R Basics and Rfacebook

Rick Jason G. Obrero
Institute of Computer Science, UPLB

In this talk

R as a programming language

RStudio as R's interface

Rfacebook and the Facebook Graph

Objectives

Workshop Proper

In this talk

R as a programming language

RStudio as R's interface

Rfacebook and the Facebook Graph

Objectives

Workshop Proper

R as a programming language

R was created primarily as a statistical programming language.

This includes a variety of statistical and graphical techniques.

R as a programming language

R is an interpreted language, and thus it is executed only on a command line.

If you want to make sense of its outputs, you need an Integrated Development Environment (IDE).

In this talk

R as a programming language

RStudio as R's interface

Rfacebook and the Facebook Graph

Objectives

Workshop Proper

RStudio as R's interface

It is a free and open source IDE.

It includes areas for storing the values of the variables used, integrated help and plot viewer, among others.

In this talk

R as a programming language

RStudio as R's interface

Rfacebook and the Facebook Graph

Objectives

Workshop Proper

Rfacebook and the Facebook Graph

Rfacebook is an R package.

A package is a set of R codes made by other developers to add functionality to the programming language.

A package may have multiple dependencies (or prerequisites) for it to work properly.

Rfacebook and the Facebook Graph

Rfacebook is an R package.

When it is installed, you can gather posts from public Facebook pages, or your own Facebook account.

In this talk

R as a programming language

RStudio as R's interface

Rfacebook and the Facebook Graph

Objectives

Workshop Proper

Objectives

In this workshop, we shall:

- gather posts from public Facebook pages
- check the top N most/least commented/liked/shared posts from those gathered
- search posts for certain keywords
- search individual Facebook posts

In this talk

R as a programming language

RStudio as R's interface

Rfacebook and the Facebook Graph

Objectives

Workshop Proper

Workshop Proper

In this workshop, we shall:

- gather posts from public Facebook pages
- check the top N most/least commented/liked/shared posts from those gathered
- search posts for certain keywords
- search individual Facebook posts

Workshop Proper

In this workshop, we shall:

- gather posts from public Facebook pages
- check the top N most/least commented/liked/shared posts from those gathered
- search posts for certain keywords
- search individual Facebook posts

Gather posts from public Facebook pages

To use Rfacebook, we shall install it first.

1) Open RStudio.
2) On the lower left, you will see R's prompt. On it, we type:

```
install.packages("Rfacebook");
```

3) We can then now use the installed package. On the same prompt, we type:

```
library(Rfacebook);
```

Gather posts from public Facebook pages

After installation, we can now get a token to access the Facebook Graph.

1) Open your web browser and go to `https://developers.facebook.com/tools/explorer/`.
2) On top, we can see a text box with a long text, which represents the token.

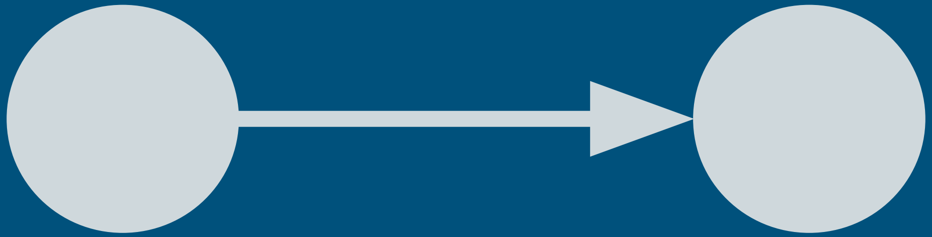
Think of the token as a security feature for those who access the graph.

But what is a graph?

A graph is a set of nodes (objects), together with links (edges) that associate such objects altogether.

Nodes in Facebook can represent people, pages, events, among others.

Edges can represent the interactions that such nodes have in other nodes.



Gather posts from public Facebook pages

After installation, we can now get a token to access the Facebook Graph.

