Thesis Simulation Results Analysis for Chapter 4

Dasha Asienga

2024-04-03

Contents

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 200 observations for each of the simulation trials, 50 from each sample size: $n = 500, 1000, 2500, 5000$.

```
lr_500 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/Res
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
lr_5000 <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/R/Simulation/LogisticRegression/R
```

```
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: 400
## 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 200 observations for each of the simulation trials, 50 from each sample size: $n = 500, 1000, 2500, 5000$.

```
#seldonian_results <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/Python/COMPAS Simulati
seldonian_results <- read.csv("/home/dasienga24/Statistics-Senior-Honors-Thesis/Data Sets/seldonian_sim
seldonian_results <- distinct(seldonian_results) #remove duplicate rows
glimpse(seldonian_results)
```

```
## Rows: 200
## Columns: 14
## $ sample_size       <int> 1000, 1000, 1000, 2500, 1000, 1000, 1000, 2500, 500,~
## $ dataset_id        <int> 25, 10, 22, 17, 34, 38, 36, 2, 49, 17, 7, 41, 8, 33,~
## $ 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> "True", "True", "True", "False", "True", "True", "Tr~
## $ sa_02_accuracy    <dbl> 0.6420, 0.6410, 0.6370, 0.7832, 0.5520, 0.6180, 0.73~
## $ sa_01_accuracy    <dbl> 0.5560, 0.5030, 0.5200, 0.4844, 0.4930, 0.5190, 0.49~
## $ sa_005_accuracy   <dbl> 0.5330, 0.5030, 0.5460, 0.4844, 0.4930, 0.5200, 0.49~
## $ sa_001_accuracy   <dbl> 0.5430, 0.5030, 0.5830, 0.4844, 0.4930, 0.5190, 0.49~
## $ sa_02_disc_stat   <dbl> 0.1791, 0.0995, 0.0948, 0.1428, 0.0787, 0.1081, 0.22~
## $ sa_01_disc_stat   <dbl> 0.0345, 0.0000, 0.0355, 0.0000, 0.0000, 0.0000, NA, ~
## $ sa_005_disc_stat  <dbl> 0.0184, 0.0000, 0.0496, 0.0000, 0.0000, NA, 0.0000, ~
## $ sa_001_disc_stat  <dbl> 0.0252, 0.0000, 0.0347, 0.0000, 0.0000, 0.0000, 0.00~
```

Combining the Data Sets from Both Simulations

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

glimpse(sim_results)
```

```
## Rows: 200
## 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.

```
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	100	90	86	70

Sample Size	LR	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
1000	100	100	100	100	88
2500	100	100	100	100	52
5000	100	100	100	96	28

