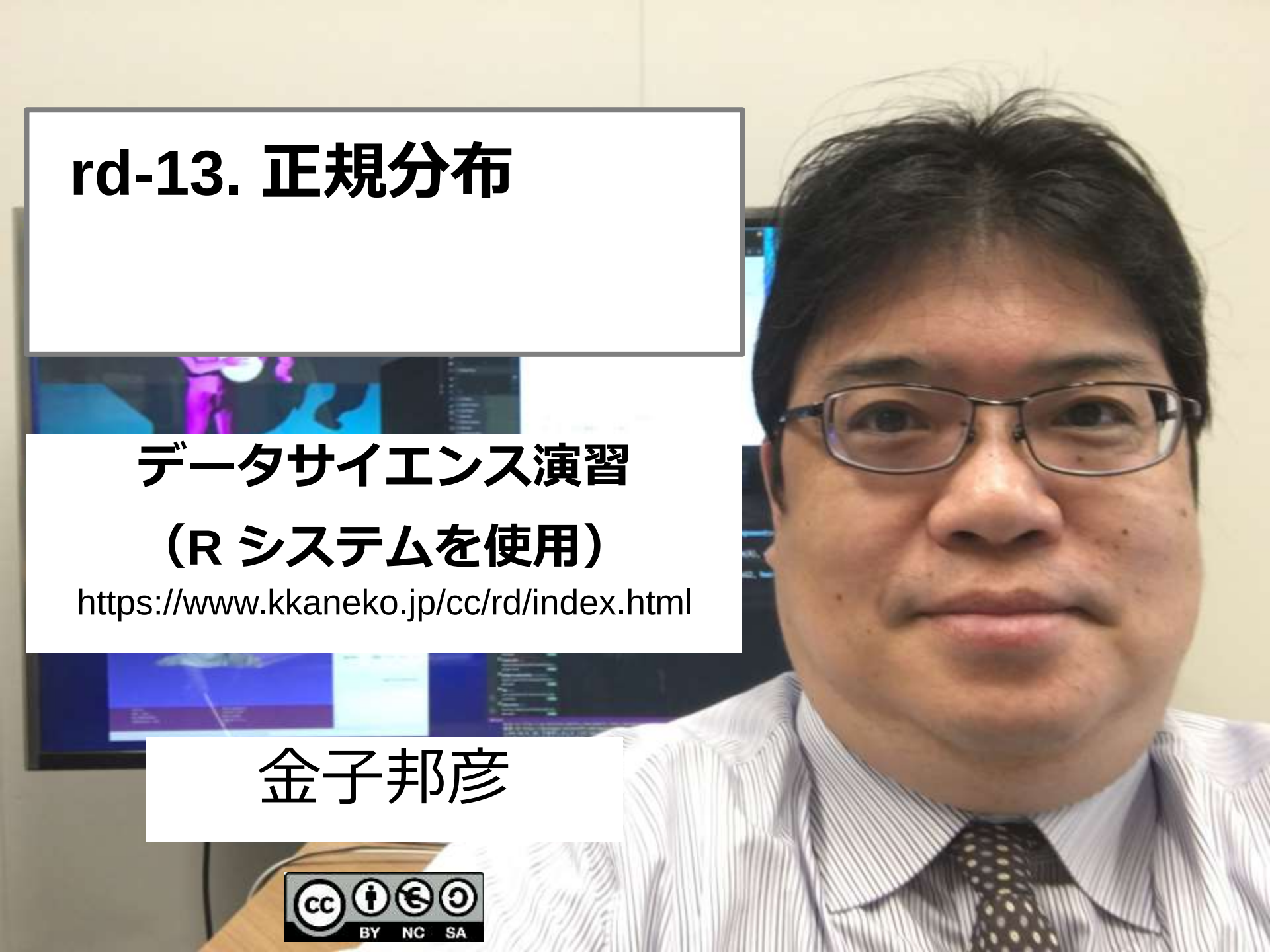
rd-13. 正規分布



データサイエンス演習

（R システムを使用）

https://www.kkaneko.jp/cc/rd/index.html

金子邦彦

コイン投げ

- コインを投げて，裏か表を出す．　コインに仕掛けなどはない

コイン投げでの「表の枚数」は変数

