短学期上机考试

汪利军 *3140105707*

July 11, 2017

Contents

1 六

```r
before = c(57, 54, 62, 64, 71, 65, 70, 75, 68, 70, 77, 74, 80, 83)
after = c(55, 60, 68, 69, 70, 73, 74, 74, 75, 76, 76, 78, 81, 90)
t.test(before, after, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  before and after
## t = -3.6927, df = 13, p-value = 0.002707
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.547628 -1.452372
## sample estimates:
## mean of the differences
##                    -3.5
```

由上述 t 检验结果可以看出，p 值为 0.002707<0.05，也就是拒绝原假设，故 alcohol 对 reflexes 有效果。

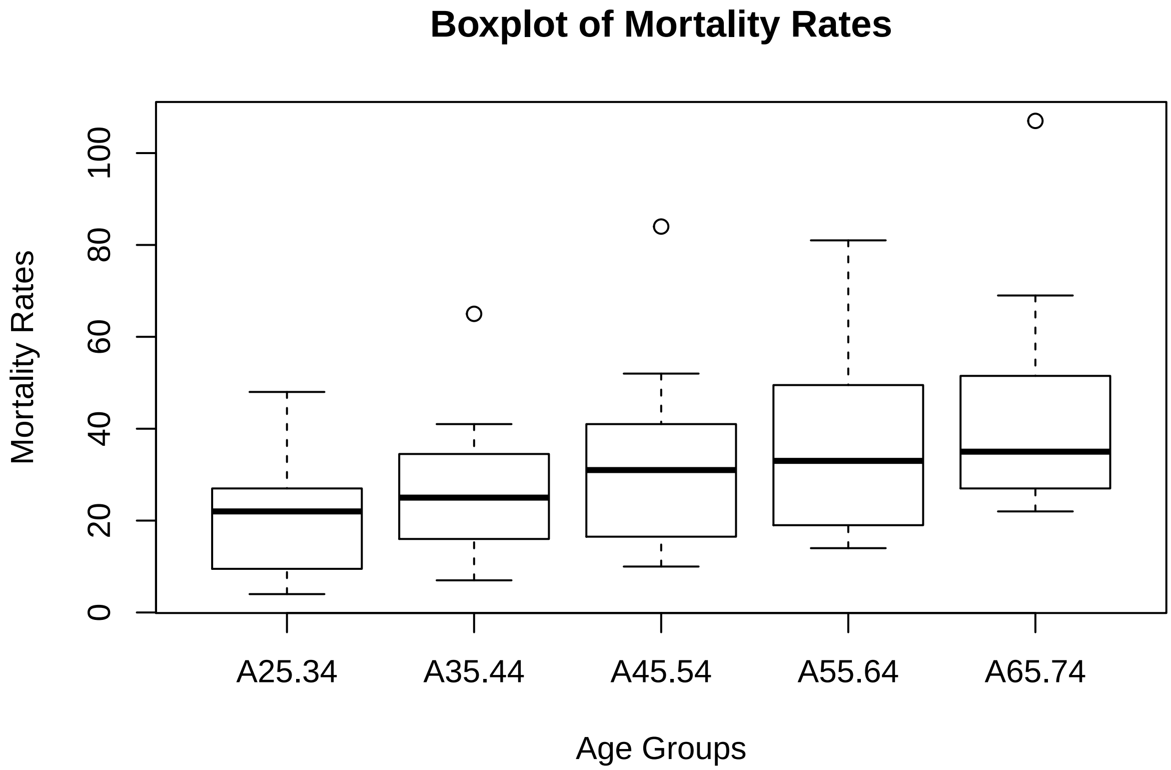
2 七

首先构造数据如下所示

```r
A25.34 <- c(22, 9, 22, 29, 16, 28, 48, 7, 8, 26, 4, 28, 22, 10, 20)
A35.44 <- c(27, 19, 19, 40, 25, 35, 65, 8, 11, 29, 7, 41, 34, 13, 22)
A45.54 <- c(31, 10, 21, 52, 36, 41, 84, 11, 18, 36, 10, 46, 41, 15, 28)
A55.64 <- c(34, 14, 31, 53, 47, 49, 81, 18, 20, 32, 16, 51, 50, 17, 33)
A65.74 <- c(24, 27, 49, 69, 56, 52, 107, 27, 28, 28, 22, 35, 51, 22, 37)
df <- data.frame(A25.34, A35.44, A45.54, A55.64, A65.74)
rownames(df) <- c("Canada", "Israel", "Japan", "Austria", "France",
                  "Germany", "Hungary", "Italy", "Netherlands", "Poland",
                  "Spain", "Sweden", "Switzerland", "UK", "USA")
```

首先对数据框进行转换，然后进一步绘出箱线图

```r
df.stack = stack(df)
boxplot(values~ind, data = df.stack, main = "Boxplot of Mortality Rates",
        xlab = "Age Groups", ylab = "Mortality Rates")
```



从箱线图可以看出，不同年龄群的 Mortality Rates 有差异，其均值随着年龄的增长而增大。

3 八

3.1 (a)

重复 n 次实验，点 (x_1, x_2) 到最近边的距离小于 0.25 的个数为 m_1，则估计的概率为

