Preliminary Analysis of RTC Data

Matthew Knowles

2023-03-11

1: Data Cleaning

We begin by converting to a binary variable by severity. Accidents with a severity of 1 stay as such, and others are converted to a 0.

```
accidents <- accidents %>%
    mutate(bin_severity = ifelse(.data$accident_severity == 1, 1, 0))
```

We now reduce the size of the dataset to contain only variables we care about.

```
sub_accidents <- accidents %>%
    select(
        "accident_index",
        "bin_severity",
        "day_of_week",
        "road_type",
        "speed_limit",
        "light_conditions",
        "weather_conditions",
        "road_surface_conditions"
    )

sub_vehicles <- vehicles %>%
    select(
        "accident_index",
        "sex_of_driver",
        "age_band_of_driver"
    )

sub_casualties <- casualty %>%
    select(
        "accident_index",
        "sex_of_casualty",
        "age_band_of_casualty"
    )

df_sub <- merge.data.frame(sub_vehicles, sub_casualties, by = "accident_index")
df_sub_unique <- unique(df_sub)
df <- merge.data.frame(sub_accidents, df_sub_unique, by = "accident_index")
```

We want all variables, except speed limit, to be factors.

```
df <- df %>%
    mutate(across(where(is.integer), as.factor))
```

```
df$speed_limit <- as.integer(df$speed_limit) #Speed limit goes back to integer
df$bin_severity <- as.factor(df$bin_severity)
head(tibble(df))
```

```
## # A tibble: 6 x 12
##   accident_index bin_s~1 day_o~2 road_~3 speed~4 light~5 weath~6 road_~7 sex_o~8
##   <chr>          <fct>   <fct>   <fct>     <int> <fct>   <fct>   <fct>   <fct>
## 1 2020010219808  0       3       6             2 1       9       9       2
## 2 2020010220496  0       2       6             2 1       1       1       1
## 3 2020010228005  0       4       6             3 4       1       2       3
## 4 2020010228006  0       4       6             3 4       1       1       1
## 5 2020010228011  0       4       6             3 4       1       1       1
## 6 2020010228012  0       4       2             2 4       1       1       1
## # ... with 3 more variables: age_band_of_driver <fct>, sex_of_casualty <fct>,
## #   age_band_of_casualty <fct>, and abbreviated variable names 1: bin_severity,
## #   2: day_of_week, 3: road_type, 4: speed_limit, 5: light_conditions,
## #   6: weather_conditions, 7: road_surface_conditions, 8: sex_of_driver
```

We need to remove any rows in which there is a -1 in a column of the data.

```
has.neg <- apply(df, 1, function(row) any(row == -1))
df <- df[-which(has.neg), ] %>%
    select(-accident_index)
```

We can now also create a train and test set for later on.

```
df$id <- 1:nrow(df)
df_train <- df %>% sample_frac(0.8)
df_test <- anti_join(df, df_train, by = "id")
```

2: Fitting

We fit the glm to the training data. We specify that the response variable is binomial, and that we wish to use a logit link function.

```
fit <- glm(
    data = df_train,
    formula = bin_severity ~ .,
    family = binomial(link = "logit")
)
```

Let us take a look at the summary and plots of this model.

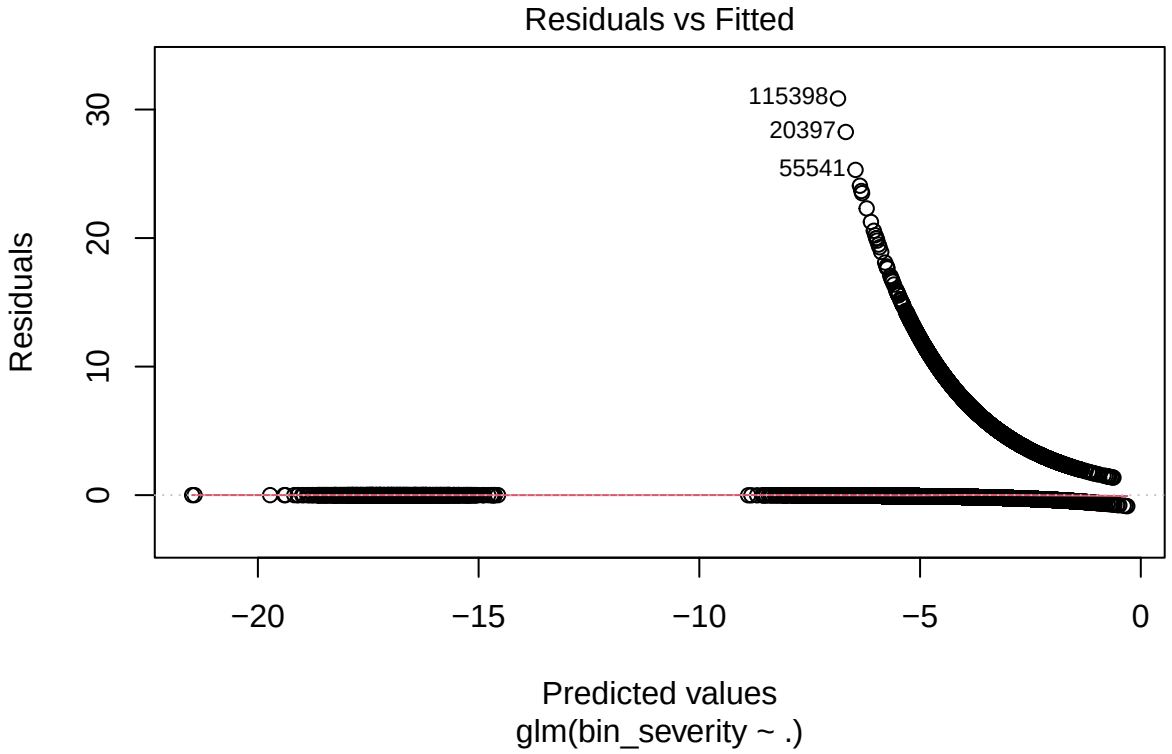
```
summary(fit)
```

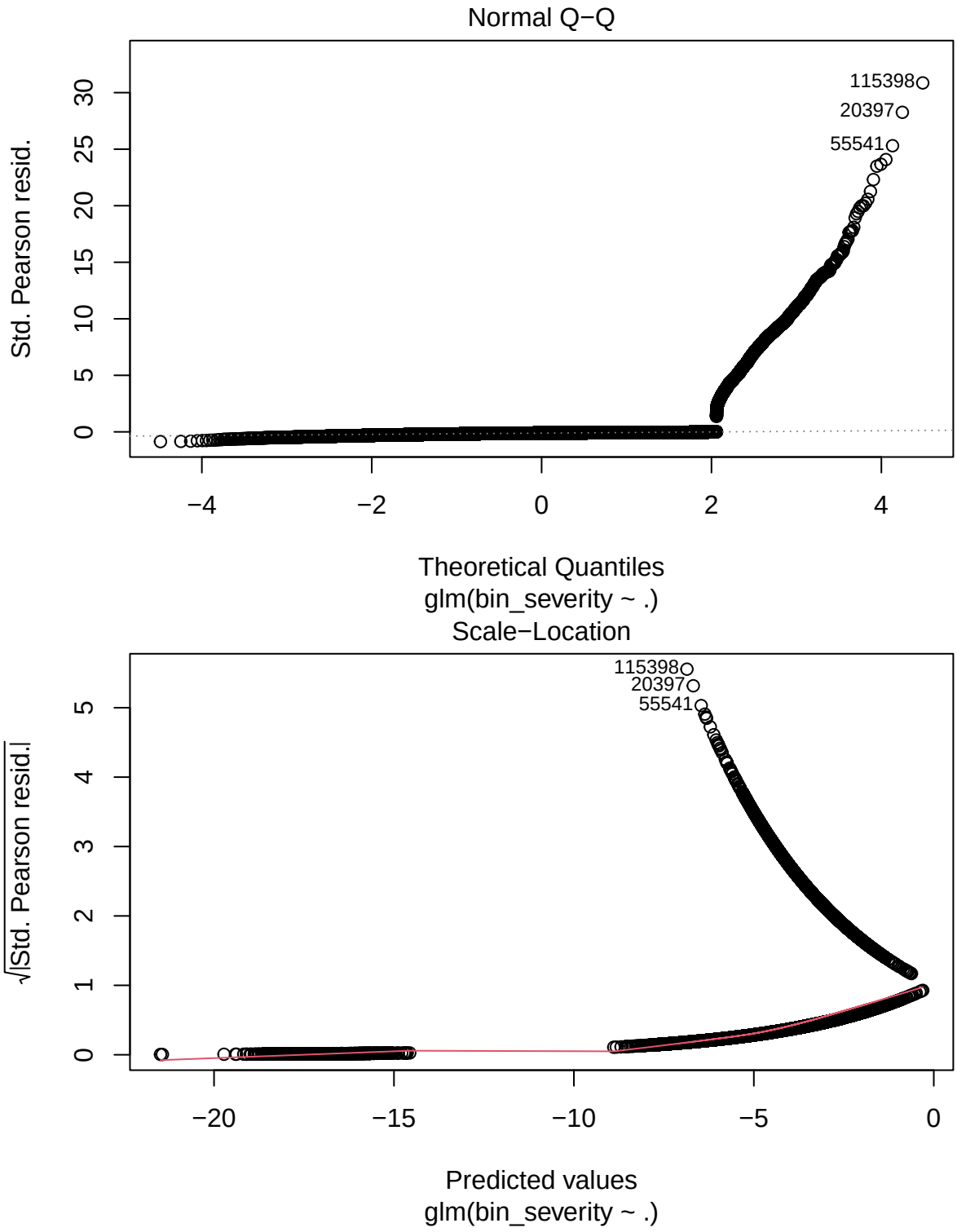
```
##
## Call:
## glm(formula = bin_severity ~ ., family = binomial(link = "logit"),
##     data = df_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0505  -0.2115  -0.1487  -0.1108   3.7040
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
```

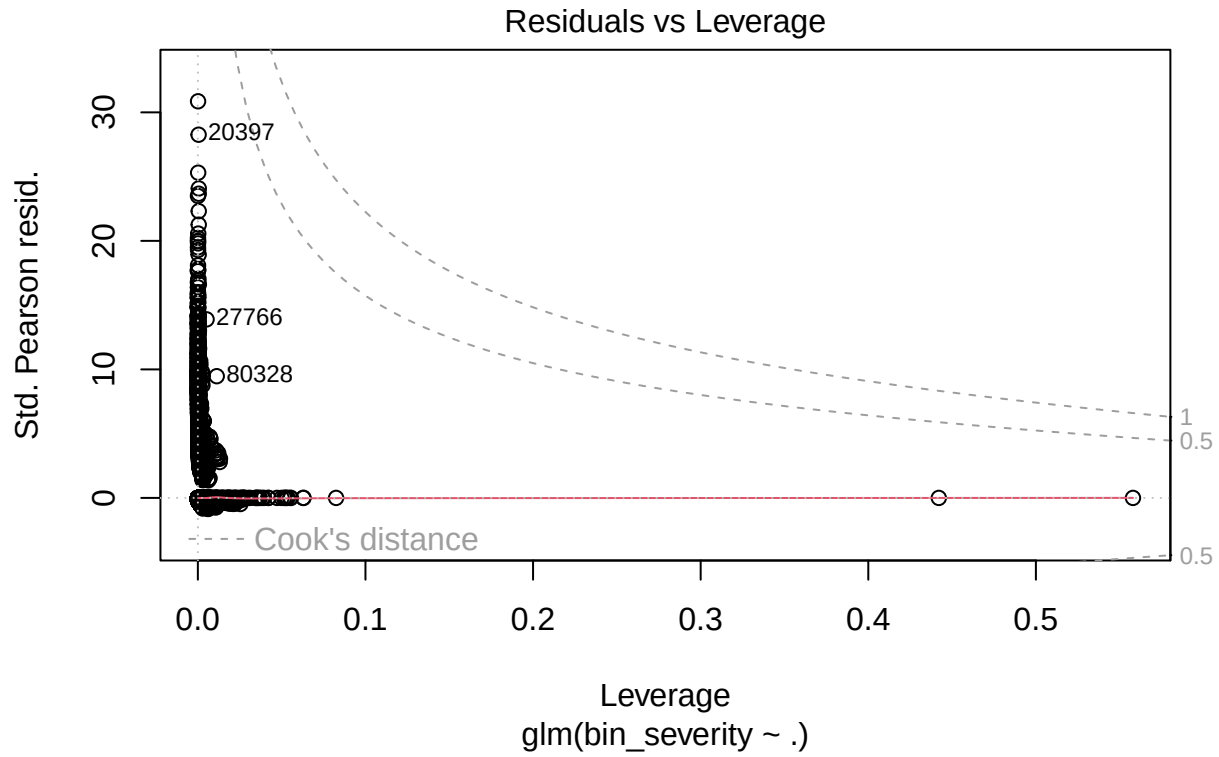
```
## (Intercept)              -1.894e+01  4.398e+02  -0.043 0.965653
## day_of_week2             -4.198e-01  7.825e-02  -5.365 8.09e-08 ***
## day_of_week3             -3.586e-01  7.628e-02  -4.701 2.58e-06 ***
## day_of_week4             -2.353e-01  7.348e-02  -3.202 0.001367 **
## day_of_week5             -1.111e-01  7.091e-02  -1.567 0.117036
## day_of_week6             -3.475e-01  7.355e-02  -4.725 2.30e-06 ***
## day_of_week7              4.479e-02  7.056e-02   0.635 0.525585
## road_type2               -3.604e-01  5.264e-01  -0.685 0.493535
## road_type3                7.415e-01  1.669e-01   4.444 8.83e-06 ***
## road_type6                1.357e+00  1.610e-01   8.430  < 2e-16 ***
## road_type7                2.150e-01  2.478e-01   0.868 0.385640
## road_type9                1.764e-01  4.777e-01   0.369 0.711888
## speed_limit               4.515e-01  1.554e-02  29.047  < 2e-16 ***
## light_conditions4         4.186e-01  5.817e-02   7.197 6.17e-13 ***
## light_conditions5         6.362e-01  1.909e-01   3.333 0.000859 ***
## light_conditions6         7.938e-01  5.686e-02  13.962  < 2e-16 ***
## light_conditions7         3.551e-02  1.978e-01   0.180 0.857506
## weather_conditions2      -4.078e-01  7.462e-02  -5.466 4.61e-08 ***
## weather_conditions3       7.027e-01  3.733e-01   1.882 0.059804 .
## weather_conditions4       3.251e-01  1.236e-01   2.630 0.008549 **
## weather_conditions5      -2.336e-01  1.379e-01  -1.694 0.090234 .
## weather_conditions6      -1.227e+01  2.154e+02  -0.057 0.954588
## weather_conditions7       1.213e-01  1.814e-01   0.669 0.503635
## weather_conditions8       3.028e-01  1.167e-01   2.594 0.009477 **
## weather_conditions9      -5.773e-01  2.399e-01  -2.406 0.016118 *
## road_surface_conditions2  1.120e-01  5.079e-02   2.205 0.027420 *
## road_surface_conditions3 -2.439e+00  1.062e+00  -2.297 0.021627 *
## road_surface_conditions4 -6.702e-01  2.317e-01  -2.893 0.003815 **
## road_surface_conditions5 -8.012e-03  3.506e-01  -0.023 0.981768
## road_surface_conditions9 -1.091e+01  9.344e+01  -0.117 0.907026
## sex_of_driver2           -3.039e-01  5.140e-02  -5.912 3.39e-09 ***
## sex_of_driver3           -1.005e+00  3.052e-01  -3.293 0.000992 ***
## age_band_of_driver2       1.144e+01  4.398e+02   0.026 0.979246
## age_band_of_driver3       1.116e+01  4.398e+02   0.025 0.979761
## age_band_of_driver4       1.168e+01  4.398e+02   0.027 0.978805
## age_band_of_driver5       1.193e+01  4.398e+02   0.027 0.978359
## age_band_of_driver6       1.198e+01  4.398e+02   0.027 0.978262
## age_band_of_driver7       1.189e+01  4.398e+02   0.027 0.978430
## age_band_of_driver8       1.198e+01  4.398e+02   0.027 0.978265
## age_band_of_driver9       1.210e+01  4.398e+02   0.028 0.978049
## age_band_of_driver10      1.185e+01  4.398e+02   0.027 0.978511
## age_band_of_driver11      1.207e+01  4.398e+02   0.027 0.978103
## sex_of_casualty2         -4.027e-01  4.655e-02  -8.651  < 2e-16 ***
## sex_of_casualty9         -1.148e+01  1.695e+03  -0.007 0.994598
## age_band_of_casualty2    -5.448e-01  2.623e-01  -2.077 0.037817 *
## age_band_of_casualty3    -3.121e-01  2.295e-01  -1.360 0.173923
## age_band_of_casualty4    -2.756e-01  1.885e-01  -1.462 0.143790
## age_band_of_casualty5    -1.419e-01  1.811e-01  -0.784 0.433226
## age_band_of_casualty6    -2.017e-01  1.752e-01  -1.151 0.249558
## age_band_of_casualty7    -6.964e-03  1.767e-01  -0.039 0.968556
## age_band_of_casualty8    -1.249e-03  1.771e-01  -0.007 0.994372
## age_band_of_casualty9     1.448e-01  1.788e-01   0.810 0.418204
## age_band_of_casualty10    5.337e-01  1.852e-01   2.881 0.003959 **
## age_band_of_casualty11    1.175e+00  1.836e-01   6.399 1.56e-10 ***
```

```
## id                        1.729e-06  4.525e-07   3.822 0.000132 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26947  on 138720  degrees of freedom
## Residual deviance: 24211  on 138666  degrees of freedom
## AIC: 24321
##
## Number of Fisher Scoring iterations: 15
```

```
plot(fit)
```

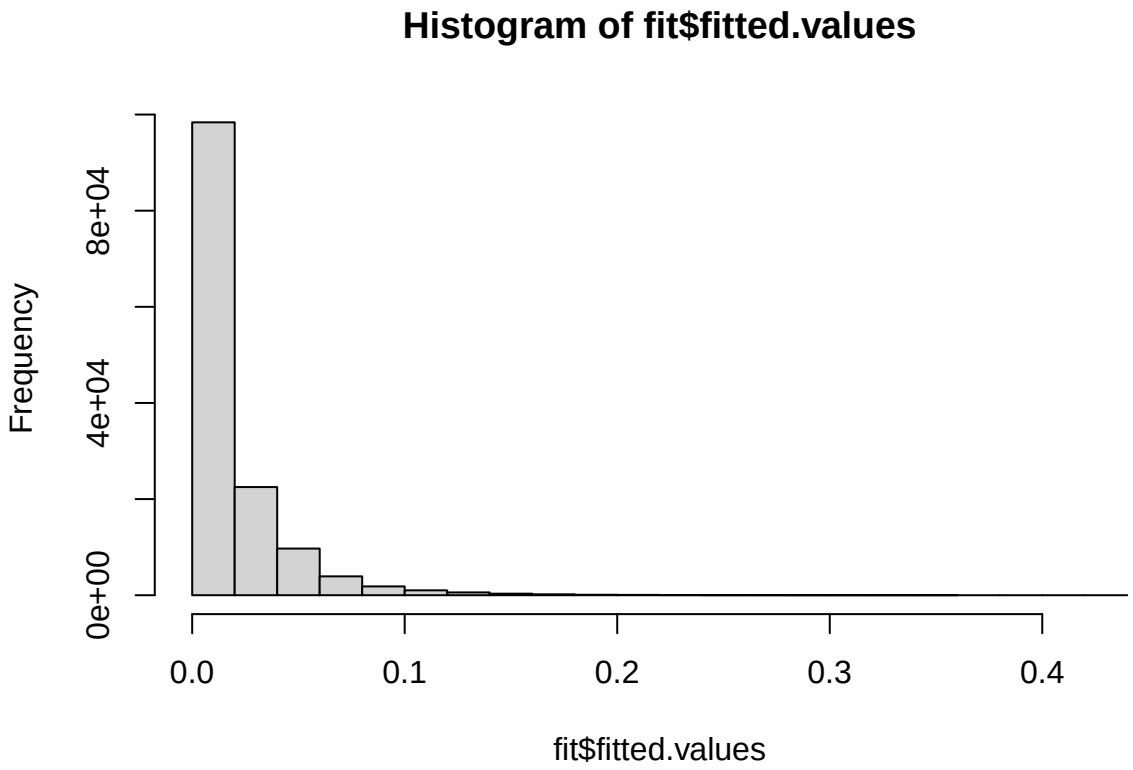



Normal Q−Q
Std. Pearson resid.
115398
20397
55541
0
5
10
15
20
25
30
−4
−2
0
2
4
Theoretical Quantiles
glm(bin_severity ~ .)
Scale−Location
√|Std. Pearson resid.|
115398
20397
55541
0
1
2
3
4
5
−20
−15
−10
−5
0
Predicted values
glm(bin_severity ~ .)



This histogram shows the distribution of fitted values from the training data.

```
hist(fit$fitted.values)
```



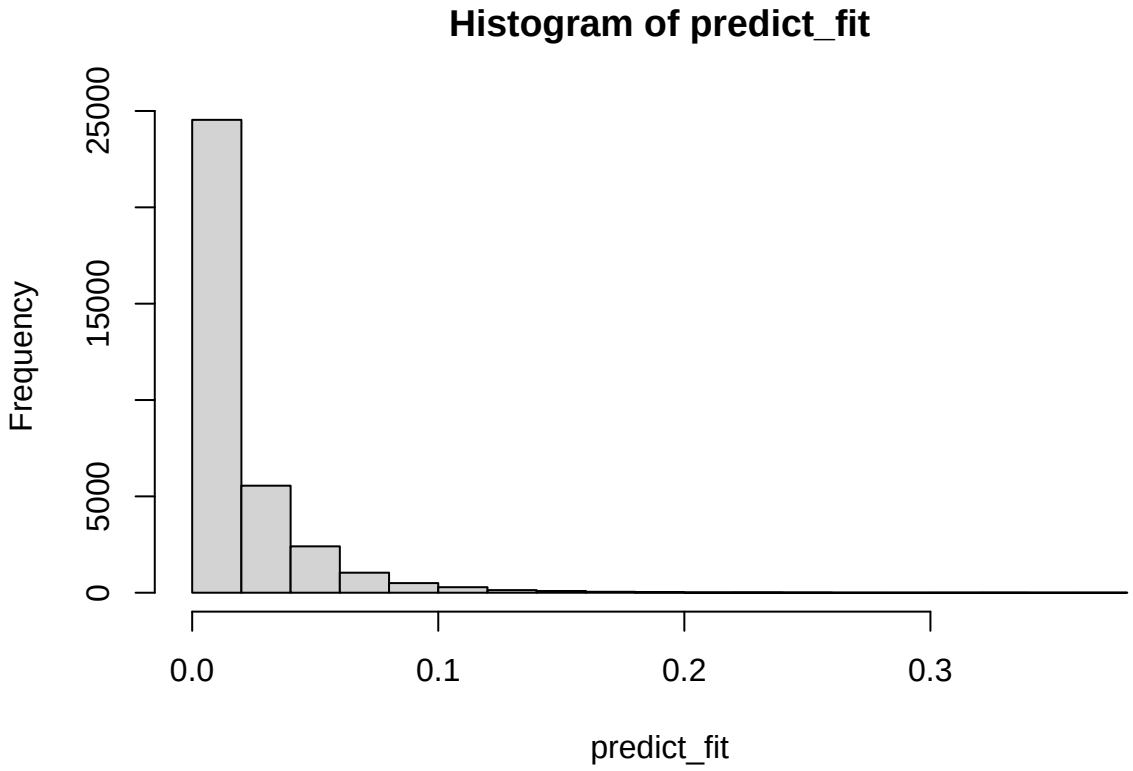
3: Predicting Unseen Values

We can use the predict function to see how well the model predicts the response on unseen data.

```
predict_fit <- predict(
    fit,
    newdata = df_test,
    type = "response"
)
```

This histogram shows the distribution of fitted values from the test data.

```
hist(predict_fit)
```



Notice the shape of this histogram and the previous one are identical.