An Oifig Buiséid Pharlaiminteach
Parliamentary Budget Office

Tithe an Oireachtais
Houses of the Oireachtas

# Working Paper Series

No. 1 of 2022

## Simulating Micro Data for Policy and Costing Analysis

Akisato Suzuki†

July 2022

† The author Akisato Suzuki is an Irish Government Economic and Evaluation Service (IGEES) policy analyst / economist in the Parliamentary Budget Office (PBO), Houses of the Oireachtas. PBO Working Papers present primary research in progress intended to elicit comments and encourage debate. The content is subject to review and revision. The analysis and views contained in this paper are those of the author only, and are not necessarily reflective of the position of the PBO or of the Houses of the Oireachtas generally. For queries, contact Akisato.Suzuki@oireachtas.ie.

# Simulating Micro Data for Policy and Costing Analysis

Akisato Suzuki
Parliamentary Budget Office
Leinster House, Kildare Street
Dublin 2, D02 XR20, Ireland
akisato.suzuki@oireachtas.ie

12 July 2022

**Abstract**

This paper aims to share the PBO's methodological approaches and expertise in using statistical techniques to simulate micro data for policy and costing analysis. While micro data are often essential for rigorous costing analysis, such data might not always be available or, if available, might not necessarily be representative of the target population of a policy. A potential strategy to tackle these issues is to simulate micro data using statistical techniques. As an example, this paper simulates micro data on earnings and validates the simulated data based on actual aggregated earnings data from the Central Statistics Office of Ireland (CSO). It also uses the simulated micro data to cost a hypothetical earnings-based welfare benefit and presents both mean and interval estimates to capture uncertainty. The paper concludes that the statistical simulation techniques outlined here are a useful addition to policy analysts' toolkit.

*Keywords* – simulation, costing, public policy, statistics

# Introduction

The PBO conducts analyses of the financial implications of policy proposals.[1] One of the common challenges faced is the availability and reliability of necessary micro data (i.e., data at the individual unit level). First, there might exist no relevant or usable micro data in the first place. Second, confidentiality and/or time constraints might make it impossible to access micro data in a timely manner. Third, there might be uncertainty on whether the available micro data is representative of the target population of a policy. Despite these issues, micro data are often essential for rigorous costing analysis. Using only aggregated data, is it oftentimes difficult to examine how much a proposed policy will cost in total, when the cost per person differs significantly across individuals.

A potential strategy to tackle these issues related to micro data is to simulate micro data using statistical techniques. This paper aims to share, in an open and transparent manner, the PBO's methodological approaches and expertise in using statistical simulation techniques for policy and costing analysis.

The paper first gives an overview of what theoretical statistical distributions are; theoretical distributions are used to simulate data. Then, as an example, it simulates micro data on earnings and validates the simulated data based on actual aggregated earnings data from the Central Statistics Office of Ireland (CSO). Finally, the paper uses the simulated micro data on earnings to cost a hypothetical earnings-based welfare benefit. All analyses were done on the statistical software, R (the R code used is available in the appendix).[2]

# An Overview of Theoretical Data Distributions

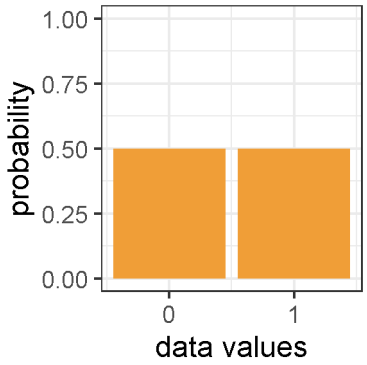
Theoretical statistical data distributions are defined based on specific theoretical properties, usually governed by a number of parameters. Different theoretical distributions are used to express various processes whereby data are generated. For example, the outcome of coin flipping can only be two values, either head or tail, while a measure of economic development (e.g., GDP per capita) can take a potentially infinite number of values.

The outcome of coin flipping, or anything that expresses two distinctive situations only (e.g., the presence or absence of a public policy across countries), can be modelled by the Bernoulli distribution. The Bernoulli distribution expresses two distinctive outcomes by values of 0 and 1, and has only one parameter $p$, which is the probability of one of these outcomes occurring (the probability of the other outcome occurring is simply $1 - p$). Figure 1 presents an example of the Bernoulli distribution where each outcome has an equal probability of occurrence ($p = 0.5$).