$$p(\text{the distance between }(x_1, x_2)\text{ and the nearest edge}) = \frac{m_1}{n}$$

编写 simPoints(n) 函数返回生成的 n 个随机点。

```
simPoints <- function(n)
{
  x1 = runif(n)
  x2 = runif(n)
  return(data.frame(x1, x2))
}
```

编写 isNeareastPointToEdge(x) 函数来判断输入的 x 点是否离最近边的距离小于 0.25，若是，则返回 TRUE，否则返回 FALSE。

```
isNeareastPointToEdge <- function(x)
{
  # 点 x 到四条边的距离
  dist = sapply(x, function(y) c(y, 1-y))
  # 点 x 到四条边最近的距离
  dist.min = min(dist)
  # 若小于 0.25, 返回 TRUE
  if (dist.min < 0.25)
    return(TRUE)
  else
    return(FALSE)
}
```

统计满足条件的点的个数

```
n = 1000
x <- simPoints(n)
x.near.edge <- apply(x, 1, function(y) isNeareastPointToEdge(y))
m1 = sum(x.near.edge)
```

则概率为

```
m1/n
```

```
## [1] 0.736
```

3.2 (b)

重复 n 次实验，点 (x_1, x_2) 到最近点的距离小于 0.25 的个数为 m_2，则估计的概率为

$$p(\text{the distance between } (x_1, x_2) \text{ and the nearest vertex}) = \frac{m_2}{n}$$

编写函数 isNeareastPointToVertex(x)，判断该点是否离最近顶点的距离小于 0.25，若是，则返回 TRUE，否则返回 FALSE。

```
isNeareastPointToVertex <- function(x)
{
  # 点 x 到四个顶点的距离
  dist = c(sqrt(x[1]^2+x[2]^2),
           sqrt((x[1]-1)^2+x[2]^2),
           sqrt(x[1]^2+(x[2]-1)^2),
           sqrt((x[1]-1)^2+(x[2]-1)^2))
  # 点 x 到四个顶点最近的距离
  dist.min = min(dist)
  # 若小于 0.25, 返回 TRUE
  if (dist.min < 0.25)
    return(TRUE)
  else
    return(FALSE)
}
```

统计满足条件的点的个数

```
x.near.vertex <- apply(x, 1, function(y) isNeareastPointToVertex(y))
m2 = sum(x.near.vertex)
```

则概率为

```
m2/n
```

```
## [1] 0.192
```

4 九

4.1 (a)

```
ChickWeight.split = with(ChickWeight, split(ChickWeight, Chick))
```

