Baseline Characteristics as Potential Moderators and Predictors of Smoking Cessation in Adults with Major Depressive Disorder

PHP2550 Project 2: A regression analysis

Yanwei (Iris) Tong

2024-10-10 (Revised 2024-12-10)

Abstract

Purpose: Building on a previous randomized, placebo-controlled study exploring factors influencing smoking cessation among adults with major depressive disorder (MDD), this project reexamined data from the same trial to accomplish the following two main objectives: 1) to identify baseline variables that may moderate the impact of behavioral treatments on end-of-treatment (EOT) smoking abstinence, and 2) to evaluate baseline variables as predictors of abstinence outcomes, accounting for the effects of both behavioral treatments and pharmacotherapy.

Methods: Missing data was imputed with MICE. For each imputed data set, we applied logistic regression to model the binary smoking abstinence outcome and implemented Lasso and Elastic Net Regression with 10-fold cross-validation to identify moderating and predictive variables. Interaction terms were carefully incorporated, especially for factors hypothesized to moderate behavioral activation effects. *Objective 1* included a comprehensive range of baseline variables and interaction terms with BA, while *Objective 2* focused solely on baseline predictors without interactions to assess their independent predictive effects. Calibration and discrimination metrics were used to evaluate model performance.

Results and conclusion: Our analysis identified key moderators and predictors of smoking cessation among individuals with MDD. Interactions between Behavioral Activation (BA) and Nicotine Metabolism Ratio (NMR) showed that faster nicotine metabolism improved BA's effectiveness, while higher nicotine dependence (FTCD score) reduced it. Readiness to quit also moderated BA's effects, with higher readiness linked to better outcomes. Current MDD status influenced BA efficacy, highlighting the need for tailored interventions for those with active symptoms. Independent predictors included NMR, FTCD score, readiness, education, and race, with higher NMR and readiness increasing abstinence odds, while higher FTCD scores and racial disparities posed barriers. Elastic Net provided the best calibration (lowest Brier score and calibration error), while Lasso offered slightly better discrimination with higher AUC values. These findings underscore the need for personalized, equitable smoking cessation interventions for individuals with MDD.

INTRODUCTION

This regression analysis project, in collaboration with Dr. George Papandonatos from Brown's Department of Biostatistics, seeked to explore factors influencing smoking cessation among adults with major depressive disorder (MDD). Individuals with MDD often demonstrate stronger nicotine dependence and experience more challenging withdrawal symptoms than those without MDD. While varenicline is a proven aid for smoking cessation, addressing psychological factors associated with MDD-related smoking behaviors might also improve quit rates in this population.

Dr. Papandonatos' previous randomized, placebo-controlled study (Hitsman et al. (2023)), which included 300 adult smokers with either current or past MDD, employed a 2x2 factorial design and compared behavioral activation for smoking cessation (BASC) against standard treatment (ST) and varenicline versus placebo. The multi-center study found no significant differences in abstinence outcomes between BASC and ST, regardless of varenicline use. However, varenicline significantly outperformed placebo at the 27-week follow-up, achieving a cessation of 16.2% compared to 7.5% for the placebo group.

Table 1: Participant characteristics by overall sample and treatment arm

Characteristic	**Overall** N = 300	**BASC+placebo** N = 68	**BASC+varenicline** N = 83	**ST+placebo** N = 68	**ST+varenicline** N = 81	**p-value**
Demographics						
Age (years)	50.0 (12.6)	50.7 (13.5)	50.3 (13.2)	50.3 (10.8)	48.7 (12.7)	0.7
Sex (female)	165 (55%)	38 (56%)	44 (53%)	39 (57%)	44 (54%)	>0.9
Race						
Non-Hispanic white	105 (35%)	24 (35%)	34 (41%)	22 (32%)	25 (31%)	
Black	157 (52%)	37 (54%)	37 (45%)	40 (59%)	43 (53%)	
Hispanic	16 (5.3%)	4 (5.9%)	3 (3.6%)	4 (5.9%)	5 (6.2%)	
Other	22 (7.3%)	3 (4.4%)	9 (11%)	2 (2.9%)	8 (9.9%)	
Income						0.8
Less than $20,000	110 (37%)	25 (37%)	30 (37%)	26 (38%)	29 (36%)	
$20,000–35,000	68 (23%)	16 (24%)	17 (21%)	14 (21%)	21 (26%)	
$35,001–50,000	46 (15%)	8 (12%)	13 (16%)	14 (21%)	11 (14%)	
$50,001–75,000	38 (13%)	12 (18%)	12 (15%)	8 (12%)	6 (7.5%)	
More than $75,000	35 (12%)	6 (9.0%)	10 (12%)	6 (8.8%)	13 (16%)	
Education						
Grade school	1 (0.3%)	1 (1.5%)	0 (0%)	0 (0%)	0 (0%)	
Some high school	16 (5.3%)	3 (4.4%)	7 (8.4%)	2 (2.9%)	4 (4.9%)	
High school graduate or GED	76 (25%)	23 (34%)	15 (18%)	11 (16%)	27 (33%)	
Some college/technical school	116 (39%)	22 (32%)	32 (39%)	38 (56%)	24 (30%)	
College graduate	91 (30%)	19 (28%)	29 (35%)	17 (25%)	26 (32%)	
Smoking						
Cigarettes per day	15.1 (7.9)	15.6 (9.1)	15.5 (8.5)	15.0 (7.2)	14.4 (6.6)	>0.9
FTCD score	5.2 (2.1)	5.3 (2.0)	5.1 (2.3)	5.4 (2.1)	5.2 (2.1)	0.7
Smoking with 5 mins of waking up (Yes)	138 (46%)	32 (47%)	33 (40%)	35 (51%)	38 (47%)	0.5
BDI score	18.7 (11.5)	19.0 (12.3)	18.0 (10.6)	18.5 (10.8)	19.5 (12.2)	>0.9
Cigarette reward value	7.2 (3.7)	7.4 (3.8)	7.2 (3.9)	7.0 (3.7)	7.1 (3.5)	>0.9
Pleasurable Events Scale (substitute reinforcers)	22.6 (19.6)	23.2 (20.3)	22.9 (19.0)	20.8 (20.1)	23.4 (19.5)	0.6
Pleasurable Events Scale (complementary reinforcers)	25.4 (19.4)	27.7 (21.5)	22.4 (17.0)	27.4 (19.9)	25.0 (19.4)	0.3
Nicotine Metabolism Ratio	0.4 (0.2)	0.3 (0.2)	0.4 (0.2)	0.4 (0.3)	0.4 (0.2)	>0.9
Exclusive mentholated cigarette user (Yes)	178 (60%)	40 (59%)	48 (59%)	43 (64%)	47 (58%)	0.9
Readiness to quit smoking	6.8 (1.2)	6.8 (1.4)	6.7 (1.2)	7.0 (1.3)	6.7 (1.1)	0.6
Psychiatric						
Anhedonia	2.2 (3.2)	2.2 (3.2)	2.3 (3.1)	2.5 (3.4)	2.1 (3.0)	0.8
Other lifetime DSM-5 diagnosis (Yes)	133 (44%)	35 (51%)	30 (36%)	28 (41%)	40 (49%)	0.2
Antidepressant medication (Yes)	82 (27%)	28 (41%)	24 (29%)	15 (22%)	15 (19%)	0.013
Current (and past) MDD vs past MDD only (Yes)	147 (49%)	32 (47%)	40 (48%)	31 (46%)	44 (54%)	0.7

[1] n (%); Mean (SD)

[2] Kruskal-Wallis rank sum test; Pearson's Chi-squared test

Building on the original study, this project reexamined data from the same trial to accomplish the following two main objectives: 1) to identify baseline variables that may moderate the impact of behavioral treatments on end-of-treatment (EOT) smoking abstinence, and 2) to evaluate baseline variables as predictors of smaking cessation, accounting for the effects of both behavioral treatments and pharmacotherapy.

DATA

Data Overview

The study population of the RCT consists of 300 adult smokers with a history of current or past MDD. 1) Sociodemographic, 2) smoking-related, and 3) psychiatric characteristics of the participants were collected at baseline and displayed in **Table 1** above. Demographics are age, sex, race, income, education; the smoking behaviors/measurements include key statistics like Fagerstrom Test for Cigarette Dependence (FTCD) score, Nicotine Metabolism Ratio (NMR), and indicator for exclusive Mentholated cigarette user; and psychiatric disgnotic and treatment history . The participants were randomized into four treatment arms: BASC with placebo (`BASC+placebo`), BASC with varenicline (`BASC+Varenicline`), ST with placebo (`ST+placebo`), and ST with varenicline (`ST+Varenicline`).

Overall, the randomization process appears successful, as key variables such as demographic characteristics, smoking intensity (cigarettes per day), and psychiatric measures like the DSM-5 diagnosis and anhedonia scores showed similar distributions across the four treatment arms with p-value much greater than 5%, suggesting that participant characteristics were well-balanced across treatment arms, as expected in an RCT. This balance across groups reinforces the original study's internal validity, as any differences in outcomes can be more confidently attributed to the interventions rather than baseline differences in participant characteristics.

Regarding the variable education, as shown in **Table 1**, two of the lowest education levels— grade school and some high school— had very few participant counts (1 and 16, respectively). To ensure adequate sample size and meaningful comparisons across education categories, we merged the first three levels (grade school, some high school, and high school graduate or GED) into a single category, considered as "High School and Below." This aggregation would improve the interpretability of the data by creating a more substantial subgroup and reducing variability, allowing for more reliable statistical analyses.

To explore the potential interaction effects between variables in the dataset, race versus exclusive menthol cigarette use would be a good example. As shown in the contingency table (**Table 2**), the distribution across racial groups is not balanced, as indicated by the significant Chi-square test statistic, suggesting that certain racial groups (Black in this case) might have a stronger inclination towards exclusive menthol use. This imbalance in distribution underscores the importance of considering racial demographics in our analysis, as they may interact with other smoking-related behaviors or biological factors.

