Predicting the Probability of Need for Student Intervention

Team 7

Amanda: Project Lead & SME (Business Objectives)

Dylan: Data Scientist// Statistician (Technical and Data Understanding)

Bridge: Data Engineer/ Statistician (Preparing and Building the Model)

Brianne: Business Analyst (Model Accuracy)

QMB6930

Dr. Janice Carrillo

Dr. James Hoover

July 6th, 2020

Summary of Student Success Analysis **4**

Description of the Data Sources **5**

Description of Key Data Tables 5

Data Table Source 5

Data Lake 6

Joining Tables in Data Lake 6

Data Dictionary Files 6

Important Fields or Variables 6

Data Wrangling **9**

Quality of the Data: Key Variables 9

Quality of the Data: Visualizations 11

Quality of the Data: Issues Addressed 14

Pseudo:Code SQL & Python 15

Missing Data & Outliers 15

Data Transformations 16

Data Munging **17**

Additional Data Sources 17

Identifying New Fields 18

Aggregation Methods 19

Development Workflow **20**

Data Source Diagram 20

Data Source Files 22

Appendix **23**

Appendix 1: SQL Historic Indicators for Student Intervention 23

Appendix 2: SQL Historic Indicators for Auto-Generated Messages 24

Appendix 3: SQL Data Tables Utilized 25

Appendix 4: SQL Data Table Creation UF_R1_SUCCESS_ANALYSIS 27

Appendix 5: SQL Data Table Creation UF_R1_SUCCESS_ANALYSIS_UGRD
UF_R1_SUCCESS_ANALYSIS_UGRD_0709
UF_R1_SUCCESS_ANALYSIS_UGRD_1719 30

Appendix 6: SQL Data Dictionary 33

Appendix 7: Python Quality of the Data Report
UF_R1_SUCCESS_ANALYSIS_UGRD 38

Appendix 8: Python Visualizations on the Quality of the Data 42

Appendix 9: SQL Summary Counts of New Variables 46

Appendix 10: SQL UF_R2_ANALYSIS_UNDERGRAD 47
Appendix 11: Student_term_enrollment_model_v3.....50

Summary of Student Success Analysis

There are multiple ways for Universities to define student success beyond a single graduation metric. Historically the University of Florida has used six key indicators when determining if a student is on track for graduation and academic success or if there is a need for student intervention.

Historic Indicators for Student Intervention

- Low Term GPA
- Part-Time Academic Load Status
- Not Registered for Classes
- Withdrawn from Classes
- Full-Time Load Status
- Greater Than Full Time Load Status

These historic indicators are displayed on a dashboard for academic coaches to assist them in advising efforts while meeting with students. Additional indicators are utilized to generate automated electronic outreach to students.

Historic Indicators for Auto-Generated Messages

- Audit Plan Status
- Career Status
- Degree Revoked
- Degree Checkout Status
- Eligible for Graduation
- Expected Graduation Term
- Program Hours
- Program Status

The code for how both sets of indicators have been calculated can be found on Github or in Appendix 1 and 2.

Using data gathered from the UF Data Lake provided by UFIT, an updated analytical model will be created to predict the likelihood of undergraduate student success. The model could then be used to provide more information to coaches who work directly with students.

These initial findings will be shared with SMEs, both at the Provost office, UF IT, and the educators that students face. By using the external team, new insight will be gained into the definition, scope, and boundaries of the findings.

Description of the Data Sources

Description of Key Data Tables

Below is a list of key data tables, including a high-level description of how each is anticipated to be used in the analysis. For a full list of all tables that will be utilized in the analysis please reference Appendix 3.

UF_B_Student_Term: The main student record information table which includes 27 variables of student record information. Notable variables include GPA, transfer credits, and residency status. Variables will be utilized:

1. To evaluate which category of success a student record has historically been placed in utilizing the UF_SUCCESS_INDICATORS table.
2. To determine if any individual variables were a statistically relevant indicator of student success.

Please reference Github or Appendix 4 and 5 for SQL queries to create the following tables.

UF_R1_SUCCESS_ANALYSIS: Was created by joining UF_SUCCESS_TARGETS to the UF_B_STDNT_TERM table.

UF_R1_SUCCESS_ANALYSIS_UNDERGRAD: This has the same variables as the UF_R1_SUCESS_ANALYSIS table but is filtered for just undergraduates.

UF_R1_SUCCESS_ANALYSIS_0709: The UF_R1_SUCCESS_ANALYSIS TABLE was narrowed to the years of 2007 to 2009. These years were chosen to compare if student success indicators from the Great Recession from 2007 to 2009, were the same indicators for student records in more recent graduating classes.

UF_R1_SUCCESS_ANALYSIS_1719: The UF_R1_SUCCESS_ANALYSIS TABLE was narrowed to the years of 2007 to 2009 and 2017 to 2019. These years were chosen so it could be compared if student success indicators from the Great Recession from 2007 to 2009, were the same indicators for student records in more recent graduating classes.

Data Table Source

In the building of this model, data were acquired in three different ways. First, available data was utilized in UF's data lake. Then new data tables were created that combined different data sources within the data lake. Finally, publicly accessible data was utilized and joined with existing student record data for further insights.

Data Lake

Within this report, the term data lake refers to the environment provided by UF IT at the request of the UF Provost office and contains comprehensive anonymized student records dating back to 1977. Before 1977 a small sample of additional records was available as they had been migrated from previous databases. Within the current data lake environment, the data is stored as SQL tables within the SQL Developer (SQL) program. In addition to SQL, Python will be utilized for additional analysis and to build the final model recommendation.

Joining Tables in Data Lake

For ease of use, variables will be joined from multiple tables into new comprehensive tables. First as the UF_R1_SUCCESS_ANALYSIS table. Then tables will be narrowed down to subset the data, based on undergraduate status, as UF_R1_SUCCESS_ANALYSIS_UNDERGRAD and by year enrolled. The goal will be to compare undergraduates enrolled during the great recession, 2007 to 2009, as the UF_R1_SUCCESS_ANALYSIS_0709 table to compare it to recent graduates. For this analysis, recent graduates will be defined as any undergraduates who graduated between 2017 and 2019 and will be subsetting their data on the UF_R1_SUCCESS_ANALYSIS_1719 table. Based on that comparison a model will be built comparing student record variables against the likelihood for student success.

The SQL code for creating and subsetting the new tables can be found on Github and in Appendix 3 & 4.

Data Dictionary Files

Please Reference Appendix 5, to see the complete Data Dictionary File which describes the contents, format, and structure of a database.

Important Fields or Variables

Below are important variables contained in the data tables and a description of why they are anticipated to be important in the analysis**.**

ACAD_CAREER: This is how data is filtered by undergraduate and graduate-level students.

ACAD_PROG_PRIMARY: Defines the academic program in which a student is enrolled in.

ACADEMIC_LOAD: Is a student less likely to graduate within the 4-year term if they are taking less than a full academic load?

Student Success Analysis

AGE_YEARS: Does a student's age have an impact on the likelihood of success? If data were bucketed in "Early" "Traditional" "Late20s" "Advanced" age groups does one group have a better likelihood for success than another?

CUM_GPA: Defines what the student's cumulative GPA is based on all classes completed.

CUR_GPA: What the student's current GPA is based on all classes completed to the current term.

ENRL_CNT: Do the number of classes a student is enrolled in per semester impact a student's likelihood to succeed?

ENRL_FLAG: Indicates whether a student is currently enrolled.

ENRL_SUMMER_A_FLAG: Shows students enrolled for Summer A for their last enrolled module.

ENRL_SUMMER_B_FLAG: Shows students enrolled for Summer B for their last enrolled module.

ENRL_SUMMER_C_FLAG: Shows students enrolled for Summer C for their last enrolled module.

JUNIOR_SENIOR_FLAG: Flags students that are juniors or seniors.

PERSON_SID: Other identifications of a student

RESIDENCY: This indicates a student's residency status while enrolled.

STRM: This identifies the term a student was enrolled as a string text, for example, 'Fall 2019'.

TERM_BEG_DT_SID: This identifies the beginning term as a student who was enrolled as a DateTime value.

TERM_BEG_DT_SID_FALL_CATGRY: By comparing other years, this categorizes a start date for classes as "early," "average," or "late" for the Fall semesters.

TERM_BEG_DT_SID_SPRING_CATGRY: By comparing other years, this categorizes a start date for classes as "early," "average," or "late" for the Spring semesters.

TERM_END_DT_SID: This identifies the end of a term a student was enrolled as a DateTime value.

TERM_END_DT_SID_FALL_EARLY: By comparing other years, this categorizes an end date for classes as "early," "average," or "late" for the Fall semesters.

TERM_END_DT_SID_SPRING_EARLY: By comparing other years, this categorizes an end date for classes as “early,” “average,” or “late” for the Spring semesters.

TERM_LENGTH_CATEGORY: Shows the term length by defining as “Short,” “Average,” and “Long” determined by K Means Clustering.

TERM_LENGTH_DAYS: The number of days of a term length.

TERM_SEASON: The season of which a term is in. I.E. “Summer,” “Spring,” “Fall”

TERM_SID: This is the unique numeric code assigned to each unique term.

TOT_GRADE_POINTS: The total points of a grade.

TOT_PASSD_PRGRSS: The total credits a student has received based on the completion of courses.

TOT_REQUIRED_PRGRSS_PROGRAMNAME: The total credit hours needed for completion of an individual undergrad program. The programs are listed under “Program Name.”

TOT_TAKEN_GPA: This is the total number of GPA points per student.

TOT_TAKEN_PRGRSS: This is the total number of classes that a student has taken.

TOT_TEST_CREDIT: Some students have tested out of classes. This shows the total amount of credits they have received from testing out.

TOT_TRNSFR: The total amount of credits transfer students have transferred with them.

Data Wrangling

Quality of the Data: Key Variables

Multiple reports were created to determine the quality of the data. You can view the outcome of these reports on Github or Appendix 7.

ACAD_CAREER: It appears that every student is coded as either a GRAD, MED, PROF, UGRD, or VEM and the data is reasonably distributed.

ACAD_PROG_PRIMARY: This is a reasonable distributed categorical variable with students categorized as either 'GRAGL', 'GRENG', 'GRLAS', 'PRPBH', 'UGENG', 'UGLAS', 'UGPBH', 'UNVEM' and 'VMVEM'.

