Obtaining Joint Distribution from Marginal with Rectilinear Restrictions

Fan Wang

2020-06-26

Contents

1 Obtaining Joint Distribution from Marginal with Rectilinear Restrictions

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

Suppose we want to know the joint probability mass function $P(E, A)$ where E and A are both discrete. E could be education groups, and A could be age groups. But we only know $P(E)$ and $P(A)$, under what conditions can we obtain the joint distribution from the marginals?

1.1 Unrestricted Joint 2 by 2 Distribution

Suppose there are two unique states for E and A. For example, suppose we know the unemployment probability for better or worse educated, and also for low and high age groups. We want to know the joint probability of been both better educated and lower age, better educated and higher age, worse educated and lower age, and worse educated and higher age. Then:

$$P(E_1) = P(A_1, E_1) + P(A_2, E_1)$$
$$P(E_2) = P(A_1, E_2) + P(A_2, E_2)$$
$$P(A_1) = P(A_1, E_1) + P(A_1, E_2)$$
$$P(A_2) = P(A_2, E_1) + P(A_2, E_2)$$

We know $P(E_1)$, $P(E_2)$, $P(A_1)$ and $P(A_2)$, but not $P(A_i, E_i)$. Let X, W, Y, Z be unknowns and A, B, C, D are known. It might seem like that with four equations and four unknowns, we can find X, W, Y, Z. But because what we know are probabilities: the marginals along each dimension sums up to 1. Without restrictions, there are no unique solutions to this problem. but many possible solutions. For example, Suppose $P(E_1) = 0.5$ and $P(A_1) = 0.5$, rewrite the above problem as:

$$0.5 = X + W$$
$$0.5 = Y + Z$$
$$0.5 = W + Y$$
$$0.5 = X + Z$$

Solutions include:

- $X = 0.2$, $W = 0.3$, $Y = 0.2$, $Z = 0.3$
- $X = 0.4$, $W = 0.1$, $Y = 0.4$, $Z = 0.1$
- and infinitely many others . . .

There are no unique solutions, because, when we write the linear system above in matrix form as shown below, the A coefficient matrix is not full rank, but has a rank of 3.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} W \\ X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}$$
$$A \cdot \mathbb{X} = b$$

We can see the rank of a matrix with the qr function (QR decomposition):

```
# Construct The coefficent Matrix
mt_a = t(matrix(data=c(1, 1, 0, 0,
                       0, 0, 1, 1,
                       1, 0, 1, 0,
                       0, 1, 0, 1), nrow=4, ncol=4))
# rank Check with the qr function:
print(qr(mt_a))
```

```
## $qr
##              [,1]        [,2]        [,3]          [,4]
## [1,] -1.4142136 -0.7071068 -0.7071068  0.000000e+00
## [2,]  0.0000000 -1.2247449  0.4082483 -8.164966e-01
## [3,]  0.7071068 -0.5773503 -1.1547005 -1.154701e+00
## [4,]  0.0000000  0.8164966 -0.4184316 -2.775558e-16
##
## $rank
## [1] 3
##
## $qraux
## [1] 1.707107e+00 1.000000e+00 1.908248e+00 2.775558e-16
##
## $pivot
## [1] 1 2 3 4
##
## attr(,"class")
## [1] "qr"
```

1.2 Rectilinear Restriction on Joint 2 by 2 Distribution

So in the section above, it is demonstrated that it is not possible to uniquely identify the joint probability mass function from marginal probability mass functions.

However, sometimes, we need to find some reasonable joint distribution, when we only observe marginal distributions. This joint distribution might be an input transition matrix in a model we simulate. If we just use one of the infinitely possible joint mass that match up with the marginals, then the model would have infinitely many simulation results depending on our arbitrary choice of joint mass.

Ideally, one should try to obtain data to estimate the underlying joint distribution, when this is not possible, we can impose additional non-parametric restrictions on the structures of the joint probability mass that would lead to unique joint mass from marginals.

