Econ 294 Assignment 3

Curtis Kephart

Winter 2015

Use R to answer the following questions.

- Due by Feb 5th 2016 (after the next lecture).
- Turn in your `.R` script by pushing it to your public github repo
  and emailing the URL to your instructor (at `curtisk+econ294_03@ucsc.edu`)
- It's important the script is able to run without error.
- Comment your code so it's clear which code blocks answer particular questions.
- Where a question asks for a specific answer, use `print()` to print the answer to the console

Just to be clear, and to help you get started, here is an example that satisfies the first set of questions, at `Assignments/CurtisKephartAssignment3Creator.R`.

This assignment closely follows the dplyr package vignette.

0. Identifying information. Print your name, student ID and email.

1. Load the following file as a data frame. For the rest of the assignment I will refer to the data frame loaded as `df.ex`.

   ```
   https://github.com/EconomiCurtis/econ294_2015/raw/master/data/org_example.dta
   ```

2. Filter

Use `dplyr` to subset `df.ex` to just the last month of 2013.

Print the number of observations that remain.

Print the number of observations in Summer 2013. (defining Summary as July, August, and Sept months)

3. Arrange

Create a new data frame called `df.ex.3a` that is sorted with year and month ascending (dates increase with higher row numbers)

4. Select

Create a new data frame called `df.ex.4a` with only columns `year` through `age`.

Create a new data frame called `df.ex.4b` with only columns year, month, and columns that start with `i`.

For the variable `state` print the distinct set of values in the original df.ex.

5. Mutate

Create two new functions.

One function called `stndz` that takes a vector of numbers, and returns the standard score for each element (ignoreing `NA`s). See Lec04 Notes for an example.

Another function called `nrmlz` that takes a vector of numbers, and returns the feature scaled value for each element (ignoring `NA`s in your `min()` and `max()` calls). Feature scaling details, this function will work similar to `stndz`.

Create a new data frame called `df.ex.5a` with two new columns, one called `rw.stndz` with the standardized score of real wages, and another called `rw_nrmlz` with feature scaled values of real wages.

Create a new data frame called `df.ex.5b` with three new columns. The three new columns should reflect the standard score (`rw.stndz`), feature scaled value (`rw_nrmlz`), and `count` at year, month groupings. (Tip1, use `group_by()` and `n()`. Tip2 to help you double check your work, In Jan 2013, the `rw` of `57.485714` should have `rw.stndz` of `2.18566760`, `rw_nrmlz` of `0.233088223` and `count` of `13342`.)

6. Summarize

Building off of `df.ex`, create a new data frame called `df.ex.6` that summarizes `rw` with min, 1st quartile, mean, median, 3rd quartile, max, and count at the `year`, `month` and `state` level. (Ignore any `NA`s. You'll want to use the `min`, `quantile`, `mean`, `median`, `max` and `n()` functions in your `summarise` call.) (Tip, `df.ex.6` should have 4284 observations.)

Use dplyr to find the year, month, state combination with the highest mean real wage.

Print which year, month, state observation has the highest mean real wage.

7. Challenge - extra credit.

Create a new data frame called `df.ex.7a` that is sorted with year and month ascending and `state` sorted in descending alphabetical order. (Double check the `state` column. Since it's a factor - `str(df.ex$state)` - it will now be sorted correctly. Consider your options to convert `state` into a char vector before your arrange call.)

Additional bonus questions may follow.