[1] For our costing service, see Parliamentary Budget Office, "Policy Costing Service Guidelines," 2022, https://data.oireachtas.ie/ie/oireachtas/parliamentaryBudgetOffice/2022/2022-06-15_policy-costing-service-guidelines_en.pdf.

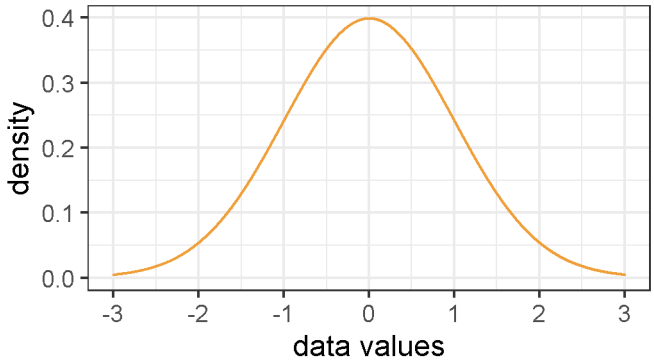
[2] R Core Team, "R: A Language and Environment for Statistical Computing," 2022, https://www.R-project.org.

Figure 1: Bernoulli distribution with the 50% probability




Another well-known theoretical distribution is the normal distribution. It is governed by two parameters: the mean and standard deviation of the distribution. Figure 2 displays a normal distribution with the mean of 0 and the standard deviation of 1. With these parameter values, the distribution is often referred to as "standard normal distribution." The standard deviation measures how spread data values are around the mean value. In the normal distribution, approximately 68% of data fall within the range of $\pm$ 1 standard deviation (between $-1$ and 1 in Figure 2); approximately 95% of data fall within the range of $\pm$ 2 standard deviations (between $-2$ and 2 in Figure 2).

Figure 2: Standard normal distribution



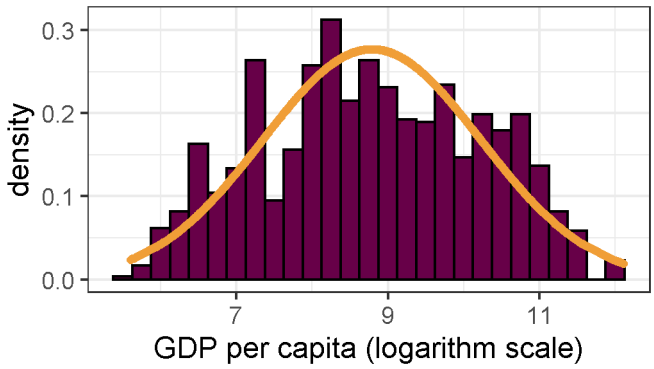

Note: In a simplified (but therefore imprecise) term, probability density captures how "frequent" each value is.

A real-world example of a variable whose data distribution tends to be well approximated by a normal distribution is the logarithm of real GDP per capita.[3] The purple bars of Figure 3 are (log) real GDP per capita data values across the world from 2015 to 2020. The mean of this

[3] The logarithm of GDP per capita is used largely for statistical modelling and forecasting as it stabilises the variability.

normal distribution is approximately 8.79 and the standard deviation is approximately 1.44.[4]. Its overall shape is normal, although it is not as "neatly" shaped as the theoretical normal distribution constructed by the above mean and standard deviation values (as presented by the orange line). Usually, the distribution of actual data is not exactly the same as a theoretical distribution. In this sense, a theoretical distribution might be considered as an idealised version.

Figure 3: GDP per capita on the logarithm scale across the world from 2015 to 2020




Source: Author's own calculation based on data from the World Bank Group.[5]
Note: The purple bars are actual data; the orange line is a constructed theoretical distribution.

From a different perspective, actual data can be considered a sample from a theoretical distribution. The size (i.e., the number of observations) of actual data is usually "finite" or limited by constraints on measurement. Meanwhile, a theoretical distribution is a data distribution whose size is infinite. Therefore, the distribution of actual data can be seen as an approximation of a theoretical distribution. In addition, just because data from a certain time period of the past have a slightly different shape from any theoretical distributions, it might not necessarily mean that the actual data do not follow any of the theoretical distributions. It might be because the specific sample or realisation of data happens to be less reflective of the underlying (unknown) true distribution of the data that follows a certain theoretical distribution.

Whichever theoretical position one may take, simulating micro data based on a theoretical distribution can be useful for policy and costing analysis to overcome data issues and limitations. The next section gives an example of simulated micro data.

[4] Note that, because these numbers are on the logarithm scale and not on the original constant US dollar scale, we cannot directly interpret them in a substantively meaningful way.

[5] World Bank Group, "World Development Indicators," 2022, https://databank.worldbank.org/source/world-development-indicators.

# Example of Micro Data Simulation

As an example, we simulate micro data for weekly (gross) earnings. The Central Statistics Office of Ireland publishes percentiles of the weekly gross earnings distribution[6]; the latest version is from the 2020 administrative data.[7] The percentile data suggest that the underlying micro data are distributed in a way similar to a log-normal distribution. The log-normal distribution takes only positive values or zero and can have a fat tail on the right side of the distribution (i.e., a large share of data takes smaller values than the average while a small share takes very large values). The left panel of Figure 4 is an example of the log-normal distribution and presents the simulated micro data on weekly earnings.[8] The shape of the log-normal distribution was determined as follows:

- The CSO reports that the average weekly earnings are €801.
- The expected value of a log-normal distribution of the variable $X$ is defined as: $E(X) = \exp\left(\mu + \frac{1}{2}\sigma^2\right)$, where $\mu$ is the mean parameter and $\sigma$ is the standard deviation parameter. If $E(X) = 801$ and either of the two parameters, $\mu$ and $\sigma$, is set at a particular value, the value of the remaining parameter can be determined.
- To this end, a value for the standard deviation parameter was identified using an iterative process (i.e., by trial and error) such that, for chosen values of the mean and standard deviation, the simulated percentiles approximate the CSO data well (when visually assessed).
- We settled on the standard deviation, $\sigma$, to be $\log(2.1)$, which then determines a value for the mean, $\mu$, as $\log(608.274)$.
- The simulated values were rounded up to two digits.

We generated a total of 2,222,500 observations, the number equal to the number of people in employment in 2020.[9] A comparison between the percentiles generated from the simulated micro data and those published by the CSO data is presented in the right panel of Figure 4. The orange bars represent the simulated data while the purple bars show the CSO data. Although

[6] A percentile measures a value up to which a certain percentage of data falls. For example, if a 47% percentile of earnings is €600, it means that 47% of the earnings data falls between zero and €600.

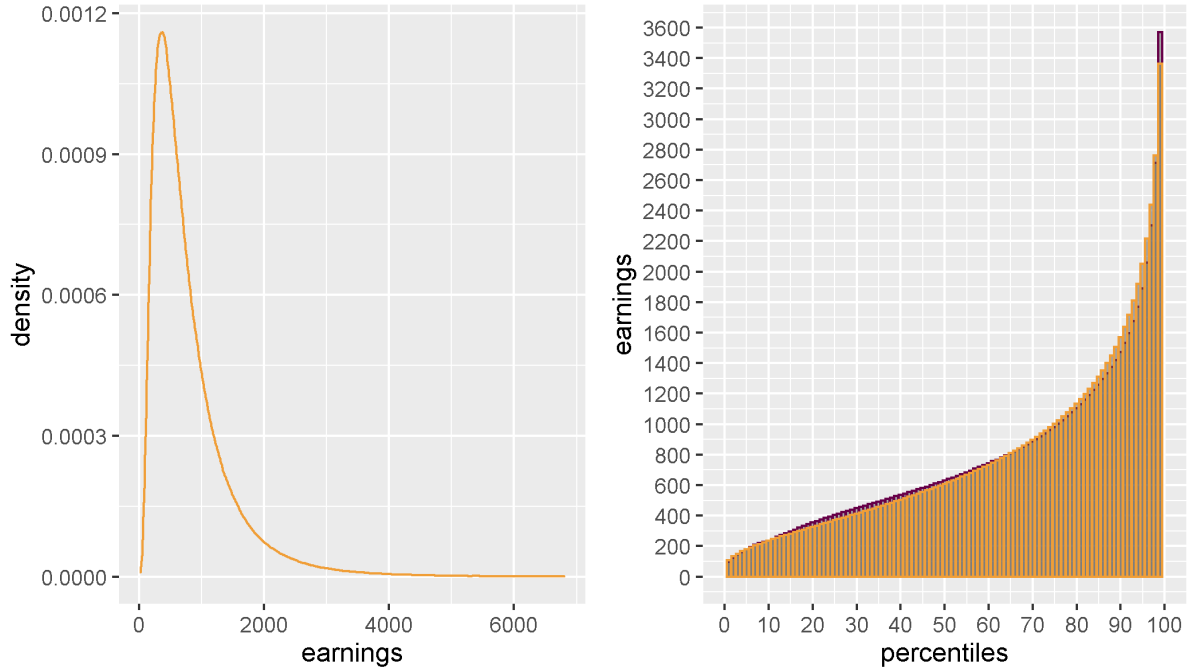
[7] Central Statistics Office, "Earnings Analysis Using Administrative Data Sources 2020," 2021, https://www.cso.ie/en/releasesandpublications/ep/p-eaads/earningsanalysisusingadministrativedatasources2020/distribution/.

