# AE 7: Exam 2 Review

Notes

## Packages

```
library(tidyverse)
library(tidymodels)
library(knitr)
library(openintro)

# fix data!
loans_full_schema <- droplevels(loans_full_schema)

knitr::opts_chunk$set(
  fig.asp = 0.618,
  out.width = "80%"
)
```

## Goal

Create a model for predicting `interest_rate`.

## View data

Note the dimensions of the data and the variable names. Review the data dictionary.

```
dim(loans_full_schema)
```

```
[1] 10000    55
```

```
num_rows <- nrow(loans_full_schema)
```

The full dataset consists of 10000 rows (observations) and 55 columns (variables).

### Split data into training and testing

Split your data into testing and training sets.

```
set.seed(2345)
loans_split    <- initial_split(loans_full_schema)
loans_training <- training(loans_split)
loans_testing  <- testing(loans_split)
```

### Write the model

Write the model for predicting interest rate (`interest_rate`) from debt to income ratio (`debt_to_income`), the term of loan (`term`), the number of inquiries (credit checks) into the applicant's credit during the last 12 months (`inquiries_last_12m`), whether there are any bankruptcies listed in the public record for this applicant (`bankrupt`), and the type of application (`application_type`). The model should allow for the effect of debt to income ratio on interest rate to vary by application type.

$$\begin{aligned}\widehat{interest\_rate} = \beta_0 &+ \beta_1 \times debt\_to\_income \\ &+ \beta_2 \times term \\ &+ \beta_3 \times inquiries\_last\_12m \\ &+ \beta_4 \times bankrupt \\ &+ \beta_5 \times application\_type \\ &+ \beta_6 \times debt\_to\_income * application\_type\end{aligned}$$

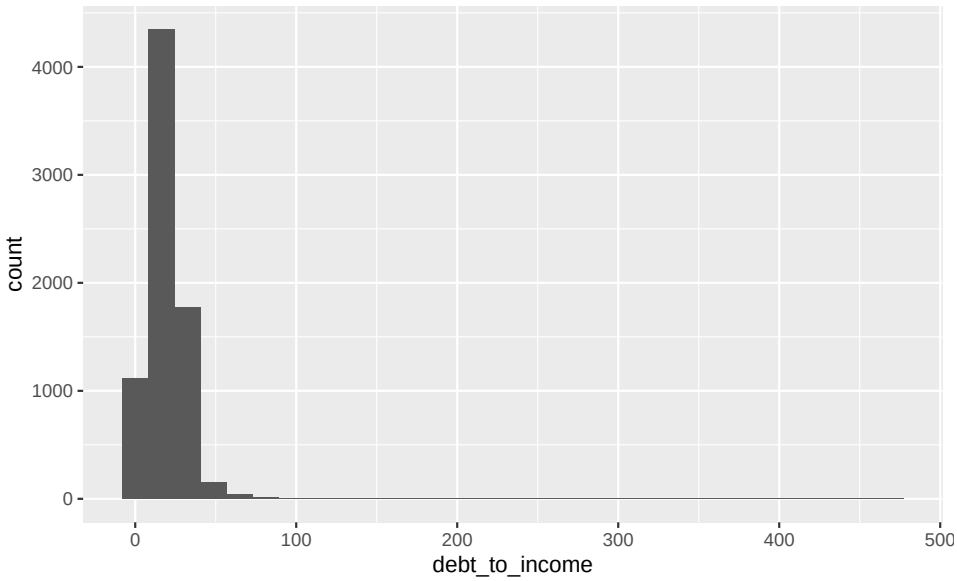
### Exploration

Explore characteristics of the variables you'll use for the model using the training data only.

```
ggplot(loans_training, aes(x = debt_to_income)) +
  geom_histogram()
```

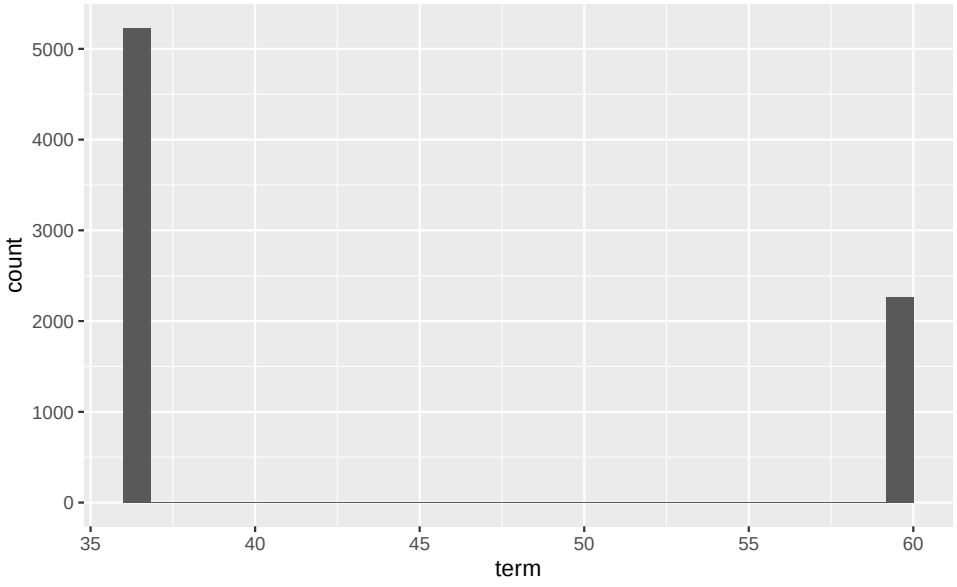
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
Warning: Removed 15 rows containing non-finite values (stat_bin).
```



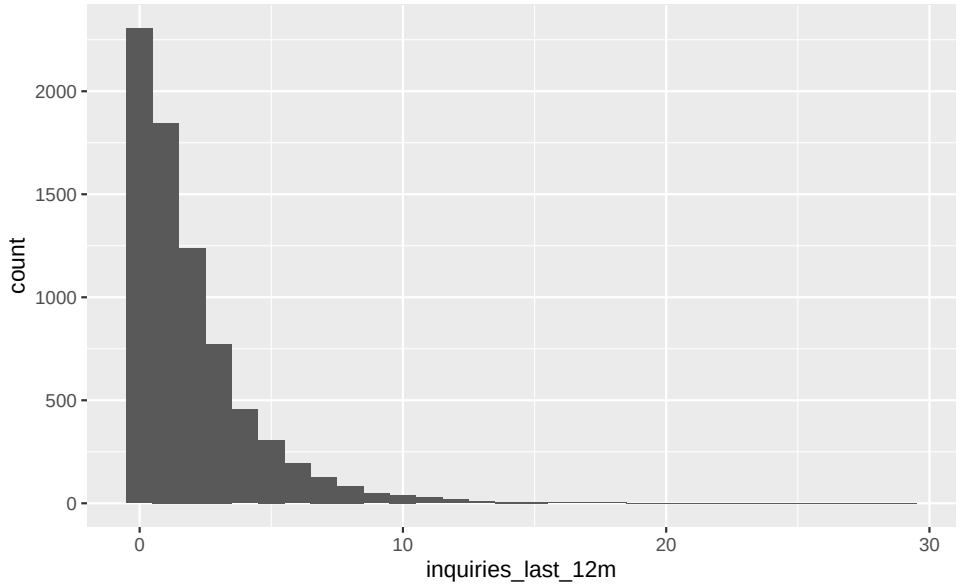

```
ggplot(loans_training, aes(x = term)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(loans_training, aes(x = inquiries_last_12m)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
loans_training %>%
  count(application_type)
```

```
# A tibble: 2 x 2
  application_type     n
  <fct>            <int>
1 individual        6384
2 joint             1116
```

## Specify model

Specify a linear regression model.

