C3M1: Peer Reviewed Assignment

Outline:

The objectives for this assignment:

1. Apply Binomial regression methods to real data.
2. Understand how to analyze and interpret binomial regression models.
3. Flex our math skills by determining whether certain distributions are members of the exponential family.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

In [53]:
```
# Load required libraries
library(tidyverse)
library(plyr)
library(dplyr)
library(gtable)
library(grid)
library(gridExtra)
```

Problem 1: Binomial (Logistic) Regression

The National Institute of Diabetes and Digestive and Kidney Diseases conducted a study of 768 adult female Pima Indians living near Phoenix, AZ. The purpose of the study was to investigate the factors related to diabetes.

Before we analyze these data, we should note that some have raised ethical issues with its collection and popularity in the statistics and data science community. We should think seriously about these concerns. For example, Maya Iskandarani wrote a brief piece on consent and privacy concerns raised by this dataset. After you familarize yourself with the data, we'll then turn to these ethical concerns.

First, we'll use these data to get some practice with GLM and Logistic regression.

In [54]:
```
# Load the data
pima = read.csv("pima.txt", sep="\t")
# Here's a description of the data: https://rdrr.io/cran/faraway/man/pima.html
head(pima)
```


A data.frame: 6 × 9

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<int>	<int>
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0

1. (a) Data Cleaning? What about Data Scrubbing? Data Sterilizing?

This is a real data set, which means that there's likely going to be gaps and missing values in the data. Before doing any modeling, we should inspect the data and clean it if necesary.

Perform simple graphical and numerical summaries of the data. Pay attention for missing or nonsensical values. Can you find any obvious irregularities? If so, take appropriate steps to correct these problems. In the markdown cell, specify what cleaning you did and why you did it.

Finally, split your data into training and test sets. Let the training set contain 80% of the rows and the test set contain the remaining 20%.

Solution to 1(a)

Basic summary and statistics:

- First we take a look at the summary of the data. It shows basic information of each variable, giving us a general idea of the distribution and some statistics of each factors.

In [55]:
```
# Your Code Here
summary(pima)
#The summary shows basic information of each variable; it gives a general
idea of how the numbers are distributed.

#Also check for missing values (NA)
sum(is.na(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
```

0

Comments on some values

The above summary shows basic statistics of the data. We can see that the minimum values for some of the factors do not make sense. Glucose, diastolic, triceps, insulin, and BMI should not have 0 as data entries. Since there are no missing values (NA) in the data, it can be assumed that the 0s were entered instead. We can swap these 0's for NAs to make sure that there are no major error in any future computations.

In [56]:
```
#Change the 0's to NA
names = c('glucose', 'diastolic', 'triceps', 'insulin', 'bmi')
pima[names][pima[names]==0] = NA
pima
```

A data.frame: 768 × 9

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<int>	<int>
1	6	148	72	35	NA	33.6	0.627	50	1
2	1	85	66	29	NA	26.6	0.351	31	0
3	8	183	64	NA	NA	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	NA	NA	25.6	0.201	30	0
7	3	78	50	32	88	31.0	0.248	26	1
8	10	115	NA	NA	NA	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	NA	NA	NA	0.232	54	1
11	4	110	92	NA	NA	37.6	0.191	30	0
12	10	168	74	NA	NA	38.0	0.537	34	1
13	10	139	80	NA	NA	27.1	1.441	57	0

