Project 3 - Data Scrap

TEAM 6

10/15/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(RCurl)
```

```
##
## Attaching package: 'RCurl'

## The following object is masked from 'package:tidyr':
##
##     complete
```

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'

## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
library(stringr)
library(tidytext)
library(wordcloud)
```

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

Load the datafile into folder

```
urlfile<-"https://raw.githubusercontent.com/nolivercuny/data607-team-6-project-3/master/data/job_listing
jobdat <- read_csv(url(urlfile))
```

```
## Rows: 838 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): job_title, company_name, region, salary, employment_type, career_l...
## dbl  (2): search_rank, applicant_count
```

```
##
## 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.
```

```
#view short file summary and class
jobdat<-data_frame(jobdat)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
glimpse(jobdat)
```

```
## Rows: 838
## Columns: 13
## $ search_rank     <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ~
## $ job_title       <chr> "Senior Data Scientist", "Data Scientist", "Data Scien~
## $ company_name    <chr> "TextNow", "Amazon", "Alldus", "Facebook", "Google", "~
## $ region          <chr> "New York, NY", "New York, NY", "New York City Metropo~
## $ applicant_count <dbl> 4, 60, 35, 47, 22, 19, 22, 4, 15, 82, 45, 3, 21, 42, 4~
## $ salary          <chr> NA, NA, NA, "$123,000/yr - $219,000/yr (LinkedIn est.)~
## $ employment_type <chr> "Full-time", "Full-time", "Full-time", "Full-time", "F~
## $ career_level    <chr> NA, NA, "Entry level", NA, NA, NA, "Mid-Senior level",~
## $ company_size    <chr> "51-200 employees", "10,001+ employees", "11-50 employ~
## $ industry        <chr> "Telecommunications", "Internet", "Staffing & Recruiti~
## $ date_queried    <chr> "10/12/21 22:10", "10/12/21 22:10", "10/12/21 22:10", ~
## $ date_posted     <chr> "10 hours ago", "2 weeks ago", "3 weeks ago", "1 week ~
## $ description     <chr> "TextNow is based around a simple idea: Communication ~
```

Unnest the text from the job description field into words and remove "stop words" (the, of, to, etc)

```
jobdat_word<- unnest_tokens(
  jobdat,
  word,
  description,
  token= "words",
  format=c("text"),
  to_lower=TRUE,
  drop=TRUE,
```

```
  collapse=NULL,
)

jobdat_word <-jobdat_word %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
glimpse(jobdat_word)
```

