Analysis

Imports

```
library(mgcv)
library(tidyverse)
library(caret)
library(mice)


df <- read.csv("data/train_simple.csv", stringsAsFactors = TRUE)
df <- select(df, -coverage)  # Doesn't seem particularly useful
```

Fit linear regression without imputing income

As lots of the incomes are zero, it makes sense to impute these, as employment status is a categorical variable anyway.

```
res <- lm(total_claim_amount ~ employment_status + location_code +
             vehicle_class + income + monthly_premium_auto +
             employment_status*monthly_premium_auto +
             location_code*monthly_premium_auto, data = df)

summary(res)
```

```
##
## Call:
## lm(formula = total_claim_amount ~ employment_status + location_code +
##     vehicle_class + income + monthly_premium_auto + employment_status *
##     monthly_premium_auto + location_code * monthly_premium_auto,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -547.60  -63.02  -28.05   55.76 1412.19
##
## Coefficients:
##                                                      Estimate Std. Error
## (Intercept)                                        -5.851e+01  2.290e+01
## employment_statusEmployed                           7.078e+01  2.057e+01
## employment_statusMedical Leave                     -6.527e+00  2.851e+01
## employment_statusRetired                            1.125e+02  3.148e+01
## employment_statusUnemployed                        -1.402e+01  2.134e+01
## location_codeSuburban                               1.934e+01  1.174e+01
## location_codeUrban                                  8.502e+00  1.466e+01
## vehicle_classLuxury Car                             3.860e+01  1.591e+01
## vehicle_classLuxury SUV                             2.375e+01  1.524e+01
## vehicle_classSports Car                            -5.391e+00  7.451e+00
## vehicle_classSUV                                   -2.224e+00  5.266e+00
```

```
## vehicle_classTwo-Door Car                            1.680e+00  3.685e+00
## income                                              -1.284e-04  8.227e-05
## monthly_premium_auto                                 1.890e+00  2.372e-01
## employment_statusEmployed:monthly_premium_auto      -7.825e-01  2.049e-01
## employment_statusMedical Leave:monthly_premium_auto  3.508e-01  2.926e-01
## employment_statusRetired:monthly_premium_auto       -1.359e+00  3.236e-01
## employment_statusUnemployed:monthly_premium_auto     1.209e+00  2.141e-01
## location_codeSuburban:monthly_premium_auto           4.063e+00  1.219e-01
## location_codeUrban:monthly_premium_auto              2.366e+00  1.548e-01
##                                                     t value Pr(>|t|)
## (Intercept)                                          -2.555 0.010652 *
## employment_statusEmployed                             3.441 0.000583 ***
## employment_statusMedical Leave                       -0.229 0.818890
## employment_statusRetired                              3.572 0.000356 ***
## employment_statusUnemployed                          -0.657 0.511302
## location_codeSuburban                                 1.647 0.099660 .
## location_codeUrban                                    0.580 0.561942
## vehicle_classLuxury Car                               2.426 0.015271 *
## vehicle_classLuxury SUV                               1.559 0.119071
## vehicle_classSports Car                              -0.724 0.469384
## vehicle_classSUV                                     -0.422 0.672753
## vehicle_classTwo-Door Car                             0.456 0.648411
## income                                               -1.560 0.118711
## monthly_premium_auto                                  7.967 1.85e-15 ***
## employment_statusEmployed:monthly_premium_auto       -3.819 0.000135 ***
## employment_statusMedical Leave:monthly_premium_auto   1.199 0.230527
## employment_statusRetired:monthly_premium_auto        -4.198 2.72e-05 ***
## employment_statusUnemployed:monthly_premium_auto      5.649 1.67e-08 ***
## location_codeSuburban:monthly_premium_auto           33.337  < 2e-16 ***
## location_codeUrban:monthly_premium_auto              15.285  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 124.5 on 7764 degrees of freedom
## Multiple R-squared:  0.8129, Adjusted R-squared:  0.8125
## F-statistic:  1776 on 19 and 7764 DF,  p-value: < 2.2e-16
```

