Feature Selection and Data Life Cycle Management

Andrew J. Otis

May 16, 2022

Introduction

Here, we will explore feature selection, specifically, Forward Feature Construction

Q1

The raw data in this question is the "Pew Research Center's American Trends Panel" Wave 69 Field dates: June 16 – June 22, 2020 Topics: Coronavirus tracking, politics, 2020 Census data and questionnaire downloaded 3/4/2021 from https://www.pewresearch.org/politics/dataset/american-trends-panel-wave-69/

The codebook was downloaded 3/5/2021 from https://www.pewresearch.org/wp-content/uploads/2018/05/Codebook-and-instructions-for-working-with-ATP-data.pdf

The data set "dat.ind" consists of responses from self-identified "Independent" respondents who provided a response to "VOTEGEN_W69" and "F_IDEO", and provided a response other than "Refused" to "F_INCOME"

1.a.

The code below converts the responses to the "NATPROBS" variables to their numeric values, setting the "Refused" response to "NA". The variable "biden" is defined. Any case with an undefined value in any of the "NATPROBS" variables is dropped.

The code below also splits the data into training and validate sets, preserving proportion of "biden".

Logistic regression model of "biden" on "F_IDEO", "F_INCOME", and the variables with the prefix "NATPROBS" on the training data, dat.train.

Test the model hypothesis for this model using "hoslem.test" from the "ResourceSelection" package. What do you conclude about the use of logistic regression from this result?

```
load("dat_independents.RData")
dat.ind$biden<-dat.ind$VOTEGEN_W69=="Joe Biden, the Democrat"
dat.ind$income<-as.numeric(dat.ind$F_INCOME)
numeric.make<-function(x){
  x<-as.numeric(x)
  x[x==5]<-NA
   return(x)
}
dat.ind<-dat.ind%>%mutate(across(starts_with("NATPROB"),numeric.make))
```

```r
nam<-names(dat.ind)[str_detect(names(dat.ind),"NATPROB")]

dat.ind<-dplyr::select(dat.ind,c(QKEY,biden,F_IDEO,F_INCOME,income,
                           NATPROBS_a_W69:NATPROBS_j_W69))

dat.ind<-dat.ind[complete.cases(dat.ind),]


# Train-validate
set.seed(23456)
dat.train<-dat.ind%>%
  group_by(F_IDEO,biden)%>%
  slice_sample(prop=.7)%>%ungroup()

dat.valid<-filter(dat.ind,!QKEY %in% dat.train$QKEY)

fmla<-str_c("biden~F_IDEO+F_INCOME+",str_c(nam,collapse="+"))
m<-glm(fmla,data=dat.train,family="binomial")


summary(m)

##
## Call:
## glm(formula = fmla, family = "binomial", data = dat.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.79117  -0.54692  -0.07526   0.64258   2.62527
##
## Coefficients:
##                                         Estimate Std. Error z value Pr(>|z|
)
## (Intercept)                             -1.18856    0.96412  -1.233 0.21765
3
## F_IDEOConservative                       0.62148    0.83825   0.741 0.45845
0
## F_IDEOModerate                           1.53059    0.79689   1.921 0.05477
0 .
## F_IDEOLiberal                            2.03787    0.83971   2.427 0.01522
9 *
## F_IDEOVery liberal                       0.80303    0.88886   0.903 0.36629
4
## F_IDEORefused                            0.51781    1.09483   0.473 0.63624
6
## F_INCOME$10,000 to less than $20,000     1.41758    0.50890   2.786 0.00534
3 **
## F_INCOME$20,000 to less than $30,000     1.03566    0.52041   1.990 0.04658
2 *
```

```
## F_INCOME$30,000 to less than $40,000    0.78501    0.52004   1.510 0.13117
1
## F_INCOME$40,000 to less than $50,000    1.47061    0.53100   2.770 0.00561
4 **
## F_INCOME$50,000 to less than $75,000    1.49949    0.49894   3.005 0.00265
3 **
## F_INCOME$75,000 to less than $100,000   1.74960    0.50352   3.475 0.00051
1 ***
## F_INCOME$100,000 to less than $150,000  1.89661    0.49625   3.822 0.00013
2 ***
## F_INCOME$150,000 or more                2.45506    0.52064   4.715 2.41e-0
6 ***
## NATPROBS_a_W69                         -0.67513    0.15710  -4.297 1.73e-0
5 ***
## NATPROBS_b_W69                         -0.77062    0.17197  -4.481 7.42e-0
6 ***
## NATPROBS_c_W69                         -0.06597    0.14622  -0.451 0.65186
9
## NATPROBS_d_W69                         -0.05810    0.18770  -0.310 0.75692
2
## NATPROBS_e_W69                          0.15394    0.15437   0.997 0.31866
8
## NATPROBS_f_W69                         -0.11380    0.18867  -0.603 0.54638
7
## NATPROBS_g_W69                          0.78317    0.14018   5.587 2.31e-0
8 ***
## NATPROBS_h_W69                          0.06691    0.17397   0.385 0.70051
2
## NATPROBS_i_W69                         -1.03646    0.14965  -6.926 4.33e-1
2 ***
## NATPROBS_j_W69                          0.14874    0.16219   0.917 0.35909
6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1101.33  on 798  degrees of freedom
## Residual deviance:  632.55  on 775  degrees of freedom
## AIC: 680.55
##
## Number of Fisher Scoring iterations: 6

# Based on the results of the logistic regression, we can observe the followi
ng variables that have a significant effect on the outcome variable (i.e. p-v
alue < 0.05, the standard accepted error).
# F_IDEOLiberal
# F_INCOME$10,000 to less than $20,000,
# F_INCOME$20,000 to less than $30,000,
# F_INCOME$40,000 to less than $50,000,
```

```
# F_INCOME$50,000 to less than $75,000,
# F_INCOME$75,000 to less than $100,000,
# F_INCOME$100,000 to less than $150,000,
# F_INCOME$150,000 or more,
# NATPROBS_a_W69,
# NATPROBS_g_W69,
# NATPROBS_i_W69

# Each one-unit change in the following variables will INCREASE the log odds
for the outcome variable "biden". Specific quantities as part of the "estimat
e" column in the model summary.
# F_INCOME$10,000 to less than $20,000,
# F_INCOME$20,000 to less than $30,000
# F_INCOME$40,000 to less than $50,000,
# F_INCOME$50,000 to less than $75,000,
# F_INCOME$75,000 to less than $100,000,
# F_INCOME$100,000 to less than $150,000,
# F_INCOME$150,000 or more

# Each one-unit change in the following variables will DECREASE the log odds
for the outcome variable "biden". Specific quantities as part of the "estimat
e" column in the model summary.
# NATPROBS_a_W69,
# NATPROBS_g_W69,
# NATPROBS_i_W69

hoslem.test(dat.train$biden, fitted(m), g = 10)

##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  dat.train$biden, fitted(m)
## X-squared = 10.42, df = 8, p-value = 0.2367

# The resulting p-value = 0.2367 > 0.05, the standard accepted error. Therefo
re there is NOT ENOUGH evidence to suggest that the model "m" is poorly fitte
d. Thus interpretation/predictions of the model's results can be considered V
ALID.
```

