# Generalised Linear Models Assessed Practical

P151, 1985 words

## Contents

# 1 Exploratory Data Analysis

The data set at hand comprises 1902 observations on 9 variables. Variables `hir, odir`, and `lvr` are ratios $\in [0, 1]$, which I rescale by a factor of 100, to simplify their interpretation. Table 1 displays summary statistics. No variables take negative values. About 88% of mortgage applications are approved, 11% of applicants are self-employed, 60% are not single, and only about 15% are non-white. The typical applicant seeks a mortgage for 74% of property value, but some applicants seek mortgages for close to double of the assessed property value. This may seem surprising, but is not implausible, as some applicants may wish to buy a property in need of renovation. The typical applicant spends about a third of their income on debt service. The data features two applications in which `hir`, ratio of monthly housing expenses to monthly income, exceeds 1, both of which were, perhaps unsurprisingly, denied. There is one observation (number 449) which features an `odir` of greater than 50% (51%), and a `hir` of 44%, for which the mortgage application was approved. This seems unusual, but is not impossible: the LTV ratio is 51%, and while the applicant may spend 95% of their regular income on debt service, we do not have any information on savings, investments, or other collateral that the mortgage officer may have considered. In conclusion, there are no observations that are prima facie implausible.

Table 1: Summary Statistics

| | Mean | Median | SD | Min | Max |
|---|---|---|---|---|---|
| hir | 25.53 | 26.00 | 7.95 | 1.00 | 110.00 |
| odir | 7.50 | 6.00 | 6.64 | 0.00 | 51.00 |
| lvr | 73.70 | 78.00 | 17.96 | 2.00 | 195.00 |
| uria | 3.76 | 3.20 | 2.01 | 1.80 | 10.60 |
| approved | 0.88 | 1.00 | 0.33 | 0.00 | 1.00 |
| mcs | 1.72 | 2.00 | 0.55 | 1.00 | 4.00 |
| self | 0.11 | 0.00 | 0.32 | 0.00 | 1.00 |
| single | 0.40 | 0.00 | 0.49 | 0.00 | 1.00 |
| white | 0.85 | 1.00 | 0.35 | 0.00 | 1.00 |

Figure 1: Continuous Predictors

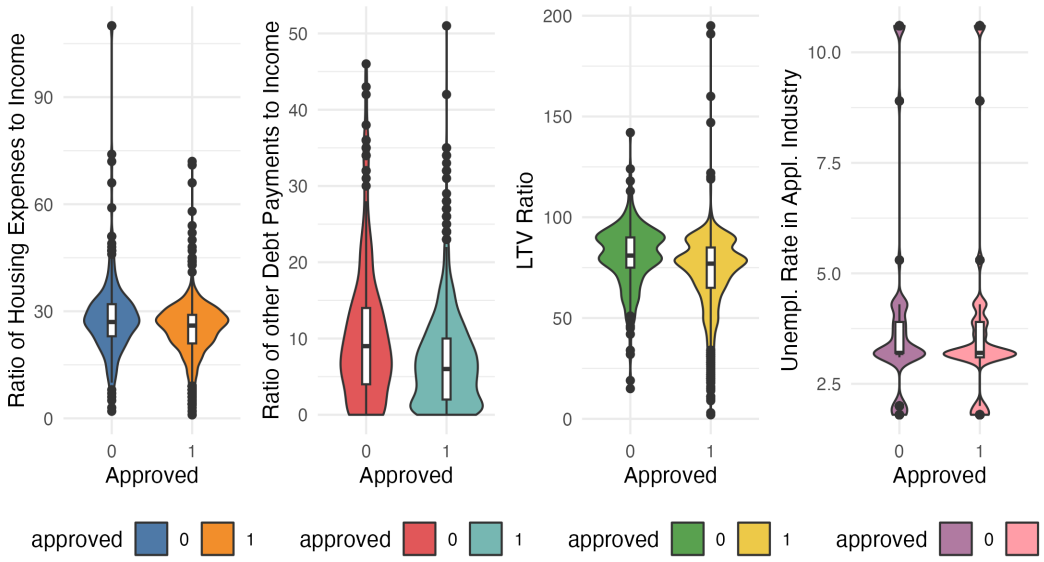



Figure 2: Continuous Predictors: Decile Means

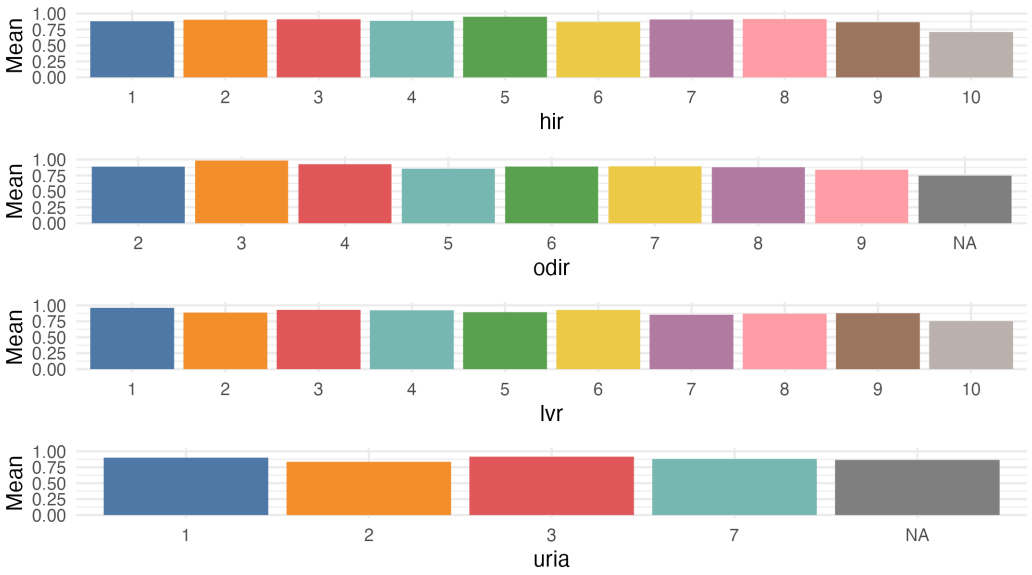



Figure 1 shows the distributions of the continuous predictors segmented by whether the application was approved. We can see that the there are no strong distributional shifts between successful and unsuccessful applications, but the means of the debt and LTV ratios are lower for successful applications, which makes economic sense. The unemployment rate in the applicant's industry is very similarly distributed between successful and unsuccessful applications. Figure 2 above shows that for each decile of the same continuous predictors, the mean approval rate is very similar, i.e., there are no large discrepancies in approval over their distributions.

Figure 3: Categorical Predictors: Absolute Proportions

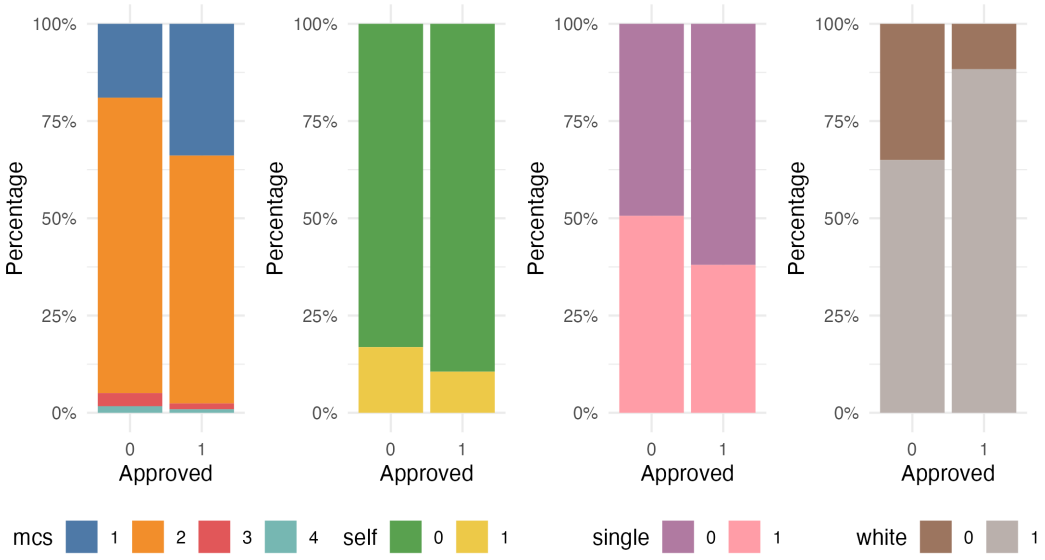
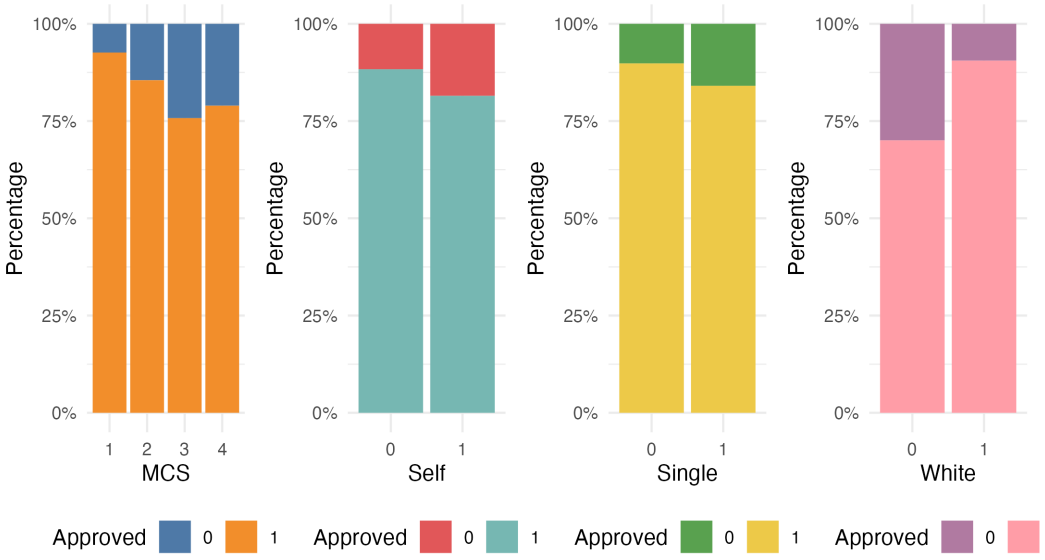



Figure 3 displays the proportions of the categorical predictors `mcs, self, single, white` segmented by approval status of the application. We can see that the overwhelming majority of applicants have a mortgage credit score of 2 of lower, with only 52 applicants (roughly 2.7%) with a credit score of 3 or 4. Similarly, only 216 applicants (11.3%) are self-employed. Figure 4 below shows the proportion of approved applications per group. We can see that even though applicants with a credit score of 3 and 4 are a small group, there is no strong discrepancy between the groups in terms of approvals, and the same holds for self-employed and single applicants. The only visible discrepancy is for nonwhite applicants, who see substantially fewer applications approved.

