DSI Summer Workshops Series

June 28, 2018

Peggy Lindner
Center for Advanced Computing & Data Science (CACDS)
Data Science Institute (DSI)
University of Houston
plindner@uh.edu

This jupyter notebook is available at: http://130.211.184.150/hub/login (http://130.211.184.150/hub/login)

How Much Money Should Machines Earn? *

- A journey into computerization (jobs that will be taken over by machines)

Let's learn some R by creating an interactive visualization of some open data because you will train many important skills of a data scientist:

- loading,
- transforming and
- combinig data,
- cleaning and
- performing a suitable visualization.

Datasets used

1. The probability of computerisation of 702 detailed occupations, obtained by Carl Benedikt Frey and Michael A. Osborne from the University of Oxford, using a Gaussian process classifier and published in this paper (https://www.oxfordmartin.ox.ac.uk/downloads/academic/The_Future_of_Employment.pdf) in 2013.
2. Statistics of jobs from (employments, median annual wages and typical education needed for entry) from the US Bureau of Labor as "Occupational projections", available here (https://www.bls.gov/emp/ind-occ-matrix/occupation.xlsx).

In [ ]:
```
R needs some additional packages to do the work ...
```

In [ ]:
```
# Load libraries
library(dplyr)
library(tabulizer)
library(rlist)
library(readxl)
```

Data (Down)Loading

In [ ]:
```
################################################################################
################
# Download and parse data about probability of computerisation
################################################################################
################

# set some variables to be used for download
urlfile <- "https://www.oxfordmartin.ox.ac.uk/downloads/academic/The_Future_o
f_Employment.pdf"
file <- "The_Future_of_Employment.pdf"

# download the pdf file (if we haven't done so already)
if (!file.exists(file)) {
    download.file(urlfile, destfile = file, mode = 'wb')
}
```

Extracting data from a pdf file

using [Tabula (https://tabula.technology/)](https://tabula.technology/) from within R

In [ ]:
```
# Extract tables using tabulizer - that looks a little bit like magic( and it
 takes some time)
out <- extract_tables(file, encoding="UTF-8")
```

In [ ]:
```
# let's have a look a the "thing" that we just got
out
```

Data Transformation

In [ ]:
```
# We are not interested in first two tables - so let's remove them
list.remove(out, c(1:2)) -> tables

# now let's look what we got
tables
```

Parse table into something that can be used in the next step

```
In [ ]: # First we create a placefolder
prob_comput_df=data.frame()

# Now we go over each of the tables
for (i in 1:length(tables))
{
  # We keep just SOC Code, rank and probability of computerisation
  # We also remove first to lines of each element of table since they are non
 interesting
  tables[[i]][-c(1,2),c(1,2,4)] %>%
    as.data.frame(stringsAsFactors = FALSE) %>%
    rbind(prob_comput_df) -> prob_comput_df
}
```

```
In [ ]: # Let's check what we got
prob_comput_df
```

```
In [ ]: # Let's give this thing some proper column names
colnames(prob_comput_df) = c("rank", "probability", "soc")

prob_comput_df
```

```
In [ ]: #### Data Cleaning
```

```
In [ ]: # what does R think it is looking at?
str(prob_comput_df)
```

```
In [ ]: prob_comput_df %>%
  # convert things that look like numbers into numbers
  mutate(rank=gsub("\\.","", rank) %>% as.numeric()) %>%
  #let's get rid of missing data
  na.omit() -> prob_comput_df
```

```
In [ ]: str(prob_comput_df)
```

```
In [ ]: # finally let's delete the file that we just downloaded
file.remove(file)
```

Data (Down)Loading

```
In [ ]: ############################################################################
################
# Download job statistics
############################################################################
################

# set some variables to be used for download
urlfile <- "https://www.bls.gov/emp/ind-occ-matrix/occupation.xlsx"
file <- "occupation.xlsx"
# Download xlsx file
if (!file.exists(file)) {
    download.file(urlfile, destfile = file, mode = 'wb')
}
```

In [ ]:
```
# read excel file into R
job_stats_df <- read_excel(file,
                           sheet="Table 1.7",
                           skip=3,
                           col_names = c("job_title",
                                         "soc",
                                         "occupation_type",
                                         "employment_2016",
                                         "employment_2026",
                                         "employment_change_2016_26_nu",
                                         "employment_change_2016_26_pe",
                                         "self_employed_2016_pe",
                                         "occupational_openings_2016_26_av",
                                         "median_annual_wage_2017",
                                         "typical_education_entry",
                                         "work_experience_related_occ",
                                         "typical_training_needed"))
```

In [ ]:
```
# now we can remove the downloaded file
file.remove(file)
```

In [ ]:
```
# let's look what we got here
job_stats_df
```

Data Transformation & Cleaning

We are going to merge (join) the 2 data sets and keep only the columns that we need.

In [ ]:
```
##############################################################################
################
# Join data frames
##############################################################################
################
results = prob_comput_df %>%
  inner_join(job_stats_df, by = "soc") %>%
  select(job_title,
         probability,
         employment_2016,
         median_annual_wage_2017,
         typical_education_entry) %>%
  mutate(probability=as.numeric(probability),
         median_annual_wage_2017=as.numeric(median_annual_wage_2017),
         typical_education_entry=iconv(typical_education_entry, "latin1", "AS
CII")) %>%
  # get rid of missing data
  na.omit()
```

In [ ]:
```
# Aehmm, can we do that a little slower?
#first, we join using the soc column
first_step <- prob_comput_df %>%
     inner_join(job_stats_df, by = "soc")

first_step
```

In [ ]:
```
#second, we select only columns that we want
second_step <- first_step %>%
  select(job_title,
         probability,
         employment_2016,
         median_annual_wage_2017,
         typical_education_entry)

second_step
```

In [ ]:
```
#third, we create 2 new columns using the existing columns

third_step <- second_step %>%
  mutate(probability=as.numeric(probability),
         median_annual_wage_2017=as.numeric(median_annual_wage_2017),
         typical_education_entry=iconv(typical_education_entry, "latin1", "AS
CII"))

third_step

#that looks the same to me, but internally we change some data types
str(second_step)
str(third_step)
```

In [ ]:
```
#do we have some missing data points?
is.na(third_step)
```

In [ ]:
```
#show me the rows with missing data
third_step[!complete.cases(third_step),]
```

In [ ]:
```
# and last but not least we remove the rows with missing data
results <- third_step %>%
    na.omit()
```

In [ ]:
```
#what did we get?
results
```

Finally, let's create a visualization

We are ging to use Highcharter (http://jkunst.com/highcharter/index.html) which is just one of many ways to create interactive visualizations in R.

In [ ]:
```
#we need some more packages
library(highcharter)
library(htmlwidgets)
library(IRdisplay)
```

```
In [ ]: #let's create an object that is actually a visual
x=hchart(results,
       "scatter",
       hcaes(x = probability*100,
             y = median_annual_wage_2017,
             group=typical_education_entry,
             size=employment_2016)) %>%
  hc_title(text = "How Much Money Should Machines Earn?") %>%
  hc_subtitle(text = "Probability of Computerisation and Wages by Job") %>%
  hc_credits(enabled = TRUE, text = "Source: Oxford Martin School and US Depa
rtment of Labor") %>%
  hc_xAxis(title = list(text = "Probability of Computerisation"), labels = li
st(format = "{value}%")) %>%
  hc_yAxis(title = list(text = "Median Annual Wage 2017"), labels = list(form
at = "{value}$")) %>%
  hc_plotOptions(bubble = list(minSize = 3, maxSize = 35)) %>%
  hc_tooltip(formatter = JS("function(){
                            return ('<b>'+ this.point.job_title + '</b><br>'+
                            'Probability of computerisation: '+ Highcharts.nu
mberFormat(this.x, 0)+'%' +
                            '<br>Median annual wage 2017 ($): '+ Highcharts.n
umberFormat(this.y, 0) +
                            '<br>Employment 2016 (000s): '+ Highcharts.number
Format(this.point.size, 0) )}")) %>%
  hc_chart(zoomType = "xy") %>%
  hc_exporting(enabled = TRUE)

# it's an object!
str(x)
```

```
In [ ]: # and now let's get this object showing up in our jupyter notebook
saveWidget(x, 'demox.html', selfcontained = FALSE)
display_html('<iframe src="demox.html", width = 900, height = 500 ></iframe>'
)
```

A full size version of the visualization can be found here (https://fronkonstin.com/wp-content/uploads/2018/06/machines_wage.html)

And thanks again to the person who wrote the original post (https://fronkonstin.com/2018/06/17/how-much-money-should-machines-earn/)!

These are some insights:

- There is a moderate negative correlation between wages and probability of computerisation.
- Around 45% of US employments are threatened by machines (have a computerisation probability higher than 80%): half of them do not require formal education to entry.
- In fact, 78% of jobs which do not require formal education to entry are threatened by machines: 0% which require a master's degree are.
- Teachers are absolutely irreplaceable (0% are threatened by machines) but they earn a 2.2% less then the average wage (unfortunately, I'm afraid this phenomenon occurs in many other countries as well).
- Don't study for librarian or archivist: it seems a bad way to invest your time
- Mathematicians will survive to machines

What do you see there?