Identifying Key Moderators and Predictors of Smoking Cessation in Adults with Major Depressive Disorder

Practical Data Analysis - Project 2: Regression Analysis

Aristofanis Rontogiannis

2024-11-04

Abstract

Purpose: This project evaluates the efficacy of smoking cessation treatments among adults with Major Depressive Disorder (MDD), focusing on identifying baseline factors that moderate and predict treatment success. This analysis aims to clarify the role of both behavioral and pharmacological interventions, examining interactions between participant characteristics and treatment components.

Methods: Using data from a randomized, placebo-controlled trial, we assessed the effectiveness of Behavioral Activation (BA) and standard treatment (ST), each paired with either varenicline or a placebo, in supporting smoking cessation. Two statistical approaches– Lasso regression and Elastic Net Regression with 10-fold cross validation– were employed to identify significant moderators and predictors of abstinence at the end of treatment, based on a sample of 300 adult smokers with current or past MDD.

Results: The findings highlight the significant influence of baseline characteristics, including FTCD score, Nicotine Metabolism Ratio (`NMR`), and demographic factors such as race and income level, on smoking cessation outcomes. FTCD score was consistently associated with lower abstinence rates, while a higher NMR significantly increased the likelihood of cessation. Key moderators, such as menthol use, current depressive episodes (`MDE_curr1`), and raceOther, emerged as significant barriers, reducing the odds of abstinence across both Lasso and Elastic Net models. Positive predictors like income level 3 (`edu_merged3:inc3`) were associated with higher abstinence rates. These results were consistent across imputation datasets and methods, demonstrating a robust relationship between these predictors and smoking cessation outcomes.

Conclusions: This study underscores the importance of tailoring smoking cessation strategies for individuals with MDD by accounting for critical predictors such as nicotine dependence, metabolic rate, and psychosocial factors. The findings support the integration of behavioral and pharmacological treatments, with a focus on addressing key barriers like menthol use and depressive episodes. By identifying consistent predictors and moderators, this study provides a strong foundation for the development of personalized, evidence-based cessation programs for this high-risk population.

Introduction

This project is a collaboration with Dr. George Papandonatos from the Department of Biostatistics at Brown University. This project examines the effectiveness of smoking cessation treatments for adults with major depressive disorder (MDD), a group that encounters unique challenges when trying to quit smoking. People with MDD often smoke more heavily, have a stronger dependence on nicotine, and face more intense withdrawal symptoms than those without MDD. While varenicline, a medication for quitting smoking, has shown promising results, psychological approaches that address depression-related issues, like behavioral activation (BA), might further enhance quit rates for this group.

This project uses data from a randomized, placebo-controlled study (Hitsman et al. (2023)) that compared behavioral activation with standard treatment (ST), both combined with either varenicline or a placebo. The study allows us to explore how baseline characteristics may affect treatment success at the end of treatment (EOT), including 300 adult smokers who have current or past MDD. Specifically, we aim to identify baseline factors that may influence the effectiveness of behavioral treatments and predict abstinence, considering both behavioral and medication-based therapies.

The findings from this research could lead to more effective, personalized smoking cessation strategies for people with MDD, addressing their specific challenges and improving treatment outcomes.

Data Overview

As shown in Table 1 below, participant characteristics were stratified by four treatment combination– behavioral activation with placebo (BASC+placebo), standard treatment with placebo (ST+placebo), behavioral activation with varenicline (BASC+varenicline), and standard treatment with varenicline (ST+varenicline)– and summarized by treatment group in three primary areas: demographics, smoking characteristics, and psychiatric characteristics.

In the demographics section, participant variables such as age, sex, race, income, and education level were included. Age was summarized as a continuous variable using mean and standard deviation, while categorical variables like sex, race, income, and education level were presented with counts and percentages for each treatment group. This section provides an overview of the baseline demographic distribution across the four treatment conditions.

Smoking-related characteristics were summarized to include baseline measures of cigarette consumption, such as cigarettes per day, FTCD score, and readiness to quit smoking. Continuous variables like cigarettes per day and FTCD score were summarized with mean and standard deviation, while categorical variables, such as whether participants smoked within five minutes of waking up, were presented with counts and percentages. This section helps illustrate smoking behaviors across the different treatment groups.

Psychiatric characteristics were also summarized to capture information related to mental health, such as anhedonia (measured by the SHAPS score), presence of other DSM-5 diagnoses, use of antidepressant medication, and current versus past MDD status. Continuous variables, like the SHAPS score, were summarized by mean and standard deviation, while categorical variables, like the presence of other DSM-5 diagnoses, were displayed with counts and percentages. This section highlights the psychiatric characteristics of participants across treatment groups.

With those three sections, Table 1 provides an overall snapshot of participant characteristics by treatment group and by overall sample. This table is organized to present a clear and comprehensive view of the data, allowing for straightforward comparisons across treatment groups and providing context for further analyses on treatment outcomes.

