HW2_Relay_Data

Victoria Haley

2023-11-15

MAR653 HW2: Relay Retail

First, load and read the excel file into R and display the first 6 rows of data. This spreadsheet is different from the original file because I added 2 columns at the end:

1. createdxfirstoder, which is the difference between the date created and first order in days
2. custAge, which is the difference between the date created and the last order date in days.

```r
library(readxl)
Relay_Data <- read_excel("Relay_Data_HW2_F23.xlsx", sheet = "rawdata")
head(Relay_Data)
```

```
## # A tibble: 6 x 17
##   custid retained created             firstorder          lastorder
##   <chr>  <chr>    <dttm>              <dttm>              <dttm>
## 1 APCENR 1        2010-12-19 00:00:00 2011-04-01 00:00:00 2014-01-19 00:00:00
## 2 7UP6MS 0        2010-10-03 00:00:00 2010-12-01 00:00:00 2011-07-06 00:00:00
## 3 99XGVM 1        2011-01-24 00:00:00 2011-05-16 00:00:00 2014-01-16 00:00:00
## 4 YMALVV 1        2010-09-22 00:00:00 2010-11-18 00:00:00 2014-01-15 00:00:00
## 5 GW8NT7 1        2009-11-16 00:00:00 2011-05-09 00:00:00 2014-01-05 00:00:00
## 6 TFKLD4 1        2009-07-25 00:00:00 2010-11-15 00:00:00 2014-01-19 00:00:00
## # i 12 more variables: esent <dbl>, eopenrate <dbl>, eclickrate <dbl>,
## #   avgorder <dbl>, ordfreq <dbl>, paperless <dbl>, refill <dbl>,
## #   doorstep <dbl>, favday <chr>, city <chr>, createdxfirstorder <dbl>,
## #   custAge <dbl>
```

Next, view the summary statistics of the data.

```r
summary(Relay_Data)
```

```
##     custid            retained            created
##  Length:11760       Length:11760       Min.   :2008-06-17 00:00:00.00
##  Class :character   Class :character   1st Qu.:2011-10-16 00:00:00.00
##  Mode  :character   Mode  :character   Median :2013-05-09 00:00:00.00
##                                        Mean   :2013-04-30 00:45:11.01
##                                        3rd Qu.:2013-11-18 00:00:00.00
##                                        Max.   :2018-01-17 00:00:00.00
##   firstorder                       lastorder
##  Min.   :2008-08-05 00:00:00.00   Min.   :2008-08-19 00:00:00.00
##  1st Qu.:2011-12-13 00:00:00.00   1st Qu.:2013-03-14 00:00:00.00
##  Median :2013-06-23 00:00:00.00   Median :2013-11-17 00:00:00.00
##  Mean   :2013-06-24 03:55:28.16   Mean   :2014-02-15 23:05:30.60
##  3rd Qu.:2013-11-30 00:00:00.00   3rd Qu.:2014-01-16 00:00:00.00
##  Max.   :2018-01-17 00:00:00.00   Max.   :2018-01-21 00:00:00.00
```

```
##      esent           eopenrate        eclickrate        avgorder
##  Min.   :  0.00   Min.   :  0.000   Min.   :  0.000   Min.   :  0.01
##  1st Qu.: 22.00   1st Qu.:  2.128   1st Qu.:  0.000   1st Qu.: 46.88
##  Median : 38.00   Median : 16.667   Median :  2.273   Median : 62.12
##  Mean   : 32.65   Mean   : 27.608   Mean   :  6.571   Mean   : 73.08
##  3rd Qu.: 45.00   3rd Qu.: 45.455   3rd Qu.:  9.091   3rd Qu.: 88.03
##  Max.   :291.00   Max.   :100.000   Max.   :100.000   Max.   :651.35
##     ordfreq            paperless         refill          doorstep
##  Min.   :0.001238   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.028571   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.064516   Median :1.0000   Median :0.0000   Median :0.00000
##  Mean   :0.098612   Mean   :0.5267   Mean   :0.1105   Mean   :0.06216
##  3rd Qu.:0.124442   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :3.250000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##     favday              city           createdxfirstorder    custAge
##  Length:11760       Length:11760       Min.   :   0.00    Min.   :   1.0
##  Class :character   Class :character   1st Qu.:   0.00    1st Qu.:  42.0
##  Mode  :character   Mode  :character   Median :   6.00    Median : 125.5
##                                        Mean   :  57.37    Mean   : 292.2
##                                        3rd Qu.:  37.00    3rd Qu.: 435.0
##                                        Max.   :1651.00    Max.   :1991.0
```

Important things to note from the Summary:

- There are 11,760 customers and 17 different variables (including the ones that were added)
- custid, retained, paperless, refill, doorstep, favday, and city columns are all qualitative variables and will need to be converted into factor data types in order to do any type of analysis
- created, firstorder, and lastorder need to be converted into date format
- The average order time after creating an account is 57.37 days
- The average customer age is 292.2 days

Next, we'll need convert the columns of qualitative variables into factors and the created, firstorder, and lastorder columns into dates.

```r
#convert columns from character to factor
Relay_Data$custid <- as.factor(Relay_Data$custid)
Relay_Data$retained <- as.factor(Relay_Data$retained)
Relay_Data$paperless <- as.factor(Relay_Data$paperless)
Relay_Data$refill <- as.factor(Relay_Data$refill)
Relay_Data$doorstep <- as.factor(Relay_Data$doorstep)
Relay_Data$favday <- as.factor(Relay_Data$favday)
Relay_Data$city <- as.factor(Relay_Data$city)

#convert time variables to date
Relay_Data$created <- as.Date(Relay_Data$created, format="%Y-%m-%d")
Relay_Data$firstorder <- as.Date(Relay_Data$firstorder, format="%Y-%m-%d")
Relay_Data$lastorder <- as.Date(Relay_Data$lastorder, format="%Y-%m-%d")

#display classes of each column
sapply(Relay_Data, class)
```

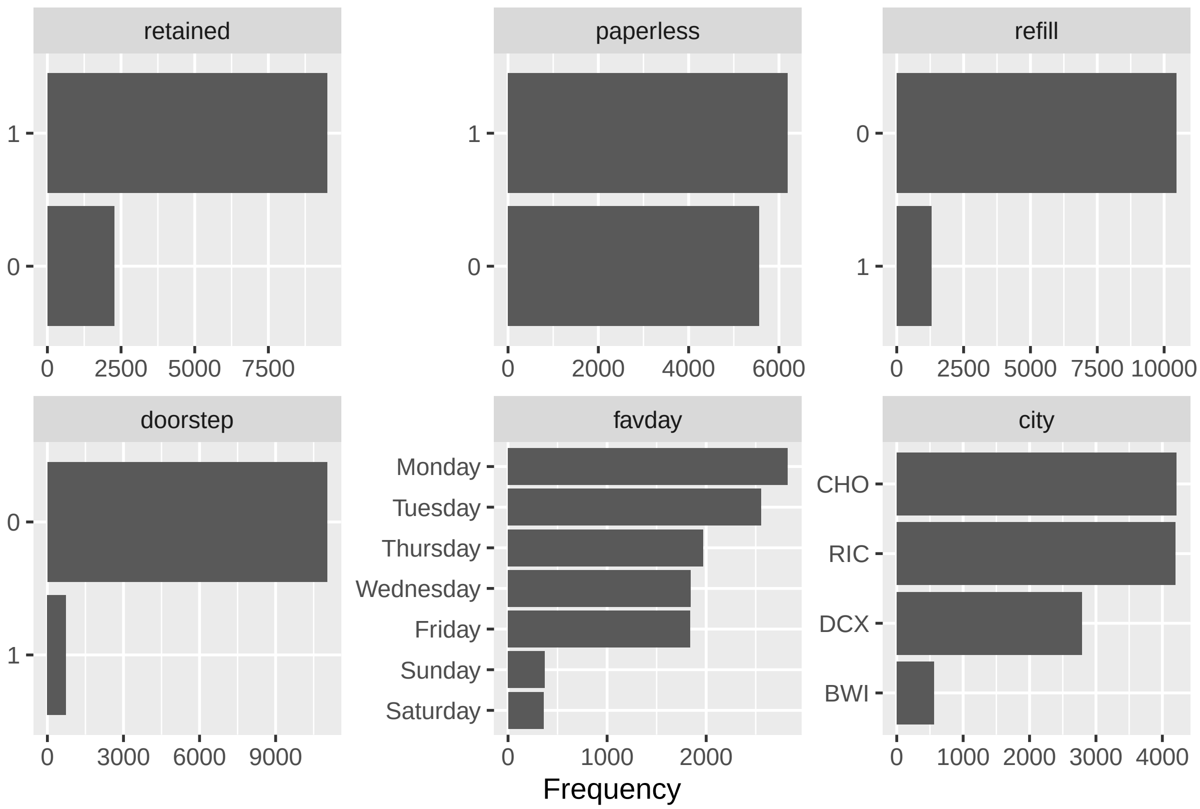
```
##          custid        retained         created      firstorder
##        "factor"        "factor"          "Date"          "Date"
##       lastorder           esent       eopenrate      eclickrate
##          "Date"       "numeric"       "numeric"       "numeric"
##        avgorder         ordfreq       paperless          refill
```

```
##          "numeric"          "numeric"          "factor"          "factor"
##           doorstep             favday              city createdxfirstorder
##           "factor"           "factor"          "factor"         "numeric"
##            custAge
##          "numeric"
```

