# Empirical Analysis II Problem Set 2

Chase Abram, Jeanne Sorin[*]

July 5, 2020

## Problem 1

### I. Your first task

We are given that the matrix A encodes the movements of prices and quantities due to a positive one-standard deviation demand shock in the first column and due to a positive one-standard deviation supply shock in the second column

$$A = \begin{bmatrix} p_d & p_s \\ q_d & q_s \end{bmatrix}$$

Note that $A$ satisfies $\Sigma = AA'$. We parameterize these matrices $A = A(\mu, \Sigma)$ per

$$A(\mu, \Sigma) = L(\Sigma) \begin{bmatrix} \cos(\mu) & -\sin(\mu) \\ \sin(\mu) & \cos(\mu) \end{bmatrix}$$

for some $\mu \in [0, 2\pi)$ and where $L = L(\Sigma)$ the lower triangular matrix with positive diagonal elements satisfying $\Sigma = LL'$: **the Cholesky factorization**.

Suppose that finite data delivers an estimator of $\Sigma$

$$\hat{\Sigma} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$$

#### Let's compute the Cholesky factor $\hat{L}$

Setting

$$L = \begin{bmatrix} a & 0 \\ c & d \end{bmatrix}$$

[*] This solution set benefited greatly from the comments and points raised by Manav Chaudhary, George Votja, and Dan Ehrlich.

Because $LL' = \hat{\Sigma}$ we solve

$$\begin{bmatrix} a^2 & ac \\ ac & c^2 + d^2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$$

Which holds

$$\begin{aligned} a^2 = 1 &\Rightarrow a = 1 \text{ because by definition } a > 0 \\ ac = 2 &\Rightarrow c = 2 \\ c^2 + d^2 = 5 &\Rightarrow d = 1 \text{ because by definition } d > 0 \end{aligned}$$

Therefore

$$\hat{L} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

## II. Your second task

```
import numpy as np
import math
import matplotlib.pyplot as plt

# Initialize the environment
J = 1000
n = 20
mu_lb = np.array(())
mu_hb = np.array(())
pd_lb = np.array(())
pd_hb = np.array(())
qd_lb = np.array(())
qd_hb = np.array(())
as_lb = np.array(())
as_hb = np.array(())
ps_lb = np.array(())
ps_hb = np.array(())
qs_lb = np.array(())
qs_hb = np.array(())
ad_lb = np.array(())
ad_hb = np.array(())

mu_keep_all = np.array(())
pd_all = np.array(())
qd_all = np.array(())
as_all = np.array(())
ps_all = np.array(())
qs_all = np.array(())
```

```
ad_all = np.array(())

# Initialize the prior on L (and therefore on )
L = np.array([[1, 0], [2, 1]])

J=1000
for j in range(0,J):

    ### Task II
    # 1. Draw n = 20 independent vectors
    v = []
    _j = [[0,0],[0,0]]
    for i in range(0,n):
        # Draw 2 independent elements
        e = np.array((np.random.normal(0,1, 2)))
        # Multiply them by L
        vi = [L.dot(e)]
        # Stack all vi in a big vector
        v.append(vi[0])
        # Update the var-cov matrix
        _j = np.add(_j, (1/n)*vi[0][:, np.newaxis]*vi[0])

    #2. Calculate the Cholesky decomposition : i.e. the lower triangular matrix L
    L_j = np.linalg.cholesky(_j)

    #3. Draw mu_k for all k = 1,...,  K = 1000 from a uniform distribution on mu in [0
    K = 1000
    pi_2 = 2*math.pi
    mu = np.random.uniform(0,pi_2,K)

    mu_keep = np.array(())
    pd = np.array(())
    ps = np.array(())
    qs = np.array(())
    qd = np.array(())
    as = np.array(())
    ad = np.array(())
    for k in range(0,K):
        matrix_trig = [[math.cos(mu[k]), -math.sin(mu[k])],[math.sin(mu[k]), math.cos(mu
        A_k = np.dot(L_j, matrix_trig)
        #a. Check whether mu in S()
        if A_k[0,0] > 0 and A_k[0,1] < 0 and A_k[1,0] > 0 and A_k[1,1] > 0:

            # Keep the draw
            mu_keep = np.append(mu_keep, mu[k])
            # Keep the implied price and quantity response
            pd = np.append(pd, A_k[0,0])
            ps = np.append(ps, A_k[0,1])
            qd = np.append(qd, A_k[1,0])
            qs = np.append(qs, A_k[1,1])
            as = np.append(as, A_k[0,0]/A_k[1,0])
            ad = np.append(ad, A_k[0,1]/A_k[1,1])


    # 4. For a given j, calculate the min and max of the kept draws
    mu_lb = np.append(mu_lb, min(mu_keep))
    mu_hb = np.append(mu_hb, max(mu_keep))
    pd_lb = np.append(pd_lb, min(pd))
    pd_hb = np.append(pd_hb, max(pd))
    qd_lb = np.append(qd_lb, min(qd))
    qd_hb = np.append(qd_hb, max(qd))
    as_lb = np.append(as_lb, min(as))
    as_hb = np.append(as_hb, max(as))
    ps_lb = np.append(ps_lb, min(ps))
    ps_hb = np.append(ps_hb, max(ps))
    qs_lb = np.append(qs_lb, min(qs))
    qs_hb = np.append(qs_hb, max(qs))
    ad_lb = np.append(ad_lb, min(ad))
    ad_hb = np.append(ad_hb, max(ad))

    # Keep track of all kept draws in a long vector of size M = total number of kept d
    mu_keep_all = np.append(mu_keep_all, mu_keep).reshape(-1)
    pd_all = np.append(pd_all, pd).reshape(-1)
    qd_all = np.append(qd_all, qd).reshape(-1)
    as_all = np.append(as_all, as).reshape(-1)
    ps_all = np.append(ps_all, ps).reshape(-1)
    qs_all = np.append(qs_all, qs).reshape(-1)
    ad_all = np.append(ad_all, as).reshape(-1)
```

4. **For the cracks and extra points: there is also an analytical way of deriving these min and max from $\Sigma^j$, what would result for $K \to \infty$, ie one wouldn't need to do the $\mu$ sampling to calculate them.**

Let $\Sigma^{(j)}$ be positive definite[1], so that it has unique Cholesky decomposition $L$. Now we may consider

[1] There is probability zero of any given $\Sigma^{(j)}$ having a zero eigenvalue.

$$\begin{aligned} L\begin{bmatrix} \cos(\mu) & -\sin(\mu) \\ \sin(\mu) & \cos(\mu) \end{bmatrix} &= \begin{bmatrix} a & 0 \\ c & d \end{bmatrix}\begin{bmatrix} \cos(\mu) & -\sin(\mu) \\ \sin(\mu) & \cos(\mu) \end{bmatrix} \\ &= \begin{bmatrix} a\cos(\mu) & -a\sin(\mu) \\ c\cos(\mu) + d\sin(\mu) & -c\sin(\mu) + d\cos(\mu) \end{bmatrix} \end{aligned}$$

To determine the min and max, we may consider the intersection of the sets of $\mu$ which satisfy the sign restriction for each element, and from that set take the min and max (the calculations assume $c > 0$ to avoid the difficulties in the domain of arctan, but in the final answer below we take the domain into account by shift $\pi$ and $\frac{\pi}{2}$ accordingly).

$$\begin{aligned} a\cos(\mu) \geq 0 &\Rightarrow \mu \in [0, \frac{\pi}{2}] \cup [\frac{3\pi}{2}, 2\pi] \\ -a\sin(\mu) \leq 0 &\Rightarrow \mu \in [0, \pi] \\ c\cos(\mu) + d\sin(\mu) \geq 0 &\Rightarrow \mu \in [0, \arctan(-\frac{c}{d}) + \pi] \cup [\arctan(-\frac{c}{d}), 2\pi] \\ -c\sin(\mu) + d\cos(\mu) \geq 0 &\Rightarrow \mu \in [0, \arctan(\frac{d}{c})] \cup [\arctan(\frac{d}{c}) + \pi, 2\pi] \end{aligned}$$

Therefore the intersecting set is the following, with its bounds as the min and max.

$$\begin{cases} [0, \arctan(\frac{d}{c})] \quad c > 0 \\ [0, \frac{\pi}{2}] \quad c = 0 \\ [\frac{\pi}{2} + \arctan(\frac{d}{c}), \frac{\pi}{2}] \quad c < 0 \end{cases}$$

# III. Your third task

## 1. Point Identification Approach

```
h = 0.02

def f(z, zm, h):
    M = len(zm)
    fac = 1/(h*(2*math.pi)**0.5)
    div = (2*h)**2
    f = 0
    for m in range(0, len(zm)):
        ins = -(z-zm[m])**2
        add = fac*np.exp(ins/div)
        f = f + add
    f = f/M
    return(f)

ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
i = 0

def makeplot(data, title, nb=1000, h=0.02):
    x = np.linspace(min(data), min(max(data), 3), nb)
    fig = plt.figure()
    ax = fig.add_subplot(1,1,1)
    plt.plot(x, f(x,data,h), 'r', label="Posterior distribution")
    plt.axvline(x=np.mean(data), color="blue", label="mean")
    plt.axvline(x=np.median(data), color="orange", label="median")
    print("Posterior distribution for ", title)
    plt.legend()
    plt.show()
```

***Why is this the appropriate posterior distribution, when imposing a flat prior on $\mu \in [0, 2\pi)$? This is a simple version of a kernel density estimator. Why is this is indeed a probability density?***

The flat prior given (restated below for showing that it is a proper density) means that no observation is given more weight, a priori, than another. So when we observe a high $\mu$ or a low $\mu$, they play into the posterior (and posterior approximation) the same. This would not be the case, if, say our prior were a Beta$(\frac{1}{2}, \frac{1}{2})$ (horizontally scaled to account for the domain of $\mu$), because then our prior would have higher weights on observations closer to the boundaries. The fact that we are mixing normal densities is just a tidbit about estimation, and we certainly could use other densities for mixing, but what matters is that the $\frac{1}{M}$ term is applied equally to each observation.

Now let's prove it is a proper density

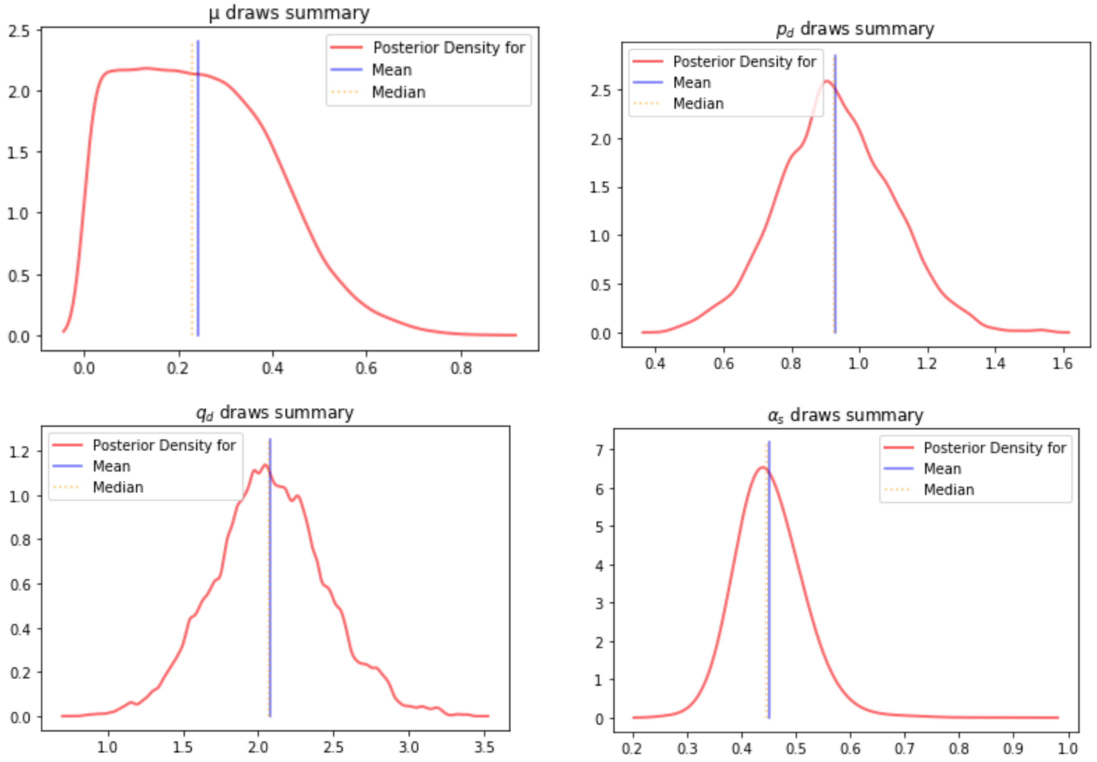
$$
\begin{aligned}
\int f(z)\mathrm{d}z &= \int \frac{1}{M}\sum_{m=1}^{M} \frac{1}{\sqrt{2\pi}h}\exp(-(z-z_m)^2/(2h)^2)\mathrm{d}z \\
&= \frac{1}{M}\sum_{m=1}^{M} \int \frac{1}{\sqrt{2\pi}h}\exp(-(z-z_m)^2/(2h)^2)\mathrm{d}z \\
& \qquad \text{(Interchanging } \textstyle\sum \text{ and } \int \text{ is legal because sum is finite)} \\
&= \frac{1}{M}\sum_{m=1}^{M} 1 \qquad\qquad \text{(Term was } \sim N(z_m, h)\text{)} \\
&= 1
\end{aligned}
$$

<u>State the means and medians</u>:

The mean of $\mu_{keep}$ is 0.246 and the median is 0.234.
The mean of $pd_{keep}$ is 0.924 and the median is 0.920.
The mean of $qd_{keep}$ is 2.057 and the median is 2.057.

The mean of $\alpha s_{keep}$ is 0.453 and the median is 0.449.

Figure 1: Posterior distributions for $\mu$, $p_d$, $q_d$, $\alpha_s$



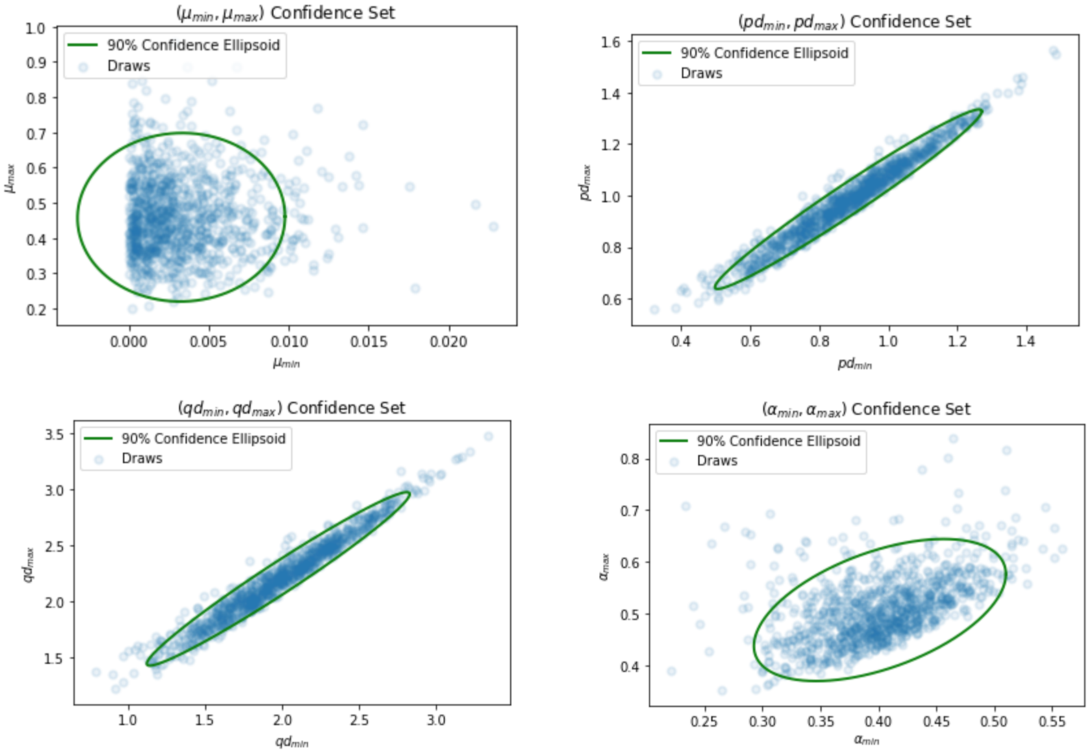
### 2. 3. Set Identification Approach

The mean of $\underline{\mu}^j$ is 0.0063, the mean of $\overline{\mu}^j$ is 0.4608.
The mean of $\underline{p_d}^j$ is 0.8834, The mean of $\overline{p_d}^j$ is 0.9861.
The mean of $\underline{q_d}^j$ is 1.9660, The mean of $\overline{q_d}^j$ is 2.0806.
The mean of $\underline{\alpha_s}^j$ is 0.4022, The mean of $\overline{\alpha_s}^j$ is 0.5090.

```
def plot_CI(x,y,x_label,y_label,p=0.9, legend_pos = 'upper left'):

    m0 = np.mean(x)
    m1 = np.mean(y)

    sigma = np.cov(x,y)
    L = np.linalg.cholesky(sigma)
    c_val = np.sqrt(chi2.ppf(p, 2))

    #unpack L matrix
    L_11, L_21, L_22 = L[0,0], L[1,0], L[1,1]
    t = np.linspace(0, 2*math.pi, num=1000)
```

```
x_val = m0 + c_val*L_11*np.cos(t)
y_val = m1 + c_val*(L_21*np.cos(t)+L_22*np.sin(t))

fig, ax = plt.subplots()
ax.plot(x_val, y_val, linewidth=3,alpha=0.5, color = 'green', label= '90% Confidence

mu = np.array([m0,m1])
draws = np.random.multivariate_normal(, sigma, 1000).T
x_draws, y_draws = draws[0], draws[1]

ax.scatter(x, y, linewidth=1,alpha=0.1, label = 'Random draws from bivariate normal'

ax.legend(loc=legend_pos,ncol=1)
ax.set_title(' Confidence Set')
plt.xlabel(x_label)
plt.ylabel(y_label)

plt.show()
```

Figure 2: Confidence Sets for $\mu$, $p_d$, $q_d$, $\alpha_s$



***For cracks and extra points***: *show that the set $S(\Sigma)$ is an interval $S(\Sigma) = [\mu_{low}, \mu_{high}]$*

Really, we already did this above with our method of intersecting sets to satisfy the sign restrictions element-wise. We there found that the intersection of the sets of $\mu$ which satisfied the sign restriction for each element was an interval, because the set for each interval was union of intervals, and by intersecting we were left with a single interval.

