**Tableau Team Assignment 2**

Download the “dognition\_data\_no\_aggregation” data set for this assignment, and the “Dognition Data Set Description” document for a description of the fields that you may see. These exercises will direct you through some more of the analyses outlined in the Dognition analysis plan (provided in Sakai; the format for this analysis plan will be explained later in the course) that you can run on the data set using Tableau.

Please submit one document per team with your team’s answers to each question in this assignment through Sakai. Make sure all of the team members are listed on the top of your submission.

The answers to these exercises will serve as the basis for Quiz 2. Although you should submit your answers to these exercises as a team, **you must take the quiz on your own without any help from anybody else.**

In the text below, all variable names will be depicted in *italics* and all properties on the Marks Card will be depicted in **bold**.

Recall that the metric you are trying to increase is the number of tests customers complete. The data set you used for the last group assignment had one row for each Dog ID, and a column (*Total Tests Completed*) that provided the total number of tests each dog associated with a Dog ID completed. In the data set you will use for this assignment, there is a separate row for each test each dog completed. That provides you with the opportunity to answer some new questions, but it also adds a layer of complexity behind the scenes of your visualizations. This is because the values in *Total Tests Completed* can only be understood at the aggregation level of Dog ID, while other variables you will be interested in have a different level of granularity (such as at the level of individual tests). Pay careful attention to this issue as you work through your analyses. If you run into trouble with your visualizations or calculations, it’s probably because you are trying to combine variables with different granularities, or trying to combine variables that are aggregated at different levels. Remember that Tableau runs on a SQL-like language behind the scenes. Thinking about how you would write a SQL query to get the data you are trying to visualize in Tableau will often help you parse and understand Tableau’s error messages. MySQL\_Exercise\_06, “*Common Pitfalls of Grouped Queries*” from my Coursera course (which you have access to) outlines the types of issues that occur in SQL languages when you try to write queries that have aggregation mismatches, and MySQL\_Exercise\_09, “*Subqueries and Derived Tables*” outlines some of the solutions that SQL languages, including the one Tableau runs on, uses to address those issues.

This data set is a little larger than the data set you used for the last set of exercises, so if you find your computer is running slowly, you may want to try saving your data as an “extract”. For a full description of what extracts are, see:

<http://www.tableau.com/about/blog/2014/7/understanding-tableau-data-extracts-part1>

For a description of how to save extracts, see:

<http://onlinehelp.tableau.com/current/pro/desktop/en-us/extracting_data.html>

If you encounter problems while working through these exercises that you and your team can not solve on your own, you can:

* Review the Tableau videos from this week
* Use resources that you find online about how to make specific types of graphs in Tableau
* Ask questions in the “team\_assignment\_2” folder in Piazza
* Ask the TAs for help

**Exercise 1**

Begin by removing entries you know are not real data. Recall from last week that Dognition’s convention for indicating when an account was a “testing” account was to enter a weight that was completely infeasible for the entered breed. In particular, their testing accounts often contained Shih Tzus that were 190 pounds (Shih Tzus typically weigh between 9 and 16 pounds).

1. In order to remove these data points from your analyses more specifically than you did in the last assignment, create a row calculation that can be used on the Filter shelf to remove all entries of Shih Tzus that weigh 190 pounds. Paste the calculation/code that you used in the “Calculated Field” for this row calculation in your response.

Ans:

﻿IF [Breed] = 'Shih Tzu' THEN (

IF [Weight] = 190 THEN 'Remove'

ELSE 'Keep'

END )

ELSE 'Keep'

END

For the rest of the exercises using the dognition\_data\_no\_aggregation data set, put the calculated field you just created on the Filter shelf to exclude Dognition’s testing accounts from your analyses. You must also exclude the Dog IDs that have a “1” in the “Exclude” field Dognition provided (make sure to keep all the Dog IDs that have a “0” or “Null” value in this field). **Unless stated otherwise, all exercises and quiz questions will assume these two groups of DogIDs have been excluded**.

Now that you can exclude these data easily, take advantage of the new level of detail the dognition\_data\_no\_aggregation data set provides over the aggregated data set you used in the last exercise. Begin by learning more about the games, themselves.

1. Which tests are in which subcategory? Make a table in your workspace that lists the Subcategory name in one column and the test name in the second column. Then go to Worksheet > Copy > Data, and paste the result into your assignment submission (to make it more readable after you paste it, you might want to format the data you paste into your assignment submission so that it appears in 2 columns instead of one).

