PeerJ Preprints NOT PEER-REVIEWED

How R Helps Airbnb Make the Most of Its Data

Ricardo Bion
Airbnb
Ricardo.Bion@airbnb.com
and
Robert Chang
Airbnb
Robert.Chang@airbnb.com
and
Jason Goodman
Airbnb
Jason.Goodman@airbnb.com

August 23, 2017

Abstract

At Airbnb, R has been amongst the most popular tools for doing data science in many different contexts, including generating product insights, interpreting experiments, and building predictive models. Airbnb supports R usage by creating internal R tools and by creating a community of R users. At the end of the post, the authors provide some specific advice for practitioners who wish to incorporate R into their day-to-day workflow.

Keywords: Exploratory Data Analysis, Machine Learning, Statistical Computing, Inference

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

PeerJ Preprints NOT PEER-REVIEWED

1 Introduction

Airbnb's data science team relies on R every day to make sense of our data. While many of our teammates use Python, R is the most commonly used tool for data analysis at Airbnb. In a recent survey of our whole team, we found that 73% of our Data Scientists and Analysts rated themselves as closer to "Expert" than "Beginner" in using R, and 58% regularly use R as a language for data analysis. R comes into play at all stages of the analysis pipeline, from exploratory data analysis and predictive modeling, to sharing results with business partners. In this article, we highlight the role that R plays at Airbnb and share some practical insights for others who seek to use R to help their teams make the most of their data.

2 Data Science at Airbnb

Airbnb is a community marketplace that provides access to millions of unique accommodations in more than 65,000 cities and 191 countries. In addition to accommodations, with Experiences, Airbnb offers unprecedented access to local communities and interests, while Places lets people discover recommendations by the people that live there.

A huge part of our company's success has been due to the data science team. Data science has played a crucial role at Airbnb since the company's beginning. The first member of our team was among the first 10 hires at Airbnb. Today, the team includes data scientists, data engineers, business analysts, machine learning engineers, and several product and infrastructure-focused data teams that touch almost every part of the Airbnb platform.

To start, what do we mean by data science? Put simply, our team's mandate is to use data to inform decisions. When guests and hosts interact with Airbnb, their interactions with the website and each other are represented by data. By sifting through that data for trends and patterns, we are better able to listen to the voice of our customers.

Data scientists at Airbnb are responsible for many tasks to empower the company to be more data informed - we instrument logs, build data pipelines, define metrics, develop education resources, build internal data tools, and create reports and dashboards. However, we can categorize the core of our work into three main focus areas: *Product Insights*,

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

PeerJ Preprints NOT PEER-REVIEWED

Experimentation, and *Predictive Modeling*.

2.1 Product Insights

Product insight work is exploratory in nature and usually somewhat open-ended. When we talk about "product," we are referring mainly to the Airbnb website and mobile app, which currently includes homes, places, and experiences. You do not quite know what you are going to find before you start, but the goal is to find opportunities to make the product better. Questions such as 'which types of guests are staying on Airbnb in these regions?', 'Why aren't some new hosts getting booked?', 'Which cities are supply constrained?' are all examples of product insights work. The insights gleaned from this work often lead directly to new product ideas and hypotheses about user behavior. Once we have these new product ideas, we can validate them in experiments.

2.2 Experimentation

Experimentation (also commonly referred as A/B testing) plays an important role in data-informed product development. The goal of experimentation is to validate or dispute hypotheses that the team has in order to make the user experience better. If the hypotheses are validated, a live change to the app/site will take place to all users. Almost all hypotheses and promising ideas at Airbnb are validated through controlled experiments where assignments are randomized. Before releasing new product features to the world, we verify that our new designs or data products impact key metrics such as bookings, customer service tickets, review scores, host churn, and dozens of other secondary metrics. Additionally, experimentation is a way to attribute the gains to the team that worked hard to design, test, and implement the new feature, and to ensure that a product change is not hurting bookings or other key metrics. Data scientists are involved in all aspects of the experimentation cycle. Before experiments, we perform power analyses to determine how long the experiments should be run. We draft the experimental design which includes details on how the experiment should be run and what metrics to track in order to measure success. Finally, we often perform additional "deep-dives" beyond basic t-tests to better determine the impact of the product feature in question.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

PeerJ Preprints NOT PEER-REVIEWED

2.3 Predictive Modeling

The third kind of work is what most people think of when they hear the term data science - Machine Learning and Predictive Analytics. One well-known application of Machine Learning at Airbnb is our Smart Pricing feature. If you would like some help in choosing how to price your listing, we provide you with a suggestion of a price that we think will work well for you. The suggestions are generated based on a machine learning algorithm that takes into account a variety of points of information, including the date for which you are setting the price, listing's location, amenities, your booking history, and many others. Smart Pricing lets hosts set their prices to automatically go up or down based on changes in demand for similar listings. Hosts are always responsible for setting prices, and are free to accept or reject any suggested price.

Pursuing an end-to-end machine learning project can often be costly and time consuming. This means that validating the ideas before fully investing in engineering and implementation is a high leverage activity. One of the ways in which data scientists would validate whether a modeling based solution would work is to build a prototype. R turns out to be an extremely powerful tool for this purpose because we can easily perform data wrangling and feature engineering once training data is loaded into an R session. Furthermore, with the training data prepared, data scientists can try out a wide variety of models to understand how much gain we would get versus a naive, non-modeling solution.

For example, when our data scientists try to predict revenue at the listing and guest level, we built a wide variety of prototypes in R to validate that a model based prediction is worth pursuing (by comparing the RMSE of the challenger models with the incumbent model). After the prototyping step, we work with engineers to take the prototype into production, using open-source technology like Aerosolve or Python Scikit-learn (Pedregosa et al. 2011).

3 R at Airbnb

R, as a tool, does not exist in a vacuum. It is usually the last tool used in a long series of tools for capturing, processing, and analyzing data.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

The first step is logging. Whenever users interact with Airbnb, such as by making a search, contacting a host, or even calling customer support, both client-side and server-side event data is logged. Similarly, collections of stateful information such as listing characteristics or transaction details are recorded in MySQL databases (the most common open-source relational database management system). These two main sources of data, logs and stateful information, are ingested by our data warehouse on AWS (Amazon Web Services) and securely stored on HDFS (Hadoop Distributed File System) using S3 (Amazon Simple Storage System is an object storage system with a simple web service interface to store and retrieve any amount of data from anywhere on the web).

Every night, a large number of ETL (Extract, Transform, and Load) jobs are kicked off in our open-source workflow manager Airflow (an open-source platform to programmatically author, schedule, and monitor data workflows), calculating useful metrics and aggregations from the raw logs for further analysis and predictive modeling. Output tables are organized in the star schema style so that any analyst can easily join fact tables with dimension tables to uncover insights.

Different data analysis tools are available for users of varying technical backgrounds. For visual analysis, users often leverage Tableau and our open-source data visualization tool Superset to understand the state of the business. Analysts who are comfortable with SQL use SQL Lab to interact with the data warehouse directly. Finally, data scientists use a wide variety of tools. Amongst the most important is the R language.

R is essential for all three focus areas we mentioned above. For product insight work, packages such as `dplyr` (Wickham & Francois 2016) make it easy to slice-and-dice data extremely quickly. Combined with graphing package like `ggplot2` (Wickham 2009), R allows data scientists to fly through a dataset in search of insights. For experimentation work, R's `pwr` (Champely 2017) package make it easy to perform statistical tests, helping us to hold high standards of rigor in our experimentation. For machine learning, data scientists build proof-of-concepts predictive models as prototypes to prove that more sophisticated and intelligent product features are good investments before building them for production systems. Below, we go into more detail and provide some examples.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

```r
# install and load Rbnb package
devtools::install_github("airbnb/Rbnb")

library(Rbnb)

# move data from Presto into R and run model
presto_get("select * from bookings") %>%
  lm(n_bookings ~ market + booking_channel, data=.)

# move data from R into S3
bookings %>%
  impute_data(impute_col="n_bookings", ds_col="date") %>%
  yoy("n_bookings", "date") %>%
  s3_put("bookings_yoy.csv")
```

Figure 1: Example of simple workflow in R: Querying data from Presto, performing analysis, and sending results back to S3

3.1 Product Insights

R enables product insight works at Airbnb in three main ways: Exploratory data analysis combining our internal packages with `tidyr` (Wickham 2017) and `dplyr`, data visualization with internal branded themes and packages such as `ggplot2`, and reproducible research with our internal Knowledge Repository and packages such as `rmarkdown` (Allaire et al. 2017).

3.1.1 Exploratory Data Analysis

Our data analysis workflow normally starts with data extraction. At Airbnb, we use Hive and Presto, two SQL-like languages, to extract data from HDFS to our local machines. Over the years, we have abstracted away the details of this operation so users can just type in a SQL query and get back the data in a R data.frame. Once data is in memory, we use magrittr pipes to chain operations together to perform data munging and analysis. The entire process of querying the database, manipulating data, graphing and predictive modeling building can happen in a few lines, allowing us to rapidly iterate through analyses and visualizations. We try to use a consistent style guide and set of packages for data exploration, relying mainly on `tidyr`, `dplyr`, `broom` (Robinson 2017), and `purrr` (Henry & Wickham 2017).

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

3.1.2 Data Visualization

We use ggplot2 as our main package to create ad-hoc exploratory graphics as well as polished-looking customized visualizations. When combined with tools to clean and transform data, ggplot2 allows analysts to quickly translate insights into high quality, compelling visualizations. In addition to the static graphics of `ggplot2`, we often make interactive visualizations or dashboards using R packages such as `plotly` (Sievert et al. 2017), `leaflet` (Cheng et al. 2017), `dygraphs` (Vanderkam et al. 2017), `DiagrammeR` (Sveidqvist et al. 2017), and `shiny` (Chang et al. 2017).

3.1.3 Reproducible Research

At Airbnb, all R analyses are documented in `rmarkdown`, where code and visualizations are combined within a single written report. Posts are carefully reviewed by experts in the content area and techniques used, both in terms of methodologies and code style, before publishing and sharing with the business partners. The peer review process is often repeated until both the author and the reviewers are satisfied with the analysis. This careful code review process helps bring credibility to the team's reports, reduces the number of bugs in analysis code, and often leads to more approachable insights and concrete recommendations.

Once the report is ready, the code is merged and the final report is shared in an internal website called the Knowledge Repository (Sharma & Overgoor 2016). The Knowledge Repository is a Github-enabled internal web application powered by Python and Flask (a micro web framework for building websites) that has search, subscription, and tagging capabilities. Some of these reports eventually are featured in academic journals or external blog posts - see "How Airbnb uses Machine Learning to Detect Host Preferences" (Ifrach 2015) and "How well does NPS predict rebooking?" (Qian 2015) as examples. The Knowledge Repository attempts to incorporate best practices from software engineering (e.g., code review, reproducibility, version control), with best practices from academic research (e.g., peer review, clear hypothesis, review of the literature).

Housing analysis in this format offers many advantages. It makes it easy for anyone in the company, including non-technical business partners, to find analyses that might

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

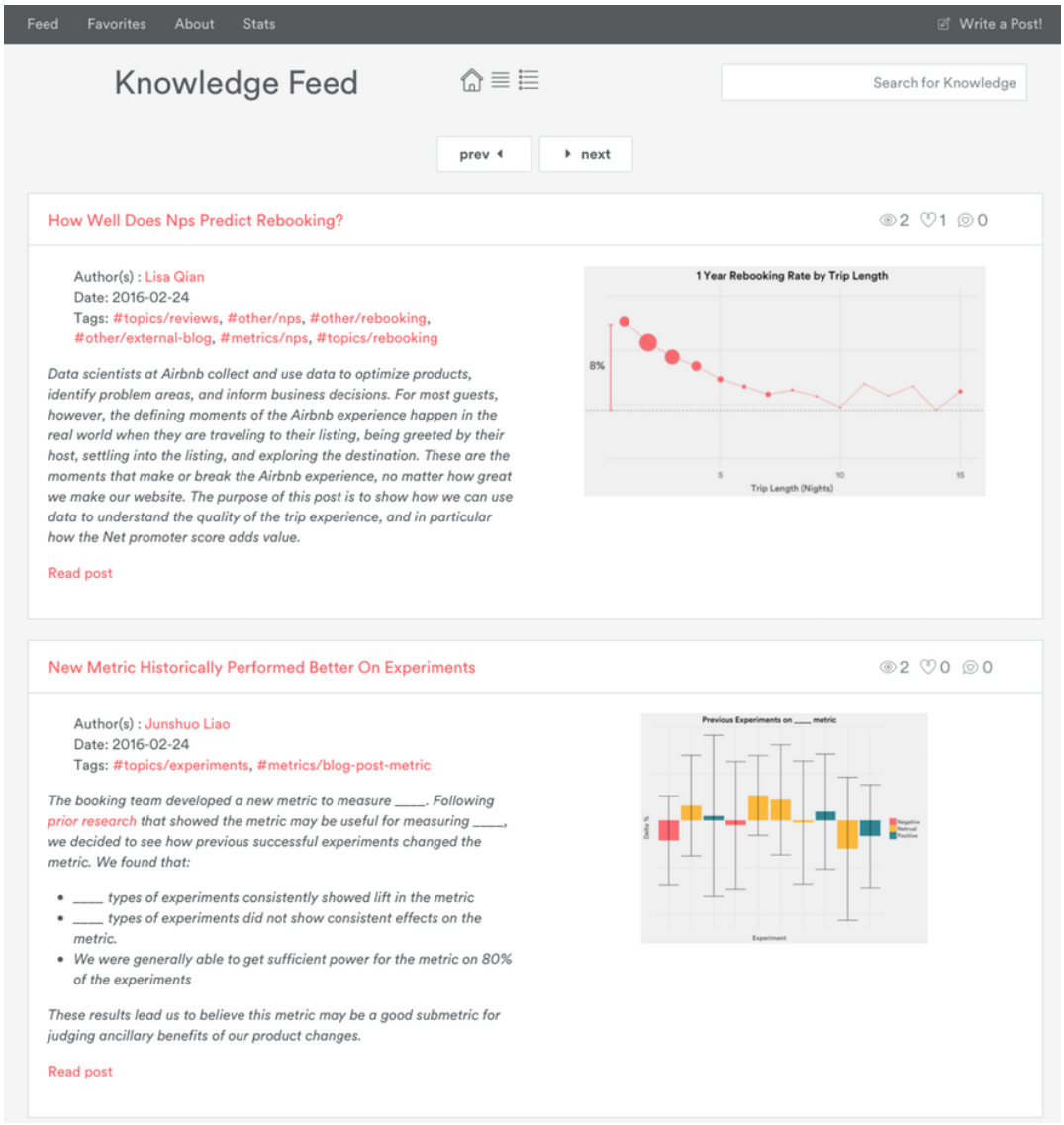

Figure 2: Example of product insights written as RMarkdown documents, rendered in our Knowledge Repository. Taken with permission from Scaling Knowledge at Airbnb. The posts include branded ggplot2 visualizations created in ggplot2.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

be relevant to their team. It allows data scientists to easily learn both context and new analytical techniques from previous work. It enables analysts to reuse code from past reports. It simplifies coordination across teams by keeping everyone in the loop on the latest research. None of this would be possible without the common usage of `rmarkdown` across the team.

3.2 Experimentation

In our team's early days, we used R to perform experiment analyses manually. We would combine experiment assignment logs with data on bookings and other important metrics before performing statistical tests to evaluate our experiments. Eventually, we built a Shiny app that did these analyses for us at scale. It reported the results of dozens of experiments to data scientists, product managers and engineers. Over time, we began running hundreds of experiments simultaneously and Shiny could no longer scale to meet our needs. We needed a fully customized solution that could be supported by our core engineering team, and easily integrated into our data infrastructure, which runs mainly in Ruby and Python. The solution was to build a production pipeline and website to analyze experiments called Experiment Reporting Framework (Moss 2014) to display experiment status. This is a great example of how we use R to prototype and demonstrate the value of having a data tool, helping us understand the needs of those who might use it, which then prompted us to build out more scalable solutions.

While today, much of our experimentation work is automated, R still plays a crucial role in more complex experiment analyses and deep dives. These analyses might, for instance, involve handling particularly complicated experimental setups that are not supported by our automated tools, target metrics that must be computed separately, or statistical analyses more sophisticated than t-tests (for more reference on experimentation at Airbnb, see Overgoor (2014)). Additionally, we use R to verify that experimental assignment is occurring properly. As with the move from our Shiny app to the production experiment tool, we try to automate our work whenever possible. If we find ourselves repeating the same ad-hoc experimental analyses in R, we try to build that functionality into our tooling. For example, we used to compute metrics that depended on the time since assignment to

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

an experiment by hand. After a few times repeating these analyses, we wrote an R function to perform the computation. Finally, this capability was incorporated into our main experimental interface.

3.3 Predictive Modeling

While R is an extremely effective tool for data cleaning and data munging, it is an even more powerful language for data modeling. Its power lies in the wide range of optimization, machine learning, and econometric techniques that the R community has built over the years. At Airbnb, many of our data scientists build models using these packages. Below, we describe three of our teammates' projects that used R's data predictive modeling capabilities.

3.3.1 Airbnb Guest Service Fee Optimization using NLopt package

When a reservation is confirmed, Airbnb charge guests a service fee between 5% and 15% of the reservation subtotal. These fees are calculated using a variety of factors, such as length of the reservation, and characteristics of the listing. One of the projects of the data science team is to optimize these fees, while keeping pricing policies that are fair to our guests and hosts. For example, Airbnb tends to charge lower guest service fee percentages to higher reservation subtotals, but we do not have surge service fees for high-demand dates such as New Years or holidays. In the past, we experimented with different service fee structures, which allowed us to learn the demand elasticity of guest bookings to service fees. Given this elasticity, we could then find the fee curve that maximized revenue on the platform. This exercise was entirely modeled as a nonlinear optimization problem using `NLopt` (Johnson 2017). The result of the model provided guidance on how we should set the guest fees. We ran a follow-up experiment to validate this new payout structure.

3.3.2 Building A Marginal Returns Model in R

Where should we invest to attract new hosts to Airbnb? How much should we spend on these efforts? To better answer these questions, our data scientists built a Cobb Douglas

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

like econometric matching model to estimate bookings by market as a function of supply and demand in that market.

$$B(S, D) = AD^{\alpha}S^{\gamma}$$

Here, ***B*** represents nights booked in a particular, ***D*** and ***S*** represent demand and supply respectively. Furthermore, ***A*** stands for matching efficiency (i.e. how well the marketplace is matching supply and demand), the exponents ***alpha*** and ***gamma*** can be interpreted as the booking elasticity of supply and demand - for a one percent increase in demand/supply, alpha and gamma represent the percent increase in bookings.

Using historical supply (nights available to book) and demand (number of searchers), we were able to estimate the elasticities in R by taking the log of the equation on both side. With the fitted model and future forecasts for supply and demand, we are able to predict the marginal returns in bookings of adding an additional supply to the market (by taking the derivative of B with respect to S). Finally, the marginal returns of supply can help us to determine which market has the highest marginal returns, and we can prioritize our acquisition efforts accordingly.

3.3.3 Causal Tree

Often, when experiments are run, data scientists often are asked questions such as which subset of users is most affected by the new feature. Many times, data scientists need to manually go through the different cohort cuts in order to identify the group that has the highest lift.

More recently, we started experimenting with Susan Athey's `CausalTree` (Athey & Imbens 2015) method, which applies Machine Learning technique to Causal Inference problems. The premise of this package is that the algorithm will recursively partition the experiment data based on splits that give us the highest lift between control and treatment.

The algorithm, along with many other new techniques, is often first implemented in R. We were able to leverage the open-sourced version of this algorithm and immediately integrate it with our offline analysis. This is a case where we did not build the model

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

ourselves, but we are able to leverage on the work of other statisticians and academics to carry out novel analyses.

4 How We Support R Usage

4.1 Community

We help our team members develop as R programmers in a variety of ways. The most structured way is through teaching classes on R. Every new data scientist at Airbnb onboards with a week of classes in a Data Bootcamp. The R Bootcamp classes are taught by experienced Airbnb data scientists who use R as their tool of choice. In the class, they help new team members set up their development environments and walk them through a series of hands-on tutorials demonstrating commonly-used R package with real Airbnb data.

Once on the job, team members mostly improve their skills by learning from one another. As detailed above, analyses are peer-reviewed by fellow data scientists before they are shared with external business partners. The process allows team members to give feedback not only on content and analysis but on the R coding itself. One of the best ways to improve is to read others' code in the Knowledge Repository to see how they solved problems that resemble your own. Another way we learn from one another is through our R Slack channel (an enterprise instant message communication platform), where team members can ask one another for help or advice on using R.

Internally, we actively run learning lunches, learning groups, and tutorials to showcase new packages and development in R. We allocate education funds and actively encourage our data scientists to learn from the best in the industry by attending R training courses such as DataCamp (a popular website for learning data science in the browser) and conferences such as UseR!.

We recently also developed three R classes for our internal Data University, which is open to all Airbnb employees. These classes cover introduction to analyses in R, (data visualization, and how to use and contribute to our internal R packages. They are taught in person by expert developers, and include custom-made online resources.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

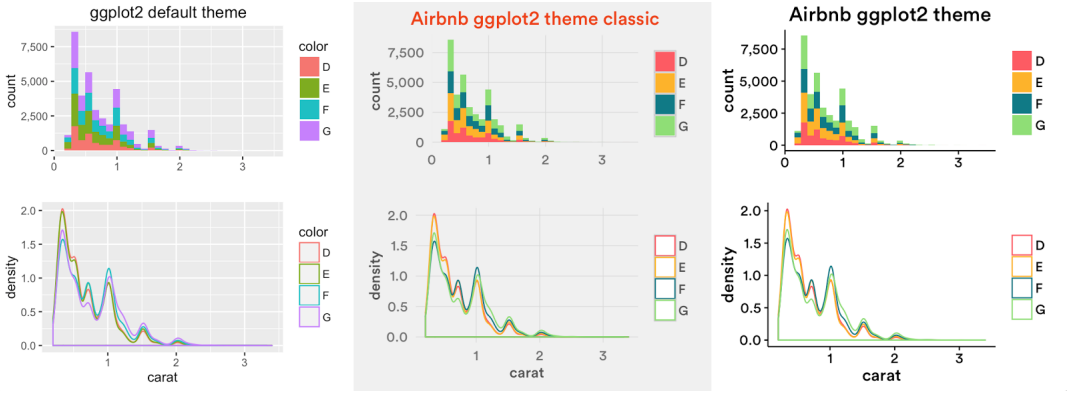

Figure 3: Example of Airbnb branded ggplot2 themes and scales to standardize our data visualizations. These extensions create a consistent internal data science brand, and can be found on Github. Taken with permission from 'Using R packages and education to scale data science at Airbnb.'

Externally, we also engage with the broader R community outside Airbnb. We participate in and sponsor rOpenSci, contribute to open source projects such as `ggtech` (Bion 2016*a*) and `RPresto` (Filiz & Goder 2017), and give talks at conferences such as the Shiny Developer Conference and UseR! We have been fortunate to have influential R developers visit our headquarters in San Francisco such as Hadley Wickham and Ramnath Vaidyanathan.

4.2 Tooling

In order to aid R analyses across the team, we developed an internal R package, `Rbnb` (Bion 2016*b*), full of useful functions and templates for data analysis specifically at Airbnb. The package is hosted on our internal Github instance, so that anyone on the team can see its contents and contribute. Updates are constantly being pushed out using the `devtools` (Wickham & Chang 2017) package.

Rbnb's primary purposes are twofold. First, it allows easy access to our infrastructure from R as we mentioned before. Without Rbnb, this process would be extremely time-consuming and inefficient. Second, it provides an Airbnb-styled theme to ggplot2, ensuring that our plots have a consistent look and feel across the team.

While these two aspects are by far the most commonly used features of the package,

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

it contains over 90 functions contributed by more than 20 different members of the team. These functions perform a wide variety of operations, from statistical procedures like imputing data to producing customizable graphics. The concept is that anytime anyone on the team solves a problem that they think others might encounter, they can generalize their code, include it into Rbnb and now the whole team can access it.

We recently launched a series of classes to teach any data scientist that is interested in how to develop for R packages. More than half of the team that uses R has participated, and Rbnb contributions are on the rise. We are also establishing a mailing list to distribute news of new contributions to the package to the team and give credit to those who create code that the whole team can use.

For its first few years, Rbnb had relatively few contributors. The team was small, and thus the returns to writing generalizable code were low. Now that the team has grown much larger, scaling our work is more valuable. However, in reality, no team or person is solely responsible for the package's maintenance, as it is a collective effort. Therefore, building the right incentives and rewards for others to continue contributing is a crucial and challenging task.

Recently, we have begun exploring new ways to incentivize contributions, for instance, emphasizing the role of contributing beyond one's product team in performance reviews and by developing special stickers that team members can display on their laptops for taking classes on how to contribute to Rbnb, and for actually contributing code. Ideally, team members can strike a middle-ground, where they spend a small amount of time learning how to contribute to the package, and then integrate contributing code the whole team can use as part of their normal workflow if they happen to solve a problem that can be easily generalized.

For a more extensive discussion on tool building and education at Airbnb, see our external post on this topic: Using R packages and Education to Scale Data Science at Airbnb.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

PeerJ Preprints NOT PEER-REVIEWED

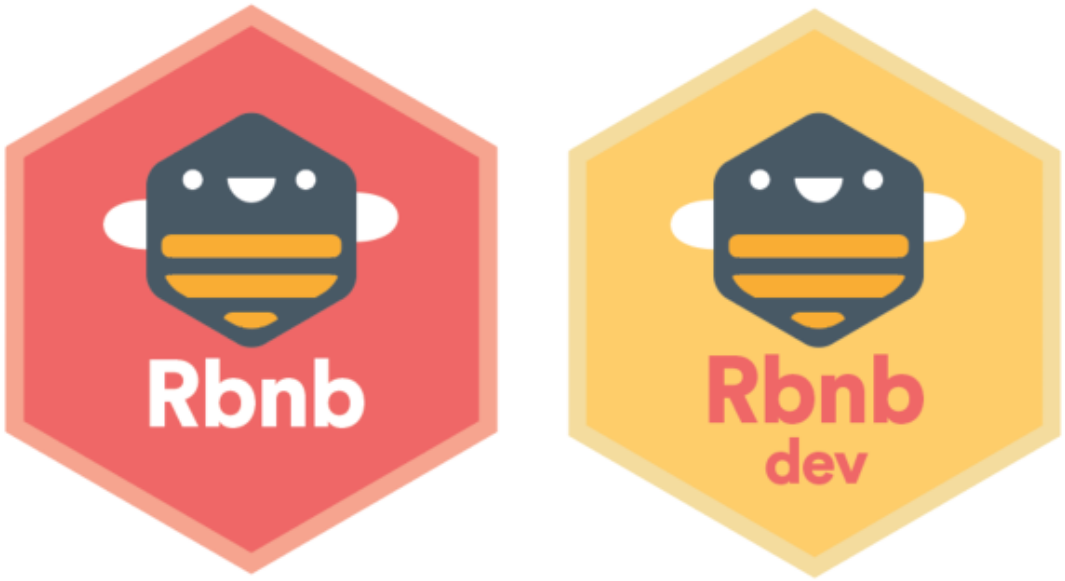

Figure 4: Example of stickers used to incentivize contribution to Rbnb, our internal R package. Initiatives like this help build a community of R developers at Airbnb.

5 Advice For Practitioners

We have come a long way as a team, and we have learned a lot from this journey. Here are a few recommendations to practitioners who are interested in using R to help their team make the most of its data: First, we recommend building helper functions to connect R to infrastructure containing data, such as production databases, as early as possible. In the early days, R is great for prototyping new data tools when building production-level data tools is costly or required evidence of usage. Once you have a sufficiently large team, create an internal R package to share code between analysts and encourage a culture of contribution and tool-building can enable the whole team for higher productivity. In that internal package, it's useful to develop a standardized set of visualization themes to reinforce the brand of the data team. We suggest using Github, `rmarkdown`, and our newly open-sourced Knowledge Repository for peer-reviewing analysis and for making research reproducible. Consider investing in ongoing education and training for team members. Finally, remember that R is not the only language for data work, and that there is a great benefit in fostering a diverse team.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

R is an essential part of the data scientist's toolset at Airbnb. We are proud to support the language, and we would encourage any company that wants to make the most of its data to consider adopting R as its language of choice.

References

Allaire, J., Cheng, J., Xie, Y., McPherson, J., Chang, W., Allen, J., Wickham, H., Atkins, A., Hyndman, R. & Arslan, R. (2017), *rmarkdown: Dynamic Documents for R*. R package version 1.6.
URL: *https://CRAN.R-project.org/package=rmarkdown*

Athey, S. & Imbens, G. (2015), 'Recursive partitioning for heterogeneous causal effects'.

Bion, R. (2016*a*), *ggtech: ggplot2 tech themes and scales*. R package version 0.1.

Bion, R. (2016*b*), *Using R packages and education to scale Data Science at Airbnb*, Airbnb.
URL: *Using R packages and education to scale Data Science at Airbnb*

Champely, S. (2017), *pwr: Basic Functions for Power Analysis*. R package version 1.2-1.
URL: *https://CRAN.R-project.org/package=pwr*

Chang, W., Cheng, J., Allaire, J., Xie, Y. & McPherson, J. (2017), *shiny: Web Application Framework for R*. R package version 1.0.3.
URL: *https://CRAN.R-project.org/package=shiny*

Cheng, J., Karambelkar, B. & Xie, Y. (2017), *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 1.1.0.
URL: *https://CRAN.R-project.org/package=leaflet*

Filiz, O. I. & Goder, S. (2017), *RPresto: DBI Connector to Presto*. R package version 1.2.1.9000.
URL: *https://github.com/prestodb/RPresto*

Henry, L. & Wickham, H. (2017), *purrr: Functional Programming Tools*. R package version 0.2.2.2.
URL: *https://CRAN.R-project.org/package=purrr*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

PeerJ Preprints NOT PEER-REVIEWED

Ifrach, B. (2015), *How Airbnb uses Machine Learning to Detect Host Preferences*, Airbnb.
URL: *https://medium.com/airbnb-engineering/how-airbnb-uses-machine-learning-to-detect-host-preferences-18ce07150fa3*

Johnson, S. G. (2017), *The NLopt nonlinear-optimization package*.
URL: *http://ab-initio.mit.edu/nlopt*

Moss, W. (2014), *Experiment Reporting Framework*, Airbnb.
URL: *https://medium.com/airbnb-engineering/experiment-reporting-framework-4e3fcd29e6c0*

Overgoor, J. (2014), *Experiments at Airbnb*, Airbnb.
URL: *https://medium.com/airbnb-engineering/experiments-at-airbnb-e2db3abf39e7*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. (2011), 'Scikit-learn: Machine learning in python', *The Journal of Machine Learning Research* **12**, 2825–2830.

Qian, L. (2015), *How well does NPS predict rebooking?*, Airbnb.
URL: *https://medium.com/airbnb-engineering/how-well-does-nps-predict-rebooking-9c84641a79a7*

Robinson, D. (2017), *broom: Convert Statistical Analysis Objects into Tidy Data Frames*. R package version 0.4.2.
URL: *https://CRAN.R-project.org/package=broom*

Sharma, C. & Overgoor, J. (2016), *Scaling Knowledge at Airbnb*, Airbnb.
URL: *https://medium.com/airbnb-engineering/scaling-knowledge-at-airbnb-875d73eff091*

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M. & Despouy, P. (2017), *plotly: Create Interactive Web Graphics via 'plotly.js'*. R package version 4.7.0.
URL: *https://CRAN.R-project.org/package=plotly*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017

PeerJ Preprints NOT PEER-REVIEWED

Sveidqvist, K., Bostock, M., Pettitt, C., Daines, M., Kashcha, A. & Iannone, R. (2017), *DiagrammeR: Create Graph Diagrams and Flowcharts Using R*. R package version 0.9.0.
URL: *https://CRAN.R-project.org/package=DiagrammeR*

Vanderkam, D., Allaire, J., Owen, J., Gromer, D., Shevtsov, P. & Thieurmel, B. (2017), *dygraphs: Interface to 'Dygraphs' Interactive Time Series Charting Library*. R package version 1.1.1.4.
URL: *https://CRAN.R-project.org/package=dygraphs*

Wickham, H. (2009), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York.
URL: *http://ggplot2.org*

Wickham, H. (2017), *tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions*. R package version 0.6.3.
URL: *https://CRAN.R-project.org/package=tidyr*

Wickham, H. & Chang, W. (2017), *devtools: Tools to Make Developing R Packages Easier*. R package version 1.13.2.
URL: *https://CRAN.R-project.org/package=devtools*

Wickham, H. & Francois, R. (2016), *dplyr: A Grammar of Data Manipulation*. R package version 0.5.0.
URL: *https://CRAN.R-project.org/package=dplyr*

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.3182v1 | CC BY 4.0 Open Access | rec: 24 Aug 2017, publ: 24 Aug 2017