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")
```

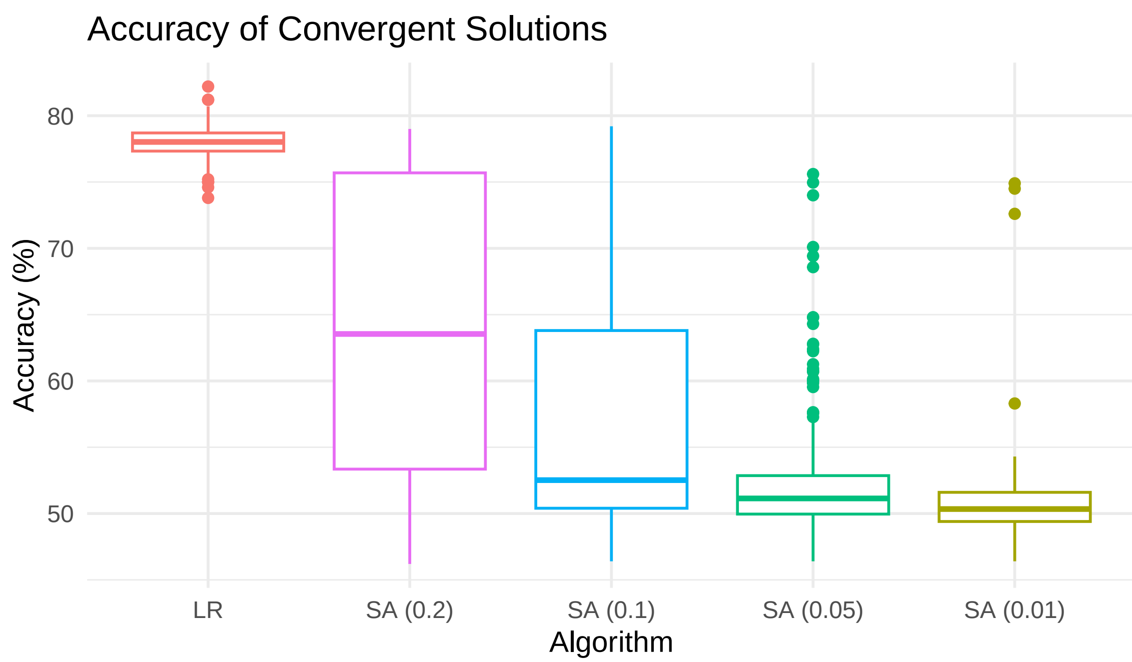
Table 2: Accuracy of Convergent Solutions

Sample Size	LR	sd	SA (0.2)	sd	SA (0.1)	sd	SA (0.05)	sd	SA (0.01)	sd
500	78.22	0.02	55.26	0.07	52.32	0.07	52.27	0.06	51.09	0.04
1000	77.85	0.02	58.21	0.08	52.24	0.05	51.55	0.04	51.54	0.05
2500	77.93	0.01	71.71	0.07	57.24	0.06	51.08	0.02	50.44	0.01
5000	77.95	0.00	71.12	0.07	65.31	0.09	55.71	0.06	50.04	0.01

Visualizations

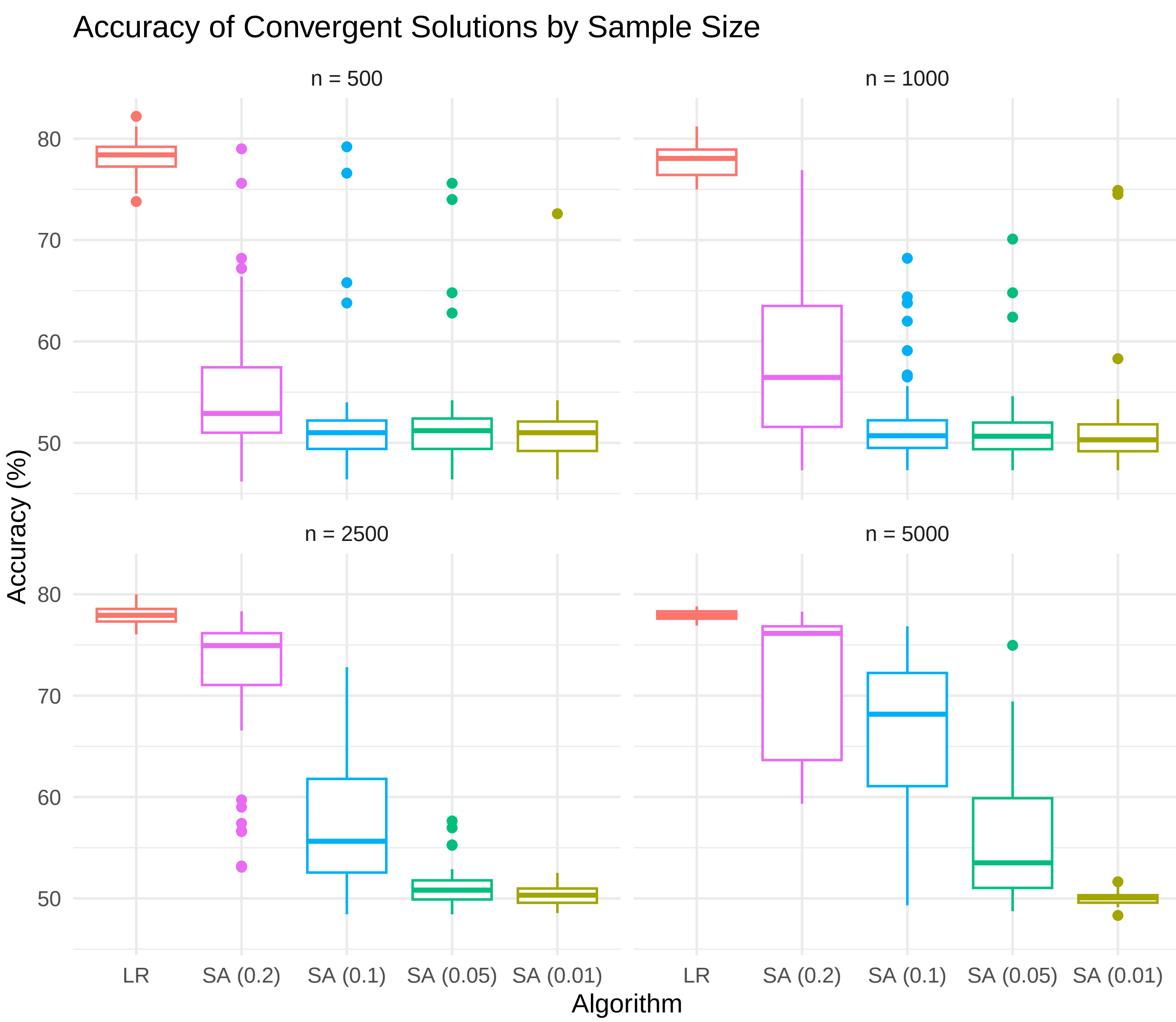
```
# visualization
sim_results_converged_accuracy |>
```

```
  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")
```




```
# 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,
```

```
         "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

```
# 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

```
# 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	0.08	0.05	0.07	0.02	0.07	0.02	0.06	0.01	0.03
1000	0.24	0.05	0.08	0.07	0.02	0.05	0.01	0.04	0.01	0.04
2500	0.24	0.03	0.16	0.07	0.07	0.06	0.01	0.02	0.00	0.00
5000	0.25	0.02	0.14	0.06	0.13	0.08	0.06	0.06	0.00	0.00

```
# 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	84	82.22	90.70	91.43
1000	88	84.00	84.00	90.91
2500	70	70.00	82.00	96.15
5000	82	30.00	52.08	100.00

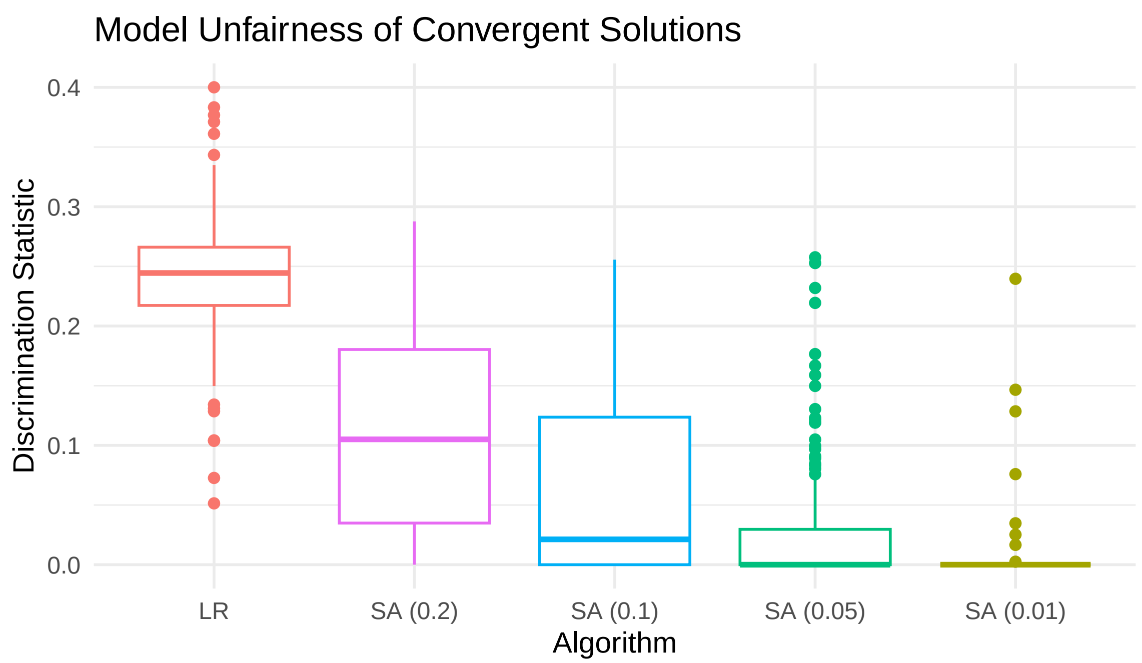
```
# what % returned a solution and also met the constraint
sim_results |>
  group_by(sample_size) |>
  summarise(`SA (0.2)` = 100*count(passed_safety_02 == "True" & sa_02_disc_stat <= 0.2)/reps,
            `SA (0.1)` = 100*count(passed_safety_01 == "True" & sa_01_disc_stat <= 0.1)/reps,
            `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

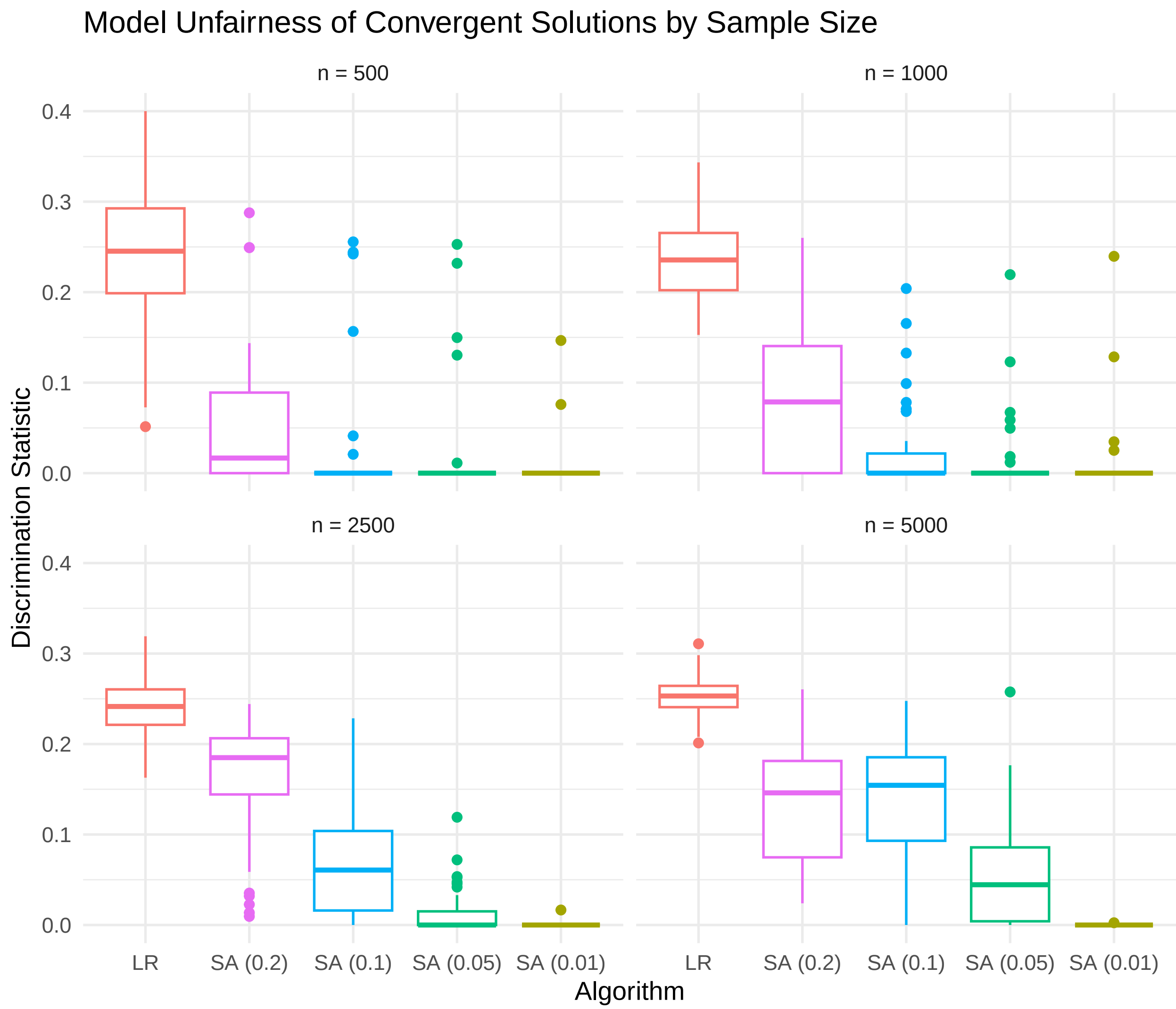
Sample Size	SA (0.2)	SA (0.1)	SA (0.05)	SA (0.01)
500	84	74	78	64
1000	88	84	84	80
2500	70	70	82	50
5000	82	30	50	28

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.

```r
# 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,
```

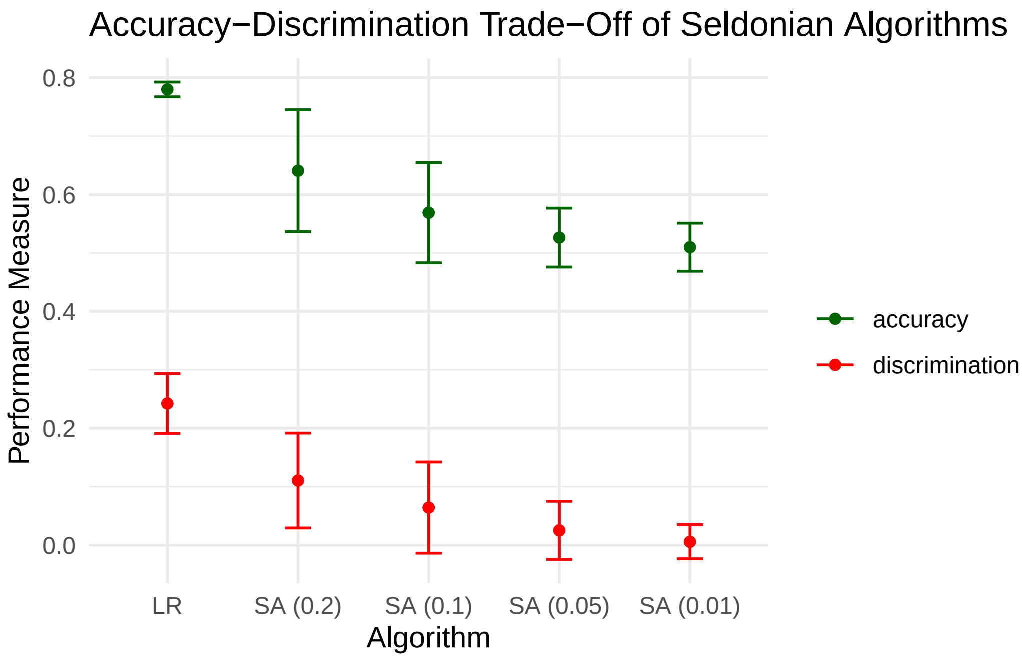
```r
         "SA (0.01)" = sa_001_disc_stat) |>
  pivot_longer(cols = -c(`Sample Size`, dataset_id),
               names_to = "model",
               values_to = "discrimination")

# 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")
```

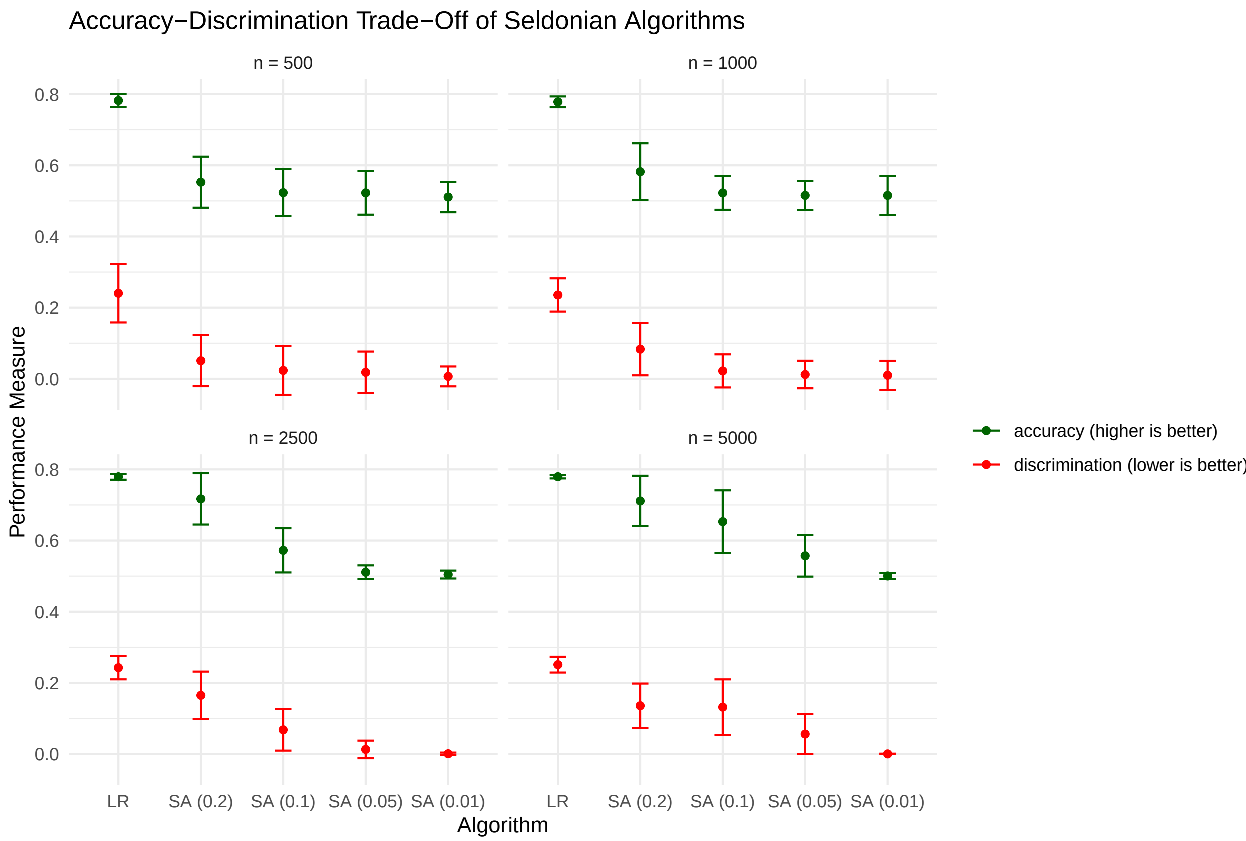
```r
# plot error-bars
color <- c("darkgreen", "red")

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 = " ")
```




```
# plot error-bars by sample size
color <- c("darkgreen", "red")

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. I need to brainstorm better ways of visualizing this relationship.

```
# 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",
```

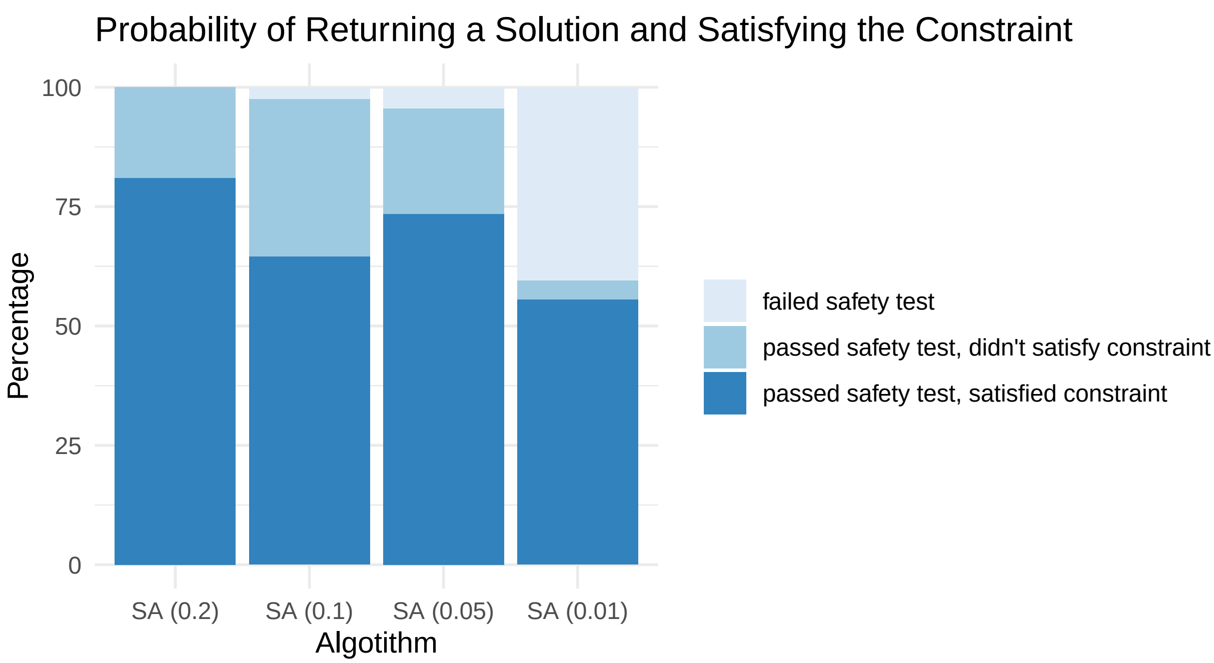
```r
               values_to = "prop_passed_safety")

# 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")))
```

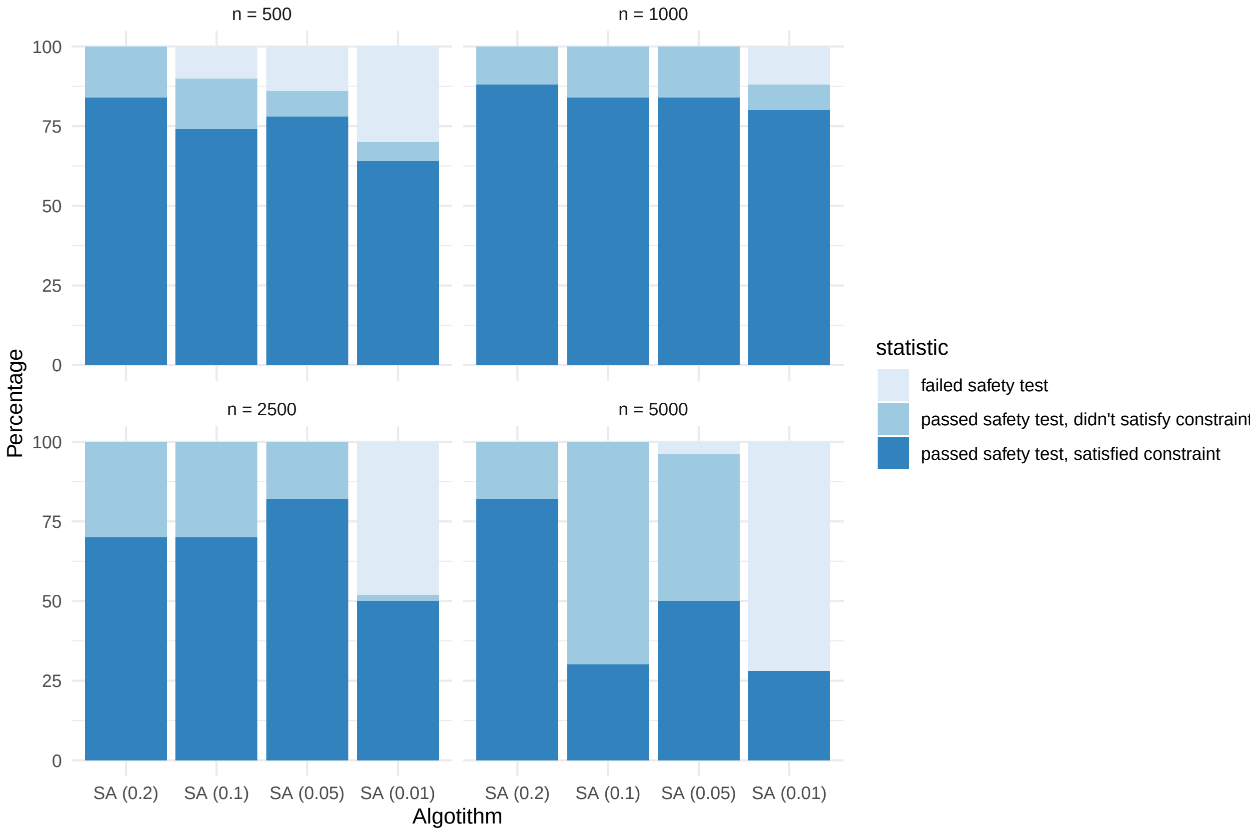
```r
# plot

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 = "Algotithm",
       title = "Probability of Returning a Solution and Satisfying the Constraint",
       fill = " ")
```




```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, 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 = "Algotithm")
```



NSF

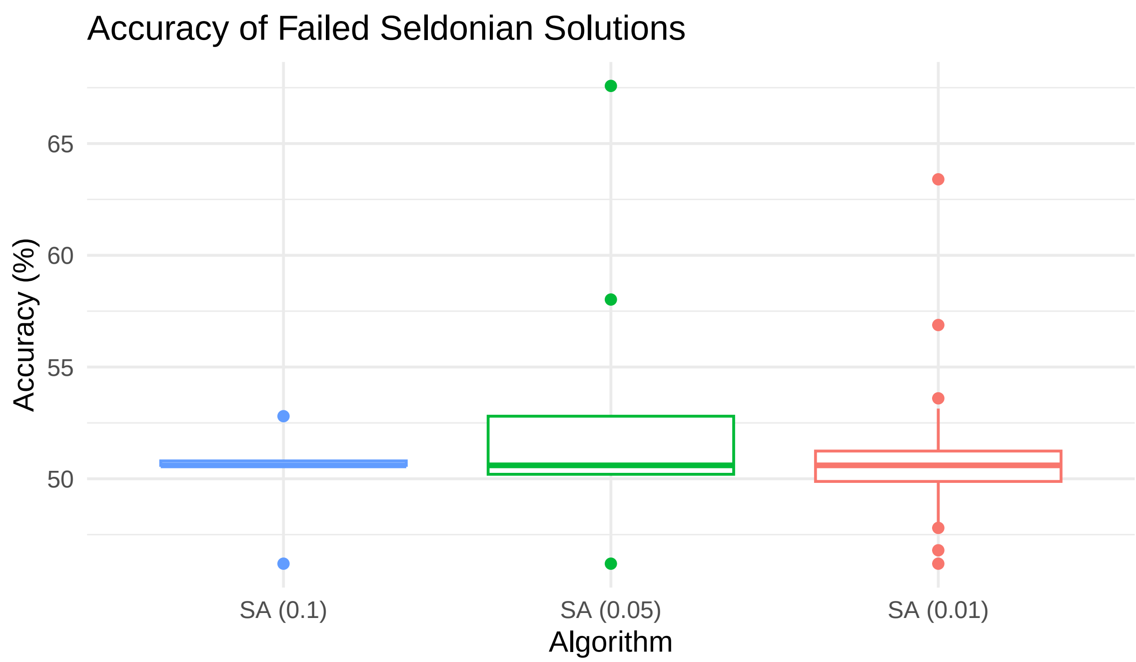
Finally, this section will assess the candidate solutions of the Seldonian algorithm that didn't pass the safety test.

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))
```

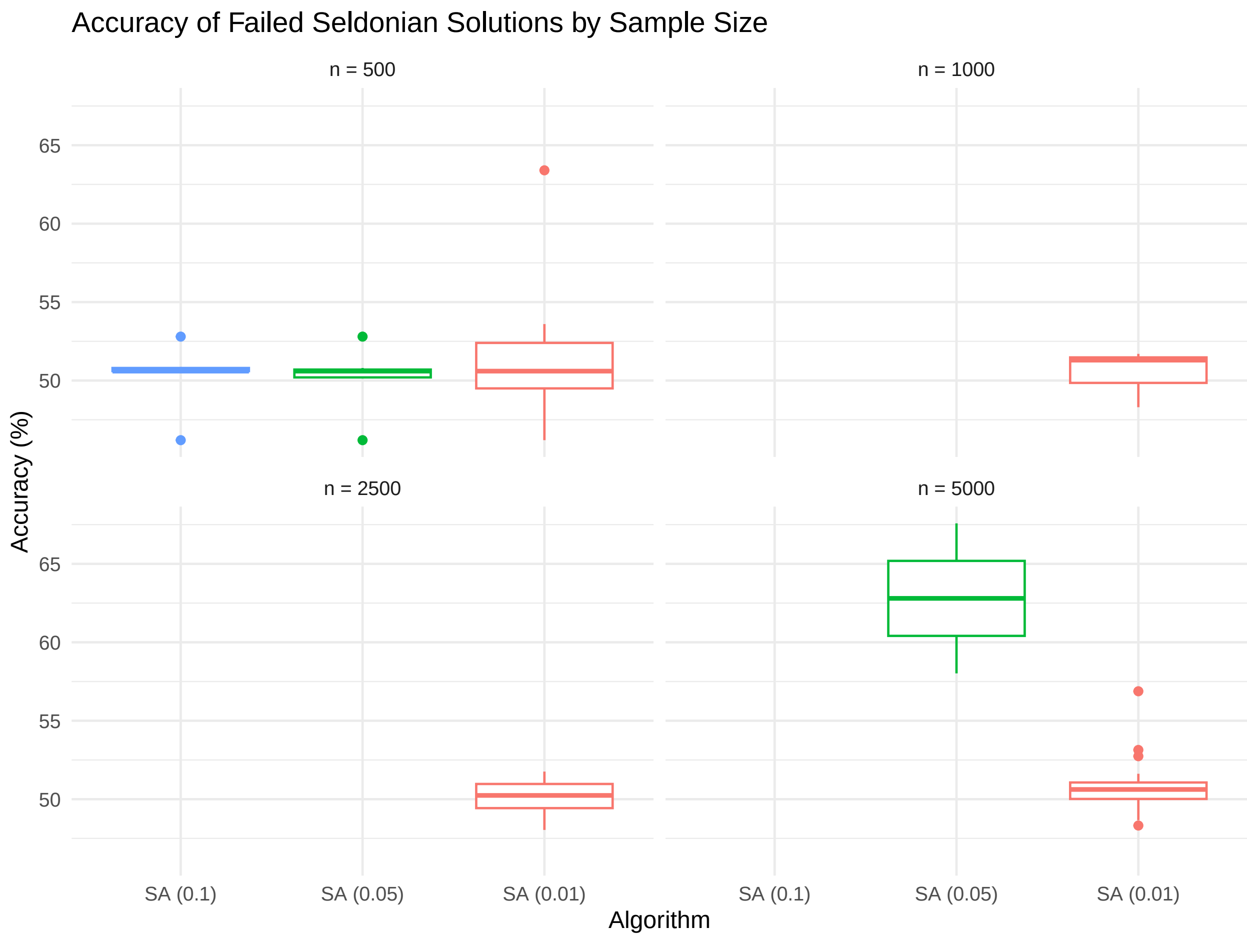
```
# visualization
sim_results_failed_accuracy |>
  dplyr::select(c(sa_02_accuracy, sa_01_accuracy, sa_005_accuracy,
                  sa_001_accuracy, sample_size)) |>
  rename("Sample Size" = sample_size,
         "SA (0.2)" = sa_02_accuracy,
         "SA (0.1)" = sa_01_accuracy,
         "SA (0.05)"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

```
  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")
```




```
# 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() +
```

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

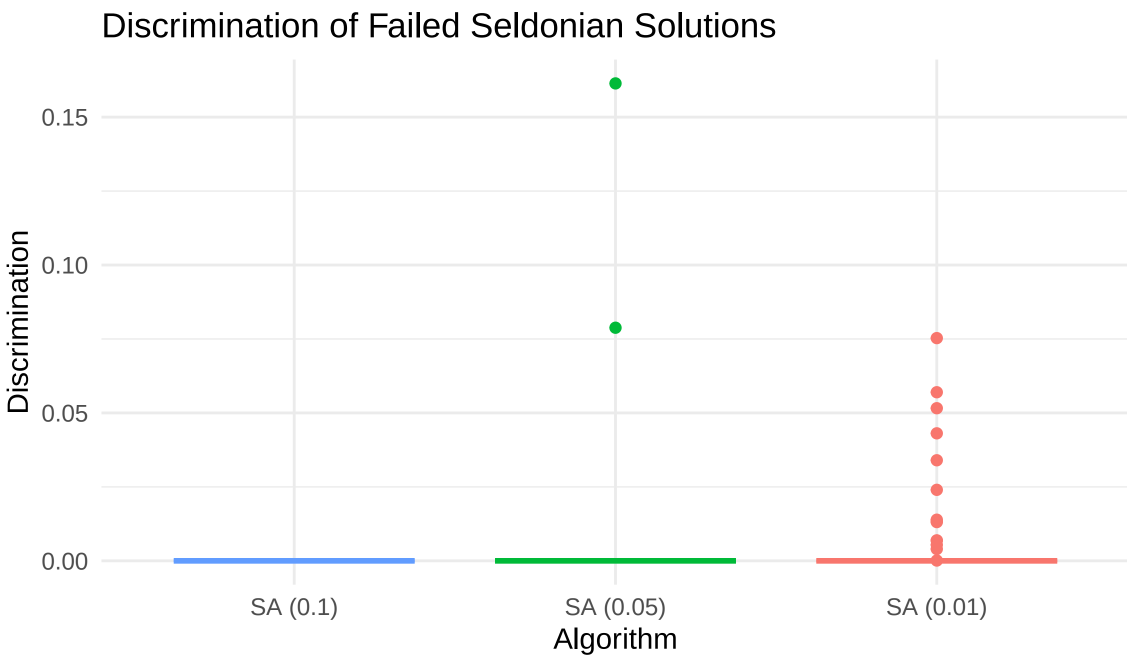


Discrimination

```r
# 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))
```

```r
# 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) |>
```

```r
  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")
```




```r
# 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"))) |>
  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)")) +
```

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