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

Please make sure you have Jupyterhub running with support for R and all the required packages installed. Data for this and other tutorials can be found in the github repsoitory for the Summer 2018 DSI Workshops https://github.com/peggylind/Materials_Summer2018 (https://github.com/peggylind/Materials_Summer2018)

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_Employ in 2013.
2. Statistics of jobs from (employments, median annual wages and typical education needed for entry) from the US Bureau of Labor, available here.

R needs some additional packages to do the work ...

In [3]:

```
# Load libraries
library(dplyr)
library(tabulizer)
library(rlist)
library(readxl)
```

```
Warning message:
“package ‘dplyr’ was built under R version 3.5.1”
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stat
s’:

    filter, lag

The following objects are masked from ‘package:bas
e’:

    intersect, setdiff, setequal, union
```

Data (Down)Loading

In [4]:

```
###############################################################
###############################
# Download and parse data about probability of computerisation
###############################################################
###############################

# set some variables to be used for download
urlfile <- "https://www.oxfordmartin.ox.ac.uk/downloads/academ
ic/The_Future_of_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/) from within R

In [5]:

```
# Extract tables using tabulizer - that works a little bit lik
e magic( and it takes some time)
out <- extract_tables(file, encoding="UTF-8")
```

In [6]:

```
# let's have a look at 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 [8]:

```
# 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 computerisa
tion
  # We also remove first to lines of each element of table sin
ce 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 [9]:

```
# Let's check what we got
prob_comput_df
```

V1	V2	V3
687.	0.98	43-4151
688.	0.98	43-4011
689.	0.98	43-9041
690.	0.98	51-2093
691.	0.99	43-9021
692.	0.99	25-4031
693.	0.99	43-4141
694.	0.99	51-9151
695.	0.99	13-2082
696.	0.99	43-5011
697.	0.99	49-9064
698.	0.99	13-2053
699.	0.99	15-2091
700.	0.99	51-6051
701.	0.99	23-2093
702.	0.99	41-9041
640.	0.96	49-9093
641.	0.96	35-2014
642.	0.96	39-3031
643.	0.96	43-3021
644.	0.97	53-6011
645.	0.97	51-7042
646.	0.97	51-2092
647.	0.97	51-6042

V1	V2	V3
648.	0.97	51-2023
649.	0.97	13-1074
650.	0.97	51-6061
651.	0.97	51-9081
652.	0.97	51-9021
⋮	⋮	⋮
10.	0.0036	33-1021
11.	0.0039	29-1031
12.	0.0039	11-9081
13.	0.004	27-2032
14.	0.0041	41-9031
15.	0.0042	29-1060
16.	0.0042	25-9031
17.	0.0043	19-3039
18.	0.0044	33-1012
19.	0.0044	29-1021
20.	0.0044	25-2021
21.	0.0045	19-1042
22.	0.0046	11-9032
23.	0.0046	29-1081
24.	0.0047	19-3031
25.	0.0048	21-1014
26.	0.0049	51-6092
27.	0.0055	27-1027

V1	V2	V3
28.	0.0055	11-3121
29.	0.0061	39-9032
30.	0.0063	11-3131
31.	0.0064	29-1127
32.	0.0065	15-1121
33.	0.0067	11-9151
34.	0.0068	25-4012
35.	0.0071	29-9091
36.	0.0073	11-9111
37.	0.0074	25-2011
38.	0.0075	25-9021
39.	0.0077	19-3091

In [10]:

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

prob_comput_df
```

rank	probability	soc
687.	0.98	43-4151
688.	0.98	43-4011
689.	0.98	43-9041
690.	0.98	51-2093
691.	0.99	43-9021
692.	0.99	25-4031
693.	0.99	43-4141
694.	0.99	51-9151
695.	0.99	13-2082
696.	0.99	43-5011
697.	0.99	49-9064
698.	0.99	13-2053
699.	0.99	15-2091
700.	0.99	51-6051
701.	0.99	23-2093
702.	0.99	41-9041
640.	0.96	49-9093
641.	0.96	35-2014
642.	0.96	39-3031
643.	0.96	43-3021
644.	0.97	53-6011
645.	0.97	51-7042
646.	0.97	51-2092
647.	0.97	51-6042

rank	probability	soc
648.	0.97	51-2023
649.	0.97	13-1074
650.	0.97	51-6061
651.	0.97	51-9081
652.	0.97	51-9021
⋮	⋮	⋮
10.	0.0036	33-1021
11.	0.0039	29-1031
12.	0.0039	11-9081
13.	0.004	27-2032
14.	0.0041	41-9031
15.	0.0042	29-1060
16.	0.0042	25-9031
17.	0.0043	19-3039
18.	0.0044	33-1012
19.	0.0044	29-1021
20.	0.0044	25-2021
21.	0.0045	19-1042
22.	0.0046	11-9032
23.	0.0046	29-1081
24.	0.0047	19-3031
25.	0.0048	21-1014
26.	0.0049	51-6092
27.	0.0055	27-1027

rank	probability	soc
28.	0.0055	11-3121
29.	0.0061	39-9032
30.	0.0063	11-3131
31.	0.0064	29-1127
32.	0.0065	15-1121
33.	0.0067	11-9151
34.	0.0068	25-4012
35.	0.0071	29-9091
36.	0.0073	11-9111
37.	0.0074	25-2011
38.	0.0075	25-9021
39.	0.0077	19-3091

In [11]:

```
#### Data Cleaning
```

In [12]:

```
# what does R think it is looking at?
str(prob_comput_df)
```

```
'data.frame':	740 obs. of  3 variables:
 $ rank       : chr  "687." "688." "689." "690."
...
 $ probability: chr  "0.98" "0.98" "0.98" "0.98"
...
 $ soc        : chr  "43-4151" "43-4011" "43-9041"
"51-2093" ...
```

In [13]:

```
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 [14]:

```
str(prob_comput_df)
```

```
'data.frame':	702 obs. of  3 variables:
 $ rank       : num  687 688 689 690 691 692 693 69
4 695 696 ...
 $ probability: chr  "0.98" "0.98" "0.98" "0.98"
...
 $ soc        : chr  "43-4151" "43-4011" "43-9041"
"51-2093" ...
 - attr(*, "na.action")= 'omit' Named int  30 57 77
92 97 108 112 117 120 125 ...
  ..- attr(*, "names")= chr  "30" "57" "77" "92"
...
```

In [15]:

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

TRUE

Data (Down)Loading

In [16]:

```
##############################################################
##############################
# 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 [17]:

```
# 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_20
16_26_nu",
                                         "employment_change_20
16_26_pe",
                                         "self_employed_2016_p
e",
                                         "occupational_opening
s_2016_26_av",
                                         "median_annual_wage_2
017",
                                         "typical_education_en
try",
                                         "work_experience_rela
ted_occ",
                                         "typical_training_nee
ded"))
```

In [18]:

```
# now we can remove the downloaded file
file.remove(file)
```

TRUE

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 [20]:

```
#####################################################################
#################################
# 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", "ASCII")) %>%
  # get rid of missing data
  na.omit()
```

```
Warning message in evalq(as.numeric(median_annual_w
age_2017), <environment>):
“NAs introduced by coercion”
```

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 [22]:

```
#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
```

job_title	probability	employment_2016	median_annua
Order clerks	0.98	179.0	33510
Brokerage clerks	0.98	60.4	49800
Insurance claims and policy processing clerks	0.98	308.5	38790
Timing device assemblers and adjusters	0.98	0.8	34800
Data entry keyers	0.99	203.8	30930
Library technicians	0.99	99.2	33690
New accounts clerks	0.99	42.0	35260
Photographic process workers and processing machine operators	0.99	26.9	27480
Tax preparers	0.99	95.9	38730
Cargo and freight agents	0.99	89.8	41820
Watch repairers	0.99	1.8	35770
Insurance underwriters	0.99	104.1	69760

job_title	probability	employment_2016	median_annua
Mathematical technicians	0.99	0.6	—
Sewers, hand	0.99	13.5	26230
Title examiners, abstractors, and searchers	0.99	69.0	46850
Telemarketers	0.99	216.6	24460
Fabric menders, except garment	0.96	0.7	28640
Cooks, restaurant	0.96	1231.9	25180
Ushers, lobby attendants, and ticket takers	0.96	117.7	20820
Billing and posting clerks	0.96	501.0	36860
Bridge and lock tenders	0.97	3.7	50240
Woodworking machine setters, operators, and tenders, except sawing	0.97	77.1	29110
Team assemblers	0.97	1130.9	—
Shoe machine operators and tenders	0.97	3.6	27420

job_title	probability	employment_2016	median_annua
Electromechanical equipment assemblers	0.97	45.7	—
Farm labor contractors	0.97	3.0	35660
Textile bleaching and dyeing machine operators and tenders	0.97	11.3	28280
Dental laboratory technicians	0.97	38.1	38670
Crushing, grinding, and polishing machine setters, operators, and tenders	0.97	30.2	35380
Grinding and polishing workers, hand	0.97	26.6	28830
⋮	⋮	⋮	⋮
First-line supervisors of fire fighting and prevention workers	0.0036	59.1	76170
Dietitians and nutritionists	0.0039	68.0	59410
Lodging managers	0.0039	47.8	51800

job_title	probability	employment_2016	median_annua
Choreographers	0.004	6.9	48420
Sales engineers	0.0041	74.9	98720
Physicians and surgeons	0.0042	713.8	>=$208,000
Instructional coordinators	0.0042	163.2	63750
Psychologists, all other	0.0043	17.4	97740
First-line supervisors of police and detectives	0.0044	104.7	87910
Dentists, general	0.0044	132.8	151440
Elementary school teachers, except special education	0.0044	1410.9	57160
Medical scientists, except epidemiologists	0.0045	120.0	82090
Education administrators, elementary and secondary school	0.0046	251.3	94390
Podiatrists	0.0046	11.0	127740

job_title	probability	employment_2016	median_annua
Clinical, counseling, and school psychologists	0.0047	147.5	75090
Mental health counselors	0.0048	157.7	—
Fabric and apparel patternmakers	0.0049	5.5	40460
Set and exhibit designers	0.0055	14.6	53090
Human resources managers	0.0055	136.1	110120
Recreation workers	0.0061	390.0	24540
Training and development managers	0.0063	34.5	108250
Speech-language pathologists	0.0064	145.1	76610
Computer systems analysts	0.0065	600.5	88270
Social and community service managers	0.0067	147.3	64100
Curators	0.0068	12.4	53770
Athletic trainers	0.0071	27.8	46630
Medical and health services managers	0.0073	352.2	98350

job_title	probability	employment_2016	median_annua
Preschool teachers, except special education	0.0074	478.5	28990
Farm and home management advisors	0.0075	10.4	49510
Anthropologists and archeologists	0.0077	7.6	62280

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", "ASCII"))

third_step

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

In [24]:

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

job_title	probability	employment_2016	median_annual_wage_2
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

job_title	probability	employment_2016	median_annual_wage_2
FALSE	FALSE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
⋮	⋮	⋮	⋮
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

job_title	probability	employment_2016	median_annual_wage_2
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE

In [25]:

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

	job_title	probability	employment_2016	median_a
13	Mathematical technicians	0.9900	0.6	NA
23	Team assemblers	0.9700	1130.9	NA
25	Electromechanical equipment assemblers	0.9700	45.7	NA
89	Electrical and electronic equipment assemblers	0.9500	218.9	NA
174	Medical and clinical laboratory technologists	0.9000	171.4	NA
187	Tour guides and escorts	0.9100	45.8	NA
195	Segmental pavers	0.8300	2.1	NA
232	Miscellaneous agricultural workers	0.8700	847.3	23710
237	Buyers and purchasing agents, farm products	0.8700	13.7	NA
246	Purchasing agents, except wholesale, retail, and farm products	0.7700	309.4	NA

	job_title	probability	employment_2016	median_a
367	Computer support specialists	0.6500	835.3	52810
381	First-line supervisors of helpers, laborers, and material movers, hand	0.4200	184.4	NA
390	Medical and clinical laboratory technicians	0.4700	164.2	NA
411	Slot supervisors	0.5400	12.1	NA
429	Gaming supervisors	0.2800	38.5	NA
431	Wholesale and retail buyers, except farm products	0.2900	123.3	NA
452	Actors	0.3700	63.8	NA
515	Travel guides	0.0570	3.6	NA
527	Musicians and singers	0.0740	172.4	NA
551	Dancers	0.1300	13.5	NA
562	Orthodontists	0.0230	6.6	NA

	job_title	probability	employment_2016	median_a
573	First-line supervisors of transportation and material-moving machine and vehicle operators	0.0290	204.2	NA
580	Postsecondary teachers	0.0320	1871.4	67140
582	Substance abuse and behavioral disorder counselors	0.0330	102.4	NA
657	Oral and maxillofacial surgeons	0.0036	6.8	NA
663	Physicians and surgeons	0.0042	713.8	NA
673	Mental health counselors	0.0048	157.7	NA

In [26]:

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

In [27]:

```
#what did we get?
results
```

	job_title	probability	employment_2016	median_anr
1	Order clerks	0.98	179.0	33510
2	Brokerage clerks	0.98	60.4	49800
3	Insurance claims and policy processing clerks	0.98	308.5	38790
4	Timing device assemblers and adjusters	0.98	0.8	34800
5	Data entry keyers	0.99	203.8	30930
6	Library technicians	0.99	99.2	33690
7	New accounts clerks	0.99	42.0	35260
8	Photographic process workers and processing machine operators	0.99	26.9	27480
9	Tax preparers	0.99	95.9	38730
10	Cargo and freight agents	0.99	89.8	41820
11	Watch repairers	0.99	1.8	35770

	job_title	probability	employment_2016	median_anr
12	Insurance underwriters	0.99	104.1	69760
14	Sewers, hand	0.99	13.5	26230
15	Title examiners, abstractors, and searchers	0.99	69.0	46850
16	Telemarketers	0.99	216.6	24460
17	Fabric menders, except garment	0.96	0.7	28640
18	Cooks, restaurant	0.96	1231.9	25180
19	Ushers, lobby attendants, and ticket takers	0.96	117.7	20820
20	Billing and posting clerks	0.96	501.0	36860
21	Bridge and lock tenders	0.97	3.7	50240
22	Woodworking machine setters, operators, and tenders, except sawing	0.97	77.1	29110
24	Shoe machine operators and tenders	0.97	3.6	27420

	job_title	probability	employment_2016	median_anr
26	Farm labor contractors	0.97	3.0	35660
27	Textile bleaching and dyeing machine operators and tenders	0.97	11.3	28280
28	Dental laboratory technicians	0.97	38.1	38670
29	Crushing, grinding, and polishing machine setters, operators, and tenders	0.97	30.2	35380
30	Grinding and polishing workers, hand	0.97	26.6	28830
31	Pesticide handlers, sprayers, and applicators, vegetation	0.97	38.0	34830
32	Log graders and scalers	0.97	4.2	37880
33	Ophthalmic laboratory technicians	0.97	29.1	30960
⋮	⋮	⋮	⋮	⋮

	job_title	probability	employment_2016	median_anr
655	Orthotists and prosthetists	0.0035	7.8	66240
656	Healthcare social workers	0.0035	176.5	54870
658	First-line supervisors of fire fighting and prevention workers	0.0036	59.1	76170
659	Dietitians and nutritionists	0.0039	68.0	59410
660	Lodging managers	0.0039	47.8	51800
661	Choreographers	0.0040	6.9	48420
662	Sales engineers	0.0041	74.9	98720
664	Instructional coordinators	0.0042	163.2	63750
665	Psychologists, all other	0.0043	17.4	97740
666	First-line supervisors of police and detectives	0.0044	104.7	87910
667	Dentists, general	0.0044	132.8	151440

	job_title	probability	employment_2016	median_ann
668	Elementary school teachers, except special education	0.0044	1410.9	57160
669	Medical scientists, except epidemiologists	0.0045	120.0	82090
670	Education administrators, elementary and secondary school	0.0046	251.3	94390
671	Podiatrists	0.0046	11.0	127740
672	Clinical, counseling, and school psychologists	0.0047	147.5	75090
674	Fabric and apparel patternmakers	0.0049	5.5	40460
675	Set and exhibit designers	0.0055	14.6	53090
676	Human resources managers	0.0055	136.1	110120
677	Recreation workers	0.0061	390.0	24540

	job_title	probability	employment_2016	median_ann
678	Training and development managers	0.0063	34.5	108250
679	Speech-language pathologists	0.0064	145.1	76610
680	Computer systems analysts	0.0065	600.5	88270
681	Social and community service managers	0.0067	147.3	64100
682	Curators	0.0068	12.4	53770
683	Athletic trainers	0.0071	27.8	46630
684	Medical and health services managers	0.0073	352.2	98350
685	Preschool teachers, except special education	0.0074	478.5	28990
686	Farm and home management advisors	0.0075	10.4	49510
687	Anthropologists and archeologists	0.0077	7.6	62280

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 [28]:

```
#we need some more packages
#highcharter works with current ggplot only in dev version
#devtools::install_github("jbkunst/highcharter")
#source("https://install-github.me/jbkunst/highcharter")
library(highcharter)
library(htmlwidgets)
library(IRdisplay)
```

In [ ]:

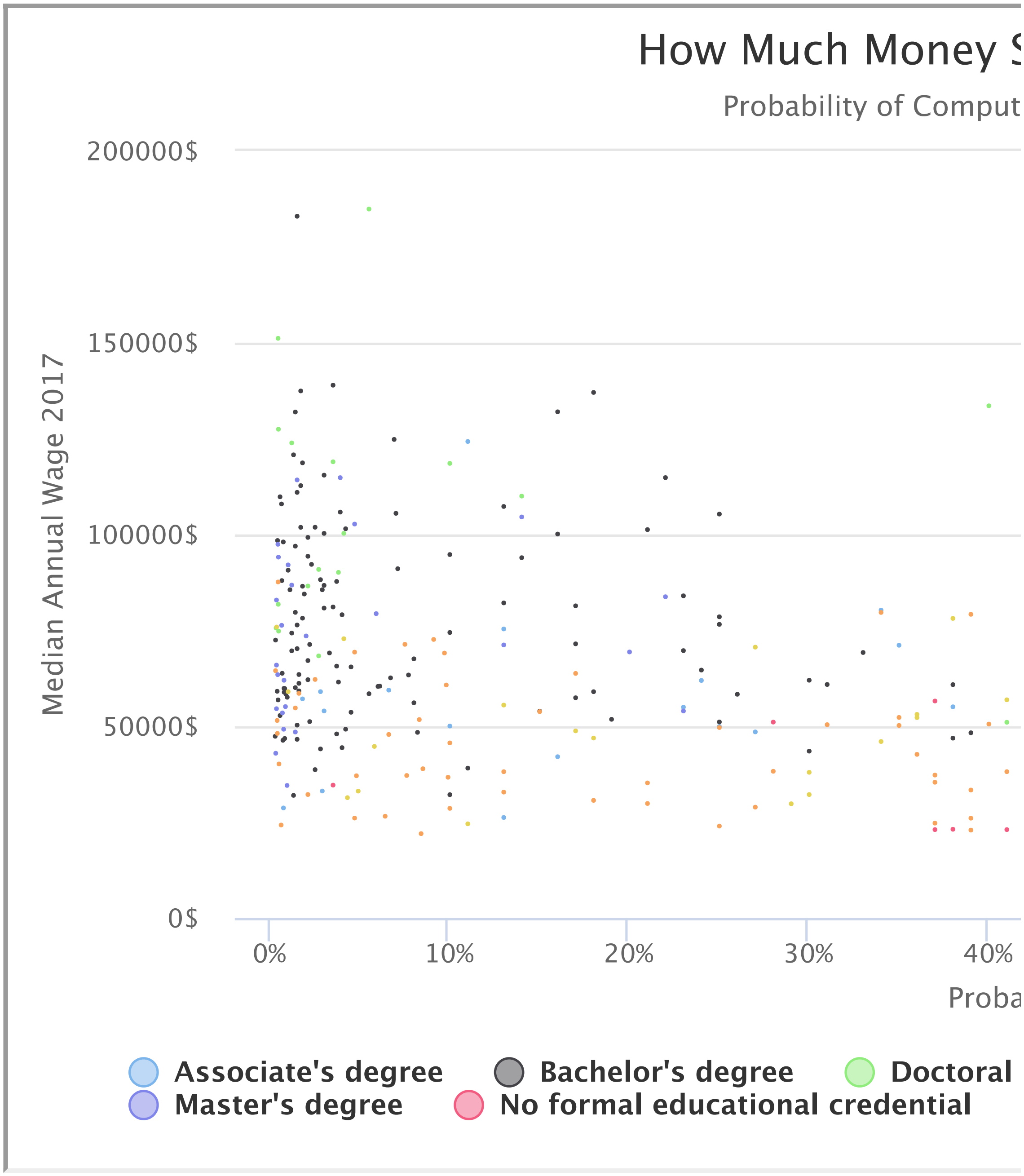
```
#let's create an object that is actually a visual
x=hchart(results,
       "scatter",
       hcaes(x = probability*100,
             y = median_annual_wage_2017,
             group=typical_education_entry,
             size=employment_2016)) %>%
  hc_title(text = "How Much Money Should Machines Earn?") %>%
  hc_subtitle(text = "Probability of Computerisation and Wages
 by Job") %>%
  hc_credits(enabled = TRUE, text = "Source: Oxford Martin Sch
ool and US Department of Labor") %>%
  hc_xAxis(title = list(text = "Probability of Computerisatio
n"), labels = list(format = "{value}%")) %>%
  hc_yAxis(title = list(text = "Median Annual Wage 2017"), lab
els = list(format = "{value}$")) %>%
  hc_plotOptions(bubble = list(minSize = 3, maxSize = 35)) %>%

  hc_tooltip(formatter = JS("function(){
                            return ('<b>'+ this.point.job_titl
e + '</b><br>'+
                            'Probability of computerisation:
 '+ Highcharts.numberFormat(this.x, 0)+'%' +
                            '<br>Median annual wage 2017 ($):
 '+ Highcharts.numberFormat(this.y, 0) +
                            '<br>Employment 2016 (000s): '+ Hi
ghcharts.numberFormat(this.point.size, 0) )}")) %>%
  hc_chart(zoomType = "xy") %>%
  hc_exporting(enabled = TRUE)

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

In [32]:

```
# and now let's get this object showing up in our jupyter note
book
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)](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?