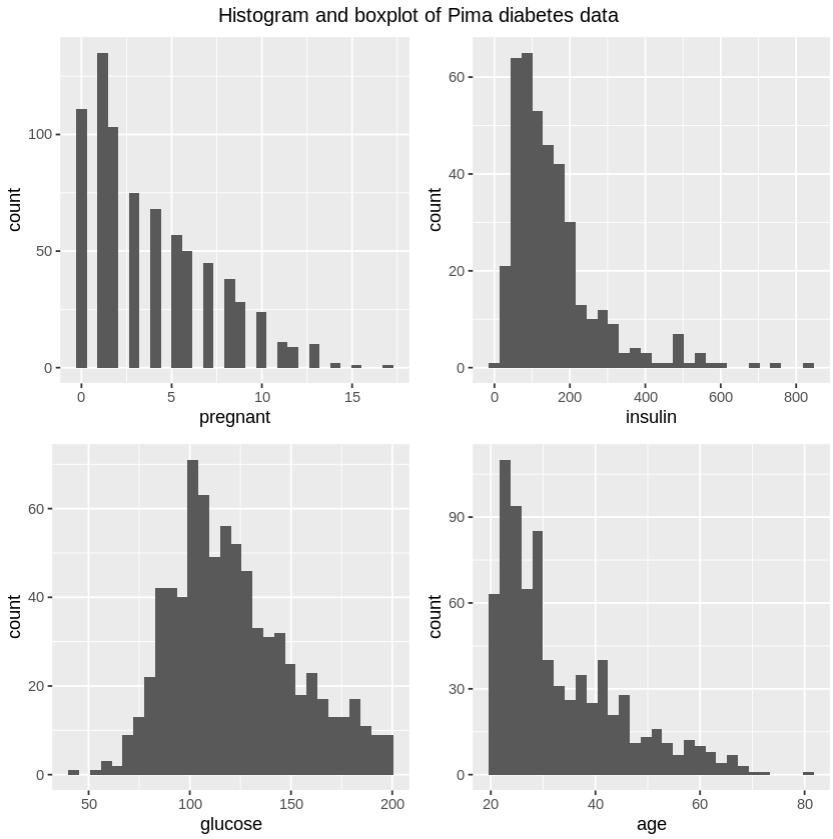
14	1	189	60	23	846	30.1	0.398	59	1
15	5	166	72	19	175	25.8	0.587	51	1
16	7	100	NA	NA	NA	30.0	0.484	32	1
17	0	118	84	47	230	45.8	0.551	31	1
18	7	107	74	NA	NA	29.6	0.254	31	1
19	1	103	30	38	83	43.3	0.183	33	0
20	1	115	70	30	96	34.6	0.529	32	1
21	3	126	88	41	235	39.3	0.704	27	0
22	8	99	84	NA	NA	35.4	0.388	50	0
23	7	196	90	NA	NA	39.8	0.451	41	1
24	9	119	80	35	NA	29.0	0.263	29	1
25	11	143	94	33	146	36.6	0.254	51	1
26	10	125	70	26	115	31.1	0.205	41	1
27	7	147	76	NA	NA	39.4	0.257	43	1
28	1	97	66	15	140	23.2	0.487	22	0
29	13	145	82	19	110	22.2	0.245	57	0
30	5	117	92	NA	NA	34.1	0.337	38	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
739	2	99	60	17	160	36.6	0.453	21	0
740	1	102	74	NA	NA	39.5	0.293	42	1
741	11	120	80	37	150	42.3	0.785	48	1
742	3	102	44	20	94	30.8	0.400	26	0
743	1	109	58	18	116	28.5	0.219	22	0
744	9	140	94	NA	NA	32.7	0.734	45	1
745	13	153	88	37	140	40.6	1.174	39	0
746	12	100	84	33	105	30.0	0.488	46	0
747	1	147	94	41	NA	49.3	0.358	27	1
748	1	81	74	41	57	46.3	1.096	32	0
749	3	187	70	22	200	36.4	0.408	36	1
750	6	162	62	NA	NA	24.3	0.178	50	1
751	4	136	70	NA	NA	31.2	1.182	22	1
752	1	121	78	39	74	39.0	0.261	28	0
753	3	108	62	24	NA	26.0	0.223	25	0
754	0	181	88	44	510	43.3	0.222	26	1
755	8	154	78	32	NA	32.4	0.443	45	1

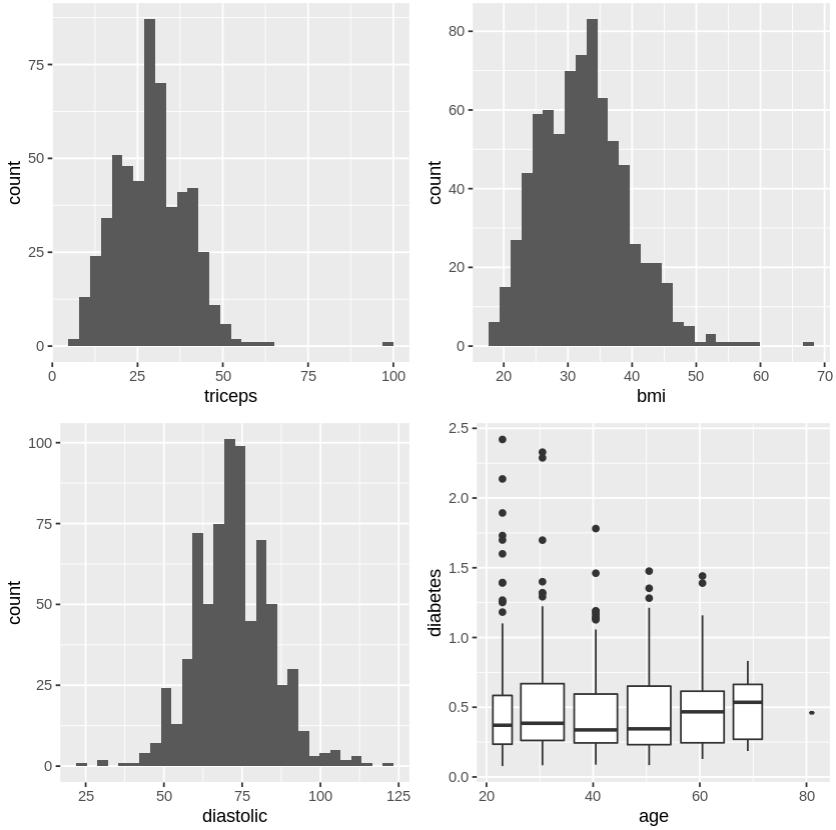
756	1	128	88	39	110	36.5	1.057	37	1
757	7	137	90	41	NA	32.0	0.391	39	0
758	0	123	72	NA	NA	36.3	0.258	52	1
759	1	106	76	NA	NA	37.5	0.197	26	0
760	6	190	92	NA	NA	35.5	0.278	66	1
761	2	88	58	26	16	28.4	0.766	22	0
762	9	170	74	31	NA	44.0	0.403	43	1
763	9	89	62	NA	NA	22.5	0.142	33	0
764	10	101	76	48	180	32.9	0.171	63	0
765	2	122	70	27	NA	36.8	0.340	27	0
766	5	121	72	23	112	26.2	0.245	30	0
767	1	126	60	NA	NA	30.1	0.349	47	1
768	1	93	70	31	NA	30.4	0.315	23	0

b) A histogram is a good way to observe variation in the data, so I plotted them for each of the variables in the dataset. Also, I used boxplots to break up the data into quartiles. The advantage of this is a more intuitive representation of large datasets consisting of continuous variables as well as the detection of outliers, as evident in the plots below.

