4.2 Case Study 1: GLMs for Continuous Target Variables

In the first case study of this chapter we will revisit the personal injury insurance dataset we first saw in Section 2.2 in the context of data exploration. The preparatory work in Section 2.2 allows us to get straight to building and comparing different GLMs, which are the core activities of this chapter, and the small number of variables in the dataset makes feature selection less of a problem so that we can concentrate on issues specific to GLMs. After completing this case study, you should be able to:

- Select appropriate distributions and link functions for a positive, continuous target variable with a right skew.
- Fit a GLM using the `glm()` function in R and specify the options of this function appropriately.
- Make predictions for GLMs using the `predict()` function and compare the predictive performance of different GLMs.
- Generate and interpret diagnostic plots for a GLM.

4.2.1 Preparatory Steps

Background. Recall from Section 2.2 that the personal injury dataset `persinj` contains the information of $n = 22,036$ settled personal injury insurance claims which were reported during the period from July 1989 to the end of 1999. Claims settled with zero payment were not included. The data dictionary for the dataset is reproduced in Table 4.2 for your convenience.

Variable	Description
`amt`	settled claim amount (continuous numeric variable)
`inj`	injury code, with seven levels: 1 (no injury), 2, 3, 4, 5, 6 (fatal), 9 (not recorded)
`legrep`	legal representation (0 = no, 1 = yes)
`op_time`	operational time (a standardized amount of time elapsed between the time when the injury was reported and the time when the claim was settled)

Table 4.2: Data dictionary for the `persinj` dataset (reproduced from Table 2.2).

Our objective here is to build GLMs to predict the size of settled claims using related risk factors, select the most promising GLM, and quantify its predictive accuracy. For claim size variables, which are continuous, positive-valued and often highly skewed, common modeling options include:

- Apply a log transformation to claim size and fit a normal linear model to the log-transformed claim size.
- Build a GLM with the normal distribution and a link function such as the log link to ensure that the target mean is non-negative.

- Build a GLM with a continuous target distribution that is positive-valued and can capture the skewness of claim size. The gamma and inverse Gaussian distributions are reasonable choices of the target distribution in this regard.

We will explore all of these modeling options in this section and *(spoiler alert!)* show that a gamma GLM outperforms traditional models based on the normal distribution.

TASK 1: Data pre-processing

Decide which numeric variables in the data, if any, should be treated as factors. Do the factor conversion.

To begin with, run CHUNK 1 to load the `persinj` data and print a summary.

```
# CHUNK 1
persinj <- read.csv("persinj.csv")
summary(persinj)

##       amt                inj           legrep          op_time
##  Min.   :     10   Min.   :1.00   Min.   :0.0000   Min.   : 0.10
##  1st Qu.:   6297   1st Qu.:1.00   1st Qu.:0.0000   1st Qu.:23.00
##  Median :  13854   Median :1.00   Median :1.0000   Median :45.90
##  Mean   :  38367   Mean   :1.83   Mean   :0.6366   Mean   :46.33
##  3rd Qu.:  35123   3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.:69.30
##  Max.   :4485797   Max.   :9.00   Max.   :1.0000   Max.   :99.10
```

All of the four variables in the data are treated as numeric variables in R (that's why the summary for each variable prints out the numeric statistics). However, the `inj` variable is merely a label of group membership indicating the injury group to which each policy belongs. Even if there is an order among the injury levels from 1 to 6 according to the data dictionary, treating `inj` as a numeric variable, with a single regression coefficient attached, implicitly implies that the change in the linear predictor is the same for every consecutive change in `inj` (i.e., from 1 to 2, from 2 to 3, etc.), which can be a severe restriction. It is therefore important to treat `inj` as a factor (equivalently, a categorical predictor) to be represented by dummy variables in a GLM. In CHUNK 2, we make the factor conversion for `inj`. (You can also make the binary variable `legrep` a factor. Whether it is a numeric variable or a factor may affect the way graphical displays are produced, but will have no effect on the results of a GLM.)

```
# CHUNK 2
persinj$inj <- as.factor(persinj$inj)
summary(persinj)

##       amt          inj          legrep          op_time
##  Min.   :     10   1:15638   Min.   :0.0000   Min.   : 0.10
##  1st Qu.:   6297   2: 3376   1st Qu.:0.0000   1st Qu.:23.00
##  Median :  13854   3: 1133   Median :1.0000   Median :45.90
##  Mean   :  38367   4:  189   Mean   :0.6366   Mean   :46.33
##  3rd Qu.:  35123   5:  188   3rd Qu.:1.0000   3rd Qu.:69.30
##  Max.   :4485797   6:  256   Max.   :1.0000   Max.   :99.10
##                    9: 1256
```

After the factor conversion, the summary for `inj` prints out a frequency table instead of the numeric statistics. Since the baseline level (1) has the most observations, there is no need to relevel `inj`.

4.2.2 Model Construction and Evaluation

Setting up the training and test sets. Before we fit and evaluate any GLMs, we should establish the training (75%) and test sets (25%) with the aid of the `createDataPartition()` function, as in CHUNK 3.

```
# CHUNK 3
library(caret)
set.seed(2019)
partition <- createDataPartition(y = persinj$amt, p = .75, list = FALSE)
train <- persinj[partition, ]
test <- persinj[-partition, ]
summary(train$amt)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      20    6297   13852   38670   35118 4485797

summary(test$amt)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      10    6299   13858   37458   35140 1450818
```

We can see that the maximum claim sizes of the two sets differ considerably even with the use of stratified sampling and the two means are not as close to each other as we expect. For a highly skewed target variable like `amt`, its mean can be easily distorted by just a small number of extreme observations and it is more reliable to look at a more robust central tendency measure like median. The two medians, 13852 and 13858, are very close to each other. As usual, we will fit our models on the training set and evaluate their predictive power on the test set with respect to the RMSE, which is an appropriate metric for a numeric target variable like `amt`.

TASK 2: Fitting a linear model as a benchmark

Run an ordinary least squares (OLS) model for claim size or a transformation of claim size on the training set and calculate its test RMSE. This provides a benchmark for further model development.

Benchmark model: Normal linear model on log of claim size. Recall from Figures 2.2.2 and 2.2.3 in Section 2.2 that the distribution of `amt` is highly right skewed, having mostly low values and a few extremely high values, and that the log of the claim size is approximately bell-shaped and has a distribution close to normal. As a benchmark model for all of the GLMs we are going to construct, our first candidate model is therefore a linear model on the log of the claim size fitted using ordinary least squares (OLS). In view of Figure 2.2.7, it seems plausible that legal representation and operational time interact with each other to affect `amt`, so we will incorporate an interaction term for these two predictors.

In CHUNK 4, we use the `glm()` function to fit the OLS linear model to the training set. As you can guess, the `glm()` function is for fitting GLMs (while the `lm()` function is for linear models). Its syntax closely resembles that of `lm()`, but in addition to the model formula and the data frame hosting your variables, we also have to specify the family of distributions to which the target variable belongs as well as the link function to be used (if it is not the default link) via the command of the form `family = <family>(link = <"link">)`.

```
# CHUNK 4
glm.ols <- glm(log(amt) ~ inj + legrep * op_time,
               family = gaussian(link = "identity"), data = train)
summary(glm.ols)

##
## Call:
## glm(formula = log(amt) ~ inj + legrep * op_time, family = gaussian(link =
"identity"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.4217  -0.5805   0.0631   0.6688   4.4581
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.5212622  0.0269021 279.579  < 2e-16 ***
## inj2            0.5912759  0.0237858  24.858  < 2e-16 ***
## inj3            0.8161053  0.0385945  21.146  < 2e-16 ***
## inj4            0.7958558  0.0901834   8.825  < 2e-16 ***
## inj5            0.6462279  0.0894089   7.228 5.12e-13 ***
## inj6            0.3823662  0.0777895   4.915 8.95e-07 ***
## inj9           -0.8151999  0.0367039 -22.210  < 2e-16 ***
## legrep          0.9120834  0.0339725  26.848  < 2e-16 ***
## op_time         0.0358738  0.0005093  70.431  < 2e-16 ***
## legrep:op_time -0.0101808  0.0006384 -15.946  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.146774)
##
##     Null deviance: 35056  on 16527  degrees of freedom
## Residual deviance: 18942  on 16518  degrees of freedom
## AIC: 49180
##
## Number of Fisher Scoring iterations: 2
```

Note the following items in the code:

- The June 2019 PA exam Rmd model solution says that "[i]t is a good idea to use a descriptive model name for each new model created." In this case study, we will name each model with

a suffix that indicates the type of model being fitted.

- In the formula, the `log()` function is applied to `amt` as the log of the claim size is being modeled by the predictors.
- The shorthand `legrep * op_time` incorporates both the main effects and the interaction effect, as we learned in Subsection 3.3.2.
- If you shorten the command `family = gaussian(link = "identity")` to `family = gaussian`, the fitting still works fine because the default link function of the Gaussian distribution is the identity link. In fact, the default value of the `family` argument of the `glm()` function is `gaussian`, so if you like you may omit `family = gaussian(link = "identity")` altogether.

Although you can also use the `lm()` function to fit the OLS linear model (which is a "linear model"), the `summary()` function applied to a GLM will return different kinds of model statistics that are more useful for comparing different GLMs. Instead of showing the R^2 and F-statistic, which are specific to linear models, the model summary above shows the deviance (as shown in `Residual deviance`) and AIC of the OLS linear model, besides the coefficient estimates, standard errors, and p-values.

Now let's use the fitted OLS model to make predictions on the test set and calculate the test RMSE. Notice that because the `predict()` function produces predictions on the scale of the target variable `log(amt)`, which is on the log scale in this case, these predictions must be exponentiated to put them back on the scale of the original claim size. This is done in CHUNK 5.

```
# CHUNK 5
# define the rmse() function again for later use
rmse <- function(observed, predicted) {
  sqrt(mean((observed - predicted)^2))
}

pred.ols <- exp(predict(glm.ols, newdata = test))
head(pred.ols)

##         3        10        13        21        24        27
## 1853.5345 1853.5345 4609.6856  820.2833  820.2833  820.2833

rmse(test$amt, pred.ols)

## [1] 72888.33

rmse(test$amt, mean(train$amt))

## [1] 79370.9
```

As a check, the first prediction on the scale of `amt` for the first observation in the test set (which is the third observation in the `persinj` dataset), which has `inj = 1`, `legrep = 0`, and `op_time = 0.1`, is

$$\exp[7.5212622 + 0 + 0.0358738(0.1) + 0] = 1,853.53,$$

which is the same as the first element in the vector `pred.ols`.

To get a feel for how the OLS model performs, we can compare its test RMSE, which is 72,888.33, with the test RMSE of the intercept-only model fitted to the training set, which uses the mean of `amt` on the training set (not test set!) as prediction. The test RMSE of the latter model is 79,370.9, so the OLS model represents a decent improvement over the intercept-only model.

TASK 3: Select a distribution and link function

Evaluate three potential combinations of distribution and link function for applying a GLM to the training dataset. (Typing ?family in the Console will provide help. Included are the combinations that can be used with the glm function.) Explain, prior to fitting the models, why your choices are reasonable for this problem. Fit the three models and select the best combination, justifying your choice. Use only that model in subsequent tasks.

For this task, we will propose three combinations of distribution and link function and calculate the test RMSE of each of the three GLMs (provided that the model can be fitted).

GLM 1: Normal GLM on claim size with a log link. Instead of fitting a linear model for the log-transformed claim size, we can consider a normal GLM that takes the claim size itself as the target variable and uses the log link to connect the mean claim size to the linear predictor. As discussed on page 221, this alternative model differs from the OLS model and is a genuine GLM rather than a linear model. This GLM ensures positive predictions, but a drawback is that it allows for the possibility that the observations of the target variable are negative.

EXAM NOTE

The June 2019 PA exam model solution says that

> "[t]he Gaussian distribution admits negative observations. It was an acceptable choice provided candidates *noted this possibility*."

Run CHUNK 6 to fit the normal GLM with the log link and calculate its test RMSE. Note that when the log link is applied, there is no longer a need to log-transform the target variable.

```
# CHUNK 6
glm.log <- glm(amt ~ inj + legrep * op_time,
               family = gaussian(link = "log"), data = train)
summary(glm.log)

##
## Call:
## glm(formula = amt ~ inj + legrep * op_time, family = gaussian(link = "log"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -448809   -17687    -5013     2628  4203835
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.559238   0.120163  71.230  < 2e-16 ***
## inj2           0.572982   0.031018  18.473  < 2e-16 ***
## inj3           0.825502   0.034683  23.801  < 2e-16 ***
## inj4           0.883796   0.057430  15.389  < 2e-16 ***
## inj5           1.531415   0.034374  44.552  < 2e-16 ***
## inj6           1.506266   0.042987  35.040  < 2e-16 ***
## inj9          -0.552178   0.146506  -3.769 0.000164 ***
## legrep         0.068766   0.138410   0.497 0.619318
## op_time        0.027185   0.001508  18.029  < 2e-16 ***
## legrep:op_time 0.003195   0.001721   1.857 0.063383 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6897153068)
##
##     Null deviance: 1.4770e+14  on 16527  degrees of freedom
## Residual deviance: 1.1393e+14  on 16518  degrees of freedom
## AIC: 421348
##
## Number of Fisher Scoring iterations: 8

pred.log <- predict(glm.log, newdata = test, type = "response")
head(pred.log)

##        3       10       13       21       24       27
## 5228.901 5228.901 5602.912 3010.251 3010.251 3010.251

rmse(test$amt, pred.log)

## [1] 69623.89
```

In the code above:

- The option `family = gaussian(link = "log")` in the `glm()` function explicitly specifies the normal (a.k.a. Gaussian) target distribution and the log link.
- By default, the `predict()` function generates predictions for the GLM on the scale of the linear predictor, i.e., the log scale, due to the use of the log link. To transform the predictions to the original scale of the target variable, we include the option `type = "response"`. This obviates the need for explicitly exponentiating the predictions. To check that the predictions are indeed on the right scale, the prediction for the first observation in the test set is

$$\exp[8.559238 + 0 + 0.027185(0.1) + 0] = 5,228.90,$$

  which is the same (up to rounding errors) as the first element in the vector `pred.log`.

To our delight, the test RMSE decreases to 69,623.89 and suggests that the normal GLM with log link outperforms the OLS linear model in terms of prediction accuracy.

GLM 2: Gamma GLM with a log link. We now completely do away with the normal distribution and adopt a distribution that aligns with the characteristics of the target variable, amt, which is a positive, continuous variable with a right skew. In this regard, the gamma distribution is an appropriate choice. Although the canonical link for the gamma distribution is the inverse link $g(\mu) = 1/\mu$, the log link is more commonly used as it not only ensures that all predictions are positive, a characteristic of the target variable, but also makes the model coefficients easy to interpret—the coefficients are closely related to multiplicative changes to the linear predictor, as we discussed on page 225. Thus we will use a gamma GLM with a log link.

EXAM NOTE

The June 2019 PA exam model solution says that when asked to select a link function for a target variable that is positive, continuous, and right-skewed,

> "[m]any candidates went with the canonical link function just because it is canonical. That is a *weak justification.*"

If the link function you select is the canonical link, you can use this fact as an *additional* reason to support your choice, but just because a link is the canonical link does not necessarily make it a good choice. There are more important factors to consider, such as interpretability and whether or not the link function guarantees reasonable predictions.

Now run CHUNK 7 to train a gamma GLM with the log link (notice the option `family = Gamma(link = "log")` with the letter G capitalized), produce predictions on the test set, and calculate the test RMSE.

```
# CHUNK 7
glm.gamma <- glm(amt ~ inj + legrep * op_time,
                 family = Gamma(link = "log"), data = train)
summary(glm.gamma)

##
## Call:
## glm(formula = amt ~ inj + legrep * op_time, family = Gamma(link = "log"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4024  -0.9097  -0.3677   0.1842   8.1978
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.2056640  0.0331403 247.604  < 2e-16 ***
## inj2          0.5975174  0.0293014  20.392  < 2e-16 ***
## inj3          0.8568408  0.0475440  18.022  < 2e-16 ***
## inj4          1.1029945  0.1110956   9.928  < 2e-16 ***
## inj5          1.5036755  0.1101415  13.652  < 2e-16 ***
```

```
## inj6           0.9004443  0.0958278   9.396  < 2e-16 ***
## inj9          -0.6321655  0.0452150 -13.981  < 2e-16 ***
## legrep         0.4393948  0.0418502  10.499  < 2e-16 ***
## op_time        0.0344292  0.0006275  54.871  < 2e-16 ***
## legrep:op_time -0.0049104  0.0007865  -6.243 4.38e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.740277)
##
##     Null deviance: 33141  on 16527  degrees of freedom
## Residual deviance: 16705  on 16518  degrees of freedom
## AIC: 365833
##
## Number of Fisher Scoring iterations: 8

rmse(test$amt, predict(glm.gamma, newdata = test, type = "response"))

## [1] 68831.79
```

It is gratifying to see that the test RMSE drops further to 68,831.79 compared with the normal GLM with the log link. This shows that the gamma GLM is the most predictive among the models we have fitted with respect to test RMSE.

GLM 3: Inverse Gaussian GLM. An alternative to a gamma GLM is an inverse Gaussian GLM. The inverse Gaussian distribution often has a fatter tail than a gamma distribution. Its canonical link function is the inverse square link $g(\mu) = 1/\mu^2$, which ensures positive predictions but is not easy to interpret, so the log link is also commonly used.

In CHUNK 8, we try to fit an inverse Gaussian GLM with a log link.

```
# CHUNK 8
glm.ig <- glm(amt ~ inj + legrep + op_time, data = train,
              family = inverse.gaussian(link = "log"))

## Warning:  step size truncated due to divergence
## Warning:  step size truncated due to divergence
## Error:  inner loop 1; cannot correct step size
```

It turns out that the fitting cannot be completed due to convergence issues.[v] This is not a mistake on our part—there was also no convergence when an inverse Gaussian GLM was fitted in the June 2019 PA exam! To make the fitting work, we would have to tweak the optimization procedure (remember that GLMs parameters are estimated by maximum likelihood, which involves optimization), which is likely beyond the scope of Exam PA.[vi] As the inverse Gaussian GLM cannot be evaluated, we will stick to the gamma GLM developed earlier and use it for subsequent tasks.

[v] The fitting algorithm also fails to converge even with the use of the canonical link, the inverse square link.

[vi] There are some very brief discussions on non-convergence on Slide 152 of PA Module 6.

EXAM NOTE

The June 2019 PA exam Rmd model solution suggests commenting out the code which produces non-convergence. This is to ensure that your Rmd file can be run from beginning to end, as PA graders expect.

4.2.3 Model Validation

TASK 4: Validate the model

Compare the recommended model from Task 3 to the OLS model from Task 2. Also provide and interpret diagnostic plots for the recommended model to check the model assumptions.

Gamma GLM vs. benchmark OLS model. From CHUNK 7, the test RMSE of the gamma GLM with the log link is 68,831.79, which not only is lower than that of the OLS model (72,888.33), but also is the lowest among all of the models we have fitted. The summary analysis of the gamma GLM shows that all coefficient estimates, with an extremely small p-value, are statistically significant, so it is wise to retain all existing features in the model.

Model diagnostics. As a final check, in CHUNK 9 we make some diagnostic plots for the gamma GLM (see Figure 4.2.1) to see if the model assumptions are largely satisfied. The results are not impeccable though.

- The "Residuals vs Fitted" plot here graphs the standardized *deviance* residuals (as opposed to the standardized residuals for linear models), which are the deviance residuals scaled by their standard error, of the gamma GLM against the fitted values on the training set. The residuals mostly scatter around 0 in a structure-less manner, but with a sizable amount of fluctuation. Also notable is that the spread of the positive residuals tends to be larger than that of the negative residuals, with a few outliers having an unusually positive residual (observations 5422, 10224, and 18870).

- The Q-Q plot allows us to assess the normality of the standardized deviance residuals, with a straight line passing through the 25th and 75th theoretical percentiles shown for reference (in contrast to the 45° straight line for linear models). The plot looks fine in the middle portion and on the left end, but seems problematic on the right end, with the points deviating quite markedly from the reference line. The same three observations, 5422, 10224, and 18870, account for the significant departure from normality on the right end. The fact that the rightmost points deviate significantly upward means that there are a lot more large, positive standardized deviance residuals than under a normal distribution, and by extension, the distributions of the deviance residuals and the data are more skewed than the gamma distribution. It appears that a fatter-tailed model may perform better.

  Note that the label of the vertical axis of the Q-Q plot is "Std. deviance resid" rather than "Standardized residuals" (the latter is observed in Figure 3.4.6).

```
# CHUNK 9
plot(glm.gamma)
```

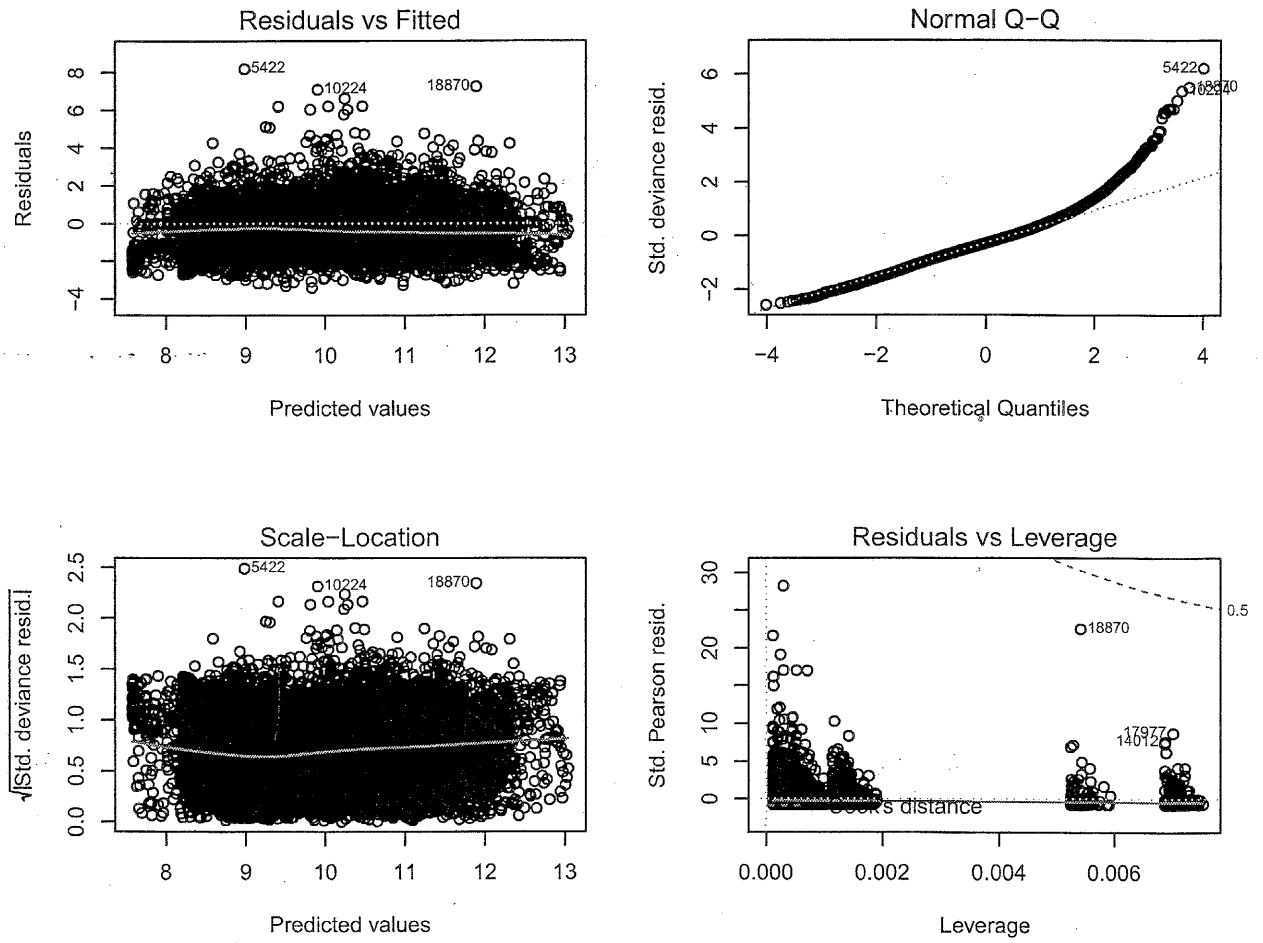

Figure 4.2.1: Diagnostic plots for the gamma GLM fitted to the training set of the `persinj` data.

TASK 5: Interpret the model

Run the selected model from Task 3 on the full dataset and provide the output. Interpret the results in a manner that will provide useful information to an insurance company or a personal injury insurance policyholder.

In CHUNK 10, we rerun the gamma GLM on the full dataset to provide a more robust prediction for future data and print the model summary.

```
# CHUNK 10
glm.final <- glm(amt ~ inj + legrep * op_time,
                 family = Gamma(link = "log"), data = persinj)
summary(glm.final)

##
## Call:
## glm(formula = amt ~ inj + legrep * op_time, family = Gamma(link = "log"),
##     data = persinj)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5289  -0.9142  -0.3648   0.1873   8.2209
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.2033907  0.0279443 293.562  < 2e-16 ***
## inj2            0.6281073  0.0248995  25.226  < 2e-16 ***
## inj3            0.8882054  0.0402472  22.069  < 2e-16 ***
## inj4            1.1199670  0.0951566  11.770  < 2e-16 ***
## inj5            1.3963478  0.0955128  14.619  < 2e-16 ***
## inj6            0.8867918  0.0816012  10.867  < 2e-16 ***
## inj9           -0.6205268  0.0381881 -16.249  < 2e-16 ***
## legrep          0.4437842  0.0354423  12.521  < 2e-16 ***
## op_time         0.0343052  0.0005303  64.685  < 2e-16 ***
## legrep:op_time -0.0050443  0.0006663  -7.571 3.86e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.675808)
##
##     Null deviance: 44010  on 22035  degrees of freedom
## Residual deviance: 22242  on 22026  degrees of freedom
## AIC: 487535
##
## Number of Fisher Scoring iterations: 8
```

We find that injury code, legal representation, and operational time are all statistically significant factors affecting the expected claim size of a personal injury. Furthermore, there is significant

interaction between legal representation and operational time, meaning that the effect of operational time on the expected claim size varies for injuries with and without legal representation. Since the log link is used, the expected claim size is multiplied by $\exp(\beta_j)$ for every unit increase in a quantitative predictor with coefficient β_j or when a qualitative predictor moves from its baseline level to a new level represented by a dummy variable with coefficient β_j, holding everything else constant. In this current context, we can say:

- With the exception of injuries with injury code 9 (which represents unclassified injuries), the expected claim size for injuries with codes 2, 3, 4, 5, 6 is higher than that for injuries with code 1 (baseline). For instance, the expected claim size for injuries with code 5 is estimated to be $\exp(1.3963478) = 4.0404$ times of that of injuries with code 1. That the estimated coefficients for codes 2, 3, 4, 5, 6 are all positive and mostly increasing across the codes makes intuitive sense as we would expect the more serious the injury, the larger the expected claim size. However, we would expect the estimated coefficient for code 6 to be the largest as code 6 corresponds to fatalities. This is not the case for the gamma GLM (perhaps nobody is working hard enough on behalf of the deceased to settle the claim!? ☻).

- Operational time is positively associated with the expected claim size, though the effect is not as positive for injuries with legal representation (the estimated coefficient for `op_time` is 0.0343052 vs. 0.0292609 for injuries without and with legal representation, respectively). The positive effect is reasonable because larger claims are often more difficult to quantify and require more time to settle, leading to longer delays.

- For most injuries, those with legal representation have a higher expected claim size than those without legal representation, unless the injuries take an extraordinarily long time to settle (operational time higher than 88,[vii] to be precise). Again, this is intuitive as clients with the assistance of the legal advice are in a better position to fight for a larger settled claim size.

The findings above generally align with our intuition and shed light on the directional impact of different factors on the settled claim amount. Using these findings, policyholders may, for example, find it justifiable to employ legal representation in the hope of having a larger settled claim.

EXAM NOTE

When you are asked to interpret the results of a GLM, the following three-part structure can be useful:

- First interpret the precise values of the estimated coefficients, e.g., every unit increase in a continuous predictor is associated with a multiplicative change of e^{β_j} in the expected value of the target variable, holding everything else fixed.
- Comment on whether the sign of the estimated coefficients conforms to intuition. Do the results make sense? You can use your common knowledge in this part.
- Relate the findings to the "big picture" and explain how they can help clients make a better decision in life.

[vii] This is determined by solving the inequality $0.0343052(\texttt{op\_time}) < 0.4437842 + 0.0292609(\texttt{op\_time})$.