Wage Inequality Decomposition for Quantiles

Jieyi Chen

June 2, 2022

Abstract

The project reviews methods on detailed decomposition for quantiles such as IPW reweighting, conditional quantile regression, and RIF regression. An approach combining reweighting and RIF regression is applied to show the white and nonwhite wage decomposition on 10th, 50th, and 90th quantile. By keeping common support and using a reweighting method, the procedures are more reliable and less computational intensive.

1 Introduction to the Problem

Although the wage gap is converging in the long term, the gap is still substantial and findings are mixed. Trying to understand the underlying factors, a branch of the research focuses on the distributional issues of the wage gap with a special focus on quantiles.

The analysis is important in terms of policy and empirical work. It is of policy interest because many programs and interventions focus on people of a certain percentile range and thus have distributional impacts. Plus, the quantile difference in inequality partly explains the persistence wage gap. For example, the gender wage gap review paper by Blau and Kahn (2017) has concluded an increase in terms of overall female-male wage ratio but a decrease in its ninetieth percentile ratio from 1980 to 2010 in both PSID and March CPS data.

In order to tease out the important factors accounting for differences in the distribution, the computational problem is how to decompose wage inequality for quantiles. The problem has been continuously discussed and the approach has been improved for the past two decades (see Machado and Mata (2005), Melly (2005), Chernozhukov et al. (2013), Firpo et al. (2018)) but there has not been a perfect approach so far.

2 Background on Decomposition

With the advent of the pioneering papers by Oaxaca (1973) and Blinder (1973), labor economists have utilized decomposition methods to understand wage differences

across gender and race. Based on linearity, ignorability and overlapping support assumptions, Oaxaca-Blinder decomposition divides difference in mean wages into two parts by adding and subtracting a counterfactual wage, i.e. how much an individual belongs to one group would earn if he or she was in the other group. Following the logic of Fortin et al. (2011), I will show the process of traditional Oaxaca-Blinder decomposition.

Suppose Y_T is the wage for group $T \in \{0, 1\}$, the relationship between wage and observable characteristics X is linear:

$$Y_T = X\beta_T + u_T, \text{ where } E[u_T|X] = 0$$

Denote Δ_O^μ as the overall mean wage difference for two subgroups:

$$\begin{aligned}
\Delta_O^\mu &= E[Y_1|T=1] - E[Y_0|T=0] \\
&= E[E(Y_1|X, T=1)|T=1] - E[E(Y_0|X, T=0)|T=0] \\
&= E[(X\beta_1 + u_1)|T=1] - E[(X\beta_0 + u_0)|T=0] \\
&= (E[X|T=1]\beta_1 + \underbrace{E[u_1|T=1]}_{0}) - (E[X|T=0]\beta_0 - \underbrace{E[u_0|T=0]}_{0})
\end{aligned}$$

Now, let us add and subtract the counterfactual term $E[X|T=1]\beta_0$:

$$\begin{aligned}
\Delta_O^\mu &= (E[X|T=1]\beta_1 - E[X|T=1]\beta_0) + (E[X|T=1]\beta_0 - E[X|T=0]\beta_0) \\
&= \underbrace{E[X|T=1](\beta_1 - \beta_0)}_{\Delta_s^\mu} + \underbrace{(E[X|T=1] - E[X|T=0])\beta_0}_{\Delta_x^\mu}
\end{aligned}$$

Using sample means and OLS estimates $\hat{\beta_T}$, we have

$$\begin{aligned}
\Delta_O^\mu &= \bar{X}_1(\hat{\beta}_1 - \hat{\beta}_0) + (\bar{X}_1 - \bar{X}_0)\beta_0 \\
&= \Delta_s^\mu + \Delta_x^\mu
\end{aligned}$$

The wage structural effect Δ_s^μ represents the wage structure difference existing in different groups given the same characteristics. The composite effect Δ_u^μ captures the wage difference due to group characteristics disparity.

From the additive linearity assumption, we can divide the two components into the contributions of each covariate $k \in \{1, 2, ...M\}$:

$$\Delta_s^\mu = \underbrace{(\hat{\beta}_{1_0} - \hat{\beta}_{0_0})}_{\text{omitted group effect}} + \sum_{k=1}^{M} \bar{X}_{1_k}(\hat{\beta}_{1_k} - \hat{\beta}_{0_k})$$

$$\Delta_x^\mu = \sum_{k=1}^{M} \bar{X}_{1_k} - \bar{X}_{0_k}\hat{\beta}_{0_k}$$

Though the method is intuitive and implementable, it has several limitations. First, the choice of base group would affect the contribution of each covariate in the wage structual effect (Oaxaca and Ransom, 1999). Changing the based group would at least revert the sign of the contribution.

Second, the procedure is path dependent because we replace the distribution of each covariate for one group with the distribution of the other sequantially and the contribution of one covariate depends on the distribution of covariates added earlier (Fortin et al., 2011). This means that the contribution of each covariate would vary with the order of decomposition.

Most importantly, it is difficult to extend Oaxaca-Blinder (OB) decomposition to distributional statistics such as quantile regression. The underlying reason is that OB method assumes linear conditional expectations and get counterfacutal mean wage as $E[X|T = 1]\beta_0$ (Barsky et al., 2002). If the conditional expectation is non-linear, then we cannot approximate the counterfactural statistics. Also, OB method assumes additive linearity so that they can implement detailed decomposition; however, quantile statistics do not have the attribute.

3 Literature Review

To deal with the third limitation above and expand OB decomposition to general distributional statistics such as quantile, the researches have come up with mainly three approaches: reweighting, conditional quantile regression,and recentered influence function (RIF) regressions.

3.1 Reweighting

Under the assumption of ignorability, DiNardo et al. (1996) propose to apply the kernel density distribution to the IPW reweighted sample as a way for constructing the counterfactural distribution. They also provide a way of detailed decomposition for dummy explanatory variables. Barsky et al. (2002) follow the logic and use a non-parametric reweighting approach to estimate the counterfactual.

However, there has not been a general way to further decompose the two effects into the contribution of each covariate under this method.

3.2 Conditional Quantile Regression

When it comes to conditional quantile regression methods, Gosling et al. (2000), Machado and Mata (2005) and Melly (2005) estimate the conditional distribution using quantile regression. The spirits of the three important papers are similar but varies in the level of complexity.

Machado and Mata (2005) have proposed to estimate all possible conditional quantile regression to characterize the conditional distribution of the dependent

variable given observables. Then, they use these estimates and simulation methods to calculate the detailed components in the aggregate gap and the wage structure gap. This approach allows for varying returns to covariates that differ across the conditional wage distribution.

Gosling et al. (2000) provide an alternative way but it is more complicated to get the cumulative distribution function. Built upon the previous work, Melly (2005) has tackled crossing different quantile curves through asymptotic distribution of the estimator.

However, the approach requires the knowledge of the estimation of the marginal density function of wages, and it requires huge computational power because it tries to estimate on all possible quantiles. Plus, it does not solve the problem of path dependece. More importantly, if the topic of interest is unconditional quantile regression, the conditional distribution cannot reveal the variability of the covariates in the population (Fortin et al., 2011).

3.3 RIF Regression

Firpo et al. (2018) introduce recentered influence function (RIF) regression to decomposition.

It is an implementable and highly applicable method of detailed decompostion under unconditional quantile regression.

They first decompose the wage difference into wage structure and composite effects by reweighting method proposed by DiNardo et al. (1996), and then decompose the two effects into the contribution of covariates using conditional expectations RIF regression. Interestingly, the basic spirit is still Oxaca-Blinder decomposition, so the implementation is easier.

It is noteworthy that: first, we can use the law of iterated expectations to get detailed decomposition because the average of the conditional expectation of RIF is equal to the distributional statistic; second, the problem of path dependency can be solved because RIF itself comprises partial derivative. However, it only gives us the partial effect of a covariate due to the nature of RIF, and it cannot solve the problem of base group.

4 A Known Solution

In this part, I would elaborate the procedures of Firpo et al. (2018) in detail and implement them in R.

4.1 First Stage: Aggregate Decomposition

The two important steps here are to reweight sample using IPW and estimate the distributional statistics.

4.1.1 Reweighting Sample

Under the assumptions of ignorability and overlapping support, we have the inverse propensity weighting of $\hat{w}_1(T)$, $\hat{w}_0(T)$, and $\hat{w}_c(T,X)$.

Assumption of Ignorability Let $(T,\ X,\ \epsilon)$ have a joint distribution. $\forall x \in X,\ \epsilon$ must be independent of T given x.

Assumption of Overlapping Support $\forall x \in X,\ P[T=1|X=x] < 1$; furthermore, $P[T=1|X=x] > 0$.

IPW weighting where $\hat{p} = \frac{\sum_{i=1}^{N} T_i}{N}$,

$$
\begin{aligned}
w_1(T) \equiv \frac{T}{p} \quad &\rightarrow \quad \hat{w}_1(T) = \frac{T}{\hat{p}} \\
w_0(T) \equiv \frac{1-T}{1-p} \quad &\rightarrow \quad \hat{w}_0(T) = \frac{1-T}{1-\hat{p}} \\
w_c(T,X) \equiv \frac{1-T}{p} \cdot \left(\frac{p(X)}{1-p(X)}\right) \quad &\rightarrow \quad \hat{w}_c(T,X) = \frac{1-T}{\hat{p}} \cdot \left(\frac{\hat{p}(X)}{1-\hat{p}(X)}\right)
\end{aligned}
$$

To have weights summing up to one, we also need to normalize the above weights.

$$
\begin{aligned}
\hat{w}_1^*(T_i) &= \frac{\hat{w}_1(T_i)}{\sum_{j=1}^{N} \hat{w}_1(T_j)} \\
\hat{w}_0^*(T_i) &= \frac{\hat{w}_0(T_i)}{\sum_{j=1}^{N} \hat{w}_0(T_j)} \\
\hat{w}_c^*(T_i, X_i) &= \frac{\hat{w}_c(T_i, X_i)}{\sum_{j=1}^{N} \hat{w}_c(T_j, X_j)}
\end{aligned}
$$

Given the parametric p-score assumption, the authors suggest calculate the probability in probit or logit model and use MLE to tune the best parameters.

4.1.2 Estimating Distributional Statistics

From the observations, we can get the data on $(Y,\ T,\ X)$ and thus can non-parametrically identify F_1 (the distribution of Y_1 given $T=1$) and F_0 (the distribution of Y_0 given $T=0$). And we want to know the counterfactural distribution of F_c (the distribution of Y_0 given $T=1$). As we know, F_c can hardly be esimated for non-linear distribution statistics; therefore, we need to utilize reweighting to approach it.

We also define v as a functional of the conditional joint distribution $(Y_1, Y_0)|T$; v would represent the distributional statistics of interest. After calculating the weights, we can estimate the distributional statistic of interest ($\hat{v}_0$, $\hat{v}_1$, and $\hat{v}_c$) under certain regularity conditions.

$$\hat{v}_t = v\Big(\hat{F}_t(y)\Big) = v\Big(\sum_{i=1}^{N} \hat{w}_t^*(T_i) \cdot \mathbb{1}\{Y_i \le y\}\Big) \quad , \quad t \in \{0,1\}$$

$$\hat{v}_c = v\Big(\hat{F}_c(y)\Big) = v\Big(\sum_{i=1}^{N} \hat{w}_c^*(T_i, X_i) \cdot \mathbb{1}\{Y_i \le y\}\Big)$$

As long as we get the estimated statistics, we can divide the wage difference into the wage structure effect $\hat{\Delta}_s^v = \hat{v}_1 - \hat{v}_c$ and the composition effect $\hat{\Delta}_x^v = \hat{v}_c - \hat{v}_0$.

4.2 Second Stage: Detailed Decomposition

To further apportion the wage structure and composition effects to every covariate, RIF Regression comes into place. The basic idea is the same as Oaxaca-Blinder decomposition, but the vital difference is to replace the distributional statistics of interest with the conditional expectation of its RIF value. After this replacement, the law of iterated expectations can be applied to detailed decomposition.

The influence function is defined as (Hampel, 1974):

$$\text{IF}(y; v, F) = \lim_{\epsilon \to 0} \frac{v(F_\epsilon) - v(F)}{\epsilon} = \lim_{\epsilon \to 0} \frac{v\Big((1-\epsilon)F + \epsilon\delta_y\Big) - v(F)}{\epsilon}$$

where $0 \le \epsilon \le 1$ and δ_y is a distribution that puts mass only at the value y. By definition, $\int_{-\infty}^{\infty} \text{IF}(y; v, F)\, dF(y) = 0$.

$$\text{RIF}(y; v, F) = v(F) + \text{IF}(y; v, F)$$

$$E(\text{RIF}) = \int \text{RIF}(y; v, F)\, dF(y) = \int \Big(v(F) + \text{IF}(y; v, F)\Big) dF(y) = v(F)$$

Using the law of iterated expectations,

$$v(F) = E(\text{RIF}) = E[E(\text{RIF}|X = x)] = \int E[\text{RIF}(y; v, F)|X = x]\, dF_X(x)$$

In our case of quantiles, as specified by the authors:

$$Q(F, \tau) = \inf\{y | F(y) \ge \tau\} = q_\tau$$

$$\text{IF}(y; q_\tau, F) = \frac{\tau - \mathbb{1}\{y \le q_\tau\}}{f_Y(q_\tau)}$$

$$\text{RIF}(y; q_\tau, F) = q_\tau + \text{IF}(y; q_\tau, F) = q_\tau + \frac{\tau - \mathbb{1}\{y \le q_\tau\}}{f_Y(q_\tau)}$$

In the replication package, the authors use kernel function to estimate the density $f_Y(q_\tau)$, and the remaining variables are observed in the data. After we get the estimate $\hat{\text{RIF}}(y; q_\tau, F)$, we simply implement linear regression on it.

Now, the value of the counterfactural wage is no longer equal to $E[X|T=1]\beta_0$ but based on the coefficient of the reweighted counterfactual, the form of wage structure and composite effects has some changes in the coefficient. In the case of quantiles, v is simply q_τ.

$$\begin{aligned}
\Delta_s^{q_\tau} &= E[X|T=1] \cdot \left(\hat{\gamma}_1^{q_\tau} - \hat{\gamma}_c^{q_\tau}\right) \\
&= \sum_{k=1}^{M} E[X^k|T=1] \cdot \left(\hat{\gamma}_{1,k}^{q_\tau} - \hat{\gamma}_{c,k}^{q_\tau}\right) \\
\Delta_x^{q_\tau} &= \left(E[X|T=1] - E[X|T=0]\right) \cdot \hat{\gamma}_0^{q_\tau} + \hat{R}^{q_\tau} \\
&= \sum_{k=1}^{M} \left(E[X^k|T=1] - E[X^k|T=0]\right) \cdot \hat{\gamma}_{0,k}^{q_\tau}
\end{aligned}$$

where γs are the coefficients of reweighted RIF regressions, and $\hat{R}^{q_\tau} = E[X|T=1] \cdot \left(\hat{\gamma}_c^{q_\tau} - \hat{\gamma}_0^{q_\tau}\right)$ is the approximation error.

4.3 Implementation in R

I use the data from the replication package of (Firpo et al., 2018) (https://sites.google.com/view/nicole-m-fortin/data-and-programs). I choose this dataset to ensure that important factors of wage inequality is covered as the paper and to show the implementation of the known solution.

There are 2 evidence that can prove my originality of code. First, The authors compare the wage gap of male in 1990s and in 2010s. However, I compare the wage gap of male in 2010s for white and nonwhite. Second, The authors write the codes in Stata and use packages for logit, quantile, kernel density, regression and wage decomposition. However, I write the whole stuff in R and write implicit codes rather than use packages.

In Appendix A, I attach the codes for the whole process following the logic mentioned above. What is worth mentioning is that it is CPS data which has sample weight for each observations. Therefore, besides the reweighting factors for each group, I also take into consideration the sample weights. Please kindly check my hardwork.

5 A New Solution

After examining the steps taken by the authors, I raise two changes in the procedures of reweighting.

First, I only keep the observations which satisfy the assumption of overlapping support. In the replication package, they do not drop the observations with the set of characteristics only appear in one group but not the other. However, this assumption is vital for identification and the criteria should be added. After dropping these variables, the number of observations decrease from 235,336 to 206,549.

Second, to save computational power, I calculate $p(X)$ in reweighting procedures using a different way. The authors use logit model to estimate the probability of being in the $T = 1$ group given covariates. Instead, I define each group as a unique combination of certain covariates. Given one group, I calculate the probability via dividing the observations in $T = 1$ by the total number of observations in that group (considering sample weight). It is also a common practice in the field of labor economics.

Interestingly, $p(X) \in (0.003, 0.997)$ is wider under my new solution than that of under logit model prediction. It manifests another advantage of using my new solution and the group indicator captures more intricacies than the logit model. Logit model might overvalue or undervalue many interactions between covariates, which allows for manipulation if people choose the best combination of interactions.

In Appendix B, I attach the codes for the whole process.

6 Results

I report the results of aggregate decomposition in Table 1 and Table 2, and the results of detailed decomposition for each covariate in Table 3 and Table 4 in Appendix C.

Table 1: Aggregate Wage Decomposition after IPW Weighting and RIF

Wage Structure Effect			Composition Effect		
10_th	50_th	90_th	10_th	50_th	90_th
0.0809	0.196	0.174	-0.0311	-0.0963	-0.0423

Table 2: Aggregate Wage Decomposition after IPW Weighting and RIF (New)

Wage Structure Effect			Composition Effect		
10_th	50_th	90_th	10_th	50_th	90_th
0.0658	0.1626	0.068	-0.0005	-0.0462	0.0342

In terms of wage structure effect, the two solutions give similar results on the 10th and 50th quantile. But the result for the 90th quantile are quite dissimilar in magnitude. This can be explained by the effect of dropping non-overlapping

observations. Intuitively, white male might have a better job, a higher education and work in higher-end industry that nonwhite male cannot be equipped with.

Overall, the story is that for people with higher wages in the nonwhite group, they are more affected by the wage structure effect. To be specific, the additional earnings for a nonwhite with above nonwhite median wage will be higher that that for a nonwhite with nonwhite bottom 10 percent wage if being in white group.

Considering composition effect, the results from the two solutions shows wider discrepancy. It should also be due to dropping non-overlapping support. Composition effects are composed of the difference between expected value of covariates given the group status and dropping the variables must change the corresponding expected value.

7 Conclusion

In this project, I follow the logic of Fortin et al. (2011) and conduct both aggregate and detailed decomposition for white and nonwhite male in terms of the 10th, 50th, and 90th quantile. The method is great at its path independence, detailed decomposition for quantile, and strong explanation power for unconditional quantile.

Extended from their original approach, I revised their reweighting procedures by dropping observations without an overlapping support and using an alternative way to calculate $p(T = 1|X = x)$. The results show that the criteria of overlapping support and weighting method do matter in the quantile setting.

References

Barsky, R., Bound, J., Charles, K. K., and Lupton, J. P. (2002). Accounting for the black-white wealth gap: A nonparametric approach. *Journal of the American Statistical Association*, 97(459):663–673.

Blau, F. D. and Kahn, L. M. (2017). The gender wage gap: Extent, trends, and explanations. *Journal of Economic Literature*, 55(3):789–865.

Blinder, A. S. (1973). Wage discrimination: Reduced form and structural estimates. *The Journal of Human Resources*, 8(4):436–455.

Chernozhukov, V., Fernández-Val, I., and Melly, B. (2013). Inference on counterfactual distributions. *Econometrica*, 81(6):2205–2268.

DiNardo, J., Fortin, N. M., and Lemieux, T. (1996). Labor market institutions and the distribution of wages, 1973-1992: A semiparametric approach. *Econometrica*, 64(5):1001–1044.

Firpo, S. P., Fortin, N. M., and Lemieux, T. (2018). Decomposing wage distributions using recentered influence function regressions. *Econometrics*, 6(2).

Fortin, N., Lemieux, T., and Firpo, S. (2011). Chapter 1 - decomposition methods in economics. volume 4 of *Handbook of Labor Economics*, pages 1–102. Elsevier.

Gosling, A., Machin, S., and Meghir, C. (2000). The Changing Distribution of Male Wages in the U.K. *The Review of Economic Studies*, 67(4):635–666.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393.

Machado, J. A. F. and Mata, J. (2005). Counterfactual decomposition of changes in wage distributions using quantile regression. *Journal of Applied Econometrics*, 20(4):445–465.

Melly, B. (2005). Decomposition of differences in distribution using quantile regression. *Labour Economics*, 12(4):577–590. European Association of Labour Economists 16th Annual Conference, Universidade Nova de Lisboa, Lisbon, 9th – 11th Sepetember, 2004.

Oaxaca, R. (1973). Male-female wage differentials in urban labor markets. *International Economic Review*, 14(3):693–709.

Oaxaca, R. L. and Ransom, M. R. (1999). Identification in detailed wage decompositions. *The Review of Economics and Statistics*, 81(1):154–157.

Appendix A

Known_Solution

Jieyi Chen

5/28/2022

Contents

Basic Setup

```
rm(list = ls())
options(kableExtra.latex.load_packages = FALSE)
library(stats)
library(tidyverse)
library(haven) # import dta file
library(fastDummies) # dummify certain columns
library(kableExtra)
options(scipen = 999) # no scienctific counting method
setwd("/Users/chenjieyi/Course/22 Spring/modeling/Final")
```

Data Manipulation

I use the data from Firpo et al. (2018) https://sites.google.com/view/nicole-m-fortin/data-and-programs. I choose this dataset to ensure that important factors of wage inequality is covered as the paper and to show the implementation of the known solution. There are 2 evidence that can prove my orginality of codework.
1. The authors compare the wage gap of male in 1990s and in 2010s. However, I compare the wage gap of male in 2010s for white and nonwhite.
2. The authors write the codes in Stata. But I write the whole stuff in R and write implicit codes rather than use known package.

First, I follow what the authors do in Stata to clean the dataset.

```
data <- read_dta("morgm_all8816.dta") %>%
  # keep 2014-2016 data
  filter(between(year, 114, 116)) %>%
  # keep non-missing wage and working hours
  filter(!is.na(lwage1) & !is.na(uhrswk)) %>%
  # generate white var
  mutate(white = 1 - nonwhite) %>%
  # generate marital status var
  mutate(nmarr = 1 - marr) %>%
  # generate public sector var
  mutate(pub = ifelse(between(class, 1, 3), 1, 0)) %>%
  # generate education level vars
  mutate(ed = case_when(educ < 9 ~ 0,
                        between(educ, 9, 11) ~ 1,
                        educ == 12 ~ 2,
                        between(educ, 13, 15) ~ 3,
                        educ == 16 ~ 4,
                        educ > 16 ~ 5)) %>%
  # generate experience level vars
  mutate(ex =  case_when(exper < 5 ~ 1,
                         between(exper, 5, 9) ~ 2,
                         between(exper, 10, 14) ~ 3,
                         between(exper, 15, 19) ~ 4,
                         between(exper, 20, 24) ~ 5,
                         between(exper, 25, 29) ~ 6,
                         between(exper, 30, 34) ~ 7,
                         between(exper, 35, 39) ~ 8,
                         exper >= 40 ~ 9)) %>%
  # generate occupation vars
  mutate(occd = case_when((between(occ3, 10, 199) | occ3 == 430) ~ 11,
                          (between(occ3, 200, 999) & occ3 != 430) ~ 12,
                          between(occ3, 1000, 1560) ~ 21,
                          between(occ3, 1600, 1999) ~ 22,
                          (between(occ3, 2000, 2099) |
                             between(occ3, 2140, 2999)) ~ 23,
                          (between(occ3, 2100, 2110) | occ3 == 3010 |
                             occ3 == 3060) ~ 24,
                          (occ3 == 3000 | between(occ3, 3030, 3050) |
                             between(occ3, 3110, 3540)) ~ 25,
                          between(occ3, 5000, 5930) ~ 30,
                          (between(occ3, 4700, 4960) & occ3 != 4810 &
                             occ3 != 4820 & occ3 != 4920) ~ 40,
```

```
                          occ3 %in% c(4810, 4920) ~ 41,
                          occ3 == 4820 ~ 42,
                          between(occ3, 3600, 4699) ~ 50,
                          between(occ3, 6000, 6130) ~ 60,
                          between(occ3, 6200, 7630) ~ 70,
                          between(occ3, 7700, 8965)~ 80,
                          (between(occ3, 9000, 9750) &
                             occ3 != 9130)~ 90,
                          occ3 == 9130 ~ 91)) %>%
  # generate industry vars
  mutate(indd = case_when(between(ind3, 170, 490) ~ 1,
                          ind3 == 770 ~ 2,
                          (between(ind3, 3360, 3690) |
                             between(ind3, 2170, 2390) | ind3 == 3960 |
                             ind3 == 3180) ~ 3,
                          ((between(ind3, 2470, 3170) |
                              between(ind3, 3190, 3290) |
                              between(ind3, 3770, 3990) |
                              between(ind3, 1070, 2090)) & ind3 != 3960) ~ 4,
                          between(ind3, 4070, 4590) ~ 5,
                          between(ind3, 4670, 5790) ~ 6,
                          (between(ind3, 6070, 6390) |
                             between(ind3, 570, 690)) ~ 7,
                          (between(ind3, 6470, 6480) |
                             between(ind3, 6570, 6670) |
                             between(ind3, 6770, 6780)) ~ 8,
                          between(ind3, 6870, 7190) ~ 9,
                          (between(ind3, 7290, 7460) | ind3 == 6490 |
                             between(ind3, 6675, 6695)) ~ 10,
                          (between(ind3, 7270, 7280) |
                             between(ind3, 7470, 7790)) ~ 11,
                          between(ind3, 7860, 8470) ~ 12,
                          between(ind3, 8560, 9290) ~ 13,
                          between(ind3, 9370, 9590) ~ 14)) %>%
  # generate base group
  mutate(base = ifelse(covered==0 & marr==1 & ed==2 & ex==5
                       & occd==70 & indd==2, 1, 0)) %>%
  select(-nonwhite) %>%
  select(white, everything())
  # create group_id
  # %>%
  # group_by(covered, nmarr, pub, ed, ex, occd, indd) %>%
  # mutate(group = cur_group_id())

# drop NA values
data <- data %>%
  filter(!is.na(occd)) %>%
  filter(!is.na(indd))

# dummify selected columns
dummy <- c("ed","ex","occd","indd")
data <- dummy_cols(data, select_columns = dummy)
```

```
# expand data by eweight/1000; need to change later...
# data <- data[rep(row.names(data), data$eweight/1000), ]
```

Second, I store three datasets of nonwhite (t=0), white (t=1),and counterfactual(t=2)

```
data_0 <- data %>%
  filter(white == 0)

data_1 <- data %>%
  filter(white == 1)

data_2 <- data
```

The First Stage

Reweighting Sample

The key point here is that CPS has its own sample weight *eweight* and I pay special attention to always inovolve it when considering weighting.

Reweighting for t=0,1

First, I reweight for nonwhite and white (considering original sample weight)

```
# calculate p considering eweight
n_white <- t(data$white) %*% data$eweight
n_all <- as.vector(sum(data$eweight))
p <- n_white/n_all

# reweight for nonwhite (t=0)
data_0 <- data_0 %>%
  mutate( weight = ( (1/(1-p)) * eweight / n_all ) ) %>%
  mutate(t = 0)

sum(data_0$weight)
```

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

```
# reweight for white (t=1)
data_1 <- data_1 %>%
  mutate( weight = ((1/p) * eweight / n_all ) ) %>%
  mutate(t = 1)

sum(data_1$weight)
```

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

Reweighting for t=c (t=2)

Second, I reweight for counterfactural group (considering original sample weight)
1. Calculate p(X) using logit from scratch
2. Use formula to reweight

```
############## Function for Logit and MLE ##############

# L(z) = (1+exp(-z))^(-1)
L <- function(z){1/(1+exp(-z))}

# Transfer argmax function to argmin function (cost)
cost <- function(delta, X, T_i, eweight){
  p <- L(X %*% delta)
  c <- (t(-(T_i*eweight))%*%log(p)-t((1-T_i)*eweight)%*%log(1-p)) / sum(eweight)
  return(c)
}

# Optimization with gradient descent
gradient <- function(delta, X, T_i, eweight){
  p <- L(X%*%delta)
  g <- (t(X)%*%((p - T_i)*eweight)) / sum(eweight)
  return(g)
}

# Get the optimal parameters
logisticReg <- function(X, T_i){
  # add intercept term for X
  X <- mutate(X, bias =1)
  # move the intercept column to the first column
  X <- as.matrix(X[, c(ncol(X), 1:(ncol(X)-1))])
  T_i <- as.matrix(T_i)
  # put the values in delta
  delta <- matrix(rep(0, ncol(X)), nrow = ncol(X))
  # find the optimal parameters using gradient defined before
  para <- optim(delta, fn = cost, gr = gradient,
                X = X, T_i = T_i, eweight = eweight)
  return(para$par)
}

# Get p(X)!
logisticProb <- function(delta, X){
  X <- mutate(X, bias =1)
  X <- as.matrix(X[,c(ncol(X), 1:(ncol(X)-1))])
  return(L(X%*%delta))
}

# modified from: https://towardsdatascience.com/logistic-regression-from-scratch-in-r-b5b122fd8e83
# modified from: https://www.baeldung.com/cs/gradient-descent-logistic-regression
############## Get p(X) ##############

# input
T_i <- data_2$white
X <- data_2[, c(22:23, 30:75)] # union, nmarr, ed, ex, indd, occd
```

```
eweight <- data_2$eweight

# output
parameter <- logisticReg(X, T_i)
pX <- logisticProb(parameter, X)
data_2 <- cbind(data_2, pX)

# Reweight for counterfactural group: white=0,
# but we want to approach their wages under white=1
data_2 <- data_2 %>%
  filter(white==0) %>%
  mutate(w = (1/p) * (pX/(1-pX)) * eweight) %>%
  mutate(weight = w/sum(w))
```

```
## Warning in (1/p) * (pX/(1 - pX)): Recycling array of length 1 in array-vector arithmetic is deprecat
##   Use c() or as.vector() instead.
```

```
data_2 <- data_2 %>%
  select(-pX, -w) %>%
  mutate(t = 2)

sum(data_2$weight)
```

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

Estimating Distributional Statistics

Empirical cdf from scratch

Here, the logic to recover the cdf is to sort the observation by lwage1 and then sequentially add their weights. And I believe it works in this case thanks to our big sample size.

```
############### CDF for t=0 ###############

cdf <- c()
cdf_0 <- c(rep(0, nrow(data_0)+1))
data_0 <- data_0 %>%
  arrange(lwage1)
for (i in 2:(nrow(data_0)+1)) {
  cdf_0[i] <- data_0$weight[i-1] + cdf_0[i-1]
}
cdf <- cdf_0[-1]
data_0 <- cbind(data_0, cdf)

############### CDF for t=1 ###############

cdf_1 <- c(rep(0, nrow(data_1)+1))
data_1 <- data_1 %>%
  arrange(lwage1)
for (i in 2:(nrow(data_1)+1)) {
  cdf_1[i] <- data_1$weight[i-1] + cdf_1[i-1]
```

```
}
cdf <- cdf_1[-1]
data_1 <- cbind(data_1, cdf)

############## CDF for t=2 ##############
cdf_2 <- c(rep(0, nrow(data_2)+1))
data_2 <- data_2 %>%
  arrange(lwage1)
for (i in 2:(nrow(data_2)+1)) {
  cdf_2[i] <- data_2$weight[i-1] + cdf_2[i-1]
}
cdf <- cdf_2[-1]
data_2 <- cbind(data_2, cdf)
```

Wage for a quantile

Write a function to find the wage for a certain quantile after weighting using cdf. My way of finding the wage is to find the cdf that is closest to the quantile we set in absolute value.

```
find_quantile <- function(tau, df){
  # find the number of row which gives the cdf closest to tau
  cdf <- df$cdf
  n_tau <- which.min(abs(tau-cdf))
  # get lwage1 of that row
  q_tau <- df$lwage1[n_tau]
  return(q_tau)
}
```

Gassian kernel density

Get empirical density ($f_y(q_\tau)$) from scratch using Gaussian kernel method.

$$K(x) = \frac{1}{\sqrt{2\pi}} \times e^{-\frac{1}{2}\left(\frac{X_i - x}{h}\right)^2} \tilde{f}(x) = \frac{1}{nh} \times \sum_{i=1}^{N} K\left(\frac{X_i - x}{h}\right) = \frac{1}{h\sqrt{2\pi}} \times \frac{1}{n} \sum_{i=1}^{N} e^{-\frac{1}{2}\left(\frac{X_i - x}{h}\right)^2}$$

We have to take weight into account, so the below codes integrate some weight issues.

```
# I choose h = 0.068 as the best bandwidth from runninig the density function
# d <- density(data_0$lwage1, weights = data_0$weight)
# plot(d, lwd = 2, main = "Default kernel density plot")

find_density <- function(tau, df, h=0.068){
  # (1) find x = q_tau
  q_tau <- find_quantile(tau, df)
  # (2) calculate the first term (weight)
  first_term <- 1 / (h*sqrt(2*pi))
  # (3) calculate the second term (weight)
  second_term <- c()
  for (i in 1:nrow(df)) {
    second_term[i] <- exp(-0.5*(((df$lwage1[i] - q_tau)/h)^2)) * df$weight[i]
  }
```

```
  second_term_sum <- sum(second_term)
  return(first_term*second_term_sum)
}

# modified from: https://medium.com/analytics-vidhya/kernel-density-estimation-kernel-construction-and-
```

The Second Stage: RIF Regression

$$\text{RIF}(y; q_\tau, F) = q_\tau + \frac{\tau - 1\{y \leq q_\tau\}}{f_y(q_\tau)}$$

RIF for each quantile

Write a function to get the RIF value for each τ

```
# create a function whether: if t==TRUE, then 1; if t==FALSE, then 0
whether <- function(t) ifelse(t, 1, 0)

find_RIF <- function(tau, df) {
  RIF <- c()
  q_tau <- find_quantile(tau, df)
  fq_tau <- find_density(tau, df)
  for (i in 1:nrow(df)) {
  IF <- (tau - whether(df$lwage1[i] <= q_tau)) / fq_tau
  RIF[i] <- q_tau + IF
  }
  return(RIF)
}
```

RIF regression for each quantile

Write a function to get reweighted RIF Regression result

```
regress_RIF <- function(tau, df) {
  # input
  df <- df %>%
    select(-c("ed_0", "ex_1", "occd_11", "indd_1"))

  X <- df[, c(22:23, 30:71)]  # collenearity concern
  X <- as.matrix(X)
  intercept <- rep(1, nrow(X))
  X <- cbind(intercept, X)

  Y <- find_RIF(tau, df)
  Y <- as.matrix(Y)
  weight <- as.vector(df$weight)

  # closed-form solutions for coefficients
  coef <- solve(t(X*weight) %*% X) %*% t(X*weight) %*% Y
```

```
  # drop the coefficient for intercept
  coef <- coef[-1]

  return(coef)
}
```

Decompose the wage difference

```
# find the expected values
E_x_1 <- c()
E_x_0 <- c()
ind.x <- c(22:23, 30:71)
for (i in 1:44) {
  E_x_1[i] <- sum(data_1[, ind.x[i]] * data_1$weight)
  E_x_0[i] <- sum(data_0[, ind.x[i]] * data_0$weight)
}

# get the names for covariates
cov_name <- names(data_0)[c(22:23, 30:71)]

############## tau = 10 ##############
g_0_10 <- regress_RIF(tau = 0.1, df = data_0)
g_1_10 <- regress_RIF(tau = 0.1, df = data_1)
g_2_10 <- regress_RIF(tau = 0.1, df = data_2)
s_10 <- t(E_x_1) %*% (g_1_10 - g_2_10)
x_10 <- t(E_x_1 - E_x_0) %*% g_0_10 + t(E_x_1) %*% (g_2_10 - g_0_10)

cat("wage structure effect for 10th quantile is", s_10)
```

```
## wage structure effect for 10th quantile is 0.08085773
```

```
cat("composite effect for 10th quantile is", x_10)
```

```
## composite effect for 10th quantile is -0.0311192
```

```
cat("report approx error,", t(E_x_1) %*% (g_2_10 - g_0_10))
```

```
## report approx error, -0.006870239
```

```
s_10_d <- E_x_1 * (g_1_10 - g_2_10)
x_10_d <- (E_x_1 - E_x_0) * g_0_10

############## tau = 50 ##############

g_0_50 <- regress_RIF(tau = 0.5, df = data_0)
g_1_50 <- regress_RIF(tau = 0.5, df = data_1)
g_2_50 <- regress_RIF(tau = 0.5, df = data_2)
s_50 <- t(E_x_1) %*% (g_1_50 - g_2_50)
```

```r
x_50 <- t(E_x_1 - E_x_0) %*% g_0_50 + t(E_x_1) %*% (g_2_50 - g_0_50)

cat("wage structure effect for 50th quantile is", s_50)
```

```
## wage structure effect for 50th quantile is 0.1960117
```

```r
cat("composite effect for 50th quantile is", x_50)
```

```
## composite effect for 50th quantile is -0.0962583
```

```r
cat("report approx error,", t(E_x_1) %*% (g_2_50 - g_0_50))
```

```
## report approx error, -0.04810887
```

```r
s_50_d <- E_x_1 * (g_1_50 - g_2_50)
x_50_d <- (E_x_1 - E_x_0) * g_0_50

############## tau = 90 ###############
g_0_90 <- regress_RIF(tau = 0.9, df = data_0)
g_1_90 <- regress_RIF(tau = 0.9, df = data_1)
g_2_90 <- regress_RIF(tau = 0.9, df = data_2)
s_90 <- t(E_x_1) %*% (g_1_90 - g_2_90)
x_90 <- t(E_x_1 - E_x_0) %*% g_0_90 + t(E_x_1) %*% (g_2_90 - g_0_90)

cat("wage structure effect for 90th quantile is", s_90)
```

```
## wage structure effect for 90th quantile is 0.1739614
```

```r
cat("composite effect for 90th quantile is", x_90)
```

```
## composite effect for 90th quantile is -0.04232585
```

```r
cat("report approx error,", t(E_x_1) %*% (g_2_90 - g_0_90))
```

```
## report approx error, 0.03458706
```

```r
s_90_d <- E_x_1 * (g_1_90 - g_2_90)
x_90_d <- (E_x_1 - E_x_0) * g_0_90

d_all <- cbind(s_10_d, s_50_d, s_90_d, x_10_d, x_50_d, x_90_d)

rownames(d_all) <- cov_name

dd <- kbl(d_all, longtable = T, booktabs = T, format = "latex", digits = 4,
    col.names = c("10_th", "50_th", "90_th", "10_th", "50_th", "90_th"),
    caption = "Detailed Wage Decomposition after IPW Weighting and RIF") %>%
add_header_above(c(" ", "Wage Structual Effect" = 3, "Composite Effect" = 3)) %>%
kable_styling(latex_options = c("repeat_header"))
```

```
write.table(dd[[1]],"newdf.txt",sep="\t",row.names=FALSE)

all <- cbind(s_10, s_50, s_90, x_10, x_50, x_90)

aa <- kbl(all, longtable = T, booktabs = T, format = "latex", digits = 4,
    col.names = c("10_th", "50_th", "90_th", "10_th", "50_th", "90_th"),
    caption = "Aggregate Wage Decomposition after IPW Weighting and RIF") %>%
add_header_above(c("Wage Structual Effect" = 3, "Composite Effect" = 3)) %>%
kable_styling(latex_options = c("repeat_header"))

write.table(aa[[1]],"newdf.txt",sep="\t",row.names=FALSE)
```

Appendix B

New_Solution

Jieyi Chen

5/28/2022

Contents

Basic Setup

```
rm(list = ls())
library(stats)
library(tidyverse)
library(haven) # import dta file
library(fastDummies) # dummify certain columns
library(kableExtra)
options(scipen = 999) # no scienctific counting method
setwd("/Users/chenjieyi/Course/22 Spring/modeling/Final")
```

Data Manipulation

I use the data from Firpo et al. (2018) https://sites.google.com/view/nicole-m-fortin/data-and-programs. I choose this dataset to ensure that important factors of wage inequality is covered as the paper and to show the implementation of the known solution. There are 2 evidence that can prove my orginality of codework.
1. The authors compare the wage gap of male in 1990s and in 2010s. However, I compare the wage gap of male in 2010s for white and nonwhite.
2. The authors write the codes in Stata. But I write the whole stuff in R and write implicit codes rather than use known package.

First, I follow what the authors do in Stata to clean the dataset.

```
data <- read_dta("morgm_all8816.dta") %>%
  # keep 2014-2016 data
  filter(between(year, 114, 116)) %>%
  # keep non-missing wage and working hours
  filter(!is.na(lwage1) & !is.na(uhrswk)) %>%
  # generate white var
  mutate(white = 1 - nonwhite) %>%
  # generate marital status var
  mutate(nmarr = 1 - marr) %>%
  # generate public sector var
  mutate(pub = ifelse(between(class, 1, 3), 1, 0)) %>%
  # generate education level vars
  mutate(ed = case_when(educ < 9 ~ 0,
                        between(educ, 9, 11) ~ 1,
                        educ == 12 ~ 2,
                        between(educ, 13, 15) ~ 3,
                        educ == 16 ~ 4,
                        educ > 16 ~ 5)) %>%
  # generate experience level vars
  mutate(ex =  case_when(exper < 5 ~ 1,
                         between(exper, 5, 9) ~ 2,
                         between(exper, 10, 14) ~ 3,
                         between(exper, 15, 19) ~ 4,
                         between(exper, 20, 24) ~ 5,
                         between(exper, 25, 29) ~ 6,
                         between(exper, 30, 34) ~ 7,
                         between(exper, 35, 39) ~ 8,
                         exper >= 40 ~ 9)) %>%
  # generate occupation vars
  mutate(occd = case_when((between(occ3, 10, 199) | occ3 == 430) ~ 11,
                          (between(occ3, 200, 999) & occ3 != 430) ~ 12,
                          between(occ3, 1000, 1560) ~ 21,
                          between(occ3, 1600, 1999) ~ 22,
                          (between(occ3, 2000, 2099) |
                             between(occ3, 2140, 2999)) ~ 23,
                          (between(occ3, 2100, 2110) |
                             occ3 == 3010 | occ3 == 3060) ~ 24,
                          (occ3 == 3000 | between(occ3, 3030, 3050) |
                             between(occ3, 3110, 3540)) ~ 25,
                          between(occ3, 5000, 5930) ~ 30,
                          (between(occ3, 4700, 4960) & occ3 !=
                             4810 & occ3 != 4820 & occ3 != 4920) ~ 40,
```

```
                          occ3 %in% c(4810, 4920) ~ 41,
                          occ3 == 4820 ~ 42,
                          between(occ3, 3600, 4699) ~ 50,
                          between(occ3, 6000, 6130) ~ 60,
                          between(occ3, 6200, 7630) ~ 70,
                          between(occ3, 7700, 8965)~ 80,
                          (between(occ3, 9000, 9750) & occ3 != 9130)~ 90,
                          occ3 == 9130 ~ 91)) %>%
  # generate industry vars
  mutate(indd = case_when(between(ind3, 170, 490) ~ 1,
                          ind3 == 770 ~ 2,
                          (between(ind3, 3360, 3690) |
                             between(ind3, 2170, 2390) |
                             ind3 == 3960 | ind3 == 3180) ~ 3,
                          ((between(ind3, 2470, 3170) |
                              between(ind3, 3190, 3290) |
                              between(ind3, 3770, 3990) |
                              between(ind3, 1070, 2090)) & ind3 != 3960) ~ 4,
                          between(ind3, 4070, 4590) ~ 5,
                          between(ind3, 4670, 5790) ~ 6,
                          (between(ind3, 6070, 6390) |
                             between(ind3, 570, 690)) ~ 7,
                          (between(ind3, 6470, 6480) |
                             between(ind3, 6570, 6670) |
                             between(ind3, 6770, 6780)) ~ 8,
                          between(ind3, 6870, 7190) ~ 9,
                          (between(ind3, 7290, 7460) | ind3 == 6490 |
                             between(ind3, 6675, 6695)) ~ 10,
                          (between(ind3, 7270, 7280) |
                             between(ind3, 7470, 7790)) ~ 11,
                          between(ind3, 7860, 8470) ~ 12,
                          between(ind3, 8560, 9290) ~ 13,
                          between(ind3, 9370, 9590) ~ 14)) %>%
  # generate base group
  mutate(base = ifelse(covered==0 & marr==1 & ed==2 &
                         ex==5 & occd==70 & indd==2, 1, 0)) %>%
  select(-nonwhite) %>%
  select(white, everything())

# drop NA values
data <- data %>%
  filter(!is.na(occd)) %>%
  filter(!is.na(indd))

# dummify selected columns
dummy <- c("ed","ex","occd","indd")
data <- dummy_cols(data, select_columns = dummy)
```

New Solution Part 1

Create weighting groups and drop non-overlapping obs. The number of observations change from 235,336 to 206,549.

```
data <- data %>%
  # create weighting group id
  group_by(covered, nmarr, ed, ex, occd, indd) %>%
  mutate(group = cur_group_id()) %>%
  # drop obs without weighting group
  filter(!is.na(group))

# find the weighting unique group in white=0
group_0 <- data %>%
  filter(white == 0) %>%
  pull(group) %>%
  unique()

# find the unique weighting group in white=1
group_1 <- data %>%
  filter(white == 1) %>%
  pull(group) %>%
  unique()

# drop weighting group that only appear in one treatment group
group_all <- c(group_0,group_1) %>%
  as.data.frame()

colnames(group_all)[1] <- "group"

group_all <- group_all %>%
  count(group) %>%
  filter(n > 1) %>%
  pull(group)

data <- data %>%
  filter(group %in% group_all)
```

Second, I store three datasets of nonwhite (t=0), white (t=1),and counterfactual (t=2)

```
data_0 <- data %>%
  filter(white == 0)

data_1 <- data %>%
  filter(white == 1)

data_2 <- data
```

The First Stage

Reweighting Sample

The key point here is that CPS has its own sample weight *eweight* and I pay special attention to always inovolve it when considering weighting.

Reweighting for t=0,1

First, I reweight for nonwhite and white (considering original sample weight)

```
# calculate p considering eweight
n_white <- t(data$white) %*% data$eweight
n_all <- as.vector(sum(data$eweight))
p <- as.vector(n_white/n_all)

# reweight for nonwhite (t=0)
data_0 <- data_0 %>%
  mutate( weight = ( (1/(1-p)) * eweight / n_all ) ) %>%
  mutate(t = 0)

sum(data_0$weight)
```

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

```
# reweight for white (t=1)
data_1 <- data_1 %>%
  mutate( weight = ((1/p) * eweight / n_all ) ) %>%
  mutate(t = 1)

sum(data_1$weight)
```

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

Reweighting for t=c (t=2)

Second, I reweight for counterfactural group (considering original sample weight)

New Solution Part 2 Change from logit to:
p(X) = dividing the number of times being in $T = 1$ group by the number of observations in that group (considering sample weight)

```
############### Get p(X) ##############
pX <- c()
for (i in 1:length(group_all)) {
  t <- data_2[which(data_2$group == group_all[i]), ]
  pX[i] <- sum(t$white*t$eweight) / sum(t$eweight)
}

group_p <- cbind(group_all, pX) %>% as.data.frame()

data_2 <- data_2 %>%
  left_join(group_p, by = c("group" = "group_all"))


############### Reweight t=2 ##############

# Reweight for counterfactural group: white=0,
```

```r
# but we want to approach their wages under white=1

data_2 <- data_2 %>%
  filter(white==0) %>%
  mutate(w = (1/p) * (pX/(1-pX)) * eweight )

weight_all <- sum(data_2$w)
data_2 <- data_2 %>%
  mutate(weight = w/weight_all)

data_2 <- data_2 %>%
  select(-pX, -w) %>%
  mutate(t = 2)

sum(data_2$weight)
```

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

Estimating Distributional Statistics

Empirical cdf from scratch

Here, the logic to recover the cdf is to sort the observation by lwage1 and then sequentially add their weights. And I believe it works in this case thanks to our big sample size.

```r
############## CDF for t=0 ##############

cdf <- c()
cdf_0 <- c(rep(0, nrow(data_0)+1))
data_0 <- data_0 %>%
  arrange(lwage1)
for (i in 2:(nrow(data_0)+1)) {
  cdf_0[i] <- data_0$weight[i-1] + cdf_0[i-1]
}
cdf <- cdf_0[-1] %>% as.data.frame()
colnames(cdf) <- "cdf"
data_0 <- cbind(data_0, cdf)

############## CDF for t=1 ##############

cdf_1 <- c(rep(0, nrow(data_1)+1))
data_1 <- data_1 %>%
  arrange(lwage1)
for (i in 2:(nrow(data_1)+1)) {
  cdf_1[i] <- data_1$weight[i-1] + cdf_1[i-1]
}
cdf <- cdf_1[-1] %>% as.data.frame()
colnames(cdf) <- "cdf"
data_1 <- cbind(data_1, cdf)

############## CDF for t=2 ##############
cdf_2 <- c(rep(0, nrow(data_2)+1))
```

```r
data_2 <- data_2 %>%
  arrange(lwage1)
for (i in 2:(nrow(data_2)+1)) {
  cdf_2[i] <- data_2$weight[i-1] + cdf_2[i-1]
}
cdf <- cdf_2[-1] %>% as.data.frame()
colnames(cdf) <- "cdf"
data_2 <- cbind(data_2, cdf)
```

Wage for a quantile

Write a function to find the wage for a certain quantile after weighting using cdf. My way of finding the wage is to find the cdf that is closest to the quantile we set in absolute value.

```r
find_quantile <- function(tau, df){
  # find the number of row which gives the cdf closest to tau
  cdf <- df$cdf
  n_tau <- which.min(abs(tau-cdf))
  # get lwage1 of that row
  q_tau <- df$lwage1[n_tau]
  return(q_tau)
}
```

Gassian kernel density

Get empirical density ($f_y(q_\tau)$) from scratch using Gaussian kernel method.

$$K(x) = \frac{1}{\sqrt{2\pi}} \times e^{-\frac{1}{2}\left(\frac{X_i - x}{h}\right)^2} \tilde{f}(x) = \frac{1}{nh} \times \sum_{i=1}^{N} K\left(\frac{X_i - x}{h}\right) = \frac{1}{h\sqrt{2\pi}} \times \frac{1}{n} \sum_{i=1}^{N} e^{-\frac{1}{2}\left(\frac{X_i - x}{h}\right)^2}$$

We have to take weight into account, so the below codes integrate some weight issues.

```r
# I choose h = 0.068 as the best bandwidth from runninig the density function
# d <- density(data_0$lwage1, weights = data_0$weight)
# plot(d, lwd = 2, main = "Default kernel density plot")

find_density <- function(tau, df, h=0.068){
  # (1) find x = q_tau
  q_tau <- find_quantile(tau, df)
  # (2) calculate the first term (weight)
  first_term <- 1 / (h*sqrt(2*pi))
  # (3) calculate the second term (weight)
  second_term <- c()
  for (i in 1:nrow(df)) {
    second_term[i] <- exp(-0.5*(((df$lwage1[i] - q_tau)/h)^2)) * df$weight[i]
  }
  second_term_sum <- sum(second_term)
  return(first_term*second_term_sum)
}

# modified from: https://medium.com/analytics-vidhya/kernel-density-estimation-kernel-construction-and-
```

The Second Stage: RIF Regression

$$\text{RIF}(y; q_\tau, F) = q_\tau + \frac{\tau - 1\{y \leq q_\tau\}}{f_y(q_\tau)}$$

RIF for each quantile

Write a function to get the RIF value for each τ

```
# create a function whether: if t==TRUE, then 1; if t==FALSE, then 0
whether <- function(t) ifelse(t, 1, 0)

find_RIF <- function(tau, df) {
  RIF <- c()
  q_tau <- find_quantile(tau, df)
  fq_tau <- find_density(tau, df)
  for (i in 1:nrow(df)) {
  IF <- (tau - whether(df$lwage1[i] <= q_tau)) / fq_tau
  RIF[i] <- q_tau + IF
  }
  return(RIF)
}
```

RIF Regression for each quantile

Write a function to get reweighted RIF Regression result

```
regress_RIF <- function(tau, df) {
  # input
  df <- df %>%
    select(-c("ed_0", "ex_1", "occd_11", "indd_1"))

  X <- df[, c(22:23, 30:71)]  # collenearity concern
  X <- as.matrix(X)
  intercept <- rep(1, nrow(X))
  X <- cbind(intercept, X)

  Y <- find_RIF(tau, df)
  Y <- as.matrix(Y)
  weight <- as.vector(df$weight)

  # closed-form solutions for coefficients
  coef <- solve(t(X*weight) %*% X) %*% t(X*weight) %*% Y

  # drop the coefficient for intercept
  coef <- coef[-1]

  return(coef)
}
```

Decompose the wage difference

```
# find the expected values
E_x_1 <- c()
E_x_0 <- c()
ind.x <- c(22:23, 30:71)
for (i in 1:44) {
  E_x_1[i] <- sum(data_1[, ind.x[i]] * data_1$weight)
  E_x_0[i] <- sum(data_0[, ind.x[i]] * data_0$weight)
}

# get the names for covariates
cov_name <- names(data_0)[c(22:23, 30:71)]

############## tau = 10 ##############
g_0_10 <- regress_RIF(tau = 0.1, df = data_0)
g_1_10 <- regress_RIF(tau = 0.1, df = data_1)
g_2_10 <- regress_RIF(tau = 0.1, df = data_2)
s_10 <- t(E_x_1) %*% (g_1_10 - g_2_10)
x_10 <- t(E_x_1 - E_x_0) %*% g_0_10 + t(E_x_1) %*% (g_2_10 - g_0_10)

cat("wage structure effect for 10th quantile is", s_10)
```

```
## wage structure effect for 10th quantile is 0.06578067
```

```
cat("composite effect for 10th quantile is", x_10)
```

```
## composite effect for 10th quantile is -0.0005485708
```

```
cat("report approx error,", t(E_x_1) %*% (g_2_10 - g_0_10))
```

```
## report approx error, 0.01998808
```

```
s_10_d <- E_x_1 * (g_1_10 - g_2_10)
x_10_d <- (E_x_1 - E_x_0) * g_0_10

############## tau = 50 ##############

g_0_50 <- regress_RIF(tau = 0.5, df = data_0)
g_1_50 <- regress_RIF(tau = 0.5, df = data_1)
g_2_50 <- regress_RIF(tau = 0.5, df = data_2)
s_50 <- t(E_x_1) %*% (g_1_50 - g_2_50)
x_50 <- t(E_x_1 - E_x_0) %*% g_0_50 + t(E_x_1) %*% (g_2_50 - g_0_50)

cat("wage structure effect for 50th quantile is", s_50)
```

```
## wage structure effect for 50th quantile is 0.1626186
```

```r
cat("composite effect for 50th quantile is", x_50)
```

```
## composite effect for 50th quantile is -0.0461798
```

```r
cat("report approx error,", t(E_x_1) %*% (g_2_50 - g_0_50))
```

```
## report approx error, 0.002587849
```

```r
s_50_d <- E_x_1 * (g_1_50 - g_2_50)
x_50_d <- (E_x_1 - E_x_0) * g_0_50

############## tau = 90 ##############
g_0_90 <- regress_RIF(tau = 0.9, df = data_0)
g_1_90 <- regress_RIF(tau = 0.9, df = data_1)
g_2_90 <- regress_RIF(tau = 0.9, df = data_2)
s_90 <- t(E_x_1) %*% (g_1_90 - g_2_90)
x_90 <- t(E_x_1 - E_x_0) %*% g_0_90 + t(E_x_1) %*% (g_2_90 - g_0_90)

cat("wage structure effect for 90th quantile is", s_90)
```

```
## wage structure effect for 90th quantile is 0.06800075
```

```r
cat("composite effect for 90th quantile is", x_90)
```

```
## composite effect for 90th quantile is 0.03423352
```

```r
cat("report approx error,", t(E_x_1) %*% (g_2_90 - g_0_90))
```

```
## report approx error, 0.1079061
```

```r
s_90_d <- E_x_1 * (g_1_90 - g_2_90)
x_90_d <- (E_x_1 - E_x_0) * g_0_90

d_all <- cbind(s_10_d, s_50_d, s_90_d, x_10_d, x_50_d, x_90_d)

rownames(d_all) <- cov_name

dd <- kbl(d_all, longtable = T, booktabs = T, format = "latex", digits = 4,
    col.names = c("10_th", "50_th", "90_th", "10_th", "50_th", "90_th"),
    caption = "Detailed Wage Decomposition after IPW Weighting and RIF (New)") %>%
add_header_above(c(" ", "Wage Structual Effect" = 3, "Composite Effect" = 3)) %>%
kable_styling(latex_options = c("repeat_header"))

write.table(dd[[1]],"newdf.txt",sep="\t",row.names=FALSE)

all <- cbind(s_10, s_50, s_90, x_10, x_50, x_90)

aa <- kbl(all, longtable = T, booktabs = T, format = "latex", digits = 4,
    col.names = c("10_th", "50_th", "90_th", "10_th", "50_th", "90_th"),
```

```
    caption = "Aggregate Wage Decomposition after IPW Weighting and RIF (New)") %>%
add_header_above(c("Wage Structual Effect" = 3, "Composite Effect" = 3)) %>%
kable_styling(latex_options = c("repeat_header"))

write.table(aa[[1]],"newdf.txt",sep="\t",row.names=FALSE)
```