Lastly in our exploratory data analysis are the visuals.

```r
library(DataExplorer)
plot_bar(Relay_Data) #plots categorical variables
```

```
## 4 columns ignored with more than 50 categories.
## custid: 11758 categories
## created: 2500 categories
## firstorder: 2378 categories
## lastorder: 1842 categories
```



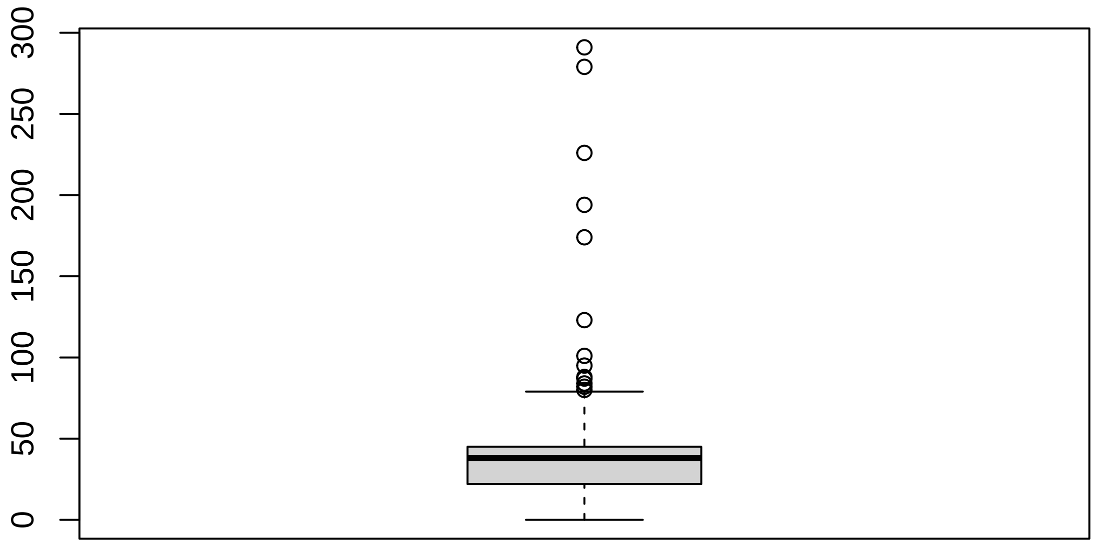
Notes:

- A small amount of customers were lost compared to those that were retained
- There is not a lot of difference between customers that subscribed to paperless communication vs. those that didn't
- A small amount of customers subscribed for automatic refill
- A very small amount of customers subscribed for doorstep delivery
- Customers preferred to have deliveries made on Mondays, followed closely by Tuesdays. The weekends are the least popular delivery days, and midweek delivery days are somewhat preferred.

- Most customers are located in Charlottesville or Richmond, while Baltimore has the least amount of customers.

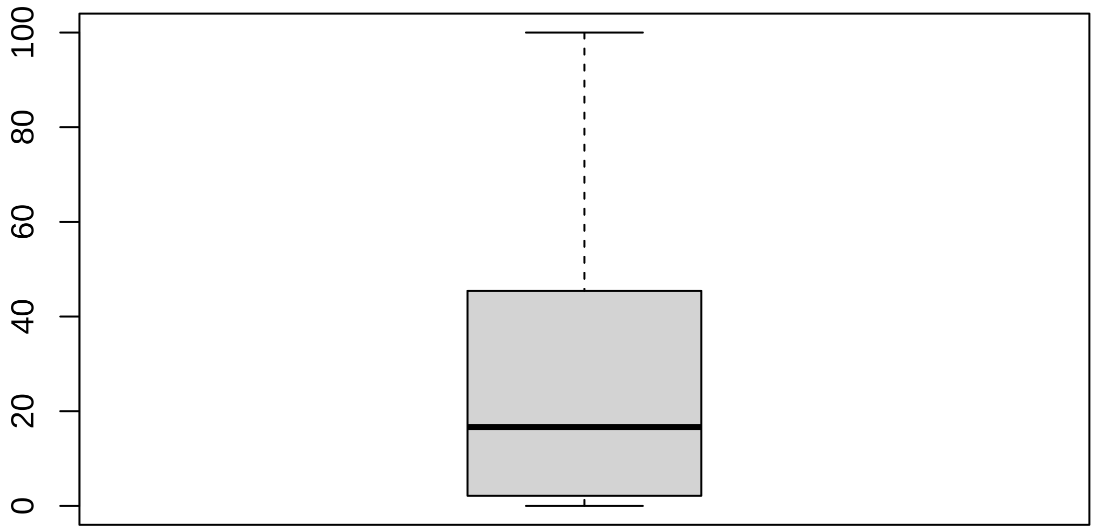
Plotting the numerical variables

```
boxplot(Relay_Data$esent,
        boxwex = 0.5,
        xlab= "Number of emails sent")
```



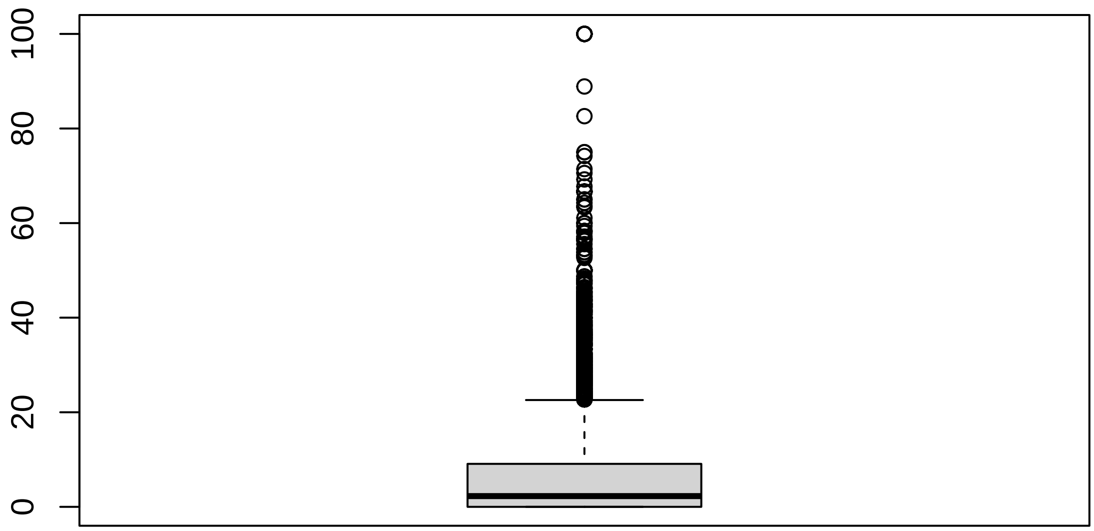
Number of emails sent

```
boxplot(Relay_Data$eopenrate,
        boxwex = 0.5,
        xlab= "Number of emails opened/Numer of emails sent")
```



