PS2

Shinjo Hiroki

2024-01-23

I owe a lot to Ichiro Sakurai.

(a)

```
pacman::p_load(readxl, tidyverse, numDeriv, Rmpfr, gmp, gt, doParallel, tictoc, fixest)

import <- function(used_group = c(1,2,3,4,5,6,7,8)) {
  sheets <- c("g870", "rt50", "t8h203", "a530875", "a530874", "a452374", "a530872", "a452372")
  G <- list()
  for (i in used_group) {
    G_new <- read_excel("rust_data_clean(1).xlsx",sheet = sheets[i], col_names = FALSE, .name_repair ="
    G <- c(G, list(G_new))
  }
  Gm <- map(G, ~.x[1:11, ] %>% t() %>% as_tibble(.name_repair = "minimal"))
  name <- c("bus_number", "month_purchased", "year_purchased", "month_1st", "year_1st",
            "meter_1st", "month_2nd", "year_2nd", "meter_2nd", "month_begin","year_begin")
  Gm <- map(Gm, ~set_names(.x, name))
  for (i in 1:length(used_group)){
      Gm[[i]] <- Gm[[i]] %>% mutate(group = used_group[i])
  }
  Gm <- bind_rows(Gm)
  Gv <- map(G, ~.x[12:nrow(.x),])
  for (i in 1:length(used_group)){
    name <- G[[i]][1,] %>% t() %>% as.vector()
    colnames(Gv[[i]]) <- name
  }
  return(list(Gm = Gm, Gv = Gv))
}
data <- import()
Gm <- data$Gm; Gv <- data$Gv
```

2a

```
df_2a <- Gm %>%
  mutate(mile_1st = meter_1st,
         mile_2nd = meter_2nd - meter_1st,
         elap_1st = (year_1st - year_purchased) * 12 + (month_1st - month_purchased),
```

```r
         elap_2nd = (year_2nd - year_1st) * 12 + (month_2nd - month_1st)) %>%
  select(group, bus_number, mile_1st, mile_2nd, elap_1st, elap_2nd) %>%
  pivot_longer(c(-group, -bus_number)) %>%
  filter(value > 0) %>%
  mutate(name = str_sub(name, 1, 4),
         value = if_else(name == "elap", value + 1, value))

df_2a_1 <- c(1, rep(0, 9), 2, rep(0, 9)) %>%
  matrix(nrow = 2, byrow = TRUE) %>%
  as_tibble(.name_repair = "minimal")
df_2a_2 <- df_2a %>%
  group_by(group) %>%
  summarize(mile_max = max(value[name == "mile"]),
            mile_min = min(value[name == "mile"]),
            mile_mean = mean(value[name == "mile"]),
            mile_sd = sd(value[name == "mile"]),
            elap_max = max(value[name == "elap"]),
            elap_min = min(value[name == "elap"]),
            elap_mean = mean(value[name == "elap"]),
            elap_sd = sd(value[name == "elap"]),
            n = n() / 2) %>%
  mutate(group = as.character(group))
df_2a_3 <- df_2a %>%
  summarize(mile_max = max(value[name == "mile"]),
            mile_min = min(value[name == "mile"]),
            mile_mean = mean(value[name == "mile"]),
            mile_sd = sd(value[name == "mile"]),
            elap_max = max(value[name == "elap"]),
            elap_min = min(value[name == "elap"]),
            elap_mean = mean(value[name == "elap"]),
            elap_sd = sd(value[name == "elap"]),
            n = n() / 2) %>%
  mutate(group = "Full Sapmle")
colnames(df_2a_1) <- colnames(df_2a_2)
df_2a_1$group <- as.character(df_2a_1$group)

bind_rows(df_2a_1, df_2a_2, df_2a_3) %>%
  gt() %>%
  tab_spanner(label = "Mileage at Replacement", columns = starts_with("mile")) %>%
  tab_spanner(label = "Elapsed Time (Months)", columns = starts_with("elap")) %>%
  cols_label(group = "Bus Group", mile_max = "Max", mile_min = "Min", mile_mean = "Mean",
             mile_sd = "Starndard Deviation", elap_max = "Max", elap_min = "Min", elap_mean = "Mean",
             elap_sd = "Standard Deviation", n = "Number of Observations") %>%
  cols_align(align = "center", columns = everything()) %>%
  tab_header(title = md("TABLE 2a")) %>%
  fmt_number(columns = c(2:7, 10), decimals = 0) %>%
  fmt_number(columns = c(8, 9), decimals = 1) %>%
  gtsave(file = "2a.pdf")

knitr::include_graphics("2a.pdf")
```

