Abl Data Science Challenge

Paul Pisani

December 2018

Contents

These slides will cover three key steps in more detail:

- ETL Process
  - Load, clean, transform, and aggregate the data
- Data Exploration
  - Look at various data splits / distributions before diving into deeper analysis
- Analysis / Insights
  - Provide actionable recommendations based on key findings

ETL Process

ETL Process Overview

- **Primary Goals**
  - Prepare raw data files for exploration / analysis by loading, cleaning, transforming, and aggregating key tables
  - Use Python to create a process that is reproducible, quick to implement, and easy to modify for future use

- **Key steps**
  - 1. Load files into Pandas dataframes and check overall contents
  - 2. Fill missing / potentially erroneous values
  - 3. Check field types to ensure data integrity across rows and columns
  - 4. Create derived fields to support deeper analysis / exploration
  - 5. Aggregate tables via joins and aggregate functions (e.g. mean, max, etc.)

- **Takeaways**
  - The files in general are fairly clean, with relatively few data issues to resolve
  - The only minor details that needed to be addressed were two Gender values and some missing Quiz data
  - There is a lot of opportunity for further development on these scripts, but they are a good starting point

Step 1 - Load / Describe Files

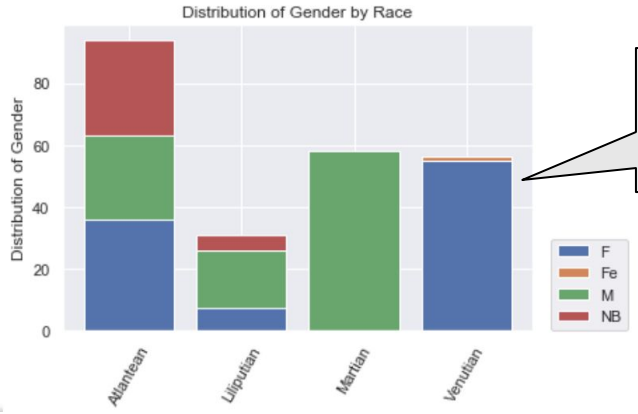
Table Name	Columns	Row Count	Items to Address
groups.csv	GroupType: 4 distinct values (DivGroup, SimGroup, Random, Manual) QuizNumber: 10 distinct values numbered sequentially from 1 to 10 ClassId: 8 distinct values numbered sequentially from 1 to 8 Fields to join on as primary key → ClassId and QuizNumber	80 rows total - 10 quizzes per class* - 8 classes total	Code categorical variables, check field type
students.csv	StudentId: 240 distinct values numbered sequentially from 1 to 240 ClassId: 8 distinct values numbered sequentially from 1 to 8 Teacher: 4 distinct values (TeacherA, TeacherB, TeacherC, TeacherD) Race: 4 distinct values (Martian, Venutian, Atlantean, Liliputian) Gender: 4 distinct values (F, Fe, M, NB) Field to join on as primary key → StudentId	240 rows total - 2 classes per teacher - 30 students per class - Varying distributions across other fields	Fix null / possibly erroneous values for Gender, code categorical variables, check field type
scores.csv	StudentId: 240 distinct values numbered sequentially from 1 to 240 QuizNumber: 10 distinct values numbered sequentially from 1 to 10 Score: Numeric value ranging from 27 to 93 (not sequential) Fields to join on as primary key → StudentId and QuizNumber	2340 total - 9-10 quizzes per student - 1 score per student-quiz	Add rows for 60 students that are missing data for Quiz 8, create derived fields based on score and quiz, check field type

* In both the groups and scores tables, QuizNumber is more appropriately thought of as as week number - see following slides for more detail

Step 2 - Fill Missing / Erroneous Values

- **Missing data**
  - 1 student, a Martian, is missing Gender - we fill with M here since all other Martians are M* (see distribution))
  - 60 students have no data for Quiz 8 - rows with null scores were filled to enable row functions (ex. diff()) later

- **Potentially erroneous values**
  - 1 student, a Venutian, is categorized as Fe (unexpected) - fill with F since all other Venutians are F (only 1 Fe)



Looking at the distribution of Gender by Race, we see than all Martians are M and all Venutians are F (or Fe)

scores_raw: Value Counts For StudentId Column

StudentId	
1	10
2	10
3	9
4	10
5	10
6	10
7	9
8	10
9	10
10	10

Even in this small sample of value counts by student for the first 10 students in scores.csv, we see 2 students with 9 scores rather than 10 as expected

* In a real work example, we should confirm these filled in / corrected values with the client team

Step 3 - Confirm Field Types

- **Field Types By File**
  - While explicit type casting can be dangerous, fields were typed to ensure that no erroneous values exist
  - For example, in the Scores field, it would be concerning if we found character strings (besides NaN)
  - We can use a helper function with try-except clauses to type fields and intelligently recommend transformations
  - Thinking about the range and types of values can be especially helpful before trying to visualize relationships

```
# make sure all fields can be successfully typed

groups_typed = type_fields(groups_raw)
groups_typed.name = 'groups_typed'

students_typed = type_fields(students_raw)
students_typed.name = 'students_typed'

scores_typed = type_fields(scores_raw)
scores_typed.name = 'scores_typed'
```

Side note: Pandas data frame naming is strange / non-intuitive (hence the explicit naming here) - the main use case here is for printing the name of the df as the script walks through validations

```
Successfully cast column GroupType as category!
Successfully cast column QuizNumber as int16!
Successfully cast column ClassId as int16!
Successfully cast column StudentId as int16!
Successfully cast column Race as category!
Successfully cast column Gender as category!
Successfully cast column ClassId as int16!
Successfully cast column Teacher as category!
Successfully cast column StudentId as int16!
Successfully cast column QuizNumber as int16!
Successfully cast column Score as int16!
-> Consider creating bucketed version of Score - many unique / non-seq
uential vals and large range!
```

QuizNumber and ClassId could both arguably be typed as categories - not as important for this exercise but worth noting

Step 4 - Create New Derived Fields

- **ScoreDiff_**
  - Quizzes were given weekly, so quiz over quiz Score change amounts to growth over time (see example below)
  - This score change is one of the key metrics we will examine to better understand the impact of GroupType

- **ScoreBucket**
  - As seen on the previous slide, Score is not sequentially ordered but has a wide range of values (~20 to ~100)
  - This is a good opportunity to create a bucketed version of the field to better understand relationships
  - In this case, we bucket scores into 8 discrete 10 point intervals (e.g. 21-30, 31-40, etc.)

	QuizNumber	Score	StudentId	ScoreBucket	ScoreDiff_1	ScoreDiff_2	ScoreDiff_3	ScoreDiff_4	ScoreDiff_5	ScoreDiff_6	ScoreDiff_7	ScoreDiff_8
0	1	45.0	1	(40.0, 50.0]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2	58.0	1	(50.0, 60.0]	13.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	53.0	1	(50.0, 60.0]	-5.0	8.0	NaN	NaN				
3		5.0	1	(50.0, 60.0]	2	-3.0	10.0	NaN				
			1	(50.0, 60.0]								

Student 1 has a score of 53 on Quiz 3, so ScoreBucket is reported as (50.0, 60.0] - meaning the interval of 51 to 60.

The difference between Student 1's score on Quiz 2 (58) and Quiz 3 is (53), so ScoreDiff_1 is equal to -5.

Two additional notes: ScoreDiff_ fields were created time intervals ranging from 2-10 weeks (numbered 1-9) - for higher time lags, NaNs are filled in cases where the interval of time lag is longer than the number of quizzes taken so far.

Step 5 - Joining Tables

- **Combined View**
  - Tables are joined by the fields called out on Slide 3 as primary keys using merge in Python
  - First, the scores data is left joined to the students data on the shared StudentId column
    - This enables us to look at all student-level related data for each score observation
  - This joined table is then joined a second time to the groups data on the shared QuizNumber and ClassId fields
    - The second join lets us look at which student grouping method was used prior to each score

```
combined_view: Dataframe Info (Entries, Nulls, Data Types)
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2340 entries, 1517 to 571
Data columns (total 20 columns):
QuizNumber          2340 non-null int16
Score               2340 non-null float16
StudentId           2340 non-null int16
ScoreBucket         2340 non-null category
ScoreDiff_1         2040 non-null float16
ScoreDiff_2         1800 non-null float16
ScoreDiff_3         1620 non-null float16
ScoreDiff_4         1380 non-null float16
ScoreDiff_5         1140 non-null float16
ScoreDiff_6         900 non-null float16
ScoreDiff_7         660 non-null float16
ScoreDiff_8         480 non-null float16
ScoreDiff_9         240 non-null float16
Race                2340 non-null category
Gender              2340 non-null category
ClassId             2340 non-null int16
Teacher             2340 non-null category
GroupType           2340 non-null category
ScoreRank           2340 non-null float16
ScoreRank_Bucket    2340 non-null category
dtypes: category(6), float16(11), int16(3)
memory usage: 96.8 KB
None
```

Note: Two additional derived fields were created after joining tables - these calculate a student's score rank with respect to other students in their class for that week / quiz. ScoreRank is the raw rank number (1 to 30), with decimals in cases of ties, while ScoreRank_Bucket is bucketing ScoreRank - 1 for Rank from 1 - 10 ('High' performers), 2 for 11 to 20 ('Mid' performers), and 3 for 21-30 ('Low' performers).

Data Exploration

Data Exploration Overview

- **Primary Goals**
  - Create data visualizations / summaries to better understand distributions and relationships across key entities
  - Highlight notable trends and differences to help inform deeper analysis

- **Key steps**
  - 1. Use aggregations (sum, avg, etc.) to summarize across larger sets of observations
  - 2. Look at distributions where possible to allow for quick comparisons across groups
  - 3. Check both static attributes (e.g. Race) as well as metrics that change over time (e.g. Score)
  - 4. Focus on relationships that can directly inform key questions (see next section)

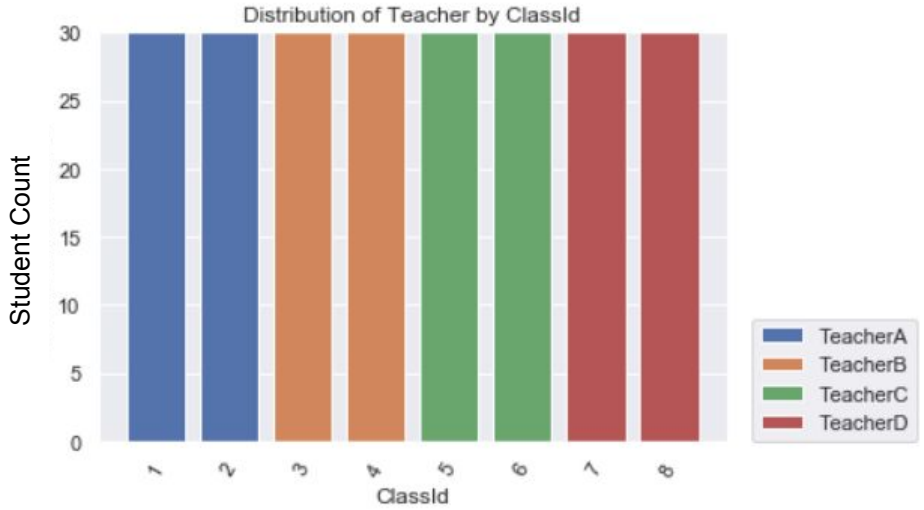
- **Takeaways**
  - Distributions by Race and / or Gender are relatively similar across classrooms and teachers
  - There are some minority populations (e.g. NB, Liliputian) that should be analyzed in more detail
  - GroupType has 80 total observations (8 classes * 10 quiz weeks) and varies substantially across classrooms
  - Scores are generally increasing, though the trend in averages is spiky / volatile

Data Exploration: Distribution of Students by Teacher / Class

- There are four teachers (labeled A through D), each with two classrooms (labeled 1-8)
- Pairings are sequential (e.g. TeacherA has classes 1 and 2, TeacherB has classes 3 and 4, etc.)
- Each classroom has exactly 30 students, so each teacher has a total of 60 students

```
students_typed: Value Counts For ClassId Column
ClassId
1    30
2    30
3    30
4    30
5    30
6    30
7    30
8    30
dtype: int64
```

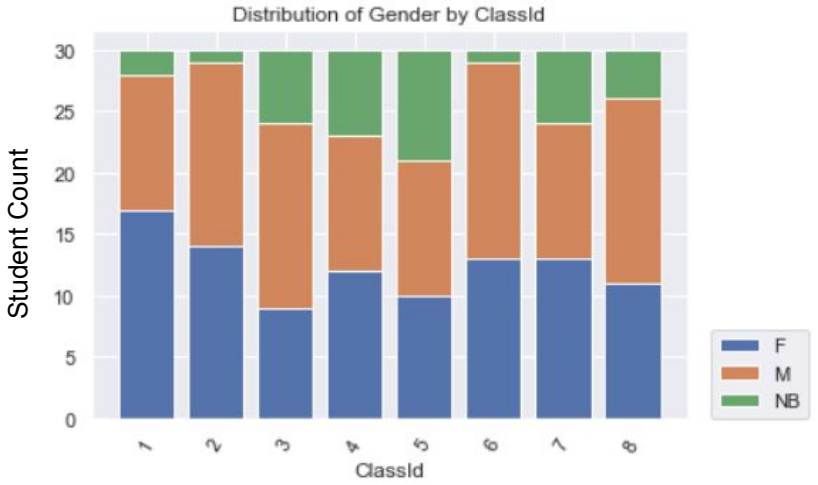
```
students_typed: Value Counts For Teacher Column
Teacher
TeacherA    60
TeacherB    60
TeacherC    60
TeacherD    60
dtype: int64
```



