Lab 4

Conie O'Malley

2025-02-09

Table of contents

```
packages <- c("rvest", "tm", "readr", "tm.plugin.mail", "Rcrawler", "RSelenium", "xml2", "ti
              "tidytext", "nycflights13")
for (i in packages){
  if (!require(i, character.only = TRUE)) {
    renv::install(i)
  }
  library(i, character.only = TRUE)
}
```

```
Loading required package: rvest

Loading required package: tm

Warning: package 'tm' was built under R version 4.3.3

Loading required package: NLP

Warning: package 'NLP' was built under R version 4.3.3

Loading required package: readr


Attaching package: 'readr'
```

```
The following object is masked from 'package:rvest':

    guess_encoding

Loading required package: tm.plugin.mail

Warning: package 'tm.plugin.mail' was built under R version 4.3.3

Loading required package: Rcrawler

Loading required package: RSelenium

Loading required package: xml2

Loading required package: tidyverse

Warning: package 'lubridate' was built under R version 4.3.3

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v purrr     1.0.2
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x ggplot2::annotate()     masks NLP::annotate()
x dplyr::filter()         masks stats::filter()
x readr::guess_encoding() masks rvest::guess_encoding()
x dplyr::lag()            masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
Loading required package: tidytext

Loading required package: nycflights13
```

```r
library(dplyr)
library(ggplot2)
```

1 Part 1: Primary Data Wrangling Verbs in dplyr and tidyr

```
dplyr::as_tibble(iris) # Convert data frame to tibble
```

```
# A tibble: 150 x 5
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
          <dbl>       <dbl>        <dbl>       <dbl> <fct>
 1          5.1         3.5          1.4         0.2 setosa
 2          4.9         3            1.4         0.2 setosa
 3          4.7         3.2          1.3         0.2 setosa
 4          4.6         3.1          1.5         0.2 setosa
 5          5           3.6          1.4         0.2 setosa
 6          5.4         3.9          1.7         0.4 setosa
 7          4.6         3.4          1.4         0.3 setosa
 8          5           3.4          1.5         0.2 setosa
 9          4.4         2.9          1.4         0.2 setosa
10          4.9         3.1          1.5         0.1 setosa
# i 140 more rows
```

```
dplyr::glimpse(iris) # view data description
```

```
Rows: 150
Columns: 5
$ Sepal.Length <dbl> 5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4, 4.9, 5.4, 4.~
$ Sepal.Width  <dbl> 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9, 3.1, 3.7, 3.~
$ Petal.Length <dbl> 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5, 1.4, 1.5, 1.5, 1.~
$ Petal.Width  <dbl> 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.2, 0.~
$ Species      <fct> setosa, setosa, setosa, setosa, setosa, setosa, setosa, s~
```

1.1 Deliverable 1: Call the iris dataset, and then use the group_by() function to group the iris data by the variable Species, and then use the summarize() function using (avg = mean(Sepal.Width)) in the argument, and then, arrange by average by using the arrange() function with avg in the argument.

```
iris %>%
  dplyr::group_by(Species) %>% # group by species
  dplyr::summarise(avg = mean(Sepal.Width)) %>% # calculate mean by sepal width
  dplyr::arrange(avg) # arrange by average
```

```
# A tibble: 3 x 2
  Species      avg
  <fct>      <dbl>
1 versicolor  2.77
2 virginica   2.97
3 setosa      3.43
```

```
dplyr::filter(iris, Sepal.Length >7) # filter rows where sepal length is greater than 7
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1           7.1         3.0          5.9         2.1 virginica
2           7.6         3.0          6.6         2.1 virginica
3           7.3         2.9          6.3         1.8 virginica
4           7.2         3.6          6.1         2.5 virginica
5           7.7         3.8          6.7         2.2 virginica
6           7.7         2.6          6.9         2.3 virginica
7           7.7         2.8          6.7         2.0 virginica
8           7.2         3.2          6.0         1.8 virginica
9           7.2         3.0          5.8         1.6 virginica
10          7.4         2.8          6.1         1.9 virginica
11          7.9         3.8          6.4         2.0 virginica
12          7.7         3.0          6.1         2.3 virginica
```

```
dplyr::distinct(iris) # view distinct values in iris
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
13           4.8         3.0          1.4         0.1     setosa
14           4.3         3.0          1.1         0.1     setosa
15           5.8         4.0          1.2         0.2     setosa
```

16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa
20	5.1	3.8	1.5	0.3	setosa
21	5.4	3.4	1.7	0.2	setosa
22	5.1	3.7	1.5	0.4	setosa
23	4.6	3.6	1.0	0.2	setosa
24	5.1	3.3	1.7	0.5	setosa
25	4.8	3.4	1.9	0.2	setosa
26	5.0	3.0	1.6	0.2	setosa
27	5.0	3.4	1.6	0.4	setosa
28	5.2	3.5	1.5	0.2	setosa
29	5.2	3.4	1.4	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa
31	4.8	3.1	1.6	0.2	setosa
32	5.4	3.4	1.5	0.4	setosa
33	5.2	4.1	1.5	0.1	setosa
34	5.5	4.2	1.4	0.2	setosa
35	4.9	3.1	1.5	0.2	setosa
36	5.0	3.2	1.2	0.2	setosa
37	5.5	3.5	1.3	0.2	setosa
38	4.9	3.6	1.4	0.1	setosa
39	4.4	3.0	1.3	0.2	setosa
40	5.1	3.4	1.5	0.2	setosa
41	5.0	3.5	1.3	0.3	setosa
42	4.5	2.3	1.3	0.3	setosa
43	4.4	3.2	1.3	0.2	setosa
44	5.0	3.5	1.6	0.6	setosa
45	5.1	3.8	1.9	0.4	setosa
46	4.8	3.0	1.4	0.3	setosa
47	5.1	3.8	1.6	0.2	setosa
48	4.6	3.2	1.4	0.2	setosa
49	5.3	3.7	1.5	0.2	setosa
50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4.0	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
56	5.7	2.8	4.5	1.3	versicolor
57	6.3	3.3	4.7	1.6	versicolor
58	4.9	2.4	3.3	1.0	versicolor

```
59           6.6         2.9          4.6         1.3 versicolor
60           5.2         2.7          3.9         1.4 versicolor
61           5.0         2.0          3.5         1.0 versicolor
62           5.9         3.0          4.2         1.5 versicolor
63           6.0         2.2          4.0         1.0 versicolor
64           6.1         2.9          4.7         1.4 versicolor
65           5.6         2.9          3.6         1.3 versicolor
66           6.7         3.1          4.4         1.4 versicolor
67           5.6         3.0          4.5         1.5 versicolor
68           5.8         2.7          4.1         1.0 versicolor
69           6.2         2.2          4.5         1.5 versicolor
70           5.6         2.5          3.9         1.1 versicolor
71           5.9         3.2          4.8         1.8 versicolor
72           6.1         2.8          4.0         1.3 versicolor
73           6.3         2.5          4.9         1.5 versicolor
74           6.1         2.8          4.7         1.2 versicolor
75           6.4         2.9          4.3         1.3 versicolor
76           6.6         3.0          4.4         1.4 versicolor
77           6.8         2.8          4.8         1.4 versicolor
78           6.7         3.0          5.0         1.7 versicolor
79           6.0         2.9          4.5         1.5 versicolor
80           5.7         2.6          3.5         1.0 versicolor
81           5.5         2.4          3.8         1.1 versicolor
82           5.5         2.4          3.7         1.0 versicolor
83           5.8         2.7          3.9         1.2 versicolor
84           6.0         2.7          5.1         1.6 versicolor
85           5.4         3.0          4.5         1.5 versicolor
86           6.0         3.4          4.5         1.6 versicolor
87           6.7         3.1          4.7         1.5 versicolor
88           6.3         2.3          4.4         1.3 versicolor
89           5.6         3.0          4.1         1.3 versicolor
90           5.5         2.5          4.0         1.3 versicolor
91           5.5         2.6          4.4         1.2 versicolor
92           6.1         3.0          4.6         1.4 versicolor
93           5.8         2.6          4.0         1.2 versicolor
94           5.0         2.3          3.3         1.0 versicolor
95           5.6         2.7          4.2         1.3 versicolor
96           5.7         3.0          4.2         1.2 versicolor
97           5.7         2.9          4.2         1.3 versicolor
98           6.2         2.9          4.3         1.3 versicolor
99           5.1         2.5          3.0         1.1 versicolor
100          5.7         2.8          4.1         1.3 versicolor
101          6.3         3.3          6.0         2.5  virginica
```

102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3.0	5.8	2.2	virginica
106	7.6	3.0	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
111	6.5	3.2	5.1	2.0	virginica
112	6.4	2.7	5.3	1.9	virginica
113	6.8	3.0	5.5	2.1	virginica
114	5.7	2.5	5.0	2.0	virginica
115	5.8	2.8	5.1	2.4	virginica
116	6.4	3.2	5.3	2.3	virginica
117	6.5	3.0	5.5	1.8	virginica
118	7.7	3.8	6.7	2.2	virginica
119	7.7	2.6	6.9	2.3	virginica
120	6.0	2.2	5.0	1.5	virginica
121	6.9	3.2	5.7	2.3	virginica
122	5.6	2.8	4.9	2.0	virginica
123	7.7	2.8	6.7	2.0	virginica
124	6.3	2.7	4.9	1.8	virginica
125	6.7	3.3	5.7	2.1	virginica
126	7.2	3.2	6.0	1.8	virginica
127	6.2	2.8	4.8	1.8	virginica
128	6.1	3.0	4.9	1.8	virginica
129	6.4	2.8	5.6	2.1	virginica
130	7.2	3.0	5.8	1.6	virginica
131	7.4	2.8	6.1	1.9	virginica
132	7.9	3.8	6.4	2.0	virginica
133	6.4	2.8	5.6	2.2	virginica
134	6.3	2.8	5.1	1.5	virginica
135	6.1	2.6	5.6	1.4	virginica
136	7.7	3.0	6.1	2.3	virginica
137	6.3	3.4	5.6	2.4	virginica
138	6.4	3.1	5.5	1.8	virginica
139	6.0	3.0	4.8	1.8	virginica
140	6.9	3.1	5.4	2.1	virginica
141	6.7	3.1	5.6	2.4	virginica
142	6.9	3.1	5.1	2.3	virginica
143	6.8	3.2	5.9	2.3	virginica
144	6.7	3.3	5.7	2.5	virginica

```
145          6.7         3.0          5.2         2.3  virginica
146          6.3         2.5          5.0         1.9  virginica
147          6.5         3.0          5.2         2.0  virginica
148          6.2         3.4          5.4         2.3  virginica
149          5.9         3.0          5.1         1.8  virginica
```

1.2 Deliverable 2: Randomly select a fraction of 0.5 rows from the iris dataset

```
dplyr::sample_frac(iris, 0.5, replace = TRUE) # sample half of the data with replacement
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           5.2         4.1          1.5         0.1     setosa
2           6.7         3.1          4.4         1.4 versicolor
3           6.8         3.2          5.9         2.3  virginica
4           6.5         3.0          5.5         1.8  virginica
5           6.3         2.3          4.4         1.3 versicolor
6           7.3         2.9          6.3         1.8  virginica
7           5.5         2.4          3.8         1.1 versicolor
8           6.5         3.2          5.1         2.0  virginica
9           5.1         3.8          1.6         0.2     setosa
10          5.5         3.5          1.3         0.2     setosa
11          5.1         3.3          1.7         0.5     setosa
12          5.5         2.4          3.7         1.0 versicolor
13          5.8         2.7          4.1         1.0 versicolor
14          5.2         3.4          1.4         0.2     setosa
15          6.4         2.7          5.3         1.9  virginica
16          6.8         2.8          4.8         1.4 versicolor
17          5.4         3.4          1.7         0.2     setosa
18          7.1         3.0          5.9         2.1  virginica
19          6.3         2.3          4.4         1.3 versicolor
20          5.1         3.7          1.5         0.4     setosa
21          5.4         3.4          1.7         0.2     setosa
22          6.3         2.7          4.9         1.8  virginica
23          5.7         2.5          5.0         2.0  virginica
24          6.7         3.1          5.6         2.4  virginica
25          7.2         3.0          5.8         1.6  virginica
26          6.3         2.9          5.6         1.8  virginica
27          7.6         3.0          6.6         2.1  virginica
28          6.3         3.3          6.0         2.5  virginica
29          5.0         2.0          3.5         1.0 versicolor
```

30	5.4	3.4	1.7	0.2	setosa
31	4.3	3.0	1.1	0.1	setosa
32	6.7	3.3	5.7	2.5	virginica
33	7.2	3.0	5.8	1.6	virginica
34	6.9	3.1	5.4	2.1	virginica
35	6.7	2.5	5.8	1.8	virginica
36	6.1	3.0	4.6	1.4	versicolor
37	7.0	3.2	4.7	1.4	versicolor
38	5.2	3.4	1.4	0.2	setosa
39	4.6	3.2	1.4	0.2	setosa
40	5.0	3.2	1.2	0.2	setosa
41	5.1	3.5	1.4	0.2	setosa
42	5.7	2.8	4.1	1.3	versicolor
43	5.1	3.5	1.4	0.2	setosa
44	6.2	2.9	4.3	1.3	versicolor
45	6.0	2.2	5.0	1.5	virginica
46	5.4	3.7	1.5	0.2	setosa
47	5.1	3.5	1.4	0.3	setosa
48	6.7	3.3	5.7	2.1	virginica
49	7.7	2.6	6.9	2.3	virginica
50	4.3	3.0	1.1	0.1	setosa
51	5.8	2.7	3.9	1.2	versicolor
52	5.8	2.6	4.0	1.2	versicolor
53	5.7	2.9	4.2	1.3	versicolor
54	5.7	2.8	4.5	1.3	versicolor
55	6.1	3.0	4.6	1.4	versicolor
56	5.5	2.6	4.4	1.2	versicolor
57	5.0	3.2	1.2	0.2	setosa
58	5.5	2.4	3.8	1.1	versicolor
59	5.7	2.5	5.0	2.0	virginica
60	6.2	2.2	4.5	1.5	versicolor
61	5.7	3.8	1.7	0.3	setosa
62	4.7	3.2	1.6	0.2	setosa
63	5.7	4.4	1.5	0.4	setosa
64	5.5	2.6	4.4	1.2	versicolor
65	6.3	2.8	5.1	1.5	virginica
66	5.0	3.4	1.5	0.2	setosa
67	5.2	3.5	1.5	0.2	setosa
68	4.8	3.4	1.6	0.2	setosa
69	5.8	2.6	4.0	1.2	versicolor
70	4.9	2.4	3.3	1.0	versicolor
71	4.6	3.4	1.4	0.3	setosa
72	4.4	2.9	1.4	0.2	setosa

```
73          5.1         3.7          1.5         0.4     setosa
74          5.4         3.7          1.5         0.2     setosa
75          5.4         3.4          1.5         0.4     setosa
```

```r
dplyr::sample_n(iris, 20, replace = TRUE) # sample 20 random rows of data
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           5.7         3.0          4.2         1.2 versicolor
2           4.9         3.1          1.5         0.1     setosa
3           7.7         3.8          6.7         2.2  virginica
4           6.9         3.1          5.4         2.1  virginica
5           7.2         3.2          6.0         1.8  virginica
6           5.4         3.4          1.7         0.2     setosa
7           6.9         3.1          5.4         2.1  virginica
8           6.5         3.0          5.8         2.2  virginica
9           4.7         3.2          1.6         0.2     setosa
10          6.7         3.3          5.7         2.5  virginica
11          6.5         3.2          5.1         2.0  virginica
12          5.4         3.4          1.7         0.2     setosa
13          4.9         3.1          1.5         0.2     setosa
14          5.7         2.8          4.5         1.3 versicolor
15          5.5         2.4          3.8         1.1 versicolor
16          5.7         2.6          3.5         1.0 versicolor
17          6.4         2.7          5.3         1.9  virginica
18          6.3         2.5          4.9         1.5 versicolor
19          4.8         3.4          1.9         0.2     setosa
20          7.1         3.0          5.9         2.1  virginica
```

```r
dplyr::slice(iris, 20:25) # slice rows from index 20 to 25
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.8          1.5         0.3  setosa
2          5.4         3.4          1.7         0.2  setosa
3          5.1         3.7          1.5         0.4  setosa
4          4.6         3.6          1.0         0.2  setosa
5          5.1         3.3          1.7         0.5  setosa
6          4.8         3.4          1.9         0.2  setosa
```

```r
dplyr::top_n(storms, 20, wind) # get top 20 storms with highest wind speed
```

```
# A tibble: 24 x 13
   name     year month   day  hour   lat  long status    category  wind pressure
   <chr>   <dbl> <dbl> <int> <dbl> <dbl> <dbl> <fct>        <dbl> <int>    <int>
 1 Allen    1980     8     5    12  15.9 -70.5 hurricane        5   155      932
 2 Allen    1980     8     7    12  21   -84.8 hurricane        5   155      910
 3 Allen    1980     8     7    18  21.8 -86.4 hurricane        5   165      899
 4 Allen    1980     8     8     0  22.2 -87.9 hurricane        5   155      920
 5 Allen    1980     8     9     6  25   -94.2 hurricane        5   155      909
 6 Gilbert  1988     9    14     0  19.7 -83.8 hurricane        5   160      888
 7 Gilbert  1988     9    14     6  19.9 -85.3 hurricane        5   155      889
 8 Mitch    1998    10    26    18  16.9 -83.1 hurricane        5   155      905
 9 Mitch    1998    10    27     0  17.2 -83.8 hurricane        5   155      910
10 Rita     2005     9    22     3  24.7 -87.3 hurricane        5   155      895
# i 14 more rows
# i 2 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>
```

1.3 Deliverable 3: Summarize the Data in the iris dataset

```
dplyr::summarize(iris, avg = mean(Petal.Length)) # calculate average petal length
```

```
    avg
1 3.758
```

```
dplyr::mutate_each(iris, funs = mean) #
```

```
Warning: `mutate_each()` was deprecated in dplyr 0.7.0.
i Please use `across()` instead.
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `Species = (function (x, ...) ...`.
Caused by warning in `mean.default()`:
! argument is not numeric or logical: returning NA
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1       5.843333    3.057333        3.758    1.199333      NA
2       5.843333    3.057333        3.758    1.199333      NA
3       5.843333    3.057333        3.758    1.199333      NA
4       5.843333    3.057333        3.758    1.199333      NA
```

```
5       5.843333    3.057333        3.758    1.199333      NA
6       5.843333    3.057333        3.758    1.199333      NA
7       5.843333    3.057333        3.758    1.199333      NA
8       5.843333    3.057333        3.758    1.199333      NA
9       5.843333    3.057333        3.758    1.199333      NA
10      5.843333    3.057333        3.758    1.199333      NA
11      5.843333    3.057333        3.758    1.199333      NA
12      5.843333    3.057333        3.758    1.199333      NA
13      5.843333    3.057333        3.758    1.199333      NA
14      5.843333    3.057333        3.758    1.199333      NA
15      5.843333    3.057333        3.758    1.199333      NA
16      5.843333    3.057333        3.758    1.199333      NA
17      5.843333    3.057333        3.758    1.199333      NA
18      5.843333    3.057333        3.758    1.199333      NA
19      5.843333    3.057333        3.758    1.199333      NA
20      5.843333    3.057333        3.758    1.199333      NA
21      5.843333    3.057333        3.758    1.199333      NA
22      5.843333    3.057333        3.758    1.199333      NA
23      5.843333    3.057333        3.758    1.199333      NA
24      5.843333    3.057333        3.758    1.199333      NA
25      5.843333    3.057333        3.758    1.199333      NA
26      5.843333    3.057333        3.758    1.199333      NA
27      5.843333    3.057333        3.758    1.199333      NA
28      5.843333    3.057333        3.758    1.199333      NA
29      5.843333    3.057333        3.758    1.199333      NA
30      5.843333    3.057333        3.758    1.199333      NA
31      5.843333    3.057333        3.758    1.199333      NA
32      5.843333    3.057333        3.758    1.199333      NA
33      5.843333    3.057333        3.758    1.199333      NA
34      5.843333    3.057333        3.758    1.199333      NA
35      5.843333    3.057333        3.758    1.199333      NA
36      5.843333    3.057333        3.758    1.199333      NA
37      5.843333    3.057333        3.758    1.199333      NA
38      5.843333    3.057333        3.758    1.199333      NA
39      5.843333    3.057333        3.758    1.199333      NA
40      5.843333    3.057333        3.758    1.199333      NA
41      5.843333    3.057333        3.758    1.199333      NA
42      5.843333    3.057333        3.758    1.199333      NA
43      5.843333    3.057333        3.758    1.199333      NA
44      5.843333    3.057333        3.758    1.199333      NA
45      5.843333    3.057333        3.758    1.199333      NA
46      5.843333    3.057333        3.758    1.199333      NA
47      5.843333    3.057333        3.758    1.199333      NA
```

```
48      5.843333    3.057333        3.758    1.199333      NA
49      5.843333    3.057333        3.758    1.199333      NA
50      5.843333    3.057333        3.758    1.199333      NA
51      5.843333    3.057333        3.758    1.199333      NA
52      5.843333    3.057333        3.758    1.199333      NA
53      5.843333    3.057333        3.758    1.199333      NA
54      5.843333    3.057333        3.758    1.199333      NA
55      5.843333    3.057333        3.758    1.199333      NA
56      5.843333    3.057333        3.758    1.199333      NA
57      5.843333    3.057333        3.758    1.199333      NA
58      5.843333    3.057333        3.758    1.199333      NA
59      5.843333    3.057333        3.758    1.199333      NA
60      5.843333    3.057333        3.758    1.199333      NA
61      5.843333    3.057333        3.758    1.199333      NA
62      5.843333    3.057333        3.758    1.199333      NA
63      5.843333    3.057333        3.758    1.199333      NA
64      5.843333    3.057333        3.758    1.199333      NA
65      5.843333    3.057333        3.758    1.199333      NA
66      5.843333    3.057333        3.758    1.199333      NA
67      5.843333    3.057333        3.758    1.199333      NA
68      5.843333    3.057333        3.758    1.199333      NA
69      5.843333    3.057333        3.758    1.199333      NA
70      5.843333    3.057333        3.758    1.199333      NA
71      5.843333    3.057333        3.758    1.199333      NA
72      5.843333    3.057333        3.758    1.199333      NA
73      5.843333    3.057333        3.758    1.199333      NA
74      5.843333    3.057333        3.758    1.199333      NA
75      5.843333    3.057333        3.758    1.199333      NA
76      5.843333    3.057333        3.758    1.199333      NA
77      5.843333    3.057333        3.758    1.199333      NA
78      5.843333    3.057333        3.758    1.199333      NA
79      5.843333    3.057333        3.758    1.199333      NA
80      5.843333    3.057333        3.758    1.199333      NA
81      5.843333    3.057333        3.758    1.199333      NA
82      5.843333    3.057333        3.758    1.199333      NA
83      5.843333    3.057333        3.758    1.199333      NA
84      5.843333    3.057333        3.758    1.199333      NA
85      5.843333    3.057333        3.758    1.199333      NA
86      5.843333    3.057333        3.758    1.199333      NA
87      5.843333    3.057333        3.758    1.199333      NA
88      5.843333    3.057333        3.758    1.199333      NA
89      5.843333    3.057333        3.758    1.199333      NA
90      5.843333    3.057333        3.758    1.199333      NA
```

91	5.843333	3.057333	3.758	1.199333	NA
92	5.843333	3.057333	3.758	1.199333	NA
93	5.843333	3.057333	3.758	1.199333	NA
94	5.843333	3.057333	3.758	1.199333	NA
95	5.843333	3.057333	3.758	1.199333	NA
96	5.843333	3.057333	3.758	1.199333	NA
97	5.843333	3.057333	3.758	1.199333	NA
98	5.843333	3.057333	3.758	1.199333	NA
99	5.843333	3.057333	3.758	1.199333	NA
100	5.843333	3.057333	3.758	1.199333	NA
101	5.843333	3.057333	3.758	1.199333	NA
102	5.843333	3.057333	3.758	1.199333	NA
103	5.843333	3.057333	3.758	1.199333	NA
104	5.843333	3.057333	3.758	1.199333	NA
105	5.843333	3.057333	3.758	1.199333	NA
106	5.843333	3.057333	3.758	1.199333	NA
107	5.843333	3.057333	3.758	1.199333	NA
108	5.843333	3.057333	3.758	1.199333	NA
109	5.843333	3.057333	3.758	1.199333	NA
110	5.843333	3.057333	3.758	1.199333	NA
111	5.843333	3.057333	3.758	1.199333	NA
112	5.843333	3.057333	3.758	1.199333	NA
113	5.843333	3.057333	3.758	1.199333	NA
114	5.843333	3.057333	3.758	1.199333	NA
115	5.843333	3.057333	3.758	1.199333	NA
116	5.843333	3.057333	3.758	1.199333	NA
117	5.843333	3.057333	3.758	1.199333	NA
118	5.843333	3.057333	3.758	1.199333	NA
119	5.843333	3.057333	3.758	1.199333	NA
120	5.843333	3.057333	3.758	1.199333	NA
121	5.843333	3.057333	3.758	1.199333	NA
122	5.843333	3.057333	3.758	1.199333	NA
123	5.843333	3.057333	3.758	1.199333	NA
124	5.843333	3.057333	3.758	1.199333	NA
125	5.843333	3.057333	3.758	1.199333	NA
126	5.843333	3.057333	3.758	1.199333	NA
127	5.843333	3.057333	3.758	1.199333	NA
128	5.843333	3.057333	3.758	1.199333	NA
129	5.843333	3.057333	3.758	1.199333	NA
130	5.843333	3.057333	3.758	1.199333	NA
131	5.843333	3.057333	3.758	1.199333	NA
132	5.843333	3.057333	3.758	1.199333	NA
133	5.843333	3.057333	3.758	1.199333	NA

```
134     5.843333    3.057333        3.758    1.199333      NA
135     5.843333    3.057333        3.758    1.199333      NA
136     5.843333    3.057333        3.758    1.199333      NA
137     5.843333    3.057333        3.758    1.199333      NA
138     5.843333    3.057333        3.758    1.199333      NA
139     5.843333    3.057333        3.758    1.199333      NA
140     5.843333    3.057333        3.758    1.199333      NA
141     5.843333    3.057333        3.758    1.199333      NA
142     5.843333    3.057333        3.758    1.199333      NA
143     5.843333    3.057333        3.758    1.199333      NA
144     5.843333    3.057333        3.758    1.199333      NA
145     5.843333    3.057333        3.758    1.199333      NA
146     5.843333    3.057333        3.758    1.199333      NA
147     5.843333    3.057333        3.758    1.199333      NA
148     5.843333    3.057333        3.758    1.199333      NA
149     5.843333    3.057333        3.758    1.199333      NA
150     5.843333    3.057333        3.758    1.199333      NA
```

```r
dplyr::count(iris, Species, wt = Sepal.Length) # count species based on sepal length
```

```
     Species     n
1     setosa 250.3
2 versicolor 296.8
3  virginica 329.4
```

```r
nycflights13::flights
```

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      544            545        -1     1004           1022
 5  2013     1     1      554            600        -6      812            837
 6  2013     1     1      554            558        -4      740            728
 7  2013     1     1      555            600        -5      913            854
 8  2013     1     1      557            600        -3      709            723
 9  2013     1     1      557            600        -3      838            846
10  2013     1     1      558            600        -2      753            745
# i 336,766 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
dplyr::filter(flights, month == 6, day == 19) # filter flights from June 19
```

```
# A tibble: 985 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     6    19        7           2359         8      355            345
 2  2013     6    19      455            500        -5      639            640
 3  2013     6    19      535            540        -5      800            807
 4  2013     6    19      540            545        -5      920            922
 5  2013     6    19      541            540         1      837            840
 6  2013     6    19      544            548        -4      900            857
 7  2013     6    19      548            545         3      833            819
 8  2013     6    19      552            600        -8      732            712
 9  2013     6    19      553            600        -7      653            725
10  2013     6    19      553            600        -7      708            725
# i 975 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
jan1 <- dplyr::filter(flights, month == 1, day ==1) # filter flights for January 1st
```

1.4 Deliverable 3: Identify Christmas Flights

```
(dec25 <- dplyr::filter(flights, month == 12, day == 25)) # filter for Christmas flights
```

```
# A tibble: 719 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013    12    25      456            500        -4      649            651
 2  2013    12    25      524            515         9      805            814
 3  2013    12    25      542            540         2      832            850
 4  2013    12    25      546            550        -4     1022           1027
 5  2013    12    25      556            600        -4      730            745
 6  2013    12    25      557            600        -3      743            752
 7  2013    12    25      557            600        -3      818            831
 8  2013    12    25      559            600        -1      855            856
 9  2013    12    25      559            600        -1      849            855
10  2013    12    25      600            600         0      850            846
# i 709 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
dplyr::count(dec25) -> dec25_flights # count number of Christmas flights
```

There were 719 flights that departed on December 25th.

```
#dplyr::filter(flights, month = 1)
dplyr::filter(flights, month == 1)
```

```
# A tibble: 27,004 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      544            545        -1     1004           1022
 5  2013     1     1      554            600        -6      812            837
 6  2013     1     1      554            558        -4      740            728
 7  2013     1     1      555            600        -5      913            854
 8  2013     1     1      557            600        -3      709            723
 9  2013     1     1      557            600        -3      838            846
10  2013     1     1      558            600        -2      753            745
# i 26,994 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

The error occurs with the use of a single '=' sign, which tells R that you want to assign month the value of 1, which cannot be done in this case with the filter function where you are trying to identify month values of 1 (January). The correct operator for equality is '=='.

```
dplyr::filter(flights, month == 11 | month == 12) # filter for flights in November or Decembe
```

```
# A tibble: 55,403 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013    11     1        5           2359         6      352            345
 2  2013    11     1       35           2250       105      123           2356
 3  2013    11     1      455            500        -5      641            651
 4  2013    11     1      539            545        -6      856            827
 5  2013    11     1      542            545        -3      831            855
 6  2013    11     1      549            600       -11      912            923
 7  2013    11     1      550            600       -10      705            659
 8  2013    11     1      554            600        -6      659            701
 9  2013    11     1      554            600        -6      826            827
10  2013    11     1      554            600        -6      749            751
# i 55,393 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
count(dplyr::filter(flights, month == 11 | month == 12)) -> nov_dec_flights # count number o
```

There were 55403 flights that departed in November or December.

```
nov_dec <- dplyr::filter(flights, month %in% c(11,12)) # filter for flights in November or De
if (nov_dec_flights == count(nov_dec)){ # use if statement to check if the outputs are equiva
  print("These flights are the same!")
}
```

```
[1] "These flights are the same!"
```

```
dplyr::arrange(flights, year, month, day) # sort by year, month, and day
```

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      544            545        -1     1004           1022
 5  2013     1     1      554            600        -6      812            837
 6  2013     1     1      554            558        -4      740            728
 7  2013     1     1      555            600        -5      913            854
 8  2013     1     1      557            600        -3      709            723
 9  2013     1     1      557            600        -3      838            846
10  2013     1     1      558            600        -2      753            745
# i 336,766 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
dplyr::arrange(flights, desc(arr_delay)) # sort by arrival delay in descending order
```

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     9      641            900      1301     1242           1530
 2  2013     6    15     1432           1935      1137     1607           2120
 3  2013     1    10     1121           1635      1126     1239           1810
 4  2013     9    20     1139           1845      1014     1457           2210
 5  2013     7    22      845           1600      1005     1044           1815
 6  2013     4    10     1100           1900       960     1342           2211
 7  2013     3    17     2321            810       911      135           1020
 8  2013     7    22     2257            759       898      121           1026
 9  2013    12     5      756           1700       896     1058           2020
10  2013     5     3     1133           2055       878     1250           2215
# i 336,766 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
dplyr::select(flights, year, month, day) # select only the year, month, and day columns
```

```
# A tibble: 336,776 x 3
    year month   day
   <int> <int> <int>
 1  2013     1     1
 2  2013     1     1
 3  2013     1     1
 4  2013     1     1
 5  2013     1     1
 6  2013     1     1
 7  2013     1     1
 8  2013     1     1
 9  2013     1     1
10  2013     1     1
# i 336,766 more rows
```

```r
dplyr::select(flights, year:day) # select from the year column to the day column
```

```
# A tibble: 336,776 x 3
    year month   day
   <int> <int> <int>
 1  2013     1     1
 2  2013     1     1
 3  2013     1     1
 4  2013     1     1
 5  2013     1     1
 6  2013     1     1
 7  2013     1     1
 8  2013     1     1
 9  2013     1     1
10  2013     1     1
# i 336,766 more rows
```

```r
dplyr::select(flights, -(year:day)) # remove the year, month, and day columns
```

```
# A tibble: 336,776 x 16
   dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier
      <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>
 1      517            515         2      830            819        11 UA
 2      533            529         4      850            830        20 UA
 3      542            540         2      923            850        33 AA
 4      544            545        -1     1004           1022       -18 B6
 5      554            600        -6      812            837       -25 DL
 6      554            558        -4      740            728        12 UA
 7      555            600        -5      913            854        19 B6
 8      557            600        -3      709            723       -14 EV
 9      557            600        -3      838            846        -8 B6
10      558            600        -2      753            745         8 AA
# i 336,766 more rows
# i 9 more variables: flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
#   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```r
dplyr::rename(flights, tail_num = tailnum) # rename tail_num column to tailnum
```

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      544            545        -1     1004           1022
 5  2013     1     1      554            600        -6      812            837
 6  2013     1     1      554            558        -4      740            728
 7  2013     1     1      555            600        -5      913            854
 8  2013     1     1      557            600        -3      709            723
 9  2013     1     1      557            600        -3      838            846
10  2013     1     1      558            600        -2      753            745
# i 336,766 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tail_num <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

1.5 Deliverable 4: Use the mutate()

```r
flights_sml <- dplyr::select(flights, year:day, tidyselect::ends_with("delay"),distance, air_

dplyr::mutate(flights_sml, gain = arr_delay - dep_delay, speed = distance/air_time*60) # add
```

```
# A tibble: 336,776 x 9
    year month   day dep_delay arr_delay distance air_time  gain speed
   <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
 1  2013     1     1         2        11     1400      227     9  370.
 2  2013     1     1         4        20     1416      227    16  374.
 3  2013     1     1         2        33     1089      160    31  408.
 4  2013     1     1        -1       -18     1576      183   -17  517.
 5  2013     1     1        -6       -25      762      116   -19  394.
 6  2013     1     1        -4        12      719      150    16  288.
 7  2013     1     1        -5        19     1065      158    24  404.
 8  2013     1     1        -3       -14      229       53   -11  259.
 9  2013     1     1        -3        -8      944      140    -5  405.
10  2013     1     1        -2         8      733      138    10  319.
# i 336,766 more rows
```

```
dplyr::mutate(flights_sml, gain = arr_delay - dep_delay, hours = air_time/60, gain_per_hour =
```

```
# A tibble: 336,776 x 10
    year month   day dep_delay arr_delay distance air_time  gain hours
   <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
 1  2013     1     1         2        11     1400      227     9 3.78
 2  2013     1     1         4        20     1416      227    16 3.78
 3  2013     1     1         2        33     1089      160    31 2.67
 4  2013     1     1        -1       -18     1576      183   -17 3.05
 5  2013     1     1        -6       -25      762      116   -19 1.93
 6  2013     1     1        -4        12      719      150    16 2.5
 7  2013     1     1        -5        19     1065      158    24 2.63
 8  2013     1     1        -3       -14      229       53   -11 0.883
 9  2013     1     1        -3        -8      944      140    -5 2.33
10  2013     1     1        -2         8      733      138    10 2.3
# i 336,766 more rows
# i 1 more variable: gain_per_hour <dbl>
```

```
dplyr::transmute(flights, gain = arr_delay - dep_delay, hours = air_time/60, gain_per_hour =
```

```
# A tibble: 336,776 x 3
    gain hours gain_per_hour
   <dbl> <dbl>         <dbl>
 1     9 3.78           2.38
 2    16 3.78           4.23
 3    31 2.67          11.6
 4   -17 3.05          -5.57
 5   -19 1.93          -9.83
 6    16 2.5            6.4
 7    24 2.63           9.11
 8   -11 0.883        -12.5
 9    -5 2.33          -2.14
10    10 2.3            4.35
# i 336,766 more rows
```

```
dplyr::summarize(flights, delay = mean(dep_delay, na.rm=TRUE)) # calculate the average depar
```

```
# A tibble: 1 x 1
  delay
  <dbl>
1  12.6
```

```r
by_day <- dplyr::group_by(flights, year, month, day) # assign flights to days
dplyr::summarize(by_day, delay = mean(dep_delay, na.rm=TRUE)) # calculate delay by day
```

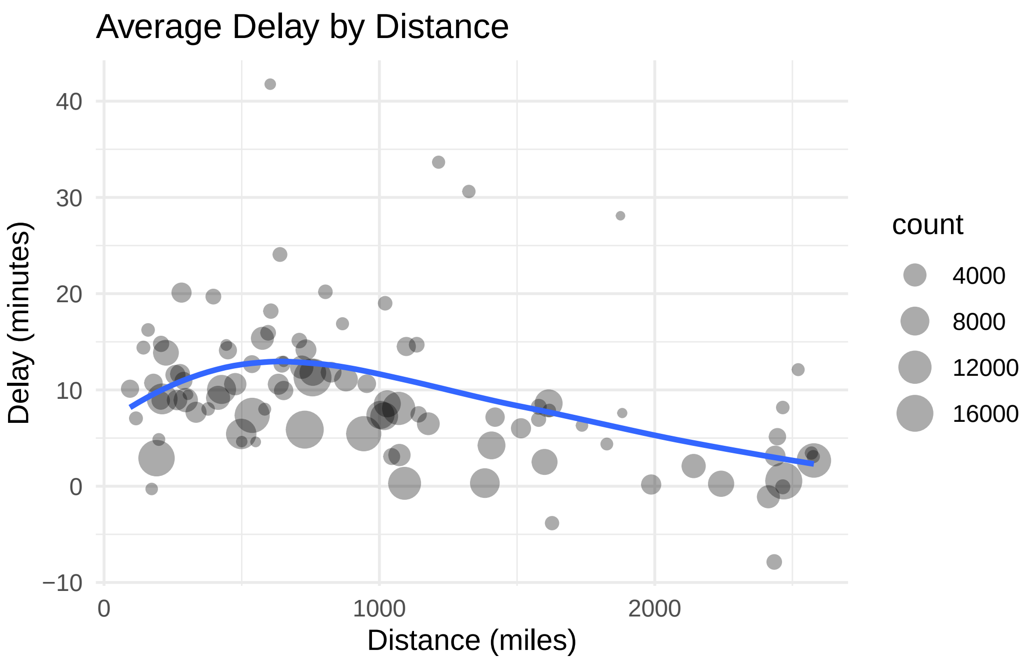
```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
# A tibble: 365 x 4
# Groups:   year, month [12]
    year month   day delay
   <int> <int> <int> <dbl>
 1  2013     1     1 11.5 
 2  2013     1     2 13.9 
 3  2013     1     3 11.0 
 4  2013     1     4  8.95
 5  2013     1     5  5.73
 6  2013     1     6  7.15
 7  2013     1     7  5.42
 8  2013     1     8  2.55
 9  2013     1     9  2.28
10  2013     1    10  2.84
# i 355 more rows
```

```r
by_dest <- dplyr::group_by(flights, dest)

delay <- dplyr::summarize(by_dest, count=n(), dist=mean(distance, na.rm=TRUE),
         delay=mean(arr_delay, na.rm=TRUE)) # calculate average distance and delay by destina
delay <- filter(delay, count >20, dest != "HNL") # remove Honolulu and flights with less than
ggplot2::ggplot(data = delay, mapping = aes(x=dist, y=delay))+
  geom_point(aes(size=count), alpha = 1/3)+
  geom_smooth(se=FALSE) +
  theme_minimal() +
  labs(title="Average Delay by Distance", x="Distance (miles)",
  y="Delay (minutes)") # plot average delay by distance
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



1.6 Deliverable 5: Use the pipe operator to create an object called delays which 1. Groups flights by destination; 2. Summarizes and computes distance, average delay, and number of flights; and 3. Filter to remove noisy points and Honolulu airport.

```
delays <- flights %>%  # create delays object
  dplyr::group_by(dest) %>%
  dplyr::summarize(count=n(), dist=mean(distance, na.rm=TRUE), delay=mean(arr_delay, na.rm=T
  dplyr::filter(count > 20, dest != "HNL")

daily <- dplyr::group_by(flights, year, month, day)
(per_day <- dplyr::summarize(daily, flights=n()))
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
# A tibble: 365 x 4
# Groups:   year, month [12]
    year month   day flights
   <int> <int> <int>   <int>
```

```
 1  2013     1     1     842
 2  2013     1     2     943
 3  2013     1     3     914
 4  2013     1     4     915
 5  2013     1     5     720
 6  2013     1     6     832
 7  2013     1     7     933
 8  2013     1     8     899
 9  2013     1     9     902
10  2013     1    10     932
# i 355 more rows
```

```
daily %>%
  dplyr::ungroup() %>%
  dplyr::summarize(flights=n()) # count the number of total flights
```

```
# A tibble: 1 x 1
  flights
    <int>
1  336776
```

2 Part 2: Handling Missing Values with dplyr

2.1 Deliverable 6: Practicing group_by

```
flights %>%
  dplyr::group_by(year, month, day) %>% # group by date
  dplyr::summarize(mean=mean(dep_delay)) # calculate mean departure delay
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
# A tibble: 365 x 4
# Groups:   year, month [12]
    year month   day  mean
   <int> <int> <int> <dbl>
 1  2013     1     1    NA
 2  2013     1     2    NA
 3  2013     1     3    NA
 4  2013     1     4    NA
 5  2013     1     5    NA
 6  2013     1     6    NA
 7  2013     1     7    NA
 8  2013     1     8    NA
 9  2013     1     9    NA
10  2013     1    10    NA
# i 355 more rows
```

```
flights %>%
  dplyr::group_by(year, month, day) %>%
  dplyr::summarize(mean=mean(dep_delay, na.rm=TRUE)) # calculate departure delay after removi
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
# A tibble: 365 x 4
# Groups:   year, month [12]
    year month   day  mean
   <int> <int> <int> <dbl>
 1  2013     1     1 11.5
 2  2013     1     2 13.9
 3  2013     1     3 11.0
 4  2013     1     4  8.95
 5  2013     1     5  5.73
 6  2013     1     6  7.15
 7  2013     1     7  5.42
 8  2013     1     8  2.55
 9  2013     1     9  2.28
10  2013     1    10  2.84
# i 355 more rows
```

3 Part 3: Practicing Data Wrangling on Real Text Mining Projects

```
impeachtidy <- readr::read_tsv("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001
```

```
Rows: 10987 Columns: 5
```

```
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr  (4): SPEAKER, MAIN SPEAKER, ROLE, TEXT
date (1): HEARING

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

3.1 Deliverable 7: Tokenize the impeachtidy dataset using the unnest_tokens() function on the "TEXT" variable/column to separate the text, so that it has one token per row and store that output in a new object called impeach_words.

```
impeach_words <- impeachtidy %>%
  tidytext::unnest_tokens(word,TEXT) # tokenize the "TEXT" column into individual words
impeach_words
```

```
# A tibble: 376,436 x 5
   HEARING    SPEAKER     `MAIN SPEAKER` ROLE     word
   <date>     <chr>       <chr>          <chr>    <chr>
 1 2019-11-20 Adam Schiff D-Schiff       Democrat your
 2 2019-11-20 Adam Schiff D-Schiff       Democrat interest
 3 2019-11-20 Adam Schiff D-Schiff       Democrat in
 4 2019-11-20 Adam Schiff D-Schiff       Democrat being
 5 2019-11-20 Adam Schiff D-Schiff       Democrat here
 6 2019-11-20 Adam Schiff D-Schiff       Democrat in
 7 2019-11-20 Adam Schiff D-Schiff       Democrat turn
 8 2019-11-20 Adam Schiff D-Schiff       Democrat we
 9 2019-11-20 Adam Schiff D-Schiff       Democrat ask
10 2019-11-20 Adam Schiff D-Schiff       Democrat for
# i 376,426 more rows
```

```
data(stop_words) # load stop words data
head(stop_words) # view first and last few rows of stop words data
```

```
# A tibble: 6 x 2
  word      lexicon
  <chr>     <chr>
1 a         SMART
```

```
2 a's       SMART
3 able      SMART
4 about     SMART
5 above     SMART
6 according SMART
```

```
tail(stop_words)
```

```
# A tibble: 6 x 2
  word     lexicon
  <chr>    <chr>
1 you      onix
2 young    onix
3 younger  onix
4 youngest onix
5 your     onix
6 yours    onix
```

3.2 Deliverable 8: Apply the built-in stopwords dictionary to our impeach_words dataset using the anti_join() function. Use the pipe capabilities %>% of the tidyverse.

```
impeach_clean <- impeach_words %>%
  dplyr::anti_join(stop_words) # remove stop words from the "word" column
```

```
Joining with `by = join_by(word)`
```

```
impeach_clean
```

```
# A tibble: 133,884 x 5
   HEARING    SPEAKER     `MAIN SPEAKER` ROLE     word
   <date>     <chr>       <chr>          <chr>    <chr>
 1 2019-11-20 Adam Schiff D-Schiff       Democrat respect
 2 2019-11-20 Adam Schiff D-Schiff       Democrat proceed
 3 2019-11-20 Adam Schiff D-Schiff       Democrat hearing
 4 2019-11-20 Adam Schiff D-Schiff       Democrat intention
 5 2019-11-20 Adam Schiff D-Schiff       Democrat committee
 6 2019-11-20 Adam Schiff D-Schiff       Democrat proceed
 7 2019-11-20 Adam Schiff D-Schiff       Democrat disruptions
 8 2019-11-20 Adam Schiff D-Schiff       Democrat chairman
 9 2019-11-20 Adam Schiff D-Schiff       Democrat ll
10 2019-11-20 Adam Schiff D-Schiff       Democrat steps
# i 133,874 more rows
```

3.3 Deliverable 9: Count the most frequently occurring words in the dataset.

```
impeach_clean %>%
  dplyr::count(word, sort = TRUE) # count occurrences of each word and sort by frequency
```

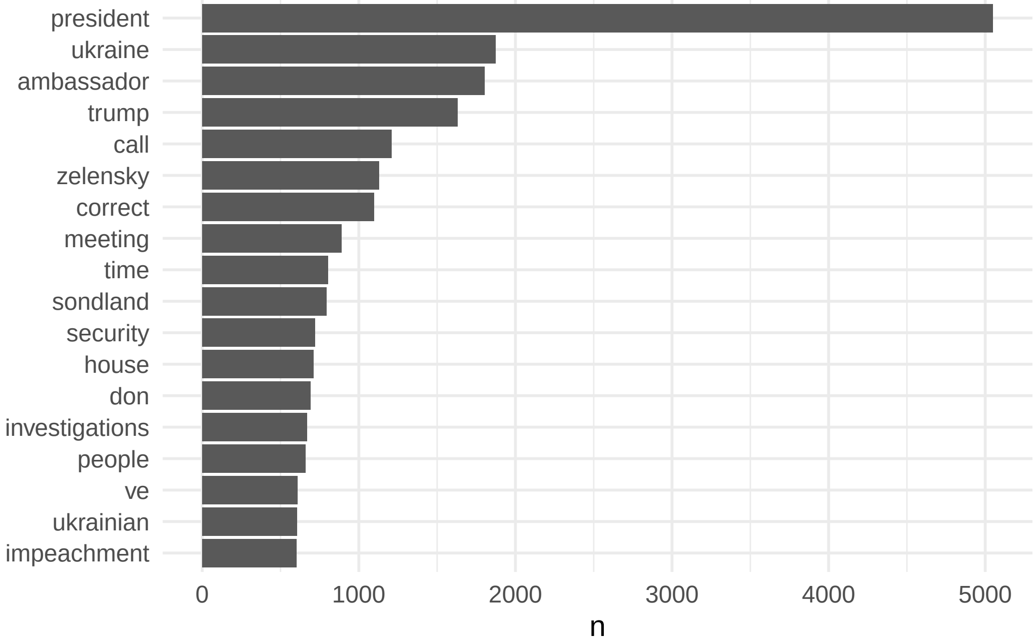
```
# A tibble: 9,176 x 2
   word           n
   <chr>      <int>
 1 president   5049
 2 ukraine     1872
 3 ambassador  1802
 4 trump       1632
 5 call        1210
 6 zelensky    1130
 7 correct     1096
 8 meeting      889
 9 time         805
10 sondland     795
# i 9,166 more rows
```

```
top_10 <- head(impeach_clean, 10)
```

The top 10 words in order are: c(18220, 18220, 18220, 18220, 18220, 18220, 18220, 18220, 18220, 18220), c(“Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”, “Adam Schiff”), c(“D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”, “D-Schiff”), c(“Democrat”, “Democrat”, “Democrat”, “Democrat”, “Democrat”, “Democrat”, “Democrat”, “Democrat”, “Democrat”, “Democrat”), c(“respect”, “proceed”, “hearing”, “intention”, “committee”, “proceed”, “disruptions”, “chairman”, “ll”, “steps”).

