*REPATS, Brasília, V. 4, nº 2, p.613-631, Jul-Dez, 2017*

# HOW MUCH ARE CHARITY, FUNDRAISING, NGO AND NON-PROFITS CURRENTLY PAYING THEIR NEW STAFF?*

## QUAL O CUSTO DA CARIDADE, CAPTAÇÃO DE RECURSOS, ONG E NÃO-LUCRATIVIDADE QUE PAGAM ATUALMENTE O SEU NOVO PESSOAL?



Eduardo Wirthmann Ferreira**

**ABSTRACT:** This article explores current salary levels and some other related questions using public recruitment data from the CharityJobs website. According to CharityJob, the site is the United Kingdom's busiest one for charity, fundraising, NGO and not for profit jobs. Data collection took place between 4 September and 20 November 2016 with some basic techniques of web scraping (web harvesting or web data extraction), which is a computer software technique of extracting information from websites. All the process is documented at: https://rpubs.com/EduardoWF/charityjobs. The source code in RMarkdown is available for download following GNU General Public License. Everything was prepared with the open-source and freely-accesible statistical computing software R (R version 3.2.0 - http://cran.r-project.org/) and the IDE RStudio (Version 0.99.441 - http://www.rstudio.com/). In addition to presenting these powerful tools and data-exploring techniques, I hope that this article can help the public, specially applicants and workers in civil-society organisations to get an update on salaries and trends in the sector. The jobs analysed here are mostly UK-based ones and published by UK-based organisations. Therefore, the results are not meant to represent the entire sector worldwide. I still hope though that this analysis can provide some positive contribuition to the evolution of work of civil-society organisation in both the southern and the northern hemispheres. This post is based on public data, is my sole responsibility and can in no way be taken to reflect the views of CharityJobs' staff.

* Artigo recebido em: 08.04.2017

Artigo aceito em 10.12.2017

** PhD in Development Economic. Master of Arts in Development Policy with focus on non-governmental organizations. Bachelor of Arts in International Relations. Guest lecturer for project management of the University of Applied Sciences of Bremen, data scientist, consultant, evaluator and facilitator in designing, managing and evaluating projects and programs in Africa, Asia, Europe, Central and South America for non-governmental organizations, governments, consultancy firms, research institutions and international organizations. Bremen, Alemanha. E-mail: eduardo@wferreira.eu

**Keywords:** Charity, Fundraising, NGO.

**RESUMO:** Este artigo explora os níveis salariais atuais e algumas outras questões relacionadas usando dados de recrutamento público do site CharityJobs. De acordo com a CharityJob, o site é o mais movimentado do Reino Unido para caridade, captação de recursos, ONG e não para fins lucrativos. A coleta de dados ocorreu entre 4 de setembro e 20 de novembro de 2016 com algumas técnicas básicas de web scraping (web harvesting ou web data extraction), que é uma técnica de software de extração de informações de websites. Todo o processo está documentado em: https://rpubs.com/EduardoWF/charityjobs. O código fonte no RMarkdown está disponível para download após a Licença Pública Geral GNU. Tudo foi preparado com o software de computação estatístico de código aberto e de acesso livre R (R versão 3.2.0 - http://cran.r-project.org/) e o IDE RStudio (Versão 0.99.441 - http: // www.rstudio.com/). Além de apresentar essas poderosas ferramentas e técnicas de exploração de dados, espero que este artigo possa ajudar o público, especialmente candidatos e trabalhadores de organizações da sociedade civil a obter uma atualização sobre salários e tendências no setor. Os trabalhos aqui analisados são, em sua maioria, baseados no Reino Unido e publicados por organizações sediadas no Reino Unido. Portanto, os resultados não pretendem representar todo o setor em todo o mundo. Ainda espero, porém, que esta análise possa fornecer alguma contribuição positiva para a evolução do trabalho da organização da sociedade civil nos hemisférios sul e norte. Este post é baseado em dados públicos, é de minha responsabilidade exclusiva e não pode de forma alguma ser tomado para refletir as opiniões do pessoal da CharityJobs.

**Palavras-chave:** Caridade, Captação de Recursos, ONG.



## INTRODUCTION

Increasing computing power of personal devices as well as open-source data-science tools have made a new wave of analysis possible. Data science employs statistics and computation to process and analyse qualitative and quantitative data at large scale. According to the United Nation's Global Pulse (2016), "the use of data analytics to inform and implement smart, agile and adaptive projects and programmes has passed beyond the inflection point and is now accelerating within development and humanitarian practice"[1].

Integration of "big data" into monitoring and evaluation of development programmes, particularly among civil-society organisations, is still lagging behind. Most of the applications of big data in international development do not currently focus directly on monitoring, and even less on evaluation[2].

[1] Kirkpatrick, Robert. in: UN Global Pulse (2016), Integrating Big Data into the Monitoring and Evaluation of Development Programmes. Available at: http://unglobalpulse.org/ (last access: 4 October 2017).
[2] UN Global Pulse (2016), Integrating Big Data into the Monitoring and Evaluation of Development Programmes. Available at: http://unglobalpulse.org/ (last access: 4 October 2017).

This article presents an example of an application based on web scrapping for monitoring salaries among civil-society organisations. The main question pursued here was: How much are charity, fundraising, NGO and non-profits currently paying their new staff? In this article I explored this and some other related questions using public recruitment data from the CharityJobs website. According to CharityJob, the site is the United Kingdom's busiest one for charity, fundraising, NGO and not for profit jobs. This article is based on public data, is my sole responsibility and can in no way be taken to reflect the views of CharityJobs' staff.



In addition to presenting these powerful open-source tools and data-exploring techniques, I hope that this post can help the public, specially applicants and workers in development aid organisations to get an update on salaries and trends in the sector. The jobs analysed here are mostly UK-based ones and published by UK-based organisations. Therefore, the results are not meant to represent the entire sector worldwide. I still hope though that this post can provide some positive contribuition to the evolution of development aid work in both the southern and the northern hemispheres.

For readers who are only interested in the end analysis, please jump to the results section. However, I encourage you to explore how these tools work. I believe that they can help speeding up and improving quality of the so-much-needed charity, social-enterprise, development-aid and humanitarian work globally.

I used here some basic techniques of web scraping (web harvesting or web data extraction), which is a computer software technique of extracting information from websites. The source code in RMarkdown is available for download and use based on GNU General Public License. All the process is documented at: https://rpubs.com/EduardoWF/charityjobs. Everything was prepared with the open-source and freely-accesible statistical computing software R (R version 3.2.0 - http://cran.r-project.org/) and the IDE RStudio (Version 0.99.441 - http://www.rstudio.com/).

## 1 DOWNLOADING DATA FROM CHARITYJOBS

Using RStudio, the first step was to download the website data. CharityJobs' search engine contains over 140 webpages, each of them with a list of 18 jobs in most cases. Hence I expected to get information about around 2,500 job announcements. For that, the first step was to download the data and get rid of what I did not wanted (e.g. css and html codes). The code chunck below describes how I did it. The code contains explanatory comments indicated by hashtags ('#').

Data was collected in multiple dates between 4 September and 20 November 2016 using the code presented below and available at:

https://rpubs.com/EduardoWF/charityjobs. After the data-collection period, CharityJobs' search engine went through an update which will demand adjustments in the code if readers would like to collect data again directly from their website.



```
#  Loading the necessary packages.

   # Credits to Hadley Wickham (2016)

suppressWarnings(suppressPackageStartupMessages(require(rv
est)))

   # Credits to Hadley Wickham (2015)
   suppressPackageStartupMessages(require(stringr))

   # Credits to Garrett Grolemund, Hadley Wickham (2011)
   suppressPackageStartupMessages(require(lubridate))

   # Credits to Hadley Wickham and Romain Francois (2015)
   suppressPackageStartupMessages(require(dplyr))

   # Credits to Hadley Wickham (2015)
   suppressPackageStartupMessages(require(xml2))

   # Credits to Gergely Daróczi and Roman Tsegelskyi (2015)
   suppressPackageStartupMessages(require(pander))

   # Credits to Hadley Wickham (2009)
   suppressPackageStartupMessages(require(ggplot2))

   # Credits to Hadley Wickham, Jim Hester and Romain Francois
(2016)
   suppressPackageStartupMessages(require(readr))

   # Credits to Ian Fellows (2014)
   suppressPackageStartupMessages(require(wordcloud))

   # Credits to Erich Neuwirth (2014)
   suppressPackageStartupMessages(require(RColorBrewer))

   # Credits to Ingo Feinerer and Kurt Hornik (2015)
   suppressPackageStartupMessages(require(tm))

   # Credits to Hadley Wickham (2015)
   suppressPackageStartupMessages(require(stringr))

# Creating list of URLs (webpages)
     urls  <- paste("https://www.charityjob.co.uk/jobs?page=",
```

```
                    seq(1:140), sep = "")

# Downloading website information into a list called `charityjobs`
    charityjobs  <- lapply(urls, . %>% read_html(.))
```

## 2 TYDING UP AND PARSING DATA



The next step was to parse or clean up the text string of each of the about 140 webpages. I decided to build a custom function for that, which I could use to loop through the content of each element of the charityjobs list. The function should also save the parsed data into a data frame. This data frame should include information on recruiters, position titles, salary ranges and deadline data. The code chuck below presents this function, which I called salarydata.

```
# Creating a function for parsing data
salarydata <- function(list) {

 # Creating auxiliary variables and databases
 list_size <- length(list)
 salaries  <- data.frame(deadline=character(),
                         recruiter=character(),
                         position=character(),
                         salary_range=character())

 for (i in seq_along(1:list_size)){
   size <- list[[i]] %>% html_nodes(".salary") %>% html_text() %>%
length()

   #Intermediary dataframe
   sal  <- data.frame(deadline=rep(NA, size),
                      recruiter=rep(NA, size),
                      position=rep(NA, size),
                      salary_range=rep(NA, size))

   ## Filling out intermediary data for deadlines for application
   sal$deadline[1:size]  <- list[[i]] %>%
     html_nodes(".closing:nth-child(4) span") %>% html_text() %>%
     .[!grepl("^[Closing:](*)",.)]  %>% rbind()

   ## Filling out intermediary data for recruiters
   sal$recruiter[1:size]  <- (list[[i]] %>%
                              html_nodes(".recruiter") %>% html_text() %>%
                              gsub("\r\n\ \\s+", "",.) %>%
                              gsub("\r\n", " ", .) %>%
                              gsub("^\\s+|\\s+$", "", .)) %>%
```

```
    rbind()

  ## Filling out intermediary data for positions
  sal$position[1:size]  <-   list[[i]] %>%
    html_nodes(".title") %>% html_text() %>%
    gsub("\r\n\ \\s+", "",.) %>%
    gsub("\r\n", " ", .) %>%
    gsub("^\\s+|\\s+$", "", .) %>%
    rbind()

  ## Filling out intermediary data for salary ranges
  sal$salary_range[1:size]  <- list[[i]] %>%
    html_nodes(".salary") %>% html_text() %>%
    gsub("(£..)\\.", "\\1", .) %>%
    gsub("\\.(.)k(+) |\\.(.)K(+)", "\\100 \\2", .) %>%
    gsub("(*)k(+) |(*)K(+)", "\\1000 \\2", .) %>%
    # Substituting remaining ks
    gsub("k", "000 ", .) %>%
    # Adding thousands for figures withou "k"
    gsub("^(£..)\\-", "\\1000; ", .) %>%
    # Removing pounds signs
    gsub("- £", "; ", .) %>% gsub("-£", "; ", .) %>% gsub("£", "", .)
%>%
    # Removing dashes
    gsub("-", ";", .) %>% gsub("–", ";", .)

  ## Excluding per-hour and per-day jobs
  sal <-  sal %>% filter(!grepl("hours", sal$salary_range))
  sal <-  sal %>% filter(!grepl("hour", sal$salary_range))
  sal <-  sal %>% filter(!grepl("p/h", sal$salary_range))
  sal <-  sal %>% filter(!grepl("week", sal$salary_range))
  sal <-  sal %>% filter(!grepl("ph", sal$salary_range))
  sal <-  sal %>% filter(!grepl("day", sal$salary_range))
  sal <-  sal %>% filter(!grepl("daily", sal$salary_range))
  sal <-  sal %>% filter(!grepl("plus", sal$salary_range))
  sal <-  sal %>% filter(!grepl("\\+", sal$salary_range))

  salaries <- rbind(salaries, sal)

 }
 return(salaries)
}
```

## 3 CREATING FULL DATAFRAME AND OTHER ADJUSTMENTS

The last step before exploring the data was to run the function salarydata to create the full dataframe. After that, I parsed lower and upper salaries into separated columns, deleted data which may have been incorrectly parsed or data concerning daily-rate and hourly-rate jobs / consulting assignments. Only yearly salaries between GBP 4,000 and GBP 150,000 have been considered. All salary data is in British Pounds (GBP) and refer to annual salaries, which sometimes do not include benefits such as pension.

Cleaning the salary-range variable was a tricky step as the website allows users to type in both salary amounts and additional text (e.g. 30,000, 30K, or 25-30k). Therefore, I had to iterate some times until the output was good enough. I am quite sure that the code chunk below can be written in a more elegant / concise way.



```
# Creating a full and clean dataframe
 salaries <- salarydata(charityjobs)

# Parsing salary-range variable
 salaries$salary_range <- gsub('', '', ',', salaries$salary_range) %>%
   gsub(' ; ', ';', .) %>% gsub('; ', ';', .) %>%
   gsub(',[:A-z:]', ' ', .) %>%
   gsub('\\(*', '', .) %>%
   gsub('\\:', '', .) %>%
   gsub('[:A-z:],[:A-z:]', ' ', .) %>%
   gsub('(..),00\\...', '\\1,000', .) %>%
   gsub('(..),0\\...', '\\1,000', .) %>%
   gsub('[A-z]', '', .) %>% gsub(',', '', .) %>%
   gsub('\\.', '', .) %>% gsub('^\\s+', '', .) %>%
   gsub('\\s([[:digit:]])', ';\\1', .) %>%
   gsub('\\s+', '', .) %>% gsub('^[[:digit:]]{1};', '', .) %>%
   # Deleting "(" and ")"
   gsub('\\(', '', .) %>% gsub('\\)', '', .) %>%
   # Deleting "/" and correcting digits
   gsub('\\/', '', .) %>% gsub('000000', '0000', .) %>%
   # Correcting number of digits
   gsub('([[:digit:]]{2})00000', '\\1000', .)  %>%
   # Correcting number of digits
   gsub('([[:digit:]]{5})00', '\\1', .)

# Adjusting data and computing lower and upper salaries using ";" as
separator
 salaries  <- suppressWarnings(salaries %>%
 mutate(upper_salary=gsub('^.*;', '', salaries$salary_range)) %>%
 mutate(lower_salary=gsub(';.*', '', salaries$salary_range)) %>%
 mutate(upper_salary=as.numeric(upper_salary)) %>%
 mutate(lower_salary=as.numeric(lower_salary)) %>%
 filter(upper_salary<150000) %>% filter(upper_salary>4000) %>%
 filter(lower_salary<150000) %>% filter(lower_salary>4000)  %>%
```

```
  mutate(lower_salary=ifelse(lower_salary>=upper_salary, NA,
lower_salary)) %>%
  filter(is.na(upper_salary)!=TRUE) %>% tbl_df() %>%
  select(deadline, recruiter, position,
         lower_salary, upper_salary, salary_range) %>%
  mutate(deadline=dmy(deadline))) %>%
filter(as.Date(deadline)<Sys.Date()+30)

# Removing entries which are too different from most entries in terms
of format
salaries  <- salaries %>%
filter(position!="WELFARE OFFICER The Cinema and Television
Benevolent Fund") %>%
filter(position!="Senior Development Manager (International)
University of Exeter") %>%
filter(position!="Head of communications TMP (UK) Limited") %>%
filter(position!="Programme Partnerships Manager (Flexible location)
Alzheimer's Society") %>%
unique() # Filtering any duplicated entries
```

The code was used to collect data in multiple dates within the period from 4 September and 20 November 2016. The multiple salary datasets have been then merged into one single dataset named dataset.csv which only contain 2,429 unique rows (each row represents a job announcement). In order to ease reproducibility of the analysis, the full dataset is available for download at: http://bit.ly/CharitySalaries in CSV-format (comma-separated values). The output below presents the summary of the full dataframe (10 first observations).

```
# A tibble: 2,429 × 6
    deadline                  recruiter
      <date>                      <chr>
1  2016-09-22           Pancreatic Cancer UK
2  2016-09-20 Samaritan's Purse International
3  2016-09-13             Housing for Women
4  2016-09-15                      INASP
5  2016-09-16                Safer Places
6  2016-10-02              Young Manchester
7  2016-10-03          South Lakeland Carers
8  2016-09-18                  Muslim Aid
9  2016-09-12         St. Catherine's Hospice
10 2016-09-25                      REFUGE
# ... with 2,419 more rows, and 4 more variables: position <chr>,
#   lower_salary <int>, upper_salary <int>, salary_range <chr>
```

## 4 RESULTS

The table below presents the summary statistics concerning the lower and upper salaries. The final dataset contains information of 2,429 jobs of various types, based on yearly-salary figures. They exclude consultancy assignments and other jobs based on hour and day rates as well as jobs which did not provide salary information.



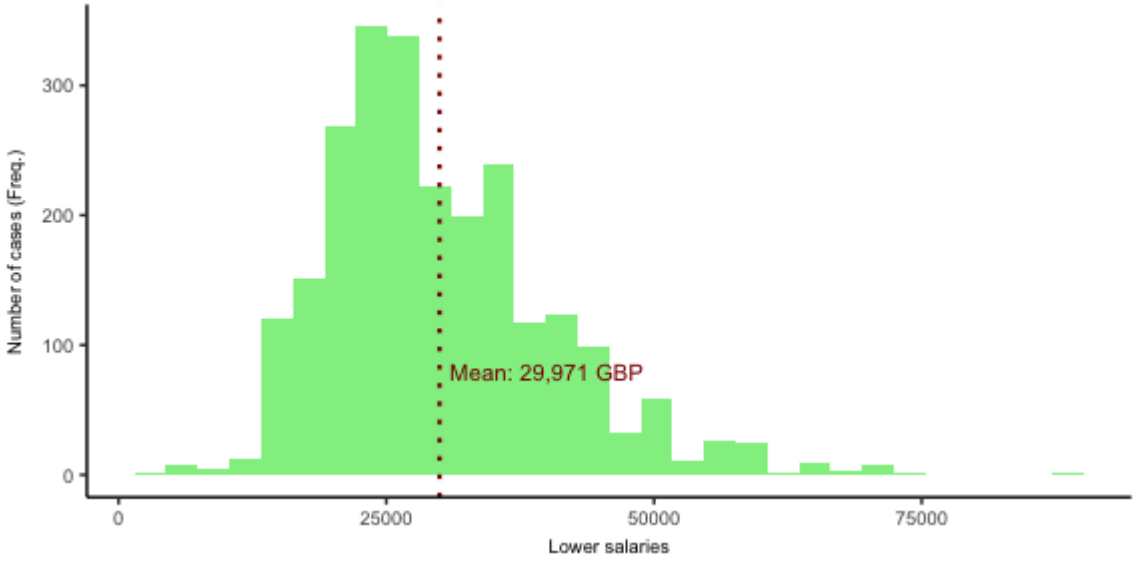
The table below presents standard descriptive statistics for lower and upper salaries. For job announcements providing a single value (not a salary range), that single amount has been incorporated to the dataset variable upper_salary while the variable lower_salary was set as NA (not available).

**Summary statistics of salaries (in British pounds)**

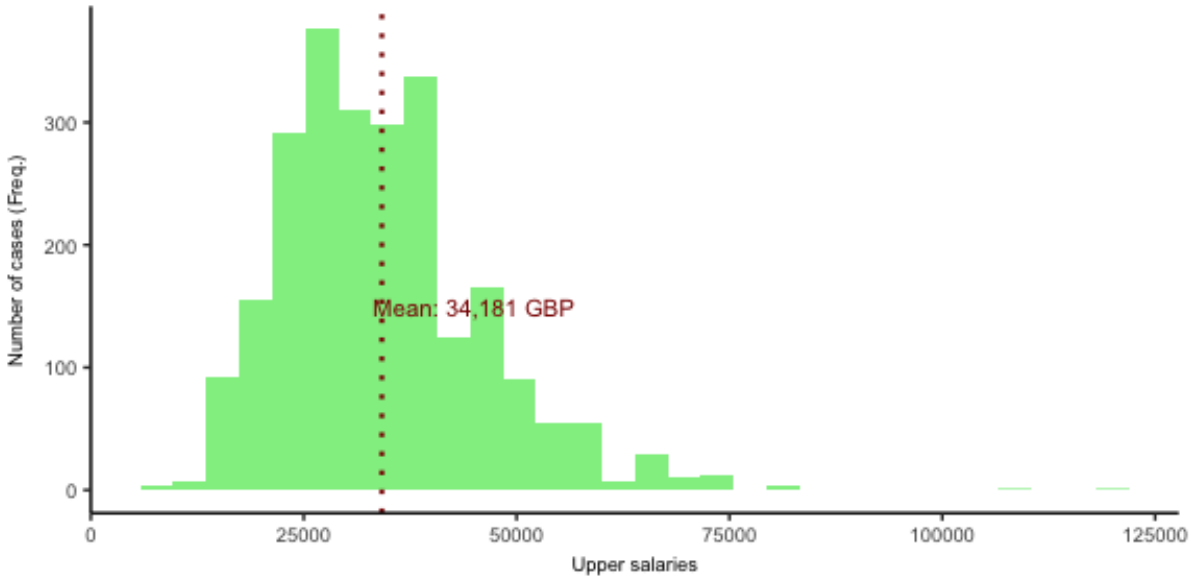
| Statistic | N | Mean | Median | St. Dev. | Min. | Max. |
|---|---|---|---|---|---|---|
| **Lower salary** | 2,429 | 29,971 | 28,000 | 10,480 | 4,290 | 90,000 |
| **Upper salary** | 2,429 | 34,181 | 32,201 | 11,725 | 7,714 | 120,000 |

In a more in-depth analysis for some future post, it can be interesting to look into payments for jobs paying by hour and by day as well for more specific job categories. One way for approaching specific job categories can be by defining tags for job titles using standard words from titles (e.g., director, management, assistant) and groupping them by tag type in a new factor variable.

**Histogram with distribution of lower salaries (in British Pounds)**




**Histogram with distribution of upper salaries (in British Pounds)**




## 5 THE 10 MOST FREQUENT RECRUITERS

The table below presents the ranking of the 10 most frequent recruiters in the dataset. Column "N" presents the number of total announcements for each recruiter while column "Freq" shows the percentage of total announcements for each recruiter. Among these are also recruitment agencies.

| Ranking | Recruiter | N | Freq |
|---|---|---|---|
| 1 | Robertson Bell | 173 | 7.12 |
| 2 | TPP Recruitment | 132 | 5.43 |
| 3 | Prospectus Ltd | 105 | 4.32 |
| 4 | Charity People Ltd | 71 | 2.92 |
| 5 | Harris Hill Charity Recruitment | 68 | 2.80 |
| 6 | Sense | 58 | 2.39 |
| 7 | Morgan Law Limited | 48 | 1.98 |
| 8 | Flow Caritas | 46 | 1.89 |
| 9 | Eden Brown | 44 | 1.81 |
| 10 | Save the Children | 37 | 1.52 |



The tables below show the ranking of the jobs with the 10 lowest and 10 highest upper salaries.

**The jobs with the 10 lowest upper salaries (in British Pounds)**

| Ranking | Title | Amount |
|---|---|---|
| 1 | Administration Assistant Stroke Association | 7,714 |
| 2 | Play Worker PACT | 7,722 |
| 3 | Mobility Aids Assistant British Red Cross | 7,725 |
| 4 | MONEY ADVISER Citizens Advice Welwyn Hatfield | 8,702 |
| 5 | Home Support Worker - Registered Service Alzheimer's Society | 9,686 |
| 6 | Deputy Shop Manager – Children's Hospice / Charity East Anglia's | 10,200 |

| | | |
|---|---|---|
| | Children's Hospices | |
| 7 | Assistant Project Co-ordinator British Red Cross | 10,428 |
| 8 | Service Support Assistant British Red Cross | 10,992 |
| 9 | Deputy Manager (North East Bristol) CLIC Sargent | 12,012 |
| 10 | Community and Events Fundraiser - 18 hours p/w Third Solutions | 12,600 |



**The jobs with the 10 highest upper salaries (in British Pounds)**

| Ranking | Title | Amount |
|---|---|---|
| 1 | Director of Finance Goodman Masson | 120,000 |
| 2 | Chief Executive Charity People Ltd | 110,000 |
| 3 | Director of Fundraising retailTRUST | 80,000 |
| 4 | Charity Director Leeds Teaching Hospitals Charitable Foundation | 80,000 |
| 5 | Director of Resources (3 days per week) King's College Hospital Charity | 80,000 |
| 6 | Operations Director Girlguiding | 75,098 |
| 7 | Director of Finance and Corporate Services National Children's Bureau | 75,000 |
| 8 | Interim International Group Financial Controller Robertson Bell | 75,000 |
| 9 | Corporate Partnerships Director Prospectus Ltd | 75,000 |

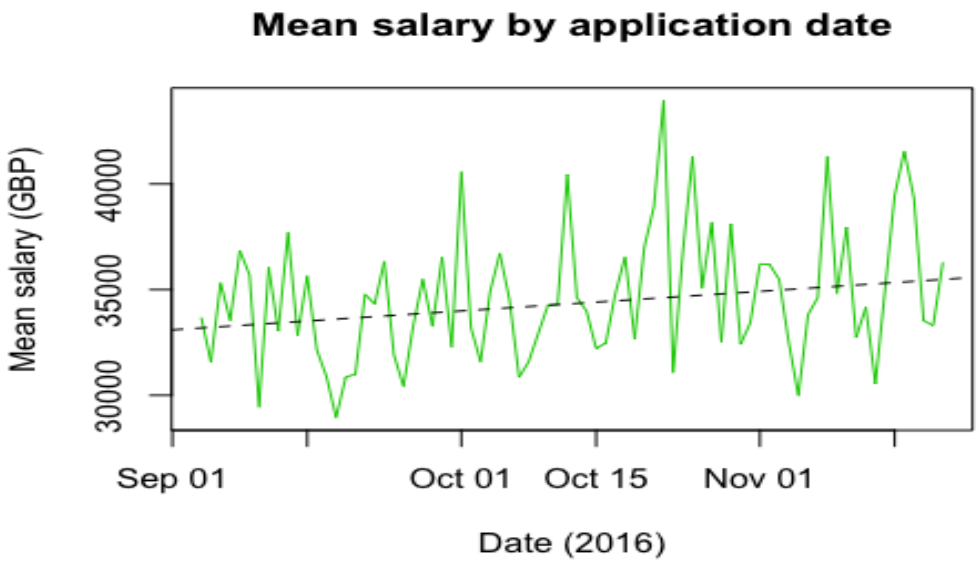
| | | |
|---|---|---|
| 10 | Director of Income Generation (Interim) TPP Recruitment | 75,000 |

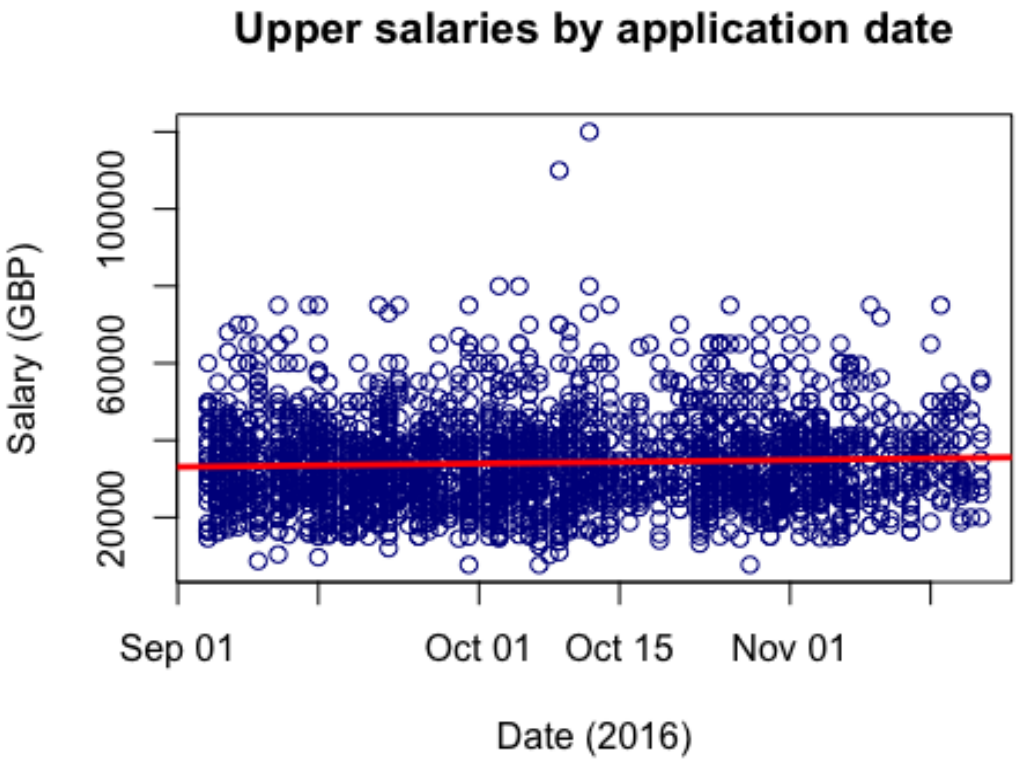
I also wanted to quickly explore possible relationships between deadline dates and salary levels just as an example of what these tools can do. It could be, for example, that some periods had lower average-salary offers than others.



Despite the large number of job announcements in the dataset (N=2,429), all observations refer to jobs with application deadlines between 04 September 2,016 and 20 November 2,016. This is a short time span for such analysis, but I explored it anyway just as an example of what these tools and techniques can do.

The plots below show upper salaries by job application deadline (restricted only to those finishing in less than 30 days counting from 20 November 2,016). The first plot uses mean salary by application date while the second plot presents single salaries by date. The dotted lines represent the results of the linear regression as a very basic attempt to describe how changes in the deadline variable (predictor) relate to changes in the upper_salary variable (response). The linear model does not help to explain the response variable variation ($R^2$ = 0.003) however it suggests a statistically significant relationship between upper salary and application-deadline date for $\alpha$ = 0.05 (p = 0.009). The output below presents the summary of the results obtained in the linear regression for the second plot (individual salaries).






**Summary of linear regression (Upper salary vs. applicaton deadline)**

```
Call:
lm(formula = upper_salary ~ deadline, data = salaries_under30d)

Residuals:
   Min     1Q Median     3Q    Max
-27073  -8164  -1691   5891  85683

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -475491.85  194141.13  -2.449  0.01439 *
deadline        29.84      11.37   2.625  0.00871 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11710 on 2427 degrees of freedom
Multiple R-squared:  0.002832, Adjusted R-squared:  0.002421
F-statistic: 6.892 on 1 and 2427 DF, p-value: 0.008712
```

## 6 WORD CLOUD OF JOB TITLES

Next, I will use word clouds to explore job titles. The following code chunk presents all transformations that took place to create the word cloud. The larger the word in the cloud, the higher is its frequency in the dataset. The words below are only those mentioned in at least 10 job announcements. The plot indicates that management positions are the most frequent ones, followed by coordination jobs, as well as officer, recruitment and fund-raising jobs.



```
# Cleaning data
## Replacing punctuation and other signs
position <- str_replace_all(dataset$position,
                            pattern = "[[:punct:]]" , "")

## Converting text to corpus format
position_corpus  <- Corpus(VectorSource(position))

## Removing stopwords
position_corpus  <- tm_map(position_corpus,
                           removeWords, c(stopwords("english"), "Ltd"))

##  Deleting empty spaces
position_corpus <- tm_map(position_corpus, stripWhitespace)

## Creating document term matrix
position_dtm <- DocumentTermMatrix(position_corpus)

# Computing word frequency in decreasing order
words.position  <- sort(colSums(as.matrix(position_dtm)),
decreasing=TRUE)

# Selecting words mentioned in at least 10 jobs
words20.position <- words.position[words.position>=10]


set.seed(56548) # Setting a random series to allow for reproducibility

## Creating matrix with corpus
matrix.cloud <- as.matrix(position_dtm)

## Estimating word frequency
freq.cloud <- sort(colSums(matrix.cloud), decreasing=TRUE)

## Defining labels
label.cloud <- names(freq.cloud)
```

```
## Criando base de dados
data.cloud <- data.frame(word=label.cloud, freq=freq.cloud)

## Plotting cloud
wordcloud(data.cloud$word, data.cloud$freq, min.freq=10,
color=brewer.pal(8,"Dark2"),
main="What are the most frequent words in the job titles from
CharityJobs?",
random.order = FALSE, max.words=80)
```




## 7 WORD CLOUD OF RECRUITERS

The cloud below shows the most frequent words in the names of the recruiting institutions. I assumed that its results could provide hints about the most active thematic areas in terms of job anouncements. The words in the plot below are also those which have been mentioned in at least 10 job anouncements. The word cloud suggests that recruitment agencies are among the leading ones, as expected (see section "The 10 most frequent recruiters"). Organisations working with children, cancer, law, international actions, education and alzheimer patients also seem to stand out.

```
# Cleaning data
## Replacing punctuation and other signs
recruiter <- str_replace_all(dataset$recruiter,
                             pattern = "[[:punct:]]" , "") %>%
  str_replace_all(pattern = "Childrens" , "children") %>%
  str_replace_all(pattern = "Alzheimers" , "Alzheimer") %>%
  str_replace_all(pattern = "Womens" , "Women")

## Converting text to corpus format
recruiter_corpus  <- Corpus(VectorSource(recruiter))

## Removing stopwords and uninformative words
recruiter_corpus  <- tm_map(recruiter_corpus, removeWords,
                            c(stopwords("english"), "The", "the",
                              "recruitment"))

##  Deleting empty spaces
recruiter_corpus <- tm_map(recruiter_corpus, stripWhitespace)

## Creating document term matrix
recruiter_dtm <- DocumentTermMatrix(recruiter_corpus)

# Computing word frequency in decreasing order
words.recruiter  <- sort(colSums(as.matrix(recruiter_dtm)),
decreasing=TRUE)

# Selecting words mentioned in at least 10 jobs
words10.recruiter <- words.recruiter[words.recruiter>=10]


set.seed(65489) # Setting a random series to allow for reproducibility

## Creating matrix with corpus
matrix.cloud <- as.matrix(recruiter_dtm)

## Estimating word frequency
freq.cloud <- sort(colSums(matrix.cloud), decreasing=TRUE)

## Defining labels
label.cloud <- names(freq.cloud)

## Creating dataframe
data.cloud <- data.frame(word=label.cloud, freq=freq.cloud)

## Plotting cloud
wordcloud(data.cloud$word, data.cloud$freq, min.freq=10,
color=brewer.pal(8,"Dark2"), main="What are the most frequent
words in the job titles from CharityJobs?", random.order = FALSE,
```

max.words=80)




## CONCLUSION

The charity, development aid, not-for-profit and social enterprise sector is evolving rapidly. This process is powered both by increasingly critical global challenges and, of course, by capable and motivated entrepreneurs, staff and service suppliers. This is a sector which is sometimes too much romantised by some people. As a consultant and entrepreneur in the sector, I am often asked how I manage to deal with all day dreamers I come accross in my way. No judgment about that but this indicates how much the sector is still unknown to the public. This is a sector which has become increasingly professional and results oriented. I believe that data science can help the sector, particularly concerning monitoring and evaluating performance including staff and beneficiary / client satisfaction.

## REFERENCES

Daróczi, G. (2014). pander: An R Pandoc Writer. R package version 0.5.1, http://cran.r-

project.org/package=pander

Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. https://CRAN.R-project.org/package=RColorBrewer

Gergely Daróczi and Roman Tsegelskyi (2015). pander: An R Pandoc Writer. R package version 0.6.0. https://CRAN.R-project.org/package=pander

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. http://www.jstatsoft.org/v40/i03/.



Hadley Wickham, James Hester and Jeroen Ooms (2017). xml2: Parse XML. R package version 1.1.1. https://CRAN.R-project.org/package=xml2.

H. Wickham (2016), ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Hadley Wickham, Jim Hester and Romain Francois (2016). readr: Read Tabular Data. R package version 1.0.0. https://CRAN.R-project.org/package=readr

Hadley Wickham (2016). rvest: Easily Harvest (Scrape) Web Pages. R package version 0.3.2, https://CRAN.R-project.org/package=rvest.

Hadley Wickham (2016). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.1.0. https://CRAN.R-project.org/package=stringr.

Hadley Wickham and Romain Francois (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.3. http://CRAN.R-project.org/package=dplyr

Ian Fellows (2014). wordcloud: Word Clouds. R package version 2.5. https://CRAN.R-project.org/package=wordcloud

Ingo Feinerer and Kurt Hornik (2015). tm: Text Mining Package. R package version 0.6-2. http://CRAN.R-project.org/package=tm

Ingo Feinerer, Kurt Hornik, and David Meyer (2008). Text Mining Infrastructure in R. Journal of Statistical Software 25(5): 1-54. http://www.jstatsoft.org/v25/i05/.

Kirkpatrick, Robert. in: UN Global Pulse (2016), Integrating Big Data into the Monitoring and Evaluation of Development Programmes. Available at: http://unglobalpulse.org/ (last access: 4 October 2017)

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/.

UN Global Pulse (2016), Integrating Big Data into the Monitoring and Evaluation of Development Programmes. Available at: http://unglobalpulse.org/ (last access: 4 October 2017)