Group Assignment 2 - Creative Gaming

Section 51
Gaurav Agrawal, Ajitesh Abhishek, Tarun Joshi

Read in the data:

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

Part 1 - Question 1

```
cg_organic %>%
  summarise (organic_probability = mean(converted))
```

```
  organic_probability
1          0.05753333
```

Part 1 - Question 2

```
skim(cg_organic)
```

```
Skim summary statistics
 n obs: 30000
 n variables: 20

-- Variable type:factor -------------------------------------------------------------------
               variable missing complete     n n_unique
      AcquiredIonWeapon       0    30000 30000        2
      AcquiredSpaceship       0    30000 30000        2
 PurchasedCoinPackLarge       0    30000 30000        2
 PurchasedCoinPackSmall       0    30000 30000        2
           UserHasOldOS       0    30000 30000        2
          UserNoConsole       0    30000 30000        2
                 top_counts ordered
   0: 29439, 1: 561, NA: 0   FALSE
  0: 21695, 1: 8305, NA: 0   FALSE
  0: 22061, 1: 7939, NA: 0   FALSE
 0: 19857, 1: 10143, NA: 0   FALSE
  0: 27411, 1: 2589, NA: 0   FALSE
  0: 24546, 1: 5454, NA: 0   FALSE

-- Variable type:integer ------------------------------------------------------------------
  variable missing complete     n  mean   sd p0 p25 p50 p75 p100     hist
 converted       0    30000 30000 0.058 0.23  0   0   0   0    1 <U+2587><U+2581><U+2581><U+2581><U+258
```

```
-- Variable type:numeric -------------------------------------------------------------
                variable missing complete     n    mean     sd  p0  p25
               DaysUser       0    30000 30000 2626.37 661.43 244 2162
              GameLevel       0    30000 30000    6.25   2.77   1    4
           NumAdsClicked       0    30000 30000    9.49   7.4    0    4
 NumFriendRequestIgnored       0    30000 30000   29.59  33.99   0    0
             NumFriends       0    30000 30000    0.44   1.52   0    0
     NumFriendsofFriends       0    30000 30000   47.73  94.33   0    0
            NumGameDays       0    30000 30000   12.24   7.1    1    6
       NumGameDaysOnline       0    30000 30000    1.26   3.19   0    0
   NumInGameMessagesSent       0    30000 30000   73.78 107.44   0    0
           TimesCaptain       0    30000 30000    1.58   8.77   0    0
            TimesKilled       0    30000 30000    0.29   3.42   0    0
      TimesLostSpaceship       0    30000 30000    4.44  11.55   0    0
          TimesNavigator       0    30000 30000    1.4    7.95   0    0
  p50  p75 p100     hist
 2557 3105 4139 <U+2581><U+2581><U+2582><U+2585><U+2587><U+2585><U+2583><U+2582>
    7    9   10 <U+2585><U+2581><U+2582><U+2583><U+2583><U+2583><U+2585><U+2587>
    8   12   38 <U+2587><U+2587><U+2585><U+2582><U+2582><U+2581><U+2581><U+2581>
   16   53  121 <U+2587><U+2582><U+2582><U+2582><U+2581><U+2581><U+2581><U+2581>
    0    0   12 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
    5   43  486 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
   13   18   28 <U+2587><U+2583><U+2585><U+2585><U+2586><U+2587><U+2582><U+2581>
    0    0   24 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
   26  112 1227 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
    0    0  429 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
    0    0  178 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
    0    4  298 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
    0    0  545 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
```

Part 2 - Question 1

```
cg_organic_train <- cg_organic[sample_train_org,]
cg_organic_test <- cg_organic[-sample_train_org,]

train_test_split = nrow(cg_organic_train)/nrow(cg_organic)

train_test_split
```

```
[1] 0.7
```

