draft

Exploratory analysis

```r
data <- read_csv("Car_Claims.csv")
```

```
## Rows: 10000 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, VEHICLE_YEAR, VEHICLE_...
## dbl (10): ID, CREDIT_SCORE, VEHICLE_OWNERSHIP, MARRIED, CHILDREN, ANNUAL_MIL...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
summary(data)
```

```
##        ID             AGE                GENDER          DRIVING_EXPERIENCE
##  Min.   :   101   Length:10000       Length:10000       Length:10000
##  1st Qu.:249639   Class :character   Class :character   Class :character
##  Median :501777   Mode  :character   Mode  :character   Mode  :character
##  Mean   :500522
##  3rd Qu.:753975
##  Max.   :999976
##
##   EDUCATION          CREDIT_SCORE     VEHICLE_OWNERSHIP VEHICLE_YEAR
##  Length:10000       Min.   :0.05336   Min.   :0.000     Length:10000
##  Class :character   1st Qu.:0.41719   1st Qu.:0.000     Class :character
##  Mode  :character   Median :0.52503   Median :1.000     Mode  :character
##                     Mean   :0.51581   Mean   :0.697
##                     3rd Qu.:0.61831   3rd Qu.:1.000
##                     Max.   :0.96082   Max.   :1.000
##                     NA's   :982
##     MARRIED          CHILDREN      ANNUAL_MILEAGE  VEHICLE_TYPE
##  Min.   :0.0000   Min.   :0.0000   Min.   : 2000   Length:10000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:10000   Class :character
##  Median :0.0000   Median :1.0000   Median :12000   Mode  :character
##  Mean   :0.4982   Mean   :0.6888   Mean   :11697
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:14000
##  Max.   :1.0000   Max.   :1.0000   Max.   :22000
##                                    NA's   :957
##  SPEEDING_VIOLATIONS PAST_ACCIDENTS     OUTCOME           CLAIMS
##  Min.   : 0.000      Min.   : 0.000   Min.   :0.0000   Min.   :     0
##  1st Qu.: 0.000      1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:     0
##  Median : 0.000      Median : 0.000   Median :0.0000   Median :     0
##  Mean   : 1.483      Mean   : 1.056   Mean   :0.3133   Mean   :  2329
##  3rd Qu.: 2.000      3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:  1748
##  Max.   :22.000      Max.   :15.000   Max.   :1.0000   Max.   :116328
##
```

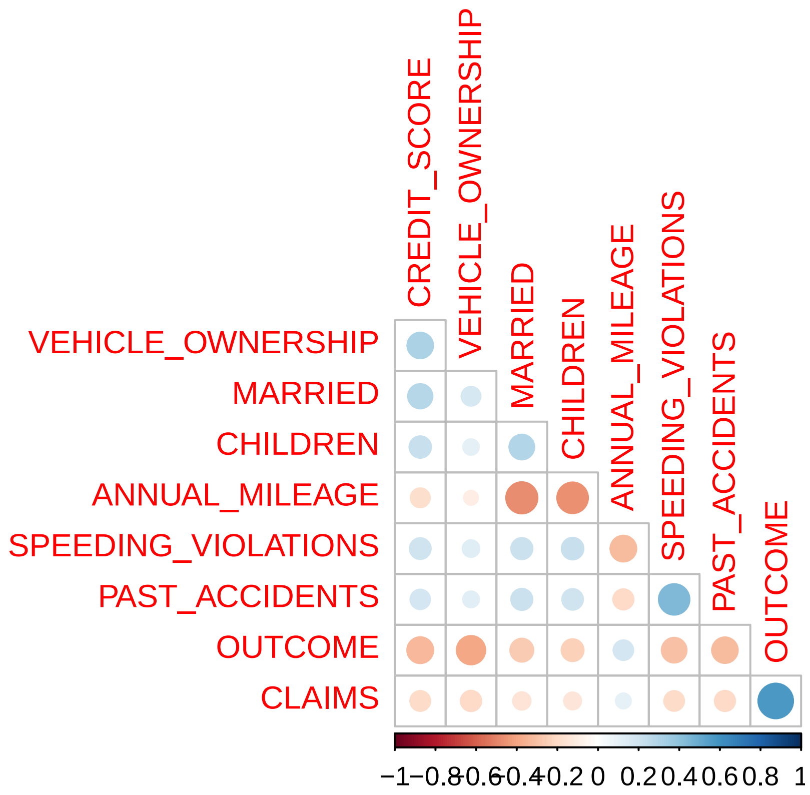
```r
# converting character covariates into factors
for (i in 1:length(data)){
  if (is.character(data[[i]])){
    data[[i]] <- as.factor(data[[i]])
  }
}
```

splitting data into train and test/validation

```r
set.seed(67)
train_indices <- sample(1:10000,8000)
train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]


train_data_x <- train_data %>% dplyr::select(-OUTCOME, -CLAIMS)
test_data_x <- test_data %>% dplyr::select(-OUTCOME, -CLAIMS)
```

correlation matrix of numeric covariates

```r
num_cols <-  data %>% select(where(is.numeric))

num_cols <-  num_cols[!is.na(num_cols$CREDIT_SCORE) &
                  !is.na(num_cols$ANNUAL_MILEAGE),]
num_cols <-  num_cols %>% select(-ID)
corrplot(cor(num_cols), type = "lower", diag = F)
```



- Annual mileage is lower for married ppl and ppl with children - past accidents has high corr with speeding violations ofc - vehicle owners are less likely to file claims?

investigating missing values

```r
print(paste("number of rows with missing annual mileage",
            sum(is.na(train_data$ANNUAL_MILEAGE)), "/8000"))
```

```
## [1] "number of rows with missing annual mileage 773 /8000"
```

```r
print(paste("number of rows with missing creditscore",
            sum(is.na(train_data$CREDIT_SCORE)), "/8000"))
```

```
## [1] "number of rows with missing creditscore 804 /8000"
```

```r
print(paste("number of rows missing both annual mileage and credit",
            sum(is.na(train_data$ANNUAL_MILEAGE) & is.na(train_data$CREDIT_SCORE)), "/8000"))
```

```
## [1] "number of rows missing both annual mileage and credit 76 /8000"
```

```r
print(paste("percentage of ppl of claim for nonempty annual mileage rows",
            mean(train_data[!is.na(train_data$ANNUAL_MILEAGE),"OUTCOME", drop = T])))
```

```
## [1] "percentage of ppl of claim for nonempty annual mileage rows 0.31451501314515"
```

```r
print(paste("percentage of ppl of claim for empty annual mileage rows",
            mean(train_data[is.na(train_data$ANNUAL_MILEAGE),"OUTCOME", drop = T])))
```

```
## [1] "percentage of ppl of claim for empty annual mileage rows 0.33764553686934"
```

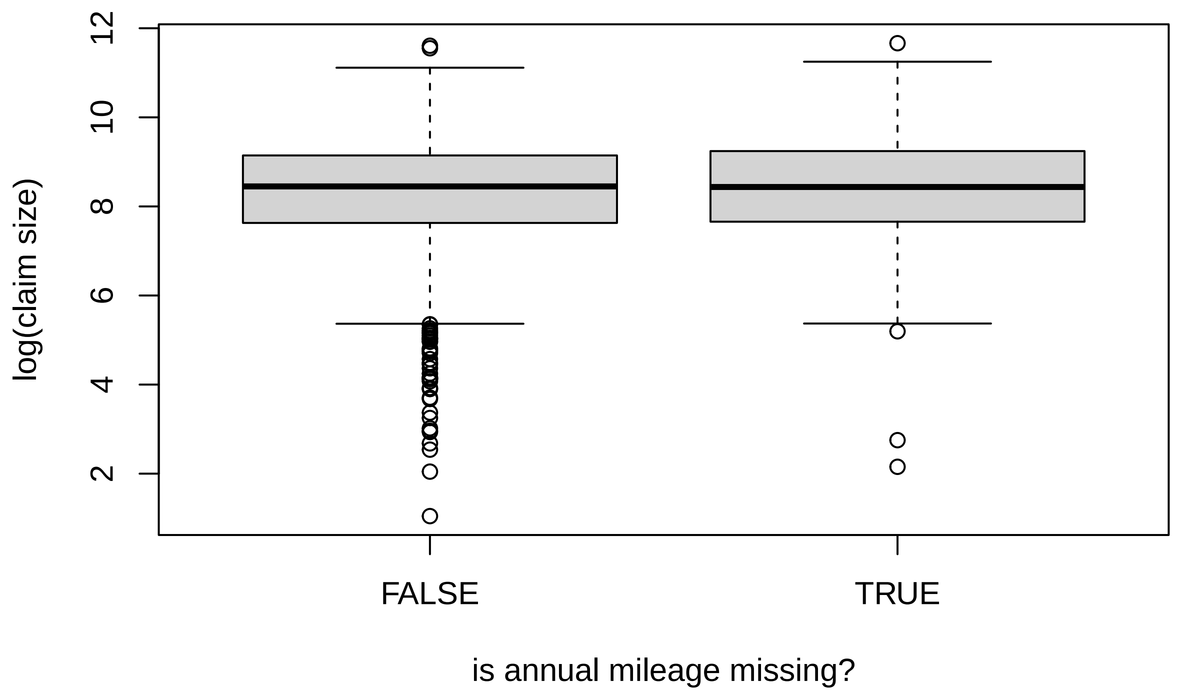
```r
print(paste("percentage of ppl of claim for nonempty creditscore rows",
            mean(train_data[!is.na(train_data$CREDIT_SCORE),"OUTCOME", drop = T])))
```

```
## [1] "percentage of ppl of claim for nonempty creditscore rows 0.317259588660367"
```

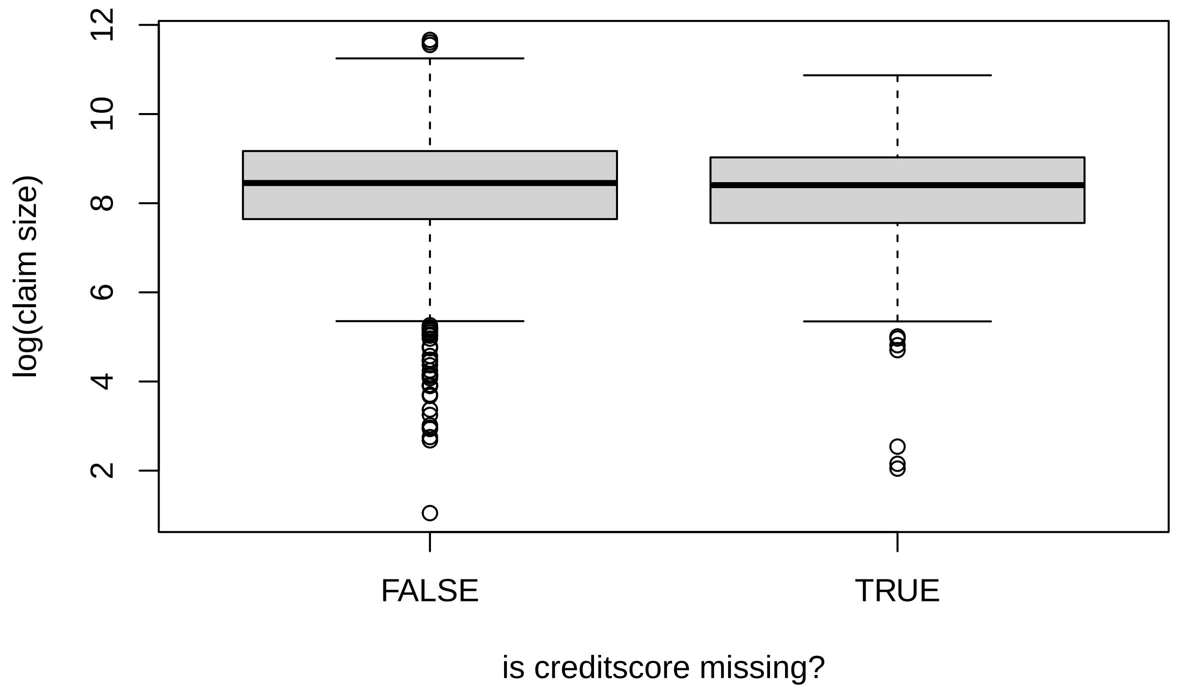
```r
print(paste("percentage of ppl of claim for empty creditscore rows",
            mean(train_data[is.na(train_data$CREDIT_SCORE),"OUTCOME", drop = T])))
```

```
## [1] "percentage of ppl of claim for empty creditscore rows 0.312189054726368"
```

```r
boxplot(log(train_data[train_data$CLAIMS>0,"CLAIMS", drop = T]) ~ is.na(train_data[train_data$CLAIMS>0,
        xlab = "is annual mileage missing?", ylab = "log(claim size)")
```



```r
boxplot(log(train_data[train_data$CLAIMS>0,"CLAIMS", drop = T]) ~ is.na(train_data[train_data$CLAIMS>0,
        xlab = "is creditscore missing?", ylab = "log(claim size)")
```



- Dont want to get rid of ~10% of the data, would be good to impute

Investigating if emptiness of the creditscore is independent to other variables i.e if creditscore is intentionally left empty - would expect younger ppl to have no credit score, so cells may be intentionally empty

```
print("TESTING FOR INDEPENDENCE OF CREDIT_SCORE EMPTINESS")
```

```
## [1] "TESTING FOR INDEPENDENCE OF CREDIT_SCORE EMPTINESS"
```

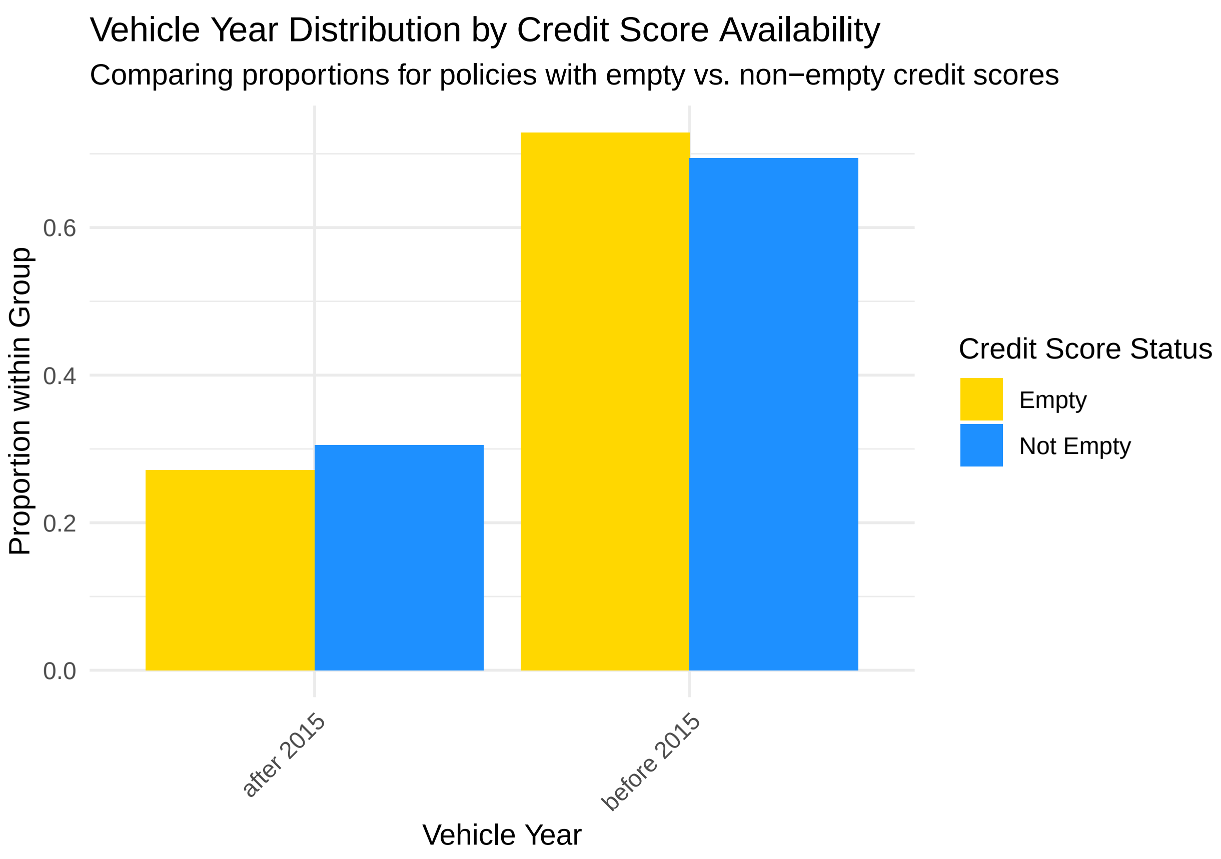
```
# Define the target variable: a binary indicator for missing credit scores
credit_score_missing <- is.na(train_data$CREDIT_SCORE)

# Get a list of all predictor variables to test, excluding ID and CREDIT_SCORE itself
predictors_to_test <- setdiff(names(train_data), c("ID", "CREDIT_SCORE"))

# Loop through each predictor and perform the appropriate test
for (var in predictors_to_test) {

  # Ignore any columns with no variation
  if (length(unique(train_data[[var]])) < 2) next

  # --- Test for association with CONTINUOUS variables ---
  if (is.numeric(train_data[[var]])) {
    # We use a t-test to see if the mean of the numeric variable is
    # different between the 'missing' and 'present' groups.
    test_result <- t.test(train_data[[var]] ~ credit_score_missing)
    p_value <- test_result$p.value
    cat(sprintf("Variable: %-20s | Test: T-test         | P-value: %.4f\n", var, p_value))

  # --- Test for association with CATEGORICAL variables ---
```

```r
  } else if (is.factor(train_data[[var]]) || is.character(train_data[[var]])) {
    # We use a Chi-squared test for independence between two categorical variables.
    # We add 'simulate.p.value = TRUE' to handle cases with low expected counts.
    test_result <- chisq.test(table(train_data[[var]], credit_score_missing), simulate.p.value = TRUE)
    p_value <- test_result$p.value
    cat(sprintf("Variable: %-20s | Test: Chi-squared    | P-value: %.4f\n", var, p_value))
  }
}
```

```
## Variable: AGE                  | Test: Chi-squared    | P-value: 0.1544
## Variable: GENDER               | Test: Chi-squared    | P-value: 0.9025
## Variable: DRIVING_EXPERIENCE   | Test: Chi-squared    | P-value: 0.1494
## Variable: EDUCATION            | Test: Chi-squared    | P-value: 0.5927
## Variable: VEHICLE_OWNERSHIP    | Test: T-test         | P-value: 0.1828
## Variable: VEHICLE_YEAR         | Test: Chi-squared    | P-value: 0.0395
## Variable: MARRIED              | Test: T-test         | P-value: 0.4997
## Variable: CHILDREN             | Test: T-test         | P-value: 0.4530
## Variable: ANNUAL_MILEAGE       | Test: T-test         | P-value: 0.6424
## Variable: VEHICLE_TYPE         | Test: Chi-squared    | P-value: 0.4308
## Variable: SPEEDING_VIOLATIONS  | Test: T-test         | P-value: 0.7981
## Variable: PAST_ACCIDENTS       | Test: T-test         | P-value: 0.9907
## Variable: OUTCOME              | Test: T-test         | P-value: 0.7688
## Variable: CLAIMS               | Test: T-test         | P-value: 0.1725
```

- pvalues very high, so likely that annual_mileage is left empty independently of other variables for all except vehicle year

```r
plot_data <- train_data %>%
  # Create a clear factor for whether the credit score is missing
  mutate(Credit_Score_Status = factor(ifelse(is.na(CREDIT_SCORE), "Empty", "Not Empty"))) %>%
  # Count the occurrences of each vehicle year for each status
  count(VEHICLE_YEAR, Credit_Score_Status) %>%
  # Group by the status so we can calculate proportions within each group
  group_by(Credit_Score_Status) %>%
  # Calculate the proportion
  mutate(Proportion = n / sum(n))


# --- 4. Create the Side-by-Side Bar Plot ---

ggplot(plot_data, aes(x = VEHICLE_YEAR, y = Proportion, fill = Credit_Score_Status)) +
  # geom_bar with stat="identity" uses the y-value directly.
  # position="dodge" places the bars next to each other.
  geom_bar(stat = "identity", position = "dodge") +

  labs(
    title = "Vehicle Year Distribution by Credit Score Availability",
    subtitle = "Comparing proportions for policies with empty vs. non-empty credit scores",
    x = "Vehicle Year",
    y = "Proportion within Group",
    fill = "Credit Score Status"
  ) +
  # Set the colors as requested
  scale_fill_manual(values = c("Empty" = "gold", "Not Empty" = "dodgerblue")) +

  # Improve readability
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



with no particular reason to take the conventional significance level of 0.05, conclude that the vehicle year is still independent to the emptiness of credit scores

- suggests that ppl with no credit score are not necessarily younger, so we assume that missing values are missing unintentionally

```
print("TESTING FOR INDEPENDENCE OF ANNUAL_MILEAGE EMPTINESS")
```

```
## [1] "TESTING FOR INDEPENDENCE OF ANNUAL_MILEAGE EMPTINESS"
```

```
AM_missing <- is.na(train_data$ANNUAL_MILEAGE)

# Get a list of all predictor variables to test, excluding ID and CREDIT_SCORE itself
predictors_to_test <- setdiff(names(train_data), c("ID", "ANNUAL_MILEAGE"))

# Loop through each predictor and perform the appropriate test
for (var in predictors_to_test) {

  # Ignore any columns with no variation
```

```r
  if (length(unique(train_data[[var]])) < 2) next

  # --- Test for association with CONTINUOUS variables ---
  if (is.numeric(train_data[[var]])) {
    # We use a t-test to see if the mean of the numeric variable is
    # different between the 'missing' and 'present' groups.
    test_result <- t.test(train_data[[var]] ~ AM_missing)
    p_value <- test_result$p.value
    cat(sprintf("Variable: %-20s | Test: T-test         | P-value: %.4f\n", var, p_value))

  # --- Test for association with CATEGORICAL variables ---
  } else if (is.factor(train_data[[var]]) || is.character(train_data[[var]])) {
    # We use a Chi-squared test for independence between two categorical variables.
    # We add 'simulate.p.value = TRUE' to handle cases with low expected counts.
    test_result <- chisq.test(table(train_data[[var]], AM_missing), simulate.p.value = TRUE)
    p_value <- test_result$p.value
    cat(sprintf("Variable: %-20s | Test: Chi-squared    | P-value: %.4f\n", var, p_value))
  }
}
```

```
## Variable: AGE                  | Test: Chi-squared    | P-value: 0.4203
## Variable: GENDER               | Test: Chi-squared    | P-value: 0.8166
## Variable: DRIVING_EXPERIENCE   | Test: Chi-squared    | P-value: 0.5192
## Variable: EDUCATION            | Test: Chi-squared    | P-value: 0.6092
## Variable: CREDIT_SCORE         | Test: T-test         | P-value: 0.3480
## Variable: VEHICLE_OWNERSHIP    | Test: T-test         | P-value: 0.5202
## Variable: VEHICLE_YEAR         | Test: Chi-squared    | P-value: 0.2689
## Variable: MARRIED              | Test: T-test         | P-value: 0.8048
## Variable: CHILDREN             | Test: T-test         | P-value: 0.4170
## Variable: VEHICLE_TYPE         | Test: Chi-squared    | P-value: 0.4728
## Variable: SPEEDING_VIOLATIONS  | Test: T-test         | P-value: 0.5506
## Variable: PAST_ACCIDENTS       | Test: T-test         | P-value: 0.0999
## Variable: OUTCOME              | Test: T-test         | P-value: 0.1960
## Variable: CLAIMS               | Test: T-test         | P-value: 0.1095
```

- pvalues very high, so likely that annual_mileage is left empty independently of other variables
- assume from here on that missing values are missing unintentionally

imputing missing values

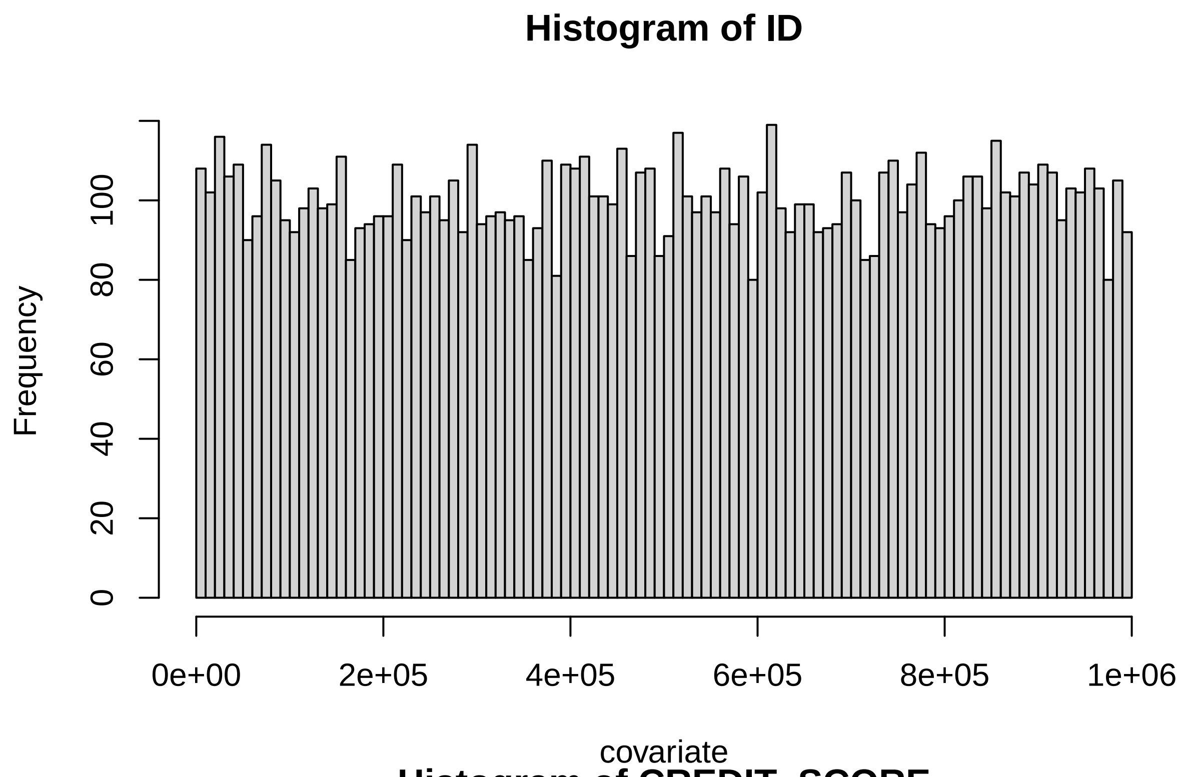
```r
# 1. Select all numeric columns from train and test sets
train_numeric <- train_data_x %>% select(where(is.numeric))
test_numeric  <- test_data_x %>% select(where(is.numeric))

# 2. Build recipe: Impute ONLY ANNUAL_MILEAGE and CREDIT_SCORE, using all numeric predictors
rec <- recipe(~ ., data = train_numeric) %>%
  step_impute_knn(c("ANNUAL_MILEAGE", "CREDIT_SCORE"))

# 3. Prep the recipe (fit on training data)
rec_prep <- prep(rec, training = train_numeric)
```

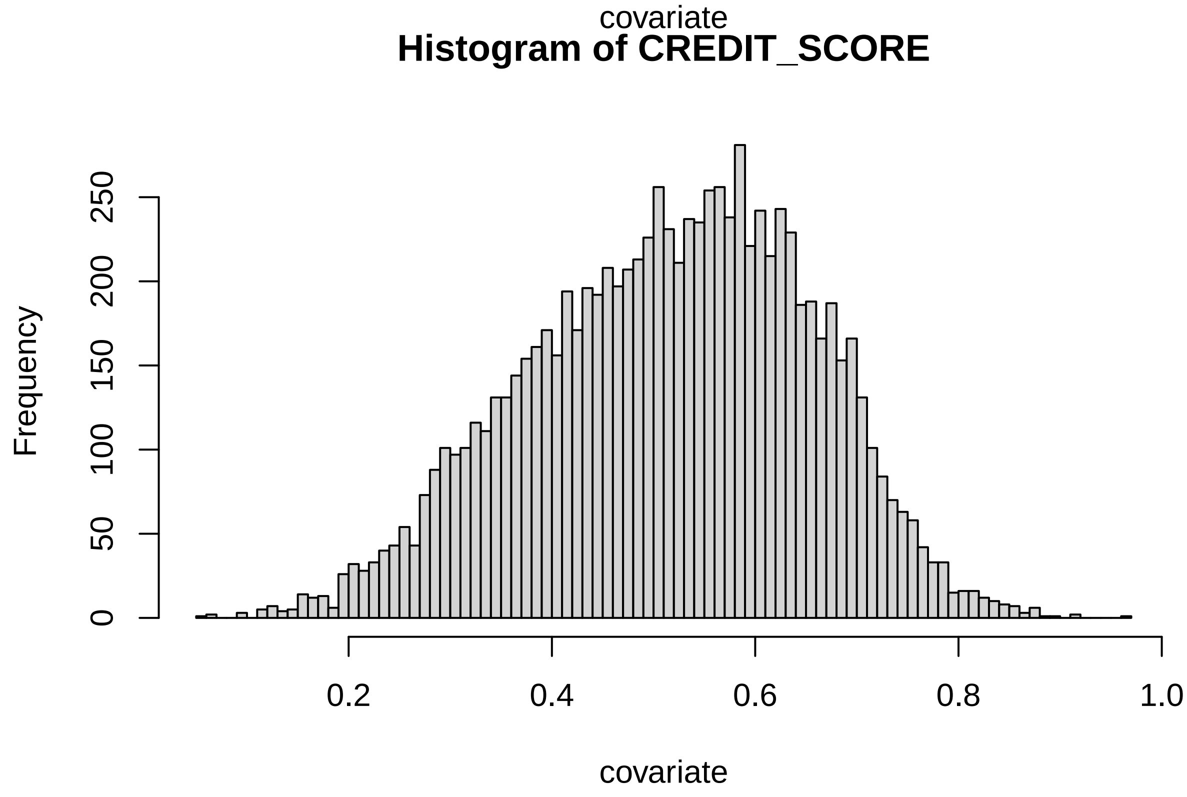
```r
# 4. Impute the training and test sets
train_imputed <- bake(rec_prep, new_data = train_numeric)
test_imputed  <- bake(rec_prep, new_data = test_numeric)

# The result: train_imputed and test_imputed contain all numeric columns,
# with only ANNUAL_MILEAGE and CREDIT_SCORE imputed, using relationships with all numeric variables.

train_data_imputed <- train_data
train_data_imputed$ANNUAL_MILEAGE <- train_imputed$ANNUAL_MILEAGE
train_data_imputed$CREDIT_SCORE <- train_imputed$CREDIT_SCORE

test_data_imputed <- test_data
test_data_imputed$ANNUAL_MILEAGE <- test_imputed$ANNUAL_MILEAGE
test_data_imputed$CREDIT_SCORE <- test_imputed$CREDIT_SCORE
```

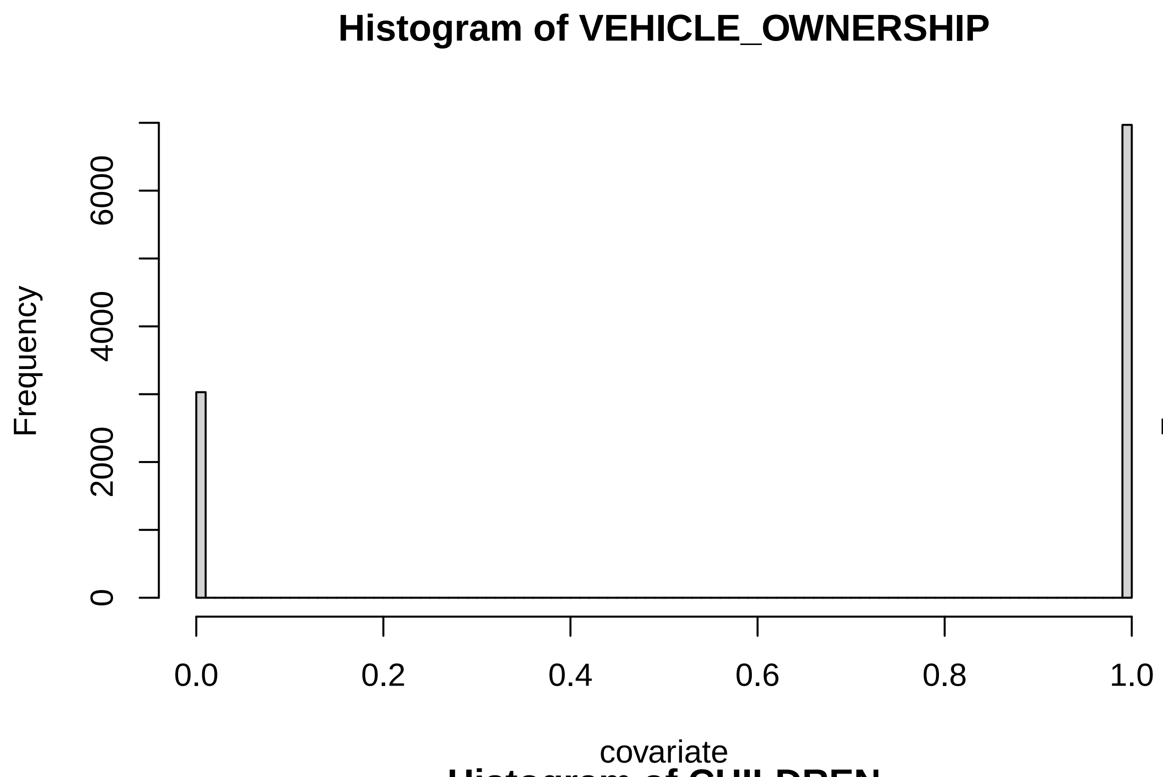
- used PCA between numeric variables to impute missing values
- wasnt possible to use the same model to also impute test data
- used knn imputation instead
- doesnt account for categorical variable values!

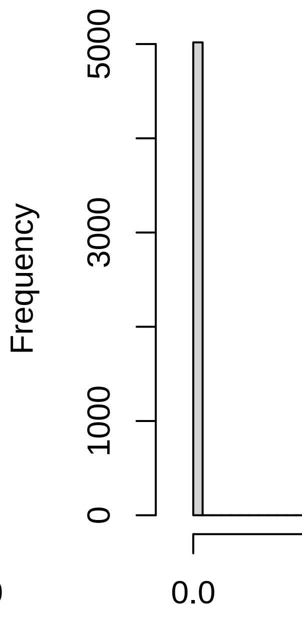
histogram of covariates

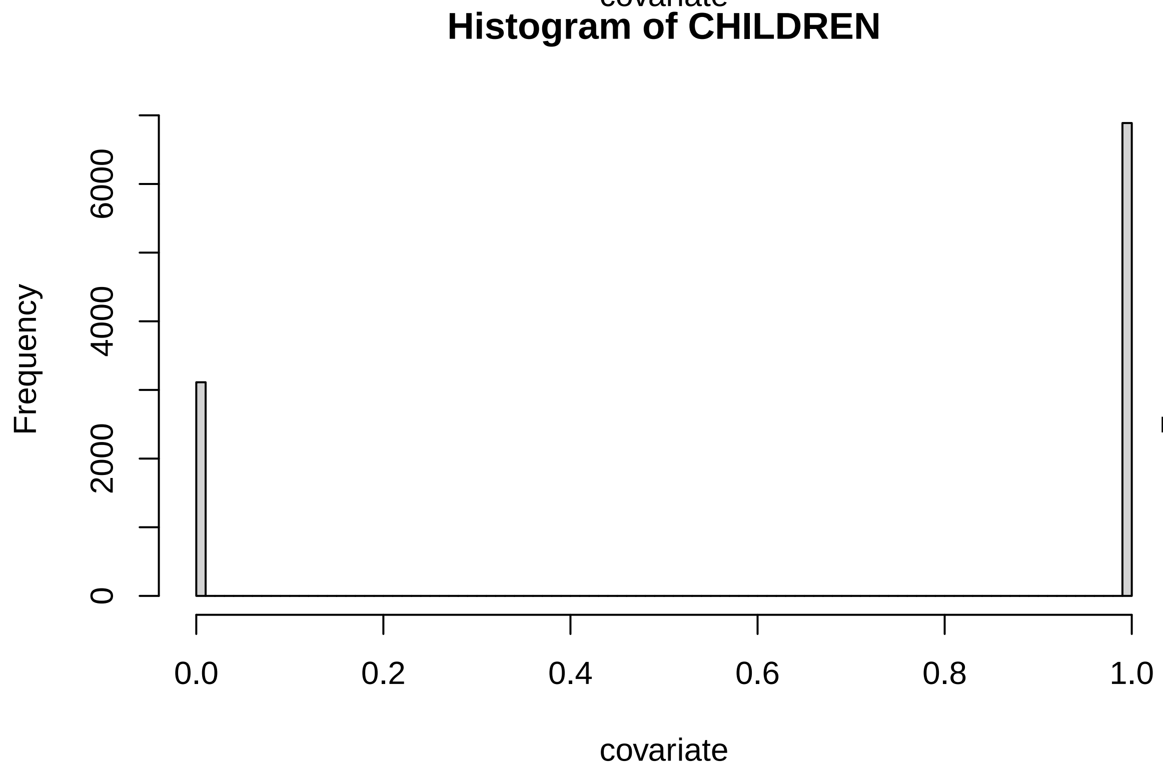
```r
for (i in 1:length(data)){
  covariate <- data[[i]]
  if (is.numeric(covariate)){
    hist(covariate, main = paste("Histogram of", names(data)[i] ), xlab = deparse(substitute(covariate)
  }
}
```

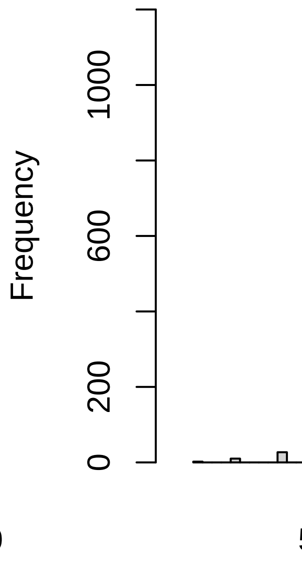

Histogram of ID
Frequency
0
20
40
60
80
100
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
covariate

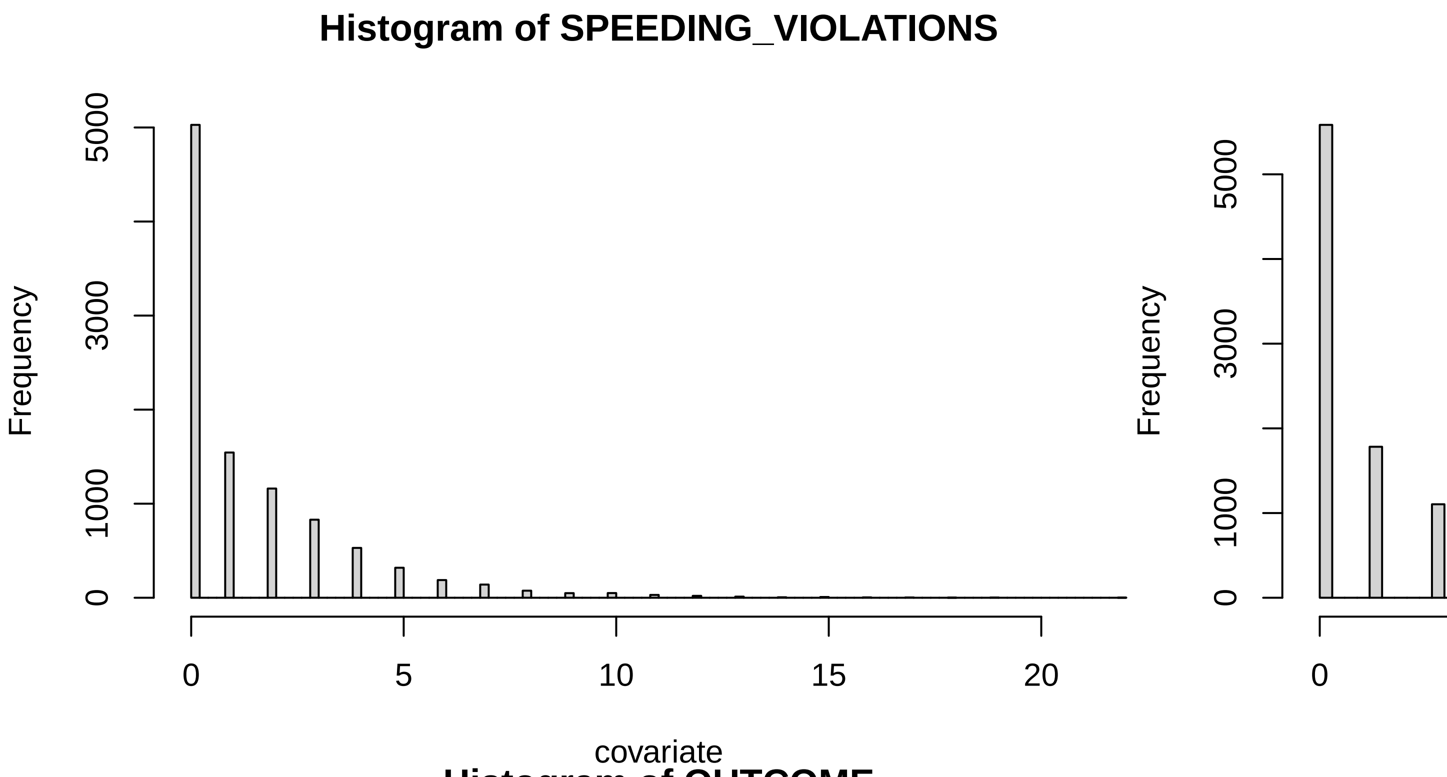

Histogram of CREDIT_SCORE
Frequency
0
50
100
150
200
250
0.2
0.4
0.6
0.8
1.0
covariate

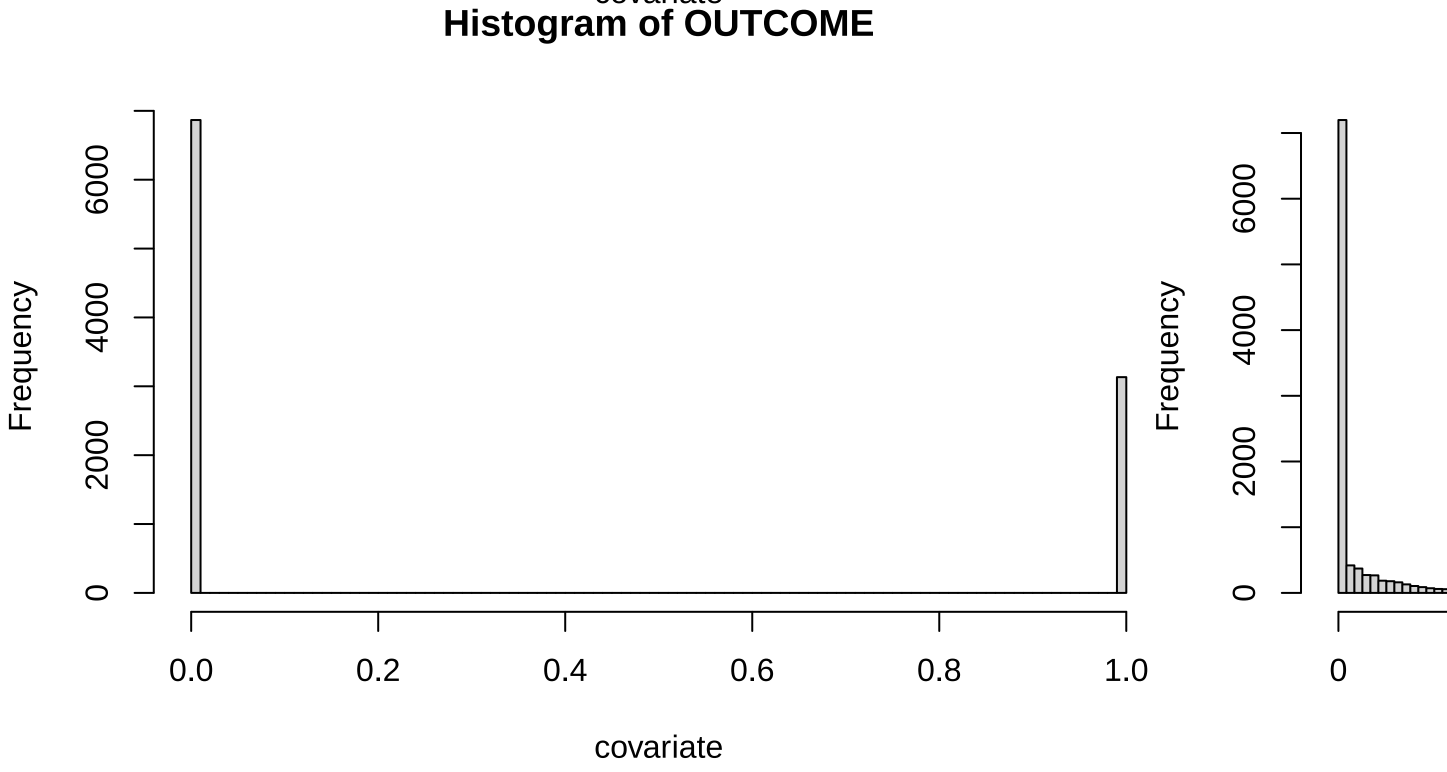

Histogram of VEHICLE_OWNERSHIP
Frequency
0
2000
4000
6000
0.0
0.2
0.4
0.6
0.8
1.0
covariate


Frequency
0
1000
3000
5000
0.0


Histogram of CHILDREN
Frequency
0
2000
4000
6000
0.0
0.2
0.4
0.6
0.8
1.0
covariate


Frequency
0
200
600
1000

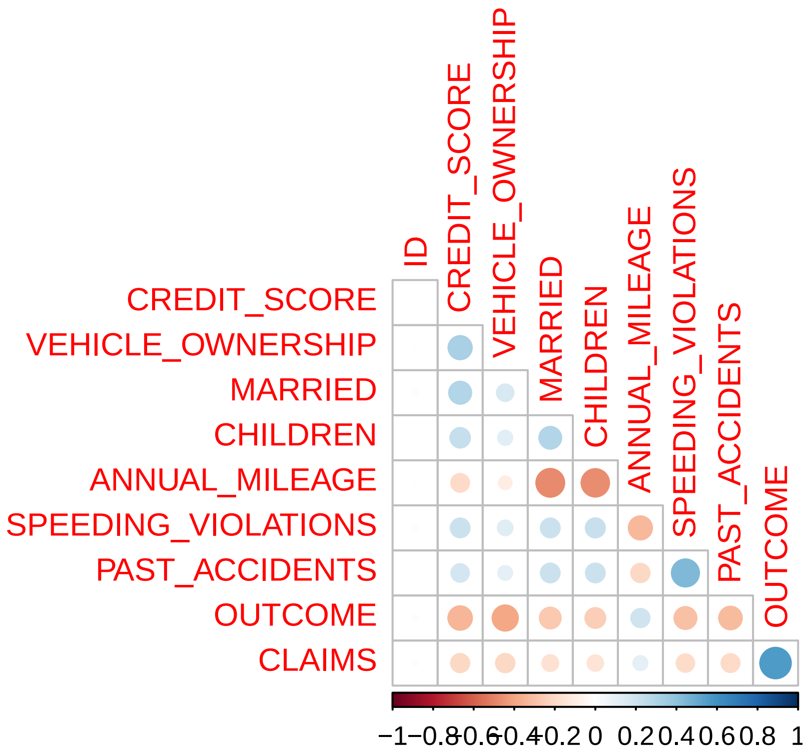




```r
summary(train_data)
```

```
##         ID             AGE          GENDER     DRIVING_EXPERIENCE
##  Min.   :   125   16-25:1611   female:3993   0-9y  :2822
##  1st Qu.:249025   26-39:2459   male  :4007   10-19y:2641
##  Median :500664   40-64:2339                  20-29y:1688
##  Mean   :499893   65+  :1591                  30y+  : 849
##  3rd Qu.:753723
##  Max.   :999976
##
##       EDUCATION     CREDIT_SCORE     VEHICLE_OWNERSHIP      VEHICLE_YEAR
##  high school:3335   Min.   :0.05336   Min.   :0.000     after 2015 :2416
##  none       :1524   1st Qu.:0.41637   1st Qu.:0.000     before 2015:5584
##  university :3141   Median :0.52456   Median :1.000
##                     Mean   :0.51528   Mean   :0.695
##                     3rd Qu.:0.61700   3rd Qu.:1.000
##                     Max.   :0.96082   Max.   :1.000
##                     NA's   :804
##     MARRIED           CHILDREN      ANNUAL_MILEAGE      VEHICLE_TYPE
##  Min.   :0.0000   Min.   :0.0000   Min.   : 3000   sedan     :7623
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:10000   sports car: 377
##  Median :0.0000   Median :1.0000   Median :12000
##  Mean   :0.4951   Mean   :0.6871   Mean   :11723
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:14000
##  Max.   :1.0000   Max.   :1.0000   Max.   :22000
##                                    NA's   :773
##  SPEEDING_VIOLATIONS PAST_ACCIDENTS      OUTCOME            CLAIMS
##  Min.   : 0.000      Min.   : 0.000   Min.   :0.0000   Min.   :     0
##  1st Qu.: 0.000      1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:     0
##  Median : 0.000      Median : 0.000   Median :0.0000   Median :     0
##  Mean   : 1.484      Mean   : 1.055   Mean   :0.3167   Mean   :  2360
##  3rd Qu.: 2.000      3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:  1838
##  Max.   :22.000      Max.   :15.000   Max.   :1.0000   Max.   :116328
##
```

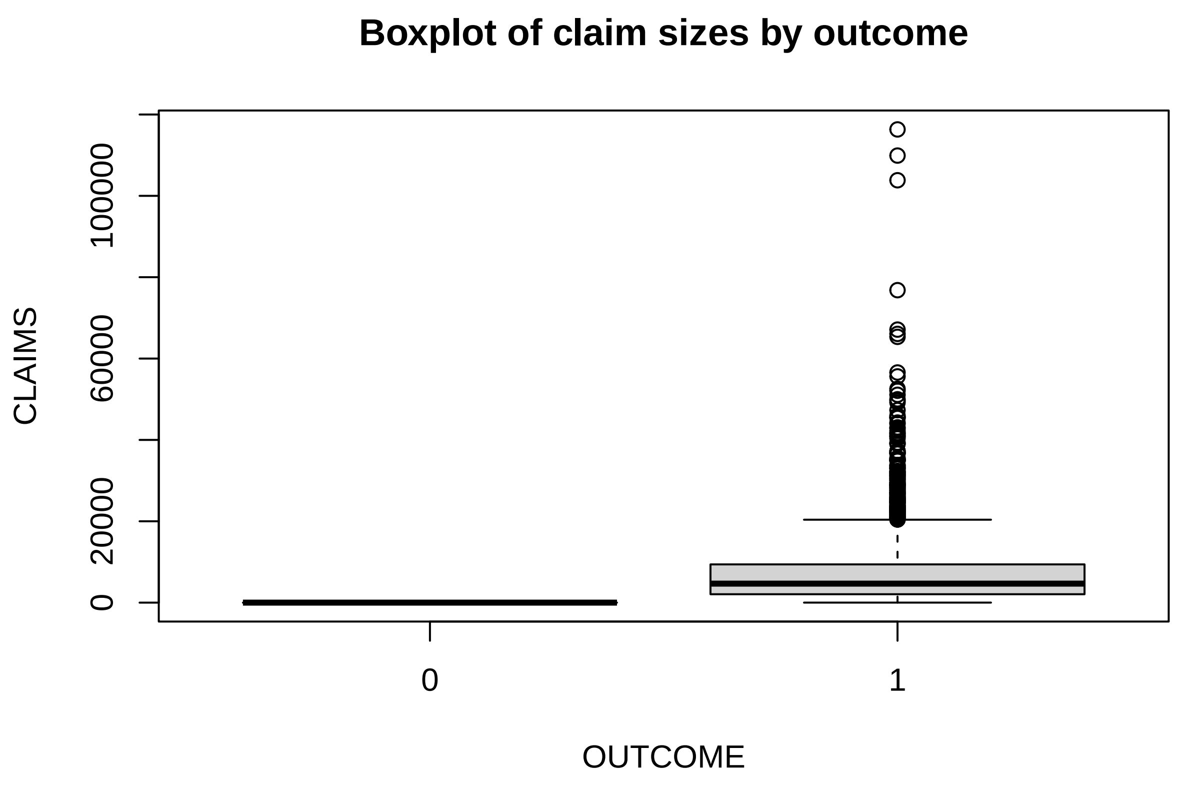
```r
num_cols = train_data_imputed %>% select_if(is.numeric)
cor_matrix = cor(num_cols)
corrplot(cor_matrix, type = "lower", diag = F)
```



- looks like annual mileage has a negative correlation with being married / having children - ppl with speeding violations are much more likely to have past accidents - surprisingly non vehicle owners have a higher correlation with making claims

investigation of claim sizes

