The Costs of Counterparty Risk in Long Term Contracts Code Guide - Section 7

Michael Duarte Gonçalves

July 4, 2025

Overview

This section develops a rigorous quantitative framework to compare the effects of public guarantees (G) and public subsidies (T) in renewable energy investment markets under risk aversion and uncertainty.

Contents

1 Modeling Markets With Counterparty Risk (CPR) — when $G = T$

1.1 Conceptual Approach

The analysis proceeds as follows:

- We determine, for each plant and scenario, the subsidy level T^* that equates total public expenditure under a subsidy scheme to that under a public guarantee scheme, ensuring:

$$T^* \cdot \sum_{i \in \mathcal{I}_\gamma^{G=T}(\theta)} q_i = \sum_{i \in \mathcal{I}_\gamma^{G=T}(\theta)} \left(q_i \cdot \gamma \cdot x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp \right)$$

  This ensures that public spending under the subsidy scheme is equivalent to the fiscal cost of guarantees for each (γ, θ) combination, taking into account the per-unit cost of public funds λ.

- For each (γ, θ) scenario, we solve for T^* numerically and verify the equivalence of public expenditures at the plant level.

- Using T^*, we recompute equilibrium contract supply, prices, and plant-level profits, and compare these outcomes to the public guarantee case.

- We construct welfare metrics for both mechanisms, including:
  - Total welfare for each risk aversion level (γ)
  - Seller and buyer profit decompositions
  - Comparative ratios of welfare and profit shares between the two policy regimes

- The results are visualized through a series of comparative plots, including welfare ratios, profit shares, and buyer/seller profit ratios as functions of risk aversion and demand.

- All scenario results are systematically exported to Excel workbooks, supporting transparency, reproducibility, and further analysis.

Summary:
This framework enables a direct, apples-to-apples comparison of public guarantees and public subsidies, quantifying their respective impacts on market equilibrium, welfare distribution, and the allocation of public funds across heterogeneous risk profiles.

1.2 Numerical Methods

1.2.1 Objective

The aim of this section is to determine, for each combination of risk aversion (γ) and plant size (θ), the value of T^* that results in the **same total public expenditure** as under a public guarantee scheme. In other words, we seek the value of T^* such that the government's outlay under both mechanisms is equivalent for each plant.

1.2.2 Methodology and Implementation

The following steps are performed:

1. **Function Definition:** We define a function, `find_T_star_for_gamma_theta()`, that computes T^* for a given pair (γ, θ). This function:
   - Filters the dataset for the specified γ and θ.
   - If no data is found, returns `NA`.
   - Otherwise, uses a root-finding algorithm (`uniroot`) to solve for T^* such that:

$$T^* \cdot \sum_{i \in \mathcal{I}_\gamma^{G=T}(\theta)} q_i = \sum_{i \in \mathcal{I}_\gamma^{G=T}(\theta)} \left(q_i \cdot \gamma \cdot x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp \right)$$

2. **Grid Construction:** All unique combinations of γ and θ present in the data are extracted to form a grid.

3. **Computation of T^*:** The function is mapped across all grid points to compute T^* for each (γ, θ) pair.

4. **Verification:** We check that, for each (γ, θ), the calculated T^* does indeed equate the public expenditures under both schemes. Specifically, we compute the left-hand side ($\lambda \cdot T^* \cdot \sum q_i$) and the right-hand side ($\sum q_i \cdot \gamma \cdot x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp$), and confirm that their difference is within a very small numerical tolerance (1×10^{-7}).

```
# Section 7: Public G. and Public Subsidies Comparison ------

# find the T* that makes public expenditure under G same as public
   expenditure under T,
# for every plant.

#  ----------------------------------- #
# Function
find_T_star_for_gamma_theta <- function(gamma_val, theta_val, df,
   lambda) {
  # Filter for the given gamma and theta
  df_sub <- df |>
    filter(gamma == gamma_val, theta == theta_val)

  # Return NA if no data
  if (nrow(df_sub) == 0) {
    return(tibble(gamma = gamma_val, theta = theta_val, T_star = NA_
       real_))
  }

  # Compute the root
```

```r
  root <- uniroot(
    f = function(T_val) {
      sum_q <- sum(df_sub$q_i_mwh, na.rm = TRUE)
      sum_q_lambda <- sum(df_sub$q_i_mwh_lambda_gamma_x_integral)
      (T_val * sum_q * lambda) - sum_q_lambda
    },
    interval = c(0, 100), # a big interval just in case
    tol = 1e-6
  )

  return(tibble(gamma = gamma_val,
                theta = theta_val,
                T_star = root$root))
}

#  ----------------------------------- #

# Extract the T_star_values and some checks

# Get all combinations from your dataset
gamma_theta_grid <- public_guarantees |>
  distinct(gamma, theta)

# Map the function across all rows
T_star_results <- gamma_theta_grid |>
  mutate(
    T_star = purrr::map2_dbl(gamma, theta,
                             ~ find_T_star_for_gamma_theta(.x, .y,
                                public_guarantees, lambda)$T_star)
  )

# This is just a dataset to check if lambda * T_star * q_i_mwh = q_i_
   mwh_lambda_gamma_x_integral
# They should be equal

check_public_expenditures <- public_guarantees |>
  left_join(T_star_results, by = c("gamma", "theta")) |>
  mutate(
    lhs = lambda * T_star * q_i_mwh,
    rhs = q_i_mwh_lambda_gamma_x_integral,
    diff = lhs - rhs
  ) |>
  arrange(theta, gamma, xf_c_cpr)



# Contains exactly the same values. We are good!
check_public_expend_summary <- check_public_expenditures |>
  filter(xf_max_cpr <= xf_equilibrium & cumulative_capacity < theta) |>
  group_by(gamma, theta) |>
  reframe(
    lhs = lambda * first(T_star) * sum(q_i_mwh, na.rm = TRUE),
    rhs = sum(q_i_mwh_lambda_gamma_x_integral, na.rm = TRUE),
    # show that lhs (i.e.) lambda * T_star * q_i_mwh = q_i_mwh_lambda_
       gamma_x_integral
    # so the equal_within_tolerance should be TRUE for all (1e-7 =
       0.0000001)
    equal_within_tolerance = abs(lhs - rhs) < 1e-7
```

```
  )
```

