Tuango: Targeting Analysis for Mobile App Push Messaging

Shiv Viswanathan Section 81

Preliminaries

IMPORTANT: Before starting this assignment you must go through logistic regression handout we covered in class to correctly install the `kelloggmktg482` package.

```
#install.packages("remotes")
library(remotes)
#install_version("vip", version="0.3.2", upgrade="never")
#devtools::install_github("blakemcshane/kelloggmktg482", upgrade = "never", force = TRUE)
```

Read in the data:

```
# use load("filename.Rdata") for .Rdata files

load("tuango.Rdata")
```

```
#Mutute the categoriacal variable to type factor
library(dplyr)

tuango <- tuango %>%
  mutate(sex = factor(sex),
         messages = factor(messages))
```

Assignment questions and answers

Part 1 (18 points)

1. Estimate a logistic regression model using "buyer" as the dependent variable using all relevant predictor variables, namely age, sex, messages, recency, frequency, monetary, and music.

```
## Logistic Regression
lr <- glm (buyer~age+sex+messages+recency+frequency+monetary+music, family = binomial, data =tuango)
```

#2. Use summary(. . .) to examine the coefficient estimates, varimpplot(. . .) to assess variable importance, and pardepplot(. . .) the effect of each predictor. What variables seem to be practically important? Describe their effects.

```
summary(lr)
```

```
##
## Call:
```

```
## glm(formula = buyer ~ age + sex + messages + recency + frequency +
##     monetary + music, family = binomial, data = tuango)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7161  -0.5521  -0.4320  -0.3110   2.7935
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.8980350  0.0946143 -30.630  < 2e-16 ***
## age         -0.0126514  0.0012583 -10.054  < 2e-16 ***
## sexM        -0.4607670  0.0468679  -9.831  < 2e-16 ***
## messagesOn   0.9895453  0.0584431  16.932  < 2e-16 ***
## recency     -0.0017088  0.0003079  -5.550 2.86e-08 ***
## frequency    0.1090890  0.0092633  11.777  < 2e-16 ***
## monetary     0.0028917  0.0001907  15.161  < 2e-16 ***
## music        0.5580193  0.0527455  10.579  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15304  on 20907  degrees of freedom
## Residual deviance: 14264  on 20900  degrees of freedom
## AIC: 14280
##
## Number of Fisher Scoring iterations: 5
```

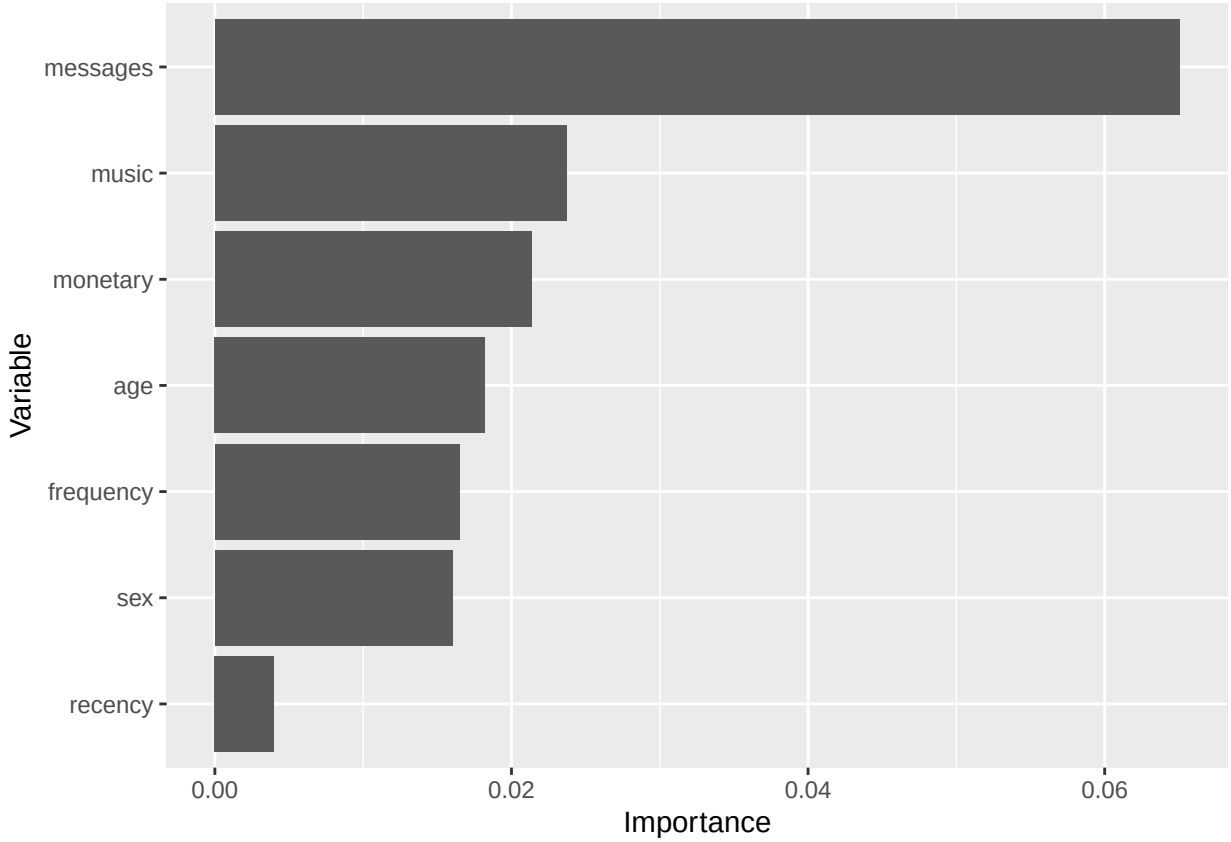
```
confint(lr)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %       97.5 %
## (Intercept) -3.084589297 -2.713669703
## age         -0.015126635 -0.010193676
## sexM        -0.553035712 -0.369292231
## messagesOn   0.876202048  1.105359361
## recency     -0.002321116 -0.001113792
## frequency    0.090857418  0.127173966
## monetary     0.002518810  0.003266617
## music        0.455408870  0.662207289
```

###varimpplot(...) to assess variable importance

```
varimpplot(lr, target="buyer")
```

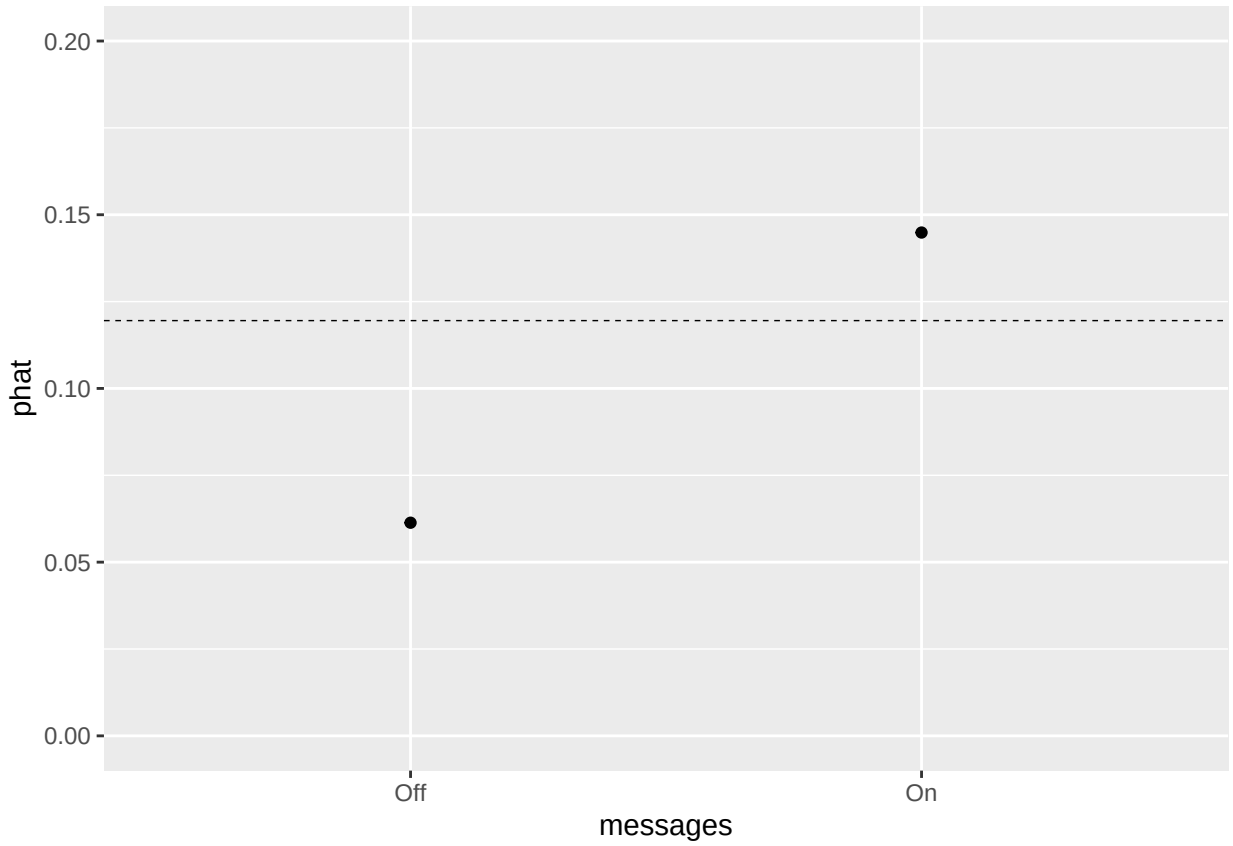


###pardepplot(...) the effect of each predictor

```
perc.buyer.overall <- mean(tuango$buyer==1)
```