Table 2: Contingency Table of Race vs. Only Menthol Use with Chi-Square Test Result

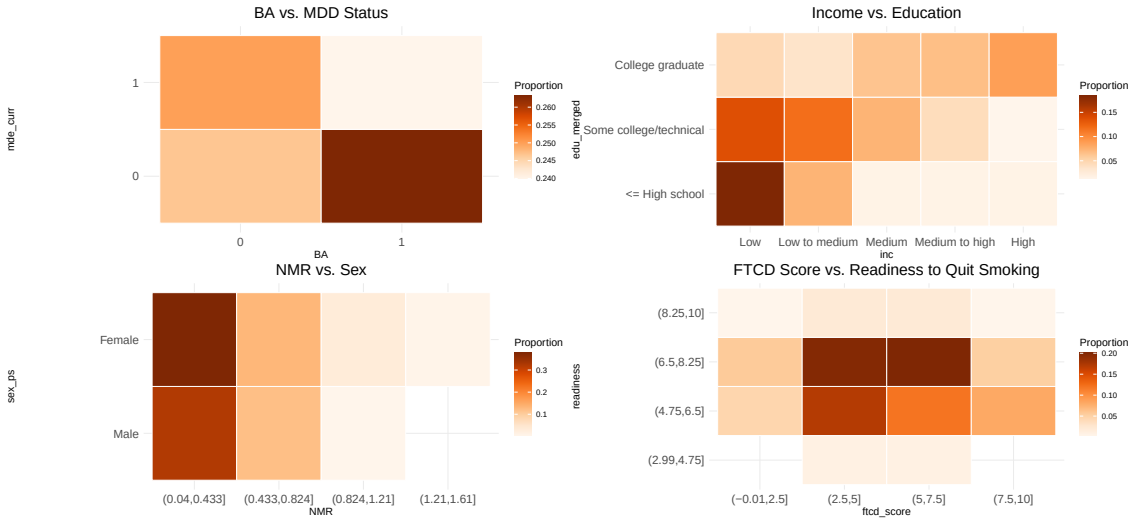
	Non-Menthol-Only	Menthol-Only
Non-Hispanic white	72	32
Black	27	129
Hispanic	12	4
Other	9	13

Note:
Chi-Square Statistic 78.49 p-value 0.0000

Similarly, the exploratory interaction heatmaps showcase four selected examples of interactions that would be explored further in our analysis. These examples illustrate potential interactions that may make biological or statistical sense in the context of smoking cessation. For instance, the interaction between behavioral activation and MDD diagnosis could be central to this study's focus on using MDD-targeted treatments

to aid smoking cessation. The interaction between income and education could reflect socioeconomic influences on smoking behaviors, while cigarette dependence (measured by FTCD score) and readiness to quit smoking might reveal motivational factors in cessation attempts. Additionally, nicotine metabolism ratio (NMR) versus sex could highlight a possible biological interaction that may affect the body's response to nicotine between genders. The color gradients in **Figure 1** indicating proportion appear to confirm that there might be potential synergies between these variables. This visual evidence supported the inclusion of these interactions for further analysis, as they may capture meaningful relationships that influence smoking cessation. In the methods section, we would discuss a more comprehensive rationale for including a certain set of interaction terms, categorizing some as moderators specifically for BA effects and others as covariates, to capture these multidimensional influences on the smoking abstinence outcome.

Figure 1: Exploratory Interaction Heatmaps−− Moderator and Covariate Examples



Data Missingness and Imputation

Table 3: Summary of Missing Values

Variable	Number	Percent
Participants with any missingness	59	19.67%
Nicotine Metabolism Ratio	21	7.00%
Cigarette reward value at baseline	18	6.00%
Baseline readiness to quit smoking	17	5.67%
Income	3	1.00%
Anhedonia	3	1.00%
Exclusive Mentholated Cigarette User	2	0.67%
FTCD score at baseline	1	0.33%

We performed the Little's MCAR test on the data, and it gave a $p-$value of ~0.22, which rejects the null hypothesis that the missingness of the dataset is not completely at random. However, the overall missingness in the dataset is approximately 20% (59/300), way exceeding the commonly accepted 5% threshold for data completeness. This level of missing data could compromise the validity of the analysis if left unaddressed, as it may lead to biased results or loss of valuable information. To preserve the dataset's integrity and maintain statistical power, we opted to impute the missing values with the Multiple Imputation by Chained Equations (MICE) method. It allows for flexible handling of various data types and patterns of missingness, thereby maximizing the use of available information and improving the robustness of the analyses. Specifically, we set the number of imputations to $m = 5$, used predictive mean matching, and iterated the imputation process for a maximum of 50 iterations to ensure reliable imputations.

METHODS

Variable Inclusion Criteria for Full Models

For *Objective 1*, psychotherapy (`BA`) is the primary predictor of interest, while pharmacotherapy (`Var`) is treated as a control variable. Thus, we ensured to include both of the treatments in all models as consistent factors. The major goal of *Objective 1* is to explore how various baseline characteristics might moderate the effect of behavioral treatment on smoking cessation. To achieve this, we included a range of interaction terms between `BA` and key sociodemographic, smoking-related, and psychiatric variables as potential moderators. These interaction terms allowed us to examine how different participant characteristics may influence the effectiveness of behavioriol activation in promoting abstinence. However, we excluded certain terms, such as interactions between `BA` and variables like indicator of smoking with 5 mins of waking up (`ftcd.5mins`) or cigarettes per day (`cpd_ps`), because these measures are components of the broader FTCD score (`ftcd_score`) and thus would provide redundant information. By prioritizing unique, non-overlapping terms, we aimed to create a comprehensive yet parsimonious model.

In addition to `BA` interaction terms as potential moderators, we included other interaction terms involving `Var` and various baseline characteristics as covariates. We believed these covariate interaction terms shall account for known associations that could impact treatment outcomes. For instance, as previously discussed, there was a recognized relationship between race and menthol-only cigarette use, which could affect smoking cessation success. Similarly, the synergy between factors like baseline readiness to quit smoking (`readiness`), Nicotine Metabolism Ratio (`NMR`), and MDD history (`mde_curr`) could have critical impacts on smoking behavior and mental health status and have plausible biological or statistical interactions with other sociodemographic and smoking-related variables. These interactions reflect meaningful covariate effects that contribute to a more nuanced understanding of how different factors influence treatment outcomes, enabling a more robust analysis of predictors and moderators in the context of smoking cessation for adults with MDD.

For *Objective 2*, the focus shifts to using baseline variables as predictors of smoking cessation outcomes, rather than as covariates or moderators. In this context, `BA` and `Var` are included as control variables across all models to account for the effects of behavioral and pharmacotherapy interventions. However, our goal here is to examine the predictive power of baseline characteristics independently, so we did not include any interaction terms to simplify the model by isolating the main effects of each baseline variable.

For the inclusion of such covariates, we also consulted previous literature (West et al. (2018), Rice et al. (1996)). Particularly, West et al. (2018) was exploring the efficacy of smoking cessation treatments and predictors of smoking abstinence, which is highly relevant to our analysis. And they provided insights on potential factors such as race, mood disorder, anxiety, and even treatment $\times$ covariate interactions.

Variable Selection Methods

Each of the five imputed datasets (N=300) was split into a training set (70%) and a test set (30%) to facilitate model evaluation and to ensure that the model's performance metrics would generalize beyond the training data. This approach allowed us to assess the predictive accuracy and robustness of the selected models on an independent dataset, which is critical for reducing overfitting and improving the reliability of our findings.

1) Lasso Regression

Lasso regression, or Least Absolute Shrinkage and Selection Operator, is a regularization technique that applies an L1 penalty to the regression coefficients. This penalty has the unique property of shrinking some coefficients exactly to zero, which effectively performs variable selection. Lasso is particularly advantageous in situations with high-dimensional data where there are many predictors but only a subset is truly relevant to the outcome. By setting irrelevant coefficients to zero, Lasso simplifies the model, improves interpretability, and reduces overfitting.

In our analysis, Lasso with with 10-fold cross-validation was used to identify a parsimonious set of predictors

that best explained the outcome. Cross-validation was employed to select the optimal lambda, ensuring that the model achieved a balance between sparsity (variable selection) and predictive accuracy. While Lasso works well for sparse datasets, it can struggle when predictors are highly correlated, as it tends to arbitrarily select one variable from a group of correlated variables, which may not always lead to the most stable solution.

2) Elastic Net Regression

Elastic Net regression is a regularization and variable selection technique that incorporates both Lasso (L1 penalty) and Ridge (L2 penalty) regression methods. It addresses some limitations of Lasso, particularly in situations where predictors are highly correlated. By combining L1 and L2 penalties, Elastic Net encourages a grouping effect where correlated variables tend to be selected or excluded together, leading to more stable and reliable models.

In our modeling, Elastic Net with 10-fold cross-validation was utilized to take advantage of both variable shrinkage and selection properties of Lasso and the grouping effect of Ridge regression. This method helps handle multi-collinearity among predictors while performing variable selection. We used cross-validation to determine the optimal value of the regularization parameter lambda, specifically selecting the lambda that minimizes the cross-validation error (λ_{min}). This approach ensures that the model achieves the best predictive performance on unseen data by effectively balancing bias and variance.

Note: In our pre-analysis plan, Ridge regression was also expected to be used as a comparison. However, since Ridge retains all variables regardless of their contribution, and our model includes numerous interaction terms, the resulting complexity and potential overfitting led us to exclude Ridge from the actual analysis.

Performance Metrics (Calibration and Discrimination)

To evaluate model performances, we used a combination of calibration and discrimination measures. Calibration plots with error bars and LOESS smoothing allowed us to visually assess the agreement between predicted probabilities and observed outcomes, indicating how well-calibrated the model is across different probability levels. The addition of error bars provides insights into the variability of predictions, while LOESS smoothing offers a flexible fit to better capture trends in calibration. The ROC curve evaluates the model's discrimination ability, reflecting its capability to distinguish between positive and negative outcomes. Furthermore, quantitative metrics such as Brier score and calibration error were included in tables (**Table 5** and **Table 7**) to provide objective assessments of model accuracy and calibration. Brier score combines both calibration and sharpness of probability estimates, while calibration error specifically measures the deviation between predicted and observed probabilities, providing complementary insights into model performance beyond visual assessments.

RESULTS

Objective 1: Baseline Characteristics as Potential Moderators

To facilitate interpretation, we exponentiate the coefficients ($exp(\beta)$) to express them as odds ratios, which indicate the multiplicative change in the odds of the outcome for each one-unit increase in a predictor, holding other variables constant. The exploration of moderators influencing the effect of Behavioral Activation (BA) on smoking cessation outcomes revealed several key findings, as detailed in **Figure 2** and **Table 4**. As shown in **Table 4**, several variables were identified as key moderators/covariates of smoking cessation outcomes through their consistent selection across all five imputed datasets. The main treatement behavioral Activation (BA) was negatively associated with smoking cessation (Average OR: 0.7734 for Lasso and 0.7983 for Elastic Net), suggesting a nuanced effect of BA on abstinence outcomes. Importantly, the interaction between BA and NMR was selected once in both models, with a strong positive effect (Average OR: 2.2782 for Lasso and 1.7530 for Elastic Net), indicating the potential moderator effect that faster nicotine metabolism amplifies BA's effectiveness.

Another key covariate, nicotine dependence measured by FTCD score, was consistently selected in Elastic Net models (Selection Times = 5), with a negative association (Average OR: 0.9634), suggesting that higher dependence diminishes the efficacy of BA. Similarly, the interaction between FTCD score and readiness to quit was selected in all Elastic Net models, reinforcing the role of motivation in enhancing treatment outcomes. Individuals with higher readiness to quit were more likely to benefit from BA, as seen in the reduced odds of abstinence for this interaction term (Average OR: 0.9871).

For the pharmacotherapy arm, Varenicline consistently showed a strong positive association with abstinence (Selection Times= 5 for both models, Average OR: 4.8757 for Lasso and 4.8444 for Elastic Net), emphasizing its efficacy as a cessation aid.

Table 4: Summary of Average Coefficients and Odds Ratios for Potential Moderator Effects across Model Selection Methods

	Lasso			Elastic Net		
Variable	Selection Times	Average Coef	OR	Selection Times	Average Coef	OR
BA1	5	-0.2569	0.7734	5	-0.2253	0.7983
BA1:NMR	1	0.8234	2.2782	1	0.5613	1.7530
NMR				1	0.0984	1.1034
Only.Menthol1:raceOther				1	-0.1101	0.8957
Var1	5	1.5843	4.8757	5	1.5778	4.8444
cpd_ps:NMR	1	0.0136	1.0137	1	0.0156	1.0157
edu_merged2:Only.Menthol1	1	-0.2999	0.7409	1	-0.2847	0.7522
ftcd_score	1	-0.0126	0.9875	5	-0.0373	0.9634
ftcd_score:raceBlack				1	-0.0027	0.9973
ftcd_score:readiness	5	-0.0227	0.9776	5	-0.0130	0.9871
ftcd_score:sex_ps2				1	-0.0080	0.9920
inc5:Only.Menthol1	1	-0.1822	0.8334	1	-0.3031	0.7385
raceBlack	1	-0.0105	0.9896	1	-0.0478	0.9534
raceHispanic:Var1	1	-0.6492	0.5225	1	-0.7145	0.4894
raceOther				1	-0.1316	0.8767
raceOther:Var1	1	-0.4319	0.6493	1	-0.2821	0.7542

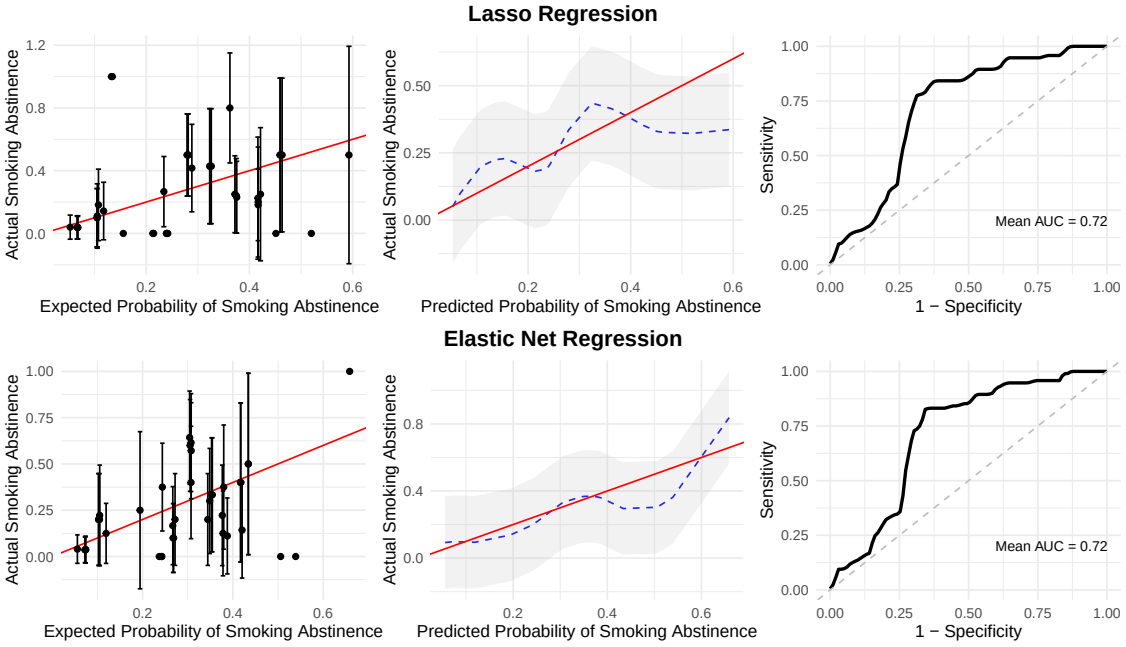
Note: Selection times indicate the number of MICE datasets (out of 5) in which a variable was selected. For example, a selection time of 5 indicates that the variable was selected in all imputed datasets.

Model performance for *Aim 1*, summarized in **Table 5**, showed that Elastic Net provided slightly superior calibration, with a Brier score of 0.1533 and a calibration error of 0.0660. These metrics indicate that Elastic Net predictions were more closely aligned with observed outcomes. However, Lasso achieved slightly better discrimination, with an AUC of 0.7177 compared to 0.7165 for Elastic Net. Both models identified consistent moderators, although Elastic Net's better calibration suggests it may be more suitable for estimating individualized probabilities of success.

Table 5: Calibration and Discrimination Metrics for Moderator Effect Modeling

	Elastic Net	Lasso
Brier score	0.1533	0.1531
Calibration error	0.0660	0.0495
AUC	0.7165	0.7177
Threshold	0.2672	0.2628
Specificity	0.6571	0.6657
Sensitivity	0.8316	0.8105

Figure 2: Calibration Plots with Error Bars and LOESS and ROC Curves (Moderator Effects)



The calibration and ROC curves in **Figure 2** illustrate the performance characteristics of Elastic Net and Lasso regression models. The calibration plots for Elastic Net closely align with the ideal diagonal line, indicating that its predicted probabilities for abstinence outcomes are well-calibrated and closely match the observed outcomes across probability levels. The LOESS curve for Elastic Net also tracks the ideal line with only minor deviations, further demonstrating its effective calibration. In contrast, Lasso exhibits slightly greater misalignment in its calibration plot, particularly at the extremes of the probability range, suggesting minor over- or under-confidence in some predictions. The ROC curves in **Figure 2** reflects the extremely similar discrimination ability between Lasso and Elastic Net, as they achieved an AUC of 0.7177 and 0.7165. While the difference in AUC is minimal, Elastic Net maintains a better balance between calibration and discrimination, making it particularly effective for estimating probabilities and modeling moderator effects with high-dimensional data.

Objective 2: Baseline Characteristics as Potential Predictors

Table 6: Summary of Average Coefficients and Odds Ratios for Potential Predictor Effects across Model Selection Methods

	Lasso			Elastic Net		
Variable	Selection Times	Average Coef	OR	Selection Times	Average Coef	OR
BA1	5	-0.1757	0.8389	5	-0.1773	0.8375
NMR				1	0.1893	1.2084
Var1	5	1.6711	5.3182	5	1.6710	5.3172
ftcd_score	4	-0.0766	0.9263	5	-0.0642	0.9378
mde_curr1	4	-0.0823	0.921	5	-0.0861	0.9175
raceHispanic	4	-0.4305	0.6502	5	-0.3776	0.6855

The identification of baseline predictors independent of interaction terms highlighted several significant factors, as shown in **Table 6**. Nicotine Metabolism Ratio (NMR) emerged as a positive predictor of smoking cessation success, selected in both Lasso and Elastic Net models (Elastic Net OR: 1.2084). This finding suggests that individuals with faster nicotine metabolism may have a greater likelihood of managing withdrawal symptoms and benefiting from treatment interventions. In contrast, FTCD score, a measure of nicotine

dependence, was consistently negatively associated with cessation outcomes (Elastic Net OR: 0.9378). This indicates that higher nicotine dependence reduces the odds of successful smoking cessation, reflecting the challenges posed by severe addiction.

Behavioral Activation as the main treetment remained an important variable in both models, with consistent negative associations across all five imputations (Elastic Net OR: 0.8375). This suggests that BA alone may not be sufficient to drive abstinence, reinforcing the need for combined approaches, such as pharmacotherapy or motivational enhancement. Varenicline was the strongest positive predictor in both models (Elastic Net OR: 5.3172), further validating its role as an effective pharmacological aid for smoking cessation.

Race and ethnicity also played a significant role, with Hispanic individuals demonstrating substantially lower odds of cessation (Elastic Net OR: 0.6855), suggesting systemic barriers or differential responses to treatment. Current MDD status (`mde_curr1`) was another notable predictor, with a negative association with abstinence (Elastic Net OR: 0.9175). This finding highlights the importance of addressing depressive symptoms to improve cessation outcomes.

Together, these results underscore the importance of biological, psychological, and demographic factors in smoking cessation. Tailored interventions that address nicotine dependence, mental health conditions, and systemic disparities, particularly among minority populations, are essential for improving outcomes.

Table 7: Calibration and Discrimination Metrics for Predictor Effect Modeling

	Elastic Net	Lasso
Brier score	0.1509	0.1517
Calibration error	0.0383	0.0339
AUC	0.7420	0.7323
Threshold	0.2272	0.1067
Specificity	0.6543	0.5571
Sensitivity	0.7684	0.8632

In terms of model performance, shown in **Table 7**, Elastic Net demonstrated superior calibration for baseline predictors, achieving a Brier score of 0.1509 and a calibration error of 0.0383, as shown in Table 3. Lasso, on the other hand, achieved slightly higher discrimination, with an AUC of 0.7323 compared to 0.7315 for Elastic Net. These complementary results suggest that Elastic Net is still slightly better suited for applications where accurate probability estimates are critical, while Lasso may be more effective for classification tasks.

CONCLUSION

This study investigated the predictors and moderators of smoking cessation outcomes among individuals with MDD, focusing on the effects of behavioral activation. As shown in **Table 4**, BA's effectiveness varied significantly depending on baseline characteristics. Nicotine Metabolism Ratio (NMR) was identified as a key moderator, with a strong positive interaction indicating that individuals with faster nicotine metabolism (higher NMR) were more likely to benefit from BA. Similarly, readiness to quit smoking positively moderated BA's effects, with individuals who were more motivated to quit showing improved abstinence outcomes. In contrast, individuals with higher nicotine dependence, as measured by FTCD score, experienced reduced efficacy of BA. Current versus past MDD status also influenced BA's effects, suggesting that addressing active depressive symptoms is critical for maximizing the benefits of behavioral interventions. These findings underscore the need to tailor smoking cessation treatments to individual characteristics to improve outcomes.

Baseline predictors, summarized in **Table 6**, highlight the significant role of biological, psychological, and demographic factors in smoking cessation outcomes. Nicotine Metabolism Ratio (NMR) emerged as a positive predictor of abstinence, indicating that faster nicotine metabolism is associated with higher odds of cessation. Conversely, higher FTCD scores were consistently associated with poorer outcomes, reflecting the challenges

posed by nicotine dependence. Varenicline was the strongest predictor of cessation success, with individuals receiving this pharmacological treatment demonstrating significantly higher odds of quitting. Current MDD status was negatively associated with cessation, suggesting that depressive symptoms may hinder abstinence efforts. Finally, race and ethnicity were also significant predictors, with Hispanic individuals showing reduced odds of cessation, likely reflecting systemic disparities.

