607 Team Project

Most Valuable Data Science Skills

Team Ready 2 Rock

Ready 2 Rock

- Anil Akyildirim
- Nicholas Chung
- Jai Jeffryes
- Tamiko Jenkins
- Joe Rovalino
- Sie Siong Wong

Agenda

- Organization and lifecycle
- Analytical approach
- Data acquisition and preparation
- Conclusions
- Wrap up

Organize

-Slack private channel

- collaboration tool
- asynchronous - group threads and breakout sessions

-Skype

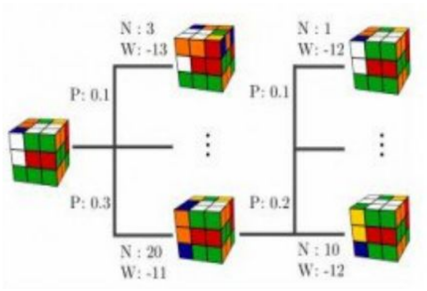
- voice and video communication
- 'putting a name to a face'

-GitHub: Project Github

-Amazon Relational Database Service: MySQL-AWS-Cloud

Lifecycle

- Formed team
- Explored articles - Sources: Towards Data Science, Kdnuggets, Stack Overflow, Kaggle.
- Brainstormed approach - Linkedin, Indeed, Google, Collection Method.
- Collected data -
- Integrated data
- Concluded and reported



Approach

- Assumption: if a data scientist is working, they possess valuable skills.
- Assumption motivated our approach:
    - Sample data scientists on LinkedIn.
    - List of their skills.
    - Count frequencies.
- Visualize and report.

Data collection and preparation

collection

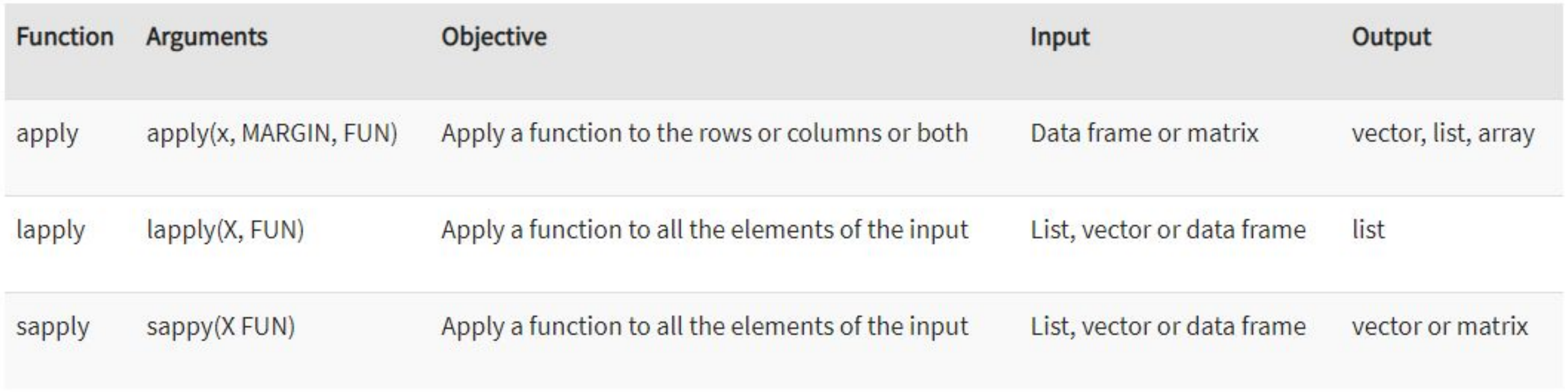
Open source javascript LinkedIn scraper from Github that uses:

- Selenium for automated browser crawling
- Scrapedin for profile scraping

Data collection and preparation

data preparation

1. Accessing relevant nested JSON objects
2. Aggregating JSON objects into dataframes
3. Removing special characters
4. Tidying dataframes



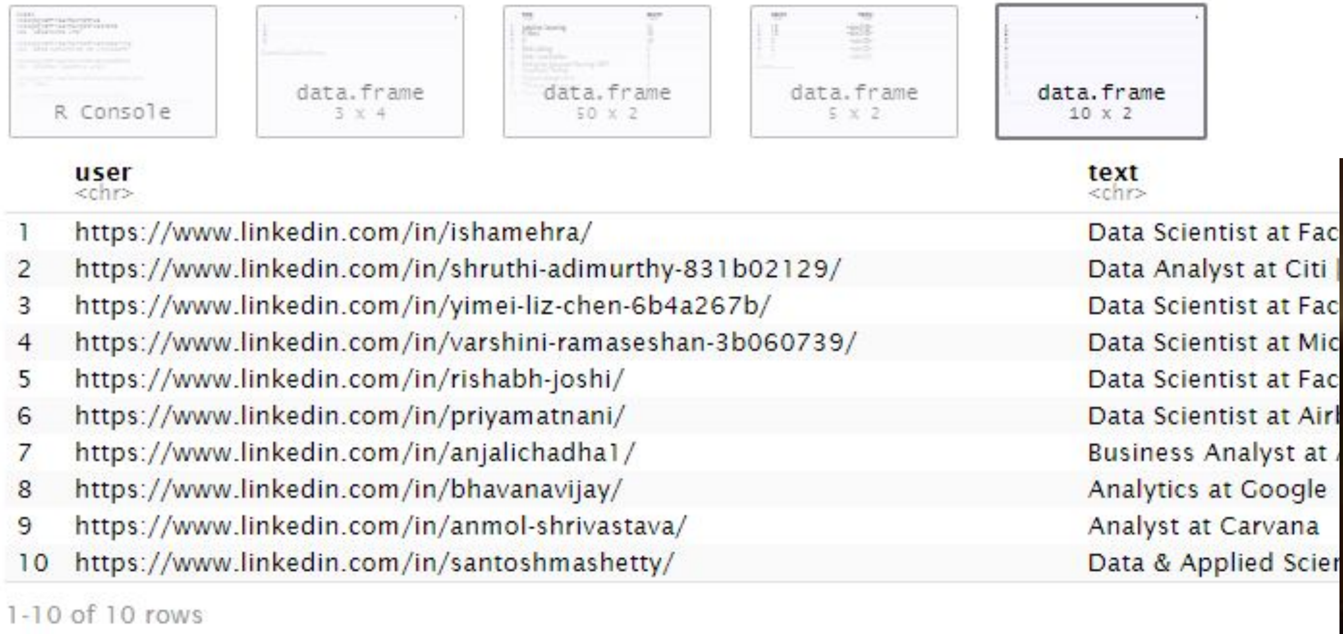
Function	Arguments	Objective	Input	Output
apply	apply(x, MARGIN, FUN)	Apply a function to the rows or columns or both	Data frame or matrix	vector, list, array
lapply	lapply(X, FUN)	Apply a function to all the elements of the input	List, vector or data frame	list
sapply	sappy(X FUN)	Apply a function to all the elements of the input	List, vector or data frame	vector or matrix

https://www.guru99.com/r-apply-sapply-tapply.html

Access

extracting relevant nested JSON objects

```
lapply(list.files("data/profiles", pattern="*.json", full.names=TRUE), function(x) jsonlite::fromJSON(txt = x))
```