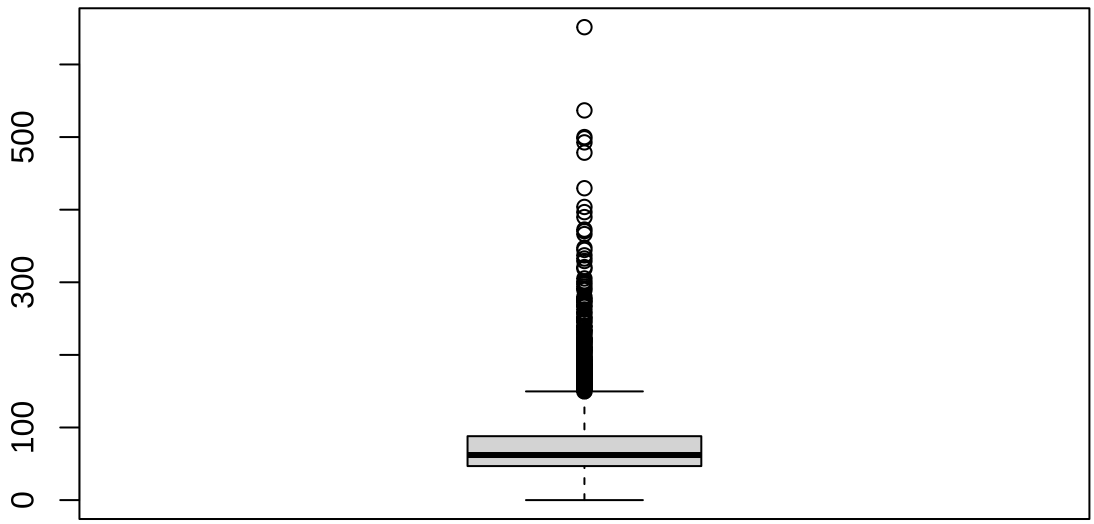
Number of emails opened/Numer of emails sent

```r
boxplot(Relay_Data$eclickrate,
        boxwex = 0.5,
        xlab= "Number of emails clicked/Numer of emails sent")
```



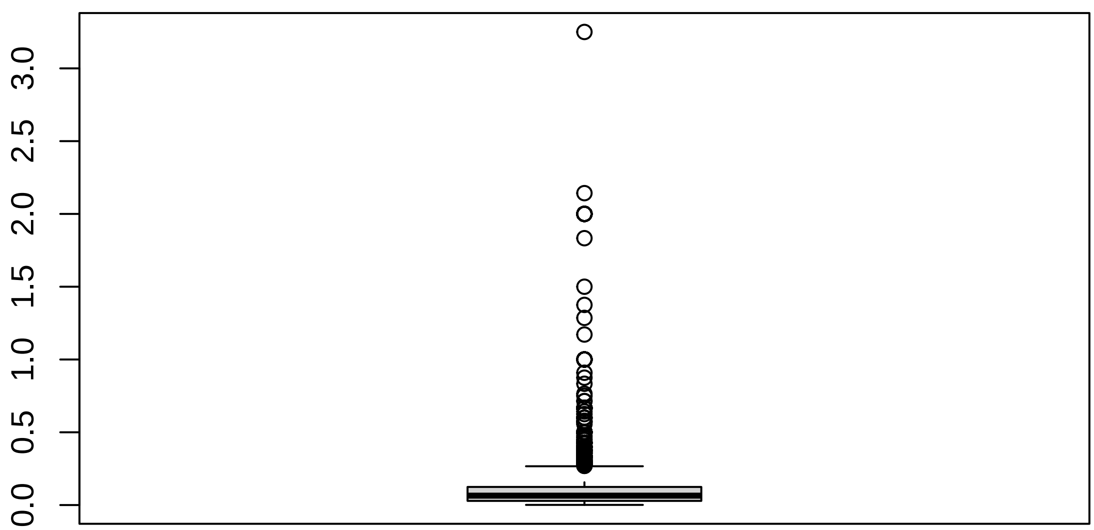
Number of emails clicked/Numer of emails sent

```r
boxplot(Relay_Data$avgorder,
        boxwex = 0.5,
        xlab= "Average Order Size per Customer")
```



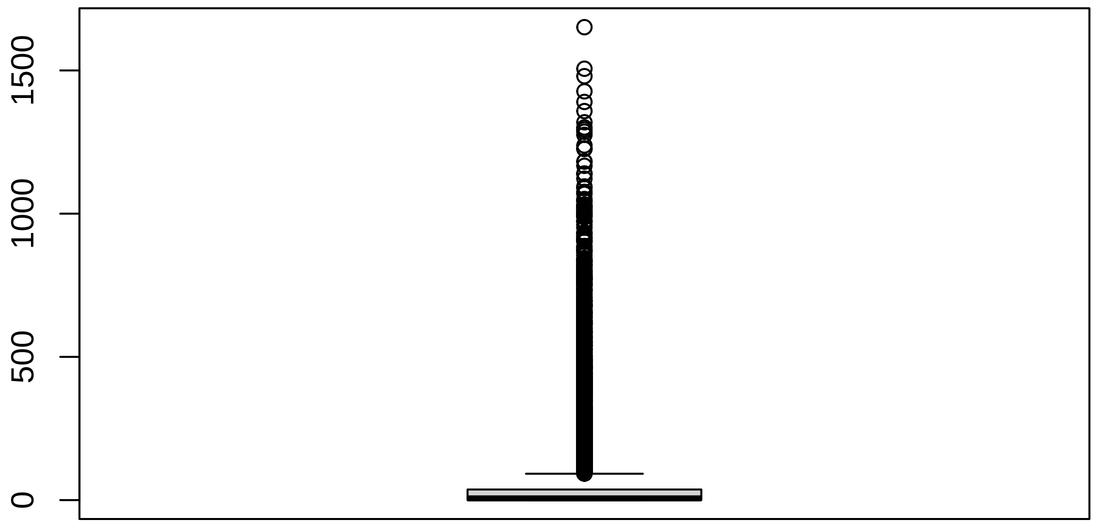
Average Order Size per Customer

```r
boxplot(Relay_Data$ordfreq,
        boxwex = 0.5,
        xlab = "Order Frequency")
```



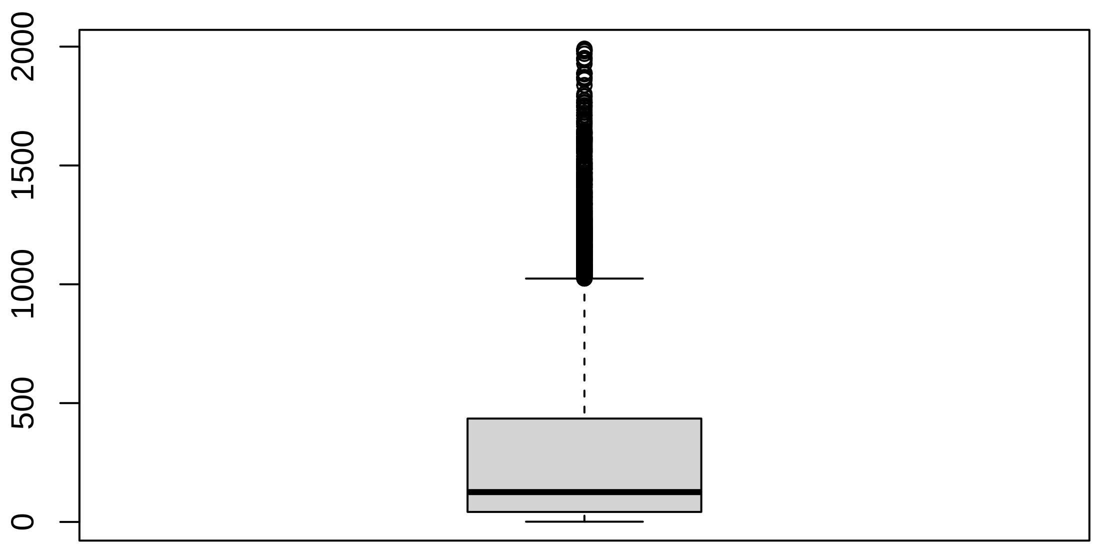
Order Frequency

```r
boxplot(Relay_Data$createdxfirstorder,
        boxwex = 0.5,
        xlab= "Days between Creation and First Order")
```



