Homework 5: Pareto and Kuznets on the Grand Tour

We continue working with the World Top Incomes Database [https://wid.world], and the Pareto distribution, as in the lab. We also continue to practice working with data frames, manipulating data from one format to another, and writing functions to automate repetitive tasks.

We saw in the lab that if the upper tail of the income distribution followed a perfect Pareto distribution, then

$$\left(\frac{P99}{P99.9}\right)^{-a+1} = 10 \quad (1)$$

$$\left(\frac{P99.5}{P99.9}\right)^{-a+1} = 5 \quad (2)$$

$$\left(\frac{P99}{P99.5}\right)^{-a+1} = 2 \quad (3)$$

We could estimate the Pareto exponent by solving any one of these equations for a; in lab we used

$$a = 1 - \frac{\log 10}{\log{(P99/P99.9)}} \ , \quad (4)$$

Because of measurement error and sampling noise, we can't find find one value of a which will work for all three equations (1)–(3). Generally, trying to make all three equations come close to balancing gives a better estimate of a than just solving one of them. (This is analogous to finding the slope and intercept of a regression line by trying to come close to all the points in a scatterplot, and not just running a line through two of them.)

```
library(dplyr)    # 数据处理
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)  # 绘图
library(readr)    # 读取 CSV 文件
```

1. We estimate a by minimizing

   $$\left(\left(\frac{P99}{P99.9}\right)^{-a+1}-10\right)^2+\left(\left(\frac{P99.5}{P99.9}\right)^{-a+1}-5\right)^2+\left(\left(\frac{P99}{P99.5}\right)^{-a+1}-2\right)^2$$

   Write a function, `percentile_ratio_discrepancies`, which takes as inputs `P99`, `P99.5`, `P99.9` and `a`, and returns the value of the expression above. Check that when `P99=1e6`, `P99.5=2e6`, `P99.9=1e7` and `a=2`, your function returns `0`.

```
# 编写函数 percentile_ratio_discrepancies
# 计算百分位数比值与理论值的偏差平方和

percentile_ratio_discrepancies <- function(P99, P99.5, P99.9, a) {
  # 输入参数:
  # P99: 99% 分位数
  # P99.5: 99.5% 分位数
  # P99.9: 99.9% 分位数
  # a: 帕累托指数

  # 根据帕累托分布理论，计算三个比值的偏差
  # 第一个等式：(P99/P99.9)^(-a+1) = 10
  term1 <- ((P99 / P99.9)^(-a + 1) - 10)^2
  # 第二个等式：(P99.5/P99.9)^(-a+1) = 5
  term2 <- ((P99.5 / P99.9)^(-a + 1) - 5)^2
  # 第三个等式：(P99/P99.5)^(-a+1) = 2
  term3 <- ((P99 / P99.5)^(-a + 1) - 2)^2
  # 返回偏差平方和
  return(term1 + term2 + term3)
}

# 检验函数：当 P99=1e6, P99.5=2e6, P99.9=1e7, a=2 时，函数应返回 0
test_result_1 <- percentile_ratio_discrepancies(P99 = 1e6, P99.5 = 2e6,
                                                P99.9 = 1e7, a = 2)
cat("问题1检验结果：", test_result_1, "\n")  # 应该输出接近 0 的值
```

```
## 问题1检验结果： 0
```

