Predicting Future Revenue:
StandDesk B2B Customers

Daniel R. Poston
www.sharpdata.co
July 31, 2016

Introduction: The Problem

Leaving money on the table is a problem most companies try to avoid. It's an issue that presents itself to business in many forms. As a business gains more customers it can become increasingly difficult to determine which of them are likely to order more products, and why. Our focus will be on maximizing revenue from current customers who are likely to make additional purchases.

My client on this project is StandDesk, Inc. They manufacture the most cost effective automatic height adjustable desks on the market (Receive a $50 discount by ordering here!). After raising over $800,000 on Kickstarter in 2014, they quickly began taking orders from businesses who were trying to create a healthier workspace for their employees. For a lean startup like StandDesk, it can become increasingly difficult to maximize revenue from a rapidly expanding business to business customer base without utilizing a laser like focus of marketing and sales resources on the right customers at the right times. We will attempt to determine which customers to focus on by creating a model that predicts the total revenue that can be generated from each customer.

Pre Project Data Prep

The original files downloaded from Shopify and Hubspot contain proprietary information. It was necessary for this information to be removed and replaced with a generic ID. First, a generic ID was generated for each company on the original Hubspot Company csv file.

Each of the three original files was loaded to R as a data frame. A "Domain" field was created in each file. Then the generic ID was added to the Hubspot Contacts data frame and the Shopify Sales data frame by matching the domain field from the Hubspot Companies data frame.

```
Company_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-companies2016-06-24wID.csv", header=TRUE, sep=",", na.strings = "")
cid_df <- data.frame(Company_Report)
```

```
# Importing Shopify order data
Order_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/Sales by customer
6.24.csv", header=TRUE, sep=",", na.strings = "")
orid_df <- data.frame(Order_Report)

orid_df <- separate(orid_df, email, c("Email Prefix", "Domain"), sep = "@")
orid_df <- filter(orid_df, total_sales > 0)
orid_df$Domain <- tolower(orid_df$Domain)

cid_df <- mutate(cid_df, Domain = Company.Domain.Name)
cid_df$Company.Domain.Name <- NULL
cid_df$Domain <- tolower(cid_df$Domain)

sub_cid_df <- select(cid_df, Domain, Company.ID)
orid_df <- inner_join(orid_df, sub_cid_df, by = "Domain")

Contact_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-
contacts2016-06-24.csv", header=TRUE, sep=",", na.strings = "")
contactid_df <- data.frame(Contact_Report)

contactid_df <- separate(contactid_df, Email, c("Email Prefix", "Domain"), sep = "@")
contactid_df$Domain <- tolower(contactid_df$Domain)
contactid_df <- filter(contactid_df, Domain != "standdesk.co")
contactid_df <- inner_join(contactid_df, sub_cid_df, by = "Domain")
```

Once each row in each file was matched with the proper ID, all identification information was removed from the data frames and new csv files were generated.

```
orid_df$name <- NULL
orid_df$`Email Prefix` <- NULL
orid_df$Domain <- NULL
orid_df$company <- NULL

cid_df$Name <- NULL
cid_df$Street.Address <- NULL
cid_df$Website.URL <- NULL
cid_df$Facebook.Company.Page <- NULL
cid_df$Google.Plus.Page <- NULL
cid_df$LinkedIn.Bio <- NULL
cid_df$LinkedIn.Company.Page <- NULL
cid_df$Twitter.Handle <- NULL
cid_df$Domain <- NULL


contactid_df$First.Name <- NULL
```

```
contactid_df$Last.Name <- NULL
contactid_df$Company.Name <- NULL
contactid_df$'Email Prefix' <- NULL
contactid_df$Domain <- NULL
contactid_df$Phone.Number <- NULL
contactid_df$Street.Address <- NULL
contactid_df$Billing.Address.Line.1 <- NULL
contactid_df$Shipping.Address.Line.1 <- NULL
contactid_df$ReferrerEmail <- NULL
contactid_df$IP.Address <- NULL
contactid_df$Website.URL <- NULL

write.csv(orid_df, file = "C:/Users/GamingFoSho/Documents/wdR/Sales by customer 6.24wID1.csv", row.names=FALSE)
write.csv(cid_df, file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-companies2016-06-24wIDclean1.csv", row.names=FALSE)
write.csv(contactid_df, file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-contacts2016-06-24wID2.csv", row.names=FALSE)
```