TABLE 2a									
	Mileage at Replacement				Elapsed Time (Months)				
Bus Group	Max	Min	Mean	Starndard Deviation	Max	Min	Mean	Standard Deviation	Number of Observations
1	0	0	0	0	0	0	0.0	0.0	0
2	0	0	0	0	0	0	0.0	0.0	0
3	273,400	124,800	199,733	37,459	74	38	59.1	10.9	27
4	387,300	121,300	257,336	65,477	116	28	73.7	23.3	33
5	322,500	118,000	245,291	60,258	127	31	85.4	29.7	11
6	237,200	82,400	150,786	61,007	127	49	74.7	35.2	7
7	331,800	121,000	208,963	48,981	104	41	68.3	16.9	27
8	297,500	132,000	186,700	43,956	104	36	58.4	22.2	19
Full Sapmle	387,300	82,400	216,354	60,475	127	28	68.1	22.4	124

2b

```
df_2b <- Gm %>%
  filter(year_1st == 0) %>%
  mutate(mile = map2_dbl(group, as.character(bus_number), ~ Gv[[.x]][nrow(Gv[[.x]]), .y] %>% unlist()),
         elap = 12*(85 - year_purchased) + 5 - month_purchased + 1) %>%
  select(group, mile, elap)

df_2b_1 <- df_2b %>%
  group_by(group) %>%
  summarize(mile_max = max(mile),
            mile_min = min(mile),
            mile_mean = mean(mile),
            mile_sd = sd(mile),
            elap_max = max(elap),
            elap_min = min(elap),
            elap_mean = mean(elap),
            elap_sd = sd(elap),
            n = n()) %>%
  mutate(group = as.character(group))

df_2b_1[is.na(df_2b_1)] <- 0
df_2b_2 <- c(7, rep(0, 9), 8, rep(0,9)) %>%
  matrix(nrow = 2, byrow = TRUE) %>%
  as_tibble(.name_repair = "minimal")
colnames(df_2b_2) <- colnames(df_2b_1)
df_2b_2$group <- as.character(df_2b_2$group)
df_2b_3 <- df_2b %>%
  summarize(mile_max = max(mile),
            mile_min = min(mile),
            mile_mean = mean(mile),
            mile_sd = sd(mile),
            elap_max = max(elap),
            elap_min = min(elap),
            elap_mean = mean(elap),
            elap_sd = sd(elap),
            n = n()) %>%
  mutate(group = "Full Sample")

bind_rows(df_2b_1, df_2b_2, df_2b_3) %>%
  gt() %>%
  tab_spanner(label = "Mileage at May 1, 1985", columns = starts_with("mile")) %>%
  tab_spanner(label = "Elapsed Time (Months)", columns = starts_with("elap")) %>%
  cols_label(group = "Bus Group", mile_max = "Max", mile_min = "Min", mile_mean = "Mean",
             mile_sd = "Starndard Deviation", elap_max = "Max", elap_min = "Min", elap_mean = "Mean",
             elap_sd = "Standard Deviation", n = "Number of Observations") %>%
  cols_align(align = "center", columns = everything()) %>%
  tab_header(title = md("TABLE 2b")) %>%
  fmt_number(columns = c(2:7, 10), decimals = 0) %>%
  fmt_number(columns = c(8), decimals = 1) %>%
  fmt_number(columns = c(9), decimals = 2) %>%
  gtsave("2b.pdf")
```

```r
knitr::include_graphics("2b.pdf")
```

TABLE 2b

	Mileage at May 1, 1985				Elapsed Time (Months)				
Bus Group	Max	Min	Mean	Starndard Deviation	Max	Min	Mean	Standard Deviation	Number of Observations
1	120,151	65,643	100,117	12,929	25	25	25.0	0.00	15
2	161,748	142,009	151,183	8,530	49	49	49.0	0.00	4
3	280,802	199,626	250,766	21,325	75	75	75.0	0.00	21
4	352,450	310,910	337,222	17,802	118	117	117.8	0.45	5
5	326,843	326,843	326,843	0	130	130	130.0	0.00	1
6	299,040	232,395	265,264	33,332	130	128	129.3	1.15	3
7	0	0	0	0	0	0	0.0	0.00	0
8	0	0	0	0	0	0	0.0	0.00	0
Full Sample	352,450	65,643	207,782	85,208	130	25	66.4	34.65	49

(b)

In the 1st step, we estimate a transition matrix. In the 2nd step, we estimate a conditional choice probabilities (CCP). Then we obtain $\hat{\theta}$, which consistent with data.

```
make_dt <- function(used_group = c(1,2,3,4), start_0 = FALSE) {
  data <- import(used_group)
  Gm <- data$Gm
  Gv <- data$Gv
  Gm$bus_number <- as.character(Gm$bus_number)
  Gv_rep <- Gv
  for (i in 1:length(used_group)) {
    Gv_rep[[i]][] <- FALSE
  }
  Gv_new <- Gv

  for (i in 1:length(used_group)) {
    for (j in names(Gv_new[[i]])) {
      mil_1st <- Gm %>% filter(group == used_group[i], bus_number == j) %>%
        select(meter_1st) %>% pull()
      mil_2nd <- Gm %>% filter(group == used_group[i], bus_number == j) %>%
        select(meter_2nd) %>% pull()
      mil_2nd_diff <- mil_2nd - mil_1st
      if (mil_1st > 0){
        vec <- Gv_new[[i]][,j] %>% pull()
      }
      if(mil_2nd > 0){
        after_2nd <- (vec > mil_2nd)
        place_2nd <- which(after_2nd) %>% min()
        vec[after_2nd] <- vec[after_2nd] - mil_2nd_diff
        Gv_rep[[i]][place_2nd-1,j] <- TRUE
      }
      if(mil_1st > 0){
        after_1st <- (vec > mil_1st)
        place_1st <- which(after_1st) %>% min()
        vec[after_1st] <- vec[after_1st] - mil_1st
        Gv_new[[i]][,j] <- vec
        Gv_rep[[i]][place_1st-1,j] <- TRUE
      }
    }
  }
  dt <- list()
  for (i in 1:length(used_group)) {
    year <- Gm[Gm$group == used_group[i], ]$year_begin[1]
    month <- Gm[Gm$group == used_group[i], ]$month_begin[1]
    if(month < 10){
      month <- paste("0", month, sep="")
    }
    start <- paste("19", year, "-", month, "-01", sep="") %>% as.Date()
    len <- Gv_new[[i]] %>% nrow()
    time <- seq(from = start, length.out = len, by="month")
    mileage <- Gv_new[[i]] %>%
      mutate(time = time) %>%