- コインが２００枚あるとする

２００枚を一斉に投げて，表の枚数を数える

→　何度も繰り返す

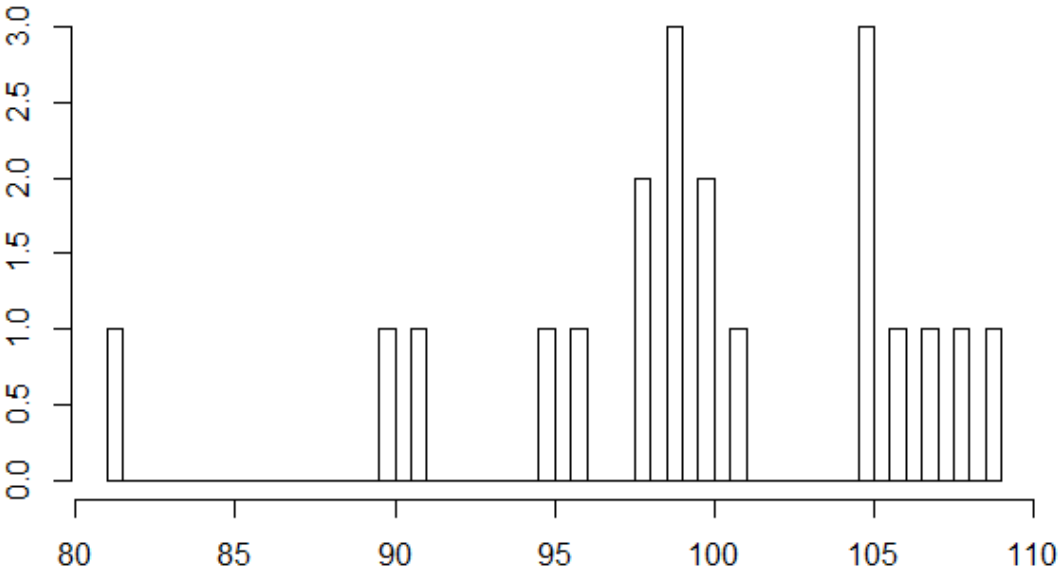
（例）

97, 100, 111, 96, 87, 93, 99, 99, 104, 92, 112, 98, 94, 101, 108, 98, 100, 117, 103, 100, ...

分布の例

- コインが２００枚あるとする
- ２００枚を一斉に投げて，表の枚数を数える

それが
起きた回数
（頻度）

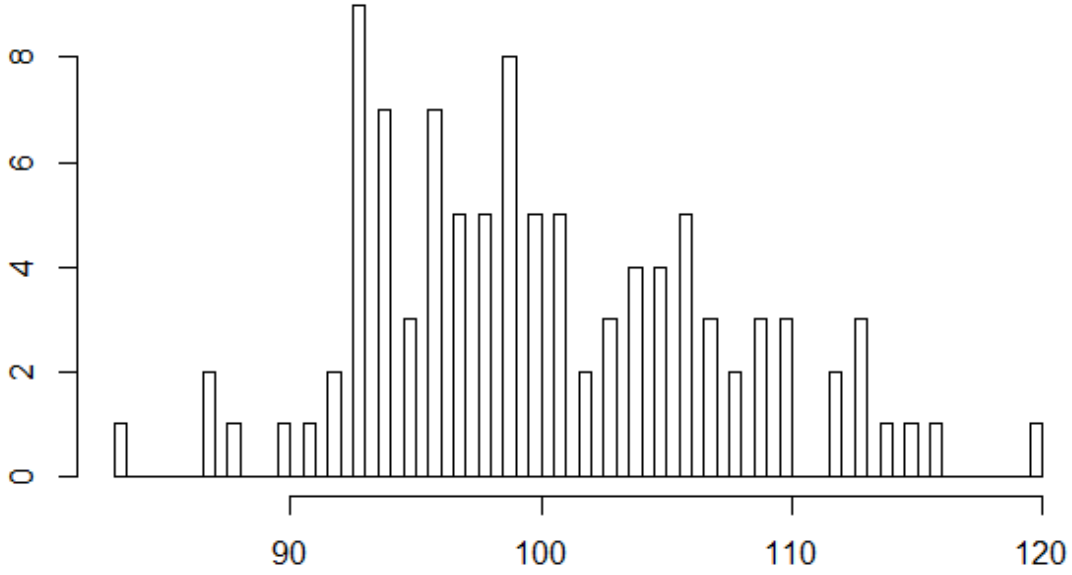


２０回投げたときの例

分布の例

- コインが２００枚あるとする
- ２００枚を一斉に投げて，表の枚数を数える

それが
起きた回数
（頻度）



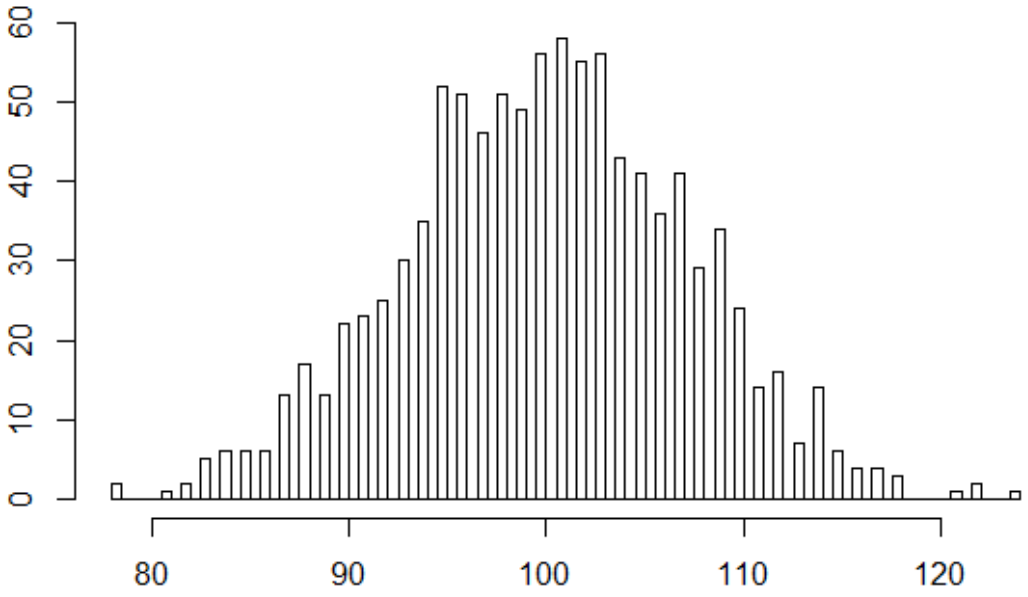
表の枚数

１００回投げたときの例

分布の例

- コインが２００枚あるとする
- ２００枚を一斉に投げて，表の枚数を数える

それが
起きた回数
（頻度）



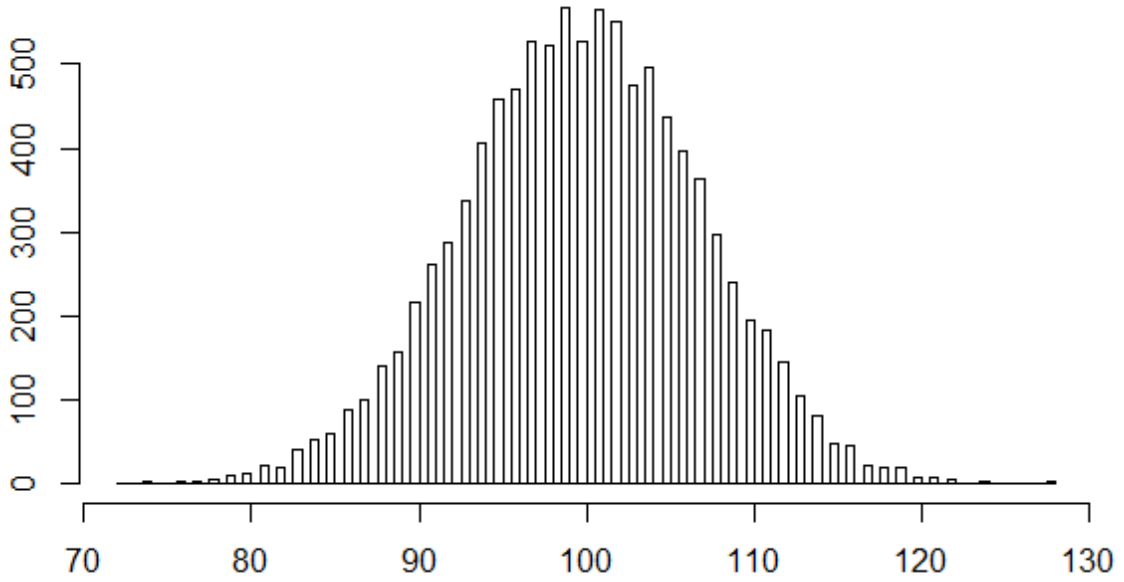
表の枚数

１０００回投げたときの例

分布の例

- コインが２００枚あるとする
- ２００枚を一斉に投げて，表の枚数を数える

それが
起きた回数
（頻度）

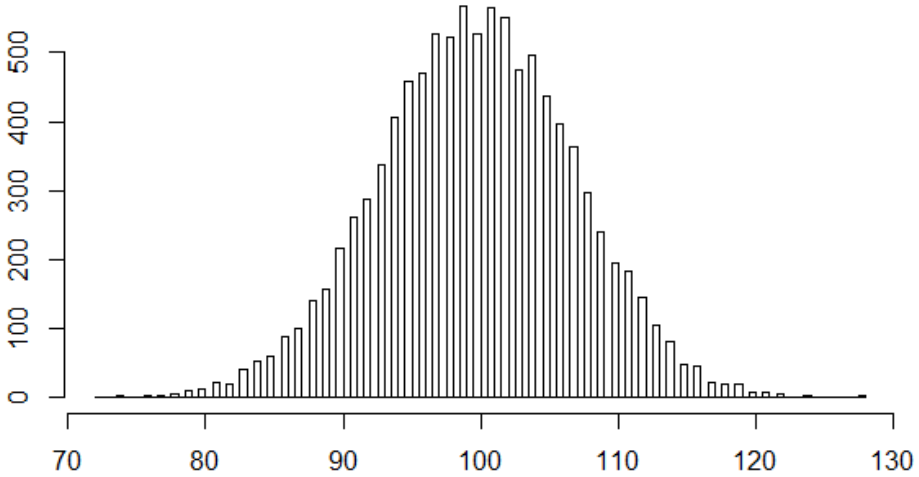


１００００回投げたときの例

- コイン投げゲーム
  - コインを２００枚を一斉に投げる（１回勝負）
  - 表の枚数が１１０枚以上なら　勝ち
  - 表の枚数が１０９枚以下なら　負け

- この勝負に勝てそうか？

- コイン投げゲーム
  - コインを２００枚を一斉に投げる（１回勝負）
  - 表の枚数が１１０枚以上なら　勝ち
  - 表の枚数が１０９枚以下なら　負け



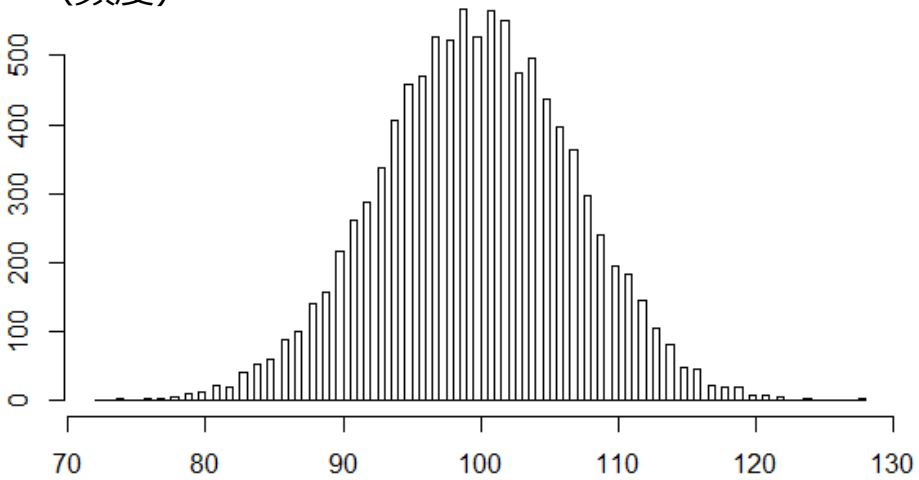
１００００回投げてみたら,
表の枚数が**１１０**枚以上
８９４回
表の枚数が**１０９**枚以下
９１０６回

８．９パーセントくらいの確率で勝てそう！

- 勝率５%のゲーム（１００回に５回勝てそうなゲーム）を作りたいとする

コイン２００枚を投げて，１１２枚以上表立ったら勝ちゲーム

それが
