EDA

Imports

```
library(tidyverse)
library(lubridate)
library(GGally)
library(randomForest)
```

Introduction

Exploratory data analysis for LV Datathon project.

Read in data

```
df <- read.csv("data/train.csv")

# Make column names consistent
cols <- colnames(df) %>%
  str_replace_all(fixed("."), "_")  %>%
  tolower()

cols[cols == "employmentstatus"] <- "employment_status"
colnames(df) <- cols

# Make variables with low number of unique values factors
df <- df %>%
  mutate(number_of_open_complaints = as.factor(number_of_open_complaints),
         number_of_policies = as.factor(number_of_policies))

orig_data <- df
```

Are there missing values?

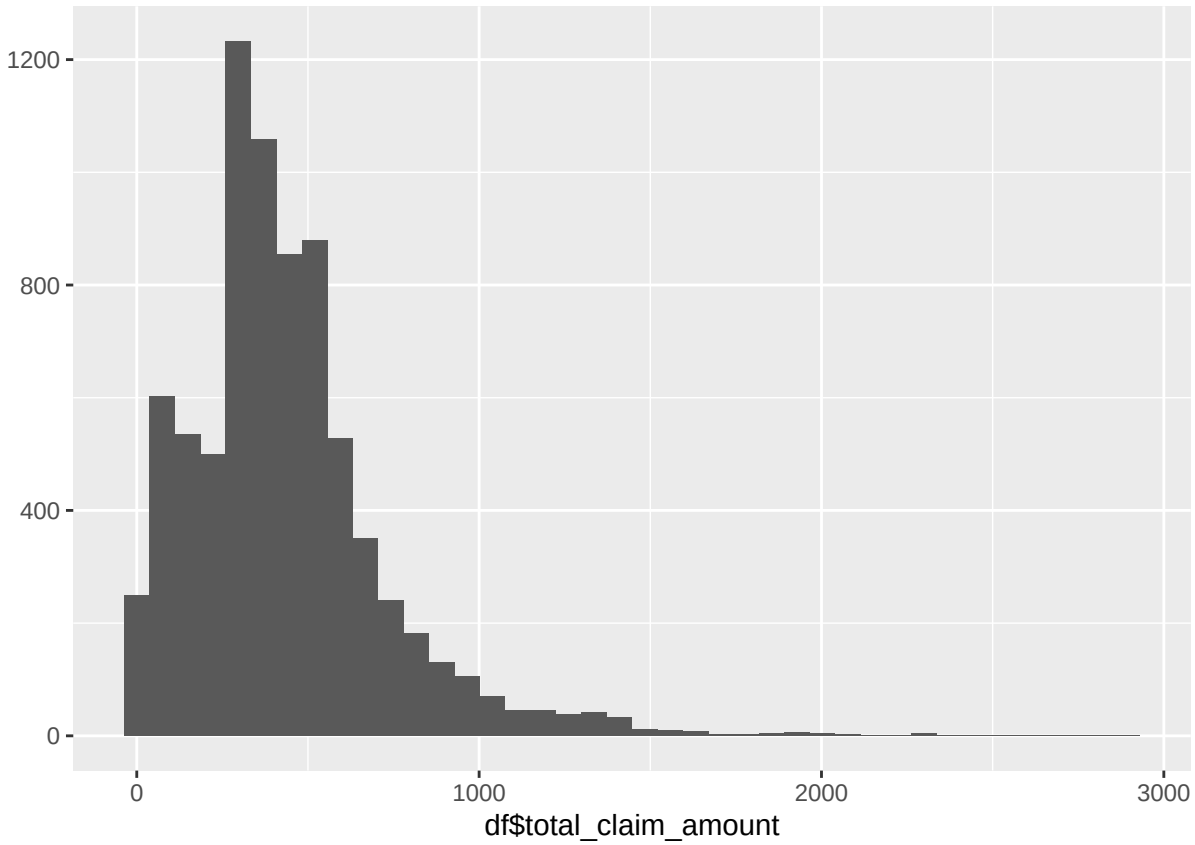
```
table(is.na(df))
```

```
##
##  FALSE
## 194600
```

