Commodity Market

Chuyang Yu

2025-03-22

```
knitr::opts_chunk$set(echo = TRUE)
```

#spot price scenario

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

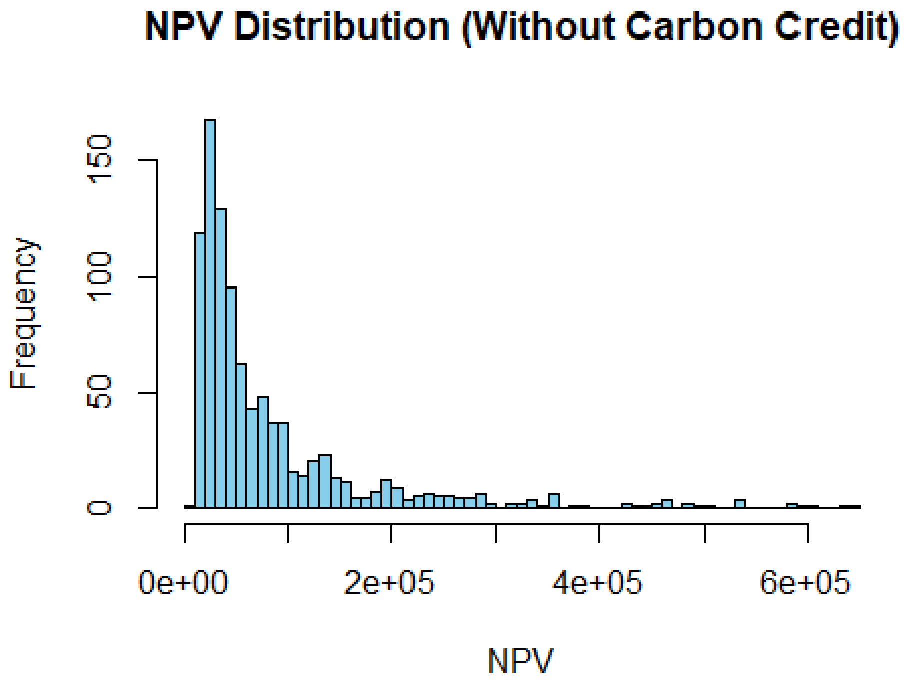
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
#electricity sales revenue

germany_spot<-readxl::read_excel("ycy price.xlsx",sheet = "germany
spot") %>%
  select(Date,`Last Price`) %>%
  na.omit()

germany_spot<-germany_spot %>%
  filter(`Last Price`>0)
```

```
#log return

germany_spot <- germany_spot %>%
  mutate(log_return = log(`Last Price` / lag(`Last Price`)))

miu<-mean(germany_spot$log_return, na.rm = TRUE)

sigma<-sd(tail(germany_spot$log_return, 60), na.rm = TRUE)

miu_mo<-miu*(252/12)

sigma_mo<-sigma*sqrt(252/12)

miu
```

```
## [1] 0.0001079593

sigma

## [1] 0.5543847

miu_mo

## [1] 0.002267145

sigma_mo

## [1] 2.54051

#monte carlo
set.seed(790)

n_simu<-1000

n_mo<-12*10

initial_price <- tail(germany_spot$`Last Price`, 1)

dt<-1/12

simulate_gbm <- function(S0, miu_mo, sigma_mo, dt, steps, n_sim) {
  paths <- matrix(NA, nrow = steps, ncol = n_simu)
  paths[1, ] <- S0
  for (i in 2:steps) {
    paths[i, ] <- paths[i-1, ] * exp((miu_mo - 0.5 * sigma_mo^2) * dt +
sigma_mo * sqrt(dt) * rnorm(n_simu))
  }
  return(paths)
}

simu_price<-simulate_gbm(initial_price, miu_mo, sigma_mo, dt, n_mo,
n_simu)

#electricity sales revenue

electr_rev<-simu_price*500

discount_rate<-0.02/12

npv_electr<-colSums(electr_rev / ((1 + discount_rate) ^ (1:n_mo)))

npv_electr <- npv_electr[npv_electr < quantile(npv_electr, 0.95)]

hist(npv_electr, breaks = 50, main = "NPV Distribution (Without Carbon
Credit)", xlab = "NPV", col = "skyblue")
```



```
#carbon revenue

carbon_spot<-readxl::read_excel("ycy price.xlsx",sheet = "EUA SPOT")
%>%
  select(Date,`Last Price`) %>%
  na.omit()

carbon_spot<-carbon_spot %>%
  filter(`Last Price`>0)

#log return

carbon_spot <- carbon_spot %>%
  mutate(log_return = log(`Last Price` / lag(`Last Price`)))

miuC <-mean(carbon_spot$log_return, na.rm = TRUE)

sigmaC<-sd(tail(carbon_spot$log_return, 60), na.rm = TRUE)

miu_Cmo<-miu*(252/12)

