Required Problems

1. Given an $m \times n$ matrix A with (full format) singular value decomposition $A = U\Sigma V^T$, with

$$\Sigma = \begin{bmatrix} \sigma_1 & & & 0 \\ & \ddots & & 0 \\ & & \sigma_r & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

define Σ^+ to be the $n \times m$ matrix

$$\Sigma^+ = \begin{bmatrix} 1/\sigma_1 & & & 0 \\ & \ddots & & 0 \\ & & 1/\sigma_r & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Define the **pseudoinverse** of A, denoted A^+, to be $A^+ = V\Sigma^+U^T$.

(a) Compute $\Sigma^+\Sigma$. How many rows and columns does it have?
(b) Compute $\Sigma\Sigma^+$. How many rows and columns does it have?
(c) How many rows and columns does A^+ have?
(d) Show that $AA^+ = U_rU_r^T$.
(e) Prove that if A has full row rank, $AA^+ = I_m$.
(f) Show that $A^+A = V_rV_r^T$.
(g) Prove that if A has full column rank, $A^+A = I_n$.
(h) Prove that if A is invertible, $A^+ = A^{-1}$.

Solution:

(a) $\Sigma^+\Sigma$ is the $n \times n$ block matrix of the form $\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$. (Note that $\Sigma^+\Sigma$ is a projection of $\mathbb{R}^n$ onto the first r coordinates.)

(b) $\Sigma\Sigma^+$ is the $m \times m$ block matrix of the form $\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$. (Note that $\Sigma\Sigma^+$ is a projection of $\mathbb{R}^m$ onto the first r coordinates.)

(c) Since V is $n \times n$, Σ^+ is $n \times m$, and U^T is $n \times n$, $A^+ = V\Sigma^+U^T$ is $n \times m$.

(d) We use the fact that the $m \times m$ block matrix of the form $\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$ is equal to its own square, $\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}^2 = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$. (After all it's a projection matrix). It is also equal to its own transpose. Thus

$$
\begin{aligned}
AA^+ &= (U\Sigma V^T)(V\Sigma^+U^T) \\
&= U\Sigma(V^TV)\Sigma^+U^T \\
&= U(\Sigma\Sigma^+)U^T \\
&= U \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} U^T \\
&= U \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} U^T \\
&= \left(U \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \right) \left(\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}^T U^T \right) \\
&= \begin{bmatrix} U_r & 0 \end{bmatrix} \begin{bmatrix} U_r & 0 \end{bmatrix}^T \\
&= U_r U_r^T
\end{aligned}
$$

(e) If A has full row rank, then $r = m$, so $AA^+ = U_rU_r^T = U_mU_m^T = UU^T = I_m$, since $U^T = U^{-1}$.
(A matrix with this property is called a right inverse for A.)

(f) We use the fact that the $n \times n$ block matrix of the form $\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$ is equal to its own square, $\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}^2 = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$. (After all it's a projection matrix). It is also equal to its own transpose. Thus

$$
\begin{aligned}
A^+A &= (V\Sigma^+U^T)(U\Sigma V^T) \\
&= V\Sigma^+(U^TU)\Sigma V^T \\
&= V(\Sigma^+\Sigma)V^T \\
&= V \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} V^T \\
&= V \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} V^T \\
&= \left(V \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \right) \left(\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}^T V^T \right) \\
&= \begin{bmatrix} V_r & 0 \end{bmatrix} \begin{bmatrix} V_r & 0 \end{bmatrix}^T \\
&= V_r V_r^T
\end{aligned}
$$

(g) If A has full column rank, then $r = n$, so $A^+A = V_rV_r^T = V_nV_n^T = VV^T = I_n$, since $V^T = V^{-1}$.
(A matrix with this property is called a left inverse for A.)

(h) If A is invertible, then A is $n \times n$, and by Parts (e) and (g) $AA^+ = A^+A = I_n$. Thus $A^+ = A^{-1}$. (If a matrix is a left inverse and a right inverse, it is an inverse.)

2. Pseudoinverse, projection, and least squares:

   (a) Prove that AA^+ is the projection matrix P onto the column space of A. (Hint: The column space of A is the same as the column space of U_r).