Target variable

Target variable is `total_claim_amount`:

```
qplot(df$total_claim_amount, geom = "histogram", bins = 40)
```



Are there zero values?

```
table(df$total_claim_amount == 0)
```

```
##
## FALSE
##  7784
```

No but there is strangely small amounts. If this is pounds and these aren't mistakes then these are some real petty people:

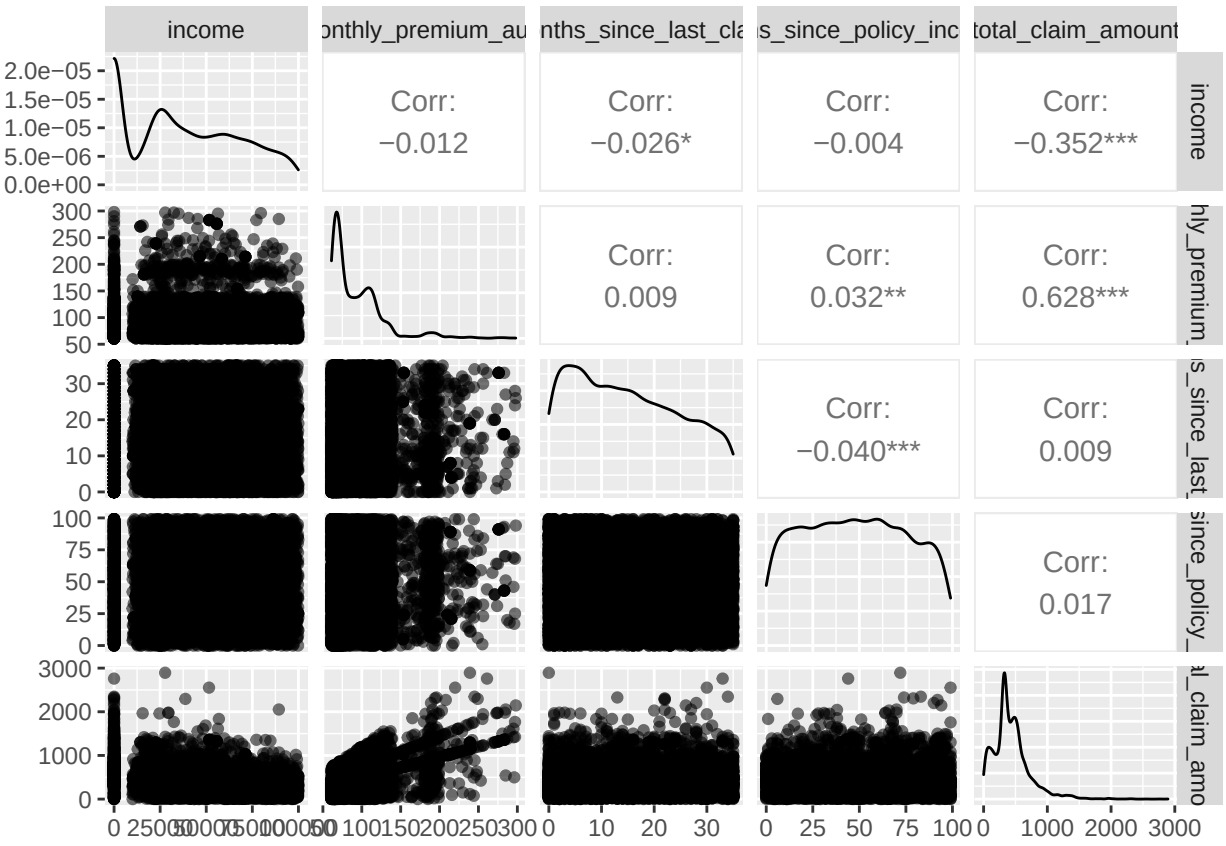
```
sum(df$total_claim_amount < 1)
```

```
## [1] 6
```