Specifically, I will assume the incremental changes across rows and across columns of the joint mass matrix are row or column specific, is this sufficient? (In Some Cases it Will Not be):

$$\Delta_{12}^E = P(A_1, E_2) - P(A_1, E_1) = P(A_2, E_2) - P(A_2, E_1)$$
$$\Delta_{12}^A = P(A_2, E_1) - P(A_1, E_1) = P(A_2, E_2) - P(A_1, E_2)$$

The assumption is non-parametric. This is effectively an rectilinear assumption on the joint Cumulative Probability Mass Function.

Given this assumption, now we have:

$$P(E_2) - P(E_1) = P(A_1, E_2) + P(A_2, E_2) - P(A_1, E_1) - P(A_2, E_1)$$
$$P(A_2) - P(A_1) = P(A_2, E_1) + P(A_2, E_2) - P(A_1, E_1) - P(A_1, E_2)$$

Which become:
$$P(E_2) - P(E_1) = 2 \cdot \Delta_{12}^E$$
$$P(A_2) - P(A_1) = 2 \cdot \Delta_{12}^A$$

Suppose $P(E_1) = 0.5$ and $P(A_1) = 0.5$:

- $\phi = 0$
- $\rho = 0$
- hence: $P(A_1, E_1) = P(A_1, E_2) = P(A_2, E_1) = P(A_2, E_2) = 0.25$

Suppose $P(E_1) = 0.4$ and $P(A_1) = 0.7$:

- $\Delta_{12}^E = 0.1$
- $\Delta_{12}^A = -0.20$

Hence:

$$0.4 = P(A_1, E_1) + P(A_2, E_1) = P(A_1, E_1) + P(A_1, E_1) + \Delta_{12}^A = 2 \cdot P(A_1, E_1) - 0.20$$

And:

$$P(A_1, E_1) = \frac{0.60}{2} = 0.30$$
$$P(A_2, E_1) = 0.4 - 0.30 = 0.10$$
$$P(A_1, E_2) = 0.7 - 0.30 = 0.40$$
$$P(A_2, E_2) = 0.3 - 0.10 = 0.20$$

These joint mass sum up to 1, satisfy the marginal mass requirements from the data, and are unique given the rectilinear assumption.

1.3 Rectilinear Restriction Diamond on Joint 2 by 2 Distribution

The rectilinear assumptions, however, do not necessarily lead to positive values for each element of the joint mass function. In this section, I discuss under what conditions the rectilinear restriction leads to positive mass at all points of the joint probability mass function.

We can write these:

$$P(A_1, E_1) = \frac{1}{4} \left(1 - \Delta_{12}^E \cdot 2 - \Delta_{12}^A \cdot 2 \right)$$
$$P(A_2, E_1) = P(A_1, E_1) + \Delta_{12}^A$$
$$P(A_1, E_2) = P(A_1, E_1) + \Delta_{12}^E$$
$$P(A_2, E_2) = P(A_1, E_1) + \Delta_{12}^A + \Delta_{12}^E$$

Plugging the Values for $P(A_1, E_1)$ in, we have:

$$P(A_1, E_1) = \frac{1}{4} \left(1 - \Delta_{12}^E \cdot 2 - \Delta_{12}^A \cdot 2 \right)$$
$$P(A_2, E_1) = \frac{1}{4} \left(1 - \Delta_{12}^E \cdot 2 + \Delta_{12}^A \cdot 2 \right)$$
$$P(A_1, E_2) = \frac{1}{4} \left(1 + \Delta_{12}^E \cdot 2 - \Delta_{12}^A \cdot 2 \right)$$
$$P(A_2, E_2) = \frac{1}{4} \left(1 + \Delta_{12}^E \cdot 2 + \Delta_{12}^A \cdot 2 \right)$$

