Lecture 3

Hisam Sabouni

9/23/2018

Civilian non-institutional population is defined as persons 16 years of age and older residing in the 50 states and the District of Columbia, who are not inmates of institutions (e.g., penal and mental facilities, homes for the aged), and who are not on active duty in the Armed Forces.

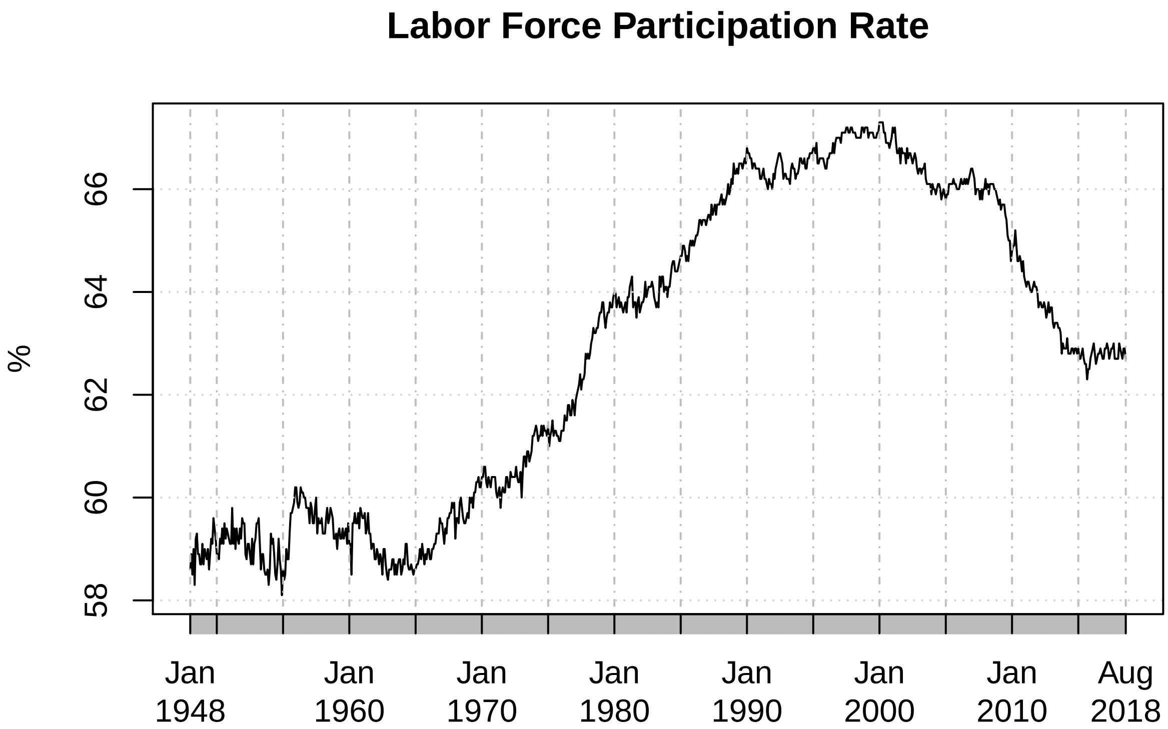
Civilian non institutional population = Labor force + Not in Labor force

The labor force participation rate is the ration of the labor force to the civilian non institutional population:

```r
library(quantmod)
# Civilian Labor Force Participation Rate
# (CIVPART)
getSymbols("CIVPART", src = "FRED")
```

```
## [1] "CIVPART"
```

```r
plot(CIVPART, ylab = "%", main = "Labor Force Participation Rate")
```



See here for more: https://www.bls.gov/opub/mlr/2016/article/labor-force-participation-what-has-happened-since
htm

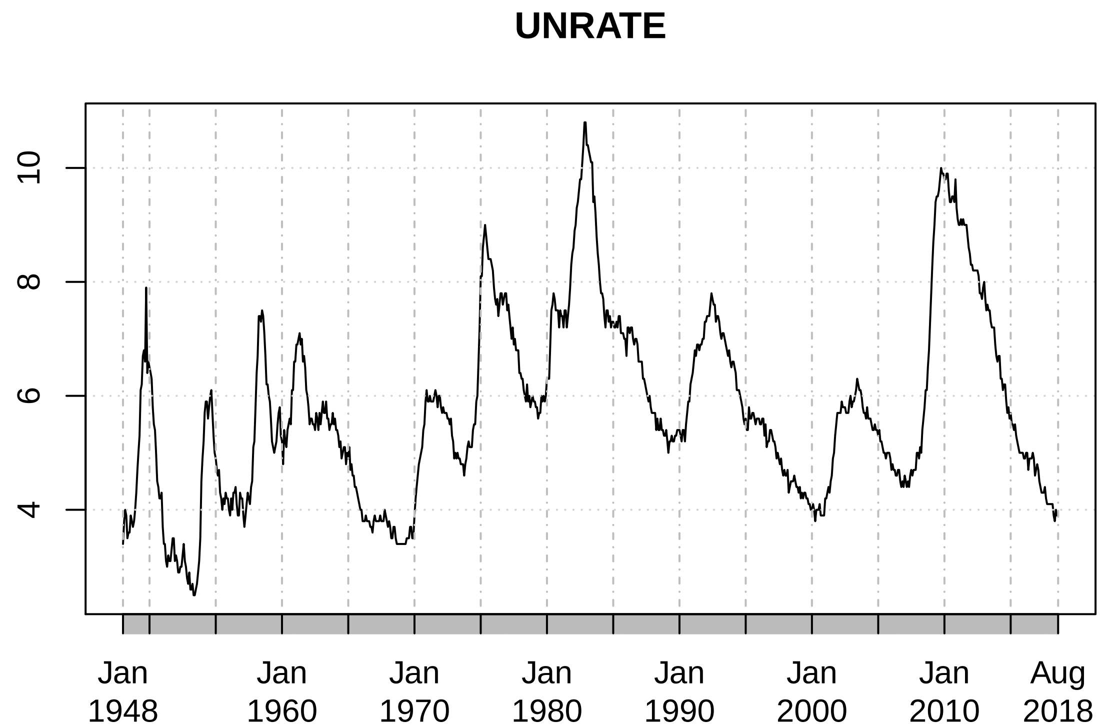
Let L denote the total labor force, E the number of employed workers (Anyone aged 16+ in the civilian non-institutional population who worked in the last week), and U the number of unemployed workers (Those aged 16 or more who weren't employed, but are available for work and are actively looked for a job within the past four weeks). The unemployment rate is U/L

$$L = E + U$$

```r
# Civilian Unemployment Rate (UNRATE) Median
# Duration of Unemployment (UEMPMED)
getSymbols(c("UNRATE", "UEMPMED"), src = "FRED")
```

```
## [1] "UNRATE"  "UEMPMED"
```

```r
plot(UNRATE)
```



Rate of Job Separation (s): The fraction of employed individuals who lose or elave thier job each month. If 1 percent of the employed lose their jobs each month ($s = 0.01$), this implies that the average employment lasts $1/0.01 = 100$ months, about 8 years.

