Tidying Data and String Processing

Prof. Roger D. Peng
Department of Statistics and Data Sciences
University of Texas, Austin

Spring 2024

Text Data

- Data do not always come in the form of numbers
- Some data (like text) need to be converted/transformed into numerical values that can be analyzed
- String / text processing tools in R can be used for this
- Tidyverse package **stringr**

Case Study: Git Log Analysis

- Local Baltimore events organization / ticket sales company contract with another company to revamp their web site

- Ticket sales company paid contractor for programming, design, and implementation

- Ticket sales company felt the contractor did not deliver agreed upon outputs

- Ticket sales company had access to the contractor's git repository and wanted to analyze it for productivity

Case Study: Git Log Analysis

- Git is a software version control system that tracks changes to a code base

- Software developers **commit** their changes to the system and write a brief description of what they changed

- The Git system is a log of all changes to a project

- Git can be used to track many files being changed by many developers

- Essentially the industry standard for software configuration management

What is the Data?

```
commit 6fe5d43383d50c698993c9b46b33f08f4897c70f
Author: ZchMr <zc@.>
Date:   Wed Oct 1 16:14:22 2014 -0400

    tickets; portal; and bo updates
```

Commit identifier
Unique Author
Date/time of commit

Brief summary of changes made

```
diff --git a/workbench/bts/boxoffice/src/views/partials/sidebarTable.blade.php b/workbench/
bts/boxoffice/src/views/partials/sidebarTable.blade.php
index 499afc6..37f18fb 100644
--- a/workbench/bts/boxoffice/src/views/partials/sidebarTable.blade.php
+++ b/workbench/bts/boxoffice/src/views/partials/sidebarTable.blade.php
@@ -21,7 +21,7 @@
                        <td class="text-right">${{money_format('%i', $cart->fees)}}</td>
                </tr>
        </tbody>
-       @if (Route::currentRouteName() != 'get.boxoffice.{boxoffice}.review')
+       @if (Route::currentRouteName() != 'get.boxoffice.{boxoffice}.review' &&
Route::currentRouteName() != 'get.cart.review')
        <thead>
                <tr>
                        <th colspan="2">Discount<!--  <a href="#"><i class="fa fa-question-
circle fa-fw"</a> --></th>
```

Exact code changes made in diff format

How Many Commits?

Total number of lines of text

Count the lines that start with "commit" followed by a space and then some random text

```
> commits
# A tibble: 7,752,495 × 1
   text
   <chr>
 1 "commit ce5b08adc101aac0370800a230cedb24a93be679"
 2 "Merge: 7f6ef08 210649f"
 3 "Author: ZchMr <zc@.>"
 4 "Date:   Wed Oct 1 16:59:21 2014 -0400"
 5 ""
 6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"
 7 "    "
 8 "    Conflicts:"
 9 "    \tworkbench/bts/boxoffice/src/views/review.blade.php"
10 "    \tworkbench/bts/frontend/src/views/checkout/review.blade.php"
# ℹ 7,752,485 more rows
```

Regular Expressions

- A language for describing language
    - 'lines that start with "commit" followed by a space and then some random text'
- A combination of **literals** and **metacharacters**
- Regular expressions define a rich set of metacharacters

Regular Expressions

- The world of regular expressions is very large, but there are few key metacharacters that get used often.

- Literals

- Beginning/End of line

- Character classes

- Repetition

- Parenthesized subexpression

Literals

- Simplest pattern consists only of literals. The literal “nuclear” would match to the following lines:

Why do we trust our government with 8,500 **nuclear** weapons, but not to administer health care?

Beginning to regret that **nuclear** taco

Apple is all about **nuclear** family level network effects.

"I’m going to destroy Android, because it’s a stolen product. I’m willing to go thermo**nuclear** war on this." - Steve Jobs

Beginning/End of Line

- ^ indicates match at the beginning of a line

- $ indicates match at the end of a line

Beginning/End of Line

^The

The lady then put a plump hand out from the bed, and Candide bathed it

The old woman spoke thus to Cunegonde:

The good old man smiled.

The skipper asked ten thousand piastres. Candide did not hesitate.