1.2.3 Comparison of Results under Equivalent Subsidy and Guarantee Schemes

Purpose The following code constructs a comprehensive dataset that enables direct comparison between the outcomes under a public subsidy scheme (T^*) and a public guarantee scheme. By leveraging the previously computed T^* values, we simulate plant-level outcomes under both policy regimes for a range of parameter values.

Methodology and Steps

1. **Dataset Expansion and Merging:**
   - We start with the long-format data for individual wind and solar projects.
   - For each project, we generate all relevant combinations of θ (demand) using `crossing()`.
   - We merge in the corresponding T^* values (from Section 7) for each (γ, θ) pair.
2. **Root Finding for Financial Quantities:**
   - For each plant and parameter combination, we use root-finding functions (`find_f_root`, `find_f_spot_root`, `find_upper_root`) to compute key thresholds (f_c, f_{spot}, f_{upper}).
   - The `future_pmap_dbl` function is used for efficient parallel computation across all parameter combinations.
3. **Post-processing and Derived Quantities:**
   - We compute maximum thresholds, aggregate quantities (e.g., cumulative production and capacity), and annotate results for further analysis.
   - We add some variables such as $\lambda \cdot T^* \cdot q_i$ and $\lambda \cdot \gamma \cdot x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp$ at the plant level, are calculated to facilitate expenditure comparisons.
4. **Merging with Analytical Integrals:**
   - We merge in pre-computed integrals from the public guarantee analysis to enable direct comparison of plant-level outcomes under both schemes.

```
results_with_T_G_comparison <- wind_solar_proj_2022_long |>
  crossing(theta = theta_values) |>
  left_join(T_star_results, by = c("gamma", "theta")) |>
  mutate(
    f_c_cpr = future_pmap_dbl(
      list(q_i_mwh, x, gamma, r_0, alpha, beta, total_cost, T_star),
      ~ coalesce(
        find_f_root(
          ..1, ..2, ..3, ..4, ..5, ..6, ..7, ..8
        ),
        1  # Default value
      )
    ),

    f_spot_cpr = future_pmap_dbl(
      list(q_i_mwh, x, gamma, r_0, alpha, beta, total_cost, expected_p,
          r, T_star),
```

```r
      ~ coalesce(
        find_f_spot_root(
          ..1, ..2, ..3, ..4, ..5, ..6, ..7, ..8, ..9, ..10
        ),
        0  # Default value
      )
    ),
    f_upper = future_pmap_dbl(
      list(q_i_mwh, x, gamma, r_0, alpha, beta, total_cost, T_star),
      ~ find_upper_root(
          ..1, ..2, ..3, ..4, ..5, ..6, ..7, ..8
        )
      )
    ) |>
  mutate(
    f_max_cpr = pmax(f_c_cpr, f_spot_cpr, na.rm = TRUE),
    xf_c_cpr = x * f_c_cpr,
    xf_spot_cpr = x * f_spot_cpr,
    xf_max_cpr = x * f_max_cpr
  )

results_with_T_G_comparison <- results_with_T_G_comparison |>
  arrange(theta, gamma, xf_max_cpr) |>
  group_by(gamma, theta, T_star) |>
  mutate(cumulative_production = cumsum(q_i_mwh),
         cumulative_capacity = cumsum(capacity),
         f_upper_message = if_else(!is.na(f_upper),
           if_else(f_upper > expected_p, "f_upper > E(p)", "f_upper <=
              E(p)"),
           NA_character_
         )
  ) |>
  ungroup() |>
  mutate(lambda = lambda,
         lambda_q_i_mwh_T = lambda * T_star * q_i_mwh
         )


# We will merge the results of the x_integral_by_theta with our main
    dataset

x_integral_by_theta <- public_guarantees |>
  distinct(theta, f_equilibrium) |>
  mutate(
    x_integral = x * compute_integral_vec(f_equilibrium, alpha = alpha,
        beta = beta)
  ) |>
  select(theta, f_equilibrium, x_integral)


results_with_T_G_comparison <- results_with_T_G_comparison |>
  left_join(x_integral_by_theta, by = "theta") |>
  mutate(q_i_mwh_lambda_gamma_x_integral = q_i_mwh * lambda * gamma * x
     _integral)
```

1.2.4 Computation of Welfare and Equilibrium Prices

Purpose We describe the process for constructing the equilibrium prices and welfare dataset, which is essential for comparing the economic outcomes of public subsidy and public guarantee schemes. The approach ensures that, for each combination of risk aversion (γ) and demand (θ), the equilibrium price and welfare are determined in a consistent and robust manner.

Step-by-Step Methodology

1. **Integrating Equilibrium Prices with the Main Results:**
   - We merge the calculated equilibrium prices and quantities back into the main comparison dataset, ensuring that all subsequent calculations use the correct market-clearing price and quantity.
   - We compute the equilibrium price per unit (f^*) and other derived variables needed for welfare and profit calculations.
2. **Calculating Welfare and Profits:**
   - For each plant, we compute then $R_i(f^*, \gamma)$ and sellers profits $\pi_S(f^*)$
3. **Reordering and Finalizing the Dataset:**
   - Variables are reordered for clarity and consistency in reporting.
   - Additional derived variables are computed for convenience in subsequent analyses.

```
# 1. For each gamma, get the last row before theta_val (preserving
   cumulative capacity)
before_theta_T <- results_with_T_G_comparison |>
  group_by(gamma, theta, T_star) |>
  arrange(cumulative_capacity) |>
  filter(cumulative_capacity < theta, xf_max_cpr < threshold_price) |>
  slice_tail(n = 1) |>
  ungroup()

# 2. For each gamma, get the first row at or after theta_val (to peek
   at the next price)
after_theta_T <- results_with_T_G_comparison |>
  group_by(gamma, theta) |>
  arrange(cumulative_capacity) |>
  filter(cumulative_capacity >= theta) |>
  slice_head(n = 1) |>
  ungroup() |>
  select(gamma, T_star, theta, next_price = xf_max_cpr, T_star, theta)

# 3. Join the two and update the price:
#    - Keep the cumulative capacity from before_theta
#    - Update the price to threshold_price only if the next row's price
     is >= threshold_price.
equilibrium_prices_T <- before_theta_T |>
  left_join(after_theta_T, by = c("gamma", "T_star", "theta")) |>
  mutate(
    equilibrium_price = if_else(
      !is.na(next_price) & next_price >= threshold_price,
      threshold_price,
      next_price
```

```
    ),
    equilibrium_quantity = cumulative_capacity
  )  |>
  select(gamma, T_star, theta, equilibrium_quantity, equilibrium_price)

# View the combined results
print(equilibrium_prices_T)

#  ---------------------------------- #

# Equilibrium Prices Dataset + Some reordering

results_with_T_G_comparison <- results_with_T_G_comparison |>
  left_join(
    equilibrium_prices_T |>
      select(gamma, theta, T_star, equilibrium_price, equilibrium_
         quantity),
    by = c("gamma", "T_star", "theta")
  )

results_with_T_G_comparison <- results_with_T_G_comparison |>
  mutate(f_equilibrium = equilibrium_price / x,
         xf_equilibrium = equilibrium_price) |>
  select(-equilibrium_price)


results_with_T_G_comparison <- results_with_T_G_comparison |>
  arrange(theta, gamma, xf_max_cpr) |>
  rowwise() |>
  mutate(R_f_equilibrium_cpr =
           compute_R_value_gamma(f_equilibrium, gamma, q_i_mwh, x, r_0,
               alpha, beta)) |>
  mutate(
    x_q_exp_p_total_costs_R = x * expected_p * q_i_mwh - total_cost
    - R_f_equilibrium_cpr
  )

results_with_T_G_comparison <- results_with_T_G_comparison |>
  mutate(
    profit_cpr_contracts =
      vectorized_pi_s(
        f = f_equilibrium,
        q_i = q_i_mwh,
        x = x,
        gamma = gamma,
        r_0 = r_0,
        alpha = alpha,
        beta = beta,
        total_costs = total_cost,
        T_values = T_star
      )
    )


ordered_vars <- c(
  "theta", "gamma", "lambda", "T_star", "f_c_cpr", "f_spot_cpr",
  "f_max_cpr", "f_equilibrium", "R_f_equilibrium_cpr",
  "xf_c_cpr", "xf_spot_cpr", "xf_max_cpr", "xf_equilibrium",
```

```
  "equilibrium_quantity", "x_q_exp_p_total_costs", "x_q_exp_p_total_
     costs_R",
  "cumulative_production", "cumulative_capacity", "profit_cpr_contracts
     ",
  "x_integral", "lambda_q_i_mwh_T", "q_i_mwh_lambda_gamma_x_integral"
)

# Reorder dataframe
results_with_T_G_comparison <- results_with_T_G_comparison |>
  select(
    everything(),              # Keeps all variables in current order
       ...
    -all_of(ordered_vars),     # ...but temporarily removes the ones
       you want to reorder
    -profits_sp_no_cpr,                # temporarily remove profits_sp_
       no_cpr to place it explicitly
    profits_sp_no_cpr,                 # put profits_sp_no_cpr in place
    all_of(ordered_vars)       # then add ordered vars in the desired
       order
  )

results_with_T_G_comparison <- results_with_T_G_comparison |>
  mutate(q_i_mwh_T = q_i_mwh * T_star,
         q_i_mwh_gamma_x_integral = q_i_mwh * gamma * x_integral) |>
  arrange(theta, gamma, T_star, xf_max_cpr)
```

