Power Analysis for Honesty Experiment

```r
suppressMessages({
  library(tidyverse)
  library(fixest)
  library(pwr)
})

set.seed(42)

hrounds <- read_csv(
  "../data/generated/honesty_rounds.csv",
  show_col_types = FALSE
) %>%
  mutate(
    experiment = factor(ifelse(
      experiment == "fhonesty",
      "Business Framing", "Neutral Framing"
    ), c("Neutral Framing", "Business Framing")),
    slack = reported_amount - true_amount,
    pct_slack_claimed = (reported_amount - true_amount)/(6000 - true_amount)
  )

hpart <- hrounds %>%
  group_by(experiment, session_code, player_id) %>%
  summarise(
    pct_slack_claimed = sum(reported_amount - true_amount)/sum(6000 - true_amount),
    .groups = "drop"
  )
```

Descriptive statistics of pretest data to standardize the power tests

```
pct_true <- mean(hrounds$pct_slack_claimed == 0, na.rm = TRUE)
sprintf("%% Truthful reporting: %.2f %%", 100*pct_true)
```

```
[1] "% Truthful reporting: 12.92 %"
```

```
mn_pct_slack_claimed <- mean(hrounds$pct_slack_claimed, na.rm = T)
sd_pct_slack_claimed <- sd(hrounds$pct_slack_claimed, na.rm = T)
mn_pct_slack_claimed_lying <- mean(
  hrounds$pct_slack_claimed[hrounds$pct_slack_claimed > 0], na.rm = T
)
mn_pct_slack_claimed_lying_start <- mean(
  hrounds$pct_slack_claimed[hrounds$pct_slack_claimed > 0 & hrounds$round == 1],
  na.rm = T
)
sd_pct_slack_claimed_lying <- sd(
  hrounds$pct_slack_claimed[hrounds$pct_slack_claimed > 0], na.rm = T
)
sprintf(
  "Mean %% Slack Claimed (SD): %.2f %% (%.2f %%)",
  100*mn_pct_slack_claimed, 100*sd_pct_slack_claimed
)
```

```
[1] "Mean % Slack Claimed (SD): 51.73 % (41.50 %)"
```

```
# Some pretest regressions to see how rounds affect our DV:
table(hrounds$pct_slack_claimed == 0, hrounds$experiment)
```

```
       Neutral Framing Business Framing
FALSE              908              784
TRUE                68              183
```

```
smp_lying <- hrounds[hrounds$pct_slack_claimed > 0,]
ols_rounds <- lm(pct_slack_claimed ~ round, data = smp_lying)
summary(ols_rounds)
```

```
Call:
lm(formula = pct_slack_claimed ~ round, data = smp_lying)

Residuals:
     Min       1Q   Median       3Q      Max
-0.58539 -0.39637 -0.02997  0.40363  0.42462

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.571176   0.020308  28.126   <2e-16 ***
round       0.004199   0.003297   1.274    0.203
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3901 on 1690 degrees of freedom
  (57 observations deleted due to missingness)
Multiple R-squared:  0.0009588, Adjusted R-squared:  0.0003677
F-statistic: 1.622 on 1 and 1690 DF,  p-value: 0.203
```

```r
summary(lm(pct_slack_claimed ~ round*experiment, data = smp_lying))
```

```
Call:
lm(formula = pct_slack_claimed ~ round * experiment, data = smp_lying)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7151 -0.3486 -0.0196  0.3122  0.5435

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       0.648175   0.026453  24.503  < 2e-16 ***
round                             0.009916   0.004290   2.312   0.0209 *
experimentBusiness Framing       -0.164818   0.038841  -4.243 2.32e-05 ***
round:experimentBusiness Framing -0.012607   0.006308  -1.999   0.0458 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3721 on 1688 degrees of freedom
  (57 observations deleted due to missingness)
```

```
Multiple R-squared:  0.09219,   Adjusted R-squared:  0.09058
F-statistic: 57.14 on 3 and 1688 DF,  p-value: < 2.2e-16
```

```r
fe_slack_claimed <- feols(
  pct_slack_claimed ~ experiment | round,
  cluster = c("round", "session_code^player_id"),
  data = smp_lying
)
fe_slack_claimed
```

```
OLS estimation, Dep. Var.: pct_slack_claimed
Observations: 1,692
Fixed-effects: round: 10
Standard-errors: Clustered (round & session_code^player_id)
                            Estimate Std. Error t value  Pr(>|t|)
experimentBusiness Framing -0.233415   0.049759 -4.6909 0.0011346 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RMSE: 0.371756     Adj. R2: 0.086128
                 Within R2: 0.089268
```

```r
fixef(fe_slack_claimed)$round
```

```
        1         2         3         4         5         6         7         8
0.6572878 0.6986086 0.6984437 0.7205994 0.7113418 0.7009629 0.6911087 0.7155882
        9        10
0.7013992 0.7299945
```

