# Labor PS

## Question 1: Data Preparation and Analysis

### (a) Use the ACS data set that you have downloaded and cleaned up from IPUMS and plot the yearly mean wages, mean hours worked (unconditional and conditional) and mean employment rates of women aged 15-65 in the downloaded sample

In this question we will load the data of the ACS dataset from IPUMS. The data was cleaned using **Stata** and the do-file can be seen in the folder of this problem set. Since we used the same cleaned dataset to be able to do comparisions, we won't talk any further about the process of cleaning the data.

Now we start by loading some packages in **R** and the data in the code chunk below. Note that we have some comments that are worth reading.

```
# Packages
library(tidyverse) # Package for everything
library(haven)     # Package for reading dta files
library(ggthemes)  # Package for themes
library(lubridate) # Converts to date format
library(np)        # Package for non parametric and semiparametric
library(purrr)
library(ks)
library(Matrix)   # Faster computations of matrices

# set.seed(666)
set.seed(666)

# Importing data ----------------------------------------------------------

# Important note: we are dealing with a database that has already been cleaned.
# We will convert the file in a Rdata format so that we can load faster the
```

```
#data.

# Only use this option if you dont't have acess to the Rdata format and use the
# haven package
# data_ps <- read_dta(file = "data_PS1.dta")

# Convert to Rdata the data_ps
# saveRDS(data_ps, file = "data_ps1.rds")

# Load data
data_ps1 <- as_tibble(readRDS(file =  "data_ps1.rds"))
```

We now create new variables that we will use for the graphs of this problem. We create the variables *real_hhincome*, *real_wage*, *labor_par*, and *non_labor_income*. Respectively, each represents the real household income, the real anual wage, the labor participation, and the non labor income.

```
# Creates real values of wages
data_ps1 <- data_ps1 %>%
  mutate(real_hhincome = (hhincome*100) / Price_Index,
         real_wage = (incwage * 100) / Price_Index,
         labor_par = ifelse(empstat %in% c(1,2),1, ifelse( empstat == 3 ,0, NA)),
         non_labor_income = real_hhincome - real_wage)
```

And finally, with the following code chunk below, we generate the graphs that are below the code chunk. We won't explain much how to graph because the ggplot grammar of graphs is very easy to understand and do not require much explanation.

```
# For women women aged 15-65

# yearly mean wages total and per hour
year_mean_wage <- data_ps1 %>%
  filter(age %in% 15:65, sex == 2, uhrswork > 0,
         incwage < 999998, real_wage >=0 ) %>%
  group_by(year) %>%
  summarise(mean_wage = mean(real_wage, na.rm = T),
            mean_wage_hour = mean(wage_hour, na.rm = T))

# Plot mean_wage
plot_wages_year <- ggplot(year_mean_wage, aes(x = year
                           , y = mean_wage ))+
  geom_line(color = "#967BB6")+
```

```
  geom_point(color = "#7B1FA2")+
  theme_few()+
  labs(title = "Yearly mean real wage - Total",
       subtitle = "Women aged 15-65",
       x = "Year",
       y = "Mean wage")+
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle= element_text(hjust = 0.5),
        legend.position = "bottom") +
  scale_y_continuous(n.breaks = 10)+
  scale_x_continuous(n.breaks = 2019-2005)

# Plot mean_wage_hour
plot_wages_hour <- ggplot(year_mean_wage, aes(x = year
                                              , y = mean_wage_hour ))+
  geom_line(color = "#967BB6")+
  geom_point(color = "#7B1FA2")+
  theme_few()+
  labs(title = "Yearly mean real wage - per hour",
       subtitle = "Women aged 15-65",
       x = "Year",
       y = "Mean hourly wage")+
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle= element_text(hjust = 0.5),
        legend.position = "bottom") +
  scale_y_continuous(n.breaks = 10)+
  scale_x_continuous(n.breaks = 2019-2005)

# mean hours worked (unconditional and conditional)
  mean_hour_conditional <- data_ps1 %>%
    filter(age %in% 15:65, sex == 2, empstat == 1) %>%
    mutate(hour_worked = wkswork2*uhrswork) %>%
    group_by(year) %>%
    summarise(mean_hours = mean(hour_worked, na.rm = T))

  mean_hour_unconditional <- data_ps1 %>%
      filter(age %in% 15:65, sex == 2) %>%
      mutate(hour_worked = wkswork2*uhrswork) %>%
    group_by(year) %>%
      summarise(mean_hours = mean(hour_worked, na.rm = T))
```

```r
  # Plot hours worked
  plot_hour_worked_Cond <- ggplot(mean_hour_conditional, aes(x = year
                                                               , y = mean_hours ))+
    geom_line(color = "#967BB6")+
    geom_point(color = "#7B1FA2")+
    theme_few()+
    labs(title = "Yearly hours worked - Conditional",
         subtitle = "Women aged 15-65",
         x = "Year",
         y = "Mean hours worked")+
    theme(plot.title = element_text(hjust = 0.5),
          plot.subtitle= element_text(hjust = 0.5),
          legend.position = "bottom") +
    scale_y_continuous(n.breaks = 10)+
    scale_x_continuous(n.breaks = 2019-2005)


  plot_hour_worked_uncond <- ggplot(mean_hour_unconditional, aes(x = year
                                                                  , y = mean_hours ))+
    geom_line(color = "#967BB6")+
    geom_point(color = "#7B1FA2")+
    theme_few()+
    labs(title = "Yearly hours worked - Unconditional",
         subtitle = "Women aged 15-65",
         x = "Year",
         y = "Mean hours worked")+
    theme(plot.title = element_text(hjust = 0.5),
          plot.subtitle= element_text(hjust = 0.5),
          legend.position = "bottom") +
    scale_y_continuous(n.breaks = 10)+
    scale_x_continuous(n.breaks = 2019-2005)



# mean employment rates of women
  mean_employment <- data_ps1 %>%
    filter(age %in% 15:65, sex == 2) %>%
    group_by(year) %>%
    summarise(mean_employed = mean(labor_par, na.rm = T))

# Plot of labor participation of women
```

```r
plot_employment <-  ggplot(mean_employment, aes(x = year, y = mean_employed))+
   geom_line(color = "#967BB6")+
   geom_point(color = "#7B1FA2")+
   theme_few()+
   labs(title = "Yearly Employment Rate",
        subtitle = "Women aged 15-65",
        x = "Year",
        y = "Percent employed")+
   theme(plot.title = element_text(hjust = 0.5),
         plot.subtitle= element_text(hjust = 0.5),
         legend.position = "bottom") +
   scale_y_continuous(n.breaks = 10, labels = scales::percent)+
   scale_x_continuous(n.breaks = 2019-2005)
```

