PLUMBER

TURN R CODE INTO WEB APIS

Jeff Allen / @trestleJeff

http://plumber.trestletech.com

EARL 2015
BOSTON 2 - 4 NOVEMBER

YOU'VE GOT R CODE...

```
buildModel <- function(myData){
  mcmcTheKmeansLogarithm(myData)
}
forecastSales <- function(date){
  linearModel(sales, date)
}
```

YOU'VE GOT A WEBSITE...

HatsForCats.info

Jeff

hatsforcats.info

Hats For Cats!

For all your feline headware needs!

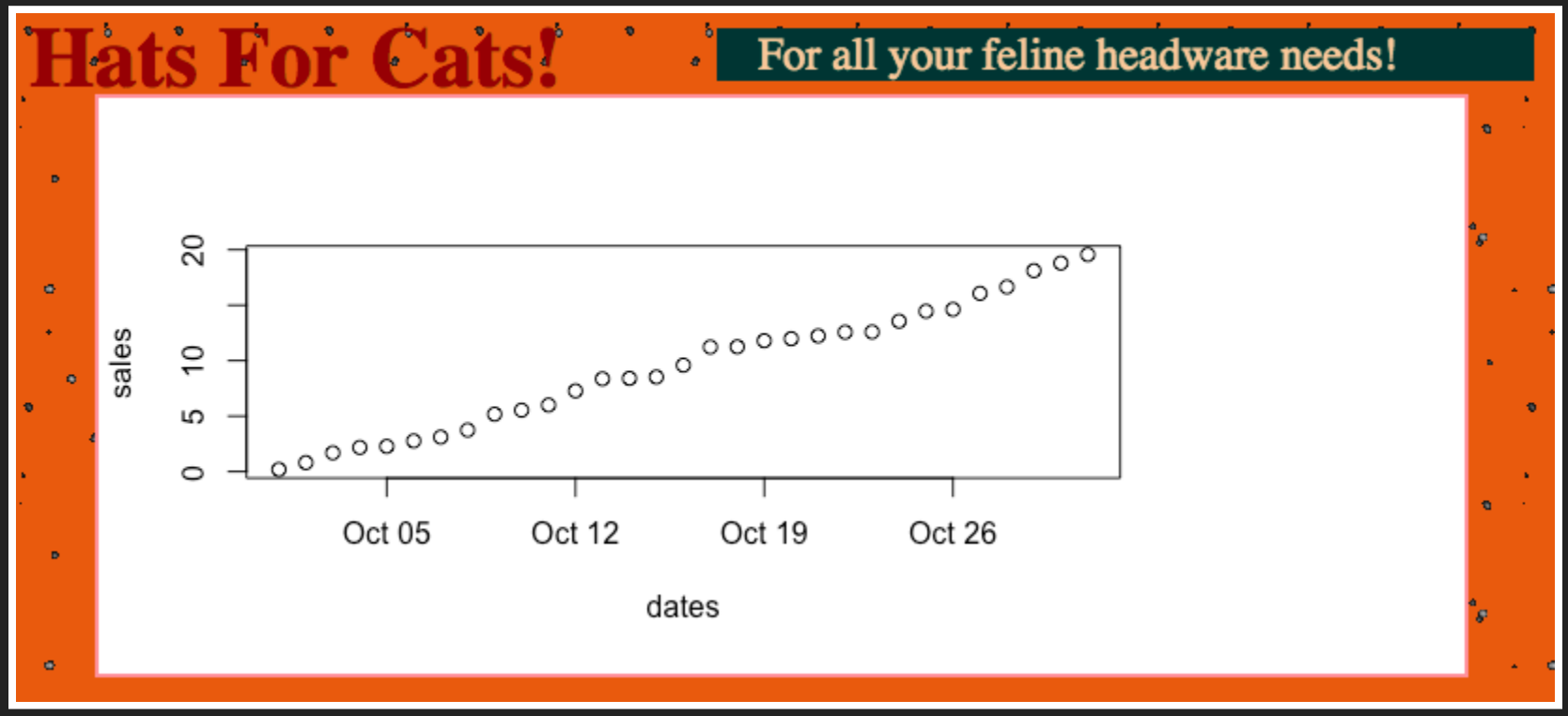
Hatless cats...

=>

Cats... with hats!

Buy now!

YOU WANT TO COMBINE THEM...



PLUMBER CAN HELP!

