Happy holiday! Remember to take care of yourself and your loved ones!

```
walk (generic function with 1 method)
```

```
function walk(num_steps)

    x = 0

    for i in 1:num_steps
        x += rand( (-1, +1) )
    end

    return x
end
```

```
0
```

```
walk(20)
```

```
experiment (generic function with 1 method)
```

```
experiment(num_steps, num_walks) = [walk(num_steps) for i in 1:num_walks]
```

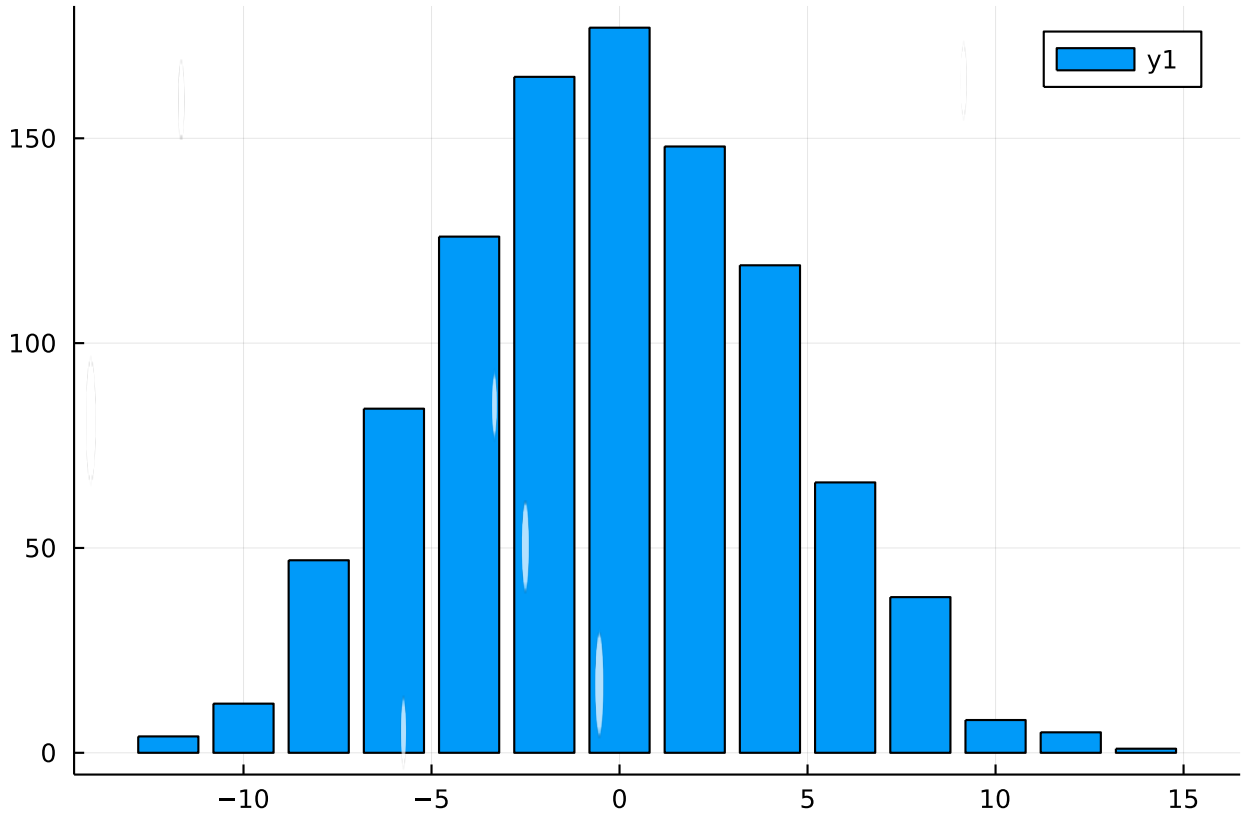
```
data =
 [4, 8, -2, -4, 2, 0, -6, 2, 0, 4, 2, 2, 2, 4, 2, 8, -2, 2, 4, -6,    more ,2, 4, 4, -2,
```

```
data = experiment(20,1000)
```

```
using StatsBase
```

```
using Plots
```

```
counts = countmap(data);
```




```
bar(counts)
```