[8] More precisely, the data were simulated according to a log-normal distribution with the maximum value being capped at €6,800, which is a weekly average of the annual earnings of €350,000. This salary level is the upper bound one of the top paid position, chief financial officers, according to a professional services recruitment consultancy. Morgan McKinley, "10 Of The Highest Paying Jobs Ireland In 2022," December 1, 2021, https://www.morganmckinley.com/ie/article/10-highest-paying-jobs-ireland-in-2022.

[9] Central Statistics Office, "Statistical Yearbook of Ireland 2020: Labour Market," 2020, https://www.cso.ie/en/releasesandpublications/ep/p-syi/statisticalyearbookofireland2020/soc/labourmarket/.

the overlap is not perfect, the percentiles of the simulated micro data approximate the CSO data well.

Figure 4: Simulated weekly earnings and comparison with the CSO data




Note: The left panel is the distribution of the simulated micro data on weekly earnings. The right panel is the comparison between the percentiles of the simulated micro data (orange bars) and the CSO percentile data (purple bars).

## Example Application

Once micro data are simulated, the next step is to use them for analysis. Here, we offer an example of a policy costing analysis, where a hypothetical welfare benefit is paid for a certain period of eligible leave from work, with the maximum payment period of 6 months (26 weeks).[10] We assume that the duration of the benefit is measured on a weekly basis. The payment is modelled according to three scenarios:

- Scenario 1: A flat payment of €200 per week
- Scenario 2: 80% of prior weekly earnings for each of the first 10 weeks, 60% of prior weekly earnings thereafter, with the minimum cap of €150 and the maximum cap of €250.
- Scenario 3: 80% of prior weekly earnings for each of the first 10 weeks, 60% of prior weekly earnings thereafter, with the minimum cap of €150 and the maximum cap of €350.

[10] We leave the scope of the benefit intentionally abstract, to avoid the analysis being mistaken as something related to actual welfare schemes.

The analysis also draws on the following assumptions. We randomly draw 5% of the population (2,222,500 individuals in employment) and consider them to be the recipients of the benefit within a year. The duration of the receipt of the benefit for each individual is modelled using a normal distribution with a mean of 10 (weeks), a standard deviation of 5 (weeks), a minimum value capped at 1 (week), and a maximum value capped at 26 (weeks).[11] This assumption means that there are more people who exit from the hypothetical scheme at earlier stages rather than at later stages. We assume that all three factors (earnings, the likelihood, and the duration) are independent of one another.[12] Having these assumptions, we can calculate the annual cost.

There is always uncertainty over whether simulated data are exactly the same as the corresponding actual (but unavailable or unknown) data. To incorporate such uncertainty, we can use multiple replications of the simulation known as Monte Carlo analysis. Monte Carlo analysis repeats a simulation many times, as simulated data vary per iteration because of the stochastic nature of drawing values from a theoretical distribution.

