Problem 1.
Orthonormal vectors are those with unit length 1 and whose dot product between any two given vectors is 0 (the columns are orthogonal). When the columns are orthonormal, we have:

$$A^T A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots a_n^T \end{bmatrix} [a_1, a_2, \dots, a_n] = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \dots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \dots & a_2^T a_n \\ \vdots & & & \vdots \\ a_n^T a_1 & a_n^T a_2 & \dots & a_n^T a_n \end{bmatrix} = \begin{bmatrix} \langle a_1, a_1 \rangle & \langle a_1, a_2 \rangle & \dots & \langle a_1, a_n \rangle \\ \langle a_2, a_1 \rangle & \langle a_2, a_2 \rangle & \dots & \langle a_2, a_n \rangle \\ \vdots & & & \vdots \\ \langle a_n, a_1 \rangle & \langle a_n, a_2 \rangle & \dots & \langle a_n a_n \rangle \end{bmatrix}$$

The dot product of any vector a_i with itself is the vector's L_2 norm squared $||a_i||_2^2$. For vectors a_i and a_j where $i \neq j$, the cross product is 0 since they are all orthogonal. Thus the above matrix can only be the identity matrix. The above result holds for the matrix product AA^T as well, which we can think of as the rows of A being multiplied by itself, thus reaffirming the rows of A are also orthonormal.

Problem 2.

$$\begin{aligned}
\sum_{i,j} Var(M_{i,j}) &= \sum_{i,j} E(M_{i,j}^2) - [E(M_{i,j})]^2 \\
&= \sum_{i,j} E((\frac{1}{r}\sum_{l=1}^{r} \frac{1}{p_{i_l}} A_{:,i_l} B_{i_l,:})^2) \\
&= \sum_{i,j} \big([\sum_{l=1}^{r}\sum_{k}^{n} \mathbb{P}\{i_l = k\} \frac{1}{(rp_k)^2} A_{i,k}^2 B_{k,j}^2]^2 - [\sum_{l=1}^{r}\sum_{k}^{n} \mathbb{P}\{i_l = k\} \frac{1}{rp_k} A_{i,k} B_{k,j}]^2\big) \\
&= \sum_{i,j} \big([\sum_{l=1}^{r}\sum_{k}^{n} \frac{1}{r^2 p_k} A_{i,k}^2 B_{k,j}^2]^2 - [\sum_{l=1}^{r}\sum_{k}^{n} \frac{1}{r} A_{i,k} B_{k,j}]^2\big) \\
&= \sum_{i,j}\sum_{l=1}^{r} \big([\sum_{k}^{n} \frac{1}{r^2 p_k} A_{i,k}^2 B_{k,j}^2]^2 - [\sum_{k}^{n} \frac{1}{r} A_{i,k} B_{k,j}]^2\big) \\
&= \sum_{i,j}\sum_{l=1}^{r} \big([\sum_{k}^{n} \frac{1}{r^2 p_k} A_{i,k}^2 B_{k,j}^2]^2 - [\frac{1}{r} A_{i,:} B_{:,j}]^2\big) \\
&= \sum_{i,j}\sum_{l=1}^{r} \big([\sum_{k}^{n} \frac{1}{r^2 p_k} A_{i,k}^2 B_{k,j}^2]^2 - \big(\frac{1}{r^2} A_{i,:}^2 B_{:,j}^2\big)\big) \\
&= \frac{1}{r}\sum_{i,j} \big([\sum_{k}^{n} \frac{1}{r^2 p_k} A_{i,k}^2 B_{k,j}^2]^2 - \sum_{i,j} \frac{1}{r}\big(A_{i,:}^2 B_{:,j}^2\big)\big)
\end{aligned}$$

Problem 3.
a. For random matrix multiplication, we first calculate non-uniform probabilities for vector p_k. After, we randomly sample indices of size r from the columns of A and the rows of B. Each respective vector from each matrix has the outer product solved for, scaled by its inverse probability, summed over all other outer products, and scaled once more by $\frac{1}{r}$. The result is the matrix M, which is a randomized product the product AB
b. We run four instances of this multiplication through the function, which can be observed in the .ipynb file.
c.

