Using R packages and education to scale Data Science at Airbnb

AirbnbEng · Follow
Published in The Airbnb Tech Blog
9 min read · Mar 29, 2016

By Ricardo Bion

One of my favorite things about being a data scientist at Airbnb is collaborating with a diverse team to solve important real-world problems. We are diverse not only in terms of gender, but also in educational backgrounds and work experiences. Our team includes graduates from Mathematics and Statistics programs, PhDs in fields from Education to Computational Genomics, veterans of the tech and finance worlds, as well as former professional poker players and military veterans. This diversity of training and experience is a tremendous asset to our team's ability to think creatively and to understand our users, but it presents challenges to collaboration and knowledge sharing. New team members

arrive at Airbnb proficient in different programming languages, including R, Python, Matlab, Stata, SAS, and SPSS. To scale collaboration and unify our data science brand, we rely on tooling, education, and infrastructure. In this post, we focus on the lessons we have learned building R tools and teaching R at Airbnb. Most of these lessons also generalize to Python.

Our approach has two main pillars: package building and education. We build packages to develop collaborative solutions to common problems, to standardize the visual presentation of our work, and to avoid reinventing the wheel. The goals of our educational efforts are to give all data scientists exposure to R and to the specific packages we use, and to provide opportunities for further learning to those who wish to deepen their skills.

R packages

In small data science teams, individual contributors often write single functions, scripts, or templates to optimize their workflows. As the team grows, different people develop their own tools to solve similar problems. This leads to three main challenges: (i) duplication of work within the team, both in writing the tools and reviewing code, (ii) lack of transparency about how tools are written and lack of documentation, often resulting in bugs or incorrect usage, (iii) difficulty sharing new developments with other users, slowing down productivity.

R packages shared through Github Enterprise address these three challenges, which makes them a great solution for our needs. Specifically, (i) multiple people can collaborate simultaneously in order to improve the tools and fix bugs, (ii) contributions are peer reviewed, and (iii) new versions can be deployed to all users as needed. Packages are the basic units of reproducible R code. They can include functions, documentation, data, tests, add-ins, vignettes, and R markdown templates. I started working on our first internal R package, called Rbnb, nearly two years ago. It was initially launched with only a couple of functions. The package now includes more than 60 functions, has several active developers, and is actively used by members of our Engineering, Data Science, Analytics, and User Experience teams. As of today, our internal Knowledge Repo has nearly 500 R Markdown research reports using the Rbnb package.

The package is developed in an internal Github Enterprise repository. There, users can submit issues and suggest enhancements. As new code is submitted in

a branch, it is peer reviewed by our Rbnb developers group. Once the changes are approved and documented, they are merged into the master codebase as a new version of the package. Team members can then install the newest release of Rbnb directly from Github using devtools. We are currently working on adding lintr checking for both style and syntax, and test coverage with testthat.

The package has four main components: (i) a consistent API to move data between different places in our data infrastructure, (ii) branded visualization themes, scales, and geoms for ggplot2, (iii) R Markdown templates for different types of reports, and (iv) custom functions to optimize different parts of our workflow.

The most used functions in Rbnb allow us to move aggregated or filtered data from a Hadoop or SQL environment into R, where visualization and in-memory analysis can happen more naturally. Before Rbnb, getting data from Presto into R in order to run a model required multiple steps. Data scientists would have to authenticate with their cluster credentials, open an SSH tunnel, enter host, port, schema, and catalog information for Presto, download a csv file, load that file into R, and only then run the desired models. Now, all of this can be done by piping two functions, as Rbnb takes care of all of the implementation details under the hood, while working with other well-maintained packages like RPresto. Similarly, getting data from R and moving it to Amazon S3 can be done with only one line of code. Data scientists no longer have to save a csv file from R, set up multi-factor authentication with our API keys, configure AWS, and run a bash command to move the csv into remote storage. More importantly, all functions follow a similar specification (i.e., ***place_action(origin, destination)***).