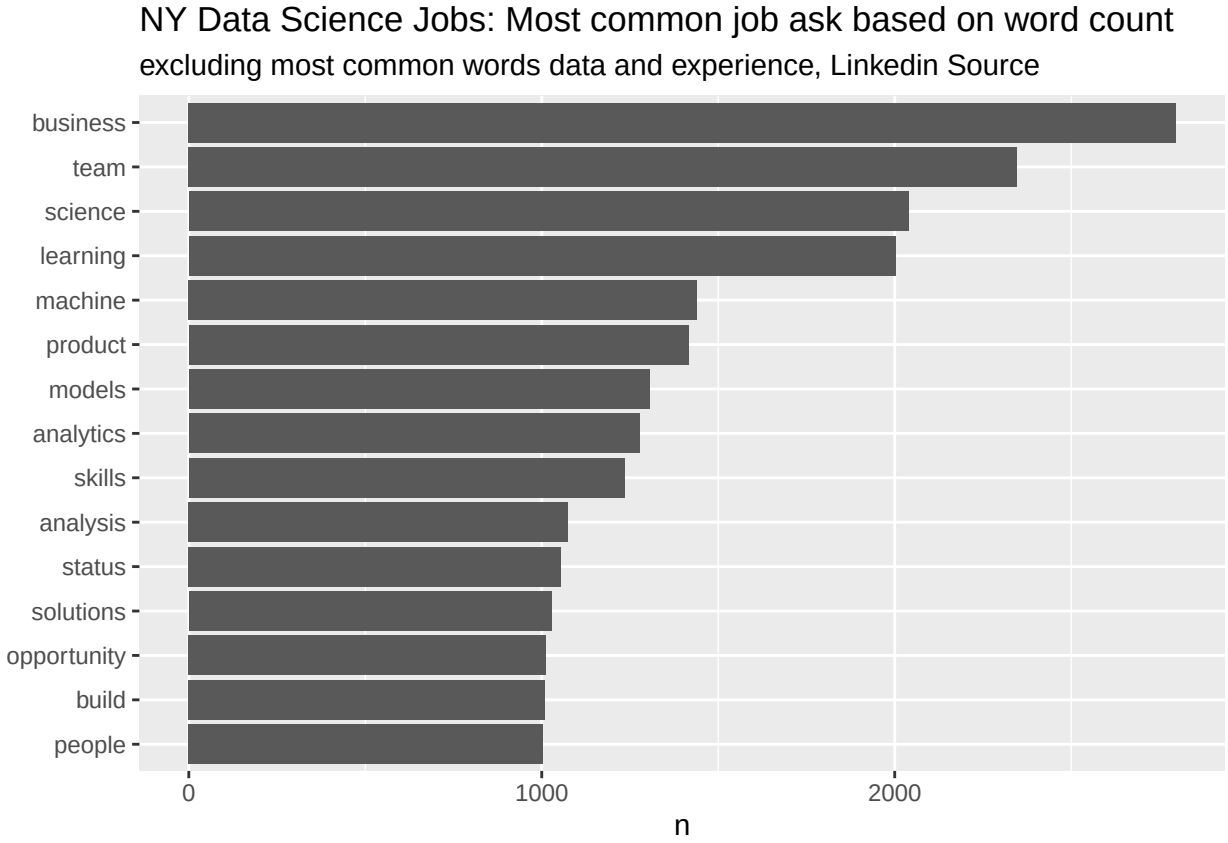
```
## Rows: 293,589
## Columns: 13
## $ search_rank     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ job_title       <chr> "Senior Data Scientist", "Senior Data Scientist", "Sen~
## $ company_name    <chr> "TextNow", "TextNow", "TextNow", "TextNow", "TextNow",~
## $ region          <chr> "New York, NY", "New York, NY", "New York, NY", "New Y~
## $ applicant_count <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ~
## $ salary          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ employment_type <chr> "Full-time", "Full-time", "Full-time", "Full-time", "F~
## $ career_level    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ company_size    <chr> "51-200 employees", "51-200 employees", "51-200 employ~
## $ industry        <chr> "Telecommunications", "Telecommunications", "Telecommu~
## $ date_queried    <chr> "10/12/21 22:10", "10/12/21 22:10", "10/12/21 22:10", ~
## $ date_posted     <chr> "10 hours ago", "10 hours ago", "10 hours ago", "10 ho~
## $ word            <chr> "textnow", "based", "simple", "idea", "communication",~
```

Look at word count in the datafile for most common words

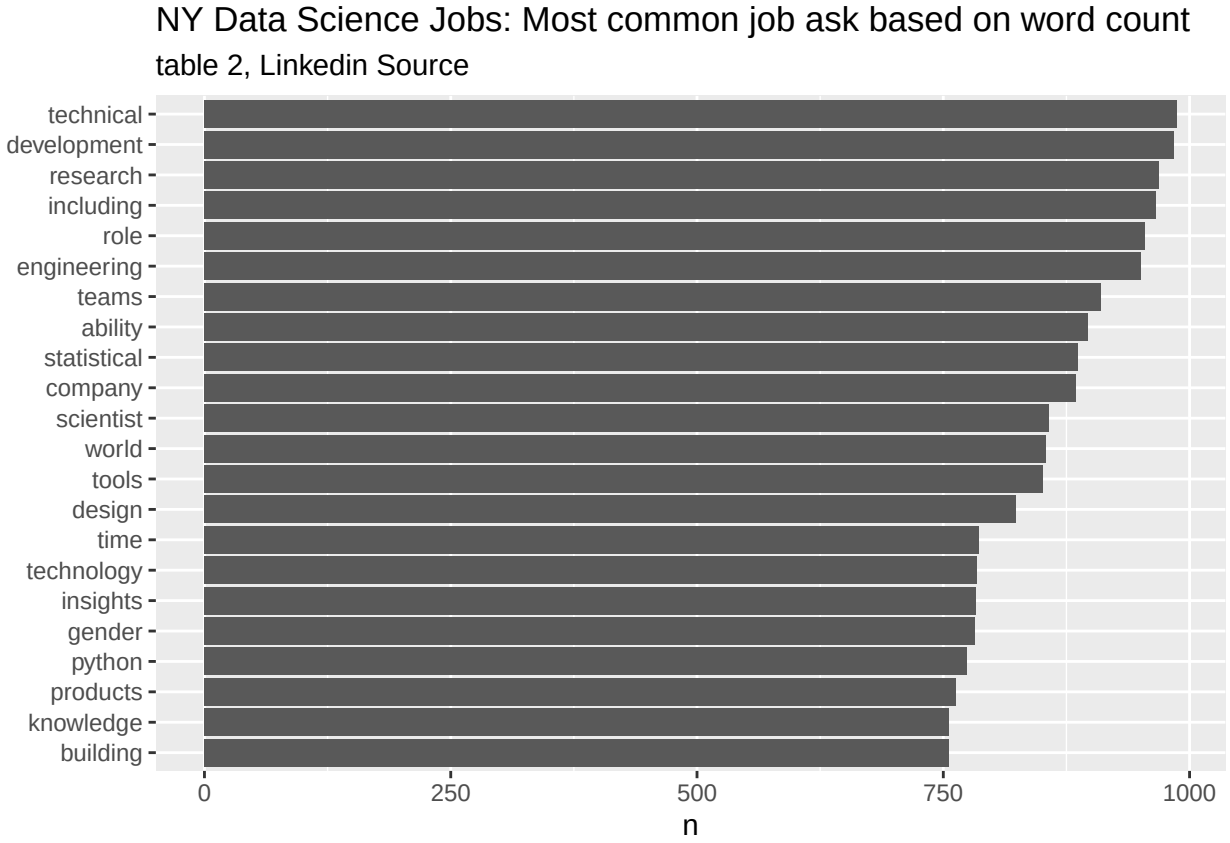
```
word<-jobdat_word %>%
  count(word,sort=TRUE)

company<-jobdat %>%
  count(company_name,sort=TRUE)

industry<-jobdat %>%
  count(industry,sort=TRUE)

region<-jobdat %>%
  count(region,sort=TRUE)

jobtitle<-jobdat %>%
  count(job_title,sort=TRUE)
```

Plot data in first simple plot to review data set of most common word from job description. Follow similar review to plot also the most common companies and industries that have data science jobs open.

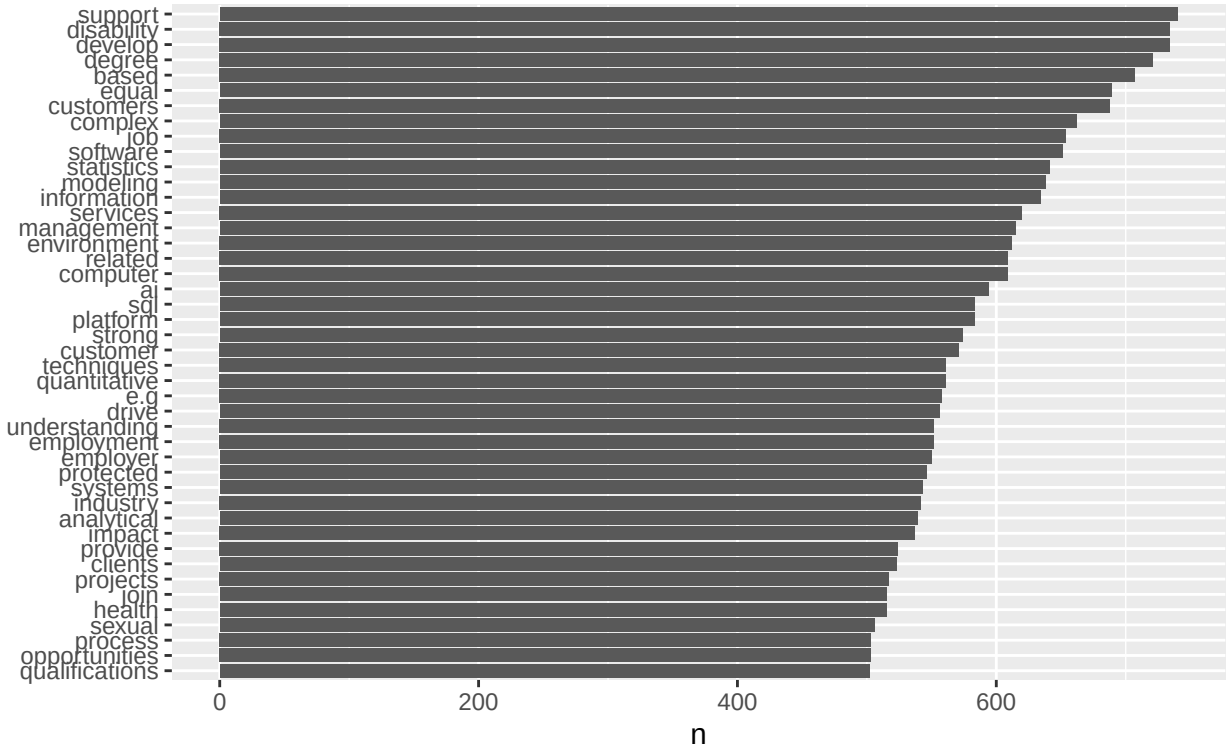
```
library(ggplot2)

jobdat_word %>%
  count(word, sort=TRUE) %>%
  filter(n> 1000, n<3000) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs: Most common job
```



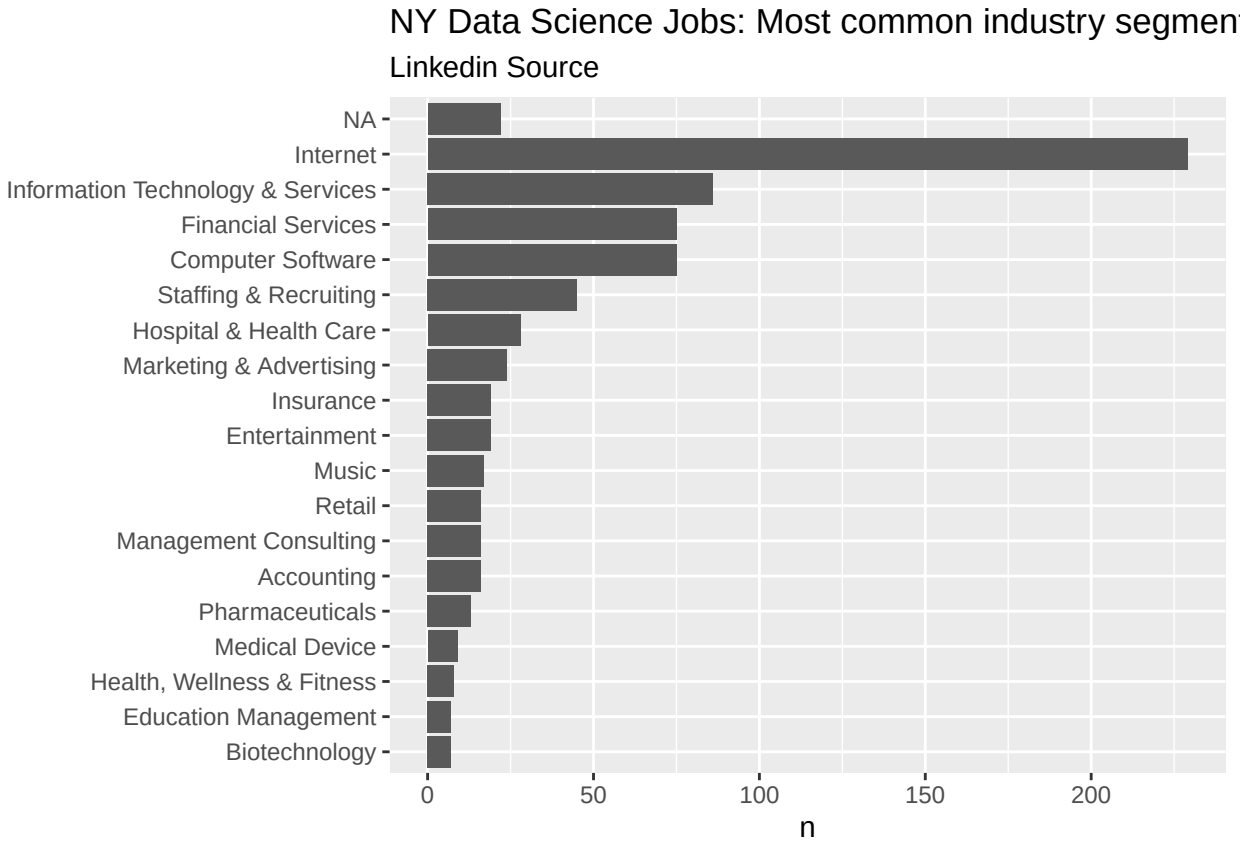

```
jobdat_word %>%
  count(word, sort=TRUE) %>%
  filter(n> 750, n<1000) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs: Most common job
```



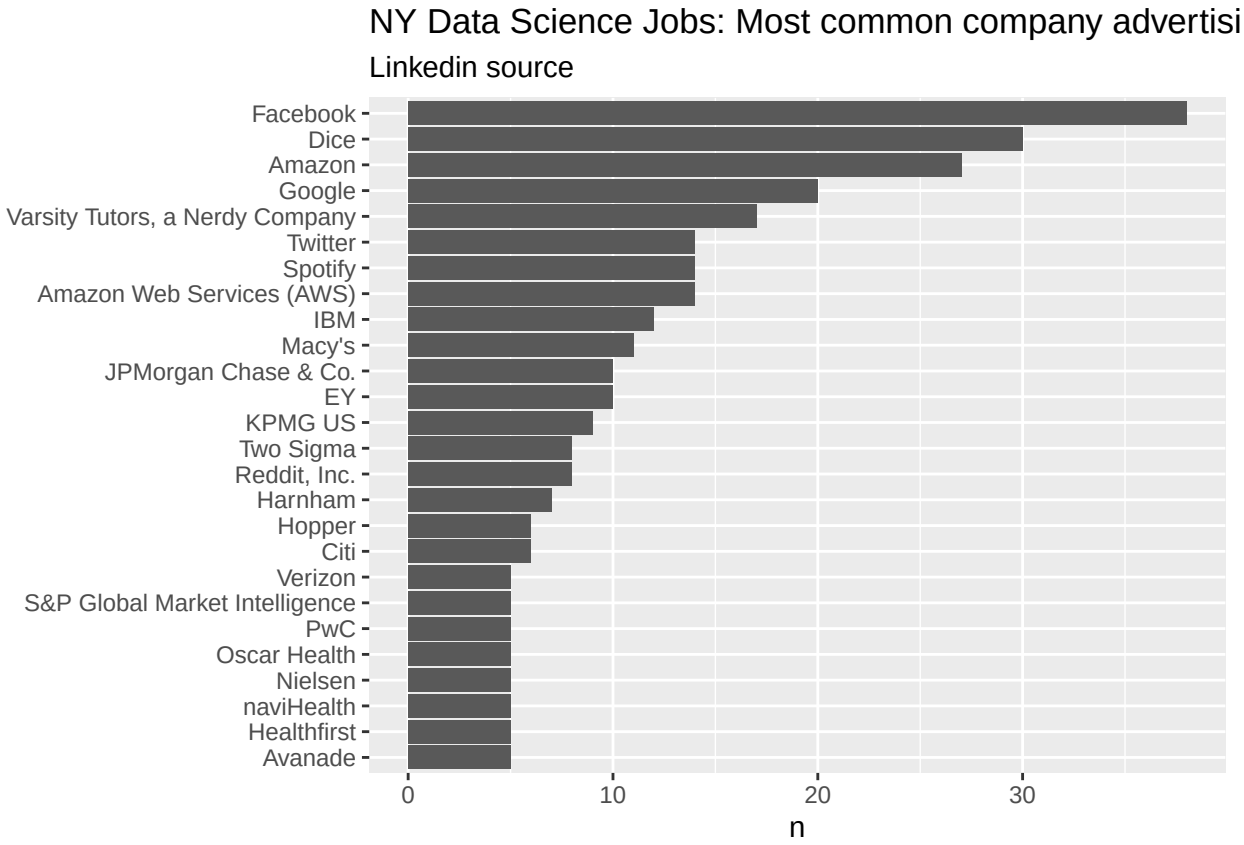

```
jobdat_word %>%
  count(word, sort=TRUE) %>%
  filter(n> 500, n<750) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs: Most common job
```