Rate of Job Finding (f): The fraction of unemployed individuals who find a job each month. If 20% of the unemployed find a job each month ($f = 0.2$), the average time unemployed is $1/0.2 = 5$ months.

The rate of job separation and job finding determine the rate of unemployment. If the labor market is in a steady-state, then the number of people finding jobs fU mus equal the number of people losing jobs sE.

$$fU = sE$$

$$fU = s(L - U)$$

$$\frac{fU}{L} = \frac{s}{L}(L - U)$$

$$\frac{fU}{L} = s(1 - \frac{U}{L})$$

$$\frac{fU}{L} + \frac{sU}{L} = s$$

$$\frac{U}{L}(f + s) = s$$

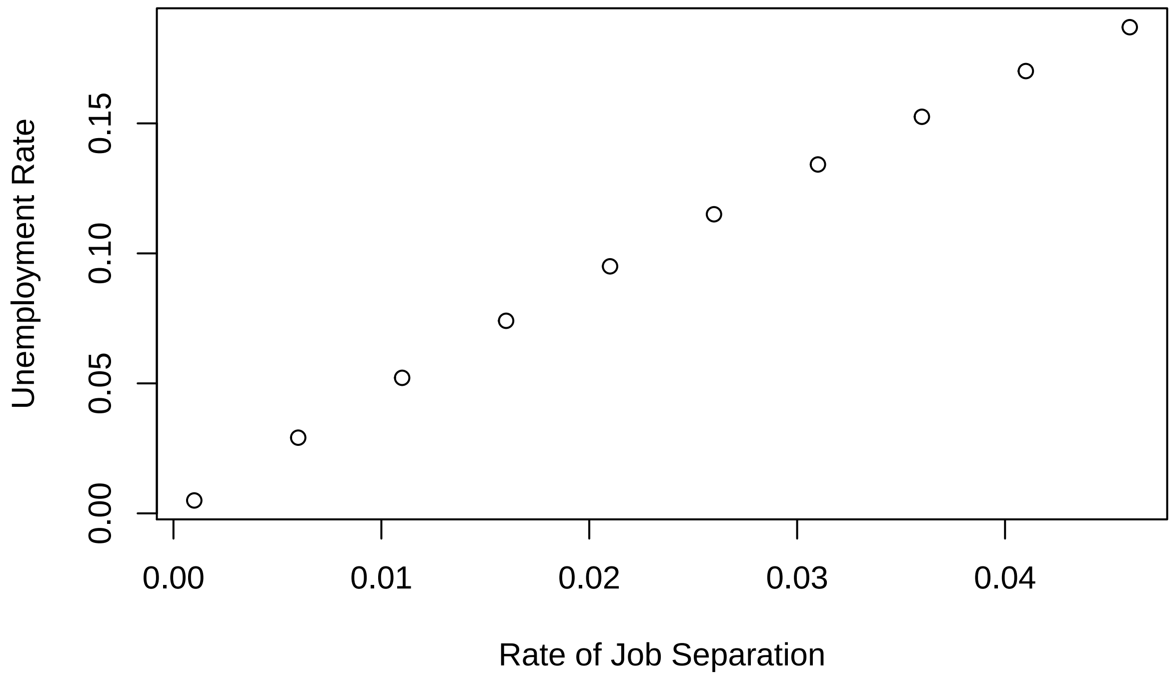
$$\frac{U}{L} = \frac{s}{(f + s)}, \text{ or, } \frac{U}{L} = \frac{1}{(f/s + 1)}$$

Here we have that the unemployment rate is a function of the rate of job finding and the rate of job separation. We can write some code to simulate this model to see how the unemployment rate varies as the rate of job separation and job finding vary:

```
job_separation <- seq(0.001, 0.05, by = 0.005)
# Average years employed before becoming
# unemployed#
(1/job_separation)/12
```

```
##  [1] 83.333333 13.888889  7.575758  5.208333  3.968254  3.205128  2.688172
##  [8]  2.314815  2.032520  1.811594
```

```
job_finding <- 0.2
unemployment_rate <- 1/(job_finding/job_separation +
    1)
plot(job_separation, unemployment_rate, ylab = "Unemployment Rate",
    xlab = "Rate of Job Separation")
```

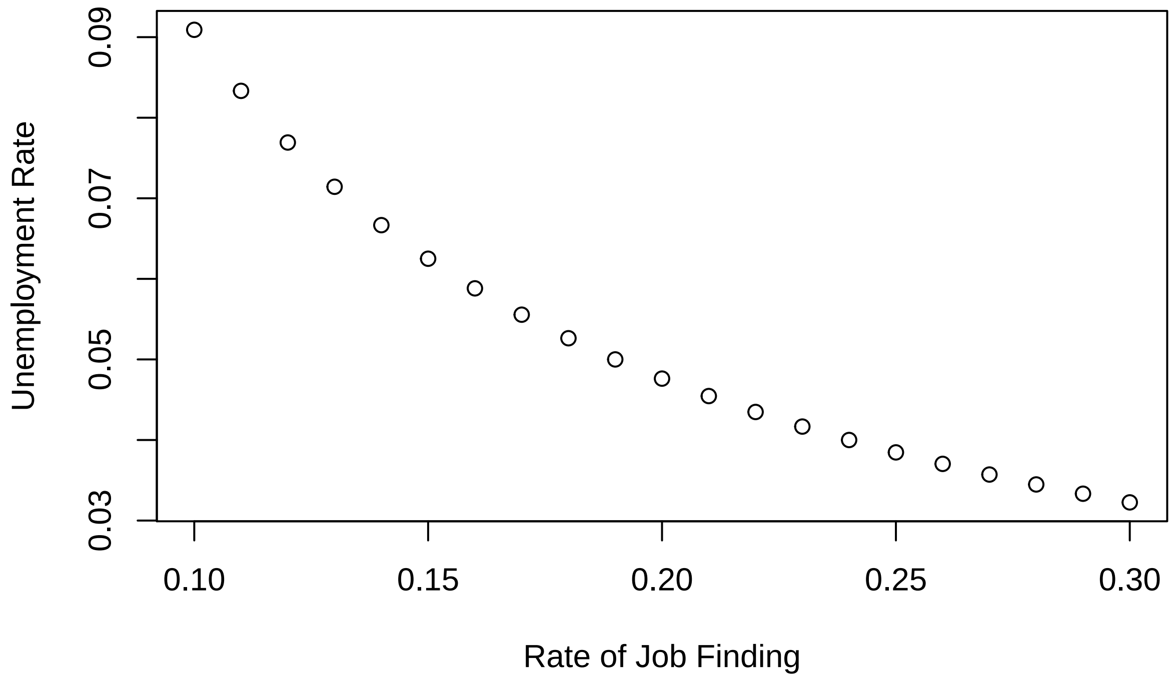


Given a rate of job finding as the rate of job separation increases we have an increasing fraction of the labor force that is unemployed.

```
job_separation <- 0.01
job_finding <- seq(0.1, 0.3, by = 0.01)
# Average months unemployed before becoming
# employed#
(1/job_finding)
```

```
##  [1] 10.000000  9.090909  8.333333  7.692308  7.142857  6.666667  6.250000
##  [8]  5.882353  5.555556  5.263158  5.000000  4.761905  4.545455  4.347826
## [15]  4.166667  4.000000  3.846154  3.703704  3.571429  3.448276  3.333333
```

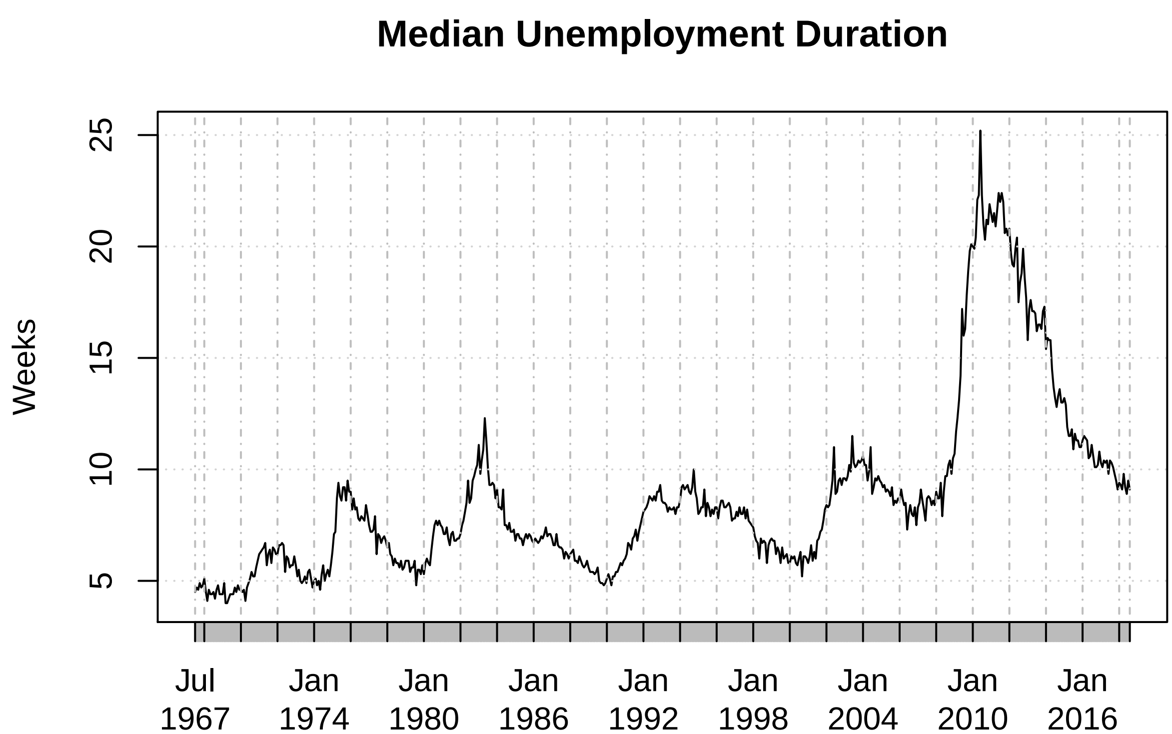
```
unemployment_rate <- 1/(job_finding/job_separation +
    1)
plot(job_finding, unemployment_rate, ylab = "Unemployment Rate",
    xlab = "Rate of Job Finding")
```