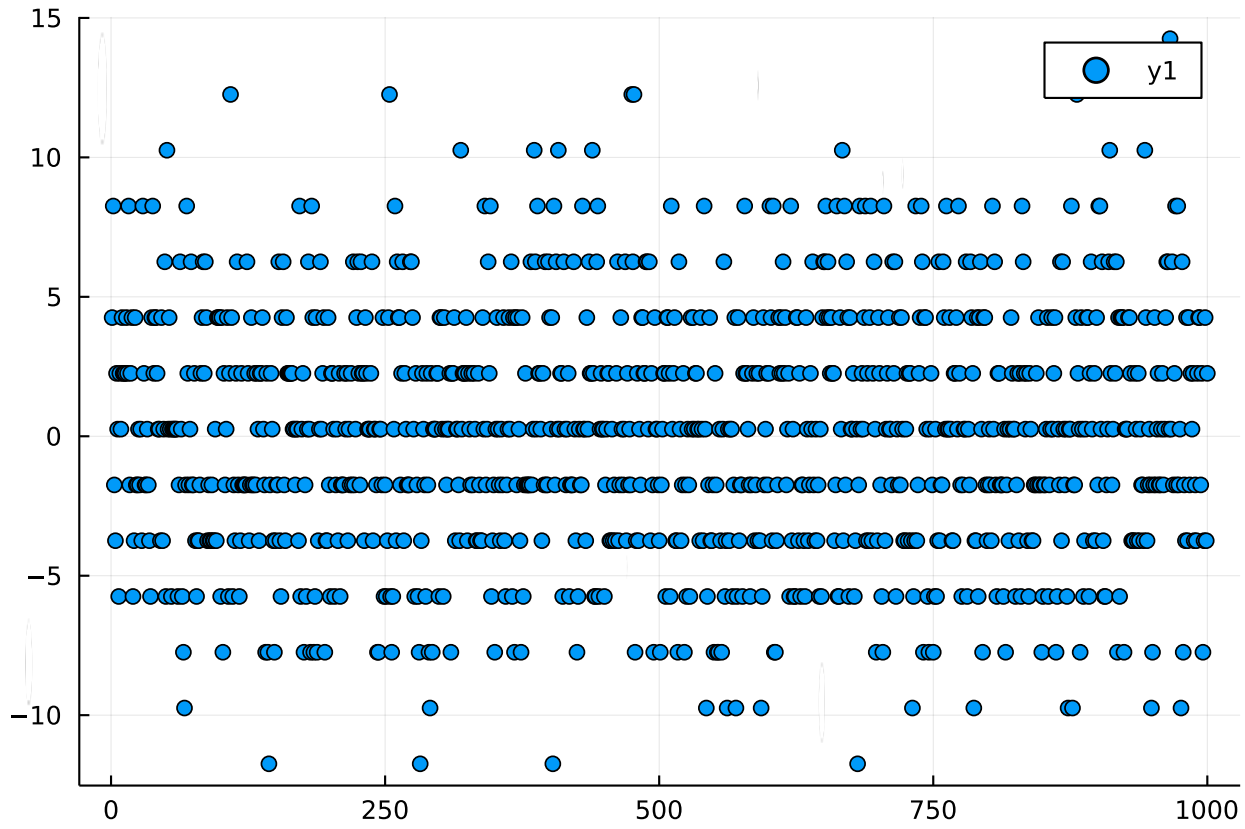
-0.256

```
mean(data)
```

```
centred_data =
  [4.256, 8.256, -1.744, -3.744, 2.256, 0.256, -5.744, 2.256, 0.256, 4.256, 2.256, 2.25
```

```
centred_data = data .-mean(data)
```

```
scatter(data);
```




```
scatter(centred_data)
```

```
1.0000000000000002
```

```
nextfloat(1.0)
```

```
2.220446049250313e-16
```

```
nextfloat(1.0)-1.0
```

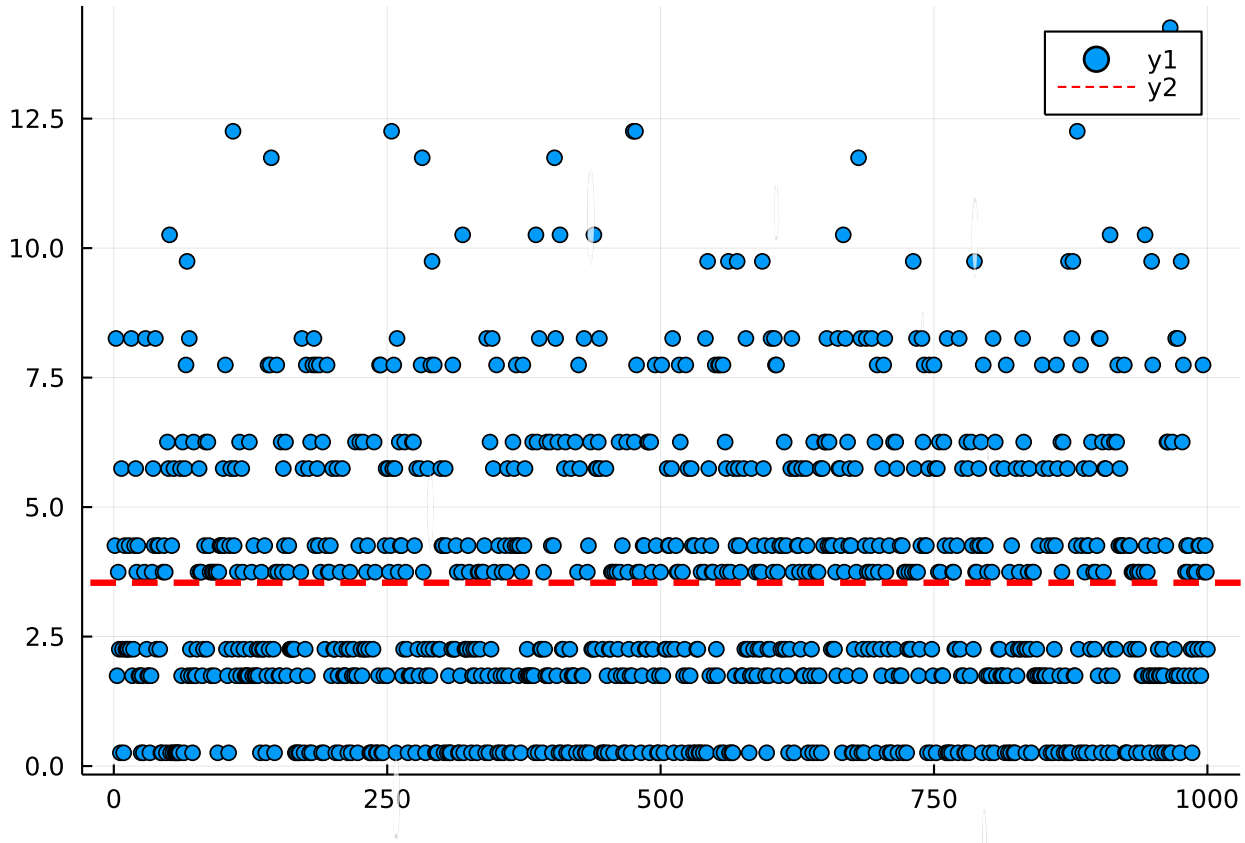
```
spread =
  [4.256, 8.256, 1.744, 3.744, 2.256, 0.256, 5.744, 2.256, 0.256, 4.256, 2.256, 2.256,
```

```
spread = abs.(centred_data)
```


```
scatter(spread)
```

3.5397440000000002

```
mean(spread)
```




```
hline!([mean(spread)], lw=3, ls=:dash, c=:red)
```

```
discrete_jump (generic function with 1 method)
```

```
discrete_jump() = rand((-1,+1))
```

```
continuous_jump (generic function with 1 method)
```

```
continuous_jump() = randn()
```

```
walk1 (generic function with 1 method)
```

```
function walk1(N)
    x=0
    for i in 1:N
        x += jump()
    end

    return x
end
```

MethodError: no method matching walk1(::Int64, ::typeof(Main.workspace#52.discrete_
Closest candidates are:
walk1(::Any) at ~/Downloads/MITCOVID6.jl#==#a43f9e4d-2d72-426c-bc4b-a3e7fcf5c8e7:1

1. **top-level scope** @ ***Local: 1*** [inlined]

```
walk1(20, discrete_jump,0.0)
```