1) Open your web browser and go to `https://developers.facebook.com/tools/explorer/`.
2) On top, we can see a text box with a long text, which represents the token.
3) Leave the tab there, and we shall use it later.

Gather posts from public Facebook pages

We can now gather posts from a public Facebook page.

1) On the tab of your browser where the token is, copy the token from the text box.
2) Switch to the RStudio interface. On its prompt, type:

   ```
   token <- "XXXXXXXXXX";
   ```

   The XXXXXXXXXX stands for the token you copied.
3) Then, press Enter.

You have just made a variable!

Gather posts from public Facebook pages

We can now gather posts from a public Facebook page.

1) Still, on the RStudio interface, on its prompt, type:

```
EndocrineWitch <- getPage("EndocrineWitch", token, n = 5000)
```

This will:

- make a variable named EndocrineWitch
- which will get the 5000 recent status updates
- from the Facebook page "Endocrine Witch".

Workshop Proper

In this workshop, we shall:

- gather posts from public Facebook pages
- check the top N most/least commented/liked/shared posts from those gathered
- search posts for certain keywords
- search individual Facebook posts

What does a FB page have?

We have now information on the FB page called Endocrine Witch.

1) On RStudio's prompt, type `EndocrineWitch$`
   You can see different fields that are available for us to process or use.

`from_name`:	the name of the FB page
`message`:	the post's message (status)
`created_time`:	the time when the post was created
`id`:	the identifier for the post; can be accessed by fb.com/id

What does a FB page have?

We have now information on the FB page called Endocrine Witch.

1) On RStudio's prompt, type `EndocrineWitch$`
   You can see different fields that are available for us to process or use.

`likes_count`:	number of likes
`comments_count`:	number of comments
`shares_count`:	number of shares

Check the top N liked posts

We have now information on the FB page called Endocrine Witch.

1) On RStudio's prompt, type `EndocrineWitch$`
   You can see different fields that are available for us to process or use.
2) We shall create a variable for the ordering of the posts based on the number of likes:

```
ord <- order(EndocrineWitch$likes_count, decreasing = TRUE);
```

We created:

- a variable named `ord`
- which arranged the likes count of the posts in non-decreasing order.

Check the top N liked posts

We have now information on the FB page called Endocrine Witch.

1) On RStudio's prompt, type `EndocrineWitch$`
   You can see different fields that are available for us to process or use.
2) We shall create a variable for the ordering of the posts based on the number of likes:
   ```
   ord <- order(EndocrineWitch$likes_count, decreasing = TRUE);
   ```
3) We shall filter the top N liked posts by applying the following filter:
   ```
   top_ten <- EndocrineWitch[ord[1:10], ]
   ```

Check the top N liked posts

We have now information on the FB page called Endocrine Witch.

3) We shall filter the top N liked posts by applying the following filter:

```
top_ten <- EndocrineWitch[ord[1:10], ]
```

We:

- collected the top ten elements from `ord` (which contains the ranking of the most liked posts)
- and used it to collect the top 10 most liked posts from EndocrineWitch.

Check the top N liked posts

We have now information on the FB page called Endocrine Witch.

3) We shall filter the top N liked posts by applying the following filter:

```
top_ten <- EndocrineWitch[ord[1:10], ]
```

We:

- collected the top ten elements from `ord` (which contains the ranking of the most liked posts)
- and used it to collect the top 10 most liked posts from EndocrineWitch.

How to do it for comments?

Check the top N liked posts

We have now information on the FB page called Endocrine Witch.

3) We shall filter the top N liked posts by applying the following filter:

```
top_ten <- EndocrineWitch[ord[1:10], ]
```

We:

- collected the top ten elements from `ord` (which contains the ranking of the most liked posts)
- and used it to collect the top 10 most liked posts from EndocrineWitch.

How to do it for shares?

Check the last N liked posts

The process below outlines for getting the top posts.