Predictor variables

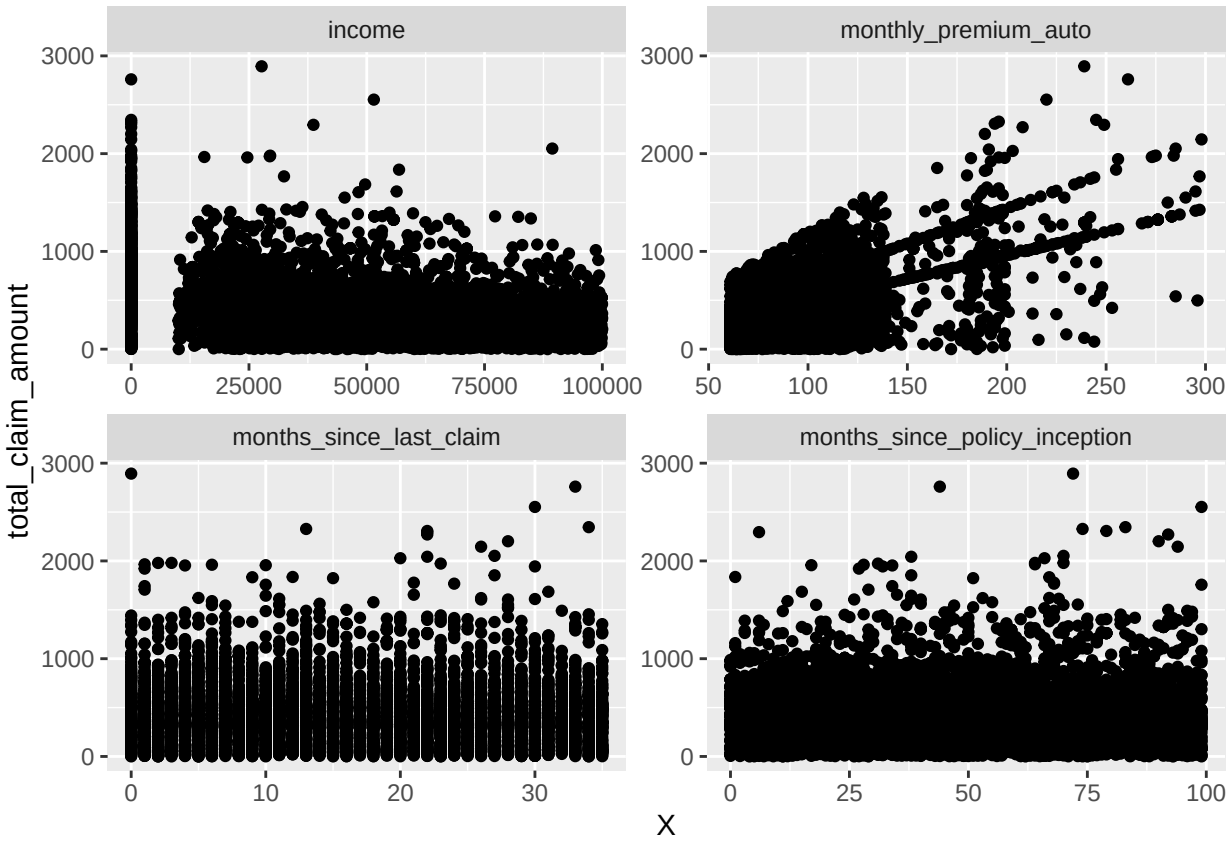
Continuous variables

```
df %>%
  select_if(is.numeric) %>%
  ggpairs(aes(alpha = 0.2))
```



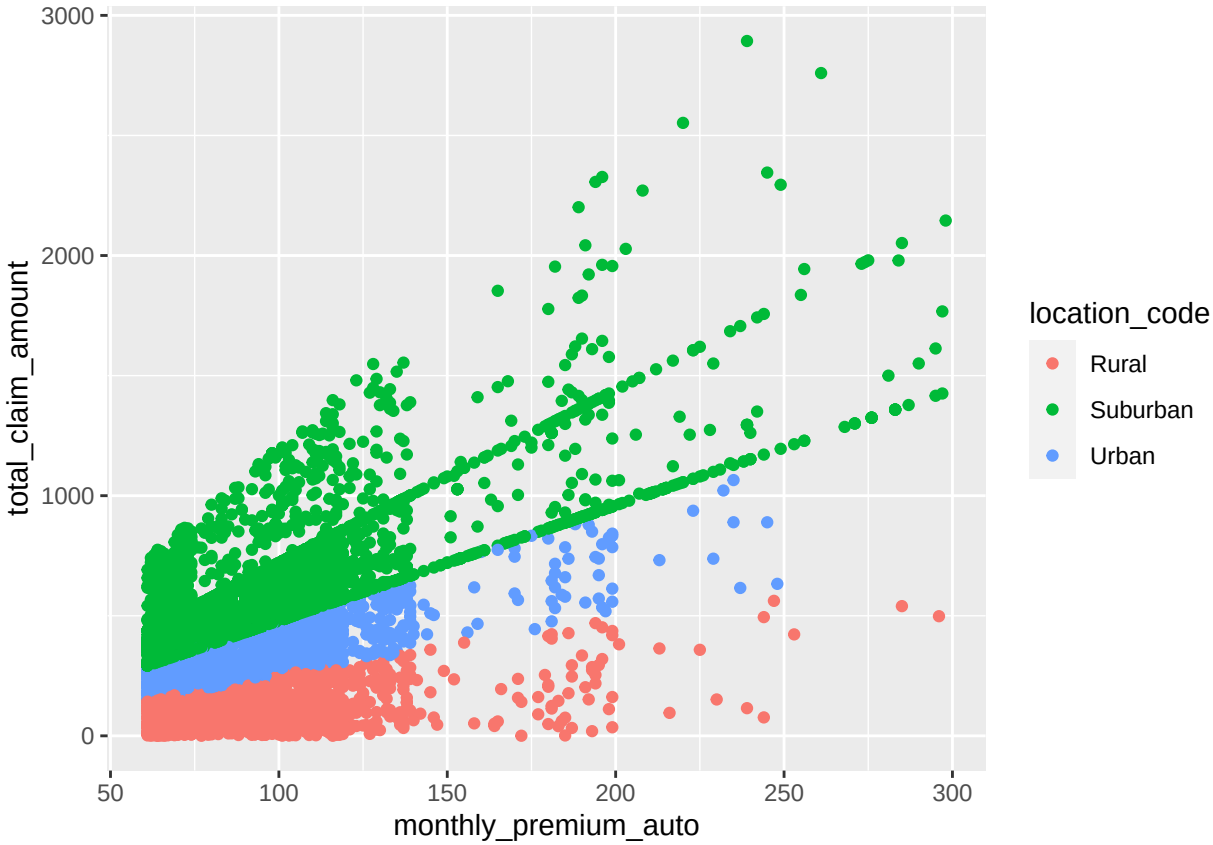
In terms of the correlation only two of the continuous variables seem to have a strong association with the `total_claim_amount`. These are `income` and `monthly_premium_auto`.

```
df %>%
  select_if(is.numeric) %>%
  pivot_longer(-total_claim_amount, names_to = "feature", values_to = "X") %>%
  ggplot(aes(x = X, y = total_claim_amount)) +
  geom_point() +
  facet_wrap(~feature, scales = "free")
```



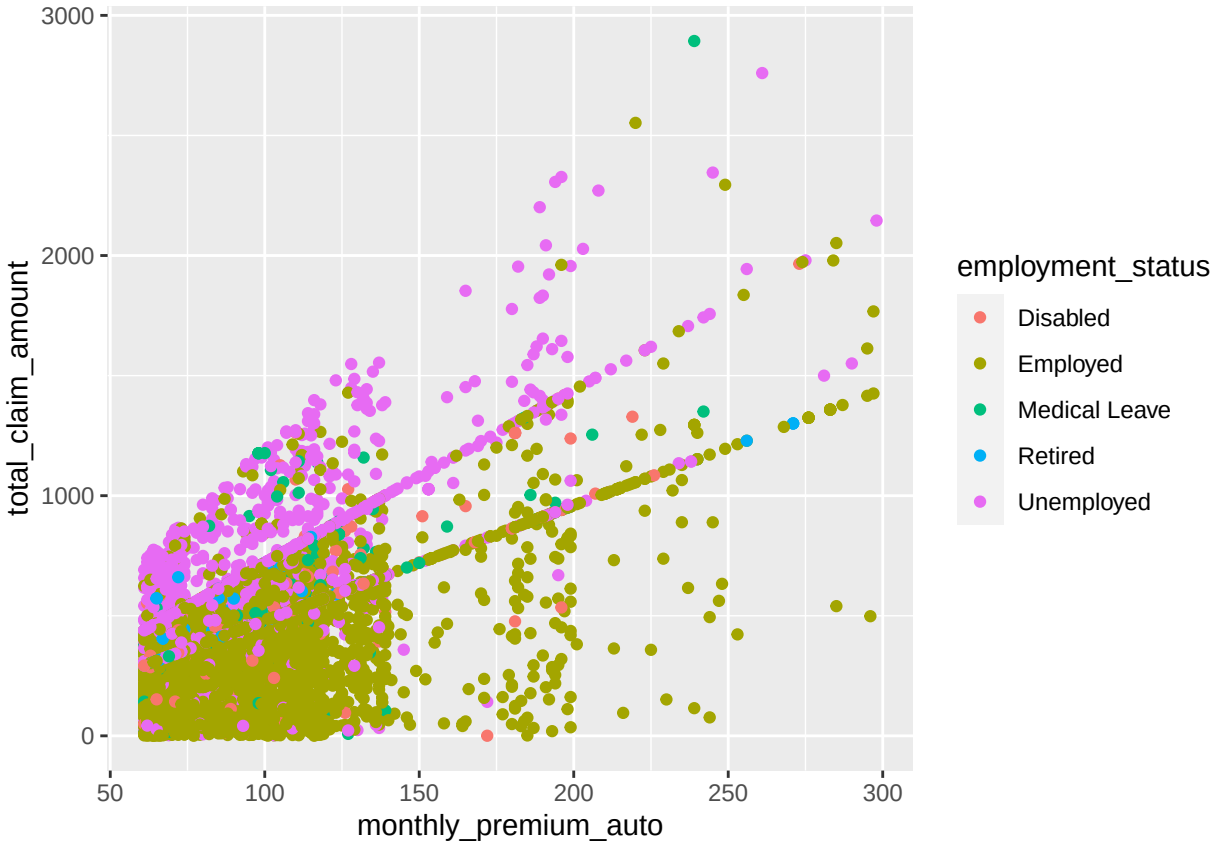
`months_since_last_claim` and `months_since_policy_inception` don't seem like they will be particularly useful features. I will drop them for now. We can see that having no income seems to be important, this is captured by the `employment_status` variable.

```
df %>%
  ggplot(aes(x = monthly_premium_auto, y = total_claim_amount,
             colour = location_code)) +
  geom_point()
```



I'm not sure exactly what is causing the weird structure in the plot above, but `location_code` separates it out extremely well. The top "segment" can be separated out using employment status.

```
df %>%
  ggplot(aes(x = monthly_premium_auto, y = total_claim_amount,
             colour = employment_status)) +
  geom_point()
```



Categorical variables

```
df %>%
  select_if(is.character) %>%
  apply(2, {function(x) length(unique(x))})
```

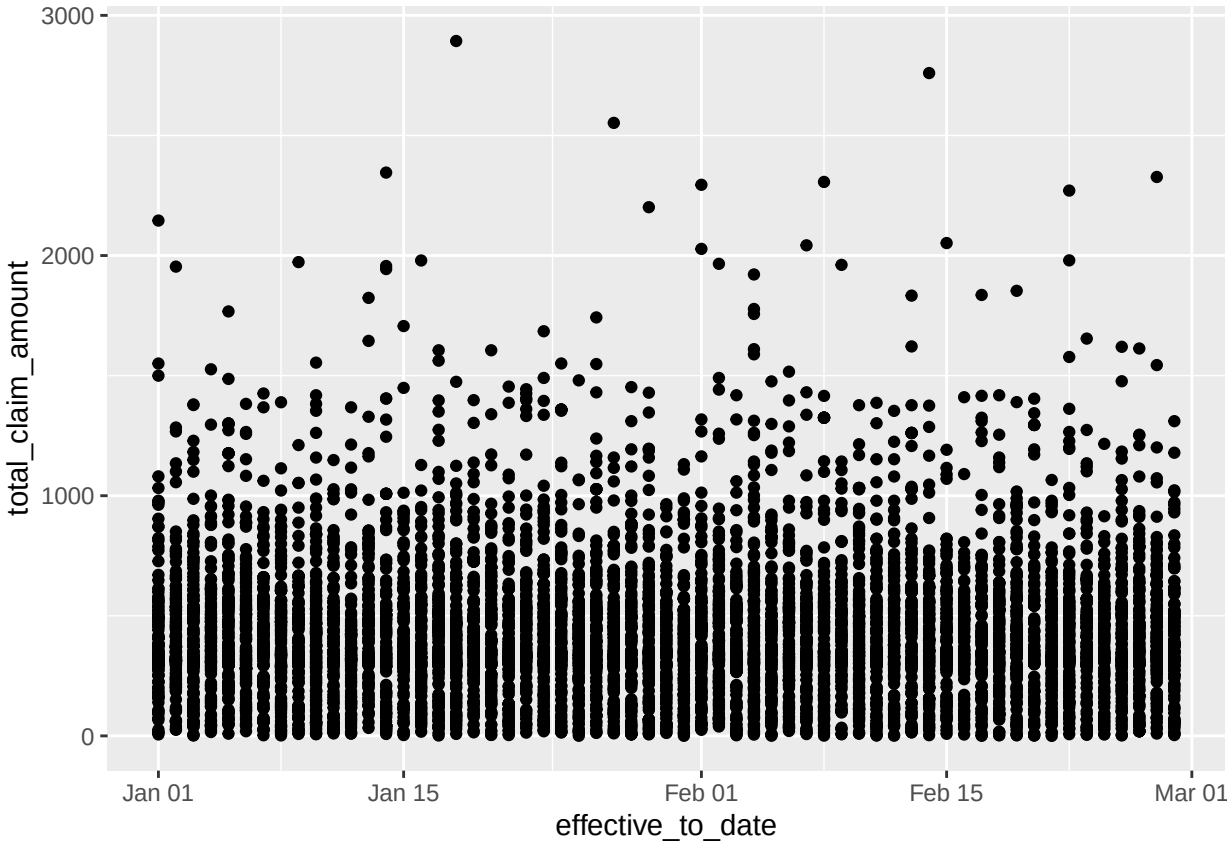
```
##          customer           country        state_code             state
##              7784                 1                 5                 5
##          response          coverage         education effective_to_date
##                 2                 3                 5                59
## employment_status            gender     location_code    marital_status
##                 5                 2                 3                 3
##       policy_type            policy      claim_reason     sales_channel
##                 3                 9                 4                 4
##     vehicle_class      vehicle_size
##                 6                 3
```

Customer is all unique so likely not useful. Country is all US so useless.

Look at effective to date see if that could be useful:

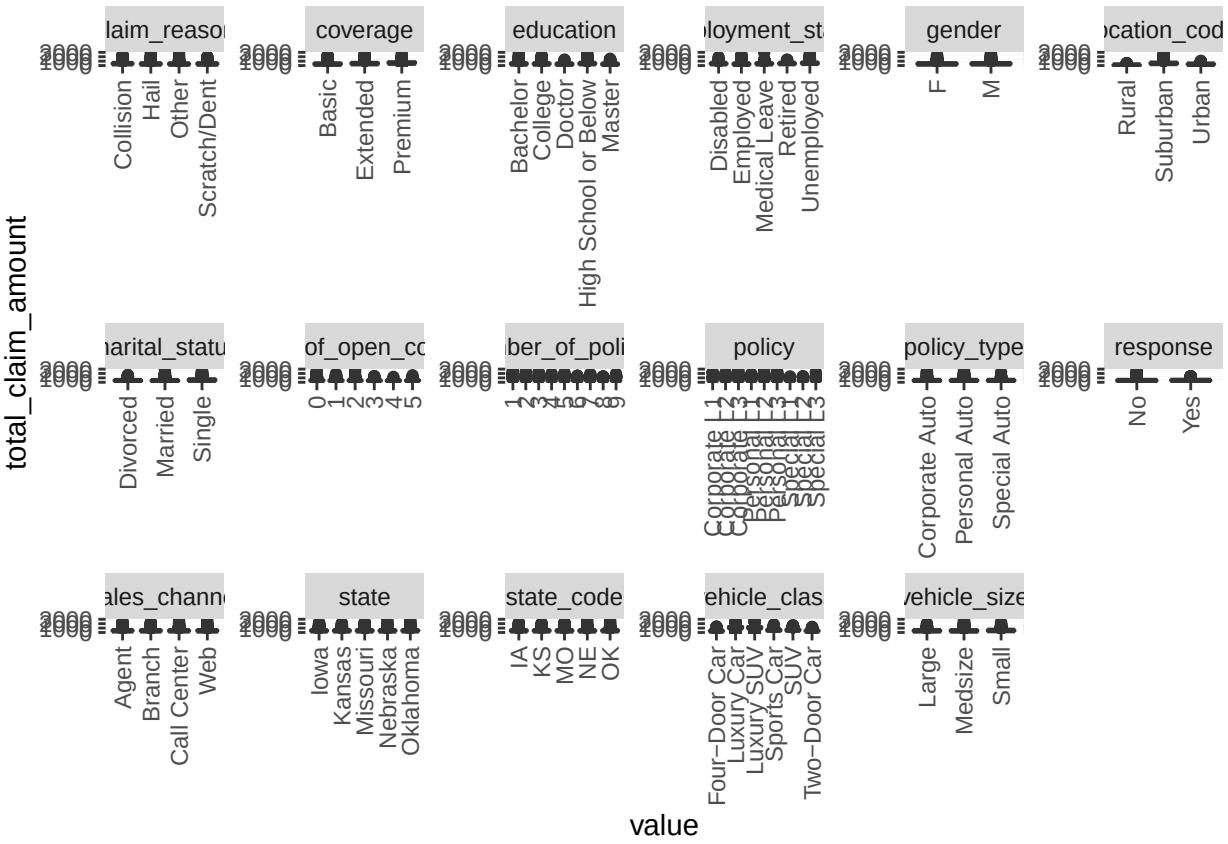
```
df$effective_to_date <- as.Date(mdy(df$effective_to_date))

df %>%
  ggplot(aes(x=effective_to_date, y=total_claim_amount)) +
  geom_point()
```



Seems it is likely not informative.

```
df %>%
  mutate_if(is.character, as.factor) %>%
  select(-customer, -country, -effective_to_date,
         -months_since_last_claim, months_since_policy_inception) %>%
  select_if(function(col) is.factor(col) | all(col == .$total_claim_amount)) %>%
  pivot_longer(-total_claim_amount) %>%
  ggplot(aes(x = value, y = total_claim_amount)) +
  geom_boxplot() +
  facet_wrap(~name, scales = "free", nrow = 3) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
# Bit cramped so will save it larger
ggsave("plots/boxplot.png", width = 10, height = 10)
```

Feature selection

From this we can remove features that don't seem informative:

```
# These ones look ok to drop (if aiming for a simple model)
df <- df %>%
  select(-sales_channel, -state, -state_code, -vehicle_size, -months_since_policy_inception,
         -months_since_last_claim, -customer, -country, -effective_to_date)

# These ones are perhaps a bit more dubious. But for a simple first model might
# be worth dropping
df <- df %>%
  select(-claim_reason, -education, -number_of_policies, -policy, -policy_type,
         -response, -gender, -number_of_open_complaints, -marital_status)

features_kept <- colnames(df)
```

Write out this simplified data set:

```
write_csv(df, "data/train_simple.csv")
```

Importances

Look at feature importances from random forest just to check we have done reasonable feature selection. This could help find for example features that are useful in a more complicated way (e.g. through interactions

with other variables), that may not be obvious from the plots above.

```
df <- orig_data

# Remove obviously useless ones
df <- df %>%
  select(-state, -country, -effective_to_date, -customer) %>%
  mutate_if(is.character, as.factor)

df %>%
  select_if(is.factor) %>%
  apply(2, function(col) length(unique(col)))
```

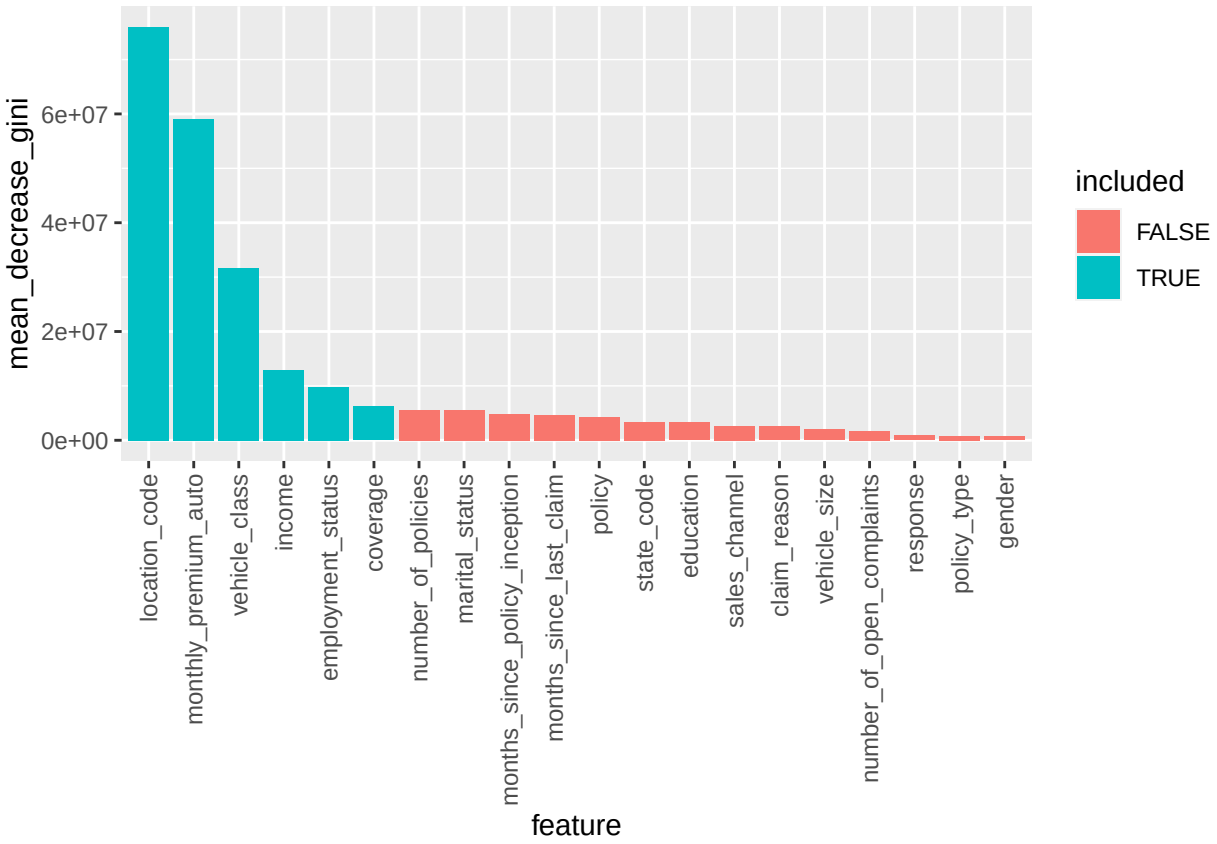
```
##                state_code                  response                  coverage
##                         5                         2                         3
##                 education         employment_status                    gender
##                         5                         5                         2
##             location_code            marital_status number_of_open_complaints
##                         3                         3                         6
##        number_of_policies               policy_type                    policy
##                         9                         3                         9
##              claim_reason             sales_channel             vehicle_class
##                         4                         4                         6
##              vehicle_size
##                         3
```

```
# Subset for speed
rf <- randomForest(total_claim_amount ~ ., data = df[sample(1:nrow(df), 3000), ])

imp <- tibble(feature = row.names(rf$importance),
              mean_decrease_gini = as.vector(rf$importance))

# colour by included
imp$included <- imp$feature %in% features_kept

imp %>%
  arrange(desc(mean_decrease_gini)) %>%
  mutate(feature = factor(feature, levels = .$feature)) %>%
  ggplot(aes(x = feature, y = mean_decrease_gini, fill = included)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Likely that random forest finds income useful as it can use it to determine employment status anyway. Could be worth imputing predicted salaries instead of 0s in unemployed people.