ACADEMIC_LOAD: This is a reasonable distributed categorical variable with students categorized as either 'F', 'H', 'L', 'N', or 'T' indicating a student's academic load. The distribution ins F: 512780, H: 369640, L: 344164, N: 493853, T: 239887

AGE_YEARS: The age is reasonably distributed between ages 17 and 26. After 26 years of age, there is not a single student that is between 26 and 31 years old. Then there are 33698 students listed as 31 years old. The issue repeats for students aged 33 years of age with 25214 of students listed.

CUR_GPA: There are 462,094 students records indicating their current GPA was between 1.9 and 4.0. However, there are 2,459,4888 student records with a 0.0 GPA. Of those 2.4 million records 501,873 student records were for students with a 0.0 current GPA listed even when their TOT_PASSD_PRGRSS is greater than 1. Please see Appendix 7 which runs the counts on CUR_GPA in further detail.

DATE_OF_BIRTH: There are 27,108 unique dates of births for student records. For a database whose main records cover 43 years of student records from 1977 to 2020, this number seems plausible.

ENRL_CNT: The number of credits a student is enrolled in seems reasonably accurate and all data falls between 0 and 13 credits.

ENRL_FLAG: This is a reasonable distributed binary variable with students categorized as either 'Y' for yes or 'N' for no indicating where they are an upperclassman.

ENRL_SUMMER_A_FLAG: Every student is listed an 'N' for No indicating that not a single student was enrolled in Summer A classes, which seems unlikely.

ENRL_SUMMER_B_FLAG: This is a reasonable distributed binary variable with students categorized as either ‘Y’ for yes or ‘N’ for no indicating whether they were enrolled in Summer B classes.

ENRL_SUMMER_C_FLAG: This is a reasonable distributed binary variable with students categorized as either ‘Y’ for yes or ‘N’ for no indicating whether they were enrolled in Summer C classes.

JUNIOR_SENIOR_FLAG: This is a reasonable distributed binary variable with students categorized as either ‘Y’ for yes or ‘N’ for no indicating where they are an upperclassman.

PERSON_SID: This is one of the main join ids used to connect the tables and represents a unique numeric identifier assigned to each student.

RESIDENCY: This is a reasonable distributed categorical variable with students categorized as either ‘F’, ‘A’ or ‘N’ indicating a student’s residency status.

STRM: There appear to be 61,056 inputs for the years 1727, 1729, 1730, 1732, and 1734 which are likely dummy variables since this database is only supposed to represent students to 1977.

TERM_BEG_DT_SID: The data appears evenly distributed with no clear outliers and formatted in the YEARMONTHDAY date format.

TERM_END_DT_SID: The data appears evenly distributed with no clear outliers and formatted in the YEARMONTHDAY date format.

TERM_SID: There are dozens of terms identified, and the data appears clean with an equal distribution between terms.

TOT_GRADE_POINTS: The data is reasonably distributed with total grade points ranging from 0 to 448.69.

TOT_PASSD_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation.. Additionally, there are 154,960 student records that report a 0 passed progress. Please reference Appendix 8 for the Quality of Data Report for additional counts.

TOT_TAKEN_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation.

TOT_TEST_CREDIT: As expected most students have tested out of 0 credits (389,720 students) and 36337 students have tested out of between 22 and 30 credits. It is unusual that there are 0 students who have tested out of less than 22 credits.

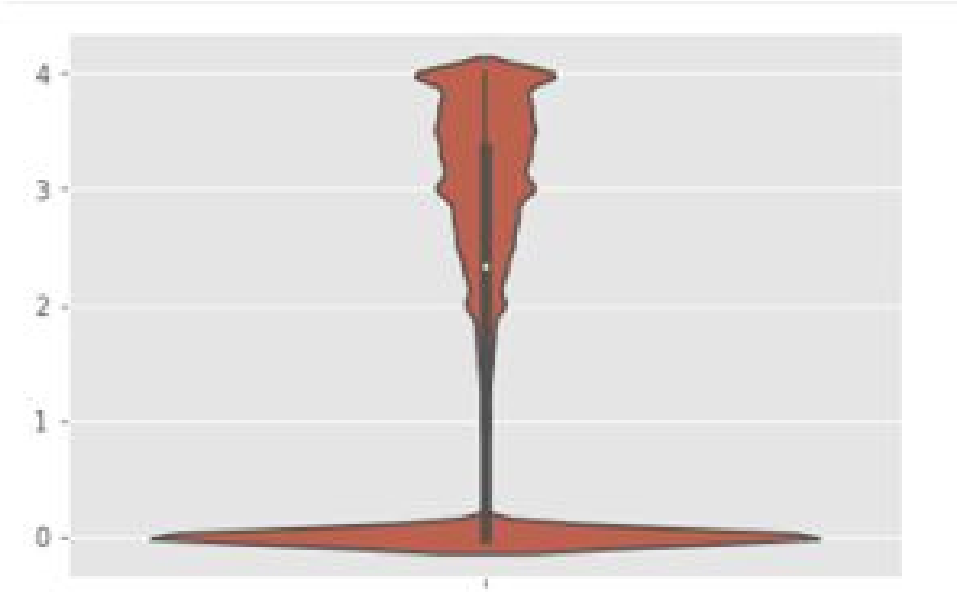
TOT_TRNSFR: As expected most students have 0 transfer credits (309,985 students) and 122248 students transferred between 3 and 15 credits from other institutions.

UF_CLASS_EOT: The data is a reasonably distributed variable ranging from 0 to 9.

Quality of the Data: Visualizations

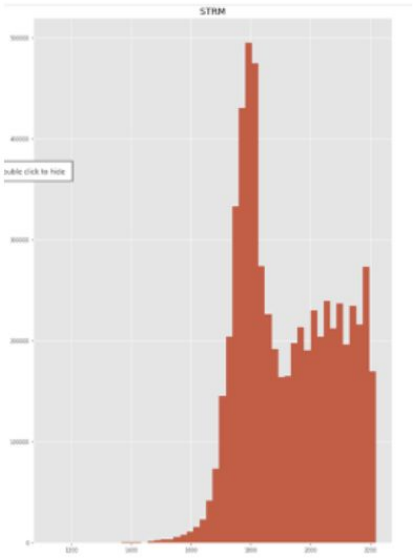
Below are data visualizations and further discussions on the quality of the data for variables where there are unexpected values. For additional Visualizations please reference the project Github page or Appendix 8.

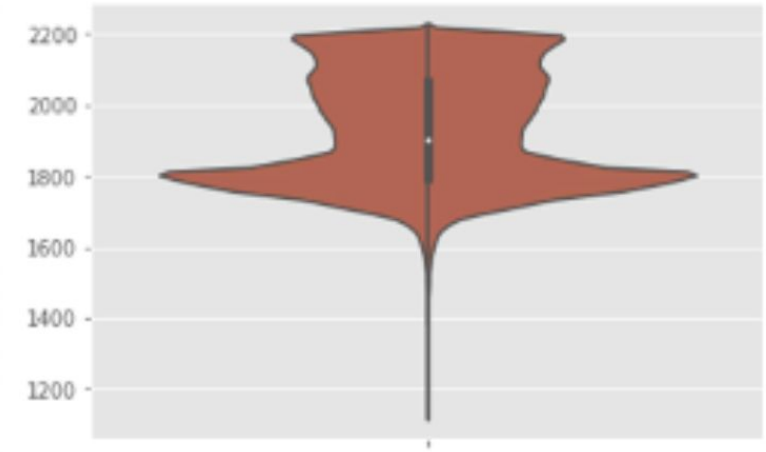
CUR_GPA: Violin Plot



CUR_GPA: Violin Plot: For less than half, 462,094, of students the current GPA is between 1.9 and 4.0. However, there are 501,873 student records with a 0.0 current GPA listed even when their TOT_PASSD_PRGRSS is greater than 1.

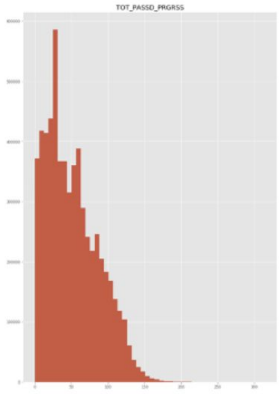
STRM:Histogram & Violin Plot

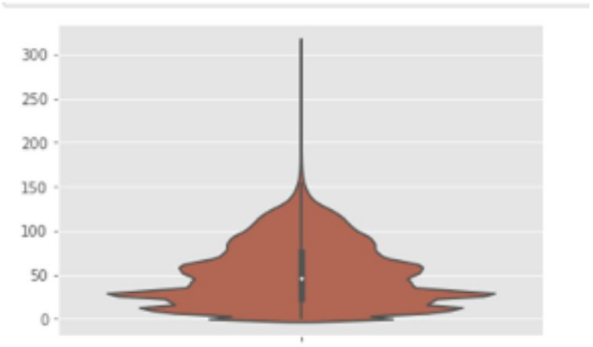




STRM: There appear to be 61,056 inputs for the years 1727, 1729, 1730, 1732, and 1734 which are likely dummy variables since this database is only supposed to represent student records to 1977.

TOT_PASSD_PRGRSS:Histogram & Violin Plot

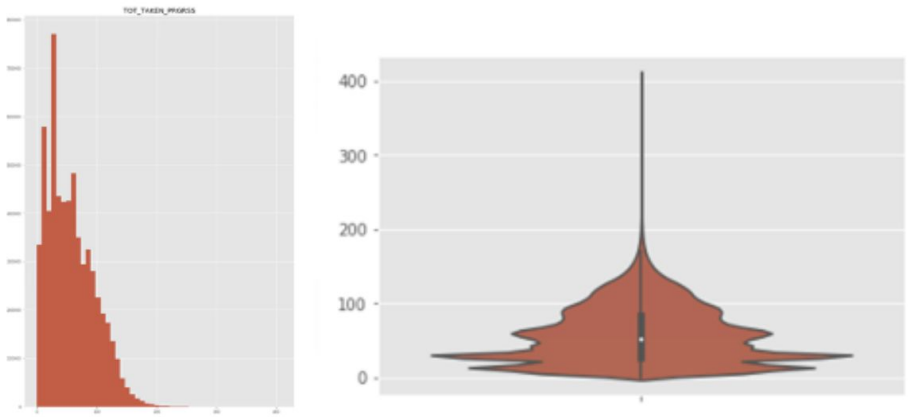




TOT_PASSD_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation.