Table 1: Participant characteristics by treatment and overall sample

Group	Characteristic	Overall N = 300	BASC+placebo N = 68	BASC+varenicline N = 83	ST+placebo N = 68	ST+varenicline N = 81
Demographics	Age (years)	50.0 (12.6)	50.7 (13.5)	50.3 (13.2)	50.3 (10.8)	48.7 (12.7)
	Sex (Female)	165 (55%)	38 (56%)	44 (53%)	39 (57%)	44 (54%)
	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					
	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 at baseline phone survey	15.1 (7.9)	15.6 (9.1)	15.5 (8.5)	15.0 (7.2)	14.4 (6.6)
	FTCD score at baseline	5.2 (2.1)	5.3 (2.0)	5.1 (2.3)	5.4 (2.1)	5.2 (2.1)
	Smoking within 5 mins of waking up (Yes)	138 (46%)	32 (47%)	33 (40%)	35 (51%)	38 (47%)
	BDI score at baseline	18.7 (11.5)	19.0 (12.3)	18.0 (10.6)	18.5 (10.8)	19.5 (12.2)
	Cigarette reward value at baseline	7.2 (3.7)	7.4 (3.8)	7.2 (3.9)	7.0 (3.7)	7.1 (3.5)
	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)
	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)
	Exclusive Mentholated Cigarette User (Yes)	178 (60%)	40 (59%)	48 (59%)	43 (64%)	47 (58%)
	Readiness to quit smoking	6.8 (1.2)	6.8 (1.4)	6.7 (1.2)	7.0 (1.3)	6.7 (1.1)
	Nicotine Metabolism Ratio	0.4 (0.2)	0.3 (0.2)	0.4 (0.2)	0.4 (0.3)	0.4 (0.2)
Psychiatric	Anhedonia	2.2 (3.2)	2.2 (3.2)	2.3 (3.1)	2.5 (3.4)	2.1 (3.0)
	Other lifetime DSM-5 diagnosis (Yes)	133 (44%)	35 (51%)	30 (36%)	28 (41%)	40 (49%)
	Taking antidepressant medication at baseline (Yes)	82 (27%)	28 (41%)	24 (29%)	15 (22%)	15 (19%)
	Current vs past MDD (Yes)	147 (49%)	32 (47%)	40 (48%)	31 (46%)	44 (54%)

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

Additionally, to facilitate certain analyses, a new variable was created to consolidate education levels. The three lower levels of education (Grade school, Some high school and high school graduate or GED) were combined into a single category representing lower education levels. This aggregation simplifies the education variable, as those three levels contained a small amount of participants, allowing for broader categorical comparisons while retaining essential information on educational background.

Data Missingness and Imputation

The missing data analysis involves examining the extent of missingness across variables in the dataset. A summary table was generated (Table 2), listing variables with missing values, the count of missing entries for each variable, and the corresponding percentage relative to the total sample size. For example, the variable with the highest missingness is the variable NMR (7%), followed by the variables crv_total_pq1 (6%) and readiness (5.67%). Variables with no missing values were excluded from this table for clarity. This missingness analysis is a critical step in data preparation, as it helps to address data quality issues and ensure that the dataset is ready for further statistical modeling and interpretation. The total percentage of missingness was around 20%. If we have deleted all the rows with missing data, we might have lost a significant portion of our dataset, which could reduce the statistical power of our analyses and lead to biased estimates.

Table 2: Summary of Missing Values

Variable	Number	Pct
missing_NMR	21	7.00%
missing_crv_total_pq1	18	6.00%
missing_readiness	17	5.67%
missing_inc	3	1.00%
missing_shaps_score_pq1	3	1.00%
missing_Only.Menthol	2	0.67%
missing_ftcd_score	1	0.33%

To handle missing data, the Multiple Imputation by Chained Equations (MICE) method was applied. MICE is an iterative process that generates multiple plausible datasets by imputing missing values based on the observed data structure. We performed five imputations, meaning five separate datasets were created with imputed values. For each imputed dataset, we conducted 10-fold cross-validation to ensure robust model evaluation.

Subsequently, we ran Lasso regression and Elastic Net regression models on each of the five imputed datasets. The results from these models were aggregated by taking the mean of the coefficients across the datasets, providing stable and interpretable parameter estimates that account for uncertainty in the imputed values.

Exploration of potential interactions

As part of the Explanatory Data Analysis, we want to examine the relationship between menthol cigarette use and race. We created a contingency table to display the frequencies of menthol and non-menthol use across racial categories, and then performed a Chi-square test to assess whether there is a statistically significant association between race and menthol cigarette use.

The Chi-square test produced a very low p-value, indicating a statistically significant association between race and menthol cigarette use. This result suggests that menthol cigarette usage depends on race, as there appears to be a notable relationship between the two variables.

We can easily observe from Table 3 that Black individuals have a preference for menthol cigarettes over regular non-menthol ones (130 out of 157 people).

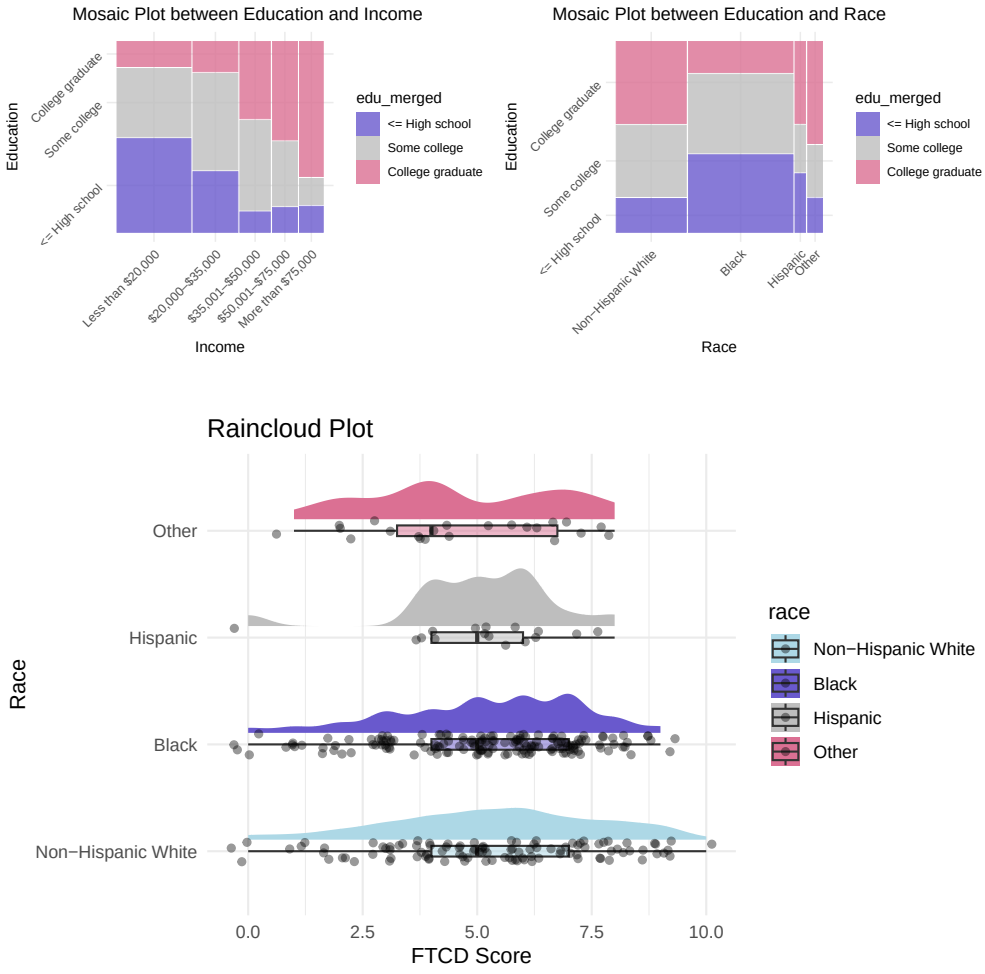
Table 3: Contingency Table of Only Menthol Use and Race

