows	count
0	1	1	6397456
1	3	3	9
2	2	2	62829
3	1	2	1

Out[16]:

	assessment_percent	assessment_grade	assessment_grade_all	assessment_percent_all	su
0	49.4	Z	D	49.4	
1	49.4	A	D	49.4	

Assessments not mapped to LO table:

Number of assessments in the main table:

Out[250]:

	count
0	174

Number of assessments not mapped to LO table:

Out[251]:

	count
0	57

Out[252]:

```
32.76
```

Students not mapped to attendance table:

The number of students available in the base tables:

Out[241]:

	count
0	452771

Number of students for which attendance details are available:

Out[242]:

	count
0	252965

Percentage not available:

Out[247]:

```
55.87
```