Data Exploration: Distribution of Students by Gender / Class

- In general, distributions of genders look relatively similar across classrooms
- Most classrooms have roughly equal distributions of M and F, and a small handful of NB
- Some minor exceptions
  - Classrooms 4 / 5 have slightly higher populations of NB students (2 and 6 on other end of distr)
  - Classrooms 1 / 2 (TeacherA) have slightly higher populations of F students

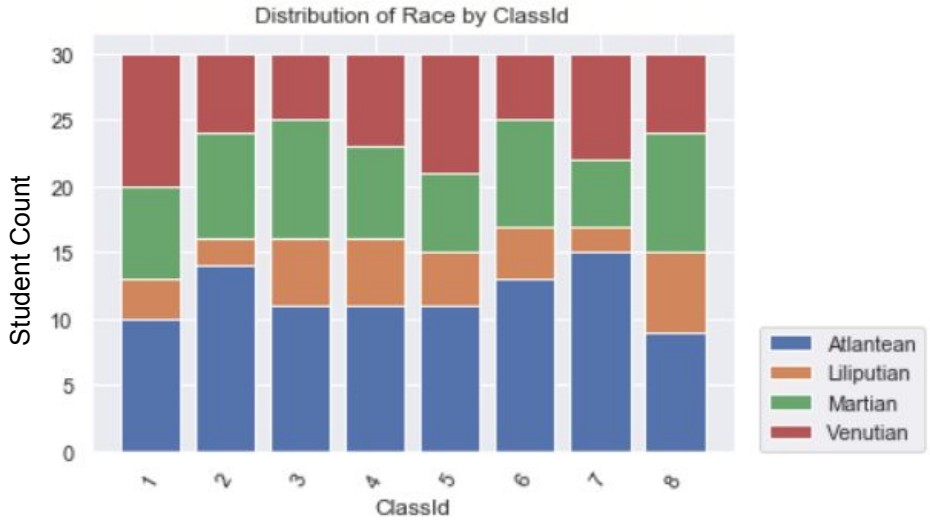
```
students_typed: Value Counts For Gender Column
Gender
F       99
M      105
NB      36
dtype: int64
```



Data Exploration: Distribution of Students by Race / Class

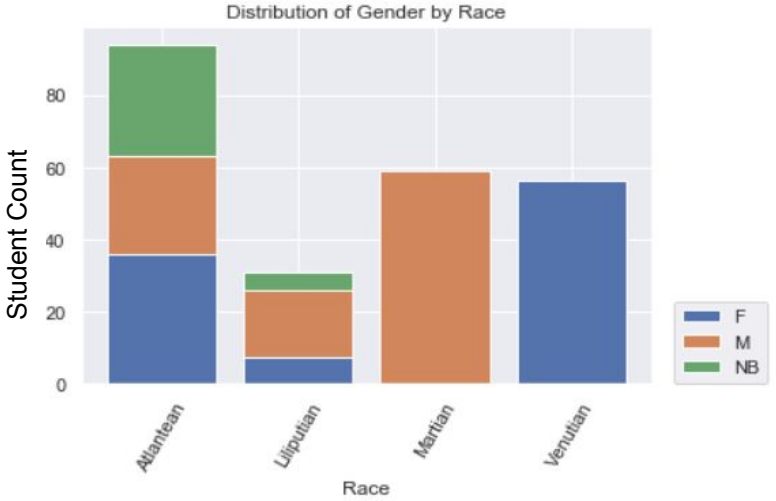
- Similar to gender, the distributions of race look approximately equal across classrooms:
  - ~30-40% of students in most classrooms are Atlantean (exception: Class 1 and 8)
  - Martians and Liliputians both represent ~20% of students
  - Liliputians are the minority in all classrooms, with only a handful of students across all teachers

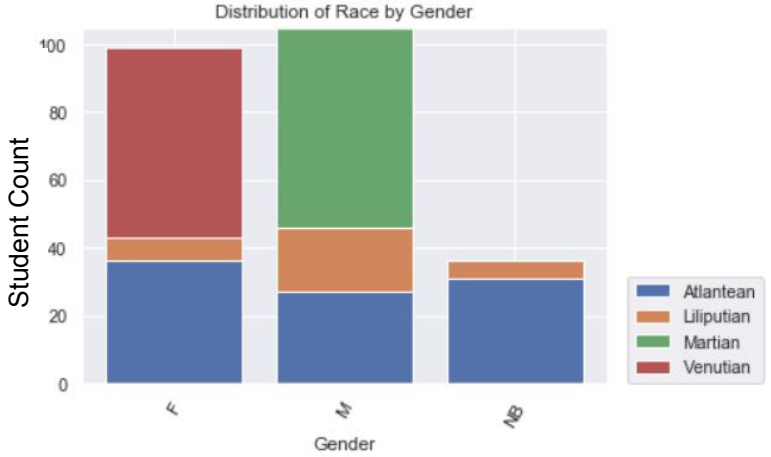
```
students_raw: Value Counts For Race Column
Race
Atlantean      94
Liliputian     31
Martian        59
Venutian       56
dtype: int64
```



Data Exploration: Distribution of Students by Race / Gender

- All Martians are M, all Venetians are F, Atlanteans are an even split, and Liliputians skew towards M
- Nearly all NB students are Atlantean, while F and M students are split across three genders each
- These charts also provide a better visual on count frequency within Race (left) or Gender (right)
    - Race: Atlanteans are most common (~90), while Liliputians are least common (~30)
    - Gender: F and M are both equally common at ~100 - NBs are in the minority at ~40