In [57]:
```
layout(matrix(c(1, 2, 3, 4), 2, 2))
p1 = ggplot(data = pima, mapping = aes(x = pregnant)) + geom_histogram()
p2 = ggplot(data = pima, mapping = aes(x = insulin)) + geom_histogram()
p3 = ggplot(data = pima, mapping = aes(x = glucose)) + geom_histogram()
p4 = ggplot(data = pima, mapping = aes(x = age)) + geom_histogram()
p5 = ggplot(data = pima, mapping = aes(x = triceps)) + geom_histogram()
p6 = ggplot(data = pima, mapping = aes(x = bmi)) + geom_histogram()
p7 = ggplot(data = pima, mapping = aes(x = diastolic)) + geom_histogram()
p8 = ggplot(data = pima, mapping = aes(x = age, y = diabetes)) +
        geom_boxplot(mapping = aes(group = cut_width(age, 10)))

grid.arrange(
    p1,
    p2,
    p3,
    p4,
    nrow = 2,
    top = "Histogram and boxplot of Pima diabetes data")

grid.arrange(
    p5,
    p6,
    p7,
    p8,
    nrow = 2)
    #top = "Histogram and boxplot of Pima diabetes data")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:
"Removed 374 rows containing non-finite values (stat_bin)."
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:
"Removed 5 rows containing non-finite values (stat_bin)."
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:
"Removed 227 rows containing non-finite values (stat_bin)."
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:
"Removed 11 rows containing non-finite values (stat_bin)."
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:
"Removed 35 rows containing non-finite values (stat_bin)."
```





Comments on factor histograms

The most common data for each factor is as follows: 0-3 for pregnancy 0 for insulin 100-120 for glucose early-mid 20s for age 0 for triceps mid 30s for bmi 80 for diastolic

Boxplot of diabetes level based on age

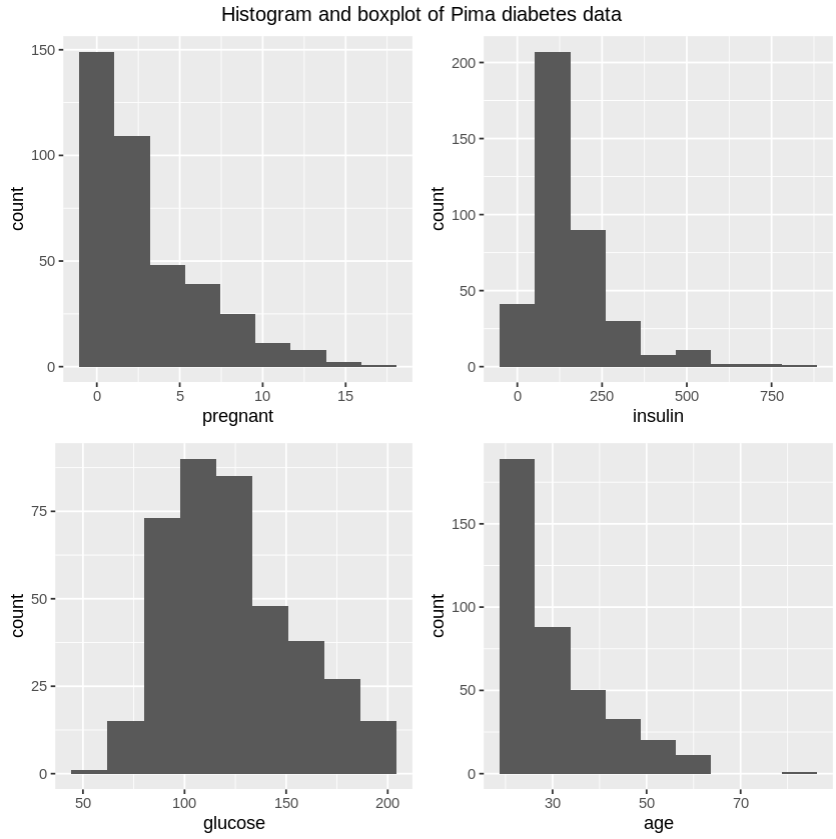
The boxplot shows that average diabetes level increases significantly after the age of 50.
There are quite a few more outliers in the 20-30 age group, possibly due to more irregular sleeping and eating habits as well as varying activity and metabolism levels compared to the older age groups.

