STAT 425 - Homework #7

PROBLEM 1

```
> ##########################
> # STAT 425               #
> # Homework 7             #
> # Date: 12/02/11         #
> # Author: Luis Lin       #
> ##########################
> library(faraway)
> library(MASS)
> library(class)
```

Part a

```
> ###############################################################################
> ### PROBLEM 1
>
> ### Part a: Fit a binomial regression with Class as the response and
> ### the other nine variables. Report the residual deviance and
> ### associated degrees of freedom. Can this information be used to
> ### determine if this model fits the data? Explain.
>
> # Class is binary, can use directly as response var. in glm function
>
> g0=glm(Class~., data=wbca, family="binomial")
> summary(g0)

Call:
glm(formula = Class ~ ., family = "binomial", data = wbca)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.48282  -0.01179   0.04739   0.09678   3.06425

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.16678    1.41491   7.892 2.97e-15 ***
Adhes       -0.39681    0.13384  -2.965  0.00303 **
BNucl       -0.41478    0.10230  -4.055 5.02e-05 ***
Chrom       -0.56456    0.18728  -3.014  0.00257 **
Epith       -0.06440    0.16595  -0.388  0.69795
Mitos       -0.65713    0.36764  -1.787  0.07387 .
NNucl       -0.28659    0.12620  -2.271  0.02315 *
Thick       -0.62675    0.15890  -3.944 8.01e-05 ***
UShap       -0.28011    0.25235  -1.110  0.26699
USize        0.05718    0.23271   0.246  0.80589
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

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

    Null deviance: 881.388  on 680  degrees of freedom
Residual deviance:  89.464  on 671  degrees of freedom
AIC: 109.46

Number of Fisher Scoring iterations: 8

> deviance(g0)
[1] 89.4642
> df.residual(g0)
[1] 671
>
> # Residual deviance: 89.464  on 671  degrees of freedom
> # No, this information cannot be used to determine if the model fits
> # the data well since there is only 1 observation at each location
```

Part b

```
> ### Part b: Use AIC as the criterion to determine the best subset
> ### of variables. (Use the step function.)
>
> g=step(g0)
Start:  AIC=109.46
Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
    UShap + USize

        Df Deviance    AIC
- USize  1   89.523 107.52
- Epith  1   89.613 107.61
- UShap  1   90.627 108.63
<none>       89.464 109.46
- Mitos  1   93.551 111.55
- NNucl  1   95.204 113.20
- Adhes  1   98.844 116.84
- Chrom  1   99.841 117.84
- BNucl  1  109.000 127.00
- Thick  1  110.239 128.24

Step:  AIC=107.52
Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
    UShap

        Df Deviance    AIC
- Epith  1   89.662 105.66
- UShap  1   91.355 107.36
<none>       89.523 107.52
- Mitos  1   93.552 109.55
- NNucl  1   95.231 111.23
- Adhes  1   99.042 115.04
```

```
- Chrom  1  100.153 116.15
- BNucl  1  109.064 125.06
- Thick  1  110.465 126.47

Step:  AIC=105.66
Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap

        Df Deviance    AIC
<none>       89.662 105.66
- UShap  1   91.884 105.88
- Mitos  1   93.714 107.71
- NNucl  1   95.853 109.85
- Adhes  1  100.126 114.13
- Chrom  1  100.844 114.84
- BNucl  1  109.762 123.76
- Thick  1  110.632 124.63
> summary(g)

Call:
glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +
    Thick + UShap, family = "binomial", data = wbca)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.44161  -0.01119   0.04962   0.09741   3.08205

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  11.0333     1.3632   8.094 5.79e-16 ***
Adhes        -0.3984     0.1294  -3.080  0.00207 **
BNucl        -0.4192     0.1020  -4.111 3.93e-05 ***
Chrom        -0.5679     0.1840  -3.085  0.00203 **
Mitos        -0.6456     0.3634  -1.777  0.07561 .
NNucl        -0.2915     0.1236  -2.358  0.01837 *
Thick        -0.6216     0.1579  -3.937 8.27e-05 ***
UShap        -0.2541     0.1785  -1.423  0.15461
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 881.388  on 680  degrees of freedom
Residual deviance:  89.662  on 673  degrees of freedom
AIC: 105.66

Number of Fisher Scoring iterations: 8

> names(g$coef)[-1]
[1] "Adhes" "BNucl" "Chrom" "Mitos" "NNucl" "Thick" "UShap"
>
> # Best subset of variables:
> # Adhes, BNucl, Chrom, Mitos, NNucl, Thick, UShap
```

Part c

```
> ### Part c: Use the reduced model to predict the outcome for a new
> ### patient with predictor variables 1, 1, 3, 2, 1, 1, 4, 1, 1
> ### (same order as above). Give a confidence interval for your
> ### prediction.
>
> newdata=wbca[1,-1]; # to keep original names for data frame
> newdata[1,]=c(1,1,3,2,1,1,4,1,1);
> new.pred=predict(g, newdata, se=T); new.pred
$fit
       1
4.834428

$se.fit
[1] 0.5815185

$residual.scale
[1] 1

> ilogit(new.pred$fit)
        1
0.9921115
> ilogit(c(new.pred$fit - 1.96*new.pred$se.fit, new.pred$fit +
+    1.96*new.pred$se.fit))
        1         1
0.9757467 0.9974629
>
> # Note: predict returns the predicted value in the scale of
> # the linear predictor
>
> # Prediction: 0.992
> # Confidence Interval: (0.978,0.997)
```

Part d

```
> ### Part d: Suppose that a cancer is classified as benign if
> ### p > 0.5 and malignant if p < 0.5. Compute the number of
> ### errors of both types that will be made if this method is
> ### applied to the current data with the reduced model.
>
> # Note: fitted value on the scale of the response variable
> yhat= g$fitted.values;
>
> # determine if prediction is benign (1) or malignant (0)
> yhat=yhat>0.5
>
> # create contingency table
> table(yhat, wbca$Class)

yhat      0   1
  FALSE 227   9
```

```
  TRUE   11 434
>
> # errors
> class0Errors=sum(wbca$Class==0 & yhat!=0); class0Errors
[1] 11
> class1Errors=sum(wbca$Class==1 & yhat!=1); class1Errors
[1] 9
> totalErrors=sum(yhat != wbca$Class); totalErrors
[1] 20
>
> # error percentage
> n=length(wbca$Class)
> PercentError=round(totalErrors/n*100,2);PercentError
[1] 2.94
>
> # 0 if malignant, 1 if benign
> # FALSE(0) if malignant, TRUE(1) if benign
>
> # Class 0: p<0.5 (malignant)
> # Class 1: p>0.5 (benign)
>
> # Number of class 0 errors = 11
> # Number of class 1 errors = 9
> # Total number of errors = 20
```

Part e

```
> ### Part e: split the data into two parts | assign every third
> ### observation to a test set and the remaining two thirds of
> ### the data to a training set. Compare the prediction accuracy
> ### (on the test data) from the following four models:
>
> n=dim(wbca)[1]
> library(class)
> totalErrors.AIC=rep(0,25)
> PercentError.AIC=rep(0,25)
> totalErrors.AIC.knn=rep(0,25)
> PercentError.AIC.knn=rep(0,25)
> totalErrors.BIC=rep(0,25)
> PercentError.BIC=rep(0,25)
> totalErrors.BIC.knn=rep(0,25)
> PercentError.BIC.knn=rep(0,25)
>
> # repeat experiment 25 times
> for(i in 1:25) {
+
+ # split the data: test data (1/3 obs), training data (2/3 obs)
+ test.id=sample(1:n, round(n/3));
+ test.data=wbca[test.id,]; train.data=wbca[-test.id,]
+
+ # use training data to determine model
```

```
+ g0=glm(Class~., data=train.data, family="binomial")
+
+ # Errors for AIC and logistic regression
+ g=step(g0, trace=0)
+ yhat=predict(g, newdata=test.data, type="response")
+ yhat=yhat>0.5
+ totalErrors.AIC[i]=sum(yhat != test.data[,1]);
+ PercentError.AIC[i]=round(totalErrors.AIC[i]/dim(test.data)[1]*100,2)
+
+ # Errors for  1-NN (knn with k=1) with variables selected by AIC
+ var.names=names(g$coef)[-1]
+ yhat=knn(train.data[,var.names], test.data[,var.names],train.data[,1])
+ totalErrors.AIC.knn[i]=sum(yhat != test.data[,1]);
+
PercentError.AIC.knn[i]=round(totalErrors.AIC.knn[i]/dim(test.data)[1]*100
,2)
+
+ # Errors for BIC and logistic regression
+ g=step(g0, k=log(dim(train.data)[1]), trace=0)
+ yhat=predict(g, newdata=test.data, type="response")
+ yhat=yhat>0.5
+ totalErrors.BIC[i]=sum(yhat != test.data[,1]);
+ PercentError.BIC[i]=round(totalErrors.BIC[i]/dim(test.data)[1]*100,2)
+
+ # 1-NN (knn with k=1) with variables selected by BIC
+ var.names=names(g$coef)[-1]
+ yhat=knn(train.data[,var.names], test.data[,var.names],train.data[,1])
+ totalErrors.BIC.knn[i]=sum(yhat != test.data[,1]);
+
PercentError.BIC.knn[i]=round(totalErrors.BIC.knn[i]/dim(test.data)[1]*100
,2)
+ }
There were 50 or more warnings (use warnings() to see the first 50)
>
> totalErrors=data.frame('AIC'=totalErrors.AIC,
+      'BIC'=totalErrors.BIC,
+      'AIC.knn'=totalErrors.AIC.knn,
+            'BIC.knn'=totalErrors.BIC.knn)
>
> PercentErrors=data.frame('AIC'=PercentError.AIC,
+        'BIC'=PercentError.BIC,
+        'AIC.knn'=PercentError.AIC.knn,
+              'BIC.knn'=PercentError.BIC.knn)
>
> totalErrors
   AIC BIC AIC.knn BIC.knn
1    4   4       5       7
2    8   9       6       8
3    3   4       5       4
4    8   8      10       9
5    8  10      10      12
6    7   7       6       8
7   13  13      11      15
8    8  11      11       7
```

```
9    6   6        4        4
10   5   5        6        7
11   6   6        7        7
12   5   5        5        5
13   8   7        8       10
14  10  11       11       12
15  10  10        9        9
16  14  14       11       11
17   9   6       14       11
18  11  11       10       10
19  10   9       12       15
20   8   8       10       11
21   6   6        5        4
22  12  11       12       13
23   9   9       10        8
24   8   9        7       11
25   3   3        9        8
> PercentErrors
    AIC  BIC AIC.knn BIC.knn
1  1.76 1.76    2.20    3.08
2  3.52 3.96    2.64    3.52
3  1.32 1.76    2.20    1.76
4  3.52 3.52    4.41    3.96
5  3.52 4.41    4.41    5.29
6  3.08 3.08    2.64    3.52
7  5.73 5.73    4.85    6.61
8  3.52 4.85    4.85    3.08
9  2.64 2.64    1.76    1.76
10 2.20 2.20    2.64    3.08
11 2.64 2.64    3.08    3.08
12 2.20 2.20    2.20    2.20
13 3.52 3.08    3.52    4.41
14 4.41 4.85    4.85    5.29
15 4.41 4.41    3.96    3.96
16 6.17 6.17    4.85    4.85
17 3.96 2.64    6.17    4.85
18 4.85 4.85    4.41    4.41
19 4.41 3.96    5.29    6.61
20 3.52 3.52    4.41    4.85
21 2.64 2.64    2.20    1.76
22 5.29 4.85    5.29    5.73
23 3.96 3.96    4.41    3.52
24 3.52 3.96    3.08    4.85
25 1.32 1.32    3.96    3.52
> summary(totalErrors)
      AIC             BIC           AIC.knn         BIC.knn
 Min.   : 3.00   Min.   : 3.00   Min.   : 4.00   Min.   : 4.00
 1st Qu.: 6.00   1st Qu.: 6.00   1st Qu.: 6.00   1st Qu.: 7.00
 Median : 8.00   Median : 8.00   Median : 9.00   Median : 9.00
 Mean   : 7.96   Mean   : 8.08   Mean   : 8.56   Mean   : 9.04
 3rd Qu.:10.00   3rd Qu.:10.00   3rd Qu.:11.00   3rd Qu.:11.00
 Max.   :14.00   Max.   :14.00   Max.   :14.00   Max.   :15.00
> summary(PercentErrors)
      AIC             BIC           AIC.knn         BIC.knn
```

```
 Min.   :1.320   Min.   :1.320   Min.   :1.760   Min.   :1.760
 1st Qu.:2.640   1st Qu.:2.640   1st Qu.:2.640   1st Qu.:3.080
 Median :3.520   Median :3.520   Median :3.960   Median :3.960
 Mean   :3.505   Mean   :3.558   Mean   :3.771   Mean   :3.982
 3rd Qu.:4.410   3rd Qu.:4.410   3rd Qu.:4.850   3rd Qu.:4.850
 Max.   :6.170   Max.   :6.170   Max.   :6.170   Max.   :6.610
>
> colMeans(totalErrors)
    AIC     BIC AIC.knn BIC.knn
   7.96    8.08    8.56    9.04
> colMeans(PercentErrors)
    AIC     BIC AIC.knn BIC.knn
 3.5052  3.5584  3.7712  3.9820
>
> # Note: predict with type='Response' returns value in scaled response
> # Results shown above
> # The number of errors (and percent errors) is higher for 1NN and BIC
> # predictions
```

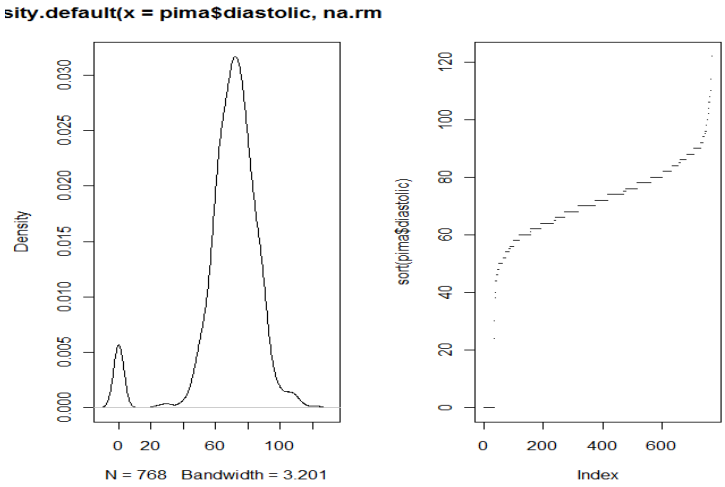
PROBLEM 2

```
> ##############################################################################
> ### PROBLEM 2
>
> data(pima)
> attach(pima)
```

Part a

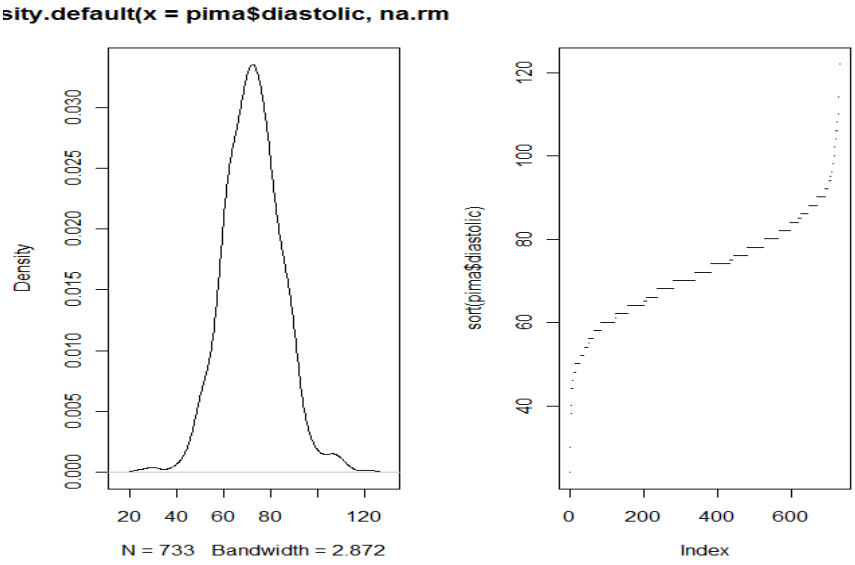
```
>
> ### Part a: Perform simple graphical and numerical summaries of the
> ### data. Can you find any obvious irregularities in the data?
> ### If you do,  take appropriate steps to correct the problems.
>
> summary(pima)
    pregnant         glucose        diastolic         triceps
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
    insulin           bmi          diabetes           age
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
 Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
```

```
      test
 Min.   :0.000
 1st Qu.:0.000
 Median :0.000
 Mean   :0.349
 3rd Qu.:1.000
 Max.   :1.000
>
> # Glucose,Diastolic,Triceps,Insulin & Bmi have minimum values of zero.
> # No blood pressure is not good for the health — something must
> # be wrong.
>
> # Kernel Densities Estimates and Sorted data vs index
> par(mfrow=c(1,2))
> plot(density(pima$diastolic,na.rm=TRUE))
> plot(sort(pima$diastolic),pch=".")
```

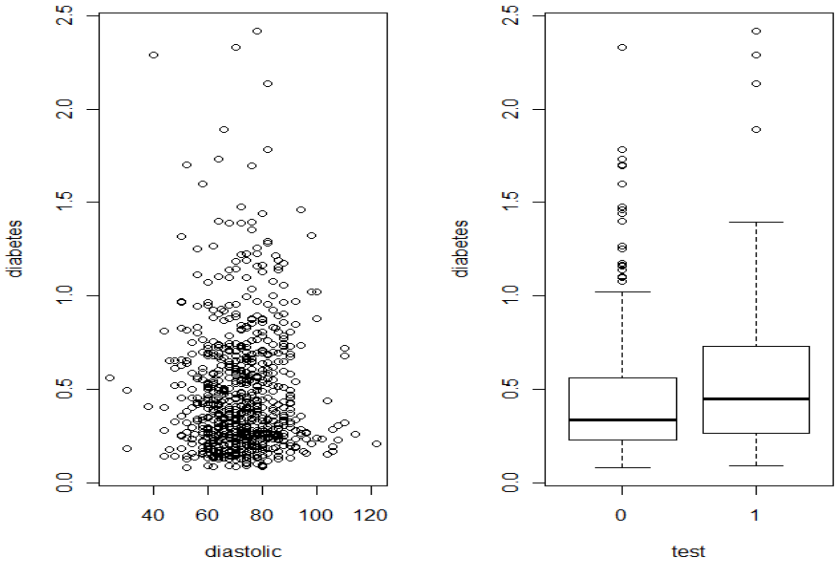



```
>
> sort(pima$diastolic)[1:40]
 [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0
[26]  0  0  0  0  0  0  0  0  0  0 24 30 30 38 40
>
> # First 36 values are zero
> # It seems likely that the zero has been used as a missing value code
> # Code zero values as NA
>
> pima$diastolic[pima$diastolic == 0] = NA
> pima$glucose[pima$glucose == 0] = NA
> pima$triceps[pima$triceps == 0] = NA
> pima$insulin[pima$insulin == 0] = NA
> pima$bmi[pima$bmi == 0] = NA
>
> # Variable test is categorical
> pima$test = factor(pima$test)
>
> # Numerical Summary
> summary(pima)
```

```
    pregnant         glucose       diastolic        triceps
 Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   :  7.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.: 22.00
 Median : 3.000   Median :117.0   Median : 72.00   Median : 29.00
 Mean   : 3.845   Mean   :121.7   Mean   : 72.41   Mean   : 29.15
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.: 36.00
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   : 99.00
                  NA's   :  5.0   NA's   : 35.00   NA's   :227.00
    insulin           bmi          diabetes          age        test
 Min.   : 14.00   Min.   :18.20   Min.   :0.0780   Min.   :21.00   0:500
 1st Qu.: 76.25   1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00   1:268
 Median :125.00   Median :32.30   Median :0.3725   Median :29.00
 Mean   :155.55   Mean   :32.46   Mean   :0.4719   Mean   :33.24
 3rd Qu.:190.00   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
 NA's   :374.00   NA's   :11.00
>
> # Graphical Summaries
>
> # Kernel Densities Estimates and Sorted data vs index
> par(mfrow=c(1,2))
> plot(density(pima$diastolic,na.rm=TRUE))
> plot(sort(pima$diastolic),pch=".")
```




```
> # Plots
> par(mfrow=c(1,2))
> plot(diabetes ~ diastolic,pima)
> plot(diabetes ~ test,pima)
```



Part b

```
> ### Part b: Fit a model with the result of the diabetes
> ### test as the response and all the other variables as
> ### predictors. Can you tell whether this model fits the data?
>
> # Test is binary, can use directly as response var. in glm function
>
> g0=glm(test ~., data=pima, family="binomial")
> summary(g0)

Call:
glm(formula = test ~ ., family = "binomial", data = pima)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7823  -0.6603  -0.3642   0.6409   2.5612

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.004e+01  1.218e+00  -8.246  < 2e-16 ***
pregnant     8.216e-02  5.543e-02   1.482  0.13825
glucose      3.827e-02  5.768e-03   6.635 3.24e-11 ***
diastolic   -1.420e-03  1.183e-02  -0.120  0.90446
triceps      1.122e-02  1.708e-02   0.657  0.51128
insulin     -8.253e-04  1.306e-03  -0.632  0.52757
bmi          7.054e-02  2.734e-02   2.580  0.00989 **
diabetes     1.141e+00  4.274e-01   2.669  0.00760 **
age          3.395e-02  1.838e-02   1.847  0.06474 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

    Null deviance: 498.10  on 391  degrees of freedom
Residual deviance: 344.02  on 383  degrees of freedom
  (376 observations deleted due to missingness)
AIC: 362.02

Number of Fisher Scoring iterations: 5

>
> # No, cannot used deviance to determine if the model fits
> # the data well since there is only 1 observation at each locatio
```

Part c

```
> ### Part c: What is the difference in the odds of testing positive
> ### for diabetes for a woman with a BMI at the first quartile
> ### compared with a woman at the third quartile, assuming that all
> ### other factors are held constant? Give a confidence interval for
> ### this difference.
>
> # Compute first and third quartile bmi
> bmi.Q25=as.numeric(quantile(pima$bmi,na.rm=TRUE,0.25))
> bmi.Q75=as.numeric(quantile(pima$bmi,na.rm=TRUE,0.75))
> diff.bmi = bmi.Q75-bmi.Q25
>
> # computer the factor difference
> odds.diff = exp(coef(g0)["bmi"]*diff.bmi)
> odds.diff = round(as.numeric(odds.diff),3)
> odds.diff
[1] 1.9
>
> # alternative 1
> x0=rep(1,dim(pima)[2])
> x0[7]=bmi.Q25
> eta0=sum(x0*g0$coeff)
>
> x1=rep(1,dim(pima)[2])
> x1[7]=bmi.Q75
> eta1=sum(x1*g0$coeff)
> exp(eta1-eta0)
[1] 1.900072
>
> # alternative 2
> newdata.25=pima[1,-dim(pima)[2]]; # to keep original names for data
frame
> newdata.25[1,]=rep(1,dim(pima)[2]-1)
> newdata.25['bmi']=bmi.Q25
> new.pred.25=predict(g0, newdata.25, se=T);
> odds.25=exp(new.pred.25$fit)
>
```

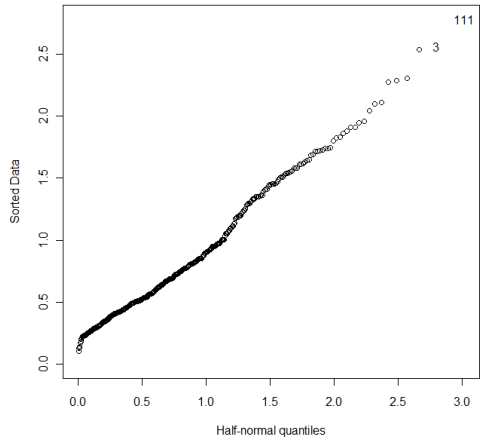
```
> newdata.75=pima[1,-dim(pima)[2]]; # to keep original names for data
frame
> newdata.75[1,]=rep(1,dim(pima)[2]-1)
> newdata.75['bmi']=bmi.Q75
> new.pred.75=predict(g0, newdata.75, se=T);
> odds.75=exp(new.pred.75$fit)
>
> odds.75/odds.25
       1
1.900072
>
> # compute confidence interval
>
> # standard error of bmi coefficient
> se.bmi = summary(g0)$coeff['bmi','Std. Error']
>
> # upper and lower estimates of bmi coefficient
> low.bmi = as.numeric(g0$coeff['bmi'] - 1.96*se.bmi)
> up.bmi = as.numeric(g0$coeff['bmi'] + 1.96*se.bmi)
>
> # upper and lower estimates of difference
> odds.diff.low=round(exp(low.bmi*diff.bmi),3)
> odds.diff.up=round(exp(up.bmi*diff.bmi),3)
>
> # estimated difference
> odds.diff.low
[1] 1.167
> odds.diff
[1] 1.9
> odds.diff.up
[1] 3.094
>
> # Note: differnce is the ratio of odds in Q3 over  odds in Q1.
> # Thus, the estimated multiplicative difference is 1.9,
> # with confience interval of (1.167,3.094).
> # Interpretation: The odds of testing positive for diabetes
> # for a woman with a BMI at the third quartile is 1.9 times more
> # than that of a woman at the first quartile. In other words,
> # being in the third quartile increases the odds of testing
> # positive by a factor of 1.9 compared to the first quartile.
>
> # NOTE: for better interpretability of difference in odds,
> # the difference is computed as the multiplicative difference,
> # that is, the factor change in odds by changing bmi
> # log(odds) = intercept + coeff.bmi*bmi + ... + coeff.var1*var1
> # For change only in bmi: odd2/odd1 = exp(coeff.bmi(bmi2-bmi1))
> # since other termms cancel out because same for bmi2 and bmi1
>
> # Interpretation of coefficients:
> # a unit increase in bmi with other variables fixed, increases the
> # log-odds of success by coeff(bmi) or increases the odds of success
> # by a factor of exp (coeff(bmi)). Thus, a incrase of bmi2-bmi
> # increases the odds by a factors of exp (coeff(bmi2-bmi1))
```

Part d

```
> ### Part d: Do women who test positive have higher diastolic
> ### blood pressures? Is the diastolic blood pressure signicant
> ### in the regression model? Explain the distinction between the
> ### two questions and discuss why the answers are only apparently
> ### contradict?
>
> t.test(pima$diastolic[pima$test==1],pima$diastolic[pima$test==0])

        Welch Two Sample t-test

data:  pima$diastolic[pima$test == 1] and pima$diastolic[pima$test == 0]
t = 4.6643, df = 504.716, p-value = 3.972e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.572156 6.316023
sample estimates:
mean of x mean of y
 75.32143  70.87734

>

> # p-value < 0.05, reject null hypothesis that difference in means
> # are zero. Thus, true difference in means is not equal to zero.
> # Which means women who test positive have higher diastolic
> # blood pressure on average since the difference in means
> # is significant
>
> summary(g0)$coef['diastolic','Pr(>|z|)']
[1] 0.9044642
>
> # The p-value of the diastolic coefficient in the full regression
> # model is greater than 0.05. Thus, it is not significant.
> # The model indicates that there is not a statistical significant
> # relationship between diastolic and the test.
>
> # However, answers don't contradict because the difference in means
> # refers to the marginal effect, while the coefficient in the
> # regression model refers to the join effect, that is conditioned
> # with other variables. Since variables can be correlated with
> # each other, the effect might not be significant in the presence
> # of othe variables when they are accounted for.
>
> summary(glm(test~diastolic, data=pima,
+    family="binomial"))$coef['diastolic','Pr(>|z|)']
[1] 5.718197e-06
>
> # Using diastloic as the only predictor shows it is significant,
> # so there is no contradiction.
```

Part e

```
> ### Part e: Perform diagnostics on the regression model, reporting
> ### any potential violations and any suggested improvements to the
model.
> ### Hint: produce the halfnorm plot for residuals and calculate the
> ### estimated dispersion.
>
> halfnorm(residuals(g0))
```




```
> (sigma2 = sum(residuals(g0,type="pearson")^2) /df.residual(g0))
[1] 1.061657
>
> # Halfnorm: no single outlier is apparent.
> # Dispersion Parameter Estimate: very close to 1, normal assumption fine
```

Part f

```
> ### Part f: Predict the outcome for a woman with predictor
> ### values 1, 99, 64, 22, 76, 27, 0.25, 25 (same order as in the
dataset).
> ### Give a confidence interval for your prediction.
>
> newdata=pima[1,-dim(pima)[2]]; # to keep original names for data frame
> newdata[1,]=c(1, 99, 64, 22, 76, 27, 0.25, 25);
> new.pred=predict(g0, newdata, se=T); new.pred
$fit
        1
-3.038116

$se.fit
[1] 0.3185671

$residual.scale
[1] 1
```

```
> ilogit(new.pred$fit)
         1
0.04573331
> ilogit(c(new.pred$fit - 1.96*new.pred$se.fit, new.pred$fit +
+    1.96*new.pred$se.fit))
         1          1
0.02502570 0.08213208
>
> # Prediction = 0.0457
> # Confidence interval : (0.0250,0.0821)
> # ilogit = pi = P(Yi = 1 | X = xi)
> # A 4.57% predicted chance of testing positive for the given women
profile
```

PROBLEM 3

```
> ##########################################################################
> ### PROBLEM 3

> data(aflatoxin)
```

Part a

```
> ### Part a: Build a model to predict the occurrence of liver cancer.
> ### Compute the ED50 level.
>
> # Construct a two-column matrix with the first column representing the
> # number of "successes" tumor and the second column the number of
> # "failures" total-tumor
>
> g0=glm(cbind(tumor,total-tumor)~dose, data=aflatoxin, family="binomial")
> summary(g0)

Call:
glm(formula = cbind(tumor, total - tumor) ~ dose, family = "binomial",
    data = aflatoxin)

Deviance Residuals:
      1        2        3        4        5        6
-1.2995   0.7959  -0.4814   0.4174  -0.1629   0.3774

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.03604    0.48226  -6.295 3.07e-10 ***
dose         0.09009    0.01456   6.189 6.04e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116.524  on 5  degrees of freedom
```

```
Residual deviance:   2.897  on 4  degrees of freedom
AIC: 17.685

Number of Fisher Scoring iterations: 5
>
> # ED50: effective dose for which there will be a 50% chance of success
> # log(p/(1-p)) = logit(p) = b0 + b1x , solving for x:
> # x=(logit(p)-b0)/b1. For p=1/2, logit(1/2)=0, so x = -b0/b1
>
> g0$coef
(Intercept)        dose
-3.03603648  0.09008878
> ED50 = as.numeric(- g0$coef[1]/g0$coef[2]); ED50
[1] 33.7005

> # ED50 = 33.7005
```

Part b

```
> ### Part b: Can you tell whether this model fits the data?
> ### Hint: lack of fit test; check whether you should use deviance
> ### or scaled deviance.
>
> # Compute Dispersion Parameter
>
> dp = sum(residuals(g0,type="pearson")^2) /df.residual(g0); dp
[1] 0.532547
>
> # Since dp=0.532 is not close to 1 and dp<1, indicates underdispersion
>
> pchisq(deviance(g0),df.residual(g0),lower=FALSE)
[1] 0.5752128
>
> # p-value>>.05: no evidence of lack of fit, current model fits
> # data sufficiently well
```

PROBLEM 4

```
> ########################################################################
> ### PROBLEM 4
>
> data(discoveries)
>
> ### The dataset discoveries lists the numbers of \great" inventions
> ### and scientific discoveries in each year from 1860 to 1959. Has the
> ### discovery rate remained constant over time?
>
> year=1860:1959
> g=glm(discoveries ~ year, family="poisson")
> summary(g)$coef['year','Pr(>|z|)']
[1] 0.006833334
>
```

```
> # When comparing Poisson models with overdispersion, an F-test rather
> # than a chi-squared test should be used
> # The drop1 function tests each predictor relative to the full.
>
> drop1(g, test="F")
Single term deletions

Model:
discoveries ~ year
       Df Deviance    AIC F value   Pr(F)
<none>      157.32 430.32
year    1   164.69 435.69  4.5904 0.03463 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Warning message:
In drop1.glm(g, test = "F") : F test assumes 'quasipoisson' family
>
> # p-value<.05, We see that year as predictor is significant.
>
> dp=sum(residuals(g, typp="pearson")^2)/df.residual(g); dp
[1] 1.605264
> (g$null.deviance - g$deviance)/dp
[1] 4.590385
>
> # dp>1: evidence that p-value in regression was overestimated
> # since didn't scale parameter (overestimate p-value means
> # that the p-value was smaller than it should be)
>
> summary(g, dispersion=dp)$coef['year','Pr(>|z|)']
[1] 0.0327716
>
> # p-value <0.05 when estimated dispersion parameter used.
> # Effect of year stat significant, which implies rate of
> # discovery not constant over time
>
```

PROBLEM 5

```
> ##########################################################################
> ### PROBLEM 5
>
> data(dvisits)
```

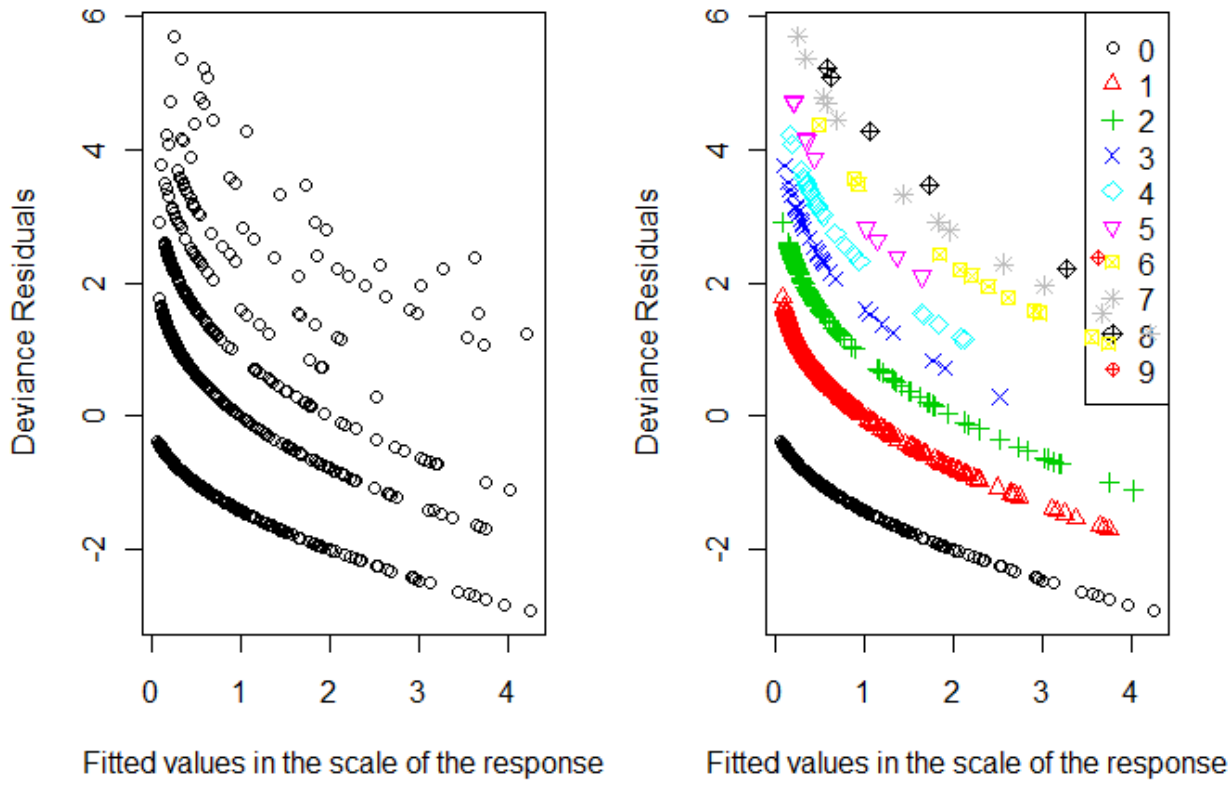
Part a

```
>
> ### Part a: Build a Poisson regression model with doctorco as the resp.
> ### and sex, age,agesq, income, levyplus, freepoor, freerepa, illness,
> ### actdays,hscore, chcond1,and chcond2 as possible predictor variables
> ### Considering the deviance of this model,does the model fit the data?
>
> g = glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor +
```

```
+                    freerepa + illness + actdays + hscore + chcond1 +
+  chcond2 ,family=poisson, dvisits)
> summary(g)

Call:
glm(formula = doctorco ~ sex + age + agesq + income + levyplus +
    freepoor + freerepa + illness + actdays + hscore + chcond1 +
    chcond2, family = poisson, data = dvisits)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9170  -0.6862  -0.5743  -0.4839   5.7005

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.223848   0.189816 -11.716   <2e-16 ***
sex          0.156882   0.056137   2.795   0.0052 **
age          1.056299   1.000780   1.055   0.2912
agesq       -0.848704   1.077784  -0.787   0.4310
income      -0.205321   0.088379  -2.323   0.0202 *
levyplus     0.123185   0.071640   1.720   0.0855 .
freepoor    -0.440061   0.179811  -2.447   0.0144 *
freerepa     0.079798   0.092060   0.867   0.3860
illness      0.186948   0.018281  10.227   <2e-16 ***
actdays      0.126846   0.005034  25.198   <2e-16 ***
hscore       0.030081   0.010099   2.979   0.0029 **
chcond1      0.114085   0.066640   1.712   0.0869 .
chcond2      0.141158   0.083145   1.698   0.0896 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5634.8  on 5189  degrees of freedom
Residual deviance: 4379.5  on 5177  degrees of freedom
AIC: 6737.1

Number of Fisher Scoring iterations: 6

> deviance(g)
[1] 4379.515
> df.residual(g)
[1] 5177
>
> pchisq(g$deviance,df.residual(g),lower=FALSE)
[1] 1
>
> dp=sum(residuals(g, typp="pearson")^2)/df.residual(g); dp
[1] 0.8459562
>
> # Residual deviance is about right for the corresponding degree of
> # freedom, also p-value >> .05, which indicates no evidence of
> # lack of fit, that is the model fits the data well.
```

Part b

```
> ### Part b: Plot the residuals and the fitted values why are there
> ### lines of observations on the plot?
>
> # note: predict function returns values in untransformed original
> # scale of the response variable, while g$fit returns value in the
> # transformed scale. For poisson, exp(predict(g))=g$fit
> # g$residuals are computed using the untransformed scale of
> # the response, that is g$residuals=dvisits$doctorco-predict(g)
>
> par(mfrow=c(1,2))
> plot(g$fit,residuals(g),xlab="Fitted values in the scale of the
response",
+ ylab="Deviance Residuals")
>
> # find the number of unique responses
>
> unique.resp.n=table(dvisits$doctorco);unique.resp.n

   0    1    2    3    4    5    6    7    8    9
4141  782  174   30   24    9   12   12    5    1
> unique.resp.id=as.numeric(names(unique.resp.n));unique.resp.id
 [1] 0 1 2 3 4 5 6 7 8 9
> n=length(unique.resp.id)
>
> # set plotting area
> plot(g$fit,residuals(g),xlab="Fitted values in the scale of the
response",
+ ylab="Deviance Residuals", type="n")
> legend("topright", col=(unique.resp.id+1),
+ pch=(unique.resp.id+1), legend=unique.resp.id)
>
> # plot with different color for each unique response value
> for(i in 1:n) {
+ ID=(dvisits$doctorco==unique.resp.id[i])
+ points(g$fit[ID],residuals(g)[ID], col=i, pch=i)
+ }
>
```




```
> # There are 10 lines of observation, each line corresponding to a
> # unique value (0,1,...10) of the response variable.
> # Since residuals invovles a difference in some form between
> # the fitted values and actual values, which are discrete in this
> # case,  then for a unique value of the response (eg. 0),
> # the residual formula will follow the same curve since the respose
> # value is just "scaled" by the fitted value. Thus,
> # for different unique values of the response, the residuals
> # will follow a different curve because the actual value is
> # diffrent for each unique value of the response.
> # So for example, for a fitted value of 1, the residuals for
> # each unique value of the response are different by
> # dvisits$doctorco, the unique value of the response.
>
```

Part c

```
> ### Part c: Use backward elimination with a critical p-value of 5%
> ### to reduce the model as much as possible. Report your model.
>
> # fit full model, and drop least significant variable
> # use drop1 command (F-test more appropriate for dispersion)
> # repeat with updated model until all variables significant at 5%
>
> fit=g
> F=drop1(fit,test="F")[-1,]
Warning message:
```

```
In drop1.glm(fit, test = "F") : F test assumes 'quasipoisson' family
> names=row.names(F)
> #determine insignificant
> test=F[,'Pr(F)']>.05
> #return insignificant
> notsig=F[test,'Pr(F)']
> names=names[test]
> #return the least significant position
> id=order(notsig,decreasing=TRUE)[1]
> #return the variable name of the least significant effect
> names[id] # drop agesq
[1] "agesq"
>
> fit=update(fit,.~. -agesq)
> F=drop1(fit,test="F")[-1,]
Warning message:
In drop1.glm(fit, test = "F") : F test assumes 'quasipoisson' family
> names=row.names(F)
> test=F[,'Pr(F)']>.05
> notsig=F[test,'Pr(F)']
> names=names[test]
> id=order(notsig,decreasing=TRUE)[1]
> names[id] # drop freerepa
[1] "freerepa"
>
> fit=update(fit,.~. -freerepa)
> F=drop1(fit,test="F")[-1,]
Warning message:
In drop1.glm(fit, test = "F") : F test assumes 'quasipoisson' family
> names=row.names(F)
> test=F[,'Pr(F)']>.05
> notsig=F[test,'Pr(F)']
> names=names[test]
> id=order(notsig,decreasing=TRUE)[1]
> names[id] # drop levyplus
[1] "levyplus"
>
> fit=update(fit,.~. -levyplus)
> F=drop1(fit,test="F")[-1,]
Warning message:
In drop1.glm(fit, test = "F") : F test assumes 'quasipoisson' family
> names=row.names(F)
> test=F[,'Pr(F)']>.05
> notsig=F[test,'Pr(F)']
> names=names[test]
> id=order(notsig,decreasing=TRUE)[1]
> names[id] # cannot drop any more variables
[1] NA
>
> g1=fit
> summary(g1)

Call:
glm(formula = doctorco ~ sex + age + income + freepoor + illness +
```

```
    actdays + hscore + chcond1 + chcond2, family = poisson, data =
dvisits)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9109  -0.6843  -0.5758  -0.4901   5.7654

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.069666   0.100158 -20.664  < 2e-16 ***
sex          0.169389   0.055669   3.043  0.00234 **
age          0.348222   0.143284   2.430  0.01509 *
income      -0.168246   0.082052  -2.050  0.04032 *
freepoor    -0.499105   0.175288  -2.847  0.00441 **
illness      0.185559   0.018238  10.175  < 2e-16 ***
actdays      0.126423   0.005021  25.180  < 2e-16 ***
hscore       0.030678   0.010045   3.054  0.00226 **
chcond1      0.124662   0.066386   1.878  0.06040 .
chcond2      0.161590   0.081691   1.978  0.04792 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5634.8  on 5189  degrees of freedom
Residual deviance: 4383.4  on 5180  degrees of freedom
AIC: 6735

Number of Fisher Scoring iterations: 6

> row.names(summary(g1)$coeff)
 [1] "(Intercept)" "sex"         "age"         "income"      "freepoor"
 [6] "illness"     "actdays"     "hscore"      "chcond1"     "chcond2"
>
> # For the response variable doctorco, the model includes
> # the intercept and predictors: sex,age,income,freepoor,
> # illness, actdays, hscore, chcond1, chcond2
```

Part d

```
> ### Part d: What sort of person would be predicted to visit the
> ### doctor the most under your selected model?
>
> # A person with a profile such that it will increase the
> # response, would be predicted to vist the docter more.
> # In other words, larger values of the variables for positive
> # coefficients and smaller values of the variables for
> # negative coefficients. Thus, this would be a person with
> # the following characteristics:
> # Sex: Female (since F = 1, and M=1)
> # Age: Older
> # Income: low - less than 200
```

```
> # Freepoor: not covered by goverment because of low income,
> #           recent immigrant or unemployed
> # Illness: with 5 or more illnesses in past 2 weeks
> # Actdays: Higher number of days of reduced activity
> #          in past two weeks due to illness or injury
> # Hscore: high score on General health questionnaire score
> #         using Goldberg's method (high score = bad health)
> # Chcond1: with chronic conditions(s) but not limited in activity
> # Chcond2: with chronic condition(s) and limited in activity
```

Part e

```
> ### Part e: For the last person in the dataset, compute the
> ### predicted probability distribution
> ### i.e., give the probability they visit 0,1,2, etc. times.
>
> # Find the last persion
> personID = dim(dvisits)[1];personID
[1] 5190
>
> # Retrieve profile
> person=dvisits[personID,]
>
> # Find the predicted number of visits to the doctor
> prediction=predict(g1, person); prediction
     5190
-1.861007
> # This is the expected log count of viists to the doctor
> lambda = round(exp(as.numeric(prediction)),3);lambda
[1] 0.156
> # This is the expected count of visits to the doctor
>
> # Probability distribution: Poisson with mean lambda=0.156
> # lambda^k/k*exp(-lambda)
>
> k=0:4
> p=round(dpois(k, lambda),4)
> p=cbind(p)
> row.names(p)=k
>
> #probabilities for number of visits
> p
       p
0 0.8556
1 0.1335
2 0.0104
3 0.0005
4 0.0000
>
```