Equation-based Power Analysis based on Pretest Data

Round Based Analysis

```r
es <- ES.h(pct_true, pct_true - 0.05)
pwr.2p.test(h = es, n = 1000)
```

```
     Difference of proportion power calculation for binomial distribution (arcsine transforma
```

```
              h = 0.1647998
              n = 1000
      sig.level = 0.05
          power = 0.9577427
    alternative = two.sided

NOTE: same sample sizes
```

```
pr <- pwr.2p.test(n = 1000, power = 0.8)
phi_baseline_true_rate = 2*asin(sqrt(pct_true))
phi_treated_true_rate = phi_baseline_true_rate - pr$h
treatm_true_rate <- sin(phi_treated_true_rate/2)^2
mde <- treatm_true_rate - pct_true
sprintf("MDE Truthful Reporting: %.2f PP", 100*mde)
```

```
[1] "MDE Truthful Reporting: -3.90 PP"
```

```
ssize_lying = round((1 - pct_true)*1000)
pwr.t.test(
  ssize_lying, (0.1*mn_pct_slack_claimed_lying)/sd_pct_slack_claimed_lying
)
```

```
     Two-sample t test power calculation

              n = 871
              d = 0.1522609
      sig.level = 0.05
          power = 0.8879616
    alternative = two.sided

NOTE: n is number in *each* group
```

```
pwr.t.test(
  d = (0.1*mn_pct_slack_claimed_lying)/sd_pct_slack_claimed_lying, power = 0.8
)
```

```
     Two-sample t test power calculation

              n = 678.073
              d = 0.1522609
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

```
pr <- pwr.t.test(n = ssize_lying, power = 0.8)
sprintf("MDE PP Slack Claimed: %.2f", 100*pr$d * sd_pct_slack_claimed_lying)
```

```
[1] "MDE PP Slack Claimed: 5.24"
```

Participant-based Analyis

```
mn_pct_slack_claimed_part <- mean(hpart$pct_slack_claimed)
sd_pct_slack_claimed_part <- sd(hpart$pct_slack_claimed)
sprintf(
  "Mean %% of total part slack claimed (SD): %.2f %% (%.2f %%)",
  100*mn_pct_slack_claimed_part, 100*sd_pct_slack_claimed_part)
```

```
[1] "Mean % of total part slack claimed (SD): NA % (NA %)"
```

```
pwr.t.test(100, (0.1*mn_pct_slack_claimed_part)/sd_pct_slack_claimed_part)
```

```
     Two-sample t test power calculation

              n = 100
              d = NA
      sig.level = 0.05
          power = NA
    alternative = two.sided

NOTE: n is number in *each* group
```

```
pr <- pwr.t.test(n = 100, power = 0.8)
sprintf(
  "MDE Part Slack claimed: %.2f %% (%.1f %% of mean)",
  100*pr$d * sd_pct_slack_claimed_part,
  100*(pr$d * sd_pct_slack_claimed_part)/mn_pct_slack_claimed_part
)
```

```
[1] "MDE Part Slack claimed: NA % (NA % of mean)"
```

Simulation for regression based tests

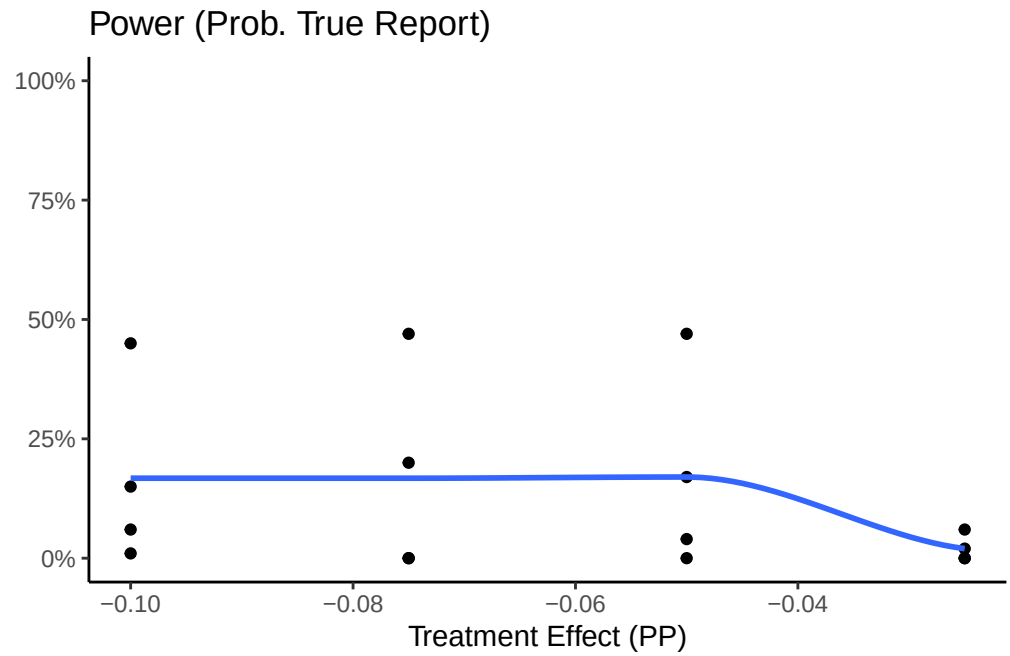
```
if (file.exists("../data/generated/honesty_sim_results.csv")) {
  honesty_sim_results <- read_csv(
    "../data/generated/honesty_sim_results.csv", show_col_types = F
    )
} else {
  ta <- read_csv(
    "../data/generated/honesty_true_amounts.csv", show_col_types = F
  )
  sim_data <- function(parms, runs = 100, rounds = 10) {
    cl <- function(val, vmin = 0, vmax = 1) {
      if (val > vmax) return(vmax)
      if (val < vmin) return(vmin)
      val
    }
    mb <- function(x) {
      steps <- seq(4000, 6000, by = 50)
      steps[which.min(abs(steps - x))][1]
    }
    tr <- function(rd, exp, p, parms) {
      prob_true = ifelse(
        exp == "fhonesty",
        parms$prob_true + parms$prob_true_teffect,
        parms$prob_true
      )
      pct_slack_claimed_lying_start = ifelse(
        exp == "fhonesty",
        parms$pct_slack_claimed_lying_start +
          parms$pct_slack_claimed_lying_start_teffect,
```

```
      parms$pct_slack_claimed_lying_start
    )
    pct_slack_claimed_lying_grate = ifelse(
      exp == "fhonesty",
      parms$pct_slack_claimed_lying_grate +
        parms$pct_slack_claimed_lying_grate_teffect,
      parms$pct_slack_claimed_lying_grate
    )

    tibble(
      experiment = factor(ifelse(
        exp == "fhonesty",
        "Business Framing", "Neutral Framing"
      ), c("Neutral Framing", "Business Framing")),
      player_id = p,
      round = rd,
      true_amount = ta$true_amount[ta$part == paste0("P",p) & ta$round == rd],
      reported_amount = ifelse(
        runif(1) < prob_true,
        true_amount,
        mb(true_amount +
        cl(pct_slack_claimed_lying_start +
          (rd-1)*pct_slack_claimed_lying_grate +
             rnorm(1, 0, parms$pct_slack_claimed_lying_evar))*
          (6000 - true_amount))
      ),
      pct_slack_claimed = (reported_amount - true_amount) /
        (6000 - true_amount)
    )
  }
  bind_rows(
    lapply(
      c("honesty", "fhonesty"),
      function(e) bind_rows(
        lapply(
          1:runs,
          function(p) bind_rows(lapply(1:rounds, tr, e, p, parms))
        )
      )
    )
  )
```

```
}

run_honesty_sim <- function(te) {
  parms <- tibble(
    prob_true = pct_true,
    prob_true_teffect = te$teffect_true,
    pct_slack_claimed_lying_start = mn_pct_slack_claimed_lying_start,
    pct_slack_claimed_lying_start_teffect = te$teffect_slack_claimed,
    pct_slack_claimed_lying_grate = coef(ols_rounds)[2],
    pct_slack_claimed_lying_grate_teffect = te$teffect_slack_claimed_round,
    pct_slack_claimed_lying_evar = sd_pct_slack_claimed_lying,
  )
  smp <- sim_data(parms)
  true_table <- table(smp$reported_amount == smp$true_amount, smp$experiment)
  true_test <- prop.test(true_table[2,], colSums(true_table))
  # Neg. sign, ci needs to be flipped
  ci_true <- -true_test$conf.int[c(2,1)]
  ci_pct_slack_claimed_fe <- confint(feols(
      pct_slack_claimed ~ experiment | round,
      cluster = c("round", "player_id"),
      data = smp %>% filter(reported_amount != true_amount)
  ))
  ci_pct_slack_claimed_round_fe <- confint(feols(
    pct_slack_claimed ~ experiment*round, cluster = c("round", "player_id"),
    data = smp %>% filter(reported_amount != true_amount)
  ))
  tibble(
    true_teffect_lb = ci_true[1],
    true_teffect_ub = ci_true[2],
    slack_claimed_teffect_lb = pull(ci_pct_slack_claimed_fe[1]),
    slack_claimed_teffect_ub = pull(ci_pct_slack_claimed_fe[2]),
    slack_claimed_round_teffect_lb = ci_pct_slack_claimed_round_fe[4, 1],
    slack_claimed_round_teffect_ub = ci_pct_slack_claimed_round_fe[4, 2]
  )
}

sim_power_honesty <- function(plan) {
  sim_results <- bind_rows(
    lapply(
      1:nrow(plan),
      function(x) {
```

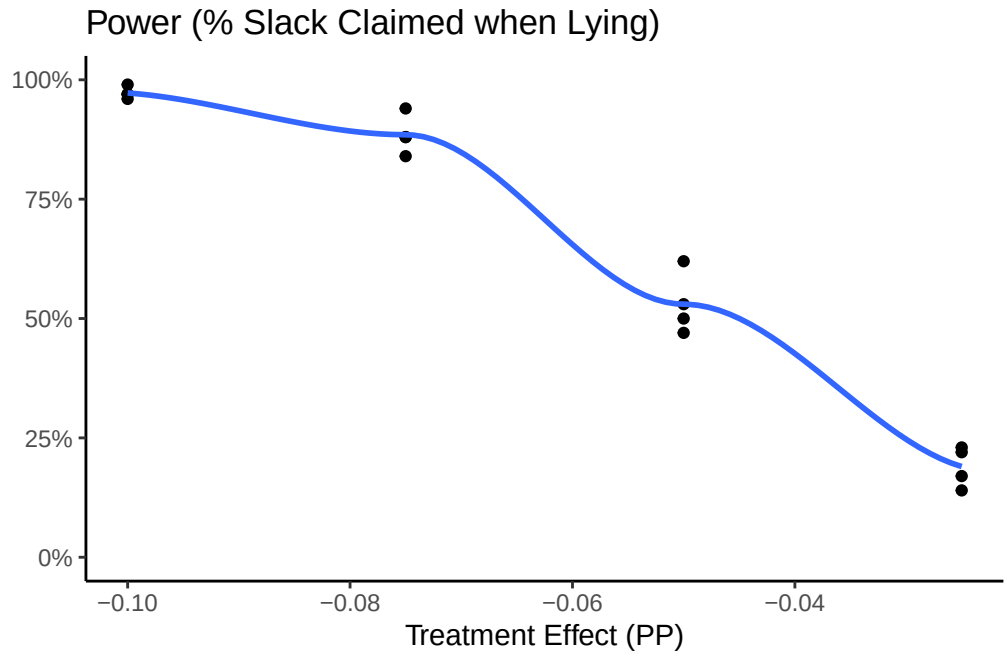
```
          message(
            sprintf("Running honesty sim, plan row %d of %d...", x, nrow(plan)),
            appendLF = F
          )
          rv <- bind_cols(plan[x,], run_honesty_sim(plan[x,]))
          message("")
          rv
        }
      )
    )
  }

  plan <- bind_rows(
    expand_grid(
      n = 1:100,
      teffect_true = c(-0.025, -0.05, -0.075, -0.1),
      teffect_slack_claimed = c(-0.025, -0.05, -0.075, -0.1),
      teffect_slack_claimed_round = 0
    ),
    expand_grid(
      n = 1:100,
      teffect_true = 0,
      teffect_slack_claimed = 0,
      teffect_slack_claimed_round = c(0.01, 0.02, 0.03, 0.04, 0.05)
    )