2. Write a function, `exponent.multi_ratios_est`, which takes as inputs `P99`, `P99.5`, `P99.9`, and estimates `a`. It should minimize your `percentile_ratio_discrepancies` function. The starting value for the minimization should come from (4). Check that when `P99=1e6`, `P99.5=2e6` and `P99.9=1e7`, your function returns an `a` of 2.

```
# 编写函数 exponent.multi_ratios_est
# 通过最小化偏差函数来估计帕累托指数 a

exponent.multi_ratios_est <- function(P99, P99.5, P99.9) {
  # 输入参数：三个百分位数值
  # 返回：估计的帕累托指数 a

  # 使用公式 (4) 计算初始值作为优化起点
  initial_a <- 1 - log(10) / log(P99 / P99.9)
  # 定义目标函数（要最小化的函数）
  objective_function <- function(a) {
    return(percentile_ratio_discrepancies(P99, P99.5, P99.9, a))
  }

  # 使用optimize函数进行一维优化
  # 搜索区间设定为[0.5, 5]，覆盖常见的帕累托指数范围
  result <- optimize(objective_function,
                     interval = c(0.5, 5),  # 搜索区间
                     tol = 1e-10)           # 精度要求
  return(result$minimum)  # 返回最优的a值
}

# 检验函数：使用测试数据验证
test_result_2 <- exponent.multi_ratios_est(P99 = 1e6, P99.5 = 2e6, P99.9 = 1e7)
cat("问题2检验结果：估计的a值 =", test_result_2, "\n")  # 应该接近2
```

```
## 问题2检验结果：估计的a值 = 2
```

3. Write a function which uses `exponent.multi_ratios_est` to estimate a for the US for every year from 1913 to 2012. (There are many ways you could do thi, including loops.) Plot the estimates; make sure the labels of the plot are appropriate.

```
# 读取数据
data <- read_csv("data/wtid-report.csv")
```

```
## Rows: 100 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country
## dbl (7): Year, P90 income threshold, P95 income threshold, P99 income thresh...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# 查看数据结构
head(data)
```

```
## # A tibble: 6 x 8
##   Country        Year `P90 income threshold` `P95 income threshold`
##   <chr>         <dbl>                  <dbl>                  <dbl>
## 1 United States  1913                    NA                     NA 
## 2 United States  1914                    NA                     NA 
## 3 United States  1915                    NA                     NA 
## 4 United States  1916                    NA                     NA 
## 5 United States  1917                 27009.                 37405.
## 6 United States  1918                 28584.                 38189.
## # i 4 more variables: `P99 income threshold` <dbl>,
## #   `P99.5 income threshold` <dbl>, `P99.9 income threshold` <dbl>,
## #   `P99.99 income threshold` <dbl>
```

```
str(data)
```

```
## spc_tbl_ [100 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Country                : chr [1:100] "United States" "United States" "United Stat
##  $ Year                   : num [1:100] 1913 1914 1915 1916 1917 ...
##  $ P90 income threshold   : num [1:100] NA NA NA NA 27009 ...
##  $ P95 income threshold   : num [1:100] NA NA NA NA 37405 ...
##  $ P99 income threshold   : num [1:100] 80088 74013 62392 74869 92341 ...
##  $ P99.5 income threshold : num [1:100] 131337 122936 118717 133777 149698 ...
##  $ P99.9 income threshold : num [1:100] 415206 397672 437523 502094 519559 ...
##  $ P99.99 income threshold: num [1:100] 1746429 1735226 2219930 2749839 2370232 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Country = col_character(),
##   ..   Year = col_double(),
##   ..   `P90 income threshold` = col_double(),
##   ..   `P95 income threshold` = col_double(),
##   ..   `P99 income threshold` = col_double(),
##   ..   `P99.5 income threshold` = col_double(),
##   ..   `P99.9 income threshold` = col_double(),
##   ..   `P99.99 income threshold` = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

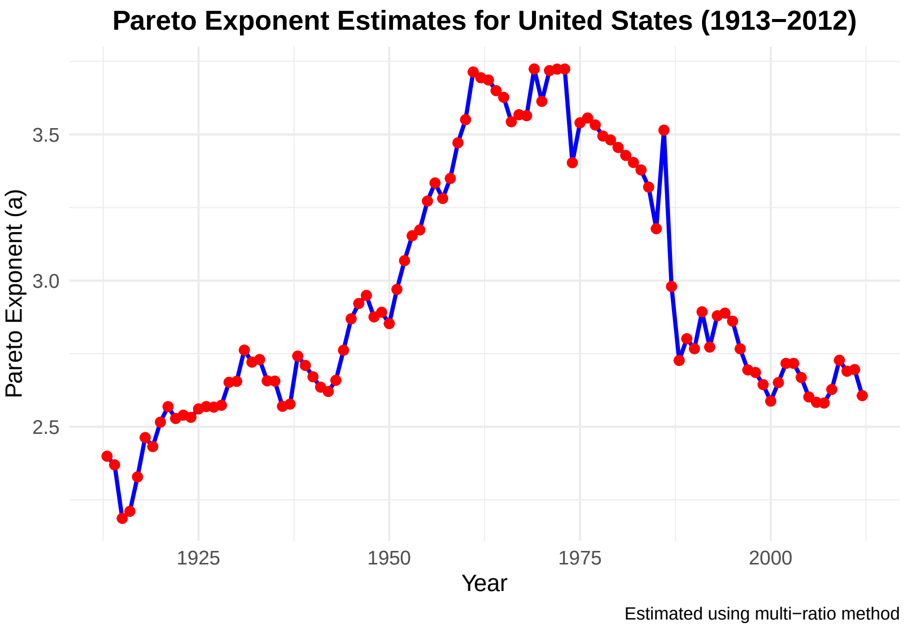
```
# 筛选美国数据、重命名列并选择需要的列
us_data <- data %>%
  # 重命名列，以匹配后续代码
  rename(
    P99 = `P99 income threshold`,
    P99.5 = `P99.5 income threshold`,
    P99.9 = `P99.9 income threshold`
  ) %>%
  filter(Country == "United States") %>%  # 筛选美国数据
  filter(Year >= 1913 & Year <= 2012) %>%  # 筛选年份范围
  select(Year, P99, P99.5, P99.9) %>%      # 选择需要的百分位数列
  filter(!is.na(P99) & !is.na(P99.5) & !is.na(P99.9))  # 移除缺失值

