DATA 115 Excel Functions

This document lists some of the functions in Excel that we used in lecture and gives examples of their syntax and uses.

- Accessing Cells:
  - Relative and Absolute referencing - To access the contents of an individual cell you first use the column label (letter) and then the row label (number). So `=A3` places the value that is currently in cell A3 in the highlighted cell. If you place a dollar sign between the letter and number like `A$3`, then the chosen cell won't change even if you transport the equation to another position (say by dragging the cell down a column to extend it to the entire dataset).
  - Ranges - To access a bunch of consecutive elements in a row or column you can use a colon, so `C2:C35` selects an array consisting of the second through thirtyfifth elements of the third column, while `C2:H2` selects six consecutive elements in the second row. You can also select rectangular arrays this way. For example, `C2:H35` starts at the C2 cell and the collects all the cells to the right and below until it reaches H35.
  - **INDEX** The index function takes two inputs - an array and a position - and returns the value that is stored in that position in the array. For example, `INDEX(B3:B10,4)` returns the value stored in B6.
  - **XMATCH** The XMATCH function returns the position of a value within an array. For example, if the C column has the integers 10,20,30,40,... stored in its rows, the command `XMATCH(30,C1:C20)` wourld return '3'.
  - **INDEX(XMATCH)** Index and Xmatch are often used together to find ways to combine data sets - we saw this in class with the dental records example. In that case we used Index to find the entry corresponding to the position found by match. The usual syntax for this kind of operation looks like: `INDEX([array whose values are being added to the previous data set],XMATCH([unique id of the row we are matching], [array of the unique ids in the new data]))`, for the in-class example, that was something like: `INDEX(K$2:K$5,XMATCH(A2,J$2:J$5))`.
- Arithmetic
  - **SUM** The sum function takes as input an array and returns the sum of all the values in that array.
  - **PROD** The prod function takes as input an array and returns the product of all the values in that array.
  - **MIN** The min function takes as input an array and returns the smallest value.
  - **MAX** The max function takes as input an array and returns the largest value. As we saw in the beer reviews example, these functions work on dates as well as numbers.
- Conditionals (remember that conditional expressions represent 'if-then' type statements)
  - **COUNTIF** The countif function returns the number of values in an array that satisfy some condition. It takes two arguments, first the array and then the condition the values should satisfy. For example, `COUNTIF(B34:K34,"> 0")` returns the number of positive numbers between B34 and K34. Other types of expressions for the condition might include the equals sign or the less than sign, always wrapped in quotation marks.

- **SUMIF** The sumif function functions identically to the countif function, except that it adds up the numbers that satisfy the condition instead of counting them.
- **IF** The if function takes three arguments. The first is the conditional that gets evaluated, followed by what to place in the cell if the conditional is true, and then finally what to put in the cell if the conditional is false. In class, we used this to assign prices to the dental procedures with `A2="Cleaning",10,200` but as with the previous functions there are many other types of conditionals that can be evaluated.
- **ISNUMBER** This returns TRUE if the given cell has a numeric type and FALSE otherwise.

- String Formatting
  - Capitalization: When talking about string formatting we covered the UPPER, LOWER, and PROPER functions which converts all the string characters in a cell or array to UPPERCASE, lowercase, or Proper Case, respectively.
  - **Trim** The trim function is used to remove extra whitespaces from before, between, or after string entries.
  - **CONCAT** The Concat function (short for concatenate) 'glues' the (string) entries of two cells together. In class we used this to join the column of first names to the column of last names to get a full name colunm.
  - **Slicing:** The LEFT and RIGHT functions return the leftmost (rightmost) characters from a cell. For example, `LEFT(B2,5)` returns the five leftmost characters from B2.
  - **Wildcards** When searching or applying conditionals to string cells, the '*' character can be used as a wildcard that will match any characters, when placed to the left or right of a substring. For example, "Spi*" will match any string that starts with the letters Spi, while "*spicy*" will match any string that contains the world spicy anywhere in the cell.

- Other
  - **COUNTA** The COUNTA function takes as input an array and returns the number of non-empty cells in that row or column.
  - **UNIQUE** The unique function takes as input an array and returns an array of the values that occur but only a single time for each value. We combined this with the COUNTA function to count the number of unique beers and breweries in the beer reviews example.
  - **SEARCH** The search function looks for a substring in a single cell and returns the position in the string where that substring starts, if it exists. If the substring is not in the cell it returns an error. For example, if the cell D3 contains the string "MY FAVORITE NUMBER" the command `SEARCH(D3,"OR")` will return 7. We used this together with the ISNUMBER function to find spicy varieties of Ramen.