UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Data Science for Biomedicine
Lab 4

Tim Boleslawsky

Part 1 – Simulate time-to-death

To simulate the time of death of each patient, I implement the function *simulate_ttd*(). The function can be seen below in the section "Code". To test the function, I simulate the TTD for a population of 175 patients, an annual death rate of 30%, and a follow-up period of 1 year. The results of this can be seen in figure 1.

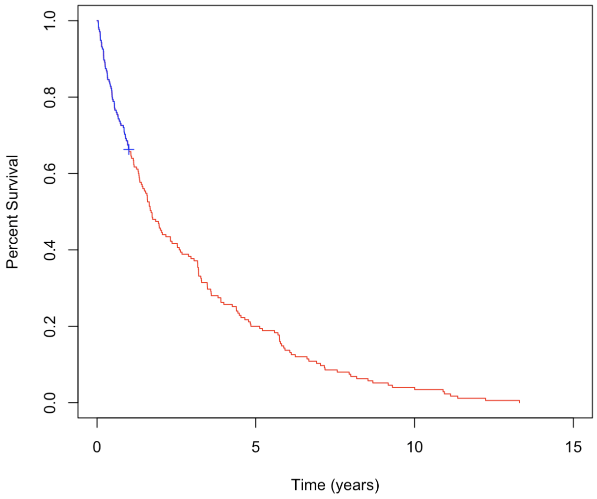


Figure 1: Survival Probability Function

Like in the example figure of the lab, the red line represents the true TTDs and the blue line represents the censored TTDs. We can see that the curve has the expected shape suggesting, that the simulation works as intended.

Code:

```
set.seed(230118001)  # For reproducibility
simulate_ttd <- function(n, annual.rate, followup) {
  lambda <- -log(1 - annual.rate)
  true_ttd <- rexp(n, rate = lambda)
  is_alive <- true_ttd > followup
  censored_ttd <- ifelse(is_alive, followup, true_ttd)
  censored_ttd <- ceiling(censored_ttd * 365) / 365
  result <- data.frame(
    true_ttd = as.numeric(true_ttd),
    censored_ttd = as.numeric(censored_ttd),
    is_alive = is_alive
  )

  return(result)
}

n_patients <- 175
yearly_death_rate <- 0.30
follow_up_years <- 1
ttd_data <- simulate_ttd(n_patients, yearly_death_rate, follow_up_years)
```

```
true_surv <- Surv(ttd_data$true_ttd, event = rep(1, n_patients))
true_fit <- survfit(true_surv ~ 1)

censored_surv <- Surv(ttd_data$censored_ttd, event = !ttd_data$is_alive)
censored_fit <- survfit(censored_surv ~ 1)

plot1 <- plot(true_fit, col = "red", xlab = "Time␣(years)", ylab = "Percent
      main = "Survival␣Probability␣Function", xlim = c(0, 15), ylim = c(0, 1
      conf.int = FALSE, mark.time = TRUE)
       lines(censored_fit, col = "blue", conf.int = FALSE, mark.time = TRUE)

print(plot1)
```

Part 2 – Simulate a study

I now want to simulate a study. For this, I implement the function *simulate_study*(). This function takes information about the study, calculates the active and control TTDs, the active and control baselines, and the active and control CFBs. The function returns a tibble with the relevant information. To test this function, I simulate a study and look at the survival functions of the active and the control group. The results can be seen in figure 2.

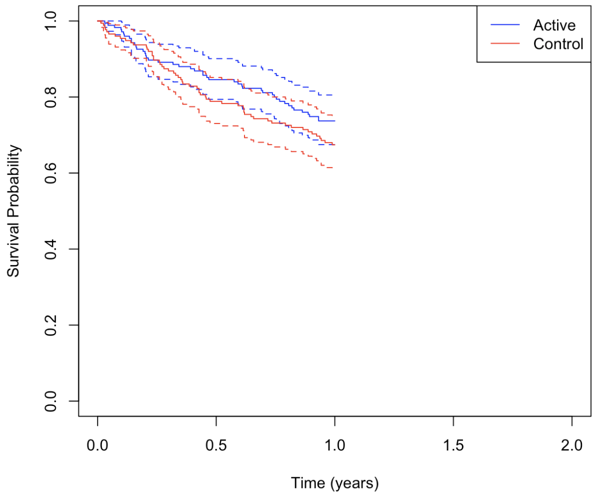


Figure 2: Survival Probability Function Active vs. Control

We can see that the survival probability functions behave as expected. Both have a slow decline, with the active having a slower decline than the control. This suggests efficacy in the data.

Code:

```
simulate_study <- function(
     n.active = 175,
     n.control = 175,
     mu.baseline = 200,
     sd.baseline = 70,
     mu.cfb.control = 0,
     mu.cfb.active = 25,
     sd.cfb = 70,
     annual.rate.death.control = 0.3,
     death.rrr.active = 0.2,
     followup = 1,
     measurement.time.cfb = followup
) {
  annual.rate.death.active <- annual.rate.death.control *
                                   (1 - death.rrr.active)
  active_ttd <- simulate_ttd(n.active,
                        annual.rate.death.active,
                        followup)
  control_ttd <- simulate_ttd(n.control,
                              annual.rate.death.control,
                              followup)
  active_baseline <- floor(rnorm(n.active,
                              mean = mu.baseline,
                              sd = sd.baseline))
  control_baseline <- floor(rnorm(n.control,
                              mean = mu.baseline,
                              sd = sd.baseline))
  active_cfb <- ifelse(active_ttd$is_alive, rnorm(n.active,
                                               mean = mu.cfb.active,
                                               sd = sd.cfb), NA)
  control_cfb <- ifelse(control_ttd$is_alive, rnorm(n.control,
                                               mean = mu.cfb.control,
                                               sd = sd.cfb), NA)

  study_data <- tibble(
    group = rep(c(”Active”, ”Control”), times = c(n.active, n.control)),
    ttd = c(active_ttd$censored_ttd, control_ttd$censored_ttd),
    alive = c(active_ttd$is_alive, control_ttd$is_alive),
    baseline = c(active_baseline, control_baseline),
    cfb = c(active_cfb, control_cfb)
  )

  return(study_data)
}

study_data <- simulate_study()

active_surv <- Surv(study_data$ttd[study_data$group == ”Active”],
                event = !study_data$alive[study_data$group == ”Active”])
control_surv <- Surv(study_data$ttd[study_data$group == ”Control”],
                event = !study_data$alive[study_data$group == ”Control”])

active_fit <- survfit(active_surv ~ 1)
control_fit <- survfit(control_surv ~ 1)

plot2 <- plot(active_fit, col = ”blue”, xlab = ”Time␣(years)”,
                ylab = ”Survival␣Probability”,
                main = ”Survival␣Probability␣Function”,
                xlim = c(0, 2), ylim = c(0, 1))
                lines(control_fit, col = ”red”)
                legend(”topright”,
                    legend = c(”Active”, ”Control”),
                    col = c(”blue”, ”red”), lty = 1)

print(plot2)
```

Part 3 - Inference on the surviving patients

When trying inference on the surviving patients, a few things become clear. First, I want to look at the ANOVA analysis. For the covariate group, the ANOVA analysis calculates a p-value of $7.12e - 05$ suggesting a very strong significance and a large effect of group on the target variable.

For the power analysis I get the following results: h = 0.3871693, n1 = 129, n2 = 118, sig.level = 0.05, power = 0.8598043, alternative = two.sided. This power is in within the usual bounds of acceptable power values (0.8, 0.9) and is therefore appropriate.

Code:

```r
survived_data <- study_data %>% filter(alive == TRUE)

anova_model <- aov(cfb ~ group, data = survived_data)

print("Summary of ANOVA Model:")
print(summary(anova_model))

success_threshold <- 25

prop_active <- mean(survived_data$cfb[survived_data$group == "Active"]
                                   >= success_threshold, na.rm = TRUE)
prop_control <- mean(survived_data$cfb[survived_data$group == "Control"]
                                   >= success_threshold, na.rm = TRUE)

n_active <- sum(survived_data$group == "Active")
n_control <- sum(survived_data$group == "Control")

power_result <- pwr.2p2n.test(
  h = ES.h(prop_active, prop_control),
  n1 = n_active,
  n2 = n_control,
  sig.level = 0.05,
  power = NULL
)

print("Power Result")
print(power_result)
```

Part 4 – Calculate power by simulation

When estimating the Type I error and the Power empirically for alpha = 0.05, I get the following results.

- Type I Error (n=8) = 0.053, CI lower = 0.03911426, CI upper = 0.06688574.
- Power (n=8) = 0.317, CI lower = 0.2881599, CI upper = 0.3458401.
- Type I Error (n=24) = 0.048, CI lower = 0.03475063, CI upper = 0.06124937.
- Power (n=24) = 0.686, CI lower = 0.6572338, CI upper = 0.7147662.