sigma_Cmo<-sigma*sqrt(252/12)*0.5

miuC

## [1] -0.0007016953
```

```
sigmaC
```

```
## [1] 0.02197787
```

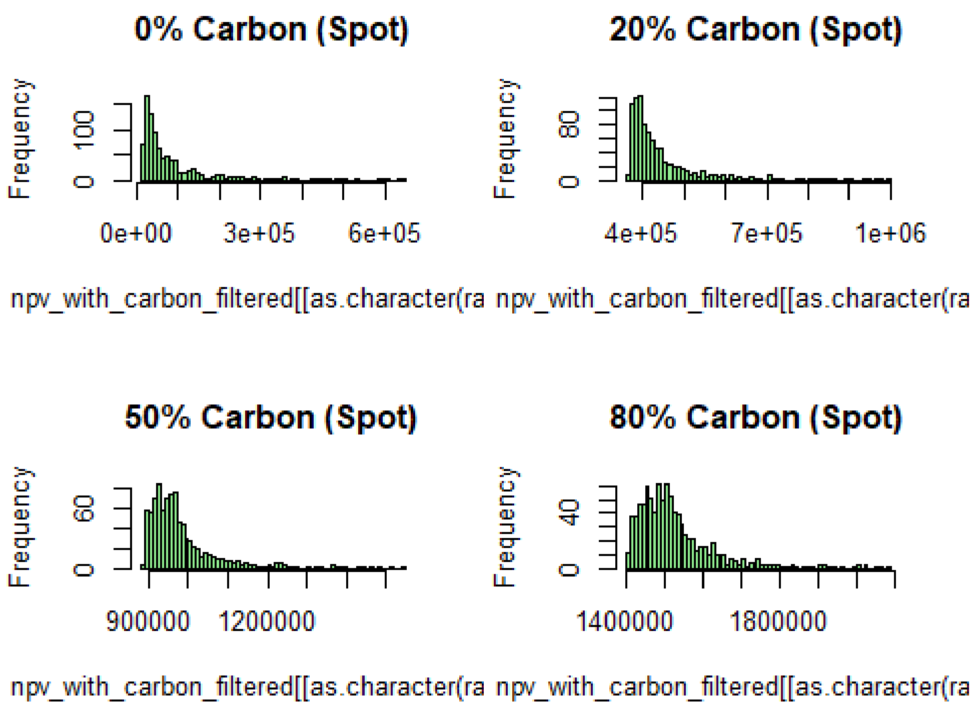
```
miu_Cmo
```

```
## [1] 0.002267145
```

```
sigma_Cmo
```

```
## [1] 1.270255
```

```
#monte carlo for carbon credit

eua_price <- carbon_spot$`Last Price`[nrow(carbon_spot)]

simuC <- matrix(NA, ncol = n_simu, nrow = n_mo)

for (i in 1:n_simu) {
  simuC[, i] <- eua_price*exp((miu_Cmo - 0.5 *
sigma_Cmo^2)*dt+sigma_Cmo * sqrt(dt) * rnorm(n_mo))
}

#carbon credit revenue

#assume our allocation rate are 20%, 50%, 80% to test different impacts

allo_rates<-c(0,0.2,0.5,0.8)

npv_with_carbon <- list()

for (rate in allo_rates) {
  carbon_rev <- simuC * (500 * rate)
  npv_with_carbon[[as.character(rate)]] <- colSums((electr_rev +
carbon_rev) / ((1 + discount_rate) ^ (1:n_mo)))
}

#remove outliers
npv_with_carbon_filtered <- lapply(npv_with_carbon, function(npv) {
  npv[npv > quantile(npv, 0.05) & npv < quantile(npv, 0.95)]
})

#total revenue with carbon credit in spot price

par(mfrow = c(2, 2))

for (rate in allo_rates) {
  hist(npv_with_carbon_filtered[[as.character(rate)]], breaks = 50,
main = paste0(rate * 100, "% Carbon (Spot)"), col = "lightgreen")
}
```



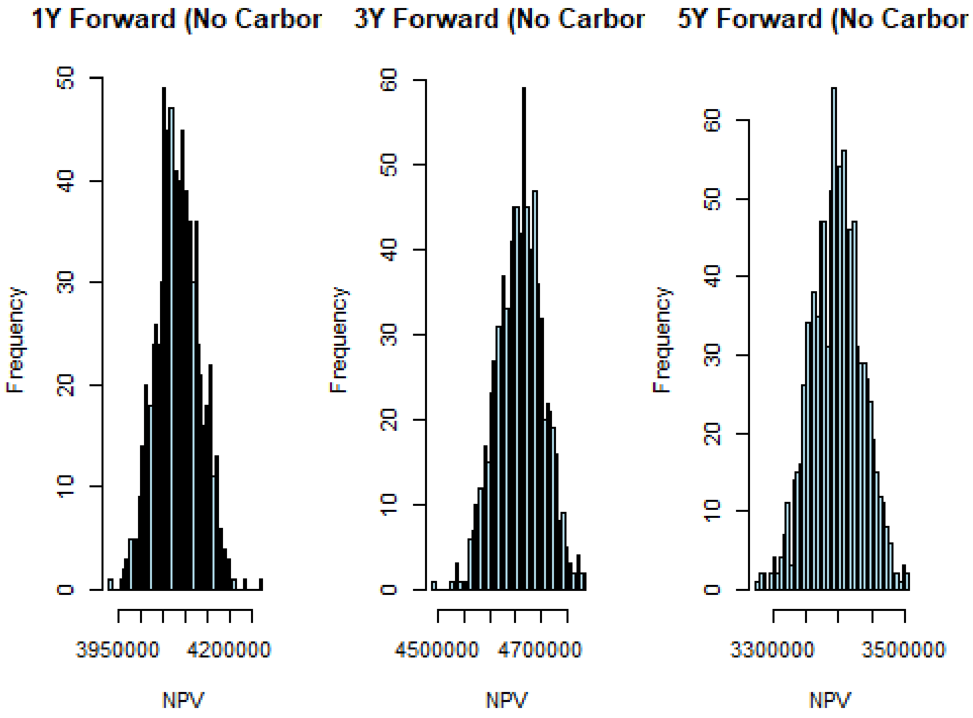
#Forward Contract Scenario

```
#electricity without carbon credit

library(tidyverse)

## ── Attaching core tidyverse packages ────────────────────────
tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.4     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.1
## ✔ readr     2.1.5
## ── Conflicts ─────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to
force all conflicts to become errors

library(readxl)

germany_forward <- list(
  "1Y" = read_excel("ycy price.xlsx", sheet = "germany yr1") %>%
    mutate(Date = as.Date(Date)) %>%
    select(Date, `Last Price`) %>%
    arrange(Date) %>%
```

```r
    na.omit(),

  "3Y" = read_excel("ycy price.xlsx", sheet = "germany yr3") %>%
    mutate(Date = as.Date(Date)) %>%
    select(Date, `Last Price`) %>%
    arrange(Date) %>%
    na.omit(),

  "5Y" = read_excel("ycy price.xlsx", sheet = "germany yr5") %>%
    mutate(Date = as.Date(Date)) %>%
    select(Date, `Last Price`) %>%
    arrange(Date) %>%
    na.omit()
)

contract_length <- c("1Y" = 12, "3Y" = 36, "5Y" = 60)

generation <- 500

discount_rate <- 0.02 / 12

n_simu <- 1000

n_mo <- 120

npv_forward_without_carbon <- list()

set.seed(42)

simu_price2 <- matrix(rnorm(n_mo * n_simu, mean = 80, sd = 10), nrow =
n_mo, ncol = n_simu)

for (contract in names(contract_length)) {

  contract_months <- contract_length[[contract]]

  electricity_price <- germany_forward[[contract]]$`Last Price`

  contract_rev <- matrix(rep(electricity_price[1:contract_months],
n_simu),
                         nrow = contract_months,
                         ncol = n_simu,
                         byrow = TRUE) * generation

  post_contract_prices <- simu_price2[(contract_months + 1):n_mo, ]

  post_contract_rev <- post_contract_prices * generation
  total_rev_without_carbon <- rbind(contract_rev, post_contract_rev)

  npv_forward_without_carbon[[contract]] <-
colSums(total_rev_without_carbon / ((1 + discount_rate) ^ (1:n_mo)))

  print(paste("NPV calculated for contract:", contract))
}
```

```
## [1] "NPV calculated for contract: 1Y"
## [1] "NPV calculated for contract: 3Y"
## [1] "NPV calculated for contract: 5Y"
```

```
par(mfrow=c(1,3))
for (contract in names(contract_length)) {
  hist(unlist(npv_forward_without_carbon[[contract]]),
       breaks = 50,
       main = paste0(contract, " Forward (No Carbon)"),
       xlab = "NPV",
       col = "lightblue")
}
```



```
#with carbon credit

carbon_forward <- list(
  "1Y" = read_excel("ycy price.xlsx", sheet = "MOZ25") %>%
select(Date...1, `Last Price`) %>% na.omit(),
  "3Y" = read_excel("ycy price.xlsx", sheet = "MOZ27") %>% select(Date,
`Last Price`) %>% na.omit(),
  "5Y" = read_excel("ycy price.xlsx", sheet = "MOZ29") %>% select(Date,
`Last Price`) %>% na.omit()
)
```

```
## New names:
## New names:
## New names:
## • `Date` -> `Date...1`
## • `` -> `...5`
## • `` -> `...7`
## • `` -> `...8`
## • `Date` -> `Date...9`
```

```
allo_rates <- c(0, 0.2, 0.5, 0.8)

npv_forward_with_carbon <- list()

for (contract in names(contract_length)) {
  contract_months <- contract_length[[contract]]

  electricity_price <- germany_forward[[contract]]$`Last Price`

  carbon_price <- carbon_forward[[contract]]$`Last Price`

  contract_rev <- matrix(rep(electricity_price, n_simu),
                         nrow = contract_months,
                         ncol = n_simu,
                         byrow = TRUE) * generation

  post_contract_prices <- simu_price2[(contract_months + 1):n_mo, ]

  post_contract_rev <- post_contract_prices * generation

  for (alloc in allo_rates) {

    contract_carbon_rev <- matrix(rep(carbon_price * alloc, n_simu),
                                  nrow = contract_months, ncol =
n_simu, byrow = TRUE) * generation

    post_contract_carbon_rev <- post_contract_prices * (alloc *
carbon_price) * generation

    total_rev_with_carbon <- rbind(contract_rev + contract_carbon_rev,
                                   post_contract_rev
+post_contract_carbon_rev)

    npv_forward_with_carbon[[paste0(contract, "_alloc_", alloc)]] <-
```

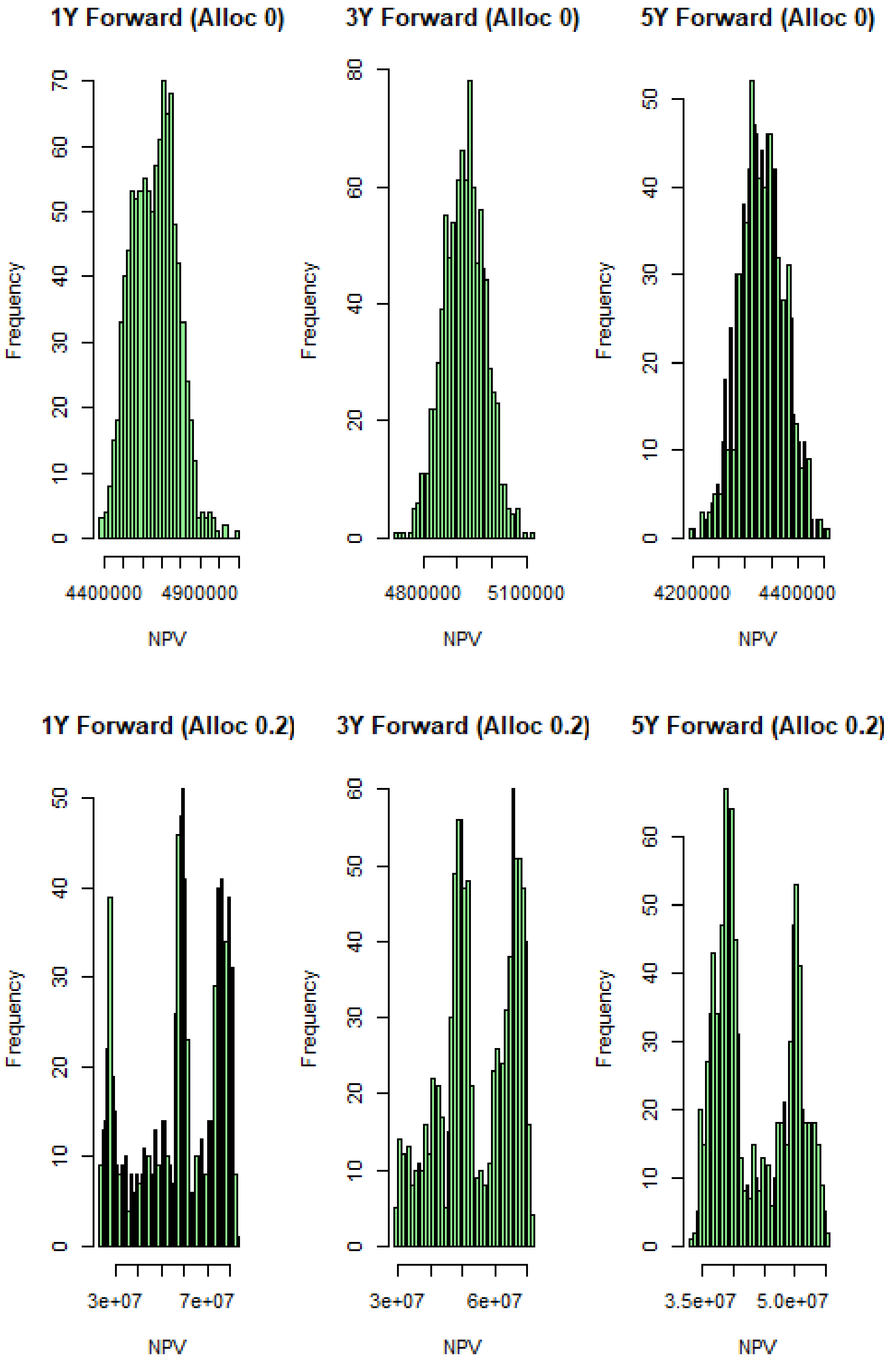
```
      colSums(total_rev_with_carbon / ((1 + discount_rate) ^ (1:n_mo)))
  }
}
```

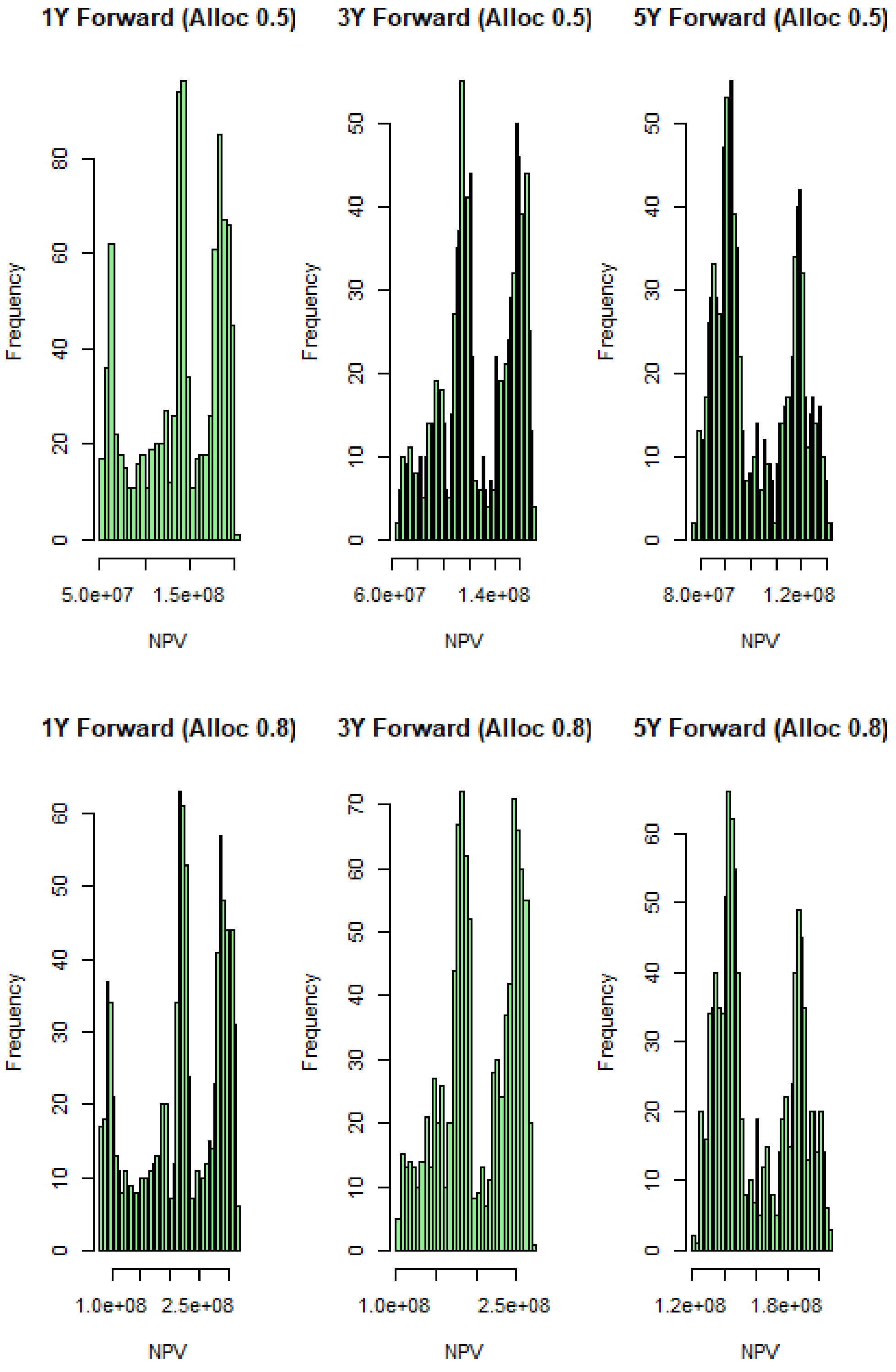
```
## Warning in matrix(rep(electricity_price, n_simu), nrow =
contract_months, :
## data length [1235000] is not a sub-multiple or multiple of the
number of rows
## [12]

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1304000] is not a sub-multiple or multiple of the
number of rows
## [12]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1304000] is not a sub-multiple or multiple of the
number of rows
## [12]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1304000] is not a sub-multiple or multiple of the
number of rows
## [12]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1304000] is not a sub-multiple or multiple of the
number of rows
## [12]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(electricity_price, n_simu), nrow =
contract_months, :
```

```
## data length [542000] is not a sub-multiple or multiple of the number
of rows
## [36]

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1100000] is not a sub-multiple or multiple of the
number of rows
## [36]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1100000] is not a sub-multiple or multiple of the
number of rows
## [36]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1100000] is not a sub-multiple or multiple of the
number of rows
## [36]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [1100000] is not a sub-multiple or multiple of the
number of rows
## [36]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(electricity_price, n_simu), nrow =
contract_months, :
## data length differs from size of matrix: [1254000 != 60 x 1000]

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [580000] is not a sub-multiple or multiple of the number
```

```
of rows
## [60]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [580000] is not a sub-multiple or multiple of the number
of rows
## [60]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [580000] is not a sub-multiple or multiple of the number
of rows
## [60]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length

## Warning in matrix(rep(carbon_price * alloc, n_simu), nrow =
contract_months, :
## data length [580000] is not a sub-multiple or multiple of the number
of rows
## [60]

## Warning in post_contract_prices * (alloc * carbon_price): longer
object length
## is not a multiple of shorter object length
```

```r
for (alloc in allo_rates) {

  par(mfrow=c(1,3))

  for (contract in names(contract_length)) {
    hist(unlist(npv_forward_with_carbon[[paste0(contract, "_alloc_",
alloc)]]),
         breaks = 50,
         main = paste0(contract, " Forward (Alloc ", alloc, ")"),
         xlab = "NPV",
         col = "lightgreen")
  }
}
```

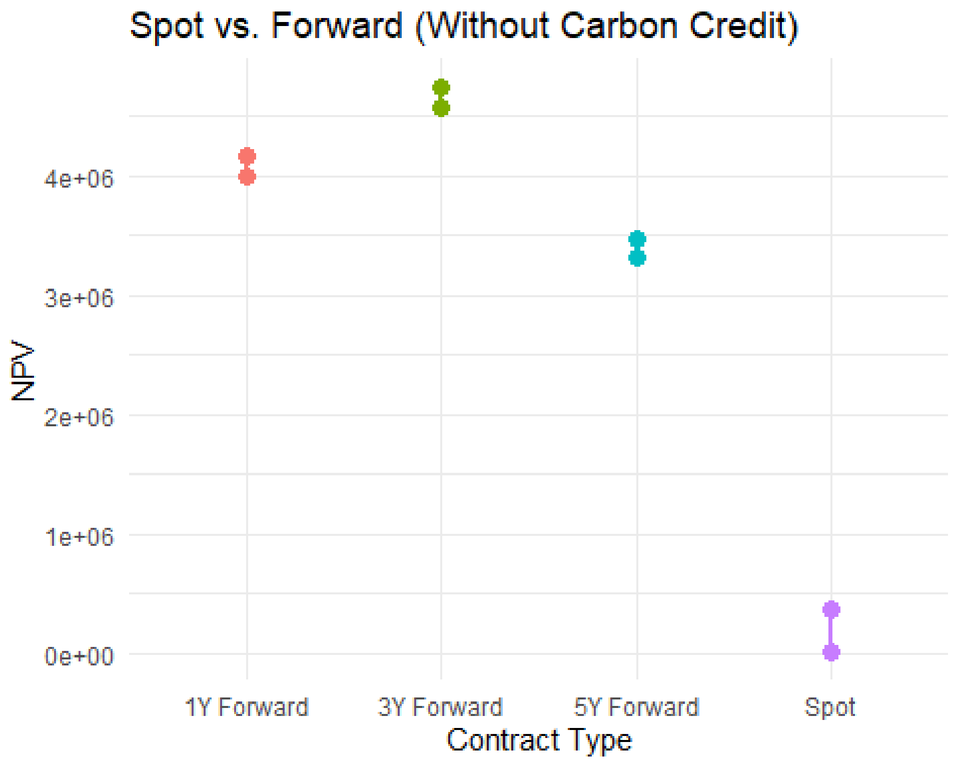

1Y Forward (Alloc 0)
3Y Forward (Alloc 0)
5Y Forward (Alloc 0)
Frequency
0
10
20
30
40
50
60
70
4400000
4900000
NPV
0
20
40
60
80
4800000
5100000
NPV
0
10
20
30
40
50
4200000
4400000
NPV
1Y Forward (Alloc 0.2)
3Y Forward (Alloc 0.2)
5Y Forward (Alloc 0.2)
0
10
20
30
40
50
3e+07
7e+07
NPV
0
10
20
30
40
50
60
3e+07
6e+07
NPV
0
10
20
30
40
50
60
3.5e+07
5.0e+07
NPV



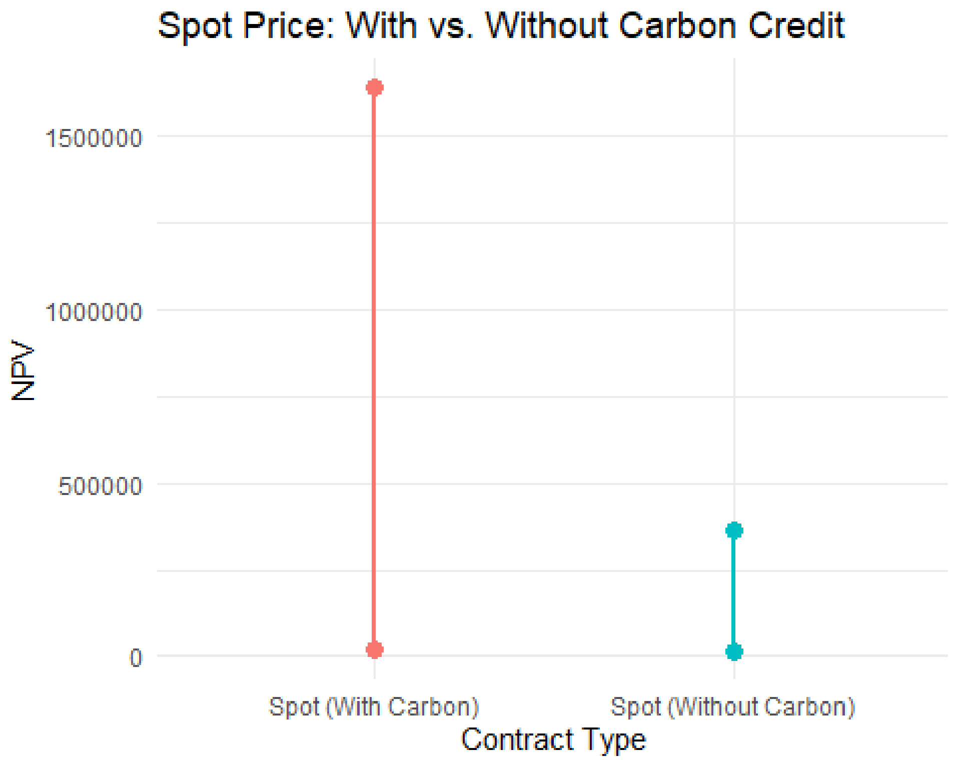
#Spot vs. Forward

```
library(ggplot2)
```

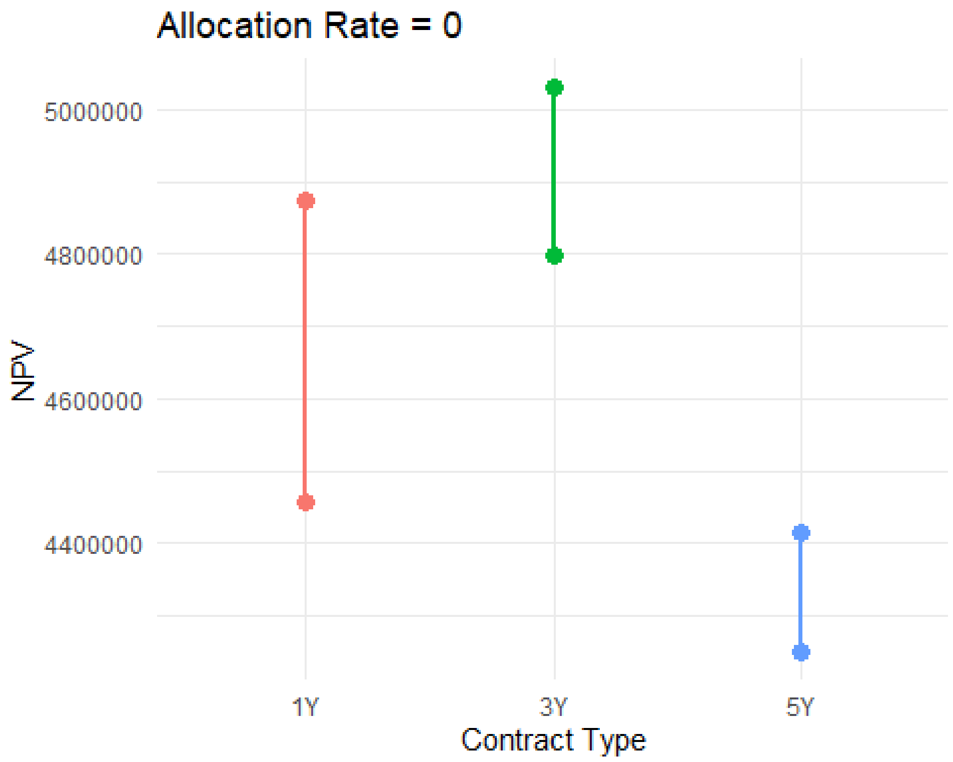
```r
library(dplyr)

plot_confidence_interval <- function(data_list, title) {
  df <- data.frame(
    Contract = rep(names(data_list), each = 2),
    Type = rep(c("Lower", "Upper"), times = length(data_list)),
    NPV = unlist(lapply(data_list, function(x) quantile(x, probs = 
c(0.025, 0.975))))
  )

  ggplot(df, aes(x = Contract, y = NPV, color = Contract)) +
    geom_point(size = 3) +
    geom_line(aes(group = Contract), linewidth = 1) +
    labs(title = title, y = "NPV", x = "Contract Type") +
    theme_minimal() +
    theme(legend.position = "none")
}

#without carbon credit

without_carbon_data <- list(
  "Spot" = npv_electr,
  "1Y Forward" = npv_forward_without_carbon[["1Y"]],
  "3Y Forward" = npv_forward_without_carbon[["3Y"]],
  "5Y Forward" = npv_forward_without_carbon[["5Y"]]
)

p1<-plot_confidence_interval(without_carbon_data, "Spot vs. Forward 
(Without Carbon Credit)")

print(p1)
```



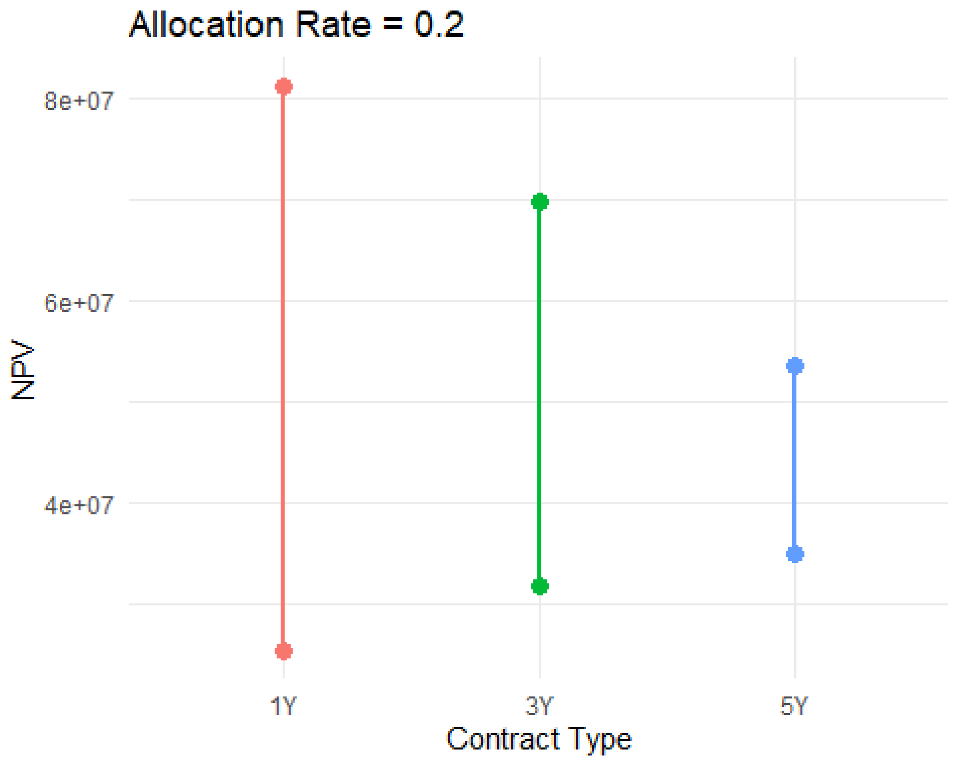
```
str(npv_electr)

##  num [1:950] 21056 23585 93813 155236 28654 ...

str(npv_with_carbon_filtered)

## List of 4
##  $ 0  : num [1:900] 21056 23585 93813 155236 28654 ...
##  $ 0.2: num [1:900] 393233 444946 521016 398012 505246 ...
##  $ 0.5: num [1:900] 889982 947705 971646 1069687 952049 ...
##  $ 0.8: num [1:900] 1411337 1502176 1498346 1618358 1506086 ...

#with carbon credit for spot

spot_carbon_data <- list(
  "Spot (Without Carbon)" = npv_electr,
  "Spot (With Carbon)" = unlist(npv_with_carbon_filtered)
)

p2 <- plot_confidence_interval(spot_carbon_data,
                               "Spot Price: With vs. Without Carbon
Credit")

print(p2)
```



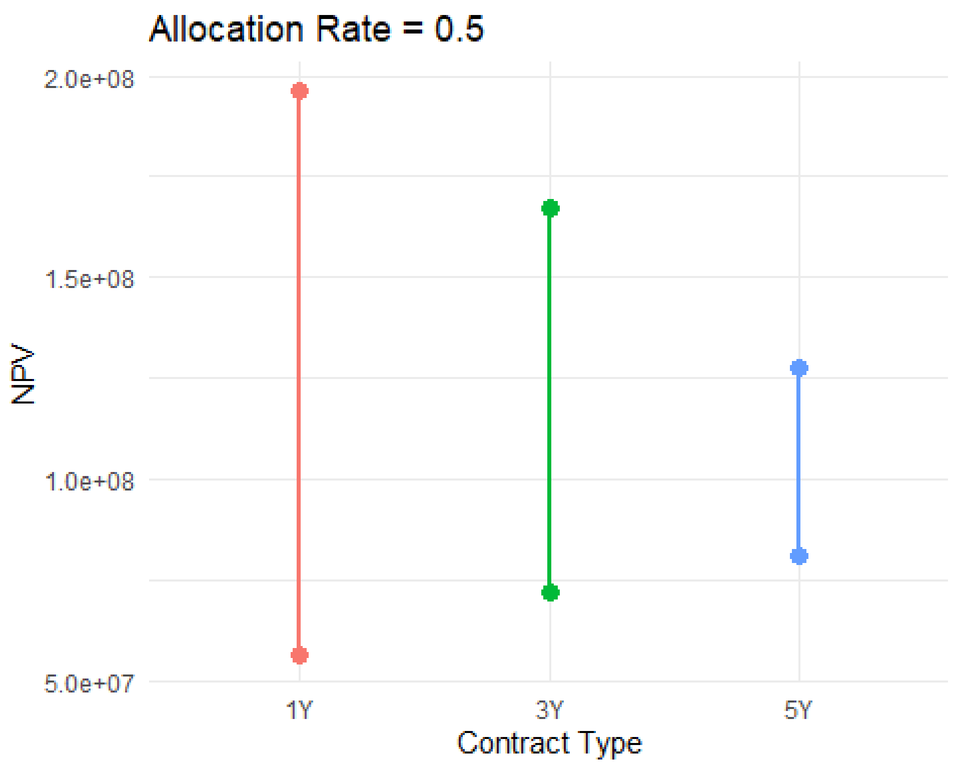
```r
#different allocation rates have impact on forward

allocation_plots <- list()

for (alloc in allo_rates) {
  alloc_data <- list(
    "1Y" = npv_forward_with_carbon[[paste0("1Y_alloc_", alloc)]],
    "3Y" = npv_forward_with_carbon[[paste0("3Y_alloc_", alloc)]],
    "5Y" = npv_forward_with_carbon[[paste0("5Y_alloc_", alloc)]]
  )

  allocation_plots[[paste0("Alloc_", alloc)]] <-
plot_confidence_interval(alloc_data, paste("Allocation Rate =", alloc))
}

print(allocation_plots)

## $Alloc_0
```


Allocation Rate = 0
5000000
4800000
4600000
4400000
NPV
1Y
3Y
5Y
Contract Type

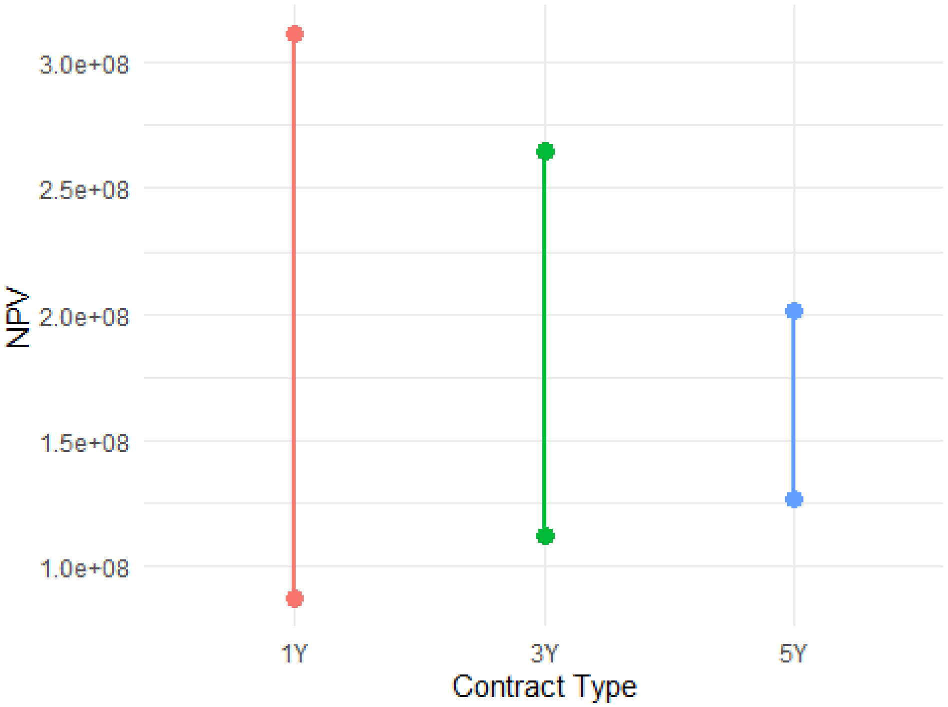
```
##
## $Alloc_0.2
```


Allocation Rate = 0.2
8e+07
6e+07
4e+07
NPV
1Y
3Y
5Y
Contract Type

```
##
## $Alloc_0.5
```



```
##
## $Alloc_0.8
```



Allocation Rate = 0.8