Optimal Pi and Importance Sampling Analysis

Contents

Overview

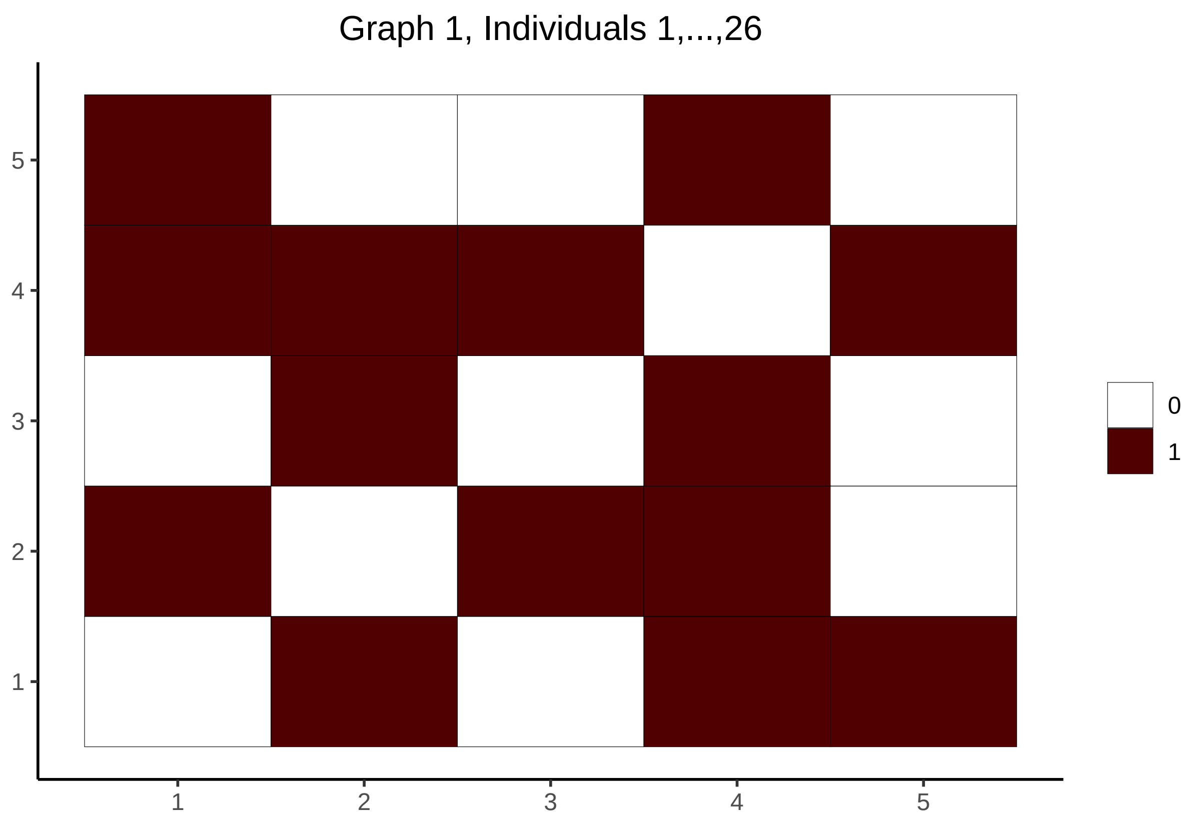
In this document, I present the results of two experiments. The first describes the distribution of the optimal hyperparameters selected by grid search. In the second, I compare the results of a novel importance sampling - grid search hybrid scheme for hyperparameter specification.

The first two main sections contain the results of these experiments, while the third describes the data and extraneous covariate generation.

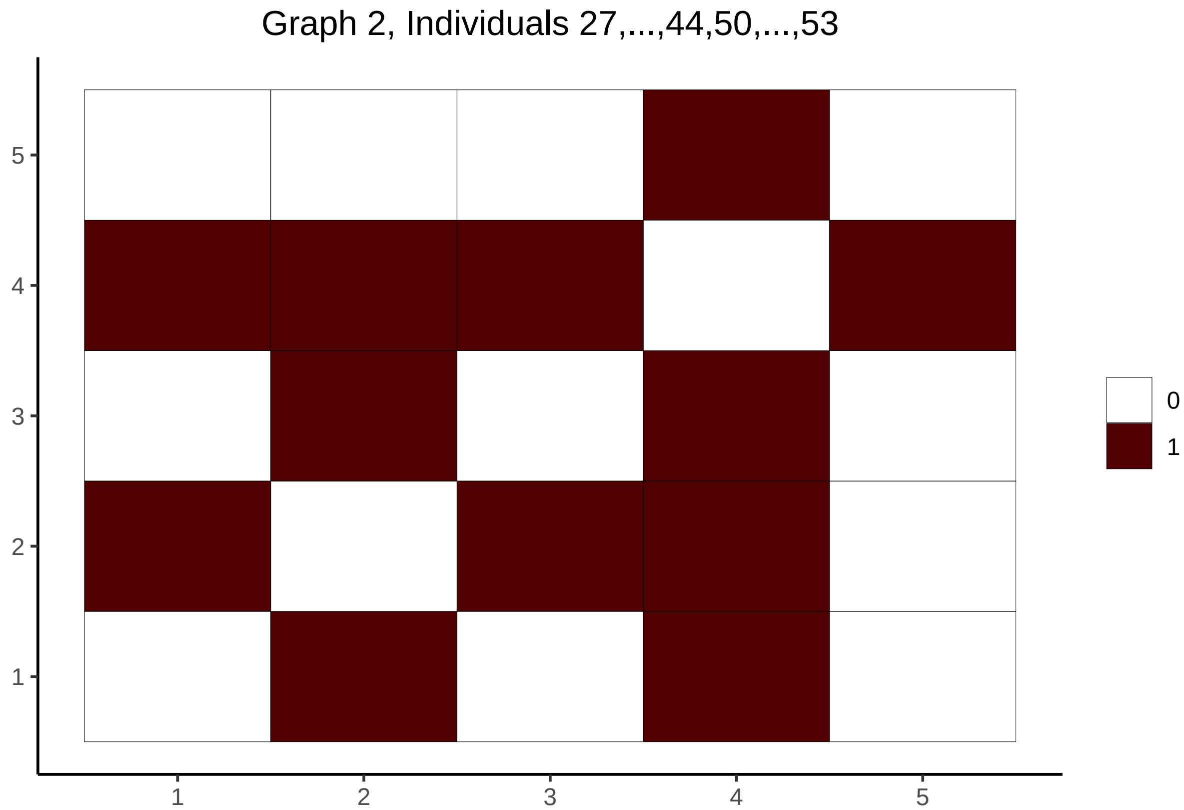
Optimal Pi

Hyperparameter specification

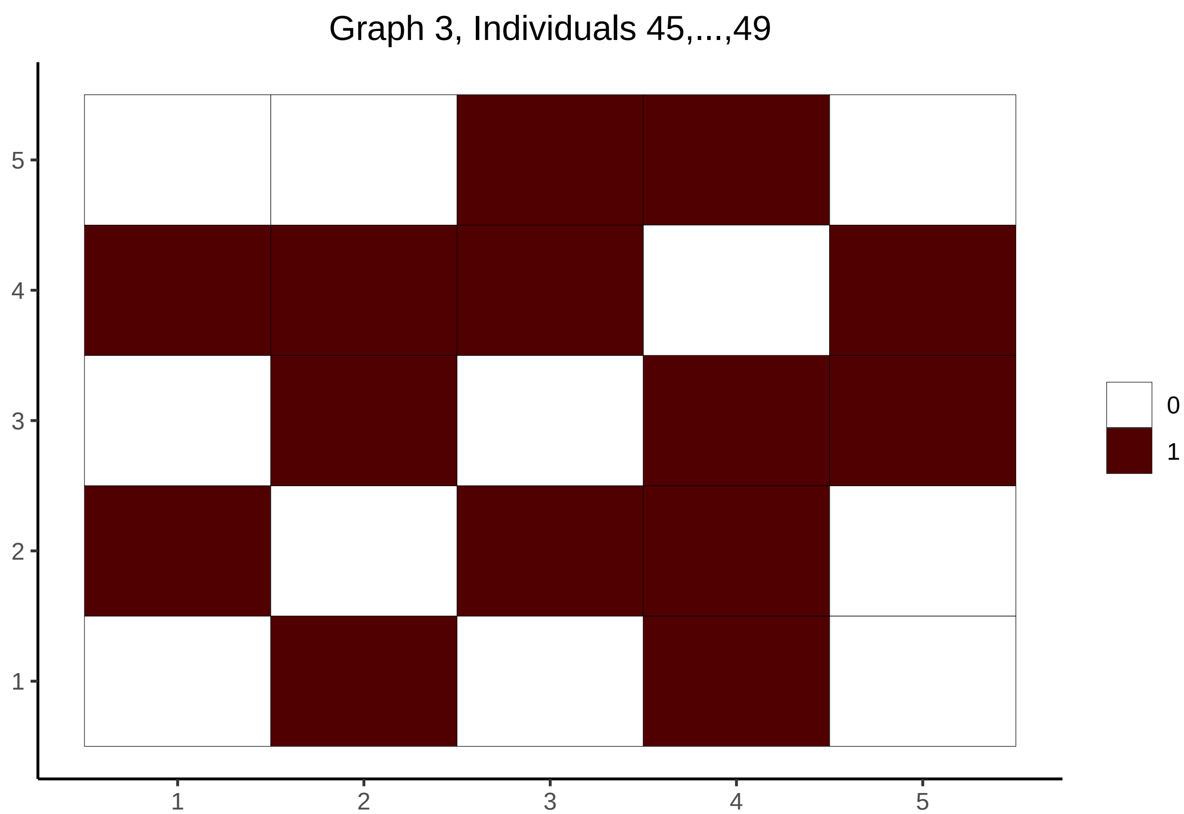
```
## [[1]]
```



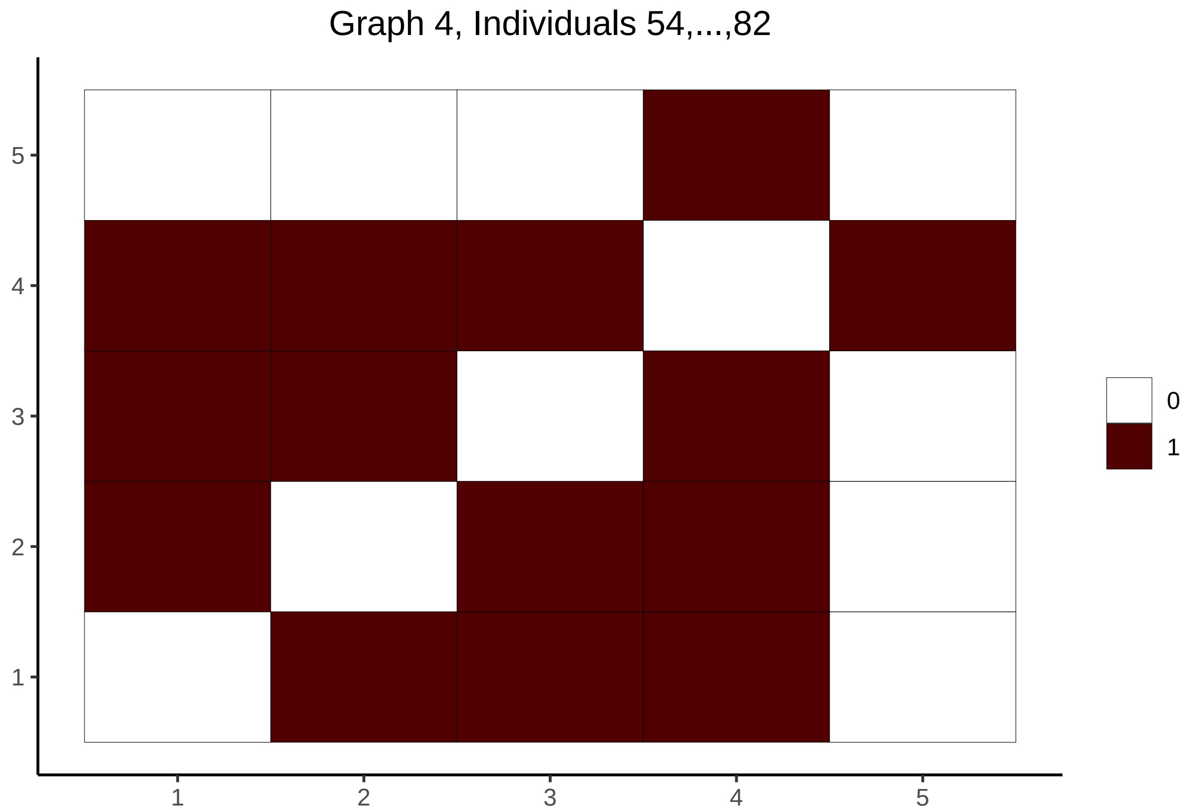

```
##
## [[2]]
```



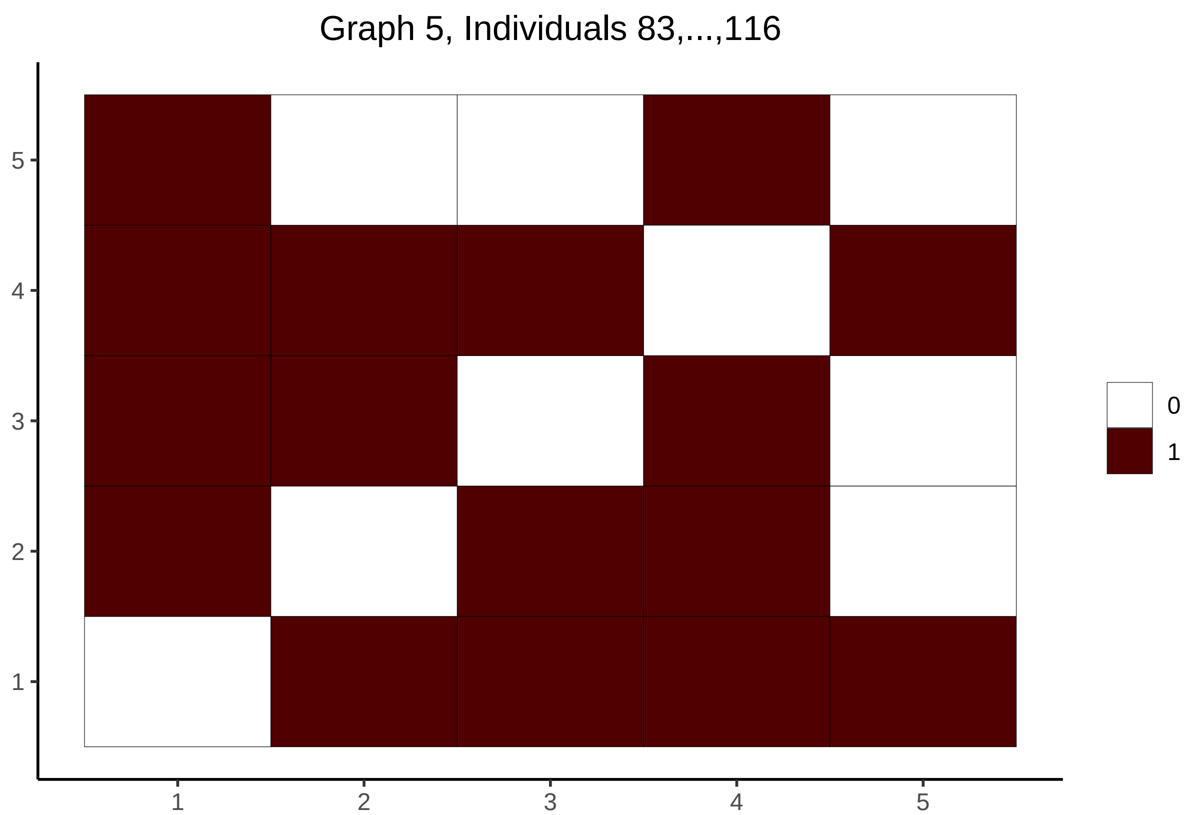

```
##
## [[3]]
```



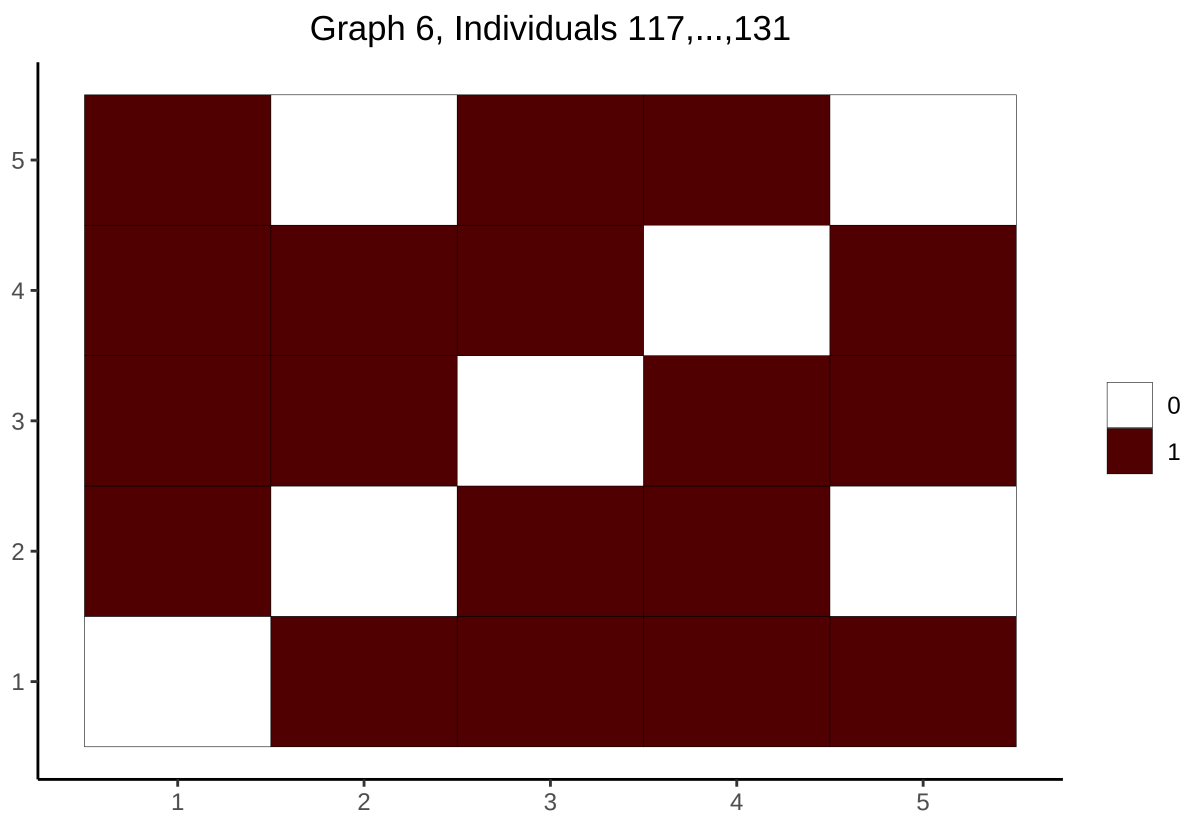

```
##
## [[4]]
```



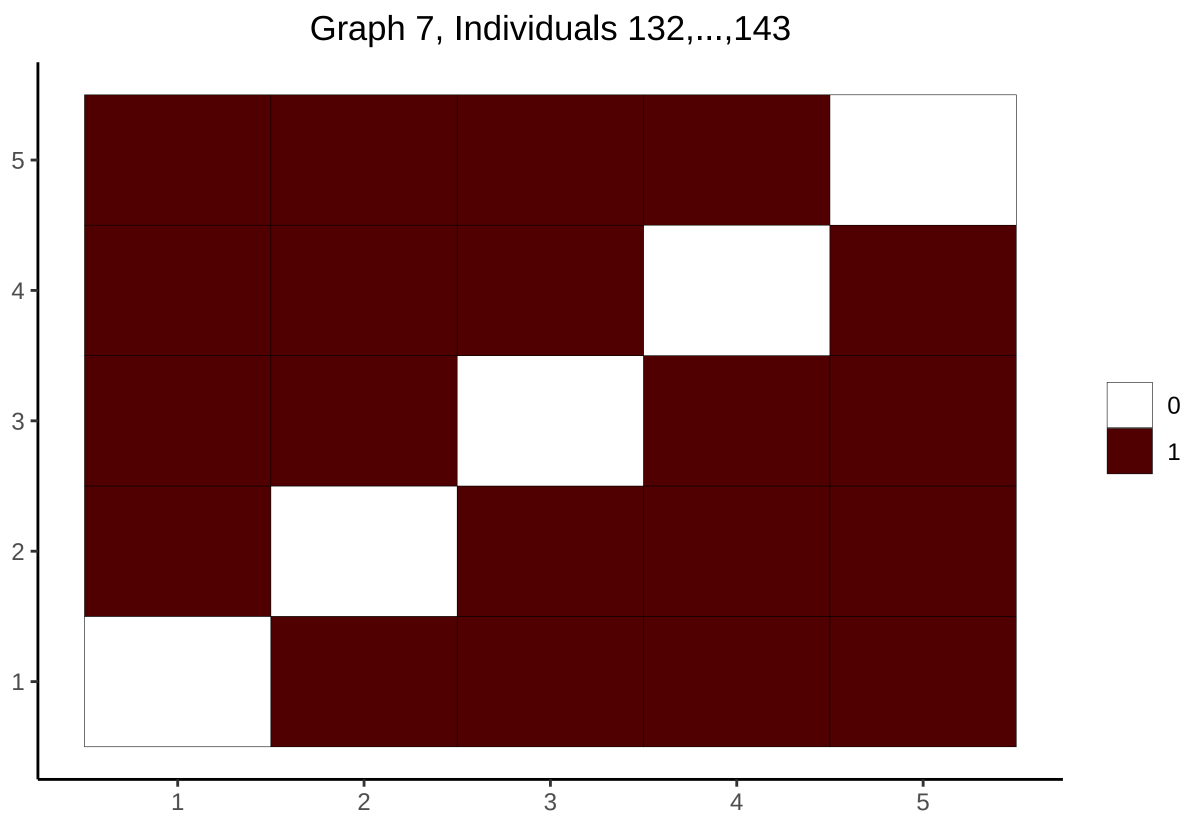

```
##
## [[5]]
```




```
##
## [[6]]
```