Ans: Ask Varun

1. Make a bubble chart that displays the total number of records each test category has. Include a title, export the image, and insert it into your assignment submission.
2. Make a tree chart that displays the total number of records each test category has. Include a title, export the image, and insert it into your assignment submission.
3. List two disadvantages common to both the bubble chart and the tree chart for displaying which subcategory has the most records effectively in Tableau.

Next, find out when in the game sequence users tend to drop out. To achieve this, you need to use *Rank by DogID*. It’s worth taking some time to understand exactly what this variable provides. Recall that Dognition customers are given the tests in the same order each time (with a few exceptions that Eliot told us about in the “Meet Your Dognition Data” video). To facilitate your initial analyses, I ordered each test a dog took by its time stamp in *Created At*, and included the rank the test received in this sorted order in its own variable called *Rank by DogID*. A rank order of 1 in this field indicates the test described in that row of data was the first test the dog took, a rank order of 5 in this field indicates that the test described in that row of data was the fifth test the dog took, and so on. I computed a similar rank for all the tests associated with a User ID. This rank, stored in *Rank by UserID,* indicates the order of tests used by each human user. When a User ID is only associated with one Dog ID, the values of *Rank by DogID* and *Rank by UserID* in a row of data will be the same. When a User ID is associated with multiple Dog IDs, the values of *Rank by DogID* and *Rank by UserID* in a row of data may differ. You will have to aggregate these variables appropriately to extract the information you want. The *Total Tests Completed* variable you used when analyzing the dognition\_data\_aggregated\_by\_dogid data set last week was the maximum *Rank by DogID* value associated with each *Dog ID* in the dognition\_data\_no\_aggregation data set. Now, when you need it, you will have to retrieve the maximum rank value associated with a Dog ID yourselves through the level of aggregation you ask for when using *Rank by DogID*.

1. How many records have “Null” in their *Rank by DogID* and *Rank by UserID* fields? How many unique User IDs and Dog IDs are contained in these records with “Null” in their *Rank by DogID* and *Rank by UserID* fields?
2. If you put the Dognition time stamps into your workspace as a blue pill and specify that you want the time stamp to be read at the SECOND level of aggregation, what kinds of values will be shown? If you put the Dognition time stamps into your workspace as a green pill and specify that you want the time stamp to be read at the SECOND level of aggregation, what kinds of values will be shown? Why do you receive different types of values in these two scenarios?
3. Since the bubble and tree charts were not optimal for interpreting data about each test, make a bar chart with the following elements:
   1. It has rank on the x-axis using the most appropriate rank field
   2. It excludes from the visualization the records that have “Null” values in the most appropriate rank field
   3. It has one panel with the number of records on the y-axis, and that color-codes each mark by the field you think best illustrates the reason customers drop out (I suggest that you try color-coding multiple fields in the data set on the bar chart you are making to determine which one might explain when customers tend to stop playing games, but only color-code one field in the chart you submit)
   4. It has a separate panel that reports the % change in the number of tests completed for each rank compared to the previous rank (so, for example, the % change in the number of times 5 tests were completed compared to the number of times 4 tests were completed)
4. Why is there one null value reported for the % change calculation in Q9?
5. Use your own judgment to describe, overall, what characterizes when users tend to drop out. Use the variables in the data set and information from the “Meet Your Dognition Data” video to inform your answer.

**Exercise 2**

In order to better target advertisements or marketing “nudges”, it would be useful to know when customers tend to play the Dognition games.

1. What field should you use for identifying when users play Dognition games?
2. Make a visualization that demonstrates which days of the week customers are most likely to play games.
   1. Make Monday be the first day of the week in your chart, and Sunday be the last day (see <https://www.interworks.com/blog/dwyers/2016/02/01/questions-tableau-training-setting-weekday-start> or

<http://onlinehelp.tableau.com/current/pro/desktop/en-us/help.html#date_properties.html> ).

* 1. Choose the graph type that you think most clearly illustrates the information.
  2. Make sure the caption of the picture describes all the data that you are excluding.
  3. Insert the resulting image into your group submission

1. Make a visualization that demonstrates which hours of the day customers are most likely to play games.
   1. Make the labels represent a 12-hour clock that displays an uppercase AM with any hour before noon and an uppercase PM with any hour between noon and 11:59 PM (see:
      1. <https://www.interworks.com/blog/dwyers/2016/02/01/questions-tableau-training-setting-weekday-start>
      2. <http://kb.tableau.com/articles/howto/changing-continuous-time-display-from-military-to-12-hour-am-pm-format>
   2. Choose the graph type that you think most clearly illustrates the information.
   3. Make sure the caption of the picture describes all the data that you are excluding.
   4. Insert the resulting image into your group submission

Look carefully at your visualizations that display which hours of the day customers are most likely to play games. Does anything look suspicious to you? It should! The data currently make it look like a lot of games are played in the middle of the night, but that doesn’t make sense. Look at the raw data behind your graphs to get a feeling for what might be going on with this field of data.

1. Are people really playing games in the middle of the night? If not, why does the data make it appear that people are playing in the middle of the night?

Unfortunately, it is not very easy to adjust the time stamps the way we need to within Tableau’s built-in date functions (or within most programs, for that matter). The best way to accomplish such a correction would be to find or assemble a separate data set that would provide the appropriate correction for every entry in one of our location-related variables, and then blend that secondary data set with dognition\_data\_no\_aggregation. Even more unfortunately, such data sets are not easy to find. For this exercise, we will focus on customers in the United States only. I have assembled a data set that provides the corrections you need for a collection of United States zip codes. Connect to the file called dognition\_data\_no\_aggregation\_with\_zip\_code\_correction from the course website in Tableau. This excel file has the original dognition\_data\_no\_aggregation data in one worksheet called “master table”, and the correction data organized by zip code in a separate worksheet called “time zone correction”. On the initial data connection screen, drag both worksheets to the box where it says “drag sheets here” (it might take a while for the data to fully load). You will need to join these worksheets together to be able to correct the time stamps of when each game is played in the Tableau workspace.

1. Explain why you need to join the two worksheets, and list which field(s) you will use to join them.
2. Given what you have learned about SQL, should you use an inner join, a left join, or a right join? If you choose left or right join, which table will be on the left and which table will be on the right? If you need reminders about how joins work, refer to MySQL Exercises 7 (*Inner Joins*) and 8 (*Joining Tables with Outer Joins*) in my Coursera course.

You should see that the worksheet names are connected with a line and two circles in the space on the right. Click on the circles to make sure that the tables in the worksheets are being joined on the correct field and with the type of join that you intend. Then go to a Tableau worksheet. You will see that *Diff from UTC*, which is the variable we essentially imported through the join we just implemented, is in its own “time\_zone\_correction” heading in the variables pane. You can now use this variable as any other variable.

1. Create a new row calculation that will correct each tests’ time stamp using the information you joined in from time zone correction worksheet. (Note that dates have their own unique functions for adding and subtracting values in Tableau, so choose your function carefully. You are welcome to look for help on the internet about what function to use.) Check the calculation in your workspace to make sure it is giving you the output you anticipate, and paste the calculation in your response.
2. Make a visualization for the US data that demonstrates which hours of the day customers are most likely to play games, this time with your new adjusted test time stamps. Show the hours on the x axis, segment the data according to month, and exclude any Null values in the month field. Sort each part of the graph so that the data for each month is sorted according to which hour has the most completed tests (the hour with the most completed tests should come first/be in the left-most position), and only show the data for the 3 hours with the most tests in every month.
   1. Make the labels represent a 12-hour clock that displays an uppercase AM with any hour before noon and an uppercase PM with any hour between noon and 11:59 PM
   2. Format the graph so that there are vertical lines between every month, but not between every hour.
   3. The only text that should be at the top of the graph is the heading for each month and the title. Remove any other text.
   4. Make sure the caption of the picture describes all the data that you are excluding.
   5. Export the image and insert it into your group submission.
3. When do US customers tend to play the most games, and what suggestions might you make to Dognition about how their business goals based on these observations?

**Exercise 3**

In the “Meet Your Dognition” video, we heard Eliot describe the story of how Dognition tried giving the Dognition tests to some customers in a different order than they typically do now. The hope was that doing so would increase the number of tests users completed overall (it didn’t, according to Eliot). During discussions off camera, Eliot shared some information about other experiments Dognition has implemented as well. In particular, Dognition periodically tries offering a “Free Start” promotion that gives customers the first four tests for free. The hypothesis (or hope) would be that once potential customers get a chance to experience the product first-hand, they will be more likely to buy a subscription. In this exercise, you should assess whether the “Free Start” promotions worked.

*Free Start User* indicates whether the user began their Dognition experience with a free start. Currently *Free Start User* has three possible values: 1, 0, and NULL. The Dognition team confirmed that the 0 and Null entries should be considered the same group, and that neither type of entry represents free starts. Given this information, make a grouped variable from *Free Start User* called “Cleaned Free Start” that has just two options: “Free Start” or “No Free Start”.

1. Make a bar chart that shows the number of unique Dog IDs in the “Free Start” category for each year and month of data. Use the original *Created At* field for this plot, not the corrected one, because we do not want to throw out all the non-US data.

Sort the bars so that the year/month combination with the greatest number of free start unique Dog IDs across the entire data set is on all the way on the left side of the chart, and the year/month combination with the least number of free start unique Dog IDs across the entire data set is all the way on the right side of the chart (NOTE: that means that the bars will NOT go in chronological order; rather, they will be ordered by the total number of free start unique DogIDs in each year/month). The labels on the bottom part of the x axis should represent the appropriate month of the year/month combination. The labels on the top part of the x axis should represent the appropriate year of the year/month combination.

HINT: The easiest way to implement the correct sorting for this chart is to combine date calculations that output the appropriate “year” and “month” for every row of data with a strategy typically used to implement a “nested sort”. Information about nested sorts can be found here:

<https://onlinehelp.tableau.com/current/pro/desktop/en-us/sortgroup_sorting_computed_ex4sorted.html>

Nested sorts usually use combined fields, as described in the link above. The sort I am asking you to do is NOT nested, but you can still use the combined field strategy. Once you put the combined field you designed onto one of your shelves, right click the pill and choose the sort option. When the sort panel appears, use the “Field” option in the “Sort by” box to specify what fields and associated aggregation you want to use to sort the marks.

Once sorted, color-code the bars by what year they are associated with. Choose Tableau’s colorblind-friendly palette for your colors, and make the colors associated with each year be as distinct from each other as possible.

Export the image of the graph you made and insert it into your group submission. In addition to all the items and characteristics described above, make sure that:

* 1. You have formatted the graph so that there are no gray vertical lines between each bar.
  2. The only text that appears at the top of the graph is the title and the year associated with each bar. The only text that appears at the bottom of the graph is the month associated with each bar. Remove any other text.
  3. The caption of the picture describes all the data that you are excluding.

1. What year had the fewest free start trials? Did Dognition try free-start trials throughout the three years, or just in one year?
2. Do you think “Free Starts” improved the number of tests customers completed? Explain your answer and provide a visualization to support your conclusion. This time, rather than tell you exactly what chart to make, I am asking you to provide whatever visualization you think best illustrates your argument.

**Exercise 4**

In this exercise, we are going to use table calculations to determine how we could use Tableau to dynamically recreate the ranks stored in the *Rank by DogID* and *Rank by UserID* variables. These variables, remember, chronologically rank each test according to its associated time stamp in *Created At*, and restart the ranking process either for every dog or for every customer.

To begin, make a group of Dog IDs that only has a few Dog IDs included (randomly choose 4 to 5 of them, preferably with different amounts of tests completed). Use your new grouped variable and the filter shelf to ensure that you only analyze the data from the few dogs you chose in your workspace. This will make troubleshooting your calculations much faster.

Create a text table with *Dog ID* in the left-most column, *Test Name* in the column after that, and *Created At* in the right-most column. Now create a column in between the *Test Name* and *Created At* columns that uses a table calculation (that does NOT use *Rank by DogID* – you will not get points for this exercise if you use *Rank by DogID* in your table calculation!) to indicate the ascending ranked order each test was completed, sorted from 1 to the last test taken. Call the new column “DogID Rank by Hand”. Pay attention to the aggregation functions you use in your calculation.

1. What calculation did you use? Paste the full syntax of your calculation in your response and explain why you chose the aggregation you did.

Sort your table so that the rows for each unique DogID are ordered in ascending order by their DogID Rank by Hand. Now include a column between your newly created *DogID Rank by Hand* and the value of *Created at* that reports the rank stored in the original *Rank by DogID* to illustrate that your table calculation arrived at the same value as *Rank by DogID*. If there are asterisks in your table, make sure to fix your calculations so that no asterisks are present.

1. Export the image of the table you made and insert it into your group submission. Make sure you include the default caption.

Now make a group of User IDs with 1-2 User IDs that have multiple unique Dog IDs associated with them. Use your new grouped variable and the filter shelf to ensure that you only analyze the data from the few users you chose in your workspace. Create a text table with *User ID* in the left-most column, *Dog ID* in the next column, *Test Name* in the column after that, and *Created At* in the right-most column. Now, like in Q23 and Q24, create a column in between the *Test Name* and *Created At* columns that uses a table calculation (that does NOT use *Rank by DogID* – you will not get points for this exercise if you use *Rank by DogID* in your table calculation!) to indicate the ascending ranked order each test was completed, sorted from 1 to the last test taken. Call the new column “DogID Rank by Hand”. Pay attention to the aggregation functions you use in your calculation. Sort your table so that the rows for each unique DogID are ordered in ascending order by their DogID Rank by Hand. Now include a column between your newly created *DogID Rank by Hand* and the value of *Created at* that reports the rank stored in the original *Rank by DogID* to illustrate that your table calculation arrived at the same value as *Rank by DogID*.

In addition, create a column in between the *Rank by DogID* and *Created At* columns that uses a table calculation (that does NOT use *Rank by UserID* – you will not get points for this exercise if you use *Rank by UserID* in your table calculation!) to indicate the ascending ranked order each test was completed by each unique User, sorted from 1 to the last test taken. Call the new column “UserID Rank by Hand”. Now include a column between your newly created *UserID Rank by Hand* and the value of *Created at* that reports the rank stored in the original *Rank by UserID* to illustrate that your table calculation arrived at the same value as *Rank by UserID*. If there are asterisks in your table, make sure to fix your calculations so that no asterisks are present.

1. Export the image of the table you made (it should have 7 columns in total) and insert it into your group submission. Make sure you include the default caption.

**Exercise 5**

In this exercise, we are going to use table calculations and a type of calculation called “level of detail” calculation to determine how we could use Tableau to dynamically recreate the numbers previously stored in the *Total Tests Completed* variable of the dognition\_data\_aggregated\_by\_dogid data set, as well as some of the types of visualizations you made in the previous assignment with the *Total Tests Completed* variable. You will also use a calculation to dynamically change the colors of marks.

To begin, read about the concept of level of detail (LOD) calculations here:

<http://www.tableaulearners.com/2016/level-detail-expressions-tableau/>

This is a helpful blog post that helps demystify LOD calculations by relating them to sub-queries in SQL:

<https://www.syntelli.com/tableau-level-of-detail-expressions-introduction-and-include>

Your goal for the chart you will be making is to plot the median number of tests completed by customers in the US, Canada, Great Britain, and Australia. To do this, you need to create an LOD calculation that outputs the number of test records created by each Dog ID (see the reference above). Test your calculation by making a table that has *Dog ID* in the left column and the result of your LOD calculation in the right column. Then look at the raw data behind each number to make sure the calculation output makes sense.

1. Paste the full syntax of your calculation in your submission.

If constructed correctly, you can use this calculation just like you would any other field. Make a bar chart with country on the x-axis (but restrict your view to the countries listed above), and sort the x-axis labels so they appear in this order (using the manual sort option in the sort panel for country field): US, Canada, Great Britain, and Australia. Your resulting chart should give you the same results you got in Q14 of the first set of Tableau Group Exercises from last week.

Next, you are going to (a) calculate what the median of all of the countries in the workspace are, and (b) color-code each mark according to whether the median from that country is more or less than the overall median for all the countries in the workspace. (NOTE: yes, I am asking you to take a median of medians here. This is not a statistically valid thing to do, but it is fine for the purposes of this exercise.) Once you have confirmed that the calculation is giving you what you intend, create a “window” table calculation called “Window\_Median\_Completed\_Tests” that computes the overall median for all of the outputs of your LOD calculation in your workspace. Put this new *Window\_Median\_Completed\_Tests* on the detail shelf and hover over items in the workspace to make sure the median it outputs make sense. (If not, you might have to adjust the details of what fields are being used to compute the table calculation.) Once you have confirmed it is correct, add a row to your visualization that illustrates the difference between the value of the LOD calculation and *Window\_Median\_Completed\_Tests*. If a country’s median number of completed tests is higher than the overall median for all the countries in the view, the output of this calculation for that country should be positive. If a country’s median number of completed tests is lower than the overall median for all the countries in the view, the output of this calculation should be negative. Once you have verified that the output of the calculation is correct, make the marks for this second row circles instead of bars.

Now you are almost done. The final steps require you to create one more calculation. In this calculation, use IF/ELSE/ELSEIF statements and the names of your newly created variables to output the value “green” if a country’s median number of completed tests is higher than the overall median for all the countries in the view, and the value “red” if a country’s median number of completed tests is lower than the overall median for all the countries in the view. Place this calculation on **Color**. Then edit the colors assigned to each value of the calculation output so that the value “green” is assigned to a green color, and the value “red” is assigned to a red color. The final result should be a visualization with:

1. bars representing the median number of completed tests for US, Canada, Great Britain, and Australia, respectively, in the top row
2. circles whose values represent the difference between each country’s median number of completed tests and the overall median of all the countries in the view, and whose colors represent whether this difference is positive or negative

Expand your visualization so that it includes China, Norway, Italy, New Zealand, Brazil. Edit the labels of the y axes so that they accurately depict what is shown. Edit the title and labels of the color legend.

1. Export the image of the visualization you just made and insert it into your group submission. Make sure you include the default caption that says what data was excluded.