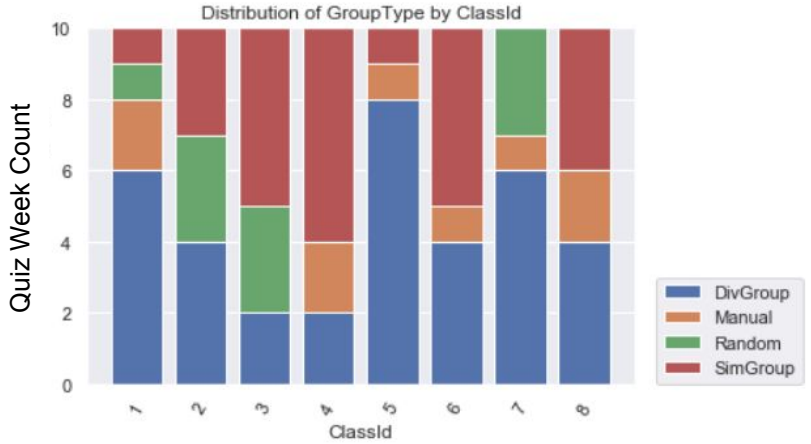




Data Exploration: Distribution of Weeks by Class / GroupType

- In aggregate, DivGroup is most frequent (36 / 80 quiz weeks, or ~40%), followed by SimGroup (~30%)
- Random and Manual groupings are less common, both at around 10% (9-10 / 80 quiz weeks)
- Interestingly, the distribution of group types across classes and teachers varies considerably:
  - There is a large variation in percent of weeks with DivGroup by class (ranges from 20% to 80%)
  - No single class tried all four group types, and only three classes had 2+ Random group weeks
  - Manual groups were used in five different classrooms, but typically only 1-2 weeks

```
groups_typed: Value Counts For GroupType Column
GroupType
DivGroup    36
Manual       9
Random      10
SimGroup    25
dtype: int64
```

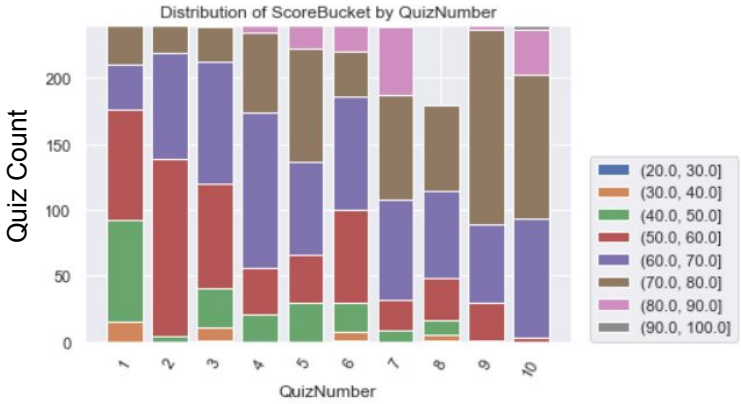


Data Exploration: Distribution of Scores by Quiz / Bucket

- On average, we see students moving into higher score buckets as quiz number increases (over time)
- For example, the % of students scoring 61-80 increases from ~20% of scores to ~80% by Quiz 10
- At the same time, the % of students scoring 31-60 decreases from ~80% of scores to ~0% by Quiz 10

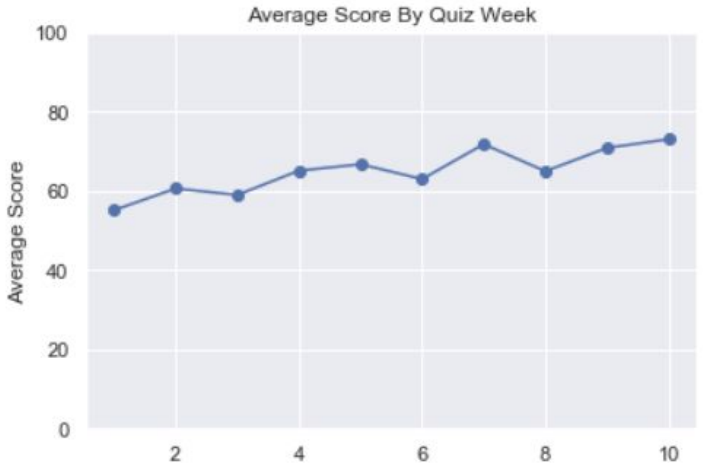
```
scores_typed: Value Counts For ScoreBucket Column
ScoreBucket
(20.0, 30.0]       3
(30.0, 40.0]      35
(40.0, 50.0]     203
(50.0, 60.0]     528
(60.0, 70.0]     775
(70.0, 80.0]     660
(80.0, 90.0]     132
(90.0, 100.0]      4
dtype: int64
```

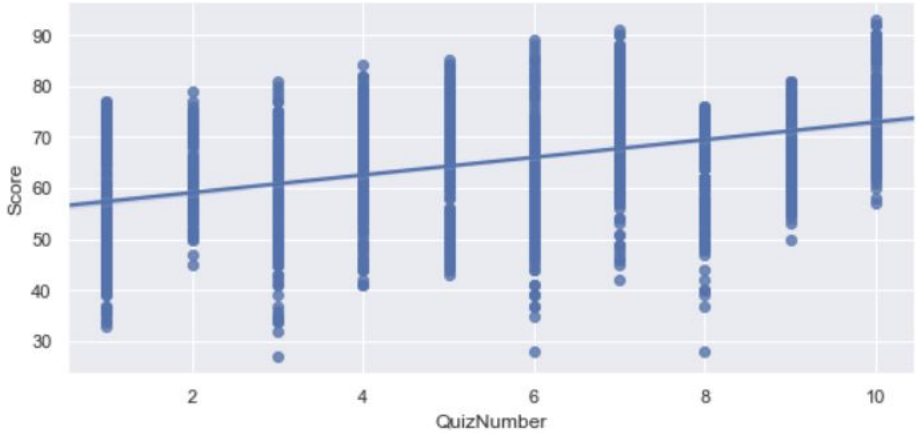
These are aggregate numbers across all 10 quiz weeks, centering around the 61-70 interval - the bar chart on the rights shows changes in this distribution over time / by quiz.



