Summary of Week 4

Contents

1 Pipeline to Generate Data

1.1 Extract the Parameters from Pre-trained Model

I run the latent space model `wages_simple3-9-21.stan` and name the new stan object with 7-13-21, then use the mean of each parameter sample as the estimate. No need to run this code chunk as a simplified file containing all parameter estimates `parm_fit_ifls_7-13-21.rda` has been created.

```r
setwd("~/Documents/Github/wages")
load('fit_ifls_7-13-21.rda')
#### Extract all parameter estiamtes from latent space model ####
all_parm <- matrix(NA, nrow = fit_ifls@sim$chains, ncol = length(fit_ifls@sim$samples[[1]]))
for (i in 1: nrow(all_parm)) {
  all_parm[i, ] <- sapply(fit_ifls@sim$samples[[1]], mean)
}
colnames(all_parm) <- names(fit_ifls@sim$samples[[1]])
all_parm <- colMeans(all_parm)
# save(all_parm, file = "parm_fit_ifls_7-13-21.rda")
```

1.2 Prepare Parameters for Data Generation

The parameters of worker and sector latent space, β_0^k, β_1^k, β_2^k, wage variance σ_i^2, sector effect ν_k from the pre-trained latent space model are used in the simulation. Not all parameters extracted are used.

```r
set.seed(8888)
N <- deflate_y_dat$N
TT <- deflate_y_dat$T
K <- deflate_y_dat$K
all_parm_names <- names(all_parm)
# Extract worker latent space
z <- matrix(all_parm[grep("z", all_parm_names)], ncol = 2)

# Extract sector latent space
w <- array(all_parm[setdiff(setdiff(grep("w", all_parm_names),
                                    grep("free_w", all_parm_names)),
                                    grep("tau", all_parm_names))],
           dim = c(K, TT, 2))

# Extract worker effect
mu <- all_parm[grep("mu", all_parm_names)]

# Extract beta
all_beta <- matrix(all_parm[setdiff(grep("beta", all_parm_names),
                                    grep("free_beta", all_parm_names))],
                   nrow = 3, byrow = T)
rownames(all_beta) <- paste("beta", 0: 2, sep = "")

# Extract edge probability between workers and sectors
p <- exp(array(all_parm[grep("p", all_parm_names, value = T)], dim = c(N, TT, K)))

# Extract sigma square for log wage distribution
sigmasq <- matrix(all_parm[grep("sigmasq", all_parm_names)], ncol = TT)
sigma <- matrix(all_parm[setdiff(grep("sigma", all_parm_names),
                                 grep("sigmasq", all_parm_names))],
                ncol = TT)
```

```
# Extract mean of firm effect
nu <- all_parm[grep("nu", all_parm_names)]

# Extract hypeparameters
tau_w0_idx <- grep("tau_w0", all_parm_names)
tau_w0 <- all_parm[tau_w0_idx]
tau_w_idx <- setdiff(grep("tau_w", all_parm_names),
                     grep("tau_w0", all_parm_names))
tau_w <- all_parm[tau_w_idx]
tau_k_idx <- grep("tau_k", all_parm_names)
tau_k <- all_parm[tau_k_idx]
tau_m_idx <- grep("tau_m", all_parm_names)
tau_m <- all_parm[tau_m_idx]
tau_idx <- setdiff(grep("tau", all_parm_names),
                   c(tau_w0_idx, tau_w_idx, tau_k_idx, tau_m_idx))
tau <- all_parm[tau_idx]

# Number of firms of each sector
L <- rep(30, K) # Each sector has 30 firms

# Hyperparameter of firm latent space
tau_x0 <- tau_w0
tau_x <- tau_w
```

