Homework 1: R Part

STAT 343: Mathematical Statistics

SOLUTIONS

Details

How to Write Up

The written part of this assignment can be either typeset using latex or hand written.

Grading

5% of your grade on this assignment is for turning in something legible and organized.

An additional 15% of your grade is for completion. A quick pass will be made to ensure that you've made a reasonable attempt at all problems.

Across both the written part and the R part, in the range of 1 to 3 problems will be graded more carefully for correctness. In grading these problems, an emphasis will be placed on full explanations of your thought process. You don't need to write more than a few sentences for any given problem, but you should write complete sentences! Understanding and explaining the reasons behind what you are doing is at least as important as solving the problems correctly.

Solutions to all problems will be provided.

Collaboration

You are allowed to work with others on this assignment, but you must complete and submit your own write up. You should not copy large blocks of code or written text from another student.

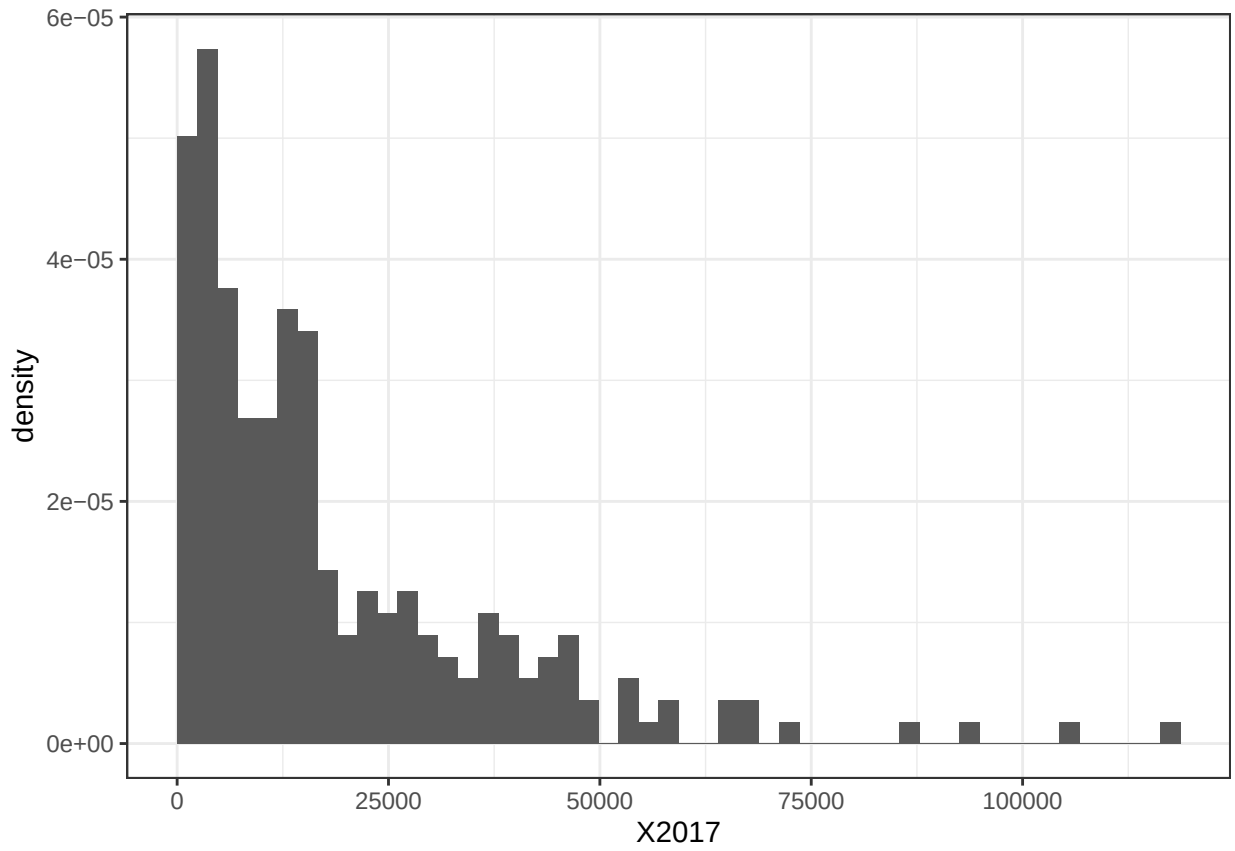
Sources

You may refer to our text, Wikipedia, and other online sources. All sources you refer to must be cited in the space I have provided at the end of this problem set.

Per Capita GDP

The code below reads in and plots a data set with measurements of per capita GDP at purchasing power parity as of 2017 for 235 countries, measured in inflation-adjusted 2011 international dollars; these data are from the World Bank, here: https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.KD. Per capita GDP can be roughly interpreted as the amount of income generated in a country in one year divided by the number of people living in that country. The purchasing power parity adjustment attempts to adjust GDP to account for differences in cost of living in different countries.

```
gdp <- read.csv("https://marievozanne.github.io/stat343/data/gdp.csv")
gdp <- gdp %>%
  filter(!is.na(X2017))
ggplot(data = gdp, mapping = aes(x = X2017)) +
  geom_histogram(mapping = aes(y = ..density..), boundary = 0, bins = 50) +
  theme_bw()
```



A lognormal distribution is often used to model non-negative variables that are skewed right, like incomes. In the written part of this assignment you will find the maximum likelihood estimator for the parameters of a lognormal distribution, and in the R part of the assignment you will fit the model to this data set.

For the purpose of this assignment, let's assume that the per capita GDP of different countries in a given year can be modeled as independent, identically distributed random variables (this is not actually reasonable, but may be good enough if our goal is to describe the distribution of values for per capita GDP across different countries).

Let's adopt the model $X_i \overset{\text{i.i.d.}}{\sim} \text{lognormal}(\mu, \sigma)$, $i = 1, \ldots, n$.

The pdf of a lognormal distribution is given by $f(x|\mu,\sigma) = x^{-1}(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\frac{\{\log(x)-\mu\}^2}{\sigma^2}\right]$

1. Using the maximum likelihood estimators that your found for μ and σ in the written part of this assignment, find the maximum likelihood estimates for the per capita GDP for 2017 (`X2017`).

```
gdp %>%
  summarize(
    mu_mle = sum(log(gdp$X2017))/nrow(gdp),
    sigma_mle = sum((log(gdp$X2017)-mu_mle)^2)/nrow(gdp)
  )
```

```
##     mu_mle sigma_mle
## 1 9.285576  1.269921
```

2. You can fit a log-normal model for the per capita GDP for 2017 using the `fitdistr` function from the `MASS` package. Fit this model and extract the maximum likelihood estimates from the model fit. (The model fit is a list, so you can index into the list using `$estimate`). Use this to verify the maximum likelihood estimates from 1.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.5
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
log_norm <- fitdistr(x=gdp$X2017, densfun = "log-normal")
log_norm$estimate
```

```
##  meanlog    sdlog
## 9.285576 1.126908
```