The training test split is 70:30

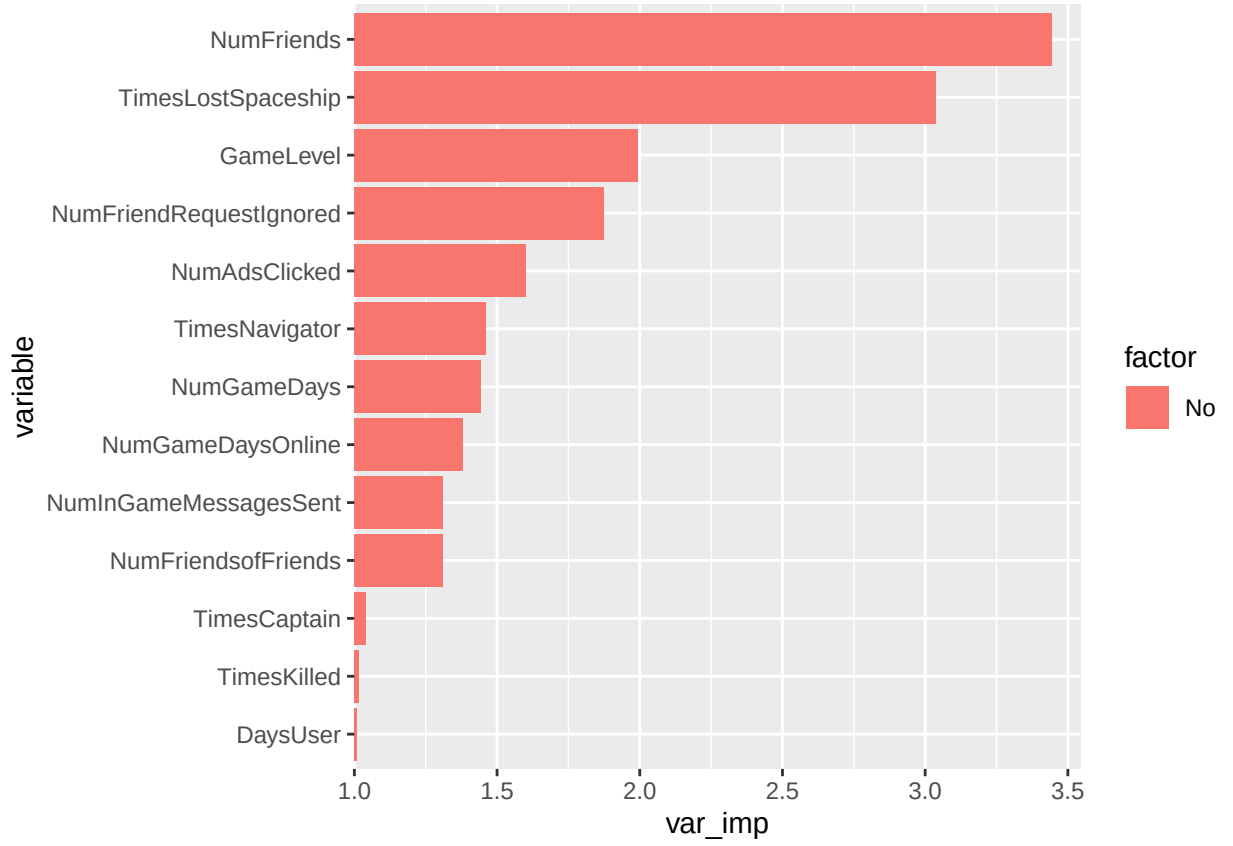
Part 2 - Question 2: Base Work

```
logit1 <- glm(converted ~ DaysUser + GameLevel + NumAdsClicked + NumFriendRequestIgnored + NumFriends +
, data=cg_organic_train)
summary(logit1)
```

```
Call:
glm(formula = converted ~ DaysUser + GameLevel + NumAdsClicked +
    NumFriendRequestIgnored + NumFriends + NumFriendsofFriends +
    NumGameDays + NumGameDaysOnline + NumInGameMessagesSent +
    TimesCaptain + TimesKilled + TimesLostSpaceship + TimesNavigator,
    family = binomial(logit), data = cg_organic_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4893  -0.3249  -0.2530  -0.1945   3.4788

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -4.563e+00  1.766e-01 -25.843  < 2e-16 ***
DaysUser                 6.301e-06  5.018e-05   0.126 0.900077
GameLevel                1.242e-01  1.406e-02   8.832  < 2e-16 ***
NumAdsClicked            3.179e-02  3.677e-03   8.645  < 2e-16 ***
NumFriendRequestIgnored -9.200e-03  1.255e-03  -7.332 2.27e-13 ***
NumFriends               4.121e-01  1.319e-02  31.249  < 2e-16 ***
NumFriendsofFriends      1.418e-03  2.912e-04   4.870 1.12e-06 ***
NumGameDays              2.579e-02  5.367e-03   4.806 1.54e-06 ***
NumGameDaysOnline        5.034e-02  8.335e-03   6.039 1.55e-09 ***
NumInGameMessagesSent    1.259e-03  3.706e-04   3.395 0.000685 ***
TimesCaptain             2.108e-03  2.728e-03   0.773 0.439656
TimesKilled              2.309e-03  8.432e-03   0.274 0.784237
TimesLostSpaceship      -4.817e-02  5.609e-03  -8.588  < 2e-16 ***
TimesNavigator          -2.446e-02  6.845e-03  -3.573 0.000352 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 9132.0  on 20999  degrees of freedom
Residual deviance: 7560.8  on 20986  degrees of freedom
AIC: 7588.8

Number of Fisher Scoring iterations: 7
```