As shown in **Table 5** and **Table 7*, Elastic Net provided superior calibration metrics, including the lowest Brier score and calibration error, making it well-suited for probability estimation. Lasso, in contrast, demonstrated slightly better discrimination, as reflected in higher AUC values, highlighting its utility for classification tasks.

In conclusion, the results underscore the importance of tailoring smoking cessation interventions to individual characteristics, particularly for individuals with MDD. Behavioral activation can be an effective strategy, but its success depends on factors such as nicotine metabolism, dependence, motivation, and mental health status. Future research should focus on validating these findings in larger, more diverse populations and integrating personalized approaches into cessation programs. Addressing disparities in access and outcomes, particularly among racial and ethnic minorities, will also be essential for ensuring equitable and effective treatment.

LIMITATIONS

A primary limitation of this analysis is the small sample size, with only 300 observations in each imputed dateset split between training and test sets. This limited data size, combined with the inclusion of multiple interaction terms, exacerbates the "small n, big p" problem, where the number of predictors may outstrip the available data points, potentially leading to overfitting and instability in model estimates. Given the small dataset, further research with larger samples and significance testing of individual predictors would enhance the reliability and generalizability of these conclusions.

Additionally, the predictor selection process focused on optimizing overall model performance metrics, such as calibration and discrimination, rather than testing the individual significance of each variable. As a result, no formal p-values or statistical tests were assigned to assess the significance of individual variables. This approach limits our ability to make definitive claims about the significance of individual variables as verified moderators or predictors and instead relies on their contribution to the model's overall predictive performance.

Consent, Data, and Code Availability

Primary data were provided by Dr. George Papandonatos from the Department of Biostatistics at Brown University. The original data cannot be shared directly for privacy. Replication scripts are available at https://github.com/YanweiTong-Iris/PHP2550-ProjectPortfolio/tree/main/Project%202.

Reference

Hitsman, B., Papandonatos, G. D., Gollan, J. K., Huffman, M. D., Niaura, R., Mohr, D. C., Veluz-Wilkins, A. K., Lubitz, S. F., Hole, A., Leone, F. T., and others (2023), "Efficacy and safety of combination behavioral activation for smoking cessation and varenicline for treating tobacco dependence among individuals with current or past major depressive disorder: A 2× 2 factorial, randomized, placebo-controlled trial," *Addiction*, Wiley Online Library, 118, 1710–1725.

Rice, V. H., Templin, T., Fox, D. H., Jarosz, P., Mullin, M., Seiggreen, M., and Lepczyk, M. (1996), "Social context variables as predictors of smoking cessation." *Tobacco control*, BMJ Publishing Group Ltd, 5, 280–285.

West, R., Evins, A. E., Benowitz, N. L., Russ, C., McRae, T., Lawrence, D., St Aubin, L., Krishen, A., Maravic, M. C., and Anthenelli, R. M. (2018), "Factors associated with the efficacy of smoking cessation treatments and predictors of smoking abstinence in EAGLES," *Addiction*, Wiley Online Library, 113, 1507–1516.