When are these terms positive:

$$P(A_1, E_1) \geq 0 \text{ iff } \frac{1}{2} \geq \Delta_{12}^E + \Delta_{12}^A$$
$$P(A_2, E_1) \geq 0 \text{ iff } \frac{1}{2} \geq \Delta_{12}^E - \Delta_{12}^A$$
$$P(A_1, E_2) \geq 0 \text{ iff } \frac{1}{2} \geq -\Delta_{12}^E + \Delta_{12}^A$$
$$P(A_2, E_2) \geq 0 \text{ iff } \frac{1}{2} \geq -\Delta_{12}^E - \Delta_{12}^A$$

Rewriting the positivity conditions, we have:

$$\Delta_{12}^E \leq \frac{1}{2} - \Delta_{12}^A$$
$$\Delta_{12}^E \leq \frac{1}{2} + \Delta_{12}^A$$
$$\Delta_{12}^E \geq -\frac{1}{2} + \Delta_{12}^A$$
$$\Delta_{12}^E \geq -\frac{1}{2} - \Delta_{12}^A$$

The four conditions above create a diamond, Following the rectilinear restriction, if the δ^E and δ^A values fall within the diamond region, then the mass for at all joint probability points will be positive. Basically the restriction is that when the jumps in mass are more extreme, rectilinear restriction will not return positive mass at all points. The requirement is that:

$$\mid \Delta_{12}^A \mid + \mid \Delta_{12}^E \mid \leq \frac{1}{2}$$

Graphically:

```r
# Labeling
st_title <- paste0('2 by 2 Joint Mass from Marginal Rectilinear Assumption\n',
                   'Intersecting Area Positive Mass at all Joint Discrete Points\n',
                   'x-axis and y-axis values will never exceed -0.5 or 0.5')
st_subtitle <- paste0('https://fanwangecon.github.io/',
                      'R4Econ/math/discrandvar/htmlpdfr/fs_drm_mass.html')
st_x_label <- 'delta A'
st_y_label <- 'delta E'

# Line 1
x1 <- seq(0, 0.5, length.out=50)
y1 <- 0.5-x1
st_legend1 <- 'Below This Line\n P_A1_E1>0 Restriction'
# Line 2
x2 <- seq(-0.5, 0, length.out=50)
y2 <- 0.5+x2
st_legend2 <- 'Below This Line\n P_A2_E1>0 Restriction'
# Line 3
x3 <- seq(0, 0.5, length.out=50)
y3 <- -0.5+x3
st_legend3 <- 'Above This Line\n P_A1_E2>0 Restriction'
# Line 4
x4 <- seq(-0.5, 0, length.out=50)
y4 <- -0.5-x4
st_legend4 <- 'Above This Line\n P_A1_E2>0 Restriction'

# line lty
st_line_1_lty <- 'solid'
st_line_2_lty <- 'dashed'
st_line_3_lty <- 'dotted'
st_line_4_lty <- 'dotdash'

# Share xlim and ylim
ar_xlim = c(-0.75, 0.75)
ar_ylim = c(-0.75, 0.75)

# Graph
par(new=FALSE, mar=c(5, 4, 4, 10))
plot(x1, y1, type="l", col = 'black', lwd = 2.5, lty = st_line_1_lty,
     xlim = ar_xlim, ylim = ar_ylim,
     ylab = '', xlab = '', yaxt='n', xaxt='n', ann=FALSE)
par(new=T)
plot(x2, y2, type="l", col = 'black', lwd = 2.5, lty = st_line_2_lty,
     xlim = ar_xlim, ylim = ar_ylim,
     ylab = '', xlab = '', yaxt='n', xaxt='n', ann=FALSE)
par(new=T)
plot(x3, y3, type="l", col = 'black', lwd = 2.5, lty = st_line_3_lty,
     xlim = ar_xlim, ylim = ar_ylim,
     ylab = '', xlab = '', yaxt='n', xaxt='n', ann=FALSE)
par(new=T)
plot(x4, y4, type="l", col = 'black', lwd = 2.5, lty = st_line_4_lty,
     xlim = ar_xlim, ylim = ar_ylim,
     ylab = '', xlab = '', yaxt='n', xaxt='n', ann=FALSE)
```

```r
# CEX sizing Contorl Titling and Legend Sizes
fl_ces_fig_reg = 1
fl_ces_fig_small = 0.75

# R Legend
title(main = st_title, sub = st_subtitle, xlab = st_x_label, ylab = st_y_label,
      cex.lab=fl_ces_fig_reg,
      cex.main=fl_ces_fig_reg,
      cex.sub=fl_ces_fig_small)
axis(1, cex.axis=fl_ces_fig_reg)
axis(2, cex.axis=fl_ces_fig_reg)
grid()

# Legend sizing CEX
legend("topright",
       inset=c(-0.4,0),
       xpd=TRUE,
       c(st_legend1, st_legend2, st_legend3, st_legend4),
       cex = fl_ces_fig_small,
       lty = c(st_line_1_lty, st_line_2_lty, st_line_3_lty, st_line_4_lty),
       title = 'Legends',
       y.intersp=2)
```


2 by 2 Joint Mass from Marginal Rectilinear Assumption
Intersecting Area Positive Mass at all Joint Discrete Points
x−axis and y−axis values will never exceed −0.5 or 0.5
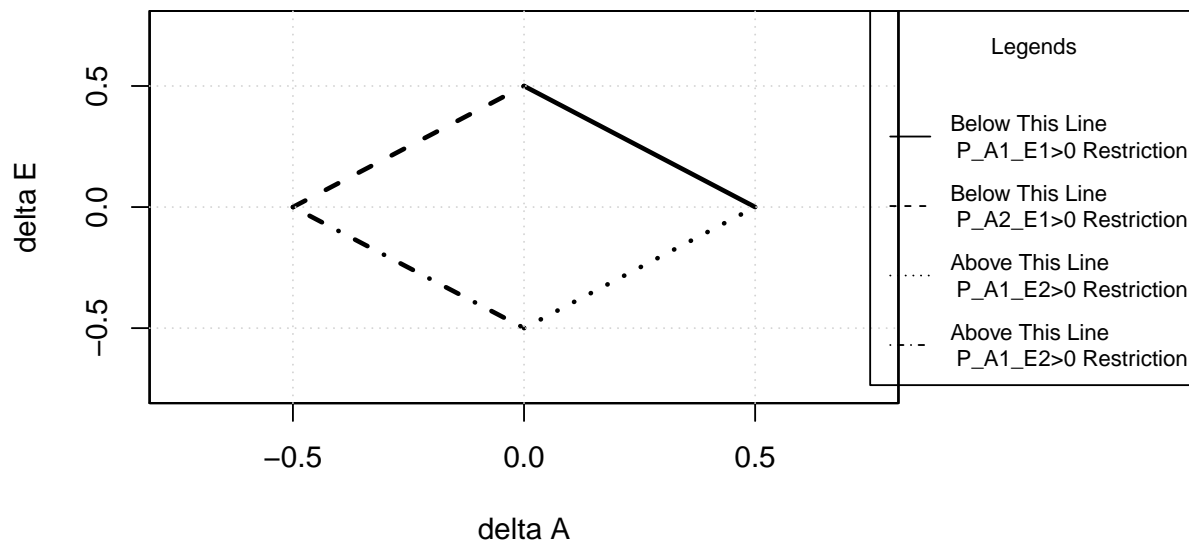
https://fanwangecon.github.io/R4Econ/math/discrandvar/htmlpdfr/fs_drm_mass.html