Appendix C

Table 3: Detailed Wage Decomposition after IPW Weighting and RIF

	Wage Structual Effect			Composite Effect		
	10_th	50_th	90_th	10_th	50_th	90_th
covered	-0.0023	0.0059	-0.0047	-0.0007	-0.0019	0.0001
nmarr	-0.0105	-0.0013	0.0354	0.0021	0.0076	0.0080
ed_0	0.0019	0.0032	0.0010	-0.0018	0.0024	0.0012
ed_1	0.0041	0.0054	0.0023	0.0013	0.0029	0.0004
ed_2	0.0161	0.0250	0.0117	-0.0014	-0.0031	-0.0005
ed_3	0.0167	0.0154	0.0224	0.0015	0.0043	0.0022
ed_4	0.0113	0.0168	-0.0254	0.0019	0.0065	0.0070
ed_5	0.0040	-0.0011	-0.0009	-0.0072	-0.0043	0.0003
ex_1	0.0046	0.0021	-0.0029	0.0007	0.0006	0.0003
ex_2	0.0047	0.0045	-0.0052	-0.0054	-0.0064	-0.0053
ex_3	0.0017	0.0038	0.0047	-0.0049	-0.0062	-0.0038
ex_4	0.0015	0.0048	0.0080	-0.0045	-0.0058	-0.0036
ex_5	0.0012	0.0041	0.0090	-0.0009	-0.0011	-0.0007
ex_6	0.0006	0.0032	0.0081	0.0012	0.0016	0.0010
ex_7	-0.0020	0.0021	0.0076	0.0037	0.0046	0.0019
ex_8	-0.0013	-0.0002	0.0156	0.0006	-0.0009	-0.0089
ex_9	0.0002	0.0019	-0.0152	0.0002	0.0001	-0.0018
occd_11	-0.0005	-0.0024	-0.0133	-0.0002	-0.0037	-0.0101
occd_12	0.0024	-0.0005	0.0038	-0.0002	-0.0017	-0.0053
occd_21	0.0004	-0.0014	-0.0119	0.0003	0.0006	-0.0145
occd_22	0.0002	0.0001	-0.0005	-0.0001	0.0001	0.0006
occd_23	-0.0015	-0.0029	0.0150	-0.0003	-0.0020	-0.0033
occd_24	0.0007	0.0018	0.0055	-0.0003	-0.0011	-0.0017
occd_25	0.0004	-0.0001	0.0021	0.0008	0.0021	0.0034
occd_30	0.0079	0.0032	-0.0080	0.0035	0.0028	-0.0008
occd_40	-0.0070	0.0019	0.0217	-0.0018	-0.0068	-0.0082
occd_41	0.0004	0.0006	0.0019	-0.0011	-0.0023	-0.0026
occd_42	0.0001	-0.0002	0.0007	0.0000	-0.0003	-0.0012
occd_50	0.0023	0.0007	0.0380	0.0011	0.0246	0.0434
occd_60	-0.0003	-0.0003	0.0034	-0.0008	-0.0045	-0.0058
occd_70	-0.0014	-0.0136	0.0392	-0.0042	-0.0392	-0.0631
occd_80	0.0038	0.0169	0.0007	-0.0001	-0.0004	-0.0001

Table 3: Detailed Wage Decomposition after IPW Weighting and RIF *(continued)*

	Wage Structual Effect			Composite Effect		
	10_th	50_th	90_th	10_th	50_th	90_th
occd_90	0.0025	0.0075	0.0008	0.0008	0.0036	0.0004
occd_91	0.0012	0.0064	0.0007	0.0001	0.0008	0.0007
indd_1	0.0013	0.0048	0.0016	-0.0004	-0.0033	-0.0027
indd_2	-0.0029	0.0113	-0.0055	-0.0066	-0.0206	-0.0076
indd_3	0.0033	0.0085	0.0011	0.0001	0.0006	0.0007
indd_4	0.0007	0.0114	0.0060	-0.0023	-0.0048	-0.0039
indd_5	0.0010	0.0041	0.0021	-0.0002	-0.0015	-0.0004
indd_6	0.0040	0.0122	-0.0049	0.0002	0.0007	-0.0007
indd_7	0.0038	0.0129	0.0019	0.0013	0.0069	0.0048
indd_8	0.0003	0.0013	0.0004	-0.0001	-0.0006	-0.0007
indd_9	0.0002	0.0077	0.0009	-0.0006	-0.0010	-0.0006
indd_10	0.0050	0.0081	-0.0007	0.0003	0.0021	0.0044

Table 4: Detailed Wage Decomposition after IPW Weighting and RIF (New)

	Wage Structual Effect			Composite Effect		
	10_th	50_th	90_th	10_th	50_th	90_th
covered	-0.0015	0.0034	-0.0062	-0.0012	-0.0035	-0.0002
nmarr	-0.0068	-0.0030	0.0120	0.0019	0.0070	0.0075
ed_0	0.0020	0.0000	0.0001	-0.0018	0.0018	0.0012
ed_1	0.0030	0.0021	0.0016	0.0008	0.0019	0.0003
ed_2	0.0127	0.0042	0.0043	-0.0001	-0.0003	-0.0001
ed_3	0.0108	0.0035	0.0093	0.0021	0.0061	0.0033
ed_4	0.0110	0.0035	-0.0234	0.0018	0.0063	0.0070
ed_5	0.0011	0.0006	-0.0015	-0.0078	-0.0047	0.0005
ex_1	0.0013	0.0027	0.0006	0.0018	0.0016	0.0007
ex_2	-0.0009	0.0034	-0.0060	-0.0047	-0.0058	-0.0045
ex_3	-0.0046	0.0015	0.0043	-0.0042	-0.0053	-0.0029
ex_4	-0.0035	0.0023	0.0143	-0.0043	-0.0056	-0.0033
ex_5	-0.0032	0.0035	0.0099	-0.0002	-0.0002	-0.0001
ex_6	-0.0024	0.0044	0.0052	0.0014	0.0019	0.0010
ex_7	-0.0032	0.0017	0.0002	0.0033	0.0041	0.0015
ex_8	0.0000	0.0010	0.0125	0.0005	-0.0009	-0.0072
ex_9	0.0006	0.0015	-0.0121	0.0001	0.0000	-0.0013

Table 4: Detailed Wage Decomposition after IPW Weighting and RIF *(continued)*

	Wage Structual Effect			Composite Effect		
	10_th	50_th	90_th	10_th	50_th	90_th
occd_11	-0.0024	-0.0031	-0.0169	0.0000	-0.0031	-0.0087
occd_12	0.0022	0.0001	0.0011	-0.0001	-0.0012	-0.0035
occd_21	0.0003	-0.0058	-0.0143	0.0003	0.0009	-0.0136
occd_22	0.0001	-0.0001	-0.0007	-0.0001	0.0003	0.0012
occd_23	0.0007	0.0004	0.0070	-0.0002	-0.0015	-0.0024
occd_24	0.0008	0.0016	0.0038	-0.0003	-0.0010	-0.0016
occd_25	0.0011	-0.0009	0.0006	0.0008	0.0023	0.0032
occd_30	0.0038	0.0066	0.0054	0.0028	0.0031	-0.0007
occd_40	-0.0008	0.0060	0.0109	-0.0018	-0.0071	-0.0084
occd_41	0.0005	0.0001	0.0007	-0.0008	-0.0019	-0.0021
occd_42	0.0001	0.0000	0.0002	0.0000	-0.0001	-0.0006
occd_50	0.0064	0.0148	0.0185	0.0006	0.0191	0.0325
occd_60	0.0002	0.0006	0.0019	-0.0006	-0.0038	-0.0047
occd_70	0.0000	0.0062	0.0190	-0.0038	-0.0427	-0.0663
occd_80	0.0043	0.0161	-0.0026	-0.0001	-0.0010	-0.0006
occd_90	0.0022	0.0044	-0.0006	0.0006	0.0045	0.0015
occd_91	0.0017	0.0044	-0.0009	0.0000	0.0013	0.0012
indd_1	0.0007	0.0024	-0.0004	-0.0001	-0.0019	-0.0017
indd_2	0.0035	0.0114	-0.0112	-0.0064	-0.0242	-0.0101
indd_3	0.0033	0.0071	-0.0024	0.0001	0.0008	0.0012
indd_4	0.0029	0.0143	0.0265	-0.0018	-0.0049	-0.0035
indd_5	0.0011	0.0034	0.0000	-0.0001	-0.0010	-0.0003
indd_6	0.0054	0.0100	-0.0012	0.0001	0.0003	-0.0002
indd_7	0.0049	0.0104	0.0023	0.0011	0.0076	0.0056
indd_8	0.0003	0.0008	0.0000	0.0001	0.0004	0.0004
indd_9	0.0014	0.0058	-0.0020	-0.0004	-0.0007	-0.0004
indd_10	0.0045	0.0092	-0.0017	0.0002	0.0027	0.0053