Figure 4: Categorical Predictors: Proportions of Approved Mortgages per Group



## 2 Baseline Model

We now begin to model the probability of a mortgage application being approved using logistic regression. Since approval is binary, `approved` $\sim Ber(\pi_i)$ with $\pi_i$ representing the probability of approval for the $i$-th mortgage application. The baseline logistic regression model used to estimate the probability of mortgage application approval is given by (1):

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_{const.} + \beta_{\text{hir}} \cdot \text{HIR}_i + \beta_{\text{odir}} \cdot \text{ODIR}_i + \beta_{\text{lvr}} \cdot \text{LVR}_i + \beta_{\text{mcs2}} \cdot \text{MCS2}_i + \beta_{\text{mcs3}} \cdot \text{MCS3}_i \tag{1}$$
$$+ \beta_{\text{mcs4}} \cdot \text{MCS4}_i + \beta_{\text{self1}} \cdot \text{SELF1}_i + \beta_{\text{single1}} \cdot \text{SINGLE1}_i + \beta_{\text{white1}} \cdot \text{WHITE1}_i + \beta_{\text{uria}} \cdot \text{URIA}_i$$

The model is fitted using a Bernoulli distribution with the canonical logit link function. Alternatively, we could choose to model `mcs` as a single, continuous predictor rather than as a categorical variable. In this case, (1) turns into:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_{const.} + \beta_{\text{hir}} \cdot \text{HIR}_i + \beta_{\text{odir}} \cdot \text{ODIR}_i + \beta_{\text{lvr}} \cdot \text{LVR}_i + \beta_{\text{mcs}} \cdot \text{MCS}_i + \beta_{\text{self1}} \cdot \text{SELF1}_i \tag{2}$$
$$+ \beta_{\text{single1}} \cdot \text{SINGLE1}_i + \beta_{\text{white1}} \cdot \text{WHITE1}_i + \beta_{\text{uria}} \cdot \text{URIA}_i$$

Table 2 below displays estimation results for both specifications. Using (2) results in a single, highly statistically significant estimated coefficient $\hat{\beta}_{MCS}$, and a marginally lower Akaike Information Criterion (AIC) than using (1). Using `mcs` in this way, however, comes with two main downsides. First, it imposes that the change in the probability of mortgage approval is constant when moving between credit score classes, that is, the change in $\pi_i$ is equal between an applicant whose credit deteriorates from rating 1 to rating 2, and an applicant whose credit deteriorates from rating 3 to rating 4. From Figure 4, this does not seem necessarily plausible, as we can see that the proportion of approved applications is actually higher for applicants with a credit score of 4 than for those with a score of 3. It also runs counter to economic logic: keeping all other predictors constant, changing an applicant's credit score from 3 to 4 might well make a larger difference than changing it from 1 to 2.

Second, Figure 3 shows that there are only 33 and 19 applicants, respectively (1.7% and 1%) with a credit score exceeding 2. Using `mcs` as a continuous variable, as in (2), means subsuming these small groups, which reasonably could have much worse odds of obtaining a mortgage, into one variable with much larger groups whose odds are a priori much higher. For these reasons, I choose to use `mcs` as a categorical variable moving forward.

Table 2: Estimation Results for Baseline Model

| | MCS as factor | MCS as numerical |
|---|---|---|
| | (1) | (2) |
| hir | −0.050*** (0.010) | −5.052*** (1.006) |
| odir | −0.074*** (0.011) | −7.381*** (1.080) |
| lvr | −0.021*** (0.005) | −2.094*** (0.476) |
| mcs2 | −0.532*** (0.195) | |
| mcs3 | −0.923* (0.498) | |
| mcs4 | −1.159* (0.616) | |
| mcs | | −0.457*** (0.141) |
| self1 | −0.704*** (0.217) | −0.691*** (0.216) |
| single1 | −0.330** (0.152) | −0.340** (0.151) |
| white1 | 1.219*** (0.170) | 1.227*** (0.169) |
| uria | −0.063* (0.035) | −0.062* (0.034) |
| Constant | 5.436*** (0.558) | 5.864*** (0.593) |
| Observations | 1,902 | 1,902 |
| Log Likelihood | -611.887 | -612.065 |
| Akaike Inf. Crit. | 1,245.773 | 1,242.130 |
| *Note:* | | *p<0.1; **p<0.05; ***p<0.01 |
| | | SE in Parentheses |

## 3 Model Selection

We can now try to improve upon this baseline model. Using stepwise selection on both AIC and BIC, we can see AIC deteriorates when dropping predictors in specification (1) and stepwise selection hence results in an identical specification. BIC, however, yields in a more parsimonious specification given by

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_{const.} + \beta_{\text{hir}} \cdot \text{HIR}_i + \beta_{\text{odir}} \cdot \text{ODIR}_i + \beta_{\text{lvr}} \cdot \text{LVR}_i + \beta_{\text{self1}} \cdot \text{SELF1}_i + \beta_{\text{white1}} \cdot \text{WHITE1}_i \tag{3}$$

. In other words, this specification drops `mcs, single`, and `uria`. Table 3 below shows the number of parameters, AIC, BIC, and $R^2_{KL}$ for (1) and (3). While BIC unsurprisingly improves by moving from (1) to (3), both AIC and $R^2_{KL}$ deteriorate. There hence does not appear to be anything gained by dropping predictors from our additive baseline.

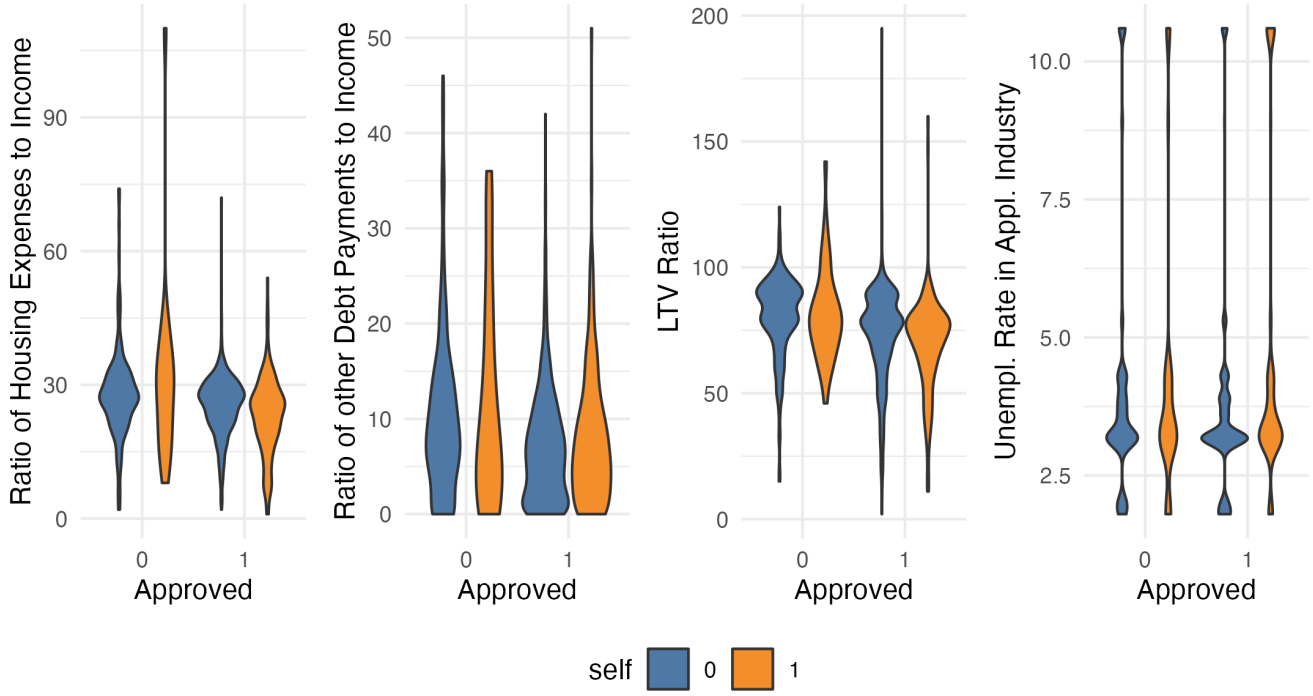
Table 3: Model Selection: Additive Models

| | p | AIC | BIC | $R^2_{KL}$ |
|---|---|---|---|---|
| Baseline | 11.000 | 1245.773 | 1306.831 | 0.144 |
| stepwise AIC on baseline | 11.000 | 1245.773 | 1306.831 | 0.144 |
| stepwise BIC on baseline | 6.000 | 1256.842 | 1290.146 | 0.130 |

We therefore turn to interaction specifications. Figure 5 below displays the distributions of the continuous predictors `hir, odir, lvr, uria` segmented not just by approval status (as in Figure 1), but additionally by self-employment status. We can see that the distribution of the unemployment rate for self-employed applicants is

more even across approval statuses. The same is true for the other three continuous predictors. Interestingly, the mass of the distribution of other debt (`odir`) is concentrated at lower levels for self-employed applicants.