```
loans_spec <- linear_reg() %>%
  set_engine("lm")
```

### Create recipe

- Predict `interest_rate` from `debt_to_income`, `term`, `inquiries_last_12m`, `public_record_bankrupt`, and `application_type`.
- Mean center `debt_to_income`.
- Make `term` a factor.
- Create a new variable: `bankrupt` that takes on the value "no" if `public_record_bankrupt` is 0 and the value "yes" if `public_record_bankrupt` is 1 or higher. Then, remove `public_record_bankrupt`.
- Interact `application_type` with `debt_to_income`.
- Create dummy variables where needed and drop any zero variance variables.

```
loans_rec <- recipe(interest_rate ~ debt_to_income +
                      term + inquiries_last_12m +
                      public_record_bankrupt + application_type,
                    data = loans_training) %>%
  step_center(debt_to_income) %>%
  step_mutate(term = as_factor(term)) %>%
  step_mutate(bankrupt = as_factor(if_else(public_record_bankrupt == 0, "no", "yes"))) %>%
  step_rm(public_record_bankrupt) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~ starts_with("application_type"):debt_to_income) %>%
  step_zv(all_predictors())
```

### Create workflow

Create the workflow that brings together the model specification and recipe.

```
loans_wflow <- workflow() %>%
  add_model(loans_spec) %>%
  add_recipe(loans_rec)
```

### Cross validation

Conduct 10-fold cross validation.

```
set.seed(345)
loans_folds <- vfold_cv(loans_training, v = 10)

loans_fit_rs <- loans_wflow %>%
```

```
  fit_resamples(loans_folds)

loans_fit_rs
```

```
# Resampling results
# 10-fold cross-validation
# A tibble: 10 x 4
   splits             id     .metrics          .notes
   <list>             <chr>  <list>            <list>
 1 <split [6750/750]> Fold01 <tibble [2 x 4]> <tibble [0 x 1]>
 2 <split [6750/750]> Fold02 <tibble [2 x 4]> <tibble [0 x 1]>
 3 <split [6750/750]> Fold03 <tibble [2 x 4]> <tibble [0 x 1]>
 4 <split [6750/750]> Fold04 <tibble [2 x 4]> <tibble [0 x 1]>
 5 <split [6750/750]> Fold05 <tibble [2 x 4]> <tibble [0 x 1]>
 6 <split [6750/750]> Fold06 <tibble [2 x 4]> <tibble [0 x 1]>
 7 <split [6750/750]> Fold07 <tibble [2 x 4]> <tibble [0 x 1]>
 8 <split [6750/750]> Fold08 <tibble [2 x 4]> <tibble [0 x 1]>
 9 <split [6750/750]> Fold09 <tibble [2 x 4]> <tibble [0 x 1]>
10 <split [6750/750]> Fold10 <tibble [2 x 4]> <tibble [0 x 1]>
```

### Summarize CV metrics

Summarize metrics from your CV resamples.

```
collect_metrics(loans_fit_rs)
```

```
# A tibble: 2 x 6
  .metric .estimator  mean     n std_err .config
  <chr>   <chr>      <dbl> <int>   <dbl> <chr>
1 rmse    standard   4.55     10  0.0398 Preprocessor1_Model1
2 rsq     standard   0.176    10  0.0124 Preprocessor1_Model1
```

Why are we focusing on R-squared and RMSE instead of adjusted R-squared, AIC, BIC?

- R-squared, AIC, and BIC all apply a penalty for additional predictors. It's important to consider this penalty when fitting and evaluating the model on the same dataset.
- When doing CV we fit the model on one part of the data (analysis) and calculate performance indicators on the other part (assessment). Using different data eliminates the need for penalization.

## Next steps...

Depending on time, either

- Create a workflow for another model with a new recipe (omitting the interaction variable), conduct CV, do model selection between these two, and then interpret the coefficients for the selected model.
- Or interpret the coefficients for the one model you fit.

Make sure to interpret the intercept and slope coefficient for at least one numerical, one categorical, and one interaction predictor.