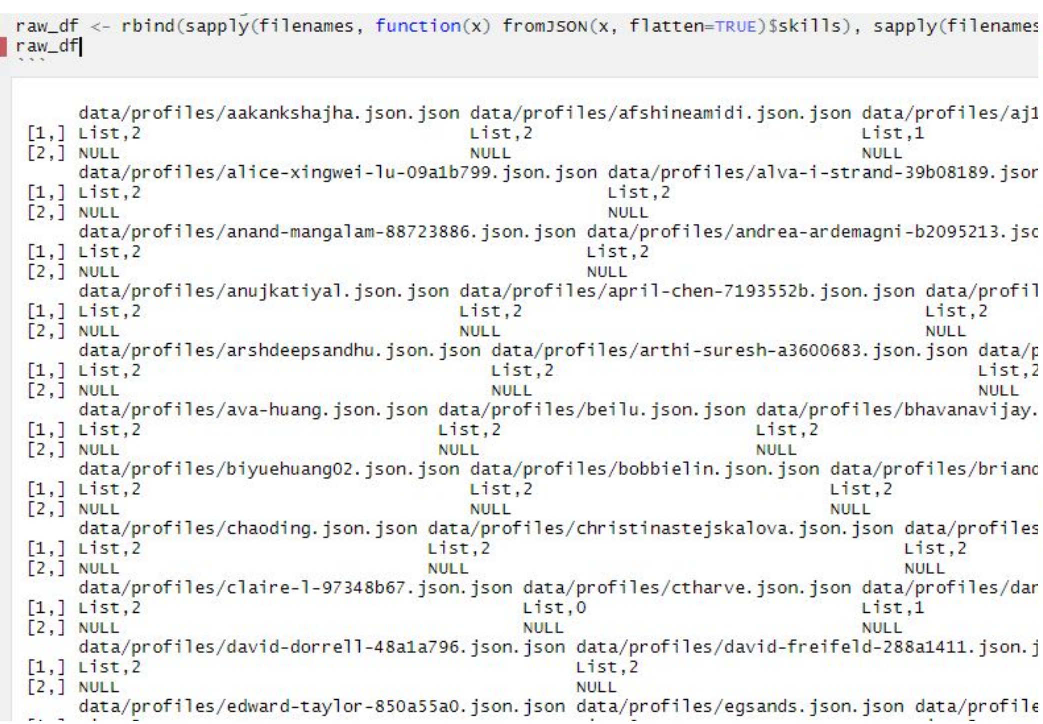


	user <chr>	text <chr>
1	https://www.linkedin.com/in/ishamehra/	Data Scientist at Fac
2	https://www.linkedin.com/in/shruthi-adimurthy-831b02129/	Data Analyst at Citi
3	https://www.linkedin.com/in/yimei-liz-chen-6b4a267b/	Data Scientist at Fac
4	https://www.linkedin.com/in/varshini-ramaseshan-3b060739/	Data Scientist at Mic
5	https://www.linkedin.com/in/rishabh-joshi/	Data Scientist at Fac
6	https://www.linkedin.com/in/priyamatnani/	Data Scientist at Airl
7	https://www.linkedin.com/in/anjalichadha1/	Business Analyst at
8	https://www.linkedin.com/in/bhavanavijay/	Analytics at Google
9	https://www.linkedin.com/in/anmol-shrivastava/	Analyst at Carvana
10	https://www.linkedin.com/in/santoshmashetty/	Data & Applied Scier

1-10 of 10 rows

Aggregating

aggregating JSON objects: binding with rbind



```
raw_df <- rbind(sapply(filenames, function(x) fromJSON(x, flatten=TRUE)$skills), sapply(filenames
raw_df
```

```
     data/profiles/aakankshajha.json.json data/profiles/afshineamidi.json.json data/profiles/aj1
[1,] List,2                               List,2                               List,1
[2,] NULL                                 NULL                                 NULL
     data/profiles/alice-xingwei-lu-09a1b799.json.json data/profiles/alva-i-strand-39b08189.jsor
[1,] List,2                                            List,2
[2,] NULL                                              NULL
     data/profiles/anand-mangalam-88723886.json.json data/profiles/andrea-ardemagni-b2095213.jsc
[1,] List,2                                          List,2
[2,] NULL                                            NULL
     data/profiles/anujkatiyal.json.json data/profiles/april-chen-7193552b.json.json data/profil
[1,] List,2                              List,2                                      List,2
[2,] NULL                                NULL                                        NULL
     data/profiles/arshdeepsandhu.json.json data/profiles/arthi-suresh-a3600683.json.json data/p
[1,] List,2                                 List,2                                           List,2
[2,] NULL                                   NULL                                             NULL
     data/profiles/ava-huang.json.json data/profiles/beilu.json.json data/profiles/bhavanavijay.
[1,] List,2                            List,2                        List,2
[2,] NULL                              NULL                          NULL
     data/profiles/biyuehuang02.json.json data/profiles/bobbielin.json.json data/profiles/briand
[1,] List,2                               List,2                            List,2
[2,] NULL                                 NULL                              NULL
     data/profiles/chaoding.json.json data/profiles/christinastejskalova.json.json data/profiles
[1,] List,2                           List,2                                       List,2
[2,] NULL                             NULL                                         NULL
     data/profiles/claire-l-97348b67.json.json data/profiles/ctharve.json.json data/profiles/dar
[1,] List,2                                    List,0                          List,1
[2,] NULL                                      NULL                            NULL
     data/profiles/david-dorrell-48a1a796.json.json data/profiles/david-freifeld-288a1411.json.j
[1,] List,2                                         List,2
[2,] NULL                                           NULL
     data/profiles/edward-taylor-850a55a0.json.json data/profiles/egsands.json.json data/profile
```

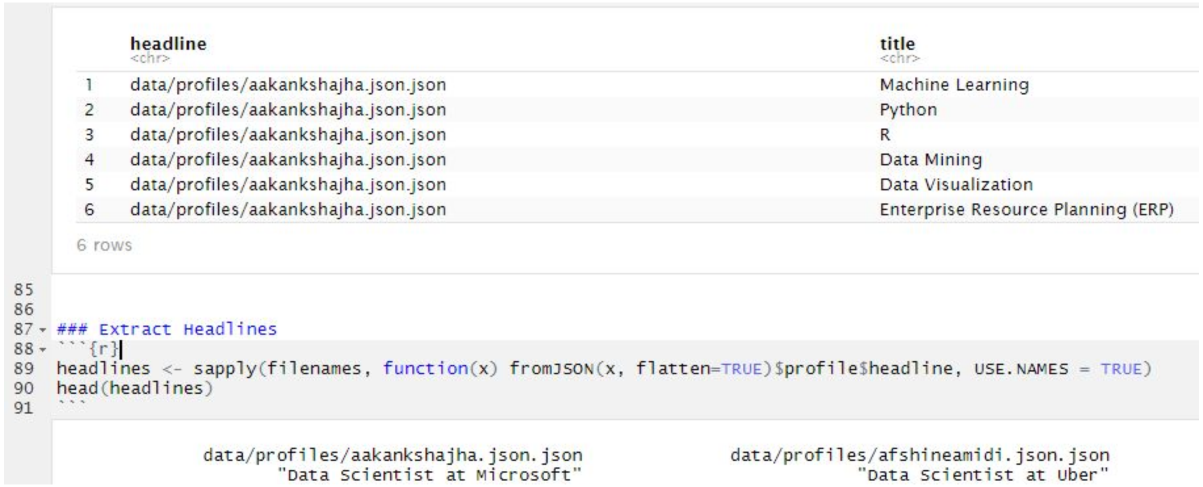
Hadley Wickham @hadleywickham

Follow

People often ask why dplyr & tibble don't support row names. I've (finally) written up my reasons at adv-r.hadley.nz/vectors-chap.h… #rstats (photo credit: @hspter)

Aggregating JSON objects cont.

```
dplyr::bind_rows(
  sapply(filenames, function(x)
    fromJSON(x,
      flatten=TRUE)$skills),
  .id="headline")
```



	headline <chr>	title <chr>
1	data/profiles/aakankshajha.json.json	Machine Learning
2	data/profiles/aakankshajha.json.json	Python
3	data/profiles/aakankshajha.json.json	R
4	data/profiles/aakankshajha.json.json	Data Mining
5	data/profiles/aakankshajha.json.json	Data Visualization
6	data/profiles/aakankshajha.json.json	Enterprise Resource Planning (ERP)

6 rows

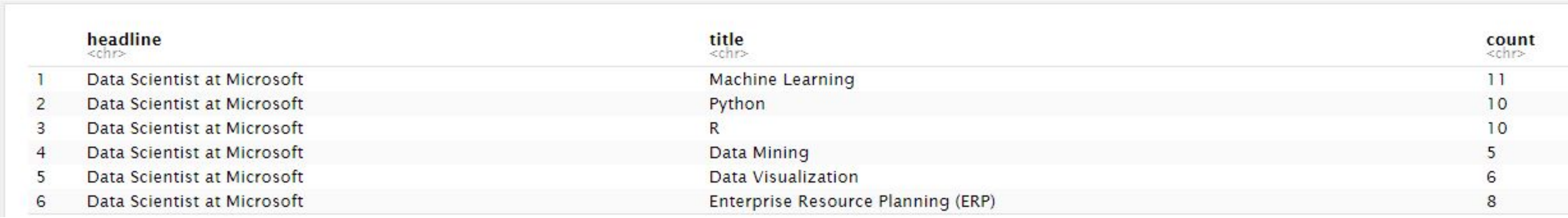
~~~

Extract Headlines
```{r}
headlines <- sapply(filenames, function(x) fromJSON(x, flatten=TRUE)$profile$headline, USE.NAMES = TRUE)
head(headlines)
```
~~~

```
         data/profiles/aakankshajha.json.json             data/profiles/afshineamidi.json.json
                "Data Scientist at Microsoft"                      "Data Scientist at Uber"
```

```
headlines <- sapply(filenames, function(x) fromJSON(x, flatten=TRUE)$profile$headline, USE.NAMES = TRUE)

raw_df$headline <- sapply(raw_df$headline, function(x) headlines[x])
```



	headline <chr>	title <chr>	count <chr>
1	Data Scientist at Microsoft	Machine Learning	11
2	Data Scientist at Microsoft	Python	10
3	Data Scientist at Microsoft	R	10
4	Data Scientist at Microsoft	Data Mining	5
5	Data Scientist at Microsoft	Data Visualization	6
6	Data Scientist at Microsoft	Enterprise Resource Planning (ERP)	8

Remove

Incomplete Data

Counting NA’s

- df_na_942 <- df_conv %>% filter_all(any_vars(is.na(.)))

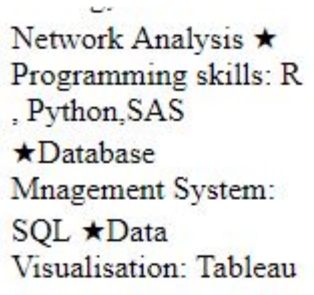
Choosing to Omit

- df_omit <- na.omit(df_conv)

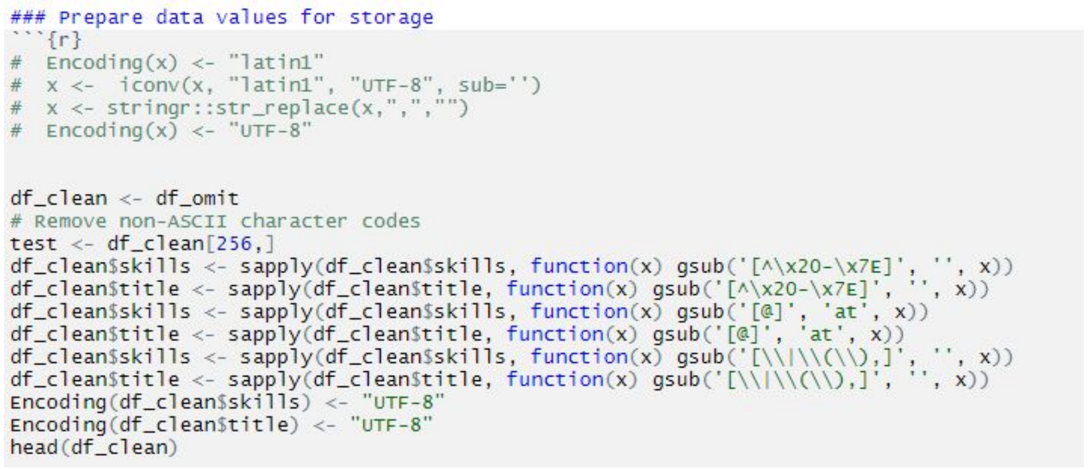
Remove

cleaning special characters

- sapply(df_clean$skills, function(x) gsub('[^\x20-\x7E]', '', x))

