C3M2: Autograded Assignment

Outline:

Here are the objectives of this assignment:

1. Understand when to apply different kinds of regression models.
2. Fit a GLM to count data and go through model diagnostics and interpretation.
3. Compare the effectiveness of GLMs to Linear Regression models.

Here are some general tips:

1. Read the questions carefully to understand what is being asked.
2. When you feel that your work is completed, feel free to hit the `Validate` button to see your results on the *visible* unit tests. If you have questions about unit testing, please refer to the "Module 0: Introduction" notebook provided as an optional resource for this course. In this assignment, there are hidden unit tests that check your code. You will not recieve any feedback for failed hidden unit tests until the assignment is submitted. **Do not misinterpret the feedback from visible unit tests as all possible tests for a given question--write your code carefully!**
3. Before submitting, we recommend restarting the kernel and running all the cells in order that they appear to make sure that there are no additional bugs in your code.

In [1]:
```
# Load required packages
library(tidyverse)
library(testthat)
library(ggplot2)
```

```
── Attaching packages ─────────────────────────────────────── tidyverse
1.3.0 ──

✔ ggplot2 3.3.0     ✔ purrr   0.3.4
✔ tibble  3.0.1     ✔ dplyr   0.8.5
✔ tidyr   1.0.2     ✔ stringr 1.4.0
✔ readr   1.3.1     ✔ forcats 0.5.0

── Conflicts ────────────────────────────────────────── tidyverse_confl
icts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()


Attaching package: 'testthat'


The following object is masked from 'package:dplyr':

    matches


The following object is masked from 'package:purrr':

    is_null


The following object is masked from 'package:tidyr':

    matches
```

Problem 1: Counts, Rates and Measurements. (15 points)

As we've seen, there are many kinds of models for the many kinds of data out there, and fitting a good model start with understanding the data. For the following questions, determine which kind of model should be used for the specified data and question.

For each question, input the string answer of the specified model in the respective answer variable. Choose your answers from the models: `"linear"`, `"binomial"` and `"poisson"`, case sensitive. Note: Some features may be suitable for different kinds of models. Pick the model that would work the best.

1. You are trying to predict the number of home run scored by baseball players during their next season. Your predictors are the player's age, the number of years spent in professional baseball, and the number of home runs they scored in the previous 5 years.
2. You are trying to determine whether people in cities buy more cereal than people in suburbs or in rural areas. Your response is the number of cereal boxes sold, rounded to the nearest 1000. Your predictors are the type of area, the population, the number of grocery stores, and the average cost.
3. You want to predict ratings for hotels based on user reviews. The rating is on a scale of 1 to 5 stars. The predictors are different statistics extracted from their review, such as word count and the number of times the review used the word "bathroom."

In [17]:
```
# Remember, your answers should be strings
prob.1.1 = NA

prob.1.2 = NA

prob.1.3 = NA

# your code here
prob.1.1 = "poisson" #poisson

prob.1.2 = "linear" #binomial, linear, poisson

prob.1.3 = "binomial" #linear, binomial, poisson
```

In [5]:
```
# Test Cell
if(!test_that("Checking answer types", {expect_is(prob.1.1, "character")
                                        expect_is(prob.1.2, "character")
                                        expect_is(prob.1.3, "character"
)})){
    print("Make sure your answers are strings!")
}
```

In [ ]:

In [ ]:

Problem 2: MLRs vs. GLMs

For each 30 Galapagos islands, we have a count of the number of plant species found on each island and the number that are endemic to that island. We also have five geographic variables for each island.

1. Species: the number of plant species found on the island
2. Endemics: the number of endemic species
3. Area: the area of the island (km^2)
4. Elevation: the highest elevation of the island (m)
5. Nearest: the distance from the nearest island (km)
6. Scruz: the distance from Santa Cruz island (km)
7. Adjacent: the area of the adjacent island (square km)

In [18]:
```
# Load the data
data.gala = read.csv("gala.csv")

colnames(data.gala)[1] = "Location"
data.gala$Location = as.character(data.gala$Location)
head(data.gala)
```