Days between Creation and First Order

```r
boxplot(Relay_Data$custAge,
        boxwex = 0.5,
        xlab= "Customer Tenure")
```



Customer Tenure

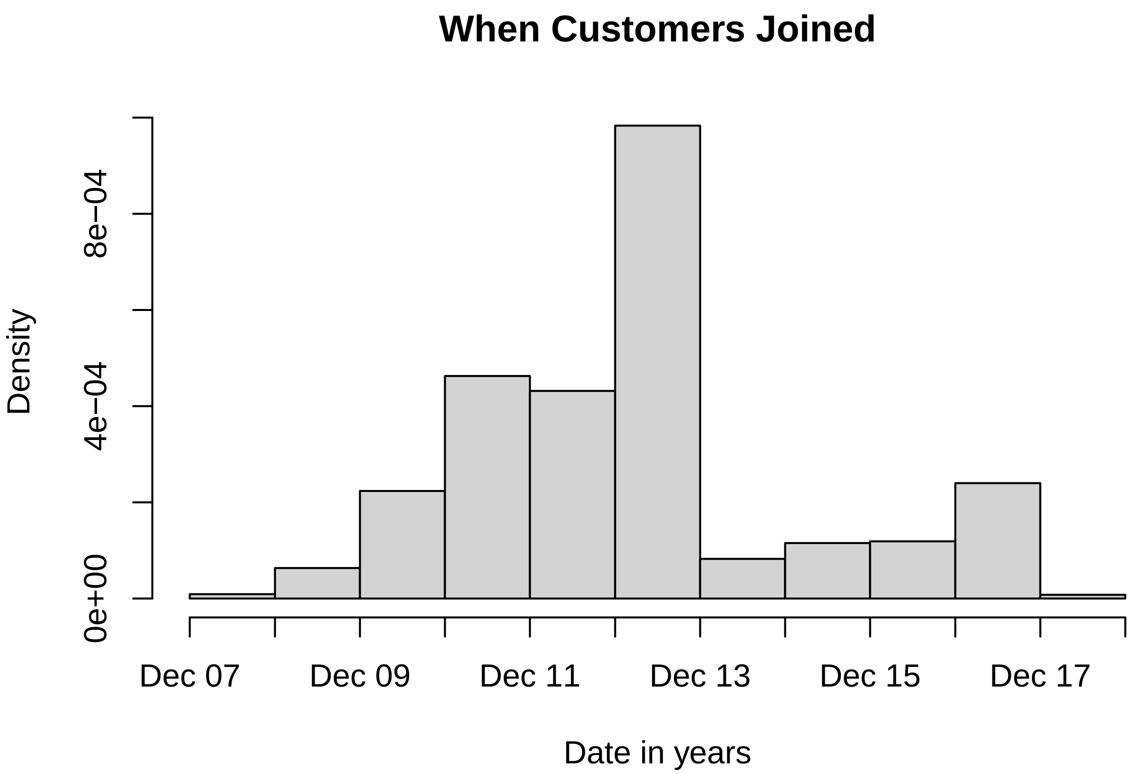
Notes:

- Number of emails sent ranges from 0 to 291, but most range from 22 to 45 with an average of ~33
- Number of emails opened/Number of emails sent ranges from 0 to 100, but most range from 2 to 25 with an average of ~28
- Number of emails clicked/Number of emails sent ranges from 0 to 100, but most range from 0 to 9 with an average of ~7

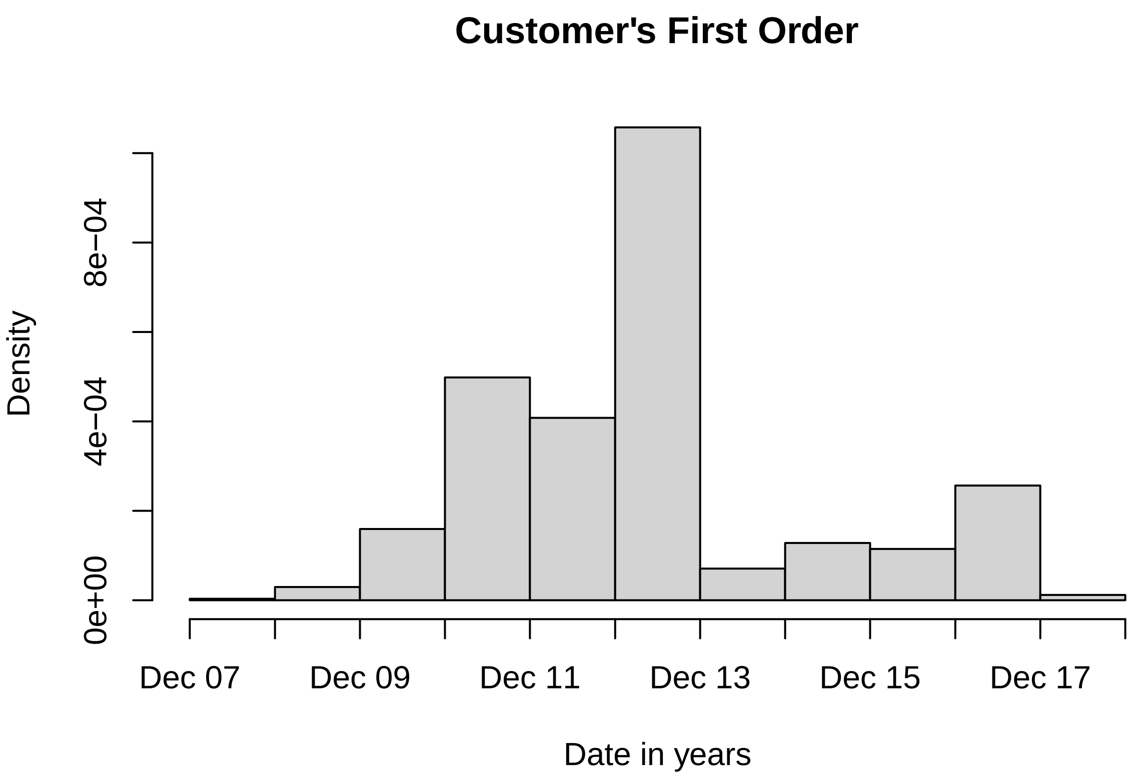
- Average Order Size per customer ranges from 0.1 to 651, but most range from 47 to 88 with an average of ~73
- Order Frequency ranges from 0 to 3.25, but most range from 0.3 to 0.12 with an average of ~0.1. Note that order frequency is Number of Orders/Customer Age
- Days between Creation and First Order (in days) ranges from 0 to 1651, but most range from 0 to 37 with an average of ~57
- Customer Tenure (in days) ranges from 1 to 1991, but most range from 42 to 435 with an average of ~292

Plotting Date Data

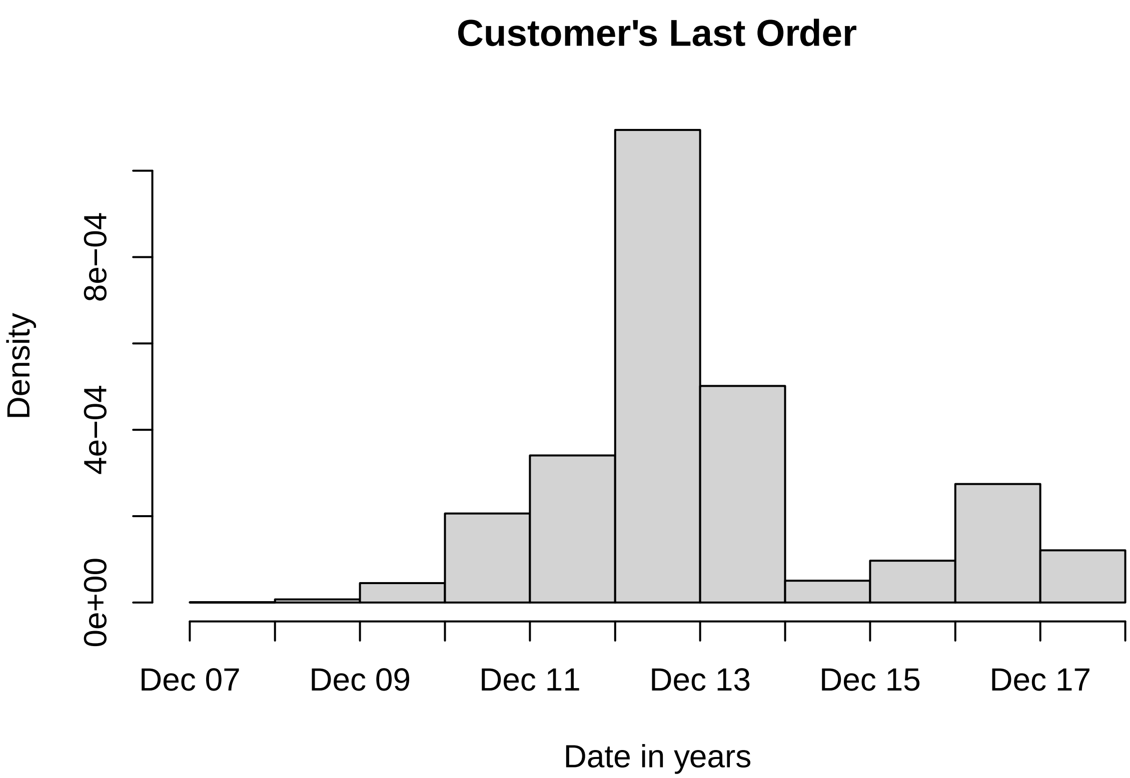
```
hist(Relay_Data$created, "years", format = "%b %y", main="When Customers Joined", xlab="Date in years")
```




```
hist(Relay_Data$firstorder, "years", format = "%b %y", main="Customer's First Order", xlab="Date in yea
```




```
hist(Relay_Data$lastorder, "years", format = "%b %y", main="Customer's Last Order", xlab="Date in years
```



Notes:

- Most customers joined and placed their first order between 2010 and 2013

- Most customers placed their last orders from 2011 to 2014, with a slight pickup from 2015 to 2017

Finally, we'll work on our logistic regression.

First step here is to split the data into a training and testing set. Here I chose the typical 80/20 random split.

```
library(caTools)
#make this example reproducible
set.seed(1)

#use 80% of dataset as training set and 20% as test set
sample <- sample.split(Relay_Data$retained, SplitRatio = 0.8)
train  <- subset(Relay_Data, sample == TRUE)
test   <- subset(Relay_Data, sample == FALSE)
```

Next, fit the model

Using the general linear model function and setting the family as binomial since we're trying to predict whether a customer is retained (1) or not (0).

For now, this chunk only addresses step 3 in the homework, and will only include esent, eclickrate, avgorder, ordfreq, paperless, refill, doorstep as independent variables.

```
#fit logistic regression model
model <- glm(retained~esent+eclickrate+avgorder+ordfreq+paperless+refill+doorstep, family="binomial", da
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#disable scientific notation for model summary
options(scipen=999)

#view model summary
summary(model)
```

```
##
## Call:
## glm(formula = retained ~ esent + eclickrate + avgorder + ordfreq +
##     paperless + refill + doorstep, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1612   0.0209   0.0512   0.1227   2.4459
##
## Coefficients:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept) -3.214291   0.157557 -20.401 < 0.0000000000000002 ***
## esent        0.220928   0.005767  38.306 < 0.0000000000000002 ***
## eclickrate   0.020289   0.003776   5.374         0.0000000772 ***
## avgorder    -0.001951   0.001357  -1.437                0.151
## ordfreq      1.471298   0.309786   4.749         0.0000020402 ***
## paperless1  -0.108221   0.129288  -0.837                0.403
## refill1      0.216335   0.189740   1.140                0.254
## doorstep1    0.243983   0.240214   1.016                0.310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 9225.1  on 9407  degrees of freedom
## Residual deviance: 2196.1  on 9400  degrees of freedom
## AIC: 2212.1
##
## Number of Fisher Scoring iterations: 8
```

Interpretation: esent: A one unit increase here increases the log odds of retention by 0.22. With an incredibly low p-value, this is an important predictor.

eclickrate: A one unit increase here increases the log odds of retention by 0.20. This also has an incredibly low p-value, so it is also an important predictor.

avgorder: A one unit increase here decreases the log odds of retention by 0.002. However, it has a high p-value meaning that it is not a statistically significant predictor.

ordfreq: A one unit increase here increases the log odds of retention by 1.47. With an incredibly low p-value, this is another important predictor.

paperless: A one unit increase here decreases the log odds of retention by 0.11. However, it has a high p-value meaning that it is not a statistically significant predictor.

refill: A one unit increase here increases the log odds of retention by 0.22. However, it has a high p-value meaning that it is not a statistically significant predictor.

doorstep: A one unit increase here increases the log odds of retention by 0.24. However, it has a high p-value meaning that it is not a statistically significant predictor.

*Note that the categorical variables were split into dummy categories.

Assessing Model Fit For logistic regression

Using the McFadden R^2 to asses how well our model fit the data

```r
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
pscl::pR2(model)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.7619448
```

At 0.762, the model does very well and has high predictive power.

Variable Importance

```r
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
caret::varImp(model)
```

```
##                Overall
## esent      38.3062348
## eclickrate  5.3736462
## avgorder    1.4371349
## ordfreq     4.7494011
## paperless1  0.8370552
## refill1     1.1401616
## doorstep1   1.0156903
```

This just confirms what our p-values indicated.

esent clearly is the most important variable, followed by elickrate and ordfreq.

Checking for multicollinearity

```r
#calculate VIF values for each predictor variable in our model
car::vif(model)
```

```
##      esent eclickrate   avgorder    ordfreq  paperless     refill   doorstep
##   1.051711   1.201614   1.034503   1.035505   1.285850   1.172784   1.129180
```

A VIF >5 indicates severe multicollinearity. The values here are well below 5, so there is no multicollinearity issue here.

Making predictions on the test data

```r
#predict probability of churning
predicted <- predict(model, test, type = 'response')
p_class <- ifelse(predicted > 0.5, "1", "0")

confusionMatrix(test$retained, factor(p_class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  410   44
##          1   25 1873
##
##                Accuracy : 0.9707
##                  95% CI : (0.963, 0.9771)
##     No Information Rate : 0.8151
##     P-Value [Acc > NIR] : < 0.0000000000000002
##
##                   Kappa : 0.9043
##
##  Mcnemar's Test P-Value : 0.03024
##
##             Sensitivity : 0.9425
##             Specificity : 0.9770
##          Pos Pred Value : 0.9031
##          Neg Pred Value : 0.9868
##              Prevalence : 0.1849
##          Detection Rate : 0.1743
```

```
##    Detection Prevalence : 0.1930
##       Balanced Accuracy : 0.9598
##
##        'Positive' Class : 0
##
```

- True positives: 1873, sensitivity (TP rate): 94%
- False positives: 25
- True negatives: 410, specificity (TN rate): 97%
- False negatives: 44
- Accuracy: 97.07%

Overall, this model does pretty well at detecting which customers churned.

