Solution: Employee Retention

```
#libraries needed
require(dplyr)
require(rpart)
require(ggplot2)
require(scales)
```

```
data = read.csv("employee_retention_data.csv") #read data set
str(data) #check the structure
```

```
## 'data.frame':    24702 obs. of  7 variables:
##  $ employee_id: int  13021 825355 927315 662910 256971
...
##  $ company_id : int  7 7 4 7 2 4 4 2 9 1 ...
##  $ dept       : Factor w/ 6 levels "customer_service",..: 1 5 5 1 2 2 1 1 4 6
...
##  $ seniority  : int  28 20 14 20 23 14 21 4 7 7 ...
##  $ salary     : num  89000 183000 101000 115000 276000 165000 107000 30000 1600
00 104000 ...
##  $ join_date  : Factor w/ 995 levels "2011-01-24","2011-01-25",..: 643 459 758
264 148 205 558 633 380 280 ...
##  $ quit_date  : Factor w/ 664 levels "2011-10-13","2011-10-14",..: 643 364 NA 2
29 428 267 NA NA 640 NA ...
```

```
data$company_id = as.factor(data$company_id) # this is a categorical var
data$join_date = as.Date(data$join_date) #make it a date
data$quit_date = as.Date(data$quit_date) #make it a date
summary(data) # everything seems to make sense, some simple plots would help doubl
e check that
```

```
##   employee_id       company_id                  dept        seniority
##  Min.   :    36   1      :8486   customer_service:9180   Min.   : 1.00
##  1st Qu.:250134   2      :4222   data_science    :3190   1st Qu.: 7.00
##  Median :500793   3      :2749   design          :1380   Median :14.00
##  Mean   :501604   4      :2062   engineer        :4613   Mean   :14.13
##  3rd Qu.:753137   5      :1755   marketing       :3167   3rd Qu.:21.00
##  Max.   :999969   6      :1291   sales           :3172   Max.   :99.00
##                   (Other):4137
##      salary          join_date            quit_date
##  Min.   : 17000   Min.   :2011-01-24   Min.   :2011-10-13
##  1st Qu.: 79000   1st Qu.:2012-04-09   1st Qu.:2013-06-28
##  Median :123000   Median :2013-06-24   Median :2014-06-20
##  Mean   :138183   Mean   :2013-06-29   Mean   :2014-05-02
##  3rd Qu.:187000   3rd Qu.:2014-09-17   3rd Qu.:2015-03-27
##  Max.   :408000   Max.   :2015-12-10   Max.   :2015-12-09
##                                        NA's   :11192
```

Let’s answer this question: You should create a table with 3 columns: day, employee_headcount, company_id.

```
unique_dates = seq(as.Date("2011/01/24"), as.Date("2015/12/13"), by = "day") # cre
ate list of unique dates for the table
unique_companies = unique(data$company_id) #ceate list of unique companies
data_headcount = merge(unique_dates, unique_companies, by = NULL) #cross join so I
get all combinations of dates and companies. Will need it later.
colnames(data_headcount) = c("date", "company_id")

#now I get for each day/company, how many people quit/got hired on that day
data_join = data %>%
            group_by(join_date, company_id) %>%
            summarise(join_count = length(join_date))

data_quit = data %>%
            group_by(quit_date, company_id) %>%
            summarise(quit_count = length(quit_date))

#Now I left outer join with data_headcount.
#NA means no people were hired/quit on that day cause there is no match.

data_headcount = merge (data_headcount, data_join,
                        by.x = c("date", "company_id"),
                        by.y = c("join_date", "company_id"),
                        all.x = TRUE)

data_headcount = merge (data_headcount, data_quit,
                        by.x = c("date", "company_id"),
                        by.y = c("quit_date", "company_id"),
                        all.x = TRUE)
#replace the NAs with 0
data_headcount$join_count[is.na(data_headcount$join_count)] = 0
data_headcount$quit_count[is.na(data_headcount$quit_count)] = 0

#Now I need the sum by company_id. Data set is already ordered by date,
# so I can simply use dplyr to group by company_id and do cumsum

data_headcount = data_headcount %>%
            group_by(company_id) %>%
            mutate ( join_cumsum = cumsum(join_count),
                     quit_cumsum = cumsum(quit_count)
                    )
# finally, for each date I just take join_count - quit_count and I am done
data_headcount$count = data_headcount$join_cumsum - data_headcount$quit_cumsum
data_headcount_table = data.frame(data_headcount[, c("date", "company_id","coun
t")])

#Another way to do it would be with a for loop.
#While you often hear that you should avoid for loops in R as much as possible,
#in some cases you don't care that much about processing time, and you are
#willing to have something slower but more understandable.
#Other data scientists reading your code in future (or even yourself) will appreci
```

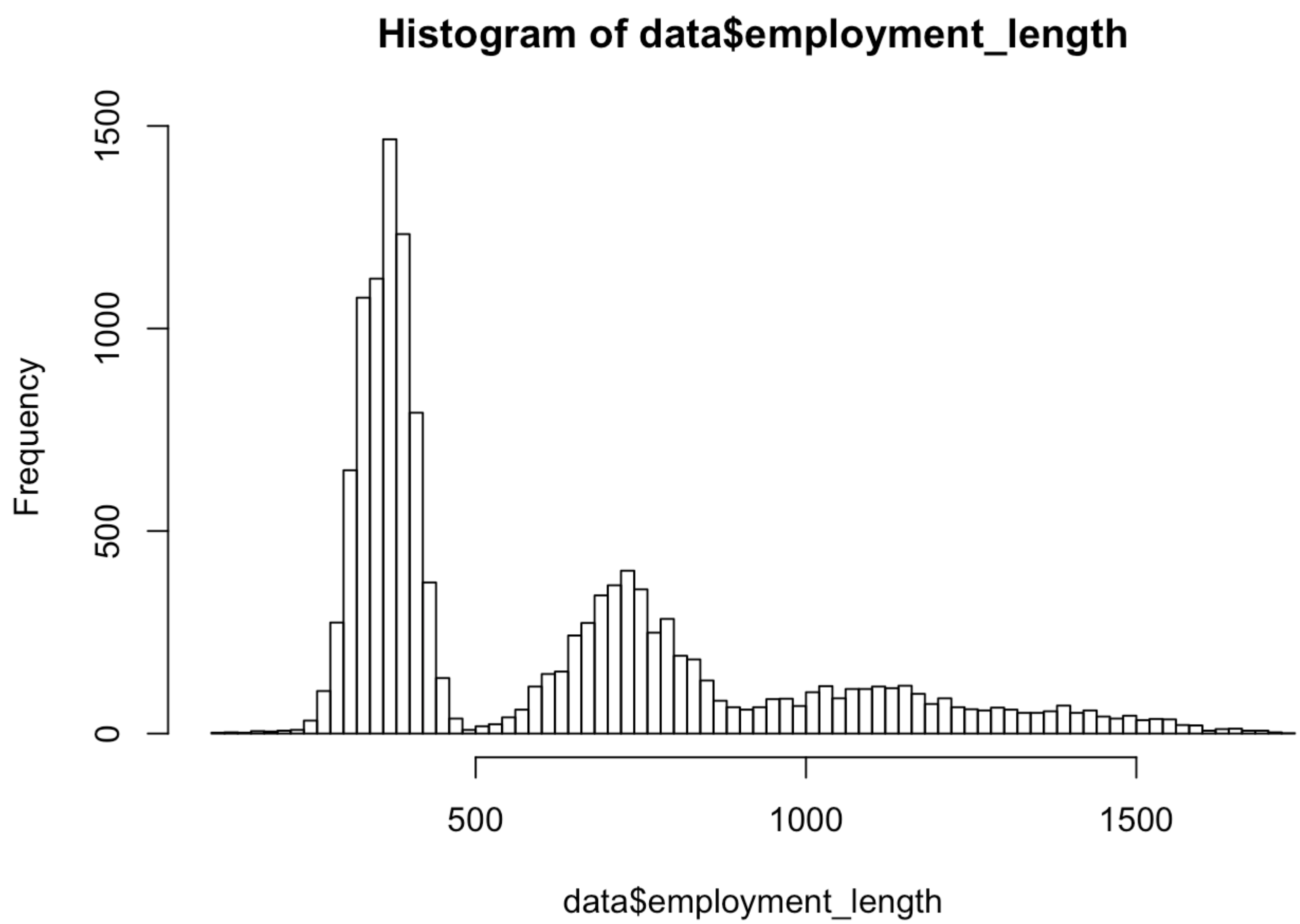
```
ate.

#Let's try with the for loop. Again here we optimize for future readibility!
# This is as slow as it can possibly be, but much clearer.

loop_cumsum = c() #intialize empty vector
loop_date = c()
loop_company = c()
for (i in seq(as.Date("2011/01/24"), as.Date("2015/12/13"), by = "day")) { #loop t
hrough all days
   for (j in unique(data$company_id)){ # loop through all companies
        tmp_join = nrow(subset(data, join_date <= i & company_id == j)) # count jo
ins until that day
        tmp_quit = nrow(subset(data, quit_date <= i & company_id == j)) # count qu
its
        loop_cumsum = c(loop_cumsum, tmp_join - tmp_quit )
        loop_date = c(loop_date, i)
        loop_company = c(loop_company, j)
   }
data_headcount_table_loop = data.frame(date = as.Date(loop_date, origin = '1970-0
1-01'), #fix R date
                                        company_id = loop_company,
                                        count = loop_cumsum)
}



#Let's finally check the two data sets are exactly the same:
identical (data_headcount_table[order(data_headcount_table[,1],
                                      as.numeric(as.character(data_headcount_tabl
e[,2] )))
                               ,],
            data_headcount_table[order(data_headcount_table[,1],
                                      as.numeric(as.character(data_headcount_table[,2]
)))
                                ,]
            )
```

```
## [1] TRUE
```