A data.frame: 6 × 8

	Location	Species	Endemics	Area	Elevation	Nearest	Scruz	Adjacent
	<chr>	<int>	<int>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
1	Baltra	58	23	25.09	346	0.6	0.6	1.84
2	Bartolome	31	21	1.24	109	0.6	26.3	572.33
3	Caldwell	3	3	0.21	114	2.8	58.7	0.78
4	Champion	25	9	0.10	46	1.9	47.4	0.18
5	Coamano	2	1	0.05	77	1.9	1.9	903.82
6	Daphne.Major	18	11	0.34	119	8.0	8.0	1.84

2. (a) Trying a Linear Model (15 points)

Fit a linear model called `lmod.gala` with `Species` as the response and all other variables, except `Location` and `Endemics`, as predictors. Run some diagnostics and think about why this model may not be the best fit. For each assumption variable, answer `TRUE` if the assumptoin is being met by the model, and `FALSE` if the assumption is not being met by the model.

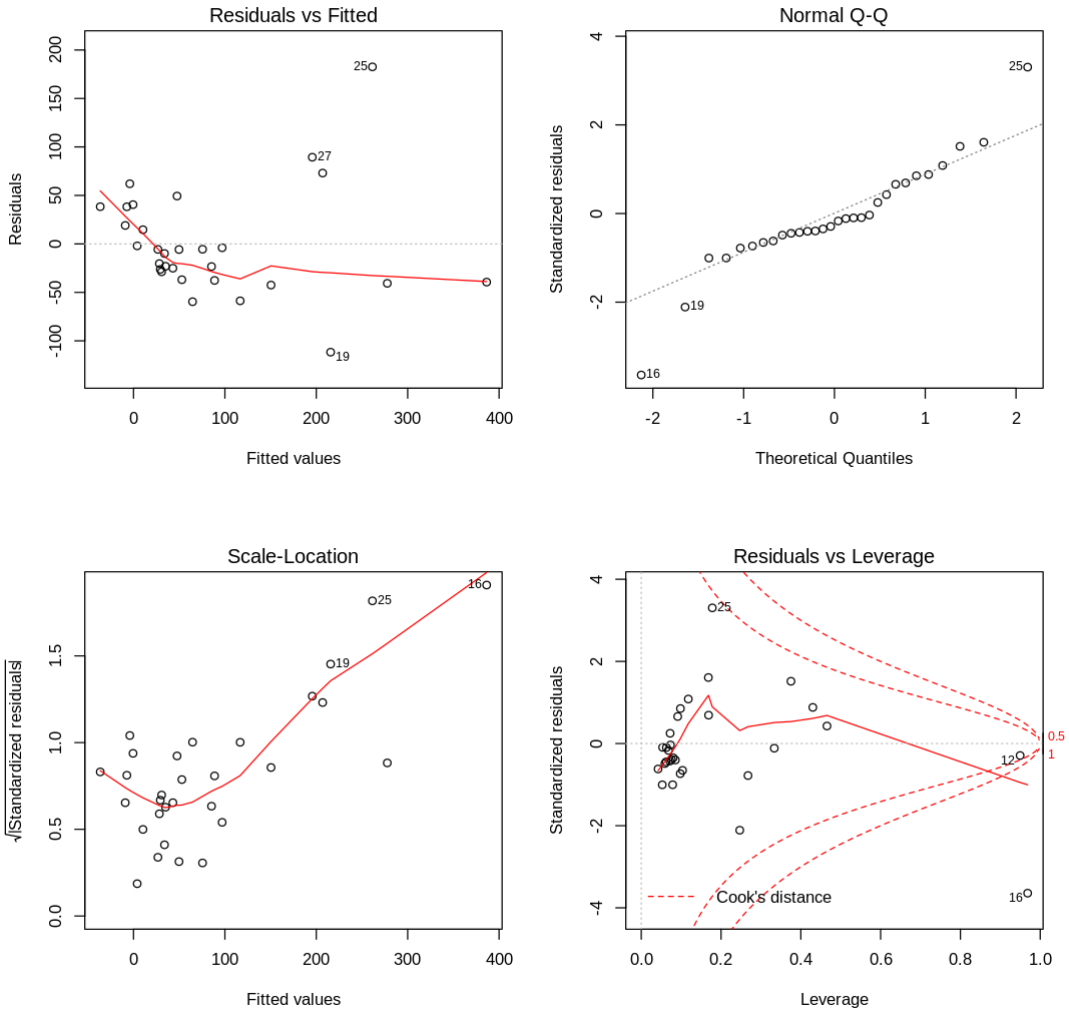
In [34]:
```
lmod.gala = NA
# Code the following as TRUE or FALSE
lmod.gala.linear = NA
lmod.gala.homoskedasticity = NA
lmod.gala.normality = NA

