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 [2]:
```
# Load required libraries
library(tidyverse)
library(dplyr)
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
Registered S3 method overwritten by 'rvest':
  method            from
  read_xml.response xml2
── Attaching packages ─────────────────────────────────────── tidyve
rse 1.2.1 ──
✔ ggplot2 3.1.1       ✔ purrr   0.3.2
✔ tibble  2.1.1       ✔ dplyr   0.8.0.1
✔ tidyr   0.8.3       ✔ stringr 1.4.0
✔ readr   1.3.1       ✔ forcats 0.4.0
── Conflicts ────────────────────────────────────────── tidyverse_co
nflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

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 (https://researchblog.duke.edu/2016/10/24/diabetes-and-privacy-meet-big-data/)](https://researchblog.duke.edu/2016/10/24/diabetes-and-privacy-meet-big-data/) 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.

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.

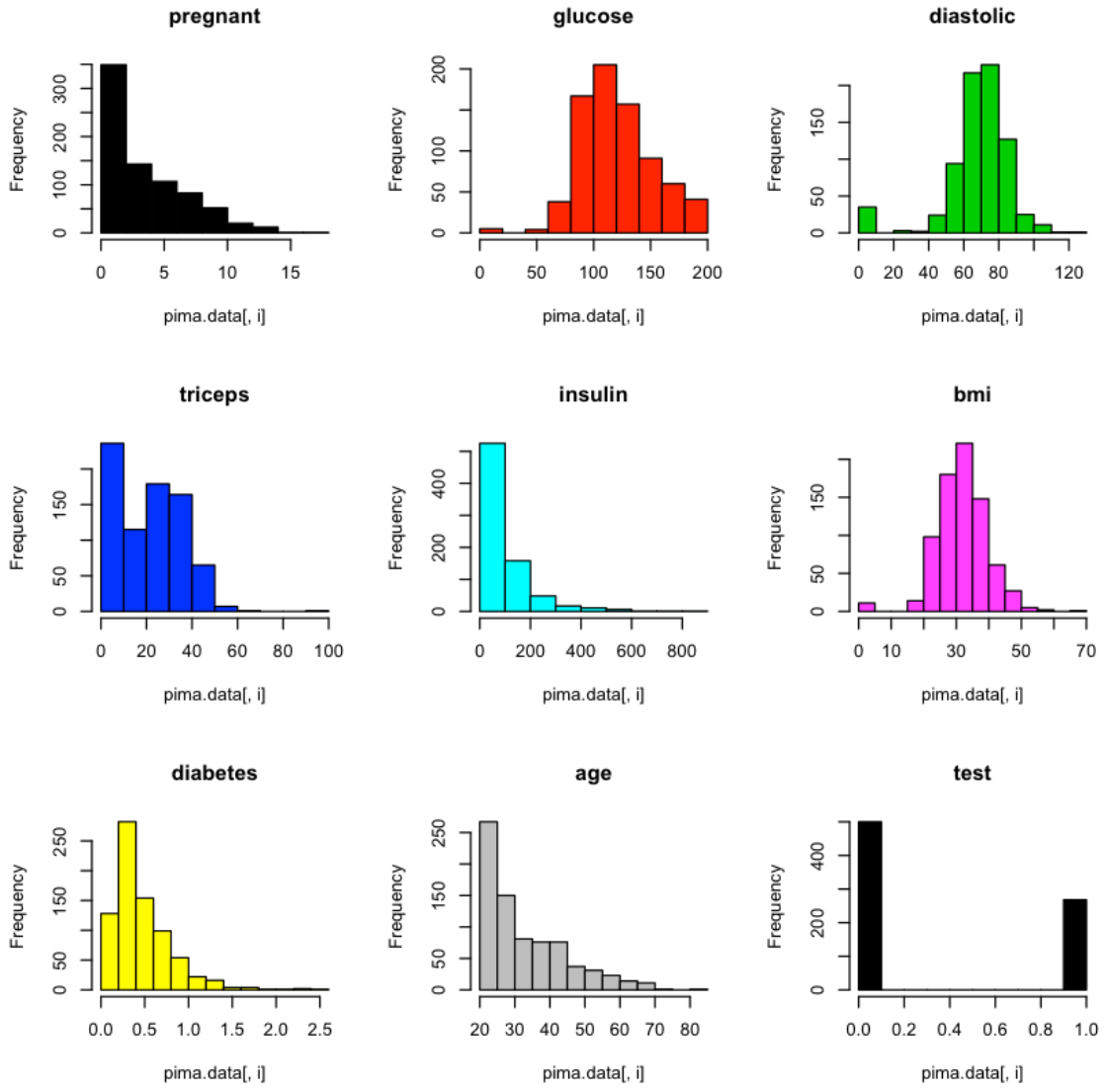
Finally, split your data into training and test sets. Let the training set contain 80% of the rows and the test set contain the remaining 20%.

In [32]:
```
pima.data = read.table("https://www.colorado.edu/amath/sites/default/files/attached-files/pima.txt",
                       sep = "\t", header = TRUE)
# Here's a description of the data: https://rdrr.io/cran/faraway/man/pima.html
```

In [33]:

```
# no missing values
sum(is.na(pima.data))

par(mfrow=c(3,3))
for (i in 1:9) hist(pima.data[,i], col = i, main = names(pima.dat
a)[i])
# histograms show weirdness -- glucose, diastolic, triceps, BMI, an
d insulin should never be zero
par(mfrow=c(1,1))

# recode zeros to NAs for values that can't be zero
metricTraits = c('glucose', 'diastolic', 'triceps', 'bmi', 'insulin
')
pima.data[metricTraits][pima.data[metricTraits]==0] = NA
pima.data = na.omit(pima.data)
pima.data = pima.data %>%
    mutate(test = as.factor(test))

summary(pima.data)

set.seed(1989)

n = floor(0.8 * nrow(pima.data))
index = sample(seq_len(nrow(pima.data)), size = n)

train = pima.data[index, ]
test = pima.data[-index, ]
#summary(train)
```

```
    pregnant         glucose        diastolic         triceps
 Min.   : 0.000   Min.   : 56.0   Min.   : 24.00   Min.   : 7.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:21.00
 Median : 2.000   Median :119.0   Median : 70.00   Median :29.00
 Mean   : 3.301   Mean   :122.6   Mean   : 70.66   Mean   :29.15
 3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00   3rd Qu.:37.00
 Max.   :17.000   Max.   :198.0   Max.   :110.00   Max.   :63.00
    insulin            bmi            diabetes           age         t
est
 Min.   : 14.00   Min.   :18.20   Min.   :0.0850   Min.   :21.00
0:262
 1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697   1st Qu.:23.00
1:130
 Median :125.50   Median :33.20   Median :0.4495   Median :27.00
 Mean   :156.06   Mean   :33.09   Mean   :0.5230   Mean   :30.86
 3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870   3rd Qu.:36.00
 Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
```



Some measurements are recorded as zero when clearly they shouldn't be (e.g., glucose). We should store these values as NA.

1. (b) Initial GLM modelling

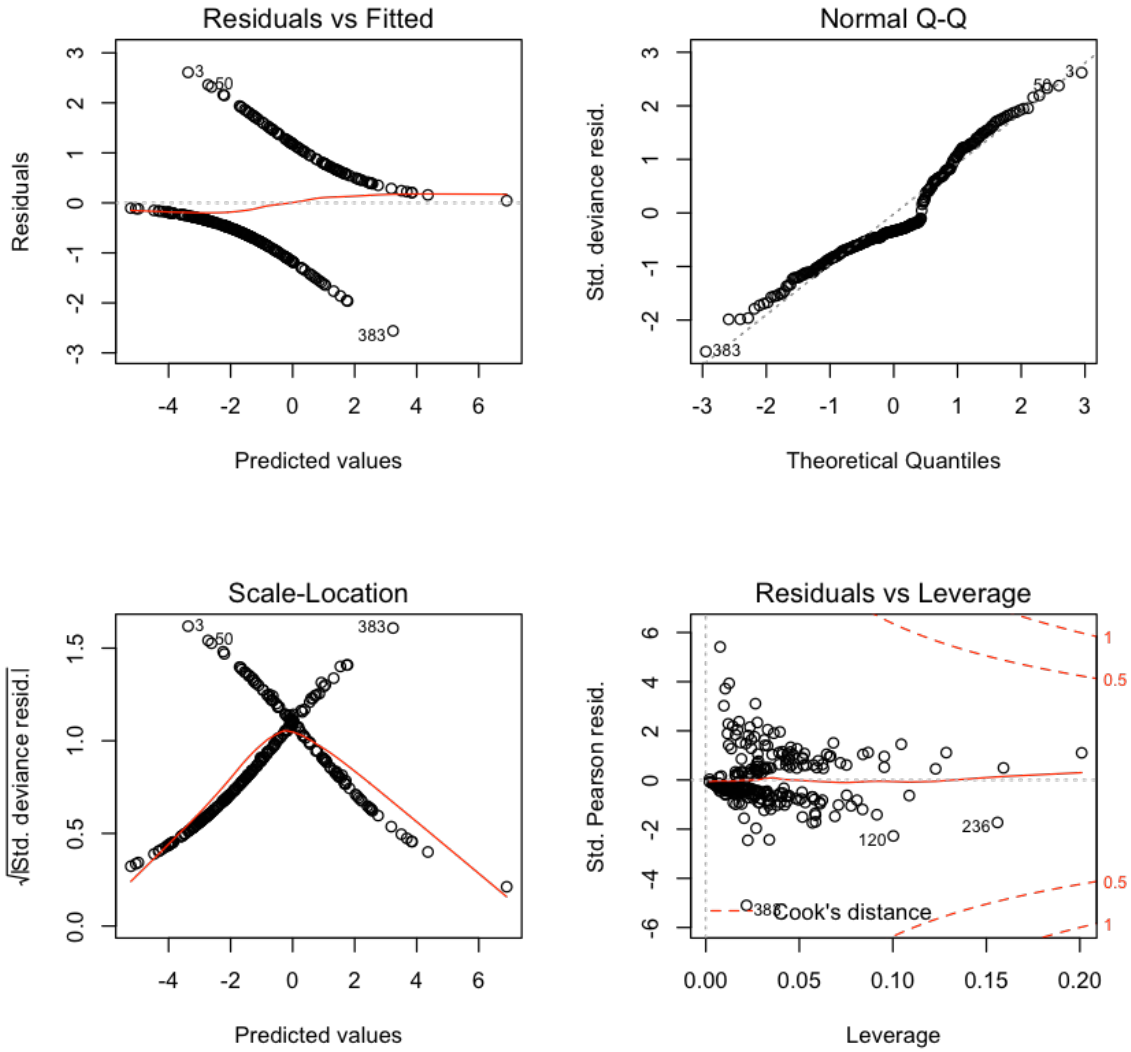
Our data is clean and we're ready to fit! What kind of model should we use to fit these data? Notice that the `test` variable is either 0 or 1, for whether the individual tested positive for diabetes. Because `test` is binary, we should use logistic regression (which is a kind of binomial regression).

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 [34]:

```
glmod_pima = glm(test ~ ., data = train, family = binomial)
summary(glmod_pima)

par(mfrow = c(2,2)); plot(glmod_pima)
```

```
Call:
glm(formula = test ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5593  -0.6437  -0.3396   0.5858   2.6094

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.086e+01  1.484e+00  -7.318 2.51e-13 ***
pregnant     9.844e-02  6.340e-02   1.553 0.120504
glucose      3.714e-02  6.700e-03   5.543 2.97e-08 ***
diastolic   -8.918e-03  1.414e-02  -0.631 0.528130
triceps      1.355e-03  1.965e-02   0.069 0.945017
insulin     -3.033e-04  1.414e-03  -0.214 0.830170
bmi          1.076e-01  3.265e-02   3.297 0.000976 ***
diabetes     1.815e+00  4.960e-01   3.659 0.000253 ***
age          3.624e-02  2.096e-02   1.729 0.083768 .
---
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: 266.25  on 304  degrees of freedom
AIC: 284.25

Number of Fisher Scoring iterations: 5
```



In the case where the response is binary, $Y = 0, 1$, as opposed to $Y = 0, 1, \ldots, n$, residuals won't fill a normal distribution and deviance will not follow a chisquared distribution, so we won't have any test for model fit. You might split the data into a training and test set, and see how well the model does at predicting values in the test set.

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 [35]:
```
#cor(glmod_pima$model)
lm_diastolic = lm(diastolic ~ test, data = train)
summary(lm_diastolic)
```

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

Residuals:
    Min      1Q  Median      3Q     Max 
-45.048  -8.250   0.952   7.750  36.952 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  69.0478     0.8416  82.046  < 2e-16 ***
test1         5.2022     1.4600   3.563 0.000424 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.17 on 311 degrees of freedom
Multiple R-squared:  0.03922,   Adjusted R-squared:  0.03613 
F-statistic:  12.7 on 1 and 311 DF,  p-value: 0.0004239
```

Women who test positive do have a higher diastolic blood pressure, on average. However, the coefficient for diastolic is not significant in the model. One is a question about the result of the test conditional on diastolic pressure; the other is a question about diastolic blood pressure conditional on a test result. We know from Bayes' theorem that these are not the same!

