Group Assignment 4 - Creative Gaming

Section 51
Gaurav Agrawal, Ajitesh Abhishek, Tarun Joshi

```
   checking for file 'C:\Users\a1aji\AppData\Local\Temp\RtmpQhl1bv\remotes11106f27160e\fzettelmeyer-mktg

v  checking for file 'C:\Users\a1aji\AppData\Local\Temp\RtmpQhl1bv\remotes11106f27160e\fzettelmeyer-mktg



-  preparing 'mktg482':
   checking DESCRIPTION meta-information ...

   checking DESCRIPTION meta-information ...

v  checking DESCRIPTION meta-information



-  checking for LF line-endings in source and make files and shell scripts


-  checking for empty or unneeded directories



-  building 'mktg482_0.0.3.0.tar.gz'
```

Read in the data:

```
# use load("filename.Rdata") for .Rdata files
data = load("smobile_churn.Rdata")
smobile <- smobile %>% mutate(churn = ifelse(churn == "X1", 1, 0))
```

##Question 1: Step 1

```
data.train <- smobile %>%
  filter(training==1)
```

```r
data.test <- smobile %>%
filter(training==0)
```

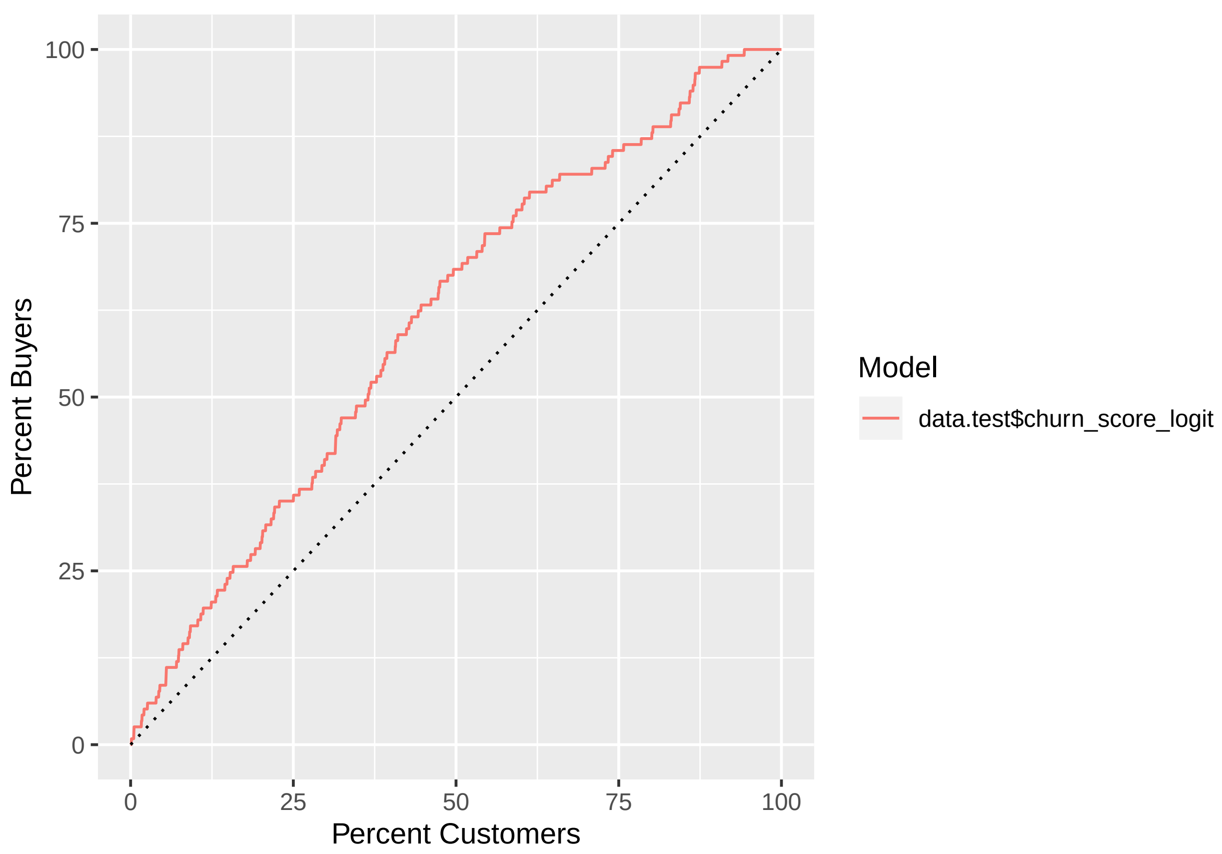
Question 1

```r
fm <- formula(churn ~.  -customer)
logit.churn <- glm(fm, family=binomial(logit), data = data.train)

predictions_logit.churn <- predict(logit.churn,
                                   newdata = data.test,
                                   type = "response")

data.test <- data.test %>%
  mutate(churn_score_logit = predictions_logit.churn)

gainsplot(data.test$churn_score_logit,label.var = data.test$churn)
```




```
# A tibble: 1 x 2
  model                          auc
  <chr>                        <dbl>
1 data.test$churn_score_logit 0.614
```

```r
summary(logit.churn)
```

```
Call:
glm(formula = fm, family = binomial(logit), data = data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9621  -0.3140  -0.2634  -0.2158   3.1756

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.0790100  0.3927087  -7.840 4.49e-15 ***
revenue      0.0023081  0.0026485   0.871 0.383512
mou         -0.0005223  0.0002282  -2.289 0.022082 *
overage      0.0012991  0.0008707   1.492 0.135689
roam         0.0011300  0.0091256   0.124 0.901454
changem     -0.0001456  0.0003150  -0.462 0.644007
changer     -0.0011700  0.0020578  -0.569 0.569664
dropvce      0.0003786  0.0088781   0.043 0.965982
blckvce      0.0157017  0.0039198   4.006 6.18e-05 ***
unansvce     0.0013042  0.0021420   0.609 0.542628
custcare    -0.0059974  0.0168837  -0.355 0.722428
threeway    -0.0233032  0.0328647  -0.709 0.478284
months      -0.0181096  0.0107572  -1.683 0.092281 .
uniqsubs     0.1638001  0.0636055   2.575 0.010017 *
phones       0.0311120  0.0708582   0.439 0.660607
eqpdays      0.0013142  0.0003807   3.452 0.000557 ***
age         -0.0162599  0.0068316  -2.380 0.017308 *
children1   -0.0434708  0.1643164  -0.265 0.791352
creditaa1   -0.5547280  0.2264624  -2.450 0.014304 *
refurb1      0.0265338  0.1917298   0.138 0.889931
occprof1     0.2074751  0.1891593   1.097 0.272717
occcler1    -0.0641170  0.5219908  -0.123 0.902240
occcrft1     0.5967373  0.3194304   1.868 0.061744 .
occstud1     0.2242857  0.7366422   0.304 0.760770
occhmkr1     0.8009078  0.7505756   1.067 0.285946
occret1      0.1974465  0.6198244   0.319 0.750066
occself1     0.3147290  0.4757480   0.662 0.508262
travel1     -0.1672166  0.3064518  -0.546 0.585304
retcalls     0.4969203  0.2733701   1.818 0.069101 .
refer       -0.5650638  0.3783102  -1.494 0.135267
incmiss1     0.1611407  0.3196411   0.504 0.614170
income      -0.0122394  0.0354409  -0.345 0.729833
mcycle1     -0.1849379  0.5972747  -0.310 0.756838
agemiss1    -0.4348835  0.3683478  -1.181 0.237749
training            NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2325.7  on 6999  degrees of freedom
```

```
Residual deviance: 2245.4  on 6966  degrees of freedom
AIC: 2313.4

Number of Fisher Scoring iterations: 6
```

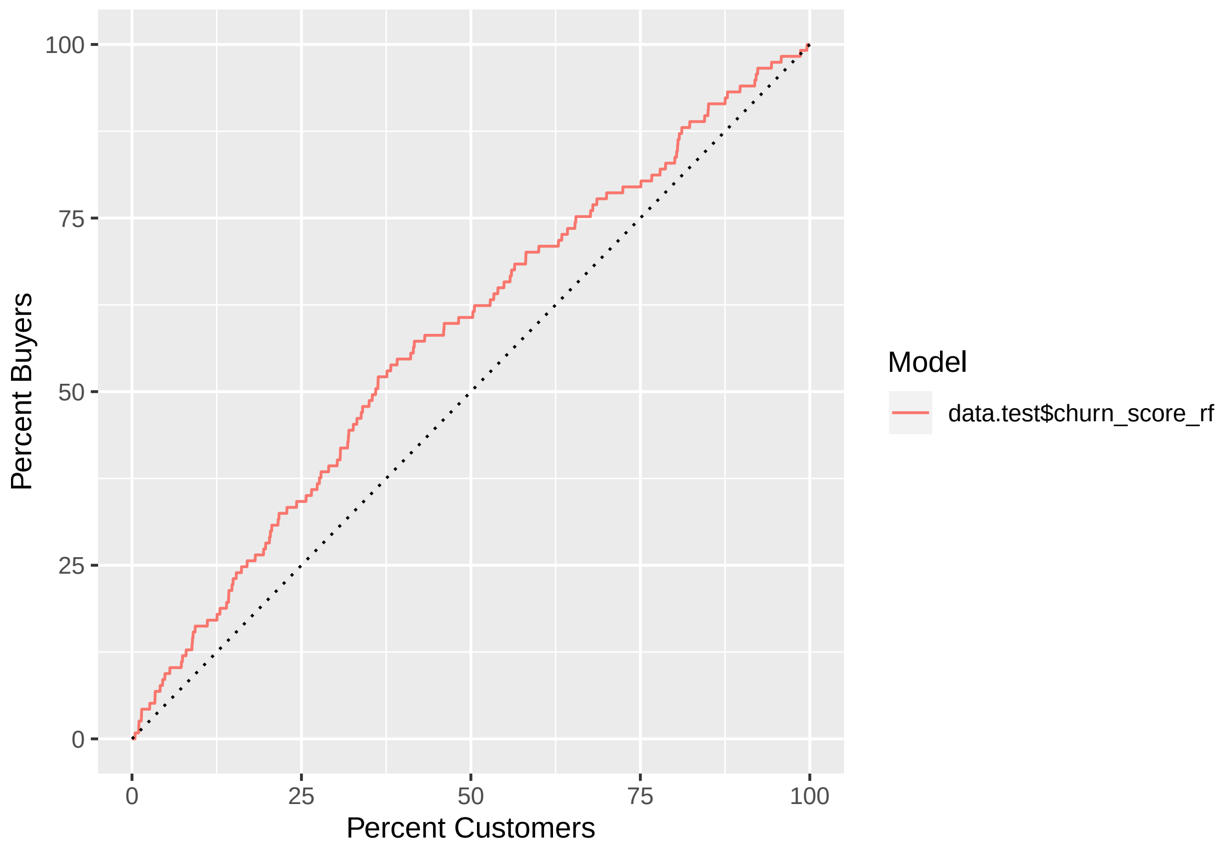
```
fm <- formula(churn ~. -customer)

random.churn <- ranger(fm, data=data.train, probability = TRUE)

predictions_rf.churn <- predict(random.churn, data = data.test, type="response")[[1]][,2]

data.test <- data.test %>%
  mutate(churn_score_rf = predictions_rf.churn)

gainsplot(data.test$churn_score_rf,label.var = data.test$churn)
```




```
# A tibble: 1 x 2
  model                       auc
  <chr>                     <dbl>
1 data.test$churn_score_rf 0.582
```

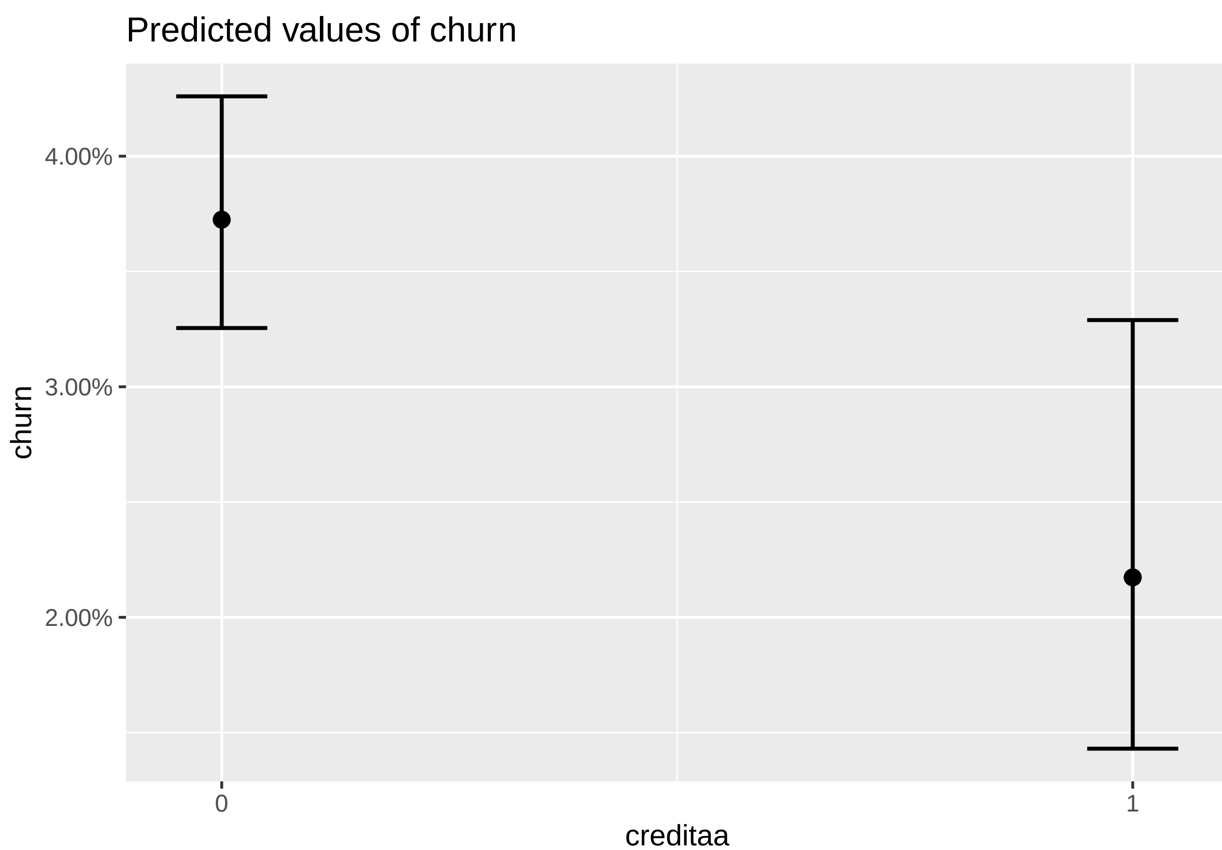
Cofficient	Value	Std Error	isDummy	Importance
creditaa1	-0.55473	0.226462	1	0.554728
uniqsubs	0.1638	0.063606	0	0.2910111
age	-0.01626	0.006832	0	0.0299231
blckvce	0.015702	0.00392	0	0.0235413
eqpdays	0.001314	0.000381	0	0.0020756
mou	-0.00052	0.000228	0	0.0009787

Figure 1: Variable Importance

Question 2

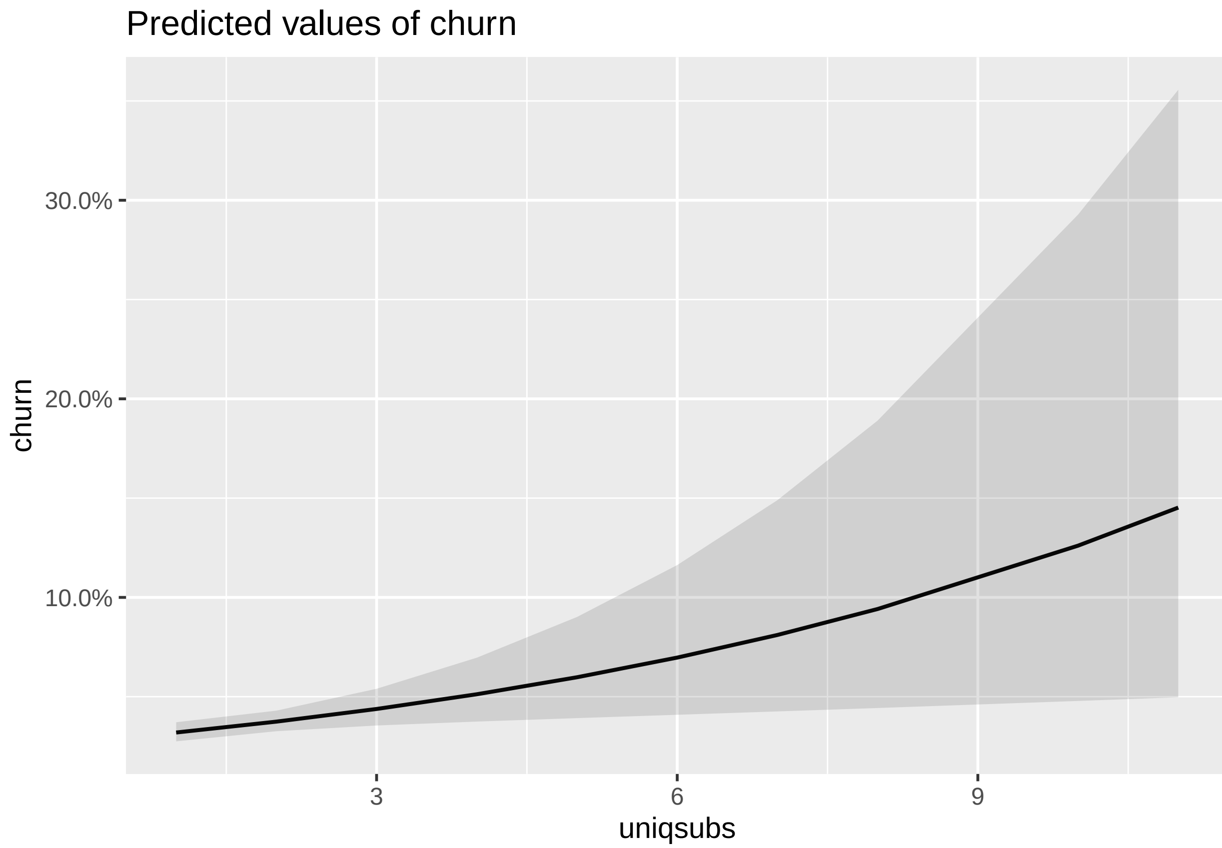
Question 3

```
plot_model(logit.churn, type="eff", terms = c("creditaa"))
```

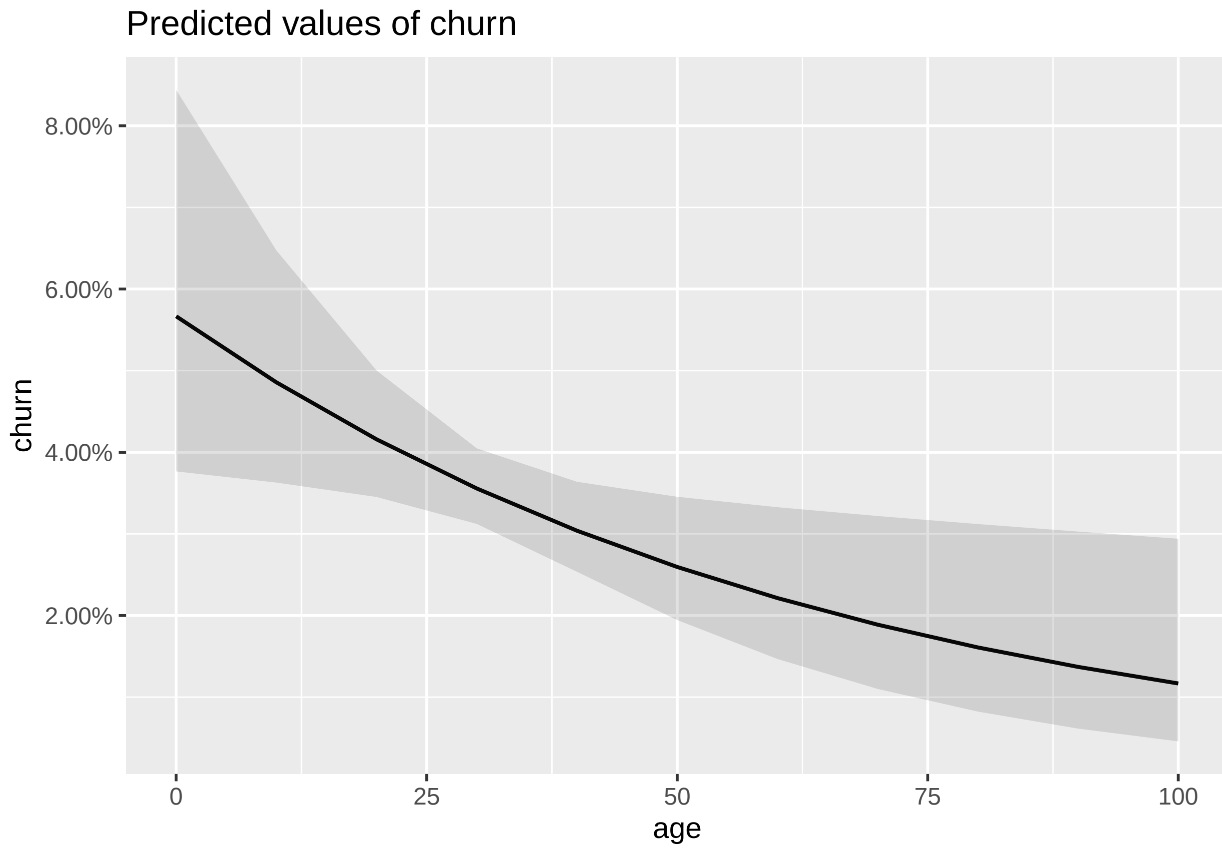


Customers with high credit rating have lower churn. Hence, the firm should take extra care of cutsomers who don't high rating - offer discounts, monitor satisfaction, priority queue for customer complaint handling etc.

```
plot_model(logit.churn, type="eff", terms = c("uniqsubs"))
```

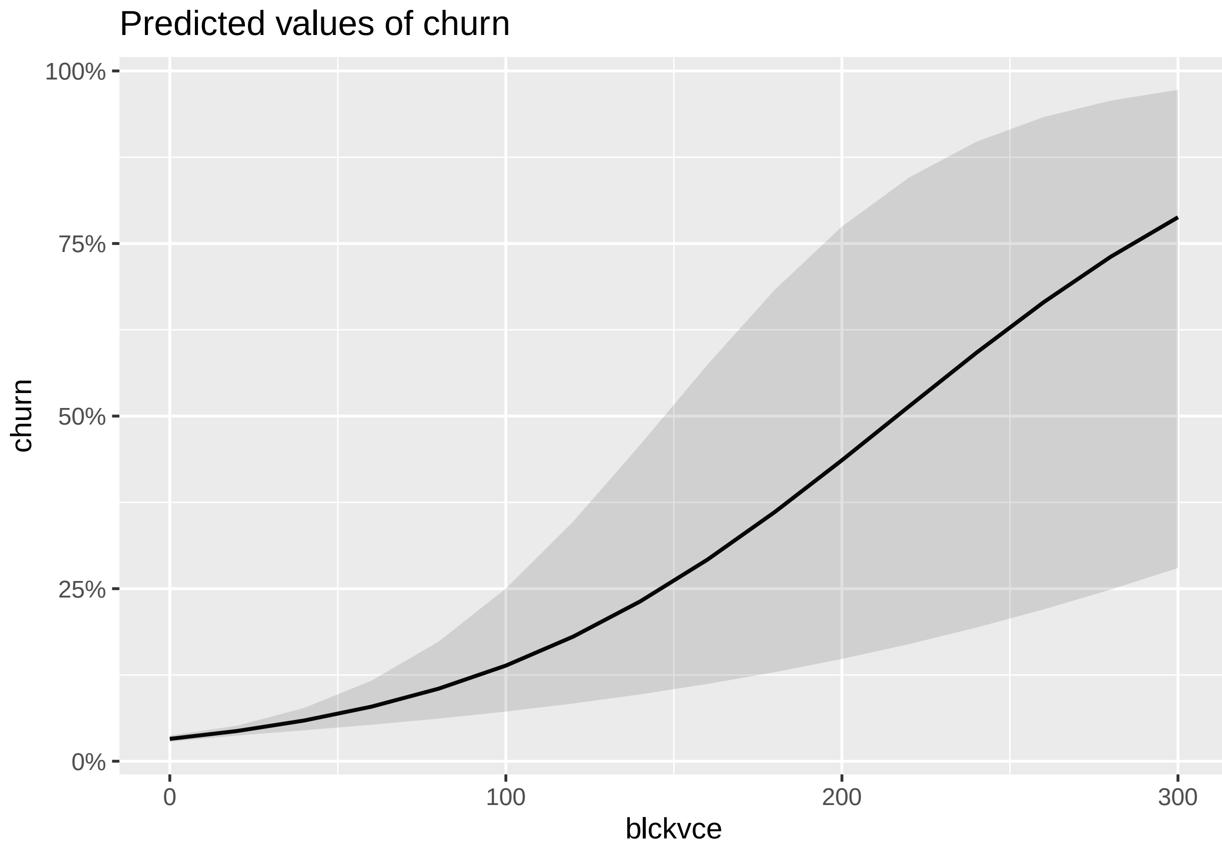



```
plot_model(logit.churn, type="eff", terms = c("age"))
```



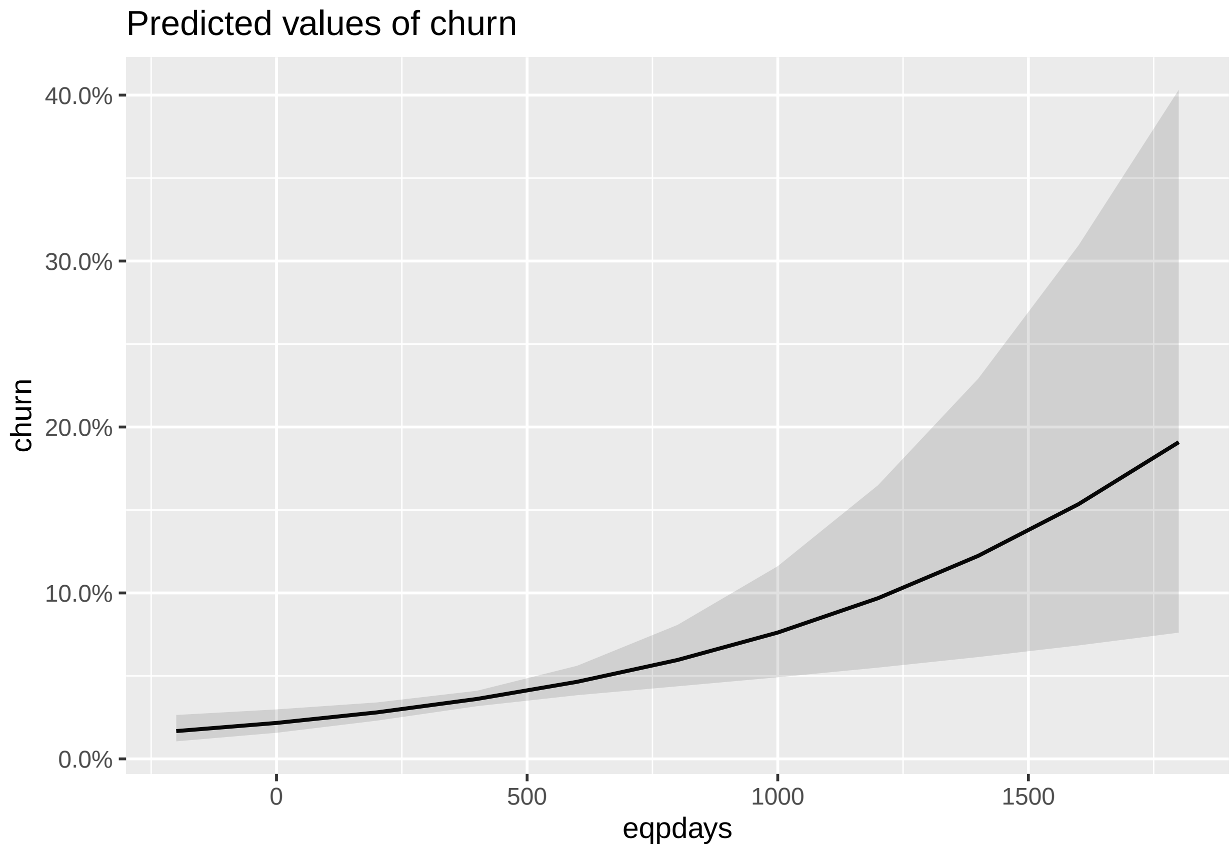
Launch “For Senior Citizen” plan offering expediated customer service and enrol people above age 50 years.

```
plot_model(logit.churn, type="eff", terms = c("blckvce"))
```



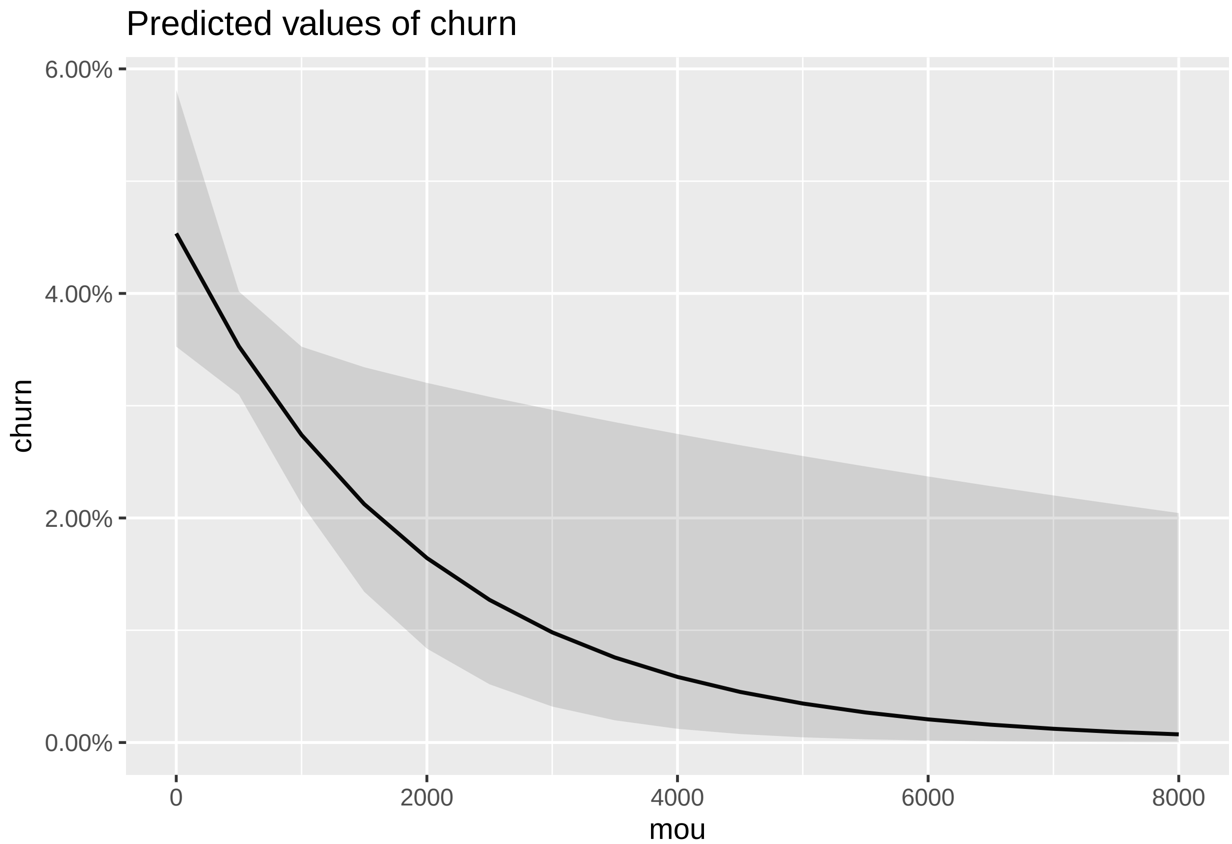
Consumers usually block the spam calls. Hence, work with developer on spam detection model. Lower spam could reduce the churn for the firm.

```
plot_model(logit.churn, type="eff", terms = c("eqpdays"))
```



Tieup with equipment provider to help customer switch to new devices. Offer attractive plan to users with new devices.

```
plot_model(logit.churn, type="eff", terms = c("mou"))
```



Build tariff plan offering high minutes to each tier. Also, reduce the call rates or added incentives such as extra 4G data to customers with high monthly minutes of use.

Question 4

```
rollout <- rollout %>%
  mutate(training = 1)

predictions_logit.churn.rollout <- predict(logit.churn, newdata = rollout, type = "response")


rollout <- rollout %>%
  mutate(churn_score_logit = predictions_logit.churn.rollout)
```

```
rollout_new <- rollout %>%
    mutate(age=age+10)

churn_age_pred <- predict(logit.churn, newdata = rollout_new, type = "response")


rollout_new  <- rollout_new  %>%
  mutate(churn_age_pred = churn_age_pred)

orig_churn <- rollout_new %>%
```

```r
  summarise(mean(churn_score_logit)*100)

new_churn_age <- rollout_new %>%
  summarise(mean(churn_age_pred)*100)

change_per_age <- new_churn_age - orig_churn

print(orig_churn)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            3.93
```

```r
print(new_churn_age)
```

```
# A tibble: 1 x 1
  `mean(churn_age_pred) * 100`
                         <dbl>
1                         3.37
```

```r
print(change_per_age)
```

```
  mean(churn_age_pred) * 100
1                 -0.5643104
```

```r
rollout_new <- rollout %>%
    mutate(creditaa = "1")

churn_credit_pred <- predict(logit.churn, newdata = rollout_new, type = "response")


rollout_new  <- rollout_new  %>%
  mutate(churn_credit_pred = churn_credit_pred)

orig_churn <- rollout_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_credit <- rollout_new %>%
  summarise(mean(churn_credit_pred)*100)

change_per_credit <- new_churn_credit - orig_churn

print(orig_churn)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            3.93
```

```r
print(new_churn_credit)
```

```
# A tibble: 1 x 1
  `mean(churn_credit_pred) * 100`
                            <dbl>
1                            2.45
```

```r
print(change_per_credit)
```

```
  mean(churn_credit_pred) * 100
1                      -1.48235
```

```r
rollout_new <- rollout %>%
    mutate(blckvce = blckvce*0.5)

churn_blckvce_pred <- predict(logit.churn, newdata = rollout_new, type = "response")


rollout_new  <- rollout_new  %>%
  mutate(churn_blckvce_pred = churn_blckvce_pred)

orig_churn <- rollout_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_blckvce <- rollout_new %>%
  summarise(mean(churn_blckvce_pred)*100)

change_per_blckvce <- new_churn_blckvce - orig_churn

print(orig_churn)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            3.93
```

```r
print(new_churn_blckvce)
```

```
# A tibble: 1 x 1
  `mean(churn_blckvce_pred) * 100`
                             <dbl>
1                             3.79
```

```r
print(change_per_blckvce)
```

```
  mean(churn_blckvce_pred) * 100
1                      -0.146341
```

```
rollout_new <- rollout %>%
    mutate(uniqsubs = uniqsubs-1)

churn_uniqsubs_pred <- predict(logit.churn, newdata = rollout_new, type = "response")


rollout_new  <- rollout_new  %>%
  mutate(churn_uniqsubs_pred = churn_uniqsubs_pred)

orig_churn <- rollout_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_uniqsubs <- rollout_new %>%
  summarise(mean(churn_uniqsubs_pred)*100)

change_per_uniqsubs <- new_churn_uniqsubs - orig_churn

print(orig_churn)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            3.93
```

```
print(new_churn_uniqsubs)
```

```
# A tibble: 1 x 1
  `mean(churn_uniqsubs_pred) * 100`
                              <dbl>
1                              3.37
```

```
print(change_per_uniqsubs)
```

```
  mean(churn_uniqsubs_pred) * 100
1                      -0.5681776
```

```
rollout_new <- rollout %>%
    mutate(eqpdays = eqpdays*0.5)

churn_eqpdays_pred <- predict(logit.churn, newdata = rollout_new, type = "response")


rollout_new  <- rollout_new  %>%
  mutate(churn_eqpdays_pred = churn_eqpdays_pred)

orig_churn <- rollout_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_eqpdays <- rollout_new %>%
  summarise(mean(churn_eqpdays_pred)*100)
```

```r
change_per_eqpdays <- new_churn_eqpdays - orig_churn

print(orig_churn)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            3.93
```

```r
print(new_churn_eqpdays)
```

```
# A tibble: 1 x 1
  `mean(churn_eqpdays_pred) * 100`
                             <dbl>
1                             3.05
```

```r
print(change_per_eqpdays)
```

```
  mean(churn_eqpdays_pred) * 100
1                     -0.8810047
```

Impact on churn by just changing for a specific segment

```r
rollout %>%
  summarise(mean(mou))
```

```
# A tibble: 1 x 1
  `mean(mou)`
        <dbl>
1        540.
```

```r
rollout_mou_new <- rollout %>%
  arrange(-mou)%>%
  slice(1:800)%>%
  mutate(mou=mou+180)

churn_mou <- predict(logit.churn, newdata = rollout_mou_new, type = "response")


rollout_mou_new  <- rollout_mou_new  %>%
  mutate(churn_mou = churn_mou)

orig_churn_mou <- rollout_mou_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_mou <- rollout_mou_new %>%
  summarise(mean(churn_mou)*100)

change_mou <- new_churn_mou - orig_churn_mou

print(orig_churn_mou)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            3.34
```

```r
print(new_churn_mou)
```

```
# A tibble: 1 x 1
  `mean(churn_mou) * 100`
                    <dbl>
1                    3.06
```

```r
print(change_mou)
```

```
  mean(churn_mou) * 100
1             -0.280125
```

