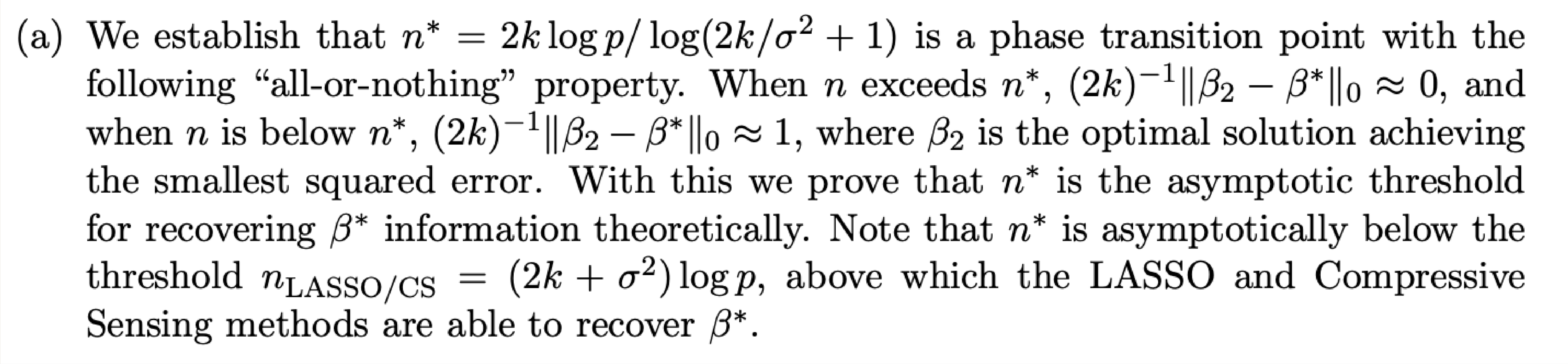
Group Meeting representation

SUMMER RESEARCH PROJECT

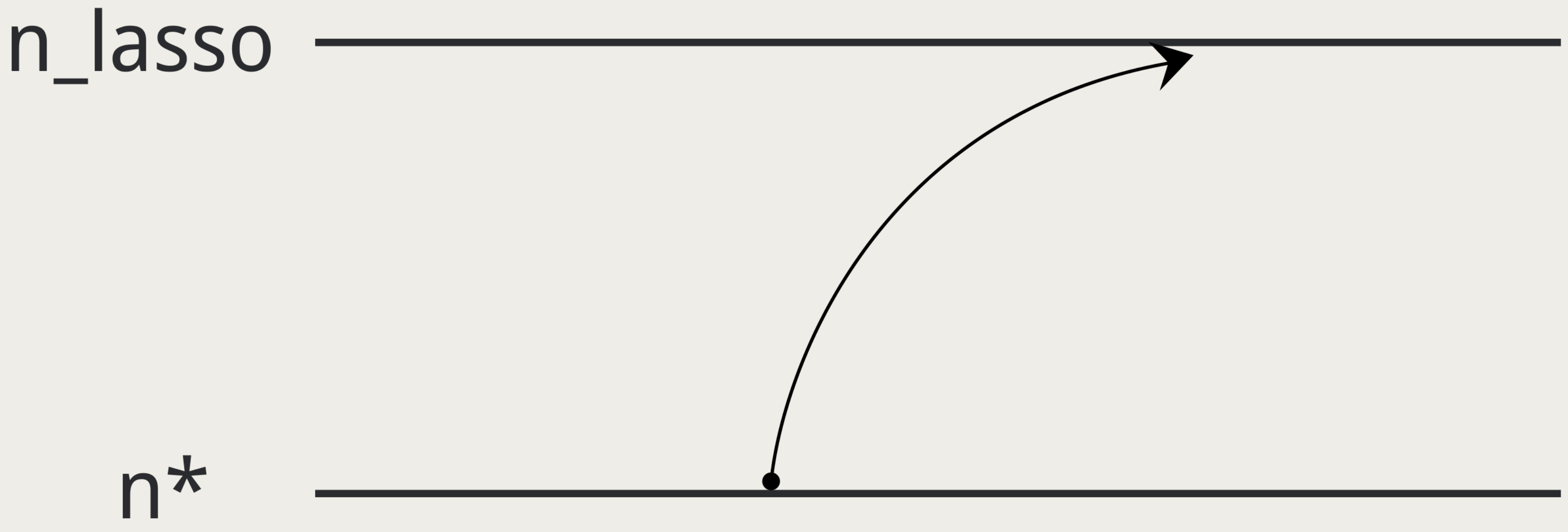
Yan Liu

8th Feb, 2024

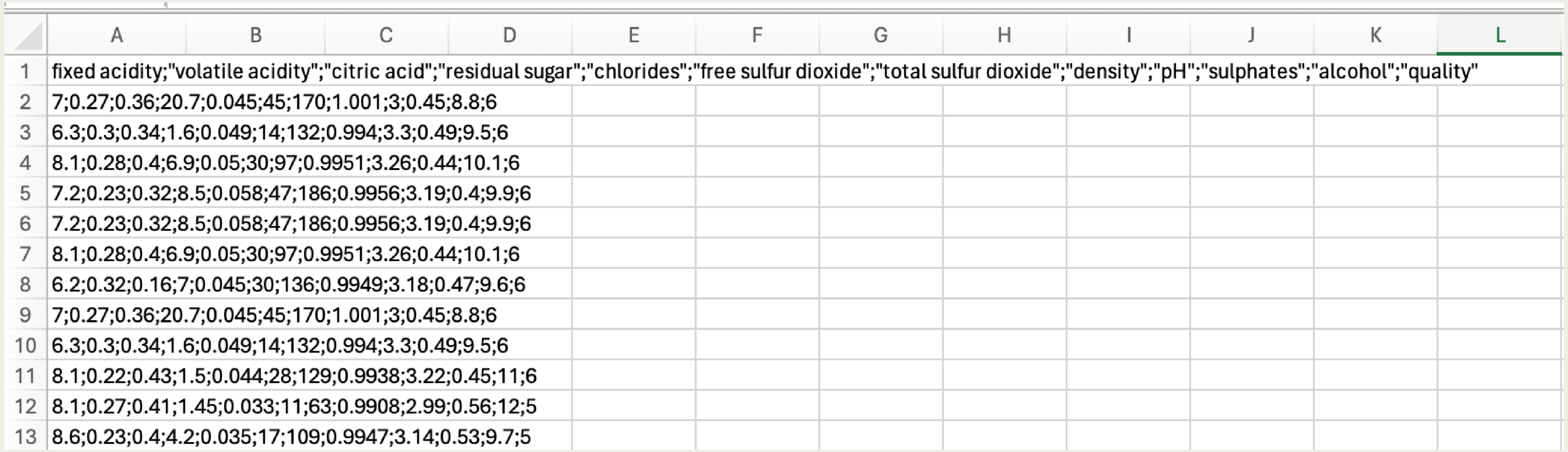
HIGH-DIMENSIONAL REGRESSION WITH BINARY COEFFICIENTS. ESTIMATING SQUARED ERROR AND THE PHASE TRANSITION



(a) We establish that $n^* = 2k \log p / \log(2k/\sigma^2 + 1)$ is a phase transition point with the following "all-or-nothing" property. When n exceeds n^*, $(2k)^{-1}\|\beta_2 - \beta^*\|_0 \approx 0$, and when n is below n^*, $(2k)^{-1}\|\beta_2 - \beta^*\|_0 \approx 1$, where β_2 is the optimal solution achieving the smallest squared error. With this we prove that n^* is the asymptotic threshold for recovering β^* information theoretically. Note that n^* is asymptotically below the threshold $n_{\mathrm{LASSO/CS}} = (2k + \sigma^2) \log p$, above which the LASSO and Compressive Sensing methods are able to recover β^*.



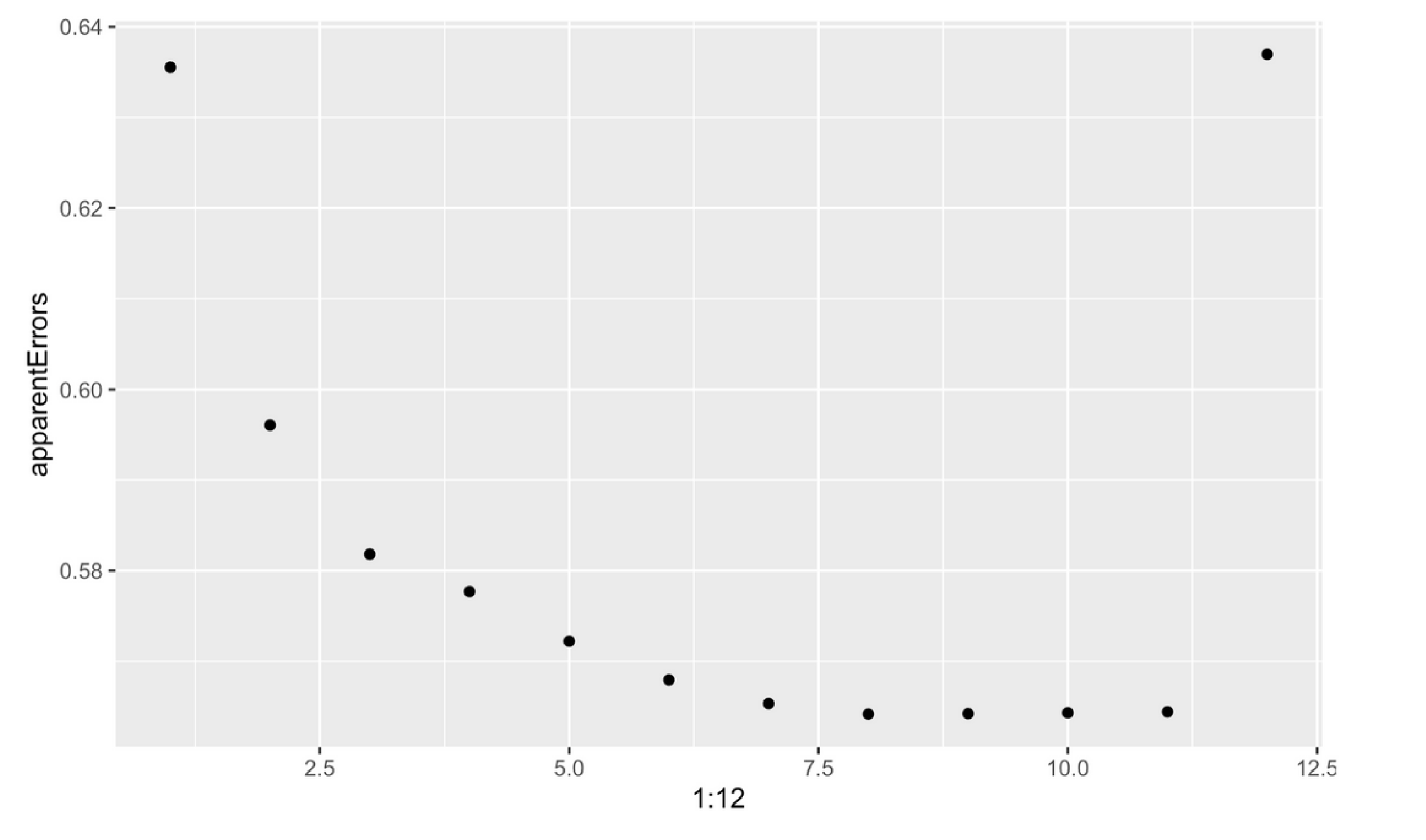
DATA SET

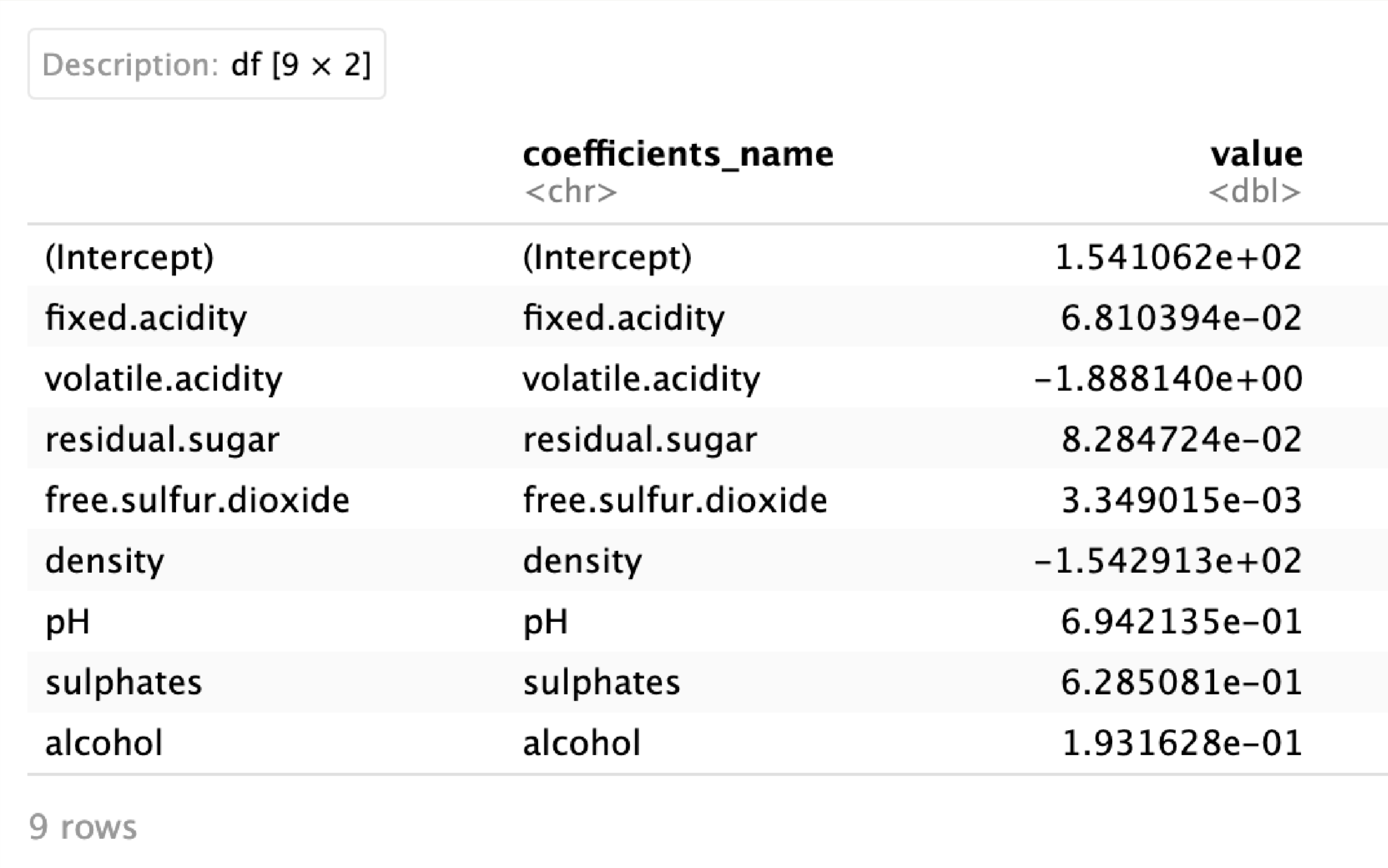


	A	B	C	D	E	F	G	H	I	J	K	L
1	fixed acidity;"volatile acidity";"citric acid";"residual sugar";"chlorides";"free sulfur dioxide";"total sulfur dioxide";"density";"pH";"sulphates";"alcohol";"quality"											
2	7;0.27;0.36;20.7;0.045;45;170;1.001;3;0.45;8.8;6											
3	6.3;0.3;0.34;1.6;0.049;14;132;0.994;3.3;0.49;9.5;6											
4	8.1;0.28;0.4;6.9;0.05;30;97;0.9951;3.26;0.44;10.1;6											
5	7.2;0.23;0.32;8.5;0.058;47;186;0.9956;3.19;0.4;9.9;6											
6	7.2;0.23;0.32;8.5;0.058;47;186;0.9956;3.19;0.4;9.9;6											
7	8.1;0.28;0.4;6.9;0.05;30;97;0.9951;3.26;0.44;10.1;6											
8	6.2;0.32;0.16;7;0.045;30;136;0.9949;3.18;0.47;9.6;6											
9	7;0.27;0.36;20.7;0.045;45;170;1.001;3;0.45;8.8;6											
10	6.3;0.3;0.34;1.6;0.049;14;132;0.994;3.3;0.49;9.5;6											
11	8.1;0.22;0.43;1.5;0.044;28;129;0.9938;3.22;0.45;11;6											
12	8.1;0.27;0.41;1.45;0.033;11;63;0.9908;2.99;0.56;12;5											
13	8.6;0.23;0.4;4.2;0.035;17;109;0.9947;3.14;0.53;9.7;5											

4898 observations and 12 variables

OPTIMAL FEATURE SUBSET SELECTION FOR MULTIPLE LINEAR REGRESSION MODELS



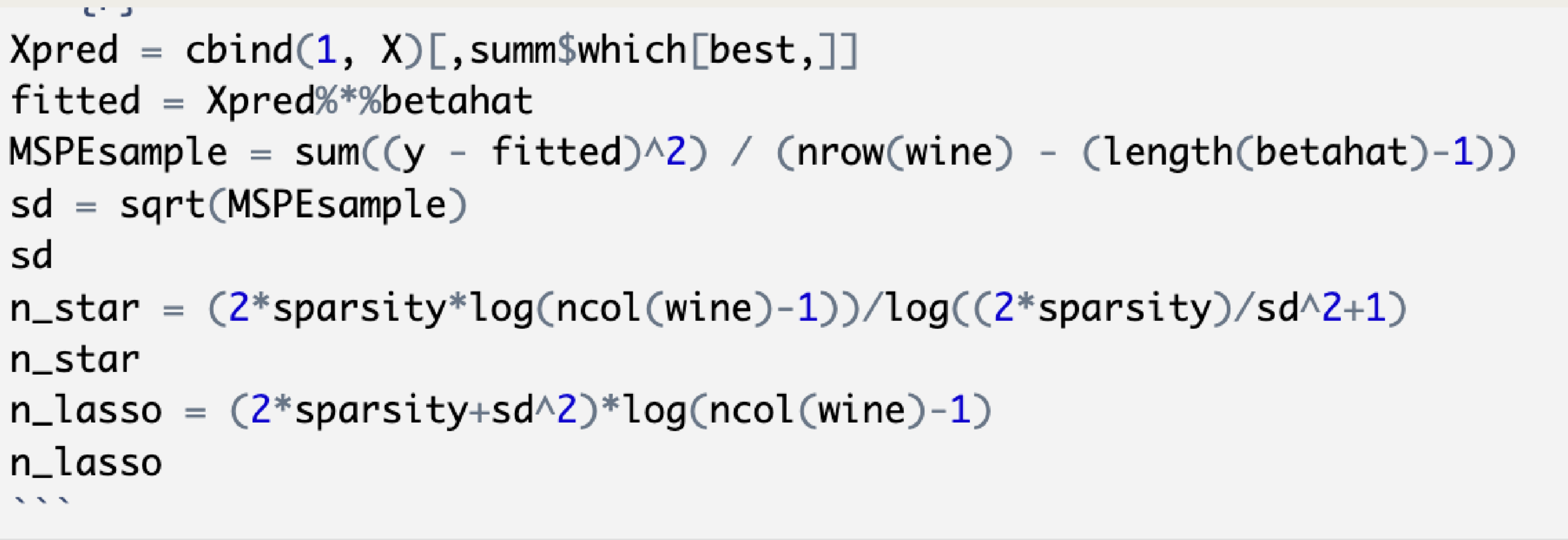


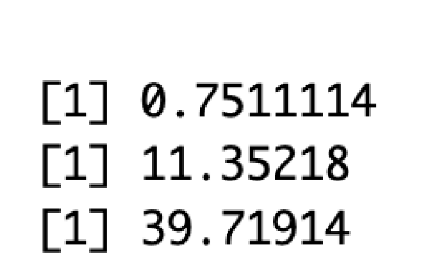
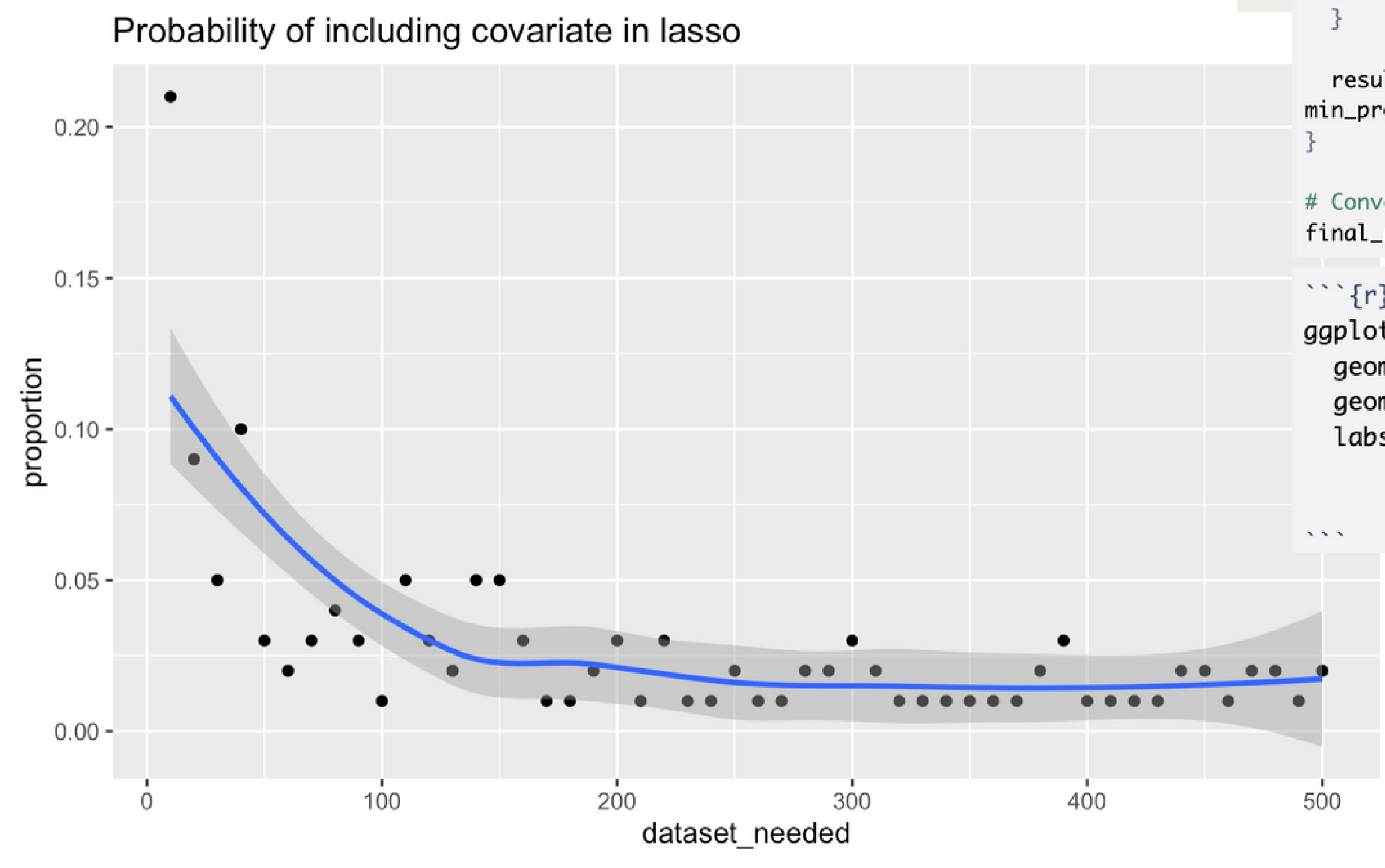
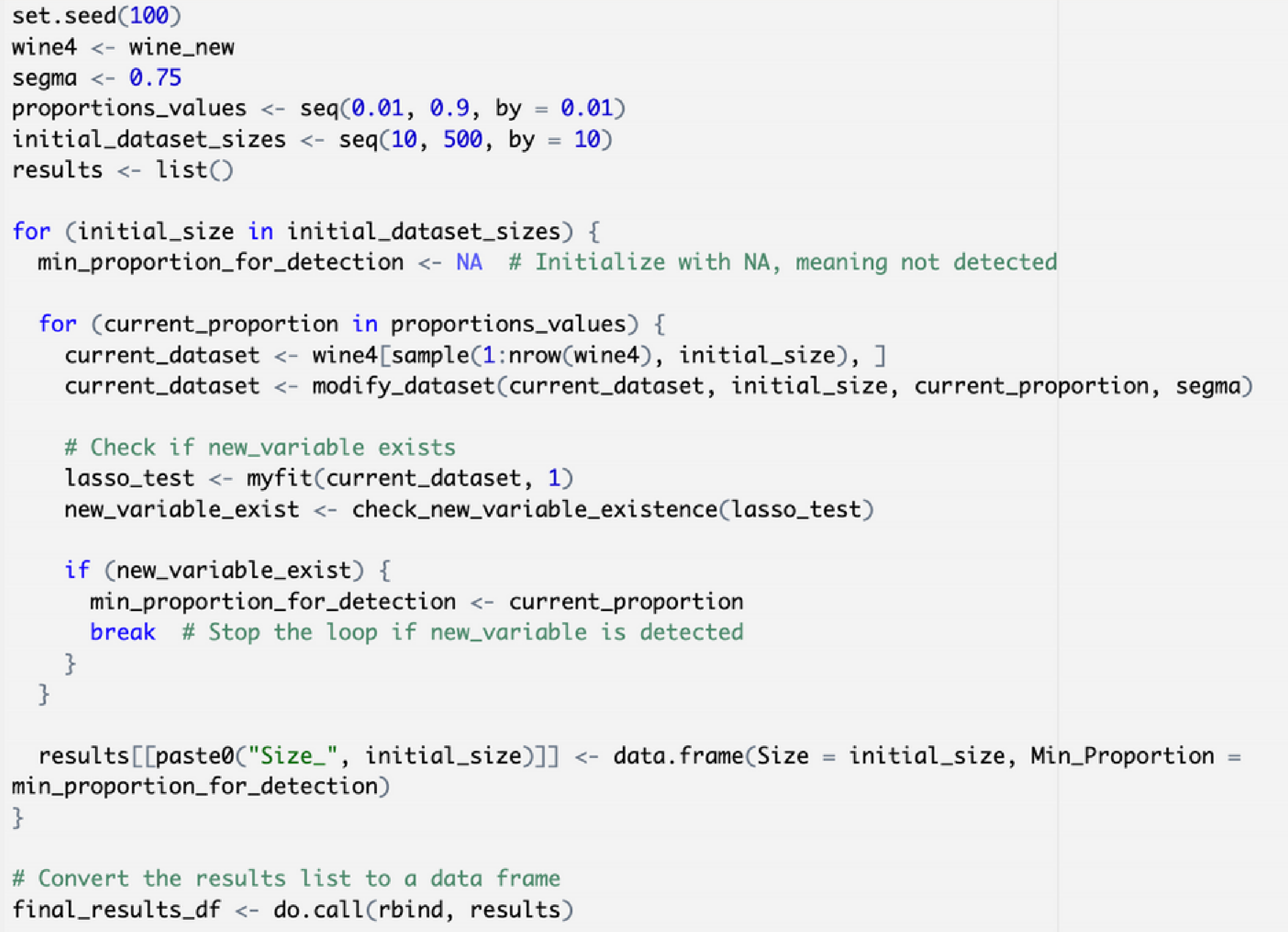
Description: df [9 × 2]

	coefficients_name <chr>	value <dbl>
(Intercept)	(Intercept)	1.541062e+02
fixed.acidity	fixed.acidity	6.810394e-02
volatile.acidity	volatile.acidity	-1.888140e+00
residual.sugar	residual.sugar	8.284724e-02
free.sulfur.dioxide	free.sulfur.dioxide	3.349015e-03
density	density	-1.542913e+02
pH	pH	6.942135e-01
sulphates	sulphates	6.285081e-01
alcohol	alcohol	1.931628e-01

9 rows

METHODS AND EXPERIMENTAL DESIGN



```
Xpred = cbind(1, X)[,summ$which[best,]]
fitted = Xpred%*%betahat
MSPEsample = sum((y - fitted)^2) / (nrow(wine) - (length(betahat)-1))
sd = sqrt(MSPEsample)
sd
n_star = (2*sparsity*log(ncol(wine)-1))/log((2*sparsity)/sd^2+1)
n_star
n_lasso = (2*sparsity+sd^2)*log(ncol(wine)-1)
n_lasso
```
```



```
[1] 0.7511114
[1] 11.35218
[1] 39.71914
```



Probability of including covariate in lasso
proportion
0.00
0.05
0.10
0.15
0.20
0
100
200
300
400
500
dataset_needed



```
set.seed(100)
wine4 <- wine_new
segma <- 0.75
proportions_values <- seq(0.01, 0.9, by = 0.01)
initial_dataset_sizes <- seq(10, 500, by = 10)
results <- list()

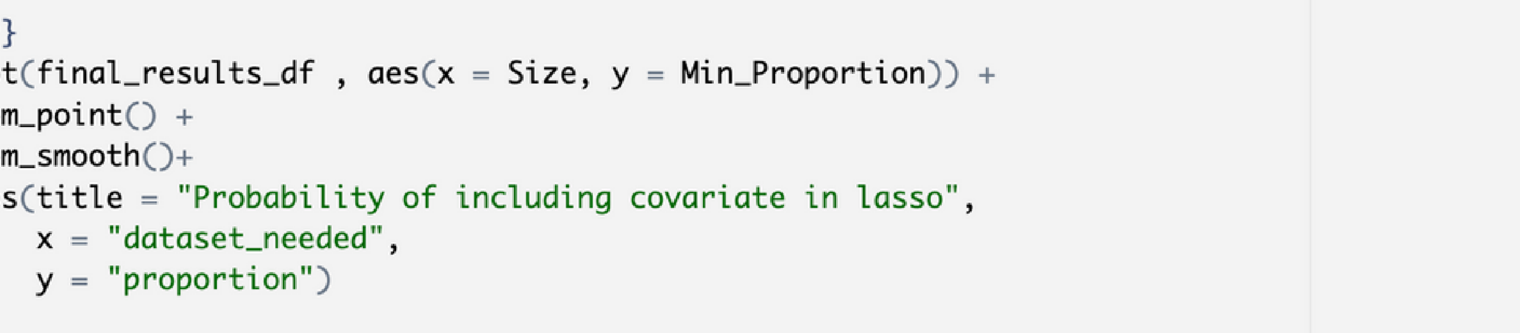
for (initial_size in initial_dataset_sizes) {
  min_proportion_for_detection <- NA  # Initialize with NA, meaning not detected

  for (current_proportion in proportions_values) {
    current_dataset <- wine4[sample(1:nrow(wine4), initial_size), ]
    current_dataset <- modify_dataset(current_dataset, initial_size, current_proportion, segma)

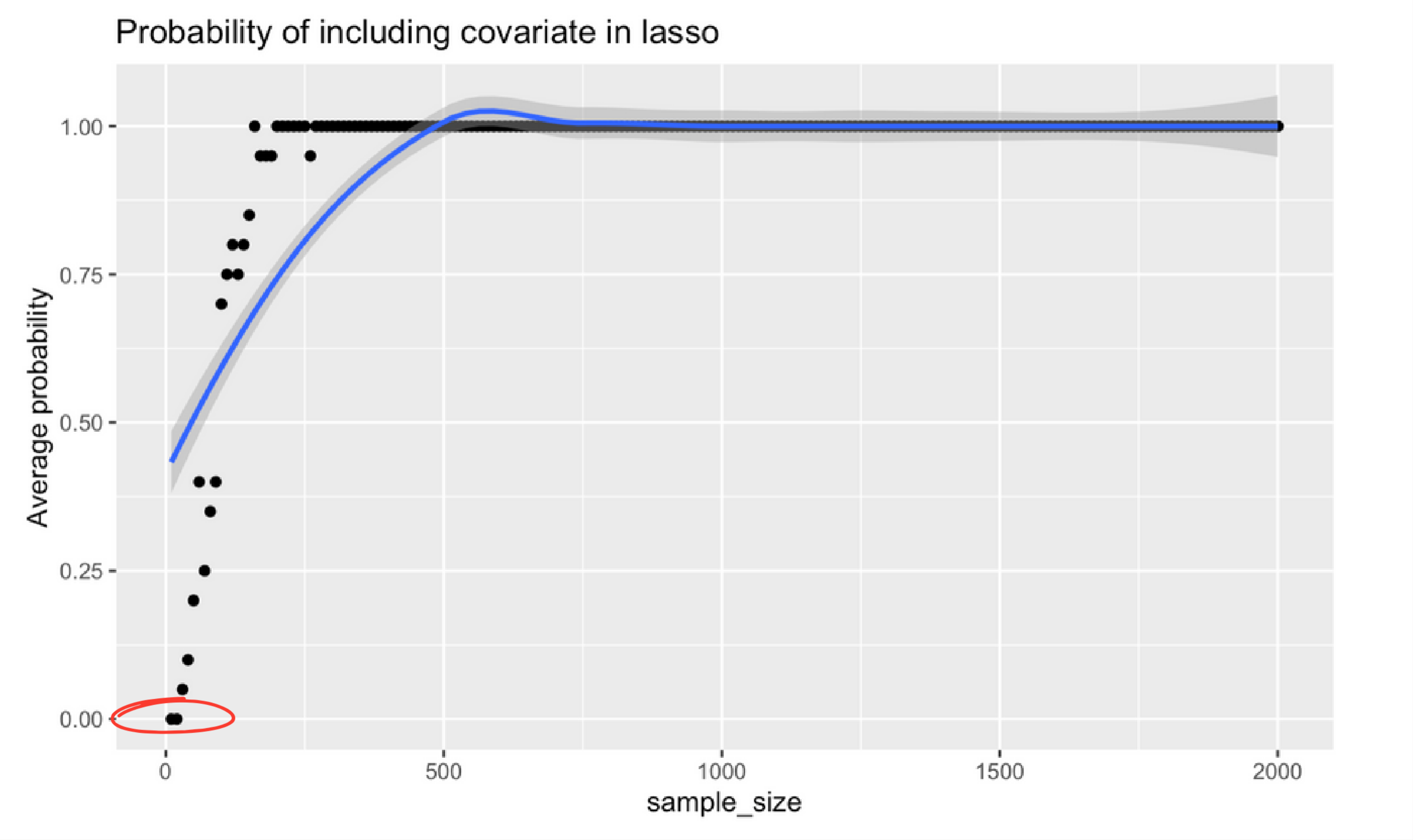
    # Check if new_variable exists
    lasso_test <- myfit(current_dataset, 1)
    new_variable_exist <- check_new_variable_existence(lasso_test)

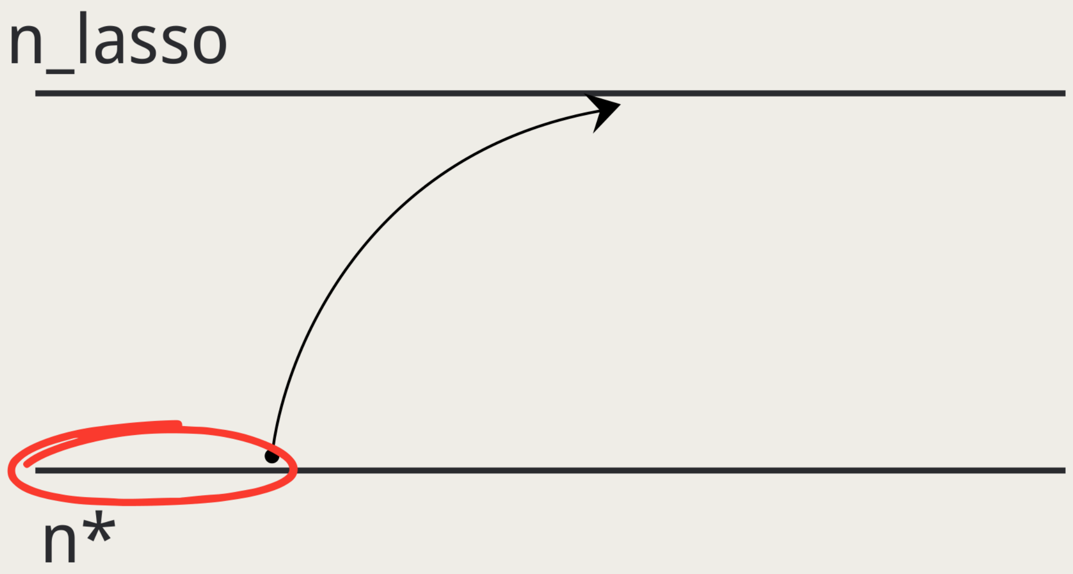
    if (new_variable_exist) {
      min_proportion_for_detection <- current_proportion
      break  # Stop the loop if new_variable is detected
    }
  }

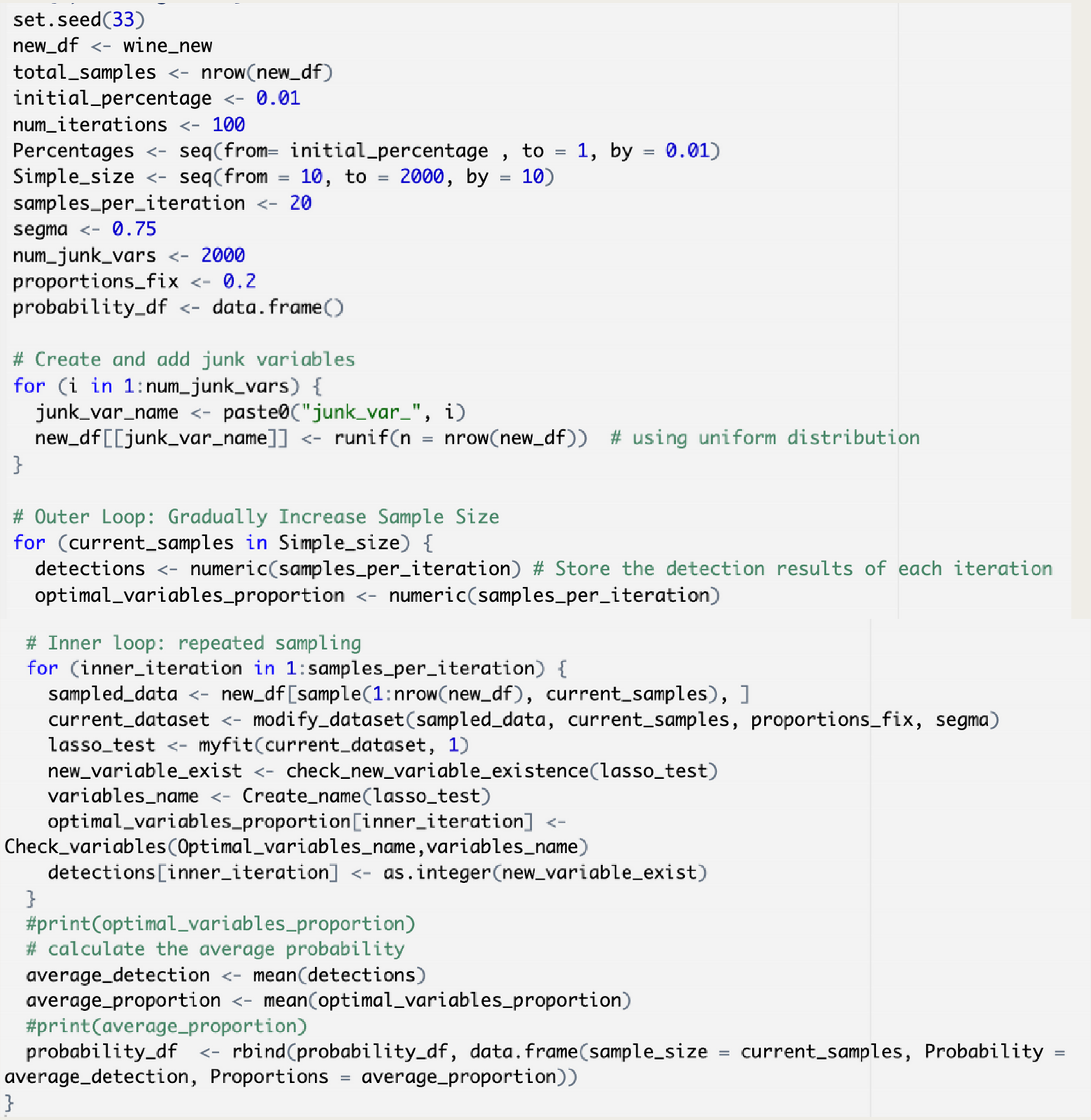
  results[[paste0("Size_", initial_size)]] <- data.frame(Size = initial_size, Min_Proportion =
min_proportion_for_detection)
}

# Convert the results list to a data frame
final_results_df <- do.call(rbind, results)
```



```
```{r}
ggplot(final_results_df , aes(x = Size, y = Min_Proportion)) +
  geom_point() +
  geom_smooth()+
  labs(title = "Probability of including covariate in lasso",
       x = "dataset_needed",
       y = "proportion")
```
```
```

METHODS AND EXPERIMENTAL DESIGN







```
set.seed(33)
new_df <- wine_new
total_samples <- nrow(new_df)
initial_percentage <- 0.01
num_iterations <- 100
Percentages <- seq(from= initial_percentage , to = 1, by = 0.01)
Simple_size <- seq(from = 10, to = 2000, by = 10)
samples_per_iteration <- 20
segma <- 0.75
num_junk_vars <- 2000
proportions_fix <- 0.2
probability_df <- data.frame()

# Create and add junk variables
for (i in 1:num_junk_vars) {
  junk_var_name <- paste0("junk_var_", i)
  new_df[[junk_var_name]] <- runif(n = nrow(new_df))  # using uniform distribution
}

# Outer Loop: Gradually Increase Sample Size
for (current_samples in Simple_size) {
  detections <- numeric(samples_per_iteration) # Store the detection results of each iteration
  optimal_variables_proportion <- numeric(samples_per_iteration)

  # Inner loop: repeated sampling
  for (inner_iteration in 1:samples_per_iteration) {
    sampled_data <- new_df[sample(1:nrow(new_df), current_samples), ]
    current_dataset <- modify_dataset(sampled_data, current_samples, proportions_fix, segma)
    lasso_test <- myfit(current_dataset, 1)
    new_variable_exist <- check_new_variable_existence(lasso_test)
    variables_name <- Create_name(lasso_test)
    optimal_variables_proportion[inner_iteration] <-
Check_variables(Optimal_variables_name,variables_name)
    detections[inner_iteration] <- as.integer(new_variable_exist)
  }
  #print(optimal_variables_proportion)
  # calculate the average probability
  average_detection <- mean(detections)
  average_proportion <- mean(optimal_variables_proportion)
  #print(average_proportion)
  probability_df  <- rbind(probability_df, data.frame(sample_size = current_samples, Probability =
average_detection, Proportions = average_proportion))
}
```

Thank you!