When comparing these empirical results to the analytical results I got in part three, we see that there is a significant difference. This is likely because the analytical power calculation assumes a specific effect size (h=0.419) and sample sizes (n1=139, n2=124), which are based on the proportions of successes in the active and control groups. The empirical power calculation, on the other hand, is based on a fixed effect size (effect=0.5) and smaller sample sizes (n=8 and n=24).

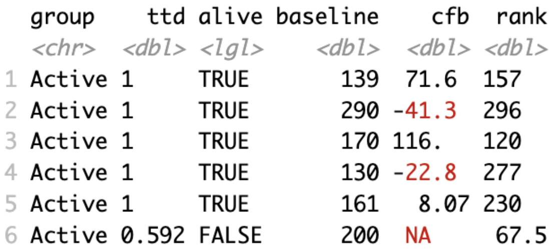
We can see that the power increases with sample size in the empirical results if we compare the power of n=8 and n=24. This leads to the obvious conclusion that the power is subject to the sample size and because the analytical power calculation assumes a higher sample size, it makes sense that the power would be higher.

Code:

```
z.test.pvalue <- function(n, effect) {
  x <- mean(rnorm(n=n, mean=effect, sd = 1))
  2*pnorm(abs(x), mean=0, sd = 1/sqrt(n), lower.tail=FALSE)
}

simulate_z.test.pvalue <- function (N.sim = 1e3, ...) {
  pvalue <- foreach(i = seq(N.sim), .combine=”c”) %do% {
    z.test.pvalue(...)
  }
  tibble(..., pvalue)
}

data <- bind_rows(
  simulate_z.test.pvalue(n=8, effect = 0),
  simulate_z.test.pvalue(n=8, effect = 0.5),
  simulate_z.test.pvalue(n=24, effect = 0),
  simulate_z.test.pvalue(n=24, effect = 0.5)
)
plot3 <- ggplot(data |> mutate(effect = factor(effect)),
       aes(x=pvalue)) +
  facet_grid(effect ~ n,
             labeller=labeller(n = function(x) paste0(”n=”,x),
                               effect = function(x) paste0(”effect=”,x)),
             scales = ”free_y”) +
  scale_y_continuous(labels = scales::percent) +
  geom_histogram(aes(y=after_stat(width*density)),
    breaks=seq(0,1,by=0.05))+
  labs(y="Proportion of p-values in bin\n(bin width=0.05)",
    x="p-value") +
  ggtitle("Histogram of p-values for a two-sided z-test")

print(plot3)

calculate_proportion_ci <- function(pvalues, alpha = 0.05) {
  prop <- mean(pvalues <= alpha)

  se <- sqrt(prop * (1 - prop) / length(pvalues))

  ci_lower <- prop - 1.96 * se
  ci_upper <- prop + 1.96 * se

  return(list(proportion = prop,
              ci_lower = ci_lower,
              ci_upper = ci_upper))
}

pvalues_effect_0_n8 <- data %>% filter(effect == 0, n == 8)
                             %>% pull(pvalue)
pvalues_effect_0.5_n8 <- data %>% filter(effect == 0.5, n == 8)
                             %>% pull(pvalue)
pvalues_effect_0_n24 <- data %>% filter(effect == 0, n == 24)
                             %>% pull(pvalue)
pvalues_effect_0.5_n24 <- data %>% filter(effect == 0.5, n == 24)
                             %>% pull(pvalue)

type1_error_n8 <- calculate_proportion_ci(pvalues_effect_0_n8)
power_n8 <- calculate_proportion_ci(pvalues_effect_0.5_n8)
type1_error_n24 <- calculate_proportion_ci(pvalues_effect_0_n24)
power_n24 <- calculate_proportion_ci(pvalues_effect_0.5_n24)

print("Type I Error (n=8):")
print(type1_error_n8)

print("Power (n=8):")
print(power_n8)

print("Type I Error (n=24):")
print(type1_error_n24)

print("Power (n=24):")
print(power_n24)
```

Part 5 – Combined inference on TTD and 6MWD

For the combined inference on the TTD and 6MWD, I first implement a function $rank\_patients()$ to rank patients based on their prognosis. The results of this ranking can be found in figure 3.



```
  group     ttd alive baseline    cfb  rank
  <chr>   <dbl> <lgl>    <dbl>  <dbl> <dbl>
1 Active 1     TRUE       139  71.6   157
2 Active 1     TRUE       290 -41.3   296
3 Active 1     TRUE       170 116.    120
4 Active 1     TRUE       130 -22.8   277
5 Active 1     TRUE       161   8.07  230
6 Active 0.592 FALSE      200  NA      67.5
```

Figure 3: Results of Ranking Function

With this data, I now implement the functions $perform\_anova\_on\_ranks()$ and $simulate\_studies\_pvalue$ to simulate multiple studies, perform ANOVA analysis on them, and then extract the p-value. When calculating the mean Type I error over this data, I get the value: 0.045. This is in line with the Type I errors we saw in the empirical analysis (0.053 for n=8 and 0.048 for n=24). This suggests that with increasing patients per arm, the Type I error sinks.

This is in line with the analysis I did on the effect of the number of patients per arm on the power. The results of this analysis can be seen in figure 4, where it is clearly visible, that an increase in number of patients per arm increases the power.

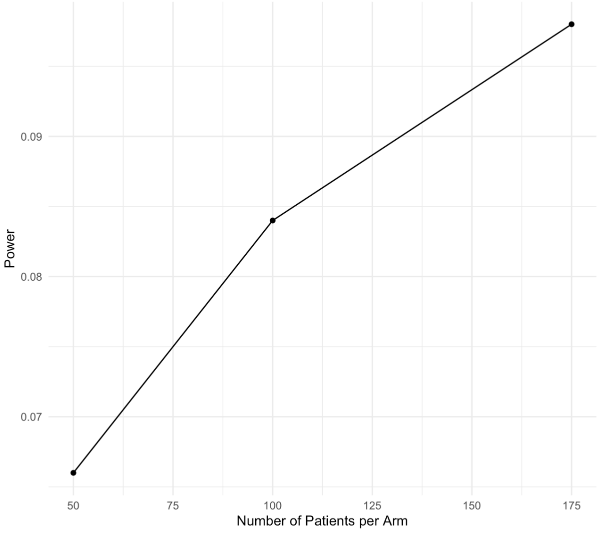


Figure 4: Power vs. Number of Patients per Arm

Code:

```
rank_patients <- function(study_data) {
  study_data %>%
    mutate(
      rank_ttd = ifelse(!alive, rank(ttd, ties.method = "average"), NA),
      rank_cfb = ifelse(alive, rank(-cfb, ties.method = ”average”), NA),
      rank = ifelse(!alive, rank_ttd, max(rank_ttd, na.rm = TRUE) + rank_cf
    ) %>%
    select(-rank_ttd, -rank_cfb)
}

ranked_data <- rank_patients(study_data)
print(head(ranked_data))

perform_anova_on_ranks <- function(ranked_data) {
  anova_model <- aov(rank ~ group, data = ranked_data)
  pvalue <- summary(anova_model)[[1]]$‘Pr(>F)‘[1]
  return(pvalue)
}

simulate_studies_pvalue <- function(
    N.sim = 1000,
    n.active = 175,
    n.control = 175,
    mu.cfb.active = 0,
    death.rrr.active = 0,
    ...
) {
  pvalues <- foreach(i = seq(N.sim), .combine = ”c”) %do% {
    study_data <- simulate_study(
      n.active = n.active,
      n.control = n.control,
      mu.cfb.active = mu.cfb.active,
      death.rrr.active = death.rrr.active,
      ...
    )

    ranked_data <- rank_patients(study_data)

    pvalue <- perform_anova_on_ranks(ranked_data)

    return(pvalue)
  }
  tibble(..., pvalues)
}

type1_error_data <- simulate_studies_pvalue(
  N.sim = 1000,
  mu.cfb.active = 0,
  death.rrr.active = 0
)

type1_error <- mean(type1_error_data$pvalues <= 0.05)
```

```
print(paste("Type␣I␣Error:", type1_error))

power_results <- foreach(n = c(50, 100, 175), .combine = "rbind") %do% {
  power_data <- simulate_studies_pvalue(
    N.sim = 1000,
    n.active = n,
    n.control = n,
    mu.cfb.active = 25,
    death.rrr.active = 0.2
  )

  power <- mean(power_data$pvalues <= 0.05)
  tibble(n = n, power = power)
}

print("Power␣Results␣for␣Varying␣Numbers␣of␣Patients␣per␣Arm:")
print(power_results)

plot4 <- ggplot(power_results, aes(x = n, y = power)) +
  geom_line() +
  geom_point() +
  labs(
    x = "Number␣of␣Patients␣per␣Arm",
    y = "Power",
    title = "Power␣vs.␣Number␣of␣Patients␣per␣Arm"
  ) +
  theme_minimal()

print(plot4)
```

Part 6 - Evaluate Power and Type I error based on SD and on effect size

To assess the effect of the baseline annual death rate, the drugs RRR and the expected 6MWD CFB on the power, I created three distinct plots. These plots help us understand how each parameter individually affects the power of the study. They are represented in figures 5, 6, 7.

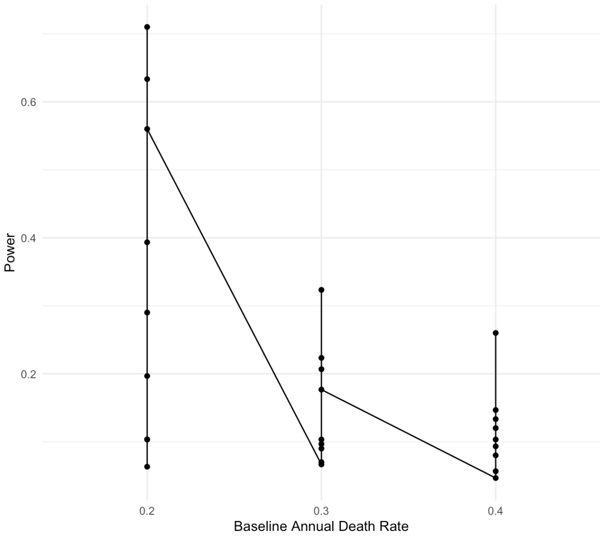


Figure 5: Effect of Baseline Annual Death Rate on Power

In figure 5 we see the effect of different baseline annual death rates (0.2, 0.3, 0.4) on the power. The dots represent different simulated studies with the annual death rates as x. The connecting lines show a trend of the power means. In this figure, we can see that the annual death rate has a fairly large negative effect on power; the higher the annual death rate, the lower the power. This can be explained like this: With a lower annual death rate, fewer patients die, so more patients survive and contribute to the 6MWD CFB analysis. This increases the effective sample size, leading to higher power.

In figure 6 we can see the effect of different RRR values (0.1,0.2,0.3) on the power. The structure of the plot is the same as before. We can again see a decrease in power, as we increase the RRR values. But this time the effect is rather small. To understand why this happens, we need to understand what RRR stands for. The Relative Risk Reduction (RRR) measures how much the drug reduces the risk of death compared to the control group. A higher RRR means therefore the drug is more effective at reducing deaths. A higher RRR then also means fewer patients in the active group die, so more patients survive and contribute to the 6MWD CFB analysis.

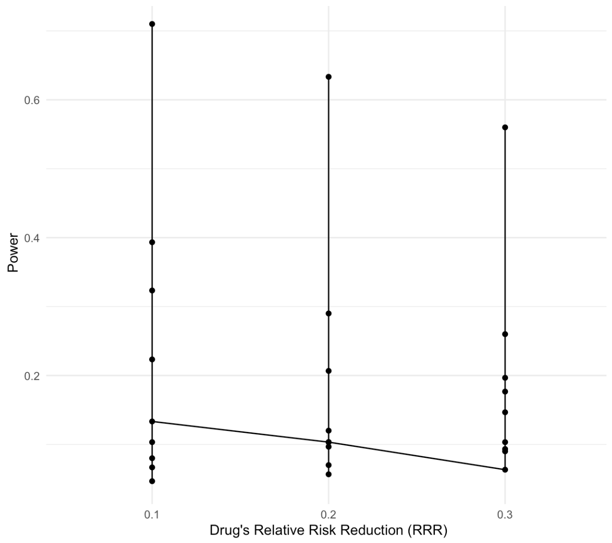


Figure 6: Effect of Drug's RRR on Power

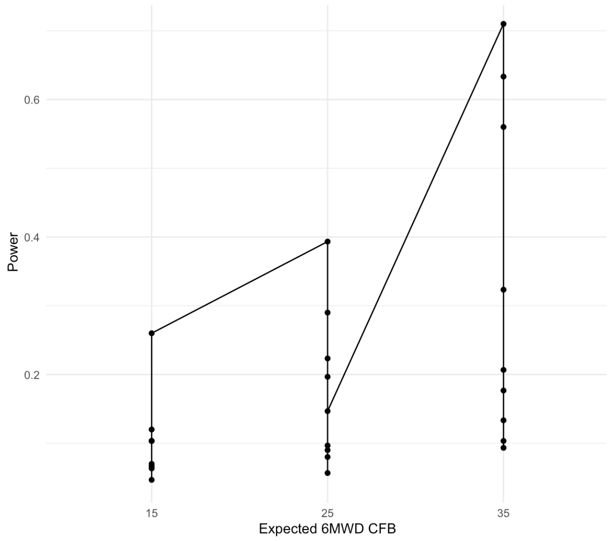


Figure 7: Effect of Expected 6MWD CFB on Power

In figure 7 we can see the effect of the expected 6MWD CFB (15,25,35) on the power. The structure of the plot is the same as before. This time, we can see a relatively stark increase in power, as we increase the expected 6MWD CFB. This is likely because a higher expected CFB means the active group has a larger improvement in 6MWD compared to the control group. This makes it easier to detect a statistically significant difference between the two groups.

It needs to be said that the different studies depicted in the plots in this section do not measure the same thing. This is because the parameters are being varied, which fundamentally changes the context and the outcomes of the study.

The different studies can be broken down like this:

- Studies with different baseline annual death rates are measuring the impact of mortality on the power of the study.

- Studies with different RRR values are measuring the effectiveness of the drug at reducing deaths.
- Studies with different expected 6MWD CFB values are measuring the magnitude of the treatment effect on the 6-minute walk distance (6MWD).

Code:

```
simulate_studies_power <- function(
    N.sim = 300,
    n.active = 175,
    n.control = 175,
    mu.baseline = 200,
    sd.baseline = 70,
    mu.cfb.control = 0,
    mu.cfb.active = 25,
    sd.cfb = 70,
    annual.rate.death.control = 0.3,
    death.rrr.active = 0.2,
    followup = 1,
    measurement.time.cfb = followup
) {
  pvalues <- foreach(i = seq(N.sim), .combine = "c") %do% {
    study_data <- simulate_study(
      n.active = n.active,
      n.control = n.control,
      mu.baseline = mu.baseline,
      sd.baseline = sd.baseline,
      mu.cfb.control = mu.cfb.control,
      mu.cfb.active = mu.cfb.active,
      sd.cfb = sd.cfb,
      annual.rate.death.control = annual.rate.death.control,
      death.rrr.active = death.rrr.active,
      followup = followup,
      measurement.time.cfb = measurement.time.cfb
    )

    ranked_data <- rank_patients(study_data)

    pvalue <- perform_anova_on_ranks(ranked_data)

    return(pvalue)
  }
  power <- mean(pvalues <= 0.05)
  return(power)
}

annual_rates <- c(0.2, 0.3, 0.4)
rrr_values <- c(0.1, 0.2, 0.3)
cfb_values <- c(15, 25, 35)
```

```
power_results <- expand.grid(
  annual.rate.death.control = annual_rates,
  death.rrr.active = rrr_values,
  mu.cfb.active = cfb_values
) %>%
  rowwise() %>%
  mutate(
    power = simulate_studies_power(
      annual.rate.death.control = annual.rate.death.control,
      death.rrr.active = death.rrr.active,
      mu.cfb.active = mu.cfb.active
    )
  ) %>%
  ungroup()

plot_annual_rate <- ggplot(power_results,
                      aes(x = factor(annual.rate.death.control),
                      y = power, group = 1)) +
                      geom_line() +
                      geom_point() +
                    labs(
                      x = "Baseline Annual Death Rate",
                      y = "Power",
                      title = "Effect of Baseline Annual Death Rate on Power"
                    ) +
                    theme_minimal()

plot_rrr <- ggplot(power_results, aes(x = factor(death.rrr.active),
                      y = power, group = 1)) +
                        geom_line() +
                        geom_point() +
                        labs(
                          x = "Drug's Relative Risk Reduction (RRR)",
                          y = "Power",
                          title = "Effect of Drug's RRR on Power"
                        ) +
                        theme_minimal()

plot_cfb <- ggplot(power_results, aes(x = factor(mu.cfb.active),
                      y = power, group = 1)) +
                        geom_line() +
                        geom_point() +
                        labs(
                          x = "Expected 6MWD CFB",
                          y = "Power",
                          title = "Effect of Expected 6MWD CFB on Power"
                        ) +
                    theme_minimal()

print(plot_annual_rate)
print(plot_rrr)
print(plot_cfb)
```