Data Set

The StandDesk dataset consists of 3 csv files:

- **Order Report: Sales by customer 6.24wID1.csv (Shopify)**
- **Company Report: hubspot-crm-view-companies2016-06-24wIDclean1.csv (Hubspot)**
- **Contact Report: hubspot-crm-view-contacts2016-06-24wID2.csv (Hubspot)**

The data was downloaded from StandDesk's CRM and Shopify account. The CRM contains information related to interactions between StandDesk and their customers as well as their customers' demographic information. Data downloaded from Shopify contains information related to purchases from StandDesk's customers.

The data from Shopify was downloaded as a csv file and contains information on each purchase a customer has made. The Order report downloaded from Shopify includes Purchase Date, Purchase Amount, Contact's Email, etc. This file was imported into R as a data frame called "or_df."

The data from the CRM was downloaded as two separate csv files. The first csv file was downloaded from the Companies tab and contains data on all of the companies which have purchased from StandDesk. This data mostly consists of information about each company such as Phone Number, Website, Address, etc., along with demographic data including Employee Count, Industry, Year Founded, etc. This file was imported into R as a data frame called "c_df."

The second csv file was downloaded from the Contacts tab which contains data on the contacts related to the companies who have purchased from StandDesk. This data contains information about the contacts such as their Title, along with activity data related to how the contact has interacted with StandDesk. The activity data includes Email communications, calls, website visits, form submissions, etc. This file was imported into R as a data frame called “contact_df.”

Data Wrangling

Each csv file was uploaded to R:

```
Order_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/Sales by customer 6.24wID1.csv", header=TRUE, sep=",", na.strings = "")
or_df <- data.frame(Order_Report)

Company_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-companies2016-06-24wIDclean1.csv", header=TRUE, sep=",", na.strings = "")
c_df <- data.frame(Company_Report)

Contact_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-contacts2016-06-24wID2.csv", header=TRUE, sep=",", na.strings = "")
contact_df <- data.frame(Contact_Report)
```

The order dates of each order in the or_df data frame were aggregated based on the Company.ID field.

```
or_day_vec <- as.Date(or_df$day, "%m/%d/%Y")
or_df <- mutate(or_df, day = or_day_vec)
or_df <- mutate(or_df, Orders = 1)
attach(or_df)
or_df <- or_df[order(or_df$Company.ID, or_df$day),]
or_do_df <- select(or_df, Company.ID, Orders)
detach(or_df)
or_do_df <- ddply(or_do_df, .(Company.ID), mutate, Order_Number = cumsum(Orders))
or_df <- mutate(or_df, Order_Number = or_do_df$Order_Number)
or_df$Orders <- NULL
or_df <- filter(or_df, Order_Number <= 14)


order_date_df <- select(or_df, Company.ID, day, Order_Number)

Order_Date_vec <- order_date_df$Order_Number
Order_Date_vec1 <- ifelse((Order_Date_vec == 1), c("Order_One_Date"), c("Unknown"))
Order_Date_vec2 <- ifelse((Order_Date_vec == 2), c("Order_Two_Date"), Order_Date_vec1)
Order_Date_vec3 <- ifelse((Order_Date_vec == 3), c("Order_Three_Date"), Order_Date_vec2)
```

```
Order_Date_vec4 <- ifelse((Order_Date_vec == 4), c("Order_Four_Date"), Order_Date_vec3)
Order_Date_vec5 <- ifelse((Order_Date_vec == 5), c("Order_Five_Date"), Order_Date_vec4)
Order_Date_vec6 <- ifelse((Order_Date_vec == 6), c("Order_Six_Date"), Order_Date_vec5)
Order_Date_vec7 <- ifelse((Order_Date_vec == 7), c("Order_Seven_Date"), Order_Date_vec6)
Order_Date_vec8 <- ifelse((Order_Date_vec == 8), c("Order_Eight_Date"), Order_Date_vec7)
Order_Date_vec9 <- ifelse((Order_Date_vec == 9), c("Order_Nine_Date"), Order_Date_vec8)
Order_Date_vec10 <- ifelse((Order_Date_vec == 10), c("Order_Ten_Date"), Order_Date_vec9)
Order_Date_vec11 <- ifelse((Order_Date_vec == 11), c("Order_Eleven_Date"),
Order_Date_vec10)
Order_Date_vec12 <- ifelse((Order_Date_vec == 12), c("Order_Twelve_Date"),
Order_Date_vec11)
Order_Date_vec13 <- ifelse((Order_Date_vec == 13), c("Order_Thirteen_Date"),
Order_Date_vec12)
Order_Date_vec14 <- ifelse((Order_Date_vec == 14), c("Order_Fourteen_Date"),
Order_Date_vec13)
order_date_df <- mutate(order_date_df, Order_Number = Order_Date_vec14)
order_date_df <- spread(order_date_df, Order_Number, day)
order_date_df <- order_date_df[c("Company.ID", "Order_One_Date", "Order_Two_Date",
"Order_Three_Date",   "Order_Four_Date", "Order_Five_Date", "Order_Six_Date",
Order_Seven_Date", "Order_Eight_Date", "Order_Nine_Date", "Order_Ten_Date",
"Order_Eleven_Date", "Order_Twelve_Date", "Order_Thirteen_Date", "Order_Fourteen_Date")]
```

The order amounts for each order in the or_df data frame were aggregated based on the Company.ID field.

```
or_num_df1 <- or_df
or_num_df1$company <- NULL
or_num_df1$shipping_city <- NULL
or_num_df1$shipping_province <- NULL
or_num_df1$day <- NULL
or_num_df1$year <- NULL
or_num_df1$email <- NULL
or_num_df1$traffic_source <- NULL
or_num_df1$host <- NULL
or_num_df1$referrer <- NULL
or_num_df1$name <- NULL
or_num_df1$`Email Prefix` <- NULL
or_num_df1$order_count <- NULL


# Aggregating each order amount based on domain for or_num_df1
Ord_Num_vec <- or_num_df1$Order_Number
Ord_Num_vec1 <- ifelse((Ord_Num_vec == 1), c("Order_One_Amount"), c("Unknown"))
Ord_Num_vec2 <- ifelse((Ord_Num_vec == 2), c("Order_Two_Amount"), Ord_Num_vec1)
Ord_Num_vec3 <- ifelse((Ord_Num_vec == 3), c("Order_Three_Amount"), Ord_Num_vec2)
Ord_Num_vec4 <- ifelse((Ord_Num_vec == 4), c("Order_Four_Amount"), Ord_Num_vec3)
Ord_Num_vec5 <- ifelse((Ord_Num_vec == 5), c("Order_Five_Amount"), Ord_Num_vec4)
Ord_Num_vec6 <- ifelse((Ord_Num_vec == 6), c("Order_Six_Amount"), Ord_Num_vec5)
Ord_Num_vec7 <- ifelse((Ord_Num_vec == 7), c("Order_Seven_Amount"), Ord_Num_vec6)
Ord_Num_vec8 <- ifelse((Ord_Num_vec == 8), c("Order_Eight_Amount"), Ord_Num_vec7)
```

```
Ord_Num_vec9 <- ifelse((Ord_Num_vec == 9), c("Order_Nine_Amount"), Ord_Num_vec8)
Ord_Num_vec10 <- ifelse((Ord_Num_vec == 10), c("Order_Ten_Amount"), Ord_Num_vec9)
Ord_Num_vec11 <- ifelse((Ord_Num_vec == 11), c("Order_Eleven_Amount"), Ord_Num_vec10)
Ord_Num_vec12 <- ifelse((Ord_Num_vec == 12), c("Order_Twelve_Amount"), Ord_Num_vec11)
Ord_Num_vec13 <- ifelse((Ord_Num_vec == 13), c("Order_Thirteen_Amount"),
Ord_Num_vec12)
Ord_Num_vec14 <- ifelse((Ord_Num_vec == 14), c("Order_Fourteen_Amount"),
Ord_Num_vec13)
or_num_df1 <- mutate(or_num_df1, Order_Number = Ord_Num_vec14)

or_num_df1 <- spread(or_num_df1, Order_Number, total_sales, fill = 0)

or_num_df1 <- mutate(or_num_df1, total_revenue = Order_One_Amount + Order_Two_Amount
+
        Order_Three_Amount + Order_Four_Amount + Order_Five_Amount +
Order_Six_Amount +
        Order_Seven_Amount + Order_Eight_Amount + Order_Nine_Amount +
Order_Ten_Amount +
        Order_Eleven_Amount + Order_Twelve_Amount + Order_Thirteen_Amount +
Order_Fourteen_Amount)
```

company_df is created by joining c_df with or_num_df1 and then order_date_df

```
company_df <- inner_join(or_num_df1, c_df, by = "Company.ID")

company_df <- inner_join(company_df, order_date_df, by = "Company.ID")
```

Several variables (Emails Delivered, Emails Opened, and Emails Clicked) for each contact in the contact_df data frame were aggregated based on the Company.ID field and then joined to company_df:

```
contact_df <- mutate(contact_df, Count = 1)
contact_df <- ddply(contact_df, .(Company.ID), mutate, Domain_Count = cumsum(Count))
contact_df <- filter(contact_df, Domain_Count <= 12)


Contact_Num_vec <- contact_df$Domain_Count
Contact_Num_vec1 <- ifelse((Contact_Num_vec == 1), c("Contact_One"), c("Unknown"))
Contact_Num_vec2 <- ifelse((Contact_Num_vec == 2), c("Contact_Two"), Contact_Num_vec1)
Contact_Num_vec3 <- ifelse((Contact_Num_vec == 3), c("Contact_Three"), Contact_Num_vec2)
Contact_Num_vec4 <- ifelse((Contact_Num_vec == 4), c("Contact_Four"), Contact_Num_vec3)
Contact_Num_vec5 <- ifelse((Contact_Num_vec == 5), c("Contact_Five"), Contact_Num_vec4)
Contact_Num_vec6 <- ifelse((Contact_Num_vec == 6), c("Contact_Six"), Contact_Num_vec5)
Contact_Num_vec7 <- ifelse((Contact_Num_vec == 7), c("Contact_Seven"), Contact_Num_vec6)
Contact_Num_vec8 <- ifelse((Contact_Num_vec == 8), c("Contact_Eight"), Contact_Num_vec7)
Contact_Num_vec9 <- ifelse((Contact_Num_vec == 9), c("Contact_Nine"), Contact_Num_vec8)
Contact_Num_vec10 <- ifelse((Contact_Num_vec == 10), c("Contact_Ten"), Contact_Num_vec9)
```

```
Contact_Num_vec11 <- ifelse((Contact_Num_vec == 11), c("Contact_Eleven"),
Contact_Num_vec10)
Contact_Num_vec12 <- ifelse((Contact_Num_vec == 12), c("Contact_Twelve"),
Contact_Num_vec11)
contact_df <- mutate(contact_df, Contact_Count = Contact_Num_vec12)

str(emails_open_df)


# Creating Total_Emails_Delivered variable in company_df
contact_df$Emails.Delivered <-
as.numeric(levels(contact_df$Emails.Delivered))[contact_df$Emails.Delivered]
contact_df$Emails.Delivered <- ifelse(is.na(contact_df$Emails.Delivered), 0,
contact_df$Emails.Delivered)

emails_del_df <- select(contact_df, Company.ID, Emails.Delivered, Contact_Count)
emails_del_df <- spread(emails_del_df, Contact_Count, Emails.Delivered, fill = 0)
emails_del_df <- mutate(emails_del_df, Total_Emails_Delivered = Contact_One + Contact_Two
+
                 Contact_Three + Contact_Four + Contact_Five + Contact_Six +
                 Contact_Seven + Contact_Eight + Contact_Nine + Contact_Ten)
emails_del_df <- select(emails_del_df, Company.ID, Total_Emails_Delivered)
company_df <- inner_join(company_df, emails_del_df, by = "Company.ID")
company_df$Emails.Delivered <- NULL



# Creating Total_Emails_Opened variable in company_df
contact_df$Emails.Opened <-
as.numeric(levels(contact_df$Emails.Opened))[contact_df$Emails.Opened]
contact_df$Emails.Opened <- ifelse(is.na(contact_df$Emails.Opened), 0,
contact_df$Emails.Opened)

emails_open_df <- select(contact_df, Company.ID, Emails.Opened, Contact_Count)
emails_open_df <- spread(emails_open_df, Contact_Count, Emails.Opened, fill = 0)
emails_open_df <- mutate(emails_open_df, Total_Emails_Opened = Contact_One +
Contact_Two +
                  Contact_Three + Contact_Four + Contact_Five + Contact_Six +
                  Contact_Seven + Contact_Eight + Contact_Nine + Contact_Ten)
emails_open_df <- select(emails_open_df, Company.ID, Total_Emails_Opened)
company_df <- inner_join(company_df, emails_open_df, by = "Company.ID")



# Creating Total_Emails_Clicked variable in company_df
contact_df$Emails.Clicked <-
as.numeric(levels(contact_df$Emails.Clicked))[contact_df$Emails.Clicked]
```

```
contact_df$Emails.Clicked <- ifelse(is.na(contact_df$Emails.Clicked), 0,
contact_df$Emails.Clicked)


emails_clicked_df <- select(contact_df, Company.ID, Emails.Clicked, Contact_Count)
emails_clicked_df <- spread(emails_clicked_df, Contact_Count, Emails.Clicked, fill = 0)
emails_clicked_df <- mutate(emails_clicked_df, Total_Emails_Clicked = Contact_One +
Contact_Two +
                    Contact_Three + Contact_Four + Contact_Five + Contact_Six +
                    Contact_Seven + Contact_Eight + Contact_Nine + Contact_Ten)
emails_clicked_df <- select(emails_clicked_df, Company.ID, Total_Emails_Clicked)
company_df <- inner_join(company_df, emails_clicked_df, by = "Company.ID")
```

