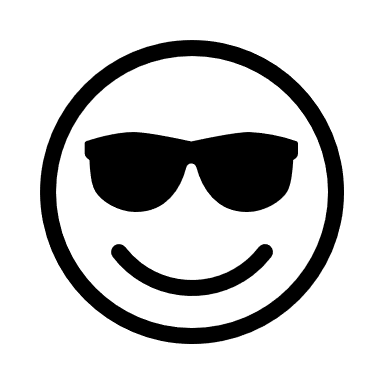
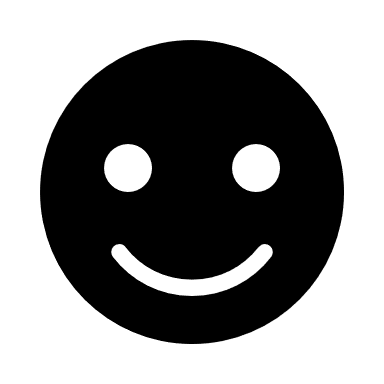
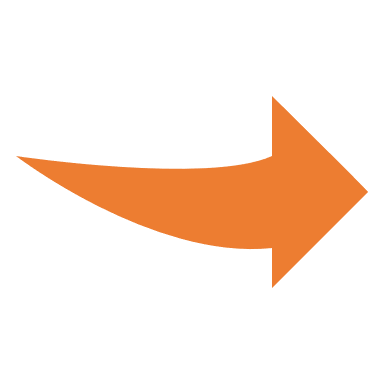
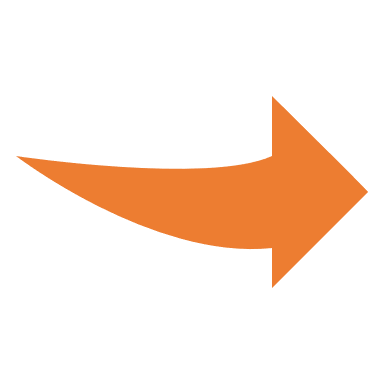
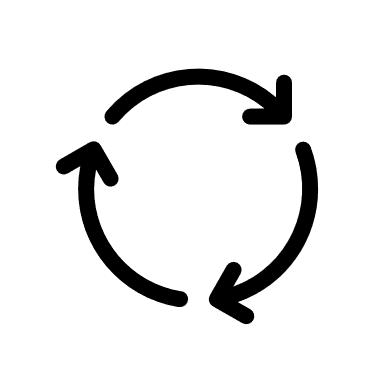
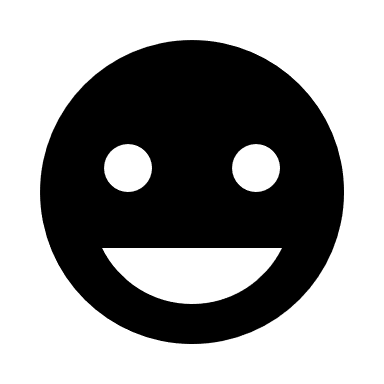
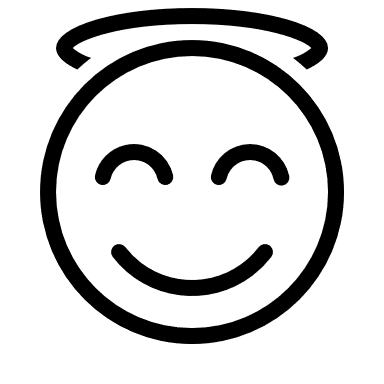
**Happiness Squad does ETL**

ETL Project Report, August 15, 2020

**INTRODUCTION**

* Team members:
* Eben Haezer
* Momotaz Mahin Khan
* Sheri Shojaie
* Vivi Santosa
* Project description / outline:  
  The goal of this project was to extract data related to our project 1 “The Pursuit of Happiness” from various disparate sources. Unlike the global scope of our first project, the focus of this ETL exercise will be on Canadian data. Once acquired, source information will undergo transformation to obtain data that is clean, disambiguous, and ready-to-consume. Transformed datasets will be loaded into a final production database.

