Class 15 Demo: Simulating an Action

```
### Load packages:
#library(knitr)
#library(janitor)
#library(readxl)
#library(psych)
#library(statar)
#library(tictoc)
library(mktg482)
library(tidyverse)
library(sjPlot)
#library(caret)
#library(glmnet)
#library(printr)
```

Introduction

This document shows a method for simuating the effect of an action, specifically how to simulate the effect of changes in variables on outcomes of interest. We will use the Intuit Quickbooks data to show how the probability of response to the offer would change if we could alter specific variables.

We begin by setting a seed, loading the data, and separating the training and test samples.

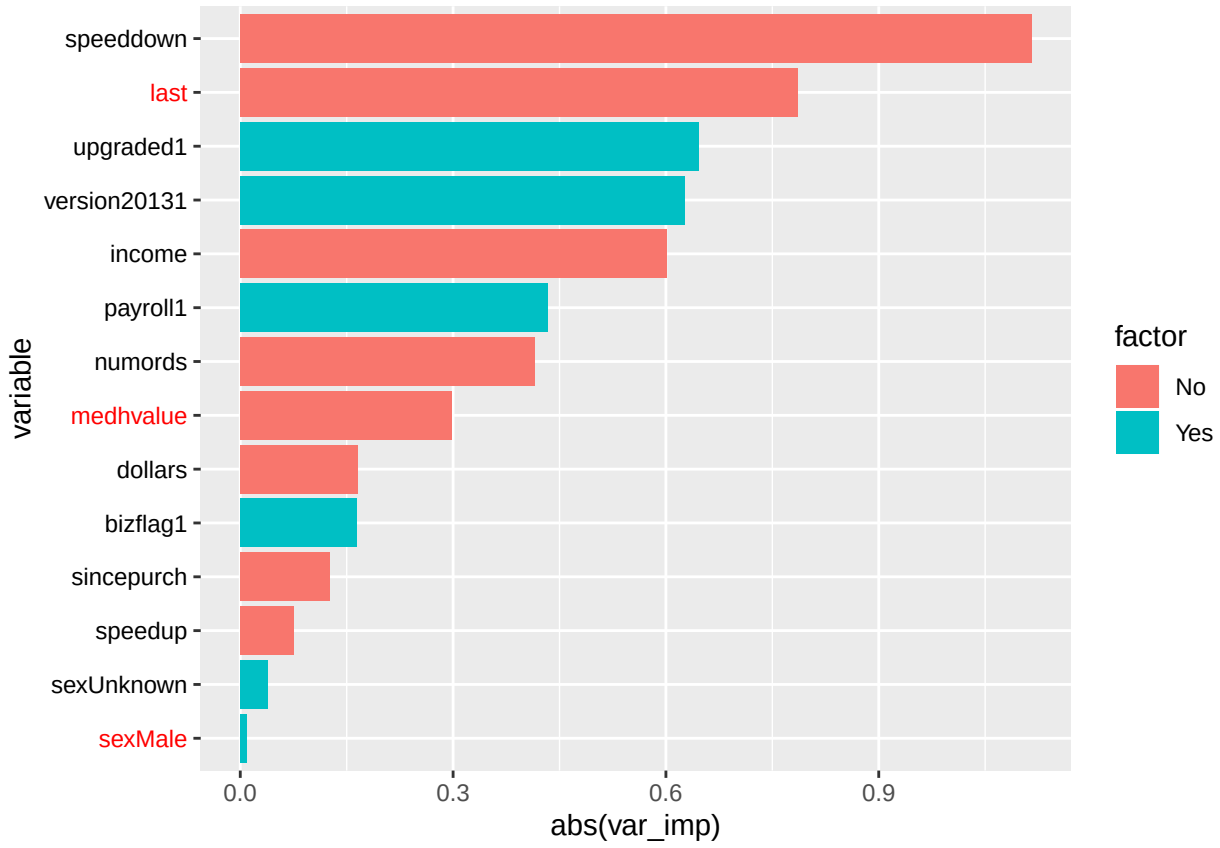
```
# Clear environment of datasets, set seed, and load data:
rm(list=ls())
set.seed(1989)
load("../../Data/intuit_online.RData")
intuit.train <- intuit %>% filter(training == 1)
intuit.test <- intuit %>% filter(training == 0)
```

