title: “Capstone Project Data Cleaning Steps”

When I first decided to investigate whether there was a correlation between podcast episode popularity and descriptions, I hoped to find a recent dataset on Kaggle or Data World that included that information. This proved difficult: the few datasets on podcasts I could find were either out of date or were simply lists of all podcasts listed on iTunes. Knowing that the data I needed didn't exsist, I set out to create the dataset myself from scracth.

This proved to be a fairly tedious task, but I knew that it would be a time saver when it came time for me to wrangle and clean my data. My initial thought was to use the top 100 podcasts overall on Apple podcasts (https://www.apple.com/itunes/podcasts/). On closer inspection, it became clear that the topics are so dispirate – covering history, politics, pop culture, sports, etc. – that it would be hard to prove trends in keywords and popularity. I decided to stick to one subcategory – Society & Culture – to focus on for my dataset. I also decided that since each podcast can have dozens, if not hundreds, of episodes, I would scale my dataset down to the Top 50 podcasts in Society & Culture knowing that this would still yield hundreds of episodes for me to comb through.

I started collecting data in a CSV document focusing on collecting the podcast name, episode title, date, running time and, most importantly, episode description and popularity rating. I wanted to focus on the bare minimum knowing that cutting out the clutter would yield cleaner data. This yielded 199 rows in my dataset and since Apple's API is not the most user friendly, I scraped a different podcast hosting site, Stitcher (https://www.stitcher.com/), to obtain the data and used Apple Podcasts for their popularity rating.

```
library(readr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
Podcast_Dataset <- read_csv("~/Documents/Podcast_Dataset.csv")
```

```
## Parsed with column specification:
## cols(
##   Podcast = col_character(),
##   `Episode Name` = col_character(),
##   `Running Time` = col_integer(),
##   `Release Date` = col_integer(),
##   `Episode Description` = col_character(),
##   `Popularity Rating` = col_integer()
## )
```

```
glimpse(Podcast_Dataset)
```

```
## Observations: 199
## Variables: 6
## $ Podcast              <chr> "Oprah's SuperSoul Conversations", "Opra...
## $ `Episode Name`       <chr> "ALANIS MORISSETTE: IS HAPPINESS TEMPORA...
## $ `Running Time`       <int> 33, 39, 35, 30, 28, 33, 30, 34, 31, 30, ...
## $ `Release Date`       <int> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8...
```

```
## $ `Episode Description` <chr> "Grammy award-winning singer/songwriter ...
## $ `Popularity Rating`   <int> 5, 5, 5, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3...
```

Once my data was collected, the next step was to create a corpus. Since I was planning to do some text mining later in my analysis, creating the corpus was a crucial step.

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.4.3
```

```
## Loading required package: NLP
```

```
podcast_corpus <- Corpus(VectorSource(Podcast_Dataset))
```

With my corpus was created, I set to work on cleaning my data. For this, I used some basic data cleaning functions. I used: + the tolower function to make my text uniform + removed stopwords, numbers and punctuation + used the stemDocument function

Once my data was cleaned, I could move forward with analysis.