1.2.5 Construction of the Welfare Dataset

Purpose This section describes the process for constructing a comprehensive welfare dataset, which is essential for comparing the economic outcomes of different policy scenarios (public subsidy and public guarantee) under varying levels of risk aversion (γ) and plant size (θ). The dataset includes not only welfare levels but also comparative metrics such as welfare gains, ratios, and gaps relative to baseline scenarios.

Step-by-Step Methodology

1. **Welfare W^0:**

   - We first compute the welfare W^0 for each θ under risk neutrality ($\gamma = 0$), considering only plants with non-negative spot market profits.

2. **Welfare $W(\gamma)$:**

   - For each (γ, θ) pair, we calculate the total welfare, adjusting for the cost of guarantees.
   - Only plants included in the market (i.e., those with contract prices below the equilibrium f^* and cumulative capacity below θ) are considered.
   - Welfare is computed, for each gamma-theta pair, as the sum of winning plants:

$$W(\gamma) = \sum_i \left(q_i x \cdot \mathbb{E}(p) - C(k_i) - R_i(f^*, \gamma) \right) - \sum_i \left(\lambda \cdot q_i \cdot T^* \right)$$

3. **Baseline Welfare $W(\gamma = 0)$ for Ratio Comparisons:**

- We extract the welfare values at $\gamma = 0$ for each θ to serve as denominators in ratio calculations. **Important:** Those baseline welfares do not correspond to those compute in Section 4 this time. They refer here simply to $W(\gamma = 0)$ from this current section.

4. **Equilibrium Quantities and Prices at $\gamma = 0$:**
   - For ratio and gap analyses, we also extract the equilibrium quantities and prices at $\gamma = 0$ for each θ.

5. **Merging and Computation of Comparative Metrics:**
   - We merge all the above components into a single dataset.
   - For each scenario, we compute:
     - Welfare gain versus baseline (in million EUR)
     - Welfare ratio (as a percentage of the baseline)
     - Welfare gap (in million EUR)
     - Equilibrium quantity and price ratios (as percentages of the baseline)

```
# Welfare Dataset Creation

# STEP 1: W_0 (welfare under gamma = 0, profits spot > 0)
W_0_T <- results_with_T_G_comparison |>
  filter(gamma == 0, profits_sp_no_cpr >= 0) |>
  group_by(theta) |>
  summarise(
    welfare_0_eur = sum(x_q_exp_p_total_costs - r, na.rm = TRUE),
    .groups = "drop"
  )

# STEP 2: W_gamma for all gamma-theta combinations (adjusted for
   guarantees)
W_gamma_T <- results_with_T_G_comparison |>
  group_by(gamma, theta) |>
  filter(xf_max_cpr <= xf_equilibrium & cumulative_capacity < theta) |>
  summarise(
    x_q_exp_p_total_costs_R = sum(
      x_q_exp_p_total_costs_R
    ),
    lambda_T_q = sum(lambda_q_i_mwh_T),
    welfare_gamma_eur = x_q_exp_p_total_costs_R - lambda_T_q,
    .groups = "drop"
  ) |>
  ungroup()

# STEP 3: Baseline gamma = 0 values for comparison
W_gamma_0_T <- W_gamma_T |>
  group_by(theta) |>
  filter(gamma == 0) |>
  select(theta, welfare_gamma_eur_0 = welfare_gamma_eur) |>
  ungroup()


# STEP 4: Equilibrium quantities/prices at gamma = 0 (for ratio
   comparisons)
eq_gamma_0_vals <- equilibrium_prices_T |>
```

```
  group_by(theta, T_star) |>
  filter(gamma == 0) |>
  ungroup() |>
  select(theta,
         equilibrium_quantity_gamma_0 = equilibrium_quantity,
         equilibrium_price_gamma_0 = equilibrium_price)



welfare_dataset_G_T <- equilibrium_prices_T |>
  # Step 1: Add welfare for each gamma
  left_join(W_gamma_T, by = c("gamma", "theta")) |>

  # Step 2: Add welfare at gamma = 0 (same theta only!)
  left_join(W_gamma_0_T, by = "theta") |>

  # Step 3: Add baseline W_0 (spot profits > 0 at gamma = 0)
  left_join(W_0_T, by = "theta") |>

  # Step 4: Add equilibrium values at gamma = 0
  left_join(eq_gamma_0_vals, by = "theta") |>


  # Step 5: Compute comparative metrics
  mutate(
    equilibrium_price_eur      = equilibrium_price,
    equilibrium_quantity_mw    = equilibrium_quantity,

    W_0 = if_else(gamma == 0, welfare_0_eur, NA_real_),

    welfare_gain_vs_baseline_meur = (welfare_gamma_eur - welfare_0_eur) / 1e6,
    welfare_ratio_percent          = (welfare_gamma_eur / welfare_gamma_eur_0) * 100,
    welfare_gap_million_eur        = (welfare_gamma_eur_0 - welfare_gamma_eur) / 1e6,

    eq_quantity_ratio_percent      = (equilibrium_quantity / equilibrium_quantity_gamma_0) * 100,
    eq_price_ratio_percent         = (equilibrium_price / equilibrium_price_gamma_0) * 100
  ) |>
  select(
    gamma, theta, T_star, W_0,
    equilibrium_price_eur,
    eq_price_ratio_percent,
    equilibrium_quantity_mw,
    eq_quantity_ratio_percent,
    welfare_gamma_eur,
    welfare_ratio_percent,
    welfare_gap_million_eur,
    welfare_gain_vs_baseline_meur
  ) |>
  arrange(gamma, theta)


welfare_dataset_G_T
```

1.2.6 Welfare Comparison Between Subsidy and Guarantee Schemes

This sub-section computes and compares the welfare achieved under the public subsidy scheme (with optimal T^*) and the public guarantee scheme, for each combination of γ and θ.

Step-by-Step Explanation

1. **Aggregate Welfare under the Subsidy Scheme:**
   - For each (γ, θ, T^*) combination, we select all plants that are included in the market (*i.e.*, those with contract prices below the equilibrium f^* and cumulative capacity below θ).
   - **Important:** The `W_gamma_T` variable, which corresponds to $W(\gamma)$, is computed twice: below and in the subsection 1.2.5. The results are exactly the same, the only difference below is that we keep the variable `T_star`, since we will use it in the subsequent datasets creation.
2. **Select Welfare under the Guarantee Scheme:**
   - From the guarantee scheme dataset (`welfare_dataset_pg`), we select the welfare and buyer profit values for each (γ, θ) combination.
3. **Compute Welfare Ratios:**
   - We merge the two welfare datasets by (γ, θ).
   - For each scenario, we compute the ratio of welfare under the subsidy scheme to that under the guarantee scheme, expressed as a percentage.
   - The final dataset is arranged for easy comparison across scenarios.

```
# Compute welfare under the subsidy scheme for each (gamma, theta, T_
    star)
W_gamma_T <- results_with_T_G_comparison |>
  group_by(gamma, theta, T_star)  |>
  filter(xf_max_cpr <= xf_equilibrium & cumulative_capacity < theta) |>
  summarise(
    x_q_exp_p_total_costs_R = sum(
      x_q_exp_p_total_costs_R
    ),
    lambda_T_q = sum(lambda_q_i_mwh_T),
    welfare_gamma_eur = x_q_exp_p_total_costs_R - lambda_T_q,
    .groups = "drop"
  )

# Select welfare under the guarantee scheme
pg_welfare <- welfare_dataset_pg |>
  select(welfare_gamma_eur_pg = welfare_gamma_eur,
         buyer_profits_eur_pg, theta, gamma)

# Compute welfare ratios
welfare_ratios <- W_gamma_T |>
  left_join(pg_welfare, by = c("gamma", "theta")) |>
  mutate(
    welfare_ratio = (welfare_gamma_eur / welfare_gamma_eur_pg) * 100
  ) |>
  select(theta, gamma, T_star, everything()) |>
  arrange(theta, gamma)
```

1.2.7 Computation and Comparison of Profits for Sellers and Buyers

This section details the process for calculating and comparing the distribution of profits between sellers and buyers when the public expenditure under the subsidy scheme (with T^*) is set equal to that under the public guarantee scheme.

Step-by-Step Explanation

1. **Aggregate Seller Profits under the Subsidy Scheme:**
   - For each (γ, θ, T^*) combination, we select all plants included in the market (i.e., those with contract prices below the equilibrium f^* and cumulative capacity below θ).
   - We sum the profits from contracts under CPR for winning plants to obtain the total seller profits.
2. **Merge Seller Profits with Welfare Data:**
   - The seller profits are merged with the previously constructed welfare dataset, so that for each scenario, we have both the total welfare and the seller profit.
3. **Compute Buyer Profits and Profit Shares:**
   - Buyer profits are calculated as the difference between total welfare and seller profits.
   - The shares of welfare accruing to sellers and buyers are computed as percentages.
4. **Compare Buyer Profits Across Schemes:**
   - The results are merged with the guarantee scheme profits, and the ratio of buyer profits under the subsidy scheme to those under the guarantee scheme is calculated.

```
# Profits under public expenditure under T = public expenditure under G
profits_by_gamma_theta_T <- results_with_T_G_comparison |>
  group_by(gamma, theta, T_star) |>
  filter(xf_max_cpr <= xf_equilibrium & cumulative_capacity < theta) |>
  summarise(
    seller_profits_eur = sum(profit_cpr_contracts, na.rm = TRUE),
    .groups = "drop"
  ) |>
  ungroup() |>
  arrange(gamma, theta, T_star, seller_profits_eur)


welfare_with_profits_G_T <- welfare_dataset_G_T |>
  left_join(profits_by_gamma_theta_T, by = c("gamma", "theta", "T_star"
     ))

# Step 3: Compute profit shares for sellers and buyers
welfare_with_profits_G_T <- welfare_with_profits_G_T |>
  mutate(
    buyer_profits_eur = welfare_gamma_eur - seller_profits_eur,
    seller_profit_share_percent = round((seller_profits_eur / welfare_
        gamma_eur) * 100, 2),
    buyer_profit_share_percent  = round((buyer_profits_eur / welfare_
        gamma_eur) * 100, 2)
  ) |>
  select(
```

```
    gamma, theta, T_star,
    welfare_gamma_eur,
    seller_profits_eur,
    buyer_profits_eur,
    seller_profit_share_percent,
    buyer_profit_share_percent
  ) |>
  arrange(theta, gamma)


profits_T_G <- welfare_with_profits_G_T |>
  left_join(pg_welfare, by = c("gamma", "theta")) |>
  mutate(
    buyer_profit_ratio = (buyer_profits_eur / buyer_profits_eur_pg) *
       100
  )
```

1.2.8 Visualization of Market Outcomes Across Subsidy Levels

```r
# Plot Welfare Ratios

# Plot: Welfare Ratio (W_gamma / W_pg)
plot_profit_metric(
  data                  = welfare_ratios,
  x_var                 = "gamma",
  y_var                 = "welfare_ratio",
  group_var             = "theta",
  color_var             = "theta",
  color_label           = expression(theta),
  colors                = theme_palette_theta,
  x_label               = expression(gamma),
  y_label               = "Welfare Ratios (%)",
  y_scale_percent       = TRUE,
  save_path             = with_T_fig_path,
  file_name             = "11_welfare_ratios_public_G_T.pdf",
  legend_position       = c(0.08, 0.1)
)

#  ----------------- #
#  ----------------- #

# Plot Profits: Buyer and Seller

# Create the plot
plot_profit_metric(
  data                  = welfare_with_profits_G_T,
  x_var                 = "gamma",
  x_label               = expression(gamma),
  y_var                 = "seller_profits_eur",
  group_var             = "theta",
  color_var             = "theta",
  colors                = theme_palette_theta,
  color_label           = expression(theta),
  y_label               = "Seller Profit (MEUR)",
  y_scale_million       = TRUE,
  save_path             = with_T_fig_path,
  file_name             = "12_seller_G_T_profit_vs_gamma_theta.pdf",
  legend_position       = c(0.08, 0.90)
)


#  ----------------- #
# Profits plot (Buyers)

# Create the plot
plot_profit_metric(
  data                  = welfare_with_profits_G_T,
  x_var                 = "gamma",
  x_label               = expression(gamma),
  y_var                 = "buyer_profits_eur",
  group_var             = "theta",
  color_var             = "theta",
  colors                = theme_palette_theta,
  color_label           = expression(theta),
  y_label               = "Buyer Profit (MEUR)",
```

```r
  y_scale_million    = TRUE,
  save_path          = with_T_fig_path,
  file_name          = "13_buyer_G_T_profit_vs_gamma_theta.pdf",
  legend_position    = c(0.08, 0.1)
)


#  ----------------- #
# Profits Share (%) - Sellers

# Create the plot
plot_profit_metric(
  data               = welfare_with_profits_G_T,
  x_var              = "gamma",
  x_label            = expression(gamma),
  y_var              = "seller_profit_share_percent",
  group_var          = "theta",
  color_var          = "theta",
  colors             = theme_palette_theta,
  color_label        = expression(theta),
  y_label            = "Seller Profit Share (%)",
  y_scale_percent    = TRUE,
  save_path          = with_T_fig_path,
  file_name          = "14_seller_G_T_profit_share_vs_gamma_theta.pdf",
  legend_position    = c(0.08, 0.90)
)

#  ----------------- #
# Profits Share (%) - Buyers

# Create the plot
plot_profit_metric(
  data               = welfare_with_profits_G_T,
  x_var              = "gamma",
  x_label            = expression(gamma),
  y_var              = "buyer_profit_share_percent",
  group_var          = "theta",
  color_var          = "theta",
  colors             = theme_palette_theta,
  color_label        = expression(theta),
  y_label            = "Buyer Profit Share (%)",
  y_scale_percent    = TRUE,
  save_path          = with_T_fig_path,
  file_name          = "15_buyer_G_T_profit_share_vs_gamma_theta.pdf",
  legend_position    = c(0.08, 0.1)
)

#  ----------------- #
# Profits Ratio - Buyer Profit (T=G) / Buyer Profit (P.G.)

# Create the plot
plot_profit_metric(
  data               = profits_T_G,
  x_var              = "gamma",
  x_label            = expression(gamma),
  y_var              = "buyer_profit_ratio",
  group_var          = "theta",
  color_var          = "theta",
```

```
  color_label        = expression(theta),
  colors             = theme_palette_theta,
  y_label            = "Buyer Profit Ratio (%)",
  y_scale_percent    = TRUE,
  save_path          = with_T_fig_path,
  file_name          = "16_buyer_profit_ratio_G_T_vs_gamma_theta.pdf",
  legend_position    = c(0.08, 0.1)
)
```

Below, we describe the purpose and content of each plot generated with the `plot_profit_metric` function.

Table 1: Summary of Figures Generated for Welfare and Profit Analysis

# Plot	Description
# 11	Welfare ratio ($W(\gamma)^{T=G}/W(\gamma)^{PG}$) as a function of γ and θ
# 12	Seller profit (in million €) as a function of γ and θ
# 13	Buyer profit (in million €) as a function of γ and θ
# 14	Seller profit share (%) as a function of γ and θ
# 15	Buyer profit share (%) as a function of γ and θ
# 16	Buyer profit ratio ($W(\gamma)^{T=G}/W(\gamma)^{PG}$) as a function of γ and θ

Each of these plots provides insight into the distributional and comparative effects of the policy instruments, highlighting how profits and welfare shares shift between market participants as risk aversion and plant size vary.

1.2.9 Exporting Public Subsidies Results

```
# Save everything

# Define dataset list for comparison of T = G and Public Guarantees
datasets_T_G <- list(
  `Wind Solar Projects`          = results_with_T_G_comparison,
  `Equilibrium P. & Q.`          = equilibrium_prices_T,
  `Welfare with T star`          = welfare_dataset_G_T,
  `Welfare with Public G.`       = welfare_dataset_pg,
  `Welfare Ratios`               = welfare_ratios,
  `Profits when T=G`             = profits_T_G,
  `Profits (P.G. Case)`          = welfare_with_profits_G_T
)

# Filter all datasets by gamma
filtered_T_G <- lapply(datasets_T_G, filter_by_gamma)

# Construct dynamic filename
excel_filename_T_G <- "02_wind_solar_projects_cpr_public_expenditure_T_
    G.xlsx"
output_path_T_G <- file.path(with_T_path, excel_filename_T_G)

# Create workbook
wb_T_G <- createWorkbook()

# Add sheets dynamically
for (sheet in names(filtered_T_G)) {
  addWorksheet(wb_T_G, sheet)
```

```
  freezePane(wb_T_G, sheet, firstActiveRow = 2, firstActiveCol = 2)
  writeData(wb_T_G, sheet = sheet, x = filtered_T_G[[sheet]])
}

# Save workbook
saveWorkbook(wb_T_G, file = output_path_T_G, overwrite = TRUE)
```

- A list of relevant datasets is created, including project-level results, equilibrium prices and quantities, welfare under both schemes, profit measures, and comparative ratios.
- Each dataset is filtered by γ for consistency.
- An Excel workbook is created, with each dataset saved as a separate sheet.
- The workbook is saved to disk for convenient access and further analysis.

1.2.10 Created Tables

Table 2: Variables in `Wind Solar Projects` Sheet

Variable	Description
projectname	Project name
capacity	Installed capacity (MW)
avgcapacityfactor	Average capacity factor
type	Technology type (Wind/Solar)
hours	Annual full-load hours
power_kw	Installed power (kW)
q_i_kwh	Annual generation (kWh)
q_i_mwh	Annual generation (MWh)
v_q_i_mwh	Consumer valuation (MWh)
c_inv	Investment cost per MW
c_om	O&M cost per MW
total_cost	Total cost
avg_cost_euro_kwh	Average cost (EUR/kWh)
avg_cost_euro_mwh	Average cost (EUR/MWh)
r_0	r_0
r	r_i
f_upper	Second root f_{upper}, if any
f_upper_message	Says if `f_upper`$>\mathbb{E}(p)$ or `f_upper`$\leq \mathbb{E}(p)$
xf_upper	Second root xf_{upper}, if any
profits_sp_no_cpr	Π_S^0
theta	Demand scenario θ (MW)
gamma	Share of opportunistic buyers γ
lambda	Per-unit Cost of Social Funds λ
T_star	Optimal T^* to have $T = G$
f_c_cpr	f_c
f_spot_cpr	f_{spot}
f_max_cpr	f_{max}
f_equilibrium	f^*
R_f_equilibrium_cpr	$R_i(f^*, \gamma)$
xf_c_cpr	xf_c
xf_spot_cpr	xf_{spot}
xf_max_cpr	xf_{max}
xf_equilibrium	xf^*
equilibrium_quantity	q^*
x_q_exp_p_total_costs	$xq_i \cdot \mathbb{E}(p) - C(k_i)$
x_q_exp_p_total_costs_R	$xq_i \cdot \mathbb{E}(p) - C(k_i) - R_i(f^*, \gamma)$
cumulative_production	Cumulative energy production (MWh)
cumulative_capacity	Cumulative capacity (MW)
profit_cpr_contracts	Profits under CPR contracts, using f^*
x_integral	$x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp$ (public guarantees case)
lambda_q_i_mwh_T	$\lambda \cdot q_i \cdot T^*$
q_i_mwh_lambda_gamma_x_integral	$q_i \cdot \lambda \cdot \gamma \cdot x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp$
q_i_mwh_T	$q_i \cdot T^*$
q_i_mwh_gamma_x_integral	$q_i \cdot \gamma \cdot x \cdot \int_0^{f^*} (f^* - p)\, \phi(p)\, dp$