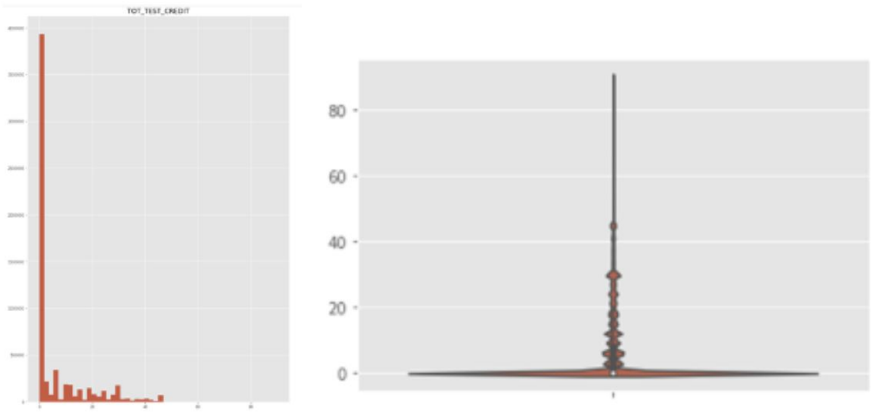
Student Success Analysis

TOT_TAKEN_PRGRSS: :Histogram & Violin Plot



TOT_TAKEN_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation.

TOT_TEST_CREDIT: :Histogram & Violin Plot



TOT_TEST_CREDIT: As expected most students have tested out of credits (252609 students) and 33556 students have tested out of between 22 and 30 credits. It is unusual that there are 108419 students who have tested out of less than 22 credits.

Quality of the Data: Issues Addressed

Below is a short summary explaining how it was decided to address quality issues with the data.

After address the data quality issues below there are 2889722 records on the UF_R1_SUCCESS_ANALYSIS_UGRD table 453265 records on the UF_R1_SUCCESS_ANALYSIS_0709 table and 496569 records on the UF_R1_SUCCESS_ANALYSIS_1719 table.

CUR_GPA: For less than half, 462,094, of students the current GPA is between 1.9 and 4.0. However, there are 501,873 students with a 0.0 current GPA listed even when their TOT_PASSD_PRGRSS is greater than 1. Any records where a GPA is 0 has been eliminated.

TOT_PASSD_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation. Any records where unreasonable decimals are the results have been eliminated.

TOT_TAKEN_PRGRSS: The majority of the data is reasonable. However, four students report their total credits taken in unreasonable decimals (28.68, 40.01, 50.68, and 61.35). Additionally, there are two students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation. Any records where 28.68, 40.01, 50.68, and 61.35 are the results have been eliminated.

TOT_TEST_CREDIT: As expected most students have tested out of credits (252609 students) and 33556 students have tested out of between 22 and 30 credits. It is unusual that there are 108419 students who have tested out of less than 22 credits. More research is needed to be done to determine with the University of Florida project stakeholders before a decision can be made on how to address the data.

Pseudo:Code SQL & Python

Code used throughout this report can be found in the Appendixes to this document or on the Github page at:

Appendix **23**
Appendix 1: SQL Historic Indicators for Student Intervention 23
Appendix 2: SQL Historic Indicators for Auto-Generated Messages 24
Appendix 3: SQL Data Tables Utilized 25
Appendix 4: SQL Data Table Creation UF_R1_SUCCESS_ANALYSIS 27
Appendix 5: SQL Data Table Creation UF_R1_SUCCESS_ANALYSIS_UGRD
UF_R1_SUCCESS_ANALYSIS_UGRD_0709
UF_R1_SUCCESS_AANALYSIS_UGRD_1719 30
Appendix 6: SQL Data Dictionary 33
Appendix 7: Python Quality of the Data Report
UF_R1_SUCCESS_ANALYSIS_UGRD 38
Appendix 8: Python Visualizations on the Quality of the Data 42
Appendix 9: SQL Summary Counts of New Variables 46
Appendix 10: SQL UF_R2_ANALYSIS_UGRDDRAFT
UF_R2_ANALYSIS_UNDERGRAD 47
Appendix 11: Student_term_enrollment_model_v3 50

Missing Data & Outliers

ENRL_SUMMER_A_FLAG: Every student record is listed an ‘N’ for No indicating that not a single student was enrolled in Summer A classes, which seems unlikely.

STRM: There appear to be 61,056 inputs for the years 1727, 1729, 1730, 1732, and 1734 which are likely dummy variables since this database is only supposed to represent student rcords to 1977.

TOT_PASSD_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation.

TOT_TAKEN_PRGRSS: The majority of the data is reasonable. However, some students report their total credits taken in unreasonable decimals (e.g.,28.68, 40.01, 50.68, and 61.35). Additionally, there are some students whose total taken credits are 146, and 161 which is unreasonably high considering students only need 120 total credits for graduation.

Data Transformations

Please reference the Identifying New Fields section on page 17 for information on the data transformation techniques we utilized.

Data Munging

Additional Data Sources

From researching the topic, it was gathered that outside forces could play a factor in student success. Currently, outside data sources that are being considered are as follows:

- **http://zipatlas.com/us/fl/zip:code:comparison/unemployment:rate.htm.**
    - This data shows unemployment rates for Florida based on zip code. It is hypothesized that income available might have an effect on student records in different ways but overall will affect their abilities to succeed in college. By finding unemployment rates in Florida, this could be compared to the student record to see if there is a correlation.
- **https://archive.catalog.ufl.edu/ugrad/1718//liberalarts/majors/home.html**
    - This data shows the different majors and credit requirements. It is hypothesized this might mean student success might need to be met differently depending on the major.
- **https://floridagators.com/sports/2015/12/10/_overview_.aspx**
    - This data shows when the University of Florida won championships. It is hypothesized this may show an increase in admissions depending on the years. This might affect the student success percentage of graduation as the University was accepting students who would not have otherwise applied and got in. Additionally, it is believed that championships were occurring that might have affected schoolwork for students. This data might show a correlation between grade drops and championships that could affect student success.
- **https://www.mapsofworld.com/hurricane/dates.html.**
    - This data shows major hurricanes that hit different states during the years. From calls with the sponsor, it was discussed that hurricanes can cause students to leave the school to help their families. It is hypothesized this might affect student success as grades might suffer from such hardships or might need to take breaks to help families in other states.

Merging Data Sources

The following tables were merged UF_R1_ANALYSIS_UNDERGRAD, UF_B_Person_Student_GRP, & UF_B_TERM_SPLAN into a new table called UF_R2_ANALYSIS_UNDERGRAD.

The tables were joined as part of the Round 2 Analysis plan, the code to create the tables can be found on Github or Appendix 10.

For the Round 3 analysis, it will be determined if one outside source will need to merge with the newly created UF_R2_ANALYSIS_UNDERGRAD table and save as UF_R3_ANALYSIS_UNDERGRAD.

Identifying New Fields

TERM_BEG_DT_SID_FALL_CATGRY: Categorize fall start dates as Early, Average, or Late compared to other years. By using K Means Clustering on Fall Start Dates in TERM_BEG_DT_SID

TERM_BEG_DT_SID_SPRING_CATGRY: Categorize spring start dates as Early, Average, or Late compared to other years. By using K Means Clustering on Spring Start Dates in TERM_BEG_DT_SID

TERM_END_DT_SID_FALL_EARLY: Categorize fall ends dates as Early, Average, or Late compared to other years. By using K Means Clustering on Fall Start Dates in TERM_END_DT_SID

TERM_END_DT_SID_SPRING_EARLY: Categorize spring end dates as Early, Average, or Late compared to other years. By using K Means Clustering on Spring Start Dates in TERM_END_DT_SID

TERM_LENGTH_CATEGORY: Term Length in categorical buckets of Short, Average, and Long determined by K Means Clustering

TERM_LENGTH_DAYS: Term Length in the number of days

TERM_SEASON: =IF(OR(TERM_SID, "Summer...", Summer, ("Spring...", Spring), ("Fall...", Fall)

TOT_REQUIRED_PRGRSS_PROGRAMNAME: The number of credit hours required for the individual undergrad program listed in "Program Name" in the variable: will likely be 50+ new variables.

Aggregation Methods

Aggregating multiple variables we considered by taking variables that exist and combining their outputs to form new variables that could be analyzed against the target definition of student success.

Historically the University of Florida created six variables, as indicators for student success. Two of those new variables LOW_TERM_GPA_IND and WITHDRWL_TERM_IND were created by aggregating information from the ACADEMIC_LOAD, UNT_TAKEN_GPA, and UNT_PASSD_NOGPA.

```
SUM(CASE WHEN "CUR_GPA" <= 2.2 AND "ACADEMIC_LOAD" <> 'N' AND UNT_TAKEN_GPA<> 0 THEN 1 ELSE 0
END) AS "LOW_TERM_GPA_IND",
```

```
SUM(CASE WHEN "ACADEMIC_LOAD" <> 'N' AND UNT_TAKEN_GPA=0 AND  UNT_PASSD_NOGPA=0   THEN 1 ELSE
0 END) AS "WITHDRWL_TERM_IND",
```

Those aggregated indicators were then utilized to determine which students were flagged for intervention and auto-generated electronic follow up. For the complete historic code please see the Github page, or reference Appendix 1 and 2.

```
WHEN ((IND.LOW_TERM_GPA_IND + IND.PARTTIME_TERM_IND + IND.NOT_REG_TERM_IND +
IND.WITHDRWL_TERM_IND)>0) THEN ('5')
```