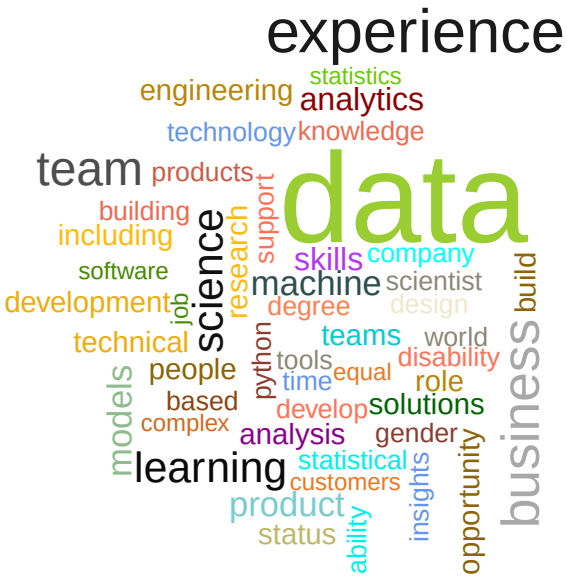

```
jobdat %>%
  count(industry, sort=TRUE) %>%
  filter(n> 6) %>%
  mutate(industry=reorder(industry,n)) %>%
  ggplot(aes(industry,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs: Most common
```




```
jobdat %>%
  count(company_name, sort=TRUE) %>%
  filter(n> 4) %>%
  mutate(company_name=reorder(company_name,n)) %>%
  ggplot(aes(company_name,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs: Most co
```



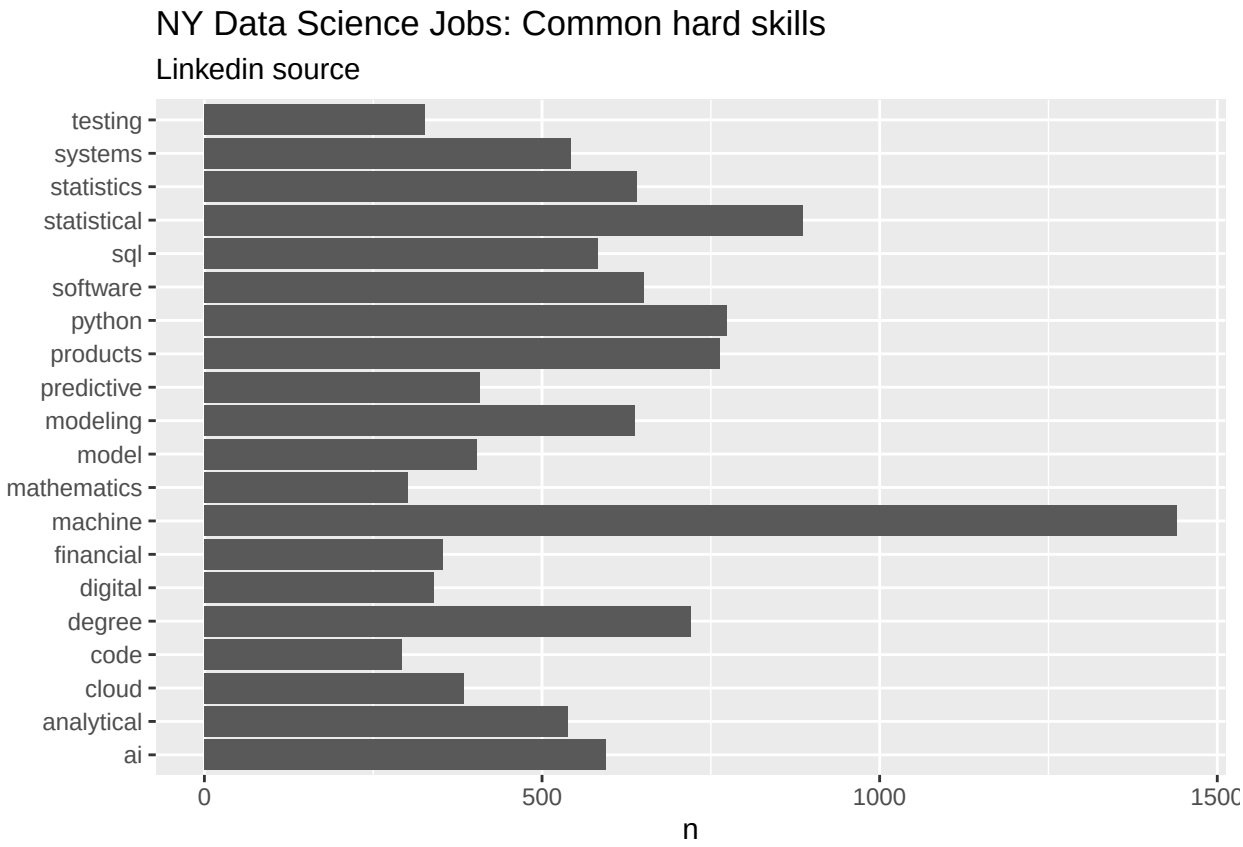

```
library(wordcloud)

jobdat_word %>%
  count(word) %>%
with(wordcloud(word,n,scale=c(4,0.5),max.words=50, random.color = FALSE,rot.per=0.25,colors = colours())
```



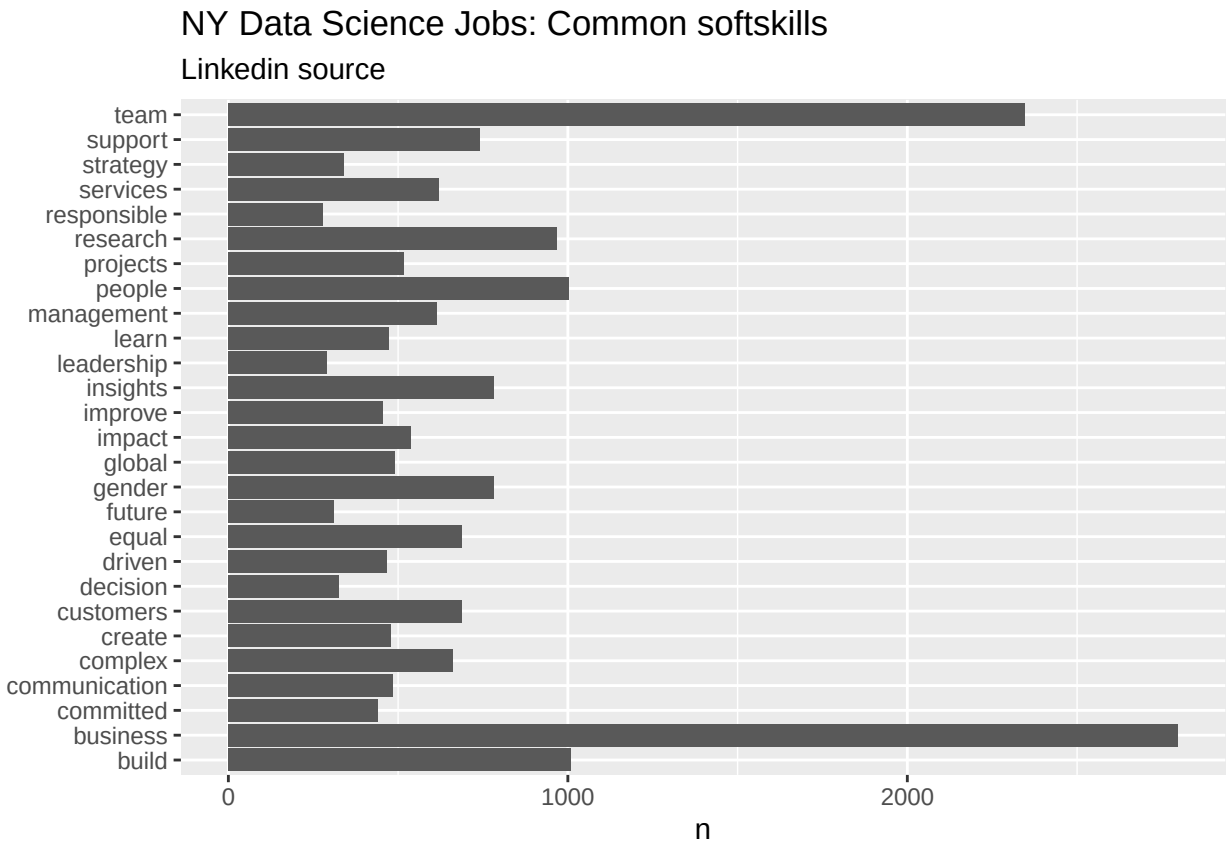

```
jobdat %>%
  count(industry) %>%
with(wordcloud(industry,n,scale=c(3,0.5),max.words=25, random.color = FALSE,rot.per=0.1,colors = colours
```


```
jobdat %>%
  count(company_name) %>%
  with(wordcloud(company_name,n,scale=c(3,0.5),max.words=25, random.color = FALSE,rot.per=0.1,colors = c
```


```
target1<-c("machine","model","modeling","modeling","python","products","statistical","statistics","produ

word_hardskill<-filter (word,word %in% target1)

target2<-c("business","team","build","people","research","insights","gender","support","communication",

word_softskills<- filter(word, word %in% target2)


  ggplot(word_hardskill,aes(word,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs: C
```




```
ggplot(word_softskills,aes(word,n))+geom_col()+xlab(NULL)+coord_flip()+ggtitle("NY Data Science Jobs:
```



R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

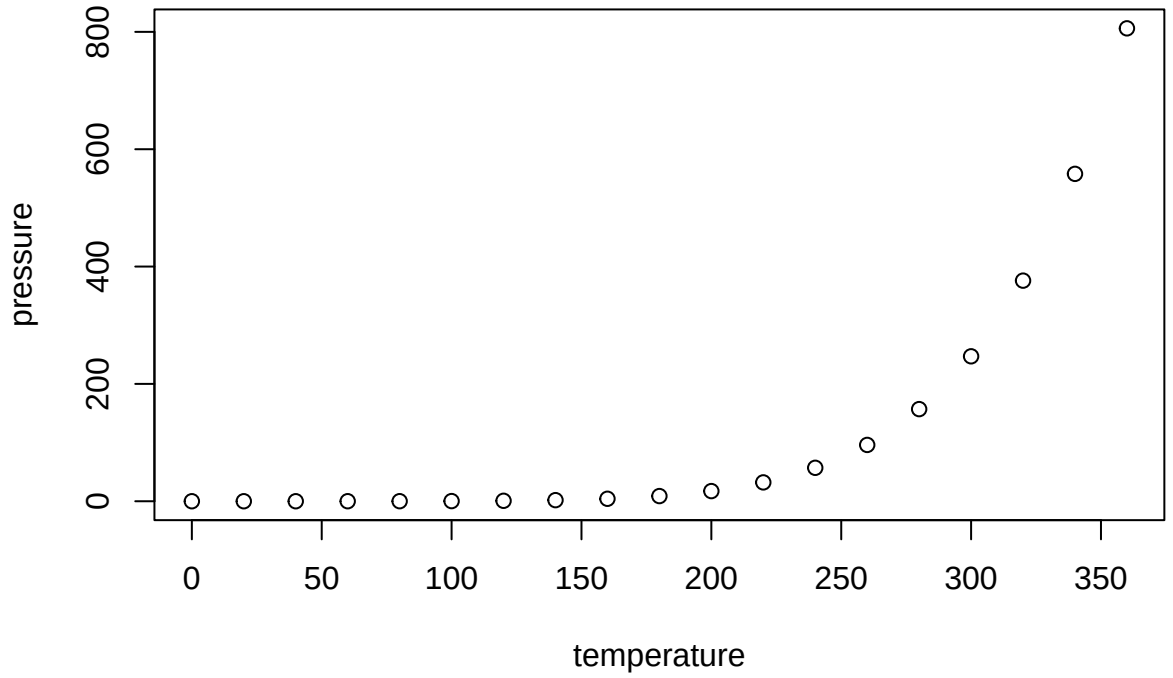
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.