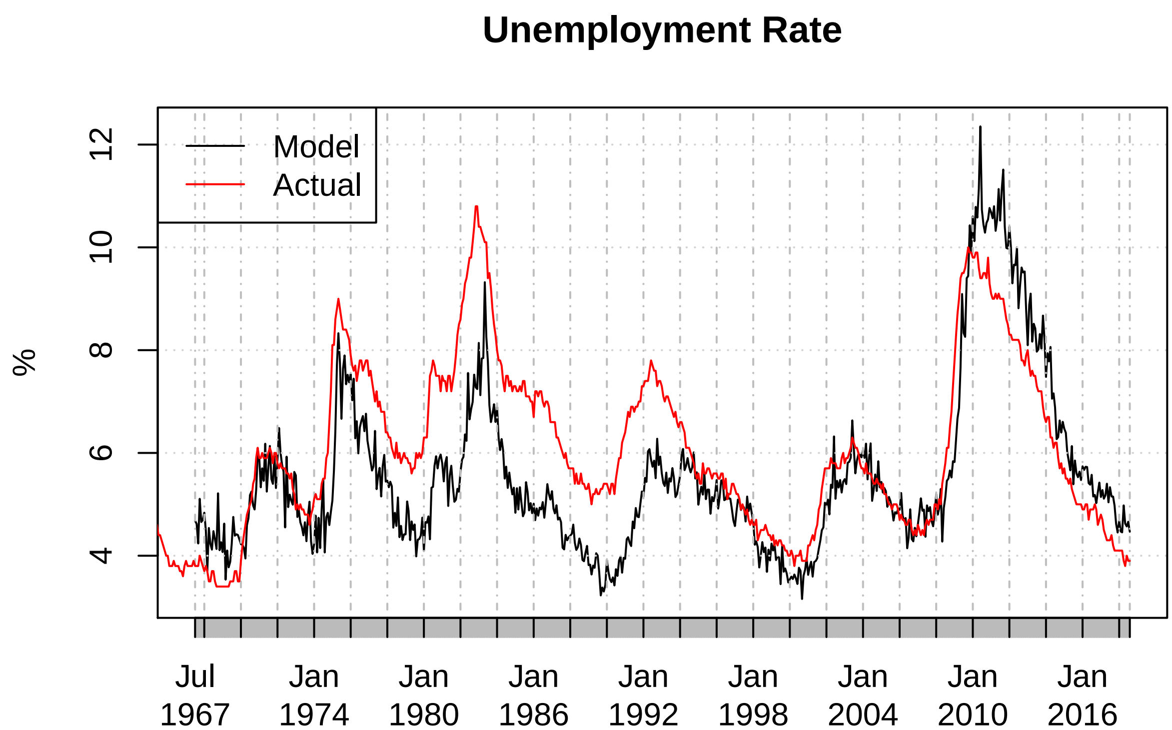

```
# Median Duration of Unemployment (UEMPMED)
getSymbols("UEMPMED", src = "FRED")
```

```
## [1] "UEMPMED"
```

```
plot(UEMPMED, ylab = "Weeks", main = "Median Unemployment Duration")
```



```r
job_finding <- 1/(UEMPMED/4)
# https://www.bls.gov/bls/news-release/home.htm#TENURE
# Median around 4 years for age groups and
# increasing over time
set.seed(2)
job_separation <- seq(2, 4, length.out = length(job_finding)) +
    rnorm(length(job_finding), 0, 0.1)
job_separation <- 1/(job_separation * 12)
unemployment_rate <- 1/(job_finding/job_separation +
    1)
plot(100 * unemployment_rate, main = "Unemployment Rate",
    ylab = "%")
lines(UNRATE, col = 2)
legend("topleft", legend = c("Model", "Actual"),
    lty = 1, col = 1:2)
```



As the rate of job finding increases we have the a decreasing fraction of the labor force that is unemployed.

Any policy aimed at lowering the natural rate of unemployment must either reduce the rate of job separation or increase the rate of job finding. For example, government programs

that help disseminate information about job vacancies are increasing the rate of job finding, while programs such as unemployment insurance tend reduce the rate of job finding (50% of former wage for up to 26 weeks).

If job finding were instantaneous ($f = 1$) then all individuals that become unemployed (s) would be able to immediately find work. When $f < 1$ there exist frictions in the labor market. Frictions in the labor market arise when workers have different abilities than what jobs require, when there are geographic restrictions (eg. shortage of workers in New York versus excess supply of workers in California),or, when there is limited information flow regarding vacancies. A change in composition of demand among industries or regions is called a sectoral shift.

```r
# https://fred.stlouisfed.org/release/tables?rid=50&eid=4881&snid=5205
# All Employees: Goods-Producing Industries
# (USGOOD) All Employees: Trade,
# Transportation and Utilities (USTPU) All
# Employees: Information Services (USINFO) All
# Employees: Financial Activities (USFIRE) All
# Employees: Professional and Business
# Services (USPBS) All Employees: Education
# and Health Services (USEHS) All Employees:
# Leisure and Hospitality (USLAH) All
# Employees: Other Services (USSERV) All
# Employees: Government (USGOVT)

symbols <- c("USGOOD", "USTPU", "USINFO", "USFIRE",
    "USPBS", "USEHS", "USLAH", "USSERV", "USGOVT")
employees_by_sector <- new.env()
getSymbols(symbols, src = "FRED", env = employees_by_sector)
```

```
## [1] "USGOOD" "USTPU"  "USINFO" "USFIRE" "USPBS"  "USEHS"  "USLAH"  "USSERV"
## [9] "USGOVT"
```

```r
employees_by_sector <- as.list(employees_by_sector)
sectoral_employment <- do.call("merge", employees_by_sector)
head(sectoral_employment)
```