Programmatically, With different random values, we have:

```r
it_warning = 0
it_neg = 0
for (it_rand_seed in 1:1000) {
  # Generate two marginal MASS
  set.seed(it_rand_seed)
```

```r
    ar_E_marginal <- runif(2)
    # ar_E_marginal <- c(0.01, 0.99)
    # ar_E_marginal <- c(0.49, 0.51)
    ar_E_marginal <- ar_E_marginal/sum(ar_E_marginal)
    ar_A_marginal <- runif(2)
    # ar_A_marginal <- c(0.01, 0.99)
    # ar_A_marginal <- c(0.01, 0.99)
    ar_A_marginal <- ar_A_marginal/sum(ar_A_marginal)
    # print(ar_E_marginal)
    # print(ar_A_marginal)
    # Differences in Marginal Points
    ar_delta_E = diff(ar_E_marginal)/2
    ar_delta_A = diff(ar_A_marginal)/2
    # print(paste0('deltaE + deltaA:', diff(ar_E_marginal) + diff(ar_A_marginal)))
    # some cell negativity condition:
    if (sum(abs(diff(ar_E_marginal))) + sum(abs(diff(ar_A_marginal))) > 1){
      it_warning = it_warning + 1
      # warning('Outside of Diamond, Rectilinear Restriction Leads to Negative Values in Some Cells\n')
    }
    # What is P(A1,E1), implemetning the formula above
    fl_P_A1_E1 = (1 - c(1,1) %*% rbind(ar_delta_E, ar_delta_A) %*% t(t(c(2))))/(4)
    # Getting the Entire P_A_E matrix
    mt_P_A_E = matrix(data=NA, nrow=2, ncol=2)
    for (it_row in 1:length(ar_E_marginal)){
      for (it_col in 1:length(ar_A_marginal)){
        fl_p_value = fl_P_A1_E1
        if (it_row >= 2){
          fl_p_value = fl_p_value + sum(ar_delta_E[1:(it_row-1)])
        }
        if (it_col >= 2){
          fl_p_value = fl_p_value + sum(ar_delta_A[1:(it_col-1)])
        }
        mt_P_A_E[it_row, it_col] = fl_p_value
      }
    }
    # print(mt_P_A_E)
    sum(mt_P_A_E)
    rowSums(mt_P_A_E)
    colSums(mt_P_A_E)
    if (length(mt_P_A_E[mt_P_A_E<0])>0){
        it_neg = it_neg + 1
    }
  }
}
print(paste0('it_warning:',it_warning))
```

```
## [1] "it_warning:283"
```

```r
print(paste0('it_neg:',it_neg))
```

```
## [1] "it_neg:283"
```

1.4 Restricted Joint 3 by 3 Distribution

The idea can be applied to when there are three discrete random outcomes along each dimension. Find an unique 3 by 3 probability joint mass distribution from marginal distributions. Similar to the 2 by 2 case,

only when marginal mass changes are within a change diamond will this method lead to positive mass at all points of the joint distribution.

Given this assumption:

$$\Delta_{12}^E = P(A_1, E_2) - P(A_1, E_1) = P(A_2, E_2) - P(A_2, E_1) = P(A_3, E_2) - P(A_3, E_1)$$
$$\Delta_{23}^E = P(A_1, E_3) - P(A_1, E_2) = P(A_2, E_3) - P(A_2, E_2) = P(A_3, E_3) - P(A_3, E_2)$$
$$\Delta_{12}^A = P(A_2, E_1) - P(A_1, E_1) = P(A_2, E_2) - P(A_1, E_2) = P(A_2, E_3) - P(A_1, E_3)$$
$$\Delta_{23}^A = P(A_3, E_1) - P(A_2, E_1) = P(A_3, E_2) - P(A_2, E_2) = P(A_3, E_3) - P(A_2, E_3)$$

Following the two by two example, the restriction above just means we can use the differences between the marginal distribution's discrete points to back out.

$$\Delta_{12}^E = \frac{P(A_2) - P(A_1)}{3}$$
$$\Delta_{23}^E = \frac{P(A_3) - P(A_2)}{3}$$
$$\Delta_{12}^A = \frac{P(E_2) - P(E_1)}{3}$$
$$\Delta_{23}^A = \frac{P(E_3) - P(E_2)}{3}$$

