

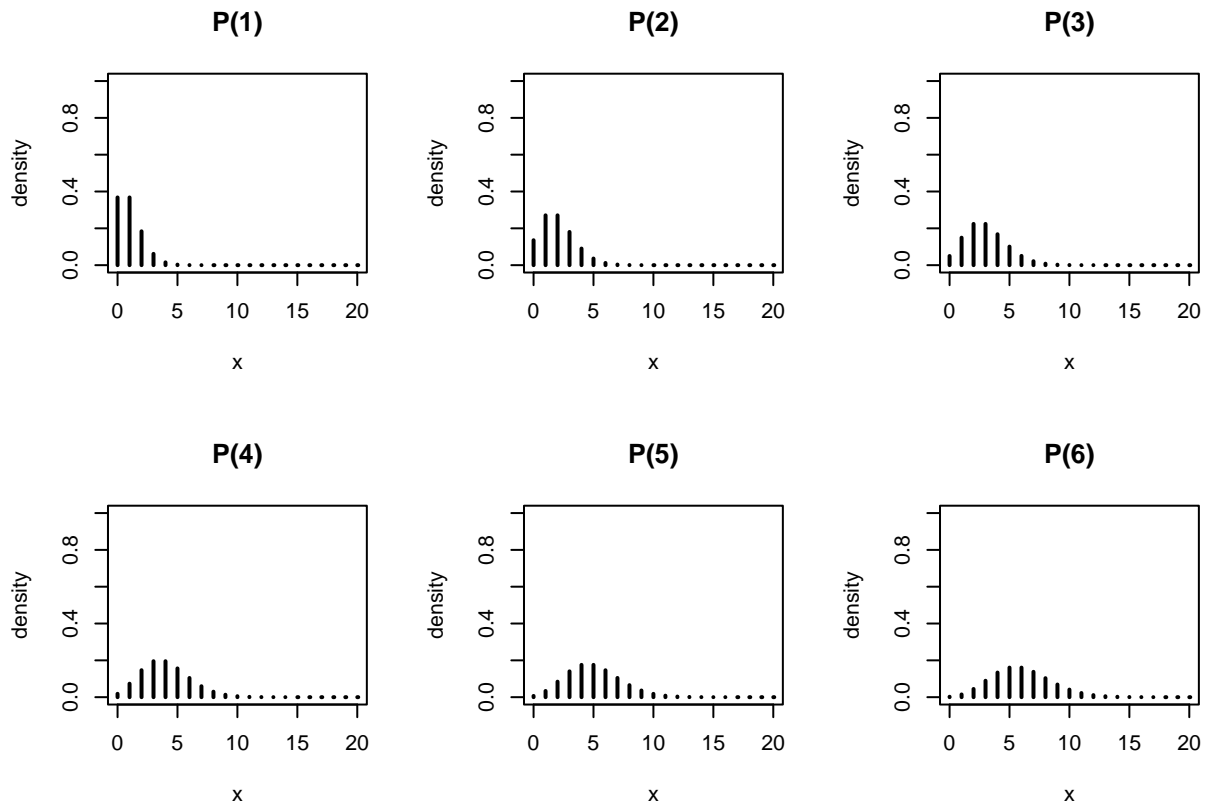
E05 上机实验 05

郑盼盼

2024-10-30

1. 绘制 $P(1), P(2), P(3), P(4), P(5), P(6)$ 的在区间 $[0, 20]$ 范围上的密度函数图像，总结泊松分布的密度图像随着参数 λ 增加的变化规律。

```
x <- 0:20 # 根据区间 [0,20] 定义一个向量 x=0:20，之后计算泊松分布在该向量每个点
↪ 上的密度
par(mfrow=c(2,3)) # 设置多张图的布局，此处需要绘制 6 张图，因此选择，2 行 3 列的
↪ 布局
# 遍历 1:6 并赋给循环变量 lambda
for (lambda in 1:6) {
  y <- dpois(x, lambda) # 计算 P(lambda) 在 {0,1,...,20} 点上的密度并赋给 y
  plot(x,y,           # 以 x 为横坐标，y 为纵坐标绘制图像
       type="h", # 设置图的类型为“直方图状的垂直线”，即每个点从 x 轴引出一条垂
       ↪ 直线到数据点
       lwd=2,    # 设置线的宽度为 2, lwd: line's width
       xlim=c(0,20), # 设置 x 轴的范围为 [0,20], xlim: x's limitations
       ylim=c(0,1), # 设置 y 轴的范围为 [0,1], ylim: y's limitations
       xlab="x", # 设置 x 轴的标签为 "x"
       ylab="density", # 设置 y 轴的标签为 "density"
       main=paste("P(", lambda, ")", sep="")) # 设置图的标题为 P(lambda)
}
```



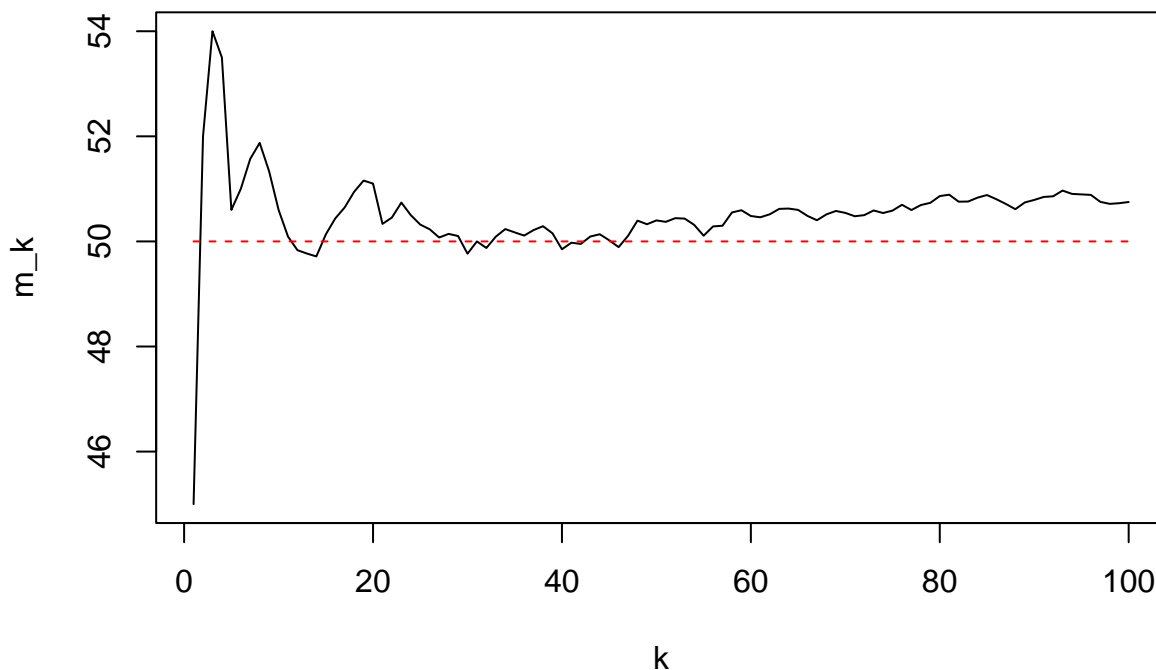
2. 已知 $\xi \sim P(50)$ ，模拟 ξ 的 100 次重复观测数据。用 m_k 表示这些数据中的前 k 个数据的算术平均值，绘制依次连接点

$$(1, m_1), (2, m_2), \dots, (100, m_{100})$$

的折线图，观察随着横坐标的增加折线的变化趋势，解释其中的原因。

```
set.seed(1) # 设置随机数生成器的种子的函数。使随机数生成具有可重复性。设定种子
            ↪ 后，每次运行同样的随机过程（如生成随机数、随机采样等）都会得到相同的结果。
m <- rpois(100,50) # 生成 100 个服从 P(50) 的随机数
mean_k <- c()      # 生成一个空向量命名为 mean_k 用于存储 m 中前 k 个元素的均值
# 遍历 1 到 100 的整数，循环变量为 k
for (k in 1:100) {
  mean_k <- c(mean_k, mean(m[1:k])) # 计算 m 中前 k 个元素的均值并添加为 mean_k
  ↪ 的最后一个元素
}
plot(1:100, mean_k, # 以 1:100 为横坐标，mean_k 为纵坐标，绘制图像
     type="l",      # 设置图的类型为折线图；l: lines
     xlab="k",      # 设置 x 轴的标签为 "k"
     ylab="m_k"     # 设置 y 轴的标签为 "m_k"
     )
# 绘制 P(50) 的数学期望
```

```
lines(c(1,100), c(50,50),
      lty=2, # 设置线的类型为虚线
      col="red" # 设置线的颜色为红色
    )
```