Changing credit rating

```r
rollout_creditaa_new <- rollout %>%
  filter(creditaa==0)%>%
  mutate(creditaa="1")

churn_creditaa <- predict(logit.churn, newdata = rollout_creditaa_new, type = "response")


rollout_creditaa_new  <- rollout_creditaa_new  %>%
  mutate(churn_creditaa = churn_creditaa)

orig_churn_creditaa <- rollout_creditaa_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_creditaa <- rollout_creditaa_new %>%
  summarise(mean(churn_creditaa)*100)

change_creditaa <- new_churn_creditaa - orig_churn_creditaa

print(orig_churn_creditaa)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            4.18
```

```r
print(new_churn_creditaa)
```

```
# A tibble: 1 x 1
  `mean(churn_creditaa) * 100`
                         <dbl>
1                         2.45
```

```
print(change_creditaa)
```

```
  mean(churn_creditaa) * 100
1                  -1.722493
```

```
avg_monthly_revenue <- rollout%>%
  summarise(mean(revenue))

avg_monthly_revenue
```

```
# A tibble: 1 x 1
  `mean(revenue)`
            <dbl>
1            58.9
```

Change Unique Subscribers:

```
rollout %>%
  tabyl(uniqsubs)
```

```
 uniqsubs    n      percent
        1 5173 0.6456565152
        2 2054 0.2563654518
        3  507 0.0632800799
        4  180 0.0224663005
        5   61 0.0076135796
        6   27 0.0033699451
        7    4 0.0004992511
        8    3 0.0003744383
        9    1 0.0001248128
       10    1 0.0001248128
       11    1 0.0001248128
```

```
rollout_uniqsubs_new <- rollout %>%
  filter(uniqsubs==2)%>%
  mutate(uniqsubs=1)

churn_uniqsubs <- predict(logit.churn, newdata = rollout_uniqsubs_new, type = "response")


rollout_uniqsubs_new  <- rollout_uniqsubs_new  %>%
  mutate(churn_uniqsubs = churn_uniqsubs)

orig_churn_uniqsubs <- rollout_uniqsubs_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_uniqsubs <- rollout_uniqsubs_new %>%
  summarise(mean(churn_uniqsubs)*100)

change_uniqsubs <- new_churn_uniqsubs - orig_churn_uniqsubs

print(orig_churn_uniqsubs)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            4.37
```

```r
print(new_churn_uniqsubs)
```

```
# A tibble: 1 x 1
  `mean(churn_uniqsubs) * 100`
                         <dbl>
1                         3.74
```

```r
print(change_uniqsubs)
```

```
  mean(churn_uniqsubs) * 100
1                 -0.6285828
```

```r
rollout_blckvce_new <- rollout %>%
  arrange(blckvce)%>%
  slice(1:800)%>%
  mutate(blckvce=blckvce*0.5)

churn_blckvce <- predict(logit.churn, newdata = rollout_mou_new, type = "response")


rollout_blckvce_new  <- rollout_blckvce_new  %>%
  mutate(churn_blckvce = churn_blckvce)

orig_churn_blckvce <- rollout_blckvce_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_blckvce <- rollout_blckvce_new %>%
  summarise(mean(churn_blckvce)*100)

change_blckvce <- new_churn_blckvce - orig_churn_blckvce

print(orig_churn_blckvce)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            4.05
```

```r
print(new_churn_blckvce)
```

```
# A tibble: 1 x 1
  `mean(churn_blckvce) * 100`
                        <dbl>
1                        3.06
```

```r
print(change_blckvce)
```

```
  mean(churn_blckvce) * 100
1                -0.9895624
```

```r
rollout %>%
  arrange(-eqpdays)%>%
  slice(1:800)%>%
  summarise(mean(eqpdays))
```

```
# A tibble: 1 x 1
  `mean(eqpdays)`
            <dbl>
1            901.
```

```r
rollout_eqpdays_new <- rollout %>%
  arrange(-eqpdays)%>%
  slice(1:800)%>%
  mutate(eqpdays=2*365)

churn_eqpdays <- predict(logit.churn, newdata = rollout_eqpdays_new, type = "response")


rollout_eqpdays_new  <- rollout_eqpdays_new  %>%
  mutate(churn_eqpdays = churn_eqpdays)

orig_churn_eqpdays <- rollout_eqpdays_new %>%
  summarise(mean(churn_score_logit)*100)

new_churn_eqpdays <- rollout_eqpdays_new %>%
  summarise(mean(churn_eqpdays)*100)

change_eqpdays <- new_churn_eqpdays - orig_churn_eqpdays

print(orig_churn_eqpdays)
```

```
# A tibble: 1 x 1
  `mean(churn_score_logit) * 100`
                            <dbl>
1                            5.81
```

```r
print(new_churn_eqpdays)
```

```
# A tibble: 1 x 1
  `mean(churn_eqpdays) * 100`
                        <dbl>
1                        4.71
```

```r
print(change_eqpdays)
```

```
  mean(churn_eqpdays) * 100
1                 -1.098725
```

Question 5

Based on impact on churn, we would target spam control and subsidizing purchase of new devices for customers. These two action have significanly higher impact on churn 4.404% to 3.05% and 5.8% to 4.7% respectively.