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 [36]:
```
summary(glmod_pima)
```

```
Call:
glm(formula = test ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5593  -0.6437  -0.3396   0.5858   2.6094

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.086e+01  1.484e+00  -7.318 2.51e-13 ***
pregnant     9.844e-02  6.340e-02   1.553 0.120504
glucose      3.714e-02  6.700e-03   5.543 2.97e-08 ***
diastolic   -8.918e-03  1.414e-02  -0.631 0.528130
triceps      1.355e-03  1.965e-02   0.069 0.945017
insulin     -3.033e-04  1.414e-03  -0.214 0.830170
bmi          1.076e-01  3.265e-02   3.297 0.000976 ***
diabetes     1.815e+00  4.960e-01   3.659 0.000253 ***
age          3.624e-02  2.096e-02   1.729 0.083768 .
---
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: 266.25  on 304  degrees of freedom
AIC: 284.25

Number of Fisher Scoring iterations: 5
```

Let p be the probability of a positive test. Then the model fitted is:

$$\begin{aligned}\eta = \log\left(\underbrace{\frac{\widehat{p}}{1-\widehat{p}}}_{\text{odds}}\right) &= \widehat{\beta}_0 + \widehat{\beta}_1 pregnant + \widehat{\beta}_2 glucose + \widehat{\beta}_3 diastolic + \widehat{\beta}_4 triceps + \widehat{\beta}_5 insulin + \widehat{\beta}_6 \\ &\quad + \widehat{\beta}_7 diabetes + \widehat{\beta}_8 age \\ &= -10 + 0.1 pregnant + 0.04 glucose - 0.009 diastolic + 0.001 triceps - 0.0003 i \\ &\quad + 0.02 bmi + 1.8 diabetes + 0.04 age\end{aligned}$$

So, we can interpret our model as follows:

Adjusting for other predictors, a one-unit increase in glucose levels increases the **log-odds** of a positive test by 0.04. Or, adjusting for other predictors, a one-unit increase in glucose levels increases the **odds** of success by a multiplicitive factor of $e^{0.04} \approx 1.04$.

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 [70]:
```
pr = ifelse(predict.glm(glmod_pima, type = "response", test, na.rm
= TRUE) > 0.5, 1, 0)
tn = sum(pr == 0 & as.numeric(levels(test$test))[test$test] == 0);
tp = sum(pr == 1 & as.numeric(levels(test$test))[test$test] == 1);
fp= sum(pr == 1 & as.numeric(levels(test$test))[test$test] == 0);
fn= sum(pr == 0 & as.numeric(levels(test$test))[test$test] == 1);

(tp+tn)/dim(test)[1]
```

Here's the confusion matrix for our predictions:

	True	False
1	17	7
0	9	46

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 [72]:
```
accuracy = (tp+tn)/dim(test)[1]; accuracy
precision = tp/(tp + fp); precision
recall = tp/(tp + fn)
F = (2*precision*recall)/(precision + recall)
```

0.79746835443038

0.708333333333333

0.68

The F score is a value between 0 and 1 and provides a way to combine both precision and recall into a single measure that captures both properties. This F score of 0.68 is reasonable.

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.

1. Consider a logistic regression model that *always* predicts a success. This model would have a high predictive accuracy on any data that had a relatively high number of successes. But this model wouldn't be doing any actual classifying. This is sometimes called the "accuracy paradox".
2. Here's an example of a confusion matrix for three levels:

	1	**2**	**3**		
1	17	7	1	4	7
2	9	46	0	9	46
3	5	8	15	9	46

Here, the diagonal entries show correct classifications, and the off diagonal entries show response measurements in category i classified as category j ($i, j = 1, 2, 3$).

1. One might argue that it would be better to have a classifier with higher type I errors. Type I errors/false positives in this case mean predicting a positive diabetes test when diabetes isn't present. In such cases, perhaps individuals who were misclassified in this way would need to undergo further screening that would correct the issue. This seems preferable to more false negatives, which would let diabetes go undetected and cause serious health problems.

1. (h) Ethical Issues in Data Collection

Read Maya Iskandarani's piece (https://researchblog.duke.edu/2016/10/24/diabetes-and-privacy-meet-big-data/) on consent and privacy concerns raised by this dataset. Summarize those concerns here.

Iskandarani's concerns about this dataset are related to privacy and consent in the age of big data. The original pima study, from which the data came, was meant to last 10 years but ended up lasting 40, and years later, was archived by the University of California Irvine Machine Learning Repository. This archiving made the pima dataset a "standard" dataset for training statistical and machine learning algorithms on. Those who signed up for the study never could have known that they were going to be signing over their data to be used in these ways.

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 $Y \sim exponential(\lambda)$, where λ is known, is a member of the exponential family.

Y is a random variable from the exponential family if the distribution (either pdf or pmf) can be written as:

$$f(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}.$$

Y is exponentially distributed if the pdf of Y is

$$f(y;\lambda) = \lambda e^{-\lambda y} = \exp\left(\log\left(\lambda e^{-\lambda y}\right)\right) = \exp(\log(\lambda) - \lambda y) = \exp\left(\frac{\lambda y - \log(\lambda)}{-1} + 0\right),$$

which is in the form of the exponential family of distributions.

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

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

$$f(y;n,\lambda) = \frac{\lambda^n}{\Gamma(n)} y^{n-1} e^{-\lambda y},$$

which is in the form of the exponential family of distributions....

In [ ]: