Question 1. For each of the following, give an example or explain why no such example exists.

(a) A linear transformation $T : R^4 \to R^3$ that is onto.

In class we showed that T is "onto" iff $Im(T) = W$, when $T : V \to W$.
Here that would be $Im(T) = \mathbb{R}^3$.

Yes, there does exsit a transformation T that is "onto".
Let T be the projection...

$$T : \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \to \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

(b) A linear transformation $T : R^4 \to R^3$ that is one to one.

In class we showed that T is "one to one" iff $Ker(T) = \bar{0}$.

Let A be a matrix that represents T where $T : R^4 \to R^3$.
The dimensions of $Ax = b$ are as follows...

$$(3 \times 4)(4 \times 1) = (3 \times 1)$$

So then A is a (3×4) matrix. Becasue A has 3 rows, A can be at most rank 3. This means the nullspace is at least one dimensional. This is because the nullspace is $n - r$, where r is the rank.

> Nullspace: at least 1 dimensional.
> Left-nullsace: can contain only $\bar{0}$.
> Columnspace: at most 3
> Rowspace: at most 3

No, there does exsit a transformation T that is "one to one", becasue the $Ker(T)$, i.e. the nullspace, is at least one dimensional (a line), and does not contain only the zero vector. This is true no matter what linear transformation T we come up with.

This is also why T can be "onto" in the previous question. It's column space is equal to it's rank. Here, that can be at most 3. So when A is a rank 3 matrices, then in would represent an onto relation.

(c) A linear transformation $T : R^3 \to R^4$ that is onto.

No, there does exsit a transformation T that is "onto".
Let A be a matrix that represents T where $T : R^3 \to R^4$.
The dimensions of Ax = b are as follows...

$$(4 \times 3)(3 \times 1) = (4 \times 1)$$

So then T is a (4×3) matrix. It can be at most rank 3, and the dimensions of the column space, rowspace, and nullspaces are...

> Nullspace: can contain only $\overline{0}$.
> Left-nullsace: at least 1 dimensional.
> Columnspace: at most 3
> Rowspace: at most 3

For the $Im(T)$ to equal R^4, the rowspace would need to be four dimensional. But the rowspace is at most 3 dimnesional, and it cannot equal the 4 dimensional space R^4.

(d) A linear transformation $T : R^3 \to R^4$ that is one to one.

Yes, there does exsit a transformation T that is "one to one".
I can't cook a good one up, but when T is represented by a rank 3 matrices, then the nullspace only contains $\overline{0}$.
Heck, maybe the identity matrix with a spare column of all zeros?

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(e) For every $i = 0, 1, 2, 3, 4$, a linear transformation $T : R^4 \to R^4$ whose rank is i.

For $T : R^4 \to R^4$, $Ax = b$ would be $(4 \times 4)(4 \times 1) = (4 \times 1)$

When T is rank 1:

> Nullspace: 3
> Left-nullsace: 3
> Columnspace: 1
> Rowspace: 1

Cannot be onto. The image is one dimensional needs to be 4 dimensional.
Cannot be one to one the nullspace needs to only contain $\overline{0}$, but it is 3 dimensional.

When T is rank 2:

> Nullspace: 2
> Left-nullsace: 2
> Columnspace: 2
> Rowspace: 2

Cannot be onto. The image is 2 dimensional needs to be 4 dimensional.
Cannot be one to one the nullspace needs to only contain $\overline{0}$, but it is 2 dimensional.

When T is rank 3:

> Nullspace: 1
> Left-nullsace: 1
> Columnspace: at most 3
> Rowspace: at most 3

Cannot be onto. The image is 3 dimensional needs to be 4 dimensional.
Cannot be one to one the nullspace needs to only contain $\bar{0}$, but it is 1 dimensional.

When T is rank 4:

> Nullspace: $\bar{0}$
> Left-nullsace: $\bar{0}$
> Columnspace: 4
> Rowspace: 4

Can be "onto". The image is 4 dimensional and it needs to be 4 dimensional.
Can be "one to one". The nullspace needs to only contain $\bar{0}$, and it only contains $\bar{0}$.

Example: Let T be represented by the multiplication of the identity matrix.

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This sends elements in R^4 to themselves. It is "one to one" and "onto".

Question 2. Every year, 1% in town A move to town B, and 2% of the people in town B move to town A. In 2015, there are 1000 people living in each town. Find a matrix M such that $M^n(1000, 1000)$ gives the number of people living in each town after n years.

Let the vector.. $$

$$\begin{pmatrix} 1000 \\ 1000 \end{pmatrix}$$

$$\begin{pmatrix} \text{population A in } 2015 \\ \text{population B in } 2015 \end{pmatrix}$$

$$Then the matrix \$M\$, shown bellow, such that \$M^n * (1000, 1000)\$ gives the number of pe$$

M =

$$\begin{pmatrix} -1.01 & 1.02 \\ 1.01 & -1.02 \end{pmatrix}$$

$$

Question 3. Suppose $J \in M_{n,n}(R)$ is a matrix satisfying $J_{i,j} = 1$ if $j - i = 1$, and 0 otherwise. Calculate J_k for every $k \in \mathbb{N}$.

Matrix J is an $(n \times n)$ matrix with a diagnol of 1's runnning just above its center diagnol. The (4×4) is given below.

$$J = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

As a test, I continue to square J and look for any trends. I can see that it aprouches a matrices of 0's after J^n where n is the number of columns and rows. So for the (4×4) case, J becomes a matrices of 0's at $n = 4$, that is J^4.

$$J^4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

This gave me some insoght as to what J does in terms of elementary operations, and how J can be acheaved from the Identity matrix. That is, a permutation where the first row is moved to the last row (then multiplied by 0), and the remaning $n - 1$ rows are moved upwards once. So using the permutation notation we learned in class recently, that would be $(2, 3, 4, \ldots, n, 1)$. Again, the first row goes to the last position (n), and all other rows are bumped up once.

Summary...
When $k \geq n$, J^k is a matrices of 0's.

When $k \leq n$, J^k will be a matrix of 0's and 1's with the one's running along the, $k+1$ column diagnol.

Thinking about the simplest case where n = 1, we have $J = (0)$. Here $k \geq n$ for all values of $k \in \mathbb{N}$, and J^k is a matrices containing only the element 0.

Question 4. Consider the following method of encrypting information: Take a text, and break it up into chunks of four characters. So ”linear algebra” turns into ”line” ”ar a” ”lgeb” ”ra ”. Convert each chunk into numbers by taking numbers $1 - 26$ for the letters a-z, numbers $27 - 36$ for the digits $0 - 9$ and number 0 for the space character. Think of these numbers as elements of F_{37} , and turn them into vectors. So ”abcd” would turn into $(1, 2, 3, 4)$. Multiply the vectors by a matrix...

$$key = \begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 3 & 4 \\ 5 & 4 & 3 & 2 \end{pmatrix}$$

Then convert the numbers back to letters. Remember, the multiplication happens in the field F_{37}.

Decode the following message: ”3sx ol0o”. How would you decode a message in general?

First I sumarize encoding a message as described in the questin.

> (1) break message in to 4 character chunks
> (2) convert the letters within each chunk to numbers in F_{37}
> (3) treat each chunk as a 4 element vector, and multiply that vector by the key (shown above).
> (4) convert this new encripted 4 element vector in F_{37} back to letters.

To decode an encripted message, I would do those steps in reverse, taking care to multiply by the inverse of the key, not the key it's self. To put on shoes, we put on socks, then put on shoes. To take off shoes, we take off shoes, then take off socks. Becasue this method is nearly semetric, the steps to decode a message are nearly identicle to encoding a message, only the inverse of the key is used. That is...

$$key^{-1} = \begin{pmatrix} 1 & 1 & 6 & 6 \\ 36 & 18 & 3 & 22 \\ 36 & 35 & 25 & 25 \\ 1 & 20 & 34 & 15 \end{pmatrix}$$

I found the inverse key using the pyhtonj function `key.inv_mod(37)`. So to decode ”3sx ol0o” I would...

> (1) break message in to 4 character chunks
> (2) convert the letters within each chunk to numbers in F_{37}
> (3) treat each chunk as a 4 element vector, and multiply that vector by the inverse of the key (shown above).
> (4) convert this new encripted 4 element vector in F_{37} back to letters.
> (5) Paste the 4 character chunks back into one character string

Once again I did this problem in R. Below is the code I used. I will describe what the code does, then I will apply it to the encripted message ”3sx ol0o”. The results are gobble-de-gook, "hnnettl2", so I encripted my own test message, then decoded the test message. The script works on my message, so either the answer is actually gobble-de-gook, or I'm doing something wrong.

Anyhow, here's the code...

In [ ]:
```
lock = function(message, key){
  # turn characters into numbers over field Z/37Z
  elem = as.character(c(" ", letters, as.character(0:9)))
  nums = vector()
  for(i in 1:length(message)) nums[i] = (which(elem == message[i]) - 1) %% 37

  # splits code into 4 letter chunks
  idx = rep(1:4, each = 4)
  idx = idx[1:length(nums)]
  chunks = split(nums, idx)

  # function for scramble/unscramble with the key
  scramble = function(bit, key) {
    input = (key %*% bit) %% 37
    input = input[,1]
    input
  }

  # function to convert back to letters
  lets = function(input, elem) elem[input + 1]

  chunks = lapply(chunks, FUN = scramble, key = key)
  chunks = lapply(chunks, FUN = lets, elem = elem)

  # output character string
  as.character(unlist(chunks))
  }

key = matrix(c(2,0,1,0,
               0,1,0,1,
               1,2,3,4,
               5,4,3,2),
             byrow = TRUE,
             nrow = 4,
             ncol = 4)

# cant seem to get a good function for finding the inverse in Z/nZ for R so
# I did it in python and just wrote the inv matrix in manually
inv_key = matrix(c( 1,  1,  6,  6,
                   36, 18,  3, 22,
```

```
                  36, 35, 25, 25,
                   1, 20, 34, 15),
                byrow = TRUE,
                nrow = 4,
                ncol = 4)
```

Now I take the function, and apply it to the code ”3sx ol0o”. I name the encoded message scramble, and I name the the decoded message eggs.

In [ ]:
```
scramble = c("3", "s", "x", " ", "o", "l", "0", "o")
eggs = lock(message = scramble, key = inv_key)
print(eggs)
```

output: "h" "n" "n" "e" "t" "t" "l" "2"

Again, this is not a word so I'm afraid I did something wrong. I test my script on a know message "testcode". I am able to both endoce the message and decode the message.

In [ ]:
```
# That's not much of a message, so I test my script on a known message
eggs = c("t", "e", "s", "t", "c", "o", "d", "e")
scramble = lock(message = eggs, key = key)
print(scramble)
```

output: "v" "y" "s" "5" "j" "t" "1" "w"

In [ ]:
```
eggs = lock(message = scramble, key = inv_key)
print(eggs)
```

output: "t" "e" "s" "t" "c" "o" "d" "e"

Okay, I am able to retrieve my original message back. I wrote the script so that it can encode and decode messages of any length. This added a bit of complexity to the script, but it shows how to encode and decode messages generaly. Given more time, I would try coding messages with different keys.

Question 5. Let $T : V \to W$, $S : W \to U$ be linear transformations. Show that $rank(ST) \leq rankS$ and $rankST \leq rankT$. Now suppose that $U = V = W$ and that B is invertible. Show that $rank(AB) = rankA$.

Here are three very informal proofs. So informal that they are not proofs.

(a) Show that $rank(ST) \leq rank(S)$.

Let the linear transformations S and T be represented by the matrices A and B respectivly. Then ST is represented by the matrices multiplication of $A \times B$, where B is performing column operations on A.
Here, B is the linear transformation T. So the basis in the product of $A \times B$ will be linear combinations of the basis of A.
If multiplying A by B would increase the number of basis in A, then B could not be a linear transformation.

On the other hand, B can decrease the number of basis in A, and still maintian it's statis as a linear transformantion.
Therefore the number of elements in the basis of $A \times B$ cannot be greater than the number of elements in the basis of A.
In other words $rank(AB) \leq rank(A)$.
In other, other words $rank(ST) \leq rank(S)$.

(b) Show that $rank(ST) \leq rankT$.

Same as before with slight variation.
Let the linear transformations S and T be represented by the matrices A and B respectivly.
Then ST is represented by the matrices multiplication of $A \times B$, where A is performing row operations on B.
Here, A is the linear transformation S. So the basis in the product of $A \times B$ will be linear combinations of the basis of B.
If multiplying A by B would increase the number of basis in B, then A could not be a linear transformation.
On the other hand, A can decrease the number of basis in B, and still maintian it's statis as a linear transformantion.
Therefore the number of elements in the basis of $A \times B$ cannot be greater than the number of elements in the basis of B.
In other words $rank(AB) \leq rank(B)$.
In other, other words $rank(ST) \leq rank(T)$.

(c) suppose that $U = V = W$ and that B is invertible. Show that $rank(AB) = rankA$.

Similar to before, AB can only produce different basis than A, but it must obtain the same or less than the number of basis in A.
But, when B is invertible, then all if it's columns and rows represent individual basis.
Furthermore, it has no rows or columns of all 0's, or dependent columns/rows (again, everything is a basis).
With that, it cannot decrease the number of basis in A.
So B can only change the basis direction, but not decrease or increase the elements in the basis of A.
Therefore $rank(AB) = rankA$.

Question 6. In the city of Oddsville there were n citizens. The citizens were obsessed with participating in various clubs. The mayor of Oddsville felt that the citizen's passion for forming clubs was getting out of hand. As a result, she 12 instituted two strange rules to curb the number of clubs. The first rule is that each club must have an odd number of members. The second rule is that every two different clubs must have an even number of members in common. Let m be the number of clubs in Oddsville. The mayor knew that with these rules, there couldn't be more clubs than there were people, that is: m ≤ n, so these rules would prevent an excessive amount of clubs from forming. In this problem you'll figure out why.

(a) Name the citizens 1, 2, 3, . . . , n and the clubs C 1 , . . . , C m . Let A be the m × n matrix with coefficients in F 2 , the field with 2 elements defined by: A i,j = 1 if citizen j is in the club C i and 0 otherwise. Make a sample town with five citizens and three clubs. Assign each citizen to as many clubs as you want and make the matrix A corresponding to your example. Make an assignment of citizens to clubs that satisfies the mayor's rules, and one that doesn't.

Below is a matrix A that works...

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Below is a matrix A that does not work.

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

There is an even number of people in the club of row 2...

(b) Recall that A T denotes the transpose matrix of A. So A T is the n × m matrix with coefficients in F 2 such that A i,j = 1 if citizen i is in club C j and 0 otherwise. Write down the matrix A T for the example you gave in the previous part.

$$A^T = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(c) Let P B = AA . This is an m × m matrix whose (i, j) coordinate is A i,k A kj, . Go over the formula for matrix multiplication and make k=1 n sure that you understand how B i,j was calculated.

$$B = A^T A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix} = B \text{ not in } F_2$$

$$B \text{ in } F_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I$$

I get how B is calculated. It's the dot product of every column in A^T with every row in A. The dot product represents how many member each club has in common with the other. That is, when the vectors are orthogonal,the dot product is 0, meaning the clubs have no members in common.

So when B is not in F_2, but say R, then each element in B represents how many people the two clubs have in common. So the diagonal is the total people in each club, and the off diagnols are the total number of their related members.

(d) Explain why the mayor’s rules imply that B = I m .

Elements of B represent the total number of members that are shared between clubs C_i and C_j where i and j are the row and column numbers respectivly. So the diagnols, $C_{1,1}, C_{2,2}, C_{3,3}$ represent the total members in clubs C_1, C_2, C_3. The off diagnos represent how many members the clubs have in common with eachother.

The diagnols of B will then be odd. This is because of rule 1. All clubs must have an odd number of members.
The off-diagnols of B will be even. This is caused by rule two, every two different clubs must have an even number of members in common.

When B is in F_2, the even numbers are sent to 0 and the odd numbers are sent to 1. So the diagnol is all 1's, and the off diagnos are all 0's. That's the identity matrix.

(e) Explain why this means that Rank(B) = m.

B is an $(m \times m)$ matrix created from the multiplication of a $(m \times n)$ and $(n \times m)$ matrix. As described in question (6d), B is the identity matrix. So it is inverstible. It has a number of unique basis equal to it's number of rows, m. So be is rank m.

(f) Show that Rank(A) ≥ m (Hint: use problem 5).

As described in (e), the multiplication of $A^T A$ produces the Identity matrix of rank m, with m basis. In $A^T A$ the matrix A^T is performing row operations on the matrix A. So the product cannot have more unique basis than A, because all basis in the product will be linear combinations of basis in A. We know that the product of $A^T A$ has m basis, so $rank(A) \geq m$

(g) Explain why Rank(A) ≤ n (Hint: it has n columns).

A is a $(m \times n)$ matrix. In (f) we explain that $rank(A) \geq m$. A has n columns, and it cannot contain more basis than it does columns. So $rank(A) \leq n$.

(h) Deduce that m ≤ n, as required.

In (f) we showed $rank(A) \geq m$. In (g) we showed $rank(A) \leq n$.
So then $m \leq rank(A) \leq n$. Therefore $m \leq n$.