Chapter 7

Point Estimation

7.1 Introduction

In this chapter, we transition to one of the most important concepts in statistics, namely, how to **estimate** population-level parameters by using a sample of observations drawn from the population. Intuitively, if $X_1, X_2, ..., X_n$ is a sample from a population distribution described by $f_X(x)$, then the observations $X_1, X_2, ..., X_n$ contain valuable information about characteristics of the population distribution; e.g., the population mean $\mu = E(X)$, the population variance $\sigma^2 = \text{var}(X)$, and so on.

The reason the estimation question emerges as relevant is that parameters associated with a population distribution (or distributions) are usually **unknown**. For example, suppose an epidemiologist observes a random sample of $n = 10$ undergraduate students and records

$$Y = \text{ the number of sexual partners within the last six months}$$

on each student. As a population-level model, she decides to use $Y \sim \text{Poisson}(\lambda)$, where $\lambda = E(Y)$, the mean of the population. Now, suppose there are over 25,000 undergraduate students at the University in question. Therefore, the only way the epidemiologist can determine the value of λ is to observe all 25,000+ students. Because it is generally not possible to "sample the entire population" in real life evaluations (especially in larger populations which may number in the millions or billions), we turn to the problem of parameter **estimation**.

Suppose $X_1, X_2, ..., X_n$ is an *iid* sample from a population distribution described by $f_X(x)$, and let $\theta \in \mathbb{R}$ denote a population-level parameter that is unknown. We will approach "the point estimation problem" from the following point of view. We have a parametric model for $\mathbf{X} = (X_1, X_2, ..., X_n)$:

$$\mathbf{X} \sim f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta}), \text{ where } \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^k,$$

and the model parameter(s) $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_k)$ is unknown. In the last example, we could write $Y \sim \text{Poisson}(\theta)$, where $\theta = E(Y)$.

The population-level parameter $\boldsymbol{\theta}$ is unknown but **fixed**; i.e., it is <u>not random</u>. The use of a generic symbol like $\boldsymbol{\theta}$ for a population-level parameter allows us to generalize our discussion. We will assume that $\boldsymbol{\theta}$ is fixed, except when we discuss Bayesian estimation for which θ is considered random. The primary goal is to estimate a function of $\boldsymbol{\theta}$, say $\tau(\boldsymbol{\theta})$, where $\tau : \mathbb{R}^k \to \mathbb{R}^q$, $q \leq k$ (often, $q = 1$; i.e., $\tau(\boldsymbol{\theta})$ is a scalar parameter). Very often, $\tau(\boldsymbol{\theta}) = \theta$.

Remark: For most of the situations we will encounter in this course, the random vector $\mathbf{X}$ will consist of $X_1, X_2, ..., X_n$, an **iid sample** from the population $f_X(x|\boldsymbol{\theta})$. However, our discussion is also relevant when the independence assumption is relaxed, the identically distributed assumption is relaxed, or both.

Definition 7.1. A **point estimator** $W(\mathbf{X}) = W(X_1, X_2, ..., X_n)$ is any function of the sample $\mathbf{X}$. Therefore, any statistic is a point estimator. We call $W(\mathbf{x}) = W(x_1, x_2, ..., x_n)$ a **point estimate**. Because a point estimator $W(\mathbf{X})$ is a statistic, it is **random** and has its own distribution (termed a sampling distribution) – fixed $W(\mathbf{x})$ is a realization of random $W(\mathbf{X})$.

Illustration: What point estimator should we use in the undergraduate student example? Suppose $X_1, X_2, ..., X_{10}$ is regarded as an *iid* sample from a Poisson distribution with mean $\theta > 0$. One obvious point estimator of θ is the **sample mean**

$$\widehat{\theta} = \overline{X} = \frac{1}{10}\sum_{i=1}^{10} X_i \quad \longleftarrow \text{ function of } X_1, X_2, ..., X_{10}.$$

We know that $E(X) = \theta$, so at least on average (i.e., across many samples) the value of $\overline{X}$ will correctly pin down the true value of the population mean. Another candidate point estimator, interestingly, is the **sample variance**; i.e.,

$$\widehat{\theta} = S^2 = \frac{1}{10-1}\sum_{i=1}^{10}(X_i - \overline{X})^2 \quad \longleftarrow \text{ function of } X_1, X_2, ..., X_{10}.$$

Recall that in the Poisson distribution, the population mean and population variance are the same; i.e., $E(X) = \text{var}(X) = \theta$. Therefore, on average (i.e., across many samples), the value of S^2 will also correctly pin down the true value of θ.

These results are a product of the **Law of Large Numbers**.

Theorem 7.2. *(WLLN). Suppose that $X_1, X_2, ..., X_n$ is an* iid *sequence of random variables with $E(X_1) = \mu$ and $var(X_1) = \sigma^2 < \infty$. Let*

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

denote the sample mean. Then $\overline{X}_n \longrightarrow \mu$, as $n \to \infty$.

Visualization: The WLLN says that we can make the proportion of observed $\bar{x}_n$ curves inside $(1-\epsilon, 1+\epsilon)$ as close to $\lambda = 10$ as we like by making the plot sufficiently wide (with more observations). Note that in Figure 7.1.1, we see that as n increases the observed $\bar{x}_n$ approaches $\lambda = 10$.

```
> x.poisson1<-rpois(n=500,lambda=10)
> x.poisson2<-rpois(n=500,lambda=10)
> x.poisson3<-rpois(n=500,lambda=10)
> x.poisson4<-rpois(n=500,lambda=10)
> x.poisson5<-rpois(n=500,lambda=10)
> xbar.cumulative1<-cumsum(x.poisson1)/1:500
> xbar.cumulative2<-cumsum(x.poisson2)/1:500
```

```
> xbar.cumulative3<-cumsum(x.poisson3)/1:500
> xbar.cumulative4<-cumsum(x.poisson4)/1:500
>xbar.cumulative5<-cumsum(x.poisson5)/1:500
> library("TTR") # For runSD function
> ssq.cumulative1<-runSD(x=x.poisson1,n = 1,cumulative=TRUE)^2
> ssq.cumulative2<-runSD(x=x.poisson2,n = 1,cumulative=TRUE)^2
> ssq.cumulative3<-runSD(x=x.poisson3,n = 1,cumulative=TRUE)^2
> ssq.cumulative4<-runSD(x=x.poisson4,n = 1,cumulative=TRUE)^2
> ssq.cumulative5<-runSD(x=x.poisson5,n = 1,cumulative=TRUE)^2
> ggdat<-data.frame(x=1:500,
+                   y1=xbar.cumulative1,
+                   y2=xbar.cumulative2,
+                   y3=xbar.cumulative3,
+                   y4=xbar.cumulative4,
+                   y5=xbar.cumulative5)
> g1<-ggplot(data=ggdat, aes(x=x,y=y1))+
+     geom_line()+
+     geom_line(aes(y=y2),color=2)+
+     geom_line(aes(y=y3),color=3)+
+     geom_line(aes(y=y4),color=4)+
+     geom_line(aes(y=y5),color=5)+
+     geom_hline(yintercept = 10)+
+     theme_bw()+
+     xlab("Observation")+
+     ylab("Cumulative Sample Mean")+
+     ggtitle("Law of Large Numbers",subtitle=bquote(bar(X[n])~"approaches"~+         E(X)==lamb
> ggdat<-data.frame(x=1:500,
+                   y1=ssq.cumulative1,
+                   y2=ssq.cumulative2,
+                   y3=ssq.cumulative3,
+                   y4=ssq.cumulative4,
+                   y5=ssq.cumulative5)
> g2<-ggplot(data=ggdat, aes(x=x,y=y1))+
+     geom_line()+
+     geom_line(aes(y=y2),color=2)+
+     geom_line(aes(y=y3),color=3)+
+     geom_line(aes(y=y4),color=4)+
+     geom_line(aes(y=y5),color=5)+
+     geom_hline(yintercept = 10)+
+     theme_bw()+
+     xlab("Observation")+
+     ylab("Cumulative Sample Mean")+
+     ggtitle("Law of Large Numbers",subtitle=bquote(s[n]^2~"approaches"~+         var(X)==lambd
> grid.arrange(g1,g2,ncol=2)
```

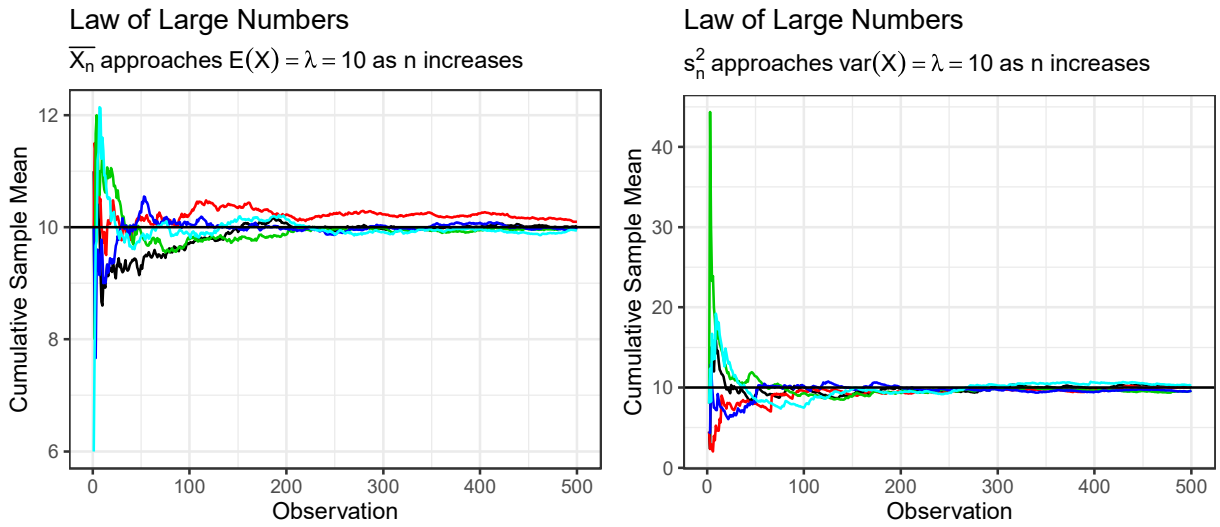

Figure 7.1.1: The cumulative mean of $n = 500$ observations from the Poisson($\lambda = 1$) distribution completed 5 times. As n increases, the sample means ($\bar{x}_n$) and the sample variances (s_n^2) approaches $E(X) = \lambda = 10$ and $var(X) = \lambda = 10$, respectively.

Remark: In the version of the WLLN stated in Theorem 7.2, we assumed finite variances; i.e., that $var(X_1) < \infty$. Can we weaken this assumption? It turns out that the WLLN still holds for *iid* sequences as long as $E(|X_1|) < \infty$, i.e., the first moment is finite (this is called **Khintchine's WLLN**).