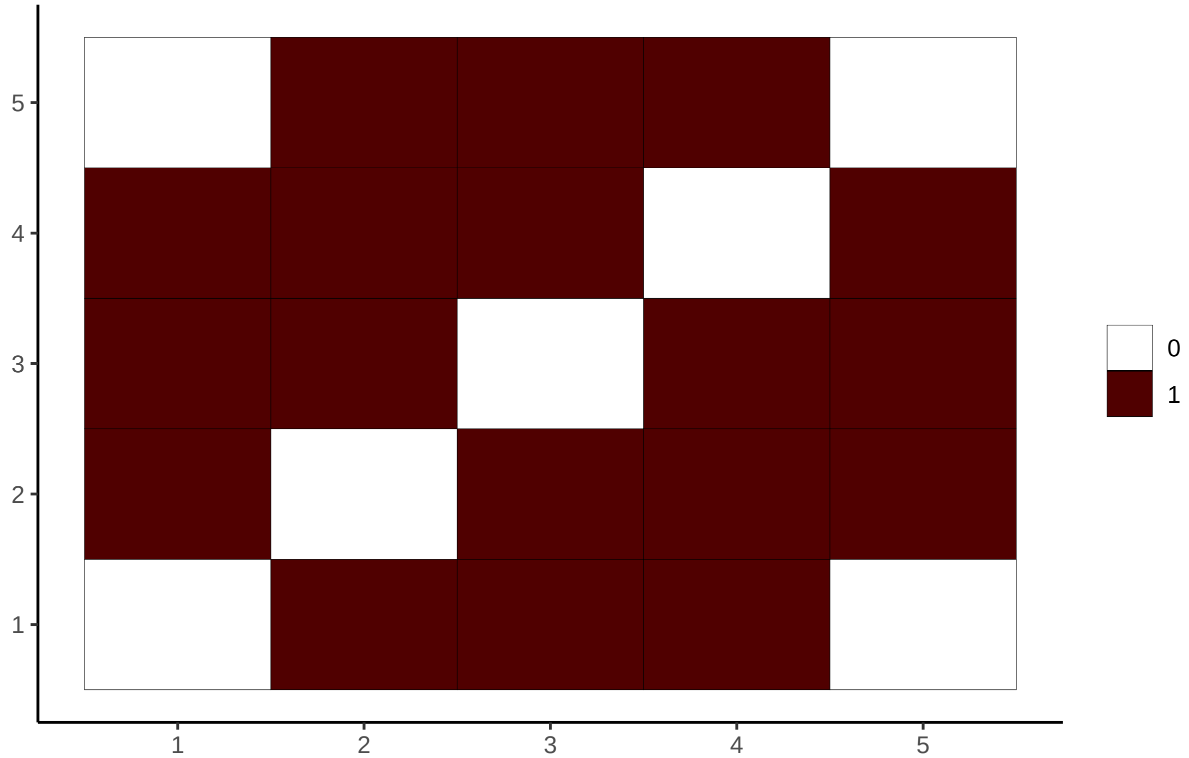



```
##
## [[7]]
```

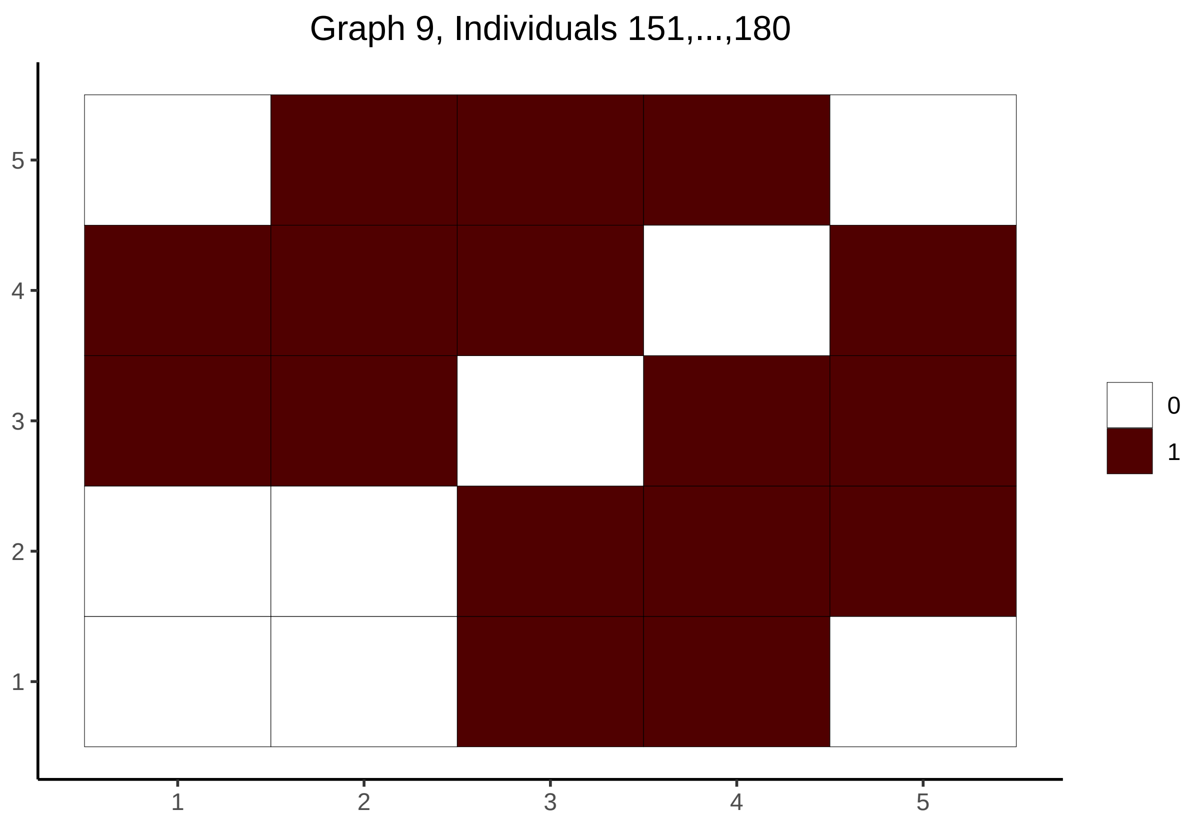



```
##
## [[8]]
```

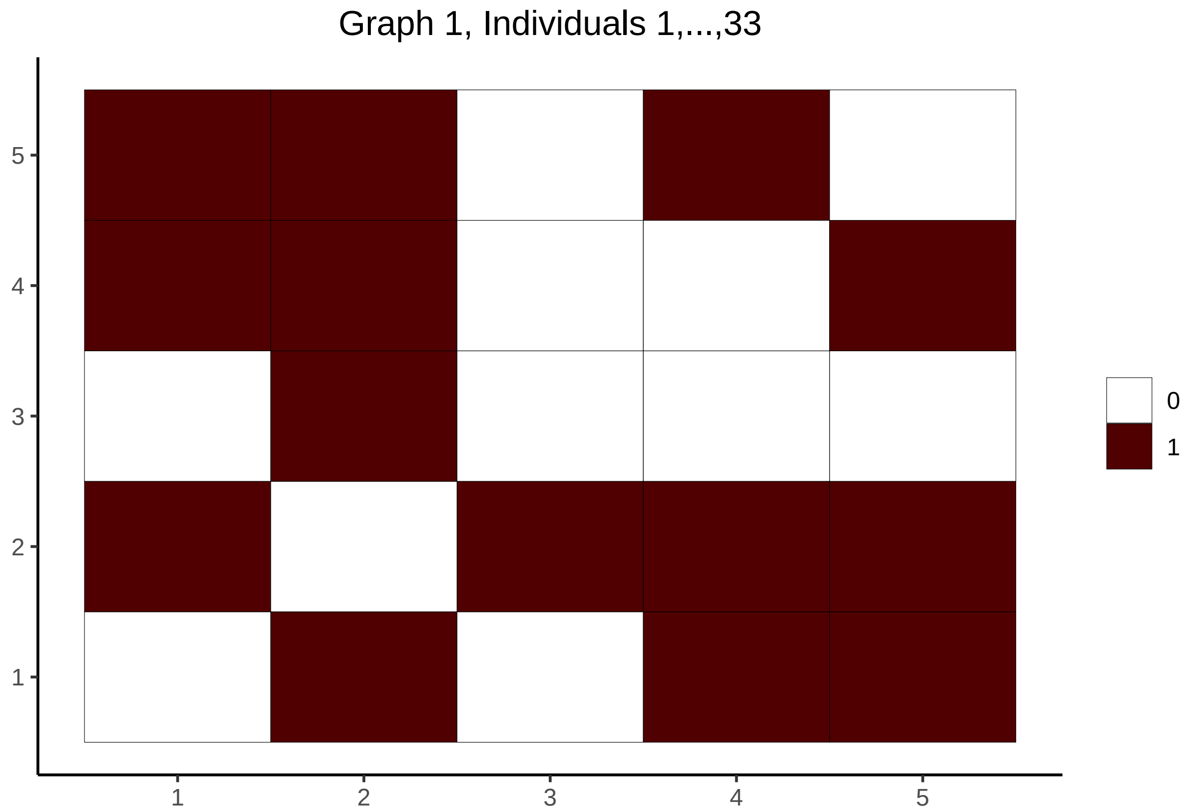
Graph 8, Individuals 144,...,150



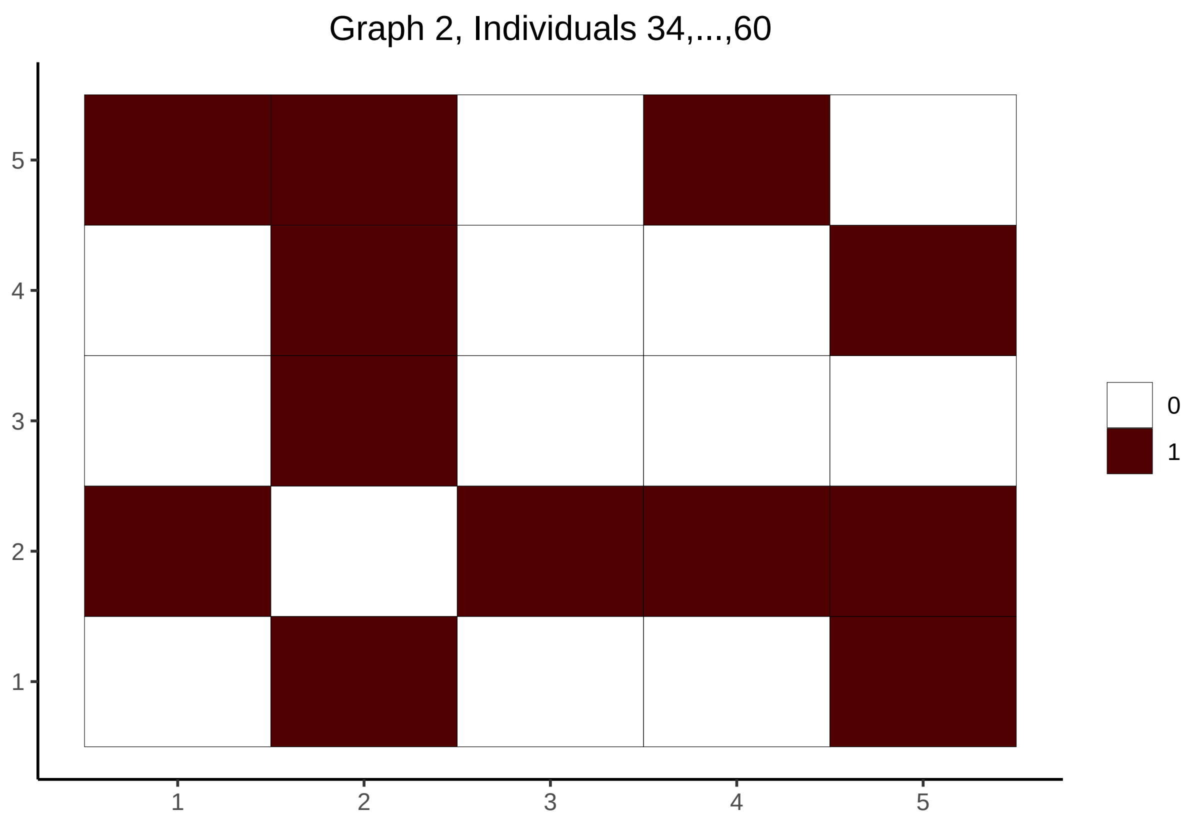

```
##
## [[9]]
```



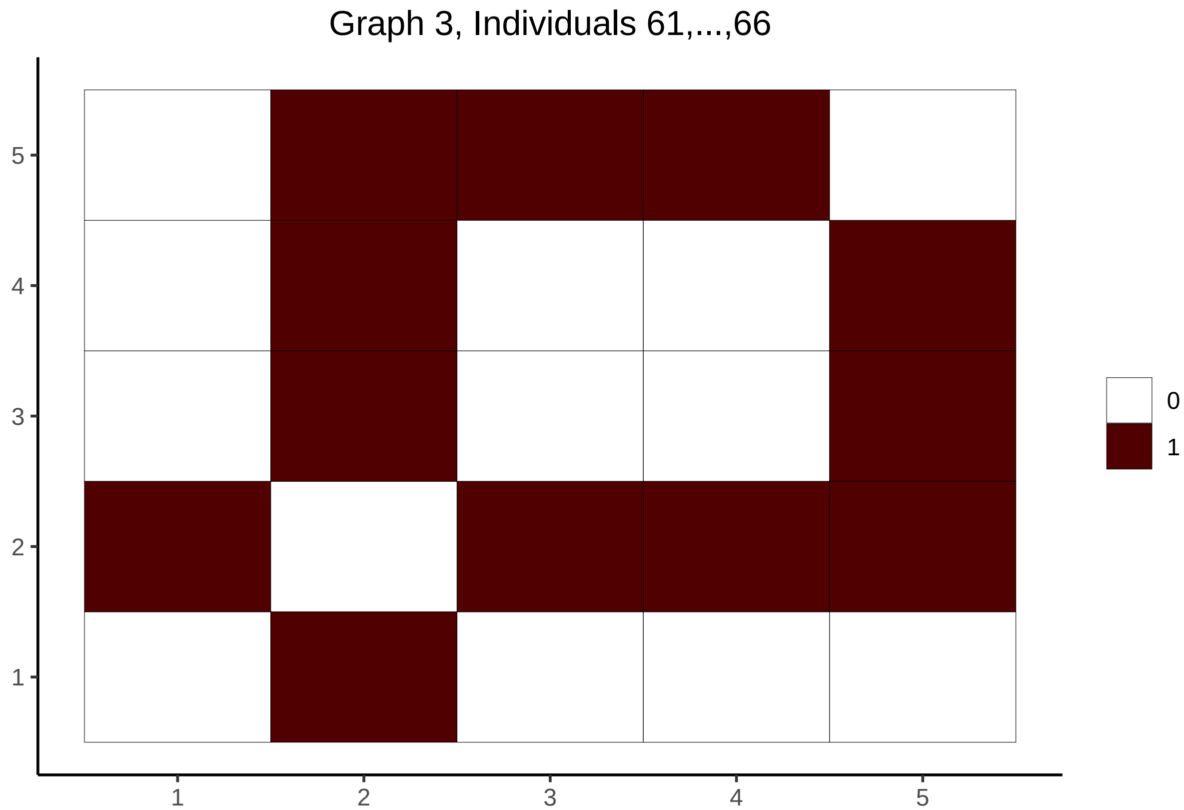

```
## [[1]]
```



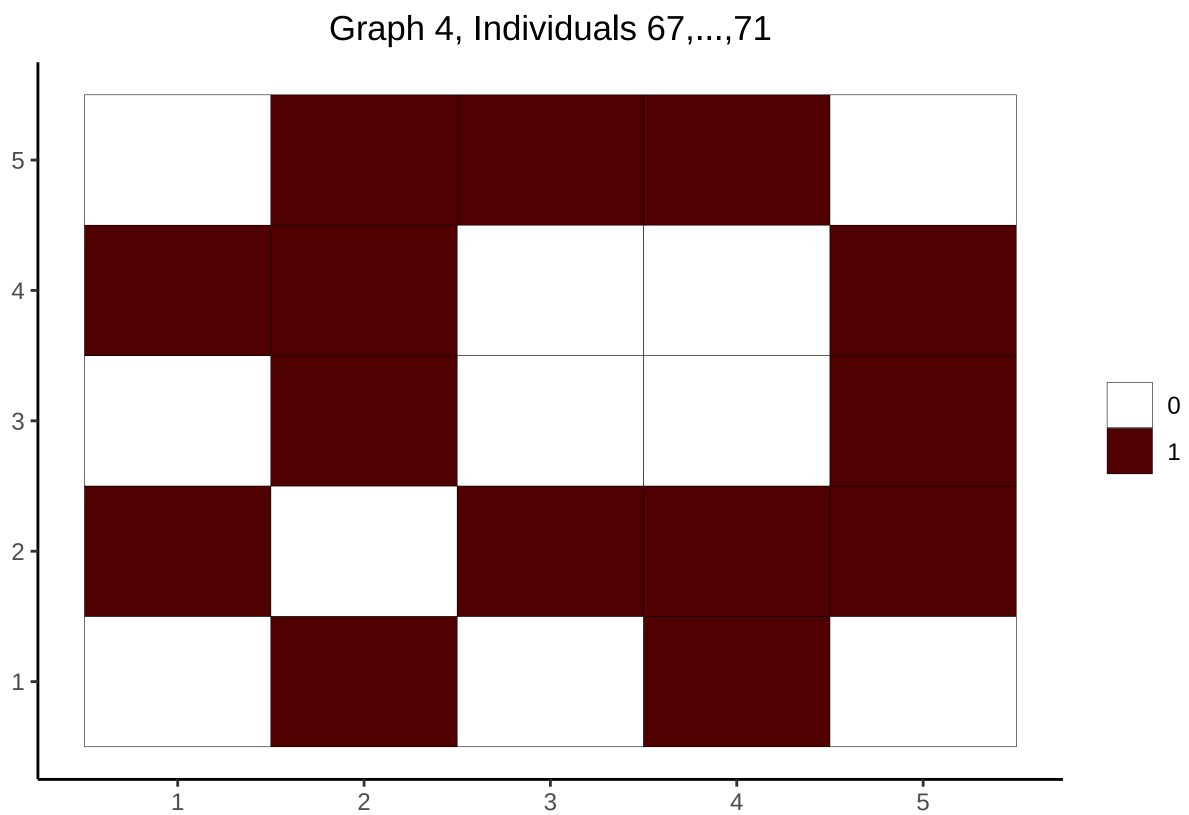

```
##
## [[2]]
```



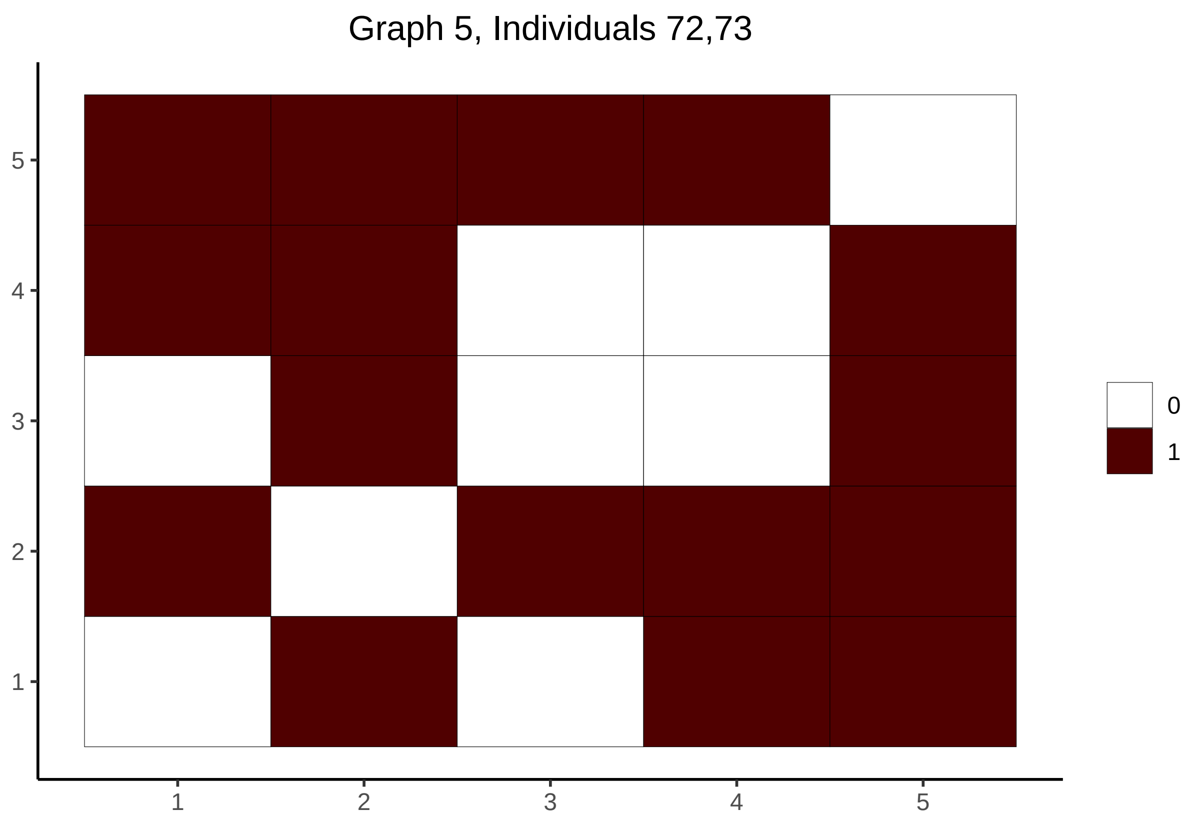

```
##
## [[3]]
```



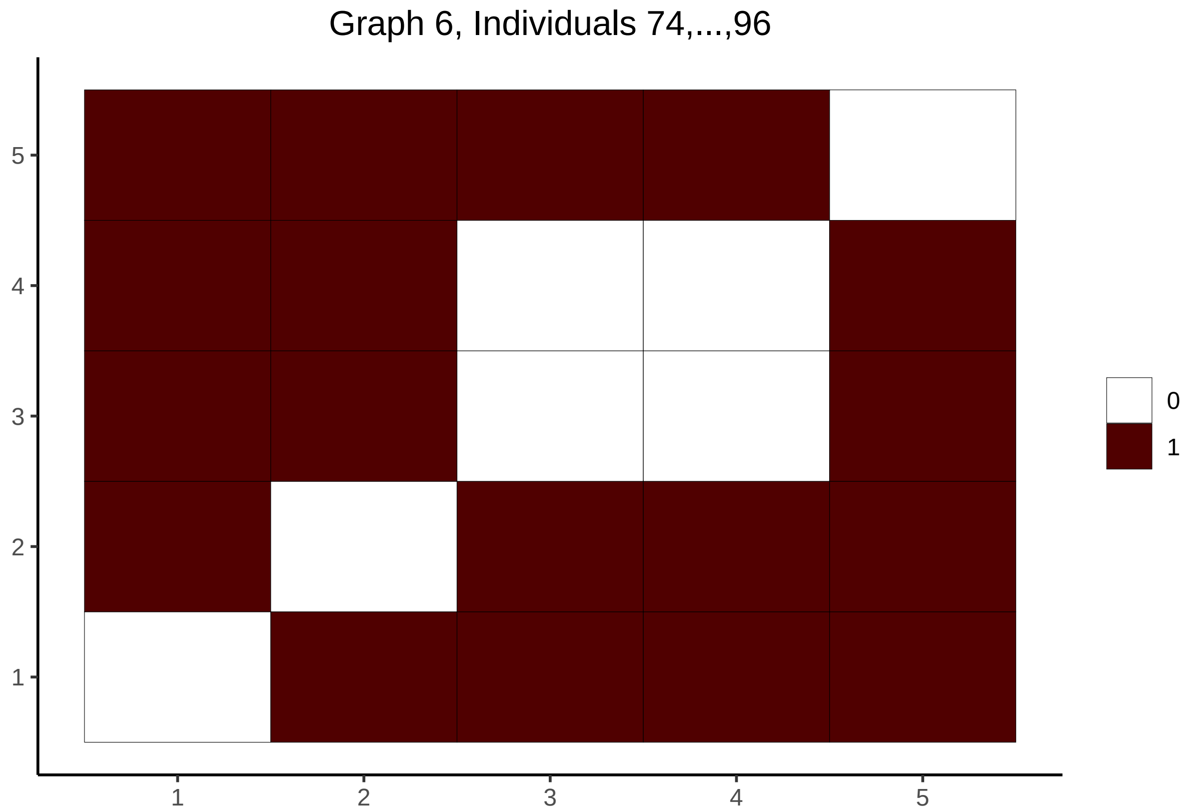

```
##
## [[4]]
```



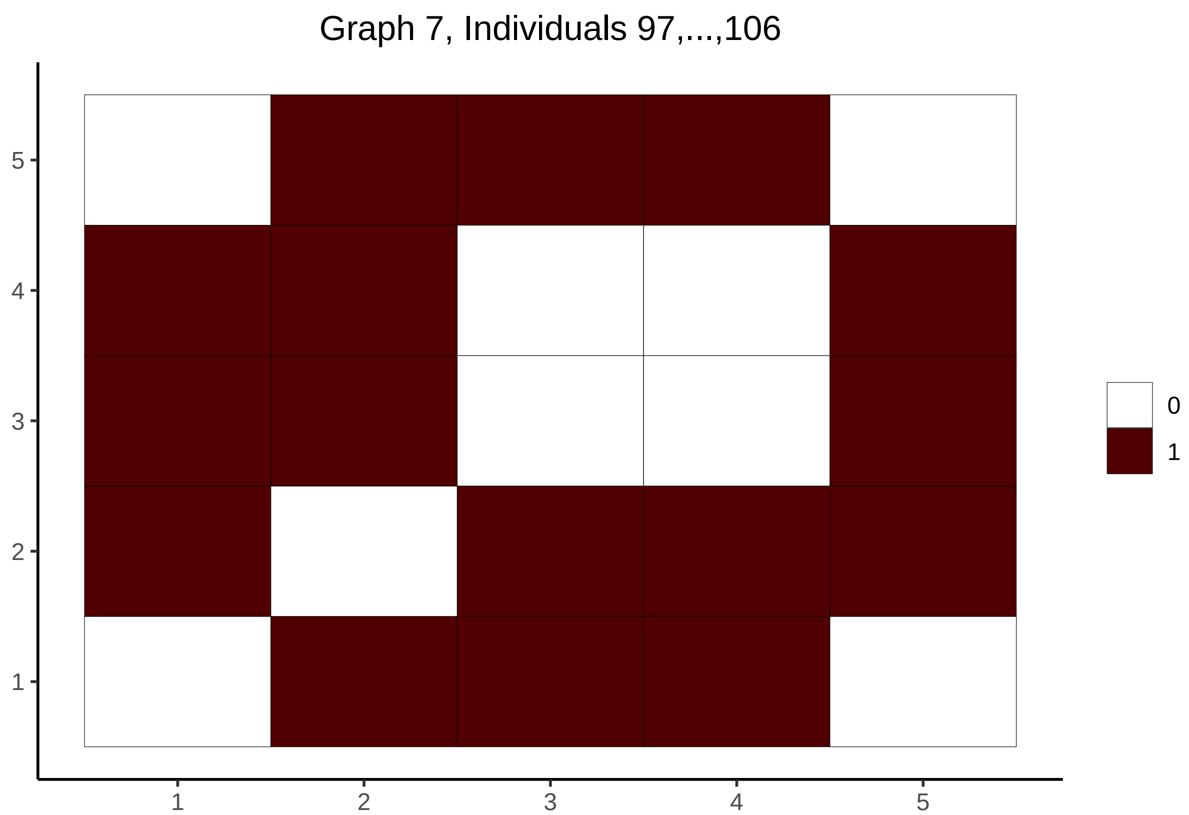

```
##
## [[5]]
```



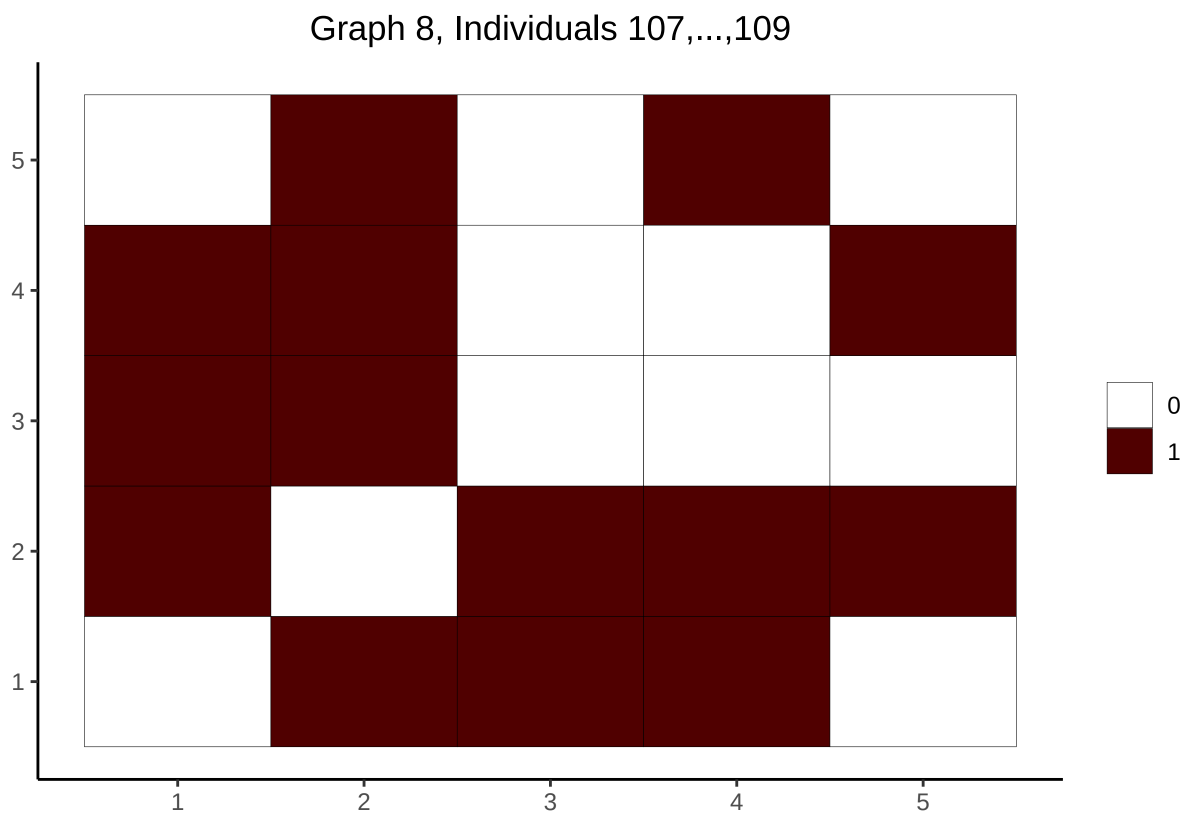

```
##
## [[6]]
```



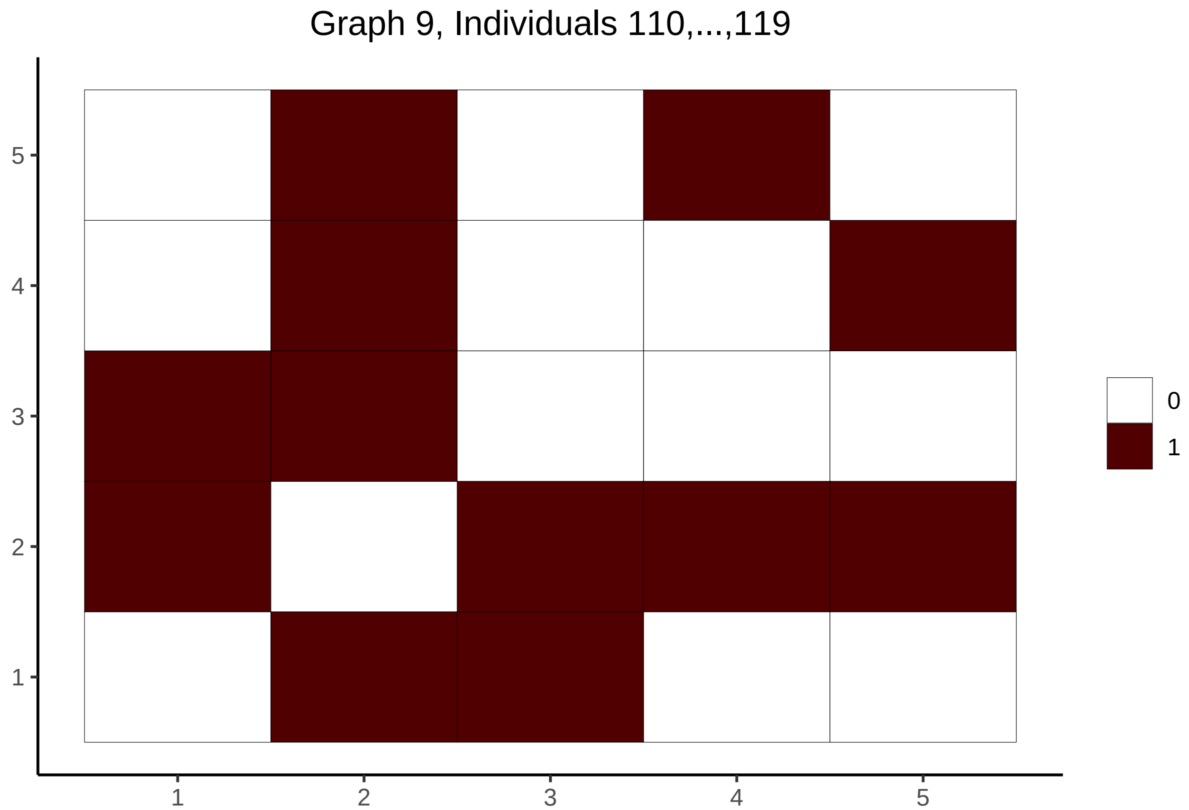

```
##
## [[7]]
```



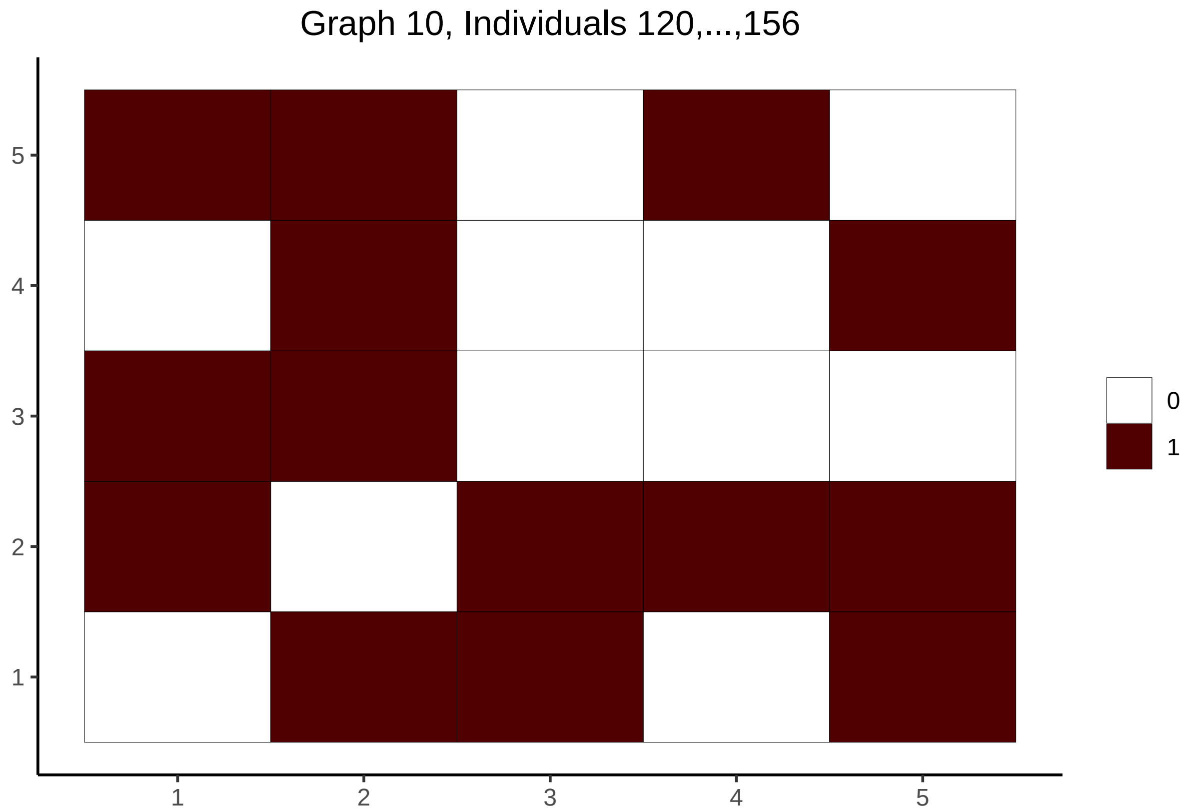

```
##
## [[8]]
```



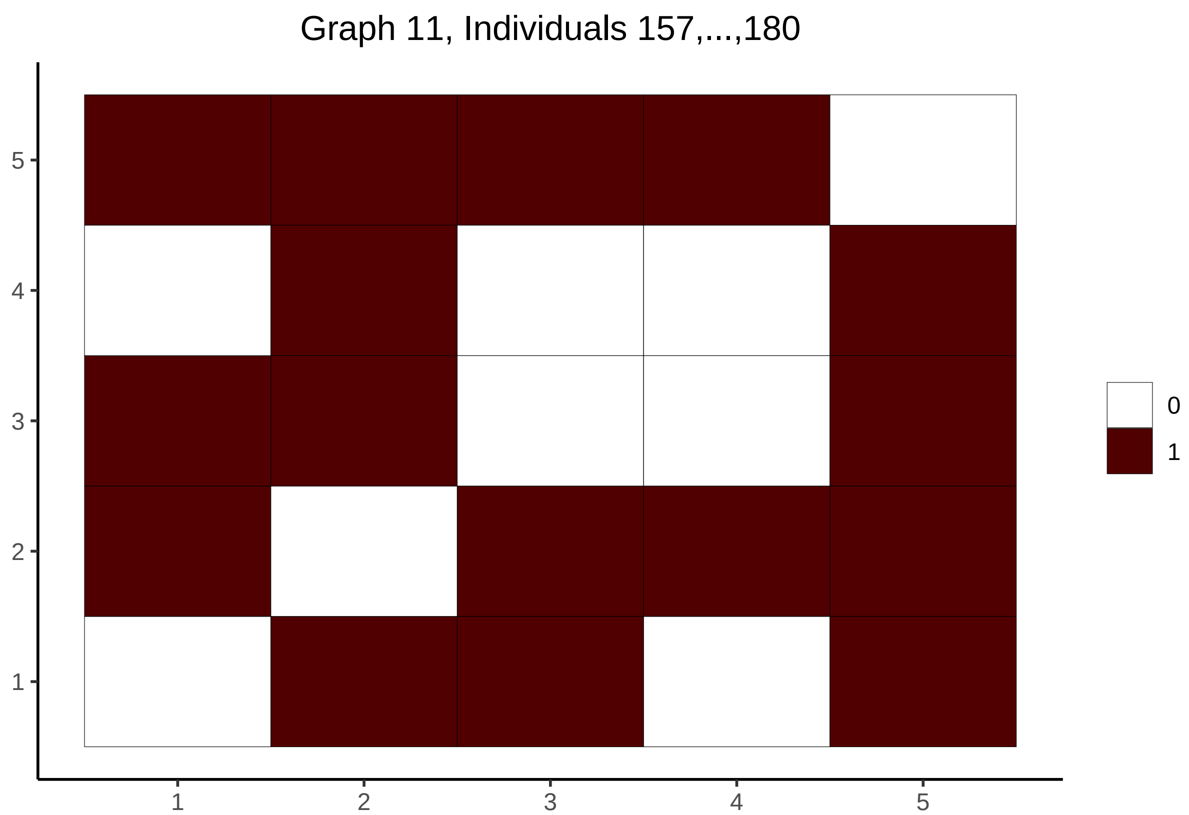

```
##
## [[9]]
```




```
##
## [[10]]
```




```
##
## [[11]]
```



In this experiment, I applied the grid search algorithm in 100 trials. I generated the grid from the cartesian product of the following marginal grids:

```r
# marginal grids
marg_grids
```

```
## $pip
##  [1] 0.00001 0.00010 0.00100 0.01000 0.02500 0.05000 0.15000 0.25000 0.35000
## [10] 0.45000 0.55000 0.65000 0.75000 0.85000 0.95000 0.99000 0.99900 0.99990
## [19] 0.99999
##
## $ssq
##  [1] 1.0e-05 1.0e-04 1.0e-03 1.0e-02 2.5e-02 5.0e-02 1.0e-01 1.5e-01 2.0e-01
## [10] 2.5e-01 3.0e-01 3.5e-01 4.0e-01 4.5e-01 5.0e-01 5.5e-01 6.0e-01 6.5e-01
## [19] 7.0e-01 7.5e-01 8.0e-01 8.5e-01 9.0e-01 9.5e-01 1.0e+00 1.5e+00 2.0e+00
## [28] 2.5e+00 3.0e+00 5.0e+00 1.0e+01
##
## $sbsq
##  [1] 1.0e-05 1.0e-04 1.0e-03 1.0e-02 2.5e-02 5.0e-02 1.0e-01 1.5e-01 2.0e-01
## [10] 2.5e-01 3.0e-01 3.5e-01 4.0e-01 4.5e-01 5.0e-01 5.5e-01 6.0e-01 6.5e-01
## [19] 7.0e-01 7.5e-01 8.0e-01 8.5e-01 9.0e-01 9.5e-01 1.0e+00 1.5e+00 2.0e+00
## [28] 2.5e+00 3.0e+00 5.0e+00 1.0e+01
```

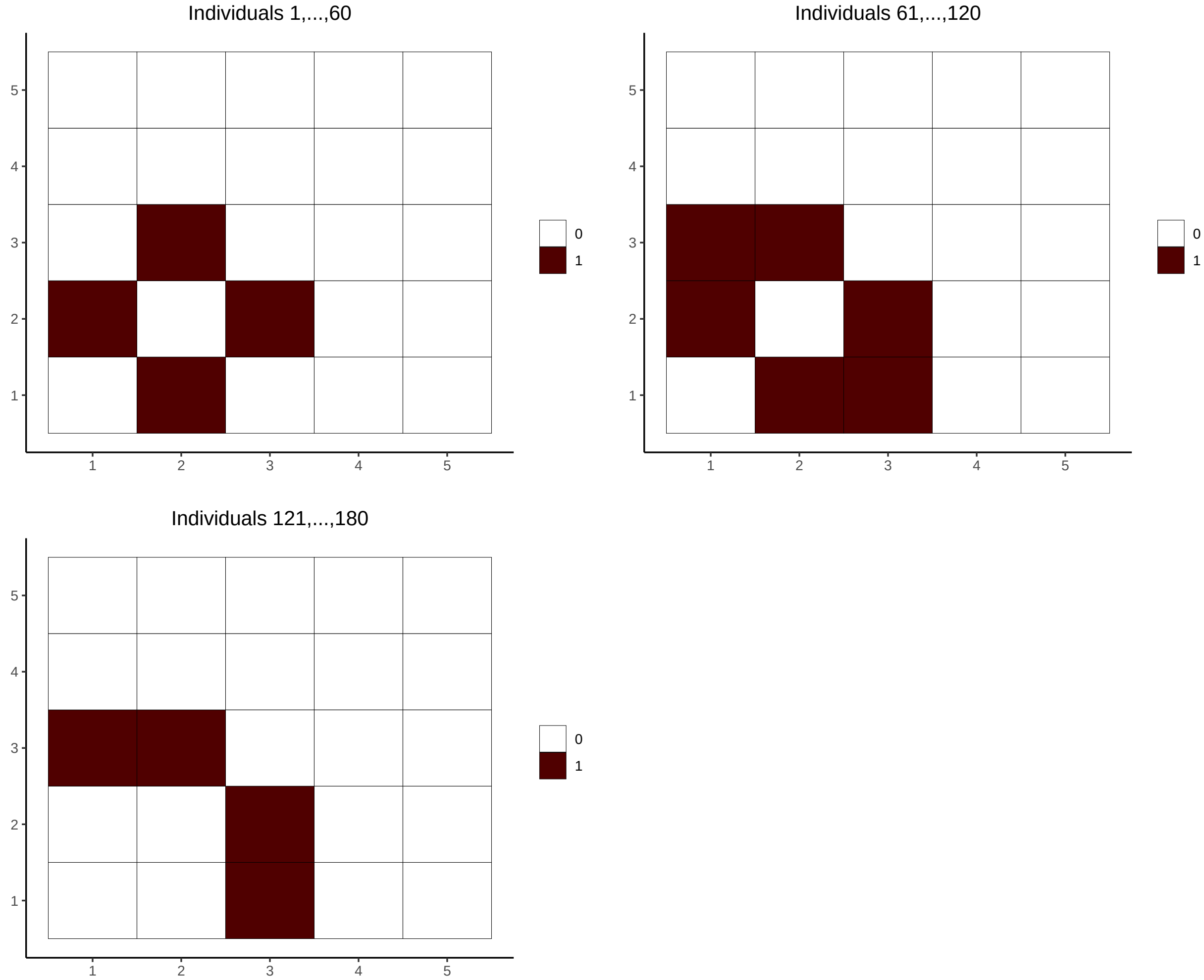
```r
# number of grid points in the cartesian product
nrow(expand.grid(marg_grids))
```

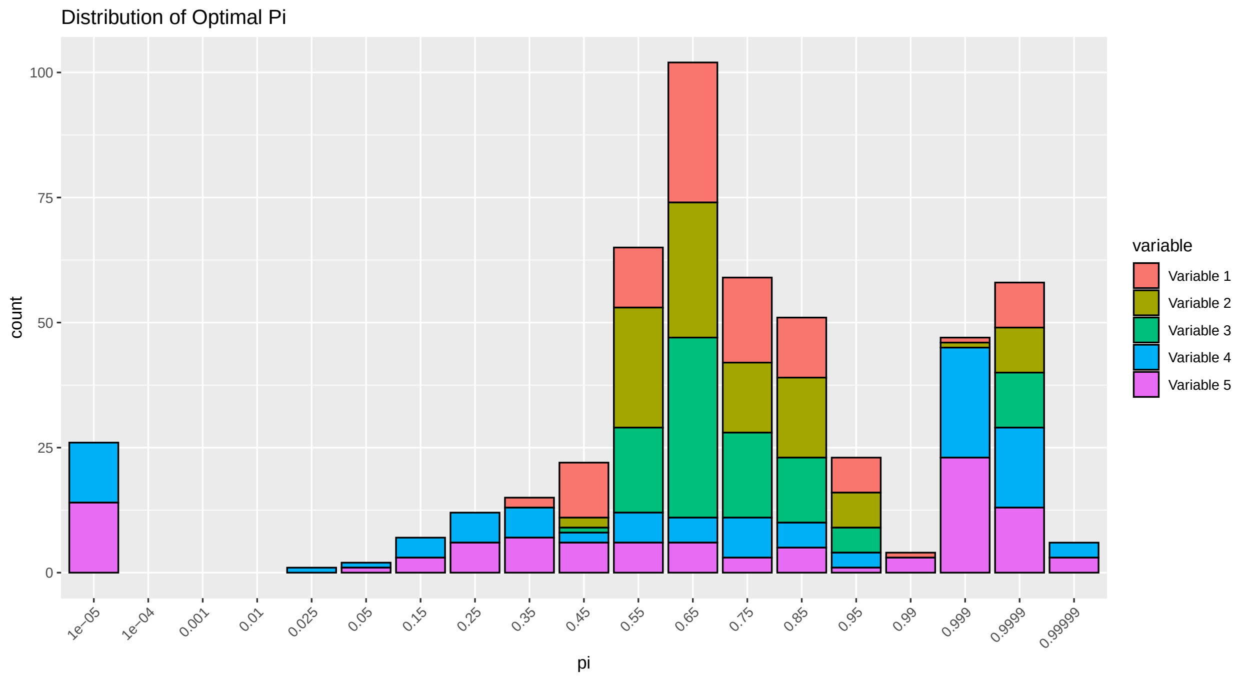
```
## [1] 18259
```

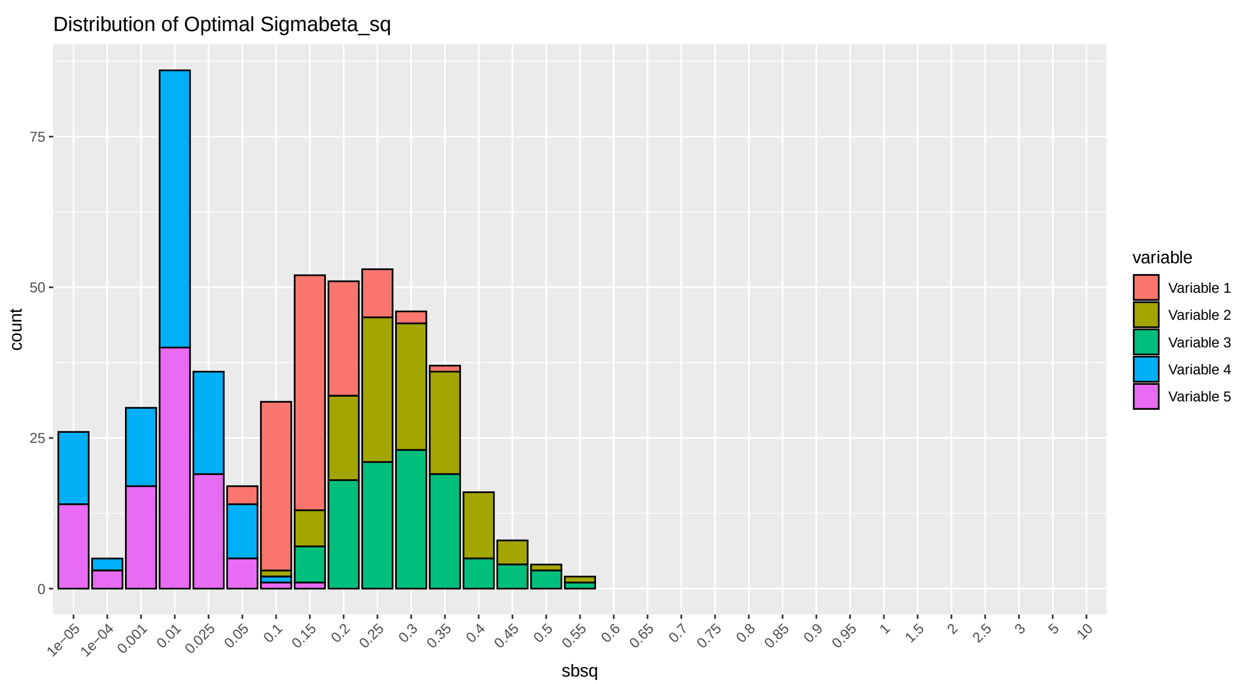
Optimal Hyperparameter Distribution

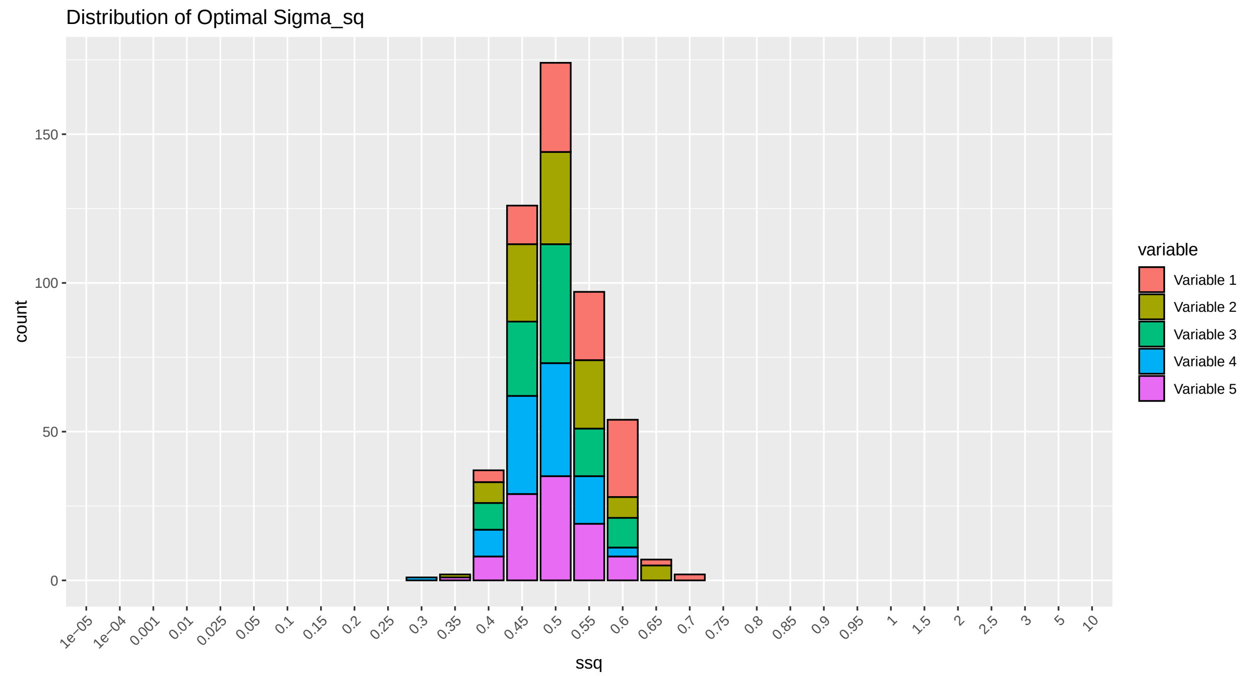
In this section, I display the distribution of optimal hyperparameters aggregated across all variables. Since a unique point in the hyperparameter grid was selected for each of the 5 variables, a total of 500 grid points are represented in each figure.

I begin by displaying the true underlying precision structures to provide context for the optimal pi.




Distribution of Optimal Pi
count
pi
variable
Variable 1
Variable 2
Variable 3
Variable 4
Variable 5
0
25
50
75
100
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999


Distribution of Optimal Sigmabeta_sq
count
sbsq
variable
Variable 1
Variable 2
Variable 3
Variable 4
Variable 5
0
25
50
75
1e−05
1e−04
0.001
0.01
0.025
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
1.5
2
2.5
3
5
10

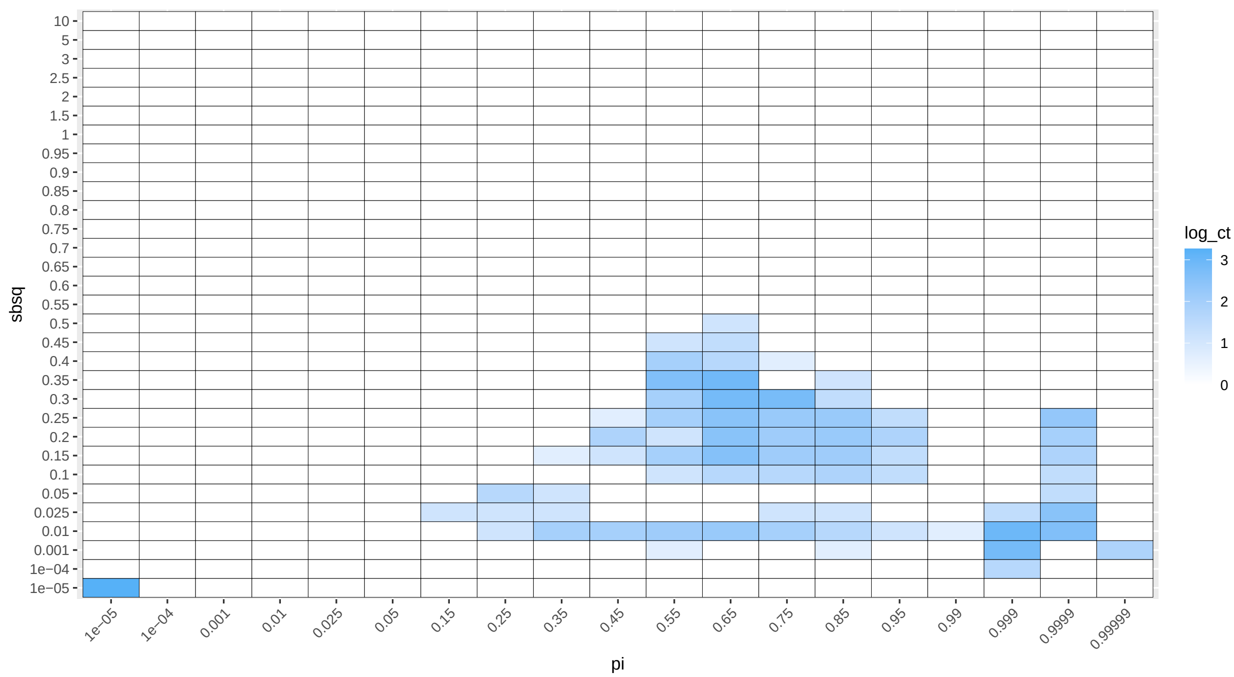


Optimal Sigmabeta_sq as a function of Pi

Here, I visualize the optimal sigmabeta_sq as a function of pi. Note that the count is on the log scale.

```
##
##           1e-05 1e-04 0.001 0.01 0.025 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45
##   1e-05      26     0     0    0     0    0   0    0   0    0   0    0   0    0
##   1e-04       0     0     0    0     0    0   0    0   0    0   0    0   0    0
##   0.001       0     0     0    0     0    0   0    0   0    0   0    0   0    0
##   0.01        0     0     0    0     0    0   0    0   0    0   0    0   0    0
##   0.025       0     0     0    1     0    0   0    0   0    0   0    0   0    0
##   0.05        0     0     0    0     0    1   0    1   0    0   0    0   0    0
##   0.15        0     0     1    1     3    1   1    0   0    0   0    0   0    0
##   0.25        0     0     0    3     3    5   1    0   0    0   0    0   0    0
##   0.35        0     0     0    7     3    3   0    2   0    0   0    0   0    0
##   0.45        0     0     0    7     1    0   0    3   6    2   1    0   1    1
##   0.55        0     0     2    8     1    1   3    7   3    7   7   14   7    3
##   0.65        0     0     1    9     1    1   5   13  12   12  17   18   5    4
##   0.75        0     0     1    7     3    0   5    8   8    9  16    0   2    0
##   0.85        0     0     2    5     3    1   6    8   9    9   4    3   1    0
##   0.95        0     0     0    3     1    0   4    4   6    4   0    1   0    0
##   0.99        0     0     0    2     1    0   1    0   0    0   0    0   0    0
##   0.999       0     5    17   19     4    0   1    0   0    0   0    1   0    0
##   0.9999      0     0     0   14    12    4   4    6   7   10   1    0   0    0
##   0.99999     0     0     6    0     0    0   0    0   0    0   0    0   0    0
##
##           0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95   1 1.5   2 2.5   3   5  10
##   1e-05     0    0   0    0   0    0   0    0   0    0   0   0   0   0   0   0   0
##   1e-04     0    0   0    0   0    0   0    0   0    0   0   0   0   0   0   0   0
##   0.001     0    0   0    0   0    0   0    0   0    0   0   0   0   0   0   0   0
##   0.01      0    0   0    0   0    0   0    0   0    0   0   0   0   0   0   0   0
```

```
##   0.025    0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.05     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.15     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.25     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.35     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.45     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.55     1    1   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.65     3    1   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.75     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.85     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.95     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.99     0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.999    0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.9999   0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
##   0.99999  0    0   0    0   0    0   0    0   0    0  0    0  0    0  0  0  0
```

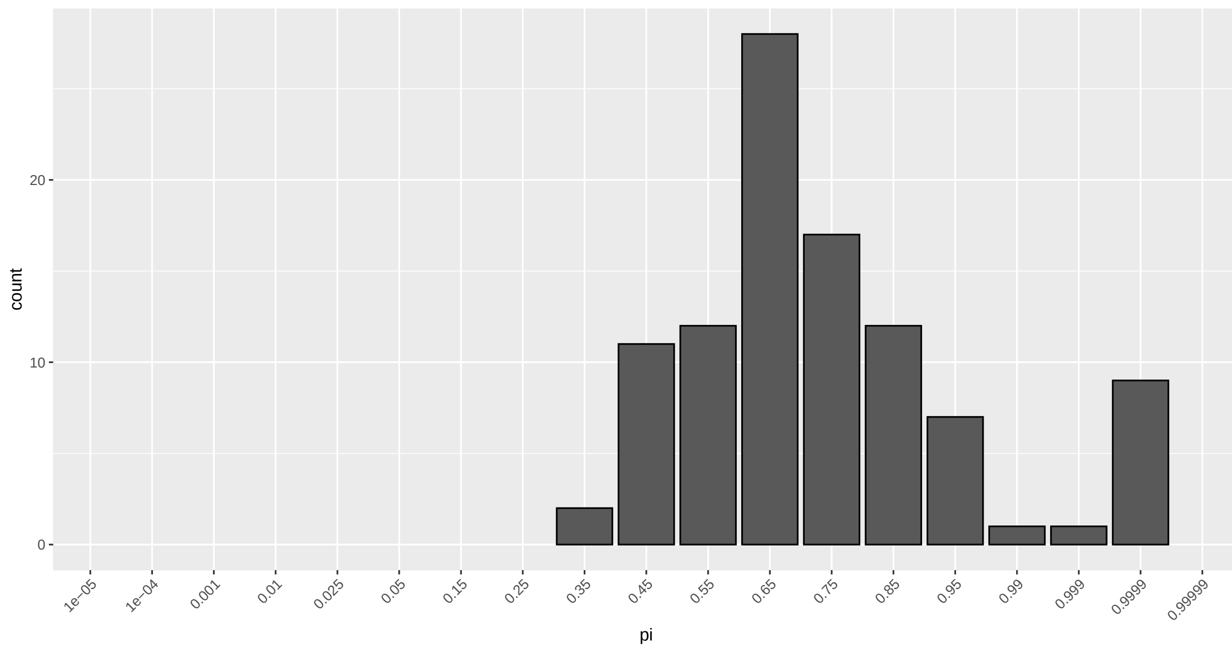


Optimal Pi by variable

In this section, I visualize the optimal pi by variable.