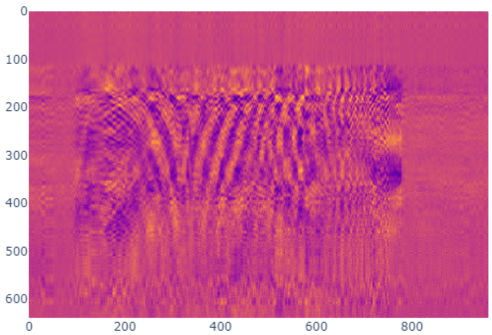
R = 20	R = 50	R = 100	R = 200
0.202161	0.155614	0.092024	0.058732

Table 1: Relative Approximation Error for Each Value of R

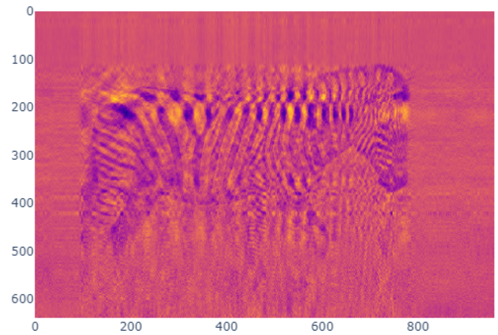
From the table, we can see that as R increases, that is, as we increase the number of vectors we sample, the approximation error between the true matrix product and our randomized approximation decreases. However, we do sacrifice computational speed when increasing R as well

d.

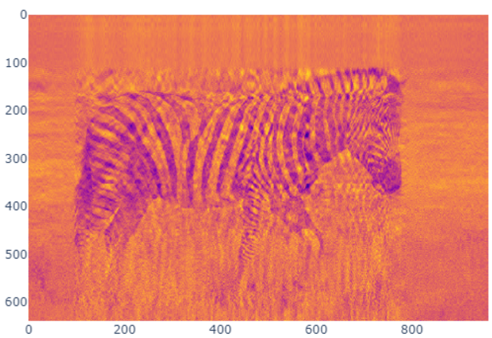
Below are the visualizations for the matrix product with the value of R increasing.



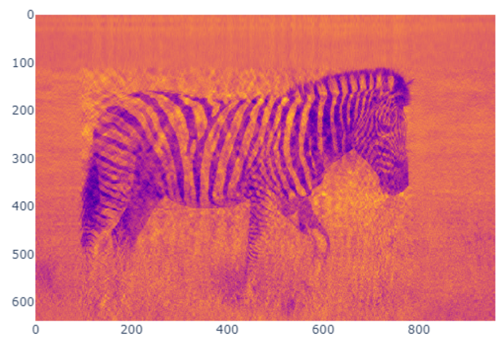
(a) $R = 20$



(b) $R = 50$



(c) $R = 100$



(d) $R = 200$

Figure 1: The visualization of the randomized matrix multiplication as R increases

From these images, we can see that as we increase R, we get a more accurate representation of the zebra, which makes sense with our earlier conclusion regarding the increasing of R and its effect on the error.

Problem 4.

Below in Figure 2, we can see the plot from the implemented power function. We can see that as the value of λ increases, we can see that the correlation between our estimated eigenvector and our true eigenvector increases.

Problem 5.

a. For sketched-OLS, implementing the direct matrix multiplication would be too intensive computationally. So in the implementation, specifically for ΦX and Φy, we vectorize the processes for $S^T HD$. While Y is just a column vector and easy to perform vector operations, X is a matrix, so we apply the vectorized functions iteratively to each column of X.

b. Please refer to the attached .ipynb for the implementation of the data generation.

c. In the table below, we can compare the calculation of the β vector with Vanilla OLS and sketched OLS with varying values of epsilon. We can see that as we decrease the value of epsilon, the run time (in seconds) increases to calculate β.

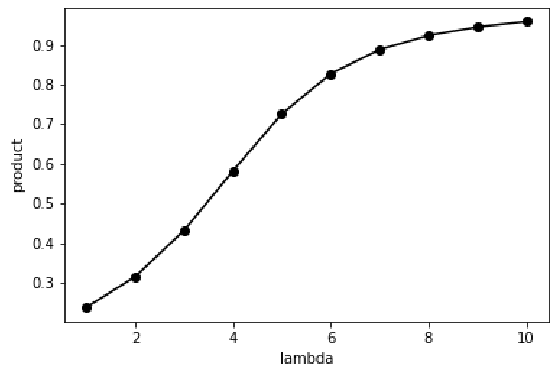


Figure 2

Vanilla OLS	S-OLS, eps = 0.1	S-OLS, eps = 0.05	S-OLS, eps = 0.01	S-OLS, eps = 0.001
0.297793	0.001001	0.001001	0.002001	0.028025

Table 2: Runtime in Seconds for Solving of Beta for Vanilla and Sketched OLS

Pledge:
Please sign below (print full name) after checking (✓) the following. If you can not honestly check each of these responses, please email me at kbala@ucdavis.edu to explain your situation.

- I pledge that I am an honest student with academic integrity and I have not cheated on this homework.
- These answers are my own work.
- I understand that to submit work that is not my own and pretend that it is mine is a violation of the UC Davis code of conduct and will be reported to Student Judicial Affairs.
- I understand that suspected misconduct on this homework will be reported to the Office of Student Support and Judicial Affairs and, if established, will result in disciplinary sanctions up through Dismissal from the University and a grade penalty up to a grade of "F" for the course.

Name: Gianni Spiga