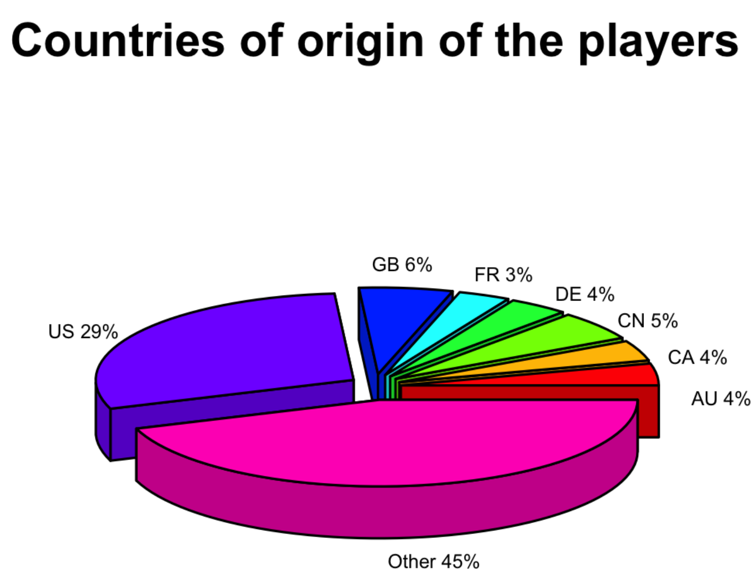
# Data-Scientist Test Data

A video-game company created this small database to test candidates who apply to a data-scientist position. I wrote the *video_game_data.R* R script to access these data through SQL queries (using both the RSQLite and sqldf packages).

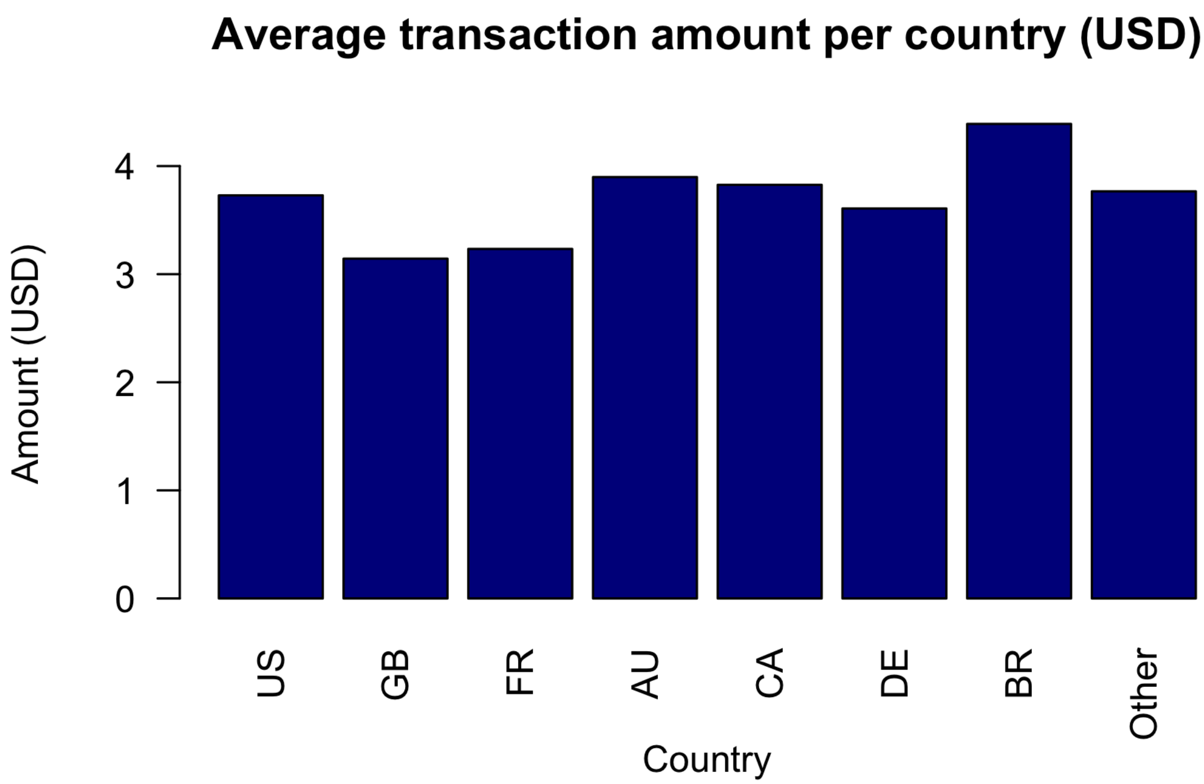
First, here are the answers to the test questions (units of quantities like the cash amount are not provided in the test data, but we can assume revenue is in USD):

```
Total revenue for 2013/02/01: 159.64 USD
Number of users using two different devices:  0
The country producing the most revenue is: US
The iPad/iPhone split in Canada is: 205 iPads, and: 328 iPhones
The proportion of lifetime revenue generated during the first week
is: 74.40579 percent
```

For the visualization part, first is the repartition of players by country. We only show the first seven countries (in term of numbers), and we list all of the others under the “other” label.

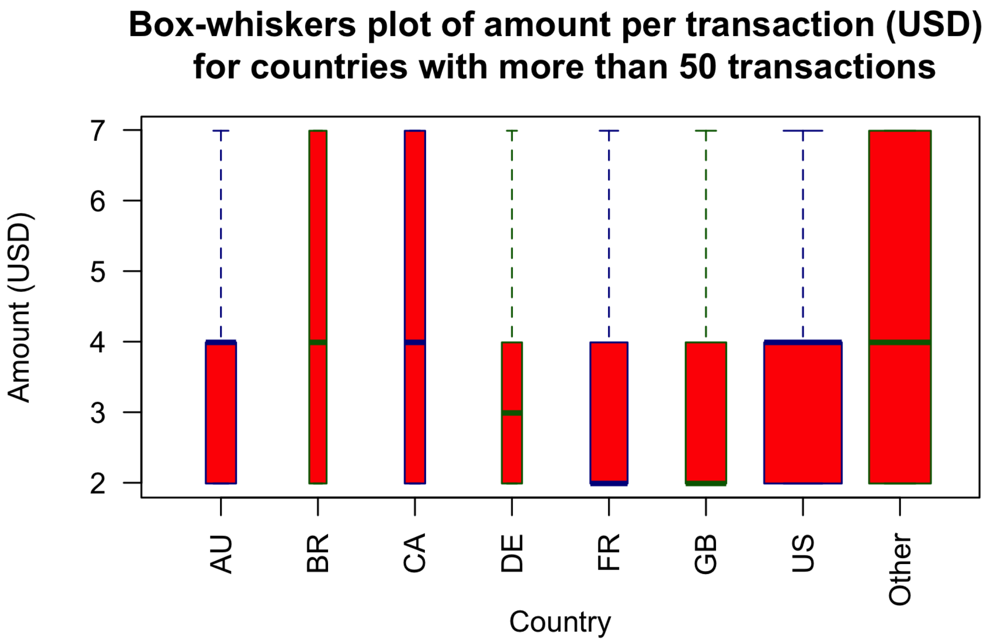


Unsurprisingly, the US is their biggest market in number of users (it's also the largest in term of revenue as previously mentioned). Let's look at the average revenue per country per transaction (we only show the seven countries with the most transactions. You will notice that those are not exactly the same as the countries with the most players, previously shown):




All of these countries are similar in term of average spending per user per transaction, although BR (Brazil?) seems to bring more cash. The average transaction in BR is 4.39 USD (but in-app purchases come with only three price tags: 1.99, 3.99, and 6.99), and with a standard error on this mean of 0.23 USD. That's a relatively large uncertainty on this sample mean, due to the low number of transactions. If the fact that BR spends more per transaction is confirmed with more transactions, it might be worth it for the company to advertise in BR in an effort to gain more customers.

Let's further study the distribution of spending by user per transaction, again for the countries with the most transactions. A box-and-whiskers plot (next page) can help visualize basic information about such distributions.




In this case though, it is not very useful since the prices fall in only three categories. Still, it's a convenient way to emphasize that, for instance, in the country with the most transactions (the US) the 3rd quartile is 3.99 (USD) (75% of the players spend 3.99 or less per transaction). This 3rd quartile is lower than for the "other" category. On the figure, the width of the boxes is proportional to the number of transaction per country. You will notice that these distributions appear strongly skewed. Again, box plots are not very informative in this case.
Also, let's look at a simple frequency table (number of transactions per price tag) of all the transactions in the database (all countries together):

| 1.99 USD | 3.99 USD | 6.99 USD |
|---|---|---|
| 1717 | 1255 | 816 |

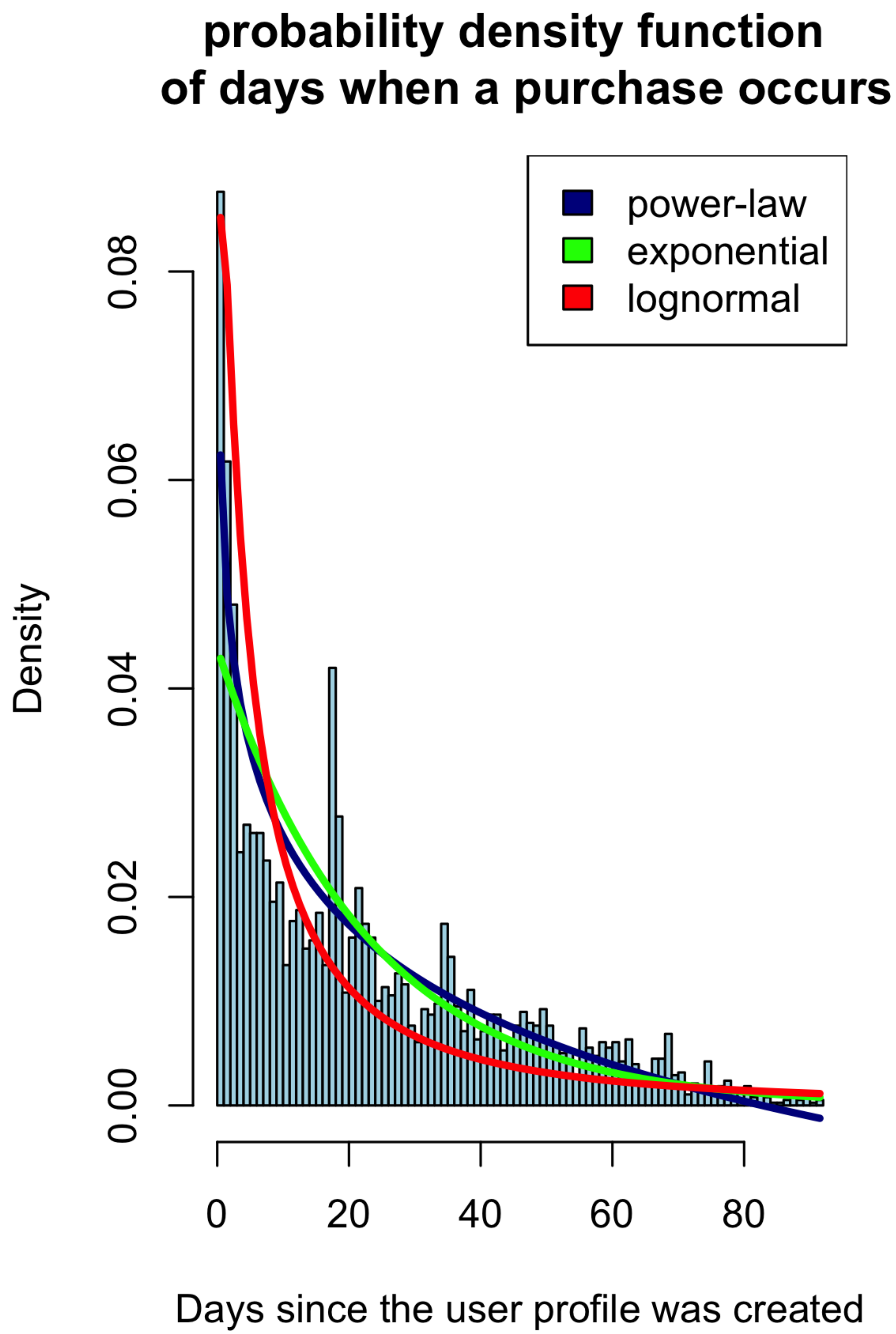
Overall, 45% of the transactions are for the cheapest in-app purchase price (1.99 USD). There are more than twice as many transactions at 1.99 USD than at 6.99 USD.
However, the 6.99 USD transactions bring 67% more revenue than the 1.99 USD ones. Maybe it would be worth it for the company to create another price tag, say 7.99 USD, and bring in even more revenue? A linear regression of the number of transactions as a function of purchase price (not the best model, but we only have 3 points) returns a prediction of 612 transactions at a 7.99 USD price tag.

Assuming that the trend in number of transactions as a function of price tag is confirmed at a price tag of 7.99 USD, that would bring in an estimated revenue of 4889 USD. This is less than the 5704 USD brought by the 6.99 USD price tag, but more than the 3417 USD brought by the 1.99 USD price tag, and the 5007 USD brought by the 3.99 USD price tag. In short, it looks like enough players might be willing to make in-app purchases at a higher price: the company should consider adding higher price tags. Alternatively, the company could replace the 3.99 USD price by a higher one (say, 4.99 USD: that would bring an estimated 5711 USD, i.e. 14% than the 3.99 USD price tag).

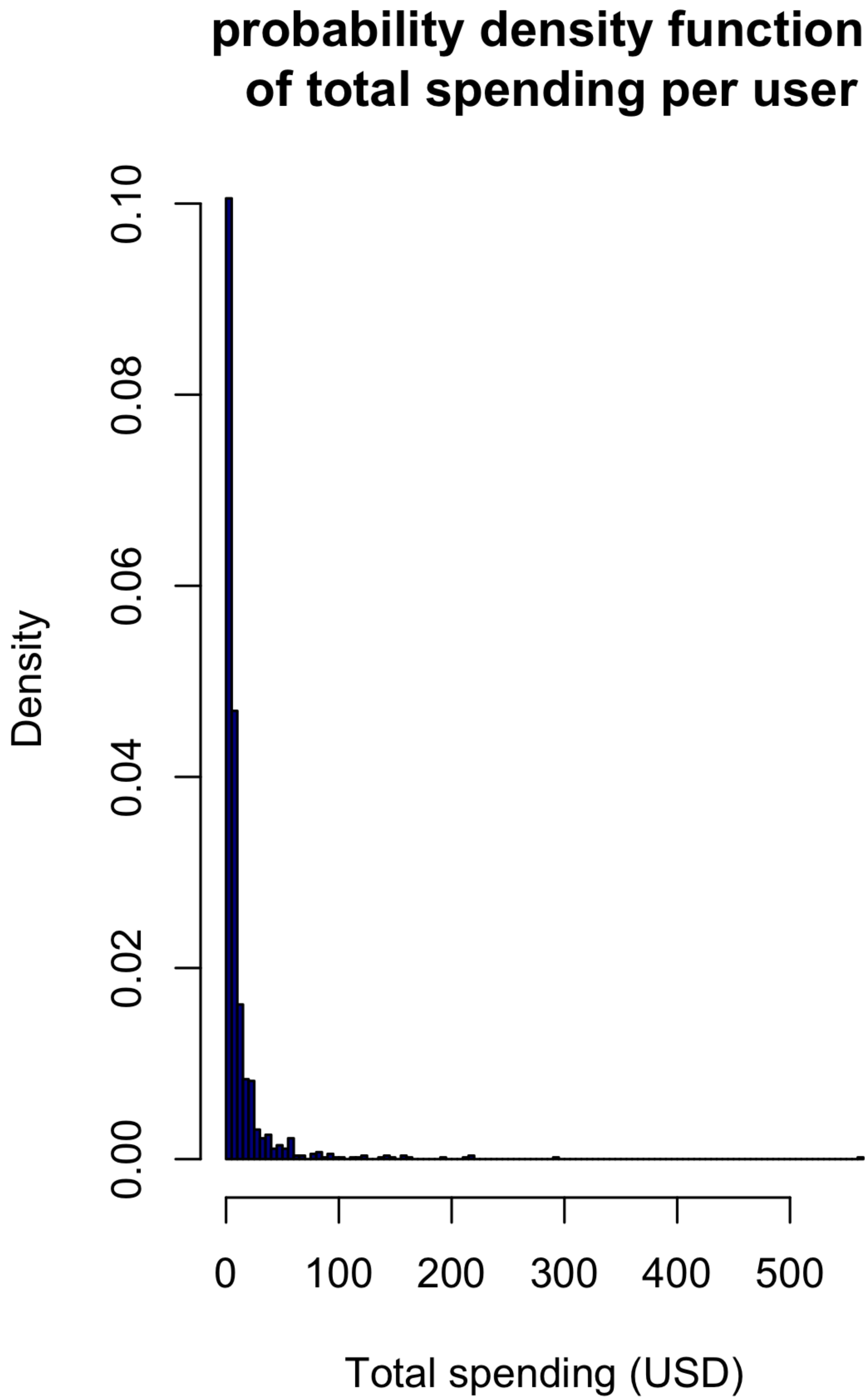
Next, we can study the distribution of the number of days from account creation when an in-app purchase is made. The probability density function (pdf) is shown on the next page. This pdf was fitted by three different distributions: power-law, exponential, and lognormal. A basic "sum of squares" argument to decide which distribution is the best fit returns the power law (alternatively a Kolmogorov-Smirnov test could have been performed). However, the lognormal seems to do a better job at fitting the tail of the distribution. This has implications on what the company should do with its customers: the lognormal distribution implies that a player may still make in-app purchases more than 80 days after this player started to play. Therefore, the company should keep engaging such players (even if, clearly, the vast majority of purchases are made within the first days after the player created his/her account).

On the contrary, the power law quickly drops to zero around 80 days, implying that players having used the game for at least 80 days are not going to make any more in-app purchase.

Overall 25% of in-app purchases are made within 5 days of a user creating an account, and 50% are made within 18 days. After that, the number of purchases drops quickly: if 31% of purchases are made after the first month, only 7.4% are made after the second month, and less than 0.1% are made after the third month. Therefore, the company should find ways to engage the players early on after they create their user account, and provide them with plenty of opportunities to realize in-app purchases as quickly as possible.

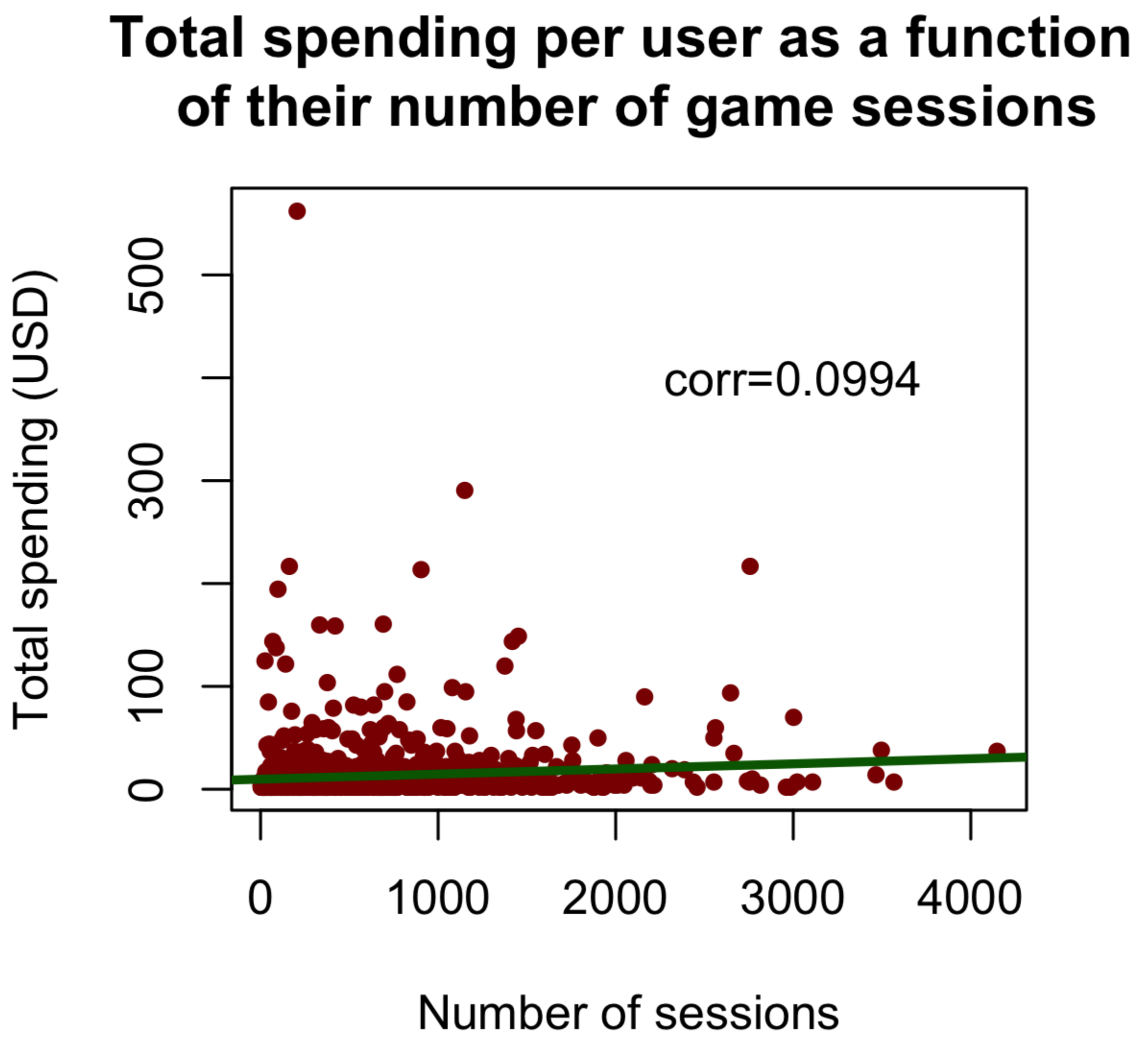

probability density function
of days when a purchase occurs
power-law
exponential
lognormal
Density
0.00
0.02
0.04
0.06
0.08
0
20
40
60
80
Days since the user profile was created

We can also look at the distribution of total spending per user:



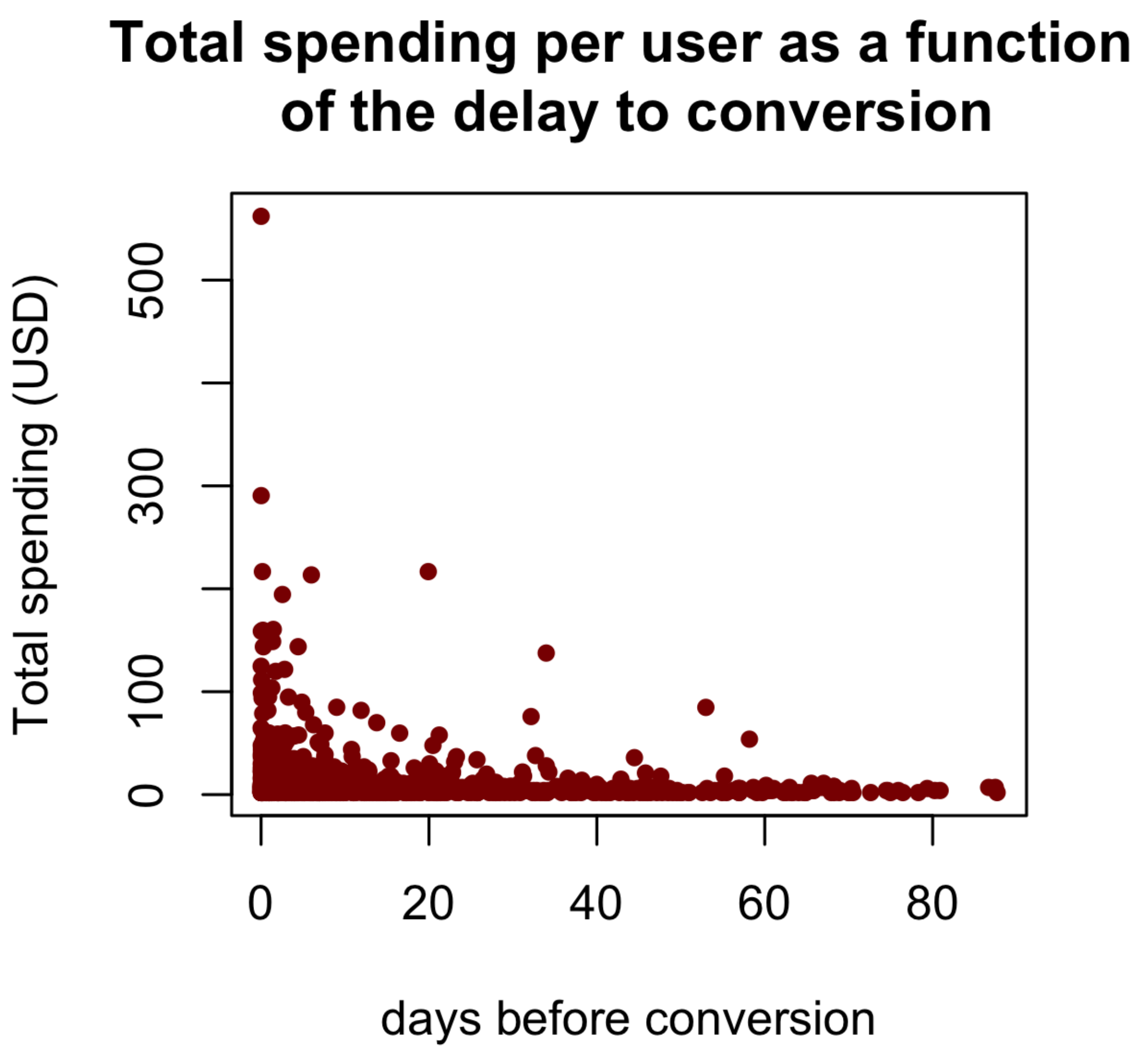
This probability density function shows that the vast majority of users only spends a small amount of money on the game, but you have a small number of users who spend significantly more (you can notice a non-zero probability of spending more than 500 USD). The company probably should do everything it can to retain such high-value players (consider in-game gifting). Still, 75% of players spend less than 10.98 USD on the game, and only 1.6% spend more than 100 USD. However, this handful of players spending more than 100 USD brings almost as much revenue to the company as the 75% of players spending less than 10.98 USD (3328 vs. 3559).

Can we find some relationships between the total spending per user and some useful predictor variable? Here we look at the total spending per user as a function of  the number of sessions this user plays. The idea behind it is that the more sessions a user plays, the more engaged he/she is with the game. Is he/she then more likely to spend more on this game? The following figure shows that this is, unfortunately, not the case.



The green line is the result of a linear regression, but as you can see the fit is not good. In fact, the Pearson correlation coefficient is very weak, at only 0.0994. Therefore, it is safe to conclude that the number of sessions a user plays the game is not a good predictor of the amount of money he/she is going to spend.

Let's study another relationship: between the total spending per user and the number of days to conversion. Conversion refers to the fact that a user who played the game for free suddenly accepts to make in-app purchases. 50% of the players convert within their first week. Still, you have 17% of the players who convert after a full month, and less than 5% who convert after 2 months! The maximum time to conversion in the database is close to 88 days. The following plot shows that the relationship between the number of days it takes a player to convert and the total amount of money he/she is going to spend on the game is, also, weak.



There does seem to be an inverse relationship: the faster a user converts, the more money he/she is going to spend on the game. However, the Pearson correlation is only -0.136 and, of course, since the relation between the two variables is visibly not linear, the Pearson coefficient is not very useful). The Spearman rank coefficient returns -0.237, which also confirms a weak negative correlation. Because the p-values for both correlations are smaller than a 5% significance level, then the null hypothesis that these correlations are actually 0 can be rejected.

Finally, the data-scientist test requires a machine learning algorithm. Here, I did a **logistic regression** using two predictor variables (again, the days to conversion and the number of game sessions per player). Since I already know that these variables are not a good predictor, the logistic regression is not expected to perform very well. Indeed, the result of:

```
summary( glm(spending ~ res$delay + res2$max_sessions,
family="binomial"))
```

is the following:

```
Call:
glm(formula = spending ~ res$delay + res2$max_sessions, family =
"binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8751  -0.3404  -0.3016  -0.2066   3.3774

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.977794   0.224739 -13.250  < 2e-16 ***
res$delay         -0.049838   0.015056  -3.310 0.000933 ***
res2$max_sessions  0.000673   0.000196   3.433 0.000596 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)


    Null deviance: 418.89  on 1099  degrees of freedom
Residual deviance: 394.78  on 1097  degrees of freedom
AIC: 400.78

Number of Fisher Scoring iterations: 7
```

For the dependent variable, which has to be categorical (0 or 1) in a logistic regression, I took a total spending larger than 50 USD (variable equal to 1) or lower than 50 USD (variable equal to 0). The p-values in this regression are very low for both predictors, confirming their uselessness at forecasting the total spending of a player.

More to the point, I could have tested other predictor variables (like the age of the player, which I believe shows potential as older players may have more disposable income than younger ones). Unfortunately, the `birth_year` field of the table `account` seems to be only filled with `NA`.

There is plenty of other information that can be extracted from this database (e.g., which days the players spend the most money? Which cities host the top spenders --- useful to decide where to advertise ---? ).