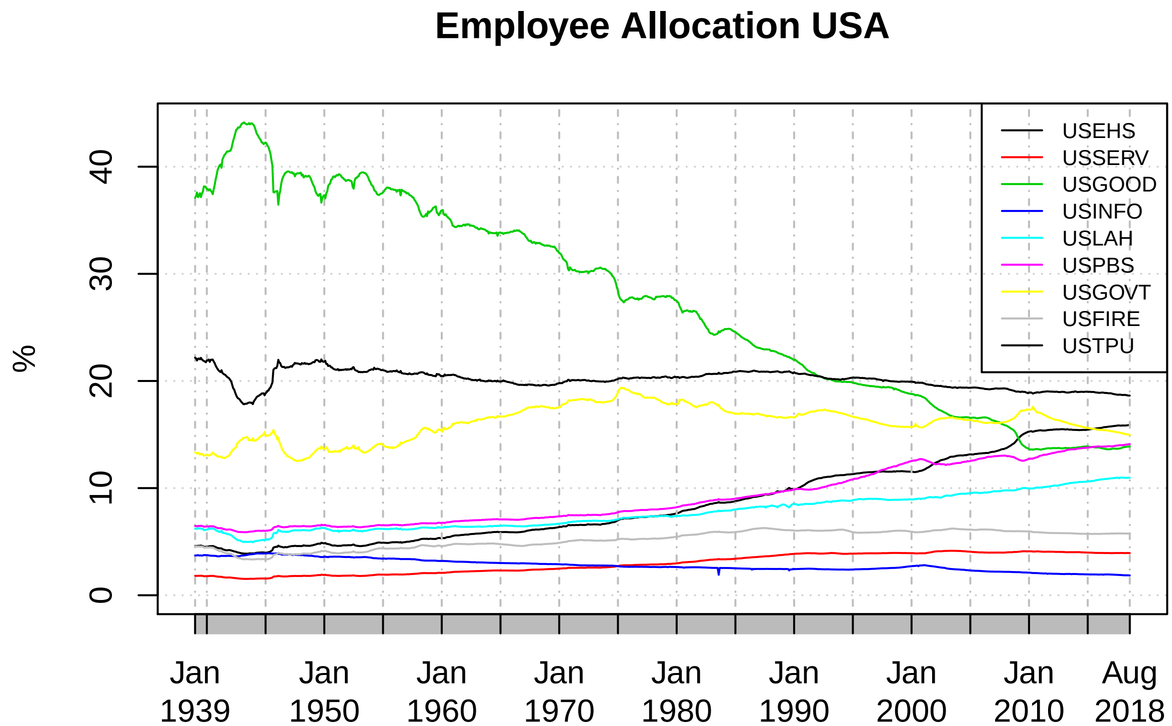
```
##            USEHS USSERV USGOOD USINFO USLAH USPBS USGOVT USFIRE USTPU
## 1939-01-01  1381    543  11098   1112  1860  1934   3988   1371  6636
```

```
## 1939-02-01  1384    544  11221   1118  1867  1941   4001   1372  6653
## 1939-03-01  1387    546  11383   1126  1877  1949   4002   1376  6634
## 1939-04-01  1389    546  11182   1127  1887  1948   4002   1370  6643
## 1939-05-01  1399    550  11298   1125  1888  1959   4006   1379  6696
## 1939-06-01  1406    553  11447   1131  1895  1970   4003   1381  6716
```

```r
sectoral_employment_percent <- 100 * sectoral_employment/apply(sectoral_employment,
    1, sum)
head(sectoral_employment_percent)
```

```
##               USEHS   USSERV   USGOOD   USINFO    USLAH    USPBS   USGOVT
## 1939-01-01 4.615179 1.814658 37.08853 3.716205 6.215954 6.463256 13.32754
## 1939-02-01 4.597854 1.807249 37.27783 3.714162 6.202452 6.448291 13.29192
## 1939-03-01 4.580581 1.803170 37.59247 3.718626 6.198811 6.436592 13.21664
## 1939-04-01 4.615538 1.814315 37.15691 3.744933 6.270353 6.473051 13.29833
## 1939-05-01 4.617162 1.815182 37.28713 3.712871 6.231023 6.465347 13.22112
## 1939-06-01 4.609534 1.812996 37.52869 3.707954 6.212707 6.458593 13.12373
##              USFIRE    USTPU
## 1939-01-01 4.581760 22.17692
## 1939-02-01 4.557988 22.10226
## 1939-03-01 4.544254 21.90885
## 1939-04-01 4.552402 22.07417
## 1939-05-01 4.551155 22.09901
## 1939-06-01 4.527572 22.01823
```

```r
plot(sectoral_employment_percent[, 1], typ = "l",
    ylim = c(0, max(sectoral_employment_percent)),
    main = "Employee Allocation USA", ylab = "%")
for (i in 2:ncol(sectoral_employment_percent)) {
    lines(sectoral_employment_percent[, i], col = i)
}
legend("topright", legend = colnames(sectoral_employment_percent),
    lty = 1, col = 1:ncol(sectoral_employment_percent),
    cex = 0.7)
```



Job Search Models

Citation Note Section is heavily borrowed from Quantecon: https://lectures.quantecon.org/py/mccall_model.html, an excellent resource by Sargent and Stachurski that I recommend you all take a look at.

Models that attempt to capture the dynamics of the labor matching market are called job search models. One of the first models was proposed in 1970 by McCall. McCall modeled job search as a function of current and likely future wages, impatience, and unemployment compensation.

Set up

An unemployed worker receives in each period a job offer at wage W_t

At time t, the worker has two choices:

1. Accept an offer and work permanently at constant wage W_t
2. Reject the offer, receive unemployment compensation c, and reconsider next period

The wage sequence $\{W_t\}$ is assumed to be iid with probability mass function $p_1, \ldots, p_n$. Here p_i is the probability of observing wage offer $W_t = w_i$ in the set $w_1, \ldots, w_n$. The worker is infinitely lived and aims to maximize the expected discounted sum of earnings:

$$E \sum_{t=0}^{\infty} \beta^t Y_t$$

$\beta \in (0, 1)$ is the discount factor, where the closer to zero β is the more the worker discounts future utility relative to current utility.

Y_t is income, that is equal to wage W_t when employed and c unemployment compensation when unemployed. The worker faces a trade-off, where waiting too long for a good offer is costly, since the future is discounted and accepting too early is costly, since better offers might arrive in the future. We can use dynamic programming to tackle this issue.

Dynamic programming refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP). The key idea of dynamic programming is the use of value functions to organize and structure the search for good policies (eg. actions to take in a particular state). See Sutton and Barto's 1998 book on Reinforcement learning for more.

The value function in our McCall model should aim to balance the trade off of current and future rewards:

1. The current payoffs we get from different choices
2. The different states that those choices will lead to next period.

Let $V(w)$ be the total lifetime value accruing to an unemployed worker who enters the current period unemployed but with wage offer w in hand. Think of $V(w)$ as a function that assigns to each possible wage w the maximal lifetime value that can be obtained with that offer in hand.

$$V(w) = max\{\frac{w}{1-\beta}, c + \beta \sum_{i=1}^{n} V(w_i) p_i\}$$

For every possible w_i in $w_1, \ldots, w_n$.

One should note that the above is a piece-wise linear function (i.e. non-linear).

The first term in the max operation is the lifetime payoff from accepting current offer w, since $\sum_{i=1}^{\infty} w\beta^i = \frac{w}{1-\beta}$.

The second term is the continuation value, which is the lifetime payoff from rejecting the current offer and then behaving optimally in all subsequent periods.