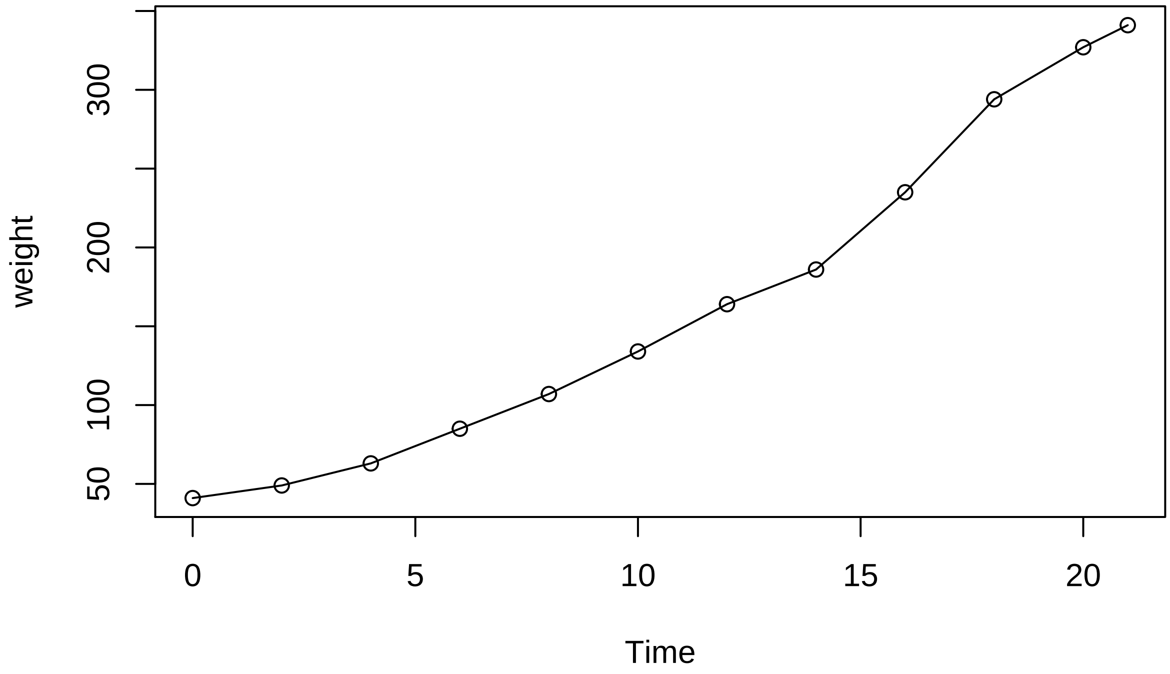
提取出 Chick 为 34 的数据

```
ChickWeight.chick34 <- ChickWeight.split$`34`
ChickWeight.chick34
```

```
##     weight Time Chick Diet
## 377     41    0    34    3
## 378     49    2    34    3
## 379     63    4    34    3
## 380     85    6    34    3
## 381    107    8    34    3
## 382    134   10    34    3
## 383    164   12    34    3
## 384    186   14    34    3
## 385    235   16    34    3
## 386    294   18    34    3
## 387    327   20    34    3
## 388    341   21    34    3
```