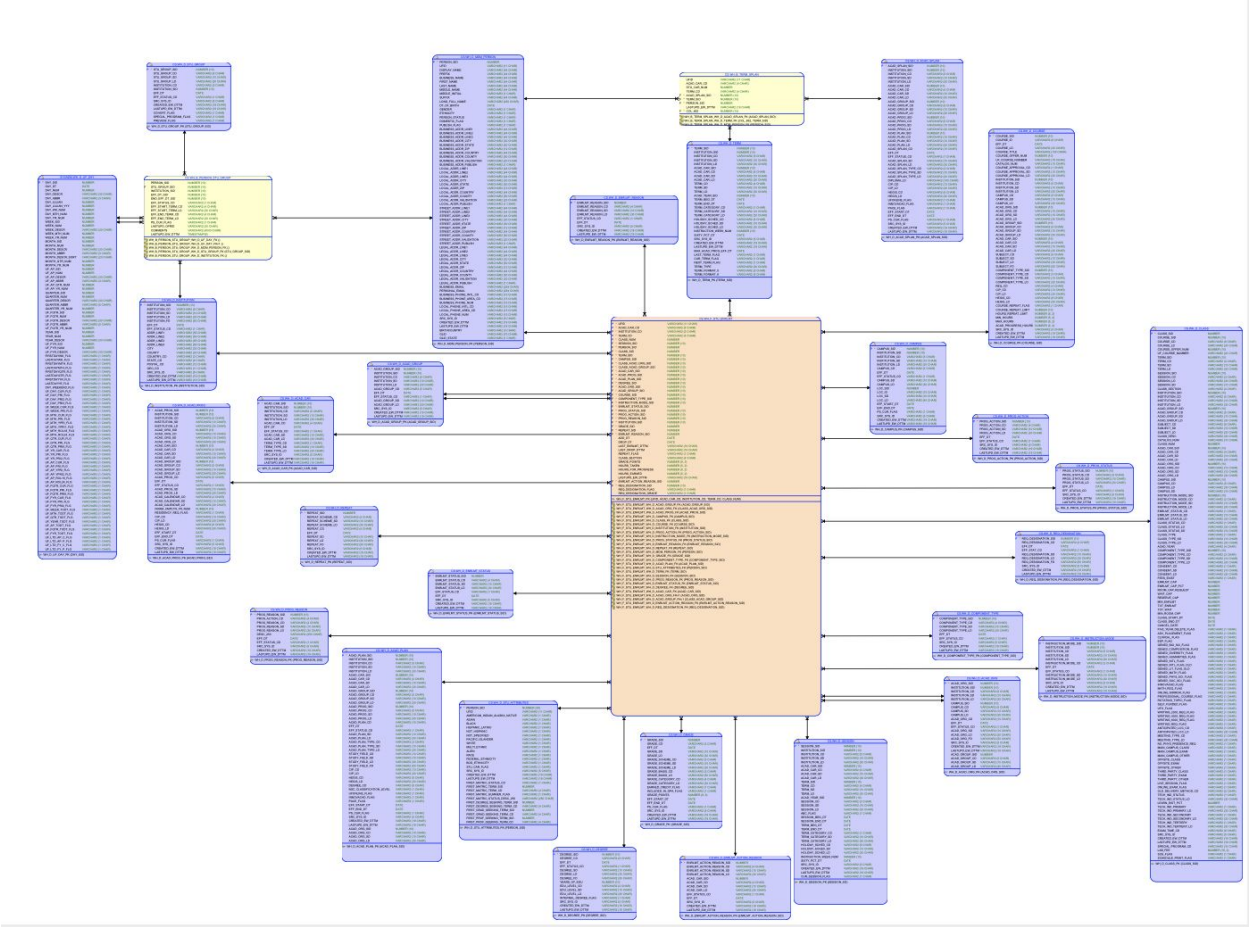
Moving forward aggregated LOW_TERM_GPA_IND and WITHDRWL_TERM_IND will be utilized and discussion, where further aggregation could be beneficial, will be continued.

Development Workflow

Data Source Diagram

While the University of Florida data lake is home to numerous tables, the tables that are most relevant to modeling student success are in the term workgroup.

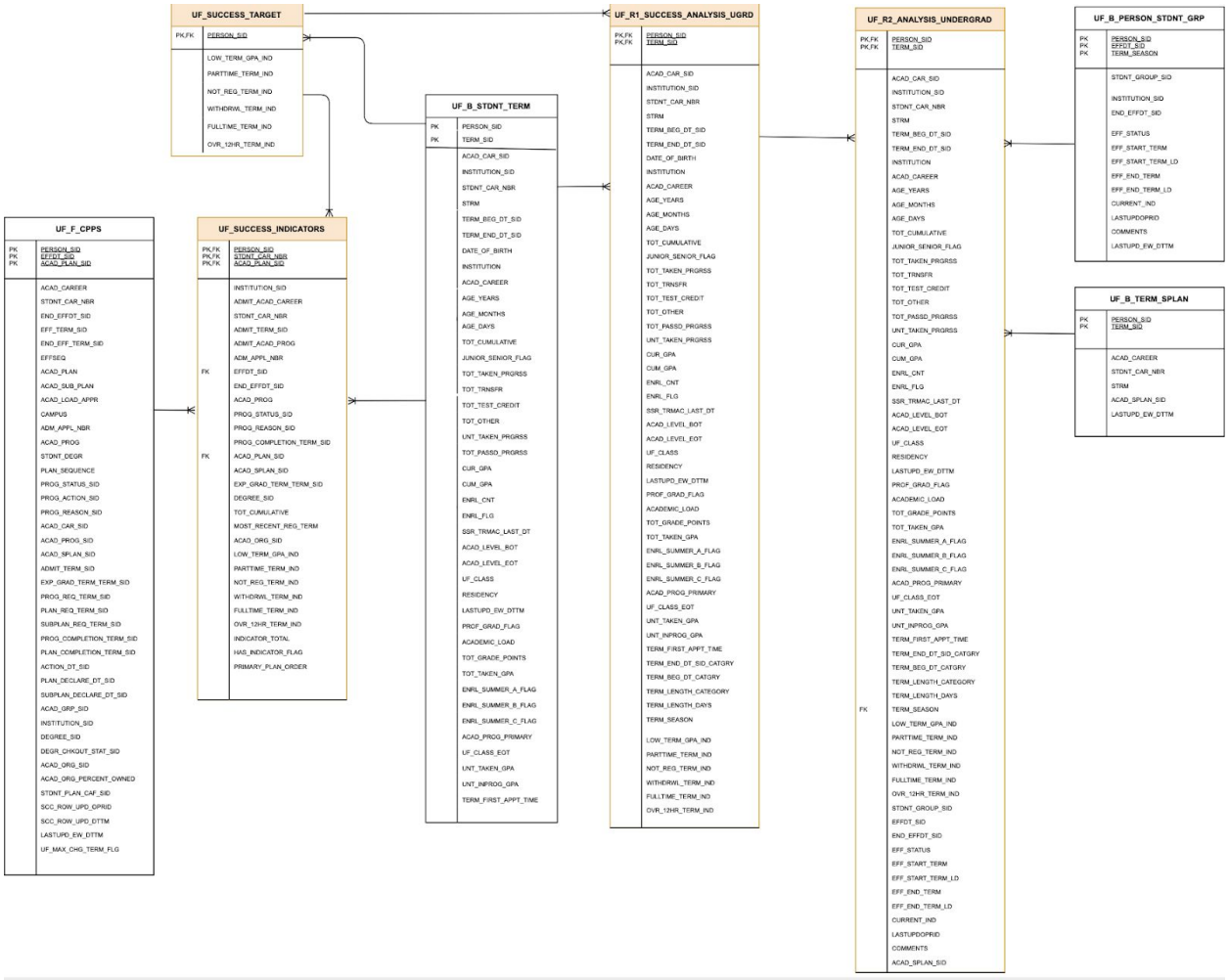
Please note that many table names and variable names have been updated, after the data was scrubbed from the original tables of students identifying information and then populated into a more anonymized format. Therefore the below historic ER diagram should be seen as a reference to the quantity of tables that could be included in the analysis from the term ecosystem. For a clearer picture of ‘Student_term_enrollment_model_v3’ please refer to the project Github page or Appendix 11.



Student Success Analysis

For purposes in the initial analysis, UF_B_STDNT_TERM table will be utilized to create UF_SUCCESS_TARGET table, then these two tables will be joined with the UF_F_CPPS table to create UF_SUCCESS_INDICATORS table. UF_B_STDNT_TERM table will be joined with UF_SUCCESS_TARGET table then filter the combination of variables to include only undergraduate students on the UF_R1_SUCCESS_ANALYSIS_UGRD table.

Below is a diagram depicting how tables were joined for our Round 1 Analysis. ,



As analysis moved to round 2 tables within the UF_Term ecosystem will continue to be joined to the UF_R1_SUCCESS_ANALYSIS_UGRD table utilizing the PERSON_SID and TERM_SID.

Moving forward there are 8 table ecosystems within the UF data lake that could be relevant when modeling student success. The following tables have been identified:

- UF_D_ACAD_CAR
- UF_D_ACAD_GRP
- UF_D_ACAD_ORG
- UF_D_ACAD_PLAN
- UF_D_ACAD_PROG
- UF_D_ACAD_SPLAN
- UF_D_CAMPUS
- UF_D_CLASS
- UF_D_CRSE
- UF_D_INSTITUTION

- UF_D_INSTRCTN_MODE

Data Source Files

When examining the data, and deciding which of the numerous tables to join for the analysis of student success below were divided:

Round 1 Data Sources

Historic ER Student_term_enrollment_model_v3: Github
UF_B_STDNT_TERM: Output available in the data lake F: Drive

Round 2 Data Sources

Round 2 ER Diagram UF_R2_ANALYSIS_UNDERGRAD: Github
UF_B_TERM_SPLAN: Output available in the data lake F: Drive

Round 3 Data Sources

Historic ER CPP_model_v10: Github
Historic ER Class_Meeting_Patterns_and_Instructors_model_v1: Github
Historic ER External_test_score_model_v2: Github
Historic ER Milestone_model_v1: Github
Historic ER Service_indicators_flags_holds_model_v2: Github
Historic ER Student_term_enrollment_model_v3: Github
Historic ER Student_term_progress_model_v3: Github
Historic ER Student_transfer_enrollment_model_v1: Github

Appendix

Appendix 1: SQL Historic Indicators for Student Intervention

```
IND AS (SELECT

    "EMPLID" EMPLID,

   SUM(CASE WHEN "CUR_GPA" <= 2.2 AND "ACADEMIC_LOAD" <> 'N' AND
UNT_TAKEN_GPA<> 0 THEN 1 ELSE 0 END) AS "LOW_TERM_GPA_IND",

   SUM(CASE WHEN "ACADEMIC_LOAD" in ('P','H','L','T') THEN 1 ELSE 0 END) AS
"PARTTIME_TERM_IND",

   SUM(CASE WHEN "ACADEMIC_LOAD" = 'N' THEN 1 ELSE 0 END) AS
"NOT_REG_TERM_IND",

   SUM(CASE WHEN "ACADEMIC_LOAD" <> 'N' AND UNT_TAKEN_GPA=0 AND
UNT_PASSD_NOGPA=0   THEN 1 ELSE 0 END) AS "WITHDRWL_TERM_IND",

   SUM(CASE WHEN "ACADEMIC_LOAD" = 'F' THEN 1 ELSE 0 END) AS
"FULLTIME_TERM_IND",

   SUM(CASE WHEN "UNT_PASSD_GPA" > 12 THEN 1 ELSE 0 END) AS
"OVR_12HR_TERM_IND"

FROM CS.UF_IPR_COHORTS CHT INNER JOIN CS.PS_STDNT_CAR_TERM CAR on
CHT.UNIV_ROW_ID=CAR.EMPLID and CHT.STRM in ('2165','2168')

  where CAR.STRM < (Select DISTINCT(TERM_CD) from CS.UF_D_TERM where
UF_D_TERM.UF_CURR_TERM_FLG = 'Y')

       AND ACAD_CAREER='UGRD'

GROUP BY EMPLID),
```

