Chapter 1

Code

For the simulations in this project, I used R; "a language and environment for statistical computing and graphics". I created a package with the functions needed to created the Kozachenko-Leonenko entropy estimator (KLEE), and then used this package to run simulations on samples from different statistical distributions to create the results in section **??**.

To create my package `Entropy-Estimators`, I used two of Hadley's **??** packages; `devtools` and `roxygen2`, I also used `ggplot2` to plot the graphs of the estimator bias against the sample size. `Entropy-Estimators` also has 2 dependency packages (alongside the base R packages); `dplyr` for the manipulation of data and `FNN` for the kth nearest neighbour function. I will outline the important code used for the simulations; however, the full package and a complete account of the code used can be found on my GitHub page *https://github.com/KarinaMarks/Entropy-Estimators*.

1.1 The Estimator

1.2 Exact Entropies

To consider the bias of the estimator, I had to find the exact value of entropy from a 1-dimensional normal, uniform and exponential distribution. The function written to return this for the normal distribution is `NormalEnt` with parameter `sd`, the standard deviation of the sample, we do not need the mean value for finding the entropy of the normal distribution. The function is defined as follows;

```
NormalEnt <- function(sd){
  (log(sqrt(2*pi*exp(1))*sd))
}
```

With `sd =1`, as is true in the samples considered here, we find the entropy to be given by;

```
> NormalEnt(sd=1)
[1] 1.418939
```

The function for the uniform distribution is `UniformEnt`, with parameters `min` and `max`, is defined as;

```
UniformEnt <- function(min, max){
  log(max - min)
}
```

Here we use `min=0` and `max=100` in the samples considered; thus we find the exact entropy to be given by;

```
> UniformEnt(min = 0, max = 100)
[1] 4.60517
```

Lastly, for the exponential distribution we have the function `ExpoEnt`, with only one parameter `rate`, defined below;

```
ExpoEnt <- function(rate){
  1 - log(rate)
}
```

In this paper we are using the exponential distribution with parameter `rate=1.5`, thus;

```
> ExpoEnt(rate = 1.5)
[1] 0.5945349
```

1.3 Simulations