Given a state w our McCall worker needs to decide whether to reject or accept the offer. Therefore, we have to map from $\mathbb{R}$ to $\{0,1\}$, where 1 means accept and 0 means reject:

$$\sigma(w) := \mathbf{1}\{\frac{w}{1-\beta} \geq c + \beta \sum_{i=1}^{n} V(w_i)p_i\}$$

or,

$$\sigma(w) := \mathbf{1}\{w \geq \bar{w}\}$$

where

$$\bar{w} = (1-\beta)(c + \beta \sum_{i=1}^{n} V(w_i)p_i)\}$$

Here $\bar{w}$ is called the reservation wage. The agent should accept if and only if the current wage offer exceeds the reservation wage.

We can iteratively approximate the value function by evaluating the function at a variety of wages.

More precisely, we can use the following algorithm to approximate the value function:

1. Pick an arbitrary initial guess $v \in \mathbb{R}^n$
2. Compute a new vector $v' \in \mathbb{R}^n$ via:

$$v_i' = max\{\frac{w}{1-\beta}, c + \beta \sum_{i=1}^{n} v_i p_i\}$$

3. Calculate a measure of the deviation between v and v', such as $max_i|v_i - v_i'|$
4. If the deviation is larger than some fixed tolerance, set $v = v'$ and go to step 2, else continue.
5. Return v

The above algorithm will converge on the true solution as the fixed tolerance goes to zero (See Banach contraction mapping).

To put the ideas above into action we can create a distribution of wages that our McCall worker will observe:

```
# McCall Search Model
n <- 50
w_min <- 10
w_max <- 60
w_vals <- seq(w_min, w_max, length.out = n + 1)
set.seed(1)
dist_pdf <- diff(pnorm(c(10, w_vals), mean = mean(w_vals),
    sd = sd(w_vals)/2))

plot(w_vals, dist_pdf, type = "h", ylab = "Probability",
    xlab = "Wage")
```

In the code above we arbitrarily create 51 wages ranging from 10 to 60 and assign probabilities to observing each wage through a normal distribution with a mean of 35 and a standard deviation of 7.4.

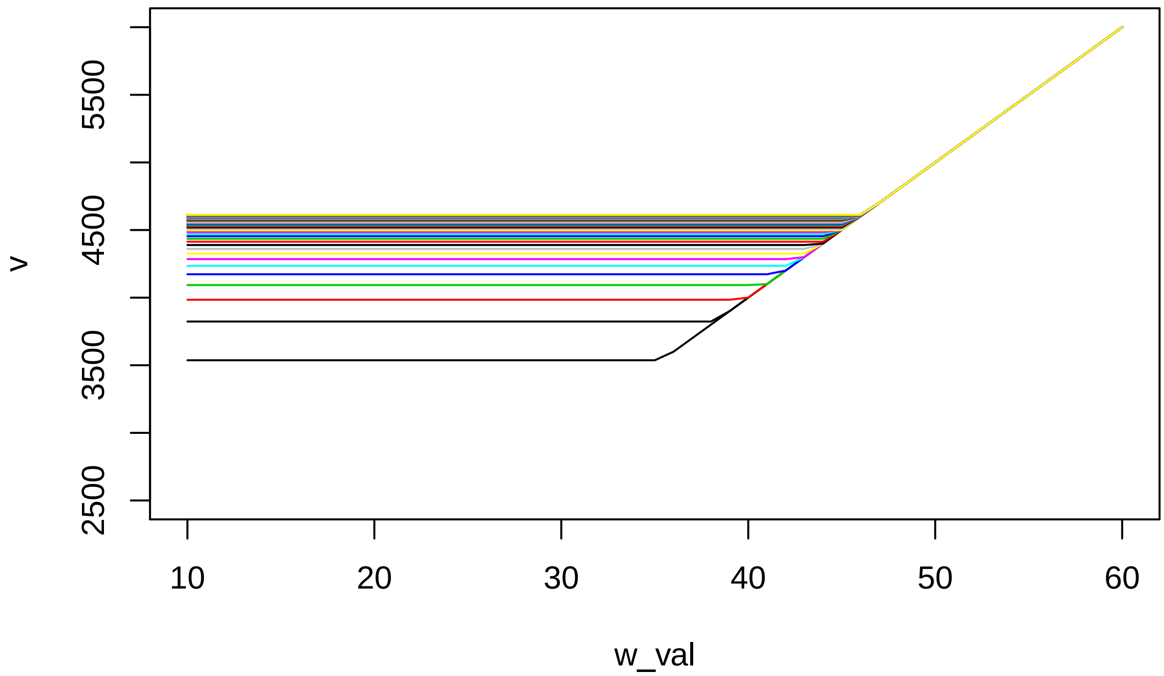
We can now write a function that implements dynamic programming for our McCall worker to find the reservation wage (i.e. the wage at which point the worker is indifferent between working and remaining unemployed). Recall that the reservation wage is reliant on the individuals discount factor as well as on the unemployment benefits the receive.

```
compute_reservation_wage <- function(benefits,
    beta, w_val, p_val, max_iter = 500, tol = 1e-06,
    generatePlots = T) {
    v <- w_val/(1 - beta)  #Value of accepting wage
    v_prime <- rep(0, length(v))  #Create v'
    i <- 0
    error <- tol + 1
    while (i < max_iter & error > tol) {
        for (j in 1:length(w_val)) {
            accept_val <- w_val[j]/(1 - beta)  #Value of accetping wage
            continue_val <- benefits + beta *
                sum(v * p_val)  #Value of continuing
            v_prime[j] <- max(accept_val, continue_val)  #Choose maximum
        }
        error <- max(abs(v_prime - v))  #Estimate error

        v <- v_prime  #Replace V with new V
        if (generatePlots) {
            if (i == 0) {
                plot(w_val, v, type = "l", ylim = c(2500,
                  6000))
            } else {
                lines(w_val, v, col = i)
            }
        }
        i <- i + 1
    }

    return((1 - beta) * (benefits + beta * sum(v *
        p_val)))
}

compute_reservation_wage(benefits = 25, beta = 0.99,
    w_val = w_vals, p_val = dist_pdf, generatePlots = T)
```




```
## [1] 46.12969
```