Values of coefficients of edge probability between firm and sector (i.e. γ_0^l, γ_1^l, γ_2^l) might require some further consideration. At this stage, for all $l = 1, \dots, L_k$, I let γ_0^l is random sample from β_0^k, $\gamma_1^l = \beta_1^k$ and $\gamma_2^l = \beta_2^k$. Using normal random sample for all $\gamma^{l'}$s might sometimes lead to weird edge pattern. (e.g. 99% of workers in each sector work for only one firm over time)

```
all_gamma <- matrix(NA, nrow = 3, ncol = max(L))
all_gamma[, 1] <- 0
all_gamma[1, -1] <- sample(all_beta[1, -1], max(L)-1, replace = T)
all_gamma[2, -1] <- rep(all_beta[2, 2], max(L)-1)
all_gamma[3, -1] <- rep(all_beta[3, 2], max(L)-1)
rownames(all_gamma) <- paste("gamma", 0: 2, sep = "")
```

1.3 Data Generation

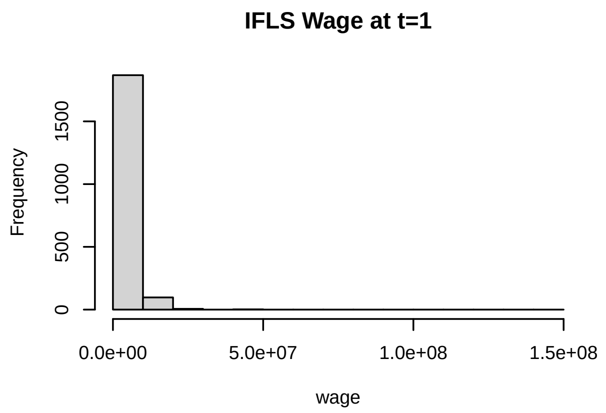
In the new data generation pipeline, in addition to using higher dimensional latent space and plugging in parameters from the pre-trained model, I modified some steps of data generation such that the simulated data can become closer to the real one. The first change is that instead of doing log normal sampling for wages with log mean as the direct sum of worker and firm effect, which always leads to wage sample significantly larger than the real data, I use a regression model: $log(Y_{i_{IFLS}}) = \alpha_1 \mu_i + \alpha_2 f_i^{(t)} + \epsilon_i$ and take the fitted values as the mean of the log normal distribution of the wages. I also reduce the variance parameters of wages with multiplicative factors different over years since those obtained from the model seem to be so large that the wage sample tend to be larger than real wage as well. The choice of the multiplicative factors is ad hoc and maybe a formal justification is needed later.

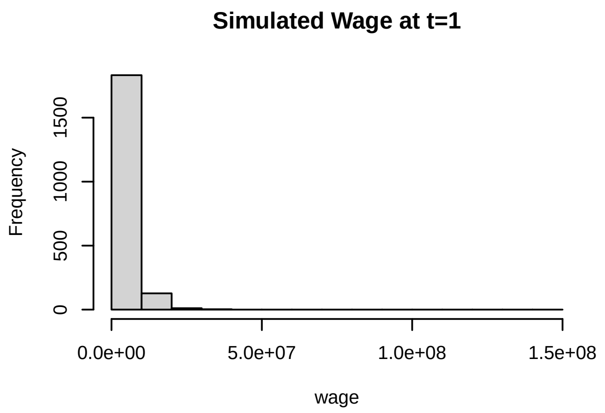
```
Y <- matrix(NA, N, TT) # wage
log_Y <- matrix(NA, N, TT) # log wage
firm_effect <- array(NA, dim = c(N, TT)) # firm effect
J <- matrix(0, N, (TT+1)) # sector indices over time
FF <- matrix(0, N, TT)
J_indicator <- array(0, dim = c(N, (TT + 1), K))  # sector indices in indicator form
```

