```
begin
    using StatsPlots  ,Random  ,StatsBase  ,DataFrames
    gr()
    theme(:bright)
end

```

数据的来源-2

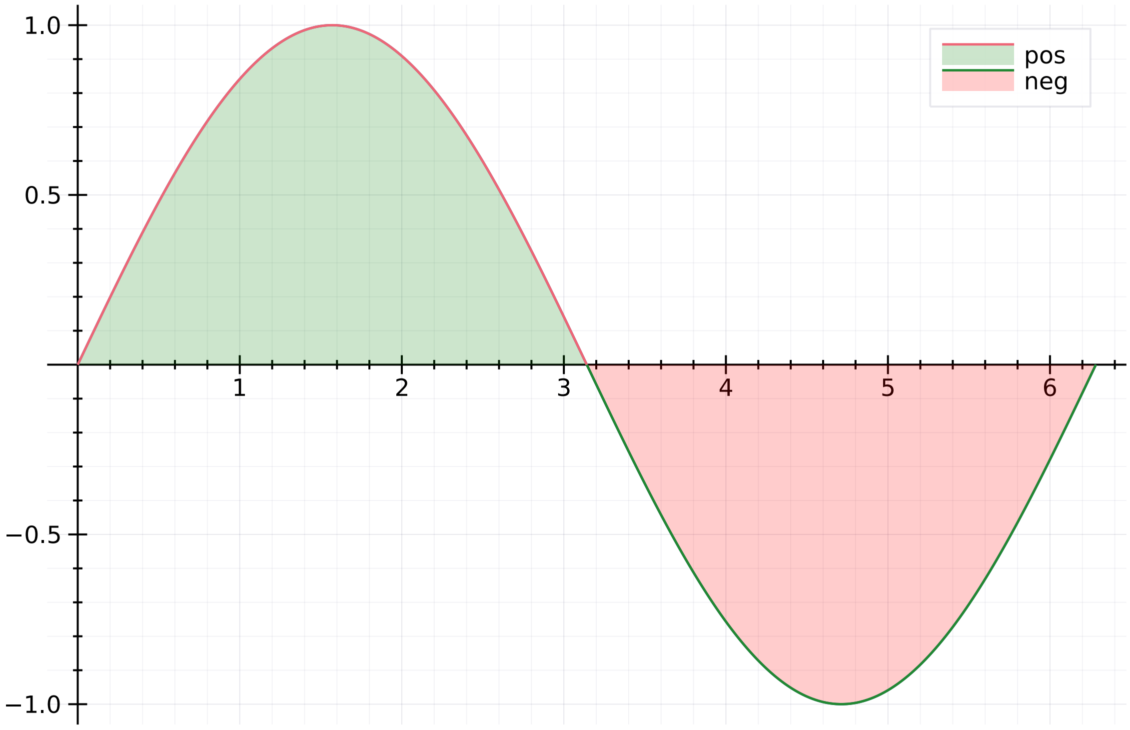
这里我们从常见的正弦函数来生成数据:

Info

统计观点:

因为正弦曲线在$[0, 2\pi]$定义域区间内的函数值分为正半周和负半周. 如果随机从定义域区间取点求函数值, 理论上说从从每次都可以获得两个绝对值相同的点(值的符号相反), 所以每次抽取的点的和理论上应该等于0, 由于随机性, 不可能正好等于0, 如果我们重复多次,得到的值应该分布在0的附近.

以正弦函数为基础采样的样本也有自己的均值和标准差(方差也可以). 对于正弦函数, 函数图形是直观的, 抽样的散点图可以看出来,但是对于一个未知的函数如果均值

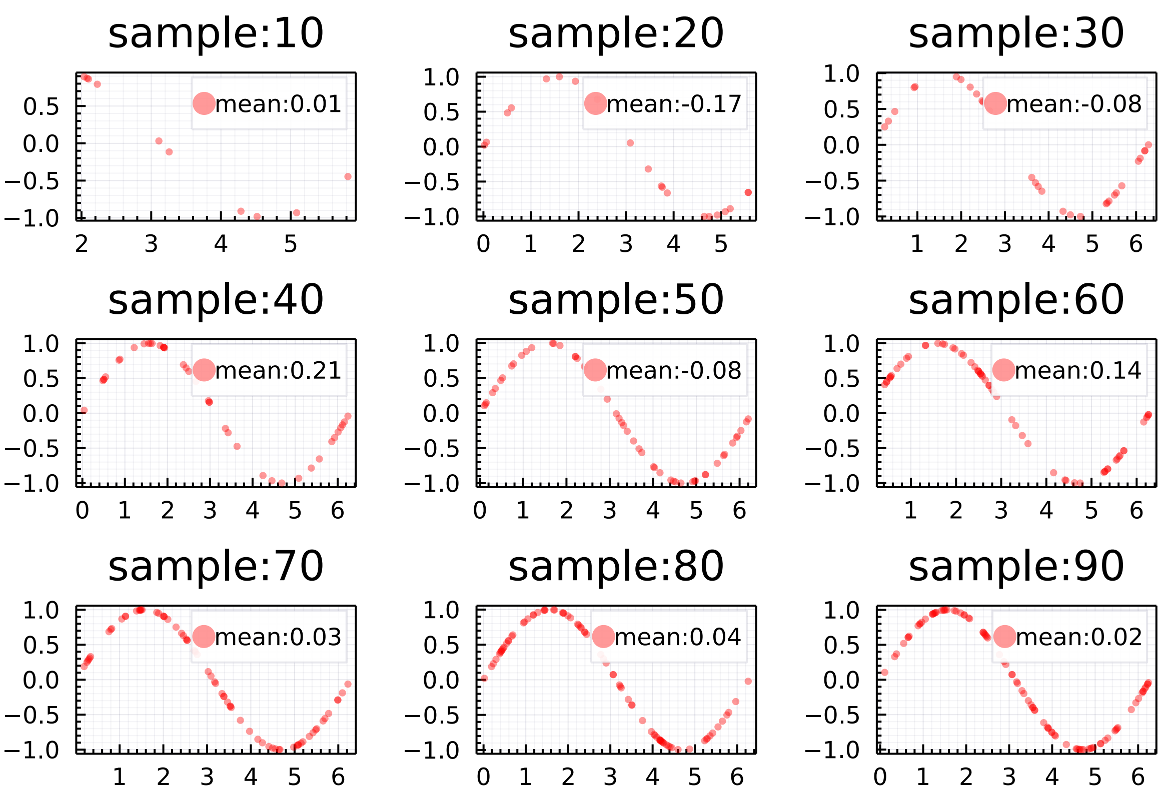



```
begin
    ######################################
     # 如果遵循随机采样, 采样 100个点 从绿色区间和
     #红色区间获取的点数量一样, 并且刚好会以绝对值相等
     # 的形式形成数据对. 函数均值为 0

    ######################################
    ran=range(0,2pi,300)
    ran1=range(0,pi,150)
    ran2=range(pi,2pi,150)
    data=sin.(ran)
    plot(ran,data,label=false,frame=:origin)
    areaplot!(ran1,sin.(ran1),fillalpha=0.2,fc=:green,label="pos")
    areaplot!(ran2,sin.(ran2),fillalpha=0.2,fc=:red,label="neg")

end
```

df =

	x	sinx
1	0.0	0.0
2	0.021014	0.0210125
3	0.042028	0.0420156
4	0.063042	0.0630002
5	0.084056	0.083957
6	0.10507	0.104877
7	0.126084	0.12575
8	0.147098	0.146568
9	0.168112	0.167321
10	0.189126	0.188001
more		
300	6.28319	-2.44929e-16

```
df=DataFrame(x=ran,sinx=data)
```

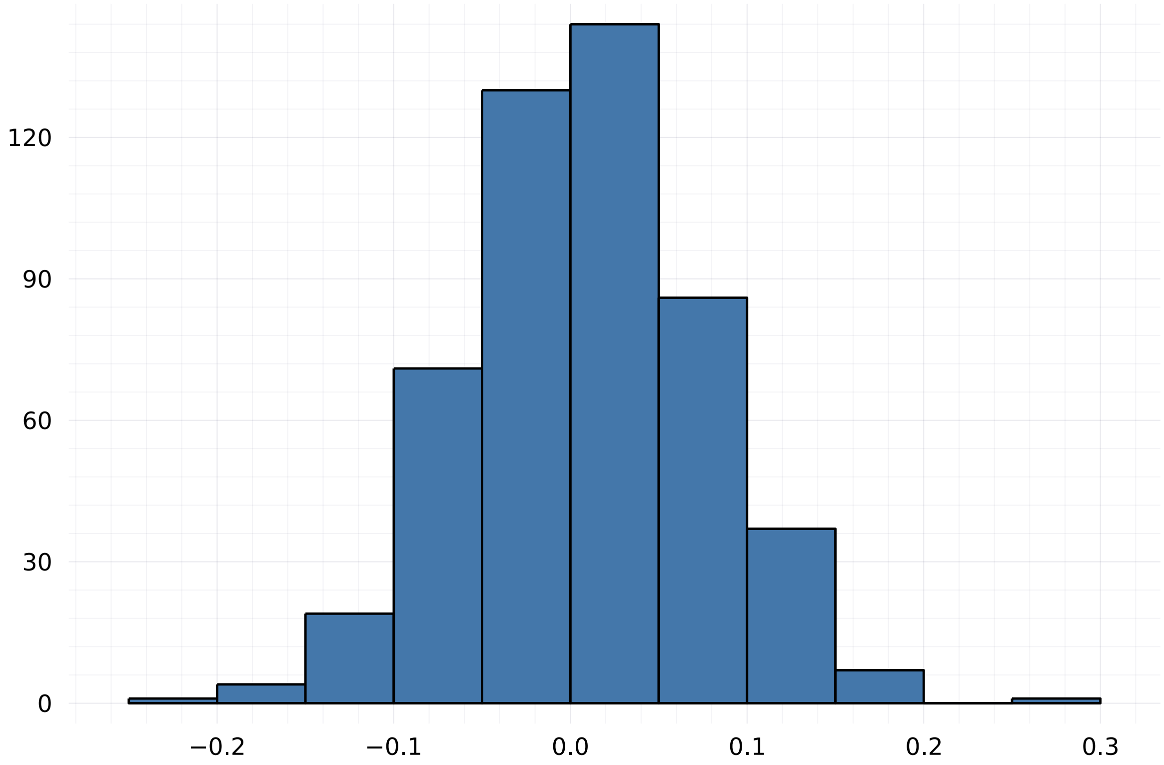



```
(10:10:90).|>(d->sin_scatter(d))|>d->plot!(d...,layout=(3,3))
```

如果定义一次实验为:从曲线中随机抽取 100 个点. 这 100 个点的值可以求出均值和标准差.

根据上面的的分析, 均值应该在 0左右. 这里的标准差和正态分布的标准差不同(因为曲线和正态分布不同),也不符合概率密度定义, 因为曲线下面积不为 1

这样的实验, 我们重复 500 次, 看看每次获得的 100 个点的均值和标准差的情况.




```
begin
    res_mean=(1:500).|>d->rand(data,100)|>mean
    res_std=(1:500).|>d->rand(data,100)|>std
    histogram(res_mean,label=false)
end
```

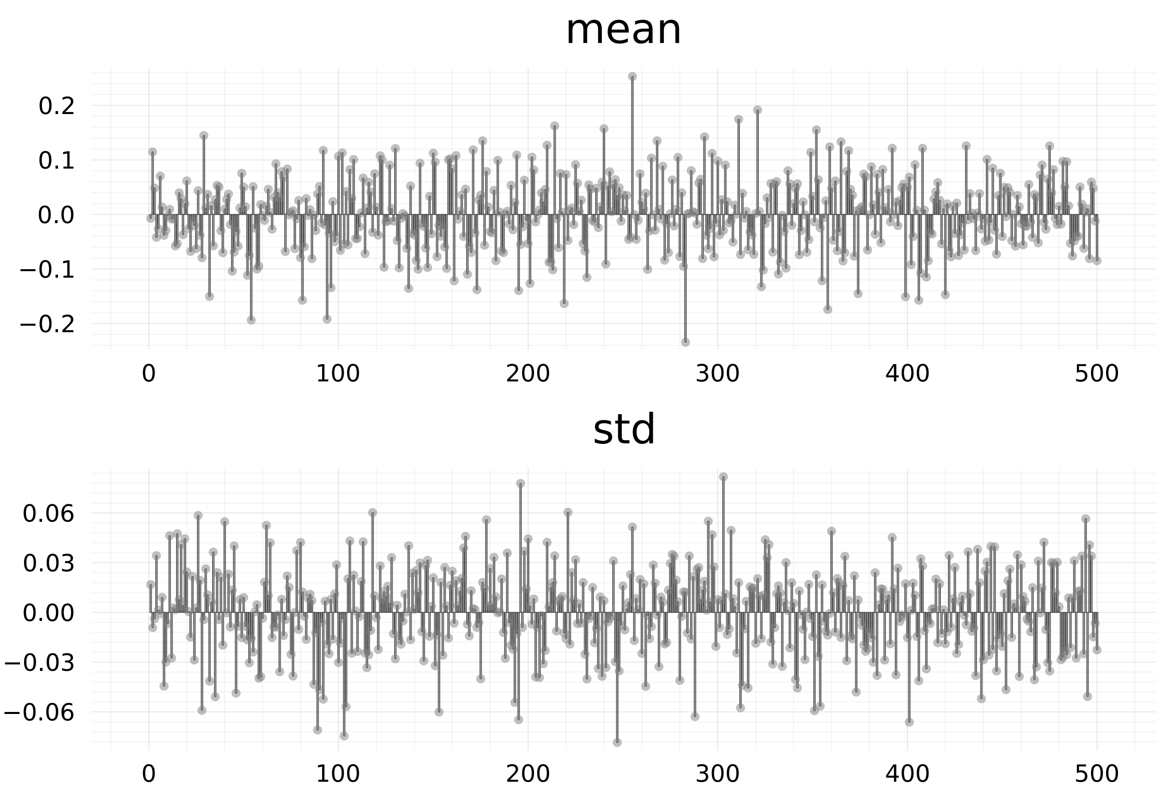
可以看到:1000 次重复实验的均值大部分都集中在 0附近.

为什么从正弦曲线抽出的点均值不是正弦曲线?

因为我们根本没有在研究曲线是什么, 我们现在观察的是函数的均值特性. 因为正弦函数的均值是一个特性. 从中抽出的点的均值也应该大致反映这个属性. 这就是大数定律和中心极限定律所说明的问题.

如果要了解数据点代表的函数是什么, 我们要用到回归方法, 对于正弦曲线, 我们要用多项式拟合的方法.

从下面的均值和标准差的残差图也可以看出, 都在很小的幅度内变化.




```
begin
    p1=res_mean.|>(x->x-mean(res_mean))|>stem
    p2=res_std.|>(x->x-mean(res_std))|>stem
    plot!(p1,p2,title=["mean" "std"],layout=(2,1))
end
```

stem (generic function with 2 methods)

```
function stem(res::Array,color=:grey)
        bar(res,fillcolor=color,width=0.5,label=false,alpha=0.5)
        plot!(1:length(res), res, color = color, markershape = :circle,ms=2.5,
            alpha = 0.5, label = "", linewidth = 0)
end
```

sin_scatter (generic function with 1 method)

```
function sin_scatter(num)
     xs=rand(df[!,1],num)|>sort
     ys=sin.(xs)
     y_means=mean(ys)|>d->round(d,digits=2)
     scatter(xs,ys, title="sample:$(num)", label="mean:$(y_means)",
ms=2,ma=0.4,mc=:red,frame=:box)
end
```