Mark D. Adams

Data Exercise

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

QUERY

```
SELECT COUNT(DISTINCT recommendation_id)
  FROM recommendations;
```

ANSWER

2,100

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

- average number of Taskers shown
- median number of Taskers shown

QUERY

```
CREATE TEMPORARY TABLE temp_distribution AS
   SELECT COUNT(tasker_id) AS taskers
   FROM recommendations
   GROUP BY recommendation_id;

 SELECT AVG(taskers) FROM temp_distribution;

 SELECT AVG(taskers)
   FROM
   ( -- order with row_id
     SELECT ROW_NUMBER() OVER (ORDER BY taskers) AS row_id
     , taskers
     FROM temp_distribution
   ) a
 WHERE row_id BETWEEN 2100 / 2 AND 2100 / 2 +1;

 DROP TABLE temp_distribution;
```

ANSWER

Average 14.2857142857142857

Median 15.0

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

QUERY

```
SELECT COUNT(DISTINCT tasker_id)
FROM recommendations;
```

ANSWER

830

4. Which Tasker has been shown the most? Which Tasker has been shown the least?

QUERY

```
 CREATE TEMPORARY TABLE temp_distribution AS
     SELECT tasker_id
     , COUNT(recommendation_id) AS seen
     , SUM(hired::int) AS hired -- my hired got imported as varchar by mistake
     FROM recommendations
     WHERE position <= 15 -- in case you record those that didn't display
     GROUP BY tasker_id;

   SELECT tasker_id
     , seen
     FROM temp_distribution
     WHERE seen = (SELECT MAX(seen) FROM temp_distribution)
       OR seen = (SELECT MIN(seen) FROM temp_distribution)
     ORDER BY SEEN DESC;
```

DISCUSSION

Maybe you wanted the least sorted by a second value?

ANSWER

Most = tasker_id 1014508755

Least = tasker_id (1006853970, 1006899551, 1007246122, 1007295623, 1007383273, 1007472083, 1007480912, 1007638825, 1007923586, 1008033678, 1008368716, 1008469117, 1008474216, 1008604368, 1008828652, 1008870833, 1008919567, 1009112003, 1009461190, 1009547227, 1009603880, 1009612428, 1009618500, 1009641175, 1009702351, 1009712638, 1009754999, 1009772528, 1009871933, 1009994950, 1010009736, 1010021990, 1010042971, 1010640007, 1010681878, 1010779242, 1011901532, 1011949117, 1011952623, 1011957940, 1011968845, 1011972750, 1011985968, 1012071620, 1012151299, 1012166729, 1012289475, 1012348656, 1012364558, 1012386513, 1012678504, 1012805440, 1013362004, 1013573125, 1013573988, 1013656032, 1013731883, 1013830691, 1013854788, 1013934937, 1014086818, 1014310300, 1014439502, 1014478773, 1014547884, 1014593279, 1006690425, 1014926743)

5. Which Tasker has been hired the most? Which Tasker has been hired the least?

QUERY

```
 SELECT tasker_id
      , hired
      FROM temp_distribution
      WHERE hired = (SELECT MAX(hired) FROM temp_distribution)
        OR hired = (SELECT MIN(hired) FROM temp_distribution)
    ORDER BY hired DESC;
```

ANSWER

Most = tasker_id 1012043028

Least = tasker_id (list of 518)

DISCUSSION

Maybe you wanted the least sorted by a second value?

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%

QUERY

```
SELECT COUNT(*) AS converted_100
    FROM temp_distribution
    WHERE hired = seen -- simplifying the ratio
    AND seen > 0; -- you probably want this


  DROP TABLE temp_distribution;
```

ANSWER

6

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

ANSWER

Yes.

If only one tasker was shown for each recommendation and someone was always hired, the CR would be 100% for all taskers. This would be a bug, or a problem with the number of taskers available for a project.

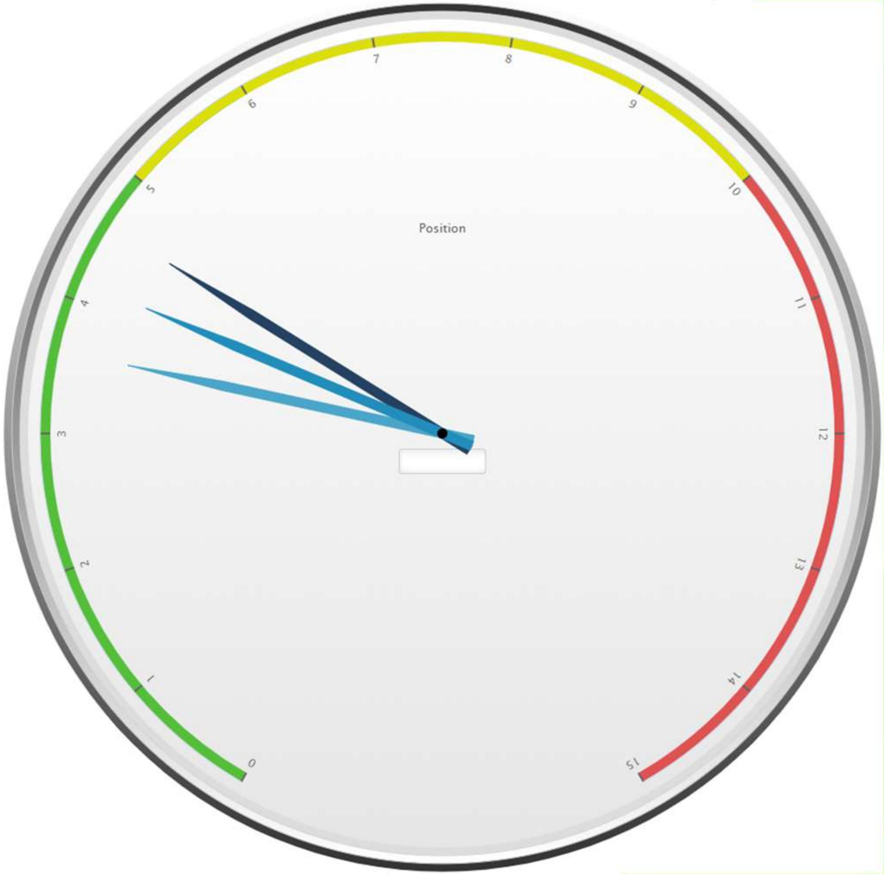
8. For each category, what is the average position of the Tasker who is hired?

QUERY

```
 SELECT category
      , AVG(position) AS avg_position
      FROM recommendations
      WHERE hired::int = 1
      GROUP BY category;
```

ANSWER

category	avg_position
Furniture Assembly	3.6118881118881119
Mounting	4.5960854092526690
Moving Help	4.1453590192644483

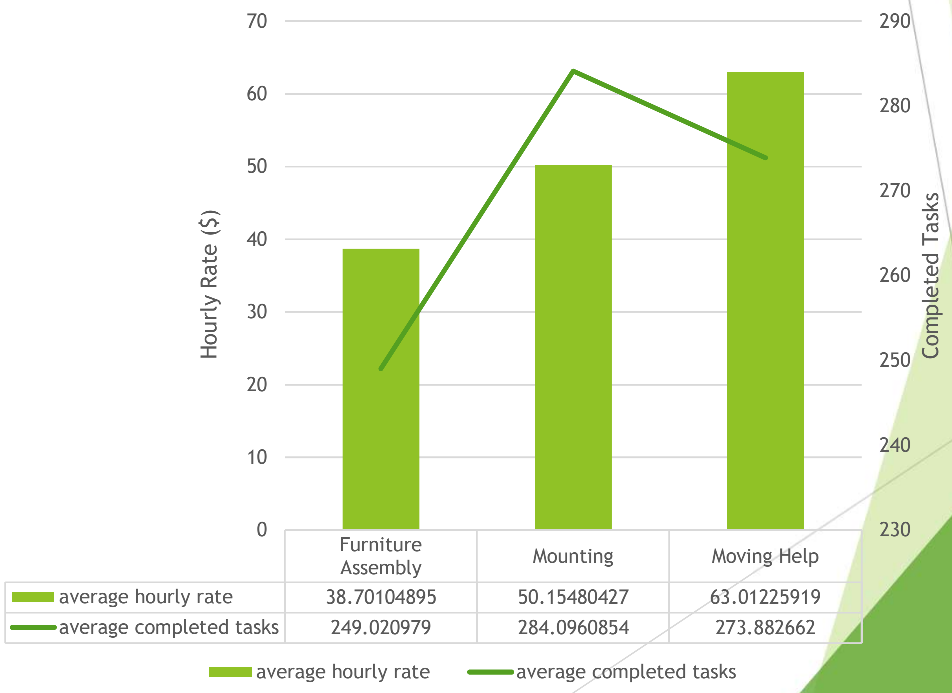


9. For each category, what is the average hourly rate and average number of completed tasks for the Taskers who are hired?

QUERY

```
 SELECT category
      , AVG(hourly_rate) AS avg_rate
      , AVG(num_completed_tasks) AS avg_comp_tasks
      FROM recommendations
      WHERE hired::int = 1
      GROUP BY category;
```

ANSWER



	Furniture Assembly	Mounting	Moving Help
🟩 average hourly rate	38.70104895	50.15480427	63.01225919
━ average completed tasks	249.020979	284.0960854	273.882662

DISCUSSION
Maybe you wanted the least sorted by a second value?

10. 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.

ANSWER

To optimize profit for the taskers you would only want to move the outliers toward the middle. Additionally, you would also want to push the price up slowly until demand plateaued to establish the pareto optimal solution, but I realize that this is not the goal of this question.

One issue you could run into optimizing for competition is the Prisoner’s Dilemma.

https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

Since your taskers are changing their prices manually, you will avoid a lot of the issues of an automated price war, but we need to be careful on how we recommend pricing. A price war could cause an influx of new customers and repeat business, though.

RECOMMENDED PRICING

What I would do is show them the average number of hires a day/week that each price is getting broken down by category. That way they can optimize their price based on how many tasks they want to do a day/week, and it could prevent price wars, as conceivably different taskers would be optimizing for different numbers of tasks a day/week. I might also show them the bracket/bucket that they're currently in.

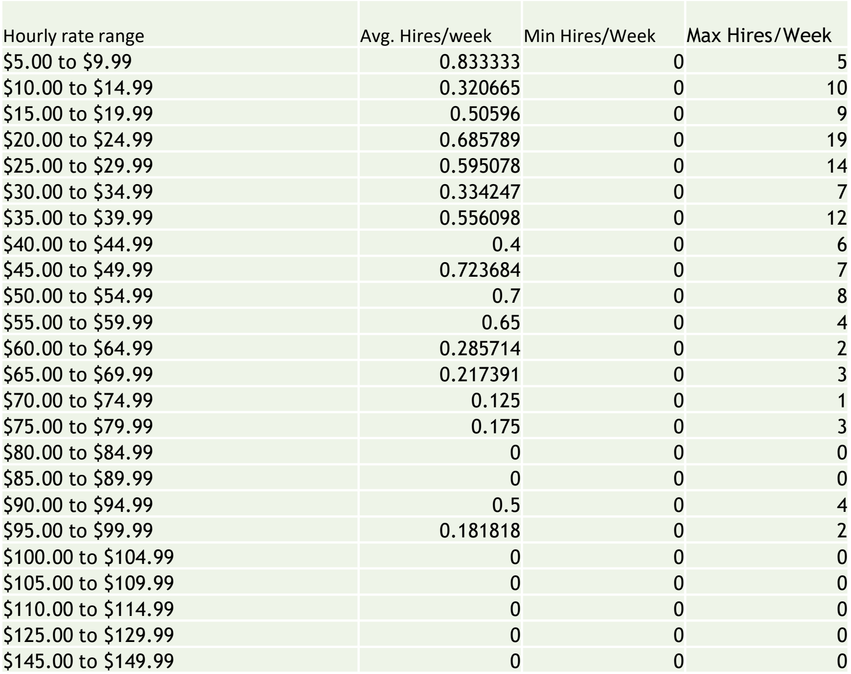
Or simpler, ask them how many tasks they want a day/week and recommend a range to them.