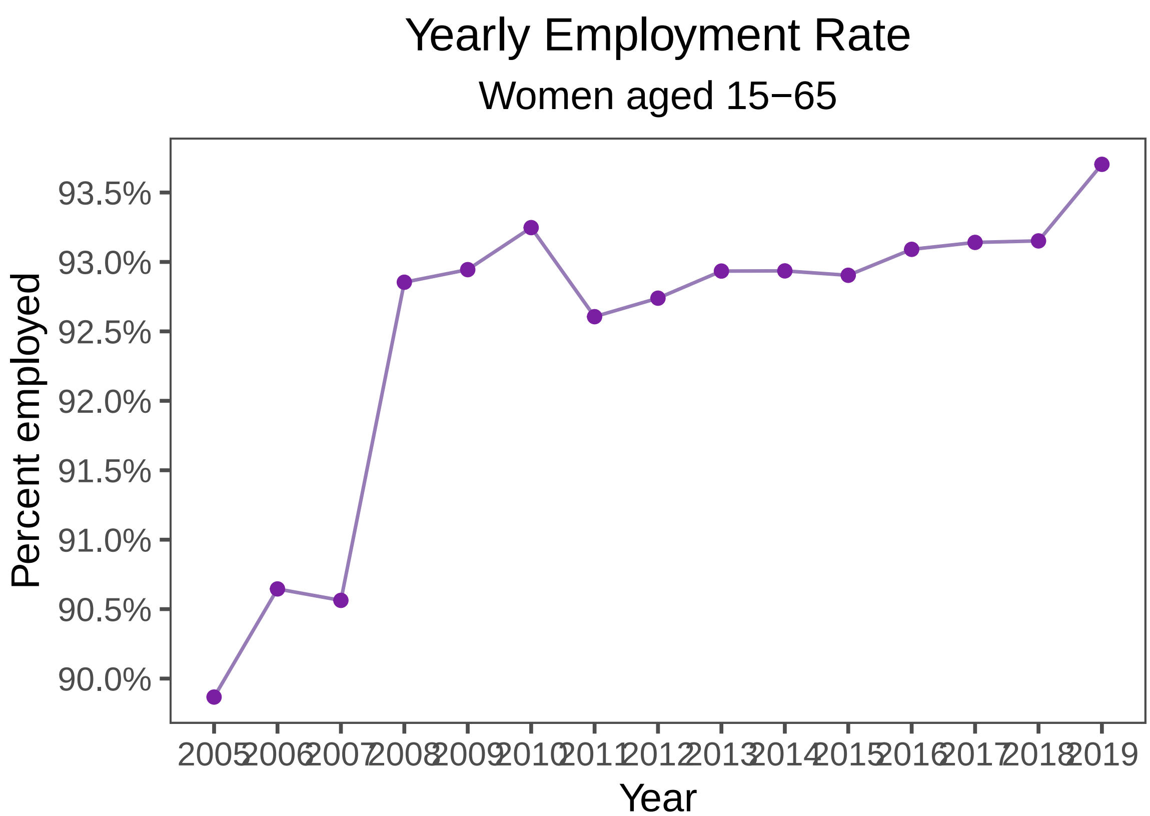


Figure 1: Mean employment - women 15-65

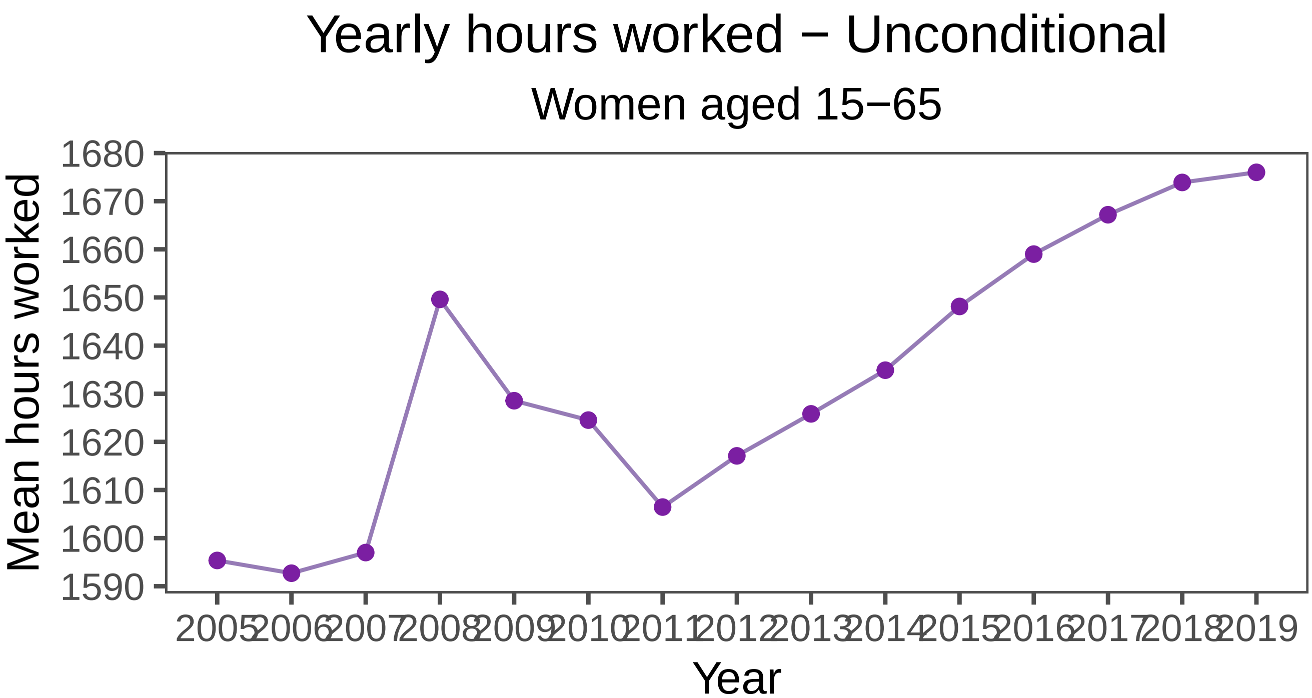


Figure 2: Hours worked per year -Unconditional - women 15-65

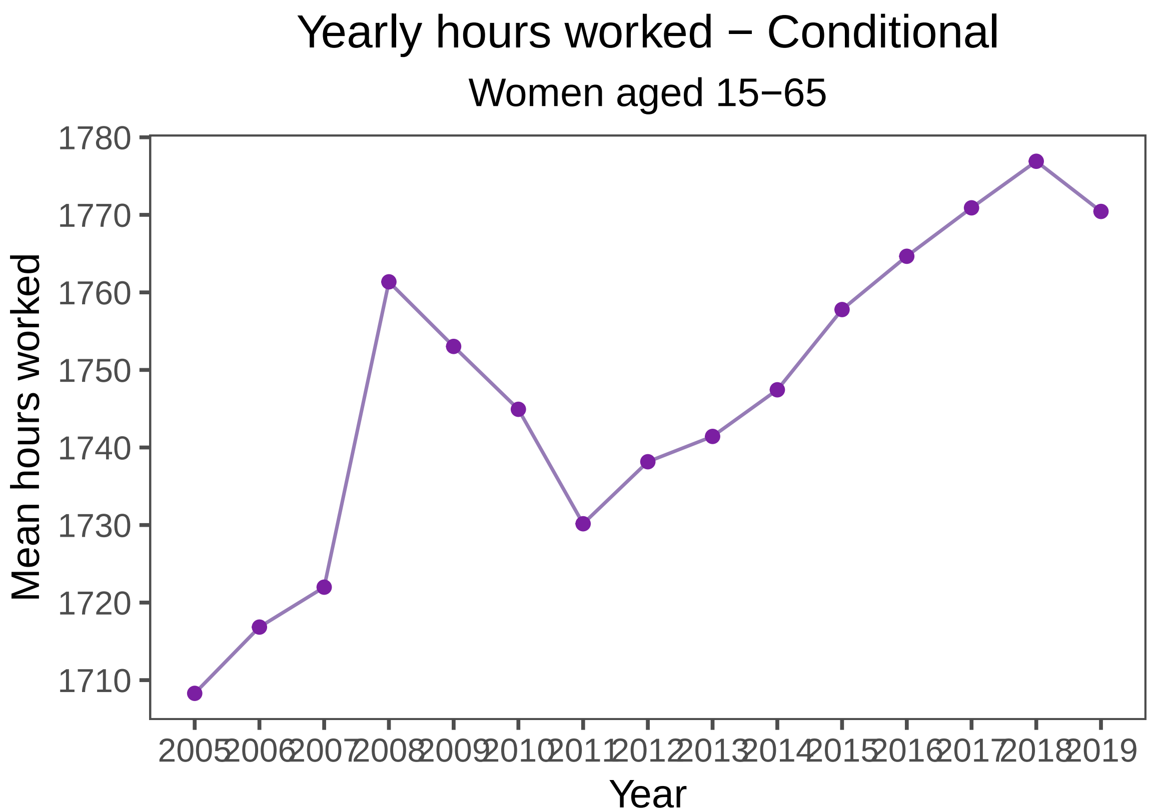


Figure 3: Hours worked per year -Unconditional - women 15-65


Figure 4: Hourly wage - women 15-65


Figure 5: Yearly total wage - women 15-65

**(b) Replicate Table 2 in Mincer (1962). Note: Make sure to impose comparable sample restrictions.**

### Question 2: Semi-Parametric, Structural Estimation

Suppose that the utility of the wife over participation and consumption follows the functional form:

$$U(C, P; \epsilon) = C + x'\alpha(1 - P) + \beta(1 - P)C + \epsilon(1 - P)$$

and the wage equation is

$$w(z, \xi) = z'\gamma + \xi$$

As shown in class, the observed wage equation can be written as

$$w(z, \xi) = z'\gamma + M(\Pr(P = 1 \mid y, z, x)) + u$$

Using the sub-sample of married women between the ages of 25 and 55, implement the following estimation steps

#### (a) Non-parametrically estimate $\Pr(P = 1 \mid y, z, x)$ using a kernel regression where $z$ includes completed education and age and $x$ includes a constant, age and current number of children

For this question, we have decided to use **R** with the **np** package and manually for the calculation. In R there is the package **np** written by Jeffrey S. Racine and Tristen Hayfield for the estimation of nonparametric (and semiparametric) kernel methods with built-in function we use for estimation. We also use **R** to manually calculate the results.

Initially we start by subsetting the data for women with the characteristics given by the question and we create a new definition of education to make it easier to work.

```
# Question 2 ------------------------------------------------------------

# Utility function
# U(c, P; e) = c + x'a*(1-P) + b(1-P)c + e(1-P)
# c: consumption
# Participation index (=1 if works)
# x: vector of covariates
# r threshold value


# 2 - a) Estimation Particiaption probabilities ---------------------------
# wage equation:
# w(z,eta) = z'y + M(Pr(P =1| y,z,x )) + u

# Subseting for married women between 25-55
# We also  subset to not include NA or negative
```

```
# values for each variable
  m_women_25_55 <- data_ps1 %>%
    filter(age %in% 25:55, marst %in% c(1,2), hhincome >=0,
           !is.na(non_labor_income),
           !is.na(age),
           !is.na(nchild)) %>%
    mutate(n_child = as.numeric(nchild),
           educ = as.numeric(educ)) %>%
    filter(educd > 1 & educd != 999) %>%  # drop if educd <= 1 or educd == 999
    mutate(education = -1) %>%  # create new column 'education' with -1 as initial value
    mutate(education = case_when(
      educd == 2 ~ 0,
      educd == 14 ~ 1,
      educd == 15 ~ 2,
      educd == 13 ~ 2.5,
      educd == 16 ~ 3,
      educd == 17 ~ 4,
      educd == 22 ~ 5,
      educd == 21 ~ 5.5,
      educd == 23 ~ 6,
      educd == 20 ~ 6.5,
      educd == 25 ~ 7,
      educd == 24 ~ 7.5,
      educd == 26 ~ 8,
      educd == 30 ~ 9,
      educd == 40 ~ 10,
      educd == 50 ~ 11,
      educd == 60 ~ 12,
      educd == 61 ~ 12,
      educd == 62 ~ 12,
      educd == 63 ~ 12,
      educd == 64 ~ 12,
      educd == 65 ~ 12,
      educd == 70 ~ 13,
      educd == 71 ~ 13,
      educd == 80 ~ 14,
      educd == 90 ~ 15,
      educd == 100 ~ 16,
      educd == 101 ~ 16,
      educd == 110 ~ 17,
      educd == 111 ~ 18,
```

```
        educd == 112 ~ 19,
        educd == 113 ~ 20,
        educd == 114 ~ 20,
        educd == 116 ~ 20,
        TRUE ~ education  # if none of the above conditions are true, keep existing value
      )) %>%
      filter(education != -1)
```

Since we do not have enough processing power, we will use the sub samples taken randomly without replacement. We will take two samples of size 1000, 5000.

```
# Take a subsample  because my laptop does not have enough computing
# power
n_1 = 1000 # Size of subsample 1
n_2 = 5000 # Size of subsample 2


# First subsample n= 1000
m_women_25_55_n_1 <- m_women_25_55[sample(nrow(m_women_25_55), n_1),] %>%
                                    mutate(educ = as.factor(educ),
                                           age = as.numeric(age),
                                           nchild = n_child)

# Second subsample n= 5000
m_women_25_55_n_2 <- m_women_25_55[sample(nrow(m_women_25_55), n_2),] %>%
  mutate(educ = as.factor(educ),
         age = as.numeric(age),
         nchild = n_child)
```

Now, we finally calculate the optimal bandwidth using the linear cross-validation and using a Gaussian mutivariate kernel function. The function perform 5 iterations of the whole process in order to find the the best bandwidth.

```
# Step 1: Non-parametric estimation of Pr(P = 1 | y,z,x) ------------------
# a)  Non-parametrically estimate Pr(P = 1|y, z, x) using a kernel regression
  # where z includes completed education and age and x includes a constant,
  # age and current number of children and y is non labor income

# Bandwidth estimation with Linear Cross-Validation
bw_par_n1 <- npregbw(formula = labor_par ~ non_labor_income  + educ + age + nchild ,
                  data = m_women_25_55_n_1, regtype = "lc",
                  ckertype = "gaussian")
```

```r
bw_par_n2 <- npregbw(formula = labor_par ~ non_labor_income  + educ + age + nchild ,
                     data = m_women_25_55_n_2, regtype = "lc",
                     ckertype = "gaussian")
```

With the following code we get the results

```r
# Results
resultados_n1 <- npreg(bw_par_n1) # first sample
resultados_n2 <- npreg(bw_par_n2) # second sample
```

Table 1: Non-parametric estimation bandwidth of P using np Package - local constant

| Sample | NL income $h$ | Educ $h$ | Age $h$ | nchild h |
|---|---|---|---|---|
| 1000 obs. | 8837.935 | 0.817675 | 4.98267 | 0.8327307 |
| 5000 obs. | 854086.6 | 0.5154519 | 3.6600053 | 2.836688 |

Now, we finally calculate the predicted probabilities of participation using the code chunk below. We also can see the table with descriptive statistics of predicted P. We note that falls in range 0 to 1, and mean of them are near 0.94 in every sample.

Table 2: Descriptive stats of the predicted P -np

| Sample | Mean | Sd | min | max |
|---|---|---|---|---|
| 1000 obs. | 0.9419313 | 0.06002464 | 0 | 1 |
| 5000 obs. | 0.9483481 | 0.02694107 | 0.9247883 | 0.9709707 |

Now, we do it again but manually. We will use the Epechnikov kernel and Silverman's Rule of Thumb.

