taskRabbithtml

Jay Whitmire

11/12/2018

Data Information

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.0.0     ✔ purrr   0.2.5
## ✔ tibble  1.4.2     ✔ dplyr   0.7.6
## ✔ tidyr   0.8.1     ✔ stringr 1.3.1
## ✔ readr   1.1.1     ✔ forcats 0.3.0
```

```
## ── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(broom)
library(rmarkdown)
library(ggplot2)
library(shiny)



task <- read_csv("https://raw.githubusercontent.com/jaywhitmire/myrepo/master/sample.cs
v")
```

```
## Parsed with column specification:
## cols(
##   recommendation_id = col_character(),
##   created_at = col_datetime(format = ""),
##   tasker_id = col_integer(),
##   position = col_integer(),
##   hourly_rate = col_integer(),
##   num_completed_tasks = col_integer(),
##   hired = col_integer(),
##   category = col_character()
## )
```

```
names(task)
```

```
## [1] "recommendation_id"   "created_at"          "tasker_id"
## [4] "position"            "hourly_rate"         "num_completed_tasks"
## [7] "hired"               "category"
```

```
head(task)
```

```
## # A tibble: 6 x 8
##   recommendation_… created_at          tasker_id position hourly_rate
##   <chr>            <dttm>                  <int>    <int>       <int>
## 1 0-0-70cf97d7-37… 2017-09-01 00:32:25    1.01e9        1          38
## 2 0-0-70cf97d7-37… 2017-09-01 00:32:25    1.01e9        2          40
## 3 0-0-70cf97d7-37… 2017-09-01 00:32:25    1.01e9        3          28
## 4 0-0-70cf97d7-37… 2017-09-01 00:32:25    1.01e9        4          43
## 5 0-0-70cf97d7-37… 2017-09-01 00:32:25    1.01e9        5          29
## 6 0-0-70cf97d7-37… 2017-09-01 00:32:25    1.01e9        6          28
## # ... with 3 more variables: num_completed_tasks <int>, hired <int>,
## #   category <chr>
```

```
str(task)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    30000 obs. of  8 variables:
##  $ recommendation_id  : chr  "0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c" "0-0-70cf97d7
-37af-4834-901c-ce3ad4893b8c" "0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c" "0-0-70cf97d7-3
7af-4834-901c-ce3ad4893b8c" ...
##  $ created_at         : POSIXct, format: "2017-09-01 00:32:25" "2017-09-01 00:32:25"
...
##  $ tasker_id          : int  1009185352 1006892359 1012023956 1009733517 1013579273 1
012043028 1013470741 1009557645 1010800768 1009072269 ...
##  $ position           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ hourly_rate        : int  38 40 28 43 29 28 29 29 28 35 ...
##  $ num_completed_tasks: int  151 193 0 303 39 2 9 8 0 59 ...
##  $ hired              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ category           : chr  "Furniture Assembly" "Furniture Assembly" "Furniture Ass
embly" "Furniture Assembly" ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 8
##   .. ..$ recommendation_id  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ created_at         :List of 1
##   .. .. ..$ format: chr ""
##   .. .. ..- attr(*, "class")= chr  "collector_datetime" "collector"
##   .. ..$ tasker_id          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ position           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ hourly_rate        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ num_completed_tasks: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ hired              : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ category           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

```
summary(task)
```

```
##  recommendation_id    created_at                    tasker_id
##  Length:30000       Min.   :2017-09-01 00:32:25   Min.   :1.007e+09
##  Class :character   1st Qu.:2017-09-08 23:10:23   1st Qu.:1.009e+09
##  Mode  :character   Median :2017-09-17 17:27:05   Median :1.011e+09
##                     Mean   :2017-09-16 23:38:30   Mean   :1.011e+09
##                     3rd Qu.:2017-09-24 21:24:44   3rd Qu.:1.013e+09
##                     Max.   :2017-09-30 23:15:51   Max.   :1.015e+09
##     position       hourly_rate     num_completed_tasks     hired
##  Min.   : 1.000   Min.   : 18.00   Min.   :   0.0      Min.   :0.00000
##  1st Qu.: 4.000   1st Qu.: 38.00   1st Qu.:  23.0      1st Qu.:0.00000
##  Median : 8.000   Median : 45.00   Median : 114.0      Median :0.00000
##  Mean   : 7.874   Mean   : 57.48   Mean   : 221.2      Mean   :0.05683
##  3rd Qu.:12.000   3rd Qu.: 60.00   3rd Qu.: 300.2      3rd Qu.:0.00000
##  Max.   :15.000   Max.   :290.00   Max.   :1406.0      Max.   :1.00000
##    category
##  Length:30000
##  Class :character
##  Mode  :character
##
##
##
```

```
glimpse(task)
```

```
## Observations: 30,000
## Variables: 8
## $ recommendation_id   <chr> "0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c"...
## $ created_at          <dttm> 2017-09-01 00:32:25, 2017-09-01 00:32:25,...
## $ tasker_id           <int> 1009185352, 1006892359, 1012023956, 100973...
## $ position            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...
## $ hourly_rate         <int> 38, 40, 28, 43, 29, 28, 29, 29, 28, 35, 40...
## $ num_completed_tasks <int> 151, 193, 0, 303, 39, 2, 9, 8, 0, 59, 68, ...
## $ hired               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...
## $ category            <chr> "Furniture Assembly", "Furniture Assembly"...
```

1. How many recommendation sets are in this data sample?

```
length(unique(task$recommendation_id))
```

```
## [1] 2100
```

2100 unique recommendation sets

2. Each recommendation set shows from 1 to 15 Taskers, what is:

- average number of Taskers shown

```
taskers_per_rec <- task %>%
  group_by(recommendation_id) %>%
  summarise(Taskers = n())

mean(taskers_per_rec$Taskers)
```

```
## [1] 14.28571
```

#####14.29

- median number of Taskers shown

```
median(taskers_per_rec$Taskers)
```

```
## [1] 15
```

15

3. How many total unique Taskers are there in this data sample?

```
length(unique(task$tasker_id))
```

```
## [1] 830
```

830 unique taskers

4. Which Tasker has been shown the most?

```
task %>%
  group_by(tasker_id) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

```
## # A tibble: 830 x 2
##     tasker_id count
##         <int> <int>
##  1 1014508755   608
##  2 1012043028   438
##  3 1014675294   387
##  4 1014629676   311
##  5 1007283421   290
##  6 1008887321   290
##  7 1015015238   290
##  8 1006892359   280
##  9 1008831520   277
## 10 1012721277   269
## # ... with 820 more rows
```

tasker_id 1014508755 has been shown 608 times

Which Tasker has been shown the least?

```
task %>%
  group_by(tasker_id) %>%
  summarise(count = n()) %>%
  arrange(count) %>%
  filter(count == 1)
```

```
## # A tibble: 68 x 2
##     tasker_id count
##         <int> <int>
##  1 1006690425     1
##  2 1006853970     1
##  3 1006899551     1
##  4 1007246122     1
##  5 1007295623     1
##  6 1007383273     1
##  7 1007472083     1
##  8 1007480912     1
##  9 1007638825     1
## 10 1007923586     1
## # ... with 58 more rows
```

The 68 taskers above are tied for the least showings with 1 each

5. Which Tasker has been hired the most?

```
task %>%
  group_by(tasker_id) %>%
  summarise(hirings = sum(hired)) %>%
  arrange(desc(hirings))
```

```
## # A tibble: 830 x 2
##     tasker_id hirings
##         <int>   <int>
##  1 1012043028      59
##  2 1013131759      39
##  3 1013359522      37
##  4 1013165984      36
##  5 1013794735      35
##  6 1014877120      32
##  7 1009530281      29
##  8 1009062537      25
##  9 1009558627      25
## 10 1013443601      22
## # ... with 820 more rows
```

tasker_id 1012043028 has been hired 59 times

Which Tasker has been hired the least?

```
task %>%
  group_by(tasker_id) %>%
  summarise(hirings = sum(hired)) %>%
  arrange(hirings) %>%
  filter(hirings == 0)
```

```
## # A tibble: 518 x 2
##     tasker_id hirings
##         <int>   <int>
##  1 1006646767       0
##  2 1006655883       0
##  3 1006690425       0
##  4 1006702141       0
##  5 1006720473       0
##  6 1006751673       0
##  7 1006771484       0
##  8 1006797028       0
##  9 1006808958       0
## 10 1006853970       0
## # ... with 508 more rows
```

the 518 taskers above have not been hired at all

6. If we define the “Tasker conversion rate” as the number of times a Tasker has been hired, out of the number of times the Tasker has been shown, how many Taskers have a conversion rate of 100%

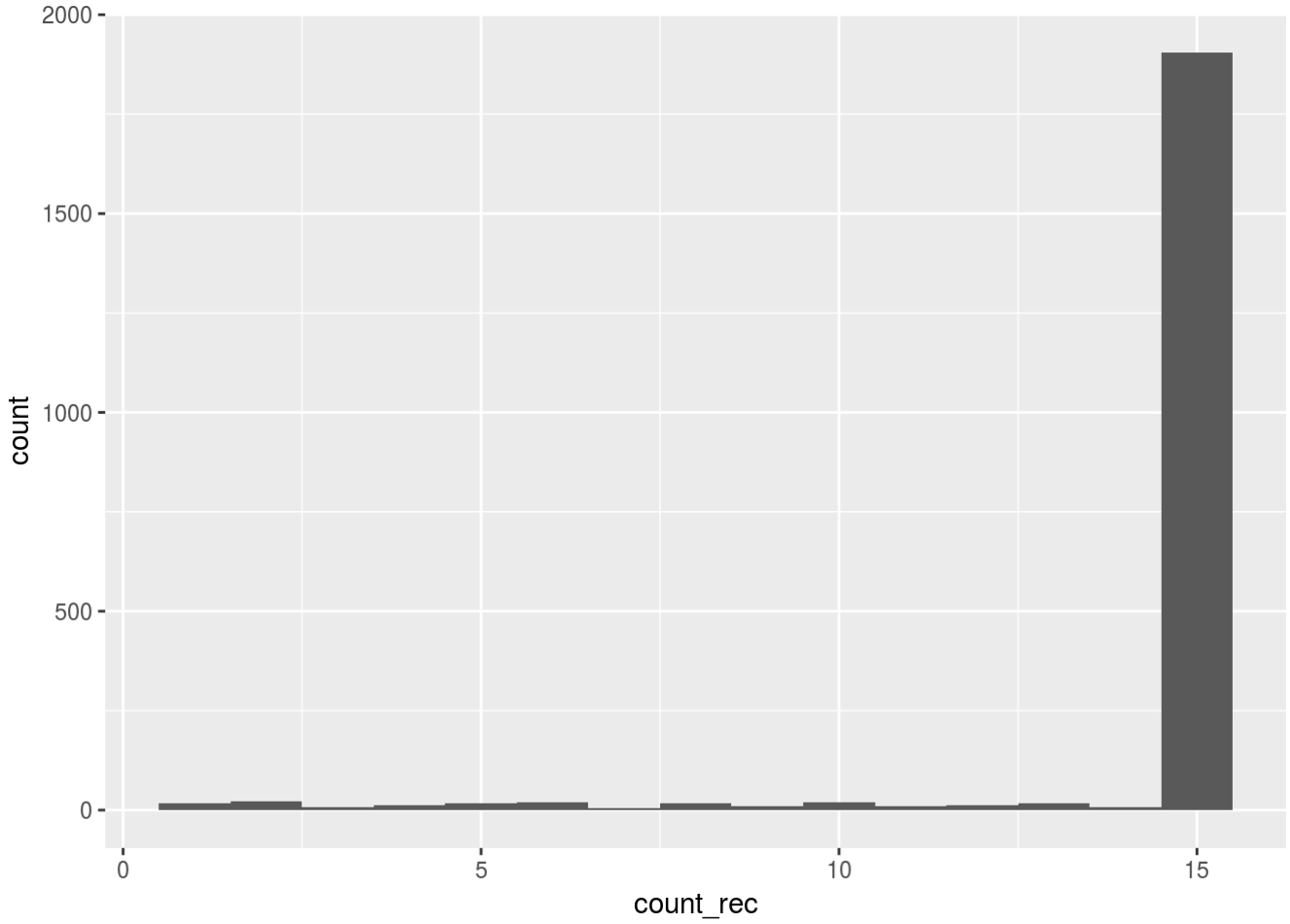
```
task %>%
  group_by(tasker_id) %>%
  summarise(count = n(),
            hirings = sum(hired),
            conversion_rate = hirings / count * 100) %>%
  filter(conversion_rate == 100) %>%
  nrow()
```

```
## [1] 6
```

6 taskers have a conversion rate of 100%

7. Would it be possible for all Taskers to have a conversion rate of 100% Please explain your reasoning.

```
task %>%
  group_by(recommendation_id) %>%
  summarise(count_rec = n(),
            hirings = sum(hired),
            conversion_rate = hirings / count_rec * 100) %>%
  arrange(desc(conversion_rate)) %>%
  ggplot(aes(x = count_rec)) +
  geom_histogram(bins = 15)
```



Only if there was one recommended tasker per recommendation and that person was hired everytime. Otherwise its not possible since only one tasker can get hired per posting

8. For each category, what is the average position of the Tasker who is hired?

```
task %>%
  filter(hired == 1) %>%
  group_by(category) %>%
  summarise(avg_position = mean(position))
```

```
## # A tibble: 3 x 2
##   category           avg_position
##   <chr>                     <dbl>
## 1 Furniture Assembly         3.61
## 2 Mounting                   4.60
## 3 Moving Help                4.15
```

Furniture Assembly = 3.61, Mounting = 4.60, Moving Help = 4.15

9. For each category, what is the average hourly rate and average number of completed tasks for the Taskers who are hired?

```
task %>%
  filter(hired == 1) %>%
  group_by(category) %>%
  summarise(avg_hourly_rate = mean(hourly_rate),
            avg_completed_tasks = mean(num_completed_tasks))
```

```
## # A tibble: 3 x 3
##   category           avg_hourly_rate avg_completed_tasks
##   <chr>                        <dbl>               <dbl>
## 1 Furniture Assembly            38.7                249.
## 2 Mounting                      50.2                284.
## 3 Moving Help                   63.0                274.
```

10. Based on the previous, how would you approach the question of:

How can we use market data to suggest hourly rates to Taskers that would maximize their opportunity to be hired?

Please describe in detail, with code and formulas that support your model.

To build a a model that recommends an hourly rate first we must build a model that predicts hourly rate. We will filter the data to only winners then use multiple regression using category, position, and number of completed tasks to predict hourly rate, then use the model output to calcualte a recommended rate

```
### Model using data only from winners

winners <- task %>%
  filter(hired == 1)

model_winners <- lm(hourly_rate ~ category + position + num_completed_tasks, data = tas
k)
summary(model_winners)
```

```
##
## Call:
## lm(formula = hourly_rate ~ category + position + num_completed_tasks,
##     data = task)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -65.245 -12.800  -2.261   6.334 261.753
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.486e+01  4.438e-01   56.01   <2e-16 ***
## categoryMounting     1.002e+01  4.200e-01   23.85   <2e-16 ***
## categoryMoving Help 4.096e+01  4.218e-01   97.10   <2e-16 ***
## position             1.129e+00  3.966e-02   28.48   <2e-16 ***
## num_completed_tasks 3.046e-02  6.157e-04   49.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.66 on 29995 degrees of freedom
## Multiple R-squared:  0.3295, Adjusted R-squared:  0.3294
## F-statistic:  3685 on 4 and 29995 DF,  p-value: < 2.2e-16
```

```
tidy(model_winners)
```

```
## # A tibble: 5 x 5
##   term                estimate std.error statistic   p.value
##   <chr>                  <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)          24.9     0.444         56.0 0.
## 2 categoryMounting     10.0     0.420         23.8 1.54e-124
## 3 categoryMoving Help  41.0     0.422         97.1 0.
## 4 position              1.13    0.0397        28.5 5.02e-176
## 5 num_completed_tasks   0.0305  0.000616      49.5 0.
```

```
task$recommended_rate <- predict(model_winners, newdata = task)

task %>%
  data.frame() %>%
  select(tasker_id, category, position, num_completed_tasks, hourly_rate, recommended_ra
te) %>%
  head(10)
```

```
##     tasker_id           category position num_completed_tasks hourly_rate
## 1  1009185352 Furniture Assembly        1                 151          38
## 2  1006892359 Furniture Assembly        2                 193          40
## 3  1012023956 Furniture Assembly        3                   0          28
## 4  1009733517 Furniture Assembly        4                 303          43
## 5  1013579273 Furniture Assembly        5                  39          29
## 6  1012043028 Furniture Assembly        6                   2          28
## 7  1013470741 Furniture Assembly        7                   9          29
## 8  1009557645 Furniture Assembly        8                   8          29
## 9  1010800768 Furniture Assembly        9                   0          28
## 10 1009072269 Furniture Assembly       10                  59          35
##    recommended_rate
## 1          30.58807
## 2          32.99694
## 3          28.24695
## 4          38.60672
## 5          31.69384
## 6          31.69611
## 7          33.03877
## 8          34.13772
## 9          35.02343
## 10         37.95018
```

This model only explains 30% of the variance in hourly rate prices so we should look to add variables to explain a larger percentage of the variance. These might be the location, time of year, relative supply and demand in the local market, job degree of difficulty, and mining text data like user reviews for key words and sentiment.