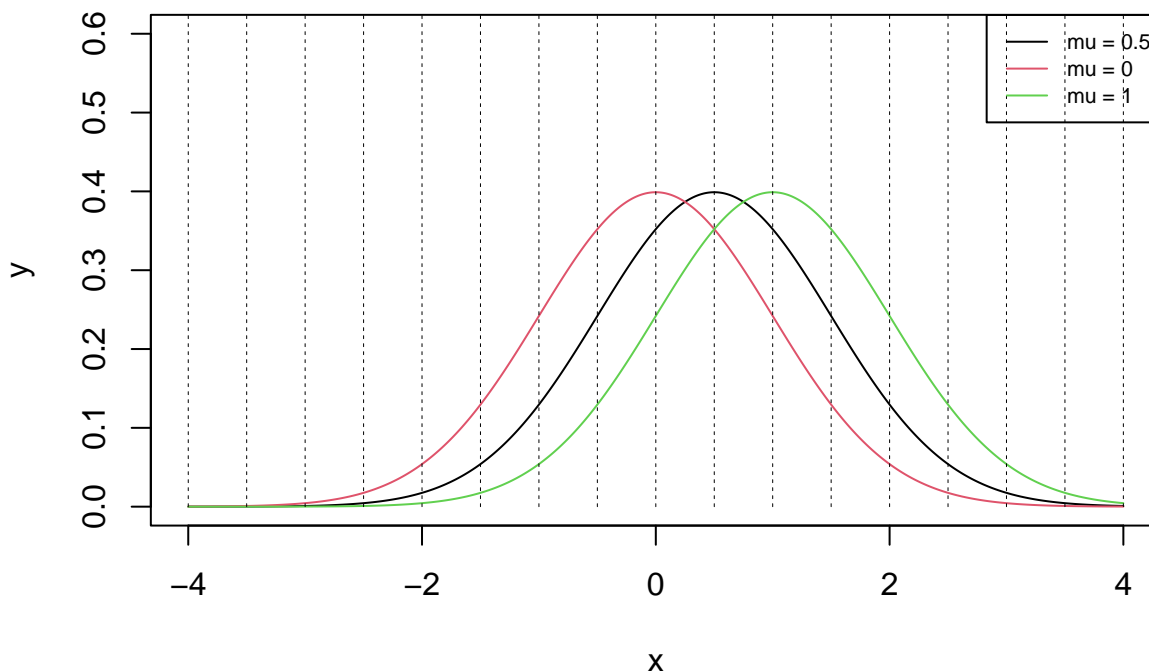
3. 在同一直角坐标系内，用不同颜色绘制 $N(0.5, 1)$, $N(0, 1)$ 和 $N(1, 1)$ 的密度函数曲线（横坐标限定在 $[-4, 4]$ 内），说明正态分布密度函数的峰值位置与其数学期望之间的关系。

```
mu <- c(0.5, 0, 1) # 将不同的参数 mu 定义为一个向量 mu
x <- seq(-4, 4, by=1e-3) # 根据横坐标的范围定义一个向量 x 其为从 -4 到 4 的等差数
# 列，公差为 0.001
y <- dnorm(x, mu[1], 1) # 计算 N(mu[1], 1) 在每个 x 点的密度
plot(x, y, # 以 x 为横坐标，y 为纵坐标绘制图像
     type="l", # 图的类型为折线图；l: lines
     col=1, # 线的颜色为 1 (1, 2, 3 在 R 中有已经定义好的不同颜色)
     lwd=1, # 线的宽度 1; lwd: lines' width
     xlim=c(-4, 4), # 设置 x 轴的范围为 -4 到 4
     ylim=c(0, 0.6) # 设置 y 轴的范围为 0 到 0.6
)
# 使用 for 循环添加在 mu[2], mu[3] 为参数时，正态分布的概率密度图像
for(i in c(2, 3)){
  y <- dnorm(x, mu[i], 1) # 计算 N(mu[i], 1) 在 x 点的密度
  lines(x, y, # 以 x 为横轴，y 为纵轴绘制折线图
```

```

        col=i, # 设置线的颜色为 i
        lwd=1  # 设置线的宽度为 1
    )
}
# 使用 for 循环添加垂直于 x 轴的虚线，方便检查密度函数图像峰的位置
for (i in seq(-4,4,by=.5)){
    lines(c(i), c(10),
          type="h",
          lwd=.5,
          lty=2)
}
# 添加图例，注释不同颜色对应的参数值
legend("topright", # 设置图例的位置在右上角
      legend = paste("mu =",mu), # 图例的内容为 "mu=i"
      col = 1:3, # 对应的颜色为 1:3
      lty=rep(1,3), # 对应的线形均为 1 (实线)
      cex=0.7, # 设置图例的大小
      )

```

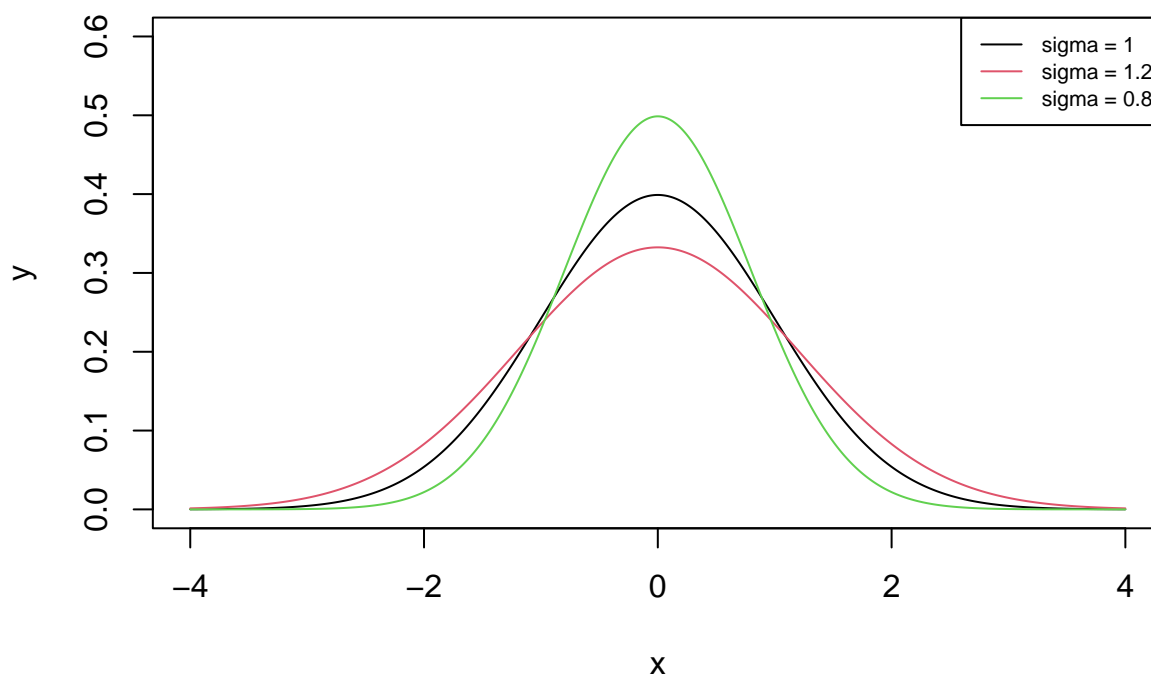


4. 在同一直角坐标系内，用不同颜色绘制 $N(0,1)$, $N(0,1.44)$, $N(0,0.64)$ 的密度函数曲线（横坐标限定在 $[-4,4]$ 内），说明正态分布密度函数的峰值陡峭程度与其方差之间的关系。

```

# 整体类似于上面的代码，只是从变化 mu 变成了变化 sigma
sigma <- c(1, 1.2, 0.8)
x <- seq(-4,4,by=1e-3)
y <- dnorm(x, 0, sigma[1])
plot(x, y,
      col=1,
      type="l",
      lwd=1,
      xlim=c(-4,4),
      ylim=c(0,0.6))
for(i in c(2,3)){
  y <- dnorm(x, 0, sigma[i])
  lines(x,y,
        col=i,
        lwd=1)
}
legend("topright",
      legend = paste("sigma =",sigma),
      col = 1:3,
      lty=rep(1,3),
      cex=0.7,
      )

```



5. 利用 R 程序代码计算 $N(0, 9)$ 的数学期望和方差

- **数学期望**-根据连续型随机变量数学期望的定义,若已知连续性随机变量 X 的概率密度函数 $f(x)$, 则其数学期望为

$$E(X) = \int_{-\infty}^{\infty} t f(t) dt$$

根据题意, 我们需要计算:

$$\int_{-\infty}^{\infty} t \cdot \varphi_{0,3}(t) dt$$

可通过如下 R 程序代码计算:

```
# 将我们需要积分的函数  $g(x) = x * f(x)$  封装成函数  $g$ 
g <- function(x){
  return(x * dnorm(x, 0, 3)) # 返回  $x$  和  $N(0,3)$  在  $x$  点概率密度的乘积
}
mean_norm <- integrate(g, -Inf, Inf) # 对函数  $g$  进行积分, 上下限均为无穷
mean_norm
```

```
## 0 with absolute error < 0
```

- **方差**: 根据连续型随机变量方差的定义: 若随机变量 X 的概率密度函数为 $f(x)$, 则其方差为:

$$D(X) = \int_{-\infty}^{\infty} [t - E(X)]^2 \cdot f(t) dt$$

根据题意可得, 我们需要计算定积分:

$$\int_{-\infty}^{\infty} (t - 0)^2 \cdot \varphi_{0,3}(t) dt$$

可通过如下 R 程序代码实现:

```
# 使用匿名函数的方式进行积分:
## function(x) (x-0)**2 * dnorm(x,0,3) 对应我们的被积函数
var_norm <- integrate(function(x) (x - 0) ** 2 * dnorm(x, 0, 3), -Inf, Inf)
var_norm
```

```
## 9 with absolute error < 1.9e-05
```