Given these Δ values, we can solve for (A_1, E_1):

$$1 = 3 \cdot 3 \cdot P(A_1, E_1) + \Delta_{12}^E \cdot 3 \cdot (3-1) + \Delta_{23}^E \cdot 3 + \Delta_{12}^A \cdot 3 \cdot (3-1) + \Delta_{12}^A \cdot 3$$
$$P(A_1, E_1) = \frac{1}{9} \left(1 - \Delta_{12}^E \cdot 3 \cdot (3-1) - \Delta_{23}^E \cdot 3 - \Delta_{12}^A \cdot 3 \cdot (3-1) - \Delta_{12}^A \cdot 3 \right)$$

In Matrix form:

$$P(A_1, E_1) = \frac{1}{3 \cdot 3} \left(1 - \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta_{12}^E & \Delta_{23}^E \\ \Delta_{12}^A & \Delta_{23}^A \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot (3-1) \\ 3 \end{bmatrix} \right)$$

Following the 2 by 2 case, the condition needed for positive mass at all points is:

$$| \Delta_{12}^A | + | \Delta_{23}^A | + | \Delta_{12}^E | + | \Delta_{23}^E | \leq \frac{1}{3}$$

Implementing the formulas, we have:

```
# Generate two marginal MASS
it_warning = 0
it_neg = 0
it_concur = 0
for (it_rand_seed in 1:1000) {
  set.seed(it_rand_seed)
  # set.seed(333)
  ar_E_marginal <- runif(3)
  ar_E_marginal <- ar_E_marginal/sum(ar_E_marginal)
  ar_A_marginal <- runif(3)
  ar_A_marginal <- ar_A_marginal/sum(ar_A_marginal)
  # print(ar_E_marginal)
  # print(ar_A_marginal)
  # Differences in Marginal Points
```

```r
    ar_delta_E_m = diff(ar_E_marginal)
    ar_delta_A_m = diff(ar_A_marginal)
    ar_delta_E = diff(ar_E_marginal)/3
    ar_delta_A = diff(ar_A_marginal)/3
    # some cell negativity condition: this condition is incorrect
    bl_count_warn = FALSE
    for (it_row in 1:length(ar_delta_E)){
      for (it_col in 1:length(ar_delta_A)){
        if ((abs(sum(ar_delta_E_m[1:it_row])) + abs(sum(ar_delta_A_m[1:it_col]))) > 2/4) {
          bl_count_warn = TRUE
        }
        if ((abs(ar_delta_E_m[it_row]) + abs(ar_delta_A_m[it_col])) > 2/4) {
          bl_count_warn = TRUE
        }
      }
    }
    if (bl_count_warn) {
      # if (max(abs(diff(ar_E_marginal))) + max(abs(diff(ar_A_marginal))) > 2/3){
      it_warning = it_warning + 1
      # }
      # warning('Outside of Diamond, Rectilinear Restriction Leads to Negative Values in Some Cells\n')
    }
    # What is P(A1,E1), implemetning the formula above
    fl_P_A1_E1 = (1 - c(1,1) %*% rbind(ar_delta_E, ar_delta_A) %*% t(t(c(3*2, 3))))/(3*3)
    # Getting the Entire P_A_E matrix
    mt_P_A_E = matrix(data=NA, nrow=3, ncol=3)
    for (it_row in 1:length(ar_E_marginal)){
      for (it_col in 1:length(ar_A_marginal)){
        fl_p_value = fl_P_A1_E1
        if (it_row >= 2){
          fl_p_value = fl_p_value + sum(ar_delta_E[1:(it_row-1)])
        }
        if (it_col >= 2){
          fl_p_value = fl_p_value + sum(ar_delta_A[1:(it_col-1)])
        }
        mt_P_A_E[it_row, it_col] = fl_p_value
      }
    }
    # print(mt_P_A_E)
    sum(mt_P_A_E)
    rowSums(mt_P_A_E)
    colSums(mt_P_A_E)
    if (length(mt_P_A_E[mt_P_A_E<0])>0){
        it_neg = it_neg + 1
        if (bl_count_warn) {
          it_concur = it_concur + 1
        }
    }
  }
}
print(paste0('it_warning:',it_warning))

## [1] "it_warning:797"
```

```
print(paste0('it_neg:',it_neg))
```

```
## [1] "it_neg:592"
```

```
print(paste0('it_concur:',it_concur))
```

```
## [1] "it_concur:592"
```