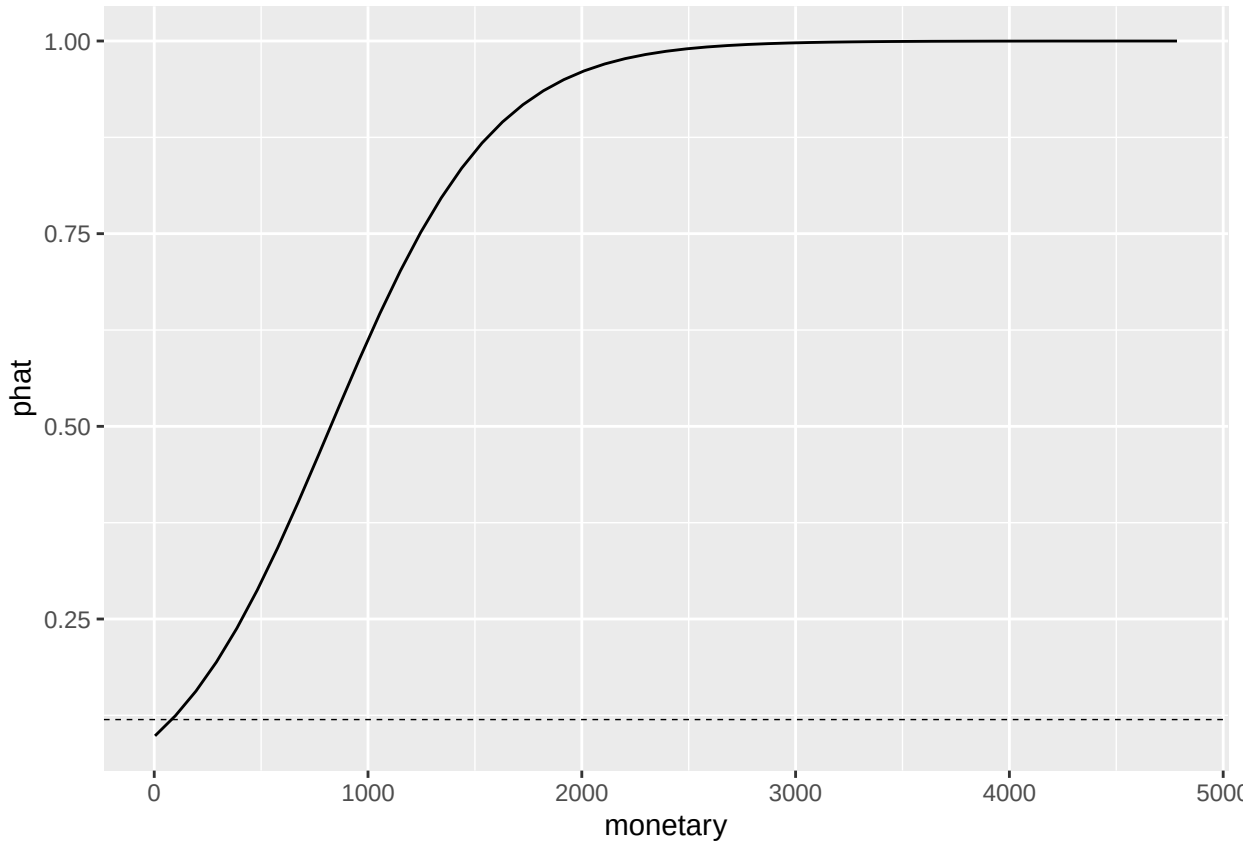
pardepplot for messages

```
pardepplot(lr, pred.var = "messages", data=tuango, hline = perc.buyer.overall, ylim = c(0,0.2))
```



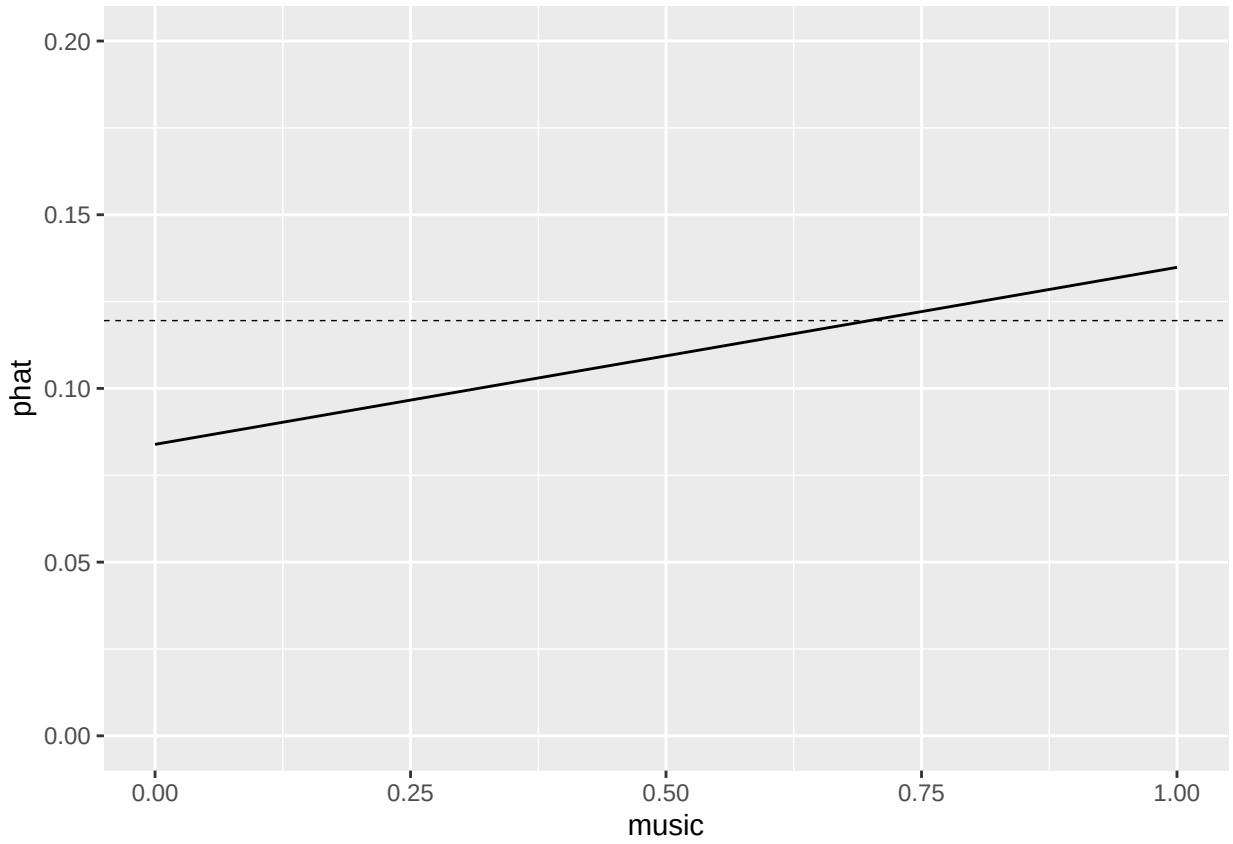
pardepplot for monetary

```
pardepplot(lr, pred.var = "monetary", data=tuango, hline = perc.buyer.overall)
```



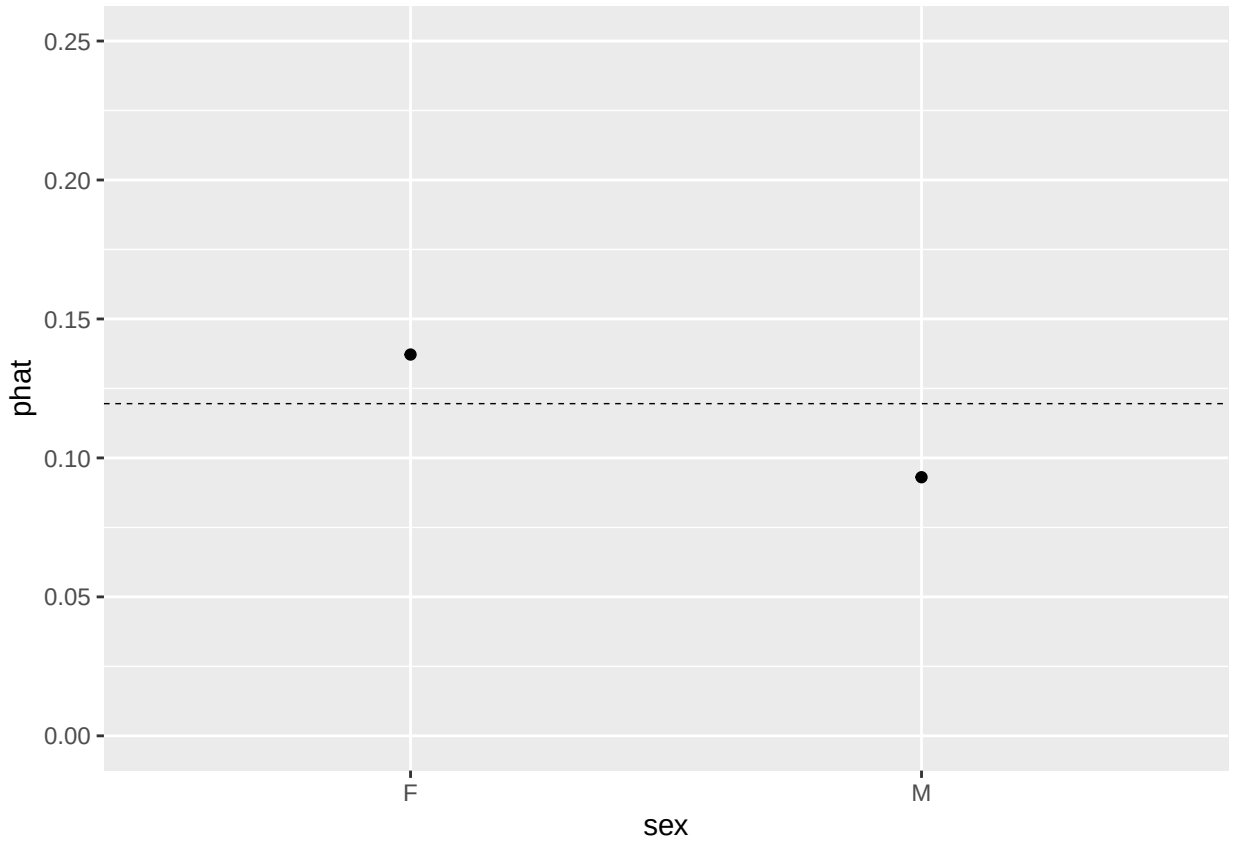
pardepplot for music

```
pardepplot(lr, pred.var = "music", data=tuango, hline = perc.buyer.overall, ylim = c(0,0.2))
```