Appendix 2: SQL Historic Indicators for Auto-Generated Messages

```
WHEN ((D.UF_COLLVL_APPRV ='Y' AND D.UF_DGC_REVOKED<> 'Y' ) or
B.DEGR_CHKOUT_STAT='AW') THEN ('1')
WHEN (D.UF_COLLVL_FINAL='Y' AND D.UF_COLLVL_APPRV='N' AND
D.EXP_GRAD_TERM='2201') THEN ('2')
WHEN (B.PROG_STATUS = 'DM') THEN ('No Message to Be Sent')
WHEN (B.PROG_STATUS = 'DC') THEN ('7')
WHEN ((D.UF_DGC_REVOKED = 'Y' OR D.UF_COLLVL_APPRV_P='N') AND
D.EXP_GRAD_TERM='2201') THEN ('1b')
WHEN (D.ACTIVE_FLAG='Y' AND D.EXP_GRAD_TERM='2201' AND (
AUD.AUD_PLAN_STATUS='FAIL' OR AUD.CAREER_STATUS='FAIL')) THEN ('1b')
WHEN (D.UF_DGC_REVOKED = 'N' AND D.UF_COLLVL_APPRV_P='Y'  AND
D.EXP_GRAD_TERM='2201') THEN ('1a')
WHEN (D.UF_DGC_REVOKED  = 'N' AND D.ACTIVE_FLAG='Y' AND
D.EXP_GRAD_TERM='2201') THEN ('1a')
WHEN (B.DEGR_CHKOUT_STAT in ('EG','WD') AND AUD.AUD_PLAN_STATUS<>'FAIL' AND
AUD.CAREER_STATUS<>'FAIL' ) THEN ('3') --EG-Eligible for Graduation, DN-Denied,
WD-Withdran   removed DN
WHEN (E.UF_PRGM_HRS>=125) THEN ('4')  --2014 and 2015 this bucket will need to be
removed WHEN ((IND.LOW_TERM_GPA_IND + IND.PARTTIME_TERM_IND +
IND.NOT_REG_TERM_IND + IND.WITHDRWL_TERM_IND)>0) THEN ('5')
```

Appendix 3: SQL Data Tables Utilized

UF_B_Student_Term: The main student information table which includes 27 variables of student information. Notable variables include GPA, transfer credits, and residency status. Variables will be utilized:

3. To evaluate which category of success a student is placed in on the UF_SUCCESS_INDICATORS table.
4. To determine if any individual variables were a statistically relevant indicator of student success.

UF_R1_SUCCESS_ANALYSIS: Was created by joining UF_SUCCESS_TARGETS to the UF_B_STDNT_TERM table. Below is the code utilized for this process.

UF_R1_SUCCESS_ANALYSIS_UNDERGRAD: This has the exact same variables as the UF_R1_SUCESS_ANALYSIS table but is filtered for just undergraduate student records.

UF_R1_SUCCESS_ANALYSIS_0709: The UF_R1_SUCCESS_ANALYSIS TABLE was narrowed to the years of 2007 to 2009. These years were chosen to compare if student success indicators from the Great Recession from 2007 to 2009, were the same indicators for students in more recent graduating classes.

UF_R1_SUCCESS_ANALYSIS1719: The UF_R1_SUCCESS_ANALYSIS TABLE was narrowed to the years of 2007 to 2009 and 2017 to 2019. These years were chosen to compare if student success indicators from the Great Recession from 2007 to 2009, were the same indicators for students in more recent graduating classes.

UF_SUCCESS_INDICATORS:

Other Data Tables Utilized

UF_B_TERM_SPLAN:

UF_D_ACAD_CAR

UF_D_ACAD_GRP

UF_D_ACAD_ORG

UF_D_ACAD_PLAN

UF_D_ACAD_PROG

UF_D_ACAD_SPLAN

UF_D_CAMPUS

UF_D_CLASS

UF_D_CRSE

UF_D_INSTITUTION

UF_D_REQUIREMENT

UF_D_SRVC_IMPACT

UF_F_CPSS:

UF_F_CPSS_Terms:

Appendix 4: SQL Data Table Creation
UF_R1_SUCCESS_ANALYSIS

```
CREATE TABLE UF_R1_SUCCESS_ANALYSIS  as (SELECT
A.ACAD_CAR_SID,
A.INSTITUTION_SID,
A.STDNT_CAR_NBR,
A.TERM_SID,
A.STRM,
A.TERM_BEG_DT_SID,
A.TERM_END_DT_SID,
A.INSTITUTION,
A.DATE_OF_BIRTH
A.ACAD_CAREER,
A.AGE_YEARS,
A.AGE_MONTHS,
A.AGE_DAYS,
A.TOT_CUMULATIVE,
A.JUNIOR_SENIOR_FLAG,
A.TOT_TAKEN_PRGRSS,
A.TOT_TRNSFR,
A.TOT_TEST_CREDIT,
A.TOT_OTHER,
A.TOT_PASSD_PRGRSS,
A.UNT_TAKEN_PRGRSS,
A.CUR_GPA,
A.CUM_GPA,
A.ENRL_CNT,
A.ENRL_FLG,
A.SSR_TRMAC_LAST_DT,
A.ACAD_LEVEL_BOT,
A.ACAD_LEVEL_EOT,
A.UF_CLASS,
A.RESIDENCY,
A.LASTUPD_EW_DTTM,
A.PROF_GRAD_FLAG,
A.ACADEMIC_LOAD,
A.TOT_GRADE_POINTS,
A.TOT_TAKEN_GPA,
A.ENRL_SUMMER_A_FLAG,
A.ENRL_SUMMER_B_FLAG,
A.ENRL_SUMMER_C_FLAG,
```

```
A.ACAD_PROG_PRIMARY,
A.UF_CLASS_EOT,
A.UNT_TAKEN_GPA,
A.UNT_INPROG_GPA,
A.TERM_FIRST_APPT_TIME,
   (CASE WHEN substr(A.TERM_BEG_DT_SID,-4)<=0230 AND
substr(A.TERM_END_DT_SID,-4)<=0430 THEN( 'EARLY')
   WHEN substr(A.TERM_BEG_DT_SID,-4)<=0230 AND
substr(A.TERM_END_DT_SID,-4)>0430 AND  substr(A.TERM_END_DT_SID,-4)<=0504
THEN( 'AVERAGE')
   WHEN substr(A.TERM_BEG_DT_SID,-4)<=0230 AND
substr(A.TERM_END_DT_SID,-4)>0504 THEN( 'LATE')
   WHEN substr(A.TERM_BEG_DT_SID,-4)>=0800 AND
substr(A.TERM_END_DT_SID,-4)<=1216 THEN( 'EARLY')
   WHEN substr(A.TERM_BEG_DT_SID,-4)>=0800 AND
substr(A.TERM_END_DT_SID,-4)>1216 AND substr(A.TERM_END_DT_SID,-4)<=1220 THEN(
'AVERAGE')
   WHEN substr(A.TERM_BEG_DT_SID,-4)>=0800 AND
substr(A.TERM_END_DT_SID,-4)>1220 THEN( 'LATE')
   ELSE NULL
   END )AS "TERM_END_DT_SID_CATGRY",
   (CASE WHEN substr(A.TERM_BEG_DT_SID,-4)<=0103 THEN ('EARLY')
   WHEN substr(A.TERM_BEG_DT_SID,-4)>0103 AND
substr(A.TERM_BEG_DT_SID,-4)<=0105 THEN ('AVERAGE')
    WHEN substr(A.TERM_BEG_DT_SID,-4)>0105 AND
substr(A.TERM_BEG_DT_SID,-4)<=0230 THEN ( 'LATE')
    WHEN substr(A.TERM_BEG_DT_SID,-4)>0800 AND
substr(A.TERM_BEG_DT_SID,-4)<=0821 THEN ('EARLY')
   WHEN substr(A.TERM_BEG_DT_SID,-4)>0821 AND
substr(A.TERM_BEG_DT_SID,-4)<=0825 THEN ('AVERAGE')
   WHEN substr(A.TERM_BEG_DT_SID,-4)>0825 THEN ( 'LATE')
    ELSE NULL
    END) AS "TERM_BEG_DT_CATGRY",
    (CASE WHEN
(TO_DATE(A.TERM_END_DT_SID,'YYYYMMDD')-TO_DATE(A.TERM_BEG_DT_SID,'YYYYM
MDD')) <=107 THEN('SHORT')
    WHEN
(TO_DATE(A.TERM_END_DT_SID,'YYYYMMDD')-TO_DATE(A.TERM_BEG_DT_SID,'YYYYM
MDD')) >107 AND
(TO_DATE(A.TERM_END_DT_SID,'YYYYMMDD')-TO_DATE(A.TERM_BEG_DT_SID,'YYYYM
MDD')) <=119 THEN ('AVERAGE')
```

```
    WHEN
(TO_DATE(A.TERM_END_DT_SID,'YYYYMMDD')-TO_DATE(A.TERM_BEG_DT_SID,'YYYYM
MDD')) >119 THEN('LONG')
    END) AS "TERM_LENGTH_CATEGORY",
(TO_DATE(A.TERM_END_DT_SID,'YYYYMMDD')-TO_DATE(A.TERM_BEG_DT_SID,'YYYYM
MDD')) AS "TERM_LENGTH_DAYS",
    (CASE WHEN substr(A.TERM_BEG_DT_SID,-4)>=0800 THEN ('FALL')
     WHEN substr(A.TERM_BEG_DT_SID,-4)<=0230 THEN ('SPRING')
     ELSE ('SUMMER')
     END) AS "TERM_SEASON",
B.PERSON_SID,
B.LOW_TERM_GPA_IND,
B.PARTTIME_TERM_IND,
B.NOT_REG_TERM_IND,
B.WITHDRWL_TERM_IND,
B.FULLTIME_TERM_IND,
B.OVR_12HR_TERM_IND
   FROM UF_SUCCESS_TARGET B INNER JOIN UF_B_STDNT_TERM A
   ON A.PERSON_SID = B.PERSON_SID
   WHERE A.CUR_GPA<>0
         AND (A.TOT_PASSD_PRGRSS-trunc(A.TOT_PASSD_PRGRSS/1,0))=0
         AND (A.TOT_TAKEN_PRGRSS-trunc(A.TOT_TAKEN_PRGRSS/1,0))=0
)
```