然后作图

```
plot(weight ~ Time, data = ChickWeight.chick34, type = "o")
```



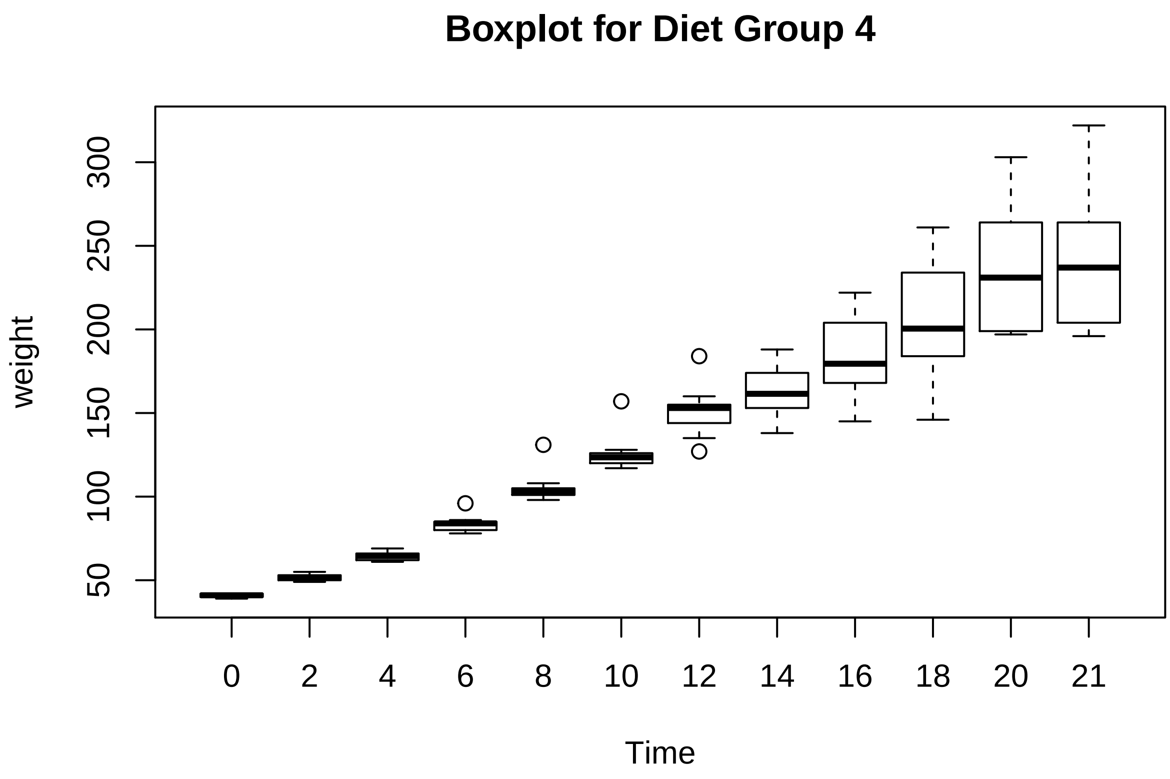
4.2 (b)

首先提取出 diet group 为 4 的数据

```
ChickWeight.split.diet = with(ChickWeight, split(ChickWeight, Diet))
ChickWeight.diet4 <- ChickWeight.split.diet$`4`
head(ChickWeight.diet4)
```

```
##     weight Time Chick Diet
## 461     42    0    41    4
## 462     51    2    41    4
## 463     66    4    41    4
## 464     85    6    41    4
## 465    103    8    41    4
## 466    124   10    41    4
```

```
boxplot(weight ~ Time, data = ChickWeight.diet4,
        xlab = "Time", ylab = "weight", main = "Boxplot for Diet Group 4")
```



4.3 (c)

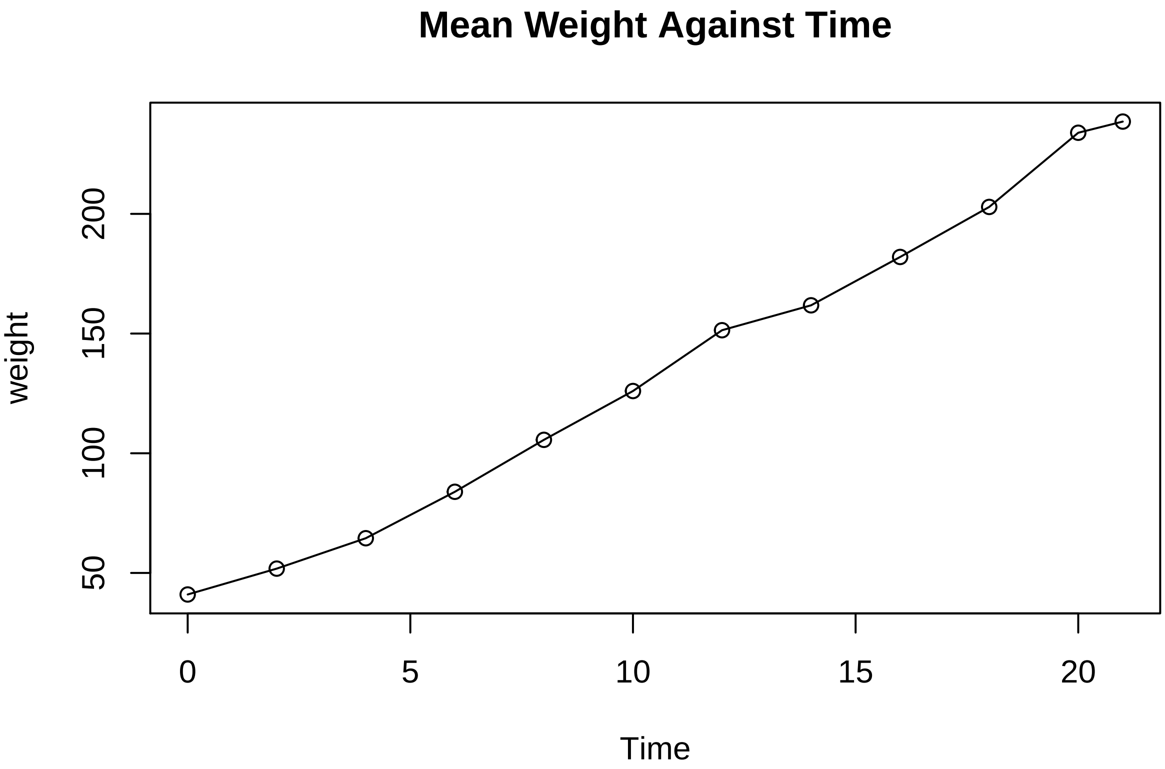
计算每个时间点 weight 的均值

```
ChickWeight.diet4.mean <- with(ChickWeight.diet4,
                               aggregate(weight, by = list(Time), mean))
colnames(ChickWeight.diet4.mean) <- c("Time", "weight")
ChickWeight.diet4.mean
```

```
##    Time   weight
## 1     0  41.0000
## 2     2  51.8000
## 3     4  64.5000
## 4     6  83.9000
## 5     8 105.6000
## 6    10 126.0000
## 7    12 151.4000
## 8    14 161.8000
## 9    16 182.0000
## 10   18 202.9000
## 11   20 233.8889
## 12   21 238.5556
```

作出 weight 的均值关于时间的图象

```
plot(weight~Time, data = ChickWeight.diet4.mean, type = "o",
     main = "Mean Weight Against Time")
```



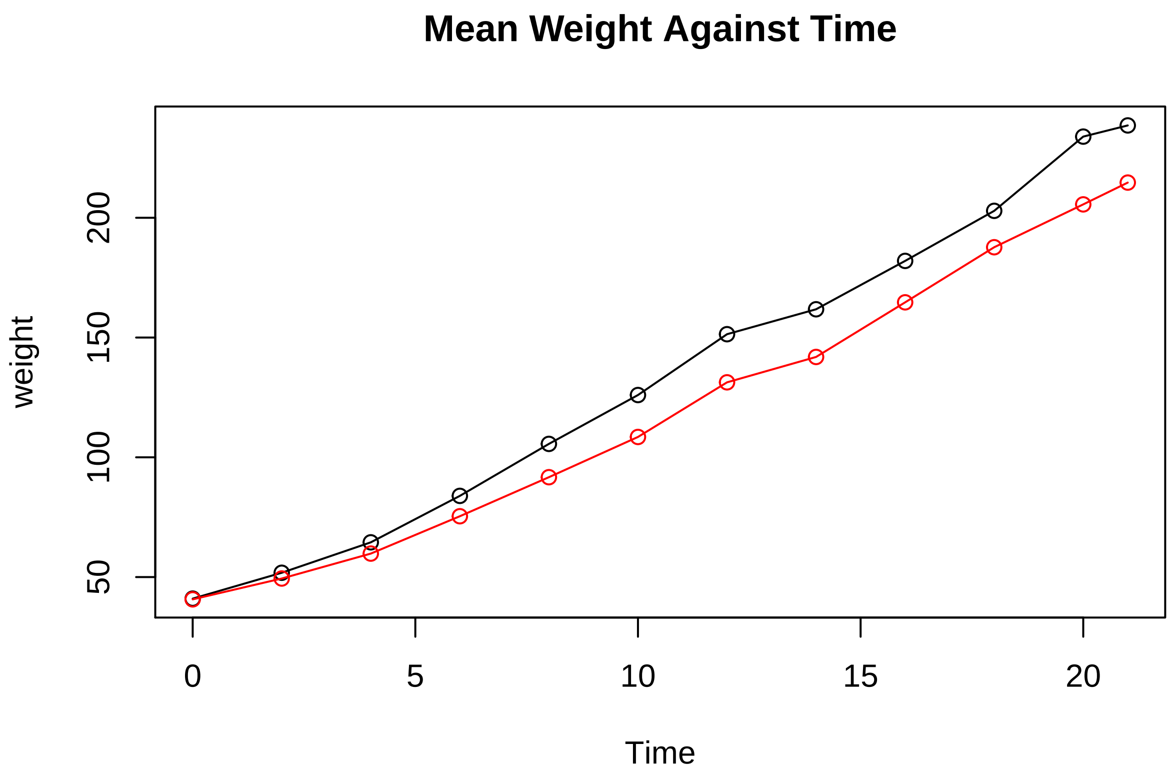
4.4 (d)

