The logistic model describes the growth of biological populations. The standard deterministic model is described by the ordinary differential equation

$$\frac{dP_t}{dt} = rP_t\left(1 - \frac{P_t}{K}\right)$$

where P_t denotes the population size at time t, r is the growth rate and K is the carrying capacity, the maximum population size that the environment can sustain.

The solution of the deterministic equation is $P_t = \frac{KP_0}{P_0+(K-P_0)e^{-rt}}$

A stochastic logistic equation is described the SDE

$$dP_t = rP_t\left(1 - \frac{P_t}{K}\right)dt + \sigma P_t dB_t$$

where $\sigma > 0$ is a parameter.

The solution to the logistic SDE is

$$P_t = \frac{P_0 K X_t}{K + P_0 r \int_0^t X_s ds}$$

where $X_t = e^{\left(r - \frac{\sigma^2}{2}\right)t + \sigma B_t}$,geometric brownian motion.

In [3]:
```
using Plots,Distributions
plotlyjs();
```

In [2]:
```
p0 = 2; r = 0.06; k = 300;
sigma_vals = [0.02,0.05,0.15];
```

In [4]:
```
function sol(t)
    num = k*p0
    den = p0 + (k- p0)*exp(-r*t)
    return num/den
end
```

Out[4]:
```
sol (generic function with 1 method)
```

In [5]:
```
tvals = range(start =0,stop=  200, length = 20000)
y = [sol(t) for t in tvals];
```

In [6]:
```
sigma = 0.02
Xt = [exp(r - sigma^2/2)*t + sigma*rand(Normal(0,sqrt(t)))
    for t in tvals]
numerator = p0*k*Xt;
```

cumsum(Xt[1:length(tvals)] .* (tvals[2:length(tvals)] .- tvals[1:length(tvals)])) # it gives dimension does not match error

In [7]:
```
n = 20_000
t = 200
```

```
x = rand(Normal(0,sqrt(t/n)),n)
bm = [0;cumsum(x)]
steps = range(start = 0,stop = t, length = n+1)
integral = cumsum(bm[1:n].* (steps[2:n+1] .- steps[1:n]))
```

Out[7]:

```
20000-element Vector{Float64}:
     0.0
     0.0007150772187182679
     0.00036007424481368863
    -0.0001469485261921814
    -0.00169877412205009
    -0.0035997842878361906
    -0.006052820372833492
    -0.007132409254204369
    -0.008113862204168416
    -0.008812131587017836
    -0.008455987964823828
    -0.008818829059129916
    -0.009164655893670584
     ⋮
 -1020.1362494528001
 -1020.2803223625731
 -1020.4242557717726
 -1020.5677170578697
 -1020.7100680207005
 -1020.8525894337878
 -1020.9961703416889
 -1021.1404093210344
 -1021.2862463683978
 -1021.4304185920092
 -1021.575711540347
 -1021.7201388789304
```

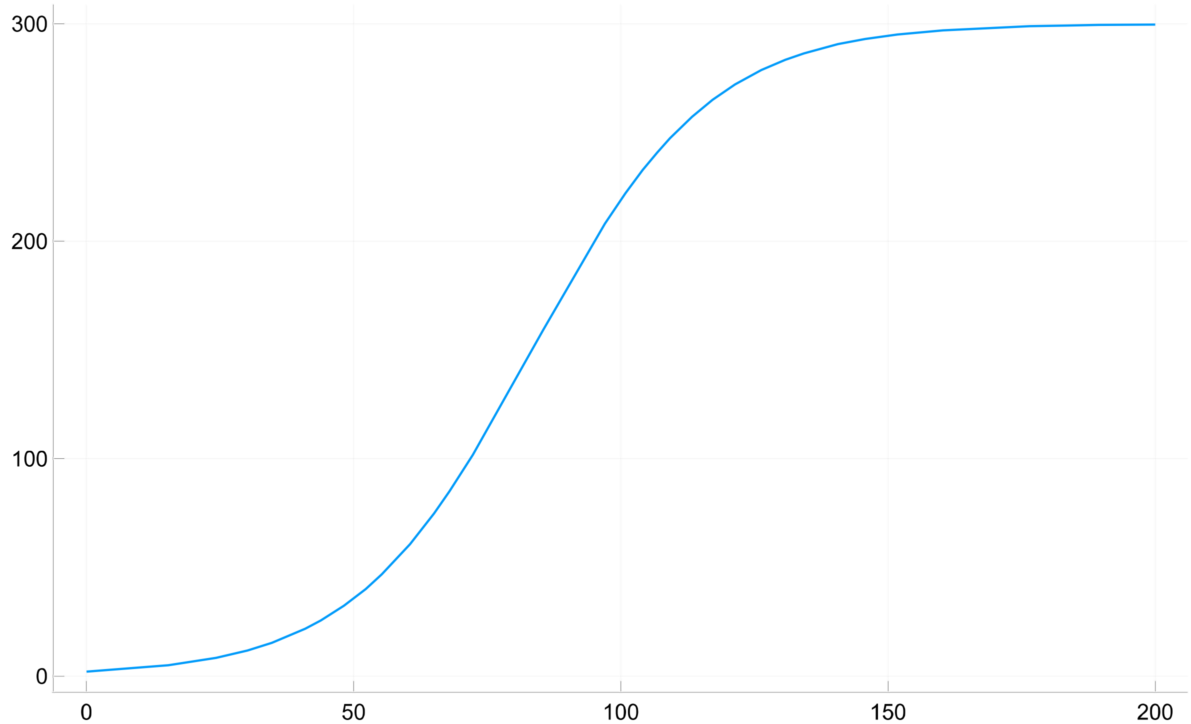
In [8]:

```
steps = range(start = 0, stop =0.01,length = 100);
```