3.4 Deliverable 10: Visualize this count using the ggplot2 package. Create a barchart of all the words occurring more than 600 times in the dataset (you could adjust that by changing the filter() parameter).

```
impeach_clean %>%
  dplyr::count(word, sort = TRUE) %>%
  dplyr::filter(n>600) %>%
  dplyr::mutate(word=reorder(word,n)) %>%
  ggplot2::ggplot(aes(word,n)) +
  ggplot2::geom_col() +
  ggplot2::xlab(NULL) +
  ggplot2::coord_flip() +
  ggplot2::theme_minimal()
```



3.5 Deliverable 11: Combinining all the steps using the pipe capabilities of dplyr.

```
impeachtidy <- readr::read_tsv("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001
```

```
Rows: 10987 Columns: 5
```

```
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr  (4): SPEAKER, MAIN SPEAKER, ROLE, TEXT
date (1): HEARING

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

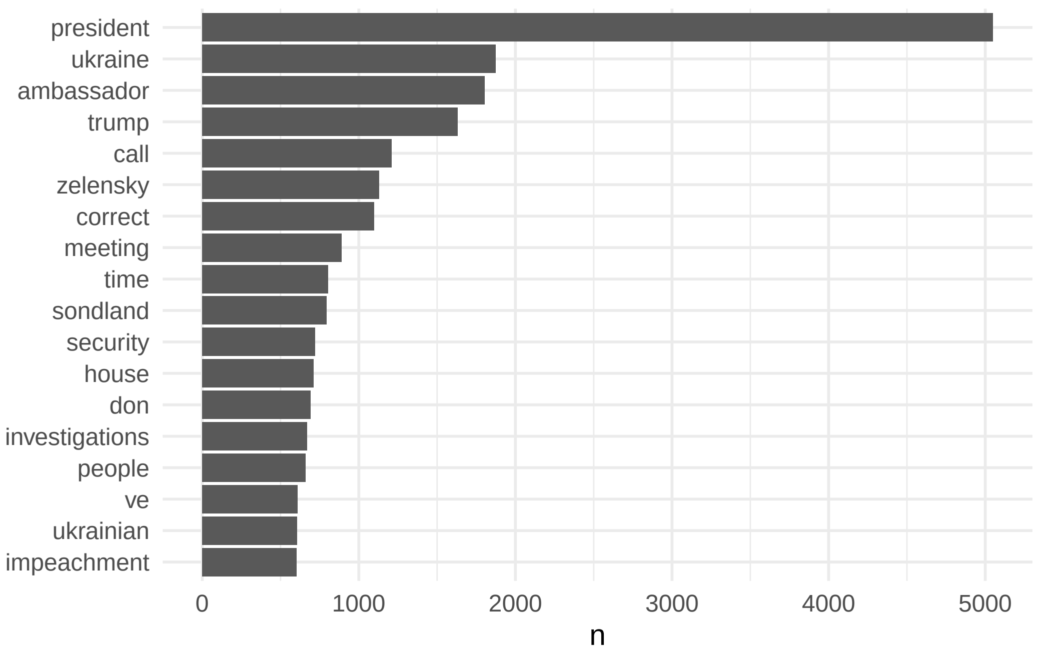
```r
impeach_words <- impeachtidy %>%
  tidytext::unnest_tokens(word,TEXT) %>%
  dplyr::anti_join(stop_words) # tokenize words and remove stop words
```

```
Joining with `by = join_by(word)`
```

```r
impeach_clean <- impeach_words %>%
  dplyr::anti_join(stop_words) # remove stop words again
```

```
Joining with `by = join_by(word)`
```

```r
impeach_clean %>%  # visualize the words used in the impeachment
  dplyr::count(word, sort = TRUE) %>%
  dplyr::filter(n>600) %>%
  dplyr::mutate(word=reorder(word,n)) %>%
  ggplot2::ggplot(aes(word,n)) +
  ggplot2::geom_col() +
  ggplot2::xlab(NULL) +
  ggplot2::coord_flip() +
  ggplot2::theme_minimal()
```



```r
impeach_words <- impeachtidy %>%
  tidytext::unnest_tokens(word,TEXT) %>%
  dplyr::count(SPEAKER, word, sort=TRUE) %>%
  dplyr::ungroup()
impeach_words # count the impeach words by speaker and word
```

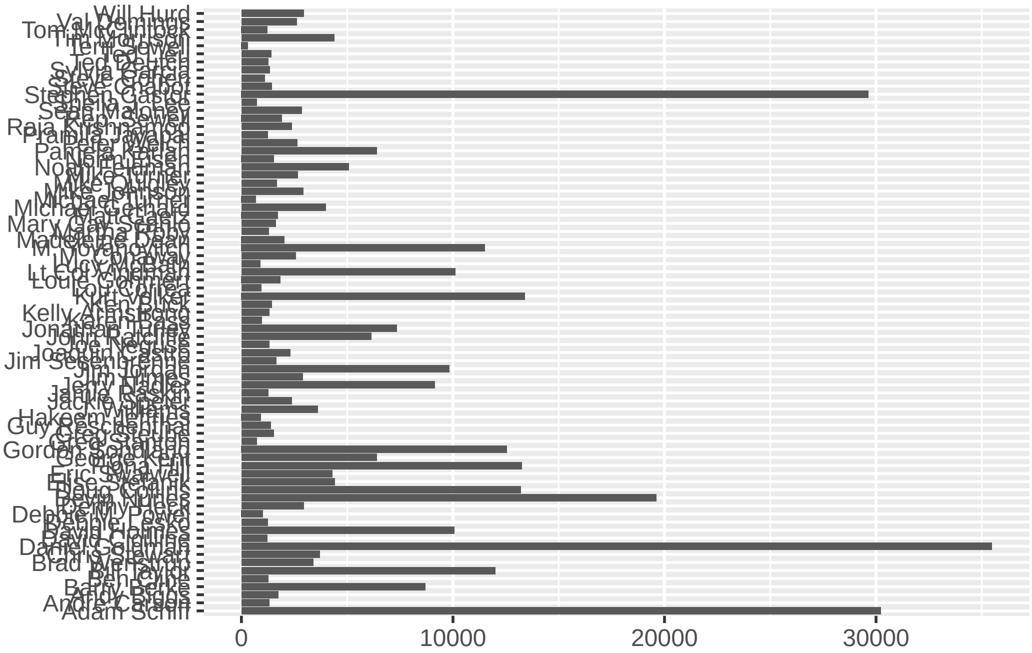
```
# A tibble: 64,655 x 3
   SPEAKER         word      n
   <chr>           <chr> <int>
 1 Daniel Goldman  the    1890
 2 Adam Schiff     the    1831
 3 Stephen Castor  the    1823
 4 Daniel Goldman  that   1603
 5 Devin Nunes     the    1219
 6 Daniel Goldman  to     1202
 7 Daniel Goldman  you    1127
 8 Daniel Goldman  and    1043
 9 Adam Schiff     to     1008
10 Adam Schiff     that    862
# i 64,645 more rows
```

3.6 Deliverable 12: Group by speaker then explore the object and visualize the results.

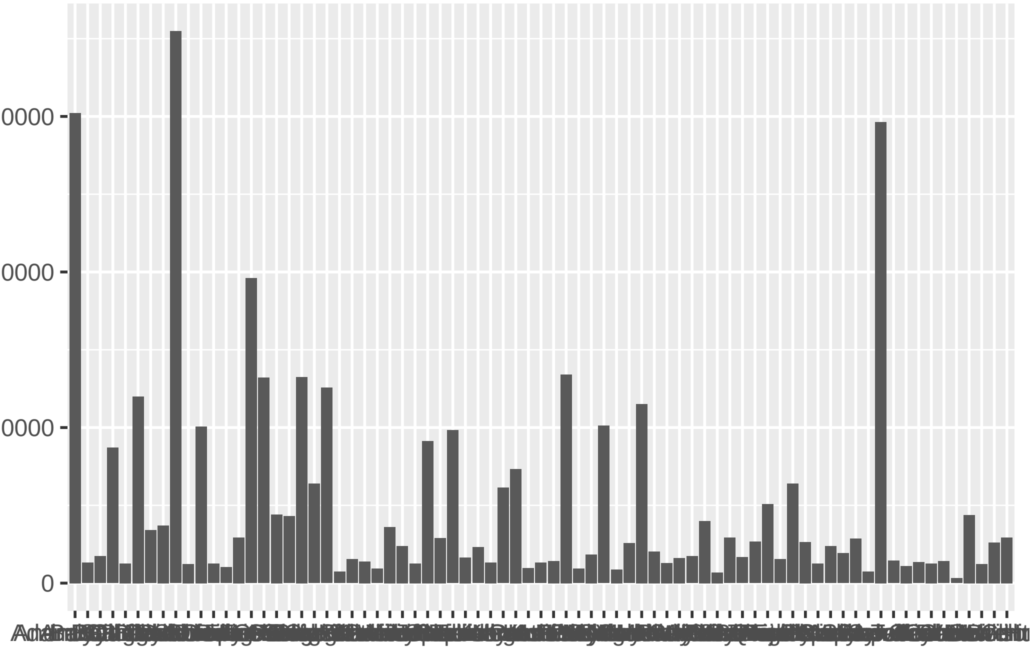
```
total_impeach <- impeach_words %>%
  dplyr::group_by(SPEAKER) %>%
  dplyr::summarize(total=sum(n)) %>%
  dplyr::arrange(desc(total)) # count words by speaker
total_impeach
```

```
# A tibble: 75 x 2
   SPEAKER         total
   <chr>           <int>
 1 Daniel Goldman  35478
 2 Adam Schiff     30222
 3 Stephen Castor  29646
 4 Devin Nunes     19602
 5 Kurt Volker     13404
 6 Fiona Hill      13245
 7 Doug Collins    13197
 8 Gordon Sondland 12558
 9 Bill Taylor     11998
10 M. Yovanovitch  11513
# i 65 more rows
```

```
total_impeach %>%
  ggplot2::ggplot(aes(SPEAKER,total)) +
  ggplot2::geom_col() +
  ggplot2::xlab(NULL) +
  ggplot2::ylab(NULL) +
  ggplot2::coord_flip() # visualize word count by speaker totals
```



```r
total_impeach %>%
  ggplot2::ggplot(aes(SPEAKER,total)) +
  ggplot2::geom_col() +
  ggplot2::xlab(NULL) +
  ggplot2::ylab(NULL) # view without flipping the coordinates
```



3.7 Deliverable 13: Exploring .txt files using tm package

```
igfbali <- tm::Corpus(tm::DirSource("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal
                      readerControl=list(reader=tm::readPlain)) # read igfbali data into corp
class(igfbali) # review the class
```

```
[1] "SimpleCorpus" "Corpus"
```

```
igfbali
```

```
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 63
```

3.8 Deliverable 14: Pre-processing the igfbali corpus

```r
igfbali <- tm::tm_map(igfbali, removeWords, stopwords("english")) # remove stopwords

more.stop.words <- c("transcript", "transcripts") # add more stop words

igfbali <- tm::tm_map(igfbali, removeWords, more.stop.words) # remove more stop words

tm::tm_map(igfbali, stemDocument) # stem document
```

```
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 63
```

3.9 Deliverable 15: Create a Document Term Matrix (DTM) of the igfbali corpus.

```r
dtm <- tm::DocumentTermMatrix(igfbali) # create a document term matrix
```

3.10 Deliverable 16: Exploring the Document Term Matrix (DTM)

```r
tm::findFreqTerms(dtm, 500) # find terms with frequency greater than or equal to 500
```

```
 [1] ". "              "actually"        "also"            "and"
 [5] "around"          "back"            "big"             "can"
 [9] "come"            "countries"       "data"            "different"
[13] "even"            "first"           "give"            "going"
[17] "good"            "governance"      "government"      "i'm"
[21] "igf"             "important"       "information"     "internet"
[25] "issues"          "just"            "kind"            "know"
[29] "know,"           "last"            "like"            "look"
[33] "lot"             "make"            "many"            "may"
[37] "maybe"           "much"            "need"            "new"
[41] "now"             "one"             "part"            "people"
[45] "point"           "policy"          "question"        "really"
[49] "right"           "say"             "see"             "something"
[53] "take"            "talk"            "talking"         "technical"
[57] "terms"           "thank"           "that"            "that's"
[61] "the"             "there"           "they"            "thing"
```

```
[65] "things"         "think"          "this"           "time"
[69] "two"            "use"            "way"            "will"
[73] "work"           "working"        "world"          "you"
[77] "access"         "but"            "community"      "content"
[81] "freedom"        "get"            "human"          "it's"
[85] "local"          "online"         "rights"         "want"
[89] "{oops/}"        "   "            "  "             "     "
[93] "    "           "              "
```

```
tm::inspect(tm::removeSparseTerms(dtm, sparse=0.4)) # remove sparse terms and inspect the DT
```

```
<<DocumentTermMatrix (documents: 63, terms: 627)>>
Non-/sparse entries: 32051/7450
Sparsity           : 19%
Maximal term length: 17
Weighting          : term frequency (tf)
Sample             :

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security a
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security a
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt
```

```
Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security a
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security a
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security a
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security an
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNET
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security an
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNET
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security an
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNET
  12 ICANN OPEN FORUM.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
  2 145 Nusa Dua Hall 1.txt
  21 WS 15 CYBERCRIME TREATY.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security an
  59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNET
```

```
 12 ICANN OPEN FORUM.txt
 15 OPENING CEREMONY AND OPENING SESSION.txt
 17 SECURITY_LEGAL_AND_OTHER_FRAMEWORKS_SPAM_HACKCYBERCRIME.txt
 2 145 Nusa Dua Hall 1.txt
 21 WS 15 CYBERCRIME TREATY.txt
 26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
 27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
 31 WS 49 Breaking Down Silos in National and International Cooperation on Cyber Security ar
 59 WS 234 DANGERS OF INTERNET ECONOMY FROM IRRESPONSIBLE SCALE.txt
```

3.11 Deliverable 17: Finding Word Associations in the DTM

```
tm::findAssocs(dtm, "activists", 0.8) # find words associated with "activists" with a min tdi
```

```
$activists
       moral coalitions.      hackers,    tunisia.
        0.83         0.82         0.82         0.80
```

```
tm::findAssocs(dtm, "cybersecurity", 0.8) # find words associated with "cybersecurity" with a
```

```
$cybersecurity
         terrorism             bleeds        increasing,           nation's
              0.94               0.91               0.91               0.91
         combating     cybersecurity,              norm,        cybercrime,
              0.90               0.88               0.87               0.83
            chris,    infrastructures              spam,          malicious
              0.83               0.83               0.82               0.82
              spam           spamming            "spam"                 '04
              0.81               0.81               0.81               0.81
              '05,               '06,              '06.                '17,
              0.81               0.81               0.81               0.81
             '990s             (beep)          (beep) --        (security):
              0.81               0.81               0.81               0.81
    1770-something    1770-something,               2016              2016?
              0.81               0.81               0.81               0.81
             24--7              5:00,               >>k.               abcs
              0.81               0.81               0.81               0.81
            accede        accomplish,              acdc,               acm,
              0.81               0.81               0.81               0.81
           acronym            adamant            adopt.            adopt --
```

0.81	0.81	0.81	0.81
advertisements,	affiliates,	agencies'	agency;
0.81	0.81	0.81	0.81
ain't	analytics,	analyzed,	answered?
0.81	0.81	0.81	0.81
anti-abuse	anti-phishing	anti-spam	antispam
0.81	0.81	0.81	0.81
anyone --	apcert,	arises?	article.
0.81	0.81	0.81	0.81
aspects;	aspects?	assault	assist,
0.81	0.81	0.81	0.81
attuned	audience --	auscert	authenticating
0.81	0.81	0.81	0.81
avail	back:	backs.	batnet
0.81	0.81	0.81	0.81
beep	body?	botnet,	botnet-like
0.81	0.81	0.81	0.81
botnets,	boundaries,	box,	boyer
0.81	0.81	0.81	0.81
boyer:	branding.	brian	building --
0.81	0.81	0.81	0.81
burst	buttons	c-level	caller
0.81	0.81	0.81	0.81
calling,	canspam.	capability.	certs?
0.81	0.81	0.81	0.81
chair's	chance.	characteristic	charities.
0.81	0.81	0.81	0.81
chris?	chris --	circus	citizen networks
0.81	0.81	0.81	0.81
classified	clean.	click,	clogging
0.81	0.81	0.81	0.81
closed.	closer,	closes	commercials
0.81	0.81	0.81	0.81
commercial --	commonwealth.	communities --	complete?
0.81	0.81	0.81	0.81
components.	computer --	conflating	congratulations,
0.81	0.81	0.81	0.81
construed	contents.	cooperate --	counterparts --
0.81	0.81	0.81	0.81
counterterrorism,	counterterrorism.	country;	crimes
0.81	0.81	0.81	0.81
cure.	cured.	cyberattacks	cybercapacity
0.81	0.81	0.81	0.81

cybercrime-related	cybercrime;	cybercrime --	cyberevent
0.81	0.81	0.81	0.81
cyberlaw.	cyberthreats,	dangerous,	daniel
0.81	0.81	0.81	0.81
debated,	deeds	defenses	define --
0.81	0.81	0.81	0.81
degradation	destroying	diplomat,	discern
0.81	0.81	0.81	0.81
discussion.	dismissed	disposal.	disservice,
0.81	0.81	0.81	0.81
dominican	donations,	done --	doorstep
0.81	0.81	0.81	0.81
doorstep.	dors	driver.	drops,
0.81	0.81	0.81	0.81
drove	drunk.	earlier --	educated.
0.81	0.81	0.81	0.81
employer	enabler.	enablers.	enentire
0.81	0.81	0.81	0.81
enforcement;	enriched,	enrichment	enter,
0.81	0.81	0.81	0.81
eu-funded	european --	ex-colleagues	except,
0.81	0.81	0.81	0.81
executive	expressions,	extra-territorial	faso hassan.
0.81	0.81	0.81	0.81
fernando,	fierce	fighting.	fines
0.81	0.81	0.81	0.81
fining	firs,	floated --	florida
0.81	0.81	0.81	0.81
follow-.	four --	frameworks:	frameworks:
0.81	0.81	0.81	0.81
fraud?	ftc,	gain,	gambling,
0.81	0.81	0.81	0.81
getting,	gideon	gideon,	give.
0.81	0.81	0.81	0.81
glove.	government --	grass-root	grass-roots
0.81	0.81	0.81	0.81
gsa,	hacking-related	hacks,	haming
0.81	0.81	0.81	0.81
handed,	hands-	harmonization,	headphones
0.81	0.81	0.81	0.81
hijack	idea --	ills	impinges
0.81	0.81	0.81	0.81
implement.	inconvenience.	increasing --	ineffective

```
           0.81                0.81                0.81                0.81
         infect           infection          infections        infections.
           0.81                0.81                0.81                0.81
  infections?               infects   innovation-based          instructor
           0.81                0.81                0.81                0.81
  integration.  internationally --  internationals.      interoperable,
           0.81                0.81                0.81                0.81
  interrelated,     investigating,          irritating             jammed,
           0.81                0.81                0.81                0.81
            jay          jayantha?               jobs?             johnson
           0.81                0.81                0.81                0.81
      johnson.              jpcert          judiciary,            jurists,
           0.81                0.81                0.81                0.81
         karen,             keshted            labeled,             last --
           0.81                0.81                0.81                0.81
      law-based            leapfrog          legislator       legislators.
           0.81                0.81                0.81                0.81
        lepris,            liaisons              litany               maawg
           0.81                0.81                0.81                0.81
         maawg,             maawg.            maawg --               mail.
           0.81                0.81                0.81                0.81
       mailbox,           mailboxes            makarim,            makarim:
           0.81                0.81                0.81                0.81
       malware,            malware.             married          mayu fumo,
           0.81                0.81                0.81                0.81
         merged         messaging.            mexico's               mic).
           0.81                0.81                0.81                0.81
    microphones,        misconduct,            mismatch            mobiles,
           0.81                0.81                0.81                0.81
     moderately              month.           montreal,                mood
           0.81                0.81                0.81                0.81
    motivation.              mpasa,           mulberry,          mulberry.
           0.81                0.81                0.81                0.81
      mulberry:             must --      national-level             natris,
           0.81                0.81                0.81                0.81
        natris:                ncic           nefarious         netterlands
           0.81                0.81                0.81                0.81
      non-south        nonsolicited            nonstate     normal," one
           0.81                0.81                0.81                0.81
        note --           notifying            nuisance          nuisance.
           0.81                0.81                0.81                0.81
          oddly           offenses,           offenses.             omnibus
           0.81                0.81                0.81                0.81
```

one? -- 0.81	onwards 0.81	onwards, 0.81	open--shut, 0.81
opt- 0.81	opted 0.81	osc, 0.81	outfits 0.81
outlining 0.81	overlap, 0.81	overwhelmed 0.81	painter 0.81
painter. 0.81	painter: 0.81	panel -- 0.81	partners. -- 0.81
pass. 0.81	pcs 0.81	perspective? 0.81	perspective -- 0.81
pillar. 0.81	pipes, 0.81	plaintiffs. 0.81	policymakers. 0.81
possible? 0.81	postgraduate 0.81	preference -- 0.81	presenters, 0.81
pretended 0.81	preventative 0.81	privacy-sensitive 0.81	profitable, 0.81
promote. 0.81	promoting -- 0.81	promotion, 0.81	promptly 0.81
pronounced 0.81	pronounced, 0.81	propaganda 0.81	proportion, 0.81
proportions, 0.81	prpt 0.81	psace 0.81	put -- 0.81
python's 0.81	quantity 0.81	question -- 0.81	raising? 0.81
rater, 0.81	realisation 0.81	receiver 0.81	receptive 0.81
reduction. 0.81	reevaluate 0.81	regarded -- 0.81	regarding -- 0.81
regardless, 0.81	region. 0.81	remember? 0.81	remit. 0.81
remote -- 0.81	reorganisation 0.81	requiring, 0.81	resnick. 0.81
resources? 0.81	revolutionary 0.81	rican 0.81	router, 0.81
routes. 0.81	sadowski. 0.81	saturate, 0.81	schedules 0.81
scheme. 0.81	segueing 0.81	self-aid 0.81	self-governance 0.81
self-regulation, 0.81	senders 0.81	servers -- 0.81	shalt 0.81
sharing? 0.81	significantly. 0.81	siphoned 0.81	sketch 0.81
socialize --	spam.	spam?	spamed,

```
              0.81               0.81               0.81               0.81
           spammer           spammers          spamming,            spam --
              0.81               0.81               0.81               0.81
          speak --     spear-phishing    spear-phishing,       specialists,
              0.81               0.81               0.81               0.81
   standards-based            stated,          statutory           stopped,
              0.81               0.81               0.81               0.81
       streamlined          subjects,         subject --           succeed,
              0.81               0.81               0.81               0.81
      sufficient.           summaries        surprising,        tailor-made
              0.81               0.81               0.81               0.81
         takeaway,          takedowns             talks.           targeted.
              0.81               0.81               0.81               0.81
           tasks.    technology-based               teed         territory.
              0.81               0.81               0.81               0.81
       terrorists,             theft,       therefore --               thou
              0.81               0.81               0.81               0.81
          thought-            tiarma.             tighten               tong
              0.81               0.81               0.81               0.81
          toolkit,             top --               tout         tradition,
              0.81               0.81               0.81               0.81
      traditions.         trainings?         transborder             tween.
              0.81               0.81               0.81               0.81
           ugandan                ult   uncharacteristic       uncontinueed
              0.81               0.81               0.81               0.81
        undisputed      unidentifying        unsolicited           variety,
              0.81               0.81               0.81               0.81
        vehicle --                vep               waas           wanteded
              0.81               0.81               0.81               0.81
            wcit.          website..                wild               wout
              0.81               0.81               0.81               0.81
             wout,             wout.               after             system
              0.81               0.81               0.81               0.81
          efforts,           minimize
              0.80               0.80
```

```r
tm::inspect(tm::DocumentTermMatrix(igfbali,
            list(dictionary = c("multistakeholder", "freedom", "development")))) # create a
```

```
<<DocumentTermMatrix (documents: 63, terms: 3)>>
Non-/sparse entries: 137/52
Sparsity           : 28%
```

```
Maximal term length: 16
Weighting          : term frequency (tf)
Sample             :

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  14 INTERNET_GOVERNANCE_PRINCIPLES.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  33 WS 57 MAKING MULTISTAKEHOLDERISM MORE EQUITABLE AND TRANSPARENT.txt
  38 WS 357 THE INTERNET AS AN ENGINE FOR GROWTH AND ADVANCEMENT.txt
  45 WS-297_PROTECTING_JOURNALISTS_BLOGGERS_AND_MEDIA_ACTORS_IN_DIGITAL_AGE.txt
  6 BUILDING BRIDGES - ENHANCING MULTI-STAKEHOLDER COOPERATION FOR GROWTH AND SUSTAINABLE.tx
  60 WS 300 DEVELOPING A STRATEGIC VISION FOR INTERNET GOVERNANCE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  14 INTERNET_GOVERNANCE_PRINCIPLES.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  33 WS 57 MAKING MULTISTAKEHOLDERISM MORE EQUITABLE AND TRANSPARENT.txt
  38 WS 357 THE INTERNET AS AN ENGINE FOR GROWTH AND ADVANCEMENT.txt
  45 WS-297_PROTECTING_JOURNALISTS_BLOGGERS_AND_MEDIA_ACTORS_IN_DIGITAL_AGE.txt
  6 BUILDING BRIDGES - ENHANCING MULTI-STAKEHOLDER COOPERATION FOR GROWTH AND SUSTAINABLE.tx
  60 WS 300 DEVELOPING A STRATEGIC VISION FOR INTERNET GOVERNANCE.txt

Docs
  10 OPENNESS HUMAN RIGHTS FREEDOM OF EXPRESSION AND FREE FLOW OF INFORMATION ON THE INTERNE
  14 INTERNET_GOVERNANCE_PRINCIPLES.txt
  15 OPENING CEREMONY AND OPENING SESSION.txt
  26 WS 44 FREEDOM ONLINE COALITION OPEN FORU1.txt
  27 WS 44 FREEDOM ONLINE COALITION OPEN FORUM.txt
  33 WS 57 MAKING MULTISTAKEHOLDERISM MORE EQUITABLE AND TRANSPARENT.txt
  38 WS 357 THE INTERNET AS AN ENGINE FOR GROWTH AND ADVANCEMENT.txt
  45 WS-297_PROTECTING_JOURNALISTS_BLOGGERS_AND_MEDIA_ACTORS_IN_DIGITAL_AGE.txt
  6 BUILDING BRIDGES - ENHANCING MULTI-STAKEHOLDER COOPERATION FOR GROWTH AND SUSTAINABLE.tx
  60 WS 300 DEVELOPING A STRATEGIC VISION FOR INTERNET GOVERNANCE.txt
```

4 Part 4: Introduction to Data Wrangling in Python

```
import nltk # import nltk
nltk.download('reuters') # download reuters corpus
```

```
True
```

```
from nltk.corpus import reuters # import reuters corpus
print("Categories:", reuters.categories()) # print categories in reuters corpus
```

```
Categories: ['acq', 'alum', 'barley', 'bop', 'carcass', 'castor-oil', 'cocoa', 'coconut', 'co
```

```
print("Number of documents:", len(reuters.fileids())) # print number of documents in reuters
```

```
Number of documents: 10788
```

```
import string

doc_id = reuters.fileids(categories="crude")[0] # get a document id from the crude category
doc_text = reuters.raw(doc_id) # get raw text from document

cleaned_text = doc_text.translate(str.maketrans('', '', string.punctuation)) # clean text of
cleaned_text = ' '.join(cleaned_text.split()) # join text
print(cleaned_text)
```

```
JAPAN TO REVISE LONGTERM ENERGY DEMAND DOWNWARDS The Ministry of International Trade and Indu
```

4.1 Deliverable 19: Tokenization, Stemming, and Lemmatization of the Reuters Corpus

```
from nltk.tokenize import word_tokenize # import word_tokenize function from nltk.tokenize mo
from nltk.corpus import stopwords # import stopwords corpus from nltk.corpus module
nltk.download('punkt_tab')
```

```
True
```

```
tokens = word_tokenize(cleaned_text) # tokenize words from above
tokens = [word for word in tokens if word not in stopwords.words('english')] # remove stop wo
print(tokens)
```

```
['JAPAN', 'TO', 'REVISE', 'LONGTERM', 'ENERGY', 'DEMAND', 'DOWNWARDS', 'The', 'Ministry', 'I
```

```
from nltk.stem import PorterStemmer, WordNetLemmatizer # import functions from nltk.stem
stemmer = PorterStemmer() # initiate instance of stemmer
lemmatizer = WordNetLemmatizer() # initiate instance of lemmatizer
stemmed = [stemmer.stem(word) for word in tokens] # stem tokens
lemmatized = [lemmatizer.lemmatize(word) for word in tokens] # lemmatize tokens
print("Stemmed:", stemmed)
```

```
Stemmed: ['japan', 'to', 'revis', 'longterm', 'energi', 'demand', 'downward', 'the', 'minist
```

```
print("Lemmatized:", lemmatized)
```

```
Lemmatized: ['JAPAN', 'TO', 'REVISE', 'LONGTERM', 'ENERGY', 'DEMAND', 'DOWNWARDS', 'The', 'M
```

4.2 Deliverable 20: Conducting a Basic Parts of Speech Tagging of the Reuters Corpus

```
from nltk import pos_tag # import function
nltk.download('averaged_perceptron_tagger_eng')
```

```
True
```

```
tagged_tokens = pos_tag(tokens) # tag parts of speech
print(tagged_tokens) # print pos tagged tokens
```

```
[('JAPAN', 'NNP'), ('TO', 'NNP'), ('REVISE', 'NNP'), ('LONGTERM', 'NNP'), ('ENERGY', 'NNP'),
```

4.3 Deliverable 21: Full Text Processing Pipeline for the Reuters Corpus

```python
def preprocess_pipeline(text): # create a data processing pipeline function
  text = text.lower().translate(str.maketrans('', '', string.punctuation))
  text = ' '.join(text.split())
  tokens = word_tokenize(text)
  tokens = [word for word in tokens if word not in stopwords.words('english')]
  lemmatized = [lemmatizer.lemmatize(word) for word in tokens]
  tagged = pos_tag(lemmatized)
  return tagged

doc_text = reuters.raw(reuters.fileids(categories='crude')[0]) # pre process data from reute
processed = preprocess_pipeline(doc_text)
print(processed)
```

```
[('japan', 'NN'), ('revise', 'NN'), ('longterm', 'JJ'), ('energy', 'NN'), ('demand', 'NN'),
```