首先提取 group 2 的数据

```
ChickWeight.diet2 <- ChickWeight.split.diet$`2`
ChickWeight.diet2.mean <- with(ChickWeight.diet2,
                               aggregate(weight, by = list(Time), mean))
colnames(ChickWeight.diet2.mean) <- c("Time", "weight")
```

再向图中添加 group 2 的图象

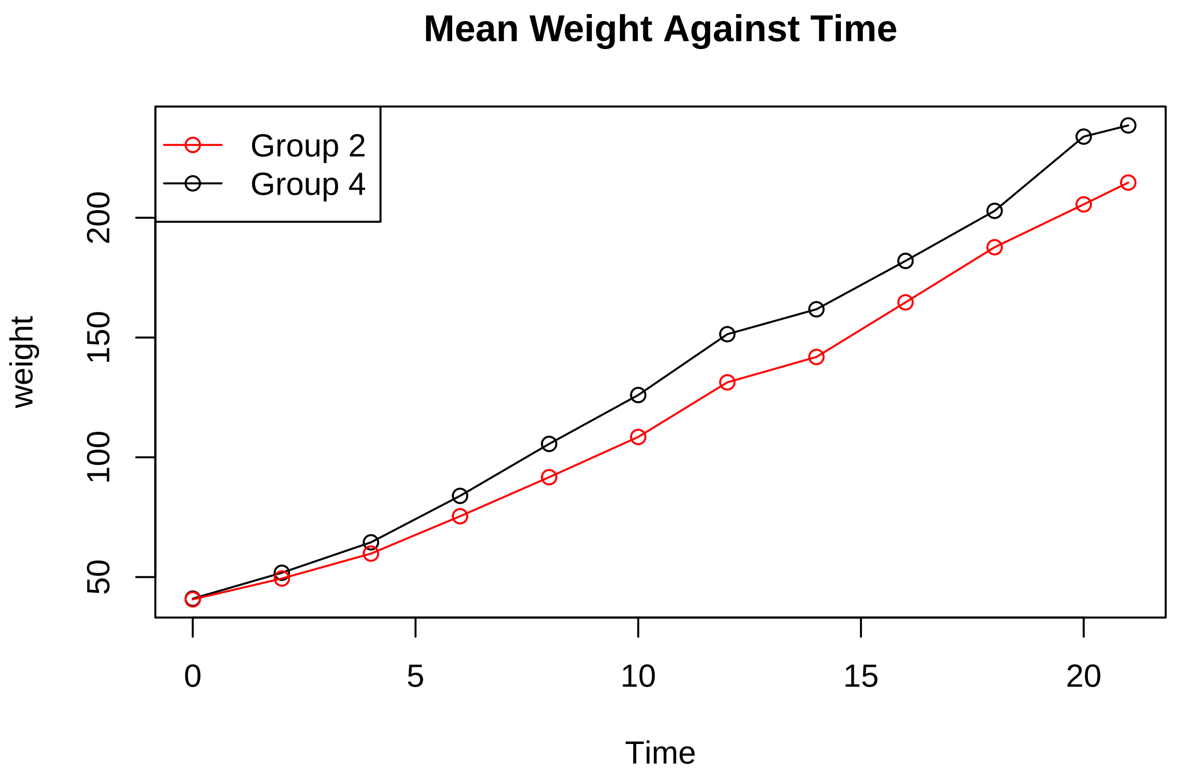
```
plot(weight~Time, data = ChickWeight.diet4.mean, type = "o",
     main = "Mean Weight Against Time")
with(ChickWeight.diet2.mean, points(Time, weight, col = "red", type="o"))
```