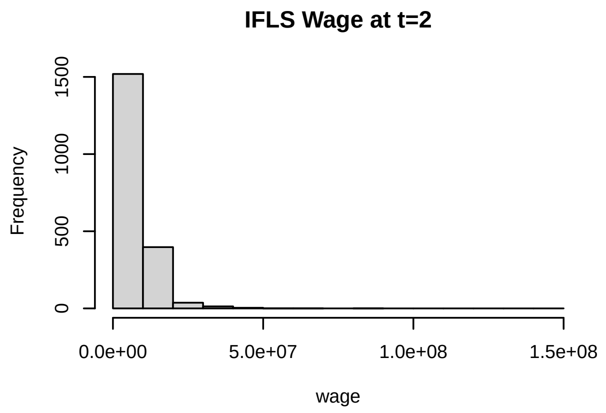
```r
F_indicator <- array(NA, dim = c(N, TT, max(L))) # firm indices in indicator form
x <- array(NA, dim = c(K, max(L), (TT + 1), 2)) # 2-d firm latent space

# Set some columns as NA for the case sectors have different number of firms
for (k in 1: K) {
  x[k, , , ][1: L[k], , ] <- 0
}

# Scale the variance of work wage distribution
correction_factor <- c(2, 2.3, 2.5)
sigma_corrected <- eachrow(sigma, correction_factor, "/")
case = "latent"
for (t in 1: TT) {
  dist_mat1 <- sqrt(matrix(rowsums(z^2), nrow = N, ncol = K) +
                      z %*% t(w[, t, ]) +
                      matrix(rowsums(w[, t, ]^2), nrow = N, ncol = K, byrow = T))
  beta0 <- all_beta[1, ]
  beta1 <- all_beta[2, ]
  beta2 <- all_beta[3, ]
  all_q <- exp(matrix(beta0, N, K, byrow = T) +
                 matrix(beta1, N, K, byrow = T)*J_indicator[, t, ] -
                 matrix(beta2, N, K, byrow = T)*dist_mat1)
  sector_of_worker <- apply(all_q/rowsums(all_q), 1,
                            function(prob){sample(1: K, 1, prob = prob)})
  J[, (t + 1)] <- sector_of_worker
  J_indicator[, (t + 1), ][cbind(1: N, sector_of_worker)] <- 1
  if(case == "latent"){
    temp_x <- x[, -1, (t + 1), ] # prepare for latent space at time t,
    temp_x_prev <- x[, -1, t, ]

    # sample for all firm latent space at time t, which are all independent
    temp_x[!is.na(temp_x)] <- rmvnorm(1, mu = temp_x_prev[!is.na(temp_x_prev)],
                                      sigma = 1/(as.numeric(t == 1)*tau_x0 +
                                              as.numeric(t != 1)*tau_x)*diag(sum(!is.na(temp_x))))
    x[, -1, (t + 1), ] <- temp_x
    dist_mat2 <- matrix(NA, nrow = K, ncol = max(L))
    for (i in 1: K) {
      dist_mat2[i, 1: L[i]] <- sqrt(rowsums(x[i, , (t + 1), ] -
                                              matrix(w[i, t, ], L[i], 2, byrow = T))^2)
    }
    firm_indicator <- matrix(as.numeric(J[, (t + 1)] == J[, t]), N, max(L))
    gamma0 <- all_gamma[1, ]
    gamma1 <- all_gamma[2, ]
    gamma2 <- all_gamma[3, ]
    all_q_firm <- exp(matrix(gamma0, N, max(L), byrow = T) +
                        matrix(gamma1, N, max(L), byrow = T)*firm_indicator -
                        matrix(gamma2, N, max(L), byrow = T)*dist_mat2[sector_of_worker, ])
    firm_of_worker <- apply(all_q_firm/rowSums(all_q_firm), 1,
                            function(prob){sample(which(!is.na(prob)), 1, prob = prob)})
    FF[, t] <- firm_of_worker
    firm_indicator[cbind(1: N, firm_of_worker)] <- 1
    F_indicator[, t, ] <- firm_indicator
  }
  firm_effect[, t] <- sapply(1: N, function(i){rnorm(1, nu[J[i, (t + 1)]],
                                                     sqrt(1/tau_k[J[i, (t + 1)]]))})
  if (t > 1){
    firm_effect[J[, (t + 1)] == J[, t], t] <- firm_effect[J[, (t + 1)] == J[, t], (t - 1)]
  }
  log_wage_IFLS <- log(deflate_y_dat$y[, t])
  fit_lm <- lm(log_wage_IFLS ~ mu + firm_effect[, t] - 1)
  log_wage_mean <- fitted(fit_lm)
  Y[, t] <- sapply(1: N, function(i){rlnorm(1, mean = log_wage_mean[i],
                                            sd = sigma_corrected[i, t])})
  log_Y[, t] <- log(Y[, t])
}

data_sim <- list(N = N, time.num = TT, K = K, firm.num = L, y = Y, sector = J, firm = FF)
# save(data_sim, file = "data_sim7-16-2021")
```

