CPTS 475 Homework 4

Andrew Balaschak

2023-09-27

1

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
## Loading required package: RColorBrewer
```

1.a. Filter the dataset (using a left join) to display the tail number, year, month, day, hour, origin, and humidity for all flights heading to Tampa International

Airport (TPA) after 12pm on November 1, 2013. How many flights happened during the given time frame that day?

```
# Filtering the dataset to only flights to Tampa on November 1, 2013 after 12 pm
flights_tpa <- nycflights13::flights %>% filter(dest == "TPA" & year == 2013 & month == 11 & day == 1 & hour >= 12)

# Left join with weather so that we can get the humidity at the origin airport
flights_tpa <- flights_tpa %>% left_join(nycflights13::weather, by = c("year", "month", "day", "hour", "origin"))

# Selecting the requested columns
flights_tpa <- flights_tpa %>% select(tailnum, year, month, day, hour, origin, humid)
print(flights_tpa)
```

```
## # A tibble: 10 × 7
##    tailnum  year month   day  hour origin humid
##    <chr>   <int> <int> <int> <dbl> <chr>  <dbl>
##  1 N580JB   2013    11     1    14 JFK     63.1
##  2 N337NB   2013    11     1    14 LGA     56.5
##  3 N567UA   2013    11     1    15 EWR     52.8
##  4 N515MQ   2013    11     1    14 JFK     63.1
##  5 N779JB   2013    11     1    15 EWR     52.8
##  6 N561JB   2013    11     1    16 LGA     50.6
##  7 N974DL   2013    11     1    18 JFK     74.8
##  8 N319NB   2013    11     1    19 LGA     60.5
##  9 N76265   2013    11     1    19 EWR     72.5
## 10 N768JB   2013    11     1    19 JFK     83.5
```

There were 10 flights with the destination of Tampa International Airport after 12pm on November 1, 2013

1.b. What is the difference between the following two joins?

```
anti_join(flights, airports, by = c("dest" = "faa"))
```

```
## # A tibble: 7,602 × 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      544            545        -1     1004           1022
##  2  2013     1     1      615            615         0     1039           1100
##  3  2013     1     1      628            630        -2     1137           1140
##  4  2013     1     1      701            700         1     1123           1154
##  5  2013     1     1      711            715        -4     1151           1206
##  6  2013     1     1      820            820         0     1254           1310
##  7  2013     1     1      820            820         0     1249           1329
##  8  2013     1     1      840            845        -5     1311           1350
##  9  2013     1     1      909            810        59     1331           1315
## 10  2013     1     1      913            918        -5     1346           1416
## # ℹ 7,592 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>
```

```
anti_join(airports, flights, by = c("faa" = "dest"))
```

```
## # A tibble: 1,357 × 8
##    faa   name                            lat    lon   alt    tz dst   tzone
##    <chr> <chr>                         <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 04G   Lansdowne Airport              41.1  -80.6  1044    -5 A     America/…
##  2 06A   Moton Field Municipal Airport  32.5  -85.7   264    -6 A     America/…
##  3 06C   Schaumburg Regional            42.0  -88.1   801    -6 A     America/…
##  4 06N   Randall Airport                41.4  -74.4   523    -5 A     America/…
##  5 09J   Jekyll Island Airport          31.1  -81.4    11    -5 A     America/…
##  6 0A9   Elizabethton Municipal Airport 36.4  -82.2  1593    -5 A     America/…
##  7 0G6   Williams County Airport        41.5  -84.5   730    -5 A     America/…
##  8 0G7   Finger Lakes Regional Airport  42.9  -76.8   492    -5 A     America/…
##  9 0P2   Shoestring Aviation Airfield   39.8  -76.6  1000    -5 U     America/…
## 10 0S9   Jefferson County Intl          48.1 -123.    108    -8 A     America/…
## # ℹ 1,347 more rows
```

The first join returns rows from flights that don’t have a matching destination in the airports table. The second join returns rows from airports that don’t have any flights going to them.

1.c. Select the origin and destination airports and their latitude and longitude for all flights in the dataset (using one or more inner joins). How many flights are

there in your result?

```
# Join the flights table with the airports table matching origin airport to populate information on origin airport
origin_destination <- flights %>% inner_join(airports, by = c("origin" = "faa")) %>% rename(origin_name = name, origin_lat = lat, origin_lon = lon)

# Join the origin_destination table with the airports table matching destination airport to populate information on destination airport
origin_destination <- origin_destination %>% inner_join(airports, by = c("dest" = "faa")) %>% rename(dest_name = name, dest_lat = lat, dest_lon = lon)

# Select requested columns and print
print(origin_destination %>% select(origin, origin_name, origin_lat, origin_lon, dest, dest_name, dest_lat, dest_lon))
```

```
## # A tibble: 329,174 × 8
##    origin origin_name    origin_lat origin_lon dest  dest_name dest_lat dest_lon
##    <chr>  <chr>               <dbl>      <dbl> <chr> <chr>        <dbl>    <dbl>
##  1 EWR    Newark Libert…       40.7      -74.2 IAH   George B…     30.0    -95.3
##  2 LGA    La Guardia           40.8      -73.9 IAH   George B…     30.0    -95.3
##  3 JFK    John F Kenned…       40.6      -73.8 MIA   Miami In…     25.8    -80.3
##  4 LGA    La Guardia           40.8      -73.9 ATL   Hartsfie…     33.6    -84.4
##  5 EWR    Newark Libert…       40.7      -74.2 ORD   Chicago …     42.0    -87.9
##  6 EWR    Newark Libert…       40.7      -74.2 FLL   Fort Lau…     26.1    -80.2
##  7 LGA    La Guardia           40.8      -73.9 IAD   Washingt…     38.9    -77.5
##  8 JFK    John F Kenned…       40.6      -73.8 MCO   Orlando …     28.4    -81.3
##  9 LGA    La Guardia           40.8      -73.9 ORD   Chicago …     42.0    -87.9
## 10 JFK    John F Kenned…       40.6      -73.8 PBI   Palm Bea…     26.7    -80.1
## # ℹ 329,164 more rows
```

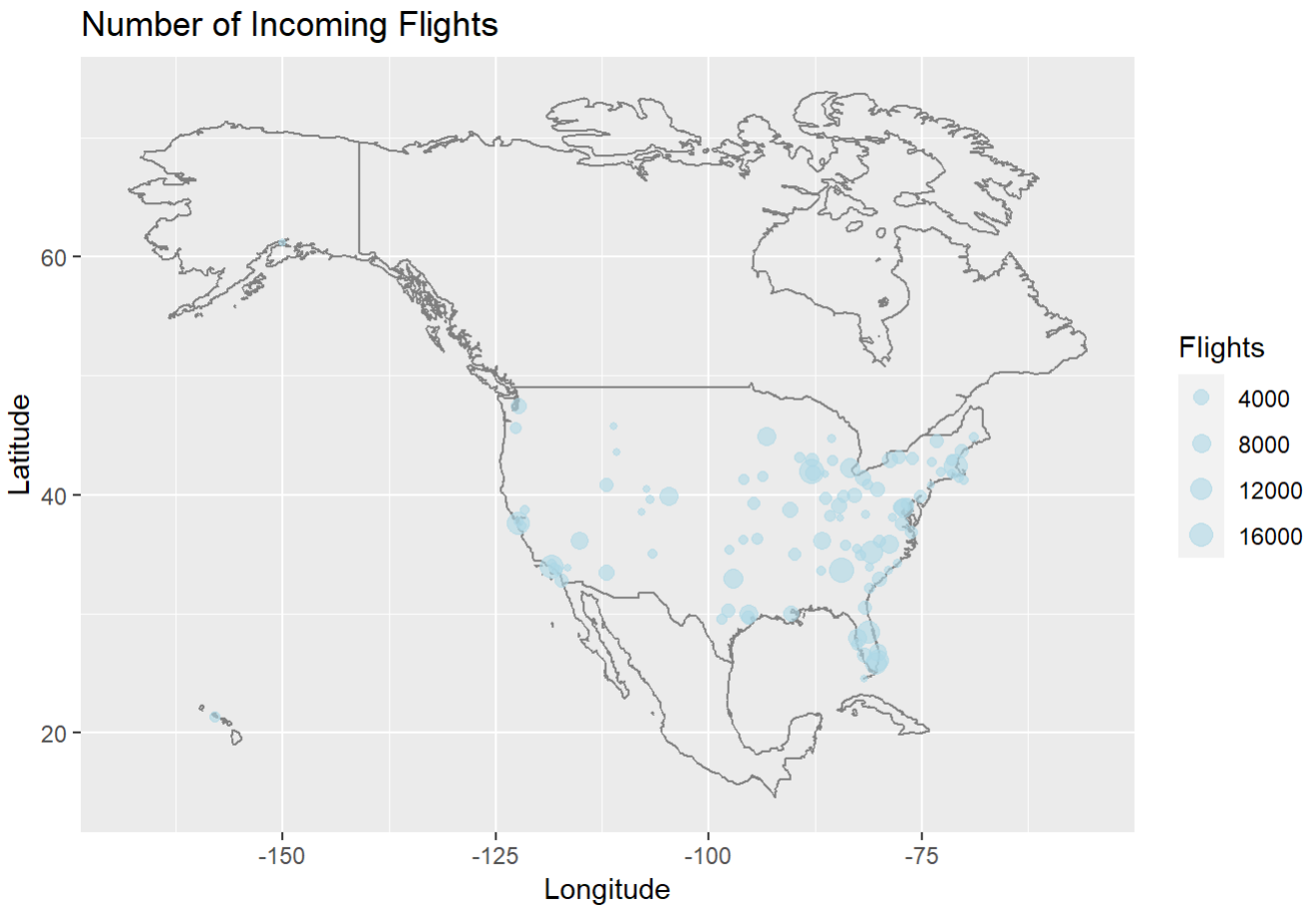
There are 329,174 flights in the dataset. Though this is using an inner join, so if a flight does not have a matching airport in the airports table for both the origin and destination airport, it will be dropped. That is why there are fewer flights in this table than in the flights table.

1.d. Produce a map that sizes each destination airport by the number of incoming flights. You may

use a continuous scale for the size.

```
# Join the flights table with the airports table
flights_airports <- flights %>% left_join(airports, c("dest" = "faa"))

# Count number of flights going to each destination airport
suppressMessages(flights_count <- flights_airports %>% group_by(dest, lat, lon) %>% summarize(fl
ights = n()))

# Plot the map with a continuous scale for the size of the points
flights_map <- flights_count %>%
  ggplot(aes(lon, lat, size = flights)) +
  borders("world", xlim = c(-160, -80), ylim = c(20, 70)) +
  geom_point(color = "lightblue", alpha = 0.6) +
  coord_quickmap() +
  scale_size_continuous(range = c(1, 4)) +
  labs(title = "Number of Incoming Flights",
       x = "Longitude",
       y = "Latitude",
       size = "Flights")

# Print the result
flights_map
```

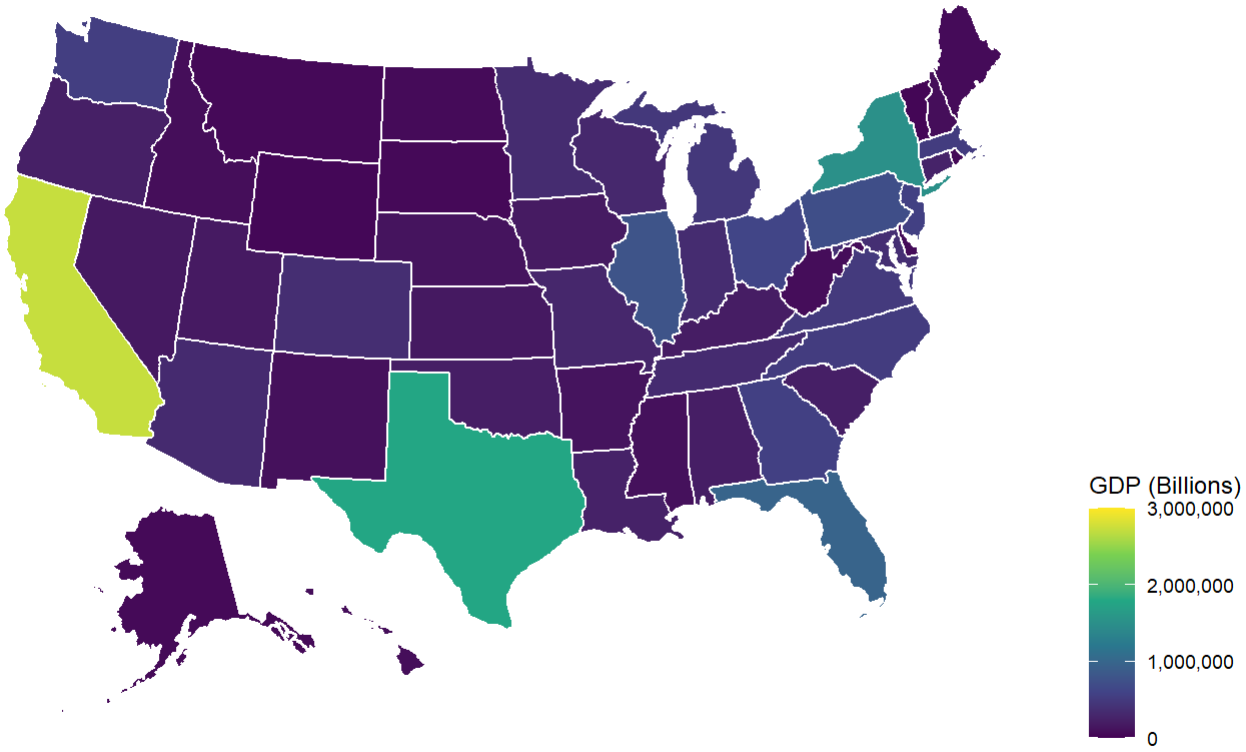
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



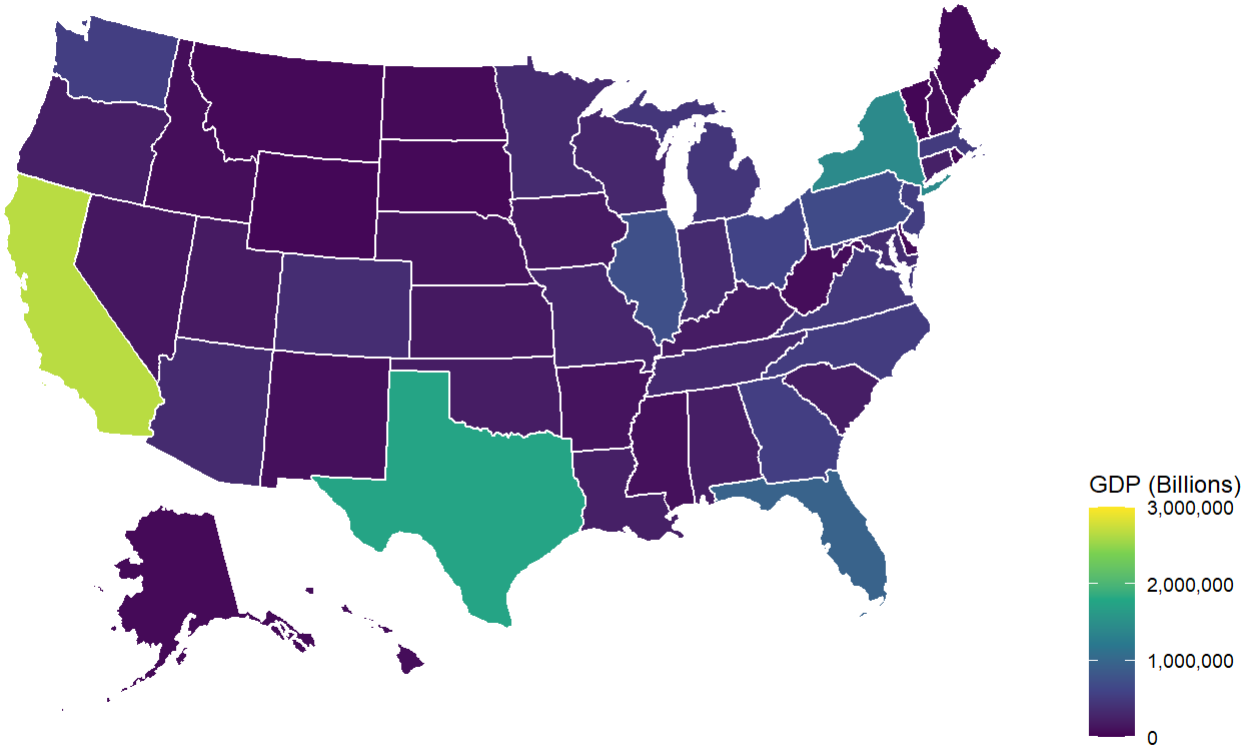
2.Create visualizations of the US map coloring the states or sizing the point/marker for the states according to the GDP for each state (one map per year). Compare the GDP of different states for all the years using the maps you generated (we recommend that you maintain a constant scale for showing the GDP in all the four maps.

```
us_states_gdp <- read.csv("us_states_gdp.csv")

for (i in seq_along(us_states_gdp)[-1]) {
  # Get the name of the current column
  column_name <- colnames(us_states_gdp[i])
  # Get the year from that column name
  year <- substr(column_name, 5, 9)

  # , min = 0, max = 3000000

  # Plot the map with a viridis color scale for the GDP
  us_map_plot <- plot_usmap(data = us_states_gdp, values = column_name, color = "white") +
    scale_fill_continuous(type = "viridis", name = "GDP (Billions)", limits = c(0, 3000000), lab
el = scales::comma) +
    theme(legend.position = "right") +
    labs(title = paste("GDP by State in", year))

  # Print the plot
  print(us_map_plot)
}
```

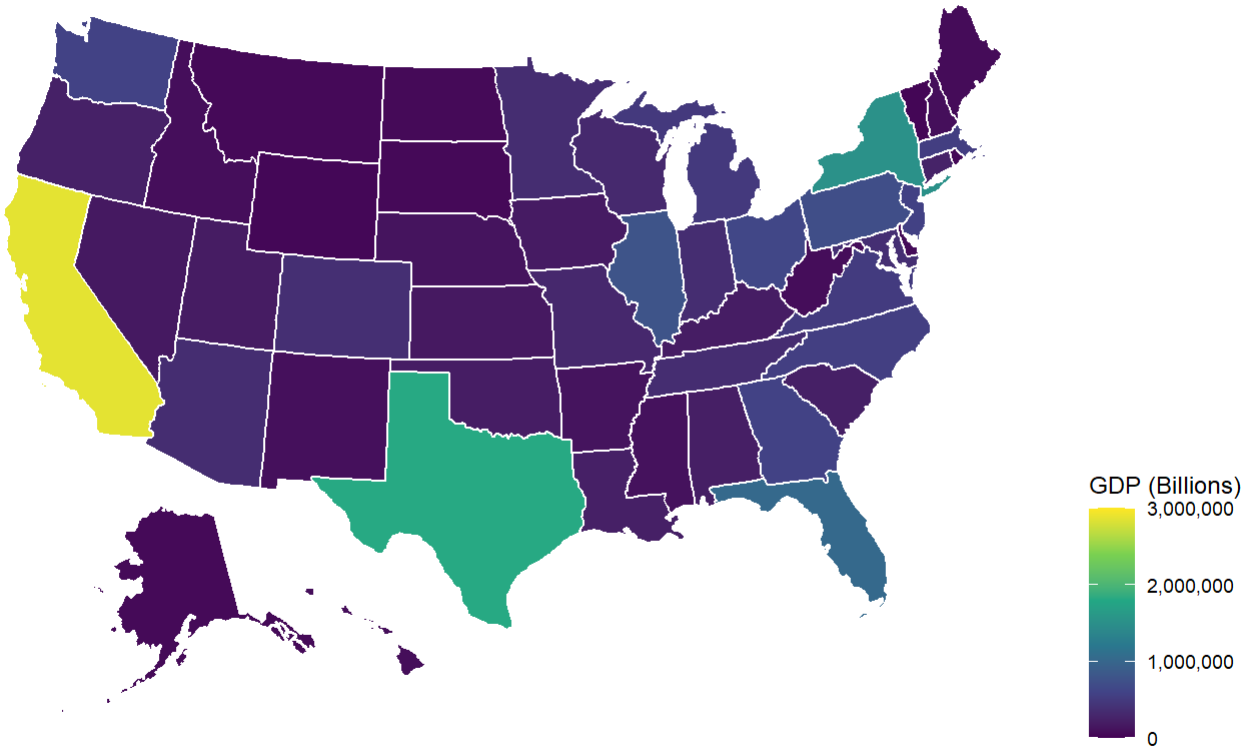
GDP by State in 2019



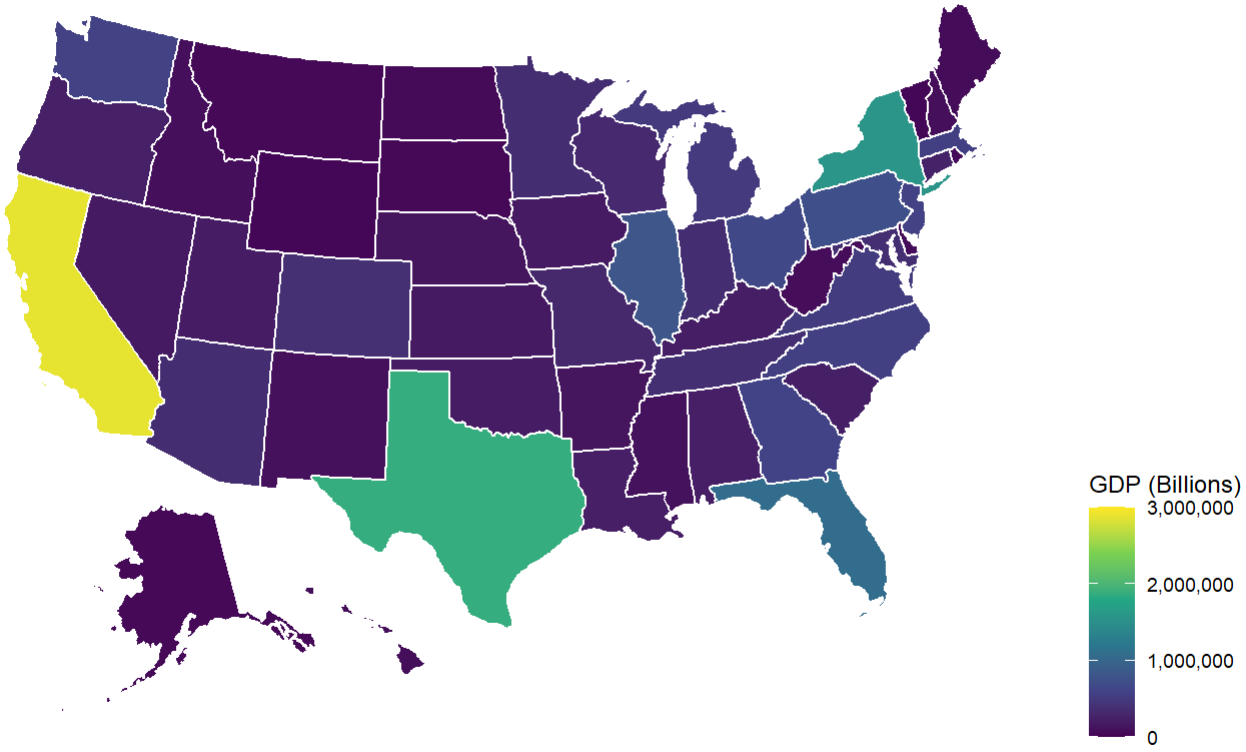
GDP by State in 2020



GDP by State in 2021



GDP by State in 2022



3. Create a word cloud for an interesting (relatively short, say a couple of pages) document of your own choice. Examples of suitable documents include: summary of a recent project you are working or have worked on; your own recent Statement of Purpose or Research Statement or some other similar document.

```
wordbase <- readtext("R-intro.pdf")
corp <- Corpus(VectorSource(wordbase))
corp <- tm_map(corp, PlainTextDocument)
```

```
## Warning in tm_map.SimpleCorpus(corp, PlainTextDocument): transformation drops
## documents
```

```
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, removeNumbers)
corp <- tm_map(corp, tolower)
```

```
## Warning in tm_map.SimpleCorpus(corp, tolower): transformation drops documents
```

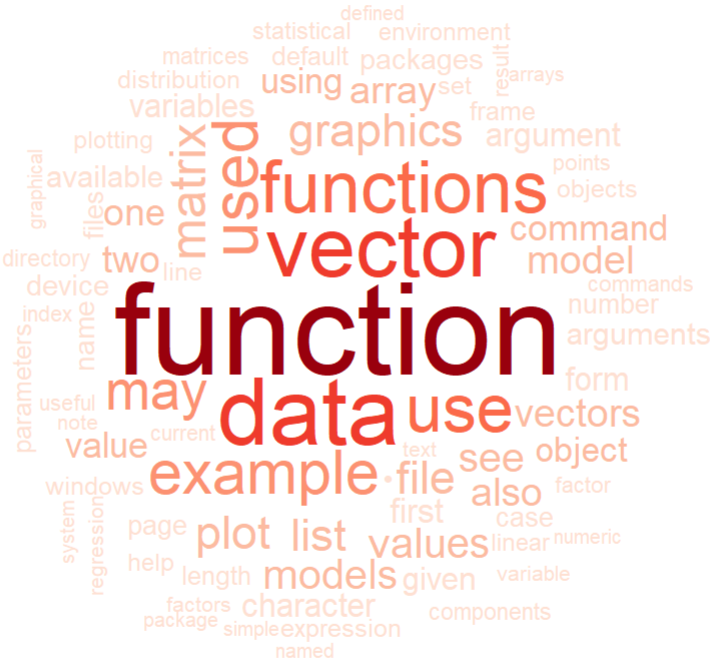
```
corp <- tm_map(corp, removeWords, stopwords(kind = "en"))
```

```
## Warning in tm_map.SimpleCorpus(corp, removeWords, stopwords(kind = "en")):
## transformation drops documents
```

```
corp <- tm_map(corp, removeWords, c("can", "will", "way", "chapter"))
```

```
## Warning in tm_map.SimpleCorpus(corp, removeWords, c("can", "will", "way", :
## transformation drops documents
```

```
color <- brewer.pal(8,"Reds")
wordcloud(corp, max.words = 80, random.order = FALSE, colors = color, scale = c(4,.1))
```



Wordcloud generated from the document “An Introduction to R” by W. N. Venables, D. M. Smith and the R Core Team following the tutorial here: https://www.ryananddebi.com/2017/07/21/r-linux-creating-a-wordcloud-from-pdf/ (https://www.ryananddebi.com/2017/07/21/r-linux-creating-a-wordcloud-from-pdf/)