In [9]:

```
plot(tvals,y,label =:none)
```

Out[9]:



Let B_t is a standard brownian motion. Let G_0 is a constant and $G_t = G_0 e^{\mu t + \sigma B_t}$. Then G_t is called geometric brownian motion. The expectation and variance of geometric brownian motion are:

- $E[G_t] = G_0 e^{t\left(\mu + \frac{\sigma^2}{2}\right)}$
- $Var[G_t] = G_0^2 e^{2t\left(\mu + \frac{\sigma^2}{2}\right)} \left(e^{\sigma^2 t} - 1\right)$

Computation of Ito Integral:

Compute the integral $\int_0^t B_s dB_s$. We compute this integral by the following way:

$$\int_0^t B_s dB_s = \lim_{n\to\infty} \sum_{k=1}^{n} B_{t_{k-1}} \left(B_{t_k} - B_{t_{k-1}}\right)$$

where we partition the interval $[0,t]$ into n many parts that is $0 = t_0 < t_1 < \cdots < t_n = t$. This sum will give us approximate integral to the Stochasric Integral. The exact integral will be $\frac{1}{2}\left(B_t{}^2 - t\right)$.

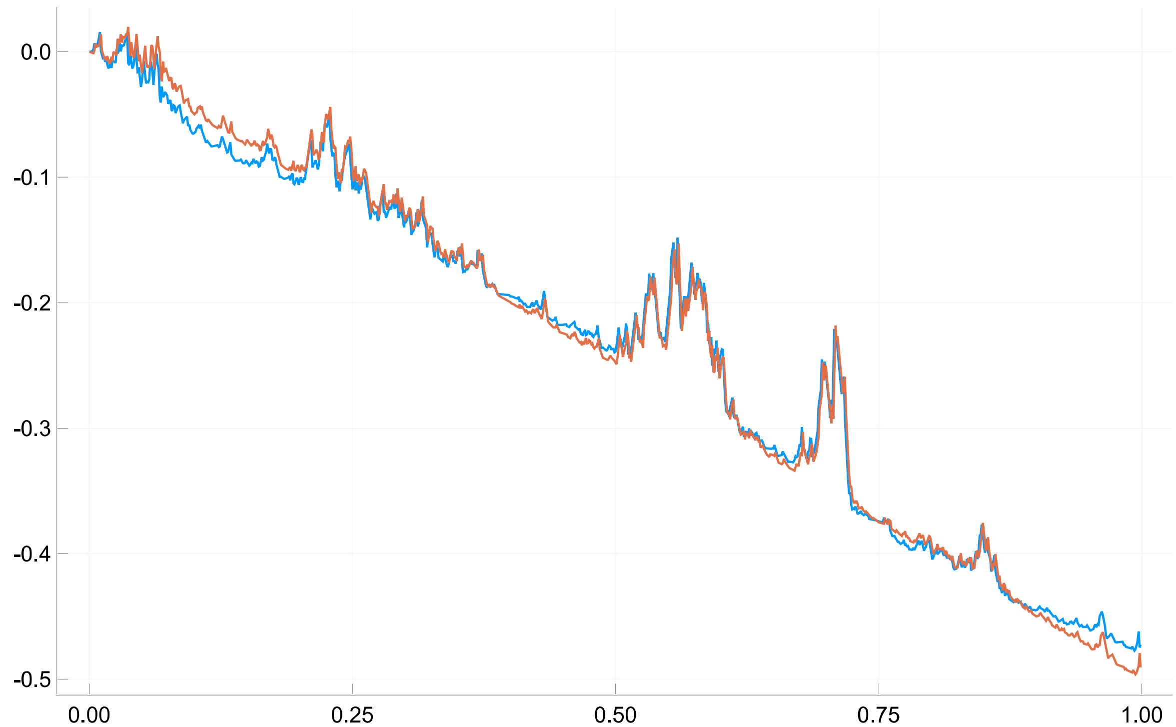
In [10]:
```
n = 1000
t = 1
x = rand(Normal(0,sqrt(t/n)),n)
bm = [0;cumsum(x)]
steps = range(start = 0,stop = t, length = n+1)
```

Out[10]:
```
0.0:0.001:1.0
```

In [11]:
```
int = cumsum(bm[1:n].* (bm[2:n+1] .- bm[1:n]));
```

In [12]:
```
plot(steps[1:n],int,label =:none) #,label="approx_integral",lw = 2,fg_legend = :none)
plot!(steps[1:n],(0.5*(bm.^2 - steps))[1:n],label =:none)
```

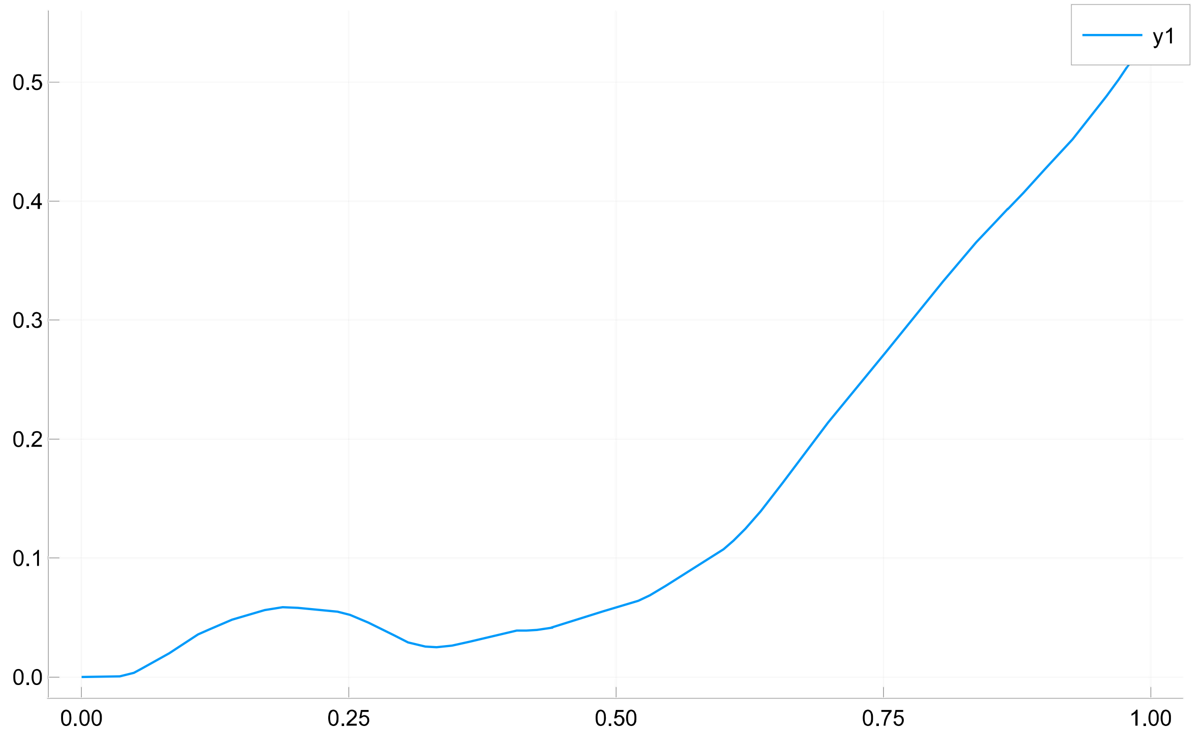
Out[12]:



Consider the integral $\int_0^t B_s ds$.

In [13]:
```
n = 10_00_000
t = 1
x = rand(Normal(0,sqrt(t/n)),n)
bm = [0;cumsum(x)]
steps = range(start = 0,stop = t, length = n+1)
integral = cumsum(bm[1:n].* (steps[2:n+1] .- steps[1:n]))
plot(steps[1:n],integral)
```

Out[13]:



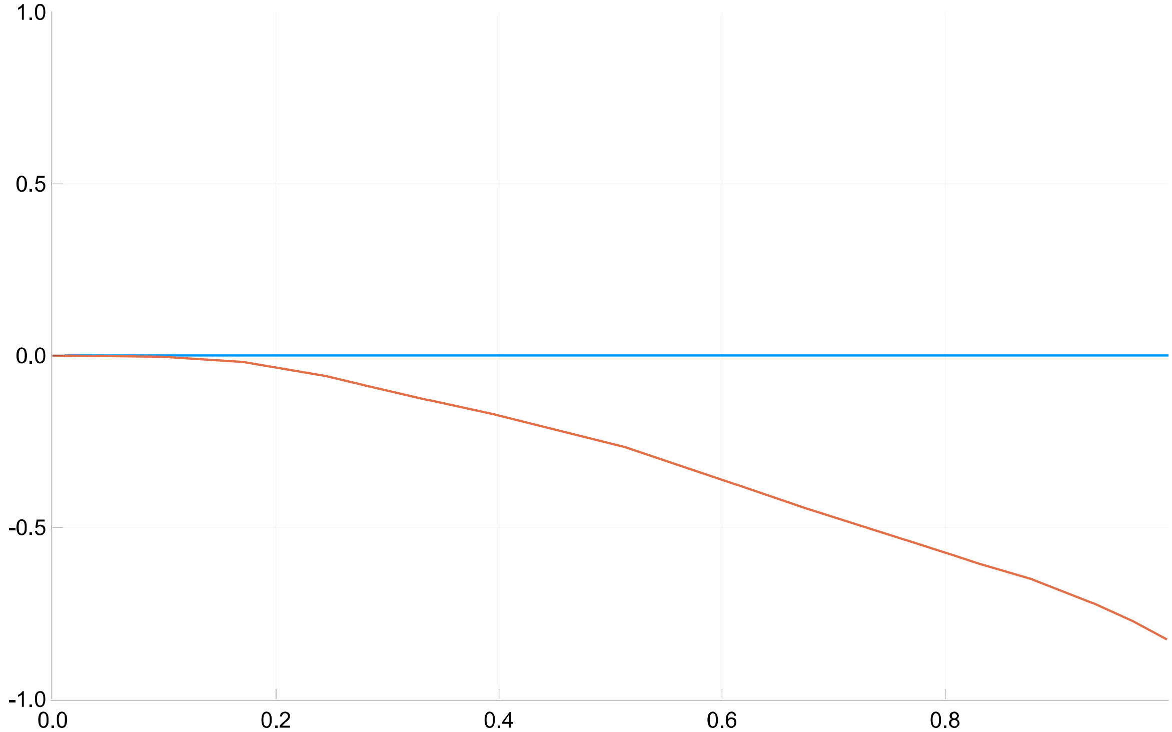
In [14]:

```
mean(integral)
```

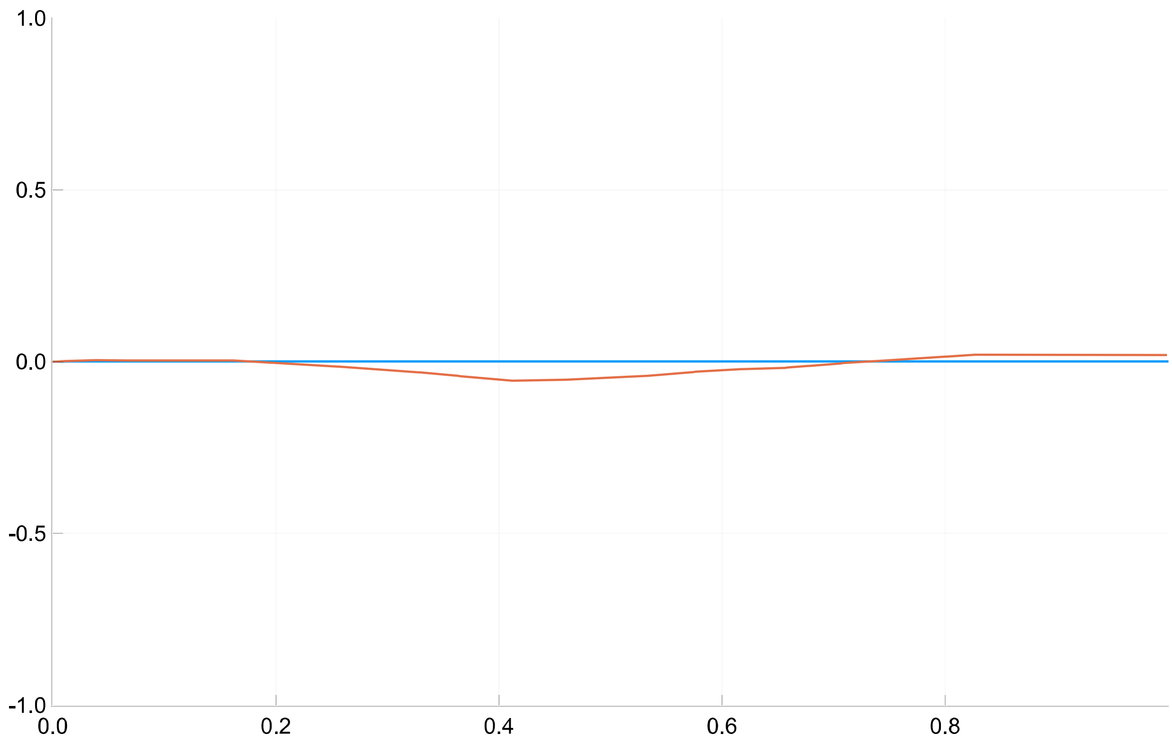
Out[14]: 0.15488185830666562

In [24]:

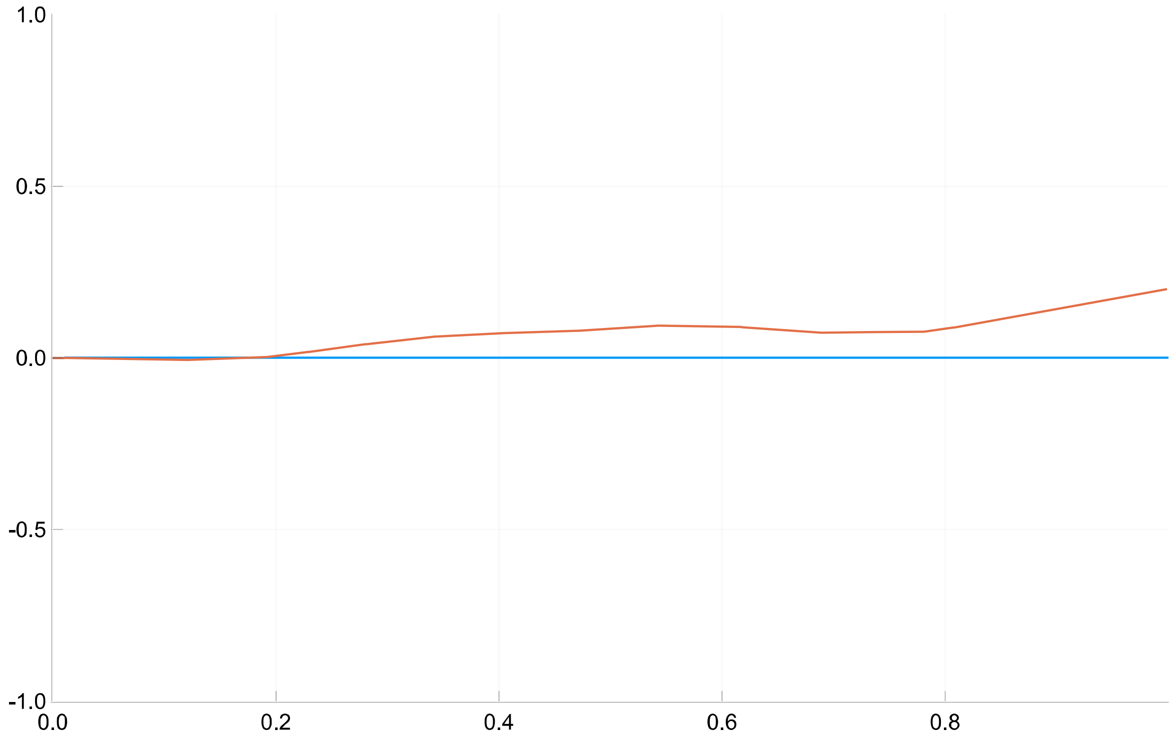
```
for i in range(start = 1,step =1,stop=9)
    p = plot(xlims = (0,1),label = :none,ylims = (-1.0,1.0))
    hline!([0],label =:none)
    n = 1000
    t = 1
    x = rand(Normal(0,sqrt(t/n)),n)
    bm = [0;cumsum(x)]
    steps = range(start = 0,stop = t, length = n+1)
    integral = cumsum(bm[1:n].* (steps[2:n+1] .- steps[1:n]))
    p = plot!(steps[1:n],integral,label = :none)
    display(p)
    println("Area : ", mean(integral))
end
```