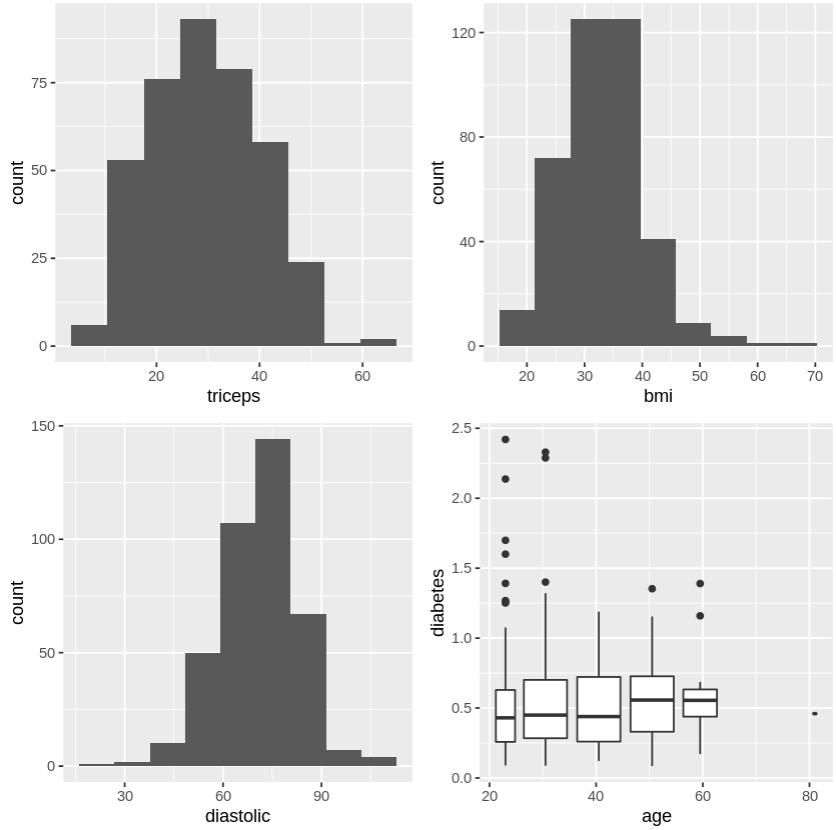
Overall spread and unusual values

The boxplot and histograms show some unusual dips and 0 values which make them visually less appealing. Adjusting the bin width along with getting rid of the NA values will help make the data and graphs cleaner.

In [64]:
```
pima = na.omit(pima)

layout(matrix(c(1, 2, 3, 4), 2, 2))
p1 = ggplot(data = pima, mapping = aes(x = pregnant)) + geom_histogram(bin
s = 9)
p2 = ggplot(data = pima, mapping = aes(x = insulin)) + geom_histogram(bins
 = 9)
p3 = ggplot(data = pima, mapping = aes(x = glucose)) + geom_histogram(bins
=9)
p4 = ggplot(data = pima, mapping = aes(x = age)) + geom_histogram(bins = 9
)
p5 = ggplot(data = pima, mapping = aes(x = triceps)) + geom_histogram(bins
 = 9)
```

```
p6 = ggplot(data = pima, mapping = aes(x = bmi)) + geom_histogram(bins = 9
)
p7 = ggplot(data = pima, mapping = aes(x = diastolic)) + geom_histogram(bi
ns = 9)
p8 = ggplot(data = pima, mapping = aes(x = age, y = diabetes)) +
        geom_boxplot(mapping = aes(group = cut_width(age, 10)))

grid.arrange(
    p1,
    p2,
    p3,
    p4,
    nrow = 2,
    top = "Histogram and boxplot of Pima diabetes data")

grid.arrange(
    p5,
    p6,
    p7,
    p8,
    nrow = 2)
    #top = "Histogram and boxplot of Pima diabetes data")

# Randomize the rows for splitting into test and training sets.
set.seed(42)
randomize.rows = sample(nrow(pima))
pima.shuffled = pima[randomize.rows, ]
pima.train = pima.shuffled[1:313, ]
pima.test = pima.shuffled[-(1:313),]
```





1. (b) Initial GLM modelling

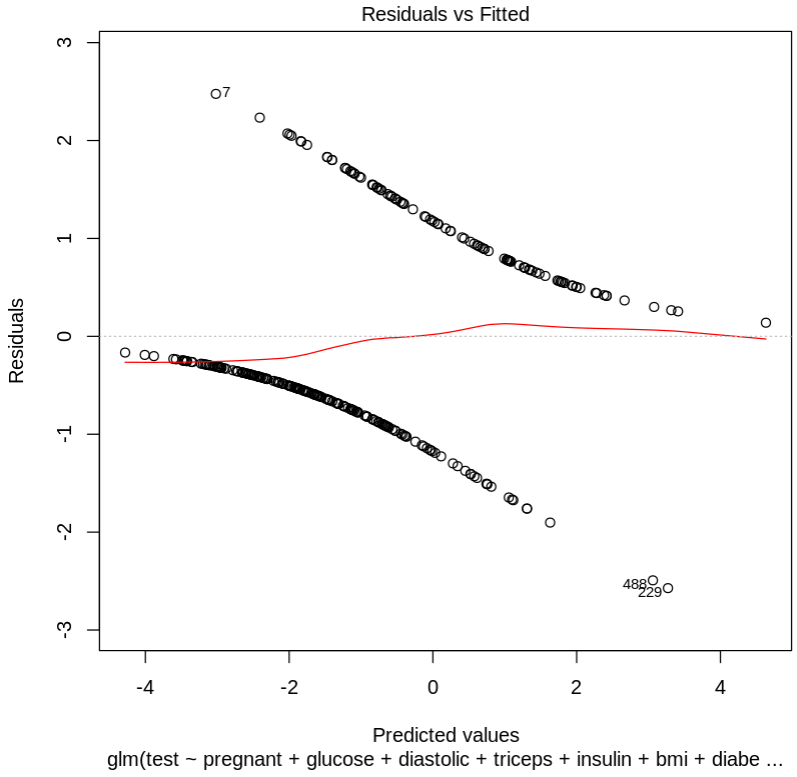
Our data is clean and we're ready to fit! What kind of model should we use to fit these data? Notice that the `test` variable is either 0 or 1, for whether the individual tested positive for diabetes. Because `test` is binary, we should use logistic regression (which is a kind of binomial regression).