Question: The WLLN guarantees that $\overline{X}_n \longrightarrow \mu$, as $n \to \infty$. Does a similar result hold for S^2, the sample variance? That is, does $S^2 \longrightarrow \sigma^2$, as $n \to \infty$?
Answer: Yes, in most cases – We see that in Figure 7.1.1 as n increases the observed $\bar{x}_n$ approaches $\lambda = 10$. This result requires finite fourth moments; i.e., $E(X_1^4) < \infty$. However, $S^2 \longrightarrow \sigma^2$ under the weaker assumption of $E(X_1^2) < \infty$.

Remark: When the limiting random variable is a constant, e.g., $\lambda = 10$, this type of convergence referred to as "consistency." We might say, "$\overline{X}_n$ is a consistent estimator of λ. In general, $\overline{X}_n$ is a consistent estimator of μ" and "S^2 is a consistent estimator of σ^2."

Question: Which point estimator should we use: $\overline{Y}$ or S^2? Maybe another point estimator is "better," say, the **sample median**

$$\widehat{\theta} = \frac{X_{(5)} + X_{(6)}}{2}.$$

The point is that in any estimation problem, there may be many point estimators to consider. Therefore, we should have a way to evaluate the quality of a point estimator so that we can judge whether or not it does a good job at estimation. This leads us to ask questions about how *accurate* and how *precise* a point estimator $\widehat{\theta}$ is.

A **point estimator** $\widehat{\theta} = W(X_1, X_2, ..., X_n)$ is random because its value depends on $X_1, X_2, ..., X_n$ which are random themselves. However, after the values $X_1 = x_1$, $X_2 = x_2$, ..., $X_n = x_n$ have been observed, we can calculate the value of the **point estimate**

$$\widehat{\theta} = W(x_1, x_2, ..., x_n).$$

This is a numerical value because it is based on the observed values $x_1, x_2, ..., x_n$; i.e., "the observed data." To illustrate, in the undergraduate student example, a point estimator of θ based on the

$n = 10$ students is

$$\overline{X} = \frac{1}{10}\sum_{i=1}^{10} X_i,$$

the sample mean. If the observed data are

$$x_1 = 4,\ x_2 = 2,\ x_3 = 1,\ x_4 = 3,\ x_5 = 2,\ x_6 = 5,\ x_7 = 0,\ x_8 = 1,\ x_9 = 0,\ x_{10} = 0,$$

then the point estimate is the realized value of the point estimator; i.e.,

$$\overline{x} = \frac{1}{10}\sum_{i=1}^{10} x_i = 1.8.$$

Note that a point estimate is no longer random (i.e., it's a fixed number).

7.1.1 Populations and Samples

Overview: We now focus on **statistical inference**. This deals with making (probabilistic) statements about a population of individuals based on information that is contained in a sample taken from the population.

Example 7.3. Suppose we are studying the performance of lithium batteries used in a certain calculator. We would like to learn about the lifetime of these batteries so we can place a limited warranty on them in the future. Because this type of battery has not been used in this calculator before, no one (except the Oracle) can tell us the distribution of Y , the battery's lifetime. In fact, not only is the distribution not known, but all parameters which index this distribution aren't known either.

Terminology: A population refers to the entire group of "individuals" (e.g., parts, people, batteries, etc.) about which we would like to make a statement (e.g., proportion defective, median IQ score, mean lifetime, etc.).

- It is generally accepted that the entire population can not be measured. It is too large and/or it would be too time consuming to do so.
- To draw inferences (make statements) about a population, we therefore observe a sample of individuals from the population.
- We will assume that the sample of individuals constitutes a random sample. Mathematically, this means that all observations are independent and follow the same probability distribution. Informally, this means that each sample (of the same size) has the same chance of being selected.
- Taking a random sample of individuals is our best hope of obtaining individuals that are "representative" of the entire population.

Notation: We will denote a random sample of observations by

$$Y_1, Y_2, ..., Y_n.$$

That is, Y_1 is the value of Y for the first individual in the sample, Y2 is the value of Y for the second individual in the sample, and so on. The sample size tells us how many individuals are in the sample and is denoted by n. We refer to the set of observations Y1, Y2, ..., Yn generically as **data**. Lower case notation $y_1, y_2, ..., y_n$ is used when citing numerical values.

4285	2066	2584	1009	318	1429	981	1402	1137	414
564	604	14	4152	737	852	1560	1786	520	396
1278	209	349	478	3032	1461	701	1406	261	83
205	602	3770	726	3894	2662	497	35	2778	1379
3920	1379	99	510	582	308	3367	99	373	454

Example 7.4. Consider the following random sample of$n = 50$ battery lifetimes $y_1, y_2, ..., y_{50}$ measured in hours: In Figure 7.1.2, we display a histogram of the battery lifetime data. We see that the (empirical) distribution of the battery lifetimes is skewed towards the high side.

- Which continuous probability distribution seems to display the same type of pattern that we see in the histogram?
- An exponential(λ) model seems reasonable here (based on the histogram shape). What is λ? The **exponential distribution** serves as a very good model for measurements like waiting times or lifetimes.

$$\begin{aligned}
\lambda &\in \mathbb{R}^+ && \textbf{[Parameters]} \\
\mathcal{X} &= \{\omega : \omega \in \mathbb{R}^+\} && \textbf{[Support]} \\
f_X(x|\mu,\sigma) &= \lambda e^{-\lambda x}\ I(x \in \mathbb{R}^+) && \textbf{[PDF]} \\
F_X(x|\mu,\sigma) &= 1 - e^{-\lambda x} && \textbf{[CDF]} \\
E(X) &= \frac{1}{\lambda} && \textbf{[Expected Value]} \\
var(X) &= \frac{1}{\lambda^2} && \textbf{[Variance]}
\end{aligned}$$

- In this example, λ is called a (population) parameter. It describes the distribution which is used to model the entire population of batteries.
- In general, (population) parameters which index probability distributions (like the exponential) are unknown.
- All the probability distributions we have discussed so far are meant to describe population-level behavior. With point estimation, we're trying to assess what might be going on in the population; e.g., which parameter values do data may suggest may be good estimates for the population distribution.

```
> dat<-c(4285,2066,2584,1009,318,1429,981,1402,1137,414,
+        564, 604, 14, 4152, 737, 852, 1560, 1786, 520, 396,
+        1278,209, 349, 478, 3032, 1461, 701, 1406, 261, 83,
+        205, 602, 3770, 726, 3894, 2662, 497, 35, 2778, 1379,
+        3920, 1379, 99, 510, 582, 308, 3367, 99, 373, 4540)
> ggdat<-data.frame(lifetime=dat)
> g1<-ggplot(data=ggdat,aes(x=lifetime))+
+   geom_histogram(aes(y=..density..),bins=10,color="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   geom_density(color="red")+
+   theme_bw()+
```

```
+   xlab("Battery Lifetime (in hours)")+
+   ylab("Density")
> ggdat.density<-data.frame(x=seq(0,5000,1),
+                           f1=dexp(x=seq(0,5000,1), rate = 1/1100),
+                           f2=dexp(x=seq(0,5000,1), rate = 1/1200),
+                           f3=dexp(x=seq(0,5000,1), rate = 1/1300),
+                           f4=dexp(x=seq(0,5000,1), rate = 1/1400),
+                           f5=dexp(x=seq(0,5000,1), rate = 1/1500))
> g2<-ggplot(data=ggdat,aes(x=lifetime))+
+   geom_histogram(aes(y=..density..),bins=10,color="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   geom_line(data=ggdat.density,aes(x=x,y=f1,color="lambda1"))+
+   geom_line(data=ggdat.density,aes(x=x,y=f2,color="lambda2"))+
+   geom_line(data=ggdat.density,aes(x=x,y=f3,color="lambda3"))+
+   geom_line(data=ggdat.density,aes(x=x,y=f4,color="lambda4"))+
+   geom_line(data=ggdat.density,aes(x=x,y=f5,color="lambda5"))+
+   theme_bw()+
+   xlab("Battery Lifetime (in hours)")+
+   ylab("Density")+
+   scale_color_discrete("",
+                labels=c(bquote(lambda==1000),bquote(lambda==2000),bquote(lambda==3000),
+                                    bquote(lambda==4000),bquote(lambda==5000)))
> grid.arrange(g1,g2,ncol=2)
```

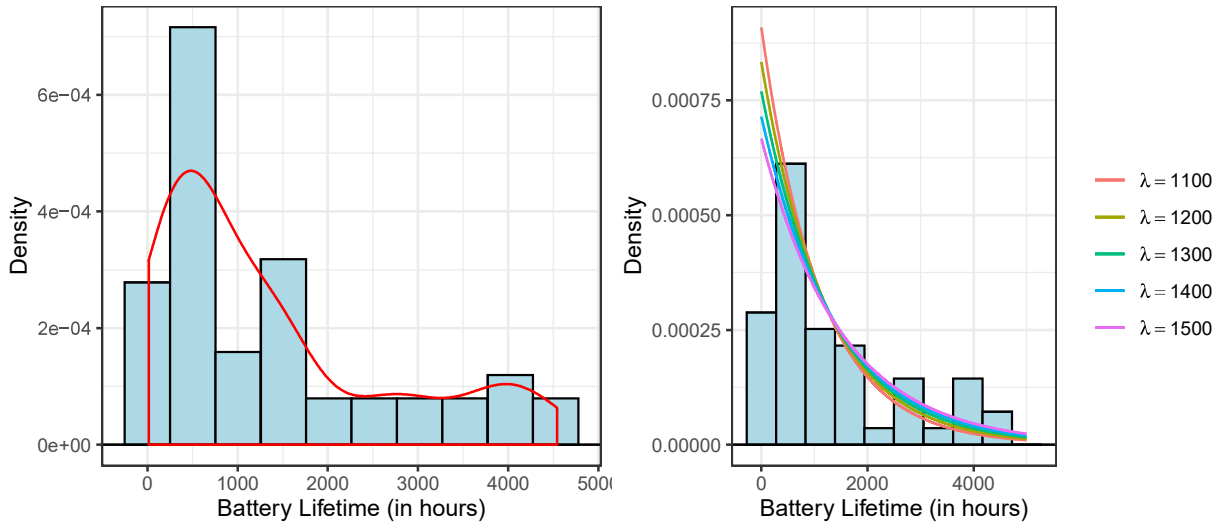

Figure 7.1.2: Histogram with a nonparametric density estimate (left) and a histogram with several exponential distributions superimposed (right) of the battery lifetime data (measured in hours).

7.2 Parameters and statistics

Terminology: A parameter is a numerical quantity that describes a population. In general, population parameters are unknown. Some very common examples are:

$$\mu = \text{population mean}$$
$$\sigma^2 = \text{population variance}$$
$$p = \text{population proportion.}$$

Connection: All of the probability distributions that we talked about are indexed by population (model) parameters.

For example,

- The Gaussian distribution is indexed by two parameters, the population mean μ and the population variance σ^2
- The Poisson distribution is indexed by one parameter, the population mean λ which is also the variance.
- The exponential distribution indexed by one parameter, the mean λ.
- The Bernoulli distribution is indexed by one parameter, the population proportion of successes p.

