Predicting Apps Installations from Google Play Store through Machine Learning Models

Yiheng An
UCLA Extension Data Science Intensive

Presentation Content

1. Background and Question
2. About the Data
   - Dataset
   - Data Description
3. Exploratory Data Analysis
   - Missing Values Management
   - Dependent Variable
   - Explanatory Variables
4. Initial Model Selection
   - Linear Regression Model
   - Ordered Logistic Regression
   - Other ML Models
5. Model Refinement
6. Key Takeaways

Background

With more than 2 billion active users, the Google Play platform has become one of the most attractive and competitive market of Android App development.

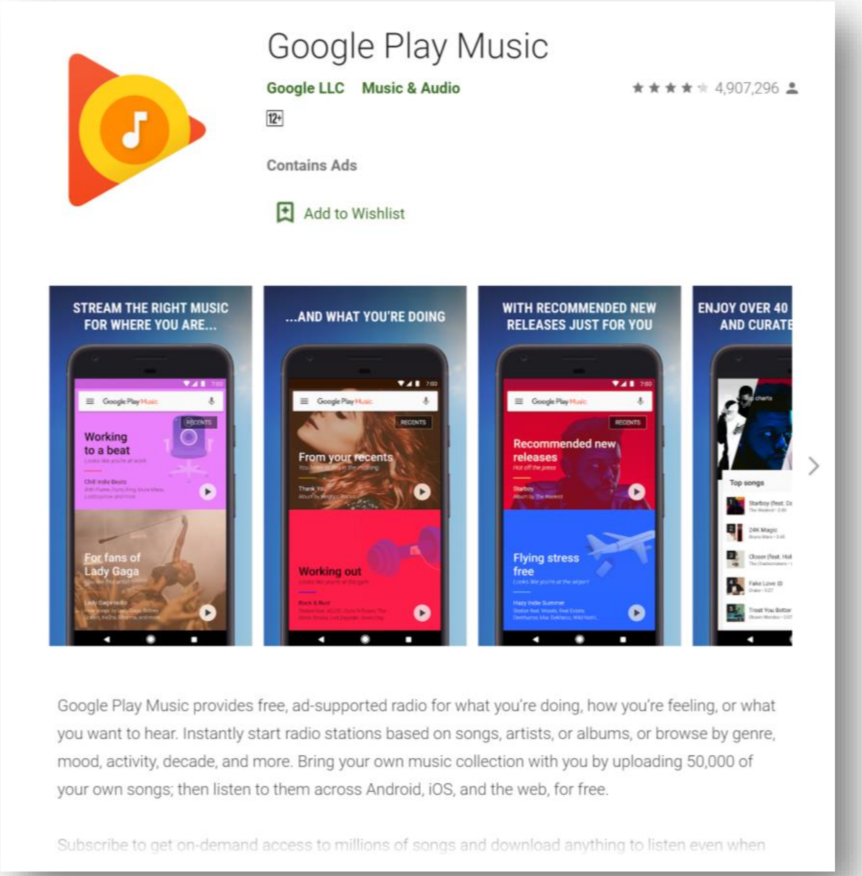
Millions of developers and data-driven businesses need actionable insights to capture their market strategically.

Data from Google Play Store has enormous potential to empower them to success.

Google Play Music will begin shutting down in September

The initial shutdown will affect some users, but it'll be off for all users in October

By Cameron Faulkner | @camfaulkner | Aug 4, 2020, 12:12pm EDT



Research Questions

What do the most popular Apps look like?

Can we build a ML model to predict how popular an App will be?

Dataset

Web scraped data of 10k Google Play Store Apps for exploring the Android market.

Source: https://www.kaggle.com/lava18/google-play-store-apps

Data Description

	Variable	Description	Type	Raw Data	Action
Dependent	Installs	the number of installations for each App	Categorical	0+, 1+, 10+, 100+, 1,000+ …	Convert to numeric variable
Identifier	App	English text intended to be the App’s name	Categorical	Character	Delete duplicates
Explanatory	Category	the category that the App falls in	Categorical	Character	---
	Rating	the users' rating from 0 to 5	Numerical	1.0 - 5.0	---
	Android.Ver	the Android version required for the App	Categorical	"4.1 and up", "4.0.3 and up"…	Convert to numeric variable (keep first 2 digits)

Data Description

	Variable	Description	Type	Raw Data	Action
Explanatory	Reviews	the number of review for the App	Numerical	0-78158306	---
	Translated_ Review	English Text intended to describe users' feeling	Categorical	Character	---
	Size	the size for a specific App	Categorical	994k, 1.0M, 1.1M…	Convert to numeric variable (unify the scale as MB)
	Type	If the App is paid	Categorical	Free, Paid	Convert to factor
	Price	the price needed to pay for the App	Categorical	$0.99, $1.00, $1.04, ...	Convert to numeric variable
	Content.Rating	one of six content rating category	Categorical	Everyone, Everyone 10+, Teen, Mature 17+ ...	Convert to factor
	Last.Updated	the date of the last update	Categorical	"27-Jun-18", "16-Jul-18"…	Convert to numeric variable (the days until 8/8/2018)

Exploratory Data Analysis: Roadmap

1. Missing Values Management

- Visualization
- Data Description
- Source Analysis
- Imputation Method
- Imputation Results

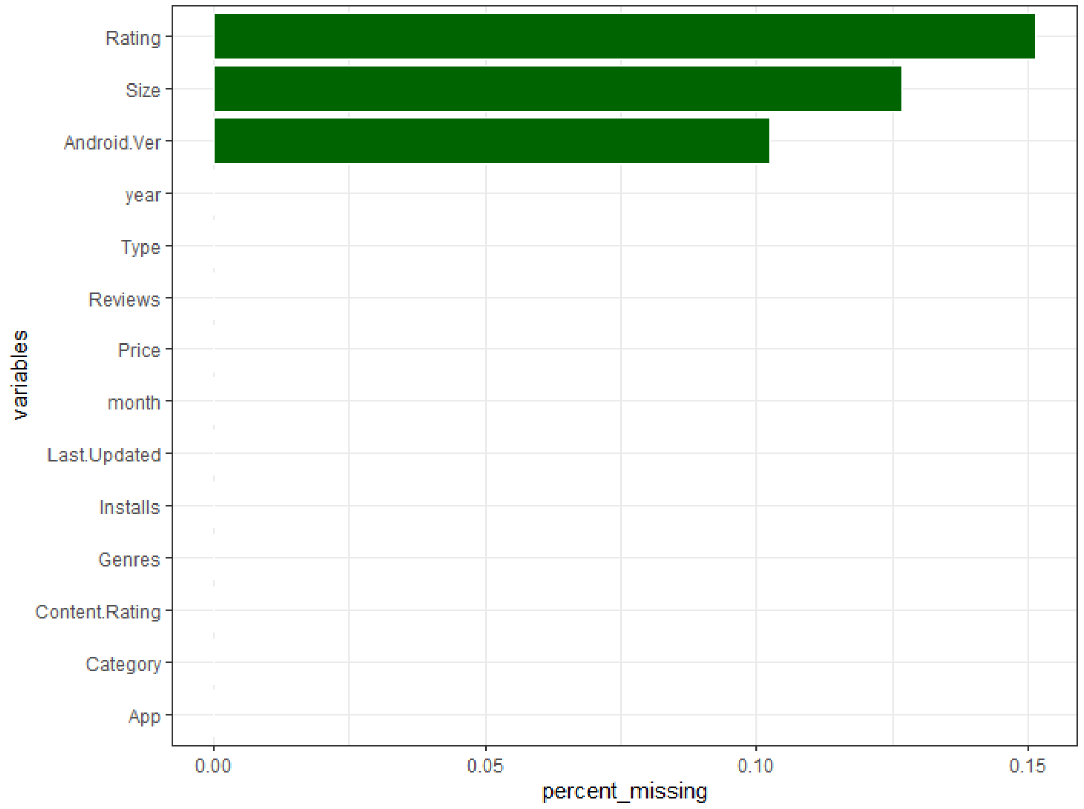
2. Dependent Variable

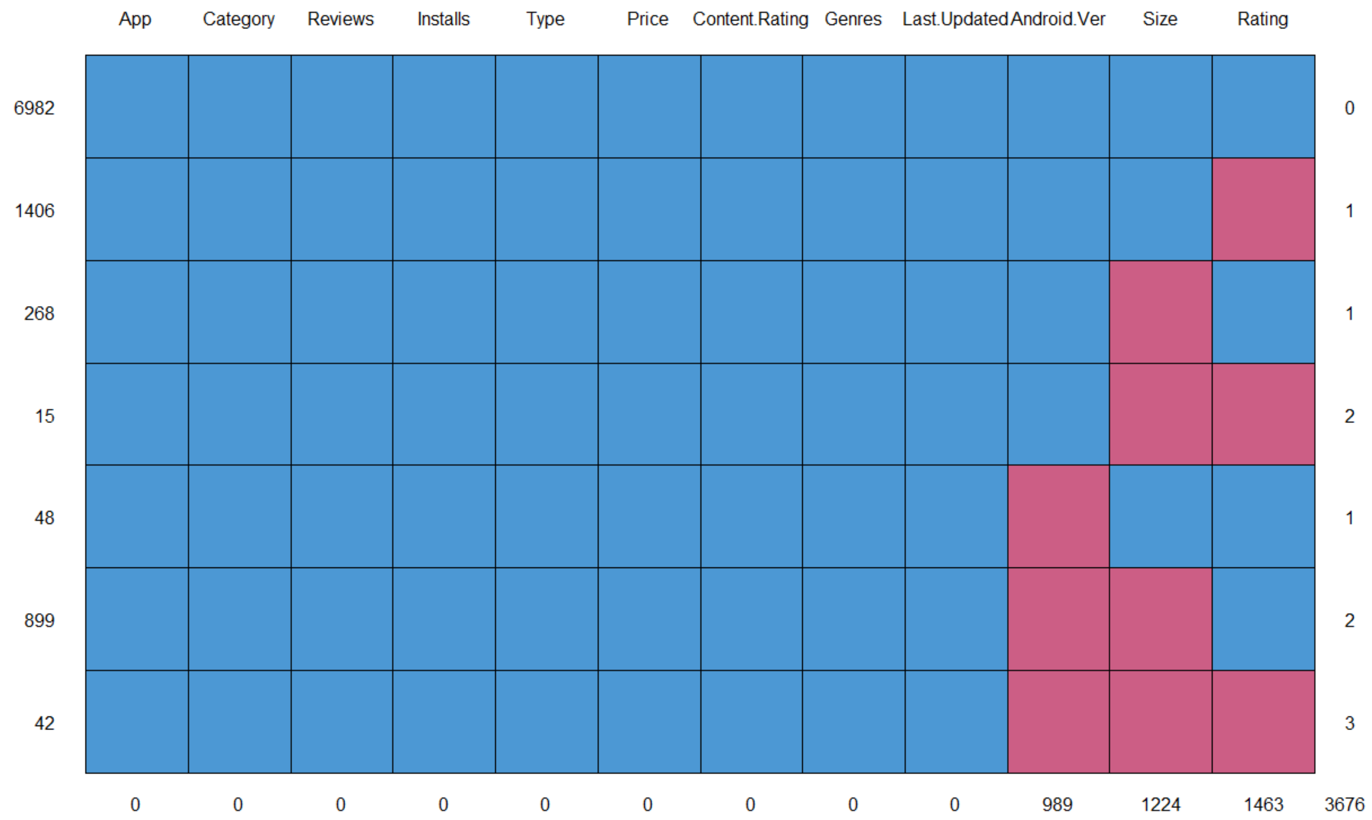
- Different Groups
- Distribution by other Variables

3. Explanatory Variables

- Categorical - Binary
- Categorical - Multi-valued
- Numerical
- Correlation Matrix

Missing Values Management: Visualization





Where do the NAs come from?

Missing Values Management: Source Analysis

Variable: ***Size***
NAs source: *"Varies with device"*

Variable: ***Android.Ver***
NAs source: *"Varies with device"*

Fix1: Dummy variables:

- ***Size.varies: 0 or 1***
- ***Android.varies: 0 or 1***

Fix2: Data Imputation:

- ***Keep potential predictors***
- ***Minimize the distortion in prediction***

Variable: ***Rating***
NAs source: *User behavior*

Neutral attitude? *(Impute 3.5)*

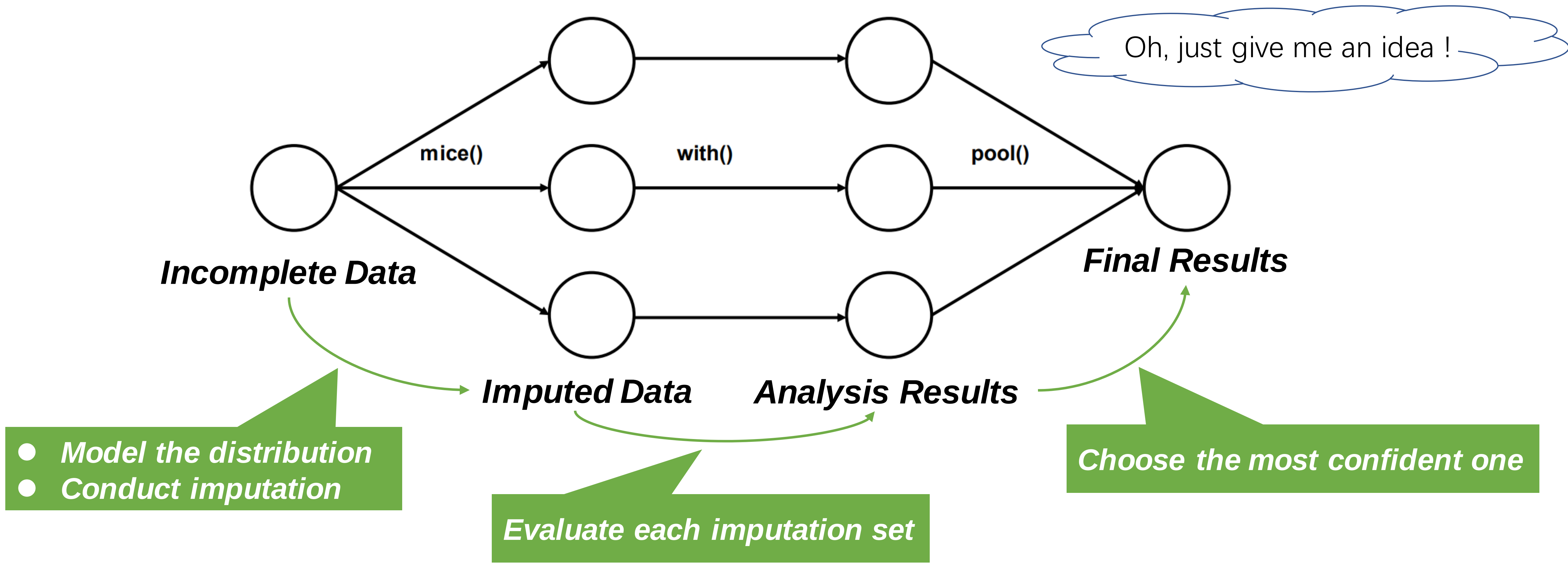
Drawbacks

- ***Totally change the distribution***
- ***Dismiss user habits***

We need a safe way!

Missing Values Management: Imputation Method

Multiple Imputation： Predictive Mean Matching



Takeaway:

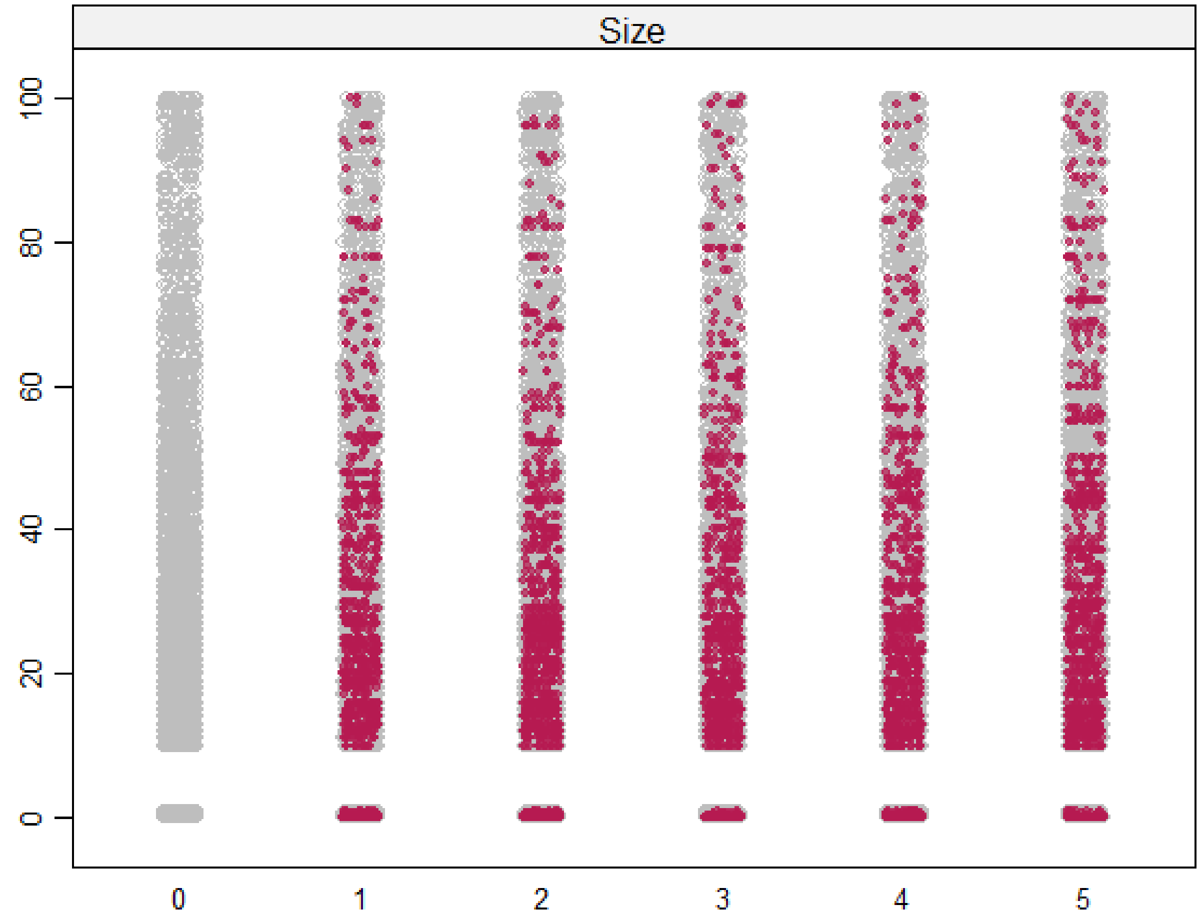
PMM allow us to impute data perfectly with the original distribution.

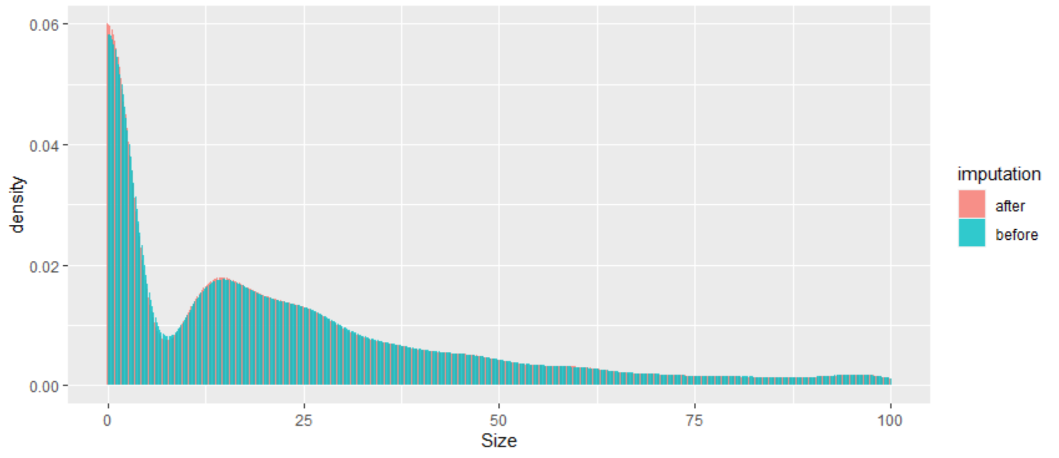
Refence: *Van Buuren S, Groothuisoudshoorn K. mice : Multivariate Imputation by Chained Equations in R[J]. Journal of Statistical Software, 2011, 45(3): 1-67*

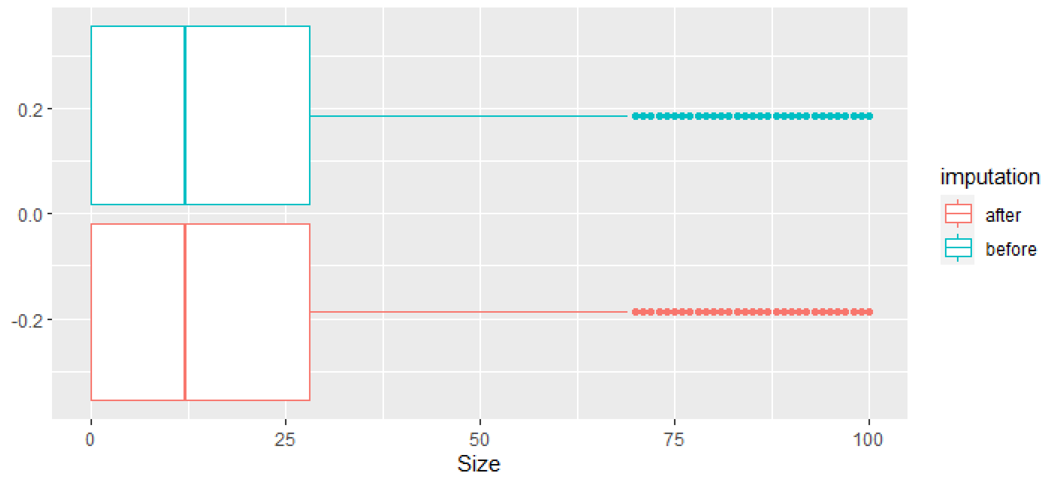
Missing Values Management: Imputation Results

Variable: ***Size***

NAs source: *"Varies with device"*



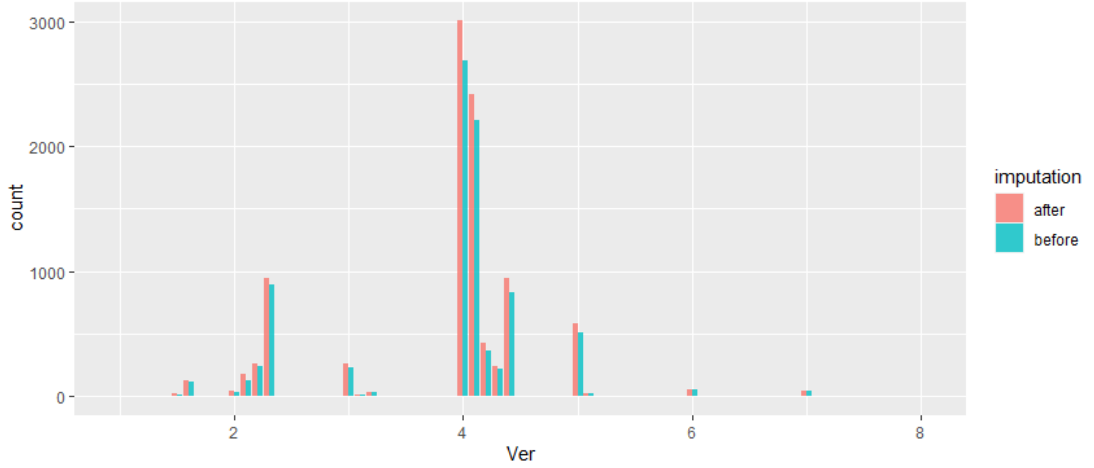


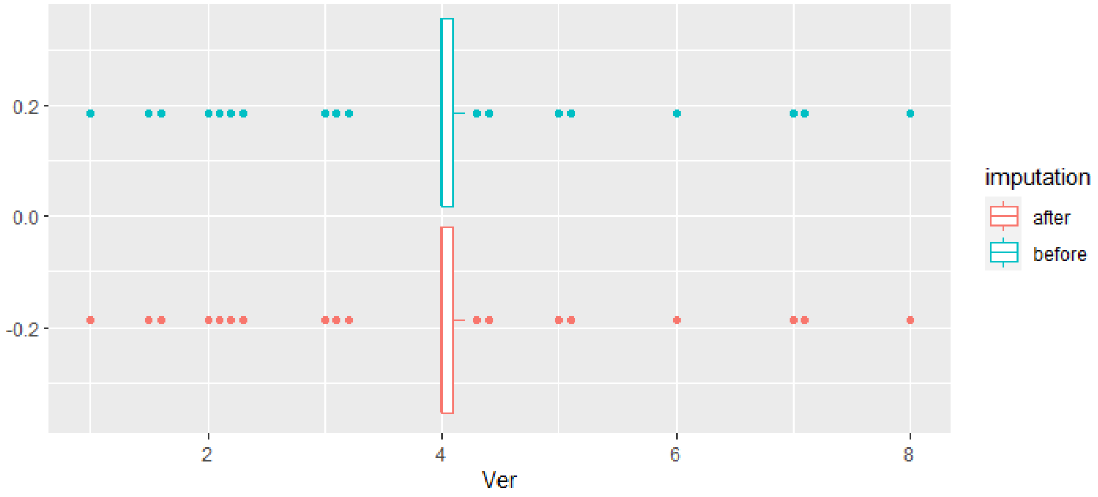


Missing Values Management: Imputation Results

Variable: ***Android.Ver***

NAs source: *"Varies with device"*

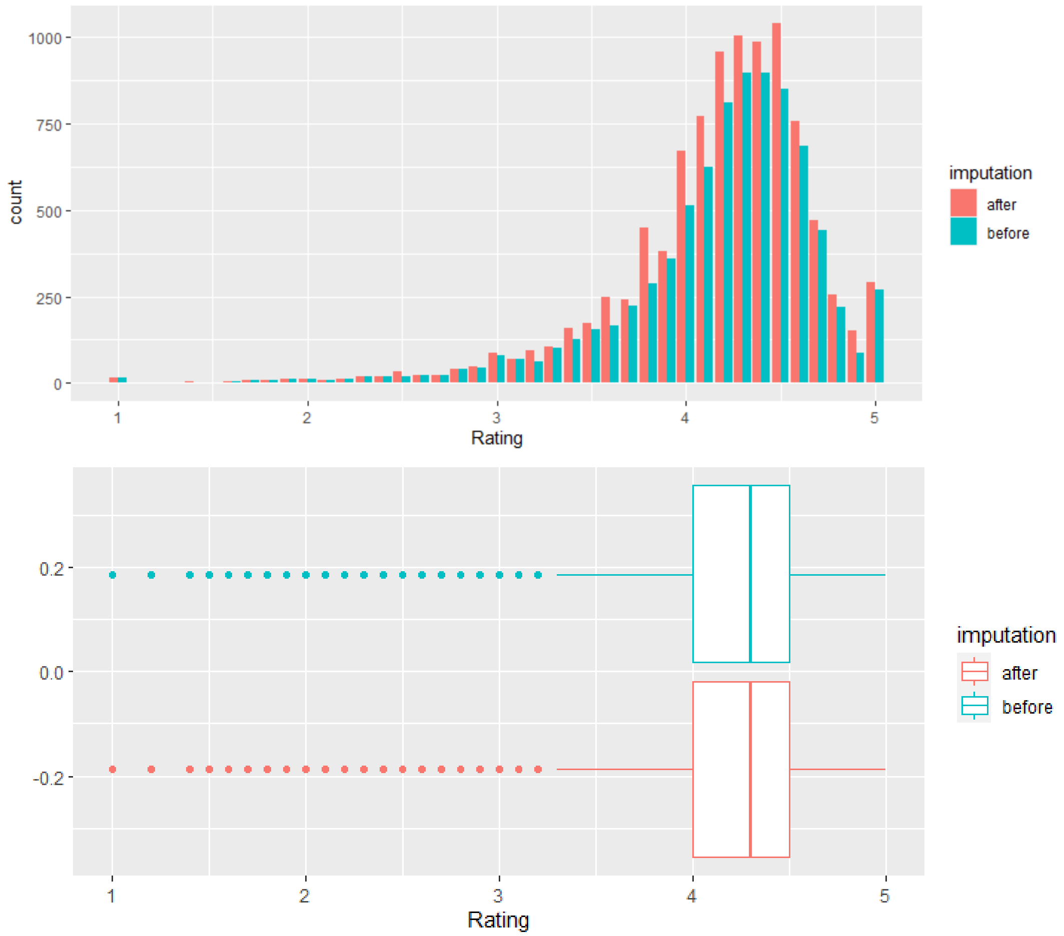




Missing Values Management: Imputation Results

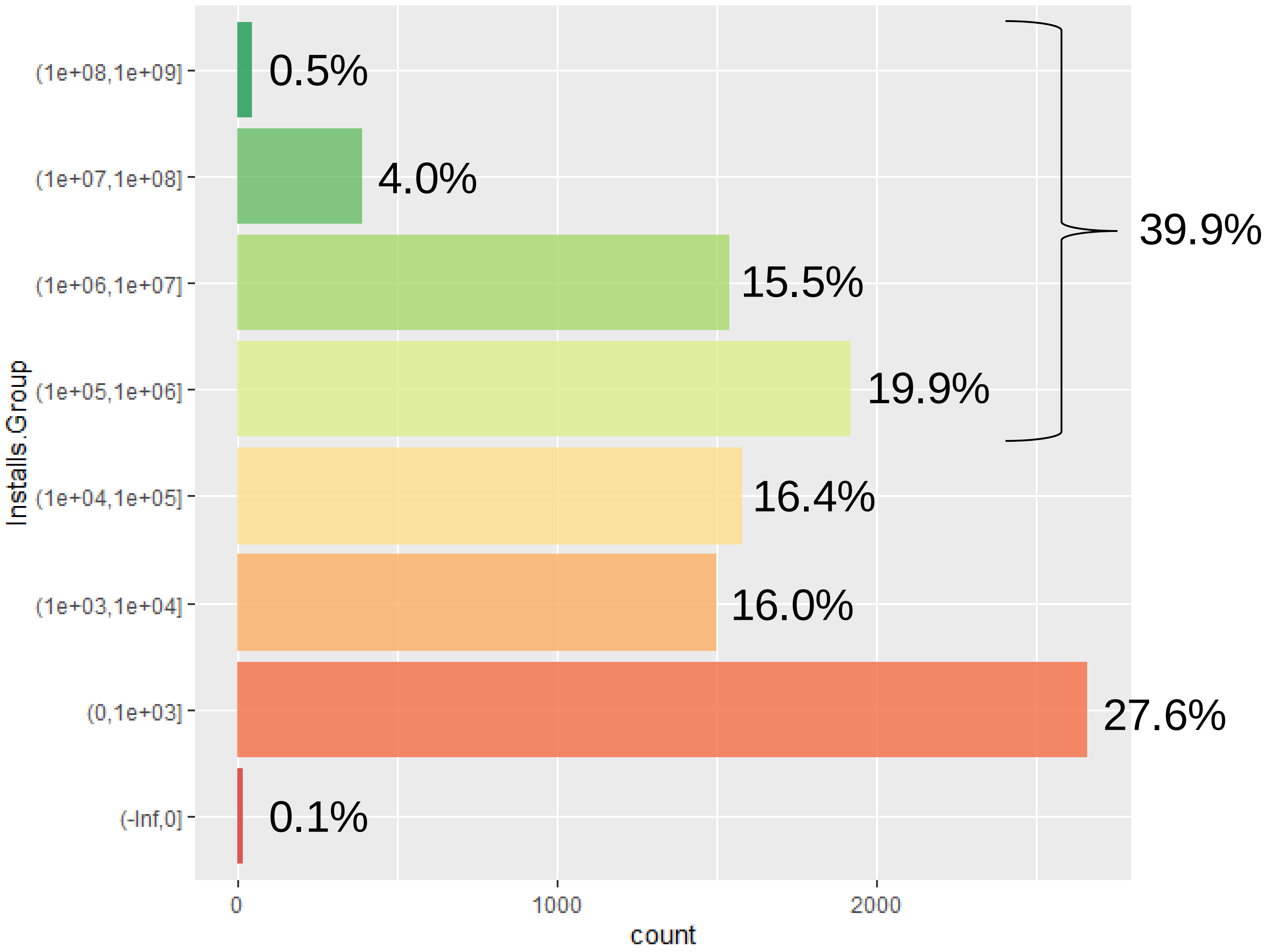
Variable: ***Rating***

NAs source: *User behavior*



EDA: Dependent Variable

Variable: ***Installs***



Takeaway

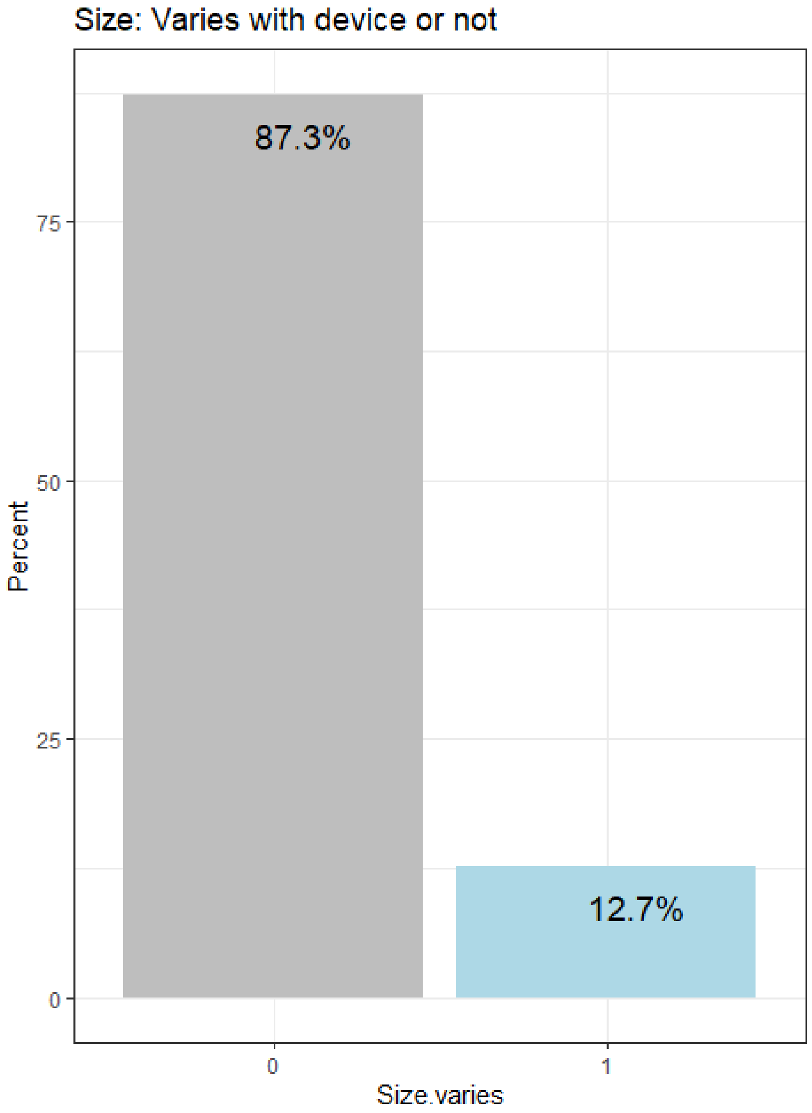
- ***In the largest group, Apps have only thousands of installations.***
- ***Almost 40% of the Apps are doing well, which have 100,000+ installations.***
- ***20% of the Apps have Millions of installations.***

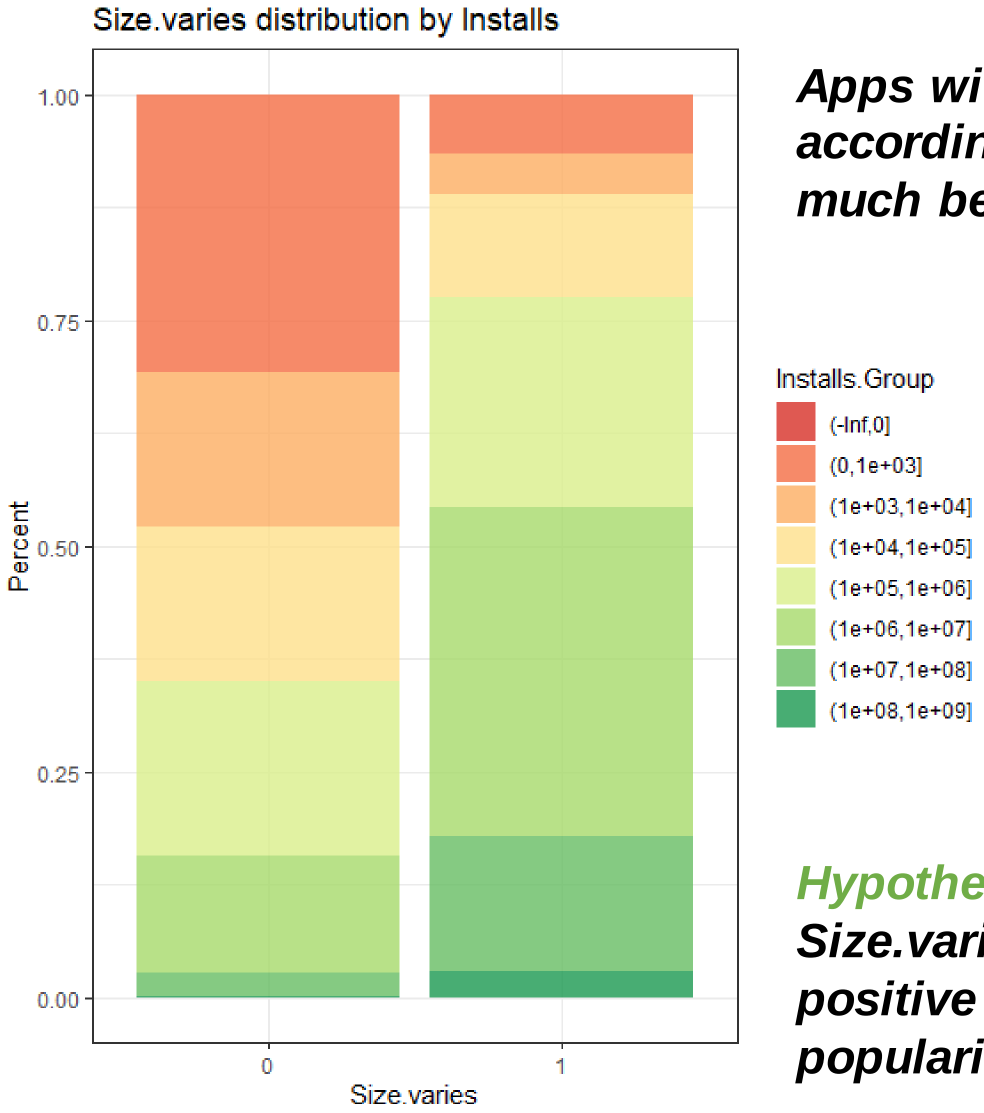
A trick of understanding

- ***Associate the color red or orange with dangerous or warning.***
- ***Associate the color green with safe or healthy.***

EDA: Explanatory Variables (Binary)

Variable: ***Size.varies***





Takeaway

Most Apps have only one single size for any devices.

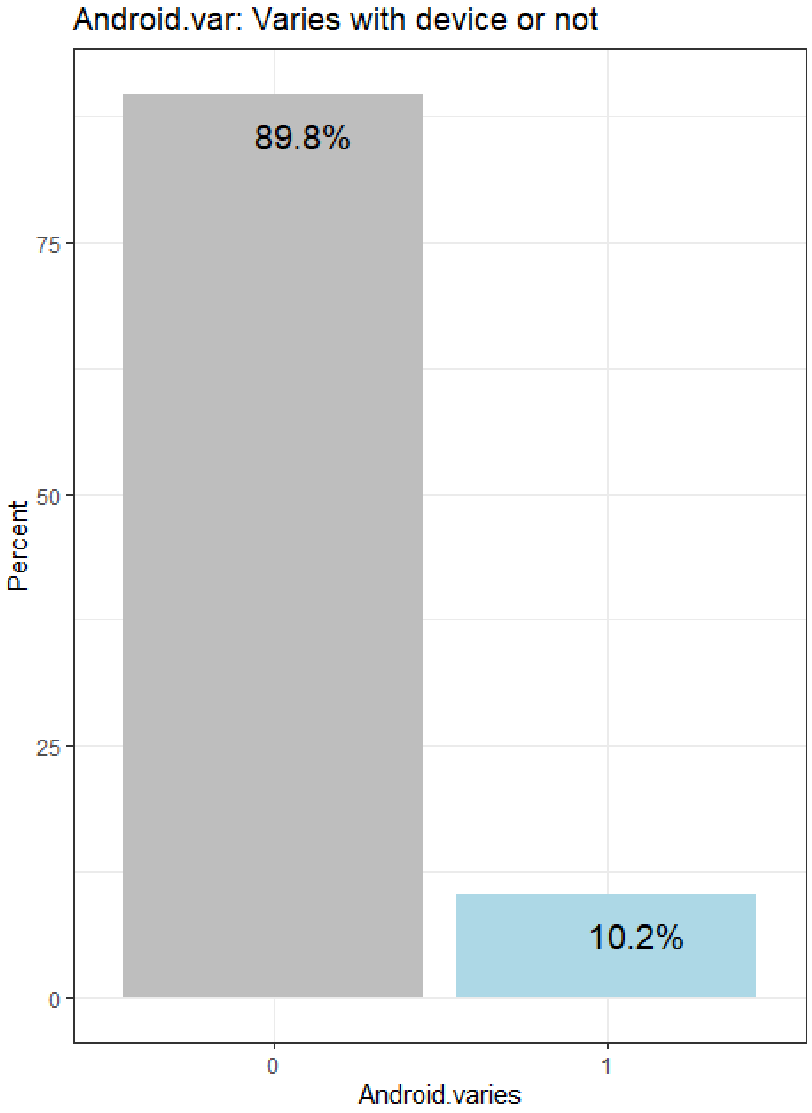
Apps with various sizes according to devices are doing much better.

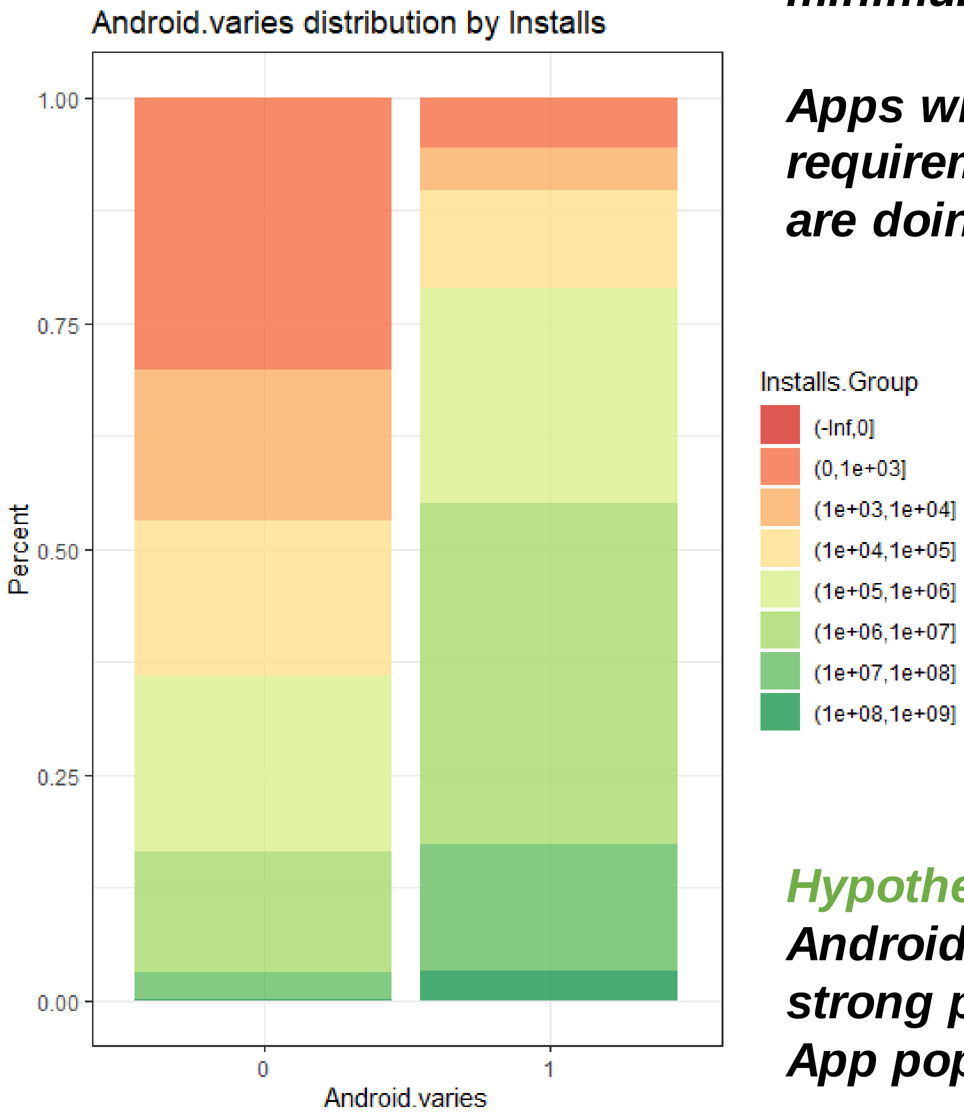
Hypothesis

Size.varies could be a strong positive predictor of the App popularity.

EDA: Explanatory Variables (Binary)

Variable: ***Android.varies***





Takeaway

Most Apps require one specific minimum Android version.

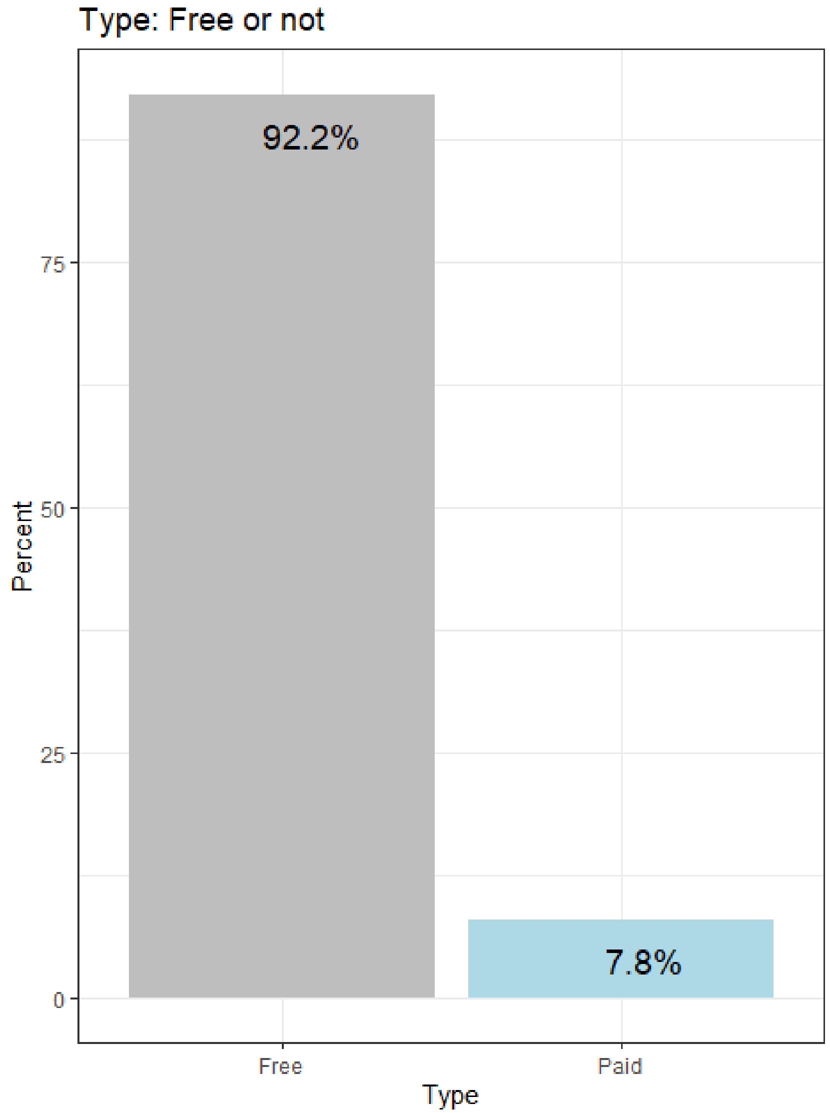
Apps which version requirement varies with device are doing much better.

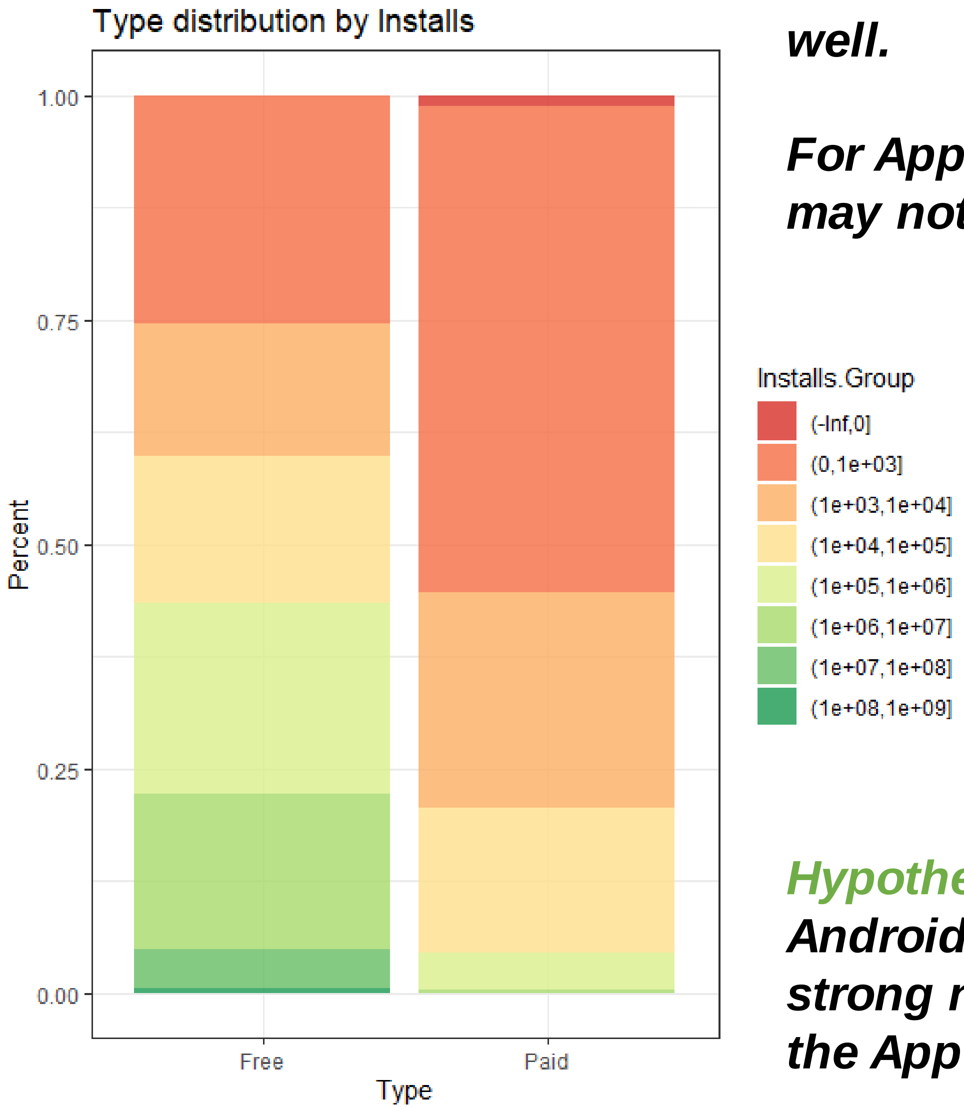
Hypothesis

Android.varies could be a strong positive predictor of the App popularity.

EDA: Explanatory Variables (Binary)

Variable: ***Type***





Takeaway

Most Apps are free, but less than half of them are doing well.

For Apps installation, charging may not be a good idea.

Hypothesis

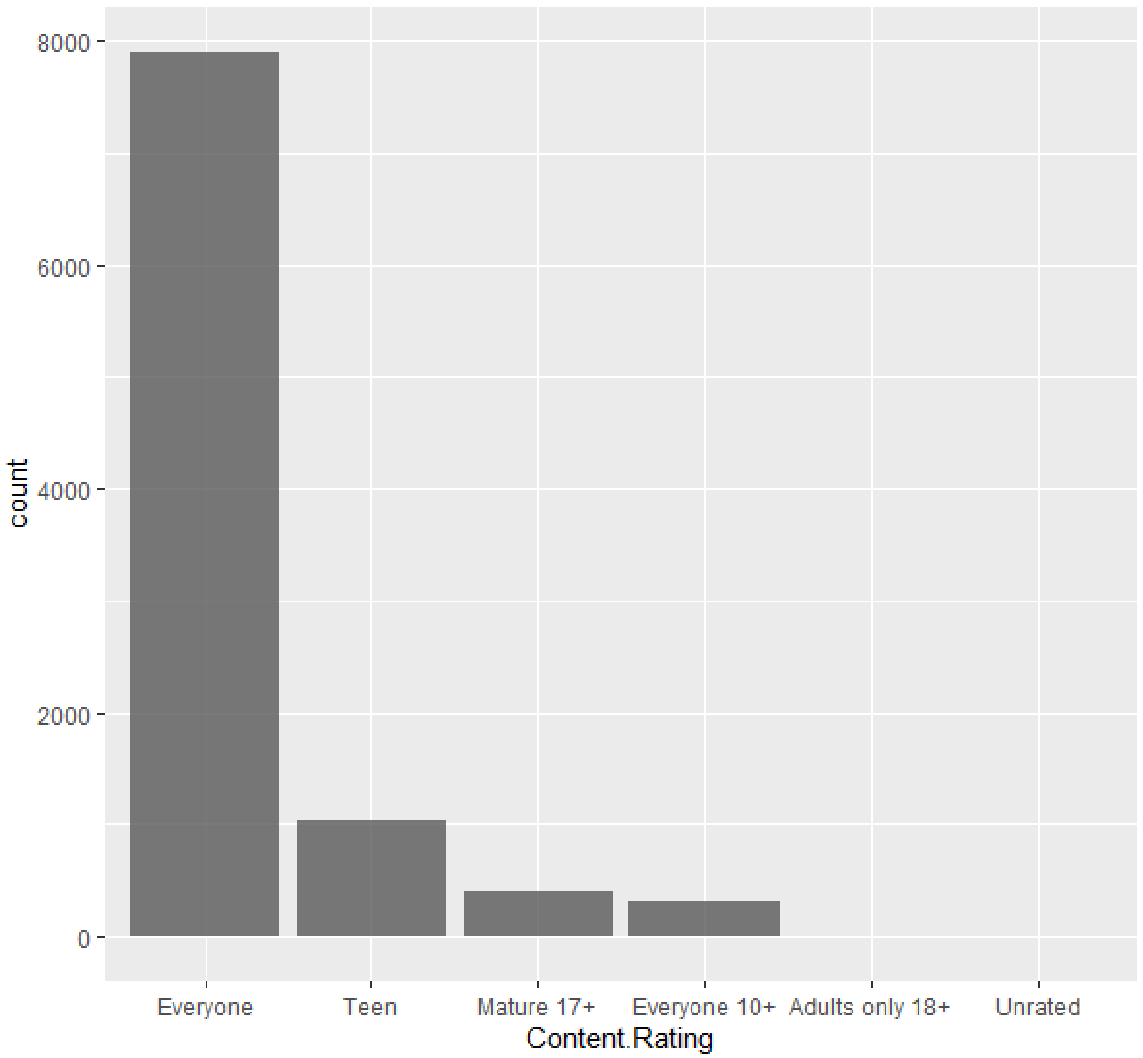
Android.varies could be a strong negative predictor of the App popularity.

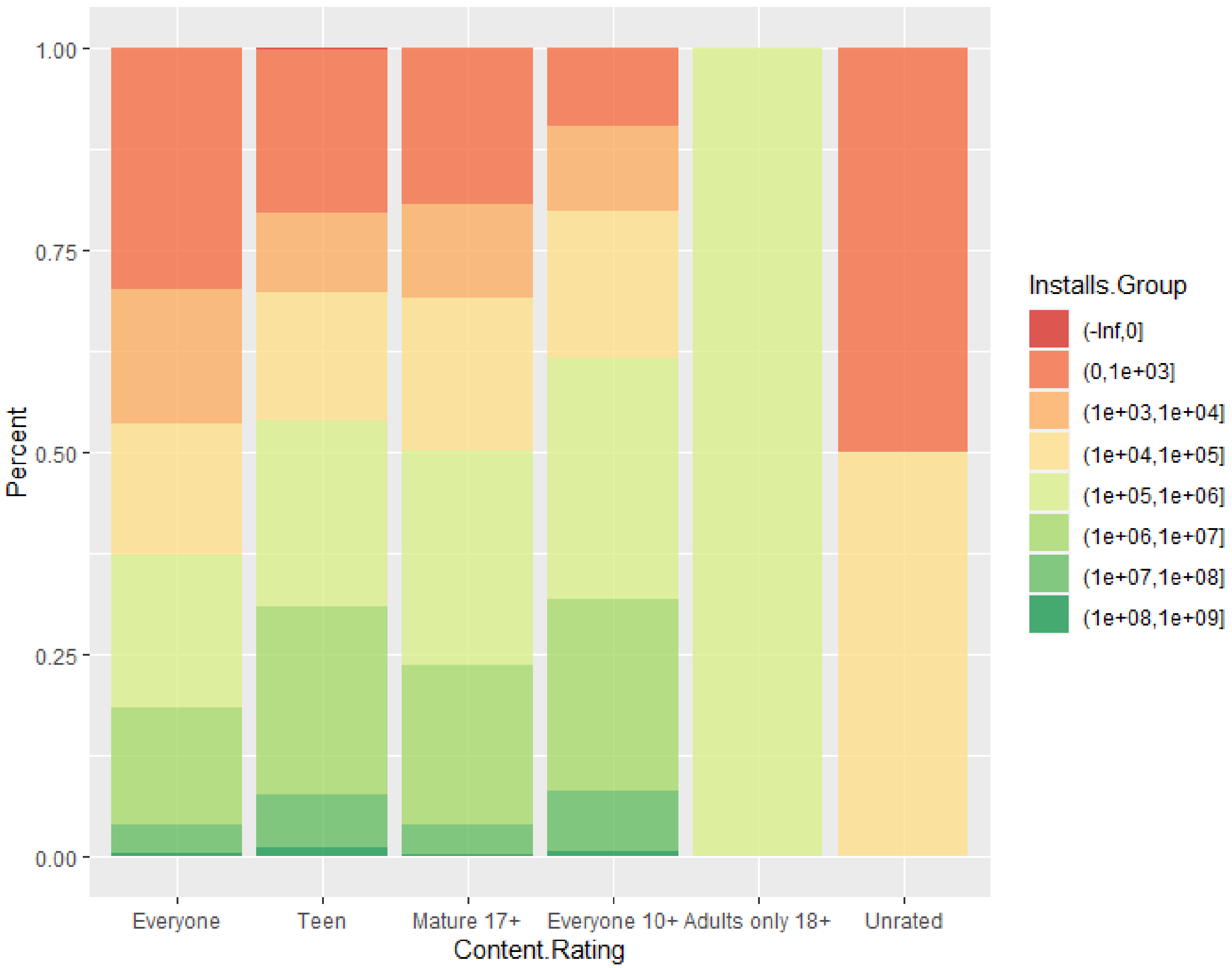
EDA: Explanatory Variables (Multi-valued)

Hypothesis

Content.Rating might not be a strong predictor.

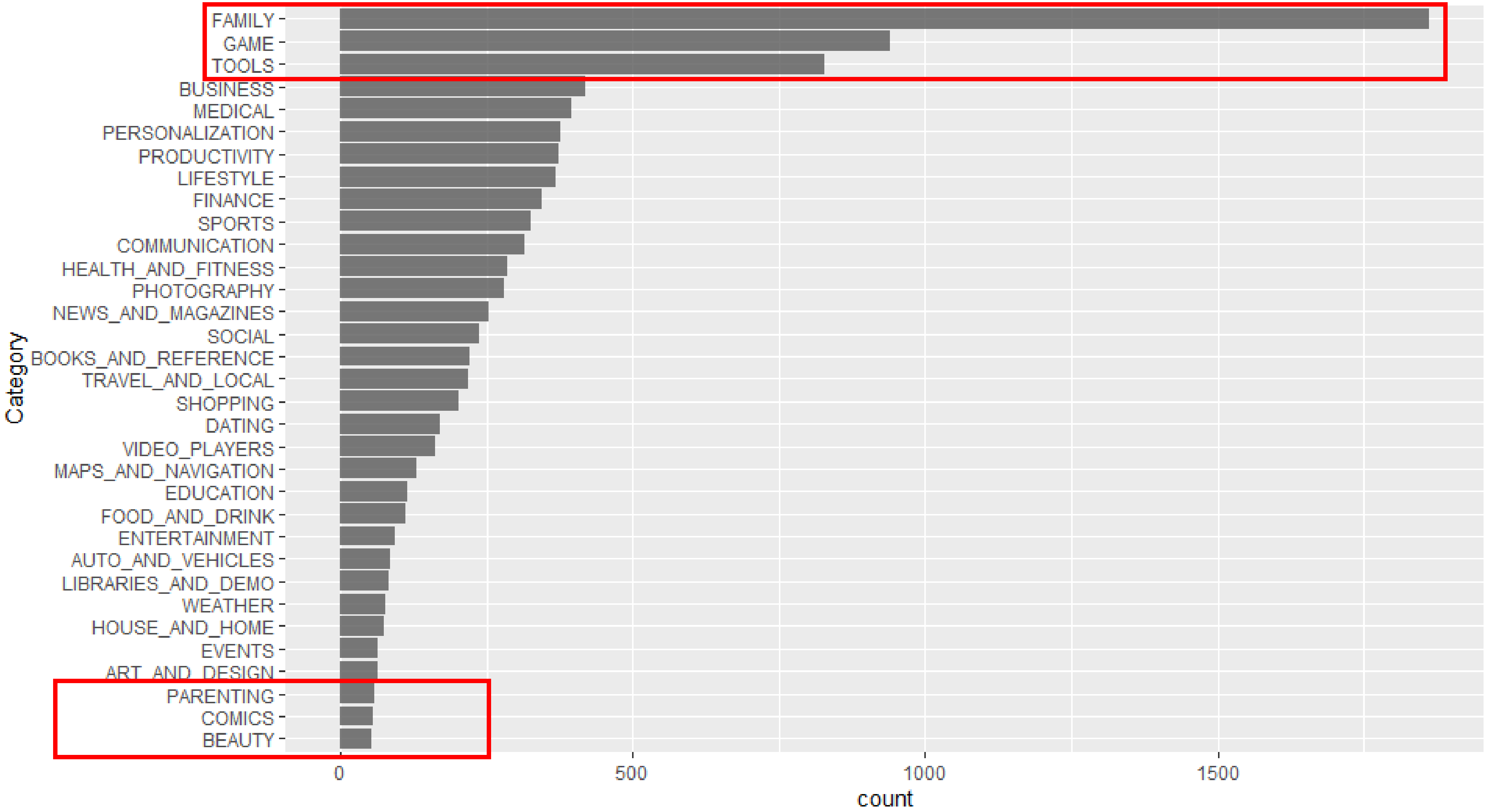
Variable: ***Content.Rating***





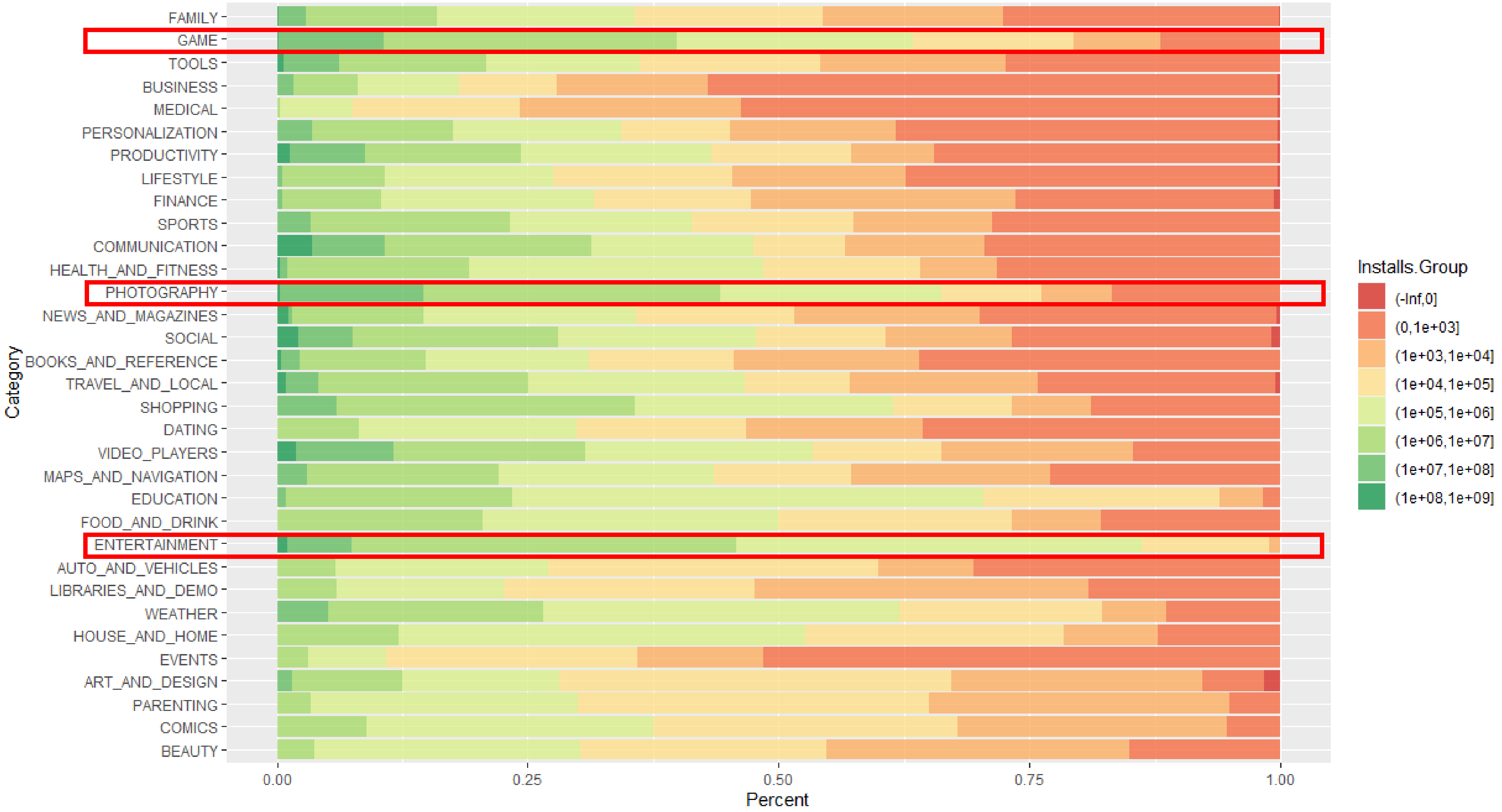
EDA: Explanatory Variables (Multi-valued)

Variable: ***Category (33 values)***



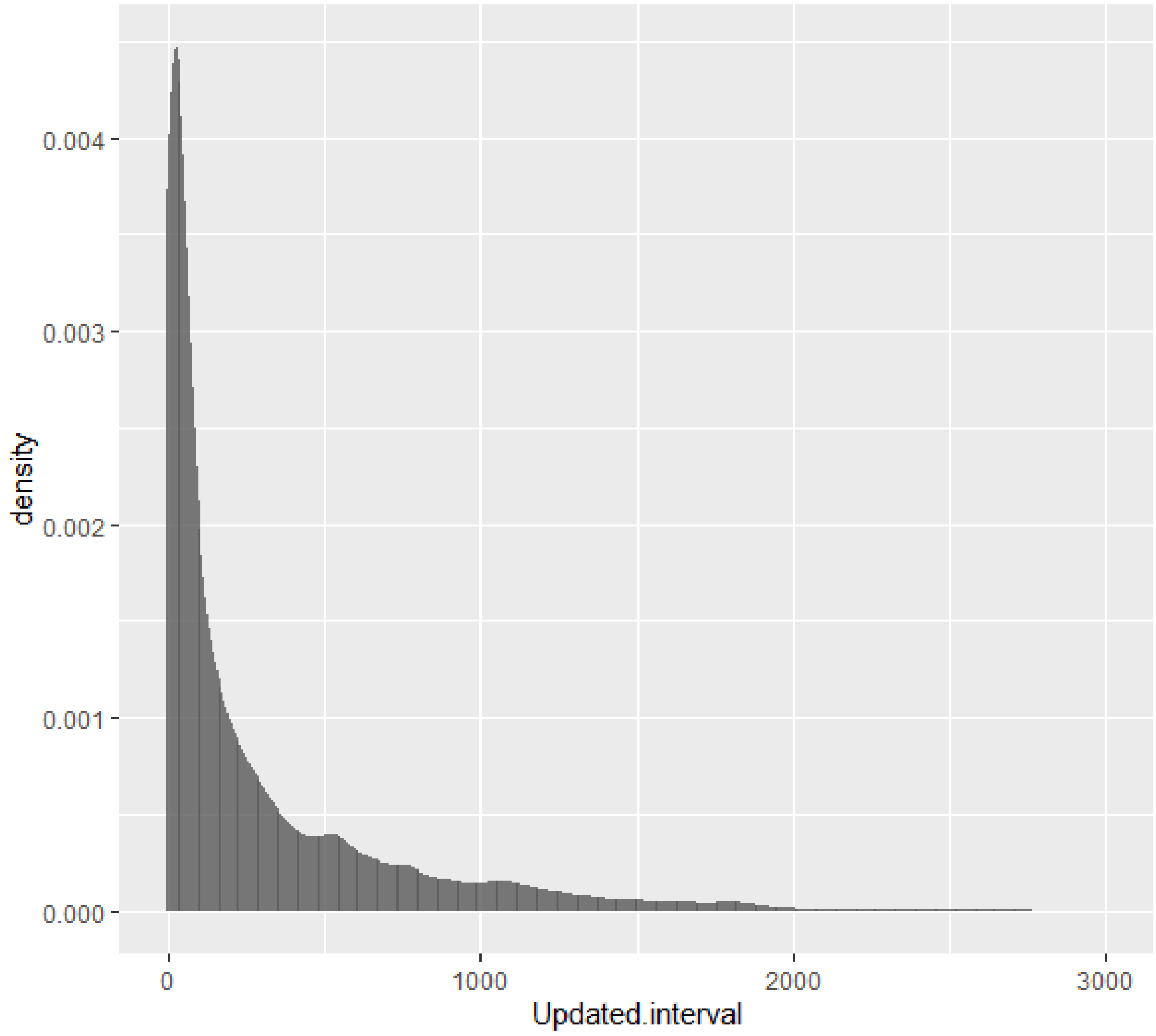
EDA: Explanatory Variables (Multi-valued)

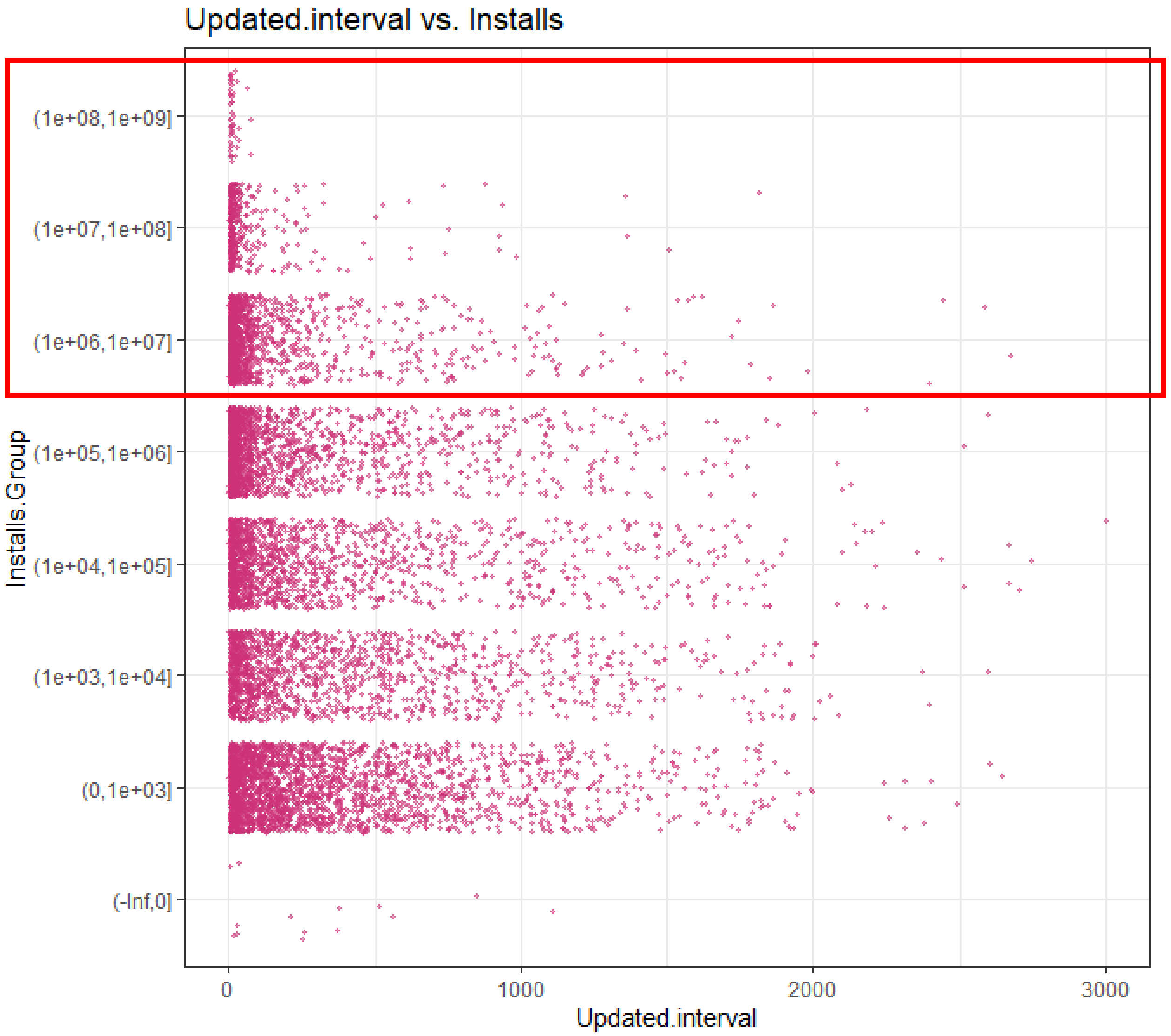
Variable: ***Category (33 values)***



EDA: Explanatory Variables (Numerical)

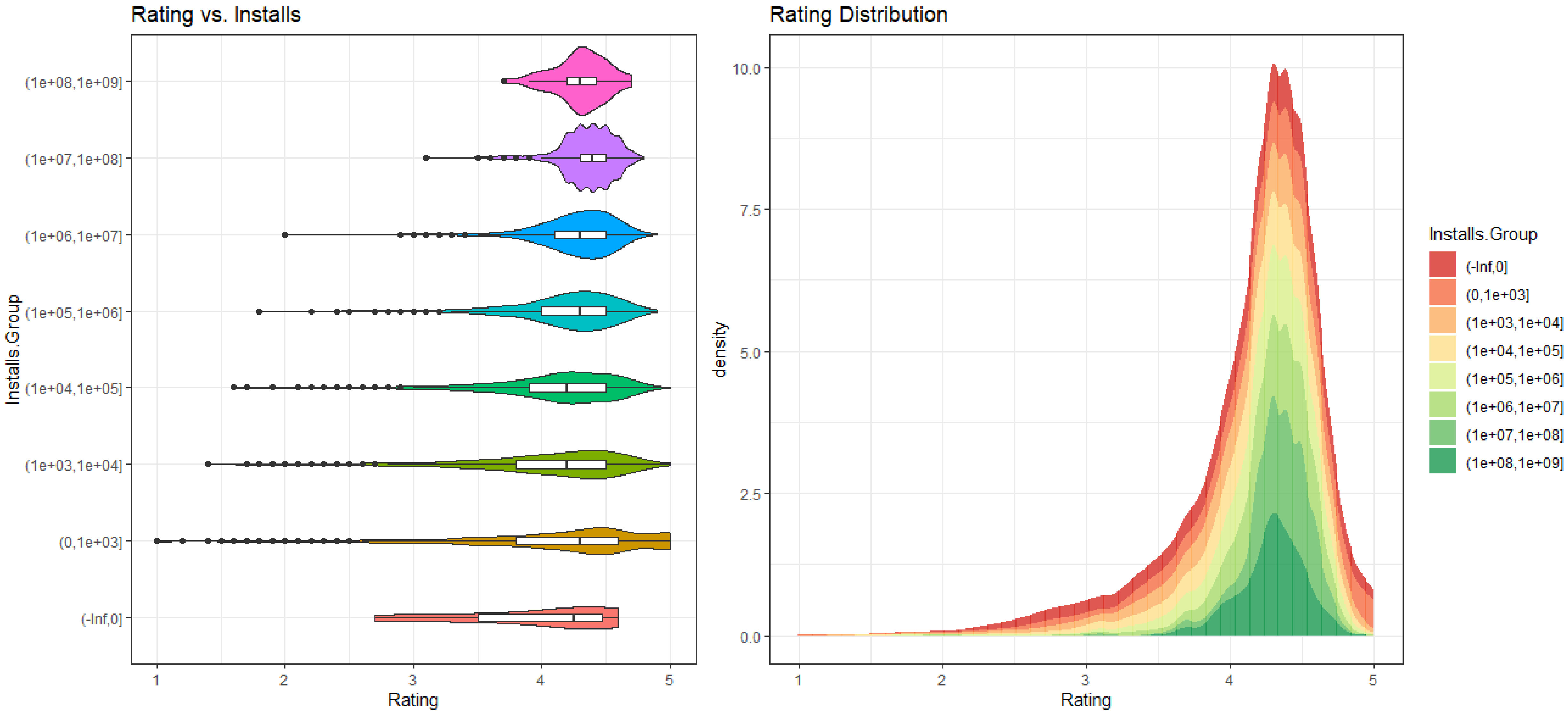
Variable: ***Updated.interval (days)***





EDA: Explanatory Variables (Numerical)

Variable: ***Rating vs. Installs***

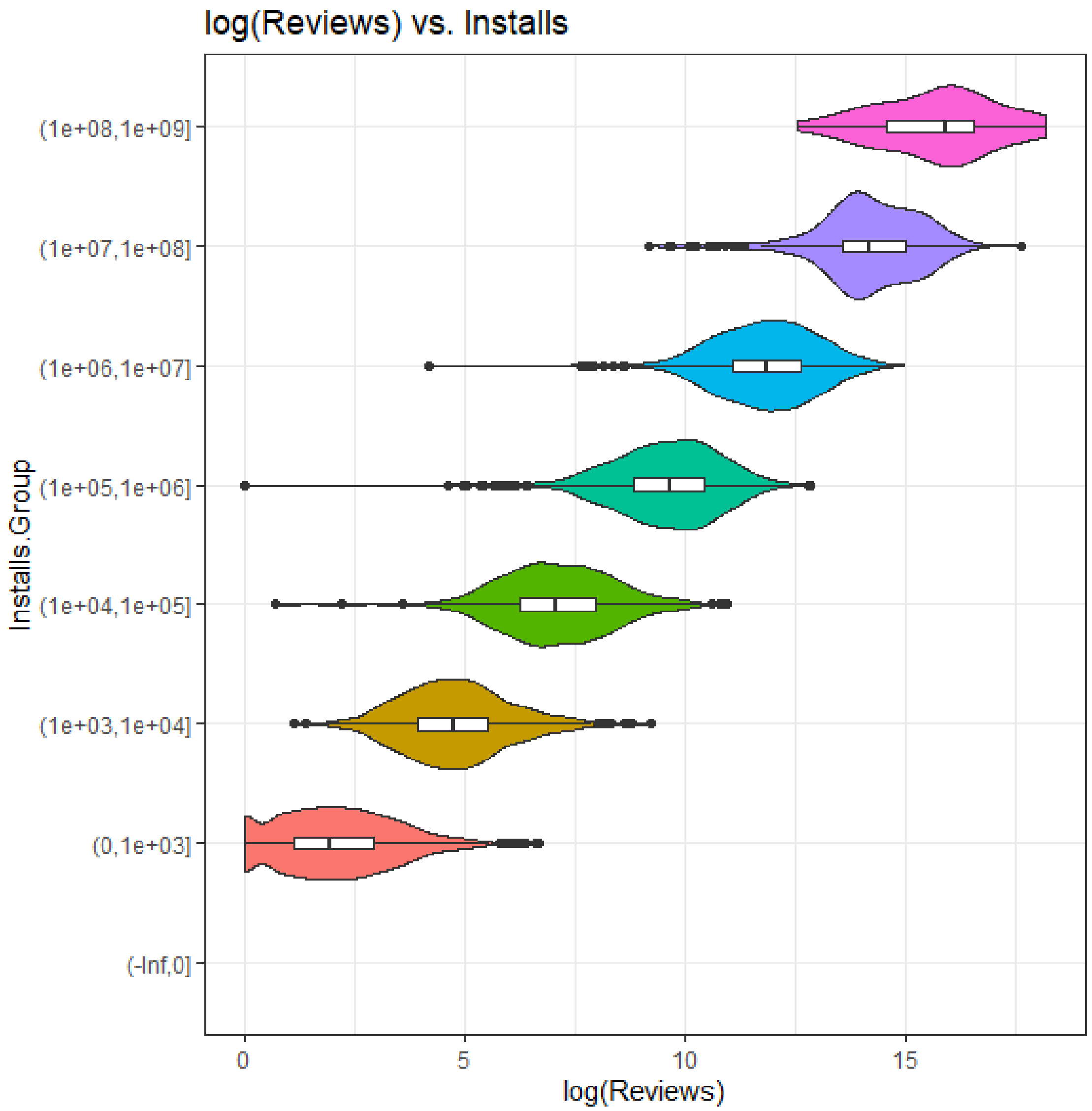


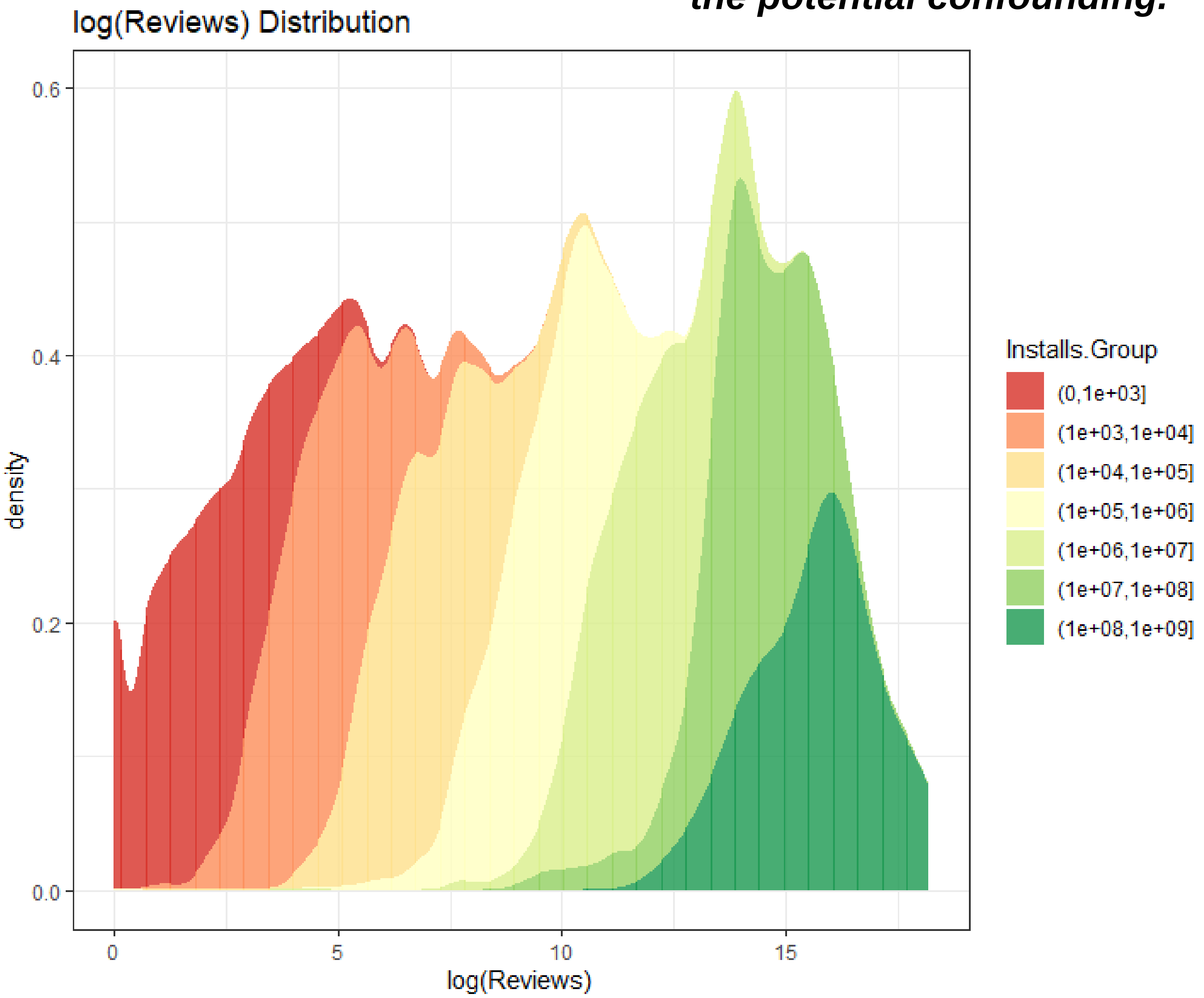
EDA: Explanatory Variables (Numerical)

Variable: ***Reviews vs. Installs***

Hypothesis
Reviews might be a strong positive predictor.

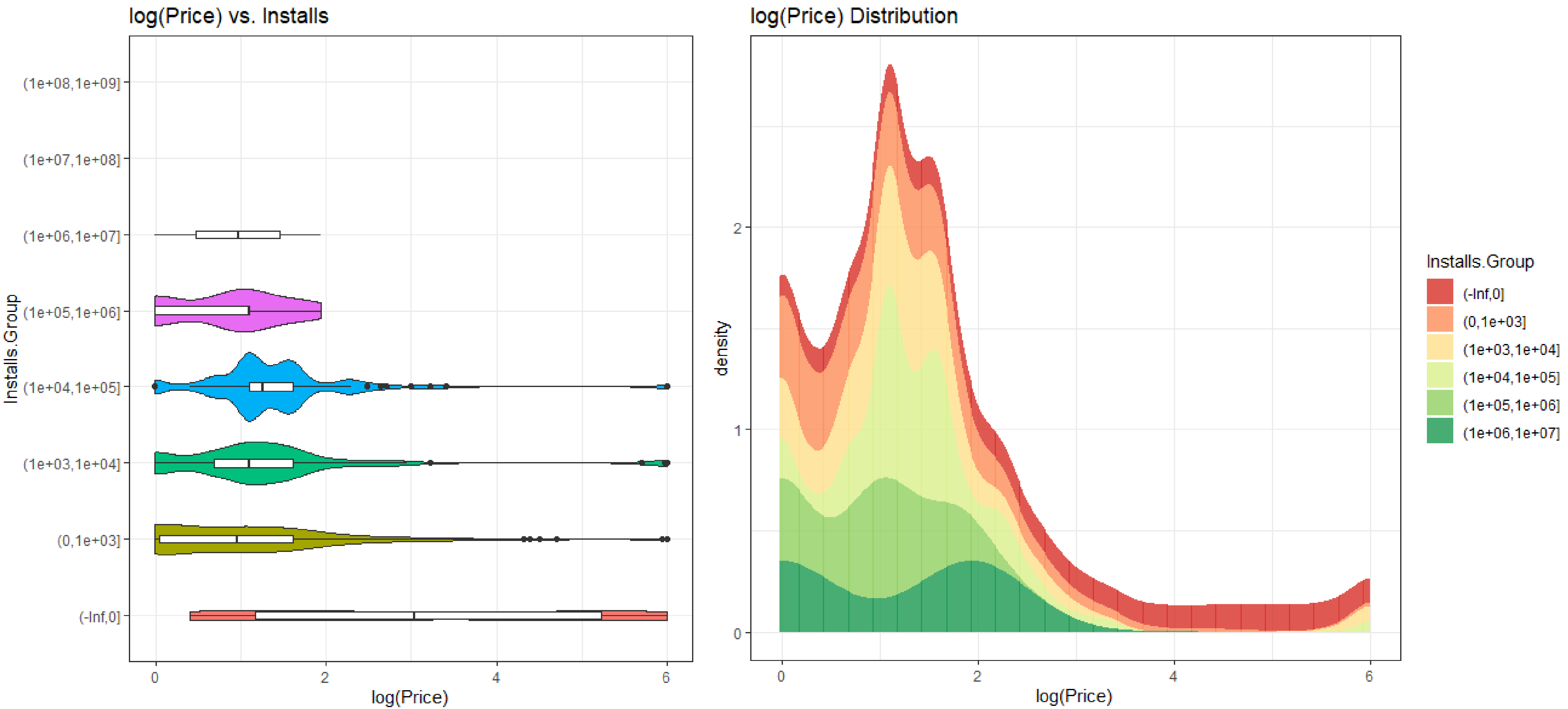
However, be careful about the potential confounding.





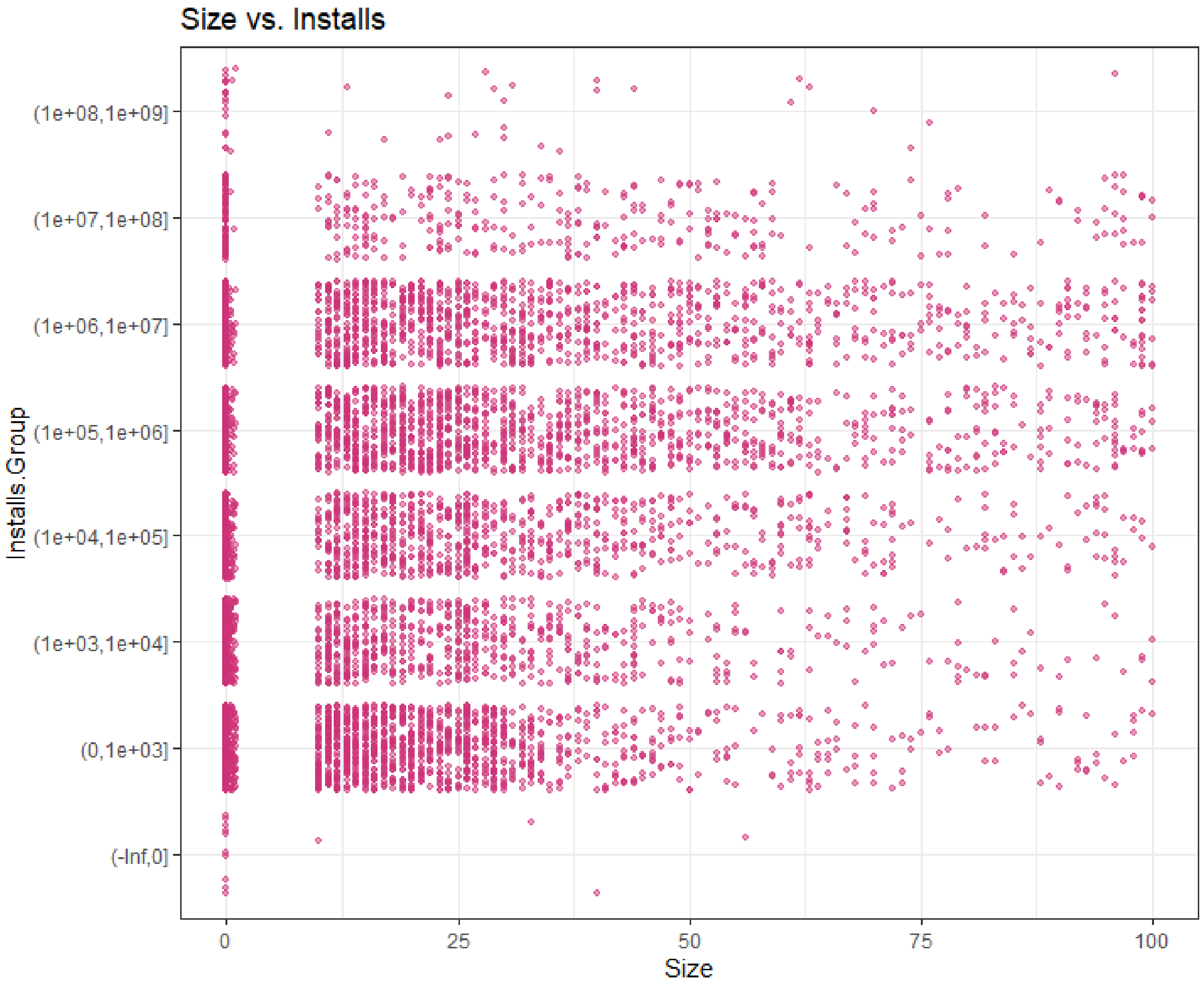
EDA: Explanatory Variables (Numerical)

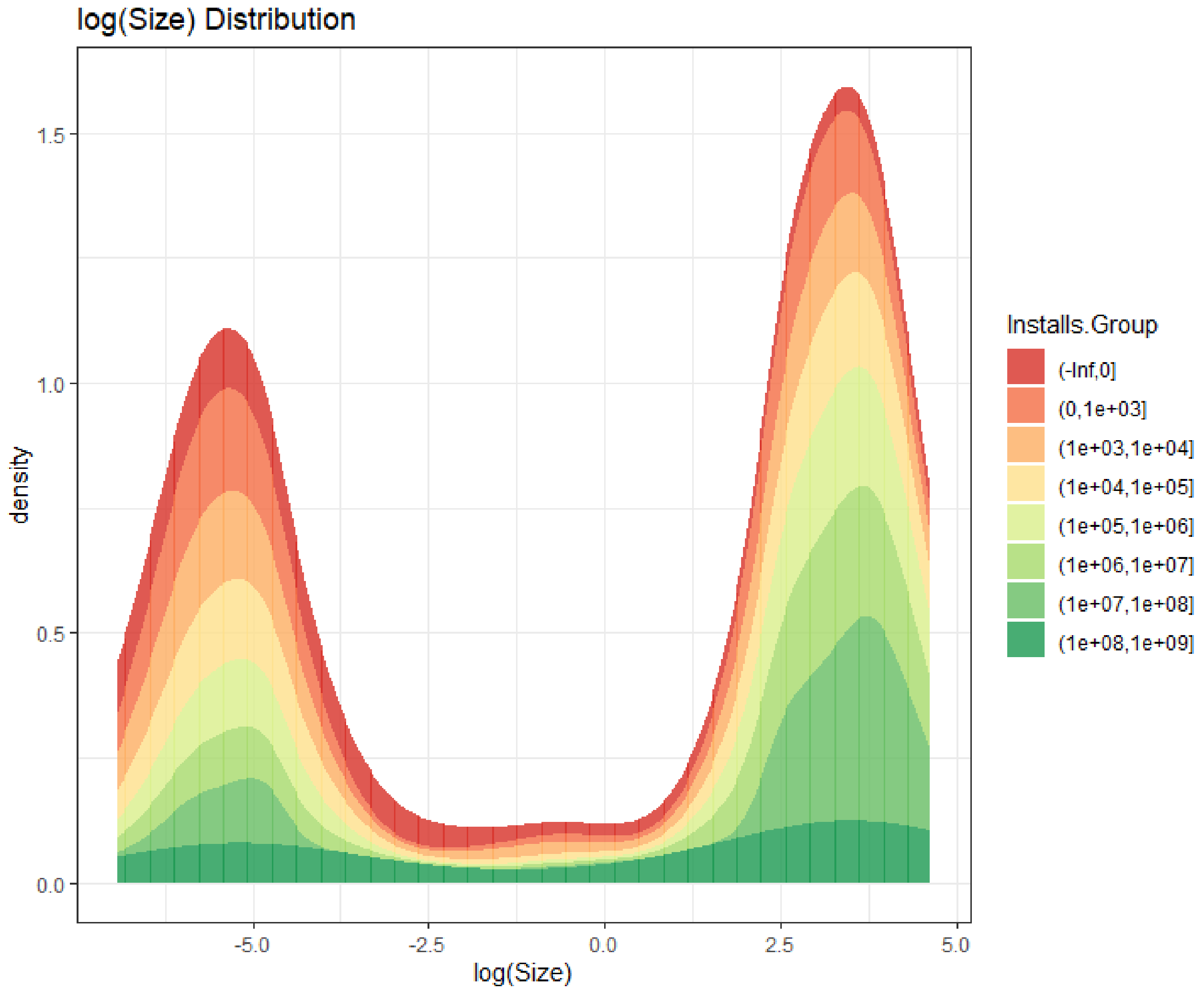
Variable: ***Price vs. Installs***



EDA: Explanatory Variables (Numerical)

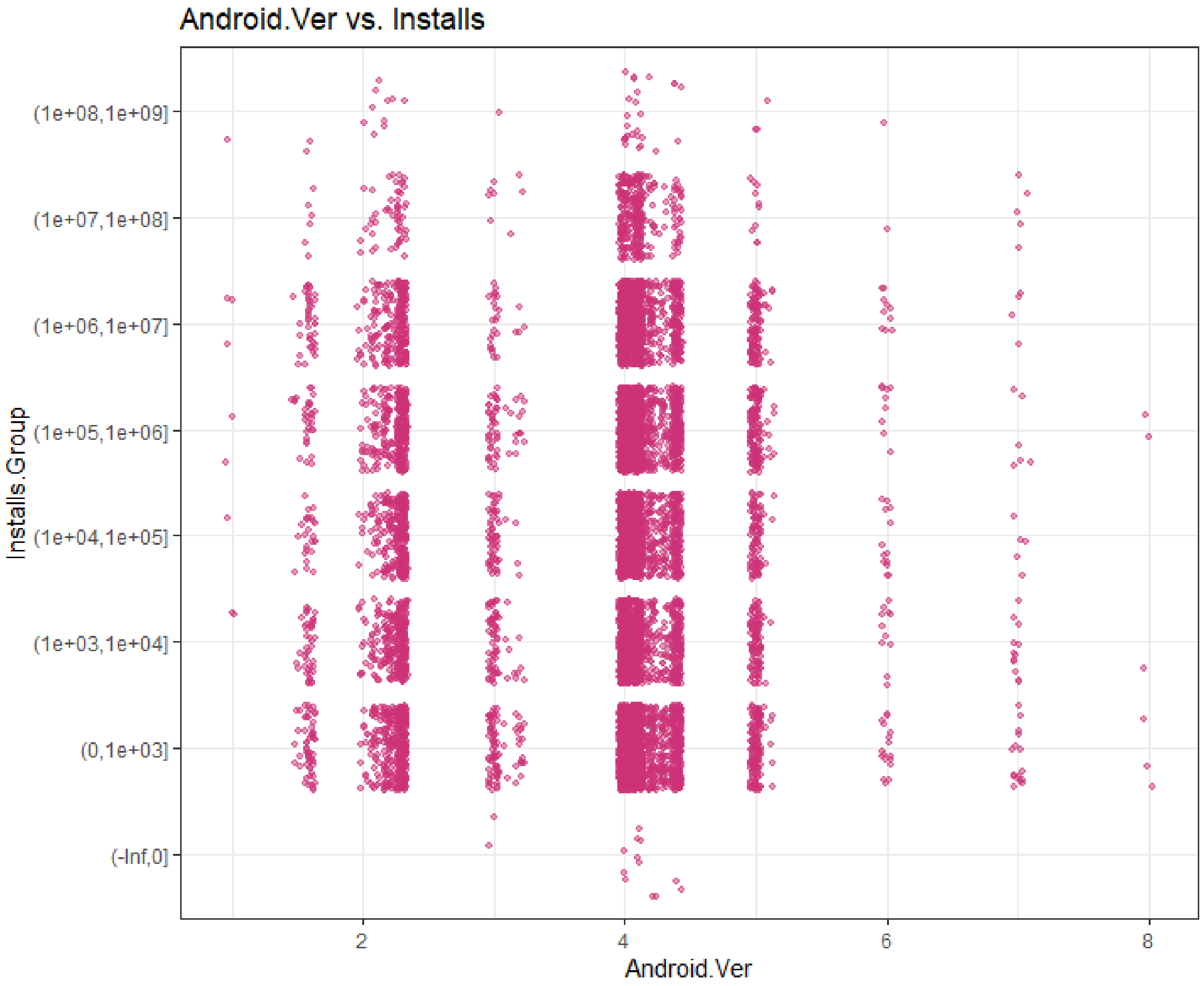
Variable: ***Size vs. Installs***

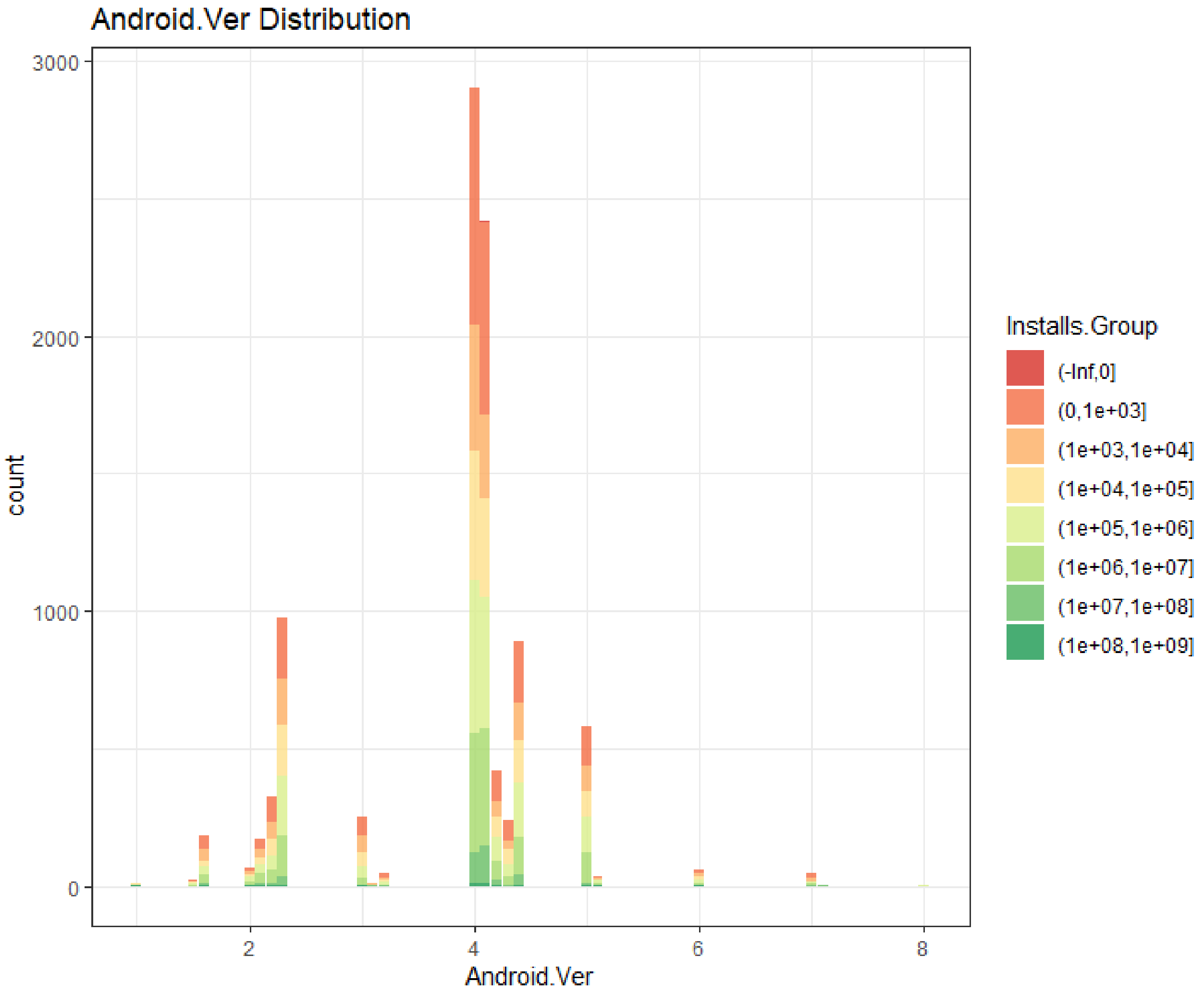




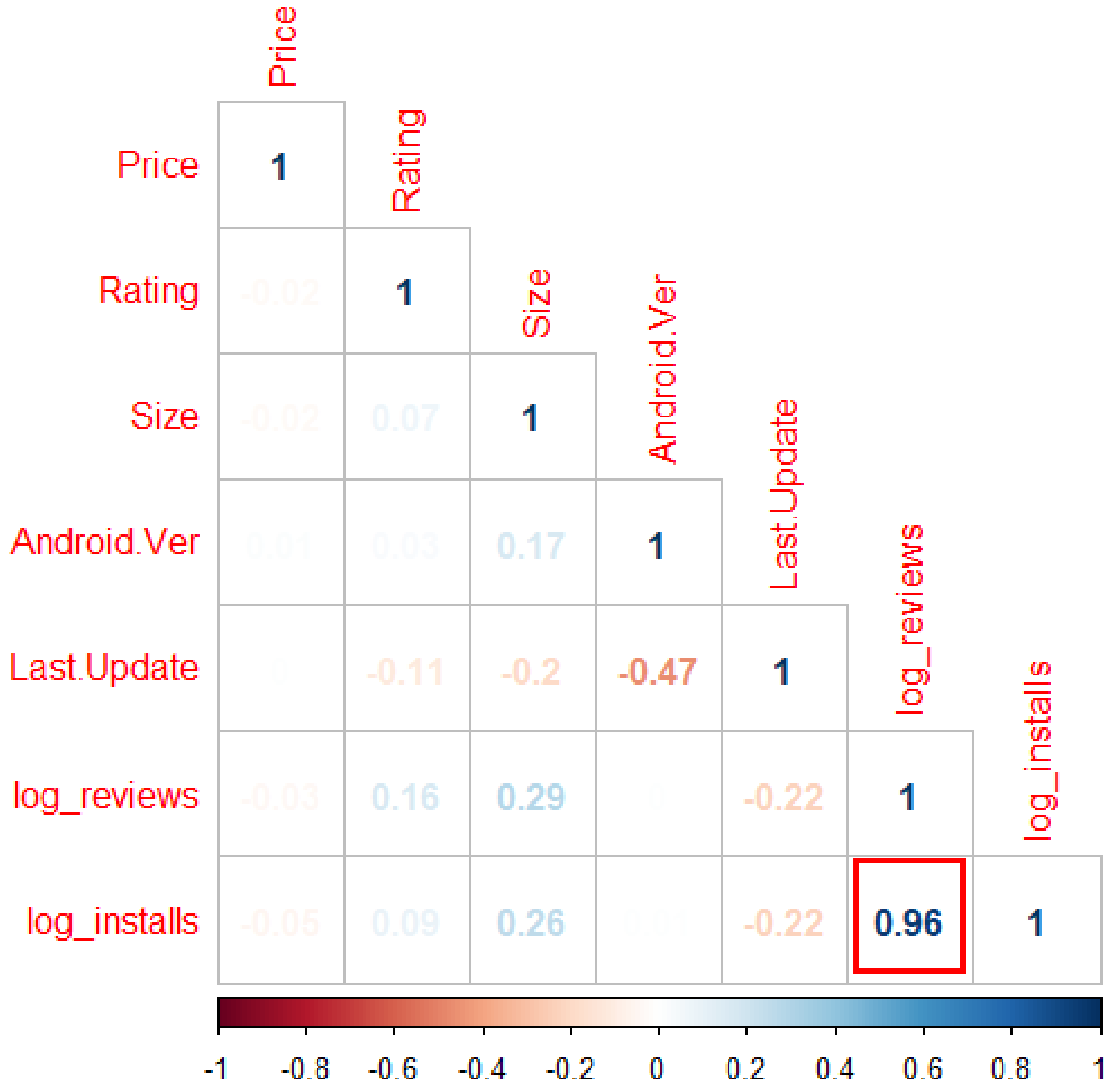
EDA: Explanatory Variables (Numerical)

Variable: ***Android.Ver vs. Installs***





EDA: Correlation Matrix for Numerical Variables



Analysis

- ***There is a highly positive correlation between log(reviews) and log(installs).***

- ***Such correlation will cause multicollinearity and confound how other variables correlate to the response.***

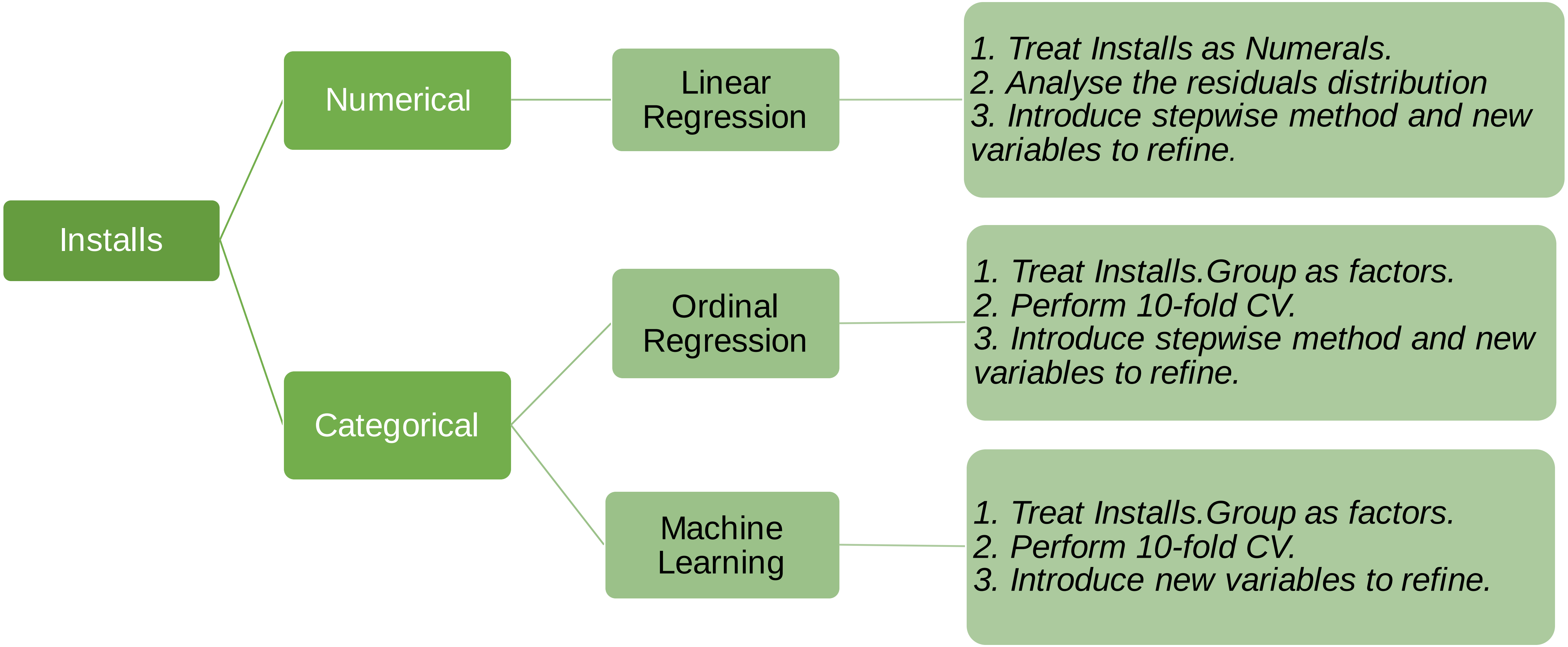
Action

To fix this issue, we drop the variable log(reviews)

Initial Model Selection: Roadmap

Interpretability vs. Prediction Accuracy

Dependent variable Types ***Model Options*** ***Considerations and treatments***



Initial Model: Variables Update

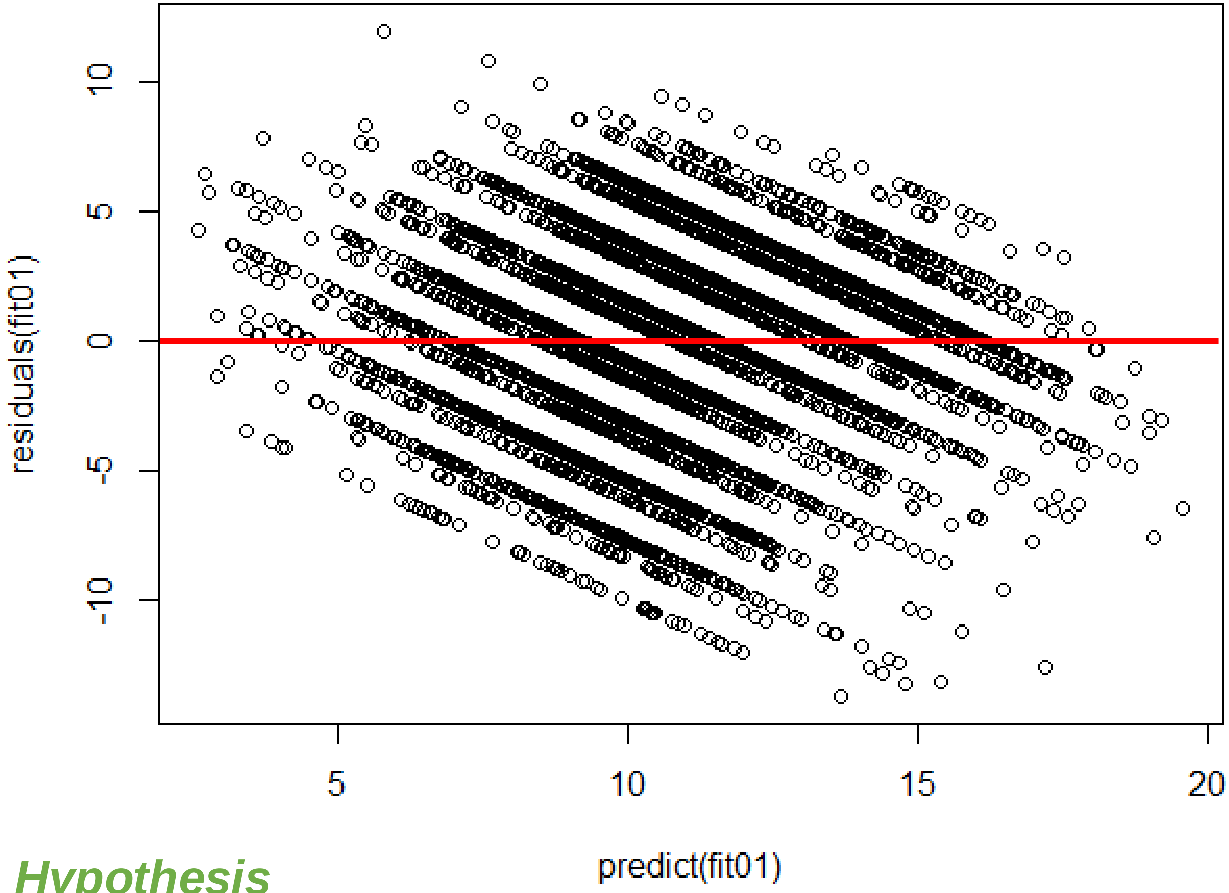
```
Rows: 9,644
Columns: 14
$ App              <chr> "Axe Champ Hit", "BS-Mobile", "CZ-Help", "PrimeD
$ Category         <fct> GAME, COMMUNICATION, BOOKS_AND_REFERENCE, MEDICA
$ Reviews          <int> 1, 1, 2, 3, 11, 20, 21, 30, 31, 49, 68, 2717, 0,
$ Installs         <int> 100, 50, 5, 10, 10, 1000, 5000, 5000, 500, 10000
$ Type             <fct> Free, Free, Free, Free, Free, Free, Free, Free,
$ Price            <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
$ Content.Rating   <fct> Everyone, Everyone, Everyone, Everyone, Everyone
$ Size.varies      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
$ Android.varies   <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
$ Rating           <dbl> 4.4, 5.0, 5.0, 5.0, 5.0, 4.3, 4.4, 4.4, 4.6, 3.4
$ Size             <dbl> 15.0000, 0.6830, 0.0014, 53.0000, 0.0061, 0.0048
$ Android.Ver      <dbl> 4.1, 2.3, 4.4, 4.1, 2.3, 2.3, 4.1, 4.0, 2.3, 2.3
$ Updated.interval <int> 75, 1070, 26, 26, 515, 579, 72, 130, 462, 544, 4
$ Installs.level   <ord> 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 2, 3, 1, 2, 3, 1,
```

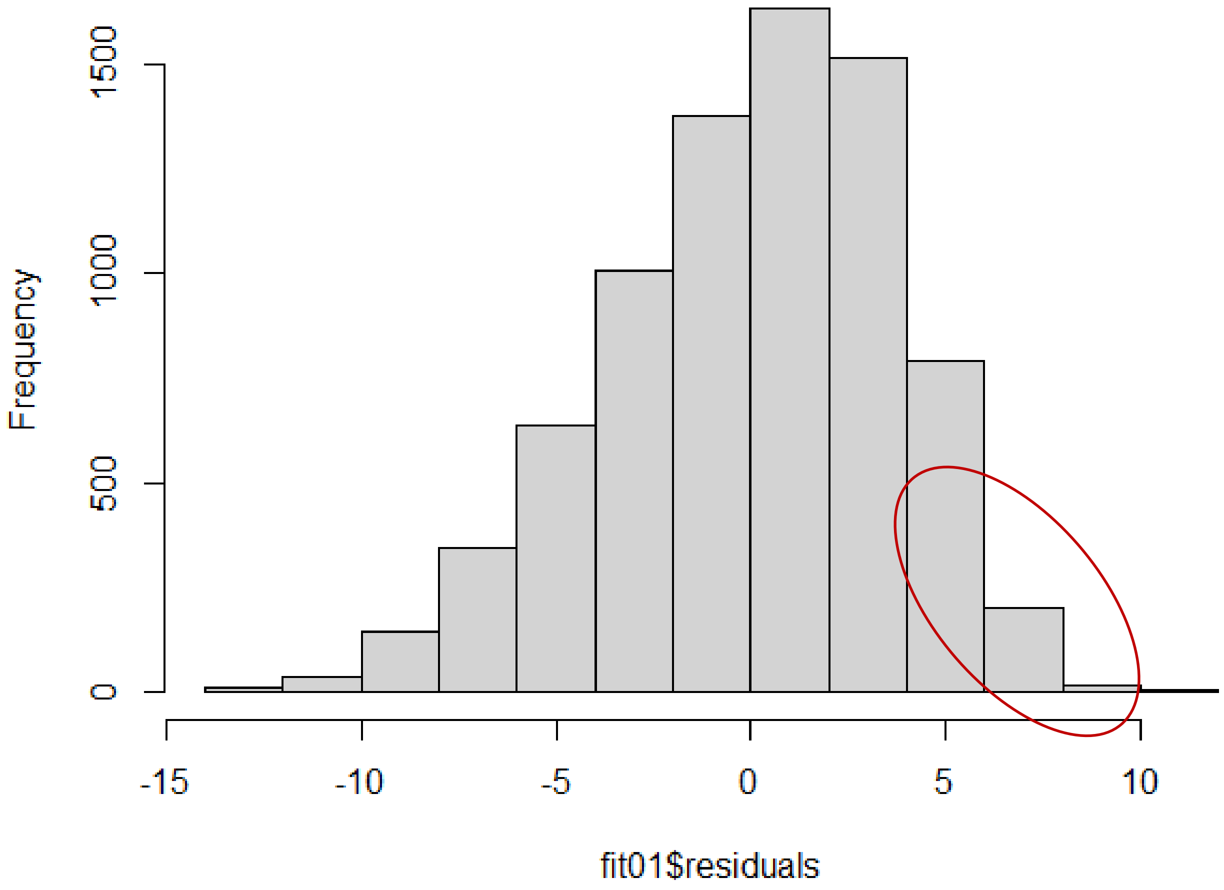
Response

- Numerical: ***Installs*** *(integers)*
- Categorical: ***Installs.level*** *(ordinal factors)*

Initial Model: Liner Regression

```
fit01 = lm(log(Installs)~Category+Rating+Price+Type+Content.Rating+Size+Size.varies+
        Android.Ver+Android.varies+Updated.interval, data = trainset)
```





Hypothesis

Linear regression might not be a good model to predict installations.

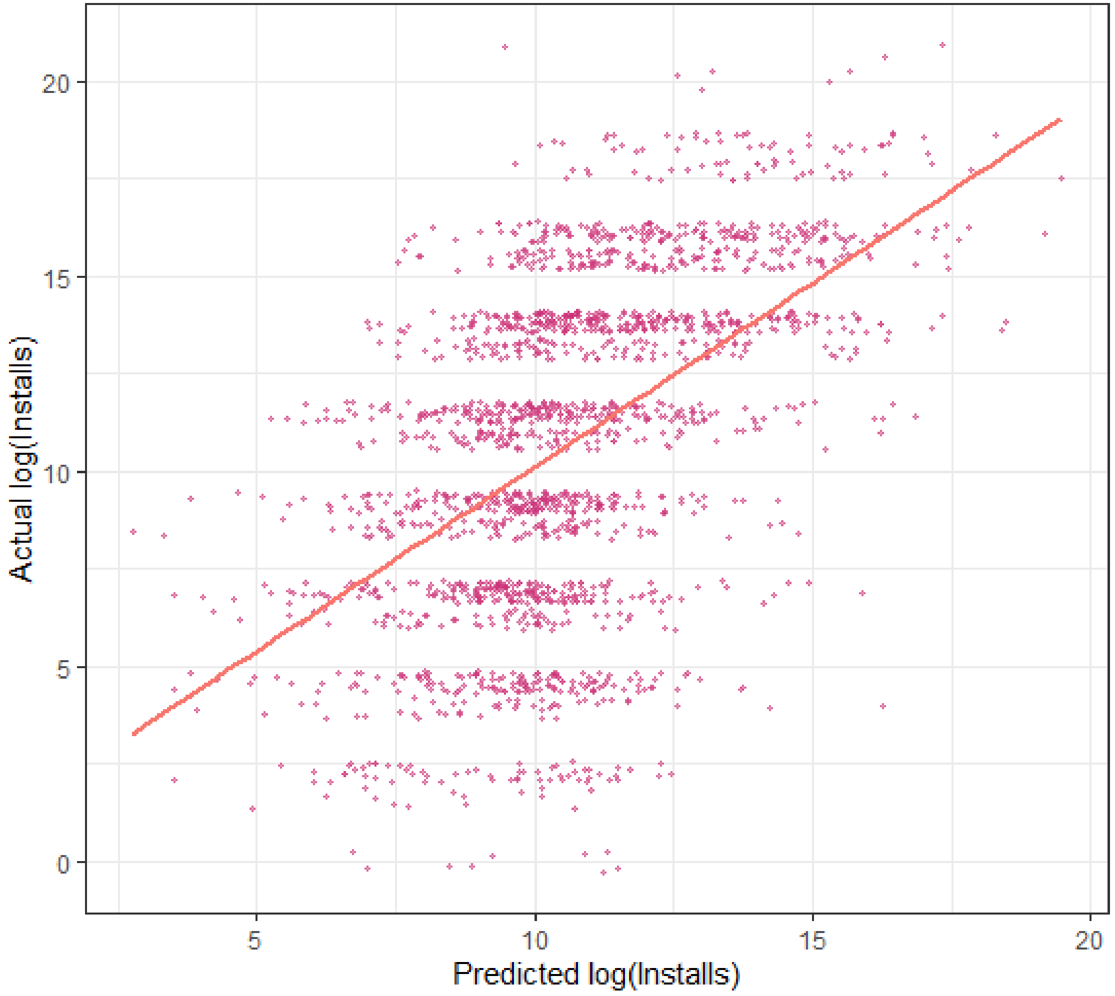
Initial Model: Liner Regression

Tips

$$\textbf{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

fit01 = lm(log(Installs)~**Category**+Rating+Price+Type+Content.Rating+Size+Size.varies+
Android.Ver+Android.varies+Updated.interval, data = trainset)

Predicted vs. Actual log(Installs)



Adjusted R-square = 0.297
RMSE = 3.639

Takeaway

Due to the limitation of the data, linear regression is far from an ideal choice.

Action

Refining our regression method by grouping up the dependent variable.

Initial Model: Ordered Logistic Regression

Conduct a logistic regression with ordinal response

Response:

- *Convert to ordinal factors*

Installs	***Installs.level***
$[0, 10^3]$	1
$(10^3, 10^4]$	2
$(10^4, 10^5]$	3
$(10^5, 10^6]$	4
$(10^6, 10^7]$	5
$(10^7, 10^8]$	6
$(10^8, 10^9]$	7

Predictors:

- *Treat coefficients as exponents*

How to interpret

$$\boldsymbol{Odds\ Ratios = e^{Coefficient}}$$

$(\boldsymbol{Odds\ Ratios - 1})$ *is the* ***probability increment*** *of the response* ***moving up to the next class.***

Here means the likelihood that installations jump by 10 times!

Ordered Logistic Regression: Driving Factors

```
order01 = polr(Installs.level~Rating+Type+Size+Size.varies+Android.Ver+Android.varies+Category+
                     Updated.interval, data = store.log, Hess=TRUE)
```

Significant Predictors	*t value*	*Odds ratio*	*Probability of 10x installations*
Type-Paid	-22.03	0.18	-82.01%
Size	20.80	1.02	1.92%
Size.varies-Yes	13.48	4.17	316.73%
Last.update	-12.16	1.00	-0.07%
Rating	7.60	1.31	31.48%
Android.Ver	-8.03	0.82	-18.03%
Android.varies-Yes	4.34	1.65	65.44%

AIC: 29514.77

Warning message
Rank-deficient
(Multicollinearity between Factors)
Fix
*Dropping **Price**, **Content.Rating***

Takeaway

- *If you think your App is too popular , you may want to **Charge.***
- ***Size** is a strong predictor because it may represent the App function or UX to some extent.*
- *Providing **multiple versions or system solutions** could be a strong signal of highly popularity.*

Ordered Logistic Regression: Driving Factors

```
order01 = polr(Installs.level~Rating+Type+Size+Size.varies+Android.Ver+Android.varies+Category+
                        Updated.interval, data = store.log, Hess=TRUE)
```

Top10 most significant Categories	*t value*	*Odds ratio*	*Probability of 10x installations*
GAME	16.18	3.01	201.26%
PHOTOGRAPHY	13.66	4.43	343.17%
BUSINESS	-10.05	0.37	-63.02%
MEDICAL	-9.91	0.37	-62.84%
ENTERTAINMENT	8.15	4.14	314.21%
TOOLS	8.11	1.73	73.00%
SHOPPING	7.12	2.45	145.03%
EDUCATION	6.40	2.64	164.22%
VIDEO_PLAYERS	6.35	2.45	144.60%
COMMUNICATION	5.47	1.81	81.33%

Is a good choice more important than effort?

Takeaway

- *Certain industries will provide significant higher success rate than others in terms of getting more users.*
- *In Such industries, you will not compete with many rivals.*
- *Deeper insights in specific industry are still indispensable.*

Machine Learning Models: 10-fold CV

Action

Conduct One-Hot Encoding on Category.

Model used:

1. Ordered Logistic Regression
2. Linear Discriminant Analysis
3. Classification Trees
4. k-Nearest Neighbors (KNN)
5. Support Vector Machines
6. Bayesian GLM
7. Random Forest
8. Gradient Boosting

Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
orderlog	0.3441558	0.3568435	0.3642274	0.3617475	0.3683700	0.3714286	0
lda	0.3648124	0.3790966	0.3851122	0.3840354	0.3895558	0.3976684	0
cart	0.5536869	0.5681011	0.6114486	0.6132814	0.6622401	0.6740260	0
knn	0.7037516	0.7117805	0.7281536	0.7279156	0.7412464	0.7558442	0
svm	0.3880983	0.3953317	0.4018086	0.4016614	0.4066764	0.4158031	0
logi	0.2616580	0.2732861	0.2856214	0.2815387	0.2894567	0.2962484	0
rf	0.7797927	0.7834671	0.7926060	0.7933406	0.8019390	0.8090909	0
xgb	0.7655440	0.7812706	0.7849754	0.7899731	0.8023366	0.8150065	0

Kappa

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
orderlog	0.13940775	0.1532558	0.1629287	0.1596966	0.1669348	0.1700572	0
lda	0.18635940	0.2062153	0.2123604	0.2105968	0.2153024	0.2273170	0
cart	0.43352281	0.4519320	0.5082475	0.5108662	0.5748667	0.5891911	0
knn	0.63077459	0.6407065	0.6619165	0.6613293	0.6777348	0.6952395	0
svm	0.20946961	0.2161790	0.2245578	0.2249220	0.2315387	0.2416040	0
logi	0.09222187	0.1056743	0.1225169	0.1167095	0.1266969	0.1322278	0
rf	0.72597820	0.7311959	0.7427092	0.7432632	0.7532196	0.7626228	0
xgb	0.70830219	0.7287007	0.7334286	0.7391735	0.7543938	0.7697632	0

Result

Random Forest Model has the highest Kappa and Accuracy.

Model Refinement: Strategies for Improving

Introduce Useful Predictors

- *Sentiment Analysis*

Remove Useless Predictors

- *Reduce Model AIC - Stepwise Method*

Refinement Effect Analysis

- *Out-of-sample Testing*

Text Analysis: Reviews Sentiment Description

Top5 Positive Reviews

Polarity score	*Description*	*Installs.level*
5.33	He's amazingly gifted, caring, enlightening, uplifting & generous…	3
4.16	Very good, very convenient, easy to use, very beautiful, very good…	4
4.05	Such nice , great, awesome, amazing, wonderful, interesting, beautiful game I like…	5
4.00	Great easy benefits great.	3
3.85	Great Great extremely user friendly	3

Top50 Positive Words

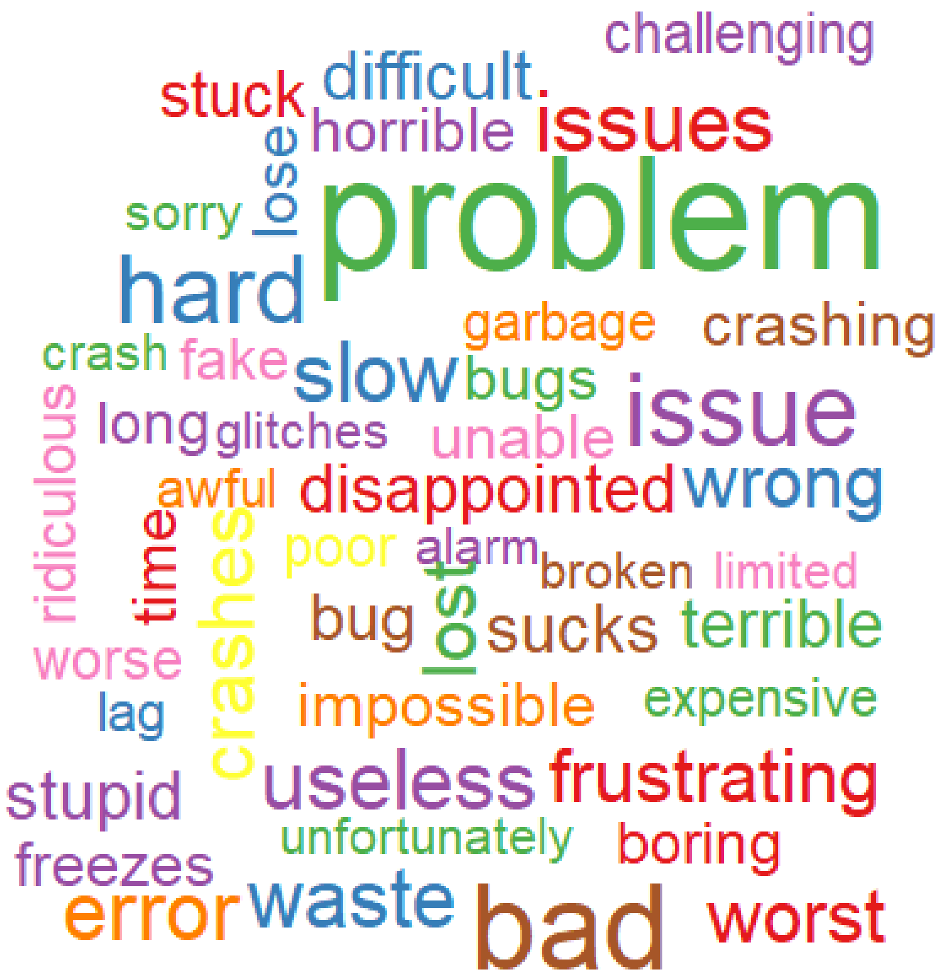
Text Analysis: Reviews Sentiment Description

Hypothesis
***Sd** of Polarity score could be an important variable to control.*

Top5 Negative Reviews

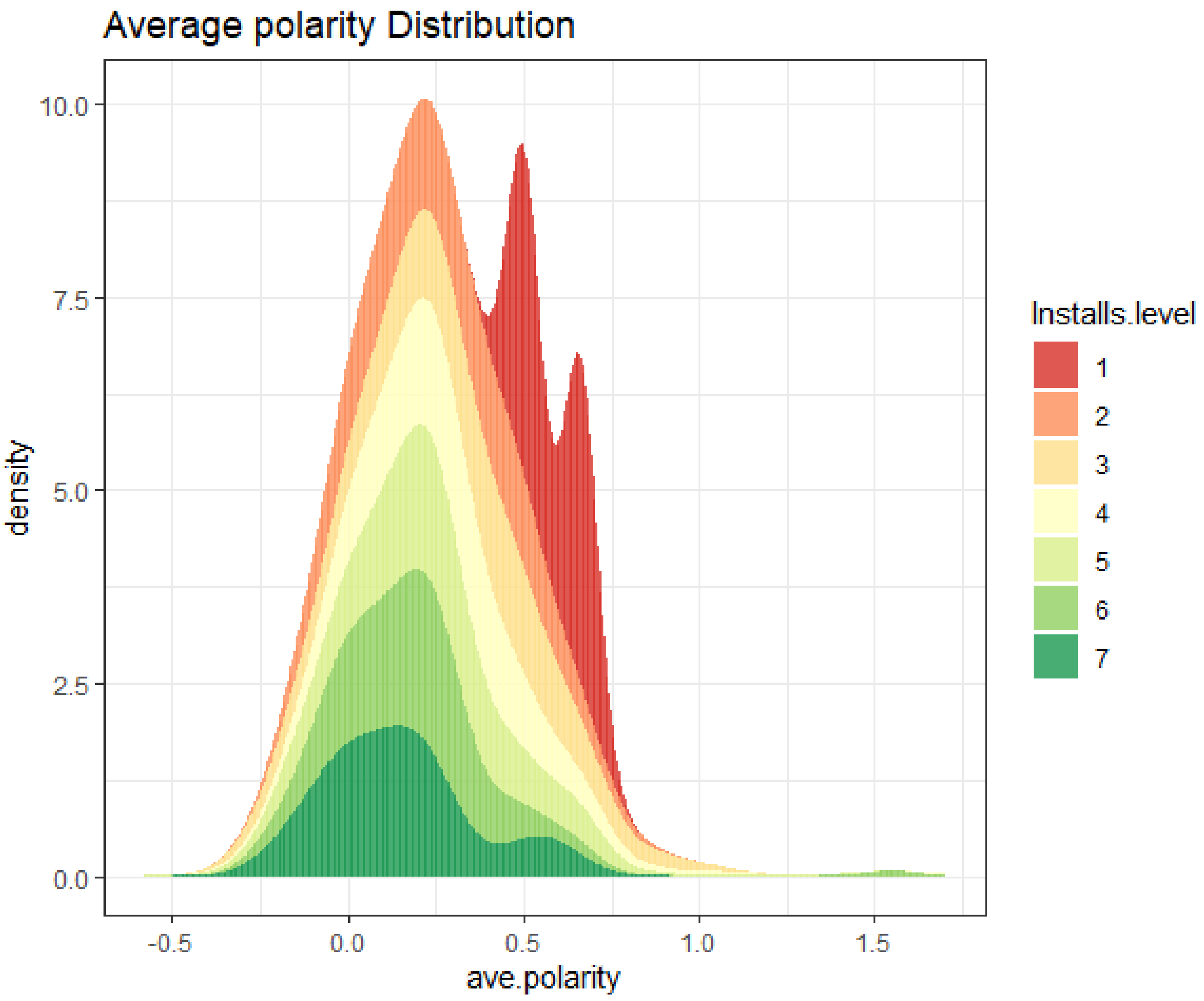
Polarity score	*Description*	*Installs.level*
-2.92	This game addicting, quite challenging times also extremely worried. There's move limits…	5
-2.83	Bad bad bad bad bad bad bad bad	4
-2.67	Very slow crashing app. Very bias articles, bad journalism.	5
-2.28	Unstable, extremely slow unresponsive exits wants to! So Crap app!!!!	4
-2.26	It's nonsense.... Just pictures more... SCAM SCAM SCAM!!!	4

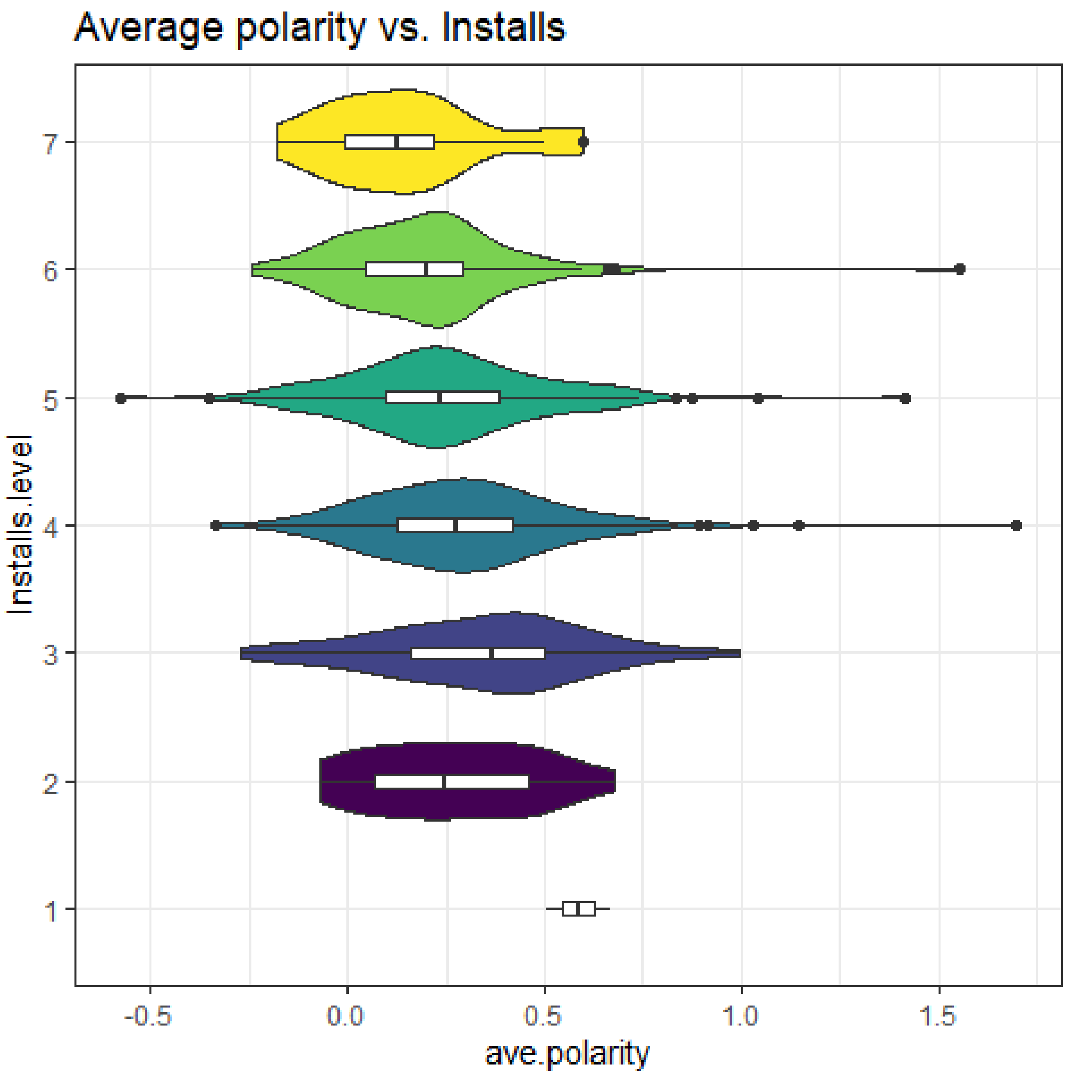
Top50 Negative Words



Text Analysis: Sentiment Variables vs. Installation

Variable: ***ave.polarity vs. Installs***





Model Revision: Introducing Sentiment Variables

Action01

NAs Imputation

- ***ave.polarity*** *Impute 0 as Neutral Sentiment*
- ***sd.polarity*** *Impute based on Original Distribution*

Action02

StepAIC Method

- *Use stepAIC() from MASS*
- *"both" way*

Results

OLR Model	AIC	Accuracy	Kappa
Old model	29514.77	0.362	0.160
New model	29306.31	0.368	0.178

Linear Regression Model	R2	RMSE
Old model	0.297	3.639
New model	0.313	3.591

Sentiment Variable	t value	Odds ratio	Probability of 10x installations
ave.polarity	13.78	12.39	1139.34%
sd.polarity	-3.00	0.66	-33.81%

Model AIC have already reached the lowest level.

Sentiments are so powerful!

Model Revision: Updated ML Models

Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
orderlog	0.3570505	0.3604275	0.3664073	0.3693913	0.3758085	0.3945666	0
lda	0.3673997	0.3918263	0.3946884	0.3959562	0.4018112	0.4170984	0
cart	0.5536869	0.5681011	0.6114486	0.6132814	0.6622401	0.6740260	0
knn	0.6977951	0.7162752	0.7297484	0.7260987	0.7375935	0.7402597	0
svm	0.3932730	0.4086195	0.4102401	0.4098225	0.4139395	0.4178525	0
logi	0.2590674	0.2749086	0.2843261	0.2816696	0.2901035	0.2962484	0
rf	0.7733161	0.7822023	0.7940428	0.7937326	0.8064074	0.8111255	0
xgb	0.7707254	0.7787844	0.7909338	0.7932064	0.8064710	0.8214748	0

Kappa

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
orderlog	0.16070957	0.1671604	0.1745960	0.1780079	0.1861326	0.2100636	0
lda	0.18986813	0.2209106	0.2239378	0.2260928	0.2327192	0.2523582	0
cart	0.43352281	0.4519320	0.5082475	0.5108662	0.5748667	0.5891911	0
knn	0.62477755	0.6469489	0.6637423	0.6591556	0.6720665	0.6772368	0
svm	0.21655123	0.2350007	0.2357792	0.2355961	0.2408423	0.2460939	0
logi	0.08851207	0.1068062	0.1206720	0.1164266	0.1269858	0.1317279	0
rf	0.71908360	0.7296376	0.7445376	0.7437675	0.7593588	0.7643371	0
xgb	0.71457993	0.7252592	0.7406609	0.7431081	0.7599813	0.7774259	0

Results

Performance improved

- ***OLR Model***
- ***LDA Model***
- ***SVM***
- ***Bayesian GLM***
- ***Random Forest***
- ***Gradient Boosting***

Performance remained the same

- ***Tree Model***

Performance declined

- ***KNN Model***

Model Revision: Out-of-sample Testing

ML Model	Training Accuracy	Testing Accuracy
Random Forest	0.7937	0.7934
Gradient Boosting	0.7909	0.7923

Install.level	Sensitivity		Specificity	
	Random Forest	Gradient Boosting	Random Forest	Gradient Boosting
1	0.9193	0.9212	0.9627	0.9684
2	0.7057	0.7425	0.9484	0.9478
3	0.7658	0.7468	0.9385	0.9416
4	0.7422	0.7422	0.9358	0.9416
5	0.7695	0.7565	0.9672	0.9629
6	0.78205	0.73077	0.99405	0.99188
7	0.25	0.25	1	0.999479

Results

- ***For both models, training and testing accuracy are very close. Overfitting does not exist.***
- ***Random Forest Model has higher training and testing accuracy, but it does not show absolute superiority.***
- ***Both models exhibit excellent specificity in each install group, meaning they can avoid Type I error very well.***
- ***RF Model shows better performance in identifying Apps in group2, and GB Model did a better job in identifying Apps in group6***

Key Takeaways:

1. Type (paid or free) is the most significant negative predictor. Don’t charge at least at the beginning.

2. Accommodation to different devices and system versions are important, but unfortunately, not every developer can do that.

3. Be cautious about the industry your want to enter. If you are not an expert with industry insights, don’t choose a tough road to start your App.

4. Sentiment Analysis should never be dismissed in App market research. Controversies or debates about your App are not always the bad things.

5. Choose proper ML Models based on your major concerns.

Predicting Apps Installations from Google Play Store through Machine Learning Models

Yiheng An
UCLA Extension Data Science Intensive