1.b.

Compute and display the confusion matrix, the accuracy, the precision, F1 on the training data and the McFadden Pseudo R^2 for this model.

McFadden Pseudo R^2

```
PseudoR2(m)

##  McFadden
## 0.4256461
```

Confusion matrix

```
# Code relevant to our model test
prob<-predict(m,dat.train,type="response")
pred<-prob>=.5
print("Confusion Matrix")
```

```
## [1] "Confusion Matrix"
```

```
table(dat.train$biden,pred)
```

```
##        pred
##         FALSE TRUE
##   FALSE   357   78
##   TRUE     64  300
```

Accuracy

```
# Accuracy is the proportion correct.
fitted <- pred*1
accuracy <- mean(dat.train$biden==fitted)
print("Accuracy")
```

```
## [1] "Accuracy"
```

```
accuracy
```

```
## [1] 0.8222778
```

Recall

```
recall<-sum(dat.train$biden==1 & fitted==1)/sum(dat.train$biden==1)
print("Recall")
```

```
## [1] "Recall"
```

```
recall
```

```
## [1] 0.8241758
```

Precision & F1

```
# Precision is the proportion of true positives to positives.
precision<-sum(dat.train$biden==1 & fitted==1)/sum(fitted==1)
print("Precision")
```

```
## [1] "Precision"
```

```
precision
```

```
## [1] 0.7936508
```

```
# F1 Score = 2(Recall)(Precision)/ (Recall + Precision)
f1<-2*recall*precision/(precision+recall)
print("F1")
```

```
## [1] "F1"
```

```
f1
```

```
## [1] 0.8086253
```

1.c.

Compute and display the confusion matrix, the accuracy, the precision, and F1 when the model above is used to predict the validation outcome values.

Confusion matrix

```
# Code relevant to our model test
prob<-predict(m,dat.valid,type="response")
pred<-prob>=.5
print("Confusion Matrix")
```

```
## [1] "Confusion Matrix"
```

```
table(dat.valid$biden,pred)
```

```
##        pred
##         FALSE TRUE
##   FALSE   146   44
##   TRUE     34  126
```

Accuracy

```
# Accuracy is the proportion correct.
fitted <- pred*1
accuracy <- mean(dat.valid$biden==fitted)
print("Accuracy")
```

```
## [1] "Accuracy"
```

```
accuracy
```

```
## [1] 0.7771429
```

Recall

```
recall<-sum(dat.valid$biden==1 & fitted==1)/sum(dat.valid$biden==1)
print("Recall")
```

```
## [1] "Recall"
```

```
recall
```

```
## [1] 0.7875
```

Precision & F1

```
# Precision is the proportion of true positives to positives.
precision<-sum(dat.valid$biden==1 & fitted==1)/sum(fitted==1)
print("Precision")
```

```
## [1] "Precision"

precision

## [1] 0.7411765

# F1 Score = 2(Recall)(Precision)/ (Recall + Precision)
f1<-2*recall*precision/(precision+recall)
print("F1")

## [1] "F1"

f1

## [1] 0.7636364
```

1.d.

Fit the forward model by AIC on the training data and the full data. Is there evidence that the smaller data set has biased the forward fit toward a simpler model?

