Datathon Submission Template

Retina AI

27 September, 2020

Retina AI R Datathon - Submission Template

Team Name: Triple A
Team Members: Adhvaith Vijay, Andrew Liu, Anurag Pamuru

Directions: Write your text and codes for Task 1, 2 and 3 in the provided space below.

Task 1 insights to the visualization can be written as plain text in the field provided below. You can also add new bullet points to each slide's insights section.

For Task 2 and 3, provide your codes in the code chunks below. If your team needs additional code chunks to run your code, you can add new code chunks.

Task 1 (Required)

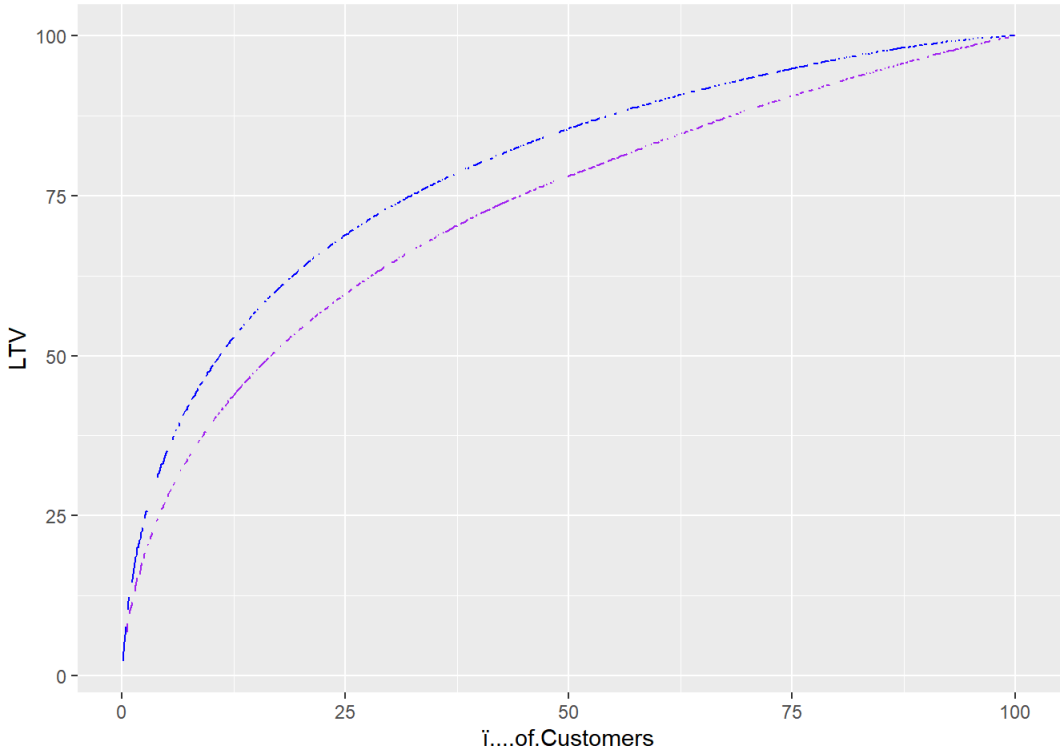
```
convert <- function(x) {
  vector <- as.numeric()
  vector <- c(vector, x[1])
  for (i in seq_along(x)) {
    if (i != 1) {
      x[i] = x[i] - sum(vector)
      vector <- c(vector, x[i])
    }
  }
  vector
}

rows_of_interest1 <- which(data14$ï....of.Customers %in% c(20.0, 40.1, 59.9, 80.1, 100.0))
revenues <- data14$Revenue[rows_of_interest1]
# these are each 20% intervals for revenue
print(convert(revenues))
```

```
## [1] 54.2 18.0 11.1  9.5  7.2
```

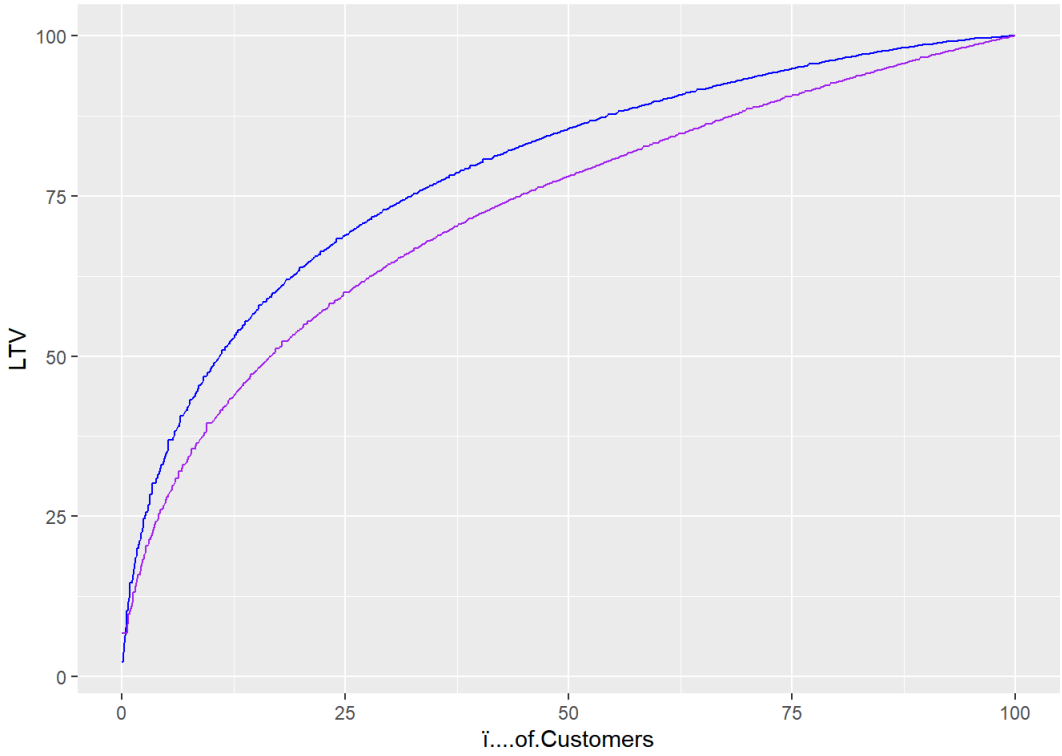
```
rows_of_interest2 <- which(data14$ï....of.Customers %in% c(20.1, 40.0, 60.1, 79.9, 99.9))
LTVs <- data14$LTV[rows_of_interest2]
# these are each 20% intervals for LTVs
print(convert(LTVs))
```

```
## [1] 63.8 16.3  9.7  6.5  3.7
```

```
# Graph without imputed values (notice holes that exist)
g <- ggplot(data14, aes(ï....of.Customers)) +
     geom_line(aes(y=LTV), colour="blue") +
     geom_line(aes(y=Revenue), colour="purple")
g
```




```
# Impute NA values using backwards fill to ensure continuous distribution
data14$ï....of.Customers <- na.locf(data14$ï....of.Customers, fromLast = TRUE)
data14$LTV <- c(na.locf(data14$LTV, fromLast = TRUE), 100.00)
data14$Revenue <- na.locf(data14$Revenue, fromLast = TRUE)

# Graph with imputed values (notice holes are filled)
g2 <- ggplot(data14, aes(ï....of.Customers)) +
      geom_line(aes(y=LTV), colour="blue") +
      geom_line(aes(y=Revenue), colour="purple")
g2
```



Slide Number: 14

Key Insights:

- **Groomers should target high spenders**

Groomers should focus their efforts on targeting demographics related to their top spenders when acquiring new customers. From

observing the graph, about 17% of their customers construct about 60% and 50% of their 5-year LTV and revenue respectively. Furthermore, Groomers' top 1% of customers bring in about 15% and 10% of their 5-year LTV and revenue respectively. On our end, we divided the data into 5 quantiles and calculated the revenue for each quantile. Our quantiles are as follows: quantile 1 generated 54.2% of the revenue, quantile 2 generates 18.0% of the revenue, quantile 3 generates 11.1% of the revenue, quantile 4 generates 9.5% of the revenue, and quantile 5 generates 7.2% of the revenue. Once again, we see that the upper quantile of spenders generate the majority of the revenue.

