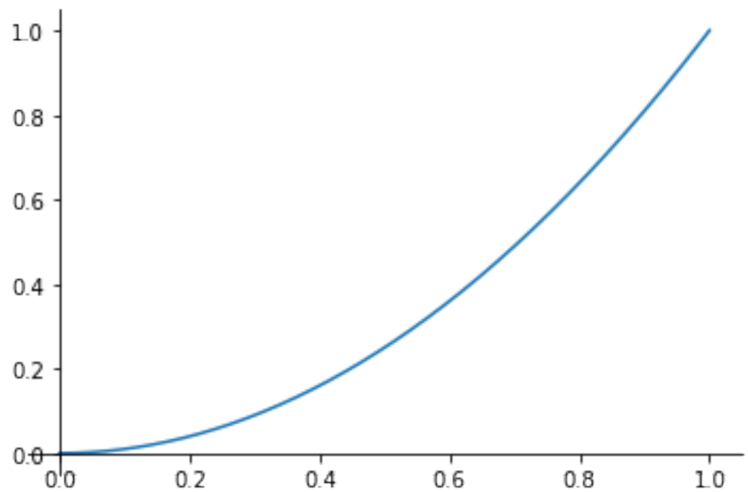
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 1, 1000)
y = x**2

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
```

Out[2]:
```
[<matplotlib.lines.Line2D at 0x12028ff10>]
```



The function graphed above is $y = x^2$. It is graphed on the interval $[0, 1]$. The theme we are going to pursue is that if we graph this function on the interval $[a, b]$ and measure the area under the graph, then it is computed as follows. We compute the anti-derivative and obtain $\frac{x^3}{3} + C$, where C is an arbitrary constant. We then evaluate this anti-derivative at b and subtract its value at a, obtaining

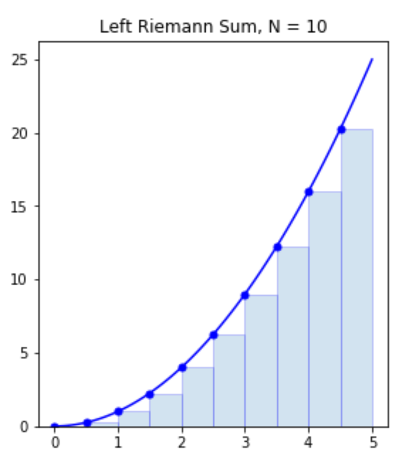
$$\frac{b^3}{3} + C - \left(\frac{a^3}{3} + C\right) = \frac{b^3}{3} - \frac{a^3}{3}.$$

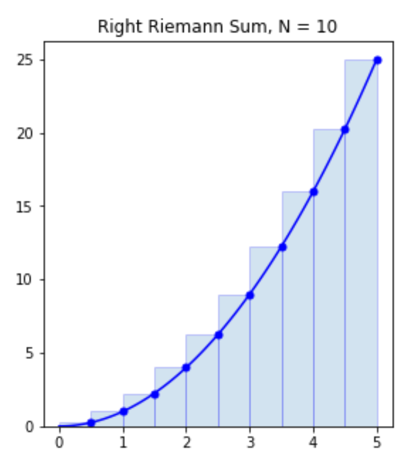
Notice that that once we considered the endpoints of the interval, the value of the arbitrary constant C did not matter.

It is going to take a bit of work to build up to the idea described above, so let's get to work.

In [4]:
```
f = lambda x : x**2
a = 0; b = 5; N = 10
n = 10 # Use n*N+1 points to plot the function smoothly

x = np.linspace(a,b,N+1)
y = f(x)