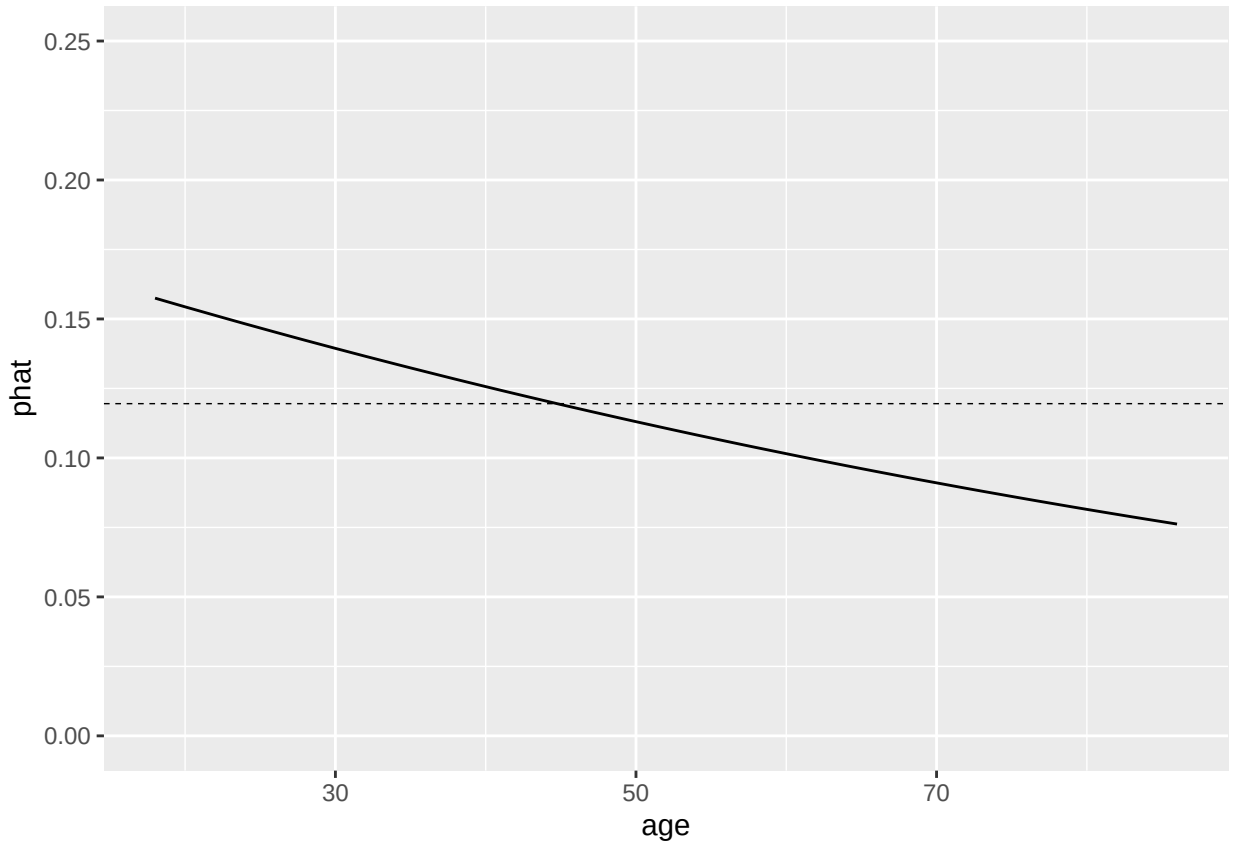
pardepplot for sex

```
pardepplot(lr, pred.var = "sex", data=tuango, hline = perc.buyer.overall, ylim = c(0,0.25))
```



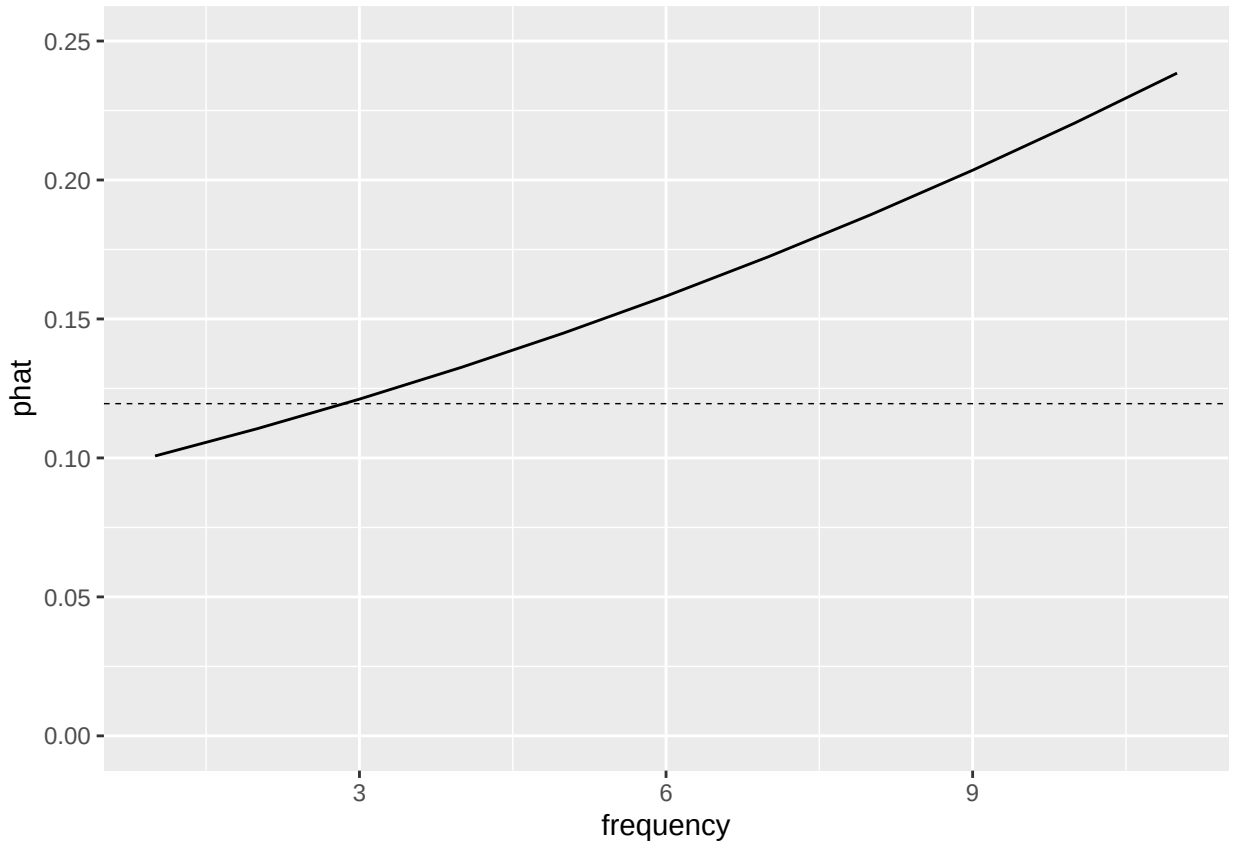
pardepplot for age

```
pardepplot(lr, pred.var = "age", data=tuango, hline = perc.buyer.overall, ylim = c(0,0.25))
```



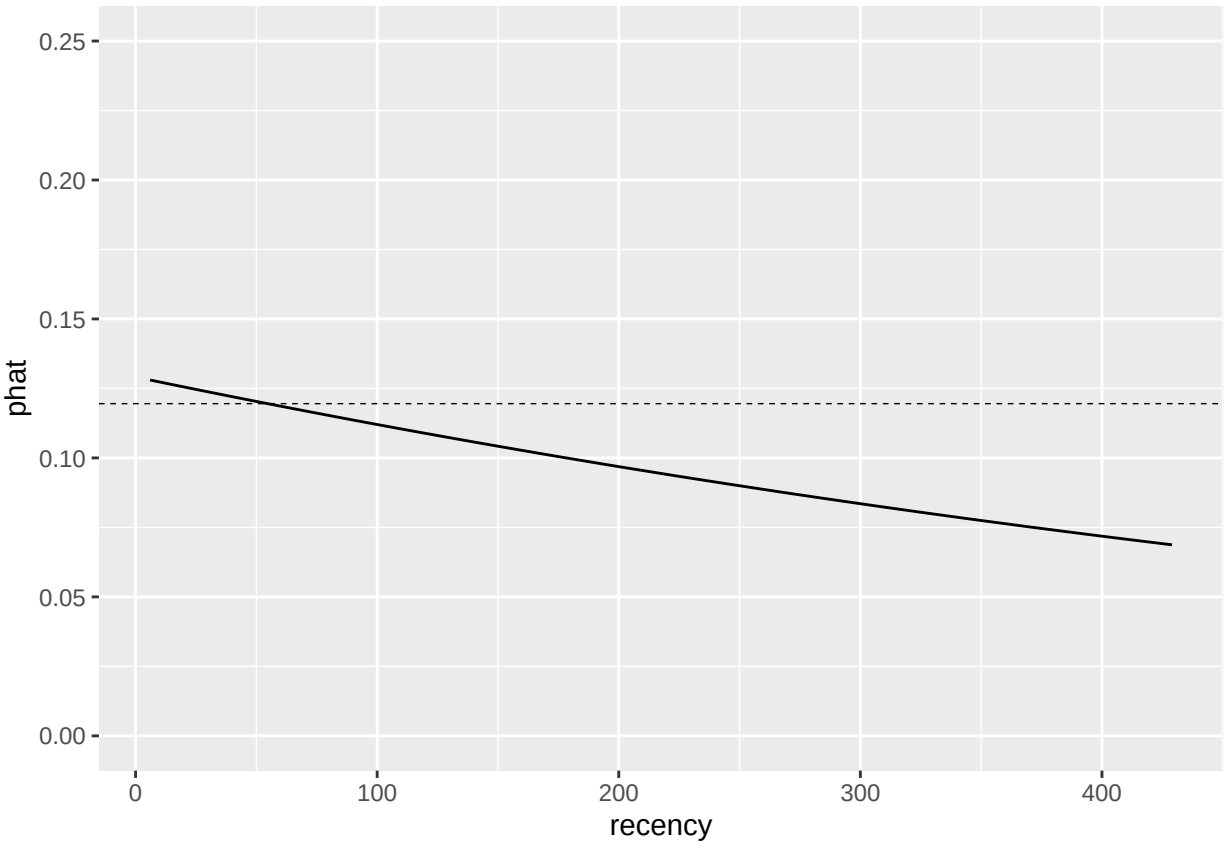
pardepplot for frequency

```
pardepplot(lr, pred.var = "frequency", data=tuango, hline = perc.buyer.overall, ylim = c(0,0.25))
```



pardepplot for recency

```
pardepplot(lr, pred.var = "recency", data=tuango, hline = perc.buyer.overall, ylim = c(0,0.25))
```



What variables seem to be practically important? Describe their effects.

###age : Age <45 seem to have higher probability above the buyer mean to buy karaoke packages

###sexM : Females have higher probability to be buying karaoke packages

messagesON : Message being ON on mobile have higher probabiity of purchasing karaoke package

recency : negatively trending, ie. people who purchased a deal recently have a higher probability of buying the karaoke package than someone who did not buy a package for a few months.

frequency : more than 3 purchases

monetary : 0-2000$ increasing probability trend – above 2000% definitely purchase karaoke package

music : people who purchased music before, tend to buy karaoke deals - linear

3. Add the predicted values from the logistic regression to the "tuango" dataframe. Compare the average of the predicted values to the overall response rate (i.e., percentage of buyers). What do you notice? Why is that?

```
##Add the predicted values from the logistic regression to the "tuango" dataframe.

tuango <- tuango %>%
  mutate(pred_prob = predict(lr, type="response"))
```

```
tuango %>%
  select(userid, pred_prob) %>%
  head()
```

```
## # A tibble: 6 x 2
##     userid pred_prob
##      <int>     <dbl>
## 1 15889344    0.207
## 2 60246497    0.0546
## 3 22965759    0.0513
## 4 40811142    0.126
## 5 76283952    0.115
## 6 37412566    0.0550
```

```
mean.pred.prob = mean(tuango$pred_prob)

##average of the predicted values and percentage of buyers
tuango %>% summarise(mean.pred.prob, perc.buyer.overall)
```

```
## # A tibble: 1 x 2
##   mean.pred.prob perc.buyer.overall
##            <dbl>              <dbl>
## 1          0.120              0.120
```

The average of the predicted value is the same as the overall percentage response rate i.e 0.11952

#4. Assign each customer to a decile based on his or her predicted probability of purchase. Assign those with the highest predicted probability to decile 1 and those with the lowest predicted probability to decile 10. Generate a table with ten rows and four columns with columns being the decile number; the number of customers in that decile, the number of buyers in that decile, and the response rate in that decile.

```
tuango <- tuango %>%
  mutate(decile = ntile(pred_prob, 10))

tuango %>%
  select(userid, pred_prob, decile) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##      userid pred_prob decile
##       <int>     <dbl>  <int>
##  1 15889344    0.207       9
##  2 60246497    0.0546      2
##  3 22965759    0.0513      2
##  4 40811142    0.126       7
##  5 76283952    0.115       6
##  6 37412566    0.0550      2
##  7 45474095    0.115       6
##  8 15371036    0.119       6
##  9 79932394    0.0320      1
## 10 84213856    0.0356      1
```

```
# Create a summary table with decile, number of customers, number of buyers, and response rate
summary_table <- tuango %>%
  group_by(decile) %>%
  summarise(
    Count = n(),
    Buyers = sum(buyer),
    ResponseRate = mean(buyer)
  ) %>%
  arrange(desc(decile))  # Sort the table by decile in descending order

print(summary_table)
```

```
## # A tibble: 10 x 4
##    decile Count Buyers ResponseRate
##     <int> <int>  <int>        <dbl>
##  1     10  2090    588       0.281
##  2      9  2090    360       0.172
##  3      8  2091    356       0.170
##  4      7  2091    266       0.127
##  5      6  2091    239       0.114
##  6      5  2091    207       0.0990
##  7      4  2091    169       0.0808
##  8      3  2091    155       0.0741
##  9      2  2091    107       0.0512
## 10      1  2091     52       0.0249
```

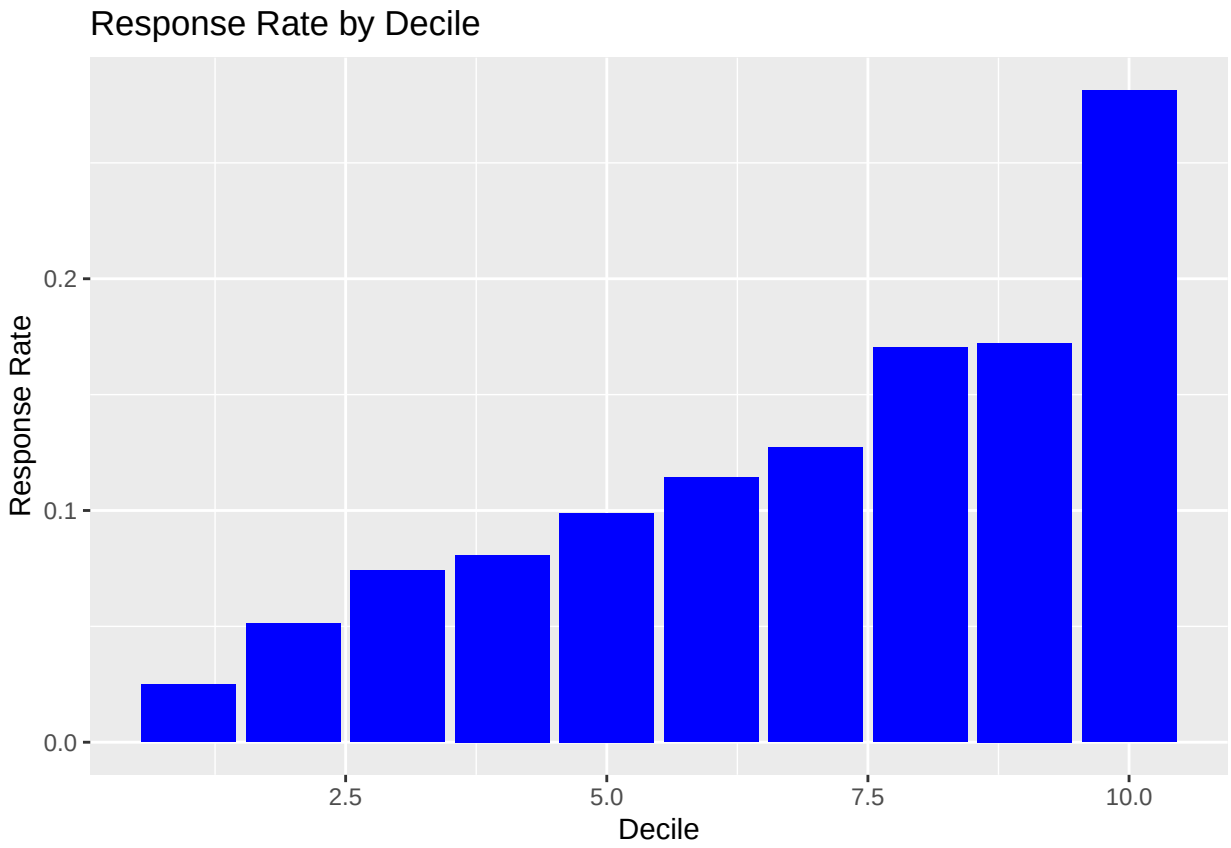
5. Use the table created in the prior question to make a barchart that plots the response rate in each decile defined. Comment on your findings.

```
library(ggplot2)
library(scales)
```

```
##
```

```
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## The following objects are masked from 'package:psych':
##
##     alpha, rescale
```

```r
# Create a bar chart
bar_chart <- ggplot(summary_table, aes(x = decile, y = ResponseRate)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Response Rate by Decile",
    x = "Decile",
    y = "Response Rate"
  )

# Display the bar chart
print(bar_chart)
```



#6.Estimate a linear regression model using "ordersize" as the dependent variable using all relevant predictor variables, namely age, sex, messages, recency, frequency, monetary, and music. Estimate this regression using only those customers who purchased the Karaoke deal. Hint: Use filter(buyer==1) in the dplyr package to create a dataframe limited to only those customers who purchased the Karaoke deal and use this dataframe

as an input to the linear regression. Consider: why are we limiting this analysis only to these customers?

```
## Linear Regression

load("tuango.Rdata")

#Mutute the categoriacal variable to type factor

tuango <- tuango %>%
  mutate(sex = factor(sex),
         messages = factor(messages))

customer_purchase_karoake <- tuango %>%
  filter(buyer==1)
```

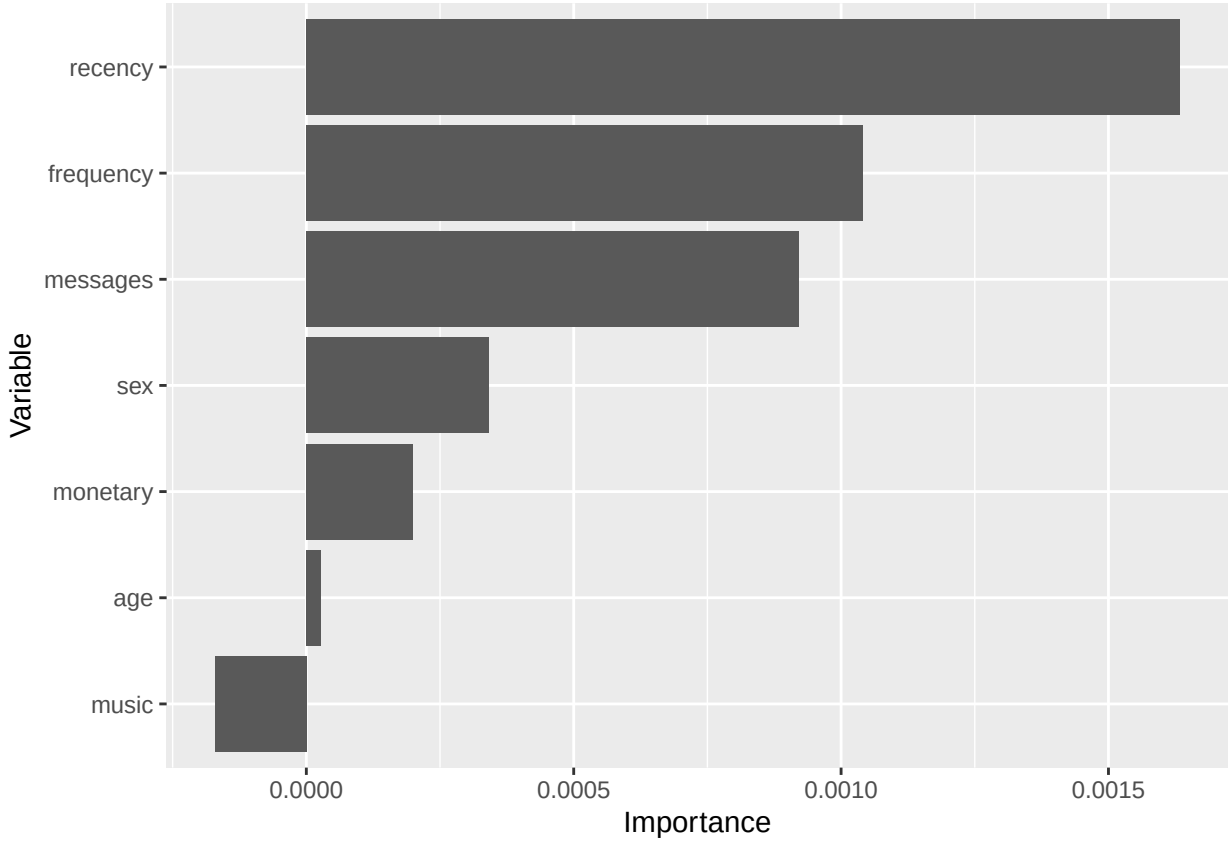
```
lr_buyer <- lm (ordersize~age+sex+messages+recency+frequency+monetary+music, data = customer_purchase_ka
```

#7. Use summary(...) to examine the coefficient estimates, varimpplot(...) to assess variable importance, and pardepplot(...) the effect of each predictor. What variables seem to be practically important? Describe their effects. What does this suggest about our ability to predict the order size of those customers who responded to the deal?

```
summary(lr_buyer)
```

```
##
## Call:
## lm(formula = ordersize ~ age + sex + messages + recency + frequency +
##     monetary + music, data = customer_purchase_karoake)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2745 -1.0672 -0.0503  1.0285  7.8740
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.971e+00  1.605e-01  24.740   <2e-16 ***
## age         -8.249e-05  2.527e-03  -0.033    0.974
## sexM        -2.137e-02  7.479e-02  -0.286    0.775
## messagesOn   7.217e-02  9.637e-02   0.749    0.454
## recency      4.642e-04  4.748e-04   0.978    0.328
## frequency    2.227e-02  1.412e-02   1.576    0.115
## monetary    -8.590e-05  1.575e-04  -0.545    0.586
## music       -1.185e-01  8.525e-02  -1.390    0.165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.722 on 2491 degrees of freedom
## Multiple R-squared:  0.002567,   Adjusted R-squared:  -0.0002357
## F-statistic: 0.9159 on 7 and 2491 DF,  p-value: 0.4928
```

```
varimpplot(lr_buyer, target="ordersize")
```



What variables seem to be practically important? Describe their effects.

frequency - frequency seems to be the biggest variable that has an effect on the order size.

recency : recency comes second – this is different from the overall data set – this suggests that those who purchased recently have a smaller order size.

```
confint(lr_buyer)
```

```
##                     2.5 %       97.5 %
## (Intercept)  3.6560413458 4.2854864861
## age         -0.0050373293 0.0048723531
## sexM        -0.1680277663 0.1252923806
## messagesOn  -0.1167980783 0.2611480476
## recency     -0.0004668828 0.0013953381
## frequency   -0.0054300274 0.0499639190
## monetary    -0.0003947232 0.0002229243
## music       -0.2856314660 0.0487017003
```

```
##Add the predicted values from the logistic regression to the "tuango" dataframe.

customer_purchase_karoake <- customer_purchase_karoake %>%
  mutate(pred_prob = predict(lr_buyer, type="response"))
```

```
customer_purchase_karoake %>%
  select(userid, pred_prob) %>%
```

```
  head()
```

```
## # A tibble: 6 x 2
##     userid pred_prob
##      <int>     <dbl>
## 1 17472226      4.01
## 2 36563987      4.12
## 3 42231219      4.02
## 4 23389814      4.17
## 5 25747882      3.96
## 6 59291906      3.98
```

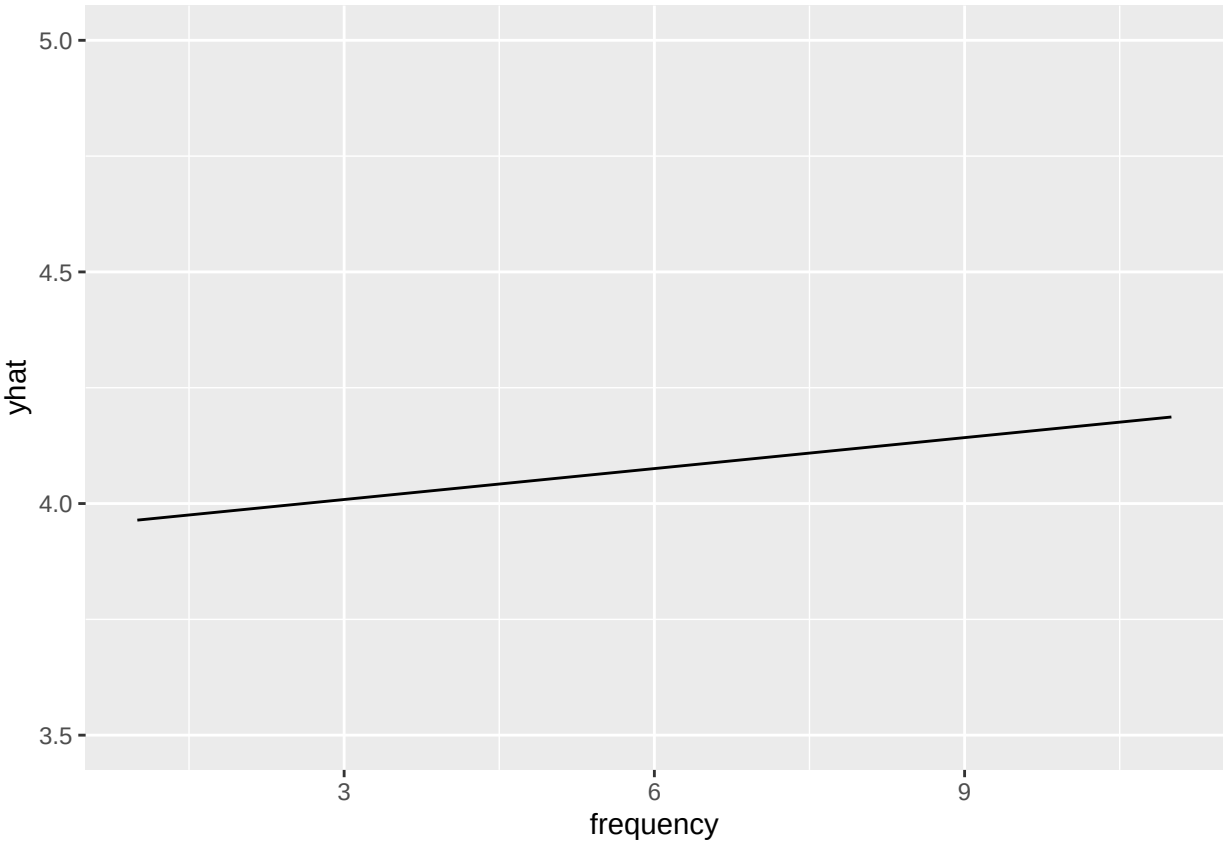
```
##average of the predicted values and percentage of buyers
customer_purchase_karoake %>% summarise(mean(customer_purchase_karoake$pred_prob), mean(customer_purcha
```

```
## # A tibble: 1 x 2
##   `mean(customer_purchase_karoake$pred_prob)` mean(customer_purchase_karoake$o~1
##                                         <dbl>                              <dbl>
## 1                                        4.01                               4.01
## # i abbreviated name: 1: `mean(customer_purchase_karoake$ordersize)`
```

###The (mean(customer_purchase_karoake$pred_prob), mean(customer_purchase_karoake$ordersize)) are the same! 4.011

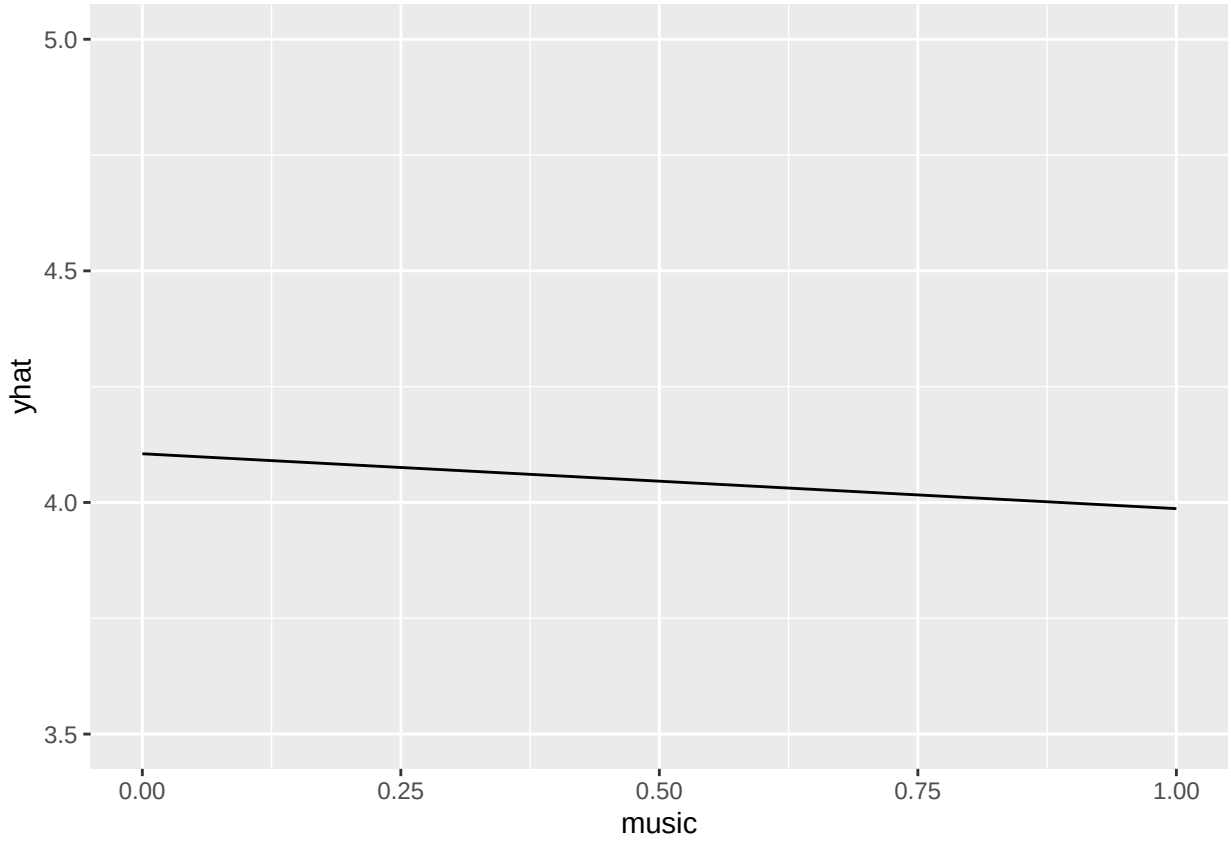
```
pardepplot(lr_buyer, pred.var = "frequency", data=customer_purchase_karoake, hline = perc.buyer.overall
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```



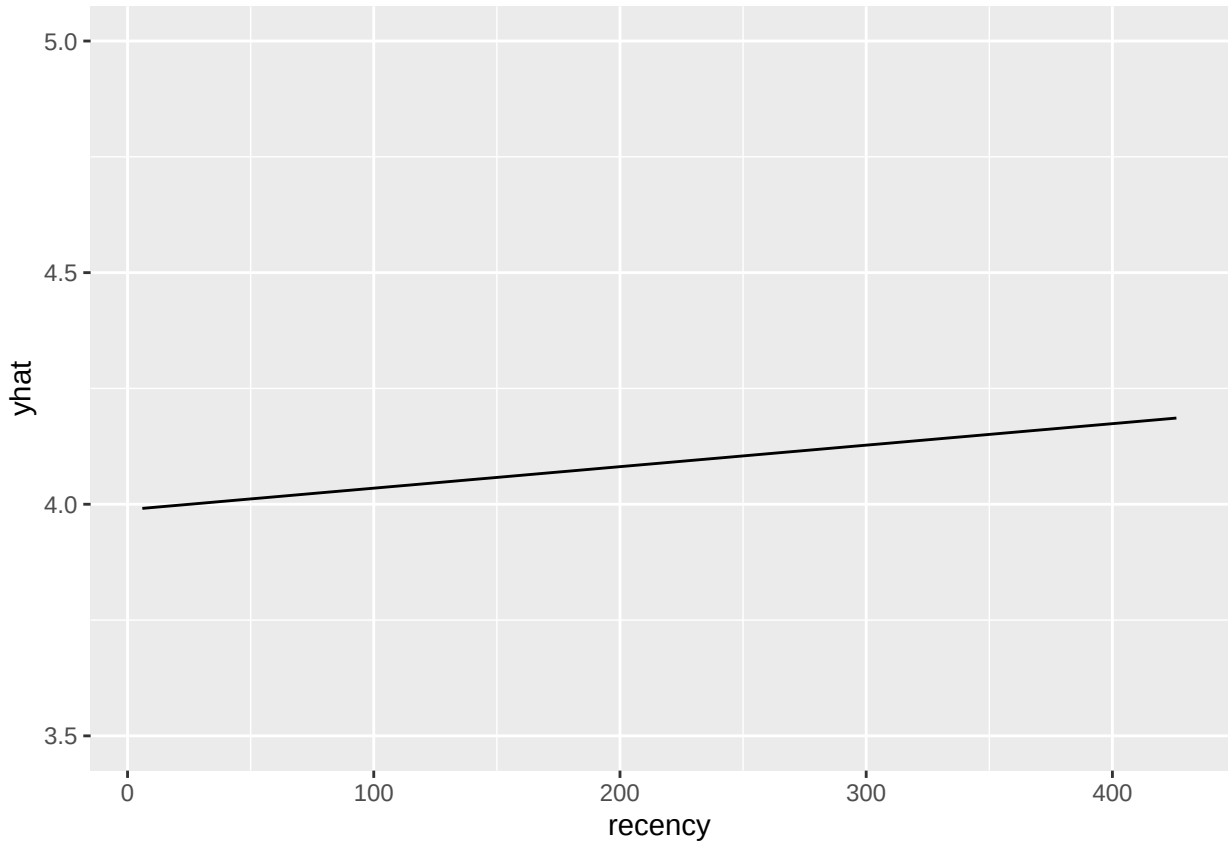
```
pardepplot(lr_buyer, pred.var = "music", data=customer_purchase_karoake, hline = perc.buyer.overall, yli
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```



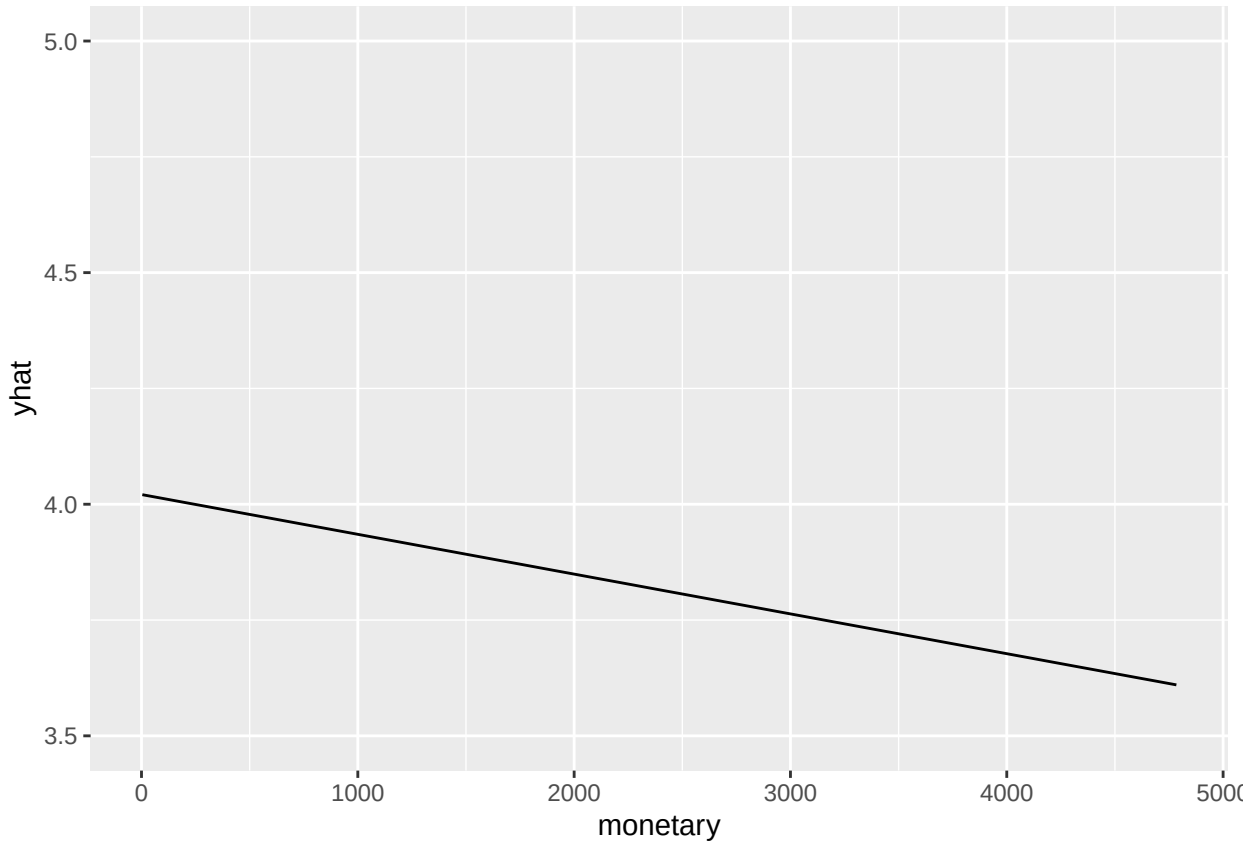

```
pardepplot(lr_buyer, pred.var = "recency", data=customer_purchase_karoake, hline = perc.buyer.overall,yl
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```



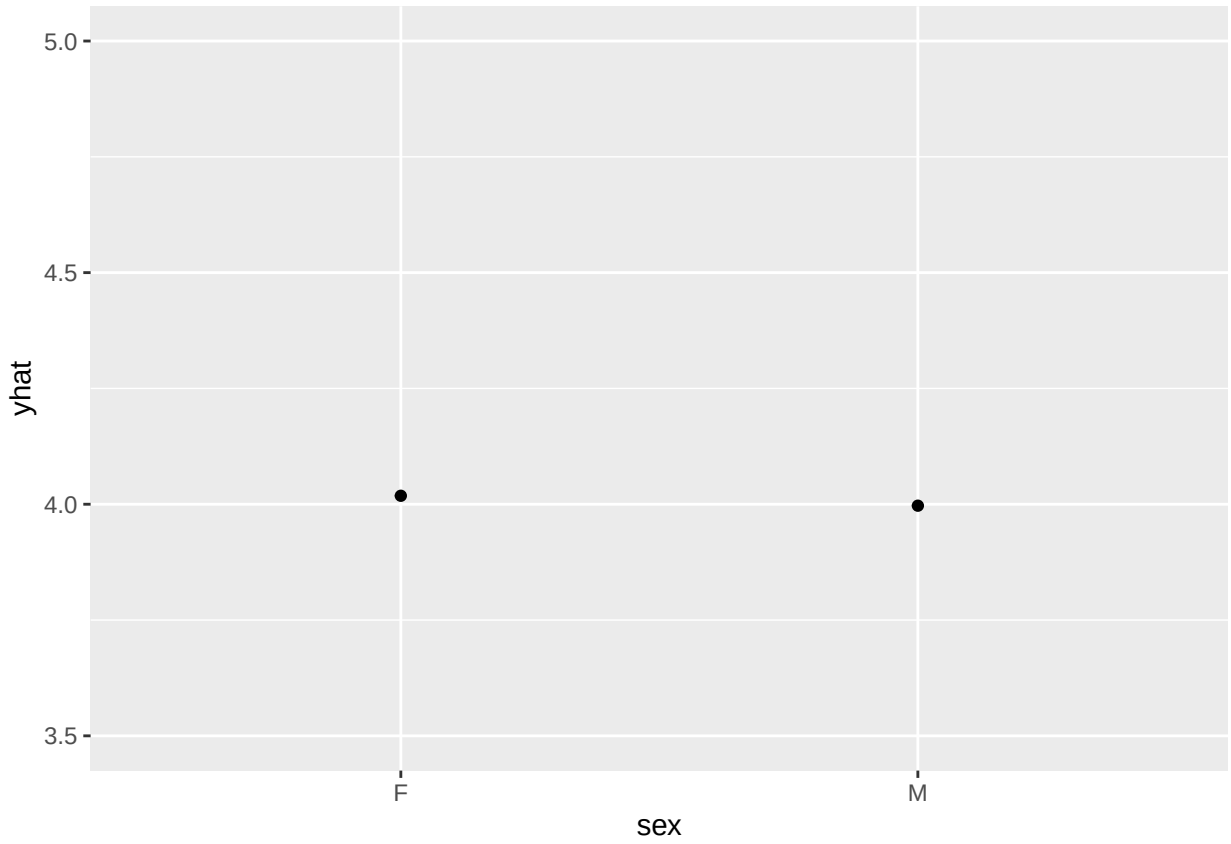

```
pardepplot(lr_buyer, pred.var = "monetary", data=customer_purchase_karoake, hline = perc.buyer.overall,y
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```