Terminology: Suppose that $Y_1, Y_2, ..., Y_n$ is a random sample from a population. The sample mean is

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

The sample variance is

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2.$$

The sample standard deviation is the positive square root of the sample variance; i.e.,

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}.$$

Important: Unlike their population analogues (which are unknown), these quantities can be computed from the sample $Y_1, Y_2, ..., Y_n$.

Terminology: A statistic is a numerical quantity that can be calculated from a sample of data. Some very common examples are:

$$\bar{Y} = \text{sample mean}$$
$$S^2 = \text{sample variance}$$
$$p = \text{sample proportion.}$$

For example, with the battery lifetime data (a random sample of $n = 50$ lifetimes),

$$\begin{aligned} \bar{Y} &= \text{sample mean} = 1355.86 \\ S^2 &= \text{sample variance} = 1702284 \\ p &= \text{sample proportion} = 1304.716, \end{aligned}$$

which have been calculated in `R` as follows.

```
> mean(dat.battery)
[1] 1355.86
> var(dat.battery)
[1] 1702284
> sd(dat.battery)
[1] 1304.716
```

Summary: The table below succinctly summarizes the differences between a population and a sample (a parameter and a statistic):

Group of individuals	Numerical quantity	Status
Population (not observed)	Parameter	Unknown
Sample (observed)	Statistic	Calculated from sample data

Statistical inference deals with making (probabilistic) statements about a population of individuals based on information that is contained in a sample taken from the population. We do this by

- estimating unknown population parameters with sample statistics
- quantifying the uncertainty (variability) that arises in the estimation process.

7.3 Methods of Finding Estimators

7.3.1 Method of Moments

Definition 7.5. Suppose $\mathbf{X} \sim f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_k) \in \Theta \subseteq \mathbb{R}^k$. The **method of moments (MOM)** approach says to equate the first k sample moments to the first k population moments and then to solve for $\boldsymbol{\theta}$. For our discussion, we'll focus on $k = 1$ and $k = 2$.

The idea here is to exploit the WLLN, i.e.

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \to E(X)$$

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \to var(X)$$

which denotes that, as $n \to \infty$, $\bar{X}_n$ becomes a "better" estimate of $E(X)$ and S_n^2 becomes a "better" estimate of $var(X)$. We will often used estimates to calculate $E(X)$ in the calculation of S_n^2.
Remark: For $k = 3$ we can use skewness and for $k = 4$ we can use kurtosis, but this discussion is omitted here.

The first k **sample moments** depend on the sample $\mathbf{X}$. The first k **population moments** will generally depend on $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_k)$. Therefore, the system of equations

$$\bar{X}_n \stackrel{\text{set}}{=} E(X)$$
$$S_n^2 \stackrel{\text{set}}{=} var(X)$$

can (at least in theory) be solved for $\theta_1, \theta_2, ..., \theta_k$. A solution to this system of equations is called a **method of moments (MOM) estimator**.

Example 7.6. Suppose that $X_1, X_2, ..., X_n$ are *iid* Uniform$(0, b)$, where $b > 0$. The first sample moment is

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i = \overline{X}.$$

The first population moment is

$$E(X) = \frac{b}{2}.$$

We set these moments equal to each other; i.e.,

$$\overline{X} \stackrel{\text{set}}{=} \frac{b}{2}$$

and solve for b. The solution

$$\widehat{b} = 2\overline{X}$$

is a method of moments estimator for b.

For example, consider $b = 3$. The result of the following simulation, displayed in Figure 7.3.3, shows that as n increases, $\widehat{b}_n = 2\overline{X}$ approaches b.

```
> n<-5000
> x.uniform0b<-runif(n=n,min=0,max=3)
> ggdat<-data.frame(x=x.uniform0b)
> g1<-ggplot(data=ggdat,aes(x=x))+
+   geom_histogram(aes(y=..density..),breaks=seq(0,3,0.5),color="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   geom_density(color="red")+
+   theme_bw()+
+   xlab("X")+
+   ylab("Density")+
+   ggtitle("Data from Uniform(0,b)",subtitle="Where b is unknown")
> ggdat<-data.frame(x=1:5000,
+                   y=2*cumsum(x.uniform0b)/1:5000) #cumulative 2*xbar
> g2<-ggplot(data=ggdat,aes(x=x,y=y))+
+   geom_line()+
+   geom_hline(yintercept=3,color="red",linetype="dashed")+
+   theme_bw()+
```

```
+   xlab("Observation")+
+   ylab(bquote(2*bar(X)[n]))
> grid.arrange(g1,g2,ncol=2)
```

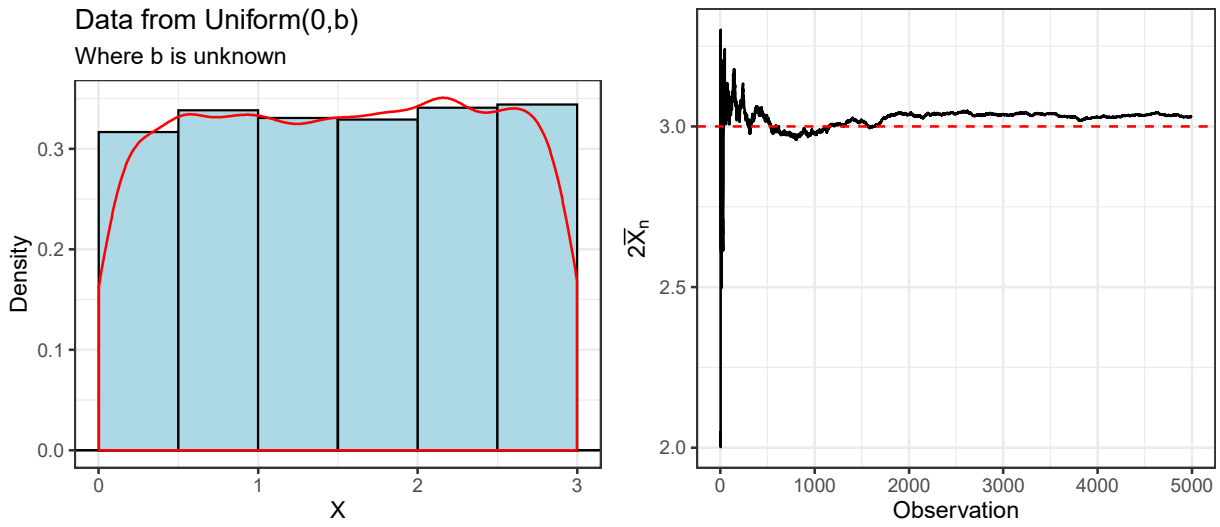

Figure 7.3.3: Histogram with a nonparametric density estimate (left) and the cumulative method of moments estimate for $n = 5000$ observations from the uniform$(0, b)$ distribution. As n increases the estimate $\widehat{b}_n = 2\overline{X}$ approaches $b = 3$.

The method of moments estimators can be found in R as follows.

```
> g<-function(theta,dat.x){ #function setting sample and population moments equal
+   #unknown parameter is b
+   b<-theta
+   EX<- (0+b)/2 #expected value of uniform distribution
+   EX - mean(dat.x)
+ }
> library(nleqslv)
> nleqslv(fn = g, #function(s) we want to solve for 0
+         x=c(max(x.uniform0b)),#reasonable starting guess for b
+         dat.x=x.uniform0b) #pass data in
$x
[1] 3.030194
$fvec
[1] 0
$termcd
[1] 1
$message
[1] "Function criterion near zero"
$scalex
[1] 1
$nfcnt
[1] 1
$njcnt
```

```
[1] 1
$iter
[1] 1
> 2*mean(x.uniform0b)
[1] 3.030194
```

Example 7.7. Suppose that $X_1, X_2, ..., X_n$ are iid Uniform$(-\theta, \theta)$, where $\theta > 0$. For this population, $E(X) = 0$ so this will not help us as there is no θ to solve for; i.e.,

$$\bar{X}_n = \frac{-\theta + \theta}{2} = 0$$

cannot be solved for θ.

Moving to second moments, we have

$$var(X) = \frac{(\theta - (-\theta))^2}{12} = \frac{\theta^2}{3}.$$

Therefore, we can set

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \stackrel{\text{set}}{=} \frac{\theta^2}{3}.$$

and solve for θ. The solution

$$\widehat{\theta} = \pm\sqrt{\frac{3}{n}\sum_{i=1}^{n} X_i^2}$$

is a method of moments estimator for θ. We keep the positive solution because $\theta > 0$ (although, technically, the negative solution is still a MOM estimator).

For example, consider $\theta = 3$. The result of the following simulation, displayed in Figure 7.3.4, shows that as n increases, $\widehat{\theta}_n = \pm\sqrt{\frac{3}{n}\sum_{i=1}^{n} X_i^2}$ approaches $\pm\theta$.

```
> n<-5000
> x.uniformtt<-runif(n=n,min=-3,max=3)
> max(x)
[1] 2.999112
> ggdat<-data.frame(x=x.uniformtt)
> g1<-ggplot(data=ggdat,aes(x=x))+
+   geom_histogram(aes(y=..density..),breaks=seq(-3,3,0.5),color="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   geom_density(color="red")+
+   theme_bw()+
+   xlab("X")+
+   ylab("Density")+
+   ggtitle(bquote("Data from Uniform("-theta*","*theta*")"),
+       subtitle=bquote("Where"~theta~"is unknown"))
> ggdat<-data.frame(x=1:5000,
+                   y=sqrt(3*cumsum(x.uniformtt^2)/1:5000)) #cumulative 2*xbar
> g2<-ggplot(data=ggdat,aes(x=x,y=y))+
+   geom_line()+
+   geom_hline(yintercept=3,color="red",linetype="dashed")+
+   theme_bw()+
```

```
+   xlab("Observation")+
+   ylab(bquote(sqrt((3/n)* sum(x[i]^2,i==1,n))))
> grid.arrange(g1,g2,ncol=2)
```

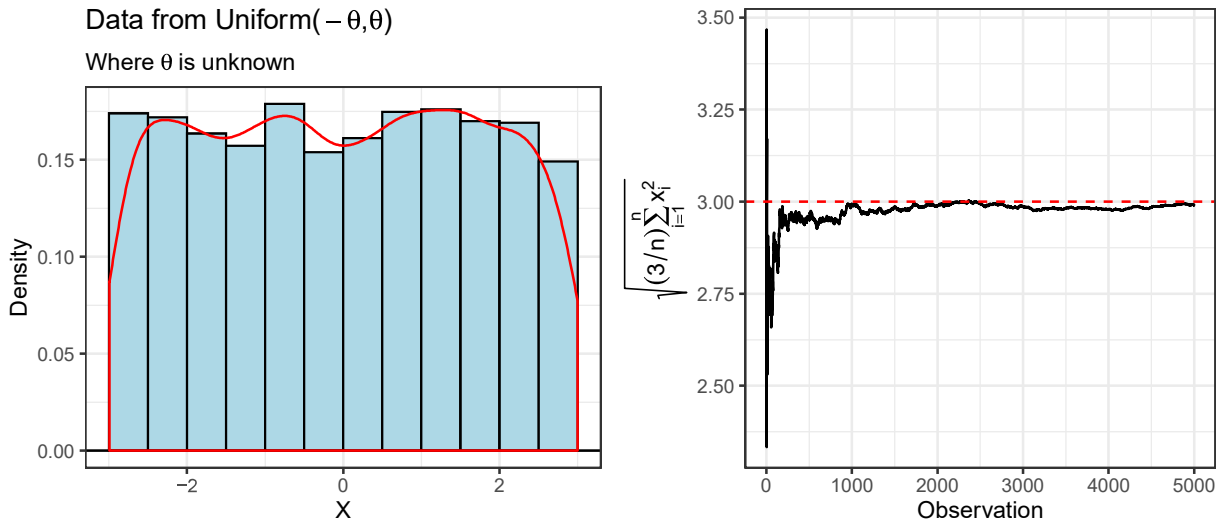

Figure 7.3.4: Histogram with a nonparametric density estimate (left) and the cumulative method of moments estimate for $n = 5000$ observations from the uniform$(-\theta, \theta)$ distribution. As n increases the estimate $\widehat{\theta}_n = \pm\sqrt{\frac{3}{n}\sum_{i=1}^{n} X_i^2}$ approaches $\theta = \pm 3$..