      pivot_longer(cols = !"time", names_to = "bus_number", values_to = "mileage")
```

```r
    replace <- Gv_rep[[i]] %>%
      mutate(time = time) %>%
      pivot_longer(cols = !"time", names_to = "bus_number", values_to = "replacement")
    dt[[i]] <- mileage %>%
      left_join(replace, by = c("time", "bus_number")) %>%
      arrange(bus_number, time) %>%
      mutate(group = used_group[i])
  }
  dt <- bind_rows(dt)
  dim <- 90
  delta <- 5000
  if (start_0) {
    dt <- dt %>%
      mutate(mil = cut(mileage,
                       breaks = seq(0, dim * delta, length.out=dim+1),
                       labels = seq(0, dim-1),
                       include.lowest = TRUE) %>% as.character() %>% as.numeric())
  } else {
    dt <- dt %>%
      mutate(mil = cut(mileage,
                       breaks = seq(0, dim * delta, length.out=dim+1),
                       labels = seq(0, dim-1),
                       include.lowest = TRUE) %>% as.numeric())
  }
  return(dt)
}

transition_mat <- function(used_group = c(1,2,3,4), remove_last = TRUE, start_0 = FALSE) {
  dt <- make_dt(used_group, start_0)
  dt_diff <- dt %>%
    group_by(bus_number) %>%
    mutate(mil_next = lead(mil, n = 1)) %>%
    ungroup() %>%
    drop_na() %>%
    mutate(diff = mil_next - mil)
  dt_0 <- dt_diff %>%
    filter(!replacement)
  theta3_0 <- dt_0 %>%
    group_by(diff) %>%
    summarise(p = n() / nrow(dt_0)) %>%
    mutate(se = sqrt(p * (1 - p) / nrow(dt_0)))
  dt_1 <- dt_diff %>%
    filter(replacement)
  theta3_1 <- dt_1 %>%
    group_by(mil_next) %>%
    summarise(p = n() / nrow(dt_1))
  dim <- 90
  n <- nrow(theta3_0)
  P0 <- matrix(0, nrow = dim, ncol = dim)
  for (i in 1:dim) {
    for (j in 1:n) {
      if (i + j - 1 > dim){
        P0[i, dim] <- P0[i, dim] + theta3_0 %>% filter(diff == j - 1) %>% pull(p)
```

```
      } else {
        P0[i, i + j - 1] <- theta3_0 %>% filter(diff == j - 1) %>% pull(p)
      }
    }
  }
  n <- nrow(theta3_1)
  P1 <- matrix(0, nrow = dim, ncol = dim)
  for (i in 1:dim) {
    for (j in 1:n) {
      if (start_0) {
        P1[i, j] <- theta3_1 %>% filter(mil_next == j-1) %>% pull(p)
      } else {
        P1[i, j] <- theta3_1 %>% filter(mil_next == j) %>% pull(p)
      }
    }
  }
  if (remove_last) {
    dt <- dt_diff
  } else {
    dt <- dt %>%
      group_by(bus_number) %>%
      mutate(mil_prev = lag(mil, n = 1)) %>%
      ungroup() %>%
      drop_na()
  }
  return(list(dt, P0, P1, dim, theta3_0))
}
```

2 Step Estimation

```
two_step <- function(used_group = c(1,2,3,4),
                     beta = 0.9999,
                     parameter = c(0,0),
                     method="Nelder-Mead",
                     approach = "matrix",
                     ccp_logit = TRUE) {
  data <- transition_mat(used_group = used_group)
  dt <- data[[1]];P0 <- data[[2]];P1 <- data[[3]];dim <- data[[4]];theta3_0 <- data[[5]]
  state <- seq(1, dim)
  Euler_const <- - digamma(1)
  num_choice <- 2
  if (ccp_logit) {
    logit_model <- fixest::feglm(replacement ~ mileage + mileage^2,
                                 dt,
                                 family = binomial("logit"))
    CCP_1st <- cbind(
      1 - predict(logit_model, tibble(mileage=state)),
      predict(logit_model, tibble(mileage=state))
    )
  } else {
    tol <- 1e-10
```

```r
  CCP_1st <- dt %>%
    group_by(mil) %>%
    summarise(p1 = sum(if_else(replacement, 1, 0) / n())) %>%
    mutate(p1 = p1 + tol,
           p0 = 1 - p1) %>%
    select(-mil) %>%
    select(p0, p1) %>%
    as.matrix() %>%
    rbind(. , cbind(rep(1 - tol, dim - nrow(. )), rep(tol, dim - nrow(. ))))
}
util <- function(theta, state){
    RC <- theta[1]
    theta1 <- theta[2]
    U_0 <- -0.001 * theta1 * state
    U_1 <- -RC
    U <- cbind(U_0, U_1)
    return(U)
}
mat <- Vectorize(
    function(i,j,mat) {mat[i,j]},
    vectorize.args = c("i", "j"))
G <- array(c(P0, P1), dim = c(dim, dim, num_choice))
policy_inv <- function(theta,  CCP, beta, G, state){
  U <- util(theta, state)
  num_states <- length(state)
  psi <- Euler_const * matrix(1, num_states, num_choice) - log(CCP)
  V <- solve(diag(num_states) - beta * (CCP[,1] %*% matrix(1, 1, num_states) * G[,,1] +
    rowSums(CCP * (U + psi))
  CV <- U + beta * cbind(G[,,1] %*% V, G[,,2] %*% V)
  CV <- mpfr(CV, precBits = 53)
  CCP <- exp(CV) / rowSums(exp(CV))
  CCP <- gmp::asNumeric(CCP)
  return(CCP)
}

policy_dep <- function(theta, CCP, beta, G, state){
  U <- util(theta, state)
  CV_dif <- U[,2] - U[,1] + beta * (G[,,2] %*% (-log(CCP[,2])) - G[,,1] %*% (-log(CCP[,2])))
  prob_buy <- exp(CV_dif) / (1 + exp(CV_dif))
  CCP <- cbind(1-prob_buy, prob_buy)
  return(CCP)
}

policy_for <- function(theta,  CCP, beta, G, state){
}
likelihood <- function(theta, CCP, dt, beta, G, state, policy_operator){
    CCP <- policy_operator(theta, CCP, beta, G, state)
    obj <- sum(log(mat(dt$mil, dt$replacement + 1, CCP)))
  return(obj)
}
init <- parameter
operator <- switch(approach,
                   matrix = policy_inv,
```

```r
                     finite = policy_dep,
                     forward = policy_for)
  result <- optim(init, likelihood,
                            CCP = CCP_1st, dt = dt,
                            beta = beta, G = G, state = state,
                            policy_operator = operator,
                            control = list(fnscale = -1),
                            method = method)
  theta_hat <- result$par
  LL <- result$value
  hessian <- numDeriv::hessian(func = likelihood, x = theta_hat,
                            CCP = CCP_1st, dt = dt,
                            beta = beta, G = G, state = state,
                            policy_operator = operator)
  se_hat <- sqrt(diag(solve(-hessian)))
  return(list(theta_hat = theta_hat, se_hat = se_hat, LL = LL,
              theta3_0 = theta3_0, obs = dt %>% nrow()))
}
```

```r
beta <- list(0.9999, 0.9999,0.9999, 0, 0, 0)
used_group <-  list(c(1,2,3), c(4), c(1,2,3,4), c(1,2,3), c(4), c(1,2,3,4))
cores <- detectCores()-1
cl <- makeCluster(cores)
registerDoParallel(cl)
Ts <- foreach(beta = beta, used_group = used_group,
              .packages = c("readxl", "tidyverse", "numDeriv", "Rmpfr", "gmp", "fixest")) %dopar% {
  two_step(used_group = used_group, beta = beta,
           parameter = c(10,3), approach = "matrix", ccp_logit = TRUE)
}
stopCluster(cl)
```

Table 9

```r
make_table <- function(Ts, name) {
  T1 <- Ts[[1]]; T2 <- Ts[[2]]; T3 <- Ts[[3]]; T4 <- Ts[[4]]; T5 <- Ts[[5]]; T6 <- Ts[[6]]
  f <- function(x) {
    x$theta3_0 %>% select(p) %>% pull() %>% round(4) -> p
    x$theta3_0 %>% select(se) %>% pull() %>% round(4) -> se
    return(paste0(p, "(", se, ")"))
  }
  tb <- matrix("", nrow = 13, ncol = 5)
  tb[1,] <- c("", "", paste0(T1$obs, " Observation"), paste0(T2$obs, " Observation"), paste0(T3$obs, " 
  tb[2,] <- c(" =.9999", "RC",
              paste0(round(T1$theta_hat[1], 4), "(", round(T1$se_hat[1],4), ")"),
              paste0(round(T2$theta_hat[1], 4), "(", round(T2$se_hat[1],4), ")"),
              paste0(round(T3$theta_hat[1], 4), "(", round(T3$se_hat[1],4), ")"))
  tb[3,] <- c("", "$theta_{11}$",
              paste0(round(T1$theta_hat[2], 4), "(", round(T1$se_hat[2],4), ")"),
              paste0(round(T2$theta_hat[2], 4), "(", round(T2$se_hat[2],4), ")"),
              paste0(round(T3$theta_hat[2], 4), "(", round(T3$se_hat[2],4), ")"))
  tb[4,] <- c("", "$theta_{30}$",f(T1)[1], f(T2)[1], f(T3)[1])
```

```r
  tb[5,] <- c("", "$theta_{31}$",f(T1)[2], f(T2)[2], f(T3)[2])
  tb[6,] <- c("", "LL", round(T1$LL, 4), round(T2$LL, 4), round(T3$LL, 4))
  tb[7,] <- c(" =0", "RC",
            paste0(round(T4$theta_hat[1], 4), "(", round(T4$se_hat[1],4), ")"),
            paste0(round(T5$theta_hat[1], 4), "(", round(T5$se_hat[1],4), ")"),
            paste0(round(T6$theta_hat[1], 4), "(", round(T6$se_hat[1],4), ")"))
  tb[8,] <- c("", "$theta_{11}$",
             paste0(round(T4$theta_hat[2], 4), "(", round(T4$se_hat[2],4), ")"),
             paste0(round(T5$theta_hat[2], 4), "(", round(T5$se_hat[2],4), ")"),
             paste0(round(T6$theta_hat[2], 4), "(", round(T6$se_hat[2],4), ")"))
  tb[9,] <- c("", "$theta_{30}$",f(T4)[1], f(T5)[1], f(T6)[1])
  tb[10,] <- c("", "$theta_{30}$",f(T4)[2], f(T5)[2], f(T6)[2])
  tb[11,] <- c("", "LL", round(T4$LL, 4), round(T5$LL, 4), round(T6$LL, 4))
  tb[12,] <- c("Myopia test", "LR Statistic",
             2 * (T1$LL - T4$LL) %>% round(3),
             2 * (T2$LL - T5$LL) %>% round(3),
             2 * (T2$LL - T5$LL) %>% round(3))
  tb[13,] <- c(" =0 vs.  =.9999", "Marginal Significance Level",
             (1 - pchisq(2 * (T1$LL - T4$LL), 1)) %>% round(4),
             (1 - pchisq(2 * (T2$LL - T5$LL), 1)) %>% round(4),
             (1 - pchisq(2 * (T3$LL - T6$LL), 1)) %>% round(4))
  tb %>%
    as.data.frame() %>%
    setNames(object = ., c("Discount Factor", "Estimates/Loglikelihood",
                          "Group 1, 2, 3", "Group 4", "Group 1, 2, 3, 4")) %>%
    gt() %>%
    tab_spanner(label = "Parameter", columns = c("Discount Factor", "Estimates/Loglikelihood")) %>%
    tab_spanner(label = "Data Sample", columns = c("Group 1, 2, 3", "Group 4", "Group 1, 2, 3, 4")) %>%
    cols_align(align = "center", columns = everything()) %>%
    tab_header(title = md("TABLE IX<br>STRUCTURAL ESTIMATES FOR COST FUNCTION c(x, theta_1$) = .001$thet
                         <br>FIXED POINT DIMENSION = 90"),
             subtitle = md("(Standard errors in parentheses)"))  %>%
    gtsave(paste0(name, ".pdf"))
}
make_table(Ts, name = "b")
knitr::include_graphics("b.pdf")
```

TABLE IX
STRUCTURAL ESTIMATES FOR COST FUNCTION c(x, theta_1$) =
.001$theta_{11}$x
FIXED POINT DIMENSION = 90

(Standard errors in parentheses)

Parameter		Data Sample		
Discount Factor	Estimates/Loglikelihood	Group 1, 2, 3	Group 4	Group 1, 2, 3, 4
		3864 Observation	4292 Observation	8156 Observation
β=.9999	RC	9.8164(1.0812)	9.1665(0.9384)	8.6861(0.5797)
	$theta_{11}$	1.5169(0.2725)	1.0066(0.1782)	1.028(0.1244)
	$theta_{30}$	0.3031(0.0074)	0.3949(0.0075)	0.3514(0.0053)
	$theta_{31}$	0.6862(0.0075)	0.5922(0.0075)	0.6367(0.0053)
	LL	-132.1262	-163.6437	-300.797
β=0	RC	8.5578(0.7804)	7.7193(0.5942)	7.3824(0.3773)
	$theta_{11}$	114.9339(18.8306)	71.9627(11.0122)	70.7831(7.6497)
	$theta_{30}$	0.3031(0.0074)	0.3949(0.0075)	0.3514(0.0053)
	$theta_{30}$	0.6862(0.0075)	0.5922(0.0075)	0.6367(0.0053)
	LL	-133.3251	-165.0994	-305.6453
Myopia test	LR Statistic	2.398	2.912	2.912
β=0 vs. β=.9999	Marginal Significance Level	0.1215	0.088	0.0018

(c)

```r
nfp <- function(used_group, beta = 0.9999, parameter = c(0,0), method="Nelder-Mead",
                start_0 = FALSE,remove_last = TRUE, H_method = "Richardson",precBits = 53,
                tol_EV = 0.001, output = FALSE){
  data <- transition_mat(used_group, remove_last, start_0)
  dt <- data[[1]]
  P0 <- data[[2]]
  P1 <- data[[3]]
  dim <- data[[4]]
  theta3_0 <- data[[5]]
  Eular_const <- - digamma(1)
  util <- function(theta, state){
    RC <- theta[1]
    theta1 <- theta[2]
    U_0 <- -0.001 * theta1 * state
    U_1 <- -RC
    U <- cbind(U_0, U_1)
    return(U)
  }
  contraction <- function(theta, beta, P0, P1, dt, state, precBits, tol_EV){
    U <- util(theta, state)
    EV_old <- matrix(0, nrow = dim, ncol = 2)
    diff <- 1000 + tol_EV
    if (output){
      i = 0
      print("RC")
      print(theta[1])
      print("theta1")
      print(theta[2])
      print("tol_EV")
      print(tol_EV)
    }
    while(diff > tol_EV){
      EV_0 <- Eular_const + P0 %*% log(rowSums(exp(U + beta * EV_old)))
      EV_1 <- Eular_const + P1 %*% log(rowSums(exp(U + beta * EV_old)))
      EV_new <- cbind(EV_0, EV_1)
      diff <- max(abs((EV_new - EV_old) / EV_old))
      if(max(EV_new) > 709){
        EV_new <- mpfr(EV_new, precBits = precBits)
        if (output) {print("EXCEEDED!!")}
      }
      EV_old <- EV_new
      if (output) {
        i = i + 1
        print(i)
        print(paste("diff", asNumeric(diff)))
      }
    }
    return(EV_old)
  }
  mat <- Vectorize(
    function(i,j,mat) {mat[i,j]},
```

```r
    vectorize.args = c("i", "j"))
    LL <- function(theta, beta, P0, P1, dt, state, precBits, tol_EV){
    EV <- contraction(theta=theta,
                      beta=beta,
                      P0=P0,
                      P1=P1,
                      dt=dt,
                      state=state,
                      precBits=precBits,
                      tol_EV=tol_EV)
    if (is.mpfr(EV)){
      EV <- asNumeric(EV)
    }
    U <- util(theta, state)
    V_CS <- U + beta * EV
    prob_C <- exp(V_CS) / rowSums(exp(V_CS))
    LL <- sum(log(mat(dt$mil, dt$replacement + 1, prob_C)))
    return(LL)
  }
  if (start_0) {
    state <- seq(0, dim-1)
  } else {
    state <- seq(1, dim)
  }
  theta <- parameter
  opt <- optim(theta, LL,
               beta=beta, P0=P0, P1=P1, dt=dt, state=state, precBits=precBits, tol_EV=tol_EV,
               control = list(fnscale=-1),
               method = method)
  if (output) {print("iteration completed")}
  theta_hat <- opt$par
  H <- numDeriv::hessian(func = LL, x = opt$par,
               beta=beta, P0=P0, P1=P1, dt=dt, state=state, precBits=precBits, tol_EV=tol_EV,
               method = H_method)
  se_hat <- sqrt(diag(solve(-H)))
  LL <- opt$value
  return(list(theta_hat = theta_hat, se_hat = se_hat,
              LL = LL, theta3_0 = theta3_0, obs = dt %>% nrow()))
}
```

```r
beta <- list(0.9999, 0.9999,0.9999, 0, 0, 0)
used_group <-  list(c(1,2,3), c(4), c(1,2,3,4), c(1,2,3), c(4), c(1,2,3,4))
cores <- detectCores()-1
cl <- makeCluster(cores)
registerDoParallel(cl)
Ts <- foreach(beta = beta, used_group = used_group,
              .packages = c("readxl", "tidyverse", "numDeriv", "Rmpfr", "gmp")) %dopar% {
  nfp(used_group = used_group, parameter = c(10, 3), tol_EV = 0.001, beta = beta,
      method = "Nelder-Mead")
}
stopCluster(cl)
```

```r
make_table(Ts, name = "c")
knitr::include_graphics("c.pdf")
```

TABLE IX
STRUCTURAL ESTIMATES FOR COST FUNCTION c(x, theta_1$) = .001$theta_{11}$x
FIXED POINT DIMENSION = 90

(Standard errors in parentheses)

Parameter		Data Sample		
Discount Factor	Estimates/Loglikelihood	Group 1, 2, 3	Group 4	Group 1, 2, 3, 4
		3864 Observation	4292 Observation	8156 Observation
β=.9999	RC	12.2809(2.0716)	10.2288(1.3917)	9.9364(0.9337)
	$theta_{11}$	5.1132(1.4338)	2.3066(0.5574)	2.666(0.4775)
	$theta_{30}$	0.3031(0.0074)	0.3949(0.0075)	0.3514(0.0053)
	$theta_{31}$	0.6862(0.0075)	0.5922(0.0075)	0.6367(0.0053)
	LL	-131.2127	-163.2717	-299.1956
β=0	RC	8.5578(0.7804)	7.7193(0.5942)	7.3824(0.3773)
	$theta_{11}$	114.9339(18.8306)	71.9627(11.0122)	70.7831(7.6497)
	$theta_{30}$	0.3031(0.0074)	0.3949(0.0075)	0.3514(0.0053)
	$theta_{30}$	0.6862(0.0075)	0.5922(0.0075)	0.6367(0.0053)
	LL	-133.3251	-165.0994	-305.6453
Myopia test	LR Statistic	4.224	3.656	3.656
β=0 vs. β=.9999	Marginal Significance Level	0.0398	0.0559	3e-04