Table 3: Variables in `Equilibrium P. & Q.` Sheet

Variable	Description
gamma	Share of opportunistic buyers γ
T_star	Optimal subsidy T^*
theta	Demand scenario (MW) - θ
equilibrium_quantity	Total contracted quantity at equilibrium (MW), q^*
equilibrium_price	Equilibrium contract price (EUR/MWh)

Table 4: Variables in `Welfare with T star` Sheet

Variable	Description
gamma	Share of opportunistic buyers γ
theta	Demand scenario (MW) - θ
T_star	Optimal subsidy T^*
equilibrium_price_eur	Equilibrium contract price (EUR/MWh)
eq_price_ratio_percent	Price as % of $\gamma = 0$ baseline
equilibrium_quantity_mw	Equilibrium contracted quantity (MW)
eq_quantity_ratio_percent	Quantity as % of $\gamma = 0$ baseline
W_0	Welfare W^0
welfare_gamma_eur	Welfare $W^{T=G}$ at given γ and θ (EUR)
welfare_ratio_percent	Welfare as % of $W^{T=G}(\gamma = 0)$ baseline
welfare_gap_million_eur	Welfare gap vs. baseline $\left[W^{T=G}(\gamma = 0) - W^{T=G}(\gamma)\right]$ (million EUR)
welfare_gain_vs_baseline_meur	Welfare gain vs. baseline $\left[W^{T=G}(\gamma) - W^0\right]$ (million EUR)

Table 5: Variables in `Welfare with Public G.` Sheet - Public Guarantees Case

Variable	Description
theta	Demand scenario (MW) - θ
gamma	Share of opportunistic buyers γ
equilibrium_price_eur	Equilibrium contract price (EUR/MWh)
eq_price_ratio_percent	Price as % of $\gamma = 0$ baseline
equilibrium_quantity_mw	Equilibrium contracted quantity (MW)
W_0	Welfare at $\gamma = 0$ (EUR)
welfare_gamma_eur	Welfare W^{PG} at given γ and θ (EUR)
welfare_ratio_percent	Welfare as % of $W^{\mathrm{PG}}(\gamma = 0)$ baseline
welfare_gap_million_eur	Welfare gap vs. baseline $\left[W^{\mathrm{PG}}(\gamma = 0) - W^{\mathrm{PG}}(\gamma)\right]$ (million EUR)
welfare_gain_vs_baseline_meur	Welfare gain vs. baseline $\left[W^{\mathrm{PG}}(\gamma) - W^0\right]$ (million EUR)
seller_profits_eur	Seller profits under public guarantee (EUR)
buyer_profits_eur	Buyer profits under public guarantee (EUR)
buyer_profit_share_percent	Buyer share of welfare (%)
seller_profit_share_percent	Seller share of welfare (%)

Table 6: Variables in `Welfare Ratios` Sheet

Variable	Description
theta	Demand scenario (MW) - θ
gamma	Share of opportunistic buyers γ
T_star	Optimal subsidy T^*
x_q_exp_p_total_costs_R	$\sum_i xq_i \cdot \mathbb{E}(p) - C(k_i) - R_i(f^*, \gamma)$ under $T^* = G$ case
lambda_T_q	$\lambda \cdot T^* \cdot \sum_i q_i$
welfare_gamma_eur	Welfare (EUR), with public subsidies $T^* = G$
welfare_gamma_eur_pg	Welfare (EUR), with public guarantees (PG - Section 5)
welfare_ratio	Welfare ratio (%): $W(\gamma)^{T=G} / W(\gamma)^{PG}$
buyer_profits_eur_pg	Buyer profits under public guarantee (EUR)
buyer_profit_ratio	Buyer profit ratio (%): $(\text{Buyer Profits}_{T=G} / \text{Buyer Profits}_{PG}) \times 100$

Table 7: Variables in `Profits when T=G` Sheet

Variable	Description
gamma	Share of opportunistic buyers γ
theta	Demand scenario (MW) - θ
T_star	Optimal subsidy T^*
welfare_gamma_eur	Welfare (EUR), under subsidies T^*
seller_profits_eur	Seller profits under subsidies T^* (EUR)
buyer_profits_eur	Buyer profits under subsidies T^* (EUR)
seller_profit_share_percent	Seller share of welfare (%)
buyer_profit_share_percent	Buyer share of welfare (%)