Full model

```
# Full model
full.model <- glm(biden ~., data = dat.train, family = binomial)
summary(full.model)

##
## Call:
## glm(formula = biden ~ ., family = binomial, data = dat.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.75709  -0.54648  -0.07685   0.63532   2.63162
##
## Coefficients: (1 not defined because of singularities)
##                                        Estimate Std. Error z value Pr(>|
z|)
## (Intercept)                          -9.591e-01  9.918e-01  -0.967 0.333
548
## QKEY                                 -1.433e-12  1.370e-12  -1.046 0.295
540
## F_IDEOConservative                    6.349e-01  8.422e-01   0.754 0.450
920
## F_IDEOModerate                        1.565e+00  8.016e-01   1.952 0.050
922
## F_IDEOLiberal                         2.071e+00  8.442e-01   2.453 0.014
173
## F_IDEOVery liberal                    8.235e-01  8.934e-01   0.922 0.356
686
## F_IDEORefused                         5.975e-01  1.100e+00   0.543 0.587
197
```

```
## F_INCOME$10,000 to less than $20,000    1.390e+00  5.096e-01   2.727 0.006
386
## F_INCOME$20,000 to less than $30,000    9.993e-01  5.213e-01   1.917 0.055
251
## F_INCOME$30,000 to less than $40,000    7.630e-01  5.211e-01   1.464 0.143
162
## F_INCOME$40,000 to less than $50,000    1.412e+00  5.345e-01   2.642 0.008
239
## F_INCOME$50,000 to less than $75,000    1.471e+00  5.006e-01   2.938 0.003
302
## F_INCOME$75,000 to less than $100,000   1.712e+00  5.058e-01   3.384 0.000
714
## F_INCOME$100,000 to less than $150,000  1.855e+00  4.991e-01   3.716 0.000
202
## F_INCOME$150,000 or more                2.384e+00  5.264e-01   4.530 5.91e
-06
## income                                         NA         NA      NA
NA
## NATPROBS_a_W69                         -6.886e-01  1.582e-01  -4.353 1.34e
-05
## NATPROBS_b_W69                         -7.638e-01  1.726e-01  -4.424 9.67e
-06
## NATPROBS_c_W69                         -5.338e-02  1.467e-01  -0.364 0.716
002
## NATPROBS_d_W69                         -5.423e-02  1.881e-01  -0.288 0.773
086
## NATPROBS_e_W69                          1.658e-01  1.548e-01   1.071 0.284
194
## NATPROBS_f_W69                         -1.200e-01  1.885e-01  -0.637 0.524
376
## NATPROBS_g_W69                          7.814e-01  1.404e-01   5.566 2.61e
-08
## NATPROBS_h_W69                          7.113e-02  1.746e-01   0.407 0.683
696
## NATPROBS_i_W69                         -1.024e+00  1.501e-01  -6.821 9.03e
-12
## NATPROBS_j_W69                          1.287e-01  1.635e-01   0.787 0.431
381
##
## (Intercept)
## QKEY
## F_IDEOConservative
## F_IDEOModerate                         .
## F_IDEOLiberal                          *
## F_IDEOVery liberal
## F_IDEORefused
## F_INCOME$10,000 to less than $20,000   **
## F_INCOME$20,000 to less than $30,000   .
## F_INCOME$30,000 to less than $40,000
## F_INCOME$40,000 to less than $50,000   **
```

```
## F_INCOME$50,000 to less than $75,000   **
## F_INCOME$75,000 to less than $100,000  ***
## F_INCOME$100,000 to less than $150,000 ***
## F_INCOME$150,000 or more               ***
## income
## NATPROBS_a_W69                         ***
## NATPROBS_b_W69                         ***
## NATPROBS_c_W69
## NATPROBS_d_W69
## NATPROBS_e_W69
## NATPROBS_f_W69
## NATPROBS_g_W69                         ***
## NATPROBS_h_W69
## NATPROBS_i_W69                         ***
## NATPROBS_j_W69
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1101.33  on 798  degrees of freedom
## Residual deviance:  631.45  on 774  degrees of freedom
## AIC: 681.45
##
## Number of Fisher Scoring iterations: 6
```

