# Regression: Predicting Household Size

Samson Huynh, Su-Han Tan, Jolina Hor, Eric Du, Irisa Le, Flora Ling

## 1. Introduction

The objective of this project is to predict the household size of Amazon customers based on their purchase history and limited demographic survey data. Understanding household size is critical for e-commerce platforms to optimize recommendation engines. For example, suggesting bulk "family-size" products versus single-serve items.

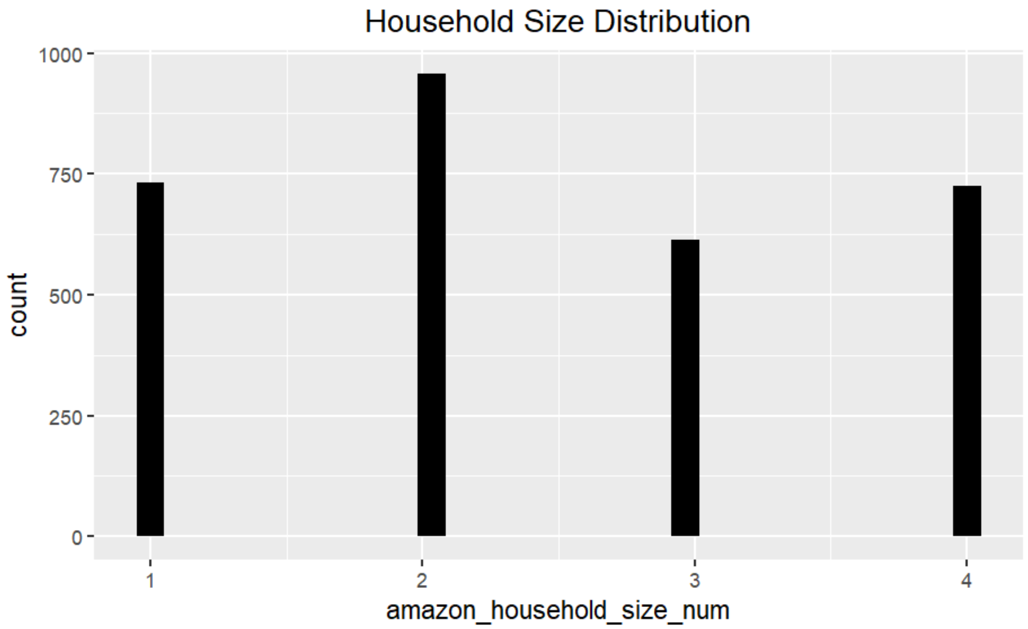
Our hypothesis is that household size is not determined solely by total spending, but by the composition of the basket. We expect that specific "life-stage" indicators, such as the presence of baby products, toys, and school supplies, will be stronger predictors than raw volume of a larger household with children (Møller et al., 2023). Additionally, we anticipate that larger households will exhibit higher consumption rates of recurring staples like groceries and paper goods (Ren et al., 2018) . This analysis utilizes a dataset of Amazon purchase logs and survey responses to build a regression model that accurately estimates the number of occupants in a household.

## 2. Exploratory Data Analysis

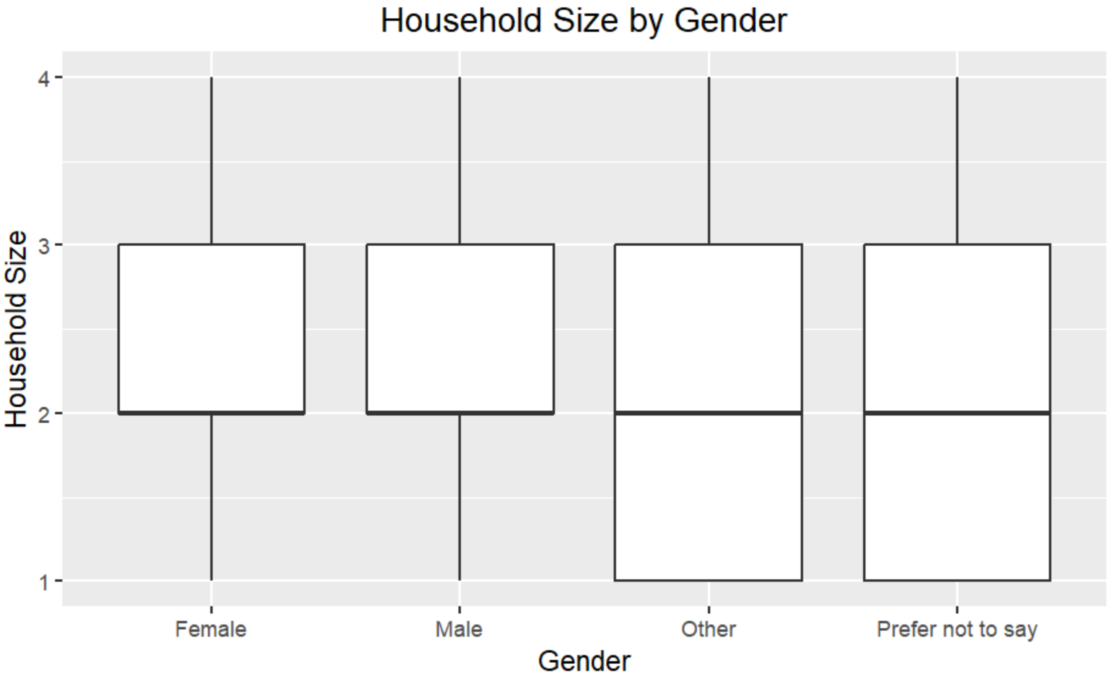
Overview: Because we had two very large datasets to work with, we wanted to understand the distributions of the data and relationships between various variables. Certain variables that seem to be statistically different across the 4 household sizes will be useful to include into the final model. Lastly, EDA helped support any initial hypotheses on features that may need to be engineered to improve our model.

### 2.1 Survey Dataset

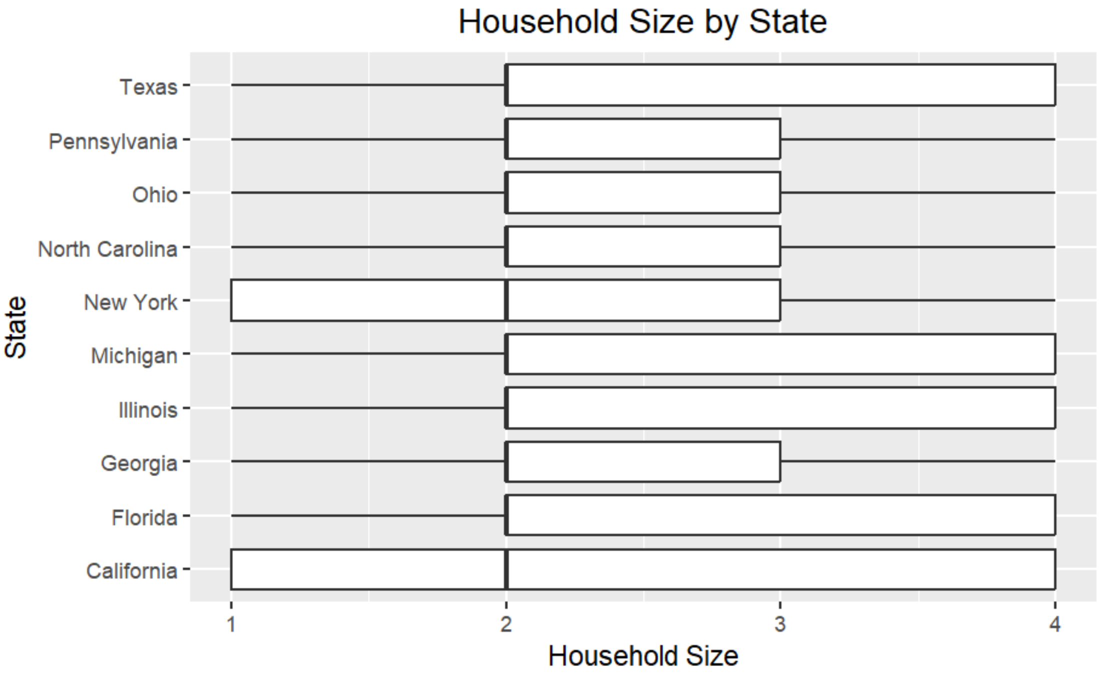
- From the survey dataset, we looked at the state and gender variables, as this is the only information that will be available at sign-up. The visualizations helped support any initial hypotheses on what variables might be useful for the model.



There seems to be mostly 2 person households, then 1 and 4 person households, and lastly 3 person households. We should figure out what types of purchasing behavior is correlated with the different household sizes.



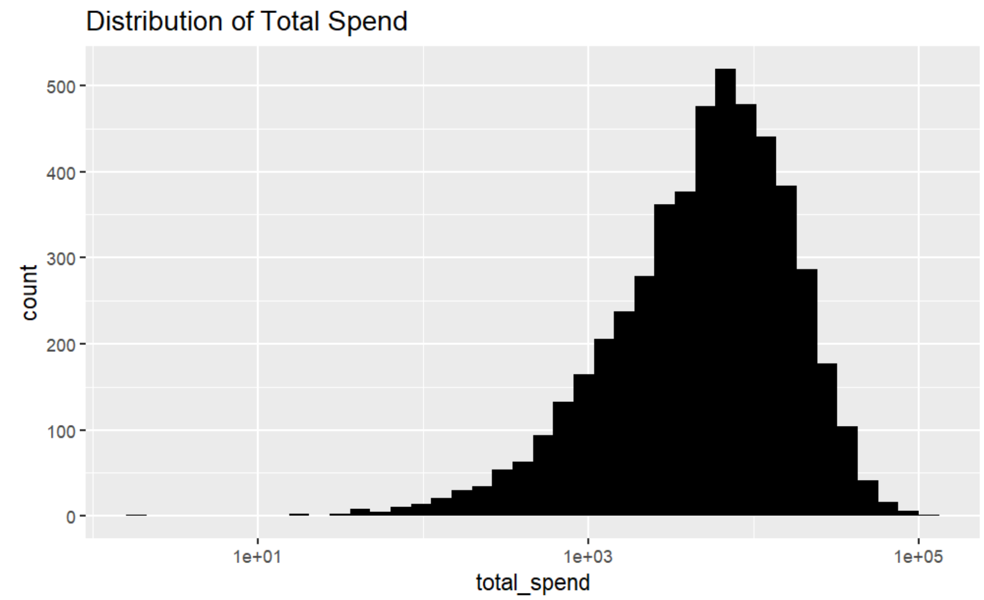
For this graph, we can tell there is barely a difference in distribution of household size for males and females. Hence, gender may not be a very useful predictor in our models.



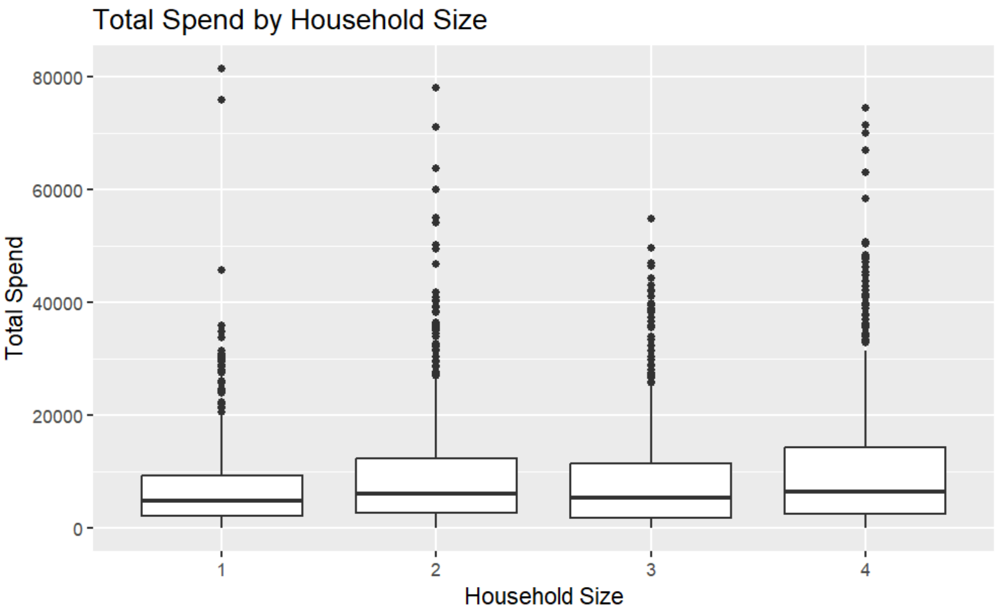
For this graph, we can tell there is a slight difference in distribution of household size for the top 10 states. However, it is not very significant, so it will probably not be a helpful predictor in our final model.

### 2.2 Amazon Purchases Dataset

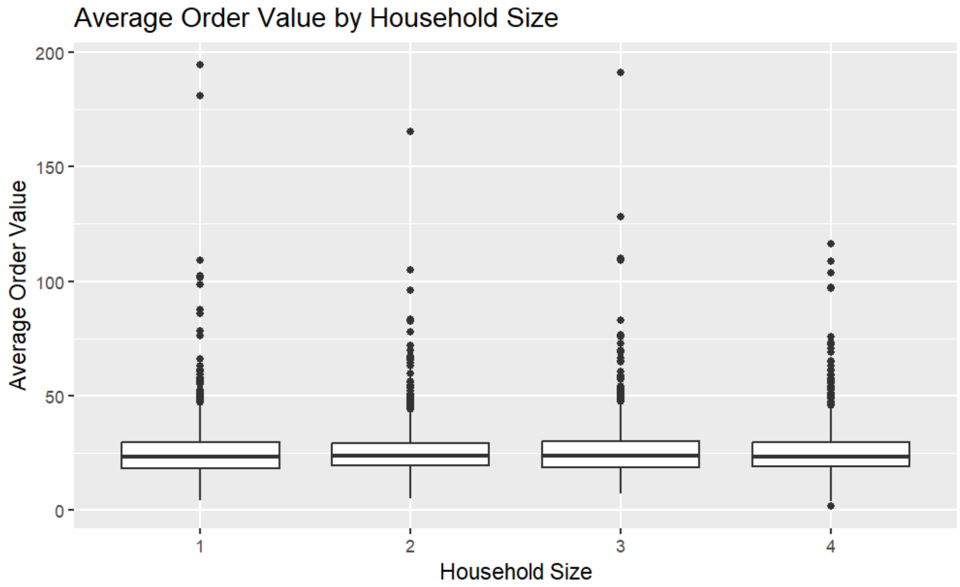
- From this dataset, we first joined it with the survey data to match a household size with each user, and then extracted information from the different purchase based features we engineered. We looked at the spending habits, including total spend, number of orders, time between orders etc. to understand any potential relationships that would help us understand what predictors would be best for our model.



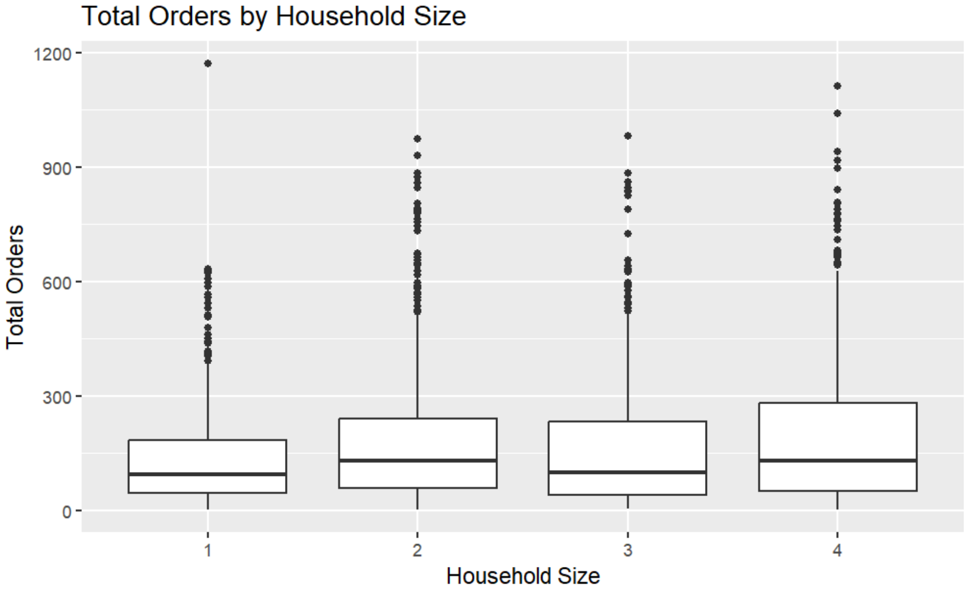
From this visualization, we can see that the distribution of total spendings on Amazon looks relatively left skewed, which makes sense given the exorbitant amount of purchases on the platform. We performed a log transformation for the total spend variable.



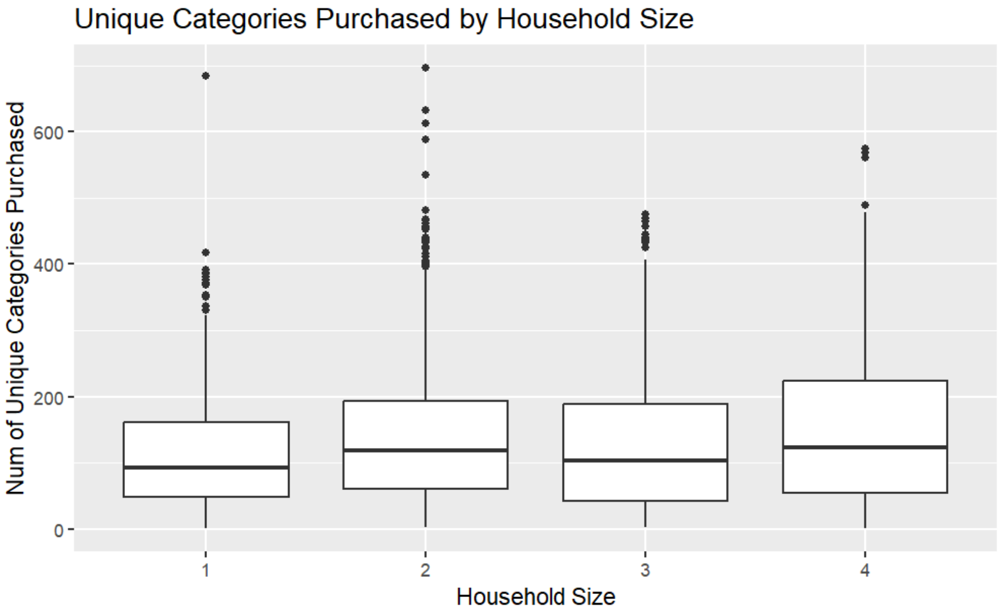
Across the different household sizes, the distribution of total spend is not as different as one may perceive. For example, there are a good amount of one person households which spend a lot, so this is something to take into consideration during variable selection.



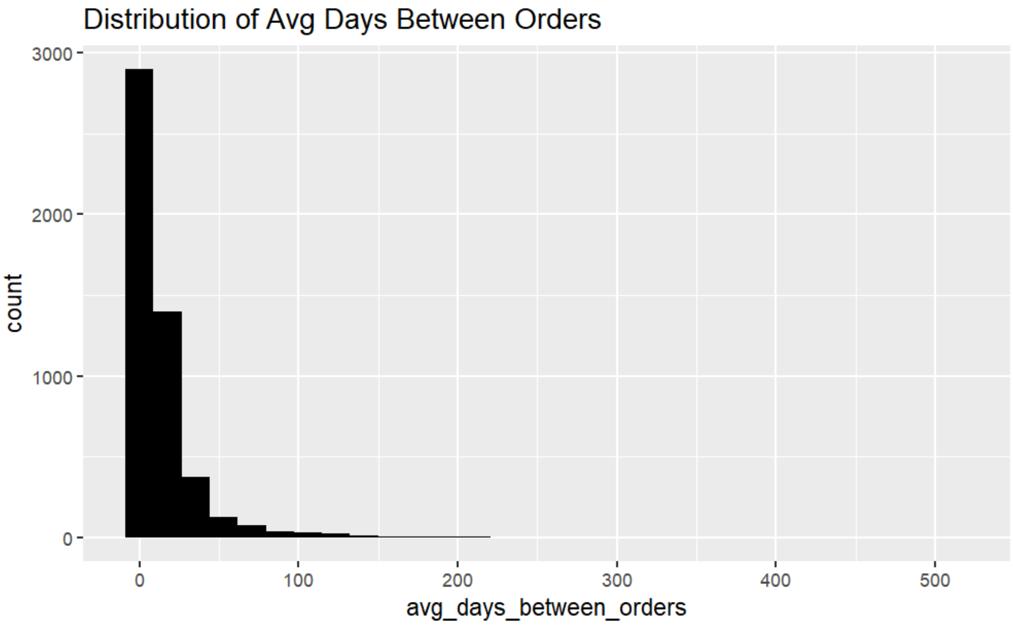
The average order value per household is also not starkly different across the different household sizes, with some very high outliers in the 1 person households. Average order value will likely not be very helpful for our model.



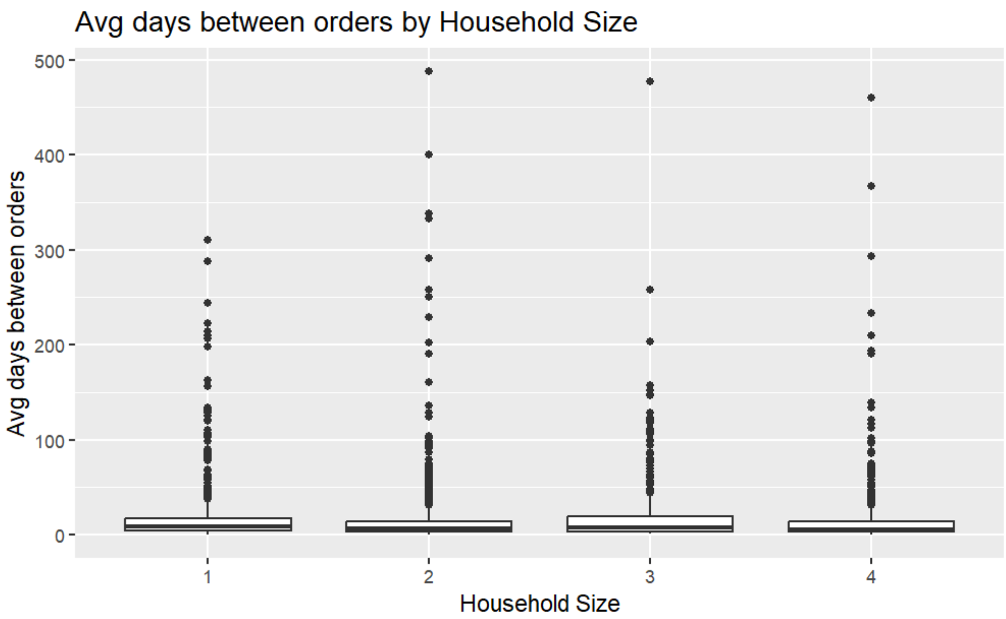
Total order distribution appears statistically different across the different household sizes, with a lot of outliers for two-person households. This could be due to new couples ordering lots of items for a new living space, so the total number of orders will probably be a helpful predictor.



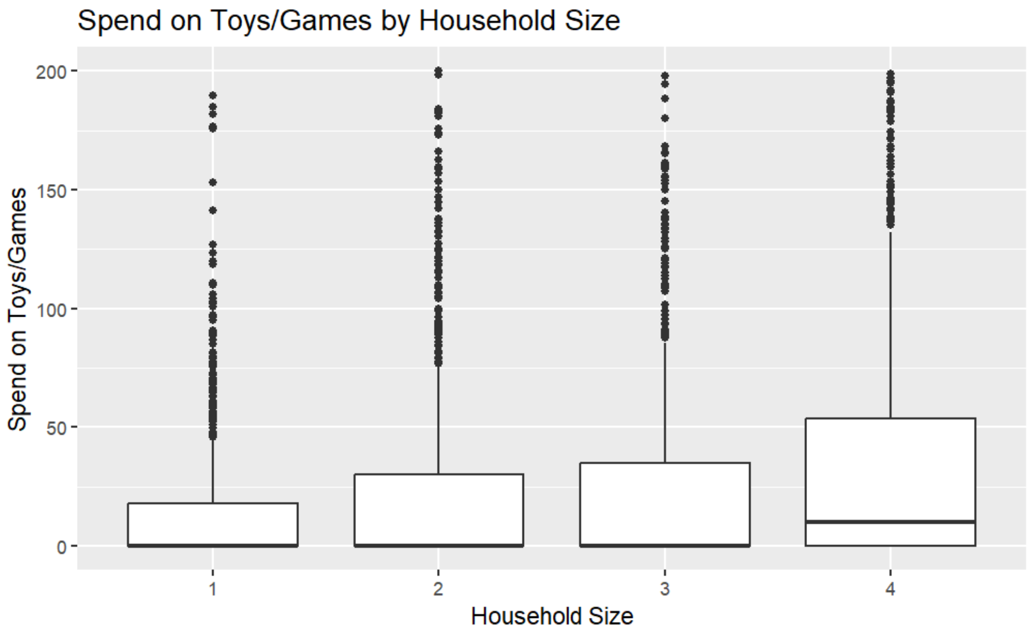
The number of unique categories purchased by a household also seems starkly different across the different sizes with a higher median for 2 and 4 person households, and many outliers for 2 person households. Unique categories will probably be a helpful predictor.



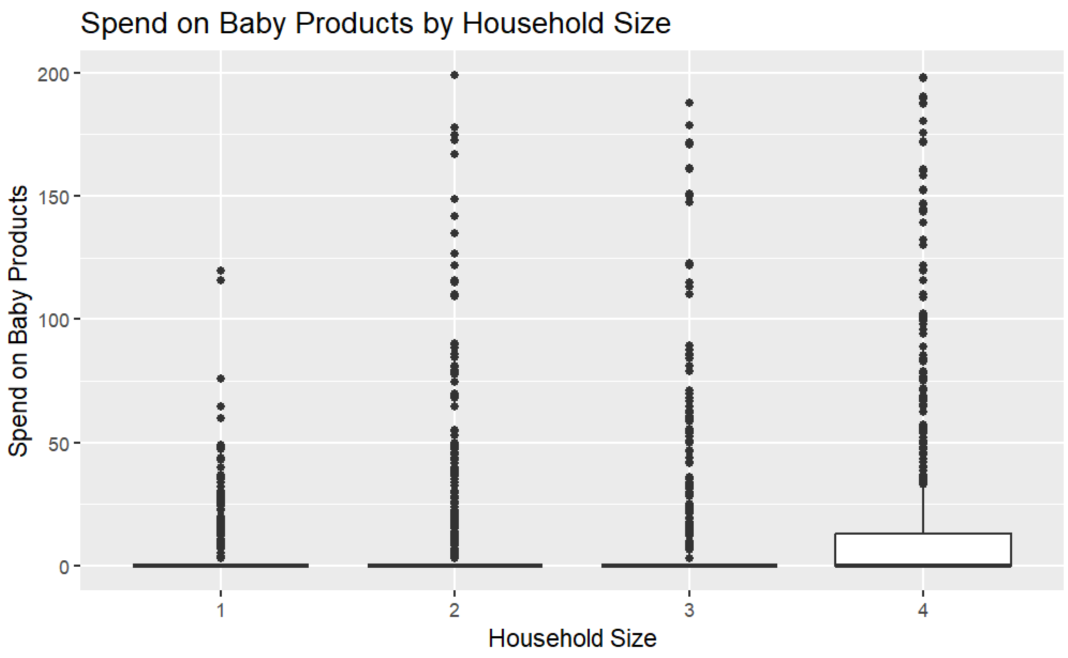
The distribution looks right skew, which implies that purchasing behavior seems to be relatively consistent throughout the year with orders spaced relatively close to each other.



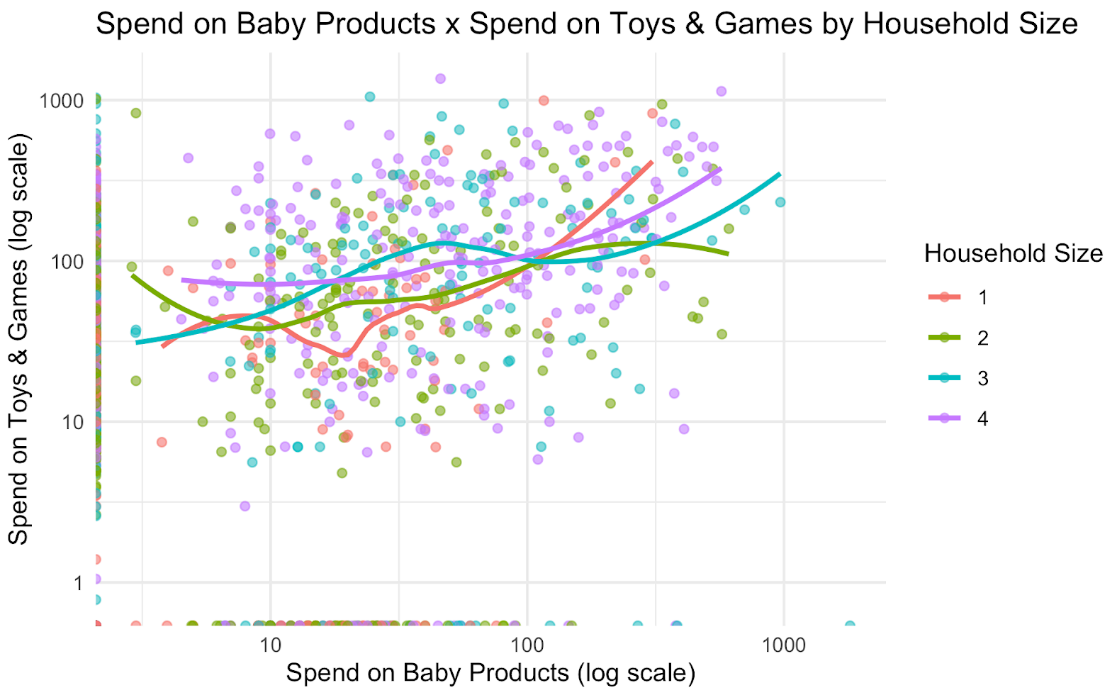
The distribution is similar across the different household sizes, with more outliers for 2-person households similar to some of our previous variables. Smaller households may tend to have more sporadic purchasing behavior, so average time between orders could be useful in the model.



Based on this visualization, we can see that households of 3 or 4 tend to spend more on toys and games. This affirms our initial hypothesis, so this will probably be a strong predictor for household size.



Based on this visualization, we can see that there are many 4 person households that tend to spend more on baby products. Families of 4 likely have at least 1-2 children, so this will also be a helpful predictor for our model.



Based on this visualization, predictors relating to life-stage spending patterns (e.g. baby and toy purchases) strongly differentiate household sizes.  Households of 3-4 display clear upwards trends, while 1-2 person households remain low, further indicating that child-related categories will likely be influential predictors for our model.

## 3. Preprocessing

We performed extensive preprocessing to transform raw transactional data into behavioral features:

1. **Data Cleaning**: Survey demographic fields were converted from text to numeric. Purchase dates were parsed to calculate the time span of account activity. We did not include predictor columns where the majority of entries were NA, (i.e. variables besides state and gender from the survey data).

2. **Joining the Datasets:** Because we had two disparate datasets, we decided the best strategy was to first convert the purchases dataset from a long to wide format. Afterwards, we used library (dplyr) to perform a left join on the responseID column to the important columns from the survey dataset (i.e. household size, gender, state).

3. **Feature Engineering**: We aggregated the raw purchase log (one row per item) into a user-level summary (one row per user). We created "Ratio Features" (e.g., pct_grocery, pct_toys) to normalize behavior, ensuring that a high-spending single person buying expensive groceries wasn't confused with a large family buying bulk cheap groceries and to understand features that could indicate a larger household size.

4. **Dimensionality Reduction (PCA)**: The dataset contained 1000 unique product categories (filtered from the top categories). Using them all would cause overfitting and multicollinearity. We applied Principal Component Analysis (PCA) to the item counts, extracting the top 10 components (PC1–PC10). These components capture latent "lifestyle" variances without adding noise.

5. **Log Transformations**: Financial variables (total_spend, total_items) were left-skewed (long tail of low spenders). We applied a log1p transformation to linearize these relationships for the regression models.

6. **Oversampling**: The training data was biased toward 2-3 person households. We created a train_df_balanced by duplicating rows for household size 1 (2x weight) and household sizes 4+ (4x weight). This forced the models to pay attention to the minority classes rather than defaulting to the mean.

## 4. Candidate Models / Model Evaluation / Tuning

We evaluated **five** distinct modeling approaches to solve this regression problem.

1. **Linear Regression:** A standard Ordinary Least Squares (OLS) model (regr.lm). This served as our baseline to test if the relationship between purchases and household size was linear. It performed poorly due to the non-linear nature of purchasing behavior.

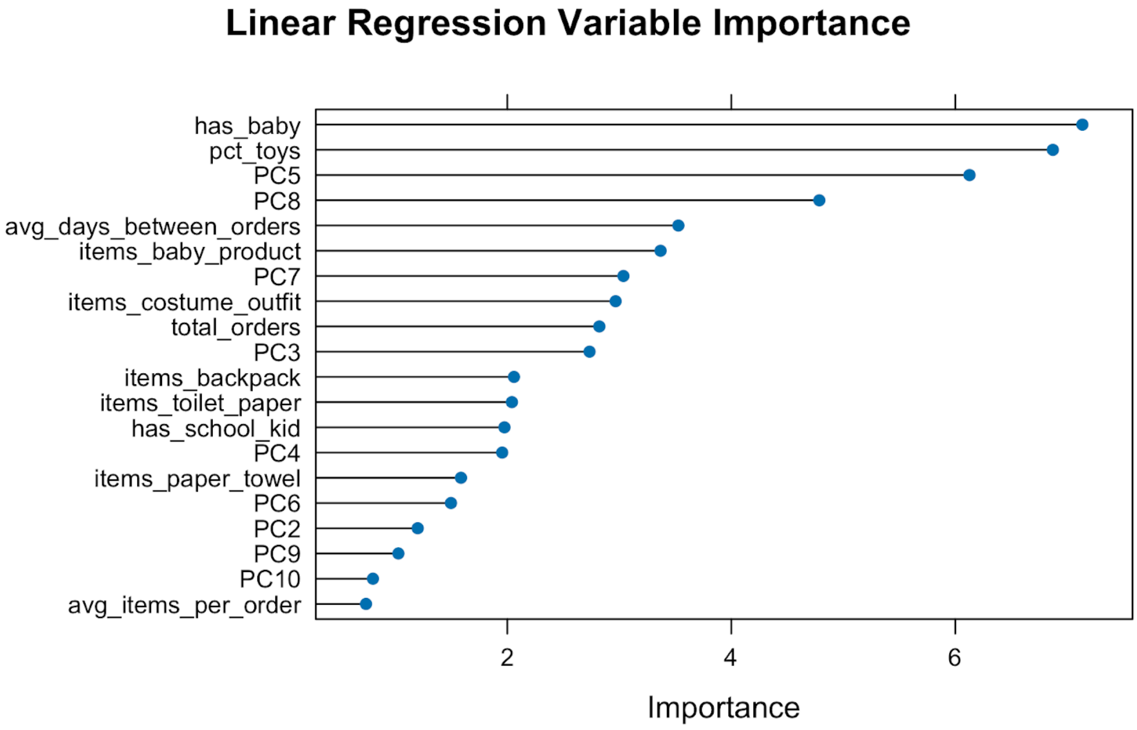


Figure: Linear Regression Variable Importance Plot

2. **Lasso Regression:** We utilized Lasso Regression (regr.cv_glmnet with alpha=1) to perform feature selection. By penalizing the absolute size of coefficients, Lasso makes

irrelevant feature coefficients zero. We tuned the regularization parameter λ using 5 fold cross-validation, selecting a value of approximately 0.0066 to minimize error.

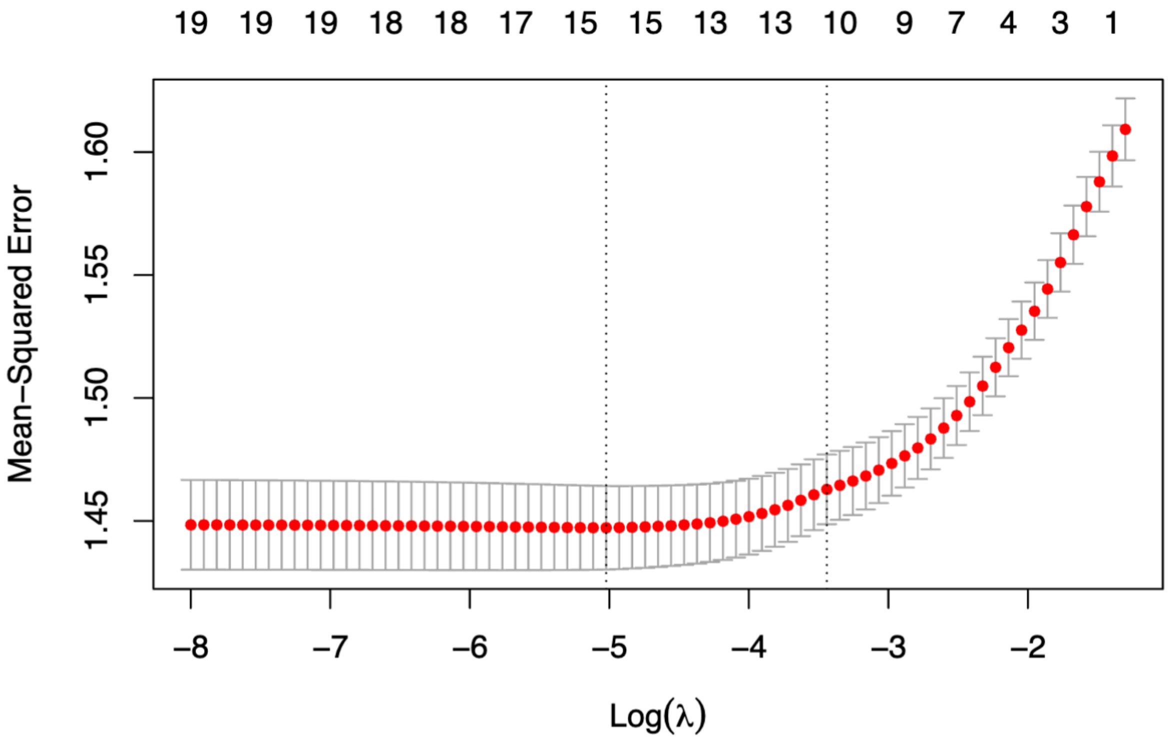


Figure: Finding Most Optimal Lasso Lambda Value

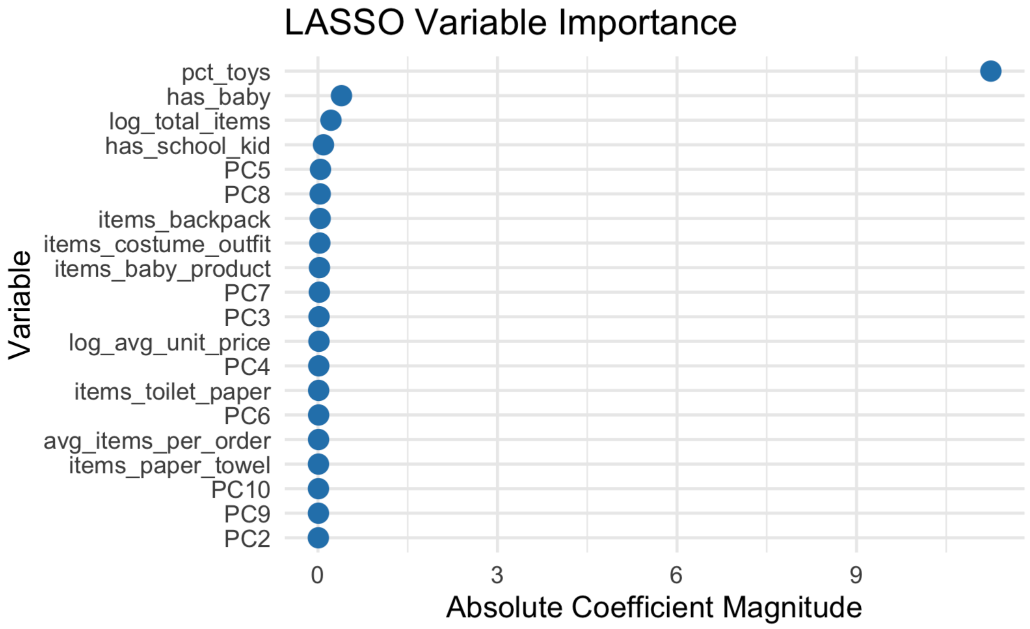


Figure: LASSO Variable Importance Plot

3. **Ridge Regression:** We also tested Ridge Regression (regr.cv_glmnet with alpha=0). Unlike Lasso, Ridge does not eliminate variables but shrinks coefficients to prevent overfitting from multicollinearity. We allowed the model to automatically select the optimal λ (lambda.min).

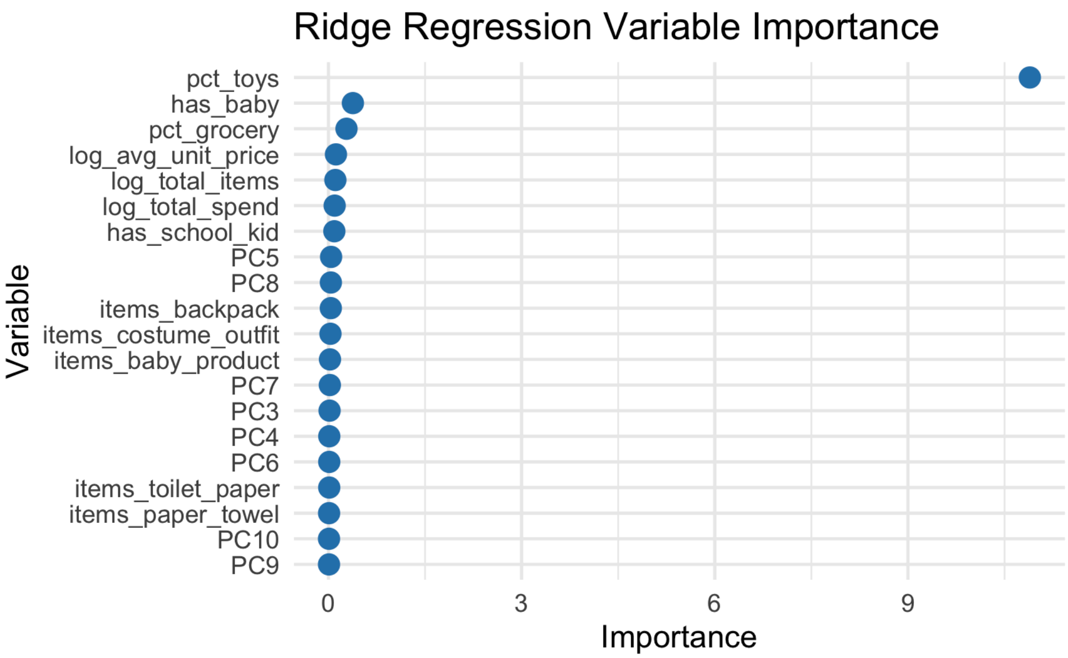


Figure: Ridge Regression Variable Importance Plot

4. **Random Forest:** A non-linear ensemble method (regr.ranger) using 100 decision trees. Our hyperparameter tuning process led us to set mtry to 5 and min.node.size to 10. Random Forest was highly effective at capturing interactions.
   - Variable Importance Insights: The variable importance plot revealed that PC4, PC5, and PC3 were dominant predictors. Our findings suggests that overall volume and the primary lifestyle component (PC4) are the strong signals for household size, outweighing specific category flags like has_baby and has_school_kid.

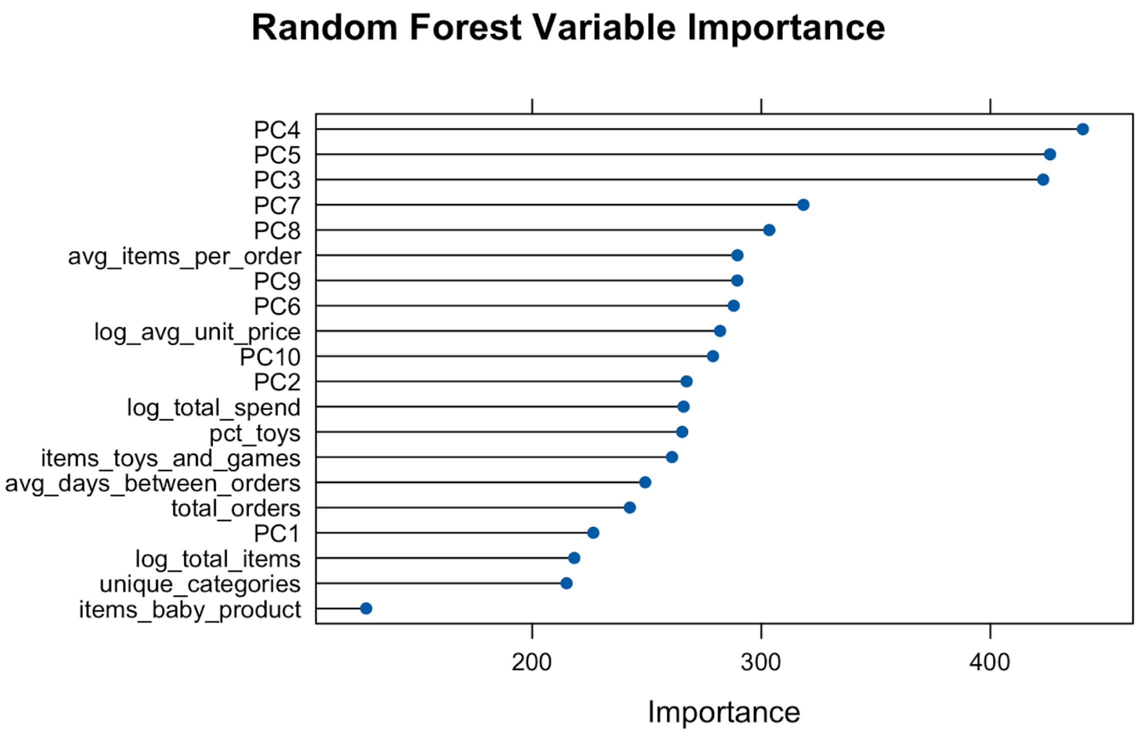


Figure: Random Forest Variance Importance Plot

We utilized this plot to then choose a smaller subset of predictors to test more models and see if that would improve overall performance.

5. **XGBoost:** A Gradient Boosting Machine (regr.xgboost). We used a pipeline to one-hot encode categorical variables before passing them to the learner. We used the default parameters provided by the xgboost package (using reg:squarederror objective) to make a strong boosting baseline.

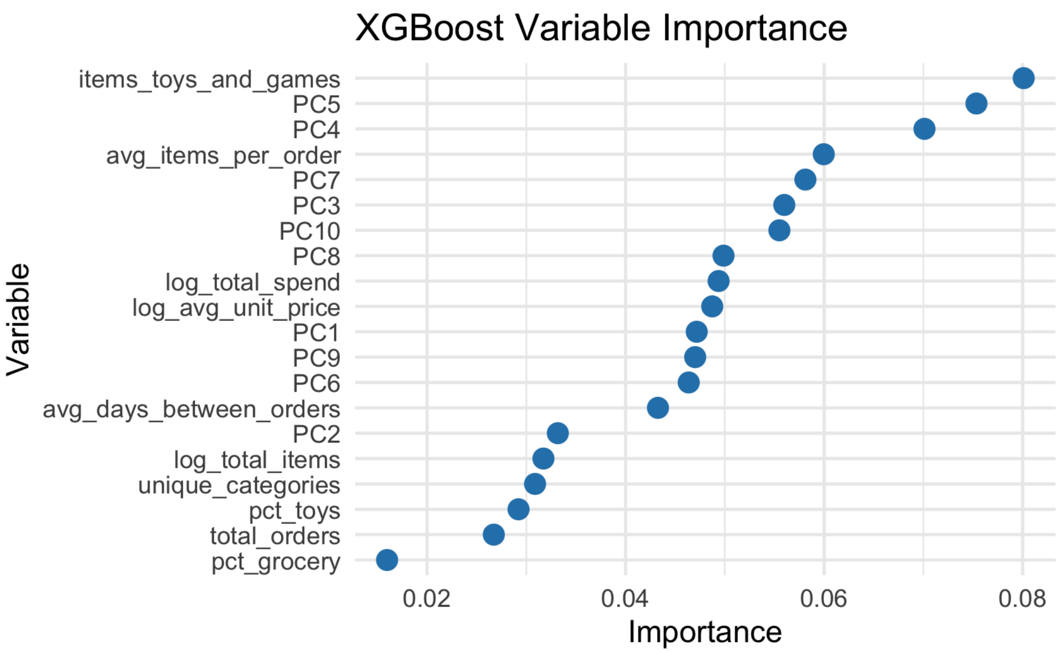


Figure: XGBoost Variable Importance Plot

**Model Summary Table**

| ***Model Identifier*** | ***Type*** | ***Variables*** | ***Hyperparameters*** |
|---|---|---|---|
| lrn_lm | Linear Regression | All Predictors | None |
| lrn_lasso | Lasso Regression | All Predictors | alpha = 1, Lambda.min = 0.006586 |
| lrn_ridge | Ridge Regression | All Predictors | alpha = 0, Lambda.min = 0.006586 |
| lrn_rf | Random Forest | All Predictors | ntree = 100, mtry = 5, nodesize = 10 |
| lrn_xgb | XG Gradient Boosting | All Predictors | Default (one-hot encoding) |

**Model Evaluation and Tuning**

We implemented a 5-fold cross validation in our code that would essentially split the training data into 5 parts, train the model on 4 of those, and validate on the 5th split, averaging performance across all of them. The k-fold CV was implemented across all of the models to help improve model performance. In terms of hyperparameter tuning, we tried different hyperparameters for our XGBoost model, but it actually made the RMSE and r-squared worse. Hence, we decided to keep our original XGBoost model.

The primary metric was RMSE (Root Mean Squared Error), but we also monitored Exact Match Accuracy (percentage of predictions that rounded to the correct integer). The Random Forest provided the most stable results, but XGBoost offered the best handling of the count-based nature of the target. We found that the Linear models (1 & 2) had significantly higher RMSE (1.10) compared to the tree-based models (1.02).

| ***Model Identifier*** | ***RMSE*** | ***R^2*** |
|---|---|---|
| lrn_lm | 1.199 | 0.106 |
| lrn_lasso | 1.199 | 0.106 |
| lrn_ridge | 1.199 | 0.106 |
| lrn_rf | 0.622 | 0.759 |
| lrn_xgb | 0.529 | 0.825 |

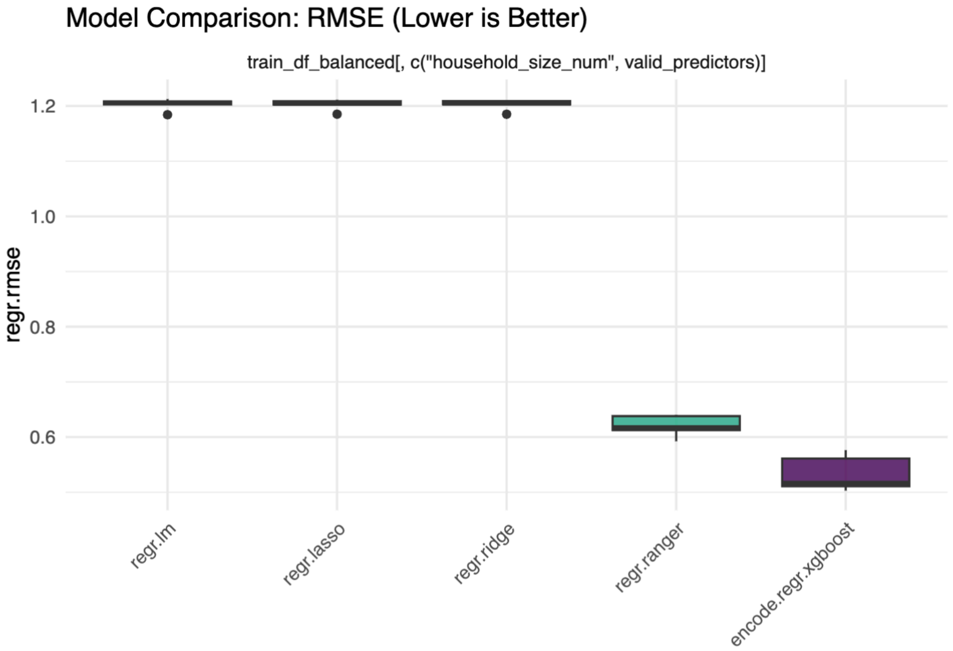


Figure: RMSE Model Comparison

We can see that the XGBoost model has the best RMSE, with Random Forest right behind.

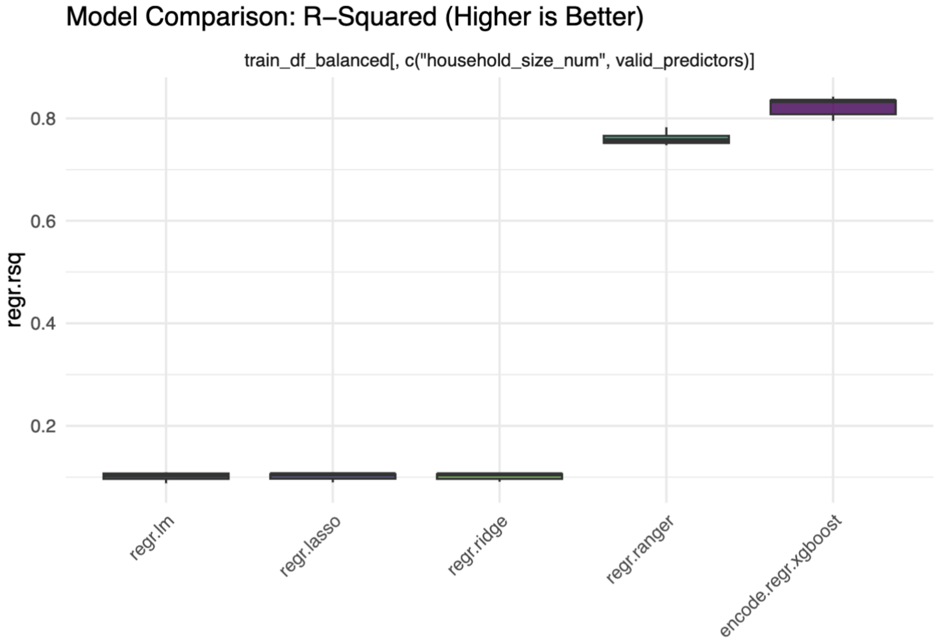


Figure: R-Squared Model Comparison

We can see that the XGBoost model also has the best R^2, with Random Forest right behind.

**Individual Prediction Examples**

***Linear Regression***

Response Id: R_2UbJL30HRjK1sdD

- Actual Household Size = 4, Predicted Household Size: 3.47
- has_baby = 1, pct_toys = 0.01524256

For this household, the linear model mildly underestimates the true size of 4, with a prediction of size 3.47. A key factor influencing this could be the presence of child-related indicators such as has_baby and a non-zero pct_toys, pushing the prediction value upwards because they strongly correlate with larger household sizes.

Response Id: R_2bOT9I7PH69jROd

- Actual Household Size = 1, Predicted Household Size: 2.57
- has_baby = 0, pct_toys = 0

This case represents an overestimation by the linear regression model. Although the true household size is 1, the model predicted a value above 2.5. Top predictors has_baby and pct_toy both equal to 0, showing that the household has no child-related indications. However, the addition of PC5, PC8 (which are uninterpretable), and additional predictors considered could have inflated the results upward even when child-related variables are absent.

***Lasso Regression***

Response Id: R_3EF584Zf9pAwkir

- Actual Household Size = 3, Predicted Household Size = 2.66
- Pct_toys = 0, has_baby = 0, log_total_items = 5.337538, has_school_kids = 1

The Lasso model slightly underestimates this household's true size of 3, instead giving it a prediction of 2.66. Because the household appears to have no spending in toy-related categories or no baby related purchases, and that these are the strongest signals of larger families, the absence of the two may have led to a prediction slightly below value. Nonetheless, this is still a good estimation.

Response Id: R_24f9hBh5NSWGX7k

- Actual Household Size = 2, Predicted Household Size = 2.55
- Pct_toys = 0.005006654, has_baby = 0, log_total_items = 5.31812, has_school_kids = 0

For this household, the model slightly overestimates the true size of 2 to 2.55. The model likely increased its prediction because of moderate overall purchase value and pct_toys, which suggests some indication of the presence of a child.  However because the household has no baby or school kids indicator, the predicted value was not pushed up too high.

***Ridge Regression***

Response Id: R_1LBvKYfg8hgoloJ

- Actual Household Size = 4, Predicted Household Size = 3.05
- Pct_toys = 0, has_baby = 0, pct_grocery = 0, log_avg_unit_price =  3.766043

It appears that the Ridge model underestimates the true household size of 4, instead, predicting 3.05.  Because we have no life-stage indicators, the model has little evidence to infer a large household.  Due to the mechanics of ridge regression, predictors like log_avg_unit_price and others may have moderately increased the prediction value, but it is not enough to reach the true household size.

Response Id: R_2us4qPSxiD4clPq

- Actual Household Size = 2, Predicted Household Size = 2.75
- Pct_toys = 0, has_baby = 0, pct_grocery = 0, log_avg_unit_price =  3.500664

This case represents an overestimation, in which the predicted household size is predicted higher by a value of 0.75. Like the previous case, there are no child related features, causing the predicted value to be pushed down. However, a moderately high log_avg_unit_price (and possibly other features not listed), may have pushed the prediction upward slightly.

***Random Forest***

Response Id: R_3HBSzChLJ0THGv0

- Actual Household Size = 4, Predicted Household Size = 3.81

Since the top predictors for the Random Forest model come from PCA components, it is hard to interpret the specific drivers behind the prediction. Despite this, the model's predicted household size of 3.81 (though underestimated) is reasonably close to the true value of 4, suggesting that the patterns captured by the PCA features strongly resemble those of larger households.

Response Id: R_1jImmHuCDaof0wB

- Actual Household Size = 1, Predicted Household Size = 1.52

The comments mentioned in the case above are applicable to this case too, although it appears the Random Forest model slightly overestimates the household size for this observation. This may be due to the fact that this household shows some purchasing patterns captured through the PCA dimensions that overlap with those of slightly larger households.

***XG Gradiant Boosting***

Response Id: R_2rJyfdZGQD2rif2

- Actual Household Size = 4, Predicted Household Size = 4
- Items_toys_and_games = 10, avg_items_per_order = 2.044218

XG-Boost correctly predicts this household as size 4. The presence of high toy-related purchases may have provided a strong indicator of children in the household, shown in general to be correlated with household sizes of 3-4. Because of this and the consideration of other predictors in the model, it may have pushed the predicted value higher.

Response Id: R_2zNpJuWE3ykIlGh

- Actual Household Size = 1, Predicted Household Size =  1
- Items_toys_and_games = 0, avg_items_per_order = 2.538462

In this case, XGBoost also performs well, correctly identifying a single-person household. The absence of toy and game purchases removes any indicator of children, and with the average items per order being slightly above 2, it is not high enough to indicate a larger family. Without these indications, the model appropriately assigns the lowest household size to this responseID.

# 5. Discussion of Final Model

Although the XGBoost model seemed to perform better in terms of RMSE and R-squared, it was prone to overfitting as we noticed when submitting our model to Kaggle. Hence, we selected Random Forest as our final model for generating predictions.

**Strengths:** The primary strengths of our model was the random forest has many trees, which allows for random selection of predictors and averaging of predictions across multiple trees.

**Weaknesses:** The model is a bit more difficult to interpret due to the Principal Components that we used as predictors in the model and additional feature engineering that was conducted. Random forests also tend to perform poorly for extrapolation of data. Additionally, Random Forest is really slow and computationally expensive, making our machine’s power consumption higher.

**Improvements**: To improve this model, we would benefit from temporal granularity. Currently, we aggregate all purchases. Knowing when items were bought (e.g., "Back to School" season vs. Christmas) could provide stronger signals for school-aged children. Additionally, geographic data (Zip Code) would allow us to infer household size based on local housing density (apartments vs. single-family homes).

## 6. Appendix

All Predictors:

```
{
  "log_total_spend", "log_total_items", "log_avg_unit_price",
  "total_orders", "unique_categories",
  "avg_items_per_order", "avg_days_between_orders",

  "PC1", "PC2", "PC3", "PC4", "PC5",
  "PC6", "PC7", "PC8", "PC9", "PC10",

  "has_school_kid", "has_baby",
  "pct_grocery", "pct_toys",

  "items_toys_and_games",
  "items_costume_outfit",
  "items_toy_building_block",
  "items_backpack",
  "items_baby_product",
  "items_grocery",
  "items_paper_towel",
  "items_toilet_paper"
}
```

Sources:

https://doi.org/10.1038/s41598-023-47534-6

https://link.springer.com/article/10.1140/epjds/s13688-017-0128-2