Part 2 - Question 2a.

```
varimp.logistic(logit1) %>% plotimp.logistic()
```

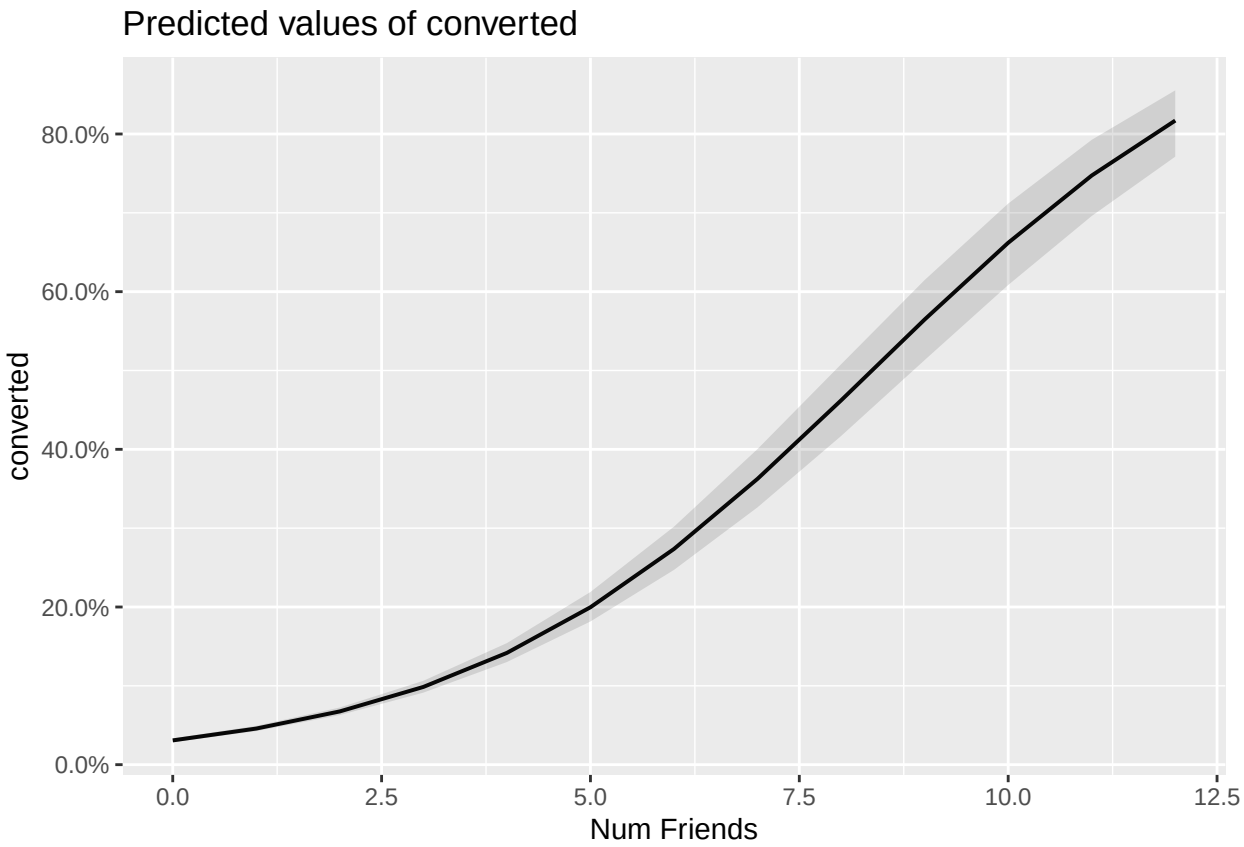



```
# A tibble: 13 x 9
   variable    var_imp p_value factor    OR OR_perc     sd OR_sd OR_sd_perc
   <chr>         <dbl>   <dbl> <chr>  <dbl> <chr>    <dbl> <dbl> <chr>
 1 NumFriends     3.44   0     No     1.51  51.0%     1.50 1.86  85.6%
 2 TimesLostS~    3.04   0     No     0.953 -4.7%    11.5  0.574 -42.6%
 3 GameLevel      1.99   0     No     1.13  13.2%     2.78 1.41  41.2%
 4 NumFriendR~    1.87   0     No     0.991 -0.9%    34.1  0.731 -26.9%
 5 NumAdsClic~    1.60   0     No     1.03  3.2%      7.39 1.26  26.5%
 6 TimesNavig~    1.46   0     No     0.976 -2.4%     7.72 0.828 -17.2%
 7 NumGameDays    1.44   0     No     1.03  2.6%      7.08 1.20  20.0%
 8 NumGameDay~    1.38   0     No     1.05  5.2%      3.20 1.17  17.5%
 9 NumInGameM~    1.31   0.001 No     1.00  0.1%    107.   1.14  14.4%
10 NumFriends~    1.31   0     No     1.00  0.1%     94.9  1.14  14.4%
11 TimesCapta~    1.04   0.44  No     1.00  0.2%      8.96 1.02  1.9%
12 TimesKilled    1.02   0.784 No     1.00  0.2%      3.49 1.01  0.8%
13 DaysUser       1.01   0.9   No     1.00  0.0%    659.   1.00  0.4%
```

The three most important features are NumFriends, TimesLostSpaceship, GameLevel

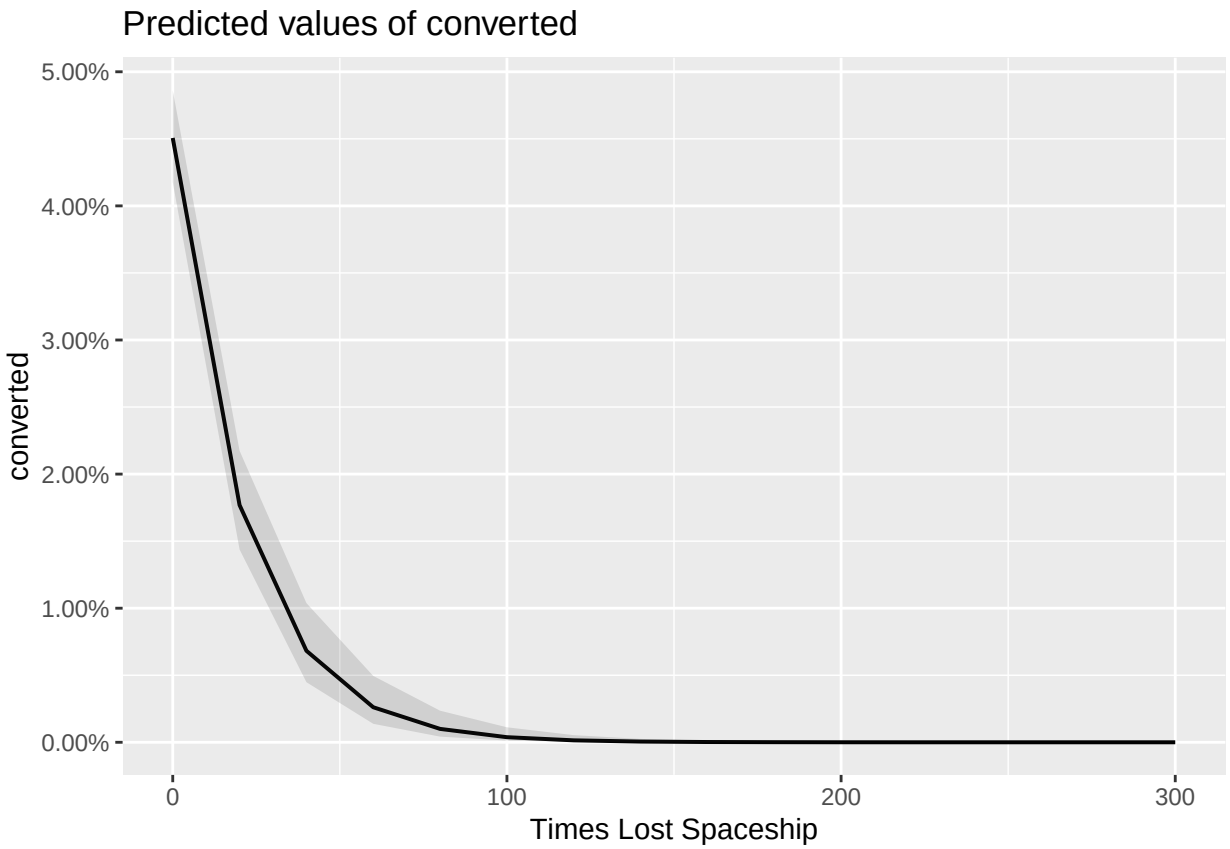
Part 2 - Question 2b.

```
plot_model(logit1, type = "eff", terms = c("NumFriends"))
```