The following new variables were created:

total_revenue: The sum of the revenue from all orders related to each of the companies.

total_order_count: The sum of the number of orders related to each company.

Days_Between_All_Orders: The number of days between the first and last order from each company.

Ave_Days_Between_Orders: Total days between first and last order divided by Days_Between_All_Orders minus one.

days_since_last_order: The number of days since the most recent order for each company.

days_since_first_order: The number of days since the first order date for each company and today's date.

Ave_Order_Amount: The average amount spent per order for each company.

Ave_Reorder: The average amount spent per order (excluding the first order) for each company.

after_cutoff_date: A Boolean field where 1 represents companies that made their first order after 2016-01-12, and 0 represents all other companies.

Total_Emails_Delivered: The sum of emails delivered to contacts related to each company

Total_Emails_Opened: The sum of emails opened by contacts related to each company.

Total_Emails_Clicked: The sum of emails clicked by contacts related to each company.

Emails_Opened_Percent: Emails opened divided by emails delivered.

Emails_Clicked_Percent: Emails clicked divided by emails delivered.

Order_One_Date – Order_Fourteen_Date: Fourteen date fields for each of the company's order dates.

Order_One_Amount – Order_Fourteen_Amount: Fourteen fields for the revenue amount of each company's orders.

First_Order_Traffic_Source: The traffic source of each company's initial order.

Tidy Data

total_revenue	days_since_last_order	Industry	Total_Emails_Opened	Total_Emails_Delivered
2501.06	109	Unknown	2	3
2514.45	76	Real Estate	0	0
2532.12	125	Unknown	1	1
2541.20	26	Non-Profit Organization Management	0	1
2604.90	13	Non-Profit Organization Management	4	4
2691.45	75	Information Technology and Services	1	1
2778.76	50	Unknown	1	1
2812.51	127	Unknown	1	1
2822.43	21	Unknown	3	3
2965.94	13	Unknown	0	1
2975.64	21	Unknown	1	1
3149.93	55	Sports	5	5
3166.02	14	Unknown	0	0
3166.52	138	Unknown	0	3
3198.22	98	Construction	1	3
3336.14	74	Primary/Secondary Education	1	1
3349.22	28	Oil & Energy	2	2
3369.95	56	Computer Software	1	1
3385.46	85	Unknown	0	1
3744.73	97	Computer Software	1	1
3747.98	95	Telecommunications	0	1
4067.95	77	Machinery	2	5
4367.35	78	Banking	7	19
4556.18	63	Events Services	0	0
4641.43	47	Primary/Secondary Education	1	1
4667.20	22	Accounting	1	1
5119.70	117	Hospital & Health Care	3	6
5989.17	49	Architecture & Planning	0	2
6285.86	145	Information Technology and Services	0	2
7299.51	89	Unknown	1	2
7398.02	162	Unknown	1	1
7418.67	13	International Trade and Development	1	1
8561.58	109	Marketing and Advertising	2	8
11998.07	12	Market Research	1	6
12296.26	118	Unknown	2	5
14054.24	28	Music	2	7
14174.95	14	Renewables & Environment	5	5
14500.30	76	Consumer Goods	1	1
14995.92	47	Semiconductors	3	5
28050.00	48	Unknown	5	6

Analysis of Model

As mentioned previously, the total lifetime value of a Business to Business customer is what I am attempting to predict. The linear regression model will be created for this purpose using the dependent variable total_revenue which represents total lifetime value.

StandDesk began using their current CRM on January 13th, 2016. The data related to interactions between contacts and StandDesk is fairly accurate for companies that made a first purchase on January 13th or later. Unfortunately, this is not the case for the data related to companies that purchased prior to January 13th, 2016. This lack of accurate data has become a significant limitation to creating an accurate model.

Sub_company_df was created by filtering out companies with less than $2,500 in total revenue and any company that made a first purchase prior to Jan. 12th, 2016.

```
sub_company_df <- filter(company_df, total_revenue > 2500)
sub_company_df$after_cutoff_date <- ifelse(sub_company_df$Order_One_Date > "2016-01-12", 1, 0)
sub_company_df <- filter(sub_company_df, after_cutoff_date == 1)
```

The best model for predicting total revenue for sub_company_df:

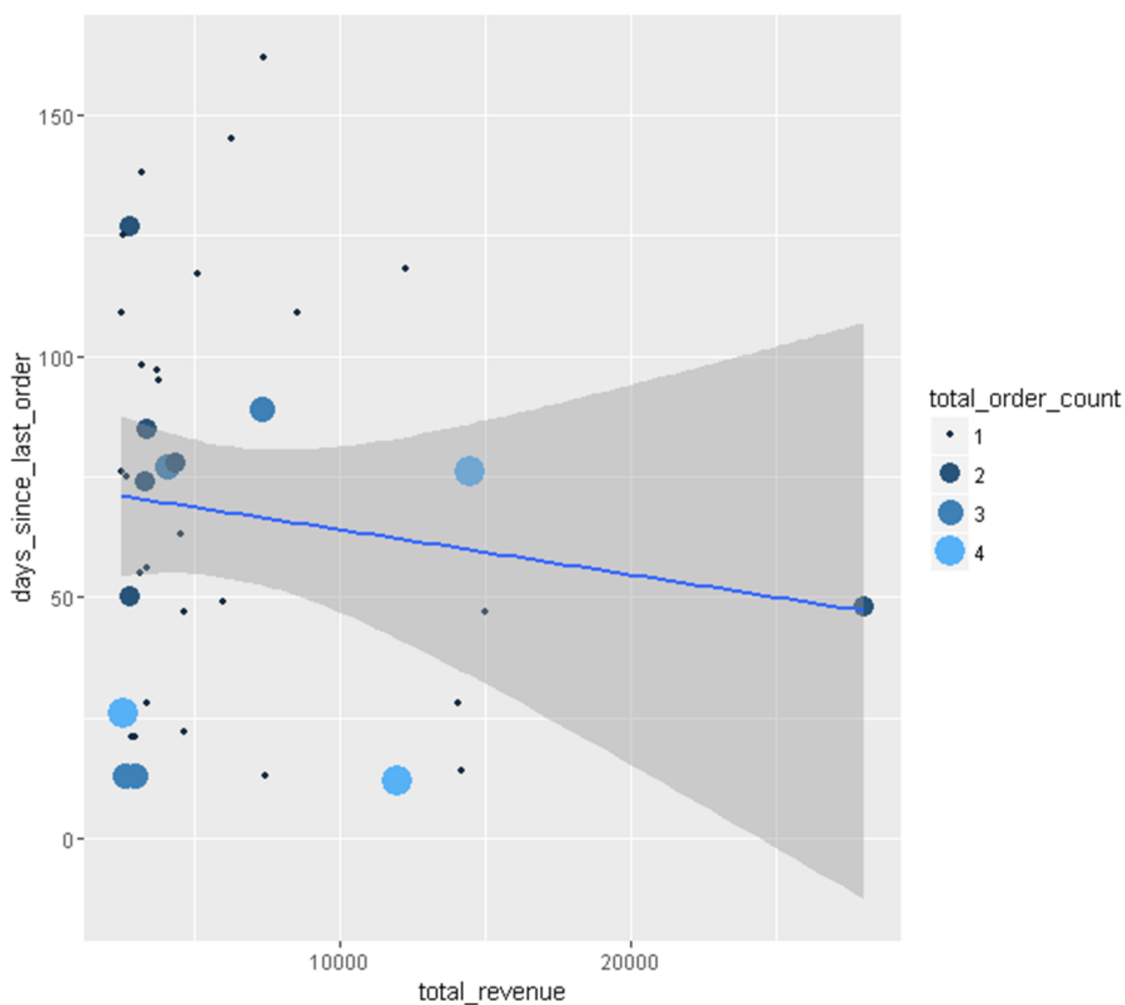
```
modelsub1 <- lm(total_revenue ~ days_since_last_order + Industry +
Total_Emails_Opened*Total_Emails_Delivered, data = sub_company_df)
```