We start by defining functions that will be used in this problem. We define the function to perform the epanechnikov_kernel that receives as input

```r
# Write auxiliary functions to manually calculate Nadaraya-Watson
# and the kernel estimator

# Function to calculate the multivariate Epanechnikov kernel
epanechnikov_kernel <- function(x, y, h) {
  d <- x - y
  t <- sqrt(diag(tcrossprod(d, solve(h) %*% d)))
  ifelse(t < 1, 0.75 * (1 - t^2), 0)
```

```
}

# define the multivariate Nadaraya-Watson estimator
nadaraya_watson <- function(X,y, H){
  # Number of observations
  n <- nrow(X)

  # Vector of predicted y
  y_hat <- rep(0, n)

  # Loop to calculate m(X - x_i)
  for (i in 1:n) {
    kernel_weights <- rep(0, n)
    for (j in 1:n) {
      kernel_weights[j] <- epanechnikov_kernel(X[j,], X[i,], H)
    }
    y_hat[i] <- sum(kernel_weights * y) / sum(kernel_weights)
  }

  # Print the estimated values of y
  y_hat
}


# Estimate the optimal bandwidth using Silverman's rule of thumb
Silverman <- function(X){
  n <- nrow(X) # Number of rows
  d <- ncol(X) # Number of columns
  std <- apply(X, 2, sd)  # Sd
  H <- diag(d) * std * (4 / (d + 2))^(1/(d + 4) ) * (n ^(-1*(1/(d+4))))
  return(H)
  } # Returns the diagonal matrix for bandwidth
```

Then, we define the vectors and matrices X, Y, Z and P (predicted participation probability) to represents the variables given by the question and add the suffix n1 or n2 to represent. See the comments in the code for a complete description of the name of each variable. Also note T is column bind of matrices X,Y,Z„ to represent observed labor particpation we use L and H is the Silverman's Bandwidth.

```
# Subset matrices
# X variables age education
```

```
X_n1 <- as.matrix(cbind(const = rep(1, times = nrow(m_women_25_55_n_1)),m_women_25_55_n_1[
X_n2 <- as.matrix(cbind(const = rep(1, times = nrow(m_women_25_55_n_2)),m_women_25_55_n_2[

# Z variables completed education and age
Z_n1 <- as.matrix(m_women_25_55_n_1[, c("age", "education")])
Z_n2 <- as.matrix(m_women_25_55_n_2[, c("age", "education")])

# Y non labor income
Y_n1 <- as.vector(m_women_25_55_n_1[,"non_labor_income"])
Y_n2 <- as.vector(m_women_25_55_n_2[,"non_labor_income"])

# Matrix containing everything to estimate P
T_n1 <- as.matrix(cbind(X_n1[,-1], Z_n1, Y_n1))
T_n2 <- as.matrix(cbind(X_n2[,-1], Z_n2, Y_n2))

# Matrix of labor participation
L_n1 <- as.vector(m_women_25_55_n_1[,"labor_par"])
L_n2 <- as.vector(m_women_25_55_n_2[,"labor_par"])

# bandwith Matrix according to Silverman
H_n1 <- Silverman(T_n1)
H_n2 <- Silverman(T_n2)

# Estimation of P
P_n1 <- nadaraya_watson(X = T_n1, y = L_n1, H = H_n1)
P_n2 <- nadaraya_watson(X = T_n2, y = L_n2, H = H_n2)
```

We get the following bandwidths and descriptive statistics for the estimation

Table 3: Non-parametric estimation Silverman bandwidth of P using manual - local constant

| Sample | Non-labor income $h$ | Educ $h$ | Age $h$ | nchild $h$ |
|---|---|---|---|---|
| 1000 obs. | 55729.29 | 1.515574 | 3.726225 | 0.5274278 |
| 5000 obs. | 60356.78 | 1.211947 | 3.077618 | 0.4272879 |

Table 4: Descriptive stats of the predicted P - manual

| Sample | Mean | Sd | min | max |
|---|---|---|---|---|
| 1000 obs. | 0.952886 | 0.02333114 | 0.90723 | 0.98723 |
| 5000 obs. | 0.9488085 | 0.01610487 | 0.9025235 | 0.9697569 |

**(b) Take your predicted working probabilities estimated in part (a) $\widehat{\Pr}(P = 1 \mid y, z, x)$ in the sample over which you implemented the non-parametric regression in part (a). Use Robinson's partial regression model to estimate $\gamma$ and $M$. See pg. 62 in the Nonparametrics6.pdf file located in the Readings subfolder of the shared Dropbox folder.**

In **R** we can easily perform the Robinson's Partial Regression by using the following command:

```
# Semi parametric model
robinson_reg_n1 <- npplreg(as.numeric(real_wage) ~ educ + age | predicted_p, data = m_wome
robinson_reg_n2 <- npplreg(as.numeric(real_wage) ~ educ + age | predicted_p, data = m_wome
robinson_reg_n3 <- npplreg(as.numeric(real_wage) ~ educ + age | predicted_p, data = m_wome

summary(robinson_reg_n1)
summary(robinson_reg_n2)
summary(robinson_reg_n3)
```

Table 5: Estimation of gamma coefficients using np

| Variable | 1000 obs. | 5000 obs. |
|---|---|---|
| educ | 2934.344 | 2613.985 |
| age | 141.0934 | -372.652 |

For the manual case we define functions for the univariate case o local constant estimation because our original functions were for multivariate form. We start by defining the Epanechnikov kernel in univariate case, and the nadaraya-watson estimator in univariate case.

```
# We will need to define additional functions to perform in the univariate case
# Define the kernel function (Epanechnikov kernel)
epa_uni <- function(x, x0, h) {
  u <- abs((x - x0) / h) # argument
  k <- 0.75 * (1 - u^2) * as.numeric(abs(u) <= 1) # value
  return(k)
}

# Univariate NW Regression
nw_regression <- function(x0, x, y, h) {
  k <- epa_uni(x, x0, h)
  sum(k * y) / sum(k)
}
```

We then define the bandwidth for each sample. as we can see below in the code.

```
# Bandwidth for P
HP_1 <-  sd(P_n1) * (4 / (1 + 2))^(1/(1 + 4) ) * (1000 ^(-1*(1/(1+4))))
HP_2 <-  sd(P_n2) * (4 / (1 + 2))^(1/(1 + 4) ) * (5000 ^(-1*(1/(1+4))))
```

Table 6: Silverman's Bandwidth for predicted P - manual

| Bandwidth | 1000 Obs. | 5000 Obs. |
|---|---|---|
| h | 0.006207599 | 0.003105639 |

We use the function sapply and nw_regression to non-parametrically regress each variable and estimate it to obtain $\hat{g}_t, t \in \{y, X\}$ that will be used to to calculate $e_y = y - g_y$ and $e_X = X - g_x$.

```
# First Calculate gy
# First sample
gy_n1 <- sapply(P_n1, nw_regression, x = P_n1, y = W_n1, h = HP_1 )
plot(P_n1, W_n1)
lines(P_n1, gy_n1, col = "blue", lwd = 2)

# Second sample
gy_n2 <- sapply(P_n2, nw_regression, x = P_n2, y = W_n2, h = HP_2)
plot(P_n2, W_n2)
lines(P_n2, gy_n2, col = "blue", lwd = 2)

# Second: calculate gx
gx_n1 <- as.matrix(cbind(sapply(P_n1, nw_regression, x = P_n1, y = Z_n1[,1], h = HP_1 ), s
gx_n2 <- as.matrix(cbind(sapply(P_n2, nw_regression, x = P_n2, y = Z_n2[,1], h = HP_2 ), s


# Calculate residual e_y = y - g_y, e_x = X- g_x
ey_n1 <- W_n1 - gy_n1 # First sample
ex_n1 <- Z_n1 - gx_n1

ey_n2 <- W_n2 - gy_n2 # Second samples
ex_n2 <- Z_n2 - gx_n2
```

We plot below for each variable and sample the nonparametric regression.