New Model

Adding favday and city to the above model.

```r
#fit logistic regression model
model2 <- glm(retained~esent+eclickrate+avgorder+ordfreq+paperless+refill+doorstep+favday+city, family=
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#disable scientific notation for model summary
options(scipen=999)

#view model summary
summary(model2)
```

```
##
## Call:
## glm(formula = retained ~ esent + eclickrate + avgorder + ordfreq +
##     paperless + refill + doorstep + favday + city, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1889   0.0203   0.0500   0.1227   2.3881
##
## Coefficients:
##                  Estimate Std. Error z value             Pr(>|z|)
## (Intercept)     -3.023125   0.348217  -8.682 < 0.0000000000000002 ***
## esent            0.222074   0.005833  38.070 < 0.0000000000000002 ***
## eclickrate       0.022288   0.003860   5.774        0.00000000774 ***
## avgorder        -0.002195   0.001372  -1.600                0.110
## ordfreq          1.510595   0.315881   4.782        0.00000173424 ***
## paperless1      -0.011775   0.139248  -0.085                0.933
## refill1          0.123156   0.193235   0.637                0.524
## doorstep1        0.368070   0.249752   1.474                0.141
## favdayMonday    -0.136115   0.185976  -0.732                0.464
## favdaySaturday  -0.529365   0.393031  -1.347                0.178
## favdaySunday    -0.097237   0.382074  -0.254                0.799
## favdayThursday  -0.167581   0.199848  -0.839                0.402
## favdayTuesday   -0.281367   0.186205  -1.511                0.131
```

```
## favdayWednesday -0.068490   0.202689  -0.338                0.735
## cityCHO         -0.134891   0.287817  -0.469                0.639
## cityDCX         -0.404653   0.272828  -1.483                0.138
## cityRIC          0.119070   0.287437   0.414                0.679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9225.1  on 9407  degrees of freedom
## Residual deviance: 2180.7  on 9391  degrees of freedom
## AIC: 2214.7
##
## Number of Fisher Scoring iterations: 8
```

Interpretations esent: No change from first model.

eclickrate: The coefficient went from 0.20 to 0.22, no changes other than that.

avgorder: No change from first model.

ordfreq: The coefficient went from 1.47 to 1.51, no changes other than that.

paperless: No change from first model.

refill: The coefficient went from 0.22 to 0.12, no changes other than that.

doorstep: The coefficient went from 0.24 to 0.37, no changes other than that.

favdayMonday: A one unit increase here decreases the log odds of retention by 0.13. However, it has a high p-value meaning that it is not a statistically significant predictor.

favdaySaturday: A one unit increase here decreases the log odds of retention by 0.53. However, it has a high p-value meaning that it is not a statistically significant predictor.

favdaySunday: A one unit increase here decreases the log odds of retention by 0.1. However, it has a high p-value meaning that it is not a statistically significant predictor.

favdayThursday: A one unit increase here decreases the log odds of retention by 0.17. However, it has a high p-value meaning that it is not a statistically significant predictor.

favdayTuesday: A one unit increase here decreases the log odds of retention by 0.28. However, it has a high p-value meaning that it is not a statistically significant predictor.

favdayWednesday: A one unit increase here decreases the log odds of retention by 0.07. However, it has a high p-value meaning that it is not a statistically significant predictor.

cityCHO: A one unit increase here decreases the log odds of retention by 0.13. However, it has a high p-value meaning that it is not a statistically significant predictor.

cityDCX: A one unit increase here decreases the log odds of retention by 0.40. However, it has a high p-value meaning that it is not a statistically significant predictor.

cityRIC: A one unit increase here increases the log odds of retention by 0.12. However, it has a high p-value meaning that it is not a statistically significant predictor.

Now, let's check the McFadden $\mathbf{R^2}$

```r
pscl::pR2(model)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.7619448
```

At 0.763, the model does slightly better than the first model.

Variable importance of new model

```r
caret::varImp(model2)
```

```
##                      Overall
## esent            38.06952351
## eclickrate        5.77408221
## avgorder          1.60004257
## ordfreq           4.78215686
## paperless1        0.08456227
## refill1           0.63733743
## doorstep1         1.47374171
## favdayMonday      0.73189562
## favdaySaturday    1.34687821
## favdaySunday      0.25449848
## favdayThursday    0.83854357
## favdayTuesday     1.51106187
## favdayWednesday   0.33790510
## cityCHO           0.46866972
## cityDCX           1.48317917
## cityRIC           0.41424545
```

More clear visualization of which variables actually influence the model.

Checking for multicollinearity in new model

```r
#calculate VIF values for each predictor variable in our model
car::vif(model2)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## esent      1.072229  1        1.035485
## eclickrate 1.251454  1        1.118684
## avgorder   1.049607  1        1.024503
## ordfreq    1.040397  1        1.019998
## paperless  1.479530  1        1.216360
## refill     1.198222  1        1.094633
## doorstep   1.218543  1        1.103876
## favday     1.341036  6        1.024755
## city       1.743104  3        1.097035
```

By looking at either of the GVIF columns, we can see that all variables are well below 5 and there is no multicollinearity in the model.

Make predictions on test data

```r
#predict probability of churning
predicted <- predict(model2, test, type = 'response')
p_class <- ifelse(predicted > 0.5, "1", "0")

confusionMatrix(test$retained, factor(p_class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  411   43
##          1   27 1871
##
##                Accuracy : 0.9702
##                  95% CI : (0.9625, 0.9767)
##     No Information Rate : 0.8138
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.9032
##
##  Mcnemar's Test P-Value : 0.073
##
##             Sensitivity : 0.9384
##             Specificity : 0.9775
##          Pos Pred Value : 0.9053
##          Neg Pred Value : 0.9858
##              Prevalence : 0.1862
##          Detection Rate : 0.1747
##    Detection Prevalence : 0.1930
##       Balanced Accuracy : 0.9579
##
##        'Positive' Class : 0
##
```

- True positives: 1871, sensitivity (TP rate): 93.8%
- False positives: 43
- True negatives: 411, specificity (TN rate): 98%
- False negatives: 27
- Accuracy: 97.02%

Overall, this model does just as well at detecting which customers churned. New Model without least important variables (paperless and refill) and adding the 2 new columns

```r
#fit logistic regression model
model3 <- glm(retained~esent+eclickrate+avgorder+ordfreq+doorstep+favday+city+createdxfirstorder+custAge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#disable scientific notation for model summary
options(scipen=999)

#view model summary
summary(model3)
```

```
##
## Call:
## glm(formula = retained ~ esent + eclickrate + avgorder + ordfreq +
##     doorstep + favday + city + createdxfirstorder + custAge,
##     family = "binomial", data = train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0869   0.0245   0.0538   0.1260   2.7971
##
## Coefficients:
##                      Estimate Std. Error z value            Pr(>|z|)
## (Intercept)        -2.8641174  0.3403477  -8.415 < 0.0000000000000002 ***
## esent               0.2262680  0.0058774  38.498 < 0.0000000000000002 ***
## eclickrate          0.0244963  0.0038431   6.374       0.000000000184 ***
## avgorder           -0.0010968  0.0014098  -0.778              0.43659
## ordfreq             0.9124648  0.2805063   3.253              0.00114 **
## doorstep1           0.4789521  0.2493379   1.921              0.05474 .
## favdayMonday       -0.1818497  0.1920584  -0.947              0.34372
## favdaySaturday     -0.5907944  0.4033261  -1.465              0.14297
## favdaySunday       -0.1122348  0.3799903  -0.295              0.76772
## favdayThursday     -0.2782477  0.2052618  -1.356              0.17523
## favdayTuesday      -0.3762017  0.1920205  -1.959              0.05009 .
## favdayWednesday    -0.1620470  0.2086359  -0.777              0.43734
## cityCHO             0.2855584  0.2895629   0.986              0.32405
## cityDCX            -0.4333941  0.2768927  -1.565              0.11753
## cityRIC             0.3088056  0.2879083   1.073              0.28346
## createdxfirstorder  0.0010444  0.0005232   1.996              0.04590 *
## custAge            -0.0022625  0.0002602  -8.694 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9225.1  on 9407  degrees of freedom
## Residual deviance: 2093.7  on 9391  degrees of freedom
## AIC: 2127.7
##
## Number of Fisher Scoring iterations: 8
```

Interpretations: esent: The coefficient went from 0.22 to 0.23, no other changes from previous model.

eclickrate: The coefficient went from 0.22 to 0.02, no other changes from previous model.

avgorder: The coefficient went from -0.0022 to -0.001, and is not statistically anymore.

ordfreq: The coefficient went from 1.47 to 0.91, no other changes from previous model.

doorstep: The coefficient went from 0.24 to 0.47, no other changes from previous model.

favdayMonday: The coefficient went from -0.13 to -0.18, no other changes from previous model.

avdaySaturday: The coefficient went from -0.53 to -0.59, no other changes from previous model.

favdaySunday: The coefficient went from -0.09 to -0.11, no other changes from previous model.

favdayThursday: The coefficient went from -0.17 to -0.27, no other changes from previous model.

favdayTuesday: The coefficient went from -0.28 to -0.38, no other changes from previous model.

favdayWednesday: The coefficient went from -0.07 to -0.16, no other changes from previous model.

cityCHO: The coefficient went from -0.13 to 0.29. However, it now has a high p-value again meaning that it is not a statistically significant predictor.

cityDCX: The coefficient went from -0.4 to -0.43. However, it now has a high p-value again meaning that it is not a statistically significant predictor.

cityRIC: The coefficient went from 0.12 to 0.31. However, it now has a high p-value again meaning that it is not a statistically significant predictor.

createdxfirstorder: A one unit increase here increases the log odds of retention by 0.001. It has a p-value of <0.05, meaning that it is statistically significant.

custAge: A one unit increase here decreases the log odds of retention by 0.002. It has an incredibly low p-value, meaning that it is statistically significant.