	Non-Menthol	Menthol
Non-Hispanic White	72	33
Black	28	129
Hispanic	12	4
Other	9	13

Note:
Chi-Square Statistic: 75.8077, p-value : Approaching 0.0000

Additionally, we examine the relationships between education, income, race, and FTCD scores through mosaic and raincloud plots.

Figure 1: Mosaic/Raincloud plots for the relationships between education, income, race, and FTCD scores



Mosaic Plot between Education and Income This plot shows the distribution of education levels (edu_merged) across different income categories. We observe a gradient where higher education levels (coded in pink) tend to be associated with higher income levels, while lower education levels (in blue) were more concentrated in lower income brackets. This suggests a positive association between education and income.

Mosaic Plot between Education and Race This plot reveals the distribution of education levels across different racial groups. The distribution varies, with Non-Hispanic Whites showing a relatively higher proportion in the higher education categories, while other groups, such as Black and Hispanic, have a more balanced spread across the education levels. This pattern may indicate disparities in educational attainment across racial groups.

Raincloud Plot for FTCD Score by Race The raincloud plot shows the distribution and density of FTCD scores across racial groups. Non-Hispanic Whites have the highest density around higher FTCD scores, while other groups show a more varied distribution, with Hispanics and Black participants displaying lower scores overall. This visualization highlights potential differences in FTCD scores among racial groups, which may correlate with other socioeconomic factors.

Methods

Data separation and full model selection

To assess the model's performance and ensure its generalizability, we divided the data into training and test sets, with 70% of the observations assigned to the training set and 30% to the test set. This split allows us to train the model on one portion of the data and then evaluate its performance on a separate, unseen portion, reducing the risk of overfitting.

We ensured that the variables Var and BA are kept controlled in the model as those variables are crucial for investigating treatment interactions and their effect on abstinence. For objective one, we also added interaction terms between BA and baseline variables to explore potential moderator effects and other interaction terms, e.g., `NMR:readiness` and `income:education`, as reasonable covariates. We consulted previous literature (Jiloha (2010), Brown et al. (2015), Hitsman et al. (2003)) to help decide the inclusion of certain interaction terms and covariates. For the second objective of this project, we considered baseline characteristics as potential predictors while keeping BA and Var controlled. Thus, for simplicity, no interaction terms were included in the full models.

Lasso Regression

In this part of the analysis, we implemented Lasso regression to identify potential moderators of treatment effects on smoking abstinence. Lasso reduces the model to a subset of variables and interactions most predictive of the outcome by penalizing less relevant coefficients, setting them to zero, and enhancing interpretability.

The model was trained on the training set, with cross-validation used to determine the optimal penalty parameter (lambda), balancing model simplicity and predictive accuracy. The test set was then used to evaluate the model's generalizability to new data, ensuring robustness.

The final model coefficients at the optimal lambda include only variables and interactions with non-zero coefficients. Tables 4 and 5 present two columns: one showing how often each variable was selected across the imputed datasets generated by MICE, and another reporting the odds ratio (OR) for each selected variable. This column is more useful than the average coefficients column as the outcome is binary and thus we can interpret it easier using OR.This approach highlights significant moderators of smoking abstinence while maintaining a streamlined model.

Elastic Net Regression

Furthermore, we implemented Elastic Net regression to identify potential moderators of treatment effects on smoking abstinence. Elastic Net combines the penalties of Lasso and Ridge regression, making it well-suited for handling correlated predictors and selecting a subset of variables that are most predictive of the outcome.

Similarly, the model was trained on the training set, with cross-validation used to determine the optimal penalty parameter (lambda, while alpha=0.5), balancing model complexity and predictive accuracy. The test set was then used to evaluate the model's generalizability to new data, ensuring robustness.

Similar to Lasso above, Tables 4 and 5 present two columns: one showing how often each variable was selected across the imputed datasets generated by MICE, and the other reporting the odds ratio (OR) for each selected variable.

Results

First Objective

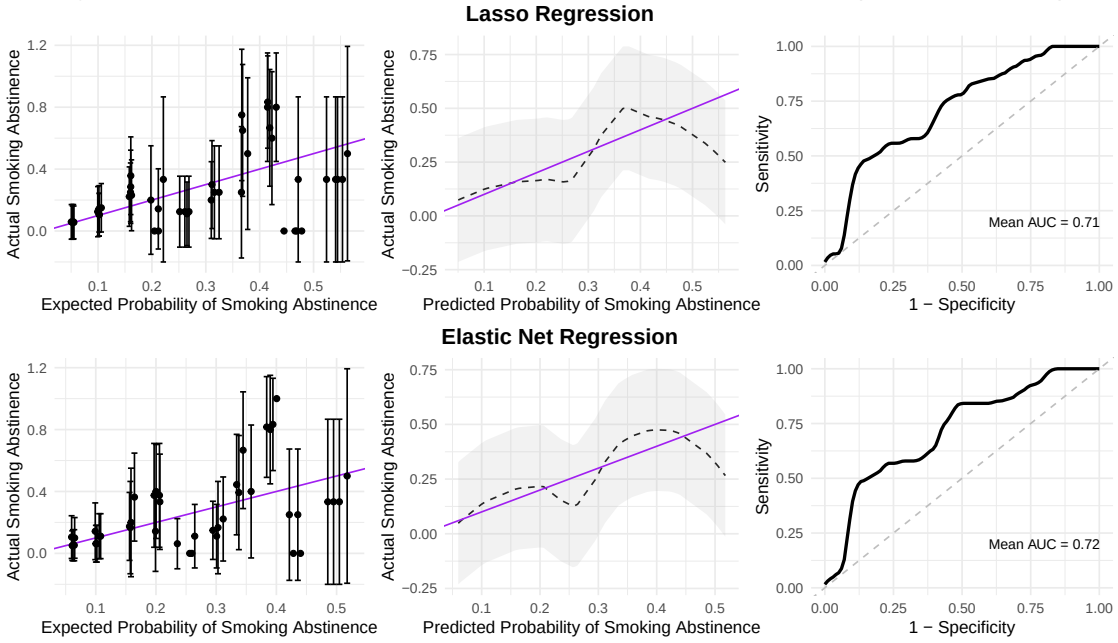
Table 4: Summary of Average Coefficients and Odds Ratios for Potential Moderator Effects across Model Selection Methods

	Lasso		Elastic Net	
Variable	Selection Times	Average OR	Selection Times	Average OR
BA1	5	1.0198	5	1.0465
NMR	3	1.4188	4	1.313
NMR:sex_ps2	3	1.406	4	1.2354
Only.Menthol1:raceOther	5	0.8696	5	0.9164
Var1	5	5.3352	5	5.269
age_ps	3	1.0013	2	1.0001
age_ps:NMR	3	1.0098	3	1.0071
antidepmed1	1	1.0186		
antidepmed1:readiness	2	1.0016		
edu_merged2:Only.Menthol1	5	0.4953	5	0.6152
edu_merged3:inc3	5	1.1509	5	1.1175
edu_merged3:inc4	5	0.8173	4	0.9362
ftcd_score	5	0.8574	5	0.8868
mde_curr1	5	0.6287	5	0.7187
mde_curr1:readiness			2	0.9893
raceBlack	5	0.8968	5	0.9524
raceHispanic:Var1	5	0.371	5	0.5138
raceOther	5	0.8075	5	0.7852
raceOther:Var1	5	0.5602	5	0.6935

The table summarizes results from Lasso and Elastic Net models, highlighting key moderators of smoking abstinence. Variables selected consistently (4 or 5 times) are most important. In both models, `NMR` (Lasso OR = 1.4188; Elastic Net OR = 1.313) and `edu_merged3:inc3` (Lasso OR = 1.1509; Elastic Net OR = 1.1175) increase the odds of abstinence, while `edu_merged2:Only.Menthol1` (Lasso OR = 0.4953; Elastic Net OR = 0.6152) and `mde_curr1` (Lasso OR = 0.6287; Elastic Net OR = 0.7187) significantly reduce the odds.

Race and interaction effects also play a role. The variable `raceHispanic:Var1` (Lasso OR = 0.371; Elastic Net OR = 0.5138) strongly decreases the likelihood of abstinence, while `raceBlack` shows a smaller negative effect in both models. The results emphasize `NMR` and income (`inc3`) as positive moderators and menthol use, depression, and some racial/interaction variables as barriers to quitting. The consistency across models highlights the robustness of these findings.

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



Calibration Plots (Left Panel)

Lasso Model The calibration plot for the Lasso model shows a generally positive alignment between expected and actual probabilities of smoking abstinence, though with some variability at higher predicted probabilities. Error bars indicate increasing uncertainty at these levels.

Elastic Model Similar to the Lasso model, the Elastic Net calibration plot demonstrates reasonable alignment between expected and actual abstinence probabilities, with slightly tighter error bars at intermediate predicted probabilities.

LOESS-Smoothing Calibration Curves (Middle Panel)

Lasso Model The LOESS-smoothed curve for the Lasso model deviates somewhat from the ideal diagonal line, indicating moderate under- and over-predictions in certain ranges of predicted probabilities.

Elastic Model The LOESS-smoothed curve for the Elastic Net model follows the ideal diagonal more closely than Lasso, suggesting slightly better calibration overall.

ROC Curves (Right Panel)

Lasso Model The ROC curve for the Lasso model achieves a mean AUC of 0.71, reflecting good but not perfect discrimination between abstinent and non-abstinent individuals.

Elastic Model The Elastic Net model achieves a slightly higher mean AUC of 0.72, indicating slightly better discrimination compared to Lasso.

Second Objective

For the second objective, we explored baseline characteristic as potential predictors. Overall, we followed similar analysis procedure as in objective one, but due to lower model complexity, we could employ exhaustive

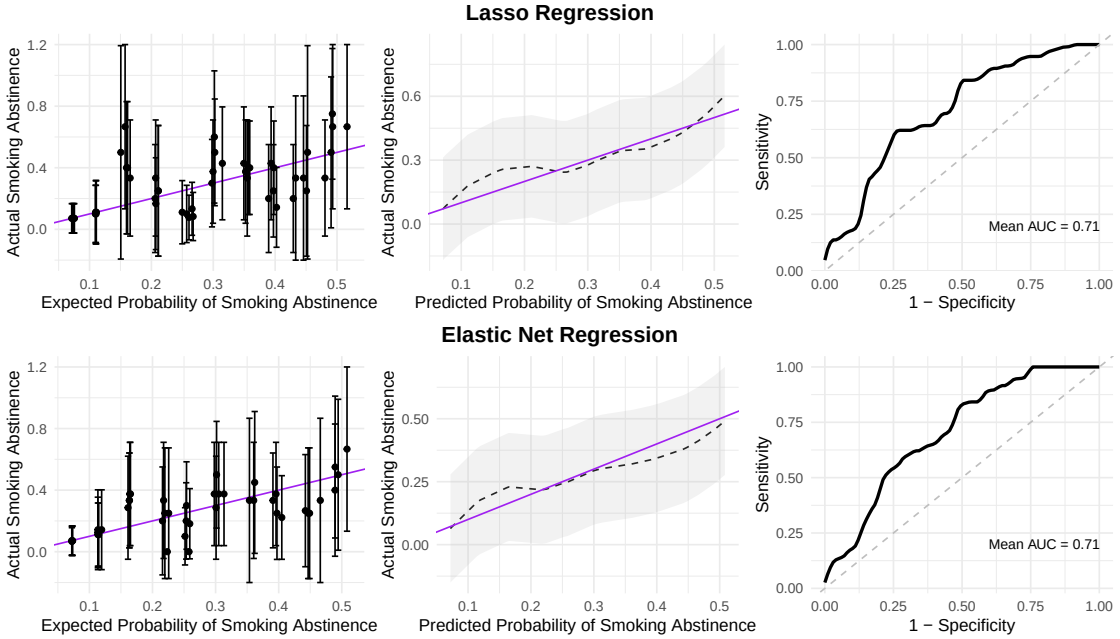
replacement in best subset selection. Table 5 compares the ORs of variables selected by the three different moThe Elastic Net model achieves the same mean AUC of 0.71, reflecting comparable discrimination to the Lasso model. deling approaches, showing how many time these variables were selected from MICE as well.

Table 5: Summary of Average Coefficients and Odds Ratios for Potential Predictor Effects across Model Selection Methods

	Lasso		Elastic Net	
Variable	Selection Times	Average OR	Selection Times	Average OR
BA1	5	1.0725	5	1.0310
NMR	5	1.5813	5	1.6650
Var1	5	5.4012	5	5.4363
age_ps			5	1.0026
antidepmed1			5	1.0607
ftcd_score	5	0.8831	5	0.8736
inc2			5	0.9538
mde_curr1	5	0.7224	5	0.6910
raceBlack	3	0.9865	5	0.8736
raceOther	5	0.5914	5	0.4960
shaps_score_pq1			4	0.9991

Both Lasso and Elastic Net consistently identify `NMR` (Lasso OR = 1.5813; Elastic Net OR = 1.6650) and `Var1` (Lasso OR = 5.4012; Elastic Net OR = 5.4363) as strong positive predictors of smoking abstinence. The variable `BA1` also shows a modest positive effect in both models (Lasso OR = 1.0725; Elastic Net OR = 1.0310). Conversely, `ftcd_score` (Lasso OR = 0.8831; Elastic Net OR = 0.8736), `mde_curr1` (Lasso OR = 0.7224; Elastic Net OR = 0.6910), and `raceOther` (Lasso OR = 0.5914; Elastic Net OR = 0.4960) consistently act as barriers, significantly reducing the odds of abstinence. The results are highly consistent across both models, reinforcing the importance of these predictors.

Figure 3: Calibration Plots with Error Bars and LOESS and ROC Curves (Predictor Effects)



Calibration Plots (Left Panel)

Lasso Model The calibration plot shows a generally positive alignment between expected and actual probabilities of smoking abstinence, with some variability and wider error bars at higher predicted probabilities.

Elastic Model Similar to Lasso, the Elastic Net plot shows reasonable alignment with expected probabilities, though the variability is slightly less pronounced at intermediate probability levels.

LOESS-Smoothing Calibration Curves (Middle Panel)

Lasso Model The LOESS-smoothed curve demonstrates moderate deviations from the ideal diagonal, indicating some under- and over-predictions across the range of probabilities.

Elastic Model The Elastic Net curve follows an almost ideal diagonal (more closely than Lasso), indicating slightly better calibration of predicted probabilities.

ROC Curves (Right Panel)

Lasso Model The ROC curve achieves a mean AUC of 0.71, indicating good discrimination between abstinent and non-abstinent individuals.

Elastic Model The Elastic Net model achieves the same mean AUC of 0.71, reflecting comparable discrimination to the Lasso model.

Conclusions and Limitations

This study underscores the potential benefits of tailored smoking cessation interventions for individuals with Major Depressive Disorder (MDD). Our analysis revealed that baseline characteristics, particularly FTCD score, Nicotine Metabolism Ratio (`NMR`), and demographic factors such as race, income, and menthol use, significantly impact abstinence outcomes. Higher nicotine dependence (FTCD score) was consistently associated with lower cessation success, while a higher NMR indicated a greater likelihood of quitting. Additionally, barriers such as current depressive episodes, menthol cigarette use, and `raceOther` significantly reduced abstinence rates, reinforcing the need for targeted interventions. These findings suggest that individualized treatments combining pharmacotherapy and behavioral strategies may enhance outcomes for this high-risk population.