Code Appendix

```
# to prevent scientific notation
options(scipen=999)

# Set up knit environment
knitr::opts_chunk$set(echo = FALSE,
                      message = FALSE,
                      warning = FALSE,
                      error = FALSE)

# Load necessary packages
library(tidyverse)
library(kableExtra)
library(knitr)
library(ggplot2)
library(gridExtra)
library(grid)
library(naniar)
library(gtsummary)
library(gt)
library(patchwork)
library(knitcitations)
library(mice)
library(glmnet)
library(pROC)
library(MASS)
library(leaps)
library(RColorBrewer)
library(cowplot)
library(caret)
# Define data path and import data
data_path = "/Users/yanweitong/Documents/PHP2550-Data/Project2"
data = read.csv(paste0(data_path, "/project2.csv"))

mcar_test(data)

# Data preprocessing
data = data  %>%
  # create race variable
  mutate(race = factor(case_when(
    NHW == 1 ~ "Non-Hispanic white",
    Black == 1 ~ "Black",
    Hisp == 1 ~ "Hispanic",
    TRUE ~ "Other"  # Handle cases where none of the above conditions are met
  ), levels = c("Non-Hispanic white", "Black", "Hispanic", "Other"))) %>%
  # create treatment categories
  mutate(treatment_cat = factor(case_when(BA == 1 & Var == 0 ~ "BASC+placebo",
                                  BA == 0 & Var == 0 ~ "ST+placebo",
                                  BA == 1 & Var == 1 ~ "BASC+varenicline",
                                  BA == 0 & Var == 1 ~ "ST+varenicline"))) %>%
  # factorize categorical/ordinal variables
  mutate(
    abst = factor(abst),
```

```
    Var = factor(Var),
    BA = factor(BA),
    sex_ps = factor(sex_ps),
    NHW = factor(NHW),
    Black = factor(Black),

    ftcd.5.mins = factor(ftcd.5.mins),
    otherdiag = factor(otherdiag),
    antidepmed = factor(antidepmed),
    mde_curr = factor(mde_curr),
    Only.Menthol = factor(Only.Menthol),
    edu = factor(edu, levels = c(1, 2, 3, 4, 5)),
    inc = factor(inc, levels = c(1, 2, 3, 4, 5))
  ) %>%
  # make integers numeric
  mutate(across(
    .cols = where(is.integer) & !all_of("id"),
    .fns = as.numeric
  ))
# for sub-tab purpose
table1_data = data %>%
  mutate(
    Demographics = NA,
    Smoking = NA,
    Psychiatric = NA
  ) %>%
  mutate(edu = factor(edu, levels = c(1, 2, 3, 4, 5),
                labels = c("Grade school",
                           "Some high school",
                           "High school graduate or GED",
                           "Some college/technical school",
                           "College graduate")),
    inc = factor(inc, levels = c(1, 2, 3, 4, 5),
                labels = c("Less than $20,000",
                           "$20,000-35,000",
                           "$35,001-50,000",
                           "$50,001-75,000",
                           "More than $75,000")))

table1_data %>%
  dplyr::select(
    treatment_cat,
    Demographics,
    age_ps,
    sex_ps,
    race,
    inc,
    edu,
    Smoking,
    cpd_ps,
    ftcd_score,
    ftcd.5.mins,
    bdi_score_w00,
```

```
  crv_total_pq1,
  hedonsum_n_pq1,
  hedonsum_y_pq1,
  NMR,
  Only.Menthol,
  readiness,
  Psychiatric,
  shaps_score_pq1,
  otherdiag,
  antidepmed,
  mde_curr
) %>%
tbl_summary(
  statistic = list(all_continuous() ~ c("{mean} ({sd})"),
                   all_categorical() ~ "{n} ({p}%)"),
  by = treatment_cat,
  digits = all_continuous() ~ 1,
  missing = "no",
  type = list(
    age_ps ~ "continuous",
    sex_ps ~ "dichotomous",
    race ~ "categorical",
    inc ~ "categorical",
    edu ~ "categorical",
    cpd_ps ~ "continuous",
    ftcd_score ~ "continuous",
    ftcd.5.mins ~ "dichotomous",
    bdi_score_w00 ~ "continuous",
    crv_total_pq1 ~ "continuous",
    hedonsum_n_pq1 ~ "continuous",
    hedonsum_y_pq1 ~ "continuous",
    NMR ~ "continuous",
    Only.Menthol ~ "dichotomous",
    readiness ~ "continuous",
    shaps_score_pq1 ~ "continuous",
    otherdiag ~ "dichotomous",
    antidepmed ~ "dichotomous",
    mde_curr ~ "dichotomous"
  ),
  label = list(
    age_ps = "Age (years)",
    sex_ps = "Sex (female)",
    race = "Race",
    inc = "Income",
    edu = "Education",
    cpd_ps = "Cigarettes per day",
    ftcd_score = "FTCD score",
    ftcd.5.mins = "Smoking with 5 mins of waking up (Yes)",
    bdi_score_w00 = "BDI score",
    crv_total_pq1 = "Cigarette reward value",
    hedonsum_n_pq1 = "Pleasurable Events Scale (substitute reinforcers)",
    hedonsum_y_pq1 = "Pleasurable Events Scale (complementary reinforcers)",
    Only.Menthol = "Exclusive mentholated cigarette user (Yes)",
```

```
      readiness = "Readiness to quit smoking",
      NMR = "Nicotine Metabolism Ratio",
      shaps_score_pq1 = "Anhedonia",
      otherdiag = "Other lifetime DSM-5 diagnosis (Yes)",
      antidepmed = "Antidepressant medication (Yes)",
      mde_curr = "Current (and past) MDD vs past MDD only (Yes)"
    ),
    value = list(
      sex_ps ~ "2",
      Only.Menthol ~ "1",
      otherdiag ~ "1",
      antidepmed ~ "1",
      mde_curr ~ "1",
      ftcd.5.mins ~ "1"
    )
  ) %>%
  add_overall() %>%
  add_p() %>%
  modify_header(label ~ "**Characteristic**") %>%
  modify_caption(caption = "Participant characteristics by overall sample and treatment arm") %>%
  # for sub-tab purpose
  modify_table_body(
    ~ .x %>%
      mutate(across(everything(), ~ ifelse(. == "0 (NA%)", "", .)))
  ) %>%
  modify_table_styling(
    rows = label %in% c("Sociodemographics", "Smoking", "Psychiatric"),
    columns = label,
    text_format = "bold"
  )   %>%
  as_kable_extra(
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    format = "latex"
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header", "hold_position", "scale_down"),
    stripe_color = "gray!15",
    font_size = 8
  )
# Create a new variable that contains the 3 first levels of edu
data <- data %>%
  mutate(edu_merged = factor(case_when(
    edu %in% c("1", "2", "3") ~ "1",
    edu == "4" ~ "2",
    edu == "5" ~ "3"
  )))

# Create and display the contingency table between race and menthol useage
table_race_menthol <- table(data$race, data$Only.Menthol)
```

```r
# Perform the chi-square test
chi_square_test <- chisq.test(table_race_menthol)

chi_square_text <- paste0(
  sprintf("Chi-Square Statistic  %.2f", chi_square_test$statistic),
  sprintf("  p-value  %.4f", chi_square_test$p.value)
)

kable(table_race_menthol,
      caption = "Contingency Table of Race vs. Only Menthol Use with Chi-Square Test Result",
      col.names = c("Non-Menthol-Only", "Menthol-Only"),
      row.names = TRUE) %>%
   footnote(chi_square_text,
            footnote_as_chunk = FALSE) %>%
  kable_styling(full_width = F, position = "center", font_size = 9)

# Define function to bin continuous variables and create proportion heatmaps
create_heatmap <- function(data, var1, var2, title, bin_var1 = FALSE,
                           bin_var2 = FALSE, levels_var1 = NULL, levels_var2 = NULL) {
  # Remove NA values for the specified columns
  data <- data %>% drop_na(all_of(c(var1, var2)))

  # Optionally bin continuous variables
  if (bin_var1) data <- data %>%
      mutate(!!sym(var1) := cut(!!sym(var1), breaks = 4))
  if (bin_var2) data <- data %>%
      mutate(!!sym(var2) := cut(!!sym(var2), breaks = 4))

  # Set custom factor levels for better readability
  if (!is.null(levels_var1)) {
    data <- data %>%
      mutate(!!sym(var1) := factor(!!sym(var1), levels = names(levels_var1),
                                   labels = levels_var1))
  }
  if (!is.null(levels_var2)) {
    data <- data %>%
      mutate(!!sym(var2) := factor(!!sym(var2), levels = names(levels_var2),
                                   labels = levels_var2))
  }

  # Calculate proportions
  prop_data <- data %>%
    group_by(!!sym(var1), !!sym(var2)) %>%
    summarise(count = n(), .groups = 'drop') %>%
    mutate(prop = count / sum(count))

  ggplot(prop_data, aes_string(x = var1, y = var2)) +
    geom_tile(aes(fill = prop), color = "white") +
    scale_fill_gradientn(colors = brewer.pal(9, "Oranges"), name = "Proportion") +
    labs(title = title, x = var1, y = var2) +
    theme_minimal(base_size = 8) +  # Set smaller base font size
    theme(
      plot.title = element_text(size = 12, hjust = 0.5),
```

```
      axis.text.x = element_text(size = 9),
      axis.text.y = element_text(size = 9),
      axis.title.y = element_text(vjust = 0.5),
      legend.key.size = unit(0.45, "cm"),
      plot.margin = margin(1, 1, 1, 1)  # Add space around the plot
    )
}

# Create the plots with adjusted text and layout

gender_levels <- c("1" = "Male", "2" = "Female")
income_levels <- c("1" = "Low",
                   "2" = "Low to medium",
                   "3" = "Medium",
                   "4" = "Medium to high",
                   "5" = "High")
edu_levels <- c("1" = "<= High school",
                   "2" = "Some college/technical",
                   "3" = "College graduate")


p1 <- create_heatmap(data, "BA", "mde_curr", "BA vs. MDD Status")
p2 <- create_heatmap(data, var1 = "inc", var2 = "edu_merged", title = "Income vs. Education",
                     levels_var1 = income_levels,
                     levels_var2 = edu_levels)

p3 <- create_heatmap(data, "NMR", "sex_ps", "NMR vs. Sex", bin_var1 = TRUE,
                     levels_var2 = gender_levels)
p4 <- create_heatmap(data, "ftcd_score", "readiness",
                     "FTCD Score vs. Readiness to Quit Smoking",
                     bin_var1 = TRUE, bin_var2 = TRUE)

# Arrange the plots in a grid with a title using cowplot
final_plot <- plot_grid(p1, p2, p3, p4, ncol = 2, align = "hv",
                        rel_widths = c(1, 1), rel_heights = c(1, 1))

# Add a title to the entire grid
title <- ggdraw() +
  draw_label("Figure 1: Exploratory Interaction Heatmaps-- Moderator and Covariate Examples",
             size = 16)

# Combine title and grid plot
plot_grid(title, final_plot, ncol = 1, rel_heights = c(0.1, 1))
# Missingness table
# Calculate missing values for each variable
missing_summary <- data %>%
  summarise(across(everything(), ~ sum(is.na(.)), .names = "{col}")) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Number") %>%
  mutate(Percent = (Number / nrow(data)) * 100) %>%
  filter(Number > 0)  # Exclude variables with 0 missingness

# Define a named vector with old and new names for variables
variable_names <- c(
```

```
  "inc" = "Income",
  "ftcd_score" = "FTCD score at baseline",
  "crv_total_pq1" = "Cigarette reward value at baseline",
  "shaps_score_pq1" = "Anhedonia",
  "NMR" = "Nicotine Metabolism Ratio",
  "Only.Menthol" = "Exclusive Mentholated Cigarette User",
  "readiness" = "Baseline readiness to quit smoking"
)

# Rename variables in the summary table
missing_summary <- missing_summary %>%
  mutate(Variable = recode(Variable, !!!variable_names))


# Calculate total missing values and total missing percentage
total_rows_with_missing <- sum(rowSums(is.na(data)) > 0)
total_rows_with_missing_pct <- (total_rows_with_missing / nrow(data)) * 100


missing_summary <- missing_summary %>%
  arrange(desc(Percent)) %>%
  mutate(Percent = sprintf("%.2f%%", Percent))

# Combine the total missingness row with the summary table
total_missing_row <- tibble(
  Variable = "Participants with any missingness",
  Number = total_rows_with_missing,
  Percent = sprintf("%.2f%%", total_rows_with_missing_pct)
)

missing_summary <- bind_rows(total_missing_row, missing_summary)

# Display the final table
missing_summary %>%