We can be a bit more mathematical about understanding why we find an interval an particular, though, and not, say, a countable dense set of points, like $\mathbb{Q}$. The reason is that each element of $A = LT$ (where $T$ is the matrix with trig functions) is a continuous function of $\mu$, so when we consider the pre-image of each element for $[0, \infty)$ or $(-\infty, 0]$, our set is closed, and in $\mathbb{R}$ this means our set is a union of closed intervals (where we allow a single point to be considered an interval, here). To actually be sure that we have a single interval, we must consider the trig functional forms, as above, to rule out sets by intersection until we are left with single set of $\mu$s which satisfy the sign restrictions (see above for the set descriptions using arctan).

4. (**For the cracks and extra points**)
***Why do you think I wanted you to investigate*** $p_D$,$q_D$ ***and*** $\alpha_s$ ***here? What about*** $p_S$, $q_S$ ***and*** $\alpha_D$***? How would your results change if***

$$\hat{\Sigma} = \begin{bmatrix} 1 & -2 \\ -2 & 5 \end{bmatrix}$$

***and why ?***

Remember that

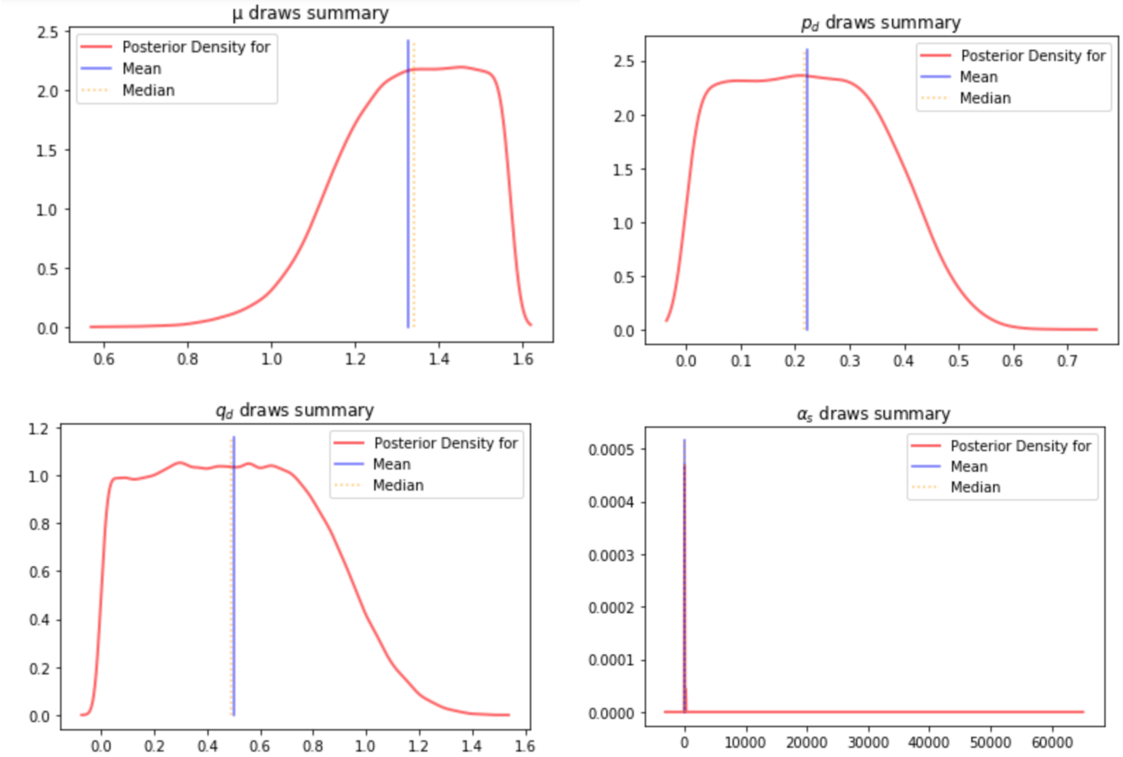
$$\Sigma = AA' = \begin{bmatrix} p_d^2 + p_s^2 & p_d q_d + p_s q_s \\ p_d q_d + p_s q_s & q_d^2 + q_s^2 \end{bmatrix}$$

In the original exercise, this implies:

$$2 = p_d q_d + p_s q_s$$

Assuming we are drawing independent shocks $\epsilon_{d,t}, \epsilon_{s,t}$ from $\mathcal{N}(0,1)$, the $\Sigma$ matrix resulting from our data reveals that on average (we are drawing $\epsilon_{d,t}, \epsilon_{s,t}$ from the same distribution), we observe a positive relationship between price and quantity movements in response to these shocks: demand shocks dominate, or at least are easier to parse out from the data.
Sufficient variations of demand (large enough $p_d, q_d$) allows us to estimate the elasticity of supply (remember how we define elasticities using shifts of the other side).
On the other hand, this $\Sigma$ is much less informative about $q_s, p_s, \alpha_d$ because shifts of the supply curve are much smaller than shifts of the demand curve. Not surprisingly, the posterior distributions for $q_s, p_s, \alpha_d$ are much flatter than for $q_d, p_d, \alpha_s$, as one can see on the graphs below.

Figure 3: Confidence Sets for $\mu$, $p_d$, $q_d$, $\alpha_s$ using the alternative $\Sigma$: flatter posteriors



In a nutshell, the story is opposite with the new $\Sigma' = \begin{bmatrix} 1 & -2 \\ -2 & 5 \end{bmatrix}$: the variance-covariance matrix reveals that the magnitude of the supply shocks dominates the magnitude of the demand shocks as $p_s q_s + p_d q_d = -2$. In this case, our simulated data (or equivalently actual data if we had them) allows us to improve our estimates of $p_d, q_d$ and therefore $\alpha_s$ much better than $p_s, q_s, \alpha_d$, whose posterior distributions are much flatter this time.

**5. (For the cracks and extra points)** *In order to investigate, how much the prior already determines the results, and how much we can learn in the limit, redo everything, where*

**a) (prior) the $\Sigma^{(j)}$ are drawn per drawing $v_i = \begin{bmatrix} v_{p,i} \\ v_{q,i} \end{bmatrix} \sim \mathcal{N}(0, \hat{I}_2)$ and setting n=2.**

Drawing from a distribution with the identity matrix as prior on the variance-covariance structure makes it more difficult to estimate our parameters of interest $p_d, q_d, \alpha_s$, as one can see on the graphs below.

Figure 4: Posterior distributions for $\mu$, $p_d$, $q_d$, $\alpha_s$ using the identity $\Sigma$

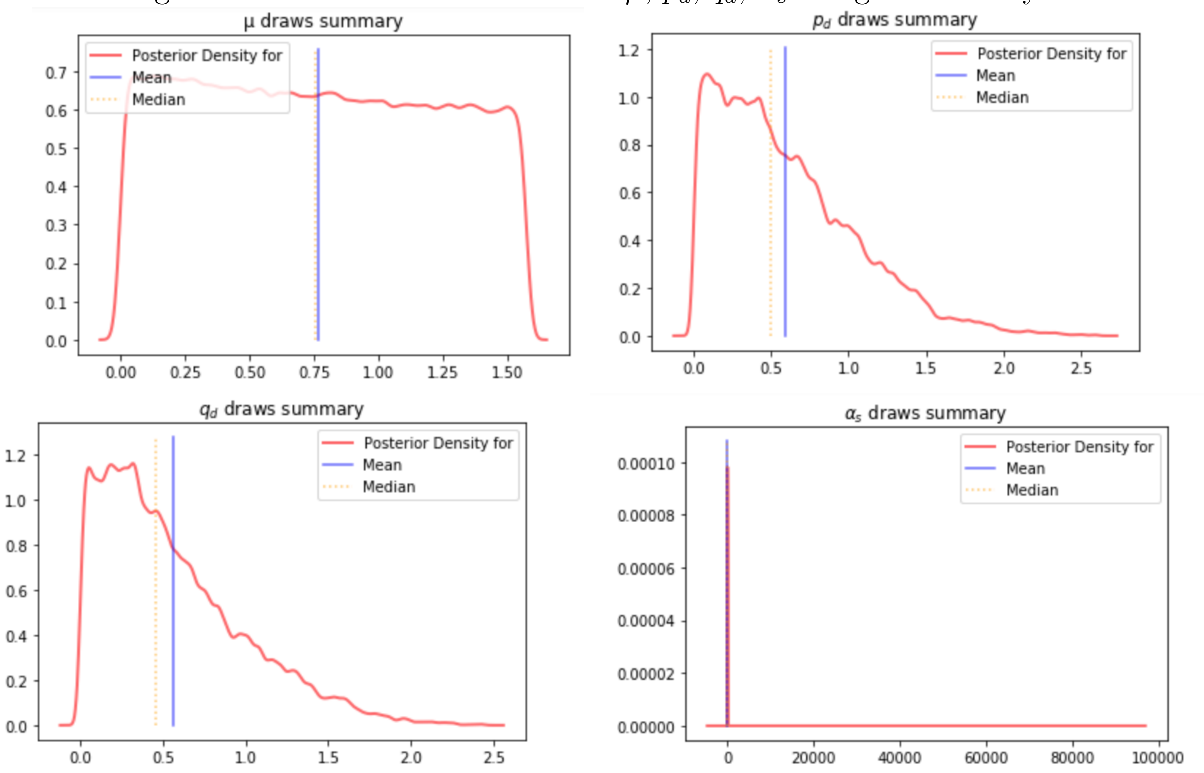
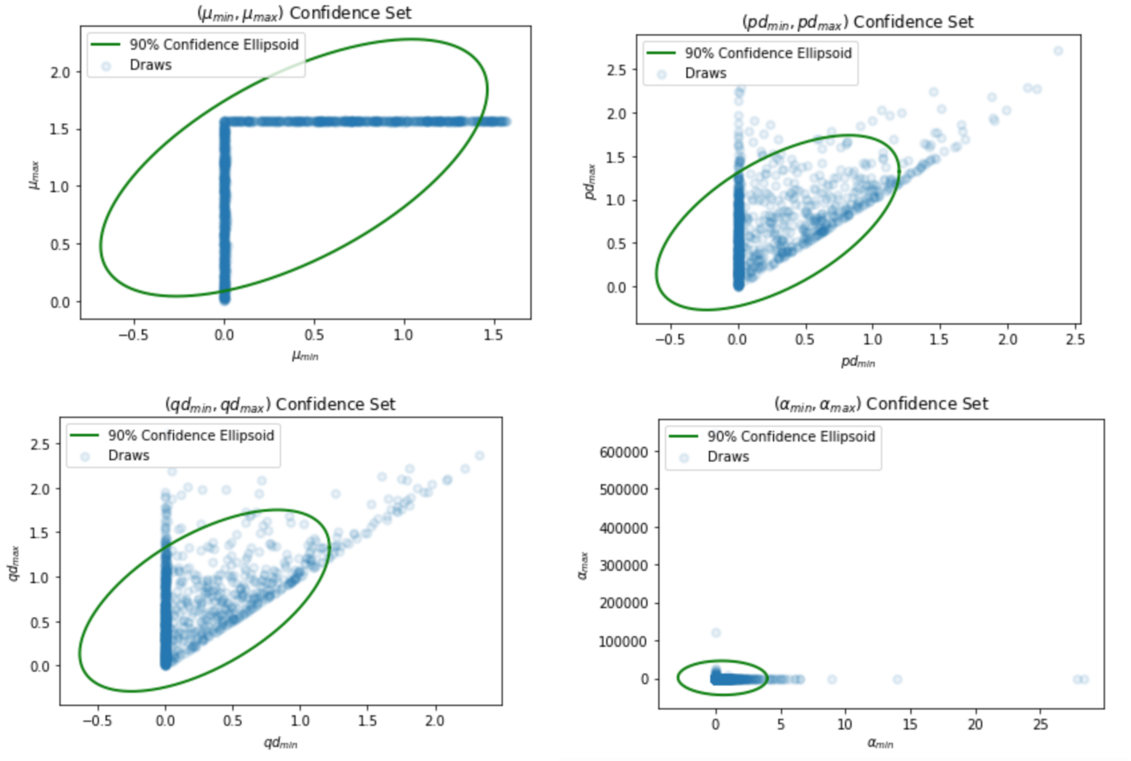


Figure 5: Confidence Sets for $\mu$, $p_d$, $q_d$, $\alpha_s$ using the identity $\Sigma$



The posterior distribution of $\mu$ is a uniform distribution on the whole domain of $\mu$ because the identity variance-covariance matrix imposes that $\varepsilon_{p,t}$ and $\varepsilon_{s,t}$ don't interact and enter price and quantity movements exactly the same way.
Posterior distributions for both $p_d$ and $q_d$ are skewed and put a high weight to values close to 0, which is consistent with the identity $\Sigma$ matrix not allowing us to categorize shocks and attribute price and quantity movements to demand or supply.

### b) (asymptotics) $\Sigma^j = \hat{\Sigma}$ always.

Figure 6: Posterior distributions for $\mu$, $p_d$, $q_d$, $\alpha_s$ when using the asymptotic $\Sigma$

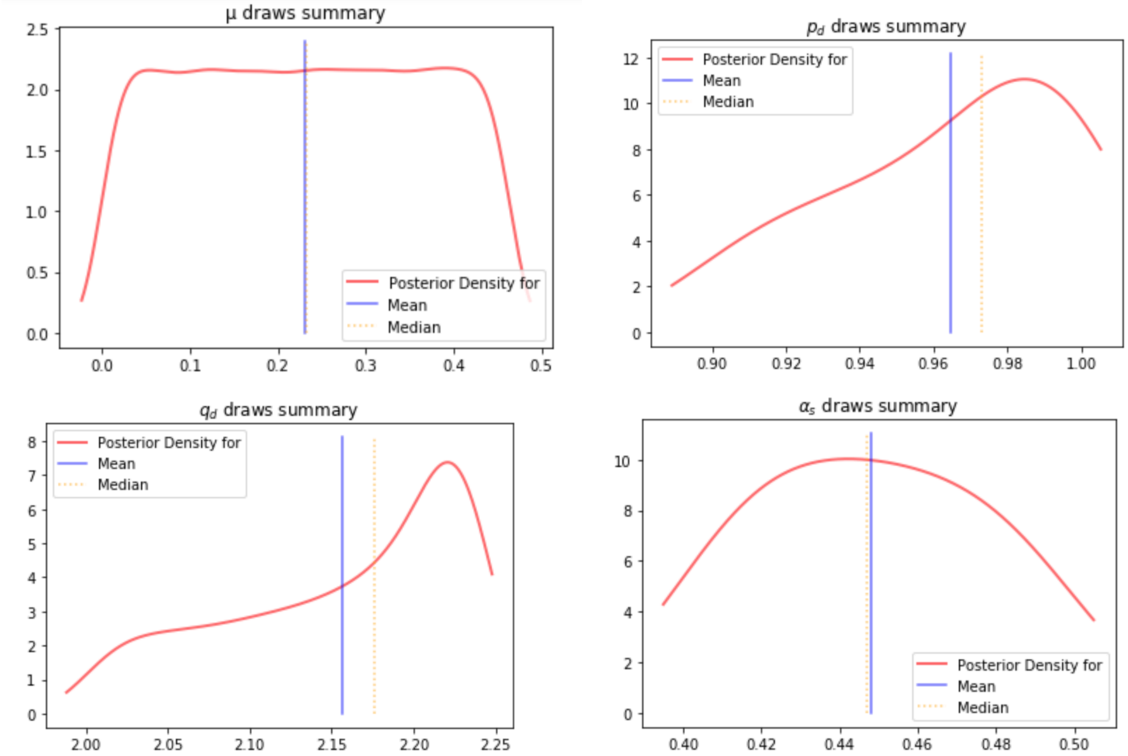
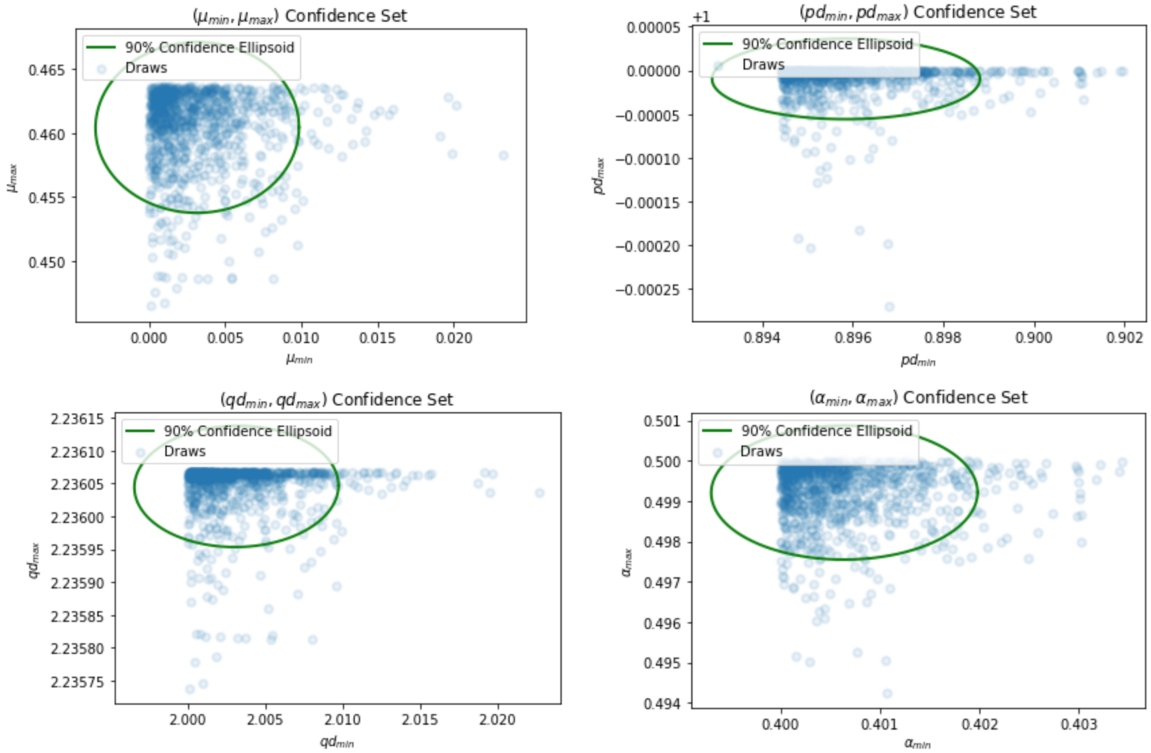


Figure 7: Confidence Sets for $\mu$, $p_d$, $q_d$, $\alpha_s$ using the asymptotic $\Sigma$



As we might expect the distribution for draws $\mu$ really is not that different from when we used the $\Sigma^{(j)}$, and the same idea holds somewhat for the other data types. The big point here is that, since we only have the true $^{(j)}$, the distributions are smoother, and perhaps slightly more disperse, since we have more data for the single $\Sigma$ than we did for any single $\Sigma^{(j)}$ above.

As for the confidence regions plots, the results look much sharper here, because of the lack of variance which above came from $\Sigma^{(j)}$. Here we can clearly see the "cutoff lines" in each plot, even without the confidence ellipsoids, and so we can see what is probably the "true" min and max values at the corner of these "cutoff lines".

**6. Comment on your findings.** We did a fair amount of commenting as we went!

Perhaps the "big idea" we got out here was that sign restrictions can be used as a way to generate/filter data so as to try to separate supply and demand shocks, and therefore also the slopes of the supply and demand curves. This question also (at the end in the cracks) emphasized how the structure of the covariance matrix, along with the sign restrictions, can lead us to believe that inference may be better by using shocks from the curve which will tend to generate more clear-cut changes in prices and quantities.

# Problem 2

### 1. State the log-likelihood function $l(\theta, x)$

The likelihood function $L(\theta, x)$ is

$$L(\theta, x) = \prod_{t=1}^{T} f(x_t|\theta) = \prod_{t=1}^{T} \theta^{x_t}(1-\theta)^{1-x_t}$$

Where $x_t = 1$ if observation t is a "head", $= 0$ if observation t is a "tails".
The log likelihood[2]:

$$\ell(\theta, x) = \frac{1}{T}\sum_{t=1}^{T}[x_t \log(\theta) + (1-x_t)\log(1-\theta)]$$

Where $T$ is the number of draws.

### 2. Show (or provide a good argument) that $\kappa$ is a sufficient statistic

A conclusion I have reached: taking care with notation in Bayesian analysis is essential. So, I will use $\kappa(x) = \frac{1}{T}\sum_{t=1}^{T} x_t$, to emphasize that $\kappa(x)$ denotes a function from the data to a statistic. I will use $\bar{\kappa}$ to denote a specific value in $\mathbb{R}$. Then $L(x \in A \mid \kappa(x) \in B, \theta \in C)$ indicates the likelihood that the data are in $A$, *given* that the statistic $\kappa(x) \in B$ and the parameters $\theta \in C$, whereas $L(x \in A, \kappa(x) \in B \mid \theta \in C)$ indicates the likelihood that the data $x \in A$ *and* the statistic $\kappa(x) \in B$, given the parameters $\theta \in C$.

With just the understanding of notation in the previous paragraph, the below calculation does go through, to show that the likelihood of data $x$, given $\kappa(x)$ and $\theta$, actually does not depend on $\theta$. We use the same proxy for the Dirac delta distribution as in the previous problem set.

$$\begin{aligned}
L(x = \bar{x} \mid \kappa(x) = \bar{\kappa}, \theta = \bar{\theta}) &= \frac{L(x = \bar{x}, \kappa(x) = \bar{\kappa} \mid \theta = \bar{\theta})}{L(\kappa(x) = \bar{\kappa} \mid \theta = \bar{\theta})} \\
&= \frac{\bar{\theta}^{T\kappa(x)}(1-\bar{\theta})^{T(1-\kappa(x))}\mathbf{1}_{\kappa(\bar{x})=\bar{\kappa}}}{\bar{\theta}^{T\bar{\kappa}}(1-\bar{\theta})^{T(1-\bar{\kappa})}} \\
&= \frac{\bar{\theta}^{T\bar{\kappa}}(1-\bar{\theta})^{T(1-\bar{\kappa})}\mathbf{1}_{\kappa(\bar{x})=\bar{\kappa}}}{\bar{\theta}^{T\bar{\kappa}}(1-\bar{\theta})^{T(1-\bar{\kappa})}} \\
&= \mathbf{1}_{\kappa(\bar{x})=\bar{\kappa}}
\end{aligned}$$

The interpretation here is that, if the data imply an average that is in the conditioned set, then the likelihood is constant (not dependent upon $\theta$). Otherwise the event cannot occur, since then the data contradict their statistic.

Now let's be honest: we played a bit fast and loose by just dividing fractions above, and we even had to use a stand-in for the Dirac distribution (it does not have a density proper),

[2] We scale by $\frac{1}{T}$ for convenience, and this is fine, by the likelihood principle, since the shape will be unaffected by positive constant scaling.

so we probably should have to check a bunch of measure theory stuff before the method above is legal. Luckily, the Fisher–Neyman Factorization Theorem is a simple alternative we can apply in this case: if the joint density $f(\theta, x)$ (here unconditional likelihood function) can be decomposed into functions $g$ and $h$ such that $f(\theta, x) = g(x)h(\theta, T(x))$, then $T$ is a sufficient statistic. In this case, let $f(x) = 1$ and $h(\theta, \kappa(x)) = \theta^{T\kappa(x)}(1-\theta)^{T(1-\kappa(x))}$. So we know[3] from Neyman-Pearson that $\kappa(x)$ is a sufficient statistic.

### 3. What is $E[\kappa]$, given $\theta$?

A direct argument:

$$\begin{aligned}
E[\kappa \mid \theta] &= E[\frac{1}{T}\sum_{t=1}^{T} x_t \mid \theta] \\
&= \frac{1}{T}\sum_{t=1}^{T} E[x_t \mid \theta] \\
&= E[x_1 \mid \theta] && (x_t \text{ iid}) \\
&= \theta \cdot 1 + (1-\theta)\cdot 0 && (\text{Def. of } x_t \mid \theta) \\
&= \theta
\end{aligned}$$

I think this is just getting at the frequentist style of thinking, where if $\theta$ is known, then the expected value of the mean is $\theta$. Note that this is a pretty one-dimensional (not whole distribution) statement, and we are conditioning on the parameter to look at the possible data, whereas Bayesian thinking typically emphasizes conditioning on the data to understand the distribution of the parameter.

### 4. Calculate the score, the MLE and the information matrix.

**MLE:**

$$\begin{aligned}
\ell(\theta, x) &= \frac{1}{T}\sum_{t=1}^{T}[x_t \log(\theta) + (1-x_t)\log(1-\theta)] \\
&= \log(\theta)\frac{1}{T}\sum_{t=1}^{T}(x_t) + \log(1-\theta)\frac{1}{T}\sum_{t=1}^{T}(1-x_t) \\
&= \log(\theta)\kappa + \log(1-\theta)(1-\kappa)
\end{aligned}$$

[3] In a definitely mathematically legal way.

Maximizing wrt $\theta$:

$$\begin{aligned}\frac{\partial \ell}{\partial \theta} &= 0\\ \Leftrightarrow \frac{\kappa(x)}{\theta} - \frac{1-\kappa(x)}{1-\theta} &= 0\\ \Leftrightarrow \kappa(x)(1-\theta) &= (1-\kappa(x))\theta\\ \Leftrightarrow \hat{\theta} &= \kappa(x)\end{aligned}$$

**Score:**

$$\begin{aligned}S(\theta) &= \frac{\partial \ell(\theta|x)}{\partial \theta}\\ &= \frac{\kappa(x)}{\theta} - \frac{1-\kappa(x)}{1-\theta}\end{aligned}$$

**Information matrix (scalar case!):**

$$\begin{aligned}\mathcal{I}(\theta) &= \mathbb{E}[S(\theta|x)S(\theta|x)]\\ &= -\mathbb{E}[\frac{\partial^2 \ell(\theta|x)}{\partial^2 \theta}] && \text{(Information Matrix Equality)}\\ &= \mathbb{E}[\frac{\kappa(x)}{\theta^2} + \frac{1-\kappa(x)}{(1-\theta)^2}]\\ &= \frac{\theta}{\theta^2} + \frac{1-\theta}{(1-\theta)^2} && (\mathbb{E}(\kappa) = \theta)\\ &= \frac{1}{\theta} + \frac{1}{1-\theta}\\ &= \frac{1-\theta+\theta}{\theta(1-\theta)}\\ &= \frac{1}{\theta(1-\theta)}\end{aligned}$$

### 5. Construct Jeffrey's prior, up to the constant of proportionality. Does it coincide with a flat prior for $\theta$?

Jeffreys prior (up to the constant of proportionality) for the Bernouilli distribution[4]:

$$\begin{aligned}\pi^*(\theta) &\propto det(I(\theta))^{\frac{1}{2}}\\ &\propto (\frac{1}{\theta(1-\theta)})^{\frac{1}{2}}\\ &\propto \frac{1}{\sqrt{\theta(1-\theta)}}\end{aligned}$$

[4] Two interesting references `https://www2.stat.duke.edu/courses/Fall11/sta114/jeffreys.pdf` and `http://jse.amstat.org/v12n2/zhu.pdf`.

This prior is not flat. Note that as $\theta$ approaches 0 or 1 the density goes to infinity, and it is monotonically decreasing in the interval $(0, \frac{1}{2}]$, and monotonically increasing in $[\frac{1}{2}, 1)$. We can also note the distribution is symmetric about $\frac{1}{2}$, and even go so far as to realize that this is the density (up to the constant $\frac{1}{\pi}$) for the Beta$(\frac{1}{2}, \frac{1}{2})$ distribution, so that must be our Jeffrey's prior for this problem.

**6. Consider the parameterization $v = arcsin(\sqrt{\theta})$, ie $\theta = (sin(v))^2$. Given your Jeffrey's prior for $\theta$, what is the equivalent prior for $v$?**

One of the characteristics of the Jeffreys prior is that it is invariant to reparametrizations. From previous question, our Jeffreys prior for $\theta$ is

$$\pi(\theta)^* \propto \frac{1}{\sqrt{\theta(1-\theta)}}$$

Let's define the function

$$v(\theta) = \arcsin\sqrt{\theta}$$

We know that, in our case (where the information matrix is actually a scalar):

$$\begin{aligned}
\det(I(\theta)) &= \det(I(v(\theta))\det(v'(\theta))^2 \\
\det(I(\theta)) &= \det(I(v(\theta))\det(\frac{\partial v(\theta)}{\partial\theta})^2 \\
\det(I(\theta))^{\frac{1}{2}} &= \det(I(v(\theta))^{\frac{1}{2}}\det(\frac{\partial v(\theta)}{\partial\theta}) \\
\det(I(v(\theta))^{\frac{1}{2}} &= \det(I(\theta))^{\frac{1}{2}}\det(\frac{\partial v(\theta)}{\partial\theta})^{-1} \\
\pi(v(\theta))^* &= \det(I(\theta))^{\frac{1}{2}}\det(\frac{\partial v(\theta)}{\partial\theta})^{-1}
\end{aligned}$$

From the previous question we know that $\det(I(\theta))^{\frac{1}{2}} = \pi(\theta)^* \propto \frac{1}{\sqrt{\theta(1-\theta)}}$. Let's compute $\det(\frac{\partial v(\theta)}{\partial\theta})$.
Taking the derivative of $v(\theta)$ wrt $\theta$ holds

$$\begin{aligned}
\frac{\partial v(\theta)}{\partial\theta} &= \frac{1}{\sqrt{1-(\sqrt{\theta})^2}}\frac{\partial\sqrt{\theta}}{\partial\theta} \\
&= \frac{\frac{1}{2}\theta^{-\frac{1}{2}}}{\sqrt{1-(\sqrt{\theta})^2}} \\
&= \frac{1}{2}\frac{1}{\sqrt{(1-\theta)\theta}}
\end{aligned}$$

As we are in the scalar case:

$$\begin{aligned}\det(\frac{\partial v(\theta)}{\partial\theta}) &= \frac{\partial v(\theta)}{\partial\theta}\\ \Leftrightarrow \det(\frac{\partial v(\theta)}{\partial\theta})^{-1} &= 2\sqrt{(1-\theta)\theta}\end{aligned}$$

Finally

$$\begin{aligned}\pi(v(\theta))^* &\propto \det(I(\theta))^{\frac{1}{2}}\det(\frac{\partial v(\theta)}{\partial\theta})^{-1}\\ &\propto \frac{1}{\sqrt{(1-\theta)\theta}} * 2\sqrt{(1-\theta)\theta}\\ &\propto 2\end{aligned}$$

We see that the Jeffrey's prior for $v(\theta)$ is flat. However, this was not the case of the prior for $\theta = v^{-1}$, a reparametrization of $\theta$. This problem is a great example of why uninformative $\neq$ flat in general. Jeffrey's prior is invariant to reparameterization in the sense that both of the parameterizations contain the same information, but one is flat, and the other has density which approaches infinity at the boundaries.

# Problem 3

Let $A$ be a $\Pi$-measurable set. Recall the balance condition is

$$k(\theta' \mid \theta)\pi(\theta) = k(\theta \mid \theta')\pi(\theta')$$

We would like to apply the sequence of equalities below, and we almost can, but we need to check that applying Fubini's theorem to switch the order of integration is legal. The first condition we need is that $k(\theta' \mid \theta)$ is $\Omega \times \Omega$ measurable (assuming $\Omega$ is the underlying probability space). This is true by the assumption of the existence of such a $k(\cdot \mid \cdot)$ which represents the conditional density. We also need that $\Omega$ is a $\sigma$-finite measure space, which is also trivially true because we can take any countable disjoint union of Borel sets which cover $\Omega$ and the sum of the probabilities of these sets will be 1. Thus, we can apply Fubini's theorem to change the order of integration.

$$\begin{aligned}
\int_\theta P(\theta' \in A \mid \theta)\pi(\theta)\lambda(\mathrm{d}\theta) &= \int_\theta \int_{\theta' \in A} k(\theta' \mid \theta)\pi(\theta)\lambda(\mathrm{d}\theta')\lambda(\mathrm{d}\theta) && \text{(Def. of } P(\theta' \in A \mid \theta))\\
&= \int_{\theta' \in A} \int_\theta k(\theta' \mid \theta)\pi(\theta)\lambda(\mathrm{d}\theta)\lambda(\mathrm{d}\theta') && \text{(Fubini)}\\
&= \int_{\theta' \in A} \int_\theta k(\theta \mid \theta')\pi(\theta')\lambda(\mathrm{d}\theta)\lambda(\mathrm{d}\theta') && \text{(Balance)}\\
&= \int_{\theta' \in A} \pi(\theta')\lambda(\mathrm{d}\theta') && (k \text{ is prob. measure})\\
&= \Pi(A)
\end{aligned}$$

# Problem 4

## Part 1

Stack time! Let $x_t = [y_t, y_{t-1}, y_{t-2}]'$. Then

$$\begin{aligned}
y_t &= 2.1y_{t-1} - 1.6y_{t-2} + 0.4y_{t-3} + \epsilon_t\\
\Rightarrow \begin{bmatrix} y_t \\ y_{t-1} \\ y_{t-2} \end{bmatrix} &= \begin{bmatrix} 2.1 & -1.6 & 0.4 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \\ y_{t-3} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_t \\ \epsilon_{t-1} \\ \epsilon_{t-2} \end{bmatrix}\\
x_t &= Bx_{t-1} + A\epsilon_t
\end{aligned}$$

## Part 2

I'm going to go a bit out of order in solving this question, because in order to calculate the variance, I will need some of the autocovariances. First note that

$$\begin{aligned}
E[y_t] &= 2.1E[y_{t-1}] - 1.6E[y_{t-2}] + 0.4E[y_{t-3}] \\
\Rightarrow E[y_t] &= 2.1E[y_t] - 1.6E[y_t] + 0.4E[y_t] \\
\Rightarrow 0.1E[y_t] &= 0 \\
\Rightarrow E[y_t] &= 0
\end{aligned}$$

Now, to use the trick on slide 24 of Topic 5, we need to check that the eigenvalues are all strictly inside the unit circle (it would be great if they were real, also, but alas).

$$\begin{aligned}
\det(B - \lambda I) &= \det\left(\begin{bmatrix} 2.1-\lambda & -1.6 & 0.4 \\ 1 & -\lambda & 0 \\ 0 & 1 & -\lambda \end{bmatrix}\right) \\
&= -1((-1)\cdot 0.4) - \lambda((2.1-\lambda)(-\lambda) - 1\cdot(-1.6)) \\
&= 0.4 - \lambda(\lambda^2 - 2.1\lambda + 1.6) \\
&= -(\lambda^3 - 2.1\lambda^2 + 1.6\lambda - 0.4) \\
&= -(\lambda - \frac{1}{2})(\lambda^2 - 1.6\lambda + 0.8) && \text{(From hint)}
\end{aligned}$$

Using the quadratic equation we find

$$\begin{aligned}
\lambda_1 &= \frac{1}{2} \\
\lambda_2 &= \frac{1.6 + \sqrt{2.56 - 4(1)(0.8)}}{2} = 0.8 + 0.4i \\
\lambda_3 &= \frac{1.6 - \sqrt{2.56 - 4(1)(0.8)}}{2} = 0.8 - 0.4i
\end{aligned}$$

We can also go ahead and verify that our above equation matches (up to sign) the characteristic polynomial

$$p(\lambda) = \lambda^3 - 2.1\lambda^2 + 1.6\lambda - 0.4$$

So the eigenvalues of $B$ and the roots of the characteristic polynomial are the same (as they always are).

Now we check that all the roots are inside the unit circle.

$$\begin{aligned}
\lambda_1\bar{\lambda}_1 &= 0.25 < 1 \\
\lambda_2\bar{\lambda}_2 &= 0.8 < 1
\end{aligned}$$

Now we can use the vectorizing trick (with an inversion we know exists) from the slides to calculate $\Gamma_0$, and from there calculate $\Gamma_i$ to get higher autocovariances

$$\begin{aligned}
\text{vec}(\Gamma_0) &= (I_9 - B \otimes B)^{-1} \text{vec}(A\Omega A') \\
B &= \begin{bmatrix} 2.1 & -1.6 & 0.4 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\Omega &= \begin{bmatrix} 2.55 & 0 & 0 \\ 0 & 2.55 & 0 \\ 0 & 0 & 2.55 \end{bmatrix}
\end{aligned}$$

Since we are now dealing with $9 \times 9$ matrices (and inverting them) I'm now going to use the computer, since the calculation is fairly rote. I find

$$\Gamma_0 = \begin{bmatrix} 100 & 91.25 & 68.125 \\ 91.25 & 100 & 91.25 \\ 68.125 & 91.25 & 100 \end{bmatrix}$$

Therefore $\text{var}[y_t] = E[y_t^2] = 100$.

## Part 3

We can use our framework from above, and the knowledge that $E[x_t x_{t-k}] = B^k \Gamma_0$, from the Yule-Walker equation, to realize that by considering the upper left entry of $B^j \Gamma_0$ for $j \in \{1, \dots, 6\}$, we find the $j$-th autocovariance. We find

$$\begin{aligned}
E[y_t y_{t-1}] &= 91.25 \\
E[y_t y_{t-2}] &\approx 68.13 \\
E[y_t y_{t-3}] &\approx 37.06 \\
E[y_t y_{t-4}] &\approx 5.33 \\
E[y_t y_{t-5}] &\approx -20.85 \\
E[y_t y_{t-6}] &\approx -37.50
\end{aligned}$$

## Part 4

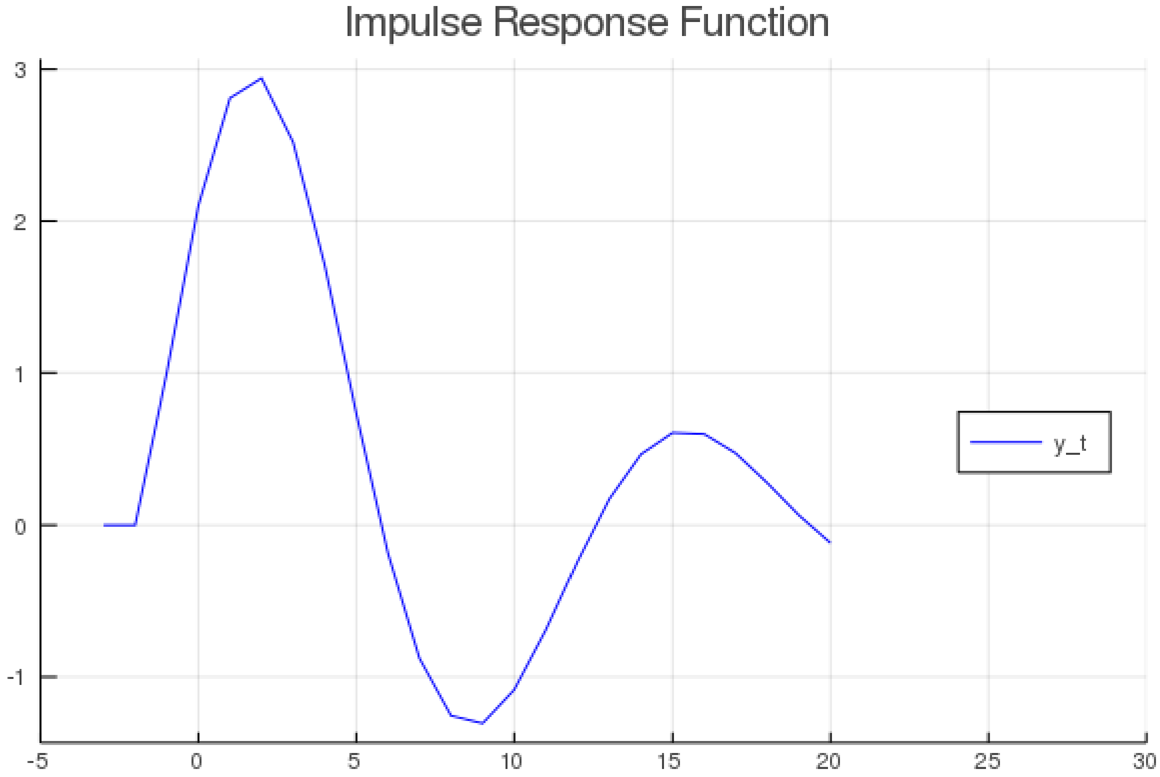
We did this above

$$\begin{aligned} p(\lambda) &= \lambda^3 - 2.1\lambda^2 + 1.6\lambda - 0.4 \\ \lambda_1 &= \frac{1}{2} \\ \lambda_2 &= 0.8 + 0.4i \\ \lambda_3 &= 0.8 - 0.4i \end{aligned}$$

## Part 5

Again, already done, and they match the roots.

## Part 6



## Part 7

Our impulse response shows an initial jump with the shock, then a rise for a few more periods, before a decline to below zero, then an increase to above zero, before a final decrease. Generally, this curve looks like a damped oscillation, which is what we would expect given two complex roots less than one in absolute value[5].

[5] Here we are saying the true roots are complex, so this is different than the below problem, where the *estimated* roots are complex.

I have alluded to the eigenvalue interpretation of this plot, which basically says that the real root should lead to standard decay, but the complex conjugate pair of roots will lead to damped oscillation, so altogether that is what we see. An extrapolation to $t = 50$ confirm that the stability kicks in and the oscillation bascially disappears due to convergence back to zero.

This plot also makes sense with our autocovariance calculations. We had positive autocovariance for the first few periods, then a swap to negative autocovariance. Since we only have one shock, it is not surprising that this autocovariance structure plays out in our imulse response.

## Part 8

The analysis is much the same as above, but the major tweak is that we now have a unit root.

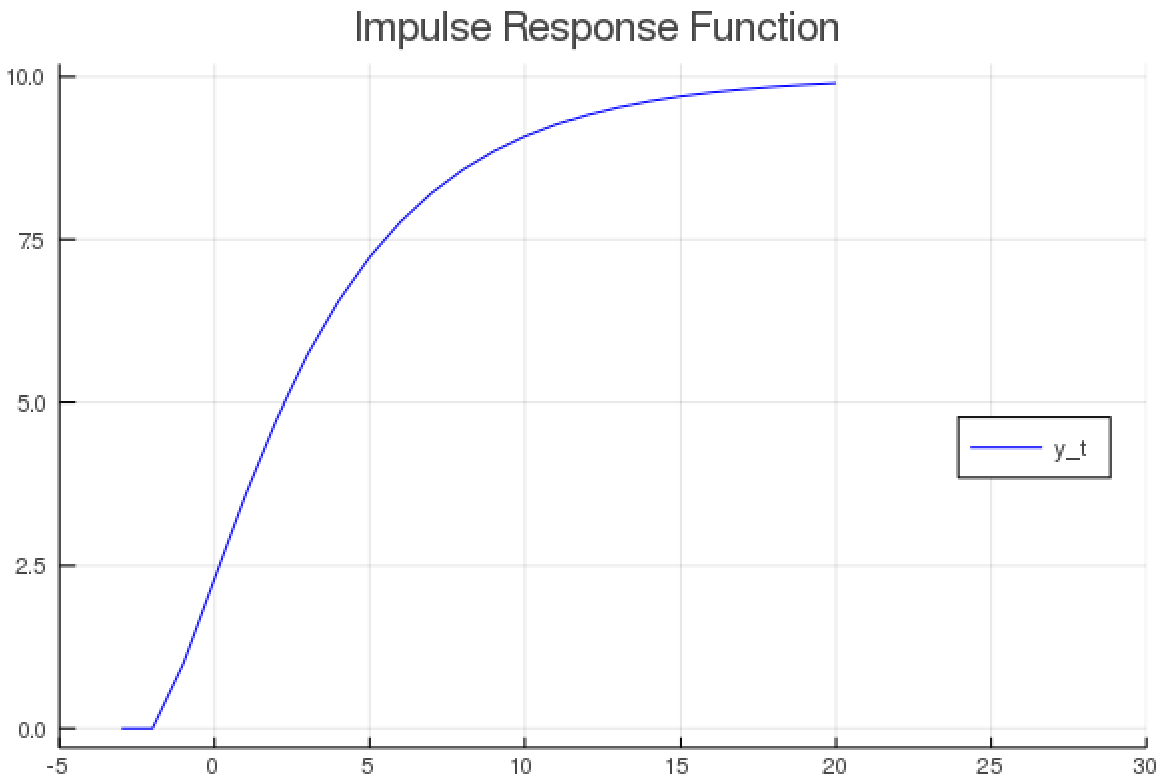
$$\begin{aligned} y_t &= 2.3y_{t-1} - 1.7y_{t-2} + 0.4y_{t-3} + \epsilon_t \\ \Rightarrow \begin{bmatrix} y_t \\ y_{t-1} \\ y_{t-2} \end{bmatrix} &= \begin{bmatrix} 2.1 & -1.6 & 0.4 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \\ y_{t-3} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_t \\ \epsilon_{t-1} \\ \epsilon_{t-2} \end{bmatrix} \\ x_t &= Bx_{t-1} + A\epsilon_t \end{aligned}$$

The eigenvalues/roots (calculated as above, from $\det(B - \lambda I)$ or the roots of $p(\lambda)$) are

$$\begin{aligned} \lambda_1 &= 0.5 \\ \lambda_2 &= 0.8 \\ \lambda_3 &= 1 \end{aligned}$$

Therefore we cannot simply apply the vectorizing trick to use Yule-Walker to find autocovariances. In fact, if we only have explosive roots then we could "flip" some of them in such a way as to find the covariances, but since we have a unit root, there will be non-stationarity, so the covariances will be a function of time. Additionally, since we can understand this process as having a "random walk" component, the variance grows infinitely large as $t \to \infty$. But computing the specific autocovariances is now quite tricky due to the time dependence.

The impulse response is



We no longer have the oscillatory or pure stationary elements of the other process. Note that the oscillations died because all the roots are real, and the stationarity died because of a unit root. The apparently convergent behavior is easily understood by considering the eigenvalues. If, for example, we were dealing with an AR(1) with unit root, then the impulse response would be constant at the shock level. Here however, there are also positive, less than unity roots, so that the shock is persistently increase $y_t$ over time, but the lagged power of the initial shock is decreasing over time. So the process should be converging upward to some value, where the unit root maintains the high level, and the other roots guarantee a convergent process, in this case up, due to positivity.

```
# Initialize lag coefficients
rho1 = 2.3
rho2 = -1.7
rho3 = 0.4

# Companion matrix
B = [rho1 rho2 rho3; 1 0 0; 0 1 0]

# Shock matrix
A = [1.0 0 0; 0 0 0; 0 0 0]

# variance of sigma
sigmas = 2.55
```

```
# iid shocks
Omega = [sigmas 0 0; 0 sigmas 0; 0 0 sigmas]
lag = length(B[1,:])


# Eigenvale check
e = eigvals(B)

# Prints whether each eigenvalue is going to be problematic or not
for x in e
    print("Eigenvalue: ", x, " -> ")
    if abs(x) >= 1.0
        println("Unit Root or Explosive!")
    else
        println("Nice!")
    end
end

# vectorizing is pretty easy in Julia, just X[:] = vec(X)

# Use handy vectorizing inversion trick if all eigenvalues are inside unit circle
vA = A*Omega*A'
vGamma0 = inv(I - kron(B,B))vA[:]

Gamma0 = reshape(vGamma0, size(B))

# Function for iterating to find covariances
function Gammai(i, G)
    return B^i * G
end

# Find covariance matrices
for i in 0:6
    println("Cov y_t and y_t-", i, " : ", Gammai(i, Gamma0)[1,1])
end

# Horizon
t = 20

# x-axis values
x = [-1*lag:t]

# initialize y values
y = zeros(t + lag + 1)
```

```
# shock magnitude
ep = 1

# Shock at t =0
y[lag] = ep

# Iterate to find time series
for i in lag + 1: lag + t + 1
   y[i] = rho1*y[i-1] + rho2*y[i-2] + rho3*y[i-3]
end

# Plot time series
plt = plot(x, y, label = "y_t", legend = :right, seriescolor = "blue", fillcolor =
"blue", markerstrokecolor = "blue", linecolor="blue", xlim=(-1*lag - 2,t + 10),
title = "Impulse Response Function")

display(plt)
savefig("IRF2.png")
```

## Problem 5

The general idea here is that no, estimates of complex roots do not necessarily indicate truly oscillatory behavior. In this case, we know the "true" eigenvalues of the companion matrix (roots of characteristic polynomial) are all zeros, since the true matrix is

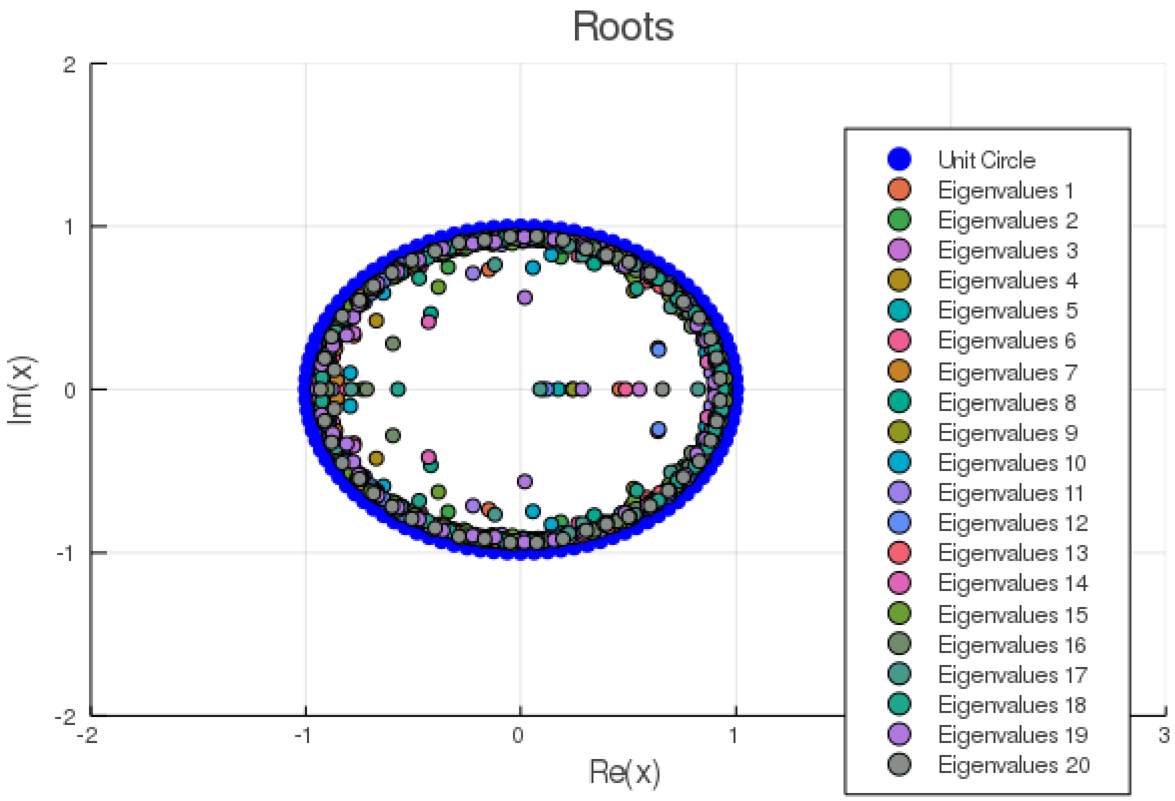
$$\begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}$$

because all $\rho_i = 0$ due to iid. However, we end up seeing this strange equispacing of eigenvalues around the inside of the unit circle, and a little tampering with the AR lag (say, try AR(100)) reveals that the norm, ($\lambda \cdot \bar{\lambda}$, where the bar indicates the complex conjugate) appears to be approaching 1.

So let's step back. It is not too surprising that OLS estimates a bunch of near unital complex roots, which are sometimes thought of as indicating oscillation, because when we have a high enough number of draws, we could have eigenvalues that are "jumping" back and forth between consecutive lags, and over the last 50 iid draws, they should roughly cancel to zero. In fact, if the data "appears" at all correlated (which will happen with probability 1, since there is probability zero that our draws generate an OLS with all zero eigenvalues), then we should expect that the BLUE will have all near unital eigenvalues, evenly dispersed around the inside perimeter of the unit circle.

Returning to the limit conjecture: what if we had AR($\infty$)? Then we would have infinitely many past observations to regress upon, and by the weak law of large numbers tells us their mean will be zero. Therefore, in this case perhaps the eigenvalues *will* be unit roots spread around the unit circle, such that, *for the data*, the sum $\rho_1 y_{t-1} + \rho_2 y_{t-2} + \cdots$ is zero always (we can make this happen because our estimator has infinitely many degrees of freedom to play with). Note that in this case, we could consider the process an MA(1), because each new observation is really just a shock. The difficulty here is the interplay between the amount of data and the lag process. More data will push the eigenvalues inward, but a longer lag tends to push them outward (all in terms of OLS estimation). In fact, with a sufficiently long lag and sufficiently short data set, we can even generate explosive roots.

Another way to understand this result is by realizing that white noise, in the frequency domain, has equal intensity at all levels. So it should be somewhat unsurprising that perhaps the eigenvalues of a regression on a white noise process are equally spaced around the unit circle.



```
# Number of times to repeat experiment
m = 20

# time series length
t = 10000

# lag for regression
ar = 50
```

```
# Initialize eigenvalues
e = zeros(Complex{Float32}, m, ar)
for j in 1:m

    # Draw
    y = randn(t)

    # When stuck, stack! Make AR(50) into VAR(1)
    x = zeros(t-ar,ar)
    for i = 1:t - ar
        x[i,:] = y[i:i+ar-1]
    end

    # format into X_t and X_{t-1}
    xt = x[2:t-ar,:]
    xtm = x[1:t-ar-1,:]

    # Run OLS (manually)
    hatA = inv(xtm'*xtm)*(xtm'*xt)

    # Get eigenvaluess
    e[j,:] = eigvals(hatA)
end

# Plot the unit circle
gran = 0.01
range = 0:gran:1
z = [exp(2im*pi*i) for i in range]
plt = plot(z, seriestype=:scatter, label = "Unit Circle", legend = :right, seriescolor =
    markerstrokecolor = "blue", linecolor="blue", xlim=(-2,3), ylim=(-2,2), title = "Roo

# Add the eigenvalues
for j in 1:m
    plot!(e[j,:], label = string("Eigenvalues ",j), seriestype=:scatter)
end

display(plt)

savefig("whitenoiseroots.png")
```