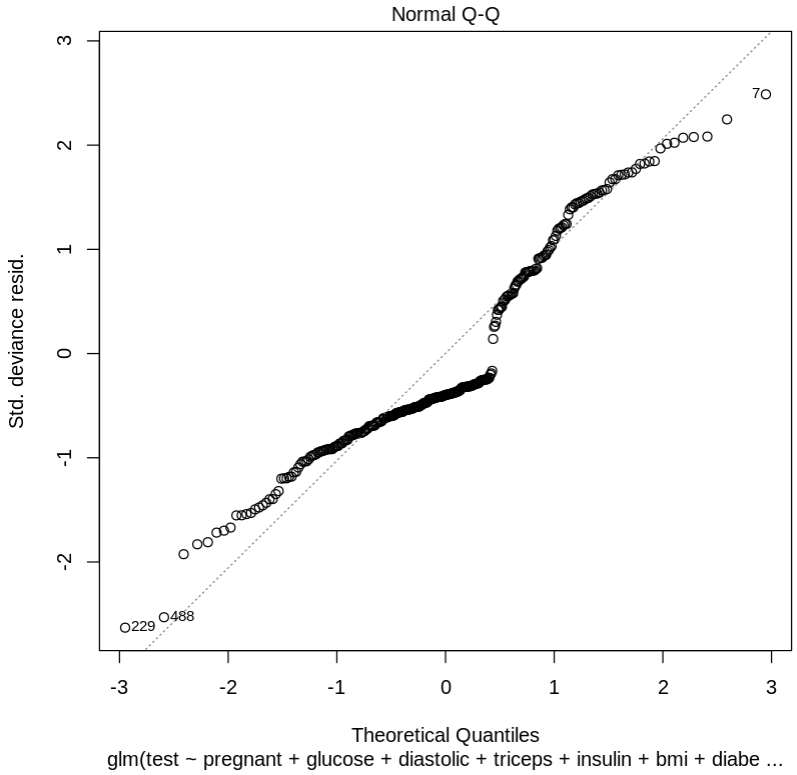
Fit a model with `test` as the response and all the other variables as predictors. Can you tell whether this model fits the data?

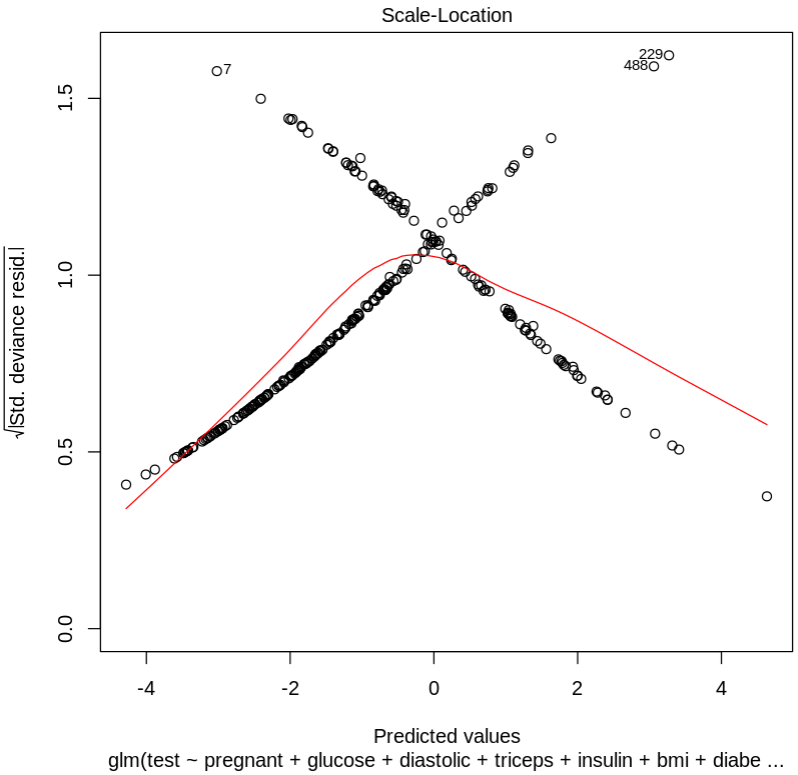
In [65]:
```
# Your Code Here
# For binary data, we can use the logistic regression.
glmod = glm(test ~ pregnant + glucose + diastolic + triceps + insulin + bm
i + diabetes + age, data = pima.train, family = "binomial")
summary(glmod)
plot(glmod)
```

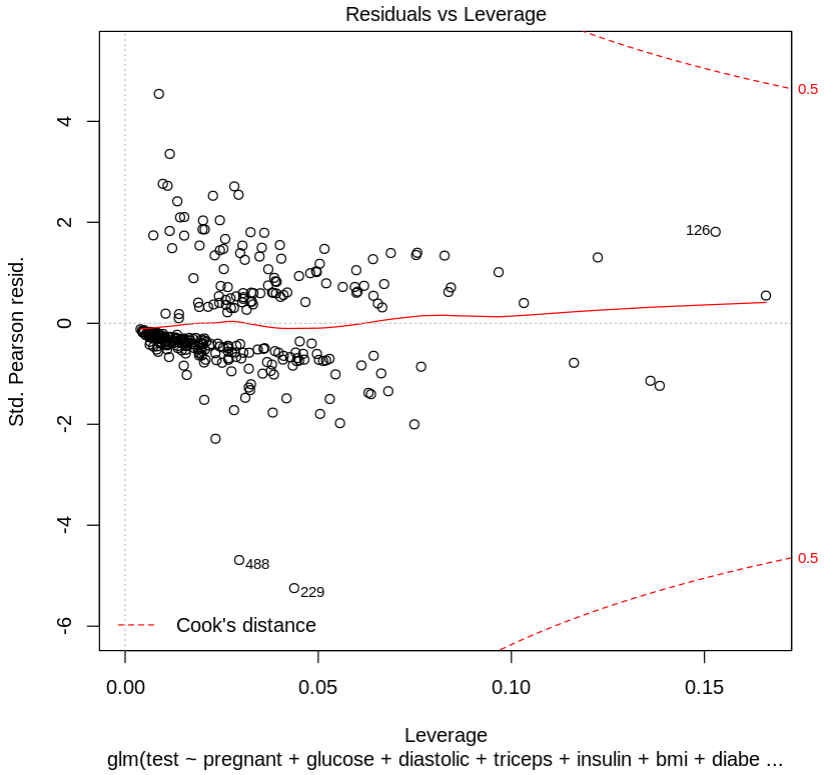
```
Call:
glm(formula = test ~ pregnant + glucose + diastolic + triceps +
    insulin + bmi + diabetes + age, family = "binomial", data = pima.train
)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5719  -0.6861  -0.3945   0.6791   2.4766

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.309342   1.345590  -6.918 4.57e-12 ***
pregnant     0.067247   0.061131   1.100   0.2713
```

```
glucose      0.036841   0.006312   5.837 5.33e-09 ***
diastolic   -0.004729   0.013684  -0.346   0.7296
triceps      0.016822   0.019046   0.883   0.3771
insulin     -0.001188   0.001386  -0.857   0.3914
bmi          0.052714   0.030177   1.747   0.0807 .
diabetes     0.997098   0.467373   2.133   0.0329 *
age          0.043704   0.020877   2.093   0.0363 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 397.99  on 312  degrees of freedom
Residual deviance: 283.33  on 304  degrees of freedom
AIC: 301.33

Number of Fisher Scoring iterations: 5
```




Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(test ~ pregnant + glucose + diastolic + triceps + insulin + bmi + diabe ...
7
229
488


Scale-Location
√|Std. deviance resid.|
Predicted values
glm(test ~ pregnant + glucose + diastolic + triceps + insulin + bmi + diabe ...
7
229
488



Interpretation

Whether the data fits the model is not completely intuitive, but we can see from the standard deviance and residuals that there are certain values that indicate some coherence between the training data and predicted values.

1. (c) Remember Bayes

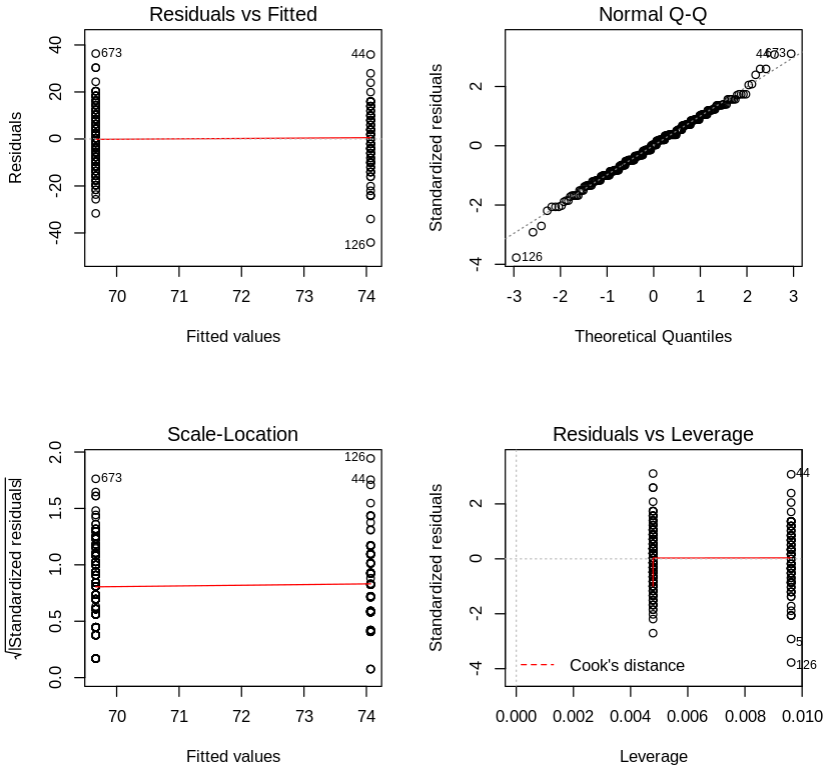
A quick analytical interlude.

Is diastolic blood pressure significant in the regression model? Do women who test positive have higher diastolic blood pressures? Explain the distinction between the two questions and discuss why the answers are only apparently contradictory.

In [66]:
```
# Your Code Here
lmod2 = lm(diastolic ~ test, data = pima.train)
summary(lmod2)
par(mfrow = c(2,2))
plot(lmod2)
```

```
Call:
lm(formula = diastolic ~ test, data = pima.train)

Residuals:
    Min      1Q  Median      3Q     Max 
-44.067  -7.665   0.335   7.933  36.335 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  69.6651     0.8112  85.880  < 2e-16 ***
test          4.4022     1.4073   3.128  0.00193 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.73 on 311 degrees of freedom
Multiple R-squared:  0.03051,   Adjusted R-squared:  0.02739
F-statistic: 9.786 on 1 and 311 DF,  p-value: 0.001925
```



Interpretation

The summary returns a p-value of less than a typical alpha value of 0.05, which means that there is a statistically significant relationship between test results and diastolic blood pressure.

The point of contradiction between the two questions "is diastolic blood pressure significant in the regression model" and "do women who test positive have higher diastolic blood pressures" is in the ordering of cause and effect. Bayes' theorem is used to determine the probability of an event when a certain condition is present. Mathematically, it is $P(A \mid B) = P(A|B)P(B)P(A)$, which is not the same when A and B are switched.

1. (d) GLM Interpretation

We've seen so many regression summaries up to this point, how is this one different from all the others? Well, to really understand any model, it can be helpful to loop back and plug the fitted results back into the model's mathematical form.

Explicity write out the equation for the binomial regression model that you fit in (b). Then, in words, explain how a 1 unit change of `glucose` affects `test`, assuming all other predictors are held constant.

In [67]:
```
# Your Code Here
summary(glmod)
```

```
Call:
glm(formula = test ~ pregnant + glucose + diastolic + triceps +
    insulin + bmi + diabetes + age, family = "binomial", data = pima.train
)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5719  -0.6861  -0.3945   0.6791   2.4766

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.309342   1.345590  -6.918 4.57e-12 ***
pregnant     0.067247   0.061131   1.100   0.2713
glucose      0.036841   0.006312   5.837 5.33e-09 ***
diastolic   -0.004729   0.013684  -0.346   0.7296
triceps      0.016822   0.019046   0.883   0.3771
insulin     -0.001188   0.001386  -0.857   0.3914
bmi          0.052714   0.030177   1.747   0.0807 .
diabetes     0.997098   0.467373   2.133   0.0329 *
age          0.043704   0.020877   2.093   0.0363 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 397.99  on 312  degrees of freedom
Residual deviance: 283.33  on 304  degrees of freedom
AIC: 301.33

Number of Fisher Scoring iterations: 5
```

Solution

For binomial/logistic regression, the model can be formulated in terms of log-odds:

$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots = \log(p1 - p)$

The predictors x_i are pregnancy, glucose, diastolic, etc.

The parameters β_i are the numbers given in the "Estimate" column of the summary table above.

Then, the explicit equation becomes:

$\eta = -9.309 + 0.067\text{pregnant} + 0.037\text{glucose} - 0.005\text{diastolic} + 0.017\text{triceps} - 0.001\text{insulin} + 0.053\text{bmi} +$

The coefficient for glucose indicates that a one unit increase results in a 0.037 increase in the log-odds of a positive test, holding other predictors constant.

1. (e) GLM Prediction

One of the downsides of Logistic Regression is that there isn't an easy way of evaulating the goodness of fit of the model without predicting on new data. But, if we have more data to test with, then there are many methods of evaluation to use. One of the best tools are confusion matrices,

which (despite the name) are actually not that hard to understand.

A confusion matrix compares the predicted outcomes of a Logistic Regression Model (or any classification model) with the actual classifications. For binary classification, it is a 2 × 2 matrix where the rows are the models' predicted outcome and the columns are the actual classifications. An example is displayed below.

	True	False
1	103	37
0	55	64

In the example, we know the following information:

- The [1,1] cell is the number of datapoints that were correctly predicted to be 1. The value (103) is the number of True Positives (TP).
- The [2,2] cell is the number of datapoints that were correctly predicted to be 0. The value is the number of True Negatives (TN).
- The [1, 2] cell is the number of datapoints that were predicted to be 1 but where actually 0. This is the number of False Positives (FP), also called Type I error. In the context of our diabetes dataset, this would mean our model predicted that the person would have diabetes, but they actually did not.
- The [2, 1] cell is the number of datapoints that were predicted to be 0 but where actually 1. This is the number of False Negatives (FN), also called Type 2 error. In the context of our diabetes dataset, this would mean our model predicted that the person would not have diabetes, but they actually did have diabetes.

Use your model to predict the outcomes of the test set. Then construct a confusion matrix for these predictions and display the results.

In [99]:
```
# Your Code Here
pima.predict = predict.glm(glmod, pima.test, type = "response")
pima.test$predict = ifelse(pima.predict > 0.5, 1, 0)
table(factor(pima.test$predict, levels = min(pima.test$test):max(pima.test
$test)),
      factor(pima.test$test, levels = min(pima.test$test):max(pima.test$te
st)))
```

```
   0  1
0 49 10
1  4 16
```

Interpretation
Here, the 0's represent True and 1's represent False.
The table shows that there were 10 type 1 errors and 4 type 2 errors.

1. (f) Evaluation Statistics

Using the four values from the confusion matrix, we can construct evaulation statistics to get a numerical approximation for our model's performance. Spend some time researching accuracy,

precision, recall and F score.

Calculate these values for your model's predictions on the test set. Clearly display your results. How well do you think your model fits the data?

In [108]:
```
# Your Code Here
# Accuracy measures how often the matrix is correct.
accuracy = (49 + 16) / 79

# Precision is the measure of how often the 'TRUE' prediction is correct.
precision = 49 / 59

# Recall is a measurement of how often it is predicted TRUE when it is ind
eed actually TRUE.
recall = 49 / 53

# F score is the weighted average of recall and precision.
fscore = 2*precision*recall / (precision + recall)

accuracy
precision
recall
fscore
```

0.822784810126582

0.830508474576271

0.924528301886792

0.875

Interpretation
The overall accuracy, precision, and recall (measures of how well the prediction model performs, put in confusion matrix terms) came out to be quite high. Just by looking at the matrix, we can tell that false positives and negatives are not as common as the correctly predicted outcomes.

1. (g) Understanding Evaluation Statistics

Answer the following questions in the markdown cell below.

1. Give an example scenario for when accuracy would be a misleading evaulation statistic.
2. Confusion matrices can also be used for non-binary classification problems. Describe what a confusion matrix would look like for a response with 3 levels.
3. You'll have to take our word on the fact (or spend some time researching) that Type I error and Type II error are inversely related. That is, if a model is very good at detecting false positives, then it will be bad at detecting false negatives. In the case of our diabetes dataset, would you prefer a model that overestimates the Type 1 error or overestimates the Type II error. Justify your answer.

Solution

1. Given a hypothetical training set data where the relationship between the predictors and response is completly linear (or binary), ff that training set is used to create a regression model, then testing the model with a separate test set would not provide any meaningful result since the data itself was highly skewed (and hence the model as well) towards a certain outcome.

1.

	1		2	3		
1	155	12	2	1	24	67
2	13	45	4	2	22	42
3	8	11	16	12	18	69

For the 2x2 confusion matrix, the diagonal entries represented the correctly predicted positives and negatives. Similarly, the diagonal entries for this confusion matrix represent the modeled values for each corresponding levels.

1. For the diabetes test set, it would be better for there to be a higher false positive (type 1 error). This is because, if an individual starts to think that he or she has diabetes, then that person would be motivated to adopt healthier lifestyles such as increased exercise and smarter eating habits. Also, a false positive is easy to correct with additional tests and diagnoses. A type 2 error would do exactly the opposite and cause more harm to the individual.

In [ ]:

1. (h) Ethical Issues in Data Collection

Read Maya Iskandarani's piece on consent and privacy concerns raised by this dataset. Summarize those concerns here.

Maya Iskandarani's article addresses concern on the broadness and difficulty of giving consent to participate in medical research. Firstly, the Pima were subjects of rheumatoid arthritis research but later became those of a 'natural laboratory' for diabetes research and observation. Their personal and sensitive data were used for many generations, including acting as a database for machine learning repository, which gave rise to privacy concerns and what it really means to give consent to participate. The main issue is that no one really knows for how long and what purpose the data we gave access to will be used.

Problem 2: Practicing those Math skills

One of the conditions of GLMs is that the "random component" of the data needs to come from the Exponential Family of Distributions. But how do we know if a distribution is in the Exponential Family? Well, we could look it up. Or we could be proper mathematicians and check the answer ourselves! Let's flex those math muscles.

2. (a) But it's in the name...

Show that Y ~ exponential(λ), where λ is known, is a member of the exponential family.

Solution
Y exponential(λ) is a one-parameter exponential family if the probability mass/density function can be expressed as: $P(Y \mid \lambda) = h(Y)\exp(\eta(\lambda)T(Y) - B(\lambda))$

We're given $f(y; \lambda) = \lambda e^{-\lambda y}$. We can take log of both sides and then exp both sides again to keep left term constant: $\exp(\log(f(y; \lambda))) = \exp(\log(\lambda e^{-\lambda y}))$ This becomes $f(y; \lambda) = \exp(\log(\lambda) + \log(\exp(-\lambda y)))$ $= \exp(\log\lambda - \lambda y)$ This is indeed in the form of the exponential family.

2. (b) Why can't plants do math? Because it gives them square roots!

Let Y_i ~ exponential(λ) where $i \in \{1, \ldots, n\}$. Then $Z = \sum_{ni=1} Y_i$ ~ Gamma(n, λ). Show that Z is also a member of the exponential family.

Solution Gamma distribution has the following pdf: $f(y; n, \lambda) = {}^{1}\Gamma(n)\lambda^{n}y^{n-1}e^{-ny}$ This belongs in the exponential family if it takes the form: $f(y) = \exp({}^{(\lambda y) - b(\lambda)n(\phi)) + c(y, \phi))}$ Taking similar steps to 2(a), we have $f(y) = \exp(-\lambda y + n\log\lambda + (n-1)\log y - \log\Gamma(n))$ $= \exp({}^{-\lambda yn + n\log\lambda n1n} + (n-1)\log y - \log\Gamma(n))$

. .

substitute and simplify

. .

$= \exp({}^{\theta y - \log\theta - \phi + c(y, \theta))}$

Hence, Z belongs to the exponential family.

In [ ]: