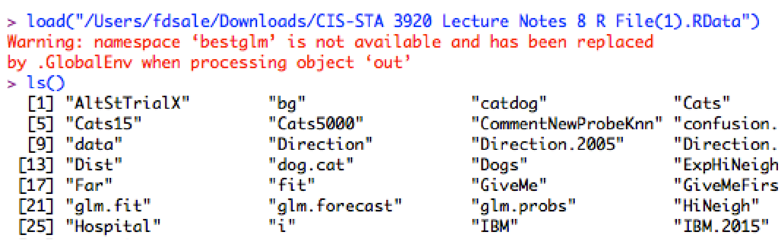
Marcus Crowder
DataMining 3920
Professor Lawrence Tatum
LN-7

7.1. Load the Hospital Beds Data file into R. Name the input “Hospital”. Select any one of the social-welfare variables as your x-variable, except Beds because I have already used it!

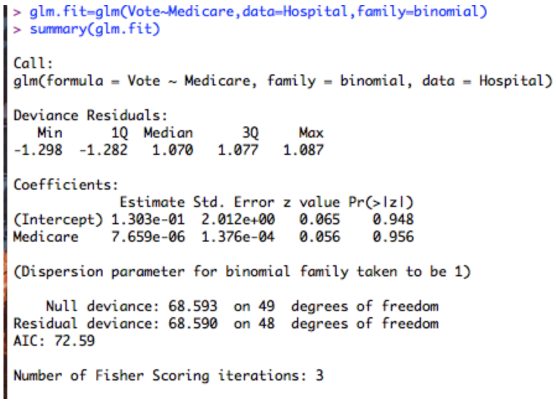
A. For this problem I slightly cheated because I saw the Hospital Data was already loaded into R.Data from week 8’s lab.



```
> load("/Users/fdsale/Downloads/CIS-STA 3920 Lecture Notes 8 R File(1).RData")
Warning: namespace ‘bestglm’ is not available and has been replaced
by .GlobalEnv when processing object ‘out’
> ls()
 [1] "AltStTrialX"        "bg"                 "catdog"             "Cats"
 [5] "Cats15"             "Cats5000"           "CommentNewProbeKnn" "confusion.
 [9] "data"               "Direction"          "Direction.2005"     "Direction.
[13] "Dist"               "dog.cat"            "Dogs"               "ExpHiNeigh
[17] "Far"                "fit"                "GiveMe"             "GiveMeFirs
[21] "glm.fit"            "glm.forecast"       "glm.probs"          "HiNeigh"
[25] "Hospital"           "i"                  "IBM"                "IBM.2015"
```

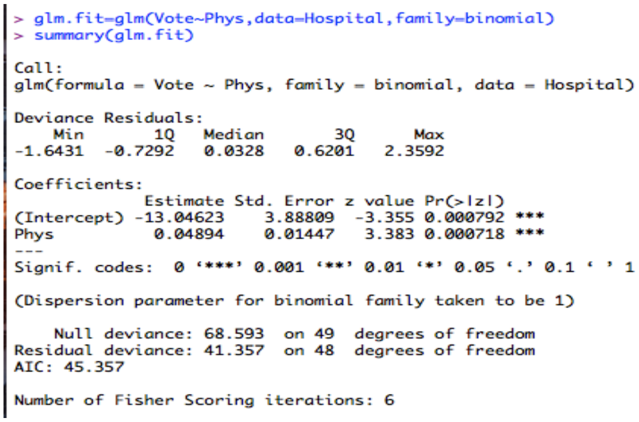
a. This is my first example using logistic Regression on Hospital Data. For this example I regressed Vote on Medicare but the resulting P value was too high for my liking at .956 the results can be tossed out when using a .95% confidence Interval. So instead I regressed Vote on Phys. The intercept is really low and positive but the P value appears to be very low at .000718 with *** which is signifies really low values in R.

Logistic Regression with Vote as Y variable and Medicare as X variable. As you can see there appear to be very high P values.



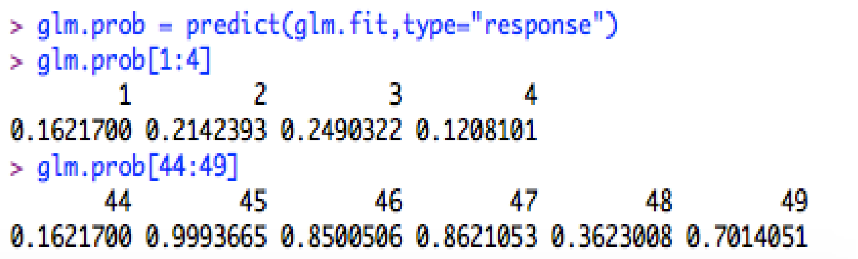
```
> glm.fit=glm(Vote~Medicare,data=Hospital,family=binomial)
> summary(glm.fit)

Call:
glm(formula = Vote ~ Medicare, family = binomial, data = Hospital)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.298  -1.282   1.070   1.077   1.087

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.303e-01  2.012e+00   0.065    0.948
Medicare    7.659e-06  1.376e-04   0.056    0.956

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 68.593  on 49  degrees of freedom
Residual deviance: 68.590  on 48  degrees of freedom
AIC: 72.59

Number of Fisher Scoring iterations: 3
```

Logistic Regression with Vote as Y variable and Phys as X variable. As you can see the P-value is a lot lower with a positive slope.



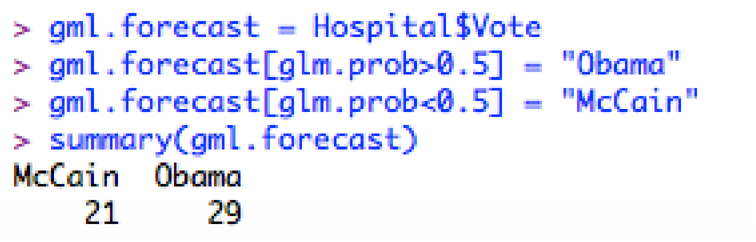
```
> glm.fit=glm(Vote~Phys,data=Hospital,family=binomial)
> summary(glm.fit)

Call:
glm(formula = Vote ~ Phys, family = binomial, data = Hospital)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6431  -0.7292   0.0328   0.6201   2.3592

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -13.04623    3.88809  -3.355 0.000792 ***
Phys          0.04894    0.01447   3.383 0.000718 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 68.593  on 49  degrees of freedom
Residual deviance: 41.357  on 48  degrees of freedom
AIC: 45.357

Number of Fisher Scoring iterations: 6
```

b. I named the vector of estimates as glm.prob and I used the predict function with the glm.fit Linear Model as the matrix input with type=”response”



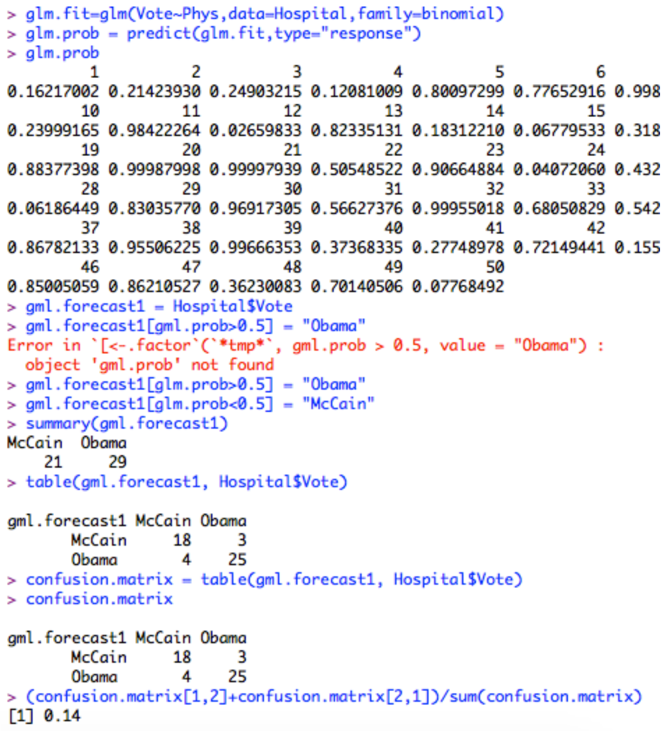
```
> glm.prob = predict(glm.fit,type="response")
> glm.prob[1:4]
        1         2         3         4 
0.1621700 0.2142393 0.2490322 0.1208101 
> glm.prob[44:49]
       44        45        46        47        48        49 
0.1621700 0.9993665 0.8500506 0.8621053 0.3623008 0.7014051 
```

d. To create a vector of Vote forecasts I placed all the hospital$Vote values into gml.forecast then used an R trick to make all values less than .05 to be equal to McCain and all other Values above equal to Obama The final rates became McCain 21 and Obama 29.



```
> gml.forecast = Hospital$Vote
> gml.forecast[glm.prob>0.5] = "Obama"
> gml.forecast[glm.prob<0.5] = "McCain"
> summary(gml.forecast)
McCain  Obama 
    21     29 
```

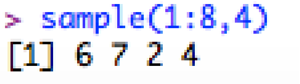
e. For this problem I used the Vote Y variable and Phys X varianle to produce glm.fit. After that I used the predict function on glm.fit to create glm.prob. Finally I calculated the ones and 0s of the gml.forecast1 to decide if the index would contain “Obama” or “McCain”. I produced a confusion matrix out of the table values of gml.forecast1 and Hospital$Vote my error rate ended up being 0.14. In the end creating a regression model based on Vote~Phys had a lower error rate than Vote~Medicaid+Beds.



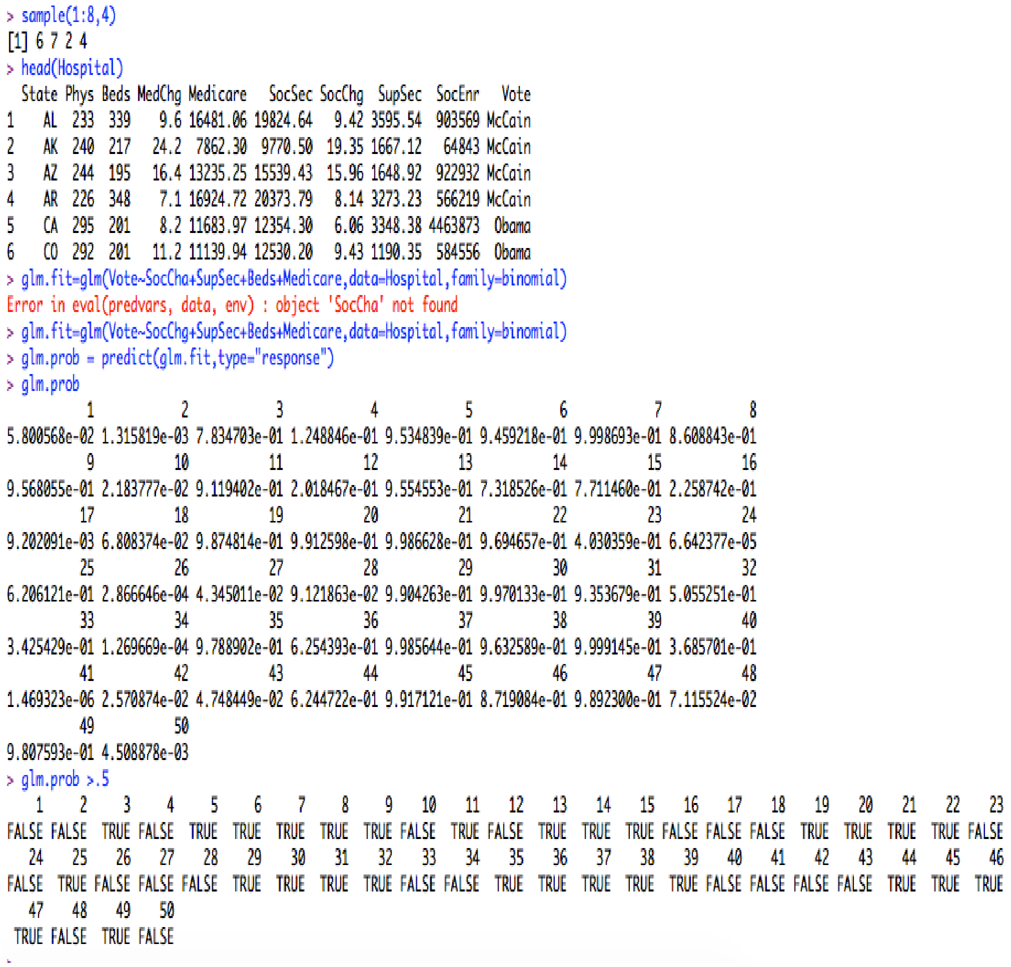
```
> glm.fit=glm(Vote~Phys,data=Hospital,family=binomial)
> glm.prob = predict(glm.fit,type="response")
> glm.prob
         1          2          3          4          5          6 
0.16217002 0.21423930 0.24903215 0.12081009 0.80097299 0.77652916 0.998
        10         11         12         13         14         15 