# your code here
lmod.gala = lm(Species ~ Area + Elevation + Nearest + + Scruz + Adjacent
, data= data.gala)
summary(lmod.gala)
```

```
Call:
lm(formula = Species ~ Area + Elevation + Nearest + +Scruz +
    Adjacent, data = data.gala)

Residuals:
     Min       1Q   Median       3Q      Max
-111.679  -34.898   -7.862   33.460  182.584

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.068221  19.154198   0.369 0.715351
Area        -0.023938   0.022422  -1.068 0.296318
Elevation    0.319465   0.053663   5.953 3.82e-06 ***
Nearest      0.009144   1.054136   0.009 0.993151
Scruz       -0.240524   0.215402  -1.117 0.275208
Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 60.98 on 24 degrees of freedom
Multiple R-squared:  0.7658,    Adjusted R-squared:  0.7171
F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

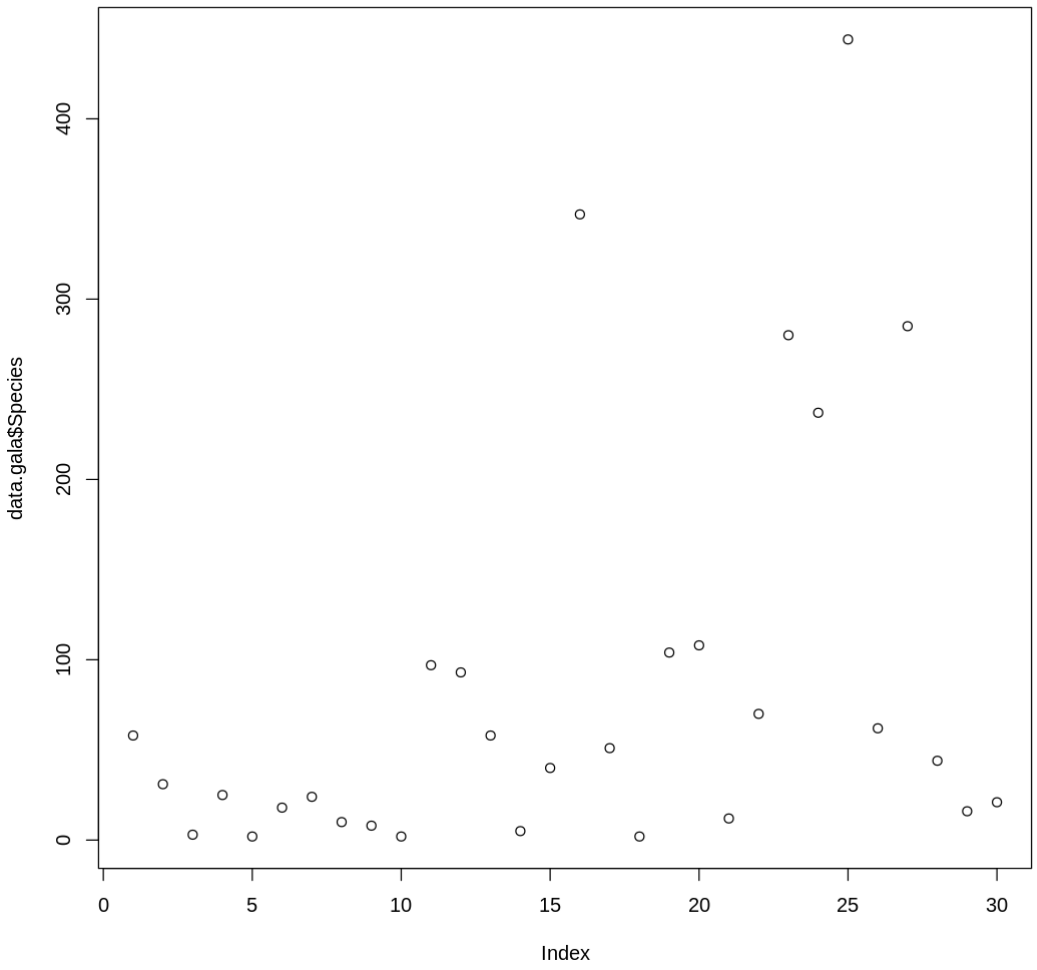
In [31]:
```
lmod.gala.linear = FALSE
lmod.gala.homoskedasticity = FALSE
lmod.gala.normality = FALSE
```

In [35]:
```
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow= c(2,2))
plot(lmod.gala)
```

