Search earth data science site

Home/Courses /Earth analytics / Week 13. Get data using apis /**Twitter Data in R Using Rtweet: Analyze and Download Twitter Data**

Earth-analytics Home

UNITS

OVERVIEW

Course Syllabus
Midterm Project
Final Project
Data Source List
Search Course Lessons

earth-analytics Home

Lesson 2. Twitter Data in R Using Rtweet: Analyze and Download Twitter Data

Leah Wasser, Carson Farmer

Learning Objectives

After completing this tutorial, you will be able to:

- Query the twitter RESTful API to access and import into `R` tweets that contain various text strings.
- Generate a list of users who are tweeting about a particular topic.

What You Need

You will need a computer with internet access to complete this lesson.

In this lesson you will explore analyzing social media data accessed from twitter, in R. You will use the Twitter RESTful API to access data about both twitter users and what they are tweeting about

Getting Started

To get started you'll need to do the following things:

1. Set up a twitter account if you don't have one already.
2. Using your account, setup an application that you will use to access twitter from R
3. Download and install the `rtweet` and `tidytext` packages for R.

Once you've done these things, you are ready to begin querying Twitter's API to see what you can learn about tweets!

Set up Twitter App

Let's start by setting up an application in twitter that you can use to access tweets. To setup your app, follow the documentation from `rtweet` here:

TUTORIAL: How to setup a twitter application using your twitter account

NOTE: you will need to provide your cell phone number to twitter to verify your use of the API.

DELLWOOD AVE
CEDAR AVE
BALSAM AVE
EDGEWOOD DR
VALMONT RD
VALM
VALMONT RD
BLUFF ST
STERLING DR
PEARL ST
PEARL PKWY
ALPINE DR
NORTH ST
MAPLETON AVE
PINE ST
DEWEY AVE
PORTLAND PL
HIGH ST
MAXWELL AVE
MAPLETON AVE
SPRUCE ST
HIGHLAND AVE
WALNUT ST
WAL WALNUT ST
SPRUCE ST
PEARL ST
GROVE ST
ARAPAHOE AVE
ARAPAHOE AVE
MARINE ST
CANON PK
MARINE ST
GRANDVIEW AVE
UNIVERSITY AVE
PLEASANT ST
TAFT DR
COLLEGE AVE
COLORADO AVE
MADISON AVE
CENTENNIAL TRAIL
PENNSYLVANIA AVE
E EUCLID AVE
EUCLID AVE
AURORA AVE
AURORA AVE
BOY SCOUT
CASCADE AVE
EVANS DR
BASELINE RD
BASELINE RD
BASELINE RD
BASELINE
COLUMBINE AVE
MARIPOSA AVE
BLUEBELL AVE
KING AVE
PINON DR
CHIPPEWA DR
TENINO AVE
ASH AVE
DENVER BOULDER TURNPIKE
FOOTHILLS PKWY
28TH ST
30TH ST
55TH ST

A heat map of the distribution of tweets across the Denver / Boulder region source: socialmatt.com

Twitter in R

Once you have your twitter app setup, you are ready to dive into accessing tweets in `R`.

You will use the `rtweet` package to do this.

```
# load twitter library - the rtweet library is recommended now over twitteR
library(rtweet)
# plotting and pipes - tidyverse!
library(ggplot2)
library(dplyr)
# text mining library
library(tidytext)
```

The first thing that you need to setup in your code is your authentication. When you set up your app, it provides you with 3 unique identification elements:

1. appnam
2. key
3. secret

These keys are located in your twitter app settings in the `Keys and Access Tokens` tab. You will need to copy those into your code as i did below replacing the filler text that I used in this lesson for the text that twitter gives you in your app.

Next, you need to pass a suite of keys to the API.

```r
# whatever name you assigned to your created app
appname <- "your-app-name"

## api key (example below is not a real key)
key <- "yourLongApiKeyHere"

## api secret (example below is not a real key)
secret <- "yourSecretKeyHere"
```

Finally, you can create a token that authenticates access to tweets! Note that the authentication process below will open a window in your browser.

```r
# create token named "twitter_token"
twitter_token <- create_token(
  app = appname,
  consumer_key = key,
  consumer_secret = secret)
```

If authentication is successful works, it should render the following message in a browser window:

`Authentication complete. Please close this page and return to R.`

Send a Tweet

Note that your tweet needs to be 140 characters or less.

```r
# post a tweet from R
post_tweet("Look, i'm tweeting from R in my #rstats #earthanalytics class! @EarthLabCU")
## your tweet has been posted!
```

Search Twitter for Tweets

Now you are ready to search twitter for recent tweets! Let's start by finding all tweets that use the `#rstats` hashtag. Notice below you use the `rtweet::search_tweets()` function to search. `search_tweets()` requires the following arguments:

1. **q:** the query word that you want to look for
2. **n:** the number of tweets that you want returned. You can request up to a maximum of 18,000 tweets.

To see what other arguments you can use with this function, use the `R` help:

`?search_tweets`

```
## search for 500 tweets using the #rstats hashtag
rstats_tweets <- search_tweets(q = "#rstats",
                               n = 500)
# view the first 3 rows of the dataframe
head(rstats_tweets, n = 3)
##       screen_name   user_id          created_at          status_id
## 1 LuckyStrike1984  93801333 2018-01-10 22:57:50 951226606747561984
## 2   renato_umeton  31367101 2018-01-10 22:56:42 951226322117955584
## 3   itknowingness 213339721 2018-01-10 22:55:03 951225904046526464
##
## 1 RT @walkingrandomly: x=seq(-2,2,0.001)\ny=Re((sqrt(cos(x))*cos(200*x)+sqrt(abs(x))-0.7)*(4-x*x)^0.01)
## 2                                               RT @Rbloggers: Direct forecast X Recursive forecast ht
## 3                                               RT @Rbloggers: Direct forecast X Recursive forecast ht
##   retweet_count favorite_count is_quote_status quote_status_id is_retweet
## 1            36              0           FALSE            <NA>       TRUE
## 2             2              0           FALSE            <NA>       TRUE
## 3             2              0           FALSE            <NA>       TRUE
##    retweet_status_id in_reply_to_status_status_id
## 1 696747727909093378                         <NA>
## 2 951225902259679233                         <NA>
## 3 951225902259679233                         <NA>
##   in_reply_to_status_user_id in_reply_to_status_screen_name lang
## 1                       <NA>                           <NA>   en
## 2                       <NA>                           <NA>   en
## 3                       <NA>                           <NA>   en
##                  source media_id media_url media_url_expanded urls
## 1    Twitter for iPhone     <NA>      <NA>               <NA> <NA>
## 2        rtapp315156161     <NA>      <NA>               <NA> <NA>
## 3 ttools it knowingness     <NA>      <NA>               <NA> <NA>
##      urls_display           urls_expanded mentions_screen_name
## 1            <NA>                    <NA>      walkingrandomly
## 2 wp.me/pMm6L-F4y https://wp.me/pMm6L-F4y            Rbloggers
## 3 wp.me/pMm6L-F4y https://wp.me/pMm6L-F4y            Rbloggers
##   mentions_user_id symbols           hashtags coordinates place_id
## 1         92746008      NA             rstats          NA     <NA>
## 2        144592995      NA rstats DataScience          NA     <NA>
## 3        144592995      NA rstats DataScience          NA     <NA>
##   place_type place_name place_full_name country_code country
## 1       <NA>       <NA>            <NA>         <NA>    <NA>
## 2       <NA>       <NA>            <NA>         <NA>    <NA>
## 3       <NA>       <NA>            <NA>         <NA>    <NA>
##   bounding_box_coordinates bounding_box_type
## 1                     <NA>              <NA>
## 2                     <NA>              <NA>
## 3                     <NA>              <NA>
```

Retweets

A retweet is when you or someone else shares someone elses tweet so your / their followers can see it. It is similar to sharing in Facebook where you can add a quote or text above the retweet if you want or just share the post. Let’s use the same query that you used above but this time ignore all retweets by setting the `include_rts` argument to `FALSE`. You can get tweet / retweet stats from your dataframe, separately.

```
# find recent tweets with #rstats but ignore retweets
rstats_tweets <- search_tweets("#rstats", n = 500,
                             include_rts = FALSE)
# view top 2 rows of data
head(rstats_tweets, n = 2)
##     screen_name   user_id          created_at          status_id
## 1     Rbloggers 144592995 2018-01-10 22:55:02 951225902259679233
## 2 guangchuangyu  20828110 2018-01-10 22:54:10 951225682578821127
##
## 1                         Direct forecast X Recursive forecast https://t.co/tGLPqglRD3 #rstats #DataSci
## 2 custom background list for ReactomePA https://t.co/o51G2edp82 https://t.co/4guwmJBYKw #reactomepa #rs
##   retweet_count favorite_count is_quote_status quote_status_id is_retweet
## 1             2              0           FALSE            <NA>      FALSE
## 2             0              0           FALSE            <NA>      FALSE
##   retweet_status_id in_reply_to_status_status_id
## 1              <NA>                         <NA>
## 2              <NA>                         <NA>
##   in_reply_to_status_user_id in_reply_to_status_screen_name lang
## 1                       <NA>                           <NA>   en
## 2                       <NA>                           <NA>   en
##           source media_id media_url media_url_expanded urls
## 1 r-bloggers.com     <NA>      <NA>               <NA> <NA>
## 2          IFTTT     <NA>      <NA>               <NA> <NA>
##                     urls_display
## 1              wp.me/pMm6L-F4y
## 2 ift.tt/2Di8hsq ift.tt/1ZjPYGD
##                                urls_expanded mentions_screen_name
## 1                      https://wp.me/pMm6L-F4y                 <NA>
## 2 http://ift.tt/2Di8hsq http://ift.tt/1ZjPYGD                 <NA>
##   mentions_user_id symbols           hashtags coordinates place_id
## 1             <NA>      NA rstats DataScience          NA     <NA>
## 2             <NA>      NA  reactomepa rstats          NA     <NA>
##   place_type place_name place_full_name country_code country
## 1       <NA>       <NA>            <NA>         <NA>    <NA>
## 2       <NA>       <NA>            <NA>         <NA>    <NA>
##   bounding_box_coordinates bounding_box_type
## 1                     <NA>              <NA>
## 2                     <NA>              <NA>
```