If our data infrastructure changes — for instance, if a cluster moves or our Amazon S3 authentication details change — we can change our implementation of Rbnb without changing our functions' interface.

```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. Sample code showing a few functions from Rbnb, our internal R package. All functions follow a similar specification, go through comprehensive code review, and substantially improve our workflow by abstracting away common tasks.

The package has also helped us brand our work across Airbnb through the use of consistent styles for data visualizations — see these posts by Bar Ifrach and Lisa Qian for examples. We have built custom themes, scales, and geoms for ggplot2, CSS templates for htmlwidgets and Shiny, and custom R Markdown templates for different types of reports. These features override R defaults with fonts and colors that are consistent with the Airbnb brand.

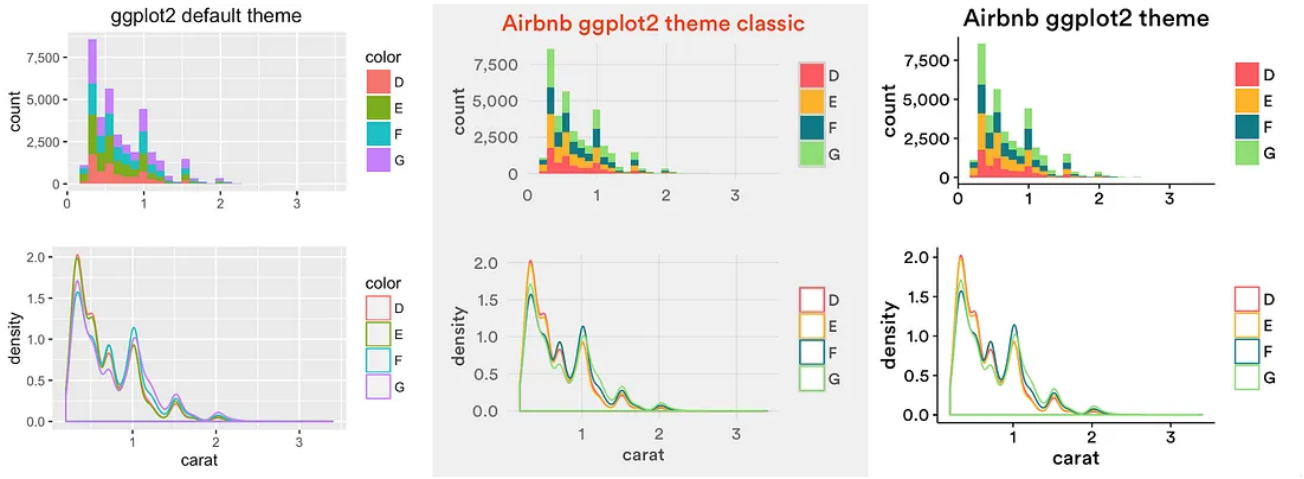


Figure 2. Airbnb branded themes and scales for ggplot2. These extensions create a consistent internal data science brand, and can be found on Github.

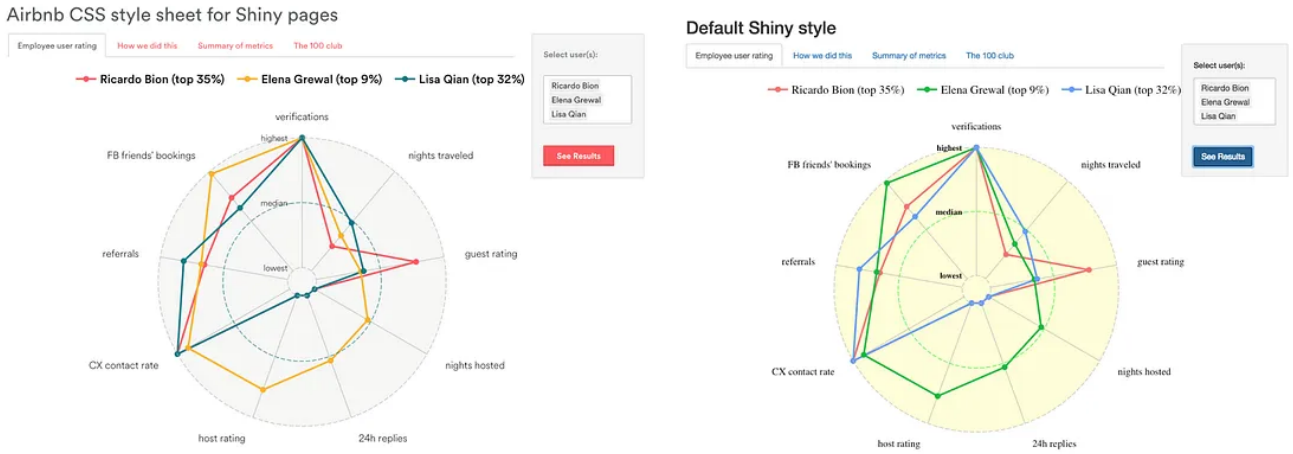


Figure 3. Airbnb branded CSS style sheet for Shiny exploring employee Airbnb usage. These style sheets give our internal tools an aesthetic that is consistent with the Airbnb.com website, making them more familiar and engaging to users.

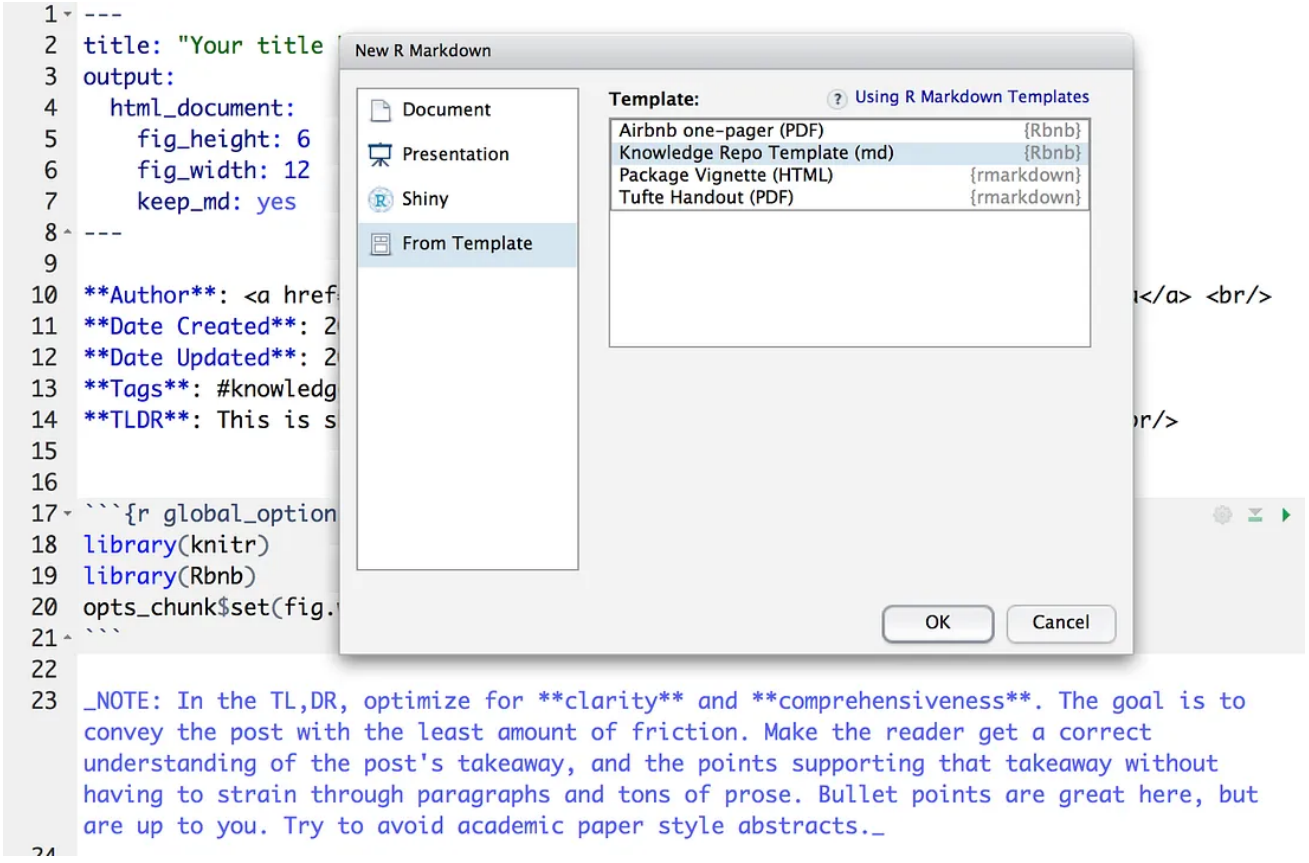


Figure 4. Airbnb branded R Markdown template. These templates include the main requirements of different types of reports.