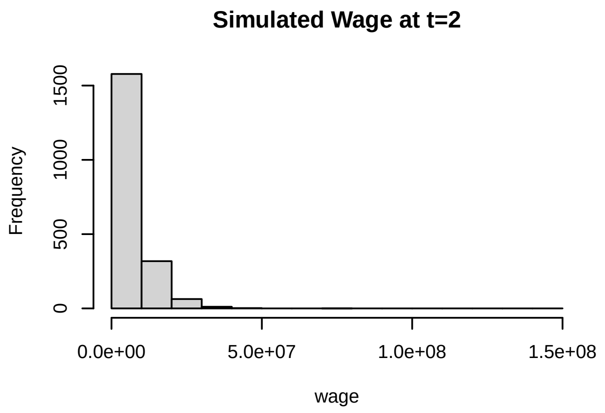
2 Comparison between Simulated and IFLS Data

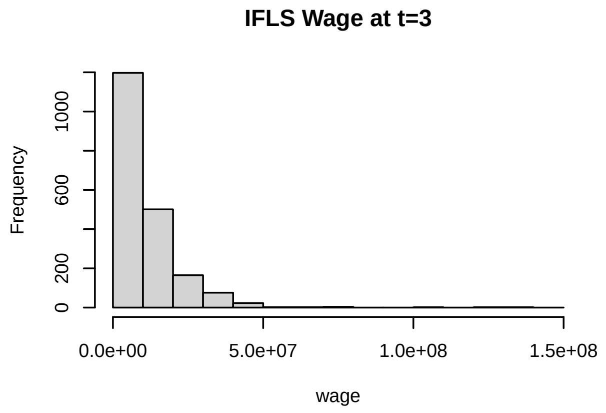
2.1 Comparison of Wage

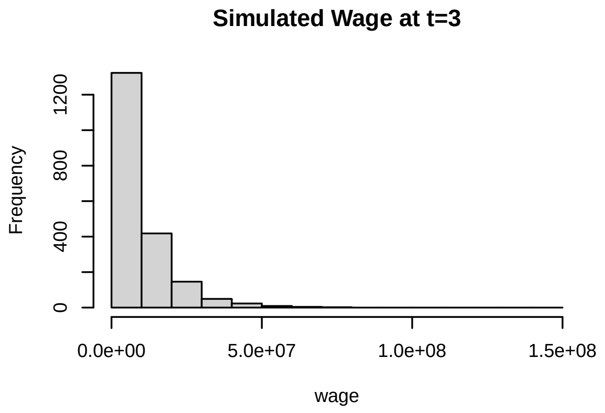




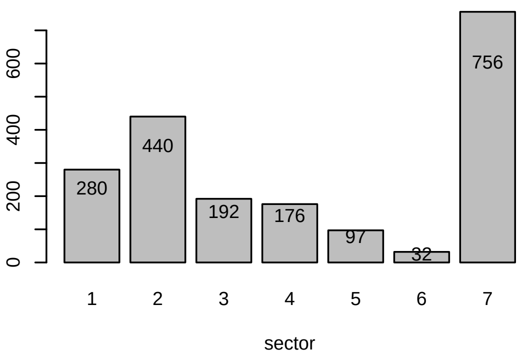


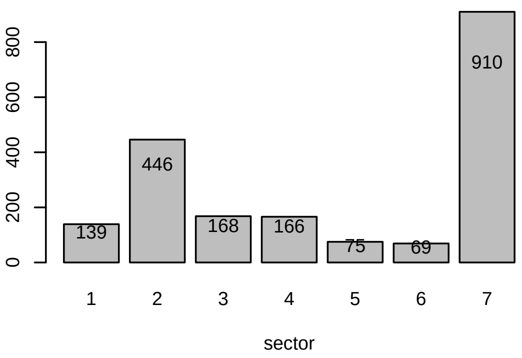


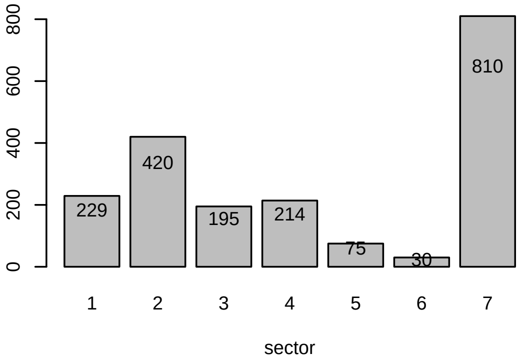