- Convert existing R code to web APIs
- Just add comments to your existing code
- Leverage R from existing websites and APIs

OUTLINE

- Decorating
- Running
- Routing
- Filtering

OUTLINE

- Decorating
- Running
- Routing
- Filtering

DECORATING

```
makePlot <- function(){
  par(mar = rep(2, 4)) # margins
  dates <- seq(as.Date("2015-10-01"),
      as.Date("2015-10-31"), by=1)
  sold <- 1:31

  plot(dates, sold, type="b", main="Sales")
}
```

DECORATING

```
#* @get /salesgraph
#* @png
makePlot <- function(){
  par(mar = rep(2, 4)) # margins
  dates <- seq(as.Date("2015-10-01"),
      as.Date("2015-10-31"), by=1)
  sold <- 1:31

  plot(dates, sold, type="b", main="Sales")
}
```

DECORATING

```
sales <- NULL
#* @post /addsale
function(qty, item, id){
  sales <<- rbind(sales, data.frame(
    id = id,
    time = Sys.time(),
    item = item,
    qty = qty
  ))
  id
}
```

AUTOMATIC PARSING

```
GET plumber.trestletech.com/sale?qty=2
```

```
#* @get /sale
function(qty){
  qty == "2" # TRUE
}
```

POST FORMS TOO!

```
POST plumber.trestletech.com/ HTTP/1.1

qty=3
```

```
#* @post /
function(qty){
  qty == "3" # TRUE
}
```

YOU ALWAYS GET REQ AND RES

```
#* @get /
function(req, res){
  ip <- req$REMOTE_ADDR
  res$setHeader("Last-Modified",
        "Fri, 16 Oct 2015 12:45:26 GMT")
  # ...
}
```

OUTLINE

- Decorating
- **Running**
- Routing
- Filtering

RUNNING

```
library(plumber)

pr <- plumber::plumb("myfile.R")
pr$run(port=8000)
```

DEMO

- Decorating
- Running
- **Routing**
- Filtering

STATIC ROUTES

```
#* @post /transaction
function(item, qty, id){
  sales <<- rbind(sales, list(time=Sys.time(),
                              item=item, qty=qty, id=id))
}
```

```
POST myplumber.com/transaction HTTP/1.1

item=black-hat&qty=3&id=429
```

VARIABLE ROUTES

```
#* @get /transaction/<id>
function(id){
  id <- as.integer(id)
  sales[sales$id == id, ]
}
```

```
GET myplumber.com/transaction/429 HTTP/1.1

[{"id":429,"time":"2015-10-16 20:52:06",
    "item":"blue-hat","qty":"3"}]
```

TYPED VARIABLE ROUTES

```
#* @get /transaction/<id:int>
function(id){
  sales[sales$id == id, ]
}
```

```
GET myplumber.com/transaction/429 HTTP/1.1

[{"id":429,"time":"2015-10-16 20:52:06",
    "item":"blue-hat","qty":"3"}]
```

COMPLEX ROUTES

```
#* @get /hats/<color>/size/<size:int>
function(color, size){
  hats[hats$color == color
       & hats$size == size]
}
```

OUTLINE

- Decorating
- Running
- Routing
- **Filtering**

ENDPOINTS

[Always serve things]

```
#* @get /transaction/<id>
function(id){
  id <- as.integer(id)
  sales[sales$id == id, ]
}
```

SPECIAL ENDPOINTS

[Serve Images]

```
#* @get /transaction/plot
#* @png
function(id){
  plot(sales$date, sales$qty,
    main="Qty/Purchase Over Time",
    xlab="Date", ylab="Qty")
}
```

Returns a PNG image of your graph

FILTERS

[1. Passively do things]

```
#* @filter logger
function(req){
  print(paste0(date(), " - ",
               req$REMOTE_ADDR, " - ",
               req$REQUEST_METHOD, " ",
               req$PATH_INFO))
  forward()
}
```

FILTERS

[2. Change things on the way by]

```
#* @filter setuser
function(req){
  un <- req$HTTP_COOKIE
  # Make req$username available to endpoints
  req$username <- un

  forward()
}
```

FILTERS

[3. Can serve a response]

```
#* @filter nochrome
function(req){
    if (!grepl("Chrome", req$HTTP_USER_AGENT)){
      forward()
    } else {
      res$status <- 400
      res$body <- "NOT WELCOME HERE!"
    }
}
```

EXTRAS

- Debug your web app on a REPL!
- Static asset server
- Helpers to serve HTML, Markdown, and RMD
- Composable API, if annotations aren't your thing

GO FORWARD AND MAKE YOUR DREAMS COME TRUE!

- Website integration
- Internal analytics APIs
- Webhooks (GitHub, Google Drive, Slack, ...)
- Include a web-accessible demo in your packages
- Complete website hosted in R