The Rbnb package also has dozens of functions that we have created to automate common tasks such as imputing missing values, computing year-over-year trends, performing common data aggregations, and repeating patterns that we use to analyze our experiments. Adding a new function to the package might take some

time, but this initial investment pays off in the long run. By using the same R package we develop a common language, visualization style, and foundation of peer-reviewed code as our building blocks.

Education

It does not matter how many tools you build if people do not know how to use them. After a period of rapid growth, we started organizing monthly week-long data bootcamps for new hires and current team members. They include 3-hour R workshops, and optional mentorship in a bootcamp project coded in R and written in R Markdown.

The bootcamp R class focuses on the Rbnb package and on common R packages used to reshape and manipulate data frames (tidyr and dplyr), visualize data (ggplot2), and write dynamic reports (R Markdown). We give participants study guides and materials a few days before our class. During class, we walk through a structured tutorial using our own data, including challenges that we commonly face on the job as working examples.

This approach allows users who are not familiar with R to start coding within a few hours, without having to worry about the intricacies of more advanced programing. We also introduce users to our internal style guide and to many useful R packages, such as formattable, diagrammeR, and broom. Finally, we give them directions on how to find help and online resources.

After the bootcamp, we encourage users to continue learning. We sponsor individual memberships to DataCamp and help team members organize study groups around self-paced and interactive online courses. We also pair new hires with experienced peers who serve as mentors. These mentors walk new team members through their first contributions as data scientists. We have an internal Slack channel in which users can pose any questions related to R, and organize regular office hours in which experienced developers can help with more complex coding challenges. Our team members organize learning lunches and classes on topics such as SparkR, R object systems, and package development. Most recently, four team members attended a Master R Developer Workshop organized by RStudio, and shared what they learned with the team afterwards.

Members of our Data Science team are also encouraged to contribute code to Rbnb. The process of going through a comprehensive code review allows users to

develop new skills that are valuable to future projects. In addition, they feel ownership of an important internal tool and see how their contributions can benefit their peers' work. We guide new contributors on best practices, function documentation, testing, and style.

We also engage with the broader R community outside Airbnb. We sponsor conferences like the upcoming rOpenSci Unconf, contribute to open source projects (e.g., ggtech, ggradar), and give talks at meetings such as the Shiny Developer Conference and UseR Conference. We have been fortunate to have influential R developers visit our headquarters in San Francisco last year, including Hadley Wickham and Ramnath Vaidyanathan.

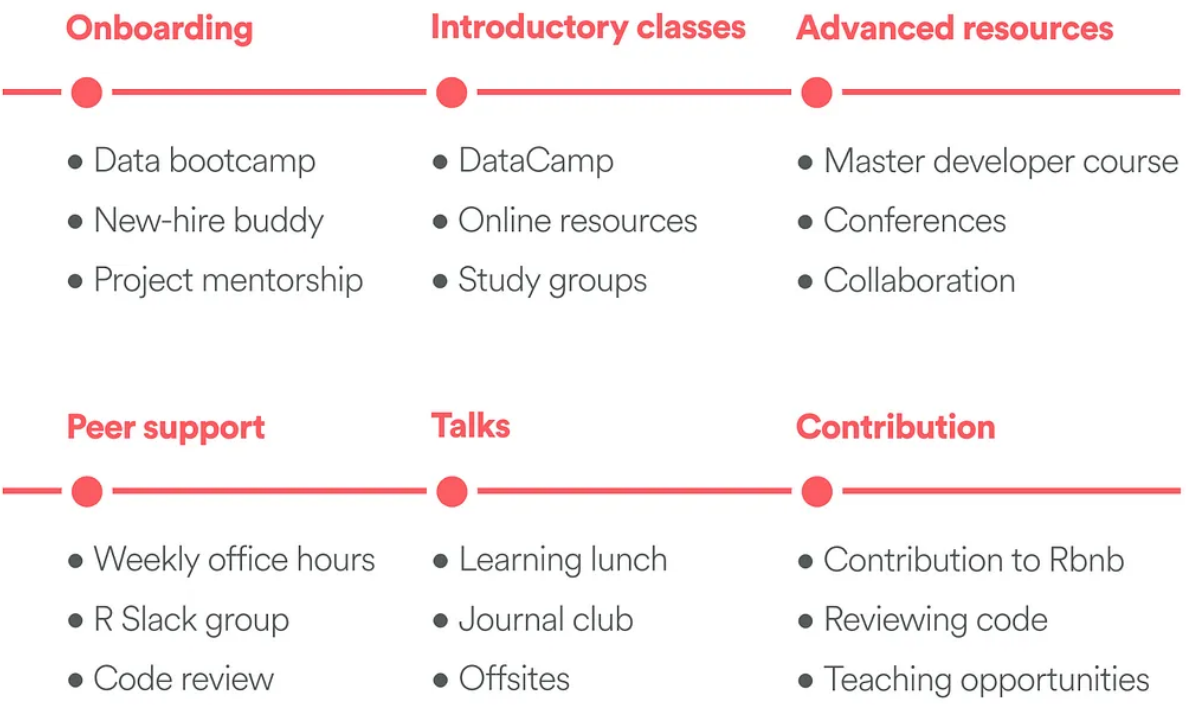


Figure 5. R education at Airbnb. The tools we develop become more impactful with structured learning resources.