As we can see above, we have a piece-wise linear function that slowly converges on a reservation wage of approximately 46.1, for a worker with unemployment benefits of 25 and a discount factor (β) of 0.99. We can now use this model to see how an individuals reservation wage will vary based on the unemployment benefits they receive and on their individual patience.

```
unemployment_benefits <- seq(0, 30, length.out = 10)
discount_factor <- seq(0.9, 0.99, length.out = 10)
statics_matrix <- matrix(NA, 10, 10)
for (j in 1:ncol(statics_matrix)) {
    for (i in 1:nrow(statics_matrix)) {
        statics_matrix[i, j] <- compute_reservation_wage(benefits = unemployment_benefi
            beta = discount_factor[i], w_val = w_vals,
            p_val = dist_pdf, generatePlots = F)
    }
}
plot(statics_matrix[, 1], discount_factor, ylab = "beta",
    xlab = "Wage", xlim = c(min(statics_matrix),
        max(statics_matrix)), type = "l")
for (i in 2:ncol(statics_matrix)) {
    lines(statics_matrix[, i], discount_factor,
```

```r
        col = i)
}
text(statics_matrix[1, ], rep(0.91, ncol(statics_matrix)),
    round(unemployment_benefits), pos = 4)
text(44, 0.91, "<- Unemployment Benefits")
```


```r
pnorm(statics_matrix[nrow(statics_matrix), ncol(statics_matrix)],
    mean(w_vals), sd(w_vals)/2)
```

```
## [1] 0.9432125
```

As you can see above, as individuals have higher patience they will have a higher reservation wage. Furthermore, as they have higher unemployment benefits, they will also have a higher reservation wage. In both cases, having higher benefits and being more patient allows individuals to wait out longer for a potentially higher wage. In our highest simulated case with unemployment benefits at 30 and a discount factor of 0.99, the McCall worker has a reservation wage of 46.7, which is in the 94-th percentile of wages implied by our simulated distribution.

Wage Rigidity

Unemployment is also caused by wage rigidity; the failure of wages to adjust to a level at which labor supply equals labor demand. Wage rigidity arises due to:

- Minimum wage laws: The minimum wage may exceed the equilbrium wage of unskilled wokers (effects teenagers). Half of all hourly-paid workers earning the minimum wage or less are under 25.
- Labor unions: The wages of unionized workers are determined by bargaining between union leaders and firm managment. Often, the final agreement raises the wages above the equilibrium level and allows the firm to decide how many workers to employ (usually less workers). See for example page 62: https://hotelworkers.org/images/uploads/NYC_Hotel_Industry_Wide_Agreement.pdf
- Efficency wages: These theories hold that high wages make workers more productive and that a cut to wages would reduce worker morale/productivity.
  1. A higher wage can increase workers' food consumption, and thereby cause them to be better nourished and more productive.
  2. A higher wage can increase workers' effort in situations where the firm cannot monitor them perfectly.
  3. Paying a higher wage can improve workers' ability along dimensions the firm cannot observe. For instance, a firm offering higher wages may intice an applicant pool of mostly high-type workers which could lead to a more productive work force.
  4. A high wage can build loyalty among workers and induce high effort (Ford's $5 work day).

Suppose we have a large number, N, of identically competitive firms. The representative firm seeks to maximize its profits:

$$Profit = Y - wL$$

Where Y is the firm's output, w is the wage that it pays, and L is the amount of labor it hires. A firms output depends on the number of workers it employs and on their effort.

$$Y = F(eL),\ F' > 0,\ F'' < 0$$

where e denotes workers' effort. Let the level of effort could be a function of the wage they receive w, the wage paid by other firms w_a, and the unemployment rate u:

$$e = e(w, w_a, u),\ e_1 > 0,\ e_2 < 0,\ e_3 > 0$$

For all L identical workers.

$$max\ Profit = Y - wL$$

$$max\ Profit = F(e(w, w_a, u)L) - wL$$

$$\frac{\partial Profit}{\partial L} : F'(e(w, w_a, u)L)e(w, w_a, u) - w = 0$$

$$\frac{\partial Profit}{\partial L} : F'(e(w, w_a, u)L) = w/e(w, w_a, u)$$

Firms hire workers until the marginal product of effective labor equals its cost.

$$\frac{\partial Profit}{\partial w} : F'(e(w, w_a, u)L)Le_1(w, w_a, u) - L = 0$$

$$\frac{\partial Profit}{\partial w} : we_1(w, w_a, u)/e(w, w_a, u) = 1$$

The elasticity of effort with respect to wage is 1.

Let $w*$ and $L*$ denote the values of w and L that satisfy our above conditions. Total labor demanded becomes $NL*$ if the labor supply is $\bar{L}$ the fraction unemployed is $\bar{L} - NL*$. If $NL* > \bar{L}$, wages will increase to the point where supply and demand are in balance. Summers (1988) supposed that effort is given by:

$$e = (\frac{w-x}{x})^{\beta} \text{ if } w > x$$

$$e = 0 \text{ if } w < x$$

$$x = (1-bu)w_a$$

where $0 < \beta < 1$ and $b > 0$. x is a measure of labor-market conditions. β represents the elasticity of effort with respect to the premium firms pay over the index of labor-market conditions. If b equals 1, x is the wage paid at other firms multiplied by the fraction of workers who are employed. If $b < 1$, workers put less weight on unemployment; this could occur if there are unemployment benefits or if workers value leisure. If $b > 1$ workers greatly fear unemployment. If $w > x$, effort increases less than proportionately with $w - x$.

Recall that:

$$\frac{\partial Profit}{\partial w} : we_1(w, w_a, u)/e(w, w_a, u) = 1$$

$$\beta \frac{w}{(\frac{w-x}{x})^{\beta}} (\frac{w-x}{x})^{\beta-1} (\frac{1}{x}) = 1$$

$$\beta \frac{w}{w-x} = 1$$

$$w = \frac{1}{\beta}(w-x)$$

$$w - \frac{1}{\beta}(w) = \frac{-x}{\beta}$$

$$w(\frac{\beta-1}{\beta}) = \frac{-x}{\beta}$$

$$w = \frac{x}{1-\beta} = \frac{(1-bu)w_a}{1-\beta}$$

$\therefore$ when β is small, firms offer a premium of approximately fraction β over the index of labor-market opportunities x. As the unemployment rate decreases, wages should fall. In equilibrium $w_a = w$:

$$w_a(1-\beta) = (1-bu)w_a$$

$$\frac{\beta}{b} = u$$

Meaning, the equilibrium unemployment rate depends only on the parameters of the effort function.

