

# Profession Survey

Moksha Shah

February 9, 2019

## Introduction

This is an analysis of the responses to kaggle's 2017 user survey. In total, the people responded to enough of the survey to be analyzed. The following data analysis is just one approach to the data.

## Data Source

All data was collected by [Kaggle](#)

## Installing Necessary Packages

```
#install.packages("tidyverse")
#install.packages("tidyverse", repos = "https://cran.rstudio.com")
#install.packages("car")

# For Data Cleaning
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0      v purrr  0.2.5
## v tibble  2.0.1      v dplyr  0.8.0.1
## v tidyr   0.8.2      v stringr 1.3.1
## v readr   1.3.1      v forcats 0.3.0
```

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

```
library(dplyr)
library(rlang)
```

```
##
## Attaching package: 'rlang'
```

```
## The following objects are masked from 'package:purrr':
##
##   %@%, %||%, as_function, flatten, flatten_chr, flatten_dbl,
##   flatten_int, flatten_lgl, invoke, list_along, modify, prepend,
##   rep_along, splice
```

```
library(stringr)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##   recode
```

```
## The following object is masked from 'package:purrr':
##
##      some
```

## Loading Data

```
#importing Dataset
rawPIdata <- read.csv("ProfessionInformation.csv", stringsAsFactors = T, header = T)
rawFFdata = read.csv("freeformResponses.csv", stringsAsFactors = F, header = T)
rawSdata= read.csv("schema.csv", stringsAsFactors = F, header = T)
```

```
# Number of rows
nrow(rawPIdata)
```

```
## [1] 16684
```

```
ncol(rawPIdata)
```

```
## [1] 228
```

The data looks very clumsy, but to ensure that we keep our raw data un-touched, we'll create a duplicate dataframe called "cleanPIdata".

```
cleanPIdata <- rawPIdata
```

For functions, the only arguments are the question number and the option to feed in filtered data if necessary.

## Function for single choice questions

```
# A function to analyze questions where you choose only one answer
chooseOne = function(question, filteredData = cleanPIdata){

  filteredData %>%
    # Remove any rows where the respondent didn't answer the question
    filter(!UQ(sym(question)) == "") %>%
    # Group by the responses to the question
    group_by(question) %>%
    # Count how many respondents selected each option
    summarise(count = n()) %>%
    # Calculate what percent of respondents selected each option
    mutate(percent = (count / sum(count)) * 100) %>%
    # Arrange the counts in descending order
    arrange(desc(count))

}
```

## Demographics

### Current Residence

```
##                                     Question
## 1 Select the country you currently live in.
```

```
residence <- chooseOne("Country")
```

```
## Warning: group_by() is deprecated.
## Please use group_by() instead
##
## The 'programming' vignette or the tidyeval book can help you
## to program with group_by() : https://tidyeval.tidyverse.org
## This warning is displayed once per session.
```

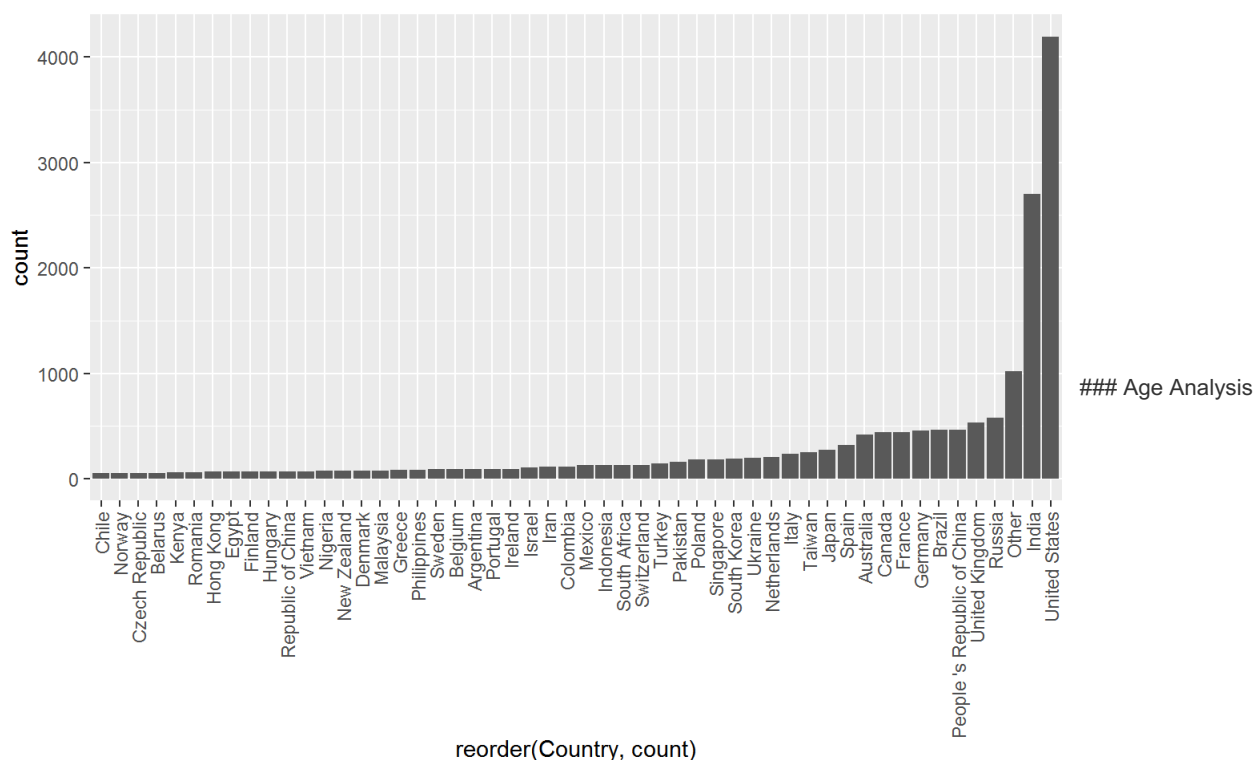
```
residence
```

```
## # A tibble: 52 x 3
##   Country                count percent
##   <fct>                  <int>   <dbl>
## 1 United States          4196    25.3
## 2 India                  2699    16.3
## 3 Other                  1020     6.16
## 4 Russia                 578     3.49
## 5 United Kingdom         535     3.23
## 6 People 's Republic of China 466     2.81
## 7 Brazil                 465     2.81
## 8 Germany                 459     2.77
## 9 France                  442     2.67
## 10 Canada                 440     2.66
## # ... with 42 more rows
```

(only countries with more than 20 people are displayed)

```
residenceFilter <- residence %>%
  filter(count >= 20)

ggplot(residenceFilter, aes(x = reorder(Country, count), y = count)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90,
                                    vjust = 0.5,
                                    hjust = 1))
```



```
##           Question
## 1 What's your age?
```

```
# This column needs to be read as numbers
cleanPIData$Age <- as.numeric(cleanPIData$Age)

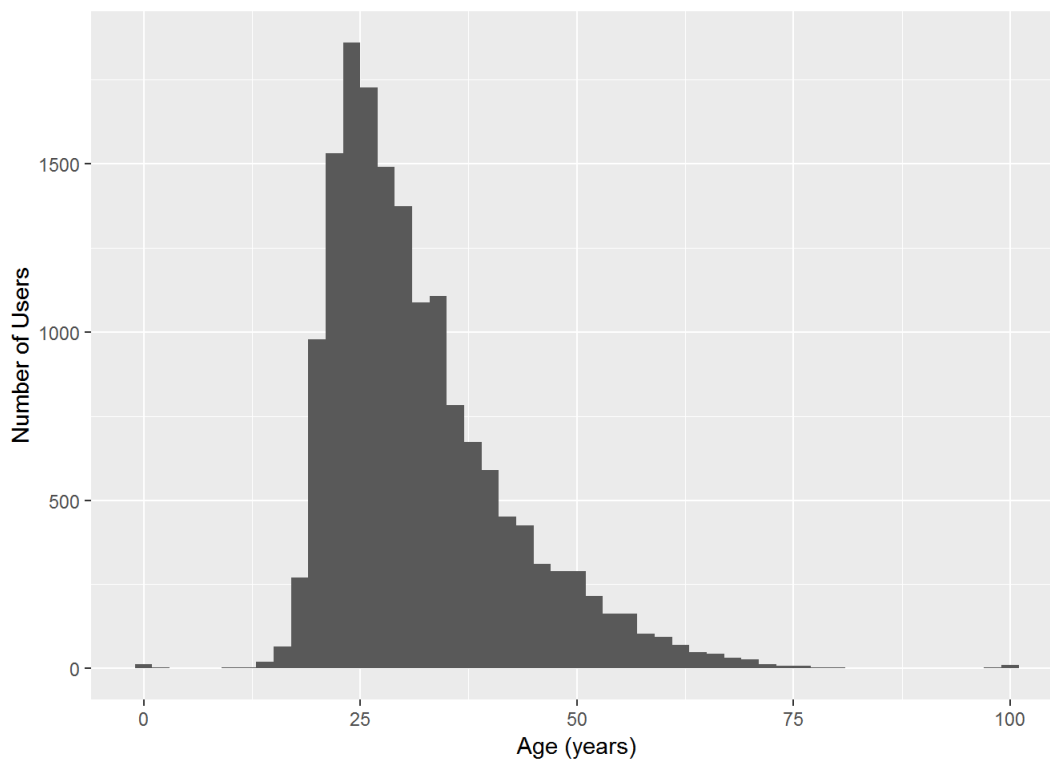
age <- chooseOne("Age") %>%
  # Remove values < 1 year
  filter(!Age < 1)
age
```

```
## # A tibble: 83 x 3
##   Age count percent
##   <dbl> <int>   <dbl>
## 1    25   964    5.89
## 2    24   895    5.47
## 3    26   885    5.41
## 4    27   840    5.14
## 5    23   838    5.12
## 6    30   776    4.74
## 7    28   759    4.64
## 8    29   731    4.47
## 9    22   693    4.24
## 10   31   597    3.65
## # ... with 73 more rows
```

What is the age distribution of users?

```
agedata <- cleanPIData %>%
  # Remove any rows where the respondent didn't answer the question
  filter(!Age == "") %>%
  select(Age)

ggplot(agedata, aes(x = Age)) +
  geom_histogram(binwidth = 2) +
  xlab("Age (years)") +
  ylab("Number of Users")
```



The vast majority of Kaggle users are young adults (early 20's to 30's).

```

top5 <- residence %>%
  # add a row number to each row

mutate(row = row_number()) %>%
  # select only the top 5 countries
  filter( row <= 5) %>%
  # keep only the country name column
  select(Country) %>%
  # change these to character elements, instead of factors
  mutate(Country = as.character(Country))

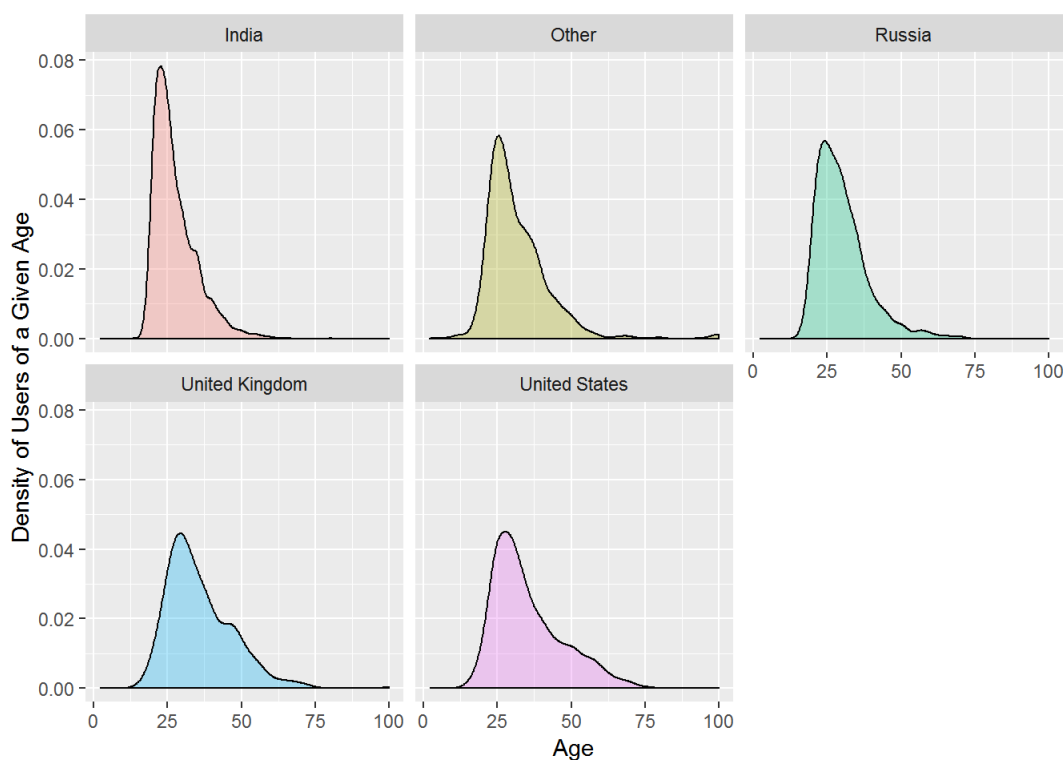
# Create a list of the top 5 countries
top5List <- top5$Country

top5Age <- cleanPIData %>%
  # Keep only entries whose country is included in the top 5 list

  filter(Country %in% top5List) %>%
  # Remove any ages that are under a year or NA or blank
  filter(Age > 1,
         !is.na(Age)) %>%
  filter(!Age == "") %>%
  # Group the data by country and then age
  group_by(Country, Age)

ggplot(top5Age, aes(x = Age, fill = Country)) +
  geom_density(alpha = 0.3) +
  facet_wrap(~Country) +
  ylab("Density of Users of a Given Age") +
  theme(legend.position="none")

```



there's a wider age-range of users in the US and UK.

## Employment Status

```

##                                     Question
## 1 What's your current employment status?

```

```
## # A tibble: 7 x 3
##   EmploymentStatus      count percent
##   <fct>              <int>   <dbl>
## 1 Employed full-time    10878   65.2
## 2 Not employed, but looking for work    2106   12.6
## 3 Independent contractor, freelancer, or self-employed    1326    7.95
## 4 Not employed, and not looking for work     923    5.53
## 5 Employed part-time      914    5.48
## 6 I prefer not to say      419    2.51
## 7 Retired                118    0.707
```

About 65% of the 16,716 users who answered this question are currently employed full-time, while 12.6% are unemployed and looking for work. Nearly 8% of respondents consider themselves self-employed or freelancers.

## Career Profile (Non-Workers)

### Student Status

```
##                                     Question
## 1 Are you currently enrolled as a student at a degree granting school?
```

```
## # A tibble: 2 x 3
##   StudentStatus count percent
##   <fct>         <int>   <dbl>
## 1 Yes           979    76.6
## 2 No            299    23.4
```

76% are currently in degree-granting schools.

### Learning Data Science

```
##                                     Question
## 1 Are you currently focused on learning data science skills either formally or informally?
```

```
## # A tibble: 3 x 3
##   LearningDataScience      count percent
##   <fct>              <int>   <dbl>
## 1 Yes, I'm focused on learning mostly data science skills    799    62.3
## 2 Yes, but data science is a small part of what I'm focused ~   428    33.4
## 3 No, I am not focused on learning data science skills        55     4.29
```

## Career Profile (Workers)

### Job Tasks

```
##                                     Question
estion
## 1 Do you write code to analyze data in your current job, freelance contracts, or most recent job if retired?
```

```
## # A tibble: 2 x 3
##   CodeWriter count percent
##   <fct>         <int>   <dbl>
## 1 Yes          10133    77.0
## 2 No           3028    23.0
```

So 77% of employed Kaggle users write code in their current job.

```
##                                     Question
## 1 Are you actively looking to switch careers to data science?
```

```
## # A tibble: 2 x 3
##   CareerSwitcher count percent
##   <fct>          <int>   <dbl>
## 1 Yes             2121    70.5
## 2 No              886    29.5
```

70% of the employed Kaggle users that don't currently write code in their job are planning to switch into a data science field.

## Job Titles

```
questionText("CurrentJobTitleSelect")
```

```
##
Question
## 1 Select the option that's most similar to your current job/professional title (or most recent title if r
etired). - Selected Choice
```

```
chooseOne("CurrentJobTitleSelect")
```

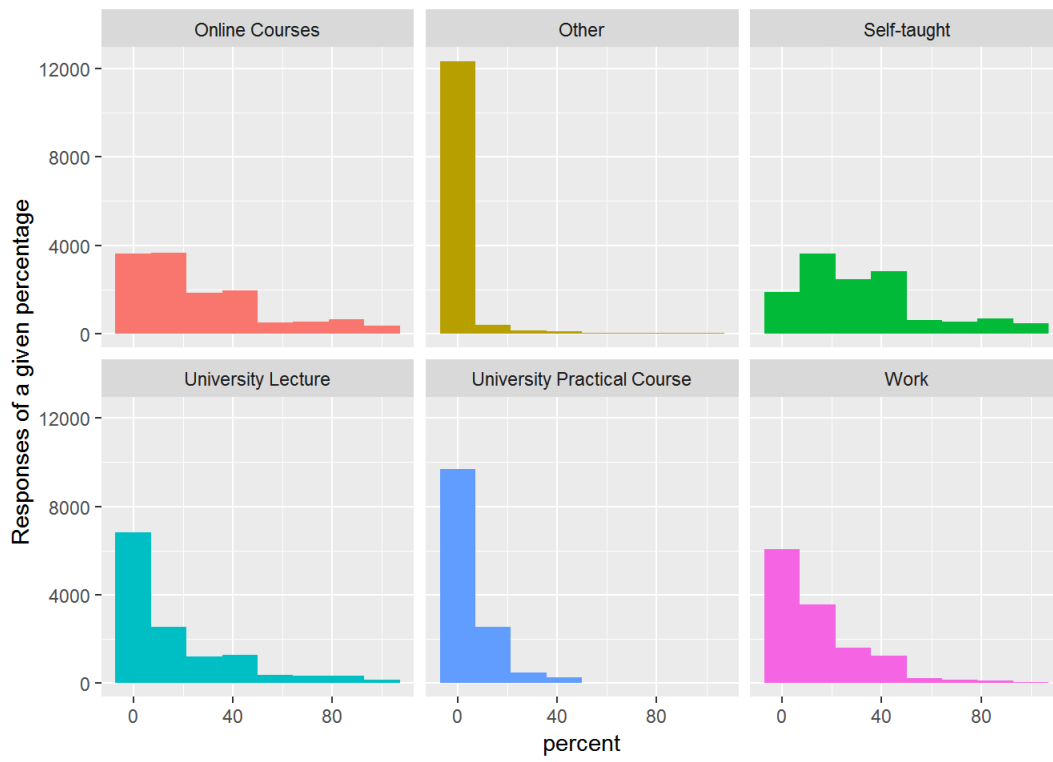
```
## # A tibble: 16 x 3
##   CurrentJobTitleSelect count percent
##   <fct>                <int>   <dbl>
## 1 Data Scientist      2430    20.6
## 2 Software Developer/Software Engineer 1758    14.9
## 3 Other                1230    10.4
## 4 Data Analyst        1208    10.2
## 5 Scientist/Researcher   977     8.27
## 6 Business Analyst      793     6.72
## 7 Researcher           619     5.24
## 8 Machine Learning Engineer   617     5.22
## 9 Engineer            552     4.67
## 10 Programmer          459     3.89
## 11 Computer Scientist    335     2.84
## 12 Statistician         288     2.44
## 13 DBA/Database Engineer  186     1.58
## 14 Predictive Modeler    181     1.53
## 15 Data Miner           118     0.999
## 16 Operations Research Practitioner   58     0.491
```

About 45% of Kaggle users are either Data Scientists, Software Developers/Engineers or Data Analysts. Predictive Modeler, Data Miner, and Operations Research Practitioner are among the least common job titles.

## training in each category

```
training <- cleanPIData %>%
  # Keep only the columns that start with "LearningCategory" and don't include "FreeForm"
  select(starts_with("LearningCategory"), -contains("FreeForm")) %>%
  # Set column names
  purrr::set_names(c("Self-taught", "Online Courses", "Work", "University Lecture", "University Practical Co
urse", "Other")) %>%
  # Re-structure the data
  gather(key = response, value = percent) %>%
  # Remove any rows where the percentage was NA
  filter(!is.na(percent)) %>%
  # Change the percentage column to a number
  mutate(percent = as.numeric(percent))

ggplot(training, aes(x = percent, fill = response)) +
  geom_histogram(bins = 8) +
  facet_wrap(~response) +
  ylab("Responses of a given percentage") +
  theme(legend.position="none")
```



Online courses and self-

teaching seem to have the widest range of percentages reported