Appendix 5: SQL Data Table Creation
UF_R1_SUCCESS_ANALYSIS_UGRD
UF_R1_SUCCESS_ANALYSIS_0709
UF_R1_SUCCESS_ANALYSIS_1719

```
CREATE TABLE UF_R1_SUCCESS_ANALYSIS_UGRD as (SELECT
*  from UF_R1_SUCCESS_ANALYSIS
WHERE A.ACAD_CAR_SID = 8
)

CREATE TABLE UF_R1_SUCCESS_ANALYSIS_0709 AS(SELECT
* from UF_R1_SUCCESS_ANALYSIS_UGRD
  WHERE TERM_BEG_DT_SID>20070000 AND TERM_BEG_DT_SID<20091231
)

CREATE TABLE UF_R1_SUCCESS_ANALYSIS_1719 AS(SELECT
* from UF_R1_SUCCESS_ANALYSIS_UGRD
  WHERE TERM_BEG_DT_SID>20170000 AND TERM_BEG_DT_SID<20191231
```

Appendix 6: SQL Data Dictionary

Table	Variable	Description	Data Type
UF_D_ACAD_CAR	ACAD_CAR_SID	Academic Career surrogate identification	NUMBER(10,0)
UF_D_ACAD_PLAN	ACAD_CAR_SID	Academic Career surrogate identification	NUMBER(10,0)
UF_D_ACAD_PROG	ACAD_CAR_SID	Academic Career surrogate identification	NUMBER(10,0)
UF_D_ACAD_SPLAN	ACAD_CAR_SID	Academic Career surrogate identification	NUMBER(10,0)
UF_D_CLASS	ACAD_CAR_SID	Academic Career surrogate identification	NUMBER(10,0)
UF_D_CRSE	ACAD_CAR_SID	Academic Career surrogate identification	NUMBER(10,0)
UF_B_STDNT_TERM	ACAD_CAREER	Describes whether the student is in the undergrad or graduate program.	VARCHAR2(4 CHAR)
UF_B_TERM_SPLAN	ACAD_CAREER		VARCHAR2(4 CHAR)
UF_B_STDNT_TERM	ACAD_LEVEL_BOT		VARCHAR2(3 CHAR)
UF_B_STDNT_TERM	ACAD_LEVEL_EOT		VARCHAR2(3 CHAR)
UF_B_STDNT_TERM	ACAD_PROG_PRIMARY	Which Academic program a student is enrolled in ie UGENG = Undergraduate Engineering	VARCHAR2(5 CHAR)
UF_B_TERM_SPLAN	ACAD_SPLAN_SID	Academic Sub Plan Surrogate Identification	NUMBER(10,0)
UF_B_STDNT_TERM	ACADEMIC_LOAD	The number of classes a student takes per term, also known as the academic load.	VARCHAR2(1 CHAR)
UF_B_STDNT_TERM	AGE_YEARS	Age in Years of the person for the term	NUMBER(10,0)
UF_B_STDNT_TERM	CUM_GPA	Cumulative GPA for all activity up to the term	NUMBER(8,3)

UF_B_STDNT_TERM	CUR_GPA	Current GPA for the term activity	NUMBER(8,3)
UF_B_Person_Student_GRP	EFF_END_TERM		VARCHAR2(4 CHAR)
UF_B_Person_Student_GRP	EFF_END_TERM_LD	Effective End PeopleSoft Term Long Description	VARCHAR2(30 CHAR)
Create_New	EFF_LENGTH_TERM	=EFF_END_TERM : EFF_START_TERM	VARCHAR2(4 CHAR)
UF_B_Person_Student_GRP	EFF_START_TERM		VARCHAR2(4 CHAR)
UF_B_Person_Student_GRP	EFF_START_TERM_LD	Effective Start PeopleSoft Term Long Description	VARCHAR2(30 CHAR)
UF_B_Person_Student_GRP	END_EFFDT_SID	unsure what these numbers stand for.	NUMBER(10,0)
UF_B_STDNT_TERM	ENRL_SUMMER_A_FLAG	If a student had enrolled in summer A classes for their last reported Mod	VARCHAR2(1 CHAR)
UF_B_STDNT_TERM	ENRL_SUMMER_B_FLAG	If a student had enrolled in summer B classes for their last reported Mod	VARCHAR2(1 CHAR)
UF_B_STDNT_TERM	ENRL_SUMMER_C_FLAG	If a student had enrolled in summer C classes for their last reported Mod	VARCHAR2(1 CHAR)
UF_D_ACAD_CAR	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_ACAD_GRP	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_ACAD_ORG	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_ACAD_PLAN	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_ACAD_PROG	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_ACAD_SPLAN	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_CAMPUS	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)

UF_D_CLASS	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_CRSE	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_INSTITUTION	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_INSTRCTN_MODE	INSTITUTION_SID	Institution surrogate identification	NUMBER(10,0)
UF_D_REQUIREMENT	INSTITUTION_SID	Institution surrogate identification	
UF_D_SRVC_IMPACT	INSTITUTION_SID	Institution surrogate identification	NUMBER(38,0)
UF_B_STDNT_TERM	JUNIOR_SENIOR_FLAG	Flag (Y/N) indicating if Student is a Junior or Senior	VARCHAR2(1 CHAR)
UF_B_Person_Student_GRP	PERSON_SID	Person surrogate identification	NUMBER(10,0)
UF_B_STDNT_TERM	PERSON_SID	Person surrogate identification	NUMBER(38,0)
UF_B_TERM_SPLAN	PERSON_SID	Person surrogate identification	NUMBER(10,0)
UF_B_STDNT_TERM	RESIDENCY		VARCHAR2(5 CHAR)
UF_D_SRVC_IMPACT	SRVC_IMPACT_LD		VARCHAR2(30 CHAR)
UF_D_SRVC_IMPACT	SRVC_IMPACT_SD		VARCHAR2(10 CHAR)
UF_B_STDNT_TERM	SSR_TRMAC_LAST_DT		DATE
UF_B_Person_Student_GRP	STDNT_GROUP_SID		NUMBER(10,0)
UF_B_STDNT_TERM	STRM		VARCHAR2(4 CHAR)
UF_B_STDNT_TERM	TERM_BEG_DT_SID		NUMBER(38,0)

Create_New_UF_B_STDNT_TERM	TERM_BEG_DT_SID_FALL_CATGRY	Categorize fall start dates as Early, Average, or Late compared to other years. By using K Means Clustering on Fall Start Dates in TERM_BEG_DT_SID	Categorical
Create_New_UF_B_STDNT_TERM	TERM_BEG_DT_SID_SPRING_CATGRY	Categorize spring start dates as Early, Average, or Late compared to other years. By using K Means Clustering on Spring Start Dates in TERM_BEG_DT_SID	Categorical
UF_B_STDNT_TERM	TERM_END_DT_SID		NUMBER(38,0)
Create_New_UF_B_STDNT_TERM	TERM_END_DT_SID_FALL_EARLY	Categorize fall ends dates as Early, Average, or Late compared to other years. By using K Means Clustering on Fall Start Dates in TERM_END_DT_SID	Categorical
Create_New_UF_B_STDNT_TERM	TERM_END_DT_SID_SPRING_EARLY	Categorize spring end dates as Early, Average, or Late compared to other years. By using K Means Clustering on Spring Start Dates in TERM_END_DT_SID	Categorical
Create_New_UF_B_STDNT_TERM	TERM_LENGTH_CATEGORY	Term Length in categorical buckets of Short, Average, and Long determined by K Means Clustering	Number
Create_New_UF_B_STDNT_TERM	TERM_LENGTH_DAYS	Term Length in the number of days	Number
Create_New_UF_B_STDNT_TERM	TERM_SEASON	=IF(OR(TERM_SID, "Summer...", Summer, ("Spring...", Spring), ("Fall...", Fall)	Categorical
UF_B_STDNT_TERM	TERM_SID	Term surrogate identification	NUMBER(38,0)
UF_B_STDNT_TERM	TOT_GRADE_POINTS	Total Grade Points	NUMBER(9,3)
UF_B_STDNT_TERM	TOT_PASSD_PRGRSS	Total number of credits a student has passed	NUMBER(8,3)
Create_New_UF_B_STDNT_TERM	TOT_REQUIRED_PRGRSS_*PROGRAMNAME*	The number of credit hours required for the individual undergrad program listed in "Program Name" in the variable: will likely be 50+ new variables	Number

UF_B_STDNT_TERM	TOT_TAKEN_GPA		NUMBER(8,3)
UF_B_STDNT_TERM	TOT_TAKEN_PRGRSS	The total number of credit hours a student has taken? Amanda asking Andrew to clarify.	NUMBER(22,8)
UF_B_STDNT_TERM	TOT_TEST_CREDIT	The total number of credits a student tested out of.	NUMBER(22,8)
UF_B_STDNT_TERM	TOT_TRNSFR	The total number of transfer credits	NUMBER(22,8)
UF_B_STDNT_TERM	UF_CLASS		VARCHAR2(1 CHAR)
UF_B_STDNT_TERM	UF_CLASS_EOT		VARCHAR2(1 CHAR)
UF R1 SUCCESS ANALYSIS:			
UF_R1_SUCCESS_ANALYSIS_UNDERGRAD:			
UF_R1_SUCCESS_ANALYSIS_0709:			
UF_R1_SUCCESS_ANALYSIS1719:			
UF_SUCCESS_INDICATORS:			

Appendix 7: Python Quality of the Data Report
UF_R1_SUCCESS_ANALYSIS_UGRD

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
import datetime as dt
%matplotlib inline

uf = pd.read_csv('UF_R1_SUCCESS_ANALYSIS.csv')
df2=uf.loc[:,['CUR_GPA']]
df3=uf.loc[:,['DATE_OF_BIRTH']]
df4=uf.loc[:,['STRM']]
df5=uf.loc[:,['TOT_PASSD_PRGRSS']]
df6=uf.loc[:,['TOT_TAKEN_PRGRSS']]
df7=uf.loc[:,['TOT_TEST_CREDIT']]

sns.violinplot(data=uf,y=df2)
plt.show()

df2.hist(figsize=(10, 15), bins=50, xlabelsize=8, ylabelsize=8)
plt.show()

f, ax = plt.subplots(figsize=(10,15))
sns.countplot(x="DATE_OF_BIRTH", data=uf, palette="Greens_d")
DOB=uf['DATE_OF_BIRTH'].value_counts()

sns.violinplot(data=uf,y=df4)
plt.show()

df4.hist(figsize=(10, 15), bins=50, xlabelsize=8, ylabelsize=8)
plt.show()

sns.violinplot(data=uf,y=df5)
plt.show()

df5.hist(figsize=(10, 15), bins=50, xlabelsize=8, ylabelsize=8)
plt.show()

sns.violinplot(data=uf,y=df6)
```

```
plt.show()

df6.hist(figsize=(10, 15), bins=50, xlabelsize=8, ylabelsize=8)
plt.show()

sns.violinplot(data=uf,y=df7)
plt.show()

df7.hist(figsize=(10, 15), bins=50, xlabelsize=8, ylabelsize=8)
plt.show()
```

CUR_GPA :