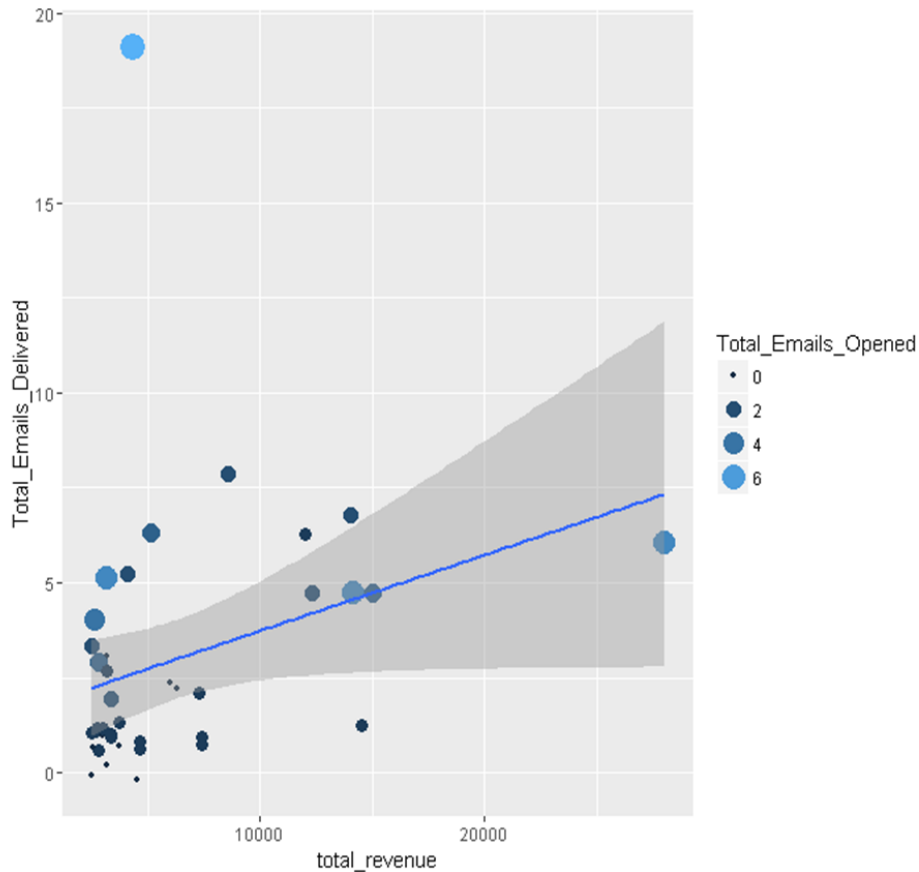
summary(modelsub1):

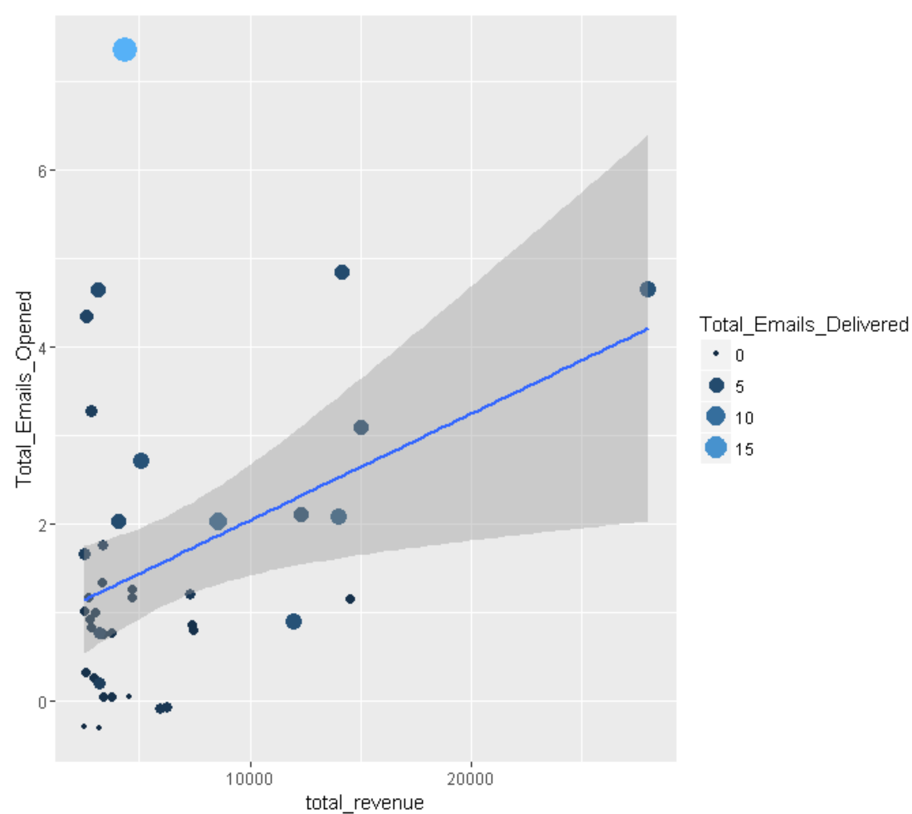
```
                                            Pr(>|t|)
(Intercept)                                 0.007163 **
days_since_last_order                       0.073802 .
IndustryArchitecture & Planning             0.655837
IndustryBanking                             2.31e-06 ***
IndustryComputer Software                   0.394870
IndustryConstruction                        0.171733
IndustryConsumer Goods                      0.025273 *
IndustryEvents Services                     0.162865
IndustryHospital & Health Care              0.000893 ***
IndustryInformation Technology and Services 0.256142
IndustryInternational Trade and Development 0.373463
IndustryMachinery                           0.033101 *
IndustryMarket Research                     0.433913
IndustryMarketing and Advertising           0.019597 *
IndustryMusic                               0.585836
IndustryNon-Profit Organization Management  0.018171 *
IndustryOil & Energy                        0.804494
IndustryPrimary/Secondary Education         0.561998
IndustryReal Estate                         0.053572 .
IndustryRenewables & Environment            0.046556 *
IndustrySemiconductors                      0.820650
IndustrySports                              0.000140 ***
IndustryTelecommunications                  0.112001
IndustryUnknown                             0.169794
```

```
Total_Emails_Opened                          0.000817 ***
Total_Emails_Delivered                       0.257120
Total_Emails_Opened:Total_Emails_Delivered   2.32e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2292 on 13 degrees of freedom
Multiple R-squared:  0.9356,    Adjusted R-squared:  0.8069
F-statistic: 7.269 on 26 and 13 DF,  p-value: 0.0002812
```




20
15
10
5
0
Total_Emails_Delivered
10000
20000
total_revenue
Total_Emails_Opened
0
2
4
6


6
4
2
0
Total_Emails_Opened
10000
20000
total_revenue
Total_Emails_Delivered
0
5
10
15

Recommended Use

Update the model to predict revenue for the next 60 days. Import new data once per month and run the model on updated data to determine which companies the model predicts will have an increase in total revenue.

Create a specific marketing campaign to reach out to companies that are predicted to have an increase in revenue.

Next Steps

I. Update Model to Predict Revenue for the next 30 days
II. Improve model by adding demographic data

Conclusion

Unfortunately, the final result of this project is that we currently have insufficient accurate data to use to predict future revenue of StandDesk's B2B customers. These results underscore the importance of tracking and maintaining accurate data in the 21st century economy. We have a myriad of ways of collecting and storing data related to potential and current customers and that data is extremely valuable if utilized properly. On the other hand, if it is not collected or stored properly, it has no value to us.