The stochastic nature, or "stochasticity," can be described as follows. Imagine that we were drawing a sample of 50 balls from a bag that has a million red balls and a million blue balls (the universe or the population). While we can expect to draw 25 red balls and 25 blue balls on average, each attempt may not always result in this (e.g., it might be 24 red balls and 26 blue balls). Stochasticity here refers to the possibility of a sample's nature deviating from the nature of the population (here, two million balls) by random chance.

In Monte Carlo analysis, on the basis that a simulation is repeated many times over, the estimate of interest (e.g., the total cost of the benefit) is computed per iteration. Once all iterations are complete, we obtain the distribution of these cost estimates. This distribution captures the uncertainty around the estimate, and we can, for example, compute the 95% credible/confidence interval based on that.[13] The downside of Monte Carlo analysis is that it is computationally very demanding.

Stochasticity can come not only from the process of drawing values from a theoretical distribution, but also from the process of determining parameter values for a theoretical distribution (e.g., values for the mean and the standard deviation parameters). When the analyst is unsure exactly what parameter values to use, they can express this uncertainty by using a theoretical distribution from which to draw parameter values, instead of determining a single specific parameter value. As a result, the distribution of the estimates after Monte Carlo analysis will generally be wider; in other words, there will be a greater variation in the

[11] The values generated are then rounded up to a zero digit because the unit used here is weeks. A normal distribution with the minimum/maximum value capped is called "truncated normal distribution."

[12] If at least two of these factors are modelled as dependent on one another, it will be necessary to determine the parameter values that govern the type of relationship between these factors. Therefore, if these factors are interlinked, it is more complex to simulate micro data.

[13] A credible/confidence interval here means an interval that covers the range of the values that are likely to be observed with a certain probability level. For example, a 95% credible interval covers the range of the values that are 95% likely to be observed; the values outside this interval are likely to be observed with the remaining 5% probability. The 95% probability level is one conventional threshold applied in statistical analysis.

estimates, implying a greater level of uncertainty around the estimates. Here, the stochasticity on parameters is, as an example, modelled for the standard deviation parameter of the duration of the receipt of the benefit.[14]

We then ran the Monte Carlo analysis, iterating the simulation 100 times.[15] The total cost estimates for the three scenarios are presented in Table 1, where the uncertainty due to the stochasticity of the simulation process is expressed as the 95% credible interval.

Table 1: Results of the Monte Carlo analysis of a hypothetical welfare scheme

| | Mean | Lower Bound | Upper Bound |
|---|---|---|---|
| Scenario 1 | €231.62 million | €221.34 million | €248.74 million |
| Scenario 2 | €289.53 million | €276.67 million | €310.92 million |
| Scenario 3 | €405.34 million | €387.34 million | €435.29 million |

Note: the lower and upper bounds are those of the 95% credible interval.

One interesting observation is that, although Scenario 2 has the minimum cap of €150 and the maximum cap of €250 and the average value between these two caps is $€200 = \frac{150+250}{2}$, it costs 25% more than Scenario 1, where the flat rate of €200 is applied. This is because the cost per individual is not a function of the average value of the two caps but a function of their prior earnings with these caps applied. For example, if all recipients had prior earnings of €500, they would all receive €250 for their entire duration of the receipt of the benefit. Recall that the weekly payment for the first 10 weeks is 80% of prior earnings; that for the remaining weeks is 60% of prior earning; and there is the minimum cap of €150 and the maximum cap of €250. Then, $€400 = €500 \times 80\%$; $€300 = €500 \times 60\%$; and both figures are larger than €250, so that all recipients would receive €250.

Of course, the analysis here is a simple example based on a hypothetical welfare benefit payment. Nonetheless, the main takeaway is that the calculation of the total cost is transparent

[14] A value for the standard deviation parameter for the duration is drawn from a normal distribution with the mean of 5 and the standard deviation of 1 with the minimum possible value capped at 1 (as the standard deviation must be greater than zero by definition). The number drawn is then rounded up to a zero digit as the unit used here is weeks. This setup means that the value of the standard deviation parameter should be, on average, 5 weeks but can be 3, 4, 6, or 7 weeks with approximately the 95% probability (and can be smaller than 3, or larger than 7, with the remaining probability).

[15] How many iterations of a simulation are adequate depends mainly on the size of data generated within the simulation, the complexity of the data simulation process, and the available computational power. Doing more iterations means more stability in the estimates but also greater demand for computational power and time. In the current setup, 100 iterations were enough to achieve a good balance between the stability in the estimates and fast completion in a standard computer; if we did 1,000 iterations, the estimates were only marginally different in proportional terms (the maximum difference of a 0.3% change).

once we simulate the underlying micro data. Namely, we calculate the cost per individual and sum all up.

It is worth noting that Monte Carlo analysis is useful even when micro data are available. It can be used to model uncertainty around how representative the micro data one has are of the target population of a policy (i.e., whether it is plausible to extrapolate findings from observed data into the target population). As discussed previously, observed data might be considered as one realisation of some underlying theoretical distribution. Assuming that, it is worth recalling now that data drawn from (i.e., simulated by) a theoretical distribution may not necessarily be representative (as has been explained by an example of drawing balls from a bag of blue and red balls). In the same vein, observed data may not necessarily be representative of the underlying true distribution. It is possible to quantify this uncertainty by Monte Carlo analysis, assuming that observed data were drawn from a certain theoretical distribution.

# Conclusion

The statistical simulation techniques outlined in this paper are a useful addition to policy analysts' toolkit. While computationally resource-intensive in some cases, it can facilitate more sophisticated policy and costing analysis. When micro data are not available, Monte Carlo analysis can simulate micro data while modelling uncertainty around the process whereby data are generated. Furthermore, even when micro data is available, Monte Carlo analysis can help quantify uncertainty around the observed micro data.