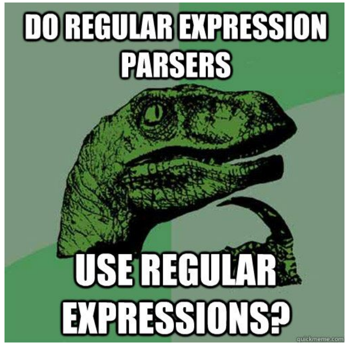


Network Analysis ★
Programming skills: R
, Python,SAS
★Database
Mnagement System:
SQL ★Data
Visualisation: Tableau



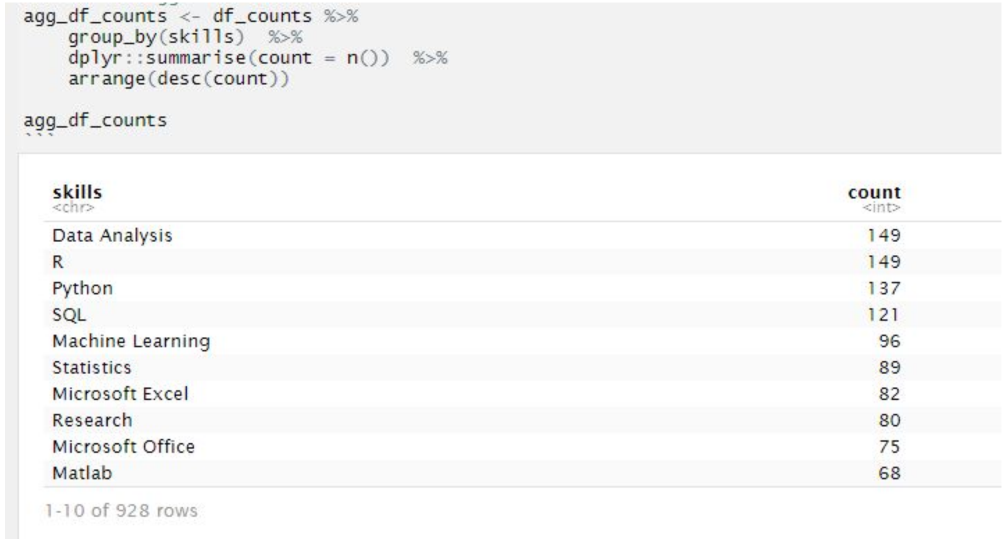
~~~
### Prepare data values for storage
```{r}
#  Encoding(x) <- "latin1"
#  x <-  iconv(x, "latin1", "UTF-8", sub='')
#  x <- stringr::str_replace(x,",","")
#  Encoding(x) <- "UTF-8"


df_clean <- df_omit
# Remove non-ASCII character codes
test <- df_clean[256,]
df_clean$skills <- sapply(df_clean$skills, function(x) gsub('[^\x20-\x7E]', '', x))
df_clean$title <- sapply(df_clean$title, function(x) gsub('[^\x20-\x7E]', '', x))
df_clean$skills <- sapply(df_clean$skills, function(x) gsub('[@]', 'at', x))
df_clean$title <- sapply(df_clean$title, function(x) gsub('[@]', 'at', x))
df_clean$skills <- sapply(df_clean$skills, function(x) gsub('[\\|\\(\\),]', '', x))
df_clean$title <- sapply(df_clean$title, function(x) gsub('[\\|\\(\\),]', '', x))
Encoding(df_clean$skills) <- "UTF-8"
Encoding(df_clean$title) <- "UTF-8"
head(df_clean)
~~~



Tidying

Tidying dataframes with dplyr

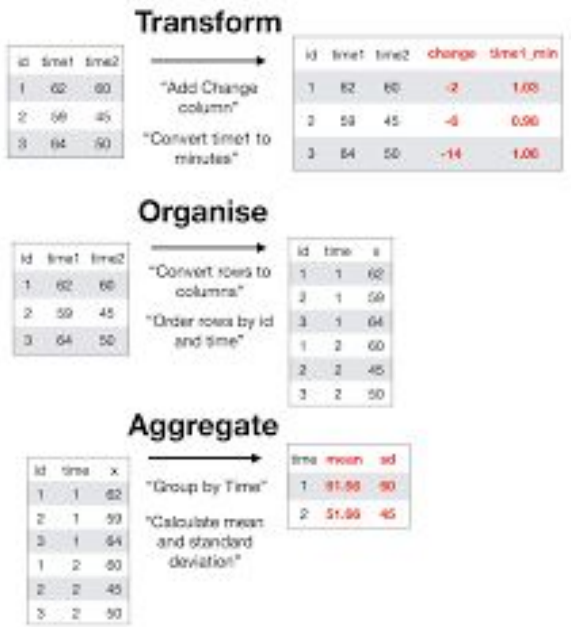


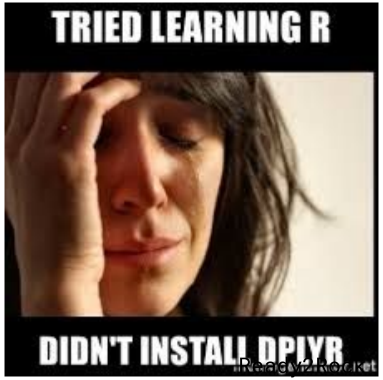
```
agg_df_counts <- df_counts %>%
    group_by(skills)  %>%
    dplyr::summarise(count = n())  %>%
    arrange(desc(count))