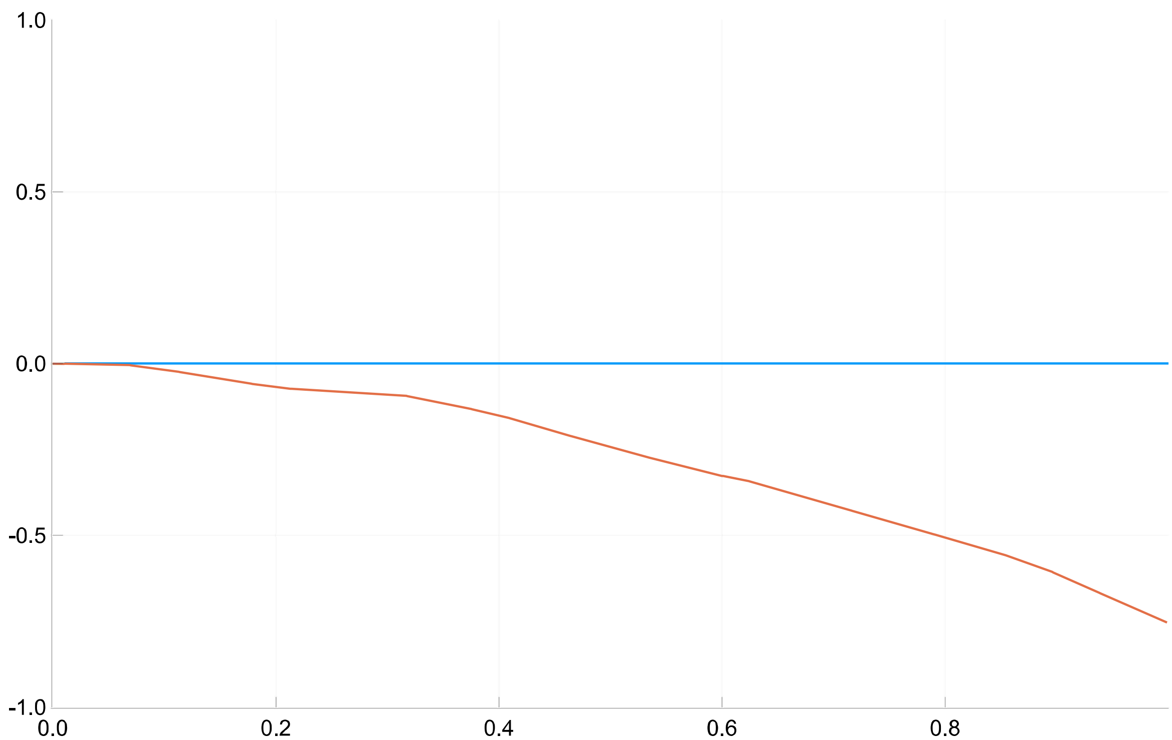

```
Area : -0.3044446120029065
```



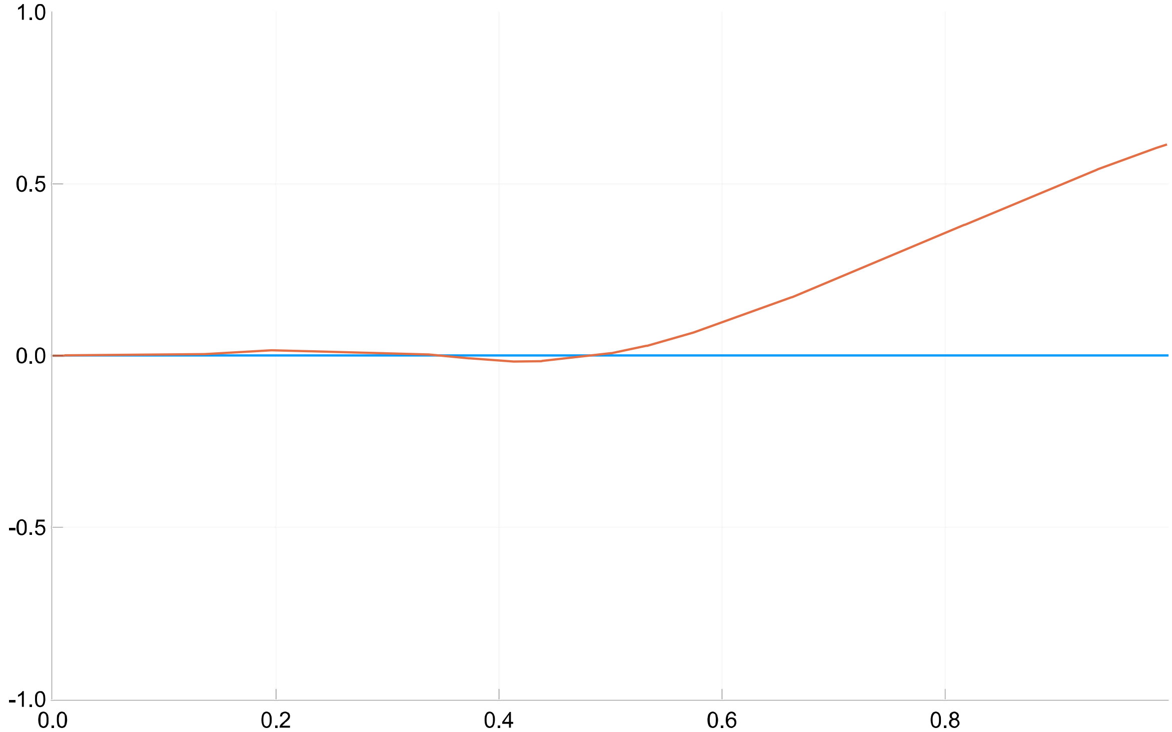

```
Area : -0.011310643774664415
```



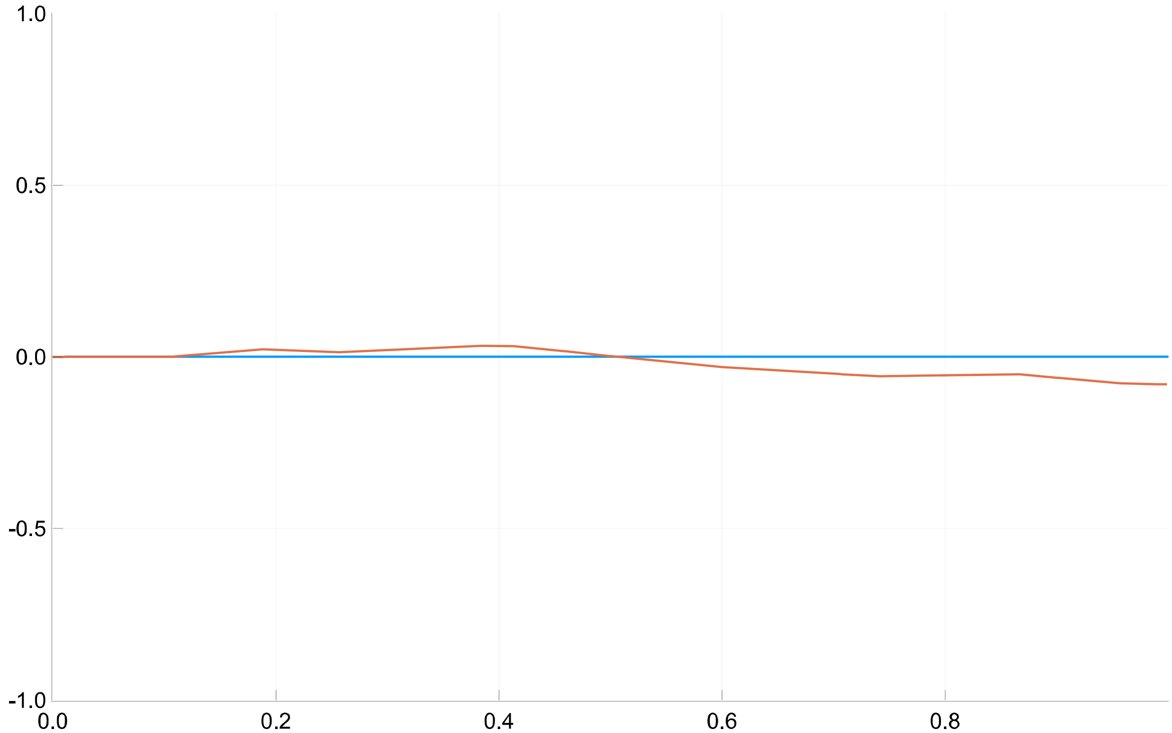