```
Warning message in sqrt(crit * p * (1 - hh)/hh):
“NaNs produced”
Warning message in sqrt(crit * p * (1 - hh)/hh):
“NaNs produced”
```



In [36]:
```
plot(data.gala$Species)
```



In [37]:
```
lmod.gala$coef[1]
```

(Intercept): 7.06822070912069

In [38]:
```
# Test Cell
if(!test_that("Checking if model coefficients are correct", {expect_equa
l(7.068221, as.numeric(lmod.gala$coef[1]), tol=1e-4)
                                                              expect_equa
l(-0.023938, as.numeric(lmod.gala$coef[2]), tol=1e-4)
                                                              expect_equa
l(0.319456, as.numeric(lmod.gala$coef[3]), tol=1e-4)})){
    print("At least one of the coefficients was wrong. Make sure your mo
del is correct before doing diagnostics.")
}
# This cell has hidden test cases that will run after submission.
```

In [ ]:

In [ ]:

2. (b) Linear Transformations (8 points)

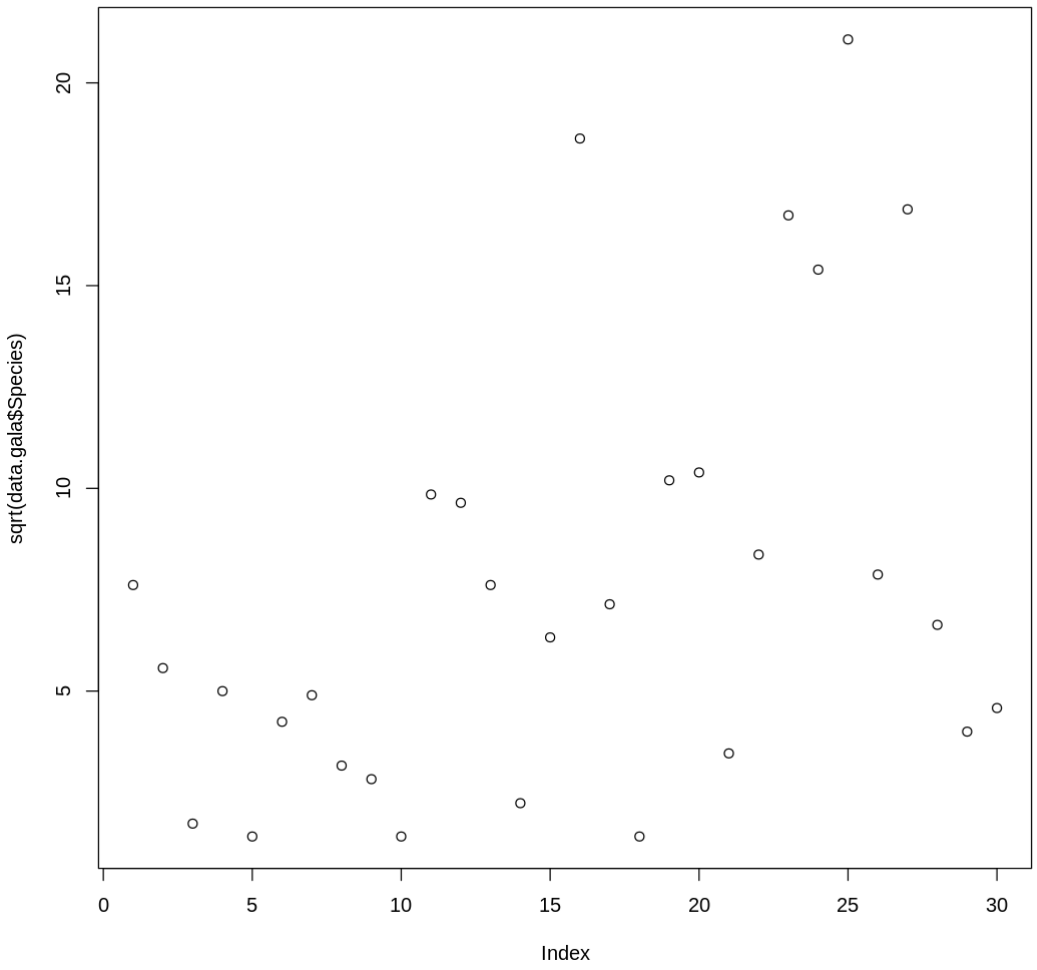
Recall that one strategy we used to address models that had nonconstant variance was to transform the response variable. Try the square root transform on the response fit to the same predictors. Store this model as `lmod.gala.sqrt`. Look at the diagnostic plots and consider if this model's assumptions are better than the last. Similar to the previous problem, for each assumption, answer `TRUE` if the model meets the assumption `FALSE` if not. Note that if a plot looks ambiguous, you can interpret it as "no evidence of a violation" and answer `TRUE`.

One thing to keep in mind is that transformations make the model harder to interpret. Think about how a 1 unit increase in `Nearest` for your transformed model would affect `Species`. Put your answers into `sqrt.gala.linearity`, `sqrt.gala.homoskedasticity` and `sqrt.gala.normality`.

In [39]:
```
lmod.gala.sqrt = NA
sqrt.gala.linearity = NA
sqrt.gala.homoskedasticity = NA
sqrt.gala.normality = NA