agg_df_counts
```

skills <chr>	count <int>
Data Analysis	149
R	149
Python	137
SQL	121
Machine Learning	96
Statistics	89
Microsoft Excel	82
Research	80
Microsoft Office	75
Matlab	68

1-10 of 928 rows

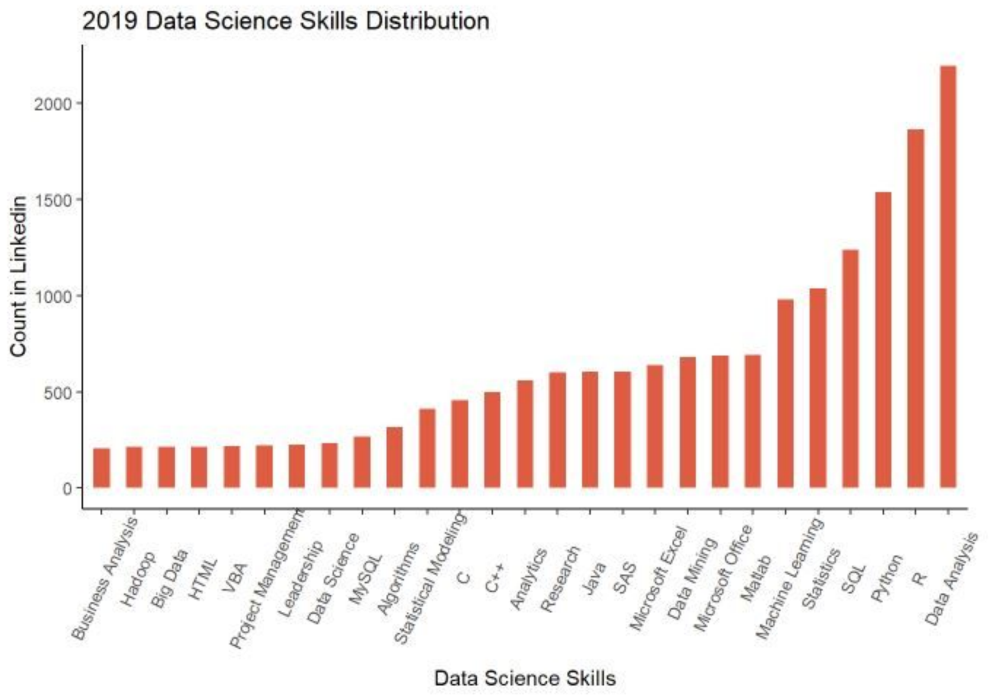

Transform
"Add Change column"
"Convert time1 to minutes"
Organise
"Convert rows to columns"
"Order rows by id and time"
Aggregate
"Group by Time"
"Calculate mean and standard deviation"



Data integration

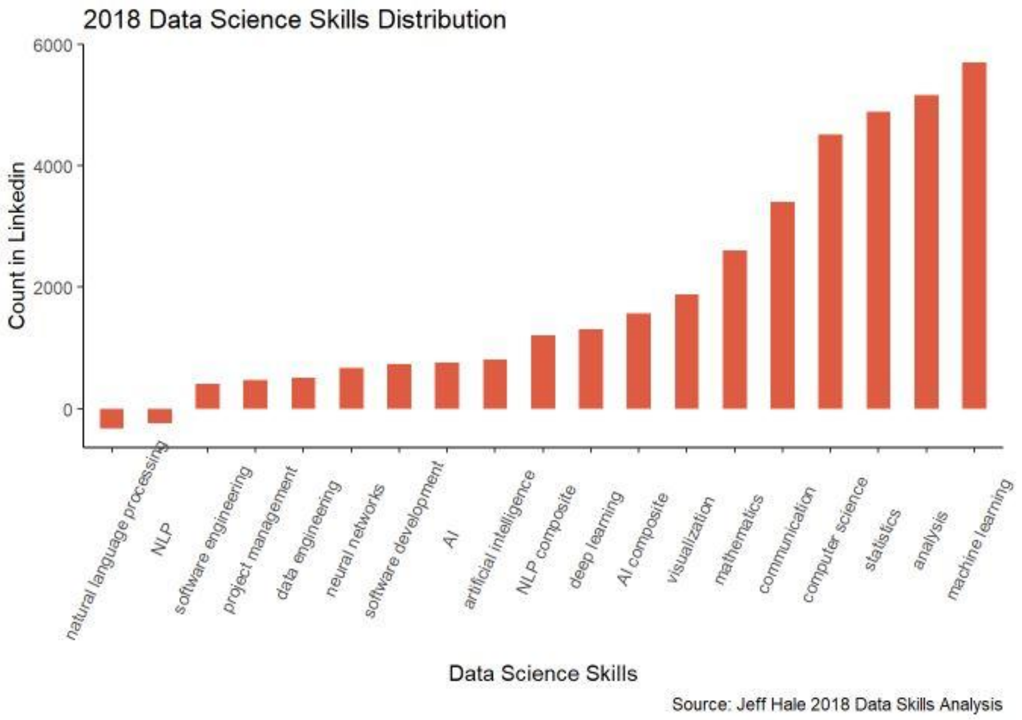
- Data prep.
- Stage in local MySQL.