**DATA SOURCES**

* + StatCan Life Satisfaction - API  
    <https://www.statcan.gc.ca/eng/developers/wds/user-guide>
  + Global News – web scrape  
    <https://globalnews.ca/tag/happiness/>
  + UWaterloo Canadian Index of Wellbeing – web scape  
    <https://uwaterloo.ca/canadian-index-wellbeing/reports/2016-canadian-index-wellbeing-national-report/trends-and-statistical-highlights>

**EXTRACT**

* StatCan Life Satisfaction – API
  1. The first step is to look for the desired data on the StatCan site. We pasted the URL 'https://www150.statcan.gc.ca/t1/wds/rest/getAllCubesListLite' to chrome and did a search function.
  2. The desired table was identified as "Average satisfaction with life and with selected domains of life by age group and sex". The Product ID for this table was used to request the information from the StatCan API.
  3. Using *get* and product ID, a url was returned that allowed for download of the data.
  4. A request for download was created to download all data as a zip file. Python was used to unzip and extract the data (csv format), and pandas library was used to read the data.
* Global News, web scrape
  1. Create path for chrome driver. Using Browser from the splinter library, visited the Global new web page including the tag for “happiness”. This will return news articles that are related to happiness.
  2. Scrape page into beautifulsoup and using *find\_all* and specific tag, get desired data from news articles: title, an excerpt, and the publication date.
  3. A for loop was used to retrieve the top 5 most relevant articles, stored in a dictionary.
* UWaterloo Canadian Wellbeing Index
  1. Create path for chrome driver. Using Browser from the splinter library, visited the University of Waterloo website for the Canadian Wellbeing Index study reports.
  2. The time library was used to delay execution
  3. From the main page (‘url1’), data from two tables were scraped and pandas was used to read the tables
  4. Data from eight additional “feature tables” were also scraped. Feature tables were stored in a similar format on the UWaterloo webpage, therefore to streamline scrape (and some of the data transformation steps), a function was created called “table\_maker\_function”.
  5. This function was used to scrape the table data from each of the feature table web pages and used pandas to read the tables. See the transformation steps taken in the below section.

**TRANSFORM**

* StatCan Life Satisfaction (API) transformation steps included:
  + Select only columns of interest
  + Drop na values
  + Rename columns to match desired schemata in database
  + Default index column used for index and primary key
* Global News (web scrape) transformation steps included:
  + Only desired fields were scraped during the extraction step. Otherwise data was clean and placed into a dataframe prior to loading into database
  + Default index column used for index and primary key
* UWaterloo Canadian Wellbeing Index (web scrape) transformation steps included:
  + String data was split to retrieve desired data from table for certain fields exracted
  + Data type was updated to integer or float
  + Index was set to year (common to all tables from this data source)
  + The two main tables were joined on ‘year’
  + Unwanted fields were dropped
  + Column names were updated. Especially in the eight feature tables, very long column names were updated to short and intuitive column names to match table schemata in database

**LOAD**

* Background: relational database PostgreSQL was selected as the production database to load and store all data  
  PostgreSQL was selected as our production database for the following reasons:
  + SQL prioritises data integrity and outlining the relationships between the different variables obtained. As this is a continuation of our previous project, we have familiarity with the different data sources and the transformation steps that will be taken in order to prepare the data for loading into defined table schemata
  + Data sources are relatively static and we do not expect the format of the data to change over time
  + SQL will allow us to map the data relationships between the different sources to obtain optimal table schemata for future data additional, refreshes, analysis and consumption
* Table Schemata creation:
  + Database was created in PostgreSQL called happiness001\_db
  + Using Query Tool, tables with appropriate schemata were created (refer to schemata.sql for schemata queries) for 11 data tables
    - Table titles, fields (with primary key) and field types were specified
  + Tables include:
    - ‘life\_domain’ (from StatCan API)
    - 'happiness\_news' (from Global News web scrape)
    - 'ciw\_summary' (joined tabled from UWaterloo CIW study web scrape)
    - 'leisure' (from UWaterloo CIW study web scrape)
    - 'living\_standards' (from UWaterloo CIW study web scrape)
    - 'time\_use' (from UWaterloo CIW study web scrape)
    - 'environment' (from UWaterloo CIW study web scrape)
    - 'education' (from UWaterloo CIW study web scrape)
    - 'healthy\_pop' (from UWaterloo CIW study web scrape)
    - 'comm\_vitality' (from UWaterloo CIW study web scrape)
    - 'democratic\_engagement' (from UWaterloo CIW study web scrape)
* Load steps:
  + Connection to database was created using *create\_engine* from sqlalchemy
  + Credentials were provided in connection string and engine was created
  + Confirmed successful connection to the database and correct tables
  + Fully transformed dataframes were appended to database tables.
  + Query was performed in PostgreSQL to confirm succesful data load