X = np.linspace(a,b,n*N+1)
Y = f(X)

plt.figure(figsize=(15,5))

plt.subplot(1,3,1)
plt.plot(X,Y,'b')
x_left = x[:-1] # Left endpoints
y_left = y[:-1]
plt.plot(x_left,y_left,'b.',markersize=10)
plt.bar(x_left,y_left,width=(b-a)/N,alpha=0.2,align='edge',edgecolor='
b')
plt.title('Left Riemann Sum, N = {}'.format(N))

plt.subplot(1,3,3)
plt.plot(X,Y,'b')
x_right = x[1:] # Left endpoints
y_right = y[1:]
plt.plot(x_right,y_right,'b.',markersize=10)
plt.bar(x_right,y_right,width=-(b-a)/N,alpha=0.2,align='edge',edgecolo
r='b')
plt.title('Right Riemann Sum, N = {}'.format(N))
```

Out[4]:
```
Text(0.5, 1.0, 'Right Riemann Sum, N = 10')
```



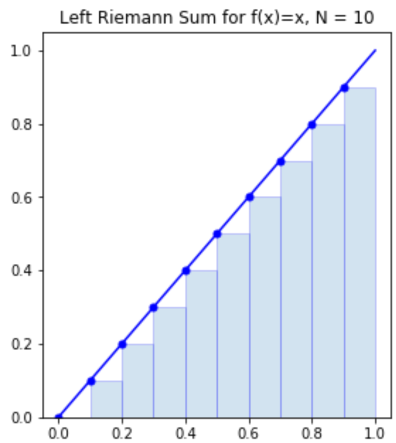


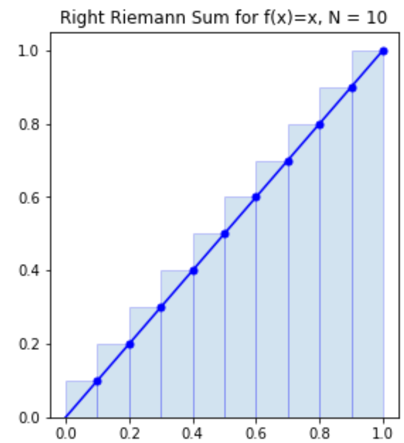
The first sum above is called a "left" Riemann sum because the height of the approximating rectangle is determined by the left endpoints. Correspondingly, the height of the approximating rectangle in the "right" Riemann sum above is determined by the right endpoints.

Let us now consider an even more basic example where we will be able to compute everything.

In [6]:
```
f = lambda x : x
a = 0; b = 1; N = 10
n = 10 # Use n*N+1 points to plot the function smoothly

x = np.linspace(a,b,N+1)
y = f(x)

X = np.linspace(a,b,n*N+1)
Y = f(X)

plt.figure(figsize=(15,5))

plt.subplot(1,3,1)
plt.plot(X,Y,'b')
x_left = x[:-1] # Left endpoints
y_left = y[:-1]
plt.plot(x_left,y_left,'b.',markersize=10)
plt.bar(x_left,y_left,width=(b-a)/N,alpha=0.2,align='edge',edgecolor='
b')
plt.title('Left Riemann Sum for f(x)=x, N = {}'.format(N))