Infrastructure

In addition to tools and education, we also invest in strong data infrastructure. Our Shiny apps have had nearly 100k page views since our server was first started three years ago. We recently started supporting a new RStudio Server and SparkR cluster. We have a single Chef recipe with R packages and version control across all of the machines in our clusters, allowing for rapid updates and large-scale deployment.

Summary

Powerful R tools, continuous education, engagement with the R community, and strong data infrastructure have helped our Data Science team scale. Since we started this initiative nearly two years ago, we have watched team members who had never before opened R transform into strong R developers who now teach R to our new hires. The foundation we have built allows us to hire a wide range of data scientists, sharing a growth mindset and excitement to learn new skills. This approach has helped us build a diverse team that brings new insights and perspectives to our work.

The creation of the Rbnb R package has inspired our Python developers to release an internal Python package for data scientists, called Airpy. Our developers collaborate so that the packages have a similar interface and set of functions. We encourage team members to contribute code to both Rbnb and Airpy, and we work together to develop more effective education resources and tools to empower our team. Today, many members of our team are proficient in both R and Python, and are able to review and write reliable code in both languages. In a recent survey with 66 members of our team, we found that 80% of our data scientists and analysts rated themselves as closer to “Expert” than “Beginner” in using R for data analysis, even though only 64% of them use R as their primary data analysis language. Similarly, 47% of the team members rated themselves as closer to “Expert” than “Beginner” in using Python for data analysis, though only 31% use it as their primary data analysis tool. The remaining 5% said they used both languages around equally. We focus on building a balanced team with strong developers using both languages, and have no preference or bias for either in our hiring process. This is yet another way through which diversity of skills, experiences, and backgrounds, have helped increase the impact of our team.

Thanks to Jenny Bryan, Mine Cetinkaya-Rundel, Scott Chamberlain, Garrett Grolemund, Amelia McNamara, Hilary Parker, Karthik Ram, Hadley Wickham, and to the Airbnb Engineering and Data Science teams for comments on an earlier version of this post.

Check out all of our open source projects over at airbnb.io and follow us on Twitter — @AirbnbEng + @AirbnbData

Data Science | R Programming

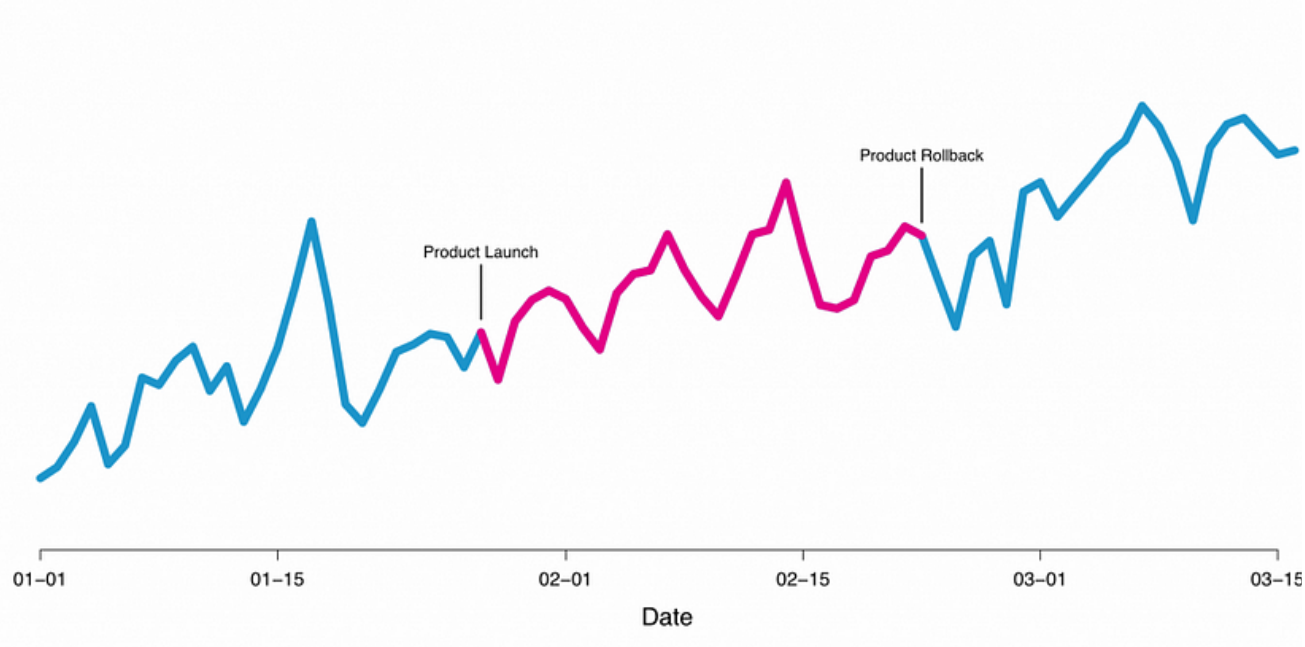
Follow

Written by AirbnbEng

62K Followers · Editor for The Airbnb Tech Blog

Creative engineers and data scientists building a world where you can belong anywhere. http://airbnb.io

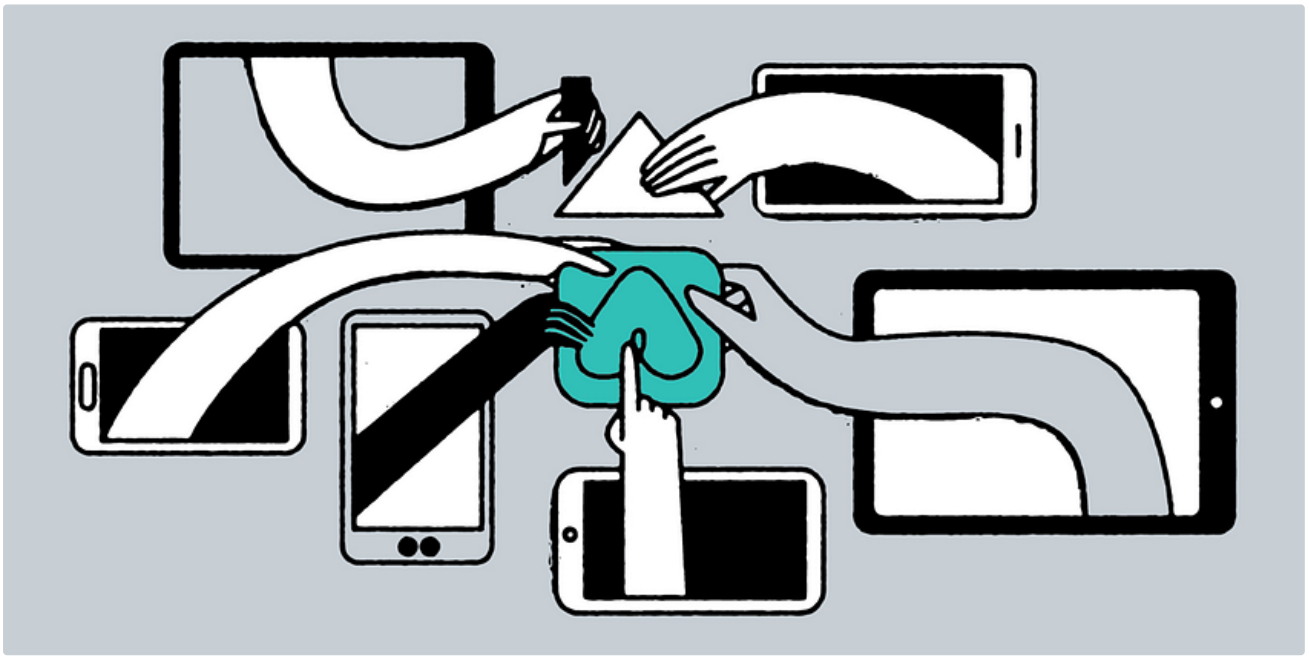
More from AirbnbEng and The Airbnb Tech Blog



AirbnbEng in The Airbnb Tech Blog

Experiments at Airbnb

May 28, 2014 3.8K 15



Ryan Brooks in The Airbnb Tech Blog

A Deep Dive into Airbnb's Server-Driven UI System

How Airbnb ships features faster across web, iOS, and Android using a server-driven UI system named Ghost Platform .

Jun 30, 2021 4.1K 40

Steven Bassett in The Airbnb Tech Blog

Rethinking Text Resizing on Web

Airbnb has made significant strides in improving web accessibility for Hosts and guests who require larger text sizes.

May 17 619 7

AirbnbEng in The Airbnb Tech Blog

Data Infrastructure at Airbnb

By James Mayfield, Krishna Puttaswamy, Swaroop Jagadish, and Kevin Long

Feb 23, 2016 2.3K 16

See all from AirbnbEng

See all from The Airbnb Tech Blog

Recommended from Medium

Thijs Sluijter in Picnic Engineering

Generating your shopping list with AI: recommendations at Picnic

How we find the best items for our customers.

Jan 9 331 3

Maham Haroon in Towards Data Science

Exploring Counterfactual Insights: From Correlation to Causation in Data Analysis

Use of counterfactuals for informed decision-making in data science

Oct 7, 2023 244 1

Lists

Predictive Modeling w/ Python

20 stories · 1357 saves

Practical Guides to Machine Learning

10 stories · 1635 saves

Coding & Development

11 stories · 691 saves

ChatGPT prompts

48 stories · 1765 saves

Netflix Technology Blog in Netflix TechBlog

How Product Teams Can Build Empathy Through Experimentation

A conversation between Travis Brooks, Netflix Product Manager for Experimentation Platform, and George Khachatryan, OfferFit CEO

Oct 13, 2022 361 1

Bex T. in The Startup

I Beat Procrastination With $40 For Good

With a system that charges real money for skipping a habit

3d ago 384 3

William Christiansen, Ph.D.

Dario Radečić in Appsilon

Introduction to R Shiny Reactivity with Hands-on Examples

Reactivity empowers the interactivity and responsivness of Shiny applications

Oct 13, 2023 153 1