With the calculated residuals we regress them to obtain $\hat{\gamma}$ manually.

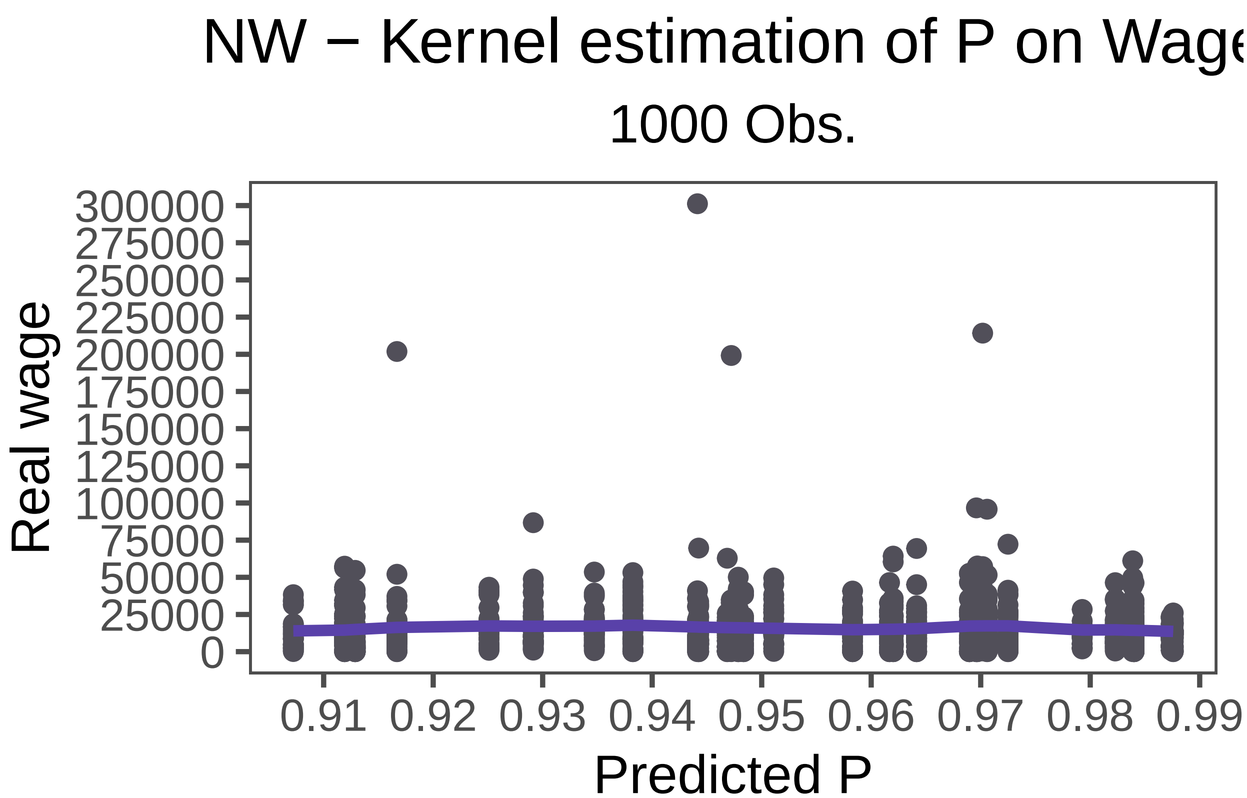


Figure 6: NP regression - Wages on P (1000 obs.)

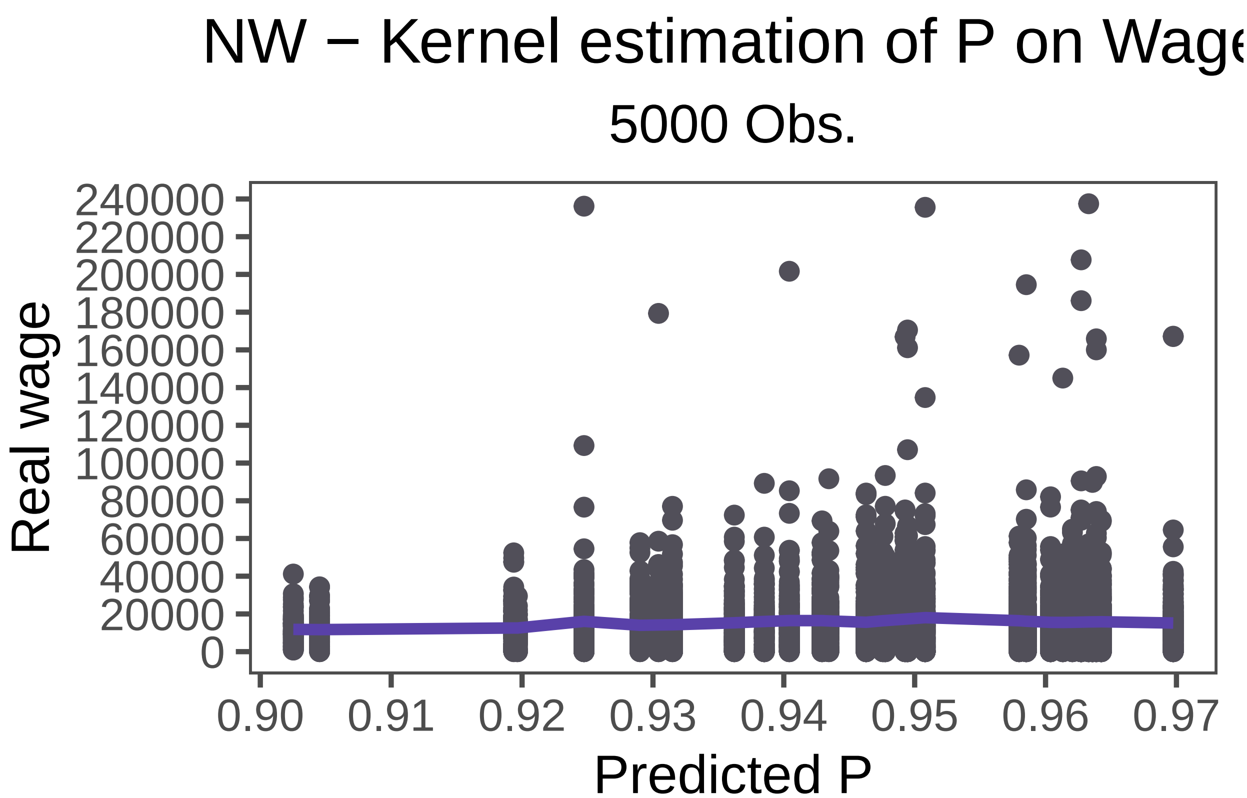


Figure 7: NP regression - Wages on P (5000 obs.)

