

Sourcing Data into R  
CS9223/CS6513  
Big Data Management and Analysis

# Sources

- Programmatic
  - TextConnection
- External
  - Loading CSV
  - Connecting to sources
    - Databases mysql, mongo
    - Twitter and other RSS

# textConnection

- Most suitable when you have data within the R script in constant quoted character format
- or from a string variable

```
> t<-read.table(textConnection('
+ id Source
+ 1 A10
+ 2 A32
+ 3 A10
+ 4 A25'),header=T)
> t
  id Source
1 1 A10
2 2 A32
3 3 A10
4 4 A25
> |
```

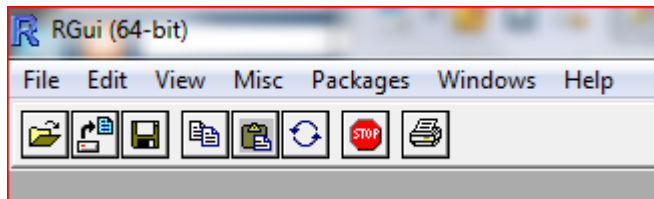
```
> xstr<-'
+ id Source
+ 1 A10
+ 2 A32
+ 3 A10
+ 4 A25'
> xstr
[1] "\nid Source\n1 A10\n2 A32\n3 A10\n4 A25"
> t2<-read.table(textConnection(xstr),header=T)
> t2
  id Source
1 1 A10
2 2 A32
3 3 A10
4 4 A25
> |
```

# Loading data from CSV

```
> sp2014dimrank<-read.csv("E:/poly/2014spring/grading/peer-eval-trend-by-hw-dim-coded.txt",header=FALSE,sep=",")
> str(sp20144dimrank)
Error in str(sp20144dimrank) : object 'sp20144dimrank' not found
> str(sp2014dimrank)
'data.frame':   65 obs. of  3 variables:
 $ V1: Factor w/ 14 levels "A","B","C","D",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ V2: Factor w/ 5 levels "HW01","HW02",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ V3: num  3.75 4.33 5 3.5 4.33 ...
> head(sp2014dimrank)
  V1  V2      V3
1  A HW01 3.750000
2  B HW01 4.333333
3  C HW01 5.000000
4  D HW01 3.500000
5  E HW01 4.333350
6  F HW01 3.666650
> tail(sp2014dimrank)
  V1  V2      V3
60  J HW05 5.000000
61  G HW05 3.166667
62  A HW05 3.666700
63  K HW05 3.250000
64  H HW05 4.416650
65  M HW05 4.333367
```