For the purpose of this demo, we will keep it simple and stick to logistic regression. Consequently, we specificy a formula for the predictor variables (or features) we want to use in our logistic regression and estimate it. Again, to keep it simple, we will use all variables except for `state` (and of course `zip`).

```
fm <- as.formula(res ~ speeddown + speedup + last + numords +
                 dollars + sincepurch + version2013 + upgraded +
                 payroll + bizflag + sex + income + medhvalue)
lr1 <- glm(fm, data=intuit.train, family=binomial)
```

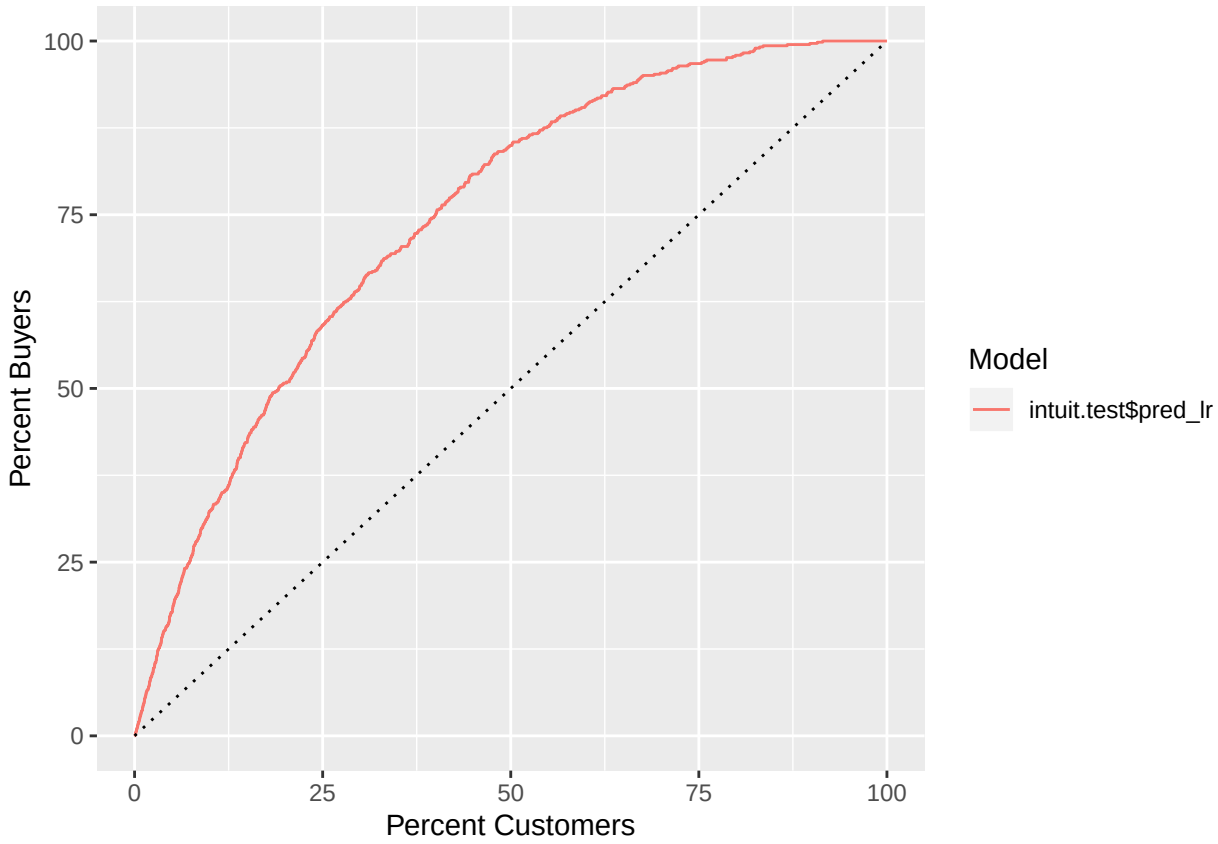
To remind ourselves, we will assess variable importance and predictive performance

```
varimp.logistic(lr1) %>% plotimp.logistic()
```




```
# A tibble: 14 x 6
   variable     factor  var_imp var_imp_lower var_imp_upper p_value
   <chr>        <chr>     <dbl>         <dbl>         <dbl>   <dbl>
 1 speeddown    No      1.12           0.973         1.26    0
 2 last         No     -0.786         -0.906        -0.667   0
 3 upgraded1    Yes     0.647          0.424         0.870   0
 4 version20131 Yes     0.627          0.403         0.852   0
 5 income       No      0.601          0.469         0.734   0
 6 payroll1     Yes     0.434          0.170         0.698   0.001
 7 numords      No      0.415          0.290         0.540   0
 8 medhvalue    No     -0.298         -0.449        -0.147   0
 9 dollars      No      0.166          0.0502        0.282   0.005
10 bizflag1     Yes     0.164          0.0393        0.288   0.01
11 sincepurch   No      0.125         -0.0812        0.332   0.234
12 speedup      No      0.0748        -0.0273        0.177   0.151
13 sexUnknown   Yes     0.0390        -0.153         0.231   0.69
14 sexMale      Yes    -0.00983       -0.148         0.129   0.889
```

```
intuit.test <- intuit.test %>%
  mutate(pred_lr = predict(lr1, newdata=intuit.test, type="response"))
gainsplot(intuit.test$pred_lr, label.var=intuit.test$res)
```