Now let’s try to understand employee retention. Here the main challenge is about feature engineering. That is, extract variables from the quitting_date column.

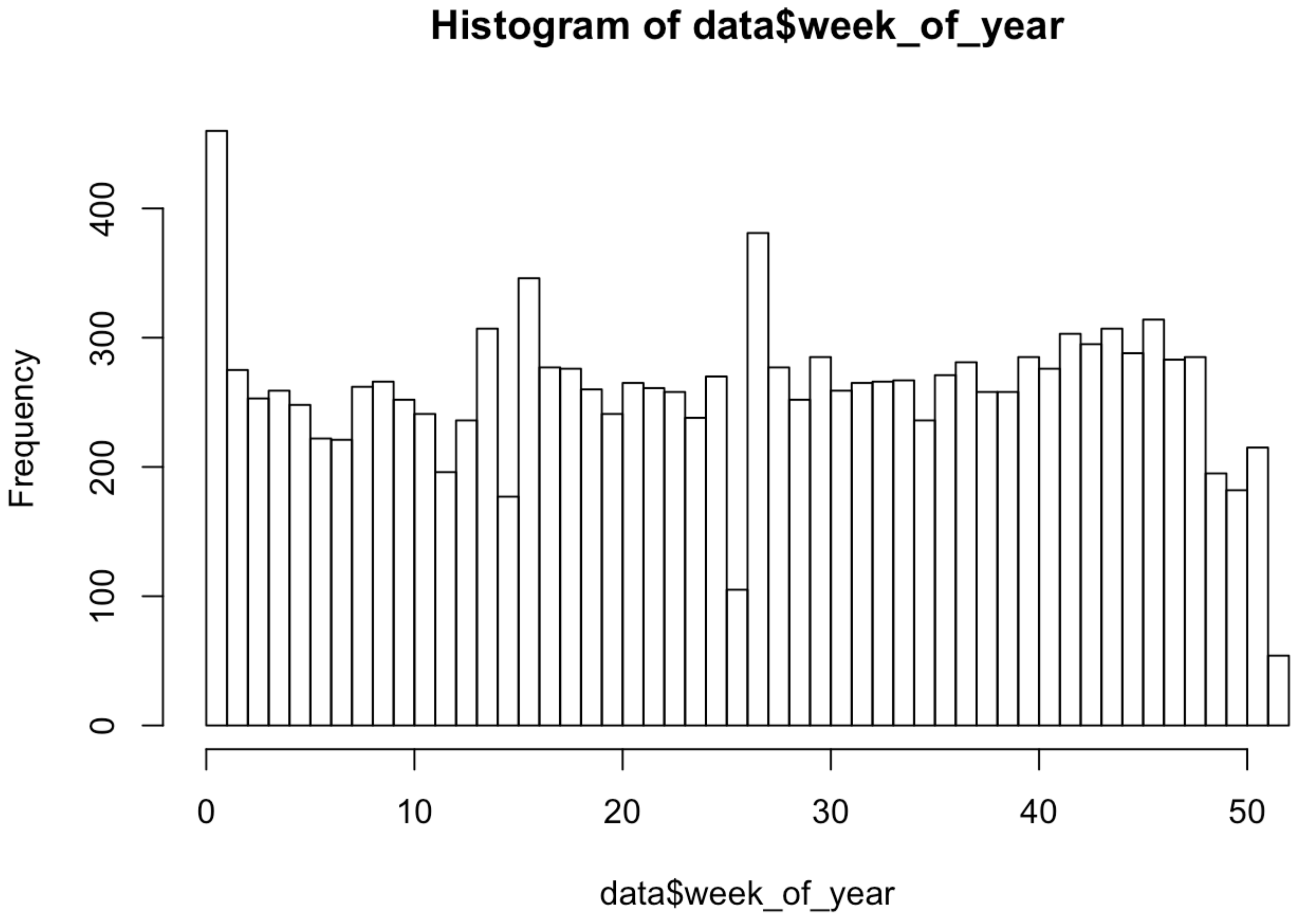
```
# how many days was she employed? This should matter.
#People might get bored in the same place for too long
data$employment_length = as.numeric(data$quit_date - data$join_date)

#In general, whenever you have a date, extract week of the year, and day of the we
ek. They tend to give an idea of seasonlity and weekly trends.
#In this case, weekly trends probably don't matter. So let's just get week of the
year
data$week_of_year = as.numeric(format(data$quit_date, "%U"))
```

```
#Let's plot employee length in days
hist(data$employment_length, breaks = 100)
```