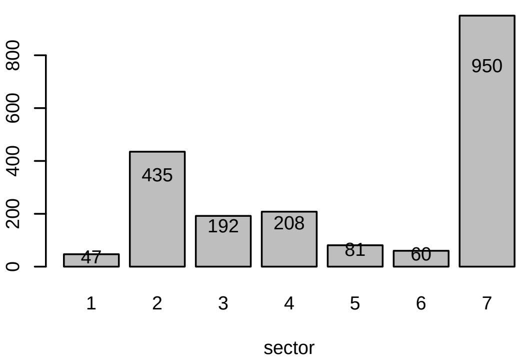


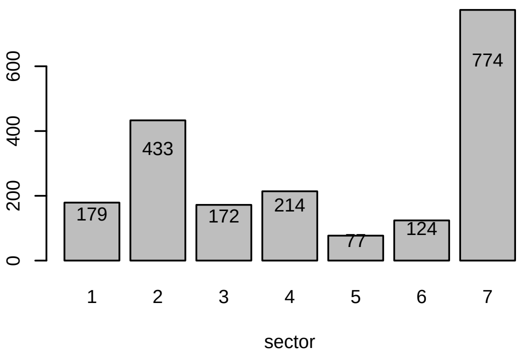
2.2 Comparison of Sector Distribution

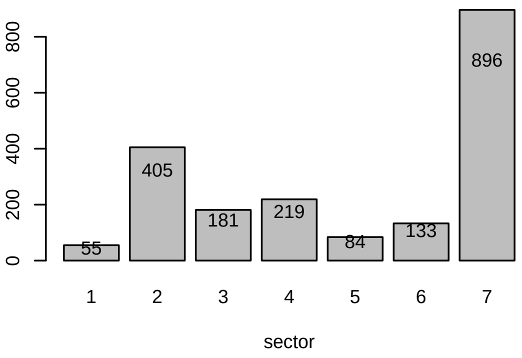

IFLS #workers in Sectors at t=1


Simulated #workers in Sectors at t=1

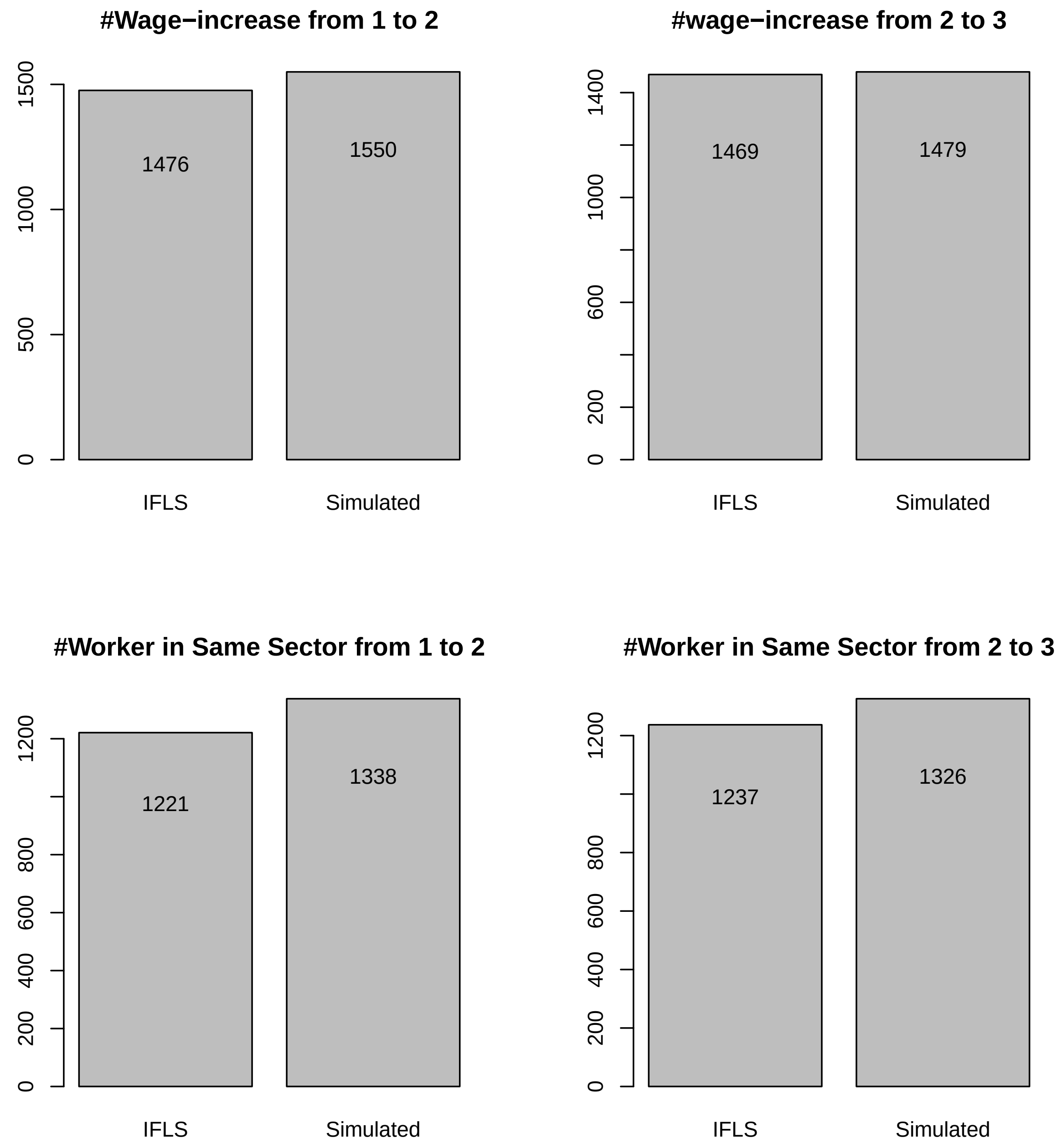

IFLS #workers in Sectors at t=2


Simulated #workers in Sectors at t=2


IFLS #workers in Sectors at t=3


Simulated #workers in Sectors at t=3

2.3 Comparison of Number of Workers with Wage Increase and Same Sector


#Wage−increase from 1 to 2
IFLS
Simulated
#wage−increase from 2 to 3
IFLS
Simulated
#Worker in Same Sector from 1 to 2
IFLS
Simulated
#Worker in Same Sector from 2 to 3
IFLS
Simulated