1) On RStudio's prompt, type `EndocrineWitch$`
   You can see different fields that are available for us to process or use.
2) We shall create a variable for the ordering of the posts based on the number of likes:
   ```
   ord <- order(EndocrineWitch$likes_count, decreasing = FALSE);
   ```
3) We shall filter the top N liked posts by applying the following filter:
   ```
   top_ten <- EndocrineWitch[ord[1:10], ]
   ```

How about for the bottom posts?

Workshop Proper

In this workshop, we shall:

- gather posts from public Facebook pages
- check the top N most/least commented/liked/shared posts from those gathered
- search posts for certain keywords
- search individual Facebook posts

Search posts for certain keywords

We shall use the `message` field in gathering posts. In a message, we can have an exact keyword search.

1) On RStudio's prompt, type the following command:

```
dbts <- EndocrineWitch[grep("diabetes", EndocrineWitch$message), ]
```

In this command, we:

- used the `grep` command to get messages with the word diabetes
- and selected the rows which contain the said word
- and stored them in a variable named `dbts`.

Search individual Facebook posts

We shall use the `getReactions` command to get reactions (like, love, wow, sad, angry, haha) for a specific Facebook post.

1. On RStudio's prompt, type the following command:

   ```
   rxn <- getReactions(EndocrineWitch[1,]$id, token)
   ```

In this manner, we:

- gathered the id of the latest post in EndocrineWitch
- and used the Facebook Graph API to get the reactions of the said Facebook post.

Search individual Facebook posts

We shall use the `getPost` command to get contents for a specific Facebook post.

1. On RStudio's prompt, type the following command:

   ```
   first <- getPost(EndocrineWitch[1,]$id, token, n = 10000)
   ```

In this manner, we:

- gathered the id of the latest post in EndocrineWitch
- and used the Facebook Graph API to get the likes and comments of the said Facebook post.

What does a FB post have?

We shall use the `getPost` command to get contents for a specific Facebook post.

1. On RStudio's prompt, type the following command:

```
first <- getPost(EndocrineWitch[1,]$id, token, n = 10000)
```

There are three parts of a Facebook post:

a) `page:` The content of the Facebook post
b) `likes:` The likes / reaction information of the post
c) `comments:` The comments information of the post

What does a FB post have?

To get information about the post:

1. On RStudio's prompt, type the following command:
   `first$page`

There are three parts of a Facebook post:

a) `page:` The content of the Facebook post
b) `likes:` The likes / reaction information of the post
c) `comments:` The comments information of the post

What does a FB post have?

To get information about the likers of the post:

1. On RStudio's prompt, type the following command:

```
first$likes
```

There are three parts of a Facebook post:

a)	`page`:	The content of the Facebook post
b)	`likes`:	The likes / reaction information of the post
c)	`comments`:	The comments information of the post

What does a FB post have?

To get information about the commenters and the comments of the post:

1. On RStudio's prompt, type the following command:

   ```
   first$comments
   ```

There are three parts of a Facebook post:

a) `page:` The content of the Facebook post
b) `likes:` The likes / reaction information of the post
c) `comments:` The comments information of the post

Getting comment replies

To select the ID of a particular comment:

1. On RStudio's prompt, type the following command:

   ```
   id <- first$comments[1, ]$id
   ```

This ID shall be used to get the comment's replies.

Getting comment replies

To select the ID of a particular comment:

1. On RStudio's prompt, type the following command:
   ```
   id <- first$comments[1, ]$id
   ```
2. Continuing with the prompt, we shall use the `getCommentReplies` command.
   ```
   getCommentReplies(id, token)
   ```

Using this command, we gathered the replies on the specific comment.

What does a FB comment have?

1. On RStudio's prompt, type the following command:

```
id <- first$comments[1, ]$id
```

2. Continuing with the prompt, we shall use the `getCommentReplies` command.

```
getCommentReplies(id, token)
```

FB comments have two subfields:

a) `comment`: information about the comment

b) `replies`: information about the replies to the comment