```
: 0.00    2459488
  4.00     454056
  3.00     197666
  3.50     134245
  2.00      93559
           ...
  0.48          8
  0.05          5
  0.04          5
  0.98          3
  0.03          1
  Name: CUR_GPA, Length: 397, dtype: int64
```

DATE_OF_BIRTH:

```
17-SEP-57    1207
11-NOV-58    1180
17-SEP-58    1038
08-SEP-58    1010
25-SEP-59     992
             ...
25-JUN-35       1
01-JAN-39       1
11-MAR-30       1
24-NOV-02       1
06-JUN-16       1
Name: DATE_OF_BIRTH, Length: 27108, dtype: int64
```

STRM:

```
2208    44798
2198    44473
2188    43287
2201    42433
2191    40902
        ...
1241        1
1246        1
1249        1
1251        1
1129        1
Name: STRM, Length: 631, dtype: int64
```

TOT_PASSD_PRGRSS:

```
0.00      154960
6.00      105001
12.00      99684
28.00      70151
27.00      69492
           ...
109.50         1
192.37         1
195.38         1
183.70         1
255.67         1
Name: TOT_PASSD_PRGRSS, Length: 8797, dtype: int64
```

TOT_TAKEN_PRGRSS:

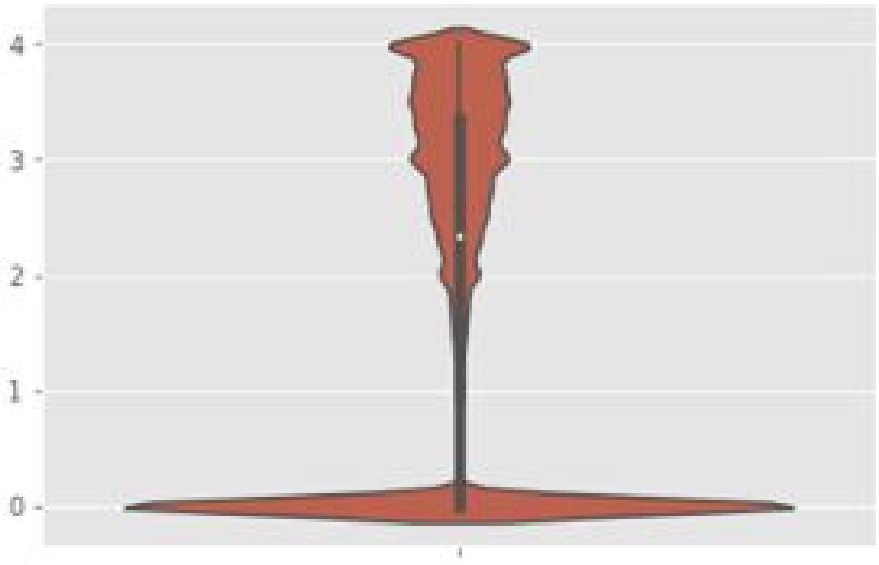
```
12.00     113998
6.00      108929
13.00      86119
28.00      77261
30.00      74704
           ...
181.73         1
236.69         1
91.50          1
243.69         1
163.76         1
Name: TOT_TAKEN_PRGRSS, Length: 10007, dtype: int64
```

TOT_TEST_CREDIT:

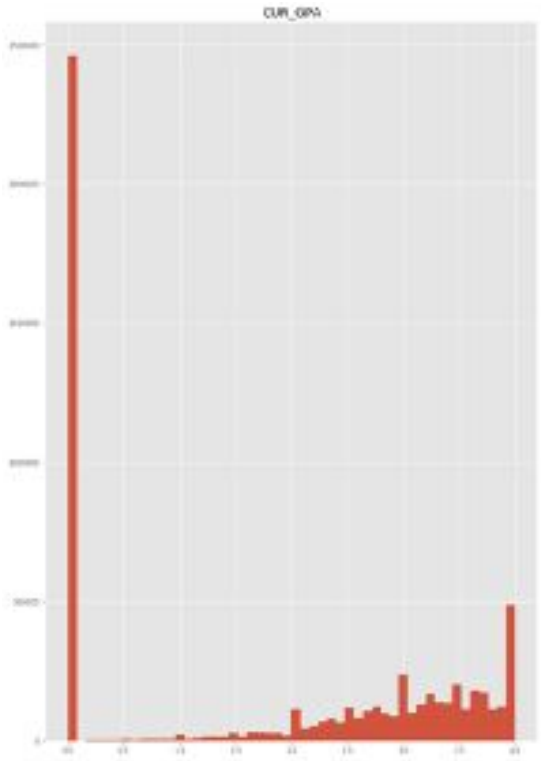
```
0.00     3930180
6.00      288777
3.00      194404
12.00     164567
30.00     157111
          ...
1.34           4
58.00          4
13.68          4
7.66           4
30.70          2
Name: TOT_TEST_CREDIT, Length: 258, dtype: int64
```

Appendix 8: Python Visualizations on the Quality of the Data

CUR_GPA Violin Plot:
Y-axis: All values in this variable, X-axis: Wilder places mean the larger amounts.

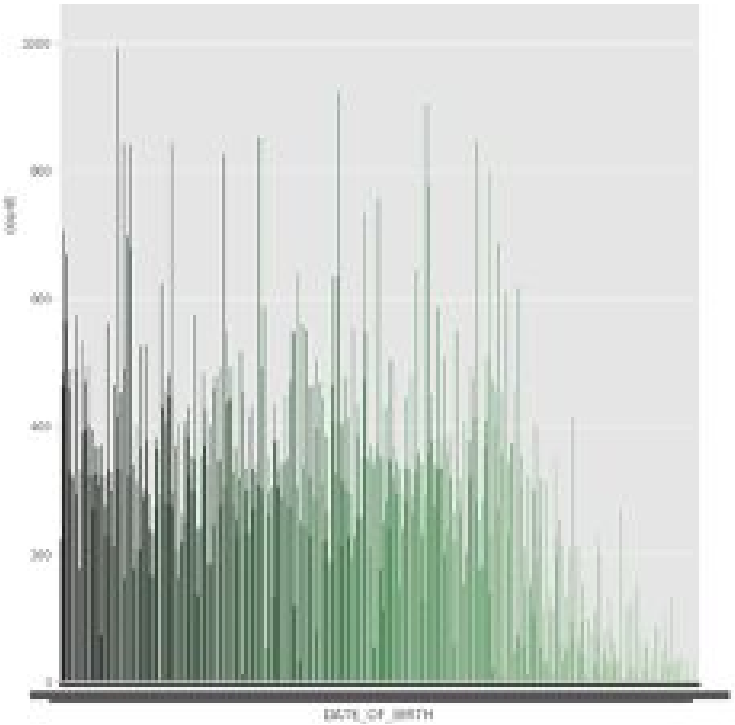


CUR_GPA Histogram:
Y-axis: count of one value in this variable, X-axis: All values in this variable.



Student Success Analysis

DATE_OF_BIRTH: CountPlot

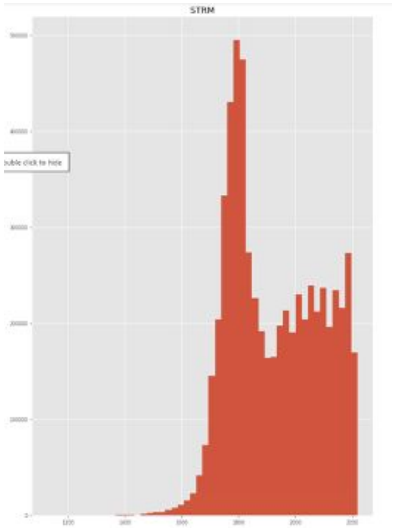


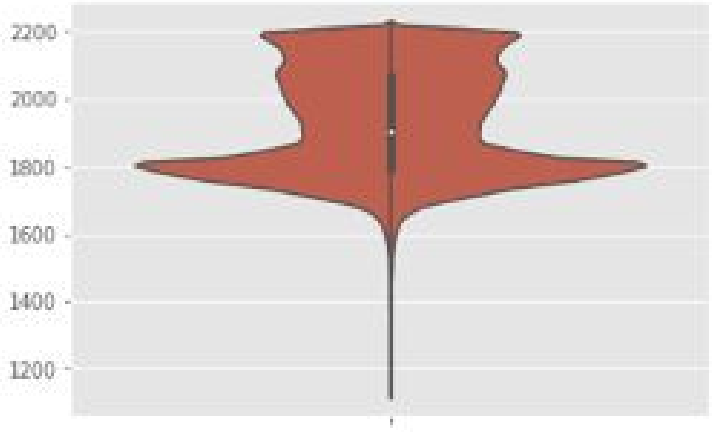
```
[ ]: DOB