Remark: Sometimes, even if $dim(\theta) = 1$, we have to move to later moments yield an equation for which we can solve for θ. In the most extreme case, we may not be able to find method of moments estimators; e.g., the moments of the Cauchy distribution do not exist because for $X \sim$ Cauchy(μ, σ),

$$E(X^n) = \int_{-\infty}^{\infty} x^n \left[\frac{1}{\pi\sigma\left(1 + \left(\frac{x-\mu}{\sigma}\right)^2\right)} \right]$$

diverges for $n \geq 1$

The method of moments estimators can be found in R as follows.

```
> g<-function(theta,dat.x){ #(theta,x) order is important
+   #unknown parameter is theta
+   #EX<- (a+b)/2 = 0 here
+   VARX<- (theta-(-1*theta))^2/12 #variance of a uniform distribution
+   VARX - var(dat.x)
+ }
> nleqslv(fn = g, #function(s) we want to solve for 0
+         x=c(max(x.uniformtt)),#reasonable starting guess for b
+         dat.x=x.uniformtt) #pass data in
$x
[1] 2.991059
$fvec
[1] 2.490008e-10
```

```
$termcd
[1] 1
$message
[1] "Function criterion near zero"
$scalex
[1] 1
$nfcnt
[1] 4
$njcnt
[1] 1
$iter
[1] 2
> sqrt((3/5000)*sum(x.uniformtt^2))
[1] 2.991059
```

Example 7.8. Suppose that $X_1, X_2, ..., X_n$ are iid Uniform(a, b), where $a < b \in \mathbb{R}$. For this population, we must solve the following system of equations.

$$E(X) = \frac{a+b}{2} \stackrel{\text{set}}{=} \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$
$$var(X) = \frac{(b-a)^2}{12} \stackrel{\text{set}}{=} \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

The first equation gives us

$$a = 2\bar{x} - b$$

and substituting this into the second yields

$$\frac{(b-a)^2}{12} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$
$$\frac{(b-(2\bar{x}-b))^2}{12} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad \textbf{[Plugging in]}$$
$$\frac{4b^2 - 8b\bar{x} + 4\bar{x}^2}{12} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad \textbf{[Expanding]}$$
$$\frac{1}{3}\left(b^2 - 2b\bar{x} + \bar{x}^2\right) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad \textbf{[Simplifying]}$$
$$\frac{1}{3}b^2 + \frac{2\bar{x}}{3}b + \left(\frac{\bar{x}^2}{3} - S^2\right) = 0 \qquad \textbf{[Rewriting]}$$

A solution, found using the quadratic formula, is

$$\widehat{b} = \overline{x} + \sqrt{3S^2}$$
$$\widehat{a} = 2\bar{x} - \widehat{b} = \overline{x} - \sqrt{3S^2}$$

is a method of moments estimator for $\boldsymbol{\theta} = (a, b)$. We keep the positive solution for $\hat{b}$ because b is the maximum and *should* be larger than a (although, technically, the negative solution is still a MOM estimator).

Remark: Method of moments estimators may not be unique. Here, if we were to solve for b first, we would yield a different set of estimators.

For example, consider $a = -1$, and $b = 5$. The result of the following simulation, displayed in Figure 7.3.5, shows that as n increases, $\widehat{a}_n = \overline{x} - \sqrt{3S^2}$ approaches a, and $\widehat{b}_n = \overline{x} + \sqrt{3S^2}$ approaches b.

```
> n<-5000
> x.uniformab<-runif(n=n,min=-1,max=5)
> ggdat<-data.frame(x=x.uniformab)
> g1<-ggplot(data=ggdat,aes(x=x))+
+   geom_histogram(aes(y=..density..),breaks=seq(-1,5,0.5),color="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   geom_density(color="red")+
+   theme_bw()+
+   xlab("X")+
+   ylab("Density")+
+   ggtitle(bquote("Data from Uniform(a,b)"),subtitle=bquote("Where"~a~"and"~b~"are unknown"))
> ggdat<-data.frame(x=1:5000,
+           a=(cumsum(x.uniformab)/1:5000) - sqrt(3*((cumsum(x.uniformab^2)/1:5000)-
+                (cumsum(x.uniformab)/1:5000)^2)),
+           b=(cumsum(x.uniformab)/1:5000) + sqrt(3*((cumsum(x.uniformab^2)/1:5000)-
+                (cumsum(x.uniformab)/1:5000)^2)))
> g2<-ggplot(data=ggdat,aes(x=x,y=a))+
+   geom_line(aes(color="a"))+
+   geom_hline(yintercept=-1,color="black",linetype="dashed")+
+   geom_line(aes(y=b,color="b"))+
+   geom_hline(yintercept=3,color="black",linetype="dashed")+
+   theme_bw()+
+   xlab("Observation")+
+   ylab("Cumulative Estimates")+
+   scale_color_manual("",breaks=c("a","b"),values=c("blue","red"))
> grid.arrange(g1,g2,ncol=2)
```

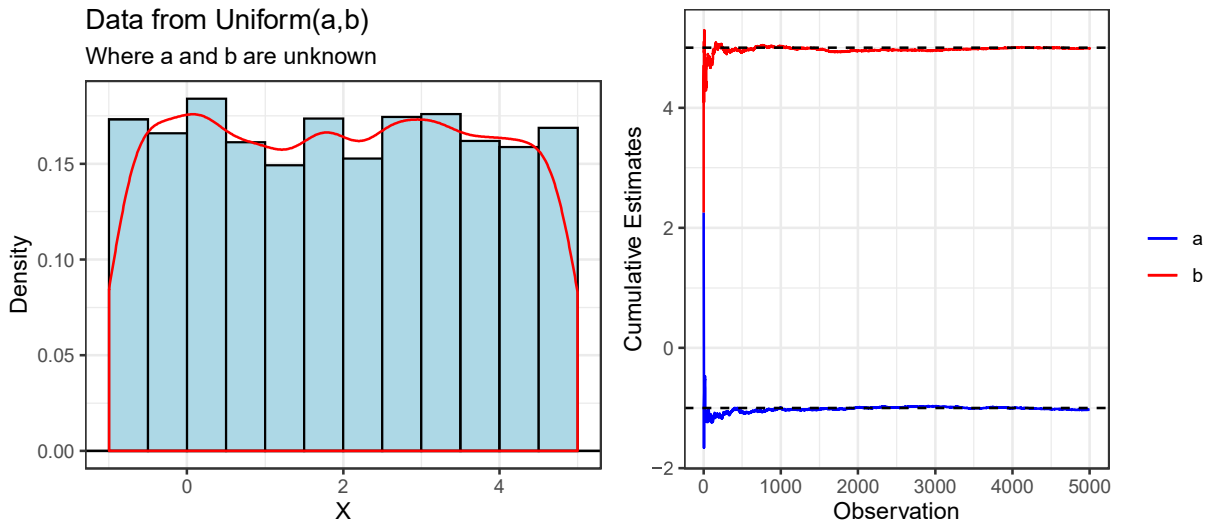

Figure 7.3.5: The cumulative mean of $n = 5000$ observations from the uniform(-1,5) distribution. As n increases $\widehat{a}_n = \overline{x} - \sqrt{3S^2}$ approaches $a = -1$, and $\widehat{b}_n = \overline{x} + \sqrt{3S^2}$ approaches $b = 5$.

The method of moments estimators can be found in R as follows.

```
> g<-function(theta,dat.x){
+   a<-theta[1]
+   b<-theta[2]
+   EX<- (a+b)/2 #expected value of a uniform distribution
+   VARX<- (b-a)^2/12 #variance of a uniform distribution
+   eq1<- EX - mean(dat.x)
+   eq2<- VARX - var(dat.x)
+   c(eq1,eq2)
+ }
+ nleqslv(fn = g, #function(s) we want to solve =c(0,0)
+         x=c(min(x.uniformab),max(x.uniformab)),#starting guess
+         dat.x=x.uniformab) #pass data in
$x
[1] -1.027230  4.992357
$fvec
[1] 0.000000e+00 -3.046452e-13
$termcd
[1] 1
$message
[1] "Function criterion near zero"
$scalex
[1] 1 1
$nfcnt
[1] 3
$njcnt
[1] 1
$iter
[1] 3
> mean(x.uniformab) - sqrt(3*var(x.uniformab))  #a
```

```
[1] -1.02723
> mean(x.uniformab) + sqrt(3*var(x.uniformab))  #b
[1] 4.992357
```

Example 7.9. Recall Example 7.4 which provided data on the lifetime of batteries in hours. Suppose that $X_1, X_2, ..., X_{50}$ are iid Exponential(λ), where $\lambda \in \mathbb{R}^+$. For this population, we must solve the following equation.

$$E(X) = \frac{1}{\lambda} \stackrel{\text{set}}{=} \bar{X}$$

This yields a method of moments estimator of

$$\widehat{\lambda} = \frac{1}{\overline{x}},$$

for $\theta = \lambda$.

```
> ggdat<-data.frame(lifetime=dat.battery)
> ggdat.density<-data.frame(x=seq(0,5000,1),
+                           f1=dexp(x=seq(0,5000,1), rate = 1/mean(dat.battery)))
> g1<-ggplot(data=ggdat,aes(x=lifetime))+
+   geom_histogram(aes(y=..density..),bins=10,color="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   geom_line(data=ggdat.density,aes(x=x,y=f1),color="red")+
+   theme_bw()+
+   xlab("Battery Lifetime (in hours)")+
+   ylab("Density")+
+   ggtitle(bquote("Data from Exponential"(lambda)),+        subtitle = bquote("Where"~lambda~"i
> ggdat<-data.frame(x=1:length(dat.battery),
+                   lambda.xbar=1/(cumsum(dat.battery)/1:length(dat.battery)))
> g2<-ggplot(data=ggdat,aes(x=x,y=lambda.xbar))+
+   geom_line()+
+   theme_bw()+
+   xlab("Observation")+
+   ylab(bquote(1/bar(x)[n]))
> grid.arrange(g1,g2,ncol=2)
```