# 为每年估计 Pareto 指数
estimate_pareto_by_year <- function(data) {
  # 初始化结果向量
  years <- c()
  pareto_estimates <- c()

  # 遍历每年的数据
  for (year in unique(data$Year)) {
    year_data <- data[data$Year == year, ]
    # 检查是否有必要的百分位数据
    if (all(c("P99", "P99.5", "P99.9") %in% names(year_data)) &&
          !any(is.na(c(year_data$P99, year_data$P99.5, year_data$P99.9)))) {
      # 估计该年的 Pareto 指数
      a_est <- exponent.multi_ratios_est(
        P99 = year_data$P99,
        P99.5 = year_data$P99.5,
        P99.9 = year_data$P99.9
      )
      years <- c(years, year)
      pareto_estimates <- c(pareto_estimates, a_est)
    }
  }
  return(data.frame(Year = years, Pareto_Exponent = pareto_estimates))
}

# 估计所有年份的 Pareto 指数
us_pareto_estimates <- estimate_pareto_by_year(us_data)

# 绘制时间序列图
ggplot(us_pareto_estimates, aes(x = Year, y = Pareto_Exponent)) +
  geom_line(color = "blue", linewidth = 1) +
  geom_point(color = "red", size = 2) +
  labs(
    title = "Pareto Exponent Estimates for United States (1913-2012)",
    # 美国 Pareto 指数估计 (1913-2012)
    x = "Year",  # 年份
    y = "Pareto Exponent (a)",  # Pareto 指数 (a)
    caption = "Estimated using multi-ratio method"  # 使用多比率方法估计
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)
  )
```



4. Use (4) to estimate a for the US for every year. Make a scatter-plot of these estimates against those from problem 3. If they are identical or completely independent, something is wrong with at least one part of your code. Otherwise, can you say anything about how the two estimates compare?

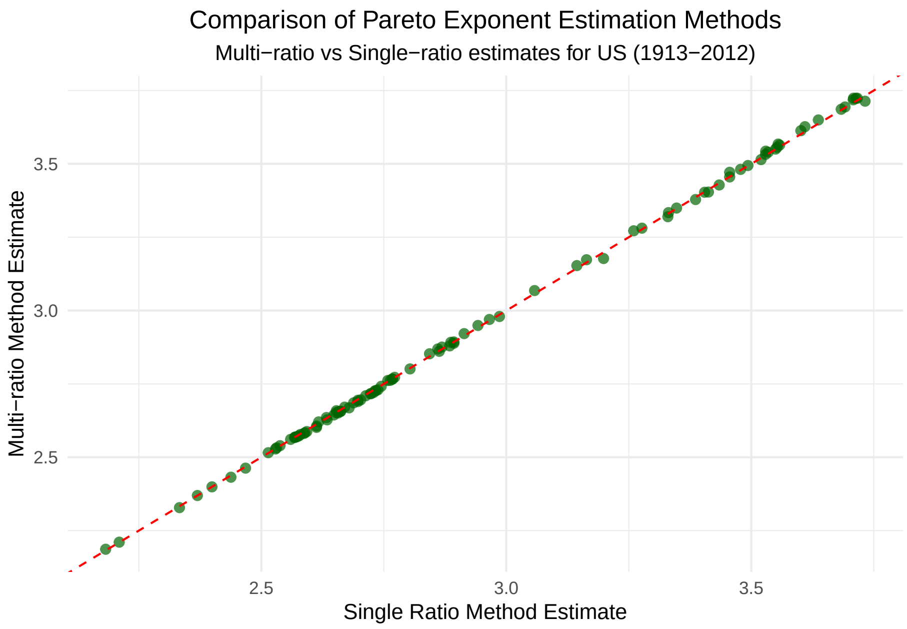
```
# 第4题：使用单一比率公式估计并比较两种方法

# 使用公式 a = 1 - log(10) / log(P99/P99.9) 估计帕累托指数
us_data$single_ratio_a <- 1 - log(10) / log(us_data$P99 / us_data$P99.9)

# 合并两种估计方法的结果
comparison_data <- data.frame(
  Year = us_data$Year,
  Multi_Ratio = us_pareto_estimates$Pareto_Exponent,  # 多比率方法
  Single_Ratio = us_data$single_ratio_a  # 单一比率方法
)
```

```r
# 创建散点图比较两种方法
plot2 <- ggplot(comparison_data, aes(x = Single_Ratio, y = Multi_Ratio)) +
  geom_point(color = "darkgreen", alpha = 0.7, size = 2) +  # 深绿色半透明点
  geom_abline(intercept = 0, slope = 1, color = "red",
              linetype = "dashed") +  # 45 度参考线
  labs(
    title = "Comparison of Pareto Exponent Estimation Methods",
    # 主标题：帕累托指数估计方法比较
    subtitle = "Multi-ratio vs Single-ratio estimates for US (1913-2012)",
    # 副标题：美国多比率与单一比率估计比较
    x = "Single Ratio Method Estimate",  # x 轴：单一比率方法估计
    y = "Multi-ratio Method Estimate"   # y 轴：多比率方法估计
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5)
  )

print(plot2)  # 显示比较图
```



Comparison of Pareto Exponent Estimation Methods

Multi−ratio vs Single−ratio estimates for US (1913−2012)

```
# 结果分析和统计总结

# 计算两种方法的相关系数
correlation <- cor(comparison_data$Single_Ratio,
                   comparison_data$Multi_Ratio, use = "complete.obs")
cat("\n两种估计方法的相关系数：", round(correlation, 4), "\n")
```

```
##
## 两种估计方法的相关系数： 0.9999
```

```
# 计算差异的统计量
differences <- comparison_data$Multi_Ratio - comparison_data$Single_Ratio
cat("估计差异的统计摘要：\n")
```

```
## 估计差异的统计摘要：
```

```
print(summary(differences))
```

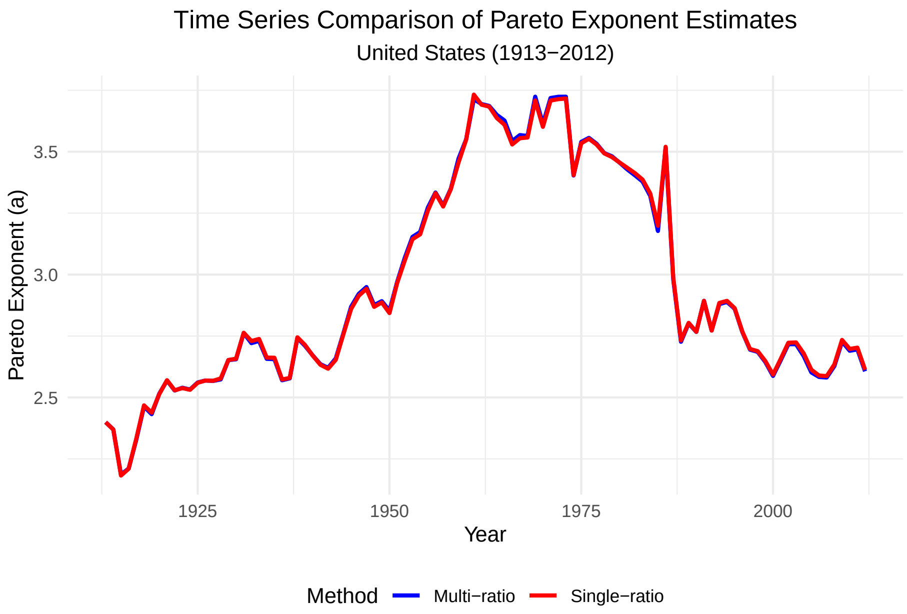
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0210042 -0.0052329 -0.0003906  0.0001077  0.0041082  0.0173404
```

```r
# 计算均方根误差
rmse <- sqrt(mean(differences^2, na.rm = TRUE))
cat(" 均方根误差 (RMSE)： ", round(rmse, 4), "\n")
```

```
## 均方根误差 (RMSE)： 0.007
```

```r
# 创建时间序列比较图
plot3 <- ggplot(comparison_data) +
  geom_line(aes(x = Year, y = Multi_Ratio, color = "Multi-ratio"),
            linewidth = 1) +  # 多比率方法线条
  geom_line(aes(x = Year, y = Single_Ratio, color = "Single-ratio"),
            linewidth = 1) +  # 单一比率方法线条
  scale_color_manual(
    name = "Method",  # 图例标题：方法
    values = c("Multi-ratio" = "blue", "Single-ratio" = "red")  # 颜色设置
  ) +
  labs(
    title = "Time Series Comparison of Pareto Exponent Estimates",
    # 主标题：帕累托指数估计的时间序列比较
    subtitle = "United States (1913-2012)",  # 副标题：美国
    x = "Year",  # x 轴：年份
    y = "Pareto Exponent (a)"  # y 轴：帕累托指数
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "bottom"  # 图例位置在底部
  )

print(plot3)  # 显示时间序列比较图
```



```
# 输出部分数据查看结果
cat("\n前 10 年的估计结果对比: \n")
```

```
##
## 前10年的估计结果对比:
```

```
print(head(comparison_data, 10))
```

```
##    Year Multi_Ratio Single_Ratio
## 1  1913    2.399102     2.399194
## 2  1914    2.369603     2.369454
## 3  1915    2.186477     2.182215
## 4  1916    2.210628     2.209948
## 5  1917    2.328325     2.332909
## 6  1918    2.462953     2.467761
## 7  1919    2.432006     2.437956
## 8  1920    2.515628     2.514182
## 9  1921    2.568826     2.570057
## 10 1922    2.528207     2.528930
```