  )

  message(sprintf(
    "Starting honesty power simulations (%d runs): %s", nrow(plan), Sys.time()
  ))
  honesty_sim_results <- sim_power_honesty(plan)
  write_csv(honesty_sim_results, "../data/generated/honesty_sim_results.csv")
  message(sprintf("Done: %s", Sys.time()))
}

honesty_power <- honesty_sim_results %>%
  group_by(
    teffect_true, teffect_slack_claimed, teffect_slack_claimed_round
  ) %>%
  summarise(
```

```
    power_true = mean(true_teffect_ub < 0),
    power_slack_claimed = mean(slack_claimed_teffect_ub < 0),
    power_slack_claimed_round = mean(slack_claimed_round_teffect_lb > 0),
    .groups = "drop"
  )

ggplot(
  honesty_power %>% filter(teffect_slack_claimed_round == 0),
  aes(x = teffect_true, y = power_true)
) + geom_point() + geom_smooth(se = FALSE) +
  scale_y_continuous(limits = c(0, 1), labels = scales::percent) +
  labs(title = "Power (Prob. True Report)", x = "Treatment Effect (PP)", y = "") +
  theme_classic()
```

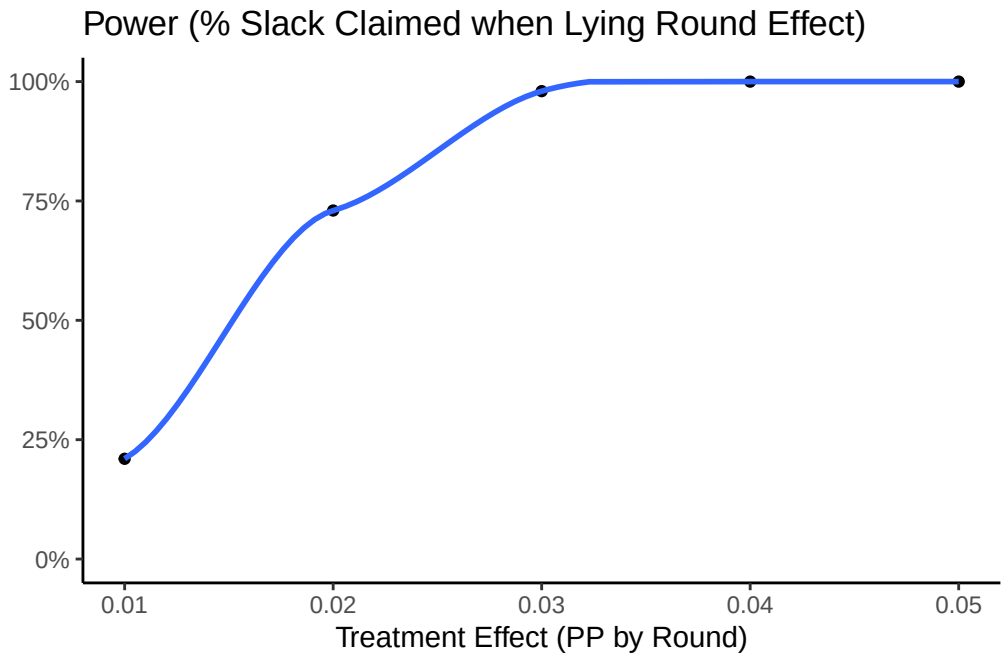



```
ggplot(
  honesty_power %>% filter(teffect_slack_claimed_round == 0),
  aes(x = teffect_slack_claimed, y = power_slack_claimed)
) + geom_point() + geom_smooth(se = FALSE) +
  scale_y_continuous(limits = c(0, 1), labels = scales::percent) +
  labs(
    title = "Power (% Slack Claimed when Lying)",
```

```
    x = "Treatment Effect (PP)", y = ""
  ) +
  theme_classic()
```




```
ggplot(
  honesty_power %>% filter(teffect_slack_claimed == 0),
  aes(x = teffect_slack_claimed_round, y = power_slack_claimed_round)
) + geom_point() + geom_smooth(se = FALSE) +
  scale_y_continuous(limits = c(0, 1), labels = scales::percent) +
  labs(
    title = "Power (% Slack Claimed when Lying Round Effect)",
    x = "Treatment Effect (PP by Round)", y = ""
  ) +
  theme_classic()
```


Power (% Slack Claimed when Lying Round Effect)
100%
75%
50%
25%
0%
0.01
0.02
0.03
0.04
0.05
Treatment Effect (PP by Round)