The Baron's lady weighed about three hundred and fifty pounds, and was

Beginning/End of Line

^i think

i think should text me.

i think i lost my laptop...

i think "hipster shorts" are the ugliest stupidest things ever

i think i’m gonna go for a run after work today.....i think....=\

Beginning/End of Line

the$

arms. My dear Martin, yet once more Pangloss was right: all is for **the**

seen any one so beautiful as I, and that he never so much regretted **the**

the weak execrate the powerful, before whom they cringe; and **the**

plays with her is yet worse; and the play is still worse than **the**

II. What became of Candide among **the**

Character Classes

- Indicated with square brackets [ ]
- [a-z] any single lower case letter
- [A-Z] any single upper case letter
- [a-zA-Z] any single lower or upper case letter
- [0-9] any single digit
- [aL] just the letters a or L

Character Classes

[Tt]he

volumes of **the**ology, you may well imagine that neither I nor any one

black eunuchs and twenty soldiers. **The** Turks killed prodigious numbers

the harbour which could be sent to Buenos Ayres. The person to whom they

Character Classes

^[Ii] am

I am BEYOND irritated right now

i am boycotting the apple store

I am twittering from iPhone

i am going to dream in XML tonight. ugh.

I am so over this. I need food. Mmmm bacon...

Character Classes

^[0-9][a-zA-Z]

7th inning stretch

2nd half soon to begin. OSU did just win something

3am - cant sleep - too hot still.. :(

5ft 7 sent from heaven

1st sign of starvagtion

Match Anything

The . is used to match anything (including nothing)

9.1

Fairbanks, AK, 99712., but its volunteers and employees are scattered

[25] P. 109. Élie-Catherine Fréron was a French critic (1719-1776) who

[26] P. 111. Gabriel Gauchat (1709-1779), French ecclesiastical writer,

Repetition

- The + is used to indicate “repeat the immediately preceding symbol 1 or more times”

- The * is used to indicate “repeat the immediately preceding symbol 0 or more times”

- Curly braces { } can be used to indicate a range of repetition

Repetition

[0-9]+

| Notes [p. **170**]; spelt Robek in the text [p. 53]) have |

Release Date: November **27**, 2006 [EBook #19942]

voluntary death, first printed in **1735**.

Repetition

2[0-9]*

[**27**] P. 112. Nicholas Charles Joseph Trublet (1697-1770) was a French

| Page 172: rougish amended to roguish; crows amended to |

[**26**] P. 111. Gabriel Gauchat (1709-1779), French ecclesiastical writer,

Repetition

[0-9]{4,6}

uninteresting. Achmet III. (_b._ **1673**, _d._ 1739) was dethroned in 1730.

Release Date: November 27, **2006** [EBook #19942]

Christian rites. In **1730** the "honours of sepulture" were refused to

Repetition

he .* good

surprised at what **he heard. Martin found there was a good** deal of reason

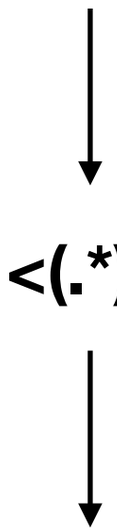
Cacambo, and **he loved his master, because his master was a very good**

wish that **he were here. Certainly, if all things are good**, it is in El

Parenthesized Sub-expression

- Parentheses ( ) can be used to "capture" or extract sub-expressions in a larger string

Roger D. Peng <roger.peng@austin.utexas.edu>



roger.peng@austin.utexas.edu

stringr Functions

- **str_subset()** / **str_detect()** - determine if a string matches a specific regular expression

- **str_extract()** - extract the regular expression matches from a string

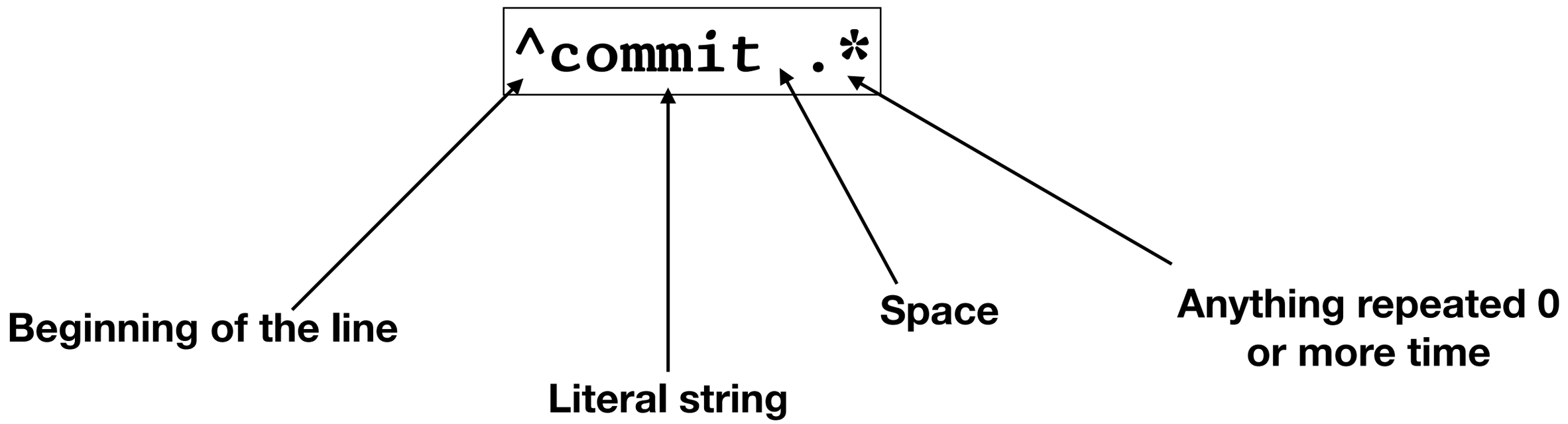
- **str_match()** - extract any parenthesized sub-expressions from a regular expression match

- **str_replace()** - replace a regular expression match with another string

How Many Commits?

```
> commits
# A tibble: 7,752,495 × 1
   text
   <chr>
 1 "commit ce5b08adc101aac0370800a230cedb24a93be679"
 2 "Merge: 7f6ef08 210649f"
 3 "Author: ZchMr <zc@.>"
 4 "Date:   Wed Oct 1 16:59:21 2014 -0400"
 5 ""
 6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"
 7 "    "
 8 "    Conflicts:"
 9 "    \tworkbench/bts/boxoffice/src/views/review.blade.php"
10 "    \tworkbench/bts/frontend/src/views/checkout/review.blade.php"
# ℹ 7,752,485 more rows
```

Count the lines that start with "commit" followed by a space and then some random text



How Many Commits?

```
> commits
# A tibble: 7,752,495 × 1
   text
   <chr>
 1 "commit ce5b08adc101aac0370800a230cedb24a93be679"
 2 "Merge: 7f6ef08 210649f"
 3 "Author: ZchMr <zc@.>"
 4 "Date:   Wed Oct 1 16:59:21 2014 -0400"
 5 ""
 6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"
 7 "    "
 8 "    Conflicts:"
 9 "    \tworkbench/bts/boxoffice/src/views/review.blade.php"
10 "    \tworkbench/bts/frontend/src/views/checkout/review.blade.php"
# ℹ 7,752,485 more rows
```

Count the lines that start with "commit" followed by a space and then some random text

```
^commit .*
```

```
commits |>
    filter(str_detect(text, "^commit .*"))
```

Return the lines of text that match the pattern

```
# A tibble: 3,259 × 1
   text
   <chr>
 1 commit ce5b08adc101aac0370800a230cedb24a93be679
 2 commit 7f6ef08e80191712a5eb0d75c42931466e7bbe73
 3 commit 210649f927d9233131a0b98c8db4d3c2ef9aa26a
 4 commit ccee79f90c2937ea902a26d0d39af3ec03542ad7
 5 commit 6fe5d43383d50c698993c9b46b33f08f4897c70f
```

How Many Commits?

```
commits |>
    filter(str_detect(text, "^commit .*")) |>
    nrow()
```

```
[1] 3259
```

Who Authored the Commits?

```
# A tibble: 6 × 1
  text
  <chr>
1 "commit ce5b08adc101aac0370800a230cedb24a93be679"
2 "Merge: 7f6ef08 210649f"
3 "Author: ZchMr <zc@.>"
4 "Date:   Wed Oct 1 16:59:21 2014 -0400"
5 ""
6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"
```

```
commits |>
    filter(str_detect(text, "^Author:"))
```

```
# A tibble: 3,259 × 1
   text
   <chr>
 1 Author: ZchMr <zc@.>
 2 Author: ZchMr <zc@.>
 3 Author: DvdG <wh@.>
 4 Author: ZchMr <zc@.>
 5 Author: ZchMr <zc@.>
 6 Author: DvdG <wh@.>
 7 Author: DvdG <wh@.>
 8 Author: dvd- <d.@->
 9 Author: DvdG <wh@.>
10 Author: dvd- <d.@->
```

Return lines that begin with the literal "Author:"

Can we extract their email addresses?