Impute income variable

```
df_big <- read.csv("data/train.csv", stringsAsFactors = TRUE)
df_big <- df_big %>% select(-Country, -Customer)
df_big$Income[df_big$Income == 0] <- NA

imputations <- complete(mice(df_big, method = "pmm", seed=1))
```

```
## Warning: Number of logged events: 26
```

```
df$income <- imputations$Income
```

```
head(df)
```

```
##   employment_status income location_code monthly_premium_auto
## 1          Employed  56274      Suburban                   69
## 2        Unemployed  20325      Suburban                   94
## 3          Employed  48767      Suburban                  108
```

```
## 4        Unemployed  17723      Suburban                  106
## 5          Employed  43836         Rural                   73
## 6          Employed  62902         Rural                   69
##   total_claim_amount vehicle_class
## 1           384.8111  Two-Door Car
## 2          1131.4649 Four-Door Car
## 3           566.4722  Two-Door Car
## 4           529.8813           SUV
## 5           138.1309 Four-Door Car
## 6           159.3830  Two-Door Car
```

Look at linear regression for baseline

```r
res <- lm(total_claim_amount ~ employment_status + location_code +
             vehicle_class + income + monthly_premium_auto +
             employment_status*monthly_premium_auto +
             location_code*monthly_premium_auto, data = df)

summary(res)
```

```
## 
## Call:
## lm(formula = total_claim_amount ~ employment_status + location_code + 
##     vehicle_class + income + monthly_premium_auto + employment_status * 
##     monthly_premium_auto + location_code * monthly_premium_auto, 
##     data = df)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -548.18  -62.94  -28.02   55.71 1411.93 
## 
## Coefficients:
##                                                   Estimate Std. Error
## (Intercept)                                     -5.927e+01  2.290e+01
## employment_statusEmployed                        6.971e+01  2.057e+01
## employment_statusMedical Leave                  -6.445e+00  2.851e+01
## employment_statusRetired                         1.124e+02  3.149e+01
## employment_statusUnemployed                     -1.143e+01  2.127e+01
## location_codeSuburban                            1.945e+01  1.174e+01
## location_codeUrban                               8.446e+00  1.466e+01
## vehicle_classLuxury Car                          3.854e+01  1.591e+01
## vehicle_classLuxury SUV                          2.378e+01  1.524e+01
## vehicle_classSports Car                         -5.370e+00  7.452e+00
## vehicle_classSUV                                -2.232e+00  5.266e+00
## vehicle_classTwo-Door Car                        1.661e+00  3.685e+00
## income                                          -9.665e-05  8.128e-05
## monthly_premium_auto                             1.889e+00  2.372e-01
## employment_statusEmployed:monthly_premium_auto  -7.824e-01  2.049e-01
## employment_statusMedical Leave:monthly_premium_auto  3.499e-01  2.926e-01
## employment_statusRetired:monthly_premium_auto   -1.359e+00  3.237e-01
## employment_statusUnemployed:monthly_premium_auto  1.210e+00  2.141e-01
## location_codeSuburban:monthly_premium_auto       4.064e+00  1.219e-01
## location_codeUrban:monthly_premium_auto          2.367e+00  1.548e-01
##                                                 t value Pr(>|t|)
```

```
## (Intercept)                                          -2.588 0.009675 **
## employment_statusEmployed                             3.389 0.000704 ***
## employment_statusMedical Leave                       -0.226 0.821158
## employment_statusRetired                              3.571 0.000358 ***
## employment_statusUnemployed                          -0.538 0.590877
## location_codeSuburban                                 1.656 0.097726 .
## location_codeUrban                                    0.576 0.564584
## vehicle_classLuxury Car                               2.423 0.015425 *
## vehicle_classLuxury SUV                               1.561 0.118629
## vehicle_classSports Car                              -0.721 0.471182
## vehicle_classSUV                                     -0.424 0.671777
## vehicle_classTwo-Door Car                             0.451 0.652116
## income                                               -1.189 0.234440
## monthly_premium_auto                                  7.963 1.92e-15 ***
## employment_statusEmployed:monthly_premium_auto       -3.819 0.000135 ***
## employment_statusMedical Leave:monthly_premium_auto   1.196 0.231792
## employment_statusRetired:monthly_premium_auto        -4.198 2.73e-05 ***
## employment_statusUnemployed:monthly_premium_auto      5.649 1.67e-08 ***
## location_codeSuburban:monthly_premium_auto           33.347  < 2e-16 ***
## location_codeUrban:monthly_premium_auto              15.289  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 124.5 on 7764 degrees of freedom
## Multiple R-squared:  0.8129, Adjusted R-squared:  0.8125
## F-statistic:  1776 on 19 and 7764 DF,  p-value: < 2.2e-16
```

Exactly the same performance, great.

GAM

Does using a GAM improve performance:

```r
res <- gam(total_claim_amount ~ employment_status + location_code +
             vehicle_class + s(income) +
             s(monthly_premium_auto, by = employment_status) +
             s(monthly_premium_auto, by = location_code), data = df)

summary(res)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## total_claim_amount ~ employment_status + location_code + vehicle_class +
##     s(income) + s(monthly_premium_auto, by = employment_status) +
##     s(monthly_premium_auto, by = location_code)
##
## Parametric coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       116.989      8.308  14.082  < 2e-16 ***
## employment_statusEmployed          -4.566      8.428  -0.542  0.58799
## employment_statusMedical Leave     25.506      9.449   2.699  0.00696 **
## employment_statusRetired          -15.802     10.587  -1.493  0.13560
```

```
## employment_statusUnemployed     100.847      7.324  13.769  < 2e-16 ***
## location_codeSuburban           395.907      3.984  99.380  < 2e-16 ***
## location_codeUrban              228.393      4.650  49.113  < 2e-16 ***
## vehicle_classLuxury Car           8.303     19.458   0.427  0.66961
## vehicle_classLuxury SUV         -24.152     19.409  -1.244  0.21341
## vehicle_classSports Car          -5.686      7.910  -0.719  0.47224
## vehicle_classSUV                 -1.174      5.858  -0.200  0.84117
## vehicle_classTwo-Door Car         1.318      3.650   0.361  0.71807
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                       edf Ref.df     F
## s(income)                                           7.572  8.488 4.088
## s(monthly_premium_auto):employment_statusDisabled      4.530  5.395 2.656
## s(monthly_premium_auto):employment_statusEmployed      0.875  0.875 2.189
## s(monthly_premium_auto):employment_statusMedical Leave 4.146  4.868 5.605
## s(monthly_premium_auto):employment_statusRetired       0.875  0.875 0.664
## s(monthly_premium_auto):employment_statusUnemployed    8.814  8.870 6.874
## s(monthly_premium_auto):location_codeRural             8.491  8.793 2.355
## s(monthly_premium_auto):location_codeSuburban          8.248  8.728 3.946
## s(monthly_premium_auto):location_codeUrban             0.875  0.875 5.008
##                                                      p-value
## s(income)                                           6.40e-05 ***
## s(monthly_premium_auto):employment_statusDisabled        0.0262 *
## s(monthly_premium_auto):employment_statusEmployed        0.1664
## s(monthly_premium_auto):employment_statusMedical Leave 5.29e-05 ***
## s(monthly_premium_auto):employment_statusRetired         0.4461
## s(monthly_premium_auto):employment_statusUnemployed     < 2e-16 ***
## s(monthly_premium_auto):location_codeRural               0.0144 *
## s(monthly_premium_auto):location_codeSuburban          3.39e-05 ***
## s(monthly_premium_auto):location_codeUrban               0.0364 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 92/93
## R-sq.(adj) =  0.816   Deviance explained = 81.8%
## GCV =  15297  Scale est. = 15186     n = 7784
```

Hmmm seems to do about the same, but I don't really know what I am doing. Perhaps this is due to the fact that it is mostly linear relationships (from EDA) so these result could be reasonable.

Things to do

See if we can get GAMs to work better. Calculate cross-validation absolute error for each model. Consider robust regression or quantreg as we are interested in absolute error, rather than the mean square error. Median regression is a thing. This will probably work better for absolute error?