**Goal:**

* To explore various API through R.
  + Facebook
  + Twitter
  + Instagram
  + Google
  + Weather
* Specifically investigate what information is available on these APIs
* What tools are available in R to help get these information

**Basics:**

* All APIs are available via http requests.
* This means by formatting a specific URL with the appropriate authentication and calling the GET method, or simply posting the url into a web browser. A call to the API will be made and
  + the response is returned.
* An Http request :
* https://maps.googleapis.com/maps/api/geocode/json?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&key=***YOUR\_API\_KEY***
* Typically all the APIs support multiple response formats. Xml and Json are typically the most commonly supported and used.
* Typically the authorization key/token is created through OAuth protocol, which will be covered in the following slide.
* Typical response includes date, status, response content.
* In order to access the api, an application needs to created.

**OAuth:**

* This slide is just a brief introduction to what happens during the OAuth Protocol.
* OAuth is an internet security protocol where the user would grant a 3rd party (the application) to access data of a service provider (resource sever) on the user’s behalf without the user supplying his/her credentials to the 3rd party directly.
* The libraries in R masks this dance internally and only requires the user to authorize the access via a brower pop up.

1. The user accesses the application and requests the application to access the API on his/her behalf. The application would go to the authorization provider and ask for a request token to start the Authorization flow. The application provides its credentials, normally the consumer key.
2. With this request token, the application would redirect the user to the authorization sever, where the user signs in and grants access to the user application for access.
3. After this authorization, the application exchanges the request token for an authorization token.
4. This authorization token, combined with other parameters for the HTTP request is used to access the API.

**Facebook API:**

* Intro:
  + The structure of the facebook api is a social graph. The nodes are posts, users, comments etc. While the edges are what connects different nodes together, for example, a user’s photo. An edge would connect these two nodes. Fields are information associated with each node, for example, the owner of the photo, information about the user etc.
  + The results/response from fakebook is often paginated for readability. The limit of responses each page varies depending on type of end point requested. Facebook uses three types of pagination, cursor based, time based and offset based. Most responses use cursor paged pagination, where each cursor is a http request that points to more responses.
  + The rate limit specifies that it is the user aggregate, meaning the total requests are shared amongst all users. No specific definitions because not all calls are rate limited.
  + Use basic Oauth
  + Does not provide a commercial version, has 2 apis for advertisement. Ad api and atlas api.
* Advantages:
  + Public data is readily available via search methods. Does not seem to be limited by time. Public page information, public group information.
  + Easy to keep track of conversations via comments and replies to comments.
* Disadvantages:
  + Due to new update of facebook API. User information is no longer publicly available. Information like location, likes dislikes, user posted comments are no longer publicly available.
  + All references to specific comments, posts, ids require the user to know the id in order to extract information, rather than the ability to use names. This makes it difficult to be able to locate specific interests.
  + Many potentially interesting endpoints require proper information, which is annoying
  + Generally, a public search needs to be conducted to extract public comments etc, but the underlying identity of the user is not available.
  + Even though there is an api field describing location for posts, pages, photos. The information is often not provided.

**Facebook Rfacebook 1:**

* Advantages:
  + The package provides public searches for groups and pages requiring names of the groups and pages respectively. For groups search, it will return list of group names and their ids along with a privacy indicator (open or closed) as a data frame. For pages, names, id, general information and potential location information is returned.
  + Provides functions to extract comments, posts, replies to comments information. No direct method to search for comment, but comment is returned as part of posts.

**Facebook Rfacebook 2:**

* Since this was done as the first API. A lot of learning happened. For this package, most of the work was just done to explore the provided functionalities.
* Most of the methods that has not been deprecated has been tested and the result documented in the report.
* An interesting point is the ability to chain multiple functions to examine facebook data in details.

1. A facebook page can be searched via *searchPage* function.
2. Use the page id and call the *getPage*  function.

**MiniProject:**

* Mini 1:
  + A search was conducted on upto 2000 pages, only 494 returned that mentions flowers. From these pages, the location of the pages are extracted.
  + If location information not provided, `not provided` is marked.
  + A simple bar plot is made.
  + Illustrating the fact that much missing values.
* Mini 2:
  + The main purpose in this project was to manually extract **custom requests** and manually extract extra responses via paging.
  + Format the data into a data frame
  + A search for 500 posts on a travelling web page and the locations (if available) recorded.
  + Which country has been visited the most.

**Twitter REST API:**

* Intro:
  + Twitter API mainly stores information about 4 objects.
    - Tweets are the messages produced by users
    - Users are the twitter accounts with user names
    - Entities: meta information and contextual information like possible media, hashtags, urls that is embedded in the response for other objects.
    - Places: geographical information. Can be tagged within tweets and can be searched.
  + All objects have their own Id for identification.
  + The results are often pagingated using cursors. During a typical search, the user can also specify with Ids to search for. Like every other tweet up to id: 27568
  + Rate limit is based on 15 windows. Each api call to an endpoint have different rate limits. Unlike facebook, the rate limit applies to each user tokens, not as a total pool.
  + Does not have a commercial version of the API. Offer an ad api for managing ad campaigns.
  + The twitter Search API supports powerful queries, even though the public version only supports searches for upto 1 week. You can query for information using hashtags, positive/negative attitudes, filters for media and much more.
* Advantages:
  + Search API
  + The nature of tweeter is to spread news quickly, thus most of the information on twitter is available and mostly can be accessed publicly and freely. User information is also typically public because the idea is to get people to know you.
  + If the timeline of a particular user is searched (ie the tweets that specific user makes), the limit is around 3000 tweets and it can be historic. Meaning it can be tweets from older than a week.
* Disadvantages:
  + The public search api only returns recent tweets. It is recommended to keep an data base of new tweets so the old tweets can be referenced.
  + There is not direct replies to a tweet like comments on facebook. Twitter also has retweet option. The combination of these two makes it hard to keep track of how people react to posts.

**TwitteR:**

* Advantages:
  + The twitter package provides 2 main class wrappers, status and users. The status class is the equivalent of tweets and any time a tweet is returned, it is parsed into a status class. The same applies to users.
    - Status includes: text, screen name, id, reply information(who this message was meant for), retweet count, truncated and created.
    - User includes: name, screen name, user id, url, statues count, favorites count, friends count, number of followers, protected, profile image URL.
  + By providing getter functions and the ability to transform the data into a data frame and basing all returns based on these two classes. It is very institutive to use and easily formatted.
  + Allows public search, search for friendships data, searching for timelines of users, searching for users. Which covers most of the methods for extracting public data on twitter.
  + Allows mysql data base, also a function for generic data base handling (not tested).
  + Easy token function that caches internally.
* Disadvantages:
  + The cached token can not be easily accessed via other functions outside of the package, needs to create token again.
  + Does not support the extraction of entities from a tweet response.

**Stream API & StreamR:**

* Intro:
  + The Stream API is essentially a never ending http request. A connection between the data server and the client application is established. A query is pre-defined to select what information need to be returned. Any updates fitting the query is returned live.
  + The API is divided into 2 publicly accessible components. The public stream api, which returns live tweet updates fitting the specified query and the user stream api, which tracks a user and notifies any changes to messaging, replies to status and friendship changes. This API requires the proper permissions to read user messaging.
  + Once the connection is established, the connection is open unless the user closes the connection manually or twitter sever shuts down the connection.
    - Shut down will occur for (put in potential answers in questions)
* Stream R:
  + An R package providing functionalities to connect to both the public and user streams. The response is saved to a custom file.
  + Tweet stream allows filtering based on key words, language, location and allows the user to specify tweet number limit, how long to sustain the connection.
  + This package outputs to Rstudio console for information regarding connection status,
  + Masks away from having to deal with reconnecting in case of connection drops.
  + Documentation provides more details