The use of two modeling approaches—Lasso and Elastic Net—allowed for a robust evaluation of predictors, capturing both consistent and unique factors across methods. Key predictors such as `NMR`, FTCD score, and income level 3 emerged as robust across models, while variability in the selection of interaction terms highlighted the complexity of cessation success in individuals with MDD. These results demonstrate the value of integrating insights from multiple analytical frameworks to identify a broader range of influential factors, thereby enhancing the personalization of treatment approaches.

However, this study is not without limitations. The reliance on a moderate sample size and self-reported smoking data may limit the generalizability of findings and introduce potential reporting biases. Additionally, the inclusion of a high number of interaction terms in moderator effect exploration could increase the risk of overfitting, potentially reducing the external validity of the models. Finally, while the models performed well, the average AUC values (~0.71–0.72) suggest room for improvement in predictive accuracy.

Future research should build on these findings by employing larger sample sizes, objective measures of smoking, and long-term follow-up assessments to validate and refine predictive models. Incorporating additional predictors such as genetic markers, psychiatric profiles, and environmental factors could deepen our understanding of treatment responses. Ultimately, this work lays the foundation for developing more personalized and effective cessation strategies tailored to the unique needs of individuals with MDD.

Data Privacy and Code Availability

The analysis dataset was obtained by Dr. George Papandonatos from the Department of Biostatistics at Brown University and cannot be shared due to privacy. The replication code can be found at https://github.com/AristofanisR/Practical_Data_Analysis_Portfolio/tree/main/Project2

References

Brown, Clayton H, Deborah Medoff, Faith B Dickerson, Li Juan Fang, Alicia Lucksted, Richard W Goldberg, Julie Kreyenbuhl, Seth Himelhoch, and Lisa B Dixon. 2015. "Factors Influencing Implementation of Smoking Cessation Treatment Within Community Mental Health Centers." *Journal of Dual Diagnosis* 11 (2): 145–50.

Hitsman, Brian, Belinda Borrelli, Dennis E McChargue, Bonnie Spring, and Raymond Niaura. 2003. "History of Depression and Smoking Cessation Outcome: A Meta-Analysis." *Journal of Consulting and Clinical Psychology* 71 (4): 657.

Hitsman, Brian, George D Papandonatos, Jacqueline K Gollan, Mark D Huffman, Raymond Niaura, David C Mohr, Anna K Veluz-Wilkins, et al. 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* 118 (9): 1710–25.

Jiloha, RC. 2010. "Biological Basis of Tobacco Addiction: Implications for Smoking-Cessation Treatment." *Indian Journal of Psychiatry* 52 (4): 301–7.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE,
                      message = FALSE,
                      warning = FALSE,
                      error = FALSE)

library(summarytools)
library(ggplot2)
library(knitr)
library(kableExtra)
library(GGally)
library(patchwork)
library(dplyr)
library(reshape2)
library(tidyr)
library(grid)
library(lubridate)
library(gtsummary)
library(gt)
library(ggcorrplot)
library(glmnet)
library(MASS)
library(pROC)
library(gridExtra)
library(VIM)
library(mice)
library(leaps)
library(ggplot2)
library(ggmosaic)
library(ggbeeswarm)
library(ggdist)
library(cowplot)
library(caret)
#Read main data set
data<-read.csv("project2.csv")

# Data processing
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"))) %>%
  # Change variables attributes to be only Numeric or Factor at the end
  #Factor
```

```
  mutate(
    abst = factor(abst),
    Var = factor(Var),
    BA = factor(BA),
    sex_ps = factor(sex_ps),
    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))
  ) %>%
 #Numeric (except id)
  mutate(across(
    .cols = where(is.numeric) & !all_of("id"),
    .fns = as.numeric
  ))

# Summary Table
table1_data = data %>%
  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")),
    race = factor(race, labels = c("    Non-Hispanic White",
                                   "    Black",
                                   "    Hispanic",
                                   "    Other"))
    )

# Demographics table
demographics_table <- table1_data %>%
  dplyr::select(
    treatment_cat,
    age_ps,
    sex_ps,
    race,
    inc,
    edu
  ) %>%
  tbl_summary(
    by = treatment_cat,
    label = list(
      age_ps = "Age (years)",
```

```
      sex_ps = "Sex (Female)",
      race = "Race",
      inc = "Income",
      edu = "Education"
    ),
    type = list(
      age_ps ~ "continuous",
      sex_ps ~ "dichotomous",
      race ~ "categorical",
      inc ~ "categorical",
      edu ~ "categorical"),
    value = list(
      sex_ps ~ "2"),
    statistic = list(all_continuous() ~ "{mean} ({sd})",
                     all_categorical() ~ "{n} ({p}%)"),
    digits = all_continuous() ~ 1,
    missing = "no"
  ) %>% add_overall()


# Smoking table
smoking_table <- table1_data %>%
  dplyr::select(
    treatment_cat,
    cpd_ps,
    ftcd_score,
    ftcd.5.mins,
    bdi_score_w00,
    crv_total_pq1,
    hedonsum_n_pq1,
    hedonsum_y_pq1,
    Only.Menthol,
    readiness,
    NMR
  ) %>%
  tbl_summary(
    by = treatment_cat,
    label = list(
      cpd_ps = "Cigarettes per day at baseline phone survey",
      ftcd_score = "FTCD score at baseline",
      ftcd.5.mins = "Smoking within 5 mins of waking up (Yes)",
      bdi_score_w00 = "BDI score at baseline",
      crv_total_pq1 = "Cigarette reward value at baseline",
      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"
    ), type = list(
      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"),
    value = list(
      Only.Menthol ~ "1",
      ftcd.5.mins ~ "1"),
    statistic = list(all_continuous() ~ "{mean} ({sd})", all_categorical() ~ "{n} ({p}%)"),
    digits = all_continuous() ~ 1,
    missing = "no"
  ) %>% add_overall()

# Psychiatric table
psychiatric_table <- table1_data %>%
  dplyr::select(
    treatment_cat,
    shaps_score_pq1,
    otherdiag,
    antidepmed,
    mde_curr
  ) %>%
  tbl_summary(
    by = treatment_cat,
    label = list(
      shaps_score_pq1 = "Anhedonia",
      otherdiag = "Other lifetime DSM-5 diagnosis (Yes)",
      antidepmed = "Taking antidepressant medication at baseline (Yes)",
      mde_curr = "Current vs past MDD (Yes)"
    ),
    type = list(shaps_score_pq1 ~ "continuous",
      otherdiag ~ "dichotomous",
      antidepmed ~ "dichotomous",
      mde_curr ~ "dichotomous"),
    value = list(otherdiag ~ "1",
      antidepmed ~ "1",
      mde_curr ~ "1"),
    statistic = list(all_continuous() ~ "{mean} ({sd})",
                     all_categorical() ~ "{n} ({p}%)"),
    digits = all_continuous() ~ 1,
    missing = "no"
  ) %>% add_overall()


# Merge tables
final_table <- tbl_stack(
  tbls = list(demographics_table, smoking_table, psychiatric_table),
  group_header = c("Demographics", "Smoking", "Psychiatric")
) %>%
  modify_caption("Participant characteristics by treatment and overall sample") %>%
  as_kable_extra(
    booktabs = TRUE,
```

```
    longtable = TRUE,
    linesep = "",
    format = "latex"
  ) %>%
  kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15",
    font_size = 8
  )


final_table
#Missingness Table
# Calculate missing values for each variable
missing_summary <- data %>%
  summarise(across(everything(), ~ sum(is.na(.)), .names = "missing_{col}")) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Number") %>%
  mutate(Pct = (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",
  "crv_total_pqr" = "Cigarette reward at baseline",
  "shaps_score_pqi" = "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))

# Load knitr for table formatting (optional)
library(knitr)

# Create and display the table
missing_summary %>%
  arrange(desc(Pct)) %>%
  mutate(Pct = sprintf("%.2f%%", Pct)) %>%
  kable(col.names = c("Variable", "Number", "Pct"), caption = "Summary
        of Missing Values")
# 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"
  )))
```

```r
# MICE
# Perform MICE imputation
data_mice <- mice(data, m = 5
                  , method = "pmm", maxit = 50, seed = 815, printFlag= FALSE)

data_mice_4 = complete(data_mice, action = 4)

# Check the relationship between Menthol Cigarettes and Race
table_race_menthol <- table(data_mice_4$race, data_mice_4$Only.Menthol)

# Chi-square test
chi_square_test <- chisq.test(table_race_menthol) #p-value too small
#We reject the null hypothesis (which is there is no association)

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

kable(table_race_menthol,
      caption = "Contingency Table of Only Menthol Use and Race",
      col.names = c("Non-Menthol", "Menthol"),
      row.names = TRUE,
      format = "markdown")  %>%
  footnote(chi_square_test_note, footnote_as_chunk = FALSE)

# Mosaic between education, income
# mosaic = vcd::mosaic(~ inc + edu_merged, data = data_mice_4, shade = TRUE,
#                      legend = TRUE, labeling = labeling_values)

data_viz = data_mice_4 %>%
  mutate(
    edu_merged = recode(factor(edu_merged),
                        `1` = "<= High school",
                        `2` = "Some college",
                        `3` = "College graduate"),
    inc = recode(factor(inc),
                 `1` = "Less than $20,000",
                 `2` = "$20,000-$35,000",
                 `3` = "$35,001-$50,000",
                 `4` = "$50,001-$75,000",
                 `5` = "More than $75,000"),
    race = factor(race, levels = c("Non-Hispanic White",
                                   "Black",
                                   "Hispanic",
                                   "Other"))
    )

# Heatmap between education, income
heatmap1 = ggplot(data = data_viz) +
  geom_mosaic(aes(weight = 1, x = product(inc), fill = edu_merged)) +
  labs(
    title = "           Mosaic Plot between Education and Income",
```

```
    x = "Income",
    y = "Education"
  ) +
  scale_fill_manual(values = c("slateblue3", "grey", "palevioletred")) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)
  )

# Heatmap between race, education
heatmap2 = ggplot(data = data_viz) +
  geom_mosaic(aes(weight = 1, x = product(race), fill = edu_merged)) +
  labs(
    title = "Mosaic Plot between Education and Race",
    x = "Race",
    y = "Education"
  ) +
  scale_fill_manual(values = c("slateblue3", "grey", "palevioletred")) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)
  )

# Beeswarm plot between Sex, NMR
#beeswarm = ggplot(data_mice_4, aes(x = sex_ps, y = NMR, color = sex_ps)) +
 # geom_beeswarm(size = 2, alpha = 0.7) +
  #labs(x = "Sex", y = "NMR", title = "Beeswarm Plot") +
  #theme_minimal() +
  #theme(legend.position = "none") +
  #scale_color_manual(values = c("slateblue3", "palevioletred"))



# Raincloud plot between edu, income
raincloud = ggplot(data_mice_4, aes(x = race, y = ftcd_score, fill = race)) +
  ggdist::stat_halfeye(adjust = 0.5, width = 0.6, .width = 0,
                       justification = -0.2, point_colour = NA) +
  geom_boxplot(width = 0.1, outlier.shape = NA, alpha = 0.5) +
  geom_jitter(width = 0.1, alpha = 0.4) +
  coord_flip() +  # Flip coordinates for a horizontal layout
  labs(x = "Race", y = "FTCD Score", title = "Raincloud Plot") +
  theme_minimal() +
  scale_fill_manual(values = c("lightblue","slateblue3", "grey", "palevioletred"))

title <- ggdraw() +
  draw_label("Figure 1: Mosaic/Raincloud plots for the relationships between education, income, race, a
fontface = "bold",
x = 0, hjust = 0)
```

```r
combined_plot <- plot_grid(heatmap1, heatmap2, ncol = 2, align = "hv")


combined_plot_title <- plot_grid(title, combined_plot, ncol = 1, rel_heights = c(0.1, 1))
combined_plot_title

#combined_plot <- grid.arrange(mosaic, heatmap, beeswarm, raincloud, ncol = 2)
#combined_plot
#par(mfrow= c(2,2))
#mosaic

raincloud
# 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",

  "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", "BA1:cpd_ps",


  "Var1:BA1",
  "Var1:age_ps", "Var1:sex_ps2",
  "Var1:raceBlack", "Var1:raceHispanic",
  "Var1:raceOther", "Var1:ftcd_score", "Var1:cpd_ps",

   "inc2:edu_merged2", "inc2:edu_merged3",
  "inc3:edu_merged2", "inc3:edu_merged3",
  "inc4:edu_merged2", "inc4:edu_merged3",
  "inc5:edu_merged2", "inc5:edu_merged3",
  "antidepmed1:readiness", "Only.Menthol1:readiness",
  "mde_curr1:readiness", "ftcd.5.mins1:readiness",
  "bdi_score_w00:readiness", "Var1:shaps_score_pq1", "shaps_score_pq1:mde_curr1",
  "sex_ps2:ftcd_score", "raceBlack:ftcd_score",
  "raceHispanic:ftcd_score", "raceOther:ftcd_score",
  "age_ps:ftcd_score", "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",
```

```
  "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")]


  # Split into train and test sets
  set.seed(815)
  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))
```

```r
  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(data_mice
                      , 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
```

```r
  # 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)

# 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[,c(1,2,4)], enet_df[,c(1,2,4)], by = "variable") %>%
  mutate(across(where(is.numeric), ~ round(.x, 4)))

# 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()
```

```r
# 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 OR", "Selection Times", "Average OR"),
        caption = "Summary of Average Coefficients and Odds Ratios for Potential Moderator Effects acros
  add_header_above(c(" " = 1, "Lasso" = 2, "Elastic Net" = 2)) %>%
  kable_styling(full_width = F, position = "center", font_size = 9,
                latex_options = c("striped"),
                stripe_color = "gray!15")

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

roc_results_lasso <- list()
calib_results_lasso <- list()
auc_values_lasso <- list()

num_cuts <- 10  # Number of bins for calibration

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

  # Split the dataset into training and test sets
  set.seed(815)
  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)
```

```r
# 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) +
  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())
  )

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 = "purple") +
  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 = "black", 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 = "purple") +  # Perfect calibration line
  scale_color_manual(values = c("Ideal" = "purple",
                                "Flexible calibration" = "black")) +
  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 = "purple") +
  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 = "black", 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 = "purple") +  # Perfect calibration line
  scale_color_manual(values = c("Ideal" = "purple",
                                "Flexible calibration" = "black")) +
  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))


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)
))
```

```r
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
)
### Second objective - Predictors

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")

# Lasso Regression
# To identify the potential interaction terms for moderator effects
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"
)

# 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")]
```

```r
  # Split into train and test sets
   set.seed(815)
   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
  )
}
```

```r
# Fit models across imputed datasets
aim2_results <- list()
for (i in 1:5) {
  dataset <- complete(data_mice, 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
  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)

# 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
```

```r
combined_df <- full_join(lasso_df[,c(1,2,4)], enet_df[, c(1,2,4)], by = "variable") %>%
  mutate(across(where(is.numeric), ~ round(.x, 4)))

# 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 OR", "Selection Times", "Average OR"),
        caption = "Summary of Average Coefficients and Odds Ratios for Potential Predictor Effects acros
  add_header_above(c(" " = 1, "Lasso" = 2, "Elastic Net" = 2)) %>%
  kable_styling(full_width = F, position = "center", font_size = 9,
                latex_options = c("striped"),
                stripe_color = "gray!15")

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

roc_results_lasso <- list()
calib_results_lasso <- list()
auc_values_lasso <- list()

num_cuts <- 10  # Number of bins for calibration

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

```r
# Split the dataset into training and test sets
set.seed(815)
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[, predictor_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_predictors_dummy_inclu
test_outcome <- test_data$abst

# Predict probabilities for Elastic Net
predicted_prob_enet <- as.numeric(predict(aim2_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(aim2_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) +
  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, c
  labs(
    x = "1 - Specificity",
    y = "Sensitivity"
  ) +
  theme_minimal()
```

```r
# 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, co
  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())
  )

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 = "purple") +
  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 = "black", 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 = "purple") +  # Perfect calibration line
  scale_color_manual(values = c("Ideal" = "purple",
                                "Flexible calibration" = "black")) +
  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 = "purple") +
  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 = "black", 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 = "purple") +  # Perfect calibration line
  scale_color_manual(values = c("Ideal" = "purple",
                                "Flexible calibration" = "black")) +
  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))


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 3: Calibration Plots with Error Bars and LOESS and ROC Curves (Predictor Effects)",
  gp = gpar(fontsize = 16)
)

# Arrange everything with the bold title
grid.arrange(
  plots_lasso,
  plots_enet,
  nrow = 2,
  top = main_title
)
```