起きた回数
（頻度）



１００００回投げてみたら，
表の枚数が**１１２**枚以上
５１８回
表の枚数が**１１１**枚以下
９４８２回

５．２パーセントくらいの確率で勝てそう

4-5 母平均と母分散の活用例

今から行うことのイメージ

値が変化する何か

<変数>

→ たくさんの標本

↓

母平均，母分散の推定値

↓

合成データを生成し，
その分布をみる

Rで，母平均と母分散から，データを合成

- rnorm(10, 100, sqrt(400))

rnorm(<合成したいデータ数> , <母平均値> , sqrt(<母分散値>))

母平均　１００，母分散　４００のとき

■　合成データの生成（サイズ：１０）

```
> rnorm(10, 100, sqrt(400))
 [1]  92.38700  91.20082 108.99707 107.13639  77.12594
 [6] 119.15499  84.30894  97.81648 119.54392 100.52586
>
```

■　合成データを生成し，
その後，小数点以下を四捨五入（サイズ：１０）

```
round( rnorm(10, 100,
sqrt(400) ) )
```

```
> round( rnorm(10, 100, sqrt(400)) )
 [1]  60  95  92  80  67 106 104  88 103 107
>
```

小数点以下の四捨五入には round を使う

Rで，母平均と母分散から，データを合成

＜変数＞

たくさんの標本

母平均，母分散の推定値

母平均　１００
母分散　４００

合成データを生成する

元の変数と**性質が同じ**ような合成データを生成

```
> round( rnorm(10, 100, sqrt(400)) )
 [1]  60  95  92  80  67 106 104  88 103 107