If a firm simply sets its wages to the prevailing market rate, w_a, the firm has effective cost per unit of labor, w_a/e, of:

$$C_{w*} = \frac{w_a}{e(w, w_a, u)}$$

$$C_{w*} = \frac{w_a}{(\frac{w_a - x}{x})^\beta}$$

$$C_{w*} = \frac{w_a}{(\frac{w_a - (1-bu)w_a}{(1-bu)w_a})^\beta}$$

$$C_{w*} = \frac{w_a}{(\frac{bu}{(1-bu)})^\beta}$$

$$C_{w*} = w_a(\frac{1-bu}{bu})^\beta$$

If a firm sets its own wage, the effective cost per unit of labor becomes:

$$C_w = \frac{w}{e(w, w_a, u)}$$

$$C_w = \frac{x/(1-\beta)}{(\frac{x/(1-\beta) - x}{x})^\beta}$$

$$C_w = x/(1-\beta)((1-\beta)/\beta)^\beta$$

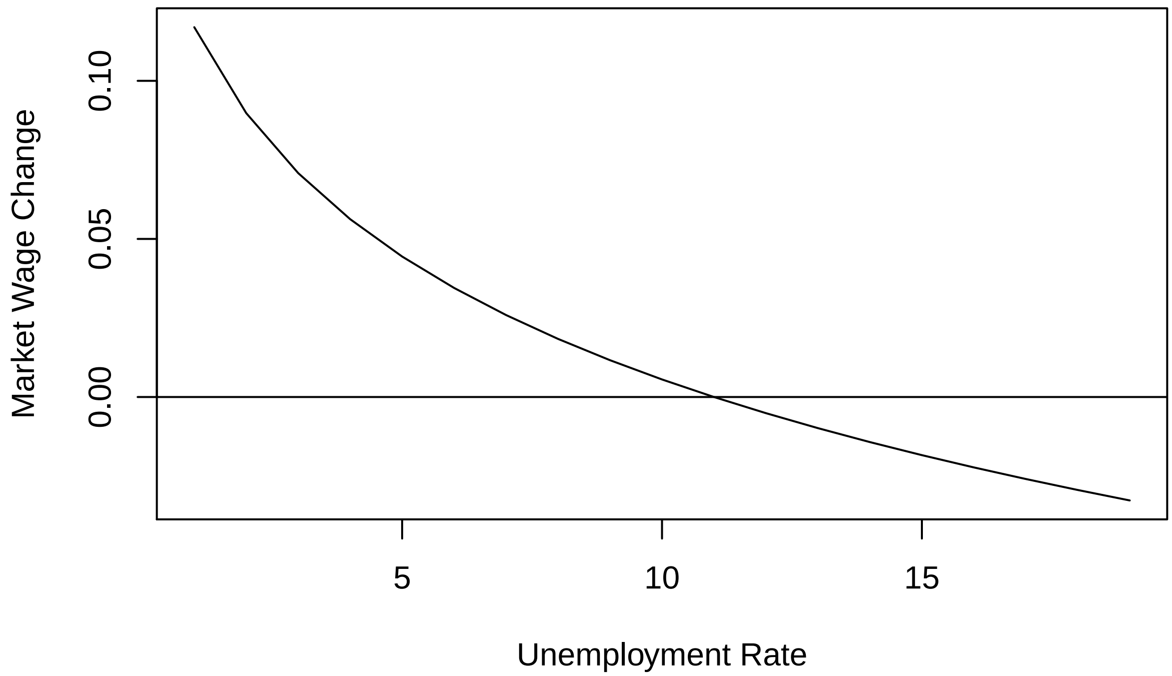
$$C_w = (1-bu)w_a/(1-\beta)^{\beta-1}(1/\beta)^\beta$$

We can use these two derivations to analyze the expected gain for a firm if they decide to cut costs. Suppose $\beta = 0.06$ and $b = 1$, this implies an unemployment rate of 6%. If the unemployment rate increases to 9% and other firms choose not to increase wages, we can analyze the gains of firms that stick to prevailing market wages versus those that set their own wages:

```
marketWage <- function(mkt, b, beta, newUnemployment) {
    unemployment <- beta/b
    # print(unemployment)
    previousMkt <- mkt * ((1 - b * unemployment)/(b *
        unemployment))^(beta)
    newMkt <- mkt * ((1 - b * newUnemployment)/(b *
        newUnemployment))^(beta)
    return((newMkt - previousMkt)/previousMkt)
}
marketWage(1, 1, 0.06, 0.09)  #Firms that use market wages have cost savings of 2.6%
```

```
## [1] -0.02593186
```

```
plot(marketWage(1, 1, 0.06, seq(0.01, 0.1, by = 0.005)),
    typ = "l", xlab = "Unemployment Rate", ylab = "Market Wage Change")
abline(h = 0)
```




```
firmWage <- function(mkt, b, beta, newUnemployment) {
    unemployment <- beta/b
    # print(unemployment)
    previousMkt <- mkt * (1 - b * unemployment) *
        (1/(beta^beta)) * 1/((1 - beta)^(1 - beta))
    newMkt <- mkt * (1 - b * newUnemployment) *
        (1/(beta^beta)) * 1/((1 - beta)^(1 - beta))
    return((newMkt - previousMkt)/previousMkt)
}
firmWage(1, 1, 0.06, 0.09)  #Firms that set their own wages have cost savings of 3.2%
```

```
## [1] -0.03191489
```

```
# Gain from setting own wage over the market
# rate is 0.6%. Worth it with all of the
# negative consequences? Probably not.
marketWage(1, 1, 0.06, 0.09) - firmWage(1, 1,
    0.06, 0.09)
```

```
## [1] 0.00598303
```

```
plot(seq(0.01, 0.1, by = 0.005), marketWage(1,
    1, 0.06, seq(0.01, 0.1, by = 0.005)), typ = "l",
```

```r
    xlab = "Unemployment Rate", ylab = "Wage Change",
    ylim = c(-0.05, 0.1))
lines(seq(0.01, 0.1, by = 0.005), firmWage(1,
    1, 0.06, seq(0.01, 0.1, by = 0.005)), col = 2)
abline(h = 0)
```