  kable(
    col.names = c("Variable", "Number", "Percent"),
    caption = "Summary of Missing Values"
  ) %>%
  kable_styling(full_width = F, position = "center", font_size = 9)
# Perform MICE imputation
imputed_data <- mice(data, m = 5, method = "pmm", maxit = 50, seed = 46,  printFlag = FALSE)
# To identify the potential interaction terms for moderator effects
train_variables_dummy_include_names <- c(
  "Var1", "BA1", "age_ps", "sex_ps2", "inc2", "inc3",
  "inc4", "inc5", "edu_merged2", "edu_merged3",
  "raceBlack", "raceHispanic", "raceOther",
  "ftcd_score", "ftcd.5.mins1", "bdi_score_w00", "cpd_ps",
  "crv_total_pq1", "hedonsum_n_pq1", "hedonsum_y_pq1",
  "shaps_score_pq1", "otherdiag1", "antidepmed1",
  "mde_curr1", "NMR", "Only.Menthol1",
  "readiness", "Var1:BA1",
  #Behavioral treatment (MAIN)
  "BA1:mde_curr1",
```

```
  "BA1:age_ps", "BA1:sex_ps2",
  "BA1:raceBlack", "BA1:raceHispanic",
  "BA1:raceOther", "BA1:ftcd_score",
  "BA1:shaps_score_pq1","BA1:bdi_score_w00",
  "BA1:otherdiag1", "BA1:antidepmed1",
  "BA1:mde_curr1","BA1:NMR",
  "BA1:Only.Menthol1", "BA1:readiness",
  # Pharmacotherapy
  "Var1:mde_curr1",
  "Var1:age_ps", "Var1:sex_ps2",
  "Var1:raceBlack", "Var1:raceHispanic",
  "Var1:raceOther", "Var1:ftcd_score",
  # Income*Edu
  "inc2:edu_merged2", "inc2:edu_merged3",
  "inc3:edu_merged2", "inc3:edu_merged3",
  "inc4:edu_merged2", "inc4:edu_merged3",
  "inc5:edu_merged2", "inc5:edu_merged3",
  # Readiness to quit
  "Only.Menthol1:readiness",
  "mde_curr1:readiness", "ftcd_score:readiness",
  # FTCD Score
  "sex_ps2:ftcd_score", "raceBlack:ftcd_score",
  "raceHispanic:ftcd_score", "raceOther:ftcd_score",
  "age_ps:ftcd_score",
  # Menthol exclusive
  "sex_ps2:Only.Menthol1", "raceBlack:Only.Menthol1",
  "raceHispanic:Only.Menthol1", "raceOther:Only.Menthol1",
  "inc2:Only.Menthol1", "inc3:Only.Menthol1",
  "inc4:Only.Menthol1", "inc5:Only.Menthol1",
  "edu_merged2:Only.Menthol1", "edu_merged3:Only.Menthol1",
  # NMR
  "sex_ps2:NMR", "age_ps:NMR", "cpd_ps:NMR",
  "NMR:readiness", "ftcd_score:NMR"
)


variable_names <- c("Var", "BA", "age_ps", "sex_ps", "inc", "edu_merged", "race",
                    "ftcd_score", "ftcd.5.mins", "bdi_score_w00", "cpd_ps",
                    "crv_total_pq1", "hedonsum_n_pq1", "hedonsum_y_pq1",
                    "shaps_score_pq1", "otherdiag", "antidepmed", "mde_curr",
                    "NMR", "Only.Menthol", "readiness")

# Helper function to perform model fitting
fit_models <- function(data) {
  # Define predictors and outcome

  outcome <- data$abst
  variables <- data[, variable_names]
  # for Lasso (to break down factors with >2 levels)
  variables_dummy <- model.matrix( ~ 0 + ., data = variables)
  # remove the extra reference group
  variables_dummy <- variables_dummy[, -which(colnames(variables_dummy) ==
                                                "Var0")]
```

```r
# Split into train and test sets
set.seed(58)
train_index <- createDataPartition(outcome, p = 0.7, list = FALSE)
train_data <- data[train_index, ]
train_outcome = train_data$abst
test_data <- data[-train_index, ]
test_outcome = test_data$abst

train_variables_dummy <- variables_dummy[train_index, ]
test_variables_dummy <- variables_dummy[-train_index, ]

# ^2 generates all pairwise interactions
train_variables_dummy_df <- as.data.frame(train_variables_dummy)
train_variables_dummy_full_interactions <- model.matrix( ~ . ^ 2, data = train_variables_dummy_df)

test_variables_dummy_df <- as.data.frame(test_variables_dummy)
test_variables_dummy_full_interactions <- model.matrix( ~ . ^ 2, data = test_variables_dummy_df)

train_variables_dummy_include =
  train_variables_dummy_full_interactions[, train_variables_dummy_include_names]
test_variables_dummy_include =
  test_variables_dummy_full_interactions[, train_variables_dummy_include_names]

# Set penalty factors to enforce keeping Var and BA
# Initialize penalty factors to 1 for all variables
penalty_factors <- rep(1, ncol(train_variables_dummy_include))

# Identify columns corresponding exactly to "Var1" and "BA1" (not their interactions)
var1_col <- grep("^Var1$", colnames(train_variables_dummy_include))
ba1_col <- grep("^BA1$", colnames(train_variables_dummy_include))

penalty_factors[c(var1_col, ba1_col)] <- 0
names(penalty_factors) <- colnames(train_variables_dummy_include)

# Fit Elastic Net model
enet_model <- cv.glmnet(
  as.matrix(train_variables_dummy_include),
  train_outcome,
  penalty.factor = penalty_factors,
  alpha = 0.5,
  family = "binomial",
  nfold = 10
)

# Fit Lasso model (alpha = 1)
lasso_model <- cv.glmnet(
  as.matrix(train_variables_dummy_include),
  train_outcome,
  penalty.factor = penalty_factors,
  alpha = 1,
  family = "binomial",
  nfold = 10
```

```
  )

  list(
    elastic_net = enet_model,
    lasso = lasso_model
  )
}
# Fit models across imputed datasets
aim1_results <- list()
for (i in 1:5) {
  dataset <- complete(imputed_data, action = i)
  aim1_results[[i]] <- fit_models(dataset)
}
# Initialize lists to store coefficients for Elastic Net and Lasso
elastic_net_coefs <- lapply(aim1_results, function(res) coef(res$elastic_net, s = res$elastic_net$lambda
lasso_coefs <- lapply(aim1_results, function(res) coef(res$lasso, s = res$lasso$lambda.min))

# Function to process coefficients for averaging and selection count
process_coefs <- function(coefs_list) {
  # Combine coefficients into a matrix
  coefs_matrix <- do.call(cbind, lapply(coefs_list, function(coef) as.numeric(coef)))
  rownames(coefs_matrix) <- rownames(coefs_list[[1]])

  # Count non-zero selections for each variable
  selection_counts <- rowSums(coefs_matrix != 0)

  # Compute the average of coefficients only when selected (non-zero)
  averaged_coefs <- rowSums(coefs_matrix) / selection_counts
  averaged_coefs[is.na(averaged_coefs)] <- 0  # Handle cases where selection_counts is 0

  # Create result table
  result_table <- data.frame(
    variable = rownames(coefs_matrix),
        SelectionTimes = selection_counts,
    AverageCoefficient = averaged_coefs
  ) %>%
    filter(SelectionTimes > 0)  # Keep only variables selected at least once

  return(result_table)
}

# Process Elastic Net and Lasso coefficients
result_table_enet <- process_coefs(elastic_net_coefs)
result_table_lasso <- process_coefs(lasso_coefs)
# Summary table of coef
large_threshold <- 100

# Calculate OR for each coefficient for each method and rename columns correctly
enet_df <- result_table_enet %>%
  rename(`Selection Times` = SelectionTimes) %>%
  rename(`Elastic Net Average Coef` = AverageCoefficient) %>%
  mutate(`Elastic Net Average OR` = exp(`Elastic Net Average Coef`))
```

```r
lasso_df <- result_table_lasso %>%
  rename(`Selection Times` = SelectionTimes) %>%
  rename(`Lasso Average Coef` = AverageCoefficient) %>%
  mutate(`Lasso OR` = exp(`Lasso Average Coef`))



# Merge all data frames based on variable names
combined_df <- full_join(lasso_df, enet_df, by = "variable") %>%
  mutate(across(where(is.numeric), ~ round(.x, 4))) %>%
  mutate(across(where(is.numeric), ~ ifelse(as.numeric(.) > large_threshold, "*", .)))

# Remove the intercept row
combined_df <- combined_df[combined_df$variable != "(Intercept)", ]

# Standardize interaction terms by sorting them alphabetically
combined_df <- combined_df %>%
  mutate(
    variable = sapply(variable, function(x) {
      terms <- unlist(strsplit(x, ":"))
      if (length(terms) > 1) {
        paste(sort(terms), collapse = ":")
      } else {
        x
      }
    })
  )

# Combine rows with the same standardized interaction term names
combined_df <- combined_df %>%
  group_by(variable) %>%
  summarize(across(everything(), ~ ifelse(is.numeric(.), sum(as.numeric(.), na.rm = TRUE), .))) %>%
  ungroup()

# Replace zeroes and any remaining NAs with an empty space for readability
combined_df <- combined_df %>%
  mutate(across(where(is.numeric), ~ ifelse(. == 0, "", .))) %>%
  replace(is.na(.), " ")

# Display the final combined table with grouped headers
combined_df %>%
  kable(row.names = F,
        col.names = c("Variable", "Selection Times","Average Coef", "OR", "Selection Times", "Average C
        caption = "Summary of Average Coefficients and Odds Ratios for Potential Moderator Effects acros
  add_header_above(c(" " = 1, "Lasso" = 3, "Elastic Net" = 3)) %>%
  kable_styling(full_width = F, position = "center", font_size = 9)

# Initialize lists to store results for each imputation
roc_results_enet <- list()
calib_results_enet <- list()
auc_values_enet <- list()

roc_results_lasso <- list()
```

```r
calib_results_lasso <- list()
auc_values_lasso <- list()

num_cuts <- 10  # Number of bins for calibration

for (i in 1:5) {
  dataset <- complete(imputed_data, action = i)

  # Split the dataset into training and test sets
  set.seed(58)
  train_index <- createDataPartition(dataset$abst, p = 0.7, list = FALSE)
  train_data <- dataset[train_index, ]
  test_data <- dataset[-train_index, ]

  # Prepare predictors and outcome for test set
  test_variables_dummy <- model.matrix(~ 0 + ., data = test_data[, variable_names])
  test_variables_dummy <- test_variables_dummy[, -which(colnames(test_variables_dummy) == "Var0")]
  test_variables_dummy_full_interactions <- model.matrix(~ . ^ 2, data = as.data.frame(test_variables_du
  test_variables_dummy_include <- test_variables_dummy_full_interactions[, train_variables_dummy_include
  test_outcome <- test_data$abst

  # Predict probabilities for Elastic Net
  predicted_prob_enet <- as.numeric(predict(aim1_results[[i]]$elastic_net,
                                            newx = as.matrix(test_variables_dummy_include),
                                            s = "lambda.min", type = "response"))

  # Predict probabilities for Lasso
  predicted_prob_lasso <- as.numeric(predict(aim1_results[[i]]$lasso,
                                             newx = as.matrix(test_variables_dummy_include),
                                             s = "lambda.min", type = "response"))

  # Compute ROC and AUC for Elastic Net
  roc_enet <- roc(test_outcome, predicted_prob_enet)
  roc_results_enet[[i]] <- data.frame(
    Specificity = rev(roc_enet$specificities),
    Sensitivity = rev(roc_enet$sensitivities)
  )
  auc_values_enet[[i]] <- auc(roc_enet)  # Store AUC separately

  # Compute ROC and AUC for Lasso
  roc_lasso <- roc(test_outcome, predicted_prob_lasso)
  roc_results_lasso[[i]] <- data.frame(
    Specificity = rev(roc_lasso$specificities),
    Sensitivity = rev(roc_lasso$sensitivities)
  )
  auc_values_lasso[[i]] <- auc(roc_lasso)  # Store AUC separately


  # Calibration for Elastic Net
  calib_data_enet <- data.frame(
    prob = predicted_prob_enet,
    bin = cut(predicted_prob_enet, breaks = num_cuts),
    class = as.numeric(test_outcome) - 1
```

```
  )
  calib_results_enet[[i]] <- calib_data_enet %>%
    group_by(bin) %>%
    summarise(
      observed = mean(class),
      predicted = mean(prob),
      se = sqrt(observed * (1 - observed) / n())
    )

  # Calibration for Lasso
  calib_data_lasso <- data.frame(
    prob = predicted_prob_lasso,
    bin = cut(predicted_prob_lasso, breaks = num_cuts),
    class = as.numeric(test_outcome) - 1
  )
  calib_results_lasso[[i]] <- calib_data_lasso %>%
    group_by(bin) %>%
    summarise(
      observed = mean(class),
      predicted = mean(prob),
      se = sqrt(observed * (1 - observed) / n())
    )
}


# Extract numeric AUC values from the list of AUC objects
auc_numeric_enet <- sapply(auc_values_enet, function(x) as.numeric(x))
auc_numeric_lasso <- sapply(auc_values_lasso, function(x) as.numeric(x))

# Compute the mean AUC
mean_auc_enet <- mean(auc_numeric_enet)
mean_auc_lasso <- mean(auc_numeric_lasso)

# Define a common set of specificity thresholds
common_specificities <- seq(0, 1, length.out = 100)

# Interpolate sensitivity for each ROC curve at the common specificities
interp_sensitivities_lasso <- sapply(roc_results_lasso, function(roc_data) {
  approx(x = roc_data$Specificity, y = roc_data$Sensitivity, xout = common_specificities)$y
})

# Compute the mean sensitivity across imputations
mean_sensitivity_lasso <- rowMeans(interp_sensitivities_lasso, na.rm = TRUE)

# Create a data frame for the averaged ROC curve
mean_roc_lasso <- data.frame(
  Specificity = common_specificities,
  Sensitivity = mean_sensitivity_lasso
)

# Plot the averaged ROC curve
ROC_lasso = ggplot(mean_roc_lasso, aes(x = 1 - Specificity, y = Sensitivity)) +
  geom_line(color = "black", size = 1) +
```

```r
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "grey") +
  annotate("text", x = 0.8, y = 0.2, label = paste("Mean AUC =", round(mean_auc_lasso, 2)), size = 3, co
  labs(
    x = "1 - Specificity",
    y = "Sensitivity"
  ) +
  theme_minimal()

# Define a common set of specificity thresholds
common_specificities <- seq(0, 1, length.out = 100)

# Interpolate sensitivity for each ROC curve at the common specificities
interp_sensitivities_enet <- sapply(roc_results_enet, function(roc_data) {
  approx(x = roc_data$Specificity, y = roc_data$Sensitivity, xout = common_specificities)$y
})

# Compute the mean sensitivity across imputations
mean_sensitivity_enet <- rowMeans(interp_sensitivities_enet, na.rm = TRUE)

# Create a data frame for the averaged ROC curve
mean_roc_enet <- data.frame(
  Specificity = common_specificities,
  Sensitivity = mean_sensitivity_enet
)

# Plot the averaged ROC curve
ROC_enet = ggplot(mean_roc_enet, aes(x = 1 - Specificity, y = Sensitivity)) +
  geom_line(color = "black", size = 1) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "grey") +
  annotate("text", x = 0.8, y = 0.2, label = paste("Mean AUC =", round(mean_auc_enet, 2)), size = 3, col
  labs(
    x = "1 - Specificity",
    y = "Sensitivity"
  ) +
  theme_minimal()

# Combine Calibration Results Across Imputations
calib_combined_enet <- do.call(rbind, calib_results_enet) %>%
  group_by(bin) %>%
  summarise(
    observed = mean(observed),
    predicted = mean(predicted),
    se = sqrt(sum(se^2) / n())
  )

calib_combined_lasso <- do.call(rbind, calib_results_lasso) %>%
  group_by(bin) %>%
  summarise(
    observed = mean(observed),
    predicted = mean(predicted),
    se = sqrt(sum(se^2) / n())
  )
```

```r
num_cuts <- 10  # Number of bins for calibration

# Add Loess Fit for Flexible Calibration Line
loess_fit <- loess(observed ~ predicted, data = calib_combined_lasso, span = 0.75)
calib_combined_lasso$loess_pred <- predict(loess_fit, calib_combined_lasso$predicted)

# Plot Calibration Curve with Error Bars
calib_error_bar_lasso = ggplot(calib_combined_lasso) +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  geom_errorbar(aes(x = predicted, ymin = observed - 1.96 * se,
                    ymax = observed + 1.96 * se),
                colour="black", width=.01)+
  geom_point(aes(x = predicted, y = observed)) +
  labs(x = "Expected Probability of Smoking Abstinence",
       y = "Actual Smoking Abstinence") +
  #       title = "Calibration Plot for Elastic Net Model with Error Bars"
  theme_minimal()


# Plot Calibration Curve with Loess
calib_combined_lasso <- calib_combined_lasso %>%
  mutate(loess_ci_lower = loess_pred - 1.96 * sd(loess_pred),
         loess_ci_upper = loess_pred + 1.96 * sd(loess_pred))

calib_loess_lasso = ggplot(calib_combined_lasso, aes(x = predicted, y = observed)) +
  # Flexible calibration (Loess)
  geom_line(aes(y = loess_pred), color = "blue", linetype = "dashed") +
  geom_ribbon(aes(ymin = loess_ci_lower, ymax = loess_ci_upper), alpha = 0.2, fill = "grey") +
  geom_abline(intercept = 0, slope = 1, color = "red") +  # Perfect calibration line
  scale_color_manual(values = c("Ideal" = "red",
                                "Flexible calibration" = "blue")) +
  scale_linetype_manual(values = c("Ideal" = "solid",
                                   "Flexible calibration" = "dashed")) +
  labs(x = "Predicted Probability of Smoking Abstinence",
       y = "Actual Smoking Abstinence",
       color = "Legend", linetype = "Legend") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

num_cuts <- 10  # Number of bins for calibration

# Add Loess Fit for Flexible Calibration Line
loess_fit <- loess(observed ~ predicted, data = calib_combined_enet, span = 0.75)
calib_combined_enet$loess_pred <- predict(loess_fit, calib_combined_enet$predicted)

# Plot Calibration Curve with Error Bars
calib_error_bar_enet = ggplot(calib_combined_enet) +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  geom_errorbar(aes(x = predicted, ymin = observed - 1.96 * se,
                    ymax = observed + 1.96 * se),
                colour="black", width=.01)+
  geom_point(aes(x = predicted, y = observed)) +
  labs(x = "Expected Probability of Smoking Abstinence",
```

```
       y = "Actual Smoking Abstinence") +
  #       title = "Calibration Plot for Elastic Net Model with Error Bars"
  theme_minimal()


# Plot Calibration Curve with Loess
calib_combined_enet <- calib_combined_enet %>%
  mutate(loess_ci_lower = loess_pred - 1.96 * sd(loess_pred),
         loess_ci_upper = loess_pred + 1.96 * sd(loess_pred))

calib_loess_enet = ggplot(calib_combined_enet, aes(x = predicted, y = observed)) +
  # Flexible calibration (Loess)
  geom_line(aes(y = loess_pred), color = "blue", linetype = "dashed") +
  geom_ribbon(aes(ymin = loess_ci_lower, ymax = loess_ci_upper), alpha = 0.2, fill = "grey") +
  geom_abline(intercept = 0, slope = 1, color = "red") +  # Perfect calibration line
  scale_color_manual(values = c("Ideal" = "red",
                                "Flexible calibration" = "blue")) +
  scale_linetype_manual(values = c("Ideal" = "solid",
                                   "Flexible calibration" = "dashed")) +
  labs(x = "Predicted Probability of Smoking Abstinence",
       y = "Actual Smoking Abstinence",
       color = "Legend", linetype = "Legend") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# Initialize lists to store results for each imputed dataset
brier_enet <- numeric(5)
brier_lasso <- numeric(5)
CE_enet <- numeric(5)
CE_lasso <- numeric(5)
auc_enet <- numeric(5)
auc_lasso <- numeric(5)
threshold_enet <- numeric(5)
specificity_enet <- numeric(5)
sensitivity_enet <- numeric(5)
threshold_lasso <- numeric(5)
specificity_lasso <- numeric(5)
sensitivity_lasso <- numeric(5)

# calibration error
ce_calculate <- function(predictions, actuals, n_bins = 10) {
  bins <- cut(predictions, breaks = seq(0, 1, length.out = n_bins + 1),
              include.lowest = TRUE)
  bin_means <- tapply(predictions, bins, mean)
  bin_actuals <- tapply(actuals, bins, mean)
  bin_weights <- table(bins) / length(predictions)
  ce <- sum(bin_weights * abs(bin_means - bin_actuals))

  # Remove NA values from bin_means and bin_actuals
  valid_bins <- !is.na(bin_means) & !is.na(bin_actuals)
  bin_means <- bin_means[valid_bins]
  bin_actuals <- bin_actuals[valid_bins]
  bin_weights <- bin_weights[valid_bins]
```

```
  # Calculate ECE with valid bins only
  CE <- sum(bin_weights * abs(bin_means - bin_actuals))

  return(CE)
}

for (i in 1:5) {
  dataset <- complete(imputed_data, action = i)
  set.seed(58)
  train_index <- createDataPartition(dataset$abst, p = 0.7, list = FALSE)
  test_data <- dataset[-train_index, ]

  # Prepare predictors and outcome for test set
  test_variables_dummy <- model.matrix(~ 0 + ., data = test_data[, variable_names])
  test_variables_dummy <- test_variables_dummy[, -which(colnames(test_variables_dummy) == "Var0")]
  test_variables_dummy_full_interactions <- model.matrix(~ . ^ 2, data = as.data.frame(test_variables_du
  test_variables_dummy_include <- test_variables_dummy_full_interactions[, train_variables_dummy_include
  test_outcome_numeric <- as.numeric(test_data$abst) - 1

  # Predict probabilities for Elastic Net and Lasso
  predicted_prob_enet <- as.numeric(predict(aim1_results[[i]]$elastic_net,
                                            newx = as.matrix(test_variables_dummy_include),
                                            s = "lambda.min", type = "response"))
  predicted_prob_lasso <- as.numeric(predict(aim1_results[[i]]$lasso,
                                             newx = as.matrix(test_variables_dummy_include),
                                             s = "lambda.min", type = "response"))

  # Brier Score
  brier_enet[i] <- mean((predicted_prob_enet - test_outcome_numeric)^2)
  brier_lasso[i] <- mean((predicted_prob_lasso - test_outcome_numeric)^2)

  # Calibration Error
  CE_enet[i] <- ce_calculate(predicted_prob_enet, test_outcome_numeric)
  CE_lasso[i] <- ce_calculate(predicted_prob_lasso, test_outcome_numeric)

  # AUC
  roc_enet <- roc(test_outcome_numeric, predicted_prob_enet)
  roc_lasso <- roc(test_outcome_numeric, predicted_prob_lasso)
  auc_enet[i] <- as.numeric(auc(roc_enet))
  auc_lasso[i] <- as.numeric(auc(roc_lasso))

  # Optimal Threshold, Specificity, and Sensitivity
  coords_enet <- as.numeric(coords(roc_enet, "best",
                                   ret = c("threshold", "specificity", "sensitivity")))
  coords_lasso <- as.numeric(coords(roc_lasso, "best",
                                    ret = c("threshold", "specificity", "sensitivity")))

  threshold_enet[i] <- coords_enet[1]
  specificity_enet[i] <- coords_enet[2]
  sensitivity_enet[i] <- coords_enet[3]

  threshold_lasso[i] <- coords_lasso[1]
  specificity_lasso[i] <- coords_lasso[2]
```

```
  sensitivity_lasso[i] <- coords_lasso[3]
}

# Aggregate metrics across imputations
brier_mean_enet <- mean(brier_enet)
brier_mean_lasso <- mean(brier_lasso)
CE_mean_enet <- mean(CE_enet)
CE_mean_lasso <- mean(CE_lasso)
auc_mean_enet <- mean(auc_enet)
auc_mean_lasso <- mean(auc_lasso)
threshold_mean_enet <- mean(threshold_enet)
threshold_mean_lasso <- mean(threshold_lasso)
specificity_mean_enet <- mean(specificity_enet)
specificity_mean_lasso <- mean(specificity_lasso)
sensitivity_mean_enet <- mean(sensitivity_enet)
sensitivity_mean_lasso <- mean(sensitivity_lasso)
# Combine metrics into a table
predictor_df_performance <- rbind(
  `Brier score` = round(c(brier_mean_enet, brier_mean_lasso), 4),
  `Calibration error` = round(c(CE_mean_enet, CE_mean_lasso), 4),
  AUC = round(c(auc_mean_enet, auc_mean_lasso), 4),
  Threshold = round(c(threshold_mean_enet, threshold_mean_lasso), 4),
  Specificity = round(c(specificity_mean_enet, specificity_mean_lasso), 4),
  Sensitivity = round(c(sensitivity_mean_enet, sensitivity_mean_lasso), 4)
)

# Rename columns
colnames(predictor_df_performance) <- c("Elastic Net", "Lasso")

# Display the final table
kable(
  predictor_df_performance,
  caption = "Calibration and Discrimination Metrics for Moderator Effect Modeling"
)


plots_enet = arrangeGrob(
  calib_error_bar_enet, calib_loess_enet,  ROC_enet,
  ncol = 3,
  top = textGrob("Elastic Net Regression",