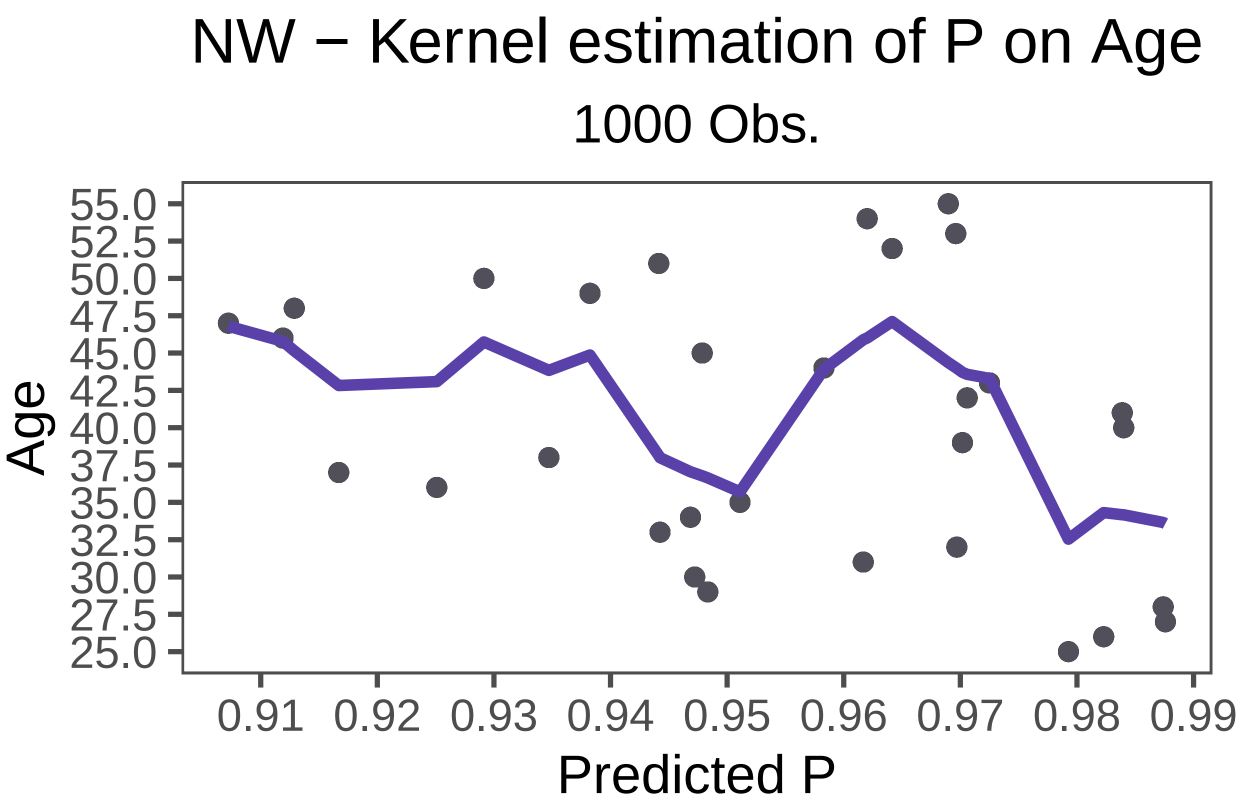


Figure 8: NP regression - Age on P (1000 obs.)

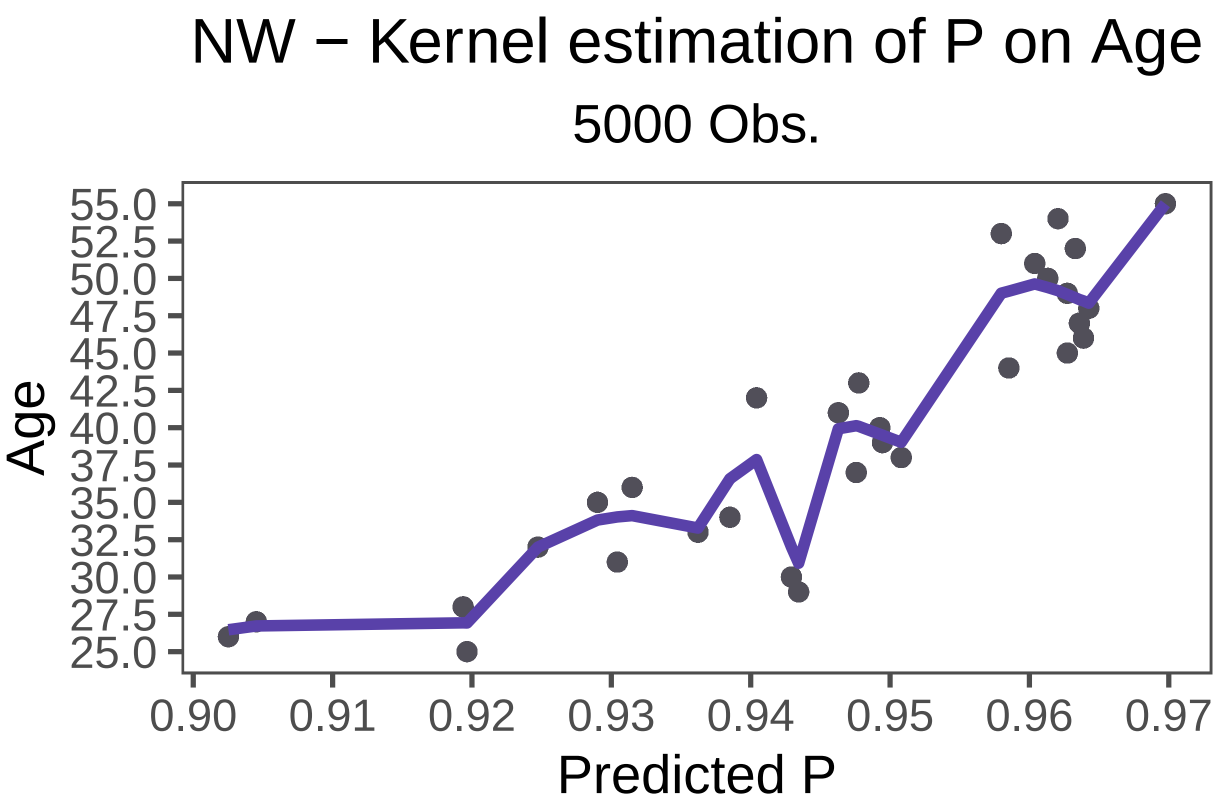


Figure 9: NP regression - Age on P (5000 obs.)