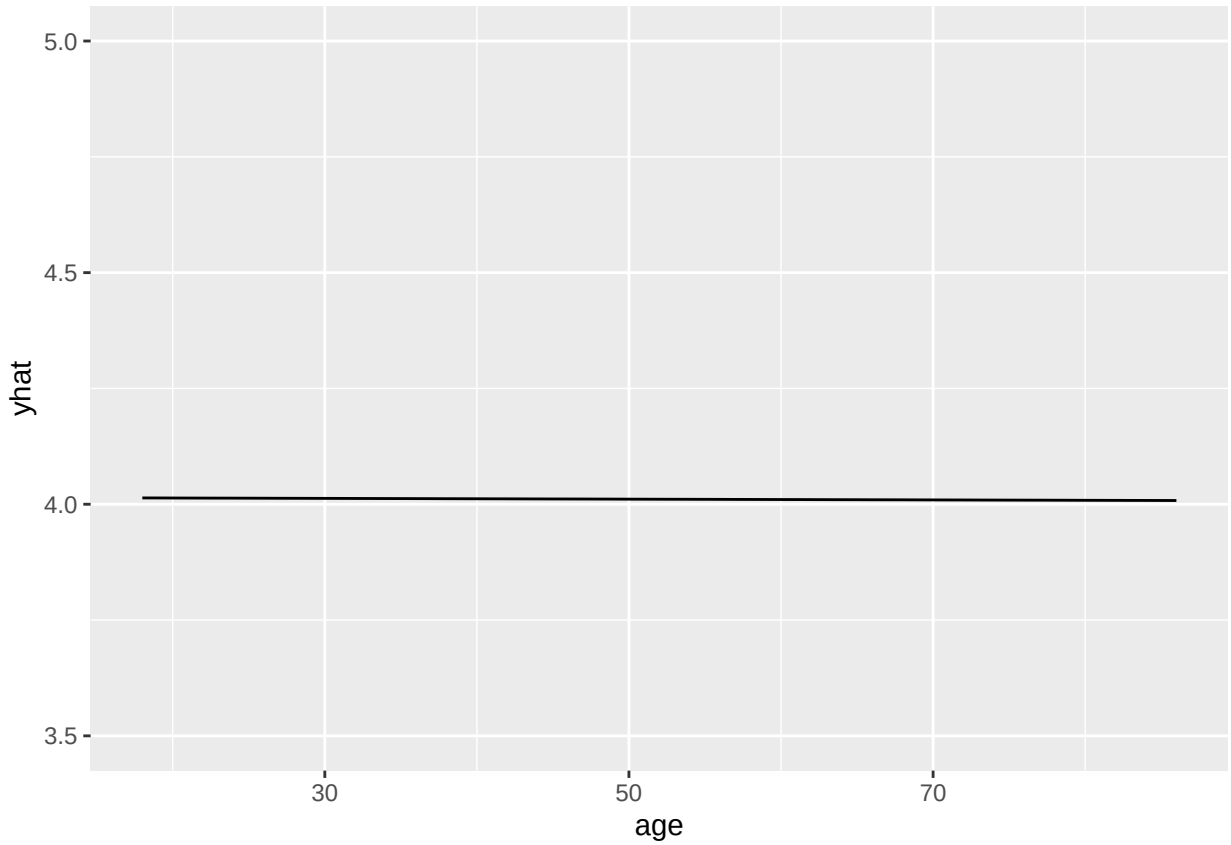

```
pardepplot(lr_buyer, pred.var = "sex", data=customer_purchase_karoake, hline = perc.buyer.overall,ylim =
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```



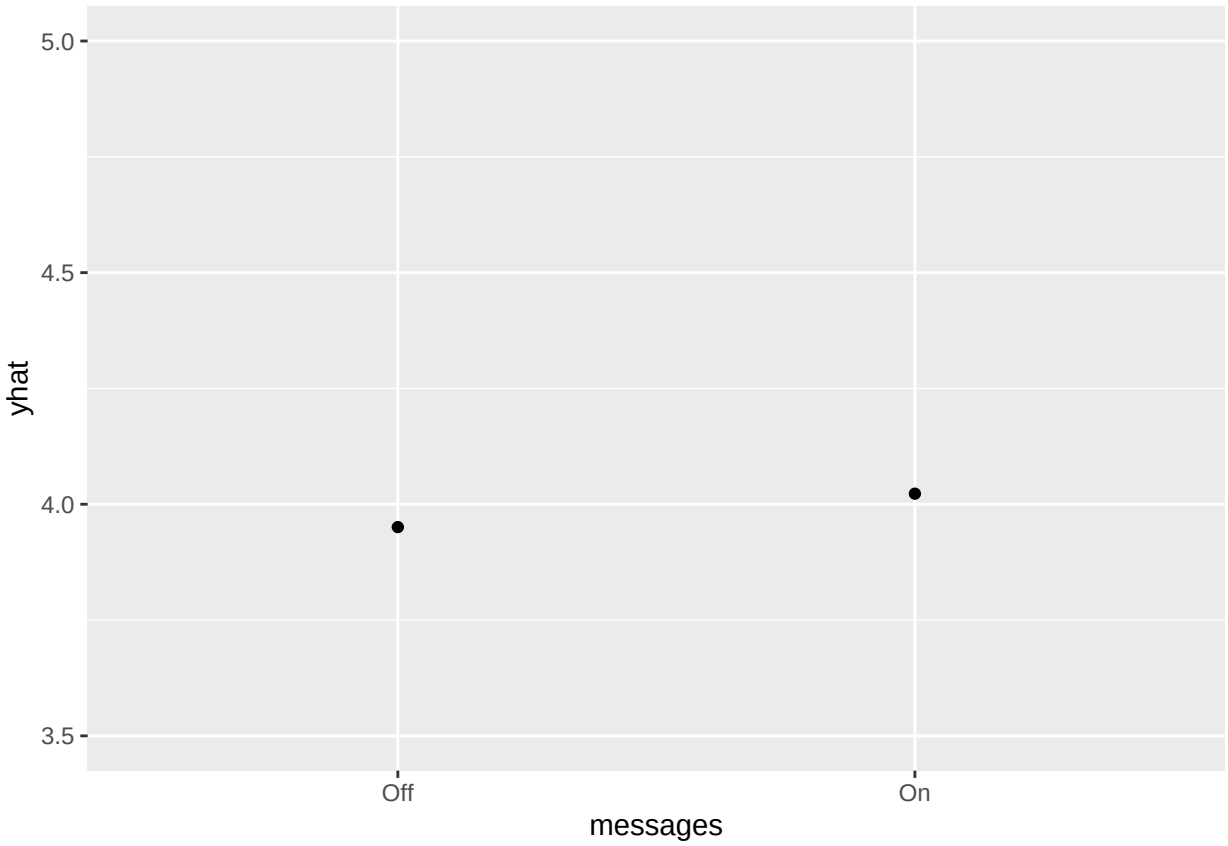

```
pardepplot(lr_buyer, pred.var = "age", data=customer_purchase_karoake, hline = perc.buyer.overall,ylim =
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```




```
pardepplot(lr_buyer, pred.var = "messages", data=customer_purchase_karoake, hline = perc.buyer.overall,y
```

```
## Warning: Removed 1 rows containing missing values (`geom_hline()`).
```



Part II: Profitability Analysis (8 points)

##The following questions will ask you to use your data to forecast the profit and the return on marketing expenditures of offering the deal to the remaining 397,252 potential customers (i.e. 418,160 – 20,908).

To calculate profit assume the following:

##Price per 30-minute session = 49 RMB ##Marginal cost to offer a deal is = 11 RMB ##Fee on each deal sold = 50% (of sales revenues)

1. What is the breakeven response rate?

```
price.per.30.min = 49
marginal.cost.deal = 11
perc.fees = 0.5

profit = price.per.30.min * perc.fees

breakeven.response.rate =  marginal.cost.deal/ profit

print("The Breakeven response rate is")
```

```
## [1] "The Breakeven response rate is"
```

```
print(breakeven.response.rate)
```

```
## [1] 0.4489796
```

2. What is the projected profit in RMB if you offer the deal to all 397,252 remaining customers?

```
remaining.customers = 397252



projected.profit = perc.buyer.overall * remaining.customers * profit


print("Projected profit in RMB based on percentage of buyer response is")
```

```
## [1] "Projected profit in RMB based on percentage of buyer response is"
```

```
print(projected.profit)
```

```
## [1] 1163284
```

3. What is the projected profit if you offer the deal to only to those of the 397,252 remaining customers that targeting model deems profitable?

```
## Its the same as above, as the above, since the percentage of buyer overall is the same as the model p


remaining.customers = 397252

projected.profit = mean.pred.prob * remaining.customers * profit

print("Projected profit in RMB based on model predicts is")
```

```
## [1] "Projected profit in RMB based on model predicts is"
```

```
print(projected.profit)
```

```
## [1] 1163285
```

Part III: Build Targeting Models Redux (4 points)

1. Estimate a neural network model using "buyer" as the dependent variable using the same predictor variables as the logistic regression estimated in Part I. Set the size tuning parameter to 5 and the decay tuning parameter to 0.1.

```
#install.packages("nnet")
library(nnet)

nn <- nnet(buyer~age+sex+messages+recency+frequency+monetary+music, data=tuango, size=5, decay=0.1, maxi
```

```
## # weights:  46
## initial  value 7730.054332
## iter  10 value 2490.056030
## iter  20 value 2276.892736
## iter  30 value 2179.493561
## iter  40 value 2168.334693
## iter  50 value 2156.897324
## iter  60 value 2132.888863
```

```
## iter  70 value 2107.147723
## iter  80 value 2091.612503
## iter  90 value 2084.237225
## iter 100 value 2081.630912
## iter 110 value 2076.461448
## iter 120 value 2067.209040
## iter 130 value 2059.518153
## iter 140 value 2052.920453
## iter 150 value 2044.371271
## iter 160 value 2032.120680
## iter 170 value 2026.200082
## iter 180 value 2024.862526
## iter 190 value 2024.055878
## iter 200 value 2022.513242
## iter 210 value 2019.945700
## iter 220 value 2018.697689
## iter 230 value 2018.623099
## iter 240 value 2018.195832
## iter 250 value 2017.752865
## iter 260 value 2017.631221
## iter 270 value 2017.404788
## iter 280 value 2017.373572
## final  value 2017.372449
## converged
```

2. Add the predicted values from the neural network to the "tuango" dataframe and repeat the profitability analysis in Part II but using these predicted values rather than those from the logistic regression. Comment on why your profits are similar to or different from those found in Part II.

```
tuango <- tuango %>%
mutate(pred_nn = predict(nn, type="raw")[,1])


tuango %>%
  select(userid, pred_nn) %>%
  head()
```

```
## # A tibble: 6 x 2
##     userid pred_nn
##      <int>   <dbl>
## 1 15889344  0.102
## 2 60246497  0.0190
## 3 22965759  0.0433
## 4 40811142  0.138
## 5 76283952  0.178
## 6 37412566  0.0363
```

```
mean.pred.nn = mean(tuango$pred_nn)

##average of the predicted values and percentage of buyers
tuango %>% summarise(mean.pred.nn, perc.buyer.overall)
```

```
## # A tibble: 1 x 2
```

```
##   mean.pred.nn perc.buyer.overall
##          <dbl>              <dbl>
## 1        0.121              0.120
```

the predicted probability is slightly higher here than the mean buyer percentage.

```
remaining.customers = 397252

projected.profit = mean.pred.nn * remaining.customers * profit

print("Projected profit in RMB based on model predicts is")
```

```
## [1] "Projected profit in RMB based on model predicts is"
```

```
print(projected.profit)
```

```
## [1] 1175034
```

the nn model based targeting provides slightly more profit – in this case a 1.4% increase in profit

Disclaimer – ChatGPT is used for debugging purposes only. All the code here is done by me and I take full responsibility.