Now, let's check the McFadden R2

```r
pscl::pR2(model3)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.7730382
```

Despite the amount of statistically insignificant variables, this model is slightly more accurate than the previous ones.

Variable Importance

```r
coeff <-caret::varImp(model3)
coeffDF <- data.frame(coeff)
library(tibble)
coeffDF <- tibble::rownames_to_column(coeffDF, "Variable")
head(coeffDF)
```

```
##      Variable    Overall
## 1       esent 38.4982893
## 2  eclickrate  6.3741486
## 3    avgorder  0.7779632
## 4     ordfreq  3.2529202
## 5   doorstep1  1.9208955
## 6 favdayMonday  0.9468460
```

Not much has changed, but customer tenure is pretty important.

Checking for multicolllinearity

```r
#calculate VIF values for each predictor variable in our model
car::vif(model3)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## esent              1.145087  1        1.070088
## eclickrate         1.150847  1        1.072775
## avgorder           1.047961  1        1.023700
## ordfreq            1.081898  1        1.040143
## doorstep           1.128166  1        1.062152
## favday             1.297514  6        1.021941
## city               1.720800  3        1.094683
## createdxfirstorder 1.487094  1        1.219465
## custAge            1.799688  1        1.341525
```

No multicollinearity here either.

Predictions

```
#predict probability of churning
predicted <- predict(model3, test, type = 'response')
p_class <- ifelse(predicted > 0.5, "1", "0")

confusionMatrix(test$retained, factor(p_class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  414   40
##          1   28 1870
##
##                Accuracy : 0.9711
##                  95% CI : (0.9635, 0.9775)
##     No Information Rate : 0.8121
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.9063
##
##  Mcnemar's Test P-Value : 0.1822
##
##             Sensitivity : 0.9367
##             Specificity : 0.9791
##          Pos Pred Value : 0.9119
##          Neg Pred Value : 0.9852
##              Prevalence : 0.1879
##          Detection Rate : 0.1760
##    Detection Prevalence : 0.1930
##       Balanced Accuracy : 0.9579
##
##        'Positive' Class : 0
##
```

- True positives: 1870, sensitivity (TP rate): 93.7%
- False positives: 28
- True negatives: 414, specificity (TN rate): 97.9%
- False negatives: 40
- Accuracy: 97.11%

Overall, this model does just as well at detecting which customers churned.

Model with all variables

```
#fit logistic regression model
model4 <- glm(retained~esent+eclickrate+avgorder+ordfreq+doorstep+favday+city+paperless+refill+createdx
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

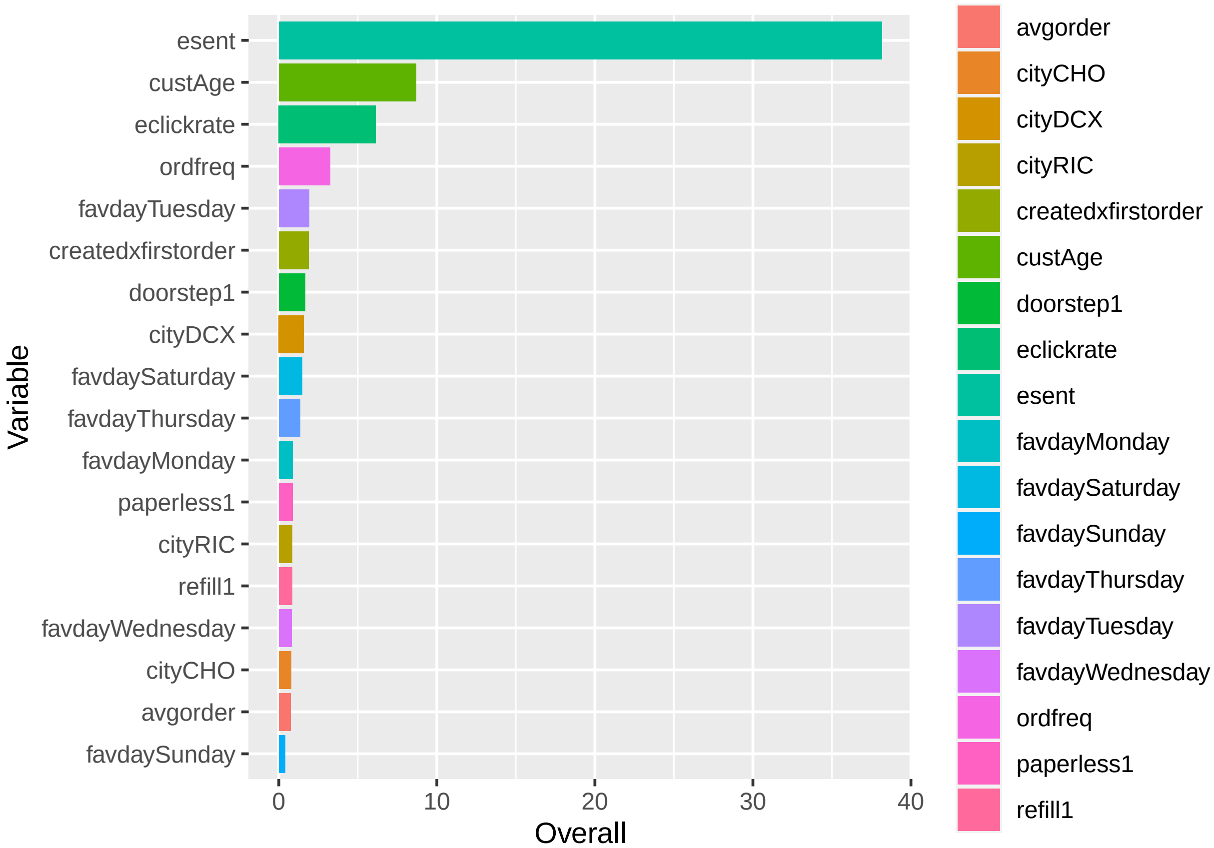
```r
#disable scientific notation for model summary
options(scipen=999)

summary(model4)
```

```
##
## Call:
## glm(formula = retained ~ esent + eclickrate + avgorder + ordfreq +
##     doorstep + favday + city + paperless + refill + createdxfirstorder +
##     custAge, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0706   0.0243   0.0539   0.1260   2.7850
##
## Coefficients:
##                       Estimate Std. Error z value             Pr(>|z|)
## (Intercept)         -2.7825001  0.3545498  -7.848  0.00000000000000423 ***
## esent                0.2266559  0.0059399  38.158 < 0.0000000000000002 ***
## eclickrate           0.0248228  0.0040453   6.136  0.00000000084530945 ***
## avgorder            -0.0010514  0.0014177  -0.742              0.45830
## ordfreq              0.9153406  0.2815679   3.251              0.00115 **
## doorstep1            0.4375959  0.2612116   1.675              0.09388 .
## favdayMonday        -0.1693042  0.1926587  -0.879              0.37952
## favdaySaturday      -0.5929680  0.4027961  -1.472              0.14099
## favdaySunday        -0.1563495  0.3852791  -0.406              0.68488
## favdayThursday      -0.2785711  0.2054324  -1.356              0.17509
## favdayTuesday       -0.3691057  0.1922396  -1.920              0.05485 .
## favdayWednesday     -0.1663307  0.2089227  -0.796              0.42595
## cityCHO              0.2280443  0.2966122   0.769              0.44199
## cityDCX             -0.4397976  0.2776185  -1.584              0.11315
## cityRIC              0.2508935  0.2939084   0.854              0.39330
## paperless1          -0.1277611  0.1454295  -0.879              0.37967
## refill1              0.1698829  0.2056664   0.826              0.40880
## createdxfirstorder   0.0009834  0.0005256   1.871              0.06137 .
## custAge             -0.0022687  0.0002612  -8.685 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9225.1  on 9407  degrees of freedom
## Residual deviance: 2092.5  on 9389  degrees of freedom
## AIC: 2130.5
##
## Number of Fisher Scoring iterations: 8
```

Important Variables