Area : 0.06922130678983564



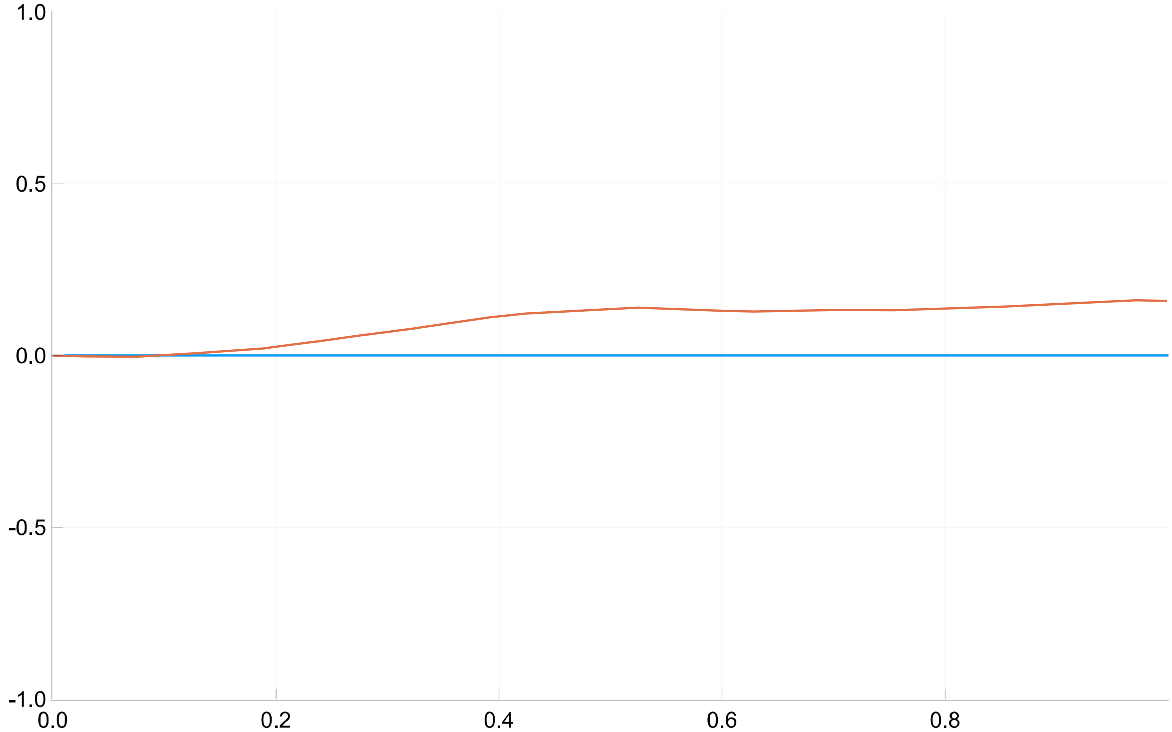
Area : -0.2791050927104301



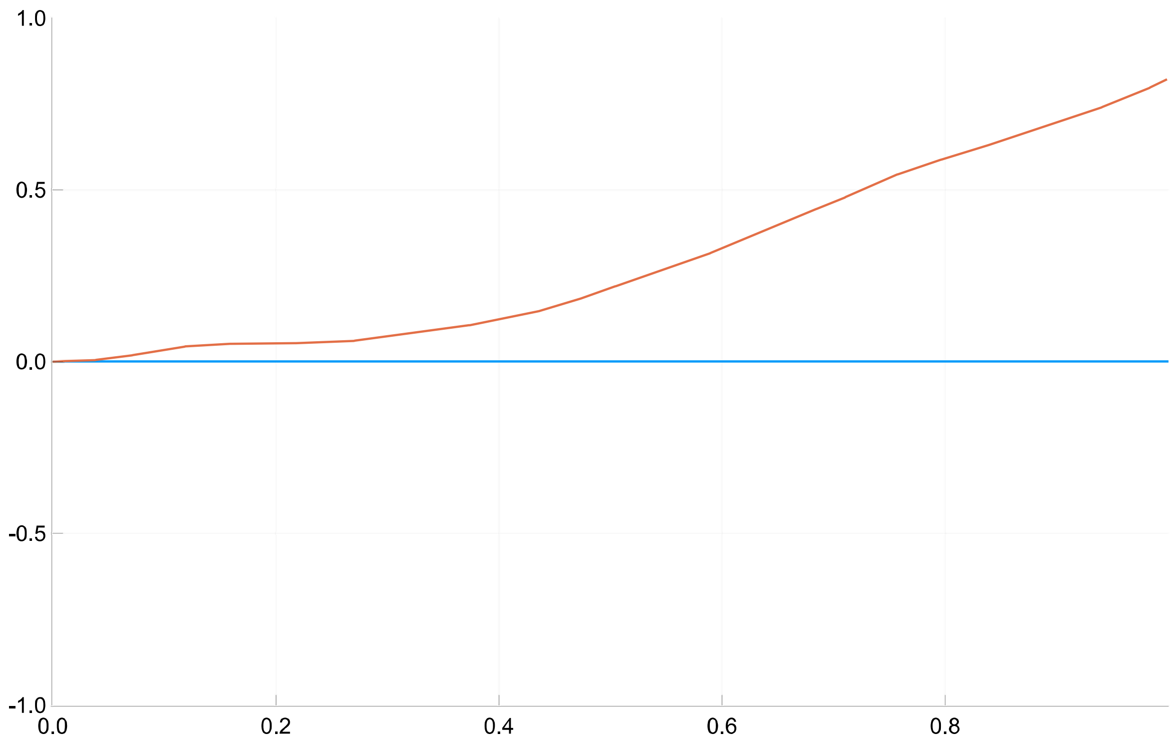
Area : 0.14751287719392545



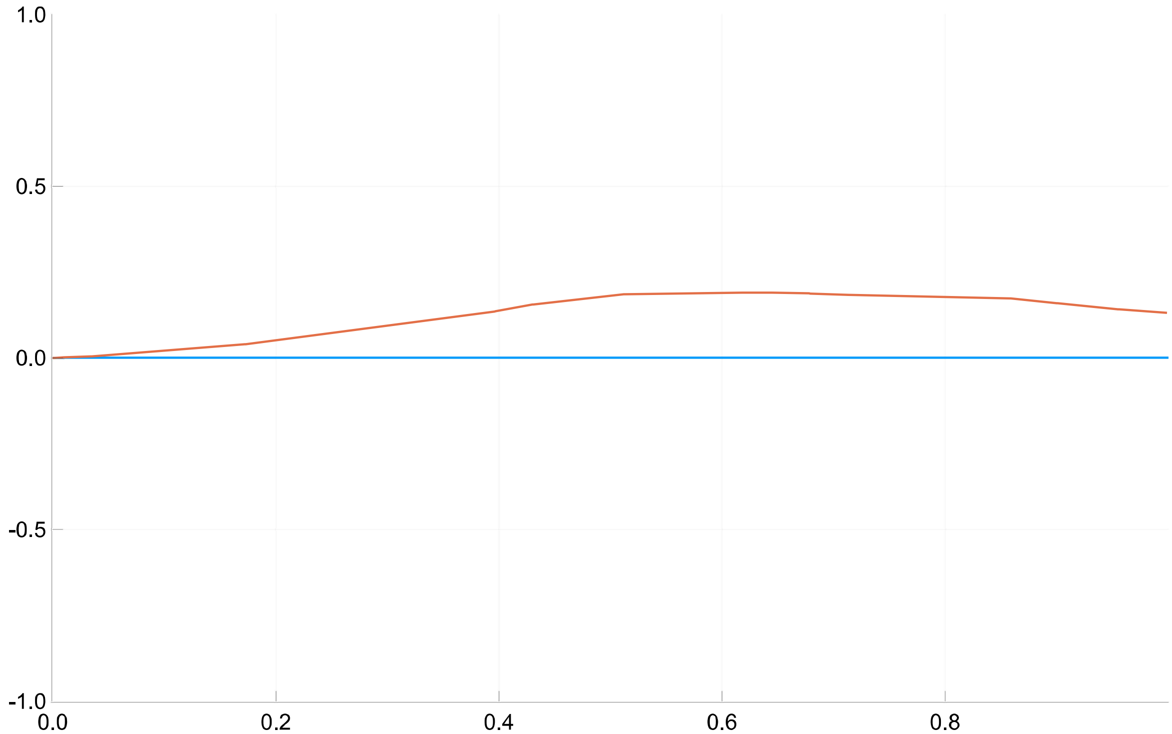
Area : -0.015789583592238993




```
Area : 0.0973206529210069
```



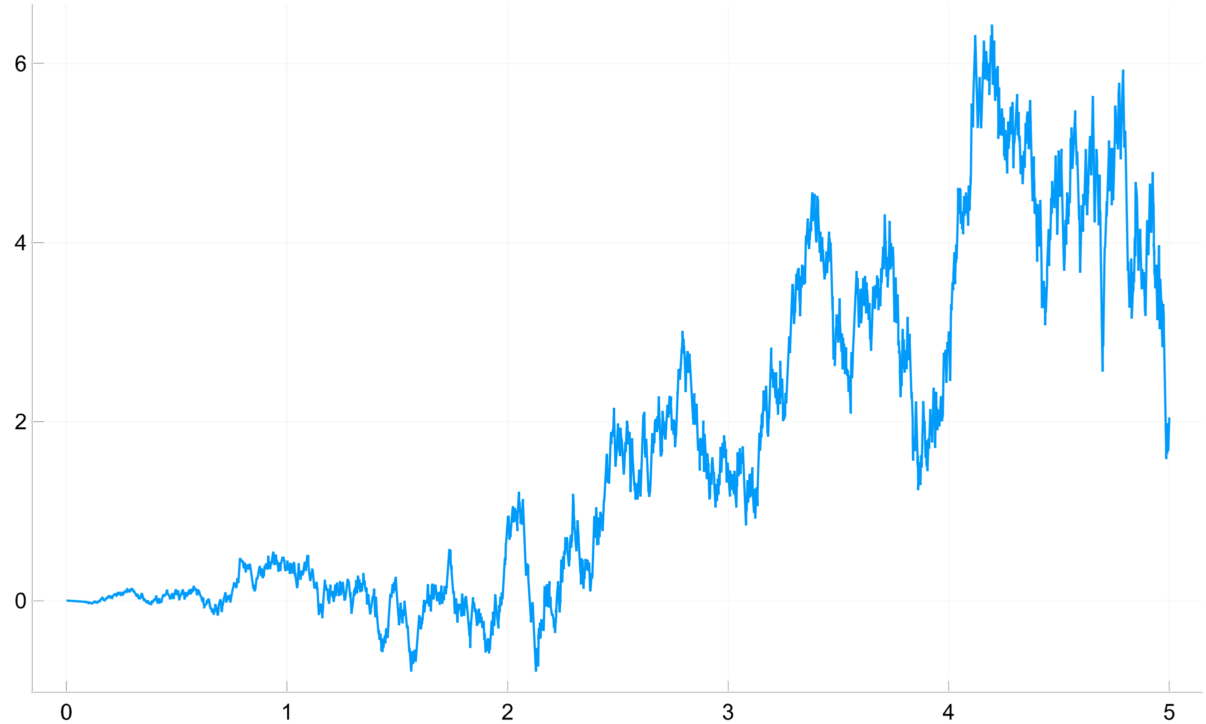

```
Area : 0.29717777526685113
```



Compute the integral $\int_0^t s\mathrm{dB_s}$

In [16]:
```
n = 10_000
t = 5
x = rand(Normal(0,sqrt(t/n)),n)
bm = [0;cumsum(x)]
steps = range(start = 0,stop = t, length = n+1)
integral = cumsum(steps[1:n].* (bm[2:n+1] .- bm[1:n]))
plot(steps[1:n],integral,label = :none)
```

Out[16]:



In [ ]: