Project 3 - Data Scrap

Mark Schmalfeld

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)
```

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, ~
## $ 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, ~
## $ 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

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

```
## # A tibble: 18,018 x 2
##    word           n
##    <chr>      <int>
##  1 data        8429
##  2 experience  3474
##  3 business    2797
##  4 team        2346
##  5 science     2039
##  6 learning    2003
##  7 machine     1440
##  8 product     1416
##  9 models      1305
## 10 analytics   1277
## # ... with 18,008 more rows
```

```
jobdat %>%
  count(company_name,sort=TRUE)
```

```
## # A tibble: 419 x 2
##    company_name                        n
```

```
##    <chr>                              <int>
##  1 Facebook                              38
##  2 Dice                                  30
##  3 Amazon                                27
##  4 Google                                20
##  5 Varsity Tutors, a Nerdy Company       17
##  6 Amazon Web Services (AWS)             14
##  7 Spotify                               14
##  8 Twitter                               14
##  9 IBM                                   12
## 10 Macy's                                11
## # ... with 409 more rows
```

```
jobdat %>%
  count(industry,sort=TRUE)
```

```
## # A tibble: 62 x 2
##    industry                              n
##    <chr>                             <int>
##  1 Internet                            229
##  2 Information Technology & Services    86
##  3 Computer Software                    75
##  4 Financial Services                   75
##  5 Staffing & Recruiting                45
##  6 Hospital & Health Care               28
##  7 Marketing & Advertising              24
##  8 <NA>                                 22
##  9 Entertainment                        19
## 10 Insurance                            19
## # ... with 52 more rows
```

```
jobdat %>%
  count(region,sort=TRUE)
```

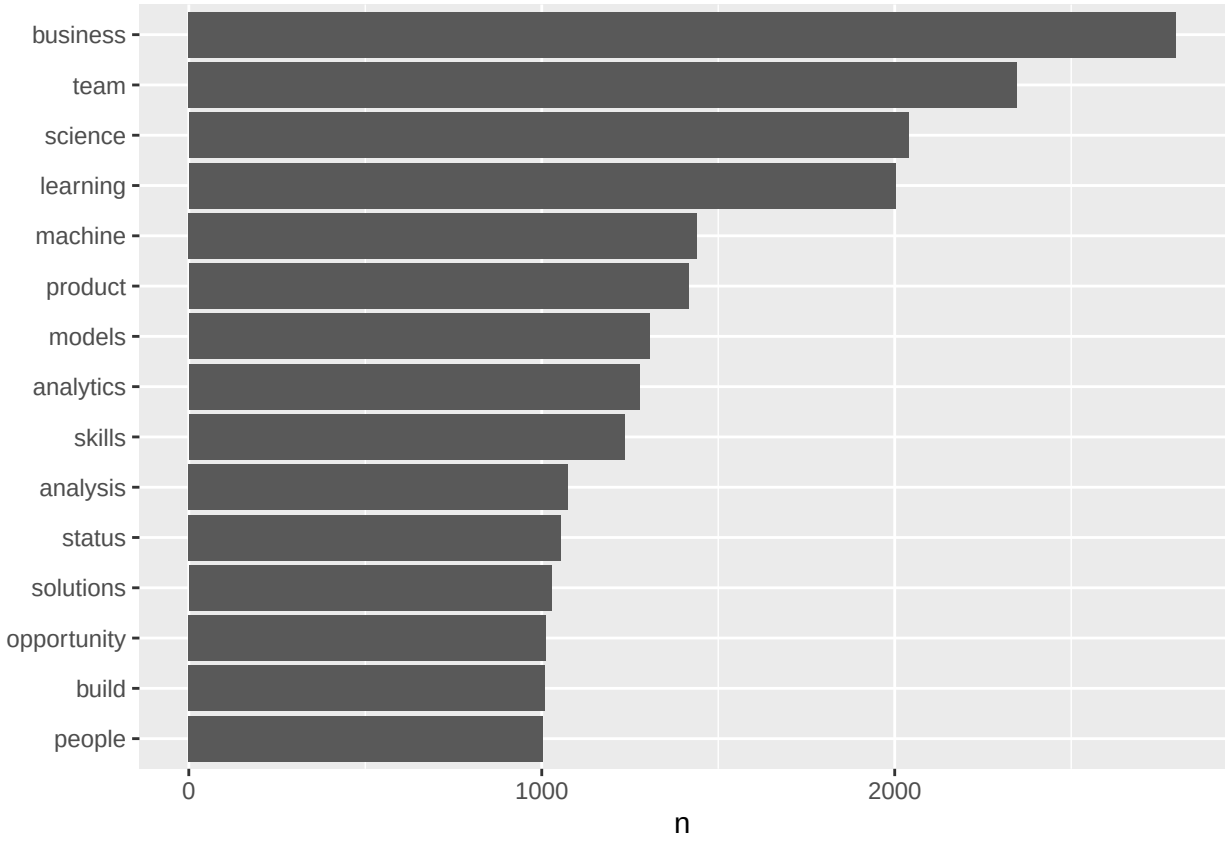
```
## # A tibble: 67 x 2
##    region                               n
##    <chr>                            <int>
##  1 New York, NY                       376
##  2 United States                      277
##  3 New York City Metropolitan Area     31
##  4 Jersey City, NJ                     16
##  5 Princeton, NJ                       10
##  6 New York County, NY                  8
##  7 Newark, NJ                           7
##  8 Poughkeepsie, NY                     6
##  9 Piscataway, NJ                       5
## 10 Secaucus, NJ                         5
## # ... with 57 more rows
```

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

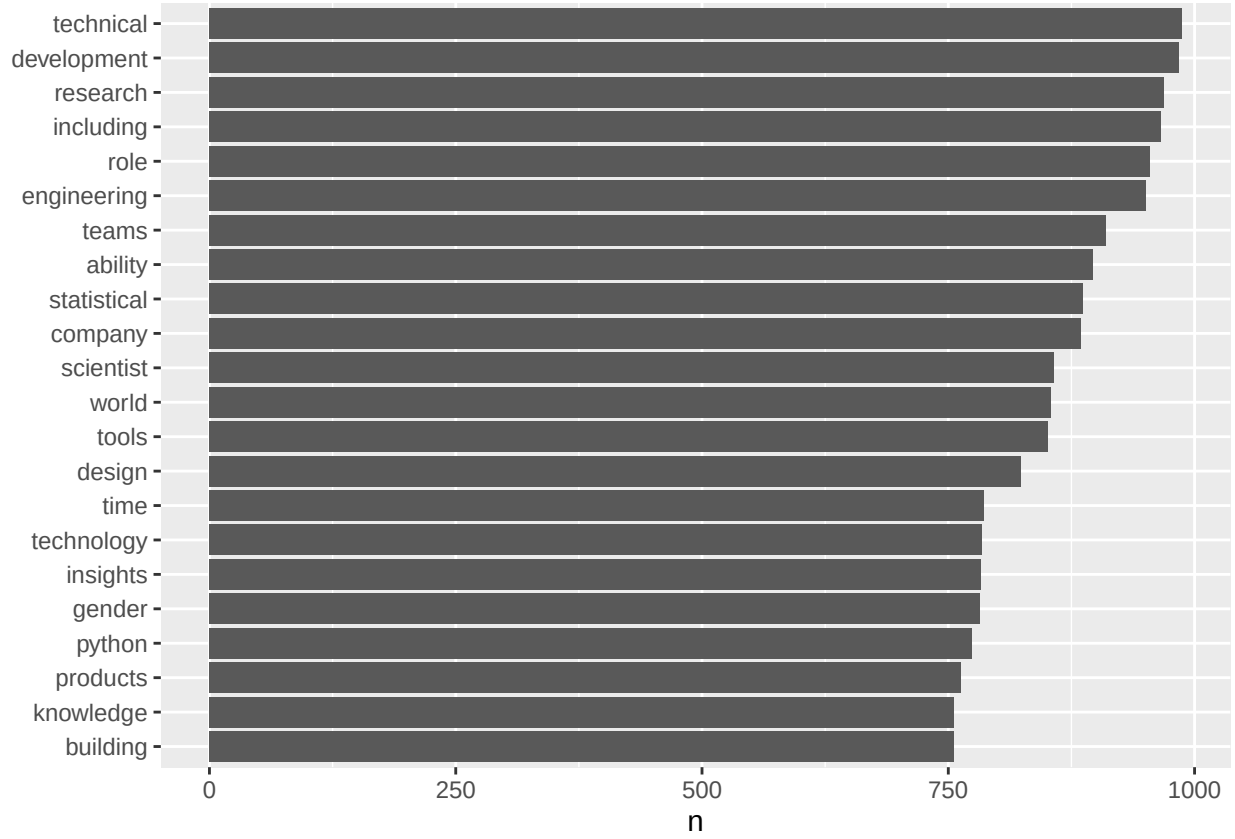
```
## # A tibble: 499 x 2
##    job_title                                        n
##    <chr>                                        <int>
##  1 Data Scientist                                 109
##  2 Senior Data Scientist                           93
##  3 Lead Data Scientist                             18
##  4 Remote Data Analysis Tutor Jobs                 17
##  5 Principal Data Scientist                        11
##  6 Sr. Data Scientist                              11
##  7 Data Analyst                                     8
##  8 Sr Data Scientist                                5
##  9 Azure Data Scientist, Senior, Tech Consulting    4
## 10 Data Analyst - Tech Consulting Staff             4
## # ... with 489 more rows
```

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.

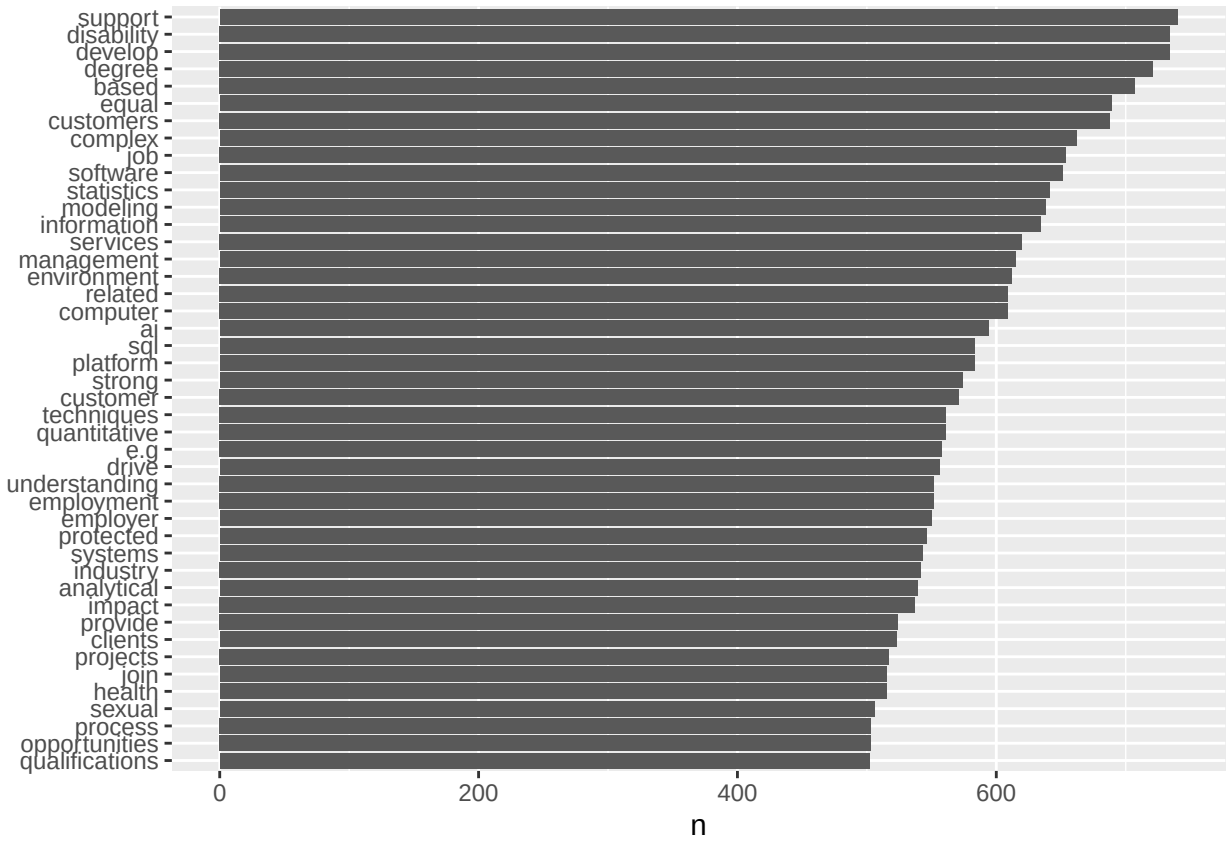
```
library(ggplot2)

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()
```



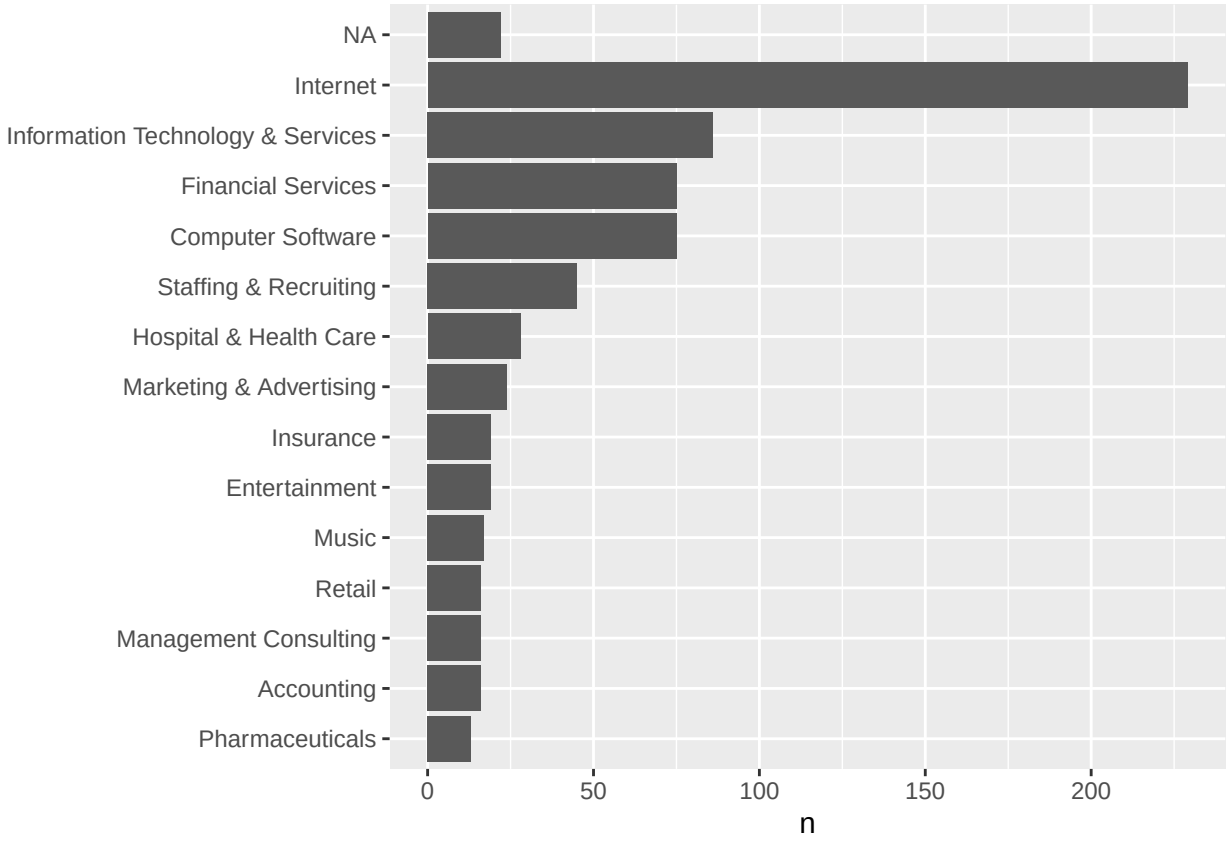
```
jobdat_word %>%
  count(word, sort=TRUE) %>%
  filter(n> 750, n<1000) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n))+geom_col()+xlab(NULL)+coord_flip()
```



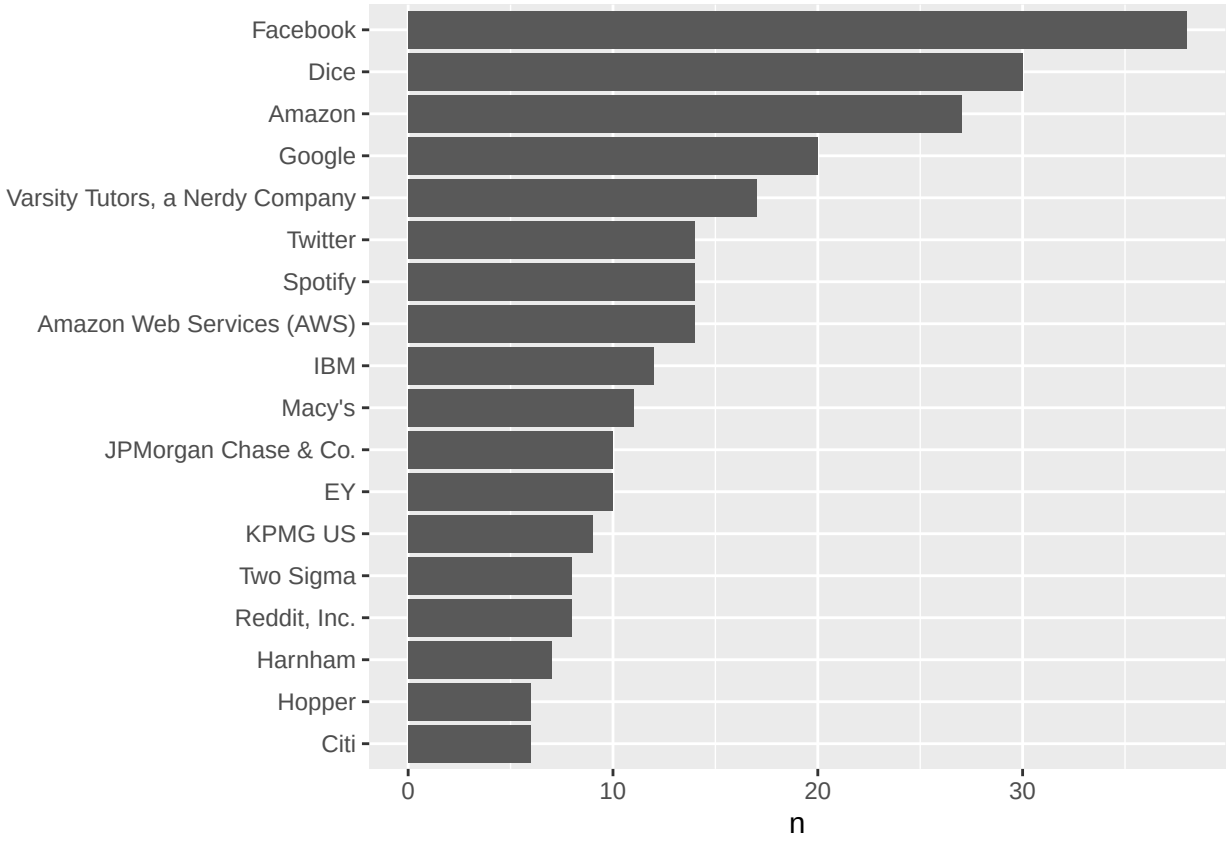

```
jobdat_word %>%
  count(word, sort=TRUE) %>%
  filter(n> 500, n<750) %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n))+geom_col()+xlab(NULL)+coord_flip()
```




```
jobdat %>%
  count(industry, sort=TRUE) %>%
  filter(n> 10) %>%
  mutate(industry=reorder(industry,n)) %>%
  ggplot(aes(industry,n))+geom_col()+xlab(NULL)+coord_flip()
```




```
jobdat %>%
  count(company_name, sort=TRUE) %>%
  filter(n> 5) %>%
  mutate(company_name=reorder(company_name,n)) %>%
  ggplot(aes(company_name,n))+geom_col()+xlab(NULL)+coord_flip()
```



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.

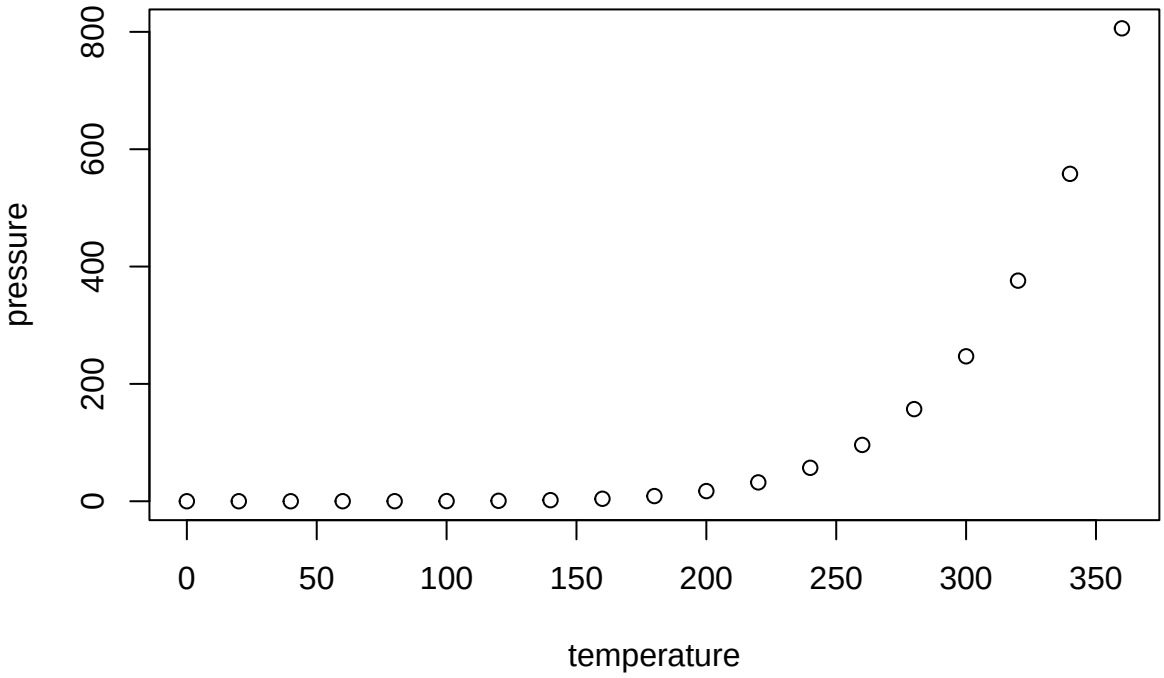
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.