0.23999165 0.98422264 0.02659833 0.82335131 0.18312210 0.06779533 0.318
        19         20         21         22         23         24 
0.88377398 0.99987998 0.99997939 0.50548522 0.90664884 0.04072060 0.432
        28         29         30         31         32         33 
0.06186449 0.83035770 0.96917305 0.56627376 0.99955018 0.68050829 0.542
        37         38         39         40         41         42 
0.86782133 0.95506225 0.99666353 0.37368335 0.27748978 0.72149441 0.155
        46         47         48         49         50 
0.85005059 0.86210527 0.36230083 0.70140506 0.07768492 
> gml.forecast1 = Hospital$Vote
> gml.forecast1[gml.prob>0.5] = "Obama"
Error in `[<-.factor`(`*tmp*`, gml.prob > 0.5, value = "Obama") : 
  object 'gml.prob' not found
> gml.forecast1[glm.prob>0.5] = "Obama"
> gml.forecast1[glm.prob<0.5] = "McCain"
> summary(gml.forecast1)
McCain  Obama 
    21     29 
> table(gml.forecast1, Hospital$Vote)
             
gml.forecast1 McCain Obama
       McCain     18     3
       Obama       4    25
> confusion.matrix = table(gml.forecast1, Hospital$Vote)
> confusion.matrix
             
gml.forecast1 McCain Obama
       McCain     18     3
       Obama       4    25
> (confusion.matrix[1,2]+confusion.matrix[2,1])/sum(confusion.matrix)
[1] 0.14
```

7.2

a. For this problem I will be using the 6th ,7th , 2nd and 4th X Variables.



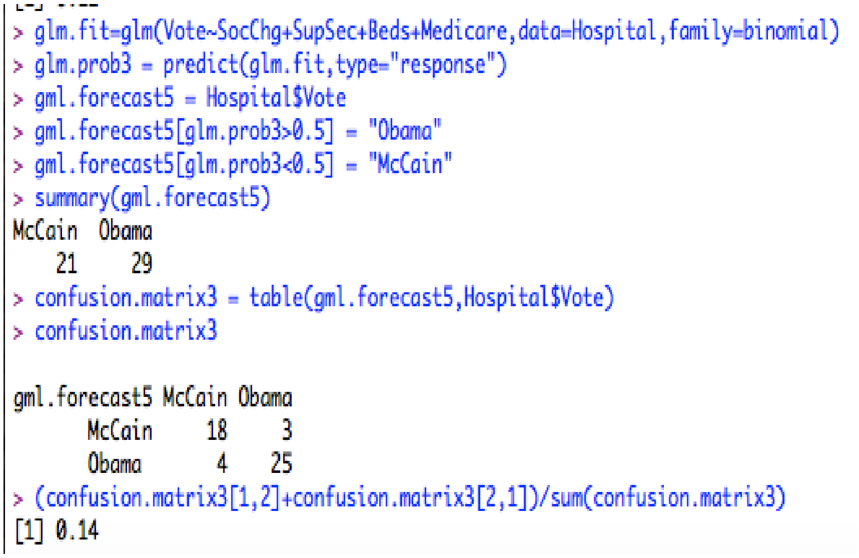
```
> sample(1:8,4)
[1] 6 7 2 4
```



```
> sample(1:8,4)
[1] 6 7 2 4
> head(Hospital)
  State Phys Beds MedChg Medicare   SocSec SocChg  SupSec  SocEnr   Vote
1    AL  233  339    9.6 16481.06 19824.64   9.42 3595.54  903569 McCain
2    AK  240  217   24.2  7862.30  9770.50  19.35 1667.12   64843 McCain
3    AZ  244  195   16.4 13235.25 15539.43  15.96 1648.92  922932 McCain
4    AR  226  348    7.1 16924.72 20373.79   8.14 3273.23  566219 McCain
5    CA  295  201    8.2 11683.97 12354.30   6.06 3348.38 4463873  Obama
6    CO  292  201   11.2 11139.94 12530.20   9.43 1190.35  584556  Obama
> glm.fit=glm(Vote~SocCha+SupSec+Beds+Medicare,data=Hospital,family=binomial)
Error in eval(predvars, data, env) : object 'SocCha' not found
> glm.fit=glm(Vote~SocChg+SupSec+Beds+Medicare,data=Hospital,family=binomial)
> glm.prob = predict(glm.fit,type="response")
> glm.prob
           1            2            3            4            5            6            7            8 
5.800568e-02 1.315819e-03 7.834703e-01 1.248846e-01 9.534839e-01 9.459218e-01 9.998693e-01 8.608843e-01 
           9           10           11           12           13           14           15           16 
9.568055e-01 2.183777e-02 9.119402e-01 2.018467e-01 9.554553e-01 7.318526e-01 7.711460e-01 2.258742e-01 
          17           18           19           20           21           22           23           24 
9.202091e-03 6.808374e-02 9.874814e-01 9.912598e-01 9.986628e-01 9.694657e-01 4.030359e-01 6.642377e-05 
          25           26           27           28           29           30           31           32 
6.206121e-01 2.866646e-04 4.345011e-02 9.121863e-02 9.904263e-01 9.970133e-01 9.353679e-01 5.055251e-01 
          33           34           35           36           37           38           39           40 
3.425429e-01 1.269669e-04 9.788902e-01 6.254393e-01 9.985644e-01 9.632589e-01 9.999145e-01 3.685701e-01 
          41           42           43           44           45           46           47           48 
1.469323e-06 2.570874e-02 4.748449e-02 6.244722e-01 9.917121e-01 8.719084e-01 9.892300e-01 7.115524e-02 
          49           50 
9.807593e-01 4.508878e-03 
> glm.prob >.5
    1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20    21    22    23 
FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE 
   24    25    26    27    28    29    30    31    32    33    34    35    36    37    38    39    40    41    42    43    44    45    46 
FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE 
   47    48    49    50 
 TRUE FALSE  TRUE FALSE 
```

b. For this problem I selected the X values based above Which became SocChg+SupSec+Beds+Medicare as X variables and Vote as the Y variable. I then created a glm model of the data points and named it glm.fit and predicted the points with glm.prob and the predict function. The Values ended up with numbers based on e making it hard to notice if the variables were higher or lower than 0.5. To conquer tha problem and see who voted for Obama I used glm.prob > .5 for being any value above .5 would be equal to an Obama vote or “TRUE” anything else would be a McCain vote or “FALSE”.

c. To convert the glm.fit into vote forecast I used the same method as in 7.e but with many different X values. In addition the outcomes turned out to be exactly the same so I retried the problem again with different x-valuables and they produced a different result so I know there were no errors.



```
> glm.fit=glm(Vote~SocChg+SupSec+Beds+Medicare,data=Hospital,family=binomial)
> glm.prob3 = predict(glm.fit,type="response")
> gml.forecast5 = Hospital$Vote
> gml.forecast5[glm.prob3>0.5] = "Obama"
> gml.forecast5[glm.prob3<0.5] = "McCain"
> summary(gml.forecast5)
McCain  Obama
    21     29
> confusion.matrix3 = table(gml.forecast5,Hospital$Vote)
> confusion.matrix3

gml.forecast5 McCain Obama
       McCain     18     3
       Obama       4    25
> (confusion.matrix3[1,2]+confusion.matrix3[2,1])/sum(confusion.matrix3)
[1] 0.14
```

d. There is no improvement from the previous question the error rate was exactly the same at 0.14.