Very interesting, there are peaks around each employee year anniversary!

```
#Let's plot week of the year
hist(data$week_of_year, breaks = length(unique(data$week_of_year)))
```



And it also peaks around the new year. Makes sense, companies have much more money to hire at the beginning of the year.

Now, let's see if we find the characteristics of the people who quit early. Looking at the histogram of employment_length, it looks like we could define early quitters as those people who quit within 1 yr or so. So, let's create two classes of users : quit within 13 months or not (if they haven't been in the current company for at least 13 months, we remove them).

```
#Create binary class
data = subset(data, data$join_date < as.Date("2015/12/13") - (365 + 31)) # only ke
ep people who had enough time to age
data$early_quitter = as.factor(ifelse( is.na(data$quit_date) | as.numeric(data$qui
t_date - data$join_date) > 396, 0, 1))
```

Let's now build a model. Here we can just care about: seniority, salary, dept and company. A simple decision tree is probably more than enough.

```r
tree = rpart(early_quitter ~ .,data[, c("company_id", "dept", "seniority", "earl
y_quitter", "salary")], #put salary
             control = rpart.control(minbucket = 30, maxdepth = 3 , cp = 0.00000
1),
             parms = list(prior = c(0.5,0.5))
             )
tree #we are not too interested in predictive power, we are mainly using the tree
as a descriptive stat tool.
```