# Appendix: R Code

```
# Install the necessary packages if not installed yet
# install.packages("ggplot2")
# install.packages("WDI")
# install.packages("readxl")
# install.packages("gridExtra")
# install.packages("EnvStats")


# Load the necessary libraries
library("ggplot2")
library("WDI")
library("readxl")
library("gridExtra")
library("EnvStats")



####################################
# Theoretical distribution examples #
####################################


# Bernoulli distribution

x <- c(0,1)

d <- data.frame(prob = dbinom(x, 1, 0.5), x = x)

ggplot(data = d, aes(x = x, y = prob)) +
  geom_bar(stat = "identity", fill = "#F09E37") +
  scale_x_continuous(breaks = c(0, 1)) +
  scale_y_continuous(limits = c(0, 1)) +
```

```
  labs(y = "probability", x = "data values") +
  theme_bw() -> bern

ggsave("bern.png", plot = bern, width = 2, height = 2, units = "in")


# Standard normal distribution

x <- seq(from = -3, to = 3, by = 0.001)

d <- data.frame(dens = dnorm(x, 0, 1), x = x)

ggplot(data = d, aes(x = x, y = dens)) +
  geom_density(stat = "identity", color = "#F09E37") +
  scale_x_continuous(breaks = seq(from = -3, to = 3, by = 1)) +
  labs(y = "density", x = "data values") +
  theme_bw() -> snd

ggsave("snd.png", plot = snd, width = 3.5, height = 2, units = "in")


# GDP per capita across the world from 2015 to 2020
wdidata <- WDI(country = "all",
               indicator = c(gdppc = "NY.GDP.PCAP.KD"),
               start = 2015, end = 2020, extra = TRUE)

wdidata_ss <- subset(wdidata, region != "Aggregates")
wdidata_ss$lngdppc <- log(wdidata_ss$gdppc)

# Mean
mean(wdidata_ss$lngdppc, na.rm = TRUE)

# Standard deviation
sd(wdidata_ss$lngdppc, na.rm = TRUE)

wdidata_ss$nd <- dnorm(wdidata_ss$lngdppc,
                       mean(wdidata_ss$lngdppc, na.rm = TRUE),
                       sd(wdidata_ss$lngdppc, na.rm = TRUE))

ggplot(data = wdidata_ss) +
  geom_histogram(aes(y = ..density.., x = lngdppc), binwidth = 0.25,
                 color = "black", fill = "#670048") +
  geom_density(aes(y = nd, x = lngdppc), stat = "identity",
               color = "#F09E37", size = 1.5) +
  labs(y = "density", x = "GDP per capita (logarithm scale)") +
  theme_bw() -> gdppcdist

ggsave("gdppcdist.png", plot = gdppcdist, width = 3.5, height = 2, units = "in")




###########
# Example #
###########


# Earnings distribution 2020
#
# Source: CSO
# https://www.cso.ie/en/releasesandpublications/ep/p-
eaads/earningsanalysisusingadministrativedatasources2020/distribution/
earnings_dist <- data.frame(percentile = 1:99,
                            value = c(
                              95, 121, 144, 162, 180, 195, 208, 221, 226, 235,
                              248, 260, 272, 285, 297, 308, 320, 332, 344, 354,
                              364, 375, 384, 394, 403, 412, 422, 431, 440, 449,
                              458, 466, 475, 483, 492, 500, 509, 518, 527, 536,
                              544, 554, 563, 572, 581, 590, 600, 610, 619, 629,
                              640, 650, 661, 673, 683, 695, 708, 720, 732, 744,
                              757, 767, 780, 794, 808, 823, 838, 853, 868, 885,
                              902, 921, 940, 960, 980, 1002, 1026, 1050, 1076,
                              1103, 1132, 1160, 1192, 1223, 1259, 1296, 1335,
                              1376, 1421, 1473, 1532, 1598, 1676, 1768, 1889,
                              2058, 2303, 2714, 3571
                            )
```

```
)


# Simulate micro data
ev <- 801
sd <- 2.1
logsd <- log(sd)
mean <- exp( log(ev) - (1/2 * (logsd^2)) )
logmean <- log(mean)

set.seed(2022)
n <- 2222500
simweekpay <- round(rlnormTrunc(n, logmean, logsd, max = 6800), digit = 2)


# Aggregate simulated weekly earnings data into the percentile
earnings_dist$simweekpay_percentile <- quantile(simweekpay, probs = rep(1:99)/100)


# Plot the CSO data and the simulated data for comparison
ggplot(data = earnings_dist, aes(x = percentile)) +
  geom_bar(stat = "identity", aes(y = value),
           alpha = 0.5, color = "#670048") +
  geom_bar(stat = "identity", aes(y = simweekpay_percentile),
           alpha = 0.5, color = "#F09E37") +
  scale_x_continuous(breaks = rep(0:10)*10,
                     lim = c(0,100)) +
  scale_y_continuous(breaks = seq(from = 0,
                                  to = 3600,
                                  by = 200)) +
  labs(y = "earnings", x = "percentiles") -> earnings_comp

ggplot(data = data.frame(simweekpay)) +
  geom_density(aes(x = simweekpay), color = "#F09E37") +
  labs(y = "density", x = "earnings") -> earnings_sim


earnings_plots <- grid.arrange(earnings_sim, earnings_comp, nrow = 1)


ggsave("earnings_plots.png", plot = earnings_plots,