   (b) Prove that A^+A is the projection matrix P onto the row space of A. (Hint: The row space of A^T is the same as the column space of V_r).

   (c) Consider the equation $A\mathbf{x} = \mathbf{b}$. Show that the best least squares solution $\hat{\mathbf{x}}$ to this equation is $\hat{\mathbf{x}} = A^+\mathbf{b}$.

   Note that when we discussed least squares solutions in Chapter 4, we required A to have independent columns. Using the SVD and pseudoinverse, we can now do projections and least squares approximation for any matrix. For a matrix with dependent columns, there are an infinite number of solutions to the equation $A^TA\mathbf{x} = A^T\mathbf{b}$, and $A^+\mathbf{b}$ gives the shortest one (the contribution from $N(A)$ is $\mathbf{0}$).

   To compute the pseudoinverse of A in Julia, use the command "pinv(A)" (using the LinearAlgebra package). See Julia code for HW 11 for examples.

   Solution:

   (a) Since $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ is a basis for $C(A)$, the column space of U_r is the same as the column space of A, and the columns of U_r are independent. Thus the projection matrix P onto this space is $P = U_r(U_r^TU_r)^{-1}U_r^T$. But $U_r^TU_r = I_r$, so $P = U_rU_r^T = AA^+$.

   (b) Since $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ is a basis for $C(A^T)$, the column space of V_r is the same as the row space of A, and the columns of V_r are independent. Thus the projection matrix P onto this space is $P = V_r(V_r^TV_r)^{-1}V_r^T$. But $V_r^TV_r = I_r$, so $P = V_rV_r^T = A^+A$.

   (c) The best least squares solution to $A\mathbf{x} = \mathbf{b}$ is a vector $\hat{\mathbf{x}}$ such that $A\hat{\mathbf{x}} = P\mathbf{b} = AA^+\mathbf{b}$. One solution to this equation is $\hat{\mathbf{x}} = A^+\mathbf{b}$. (Note that if A has dependent columns, there are an infinite number of solutions.)

3. Suppose A has (full format) singular value decomposition $A = U\Sigma V^T$, where $\mathbf{u}_i$ is the ith column of U and $\mathbf{v}_i$ is the ith column of V, as usual. Prove that for $1 \le i \le r$, $A^T\mathbf{u}_i = \sigma_i\mathbf{v}_i$. What is $A^T\mathbf{u}_i$ when $r < i \le m$?

   Solution: **Proof:** Let $1 \le i \le r$. Then

$$\begin{aligned} A^T\mathbf{u}_i &= (U\Sigma V^T)^T\mathbf{u}_i \\ &= V\Sigma^TU^T\mathbf{u}_i \\ &= V\Sigma^T\mathbf{e}_i \\ &= V\sigma_i\mathbf{e}_i \\ &= \sigma_iV\mathbf{e}_i \\ &= \sigma_i\mathbf{v}_i. \end{aligned}$$

■

If $i > r$, $A^T \mathbf{u}_i = \mathbf{0}$.

4. In this problem we use PCA/SVD for data visualization. We will use the famous data set of four measurements on three types of iris flowers–see

   `https://en.wikipedia.org/wiki/Iris_flower_data_set`

   The first 50 rows are flowers of one species, the next 50 are flowers from a second species, and the last 50 are flowers of a third species. For flower i, the ith row contains measurements of

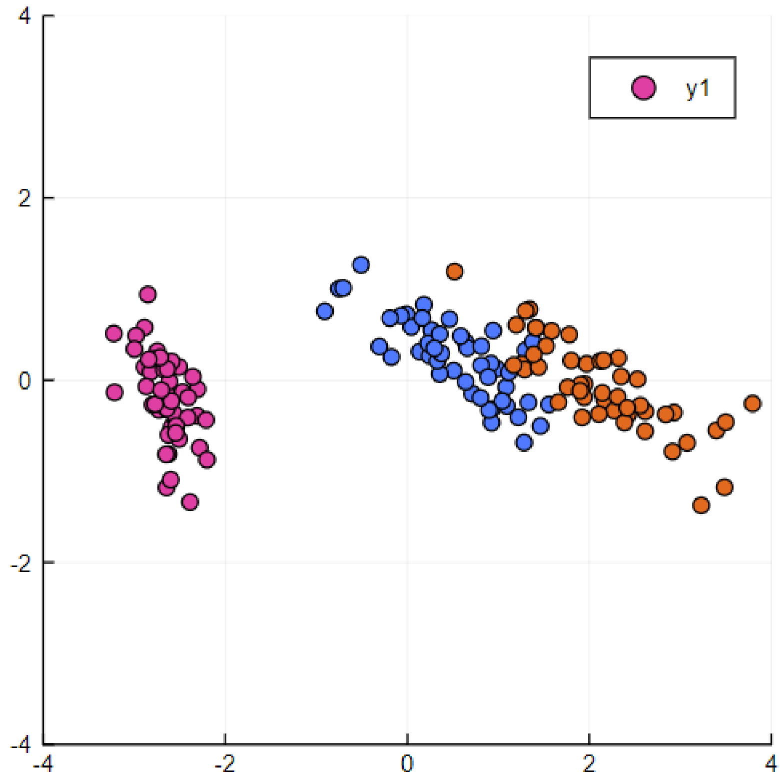
   $$\mathbf{x}_i^T = (\text{SepalLength}(i), \text{SepalWidth}(i), \text{PetalLength}(i), \text{PetalWidth}(i)),$$

   along with its species in the fifth column. Since $\mathbf{x}_i \in \mathbb{R}^4$ for each i, each flower corresponds to a vector in $\mathbb{R}^4$. We want to visualize this data, so we find the best least-squares projection onto a plane. To do this, you should

   - Import the data set.
   - Center the data to create the 150×4 matrix A.
   - Find the SVD for A, $A = U\Sigma V^T$.
   - Let $\mathbf{y}_1 = A\mathbf{v}_1$. The ith component of the vector $\mathbf{y}_1$ contains the length of the projection of $\mathbf{x}_i$ onto $\mathbf{v}_1$.
   - Let $\mathbf{y}_2 = A\mathbf{v}_2$. The ith component of the vector $\mathbf{y}_2$ contains the length of the projection of $\mathbf{x}_i$ onto $\mathbf{v}_2$.
   - Make a scatter plot of $(\mathbf{y}_1, \mathbf{y}_2)$ where the color of each point corresponds to the species of the flower.
   - What is a typical coefficient for $\mathbf{v}_1$ and $\mathbf{v}_2$ for a given flower of each species?

   The "Julia code for Homework 11" notebook on Canvas shows how to complete each step. Your hw does not need to include your code, but it should include your scatter plot.

   Solution: Here is the projection of the 150 data points onto the plane spanned by $\mathbf{v}_1$ (horixontal axis) and $\mathbf{v}_2$ (vertical axis)



To calculate the average length of projection onto $\mathbf{v}_1$ and $\mathbf{v}_2$ for each flower, we take the mean of the first 50, middle 50, and last 50 entries of the vectors $\mathbf{y}_1 = A\mathbf{v}_1$ and $\mathbf{y}_2 = A\mathbf{v}_2$. (See the accompanying "Julia code for Homework 11 solutions" notebook for the calculations.)

For the first species, the average projection lengths (coordinates in the scatter plot) are -2.64 and $-.19$ onto $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively.

For the second species, the average projection lengths (coordinates in the scatter plot) are .53 and .25 onto $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively.

For the third species, the average projection lengths (coordinates in the scatter plot) are 2.11 and $-.05$ onto $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively.

5. Low rank approximation: In this problem we will find a low-rank approximation for a data matrix. We will use a data set of heptathlon scores for 25 athletes in the heptathlon (7 events) from the 1988 Olympics (25 rows, 7 columns)–note the original data matrix has nine columns, where the first column is the athlete's name and the last column is their cumulative score. There are three races in the heptathlon (hurdles, 200m, 800m, columns 1, 4, and 7) so the lower scores for these events are better, while for the rest of the events (high jump, long jump, shot put, javelin) higher scores are better. Similar to the previous problem, you first want to import the data set and center it to get the 25×7 data matrix A. The "Julia code for Homework 11" notebook on Canvas shows how to complete each step.

   Let's see if doing PCA on A tells us anything useful.