Forward Model

```
nothing.model <- glm(biden ~ 1, data = dat.train, family = binomial)

# fit the forward model by AIC on the training data and the full data
forward.model <- step(nothing.model, scope = list(lower=formula(nothing.model
), upper = formula(full.model)) ,direction = "forward")

## Start:  AIC=1103.33
## biden ~ 1
##
##                  Df Deviance     AIC
## + NATPROBS_i_W69  1   849.39  853.39
## + NATPROBS_a_W69  1   929.64  933.64
## + F_IDEO          5   934.95  946.95
## + NATPROBS_g_W69  1   949.13  953.13
## + NATPROBS_b_W69  1   990.07  994.07
## + NATPROBS_f_W69  1  1033.89 1037.89
## + NATPROBS_j_W69  1  1054.38 1058.38
## + income          1  1072.48 1076.48
## + F_INCOME        8  1061.94 1079.94
## + NATPROBS_e_W69  1  1079.92 1083.92
## + NATPROBS_d_W69  1  1081.53 1085.53
## + NATPROBS_h_W69  1  1082.36 1086.36
## + QKEY            1  1097.98 1101.98
```

```
## <none>                1101.33 1103.33
## + NATPROBS_c_W69  1  1100.51 1104.51
##
## Step:  AIC=853.39
## biden ~ NATPROBS_i_W69
##
##                   Df Deviance    AIC
## + NATPROBS_g_W69  1   753.06 759.06
## + income          1   803.35 809.35
## + F_IDEO          5   795.85 809.85
## + NATPROBS_a_W69  1   805.75 811.75
## + NATPROBS_j_W69  1   807.14 813.14
## + F_INCOME        8   797.08 817.08
## + NATPROBS_e_W69  1   820.32 826.32
## + NATPROBS_b_W69  1   822.44 828.44
## + NATPROBS_c_W69  1   845.51 851.51
## + NATPROBS_f_W69  1   846.18 852.18
## + QKEY            1   847.24 853.24
## + NATPROBS_d_W69  1   847.29 853.29
## <none>                849.39 853.39
## + NATPROBS_h_W69  1   849.09 855.09
##
## Step:  AIC=759.06
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69
##
##                   Df Deviance    AIC
## + NATPROBS_b_W69  1   721.09 729.09
## + income          1   724.32 732.32
## + NATPROBS_a_W69  1   724.37 732.37
## + F_INCOME        8   714.05 736.05
## + F_IDEO          5   727.90 743.90
## + NATPROBS_j_W69  1   748.49 756.49
## + NATPROBS_f_W69  1   749.16 757.16
## + QKEY            1   750.02 758.02
## + NATPROBS_d_W69  1   750.44 758.44
## <none>                753.06 759.06
## + NATPROBS_e_W69  1   751.49 759.49
## + NATPROBS_h_W69  1   752.62 760.62
## + NATPROBS_c_W69  1   752.84 760.84
##
## Step:  AIC=729.09
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69
##
##                   Df Deviance    AIC
## + income          1   691.08 701.08
## + F_INCOME        8   682.79 706.79
## + F_IDEO          5   691.97 709.97
## + NATPROBS_a_W69  1   705.00 715.00
## + NATPROBS_j_W69  1   715.41 725.41
## + NATPROBS_e_W69  1   718.04 728.04
```

```
## + QKEY            1   718.43 728.43
## + NATPROBS_d_W69  1   718.63 728.63
## + NATPROBS_f_W69  1   718.76 728.76
## <none>                721.09 729.09
## + NATPROBS_c_W69  1   720.77 730.77
## + NATPROBS_h_W69  1   720.98 730.98
##
## Step:  AIC=701.08
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income
##
##                  Df Deviance    AIC
## + NATPROBS_a_W69  1   668.71 680.71
## + F_IDEO          5   665.29 685.29
## + NATPROBS_f_W69  1   688.01 700.01
## <none>                691.08 701.08
## + NATPROBS_d_W69  1   689.90 701.90
## + NATPROBS_e_W69  1   689.93 701.93
## + NATPROBS_j_W69  1   690.01 702.01
## + QKEY            1   690.43 702.43
## + NATPROBS_h_W69  1   690.73 702.73
## + NATPROBS_c_W69  1   691.08 703.08
## + F_INCOME        7   682.79 706.79
##
## Step:  AIC=680.71
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
##     NATPROBS_a_W69
##
##                  Df Deviance    AIC
## + F_IDEO          5   645.20 667.20
## + NATPROBS_e_W69  1   665.70 679.70
## + NATPROBS_j_W69  1   666.69 680.69
## <none>                668.71 680.71
## + QKEY            1   667.59 681.59
## + NATPROBS_f_W69  1   667.98 681.98
## + NATPROBS_c_W69  1   668.63 682.63
## + NATPROBS_h_W69  1   668.67 682.67
## + NATPROBS_d_W69  1   668.69 682.69
## + F_INCOME        7   659.64 685.64
##
## Step:  AIC=667.2
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
##     NATPROBS_a_W69 + F_IDEO
##
##                  Df Deviance    AIC
## + NATPROBS_e_W69  1   642.85 666.85
## <none>                645.20 667.20
## + QKEY            1   643.61 667.61
## + NATPROBS_j_W69  1   643.77 667.77
## + NATPROBS_f_W69  1   644.96 668.96
## + NATPROBS_h_W69  1   645.05 669.05
```

```
## + NATPROBS_d_W69  1   645.19 669.19
## + NATPROBS_c_W69  1   645.19 669.19
## + F_INCOME        7   635.97 671.97
##
## Step:  AIC=666.85
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
##     NATPROBS_a_W69 + F_IDEO + NATPROBS_e_W69
##
##                  Df Deviance    AIC
## <none>                642.85 666.85
## + QKEY            1   641.05 667.05
## + NATPROBS_j_W69  1   642.39 668.39
## + NATPROBS_f_W69  1   642.62 668.62
## + NATPROBS_c_W69  1   642.69 668.69
## + NATPROBS_h_W69  1   642.72 668.72
## + NATPROBS_d_W69  1   642.85 668.85
## + F_INCOME        7   634.21 672.21
```

```
summary(forward.model)
```

```
##
## Call:
## glm(formula = biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 +
##     income + NATPROBS_a_W69 + F_IDEO + NATPROBS_e_W69, family = binomial,
##     data = dat.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.76005  -0.56815  -0.08165   0.66277   2.69338
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.89368    0.85899  -1.040   0.2982
## NATPROBS_i_W69     -1.05626    0.14123  -7.479 7.48e-14 ***
## NATPROBS_g_W69      0.76949    0.13330   5.773 7.80e-09 ***
## NATPROBS_b_W69     -0.76812    0.16600  -4.627 3.71e-06 ***
## income              0.21629    0.04108   5.265 1.40e-07 ***
## NATPROBS_a_W69     -0.66882    0.14871  -4.497 6.88e-06 ***
## F_IDEOConservative  0.59028    0.82662   0.714   0.4752
## F_IDEOModerate      1.52210    0.78768   1.932   0.0533 .
## F_IDEOLiberal       1.96551    0.82647   2.378   0.0174 *
## F_IDEOVery liberal  0.76319    0.87268   0.875   0.3818
## F_IDEORefused       0.48855    1.07390   0.455   0.6492
## NATPROBS_e_W69      0.21304    0.13931   1.529   0.1262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1101.33  on 798  degrees of freedom
```

```
## Residual deviance:  642.85  on 787  degrees of freedom
## AIC: 666.85
##
## Number of Fisher Scoring iterations: 6

# I've interpreted full model as the outcome variable and ALL other independe
nt variables and the simpler model as the model with outcome variable and SEL
ECT independent variables


formula(full.model)

## biden ~ QKEY + F_IDEO + F_INCOME + income + NATPROBS_a_W69 +
##     NATPROBS_b_W69 + NATPROBS_c_W69 + NATPROBS_d_W69 + NATPROBS_e_W69 +
##     NATPROBS_f_W69 + NATPROBS_g_W69 + NATPROBS_h_W69 + NATPROBS_i_W69 +
##     NATPROBS_j_W69

print("Full Model AIC = 681.45")

## [1] "Full Model AIC = 681.45"

formula(forward.model)

## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
##     NATPROBS_a_W69 + F_IDEO + NATPROBS_e_W69

print("Forward AIC = 666.85")

## [1] "Forward AIC = 666.85"

# Since Lower AIC values are indicative of a better model fit, and the simple
r model being the model with less independent variables when compared to the
full model. We can conclude that there IS EVIDENCE the forward model IS INDEE
D biased toward a simpler model.
```

1.e.

The goal of this question is to compare the model in the forward model construction sequence selected on the basis of minimizing the AIC to the model in the forward model construction sequence selected minimize the deviance on the validation set.

The code below generates the full sequence of forward models by setting the argument “k” equal to 0, thereby removing the penalty for additional variables.

```
nam<-names(dat.ind)[str_detect(names(dat.ind),"NATPROB")]
fmla<-str_c("biden~F_IDEO+F_INCOME+",str_c(nam,collapse="+"))
m.1<-glm(biden~1,data=dat.train,family="binomial")
m.forward.k0<-step(m.1,scope=fmla,direction="forward",k=0,trace=0)
# The sequence in which variables are added
vars.add<-m.forward.k0$anova[,1]
vars.add<-str_replace(vars.add,"\\+ ","")
vars.add[1]<-1 # Intercept only
```

The code below makes a vector of the formulas corresponding to the variables used at each stage in the forward model construction.

```
# function to collect the first "i" variables added during
# forward selection and
# create a formula for "chd" in terms of the remaining
# variables.

fmla.add.fnc<-function(i,vars.add){
  vars.in<-vars.add[1:i]
  return(as.formula(str_c("biden~",str_c(vars.in,collapse="+"))))
}

# Apply "fmla.add.fnc" to each value of "i". This
# gives the formulas for the models generated by forward
# selection.

fmlas<-apply(matrix(1:length(vars.add),ncol=1),1,
   fmla.add.fnc,vars.add=vars.add)
```

Construct the models on the training data corresponding to this list of formulas and store them in a list.

```
# Use an anonymous function, defined in place.
models<-
  lapply(fmlas,function(x){glm(x,data=dat.train,family="binomial")})
```

Implement a deviance function to calculate the deviance of a model on a validation set.

```
valid.dev<-function(m.pred, dat.this){
  pred.m<-predict(m.pred,dat.this, type="response")
-2*sum(dat.this$biden*log(pred.m)+(1-dat.this$biden)*log(1-pred.m))
}
```

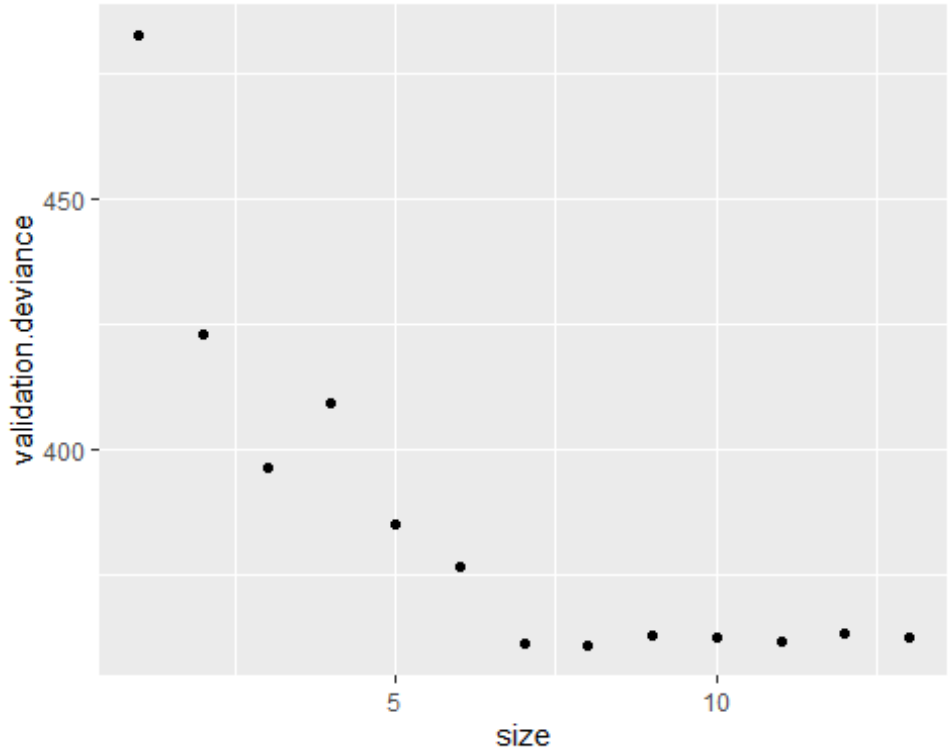
Calculate the deviance for on the validation set when probabilities are predicted according to the coefficients for each model in the list.

```
dev.valid<-sapply(models,valid.dev,dat.this=dat.valid)
```

Visualize the results.

```
devs<-data.frame(size=1:length(dev.valid),
validation.deviance=dev.valid)

ggplot(data=devs, aes(x=size,y=validation.deviance))+geom_point()
```



Finally, is the model selected by validation deviance the same as the one selected by AIC?

```
# forward Model
# biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
  #       NATPROBS_a_W69 + F_IDEO + NATPROBS_e_W69

formula(forward.model)

## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
##     NATPROBS_a_W69 + F_IDEO + NATPROBS_e_W69

print("Model Selection based on validation deviance")

## [1] "Model Selection based on validation deviance"

dev.valid

##  [1] 482.6319 422.9802 396.2518 409.0398 384.8595 376.6006 361.1710 360.72
68
##  [9] 362.5105 362.1684 361.5152 363.0050 362.4369

min.dev <- which(dev.valid==min(dev.valid))
names(models[[min.dev]]$coefficients)

##  [1] "(Intercept)"
##  [2] "NATPROBS_i_W69"
##  [3] "NATPROBS_g_W69"
##  [4] "F_INCOME$10,000 to less than $20,000"
```

```
##  [5] "F_INCOME$20,000 to less than $30,000"
##  [6] "F_INCOME$30,000 to less than $40,000"
##  [7] "F_INCOME$40,000 to less than $50,000"
##  [8] "F_INCOME$50,000 to less than $75,000"
##  [9] "F_INCOME$75,000 to less than $100,000"
## [10] "F_INCOME$100,000 to less than $150,000"
## [11] "F_INCOME$150,000 or more"
## [12] "NATPROBS_a_W69"
## [13] "F_IDEOConservative"
## [14] "F_IDEOModerate"
## [15] "F_IDEOLiberal"
## [16] "F_IDEOVery liberal"
## [17] "F_IDEORefused"
## [18] "NATPROBS_b_W69"
## [19] "NATPROBS_j_W69"
```

```
# Validation Deviance
# biden ~  NATPROBS_i_W69 +NATPROBS_g_W69 + F_INCOME$10,000 to less than $20,
000 + F_INCOME$20,000 to less than $30,000 + F_INCOME$30,000 to less than $40
,000 + F_INCOME$40,000 to less than $50,000 + "F_INCOME$50,000 to less than $
75,000"

# Notice, validation deviance is at its lowest (i.e. best model) by the 8th v
ariable.

print("Model based of forward AIC")
```

```
## [1] "Model based of forward AIC"
```

```
formula(forward.model)
```

```
## biden ~ NATPROBS_i_W69 + NATPROBS_g_W69 + NATPROBS_b_W69 + income +
##     NATPROBS_a_W69 + F_IDEO + NATPROBS_e_W69
```

```
print("Forward AIC = 666.85")
```

```
## [1] "Forward AIC = 666.85"
```

```
print("Model based on validation deviance")
```

```
## [1] "Model based on validation deviance"
```

```
print("biden ~  NATPROBS_i_W69+ NATPROBS_g_W69+ F_INCOME$10,000 to less than
$20,000+ F_INCOME$20,000 to less than $30,000+ F_INCOME$30,000 to less than $
40,000+ F_INCOME$40,000 to less than $50,000")
```

```
## [1] "biden ~  NATPROBS_i_W69+ NATPROBS_g_W69+ F_INCOME$10,000 to less than
$20,000+ F_INCOME$20,000 to less than $30,000+ F_INCOME$30,000 to less than $
40,000+ F_INCOME$40,000 to less than $50,000"
```

```
print("Forward validation deviance = 360.7268")
```

```
## [1] "Forward validation deviance = 360.7268"
```

```
# Based on these results, we can conclude that the model selected based on AI
C is different than the model selected based on validation deviance.
```

1.f.

Fit the forward model selected by AIC, the forward model selected to minimize the deviance on the validation set, and the full model, all on the full data set. Using "lrtest" from the "lmtest" library, which models improve significantly on the smaller model(s)? Can you relate this to the validation deviances plotted above?

```
lrtest(models[[min.dev]], m)
```

```
## Likelihood ratio test
##
## Model 1: biden ~ 1 + NATPROBS_i_W69 + NATPROBS_g_W69 + F_INCOME + NATPROBS
_a_W69 +
##     F_IDEO + NATPROBS_b_W69 + NATPROBS_j_W69
## Model 2: biden ~ F_IDEO + F_INCOME + NATPROBS_a_W69 + NATPROBS_b_W69 +
##     NATPROBS_c_W69 + NATPROBS_d_W69 + NATPROBS_e_W69 + NATPROBS_f_W69 +
##     NATPROBS_g_W69 + NATPROBS_h_W69 + NATPROBS_i_W69 + NATPROBS_j_W69
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  19 -317.08
## 2  24 -316.28  5 1.6088     0.9002
```

```
print("lrtest p-value = 0.9002")
```

```
## [1] "lrtest p-value = 0.9002"
```

```
# The resulting p-value = 0.9002 > 0.05, the standard accepted error.

# Therefore, we can conclude that the model for forward selection based on va
lidation deviances is a significant improvement over the simpler model from t
he forward selection based on AIC. This is supported by the model based on va
lidation deviances having a smaller AIC value (i.e. indication of better mode
l fit).
```

Q2

In logistic regression, if a categorical predictor has a category such that all of the values of the outcome variable are the same for that category, this is an example of *Quasi complete separation*. The questions below explore consequences of quasi complete separation.

2.a.

Note that for the model below, the category "x" is not a statistically significant predictor of "y", despite the fact that for all observations with "x" equal to "a", the value of "y" is 1, while only 30 out of 50 of the values of "y" are 1 for "x" equal to "b". What values do the predicted probabilities take on? Plot the predicted probabilities of "y" and the observed values of "y" in the model below using the x-axis to represent the value of "ind" and the y-axis to represent both the value and the fitted probability of the outcome at the corresponding

index. Do the fitted probabilities appear to correspond well to the probabilities in each category in the data?

```
ind<-1:100
x<-rep(c("a=0","b=1"),each=50)
y<-c(rep(1,50),rep(1,30),rep(0,20))
m.q<-glm(y~x,family="binomial")
print("Predicted Probabilities")
```

```
## [1] "Predicted Probabilities"
```

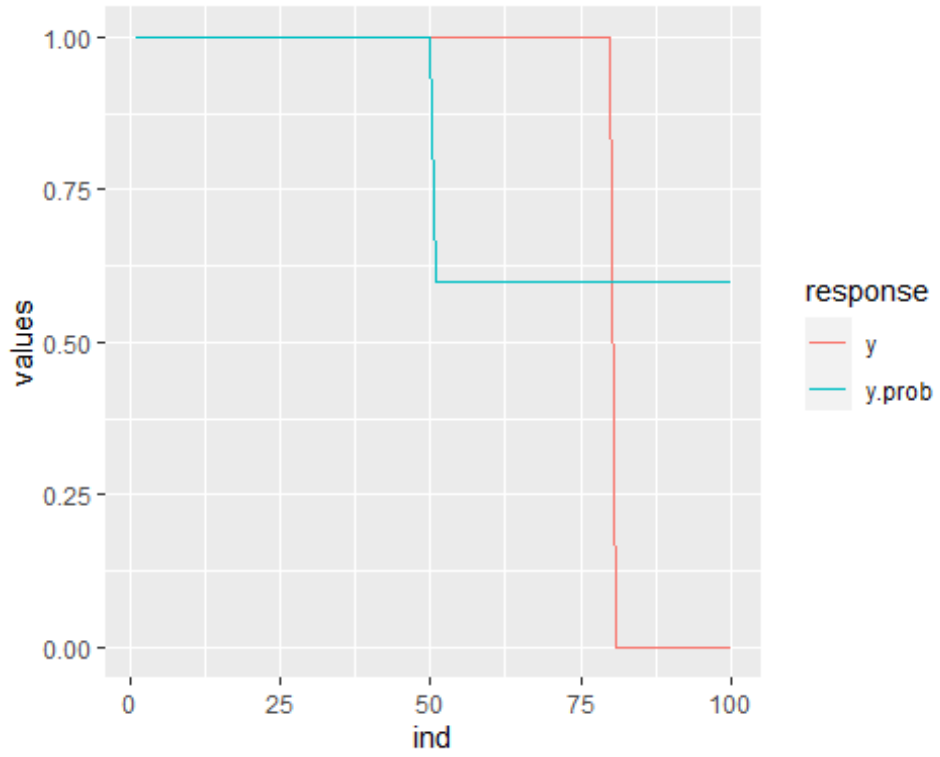
```
unique(predict(m.q,type="response"))
```

```
## [1] 1.0 0.6
```

```
summary(m.q)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.35373   0.00008   0.00008   1.01077   1.01077
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    19.57    1520.85   0.013     0.99
## xb=1          -19.16    1520.85  -0.013     0.99
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 100.080  on 99  degrees of freedom
## Residual deviance:  67.301  on 98  degrees of freedom
## AIC: 71.301
##
## Number of Fisher Scoring iterations: 18
```

```
dat.q<-data.frame(ind=ind,x=x,y=y,y.prob=predict(m.q,type="response"))
dat.q %>%
pivot_longer(!c(ind, x), names_to = "response", values_to = "values") %>%
ggplot(aes(x=ind, y= values, group=response))+
geom_line(aes(color=response))
```




```
table(x,y)
```

```
##      y
## x      0  1
##   a=0  0 50
##   b=1 20 30
```

```
# With the parameter estimate for xb being really large (>15), we suspect tha
t there is a problem of complete or quasi-complete separation.

# The standard errors for the parameter estimates are way too large. This usu
ally indicates a convergence issue or some degree of data separation.

# Fitted probabilities fit the probabilities in each categorical data fairly
well until ind = 50. At this point the probability lines deviate from one ano
ther.
```

2.b.

Repeat the plotting for the data below, in which one of the observations in category "a" has been changed to 0. Is the category "x" is a statistically significant predictor of "y"? Does quasicomplete separation present a challenge to interpretation of significance of predictors?

```
x<-rep(c("a=0","b=1"),each=50)
y<-c(rep(1,49),0,rep(1,30),rep(0,20))
```

```r
m.ok<-glm(y~x,family="binomial")
print("Predicted Probabilities")
```

```
## [1] "Predicted Probabilities"
```

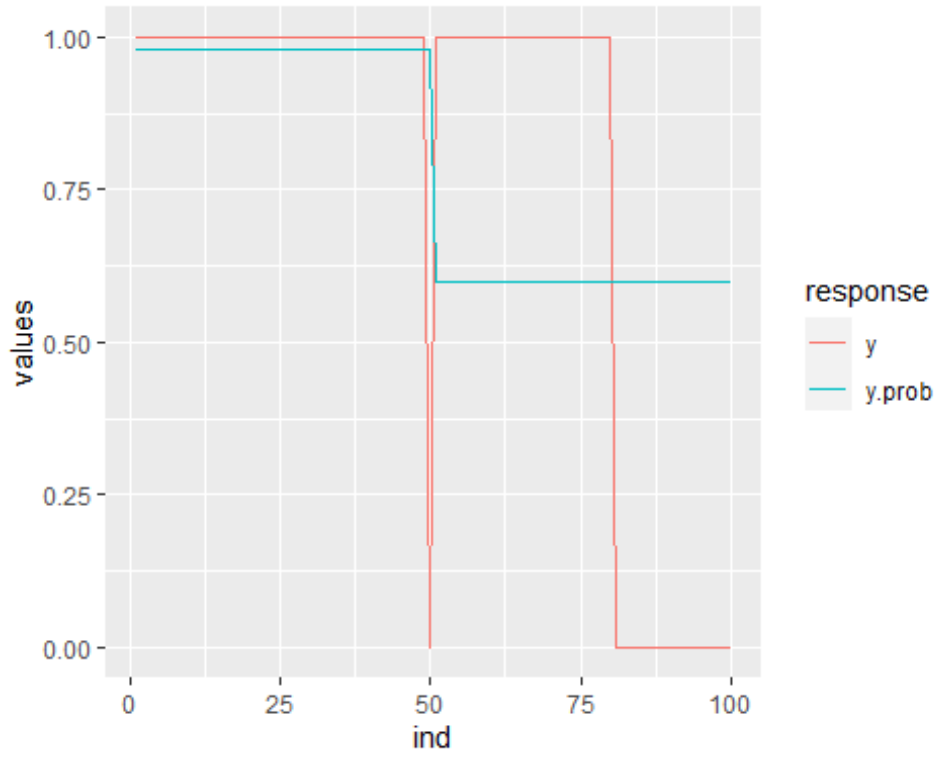
```r
unique(predict(m.ok,type="response"))
```

```
## [1] 0.98 0.60
```

```r
summary(m.ok)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.797   0.201   0.201   1.011   1.011
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.892      1.010   3.853 0.000117 ***
## xb=1          -3.486      1.051  -3.319 0.000905 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 102.791  on 99  degrees of freedom
## Residual deviance:  77.105  on 98  degrees of freedom
## AIC: 81.105
##
## Number of Fisher Scoring iterations: 6
```

```r
dat.q<-data.frame(ind=ind,x=x,y=y,y.prob=predict(m.ok,type="response"))
dat.q %>%
pivot_longer(!c(ind, x), names_to = "response", values_to = "values") %>%
ggplot(aes(x=ind, y= values, group=response))+
geom_line(aes(color=response))
```




```
table(x,y)
```

```
##      y
## x      0  1
##   a=0  1 49
##   b=1 20 30
```

```
# Notice that the predictor "xb=1" is a statistically significant predictor o
f y, supported by p-values < 0.05, the standard accepted error.

# Additionally, parameter estimate, and standard error are much lower, free o
f the separation noticed in the previous part of the problem.

# These drastic differences from a predictor variable going from insignifican
t to significant demonstrates the degree to which the results are effected by
quasi complete separation. Therefore, quasi complete separation DO PRESENT a
challenge to interpretation of significance of predictors.
```