

# Power Analysis

*Jeremy Albright*

*August 10, 2018*

## Motivation

This document summarizes the statistical power to detect a change of .25 standard deviations in an outcome variable between a start date and end date, varying the number of observations in between. The analysis will use Monte Carlo methods to generate 10,000 fake individuals, each showing the same .25 standard deviation change between the start and end dates, but with random Gaussian noise added to make the subjects unique. The percentage of samples with a statistically significant slope in a linear regression will be recorded. This exercise will be repeated while increasing the number of measurements between the start and end date will from 3 to 10.

The trend is exactly the same for each simulated subject except for the random noise. Thus, the regression models are all attempting to pick up the same signal. As will be shown, the probability of finding a statistically significant signal will increase with more data points.

## Functions

The following functions define the Monte Carlo analysis. First, a function is created to pull the p-value for the slope from the regression of y on x.

```
get_p <- function(x, y){  
  
  lm(y ~ x) %>%  
  tidy %>%  
  filter(term == "x") %>%  
  dplyr::select(p.value) %>%  
  unlist %>%  
  as.numeric  
  
}
```

The next function iterates through `reps` random samples, given `k` measurements over time.

```
get_power <- function(k, reps){  
  
  x <- 1:k  
  
  ps <- sapply(1:reps, function(i){  
  
    y <- .25*max(k)*x + rnorm(k)  
    get_p(x, y)  
  
  })  
  
  tibble(`Time Points` = k,  
         `Proportion Significant` = sum(ps < .05)/reps)  
  
}
```

Running these functions, we can build our dataset of true positives.

```
set.seed(12345)

plan(multisession)

tbl <- future_map_dfr(3:10, ~get_power(.x, reps = 10000))

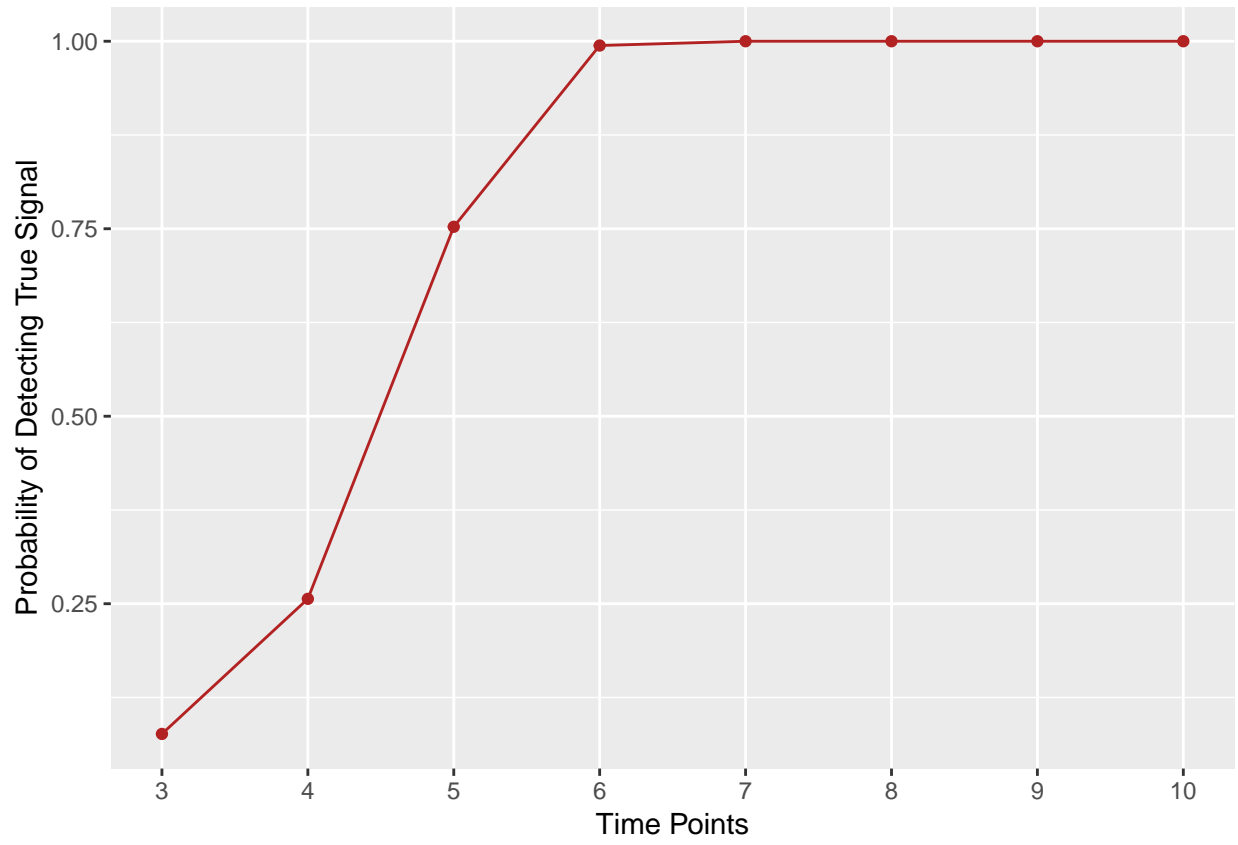
tbl %>%
  knitr::kable()
```

Time Points	Proportion Significant
3	0.0764
4	0.2565
5	0.7526
6	0.9943
7	1.0000
8	1.0000
9	1.0000
10	1.0000

## Figure

And the plot.

```
tbl %>%
  ggplot(aes(x = `Time Points`, y = `Proportion Significant`)) +
  geom_point(color = "firebrick") + geom_line(color = "firebrick") +
  labs(y = "Probability of Detecting True Signal") +
  scale_x_continuous("Time Points", 3:10, 3:10)
```



```
ggsave("power-plot.png")
```