Data Exploration: Average Score By Quiz / Week

- On average, student scores on quizzes climb steadily from week 1 to week 10
- However, looking at week over week changes, there is sizable volatility in scores
- This is likely due to the various group types used across classrooms each week (see next section)





Analysis / Insights

Analysis / Insights Overview

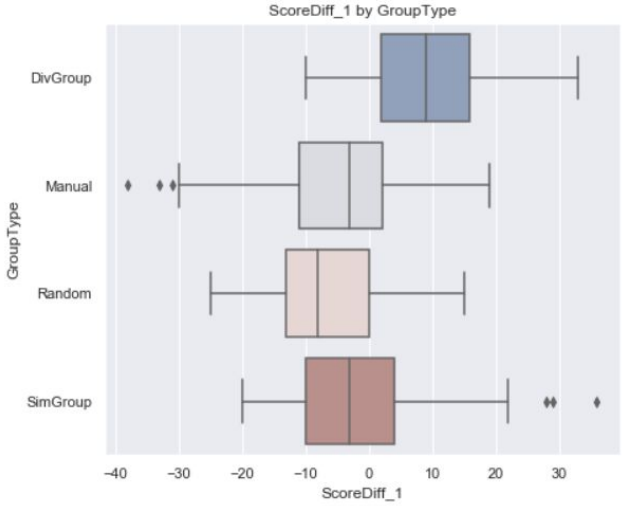
- **Primary Goals**
  - Use what we've learned so far, as well as deeper analysis on primary outcome metrics, to help answer our stakeholders' two key questions:
    - Are there any types of groups that are significantly more or less effective than others?
    - What groups of students, based on common attributes, are consistently underperforming?

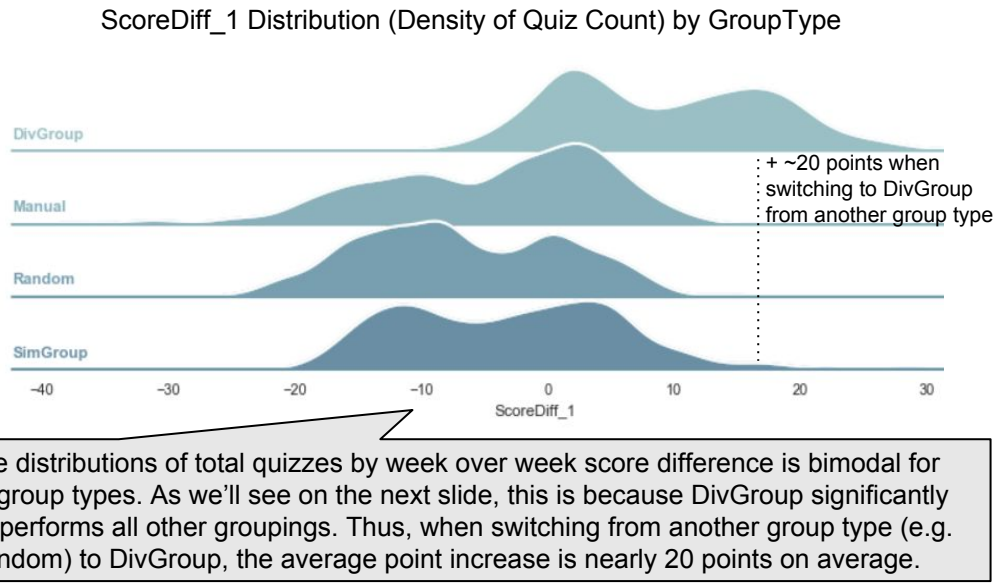
- **Key steps**
  - 1. Create additional fields / aggregations where necessary to drill into results for key metrics
  - 2. Systematically split out these metrics by relevant segmentations and subpopulations
  - 3. Look at varying levels of depth to confirm that results / insights are robust

- **Takeaways**
  - On average, DivGroup significantly outperforms all other student grouping methodologies
  - This holds true across Race, Gender, QuizNumber, Teacher, ClassId, ScoreBucket, and varying time lags
  - Liliputian F, Martian M, and Atlantean M students are consistently scoring below peers on average (+ High)
  - Teacher A's students are also consistently scoring lower than other students (though likely due to GroupType)

Score Growth By GroupType: Overall

- In the boxplot below, it is clear that DivGroup is significantly outperforming other group strategies
- The average week over week score difference for DivGroup is ~15 points higher than the other three types
- Manual, Random, and SimGroup all have negative average week over week growth
- Looking at the ridge plot on the right, we see that these distributions are bimodal - more on this in the next slide

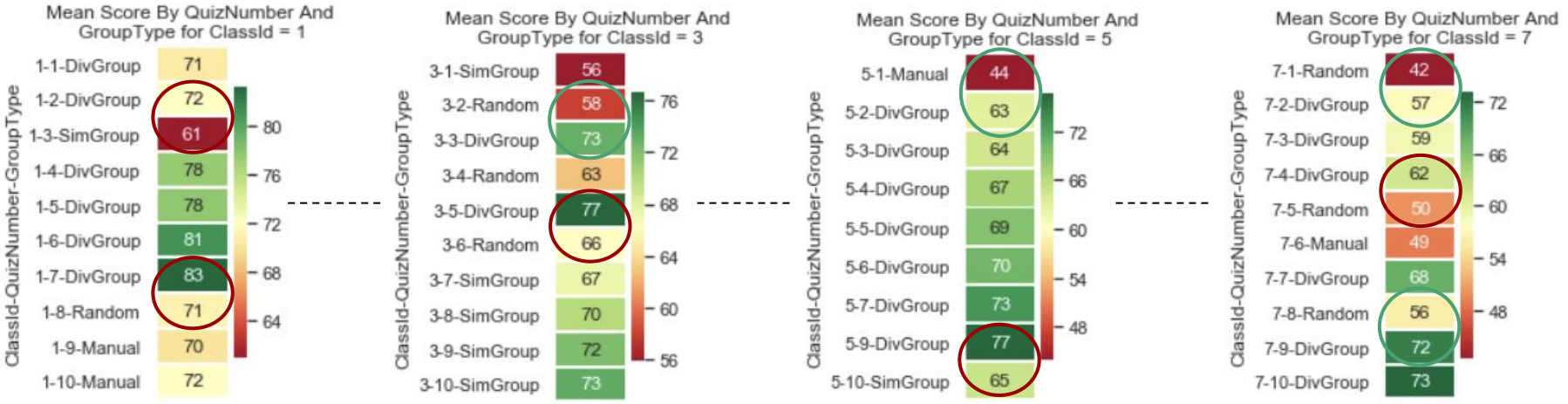




The distributions of total quizzes by week over week score difference is bimodal for all group types. As we'll see on the next slide, this is because DivGroup significantly outperforms all other groupings. Thus, when switching from another group type (e.g. Random) to DivGroup, the average point increase is nearly 20 points on average.