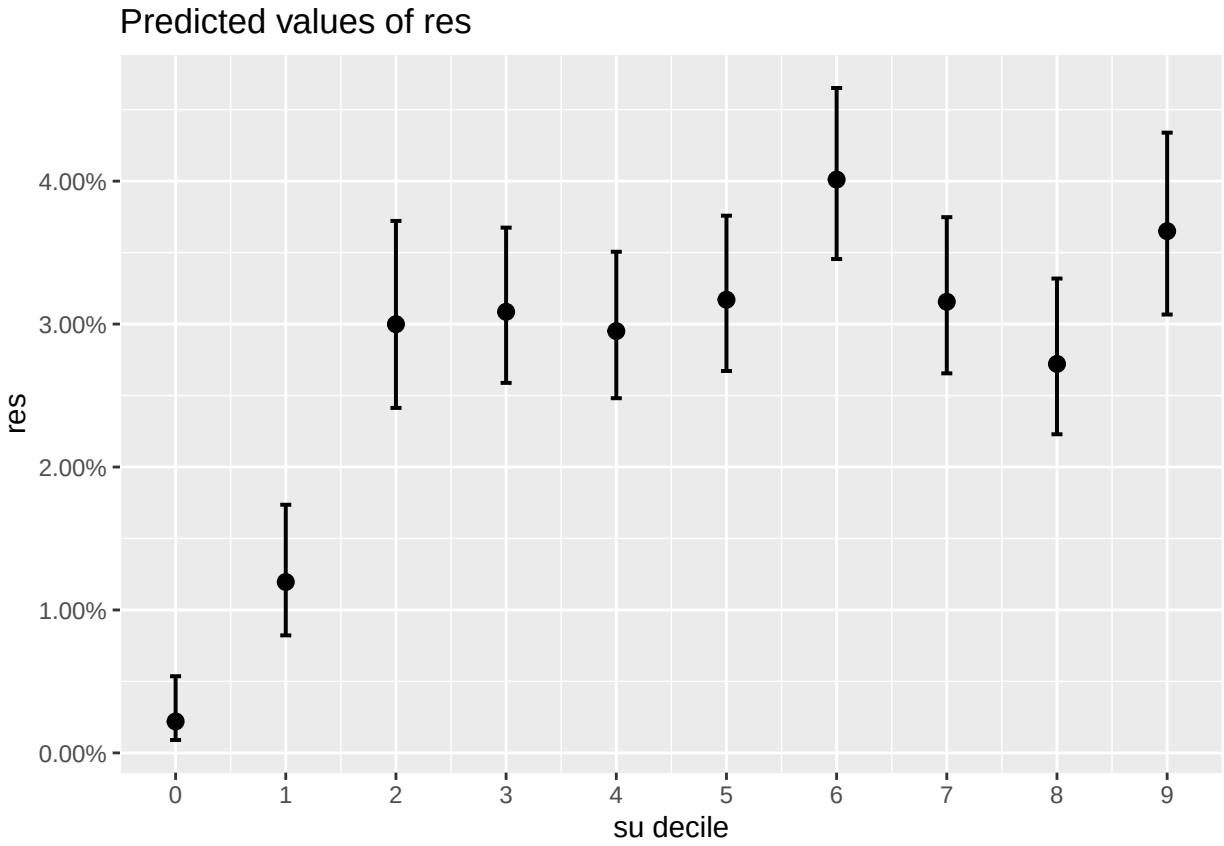

```
# A tibble: 1 x 2
  model                 auc
  <chr>               <dbl>
1 intuit.test$pred_lr 0.755
```

Recall Internet Upload Speed

The variable importance plot places `speedup` near the bottom of the list, but in the case debrief we noticed it had a highly nonlinear effect. We were able to visualize this by forming deciles of `speedup`, refitting the model, and using `plot_model`:

```
intuit <- intuit %>%
  mutate(su_decile = factor(ntile(speedup,10)-1))
intuit.train <- intuit %>% filter(training==1)
intuit.test <- intuit %>% filter(training==0)

fm2 <- as.formula(res ~ speeddown + su_decile + last + numords +
                  dollars + sincepurch + version2013 + upgraded +
                  payroll + bizflag + sex + income + medhvalue)
lr2 <- glm(fm2, data=intuit.train, family=binomial)
plot_model(lr2, type = "eff", terms = "su_decile")
```



We see a clear difference between the first two deciles and the other eight! We can get the decile cutoff points to figure out what value of upload speed creates this difference

```
intuit.train %>%
  group_by(su_decile) %>%
  summarize(min(speedup), max(speedup))
```

```
# A tibble: 10 x 3
   su_decile `min(speedup)` `max(speedup)`
   <fct>              <dbl>          <dbl>
 1 0                  0              0.300
 2 1                  0.300          1.1
 3 2                  1.1            1.8
 4 3                  1.8            2.7
 5 4                  2.7            3
 6 5                  3              4.30
 7 6                  4.30           7.10
 8 7                  7.10           8.70
 9 8                  8.70          11.5
10 9                 11.5           50
```

The difference occurs for `speedup` greater than or less than 1.1. Since the plot is flat after that, we can capture this effect with a single binary variable. Let's do that:

```
intuit <- intuit %>%
  mutate(su_cutoff = 1*(speedup > 1.1))
intuit.train <- intuit %>% filter(training==1)
intuit.test <- intuit %>% filter(training==0)
```

```
fm3 <- as.formula(res ~ speeddown + su_cutoff + last + numords +
                  dollars + sincepurch + version2013 + upgraded +
                  payroll + bizflag + sex + income + medhvalue)
lr3 <- glm(fm3, data=intuit.train, family=binomial)
```

Simulating an Action

What if we could increase our customers' upload speeds? In particular, what would happen if we could take customers with upload speeds lower than 1.1 and make them bigger than 1.1?

What is the probability on average that such customers buy?

```
predict(lr3,
        newdata = intuit.test %>% filter(su_cutoff==0),
        type = "response") %>%
  mean()
```

```
[1] 0.00614
```

What is the probability on average that they buy with higher upload speed?

```
predict(lr3,
        newdata = intuit.test %>% filter(su_cutoff==0) %>% mutate(su_cutoff=1),
        type = "response") %>%
  mean()
```

```
[1] 0.0223
```

Probability of purchase increases by a factor of 3.6!