- Load integration MySQL.

Analysis and Conclusion



Data Science Skills- Data Analysis, Statistics ML, R, Python, SQL

Jeff Hale 2018 Analysis

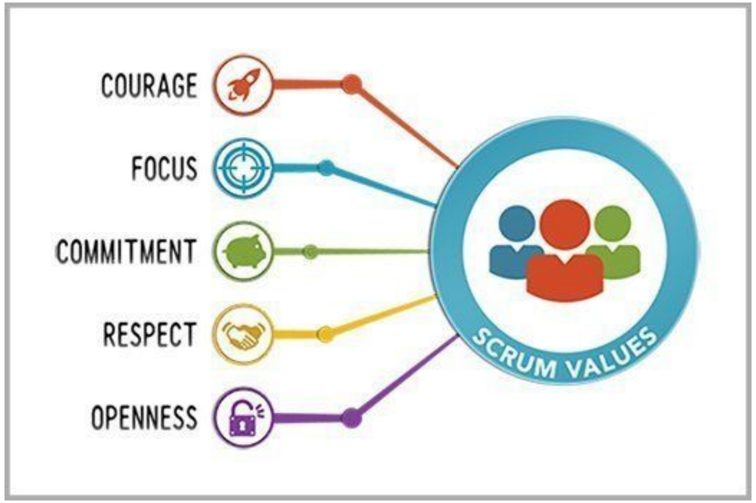


Top General Data Science Skills are same for 2018 Jeff Hale's and our Analysis.

Ready2Rock

Success factors

- Self-organization
- Open communication
- Containment of scope



Version 2.0

Next questions. (What else we could have explored.)

- PayScale: Website.

Further research questions on 2018 (Jeff Hale) vs team 2019 data