Who Authored the Commits?

```
commits |>
    filter(str_detect(text, "^Author:")) |>
    mutate(email = str_extract(text, "<.*>"))
```

Return lines that begin with the literal "Author:"

Extract the part of the line that matches the regular expression

```
# A tibble: 3,259 × 2
   text                  email
   <chr>                 <chr>
 1 Author: ZchMr <zc@.> <zc@.>
 2 Author: ZchMr <zc@.> <zc@.>
 3 Author: DvdG <wh@.>  <wh@.>
 4 Author: ZchMr <zc@.> <zc@.>
 5 Author: ZchMr <zc@.> <zc@.>
 6 Author: DvdG <wh@.>  <wh@.>
 7 Author: DvdG <wh@.>  <wh@.>
 8 Author: dvd- <d.@->  <d.@->
 9 Author: DvdG <wh@.>  <wh@.>
10 Author: dvd- <d.@->  <d.@->
```

```
commits |>
    filter(str_detect(text, "^Author:")) |>
    mutate(email = str_extract(text, "<.*>")) |>
    select(email) |>
    distinct()
```

Reduce a vector down to its unique elements

```
[1] 17
```

Who Authored the Commits?

```
commits |>
    filter(str_detect(text, "^Author:")) |>
    mutate(email = str_extract(text, "<.*>")) |>
    select(email) |>
    distinct()
```

Only keep the unique values of this data frame

```
# A tibble: 17 × 1
   email
   <chr>
 1 <zc@.>
 2 <wh@.>
 3 <d.@->
 4 <a.@->
 5 <sp@.>
 6 <x41l>
 7 <hn@.>
 8 <z@-.>
 9 <z@Z->
10 <R@.2>
11 <R-@.>
12 <R.@->
13 <R@FO>
14 <m@r.>
15 <ml@.>
16 <z@-->
17 <h@..>
```

Author Names

```
# A tibble: 6 × 1
  text
  <chr>
1 "commit ce5b08adc101aac0370800a230cedb24a93be679"
2 "Merge: 7f6ef08 210649f"
3 "Author: ZchMr <zc@.>"
4 "Date:   Wed Oct 1 16:59:21 2014 -0400"
5 ""
6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"
```

```
^Author: ([a-zA-Z]+) <.*>
```

Author Names

str_match() returns a matrix with two columns

```
commits |>
    mutate(author = str_match(text, "^Author: ([a-zA-Z]+) <.*>"))
```

```
# A tibble: 7,752,495 × 2
   text                                                                                                  author[,1]            [,2]
   <chr>                                                                                                 <chr>                 <chr>
 1 "commit ce5b08adc101aac0370800a230cedb24a93be679"                                                     NA                    NA
 2 "Merge: 7f6ef08 210649f"                                                                              NA                    NA
 3 "Author: ZchMr <zc@.>"                                                                                Author: ZchMr <zc@.> ZchMr
 4 "Date:   Wed Oct 1 16:59:21 2014 -0400"                                                               NA                    NA
 5 ""                                                                                                    NA                    NA
 6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"             NA                    NA
 7 "    "                                                                                                NA                    NA
 8 "    Conflicts:"                                                                                      NA                    NA
 9 "    \tworkbench/bts/boxoffice/src/views/review.blade.php"                                            NA                    NA
10 "    \tworkbench/bts/frontend/src/views/checkout/review.blade.php"                                    NA                    NA
# ℹ 7,752,485 more rows
```

Original data

Author Names

```
commits |>
    mutate(author = str_match(text, "^Author: ([a-zA-Z]+) <.*>")) |>
    mutate(author = author[,2])
```

Subset str_match() outpu to just second column

```
# A tibble: 7,752,495 × 2
   text                                                                                                  author
   <chr>                                                                                                 <chr>
 1 "commit ce5b08adc101aac0370800a230cedb24a93be679"                                                     NA
 2 "Merge: 7f6ef08 210649f"                                                                              NA
 3 "Author: ZchMr <zc@.>"                                                                                ZchMr
 4 "Date:   Wed Oct 1 16:59:21 2014 -0400"                                                               NA
 5 ""                                                                                                    NA
 6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"             NA
 7 "    "                                                                                                NA
 8 "    Conflicts:"                                                                                      NA
 9 "    \tworkbench/bts/boxoffice/src/views/review.blade.php"                                            NA
10 "    \tworkbench/bts/frontend/src/views/checkout/review.blade.php"                                    NA
# ℹ 7,752,485 more rows
```

Original data

Author Names

```
commits |>
    mutate(author = str_match(text, "^Author: ([a-zA-Z]+) <.*>")) |>
    mutate(author = author[,2]) |>
    filter(!is.na(author))
```

Filter out NA rows with no author information

```
# A tibble: 3,020 × 2
   text                  author
   <chr>                 <chr>
 1 Author: ZchMr <zc@.> ZchMr
 2 Author: ZchMr <zc@.> ZchMr
 3 Author: DvdG <wh@.>  DvdG
 4 Author: ZchMr <zc@.> ZchMr
 5 Author: ZchMr <zc@.> ZchMr
 6 Author: DvdG <wh@.>  DvdG
 7 Author: DvdG <wh@.>  DvdG
 8 Author: DvdG <wh@.>  DvdG
 9 Author: DvdG <wh@.>  DvdG
10 Author: DvdG <wh@.>  DvdG
# ℹ 3,010 more rows
```

Original data

When Did Commits Occur?

```
# A tibble: 6 × 1
  text
  <chr>
1 "commit ce5b08adc101aac0370800a230cedb24a93be679"
2 "Merge: 7f6ef08 210649f"
3 "Author: ZchMr <zc@.>"
4 "Date:   Wed Oct 1 16:59:21 2014 -0400"
5 ""
6 "    Merge branch 'develop' of https://github.com/btscommercial/missiontix into develop"
```

Only keep rows that start with "Date:"

```
commits |>
    filter(str_detect(text, "^Date:")) |>
    mutate(datetime = str_replace(text, "^Date: +", ""))
```

Replace "Date: " with nothing (empty space)

When Did Commits Occur?

```
commits |>
    filter(str_detect(text, "^Date:")) |>
    mutate(datetime = str_replace(text, "^Date: +", ""))
```

```
# A tibble: 3,259 × 2
   text                                    datetime
   <chr>                                   <chr>
 1 Date:   Wed Oct 1 16:59:21 2014 -0400 Wed Oct 1 16:59:21 2014 -0400
 2 Date:   Wed Oct 1 16:55:12 2014 -0400 Wed Oct 1 16:55:12 2014 -0400
 3 Date:   Wed Oct 1 16:51:19 2014 -0400 Wed Oct 1 16:51:19 2014 -0400
 4 Date:   Wed Oct 1 16:14:58 2014 -0400 Wed Oct 1 16:14:58 2014 -0400
 5 Date:   Wed Oct 1 16:14:22 2014 -0400 Wed Oct 1 16:14:22 2014 -0400
 6 Date:   Wed Oct 1 14:43:11 2014 -0400 Wed Oct 1 14:43:11 2014 -0400
 7 Date:   Wed Oct 1 14:42:44 2014 -0400 Wed Oct 1 14:42:44 2014 -0400
 8 Date:   Wed Oct 1 13:24:22 2014 -0400 Wed Oct 1 13:24:22 2014 -0400
 9 Date:   Wed Oct 1 13:23:31 2014 -0400 Wed Oct 1 13:23:31 2014 -0400
10 Date:   Wed Oct 1 13:14:18 2014 -0400 Wed Oct 1 13:14:18 2014 -0400
# ℹ 3,249 more rows
```

Original data

Dates and Times

When Did Commits Occur?

```
# A tibble: 3,259 × 1
   datetime
   <chr>
 1 Wed Oct 1 16:59:21 2014 -0400
 2 Wed Oct 1 16:55:12 2014 -0400
 3 Wed Oct 1 16:51:19 2014 -0400
 4 Wed Oct 1 16:14:58 2014 -0400
 5 Wed Oct 1 16:14:22 2014 -0400
 6 Wed Oct 1 14:43:11 2014 -0400
 7 Wed Oct 1 14:42:44 2014 -0400
 8 Wed Oct 1 13:24:22 2014 -0400
 9 Wed Oct 1 13:23:31 2014 -0400
10 Wed Oct 1 13:14:18 2014 -0400
```

```
a
Abbreviated weekday name in the current locale. (Also matches full name)

b
Abbreviated or full month name in the current locale.

d
Day of the month as decimal number (01–31 or 0–31

H
Hours as decimal number (00–24 or 0–24)

M
Minute as decimal number (00–59 or 0–59).

S
Second as decimal number (00–61 or 0–61), allowing for up to two leap-seconds

Y
Year with century

z
ISO8601 signed offset in hours and minutes from UTC
```

```
commits |>
    filter(str_detect(text, "^Date:")) |>
    mutate(datetime = str_replace(text, "^Date: +", "")) |>
    select(datetime) |>
    mutate(datetime = parse_date_time(datetime,
                                      orders = "a b d H:M:S Y z",
                                      tz = "America/New_York"))
```

Check ?parse_date_time for format codes

When Did Commits Occur?

```
# A tibble: 3,259 × 1
   datetime
   <chr>
 1 Wed Oct 1 16:59:21 2014 -0400
 2 Wed Oct 1 16:55:12 2014 -0400
 3 Wed Oct 1 16:51:19 2014 -0400
 4 Wed Oct 1 16:14:58 2014 -0400
 5 Wed Oct 1 16:14:22 2014 -0400
 6 Wed Oct 1 14:43:11 2014 -0400
 7 Wed Oct 1 14:42:44 2014 -0400
 8 Wed Oct 1 13:24:22 2014 -0400
 9 Wed Oct 1 13:23:31 2014 -0400
10 Wed Oct 1 13:14:18 2014 -0400
```

```
a
Abbreviated weekday name in the current locale. (Also matches full name)

b
Abbreviated or full month name in the current locale.

d
Day of the month as decimal number (01–31 or 0–31

H
Hours as decimal number (00–24 or 0–24)

M
Minute as decimal number (00–59 or 0–59).

S
Second as decimal number (00–61 or 0–61), allowing for up to two leap-seconds

Y
Year with century

z
ISO8601 signed offset in hours and minutes from UTC
```

```
commits |>
    filter(str_detect(text, "^Date:")) |>
    mutate(datetime = str_replace(text, "^Date: +", "")) |>
    select(datetime) |>
    mutate(datetime = parse_date_time(datetime,
                                      orders = "a b d H:M:S Y z",
                                      tz = "America/New_York"))
```

Check ?parse_date_time for format codes

When Did Commits Occur?

```
commits |>
    filter(str_detect(text, "^Date:")) |>
    mutate(datetime = str_replace(text, "^Date: +", "")) |>
    select(datetime) |>
    mutate(datetime = parse_date_time(datetime,
                                      orders = "a b d H:M:S Y z",
                                      tz = "America/New_York"))
```

```
# A tibble: 3,259 × 1
   date
   <dttm>
 1 2014-10-01 16:59:21
 2 2014-10-01 16:55:12
 3 2014-10-01 16:51:19
 4 2014-10-01 16:14:58
 5 2014-10-01 16:14:22
 6 2014-10-01 14:43:11
 7 2014-10-01 14:42:44
 8 2014-10-01 13:24:22
 9 2014-10-01 13:23:31
10 2014-10-01 13:14:18
```

Column in date/time format

When Did Commits Occur?

```
commits |>
    filter(str_detect(text, "^Date:")) |>
    mutate(datetime = str_replace(text, "^Date: +", "")) |>
    select(datetime) |>
    mutate(datetime = parse_date_time(datetime,
                                      orders = "a b d H:M:S Y z",
                                      tz = "America/New_York")) |>
    ggplot(aes(x = datetime)) +
    geom_histogram(bins = 20) +
    labs(x = "Commit date/time", y = "Count",
         title = "Number of Commits Over Time")
```