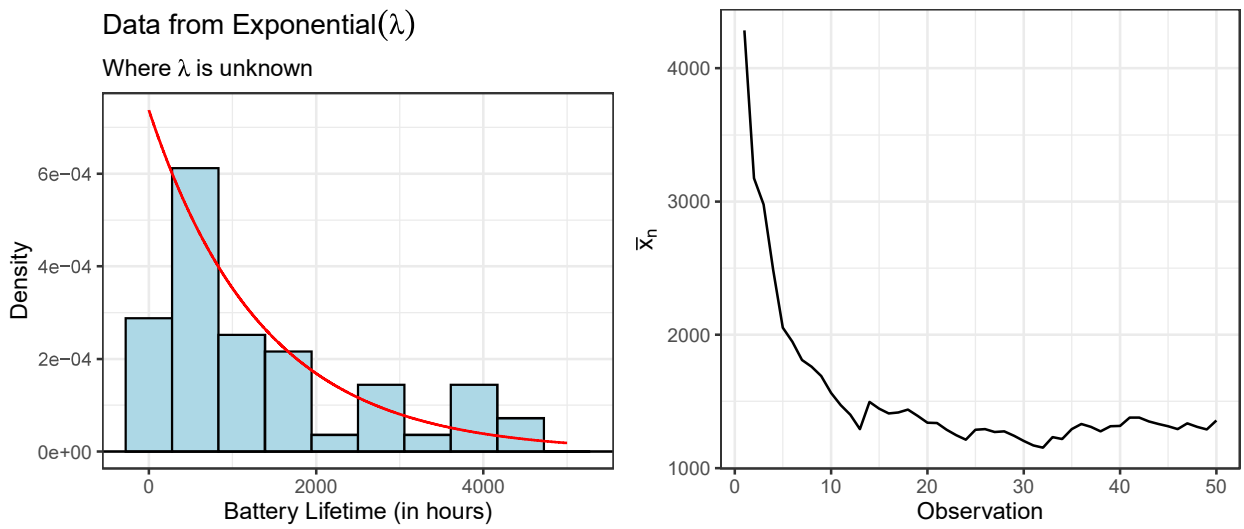

Figure 7.3.6: The cumulative mean of the batter data. As n increases $\widehat{\lambda}_n = \overline{x}$ approaches unknown λ.

The method of moments estimators can be found in R as follows.

```
> g<-function(theta,dat.x){
+   lambda<-theta
+   EX<- 1/lambda #expected value of an exponential distribution
+   EX-mean(dat.x)
+ }
> nleqslv(fn = g, #function(s) we want to solve =c(0,0)
+         x=1/mean(dat.battery),#starting guess
+         dat.x=dat.battery) #pass data in
$x
[1] 0.0007375393
$fvec
[1] 0
$termcd
[1] 1
$message
[1] "Function criterion near zero"
$scalex
[1] 1
$nfcnt
[1] 0
$njcnt
[1] 0
$iter
[1] 0
> 1/mean(dat.battery)
[1] 0.0007375393
```

We think of MOM estimation as a "quick and dirty" approach. All we are doing is matching moments. We are attempting to learn about a population $f_X(x|\theta)$ by using moments only. Some-

times MOM estimators have good finite-sample properties (e.g., unbiasedness, small variance, etc.). Sometimes they do not. MOM estimators generally do have desirable large-sample properties (e.g., asymptotically (large-sample) Gaussian, etc.) but are usually less (asymptotically) efficient than other estimators, e.g. they tend to require a larger sample size than other estimators. MOM estimators can be nonsensical. In fact, sometimes MOM estimators fall outside the parameter space Θ. For example, in linear models with random effects, variance components estimated via MOM can be negative.

7.3.2 Maximum likelihood estimation

Definition 7.10. Suppose $\mathbf{X} \sim f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^k$. Given that $\mathbf{X} = \mathbf{x}$ is observed, the function

$$L(\boldsymbol{\theta}|\mathbf{x}) = f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})$$

is called the **likelihood function**.

Remark: The likelihood function $L(\boldsymbol{\theta}|\mathbf{x})$ is the same function as the joint PDF/PMF $f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})$. The difference is in how we interpret each one. The function $f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})$ is a model that describes the random behavior of $\mathbf{X}$ when $\boldsymbol{\theta}$ is fixed. The function $L(\boldsymbol{\theta}|\mathbf{x})$ is viewed as a function of $\boldsymbol{\theta}$ with the data $\mathbf{X} = \mathbf{x}$ held fixed.

When $\mathbf{X}$ is discrete,

$$L(\boldsymbol{\theta}|\mathbf{x}) = f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta}) = P_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}).$$

That is, when $\mathbf{X}$ is discrete, we can interpret the likelihood function $L(\boldsymbol{\theta}|\mathbf{x})$ literally as a joint probability. Suppose that $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are two possible values of $\boldsymbol{\theta}$. Suppose $\mathbf{X}$ is discrete and

$$L(\boldsymbol{\theta}_1|\mathbf{x}) = P_{\boldsymbol{\theta}_1}(\mathbf{X} = \mathbf{x}) > P_{\boldsymbol{\theta}_2}(\mathbf{X} = \mathbf{x}) = L(\boldsymbol{\theta}_2|\mathbf{x}).$$

This suggests the sample $\mathbf{x}$ is more likely to have occurred with $\boldsymbol{\theta} = \boldsymbol{\theta}_1$ rather than if $\boldsymbol{\theta} = \boldsymbol{\theta}_2$. Therefore, in the discrete case, we can interpret $L(\boldsymbol{\theta}|\mathbf{x})$ as "the probability of the data $\mathbf{x}$."

Of course, this interpretation of $L(\boldsymbol{\theta}|\mathbf{x})$ is not appropriate when $\mathbf{X}$ is continuous because $P_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}) = 0$. However, this description is still used informally when describing the likelihood function with continuous data.

Definition 7.11. Any maximizer $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}(\mathbf{x})$ of the likelihood function $L(\boldsymbol{\theta}|\mathbf{x})$ is called a **maximum likelihood estimate**. With our previous interpretation, we can think of $\widehat{\boldsymbol{\theta}}$ as "the value of $\boldsymbol{\theta}$ that maximizes the probability of observing the data $\mathbf{x}$." We call $\widehat{\boldsymbol{\theta}}(\mathbf{X})$ a **maximum likelihood estimator** (MLE).

Finding the MLE $\widehat{\boldsymbol{\theta}}$ is essentially a maximization problem. The estimate $\widehat{\boldsymbol{\theta}}(\mathbf{x})$ must fall in the parameter space Θ because we are maximizing $L(\boldsymbol{\theta}|\mathbf{x})$ over Θ; i.e.,

$$\widehat{\boldsymbol{\theta}}(\mathbf{x}) = \arg\max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}|\mathbf{x}).$$

There is no guarantee that an MLE $\widehat{\boldsymbol{\theta}}(\mathbf{x})$ will be unique, although it often is.

In general, when the likelihood function $L(\boldsymbol{\theta}|\mathbf{x})$ is a differentiable function of $\boldsymbol{\theta}$, we can use calculus to maximize $L(\boldsymbol{\theta}|\mathbf{x})$. If an MLE $\widehat{\boldsymbol{\theta}}$ exists, it must satisfy

$$\frac{\partial}{\partial \theta_j} L(\widehat{\boldsymbol{\theta}}|\mathbf{x}) = 0, \quad j = 1, 2, ..., k.$$

Of course, second-order conditions must be verified to ensure that $\widehat{\boldsymbol{\theta}}$ is a maximizer and not a minimizer or some other value. In most "real" applications, however, the likelihood function $L(\boldsymbol{\theta}|\mathbf{x})$ must be maximized numerically to calculate $\widehat{\boldsymbol{\theta}}(\mathbf{x})$.

Definition 7.12. Suppose that $L(\boldsymbol{\theta}|\mathbf{x})$ is a likelihood function. Then

$$\begin{aligned}\widehat{\boldsymbol{\theta}}(\mathbf{x}) &= \arg\max_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta}|\mathbf{x}) \\ &= \arg\max_{\boldsymbol{\theta}\in\Theta} \ln\left[L(\boldsymbol{\theta}|\mathbf{x})\right].\end{aligned}$$

The function $\ln\left[L(\boldsymbol{\theta}|\mathbf{x})\right]$ is called the **log-likelihood function**. Analytically, it is usually easier to work with $\ln\left[L(\boldsymbol{\theta}|\mathbf{x})\right]$ than with the likelihood function directly. The equations

$$\frac{\partial}{\partial\theta_j}\ln\left[L(\boldsymbol{\theta}|\mathbf{x})\right] = 0, \quad j = 1, 2, ..., k,$$

are called the **score equations**.

Remark: Often, we will use the **log-likelihood function** because the likelihood function itself can be too small to represent numerically; i.e., it can be reported as zero even if it is infinitesimal and positive. Maximizing the log-likelihood is equivalent to maximizing the likelihood function.

Example 7.13. Recall Example 7.6. Suppose $X_1, X_2, ..., X_n$ are *iid* Uniform$[0, b]$, where $b > 0$. To find the MLE of b, we first find the likelihood function.

$$\begin{aligned}L(b|\mathbf{x}) &= \prod_{i=1}^{n}\frac{1}{b}\, I(0 \le x_i \le b) \\ &= \frac{1}{b^n}\, I(x_{(n)} \le b)\, I(x_{(1)} \ge 0)\end{aligned}$$

Note that for $b \ge x_{(n)}$, $L(b|\mathbf{x}) = 1/b^n$ decreases as b increases because

$$\frac{\partial L}{\partial b} = -n\frac{1}{b^{n+1}} < 0.$$

For $b < x_{(n)}$, $L(b|\mathbf{x}) = 0$. Thus, the MLE of b is $\widehat{b} = X_{(n)}$, the smallest estimate under the constraint that $b \ge x_{(n)}$. This maximization can be done in R as follows.

```
> uniform.LL<-function(x,theta,neg=FALSE){
+   b<-theta
+   LL<- sum(dunif(x=x,min=0,max=b,log=TRUE))
+   ifelse(!neg,LL,-LL)
+ }
> (maximization<-optim(fn=uniform.LL, #function to optimize
+       x=x.uniform0b,
+       par = max(x.uniform0b), #reasonable guess for b
+       method = "Brent", #univariate optimizer
+       lower = 0, #lower bound for b
+       upper = 10, #reasonable upper bound for b
+       neg=TRUE)) #optim minimizes so we want the negative log-likelihood
$par
[1] 2.99932
$value
```

```
[1] 5491.677
$counts
function gradient
NA       NA
$convergence
[1] 0
$message
NULL
```