Next, let’s figure out who is tweeting about `R` using the `#rstats` hashtag.

```
# view column with screen names - top 6
head(rstats_tweets$screen_name)
## [1] "Rbloggers"       "guangchuangyu"   "ImDataScientist" "LessCrime"
## [5] "martinjhnhadley" "Rbloggers"
# get a list of unique usernames
unique(rstats_tweets$screen_name)
##   [1] "Rbloggers"       "guangchuangyu"   "ImDataScientist"
##   [4] "LessCrime"       "martinjhnhadley" "hspter"
##   [7] "tipsder"         "kierisi"         "dataandme"
##  [10] "pscheid92"       "leila_etaati"    "Nujcharee"
##  [13] "mauro_lepore"    "JidduAlexander"  "ludmila_janda"
##  [16] "albz_marocchino" "CRANberriesFeed" "LearnRinaDay"
##  [19] "rweekly_live"    "devlintufts"     "zentree"
##  [22] "mementonature"   "edelponte"       "annakasdan"
##  [25] "mattwilkinsbio"  "DerFredo"        "ma_salmon"
##  [28] "MikeRSpencer"    "statstools"      "jtrnyc"
##  [31] "bettytalknerdy"  "AriLamstein"     "maxheld"
##  [34] "NovasTaylor"     "mackfinkel"      "RLadiesNYC"
##  [37] "thosjleeper"     "benmarwick"      "StefanieButland"
##  [40] "sammydeprez"     "nataliemahara"   "KenSteif"
##  [43] "EarthLabCU"      "tangming2005"    "Cruz_Julian_"
##  [46] "taavipall"       "swansea_r"       "travisgerke"
##  [49] "DruidSmith"      "thanhtungmilan"  "osazuwa"
##  [52] "joranelias"      "AppsilonDS"      "gjmount"
##  [55] "Progressive_MA"  "lincolnmullen"   "cainesap"
##  [58] "rstatsdata"      "jebyrnes"        "marinereilly"
##  [61] "MorrisonLisbeth" "mtrost2"         "RebeccaNLewis"
##  [64] "Gaming_Dude"     "SamuelJenness"   "lisafederer"
##  [67] "rtelmore"        "AniMove"         "LeafyEricScott"
##  [70] "ucfagls"         "FrancoisKeck"    "kklmmr"
##  [73] "MatthewRenze"    "_ColinFay"       "statwonk"
##  [76] "WarwickRUG"      "alex__morley"    "boxuancui"
##  [79] "ZipperSam"       "pkqstr"          "microheather"
##  [82] "neuromusic"      "FourMInfo"       "nielsberglund"
##  [85] "danws7"          "paulvanderlaken" "BigDataInsights"
##  [88] "Jemus42"         "JMFradeRue"      "AnalyticsVidhya"
##  [91] "sharon000"       "d_mathlete"      "RLangTip"
##  [94] "VascoElbrecht"   "thinkR_fr"       "anthonytowry"
##  [97] "jarostat"        "_arnaudr"        "zevross"
## [100] "jlisic"          "BradleyJEck"     "BroVic"
## [103] "humeursdevictor" "j_ken95"         "henrikbengtsson"
## [106] "n_ashutosh"      "f_chiare"        "Ognyanova"
## [109] "thonoir"         "bizScienc"       "eleafeit"
## [112] "axiomsofxyz"     "pofigster"       "GeostatsGuy"
## [115] "DesertIsleSQL"   "SimplyApprox"    "peterdalle"
## [118] "njogukennly"     "sthda_en"        "moorejh"
## [121] "jhollist"        "ecoevoenviro"    "juliasilge"
## [124] "datascienceplus" "londonaesthetik" "bobehayes"
## [127] "tjmahr"          "expersso"        "GarethNetto"
## [130] "rushworth_a"     "blebeau11"       "OilGains"
## [133] "murnane"         "hrbrmstr"        "Ed_pheasant"
## [136] "Benavent"        "coraman"         "MikeTaylorSEMO"
## [139] "ilustat"         "ingorohlfing"    "m4xl1n"
## [142] "znmeb"           "f2harrell"       "jonclayden"
## [145] "mdsumner"        "ixek"            "RLadiesPhilly"
## [148] "LeNematode"      "mgvolz"          "moroam"
## [151] "auth0"           "Jadirectivestwt" "ZarahPattison"
## [154] "zx8754"          "jamesday87"      "DBNTechEvents"
## [157] "Xtophe_Bontemps" "jananivijayan1"  "sqlsatvienna"
## [160] "statsforbios"    "jetrubyagency"   "ParmutiaMakui"
## [163] "deekareithi"     "msciain"         "datasetfree"
## [166] "MattOldach"      "statlurker"      "marcbeldata"
## [169] "cvonbastian"     "KevinWang009"    "lassehmadsen"
## [172] "biobenkj"        "gombang"         "whatsgoodio"
## [175] "forrestdougan"   "GVPhD"           "MikeTreglia"
## [178] "ChristinZasada"  "tonmcg"          "acalatr"
## [181] "John_deVillier"  "brodriguesco"    "ThomasSpeidel"
## [184] "daattali"        "USGS_LMG"        "gdisney_melb"
## [187] "jasonbaik94"     "bramasolo"       "translatedmed"
## [190] "DeanLittle"      "TermehKousha"    "tetsuroito"
## [193] "LuisDVerde"      "jrescalante"     "nlj"
## [196] "mlandstat"       "brendankness"    "scottyd22"
## [199] "orlandomezquita" "peterdavenport8" "d_joseph_parker"
## [202] "NicoleAlineData" "BrockTibert"     "monkmanmh"
## [205] "chendaniely"     "spkaluzny"       "julianjon"      
## [208] "PlantLearner"    "StatStas"        "tjardine"       
## [211] "Benjaming_G"     "NickDoesData"    "_jwinget"       
## [214] "starryflo"       "nj_tierney"      "rudeboybert"    
## [217] "presidual"       "earowang"        "ReddTrain"      
## [220] "GojThomson"      "abresler"        "CasalsTMarti"   
## [223] "alspur"          "AedinCulhane"    "sellorm"        
## [226] "DJAnderson_07"   "Gui42"           "pabloc_ds"      
## [229] "mjhendrickson"   "adolfoalvarez"   "daniellequinn88"
## [232] "jbryer"          "paleolimbot"     "TheBIccountant" 
## [235] "gladwinmuchena"  "HFazelinia"      "madforsharks"   
## [238] "SwindleApe"      "lumbininep"      "dgkeyes"        
## [241] "_djli"           "Kwarizmi"        "dj_shaily"      
## [244] "NCrepalde"       "JohnBVincent"    "jdossgollin"    
## [247] "PyData"          "RLadiesQuito"    "TELLlab"        
## [250] "timelyportfolio" "revodavid"       "villasenor_jc"  
## [253] "hadleywickham"   "NumFOCUS"        "joshua_ulrich"  
## [256] "dccc_phd"        "abiyugiday"      "HansLive"       
## [259] "natedayta"       "gbasultoe"       "MineDogucu"     
## [262] "riverpeek"       "southerndsc"     "nolauren"       
## [265] "StatGarrett"     "Elmore_Ecology"  "GIST_ORNL"      
## [268] "thmscwlls"       "dpereira14"      "kwbroman"       
## [271] "JesseOPiburn"    "seabbs"          "AgentZeroNine"  
## [274] "StrictlyStat"    "robertstats"     "SteffLocke"     
## [277] "gokhan_ciflikli" "d8aninja"        "DWPDigital"     
## [280] "verajosemanuel"  "RR_Oxford"
```

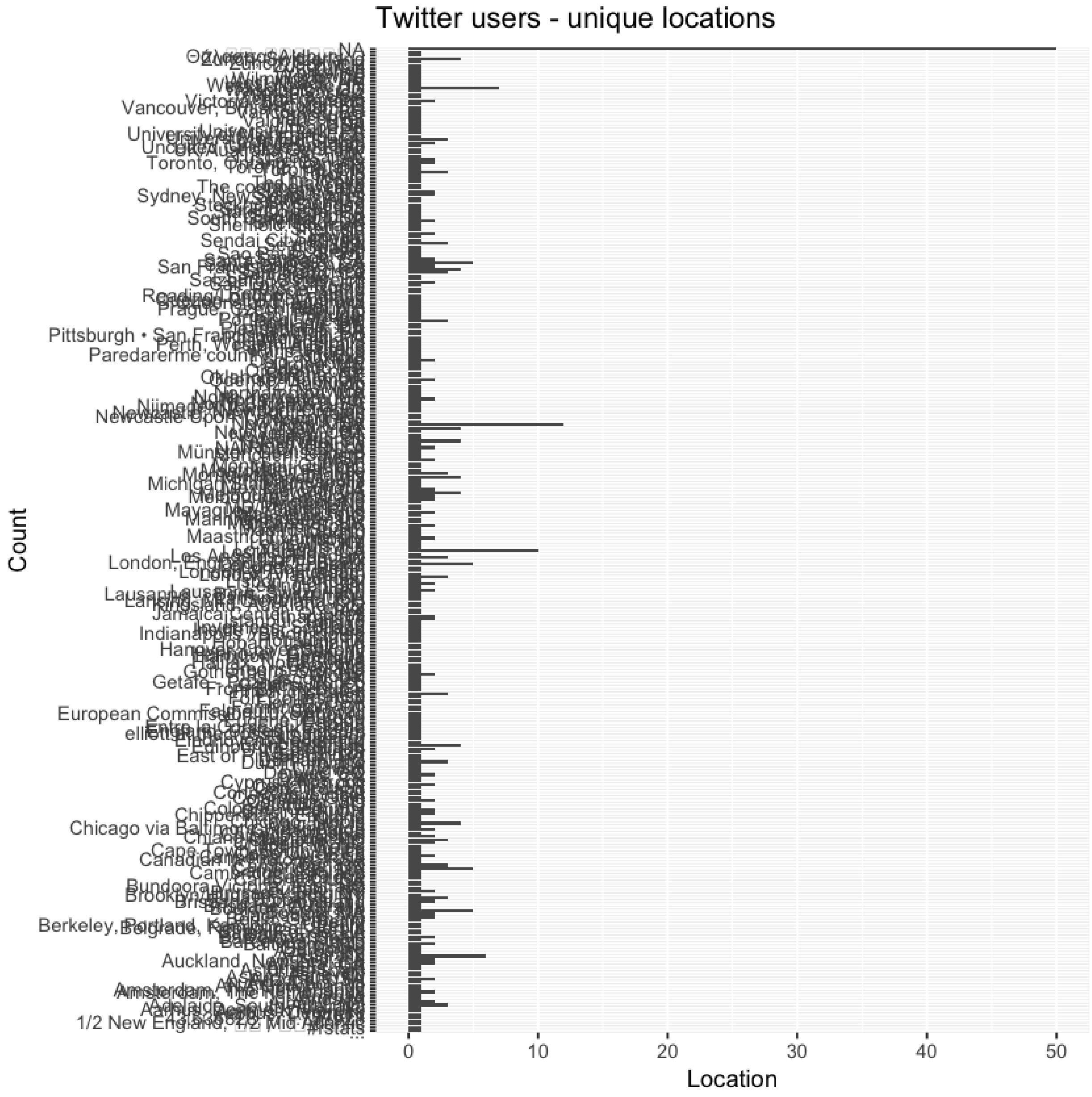
You can similarly use the `search_users()` function to just see what users are tweeting using a particular hashtag. This function returns just a data.frame of the users and information about their accounts.

```
# what users are tweeting with #rstats
users <- search_users("#rstats",
                      n = 500)
# just view the first 2 users - the data frame is large!
head(users, n = 2)
##     user_id        name screen_name                              location
## 1 517921400 Leah Wasser LeahAWasser                     Boulder, Colorado
## 2 342250615    rOpenSci    rOpenSci Berkeley, Portland, Kamloops, Utrecht
##
## 1 Director, Earth Analytics Education @CUBoulder @EarthLabcu #remoteSensing #ecology #openScience #GIS
## 2                                           rOpenSci develops #rstats-based tools to facilitate open sc
##   protected followers_count friends_count listed_count          created_at
## 1     FALSE            1579          1392          132 2012-03-07 20:42:07
## 2     FALSE           16560           508          719 2011-07-25 18:24:54
##   favourites_count utc_offset                   time_zone geo_enabled
## 1             2994     -25200 Mountain Time (US & Canada)       FALSE
## 2              875     -28800  Pacific Time (US & Canada)        TRUE
##   verified statuses_count lang contributors_enabled is_translator
## 1    FALSE           4044   en                FALSE         FALSE
## 2     TRUE           3915   en                FALSE         FALSE
##   is_translation_enabled profile_background_color
## 1                  FALSE                   000000
## 2                  FALSE                   C0DEED
##                       profile_background_image_url
## 1 http://abs.twimg.com/images/themes/theme8/bg.gif
## 2 http://abs.twimg.com/images/themes/theme1/bg.png
##                   profile_background_image_url_https
## 1 https://abs.twimg.com/images/themes/theme8/bg.gif
## 2 https://abs.twimg.com/images/themes/theme1/bg.png
##   profile_background_tile
## 1                   FALSE
## 2                   FALSE
##                                                             profile_image_url
## 1 http://pbs.twimg.com/profile_images/895843973704466432/eBl5QwIb_normal.jpg
## 2 http://pbs.twimg.com/profile_images/878348237496762368/yUU7Pefs_normal.jpg
##                                                        profile_image_url_https
## 1 https://pbs.twimg.com/profile_images/895843973704466432/eBl5QwIb_normal.jpg
## 2 https://pbs.twimg.com/profile_images/878348237496762368/yUU7Pefs_normal.jpg
##                                                           profile_image_url.1
## 1 http://pbs.twimg.com/profile_images/895843973704466432/eBl5QwIb_normal.jpg
## 2 http://pbs.twimg.com/profile_images/878348237496762368/yUU7Pefs_normal.jpg
##                                                      profile_image_url_https.1
## 1 https://pbs.twimg.com/profile_images/895843973704466432/eBl5QwIb_normal.jpg
## 2 https://pbs.twimg.com/profile_images/878348237496762368/yUU7Pefs_normal.jpg
##   profile_link_color profile_sidebar_border_color
## 1             981CEB                       000000
## 2             1DA1F2                       C0DEED
##   profile_sidebar_fill_color profile_text_color
## 1                     000000             000000
## 2                     DDEEF6             333333
##   profile_use_background_image default_profile default_profile_image
## 1                        FALSE           FALSE                 FALSE
## 2                         TRUE            TRUE                 FALSE
##                                            profile_banner_url
## 1 https://pbs.twimg.com/profile_banners/517921400/1439854963
## 2 https://pbs.twimg.com/profile_banners/342250615/1398878552
```

Let’s learn a bit more about these people tweeting about `R`. First, where are they from?

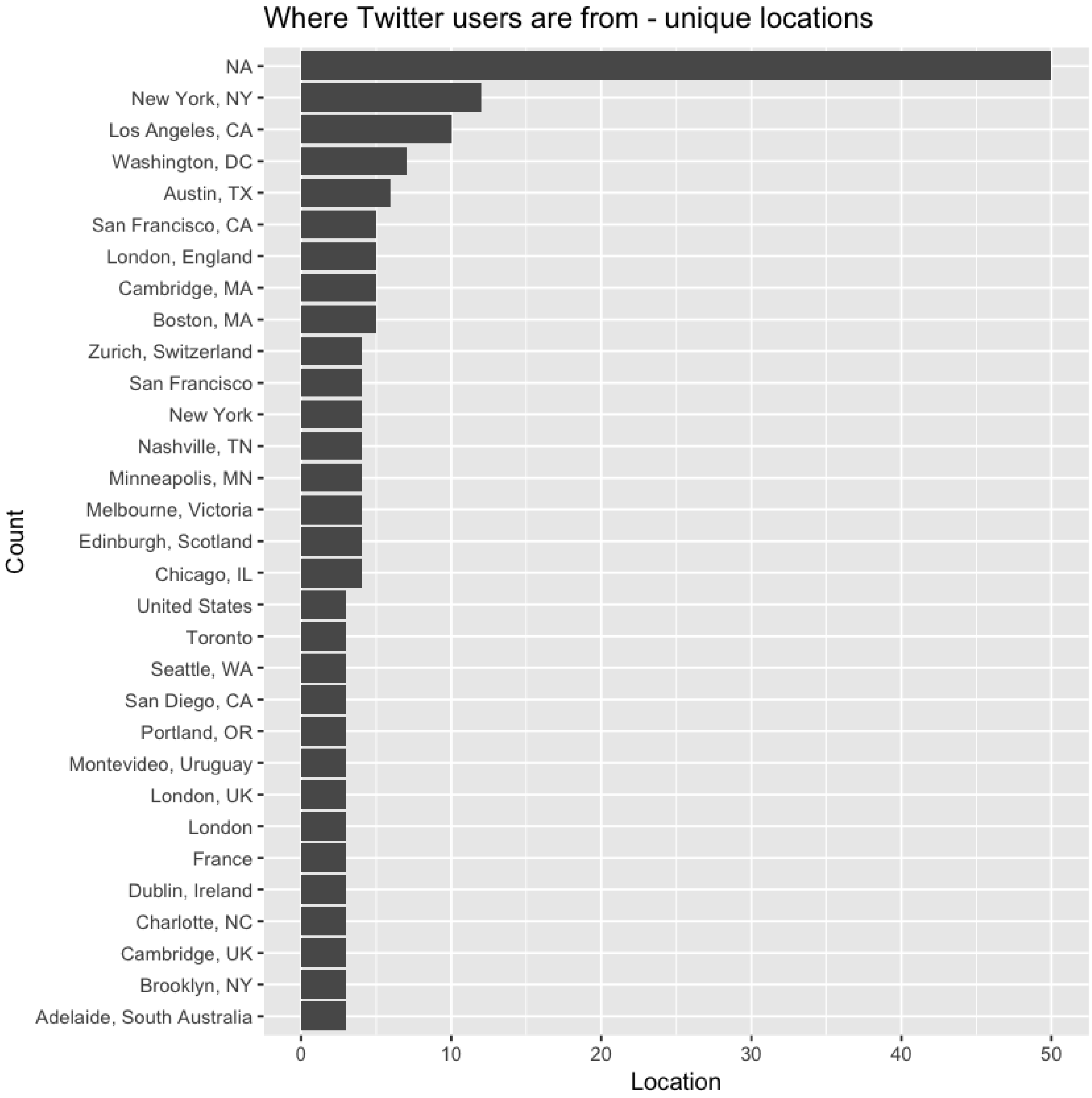
```
# how many locations are represented
length(unique(users$location))
## [1] 304

users %>%
  ggplot(aes(location)) +
  geom_bar() + coord_flip() +
      labs(x = "Count",
      y = "Location",
      title = "Twitter users - unique locations ")
```



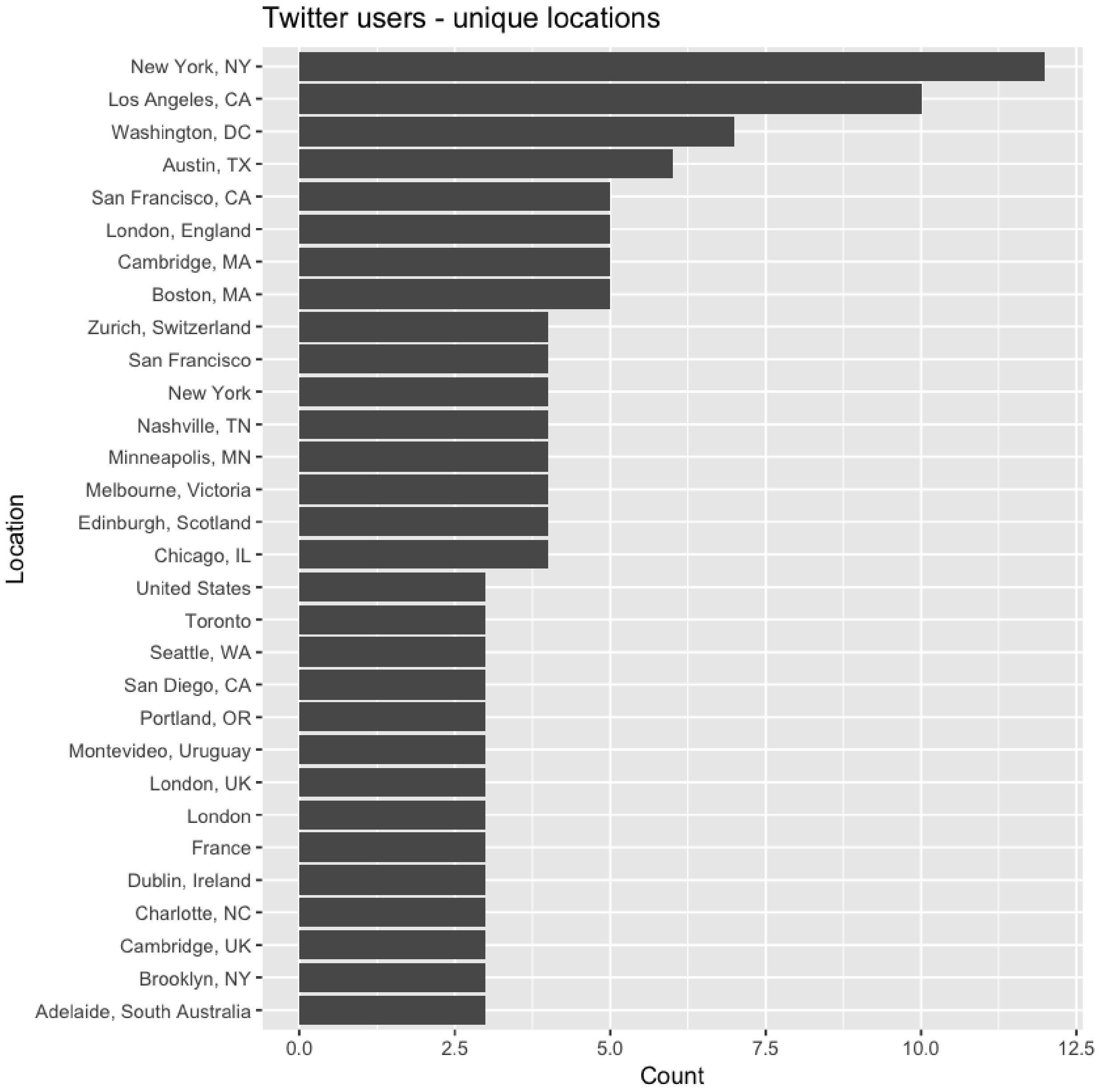
Let’s sort by count and just plot the top locations. To do this you use top_n(). Note that in this case you are grouping your data by user. Thus top_n() will return locations with atleast 15 users associated with it.

```
users %>%
  count(location, sort = TRUE) %>%
  mutate(location = reorder(location, n)) %>%
  top_n(20) %>%
  ggplot(aes(x = location, y = n)) +
  geom_col() +
  coord_flip() +
      labs(x = "Count",
      y = "Location",
      title = "Where Twitter users are from - unique locations ")
```



It looks like you have some `NA` or no data values in your list. Let's remove those with `na.omit()`.

```
users %>%
  count(location, sort = TRUE) %>%
  mutate(location = reorder(location,n)) %>%
  na.omit() %>%
  top_n(20) %>%
  ggplot(aes(x = location,y = n)) +
  geom_col() +
  coord_flip() +
      labs(x = "Location",
      y = "Count",
      title = "Twitter users - unique locations ")
```



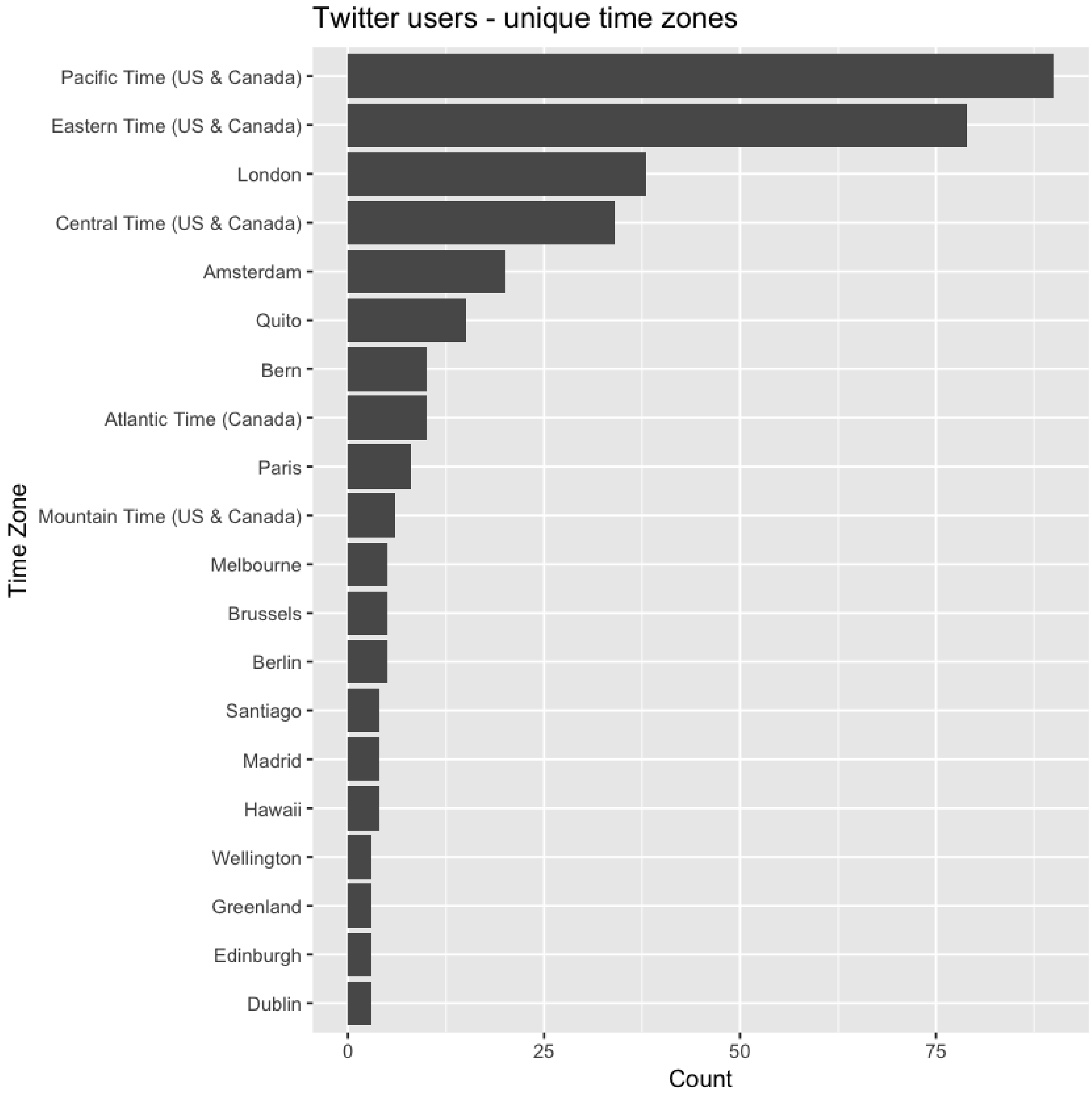
Looking at your data, what do you notice that might improve this plot? There are 314 unique locations in your list. However, everyone didn't specify their locations using the approach. For example some just identified their country: United States for example and others specified a city and state. You may want to do some cleaning of these data to be able to better plot this distribution - especially if you want to create a map of these data!

Users by Time Zone

Let's have a look at the time zone field next.

Optional Challenge

Use the example above, plot users by time zone. List time zones that have at least 20 users associated with them. What do you notice about the data?



The plots above aren't perfect. What do you start to notice about working with these data? Can you simply download them and plot the data?

Data munging 101

When you work with data from sources like NASA, USGS, etc. there are particular cleaning steps that you often need to do. For instance:

- you may need to remove nodata values
- you may need to scale the data
- and others

In the next lesson you will dive deeper into the art of "text-mining" to extract information about a particular topic from twitter.

Additional Resources

- Tidy text mining online book
- A great overview of the rtweet package by Mike Kearny
- A blog post on tidytext by Francois Michonneau
- About the twitter API rate limit

Twitter data for Science Twitter Data Text Mining

Tags **Social science**: social media **Data exploration and analysis**: text mining **Find and manage data**: apis , find data
Updated: January 10, 2018

LEAVE A COMMENT

13 Comments **earthlabcu** 1 **Login**

Recommend Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

skjainmiah • a year ago

when i executed the above code i got this error:
Error in init_oauth1.0(self$endpoint, self$app, permission = self$params$permission, :
Unauthorized (HTTP 401).

17 ^ | v • Reply • Share ›

earthlab Mod ➦ skjainmiah • 8 months ago

Yes - to be clear - you can't just run the code below. you have to first setup an app as suggested below. the steps to set this app up are at the top of the page.

-- https://cran.r-project.org/...

```
# create token named "twitter_token"
twitter_token <- create_token(
app = appname,
consumer_key = key,
consumer_secret = secret)
```

then you'll replace the appnam, key and secret in the code above with your specific appname, key and secret!

^ | v • Reply • Share ›

⌃ | ⌄ • Reply • Share ›

Ahsaas Chawla ➦ skjainmiah • 10 months ago

Just create another token from the Twitter interface and use that. If you haven't created an App, go and create one at apps.twitter.com. It's an authentication error that you get while using the wrong appname, key or secret

⌃ | ⌄ • Reply • Share ›

Leah Wasser ➦ skjainmiah • a year ago

it looks like a few other people upvoted this so you are not the only one receiving this error! It would be helpful if someone sees this to specifically let us know where the failure occurs. that *looks* like a twitter authentication error. This may suggest that your api key isnt correct however i am just guessing here!

⌃ | ⌄ • Reply • Share ›

Martyns Nwaokocha ➦ Leah Wasser • 8 months ago

In your Twitter app settings, uncheck "Enable Callback Locking (It is recommended to enable callback locking to ensure apps cannot overwrite the callback url)"

⌃ | ⌄ • Reply • Share ›

earthlab Mod ➦ Martyns Nwaokocha • 8 months ago

thanks Martyns. Has this worked for anyone else?

⌃ | ⌄ • Reply • Share ›

Justin Rojas • 10 months ago

has anyone figured how find the function for time_zone()? for the optional challenge really having a hard time

⌃ | ⌄ • Reply • Share ›

Daegun Bong • 10 months ago

the search_tweets() and search_users() functions don't have utc_offset or time_zone columns returned in the data frame

⌃ | ⌄ • Reply • Share ›

Koleen BP • 10 months ago

March 2018:
I just noticed that we need to add "httpuv" package as a dependency in `twitter_token` variable.
In order to do that, add the following codes on the beginning, before you declare variables for `appname`, `key` and `secret`:

install.packages("httpuv")
library(httpuv)

Failure to add these code will give you this error:

"Error in oauth_listener(authorize_url, is_interactive) :
httpuv package required to capture OAuth credentials."

Thanks!

⌃ | ⌄ • Reply • Share ›

Sutter_Cane • a year ago

Guys, na.omit() by itself doesn't do anything. Can you explain what should be in the ()?

⌃ | ⌄ • Reply • Share ›

Leah Wasser ➦ Sutter_Cane • a year ago

hey Sutter. it is true that na.omit() on it's own in the console doesnt work run without being passed an object. however in a pipe, it should recieve an object from the previous line of the pipe. In this case it should receive a data.frame object. did something specific in the code not work for you?

⌃ | ⌄ • Reply • Share ›

Sutter_Cane ➦ Leah Wasser • a year ago

Yes, I did my own example based on this one. Everything was pretty much similar (in terms of coding) but na.omit() did not get rid of no data values.

⌃ | ⌄ • Reply • Share ›

Leah Wasser ➦ Sutter_Cane • a year ago

this works for me! Perhaps be a bit more specific?

```
some_data <- data.frame(location = c("one", "two", "three"),
count = c(1,2,3))

sum(is.na(some_data))

some_data$count[3] <- NA
sum(is.na(test))

some_data_withoutna <- some_data %>%
na.omit()

sum(is.na(some_data_withoutna))
```

⌃ | ⌄ • Reply • Share ›

ALSO ON **EARTHLABCU**

Lesson 4. Calculate a remote sensing derived vegetation …

7 comments • a year ago

Avatar **earthlab** — i'd check out some of the tools on Ropensci. i believe there are some API …

Lesson 4. Use tidytext to text mine social media - twitter …

2 comments • a year ago

Avatar **Sina Özdemir** — For the life of me, I cannot

Setup Git, Bash, and Anaconda on Your …

6 comments • 7 months ago

Avatar **nasa** — hi jenny, hi don. im a new anaconda user and i installed using the "Add …

Download MODIS data programmatically with …

1 comment • 2 years ago

Avatar **Leah A Wasser** — Testing comments! the

Avatar figure out how you got the removeWords … Avatar glob library is my favorite :)