1.5 Restricted Joint 5 by 5 Distribution

For a Five by Five Problem, we have:

In Matrix form:

$$P(A_1, E_1) = \frac{1}{5 \cdot 5} \left(1 - \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta_{12}^E & \Delta_{23}^E & \Delta_{34}^E & \Delta_{45}^E \\ \Delta_{45}^A & \Delta_{45}^A & \Delta_{45}^A & \Delta_{45}^A \end{bmatrix} \cdot \begin{bmatrix} 5 \cdot 4 \\ 5 \cdot 3 \\ 5 \cdot 2 \\ 5 \cdot 1 \end{bmatrix} \right)$$

```r
# pi_j=[0.22;0.175;0.16;0.165;0.22]; % Probability of unemployment in 2020 by age groups from Cajner et
# pi_w=[0.360;0.22;0.17;0.14;0.09]; % Probability of unemployment in 2020 by wage quintiles from Cajner
# Generate two marginal MASS
set.seed(111)
# set.seed(333)
ar_E_marginal <- c(0.22, 0.175, 0.16, 0.165, 0.22)
ar_E_marginal <- ar_E_marginal/sum(ar_E_marginal)
ar_A_marginal <- c(0.360, 0.22, 0.17, 0.14, 0.09)
ar_A_marginal <- ar_A_marginal/sum(ar_A_marginal)
print(ar_E_marginal)
```

```
## [1] 0.2340426 0.1861702 0.1702128 0.1755319 0.2340426
```

```r
print(ar_A_marginal)
```

```
## [1] 0.36734694 0.22448980 0.17346939 0.14285714 0.09183673
```

```r
# Differences in Marginal Points
ar_delta_E = diff(ar_E_marginal)/5
ar_delta_A = diff(ar_A_marginal)/5
# What is P(A1,E1), implemetning the formula above
fl_P_A1_E1 = (1 - c(1,1) %*% rbind(ar_delta_E, ar_delta_A) %*% t(t(c(20,15,10,5))))/(5*5)
# Getting the Entire P_A_E matrix
mt_P_A_E = matrix(data=NA, nrow=5, ncol=5)
for (it_row in 1:length(ar_E_marginal)){
  for (it_col in 1:length(ar_A_marginal)){
    fl_p_value = fl_P_A1_E1
    if (it_row >= 2){
      fl_p_value = fl_p_value + sum(ar_delta_E[1:(it_row-1)])
    }
    if (it_col >= 2){
      fl_p_value = fl_p_value + sum(ar_delta_A[1:(it_col-1)])
    }
    mt_P_A_E[it_row, it_col] = fl_p_value
  }
}
print(mt_P_A_E)
```

```
##              [,1]       [,2]       [,3]       [,4]       [,5]
## [1,] 0.08027790 0.05170647 0.04150239 0.03537994 0.02517586
## [2,] 0.07070343 0.04213200 0.03192792 0.02580547 0.01560139
## [3,] 0.06751194 0.03894051 0.02873643 0.02261398 0.01240990
## [4,] 0.06857577 0.04000434 0.02980026 0.02367781 0.01347373
## [5,] 0.08027790 0.05170647 0.04150239 0.03537994 0.02517586
```

sum(mt_P_A_E)

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

rowSums(mt_P_A_E)

```
## [1] 0.2340426 0.1861702 0.1702128 0.1755319 0.2340426
```

colSums(mt_P_A_E)

```
## [1] 0.36734694 0.22448980 0.17346939 0.14285714 0.09183673
```