**实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**： **应用回归分析实验** | **学期：2023年秋季学期** | **成绩**： |
| **指导教师**：**陈丹** | **学生姓名**： **枫叶** | **学生学号**： |
| **实验名称**： **一元线性回归分析实验** | | |
| **实验编号**：2 | **实验日期**：2023.10.12 | **实验学时**： |
| **学院： 数学与统计学院** | **专业： 统计学** | **年级**： **2021级** |

1. **实验目的**

了解一元线性回归的建模过程。

掌握用R软件进行线性回归分析。

1. **使用环境**

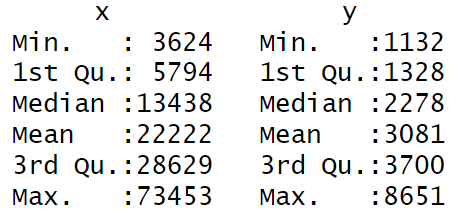
R语言

1. **实验内容**

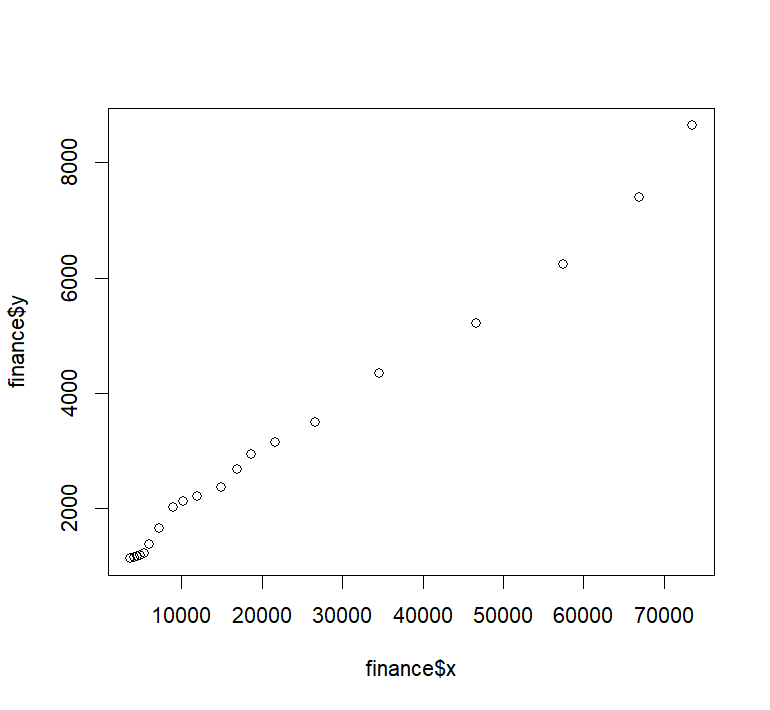
用R自带程序包进行一元线性回归分析。

自我编程实现一元线性回归分析。

1. **算法介绍及结果**



x、y的最小值、均值和最大值如图所示，画出如下散点图，猜测财政收入和国民生产总值之间存在线性关系，构建理论模型



经计算，得到估计模型为,这意味着国内生产总值每上升1亿元，财政收入平均增长0.1亿元。对1998年的预测值为8662.491，预测区间为(8646.869,8678.114)

|  |
| --- |
| #实验二  library(readxl)  finance <- read\_xlsx("D:/预删除文件夹/大三上/应用回归分析/task2数据.xlsx")  plot(finance$x,finance$y)  #自定义一元线性回归函数  my\_lm <- function(x, y, alpha) {  n <- length(x)  #计算相关系数  r <- cor(x, y)  #计算斜率和截距  b <- r \* sd(y) / sd(x)  a <- mean(y) - b \* mean(x)  # 计算t统计量和显著性检验  df\_residual <- n - 2  t\_statistic <- b / (sd(y) / sqrt(sum((x - mean(x))^2) / df\_residual))  p\_value <- 2 \* (1 - pt(abs(t\_statistic), df\_residual))  # 计算F统计量  F\_statistic <- (r^2 \* (n - 2)) / (1 - r^2)  # 计算参数的区间估计  t\_critical <- qt(1 - alpha / 2, df\_residual)  se\_a <- sqrt((1 / n + (mean(x)^2) / sum((x - mean(x))^2)) \* (1 - r^2) \* var(y) / n)  se\_b <- sqrt(var(y) / sum((x - mean(x))^2))  a\_ci <- a + c(-t\_critical \* se\_a, t\_critical \* se\_a)  b\_ci <- b + c(-t\_critical \* se\_b, t\_critical \* se\_b)  # 输出结果  result <- list(  correlation\_coefficient = r,  intercept = a,  slope = b,  t\_statistic = t\_statistic,  F\_statistic = F\_statistic,  intercept\_ci = a\_ci,  slope\_ci = b\_ci  )  return(result)  }  #自定义预测区间估计函数  predict\_interval <- function(new\_x, lm\_result, alpha, point = TRUE) {  b <- lm\_result$slope  a <- lm\_result$intercept  t\_c <- qt(1 - alpha / 2, length(lm\_result$y) - 2)  se <- sqrt(sum((lm\_result$x - mean(lm\_result$x))^2) / (length(lm\_result$y) - 2))  pred <- a + b \* new\_x  if (point) {  pred\_se <- se \* sqrt(1 + 1 / length(lm\_result$y) + ((new\_x - mean(lm\_result$x))^2) / sum((lm\_result$x - mean(lm\_result$x))^2))  pred\_ci <- pred + c(-t\_c \* pred\_se, t\_c \* pred\_se)  } else {  mean\_pred\_se <- se \* sqrt(1 / length(lm\_result$y) + ((new\_x - mean(lm\_result$x))^2) / sum((lm\_result$x - mean(lm\_result$x))^2))  pred\_ci <- pred + c(-t\_c \* mean\_pred\_se, t\_c \* mean\_pred\_se)  }  return(pred\_ci)  }  #应用  alpha <- 0.05  results1 <- my\_lm(finance$x,finance$y,alpha)  results2 <- lm(data=finance,y~x)  result <- data.frame(cor = results1$correlation\_coefficient,  my\_intercept = results1$intercept,  my\_slope = results1$slope,  intercept = results2$coefficients[1],  slope = results2$coefficients[2])  rownames(result) <- "result"  result  predict(results2,new=data.frame(x=78017.8),level=0.05,interval="prediction") |

**六、参考文献**

**七、教师评语**