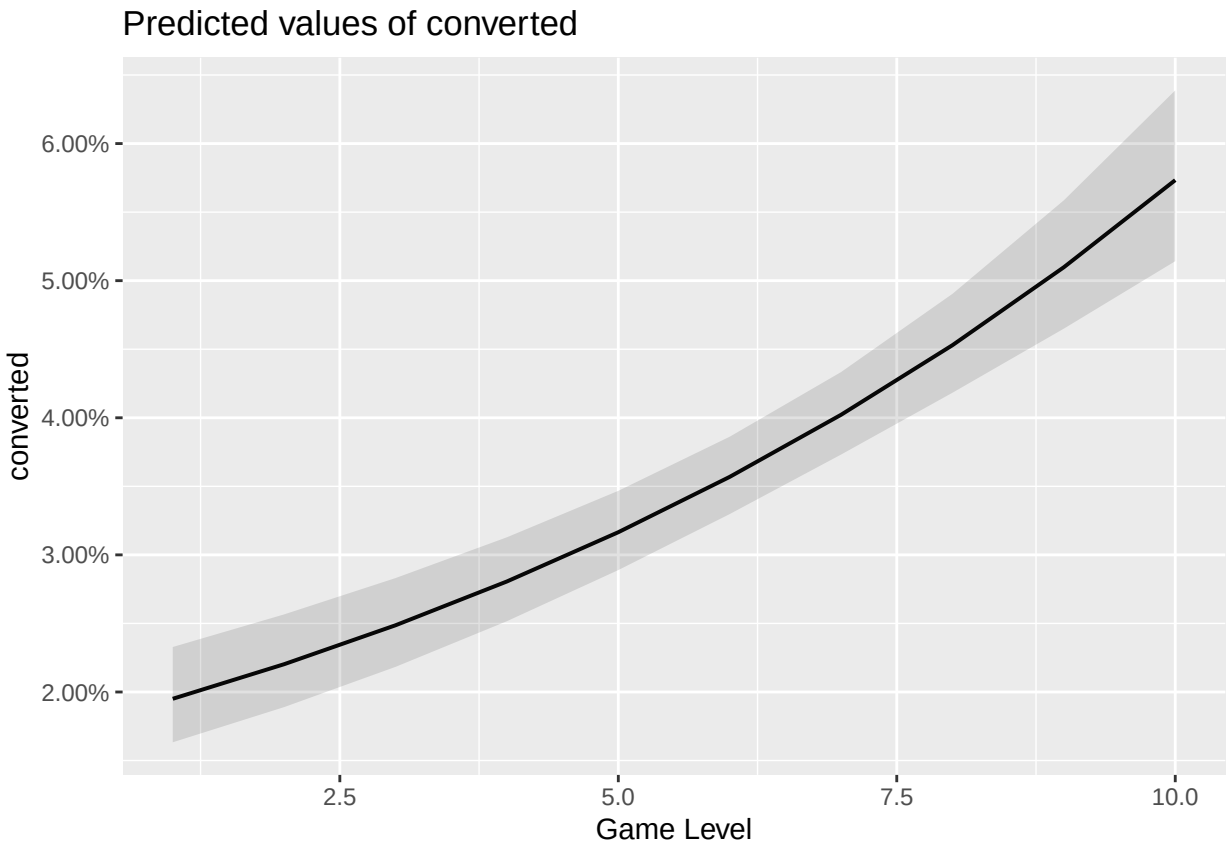
As number of friends goes from 0 to 12, the probability of conversion reaches as high as 80%. This is a high and significant correlation.

```
plot_model(logit1, type = "eff", terms = c("TimesLostSpaceship"))
```



As number of times of spaceship loss goes up, there is negligible change in probability of conversion i.e. it drops from 4.5% to 0

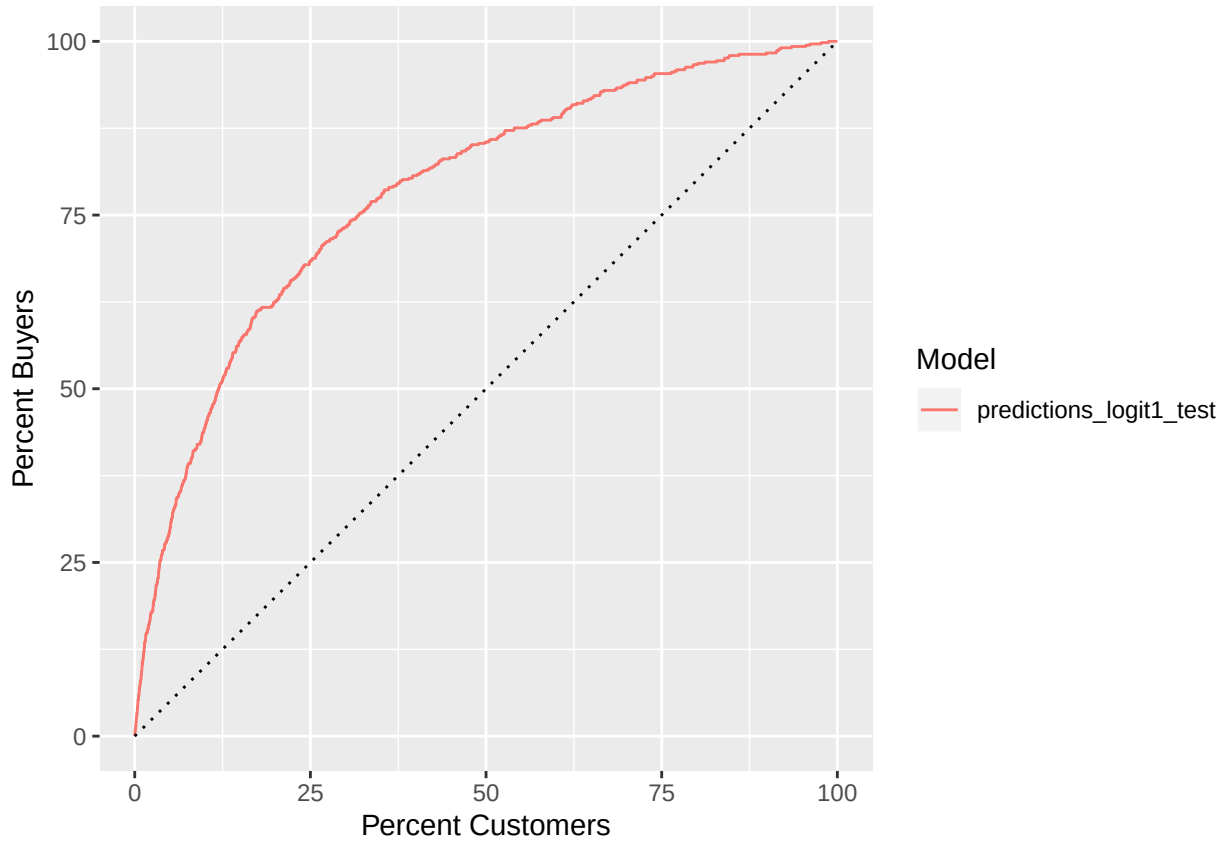
```
plot_model(logit1, type = "eff", terms = c("GameLevel"))
```



As the lebel of game achieved by user goes up from 1 to 10, the probability of conversion rises marginally from 2% to ~6%. Secondly, the confidence interval is high for all game levels which means it is difficult to accurately predict conversion for particular game levels.

Part 2 - Question 2c.

```
predictions_logit1_test <- predict(logit1, newdata = cg_organic_test, type = "response")

gainsplot(predictions_logit1_test, label.var = cg_organic_test$converted)
```




```
# A tibble: 1 x 2
  model                     auc
  <chr>                   <dbl>
1 predictions_logit1_test 0.802
```

Area under curve for gains plot is 0.802 which is a good prediction

Part 2 - Question 2d.

Choosing 30,000 customers from 200,000 means this is 30,000/200,000 = 15% of total customers model we are targeting. Now, for 15% customers, based on gains plot we will have 60% buyers.

```
cust_targeted = 30000
cust_converted = 60/100 * 5.75/100 * 200000

profit = cust_converted*14.99 - cust_targeted*1.5
profit
```

```
[1] 58431
```

The profit from model-selected 30,000 customers would be $58,431

Part 3 - Question 1.

```
group1_probability <- cg_organic_control %>%
  summarise (group1_probability = mean(converted))
```

```
profit <- group1_probability*30000*14.99
profit
```

```
  group1_probability
1              25573
```

The total profit for Group 1 is $25,573

Part 3 - Question 2.

```
cg_ad_random <- cg_ad_treatment[sample_random_30000,]

group2_probability <- cg_ad_random %>%
  summarise (group2_probability = mean(converted))
group2_probability
```

```
  group2_probability
1               0.13
```

```
profit2 <- group2_probability*30000*14.99 - 30000*1.5
profit2
```

```
  group2_probability
1              13656
```

The total profit for Group 2 is $13,656 which is lower than Group 1. This could be because of non selective targeting as well as higher advertisement cost as compared to group 1

Part 3 - Question 3.

```
cg_ad_scoring <- cg_ad_treatment[-sample_random_30000,]

predictions_logit1_scoring <- predict(logit1, newdata = cg_ad_scoring, type = "response")

cg_ad_scoring <- cg_ad_scoring %>%
  mutate(score_logit = predictions_logit1_scoring)

cg_ad_scoring_sorted <- cg_ad_scoring

cg_ad_scoring_sorted <- cg_ad_scoring_sorted %>% arrange(desc(score_logit))
```

Selecting 30,000 users having top probability of conversion and finding their mean probability of purchase

```
cg_ad_scoring_sorted_group3 <- cg_ad_scoring_sorted[1:30000,]

group3_probability <- cg_ad_scoring_sorted_group3 %>%
  summarise (group3_probability = mean(converted))
group3_probability
```

```
  group3_probability
1               0.21
```

```
profit3 <- group3_probability*30000*14.99 - 30000*1.5
profit3
```

```
  group3_probability
1              49407
```

The total profit for Group 3 is $49,407.

