Base model fitting for RTC Data

Matthew Knowles

2023-03-15

1: Data Cleaning

We begin by converting to a binary variable by severity. Accidents with a severity of 3 become a 0, and others are converted to a 1.

```
accidents <- accidents %>%
    mutate(bin_severity = ifelse(.data$accident_severity == 3, 0, 1))
```

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",
        "time",
        "light_conditions",
        "weather_conditions",
        "local_authority_ons_district",
        "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")
```

Before dealing with other variables in the dataset, we deal with time on its own. The times provided in the dataset are characters, for example of the form "12:14". There are many unique times in the dataset, which

would cause issues when fitting models as these would all be treated as individual factors. We therefore split the 24 hour day into 6 chunks of 4 hours, as this dramatically reduces the number of factors, but also gives good information about the relationship between time of day and accident severity.

The first thing to do is build a function to parse the times by splitting the character before the colon, and converting that to an integer. A series of if-statements are then checked to put the time into the correct category. This isn't the most computationally efficient method, but it does the job. We then apply this function to every time observation to create a new column of data called "time_group", and drop the original time variable from the data so it doesn't cause problems later on.

```
# Split time into 6 chunks of four hours
parse_times <- function(time){
    split_time <- stringr::str_split(time, ":", simplify = TRUE)
    hour <- as.integer(split_time[1])
    if (hour < 4) return(1)
    if (hour < 8 ) return(2)
    if (hour < 12) return(3)
    if (hour < 16) return(4)
    ifelse(hour < 20, return(5), return(6))
}

df <- df %>%
    mutate(time_group = purrr::map_int(time, parse_times)) %>%
    select(-time)
```

Location is to be included, but in the current state, the *local_authority_ons_district* variable has too many unique values. To account for this, the region codes (those beginning with E0, N, S or W) are matched to a wider local area, such as the West Midlands, Yorkshire and The Humber etc. Across England, Wales, Scotland & Northern Ireland, there are 12 unique regions. Each region is assigned a numerical valuable. This assignment is done alphabetically, so East Midlands is 1, all the way through to Yorkshire and The Humber at 12. With this in mind, we can add region code to the data.

```
# Obtains the numerical value of a region from a given ONS coded
obtain_region <- function(region_code){
    return(regions$Region.Code[which(regions$Code == region_code)])
}

df <- df %>%
    mutate(region = purrr::map_int(local_authority_ons_district, obtain_region)) %>%
    select(-local_authority_ons_district)
```

We want all variables, except speed limit, to be factors. To achieve this we mutate across all integer columns and change the type to factor. However, we need speed limit to stay as an integer to make the model adaptable to non-standard speed limits. The "bin_severity" column needs to be a factor, so we ensure this by calling the as.factor function once more just to ensure it is infact a factor. By viewing the head of the data as a tibble we can check that the columns are of expected type.

```
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 14
##   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  1       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 5 more variables: age_band_of_driver <fct>, sex_of_casualty <fct>,
## #   age_band_of_casualty <fct>, time_group <fct>, region <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
```

Some observations of the data are -1. We don't want these in the data, so we trim the data down to remove any rows that contain -1 in any of the columns. This helps with run-time of the model as well due to the size of the data before this is done. At this stage we also remove the accident index variable, as it was only needed for combining the three datasets, and isn't needed in fitting the GLM itself.

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

Finally before fitting, split the data into two sets. A training and test set. The training set has been selected to contain a random sample of 75% of the original data, and the other 25% becomes the test set for later on. There is a of maths one could do to identify an optimal training data set size, but in this case we have selected an 75:25 split as a general rule of thumb.

```
df$id <- 1:nrow(df)
df_train <- df %>% sample_frac(0.75)
df_test <- anti_join(df, df_train, by = "id")
df_train <- df_train %>% select(-id)
df_test <- df_test %>% select(-id)
```

2: Fitting

We fit a general glm to the training data. We specify that the response variable is binomial, and that we wish to use a logit link function. No other interaction terms are included, as this will be built upon later.

```
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.

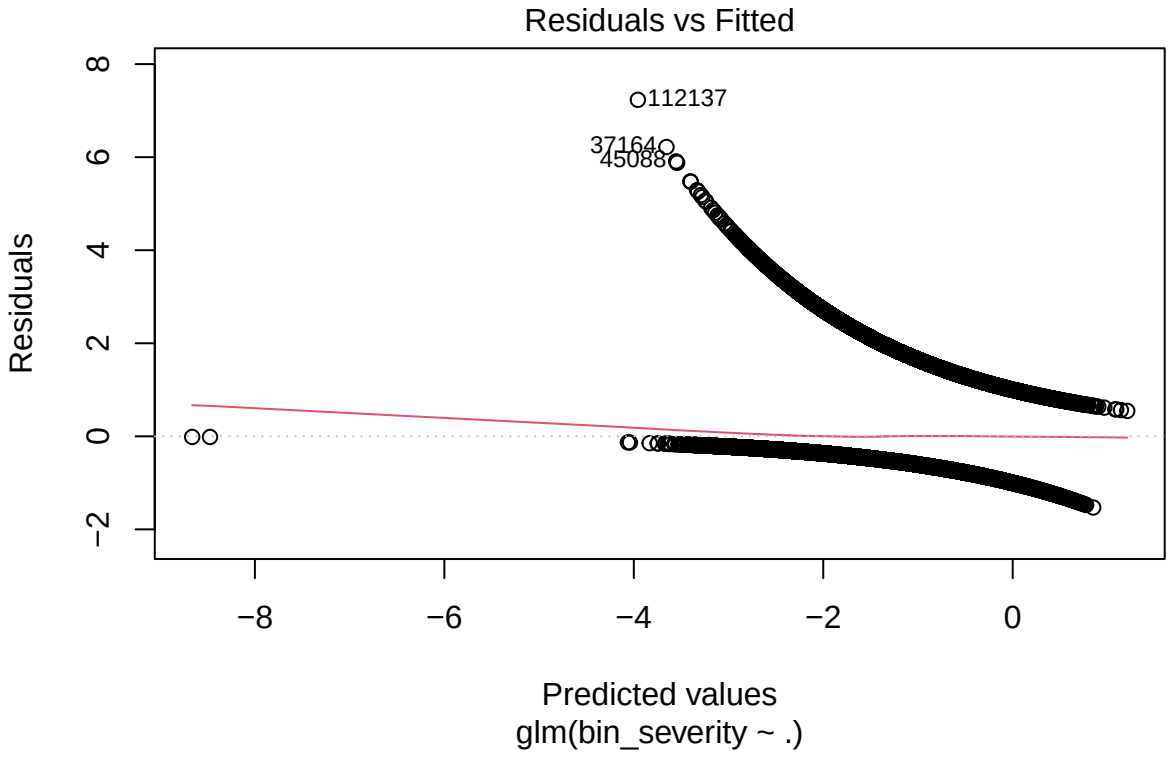
```
summary(fit)
```

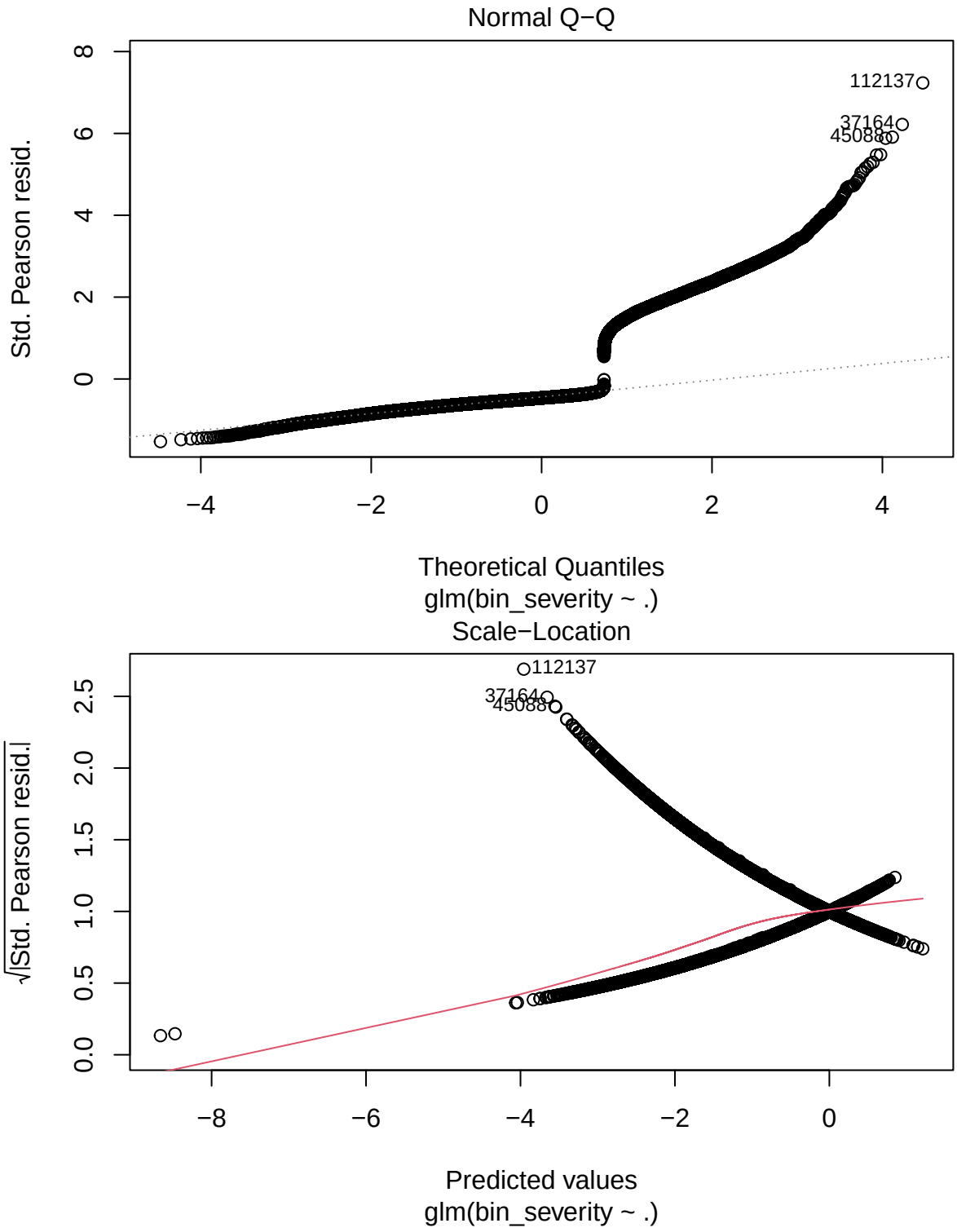
```
##
## Call:
## glm(formula = bin_severity ~ ., family = binomial(link = "logit"),
##     data = df_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5530  -0.7554  -0.6234  -0.4143   2.8199
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
```

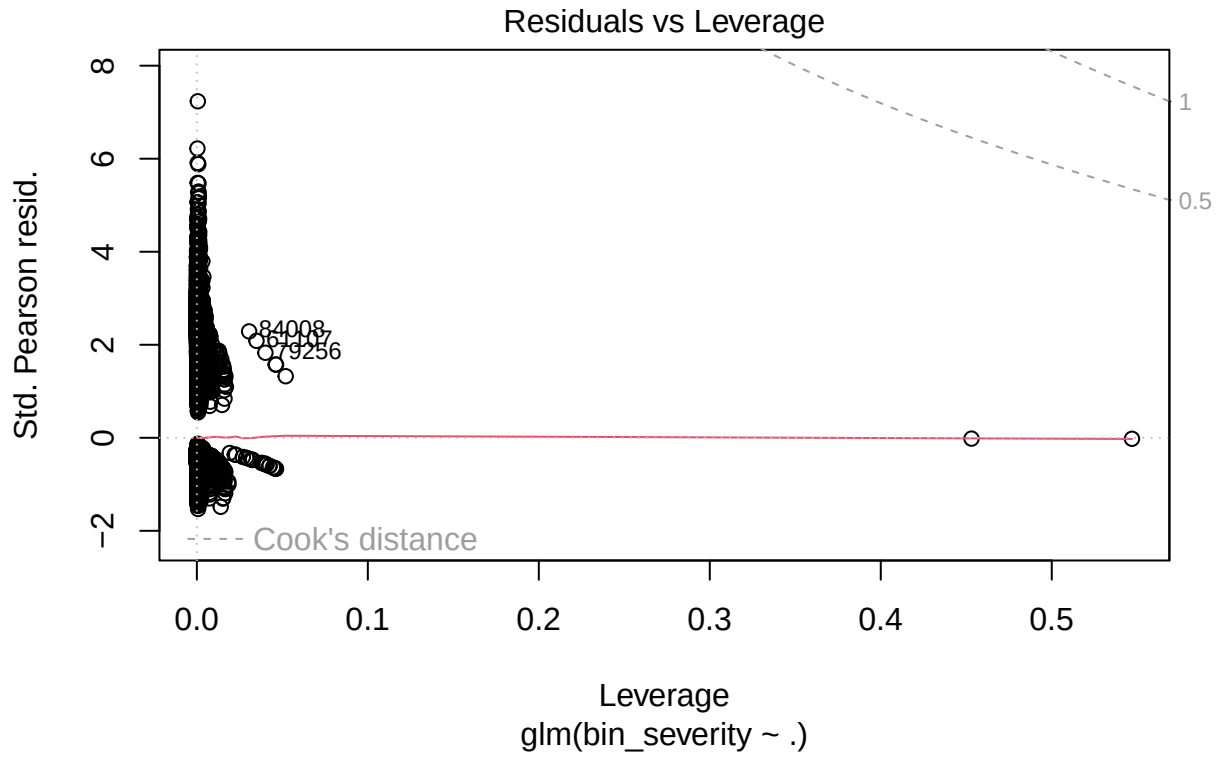
```
## (Intercept)              -1.464127   0.473471  -3.092 0.001986 **
## day_of_week2             -0.197590   0.026688  -7.404 1.32e-13 ***
## day_of_week3             -0.208837   0.026548  -7.866 3.65e-15 ***
## day_of_week4             -0.196915   0.026283  -7.492 6.78e-14 ***
## day_of_week5             -0.125942   0.025920  -4.859 1.18e-06 ***
## day_of_week6             -0.165489   0.025562  -6.474 9.54e-11 ***
## day_of_week7             -0.041305   0.026135  -1.580 0.114000
## road_type2                0.287052   0.071085   4.038 5.39e-05 ***
## road_type3                0.090401   0.037048   2.440 0.014684 *
## road_type6                0.497909   0.033171  15.010  < 2e-16 ***
## road_type7                0.029278   0.064746   0.452 0.651124
## road_type9               -0.307577   0.098255  -3.130 0.001746 **
## speed_limit               0.188313   0.005430  34.679  < 2e-16 ***
## light_conditions4         0.041178   0.023057   1.786 0.074114 .
## light_conditions5         0.060947   0.078382   0.778 0.436828
## light_conditions6         0.226524   0.031151   7.272 3.55e-13 ***
## light_conditions7        -0.447979   0.064432  -6.953 3.58e-12 ***
## weather_conditions2      -0.104974   0.025669  -4.090 4.32e-05 ***
## weather_conditions3       0.155919   0.170115   0.917 0.359380
## weather_conditions4       0.191685   0.050619   3.787 0.000153 ***
## weather_conditions5      -0.047820   0.051289  -0.932 0.351142
## weather_conditions6       0.489616   0.250521   1.954 0.050655 .
## weather_conditions7      -0.012444   0.084195  -0.148 0.882497
## weather_conditions8      -0.047748   0.046018  -1.038 0.299460
## weather_conditions9      -0.366618   0.063665  -5.759 8.48e-09 ***
## road_surface_conditions2 -0.028998   0.019280  -1.504 0.132567
## road_surface_conditions3 -1.069779   0.226799  -4.717 2.40e-06 ***
## road_surface_conditions4 -0.319076   0.078275  -4.076 4.58e-05 ***
## road_surface_conditions5 -0.238295   0.152297  -1.565 0.117659
## road_surface_conditions9 -0.101878   0.157832  -0.645 0.518611
## sex_of_driver2           -0.169964   0.016644 -10.211  < 2e-16 ***
## sex_of_driver3           -0.527675   0.070322  -7.504 6.20e-14 ***
## age_band_of_driver2      -0.447634   0.511529  -0.875 0.381524
## age_band_of_driver3       0.009809   0.478789   0.020 0.983655
## age_band_of_driver4       0.000186   0.473351   0.000 0.999686
## age_band_of_driver5       0.012298   0.473035   0.026 0.979259
## age_band_of_driver6      -0.009524   0.472728  -0.020 0.983927
## age_band_of_driver7      -0.001460   0.472812  -0.003 0.997537
## age_band_of_driver8       0.052097   0.472878   0.110 0.912274
## age_band_of_driver9       0.132831   0.473019   0.281 0.778852
## age_band_of_driver10      0.114663   0.473553   0.242 0.808678
## age_band_of_driver11      0.031305   0.474275   0.066 0.947372
## sex_of_casualty2         -0.348639   0.015559 -22.407  < 2e-16 ***
## sex_of_casualty9         -7.394321  31.046834  -0.238 0.811752
## age_band_of_casualty2    -0.113647   0.074089  -1.534 0.125048
## age_band_of_casualty3     0.074811   0.067880   1.102 0.270415
## age_band_of_casualty4    -0.051071   0.060757  -0.841 0.400585
## age_band_of_casualty5    -0.198737   0.059698  -3.329 0.000871 ***
## age_band_of_casualty6    -0.206836   0.057575  -3.592 0.000328 ***
## age_band_of_casualty7    -0.165834   0.058377  -2.841 0.004501 **
## age_band_of_casualty8    -0.081462   0.058560  -1.391 0.164200
## age_band_of_casualty9     0.081698   0.059625   1.370 0.170623
## age_band_of_casualty10    0.320134   0.063133   5.071 3.96e-07 ***
## age_band_of_casualty11    0.455002   0.065967   6.897 5.29e-12 ***
```

```
## time_group2              -0.359823   0.046149  -7.797 6.34e-15 ***
## time_group3              -0.562477   0.044602 -12.611  < 2e-16 ***
## time_group4              -0.491373   0.043513 -11.292  < 2e-16 ***
## time_group5              -0.456832   0.040378 -11.314  < 2e-16 ***
## time_group6              -0.282406   0.041447  -6.814 9.51e-12 ***
## region2                  -0.049977   0.031182  -1.603 0.108991
## region3                  -0.305245   0.031348  -9.737  < 2e-16 ***
## region4                  -0.113491   0.044028  -2.578 0.009945 **
## region5                  -0.013708   0.031938  -0.429 0.667764
## region7                   0.581724   0.036562  15.910  < 2e-16 ***
## region8                   0.036149   0.028532   1.267 0.205176
## region9                  -0.256734   0.032974  -7.786 6.92e-15 ***
## region10                 -0.086360   0.043204  -1.999 0.045619 *
## region11                 -0.176308   0.034336  -5.135 2.83e-07 ***
## region12                  0.023450   0.032769   0.716 0.474221
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 140890  on 130050  degrees of freedom
## Residual deviance: 134516  on 129982  degrees of freedom
## AIC: 134654
##
## Number of Fisher Scoring iterations: 7
```

```
plot(fit)
```

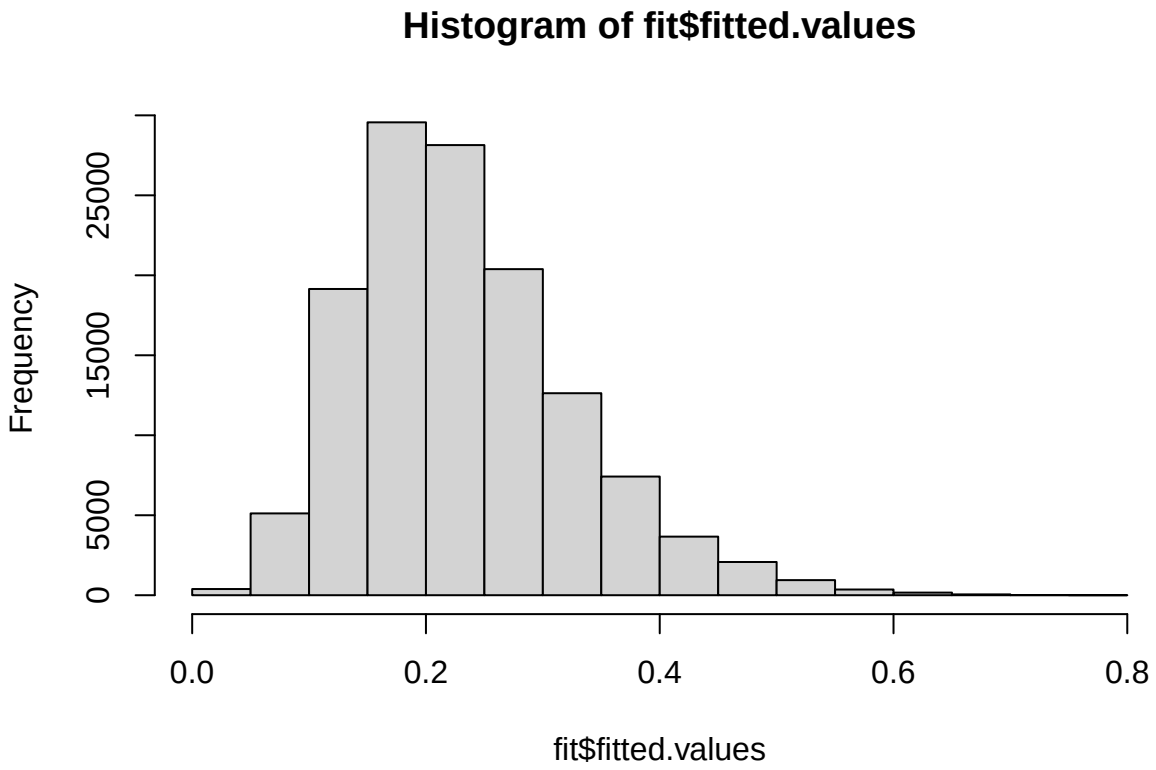



Normal Q−Q
Std. Pearson resid.
112137
37164
45088
Theoretical Quantiles
glm(bin_severity ~ .)
Scale−Location
√|Std. Pearson resid.|
112137
37164
45088
Predicted values
glm(bin_severity ~ .)



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

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



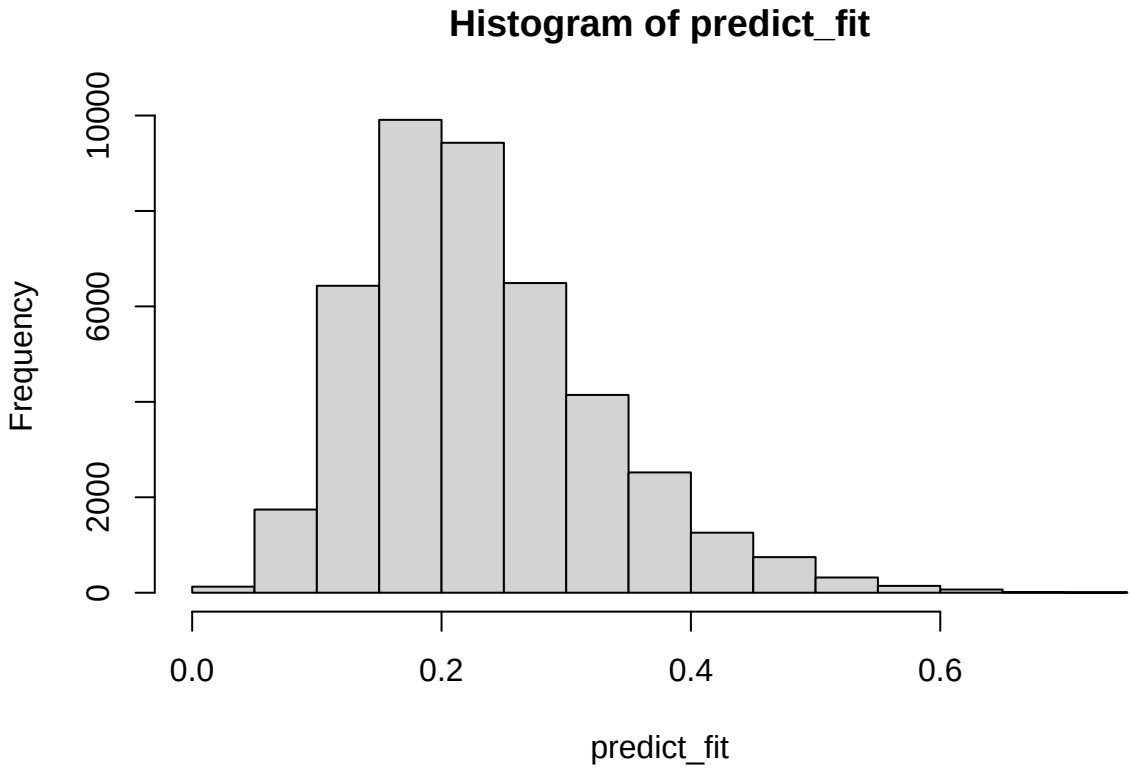
3: Predicting Unseen Values

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.

4: Speed Limit and Road Type analysis

We now present a slightly more specific model than the one we have just seen. We include an interaction term between speed limit and road type, allowing us to assess whether or not speed limits on a given road type are appropriate based on the contribution they make towards accident severity.

```
fit_rt_sl <- glm(
    data = df_train %>% select(-region),
    formula = bin_severity ~ . + speed_limit*road_type,
    family = binomial(link = "logit")
)
```

We can view the summary of this model as before.

```
summary(fit_rt_sl)
```

```
##
## Call:
## glm(formula = bin_severity ~ . + speed_limit * road_type, family = binomial(link = "logit"),
##     data = df_train %>% select(-region))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5556  -0.7445  -0.6324  -0.4427   2.8551
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.2955113  0.4784707  -2.708 0.006777 **
## day_of_week2             -0.1868274  0.0266261  -7.017 2.27e-12 ***
## day_of_week3             -0.2011824  0.0264936  -7.594 3.11e-14 ***
## day_of_week4             -0.1886120  0.0262256  -7.192 6.39e-13 ***
## day_of_week5             -0.1218249  0.0258619  -4.711 2.47e-06 ***
## day_of_week6             -0.1535451  0.0255027  -6.021 1.74e-09 ***
## day_of_week7             -0.0352020  0.0260743  -1.350 0.176996
## road_type2               -0.3684813  0.2716767  -1.356 0.174997
## road_type3                0.2674171  0.1108476   2.412 0.015845 *
## road_type6                0.0389711  0.1014171   0.384 0.700782
## road_type7                0.1716657  0.1853417   0.926 0.354336
## road_type9               -0.8154183  0.2656643  -3.069 0.002145 **
## speed_limit               0.1337331  0.0246885   5.417 6.07e-08 ***
## light_conditions4         0.0529199  0.0230116   2.300 0.021465 *
## light_conditions5         0.0791213  0.0783630   1.010 0.312650
## light_conditions6         0.2147569  0.0311500   6.894 5.41e-12 ***
## light_conditions7        -0.4607163  0.0642667  -7.169 7.56e-13 ***
## weather_conditions2      -0.1118866  0.0255824  -4.374 1.22e-05 ***
## weather_conditions3       0.2806343  0.1691067   1.660 0.097013 .
## weather_conditions4       0.1948668  0.0504594   3.862 0.000113 ***
## weather_conditions5      -0.0151013  0.0510060  -0.296 0.767177
## weather_conditions6       0.5453930  0.2486649   2.193 0.028287 *
## weather_conditions7      -0.0514781  0.0842852  -0.611 0.541358
## weather_conditions8      -0.0905964  0.0457269  -1.981 0.047563 *
## weather_conditions9      -0.3388779  0.0633148  -5.352 8.69e-08 ***
## road_surface_conditions2 -0.0107237  0.0191906  -0.559 0.576297
## road_surface_conditions3 -0.9820549  0.2263410  -4.339 1.43e-05 ***
## road_surface_conditions4 -0.3093870  0.0778679  -3.973 7.09e-05 ***
## road_surface_conditions5 -0.1705393  0.1519669  -1.122 0.261772
## road_surface_conditions9 -0.2710922  0.1571967  -1.725 0.084610 .
## sex_of_driver2           -0.1636083  0.0166154  -9.847  < 2e-16 ***
## sex_of_driver3           -0.5508560  0.0699396  -7.876 3.38e-15 ***
## age_band_of_driver2      -0.4316815  0.5083705  -0.849 0.395799
## age_band_of_driver3       0.0015594  0.4755592   0.003 0.997384
## age_band_of_driver4      -0.0288869  0.4701107  -0.061 0.951003
## age_band_of_driver5      -0.0222741  0.4697892  -0.047 0.962184
## age_band_of_driver6      -0.0425563  0.4694812  -0.091 0.927774
## age_band_of_driver7      -0.0338479  0.4695625  -0.072 0.942535
## age_band_of_driver8       0.0191896  0.4696323   0.041 0.967407
## age_band_of_driver9       0.1039909  0.4697758   0.221 0.824810
## age_band_of_driver10      0.0890253  0.4703141   0.189 0.849866
## age_band_of_driver11      0.0001553  0.4710454   0.000 0.999737
## sex_of_casualty2         -0.3434003  0.0155389 -22.099  < 2e-16 ***
```

```
## sex_of_casualty9         -7.2817289 31.0474580  -0.235 0.814569
## age_band_of_casualty2    -0.1176783  0.0739108  -1.592 0.111347
## age_band_of_casualty3     0.0697951  0.0676923   1.031 0.302511
## age_band_of_casualty4    -0.0831941  0.0605838  -1.373 0.169688
## age_band_of_casualty5    -0.2368996  0.0595137  -3.981 6.87e-05 ***
## age_band_of_casualty6    -0.2480771  0.0573749  -4.324 1.53e-05 ***
## age_band_of_casualty7    -0.2044044  0.0581806  -3.513 0.000443 ***
## age_band_of_casualty8    -0.1143122  0.0583787  -1.958 0.050217 .
## age_band_of_casualty9     0.0542645  0.0594533   0.913 0.361387
## age_band_of_casualty10    0.2919038  0.0629713   4.636 3.56e-06 ***
## age_band_of_casualty11    0.4286917  0.0658077   6.514 7.30e-11 ***
## time_group2              -0.3477043  0.0459603  -7.565 3.87e-14 ***
## time_group3              -0.5453429  0.0443941 -12.284  < 2e-16 ***
## time_group4              -0.4689939  0.0433080 -10.829  < 2e-16 ***
## time_group5              -0.4366708  0.0401716 -10.870  < 2e-16 ***
## time_group6              -0.2766503  0.0412474  -6.707 1.99e-11 ***
## road_type2:speed_limit    0.1750050  0.0941224   1.859 0.062980 .
## road_type3:speed_limit   -0.0171048  0.0264023  -0.648 0.517080
## road_type6:speed_limit    0.1249034  0.0254100   4.916 8.85e-07 ***
## road_type7:speed_limit   -0.0118573  0.0382760  -0.310 0.756725
## road_type9:speed_limit    0.1121584  0.0802255   1.398 0.162102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 140890  on 130050  degrees of freedom
## Residual deviance: 135143  on 129987  degrees of freedom
## AIC: 135271
##
## Number of Fisher Scoring iterations: 7
```

Notice the coefficients for road type 2 and 6 (one way street, single carriageway respectively) are on the same order as speed limit overall. This would suggest that perhaps a lower speed limit would be appropriate on thse particular road types.