Illustration: The likelihood function is displayed in Figure 7.3.7, and is created with the following R code. Here, we can visually confirm that $\widehat{b} = x_{(n)}$ maximizes the likelihood function for given a sample.

```
> ggdat<-data.frame(theta=seq(0,10,0.01),
+                   LL= sapply(X=seq(0,10,0.01),FUN=uniform.LL,x=x.uniform0b,neg=FALSE),
+                   nLL= sapply(X=seq(0,10,0.01),FUN=uniform.LL,x=x.uniform0b,neg=TRUE))
> ggdat.highlight<-data.frame(x=maximization$par,
+                             y=uniform.LL(x=x.uniform0b,theta=maximization$par,neg=FALSE))
> g1<-ggplot(data=ggdat,aes(x=theta,y=LL))+
+   geom_line()+
+   geom_point(data=ggdat.highlight,aes(x=x,y=y),color="red")+
+   theme_bw()+
+   xlab(bquote(b))+
+   ylab(bquote(log(L(blx))))+
+   ggtitle("Log Likelihood Function",subtitle="Parameter b given fixed sample")
> ggdat.highlight<-data.frame(x=maximization$par,
+                             y=uniform.LL(x=x.uniform0b,theta=maximization$par,neg=TRUE))
> g2<-ggplot(data=ggdat,aes(x=theta,y=nLL))+
+   geom_line()+
+   geom_point(data=ggdat.highlight,aes(x=x,y=y),color="red")+
+   theme_bw()+
+   xlab(bquote(b))+
+   ylab(bquote(log(L(blx))))+
+   ggtitle("Negative Log Likelihood Function",subtitle="Parameter b given fixed sample")
> grid.arrange(g1,g2,ncol=2)
```

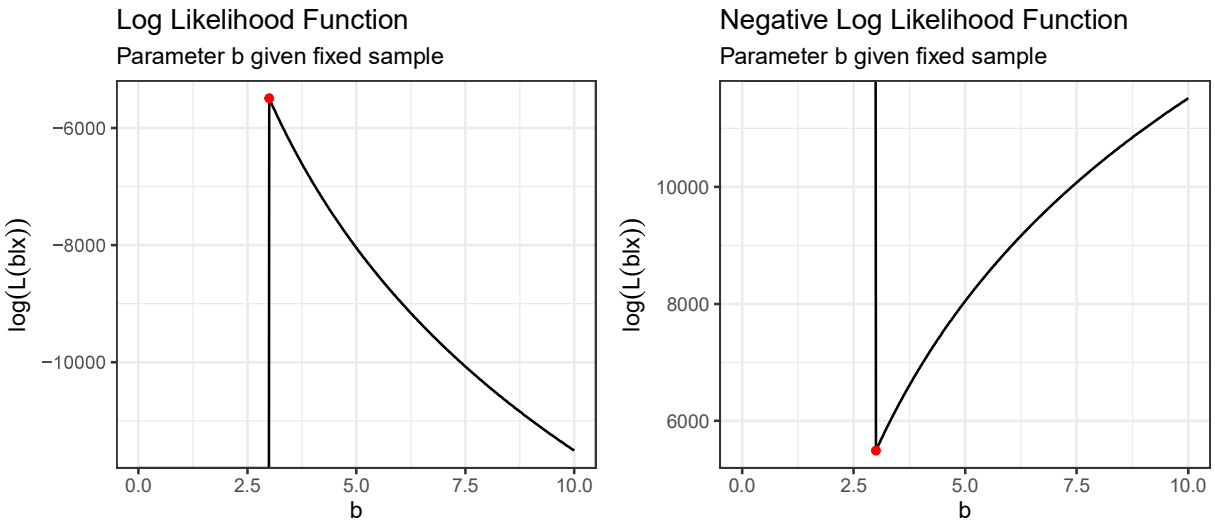

Figure 7.3.7: Plot of $log(L(b|\mathbf{x}))$ versus b (left) and $-log(L(b|\mathbf{x}))$ (right) with the maximum likelihood estimate highlighted using a red point.

Example 7.14. Recall Example 7.7. Suppose $X_1, X_2, ..., X_n$ are *iid* Uniform$[-\theta, \theta]$, where $\theta > 0$. To find the MLE of θ, we first find the likelihood function.

$$\begin{aligned} L(\theta|\mathbf{x}) &= \prod_{i=1}^{n} \frac{1}{\theta - (-\theta)} \; I(-\theta \leq x_i \leq \theta) \\ &= \frac{1}{(2\theta)^n} \; I(x_{(n)} \leq \theta) \; I(x_{(1)} \geq -\theta) \\ &= \frac{1}{(2\theta)^n} \; I(x_{(n)} \leq \theta) \; I(-x_{(1)} \leq \theta) \end{aligned}$$

Note that $L(\theta|\mathbf{x}) = 1/(2\theta)^n$ decreases as θ increases because

$$\frac{\partial L}{\partial \theta} = -n\frac{1}{2^n\theta^{n+1}} < 0.$$

For $\theta < \max(-x_{(1)}, x_{(n)})$, $L(\theta|\mathbf{x}) = 0$. Thus, the MLE of θ is $\widehat{\theta} = \max(-X_{(1)}, X_{(n)})$, the smallest estimate under the constraints that $x_{(1)} \geq -\theta$ and $x_{(n)} \leq \theta$. This maximization can be done in R as follows.

```
> uniform.LL<-function(x,theta,neg=FALSE){
+   LL<- sum(dunif(x=x,min=-theta,max=theta,log=TRUE))
+   ifelse(!neg,LL,-LL)
+ }
> (maximization<-optim(fn=uniform.LL, #function to optimize
+       x=x.uniformtt, #data
+       par = max(abs(min(x.uniformtt)),max(x.uniformtt)), #reasonable guess for theta
+       method="Brent", #univariate optimizer
+       lower=0, #lower bound for theta
+       upper=10, #reasonable upper bound for theta
+       neg=TRUE)) #optim minimizes so we want the negative log-likelihood
$par
[1] 2.999528
```

```
$value
[1] 8958.011
$counts
function gradient
NA       NA
$convergence
[1] 0
$message
NULL
```

Illustration: The likelihood function is displayed in Figure 7.3.8, and is created with the following R code. Here, we can visually confirm that $\widehat{theta} = x_{(n)}$ maximizes the likelihood function for given a sample.

```
> ggdat<-data.frame(theta=seq(0,10,0.01),
+                   LL= sapply(X=seq(0,10,0.01),FUN=uniform.LL,x=x.uniformtt,neg=FALSE),
+                   nLL= sapply(X=seq(0,10,0.01),FUN=uniform.LL,x=x.uniformtt,neg=TRUE))
> ggdat.highlight<-data.frame(x=maximization$par,
+                             y=uniform.LL(x=x.uniformtt,theta=maximization$par,neg=FALSE))
> g1<-ggplot(data=ggdat,aes(x=theta,y=LL))+
+   geom_line()+
+   geom_point(data=ggdat.highlight,aes(x=x,y=y),color="red")+
+   theme_bw()+
+   xlab(bquote(theta))+
+   ylab(bquote(log(L(theta*lx))))+
+   ggtitle("Log Likelihood Function",
+           subtitle=bquote("Parameter"~theta~"given fixed sample"))
> ggdat.highlight<-data.frame(x=maximization$par,
+                             y=uniform.LL(x=x.uniformtt,theta=maximization$par,neg=TRUE))
> g2<-ggplot(data=ggdat,aes(x=theta,y=nLL))+
+   geom_line()+
+   geom_point(data=ggdat.highlight,aes(x=x,y=y),color="red")+
+   theme_bw()+
+   xlab(bquote(theta))+
+   ylab(bquote(log(L(theta*lx))))+
+   ggtitle("Negative Log Likelihood Function",
+           subtitle=bquote("Parameter"~theta~"given fixed sample"))
> grid.arrange(g1,g2,ncol=2)
```

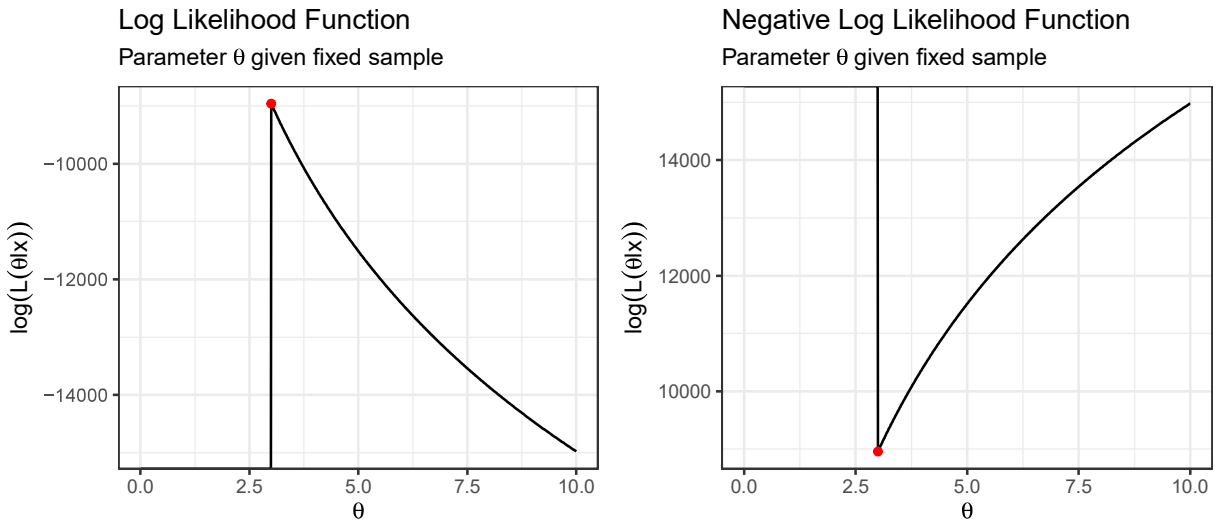

Figure 7.3.8: Plot of $log(L(\theta|\mathbf{x}))$ versus θ (left) and $-log(L(\theta|\mathbf{x}))$ (right) with the maximum likelihood estimate highlighted using a red point.

Example 7.15. Recall Example 7.8. Suppose $X_1, X_2, ..., X_n$ are *iid* Uniform$[a, b]$, where $a < b$. To find the MLEs of a and b, we first find the likelihood function.

$$\begin{aligned} L(a, b|\mathbf{x}) &= \prod_{i=1}^{n} \frac{1}{b-a}\ I(a \leq x_i \leq b) \\ &= \frac{1}{(b-a)^n}\ I(x_{(n)} \leq b)\ I(x_{(1)} \geq a) \\ &= \frac{1}{(b-a)^n}\ I(x_{(n)} \leq b)\ I(-x_{(1)} \leq a) \end{aligned}$$

Note that $L(a, b|\mathbf{x}) = 1/(b-a)^n$ is a three dimensional function of a, b, and $\mathbf{x}$ (through n). Finding the maximium of these types of functions can be difficult – we must simultaneously solve the system of equations made up of the partial derivatives of the likelihood to find the critical point and check for maximums by showing the matrix of second derivatives is negative definite. Instead of performing this complicated calculus, we'll ask R to do the maximization for us. This multi-dimensional maximization can be done in R as follows.

```
> uniform.LL<-function(x,theta,neg=FALSE){
+   a<-theta[1]
+   b<-theta[2]
+   LL<- sum(dunif(x=x,min=a,max=b,log=TRUE))
+   ifelse(!neg,LL,-LL)
+ }
> (maximization<-optim(fn=uniform.LL, #function to optimize
+     x=x.uniformab, #data
+     par = c(min(x.uniformab),max(x.uniformab)), #reasonable guess for a and b
+     neg=TRUE)) #optim minimizes so we want the negative log-likelihood
$par
[1] -0.9992082  4.9998911
$value
[1] 8958.047
```

```
$counts
function gradient
183       NA
$convergence
[1] 0
$message
NULL
```

Illustration: The likelihood function is displayed in Figure 7.3.9, and is created with the following R code. Here, we can visually confirm that $\widehat{theta} = (x_{(1)}, x_{(n)})$ maximizes the likelihood function for given a sample.

```
> a<-seq(-5,5,0.1)
> b<-seq(0,10,0.1)
> theta<-expand.grid(a,b)
> ll<-apply(X=theta,FUN=uniform.LL,MARGIN=1,x=x.uniformab,neg=FALSE)
> nll<-apply(X=theta,FUN=uniform.LL,MARGIN=1,x=x.uniformab,neg=TRUE)
> ggdat<-data.frame(a=theta[,1],b=theta[,2],ll=ll,nll=nll)
> ggdat.highlight<-data.frame(a=maximization$par[1],
+                   b=maximization$par[2],
+                   ll=uniform.LL(x=x.uniformab,theta=maximization$par,neg=FALSE),
+                   nll=uniform.LL(x=x.uniformab,theta=maximization$par,neg=TRUE))
> library(viridis) #for a nicer pallete
> g1<-ggplot(ggdat, aes(x=a,y=b,z=ll, fill=ll)) +
+   geom_raster(aes(fill = ll)) +
+   geom_contour(colour="white")+
+   geom_point(data=ggdat.highlight,aes(x=a,y=b,z=ll),color="red",show.legend=FALSE)+
+   scale_fill_viridis()+
+   theme_bw()+
+   ggtitle("Log Likelihood Function",
+           subtitle="Parameters a and b given a fixed sample")+
+   guides(fill=guide_legend(title="Log Likelihood"))
> g2<-ggplot(ggdat, aes(x=a,y=b,z=nll,fill=nll)) +
+   geom_raster(aes(fill=nll)) +
+   geom_contour(colour="white")+
+   geom_point(data=ggdat.highlight,aes(x=a,y=b,z=nll),color="red",show.legend=FALSE)+
+   scale_fill_viridis()+
+   theme_bw()+
+   ggtitle("Negative Log Likelihood Function",
+           subtitle="Parameters a and b given a fixed sample")+
+   guides(fill=guide_legend(title="Negative Log Likelihood"))
> grid.arrange(g1,g2,ncol=2)
```

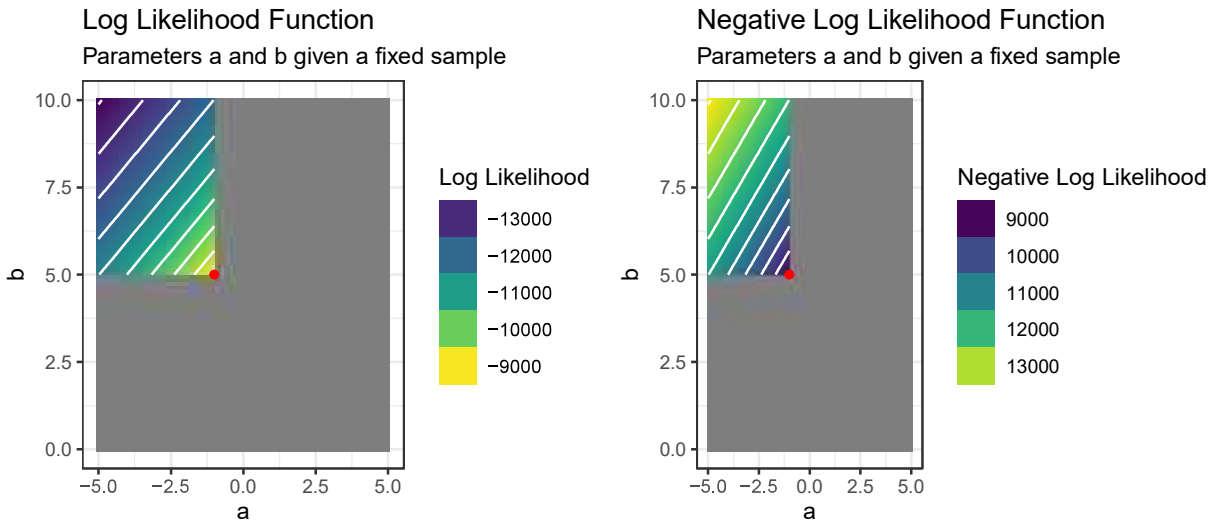

Figure 7.3.9: Plot of $log(L(\theta|\mathbf{x}))$ versus θ (left) and $-log(L(\theta|\mathbf{x}))$ (right) with the maximum likelihood estimate highlighted using a red point. The grey areas represent observations that have non-finite log liklihood values.

Example 7.16. Recall Example 7.9. Suppose $X_1, X_2, ..., X_n$ are *iid* exponential(λ), where $\lambda > 0$. To find the MLE of λ, we first find the likelihood function.

$$\begin{aligned} L(\lambda|\mathbf{x}) &= \prod_{i=1}^{n} \lambda e^{-\lambda x_i} \; I(x_i \in \mathbb{R}^+, \lambda > 0) \\ &= \lambda e^{-\lambda \sum_{i=1}^{n} x} \; I(x_{(1)} \in \mathbb{R}^+, \lambda > 0) \end{aligned}$$

Note that $L(\lambda|\mathbf{x})$ decreases as θ increases because

$$\frac{\partial L}{\partial \lambda} = (1 - \lambda \sum_{i=1}^{n} x) e^{-\lambda \sum_{i=1}^{n} x}.$$

For $\lambda <= 0$, $L(\theta|\mathbf{x}) = 0$. For $\lambda < 1/(\sum_{i=1}^{n} x)$ the likelihood is increasing and for $\lambda > 1/(\sum_{i=1}^{n} x)$ the likelihood is decreasing.Thus, the MLE of λ is the critical point $1/(\sum_{i=1}^{n} x$. This maximization can be done in R as follows.

```
> exp.LL<-function(x,theta,neg=FALSE){
+   lambda<-theta
+   LL<- sum(dexp(x=x,rate=lambda,log=TRUE))
+   ifelse(!neg,LL,-LL)
+ }
> (maximization<-optim(fn=exp.LL, #function to optimize
+     x=dat.battery,
+     par=1/mean(dat.battery), #reasonable guess for lambda
+     method="Brent", #univariate optimizer
+     lower=0, #lower bound for lambda
+     upper=1, #reasonable upper bound for lambda
+     neg=TRUE)) #optim minimizes so we want the negative log-likelihood
$par
```

```
[1] 0.0007375392
$value
[1] 410.6096
$counts
function gradient
NA       NA
$convergence
[1] 0
$message
NULL
```

Illustration: The likelihood function is displayed in Figure 7.3.10, and is created with the following R code. Here, we can visually confirm that $\widehat{theta} = x_{(n)}$ maximizes the likelihood function for given a sample.

```
> ggdat<-data.frame(theta=seq(0,1,0.0001),
+                   LL= sapply(X=seq(0,1,0.0001),FUN=exp.LL,x=dat.battery,neg=FALSE),
+                   nLL= sapply(X=seq(0,1,0.0001),FUN=exp.LL,x=dat.battery,neg=TRUE))
> ggdat.highlight<-data.frame(x=maximization$par,
+                             y=exp.LL(x=dat.battery,theta=maximization$par,neg=FALSE))
> g1<-ggplot(data=ggdat,aes(x=theta,y=LL))+
+   geom_line()+
+   geom_point(data=ggdat.highlight,aes(x=x,y=y),color="red")+
+   theme_bw()+
+   xlab(bquote(lambda))+
+   ylab(bquote(log(L(lambda*lx))))+
+   ggtitle("Log Likelihood Function",subtitle=bquote("Parameter"~lambda~"given fixed sample"))
> ggdat.highlight<-data.frame(x=maximization$par,
+                             y=exp.LL(x=dat.battery,theta=maximization$par,neg=TRUE))
> g2<-ggplot(data=ggdat,aes(x=theta,y=nLL))+
+   geom_line()+
+   geom_point(data=ggdat.highlight,aes(x=x,y=y),color="red")+
+   theme_bw()+
+   xlab(bquote(lambda))+
+   ylab(bquote(log(L(lambda*lx))))+
+   ggtitle("Negative Log Likelihood Function",
+           subtitle=bquote("Parameter"~lambda~"given fixed sample"))
> grid.arrange(g1,g2,ncol=2)
```

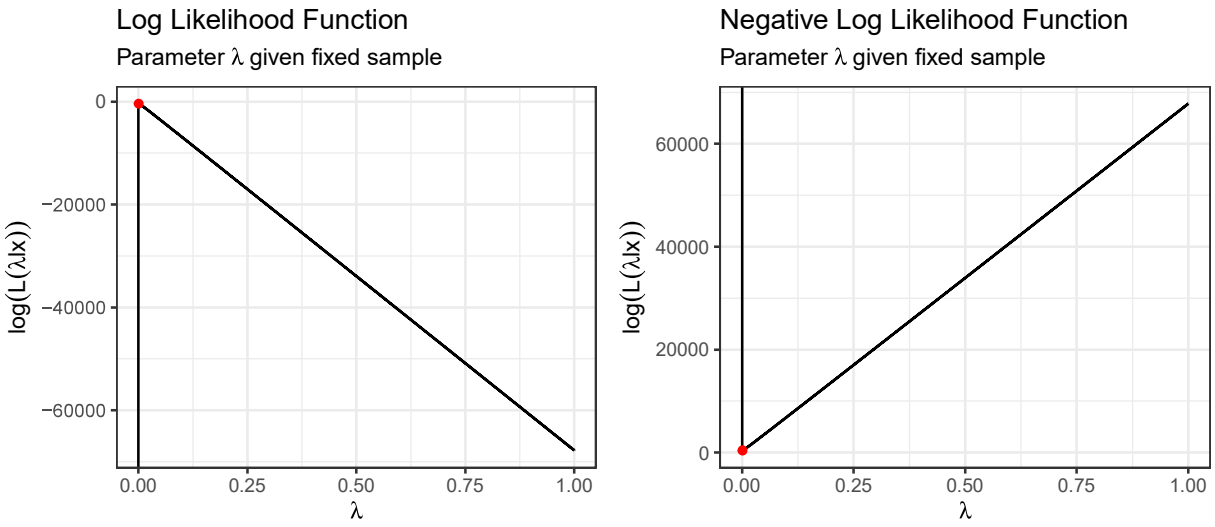

Figure 7.3.10: Plot of $log(L(\theta|\mathbf{x}))$ versus θ (left) and $-log(L(\theta|\mathbf{x}))$ (right) with the maximum likelihood estimate highlighted using a red point.

7.4 Methods of Evaluating Estimators

In Example 7.6, we derived the MOM estimator to be $\widehat{b} = 2\overline{X}$ and in Example 7.13 we derived the MLE estimator to be $\widehat{b} = X_{(n)}$.

Curiosity: Which estimator is "better?"

We will define what we mean by better and answer the question in the following subsection, but we can compare them empirically via simulation.

Example 7.17. Below, we simulate 1000 random samples of $X_1, X_2, ..., X_{10}$ *iid* Uniform$[0, 3]$ and calculate the MOM and MLE estimates for each sample.

```
> mom<-rep(NA,1000)
> mle<-rep(NA,1000)
> for(i in 1:1000){
+   n<-1000
+   x<-runif(n=n,min=0,max=3)
+   mom[i]<-2*mean(x)
+   mle[i]<-max(x)
+ }
> ggdata<-data.frame(mom=mom,
+                    mle=mle)
> g1<-ggplot(data=ggdata,aes(x=mom))+
+   geom_histogram(aes(y=..density..),col="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   theme_bw()+
+   xlab(bquote("Method of Moments Estimator"~(hat(b))))+
+   ylab("Density")+
+   ggtitle("Method of Moment Estimates",
+           subtitle=bquote("For data from"~X~"~ Uniform(0,b=3)"))
> g2<-ggplot(data=ggdata,aes(x=mle))+
+   geom_histogram(aes(y=..density..),col="black",fill="lightblue")+
+   geom_hline(yintercept=0)+
+   theme_bw()+
+   xlab(bquote("Maximum Likelihood Estimator"~(hat(b))))+
+   ylab("Density")+
+   ggtitle("Maximum Likelihood Estimates",
+           subtitle=bquote("For data from"~X~"~ Uniform(0,b=3)"))
> grid.arrange(g1,g2,ncol=2)
> c(mean=mean(mom),variance=var(mom))
mean    variance
3.001323205 0.003177065
> c(mean=mean(mle),variance=var(mle))
mean     variance
2.997080e+00 7.687222e-06
```

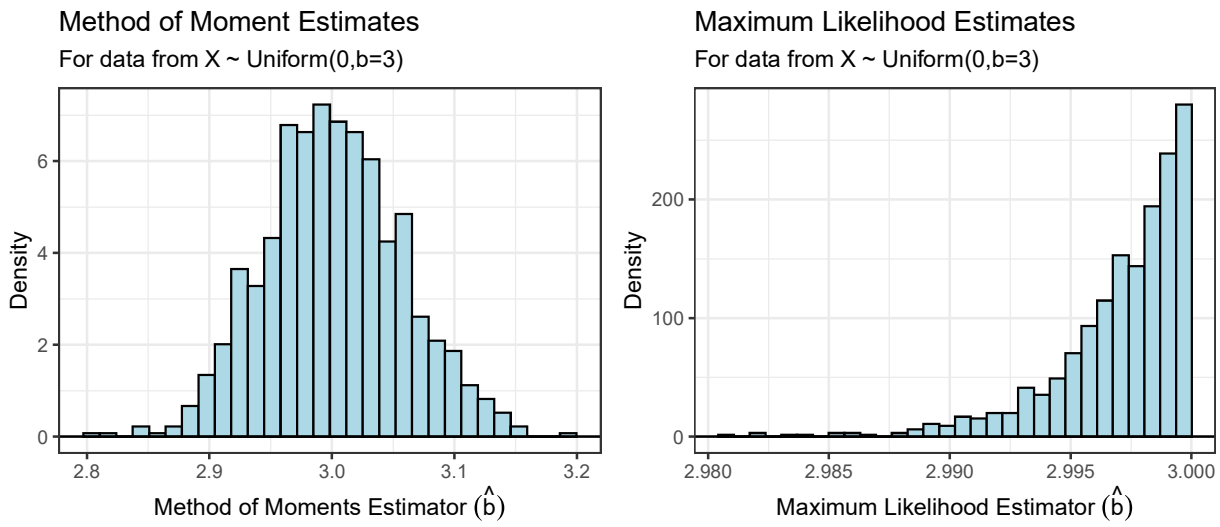

Figure 7.4.11: Histograms of MOM estimates (left) and MLE estimates (right) for 1000 repeated random samples of $X_1, X_2, ..., X_{1000}$ *iid* Uniform$[0, 3]$.

From our simulation, it appears that the MOM estimates are "good" on average ($\bar{\hat{b}} = 3.001323205$) whereas the MLE estimates, in this scenario, underestimate b on average ($\bar{\hat{b}} = 2.997080$), but only slightly. However, note that the variance of the MLE estimates ($S^2_{\hat{b}} = 7.687222 \times 10^{-6}$) is smaller than the variance of the MOM estimates ($S^2_{\hat{b}} = 0.003177065$). The implication of this is seen in Figure 7.4.11 where MLE estimates largely fall on (2.98, 3) and the MOM estimates largely fall on (2.8, 3.2).

Unresolved Issue: Which estimator is "better" – an estimator that's close on average or an estimator with smaller variance?

7.4.1 Bias, variance, and Mean Square Error

Definition 7.18. Suppose $\widehat{\theta} = W(\mathbf{X})$ is a point estimator. We call $\widehat{\theta}$ an **unbiased estimator** of θ if

$$E_\theta(\widehat{\theta}) = \theta \quad \text{for all } \theta \in \Theta.$$

In other words, the mean of the sampling distribution of $\widehat{\theta}$ is equal to θ. If

$$E_\theta(\widehat{\theta}) \neq \theta \quad \text{for some } \theta \in \Theta,$$

then we say that $\widehat{\theta}$ is biased.

Definition 7.19. Bias deals with *accuracy*; i.e., how accurate a point estimate is at estimating the population-level parameter θ, on average. We can calculate the Bias as follows

$$\text{Bias}_\theta(\widehat{\theta}) = E(\widehat{\theta} - \theta).$$

This value is often squared and interpreted as the expected squared distance between the estimator and what is being estimated.

For example, consider the bias of estimators in Example 7.17. It appears that the MOM estimates "practically unbiased" because in a large simulation we have

$$\begin{aligned} \text{Bias}(\hat{b}) &= 3.001323205 - 3 && \textbf{[MOM]} \\ &= 0.001323205 \\ \text{Bias}(\hat{b}) &= 2.997080 - 3 && \textbf{[MLE]} \\ &= -0.00292. \end{aligned}$$

The average MOM estimate has bias of 0.001 whereas the MLE estimates, in this scenario, underestimate b on average with a bias of 0.003, over twice as large as the bias of the MOM. The increased bias in the MLE estimate stems from the estimate being $X_{(n)}$ the maximum observed value; unless we observe the true maximum $b = 3$ our estimate will be less than the true value. We see here, however, that there is little difference in the accuracy of the two estimators despite this.

Question: Suppose we have two point estimators $\widehat{\theta}_1$ and $\widehat{\theta}_2$? Which one should we use? How can we compare them?
Answer: If both point estimators are unbiased; i.e., if $E(\widehat{\theta}_1) \approx E(\widehat{\theta}_2) \approx \theta$, then we would prefer the estimator with the smaller variance.

Definition 7.20. Whereas bias deals with accuracy, the variance of a point estimator describes its **precision**; i.e., how much an estimator is expected to vary in repeated use. We can calculate the Precision as follows

$$\text{Precision}_\theta(\widehat{\theta}) = \frac{1}{var_\theta(\widehat{\theta})}.$$

Small variance means high precision.
Remark: If we have two unbiased point estimators $\widehat{\theta}_1$ and $\widehat{\theta}_2$, for $\theta = 10$ and $\widehat{\theta}_1$ generally provides estimates on (-10,30) while $\widehat{\theta}_2$ generally provides estimates on (9,11) we prefer $\widehat{\theta}_2$ as an estimator. This spread is measured by the variance of the estimator.

For example, consider the variance of estimators in Example 7.17. The variance of the MLE estimates is much smaller than the variance of the MOM estimates.

$$\begin{aligned} \text{Precision}(\hat{b}) &= \frac{1}{0.003177065} && \textbf{[MOM]} \\ &= 314.7559 \\ \text{Precision}(\hat{b}) &= \frac{1}{7.687222 \times 10^{-6}} && \textbf{[MLE]} \\ &= 130086. \end{aligned}$$

This means that the MLE is more precise than the MOM, that is the MLE generally doesn't wander as far from $b = 3$ compared to the MOM estimator.

Question: How should we compare point estimators $\widehat{\theta}_1$ and $\widehat{\theta}_2$ if one of them is biased (or perhaps both are biased)? **Answer:** We prefer the estimator with the smaller mean-squared error.

Definition 7.21. The **mean-squared error** (**MSE**) of a point estimator $\widehat{\theta} = W(\mathbf{X})$ is

$$\begin{aligned} \text{MSE}_\theta(\widehat{\theta}) &= E_\theta[(\widehat{\theta} - \theta)^2] \\ &= \text{var}_\theta(\theta) + [E_\theta(\theta) - \theta]^2 \\ &= \text{var}_\theta(\theta) + \text{Bias}^2_\theta(\theta). \end{aligned}$$

Note that if θ is an unbiased estimator of θ, then for all $\theta \in \Theta$,

$$E_\theta(\theta) = \theta \implies \text{Bias}_\theta(\theta) = E_\theta(W) - \theta = 0.$$

In this case,

$$\text{MSE}_\theta(\theta) = \text{var}_\theta(\theta).$$

Remark: In general, the MSE incorporates two components:

- $\text{var}_\theta(\theta)$; this measures **precision** – variability of the estimator
- $\text{Bias}_\theta(\theta)$; this measures **accuracy** – how close the estimator on average.

We prefer estimators with small MSE because these estimators have small bias (i.e., high accuracy) and small variance (i.e., high precision).

There is no guarantee that one estimator, say $\widehat{\theta}_1$, will **always** beat the other for all $\theta \in \Theta$ (i.e., for all values of θ in the parameter space). For example, it may be that $\widehat{\theta}_1$ has smaller MSE for some values of $\theta \in \Theta$, but larger MSE for other values.

For example, consider the variance of estimators in Example 7.17. We've seen that the MOM estimates were less biased, but the MLEs were more precise. Below, we calculate the MSE which is a measure of both.

$$\begin{aligned} \text{MSE}(\hat{b}) &= 0.003177065 + 0.001323205^2 && \textbf{[MOM]} \\ &= 0.003178816 \\ \text{MSE}(\hat{b}) &= 7.687222 \times 10^{-6} + (-0.00292)^2 && \textbf{[MLE]} \\ &= 1.621362 \times 10^{-5}. \end{aligned}$$

The empirical MSE of the MLE is smaller meaning it is a "better" estimator in terms of MSE, which balances bias and precision.

Remark: The bias, variance and MSE of the estimators discussed in this section are estimated via simulation. In reality, the estimators are random variables themselves with their own distributions, their own expected values, and variances. We approximate these by simulating a sample of estimates and leaning on the law of large numbers.