Note that some important details are missing such as zip code, as I'm sure that price varies by location, and the number of hours the person was ultimately hired for would be valuable here.

QUERY

QUERY

```
 SELECT category
      , hourly_rate / 5 AS bracket
      , AVG(hired) AS avg_hired
      , MIN(hired) AS min_hired
      , MAX(hired) AS max_hired
      FROM
    (
    SELECT tasker_id
      , category
      , DATE_TRUNC('week', created_at) AS week
      , hourly_rate
      , SUM(hired::int) AS hired
      FROM recommendations
      GROUP BY tasker_id, category, week, hourly_rate
      HAVING MIN(hourly_rate) = MAX(hourly_rate) -- only use data if
tasker had same rate for entire week
    ) a
      GROUP BY category, bracket
      ORDER BY category, bracket;
```



Hourly rate range	Avg. Hires/week	Min Hires/Week	Max Hires/Week
$5.00 to $9.99	0.833333	0	5
$10.00 to $14.99	0.320665	0	10
$15.00 to $19.99	0.50596	0	9
$20.00 to $24.99	0.685789	0	19
$25.00 to $29.99	0.595078	0	14
$30.00 to $34.99	0.334247	0	7
$35.00 to $39.99	0.556098	0	12
$40.00 to $44.99	0.4	0	6
$45.00 to $49.99	0.723684	0	7
$50.00 to $54.99	0.7	0	8
$55.00 to $59.99	0.65	0	4
$60.00 to $64.99	0.285714	0	2
$65.00 to $69.99	0.217391	0	3
$70.00 to $74.99	0.125	0	1
$75.00 to $79.99	0.175	0	3
$80.00 to $84.99	0	0	0
$85.00 to $89.99	0	0	0
$90.00 to $94.99	0.5	0	4
$95.00 to $99.99	0.181818	0	2
$100.00 to $104.99	0	0	0
$105.00 to $109.99	0	0	0
$110.00 to $114.99	0	0	0
$125.00 to $129.99	0	0	0
$145.00 to $149.99	0	0	0

This is simplified by removing category and doing $10 brackets.

This data would be less noisy with a bigger data set, providing that price is a driving factor to the hire decision.

CAVEATS

Adjusting pricing of course only works for price sensitive customers, and some other major factors will be:

1. Competition (how many other taskers were in the same recommendation)
2. Display order
3. Sentiment/social proof data displayed with the tasker (ratings, # completed of tasks)
4. Quality of the tasker’s image/bio
5. Location
6. Other demographic information
7. Tasker availability
8. How many taskers didn’t make the list of 15 even though they were in parameters?
9. Repeat business from satisfied customers

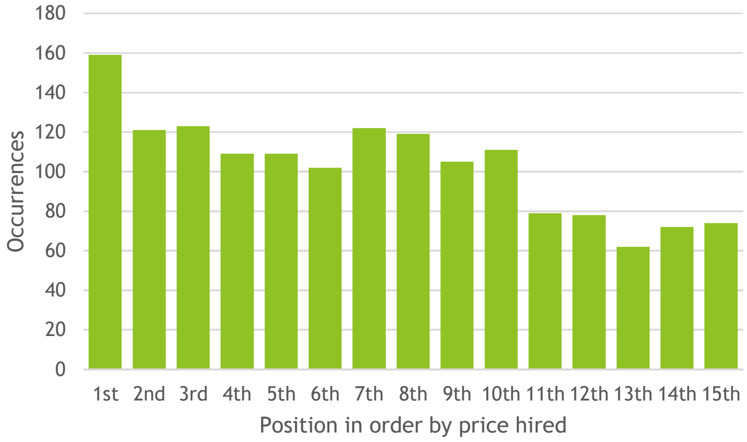
SUMMARY

More analysis definitely needs to be done on how to approach the goal of recommending a price to taskers to “maximize their opportunity to be hired”.

I think we really want to educate them on what a successful tasker looks like and give them a dashboard that shows their number of recommendations and conversion rate over time.

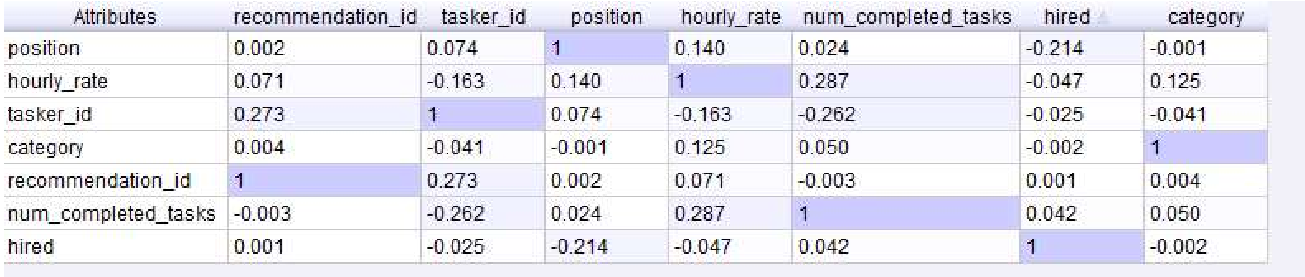
How important is price in a recommendation hire decision?

```
 SELECT price_order
      , SUM(hired::int) AS hired
      FROM
    (
    SELECT ROW_NUMBER() OVER (PARTITION BY recommendation_id ORDER BY
hourly_rate) AS price_order
      , hired
      , COUNT(*) OVER (PARTITION BY recommendation_id) AS total_taskers
      FROM recommendations
    ) a
      WHERE total_taskers = 15
      GROUP BY price_order
      ORDER BY price_order;
```

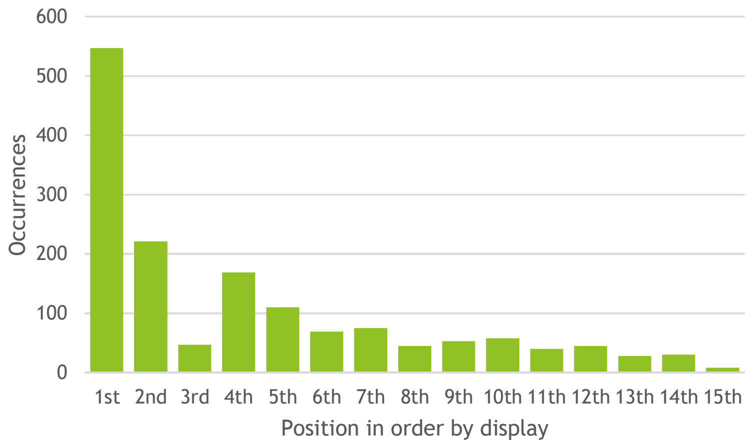


Not very, based on this dataset

This weighted correlation matrix shows that display position actually has a 4-5 times heavier correlation on hiring than hourly rate. So really, people are sorting on what’s important to them, and it’s not always price.



Attributes	recommendation_id	tasker_id	position	hourly_rate	num_completed_tasks	hired	category
position	0.002	0.074	1	0.140	0.024	-0.214	-0.001
hourly_rate	0.071	-0.163	0.140	1	0.287	-0.047	0.125
tasker_id	0.273	1	0.074	-0.163	-0.262	-0.025	-0.041
category	0.004	-0.041	-0.001	0.125	0.050	-0.002	1
recommendation_id	1	0.273	0.002	0.071	-0.003	0.001	0.004
num_completed_tasks	-0.003	-0.262	0.024	0.287	1	0.042	0.050
hired	0.001	-0.025	-0.214	-0.047	0.042	1	-0.002



Distribution shows the same, also people don’t like the third place?

Questions?

- mark@taskrabbit.com (coming soon)
- mark@bigdatamark.com
- Cell: 707-205-0982
- https://www.bigdatamark.com/