[ ]:
```

	17-SEP-57	1207
0	11-NOV-58	1180
1	17-SEP-58	1038
2	08-SEP-58	1010
3	25-SEP-59	992
4	06-OCT-59	976
...	...	...
27102	25-JUN-35	1
27103	01-JAN-39	1
27104	11-MAR-30	1
27105	24-NOV-02	1
27106	06-JUN-16	1

27107 rows × 2 columns

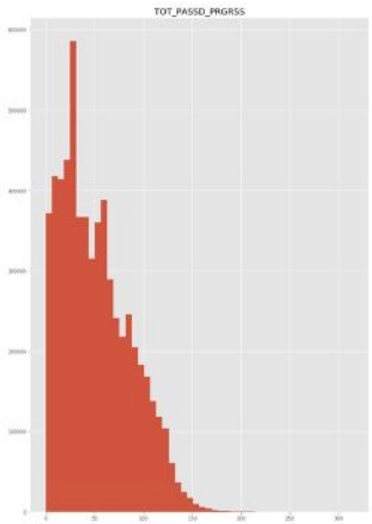
STRM:Histogram & Violin Plot

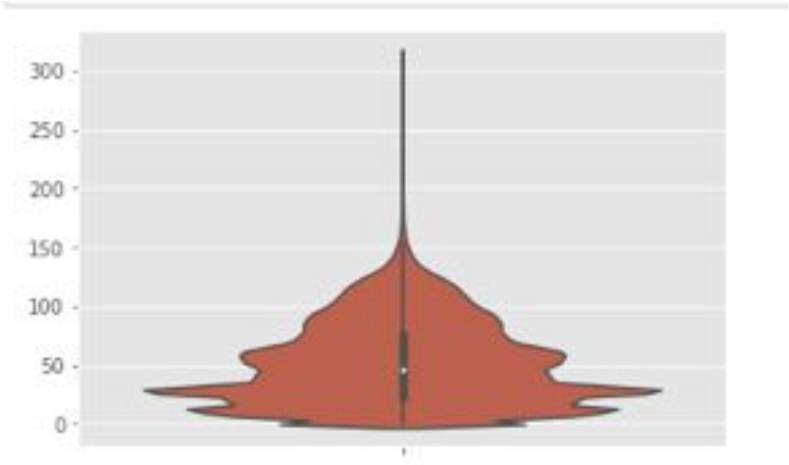




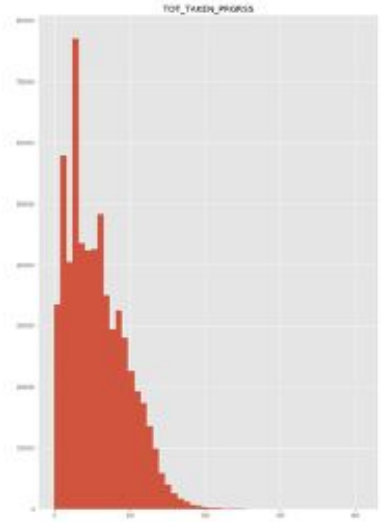
Student Success Analysis

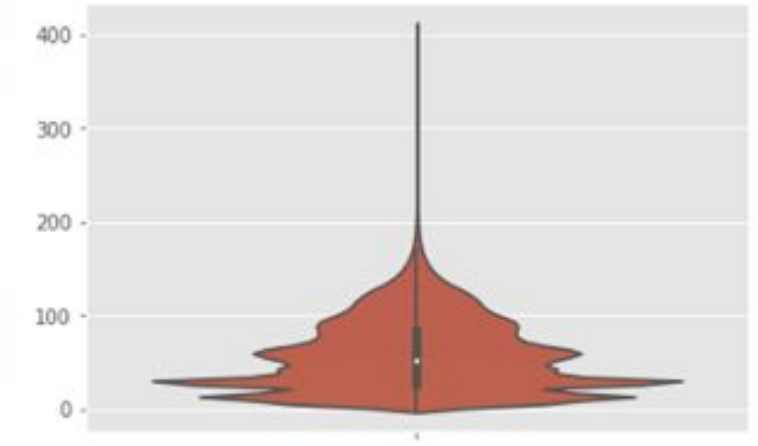
TOT_PASSD_PRGRSS:Histogram & Violin Plot





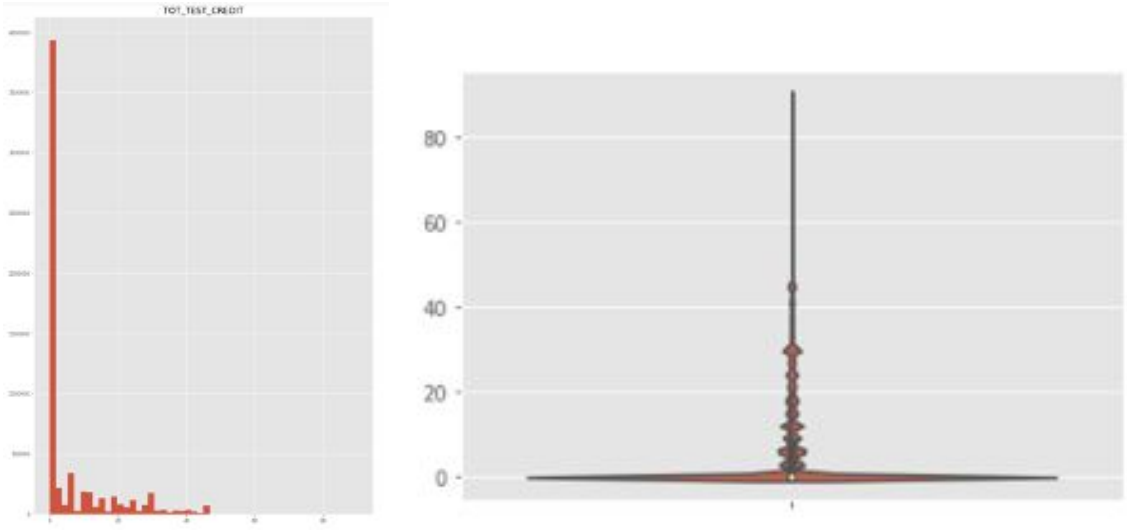
TOT_TAKEN_PRGRSS: :Histogram & Violin Plot





Student Success Analysis

TOT_TEST_CREDIT: :Histogram & Violin Plot



Appendix 9: SQL Summary Counts of New Variables Created Using KMeans Term_End & Term_BEG

By creating new variables that categorized the start and end of the Fall and Spring terms as “Early”, “Average” or “Late” using a K Means methodology it is able to be determined if the term calendar had an impact on the likelihood for student success.

	EARLY	AVERAGE	LATE
TERM_END_DT_SID_SPRING_CATGRY	657440	935462	292170
TERM_END_DT_SID_FALL_CATGRY	648182	997707	341341
TERM_BEG_DT_SID_SPRING_CATGRY	701164	1162689	21219
TERM_BEG_DT_SID_FALL_CATGRY	843782	535677	607770

As you can see from the above chart:

The Fall Semester was well distributed between early, average, and late start dates while leaning slightly on the early side. In contrast, the Spring was much more consistent and the majority of start dates fell between either the early or average category with a very small minority of years having a late Spring term start date.

In comparison regardless of start or end dates the end of the term tended to fall around the same average time period. When there were variations the term-end dates tended to lean on the early side.

Appendix 10: SQL
UF_R2_ANALYSIS_UGRDDRAFT
UF_R2_ANALYSIS_UNDERGRAD

```
CREATE TABLE UF_R2_ANALYSIS_UGRDDRAFT as (SELECT
A.PERSON_SID,
A.ACAD_CAR_SID,
A.INSTITUTION_SID,
A.STDNT_CAR_NBR,
A.TERM_SID,
A.STRM,
A.TERM_BEG_DT_SID,
A.TERM_END_DT_SID,
A.INSTITUTION,
A.ACAD_CAREER,
A.AGE_YEARS,
A.AGE_MONTHS,
A.AGE_DAYS,
A.TOT_CUMULATIVE,
A.JUNIOR_SENIOR_FLAG,
A.TOT_TAKEN_PRGRSS,
A.TOT_TRNSFR,
A.TOT_TEST_CREDIT,
A.TOT_OTHER,
A.TOT_PASSD_PRGRSS,
A.UNT_TAKEN_PRGRSS,
A.CUR_GPA,
A.CUM_GPA,
A.ENRL_CNT,
A.ENRL_FLG,
A.SSR_TRMAC_LAST_DT,
A.ACAD_LEVEL_BOT,
A.ACAD_LEVEL_EOT,
A.UF_CLASS,
A.RESIDENCY,
A.LASTUPD_EW_DTTM,
A.PROF_GRAD_FLAG,
A.ACADEMIC_LOAD,
A.TOT_GRADE_POINTS,
A.TOT_TAKEN_GPA,
A.ENRL_SUMMER_A_FLAG,
```

```
A.ENRL_SUMMER_B_FLAG,
A.ENRL_SUMMER_C_FLAG,
A.ACAD_PROG_PRIMARY,
A.UF_CLASS_EOT,
A.UNT_TAKEN_GPA,
A.UNT_INPROG_GPA,
A.TERM_FIRST_APPT_TIME,
A.TERM_END_DT_SID_CATGRY,
A.TERM_BEG_DT_CATGRY,
A.TERM_LENGTH_CATEGORY,
A.TERM_LENGTH_DAYS,
A.TERM_SEASON,
A.LOW_TERM_GPA_IND,
A.PARTTIME_TERM_IND,
A.NOT_REG_TERM_IND,
A.WITHDRWL_TERM_IND,
A.FULLTIME_TERM_IND,
A.OVR_12HR_TERM_IND,
B.STDNT_GROUP_SID,
B.EFFDT_SID,
B.END_EFFDT_SID,
B.EFF_STATUS,
B.EFF_START_TERM,
B.EFF_START_TERM_LD,
B.EFF_END_TERM,
B.EFF_END_TERM_LD,
B.CURRENT_IND,
B.LASTUPDOPRID,
B.COMMENTS,
C.ACAD_SPLAN_SID
  FROM UF_R1_SUCCESS_ANLS_UDINTSMALL A INNER JOIN
UF_B_PERSON_STDNT_GRP B
  ON A.PERSON_SID = B.PERSON_SID
  AND ((A.TERM_SEASON = substr(B.EFF_START_TERM_LD,0,4) AND
substr(A.TERM_BEG_DT_SID,0,4) = substr(B.EFF_START_TERM_LD,-4))
  OR (A.TERM_SEASON = substr(B.EFF_END_TERM_LD,0,4)AND
substr(A.TERM_BEG_DT_SID,0,4) = substr(B.EFF_END_TERM_LD,-4)))
  INNER JOIN UF_B_TERM_SPLAN C
  ON A.PERSON_SID = C.PERSON_SID
  AND A.TERM_SID = C.TERM_SID
)

CREATE TABLE UF_R2_ANALYSIS_UNDERGRAD as (SELECT
```

Student Success Analysis

```
*
FROM UF_R2_ANALYSIS_UNDERGRADDRAFT A

WHERE (
A.EFFDT_SID = (SELECT MAX(B.EFFDT_SID)
          FROM UF_R2_ANALYSIS_UNDERGRADDRAFT B
                    WHERE A.PERSON_SID = B.PERSON_SID
                AND A.STDNT_GROUP_SID = B.STDNT_GROUP_SID
                     AND A.STDNT_CAR_NBR = B.STDNT_CAR_NBR
                     )
    )
)
```

Appendix 11: Student_term_enrollment_model_v3