Figure 5: Continuous Predictors segmented by Approved and Self-Employment




To incorporate the role of self-employment into our baseline, we fit a specification that contains all possible interaction terms involving `self`:

$$\begin{aligned}\log\left(\frac{\pi_i}{1-\pi_i}\right) = {} & \beta_0 + \beta_{\text{self1}} \cdot \text{SELF1}_i + \beta_{\text{hir}} \cdot \text{HIR}_i + \beta_{\text{odir}} \cdot \text{ODIR}_i + \beta_{\text{lvr}} \cdot \text{LVR}_i \\ & + \beta_{\text{mcs2}} \cdot \text{MCS2}_i + \beta_{\text{mcs3}} \cdot \text{MCS3}_i + \beta_{\text{mcs4}} \cdot \text{MCS4}_i + \beta_{\text{single1}} \cdot \text{SINGLE1}_i \\ & + \beta_{\text{white1}} \cdot \text{WHITE1}_i + \beta_{\text{uria}} \cdot \text{URIA}_i \\ & + \beta_{\text{self1:hir}} \cdot (\text{SELF1}_i \times \text{HIR}_i) + \beta_{\text{self1:odir}} \cdot (\text{SELF1}_i \times \text{ODIR}_i) \\ & + \beta_{\text{self1:lvr}} \cdot (\text{SELF1}_i \times \text{LVR}_i) + \beta_{\text{self1:mcs2}} \cdot (\text{SELF1}_i \times \text{MCS2}_i) \\ & + \beta_{\text{self1:mcs3}} \cdot (\text{SELF1}_i \times \text{MCS3}_i) + \beta_{\text{self1:mcs4}} \cdot (\text{SELF1}_i \times \text{MCS4}_i) \\ & + \beta_{\text{self1:single1}} \cdot (\text{SELF1}_i \times \text{SINGLE1}_i) + \beta_{\text{self1:white1}} \cdot (\text{SELF1}_i \times \text{WHITE1}_i) \\ & + \beta_{\text{self1:uria}} \cdot (\text{SELF1}_i \times \text{URIA}_i)\end{aligned} \tag{4}$$

As we can see in Table 4 below, this specification ("Maximal Interaction Model") results in a marginally better AIC and $R^2KL$ than (1), but does so at the cost of including 9 additional parameters, with results in a substantially worse BIC. To see if we can improve either (or both) information criteria, we can now again execute stepwise selection using both BIC and AIC on (4). The former results in the same specification as BIC selection on (1), that is, in specification (3). This again represents a deterioration in AIC and $R^2_{KL}$ compared to (4).

The latter, however, results in the following specification:

$$\begin{aligned}\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 &+ \beta_{\text{self1}} \cdot \text{SELF1}_i + \beta_{\text{hir}} \cdot \text{HIR}_i + \beta_{\text{odir}} \cdot \text{ODIR}_i + \beta_{\text{lvr}} \cdot \text{LVR}_i \\ &+ \beta_{\text{mcs2}} \cdot \text{MCS2}_i + \beta_{\text{mcs3}} \cdot \text{MCS3}_i + \beta_{\text{mcs4}} \cdot \text{MCS4}_i + \beta_{\text{single1}} \cdot \text{SINGLE1}_i \\ &+ \beta_{\text{white1}} \cdot \text{WHITE1}_i + \beta_{\text{uria}} \cdot \text{URIA}_i \\ &+ \beta_{\text{self1:odir}} \cdot (\text{SELF1}_i \times \text{ODIR}_i) \\ &+ \beta_{\text{self1:white1}} \cdot (\text{SELF1}_i \times \text{WHITE1}_i) \\ &+ \beta_{\text{self1:uria}} \cdot (\text{SELF1}_i \times \text{URIA}_i)\end{aligned} \tag{5}$$

Table 4 below shows the number of paramters, information criteria, and $R^2_{KL}$ for (4), (3), and (5). We can see that (5) is substantially more parsimonious than (4), and substantially improves upon it both in terms of AIC and BIC, with a negligible drop of 0.4 percentage points in $R^2_{KL}$.

Table 4: Model Selection: Interaction Models

| | p | AIC | BIC | $R^2_{KL}$ |
|---|---|---|---|---|
| Maximal Interaction Model | 20.000 | 1244.644 | 1355.657 | 0.158 |
| stepwise BIC on interaction model | 6.000 | 1256.842 | 1290.146 | 0.130 |
| Final Model | 14.000 | 1237.667 | 1315.376 | 0.154 |

Table 5 shows the results of a likelihood-ratio test of these three nested models. We can see that there is a highly significant difference in model fit between (3) and (5), but not between (5) and (4). I therefore conclude that (5) is the best-fitting model we can obtain when constraining ourselves to including only interaction terms involving `self` and take specification (5) as my final model.

Table 5: Model Selection: LRT Results

| | Resid. Df | Resid. Dev | Df | Deviance | Pr(>Chi) |
|---|---|---|---|---|---|
| 1 | 1896 | 1244.84 | | | |
| 2 | 1888 | 1209.67 | 8 | 35.17 | 0.0000 |
| 3 | 1882 | 1204.64 | 6 | 5.02 | 0.5409 |

# 4 Model Interpretation

Table 6 below shows estimation results for (3), (4), and (5). Focusing on the rightmost column, we can see that all estimated coefficients with the exceptions of $\hat{\beta}_{\text{self1}}$ and $\hat{\beta}_{\text{mcs4}}$ are individually statistically significant at the 5% level. $\hat{\beta}_{\text{mcs4}}$ is weakly stat. significantly different from zero, while $\hat{\beta}_{\text{self}}$ is only stat. sig. in the interaction term, not in the main effect.

Table 6: Estimation Results for Interaction Models

| | BIC Model | Maximal Interation Model | Final Model |
|---|---|---|---|
| | (1) | (2) | (3) |
| self1 | −0.681*** (0.212) | 0.301 (1.461) | −0.963 (0.710) |
| odir:self1 | | | 0.054** (0.025) |
| white1:self1 | | | −1.389** (0.656) |
| uria:self1 | | | 0.212** (0.087) |
| hir | −0.054*** (0.010) | −0.053*** (0.012) | −0.053*** (0.010) |
| odir | −0.073*** (0.011) | −0.085*** (0.013) | −0.087*** (0.013) |
| lvr | −0.022*** (0.005) | −0.019*** (0.005) | −0.021*** (0.005) |
| mcs2 | | −0.538** (0.221) | −0.540*** (0.195) |
| mcs3 | | −1.176** (0.562) | −0.994** (0.489) |
| mcs4 | | −1.501** (0.656) | −1.069* (0.619) |
| single1 | | −0.271 (0.168) | −0.331** (0.153) |
| white1 | 1.262*** (0.166) | 1.360*** (0.179) | 1.340*** (0.179) |
| uria | | −0.101*** (0.038) | −0.105*** (0.038) |
| self1:hir | | −0.007 (0.024) | |
| self1:odir | | 0.052** (0.026) | |
| self1:lvr | | −0.010 (0.012) | |
| self1:mcs2 | | −0.061 (0.481) | |
| self1:mcs3 | | 0.653 (1.133) | |
| self1:mcs4 | | 13.943 (428.545) | |
| self1:single1 | | −0.386 (0.427) | |
| self1:white1 | | −1.518** (0.683) | |
| self1:uria | | 0.202** (0.090) | |
| Constant | 4.808*** (0.526) | 5.486*** (0.631) | 5.717*** (0.575) |
| Observations | 1,902 | 1,902 | 1,902 |
| Log Likelihood | -622.421 | -602.322 | -604.834 |
| Akaike Inf. Crit. | 1,256.842 | 1,244.644 | 1,237.667 |

*Note:* *p<0.1; **p<0.05; ***p<0.01
SE in Parentheses

Interpreting the estimated effects of `hir, lvr, mcs,` and `single` is relatively straightforward, as they are not included in the interaction terms in (5). Table 7 below displays the estimated coefficients, standard errors, and 95% confidence intervals (CI) for each of these variables on the odds ratio scale. They can be interpreted as follows: the point estimate of 0.948 suggests that for each percentage point increase in the housing-to-income ratio, the estimated odds of approval decrease by approximately 5.3% (as $log(0.948) = -0.053$). The 95% CI (0.929 to 0.967) indicates that this decrease is statistically significant, as the interval does not include 1. Similarly, for one-percentage-point increase in the loan-to-value-ratio, the estimated odds of approval decrease by 2%. An applicant with a `mcs` value of 2 has about 54% lower odds of being approved for a mortgage compared to an applicant with an `mcs` score of 1 (the reference category), for an `mcs` score of 3, the odds decrease by 99.4% compared to a score of 1. For an `mcs` score of 4, the estimated decrease in the odds of approval is $log(0.343) = -107\%$, but the 95% CI for this estimate contains 1, which means that this estimated effect is not statistically significant. Finally, `single` applicants have about 33% lower odds of seeing their mortgage application approved, all else equal.

Table 7: CIs for Predictors not included in Interaction Terms (Odds Ratio Scale)

| | Point Estimate | SE | 95% CI Lower | 95% CI Upper | CI contains 1 |
|---|---|---|---|---|---|
| hir | 0.948 | 1.01 | 0.929 | 0.967 | FALSE |
| lvr | 0.98 | 1.005 | 0.97 | 0.989 | FALSE |
| mcs2 | 0.583 | 1.216 | 0.397 | 0.855 | FALSE |
| mcs3 | 0.37 | 1.631 | 0.142 | 0.966 | FALSE |
| mcs4 | 0.343 | 1.856 | 0.102 | 1.154 | TRUE |
| single | 0.718 | 1.165 | 0.532 | 0.97 | FALSE |

For the predictors that are included in the interaction terms in specification (5), the interpretation becomes a bit more involved. When interacting on `self`, interpretation for observations with `self=0` remains straightforward as above. When applicants are self-employed, however, the effect of increasing `odir` by one percentage point can be computed as $e^{\hat{\beta}_{\text{odir}}+\hat{\beta}_{\text{self1:odir}}}$. The corresponding standard error (on the odds ratio scale) is computed as $exp\left[\sqrt{SE^2_{\text{odir}} + SE^2_{\text{self1:odir}} + 2 \cdot \widehat{\text{Cov}}(\hat{\beta}_{\text{odir}}, \hat{\beta}_{\text{self1:odir}})}\right]$. Table 8 below displays estimated coefficients, standard errors, and 95% confidence intervals for each of the variables included in interaction terms on the odds ratio scale. We can see that for employed applicants, an increase by one percentage point in `odir` is estimated to decrease the odds of approval by 8.6% ($log(0.917) = -0.086$). For self-employed applicants, however, the estimated odds of approval only decrease by 3.25%, although the CI for this estimate contains 1, which means that it is not statistically significant. Similarly, `white` applicants have 33% higher odds of approval, but 38.9% lower odds when they are self-employed (although this estimate is also not statistically significant). Finally, for a one-percentage-point increase in the unemployment rate in an applicant's industry, the estimated odds of approval decrease by 10.4%. For self-employed applicants, this effect changes sign: the estimated odds of approval *increase* by 21.2% (again insignificant).

Table 8: CIs for Predictors included in Interaction Terms (Odds Ratio Scale)

| | Point Estimate | SE | 95% CI Lower | 95% CI Upper | CI contains 1 |
|---|---|---|---|---|---|
| odir:self0 | 0.917 | 1.013 | 0.895 | 0.94 | FALSE |
| white1:self0 | 3.817 | 1.196 | 2.689 | 5.419 | FALSE |
| uria:self0 | 0.901 | 1.039 | 0.835 | 0.971 | FALSE |
| odir:self1 | 0.968 | 1.022 | 0.929 | 1.01 | TRUE |
| white1:self1 | 0.952 | 1.884 | 0.275 | 3.292 | TRUE |
| uria:self1 | 1.113 | 1.082 | 0.955 | 1.298 | TRUE |

## Average Marginal Effects

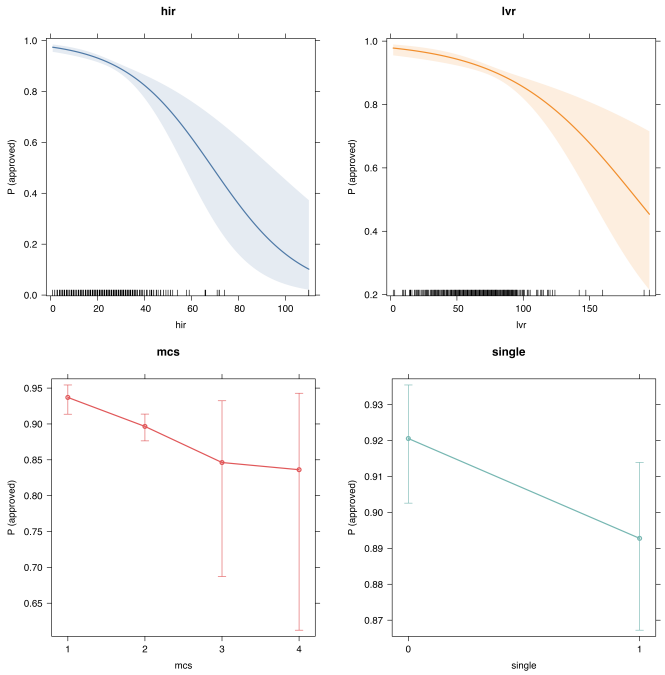
One problem when interpreting our estimation results based on Table 7 and Table 8 is that the overall effect of a variable such as `self` or `odir` is unclear, as we can only interpret their effects for subsets of our sample. While it is possible to compute overall effects by hand (by adapting the two expression given in the preceding paragraph to 3 variables), such computations quickly become tedious. One remedy for this is to compute Average Marginal Effects (AMEs) for each variable. Various R packages offer this functionality. I use `marginaleffects::avg_slopes(final, type = "link")` and exponentiate the results to obtain the AMEs presented in Table 9 below.

Table 9: Average Marginal Effects on Odds Ratio Scale

| | Term | Contrast | Est. Coef. (OR scale) | SE | 95% CI Lower | 95% CI Upper | CI contains 1 |
|---|---|---|---|---|---|---|---|
| 1 | hir | dY/dX | 0.948 | 1.010 | 0.929 | 0.967 | FALSE |
| 2 | lvr | dY/dX | 0.980 | 1.005 | 0.970 | 0.989 | FALSE |
| 3 | mcs | 2 - 1 | 0.583 | 1.216 | 0.397 | 0.855 | FALSE |
| 4 | mcs | 3 - 1 | 0.370 | 1.631 | 0.142 | 0.966 | FALSE |
| 5 | mcs | 4 - 1 | 0.343 | 1.856 | 0.102 | 1.154 | TRUE |
| 6 | odir | dY/dX | 0.923 | 1.011 | 0.902 | 0.944 | FALSE |
| 7 | self | 1 - 0 | 0.390 | 1.245 | 0.253 | 0.599 | FALSE |
| 8 | single | 1 - 0 | 0.718 | 1.165 | 0.532 | 0.970 | FALSE |
| 9 | uria | dY/dX | 0.923 | 1.036 | 0.861 | 0.988 | FALSE |
| 10 | white | 1 - 0 | 3.260 | 1.191 | 2.315 | 4.592 | FALSE |

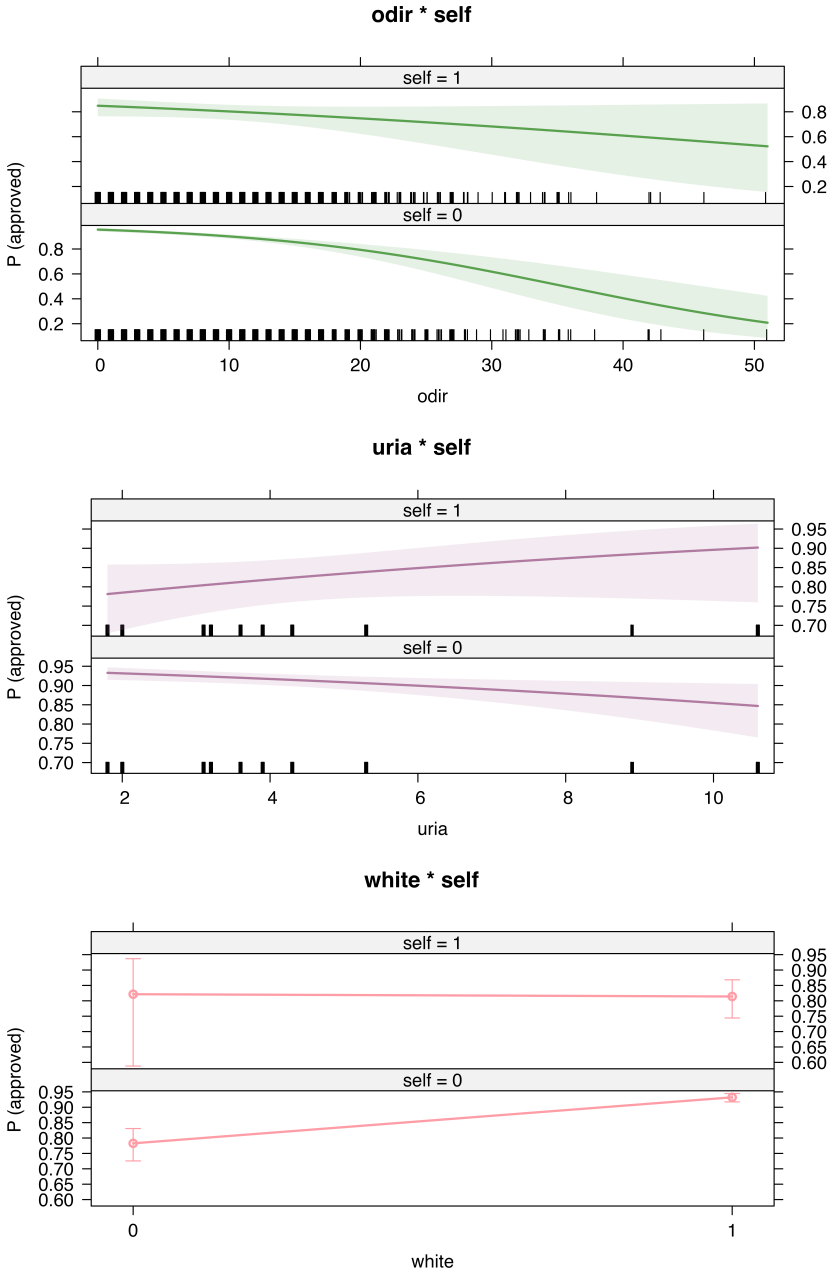
Figure 6 displays effect plots for based on the AMEs for all terms in (5) on the probability scale. The fewer observations we have for levels of variables, such as for housing-to-income ratios in excess of 60%, the wider the associated 95% CIs become.

Figure 6: Effect Plots for Predictors not included in Interaction Terms (Probability Scale)



Finally, Figure 7 displays effect plots for based on the AMEs for the three interaction terms in (5) on the probability scale. The uncertainty as reflected in the width of the CIs is larger here, as we do not have many observations at all levels of `odir, uria`, and `white` for which `self=1`, as can be seen from the rug plots in figures Figure 6 and 7. Figure 7 also gives some helpful intuition as to why the AME, that is, the overall effect considering all interactions, of `uria` is lower in Table 9 than it is in the last line of Table 8: the AME of `uria` for self-employed applicants is positive, but not very precisely estimated due to the low proportion of self-employed applicants (see Table 1)), which means that the negative effect in line 3 of Table 8 dominates and the overall effect is still unambiguously negative.[1] This presents one of the advantages of analyzing AMEs rather than separate point estimates and CIs.

Figure 7: Effect Plots for Predictors included in Interaction Terms (Probability Scale)



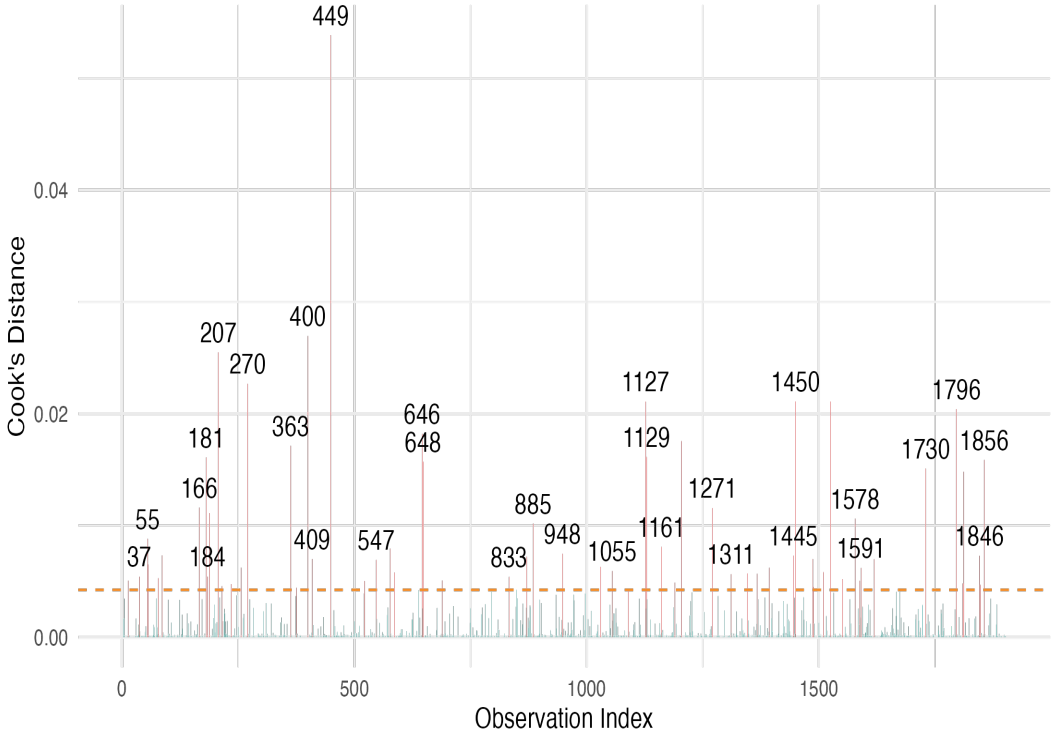

[1] and statistically significant, which cannot be said about the effect of `uria` on self-employed applicants, as per line 6 of Table 8.

# 5 Model Diagnostics

Ideally, we would like to see the variance of the standardized deviance residuals to be approximately equal to one. With specification (5), however, this quantity equals 0.6164, which is indicative of some misfit.

Figure 8 below shows Cook's distances for (5). We can see that 62 observations have a Cook's distance value that exceeds the heuristic of $8/(n-2p)$. Table 11 in Appendix A lists all observations that exceed this threshold. None of them are implausible, so I will not remove them. Overall, specification (5) yields an $R^2_{KL}$ of 15.42%, which is fairly low. Considering the lower-than ideal variance in the standardized residuals, substantial number of outliers, and low $R^2_{KL}$, we can investigate whether our (implicit) assumption of a dispersion parameter of 1 fits our data well.

Figure 8: Cook's Distance for Final Specification




Cutoff line drawn as $8/(n-2p)$

# 6 Estimation of Dispersion Parameter

Using $\hat{\phi} = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i-\hat{\mu}_i)^2}{V(\hat{\mu}_i)}$, we can estimate the dispersion parameter for (5) to be $\hat{\phi} = 1.102$. This means that the observed variance is slightly greater than what the model expects. Overdispersion can be caused –a.o.– by unmodeled heterogeneity, clustering, or model misspecification. To adjust for this, we can multiply our estimated standard error by $\hat{\phi}$ to adjust our confidence intervals, as shown in Table 10 below. Notably, adjusting the Confidence Intervals leads to the CI for `mcs = 3` to now include 1, that is, the estimated coefficient for a credit score of 3 is no longer statistically significant at the 5% level, which it was in Table 9.

Table 10: Average Marginal Effects on Odds Ratio Scale, adjusted for Overdispersion

| | Term | Contrast | Est. Coef. (OR scale) | SE | 95% CI Lower | 95% CI Upper | CI contains 1 |
|---|---|---|---|---|---|---|---|
| 1 | hir | dY/dX | 0.948 | 1.010 | 0.929 | 0.968 | FALSE |
| 2 | lvr | dY/dX | 0.980 | 1.005 | 0.970 | 0.989 | FALSE |
| 3 | mcs | 2 - 1 | 0.583 | 1.216 | 0.390 | 0.871 | FALSE |
| 4 | mcs | 3 - 1 | 0.370 | 1.631 | 0.135 | 1.013 | TRUE |
| 5 | mcs | 4 - 1 | 0.343 | 1.856 | 0.096 | 1.226 | TRUE |
| 6 | odir | dY/dX | 0.923 | 1.011 | 0.901 | 0.945 | FALSE |
| 7 | self | 1 - 0 | 0.390 | 1.245 | 0.248 | 0.612 | FALSE |
| 8 | single | 1 - 0 | 0.718 | 1.165 | 0.524 | 0.984 | FALSE |
| 9 | uria | dY/dX | 0.923 | 1.036 | 0.858 | 0.992 | FALSE |
| 10 | white | 1 - 0 | 3.260 | 1.191 | 2.275 | 4.671 | FALSE |

# 7 Conclusion

The final model fits the data well enough to be an improvement over either a naive additive or overly complex maximal interaction model. We can potentially improve our inferential results by adjusting for the overdispersion present in our data. Even so, specification (5) does not have a lot of explanatory power, and is influenced by outliers that we have no good reason to drop. It is questionable whether the functional form specification is close to the true DGP, but given the data at hand and the modelling restrictions imposed, I have not been able to find a better model. It is plausible that additional information on wealth, savings, or other collateral an applicant indicated on their mortgage application could be employed to remove outliers that seem implausible given our data (such as observation 449) and improve our model.

# 8 Appendix A: Influential Points

Table 11: Observations with High Cook's distance

| Index | approved | hir | odir | lvr | mcs | self | single | white | uria |
|---|---|---|---|---|---|---|---|---|---|
| 13 | 1 | 32.00 | 5.00 | 86.00 | 2 | 1 | 1 | 0 | 1.80 |
| 37 | 0 | 23.00 | 1.00 | 90.00 | 2 | 0 | 1 | 1 | 10.60 |
| 55 | 0 | 28.00 | 7.00 | 15.00 | 1 | 0 | 1 | 1 | 3.20 |
| 56 | 0 | 8.00 | 11.00 | 75.00 | 2 | 1 | 1 | 1 | 1.80 |
| 78 | 0 | 15.00 | 22.00 | 78.00 | 3 | 0 | 0 | 0 | 3.90 |
| 86 | 1 | 19.00 | 3.00 | 195.00 | 1 | 0 | 0 | 0 | 3.20 |
| 166 | 1 | 22.00 | 3.00 | 160.00 | 3 | 1 | 0 | 1 | 10.60 |
| 181 | 0 | 7.00 | 13.00 | 80.00 | 3 | 0 | 1 | 1 | 3.60 |
| 184 | 0 | 13.00 | 28.00 | 80.00 | 1 | 0 | 0 | 1 | 8.90 |
| 188 | 1 | 1.00 | 35.00 | 147.00 | 2 | 1 | 0 | 1 | 1.80 |
| 207 | 0 | 10.00 | 0.00 | 67.00 | 2 | 1 | 1 | 1 | 10.60 |
| 215 | 0 | 10.00 | 7.00 | 80.00 | 1 | 0 | 0 | 1 | 3.10 |
| 235 | 0 | 23.00 | 7.00 | 68.00 | 2 | 0 | 1 | 1 | 10.60 |
| 256 | 0 | 20.00 | 0.00 | 92.00 | 2 | 0 | 1 | 1 | 10.60 |
| 270 | 0 | 18.00 | 5.00 | 61.00 | 1 | 1 | 0 | 1 | 10.60 |
| 363 | 0 | 35.00 | 1.00 | 69.00 | 2 | 1 | 1 | 0 | 3.20 |
| 375 | 0 | 19.00 | 2.00 | 71.00 | 2 | 1 | 1 | 1 | 4.30 |
| 400 | 0 | 31.00 | 25.00 | 57.00 | 3 | 1 | 0 | 1 | 10.60 |
| 409 | 0 | 13.00 | 34.00 | 80.00 | 3 | 0 | 1 | 1 | 3.10 |
| 449 | 1 | 44.00 | 51.00 | 55.00 | 3 | 1 | 0 | 1 | 10.60 |
| 522 | 0 | 22.00 | 5.00 | 73.00 | 2 | 0 | 0 | 1 | 10.60 |
| 547 | 1 | 72.00 | 11.00 | 72.00 | 2 | 0 | 1 | 1 | 3.60 |
| 577 | 0 | 16.00 | 23.00 | 80.00 | 2 | 1 | 0 | 1 | 1.80 |
| 586 | 0 | 35.00 | 8.00 | 95.00 | 4 | 0 | 1 | 0 | 1.80 |
| 646 | 0 | 18.00 | 12.00 | 90.00 | 2 | 1 | 0 | 0 | 1.80 |
| 648 | 0 | 37.00 | 36.00 | 46.00 | 1 | 1 | 1 | 1 | 3.20 |
| 689 | 0 | 23.00 | 27.00 | 51.00 | 1 | 0 | 1 | 1 | 3.20 |
| 833 | 1 | 30.00 | 4.00 | 71.00 | 4 | 1 | 1 | 1 | 3.20 |
| 871 | 1 | 27.00 | 7.00 | 191.00 | 2 | 0 | 1 | 1 | 3.10 |
| 885 | 0 | 31.00 | 16.00 | 89.00 | 2 | 1 | 1 | 0 | 3.20 |
| 948 | 0 | 3.00 | 1.00 | 67.00 | 2 | 0 | 1 | 1 | 4.30 |
| 1030 | 0 | 37.00 | 0.00 | 65.00 | 1 | 1 | 0 | 1 | 4.30 |
| 1055 | 0 | 13.00 | 10.00 | 34.00 | 2 | 0 | 0 | 1 | 3.60 |
| 1090 | 0 | 21.00 | 30.00 | 64.00 | 2 | 0 | 0 | 1 | 1.80 |
| 1127 | 0 | 13.00 | 8.00 | 80.00 | 2 | 1 | 1 | 0 | 3.20 |
| 1129 | 1 | 39.00 | 42.00 | 52.00 | 2 | 0 | 1 | 0 | 3.20 |
| 1161 | 0 | 21.00 | 22.00 | 56.00 | 1 | 1 | 0 | 1 | 3.20 |
| 1190 | 0 | 12.00 | 0.00 | 80.00 | 2 | 0 | 1 | 1 | 1.80 |
| 1204 | 0 | 20.00 | 11.00 | 100.00 | 4 | 0 | 0 | 1 | 10.60 |
| 1271 | 0 | 5.00 | 13.00 | 19.00 | 2 | 0 | 1 | 0 | 2.00 |
| 1311 | 0 | 13.00 | 3.00 | 142.00 | 2 | 1 | 1 | 1 | 3.20 |
| 1346 | 0 | 16.00 | 20.00 | 81.00 | 2 | 1 | 1 | 1 | 1.80 |
| 1367 | 1 | 33.00 | 9.00 | 80.00 | 3 | 0 | 1 | 0 | 3.20 |
| 1393 | 0 | 2.00 | 5.00 | 77.00 | 2 | 0 | 0 | 1 | 3.60 |
| 1445 | 0 | 32.00 | 11.00 | 79.00 | 3 | 1 | 0 | 1 | 3.20 |
| 1450 | 0 | 11.00 | 36.00 | 90.00 | 1 | 1 | 0 | 1 | 3.20 |
| 1487 | 0 | 21.00 | 0.00 | 61.00 | 2 | 0 | 0 | 0 | 10.60 |
| 1488 | 0 | 23.00 | 5.00 | 70.00 | 1 | 1 | 1 | 1 | 3.20 |
| 1510 | 0 | 26.00 | 0.00 | 69.00 | 2 | 0 | 0 | 1 | 10.60 |
| 1525 | 0 | 6.00 | 12.00 | 80.00 | 4 | 0 | 0 | 0 | 3.20 |
| 1551 | 0 | 38.00 | 31.00 | 80.00 | 2 | 1 | 0 | 1 | 3.20 |
| 1578 | 0 | 27.00 | 13.00 | 90.00 | 3 | 0 | 0 | 1 | 3.60 |
| 1588 | 0 | 26.00 | 6.00 | 79.00 | 1 | 1 | 1 | 1 | 1.80 |
| 1591 | 0 | 16.00 | 5.00 | 60.00 | 2 | 1 | 1 | 1 | 1.80 |
| 1619 | 0 | 43.00 | 0.00 | 64.00 | 1 | 1 | 0 | 1 | 3.10 |
| 1730 | 0 | 39.00 | 0.00 | 63.00 | 3 | 0 | 1 | 1 | 10.60 |
| 1796 | 0 | 30.00 | 11.00 | 72.00 | 4 | 0 | 0 | 1 | 3.20 |
| 1810 | 0 | 26.00 | 0.00 | 50.00 | 1 | 0 | 0 | 1 | 3.60 |
| 1812 | 0 | 29.00 | 0.00 | 69.00 | 1 | 1 | 1 | 1 | 8.90 |
| 1846 | 0 | 18.00 | 16.00 | 50.00 | 1 | 0 | 0 | 1 | 10.60 |
| 1848 | 0 | 33.00 | 5.00 | 42.00 | 1 | 0 | 0 | 1 | 4.30 |
| 1856 | 0 | 43.00 | 3.00 | 80.00 | 2 | 1 | 0 | 1 | 10.60 |