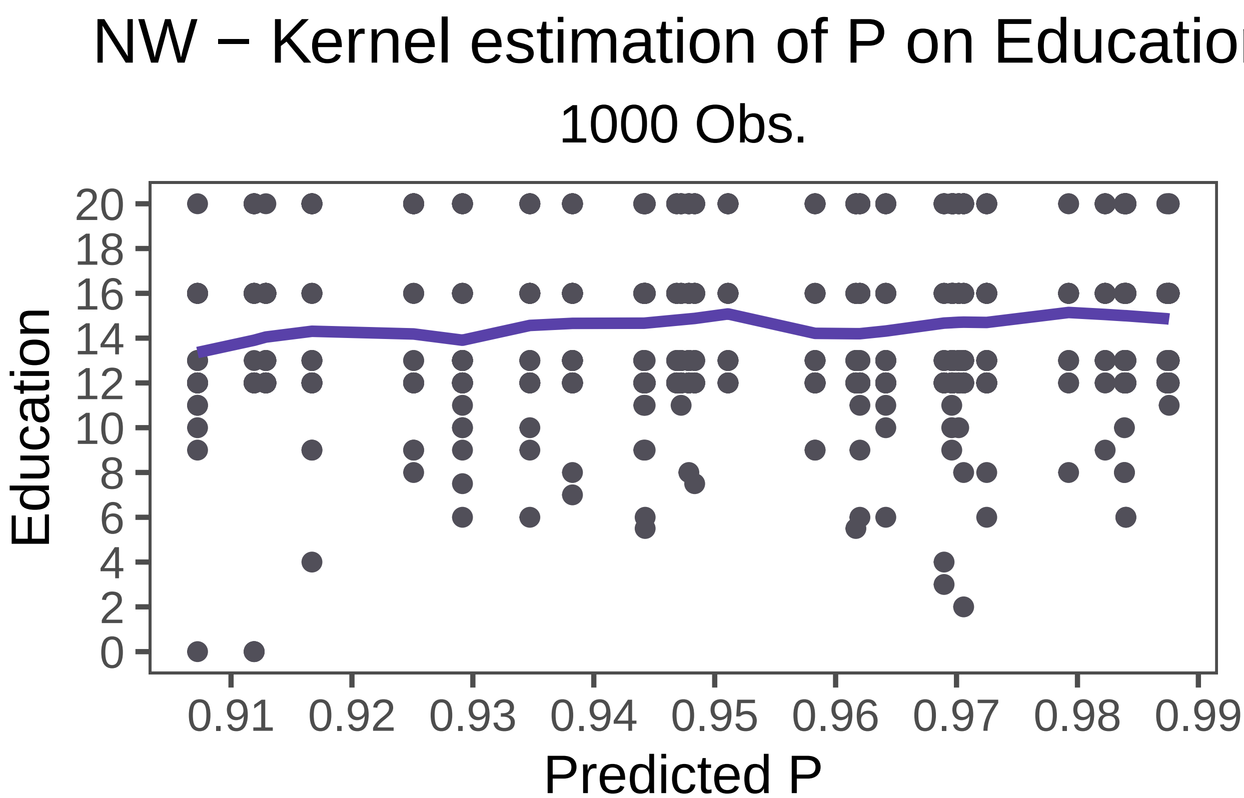


Figure 10: NP regression - Education on P (1000 obs.)

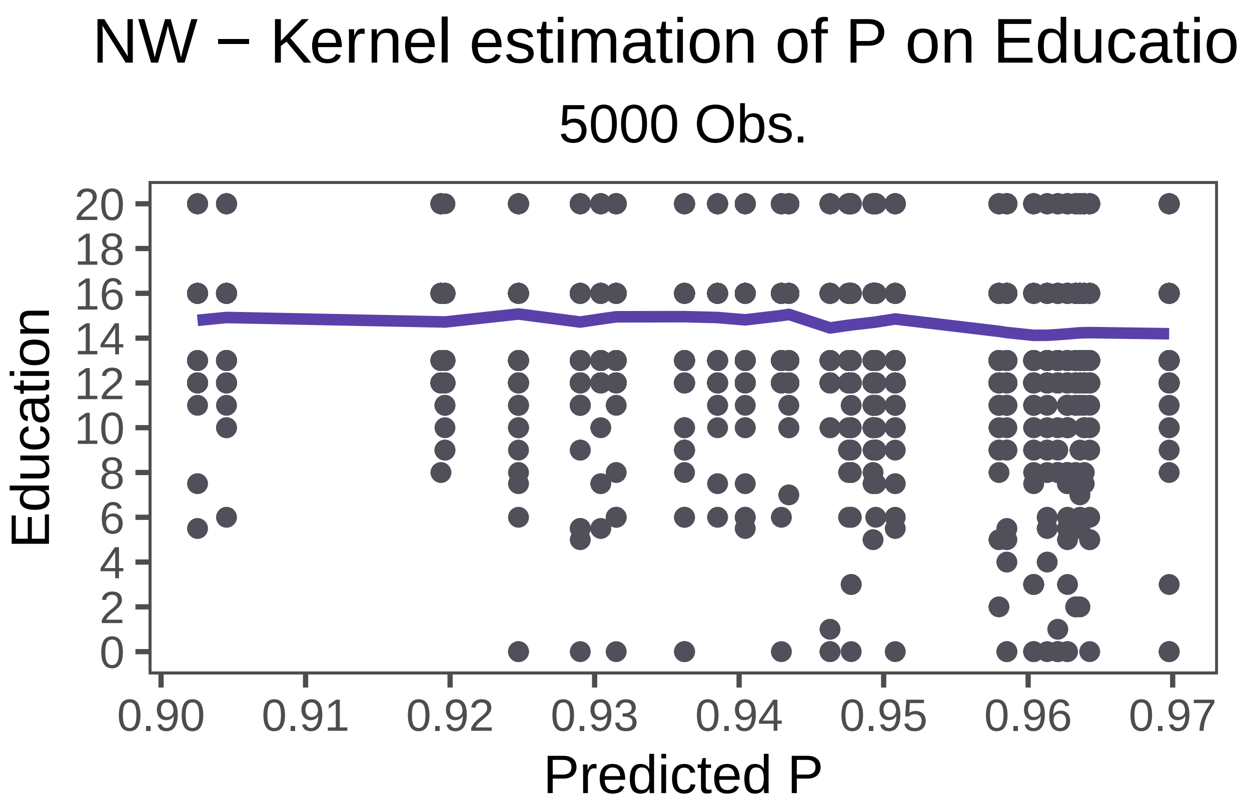


Figure 11: NP regression - Education on P (5000 obs.)

```r
# Finaly, OLS residuals
gamma_n1 <- solve(t(ex_n1) %*% ex_n1) %*% t(ex_n1) %*% ey_n1
gamma_n2 <- solve(t(ex_n2) %*% ex_n2) %*% t(ex_n2) %*% ey_n2
```

We get the following results:

Table 7: Estimation of gamma coefficients manually

| Variable | 1000 obs. | 5000 obs. |
|---|---|---|
| educ | 1928.1576 | 1541.002 |
| age | 173.31590 | 38.54569 |

**(c) Use your estimates for $\gamma$ obtained in part (b) to estimate $\alpha$ and $\beta$ using the Klein and Spady single index estimator. Be careful to adjust the estimator to account for the fact that $\Pr(P = 1 \mid y, z, x) = 1 - F\left(x'\alpha + \beta y - \hat{\gamma}' z\right)$.**

The built-in Klein and Spady estimation does not permit to change the betas, so for this question we will only manually perform the estimation using a loglikelihood approach.

**(d) Do you fail to reject the theoretical implications of the model? Discuss.**

## Question 3: (Fully) Parametric, Structural Estimation

Under the assumption that $\epsilon$ and $\xi$ follow a bivariate normal distribution

$$\binom{\epsilon}{\xi} \sim \mathcal{N}\left(\left[\begin{array}{l}0 \\ 0\end{array}\right],\left[\begin{array}{cc}\sigma_\epsilon^2 & \rho \sigma_\epsilon \sigma_\xi \\ \rho \sigma_\epsilon \sigma_\xi & \sigma_\xi^2\end{array}\right]\right)$$

**(a) Estimate the model using full information maximum likelihood (that is, deriving a likelihood function in which we are estimating the utility function and the wage equation jointly).**

In this exercise we will use the estimation procedure in the slides of Static Labor Participation Estimation. In that specific slide we have:

"… the likelihood can be written in the following way:

$$\mathcal{L}(\theta ; \text { data })=\prod_{i=1}^N\left(\operatorname{Pr}\left(P_i=1, w=w_i \mid y_i, \mathbf{z}, \mathbf{X}\right)\right)^{P_i}\left(\operatorname{Pr}\left(P_i=0 \mid y_i, \mathbf{z}, \mathbf{X}\right)\right)^{1-P_i}$$

where:

$$\operatorname{Pr}\left(P_i=0 \mid y_i, \mathbf{z}, \mathbf{X}\right)=\Phi\left(\frac{\mathbf{X}_i^{\prime} \alpha+\beta y_i-\mathbf{z}^{\prime} \gamma}{\sigma_\nu}\right)$$

with $\sigma_\nu^2=\sigma_\epsilon^2+\sigma_\xi^2-2 \rho \sigma_\epsilon \sigma_\xi$, and

$$\begin{aligned}
& \operatorname{Pr}\left(P_i=1, w=w_i \mid y_i, \mathbf{z}, \mathbf{X}\right)=\frac{1}{\sigma_\xi} \phi\left(\frac{w_i-\mathbf{z}^{\prime} \gamma}{\sigma_\nu}\right) \\
& \times \Phi\left(\frac{w_i-\mathbf{X}_i^{\prime} \alpha-\beta y_i-\rho \frac{\sigma_\epsilon}{\sigma_\xi}\left[w_i-\mathbf{z}_i^{\prime} \gamma\right]}{\sigma_\epsilon \sqrt{1-\rho^2}}\right)
\end{aligned}$$

" In our context, if we let $X_i^{\prime}=[1, age_i, nchild_i]^{\prime}$, $y_i$ is non-labor income and $z_i^{\prime}=[age_i, education_i]$,and $\Phi(.), \phi(.)$ represents the cdf and density of a standard normal.

