

Tracking Your FitBit Data with R and Shiny

Graham Parsons

2017-04-06

About me

About me

- Data Scientist at Mango
- Previously Analyst at UWE University, where I...
- Spent a long time sat at my desk
- Or sat in a car on my commute



- Bought a Fitbit over a year ago

About FitBit

Sell devices that collect fitness data

- Heart Rate
- Activities (running, cycling, steps, etc.)
- Sleep

Provide data services that

- Provide insights into your activity
- Change your behaviour through gamification
- Provide your data through an API



Why get the data?

- It's yours
- More analysis less gamification
- Mix it with your private data
- Great way to explore analysis techniques
- It's fun!

How to get the data

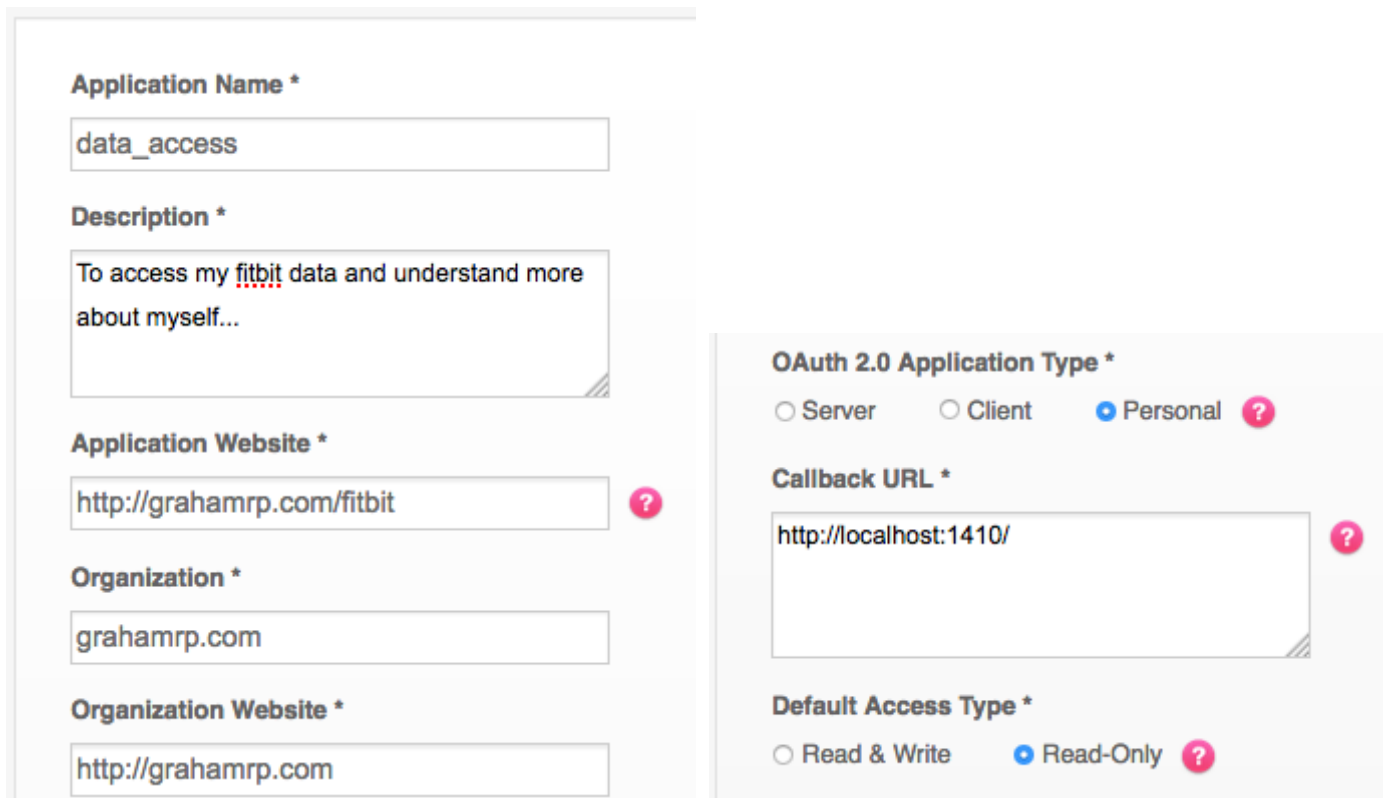
Register

- Register on dev.fitbit.com



Create an App

- Create a "Personal" app to get an API key



The image shows a screenshot of the Google API Console 'Create an App' form. The form is divided into two main sections: 'Application Information' on the left and 'OAuth 2.0 Application Type' on the right. The 'Application Information' section includes fields for 'Application Name *' (filled with 'data_access'), 'Description *' (filled with 'To access my fitbit data and understand more about myself...'), 'Application Website *' (filled with 'http://grahamrp.com/fitbit'), 'Organization *' (filled with 'grahamrp.com'), and 'Organization Website *' (filled with 'http://grahamrp.com'). The 'OAuth 2.0 Application Type' section includes a radio button selection for 'Server', 'Client', and 'Personal' (selected), a 'Callback URL *' field (filled with 'http://localhost:1410/'), and a 'Default Access Type *' section with radio buttons for 'Read & Write' and 'Read-Only' (selected). Red question mark icons are visible next to the 'Application Website', 'Callback URL', and 'Default Access Type' fields.

