Sample solutions

Stat 8051 Homework 7

Problem 1: Faraway Exercise 2.2

(a)

```
> lmod1 = glm(Class~., data=wbca, family=binomial)
> summary(lmod1)

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
```

The variables `Adhes, BNucl, Chrom, NNucl, Thick` turn out to be significant. `Mitos` has a marginal p-value. Residual deviance is 89.464, and 671 is the associated df.

This is ungrouped data, so we cannot do Hosmer and Lemeshow's goodness-of-fit test. However, we can do the Pearson's X^2 test:

```
> (X2 = sum(residuals(lmod1, type="pearson")^2))
[1] 221.3822
> 1-pchisq(X2, df=lmod1$df.residual)
[1] 1
```

The test statistic is 221.38, and the p-value is very large, which indicates a good fit.

(b)

```
> (lmod2 = step(lmod1, trace=F))

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

Coefficients:
(Intercept)        Adhes        BNucl        Chrom        Mitos        NNucl        Thick
    11.0333      -0.3984      -0.4192      -0.5679      -0.6456      -0.2915      -0.6216
      UShap
    -0.2541

Degrees of Freedom: 680 Total (i.e. Null);  673 Residual
Null Deviance:      881.4
Residual Deviance: 89.66  AIC: 105.7
```

The variables `Epith` and `UShap` are left out in the final model.

(c)

```
> newdata = c(1,1,3,2,1,1,4,1,1)
> newdata = data.frame(t(newdata))
> colnames(newdata) = colnames(wbca)[-1]
>
> p = predict(lmod2, newdata=newdata, se.fit=TRUE)
> (CI = with(p, c(fit-1.96*se.fit, fit, fit+1.96*se.fit)))
       1        1        1
3.694652 4.834428 5.974204
```

It is important here to first obtain the CI for log-odds, because the normality assumption of errors hold in log-odds scale in logistic regression. Once we have the CI, we can just revert back to the probability scale:

```
> lmod2$family$linkinv(CI)
        1         1         1
0.9757467 0.9921115 0.9974629
```

So this is our required 95% CI of predicted probability.

(d)

```
> fullpred = predict(lmod2, newdata=wbca,
+                     type="response")
> # or use fullpred = lmod2$fitted
> pred5 = ifelse(fullpred>.5, 1, 0)
>
> sum(pred5!=wbca$Class)
[1] 20
> table(pred5, wbca$Class)

pred5   0   1
    0 227   9
    1  11 434
```

There are a total 20 misclassified samples when 0.5 is taken as the cutoff.

(e)

```
> pred9 = ifelse(fullpred>.9, 1, 0)
>
> sum(pred9!=wbca$Class)
[1] 17
> table(pred9, wbca$Class)

pred9   0   1
    0 237  16
    1   1 427
```

With 0.9 as the cutoff, there are 17 misclassified observations. However 16 zeros are classified as ones, but only 1 one is misclassified as 0. This is expected because as you raise the cutoff you classify more and more samples as zeros. To get rid of this effect it is wise to choose a cutoff so that the two types of errors (type -I and type-II) stay somewhat balanced. A rule of thumb is to go for the sample average class probability.

(f)

```
> test = seq(3, nrow(wbca), by=3)
> lmod21 = update(lmod2, subset = -test)
>
> fullpred1 = predict(lmod21, newdata=wbca[test,],
+                      type="response")
>
> pred51 = ifelse(fullpred1>.5, 1, 0)
> sum(pred51!=wbca$Class[test])
[1] 7
```

```
> table(pred51, wbca$Class[test])

pred51   0   1
     0  70   2
     1   5 150
>
> pred91 = ifelse(fullpred1>.9, 1, 0)
> sum(pred91!=wbca$Class[test])
[1] 5
> table(pred91, wbca$Class[test])

pred91   0   1
     0  73   3
     1   2 149
```

Remember that here you are required to take the training-test split **NOT at random**. In this external validation though, both the cutoffs work more or less same.

Problem 2: Faraway Exercise 2.3

(a)

```
    pregnant         glucose        diastolic        triceps         insulin
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0
      bmi           diabetes           age             test
 Min.   : 0.00  Min.   :0.0780   Min.   :21.00   Min.   :0.000
 1st Qu.:27.30  1st Qu.:0.2437   1st Qu.:24.00   1st Qu.:0.000
 Median :32.00  Median :0.3725   Median :29.00   Median :0.000
 Mean   :31.99  Mean   :0.4719   Mean   :33.24   Mean   :0.349
 3rd Qu.:36.60  3rd Qu.:0.6262   3rd Qu.:41.00   3rd Qu.:1.000
 Max.   :67.10  Max.   :2.4200   Max.   :81.00   Max.   :1.000
```

The summary shows that for 6 variables there are 0 entries, which is impossible for `glucose`, `diastolic`, `triceps`, `insulin` and `bmi`. This can only mean NA entries which are put in as zeros. Apart from this the scatterplot matrix doesn't show any other irregularities (Fig. 1).

(b) We get rid of 0-values for these 5 variables and fit a model with all predictors:

```
> ind = with(pima, which(insulin==0 | triceps==0 |
+ glucose==0 | diastolic==0 | bmi==0))
```

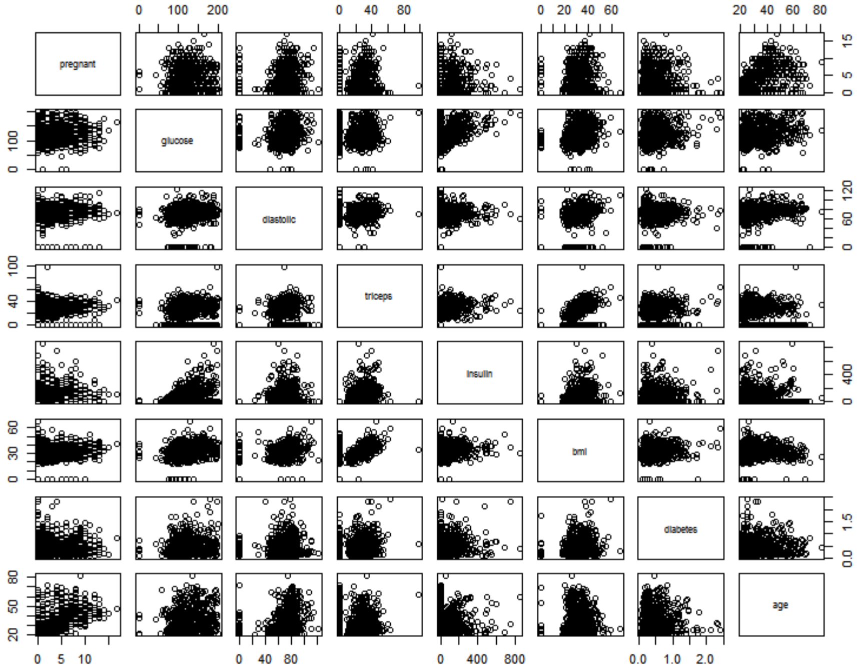

Figure 1: Scatterplot matrix for `pima` dataset

```
> lmod.pima = glm(test˜pregnant+glucose+diastolic+triceps+insulin+bmi+diabetes+age,
+ data=pima, subset=-ind,
+ family=binomial)
> summary(lmod.pima)

Call:
glm(formula = test ˜ pregnant + glucose + diastolic + triceps +
    insulin + bmi + diabetes + age, family = binomial, data = pima,
    subset = -ind)

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
AIC: 362.02

Number of Fisher Scoring iterations: 5
```

Glucose level, BMI and diabetes turn out to be significant while age has a marginal p-value.

```
> (X2 = sum(residuals(lmod.pima, type="pearson")^2))
[1] 406.6145
> 1-pchisq(X2, df=lmod.pima$df.residual)
[1] 0.1948219
```

Pearson's X^2 test indicates a good fit.

(c)

```
> (diff.bmi = with(pima,
+                  quantile(bmi, .75) - quantile(bmi, .25)))
75%
9.3
> (diff.logodd = 0.087*diff.bmi)
   75%
0.8091
> (se.logodd = 0.015*diff.bmi)
   75%
0.1395
> (CI.logodd = c(diff.logodd-1.96*se.logodd, diff.logodd, diff.logodd+1.96*se.logodd))
    75%     75%     75%
0.53568 0.80910 1.08252
> (CI.odd = exp(CI.logodd))
     75%      75%      75%
1.708610 2.245886 2.952110
```

Here again, get the CI for log-odds first. The IQR is 9.3, which at last yields an estimated odd of 2.246 and CI (1.71, 2.95).

(d)

```
> with(pima[-which(pima$diastolic==0),], t.test(diastolic~test))

Welch Two Sample t-test

data:  diastolic by test
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:
 -6.316023 -2.572156
sample estimates:
mean in group 0 mean in group 1
       70.87734        75.32143
```

A t-test shows that there is significant difference of `diabetes` between the two test groups. However, in the logistic regression model it is not a significant predictor. This happens because the two questions asked are different. The first only only enquires whether there is any relation between the two variables, while the second one asks for significance in a larger regression model with all other variables considered. Here the apparent effect of diastolic pressure in the t-test is actually due to masking effect of other predictors.

(e) The diagnostic plots (Fig. 2) show no outliers or influential points. The residual plot and QQ plot indicate that the current fit can be improved for better estimation of the mean function.

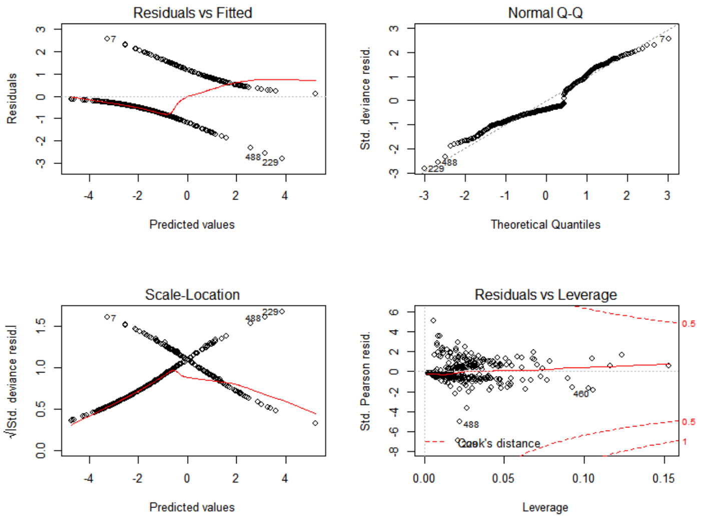

Figure 2: Diagnostic plots for diabetes test data

(f)

```
> newpima = data.frame(t(c(1, 99, 64, 22, 76, 27, .25, 25)))
> colnames(newpima) = colnames(pima)[-ncol(pima)]
>
> p.pima = predict(lmod.pima, newdata=newpima, se.fit=TRUE)
> (CI.pima = with(p.pima, c(fit-1.96*se.fit, fit, fit+1.96*se.fit)))
        1         1         1
-3.662508 -3.038116 -2.413725
> lmod.pima$family$linkinv(CI.pima)
         1          1          1
0.02502570 0.04573331 0.08213208
```

The CI in log-odds is (-3.66,-2.41) while in probability scale it is (0.025,0.082). 0.046 is the predicted probability of having a positive test.