# your code here
lmod.gala.sqrt = lm(sqrt(Species) ~ Area + Elevation + Nearest + + Scruz
+ Adjacent, data= data.gala)
summary(lmod.gala.sqrt)
```

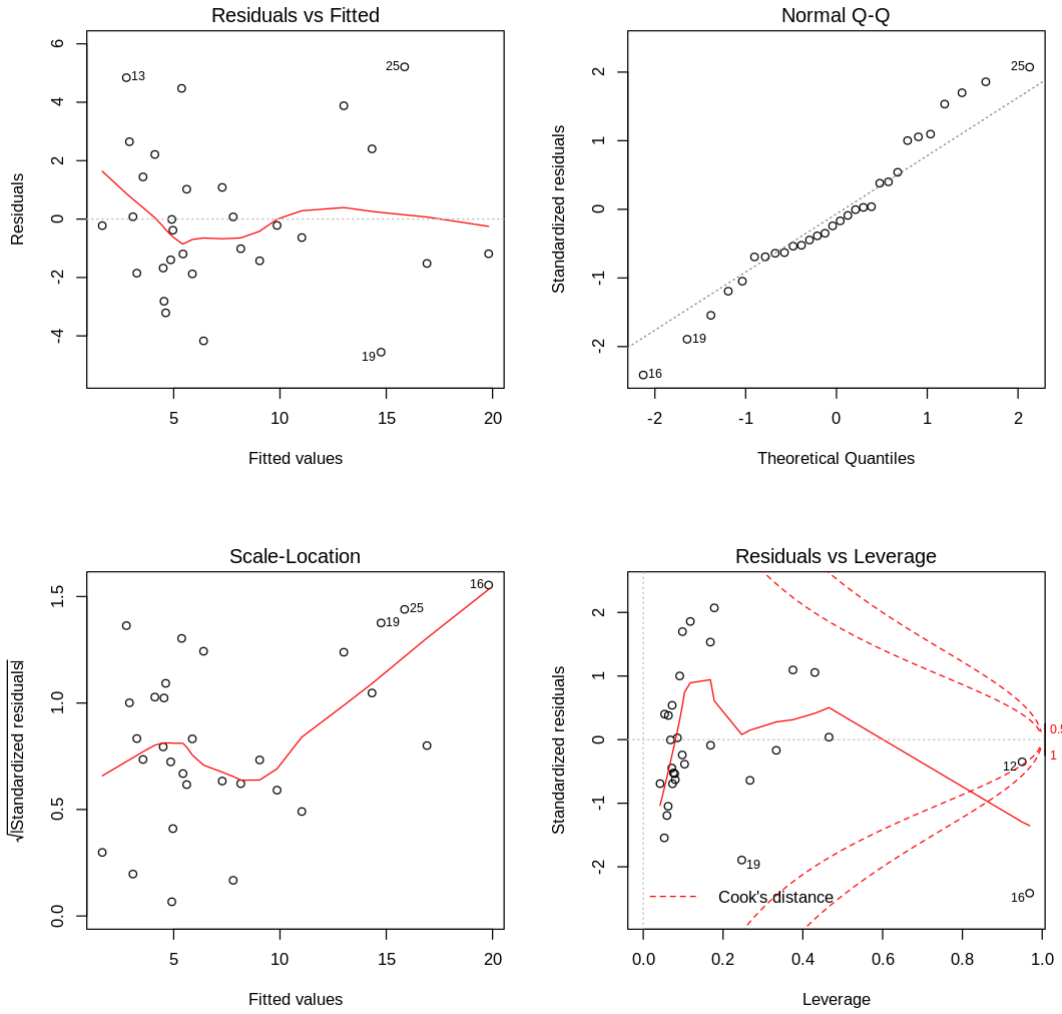
```
Call:
lm(formula = sqrt(Species) ~ Area + Elevation + Nearest + +Scruz +
    Adjacent, data = data.gala)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5572 -1.4969 -0.3031  1.3527  5.2110

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.3919243  0.8712678   3.893 0.000690 ***
Area        -0.0019718  0.0010199  -1.933 0.065080 .
Elevation    0.0164784  0.0024410   6.751 5.55e-07 ***
Nearest      0.0249326  0.0479495   0.520 0.607844
Scruz       -0.0134826  0.0097980  -1.376 0.181509
Adjacent    -0.0033669  0.0008051  -4.182 0.000333 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.774 on 24 degrees of freedom
Multiple R-squared:  0.7827,    Adjusted R-squared:  0.7374
F-statistic: 17.29 on 5 and 24 DF,  p-value: 2.874e-07
```

In [41]:
```
plot(sqrt(data.gala$Species))
```



In [42]:
```
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow = c(2,2))
plot(lmod.gala.sqrt)
```

```
Warning message in sqrt(crit * p * (1 - hh)/hh):
“NaNs produced”
Warning message in sqrt(crit * p * (1 - hh)/hh):
“NaNs produced”
```



In [46]:
```
sqrt.gala.linearity = FALSE
sqrt.gala.homoskedasticity = TRUE #CORRECT
sqrt.gala.normality = FALSE #CORRECT
```

In [47]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

In [ ]:

2. (c) GLMs to the Rescue (6 points)

There's still some problems with the model. Because our response variable is counts, maybe linear models aren't the best anyways. Fit a GLM of appropriate family to the (untransformed) data, using the same predictors. Store this model as `glm.gala`. Plot the diagnostics plots and think about what assumptions should be met.

How do we interpret this model? In particular, fill in the blank: "A 1-unit increase in `Elevation` is associated with a multiplicative increase of _____ in `Species`, on average." Store this value as `glm.interp`.

In [57]:
```
glm.gala = NA
glm.interp = NA

# your code here
glm.gala = glm(Species ~ Area + Elevation + Nearest + + Scruz + Adjacent
, data= data.gala,
               family=poisson)
summary(glm.gala)
glm.interp = 1.00354059400884341
```

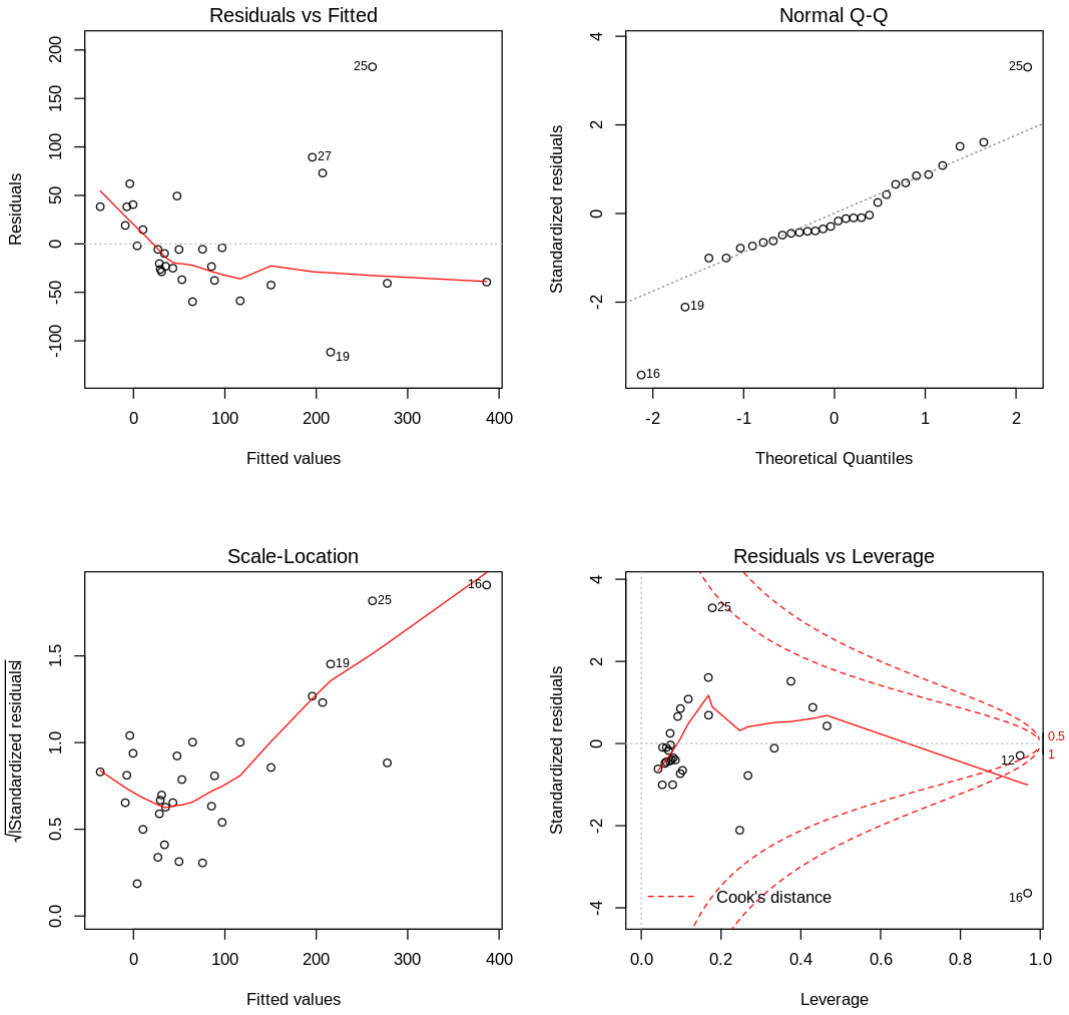
```
Call:
glm(formula = Species ~ Area + Elevation + Nearest + +Scruz +
    Adjacent, family = poisson, data = data.gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.2752  -4.4966  -0.9443   1.9168  10.1849

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

In [56]:
```
coef(glm.gala)[3]
```

Elevation: 0.00354059400884341

In [50]:
```
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow= c(2,2))
plot(lmod.gala)
```

```
Warning message in sqrt(crit * p * (1 - hh)/hh):
“NaNs produced”
Warning message in sqrt(crit * p * (1 - hh)/hh):
“NaNs produced”
```



In [ ]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

In [ ]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

In [ ]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

2. (d) GLM Goodness of Fit (6 points)

Our linear models didn't do a great job of fitting the data, how do we know if our GLM fits the data any better? Well, we don't have an easy scale of reference, like the R^2 value, for GLMs. What we can do is compare our model to other models, such as the null model, and see if ours performs significantly better.

Calculate the deviance of your model and store it as `glm.deviance`. Then check the goodness of fit of your model using Pearson's χ^2 statistic. Store this value as `glm.chisq.stat`. Calculate the p-value for this statistic and store it as `glm.chisq.pval`. What does this tell you about your model?

In [67]:
```
glm.deviance = NA
glm.chisq.stat = NA
glm.chisq.stat = NA

# your code here
glm.deviance = glm.gala$deviance
```

1

In [83]:
```
qchisq(0, df=24)
```

0

In [99]:
```
glm.chisq.stat = sum((data.gala$Species - fitted(glm.gala))^2 / fitted(g
lm.gala))
glm.chisq.stat
```

761.979247761282

In [100]:
```
pchisq(glm.chisq.stat, df=glm.gala$df.resid, lower.tail=FALSE)
```

2.1871899185534e-145

In [82]:
```
glm.chisq.pval = pchisq(glm.gala$deviance, df=glm.gala$df.resid, lower.t
ail=FALSE)
glm.chisq.pval
```

7.07315731341498e-136

In [ ]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

In [ ]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

In [ ]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```