FabFitFun Coding Challenge

Kimberly Insigne

This notebook contains analysis for transaction data from a flash sale on the FabFitFun website. `product_data.xlsx` contains product IDs and product names. `sale_data.xlsx` contains member IDs, product IDs, and quantity of each item purchased.

```
product_data <- read.xlsx('product_data.xlsx')
sale_data <- read.xlsx('sale_data.xlsx') %>%
    left_join(product_data, by = 'product_id')
```

Slide 1

How many products does each member purchase? Do a small group of members make up the majority of sales? Pareto principle: "80% of sales come from 20% of clients"

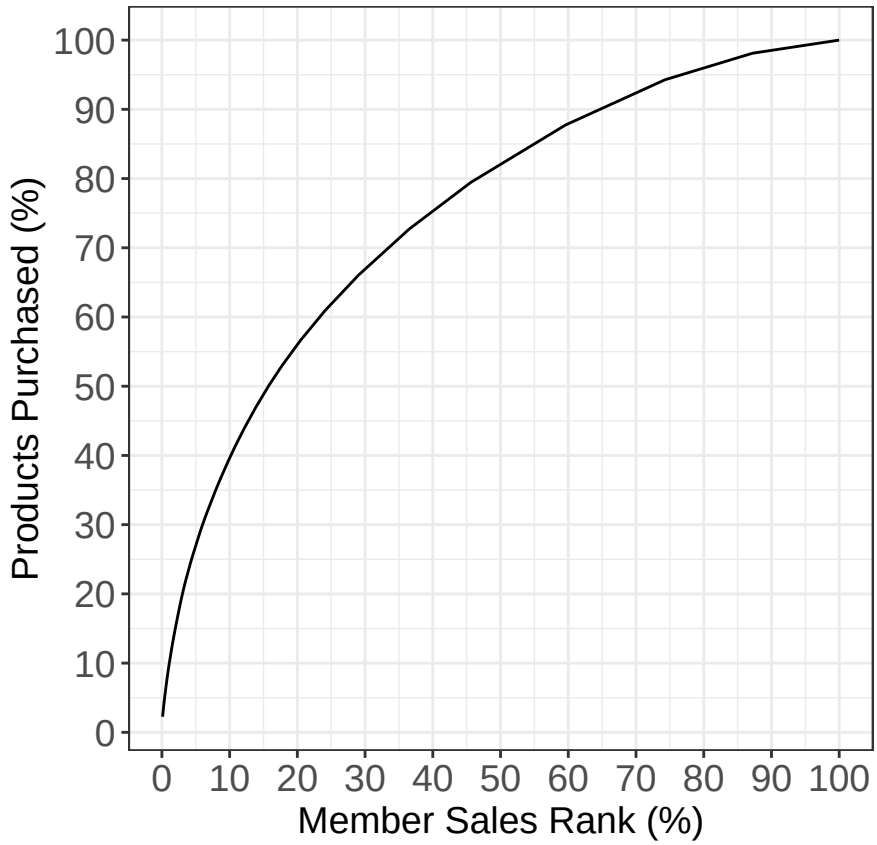
```
num_members <- length(unique(sale_data$member_id))

member_num_purchases <- sale_data %>%
    group_by(member_id) %>%
    summarise(num_products = sum(quantity)) %>%
    arrange(desc(num_products)) %>%
    mutate(member_rank = seq(1:num_members))

member_num_purchases %>%
    ggplot(aes(member_rank, num_products)) +
    geom_bar(stat = 'identity', color = 'navyblue') +
    theme(axis.text.x = element_blank(),
          axis.ticks.x = element_blank(),
          axis.text = element_text(size = 16),
          axis.title = element_text(size = 16)) +
    labs(x = 'Member Sales Rank', y = 'Number of Products Purchased')
```


```
member_num_purchases <- member_num_purchases %>%
    mutate(product_percentage = (num_products / sum(num_products)) * 100) %>%
    arrange(desc(product_percentage)) %>%
    mutate(cumulative_products = cumsum(product_percentage),
           member_rank_percentage = (member_rank / num_members) * 100)

ggplot(member_num_purchases, aes(member_rank_percentage, cumulative_products)) +
    geom_line() +
    scale_x_continuous(breaks = seq(0, 100, 10)) +
    scale_y_continuous(breaks = seq(0, 100, 10)) +
    theme_bw() +
    theme(axis.text = element_text(size = 14),
          axis.title = element_text(size=14),
          aspect.ratio = 1) +
    labs(x = 'Member Sales Rank (%)', y = 'Products Purchased (%)')
```



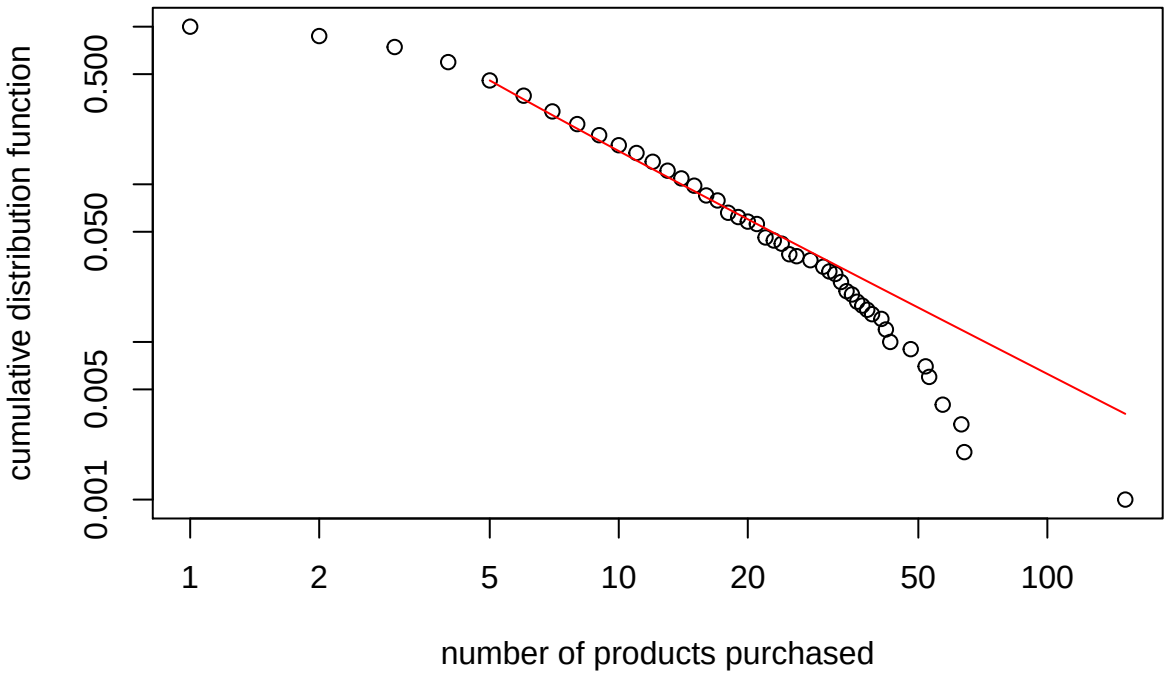
Let's fit a power law distribution.

```
sales_model <- displ$new(member_num_purchases$num_products)
sales_model_min <- estimate_xmin(sales_model)
sales_model$setXmin(sales_model_min)
print(c(sales_model$xmin, sales_model$pars))
```

```
## [1] 5.000000 2.386842
```

We can fit a power law distribution once the product quantity reaches 5 or higher, with a scaling factor alpha of 2.38.

```
plot(sales_model,
     xlab = 'number of products purchased',
     ylab = 'cumulative distribution function')
lines(sales_model, col=2)
```

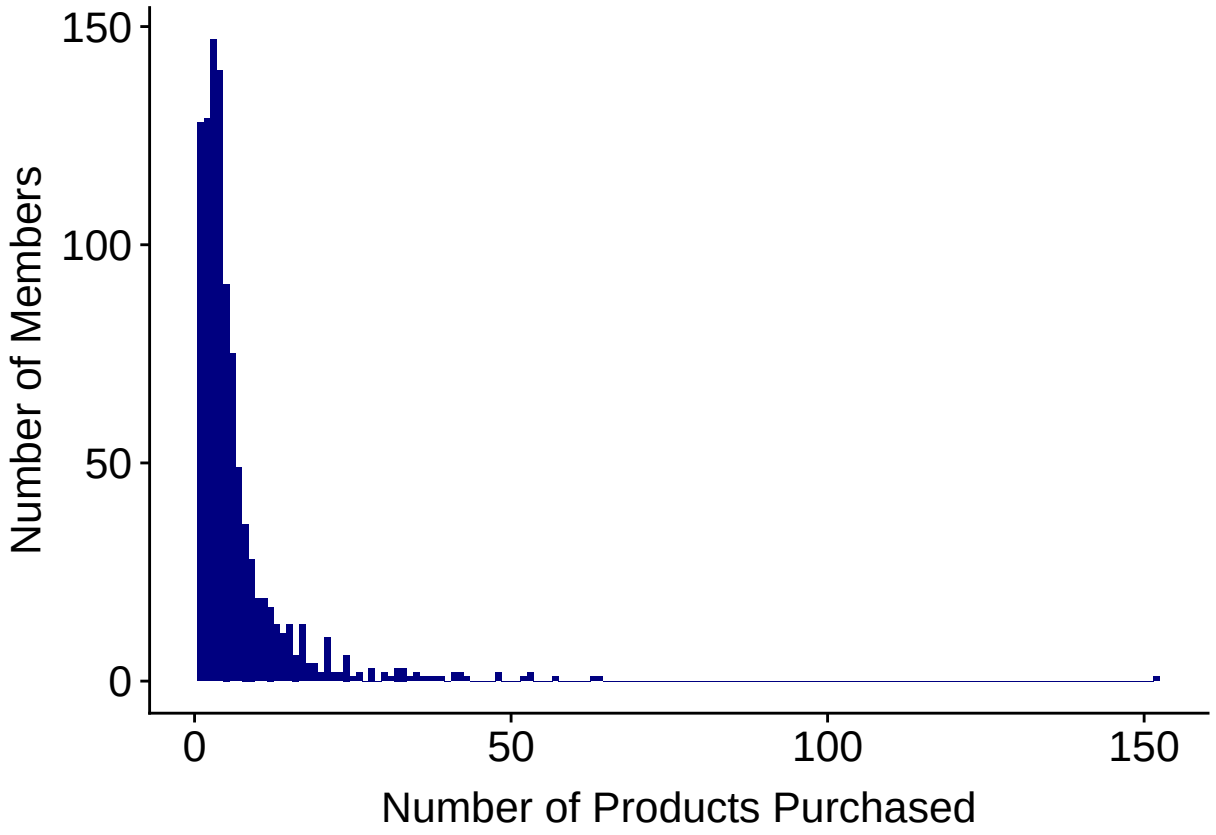


A minority of clients make up a majority of the sales. 20% of members make up 56% of the sales, 80% of sales are made by 46.6% of members. Not quite the 80-20 rule, but still interesting. In an ideal world, we would want all of our members to be equally engaged and participate in the flash sales, although number of sales is a different metric of engagement than simply buying anything from the sale (there are many members who don't buy anything).

Slide 2

How many items do members tend to purchase?

```
ggplot(member_num_purchases, aes(num_products)) +
    geom_histogram(binwidth = 1, fill='navyblue') +
    labs(x = 'Number of Products Purchased', y = 'Number of Members') +
    theme(axis.text = element_text(size=16),
          axis.title = element_text(size=16))
```



The most common number of products purchased is 3 items, followed by 4, 2, and 1. We don't have the price for each product, so we can't tell if members are purchasing more cheaper items or fewer expensive items. It would be worth exploring why members tend to buy multiple items. Is there a free shipping minimum they are trying to hit? If so, it could be a potential incentive to increase sales. Is it better to offer more cheap products or more expensive products?