```r
#variable importance
coeff <-caret::varImp(model4)
coeffDF <- data.frame(coeff)
coeffDF <- tibble::rownames_to_column(coeffDF, "Variable")
```

```r
library(forcats)
ggplot(coeffDF, aes(x = fct_reorder(Variable, Overall) , y = Overall, fill = Variable)) + geom_col() + 
```



McFadden R^2

```r
pscl::pR2(model4)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
##  McFadden
## 0.7731728
```

Slightly more accurate than previous model

Predictions

```r
#predict probability of churning
predicted <- predict(model4, test, type = 'response')
p_class <- ifelse(predicted > 0.5, "1", "0")

cm <- confusionMatrix(test$retained, factor(p_class))
cm
```

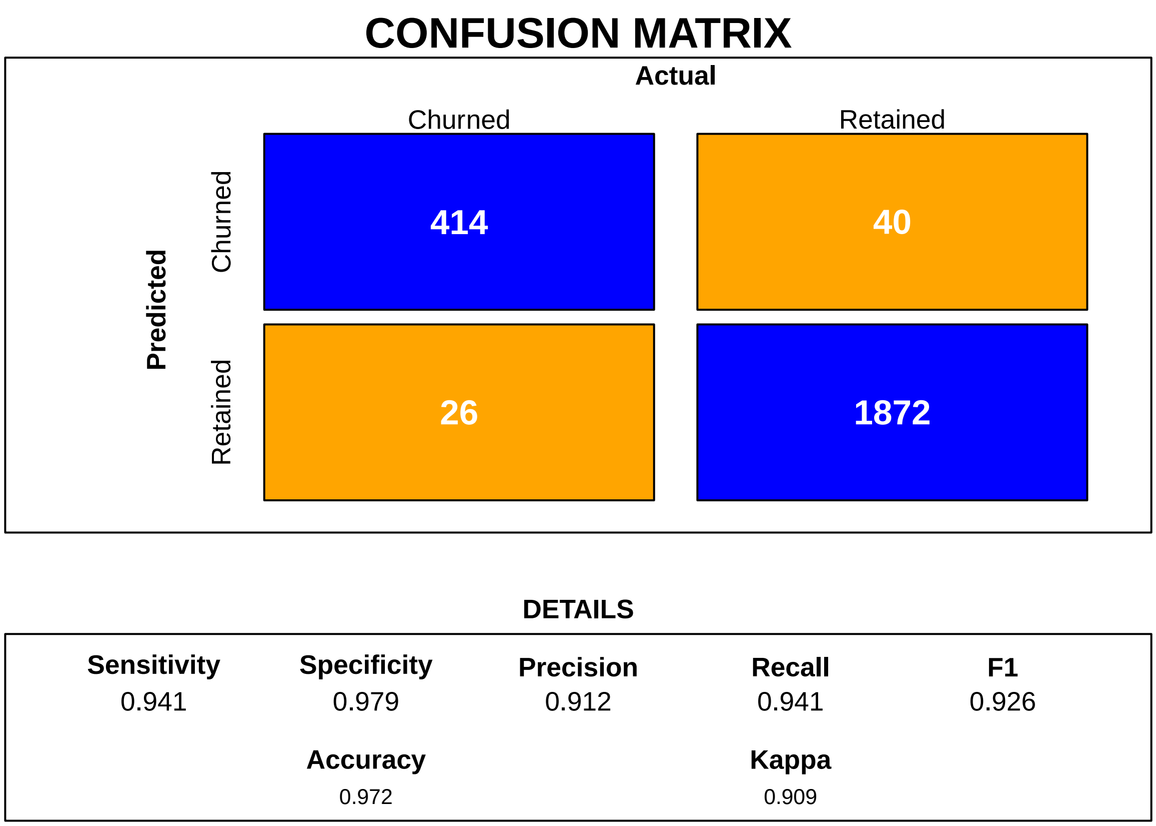
```
## Confusion Matrix and Statistics
##
##          Reference
```

```
## Prediction    0    1
##          0  414   40
##          1   26 1872
##
##                Accuracy : 0.9719
##                  95% CI : (0.9644, 0.9782)
##     No Information Rate : 0.8129
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.9089
##
##  Mcnemar's Test P-Value : 0.1096
##
##             Sensitivity : 0.9409
##             Specificity : 0.9791
##          Pos Pred Value : 0.9119
##          Neg Pred Value : 0.9863
##              Prevalence : 0.1871
##          Detection Rate : 0.1760
##    Detection Prevalence : 0.1930
##       Balanced Accuracy : 0.9600
##
##        'Positive' Class : 0
##
```

Plotting the Confusion Matrix

```r
draw_confusion_matrix <- function(cm) {

  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  rect(150, 430, 240, 370, col='blue')
  text(195, 435, 'Churned', cex=1.2)
  rect(250, 430, 340, 370, col='orange')
  text(295, 435, 'Retained', cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='orange')
  rect(250, 305, 340, 365, col='blue')
  text(140, 400, 'Churned', cex=1.2, srt=90)
  text(140, 335, 'Retained', cex=1.2, srt=90)

  # add in the cm results
  res <- as.numeric(cm$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')
```

```
  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 65, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 65, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
  text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(50, 65, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
  text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
  text(70, 65, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
  text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
  text(90, 65, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

  # add in the accuracy information
  text(30, 30, names(cm$overall[1]), cex=1.3, font=2)
  text(30, 10, round(as.numeric(cm$overall[1]), 3), cex=1.0)
  text(70, 30, names(cm$overall[2]), cex=1.3, font=2)
  text(70, 10, round(as.numeric(cm$overall[2]), 3), cex=1.0)
}

draw_confusion_matrix(cm)
```



Interpretation

Top 4 Statistically Significant Variables:

1. esent (Emails Sent):
   - A one-unit increase in emails sent increases the log odds of retention by 0.23.

- Higher email engagement correlates with increased customer retention.

2. eclickrate (Email Click Rate):
   - A one-unit increase in click rate increases the log odds of retention by 0.22.
   - Higher click rates indicate more engaged customers likely to be retained.
3. ordfreq (Order Frequency):
   - A one-unit increase in order frequency increases the log odds of retention by 0.91.
   - Customers who make frequent orders are more likely to be retained.
4. custAge (Customer Tenure):
   - A one-unit increase in customer tenure decreases the log odds of retention by 0.002.
   - Longer customer tenure correlates with slightly lower retention odds.

Additional Insights

- Email Engagement Impact:
  - High significance of esent and eclickrate highlights the importance of personalized and engaging email campaigns.
- Order Behavior Significance:
  - Ordfreq's high significance emphasizes the role of consistent ordering behavior in predicting customer retention.
- Customer Tenure Consideration:
  - Longer customer tenure, although statistically significant, has a minor impact on retention odds.

Recommendations

- Enhance Email Campaigns:
  - Invest in targeted and engaging email campaigns to boost customer retention.
- Encourage Order Consistency:
  - Implement strategies to encourage frequent and consistent customer orders.
- Optimize Customer Tenure Impact:
  - While longer customer tenure slightly decreases retention odds, focus on strategies to enhance overall customer satisfaction and loyalty.