

# Stat405

String processing

Hadley Wickham

1. Intro to email

2. String basics

3. Finding & splitting

4. Next time:

Regular expressions

# Motivation

Want to try and classify **spam** vs. **ham** (non-spam) email.

Need to process character vectors containing complete contents of email. Eventually, will create variables that will be useful for classification.

Same techniques **very** useful for data cleaning.

# Getting started

```
load("email.rdata")  
str(contents) # second 100 are spam
```

```
install.packages("stringr")  
# all functions starting with str_  
# come from this package  
help(package = "stringr")
```

Received: from NAHOU-MSMBX07V.corp.enron.com ([192.168.110.98]) by NAHOU-MSAPP01S.corp.enron.com with Microsoft SMTPSVC(5.0.2195.2966);

Mon, 1 Oct 2001 14:39:38 -0500

x-mimeole: Produced By Microsoft Exchange V6.0.4712.0

content-class: urn:content-classes:message

MIME-Version: 1.0

Content-Type: text/plain;

Content-Transfer-Encoding: binary

Subject: MERCATOR ENERGY INCORPORATED and ENERGY WEST INCORPORATED

Date: Mon, 1 Oct 2001 14:39:38 -0500

Message-ID: <053C29CC8315964CB98E1BD5BD48E3080B522E@NAHOU-MSMBX07V.corp.enron.com>

X-MS-Has-Attach:

X-MS-TNEF-Correlator: <053C29CC8315964CB98E1BD5BD48E3080B522E@NAHOU-MSMBX07V.corp.enron.com>

Thread-Topic: MERCATOR ENERGY INCORPORATED and ENERGY WEST INCORPORATED

Thread-Index: AcFKsM5wASRhZ102QuKX13U5Ww741Q==

X-Priority: 1

Priority: Urgent

Importance: high

From: "Landau, Georgi" <Georgi.Landau@ENRON.com>

To: "Bailey, Susan" <Susan.Bailey@ENRON.com>,

"Boyd, Samantha" <Samantha.Boyd@ENRON.com>,

"Heard, Marie" <Marie.Heard@ENRON.com>,

"Jones, Tana" <Tana.Jones@ENRON.com>,

"Panus, Stephanie" <Stephanie.Panus@ENRON.com>

Return-Path: [Georgi.Landau@ENRON.com](mailto:Georgi.Landau@ENRON.com)

Please check your records and let me know if you have any kind of documentation evidencing a merger indicating that Mercator Energy Incorporated merged with and into Energy West Incorporated?

I am unable to find anything to substantiate this information.

Thanks for your help.

Georgi Landau  
Enron Net Works

...

Date: Sun, 11 Nov 2001 11:55:05 -0500  
Message-Id: <[200503101247.j2ACloAq014654@host.high-host.com](mailto:200503101247.j2ACloAq014654@host.high-host.com)>  
To: [benrobinson13@shaw.ca](mailto:benrobinson13@shaw.ca)  
Subject: LETS DO THIS TOGHTHER  
From: ben1 <[ben\\_1\\_wills@yahoo.com.au](mailto:ben_1_wills@yahoo.com.au)>  
X-Priority: 3 (Normal)  
CC:  
Mime-Version: 1.0  
Content-Type: text/plain; charset=us-ascii  
Content-Transfer-Encoding: 7bit  
X-Mailer: RLSP Mailer

Dear Friend,

Firstly, not to cause you embarrassment, I am Barrister Benson Wills, a Solicitor at law and the personal attorney to late Mr. Mark Michelle a National of France, who used to be a private contractor with the Shell Petroleum Development Company in Saudi Arabia, herein after shall be referred to as my client. On the 21st of April 2001, him and his wife with their three children were involved in an auto crash, all occupants of the vehicle unfortunately lost their lives.

Since then, I have made several enquiries with his country's embassies to locate any of my clients extended relatives, this has also proved unsuccessful. After these several unsuccessful attempts, I decided to contact you with this business partnership proposal. I have contacted you to assist in repatriating a huge amount of money left behind by my client before they get confiscated or declared unserviceable by the Finance/Security Company where these huge deposit was lodged.

The deceased had a deposit valued presently at £30million Pounds Sterling and the Company has issued me a notice to provide his next of kin or Beneficiary by Will otherwise have the account confiscated within the next thirty official working days.

Since I have been unsuccessful in locating the relatives for over two years now, I seek your consent to present you as the next of kin/Will Beneficiary to the deceased so that the proceeds of this

# Structure of an email

Headers give metadata

Empty line

Body of email

Other major complication is attachments and alternative content, but we'll ignore those for class

# Tasks

- Split into header and contents
- Split header into fields
- Generate useful variables for distinguishing between spam and non-spam

# String basics

```
# Special characters
```

```
a <- "\\\"
```

```
b <- "\"\"
```

```
c <- "a\nb\nc"
```

```
a
```

```
cat(a, "\n")
```

```
b
```

```
cat(b, "\n")
```

```
c
```

```
cat(c, "\n")
```

# Special characters

- Use `\` to “escape” special characters
  - `\” = ”`
  - `\n = new line`
  - `\\ = \`
  - `\t = tab`
- ?Quotes for more

# Displaying strings

`print()` will display quoted form.

`cat()` will display actual contents (but need to add a newline to the end).

Generally better to use `message()` if you want to send a message to the user of your function

# Your turn

Create a string for each of the following strings:

`:-\`

`(^_^")`

`@_'-'`

`\m/`

Create a multiline string.

Compare the output from `print` and `cat`

```
a <- ":-\\"
b <- "(^_^\")"
c <- "@_-''"
d <- "\\m/"
e <- "This string\ngoes over\nmultiple lines"

a; b; c; d; e
cat(str_join(a, b, c, d, e, "\n", sep = "\n"))
```

**Back to the problem**

# Header vs. content

Need to split the string into two pieces, based on the the **location** of double line break:

```
str_locate(string, pattern)
```

Splitting = creating two substrings, one to the right, one to the left:

```
str_sub(string, start, end)
```

```
str_locate("great")  
str_locate("fantastic", "a")  
str_locate("super", "a")
```

```
superlatives <- c("great", "fantastic", "super")  
res <- str_locate(superlatives, "a")  
str(res)  
str(str_locate_all(superlatives, "a"))
```

```
str_sub("testing", 1, 3)  
str_sub("testing", start = 4)  
str_sub("testing", end = 4)
```

```
input <- c("abc", "defg")  
str_sub(input, c(2, 3))
```

# Your turn

Use `sub_locate()` to identify the location where the double break is (make sure to check a few!)

Split the emails into header and content with `str_sub()`

```
breaks <- str_locate(contents, "\n\n")

# Remove invalid emails
valid <- !is.na(breaks[, "start"])
contents <- contents[valid]
breaks <- breaks[valid, ]

# Extract headers and bodies
header <- str_sub(contents, end = breaks[, 1])
body <- str_sub(contents, start = breaks[, 2])
```

# Headers

- Each header starts at the beginning of a new line
- Each header is composed of a name and contents, separated by a colon

```
h <- header[2]
```

```
# Does this work?
```

```
str_split(h, "\n")[[1]]
```

```
# Why / why not?
```

```
# How could you fix the problem?
```

```
# Split & patch up

lines <- str_split(h, "\n")

continued <- str_sub(lines, 1, 1) %in% c(" ", "\t")

# This is a neat trick!
groups <- cumsum(!continued)

fields <- rep(NA, max(groups))
for (i in seq_along(fields)) {
  fields[i] <- str_join(lines[groups == i],
    collapse = "\n")
}
```

# Your turn

Write a small function that given a single header field splits it into name and contents.

Do you want to use `str_split()`, or `str_locate()` & `str_sub()`?

Remember to get the algorithm working before you write the function

```
test1 <- "Sender: <Lighthouse@independent.org>"
```

```
test2 <- "Subject: Alice: Where is my coffee?"
```

```
f1 <- function(input) {  
  str_split(input, ": ")[[1]]  
}
```

```
f2 <- function(input) {  
  colon <- str_locate(input, ": ")  
  c(  
    str_sub(input, end = colon[, 1] - 1),  
    str_sub(input, start = colon[, 2] + 1)  
  )  
}
```

# Your turn

Write a function that given an email returns a list content the body of the email, and a data frame of headers.

# Next time

We split the content into header and body. And split up the header into fields. Both of these tasks used fixed strings.

What if the pattern we need to match is more complicated?