- Find the SVD for A, $A = U\Sigma V^T$.
- What event explains almost all of the variance (find the element of $\mathbf{v}_1$ that is much larger than all the others).
- The first few principal components are each close to a standard basis vector (one component dominates). What are these components telling us?

The example above shows how PCA is sensitive to the units of a data set. To find some more useful information, let's create another matrix B where we normalize each column so it is a unit vector. (Statistically, we are now working with with correlations, not covariances).

Just to make our results easier to interpret for humans, let's also negate the entries in columns 1, 4, and 7 (note they are still unit vectors!) so that now for every column, a higher number corresponds to a better performance.

- Can you interpret the first couple of principal components? (Before looking at the data, I expected there might be one component corresponding to track events, one to jumping events, one to throwing events, etc. But it doesn't work out that way. The first principal component seems to correspond to general excellence (or deficiency) in all events except one.)
- Compute the rank 1 approximation for B. Find the coefficient c_1 so that the first row of B is approximated by $c_1\mathbf{v}_1$.
- Compute the rank 2 approximation for B. Find the coefficient c_2 so that the first row of B is approximated by $c_1\mathbf{v}_1 + c_2\mathbf{v}_2$.

Solution: With the original data matrix A (centered, but not normalized), the first few principal components are

$$\begin{aligned}
\mathbf{v}_1 &= (0.069, -0.006, -0.078, 0.073, -0.040, 0.007, 0.991) \\
\mathbf{v}_2 &= (-0.009, 0.001, 0.136, -0.101, 0.015, 0.985, 0.013) \\
\mathbf{v}_3 &= (0.222, -0.015, -0.884, 0.310, -0.185, 0.160, -0.117)
\end{aligned}$$

These first two vectors are close to $\mathbf{e}_7$ and $\mathbf{e}_6$, respectively, which correspond to events 7 (800m run), and 6 (javelin) explaining most of the variance in the data. But here we see these are the two events whose measurements are largest. So here the "variance explained" is just telling us which coordinates have the largest numbers–not really anything about the relationships between the variables or underlying "latent" variables.

The vector $\mathbf{v}_3$ has most significant contributions from events 1, 3 and 4, but again, this just tells us that these events have the next highest measurements (though it is interesting that event 3 has the highest contribution among these three, indicating shot put scores are a bit more varied than 200m or hurdles scores).

With the centered normalized data matrix B, the first two principal components are

$$\mathbf{v}_1 = (-0.453, -0.377, -0.363, -0.408, -0.456, -0.075, -0.375)$$
$$\mathbf{v}_2 = (0.157, 0.248, -0.289, -0.261, 0.056, -0.842, 0.224)$$

Here we see that with normalized data, most events explain about the same amount of variance, with the exception of event 6, the javelin throw. The first principal component could be interpreted as general success (athleticism), with the exception of the javelin. The second principal component has largest entry in event 6, so the highest amount of variance not explained by $\mathbf{v}_1$ is explained by the javelin throw.

See the accompanying Julia document for the rank 1 and rank 2 approximations for B. If we let $\mathbf{x}_1^T$ represent the first row of B, i.e.

$$\mathbf{x}_1 = (0.319, 0.204, 0.367, 0.440, 0.481, 0.241, 0.186),$$

then the rank 1 approximation for this row is given by

$$\mathbf{x}_1 \approx (\mathbf{x}_1^T\mathbf{v}_1)\mathbf{v}_1 = -0.841\mathbf{v}_1 = (0.381, 0.317, 0.305, 0.343, 0.384, 0.063, 0.315).$$

(The projection onto $\mathbf{v}_1$ has length $|c_1| = 0.841$.)

The rank 2 approximation for $\mathbf{x}_1$ is given by

$$\mathbf{x}_1 \approx (\mathbf{x}_1^T\mathbf{v}_1)\mathbf{v}_1 + (\mathbf{x}_1^T\mathbf{v}_2)\mathbf{v}_2 = -0.841\mathbf{v}_1 - 0.253\mathbf{v}_2 = (0.341, 0.254, 0.379, 0.409, 0.370, 0.277, 0.259).$$

(The projection onto $\mathbf{v}_2$ has length $|c_2| = 0.253$.)

6. Generate a perspective drawing with the following specifications:

   - Your image plane is 2×2, containing 1,000,000 square pixels (1000×1000).
   - Your camera is 1 unit behind the center of the image plane.

   (a) What is your camera basis?

   (b) Use the procedure outlined in class to find the "camera coordinates" and pixel locations for the following points (camera at the origin)

   $$(1,1,5), (-1,1,5), (-1,-1,5), (1,-1,5), (1,1,8), (-1,1,8), (-1,-1,8), (1,-1,8)$$

   (c) Plot the image of these eight points. Note they form the corners of a box. How many sides of the box could you see in this rendering if it were solid and opaque?

   (d) Plot the image of the same box in two other locations in space, one where two sides of the box are visible, and one where three sides of the box are visible.

Solution:

(a) Since each pixel is a square with sidelength $1/500$, $\mathbf{a}_1 = (1/500, 0, 0)$ and $\mathbf{a}_2 = (0, 1/500, 0)$. Since the upper left corner of the image plane is one unit in front (z-axis), above (y-axis), and to the left (x-axis) of the camera, $\mathbf{a}_3 = (-1, -1, 1)$. Thus the camera basis $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\} = \{(1/500, 0, 0), (0, 1/500, 0), (-1, -1, 1)\}$.

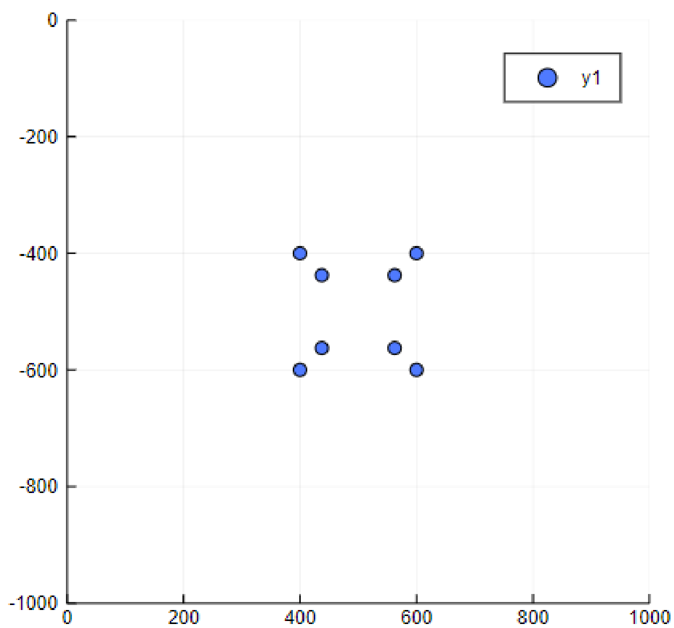
(b) The camera coordinates of the eight points are given by $A^{-1}P$, where P is the matrix each of whose columns is one of the points we wish to plot, and A is the (change of basis) matrix where $\mathbf{a}_i$ is the ith column. See the "Julia code for Homework 11 solutions" for calculations that give camera coordinates

$$A^{-1}P = \begin{bmatrix} 3000.0 & 2000.0 & 2000.0 & 3000.0 & 4500.0 & 3500.0 & 3500.0 & 4500.0 \\ 3000.0 & 3000.0 & 2000.0 & 2000.0 & 4500.0 & 4500.0 & 3500.0 & 3500.0 \\ 5.0 & 5.0 & 5.0 & 5.0 & 8.0 & 8.0 & 8.0 & 8.0 \end{bmatrix}$$

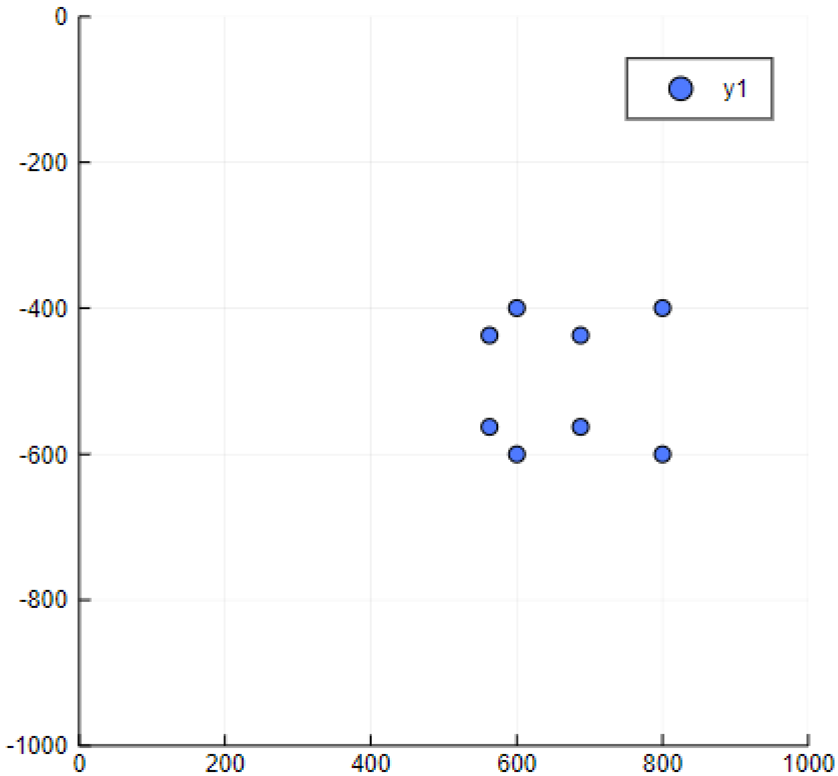
The locations of the points in the image plane of the points is found by scaling each column by the third entry to obtain

$$Q = \begin{bmatrix} 600.0 & 400.0 & 400.0 & 600.0 & 562.5 & 437.5 & 437.5 & 562.5 \\ 600.0 & 600.0 & 400.0 & 400.0 & 562.5 & 562.5 & 437.5 & 437.5 \end{bmatrix}$$

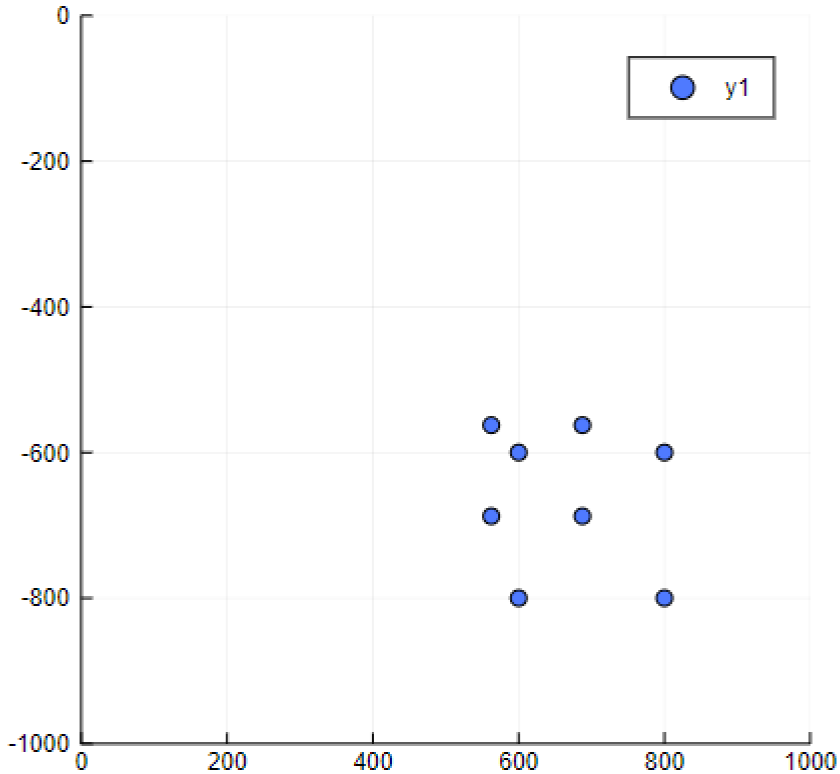
(c) Here is a scatter plot of these eight points. If the box were opaque we would only see the front side of the box.



(d) Adding 2 to the first coordinate of each point moves the box to the side, so we see 2 sides of it:



Adding 2 to the first two coordinates of each point moves the box to the side and down (recall the y-axis is oriented downward), so we see 3 sides of it:


y1
0
-200
-400
-600
-800
-1000
0
200
400
600
800
1000