                 gp = gpar(fontface = "bold", fontsize = 14)
))

plots_lasso = arrangeGrob(
  calib_error_bar_lasso, calib_loess_lasso, ROC_lasso,
  ncol = 3,
  top = textGrob("Lasso Regression",
                 gp = gpar(fontface = "bold", fontsize = 14)
))

# Bold the main title
main_title <- textGrob(
  "Figure 2: Calibration Plots with Error Bars and LOESS and ROC Curves (Moderator Effects)",
```

```
  gp = gpar(fontsize = 16)
)

# Arrange everything with the bold title
grid.arrange(
  plots_lasso,
  plots_enet,
  nrow = 2,
  top = main_title
)
train_predictors_dummy_include_names <- c(
  "Var1", "BA1", "age_ps", "sex_ps2", "inc2", "inc3",
  "inc4", "inc5", "edu_merged2", "edu_merged3",
  "raceBlack", "raceHispanic", "raceOther",
  "ftcd_score", "ftcd.5.mins1", "bdi_score_w00", "cpd_ps",
  "crv_total_pq1", "hedonsum_n_pq1", "hedonsum_y_pq1",
  "shaps_score_pq1", "otherdiag1", "antidepmed1",
  "mde_curr1", "NMR", "Only.Menthol1",
  "readiness"
)

 predictor_names <- c("Var", "BA", "age_ps", "sex_ps", "inc", "edu_merged", "race",
                      "ftcd_score", "ftcd.5.mins", "bdi_score_w00", "cpd_ps",
                      "crv_total_pq1", "hedonsum_n_pq1", "hedonsum_y_pq1",
                      "shaps_score_pq1", "otherdiag", "antidepmed", "mde_curr",
                      "NMR", "Only.Menthol", "readiness")

# Helper function to perform model fitting
 fit_models <- function(data) {
   # Define predictors and outcome
   outcome <- data$abst

   predictors <- data[, predictor_names]
   # for Lasso (to break down factors with >2 levels)
   predictors_dummy <- model.matrix( ~ 0 + ., data = predictors)
   # remove the extra reference group
   predictors_dummy <- predictors_dummy[, -which(colnames(predictors_dummy) ==
                                                   "Var0")]



  # Split into train and test sets
   set.seed(46)
   train_index <- createDataPartition(outcome, p = 0.7, list = FALSE)
   train_data <- data[train_index, ]
   train_outcome = train_data$abst
   test_data <- data[-train_index, ]
   test_outcome = test_data$abst

   train_predictors_dummy <- predictors_dummy[train_index, ]
   test_predictors_dummy <- predictors_dummy[-train_index, ]

   train_predictors_dummy_include =
```

```
    train_predictors_dummy[, train_predictors_dummy_include_names]
  test_predictors_dummy_include =
    test_predictors_dummy[, train_predictors_dummy_include_names]

  # Set penalty factors to enforce keeping Var and BA
  # Initialize penalty factors to 1 for all variables
  penalty_factors <- rep(1, ncol(train_predictors_dummy_include))

  # Identify columns corresponding exactly to "Var1" and "BA1" (not their interactions)
  var1_col <- grep("^Var1$", colnames(train_predictors_dummy_include))
  ba1_col <- grep("^BA1$", colnames(train_predictors_dummy_include))

  penalty_factors[c(var1_col, ba1_col)] <- 0
  names(penalty_factors) <- colnames(train_predictors_dummy_include)

  # Fit Elastic Net model
  enet_model <- cv.glmnet(
    as.matrix(train_predictors_dummy_include),
    train_outcome,
    penalty.factor = penalty_factors,
    alpha = 0.5,
    family = "binomial",
    nfold = 10
  )

  # Fit Lasso model (alpha = 1)
  lasso_model <- cv.glmnet(
    as.matrix(train_predictors_dummy_include),
    train_outcome,
    penalty.factor = penalty_factors,
    alpha = 1,
    family = "binomial",
    nfold = 10
  )

  list(
    elastic_net = enet_model,
    lasso = lasso_model
  )
}
# Fit models across imputed datasets
aim2_results <- list()
for (i in 1:5) {
  dataset <- complete(imputed_data, action = i)
  aim2_results[[i]] <- fit_models(dataset)
}
# Initialize lists to store coefficients for Elastic Net and Lasso
elastic_net_coefs <- lapply(aim2_results, function(res) coef(res$elastic_net, s = res$elastic_net$lambda
lasso_coefs <- lapply(aim2_results, function(res) coef(res$lasso, s = res$lasso$lambda.min))

# Function to process coefficients for averaging and selection count
process_coefs <- function(coefs_list) {
  # Combine coefficients into a matrix
```

```r
  coefs_matrix <- do.call(cbind, lapply(coefs_list, function(coef) as.numeric(coef)))
  rownames(coefs_matrix) <- rownames(coefs_list[[1]])

  # Count non-zero selections for each variable
  selection_counts <- rowSums(coefs_matrix != 0)

  # Compute the average of coefficients only when selected (non-zero)
  averaged_coefs <- rowSums(coefs_matrix) / selection_counts
  averaged_coefs[is.na(averaged_coefs)] <- 0  # Handle cases where selection_counts is 0

  # Create result table
  result_table <- data.frame(
    variable = rownames(coefs_matrix),
        SelectionTimes = selection_counts,
    AverageCoefficient = averaged_coefs
  ) %>%
    filter(SelectionTimes > 0)  # Keep only variables selected at least once

  return(result_table)
}

# Process Elastic Net and Lasso coefficients
result_table_enet <- process_coefs(elastic_net_coefs)
result_table_lasso <- process_coefs(lasso_coefs)

# Summary table of coef
large_threshold <- 100

# Calculate OR for each coefficient for each method and rename columns correctly
enet_df <- result_table_enet %>%
  rename(`Selection Times` = SelectionTimes) %>%
  rename(`Elastic Net Average Coef` = AverageCoefficient) %>%
  mutate(`Elastic Net Average OR` = exp(`Elastic Net Average Coef`))

lasso_df <- result_table_lasso %>%
  rename(`Selection Times` = SelectionTimes) %>%
  rename(`Lasso Average Coef` = AverageCoefficient) %>%
  mutate(`Lasso OR` = exp(`Lasso Average Coef`))



# Merge all data frames based on variable names
combined_df <- full_join(lasso_df, enet_df, by = "variable") %>%
  mutate(across(where(is.numeric), ~ round(.x, 4))) %>%
  mutate(across(where(is.numeric), ~ ifelse(as.numeric(.) > large_threshold, "*", .)))

# Remove the intercept row
combined_df <- combined_df[combined_df$variable != "(Intercept)", ]

# Standardize interaction terms by sorting them alphabetically
combined_df <- combined_df %>%
  mutate(
    variable = sapply(variable, function(x) {
```

```
      terms <- unlist(strsplit(x, ":"))
      if (length(terms) > 1) {
        paste(sort(terms), collapse = ":")
      } else {
        x
      }
    })
  )

# Combine rows with the same standardized interaction term names
combined_df <- combined_df %>%
  group_by(variable) %>%
  summarize(across(everything(), ~ ifelse(is.numeric(.), sum(as.numeric(.), na.rm = TRUE), .))) %>%
  ungroup()

# Replace zeroes and any remaining NAs with an empty space for readability
combined_df <- combined_df %>%
  mutate(across(where(is.numeric), ~ ifelse(. == 0, "", .))) %>%
  replace(is.na(.), " ")

# Display the final combined table with grouped headers
combined_df %>%
  kable(row.names = F,
        col.names = c("Variable", "Selection Times","Average Coef", "OR", "Selection Times", "Average C
        caption = "Summary of Average Coefficients and Odds Ratios for Potential Predictor Effects acro
  add_header_above(c(" " = 1, "Lasso" = 3, "Elastic Net" = 3)) %>%
  kable_styling(full_width = F, position = "center", font_size = 9)

# Initialize lists to store results for each imputed dataset
brier_enet <- numeric(5)
brier_lasso <- numeric(5)
CE_enet <- numeric(5)
CE_lasso <- numeric(5)
auc_enet <- numeric(5)
auc_lasso <- numeric(5)
threshold_enet <- numeric(5)
specificity_enet <- numeric(5)
sensitivity_enet <- numeric(5)
threshold_lasso <- numeric(5)
specificity_lasso <- numeric(5)
sensitivity_lasso <- numeric(5)


for (i in 1:5) {

   # Define predictors and outcome
   data <- complete(imputed_data, action = i)

  # Define predictors and outcome
   outcome <- data$abst

   predictors <- data[, predictor_names]
   # for Lasso (to break down factors with >2 levels)
```

```
 predictors_dummy <- model.matrix( ~ 0 + ., data = predictors)
 # remove the extra reference group
 predictors_dummy <- predictors_dummy[, -which(colnames(predictors_dummy) ==
                                                 "Var0")]


# Split the dataset into training and test sets
 set.seed(46)
 train_index <- createDataPartition(outcome, p = 0.7, list = FALSE)
 train_data <- data[train_index, ]
 train_outcome = train_data$abst
 test_data <- data[-train_index, ]
 test_outcome = test_data$abst

 train_predictors_dummy <- predictors_dummy[train_index, ]
 test_predictors_dummy <- predictors_dummy[-train_index, ]

 train_predictors_dummy_include =
   train_predictors_dummy[, train_predictors_dummy_include_names]
 test_predictors_dummy_include =
   test_predictors_dummy[, train_predictors_dummy_include_names]
test_outcome_numeric <- as.numeric(test_data$abst) - 1

# Predict probabilities for Elastic Net and Lasso
predicted_prob_enet <- as.numeric(predict(aim2_results[[i]]$elastic_net,
                                          newx = as.matrix(test_predictors_dummy_include),
                                          s = "lambda.min", type = "response"))
predicted_prob_lasso <- as.numeric(predict(aim2_results[[i]]$lasso,
                                           newx = as.matrix(test_predictors_dummy_include),
                                           s = "lambda.min", type = "response"))

# Brier Score
brier_enet[i] <- mean((predicted_prob_enet - test_outcome_numeric)^2)
brier_lasso[i] <- mean((predicted_prob_lasso - test_outcome_numeric)^2)

# Calibration Error
CE_enet[i] <- ce_calculate(predicted_prob_enet, test_outcome_numeric)
CE_lasso[i] <- ce_calculate(predicted_prob_lasso, test_outcome_numeric)

# AUC
roc_enet <- roc(test_outcome_numeric, predicted_prob_enet)
roc_lasso <- roc(test_outcome_numeric, predicted_prob_lasso)
auc_enet[i] <- as.numeric(auc(roc_enet))
auc_lasso[i] <- as.numeric(auc(roc_lasso))

# Optimal Threshold, Specificity, and Sensitivity
coords_enet <- as.numeric(coords(roc_enet, "best",
                                 ret = c("threshold", "specificity", "sensitivity")))
coords_lasso <- as.numeric(coords(roc_lasso, "best",
                                  ret = c("threshold", "specificity", "sensitivity")))

threshold_enet[i] <- coords_enet[1]
specificity_enet[i] <- coords_enet[2]
```

```
  sensitivity_enet[i] <- coords_enet[3]

  threshold_lasso[i] <- coords_lasso[1]
  specificity_lasso[i] <- coords_lasso[2]
  sensitivity_lasso[i] <- coords_lasso[3]
}

# Aggregate metrics across imputations
brier_mean_enet <- mean(brier_enet)
brier_mean_lasso <- mean(brier_lasso)
CE_mean_enet <- mean(CE_enet)
CE_mean_lasso <- mean(CE_lasso)
auc_mean_enet <- mean(auc_enet)
auc_mean_lasso <- mean(auc_lasso)
threshold_mean_enet <- mean(threshold_enet)
threshold_mean_lasso <- mean(threshold_lasso)
specificity_mean_enet <- mean(specificity_enet)
specificity_mean_lasso <- mean(specificity_lasso)
sensitivity_mean_enet <- mean(sensitivity_enet)
sensitivity_mean_lasso <- mean(sensitivity_lasso)
# Combine metrics into a table
predictor_df_performance <- rbind(
  `Brier score` = round(c(brier_mean_enet, brier_mean_lasso), 4),
  `Calibration error` = round(c(CE_mean_enet, CE_mean_lasso), 4),
  AUC = round(c(auc_mean_enet, auc_mean_lasso), 4),
  Threshold = round(c(threshold_mean_enet, threshold_mean_lasso), 4),
  Specificity = round(c(specificity_mean_enet, specificity_mean_lasso), 4),
  Sensitivity = round(c(sensitivity_mean_enet, sensitivity_mean_lasso), 4)
)

# Rename columns
colnames(predictor_df_performance) <- c("Elastic Net", "Lasso")

# Display the final table
kable(
  predictor_df_performance,
  caption = "Calibration and Discrimination Metrics for Predictor Effect Modeling"
)
```