Part 3 - Question 4

Based on targeting all 30,000 customers using the model, although we are targeting the top 30,000 customers but their overall response rate is still low and we are paying for advertisements to all top 30K customers. We should only be targeting the ones which are having better than breakeven response rate. In this case, these would be less than 30,000 customers.

Part 3 - Question 5

Group 1 helps reevaluate the conversion rate when some users are receiving the adverstisement. It might happen that the control group is affected by advertisements as control group users could be friends with ad treatment users. However, by doing a double check - we are making an accurate assessment of control group or organic conversion.

Part 3 - Question 6

This is because in the prediction model, we have used the organic conversion data where as the actual calculation is based on the ad_treatment data. We need to include the experimental data to train the model so that it gives accurate results.

Part 4 - Question 1

```
logit2 <- glm(converted ~ DaysUser + GameLevel + NumAdsClicked + NumFriendRequestIgnored + NumFriends +
, data=cg_ad_random)
summary(logit2)
```

```
Call:
glm(formula = converted ~ DaysUser + GameLevel + NumAdsClicked +
    NumFriendRequestIgnored + NumFriends + NumFriendsofFriends +
    NumGameDays + NumGameDaysOnline + NumInGameMessagesSent +
    TimesCaptain + TimesKilled + TimesLostSpaceship + TimesNavigator,
    family = binomial(logit), data = cg_ad_random)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.799  -0.518  -0.421  -0.341   2.637

Coefficients:
                           Estimate  Std. Error z value
(Intercept)             -3.55512330  0.09540015  -37.27
DaysUser                 0.00002517  0.00002767    0.91
GameLevel                0.05319659  0.00749643    7.10
NumAdsClicked            0.08933733  0.00224455   39.80
NumFriendRequestIgnored -0.00000423  0.00065552   -0.01
NumFriends               0.02230701  0.00792855    2.81
NumFriendsofFriends      0.00179260  0.00016529   10.85
NumGameDays              0.01486709  0.00295104    5.04
NumGameDaysOnline        0.01209034  0.00504779    2.40
NumInGameMessagesSent   -0.00017926  0.00021055   -0.85
TimesCaptain             0.00611612  0.00200570    3.05
TimesKilled             -0.00166218  0.00531526   -0.31
TimesLostSpaceship      -0.00636856  0.00200375   -3.18
TimesNavigator          -0.00179145  0.00252531   -0.71
                                    Pr(>|z|)
(Intercept)             < 0.0000000000000002 ***
DaysUser                             0.3629
GameLevel                0.0000000000013 ***
NumAdsClicked           < 0.0000000000000002 ***
NumFriendRequestIgnored              0.9949
NumFriends                           0.0049 **
NumFriendsofFriends     < 0.0000000000000002 ***
NumGameDays              0.0000004706421 ***
NumGameDaysOnline                    0.0166 *
NumInGameMessagesSent                0.3946
TimesCaptain                         0.0023 **
TimesKilled                          0.7545
TimesLostSpaceship                   0.0015 **
TimesNavigator                       0.4781
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 23233  on 29999  degrees of freedom
Residual deviance: 21073  on 29986  degrees of freedom
AIC: 21101

Number of Fisher Scoring iterations: 5
```