       width = 7, height = 4, units = "in")



###############
# Application #
###############


# Model to compute the total cost of the benefit per recipient
model <- function(weekpay, dur, lims = FALSE, min = 0, max = 1e4){


  # Return an error for wrong inputs
  if(is.numeric(weekpay) == FALSE | weekpay <= 0 |
     is.na(weekpay) == TRUE | round(weekpay, digit = 2) != weekpay){

    stop("weekpay must be a real number up to two digits.")
  }

  if(is.numeric(dur) == FALSE | dur < 0 |
     is.na(dur) == TRUE | round(dur) != dur){

    stop("dur must be a natural number or zero.")
  }

  if(is.logical(lims) == FALSE){

    stop("lims must be a logical value.")
  }

  if(lims == TRUE){
    if(is.numeric(min) == FALSE | min < 0 |
       is.na(min) == TRUE | round(min, digit = 2) != min){
```

```
    stop("min must be a real number up to two digits or zero.")
  }

  if(is.numeric(max) == FALSE | max <= 0 |
     is.na(max) == TRUE | round(max, digit = 2) != max){

    stop("max must be a real number up to two digits.")
  }
}


# Determine the threshold weeks after which the benefit is reduced
threshold1 <- 10


# Function to apply min and max limits on benefits
limit <- function(benefit){

  if(benefit > max){
    revbenefit <- max
  }

  if(benefit < min){
    revbenefit <- min
  }

  if(benefit <= max & benefit >= min){
    revbenefit <- benefit
  }

  return(revbenefit)
}


# 1st stage benefit
# Min and max limits applied or not
if(lims == TRUE){

  weekpay1 <- limit(weekpay*0.8)
}
else{

  weekpay1 <- weekpay*0.8
}

# Round the value in case it has more than two digits
weekpay1 <- round(weekpay1, digit = 2)

# Compute the total benefit at the first stage
if(dur <= threshold1){

  fststg <- weekpay1 * dur
}
else{

  fststg <- weekpay1 * threshold1
}


# 2nd stage benefit
# Min and max limits applied or not
if(lims == TRUE){

  weekpay2 <- limit(weekpay*0.6)
}
else{

  weekpay2 <- weekpay*0.6
}

# Round the value in case it has more than two digits
weekpay2 <- round(weekpay2, digit = 2)

# Compute the total benefit at the second stage
if(dur > threshold1){

  scndstg <- weekpay2 * (dur - threshold1)
```

```
  }
  else{

    scndstg <- 0
  }


  # Sum up all benefits to compute the total cost
  tlcst <- fststg + scndstg
  return(tlcst)

}


# Monte Carlo setup
iter <- 100
mat <- matrix(nrow = iter, ncol = 3)
likeli <- 0.05 # Likelihood of receiving the benefit
nsample <- n*0.1 # To make the loop finish faster; the sum of the individual
                 # costs is then multiplied by 10 to obtain the population
                 # value


for(j in 1:iter){


  # Earnings
  set.seed(j)
  simweekpay <- round(rlnormTrunc(nsample, logmean, logsd, max = 6800),
                      digit = 2)


  # Whether an individual receives the benefit
  set.seed(j)
  receipt <- rbinom(nsample, 1, likeli)


  # Subset for those who receive the benefit
  d <- data.frame(simweekpay, receipt)
  d_receipt <- subset(d, receipt == 1)


  # Number of observations
  nrow_d_receipt <- nrow(d_receipt)


  # Duration of the receipt of the benefit
  set.seed(j)
  d_receipt$dur <- round(
    rnormTrunc(nrow_d_receipt,
               mean = 10,
               sd = round(
                 rnormTrunc(1, mean = 5,
                            sd = 1, min = 1)
               ),
               min = 1,
               max = 26)
  )


  # Flat payment of 200
  results1 <- mean(d_receipt$dur) * 200 * nrow_d_receipt
  scenario1 <- sum(results1) * 10 / 1e6 # in a million euro


  # 80% of prior earnings for first 10 weeks, 60% thereafter
  # with the min and max cap of 150 and 250
  results2 <- rep(NA, nrow_d_receipt)
  for(i in 1:nrow_d_receipt){

    results2[i] <- model(d_receipt$simweekpay[i], d_receipt$dur[i],
                         lims = TRUE, min = 150, max = 250)
  }

  scenario2 <- sum(results2) * 10 / 1e6 # in a million euro
```

```
  # 80% of prior earnings for first 10 weeks, 60% thereafter
  # with the min and max cap of 150 and 350
  results3 <- rep(NA, nrow_d_receipt)
  for(i in 1:nrow_d_receipt){

    results3[i] <- model(d_receipt$simweekpay[i], d_receipt$dur[i],
                         lims = TRUE, min = 150, max = 350)
  }

  scenario3 <- sum(results3) * 10 / 1e6 # in a million euro


  # Store the results into the matrix
  mat[j,] <- c(scenario1, scenario2, scenario3)


  print(paste(j, " done", sep = ""))
}


# Mean and 95% credible interval for Scenario 1
mean(mat[,1])
quantile(mat[,1], probs = c(0.025, 0.975))


# Mean and 95% credible interval for Scenario 3
mean(mat[,2])
quantile(mat[,2], probs = c(0.025, 0.975))


# Mean and 95% credible interval for Scenario 3
mean(mat[,3])
quantile(mat[,3], probs = c(0.025, 0.975))
```