5. Limits in R(p,q)

In [10]:

```
err = 1  #the magnitude of difference in succesive terms
t1 = 2  #evaluating analytically at n=1 gives us 2 for the expression
t2 = 0  #to store the succeeding term
eps = 1e-18  #the epsilon used to test convergence

n = 1

print('Table of intermediate values:\n')

while(1):
    t2 = (1 + 1/(10**n))**(10**n)

    print(n,t2)

    if(abs(t2-t1) < eps): break

    t1 = t2
    n += 1

print('\nn_stop:',n-1)
print('Value converged to:', t2)
```

```
Table of intermediate values:

1 2.5937424601000023
2 2.7048138294215285
3 2.7169239322355936
4 2.7181459268249255
5 2.7182682371922975
6 2.7182804690957534
7 2.7182816941320818
8 2.7182817983473577
9 2.7182820520115603
10 2.7182820532347876
11 2.71828205335711
12 2.7185234960372378
13 2.716110034086901
14 2.716110034087023
15 3.035035206549262
16 1.0
17 1.0

n_stop: 16
Value converged to: 1.0
```

We find that contrary to intuition, the value of e computed using the above algorithm converges to 1.0 after 16 steps. This is because of the limited mantissa of 52 bits.

As a result, the smallest value that can be stored is 2^{-52}. Now, when we are computing (1+d) where d<<1, the d fills up the mantissa and the 1 causes the exponent to be 2^{0}.

So, the mantissa becomes filled with zeros in all the 52 bits when the number goes down to 2^{-53} which in base 10 is given by solving the equation: $10^{-n} = 2^{-53}$. This gives n = 53* log(2)/log(10) = 15.954. So the integer value of n = 16 from which the mantissa would be zeros. Therefore 1+d would essentially be 1 due to the inability of mantissa to hold smaller values of d. This is exactly the value of n where the above table of values comverges to 1.0.