CODING CHALLENGE – DATA SCIENTIST

Jae-Eun Nam

2024-06-17

1 Aufgabe

Unter https://www.openml.org/d/41214 und https://www.openml.org/d/41215 finden Sie zwei Datensätze eines französischen Automobilversicherers. Diese beinhalten Risikomerkmale und Schadeninformationen zu Kraftfahrt-Haftpflicht-Versicherungsverträgen (eine Datensatzbeschreibung finden Sie am Ende dieses Textes). Ihre Aufgabe besteht in der Modellierung der zu erwartenden Schadenhöhe pro Versicherungsnehmer und Jahr anhand der Risikomerkmale der Kunden. Dieser Wert ist Basis für die Berechnung eines fairen Versicherungsbeitrags.

2 Datensätze

2.1 freMTPL2freq

Variable	**Beschreibung**
`IDpol`	ID des Vertrags
`Exposure`	Länge des Versicherungszeitraums (in Jahren)
`BonusMalus`	Schadenfreiheitsrabatt
`ClaimNb`	Anzahl Schäden im Versicherungszeitraum
`DrivAge`	Alter des Versicherungsnehmers
`Area`	Area-Code des Versicherungsnehmers
`Region`	Region des Versicherungsnehmers
`Density`	Anzahl der Einwohner pro km2 im Wohnort des Versicherungsnehmers
`VehBrand`	Marke des versicherten Kfz
`VehGas`	Antrieb des versicherten Kfz
`VehPower`	Leistung des versicherten Kfz
`VehAge`	Alter des versicherten Kfz

2.2 freMTPL2freq

Variable	**Beschreibung**
`IDpol`	ID des Vertrags
`ClaimAmount`	Höhe der einzelnen Schadenaufwände (mehrere Einträge pro Vertrag, falls im Zeitraum mehrere Schäden vorhanden waren.)

3 Datenaufbereitung

```r
### Data Preprocessing

# read datasets
freMTPL2freq = readARFF('freMTPL2freq.arff')
```

```
## Parse with reader=readr : freMTPL2freq.arff

## header: 0.013000; preproc: 0.363000; data: 0.796000; postproc: 0.037000; total: 1.209000
```

```r
freMTPL2sev = readARFF('freMTPL2sev.arff')
```

```
## Parse with reader=readr : freMTPL2sev.arff

## header: 0.001000; preproc: 0.007000; data: 0.012000; postproc: 0.000000; total: 0.020000
```

```r
str(freMTPL2freq) # 678013 contracts
```

```
## 'data.frame':    678013 obs. of  12 variables:
##  $ IDpol     : num  1 3 5 10 11 13 15 17 18 21 ...
##  $ ClaimNb   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Exposure  : num  0.1 0.77 0.75 0.09 0.84 0.52 0.45 0.27 0.71 0.15 ...
##  $ Area      : Factor w/ 6 levels "A","B","C","D",..: 4 4 2 2 2 5 5 3 3 2 ...
##  $ VehPower  : num  5 5 6 7 7 6 6 7 7 7 ...
##  $ VehAge    : num  0 0 2 0 0 2 2 0 0 0 ...
##  $ DrivAge   : num  55 55 52 46 46 38 38 33 33 41 ...
##  $ BonusMalus: num  50 50 50 50 50 50 50 68 68 50 ...
##  $ VehBrand  : Factor w/ 11 levels "B1","B10","B11",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ VehGas    : chr  "Regular" "Regular" "Diesel" "Diesel" ...
##  $ Density   : num  1217 1217 54 76 76 ...
##  $ Region    : Factor w/ 22 levels "R11","R21","R22",..: 18 18 3 15 15 8 8 20 20 12 ...
```

```r
str(freMTPL2sev) # 26639 claims
```

```
## 'data.frame':    26639 obs. of  2 variables:
##  $ IDpol      : num  1552 1010996 4024277 4007252 4046424 ...
##  $ ClaimAmount: num  995 1128 1851 1204 1204 ...
```

```r
# sum claim amounts that belongs to same contract
groupedFreMTPL2sev = freMTPL2sev %>%
  group_by(IDpol) %>%
  summarize(TotalClaimAmount = sum(ClaimAmount))

# join data by 'IDpol'
tmpDf = left_join(freMTPL2freq, groupedFreMTPL2sev, by = 'IDpol')

# contracts without matching observation in freMTPL2sev
# should have claim amount of 0
tmpDf = tmpDf %>%
  mutate(TotalClaimAmount = replace_na(TotalClaimAmount, 0))

# however some claims are listed in freMTPL2freq, but not in freMTPL2sev
# remove these 9116 cases (need to check, if reasonable)
# 668897 from 678013 observations left
tmpDf = tmpDf %>%
  filter(!(ClaimNb > 0 & TotalClaimAmount == 0))
```

```r
# select relevant data
df = tmpDf %>%
  mutate(VehGas = factor(VehGas),
         ClaimAmountPerYear = TotalClaimAmount / Exposure) %>%
  select(!c(IDpol, ClaimNb, Exposure, TotalClaimAmount))
str(df)
```

```
## 'data.frame':    668897 obs. of  10 variables:
##  $ Area              : Factor w/ 6 levels "A","B","C","D",..: 6 2 5 6 1 4 4 5 5 4 ...
##  $ VehPower          : num  7 12 4 10 5 10 5 5 4 9 ...
##  $ VehAge            : num  1 5 0 0 0 6 0 0 10 0 ...
##  $ DrivAge           : num  61 50 36 51 45 54 34 44 24 60 ...
##  $ BonusMalus        : num  50 60 85 100 50 50 64 50 105 50 ...
##  $ VehBrand          : Factor w/ 11 levels "B1","B10","B11",..: 4 4 4 4 4 4 4 4 1 4 ...
##  $ VehGas            : Factor w/ 2 levels "Diesel","Regular": 2 1 2 2 2 1 2 2 2 2 ...
##  $ Density           : num  27000 56 4792 27000 12 ...
##  $ Region            : Factor w/ 22 levels "R11","R21","R22",..: 1 6 1 1 16 21 8 21 1 21 ...
##  $ ClaimAmountPerYear: num  404 14156 10404 17474 12860 ...
```

4 Deskriptive Analyse

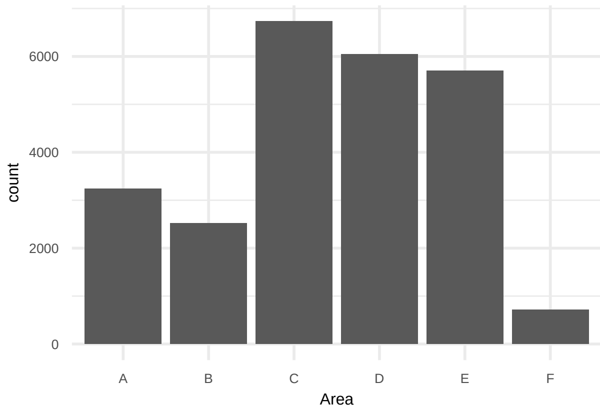
4.1 Univariate Analyse

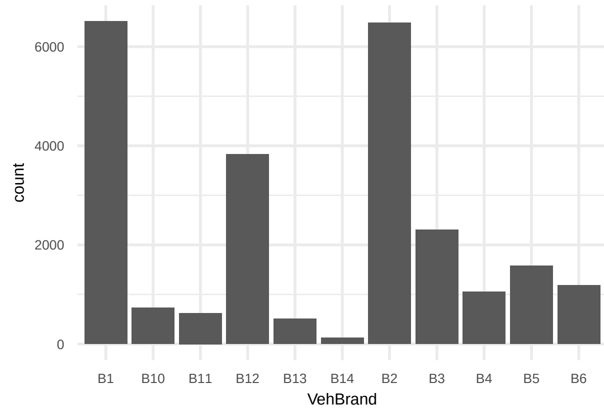
```
##  Area         VehPower          VehAge          DrivAge      
##  A:102363   Min.   : 4.000   Min.   :  0.000   Min.   : 18.00  
##  B: 74353   1st Qu.: 5.000   1st Qu.:  2.000   1st Qu.: 34.00  
##  C:189254   Median : 6.000   Median :  6.000   Median : 44.00  
##  D:149747   Mean   : 6.457   Mean   :  7.085   Mean   : 45.45  
##  E:135568   3rd Qu.: 7.000   3rd Qu.: 11.000   3rd Qu.: 55.00  
##  F: 17612   Max.   :15.000   Max.   :100.000   Max.   :100.00  
##                                                                
##    BonusMalus        VehBrand          VehGas          Density     
##  Min.   : 50.00   B12    :161594   Diesel :329127   Min.   :    1  
##  1st Qu.: 50.00   B1     :161068   Regular:339770   1st Qu.:   92  
##  Median : 50.00   B2     :158220                    Median :  393  
##  Mean   : 59.78   B3     : 53028                    Mean   : 1791  
##  3rd Qu.: 65.00   B5     : 34418                    3rd Qu.: 1658  
##  Max.   :230.00   B6     : 28349                    Max.   :27000  
##                   (Other): 72220                                   
##      Region       ClaimAmountPerYear
##  R24    :158055   Min.   :       0  
##  R82    : 83994   1st Qu.:       0  
##  R93    : 78443   Median :       0  
##  R11    : 68471   Mean   :     388  
##  R53    : 41340   3rd Qu.:       0  
##  R52    : 38340   Max.   :18524548  
##  (Other):200254
```

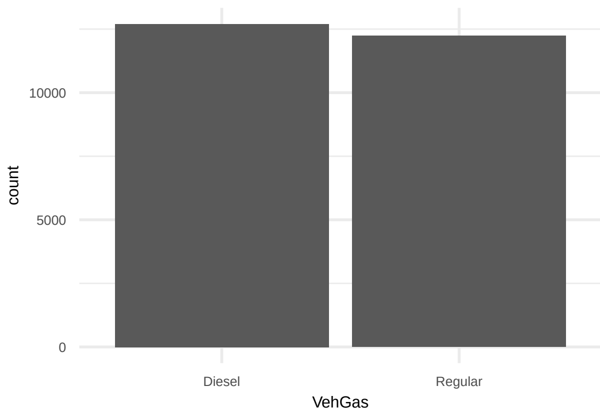
```
## [1] "5-number summary for ClaimAmountPerYear after excluding 0-valued observations:"
```

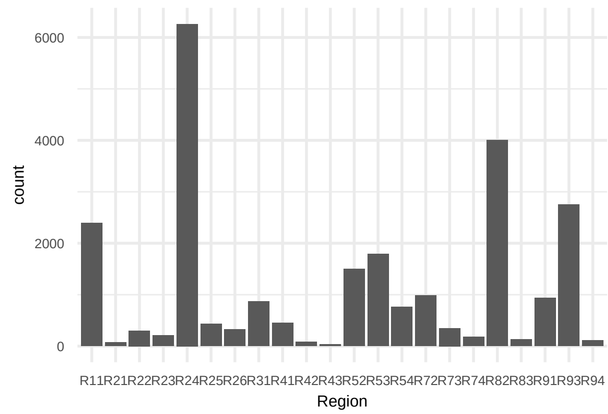
```
## [1]        1.000     1128.000     1504.160     3352.394 18524548.000
```

4.1.1 Kategoriale Variablen

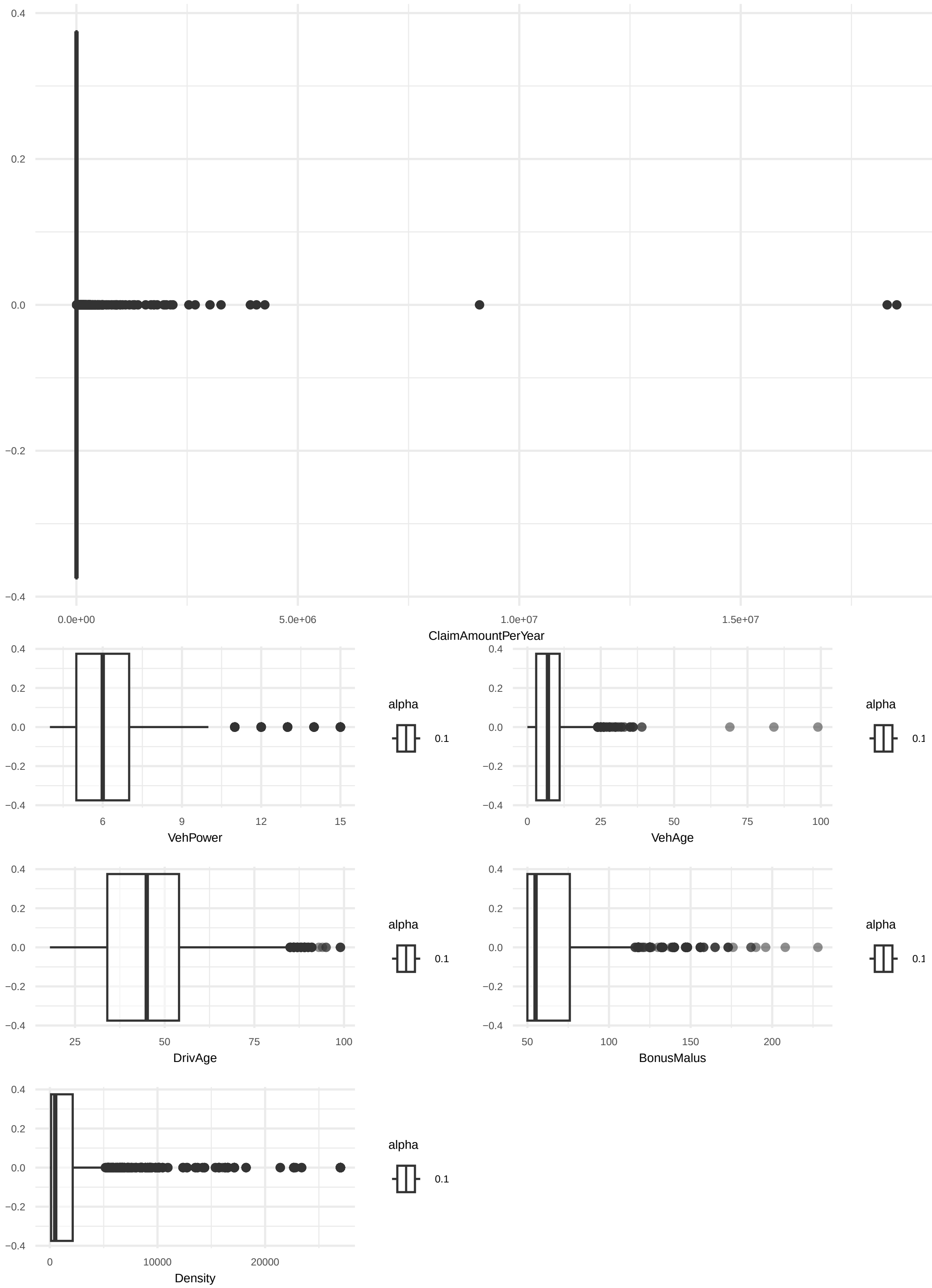






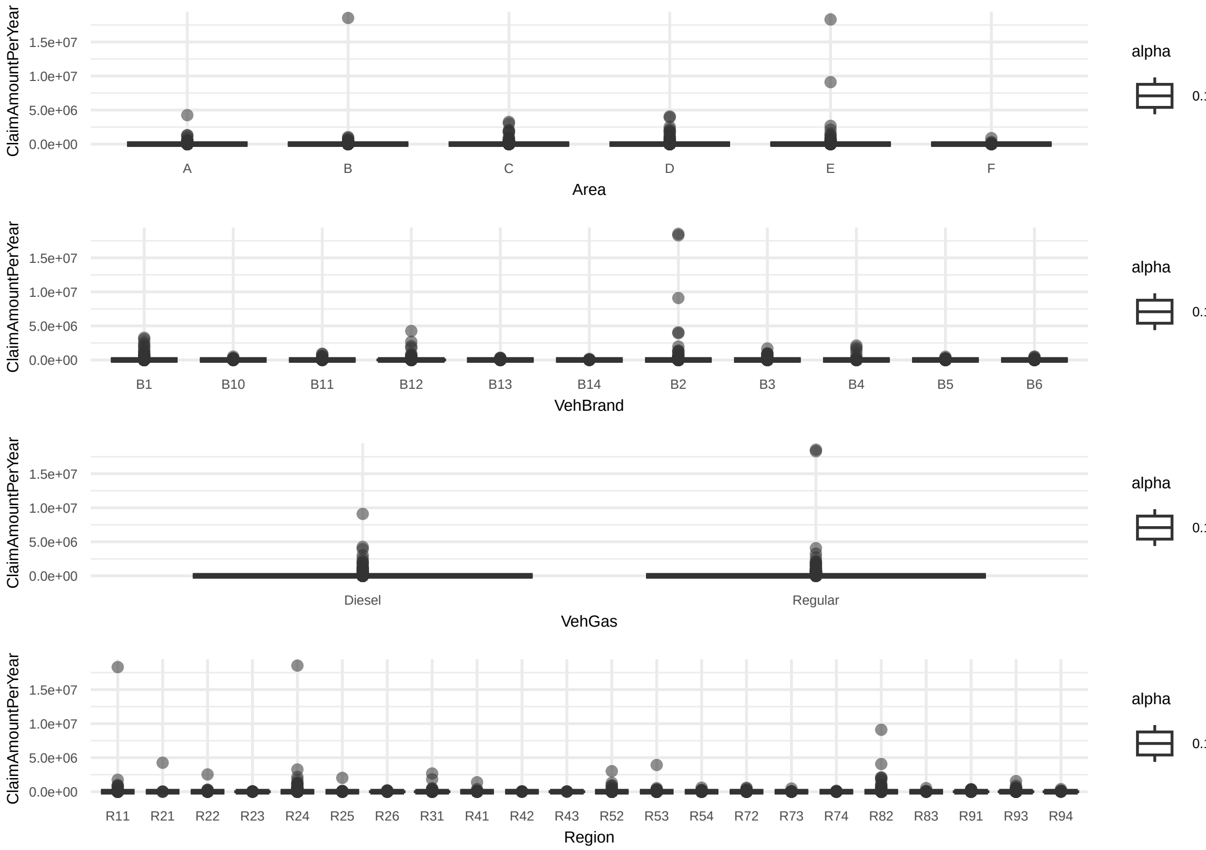


4.1.2 Numerische Variablen

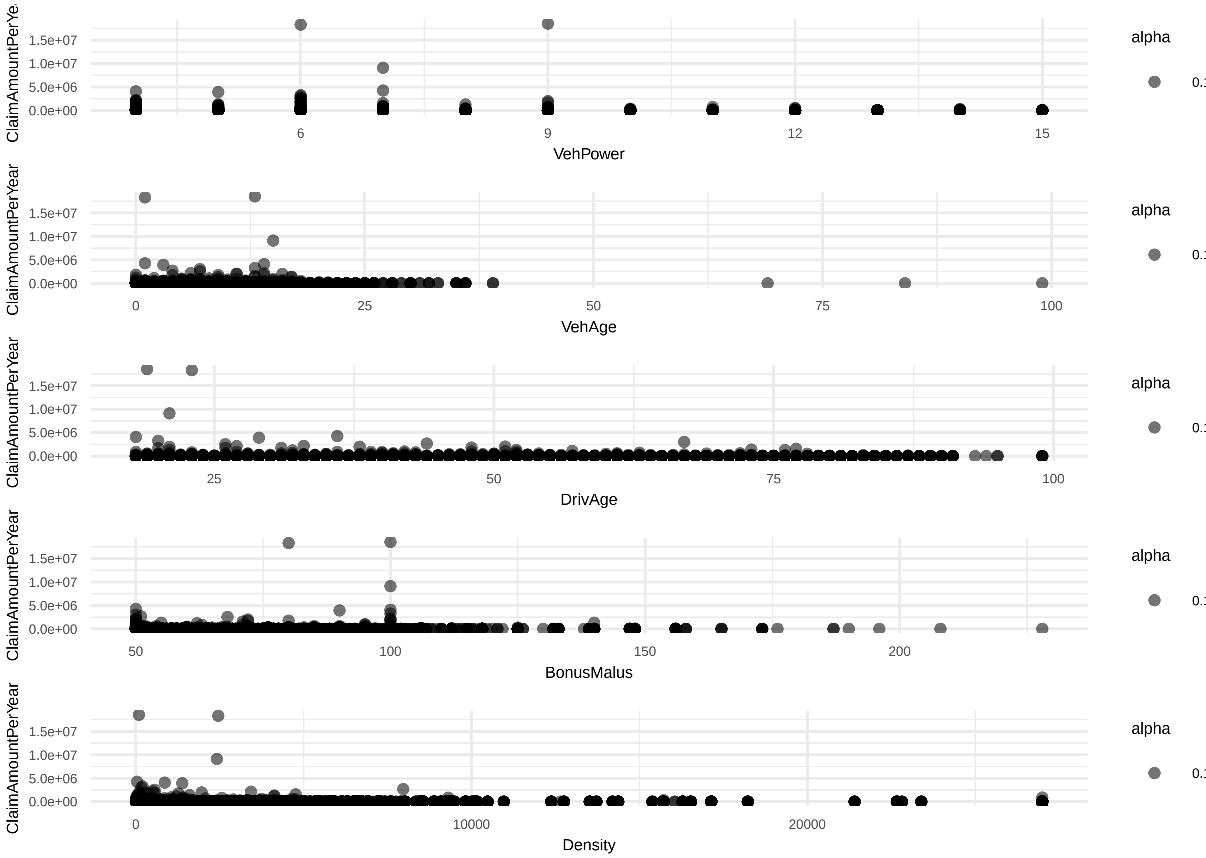


4.2 Bivariate Analyse

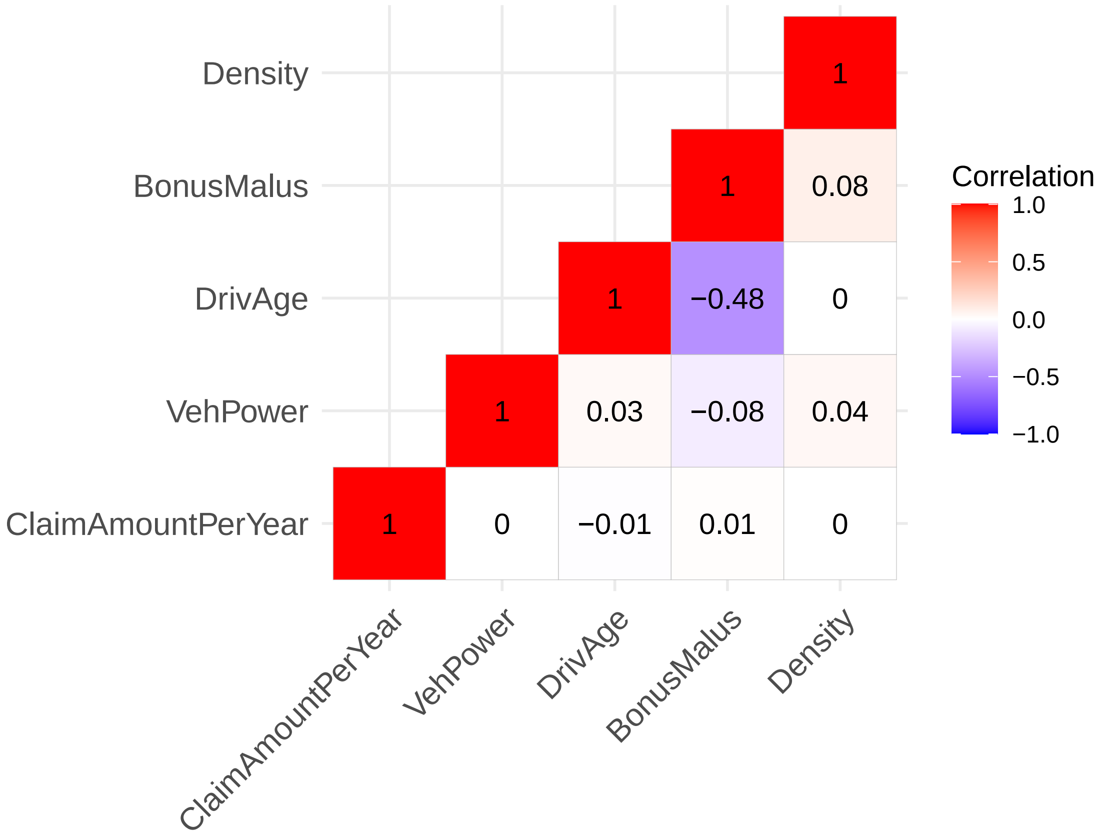
4.2.1 Kategoriale Variablen



4.2.2 Numerische Variablen



4.2.3 Korrelation zwischen numerischen Variablen



5 Modellierung

5.1 Random Forest

```
## random forest
learnerRf = lrn('regr.ranger', predict_type = 'response')
learnerRf$train(taskClaim, row_ids = splits$train)
```

```
## Growing trees.. Progress: 7%. Estimated remaining time: 7 minutes, 4 seconds.
## Growing trees.. Progress: 14%. Estimated remaining time: 6 minutes, 40 seconds.
## Growing trees.. Progress: 20%. Estimated remaining time: 6 minutes, 11 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 5 minutes, 31 seconds.
## Growing trees.. Progress: 34%. Estimated remaining time: 5 minutes, 4 seconds.
## Growing trees.. Progress: 41%. Estimated remaining time: 4 minutes, 35 seconds.
## Growing trees.. Progress: 47%. Estimated remaining time: 4 minutes, 4 seconds.
## Growing trees.. Progress: 54%. Estimated remaining time: 3 minutes, 34 seconds.
## Growing trees.. Progress: 60%. Estimated remaining time: 3 minutes, 5 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 2 minutes, 35 seconds.
## Growing trees.. Progress: 73%. Estimated remaining time: 2 minutes, 4 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 1 minute, 33 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 1 minute, 1 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 30 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
```

```
print(learnerRf$model)
```

```
## Ranger result
##
## Call:
##  ranger::ranger(dependent.variable.name = task$target_names, data = task$data(),      case.weights =
##
## Type:                             Regression
## Number of trees:                  500
## Sample size:                      401338
## Number of independent variables:  9
## Mtry:                             3
## Target node size:                 5
## Variable importance mode:         none
## Splitrule:                        variance
## OOB prediction error (MSE):       1282046957
## R squared (OOB):                  -0.03173119
```

5.2 Xgboost

```
fencoder = po("encode", method = "treatment", affect_columns = selector_type("factor"))
learnerXgboost = lrn('regr.xgboost', predict_type = 'response')
learnerXgboost = as_learner(fencoder %>>% learnerXgboost)
learnerXgboost$train(taskClaim, row_ids = splits$train)
print(learnerXgboost$model$regr.xgboost$model)
```

```
## ##### xgb.Booster
## raw: 8.7 Kb
## call:
##   xgboost::xgb.train(data = data, nrounds = 1L, verbose = 0L, nthread = 1L,
##     objective = "reg:squarederror")
```

```
## params (as set within xgb.train):
##   nthread = "1", objective = "reg:squarederror", validate_parameters = "TRUE"
## xgb.attributes:
##   niter
## # of features: 42
## niter: 1
## nfeatures : 42
```

5.3 Tweedie GLM

```r
## Tweedie-GLM
trainDf = df[splits$train, ]
testDf = df[splits$test, ]
trainedGlm = glm(ClaimAmountPerYear ~ ., data = trainDf, family = tweedie(1.1))
summary(trainedGlm)
```

```
##
## Call:
## glm(formula = ClaimAmountPerYear ~ ., family = tweedie(1.1),
##     data = trainDf)
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.687e-01  3.260e-02  17.445  < 2e-16 ***
## AreaB         -1.590e-02  1.838e-02  -0.865 0.386993
## AreaC         -1.730e-02  1.566e-02  -1.105 0.269286
## AreaD         -2.765e-02  1.624e-02  -1.702 0.088682 .
## AreaE         -7.982e-02  1.924e-02  -4.149 3.33e-05 ***
## AreaF         -1.323e-01  5.223e-02  -2.533 0.011299 *
## VehPower      -3.757e-04  1.992e-03  -0.189 0.850452
## VehAge         7.063e-04  7.596e-04   0.930 0.352434
## DrivAge        1.205e-03  3.391e-04   3.554 0.000379 ***
## BonusMalus    -1.179e-03  1.938e-04  -6.083 1.18e-09 ***
## VehBrandB10   -6.213e-03  2.771e-02  -0.224 0.822612
## VehBrandB11    1.970e-03  2.836e-02   0.069 0.944620
## VehBrandB12    2.918e-02  1.380e-02   2.114 0.034481 *
## VehBrandB13    2.732e-02  3.981e-02   0.686 0.492539
## VehBrandB14    3.376e-02  7.271e-02   0.464 0.642427
## VehBrandB2    -3.271e-02  9.786e-03  -3.342 0.000831 ***
## VehBrandB3    -3.197e-03  1.459e-02  -0.219 0.826566
## VehBrandB4     6.190e-03  2.117e-02   0.292 0.769987
## VehBrandB5     2.716e-02  2.332e-02   1.165 0.244063
## VehBrandB6     3.127e-02  2.546e-02   1.228 0.219292
## VehGasRegular -1.542e-02  7.687e-03  -2.006 0.044841 *
## Density        9.890e-06  2.994e-06   3.304 0.000954 ***
## RegionR21      1.420e-01  1.724e-01   0.824 0.410209
## RegionR22      1.725e-02  2.516e-02   0.685 0.493161
## RegionR23      1.050e-01  5.995e-02   1.751 0.079930 .
## RegionR24      4.677e-02  1.366e-02   3.423 0.000618 ***
## RegionR25      3.226e-02  2.773e-02   1.163 0.244672
## RegionR26      8.909e-02  4.952e-02   1.799 0.072021 .
## RegionR31      4.385e-02  1.981e-02   2.214 0.026833 *
## RegionR41      3.276e-02  3.006e-02   1.090 0.275747
## RegionR42      9.900e-02  1.086e-01   0.912 0.361984
```

```
## RegionR43      6.202e-02  1.047e-01   0.592 0.553740    
## RegionR52      4.075e-02  1.798e-02   2.267 0.023406 *  
## RegionR53      6.723e-02  2.244e-02   2.996 0.002740 ** 
## RegionR54      5.651e-02  2.904e-02   1.946 0.051670 .  
## RegionR72      6.669e-02  2.479e-02   2.690 0.007139 ** 
## RegionR73      9.800e-02  4.485e-02   2.185 0.028892 *  
## RegionR74      5.128e-02  6.410e-02   0.800 0.423730    
## RegionR82      1.699e-02  1.224e-02   1.389 0.164982    
## RegionR83      4.380e-02  5.039e-02   0.869 0.384667    
## RegionR91      6.360e-02  2.420e-02   2.628 0.008587 ** 
## RegionR93      4.867e-02  1.484e-02   3.280 0.001037 ** 
## RegionR94      4.432e-02  5.983e-02   0.741 0.458810    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Tweedie family taken to be 393271)
## 
##     Null deviance: 846205724  on 401337  degrees of freedom
## Residual deviance: 761893495  on 401295  degrees of freedom
## AIC: NA
## 
## Number of Fisher Scoring iterations: 17
```

6 Modellvergleich

```
## Predicting.. Progress: 54%. Estimated remaining time: 26 seconds.

## Predicting.. Progress: 79%. Estimated remaining time: 8 seconds.

## [1] "Random Forest evaluated on train set:"

##    regr.rmse     regr.rsq
## 2.047622e+04 6.625859e-01

## [1] "Random Forest evaluated on test set:"

##     regr.rmse      regr.rsq
##  4.045381e+04 -4.125532e-02

## [1] "XGboost evaluated on train set:"

##    regr.rmse     regr.rsq
## 3.131777e+04 2.106939e-01

## [1] "XGboost Forest evaluated on test set:"

##     regr.rmse      regr.rsq
##  4.002574e+04 -1.933543e-02

## [1] "GLM evaluated on train set, RMSE:"

## [1] 35235.52

## [1] "GLM evaluated on test set, RMSE:"

## [1] 39643.82
```

7 Feature Importance

```
## [1] "Variablen geordnet nach der absoluten Größe der Koeffizienten:"
```

```
##   (Intercept)     RegionR21         AreaF     RegionR23     RegionR42
##  5.687330e-01  1.419946e-01 -1.323135e-01  1.049751e-01  9.900267e-02
##     RegionR73     RegionR26         AreaE     RegionR53     RegionR72
##  9.800480e-02  8.908555e-02 -7.981652e-02  6.722508e-02  6.668574e-02
##     RegionR91     RegionR43     RegionR54     RegionR74     RegionR93
##  6.360443e-02  6.202240e-02  5.650874e-02  5.127572e-02  4.866534e-02
##     RegionR24     RegionR94     RegionR31     RegionR83     RegionR52
##  4.677252e-02  4.432372e-02  4.385164e-02  4.380328e-02  4.075110e-02
##    VehBrandB14     RegionR41     VehBrandB2     RegionR25    VehBrandB6
##  3.376023e-02  3.276202e-02 -3.270872e-02  3.226216e-02  3.127059e-02
##    VehBrandB12         AreaD   VehBrandB13    VehBrandB5         AreaC
##  2.918241e-02 -2.764565e-02  2.732183e-02  2.716402e-02 -1.729716e-02
##     RegionR22     RegionR82         AreaB VehGasRegular   VehBrandB10
##  1.724523e-02  1.698853e-02 -1.589825e-02 -1.542069e-02 -6.212516e-03
##    VehBrandB4    VehBrandB3   VehBrandB11       DrivAge    BonusMalus
##  6.189829e-03 -3.196726e-03  1.970231e-03  1.205306e-03 -1.178956e-03
##        VehAge      VehPower       Density
##  7.063092e-04 -3.756663e-04  9.890456e-06
```

8 Fazit

- Schadenaufwände sind nicht-negative Daten mit vielen Nullen.
- Tweedie Verteilung ist geeignet für die Modellierung von Schadenaufwänden.
- Von den drei Modellen (Random Forest, XGboost, GLM) hatte GLM das beste Ergebnis.
- Einzelne Regionen und Area scheinen, einen größeren Zusammenhang mit der Schadenhöhe zu haben.

9 Verbesserungsvorschläge

- Hyperparameter Tuning (insb. zur Vermeidung von Overfitting)
- Methoden für Imbalenced Data, z.B. Oversampling, Weighting