



R news and tutorials contributed by hundreds of R bloggers

- [Home](#)
- [About](#)
- [RSS](#)
- [add your blog!](#)
- [Learn R](#)
- [R jobs](#) ♦♦♦
- [Contact us](#)

Welcome!

Follow @rbloggers { 83.1k }

Here you will find daily **news and tutorials about R**, contributed by hundreds of bloggers. There are many ways to **follow us** - [By e-mail](#):

52637 readers
BY FEEDBURNER

[On Facebook](#):

R blog...
7.8 萬 按讚次數

你和其他 30 位朋友都說這個讚

If you are an **R blogger yourself** you are invited to [add your own R content feed to this site](#) (Non-English R bloggers should add themselves- [here](#))

[Jobs for R-users](#)

- [Data Science Engineer @ NY](#)
- [Data Scientist Position for Developing Software and Tools in Genomics, Big Data and Precision Medicine](#)
- [Senior Scientist, Translational Informatics @ Vancouver, BC, Canada](#)
- [Senior Principal Data Scientist @ Mountain View, California, United States](#)
- [Technical Research Analyst](#)

Recent Posts

- [Le Monde puzzle \[#1141\]](#)
- [#0: Introducing T^4: Tips, Tricks, Tools, and Toys](#)
- [Securing and monitoring ShinyProxy deployment of R Shiny apps](#)
- [How to Become a Data Scientist in 2020](#)
- [Statistics Sunday: My 2019 Reading](#)
- [Revitalizing R package.yorkr](#)
- [Updating packages on a drat repo](#)
- [An update to “An adventure in downloading books”](#)
- [Installation of R 4.0 on Ubuntu 20.04 LTS and tips for spatial packages](#)
- [Text Plots](#)
- [Riddler: Can You Flip Your Way To Freedom?](#)
- [Europe COVID-19 death map](#)
- [Generative art: Let your computer design you a painting](#)
- [Checking your Package for Compatibility with R 4.0.0](#)
- [Combine values into a vector](#)

Other sites

- [Jobs for R-users](#)
- [SAS blogs](#)

Visualising thefts using heatmaps in ggplot2

August 17, 2015

By [Teja Kodali](#)

Like 11 Share Tweet Share



[This article was first published on [DataScience+](#), and kindly contributed to [R-bloggers](#).
(You can report issue about the content on this page [here](#))

Want to share your content on R-bloggers? [click here](#) if you have a blog, or [here](#) if you don't.

Share

Tweet

This is a continuation of my [previous article](#), where I gave a basic overview of how to construct heatmaps in R. Here, I will show you how to use R packages to build a heatmap on top of the map of Chicago to see which areas have the most amount of crime. We will require two packages for the mapping, namely `maps`, and `ggmap`. We will also use two more packages, `dplyr`, and `tidyr`.

I will be using the Motor Vehicle Theft Data from Chicago, which can be obtained from the [City of Chicago Data Portal](#).

The second part of the code will contain the following steps:

- Removing empty locations. For some thefts, the location is not recorded, and the field is left blank. This caused problems for me later on, so I decided to remove all such locations. At the time of writing, empty locations made up about 2500 entries of the total 278000 entries.
- Splitting the location column into latitude and longitude. The location column consists of data in the form (x, y) and is of the character class. We want x and y to be in separate columns and be of the numeric class. We will use the dplyr and tidyr libraries for this.
- Get the map of Chicago. We will use the ggmap library for this
- Plotting the location heatmap.

Here is the code:

```
## Removing empty locations
chicagoMVT$Location[chicagoMVT$Location == ''] <- NA
chicagoMVT <- na.omit(chicagoMVT)
```

We set all empty values to NA, and then change the original dataset so that it no longer contains the NA values.

```
## Splitting location into latitude and longitude
chicagoMVT %>% extract(Location, c('Latitude', 'Longitude'), '([^\s,]+), ([^\s,]+)\s')
chicagoMVT$Longitude <- round(as.numeric(chicagoMVT$Longitude), 2)
chicagoMVT$Latitude <- round(as.numeric(chicagoMVT$Latitude), 2)
```

%>% is called the **pipe operator**. It is from the **dplyr** package. The above line of code is the same as writing

```
chicagoMVT <- extract(chicagoMVT, Location, c('Latitude', 'Longitude'), '([^\s,]+), ([^\s,]+)\s')
```

As we can see, the pipe operator is very helpful to pass output resulting from one operation to another. While it isn't particularly useful here, it's usefulness becomes apparent when you have to perform multiple operations, and don't want to create a temporary variable for the result of each of the operations. I will do a short of tutorial and demonstration of dplyr in my next article.

The extract method is from the **tidyr** package, and I am using it to separate the Location column into Latitude and Longitude. The last parameter of the extract method is a Regular Expression or RegEx for short. I often have problems with RegEx, and this time was no different, and I'd like to thank StackOverflow user [nongkrong](#) for helping me with that.

Both dplyr and tidyr are great packages written by Hadly Wickham, the man who revolutionised R. I highly recommend that you check out his body of work and read his books.

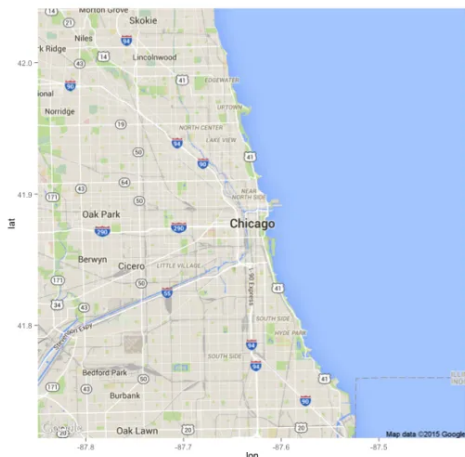
Next, we will get the map of Chicago, so that we plot on top of it.

```
library(ggmap)
chicago <- get_map(location = 'chicago', zoom = 11)
```

If you would like to see the map, you can use this command:

```
ggmap(chicago)
```

which will give the following map



Now we will create a data frame containing the coordinates of

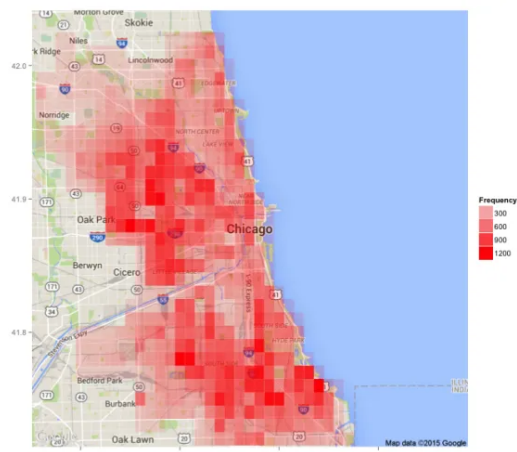
```
locationCrimes <- as.data.frame(table(chicagoMVT$Longitude, chicagoMVT$Latitude))
names(locationCrimes) <- c('long', 'lat', 'Frequency')
locationCrimes$long <- as.numeric(as.character(locationCrimes$long))
locationCrimes$lat <- as.numeric(as.character(locationCrimes$lat))
locationCrimes <- subset(locationCrimes, Freq > 0)
```

As we saw in the above map, the axes are named 'long' and 'lat', so we will use the same naming convention for our data frame. When we create the data frame, the latitude and longitude get converted to the factor class. To convert it back to numeric, we have to first convert them back to character, and then to numeric (otherwise it will give an error). Finally, we remove all data points where there were no crimes recorded. If you don't do this, the resulting plot will have a lot of tiles plotted on the water, which we don't want.

```
ggmap(chicago) + geom_tile(data = locationCrimes, aes(x = long, y = lat, alpha = Frequency),
  fill = 'red') + theme(axis.title.y = element_blank(), axis.title.x = element_blank())
```

alpha = Frequency will set how transparent/opaque each tile is, based on the frequency of crimes in that particular area. theme(axis.title.y = element_blank(), axis.title.x = element_blank()) will remove the titles for the axes.

This will generate the following plot:



The plot gives us a pretty good idea of which areas are the most prone to thefts, and thus should be avoided. The police are using advanced crime prediction algorithms to predict where crimes will happen, and prevent them before they occur. [Here](#) is a great article by MIT Technology Review on the topic.

The full repo can be found on [GitHub](#).

That's it for now! I hope you enjoyed the article, and found it helpful. Feel free to leave a comment if you have any questions or contact me on [Twitter](#)!

◆ 10 comments on this item ◆ Share on Facebook ◆ Stumble It! ◆ Digg This!

f Share

🐦 Tweet

To leave a comment for the author, please follow the link and comment on their blog: [DataScience+.](#)

R-bloggers.com offers [daily e-mail updates](#) about R news and tutorials about [learning R](#) and many other topics. [Click here if you're looking to post or find an R/data-science job.](#)

Want to share your content on R-bloggers? [click here](#) if you have a blog, or [here](#) if you don't.

If you got this far, why not [subscribe for updates](#) from the site? Choose your flavor: [e-mail](#), [twitter](#), [RSS](#), or [facebook](#)...

Like 11

Share

Tweet

Share

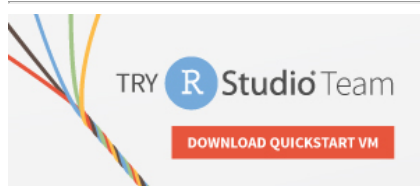
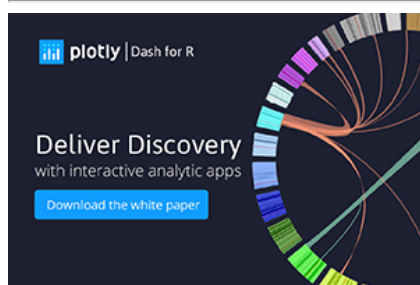
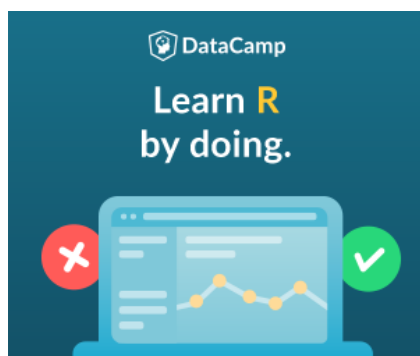
Comments are closed.

Search R-bloggers

Most visited articles of the week

1. [Free Springer Books during COVID19](#)
2. [A package to download free Springer books during Covid-19 quarantine](#)
3. [5 Ways to Subset a Data Frame in R](#)
4. [Which Technology Should I Learn?](#)
5. [How to write the first for loop in R](#)
6. [Causal Inference cheat sheet for data scientists](#)
7. [Tips before migrating to a newer R version](#)
8. [R – Sorting a data frame by the contents of a column](#)
9. [Date Formats in R](#)

Sponsors



DEAR DATA SCIENTISTS,
how to **ease your job**
YUNA


jumping rivers
TRAINING: R, SCALA, STAN

STATWORX
Data Science Service
Data Science | Consulting | Development | Training

Try the **FASTEST ML**
for **R**
Click for a Free Trial
YOTTAMINE ANALYTICS

SIGMA

SIGMA

ALPHIEN JOIN NOW
NEW ALPHATHON LAUNCHED!
Build the best asset allocation model
Win EUR 18,000 & strategy licensing fees

Save 40% on
MANNING
books with code: nlrblog40

neptune.ai Create a FREE account
Experiment Tracking Tool for ML Projects
Now available on CRAN!

Our ads respect your privacy. Read our [Privacy Policy page](#) to learn more.

[Contact us](#) if you wish to help support R-bloggers, and place **your banner here**.

[Jobs for R users](#)

- [Data Science Engineer @ NY](#)
- [Data Scientist Position for Developing Software and Tools in Genomics, Big Data and Precision Medicine](#)
- [Senior Scientist, Translational Informatics @ Vancouver, BC, Canada](#)
- [Senior Principal Data Scientist @](#)

- [Technical Research Analyst – New York, U.S.](#)
- [Movement Building Analyst](#)
- [Innovation Fellow](#)

python-bloggers.com ([python/data-science news](#))

- [Documentation+Pypi for the 'teller', a model-agnostic tool for Machine Learning explainability](#)
- [PyBoy: A Python GameBoy Emulator](#)
- [Free Springer Books during COVID19](#)
- [Encoding your categorical variables based on the response variable and correlations](#)
- [Recreating a Shiny App with Flask](#)
- [Simulating and visualizing the Monty Hall problem in Python & R](#)
- [Predictive Maintenance: Zero to Deployment in Manufacturing](#)

[Full list of contributing R-bloggers](#)

[R-bloggers](#) was founded by [Tal Galili](#), with gratitude to the [R](#) community.

Is powered by [WordPress](#) using a [bavotasan.com](#) design.

Copyright © 2020 **R-bloggers**. All Rights Reserved. [Terms and Conditions](#) for this website