Part 4 - Question 1

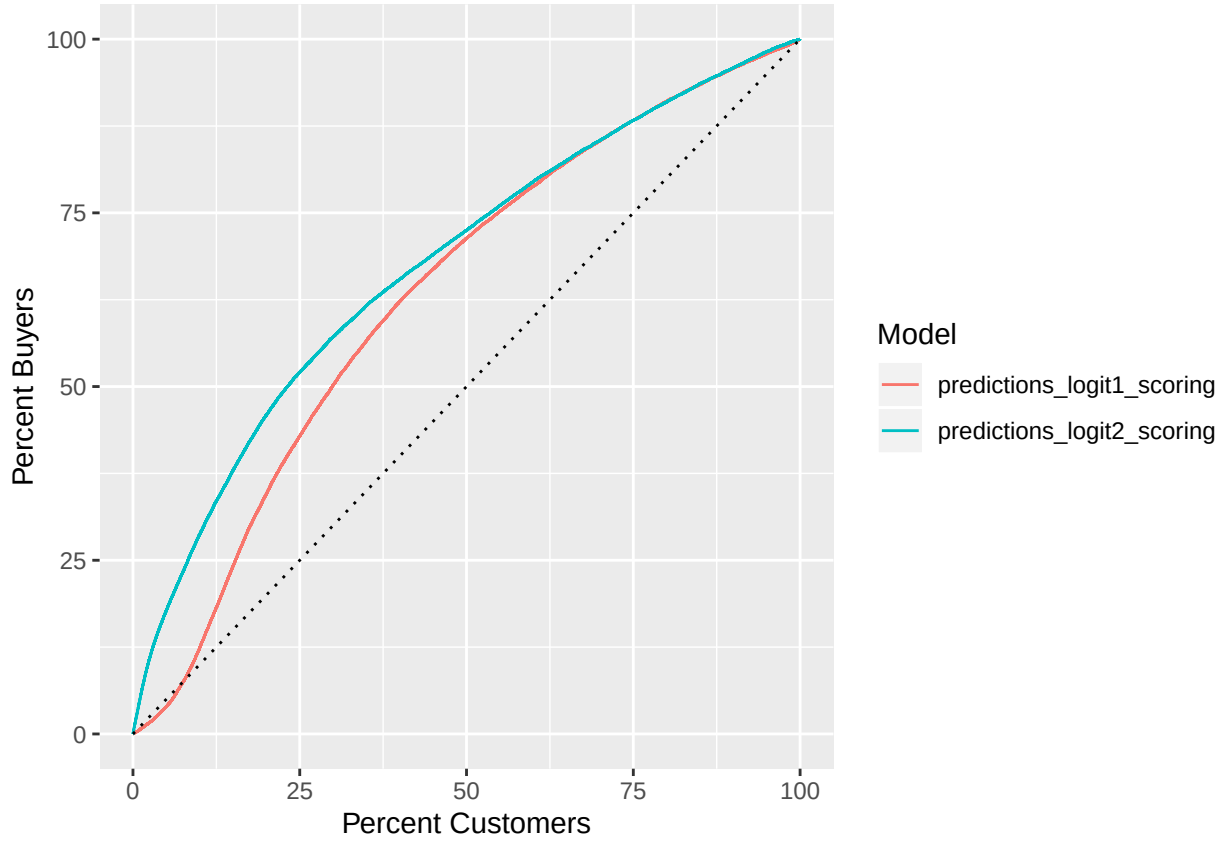
```
logit2 <- glm(converted ~ DaysUser + GameLevel + NumAdsClicked + NumFriendRequestIgnored + NumFriends +
, data=cg_ad_random)
summary(logit2)
```

```
Call:
glm(formula = converted ~ DaysUser + GameLevel + NumAdsClicked +
    NumFriendRequestIgnored + NumFriends + NumFriendsofFriends +
    NumGameDays + NumGameDaysOnline + NumInGameMessagesSent +
    TimesCaptain + TimesKilled + TimesLostSpaceship + TimesNavigator,
    family = binomial(logit), data = cg_ad_random)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.799  -0.518  -0.421  -0.341   2.637

Coefficients:
                          Estimate  Std. Error z value
(Intercept)             -3.55512330  0.09540015  -37.27
DaysUser                 0.00002517  0.00002767    0.91
GameLevel                0.05319659  0.00749643    7.10
NumAdsClicked            0.08933733  0.00224455   39.80
NumFriendRequestIgnored -0.00000423  0.00065552   -0.01
NumFriends               0.02230701  0.00792855    2.81
NumFriendsofFriends      0.00179260  0.00016529   10.85
NumGameDays              0.01486709  0.00295104    5.04
NumGameDaysOnline        0.01209034  0.00504779    2.40
NumInGameMessagesSent   -0.00017926  0.00021055   -0.85
TimesCaptain             0.00611612  0.00200570    3.05
TimesKilled             -0.00166218  0.00531526   -0.31
TimesLostSpaceship      -0.00636856  0.00200375   -3.18
TimesNavigator          -0.00179145  0.00252531   -0.71
                                    Pr(>|z|)
(Intercept)             < 0.0000000000000002 ***
DaysUser                             0.3629
GameLevel                0.0000000000013 ***
NumAdsClicked           < 0.0000000000000002 ***
NumFriendRequestIgnored              0.9949
NumFriends                           0.0049 **
NumFriendsofFriends     < 0.0000000000000002 ***
NumGameDays              0.0000004706421 ***
NumGameDaysOnline                    0.0166 *
NumInGameMessagesSent                0.3946
TimesCaptain                         0.0023 **
TimesKilled                          0.7545
TimesLostSpaceship                   0.0015 **
TimesNavigator                       0.4781
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 23233  on 29999  degrees of freedom
Residual deviance: 21073  on 29986  degrees of freedom
AIC: 21101

Number of Fisher Scoring iterations: 5
```

Part 4 - Question 2

```
predictions_logit2_scoring <- predict(logit2, newdata = cg_ad_scoring, type = "response")

cg_ad_scoring <- cg_ad_scoring %>%
  mutate(score_logit2 = predictions_logit2_scoring)

gainsplot(predictions_logit1_scoring, predictions_logit2_scoring, label.var = cg_ad_scoring$converted)
```




```
# A tibble: 2 x 2
  model                          auc
  <chr>                        <dbl>
1 predictions_logit1_scoring 0.652
2 predictions_logit2_scoring 0.702
```

The trained model which uses the experimental data is a better model as the AUC is higher.

Part 4 - Question 3

```
cg_ad_scoring_sorted_new <- cg_ad_scoring

cg_ad_scoring_sorted_new <- cg_ad_scoring_sorted_new %>% arrange(desc(score_logit2))
```

```
cg_ad_scoring_sorted_new_target30k <- cg_ad_scoring_sorted_new[1:30000,]

trained_probability <- cg_ad_scoring_sorted_new_target30k %>%
  summarise (trained_probability = mean(converted))
trained_probability
```

```
  trained_probability
1                0.33
```

```
profit_actual <- trained_probability*30000*14.99 - 30000*1.5
profit_actual
```

```
  trained_probability
1              103431
```

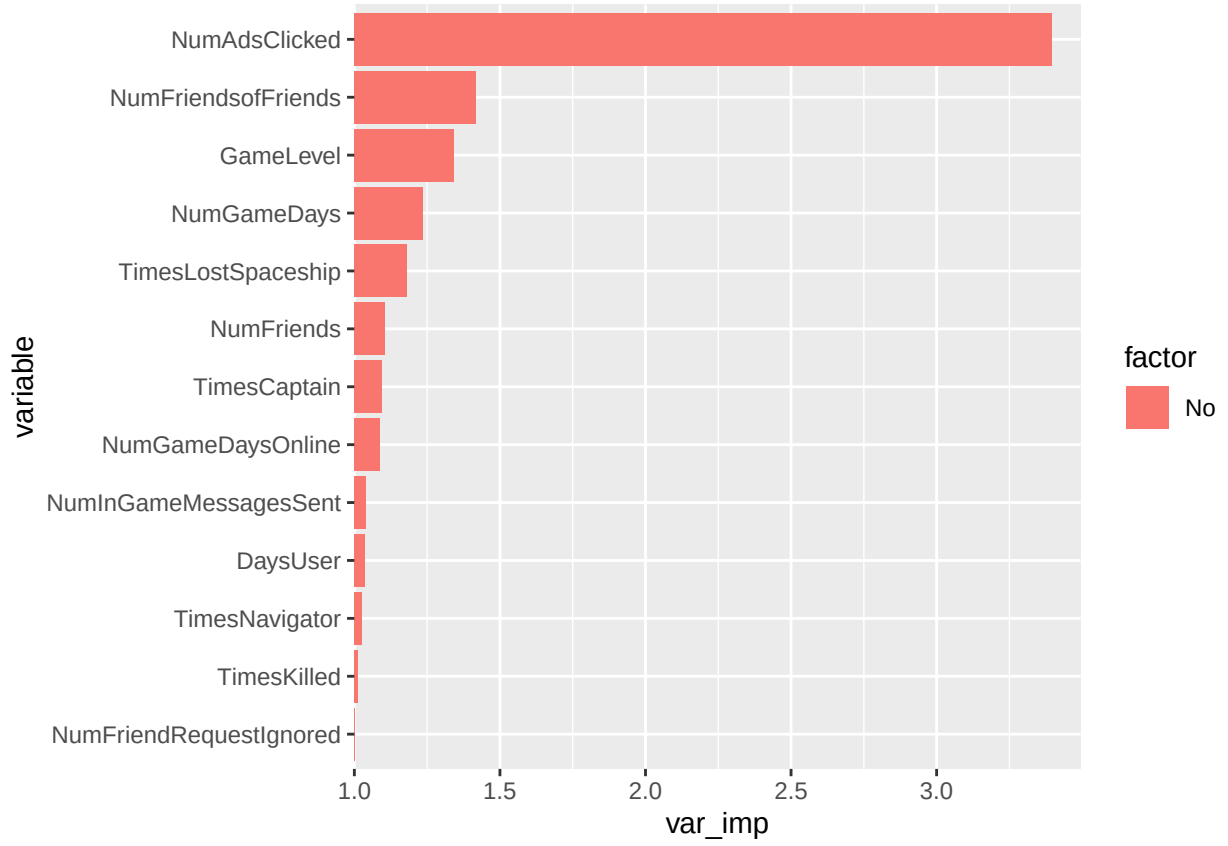
```
profit_improvement <- profit_actual - profit3
profit_improvement
```

```
  trained_probability
1               54024
```

The profit improves by $54,024

Part 4 - Question 4

```
varimp.logistic(logit2) %>% plotimp.logistic()
```




```
# A tibble: 13 x 9
   variable    var_imp p_value factor    OR OR_perc     sd OR_sd OR_sd_perc
   <chr>         <dbl>   <dbl> <chr>  <dbl> <chr>    <dbl> <dbl> <chr>
 1 NumAdsClic~    3.39   0     No     1.09  9.3%      6.84 1.84  84.2%
 2 NumFriends~    1.42   0     No     1.00  0.2%     97.5  1.19  19.1%
 3 GameLevel      1.34   0     No     1.05  5.5%      2.77 1.16  15.9%
 4 NumGameDays    1.24   0     No     1.01  1.5%      7.12 1.11  11.2%
 5 TimesLostS~    1.18   0.001 No     0.994 -0.6%    13.0  0.920 -8.0%
 6 NumFriends     1.11   0.005 No     1.02  2.3%      2.24 1.05  5.1%
 7 TimesCapta~    1.10   0.002 No     1.01  0.6%      7.47 1.05  4.7%
 8 NumGameDay~    1.09   0.017 No     1.01  1.2%      3.44 1.04  4.2%
 9 NumInGameM~    1.04   0.395 No     1.000 -0.0%   108.   0.981 -1.9%
10 DaysUser       1.03   0.363 No     1.00  0.0%    663.   1.02  1.7%
11 TimesNavig~    1.02   0.478 No     0.998 -0.2%     6.49 0.988 -1.2%
12 TimesKilled    1.01   0.754 No     0.998 -0.2%     3.53 0.994 -0.6%
13 NumFriendR~    1.00   0.995 No     1.000 -0.0%    34.3  1.000 -0.0%
```

The two models differ because we can see that the logit2 model i.e. the trained model has NumAdsClicked as the super important feature where as this variable was not in most important features in the previous model. Since the adverstisements play a crucial role in changing user behavior, we can find such difference between the results of two models.