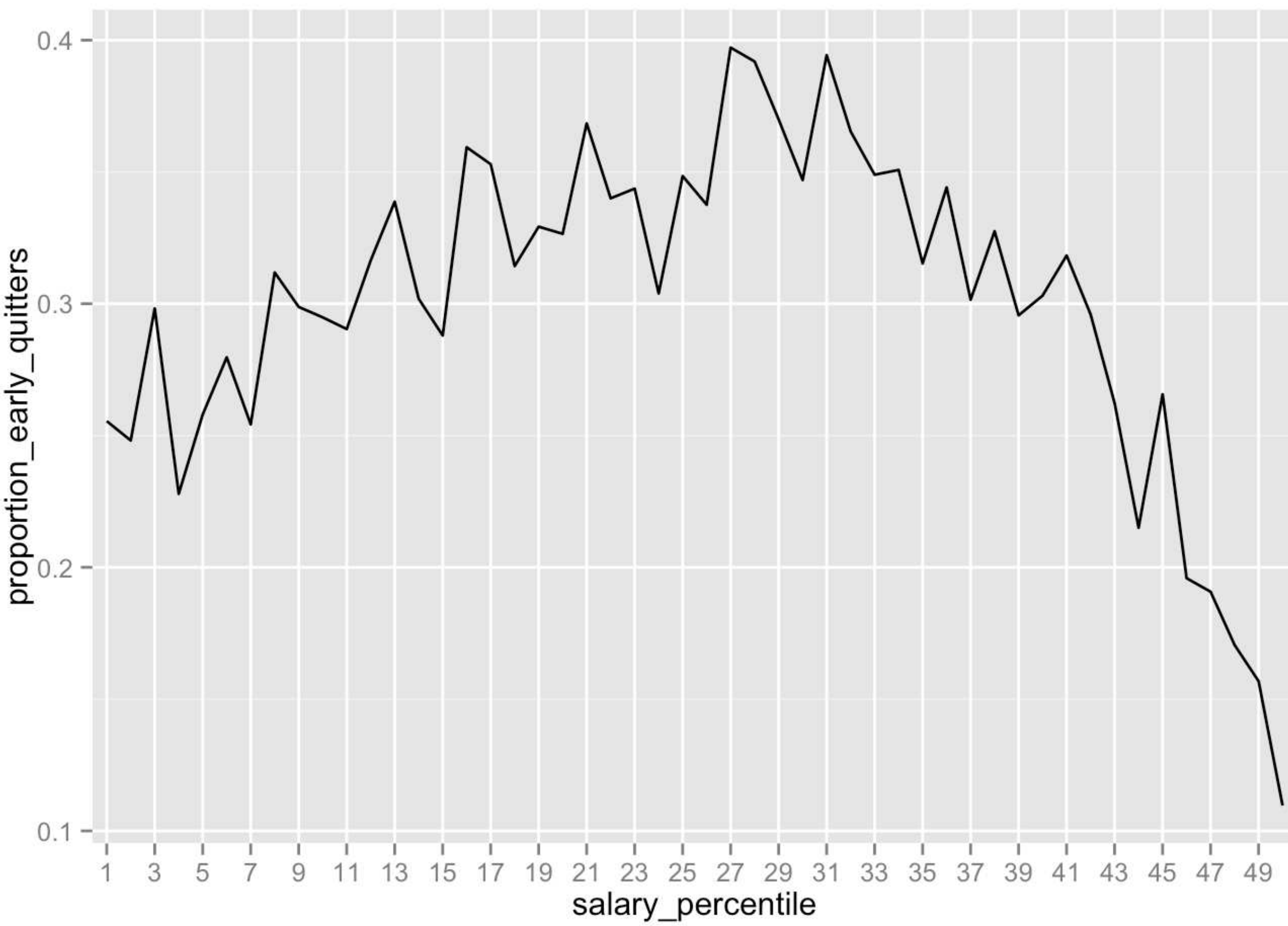
```
## n= 19270
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 19270 9635.0000 0 (0.5000000 0.5000000)
##   2) salary>=224500 2764  855.3351 0 (0.6528040 0.3471960) *
##   3) salary< 224500 16506 8026.7840 1 (0.4776014 0.5223986)
##     6) salary< 62500 2887 1249.7210 0 (0.5498859 0.4501141) *
##     7) salary>=62500 13619 6500.0510 1 (0.4632968 0.5367032) *
```

Not very surprising! Salary is what matters the most. After all, it probably has within it information about the other variables too. That is, seniority, dept and company impact salary. So salary carries pretty much all the information available.

It is interesting though that, looking at the terminal nodes, the way the tree split is: If salary between 62500 and 224500, the employee has higher probability of being an early quitter, otherwise she doesn't. That means that **people who make a lot of money and very little are not likely to quit** ("little money" by Silicon Valley standards).

By plotting the proportion of early quitter by salary percentile, this becomes quite clear:

```r
data$salary_percentile = cut(data$salary, breaks = quantile(data$salary, probs = s
eq(0, 1, 0.02)),
                             include.lowest = TRUE, labels = 1:50)
data_proportion_by_percentile = data %>%
                                group_by(salary_percentile) %>%
                                summarize(proportion_early_quitters = length(earl
y_quitter[early_quitter==1])/length(early_quitter)
                                          )
qplot(salary_percentile, proportion_early_quitters, data=data_proportion_by_percen
tile, geom="line", group =1) + scale_x_discrete(breaks = seq(1,50, by=2))
```



Conclusions

1. Given how important is salary, I would definitely love to have as a variable the salary the employee who quit was offered in the next job. Otherwise, things like: promotions or raises received during the employee tenure would be interesting.
2. The major findings are that employees quit at year anniversaries or at the beginning of the year. Both cases make sense. Even if you don't like your current job, you often stay for 1 yr before quitting + you often get stocks after 1 yr so it makes sense to wait. Also, the beginning of the year is well known to be the best time to change job: companies are hiring more and you often want to stay until end of Dec to get the calendar year bonus.
3. Employees with low and high salaries are less likely to quit. Probably because employees with high salaries are happy there and employees with low salaries are not that marketable, so they have a hard time finding a new job.