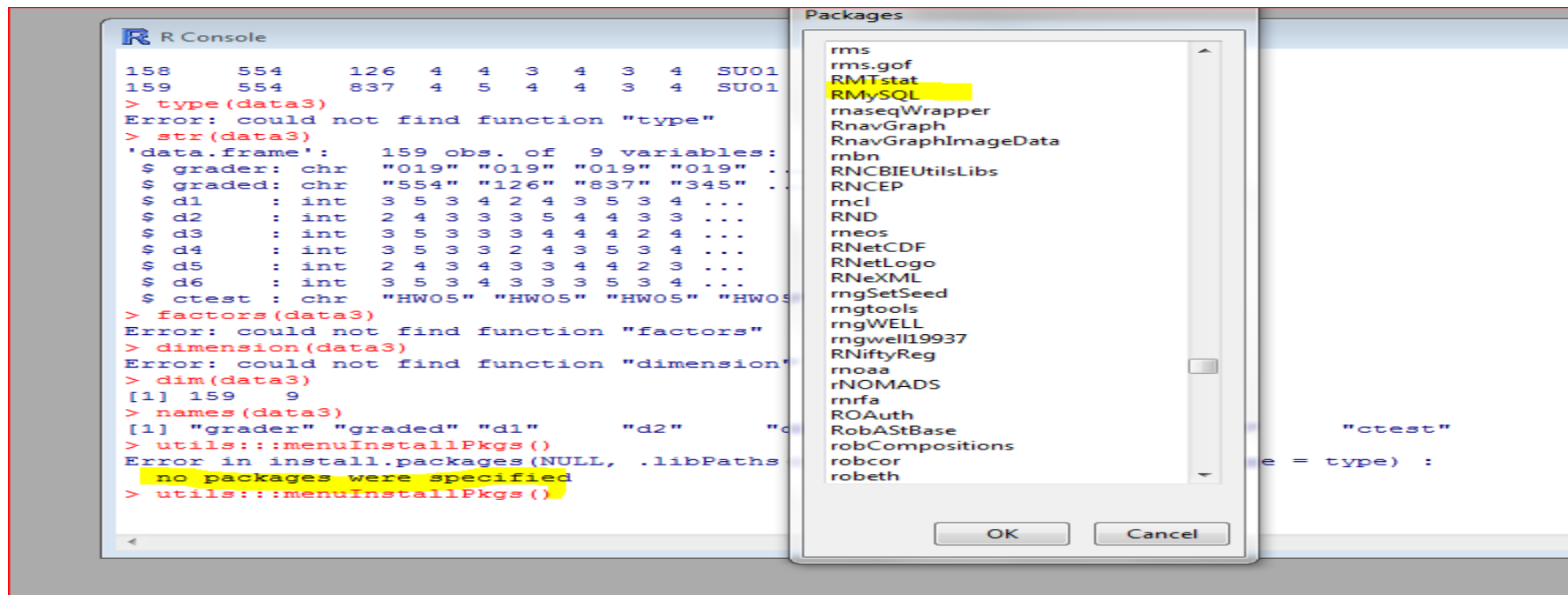
# External Source:MySQL

- Many options – RMySQL
- R Version 3.1.2
- Package RMySQL version
- <http://cran.r-project.org/web/packages/RMySQL/index.html>
- <http://cran.r-project.org/web/packages/RMySQL/RMySQL.pdf>
- Download -> [http://cran.r-project.org/bin/windows/contrib/3.1/RMySQL\\_0.10.1.zip](http://cran.r-project.org/bin/windows/contrib/3.1/RMySQL_0.10.1.zip)
- And [http://cran.at.r-project.org/bin/windows/contrib/3.2/DBI\\_0.3.1.zip](http://cran.at.r-project.org/bin/windows/contrib/3.2/DBI_0.3.1.zip)
- Install the two downloaded packages using `utils::menuInstallLocal()` on the command line or
- From the



Packages menu.

# RMySQL



You can also install packages as shown above using menu

```
> require(RMySQL)
Loading required package: RMySQL
Loading required package: DBI
```

Then issue the require command.  
Verify that both packages are loaded.

# Step 1: First start a mysql session

```
Administrator: C:\windows\system32\cmd.exe - mysql -urk -p
+-----+-----+
| id | description |
+-----+-----+
| 1 | (1) Individual Contribution |
| 2 | (2) I will certainly work u |
| 3 | (3) I enjoyed working with |
| 4 | (4) student works for team |
| 5 | (5) allows and encourages a |
| 6 | (6) delivers and keeps what |
+-----+-----+
6 rows in set (0.27 sec)

mysql> show databases ;
+-----+
| Database |
+-----+
| information_schema |
| edm |
| edm2012 |
| eds2012 |
| fall2013 |
| lcc |
| mysql |
| pfa |
| rms |
| s2014 |
+-----+
10 rows in set (0.00 sec)

mysql>
```

Confirm u have a working MySql  
Credentials  
Hostname  
Username  
Password and  
A database.

# Test RMySQL

- Test simple queries

```
> con<-dbConnect(RMySQL::MySQL(),host="localhost",user="rk",password="rk10262014",dbname="s2014")
> res <- dbSendQuery(con,"select count(*) from rk10262014")
> data <-dbFetch(res)
> data
  count(*)
1        4
> res2 <- dbSendQuery(con,"select * from rk10262014")
> data2<-dbFetch(res2)
> data2
```

```
> data2
  product rank
1    eqty    2
2     opt    1
3     cds    1
4    swap    1
> res3 <- dbSendQuery(con,"select * from blind_sdenorm")
> data3<-dbFetch(res3)
> data3
  grader graded d1 d2 d3 d4 d5 d6 ctest
1     019   554  3  2  3  3  2  3 HW05
2     019   126  5  4  5  5  4  5 HW05
3     019   837  3  3  3  3  3  3 HW05
4     019   345  4  3  3  3  4  4 HW05
5     019   554  2  3  3  2  3  3 HW04
6     019   126  4  5  4  4  3  3 HW04
7     019   837  3  4  4  3  4  3 HW04
8     019   345  5  4  4  5  4  5 HW04
9     019   554  3  3  2  3  2  3 HW03
10    019   126  4  3  4  4  3  4 HW03
```



# Retrieve data from MySQL

```
> str(data3)
'data.frame': 159 obs. of 9 variables:
 $ grader: chr "019" "019" "019" "019" ...
 $ graded: chr "554" "126" "837" "345" ...
 $ d1 : int 3 5 3 4 2 4 3 5 3 4 ...
 $ d2 : int 2 4 3 3 3 5 4 4 3 3 ...
 $ d3 : int 3 5 3 3 3 4 4 4 2 4 ...
 $ d4 : int 3 5 3 3 2 4 3 5 3 4 ...
 $ d5 : int 2 4 3 4 3 3 4 4 2 3 ...
 $ d6 : int 3 5 3 4 3 3 3 5 3 4 ...
 $ ctest : chr "HW05" "HW05" "HW05" "HW05" ...
```

Summary: a data.frame is returned  
The column names are names  
The record count matches the record  
Count of the resultset of the SQL query.

```
> dim(data3)
[1] 159 9
> names(data3)
[1] "grader" "graded" "d1" "d2" "d3" "d4" "d5" "d6" "ctest"
> utils::menuInstallPkgs()
```

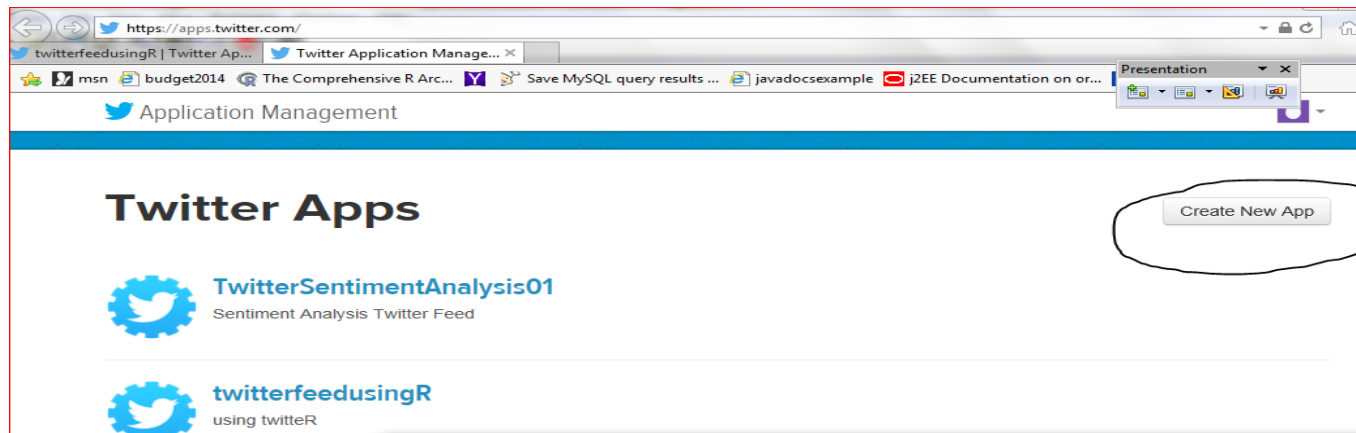
```
mysql> desc blind_sdenorm ;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| grader | varchar(3)    | YES  |     | NULL    |       |
| graded | varchar(3)    | YES  |     | NULL    |       |
| d1     | int(11)       | YES  |     | NULL    |       |
| d2     | int(11)       | YES  |     | NULL    |       |
| d3     | int(11)       | YES  |     | NULL    |       |
| d4     | int(11)       | YES  |     | NULL    |       |
| d5     | int(11)       | YES  |     | NULL    |       |
| d6     | int(11)       | YES  |     | NULL    |       |
| ctest  | varchar(6)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.04 sec)
```

# Api driven Integration

- This technique would work for any service
- Get the necessary keys from the service
- Install the necessary package to interact with the service
- Write small/simple application to verify everything works
- Graduate and move on to bigger and smarter processing

# twitter

- <https://apps.twitter.com/>



Locate and click on Create New App

Terminology may vary

This is specific to twitter.

# Fill in the App Form

<https://apps.twitter.com/app/new>

## Application Details

**Name \***

*Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.*

**Description \***

*Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.*

**Website \***

*Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.*

*(If you don't have a URL yet, just put a placeholder here but remember to change it later.)*

**Callback URL**

*Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.*

# Accept and Create

## Developer Agreement

Last Update: October 22, 2014.

This Twitter Developer Agreement ("**Agreement**") is made between you (either an individual or an entity, referred to herein as "**you**") and Twitter, Inc., on behalf of itself and its worldwide affiliates (collectively, "**Twitter**") and governs your access to and use of the Licensed Material (as defined below).

PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING WITHOUT LIMITATION ANY LINKED TERMS AND CONDITIONS APPEARING OR REFERENCED BELOW, WHICH ARE HEREBY MADE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU HAVE READ, AND THAT YOU AGREE TO COMPLY WITH AND TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALL APPLICABLE LAWS AND REGULATIONS IN THEIR ENTIRETY WITHOUT LIMITATION OR QUALIFICATION. IF YOU DO NOT AGREE TO BE BOUND BY THIS AGREEMENT, THEN YOU MAY NOT ACCESS OR OTHERWISE USE THE LICENSED MATERIAL. THIS AGREEMENT IS EFFECTIVE AS OF THE FIRST DATE THAT YOU USE THE LICENSED MATERIAL ("**EFFECTIVE DATE**").

IF YOU ARE AN INDIVIDUAL REPRESENTING AN ENTITY, YOU ACKNOWLEDGE THAT YOU HAVE THE APPROPRIATE AUTHORITY TO ACCEPT THIS AGREEMENT ON BEHALF OF SUCH ENTITY. YOU MAY NOT USE THE LICENSED MATERIAL

☒ Yes, I agree

Create your Twitter application

Create your Twitter Application...click on it...

# Get your own APP tokens

Consumer Key (API Key) LS2[REDACTED]cviDerqPT3iN4q

Consumer Secret (API Secret) wPBD29324ndmFbJVAjl2dohUIEk21M6bEhjQ1PxfvzxpP8FQiY

## Your Access Token

*This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.*

Access Token 14[REDACTED]EodfVJgZR8J0cyDYicNLWka

Access Token Secret S8dMuS[REDACTED]KA0NH04C

It is all free to setup and easy. Once you have these 4 tokens...

# twitter

- Install and load required package

```
> require(twitterR)
Loading required package: twitterR
```

Setup the keys and setup twitter\_oauth as shown here using your keys

```
> api_key="XXXXXXXXXXXXXXXXXXXX"
> api_secret="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
> access_token_secret="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
> access_token="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
> access_token_secret="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
> setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
[1] "Using direct authentication"
```

Use

```
> setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
[1] "Using direct authentication"
> user <- getUser("rk2153")
> user$getFollowersCount()
[1] 17
>
> user_followers <- user$getFollowers()
> followers_n <- length(user_followers)
```

# Who follows you

```
> user_followers <- user$getFollowers()
> followers_n <- length(user_followers)
> followers_scrennames <- vector()
> for (i in 1:followers_n) {
+   followers_scrennames[i] <- user_followers[[i]]$screenName
+ }
>
```

```
>
> followers_n
[1] 17
> user_followers
$`344417583`
[1] "amc866"

$`267838599`
[1] "jbhomerassoc"

$`388642431`
[1] "sneekieneekie"

$`407185849`
[1] "MichaelNichols2005"
```

What is being tweeted about your favorite topic?

```
> davos <- searchTwitter("davos",300)
> length(davos)
[1] 300
> davos[[1]]
[1] "SoylentHHH: RT @carlzimmer: The Earth's richest 80 people doubled their wealth between 2009 and 2014 & now own as much
> davos[[300]]
[1] "EvaDanickova: RT @LangBanks: RT if this makes you mad\n\nThe richest 1% will own more than 50% of the world's wealth by 201
> |
```



# Saving the tweets

- Convert to a data.frame
  - `davos.df<-twListToDF(davos)`
- Examine `davos.df` → `str(davos.df)`
- `head(davos.df)`
- Export to a csv
  - `write.csv(davos.df,"e:/poly/2015/spring/CS-DSC/davos.csv",row.names=F)`

More to follow on text analysis ....