- **Expected Spending Index**

We created an approximate copy of the 5-year LTV and revenue curves using the ggplot2 package. In order to fill in missing values we used backpropagation to create a continuous curve.This helped in deducing the 'area' between the two curves' - a handy metric down the line. Similar to our above insight, we split the LTV graph into 5 quantiles based on the percentage of customers. We then calculated an approximation of the area between the curves in each quantile to see how much customers were expected to spend over 5 years (the difference between LTV and revenue). This metric is the Expected Spend Index (ESI). A higher ESI denotes a quantile of spenders with a higher potential spending capacity over 5 years. We found that the first 3 quantiles had the largest ESI while the last two quantiles had the smallest ESI. We noticed that the last quantile had by far the smallest ESI indicating that those who spend less are unlikely to spend more. The exact results were as follows: quantile 1 had an ESI of 156.5, quantile 2 had an ESI of 175.7, quantile 3 had an ESI of 146.9, quantile 4 had an ESI of 99.9, and quantile 5 had an ESI of 38. As a result, we suggest that Groomers incentivize and/or market towards those who fall in the top 60% (people who have historically spent more on Groomers) with a focus being on quantile 2.

Slide Number: 16

Key Insights:

- **Lifetime Revenue Weighed against Customer Acquisition Cost**

Since we are working with Cumulative Distribute Functions, the 50th percentile of the Customers CDF in Slide 16 corresponds to the median. This median value is `$84`, and therefore Groomers' average customer acquisition cost (CAC) of `$30` is a seemingly good investment. The first quantile predicting the 10-Year LTR of a customer is about `$50`, and the third quantile predicted 10-Year LTR of a customer is about `$168`. Almost 9% of customers have a predicted 10-Year Lifetime Revenue of less than `$30` (Groomers' average CAC). This specific customer base makes up just 1% of Groomers' total predicted 10-Year Lifetime Revenue. Therefore, Groomers should stop targeting those customers with unprofitable 10-year Lifetime Revenue.

- **High value of top spenders**

The benefit and value of retaining top spenders is reinforced by the curve of the cumulative distribution function of the percentage of customers. This can be seen by the fact this CDF is heavily right skewed with many positive outliers. This is evidenced by the early plateau in the CDF plot. This right skew was noticed only after scaling the x-axis appropriately. Furthermore, the average lifetime value provided to us in the QoC is `$130`. Since this figure is higher than the median of `$84`, this further reinforces that the dataset is right skewed. Considering that Groomers is looking to further improve customer retention by the following strategies: to acquire higher lifetime value customers at the outset, launching a loyalty program, or creating a subscription offering, we recommend that Groomers tries acquiring higher lifetime value customers at the outset since they make up such a large portion of their total expected revenue. For that same reason, we recommend against strategies that pander to lower lifetime value customers - such as creating subscription offerings - since it may cause Groomers to loose out on attracting their biggest spenders (i.e. 'whales').

Task 2 (Required)

Slide Number: 14

```
# automize insights for slides 14 and 16

# helper function
calculate_differences <- function(x) {
  vector <- as.numeric()
  vector <- c(vector, x[1])
  for (i in seq_along(x)) {
    if (i != 1) {
      x[i] = x[i] - sum(vector)
      vector <- c(vector, x[i])
    }
  }
  vector
}

# Finds= the closest element to target parameter within a given parameter vector
# helper function
closest <- function(v, target) {
  which.min(abs(v-target))
}

# calculate area between curves (Method #1)
find_quantile_contributions <- function(feature_col, pct_col, bin_cnt) {
```

```
find_quantile_contributions <- function(feature_col, pct_col, bin_cnt) {
  # ideal bins
  bin_size = 100/bin_cnt
  bins <- seq(from = bin_size, to = 100, by = bin_size)
  # find closest non-NA bins to ideal bins
  non_null_pct <- pct_col[which(!is.na(feature_col))]
  quantile_limits <- non_null_pct[to_vec(for(i in bins) closest(non_null_pct, i))]
  quantile_limits <- feature_col[which(pct_col %in% quantile_limits)]
  # calculate bin differences
  calculate_differences(quantile_limits)
}

# calculate area between curves (Method #2)
calc_area_between_curves_integrate <- function(x_axis, curve_low, curve_high, x_min = 0, x_max = 100) {
  x <- na.locf(x_axis, fromLast = TRUE)
  c_l <- na.locf(curve_low, fromLast = TRUE)
  c_h <- na.locf(curve_high, fromLast = TRUE)
  f1 = approxfun(x, c_h)
  AUC1 = integrate(f1, x_min, x_max, subdivisions = 1000)
  f2 = approxfun(x, c_l)
  AUC2 = integrate(f2, x_min, x_max, subdivisions = 1000)
  AUC1$value - AUC2$value
}

find_quantile_area_between_curves <- function(curve_low, curve_high, x, bin_cnt) {
  # ideal bins
  bin = 100/bin_cnt
  bins <- seq(from = bin, to = 100, by = bin)
  # find closest non-NA bins to ideal bins
  non_null_pct <- x[which(!is.na(curve_low))]
  quantiles <- non_null_pct[to_vec(for(i in bins) closest(non_null_pct, i))]
  # calculate area between curve in quantile
  for( i in 1:length(quantiles) ) {
    x_min <- quantiles[i] - bin
    x_max <- quantiles[i]
    area <- calc_area_between_curves_integrate(x, curve_low, curve_high, x_min, x_max)
    cat("Quantile", i, "has an ESI of", area, "\n")
  }
}

plot_area_between_curves <- function(x_axis, curve_low, curve_high, x_min = 0, x_max = 100) {
  # impute missing values
  x <- na.locf(x_axis, fromLast = TRUE)
  c_l <- na.locf(curve_low, fromLast = TRUE)
  c_h <- na.locf(curve_high, fromLast = TRUE)
  # imputed dataframe
  df <- data.frame(x, c_l, c_h)
  blue<-rgb(0.8, 0.8, 1, alpha=0.25)
  clear<-rgb(1, 0, 0, alpha=0.0001)
  ggplot(df, aes(x=x, y=c_l)) +
      geom_line(aes(y = c_l)) +
      geom_line(aes(y = c_h)) +
      geom_ribbon(data=subset(df, x_min <= x & x <= x_max),
            aes(ymin=c_l,ymax=c_h), fill="blue", alpha=0.5) +
      scale_y_continuous(expand = c(0, 0), limits=c(0,100)) +
      scale_x_continuous(expand = c(0, 0), limits=c(0,100)) +
      scale_fill_manual(values=c(clear,blue))
}


# calculate area between curves across subset of data14 for testing purposes
calc_area_between_curves_integrate(data14$ï....of.Customers, data14$Revenue, data14$LTV, 20, 40)
```

```
## [1] 175.7183
```

```
# calculate total area between curves for data14 for testing purposes
calc_area_between_curves_integrate(data14$ï....of.Customers, data14$Revenue, data14$LTV)
```

```
## [1] 616.5263
```

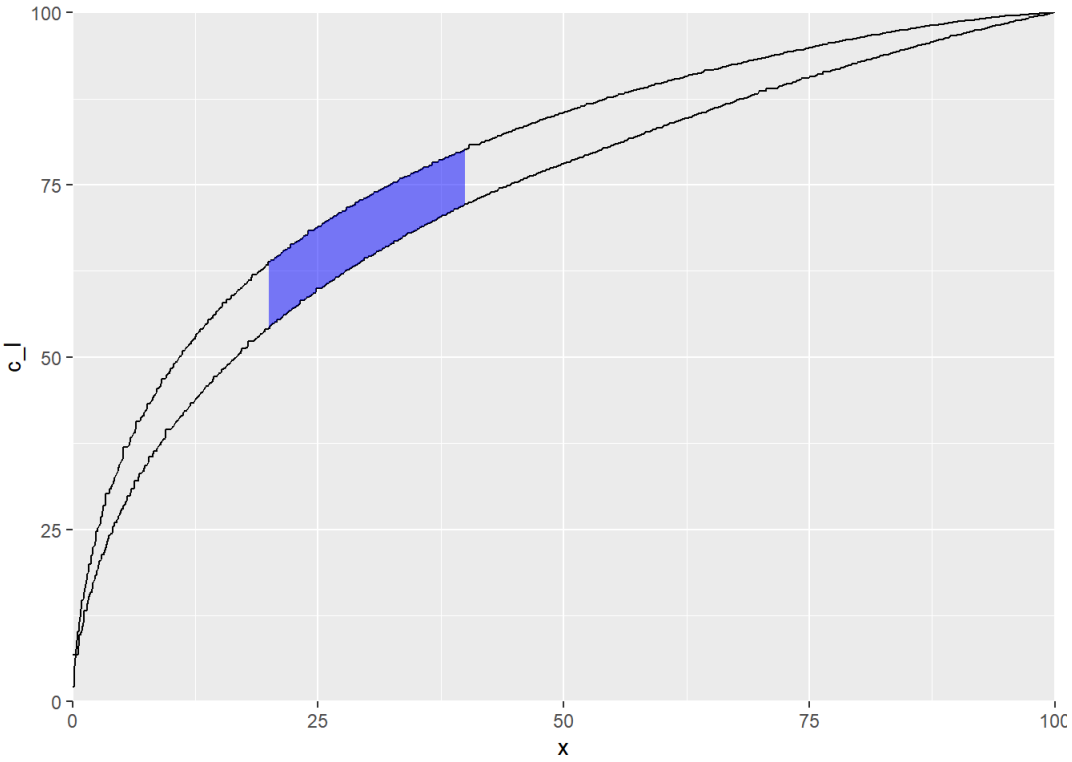
```
find_quantile_area_between_curves(data14$Revenue, data14$LTV, data14$ï....of.Customers, 5)
```

```
## Quantile 1 has an ESI of 156.466
## Quantile 2 has an ESI of 175.7183
## Quantile 3 has an ESI of 146.9054
## Quantile 4 has an ESI of 99.85634
## Quantile 5 has an ESI of 38.01204
```

```
contributions <- find_quantile_contributions(data14$Revenue, data14$ï....of.Customers, 5)
# Print automated insights using original.csv data from task 1 visualization for testing purposes
for (i in 1:length(contributions)) {
  cat("Quantile", i, "generates", contributions[i], "% of Revenue\n")
}
```

```
## Quantile 1 generates 54.2 % of Revenue
## Quantile 2 generates 18 % of Revenue
## Quantile 3 generates 11.1 % of Revenue
## Quantile 4 generates 9.5 % of Revenue
## Quantile 5 generates 7.2 % of Revenue
```

```
# plot area between curves across a subset of data14 for testing purposes
plot_area_between_curves(data14$ï....of.Customers, data14$Revenue, data14$LTV, 20, 40)
```



Slide Number: 16

```
# Enter your code to automize insights for Slide 16 here
calculate_median_from_percentile_col_and_feature_col <- function(percentile_col, feature_col) {
  feature_col[closest(percentile_col, 50)]
}

# Print automated insights. Use original.csv data from task 1 visualization.
calculate_median_from_percentile_col_and_feature_col(data16$cumpct_cust, data16$ï..Revenue.in.USD)
```

```
## [1] 84
```

Task 3 (Bonus/Optional)

Key Insights:

- **Relationship between Activity and Predicted Future CLV**

The first graph was derived using the LTV data from the additional file (ltv_tables.csv). In this graph, we plotted the probability that customers were active as a function of both 1 year and 5 year predicted CLVs. We noticed that across 1 year predicted CLVs the distribution of 'Probability Alive' percents was not nearly as extreme as when 'Probability Alive' was plotted against 5 year predicted CLVs. In essence as time progressed the notion that more active member generate a larger future CLV and vice versa holds true. The greater the probability that a customer is active, the more larger the Predicted future CLV - this relationship is directly proportional to the timeframe in question. Groomers should not pour resources into customers with a low probability of being alive and a low Lifetime revenue since this is unprofitable.

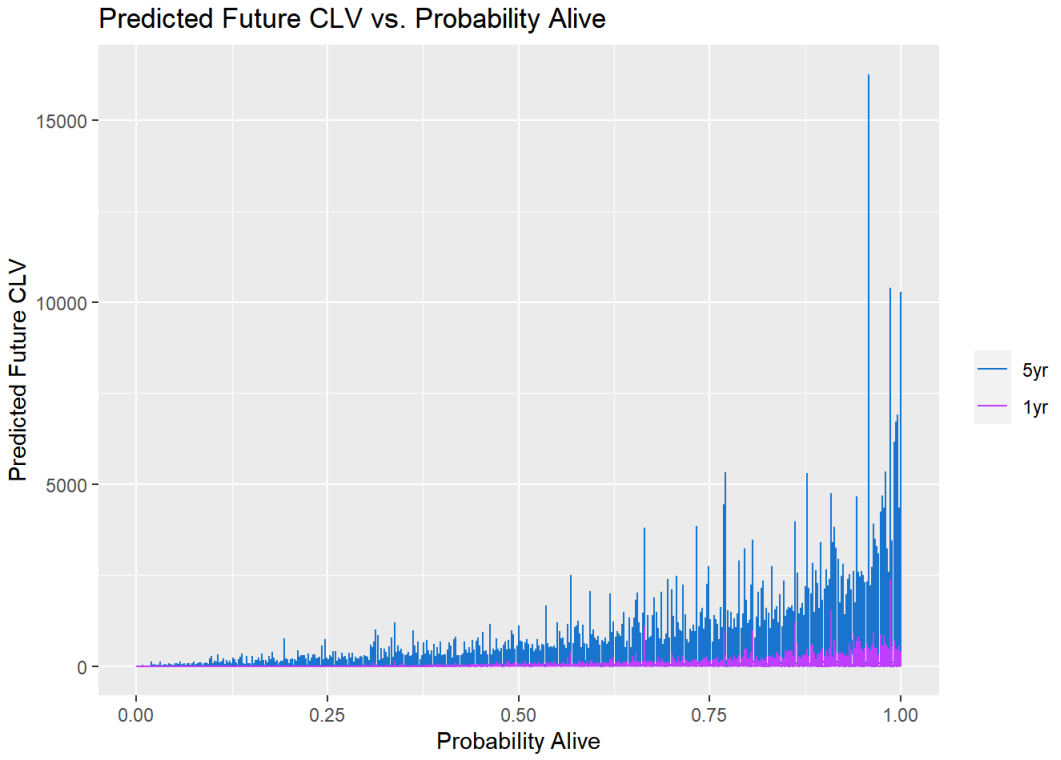
- **Relationship between Time and Retention Rate**

We created a graph from the monthly retention data in the additional files. We aggregated the data from 2016 - 2020 and plotted the retention rates against time (periods) - which indicated the fluctuating retention rate from period[X1] to period[X2] for a certain cohort. It was clear that as time went on, people were more likely to churn. This is also shown in Slide 27 of the QoC, where regardless of Cohort Year, as more time passes, a pattern of severe customer retention loss emerges - **similar to exponential decay.** Based on these observations, Groomers should plan to consistently look for newer customers in order to fill in the holes left by the inevitable churn rate that only appears to worsen year on year.

How to read this chart:

- For the Predicted Future CLV vs. Probability Alive Graph, the y-axis is the predicted future CLV. The x-axis is the probability that a customer is alive/active. The blue color represents the 5 year predicted future CLV while the purple color represents 1 year predicted future CLV.
- For the Retention Rate vs. Time Graph the y-axis represents the Retention Rate and the x-axis represents the time grouped by periods which is 'Months since the Cohort made their First Purchase'. The different colors of line each represent a different year of aggregate data from 2016 - 2020.

```
ggplot(data = ltv, aes(x = probability_alive)) +
  geom_line(aes(y = predicted_future_clv_05yr, colour = "5yr")) +
  geom_line(aes(y = predicted_future_clv_01yr, colour = "1yr")) +
  scale_colour_manual("",
                      breaks = c("5yr", "1yr"),
                      values = c("dodgerblue3", "darkorchid1")) +
  xlab("Probability Alive") +
  scale_y_continuous("Predicted Future CLV") +
  labs(title="Predicted Future CLV vs. Probability Alive")
```



```
# aggregated by year and transposed to graph mean retention rate by time (period)
monthly$cohort_yearmonth <- lapply(monthly$cohort_yearmonth, as.character)
monthly$cohort_yearmonth <- substr(monthly[,'cohort_yearmonth'], 1, nchar(monthly[,'cohort_yearmonth'])-3)
monthly <- subset(monthly, select = -c(num_new_customers))
monthly <- aggregate(. ~ cohort_yearmonth, monthly, mean)
monthly <- subset(monthly, select = -c(cohort_yearmonth))
monthly <- as.data.frame(t(as.matrix(monthly)))
colnames(monthly)<- c("2016","2017","2018", "2019", "2020")
monthly <- cbind(Period = rownames(monthly), monthly)
monthly$Period <- gsub('period', '', monthly$Period)
monthly$Period <- as.numeric(as.character(monthly$Period))

# re-order row index numbers
rownames(monthly) <- NULL
monthly <- melt(monthly, id.vars="Period")
ggplot(monthly, aes(Period, value, col=variable)) +
  geom_point() +
  stat_smooth() +
  xlab('Time (Period)') +
  ylab('Retention Rate') +
  labs(color='Year') +
  ggtitle('Retention Rate vs. Time (Period)')
```