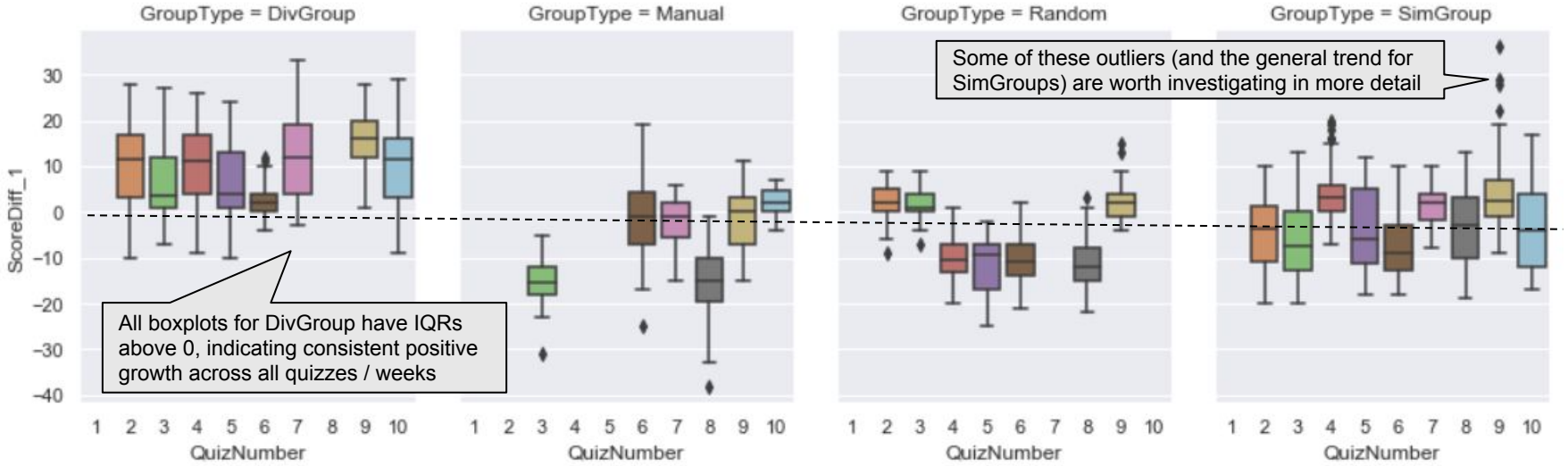
Mean Scores By Quiz Number: Class-Level Detail

- The heatmap tables below record mean quiz scores from Week 1 to Week 10 for a sample of four classrooms (1,3,5,7)
- The group style used in the week leading up to each mean score is also recorded as a table label
- Across these four classrooms, and four different teachers, the same two patterns are immediately apparent:
  - Switching from DivGroup to another group type results in a significant score decrease (or increase in opp case)
  - When the same group type is used for multiple consecutive weeks, there is steady / consistent positive growth



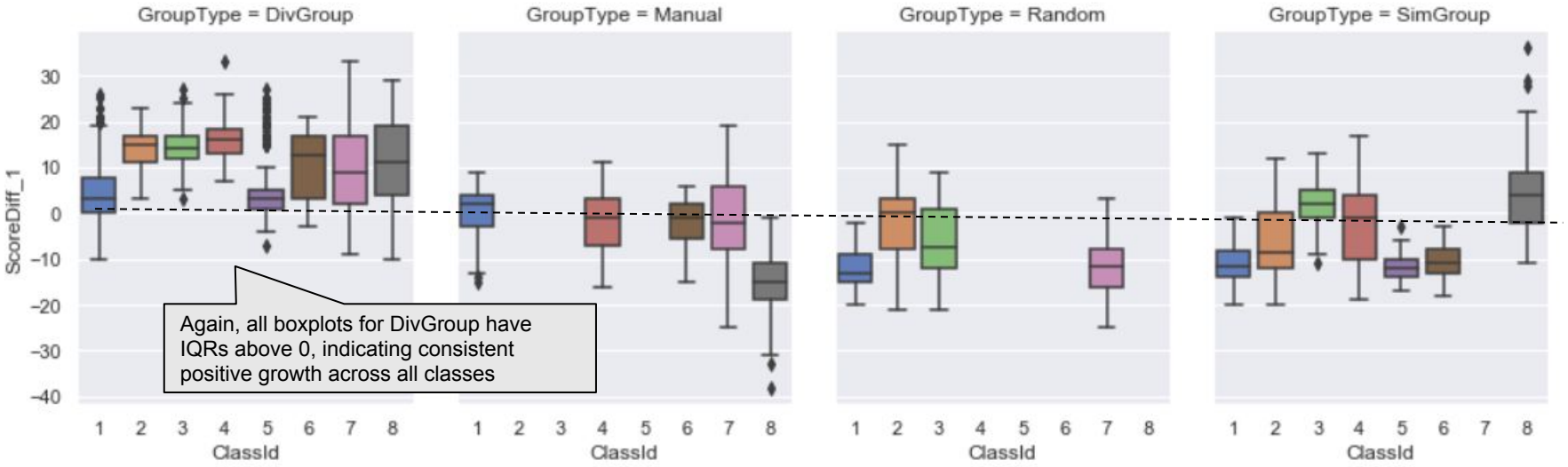
Score Growth By GroupType And QuizNumber

- The average score difference for DivGroups was positive and outperforming other group types across all quiz weeks
- There is no visible trend in DivGroup's effectiveness over time - it universally outperforms
- SimGroup is worth investigating in more detail - it appears to be notable more effective in weeks 7-10
- Hypothetically, this technique becomes more effective as teachers are armed with more data about their students