We do it by first defining the log_likelihood for each sample as we see below:

```
# 3)Under the assumption that epsilon and  follow a bivariate normal distribution
# 3-a) full information maximum likelihood --------------------------------
#  Estimate the model using full information maximum likelihood
# (that is, deriving a likelihood function in which we are estimating
# the utility function and the wage equation jointly).
```

```r
# Z = educ and age
# X = const, age , nchild
# y = non labor income

# Loglikelihood P-N First sample of 1000bs
log_likelihood_n1 <- function(par){

  # Dictionary
  # par[1] = alpha_1
  # par[2] = alpha_2
  # par[3] = alpha_3
  # par[4] = beta
  # par[5] = gamma_1
  # par[6] = gamma_2
  # par[7] = sigma_e
  # par[8] = sigma_xi
  # par[9] = rho


  # sigma_v
  sigma_v <- sqrt(par[7]^2 + par[8]^2 - 2*par[9]*par[7]*par[8])

  # Here we define the Argument of Pr(P=0)
  arg_pr_0 <- (par[1] + par[2]*X_n1[,2] + par[3]*X_n1[,3] + par[4]*Y_n1 - par[5]*Z_n1[,1]

  # Define we properly define Pr(0)
  pr_0 <- pnorm(arg_pr_0)
  pr_0 <- exp(pr_0) / (1 + exp(pr_0))

  # Here we define Arg 1 of Pr(1), the argument of the density normal
  arg_pr_1 <- (W_n1 - par[5]*Z_n1[,1] - par[6]*Z_n1[,2]) / sigma_v

  # Here we define the Arg 2 of Pr(1), the argument of the cdf normal
  arg_pr_2 <-  W_n1 - par[1] - par[2]*X_n1[,2]- par[3]*X_n1[,3] -par[4]*Y_n1
  arg_pr_2 <-  arg_pr_2 -(par[9]*par[7]/sigma_v)*(W_n1-par[5]*Z_n1[,1]-par[6]*Z_n1[,2])
  arg_pr_2 <- arg_pr_2 / (par[7]*sqrt(1-par[9]^2))

  # Values of Pr(1)
  pr_1 <- (1/par[8]) * dnorm(arg_pr_1) * rnorm(arg_pr_2)
  pr_1 <- exp(pr_1)/(1 + exp(pr_1))
```

```
  # Objective function
  log_like = sum(L_n1*log(pr_1) + (1 - L_n1)*(log(pr_0)))
  return(log_like)
}
```

Now we define the same function but for the sample of 5000 obs.

```
log_likelihood_n2 <- function(par){

  # Dictionary
  # par[1] = alpha_1
  # par[2] = alpha_2
  # par[3] = alpha_3
  # par[4] = beta
  # par[5] = gamma_1
  # par[6] = gamma_2
  # par[7] = sigma_e
  # par[8] = sigma_xi
  # par[9] = rho


  # sigma_v
  sigma_v <- sqrt(par[7]^2 + par[8]^2 - 2*par[9]*par[7]*par[8])

  # Arg pr_0
  arg_pr_0 <- (par[1] + par[2]*X_n2[,2] + par[3]*X_n2[,3] + par[4]*Y_n2 - par[5]*Z_n2[,1]

  # Define Pr(0)
  pr_0 <- pnorm(arg_pr_0)
  pr_0 <- exp(pr_0) / (1 + exp(pr_0))

  # Arg 1 of pR(1)
  arg_pr_1 <- (W_n2 - par[5]*Z_n2[,1] - par[6]*Z_n2[,2]) / sigma_v

  # Arg 2 de pr2
  arg_pr_2 <-  W_n2 - par[1] - par[2]*X_n2[,2]- par[3]*X_n2[,3] -par[4]*Y_n2
  arg_pr_2 <-  arg_pr_2 -(par[9]*par[7]/sigma_v)*(W_n2-par[5]*Z_n2[,1]-par[6]*Z_n2[,2])
  arg_pr_2 <- arg_pr_2 / (par[7]*sqrt(1-par[9]^2))

  # Pr(1)
  pr_1 <- (1/par[8]) * dnorm(arg_pr_1) * rnorm(arg_pr_2)
```

```r
  pr_1 <- exp(pr_1)/(1 + exp(pr_1))

  # Objective
  log_like = sum(L_n2*log(pr_1) + (1 - L_n2)*(log(pr_0)))
  return(log_like)
}
```

Finally with following commands we optimize the log likelihood function in order to find the parameters. For initial values we could use the *lucky guess* which are values we previously estimated and know are near the optimum or we can start by using a random draw of number between -100 and 100 and keep iterating the process until we converge to the desired point.

```r
# Guesses
lucky_guess <- c(9.156, 0.858, -0.609, 1.265, 1232.229, 60.274,100, 100, 0)
random_guess <- sample(-100:100, 9)


#  See if has correct number of parameters =9
length(lucky_guess)
length(random_guess)

# Optim with luck_guess
optim(lucky_guess, log_likelihood_n1, method = "Nelder-Mead")
optim(lucky_guess, log_likelihood_n2, method = "Nelder-Mead")

# Optim with random_guess
optim(random_guess, log_likelihood_n1, method = "Nelder-Mead")
optim(random_guess, log_likelihood_n2, method = "Nelder-Mead")
```

Table 8: Estimation Parametric-Structural

| Parameter | 5000 Obs. | 1000 Obs |
|---|---|---|
| Con$\alpha_1$ | 56.35703 | 49.70609 |
| Age $\alpha_2$ | 47.66128 | -26.41636 |
| nchild $\alpha_3$ | 44.87264 | 34.93082 |
| Non-Labor Income $\beta$ | -219.79045 | -149.24950 |
| Age $\gamma_1$ | 1277.57623 | 1322.04343 |
| education $\gamma_2$ | 113.59084 | 145.01691 |
| $\sigma_\epsilon$ | 145.38495 | 122.68997 |
| $\sigma_\chi$ | 149.16555 | 103.24569 |
| $\rho$ | 40.25317 | 103.24569 |

| Parameter | 5000 Obs. | 1000 Obs |
|---|---|---|
| Optim Value | -3465.75 | -693.1539 |

**(b) Compare your results with those obtained in Question 2. Discuss.**

## Question 4: Marshallian Labor Supply

Take the model we considered in the Estimation Lecture of the Static Intensive Labor Supply: Let the direct utility follow a Stone-Geary form

$$U = B_0 \ln (L - \gamma_L) + B_1 \ln (C - \gamma_C)$$

where

$$\begin{gathered} B_0 + B_1 = 1 \\ C_i - \gamma_C > 0; \quad L - \gamma_L > 0 \\ B_0 = x' \tilde{B}_0 + \epsilon \end{gathered}$$

**(a) Suppose that the price of consumption is 1. What are the Marshallian demand functions for consumption and leisure?**