```
begin
    using StatsPlots    ,StatsBase    ,LaTeXStrings    ,Distributions    ,Latexify

end
```

总体随机抽样实验介绍

Info

在统计学中有关总体(population)和样本(sample)的方法以及思想完全不同. 所以在这里我们都使用全称, 标注**总体-xx** 和**样本-xx**的方法要严格区分开,不要混用.尽管样本从总体获取的子集, 但是两者说明的信息不同.

总体的信息固定不变. 样本的信息随着获取的个体不同而发生变化.

当然总体的性质会随着时间发生改变, 在统计学中大多数情况下会取一个时间截面来研究.

在这里我们实验的流程如下:

1. 构建总体(population), 一旦构建完成就不能发生改变
2. 为总体中的每个个体赋予一个索引值(index,名字)
3. 随机点n个名字, 然后获取对应的值. 这个过程称为抽样(sampling), 每次抽样(sampling)获取一个样本(sample),每个样本(sample)含有n 个值

这是所有统计研究的基础步骤. 下面我们按照这个流程来说明

1.构建总体

总体是客观存在,统计学没有对总体的形式做任何限制. 比如我们可以99乘法表作为一个总体

这个总体的好处是没有时间属性, 在任何时间点,总体的信息都不会发生改变.

这么做是非常有意义的, 后面会讲到.

直接使用 julia 的方法构建一个向量就可以表示乘法表中的数字

```
p =
  [1, 2, 3, 4, 5, 6, 7, 8, 9, 2, 4, 6, 8, 10, 12, 14, 16, 18, 3, 6,    more ,72, 9, 18, 2
```

```
p=population=[x*y for x in 1:9 for  y in 1:9]|>Vector
```

下面是对99乘法表的描述统计.

描述统计列出一个总体某些属性的度量值.

```
Summary Stats:
Length:         81
Missing Count:  0
Mean:           25.000000
Minimum:        1.000000
1st Quartile:   9.000000
Median:         20.000000
3rd Quartile:   36.000000
Maximum:        81.000000
```

```
population|>summarystats
```

```
Dict("extreme" => 80, "mean" => 25.0, "std" => 19.5576)
```

```
Dict("mean"=>mean(p),"extreme"=>p|>d->maximum(d)-minimum(d),"std"=>p|>d->std(d))
```

在统计学中经常会谈到一个术语叫自由度(degree of freedom,df).

99乘法表的自由度(df)是多少?

99乘法表的自由度(df)为0, 因为一旦这个总体构建完成, 任何的度量值都不会再发生变化.

所以自由度度量了一个系统中变量的个数.

常见的二元一次方程:

$$x + y = 5$$

的自由度为1, 因为给定一个x值, y的值就确定了, 因为是等式,$x + y$要满足等式条件.y的变化是受x变化控制的.

自由度是对变量的泛化. 在二元一次方程中,也可以给定一个y值, x相应的变化.

这两种形式就可以抽象为自由度来表示,自由度度量了系统变量的数目,但是并没有划定哪个未知数是变量

平均值为什么是 25?

这是由均值的计算方法决定的, 总体均值度量了总体的一个属性

总体均值公式为:

$$mean = \frac{\sum X}{N}$$

总体均值大体反映了总体所有个体属性度量值的中心位置. 所以这是一个抽象概念, 不是一个实际值, 均值不代表任何实际值.

下面我们还是返回到99乘法表,开始抽样实验流程

2.随机抽样

2.1 每次抽取 5 个数字

随机抽样将总体中的个体排成队列, 队列中顺序并不重要, 可以打乱,也可以按照度量值大小降序或者升序排列.

队列中个体的顺序对抽样没有任何影响.因为我们随机抽取的是队列的序号.

以99乘法表的数字为例, 我们只关注序号, 所以可以以二维表的形式排列, 这就是99乘法表,可以表述如下$(1,1,1)$,$(2,4,8)$, 前两个数字表示序号, 最后一个数字表示值.