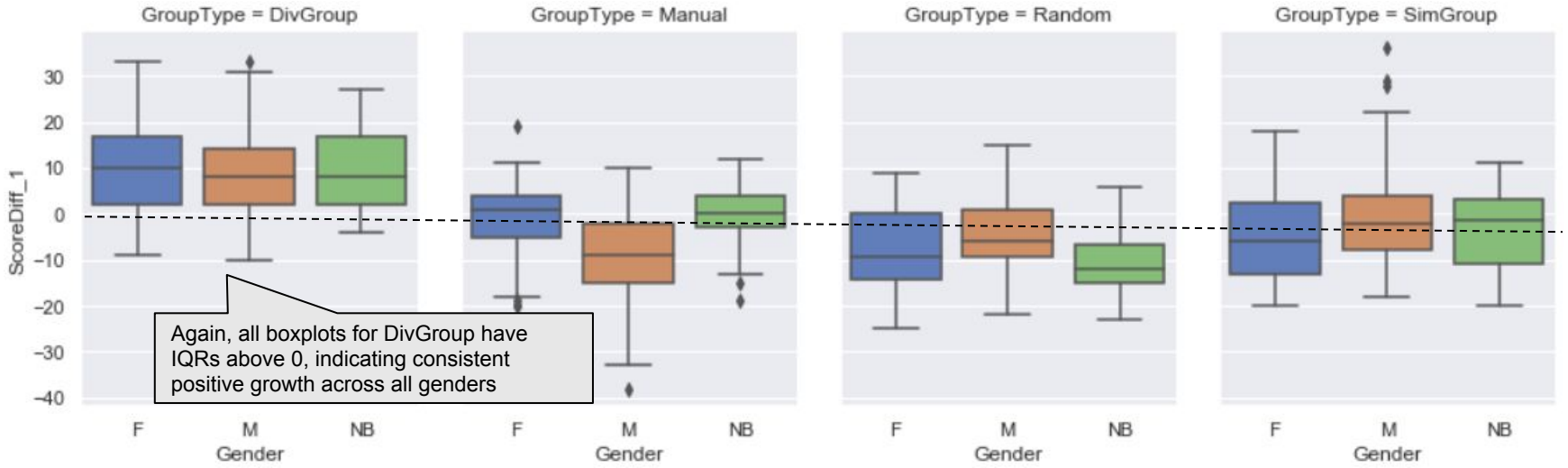
Score Growth By GroupType And ClassId

- Similarly, the avg score difference for DivGroups was positive and outperforming other group types across all classes
- This is in spite of a large positive portion of the bimodal distribution being flagged as outliers (e.g. ClassId = 1, 3, 5)
- Class 8 is worth looking at in more detail with regards to both Manual (significantly negative impact) and SimGroups



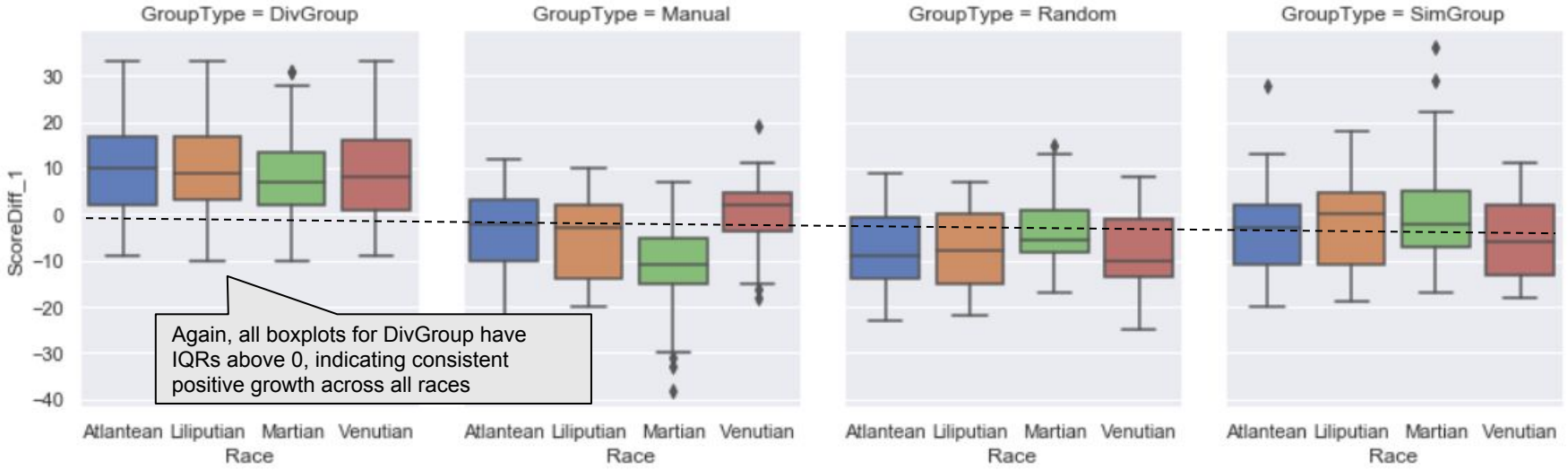
Score Growth By GroupType And Gender

- Again, the avg score difference for DivGroups was positive and outperforming other group types across all Genders
- The distribution of score changes for DivGroup looks similar across genders
- This is not the case for some other group types - Manual, for example, had more negative results for M on average



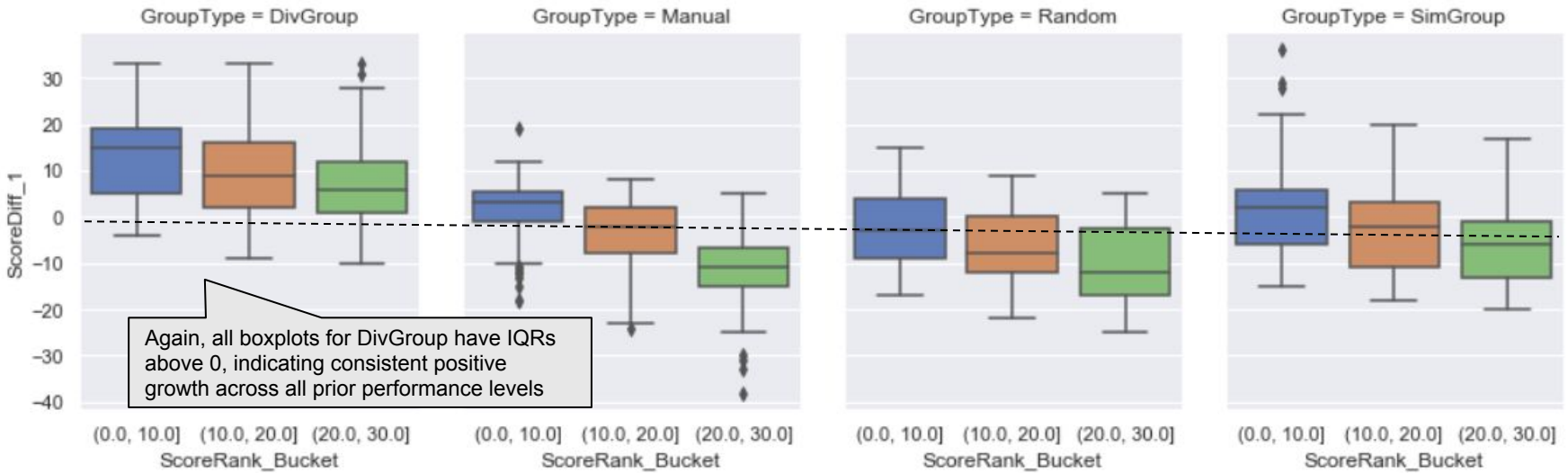
Score Growth By GroupType And Race

- Again, the avg score difference for DivGroups was positive and outperforming other group types across all races
- Additionally, the distribution of score changes for DivGroup looks similar across different races
- This is not the case for some other group types - Manual had more negative results for Martians on average



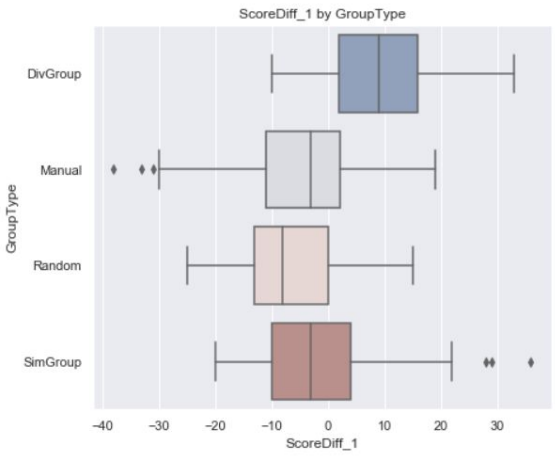
Score Growth By GroupType And Race

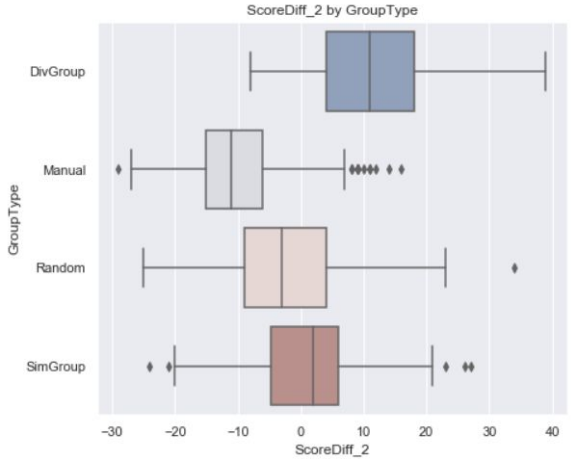
- The avg score difference for DivGroups was positive / outperforming other group types across all previous score tiers
- It is worth noting, however, that across all groups, we see lower score growth for top performing students
- Take a look at the left to right downward movement across all four charts below (Green are High scorers on previous)

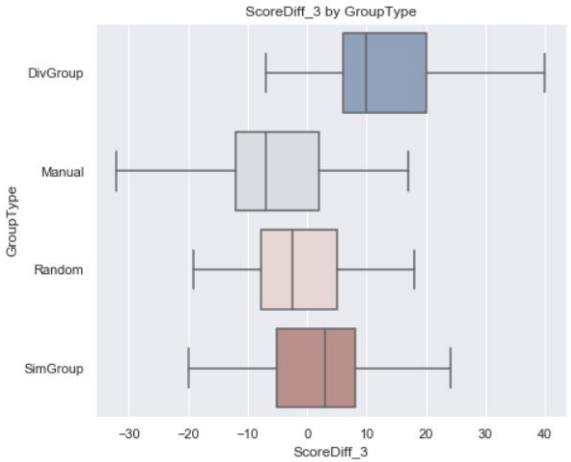


Score Growth By GroupType And Race

- For the bulk of this analysis, we have looked at week over week changes in test scores
- Looking over longer time lag intervals (e.g. 2 weeks, 3 weeks, etc.), the same relationship still holds
- DivGroup outperforms by a significant margin - the bottom quartile is above the upper quartiles for all other types

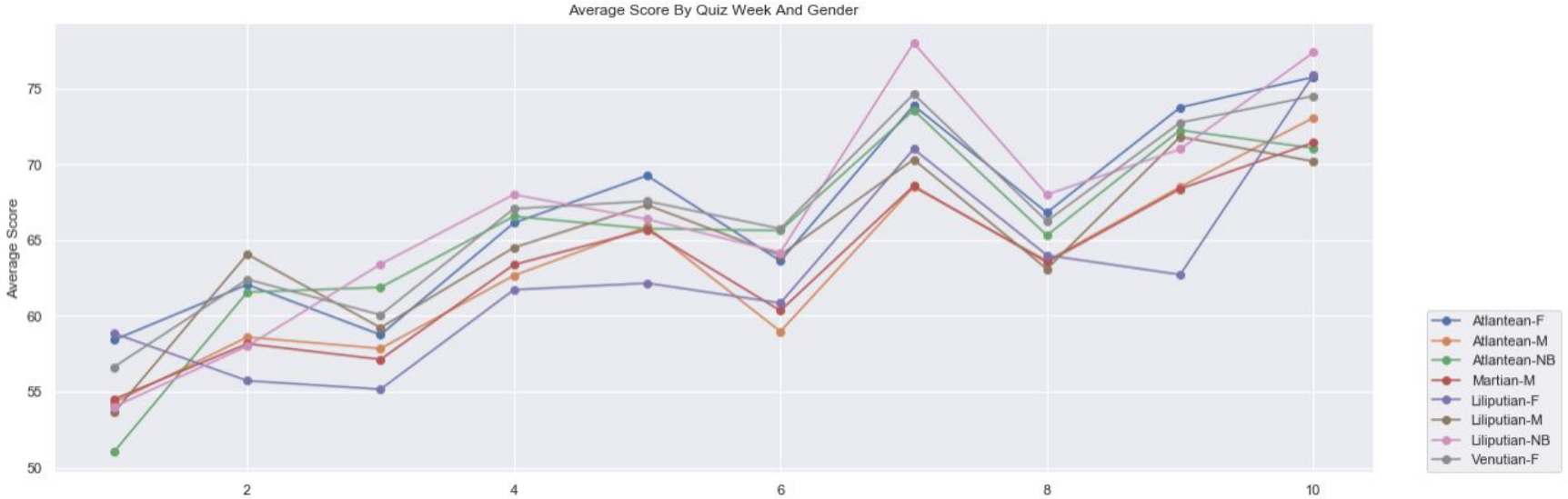






Mean Scores By Race and Gender

- For most of the analysis in this section so far, score growth as a result of group type changes has been the main focus
- Here, we instead look at changes in mean score over time across key student identifiers (e.g. race and gender)
- In the plot below, average quiz scores are plotted by week and student identifier (race, gender)
    - The three main groups that are consistently underperforming are Liliputian F, Martian M, and Atlantean M



Mean Scores By Teacher

- Splitting instead by teacher, it is clear that TeacherD's students are scoring lower on average than other students
- This likely is driven at least in part by TeacherD's selection of group types over the course of the 10 weeks
- Specifically, Class 8 had a 4-week stretch from Week 3 to Week 6 where non-DivGroup types were used