# 9 Appendix B: Source Code

```r
library(tidyverse)
library(xtable)
library(ggplot2)
library(gridExtra)
library(patchwork) # to combine plots
library(stargazer)
library(kableExtra)
library(sandwich) # for robust Standard Errors
library(MASS) # for stepwise AIC and BIC selection
library(rsq)
library(effects)
library(marginaleffects)
# Get colors from the tableau palette
plotcolors <- ggthemes::tableau_color_pal()(10)
## Set Paths for tables and figures
root = "/Users/ts/Git/Practicals"
tab = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/GLM Practical/Tables"
fig = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/GLM Practical/Figures"
code = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/GLM Practical/Code"

setwd(code)
file.copy(from = "/Users/ts/Git/Practicals/GLM-assessed-practical.R",
          to = "/Users/ts/Library/CloudStorage/Dropbox/Apps/Overleaf/GLM Practical/Code",
          overwrite = T)
setwd(root)
if (getwd() != root) {
  setwd(root)
}
# Load data
data_raw <- read_csv("mortg.csv")

### EDA ###
# Cast as factors
data <- data_raw %>%
  mutate(
    approved = as.factor(approved),
    self = as.factor(self),
    single = as.factor(single),
    white = as.factor(white),
    mcs = factor(mcs, levels = 1:4),
    hir = 100 * hir,
    lvr = 100 * lvr,
    odir = 100 * odir
  )
```

```r

attach(data)
# Summary stats functions
summary_stats <- function(x) {
  c(mean = mean(x, na.rm = T),
    median = median(x, na.rm = T),
    sd = sd(x, na.rm = T),
    min = min(x, na.rm = T),
    max = max(x, na.rm = T))
}

# Split data
data_num <- data %>% dplyr::select(hir, odir, lvr, uria)
data_fact <- data_raw %>% dplyr::select(approved, mcs, self, single, white)

# Get sumstats
# Apply the function to each column
numerical_summary <- t(sapply(data_num, summary_stats))
factor_summary <- t(sapply(data_fact, summary_stats))
sumstats <- rbind(numerical_summary, factor_summary)
colnames(sumstats) <- c("Mean", "Median", "SD", "Min", "Max")

# Make table
sumtable <- xtable(sumstats, caption = "Summary Statistics", label = "sumstats")

#### MODEL BUILDING ####
# Fit baseline model
baseline <- glm(approved ~ ., data = data, family = binomial(link = "logit"))

# Check how to cast mcs by
data_num <- data_raw %>% mutate(
  approved = as.factor(approved),
  self = as.factor(self),
  single = as.factor(single),
  white = as.factor(white),
)

# Refit full model using mcs as numerical
baseline_num <- glm(approved ~ ., data = data_num,
                    family = binomial(link = "logit"))

# stepwise AIC on baseline
step_aic <- stepAIC(baseline, direction = "backward")

# stepwise BIC on baseline
```

```r
step_bic <- step(baseline, direction = "backward", k = log(nrow(data)))

# Interaction terms
# fit maximally interacted model
maximal <- glm(approved ~ self * ., family = binomial, data = data)

step_interact_aic <- stepAIC(maximal, direction = "backward")
step_interact_bic <- step(maximal, direction = "backward", k = log(nrow(data)))

bic <- glm(approved ~ self + hir + odir + lvr + white, data = data,
           family = binomial)
final <- glm(approved ~ uria + hir + odir + lvr + mcs + single + white +
                        self*odir + self*white + self*uria + self,
                      family = binomial, data = data)

# Do LRT
LRT_selection <- anova(bic, final, maximal, test = "Chisq")
# export LRT
model_selection <- xtable(LRT_selection, caption = "Model Selection: LRT Results",
                          label = "LRT-select")

# make tables with p, AIC, BIC, RSQ_KL
modelmat_additive <- matrix(NA, 3, 4)
modelmat_interact <- matrix(NA, 3, 4)
modelmat_additive[1,1] <- length(coef(baseline))
modelmat_additive[2,1] <- length(coef(step_aic))
modelmat_additive[3,1] <- length(coef(step_bic))
modelmat_interact[1,1] <- length(coef(maximal))
modelmat_interact[2,1] <- length(coef(bic))
modelmat_interact[3,1] <- length(coef(final))

modelmat_additive[1,2] <- AIC(baseline)
modelmat_additive[2,2] <- AIC(step_aic)
modelmat_additive[3,2] <- AIC(step_bic)
modelmat_interact[1,2] <- AIC(maximal)
modelmat_interact[2,2] <- AIC(bic)
modelmat_interact[3,2] <- AIC(final)

modelmat_additive[1,3] <- BIC(baseline)
modelmat_additive[2,3] <- BIC(step_aic)
modelmat_additive[3,3] <- BIC(step_bic)
modelmat_interact[1,3] <- BIC(maximal)
modelmat_interact[2,3] <- BIC(bic)
modelmat_interact[3,3] <- BIC(final)

```

```r
modelmat_additive[1,4] <- rsq.kl(baseline)
modelmat_additive[2,4] <- rsq.kl(step_aic)
modelmat_additive[3,4] <- rsq.kl(step_bic)
modelmat_interact[1,4] <- rsq.kl(maximal)
modelmat_interact[2,4] <- rsq.kl(bic)
modelmat_interact[3,4] <- rsq.kl(final)

rownames(modelmat_additive) <- c("Baseline", "stepwise AIC on baseline",
                                 "stepwise BIC on baseline")
rownames(modelmat_interact) <- c("Maximal Interaction Model",
                                 "stepwise BIC on interaction model",
                                 "Final Model")
colnames(modelmat_additive) <- c("p", "AIC", "BIC", "$R^2_{KL}$")
colnames(modelmat_interact) <- c("p", "AIC", "BIC", "$R^2_{KL}$")
model_tab_add <- xtable(modelmat_additive,
                        caption = "Model Selection: Additive Models",
                        label = "select-add",
                        align = c("r", rep("c", 4)), digits = 3)
model_tab_int <- xtable(modelmat_interact,
                        caption = "Model Selection: Interaction Models",
                        label = "selectiontable-int",
                        align = c("r", rep("c", 4)), digits = 3)
sanitize_latex <- function(x) {
  gsub("\\$", "\\\\\\$", x, fixed = T)
}

#### Diagnostics ####
# Compute Cook's distances
cooks_distances <- cooks.distance(final)
# Create df for plotting
cooks_df <- data.frame(Index = 1:length(cooks_distances),
                       CooksDistance = cooks_distances)
# Define cutoff
cutoff <- 8 / (nrow(data) - length(coef(final)))
# Add a new column to indicate points above the cutoff
cooks_df$Label <- ifelse(cooks_df$CooksDistance > cutoff,
                         as.character(cooks_df$Index), "")

# plot
cooks <- ggplot(cooks_df, aes(x = Index, y = CooksDistance)) +
  geom_bar(stat = "identity", position = "identity",
           aes(fill = CooksDistance > cutoff), width = 1) +
  scale_fill_manual(values = c(plotcolors[4], plotcolors[3])) +
  geom_hline(yintercept = cutoff, linetype = "dashed", color = plotcolors[2]) +
  geom_text(aes(label = Label), vjust = -0.5, check_overlap = T) +
```

```
  theme_minimal() +
  labs(x = "Observation Index",
       y = "Cook's Distance") +
  theme(legend.position = "none")

# Pull points that have high cook's dist
influential_indices <- cooks_df$Index[cooks_df$CooksDistance > cutoff]

# Subset data using indices
influential_points <- data[influential_indices, ] %>%
  mutate(Index = influential_indices) %>%
  dplyr::select(Index, everything())
# Make table to print in appendix
infl <- xtable(influential_points,
               caption = "Observations with High Cook's distance",
               label = "infl")

# We would hope to see roughly unit variance of the standardized residuals
var(rstandard(final))

#### Interpretation ####
model_summary <- broom::tidy(final)

# Compute all main and interaction effects
all_effects <- allEffects(final)

# Main Effect / Focal CIs
compute_focal_CI <- function(mod, var, alpha = 0.05) {
  # Extract VCOV and coefs
  coefs_var <- vcov(mod)
  coefs <- coef(mod)
  pe <- coef(final)[var]
  # Calc the z-value using alpha
  z <- qnorm(1 - alpha / 2)

  # Calc SE for the focal term
  se <- sqrt(coefs_var[var, var])
  ci_l <- exp(pe - z * se)
  ci_u <- exp(pe + z * se)
  ci_contains_1 <- ifelse((ci_l <= 1 & ci_u >= 1), "TRUE", "FALSE")

  return(c(pe = round(exp(pe), 3),
           se = round(exp(se), 3),
           ci_lower = round(ci_l,3),
           ci_upper = round(ci_u, 3),
```

```
            ci_cont_1 = ci_contains_1))
}

## Make table for focal predictors
hir_ci <- compute_focal_CI(final, "hir")
lvr_ci <- compute_focal_CI(final, "lvr")
mcs_2_ci <- compute_focal_CI(final, "mcs2")
mcs_3_ci <- compute_focal_CI(final, "mcs3")
mcs_4_ci <- compute_focal_CI(final, "mcs4")
single_ci <- compute_focal_CI(final, "single1")

focal_cis <- rbind(hir_ci, lvr_ci, mcs_2_ci, mcs_3_ci, mcs_4_ci, single_ci)
colnames(focal_cis) <- c("Point Estimate", "SE", "95% CI Lower",
                         "95% CI Upper", "CI contains 1")
rownames(focal_cis) <- c("hir", "lvr", "mcs2", "mcs3" , "mcs4", "single")

focal_CI <- xtable(focal_cis, digits = 3, align = c("r", rep("c", 5)),
                   caption = "CIs for Predictors not included in Interaction
                   Terms (Odds Ratio Scale)", label = "focal")


# Interaction Effect CIs
compute_interaction_CI <- function(mod, var1, var2 = "self1", alpha = 0.05) {
  # Extract VCOV and coefs
  coefs_var <- vcov(mod)
  coefs <- coef(mod)
  pe <- coefs[var1] + coefs[var2]

  # Calc the z-value using alpha
  z <- qnorm(1 - alpha / 2)

  # Calc SE for the interaction term
  se <- sqrt(coefs_var[var1, var1] + coefs_var[var2,var2] +
               2 * coefs_var[var1,var2])
  ci_l <- exp(pe - z * se)
  ci_u <- exp(pe + z * se)
  ci_contains_1 <- ifelse((ci_l <= 1 & ci_u >= 1), "TRUE", "FALSE")

  return(c(pe = round(exp(pe), 3),
           se = round(exp(se), 3),
           ci_lower = round(ci_l,3),
           ci_upper = round(ci_u, 3),
           ci_cont_1 = ci_contains_1))
}
```

```r
## Make table for self = 1
selfodir_ci <- unname(compute_interaction_CI(final, "odir", "odir:self1"))
selfwhite_ci <- unname(compute_interaction_CI(final, "white1", "white1:self1"))
selfuria_ci <- unname(compute_interaction_CI(final, "uria", "uria:self1"))

## Make table for self = 0
odir_ci <- compute_focal_CI(final, "odir")
white_ci <- compute_focal_CI(final, "white1")
uria_ci <- compute_focal_CI(final, "uria")

interact_cis <- rbind(odir_ci, white_ci, uria_ci,
                      selfodir_ci, selfwhite_ci, selfuria_ci)

# Set column and row names
colnames(interact_cis) <- c("Point Estimate", "SE", "95% CI Lower",
                            "95% CI Upper", "CI contains 1")
rownames(interact_cis) <- c("odir:self0", "white1:self0", "uria:self0",
                            "odir:self1", "white1:self1", "uria:self1")


interact_CI <- xtable(interact_cis, digits = 3, align = c("r", rep("c", 5)),
                      caption = "CIs for Predictors included in Interaction
                      Terms (Odds Ratio Scale)",
                      label = "CI-inter")
# make table of AME estimates and CIs on odds-ratio scale
AME_ors <- tidy(marginaleffects::avg_slopes(final, type = "link")) %>%
  dplyr::select(-c(5:7)) %>%
  mutate(
    estimate = exp(estimate),
    std.error = exp(std.error),
    cil = exp(conf.low),
    cih = exp(conf.high),
    CI_contains_1 = (cil <= 1 & cih >= 1)
  ) %>% dplyr::select( -c(conf.low, conf.high))


colnames(AME_ors) <- c("Term", "Contrast", "Est. Coef. (OR scale)", "SE",
                       "95% CI Lower", "95% CI Upper", "CI contains 1")

resultstable <- xtable(AME_ors,
                       caption = "Average Marginal Effects on Odds Ratio Scale",
                       label = "AME",
                       align = c("r", rep("c", 7)),
                       include.rownames = F, digits = 3)

```

```r
#### Dispersion ####
mu_hat <- predict(final, type = "response")
y <- as.numeric(data$approved ) - 1
V_mu_hat <- mu_hat * (1 - mu_hat) # Variance function for binomial distribution

phi_hat <- 1 / (nrow(data) - length(coef(final)) ) *
  sum((y - mu_hat)^2 / V_mu_hat)
disp_adj <- sqrt(phi_hat)

# Adjust for overdispersion
AME_adj <- tidy(marginaleffects::avg_slopes(final, type = "link")) %>%
  dplyr::select(-c(5:7)) %>%
  mutate(
    cil = exp(estimate - 1.96 * disp_adj * std.error),
    cih = exp(estimate + 1.96 * disp_adj * std.error),
    estimate = exp(estimate),
    std.error = exp(std.error),
    CI_contains_1 = (cil <= 1 & cih >= 1)
  ) %>% dplyr::select( -c(conf.low, conf.high))

colnames(AME_adj) <- c("Term", "Contrast", "Est. Coef. (OR scale)", "SE",
                       "95% CI Lower", "95% CI Upper", "CI contains 1")
resultstable_adj <- xtable(AME_adj,
                           caption = "Average Marginal Effects on Odds Ratio Scale,
                           adjusted for Overdispersion",
                           label = "CI-adj",
                       align = c("r", rep("c", 7)),
                       include.rownames = F, digits = 3)

#### Effect plots ####
plot_list <- lapply(names(all_effects), function(name) {
  # Shorten subplot titles
  plot(all_effects[[name]], main = name)
})

focal_effects <- plot_list[1:4]
interaction_effects <- plot_list[5:7]

# Combine plots (ugly presets)
effects <- do.call(grid.arrange, c(plot_list, ncol = 3))
# Manually for plot formatting
## Focal predictors (not included in interaction terms)
eff_plot_hir <- plot(predictorEffect("hir", final),
     axes=list(y=list(type="response", lab = "P (approved)"),
               x=list(rug=T)), main = "hir",
```

```
     lines=list(col=plotcolors[1]))

eff_plot_lvr <- plot(predictorEffect("lvr", final),
               axes=list(y=list(type="response", lab = "P (approved)"),
                         x=list(rug=T)), main = "lvr",
               lines=list(col=plotcolors[2]))

eff_plot_mcs <- plot(predictorEffect("mcs", final),
               axes=list(y=list(type="response", lab = "P (approved)"),
                         x=list(rug=T)), main = "mcs",
               lines=list(col=plotcolors[3]))

eff_plot_single <- plot(predictorEffect("single", final),
               axes=list(y=list(type="response", lab = "P (approved)"),
                         x=list(rug=T)), main = "single",
               lines=list(col=plotcolors[4]))
# Collect plots
focal_effect_plot_list <- lapply(ls(pattern = "^eff_plot_"), get)
# Arrange
focal_effect_plots <- do.call(grid.arrange,
                              c(focal_effect_plot_list, ncol = 2))

## Interaction Terms
interaction_eff_plot_odir <- plot(predictorEffect("odir", final),
                     axes=list(y=list(type="response", lab = "P (approved)"),
                               x=list(rug=T)), main = "odir * self",
                     lines=list(col=plotcolors[5]))

interaction_eff_plot_white <- plot(predictorEffect("white", final),
                     axes=list(y=list(type="response", lab = "P (approved)"),
                               x=list(rug=T)), main = "white * self",
                     lines=list(col=plotcolors[8]))

interaction_eff_plot_uria <- plot(predictorEffect("uria", final),
                     axes=list(y=list(type="response", lab = "P (approved)"),
                               x=list(rug=T)), main = "uria * self",
                     lines=list(col=plotcolors[7]))

# Collect
interaction_effect_plot_list <- lapply(ls(pattern = "^interaction_eff_plot_"),
                                       get)
# Arrange
interaction_effect_plots <- do.call(grid.arrange,
                                    c(interaction_effect_plot_list, nrow = 1))
#### EDA plots ####
```

```r
# Continuous predictors, bivariate
plot_hir <- ggplot(data, aes(x = approved, y = hir, fill = approved)) +
  geom_violin(trim = T) + # Add violin plot
  geom_boxplot(width = 0.1, fill = "white") + # Add boxplot inside
  scale_fill_manual(values = plotcolors[1:2]) +
  theme_minimal() +
  labs(x = "Approved", y = "Ratio of Housing Expenses to Income") +
  theme(legend.position = "none")

plot_odir <- ggplot(data, aes(x = approved, y = odir, fill = approved)) +
  geom_violin(trim = T) + # Add violin plot
  geom_boxplot(width = 0.1, fill = "white") + # Add boxplot inside
  scale_fill_manual(values = plotcolors[3:4]) +
  theme_minimal() +
  labs(x = "Approved", y = "Ratio of other Debt Payments to Income") +
  theme(legend.position = "none")

plot_lvr <- ggplot(data, aes(x = approved, y = lvr, fill = approved)) +
  geom_violin(trim = T) + # Add violin plot
  geom_boxplot(width = 0.1, fill = "white") + # Add boxplot inside
  scale_fill_manual(values = plotcolors[5:6]) +
  theme_minimal() +
  labs(x = "Approved", y = "LTV Ratio") +
  theme(legend.position = "none")

plot_uria <- ggplot(data, aes(x = approved, y = uria, fill = approved)) +
  geom_violin(trim = T) + # Add violin plot
  geom_boxplot(width = 0.1, fill = "white") + # Add boxplot inside
  scale_fill_manual(values = plotcolors[7:8]) +
  theme_minimal() +
  labs(x = "Approved", y = "Unempl. Rate in Appl. Industry") +
  theme(legend.position = "none")

boxplots <- (plot_hir | plot_odir | plot_lvr | plot_uria) +
  plot_layout(guides = 'collect') &
  theme(legend.position = "bottom")

# Continuous predictors, trivariate, self-employed
plot_self_hir <- ggplot(data, aes(x = approved, y = hir, fill = self)) +
  geom_violin(trim = T) +
  scale_fill_manual(values = plotcolors[1:2]) +
  theme_minimal() +
  labs(x = "Approved", y = "Ratio of Housing Expenses to Income") +
  theme(legend.position = "bottom")

```

```r
plot_self_odir <- ggplot(data, aes(x = approved, y = odir, fill = self)) +
  geom_violin(trim = T) +
  scale_fill_manual(values = plotcolors[1:2]) +
  theme_minimal() +
  labs(x = "Approved", y = "Ratio of other Debt Payments to Income") +
  theme(legend.position = "bottom")

plot_self_lvr <- ggplot(data, aes(x = approved, y = lvr, fill = self)) +
  geom_violin(trim = T) +
  scale_fill_manual(values = plotcolors[1:2]) +
  theme_minimal() +
  labs(x = "Approved", y = "LTV Ratio") +
  theme(legend.position = "bottom")

plot_self_uria <- ggplot(data, aes(x = approved, y = uria, fill = self)) +
  geom_violin(trim = T) +
  scale_fill_manual(values = plotcolors[1:2]) +
  theme_minimal() +
  labs(x = "Approved", y = "Unempl. Rate in Appl. Industry") +
  theme(legend.position = "bottom")

violinplots <- (plot_self_hir|plot_self_odir|plot_self_lvr|plot_self_uria) +
  plot_layout(guides = 'collect') &
  theme(legend.position = "bottom")

# Factor Predictors (mcs, self, single, white)
plot_mcs <- ggplot(data, aes(x = approved, fill = mcs)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[1:4]) +
  theme_minimal() +
  labs(x = "Approved", y = "Percentage") +
  theme(legend.position = "bottom")

plot_self <- ggplot(data, aes(x = approved, fill = self)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[5:6]) +
  theme_minimal() +
  labs(x = "Approved", y = "Percentage") +
  theme(legend.position = "bottom")

plot_single <- ggplot(data, aes(x = approved, fill = single)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
```

```
  scale_fill_manual(values = plotcolors[7:8]) +
  theme_minimal() +
  labs(x = "Approved", y = "Percentage") +
  theme(legend.position = "bottom")

plot_white <- ggplot(data, aes(x = approved, fill = white)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[9:10]) +
  theme_minimal() +
  labs(x = "Approved", y = "Percentage") +
  theme(legend.position = "bottom")

mosaicplots <- (plot_mcs | plot_self | plot_single | plot_white)

## Use Bar prop plots instead
# function to get proportions
data_prop <- function(cat) {
  data %>%
    group_by(!!sym(cat), approved) %>%
    summarise(Count = n(), .groups = 'drop') %>%
    mutate(Fraction = Count / sum(Count))
}

prop_plot_mcs <- ggplot(data_prop('mcs'), aes(x = mcs, y = Fraction,
                                              fill = as.factor(approved))) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[1:2]) +
  labs(x = "MCS", y = "Percentage", fill = "Approved") +
  theme_minimal() +
  theme(legend.position = "bottom")

prop_plot_self <- ggplot(data_prop('self'), aes(x = self, y = Fraction,
                                                fill = as.factor(approved))) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[3:4]) +
  labs(x = "Self", y = "Percentage", fill = "Approved") +
  theme_minimal() +
  theme(legend.position = "bottom")

prop_plot_single <- ggplot(data_prop('single'), aes(x = single, y = Fraction,
                                                    fill = as.factor(approved))) +
  geom_bar(stat = "identity", position = "fill") +
```

```
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[5:6]) +
  labs(x = "Single", y = "Percentage", fill = "Approved") +
  theme_minimal() +
  theme(legend.position = "bottom")

prop_plot_white <- ggplot(data_prop('white'), aes(x = white, y = Fraction,
                                                  fill = as.factor(approved))) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = plotcolors[7:8]) +
  labs(x = "White", y = "Percentage", fill = "Approved") +
  theme_minimal() +
  theme(legend.position = "bottom")

prop_plots <- (prop_plot_mcs|prop_plot_self|prop_plot_single|prop_plot_white)

# Decile plots
plot_deciles <- function(var_name) {

  var <- data[[var_name]]
  # Calculate group-wise means
  breaks <- quantile(var, probs = seq(0, 1, by = 0.1), na.rm = T,
                     names = F, type = 7)
  breaks[length(breaks)] <- max(var, na.rm = T)  # Ensure max  is included
  group_means <- tapply(as.numeric(data$approved) - 1,
          findInterval(var, breaks, rightmost.closed = T), mean, na.rm = T)
  group_means <- na.omit(group_means)

  # Convert to data frame for ggplot
  plot_data <- data.frame(
    Deciles = factor(names(group_means),
                     levels = as.character(1:length(group_means))),
    Mean = as.numeric(group_means)
  )

  # plot
  ggplot(plot_data, aes(x = Deciles, y = Mean, fill = factor(Deciles))) +
    geom_bar(stat = "identity", show.legend = F) +
    scale_fill_manual(values = plotcolors[1:10]) +
    theme_minimal() +
    theme(legend.position = "none") +
    ylim(0, 1) +
    labs(x = var_name, y = "Mean", )
}
```

```
585
586 decile_hir <- plot_deciles('hir')
587 decile_odir <- plot_deciles('odir')
588 decile_lvr <- plot_deciles('lvr')
589 decile_uria <- plot_deciles('uria')
590
591 # Stack the plots
592 decile_plots <- decile_hir / decile_odir / decile_lvr / decile_uria
593
594 #### Export ####
595 # tables
596 setwd(tab)
597 print.xtable(sumtable, type = "latex", file = "sumstats.tex",
598              include.rownames = T, digits = 2, align = c("l", rep("c", 4)),
599              caption.placement = "top",
600              floating = T, table.placement = "H")
601
602 print.xtable(model_selection, type = "latex", file = "selection.tex",
603              include.rownames = T, digits = 2,
604              caption.placement = "top",
605              floating = T, table.placement = "H")
606
607 print.xtable(model_tab_add, type = "latex", file = "models-add.tex",
608              include.rownames = T,
609              caption.placement = "top",
610              sanitize.text.function = sanitize_latex,
611              table.placement = "H")
612
613 print.xtable(model_tab_int, type = "latex", file = "models-int.tex",
614              include.rownames = T,
615              caption.placement = "top",
616              sanitize.text.function = sanitize_latex,
617              table.placement = "H")
618
619 print.xtable(focal_CI, type = "latex", file = "focal-ci.tex",
620              include.rownames = T,
621              digits = 3,
622              caption.placement = "top",
623              table.placement = "H")
624
625 print.xtable(interact_CI, type = "latex", file = "interaction-ci.tex",
626              include.rownames = T,
627              digits = 3,
628              caption.placement = "top",
629              table.placement = "H",
```

```
              hline.after = c(-1,0,3,6))

print.xtable(resultstable, type = "latex", file = "AME.tex",
              include.rownames = T,
              caption.placement = "top",
              table.placement = "H")

print.xtable(resultstable_adj, type = "latex", file = "AME-adj.tex",
              include.rownames = T,
              caption.placement = "top",
              table.placement = "H")

print.xtable(infl, type = "latex", file = "infl.tex",
              include.rownames = F,
              caption.placement = "top",
              table.placement = "H",
              size = "tiny")

stargazer(baseline, baseline_num, title="Estimation Results for Baseline Model",
          type="latex",
          label = "baselineresults",
          align=T,
          out = "baseline.tex",
          column.labels=c("MCS as factor", "MCS as numerical"),
          dep.var.caption="",
          dep.var.labels.include = F,
          no.space=T,
          single.row=T,
          header=F,
          digits=3, notes = "SE in Parentheses")

stargazer(bic, maximal, final, title="Estimation Results for Interaction Models",
          type="latex",
          align=T,
          out = "regtable.tex",
          label = "regresults",
          column.labels=c("BIC Model", "Maximal Interation Model",  "Final Model"),
          dep.var.caption="",
          dep.var.labels.include = F,
          no.space=T,
          single.row=T,
          header=F,
          digits=3, notes = "SE in Parentheses")
setwd(root)
```

```
675 # figures
676 setwd(fig)
677 ggsave(plot = boxplots, "boxplots.png")
678 ggsave(plot = mosaicplots, "mosaicplots.png")
679 ggsave(plot = prop_plots, "prop-plots.png")
680 ggsave(plot = violinplots, "violinplots.png")
681 ggsave(plot = decile_plots, "decileplots.png")
682 ggsave(plot = cooks, "cooksdist.png")
683 ggsave(plot = effects, "effects.png", dpi = 1000, height = 30, width = 30, units = "cm")
684 ggsave(plot = focal_effect_plots, "focal-effects.png", dpi = 1000, height = 30, width = 30, units = "cm")
685 ggsave(plot = interaction_effect_plots, "interaction-effects.png", dpi = 1000, height = 30, width = 30, uni
686 setwd(root)
687
```