**Google Maps API:**

* Google maps web service is a service provider for analysing geographical data. Other APIs such as the java script ones allows the user to include google maps as part of the application.
* The google maps API for web service is based on setting HTTP requests to retrieve responses regarding geocoding, directions, etc.
* The maps api offers 5 main methods:
  + Directions: offers directions between start and end points
    - Includes estimated route time
    - Modes of travel
    - Duration of each step
    - Multiple routes
  + Distance Matrix: distance
    - Allows multiple start/end pairs
  + Elevation:
    - Elevation information for upto 512 locations at a time.
  + Geocoding: convert address to lat/lng and vice versa
  + Roads: snaps the points to nearest road or describe nearest road given points
    - Speed limit information only available to premium.
  + Time zone:
    - Time zone information for points
  + Places:
    - Places search: allows for search of locations nearby to the provided location, the search can be modified via ranking by distance, price, opening now etc.
    - Place details: using a place\_id, more detailed information regarding the place such as rating, address, phone number can be returned.
    - Place photos: allows the user to obtain google server photos for a location.
    - Place autocomplete: allows for autocomplete of a incomplete place text.
  + Extra rates is typically 0.5 us dollars per extra 1000 requests. Permium account is yearly based.
  + Google does an amazing job with the documentation, not too much time was spent on exploring what each api can do because it is very well described. Instead effort was put into visualizing the data returned from google.

**Leaflet:**

* Leaflet is an open source library which develops interactive java script maps.
* There is an R package which wraps around the java script library and makes it useable through an R interface.
* There are many other libraries available for visualizing geographical data on maps. Leaflet was chosen because it is interactive. I
* Leaflet requires internet connection to operate properly.
* Two main first exploration tasks were designed
  + Replicating Google Maps:
    - The first is to plot the route from NRC to the apartment that I am currently living in given their addresses using the directions api
    - Then using places search, a search was conducted around these two points and the points of interests were plotted.
  + Simple Density Visualization:
    - Took a crime data set in the United states. The data set contained city name, population, crime data.
    - Calculated total number of crimes and then counted it as a percentage of city population.
    - Visualized using leaflet, percentage proportional to marker size.
  + Bigger Scaled:
    - Mr.Stephane was interested to see capabilities of leaflet + google maps
    - 100 origin/destination pairs were selected, origins from cities in alberta and destination from cities outside of alberta but in Canada
    - Using the directions api with travelling mode driving, the ***OVERVIEW*** route was plotted on leaflet.

**Flickr API:**

* Flickr is a photo sharing/ hosting service where people can post their photos and maintain their rights to the photographs.
* Like all the other APIs, an standard HTTP request is sent to the service provider and an response is received.
* Interesting Things about the Flickr API:
  + It is not supported or owned by the official flickr. It is developed by the community.
  + Many of the method end points (Search for example) does not require the OAuth process, instead only the application identifications are required as a parameter of the http request.
* The main method explored is the public photo search. The search allows you to specify taken dates, up load dates, loation information, text in msg, tags as search paraemeters. Will return the information regarding the photos.
* The return responses are often limited because not all photos are public and also needs to be verified if the return is exactly what is desired.
* The meta data information for photos are available.
* Mini Project:
  + Canada Day 2015 near Parliament
  + Goal: search for photos/tweets for Canada day 2015.
  + Not possible for tweeter due to limitation
  + For flickr,
    - Time: july to august 2015
    - Both text and tags were used: Happy Canada day, Canada day 2015
    - ***Woeid: id of the area, returned more photos on the streets near parliament.***

Weather Web Scraping:

* Data Collection:
  + The goal is to collect the current weather condition and the forecasted weather from the environment Canada website.
    - These two pictures hows the interface.

The main packages used are rvest and xml2 for parsing the source code.

* + A css tag was used to extract the information needed by manually finding it from the source code of the website.
    - <http://www.computerhope.com/issues/ch000746.htm>
  + The website is updated every hour. So in order to collect the data, a windows R script was run to append information to the excel file.
    - The package used is taskscheduleR, which also provides an R studio plugin to run the script.
    - This package interfaces with the taskScheduler of windows through schtasks.ext run in command line.
  + The problem was that if the power source is unplugged, the script would not run. In order to address it, the xml file containing the parameters which specify how the task is run needs to be modified. This is again done with the XML package, then the same schtask tool is called to run the modified version of the file.
  + Now to get the predications to align with current time, columns are lagged. (explain process)Then the other columns are calculated.
* Data Processing: explain what each graph is:
  + Mention error bars,
  + Signed errors