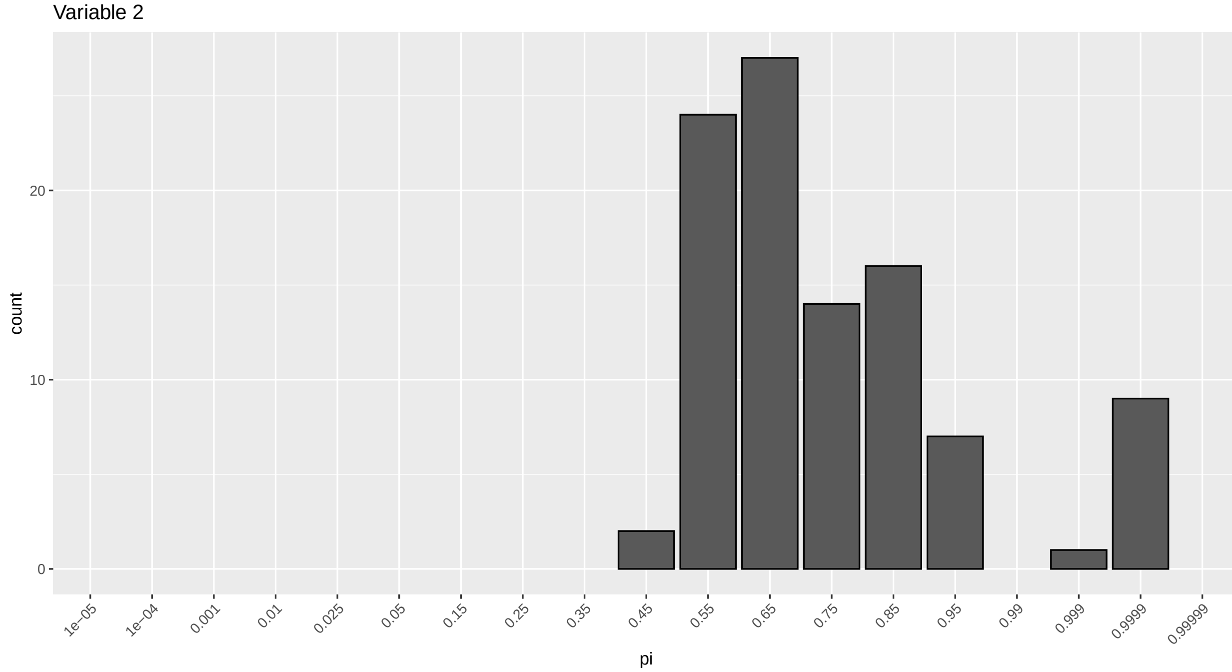
```
## [[1]]
```

Variable 1

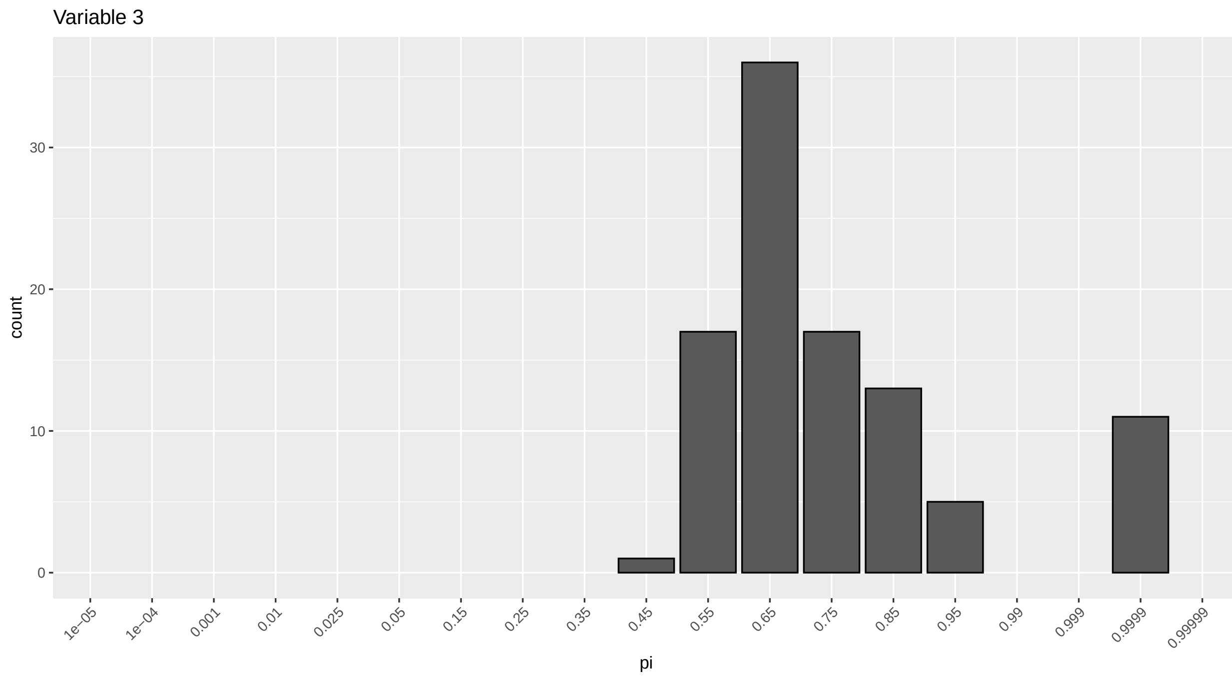



```
##
## [[2]]
```

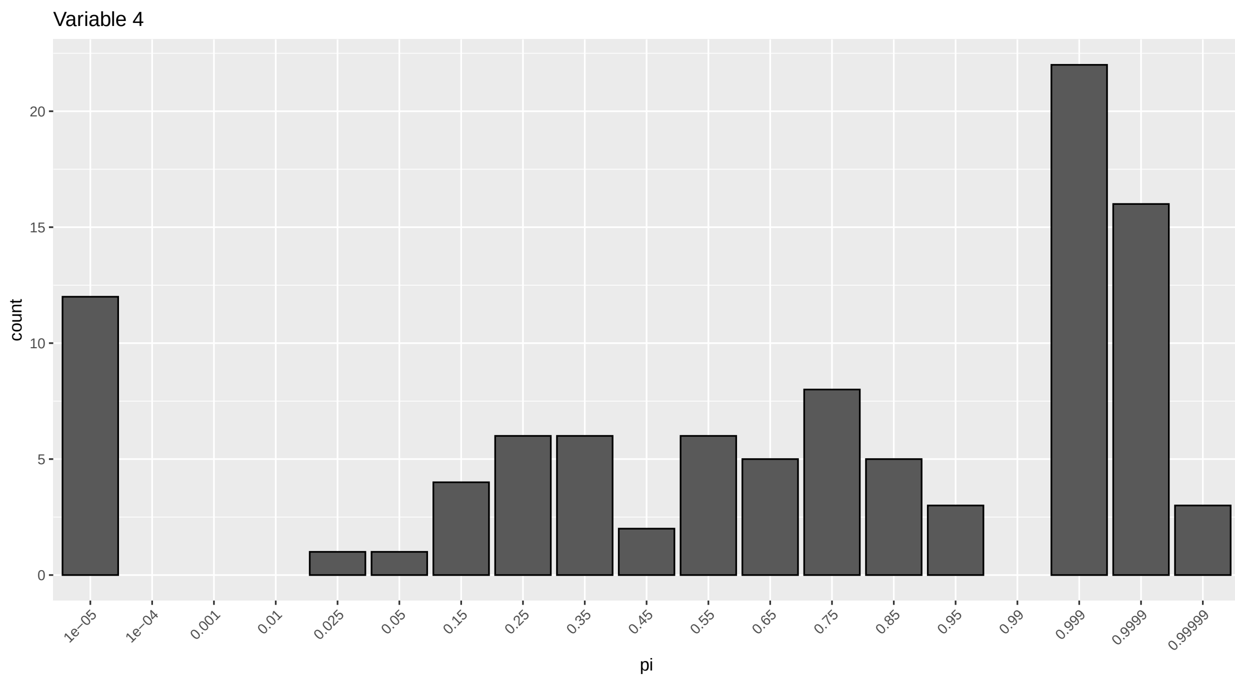
Variable 2



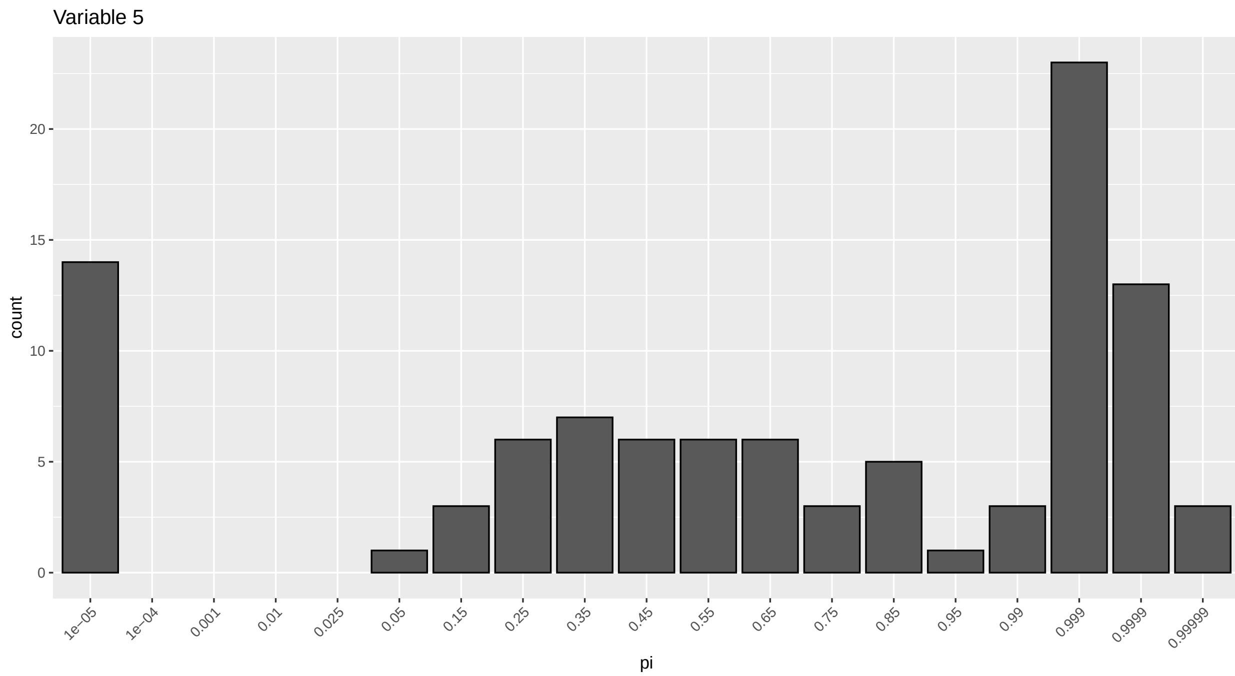

```
##
## [[3]]
```


Variable 3
count
pi
0
10
20
30
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

```
##
## [[4]]
```


Variable 4
count
pi
0
5
10
15
20
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

```
##
## [[5]]
```



$p = 20$

In this section, I present the same analyses as above for a dataset with $p = 20$.

The marginal grids are:

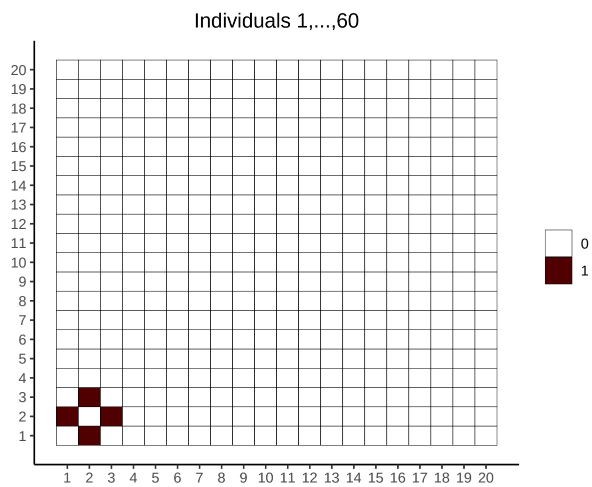
```
# marginal grids
marg_grids
```

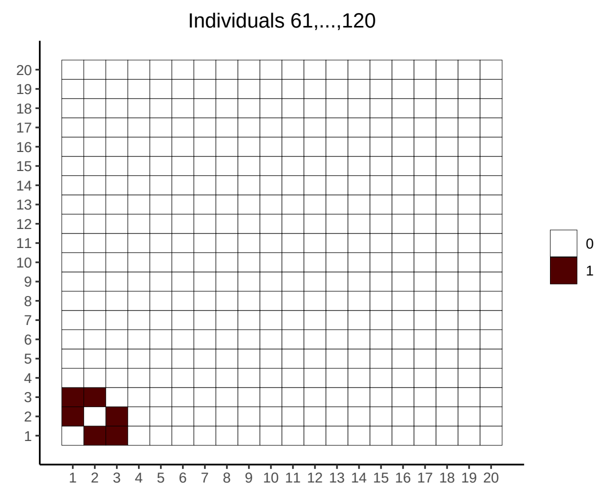
```
##         pip     ssq    sbsq
## 1  0.00001 1.0e-05 1.0e-05
## 2  0.00010 1.0e-04 1.0e-04
## 3  0.00100 1.0e-03 1.0e-03
## 4  0.01000 1.0e-02 1.0e-02
## 5  0.02500 2.5e-02 2.5e-02
## 6  0.05000 5.0e-02 5.0e-02
## 7  0.15000 2.0e-01 2.0e-01
## 8  0.25000 3.5e-01 3.5e-01
## 9  0.35000 5.0e-01 5.0e-01
## 10 0.45000 6.5e-01 6.5e-01
## 11 0.55000 8.0e-01 8.0e-01
## 12 0.65000 9.5e-01 9.5e-01
## 13 0.75000 1.0e+00 1.0e+00
## 14 0.85000 1.5e+00 1.5e+00
## 15 0.95000 2.0e+00 2.0e+00
## 16 0.99000 2.5e+00 2.5e+00
## 17 0.99900 3.0e+00 3.0e+00
## 18 0.99990 5.0e+00 5.0e+00
## 19 0.99999 1.0e+01 1.0e+01
```

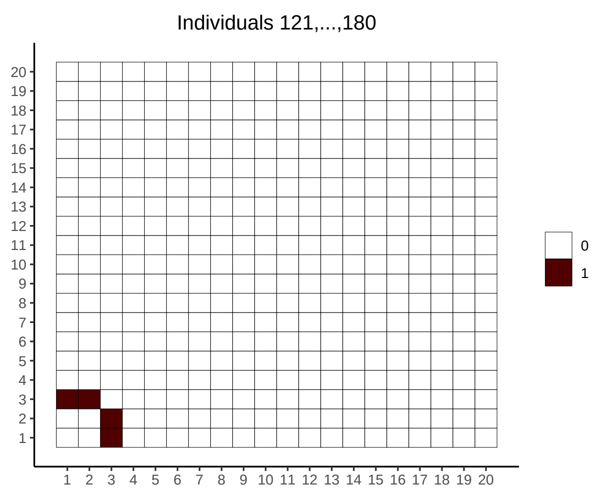
```r
# number of grid points in the cartesian product
nrow(expand.grid(marg_grids))
```

```
## [1] 6859
```

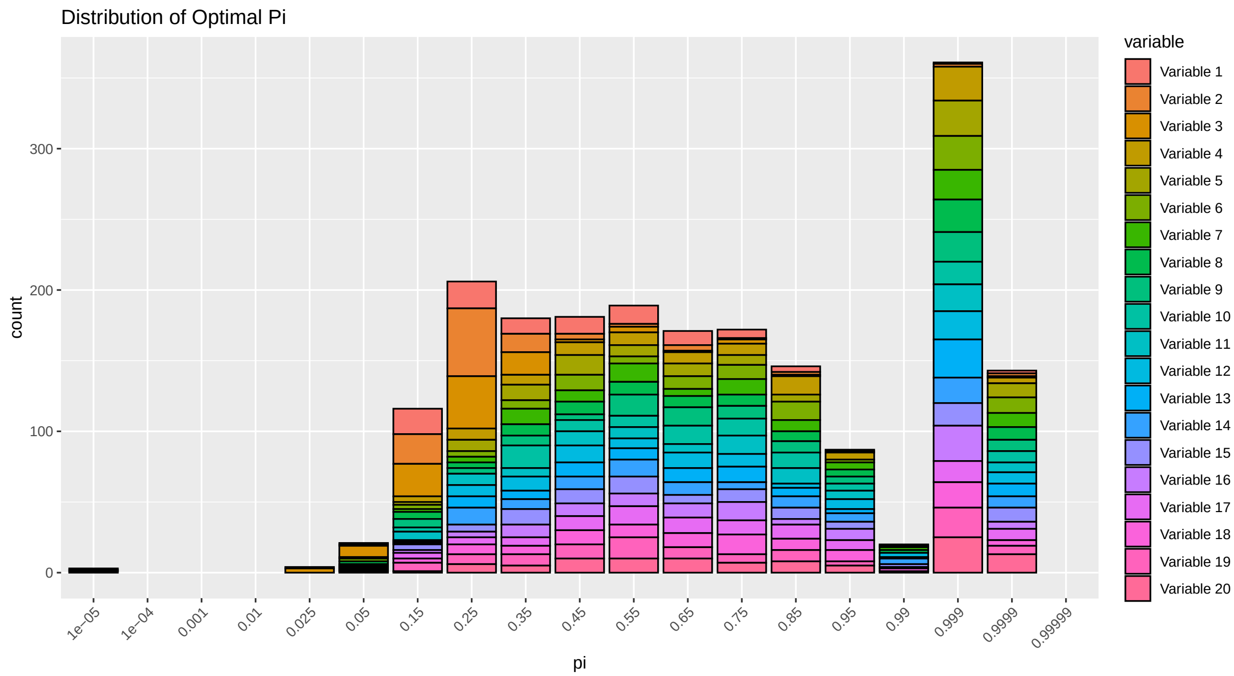
The true precision structures are given below:

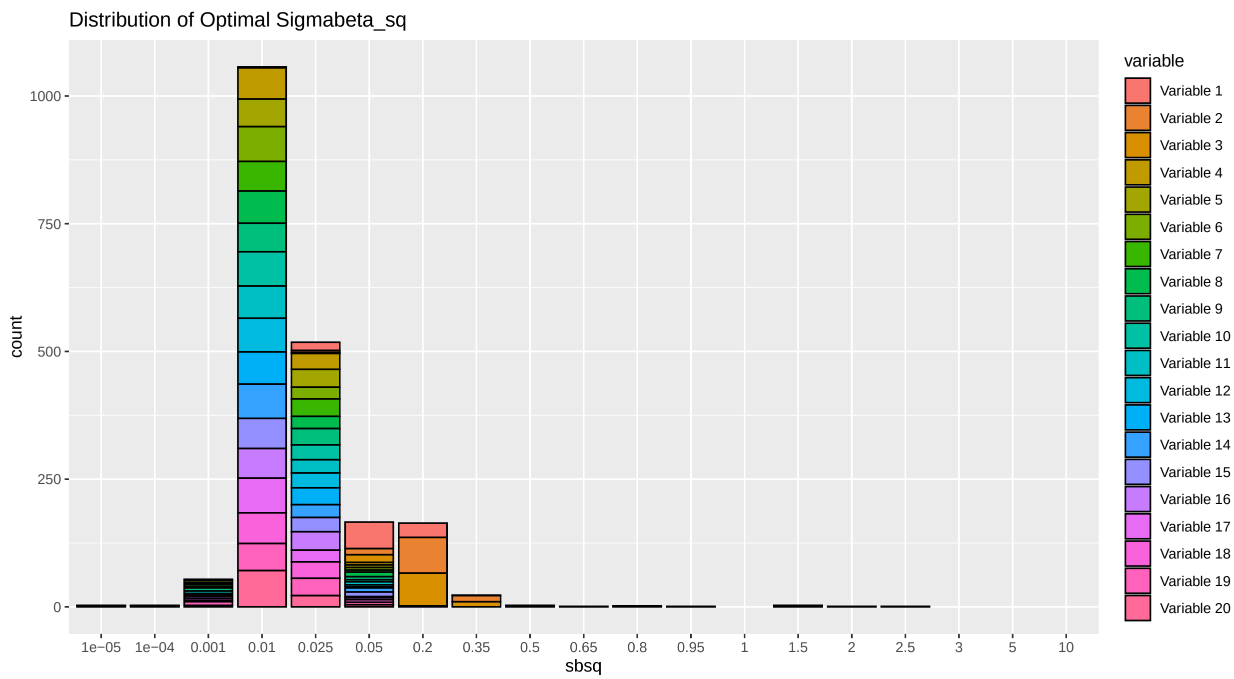


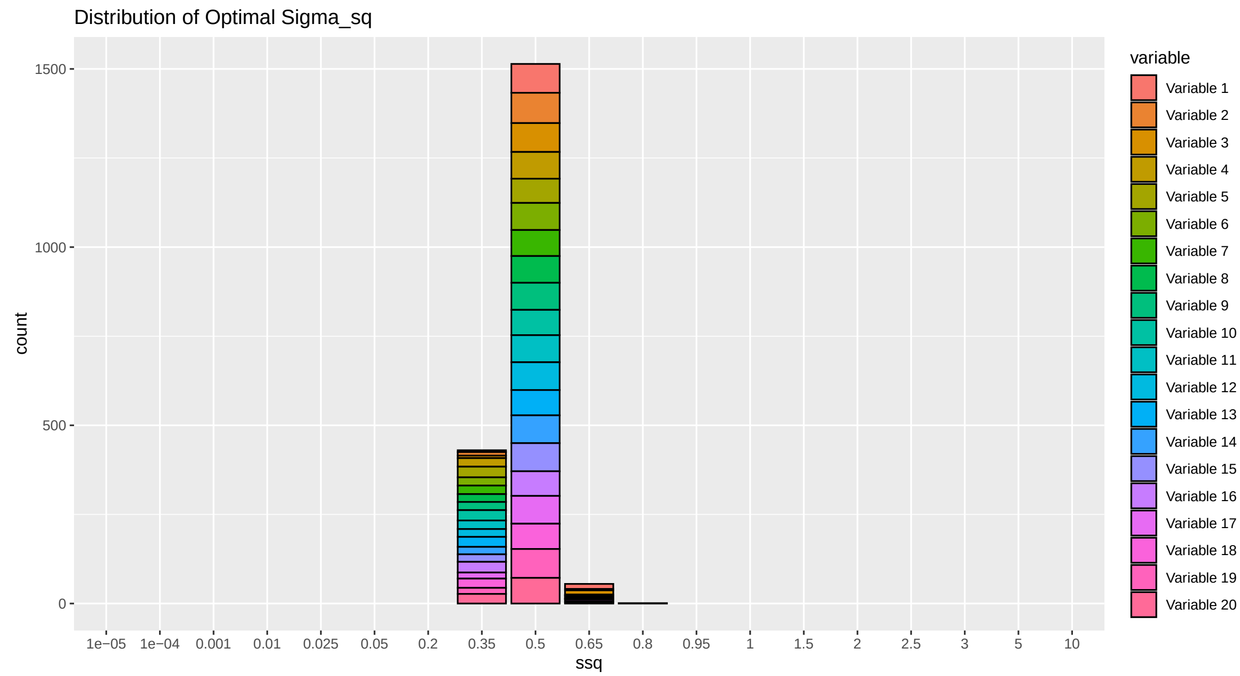




Below is the distribution of the optimal hyperparameters aggregated across all variables.


Distribution of Optimal Pi
count
300
200
100
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi
variable
Variable 1
Variable 2
Variable 3
Variable 4
Variable 5
Variable 6
Variable 7
Variable 8
Variable 9
Variable 10
Variable 11
Variable 12
Variable 13
Variable 14
Variable 15
Variable 16
Variable 17
Variable 18
Variable 19
Variable 20

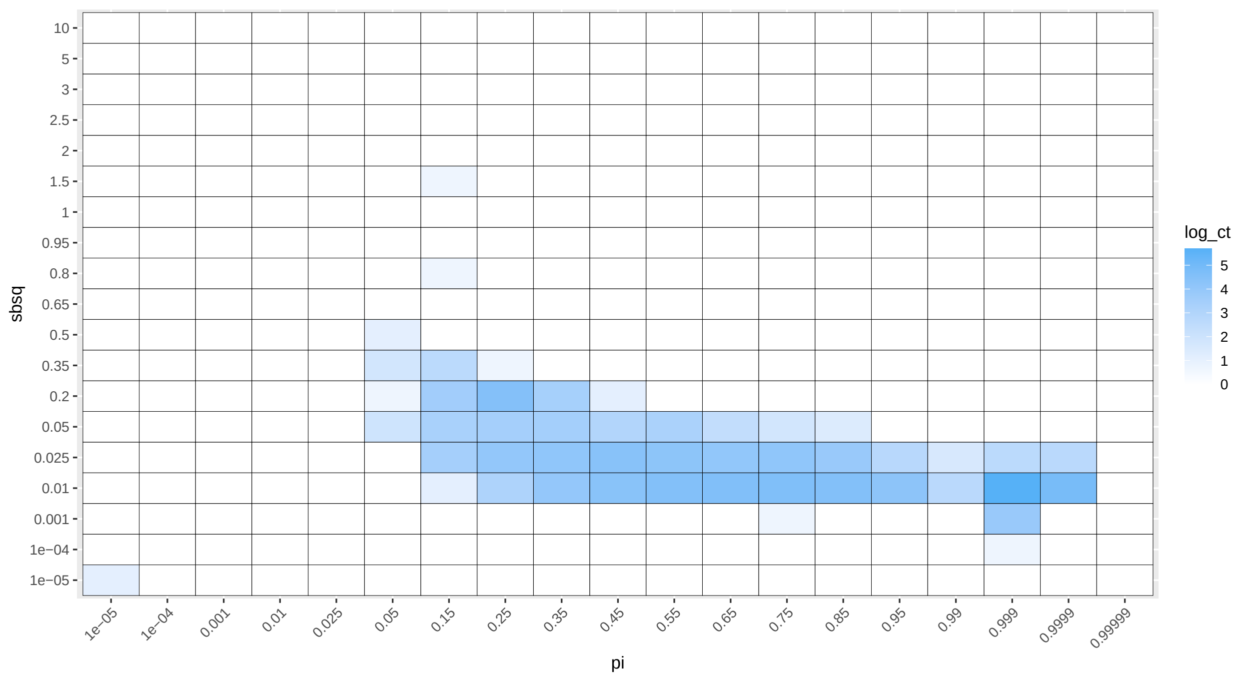

Distribution of Optimal Sigmabeta_sq
count
1000
750
500
250
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.2
0.35
0.5
0.65
0.8
0.95
1
1.5
2
2.5
3
5
10
sbsq
variable
Variable 1
Variable 2
Variable 3
Variable 4
Variable 5
Variable 6
Variable 7
Variable 8
Variable 9
Variable 10
Variable 11
Variable 12
Variable 13
Variable 14
Variable 15
Variable 16
Variable 17
Variable 18
Variable 19
Variable 20



The optimal value of sbsq as a function of pi is visualized below.

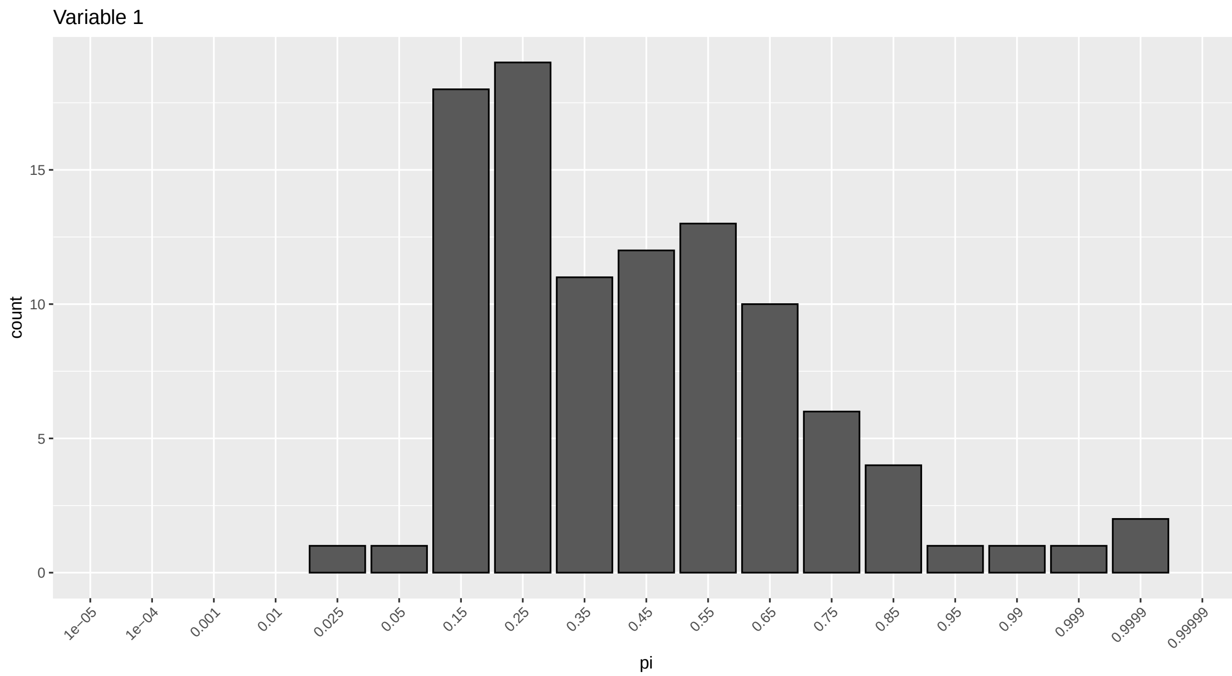
```
##
##           1e-05 1e-04 0.001 0.01 0.025 0.05 0.2 0.35 0.5 0.65 0.8 0.95   1 1.5
##   1e-05      3     0     0    0     0    0   0    0   0    0   0    0   0   0
##   1e-04      0     0     0    0     0    0   0    0   0    0   0    0   0   0
##   0.001      0     0     0    0     0    0   0    0   0    0   0    0   0   0
##   0.01       0     0     0    0     0    0   0    0   0    0   0    0   0   0
##   0.025      0     0     0    0     0    0   1    1   0    0   0    0   0   1
##   0.05       0     0     0    1     1    7   2    6   3    1   0    0   0   0
##   0.15       0     0     0    3    31   27  35   14   0    0   2    1   0   2
##   0.25       0     0     1   23    57   31  92    2   0    0   0    0   0   0
##   0.35       0     0     1   55    62   32  30    0   0    0   0    0   0   0
##   0.45       0     0     0   77    81   20   3    0   0    0   0    0   0   0
##   0.55       0     1     1   94    68   25   0    0   0    0   0    0   0   0
##   0.65       0     0     1  100    58   11   1    0   0    0   0    0   0   0
##   0.75       0     0     2  101    63    6   0    0   0    0   0    0   0   0
##   0.85       0     0     0   95    47    4   0    0   0    0   0    0   0   0
##   0.95       0     0     1   69    16    1   0    0   0    0   0    0   0   0
##   0.99       0     0     0   15     5    0   0    0   0    0   0    0   0   0
##   0.999      0     2    47  297    14    1   0    0   0    0   0    0   0   0
##   0.9999     0     0     0  127    15    1   0    0   0    0   0    0   0   0
##   0.99999    0     0     0    0     0    0   0    0   0    0   0    0   0   0
##
##              2 2.5   3   5  10
##   1e-05      0   0   0   0   0
##   1e-04      0   0   0   0   0
##   0.001      0   0   0   0   0
##   0.01       0   0   0   0   0
##   0.025      0   1   0   0   0
##   0.05       0   0   0   0   0
##   0.15       1   0   0   0   0
```

```
##    0.25       0   0   0   0   0
##    0.35       0   0   0   0   0
##    0.45       0   0   0   0   0
##    0.55       0   0   0   0   0
##    0.65       0   0   0   0   0
##    0.75       0   0   0   0   0
##    0.85       0   0   0   0   0
##    0.95       0   0   0   0   0
##    0.99       0   0   0   0   0
##    0.999      0   0   0   0   0
##    0.9999     0   0   0   0   0
##    0.99999    0   0   0   0   0
```

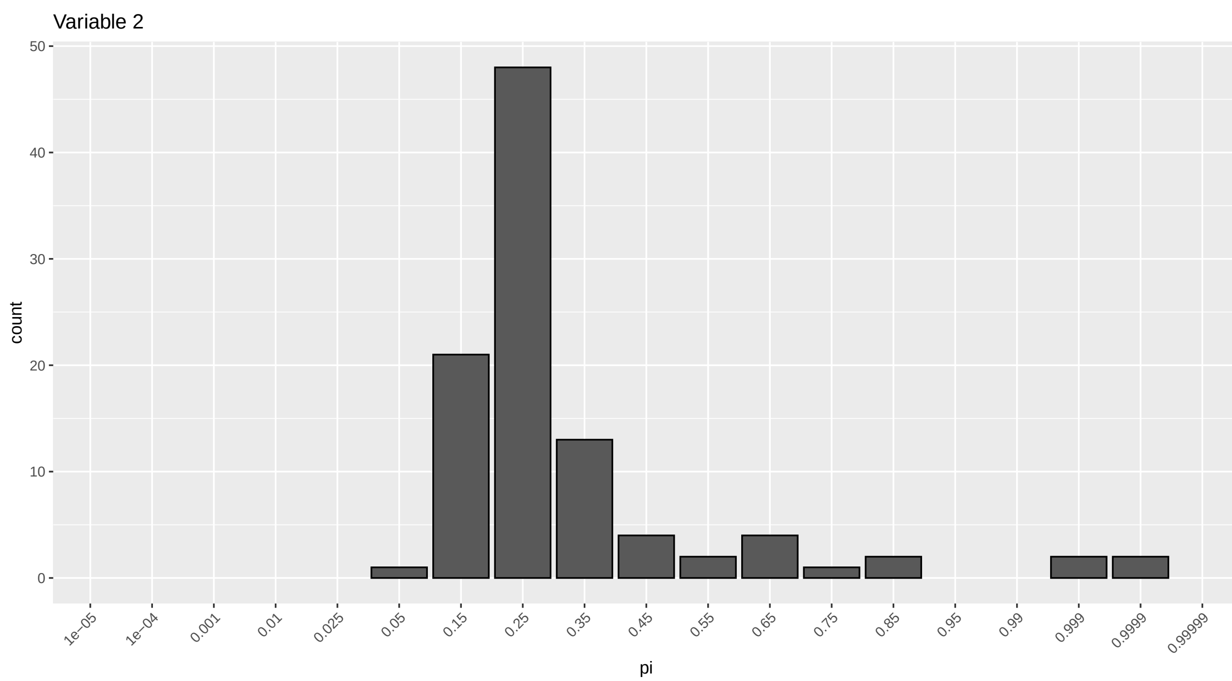


Here the distribution of the optimal pi is visualized by variable.

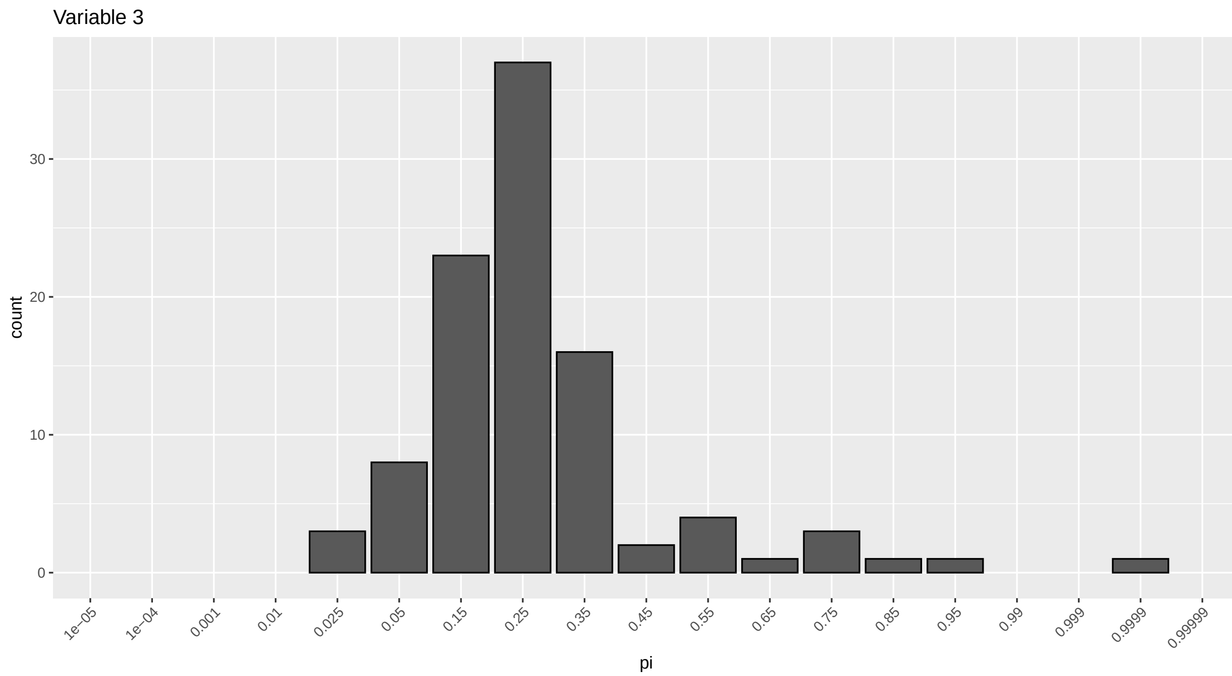
```
## [[1]]
```


Variable 1
count
pi
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

```
##
## [[2]]
```

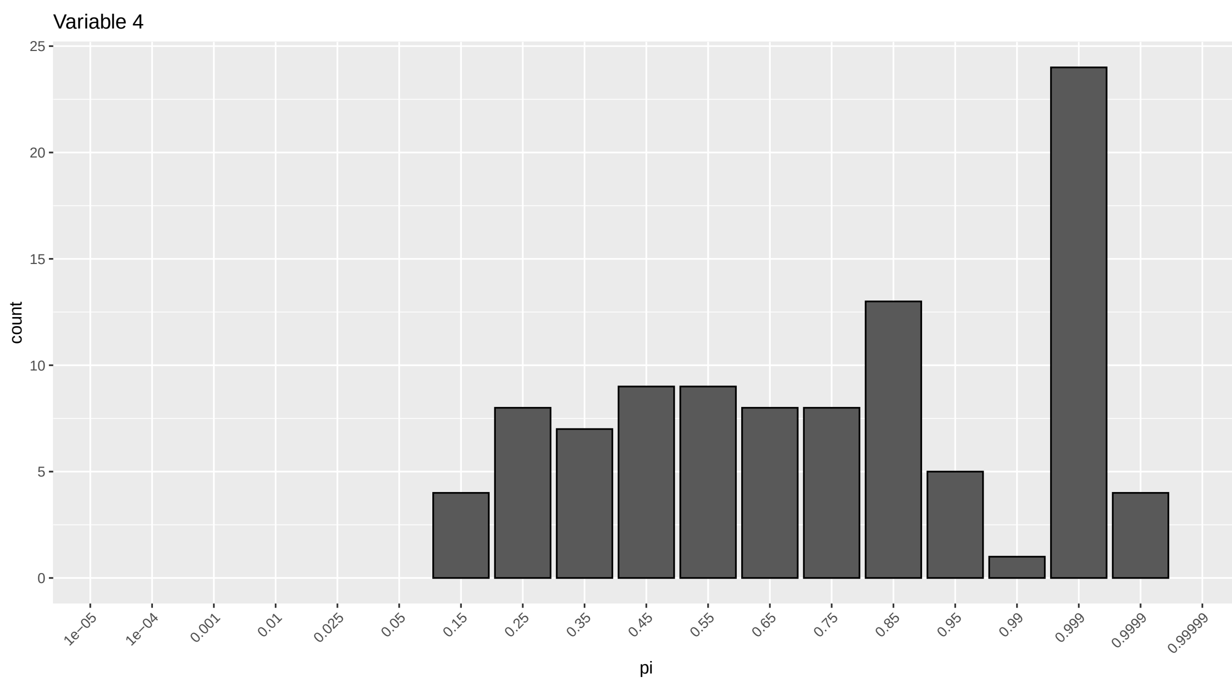

Variable 2
count
pi
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

```
##
## [[3]]
```



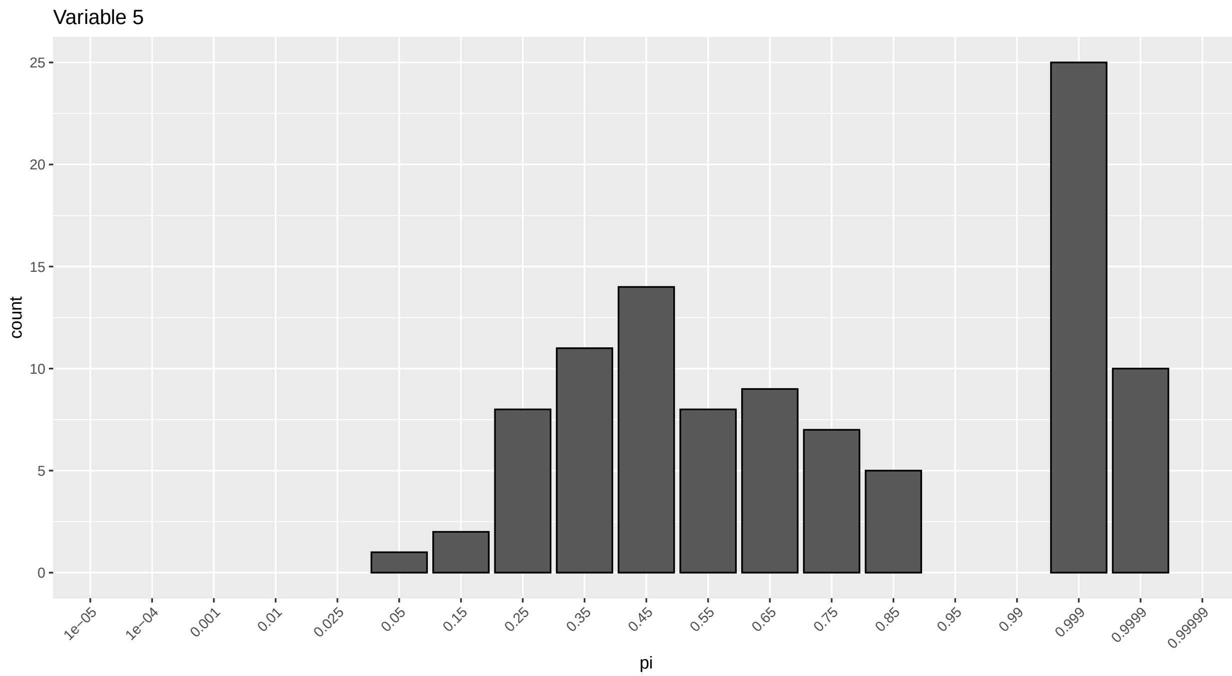
Variable 3
count
0
10
20
30
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi


```
##
## [[4]]
```



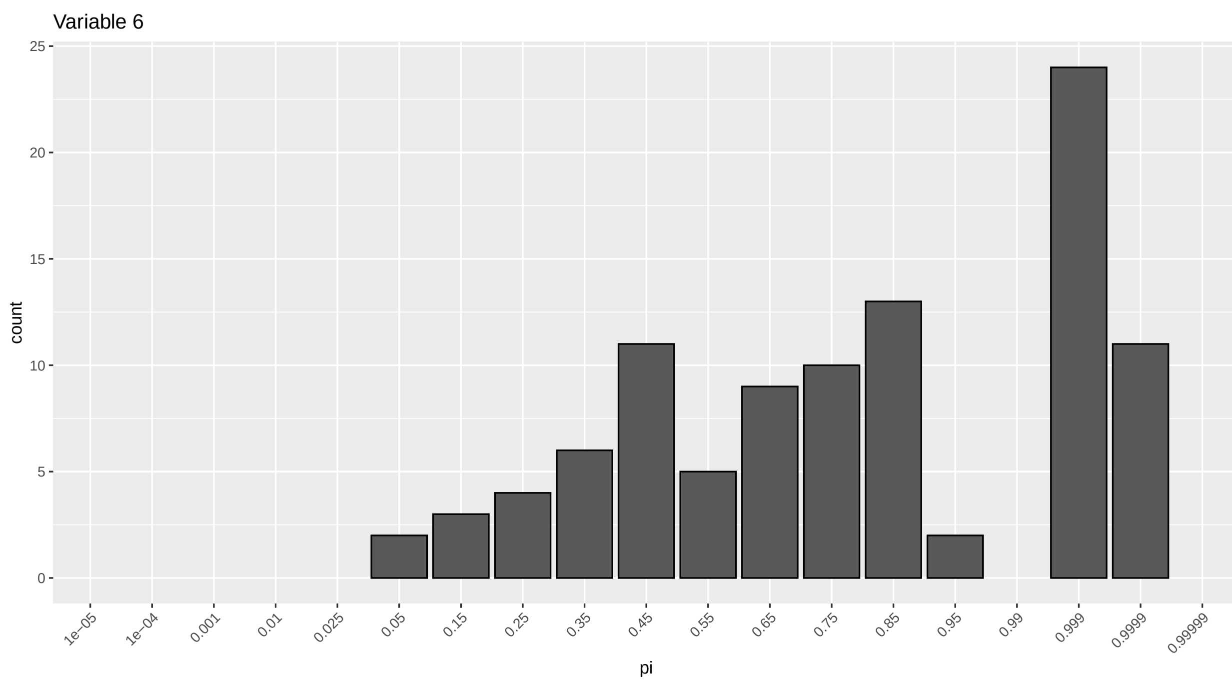
Variable 4
count
0
5
10
15
20
25
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi


```
##
## [[5]]
```



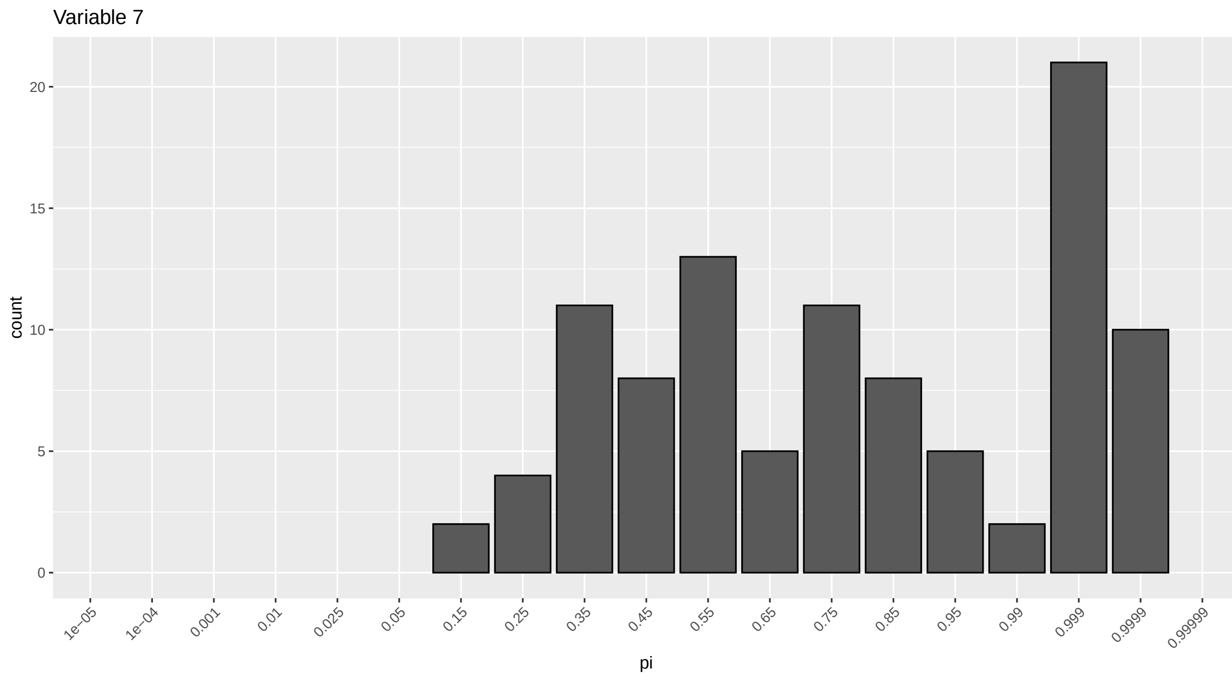
Variable 5
count
25
20
15
10
5
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi


```
##
## [[6]]
```



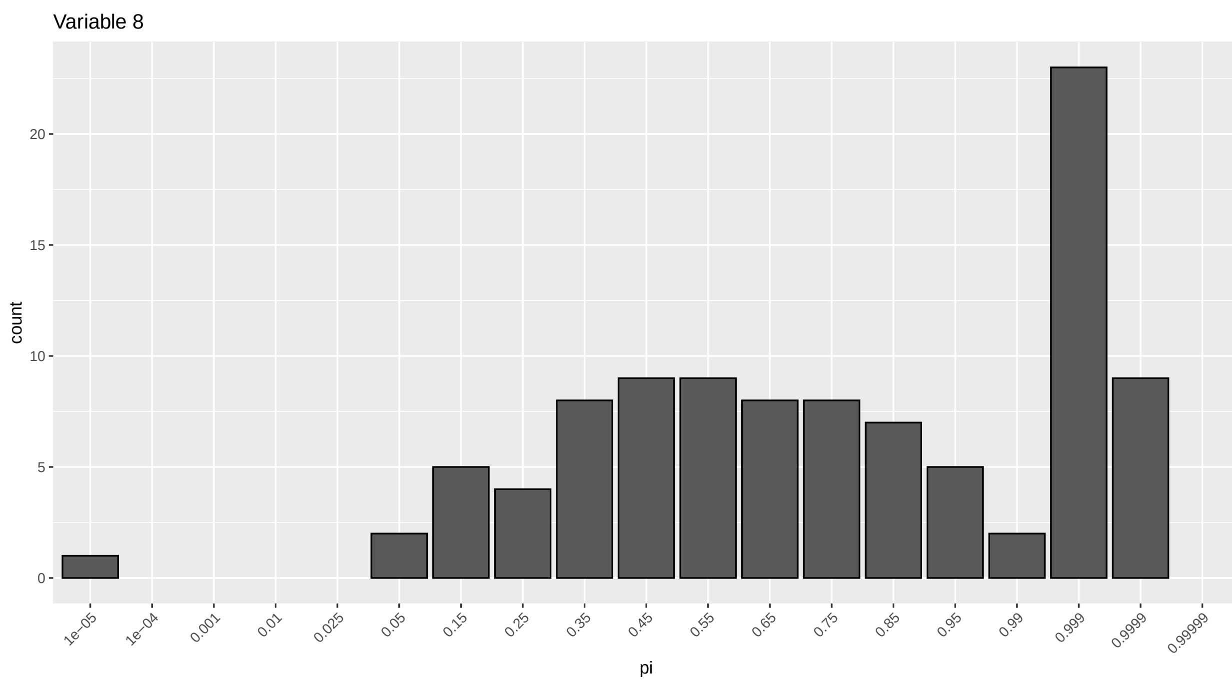
Variable 6
count
25
20
15
10
5
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi


```
##
## [[7]]
```



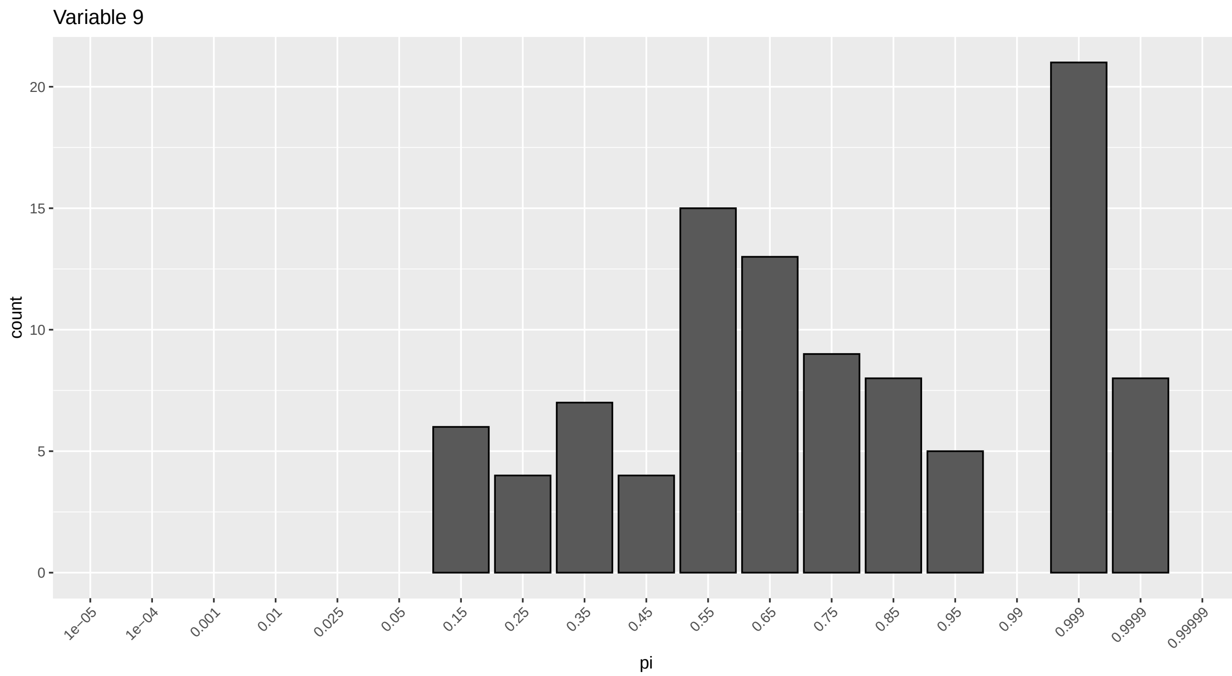
Variable 7
count
20
15
10
5
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi


```
##
## [[8]]
```

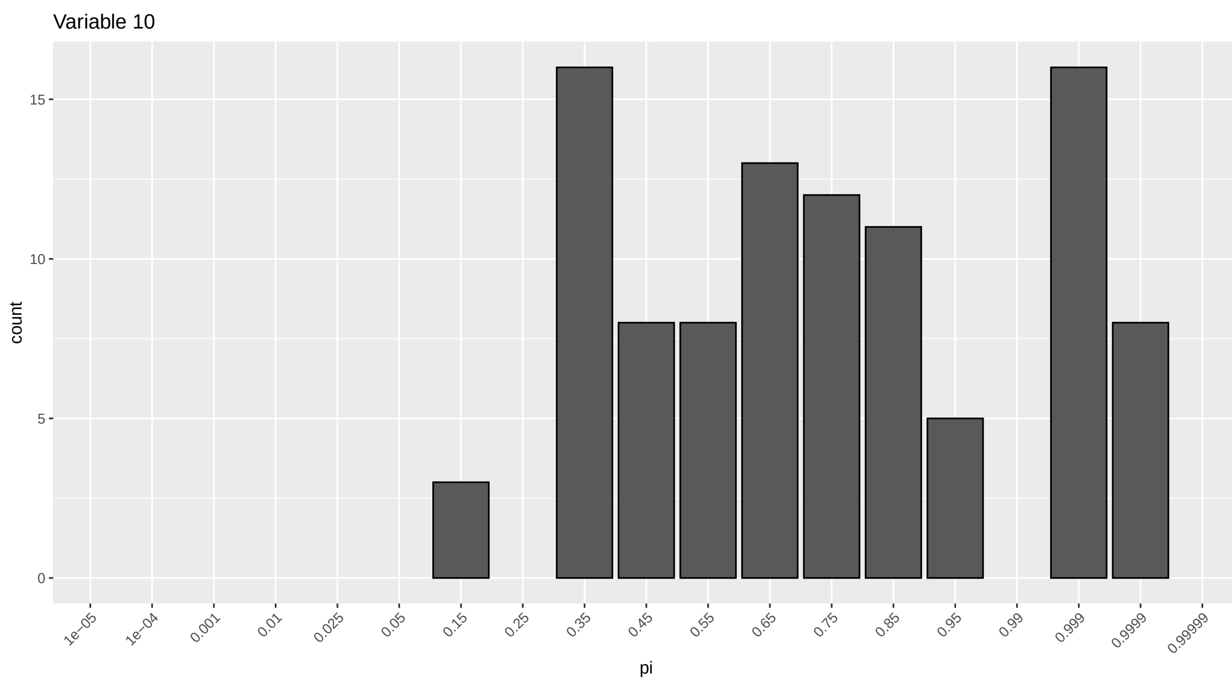


Variable 8
count
20
15
10
5
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

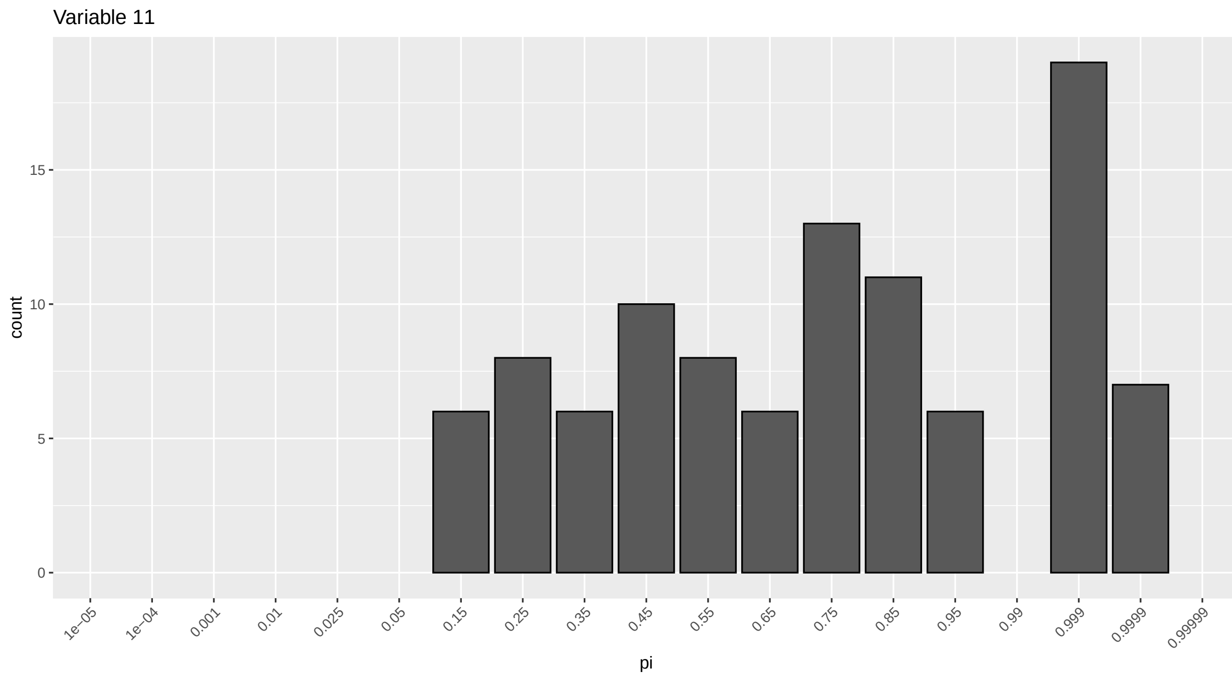

```
##
## [[9]]
```


Variable 9
20
15
10
5
0
count
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

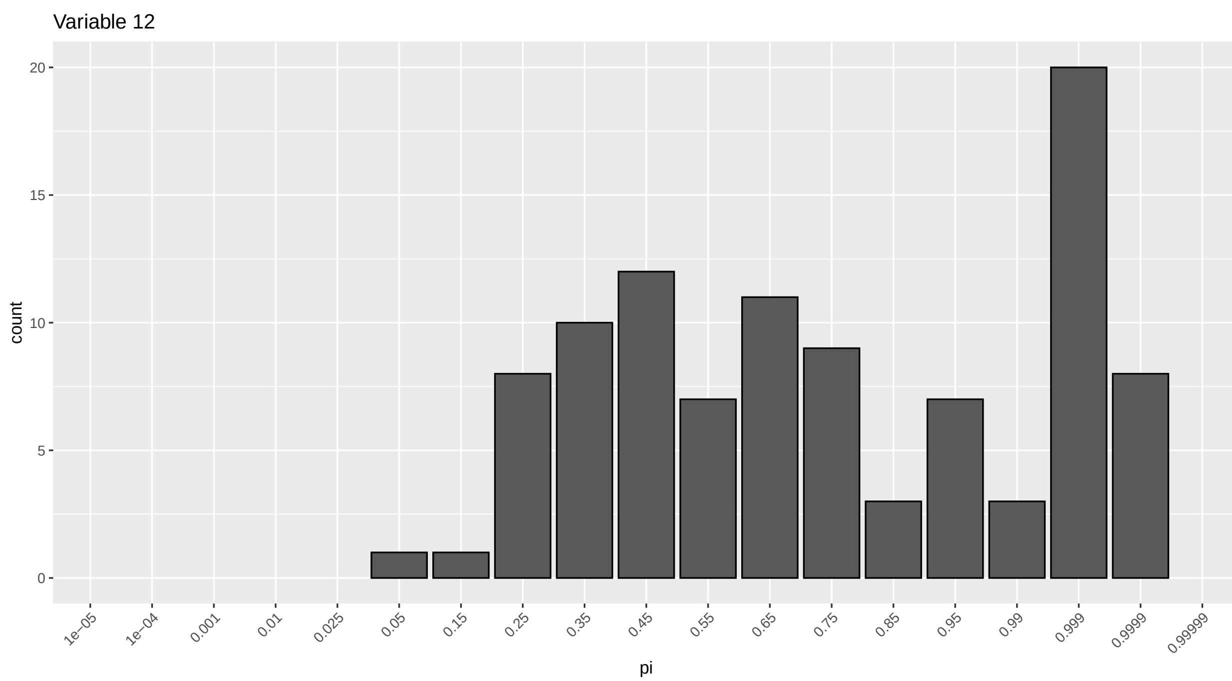
```
##
## [[10]]
```


Variable 10
15
10
5
0
count
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

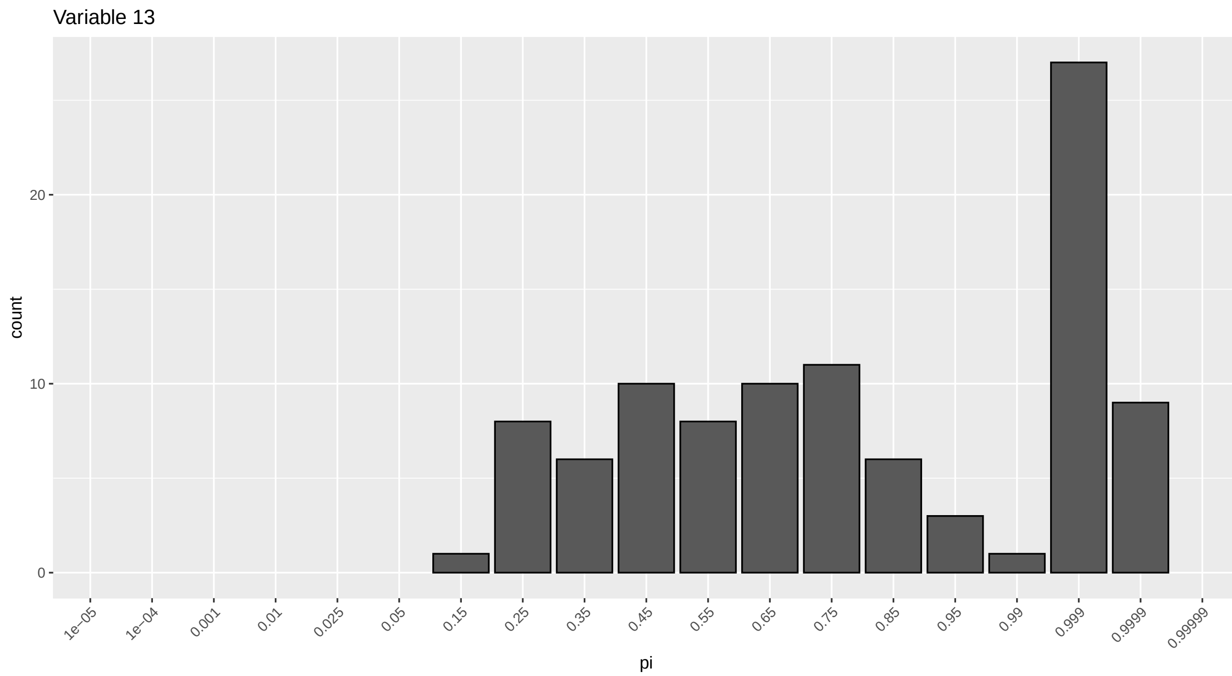
```
##
## [[11]]
```


Variable 11
count
0
5
10
15
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

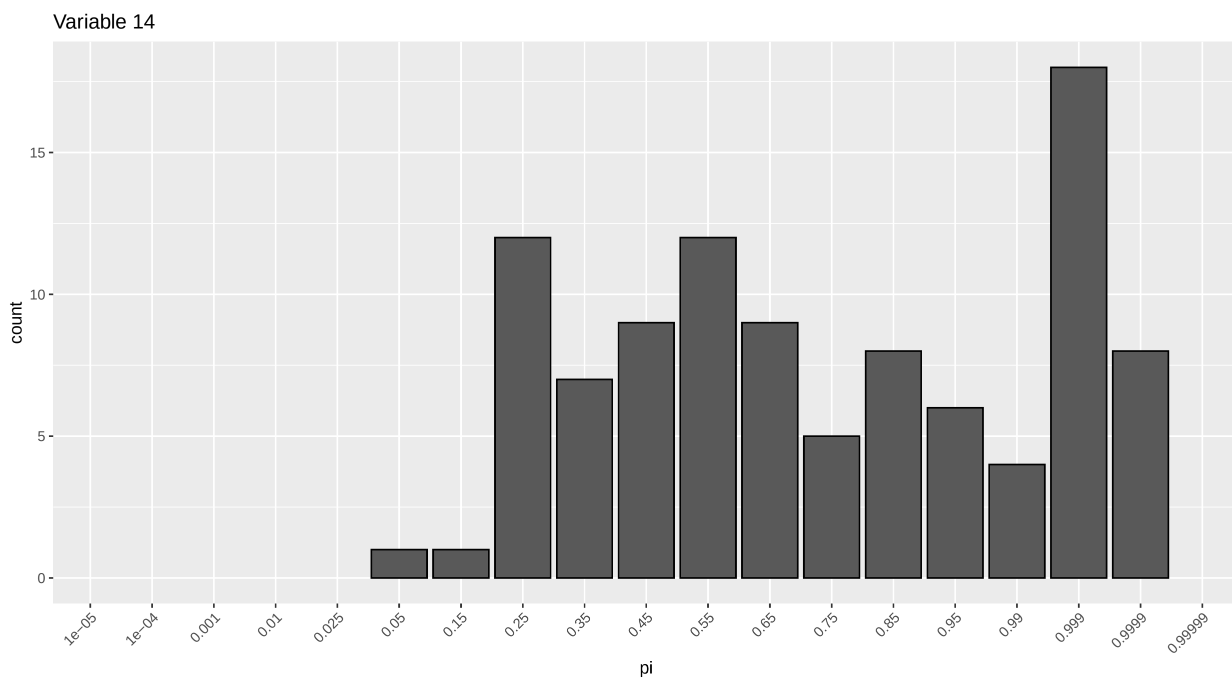
```
##
## [[12]]
```


Variable 12
count
0
5
10
15
20
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

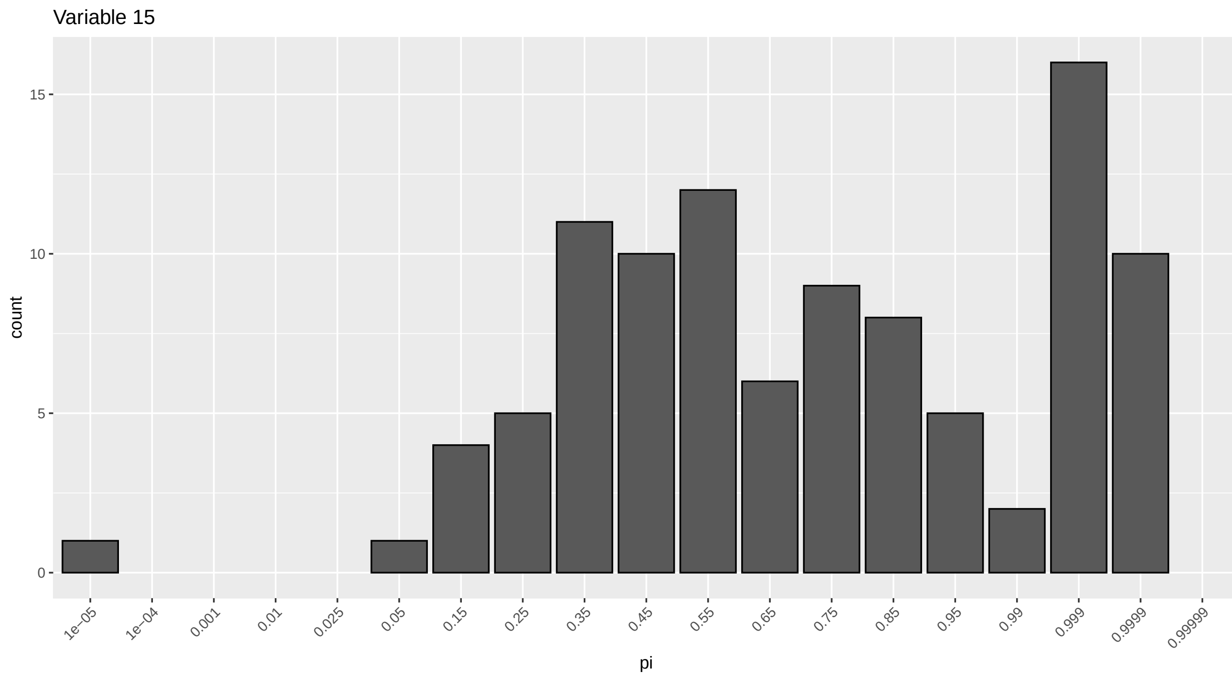
```
##
## [[13]]
```


Variable 13
count
20
10
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

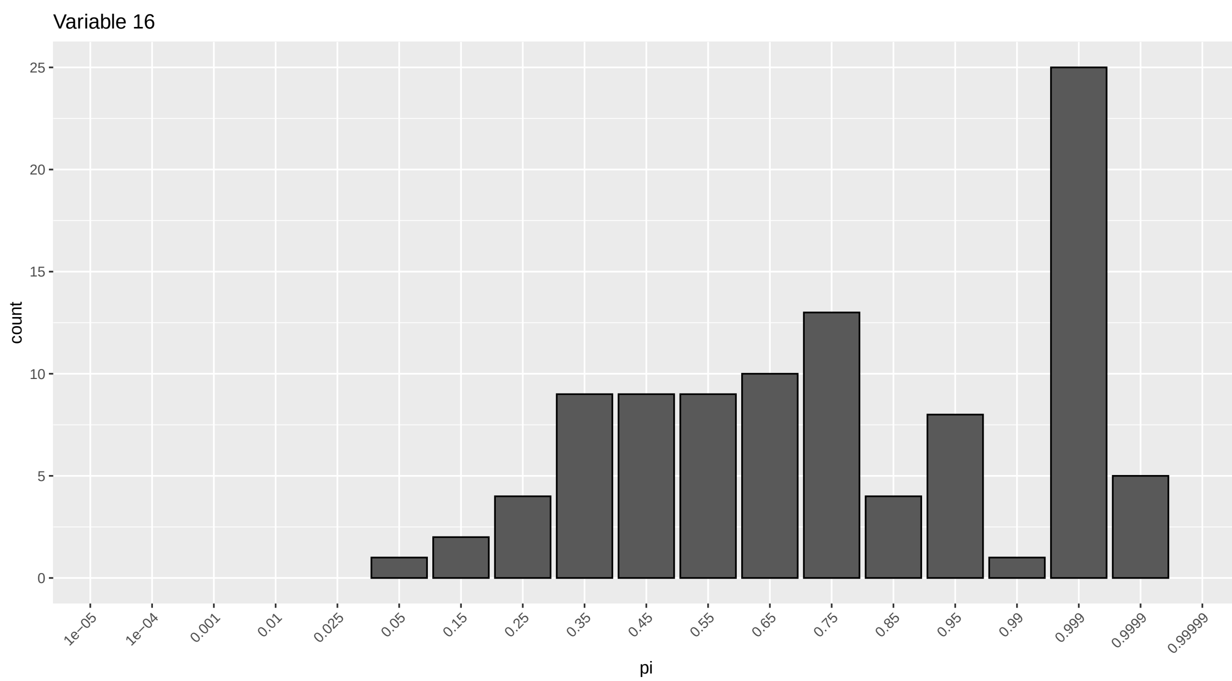
```
##
## [[14]]
```


Variable 14
count
15
10
5
0
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

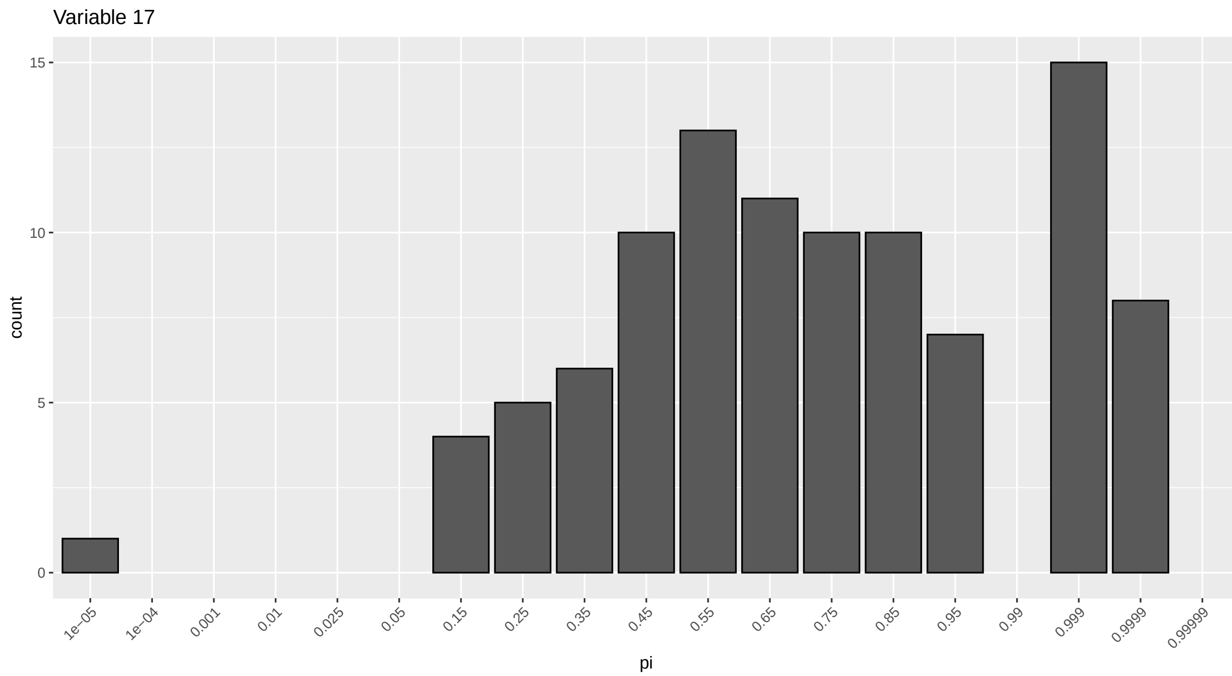
```
##
## [[15]]
```


Variable 15
count
pi
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

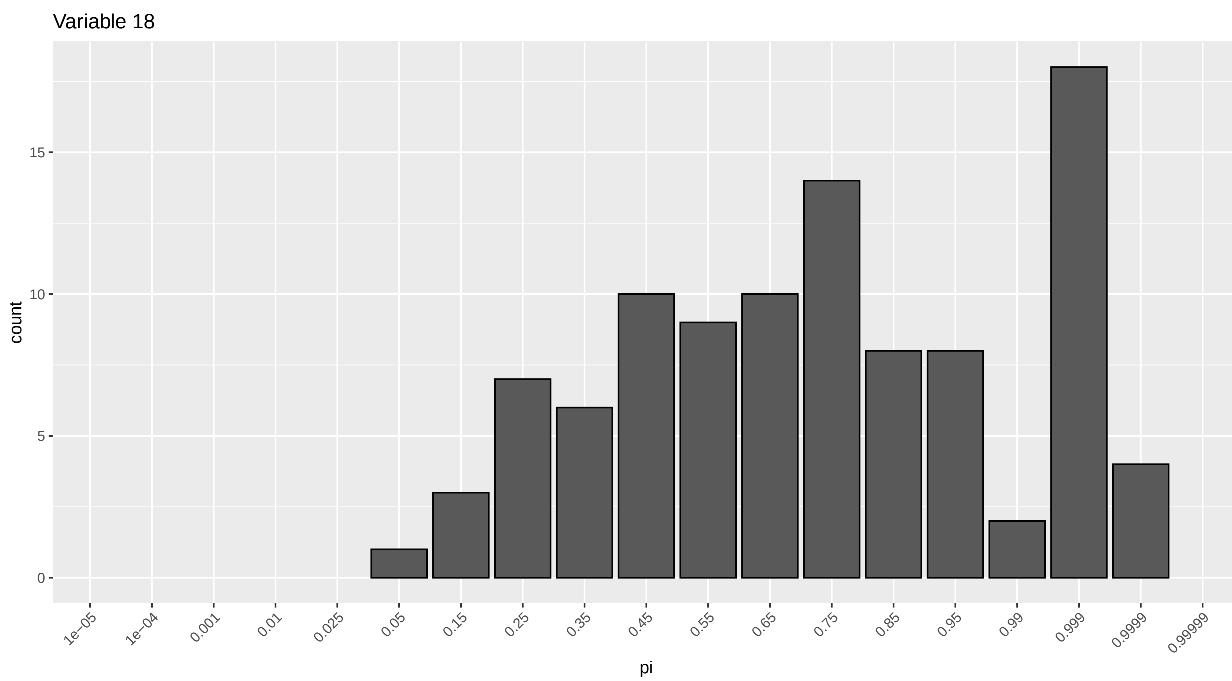
```
##
## [[16]]
```


Variable 16
count
pi
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

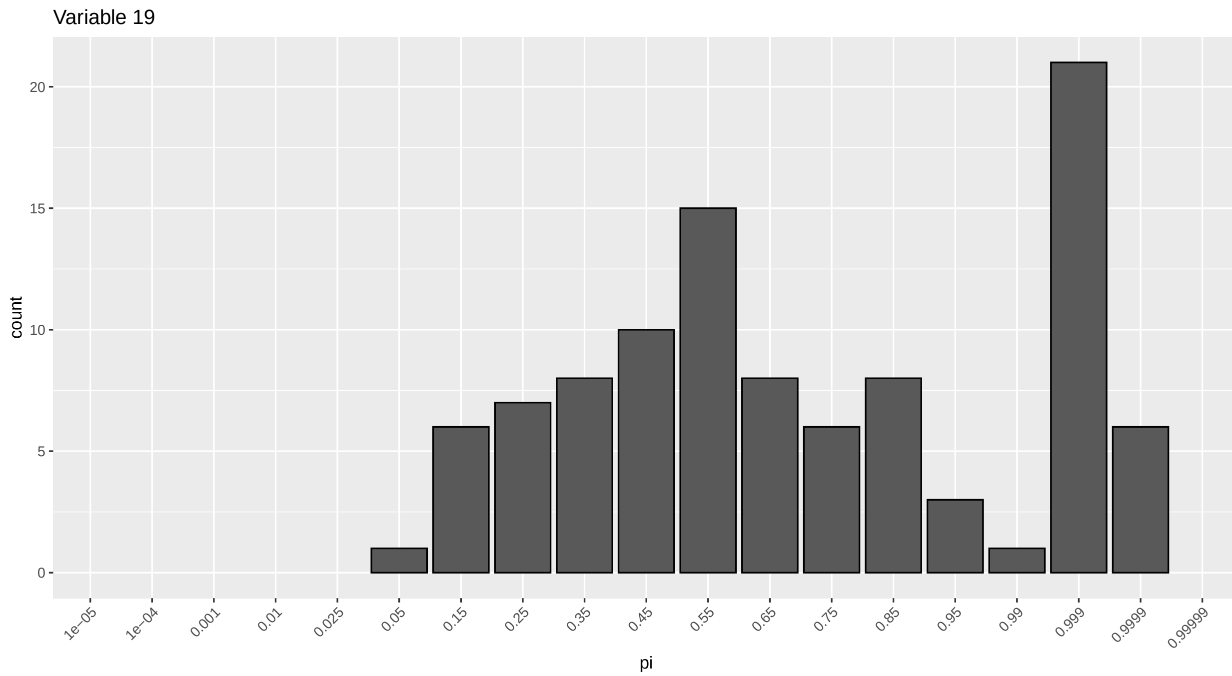
```
##
## [[17]]
```


Variable 17
15
10
5
0
count
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

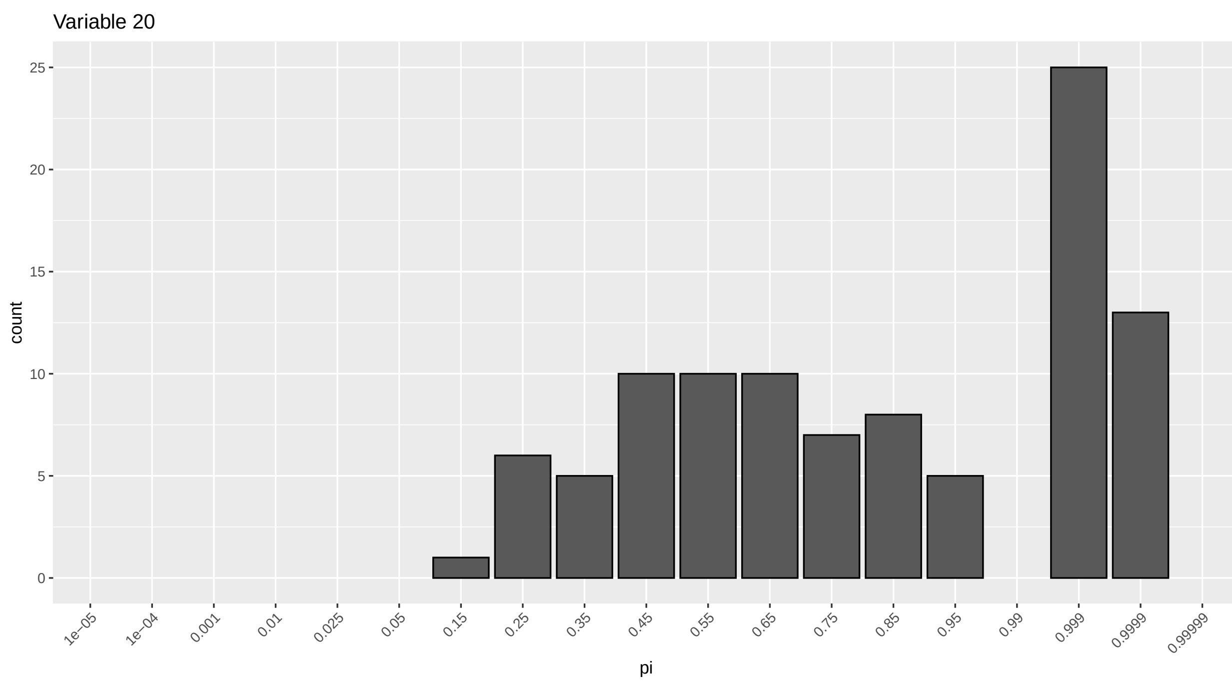
```
##
## [[18]]
```


Variable 18
15
10
5
0
count
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999
pi

```
##
## [[19]]
```


Variable 19
count
pi
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

```
##
## [[20]]
```


Variable 20
count
pi
1e−05
1e−04
0.001
0.01
0.025
0.05
0.15
0.25
0.35
0.45
0.55
0.65
0.75
0.85
0.95
0.99
0.999
0.9999
0.99999

Importance Sampling Hybrid

Algorithm Overview

This algorithm averages over a deterministic grid of pi. For each value of pi, the values of sigmasq and sigmasq_beta are chosen using grid search. The importance weights are calculated by assigning a uniform prior to each value of pi and approximating the marginal likelihood using the ELBO.

Hyperparameter Grid

Below, I display the grid of pi that are averaged over, as well as the marginal grids for sigmasq and sigma-beta_sq.

```
## $pip
## [1] 0.1 0.2 0.3 0.4 0.5
##
## $ssq
##  [1] 0.001 0.010 0.050 0.100 0.250 0.500 0.750 1.000 2.000 3.000
##
## $sbsq
##  [1] 1.0e-05 1.0e-03 1.0e-02 1.0e-01 2.5e-01 5.0e-01 1.0e+00 3.0e+00 5.0e+00
## [10] 1.0e+01
```

Sensitivity and Specificity

In this section, I compare the sensitivity and the specificity for the two methods.

```r
# grid sensitivity
summary(grid_sens)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4286  0.7732  0.8083  0.8041  0.8571  0.9738
```

```r
# importance sensitivity
summary(impt_sens)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4286  0.7065  0.7476  0.7379  0.7911  0.8929
```

```r
# difference in the sensitivities
summary(grid_sens - impt_sens)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.002381  0.023810  0.069048  0.066190  0.091071  0.216667
```

```r
# grid specificity
summary(grid_spec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9672  1.0000  1.0000  0.9991  1.0000  1.0000
```

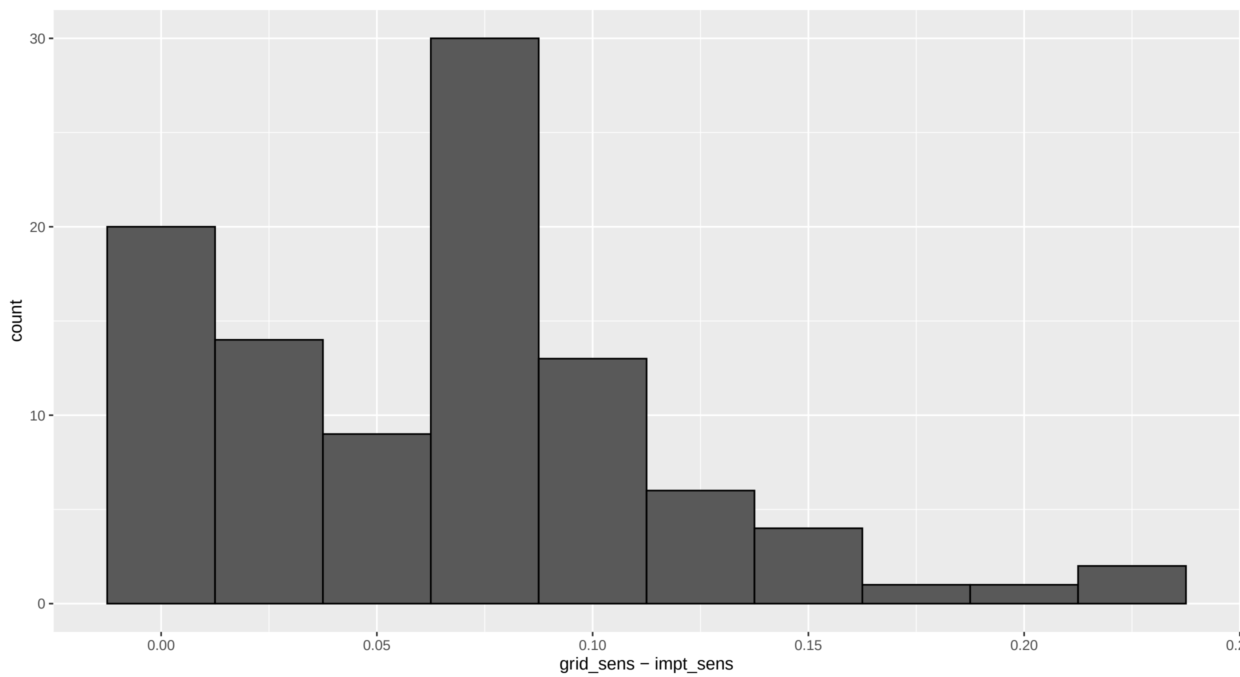
```r
# importance specificity
summary(impt_spec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9885  1.0000  1.0000  0.9999  1.0000  1.0000
```

```r
# difference in the specificities
summary(grid_spec - impt_spec)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0256831  0.0000000  0.0000000 -0.0007432  0.0000000  0.0005464
```

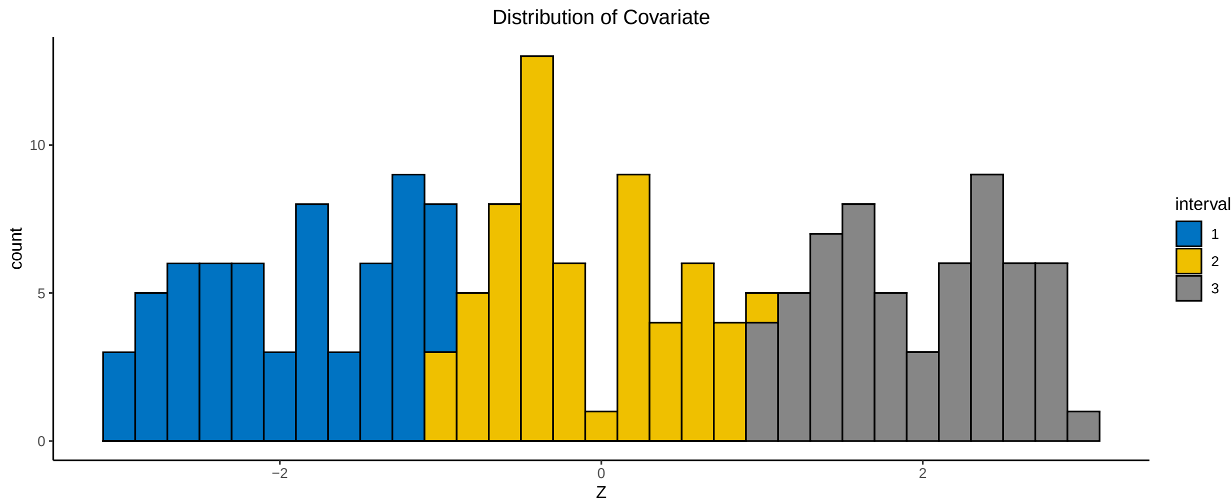
```r
# visualize the difference in the sensitivity
ggplot() + geom_histogram(aes(grid_sens - impt_sens), binwidth = 0.025, color = "black")
```



Data Generation

Extraneous Covariate

I generated the covariate, Z, as the union of three almost disjoint intervals of equal measure. That is, $Z = Z_1 \cup Z_2 \cup Z_3$ with $Z_1 = (-3, -1), Z_2 = (a, b) = (-1, 1), Z_3 = (1, 3)$. Within each interval, I generated 60 covariate values from a uniform distribution. For example:



Precision Matrix

All of the individuals in interval 1 had the same precision matrix, $\Omega^{(1)}$:

$$\Omega^{(1)}_{i,j} = \begin{cases} 2 & i = j \\ 1 & (i,j) \in \{(1,2),(2,1),(2,3),(3,2)\} \\ 0 & o.w. \end{cases}$$

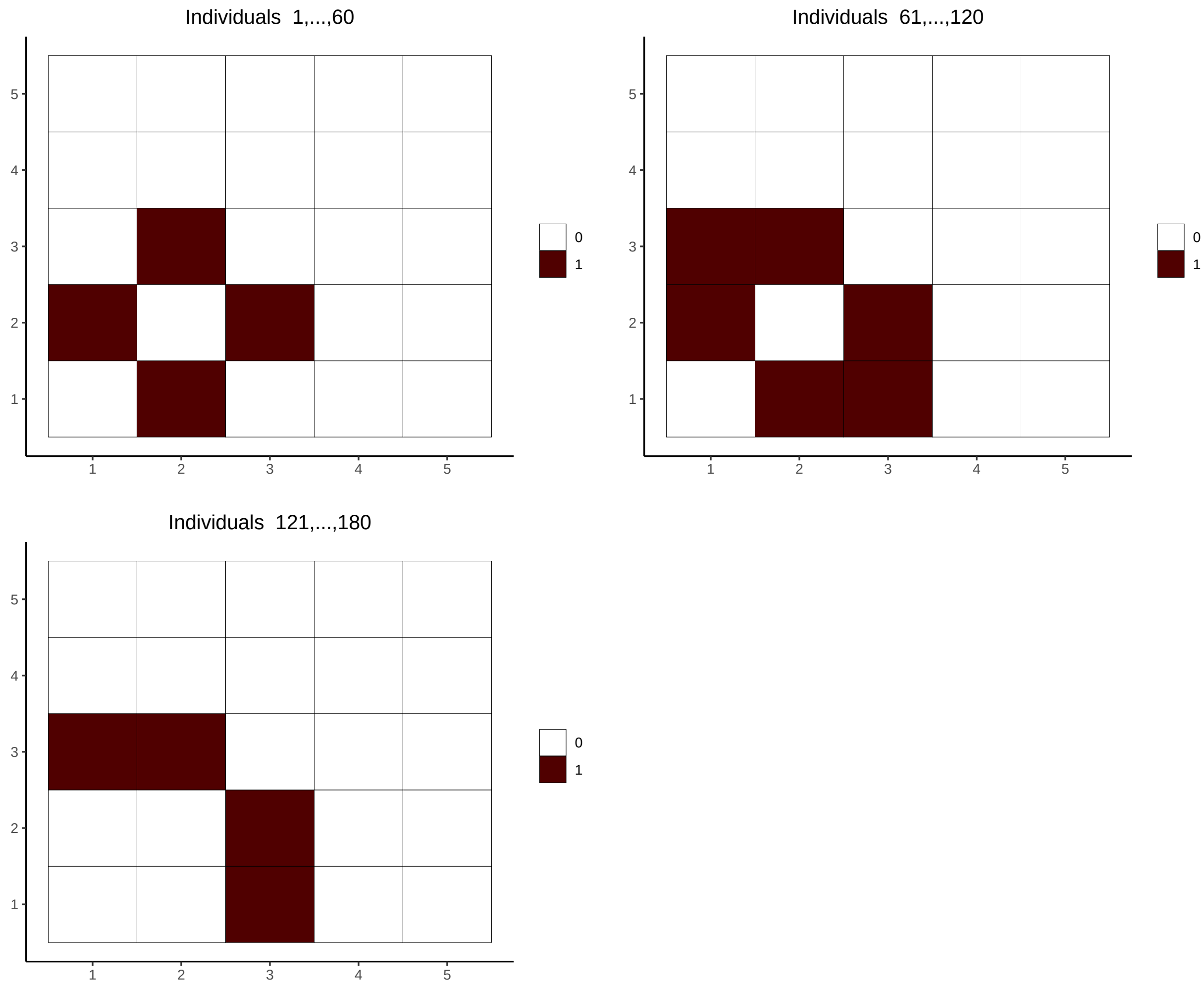
Also, all of the individuals in interval 3 had the same precision matrix, $\Omega^{(3)}$:

$$\Omega^{(3)}_{i,j} = \begin{cases} 2 & i = j \\ 1 & (i,j) \in \{(1,3),(3,1),(2,3),(3,2)\} \\ 0 & o.w. \end{cases}$$

However, the individuals in interval 2 had a precision matrix that was dependent upon Z and (a,b). Let $\beta_0 = -a/(b-a)$ and $\beta_1 = 1/(b-a)$. Then:

$$\Omega^{(2)}_{i,j}(z) = \begin{cases} 2 & i = j \\ 1 & (i,j) \in \{(2,3),(3,2)\} \\ 1 - \beta_0 - \beta_1 z & (i,j) \in \{(1,2),(2,1)\} \\ \beta_0 + \beta_1 z & (i,j) \in \{(1,3),(3,1)\} \\ 0 & o.w. \end{cases}$$

Thus, $\Omega^{(2)}(a) = \Omega^{(1)}$ and $\Omega^{(2)}(b) = \Omega^{(3)}$. That is, an individual on the left or right boundary of Z_2 would have precision matrix $\Omega^{(1)}$ or $\Omega^{(3)}$, respectively. The conditional dependence structures corresponding to each of these precision matrices are visualized below.



Data matrix

Let z_l be the extraneous covariate for the l-th individual. To generate the data matrix for the l-th individual, I took a random sample from $\mathcal{N}(0, \{\Omega_l(z_l)\}^{-1})$, where:

$$\Omega_l(z_l) = \begin{cases} \Omega^{(1)} & z_l \in Z_1 \\ \Omega^{(2)}(z_l) & z_l \in Z_2 \\ \Omega^{(3)} & z_l \in Z_3 \end{cases}$$