```r
boxplot(train_data_imputed$CLAIMS ~ train_data_imputed$OUTCOME,
        main = "Boxplot of claim sizes by outcome", xlab = "OUTCOME", ylab = "CLAIMS")
```



```r
sum(train_data_imputed[train_data_imputed$OUTCOME == 0, "CLAIMS"])
```

```
## [1] 0
```

```r
sum(train_data_imputed[train_data_imputed$CLAIMS == 0, "OUTCOME"])
```

```
## [1] 0
```

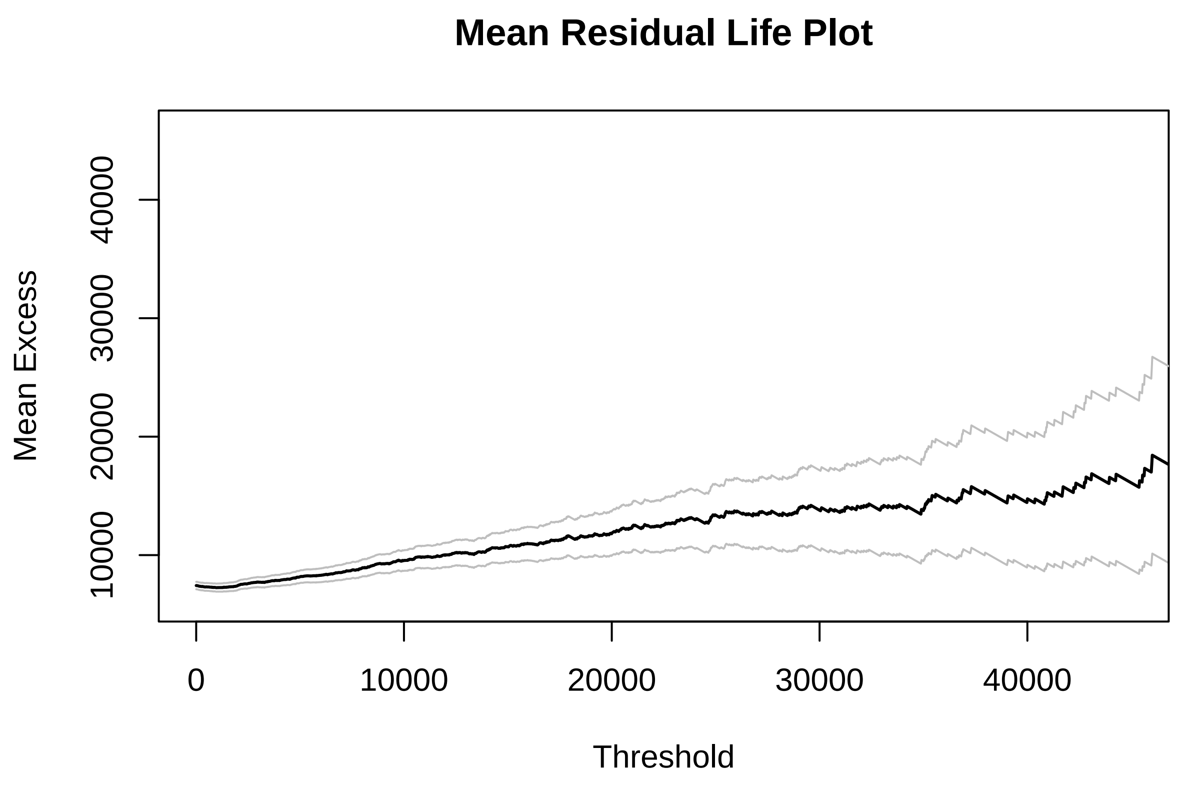
- Looks like there are no cases where outcome = 1 and there is a claimsize of 0
- suggests that claims sizes are always nonzero and zero claims should be accounted for within the counts distribution

Extreme value analysis of claimsizes

```r
nonzerosizes <- data[data$OUTCOME>0, "CLAIMS", drop = T]
quantile(nonzerosizes, c(0.95,0.97, 0.99, 0.995))
```

```
##      95%      97%      99%    99.5%
## 23303.06 29080.47 42786.68 51570.25
```

```r
# mean excess plot
mrlplot(nonzerosizes, xlim = c(0,45000))
```

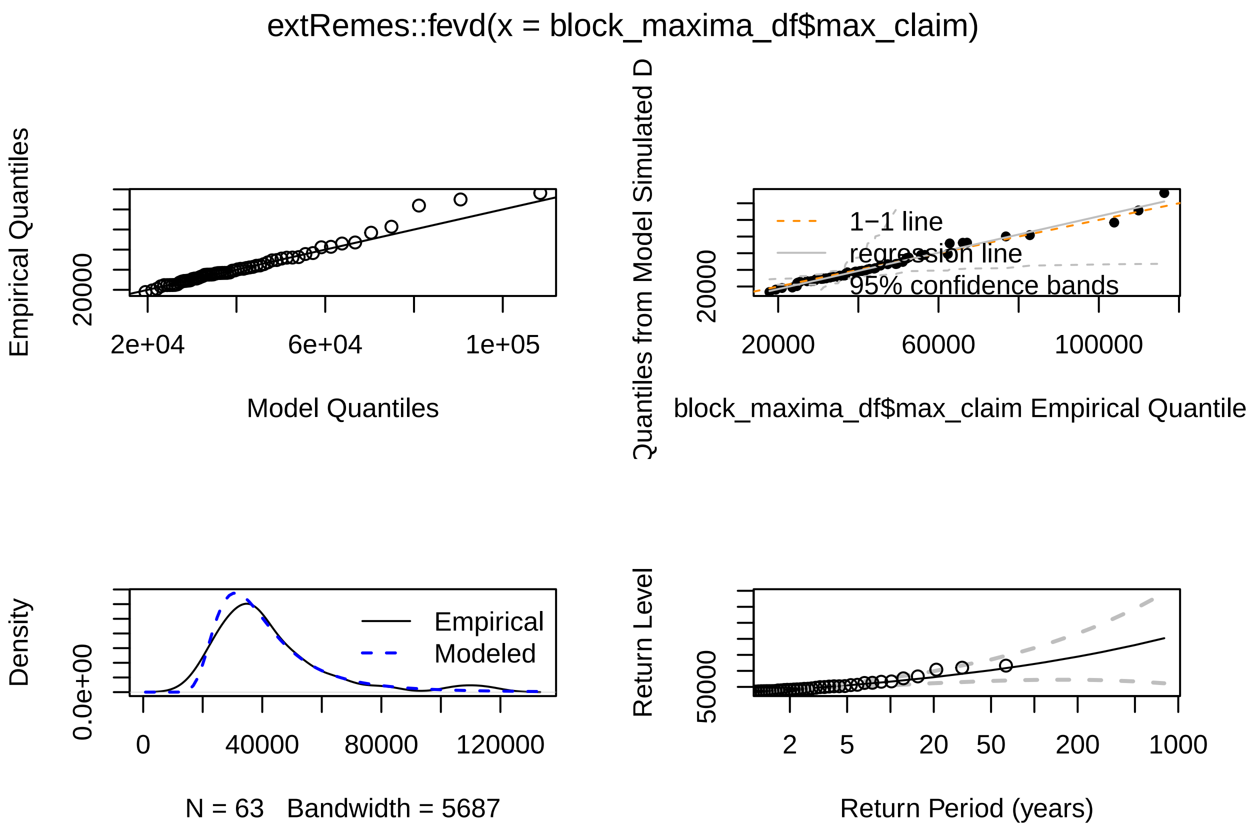


- Mean excess function is rising, clearly a heavy tailed distribution

```
# Fitting GEV distribution
block_size <- 50
blocks <- ceiling(seq_along(nonzerosizes) / block_size)
GEVdf <- data.frame(nonzerosizes, blocks)
block_maxima_df <- GEVdf %>%
  group_by(blocks) %>%
  summarise(
    max_claim = max(nonzerosizes)
  )
gev_fit <- extRemes::fevd(block_maxima_df$max_claim)
summary(gev_fit)
```

```
##
## extRemes::fevd(x = block_maxima_df$max_claim)
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  695.5335
##
##
##  Estimated parameters:
##     location        scale        shape
## 3.312419e+04 1.112776e+04 2.191021e-01
##
##  Standard Error Estimates:
##    location       scale       shape
## 1591.904652 1247.447031    0.101009
##
##  Estimated parameter covariance matrix.
##                 location         scale         shape
## location 2534160.42263  1.131005e+06 -50.31878400
## scale    1131004.89478  1.556124e+06  -6.45730616
## shape        -50.31878 -6.457306e+00   0.01020283
##
##  AIC = 1397.067
##
##  BIC = 1403.496
```

```r
plot(gev_fit)
```



```r
# Hypothesis test with H0: shape = 0
# Assuming a normal distribution of the shape parameter estimator
shape.estimate <- 0.219102
shape.stderr <- 0.101009

print(shape.estimate/shape.stderr)
```

```
## [1] 2.169133
```

- clearly, the shape parameter is significantly different from 0, so we can reject the null hypothesis of an exponential tail. It is quite obvious that we have a frechet type distribution of the nonzero claim sizess
- Suggests use of frechet family
- Could use gamma or lognormal from the exponential family

lognormal model

```r
# lognormal
# lm is equivalent to fitting glm(., gaussian(link = identity))
lognormal_claimsize_df <- train_data_imputed[train_data_imputed$OUTCOME >0,] %>%
  select(-OUTCOME)
names(train_data_imputed)
```

```
##  [1] "ID"                  "AGE"                 "GENDER"
##  [4] "DRIVING_EXPERIENCE"  "EDUCATION"           "CREDIT_SCORE"
##  [7] "VEHICLE_OWNERSHIP"   "VEHICLE_YEAR"        "MARRIED"
## [10] "CHILDREN"            "ANNUAL_MILEAGE"      "VEHICLE_TYPE"
## [13] "SPEEDING_VIOLATIONS" "PAST_ACCIDENTS"      "OUTCOME"
## [16] "CLAIMS"
```

```r
lognormal_claimsize_df$CLAIMS <- log(lognormal_claimsize_df$CLAIMS)

# base model no interactions
claimsize.0.lognormal <- lm(CLAIMS ~.,
                   data = lognormal_claimsize_df)


names(train_data_x)
```

```
##  [1] "ID"                  "AGE"                 "GENDER"
##  [4] "DRIVING_EXPERIENCE"  "EDUCATION"           "CREDIT_SCORE"
##  [7] "VEHICLE_OWNERSHIP"   "VEHICLE_YEAR"        "MARRIED"
## [10] "CHILDREN"            "ANNUAL_MILEAGE"      "VEHICLE_TYPE"
## [13] "SPEEDING_VIOLATIONS" "PAST_ACCIDENTS"
```

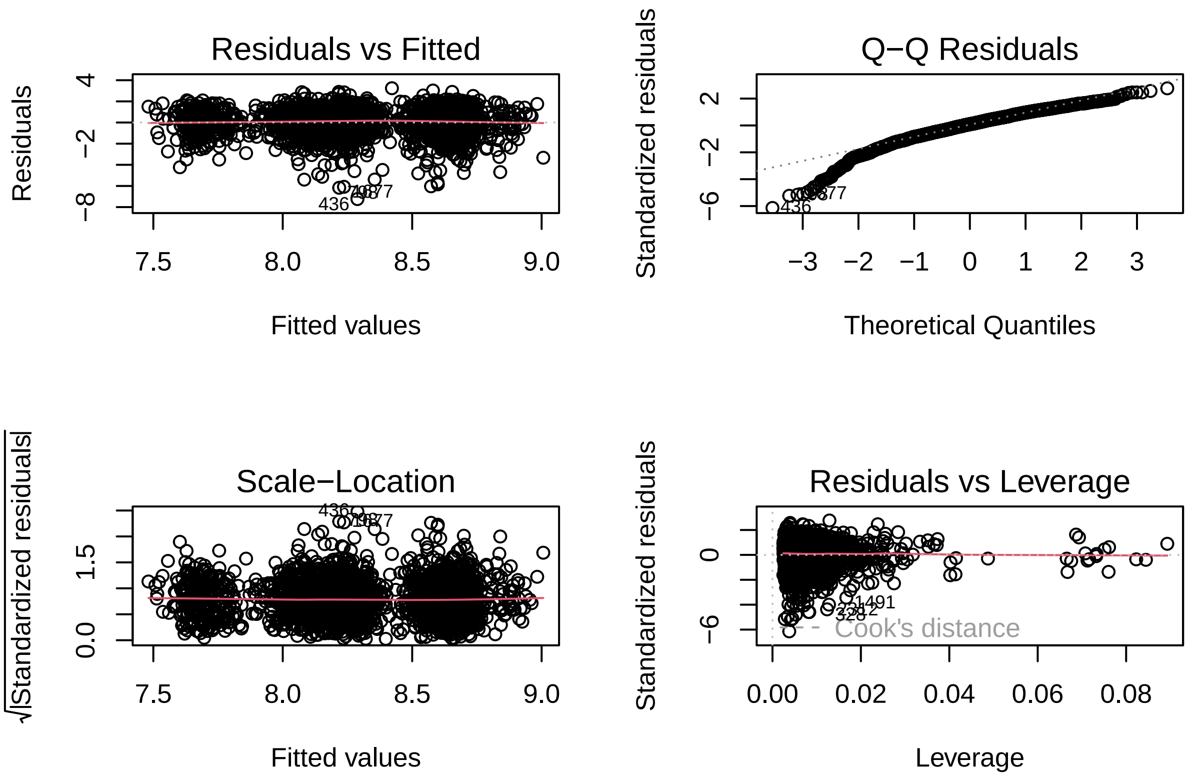
```r
# model with interactions
claimsize.1.lognormal <- lm(CLAIMS ~(AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION +
                             CREDIT_SCORE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
                             MARRIED + CHILDREN + ANNUAL_MILEAGE +
                             VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2,
                   data = lognormal_claimsize_df)
```

```r
par(mfrow = c(2,2))
print('---------------------FULL MODEL NO INTERACTION TERMS------------------------')
```

```
## [1] "---------------------FULL MODEL NO INTERACTION TERMS------------------------"
```

```r
summary(claimsize.0.lognormal);plot(claimsize.0.lognormal)
```

```
##
## Call:
## lm(formula = CLAIMS ~ ., data = lognormal_claimsize_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2413 -0.6461  0.1103  0.7844  3.2424
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              8.069e+00  2.061e-01  39.150  < 2e-16 ***
## ID                      -2.414e-08  8.025e-08  -0.301  0.76355
## AGE26-39                 2.899e-03  7.042e-02   0.041  0.96716
## AGE40-64                 1.707e-02  8.819e-02   0.194  0.84657
## AGE65+                  -9.327e-02  1.220e-01  -0.764  0.44470
## GENDERmale               3.992e-01  4.998e-02   7.986 2.10e-15 ***
## DRIVING_EXPERIENCE10-19y -5.834e-01  8.632e-02  -6.758 1.73e-11 ***
## DRIVING_EXPERIENCE20-29y -5.004e-01  1.748e-01  -2.863  0.00424 **
## DRIVING_EXPERIENCE30y+   -1.203e-01  3.363e-01  -0.358  0.72070
## EDUCATIONnone           -5.531e-03  5.795e-02  -0.095  0.92397
## EDUCATIONuniversity     -3.734e-02  6.044e-02  -0.618  0.53678
## CREDIT_SCORE             2.333e-01  2.222e-01   1.050  0.29384
## VEHICLE_OWNERSHIP        5.863e-02  5.046e-02   1.162  0.24542
## VEHICLE_YEARbefore 2015  -1.502e-02  8.035e-02  -0.187  0.85168
## MARRIED                 -6.217e-02  5.984e-02  -1.039  0.29891
## CHILDREN                -1.120e-02  5.505e-02  -0.203  0.83886
## ANNUAL_MILEAGE           8.497e-06  1.066e-05   0.797  0.42545
## VEHICLE_TYPEsports car   2.300e-01  1.089e-01   2.112  0.03477 *
## SPEEDING_VIOLATIONS      5.875e-03  2.546e-02   0.231  0.81754
## PAST_ACCIDENTS           2.089e-02  3.748e-02   0.557  0.57736
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.181 on 2514 degrees of freedom
## Multiple R-squared:  0.06563,    Adjusted R-squared:  0.05856
## F-statistic: 9.293 on 19 and 2514 DF,  p-value: < 2.2e-16
```



```
print('----------------WHITE TEST-----------------------------')
```

```
## [1] "----------------WHITE TEST-----------------------------"
```

```
white_test(claimsize.0.lognormal)
```

```
## White's test results
##
## Null hypothesis: Homoskedasticity of the residuals
## Alternative hypothesis: Heteroskedasticity of the residuals
## Test Statistic: 0.45
## P-value: 0.797812
```

```
print('-----------------VIF----------------------------')
```

```
## [1] "-----------------VIF----------------------------"
```

```
vif(claimsize.0.lognormal, type = "predictor")
```

```
## GVIFs computed for predictors
```

```
##                         GVIF Df GVIF^(1/(2*Df)) Interacts With
## ID                  1.005306  1        1.002650             --
## AGE                 3.033224  3        1.203143             --
## GENDER              1.102305  1        1.049907             --
## DRIVING_EXPERIENCE  4.569123  3        1.288167             --
## EDUCATION           1.284516  2        1.064596           --
## CREDIT_SCORE        1.491735  1        1.221366           --
## VEHICLE_OWNERSHIP   1.135285  1        1.065498           --
## VEHICLE_YEAR        1.068256  1        1.033565           --
## MARRIED             1.365148  1        1.168396           --
## CHILDREN            1.372683  1        1.171616           --
## ANNUAL_MILEAGE      1.594910  1        1.262897           --
## VEHICLE_TYPE        1.009676  1        1.004826           --
## SPEEDING_VIOLATIONS 2.000328  1        1.414329           --
## PAST_ACCIDENTS      1.657310  1        1.287366           --
##
## ID                  AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_OWNERSHIP, VEHI
## AGE                  ID, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_OWNERSHIP, VEHI
## GENDER                  ID, AGE, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_OWNERSHIP, VEHI
## DRIVING_EXPERIENCE                  ID, AGE, GENDER, EDUCATION, CREDIT_SCORE, VEHICLE_OWNERSHIP, VEHI
## EDUCATION                  ID, AGE, GENDER, DRIVING_EXPERIENCE, CREDIT_SCORE, VEHICLE_OWNERSHIP, VEHI
## CREDIT_SCORE                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, VEHICLE_OWNERSHIP, VEHI
## VEHICLE_OWNERSHIP                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHI
## VEHICLE_YEAR                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_(
## MARRIED                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_OWNERS
## CHILDREN                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_OWNEI
## ANNUAL_MILEAGE                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLI
## VEHICLE_TYPE                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLE_(
## SPEEDING_VIOLATIONS                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VI
## PAST_ACCIDENTS                  ID, AGE, GENDER, DRIVING_EXPERIENCE, EDUCATION, CREDIT_SCORE, VEHICLI
```

- good fit around the middle, but the fit near the tails suffer
- doesnt seem heteroskedastic, linear model is appropriate in that regards
- introducing interaction terms increases adj Rsq
- not much multicollinearity between variables w no interaction terms

Variable selection

- only investigate adjr2 due to the size of the predictorspace

```r
# forward selection
# not running because it takes too long to knit
print('----------------------FORWARD----------------------')
lognormal_fwdselection <- ols_step_forward_adj_r2(claimsize.1.lognormal)
lognormal_fwdselection
```

```r
# backward selection
print('----------------------backward----------------------')
```

```
## [1] "----------------------backward----------------------"
```

```r
print('adjr2')
```

```
## [1] "adjr2"
```

```r
lognormal_bwdselection <- stats::step(claimsize.1.lognormal, direction = "backward", trace = 0)
summary(lognormal_bwdselection)
```

```
##
## Call:
## lm(formula = CLAIMS ~ GENDER + DRIVING_EXPERIENCE + EDUCATION +
##     VEHICLE_OWNERSHIP + VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE +
##     VEHICLE_TYPE + SPEEDING_VIOLATIONS + GENDER:CHILDREN + GENDER:SPEEDING_VIOLATIONS +
##     EDUCATION:ANNUAL_MILEAGE + VEHICLE_OWNERSHIP:VEHICLE_YEAR +
##     VEHICLE_YEAR:MARRIED + CHILDREN:ANNUAL_MILEAGE + ANNUAL_MILEAGE:SPEEDING_VIOLATIONS,
##     data = lognormal_claimsize_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2824 -0.6446  0.1127  0.7990  3.1821
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             8.985e+00  2.733e-01  32.875
## GENDERmale                              2.794e-01  7.020e-02   3.980
## DRIVING_EXPERIENCE10-19y               -5.679e-01  6.927e-02  -8.198
## DRIVING_EXPERIENCE20-29y               -4.499e-01  1.547e-01  -2.908
## DRIVING_EXPERIENCE30y+                 -9.899e-02  3.287e-01  -0.301
## EDUCATIONnone                          -6.055e-01  2.683e-01  -2.257
## EDUCATIONuniversity                     1.865e-02  2.595e-01   0.072
## VEHICLE_OWNERSHIP                      -2.356e-01  1.502e-01  -1.569
## VEHICLE_YEARbefore 2015                -7.289e-02  1.182e-01  -0.617
## MARRIED                                 1.485e-01  1.539e-01   0.965
## CHILDREN                               -9.208e-01  2.632e-01  -3.499
## ANNUAL_MILEAGE                         -4.430e-05  1.762e-05  -2.515
## VEHICLE_TYPEsports car                  2.277e-01  1.082e-01   2.105
## SPEEDING_VIOLATIONS                    -1.705e-01  8.576e-02  -1.988
## GENDERmale:CHILDREN                     1.633e-01  9.654e-02   1.691
## GENDERmale:SPEEDING_VIOLATIONS          7.766e-02  5.128e-02   1.514
## EDUCATIONnone:ANNUAL_MILEAGE            4.567e-05  2.056e-05   2.222
## EDUCATIONuniversity:ANNUAL_MILEAGE     -3.995e-06  2.047e-05  -0.195
## VEHICLE_OWNERSHIP:VEHICLE_YEARbefore 2015  3.365e-01  1.578e-01   2.132
## VEHICLE_YEARbefore 2015:MARRIED        -2.333e-01  1.608e-01  -1.452
## CHILDREN:ANNUAL_MILEAGE                 6.365e-05  1.962e-05   3.243
## ANNUAL_MILEAGE:SPEEDING_VIOLATIONS      1.140e-05  6.538e-06   1.743
##                                        Pr(>|t|)
## (Intercept)                             < 2e-16 ***
## GENDERmale                             7.08e-05 ***
## DRIVING_EXPERIENCE10-19y               3.86e-16 ***
## DRIVING_EXPERIENCE20-29y               0.003664 **
## DRIVING_EXPERIENCE30y+                 0.763314
## EDUCATIONnone                          0.024092 *
## EDUCATIONuniversity                    0.942719
## VEHICLE_OWNERSHIP                      0.116844
## VEHICLE_YEARbefore 2015                0.537372
## MARRIED                                0.334749
## CHILDREN                               0.000476 ***
## ANNUAL_MILEAGE                         0.011963 *
## VEHICLE_TYPEsports car                      0.035401 *
## SPEEDING_VIOLATIONS                         0.046959 *
## GENDERmale:CHILDREN                         0.090965 .
## GENDERmale:SPEEDING_VIOLATIONS              0.130076
## EDUCATIONnone:ANNUAL_MILEAGE                0.026385 *
## EDUCATIONuniversity:ANNUAL_MILEAGE          0.845321
## VEHICLE_OWNERSHIP:VEHICLE_YEARbefore 2015 0.033100 *
## VEHICLE_YEARbefore 2015:MARRIED             0.146762
## CHILDREN:ANNUAL_MILEAGE                     0.001198 **
## ANNUAL_MILEAGE:SPEEDING_VIOLATIONS          0.081382 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.175 on 2512 degrees of freedom
## Multiple R-squared:  0.07705,    Adjusted R-squared:  0.06934
## F-statistic: 9.986 on 21 and 2512 DF,  p-value: < 2.2e-16
```

- improvement in adjusted R squared from full model
- backward selection models have more variables but higher adjRsq
- Select out of backward selection models for this
- all seem to have same adj R sq and AIC, choose one with the lowest number of variables, i.e the model chosen by prioritising AIC

```r
# backward stepwaise
lognormal_fwdselection<- stats::step(claimsize.1.lognormal, direction = "forward", trace = 0)
summary(lognormal_fwdselection)
```

- backward selected model is chosen as the reduced model due to the increased adjusted R squared and parsimony

```r
claimsize.2.lognormal <- lognormal_bwdselection
```

Significance test

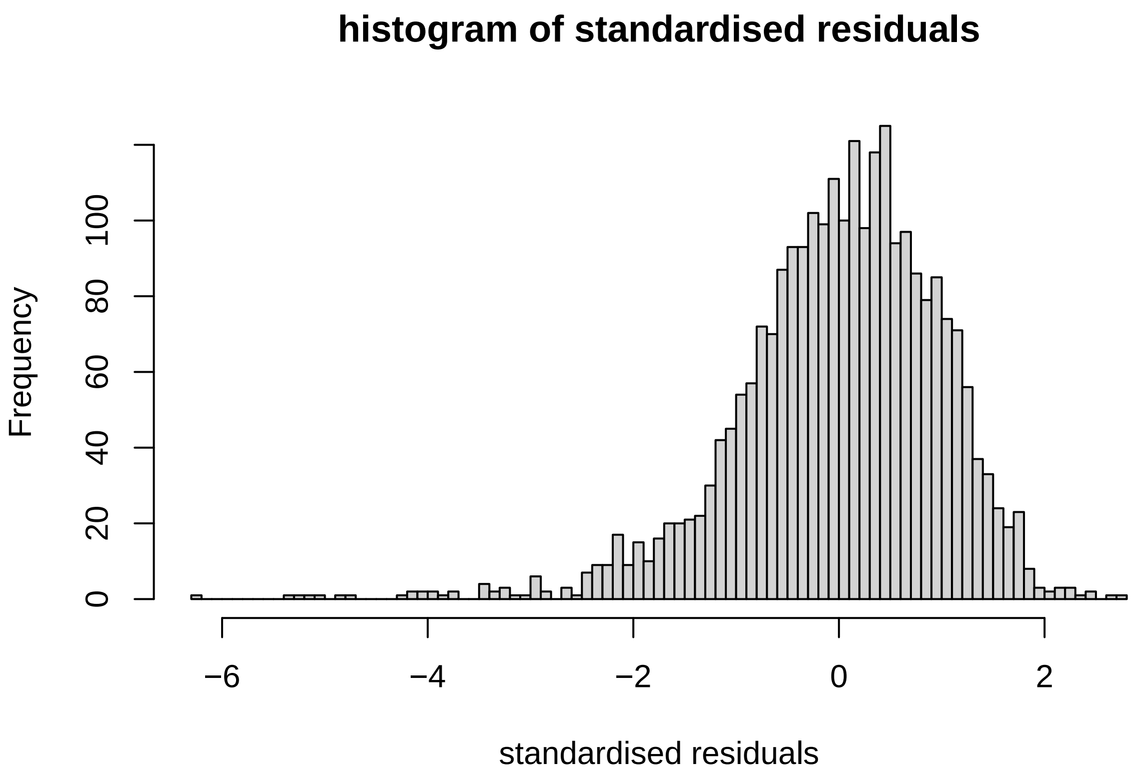
```r
# between full model with all interaction terms and backward stepwise model
anova(claimsize.2.lognormal,claimsize.1.lognormal)
```

```
## Analysis of Variance Table
##
## Model 1: CLAIMS ~ GENDER + DRIVING_EXPERIENCE + EDUCATION + VEHICLE_OWNERSHIP +
##     VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE + VEHICLE_TYPE +
##     SPEEDING_VIOLATIONS + GENDER:CHILDREN + GENDER:SPEEDING_VIOLATIONS +
##     EDUCATION:ANNUAL_MILEAGE + VEHICLE_OWNERSHIP:VEHICLE_YEAR +
##     VEHICLE_YEAR:MARRIED + CHILDREN:ANNUAL_MILEAGE + ANNUAL_MILEAGE:SPEEDING_VIOLATIONS
## Model 2: CLAIMS ~ (AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION + CREDIT_SCORE +
##     VEHICLE_OWNERSHIP + VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE +
##     VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2
##   Res.Df    RSS  Df Sum of Sq      F Pr(>F)
## 1   2512 3465.6
## 2   2379 3338.8 133    126.87 0.6797 0.9979
```

- Clearly, the dropped variables were not that significant

Residual analysis

```
# plot of type 1 standardised residuals
hist(summary(claimsize.2.lognormal)$res/ summary(claimsize.2.lognormal)$sigma, breaks = 100,
     main = "histogram of standardised residuals", xlab = "standardised residuals")
```



- looks left skewed, suggests havent taken account of patterns in the data, possibly due to poor treatment of tail

Gamma model

- consider common interaction terms and interaction terms suggested by correlation analysis

```
claimsize_df <- train_data_imputed[train_data_imputed$OUTCOME >0,] %>%
  select(-OUTCOME)
```

```
# base model no interaction
claimsize.0.gamma <- glm(CLAIMS ~. , family = Gamma(link = "log"),
                     data = claimsize_df)

# w interactions
claimsize.1.gamma <- glm(CLAIMS ~(AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION +
                             CREDIT_SCORE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
                             MARRIED + CHILDREN + ANNUAL_MILEAGE +
                             VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2, family = Gamma(lin
                     data = claimsize_df)
```

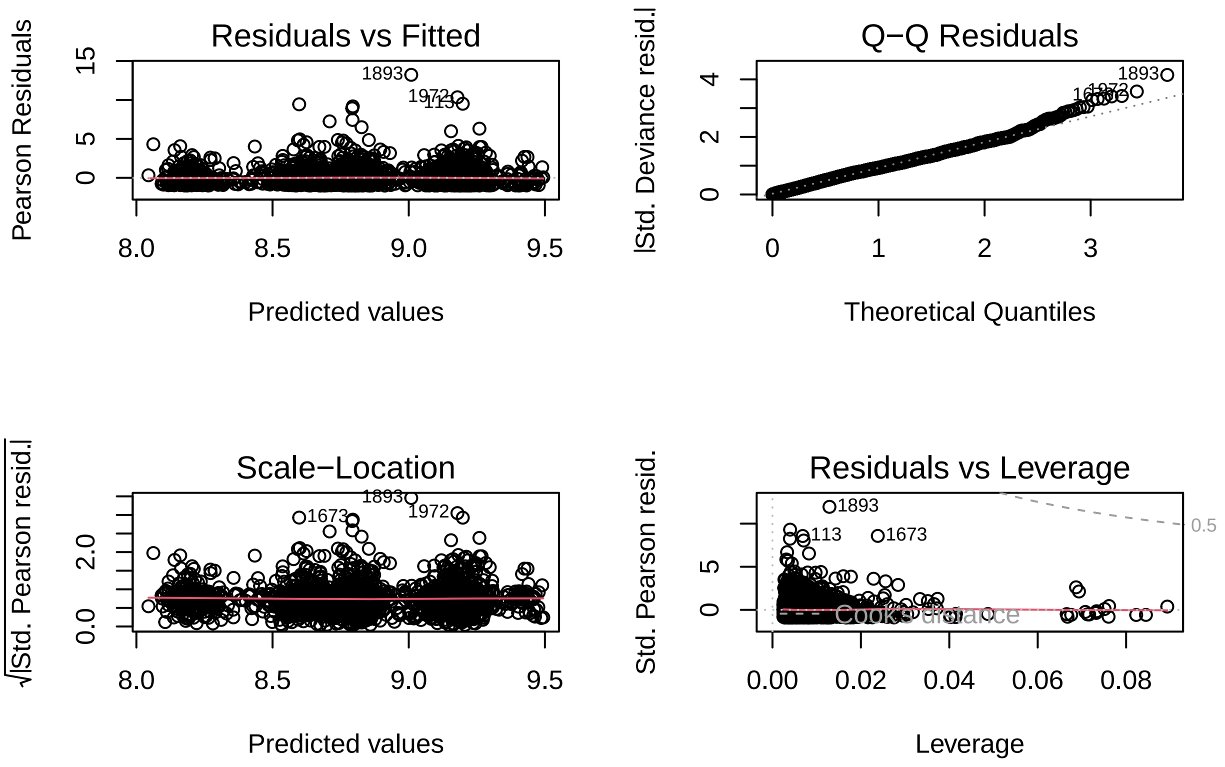
```
## Warning: glm.fit: algorithm did not converge
```

```r
par(mfrow = c(2,2))
print('---------------------FULL MODEL NO INTERACTION TERMS------------------------')
```

```
## [1] "---------------------FULL MODEL NO INTERACTION TERMS------------------------"
```

```r
summary(claimsize.0.gamma);plot(claimsize.0.gamma)
```

```
##
## Call:
## glm(formula = CLAIMS ~ ., family = Gamma(link = "log"), data = claimsize_df)
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                8.555e+00  1.944e-01  44.012  < 2e-16 ***
## ID                        -2.713e-08  7.568e-08  -0.358 0.720044
## AGE26-39                  -7.097e-02  6.641e-02  -1.069 0.285391
## AGE40-64                  -5.188e-02  8.318e-02  -0.624 0.532879
## AGE65+                    -1.822e-01  1.151e-01  -1.583 0.113531
## GENDERmale                 4.028e-01  4.714e-02   8.544  < 2e-16 ***
## DRIVING_EXPERIENCE10-19y -5.776e-01  8.141e-02  -7.095 1.68e-12 ***
## DRIVING_EXPERIENCE20-29y -5.457e-01  1.649e-01  -3.310 0.000946 ***
## DRIVING_EXPERIENCE30y+   -1.892e-01  3.172e-01  -0.597 0.550841
## EDUCATIONnone             -7.026e-04  5.466e-02  -0.013 0.989745
## EDUCATIONuniversity       -1.457e-02  5.700e-02  -0.256 0.798212
## CREDIT_SCORE               3.253e-01  2.095e-01   1.553 0.120622
## VEHICLE_OWNERSHIP         -2.409e-03  4.759e-02  -0.051 0.959626
## VEHICLE_YEARbefore 2015    8.060e-02  7.578e-02   1.064 0.287580
## MARRIED                   -2.411e-02  5.644e-02  -0.427 0.669320
## CHILDREN                   6.204e-03  5.192e-02   0.119 0.904892
## ANNUAL_MILEAGE             5.129e-06  1.005e-05   0.510 0.610004
## VEHICLE_TYPEsports car     2.353e-01  1.027e-01   2.291 0.022035 *
## SPEEDING_VIOLATIONS        2.758e-02  2.402e-02   1.148 0.250901
## PAST_ACCIDENTS            -2.947e-03  3.535e-02  -0.083 0.933551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.241379)
##
##     Null deviance: 2963.2  on 2533  degrees of freedom
## Residual deviance: 2724.3  on 2514  degrees of freedom
## AIC: 50079
##
## Number of Fisher Scoring iterations: 6
```



```r
print('----------------WHITE TEST-----------------------------')
```

```
## [1] "----------------WHITE TEST-----------------------------"
```

```r
white_test(claimsize.0.gamma)
```

```
## White's test results
##
## Null hypothesis: Homoskedasticity of the residuals
## Alternative hypothesis: Heteroskedasticity of the residuals
## Test Statistic: 1
## P-value: 0.605183
```

```r
print('-----------------VIF----------------------------')
```

```
## [1] "-----------------VIF----------------------------"
```

```r
vif(claimsize.0.gamma, type = "predictor")
```

```
## Warning in vif.lm(claimsize.0.gamma, type = "predictor"): type = 'predictor' is available only for ur
##   type = 'terms' will be used

##                         GVIF Df GVIF^(1/(2*Df))
## ID                  1.005306  1        1.002650
## AGE                 3.033224  3        1.203143
## GENDER              1.102305  1        1.049907
## DRIVING_EXPERIENCE  4.569123  3        1.288167
## EDUCATION           1.284516  2        1.064596
## CREDIT_SCORE        1.491735  1        1.221366
## VEHICLE_OWNERSHIP   1.135285  1        1.065498
## VEHICLE_YEAR        1.068256  1        1.033565
## MARRIED             1.365148  1        1.168396
## CHILDREN            1.372683  1        1.171616
## ANNUAL_MILEAGE      1.594910  1        1.262897
## VEHICLE_TYPE        1.009676  1        1.004826
## SPEEDING_VIOLATIONS 2.000328  1        1.414329
## PAST_ACCIDENTS      1.657310  1        1.287366
```

Variable selection

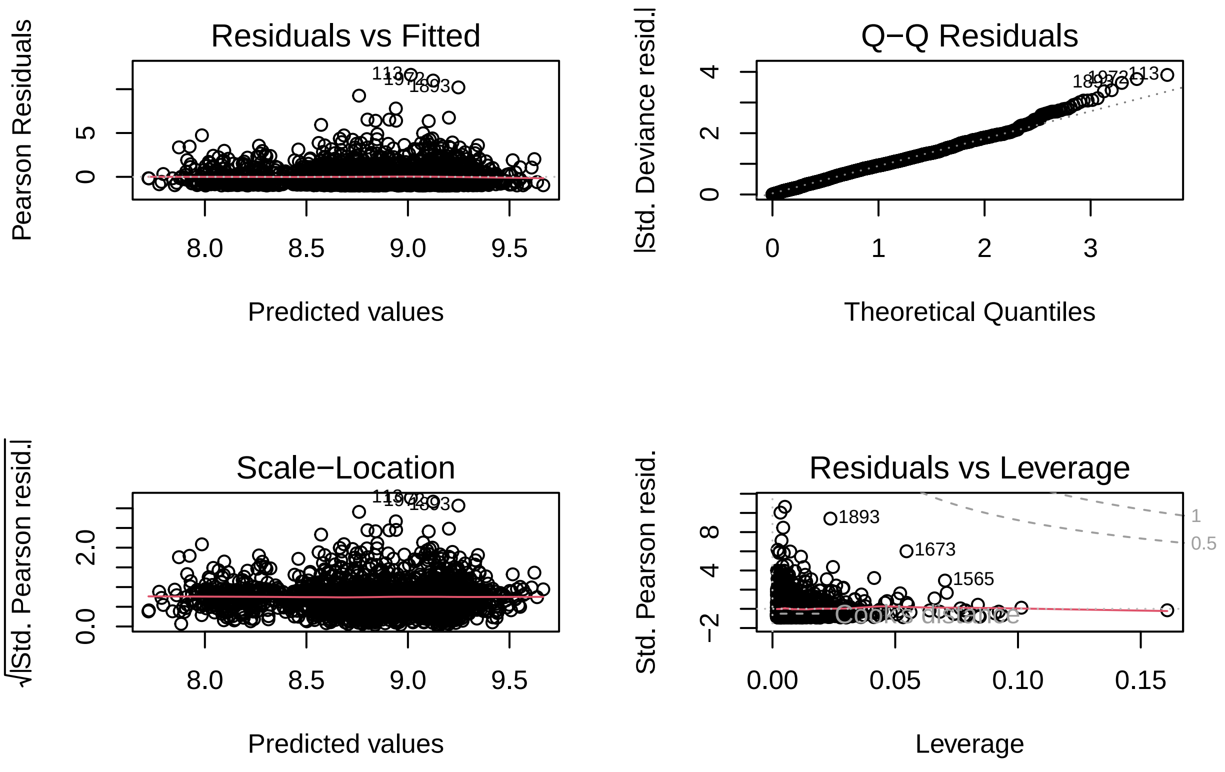
```
df <- claimsize_df


# Ensure positivity
stopifnot(all(df$CLAIMS > 0), all(is.finite(df$CLAIMS)))

# Make sure all categorical vars are factors, then drop unused levels
char_cols <- sapply(df, is.character)
df[char_cols] <- lapply(df[char_cols], factor)
df <- droplevels(df)

upper_form <- CLAIMS ~ (AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION +
                        CREDIT_SCORE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
                        MARRIED + CHILDREN + ANNUAL_MILEAGE +
                        VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2

# Start with a converged base model (intercept + main effects is a good start)
base_form <- CLAIMS ~ AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION +
                      CREDIT_SCORE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
                      MARRIED + CHILDREN + ANNUAL_MILEAGE +
                      VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS

gamma_base <- glm(base_form, family = Gamma(link = "log"),
                  data = df, control = glm.control(maxit = 100))

# Forward or both-direction stepwise within scope; pass control so refits get more iterations
gamma_step <- stats::step(gamma_base,
                   scope = list(lower = ~1, upper = upper_form),
                   direction = "both",
                   trace = 0,
                   control = glm.control(maxit = 70))
summary(gamma_step)
```

```
##
## Call:
## glm(formula = CLAIMS ~ GENDER + DRIVING_EXPERIENCE + CREDIT_SCORE +
##     VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE + VEHICLE_TYPE +
##     SPEEDING_VIOLATIONS + PAST_ACCIDENTS + GENDER:CHILDREN +
##     CHILDREN:ANNUAL_MILEAGE + CREDIT_SCORE:PAST_ACCIDENTS + VEHICLE_YEAR:MARRIED +
##     GENDER:SPEEDING_VIOLATIONS + CREDIT_SCORE:VEHICLE_TYPE, family = Gamma(link = "log"),
##     data = df, control = glm.control(maxit = 100))
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          8.764e+00  2.361e-01  37.126  < 2e-16 ***
## GENDERmale                           2.660e-01  6.497e-02   4.094 4.37e-05 ***
## DRIVING_EXPERIENCE10-19y            -5.953e-01  7.069e-02  -8.422  < 2e-16 ***
## DRIVING_EXPERIENCE20-29y            -5.480e-01  1.549e-01  -3.537 0.000412 ***
## DRIVING_EXPERIENCE30y+              -1.502e-01  3.088e-01  -0.486 0.626840
## CREDIT_SCORE                         4.418e-01  1.950e-01   2.266 0.023558 *
## VEHICLE_YEARbefore 2015              2.311e-01  9.350e-02   2.472 0.013510 *
## MARRIED                              2.449e-01  1.430e-01   1.713 0.086869 .
## CHILDREN                            -7.110e-01  2.434e-01  -2.921 0.003518 **
## ANNUAL_MILEAGE                      -2.069e-05  1.366e-05  -1.515 0.129961
## VEHICLE_TYPEsports car               7.186e-01  3.624e-01   1.983 0.047499 *
## SPEEDING_VIOLATIONS                 -4.699e-02  4.963e-02  -0.947 0.343826
## PAST_ACCIDENTS                       1.749e-01  1.134e-01   1.543 0.123067
## GENDERmale:CHILDREN                  2.082e-01  8.995e-02   2.315 0.020709 *
## CHILDREN:ANNUAL_MILEAGE              4.511e-05  1.809e-05   2.494 0.012702 *
## CREDIT_SCORE:PAST_ACCIDENTS         -4.260e-01  2.275e-01  -1.872 0.061255 .
## VEHICLE_YEARbefore 2015:MARRIED     -3.099e-01  1.493e-01  -2.076 0.037983 *
## GENDERmale:SPEEDING_VIOLATIONS       8.845e-02  4.833e-02   1.830 0.067357 .
## CREDIT_SCORE:VEHICLE_TYPEsports car -1.067e+00  7.690e-01  -1.387 0.165605
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.202638)
##
##     Null deviance: 2963.2  on 2533  degrees of freedom
## Residual deviance: 2695.0  on 2515  degrees of freedom
## AIC: 50045
##
## Number of Fisher Scoring iterations: 6
```

```
cat("\n\n--- Generating Diagnostic Plots for Final Model ---\n")
```

```
##
##
## --- Generating Diagnostic Plots for Final Model ---
```

```
claimsize.2.gamma <- gamma_step

par(mfrow = c(2, 2))
plot(claimsize.2.gamma)
```



```r
exp(coef(claimsize.2.gamma))
```

```
##                          (Intercept)                           GENDERmale
##                         6401.5457825                            1.3047019
##             DRIVING_EXPERIENCE10-19y             DRIVING_EXPERIENCE20-29y
##                            0.5513907                            0.5781134
##               DRIVING_EXPERIENCE30y+                         CREDIT_SCORE
##                            0.8605605                            1.5555631
##              VEHICLE_YEARbefore 2015                              MARRIED
##                            1.2599878                            1.2775392
##                             CHILDREN                       ANNUAL_MILEAGE
##                            0.4911376                            0.9999793
##               VEHICLE_TYPEsports car                   SPEEDING_VIOLATIONS
##                            2.0516241                            0.9540989
##                       PAST_ACCIDENTS                  GENDERmale:CHILDREN
##                            1.1911398                            1.2314638
##              CHILDREN:ANNUAL_MILEAGE          CREDIT_SCORE:PAST_ACCIDENTS
##                            1.0000451                            0.6531375
##      VEHICLE_YEARbefore 2015:MARRIED       GENDERmale:SPEEDING_VIOLATIONS
##                            0.7335132                            1.0924750
## CREDIT_SCORE:VEHICLE_TYPEsports car
##                            0.3442101
```

```r
print("--------------------------------")
```

```
## [1] "--------------------------------"
```

```r
vif(claimsize.2.gamma)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##                                   GVIF Df GVIF^(1/(2*Df))
## GENDER                        2.160919  1        1.470006
## DRIVING_EXPERIENCE            2.832738  3        1.189509
## CREDIT_SCORE                  1.333893  1        1.154943
## VEHICLE_YEAR                  1.678581  1        1.295601
## MARRIED                       9.047066  1        3.007834
## CHILDREN                     31.138872  1        5.580221
## ANNUAL_MILEAGE                3.038172  1        1.743035
## VEHICLE_TYPE                 12.979648  1        3.602728
## SPEEDING_VIOLATIONS           8.816592  1        2.969275
## PAST_ACCIDENTS               17.606395  1        4.195998
## GENDER:CHILDREN               3.670345  1        1.915814
## CHILDREN:ANNUAL_MILEAGE      24.160274  1        4.915310
## CREDIT_SCORE:PAST_ACCIDENTS 17.465245  1        4.179144
## VEHICLE_YEAR:MARRIED          9.025613  1        3.004266
## GENDER:SPEEDING_VIOLATIONS    7.373019  1        2.715330
## CREDIT_SCORE:VEHICLE_TYPE    13.055910  1        3.613296
```

- high vif because we have interaction terms now
- we opt to keep CHILDREN despite the high vif due to the pirnciple of hierarchy, to ensure model interpretation as otherwise interpretation becomes difficult

```r
# between full model and stepwise selected
anova(claimsize.2.gamma, claimsize.1.gamma, test = "LRT")
```

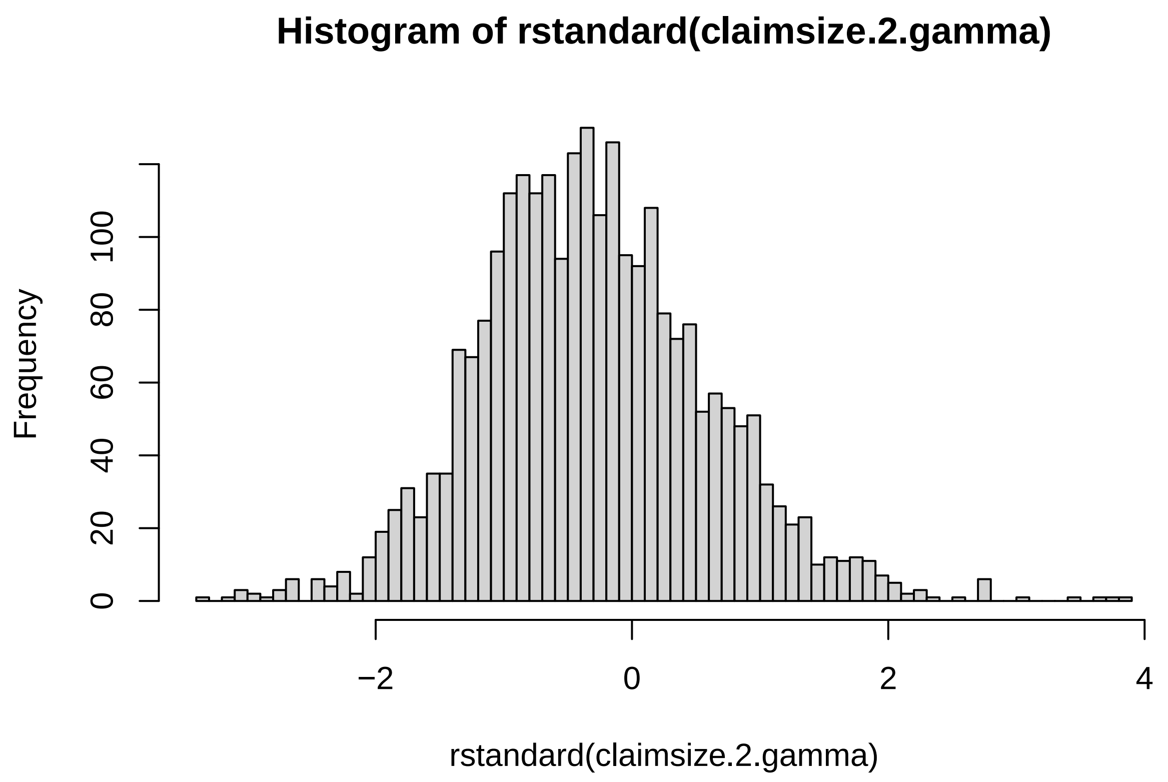
likelihood ratio tests

```
## Analysis of Deviance Table
##
## Model 1: CLAIMS ~ GENDER + DRIVING_EXPERIENCE + CREDIT_SCORE + VEHICLE_YEAR +
##     MARRIED + CHILDREN + ANNUAL_MILEAGE + VEHICLE_TYPE + SPEEDING_VIOLATIONS +
##     PAST_ACCIDENTS + GENDER:CHILDREN + CHILDREN:ANNUAL_MILEAGE +
##     CREDIT_SCORE:PAST_ACCIDENTS + VEHICLE_YEAR:MARRIED + GENDER:SPEEDING_VIOLATIONS +
##     CREDIT_SCORE:VEHICLE_TYPE
## Model 2: CLAIMS ~ (AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION + CREDIT_SCORE +
##     VEHICLE_OWNERSHIP + VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE +
##     VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      2515     2695.0
## 2      2379     2560.1 136    134.9   0.7544
```

- removal of the interaction terms doesnt seem significant under H0, meaning the stepwise selected model is preferred.

Residual analysis

```r
# histogram of type 1 standardised residuals
hist(rstandard(claimsize.2.gamma), breaks = 100)
```



- approximately centered around mean of zero, but still some skewness

- signs of heteroskedasticity as residuals are not uniform over fitted values

identifying influential points (outliers, high leverage)

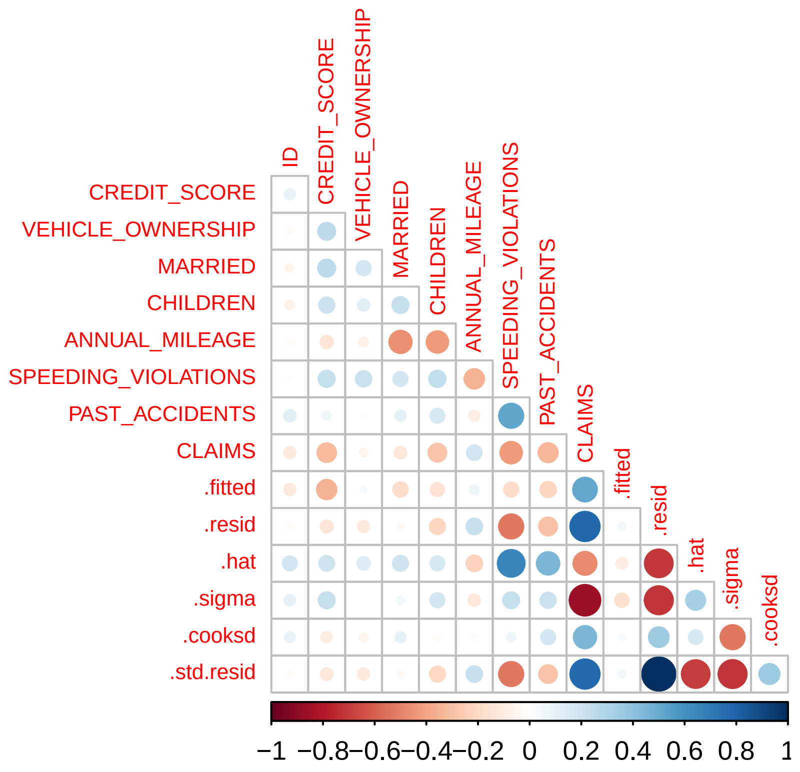
```r
# The augment() function from the broom package creates a tidy dataframe
# containing all the important diagnostic values for each observation.
model_diagnostics <- augment(claimsize.2.gamma,
                             data = claimsize_df)


# --- 3. Identify Influential Points ---
# The most common metric for influence is Cook's Distance, which combines
# leverage and outlier status. A common rule of thumb is to investigate
# points where Cook's distance is greater than 4/n.

# Calculate the Cook's distance threshold
n <- nrow(claimsize_df)
cooks_threshold <- 4 / n
print(paste('cooks threshold is ', cooks_threshold))
```

```
## [1] "cooks threshold is  0.0015785319652723"
```

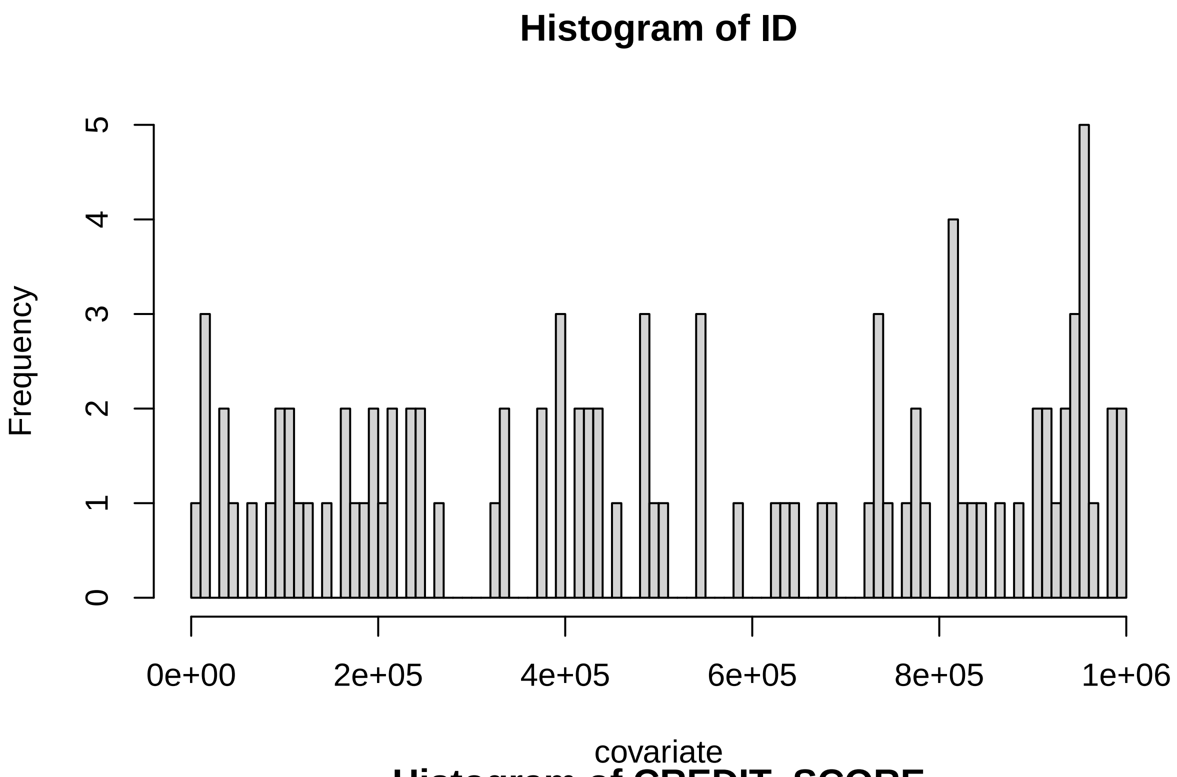
```
# Find the observations that exceed this threshold
influential_points <- model_diagnostics %>%
  filter(.cooksd > cooks_threshold) %>%
  arrange(desc(.cooksd))
print(influential_points)
```

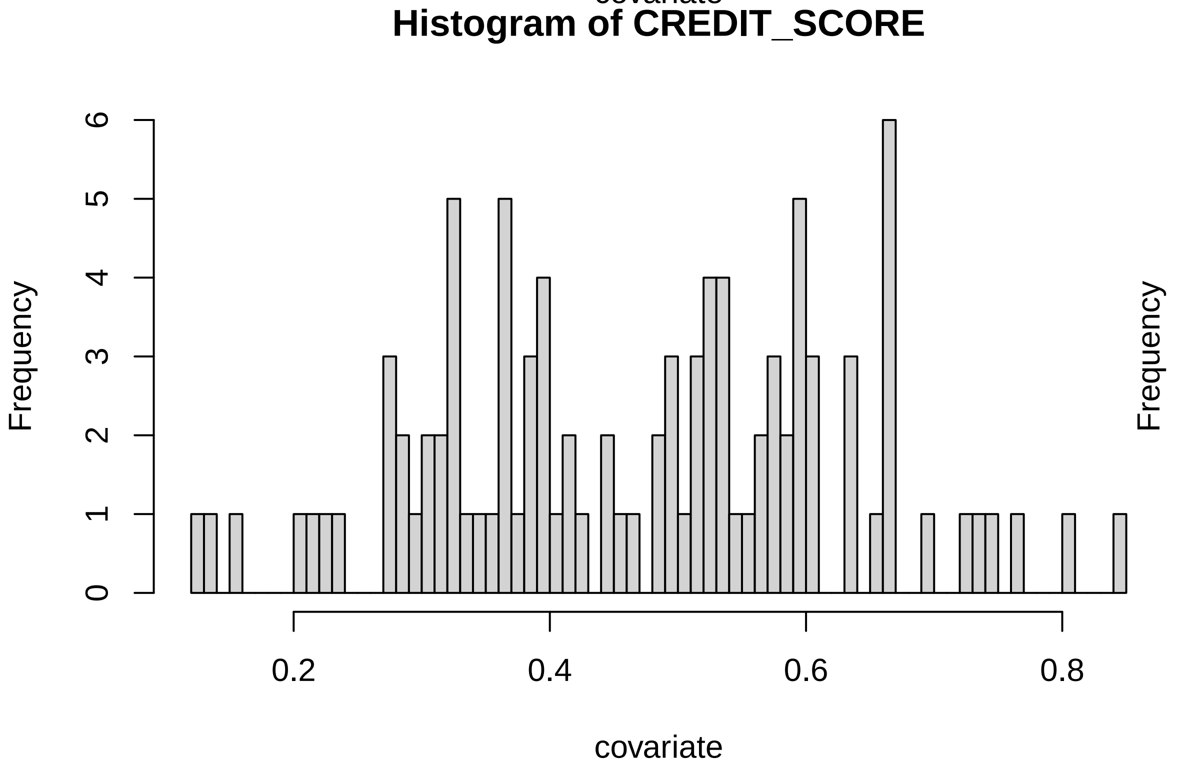
```
## # A tibble: 97 x 21
##        ID AGE   GENDER DRIVING_EXPERIENCE EDUCATION   CREDIT_SCORE
##     <dbl> <fct> <fct>  <fct>              <fct>              <dbl>
##  1 393748 16-25 female 0-9y               university         0.282
##  2 955572 26-39 male   10-19y             high school        0.290
##  3 943240 65+   female 30y+               university         0.596
##  4 167289 16-25 male   0-9y               none               0.337
##  5 991678 40-64 male   10-19y             high school        0.536
##  6  36193 26-39 female 10-19y             high school        0.604
##  7 959649 26-39 female 0-9y               high school        0.633
##  8 210243 16-25 male   0-9y               none               0.274
##  9 937825 40-64 female 0-9y               university         0.669
## 10  98177 26-39 male   10-19y             high school        0.459
## # i 87 more rows
## # i 15 more variables: VEHICLE_OWNERSHIP <dbl>, VEHICLE_YEAR <fct>,
## #   MARRIED <dbl>, CHILDREN <dbl>, ANNUAL_MILEAGE <dbl>, VEHICLE_TYPE <fct>,
## #   SPEEDING_VIOLATIONS <dbl>, PAST_ACCIDENTS <dbl>, CLAIMS <dbl>,
## #   .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,
## #   .std.resid <dbl>
```

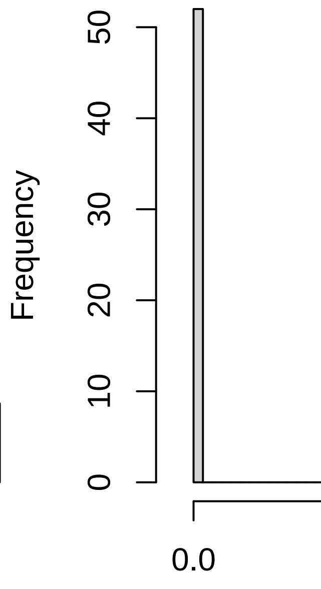
```
# investigating influential points further
# correlation between variables when influential
num_cols <- influential_points %>%
  dplyr::select(where(is.numeric))
corrplot(cor(num_cols), type = "lower", diag = F,tl.cex = 0.7)
```

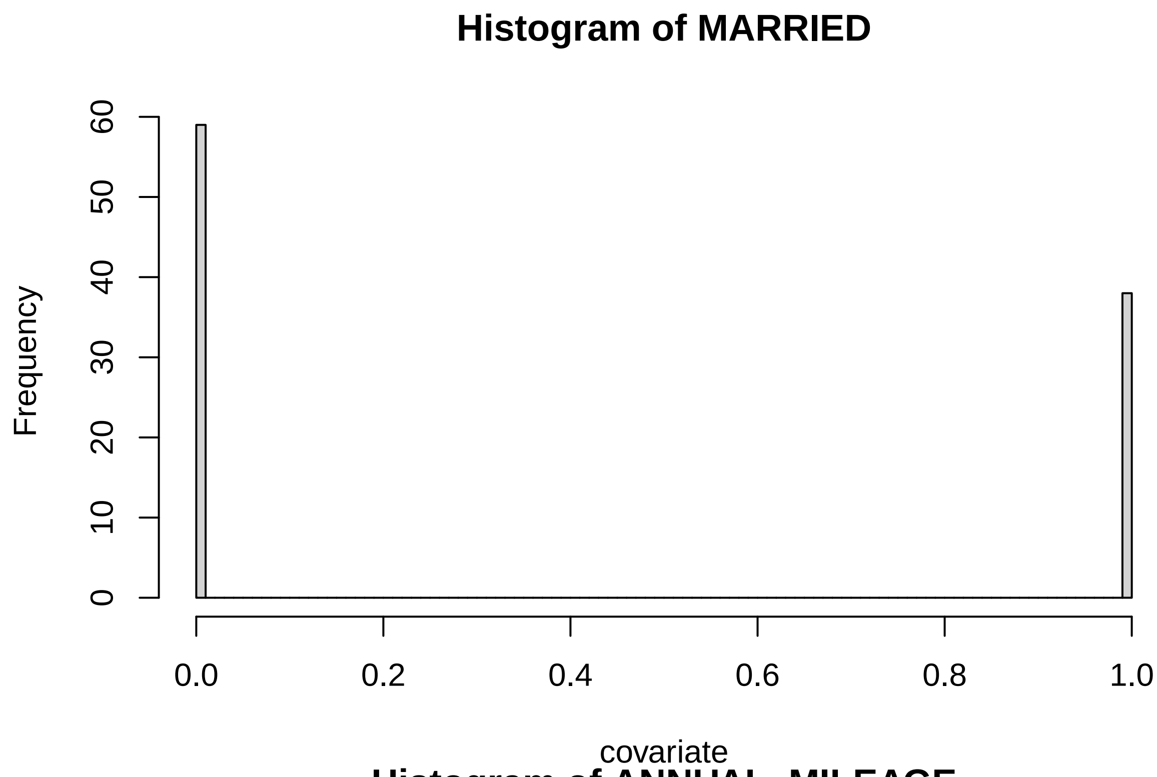


```r
#histogram
for (i in 1:length(influential_points)){
  covariate <- influential_points[[i]]
  if (is.numeric(covariate)){
    hist(covariate, main = paste("Histogram of", names(influential_points)[i] ), xlab = deparse(substitu
  }
}
```

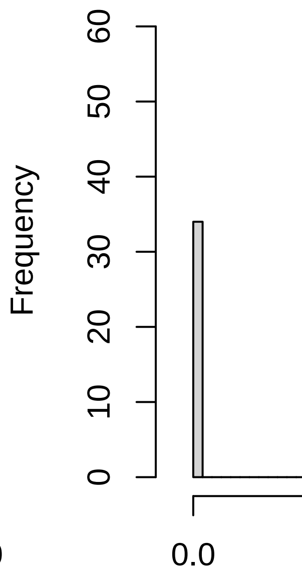

Histogram of ID
Frequency
covariate
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
0
1
2
3
4
5


Histogram of CREDIT_SCORE
Frequency
covariate
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6

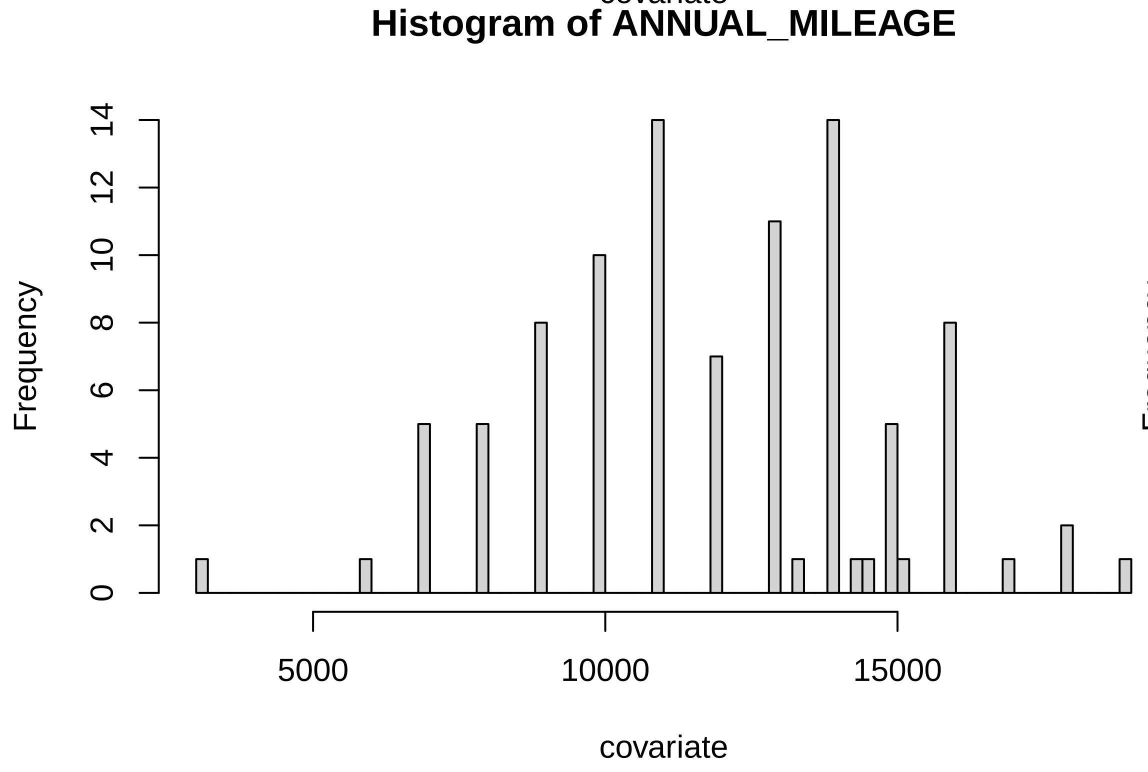

Frequency
0.0
0
10
20
30
40
50



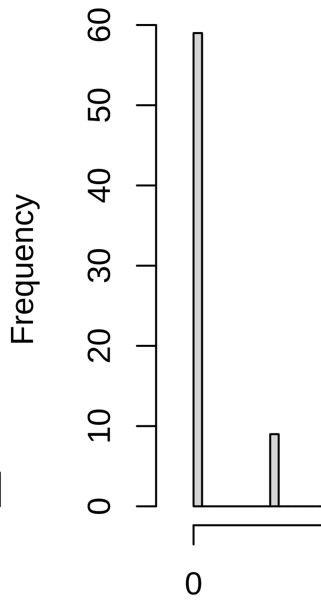
Histogram of MARRIED
Frequency
0
10
20
30
40
50
60
0.0
0.2
0.4
0.6
0.8
1.0
covariate



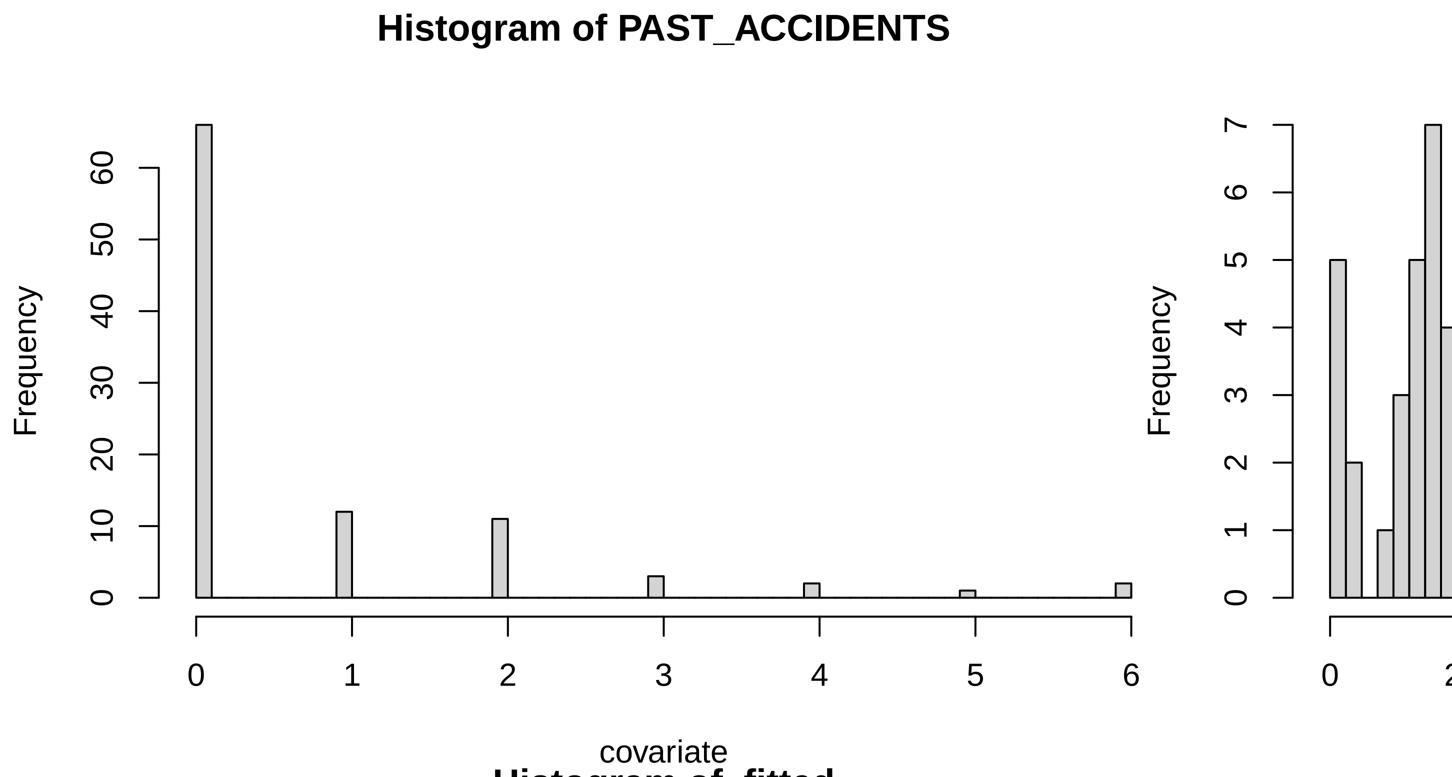
Frequency
0
10
20
30
40
50
60
0.0

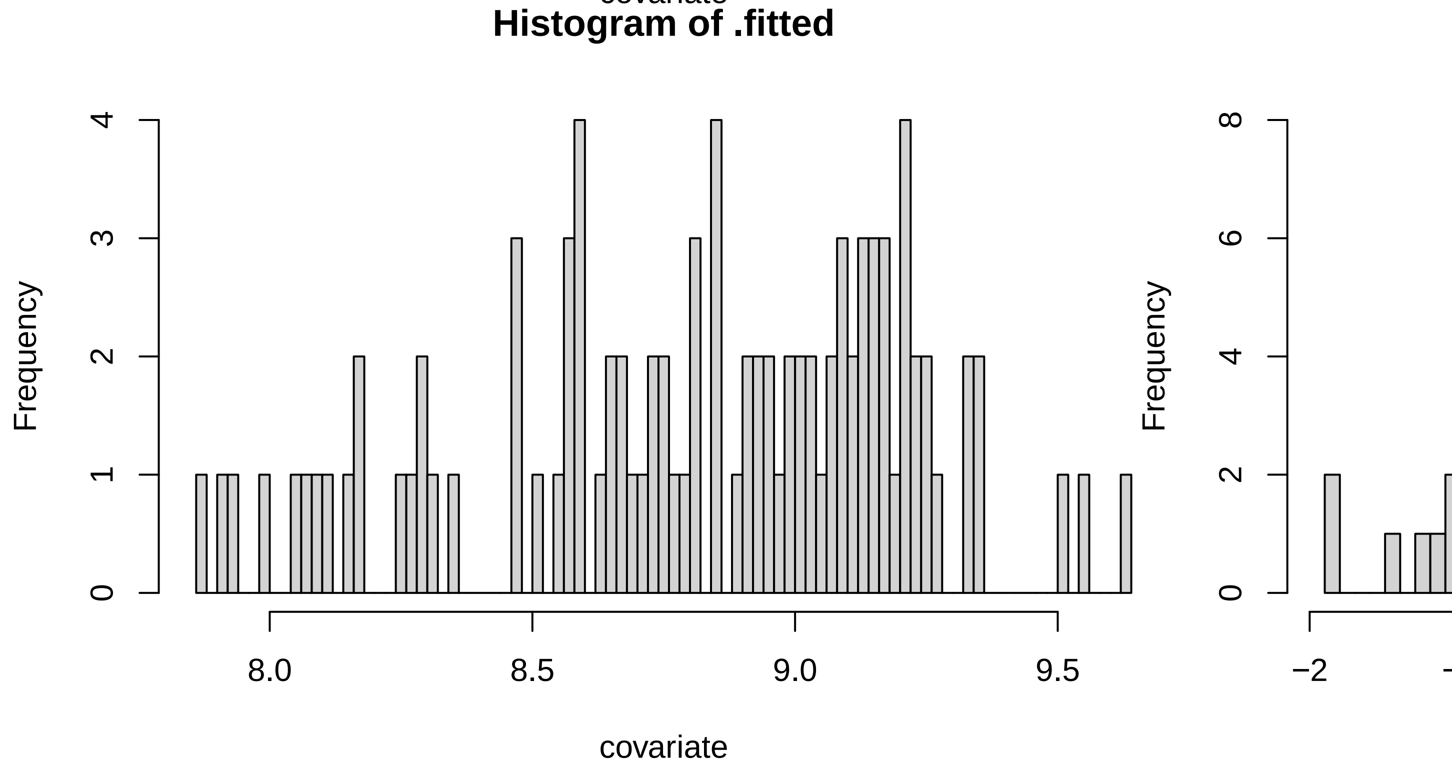


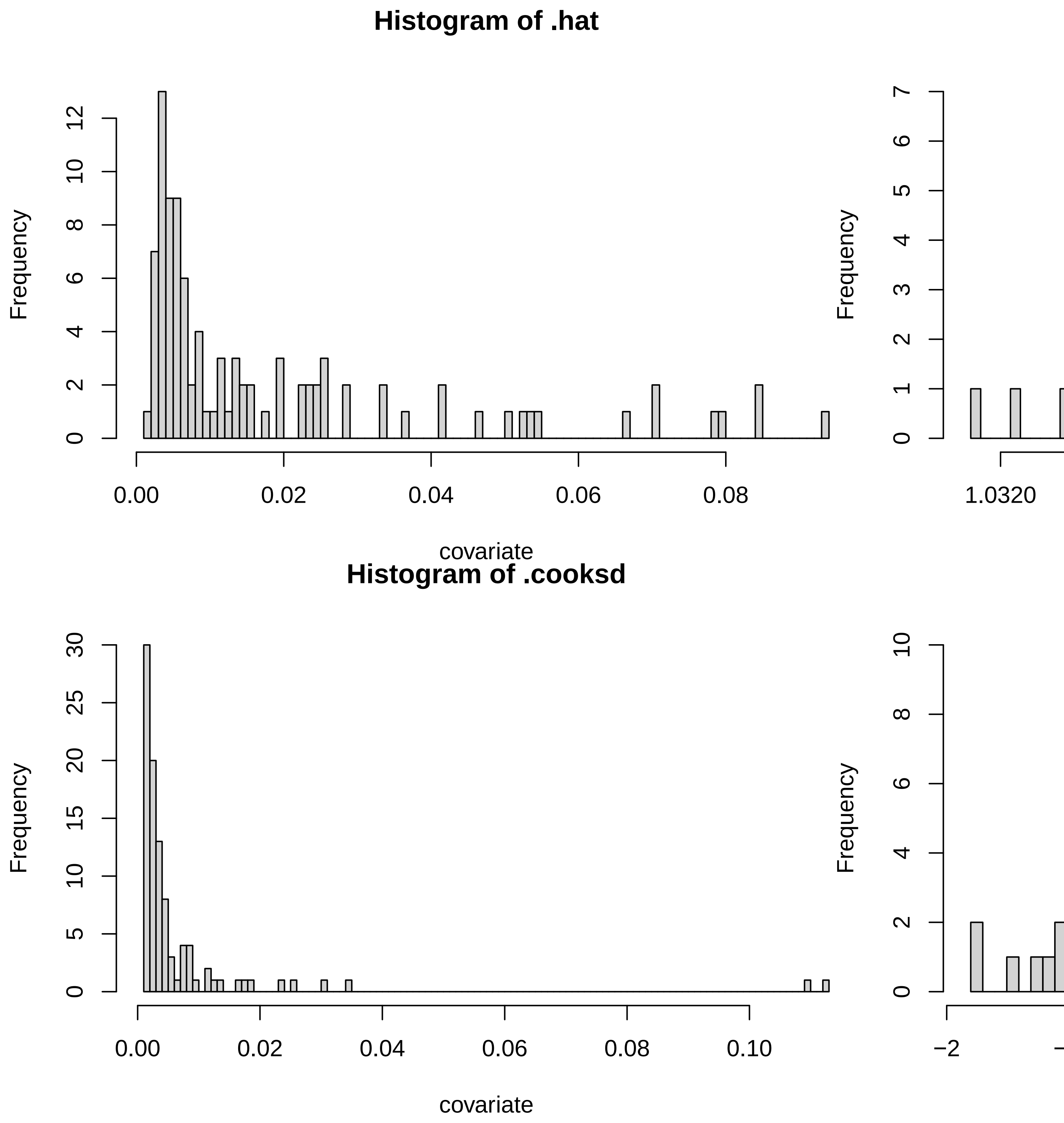
Histogram of ANNUAL_MILEAGE
Frequency
0
2
4
6
8
10
12
14
5000
10000
15000
covariate



Frequency
0
10
20
30
40
50
60
0


Histogram of PAST_ACCIDENTS
Frequency
covariate
0
1
2
3
4
5
6
0
10
20
30
40
50
60
0
1
2
3
4
5
6
7


Histogram of .fitted
Frequency
covariate
8.0
8.5
9.0
9.5
0
1
2
3
4
0
2
4
6
8
−2

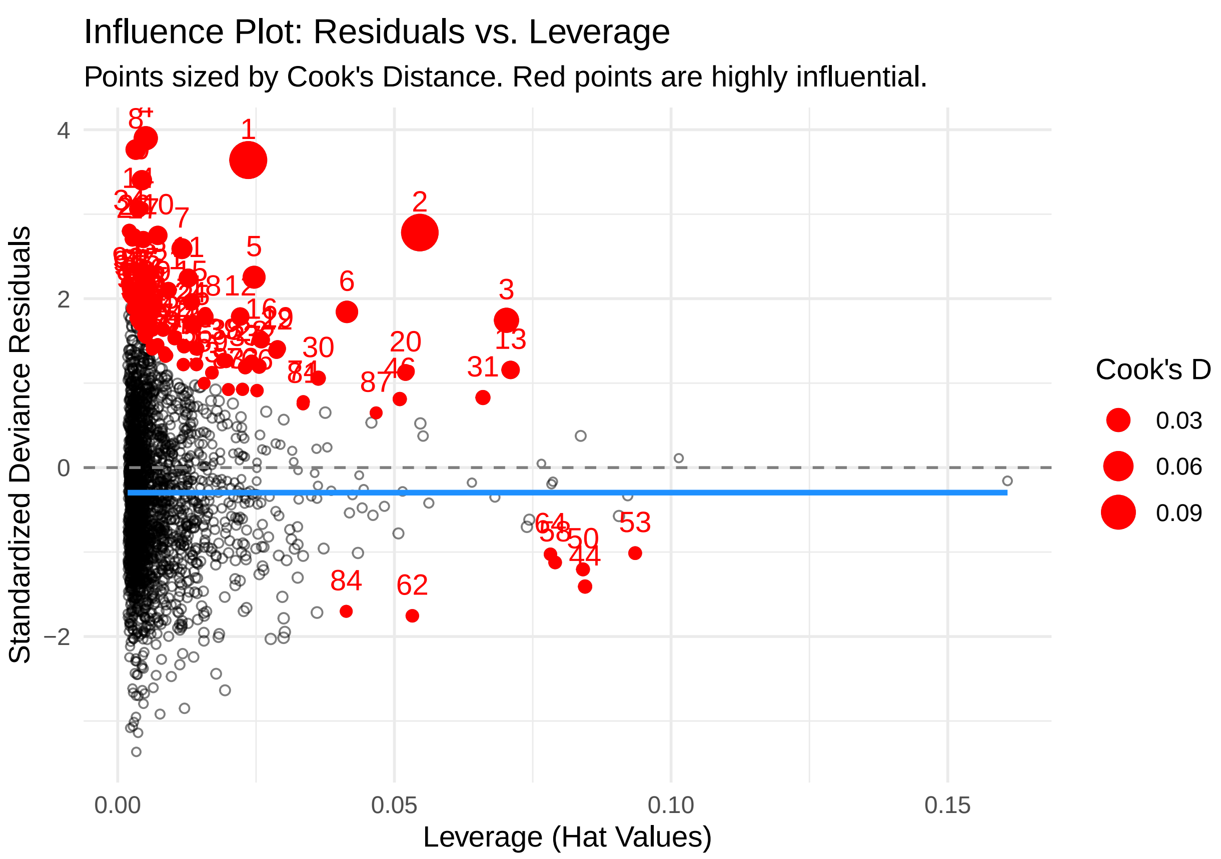


- There seems to be one case with an extremely high claim size (approx 6x the next biggest), which is likely to be a major influential point

```
# plotting influential points
ggplot(model_diagnostics, aes(x = .hat, y = .std.resid)) +
```

```r
  # Points are sized by their Cook's distance
  geom_point(aes(size = .cooksd), alpha = 0.5, shape = 1) +

  # Add a smoother to see the general trend
  geom_smooth(se = FALSE, col = "dodgerblue") +

  # Highlight the most influential points found earlier
  geom_point(data = influential_points, aes(size = .cooksd), color = "red") +

  # Add labels to the influential points (e.g., by row number)
  geom_text(data = influential_points, aes(label = rownames(influential_points)),
            vjust = -1, color = "red") +

  # Add a horizontal line at 0
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50") +

  # Add labels and a title
  labs(
    title = "Influence Plot: Residuals vs. Leverage",
    subtitle = "Points sized by Cook's Distance. Red points are highly influential.",
    x = "Leverage (Hat Values)",
    y = "Standardized Deviance Residuals",
    size = "Cook's D"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



opted for cooks distance as it combines the outlier and high leverage status of points into one metric - 96 influential points

```r
# --- Code to Quickly Assess Impact ---

# Get the row numbers of your influential points
influential_rows <- as.numeric(rownames(influential_points))

# Create a dataframe WITHOUT these points
claimsize_df_no_influencers <- claimsize_df[-influential_rows, ]

# Re-fit your final model on this new data
model_without_influencers <- glm(formula(claimsize.2.gamma),
                                 family = Gamma(link = "log"),
                                 data = claimsize_df_no_influencers)

# Compare the coefficients of the two models
# This will show you exactly how much the influential points were "pulling" the results.
comparison <- cbind(Original = coef(claimsize.2.gamma),
                    No_Influencers = coef(model_without_influencers))

print(comparison)
```

investigating how influential points affect our model

```
##                                        Original No_Influencers
## (Intercept)                        8.764295e+00   8.733942e+00
## GENDERmale                         2.659746e-01   2.489130e-01
## DRIVING_EXPERIENCE10-19y          -5.953116e-01  -5.853716e-01
## DRIVING_EXPERIENCE20-29y          -5.479853e-01  -5.591693e-01
## DRIVING_EXPERIENCE30y+            -1.501714e-01  -2.972910e-01
## CREDIT_SCORE                       4.418376e-01   4.076653e-01
## VEHICLE_YEARbefore 2015            2.311020e-01   2.237383e-01
## MARRIED                            2.449357e-01   2.563357e-01
## CHILDREN                          -7.110310e-01  -6.863137e-01
## ANNUAL_MILEAGE                    -2.068753e-05  -1.630819e-05
## VEHICLE_TYPEsports car             7.186317e-01   7.650618e-01
## SPEEDING_VIOLATIONS               -4.698791e-02  -4.791772e-02
## PAST_ACCIDENTS                     1.749107e-01   1.675351e-01
## GENDERmale:CHILDREN                2.082036e-01   1.905174e-01
## CHILDREN:ANNUAL_MILEAGE            4.510647e-05   4.385865e-05
## CREDIT_SCORE:PAST_ACCIDENTS       -4.259676e-01  -3.929006e-01
## VEHICLE_YEARbefore 2015:MARRIED   -3.099096e-01  -3.109545e-01
## GENDERmale:SPEEDING_VIOLATIONS     8.844577e-02   9.291959e-02
## CREDIT_SCORE:VEHICLE_TYPEsports car -1.066503e+00  -1.187627e+00
```

```
c(Original_Dispersion = summary(claimsize.2.gamma)$dispersion, No_Influencers_Dispersion = summary(mode
```

```
##       Original_Dispersion No_Influencers_Dispersion
##                  1.202638                  1.209761
```

- not much difference in dispersion
- influential points arent doing too much to significantly affect the model fit

Testing the Gamma and Lognormal model

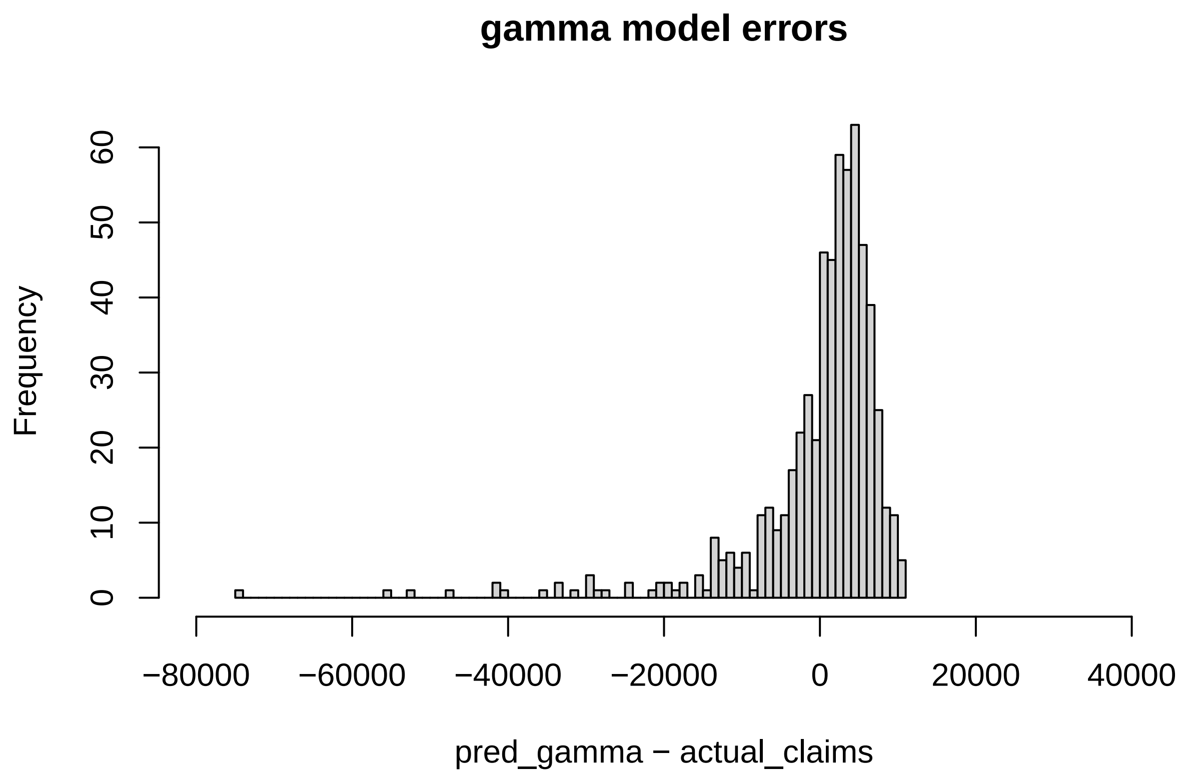
```
# Helper: coerce factor levels in newdata to training levels stored in the fit
prep_newdata_for <- function(newdata, fit) {
  nd <- newdata
  if (!is.null(fit$xlevels)) {
    for (nm in names(fit$xlevels)) {
      if (nm %in% names(nd)) {
        nd[[nm]] <- factor(nd[[nm]], levels = fit$xlevels[[nm]])
      }
    }
  }
  droplevels(nd)
}

# 1) Test set for severity evaluation (only positive claims)
test_claimsize_df <- test_data_imputed |>
  dplyr::filter(CLAIMS > 0)

# If your models used ^2 in the formula (no explicit hand-made columns), you do NOT need to mutate inte
# If you explicitly included columns like ANNUAL_MILEAGE_CHILDREN in the model formula, then recreate t
```

```r
# 2) Align test data to each model
test_for_gamma    <- prep_newdata_for(test_claimsize_df, claimsize.2.gamma)
test_for_lognorm  <- prep_newdata_for(test_claimsize_df, claimsize.2.lognormal)

# 3) Predictions
# Gamma(log): already on the dollar scale
pred_gamma <- predict(claimsize.2.gamma, newdata = test_for_gamma, type = "response")

# Lognormal (lm on log(CLAIMS)): use smearing correction
pred_log <- predict(claimsize.2.lognormal, newdata = test_for_lognorm)  # predicts log(CLAIMS)

# Duan's smearing factor from TRAINING residuals of the log model
sf <- mean(exp(residuals(claimsize.2.lognormal)), na.rm = TRUE)
pred_lognormal <- exp(pred_log) * sf
# (Alternative normal-theory correction: sigma2 <- summary(claimsize.2.lognormal)$sigma^2; pred_lognorm

# 4) Metrics
actual_claims <- test_for_lognorm$CLAIMS

rmse <- function(a, p) sqrt(mean((a - p)^2, na.rm = TRUE))
mae  <- function(a, p) mean(abs(a - p), na.rm = TRUE)

rmse_gamma <- rmse(actual_claims, pred_gamma)
mae_gamma  <- mae(actual_claims, pred_gamma)

rmse_lognormal <- rmse(actual_claims, pred_lognormal)
mae_lognormal  <- mae(actual_claims, pred_lognormal)

comparison_df <- data.frame(
  Model = c("Gamma GLM", "Lognormal (lm + smearing)"),
  RMSE  = c(rmse_gamma, rmse_lognormal),
  MAE   = c(mae_gamma,  mae_lognormal)
)
print(comparison_df)
```

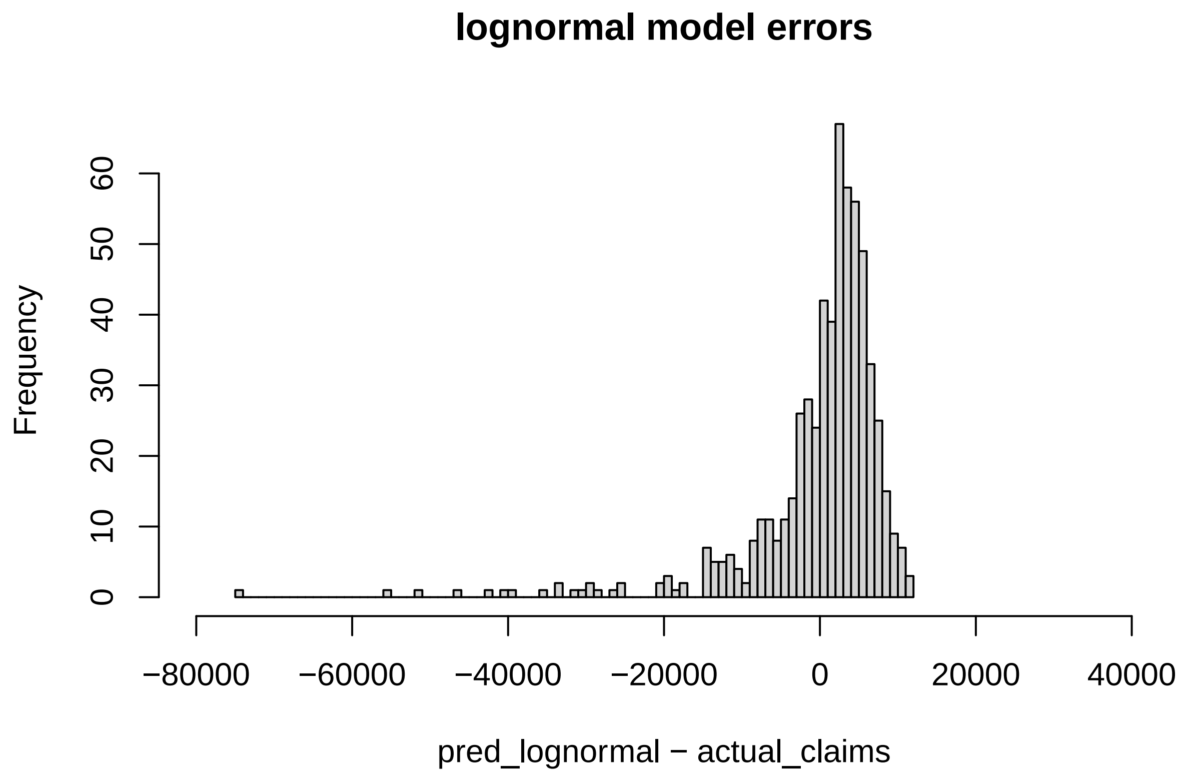
```
##                       Model     RMSE      MAE
## 1                 Gamma GLM 8950.423 5578.078
## 2 Lognormal (lm + smearing) 8970.584 5617.047
```

When we fit a lognormal severity model (lm on log(CLAIMS)), the model predicts the mean of the log outcome. Simply exponentiating those predictions gives the conditional median on the dollar scale and underestimates the mean (Jensen's inequality). To obtain unbiased mean severities, we apply a back-transformation correction: Duan's smearing factor (multiply by the average of exp(residuals)) or, under homoskedastic normal log-errors, multiply by exp(sigma2/2). Gamma GLM predictions (log link) already return the mean and need no correction.

```r
# plot errors

hist(pred_gamma - actual_claims, breaks = 100, xlim = c(-80000, 40000),
     main = 'gamma model errors')
```



```r
hist(pred_lognormal - actual_claims, breaks = 100, xlim = c(-80000, 40000),
     main = 'lognormal model errors')
```



There are some extreme tail cases that our models mis, leading to a skewed error distribution

```r
# computing average errors
print(paste("gamma model", mean(pred_gamma - actual_claims)))
```

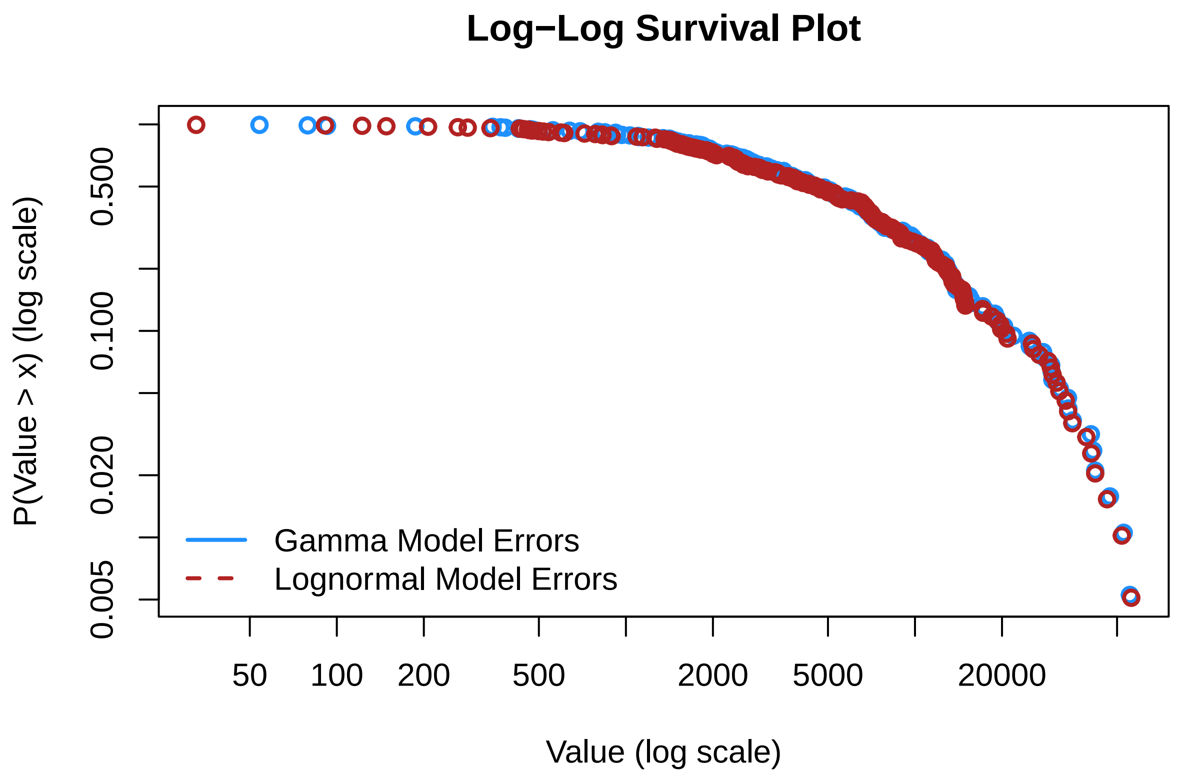
```
## [1] "gamma model -2.47898757513912"
```

```r
print(paste("lognormal model", mean(pred_lognormal - actual_claims)))
```

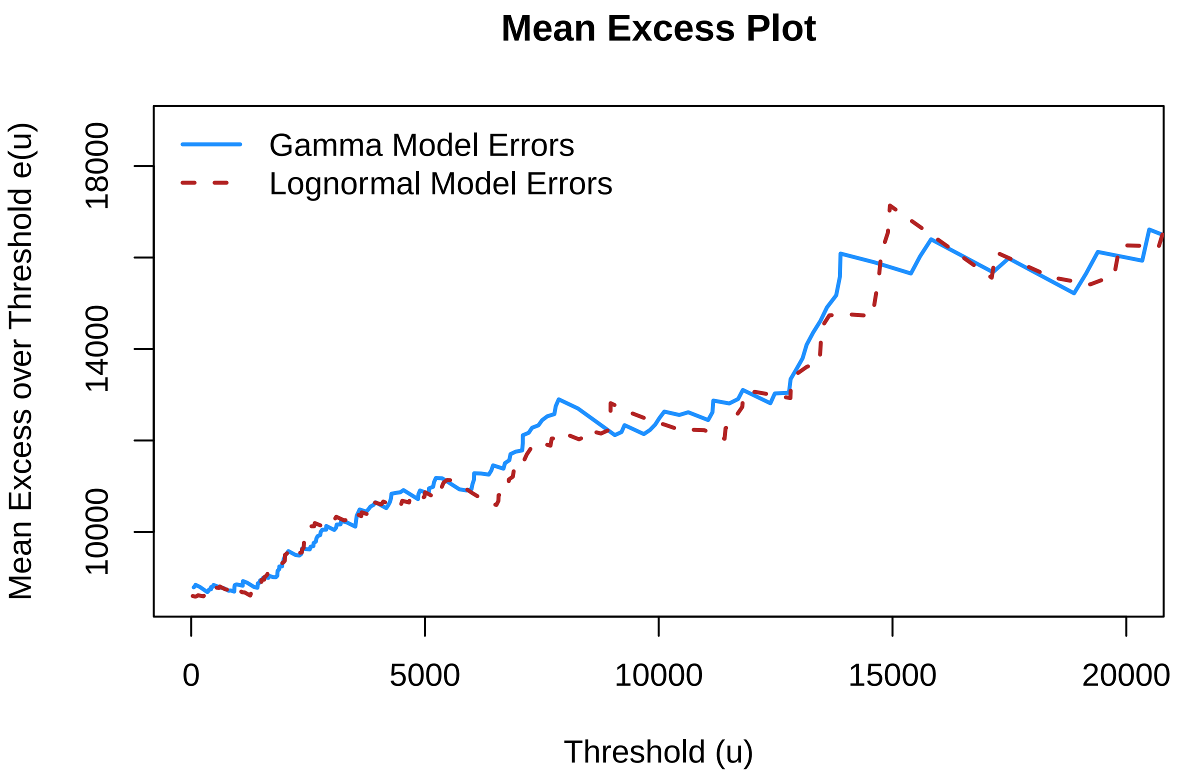
```
## [1] "lognormal model -3.15019208632837"
```

```r
# observing tail behaviours of the errors
gamma_model_errors <- pred_gamma - actual_claims
gamma_model_errors_tail <- abs(gamma_model_errors[gamma_model_errors  < (-10)])
lognormal_model_errors <- pred_lognormal - actual_claims
lognormal_model_errors_tail <- abs(lognormal_model_errors[lognormal_model_errors  < (-10)])
```

```r
# --- 1. Define Your Data and Names ---
# Replace these placeholder vectors with your actual data
vector1 <- gamma_model_errors_tail # e.g., errors_gamma
vector2 <- lognormal_model_errors_tail   # e.g., errors_lognormal

# Define names for the legend
name1 <- "Gamma Model Errors"
name2 <- "Lognormal Model Errors"




# --- Plot 1: Log-Log Survival Plot ---

ecdf_x <- ecdf(vector1)
data_x <- data.frame(val = sort(unique(vector1))) %>% mutate(prob = 1 - ecdf_x(val)) %>% filter(prob > 
ecdf_y <- ecdf(vector2)
data_y <- data.frame(val = sort(unique(vector2))) %>% mutate(prob = 1 - ecdf_y(val)) %>% filter(prob > 
xlim_range <- range(c(data_x$val, data_y$val))
ylim_range <- range(c(data_x$prob, data_y$prob))
plot(data_x$val, data_x$prob, type = "p", log = "xy", col = "dodgerblue", lwd = 2,
     xlim = xlim_range, ylim = ylim_range, main = "Log-Log Survival Plot",
     xlab = "Value (log scale)", ylab = "P(Value > x) (log scale)")
points(data_y$val, data_y$prob, col = "firebrick", lwd = 2, lty = 2)
legend("bottomleft", legend = c(name1, name2), col = c("dodgerblue", "firebrick"), lty = c(1, 2), lwd =
```



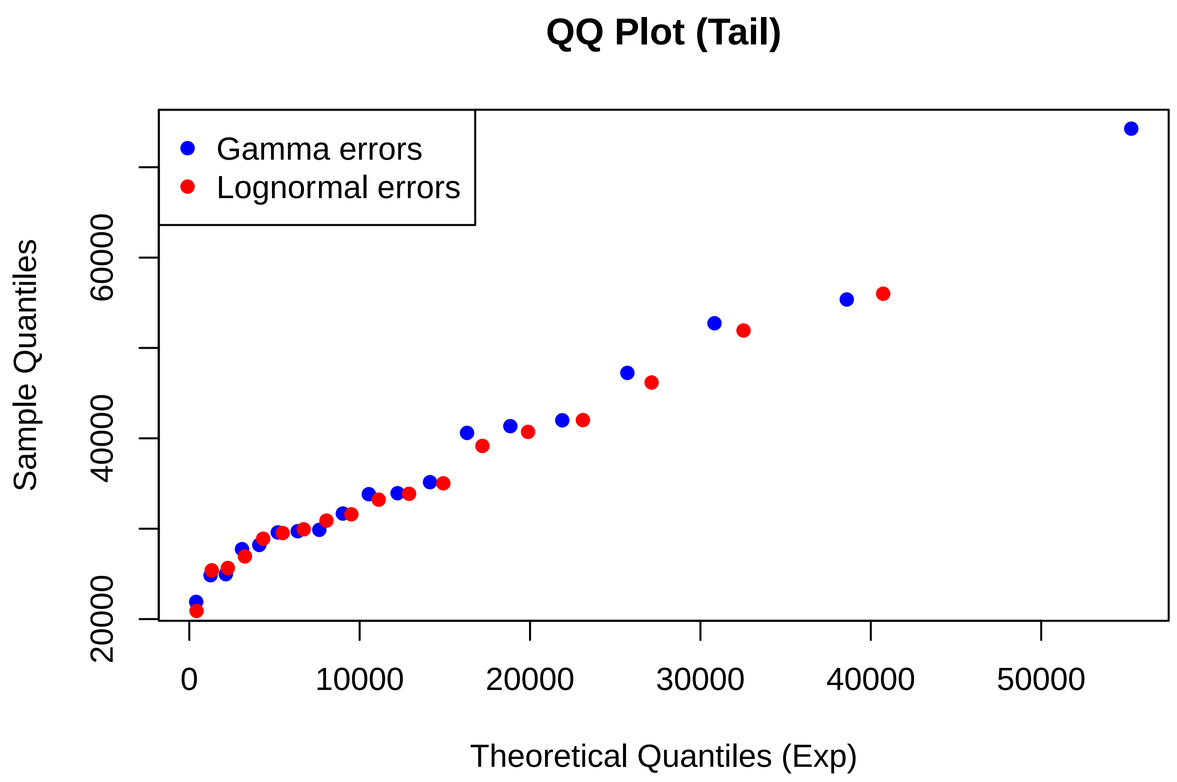
```r
# --- Plot 2: Mean Excess Plot ---

mean_excess_func <- function(data) {
  thresholds <- unique(sort(data))
  excess <- sapply(thresholds, function(u) { mean(data[data > u] - u) })
  return(data.frame(threshold = thresholds, mean_excess = excess))
}
me_x <- mean_excess_func(vector1)
me_y <- mean_excess_func(vector2)
xlim_range_me <- range(c(me_x$threshold, me_y$threshold))
ylim_range_me <- range(c(me_x$mean_excess, me_y$mean_excess), na.rm = TRUE)
plot(me_x$threshold, me_x$mean_excess, type = "l", col = "dodgerblue", lwd = 2,
     xlim = c(0, 20000), ylim = ylim_range_me, main = "Mean Excess Plot",
     xlab = "Threshold (u)", ylab = "Mean Excess over Threshold e(u)")
lines(me_y$threshold, me_y$mean_excess, col = "firebrick", lwd = 2, lty = 2)
legend("topleft", legend = c(name1, name2), col = c("dodgerblue", "firebrick"), lty = c(1, 2), lwd = 2,
```



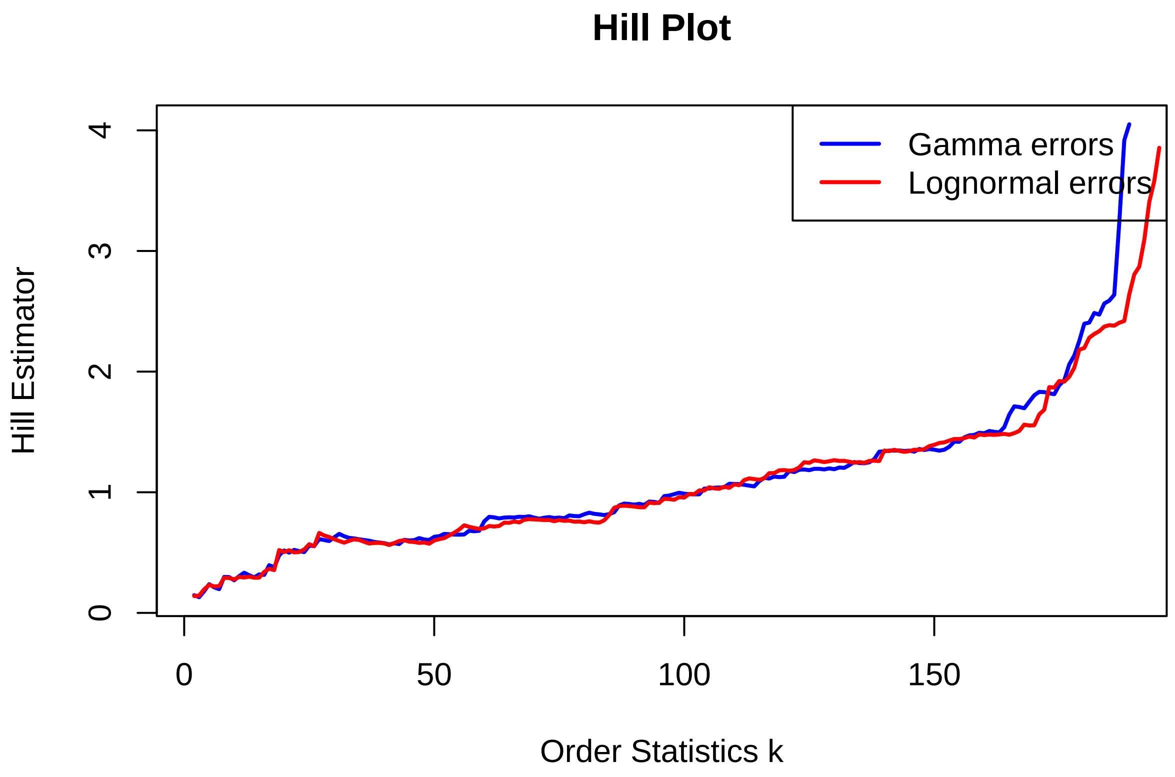
```
# AIC for the Gamma(GLM) model (already on Y-scale)
aic_gamma <- AIC(claimsize.2.gamma)

# AIC for the lognormal model from lm(log(CLAIMS) ~ ...)
# Convert the lm to a lognormal likelihood on Y
mf_ln  <- model.frame(claimsize.2.lognormal)
y_log  <- model.response(mf_ln)                 # = log(CLAIMS)
y      <- exp(y_log)                            # original scale
mu_log <- fitted(claimsize.2.lognormal)         # mean on log scale
sdlog  <- sigma(claimsize.2.lognormal)          # residual SD on log scale

ll_lognorm <- sum(dlnorm(y, meanlog = mu_log, sdlog = sdlog, log = TRUE))
k_lognorm  <- length(coef(claimsize.2.lognormal)) + 1  # +1 for sd
aic_lognorm <- -2 * ll_lognorm + 2 * k_lognorm

# (Equivalent shortcut: aic_lognorm <- AIC(claimsize.2.lognormal) + 2 * sum(log(y)))

# Compare
aics <- c(Gamma = aic_gamma, Lognormal = aic_lognorm)
delta <- aics - min(aics)
akaike_wt <- exp(-0.5 * delta) / sum(exp(-0.5 * delta))

data.frame(Model = names(aics), AIC = aics, DeltaAIC = delta, AkaikeWeight = akaike_wt)
```

```
##                 Model      AIC DeltaAIC AkaikeWeight
## Gamma           Gamma 50045.40   0.0000 1.000000e+00
## Lognormal Lognormal 50253.18 207.7781 7.612986e-46
```

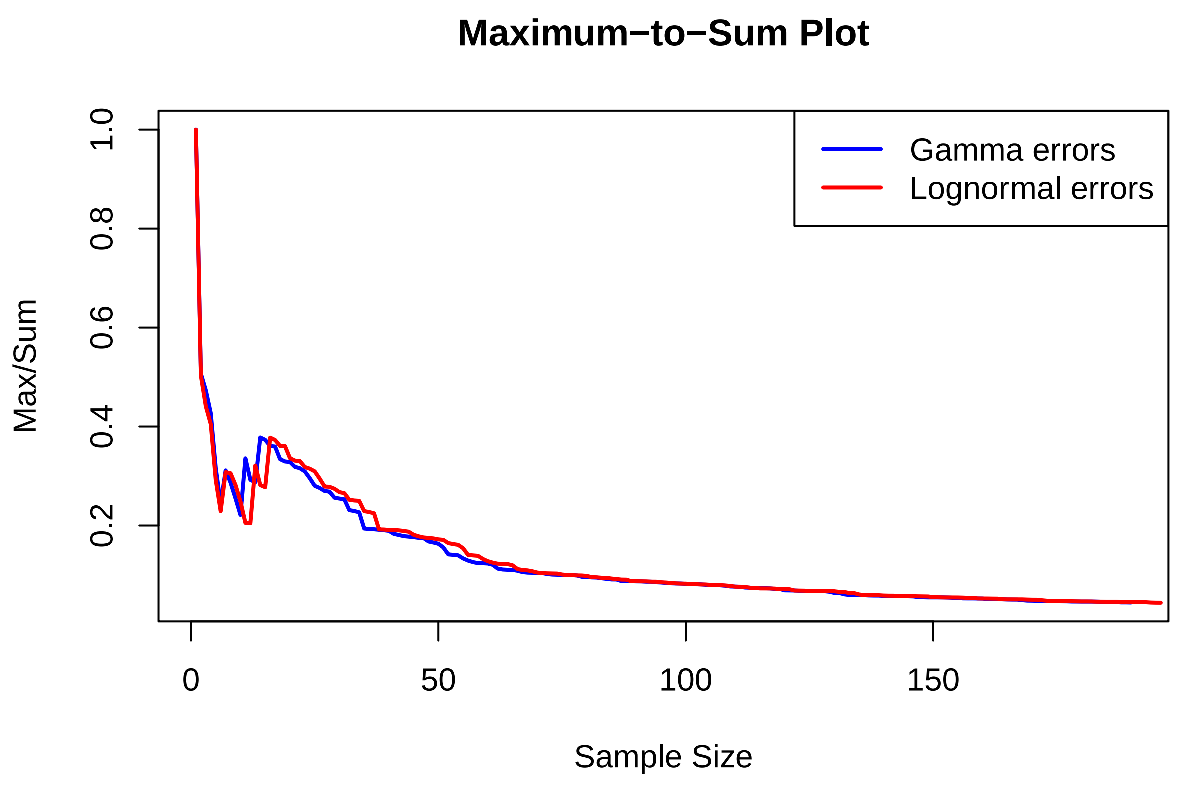
```r
# QQ Plot against Exponential (for tail diagnosis)
qqtail <- function(vec, col, add=FALSE, ...) {
  n <- length(vec)
  sorted <- sort(vec)
  k <- floor(0.1 * n)
  tail <- sorted[(n-k+1):n]
  qexp <- qexp(ppoints(k), rate=1/mean(tail - min(tail)))
  if (!add) {
    plot(qexp, tail, col=col, pch=16, xlab="Theoretical Quantiles (Exp)",
         ylab="Sample Quantiles", main="QQ Plot (Tail)", ...)
  } else {
    points(qexp, tail, col=col, pch=16)
  }
}

qqtail(gamma_model_errors_tail, "blue")
qqtail(lognormal_model_errors_tail, "red", add=TRUE)
legend("topleft", legend=c("Gamma errors", "Lognormal errors"), col=c("blue", "red"), pch=16)
```



```r
# Hill Plot (Pareto tail index estimator)
hillplot <- function(vec, col, add=FALSE, ...) {
  sorted <- sort(vec, decreasing=TRUE)
  n <- length(vec)
  k <- 2:(n-1)
  hill <- sapply(k, function(i) mean(log(sorted[1:i])) - log(sorted[i]))
  if (!add) {
    plot(k, hill, type="l", col=col, lwd=2,
         xlab="Order Statistics k", ylab="Hill Estimator", main="Hill Plot", ...)
  } else {
    lines(k, hill, col=col, lwd=2)
  }
}

hillplot(gamma_model_errors_tail, "blue")
hillplot(lognormal_model_errors_tail, "red", add=TRUE)
legend("topright", legend=c("Gamma errors", "Lognormal errors"), col=c("blue", "red"), lwd=2)
```



```r
# Maximum-to-Sum Plot (finite moments check)
max2sumplot <- function(vec, col, add=FALSE, ...) {
  n <- length(vec)
  k <- 1:n
  maxvals <- sapply(k, function(i) max(vec[1:i]))
  sumvals <- sapply(k, function(i) sum(vec[1:i]))
  ratio <- maxvals/sumvals
  if (!add) {
    plot(k, ratio, type="l", col=col, lwd=2,
         xlab="Sample Size", ylab="Max/Sum", main="Maximum-to-Sum Plot", ...)
  } else {
    lines(k, ratio, col=col, lwd=2)
  }
}

max2sumplot(gamma_model_errors_tail, "blue")
max2sumplot(lognormal_model_errors_tail, "red", add=TRUE)
legend("topright", legend=c("Gamma errors", "Lognormal errors"), col=c("blue", "red"), lwd=2)
```



- Not much difference in the tail, however we choose Gamma model largely due to the difference in AIC.
- Theory says lognormal has a heavier tail than gamma; diagnostics are consistent but the difference is small in this dataset.
- Both models exhibit very similar tail behaviour on residuals and near-identical predictive accuracy.
- the gamma model is a reasonable, slightly better-scoring choice.

Investigation of claim outcome

```
claimoutcome_df <- train_data_imputed %>%select(-CLAIMS)
mean(claimoutcome_df$OUTCOME)
```

```
## [1] 0.31675
```

logistic regression

```
# full basic model without interaction terms
outcome.0.logistic <- glm(OUTCOME ~ .,
                          family = binomial(link = "logit"),
                          data = claimoutcome_df)

# full model with interaction terms
outcome.1.logistic <- glm(OUTCOME ~(AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION +
                             CREDIT_SCORE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
                             MARRIED + CHILDREN + ANNUAL_MILEAGE +
                             VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2,
                          family= binomial(link = "logit"),
                          data = claimoutcome_df)
```

Variable selection

Stepwise selection based on AIC

```
# Stepwise (both) from null -> up to the full formula of outcome.1.logistic
logistic_both <- stats::step(
  update(outcome.1.logistic, . ~ 1),                      # start: intercept-only (same data/family)
  scope = list(lower = ~ 1, upper = formula(outcome.1.logistic)),
  direction = "both",
  trace = 0,
  k = log(nobs(outcome.1.logistic))                        # BIC; use k = 2 for AIC
)

summary(logistic_both)
```

```
##
## Call:
## glm(formula = OUTCOME ~ DRIVING_EXPERIENCE + VEHICLE_OWNERSHIP +
##     VEHICLE_YEAR + GENDER + MARRIED + PAST_ACCIDENTS + ANNUAL_MILEAGE +
##     SPEEDING_VIOLATIONS + DRIVING_EXPERIENCE:VEHICLE_YEAR + VEHICLE_OWNERSHIP:SPEEDING_VIOLATIONS +
##     GENDER:MARRIED, family = binomial(link = "logit"), data = claimoutcome_df)
##
## Coefficients:
##                                                 Estimate Std. Error z value
## (Intercept)                                    -8.220e-01  2.229e-01  -3.687
## DRIVING_EXPERIENCE10-19y                       -1.533e+00  1.808e-01  -8.477
## DRIVING_EXPERIENCE20-29y                       -2.237e+00  2.768e-01  -8.082
## DRIVING_EXPERIENCE30y+                         -2.634e+00  4.387e-01  -6.004
## VEHICLE_OWNERSHIP                              -1.897e+00  8.301e-02 -22.847
## VEHICLE_YEARbefore 2015                         2.093e+00  1.152e-01  18.169
## GENDERmale                                      8.538e-01  8.904e-02   9.588
## MARRIED                                        -6.910e-01  1.087e-01  -6.355
## PAST_ACCIDENTS                                 -2.149e-01  3.877e-02  -5.542
## ANNUAL_MILEAGE                                  6.917e-05  1.440e-05   4.803
## SPEEDING_VIOLATIONS                             4.719e-03  3.758e-02   0.126
## DRIVING_EXPERIENCE10-19y:VEHICLE_YEARbefore 2015 -4.730e-01  1.903e-01  -2.486
## DRIVING_EXPERIENCE20-29y:VEHICLE_YEARbefore 2015 -1.423e+00  2.936e-01  -4.847
## DRIVING_EXPERIENCE30y+:VEHICLE_YEARbefore 2015   -2.037e+00  5.280e-01  -3.859
## VEHICLE_OWNERSHIP:SPEEDING_VIOLATIONS           1.282e-01  3.976e-02   3.224
## GENDERmale:MARRIED                              4.189e-01  1.381e-01   3.032
##                                                Pr(>|z|)
## (Intercept)                                    0.000227 ***
## DRIVING_EXPERIENCE10-19y                        < 2e-16 ***
## DRIVING_EXPERIENCE20-29y                       6.35e-16 ***
## DRIVING_EXPERIENCE30y+                         1.92e-09 ***
## VEHICLE_OWNERSHIP                               < 2e-16 ***
## VEHICLE_YEARbefore 2015                           < 2e-16 ***
## GENDERmale                                        < 2e-16 ***
## MARRIED                                          2.08e-10 ***
## PAST_ACCIDENTS                                   2.98e-08 ***
## ANNUAL_MILEAGE                                   1.56e-06 ***
## SPEEDING_VIOLATIONS                              0.900080
## DRIVING_EXPERIENCE10-19y:VEHICLE_YEARbefore 2015 0.012923 *
## DRIVING_EXPERIENCE20-29y:VEHICLE_YEARbefore 2015 1.25e-06 ***
## DRIVING_EXPERIENCE30y+:VEHICLE_YEARbefore 2015   0.000114 ***
## VEHICLE_OWNERSHIP:SPEEDING_VIOLATIONS            0.001263 **
## GENDERmale:MARRIED                               0.002426 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9990.3  on 7999  degrees of freedom
## Residual deviance: 5717.5  on 7984  degrees of freedom
## AIC: 5749.5
##
## Number of Fisher Scoring iterations: 6
```

```
# Final chosen model
outcome.2.logistic <- logistic_both
```

hypothesis tests

```
# Align both models to the exact same rows used in the full model
mf <- model.frame(outcome.1.logistic)
reduced <- update(outcome.2.logistic, data = mf)

# Likelihood-ratio test (valid when reduced is nested in full)
anova(reduced, outcome.1.logistic, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: OUTCOME ~ DRIVING_EXPERIENCE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
##     GENDER + MARRIED + PAST_ACCIDENTS + ANNUAL_MILEAGE + SPEEDING_VIOLATIONS +
##     DRIVING_EXPERIENCE:VEHICLE_YEAR + VEHICLE_OWNERSHIP:SPEEDING_VIOLATIONS +
##     GENDER:MARRIED
## Model 2: OUTCOME ~ (AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION + CREDIT_SCORE +
##     VEHICLE_OWNERSHIP + VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE +
##     VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      7984     5717.5
## 2      7845     5542.3 139   175.15   0.0205 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# AIC and BIC (smaller is better)
AIC(reduced, outcome.1.logistic)
```

```
##                     df      AIC
## reduced             16 5749.453
## outcome.1.logistic 155 5852.303
```

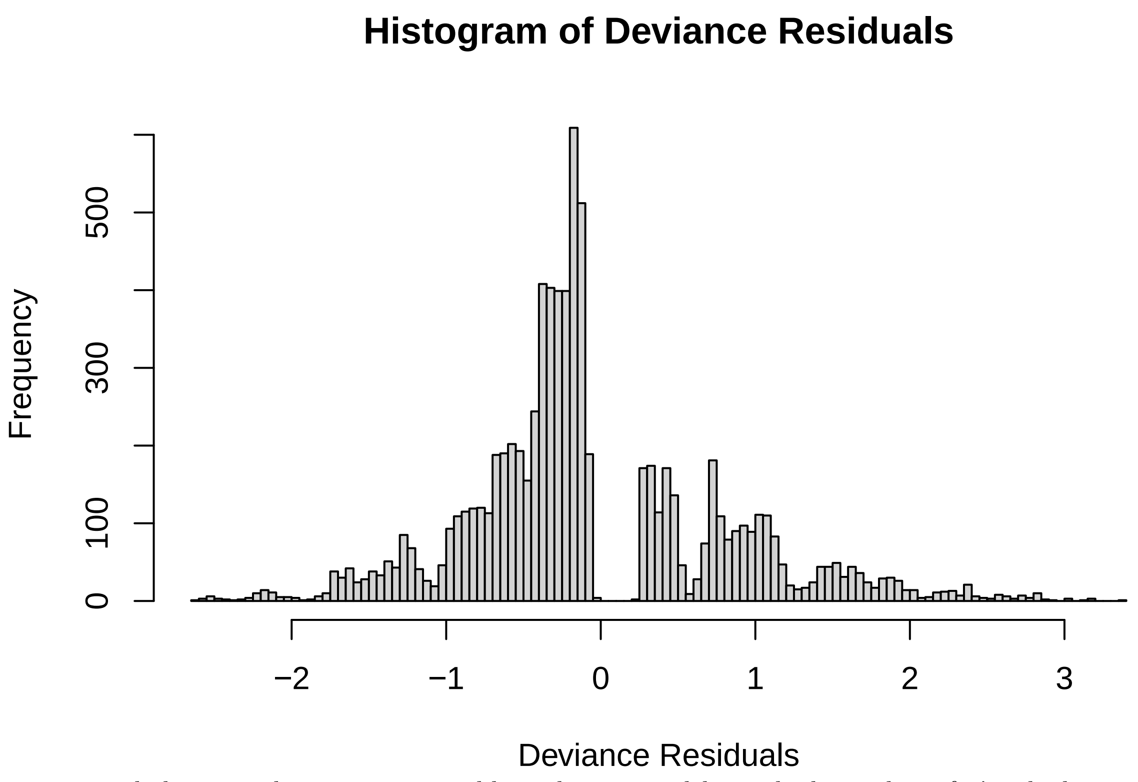
```
n <- nobs(outcome.1.logistic)
AIC(reduced, outcome.1.logistic, k = log(n))  # BIC
```

```
##                     df      AIC
## reduced             16 5861.248
## outcome.1.logistic 155 6935.318
```

- The LR test suggests some in-sample gain from the full model, but parsimony (AIC/BIC) and predictive validation likely favor the reduced model unless you see a clear lift on a hold-out set.

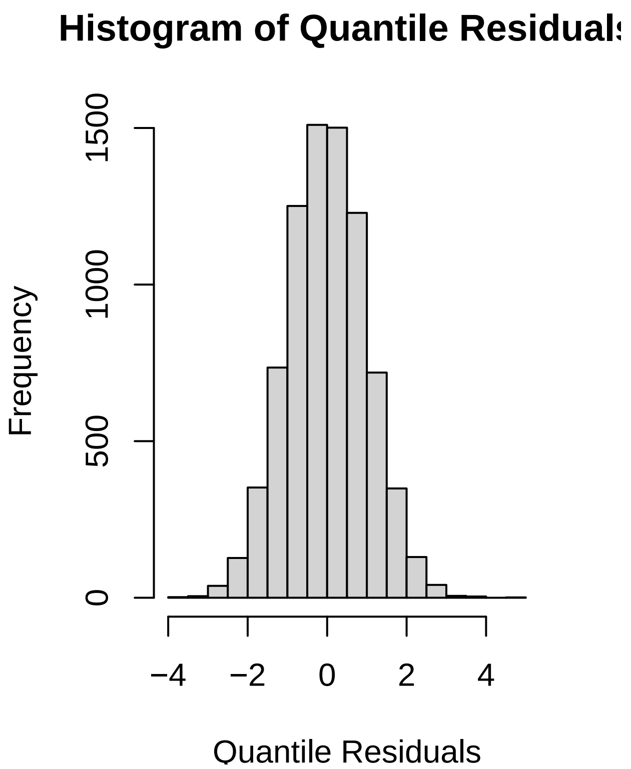
Residual analysis and finding influential points

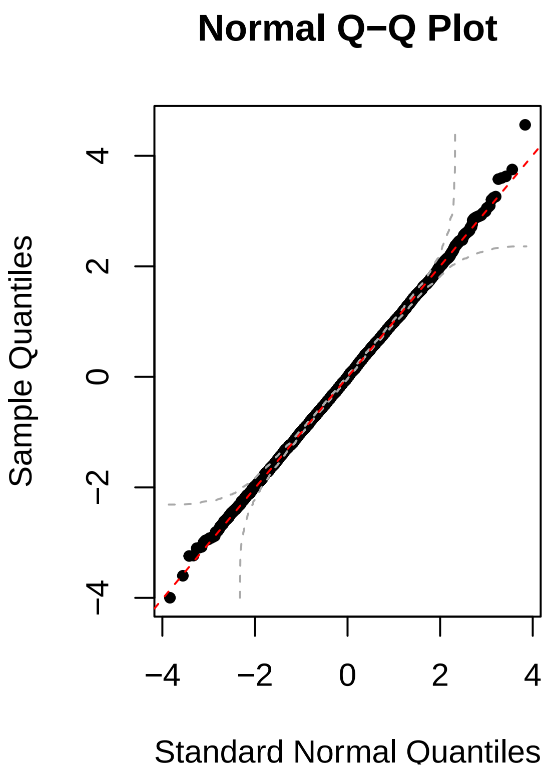
```
hist(residuals(outcome.2.logistic, type = "deviance"), main = "Histogram of Deviance Residuals", xlab =
```



- we see a non normal shape as the response variable is discrete and has a high number of 0's - look at randomised quantile residuals instead

```
qres <- qresiduals(outcome.2.logistic)
par(mfrow = c(1, 2)) # Set up a 1x2 plotting area
hist(qres, main = "Histogram of Quantile Residuals", xlab = "Quantile Residuals")
qqnorm(qres, main = "Normal Q-Q Plot")
qqline(qres, col = "red", lty = 2)
```





- We see that the quantile residuals are normally distributed, and the qq plot shows a very close fit to the normal distribution, suggesting that our logistic regression model is appropriate for the data.

Probit regression

- using binomial with probit link

```
# full basic model without interaction terms
outcome.0.probit <- glm(OUTCOME ~ .,
                        family = binomial(link = "probit"),
                        data = claimoutcome_df)

# full model with interaction terms
outcome.1.probit <- glm(OUTCOME ~(AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION +
                                  CREDIT_SCORE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
                                  MARRIED + CHILDREN + ANNUAL_MILEAGE +
                                  VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2,
                        family = binomial(link = "probit"),
                        data = claimoutcome_df)
```

Variable selection

```
probit_both <- stats::step(
  update(outcome.1.probit, . ~ 1),  # start from intercept-only
  scope = list(lower = ~ 1, upper = formula(outcome.1.probit)),
  direction = "both",
  trace = 0,
```

```r
  k = log(nobs(outcome.1.probit))   # BIC; use k = 2 for AIC
)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(probit_both)
```

```
##
## Call:
## glm(formula = OUTCOME ~ DRIVING_EXPERIENCE + VEHICLE_OWNERSHIP +
##     VEHICLE_YEAR + GENDER + MARRIED + PAST_ACCIDENTS + ANNUAL_MILEAGE +
##     SPEEDING_VIOLATIONS + DRIVING_EXPERIENCE:VEHICLE_YEAR + DRIVING_EXPERIENCE:VEHICLE_OWNERSHIP +
##     VEHICLE_OWNERSHIP:PAST_ACCIDENTS, family = binomial(link = "probit"),
##     data = claimoutcome_df)
##
## Coefficients:
##                                                   Estimate Std. Error z value
## (Intercept)                                     -5.178e-01  1.279e-01  -4.048
## DRIVING_EXPERIENCE10-19y                        -9.270e-01  1.131e-01  -8.197
## DRIVING_EXPERIENCE20-29y                        -1.676e+00  1.702e-01  -9.850
## DRIVING_EXPERIENCE30y+                          -2.302e+00  3.638e-01  -6.329
## VEHICLE_OWNERSHIP                               -1.105e+00  6.052e-02 -18.265
## VEHICLE_YEARbefore 2015                          1.234e+00  6.595e-02  18.708
## GENDERmale                                       5.655e-01  4.024e-02  14.051
## MARRIED                                         -2.609e-01  4.308e-02  -6.056
## PAST_ACCIDENTS                                  -4.949e-02  3.117e-02  -1.588
## ANNUAL_MILEAGE                                   3.933e-05  8.090e-06   4.861
## SPEEDING_VIOLATIONS                              4.211e-02  1.370e-02   3.073
## DRIVING_EXPERIENCE10-19y:VEHICLE_YEARbefore 2015 -3.399e-01  1.043e-01  -3.258
## DRIVING_EXPERIENCE20-29y:VEHICLE_YEARbefore 2015 -8.933e-01  1.392e-01  -6.417
## DRIVING_EXPERIENCE30y+:VEHICLE_YEARbefore 2015   -1.128e+00  2.288e-01  -4.931
## DRIVING_EXPERIENCE10-19y:VEHICLE_OWNERSHIP       1.597e-01  9.360e-02   1.707
## DRIVING_EXPERIENCE20-29y:VEHICLE_OWNERSHIP       8.736e-01  1.504e-01   5.809
## DRIVING_EXPERIENCE30y+:VEHICLE_OWNERSHIP         1.456e+00  3.472e-01   4.193
## VEHICLE_OWNERSHIP:PAST_ACCIDENTS                -1.342e-01  4.131e-02  -3.248
##                                                 Pr(>|z|)
## (Intercept)                                     5.16e-05 ***
## DRIVING_EXPERIENCE10-19y                        2.46e-16 ***
## DRIVING_EXPERIENCE20-29y                         < 2e-16 ***
## DRIVING_EXPERIENCE30y+                          2.47e-10 ***
## VEHICLE_OWNERSHIP                                < 2e-16 ***
## VEHICLE_YEARbefore 2015                          < 2e-16 ***
## GENDERmale                                       < 2e-16 ***
## MARRIED                                         1.39e-09 ***
## PAST_ACCIDENTS                                   0.11230
## ANNUAL_MILEAGE                                  1.17e-06 ***
## SPEEDING_VIOLATIONS                              0.00212 **
## DRIVING_EXPERIENCE10-19y:VEHICLE_YEARbefore 2015  0.00112 **
## DRIVING_EXPERIENCE20-29y:VEHICLE_YEARbefore 2015 1.39e-10 ***
## DRIVING_EXPERIENCE30y+:VEHICLE_YEARbefore 2015   8.20e-07 ***
## DRIVING_EXPERIENCE10-19y:VEHICLE_OWNERSHIP       0.08791 .
## DRIVING_EXPERIENCE20-29y:VEHICLE_OWNERSHIP      6.29e-09 ***
## DRIVING_EXPERIENCE30y+:VEHICLE_OWNERSHIP        2.76e-05 ***
## VEHICLE_OWNERSHIP:PAST_ACCIDENTS                  0.00116 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9990.3  on 7999  degrees of freedom
## Residual deviance: 5711.9  on 7982  degrees of freedom
## AIC: 5747.9
##
## Number of Fisher Scoring iterations: 7
```

```r
# Final chosen model
outcome.2.probit <- probit_both
```

hypothesis tests

```r
# Ensure both models use identical rows
mf <- model.frame(outcome.1.probit)
reduced <- update(outcome.2.probit, data = mf)

# Likelihood-ratio test (Model 1 = reduced, Model 2 = full)
anova(reduced, outcome.1.probit, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: OUTCOME ~ DRIVING_EXPERIENCE + VEHICLE_OWNERSHIP + VEHICLE_YEAR +
##     GENDER + MARRIED + PAST_ACCIDENTS + ANNUAL_MILEAGE + SPEEDING_VIOLATIONS +
##     DRIVING_EXPERIENCE:VEHICLE_YEAR + DRIVING_EXPERIENCE:VEHICLE_OWNERSHIP +
##     VEHICLE_OWNERSHIP:PAST_ACCIDENTS
## Model 2: OUTCOME ~ (AGE + GENDER + DRIVING_EXPERIENCE + EDUCATION + CREDIT_SCORE +
##     VEHICLE_OWNERSHIP + VEHICLE_YEAR + MARRIED + CHILDREN + ANNUAL_MILEAGE +
##     VEHICLE_TYPE + SPEEDING_VIOLATIONS + PAST_ACCIDENTS)^2
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      7982     5711.9
## 2      7845     5544.3 137   167.56  0.03888 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# AIC and BIC comparison (smaller is better)
AIC(reduced, outcome.1.probit)
```

```
##                   df      AIC
## reduced           18 5747.855
## outcome.1.probit 155 5854.298
```

```r
n <- nobs(outcome.1.probit)
AIC(reduced, outcome.1.probit, k = log(n))  # BIC
```

```
##                   df      AIC
## reduced           18 5873.625
## outcome.1.probit 155 6937.313
```

Comparison between logit and probit link functions

```r
# Helpers
prep_newdata_for <- function(newdata, fit) {
  nd <- newdata
  if (!is.null(fit$xlevels)) {
    for (nm in names(fit$xlevels)) if (nm %in% names(nd)) {
      nd[[nm]] <- factor(nd[[nm]], levels = fit$xlevels[[nm]])
    }
  }
  droplevels(nd)
}
log_loss <- function(y, p, eps = 1e-15) {
  p <- pmin(pmax(p, eps), 1 - eps)
  -mean(y * log(p) + (1 - y) * log(1 - p), na.rm = TRUE)
}
brier <- function(y, p) mean((p - y)^2, na.rm = TRUE)
calibration_df <- function(y, p, bins = 10, label = "model") {
  brks <- quantile(p, probs = seq(0, 1, length.out = bins + 1), na.rm = TRUE)
  g <- cut(p, breaks = unique(brks), include.lowest = TRUE)
  dplyr::summarise(dplyr::group_by(data.frame(y, p, g), g),
                   mean_p = mean(p, na.rm = TRUE),
                   obs    = mean(y, na.rm = TRUE),
                   n      = dplyr::n()) |>
    mutate(model = label)
}

# Prepare test data for each model (handles factors/levels)
test_logit  <- prep_newdata_for(test_data_imputed, outcome.2.logistic)
test_probit <- prep_newdata_for(test_data_imputed, outcome.2.probit)

# Predictions
y  <- test_logit$OUTCOME
p_logit  <- predict(outcome.2.logistic, newdata = test_logit,  type = "response")
p_probit <- predict(outcome.2.probit,   newdata = test_probit, type = "response")

# Keep rows where both models produce probabilities
keep <- is.finite(p_logit) & is.finite(p_probit) & !is.na(y)
y <- y[keep]; p_logit <- p_logit[keep]; p_probit <- p_probit[keep]

# Build ROC objects with pROC
roc_logit  <- pROC::roc(y, p_logit,  quiet = TRUE)
roc_probit <- pROC::roc(y, p_probit, quiet = TRUE)

# Get numeric AUCs (either of these styles works)
auc_logit  <- as.numeric(pROC::auc(roc_logit))
auc_probit <- as.numeric(pROC::auc(roc_probit))
# or:
# auc_logit  <- as.numeric(roc_logit$auc)
# auc_probit <- as.numeric(roc_probit$auc)

metrics <- data.frame(
  Model    = c("Logit", "Probit"),
  AUC     = c(auc_logit, auc_probit),
  LogLoss = c(log_loss(y, p_logit),  log_loss(y, p_probit)),
  Brier   = c(brier(y, p_logit),     brier(y, p_probit))
)
print(metrics, row.names = FALSE)
```

```
##   Model       AUC   LogLoss     Brier
##   Logit 0.8811706 0.3921841 0.1246430
##  Probit 0.8817748 0.3903196 0.1242813
```

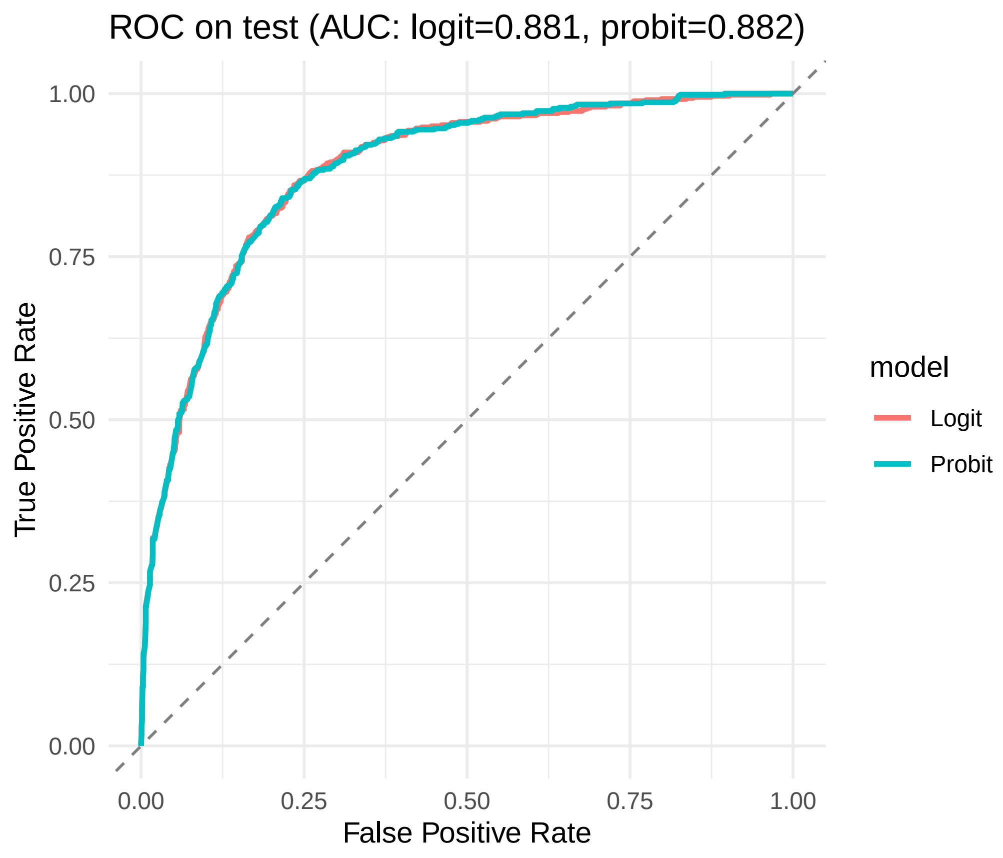
```
# AUC difference test
auc_test <- pROC::roc.test(roc_logit, roc_probit, method = "delong")
cat(sprintf("\nDeLong test for AUC difference: z = %.3f, p = %.4f\n",
            auc_test$statistic, auc_test$p.value))
```

```
##
## DeLong test for AUC difference: z = -0.580, p = 0.5618
```

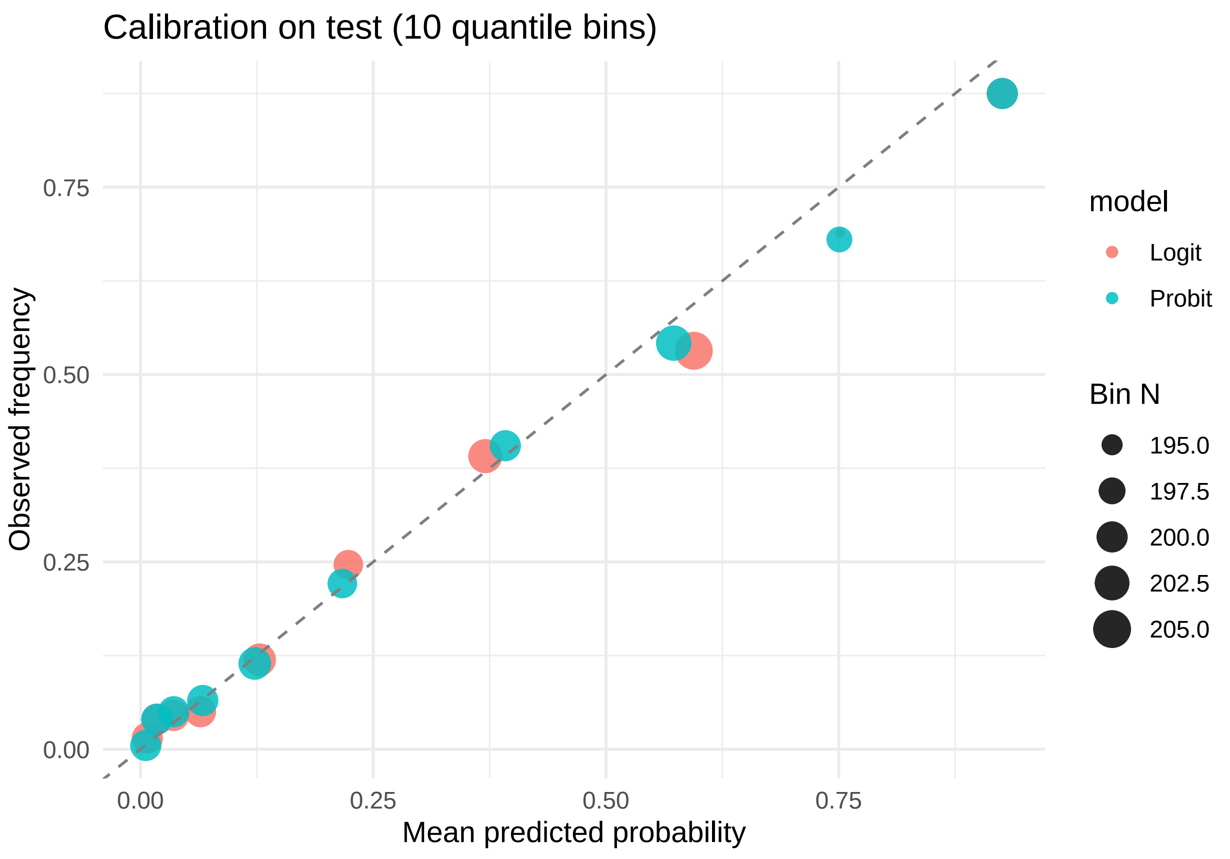
```
# ROC curves (overlaid)
df_roc_logit <- data.frame(
  fpr = rev(1 - roc_logit$specificities),
  tpr = rev(roc_logit$sensitivities),
  model = "Logit"
)
df_roc_probit <- data.frame(
  fpr = rev(1 - roc_probit$specificities),
  tpr = rev(roc_probit$sensitivities),
  model = "Probit"
)
df_roc <- rbind(df_roc_logit, df_roc_probit)

p_roc <- ggplot(df_roc, aes(x = fpr, y = tpr, color = model)) +
  geom_line(size = 1) +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "gray50") +
  labs(title = sprintf("ROC on test (AUC: logit=%.3f, probit=%.3f)",
                       metrics$AUC[metrics$Model=="Logit"],
                       metrics$AUC[metrics$Model=="Probit"]),
       x = "False Positive Rate", y = "True Positive Rate") +
  theme_minimal() + coord_equal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

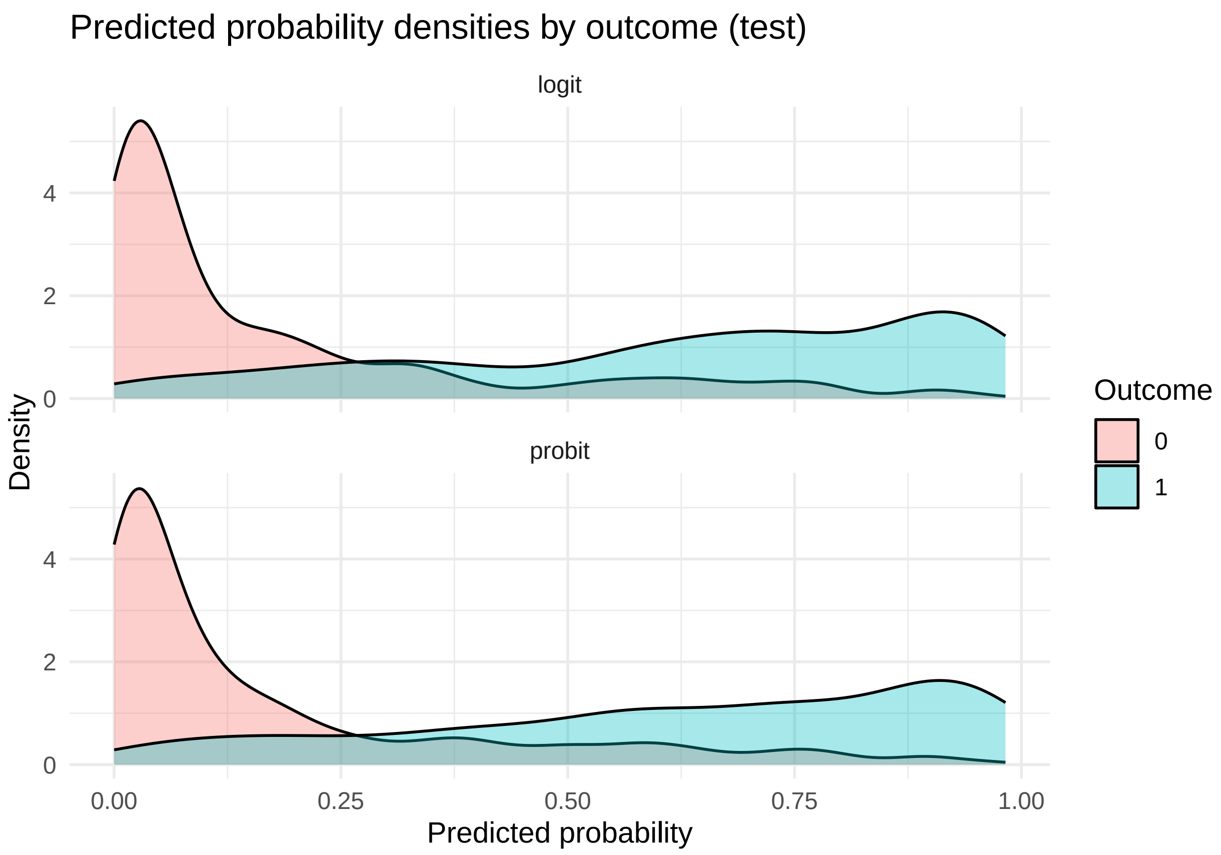
```
print(p_roc)
```



```r
# Calibration (reliability) plot
cal_logit  <- calibration_df(y, p_logit,  bins = 10, label = "Logit")
cal_probit <- calibration_df(y, p_probit, bins = 10, label = "Probit")
cal <- rbind(cal_logit, cal_probit)

p_cal <- ggplot(cal, aes(x = mean_p, y = obs, color = model, size = n)) +
  geom_point(alpha = 0.85) +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "gray50") +
  scale_size_continuous(name = "Bin N") +
  labs(title = "Calibration on test (10 quantile bins)",
       x = "Mean predicted probability", y = "Observed frequency") +
  theme_minimal()
print(p_cal)
```



```r
# Optional: density of predicted probabilities by class (separation plot)
df_prob <- data.frame(y = factor(y, levels = c(0,1)),
                      logit = p_logit, probit = p_probit) |>
  tidyr::pivot_longer(cols = c(logit, probit), names_to = "model", values_to = "p")
p_den <- ggplot(df_prob, aes(x = p, fill = y)) +
  geom_density(alpha = 0.35) +
  facet_wrap(~ model, ncol = 1) +
  labs(title = "Predicted probability densities by outcome (test)",
       x = "Predicted probability", y = "Density", fill = "Outcome") +
  theme_minimal()
print(p_den)
```



- Predicted-probability densities by outcome Clear separation: class 0 probabilities are heavily concentrated near 0; class 1 probabilities are mostly 0.5–1.0 with a peak near 0.8–0.95. Overlap is mainly in 0.1–0.3, which is where most errors will occur. Logit vs probit look almost identical; probit shows a hair more mass near very high probabilities, but the difference is tiny.
- Calibration (10 quantile bins) Points lie very close to the 45° line for both models across the range → well-calibrated probabilities. Minor, likely noise-level deviations: around 0.7–0.8 the probit is slightly underconfident (observed > predicted), around ~0.55 the logit is slightly overconfident. Bin sizes are ~200, so sampling variation can explain this.
- ROC and AUC Curves overlap almost perfectly; test AUCs are 0.881 (logit) vs 0.882 (probit). The 0.001 gap is negligible and would not be statistically or practically significant (DeLong test would almost surely be non-significant). Bottom line
- On the test set, logit and probit are essentially indistinguishable
- Choose Logit due to its superior explanatory power