Application Name *
data_access

Description *
To access my fitbit data and understand more about myself...

Application Website *
http://grahamrp.com/fitbit

Organization *
grahamrp.com

Organization Website *
http://grahamrp.com

OAuth 2.0 Application Type *
☐ Server ☐ Client ☒ Personal

Callback URL *
http://localhost:1410/

Default Access Type *
☐ Read & Write ☒ Read-Only

Get the Data

API Overview

1. Authenticate
2. Make a request
3. Parse the response

The API

1. Authenticate

```
library(httr)
# Set up credentials
fitbit_endpoint <- oauth_endpoint(
  request = "https://api.fitbit.com/oauth2/token",
  authorize = "https://www.fitbit.com/oauth2/authorize",
  access = "https://api.fitbit.com/oauth2/token")
myapp <- oauth_app(
  appname = "data_access",
  key = "123X99", # From "Manage my apps" on dev.fitbit.com
  secret = "jkjxkjlk_mysecret_jlkj9u290jkj")
# Get OAuth token
fitbit_token <- oauth2.0_token(fitbit_endpoint, myapp,
                              scope = "heartrate",
                              use_basic_auth = TRUE)
```

The API

2. Make a request

```
resp <- GET(url = "https://api.fitbit.com/1/user/-/activities/  
heart/date/2017-01-01/2017-01-30.json",  
config = config(token = fitbit_token))
```

See dev.fitbit.com/docs/ for other resource URLs

The API

3. Parse the response

```
library(jsonlite)
tmp <- fromJSON(content(resp, as = "text"))
resting_hr <- data.frame(
  date = tmp$`activities-heart`$dateTime,
  rate = tmp$`activities-heart`$value$restingHeartRate,
  stringsAsFactors = FALSE)
```

And repeat.

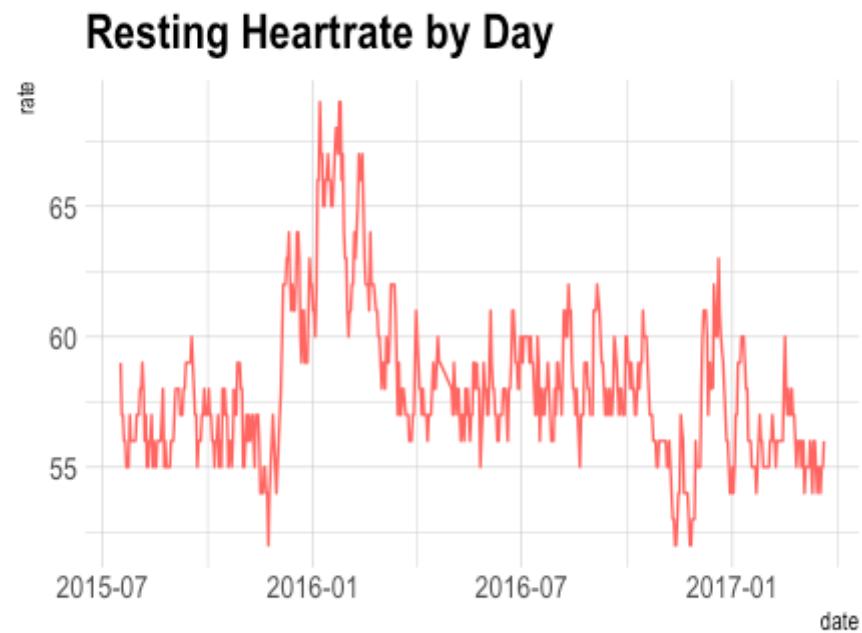
Maybe save it to a local database.

What can you do with your data?

What can you do?

Plot it!

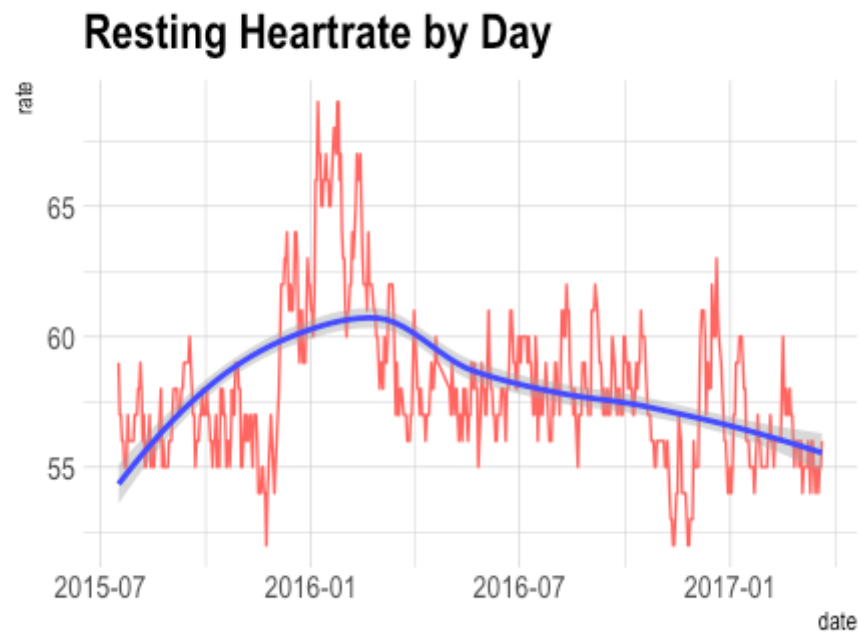
```
ggplot(rhr, aes(x = date, y = rate)) + geom_line()
```



What can you do?

Plot it!

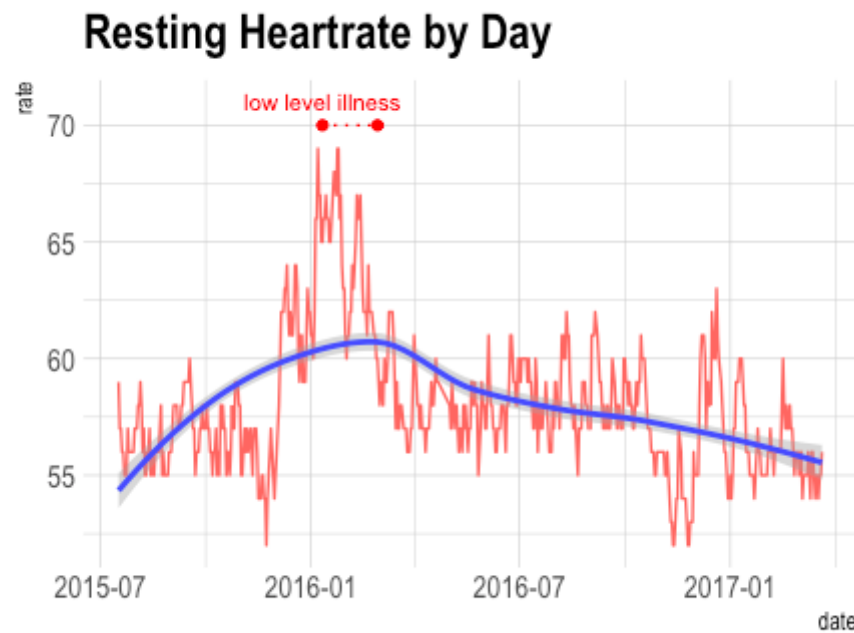
```
ggplot(rhr, aes(x = date, y = rate)) + geom_line() + geom_smooth()
```



What can you do?

Mix in data

```
ggplot(rhr, aes(x = date, y = rate)) + geom_line() + geom_smooth()
```

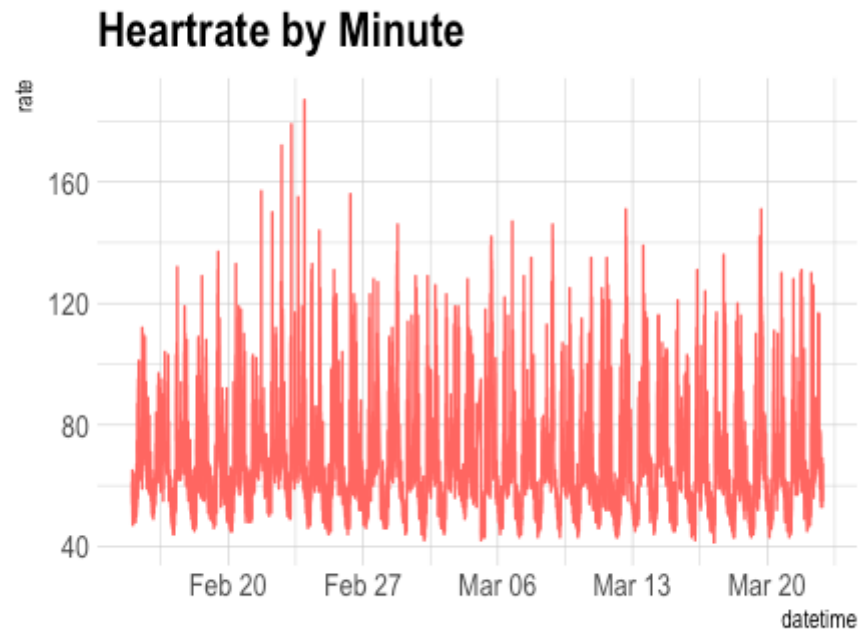


What can you do?

Use interesting techniques

[Twitter Anomaly Detection R Package](#)

Seasonal Hybrid Extreme Studentized Deviate Anomalies!

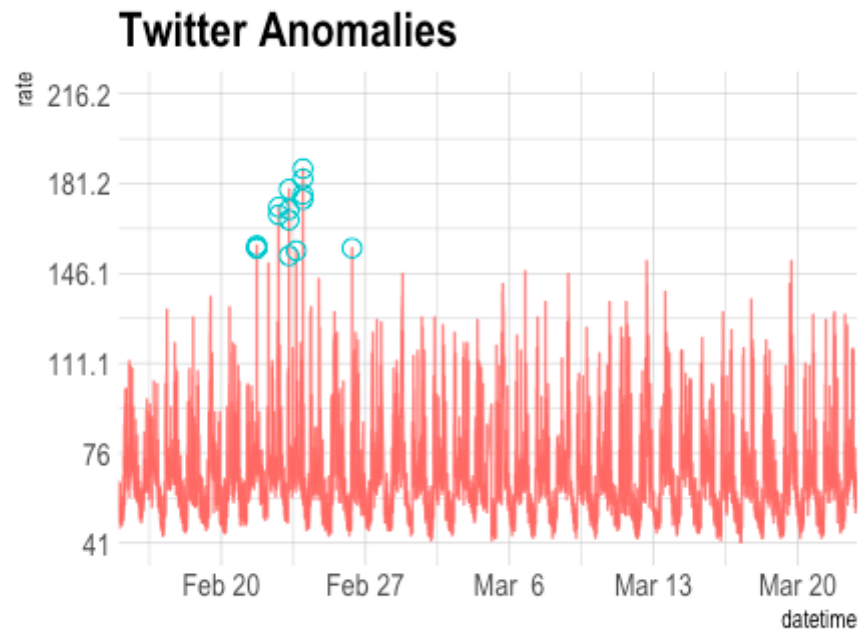


What can you do?

Explore Techniques

[Twitter Anomaly Detection R Package](#)

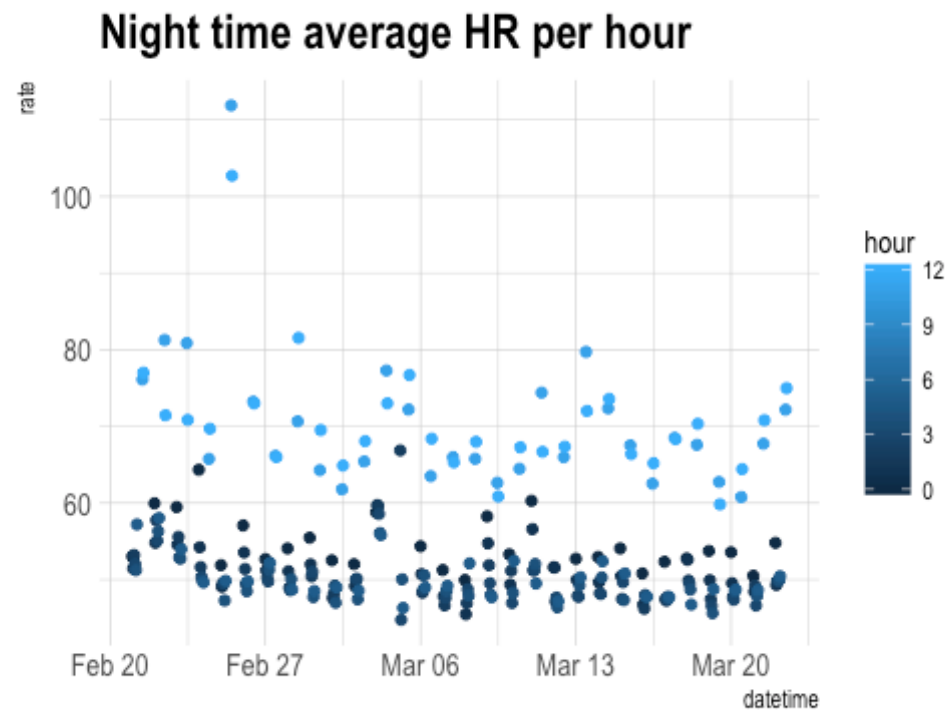
Seasonal Hybrid Extreme Studentized Deviate Anomalies!



What can you do?

Ad hoc queries

What does my sleeping heartrate look like?



What can you do?

Shiny!

- Web application framework for R
- Create interactive web apps
- No HTML, CSS, JavaScript – Just R!
- [My Shiny App](#)



Links

- [Fitbit dev API](#)
- [Digital Health: Tracking Physiomes and Activity Using Wearable Biosensors Reveals Useful Health-Related Information](#)
- [Accessing FitBit API via R](#)
- [Shiny](#)
- [Twitter Anomaly Detection](#)
- [@grpieces](#)