plt.subplot(1,3,3)
plt.plot(X,Y,'b')
x_right = x[1:] # Left endpoints
y_right = y[1:]
plt.plot(x_right,y_right,'b.',markersize=10)
plt.bar(x_right,y_right,width=-(b-a)/N,alpha=0.2,align='edge',edgecolo
r='b')
plt.title('Right Riemann Sum for f(x)=x, N = {}'.format(N))
```

Out[6]: `Text(0.5, 1.0, 'Right Riemann Sum for f(x)=x, N = 10')`





What is great about this example is that we know how to compute the area of a rectangle, so we can check our work against reality pretty easily. In order to connect the computation of areas to anti-derivaties, divide the interval $[a, b]$ up as follows. Consider the points

$$\left[a + \frac{i(b-a)}{N}\right].$$

Note that when $i = 0$, we get a. When $i = N$, we get b. The rest of the points are evenly spread out in between.

Let's use the right Riemann sums. The area of the i'th rectange, $i = 1, 2, \ldots, N$, is

$$\frac{b-a}{N} \cdot \left(a + \frac{i(b-a)}{N}\right)$$
$$= \frac{(b-a)a}{N} + \frac{i(b-a)^2}{N^2}.$$

We must now sum up those areas. We get

$$\sum_{i=1}^{N} \frac{(b-a)a}{N} + \frac{i(b-a)^2}{N^2}$$
$$= \sum_{i=1}^{N} \frac{(b-a)a}{N} + \sum_{i=1}^{N} \frac{i(b-a)^2}{N^2}$$
$$= (b-a)a + \frac{(b-a)^2}{N^2} \sum_{i=1}^{N} i.$$

In order to take this further, we must compute the sum
$$1 + 2 + \cdots + N.$$

There are many ways to do this, but here is my favorite approach. Consider an N by N grid and notice that it has N^2 points. Color the diagonal ORANGE. Color everything above the diagonal RED and everything below the diagonal BLUE. Notice that
$$ORANGE = N\$, i, ethereare\$N\$orangepoints. Nowobservethat$$

RED=BLUE\$\$ by symmetry.

Since the total number of points is N^2, we have
$$N^2 = RED + ORANGE + BLUE = N + 2 \cdot BLUE,$$

so
$$BLUE = \frac{N^2 - N}{2}.$$

But what is BLUE? It is
$$1 + 2 + 3 + \cdots + (N - 1).$$

If we throw in ORANGE, we have
$$BLUE + ORANGE = 1 + 2 + \cdots + N,$$

which is what we need.

It follows that
$$1 + 2 + \cdots + N = \frac{N^2 - N}{2} + N = \frac{N(N+1)}{2}.$$

Plugging this into the Riemann sum above, we see that the total area of the Riemann sum rectangles is

$$(b-a)a + \frac{(b-a)^2}{N^2}\sum_{i=1}^{N} i$$

$$\begin{aligned} &= (b-a)a + \frac{(b-a)^2}{N^2}\cdot\frac{N(N+1)}{2} \\ &= (b-a)a + (b-a)^2\cdot\frac{N+1}{2N} \\ &= (b-a)a + (b-a)^2\cdot\left(\frac{1}{2}+\frac{1}{2N}\right), \end{aligned}$$

which tends to

$$(b-a)a + \frac{(b-a)^2}{2} = \frac{b^2}{2} - \frac{a^2}{2}$$

as N tends to ∞.

We have just shown that the area under the graph of the function $f(x) = x$ between $x = a$ and $x = b$ is

$$\frac{b^2}{2} - \frac{a^2}{2}.$$

Note that what this calculation suggests is that we could have computed the anti-derivative to $f(x) = x$, which is $\frac{x^2}{2}$, evaluated it at $x = b$, and then subtracted the value at $x = a$.

We could have computed the area underneath the graph under the graph of $f(x) = x$ between $x = a$ and $x = b$ by noticing that it is equal to the area of the triangle of height b and length b minus the area of a triangle of length a and height a. This would give us $\frac{b^2}{2} - \frac{a^2}{2}$, which brings up an interesting question. Why did we do through all that suffering above if the answer can be easily deduced from high school geometry. We did it because the method of using Riemann sums and taking limits applies in many situations where the area cannot be computed using simple tricks.

We are now going to practice these skills in the case $f(x) = x^2$ on the interval $[a, b]$ and we are going to justify the formula we claimed in the beginning of this lecture. Using right Riemann sums and breaking up the interval $[a, b]$ as above, we see that the total area of the rectangles is

$$\sum_{i=1}^{N} \frac{b-a}{N} \cdot \left(a + \frac{i(b-a)}{N}\right)^2$$
$$= \sum_{i=1}^{N} \frac{b-a}{N} \cdot \left(a^2 + \frac{i^2(b-a)^2}{N^2} + \frac{2a(b-a)i}{N}\right)$$
$$= \sum_{i=1}^{N} \frac{a^2(b-a)}{N} + \frac{(b-a)^3}{N^3} \sum_{i=1}^{N} i^2 + \frac{2a(b-a)^2}{N^2} \sum_{i=1}^{N} i$$
$$= a^2(b-a) + \frac{(b-a)^3}{N^3} \sum_{i=1}^{N} i^2 + \frac{2a(b-a)^2}{N^2} \sum_{i=1}^{N} i$$
$$= a^2(b-a) + \frac{2a(b-a)^2}{N^2} \cdot \frac{N(N+1)}{2} + \frac{(b-a)^3}{N^3} \sum_{i=1}^{N} i^2,$$

where we used the formula we derived above, namely

$$\sum_{i=1}^{N} i = \frac{N(N+1)}{2}.$$

In order to complete our calculation, we need to compute

$$\sum_{i=1}^{N} i^2.$$

Observe that

$$i^2 = \sum_{k=1}^{i} k^2 - (k-1)^2 = \sum_{k=1}^{i} 2k - 1.$$

Do you see why?

With this in tow, we get

$$\sum_{i=1}^{N} \sum_{k=1}^{i} 2k - 1.$$

We have

$$1 \leq k \leq i \leq N,$$

so if we want to sum in i first, we have

$$\sum_{k=1}^{N} 2k-1 \sum_{i=k}^{N} 1$$

$$= \sum_{k=1}^{N}(2k-1)(N-k+1)$$

$$= *(2N+3)\sum_{k=1}^{N} k - (N+1)\sum_{k=1}^{N} 1 - 2\sum_{k=1}^{N} k^2$$

$$= \frac{(2N+3)N(N+1)}{2} - N(N+1) - 2\sum_{k=1}^{N} k^2$$

$$= \frac{(2N+1)N(N+1)}{2} - 2\sum_{k=1}^{N} k^2.$$

It follows that

$$\sum_{k=1}^{N} k^2 = \frac{(2N+1)N(N+1)}{2} - 2\sum_{k=1}^{N} k^2,$$

so we conclude that

$$\sum_{k=1}^{N} k^2 = \frac{(2N+1)N(N+1)}{6}.$$

Plugging this in above, we see that the total area of the Riemann sum rectangles is equal to

$$a^2(b-a) + \frac{2a(b-a)^2}{N^2} \cdot \frac{N(N+1)}{2} + \frac{(b-a)^3}{N^3} \cdot \frac{(2N+1)N(N+1)}{6}.$$

Taking a limit as N goes to infinity, we obtain

$$a^2(b-a) + a(b-a)^2 + \frac{(b-a)^3}{3}$$

$$= \frac{b^3}{3} - \frac{a^3}{3}.$$

This gives us a basic overview of Riemann sums. You will get the chance to compute many particular examples on webwork and in the suggested homework. Here is a small bonus item. Let's compute a few sums do get a feel for the numerics.

One of the sums we computed above is $\sum_{k=1}^{n} k^2$ and obtained the formula

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

Let's do this numerically.

In [7]:
```
# We define a function that returns the sum of squares up to n.

def sumofsquares(n):
    c=0
    for k in range(1,n+1):
        c=c+k**2
    return c
```

To evaluate this function, just type: sumofsquares(your favorite number).

For example, if you type in

In [10]:
```
sumofsquares(5)
```

Out[10]: 55

It is also useful to practice coding double sums. For example, we have seen above that

$$\sum_{k=1}^{n} k^2 = \sum_{k=1}^{n} \sum_{i=1}^{k} i^2 - (i-1)^2$$

$$\sum_{k=1}^{n} k^2 = \sum_{k=1}^{n} \sum_{i=1}^{k} (2i-1).$$

In [12]:
```
# Defining a function which also computes the sum of squares, but this time from the standpoint of the double sum
# above.

def fancysumofsquares(n):
    c=0
    for k in range(1,n+1):
        for i in range(1,k+1):
            c=c+2*i-1
    return c
```

You should be able to convince yourself that this function is the same as the function sumofsquares above.

We also saw above that this double sum can be rewritten in the form

$$\sum_{i=1}^{n}\sum_{k=i}^{n}(2i-1).$$

Let us now write python code to implement the sum of squares from this point of view.

In [17]:
```
# Defining a function which computes the sum of squares from yet anoth
er point of view.

defanciersumofsquares(n):
    c=0
    for i in range(1,n+1):
        for k in range (i,n+1):
            c=c+2*i-1
    return c
```

Once again you should experiment enough to convince yourself that this function yields the same results as the functions above. This exercise will both help you with python, if this is your interest, and with orgaizing mathematical thoughts.

In [ ]: