

The Effect of Consumer Search Costs on Entry and Quality in the Mobile App Market

Daniel Ershov*
University of Toronto
daniel.ershov@mail.utoronto.ca

September 29, 2017

Click here for current version

Abstract

This paper examines the effects of consumer search costs on entry, product design, and quality in online markets. Using 2012-2014 data from the Google Play mobile app store, I take advantage of a natural experiment that reduced search costs for one product type (game apps) in early 2014. Difference-in-differences estimates show that entry increased by 33% relative to the control group (non-games), and that most additional entry was by “niche” products. These estimates also show that lower search costs reduced the quality of new entrants. To separate out the different welfare effects of this change, I develop and estimate a structural model of demand and supply. I show that there are large welfare gains from reduced marginal search costs, smaller gains from increased product variety, and very small losses from lower product quality.

*I would like to thank Victor Aguirregabiria, Avi Goldfarb, and Heski Bar-Isaac for their time spent reading and discussing this paper and for their helpful comments. This paper also substantially benefitted from discussions with Joel Waldfogel, Ben Shiller, Philipp Schmidt-Dengler, Jason Walter, Eduardo Souza Rodrigues, and seminar participants at the University of Toronto, TSE, Tilburg, Sciences Po, KU Leuven, the 2016 EARIE and JEI conferences, the 2017 IIOC, the 8th Annual Workshop on Consumer Search and Switching Costs, and the 8th Annual Conference on Internet Commerce and Innovation. All errors are my own.

1 Introduction

The development of the commercial internet at the beginning of the 1990s was the biggest change in consumer search costs in the last fifty years. Indeed, there was an expectation that consumer search costs were going to disappear.¹ This did not happen. Recent evidence suggests that internet search is still costly to consumers.²

A potential reason for the persistence of search costs is the proliferation of new product varieties, with consumers being able to look through thousands, or hundreds of thousands, of products.³ The difficulty of product discovery or “discoverability” online is a major concern for consumers,⁴ firms,⁵ and regulators.⁶ Proliferation could be especially important in some of the new online markets where firms cannot adjust their prices (e.g., iTunes, Soundcloud) and where consumers cannot use prices as a proxy for quality. Most of the existing literature focuses on the pricing implications of search costs, rather than non-price effects such as entry or product quality.

This paper examines how consumer search costs in online markets affect market structure, product variety, quality, and consumer welfare. I empirically study these effects using new 2012-2014 data from the Google Play mobile app store, a large online market where most products are free to download. App stores illustrate the main characteristics that distinguish online markets.⁷ In particular, app stores have a large number of products. Thousands of new apps appear every week, and it is costly for consumers to search for new products.

According to industry surveys (Nielsen 2011, Ipsos 2013), consumers’ primary search method in app stores is to browse through category areas (e.g., “Productivity Apps”). These category areas include “bestseller” lists, as well as other “featured” apps (see Section 2.1). I take advantage of a natural experiment: a re-categorization of part of the Google Play store. In December 2013, Google Play announced a split of mobile game categories, which grew from 6 to 18.⁸ The split took place on March 2014. Industry observers believe that the split in the categories reduced consumer search costs.⁹ Before the change, consumers browsing through the categories would

¹Levin (2011)

²e.g., Ghose, Goldfarb, and Han (2012), Athey and Imbens (2015)

³Sorensen (2007), Hendricks and Sorensen (2009)

⁴<https://www.imore.com/cant-find-app-store-app-search-different>

⁵<http://www.imediconnection.com/article/144344/why-consumers-cant-find-you-online>

⁶<https://www.theverge.com/2017/6/27/15872354/google-eu-fine-antitrust-shopping>

⁷e.g., Ellison and Ellison (2005), Levin (2011).

⁸See Appendix A for a full list of categories before and after the split.

⁹<http://androidcommunity.com/google-play-store-sports-18-new-categories-for-games-20140317/>

see different app types together (e.g., Family Games and Action Games). Consequently, consumers looking for an app type would not necessarily find it easily.¹⁰

The re-categorization of the store was explicitly designed by Google to reduce consumer search costs rather than to induce firm entry.¹¹ Google’s split game categories matched iTunes’ existing game categories, suggesting that category selection was not driven by pre-existing trends. The alternative view, that something particular to competition or entry in games caused Google to change them, does not hold in the data.¹²

My data consists of weekly and monthly snapshots of the US Google Play store from January 2012 to December 2014. This data includes the observable characteristics of all apps available in the market. Since non-game categories were not split, I use difference in differences (DiD) to capture three key effects. First, entry increases in games relative to non-games by 33%. Second, most of the entry effects are driven by “niche” app types that were more difficult to find before the category split. Lastly, the quality of new games - as measured both by consumer ratings and app size in MB¹³ - fell after the split.

The welfare implications from these effects point in different directions. If consumers like variety, entry should be welfare enhancing. As well, the fall in *marginal* search costs should also benefit consumers. On the other hand, with a larger number of products available consumers may potentially end up searching through more objects and increase their *total* search costs (even if marginal search costs fall). They would then not fully benefit from the additional variety in the market. Moreover, consumers should also like quality. Conditional on the number of products, a greater share of low quality products would reduce consumer welfare. In the presence of search costs, a larger share of low quality products could make it harder to find high quality products. This would also offset welfare gains.

To measure and decompose the welfare implications of the re-categorization, I set up a structural model of demand and supply. On the demand side, following Moraga-Gonzalez, Sandor and Wildenbeest (2015), I propose and estimate a model that merges a logit model of differentiated product choice with a search model where consumers search through products in any category. This demand model estimates

¹⁰Increasing the number of categories should not *always* reduce search costs. At some point, having too many categories can result in a more difficult search process. There is probably an optimal number (or range) of categories to display in this market, but this paper cannot say what this number (or range) may be.

¹¹For a recent example of another re-categorization, see: <https://android-developers.googleblog.com/2016/07/introducing-new-app-categories-from-art.html>

¹²See Section 4.

¹³An indicator of the number of features in the app, and correlated with the other quality measure.

consumer utility parameters and consumer search costs, using the variation from the natural experiment to identify the change in marginal search costs. I estimate two specifications - a static specification, and a dynamic specification where apps' past downloads affect their current demand.

To capture the effect of the category split, I allow the search cost parameters (which depend on search technology) to be different before and after the split. I find large differences in these parameters, showing that marginal search costs fell by as much as 50%. The demand estimates also suggest that a 1% increase in search costs reduces consumer utility by approximately 3 cents - or 2% of a paid app's average price. I also allow the other demand parameters to vary between the pre-split and post-split periods, and I do not find any significant differences.

The supply side of the model is an incomplete information oligopoly game of market entry and choice of product characteristics (Seim 2006, Augereau, Greenstein and Rysman 2006). Unlike most previous papers in this literature, firms choose both horizontal characteristics (which categories to enter, if any) and vertical characteristics (quality level). The estimation of this game identifies entry cost parameters and variable profit parameters. In specifying firm profits, I take into account that most of the apps are free to download, and firms make money from a combination of in-app purchases and in-app advertising. I model variable profits as a nonlinear (quadratic) function of the number of downloads.¹⁴ The number of downloads, in turn, depends on the entry decisions of other apps and consumer search costs. The specification of entry costs take into account that these costs non-linearly depend on product quality.

After estimating the demand and supply models, I measure the change in consumer welfare due to the category split. I also decompose the total welfare effect into the changes due to product variety, product quality, and falling marginal search costs. These decompositions show that welfare increases by 60% after the split in the categories, relative to the pre-split baseline. Most of the increase in consumer surplus comes from the reduction in marginal search costs. There is also an increase in consumer surplus due to greater product variety. This effect is larger than the fall in consumer surplus due to lower quality.

Reports suggest that the mobile app economy created over a million jobs in the US.¹⁵ It may then be in the public interest to incentivize developers to create more high quality apps, for example, by a subsidy to app start-ups. I simulate two additional counterfactuals that reduce the entry costs in the market by 15% and 30%. Counterfactual simulations show that these policies encourage more entry by higher

¹⁴A more downloaded app would likely have more consumers to make in-app purchases, and more consumers to view ads.

¹⁵<http://www.asymco.com/2015/01/22/bigger-than-hollywood/>

quality apps relative to the baseline, but their overall welfare effects are smaller compared to the effect of reducing search costs.

This paper is part of a large literature on the effects of consumer search costs on competition and market outcomes - starting with Stigler (1961), Nelson (1970), and Diamond (1971).¹⁶ Most of this literature is on price effects, but there are several recent theory papers that examine the interaction between search costs, entry incentives, investment in quality, and product design.¹⁷ The existing empirical literature on these topics is more sparse.¹⁸ Data suggests that the introduction of the internet in the 1990s reduced consumer search costs for books, movies, and music. At the same time, the number of these products available to consumers increased. The implication is that lower search costs increased entry. However, this evidence is primarily descriptive. This evidence also does not clearly distinguish changing search costs from other effects that could increase entry.¹⁹

This paper has three key contributions to the existing literature. First, no other paper, to my knowledge, uses an exogenous policy change to identify the effects of consumer search costs on firm entry. As stated above, past literature relies on descriptive evidence. Nonetheless, my results are consistent with past theoretical and empirical findings (e.g., Cachon, Terwiesch, and Xu 2008).

Second, this paper presents evidence on questions that have not yet been examined empirically. I show that lower search costs have an effect on product design (consistent with Bar-Isaac, Caruana and Cunat 2012, Yang 2013), as well as on product quality. The effects on quality are ambiguous in theory (Fishman and Levy 2015): on the one hand, when search costs fall in a market with vertical and horizontal differentiation, consumers can find the highest quality products, which should provide incentives to improve quality. On the other hand, if a firm invests in high quality, consumers who find their product can also find competing high quality products that are better horizontal matches. Thus, lower search costs can drive quality down. My results show that average quality falls, suggesting that the second effect dominates in this application.

Lastly, the structural model allows me measure the importance of the different

¹⁶See Stahl (1989), Anderson and Renault (1999), Armstrong, Vickers, and Zhou (2009), Chen and He (2011), and Zhou (2014), for more recent theory papers. See Syverson and Hortacsu (2004), Wildenbeest (2011), Chandra and Tappata (2011), De los Santos et al (2012), Koulayev (2014), Bronnenberg et al (2016), and Koulayev and De los Santos (2017) for recent empirics.

¹⁷e.g., Bar-Isaac, Caruana, and Cunat (2012), Yang (2013), Cachon, Terwiesch, and Xu (2008), Larson (2013).

¹⁸Goldmanis, Hortacsu, Syverson and Emre (2010), Waldfogel (2011), Brynjolfsson, Hu, and Simester (2011), Zentner, Smith, and Kaya (2013), and Aguiar and Waldfogel (2016).

¹⁹The costs of producing movies, books, and music fell in the same time period.

welfare effects. From a policy perspective, this suggests that policies which increase consumer search costs in online markets can have two separate negative effects on consumer welfare. First, there is a direct effect: consumers have higher marginal search costs that reduce their welfare. Second, there is an indirect foreclosure effect: higher consumer search costs discourage firms from entering the market, reducing product variety and welfare.

The paper proceeds as follows: Section 2 provides an overview of the mobile app market. Section 3 describes the data and presents some summary statistics. The fourth section presents the reduced form results. The fifth section presents the specification and estimation of the structural model, and the counterfactuals. The final section concludes.

2 App Market Background

2.1 Users

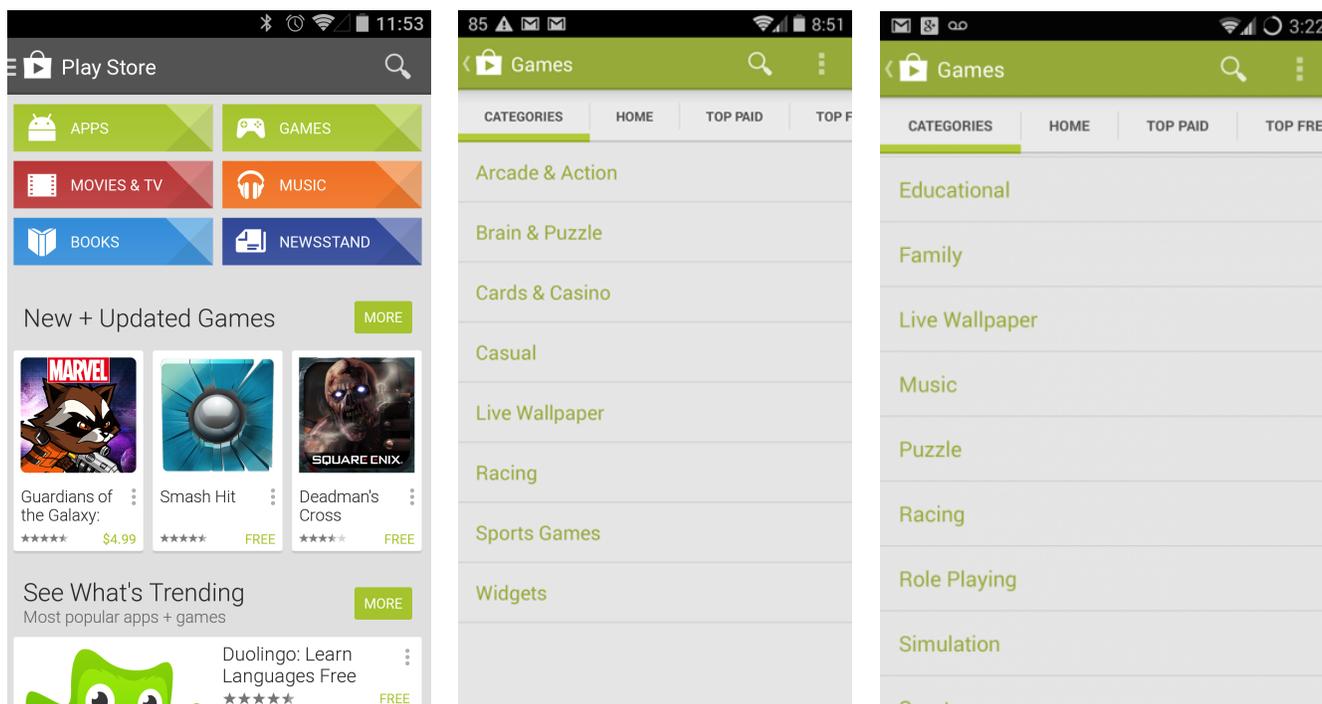
The Google Play store comes pre-installed on any phone that runs Android OS. The first screen of the Google Play store (c. 2014) appears below in the left panel of Figure 1. When a user open the Google Play store on their phone, they see a number of tabs. They can choose to look at Games, or Apps (non-game apps), or they can choose to look at individual “featured” games/apps/music/movies/books that appear directly on the first page (this would be the “New + Updated Games” in the Figure). Once they choose a product type, Games for example, users can choose to look for more specific product types by choosing a “category.” The middle panel of Figure 1 shows the choice of game categories in late 2013, and the right panel of Figure 1 shows the choice of game categories in mid 2014. I am exploiting the change in this feature as the natural policy experiment. Rather than 6 categories (in addition to the “widget” and “live-wallpaper” category), Google Play split their game categories into 18 different types in March 2014.²⁰

Once users choose a category, they can either look at a panel of “featured” products from that category, or they can look at top-lists, which display the apps with the largest number of downloads in approximately the past week. The exact algorithm determining the position of an app in the top lists is unknown, but it is correlated to downloads.²¹ The top lists are Top Paid, Top Free, Top Grossing, and Top New Paid and Top New Free, arranged in that order horizontally. The left panel of Figure

²⁰For a full list of categories before and after the split, see Appendix A.

²¹<http://www.adweek.com/digital/google-discloses-how-search-for-google-play-works-for-the-first-time-12-percent-of-dau-search-for-apps-daily/>

Figure 1



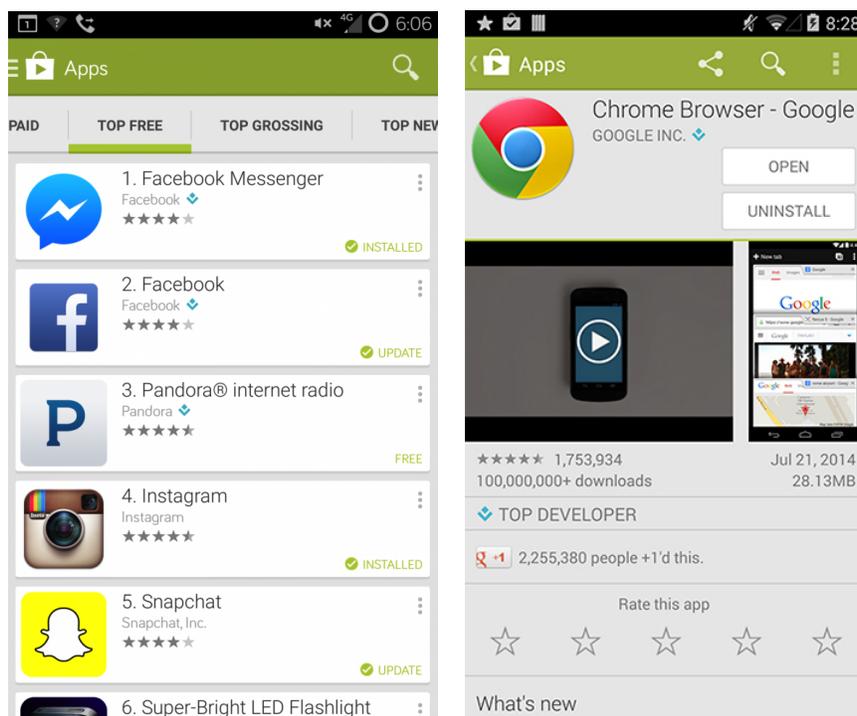
2 below shows the top list for all free Apps. At that point, users only observe apps' names, their icons, their position in the list, their average user ratings (the average "star" rating of the app), and their price. They observe the same information about featured apps. Once they click on a particular app listing they obtain more information about the app (see the right panel of Figure 2). In particular, users get to observe a number of screenshots from the app, the average rating of the app, how many people have downloaded this app, the size of the app in MB, and a textual description of the app. It is at that point that they choose whether they want to download (or purchase) the app.

2.2 Developers

The costs of publishing an app on the Google Play market are zero - each developer has to pay a one time fee of \$25 to register with Google Play and then can publish apps for free.²² On the other hand, developing an app can be costly. At the low

²²<http://www.techrepublic.com/blog/software-engineer/app-store-fees-percentages-and-payouts-what-developers-need-to-know/>

Figure 2



end, the cost of programming an app (in computer equipment, software, and wages for programmers, designers, and debuggers) can be as low as a few thousand dollars. Companies that develop those low cost apps generally consist of a few individuals who work together for a few days or weeks. On the higher end, apps that link up to databases (e.g., calendar applications, notes applications, texting applications), can cost up to and above 50,000 dollars.²³ At the very high end, apps that require more inputs can cost as much as hundreds of thousands or millions of dollars (e.g., mobile games with advanced 3D graphics, or a social media application with video chatting).²⁴ Instagram, for example, spent over \$500,000 in venture capital money on various technologies designed to accommodate a scalable photography-based social network.²⁵

When developers introduce a new app into the Android market, their key choice is

²³<https://www.appcoinc.com/blog/much-cost-build-app/>

²⁴<http://savvyapps.com/blog/how-much-does-app-cost-massive-review-pricing-budget-considerations>

²⁵<http://instagram-engineering.tumblr.com/post/13649370142/what-powers-instagram-hundreds-of-instances>

the category into which they introduce their app into, since it matters for consumers' search.²⁶ On Android (unlike the Apple App store), the category choice is mutually exclusive and you can only choose one category at a time.

Developers make money from their apps in one of three ways. First, they can charge consumers upfront for downloading their app. These are the “paid apps” and in Google Play they constitute a minority, about 20%, of the total number of apps. Median paid app price is \$1.5, the price of an app in the 90th percentile is \$5, and the price of an app in the 99th percentile of the price distribution is \$29.99 (see Figure 3 below for a more detailed price distribution).

Figure 3



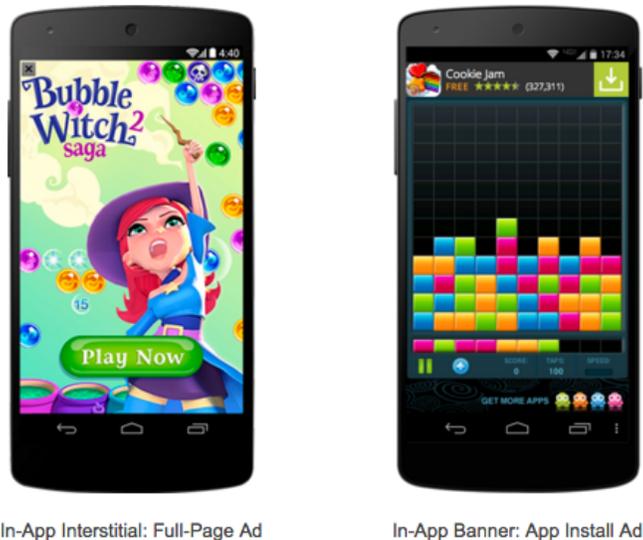
Second, the developers can allow users to download their apps for free, but place advertisements in the app. These are full-page pop-up ads (see left panel of Figure 4), banner ads (see right panel of Figure 4), or video ads. The revenues from in-app advertising are similar to the revenues from search advertising - both depend on the number of users, the type of the ads, the frequency with which ads are shown, and other factors. There is a competitive market for advertisers, who bid in second price style auctions to display their advertisements in different types of apps.²⁷ This

²⁶See previous subsection for more detail on the way consumers search the app market.

²⁷This side of the market is beyond the scope of this paper and is not modelled.

auction system is similar to the Google search advertising system, which suggests that the revenues for apps highly depend on their downloads and their type. Varian (2007) shows that bids in Google search auctions are increasing in rank and are convex (sometimes exponential), meaning that the bids for top slots are much higher than bids for lower slots. Further anecdotal evidence suggests that the average price paid for 1,000 shown ads (“impressions”) in the US is approximately \$2-3.²⁸ It is likely that there is substantial variation across apps - with top apps receiving substantially higher rates, as they are more attractive to advertisers, whereas apps which are lower ranked faring much worse.

Figure 4



Source: <https://www.google.com/admob/>

Lastly, the developers can allow users to download their apps for free, but sell them features or products within the app (so-called “in-app purchases”). Those could include additional levels for games, subscriptions for magazines (or particular magazine columns) and so on. The popularity of these business models reflects their profitability - indeed, many popular games such as Clash of Clans are free but can make hundreds of millions of dollars from a combination of in-app purchases and advertising.

²⁸<https://www.quora.com/If-you-have-made-a-free-Android-app-with-10-000+-downloads-how-much-ad-revenue-are-you-making>

As the platform, Google also makes money from the app market. First, it obtains \$25 for every new developer who enters the market. Second, Google takes a 30% cut of paid app downloads. It also takes 30% of the in-app-purchases made on free apps.²⁹ Of the remaining free apps who use app advertising to make money, about 50% are using Google’s advertising platform, Admob, meaning that Google gets a cut from the advertising revenues.³⁰ Lastly, even if Google does not earn direct money from other advertising platforms, they do get valuable data from all apps that helps them optimize their own search results and advertising.

3 Data

3.1 Data Description

The data in this paper comes from AppMonsta.com and consists of daily snapshots of all apps on the Google Play store, aggregated at the weekly or monthly level, starting from January, 2012, up to December, 2014.³¹ This is the first paper to use this particular version of the dataset. Liu, Nekipelov, and Park (2014) use a similar dataset from the same source, but their time period is 2011-2012. The dataset contains everything that consumers observe on the Play store - app price (in USD), a histogram of the ratings the app received (ranging from 1 to 5), app size (in MB), the number of screenshots the app shows, the number of video previews the app shows, and a download range for the app (referring to the total number of downloads the app received over its lifetime). Additionally, I observe the app’s category, the name of the app’s developer, and a textual description of the app, explaining what it “does.” Lastly, I observe the “top lists” for every category, which are the top 500 best-selling free and paid apps in each category for each week.

3.2 Data Management - Predicting App Downloads

Most of the reduced form results in Section 3 (e.g., the effects of the policy change on entry, quality, and prices) do not rely on measuring app downloads. However, information on per-month app downloads is necessary for the estimation of the structural demand model in Section 4.

²⁹<https://support.google.com/googleplay/android-developer/answer/112622?hl=en>

³⁰<http://www.appbrain.com/stats/libraries/ad>

³¹Weekly aggregation is used to predict the downloads of each app (see following section), and the monthly aggregation for the rest of the analysis.

The raw data includes a range of downloads that an app has accrued over its lifetime - for example [1-5] downloads, or [1,000-5,000] downloads, or [1 million - 5 million] downloads.³² Since this range is observable monthly or weekly, it is conceptually straight forward to define “per-period downloads” as the difference in lifetime downloads - for example, it can be the difference in the lower bounds of lifetime downloads, or in the average of the lifetime downloads.

An issue with this measure is that the size of the range increases with the number of downloads. For example, the ranges start at 4 downloads ([1-5], [5-10]), but then increase to 40 ([10-50]), and eventually to 400 million ([100 million - 500 million]). This introduces two sources of measurement error, which become worse for more successful apps: (1) this measure will overstate the per period downloads for successful apps that move from one bandwidth to another. For example, if a firm has a bandwidth of 100 thousand to 500 thousand downloads, and then they move up to 500 thousand to 1 million downloads in the next period, it could mean that they sold 500 thousand units this period, or 3 units. (2) this measure will understate the per-month downloads for successful apps that do not move bandwidths, since moving bandwidths is harder the more successful you are. For example, an app in the [100 million - 500 million] download bandwidth can sell millions of units every month while still remaining in the same bandwidth.

To recover the weekly or monthly downloads of apps in the Google Play market, I rely on two features of the data. First, the bandwidth of lifetime downloads for new entrants is equal to the bandwidth on per-week downloads: I know that a new app in the 10 thousand to 50 thousand range sold between 10 thousand and 50 thousand units. Second, I observe the weekly category rankings which reflect the 500 most-downloaded apps in each category roughly over the past week.³³ If I look at the market at a weekly frequency, then I know the rankings and downloads of new apps (see Appendix B for summary statistics regarding these apps).³⁴ This allows me to use these apps to predict the downloads of other apps in the market, after making an assumption about the relationship between app ranks and downloads.

³²See full list of ranges in Appendix B.

³³It is not known how the lists are determined, but Google releases (<http://www.adweek.com/socialtimes/google-discloses-how-search-for-google-play-works-for-the-first-time-12-percent-of-dau-search-for-apps-daily/539639?red=im>) as well as anecdotal industry evidence (<https://www.quora.com/What-are-ways-for-your-app-to-rank-high-in-Google-Play>) suggest that they reflect the downloads of apps over the previous several days.

³⁴I can assign the lower bound of the bandwidth as the number of weekly downloads, the upper part of the bandwidth, or the average of the bandwidth. In the rest of the analysis of this paper I will assign the lower end of the bandwidth, although the results do not change if I assign the other measures.

Several past studies of online markets with best-seller lists assume that the rank-downloads relationship is defined by a Pareto Distribution (Chevalier and Goolsbee 2003, Chevalier and Mayzlin 2006, Garg and Telang 2012).³⁵ That is, there is a negative exponential distribution where an app at rank n has exponentially fewer downloads than the app at rank $n + 1$. I can use this assumption and fit a Pareto distribution (which consists of one parameter) for every week and category around the observations that I have for new apps. I do this by running an OLS regression of the logarithm of the rank of new app j in category c at week t on the logarithm of the downloads for every category and week:

$$\ln(\text{Downloads}_{jct}) = \delta_c + \delta_{month} + \beta_{month} \ln(\text{Rank}_{jct}) + \mu_{jct}$$

where the δ s are category and month dummies, and where μ_{jct} is a random mean zero variable representing measurement error. The β_{month} is a separate slope coefficient for every month.³⁶ As the dependent variable *Downloads* I use the lower bound of the bandwidth (minimum downloads in a week).³⁷ After running the regression (see Appendix B for results tables), and knowing the number of apps in each category in each period, I can predict the downloads of all apps in the market (e.g., if a category has 2,000 apps, I can generate a prediction for the downloads of the 2,000th app). Only the top 500 ranks are observed. To generate rankings for the unranked apps, I sort them based on their number of cumulative lifetime downloads and their age, and then break up any ties by randomizing.³⁸

This prediction algorithm depends, in part, on variation in the rankings of apps over time. New apps should be able to enter into the rankings at different points in the distribution for me to estimate the Pareto relationship accurately. This seems to be true in the data. While there is a large proportion of apps that do not change their rankings from week to week, there are also many apps that move at least two

³⁵It is possible that the Pareto distribution is not entirely correct for predicting downloads in this market. In particular, Eeckhout (2004) and Gabaix (2016) show that the Pareto distribution accurately predicts the rank-size relationship for the upper tail of the distribution but not for the lower tail. There, the relationship is more accurately characterized by the log-normal distribution. However, since the Pareto distribution has only one parameter, whereas the log-normal distribution has two, the Pareto distribution is simpler to estimate.

³⁶I have also experimented with a single slope coefficient, as well as slope coefficients that vary by category, and the results do not change qualitatively.

³⁷Results using the upper bound or an average overstate the number of downloads - for example, each of the top 50 apps have over 10 million downloads weekly. However, the main results still hold qualitatively.

³⁸To check that randomization does not in itself generate any of the results, I've rerun the analysis several times with different randomized seeds. The results do not change.

spots on a weekly basis. A graph showing the distribution in weekly changes in app rankings is in Appendix B.

3.3 Descriptive Statistics

Table 1: Summary statistics at the App Level

Variable	Mean	Std. Dev.	N
<i>App Level</i>			
Game App	0.168	0.374	2.6 million
Paid App	0.2	0.4	2.6 million
<i>App-Month Level</i>			
Min. Cumulative Downloads	38,261	1.9 million	33.7 million
App Size (in MB)	21.99	29.75	33.7 million
Predicted Downloads	559	25,169	33.7 million
Number of Screenshots	4.71	3.54	33.7 million
Number of Videos	0.09	0.28	33.7 million
Mean Rating	4.0	0.66	27 million
Price (for Paid Apps)	3.27	8.93	6.8 million

Note: Mean app rating calculated for apps with 5 or more ratings.

Table 1 shows some summary statistics at the app level, where there are approximately 33.7 million app-month observations in the dataset, consisting of 2.6 million unique apps. Of these, approximately 17% belong to game categories, whereas the rest are non-game apps.

4 Reduced Form Evidence

4.1 Data Management - Categories for Reduced Form Analysis

In order to examine entry in this market at the category level, I need to calculate the number of unique apps that appear in each month in each category. As the category composition changed in March 2014, the categories are not directly comparable before and after March 2014. However, since I can track apps over time, I can construct “counterfactual categories” for the pre-March 2014 data. For example, even though

the “Family Games” category did not exist before March 2014, I can see all the apps that switched into this category from the old categories (“Arcade & Action”, “Card & Casino”, “Board & Puzzle”). Assuming that the apps that switched into “Family Games” would have always entered into this category, I can then calculate how many apps would have entered into “Family Games” before March 2014.³⁹

If an app entered and exited the market before March 2014, it does not have a post-March 2014 category. If I ignore these apps, this creates a measurement error that would be worse the further I go back in time. I resolve this problem by exploiting the detailed textual descriptions of the apps found in the data. I use a Random Forest machine learning algorithm that first maps the descriptions of the classified post-March 2014 apps into categories, and then goes back and applies this mapping to the descriptions of apps that entered and exited the market before March 2014.⁴⁰ This is very similar to how Liu, Nekipelov, and Park (2014) map Google Play categories into Apple iTunes categories.

These “counterfactual” categories do not take into account the strategic decisions of apps, but are a purely mechanical allocation based on the apps’ descriptions. It is entirely possible that if strategic decisions were taken into account, the categories would be somewhat different. The entry model in Section 5 tackles some of these issues.

Table 2 shows some summary statistics at the predicted category level. There are 42 categories, 18 of which are game categories. The game categories, on average, are much smaller than the non-game categories in terms of the number of apps - the average game category is less than a quarter of the size of the average non-game category. In terms of the number of downloads - the total cumulative downloads of apps in that category - they are more similar.

Table 3 shows the average number of apps in the predicted categories for the pre-split period, roughly splitting them up into quartiles. There are wide differences in category sizes. The smallest - most “niche” - app groups, such as Role Playing and Music Games, are roughly 1 percent of the largest. What this means, in the pre-2014 actual market structure, where there are only 6 game categories, consumers specifically looking for new music apps would have to search through the Arcade &

³⁹This is not a “true” counterfactual since it does not take into account the strategic decision-making of apps. See below for more.

⁴⁰More formally, after removing “stopwords” (e.g., “and”, “or”) I convert app descriptions into vectors of words and then into term frequency-inverse document frequencies. This method allows me to assign the highest weight to words appear frequently in a particular description relative to the average description. I use the April 2014 apps as the training set for the Random Forest classifier (I obtain similar results with other classifiers such as KNN). I then apply the classifier to apps in every month prior to March 2014.

Table 2: **Summary Statistics at the Category-Month Level**

Variable	Mean	Std. Dev.	N
Game Categories			
Number of Apps	7,256	12,990	630
Number of New Apps	452	954	630
Non-Game Categories			
Number of Apps	32,656	30,000	840
Number of New Apps	1,522	1,902	840

Action category or the Brain & Puzzle category in the hopes of finding a relevant app. Since those categories would have also contained bigger app groups (e.g., Arcade, or Puzzle), role playing and music games would be difficult to find.

Table 3: **Average Number of Apps in Game Categories in 2012**

Quartile	Quartile	Avg. N Apps in 2012
Top Quartile	Arcade, Card, Puzzle, Casual	11,788 (5,298)
2nd Quartile	Action, Board, Casino, Racing, Sports	1,251 (1,207)
3rd Quartile	Education, Strategy, Trivia, Word	209 (18)
Smallest Quartile	Adventure, Family	91 (49)
	Music, Role Playing, Simulation	

4.2 Entry

Figure 5 below is a simple plot showing the number of new apps appearing each month in the game categories and in the non-game categories. The left hand panel shows absolute monthly (log) entry levels for games and non-games. The right hand panel shows the ratio of the number of games entering the market relative to the number of non-games.

The left hand panel shows that there are large changes in absolute entry for games and non-games over the sample period. There are several spikes and troughs over the sample period, and a sharp absolute increase in the number of entering games and apps in mid-2013. There can be many drivers of this variation. For example, Christmas or “back-to-school” shopping appears to be responsible for the peaks in 2012. The release of new phones or OS can also drive absolute app entry. The large absolute increase in entry in mid-2013 appears to be driven by the release of the Samsung Galaxy S4 - a flagship Android phone which became the best-selling smartphone in the world.⁴¹ This dramatically increased the market size for Android apps and generated much higher returns for entry.

The right hand panel shows that despite the large absolute changes in game and non-game entry over time, the relative trends of game and non-game entry in 2012 and 2013 were nearly constant. The ratio of the number of new games to the number of new apps was around 0.15 (approximately 7 new apps entered for every new game) for this entire period. This changed following the announcement of the split in the game categories in late December 2013. The left hand panel shows that starting February 2014 there is an increase in entry by game apps but not by non-game apps. This increase in entry persists and peaks right after the actual split in the game categories in March 2014, but it remains afterwards as well.⁴² On the right hand panel, this is a clear shock to the ratio. Starting in February 2014, the ratio of new games to new non-games experienced a shock, moving to about 0.25 for the rest of the sample period - meaning that in 2014, only 4 new apps entered for every new game that entered.

It is reasonable to interpret the change in entry around the split in the categories as a response by developers who were incentivized by the announcement (and then creation) of new categories to produce more games rather than non-game apps.⁴³ This new entry could have come from multiple sources - either the developers started creating completely new apps from scratch to compete in the new categories, or the developers quickly released already developed products into the market early in order to take advantage of the impending changes.

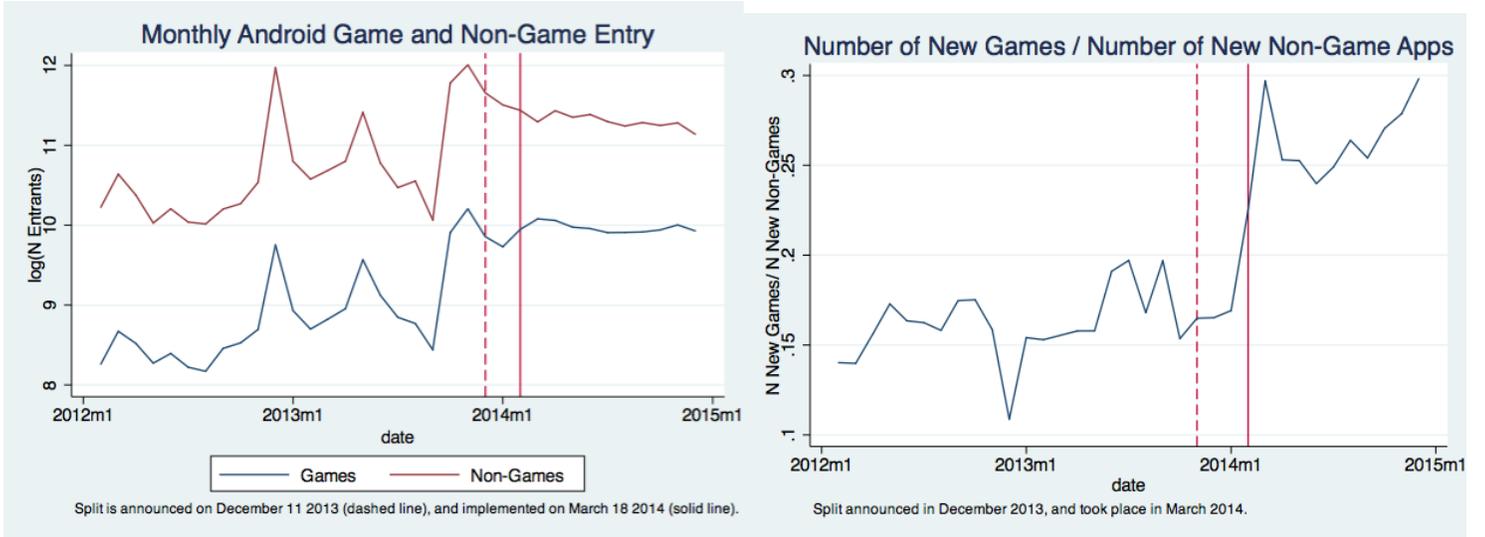
The relatively constant difference between game and non-game entry suggests that a difference-in-differences estimation strategy is appropriate to look more care-

⁴¹http://www.gsmarena.com/samsung_has_reportedly_shipped_20_million_galaxy_s4_units-news-6313.php

⁴²This is also true at the weekly level: see figure is in Appendix C.

⁴³The announcement did not set a strict date for the implementation of new categories, but mentioned that the change will happen in the first quarter of 2014: <https://9to5google.com/2013/12/10/google-play-store-to-add-new-game-categories-in-february-2014/>

Figure 5



fully at developers' entry decisions - comparing the decisions of the developers to produce game vs non-game apps before and after the split in the game categories occurs. The non-game categories thus serve as a control group, and the game categories serve as the treatment. As well, the panel nature of the data allows me to do a formal timing test of the treatment (see Appendix C).

More formally, the logarithm of the number of apps that enter into category group $c \in \{GAME, NONGAME\}$ at time t - Y_{ct} - can be represented as:

$$Y_{ct} = \tau(Game_c \times Post_t) + Game_c + \delta_t + \epsilon_{ct} \quad (1)$$

where δ_t are time fixed effects, $Post_t$ is a dummy that is equal to one after the game categories have been split, and $Game_c$ is a dummy variable equal to one for the game category group and zero for non-game category group (including category and time fixed effects causes $Post_t$ to drop out of the estimating equation). The coefficient of interest in this regression is τ , which captures the treatment effect on entry for the game categories, relative to non-game (app) categories. If this coefficient is positive, then it means that developers produced more game apps as compared to non-games apps after the split.

Estimating Equation (1) pools together entry in all game and non-game categories into "category groups" (it is Figure 5 with standard errors). If I use app descriptions to allocate apps into post-split game categories for the pre-split period (see Section 4.1), I can also run this regression at the category level. That is, c can refer to

different categories (e.g., “Productivity Apps”, “Sports Games”, etc) as follows:

$$Y_{ct} = \tau(\text{Game}_c \times \text{Post}_t) + \delta_c + \delta_t + \epsilon_{ct} \quad (2)$$

where Game_c is a dummy representing the categories which are game categories, and δ_c is a category specific fixed effect for the game categories (the average non-game category is the reference group).

Table 4 shows results from the first three regressions. Column (1) shows the regression with the data pooled across category groups, Column (2) shows the regression for separated categories without group fixed effects, and Column (3) shows the same regression as (2) but with category fixed effects, where the reference group is the average non-treated category.

Table 4: **Regression Estimates for log(N Entrants)**

VARIABLES	(1) Category Groups	(2) Separate Cats	(3) w/ Cat FE
Games \times Post Split Period	0.334*** (0.072)	1.466*** (0.130)	1.472*** (0.101)
Games (Treated Group)	-1.706*** (0.093)	-2.561*** (0.092)	
Time FE	YES	YES	YES
Category FE		NO	YES
Observations	70	1,469	1,469
R-squared	0.965	0.541	0.794

Robust standard errors in parentheses

*** p<0.01, ** p<0.05, * p<0.1

The key treatment coefficient in the first column is approximately 0.3 which suggests that the following the split, developers entered 30% more apps in the game categories than in the non-game categories relative to the time before the split. This coefficient is also highly statistically significant at the 99 percent confidence level. The treatment coefficient in Column (2) is approximately three times larger, since it looks at individual categories, which are smaller in size and thus produce higher relative changes (recall, the dependent variable is measured in logarithms). That is, 150 percent more apps entered into the average game category as compared to the average non-game category.

Appendix A, which shows the complete list of pre-split and post-split categories suggests another potential DiD regression candidate. In particular, three of the game categories that existed in the pre-split period - “Sports Games”, “Casual Games”, and “Racing Games” - remain the same in the post-split period. As a result, they appear to be natural control group candidates for the game categories that were split. This is not the case. While the split in the other game categories may not have affected them as directly, it still changed their definition.⁴⁴ In other words, the treatment does affect the control group (in this case). The category specific treatment effects in the next two subsections suggest that the treatment did affect the three previously existing categories less on average. However, this appears to be related to their size (in terms of the number of apps), as new game categories of similar size had similar treatment effects.

A potential concern about these treatment effects is that the change in entry is not driven by the split in the categories but by another factor that is then confounded with the re-categorization of games. I can do a formal timing test to check whether the τ coefficient captures something other than the split in the categories by interacting the treatment effect with the monthly time dummies. Results from the timing test, presented in Appendix C, show that the treatment effect becomes statistically significantly positive around the month of the split. This means that the treatment effect would have to confound an event or a change that occurs in the month of the treatment, which is unlikely.

Possible confounding factors could include the release of new phones or changes in the demographics of consumers or in their app usage habits. There was no significant Android phone release in early 2014. As mentioned above, the previous significant scale phone release was the Samsung Galaxy S4 in mid 2013. As well, there is no evidence that player attitudes were changing differentially for games and non-games. Aggregate data from the Nielsen Insights and Mobile NetView Surveys from 2012 to 2014 show that while per-consumer monthly game usage is increasing from 2013 to 2014, it was also growing as rapidly from 2012 to 2013. Non-game per-consumer usage was also growing at the same rate.⁴⁵ The same survey also suggests that there are no obvious changes in the demographics of Android users between 2013 and 2014.

⁴⁴For example, some apps that were previously considered to be “Sports Games” or “Casual Games” may now consider themselves to be something else and change to one of the new categories. Then, new apps that enter the market later observe the changed roster of apps in the previously existing categories and may make different decisions as a result.

⁴⁵Source: <http://www.nielsen.com/us/en/insights/news/2015/so-many-apps-so-much-more-time-for-entertainment.html>

4.3 Product Design

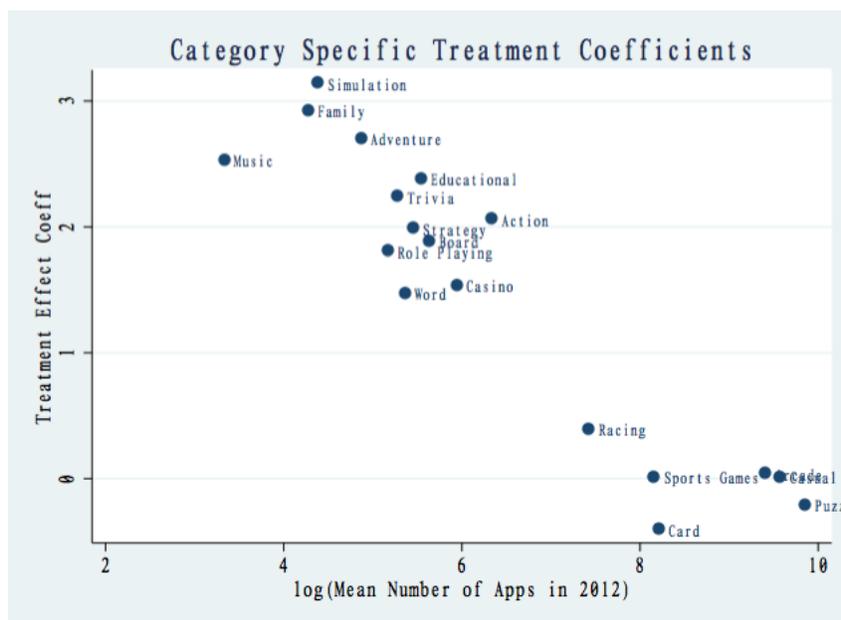
I can examine changes in product design decisions by firms by looking at the heterogeneous effects of the game category split. It is reasonable to expect that more niche game genres should benefit more from the re-categorization than the larger (more mainstream) game genres. The reason is that it was harder for consumers to find more niche products (e.g., Music Games) as compared to more popular products (e.g., Arcade Games).

The category specific treatment effects regression is as follows:

$$Y_{ct} = \sum_c \tau_c (Post_t \times \delta_c) + \delta_c + \delta_t + \epsilon_{ct} \quad (3)$$

where the τ_{cs} are the game category specific treatment coefficients, and the rest of the notation is similar to that in the previous section. The (game) category specific treatment coefficient estimates from that regression are presented in Appendix C and in Figure 6 below.

Figure 6



With the exception of a few categories, most of the treatment effects are statistically significant at the 99 percent confidence level and positive. The results make it clear that the change in entry was primarily driven by the change in search, since it is

the smaller niche categories that have the largest coefficients. Coefficient estimates are monotonically falling with category size. Based on category size in 2012, the categories in the top quartile (e.g., Card, Casual, Puzzle) have coefficient estimates that are zero (or negative), meaning that the increase in the number of categories did not incentivize more developers to enter products there. This makes sense, since a developer who wanted to enter an “arcade” app (that is an app with characteristics which fit best into the “Arcade” category) she created before the category split would be likely to enter it into the very large category “Arcade & Action”, of which “arcade” games constituted a large part. There is likely little difference in incentives between this category and the “Arcade” category in the probability of the relevant consumers finding her app. For more niche categories, the change in consumer search costs would have been large, as would the incentives of producers.

The treatment coefficients are relative, since the dependent variable is in logarithms. As such, a large treatment coefficient for the smaller categories may mean less in absolute terms than a small treatment coefficient for large categories. The treatment coefficients are monthly - suggesting that the Family app group, for example, grows at 300% per month as compared to the average non-game category - but it starts from a low baseline of 100 or so apps. Nonetheless, compared to a growth rate of the Casino category, which grows at 150% per month but started at a higher baseline of about 1,000 apps, it is conceivable that in a short period of time the number of Family apps would overtake the number of Casino apps.

These results are entirely consistent with past theoretical predictions (Bar-Isaac, Caruana and Cunat 2012, Yang 2013) which suggest that when search costs fall, producers will enter more niche products into the market as opposed to more mainstream products.

4.4 Quality

There are multiple proxies for app quality in the data, including the ratings of the apps, the size of the apps, and the information that the apps reveal about themselves (e.g., number of photo or video previews). The first proxy is direct - the ratings that consumers give to apps. I use the app’s percentage of one star reviews, following the methodology of Chevalier and Mayzlin (2006).⁴⁶ The second proxy I use is more indirect - the size of the app in MB. This is an appropriate proxy in this setting since to give an app more features, developers have to write more code, which

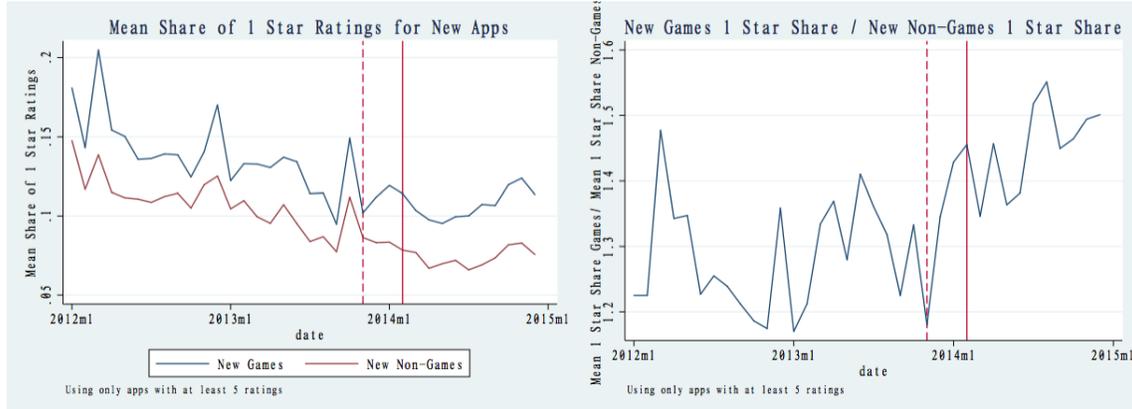
⁴⁶I limit the sample to all apps that have at least 5 reviews. The results are qualitatively and quantitatively similar when I either include all apps, or limit the sample further to apps with 10, 20, 100 or more reviews.

requires additional investment. It is possible to argue that better apps are more efficiently coded and would therefore be smaller. On the other hand, anecdotally, apps that are described as high quality integrate many features - such as Facebook or Twitter linking, or databases - that require additional lines of code and that would therefore increase the size of the app.⁴⁷ The direct and indirect measures are positively correlated - with a correlation coefficient of approximately 0.3.

4.4.1 Quality: Share of 1 Star Ratings

A time series of the mean share of one star reviews of new game and non-game apps (left panel of Figure 7 below) suggests that while the two types of apps do not have the same share of 1-star reviews - game apps have more - these shares move very similarly over time in 2012 and 2013. Both shares fall over time. Then, after the announcement of the split in the categories the mean share of 1-star game reviews starts increasing, while it continues falling (or stays constant) for new non-game apps. The ratio of the mean shares (games over non-games) is constant over time, until the announcement at which point it begins increasing (right panel of Figure 7).

Figure 7



As for entry in the previous section, I run three sets of regressions, the results of which are in Table 5. Column (1) shows the regression with the data pooled into category groups, Column (2) shows the regression for separated categories without category fixed effects, and Column (3) shows the same regression as (2) but with category fixed effects, where the reference group is again the average non-treated category:

⁴⁷<http://howmuchtomakeanapp.com/>

Table 5: **Regression Estimates for Mean App Share of 1 Star Ratings**

VARIABLES	(1) Category Groups	(2) Separate Cats	(3) w/ Cat FE
Games \times Post Split Period	0.003 (0.004)	0.033*** (0.003)	0.033*** (0.003)
Games (Treated Group)	0.031*** (0.003)	0.007*** (0.002)	
Time FE	YES	YES	YES
Category FE		NO	YES
Observations	70	1,468	1,468
R-squared	0.932	0.232	0.401

Robust standard errors in parentheses

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

The results point in the same direction as the raw data above. The average share of 1 star ratings increased after the split for games relative to non-games. Although the results at the game/non-game level are not statistically significant, the results at the category level are, and show that the share of 1 star ratings increased for game categories relative to the average non-game category by approximately 3 percentage points - suggesting a decline in quality. Considering that the average share of 1 star ratings is approximately 10%, a 3 percentage point increase is economically significant.

There are two possible explanations for this: First, the reduction in entry barriers could have resulted in the entry of infra-marginal apps of lower quality. Second, Fishman and Levy (2015) suggests that the effect could be due to the search costs themselves. In their model, as search costs effectively fall for game apps, the incentives to produce additional high quality products decline somewhat because consumers are more likely to look through a number of high quality products and find the best match, so the downloads for each high quality product potentially decline to the extent that it does not make sense for firms to make the investment.

I run an additional regression with category specific treatment effects, which provide further potential evidence to both of these effects in the data. The category specific treatment effects are pictured in Figure 8, where they are plotted against the average number of apps for each category in 2012 (left panel), and the average share

of 1 star ratings in 2012 (right panel).

Figure 8

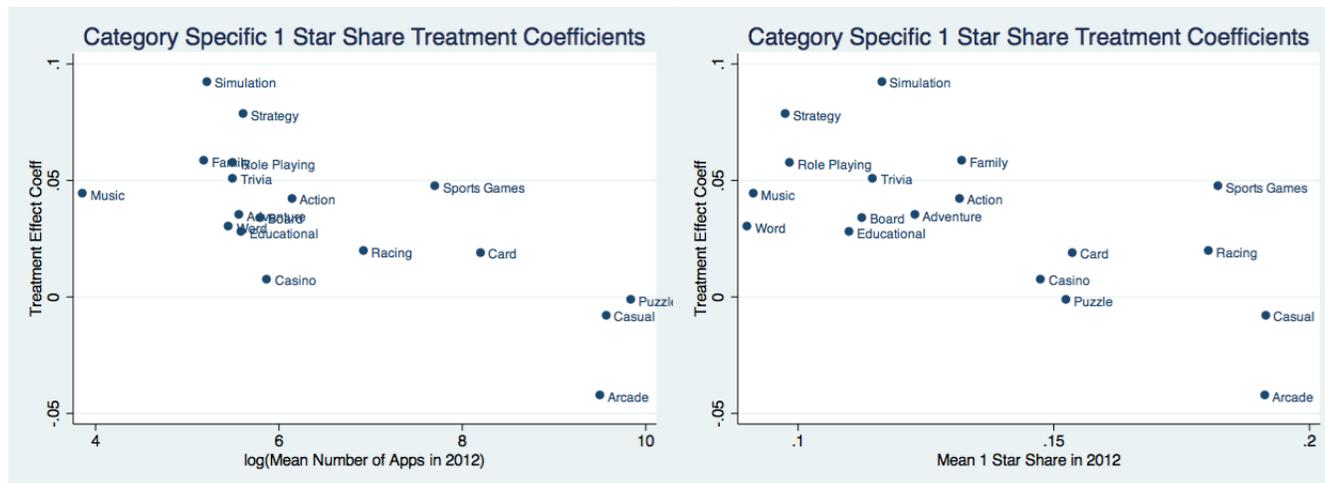


Figure 8 suggests that most of the effects came from the smaller game categories, with relatively small (or negative) average effects for the larger game categories. The largest quartile of categories includes Puzzle, Casual, and Card games. The treatment coefficients for those categories are on average close to zero or negative, as compared to the treatment coefficients for the smaller categories, which are mostly positive and statistically significant. Indeed, in the actual regression results the treatment coefficients for the largest four categories are also not statistically significant, suggesting that treatment has no effect on average entrant quality.⁴⁸ Similarly, the categories with the biggest treatment effects are the ones with the lowest average number of 1 star ratings in the pre-split period (2012) - i.e. the categories with the highest initial quality are the ones that are seeing the biggest subsequent decline in quality as a result of the split.⁴⁹

Although I am not aware of similar previous studies that explicitly look at search costs and product quality empirically, Waldfogel (2011) looks at the quality of recorded music before and after Napster in a somewhat related setting. Napster did reduce search costs in the music industry (it became much easier to find a vast number of new songs), although it also had additional effects on copyright protection and the revenues of musicians. Waldfogel (2011) suggests that the quality of recorded music did not decrease as a result of Napster, and has in fact slightly in-

⁴⁸Full coefficient estimates are in Appendix C.

⁴⁹This is not due to mean reversion for the small categories. See Appendix C for a placebo test.

creased since the pre-Napster period. This is quite different than my findings. The difference could be due to other changes in the music market at the time - such as reduced entry costs. These come through the reduced cost of recording songs, which could allow high quality artists with other barriers to entry to enter the market.

4.4.2 Quality: Mean Size of the App (in MB)

Table 6: **Regression Estimates for Mean App Size (in MB)**

VARIABLES	(1) Category Groups	(2) Separate Cats	(3) w/ Cat FE
Games \times Post Split Period	2.064*** (0.601)	-5.299*** (1.234)	-5.295*** (1.074)
Games (Treated Group)	6.325*** (0.411)	16.160*** (1.101)	
Time FE	YES	YES	YES
Category FE		NO	YES
Observations	70	1,468	1,468
R-squared	0.949	0.265	0.554

Robust standard errors in parentheses

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

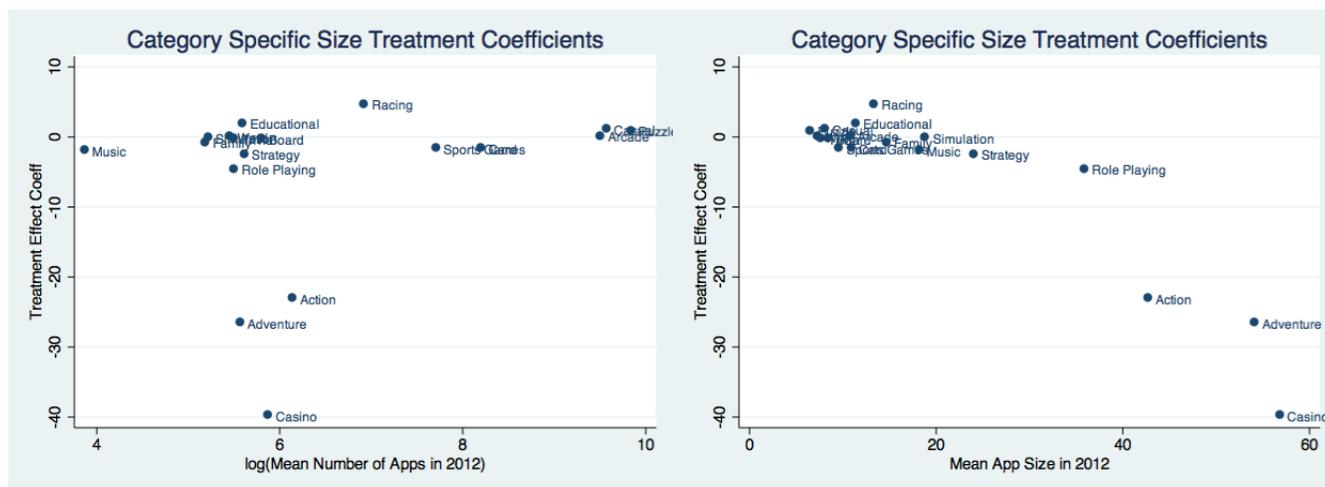
Table 6 shows regression results from the same type of regressions as Table 4, but using the mean size of the apps in MB as the dependent variable. The size of the app is representative of the quality of the app as described above. Although Column (1) at the game/non-game level suggests that average quality for games increased, Columns (2) and (3), which are at the category level, suggest the opposite. They show that, on average, new game apps experienced a decline in average app size relative to new non-game apps following the split in the game categories, representing a relative decline in quality.

The (game) category specific heterogeneous treatment effects - presented in Figure 9 below - shed some light on the difference between the aggregate and the category specific regression coefficients above. In particular, there is a large dispersion in the coefficients across the categories. Additionally, while most of the coefficients are negative, there are a few categories that have positive treatment effects, and

which could have driven the overall results (the categories with the most negative coefficients are relatively small).

The relationship between the coefficient estimates at the category level and category size is similar to the 1 star share relationship - the correlation is roughly positive. It suggests that smaller game categories experienced more substantial drops in the quality of new apps as compared to the larger game categories. Similarly, the right panel shows the relationship between average category app size in the pre-split period and the treatment coefficients. As in the ratings regression, it suggests that the highest quality categories - i.e. the categories with the highest average app size - experienced the biggest drop in quality after the split (had the largest negative coefficients).

Figure 9



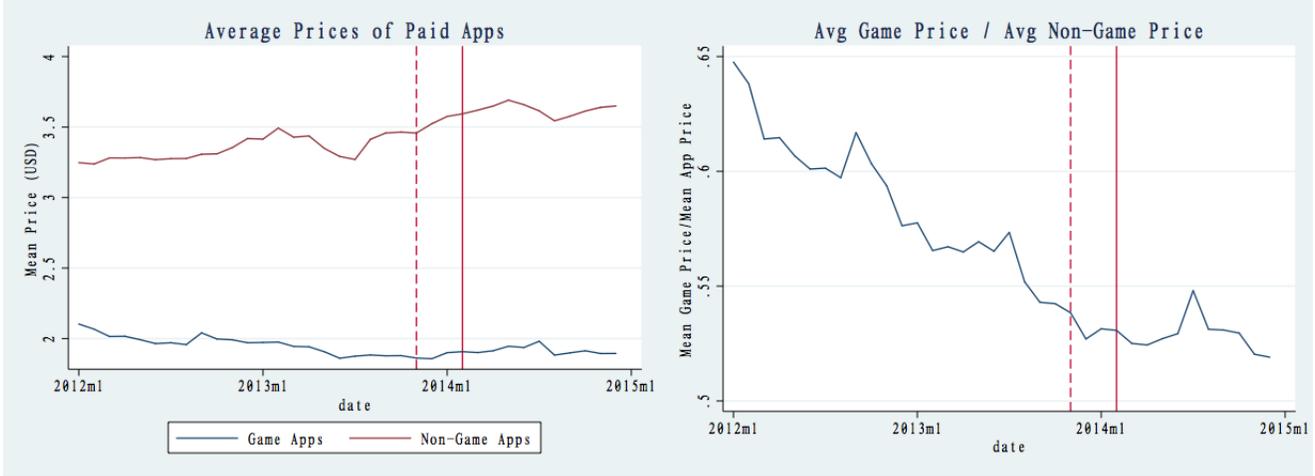
4.5 Prices

The average prices for all paid apps and for new paid apps do not follow parallel trends for games and non-games.⁵⁰ Average prices for all games fall as compared to non-games, driven by the absolute decline in game prices and the absolute increase in non-game prices over the sample period (Figure 10). The absolute decline in game pricing can be a reflection of the increasing importance of in-app advertising and in-app purchases in the app economy. Developers view more expensive paid apps

⁵⁰The ratio of free to paid apps is constant between games and non-games. It is continuously falling for both throughout the sample.

to be less profitable than less expensive paid apps or free apps, where developers recover their fixed costs more slowly over time. After the announcement of the split, the average prices of games start increasing again, meaning that the ratio of average game to non-game prices stabilizes in the post-split period.

Figure 10



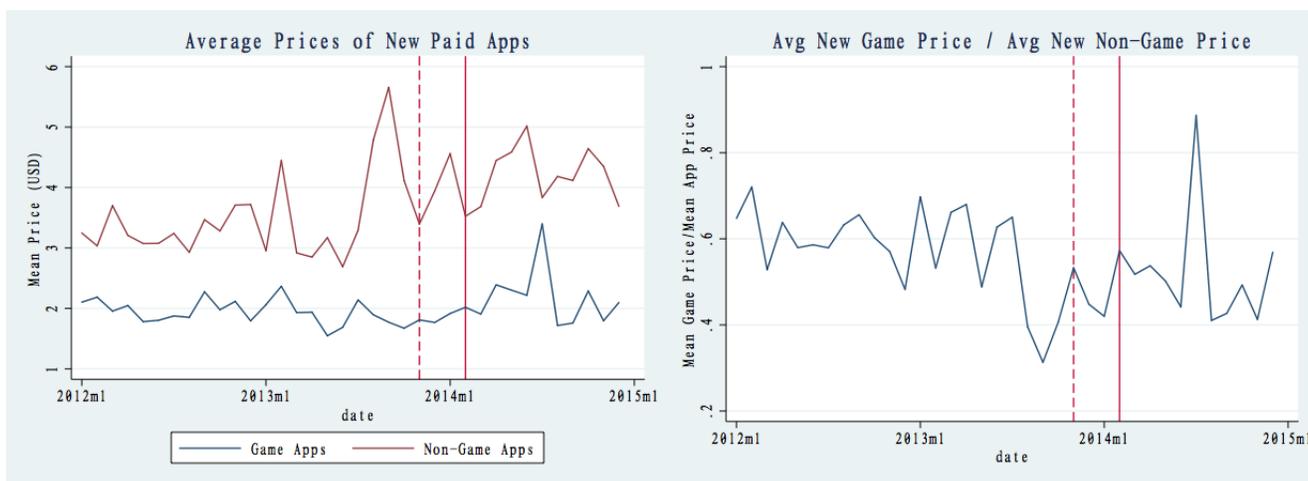
Lower search costs from the game category split could be the cause of the price changes. In particular, the average prices of new games (in the left panel of Figure 11) show a spike right after the split takes place. Similar to the effects on quality, this could be the result of increased demand or of higher valuation consumers being able to discover more preferred game-apps more easily (Bar-Isaac, Caruana and Cunat 2012).

4.6 Downloads

The results in Sections 4.1-4.4 suggest that the increase in entry (and reduction in quality) of game apps versus non-game apps was driven primarily by the increase in the number of game categories. The specific mechanism that could drive this change in developer incentives is the reduction in search costs for consumers and the improved visibility (and potentially easier marketing) for their products. If consumer search costs fall and consumers can find their preferred products faster, then I would expect downloads to increase for games vs non-games in the post-split period.

This is the case. Figure 12 shows the total number of downloads for games and non-games. The figure suggests that the aggregate number of downloads for games

Figure 11



was relatively stable and trending together with the aggregate number of downloads for non-games. At the point of the split, the aggregate number of downloads for games increased, whereas the aggregate number of downloads for non-games remained roughly constant and then trended downwards slightly.⁵¹

5 Structural Model

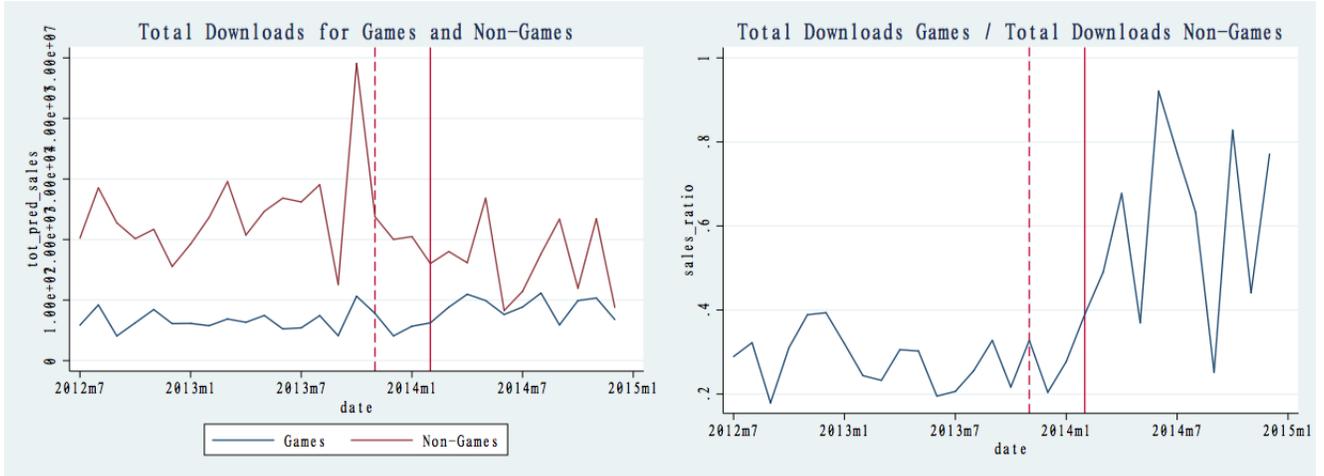
5.1 Demand

Search plays an important role in the mobile app market as it is highly unlikely that consumers are perfectly aware of all products in this market and their characteristics. Most papers integrating search into demand models make one of two assumptions about consumer search in order to identify search parameters as well as utility parameters using aggregate data: they assume that the order in which consumers examine products is known (ordered sequential search, e.g., Hortacsu and Syverson 2004, Kim et al 2010, Bronnenberg et al 2016), or they assume that consumers arrive at products randomly (random search).

In this case, these assumptions are not reasonable. It is difficult to justify as-

⁵¹If there was no market expansion, it could be possible to argue that the reduction in search costs and the increase in entry resulted in excessive entry and higher deadweight loss (Mankiw and Whinston 1986, Berry and Waldfogel 1999). Since there was market expansion, it seems that the market was initially not covered, and it is harder to claim the loss of welfare due to excess entry.

Figure 12



assumptions about the order in which consumers look at products.⁵² It is possible that consumers look at several apps in one category, and then switch to look at other apps in another category. It is similarly unreasonable to assume that consumers look at apps completely at random. App stores are specifically organized to direct consumers towards certain apps - for example, by having bestseller lists.

A small recent literature allows for demand estimation using aggregate data without making assumptions about consumer search order by focusing on the consideration sets of consumers (Sovinsky/Goeree 2008, Ching, Erdem, and Keane 2009, Moraga Gonzalez, Sandor, and Wildenbeest 2015, Honka, Hortacsu, and Vitorino 2015). The following derivation of a demand model closely follows the approach of Moraga Gonzalez, Sandor and Wildenbeest (2015).

Suppose consumers search through N apps, where the utility that consumer i receives from downloading app $j \in \{1, \dots, N\}$ is:

$$\begin{aligned} u_{ij} &= \delta_j + \epsilon_{ij} \\ &= X_j \beta + \xi_j + \epsilon_{ij} \end{aligned} \tag{4}$$

where X_j are observable product characteristics (including price), ξ_j are unobservable product characteristics, and ϵ_{ij} is a random demand shock that is consumer and product specific and that is distributed iid with an extreme value type 1 distribution

⁵²Additionally, De los Santos et al (2012) suggests that online search is unlikely to be sequential and ordered.

(mean zero, standard deviation normalized to 1). The consumer also has the choice to not buy anything (or buy the outside option), in which case they obtain:

$$u_{i0} = \epsilon_{i0} \quad (5)$$

Initially, consumers are not fully informed about all products: they do not know their ϵ s. Search resolves this uncertainty. Consumers search the market by first choosing a consideration set A of products while incurring some search cost, finding out about the ϵ s for those products, and then picking product j out of the subset A . In my application, the products in subset A can be located within one category or across categories and the subsets are unobserved. The expected benefit from choosing subset A is the maximum expected utility the consumer would obtain from those products, which due to the logit error term can be expressed as following inclusive value:

$$E[\max_{r \in A} u_{ir}] = \log[1 + \sum_{r \in A} \exp(\delta_r)] \quad (6)$$

where δ_r is simply $X_r\beta + \xi_r$. The consumer always has the outside option, regardless of the set they consider. To find set A , the consumer had to incur the following search costs:

$$c_{iA} = \sum_{r \in A} \gamma\psi_r + \lambda_{iA} \quad (7)$$

where ψ_r reflects the “distance” between the consumers and each product r in set A , and λ_{iA} is a consumer/choice set specific search cost shock, which is logit distributed mean zero with a standard error of 1.⁵³ This logit shock can be interpreted as an information shock (e.g., links) that gives consumers better access to a certain subset of products. Renaming $\sum_{r \in A} \psi_r$ as ψ_A , the utility of a consumer i of picking subset A is:

$$u_{iA} = \log[1 + \sum_{r \in A} \exp(\delta_r)] - \gamma\psi_A - \lambda_{iA} \quad (8)$$

The logit error term allows means that the probability consumer i picks subset A from the set of all possible subsets \mathbb{A} is:

⁵³The γ coefficient on ψ can vary across products or product groups. However, the assumption that the “distance” of products in a consideration set is additive in the set’s search costs is key for obtaining a closed form expression for the choice probabilities.

$$P_A = \frac{\exp(\overline{U}_A)}{\sum_{A' \in \mathbb{A}} \exp(\overline{U}_{A'})} \quad (9)$$

where $\overline{U}_A = \log[1 + \sum_{r \in A} \exp(\delta_r)] - \gamma\psi_A$.

Similarly, the probability that a consumer picks product j from subset A can be expressed as follows:

$$P_{j|A} = \frac{\exp(\delta_j)}{1 + \sum_{r \in A} \exp(\delta_r)} \quad (10)$$

Product j belongs to a large number of subsets - A_j . As a result, I can express the unconditional probability of picking product j as follows:

$$s_j = \sum_{A \in A_j} P_A P_{j|A} = \sum_{A \in A_j} \frac{\exp(\delta_j)}{1 + \sum_{r \in A} \exp(\delta_r)} \frac{\exp(\overline{U}_A)}{\sum_{A' \in \mathbb{A}} \exp(\overline{U}_{A'})} \quad (11)$$

$\exp(\overline{U}_A)$ can be expressed as $(1 + \sum_{r \in A} \exp(\delta_r)) \exp(-\gamma\psi_A)$. It is possible to show that this expression is equivalent to the following closed form expression:⁵⁴

$$s_j = \frac{\frac{\exp(\delta_j)}{1 + \exp(\gamma\psi_j)}}{1 + \sum_{k \in N} \frac{\exp(\delta_k)}{1 + \exp(\gamma\psi_k)}} \quad (12)$$

In this market share equation s_j does not depend on the subsets anymore. This is important since both the subsets and which subsets product j belongs to are unobservable to the econometrician.

5.2 Demand Estimation - Logit

The market share of the outside option is:

$$s_0 = \frac{1}{1 + \sum_{k \in N} \frac{\exp(\delta_k)}{1 + \exp(\gamma\psi_k)}} \quad (13)$$

Following Berry (1994):

$$\ln\left(\frac{s_j}{s_0}\right) = \delta_j - \ln(1 + \exp(\gamma\psi_j)) \quad (14)$$

or:

⁵⁴A full derivation can be found in Moraga Gonzalez, Sandor and Wildenbeest (2015).

$$\ln\left(\frac{s_j}{s_0}\right) = X_j\beta - \ln(1 + \exp(\gamma\psi_j)) + \xi_j \quad (15)$$

It is straight-forward to estimate this model using GMM, with BLP instruments accounting for the endogeneity of prices in the model. To properly specify this estimating equation, I need to decide which variables affect consumer purchasing utility but not search costs, and which variables affect search costs but not purchasing utility.

5.3 Specification A: Static Empirical Specification

The number of products in the category of app j is an example of a search cost shifter. Consider products in categories with more apps (larger categories). The number of apps in these categories should not affect consumer utility conditional on consumer discovery - it should only affect the probability that consumers find these products. For example, if there are 10 products in a category bestseller list, it is easier for consumers to look through all 10. If there are 50 products, it takes longer for consumers to scroll through to discover all the products. Additionally, it is also easier for products to be featured in their category if there is a relatively smaller number of products in that category.

There are a number of utility shifting product characteristics that can be included in the model. The size of the app (in MB) could be such a shifter. As well, the average star rating of the app (e.g., 2 star, 3.5 star) would reveal to potential buyers the quality of the app. These can be entered as dummy variables. For example, relative to a baseline of very low rated apps (those with an average rating of 2 stars or less), a 4 star dummy would show how much more utility consumers obtain from a higher rated app. Similarly, past literature on online markets considered information disclosure about a product as a signal of quality (Dranove and Jin 2010, Lewis 2011). With that in mind, the number of screenshots and video previews a developer includes would reveal information about the app to potential consumers. These would be two additional utility shifters.

A particularly simple demand model specification (Specification A) includes the product characteristics discussed above and the number of apps in the category as the search cost parameter. The estimating equation for app j in category c at time t would be as follows:

$$\begin{aligned}
\ln\left(\frac{s_{jct}}{s_{0t}}\right) &= \delta_{jt} - \ln(1 + \exp(\gamma \ln(N_{ct}))) \\
&= X_{jt}\beta + \beta_{paid}1\{p_{jt} > 0\} - \alpha p_{jt}1\{p_{jt} > 0\} \\
&\quad - \ln(1 + \exp(\gamma \ln(N_{ct}))) + \xi_{jt}
\end{aligned} \tag{16}$$

where X_{jt} are product characteristics, and N_{ct} is the number of apps in category c . p_{jt} reflects the price of product j which is only important if that product is not free. I also allow paid products to have different utility intercepts than free products - as reflected in β_{paid} .⁵⁵

The intuitive argument for the identification of the β parameters is similar to the standard discrete choice demand model - the X characteristics shift consumer utility and affect the purchase decisions of consumers which then affect the market shares. The intuitive argument for the cross-sectional identification of the γ parameter is as follows: two apps with the same observable utility shifting characteristics are located in categories that have a different number of apps (and different search costs). The demand for the app with more other apps in its category should be lower, and their market share should be lower as well. Alternatively, it is also possible to think about identifying the search costs over time: a given app has identical quality over time, but in different time periods there is a different number of apps in its category. Holding everything else constant, it is harder to find the app in more populous category, which reduces its demand and shifts down its market share.

Since I use observational data on app demand and characteristics, prices are likely endogenous with respect to unobservable product quality (ξ). Products with higher ξ will likely have more demand and will have higher prices, meaning that the coefficient on the price variable will not reflect the effect of prices on demand and utility. In order to control for this, I use characteristics based instruments: The average size of all other apps in the category of app j by the same developer of app j , and the average rating of all other apps in the category of app j by the same developer of app j . The characteristics of other products should be correlated with the price of product j through the supply side, but uncorrelated with the demand shock for product j .

The specification of search costs as the number of products in the market makes my estimating equation similar to the Akerberg and Rysman (2005). Their model uses the number of products in the category to account for the mechanical increase in consumer utility with a larger array of products. The intuition is that with a

⁵⁵I also run the model for free products only, since they represent 80% of the market.

larger variety, some products are “squeezed out” - for example, due to limited shelf space. This “squeezing out” effect can be related to search, but it does not necessarily reflect search costs. It can also represent some other consumer costs related to the number of products that shift the market shares. Therefore, it is not clear if the γ parameter represents marginal search costs or the Akerberg and Rysman (2005) logit model correction.

The utility parameters (β and α) are the underlying “deep” parameters that depend on consumer preferences and should be immutable. By contrast, the γ parameter is not a preference parameter, but rather a parameter that depends on the search technology available to consumers. A change in the search technology should change γ .

The Android app market setting has the benefit of a natural experiment to disentangle the search costs and other components of γ . Marginal consumer search costs should decrease for games when the split in the categories takes place, but they should stay constant for non-games. The non-search component of γ should remain stable over time. Then, the change in γ between the pre- and post-split period for games identifies the change in consumer search costs as a result of the split.⁵⁶ In the non-game categories, γ should remain stable over time.

Table 7 shows estimates of Specification A for Games and Non-Games before and after the split. Columns 1 to 4 show results for Games - Columns 1 and 3 for the pre-split data, and Columns 2 and 4 for the Post-split data. Columns 5 to 8 show results for Non-Games. The sample includes only free apps in Columns (1), (2), (5), and (6), and free and paid apps in columns (3), (4), (7), and (8).

The results from Specification A largely confirm the main hypothesis. The coefficient on the number of apps in the category falls from 0.6 to approximately 0.3 or 0.4 after the split for games in Table 7 in all specifications. This suggests that search costs fell in the games market. The coefficient on the number of apps in the category for Non-Games remains roughly constant between the two periods.

The price coefficients in Columns (3) and (4) suggest that demand for mobile games in this market is very inelastic. The median paid game has negligible market share, and a price of \$1.5, meaning that demand elasticity is at most 0.15.⁵⁷ This could be due to low price variation - there are many consumers in the market, and there are also many products with very minor price differences (recall, the vast majority of products are under 5\$) - consumers do not seem to be particularly elastic when prices change from \$1 to \$2, since these are very small changes in “real” terms.

⁵⁶In fact, the change in γ would help recover a lower bound on the “search cost” component of the parameter.

⁵⁷Mean paid game price is \$3.3, and demand elasticity in that case is 0.3.

Table 7: **Specification A - Structural Demand Coefficients Estimates**

	Games				Non-Games			
	Pre-Split (1)	Post-Split (2)	Pre-Split (3)	Post-Split (4)	Pre-Split (5)	Post-Split (6)	Pre-Split (7)	Post-Split (8)
ln(N Apps in Category)	0.679*** (0.002)	0.304*** (0.001)	0.606*** (0.002)	0.419*** (0.001)	0.321*** (0.001)	0.312*** (0.001)	0.340*** (0.001)	0.335*** (0.001)
ln(Size)	0.044*** (0.001)	0.025*** (0.001)	0.070*** (0.001)	0.036*** (0.001)	0.014*** (0.000)	-0.015*** (0.001)	0.011*** (0.000)	-0.012*** (0.000)
N Videos	0.321*** (0.003)	0.326*** (0.003)	0.242*** (0.003)	0.237*** (0.003)	0.267*** (0.002)	0.306*** (0.003)	0.177*** (0.002)	0.222*** (0.002)
N Screenshots	0.013*** (0.000)	0.023*** (0.000)	0.024*** (0.000)	0.041*** (0.000)	0.011*** (0.000)	0.037*** (0.000)	0.010*** (0.000)	0.017*** (0.002)
Paid			-1.681*** (0.009)	-2.332*** (0.005)			-2.09*** (0.001)	-1.849*** (0.002)
Price			-0.0954*** (0.005)	-0.0487*** (0.002)			0.015*** (0.001)	0.031*** (0.000)
Avg. Star Rating Dummies	YES	YES	YES	YES	YES	YES	YES	YES
Observations	1,893,941	2,005,052	1,936,683	1,958,586	11,933,563	10,797,648	15,729,906	12,834,380
R-Squared	0.395	0.373	0.534	0.568	0.363	0.388	0.527	0.490

Columns 1-4 show results for Games. Columns 5-8 show results for Non-Games.

Columns 1,2,5, and 6 include only free apps. Columns 3,4,7, and 8 include both free and paid apps.

Apps with 2 stars or less are the “baseline” category for the star rating dummies.

Price instruments for Columns 3, 4, 7, and 8 include the average characteristics of all other apps of the same developer.

Robust standard errors in parentheses

*** p<0.01, ** p<0.05, * p<0.1

Game price coefficients are very small, but Non-Game price coefficients in Columns (7) and (8) are positive, even with characteristics based instruments.⁵⁸ It is likely that the instruments used are not enough to clearly identify the effects of prices on utility separate from product quality. Figure 3 shows that the vast majority of paid apps herd on 99 cents. There is also relatively little price variation either across apps or over time. In addition, even the paid apps often have advertising or in app purchases, meaning that the prices that consumers pay in advance do not necessarily reflect the full value of the app. It is therefore not clear whether apps are competing Bertrand-Nash in prices.

There may also be identification issues with having a large number of very small firms and few markets and using characteristics based instruments. As Armstrong (2016) shows, BLP instruments can be inconsistent in these environments, since they end up having very little correlation with market shares. The weak instruments could also account for the relative instability of the utility parameters in the samples which

⁵⁸Without those instruments, they are even higher.

include paid apps and games.⁵⁹

The results from Specification A suggest that the value of increasing the size of a game by 1% (equivalent to 220 KB for the median game) is between 0.5 cents and 0.8 cents for consumers; equivalently, increasing the size of an median app by 5 MB would increase the utility of consumers by 75 cents, or half the price of the median app.

Similarly, the cost to consumers of increasing the number of apps in the category by 1% was approximately 3 cents (2% of a median app’s worth). This can reflect the marginal cost of “scrolling” through the app store.⁶⁰

The coefficients on average rating dummies (not shown for ease of reading) are positive for both games and non-game apps (relative to a baseline of apps with an average star rating of 2 or less). Apps with higher average ratings give consumers more utility. The exception to this are apps with an average rating of 5 (out of 5). The coefficient for the 5 star dummy is positive, meaning that consumers receive higher utility from a “5 star app” than a “2 star app.” It is, however, lower than all other coefficients, including of a “2.5 star app.” The reason for this is a popular strategy among firms in online markets of buying fake 5 star ratings. Luca and Zervas (2016) show that, in the context of Yelp, restaurants with low quality are more likely to buy fake reviews.⁶¹ This is likely what is happening in this market as well: low quality apps buy a large number of fake 5 star reviews which pushes their average rating up to being close to 5.

5.4 Period Specific Search Cost Coefficient Estimates

In addition to the regressions that pool together observations from different months, I also estimate *period-specific* search cost coefficients using a series of cross sectional regressions separately for every month.⁶² Figure 13 below plots the search cost coefficients for games and non-games over time. For this estimation I use Specification

⁵⁹Optimal instruments may help resolve this problem.

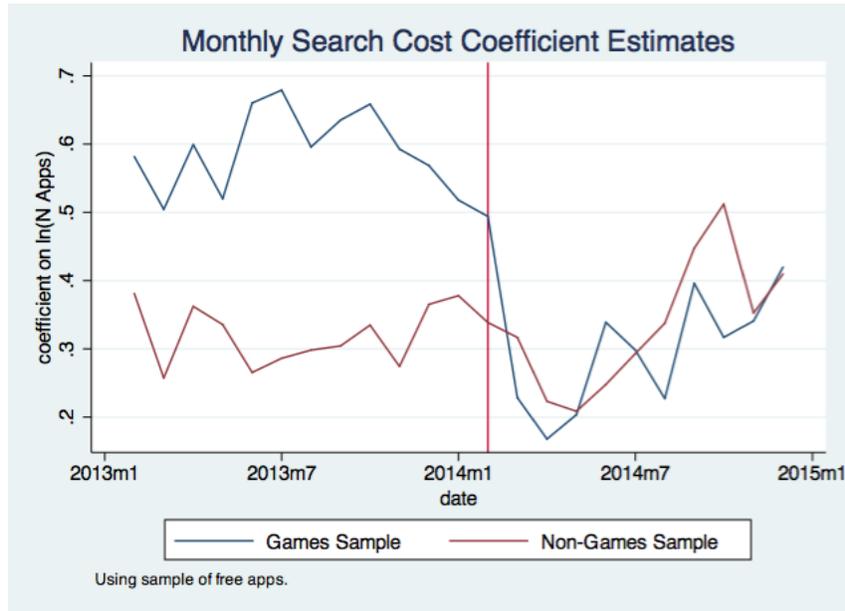
⁶⁰Search costs do not enter the utility function linearly, so the marginal rate of substitution between search cost variable ψ and price is: $\frac{MU_\psi}{MU_{price}} = \frac{exp(\gamma\psi)}{1+exp(\gamma\psi)} \frac{\gamma}{\alpha}$

⁶¹Li, Bresnahan and Yin (2016) show evidence that mobile apps buy fake downloads, though they do not discuss reviews.

⁶²I can also do the same exercise by pooling observations from all periods together and estimating interactions between the variables and time dummies. However, estimating a non-linear model with a large number of observations and fixed effects tends to present a computational problem, as the GMM criterion function becomes relatively flat and the discrete jumps from the fixed effects make convergence to the global minimum difficult (see Greene 2004 for a similar discussion about MLE estimation).

A and the sample of free games and non-games - equivalent to Columns (1), (2), (5) and (6) in Table 7.

Figure 13



The figure shows that the change in the game coefficients happens sharply at the moment of the split in the categories, rather than gradually over time.⁶³ The search cost coefficient for games hovers around 0.5-0.6 for the year prior to the split in the categories, and drops to less than 0.3 right as the categories are split.⁶⁴ By comparison, the non-games search cost coefficient does not show any drastic movement around the split. There is some general instability in the coefficients over time. The figure suggests that search costs for both games and non-games were increasing in mid to late 2014. Unlike the change right after the period of the split, this change in the coefficients is common.

⁶³Two representative monthly coefficients are plotted in Appendix D, and although they show some instability over time, the average coefficient level before the split is comparable to the average coefficient level after the split.

⁶⁴There is a small estimated drop in search costs the period before the split. This may be driven by my inability to completely capture the unobservable heterogeneity of the products. For example, if many high quality products enter the market in this period and obtain high downloads, this could appear as an overall decrease in search costs, as more products in the category will have higher sales.

5.5 Specification B: Specification with Dynamic Effects

Although Specification A captures many key features of the market, it does not capture the effect of product popularity in period t on the probability of finding the product in period $t + 1$. If a product sold more in the past, it is more likely to be on category best-seller lists and to be featured on the store, making it easier for consumers to find. Keeping the utility variables the same as Specification A, demand Specification B includes the lag of the downloads of product j as part of the search costs in addition to the previously introduced search cost shifter:

$$\begin{aligned} \ln\left(\frac{s_{jct}}{s_{0t}}\right) = & X_{jt}\beta + \beta_{paid}1\{p_{jt} > 0\} - \alpha p_{jt}1\{p_{jt} > 0\} \\ & - \ln(1 + \exp(\gamma_1 \ln(N_{ct}) + \gamma_2 \ln(q_{jt-1}))) + \xi_{jt} \end{aligned} \quad (17)$$

where q_{jt-1} are the downloads of product j at period $t - 1$. The model is now dynamic, and it has standard dynamic panel data estimation problems as described in Nickell (1981) and Arellano and Bond (1991). Namely, if there are persistent product level effects (i.e. unobserved app quality), then OLS or GMM estimates are inconsistent. Even if fixed effects are included (or the model is differenced), there is still correlation between the lagged q and the differenced error term.

Redefining the expression $X_{jt}\beta + \beta_{paid}1\{p_{jt} > 0\} - \alpha p_{jt}1\{p_{jt} > 0\}$ as $\mathbb{X}\vec{\beta}$ I estimate this model as follows: first, I difference Equation (17) above, obtaining:

$$\Delta \ln\left(\frac{s_{jct}}{s_{0t}}\right) = \Delta \mathbb{X}\vec{\beta} - \ln\left(\frac{1 + \exp(\gamma_1 \ln(N_{ct}) + \gamma_2 \ln(q_{jt-1}))}{1 + \exp(\gamma_1 \ln(N_{ct-1}) + \gamma_2 \ln(q_{jt-2}))}\right) + \Delta \xi_{jt} \quad (18)$$

where the Δ represents the difference between the variable in period t and the variable in period $t - 1$.

Standard instruments in the literature include the second lag of the downloads of product j : $\ln(q_{jt-2})$.⁶⁵ This instrument is correlated with the non-linear function of $N_{ct}, N_{ct-1}, \ln(q_{jt-1}), \ln(q_{jt-2})$ by construction. However, product downloads are imputed in this setting, likely with some measurement error. If this measurement error is correlated over time, then lagged downloads are going to be correlated with the differenced error term (which will also include the differenced measurement error between periods $t - 1$ and $t - 2$).

Instead of lags of the downloads, I use lags of BLP instruments as IVs. The intuition is straight-forward: the characteristics of product j 's competitors in period

⁶⁵Sweeting (2013) and Aguirregabiria and Ho (2012).

$t - 2$ should be correlated with the demand for product j in that period, but they should not be correlated with the demand shock for product j (or the difference in the demand shocks). In addition to these instruments, I use ΔX_{jt} .

A standard problem with estimating dynamic panel data models in differences is that of weak instruments. This may be an even greater problem in this application, since the instruments need to be correlated with a non-linear function of variables. To improve the estimates, I follow Blundell and Bond (1998) and add to the GMM system the levels estimating equation (Equation 17). The *differenced* lag of BLP instruments as IVs for the new moment conditions. Again, this instrument is by construction correlated with the lag of the downloads of product j , but it should be uncorrelated with the error term.

Table 8 shows estimates of Specification B for Games and Non-Games before and after the split. As in Table 7, Columns (1)-(4) show results for a subsample of Games apps, and columns (5)-(8) show results for a subsample of Non-Game apps. The sample includes only free apps in Columns (1), (2), (5), and (6), and free as well as paid apps in columns (3), (4), (7), and (8).

The key result from Table 8 is the persistence of the fall in category specific consumer search costs even when additional search cost controls are included. The coefficient on the number of apps in the category still falls for the Game apps regression - between Columns (1) and (2), and Columns (3) and (4) - but stays relatively constant for Non-Game apps.

The coefficient on the natural log of lagged downloads falls for both Game and Non-Game regressions. It is possible that the returns to prominence in the post-split period are higher as compared to the pre-split period, as products with more downloads find it easier to display or advertise themselves to future consumers. The magnitude of the change is very similar for both Games and Non-Games, suggesting that it is not a result of the split in the Game categories.

The magnitude of the utility coefficients is similar to the coefficients in Table 7, although there is more instability between the pre-split and post-split periods. For example, the coefficient on the presence of a video preview in the free Games regressions (Columns 1 and 2) rises from 0.18 to 0.25. This instability could be due to the presence of mis-specification of the demand function, or due to the lack of sufficiently strong instruments.

5.6 Simple Welfare Counterfactual (without Supply Side)

Consumer welfare from a set of products in a standard logit demand model can be calculated as the expected value of the products:

Table 8: **Specification B - Structural Demand Coefficients Estimates**

	Games				Non-Games			
	Pre-Split (1)	Post-Split (2)	Pre-Split (3)	Post-Split (4)	Pre-Split (5)	Post-Split (6)	Pre-Split (7)	Post-Split (8)
ln(N Apps in Category)	0.597*** (0.004)	0.444*** (0.004)	0.570*** (0.004)	0.432*** (0.001)	0.314*** (0.001)	0.313*** (0.001)	0.324*** (0.001)	0.305*** (0.001)
ln(Size)	0.047*** (0.001)	0.038*** (0.004)	0.058*** (0.003)	0.038*** (0.001)	0.018*** (0.001)	0.002*** (0.000)	-0.014*** (0.000)	-0.070*** (0.000)
N Videos	0.182*** (0.003)	0.250*** (0.004)	0.257*** (0.006)	0.210*** (0.002)	0.227*** (0.002)	0.182*** (0.002)	0.149*** (0.002)	0.138*** (0.002)
N Screenshots	0.020*** (0.000)	0.021*** (0.000)	0.023*** (0.000)	0.017*** (0.000)	0.047*** (0.002)	0.013*** (0.000)	0.043*** (0.000)	0.016*** (0.000)
ln(lagged Downloads)	-0.051*** (0.007)	-0.220*** (0.010)	-0.033*** (0.007)	-0.095*** (0.008)	-0.220*** (0.001)	-0.389*** (0.001)	-0.181*** (0.000)	-0.359*** (0.001)
Paid			-1.43*** (0.013)	-2.05*** (0.020)			-1.880*** (0.002)	-1.361*** (0.002)
Price			-0.078*** (0.004)	-0.055*** (0.002)			0.011*** (0.000)	0.038*** (0.000)
Avg. Star Rating Dummies	YES							
Observations	1,313,874	1,261,074	1,739,260	1,513,323	8,312,153	7,213,606	11,269,035	8,697,018
R-Squared	0.530	0.741	0.699	0.790	0.440	0.555	0.503	0.612

Columns 1-4 show Games Results. Columns 5-8 show Non-Games Results.

Columns 1,2,5, and 6 include only free apps. Columns 3,4,7, and 8 include both free and paid apps.

Apps with 2 stars or less are the “baseline” category for the star rating dummies.

Price instruments for Columns 3, 4, 7, and 8 include the average characteristics of all other apps of the same developer.

Instruments for lagged app downloads include the 2nd and 3rd lags of characteristics IVs.

Robust standard errors in parentheses

*** p<0.01, ** p<0.05, * p<0.1

$$CS = \ln\left(\sum_{j \in N} \exp(\delta_j - \ln(1 + \exp(\gamma\psi_j)))\right) \quad (19)$$

this measure of consumer surplus is not just the purchase utility, but also includes the search costs associated with the different products.⁶⁶ This allows me to simply examine the change in consumer surplus as the difference between the expected utilities:

⁶⁶This does not necessarily fully capture the search costs that consumers experience since those depend on the consideration set. It effectively assumes that consumers have consideration sets of single products, or alternatively a large consideration set of all the products.

$$\Delta CS = \ln\left(\sum_{j \in N} \exp(\delta_j^{powt} - \ln(1 + \exp(\gamma^{powt} \psi_j)))\right) - \ln\left(\sum_{j \in N} \exp(\delta_j^{pre} - \ln(1 + \exp(\gamma^{pre} \psi_j)))\right) \quad (20)$$

I can examine the changes in total consumer surplus as a result of the re-categorization of games. This should affect consumer surplus through two distinct welfare channels: the increase in product variety, and the reduction in search costs. By calculating consumer surplus (CS) in the post-split period but with the pre-split coefficients, I can estimate the changes in consumer surplus that come purely from changes in product variety. Then, the difference between this value and CS in the post-split period with the post-split coefficients is the value of only changing search costs (without changing product variety).

Table 9 below provides the baseline utility estimates (for the Games market) and the changes in utility for the different estimated specifications. Since there is a different number of periods (months) in the pre-split and post-split states, surplus is calculated at the monthly level, and the table below shows the average total monthly consumer surplus in the pre-split and post-split periods. Since not all specifications include paid apps and prices, welfare is not measured in dollars. Instead, I set the average pre-split consumer surplus as a 100% baseline, and then calculate the percentage change from that baseline in the post split period. For the specifications with the paid apps, I do not account for changes in prices that result from changes in competition due to changing search costs. I assume that prices remain constant.

Table 9: **Estimated Changes in Consumer Surplus**

	(1)	(2)	(3)	(4)
Average Pre-Split CS	100%	100%	100%	100%
Change in Product Variety	+22%	+17%	+16%	+11%
Reduction in Marginal Search Costs	+40%	+12%	+28%	+9%
Overall Average Increase in CS	+62%	+29%	+46%	+20%
Column (1) represents Specification A without Paid Apps				
Column (2) represents Specification A with Paid Apps				
Column (3) represents Specification B without Paid Apps				
Column (4) represents Specification B with Paid Apps				

The main takeaway from the welfare results in Table 9 is the importance of changes in search costs to consumer welfare. In two of the specifications, the change in search costs contributes over 60% of the welfare gained by consumers as a result of the policy change. The reason for this large effect is the change in the search cost coefficients. For example, marginal search costs drop by half in the first two columns of Table 7. The other specifications generate smaller coefficient changes and therefore smaller welfare changes. Also, as noted in the discussion of Table 7, coefficient estimates with the paid app sample (e.g., Columns 3 and 4 of Table 7) are not very stable due to a weak instruments problem. This instability mitigates the welfare effects of the drop in search costs, and would explain the difference in results between Columns 1 and 2, and 3 and 4 in Table 9. Even in those specifications, the reduction in marginal search costs still contributes over 40% of the increase in consumer surplus.

This welfare decomposition helps answer two key questions that the reduced form results suggest: (1) Has the re-categorization increased consumer welfare? (2) What is the relative contribution to consumer welfare of changing product variety versus changing marginal search costs?

I can gain additional insight by further decomposing the welfare effects of the change in product variety into the welfare effects from the increase in product variety *while holding the quality distribution constant*, and the welfare effects from the change in the quality distribution. In particular, holding the number of products fixed, consumers would prefer to have higher quality products. They then lose welfare from the reduction in product quality. However, if the distribution of product quality in the market was different, firm entry choices may be different as well. In order to do this decomposition, I need to model the supply side and firm entry decision.

Moreover, the decomposition in Table 9 does not assign welfare changes correctly. Though marginal search costs (the γ) do not change between Rows 1 and 2 of Table 9, the number of categories changes in the data (from 6 to 18 categories). As the number of categories increase (for a given number of products), the number of products per category falls, and search costs fall. This means that some of the welfare gains assigned to “increase in variety” in Table 9 actually come through a reduction in search costs.

To correctly decompose the welfare changes would require a counterfactual state where the marginal search costs are the same as in the pre-split period, *and* the number of categories is 18 rather than 6. This would change the entry incentives of developers and the number of products that are available in each category. As a result, it is impossible to do with the demand side of the model alone. Rather, I would need to account for entry incentives by apps, and to model and estimate the

supply side of the market.

By modelling the supply side of the market, I can explore additional counterfactuals. For example, if search frictions reduce incentives to invest in app quality, it may be in the public interest to incentivize developers to create more high quality apps. This can be done by reducing fixed costs in such a way as to incentivize the entry of more higher quality products. Conversely, this counterfactual can show how policies that would raise fixed/entry costs would affect the market.

5.7 Static Entry Model

As described in Section 2.2, key strategic choices by developers include app characteristics and quality (how much to invest in an app), whether to enter the app into the market, whether to make the app free or paid, and what (mutually exclusive) category to enter the app into. For example, a developer who develops a non-game app can technically enter it in 24 potential categories. These categories appeal to potential entrants based on the number of other firms that the app expects to compete with in the category. To model firm entry decisions, I use a static entry model under incomplete information.⁶⁷ I will not model the choices of firms to produce free or paid products. I will focus only on free games, using the demand model from Columns (1) and (2) in Table 7.⁶⁸

In each period, there are NP_t potential free game apps choosing whether to enter the market. Consider the developer of game j , who chooses whether to invest in quality $w \in \{1, 2, \dots, W\}$, as well as entry into categories $c \in \{1, 2, \dots, C\}$ in period t . I define quality via the δ measure recovered from the demand model, which is an estimated weighted index of various app characteristics.⁶⁹ Apps are subdivided into groups based on the distribution of δ - low quality apps with the bottom 33% of δ values, medium quality apps with the middle 33% of values, high quality apps with the top 33% of values, and very high quality apps with the top 1% of δ .

I will call the entry variable $a_{jcw t}$, which is equal to 0 or 1 if the firm chooses to enter an app of type w into category c at time t . The profits of an app j of type w from entering into category c at time t can be expressed as follows:

⁶⁷I assume throughout that firms enter each app independently of the other apps that they own.

⁶⁸Recall that the low price elasticity estimates from the demand model suggest that a majority of the firms have negative marginal costs.

⁶⁹This is a simplifying assumption. An alternative specification would define groups around different interactions of product characteristics: e.g., apps which are less than 1 MB but have an average rating of 4 stars, and apps which are 1 to 10 MB and have an average rating of 3.5 stars. However, this form of grouping quickly increases the state space and generates a dimensionality problem.

$$\pi_{jcwt} = \theta_1 \hat{q}(w, \mathbf{w}_{-jt}, \mathbf{a}_{-jt}) + \theta_2 \hat{q}(w, \mathbf{w}_{-jt}, \mathbf{a}_{-jt})^2 - FC_w - \epsilon_{jcwt} \quad (21)$$

where \hat{q} are the predicted downloads from the demand estimates, and the θ s are parameter which reflect how the downloads are converted into profits.⁷⁰ I use a quadratic specification to reflect the institutional background of revenues in this industry. As explained in Section 2, advertising based revenue reflects the bidding of advertisers for slots in different apps. Varian (2007) shows that advertiser bids are convex in the desirability of the placement. This means that top apps with the largest per-period download volumes should attract the highest ad revenues, which can be exponentially larger than the ad revenues of less downloaded apps.⁷¹ I therefore expect the parameter θ_2 to be positive.

The downloads are a function of the entry decisions of all other firms (\mathbf{a}_{-jt}), as well as the quality of the app (w) and the quality of all other apps that enter into the category (\mathbf{w}_{-jt}). As mentioned above, the mapping of the entry and quality decisions into downloads is done via the demand model estimated in the previous section. FC_w is a parameter representing the fixed costs of entry for an app of quality w , and ϵ_{jcwt} is an idiosyncratic cost shock, assumed to be drawn iid from an Extreme Value Type 1 distribution. I assume that the shock of each firm is known to that firm, but unknown to other firms or to the econometrician (private information).

I normalize the profits of an app from staying out of the market to zero. Since I only focus on free firms in this analysis, this outside option for the potential free firms also includes entering into the paid market.

The number of firms in the market is correlated with the structural error term. A high idiosyncratic draw for a particular market for a particular firm (high ϵ_{jcwt}) could be due to some unobservable (to the econometrician) characteristics in market c that would encourage other firms to enter as well. Then, markets with high draws of ϵ would also have more competitors - and the number of competitors would be correlated with the error term. In a game of *incomplete* information with iid draws, however, the firm's information set is by assumption closer to the econometrician's information set.

Due to the presence of the idiosyncratic information shock ϵ , firm j 's decision is unknown to other firms ($-j$). The other firms do know the distribution of the

⁷⁰For paid apps, there is no need for this parameter as there is a standard price setting game (after entry) which results in some markups above estimated marginal costs.

⁷¹Downloads only partially determine the value of apps to advertisers. The other big component is the engagement of downloading consumers with the apps they downloaded. They are likely correlated - apps with more downloads are those that consumers engage with more. With observational data at the app level, I can only focus on the downloads.

cost shock G_ϵ . A firm k can then form expectations about the behaviour of firm j by integrating its' decision over the distribution of private information. These expectations are in the form of a probability $P_{jcw t}$ that firm j will enter category c at time t as type w .

I can then re-write firm j 's profit function as follows:

$$\pi_{jcw t}^e(\mathbf{P}_{-j t}) - \epsilon_{jcw t} \quad (22)$$

where $\pi_{jcw t}^e$ are the expected profits of firm j , which are a function of other firms' choice probabilities ($\mathbf{P}_{-j t}$).⁷²

From the perspective of firm i , firm j chooses to enter a category conditional on their beliefs about the entry probabilities of other firms in the market. Their best response is to enter category $cw t$ ($a_{jcw t} = 1$) if:

$$\pi_{jcw t}^e(\mathbf{P}_{-j t}) - \epsilon_{jcw t} > \max_{n \in \{1, \dots, C\}, w' \in \{1, \dots, W\}} \pi_{jn w' t}^e(\mathbf{P}_{-j t}) - \epsilon_{jn w' t} \quad (23)$$

Under the EV Type 1 distribution of the ϵ , the best response probability function of firm j is:

$$Pr(a_{jcw t} = 1) = \frac{\exp(\pi_{jcw t}^e(\mathbf{P}_{-j t}))}{1 + \sum_{w' \in \{1, \dots, W\}} \sum_{n \in \{1, \dots, C\}} \exp(\pi_{jn w' t}^e(\mathbf{P}_{-j t}))} \quad (24)$$

⁷²Expected profits can be expressed as:

$$\pi_{jcw t}^e = \theta_1 E[\hat{q}(w, \mathbf{w}_{-j t}, \mathbf{a}_{-j t})] + \theta_2 E[\hat{q}(w, \mathbf{w}_{-j t}, \mathbf{a}_{-j t})^2] - FC_w$$

where $E[\hat{q}]$ are the *expected* sales in category c at time t . The expectation is conditional on the exogenous variables including market characteristics. The expected number of downloads in a market is a function of an expectation about the number of products. It is possible to show that the expected downloads of a firm (as a function of the number of competitors) is the following binomial expression:

$$E[\hat{q}(w, N_{ct})] = \sum_{n=1}^{NP} \binom{NP}{n} (\hat{q}(w, n)) (P_{cw t})^n (1 - P_{cw t})^{NP-n}$$

where $P_{cw t}$ is the probability that a given firm enters category c as type w at time t . I can abstract from the individual firm subscript, since all firms are ex-ante identical aside from their idiosyncratic fixed costs. NP is the total number of potential entrants in the market. This can be a very large number (e.g., 2 times the maximum number of observed firms). Since the number of potential entrants is very large, and for computational simplicity, I approximate expected sales as follows: $E[\hat{q}(w, N_{ct})] = \hat{q}(\mathbf{w}, E[N_{ct}]) = \hat{q}(\mathbf{w}, P_{cw t} NP)$. That is, I approximate the expectation of a function as a function of the expectation. Although Jensen's inequality applies, preliminary simulations suggest that as the number of products increases, the approximation is close.

This is an incomplete information Bayes-Nash game where each firm has a belief about the probability that other firms enter the market and make their own decisions based on the probability. A Bayesian Nash Equilibrium (BNE) can be defined as a vector of choice probabilities (P) that solves the following fixed point problem:

$$P_{jcw t} = \frac{\exp(\pi_{jcw t}^e(\mathbf{P}_{-jt}))}{1 + \sum_{w' \in \{1, \dots, W\}} \sum_{n \in \{1, \dots, C\}} \exp(\pi_{jnw't}^e(\mathbf{P}_{-jt}))} \quad (25)$$

In the equilibrium, the beliefs of firms about their opponents' entry probabilities are the opponents' best responses to their own beliefs. The beliefs are therefore consistent with equilibrium actions.

5.8 Entry Model Estimation

The likelihood function which describes the probability of observing the actual entry pattern in the data is:

$$\mathcal{L} = \prod_j \prod_c \prod_w \prod_t [P_{jcw t}] \quad (26)$$

The idiosyncratic error component is the only difference across firms trying to enter the same market. Therefore, the probabilities of entry for two firms in the same market are the same: $P_{jcw t} = P_{icw t}$. I can then generalize the probability for a given market as P_{cwt} , and rewrite the likelihood function as follows:

$$\mathcal{L} = \prod_c \prod_w \prod_t [P_{cwt}^{N_{apps_{cwt}}}] \quad (27)$$

where $N_{apps_{cwt}}$ is the actual number of apps of type w in category c that enter in time t . Taking the natural logarithm of that expression gives me the log likelihood function:

$$\ln(\mathcal{L}) = \sum_c \sum_w \sum_t [N_{apps_{cwt}} \ln(P_{cwt})] \quad (28)$$

The probability of entry is a function of the parameter vector as well as of all of the other optimal predicted entry probabilities (\hat{P}_{cwt}). By taking first order conditions with respect to the parameter vector, it is possible to obtain parameter estimates that maximize the probability of obtaining the observed data.

The model can be estimated using an NFXP algorithm. However, it is probable that there are multiple equilibria in the data, meaning that the likelihood function is in fact a likelihood correspondence, and the estimated set of parameters is only

one of a number of possible parameter values in the model. As is standard in the literature (Sweeting 2009, Bajari, Hong, Krainer and Nekipelov 2010), I assume that only one equilibrium is played in the data, and estimate the model using a two step algorithm following Hotz and Miller (1993).⁷³ The first step generates an estimate of the choice probabilities. Since I do not have characteristics at this stage, I can generate the choice probabilities using simple count data (the average share of apps of a particular type that enter into a category). The second step uses the estimates of the CCPs to back out the parameters from the log likelihood function.

Table 10 presents supply side parameter estimates. The fixed cost coefficients are monotonically increasing in the quality of the app, as expected. While the fixed costs of producing a low and medium quality app are pretty close, the fixed costs of producing a very high quality app are roughly double the fixed cost of a low quality app. A timing test suggests that these parameters do not significantly change over time.

5.9 Consumer Welfare Decomposition and Counterfactuals

The supply model allows for a more complete decomposition of changes in consumer welfare than the decomposition in Table 9. First, it is possible to fully decompose changes in search costs from the change in product variety.⁷⁴ Second, I can additionally decompose the welfare effects of the change in product variety into the welfare effects from the increase in product variety *while holding the quality distribution constant*, and the welfare effects from the change in the quality distribution.

I compute the counterfactuals by simulating the demand and the supply model under different parameter values. These simulations then produce an equilibrium number of entrants and an equilibrium quality distribution. For example, the “baseline” equilibrium is a market where the search costs of consumers are high (as estimated in Column 1 of Table 7). These search costs result in an equilibrium number of competitors across the available categories and an equilibrium quality distribution. It is then possible to compute the welfare of consumers in this market. I can also generate a “post-split” equilibrium where consumers have low search costs (as estimated in Column 2 of Table 7). These demand conditions result in a different

⁷³de Paula and Tang (2012) and Aguirregabiria and Mira (2015) are among recent papers that drop the “single equilibrium” assumption. However, dropping this assumption forces the econometrician to make distributional assumptions about the potential equilibria in the model. This method would be highly computationally intensive, considering the large number of firms in the sample and the large number of potential equilibria.

⁷⁴Recall, Table 9 includes some measure of changing search costs in the product variety effect.

Table 10: **Supply Parameter Estimates**

	(1)
Downloads (θ_1)	-0.008*** (0.000)
Downloads Squared (θ_2)	1.7×10^{-6} *** (1.61×10^{-7})
Low Quality App FC	2.9*** (0.24)
Medium Quality App FC	4.0*** (0.25)
High Quality App FC	4.3*** (0.25)
Very High Quality App FC	5.9*** (0.27)
Observations (months \times categories \times sizes)	1,302

Robust standard errors in parentheses

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

number of equilibrium entrants in the available categories and a different distribution of the type of apps that enter into the categories.

For the counterfactuals looking at the welfare effects of the changes in the quality distribution, I define the share of each quality group of the total number of apps in the “baseline” equilibrium as the “baseline” distribution of quality. The share of each quality group of the total number of apps in the “post-split” equilibrium is the new distribution of quality. I can then calculate consumer welfare in a market where the total number of apps is the same as in the “post-split” equilibrium, but the quality distribution is the same as in the “baseline” equilibrium.

I also estimate two counterfactuals which change entry incentives for firms without changing search costs. Reports suggest that the mobile app economy created over a million jobs in the US.⁷⁵ It may then be in the public interest to incentivize developers to create more high quality apps. The first counterfactual generates a similar level of additional entry as the search cost reduction through a fixed/entry

⁷⁵<http://www.asymco.com/2015/01/22/bigger-than-hollywood/>

cost reduction of approximately 15% for all app types. The second counterfactual shows how a larger reduction of 30% in entry/fixed costs for all product types affects the market.

On the supply side, I start each simulation of the entry probabilities from the estimated entry probabilities in the data and allow the model to converge to the new equilibrium probabilities. This means that I am looking for a stable equilibrium that is around the equilibrium in the data. There may be multiple counterfactual equilibria for any given set of demand conditions. I do not formally test for the presence of multiple equilibria. I have iterated on a grid of starting probabilities that is centred around the probabilities in the data and have only managed to find single equilibria for every set of demand conditions.

As mentioned above, each firm only takes into account other firms' entry decisions in the free market and not the paid market. The relative separation of the two markets (with different best-seller lists) makes this a reasonable assumption. There may still be strategic linkages between the two parts of the app store - firms considering entering into the free markets may still care about the number of paid apps in the same category (and vice versa). In that sense, the counterfactual equilibria that I find are holding the actions of paid apps constant. Similarly, the consumer welfare changes I examine only reflect the changes in welfare that consumers obtain from the free market.

Table 11 shows the consumer surplus change decompositions. It is analogous to the welfare decompositions from Column (1) of Table 9.⁷⁶

Table 11: **Estimated Changes in Consumer Surplus**

	(1) Search Cost Policy	(2) -15% Fixed Cost Policy	(3) -30% Fixed Cost Policy
Pre-Split CS	100%	100%	100%
Increase in Variety (Pre-Split Quality Dist.)	+3%	+3%	+3.5%
Change in Quality Distribution	-1.5%	+1%	+2%
Increase in Number of Categories	+14%		
Reduction in Marginal Search Costs	+50%		
Overall Increase in CS	+65.5%	+4%	+5.5%

⁷⁶To evaluate the other model specifications, I have to model changes not just in the app quality distribution, but also consider changes in prices and in lagged-downloads. To do that, I need price estimates that do not imply negative marginal costs.

The results in Column (1) show that the increase in product variety had a small positive effect on consumer surplus - increasing it by 3% from the baseline equilibrium. This is an order of magnitude smaller than the “variety” effect in Table 8, which suggests that changes in search costs generated most welfare effects in the model. As before, there is a large effect from the reduction in marginal search costs for consumers. There is also a reduction in total search costs that comes from the increase in the number of categories, and that also increases welfare.⁷⁷ Again, overall, the net impact of the change in the number of game categories on consumer surplus is strongly positive.

The reduced form results raise a question about the relative welfare effects of changes in product variety and product quality. Counterfactual simulations using the structural model show that while the increase in the share of low quality products reduces consumer surplus by 1.5% relative to its pre-equilibrium baseline, it is smaller than the effect of the increase in product variety.

A useful comparison is with the second counterfactual (in Column 2), where the fixed costs of product development (or entry) fall by 15%. Here, the net increase in consumer welfare is much smaller - about 4% relative to the baseline. The contribution of the increase in variety - 3% - is the same as in Column (1), which makes sense since this counterfactual is designed to generate the same level of entry as in the search cost counterfactual (see Table 12 below). However, unlike Column (1), the contribution of the change in quality distribution is positive, an increase of about 1% relative to the pre-split consumer surplus. This occurs since the reduction in fixed costs incentivizes additional more high quality apps to enter the market.

The reason why the welfare change in Column (2) is small is because search costs do not change. This is important, since the number of products almost doubles. If the number of categories increase, the number of products per category in the search cost counterfactual mostly fall. This reduces the search costs that consumers expend per product (ψ_j in the demand model). In the fixed cost counterfactual, these search costs increase, which offsets the effect of larger product variety.

Column (3) in Table 11 shows the final counterfactual - reducing entry costs by 30%. This counterfactual results in higher consumer welfare - the overall gains are 5.5% in Column (3), as opposed to 4% in Column (2). The gains from variety do not increase substantially. Column (5) in Table 12 suggests that this may be in part because the share of very high quality apps does not increase by much. In addition,

⁷⁷It is not obvious how to think of the change in the number of categories as being separate from the change in marginal consumer search costs in this setting. The two effects can be lumped into a single change that can be called the change in search costs, and that constitutes 64% of the change from the baseline.

the number of products doubles relative to the baseline, but this also has a negative effect on welfare through search costs. If the number of products double, consumers find it harder to find certain products, and the gains from variety are relatively limited.

These counterfactuals are meaningful, despite their smaller welfare effects. There are two reasons for that: First, comparing Column (2) or Column (4) in Table 9 to Column (1) in Table 11 suggests that the effects of changes in marginal search costs may be overstated in Table 11. Second, there may be decreasing returns to additional changes in consumer search costs - consumer surplus will likely not increase by an additional 60% from tripling the number of categories again. At some point, browsing through a large number of categories may make it *harder* for consumers to find the products they are looking for. Consider the extreme case where each app is its own category, for example. Overall, the positive welfare effects of reducing fixed costs suggest that the two policies may be complementary - a reduction in search costs which incentivizes higher quality firms to enter the market can mitigate the effects of a reduction in search costs which allows smaller and lower quality firms to enter the market.

Table 12: **Equilibrium App Distribution**

	(1) Baseline	(2) Post-Split	(3) Post-Split (Orig. Dist)	(4) 15% Reduction Entry Cost	(5) 30% Reduction Entry Cost
Share of Low Quality Apps	0.64	0.75	0.64	0.59	0.54
Share of Medium Quality Apps	0.20	0.15	0.20	0.22	0.24
Share of High Quality Apps	0.15	0.08	0.15	0.17	0.20
Share of V. High Quality Apps	0.01	0.01	0.01	0.01	0.02
Total Number of Apps	111,261	162,472	162,472	157,582	206,382

Table 12 shows the distribution of app quality together with the total number of apps that enter the market in equilibrium under the baseline and under each counterfactual. The total number of apps in Columns (1) and (2) allow me to test the model by comparing the size of the treatment effect in the actual data and the change in the number of apps in the simulation. The size of the average treatment effect in the data is approximately 33%, with a standard error of 7%. The percent change in the number of apps in the simulation is approximately 46%, which is within the 95% confidence interval for the treatment effect in the data. This suggests that the static demand model and the static entry model are a reasonable approximation.

6 Conclusion

This paper presents evidence regarding the effect of consumer search costs on competition, entry, quality, and product design in online markets, using new detailed data from the Google Play mobile app store. I take advantage of an exogenous shock whereby consumer search improved for some product types (games) in the Google Play store but not for others (non-games).

The data shows that lower search costs increase product entry and that niche products drive most of the entry effects. This is consistent with findings from IO theory (e.g., Bar-Isaac et al 2012), and it is the first clear evidence on this topic in the empirical literature. I also find that the average quality of the affected products falls relative to the unaffected products following the split. This is the first direct empirical evidence of the effect of changes in consumer search costs on quality.

I set up a structural demand model which estimates consumer choice coefficients related to utility and to search in order to measure the overall welfare effects, and the contribution of different factors to welfare. I find that, consistently with the reduced form results, the search costs associated with game apps declined after the split in the game categories. I also set up an entry model with an incomplete information framework where apps choose to enter into different categories and invest in quality based on expected profits (and downloads).

I show that the re-categorization of games increased consumer welfare in the mobile app market; a decomposition of the welfare changes shows that most of the increase came from a reduction in marginal search costs. There is also an increase in welfare due to the increase in greater product variety, and a smaller fall in welfare due to lower product quality. Further counterfactual simulations - reducing fixed costs - show that welfare gains are modest without changing search costs.

The main policy concern related to search is not reducing search costs for consumers, but rather increasing search costs.⁷⁸ The empirical evidence in this paper suggests that there are two consumer welfare implications to policies that *increase* search costs online: First, consumer welfare falls because marginal search costs increase. This is the direct negative effect of increasing search cost. Second, consumer welfare falls because of foreclosure; firms that would have entered the market in the presence of low search costs now choose not to enter the market, which reduces the product variety available to consumers. Overall, the results of this paper suggest that policies which increase search costs in online markets may have strongly detrimental effects to consumer welfare.

For mobile app stores, the incentives of the platform and welfare incentives are

⁷⁸<http://www.wsj.com/articles/inside-the-u-s-antitrust-probe-of-google-1426793274>

aligned. App stores should want to reduce consumer search costs in order to stimulate search, entry and downloads. However, other online platforms may not have the same incentives. The EUR2.4 billion fine to Google in the EU provides an interesting case.⁷⁹ Google's policy in the search engine market (e.g., hotels, flights) was judged to have increased consumer search costs. This clearly benefitted Google, but the harm was not limited to existing competing platforms - e.g., Expedia, or Kayak. There was also potential consumer harm due to the foreclosure of future competitors from the market.⁸⁰

⁷⁹<https://www.theverge.com/2017/6/27/15872354/google-eu-fine-antitrust-shopping>

⁸⁰There are also potential benefits to Google's search engine policies that my model does not capture. For example, the quality of Google's products is arguably higher than their competitors' qualities. Thus, by presenting the highest quality product first, Google minimizes the total search costs consumers pay.

7 Appendix A: List of Categories

Table 13: Google Play Game Categories Before and After March 2014

Before	After
Casual	Casual
Racing	Racing
Sports	Sports
Arcade & Action	Action
	Adventure
	Arcade
	Board
Cards & Casino	Card
	Casino
	Education
	Family
	Music
	Live Wallpaper
Brain & Puzzle	Puzzle
	Role Playing
	Simulation
	Strategy
	Trivia
	Word

Table 14: Google Play Non-Game Categories

Books & Reference	Libraries & Demo	Productivity
Business	Lifestyle	Shopping
Comics	Media & Video	Social
Communications	Medical	Sports
Education	Music & Audio	Tools
Entertainment	News & Magazines	Transportation
Finance	Personalization	Travel & Local
Health & Fitness	Photography	Weather

8 Appendix B

8.1 Download Ranges

Table 15: List of Cumulative Download Ranges

Lower Bound	Upper Bound
1	5
5	10
10	50
50	100
100	500
500	1,000
1,000	5,000
5,000	10,000
10,000	50,000
50,000	100,000
100,000	500,000
500,000	1 million
1 million	5 million
5 million	10 million
10 million	50 million
50 million	100 million
100 million	500 million
500 million	1 billion

8.2 Summary Statistics of New Apps

Table 16: Summary Stats of New Apps at Weekly Level

Variable	Mean	Std. Dev.	Min	Max	N Obs
Games					
Download Lower Bound	27,802	245,351	0	1 million	10,863
Rank	256	147	1	500	10,863
Non-Games					
Download Lower Bound	13,982	414,378	0	5 million	17,375
Rank	278	140	1	500	17,375

8.3 Regression Results Predicting Downloads for New Apps

Table 17: Regression Results on $\ln(\text{Downloads})$

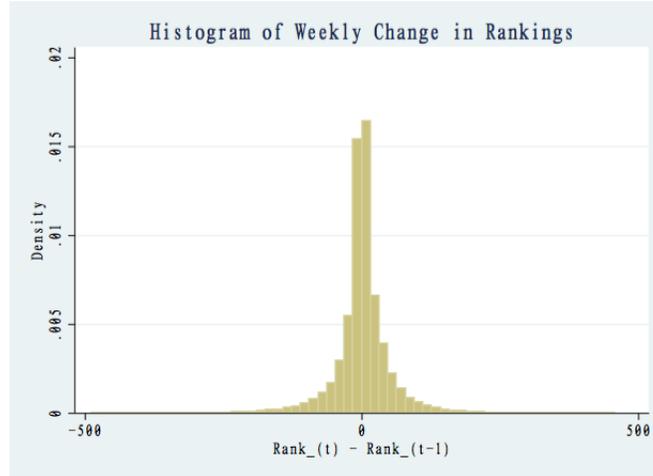
	(1) Games	(2) Non-Games
$\ln(\text{Rank})$	-1.168*** (0.079)	-0.926*** (0.055)
Month FE	YES	YES
Category FE	YES	YES
$\ln(\text{Rank})$ interactions with Month FE	YES	YES
Observations	10,020	16,738
R-squared	0.490	0.530

Robust standard errors in parentheses

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

8.4 Weekly Changes in App Rankings

Figure 14



9 Appendix C: Additional Reduced Form Results

9.1 Weekly Game and Non-Game App Entry

9.2 Test of the Timing of the Entry Treatment Effect

With monthly data I can test whether the treatment effect indeed occurred in March 2014 rather than in an earlier period (say, in mid 2013) but is still absorbed by the treatment dummy. To do that, I allow the treatment effect to vary by month by introducing interactions between monthly date dummies and the treatment dummy, effectively estimating a potential treatment effect for every month (relative to some early baseline period). The estimating equation then changes slightly to become:

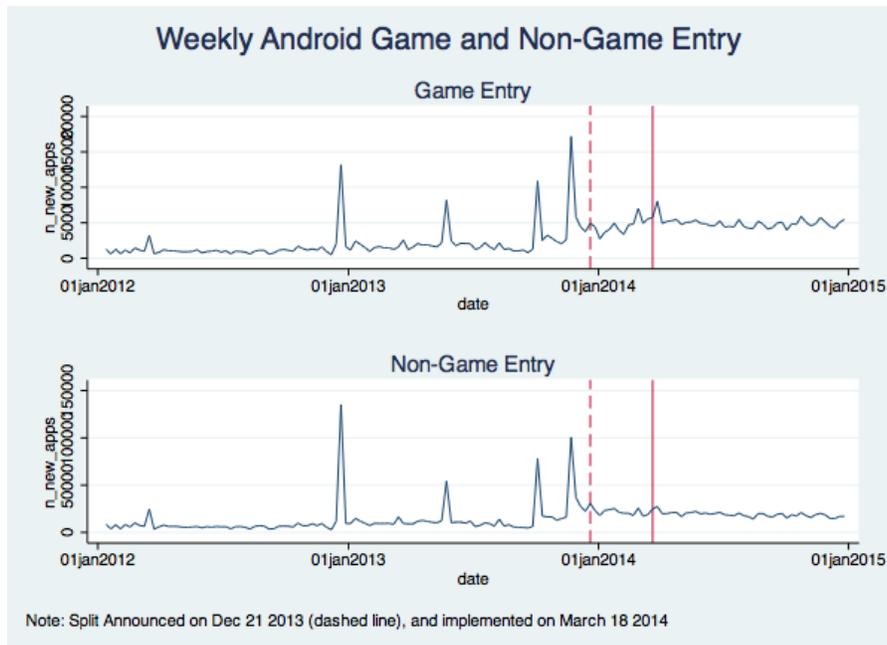
$$Y_{ct} = \sum_t \tau_t (Game_c \times \delta_t) + \delta_c + \delta_t + \epsilon_{ct} \quad (29)$$

where the τ_t s now capture period specific treatment effects, relative to a baseline period - March 2012, the first period of the sample.⁸¹

Figure 8 above shows the period specific treatment effects. It suggests that the main treatment effect hits exactly after the increase in the number of categories. There are two periods of pre-split positive effects (one of those statistically significant at the 95 percent level), but these can be explained since the split was announced in

⁸¹I can use different baselines, which give similar results.

Figure 15

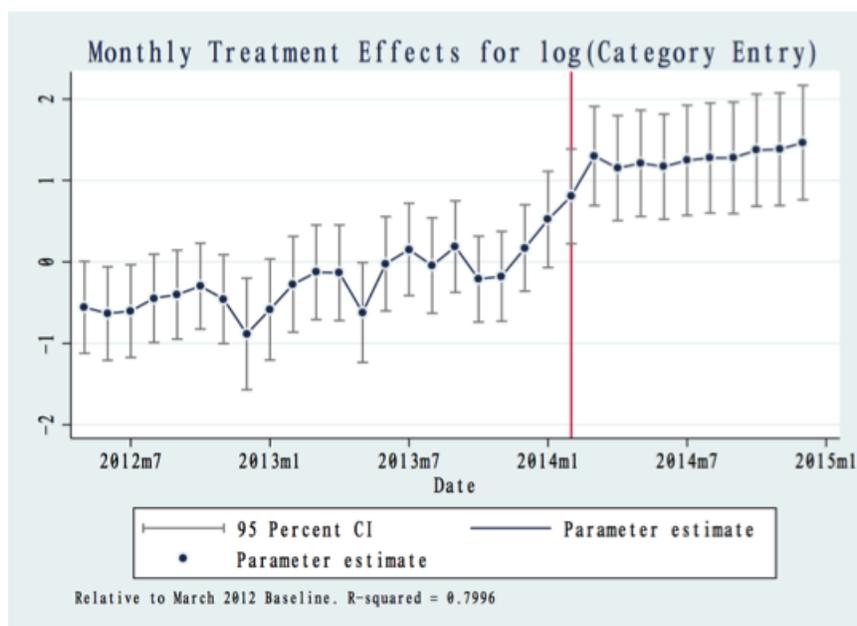


December 2013, meaning that some firms may have entered the market early to be better positioned for when the split takes place.

9.3 Category Specific Entry Treatment Coefficient Estimates

9.4 Category Specific Size Treatment Coefficient Estimates

Figure 16



9.5 Placebo Tests for Quality

While the reduced form results are suggestive, some of the category specific results, particularly those that rely on category size, could be driven by mean reversion or the size of the categories rather than the treatment effect. To disprove this hypothesis I run a placebo test.

I split my three year sample into two - a subsample of observations for January 2012 to December 2013, and a subsample of observations from January 2013 to December 2014. The split in the categories and the reduction in search costs happens during the second subsample (in March 2014), although the difference in size between different Game app categories are persistent throughout the two subsamples.

Consequently, I run a category specific regression (analogous Column 2 in Tables 4 and 5) for both subsample. In the 2012/3 subsample, I set up a placebo treatment to start in March 2013 (exactly one year before the true treatment starts). I expect that the placebo treatment will generate zero treatment effects, whereas the true treatment in the second part of the sample should generate non-zero treatment effects.

The following table shows the regression results for the share of 1 star ratings, and for the mean size of the app for the two periods. Columns (1) and (3) show the

Table 18: **Category Specific Treatment Effects on log(N Entrants)**

Category Name	τ_g	Category Name	τ_g
Arcade	0.040 (0.106)	Educational	2.383*** (0.137)
Card	-0.405** (0.163)	Strategy	1.985*** (0.123)
Casual	0.011 (0.100)	Trivia	2.241*** (0.132)
Puzzle	-0.209* (0.113)	Word	1.474*** (0.132)
Action	2.061*** (0.112)	Adventure	2.702*** (0.143)
Board	1.883*** (0.116)	Family	2.923*** (0.168)
Casino	1.536*** (0.151)	Music	2.532*** (0.204)
Racing	0.395*** (0.117)	Role Playing	1.804*** (0.119)
Sports Games	0.010 (0.143)	Simulation	3.140*** (0.160)
Observations	1,469		
R-squared	0.829		

Robust standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

2012/3 regressions, whereas Columns (2) and (4) show the 2013/4 regressions.

Both placebo treatment effects in Columns (1) and (3) are not statistically significant. The placebo treatment for the 1 star rating regression is a full order of magnitude lower than the “true” treatment, and the placebo treatment for the mean size regression is going in the opposite direction of the “true” treatment effect (both are not statistically significant).

I also run the category specific treatment effects for the placebo period for the share of 1 star ratings and for the mean size of the apps. These results are pictured in the two figures below. For the placebo regression, the category specific treatment

Table 19: **Category Specific Treatment Effects on App Size (in MB)**

Category Name	τ_g	Category Name	τ_g
Arcade	-3.58*** (1.21)	Educational	-4.68*** (1.36)
Card	-0.825 (2.82)	Strategy	-6.18*** (2.0)
Casual	-3.02*** (0.853)	Trivia	-1.27 (1.75)
Puzzle	-1.55 (1.08)	Word	-4.66** (1.82)
Action	-27.2*** (5.4)	Adventure	-30.4*** (8.5)
Board	-4.84*** (1.52)	Family	-7.72*** (1.14)
Casino	-48.7*** (7.1)	Music	-2.67 (4.42)
Racing	-9.94*** (1.36)	Role Playing	-11.84** (4.4)
Sports Games	-3.8* (2.0)	Simulation	-6.22*** (1.66)
Observations	1,469		
R-squared	0.560		

Robust standard errors in parentheses

*** p<0.01, ** p<0.05, * p<0.1

effects are plotted against the mean number of apps in the category in 2012, and for the “true” regression, the category specific treatment effects are plotted against the mean number of apps in the category in 2013. The ordering of the categories does not change between 2012 and 2013, meaning that the relationship between the treatment coefficients and category size should be comparable between the two sets of regressions.

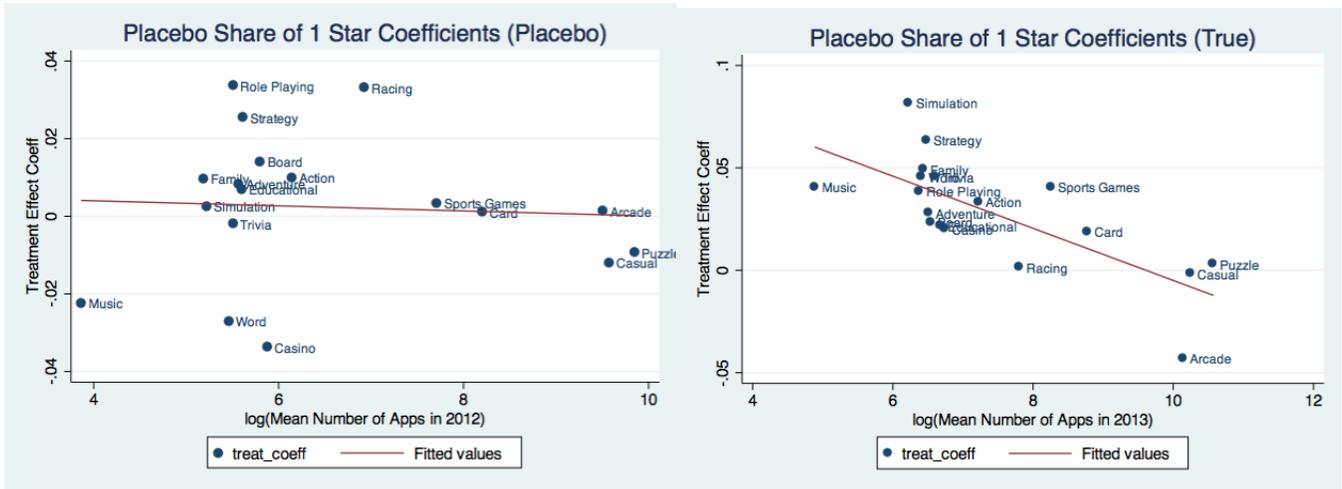
In the 1 star share regressions, the placebo category treatment coefficients have essentially a zero correlation with the size of the categories (in terms of the number of apps). This, together with their very small magnitude, suggests that the placebo

Table 20: **Placebo Test Regressions Results**

	Share of 1 Star Ratings		Mean App Size (MB)	
	(1) 2012/3	(2) 2013/4	(3) 2012/3	(4) 2013/4
Placebo Treatment	0.002 (0.004)		2.077 (1.780)	
True Treatment		0.028*** (0.003)		-5.562*** (1.022)
Time FE	YES	YES	YES	YES
Category FE	YES	YES	YES	YES
Observations	881	966	881	966
R-squared	0.350	0.468	0.602	0.646

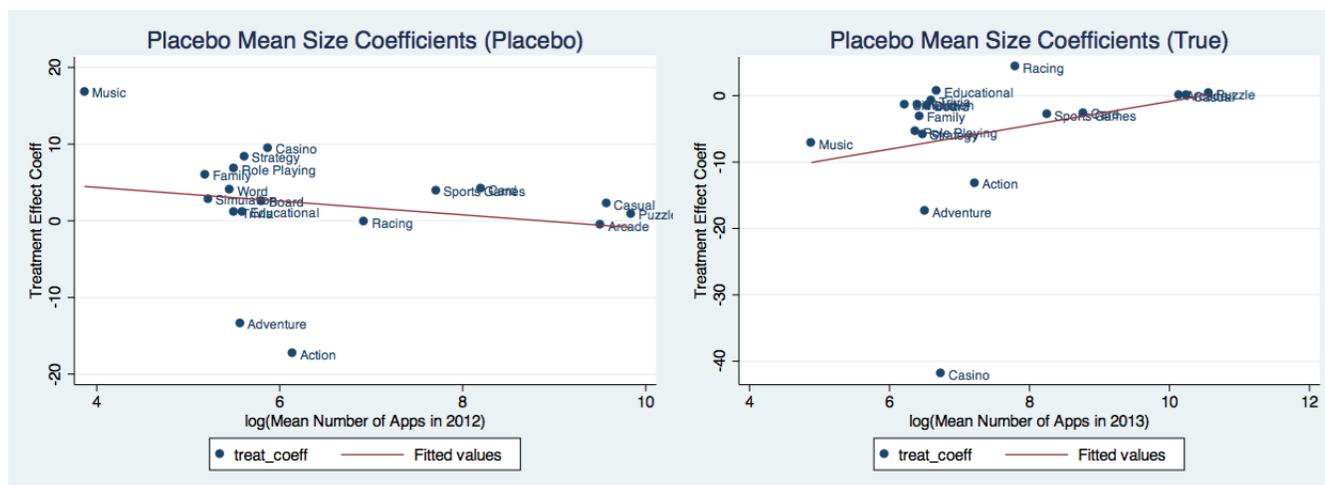
Robust standard errors in parentheses
 *** p<0.01, ** p<0.05, * p<0.1

Figure 17



treatment essentially had no effect on the share of 1 star reviews, as expected. The magnitudes are much larger (often twice as big) in the “true” regression, and there is a clearly negative correlation between category size and the treatment effects - as expected.

Figure 18



The mean size regressions are similar. In the placebo regressions, there is a weakly negative relationship between category size and the placebo treatment effects, which are mostly positive in magnitude and not statistically significant. This suggests that in the pre-split period, the smallest categories experienced a larger increase in mean size (increase in quality) relative to the larger categories. This is consistent with the story that apps in the small categories (or types) had to be “better” to be successful. By contrast, the relationship between the treatment coefficients and category size is positive in the “true” period, and the coefficients themselves are mostly negative for the small categories. Again, this is consistent with the story of the split in categories either reducing competition or otherwise reducing the incentives of developers in small categories to produce “high quality” apps.

10 Appendix D

10.1 Additional Monthly Coefficient Estimates

Figure 19