> 
```

Rで，母平均と母分散から，データを合成

```
> round( rnorm(10, 100, sqrt(400) ) )
 [1]  91  63 135  89  80 134 139 106  94  84
> round( rnorm(10, 100, sqrt(400) ) )
 [1] 108  73 103  60 109 105  73 115 116  66
> round( rnorm(10, 100, sqrt(400) ) )
 [1] 104 118  79 100 109 124  71 136 114  65
> round( rnorm(10, 100, sqrt(400) ) )
 [1]  68 117  70 121  89  86  54 115 144  99
```

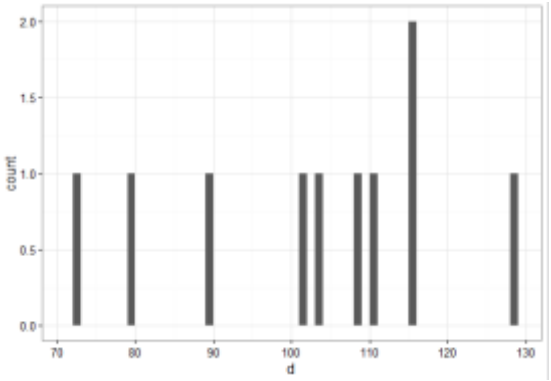
round(rnorm(10, 100, sqrt(400)))
round(rnorm(10, 100, sqrt(400)))
round(rnorm(10, 100, sqrt(400)))
round(rnorm(10, 100, sqrt(400)))

```
> round( rnorm(20, 100, sqrt(400) ) )
 [1]  79 101 134  91  98 117 144 108  89  98  78 113 111 112  81  76
 115
[18]  88 124  81
> round( rnorm(30, 100, sqrt(400) ) )
 [1] 100 108  96 134  97 106 119 109  70  70 106  84  86  89 127 100
  71
[18]  76  92  77  81  66 114 127  93  83  86 108  67  79
>
```

round(rnorm(20, 100, sqrt(400)))
round(rnorm(30, 100, sqrt(400)))

合成データの頻度分布（ヒストグラム）

```
> d
 [1]  89 128 103 110 108 115 101 115  72  79
>
```

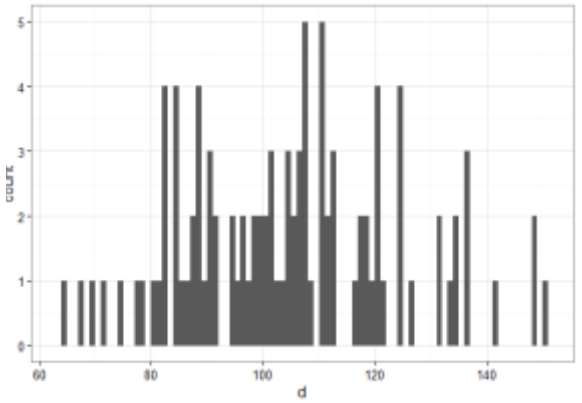



```
library(dplyr)
library(ggplot2)
d <- round( rnorm(10, 100, sqrt(400) ) )
ggplot(data_frame(d), aes(x = d)) +
  geom_histogram(binwidth=1) +
  theme_bw()
```

ベクトルデータの
頻度分布（ヒストグラム）

合成データの頻度分布（ヒストグラム）

```
> d
  [1] 107  95 124 134  98  67  78 107 148 107  88 111 120  88 106 136  89
 [18] 133  80  90  84  82  99 124  84 104  88 118  97 150 101  94 120 104
 [35]  87 100 100  82 131 112 131 111  91 106 110 124 134  77  85 105  74
 [52]  82  71  84  87  96 136 119 107  94 108 141  64 101  82  90 120  91
 [69]  84 106 110  96  81 103  98 110 110 126  88 136 120  69 102 118 104
 [86] 112 105 148 107 110  99  90 124 112 121  86 117 116 117 101
> 
```

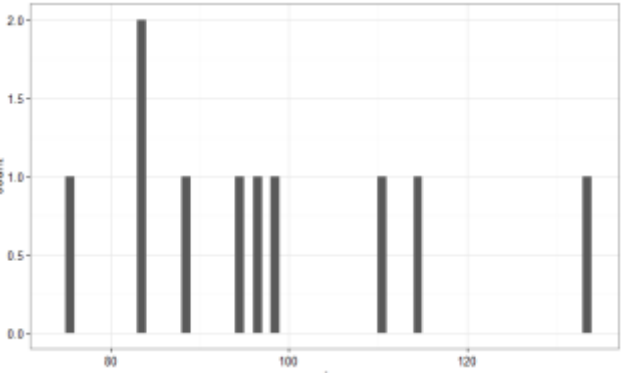


今度は 100

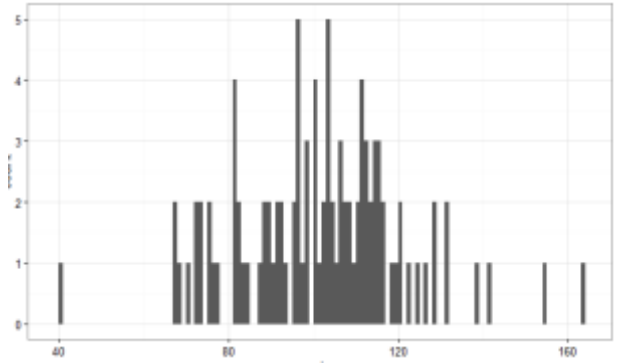
ベクトルデータの
頻度分布（ヒストグラム）

```
library(dplyr)
library(ggplot2)
d <- round( rnorm(100, 100, sqrt(400) ) )
ggplot(data_frame(d), aes(x = d)) +
  geom_histogram(binwidth=1) +
  theme_bw()
```

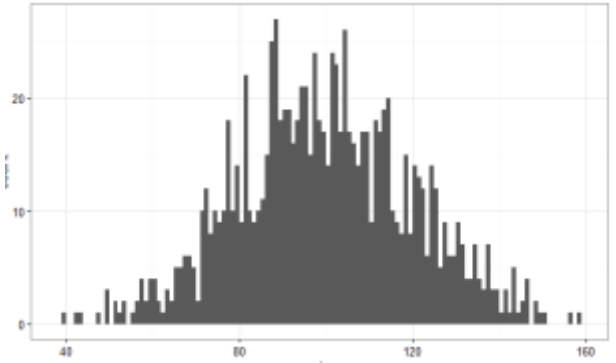
合成データの頻度分布（ヒストグラム）（１／２）



サイズ１０のときの頻度分布



サイズ１００のときの頻度分布

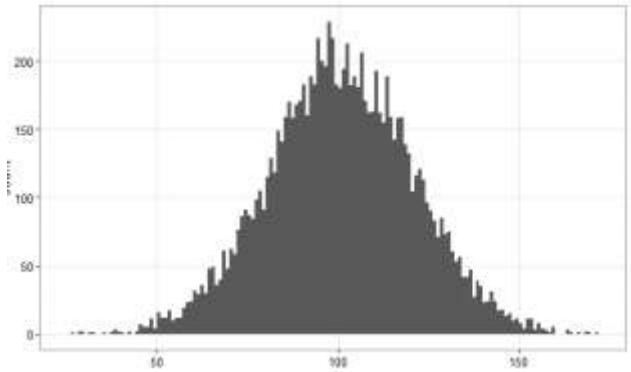


サイズ１０００のときの頻度分布

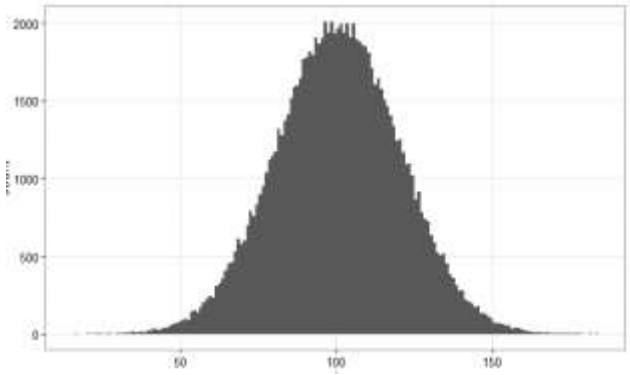
合成データの頻度分布（ヒストグラム）（２／２）

- 母平均と母分散で，合成された合成データの頻度分布（ヒストグラム）は，合成データのサイズを増やすと，正規分布になる

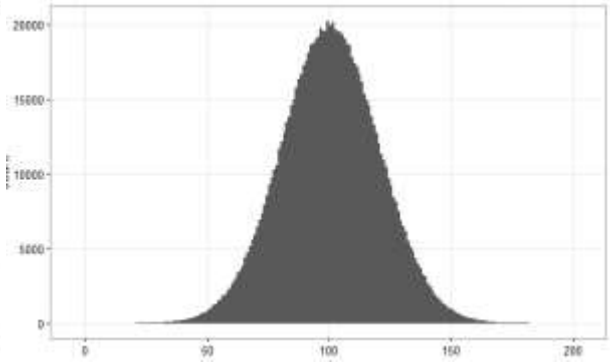
合成データのサイズを増やすほど，
頻度分布（ヒストグラム）のカーブは
滑らかになる



サイズ１００００のときの頻度分布



サイズ１０００００のときの頻度分布



サイズ１０００００００のときの頻度分布