4.5 (e)

在左上角添加 legend，效果如下所示

```
plot(weight~Time, data = ChickWeight.diet4.mean, type = "o",
     main = "Mean Weight Against Time")
with(ChickWeight.diet2.mean, points(Time, weight, col = "red", type="o"))
legend("topleft", c("Group 2", "Group 4"), lty = c(1, 1),
       pch = c(1, 1), col = c("red", "black"))
```



5 十

首先构造数据框

```
time = c(2, 3, 6, 8, 9, 10, 11, 13, 14, 16, 21, 22, 24, 26, 27, 7,
         13, 15, 18, 23, 20, 24, 1, 5, 17, 18, 25, 18, 25, 4, 19)
tumorsize = c(rep("<=3cm", 22), rep(">3cm", 9))
number = c(rep(1, 15), rep(2, 5), 3, 4, rep(1, 5), 2, 2, 3, 4)
df = data.frame(time, tumorsize, number)
```

因 tumorsize 仅取两个值，则将其看成因子然后进行下面的下面的 Poisson 回归

```
model <- glm(number ~ time + factor(tumorsize), data = df, family = "poisson")
```

结果为

```
summary(model)
```

```
##
## Call:
## glm(formula = number ~ time + factor(tumorsize), family = "poisson",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8183  -0.4753  -0.2923   0.3319   1.5446
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.14568    0.34766   0.419    0.675
## time                   0.01478    0.01883   0.785    0.433
## factor(tumorsize)>3cm  0.20511    0.30620   0.670    0.503
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 12.800  on 30  degrees of freedom
## Residual deviance: 11.757  on 28  degrees of freedom
## AIC: 88.568
##
## Number of Fisher Scoring iterations: 4
```

可以看出 tumorsize 的系数为 0.20511，在泊松回归中, 因变量以条件均值的对数形式 $ln(\lambda)$ 来建模，则 tumorsize 变大，number 的对数均值将增加 0.20511，然而该系数的 p 值为 0.503>0.05，不够显著。

如果不考虑时间，采用下面的 Poisson 回归，

```
model2 <- glm(number ~ factor(tumorsize), data = df, family = "poisson")
```

结果为

```
summary(model2)
```

```
##
## Call:
## glm(formula = number ~ factor(tumorsize), family = "poisson",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6363  -0.3996  -0.3996   0.4277   1.7326
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)             0.3747     0.1768   2.120    0.034 *
## factor(tumorsize)>3cm   0.2007     0.3062   0.655    0.512
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 12.80  on 30  degrees of freedom
## Residual deviance: 12.38  on 29  degrees of freedom
## AIC: 87.191
##
## Number of Fisher Scoring iterations: 4
```

可以看出 tumorsize 的系数为 0.2007，在泊松回归中, 因变量以条件均值的对数形式 $ln(\lambda)$ 来建模，则 tumorsize 变大，number 的对数均值将增加 0.2007，然而该系数的 p 值为 0.512>0.05，表明该系数的估计不够显著。