Thesis Simulation Results Analysis for Chapter 4

Dasha Asienga

2024-04-06

Contents

Reading in the Result Data Sets 1
- Logistic Regression . . . 1
- Seldonian Solutions . . . 2

Combining the Data Sets from Both Simulations 3

Probability of a Solution 3
- Table . . . 3

Accuracy 4
- Table . . . 4
- Visualizations . . . 4

Discrimination 7
- Tables . . . 7
- Visualizations . . . 8

Accuracy-Discrimination Trade-Off 10

Convergence-Discrimination Trade-Off 13

NSF 16
- Accuracy . . . 16
- Accuracy of Logistic Regression in Failed Seldonian Solutions . . . 19
- Discrimination . . . 20
- Discrimination of Logistic Regression in Failed Seldonian Solutions . . . 24

This file is intended to synthesize and analyze the results from the simulation.

Reading in the Result Data Sets

Logistic Regression

The results data set has 1000 observations for each of the simulation trials, 250 from each sample size: $n = 500, 1000, 2500, 5000$.

```
lr_500 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Re
lr_1000 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/R
lr_2500 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/R
```

```r
lr_5000 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Re

lr_500 <- lr_500 |>
  mutate(sample_size = 500) |>
  dplyr::select(-X)

lr_1000 <- lr_1000 |>
  mutate(sample_size = 1000) |>
  dplyr::select(-X)

lr_2500 <- lr_2500 |>
  mutate(sample_size = 2500) |>
  dplyr::select(-X)

lr_5000 <- lr_5000 |>
  mutate(sample_size = 5000) |>
  dplyr::select(-X)

logistic_results <- rbind(lr_500, lr_1000, lr_2500, lr_5000)
glimpse(logistic_results)
```

```
## Rows: 1,000
## Columns: 5
## $ dataset_id        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ lr_convergence    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ lr_accuracy       <dbl> 0.790, 0.738, 0.778, 0.758, 0.788, 0.822, 0.776, 0.7~
## $ lr_discrimination <dbl> 0.2020, 0.2402, 0.2730, 0.3123, 0.2663, 0.2171, 0.31~
## $ sample_size       <dbl> 500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 50~
```

Seldonian Solutions

The results data set has 1000 observations for each of the simulation trials, 250 from each sample size: $n = 500, 1000, 2500, 5000$.

```r
seldonian_results <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/Python/COMPAS Simulatio
#seldonian_results <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/Data Sets/seldonian_si
glimpse(seldonian_results)
```

```
## Rows: 1,000
## Columns: 14
## $ sample_size       <int> 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000~
## $ dataset_id        <int> 100, 11, 139, 127, 112, 133, 60, 144, 140, 129, 130,~
## $ passed_safety_02  <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_01  <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_005 <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_001 <chr> "False", "True", "True", "True", "True", "True", "Tr~
## $ sa_02_accuracy    <dbl> 0.5240, 0.6190, 0.6410, 0.5100, 0.5150, 0.5220, 0.72~
## $ sa_01_accuracy    <dbl> 0.5220, 0.5060, 0.5040, 0.5100, 0.5250, 0.5220, 0.50~
## $ sa_005_accuracy   <dbl> 0.5220, 0.5060, 0.5040, 0.5100, 0.5070, 0.5220, 0.50~
## $ sa_001_accuracy   <dbl> 0.5220, 0.7190, 0.5040, 0.5100, 0.5070, 0.5220, 0.50~
## $ sa_02_disc_stat   <dbl> NA, 0.1226, 0.1052, 0.0000, 0.0175, 0.0000, 0.2781, ~
## $ sa_01_disc_stat   <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0435, 0.0000, 0.00~
## $ sa_005_disc_stat  <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.00~
## $ sa_001_disc_stat  <dbl> 0.0000, 0.2273, 0.0000, 0.0000, 0.0000, 0.0000, 0.00~
```

Combining the Data Sets from Both Simulations

```r
sim_results <- inner_join(logistic_results, seldonian_results,
                          by = c("sample_size", "dataset_id"))

glimpse(sim_results)
```

```
## Rows: 1,000
## Columns: 17
## $ dataset_id        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ lr_convergence    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ lr_accuracy       <dbl> 0.790, 0.738, 0.778, 0.758, 0.788, 0.822, 0.776, 0.7~
## $ lr_discrimination <dbl> 0.2020, 0.2402, 0.2730, 0.3123, 0.2663, 0.2171, 0.31~
## $ sample_size       <dbl> 500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 50~
## $ passed_safety_02  <chr> "True", "True", "True", "True", "True", "True", "Tru~
## $ passed_safety_01  <chr> "True", "True", "False", "True", "True", "False", "T~
## $ passed_safety_005 <chr> "True", "True", "False", "True", "True", "False", "T~
## $ passed_safety_001 <chr> "True", "True", "False", "True", "True", "False", "T~
## $ sa_02_accuracy    <dbl> 0.526, 0.510, 0.528, 0.520, 0.512, 0.508, 0.608, 0.5~
## $ sa_01_accuracy    <dbl> 0.522, 0.510, 0.528, 0.520, 0.496, 0.508, 0.482, 0.5~
## $ sa_005_accuracy   <dbl> 0.522, 0.510, 0.528, 0.520, 0.496, 0.508, 0.756, 0.5~
## $ sa_001_accuracy   <dbl> 0.522, 0.510, 0.528, 0.520, 0.496, 0.508, 0.482, 0.5~
## $ sa_02_disc_stat   <dbl> NA, 0.0000, 0.0000, 0.0000, 0.0429, 0.0000, 0.0106, ~
## $ sa_01_disc_stat   <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.00~
## $ sa_005_disc_stat  <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.13~
## $ sa_001_disc_stat  <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.00~
```

Probability of a Solution

Table

This section assesses what proportion of the trials returned a solution. It is expected that all logistic regression trials will return a solution. However, for the Seldonian algorithms, while all trials will return a candidate solution (based on the logistic regression as a starting point), it is expected that not all candidate solutions will pass the safety test. The table below records the number of Seldonian solutions that passed the safety test in each sample size.

```r
reps <- nrow(sim_results)/4

sim_results |>
  group_by(sample_size) |>
  summarise(LR = 100*count(lr_convergence == 1)/reps,
            `SA (0.2)` = 100*count(passed_safety_02 == "True")/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "True")/reps,
            `SA (0.05)` = 100*count(passed_safety_005 == "True")/reps,
            `SA (0.01)` = 100*count(passed_safety_001 == "True")/reps) |>
  rename("Sample Size" = sample_size) |>
  kable(caption = "Probability of a Solution")
```

Table 1: Probability of a Solution

Sample Size	LR	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
500	100	99.6	93.2	89.6	71.2

Sample Size	LR	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
1000	100	100.0	98.8	97.6	86.8
2500	100	99.6	100.0	98.8	56.8
5000	100	99.6	99.6	96.4	24.8

Accuracy

```
# filter accuracy data out for only seldonian solutions that converged
sim_results_converged_accuracy <- sim_results |>
  mutate(sa_02_accuracy = ifelse(passed_safety_02 == "True", sa_02_accuracy, NA),
         sa_01_accuracy = ifelse(passed_safety_01 == "True", sa_01_accuracy, NA),
         sa_005_accuracy = ifelse(passed_safety_005 == "True", sa_005_accuracy, NA),
         sa_001_accuracy = ifelse(passed_safety_001 == "True", sa_001_accuracy, NA))
```

Table

```
# table
sim_results_converged_accuracy |>
  group_by(sample_size) |>
  summarise(LR = round(100*mean(lr_accuracy, na.rm = TRUE),2),
            sd_lr = round(sd(lr_accuracy, na.rm = TRUE),2),
            `SA (0.2)` = round(100*mean(sa_02_accuracy, na.rm = TRUE),2),
            sd_02 = round(sd(sa_02_accuracy, na.rm = TRUE),2),
            `SA (0.1)` = round(100*mean(sa_01_accuracy, na.rm = TRUE),2),
            sd_01 = round(sd(sa_01_accuracy, na.rm = TRUE),2),
            `SA (0.05)` = round(100*mean(sa_005_accuracy, na.rm = TRUE),2),
            sd_005 = round(sd(sa_005_accuracy, na.rm = TRUE),2),
            `SA (0.01)` = round(100*mean(sa_001_accuracy, na.rm = TRUE),2),
            sd_001 = round(sd(sa_001_accuracy, na.rm = TRUE),2)) |>
  rename("Sample Size" = sample_size,
         "sd" = sd_lr,
         "sd " = sd_02,
         "sd  " = sd_01,
         "sd   " = sd_005,
         "sd    " = sd_001) |>
  kable(caption = "Accuracy of Convergent Solutions")
```

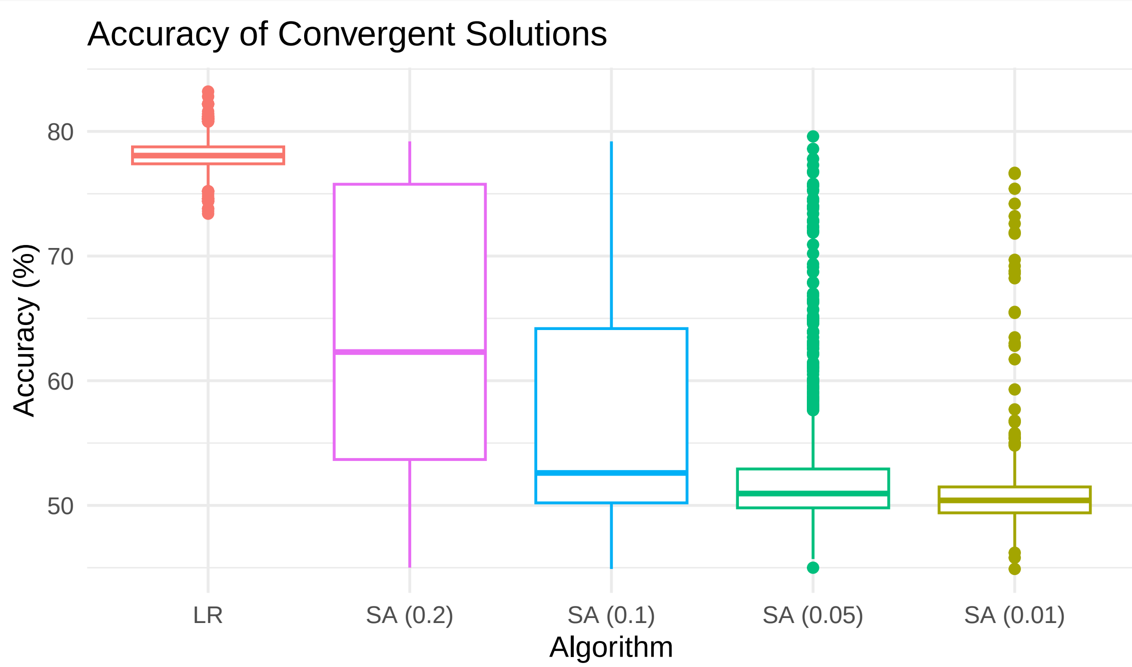
Table 2: Accuracy of Convergent Solutions

Sample Size	LR	sd	SA (0.2)	sd	SA (0.1)	sd	SA (0.05)	sd	SA (0.01)	sd
500	78.16	0.02	54.01	0.06	52.02	0.06	52.31	0.06	51.54	0.05
1000	78.10	0.01	58.32	0.07	51.90	0.05	51.47	0.05	50.88	0.04
2500	78.06	0.01	73.27	0.06	58.60	0.06	51.79	0.04	50.89	0.03
5000	77.97	0.01	69.52	0.08	66.05	0.09	54.95	0.06	50.32	0.02

Visualizations

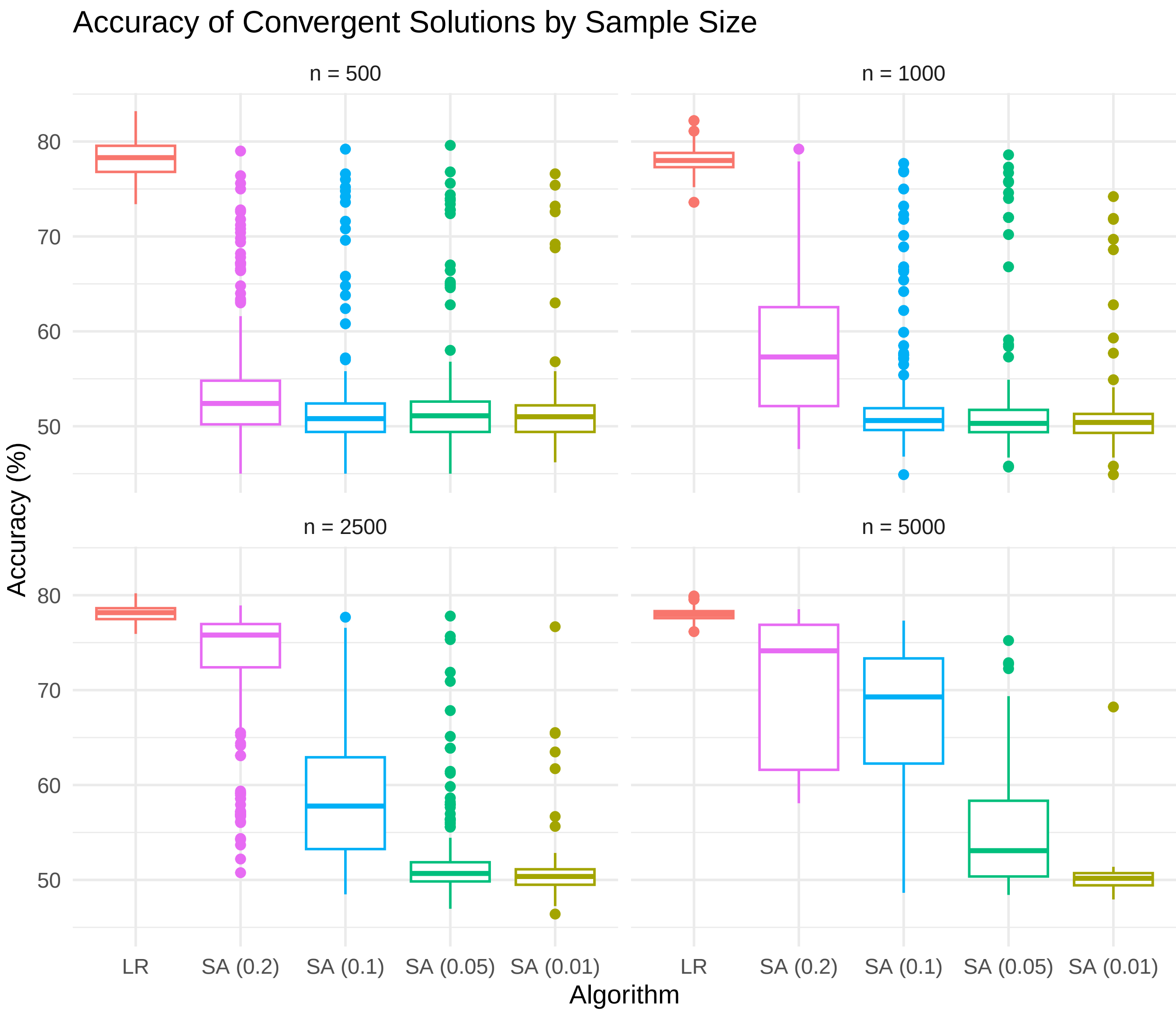
```
# visualization
sim_results_converged_accuracy |>
```

```r
  dplyr::select(c(lr_accuracy, sa_02_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size)) |>
  rename("Sample Size" = sample_size,
         "LR" = lr_accuracy,
         "SA (0.2)" = sa_02_accuracy,
         "SA (0.1)" = sa_01_accuracy,
         "SA (0.05)" = sa_005_accuracy,
         "SA (0.01)" = sa_001_accuracy) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "accuracy") |>
  mutate(accuracy = 100*accuracy) |>
  ggplot(mapping = aes(x = model, y = accuracy, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("LR", "SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  labs(x = "Algorithm",
       y = "Accuracy (%)",
       title = "Accuracy of Convergent Solutions")
```




```r
# visualization
sim_results_converged_accuracy |>
  dplyr::select(c(lr_accuracy, sa_02_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000"))) |>
  rename("Sample Size" = sample_size,
         "LR" = lr_accuracy,
```

```r
         "SA (0.2)" = sa_02_accuracy,
         "SA (0.1)" = sa_01_accuracy,
         "SA (0.05)" = sa_005_accuracy,
         "SA (0.01)" = sa_001_accuracy) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "accuracy") |>
  mutate(accuracy = 100*accuracy) |>
  ggplot(mapping = aes(x = model, y = accuracy, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("LR", "SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  facet_wrap(~`Sample Size`) +
  labs(x = "Algorithm",
       y = "Accuracy (%)",
       title = "Accuracy of Convergent Solutions by Sample Size")
```



Discrimination

```r
# filter discrimination data out for only seldonian solutions that converged
sim_results_converged_disc <- sim_results |>
  mutate(sa_02_disc_stat = ifelse(passed_safety_02 == "True", sa_02_disc_stat, NA),
         sa_01_disc_stat = ifelse(passed_safety_01 == "True", sa_01_disc_stat, NA),
         sa_005_disc_stat = ifelse(passed_safety_005 == "True", sa_005_disc_stat, NA),
         sa_001_disc_stat = ifelse(passed_safety_001 == "True", sa_001_disc_stat, NA))
```

Tables

```r
# table
sim_results_converged_disc |>
  group_by(sample_size) |>
  summarise(LR = round(mean(lr_discrimination, na.rm = TRUE),2),
            sd_lr = round(sd(lr_discrimination, na.rm = TRUE),2),
            `SA (0.2)` = round(mean(sa_02_disc_stat, na.rm = TRUE),2),
            sd_02 = round(sd(sa_02_disc_stat, na.rm = TRUE),2),
            `SA (0.1)` = round(mean(sa_01_disc_stat, na.rm = TRUE),2),
            sd_01 = round(sd(sa_01_disc_stat, na.rm = TRUE),2),
            `SA (0.05)` = round(mean(sa_005_disc_stat, na.rm = TRUE),2),
            sd_005 = round(sd(sa_005_disc_stat, na.rm = TRUE),2),
            `SA (0.01)` = round(mean(sa_001_disc_stat, na.rm = TRUE),2),
            sd_001 = round(sd(sa_001_disc_stat, na.rm = TRUE),2)) |>
  rename("Sample Size" = sample_size,
         "sd" = sd_lr,
         "sd " = sd_02,
         "sd  " = sd_01,
         "sd   " = sd_005,
         "sd     " = sd_001) |>
  kable(caption = "Discrimination of Convergent Solutions")
```

Table 3: Discrimination of Convergent Solutions

Sample Size	LR	sd	SA (0.2)	sd	SA (0.1)	sd	SA (0.05)	sd	SA (0.01)	sd
500	0.24	0.07	0.04	0.07	0.02	0.06	0.02	0.06	0.01	0.05
1000	0.24	0.05	0.09	0.07	0.02	0.06	0.02	0.05	0.01	0.03
2500	0.24	0.03	0.18	0.06	0.09	0.06	0.02	0.04	0.01	0.03
5000	0.24	0.02	0.12	0.06	0.14	0.08	0.05	0.05	0.00	0.02

```r
# table
sim_results_converged_disc |>
  group_by(sample_size) |>
  summarise(`SA (0.2)` = round(100*count(sa_02_disc_stat <= 0.2)/count(passed_safety_02 == "True"),2),
            `SA (0.1)` = round(100*count(sa_01_disc_stat <= 0.1)/count(passed_safety_01 == "True"),2),
            `SA (0.05)` = round(100*count(sa_005_disc_stat <= 0.05)/count(passed_safety_005 == "True"),
            `SA (0.01)` = round(100*count(sa_001_disc_stat <= 0.01)/count(passed_safety_001 == "True"),
  rename("Sample Size" = sample_size) |>
  kable(caption = "Satisfaction of the Behavioral Constraint by Convergent Solutions")
```

Table 4: Satisfaction of the Behavioral Constraint by Convergent Solutions

Sample Size	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
500	83.53	87.98	88.39	93.82
1000	85.20	83.40	88.11	93.09
2500	56.22	63.20	76.11	92.25
5000	90.76	24.90	61.41	95.16

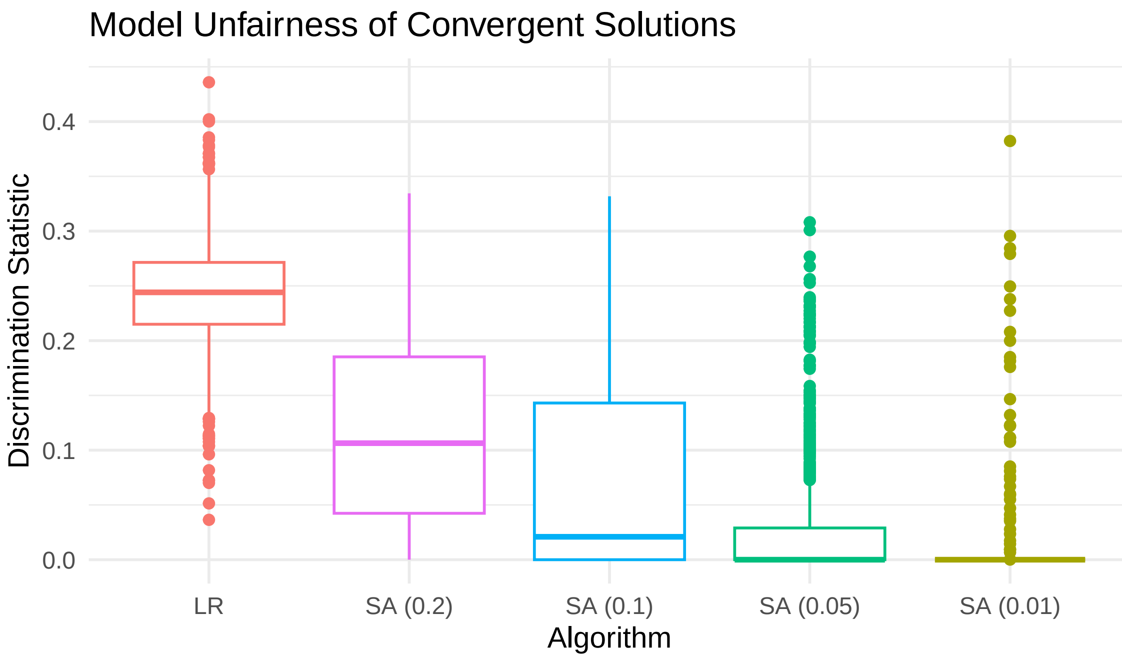
```
# what % returned a solution and also met the constraint
sim_results |>
  group_by(sample_size) |>
  summarise(`SA (0.2)` = 100*count(passed_safety_02 == "True" & sa_02_disc_stat <= 0.2)/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "True" & sa_01_disc_stat <= 0.1)/reps,
            `SA (0.05)` = 100*count(passed_safety_005 == "True" & sa_005_disc_stat <= 0.05)/reps,
            `SA (0.01)` = 100*count(passed_safety_001 == "True" & sa_001_disc_stat <= 0.01)/reps) |>
  rename("Sample Size" = sample_size) |>
  kable(caption = "Probability of a Returning a Solution and Meeting the Satisfied Constraint")
```

Table 5: Probability of a Returning a Solution and Meeting the Satisfied Constraint

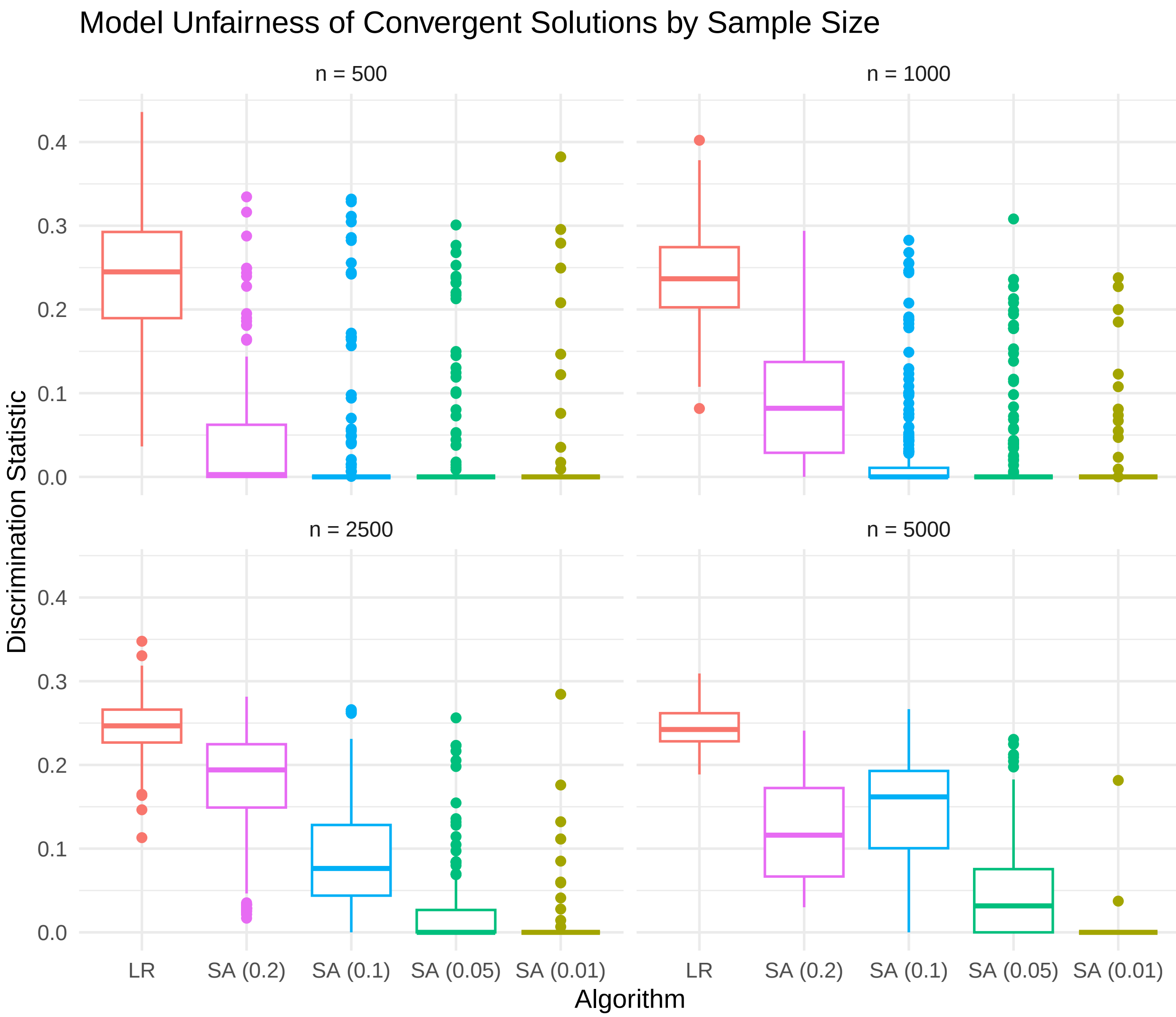
Sample Size	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
500	83.2	82.0	79.2	66.8
1000	85.2	82.4	86.0	80.8
2500	56.0	63.2	75.2	52.4
5000	90.4	24.8	59.2	23.6

Visualizations

```
# visualization
sim_results_converged_disc |>
  dplyr::select(c(lr_discrimination, sa_02_disc_stat, sa_01_disc_stat, sa_005_disc_stat,
                  sa_001_disc_stat, sample_size)) |>
  rename("Sample Size" = sample_size,
         "LR" = lr_discrimination,
         "SA (0.2)" = sa_02_disc_stat,
         "SA (0.1)" = sa_01_disc_stat,
         "SA (0.05)" = sa_005_disc_stat,
         "SA (0.01)" = sa_001_disc_stat) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "discrimination") |>
  ggplot(mapping = aes(x = model, y = discrimination, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("LR", "SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  labs(x = "Algorithm",
       y = "Discrimination Statistic",
       title = "Model Unfairness of Convergent Solutions")
```




```r
# visualization
sim_results_converged_disc |>
  dplyr::select(c(lr_discrimination, sa_02_disc_stat, sa_01_disc_stat, sa_005_disc_stat,
                  sa_001_disc_stat, sample_size)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000"))) |>
  rename("Sample Size" = sample_size,
         "LR" = lr_discrimination,
         "SA (0.2)" = sa_02_disc_stat,
         "SA (0.1)" = sa_01_disc_stat,
         "SA (0.05)" = sa_005_disc_stat,
         "SA (0.01)" = sa_001_disc_stat) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "discrimination") |>
  ggplot(mapping = aes(x = model, y = discrimination, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("LR", "SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  facet_wrap(~`Sample Size`) +
  labs(x = "Algorithm",
       y = "Discrimination Statistic",
       title = "Model Unfairness of Convergent Solutions by Sample Size")
```



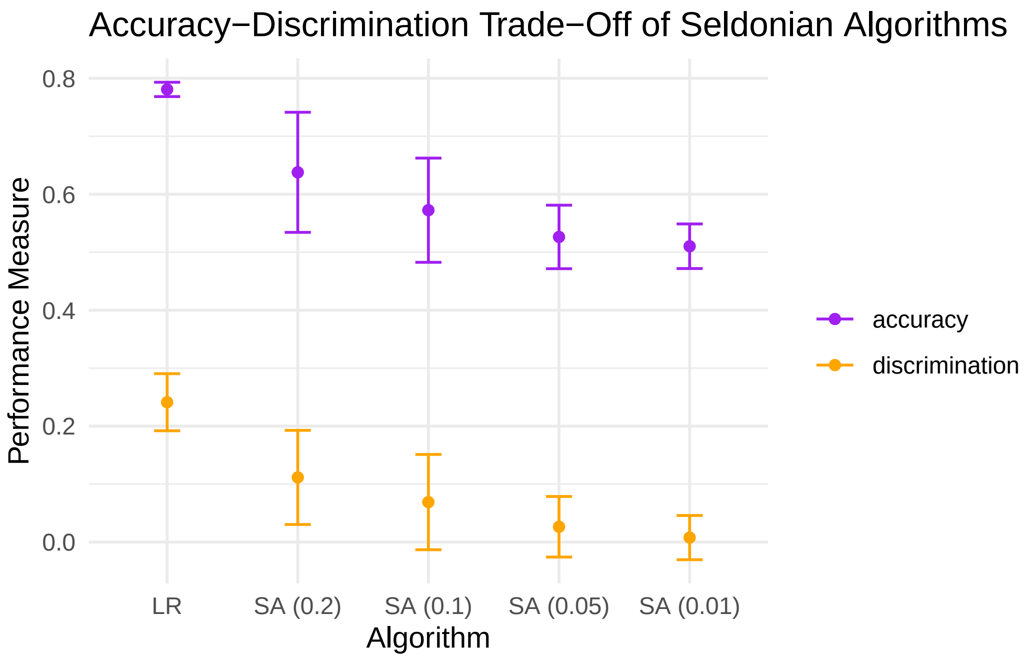
Accuracy-Discrimination Trade-Off

This section aims to visualize the accuracy-discrimination trade-off for each of the models, by sample-size.

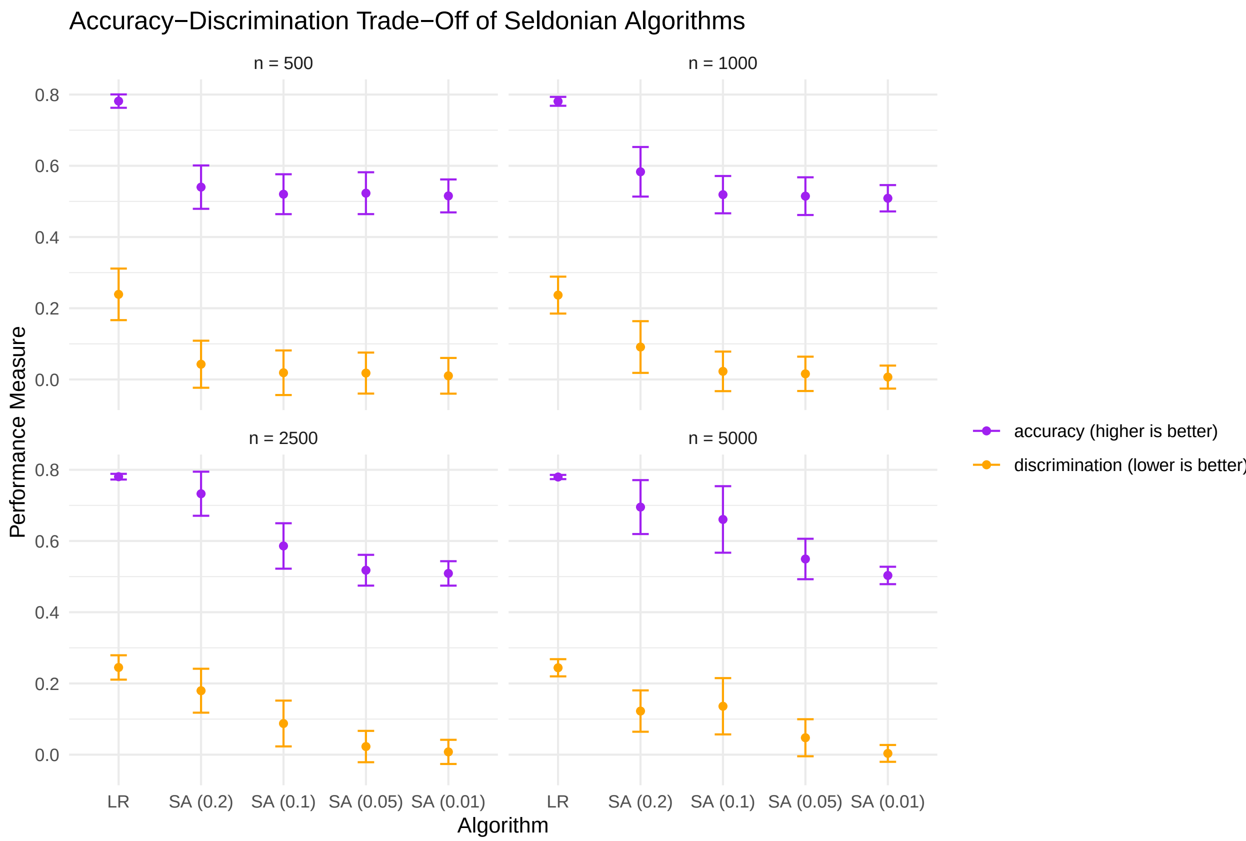
```
# get long discrimination data set for plotting (1000 rows)
sim_results_converged_disc_long <- sim_results_converged_disc |>
  dplyr::select(c(lr_discrimination, sa_02_disc_stat, sa_01_disc_stat, sa_005_disc_stat,
                  sa_001_disc_stat, sample_size, dataset_id)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                                    levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000"))) |>
  rename("Sample Size" = sample_size,
         "LR" = lr_discrimination,
         "SA (0.2)" = sa_02_disc_stat,
         "SA (0.1)" = sa_01_disc_stat,
         "SA (0.05)" = sa_005_disc_stat,
```

```r
         "SA (0.01)" = sa_001_disc_stat) |>
  pivot_longer(cols = -c(`Sample Size`, dataset_id),
               names_to = "model",
               values_to = "discrimination")

# get long accuracy data set for plotting (1000 rows)
sim_results_converged_accuracy_long <- sim_results_converged_accuracy |>
  dplyr::select(c(lr_accuracy, sa_02_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size, dataset_id)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000"))) |>
  rename("Sample Size" = sample_size,
         "LR" = lr_accuracy,
         "SA (0.2)" = sa_02_accuracy,
         "SA (0.1)" = sa_01_accuracy,
         "SA (0.05)" = sa_005_accuracy,
         "SA (0.01)" = sa_001_accuracy) |>
  pivot_longer(cols = -c(`Sample Size`, dataset_id),
               names_to = "model",
               values_to = "accuracy")

# join both data sets
accuracy_disc <- inner_join(sim_results_converged_disc_long, sim_results_converged_accuracy_long,
                            by = c("dataset_id", "Sample Size", "model")) |>
  pivot_longer(cols = -c(`Sample Size`, dataset_id, model),
               names_to = "statistic",
               values_to = "value")
```

```r
# plot error-bars
color <- c("purple", "orange")

accuracy_disc |>
  group_by(model, statistic) |>
  summarise(avg = mean(value, na.rm = TRUE),
            se = sd(value, na.rm = TRUE)) |>
  ggplot(mapping = aes(x = model, y = avg, color = statistic)) +
  geom_point() +
  geom_errorbar(mapping = aes(ymin = avg - se,
                              ymax = avg + se,
                              width = 0.2)) +
  theme_minimal() +
  scale_color_manual(values = c("accuracy" = color[1],
                                "discrimination" = color[2])) +
  scale_x_discrete(limits = c("LR", "SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  labs(x = "Algorithm",
       y = "Performance Measure",
       title = "Accuracy-Discrimination Trade-Off of Seldonian Algorithms",
       color = " ")
```




```r
# plot error-bars by sample size
color <- c("purple", "orange")

accuracy_disc |>
  group_by(model, statistic, `Sample Size`) |>
  summarise(avg = mean(value, na.rm = TRUE),
            se = sd(value, na.rm = TRUE)) |>
  mutate(statistic = ifelse(statistic == "accuracy", "accuracy (higher is better)", "discrimination (lo
  ggplot(mapping = aes(x = model, y = avg, color = statistic)) +
  geom_point() +
  geom_errorbar(mapping = aes(ymin = avg - se,
                              ymax = avg + se,
                              width = 0.2)) +
  theme_minimal() +
  scale_color_manual(values = c("accuracy (higher is better)" = color[1],
                                "discrimination (lower is better)" = color[2])) +
  scale_x_discrete(limits = c("LR", "SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  facet_wrap(~`Sample Size`) +
  labs(x = "Algorithm",
       y = "Performance Measure",
       title = "Accuracy-Discrimination Trade-Off of Seldonian Algorithms",
       color = " ")
```



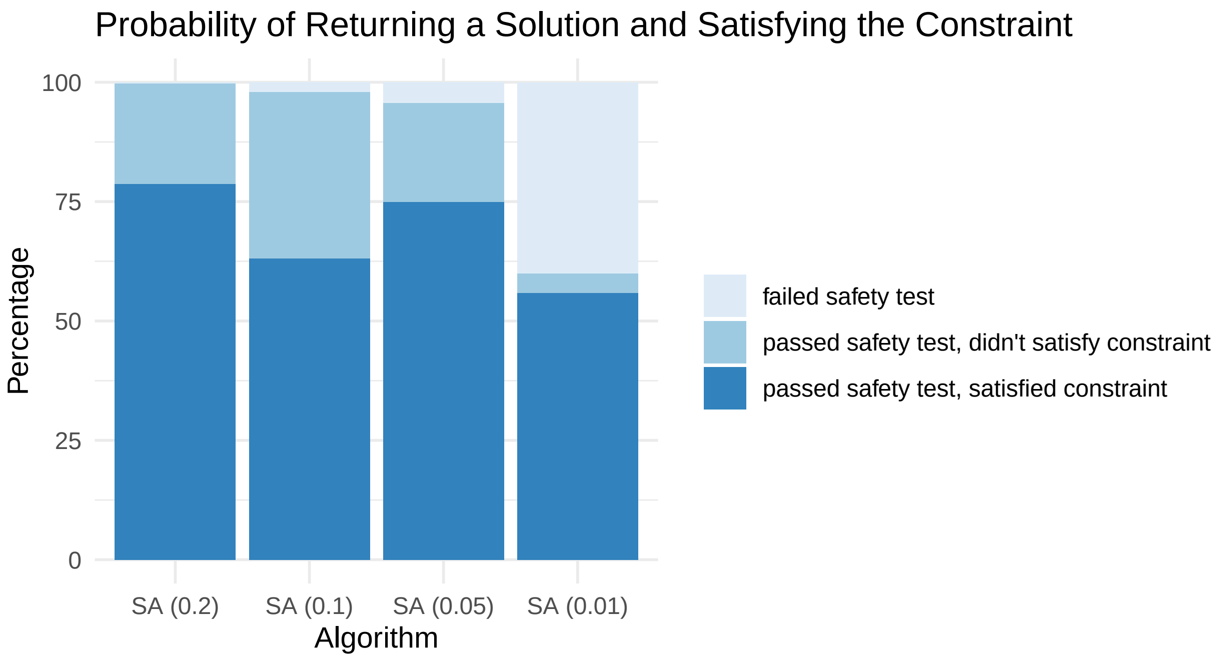
Convergence-Discrimination Trade-Off

This section examines whether there is a trade-off in the proportion of solutions that pass the safety test and the proportion of Seldonian solutions that satisfy the fairness constraint.

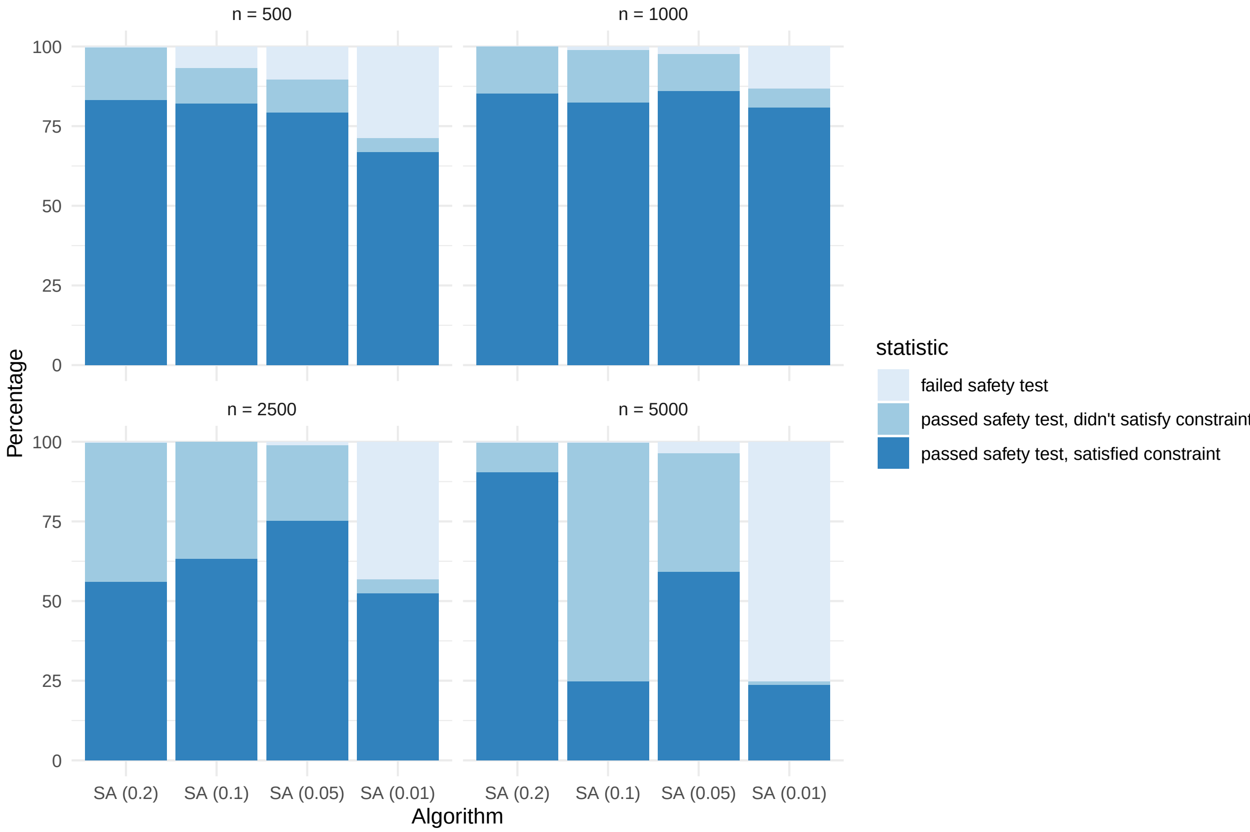
```
# get long failure data for plotting
sim_failure_long <- sim_results |>
  group_by(sample_size) |>
  summarise(`SA (0.2)` = 100*count(passed_safety_02 == "False")/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "False")/reps,
            `SA (0.05)` = 100*count(passed_safety_005 == "False")/reps,
            `SA (0.01)` = 100*count(passed_safety_001 == "False")/reps) |>
  pivot_longer(cols = -c(sample_size),
               names_to = "model",
               values_to = "prop_failed_safety")


# get long convergence data for plotting
sim_converged_long <- sim_results |>
  group_by(sample_size) |>
  summarise(`SA (0.2)` = 100*count(passed_safety_02 == "True")/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "True")/reps,
            `SA (0.05)` = 100*count(passed_safety_005 == "True")/reps,
            `SA (0.01)` = 100*count(passed_safety_001 == "True")/reps) |>
  pivot_longer(cols = -c(sample_size),
               names_to = "model",
               values_to = "prop_passed_safety")
```

```r
# get long discrimination data for plotting
sim_satisfied_cstr_long <- sim_results |>
  group_by(sample_size) |>
  summarise(`SA (0.2)` = 100*count(passed_safety_02 == "True" & sa_02_disc_stat <= 0.2)/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "True" & sa_01_disc_stat <= 0.1)/reps,
            `SA (0.05)` = 100*count(passed_safety_005 == "True" & sa_005_disc_stat <= 0.05)/reps,
            `SA (0.01)` = 100*count(passed_safety_001 == "True" & sa_001_disc_stat <= 0.01)/reps) |>
  pivot_longer(cols = -c(sample_size),
               names_to = "model",
               values_to = "prop_passed_safety_satisfied_cstr")

# join both data sets
conv_disc <- inner_join(sim_converged_long, sim_satisfied_cstr_long,
                        by = c("sample_size", "model")) |>
  mutate(prop_passed_safety = prop_passed_safety - prop_passed_safety_satisfied_cstr) |>
  rename("prop_passed_safety_failed_cstr" = prop_passed_safety) |>
  inner_join(sim_failure_long,
             by = c("sample_size", "model")) |>
  rename("failed safety test" = prop_failed_safety,
         "passed safety test, didn't satisfy constraint" = prop_passed_safety_failed_cstr,
         "passed safety test, satisfied constraint" = prop_passed_safety_satisfied_cstr) |>
  pivot_longer(cols = -c(sample_size, model),
               names_to = "statistic",
               values_to = "value") |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000")))
```

```r
# plot

conv_disc |>
  mutate(model = factor(model,
                        levels = c("SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)"))) |>
  group_by(model, statistic) |>
  summarise(avg = mean(value, na.rm = TRUE)) |>
  ggplot(mapping = aes(x = model, y = avg, fill = statistic)) +
  geom_col() +
  theme_minimal() +
  scale_fill_brewer(palette = "Blues") +
  labs(y = "Percentage",
       x = "Algorithm",
       title = "Probability of Returning a Solution and Satisfying the Constraint",
       fill = " ")
```




```
# plot

conv_disc |>
  mutate(model = factor(model,
                        levels = c("SA (0.2)", "SA (0.1)", "SA (0.05)", "SA (0.01)"))) |>
  group_by(model, statistic, sample_size) |>
  summarise(avg = mean(value, na.rm = TRUE)) |>
  ggplot(mapping = aes(x = model, y = avg, fill = statistic)) +
  geom_col() +
  theme_minimal() +
  scale_fill_brewer(palette = "Blues") +
  facet_wrap(~sample_size) +
  labs(y = "Percentage",
       x = "Algorithm")
```



NSF

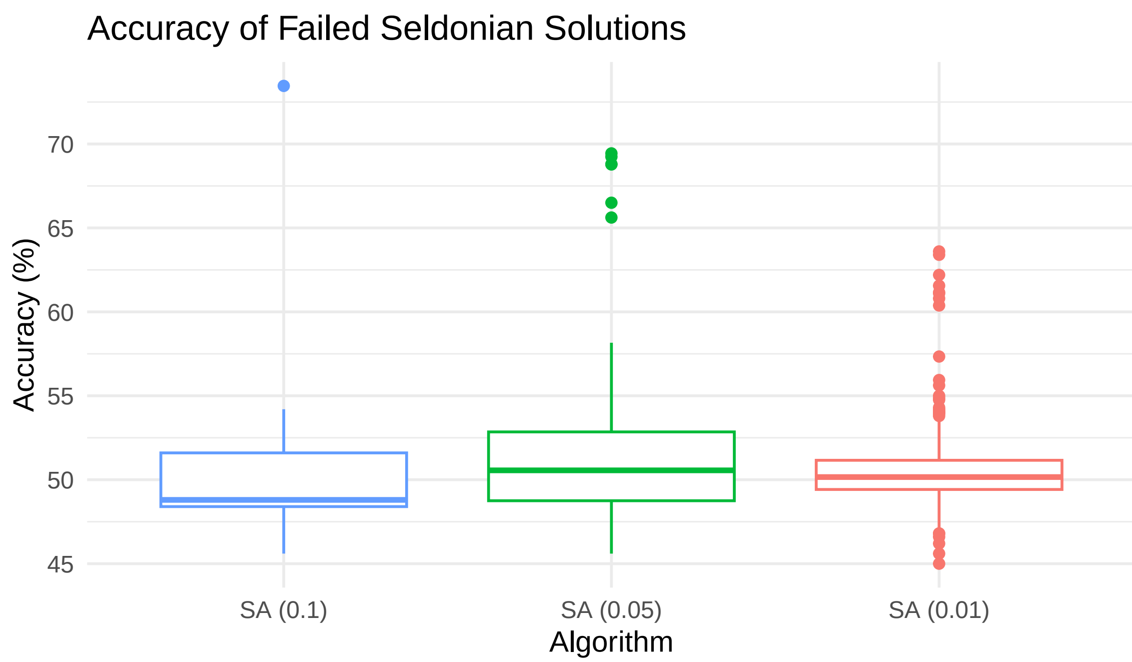
Finally, this section will assess the candidate solutions of the Seldonian algorithm that didn't pass the safety test.

Accuracy

```
# filter accuracy data out for only seldonian solutions that failed the safety test
sim_results_failed_accuracy <- sim_results |>
  mutate(sa_02_accuracy = ifelse(passed_safety_02 == "False", sa_02_accuracy, NA),
         sa_01_accuracy = ifelse(passed_safety_01 == "False", sa_01_accuracy, NA),
         sa_005_accuracy = ifelse(passed_safety_005 == "False", sa_005_accuracy, NA),
         sa_001_accuracy = ifelse(passed_safety_001 == "False", sa_001_accuracy, NA))
```

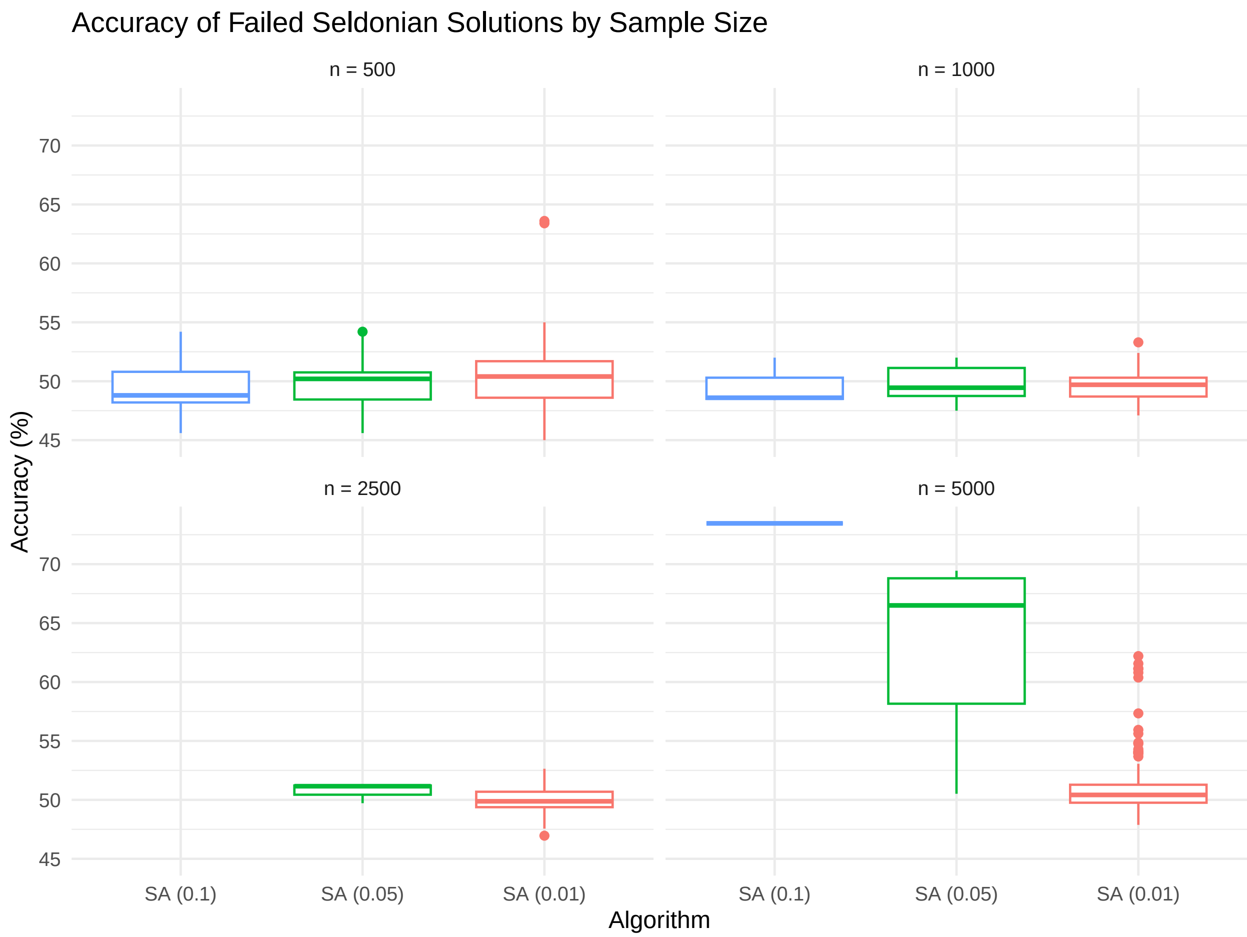
```
# visualization
sim_results_failed_accuracy |>
  dplyr::select(c(sa_02_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size)) |>
  rename("Sample Size" = sample_size,
         "SA (0.2)" = sa_02_accuracy,
         "SA (0.1)" = sa_01_accuracy,
         "SA (0.05)" = sa_005_accuracy,
         "SA (0.01)" = sa_001_accuracy) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "accuracy") |>
```

```r
  mutate(accuracy = 100*accuracy) |>
  ggplot(mapping = aes(x = model, y = accuracy, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  labs(x = "Algorithm",
       y = "Accuracy (%)",
       title = "Accuracy of Failed Seldonian Solutions")
```



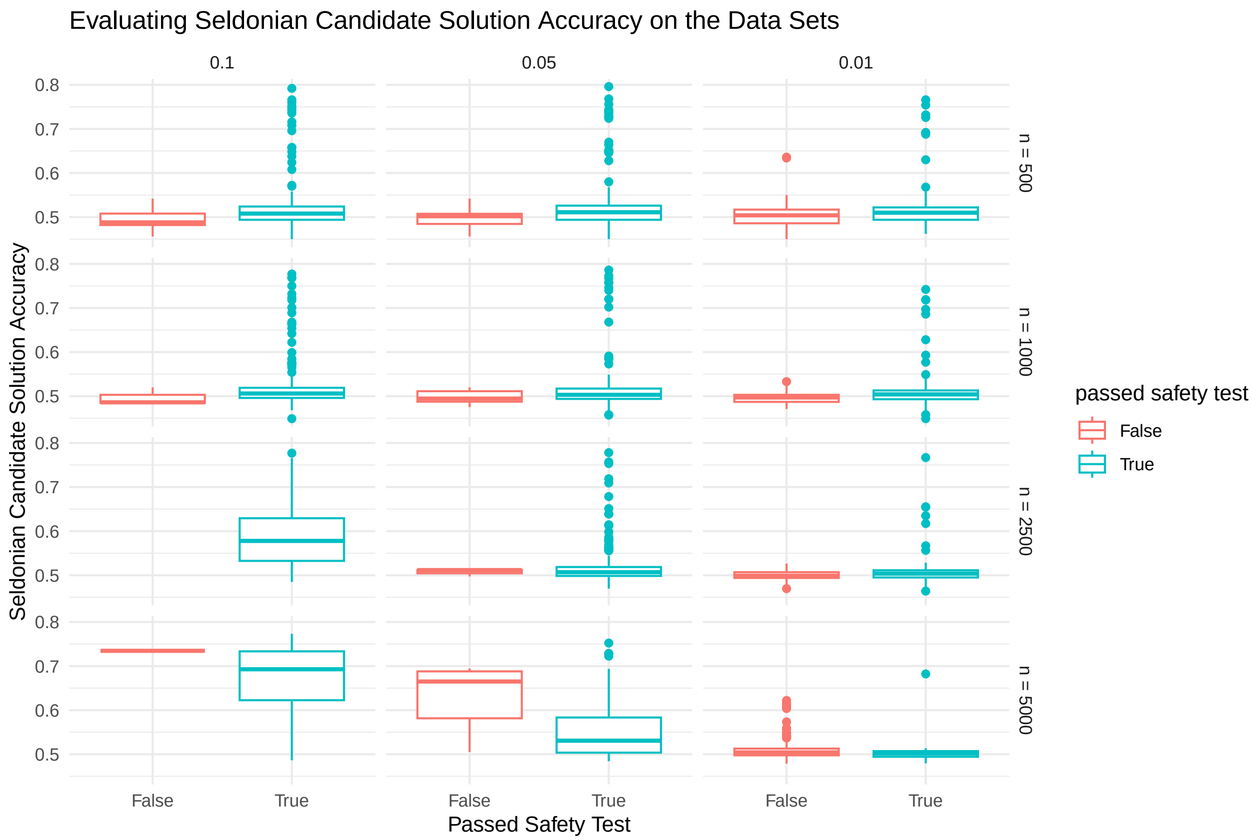

```r
# visualization
sim_results_failed_accuracy |>
  dplyr::select(c(sa_02_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000"))) |>
  rename("Sample Size" = sample_size,
         "SA (0.2)" = sa_02_accuracy,
         "SA (0.1)" = sa_01_accuracy,
         "SA (0.05)" = sa_005_accuracy,
         "SA (0.01)" = sa_001_accuracy) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "accuracy") |>
  mutate(accuracy = 100*accuracy) |>
  ggplot(mapping = aes(x = model, y = accuracy, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
```

```
  guides(color = "none") +
  facet_wrap(~`Sample Size`) +
  labs(x = "Algorithm",
       y = "Accuracy (%)",
       title = "Accuracy of Failed Seldonian Solutions by Sample Size")
```




```
sa_accuracy <- sim_results |>
  dplyr::select(c(lr_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size, dataset_id)) |>
  pivot_longer(cols = -c(dataset_id, sample_size, lr_accuracy),
               names_to = "epsilon",
               values_to = "accuracy") |>
  mutate(epsilon = case_when(epsilon == "sa_01_accuracy" ~ 0.1,
                             epsilon == "sa_005_accuracy" ~ 0.05,
                             epsilon == "sa_001_accuracy" ~ 0.01))

sim_results |>
  dplyr::select(c(lr_accuracy, passed_safety_01, passed_safety_005,
                  passed_safety_001, sample_size, dataset_id)) |>
  pivot_longer(cols = -c(dataset_id, sample_size, lr_accuracy),
               names_to = "epsilon",
               values_to = "passed_safety") |>
  mutate(epsilon = case_when(epsilon == "passed_safety_01" ~ 0.1,
                             epsilon == "passed_safety_005" ~ 0.05,
```

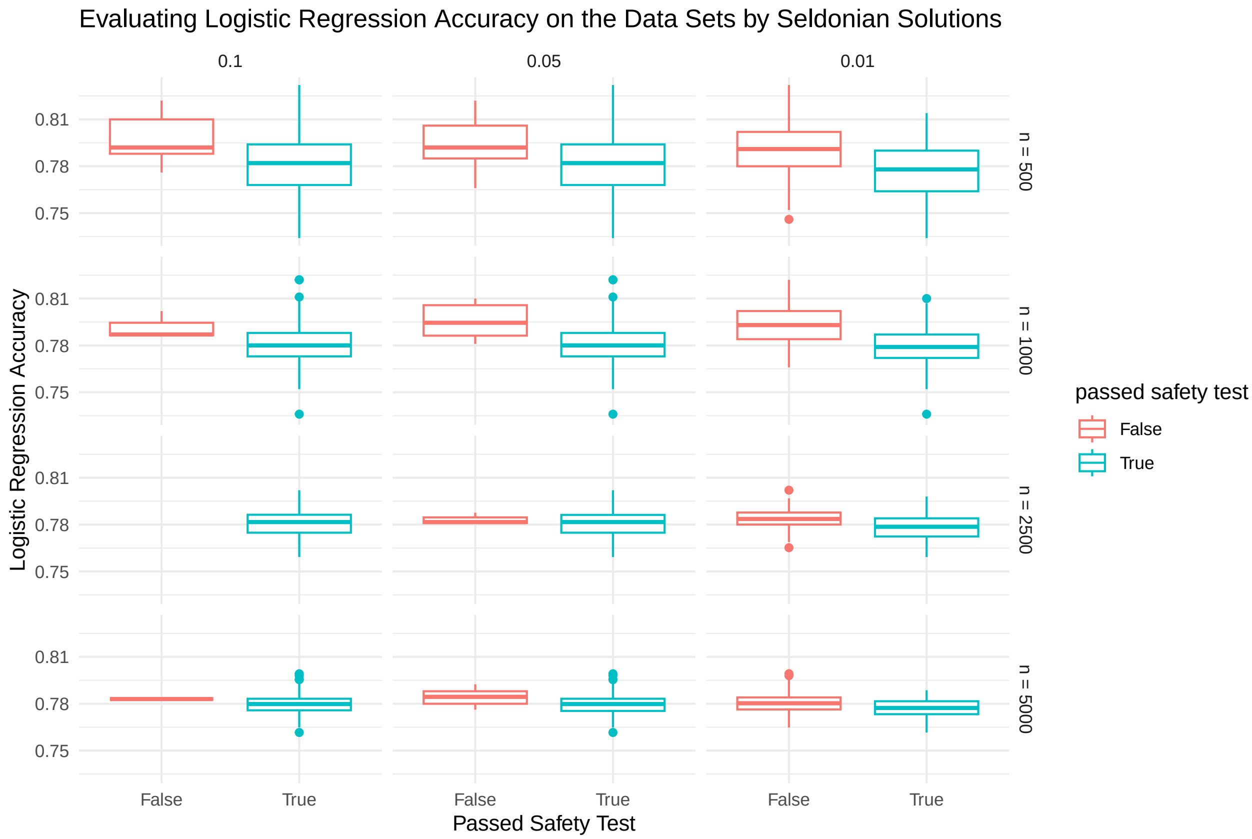
```r
                              epsilon == "passed_safety_001" ~ 0.01)) |>
  inner_join(sa_accuracy, by = c("lr_accuracy", "epsilon", "sample_size", "dataset_id")) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                  sample_size == 1000 ~ "n = 1000",
                                  sample_size == 2500 ~ "n = 2500",
                                  sample_size == 5000 ~ "n = 5000"
                                  ),
                               levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000")),
         epsilon = factor(epsilon, levels = c(0.1, 0.05, 0.01))) |>
  ggplot(mapping = aes(x = passed_safety, y = accuracy, color = passed_safety)) +
  geom_boxplot() +
  theme_minimal() +
  facet_grid(rows = vars(sample_size), cols = vars(epsilon)) +
  labs(x = "Passed Safety Test",
       y = "Seldonian Candidate Solution Accuracy",
       title = "Evaluating Seldonian Candidate Solution Accuracy on the Data Sets",
       color = "passed safety test")
```



Accuracy of Logistic Regression in Failed Seldonian Solutions

```r
sim_results |>
  dplyr::select(c(lr_accuracy, passed_safety_01, passed_safety_005,
                  passed_safety_001, sample_size)) |>
  pivot_longer(cols = -c(sample_size, lr_accuracy),
               names_to = "epsilon",
               values_to = "passed_safety") |>
```

```
  mutate(epsilon = case_when(epsilon == "passed_safety_01" ~ 0.1,
                             epsilon == "passed_safety_005" ~ 0.05,
                             epsilon == "passed_safety_001" ~ 0.01)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                                   levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000")),
         epsilon = factor(epsilon, levels = c(0.1, 0.05, 0.01))) |>
  ggplot(mapping = aes(x = passed_safety, y = lr_accuracy, color = passed_safety)) +
  geom_boxplot() +
  theme_minimal() +
  facet_grid(rows = vars(sample_size), cols = vars(epsilon)) +
  labs(x = "Passed Safety Test",
       y = "Logistic Regression Accuracy",
       title = "Evaluating Logistic Regression Accuracy on the Data Sets by Seldonian Solutions",
       color = "passed safety test")
```

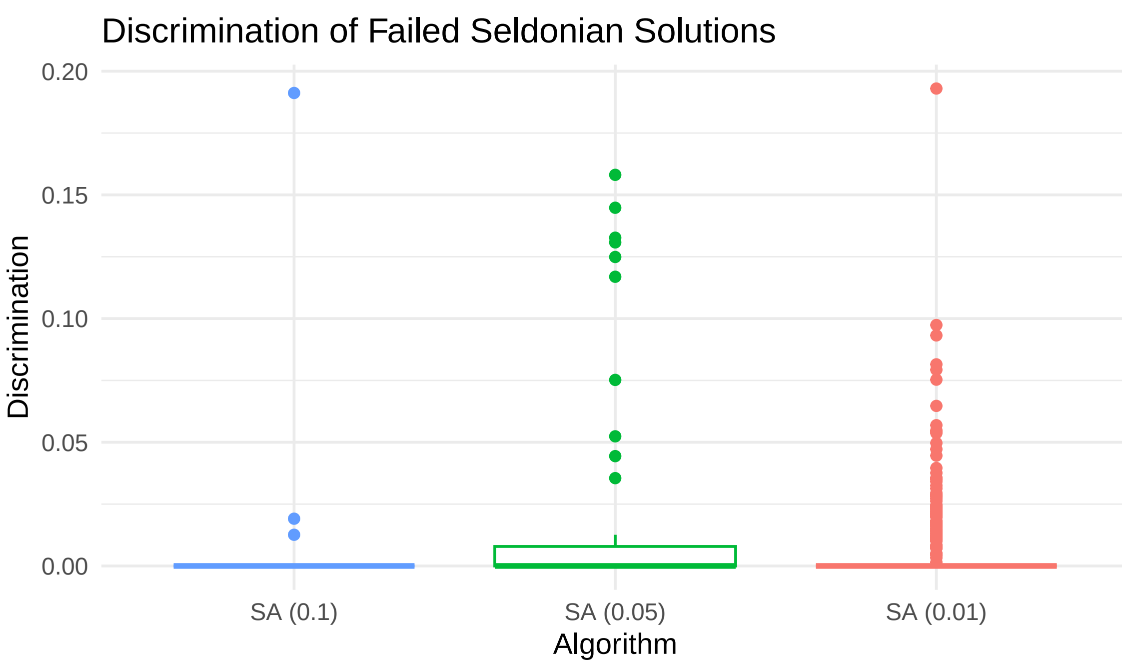


Discrimination

```
# filter accuracy data out for only seldonian solutions that failed the safety test
sim_results_failed_disc <- sim_results |>
  mutate(sa_02_disc_stat = ifelse(passed_safety_02 == "False", sa_02_disc_stat, NA),
         sa_01_disc_stat = ifelse(passed_safety_01 == "False", sa_01_disc_stat, NA),
         sa_005_disc_stat = ifelse(passed_safety_005 == "False", sa_005_disc_stat, NA),
```

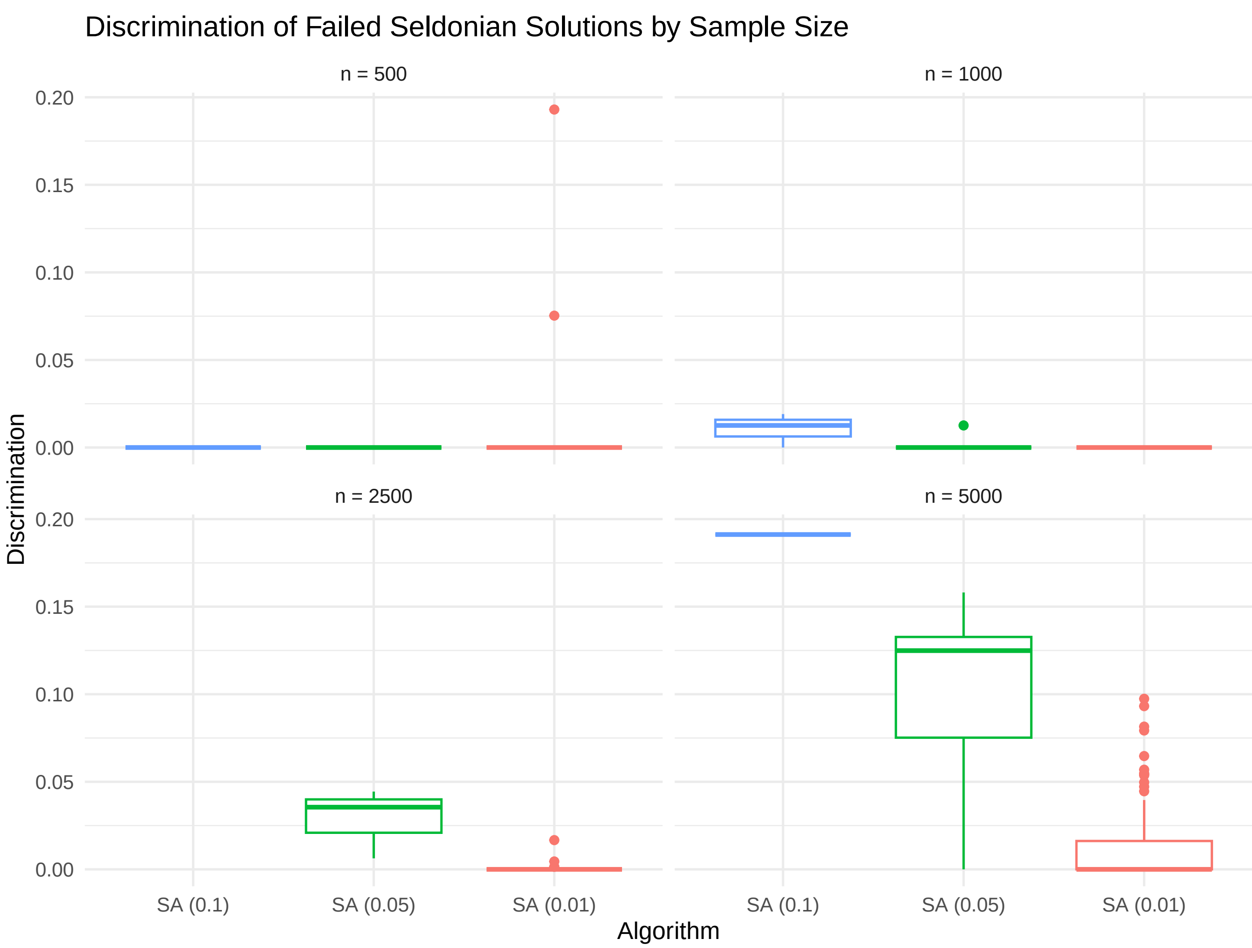
```
         sa_001_disc_stat = ifelse(passed_safety_001 == "False", sa_001_disc_stat, NA))
```

```
# visualization
sim_results_failed_disc |>
  dplyr::select(c(sa_02_disc_stat, sa_01_disc_stat, sa_005_disc_stat,
                  sa_001_disc_stat, sample_size)) |>
  rename("Sample Size" = sample_size,
         "SA (0.2)" = sa_02_disc_stat,
         "SA (0.1)" = sa_01_disc_stat,
         "SA (0.05)" = sa_005_disc_stat,
         "SA (0.01)" = sa_001_disc_stat) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "discrimination") |>
  ggplot(mapping = aes(x = model, y = discrimination, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  labs(x = "Algorithm",
       y = "Discrimination",
       title = "Discrimination of Failed Seldonian Solutions")
```



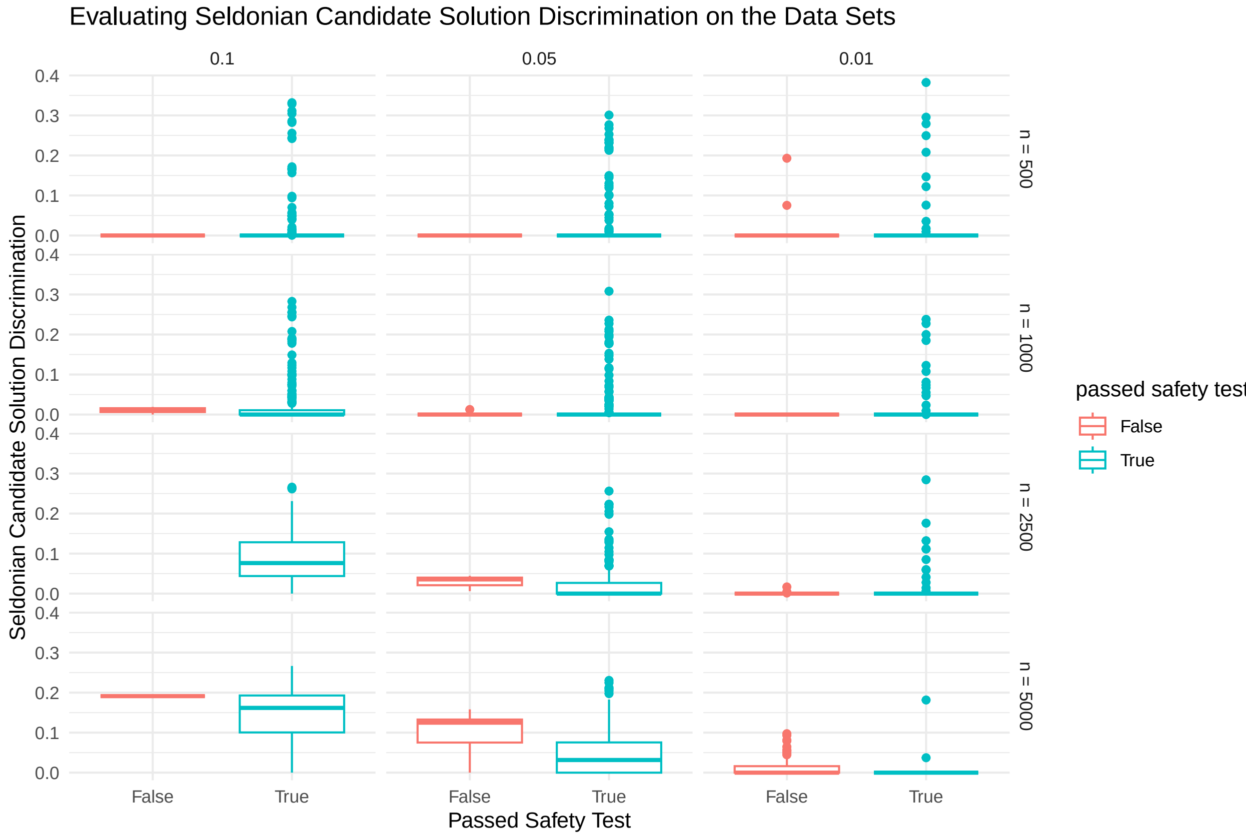

```
# visualization
sim_results_failed_disc |>
  dplyr::select(c(sa_02_disc_stat, sa_01_disc_stat, sa_005_disc_stat,
                  sa_001_disc_stat, sample_size)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000"))) |>
```

```r
  rename("Sample Size" = sample_size,
         "SA (0.2)" = sa_02_disc_stat,
         "SA (0.1)" = sa_01_disc_stat,
         "SA (0.05)" = sa_005_disc_stat,
         "SA (0.01)" = sa_001_disc_stat) |>
  pivot_longer(cols = -c(`Sample Size`),
               names_to = "model",
               values_to = "discrimination") |>
  ggplot(mapping = aes(x = model, y = discrimination, color = model)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("SA (0.1)", "SA (0.05)", "SA (0.01)")) +
  theme_minimal() +
  guides(color = "none") +
  facet_wrap(~`Sample Size`) +
  labs(x = "Algorithm",
       y = "Discrimination",
       title = "Discrimination of Failed Seldonian Solutions by Sample Size")
```



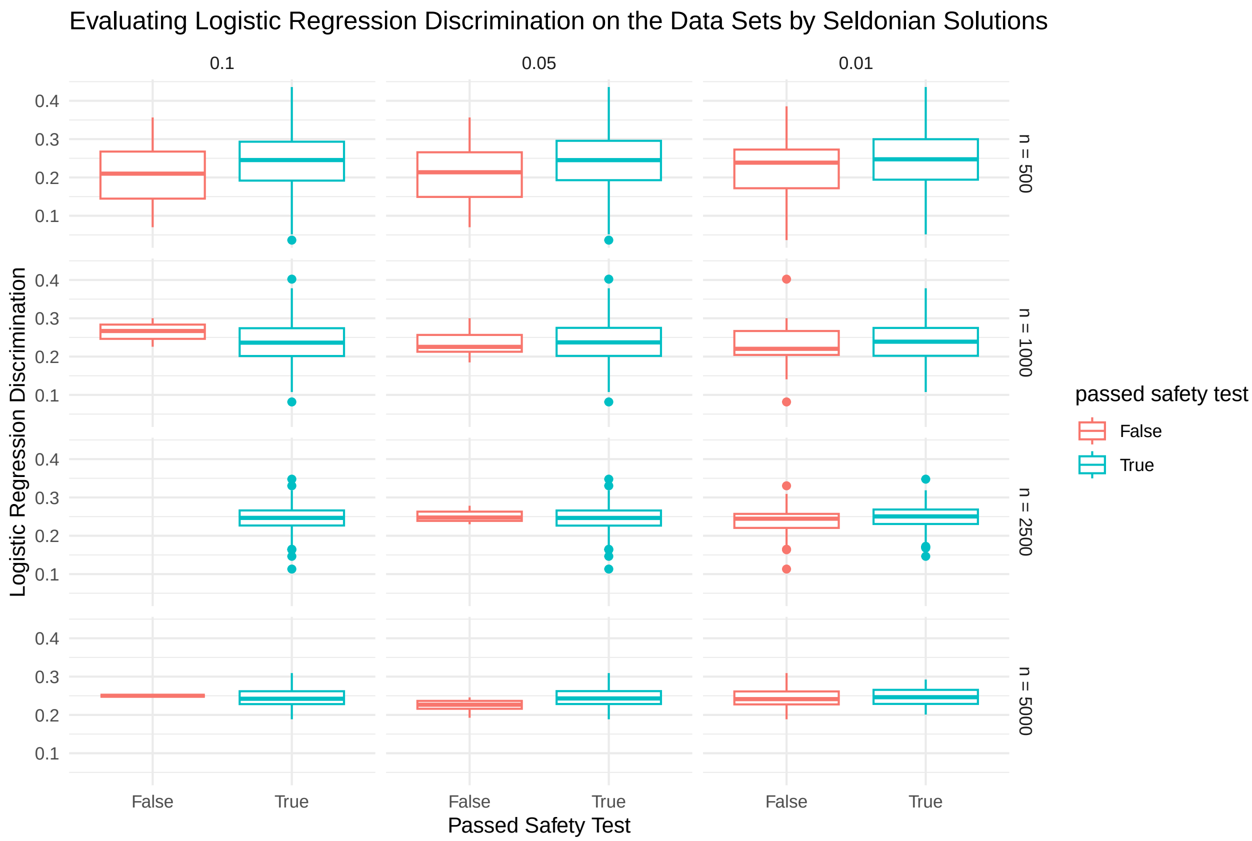

```r
sa_disc_stat <- sim_results |>
  dplyr::select(c(lr_discrimination, sa_01_disc_stat, sa_005_disc_stat,
                  sa_001_disc_stat, sample_size, dataset_id)) |>
  pivot_longer(cols = -c(dataset_id, sample_size, lr_discrimination),
               names_to = "epsilon",
               values_to = "disc_stat") |>
```

```r
  mutate(epsilon = case_when(epsilon == "sa_01_disc_stat" ~ 0.1,
                             epsilon == "sa_005_disc_stat" ~ 0.05,
                             epsilon == "sa_001_disc_stat" ~ 0.01))

sim_results |>
  dplyr::select(c(lr_discrimination, passed_safety_01, passed_safety_005,
                  passed_safety_001, sample_size, dataset_id)) |>
  pivot_longer(cols = -c(dataset_id, sample_size, lr_discrimination),
               names_to = "epsilon",
               values_to = "passed_safety") |>
  mutate(epsilon = case_when(epsilon == "passed_safety_01" ~ 0.1,
                             epsilon == "passed_safety_005" ~ 0.05,
                             epsilon == "passed_safety_001" ~ 0.01)) |>
  inner_join(sa_disc_stat, by = c("lr_discrimination", "epsilon", "sample_size", "dataset_id")) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                 sample_size == 1000 ~ "n = 1000",
                                 sample_size == 2500 ~ "n = 2500",
                                 sample_size == 5000 ~ "n = 5000"
                                 ),
                              levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000")),
         epsilon = factor(epsilon, levels = c(0.1, 0.05, 0.01))) |>
  ggplot(mapping = aes(x = passed_safety, y = disc_stat, color = passed_safety)) +
  geom_boxplot() +
  theme_minimal() +
  facet_grid(rows = vars(sample_size), cols = vars(epsilon)) +
  labs(x = "Passed Safety Test",
       y = "Seldonian Candidate Solution Discrimination",
       title = "Evaluating Seldonian Candidate Solution Discrimination on the Data Sets",
       color = "passed safety test")
```



Discrimination of Logistic Regression in Failed Seldonian Solutions

```
sim_results |>
  dplyr::select(c(lr_discrimination, passed_safety_01, passed_safety_005,
                  passed_safety_001, sample_size)) |>
  pivot_longer(cols = -c(sample_size, lr_discrimination),
               names_to = "epsilon",
               values_to = "passed_safety") |>
  mutate(epsilon = case_when(epsilon == "passed_safety_01" ~ 0.1,
                             epsilon == "passed_safety_005" ~ 0.05,
                             epsilon == "passed_safety_001" ~ 0.01)) |>
  mutate(sample_size = factor(case_when(sample_size == 500 ~ "n = 500",
                                        sample_size == 1000 ~ "n = 1000",
                                        sample_size == 2500 ~ "n = 2500",
                                        sample_size == 5000 ~ "n = 5000"
                                        ),
                                      levels = c("n = 500", "n = 1000", "n = 2500", "n = 5000")),
         epsilon = factor(epsilon, levels = c(0.1, 0.05, 0.01))) |>
  ggplot(mapping = aes(x = passed_safety, y = lr_discrimination, color = passed_safety)) +
  geom_boxplot() +
  theme_minimal() +
  facet_grid(rows = vars(sample_size), cols = vars(epsilon)) +
  labs(x = "Passed Safety Test",
       y = "Logistic Regression Discrimination",
       title = "Evaluating Logistic Regression Discrimination on the Data Sets by Seldonian Solutions",
       color = "passed safety test")
```


Evaluating Logistic Regression Discrimination on the Data Sets by Seldonian Solutions
0.1
0.05
0.01
n = 500
n = 1000
n = 2500
n = 5000
0.4
0.3
0.2
0.1
Logistic Regression Discrimination
False
True
Passed Safety Test
passed safety test
False
True