Slide 3

Do a small number of products make up a large portion of sales? (80-20 rule)

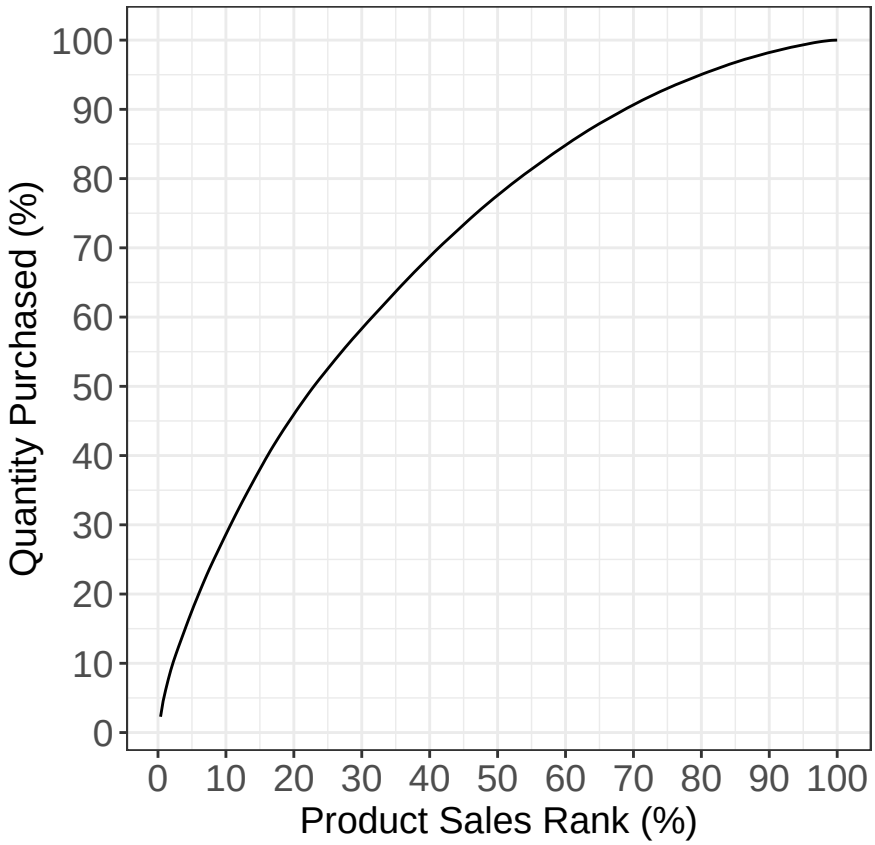
```
num_products <- length(unique(sale_data$product_name))

product_summary <- sale_data %>%
    group_by(product_name) %>%
    summarise(product_count = sum(quantity)) %>%
    arrange(desc(product_count)) %>%
    mutate(product_rank = seq(1, num_products))

product_summary %>%
    ggplot(aes(product_rank, product_count)) +
    geom_bar(stat = 'identity', color = 'navy blue') +
    scale_y_continuous(breaks = seq(0, 150, 25)) +
    theme(axis.text.x = element_blank(),
          axis.ticks.x = element_blank()) +
    labs(x = 'Product Sales Rank', y = 'Number Sold') +
    theme(axis.text = element_text(size=16),
          axis.title = element_text(size=16),
```

```
        aspect.ratio = 1)
```


```
product_summary %>%
    mutate(quantity_percentage = (product_count / sum(product_count)) * 100) %>%
    arrange(desc(quantity_percentage)) %>%
    mutate(cumulative_quant = cumsum(quantity_percentage),
           product_rank_percentage = (product_rank / num_products) * 100) %>%
    ggplot(aes(product_rank_percentage, cumulative_quant)) +
    geom_line() +
    scale_x_continuous(breaks = seq(0, 100, 10)) +
    scale_y_continuous(breaks = seq(0, 100, 10)) +
    theme_bw() +
    theme(axis.text = element_text(size = 14),
          axis.title = element_text(size=14),
          aspect.ratio = 1) +
    labs(x = 'Product Sales Rank (%)', y = 'Quantity Purchased (%)')
```



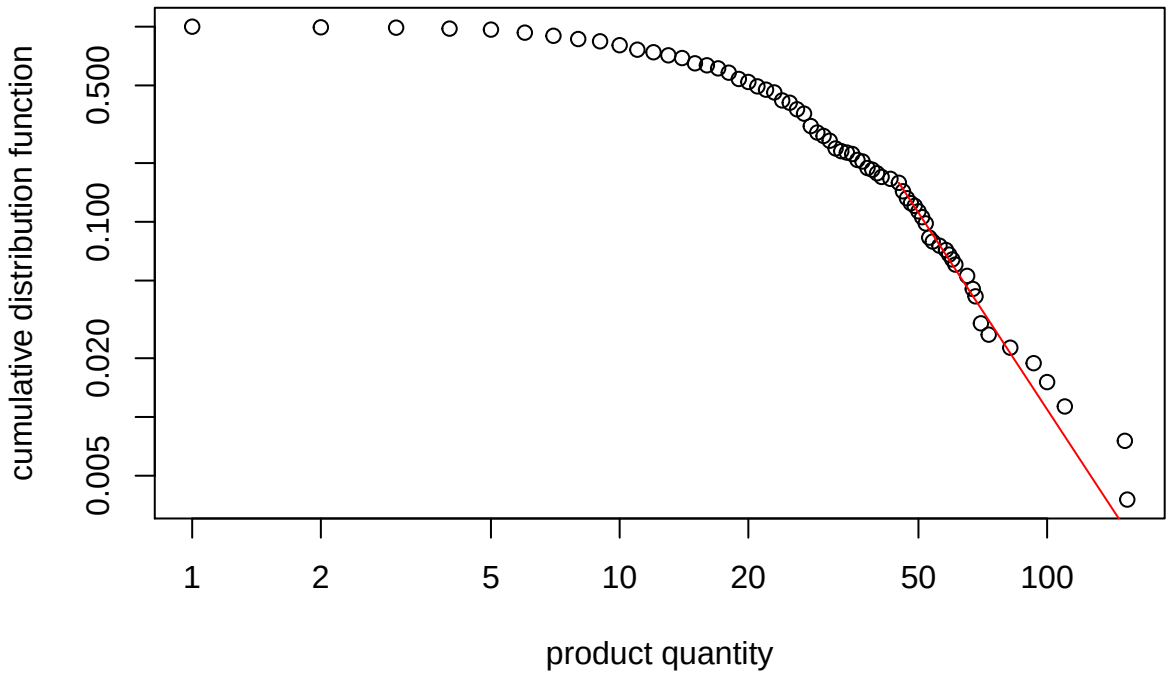
The number of sales for each product appears to follow a power law distribution, with a few products having a high number of sales. Let's fit a power law distribution.

```
# displ - discrete power law
product_model <- displ$new(product_summary$product_count)
# estimate minimum value of product count for which power law holds
product_count_min <- estimate_xmin(product_model)
# update power law
product_model$setXmin(product_count_min)
print(c(product_model$xmin, product_model$pars))
```

```
## [1] 45.000000  4.323171
```

We can fit a power law distribution once the product quantity reaches 45 or higher, with a scaling factor alpha of 4.32.

```
plot(product_model,
     ylab = 'cumulative distribution function',
     xlab = 'product quantity')
lines(product_model, col=2)
```




```
product_summary %>%
    arrange(desc(product_count)) %>%
    slice(1:10) %>%
    ggplot(aes(reorder(product_name, product_count), product_count)) +
    geom_bar(stat = 'identity', fill='navyblue') +
    coord_flip() +
    theme(axis.text = element_text(size = 15),
          title = element_text(size = 15)) +
    labs(x = '', y = 'Number Sold',
         title = 'Top 10 selling products')
```


```
product_summary %>%
    arrange(product_count) %>%
    slice(1:10) %>%
    ggplot(aes(reorder(product_name, product_count), product_count)) +
    geom_bar(stat = 'identity', fill='navyblue') +
    coord_flip() +
    theme(axis.text = element_text(size = 15),
          title = element_text(size = 15)) +
    labs(x = '', y = 'Number Sold',
         title = 'Bottom 10\nselling products')
```


Bottom 10
selling product
Additional FabFitFun Fall 2018 Box
Variation #643876 of Rachel Pally Reversible Clutch
The Vanity Project Wine Wipes
Daily Concepts Your Dual Texture Scrubber
Variation #643875 of Rachel Pally Reversible Clutch
Palmetto Derma Essential Moisturizing Cream
ISH Cheeky Lip Peach Bundle
Clear as Lake Louise Bundle
FabFitFun Backpack
Air is Fresher in Canada Bundle
0 1 2 3 4 5
Number Sold