从99乘法表中抽取一个数字是均一分布, 因为每个数字被抽到的机会均等,均为 $\frac{1}{81}$

以二维表形式呈现:, 接着就可以根据行列坐标进行抽样

9	18	27	36	45	54	63	72	81
8	16	24	32	40	48	56	64	72
7	14	21	28	35	42	49	56	63
6	12	18	24	30	36	42	48	54
5	10	15	20	25	30	35	40	45
4	8	12	16	20	24	28	32	36
3	6	9	12	15	18	21	24	27
2	4	6	8	10	12	14	16	18
1	2	3	4	5	6	7	8	9

```
nineninetable()
```

9	18	27	36	45	54	63	72	81
8	16	24	32	40	48	56	64	72
7	14	21	28	35	42	49	56	63
6	12	18	24	30	36	42	48	54
5	10	15	20	25	30	35	40	45
4	8	12	16	20	24	28	32	36
3	6	9	12	15	18	21	24	27
2	4	6	8	10	12	14	16	18
1	2	3	4	5	6	7	8	9

```
plot_sample2(;n=5)# 点击右边运行符号
```

在上面的图中, 我们以二维表格的形式展示乘法表, 每次点击会随机从表中抽取 5 个值.

因为这**5**个值来源于乘法表, 那么必然具有乘法表总体的某些相似的性质.

这个五个数的平均值是否也具有乘法表总体的性质? 性质是否稳定?

为了验证这个问题, 我们重复进行多次抽样实验(sampling)

我们要完成 500次抽样实验

```
observations =
```

```
[[72, 6, 6, 81, 16], [30, 54, 24, 12, 54], [12, 5, 42, 18, 32], [56, 35, 12, 12, 6], [
```

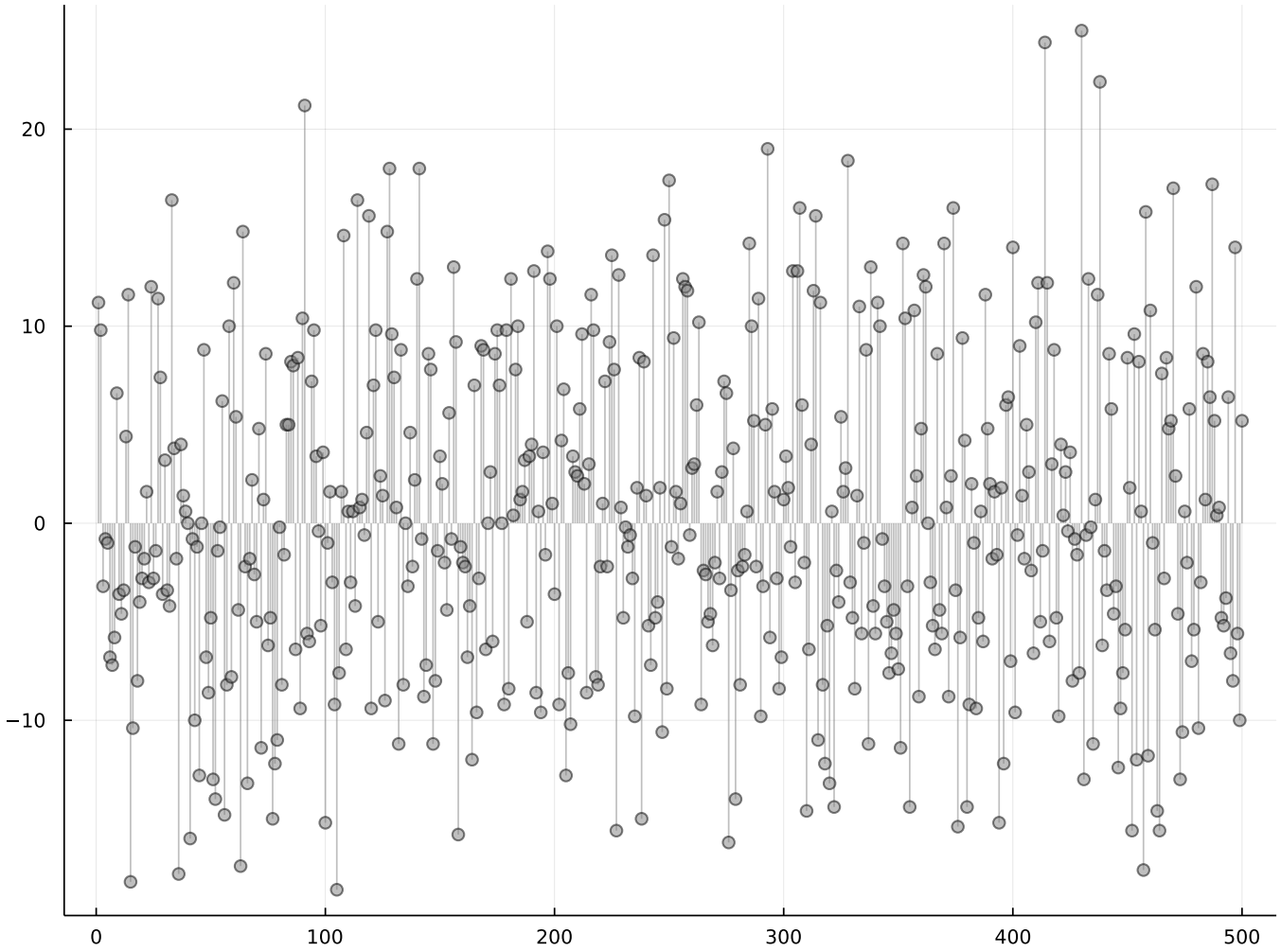
```
observations= [sample(p,5) for i in 1:500] #随机采样使用的放回式采样， 取出一个数字,读取值， 然后再放回取， 进行下一次抽取， 所以会出现重复值
```

```
mean_observations =
```

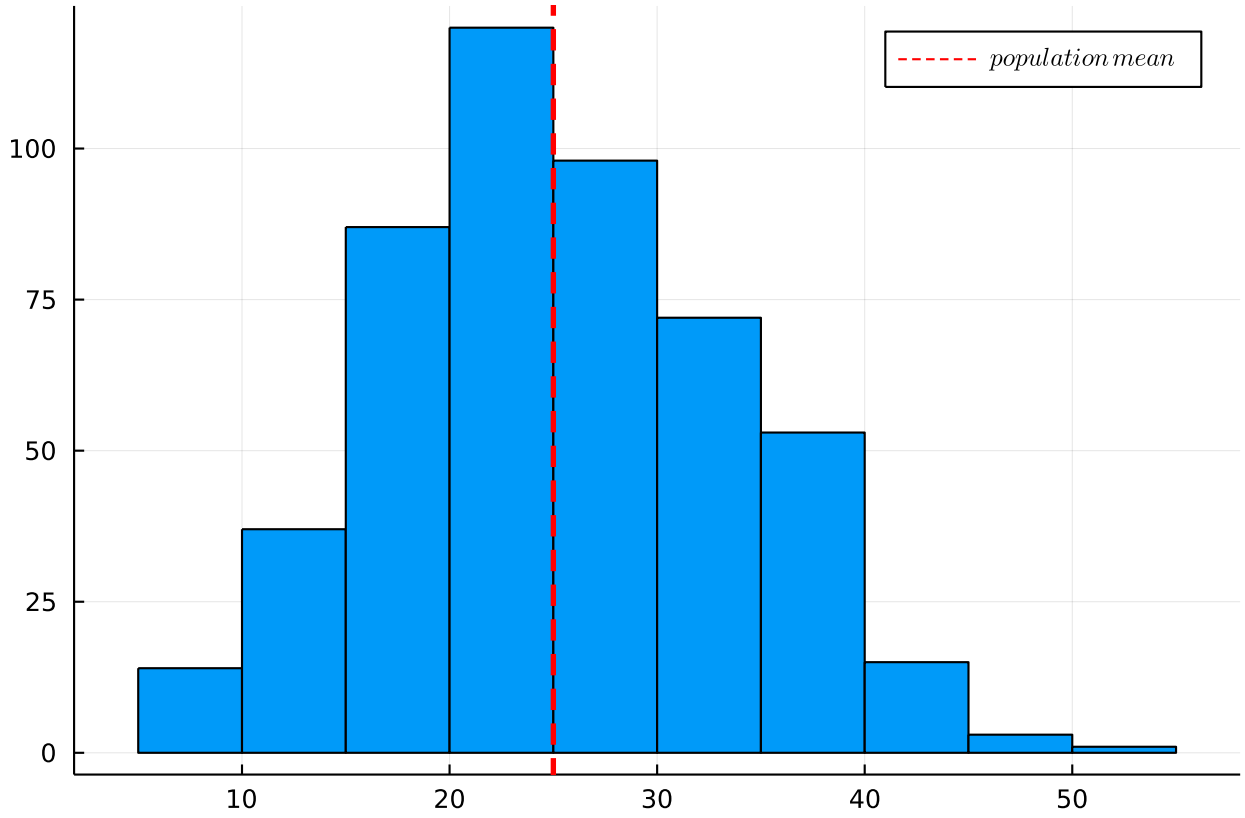
```
[36.2, 34.8, 21.8, 24.2, 24.0, 18.2, 17.8, 19.2, 31.6, 21.4, 20.4, 21.6, 29.4, 36.6,
```

```
mean_observations=observations.|>d->mean(d)
```

似乎每次取出的**5** 个数字的平均值差异很大, 下面绘制出均值的残差图, 就是 500 次抽样的均值分别减去总体的均值, 然后绘制出茎叶图




```
(mean_observations.-mean(p))|>stem
```




```
begin
    histogram(mean_observations,label=false)
    vline!([mean(p)],ls=:dash, lc=:red, label=L"population \, mean",lw=2.5)

end
```

2.2 随机采样2-加大随机采样的个数到 20 个

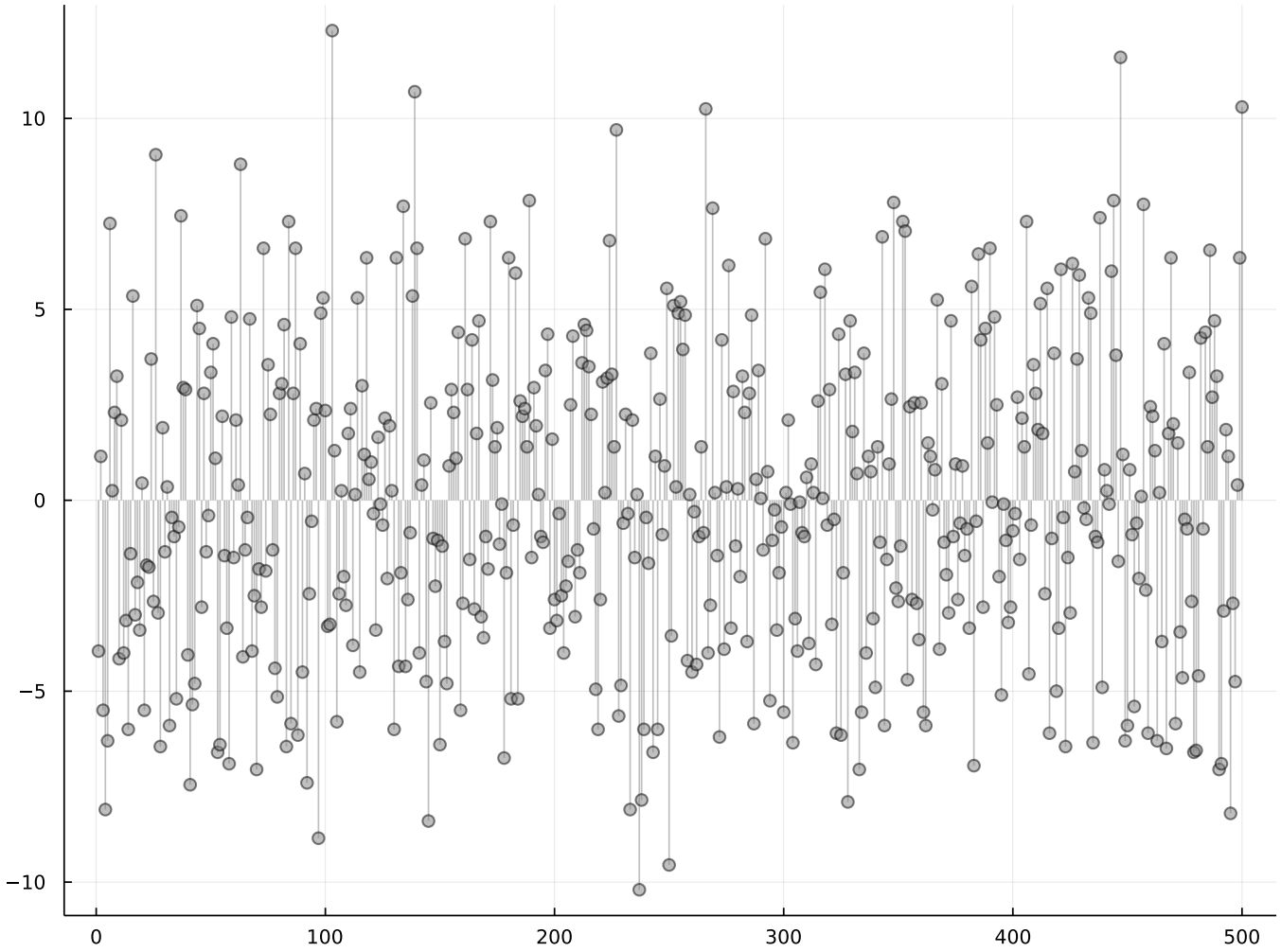
因为总体有81个, 所以每次抽样取出 5 个数字,可能提供的信息过少, 我们把每次抽取的数字增加到 20 个,看看是什么情况

9	18	27	36	45	54	63	72	81
8	16	24	32	40	48	56	64	72
7	14	21	28	35	42	49	56	63
6	12	18	24	30	36	42	48	54
5	10	15	20	25	30	35	40	45
4	8	12	16	20	24	28	32	36
3	6	9	12	15	18	21	24	27
2	4	6	8	10	12	14	16	18
1	2	3	4	5	6	7	8	9

```
plot_sample2(;n=20)# 点击右边运行符号
```

实际从上图中可以直观的看到, 当加大抽取的数字时, 每次抽取的数字覆盖的面积总和比5个数字时的面积要大. 抽取的个体越多, 提供的信息也就越多.

```
md"""
实际从上图中可以直观的看到， 当加大抽取的数字时， 每次抽取的数字覆盖的面积总和比$5$个数字时的面积要大． 抽取的个体越多， 提供的信息也就越多．
"""
```



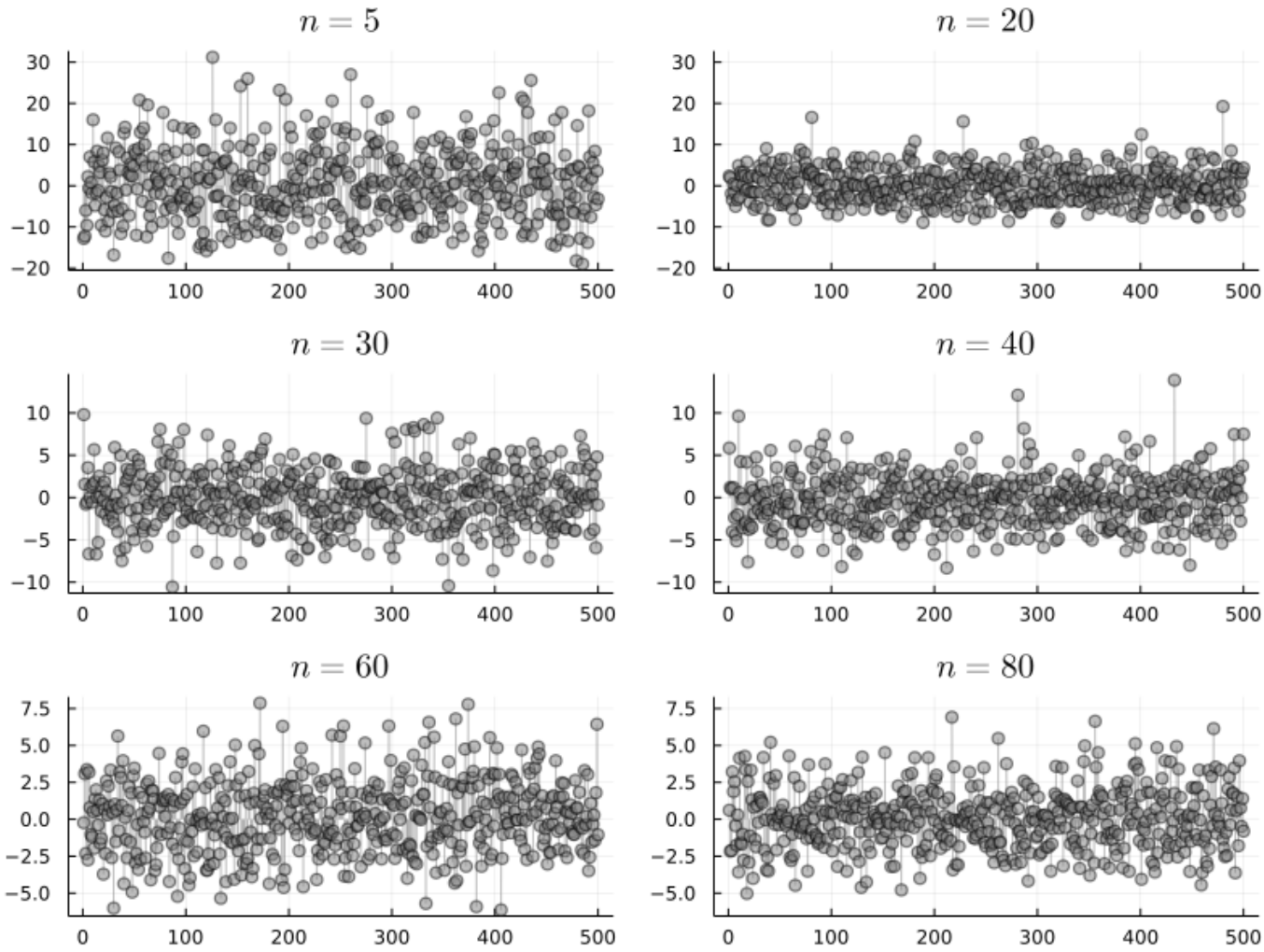

```
[sample(p,20) for i in 1:500].|>(d->mean(d)).|>(d->d-mean(p))|>stem
```

注意y 轴的值, 已经幅度已经比5 个数字时小很多

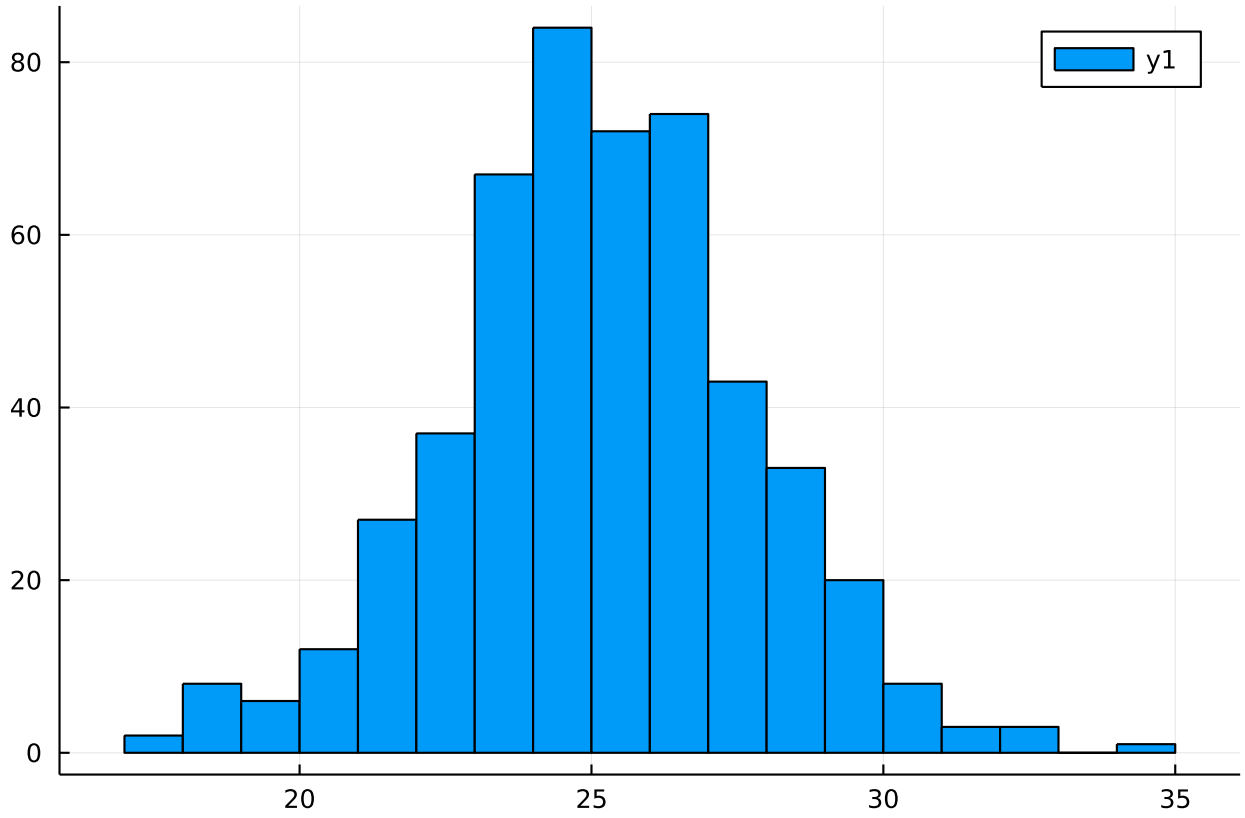
2.3 每次取出不同数量数字时的残差

当我们从总体中拿出的个体数量越多, 这些个体的性质越接近于总体的性质. 至少在均值这个属性上可以验证这一点.

但是要时刻注意, 因为每次取出的个体提供的信息总是不能完全和总体一致, 所有始终会有偏差存在. 这点是信息的变异, 变异也是总体的一个性质.




```
begin
    res_plot(p;n=20)=[sample(p,n) for i in 1:500].|>(d->mean(d)).|>(d->d-
  mean(p))|>stem
    plot_arr=[res_plot(p;n=i) for i in [5,20,30,40,60,80]]
    # p1=[sample(p,5) for i in 1:500].|>(d->mean(d)).|>(d->d-mean(p))|>stem
   title=[L"n=%$(n)" for i  in 1:1, n in [5,20,30,40,60,80]]
   plot(plot_arr...,link=:y,layout=(3,2),title=title)
end
```




```
[sample(p,60) for i in 1:500].|>(d->mean(d))|>histogram
```

结论

随机抽样实验试图通过总体中少量个体的信息来代表整个总体的信息. 当我们从一个总体中抽取的个体数达到一定的数量(越多越好, 一般取到$n = 30$),样本就可以近似的表现出总体的一些性质. 我们并不是要完全捕获总体的所有信息, 有的总体容量是无限大的, 不可能捕获所有的信息.

以均值为例, 从当总体中抽取的个体比较多的时候, 样本个体的均值应该近似和总体的均值相差不大, 特别是在抽取的个体数量较多的时候.

抽样的时候抽取的个体集合的信息不能完全和总体相符, 这里的信息偏差是抽样统计需要考虑的问题. 所以在统计中使用样本信息需要冒一定的风险.

随机抽样保证了从全局范围内捕获总体的信息, 而不是在某个局部着力过多.

这就是盲人摸象的改进版本, 一个既有扇子特性, 又有墙的属性, 又有蛇的属性的物体不可能只是这几种物体之一, 而是一个综合体. 在面对不能直观理解的事物时, 需要从全局的角度去考虑信息.

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

```
function plot_sample(;n=5)
    sample_num=n
    sample_span=0:1:9
    span=1:10
    offset=0.5
    xs,ys=rand_sample(;dist=d,n=sample_num).+offset,rand_sample(;dist=d,n=sample
    _num).+offset
    println(xs.+offset)
    println(ys.+offset)
    plot(size=(300,300),lims=(0,10),ticks=0:10,frame=:box,bg = :linen)
    hline!(span,label=false,lc=:black)
    p1=vline!(span,label=false,lc=:black)
    scatter!(xs,ys,label=false,marker=:square,ms=12,mc=:purple,ma=0.6)
end
```

```
plot_sample2 (generic function with 1 method)
```

```
function plot_sample2(;n=5)
    sample_num=n
    sample_span=1:9
    span=1:10
    offset=0.5
    function plot_num(x,y)
        num=x*y
        (x+offset,y+offset,text(L"%$(num)",pointsize=12,
        halign=:center,valign=:center))
    end
    ann=[plot_num(x,y) for x in sample_span for y in sample_span ]
    xs,ys=rand(sample_span,sample_num).+offset,rand(sample_span,sample_num).+off
    set
    #println(xs.+offset)
    #println(ys.+offset)
    plot(size=(300,300),lims=(1,10),frame=:box,bg =
    :linen,ann=ann,axis=false,ticks=false)
    hline!(span,label=false,lc=:black)
    p1=vline!(span,label=false,lc=:black)
    scatter!(xs,ys,label=false,marker=:square,ms=12,mc=:red,ma=0.3)
end
```

nineninetable (generic function with 1 method)

```
function nineninetable()
    span=1:9
    offset=0.5
    function plot_num(x,y)
        num=x*y
        (x+offset,y+offset,text(L"%$(num)",pointsize=12,
        halign=:center,valign=:center))
    end
    ann=[plot_num(x,y) for x in span for y in span ]

    plot(size=(300,300),lims=(1,10),ticks=false,frame=:box,bg = :linen,ann=ann)
    hline!(span,label=false,lc=:black)
    vline!(span,label=false,lc=:black)
end
```

stem (generic function with 2 methods)

```
function stem(res::Array,color=:grey)
    #residual=observations.-mean(p)
    n=length(res)
    range=1:n
    plot(repeat((1:n)', 2),
         [zeros(1, n); res'], label = "", color = :grey, alpha = 0.5,size=
(800,600),frame=[:zerolines, :box])
    plot!(1:n, res, color = :grey, markershape = :circle,
          alpha = 0.5, label = "", linewidth = 0)
end
```

rand_sample (generic function with 1 method)

```
rand_sample(;range=0:8,n=5)=sample(range,n,replace=false)
```

factor (generic function with 1 method)

```
function factor(num)


    for  i in 1:9
         for j in 1:9
          if i*j==num
            #println(i*j)
             return rand()-0.5 >0 ? [i,j] : [j,i]

          end
         end
    end
end
```