Applications

Outline

Application 1: Bootstrapping

Estimate the standard error of the difference between the medians associated with the rat survival data given below.

Group	Data	n	Median
Treatment	16, 23, 38, 94, 99, 141, 197	7	94
Control	10, 27, 30, 40, 46, 51, 52, 104, 146	9	46

Generate B bootstrap samples, each of which consists of $n = 7$ samples drawn with replacement from 16, 23, 38, 94, 99, 141, and 197. Calculate sample standard deviation of the medians. Bootstrap estimates of the standard error of the median:

	$B = 50$	$B = 100$	$B = 250$	$B = 1000$	$B = +\infty$
Treatment	41.18	37.63	36.88	38.98	37.83
Control	20.30	12.68	9.538	13.82	13.08

Application 1: Bootstrapping

Estimate the standard error of the difference between the medians associated with the rat survival data given below.

Group	Data	n	Median
Treatment	16, 23, 38, 94, 99, 141, 197	7	94
Control	10, 27, 30, 40, 46, 51, 52, 104, 146	9	46

Generate B bootstrap samples, each of which consists of $n = 7$ samples drawn with replacement from 16, 23, 38, 94, 99, 141, and 197. Calculate sample standard deviation of the medians. Bootstrap estimates of the standard error of the median:

	$B = 50$	$B = 100$	$B = 250$	$B = 1000$	$B = +\infty$
Treatment	41.18	37.63	36.88	38.98	37.83
Control	20.30	12.68	9.538	13.82	13.08

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}$$

Standard error: $\dfrac{2}{823543}\sqrt{24271273851938} \cong 37.8347$

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}$$

Standard error: $\dfrac{2}{823543}\sqrt{242712738519382} \cong 37.8347$

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$
f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}
$$

Standard error: $\dfrac{2}{823543}\sqrt{242712738519382} \cong 37.8347$

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}$$

Standard error: $\dfrac{2}{823543}\sqrt{242712738519382} \cong 37.8347$

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$
f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}
$$

Standard error: $\dfrac{2}{823543}\sqrt{242712738519382} \cong 37.8347$

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}$$

Standard error: $\frac{2}{823543}\sqrt{242712738519382} \cong 37.8347$

Application 1: Bootstrapping (continued)

Calculate $B = +\infty$ column via:

```
> treatment := [16, 23, 38, 94, 99, 141, 197];
> X := BootstrapRV(treatment);
> Y := OrderStat(X, 7, 4);
> sqrt(Variance(Y));
```

$$f(y) = \begin{cases} 8359/823543 & y = 16 \\ 80809/823543 & y = 23 \\ 196519/823543 & y = 38 \\ 252169/823543 & y = 94 \\ 196519/823543 & y = 99 \\ 80809/823543 & y = 141 \\ 8359/823543 & y = 197 \end{cases}$$

Standard error: $\dfrac{2}{823543}\sqrt{242712738519382} \cong 37.8347$

Application 1: Bootstrapping (continued)

Bootstrapping in reliability

Use bootstrapping to determine a 95% lower confidence bound on the system reliability for a series system of three independent components using the binary failure data (y_i, n_i), where

- y_i is the number of components of type i that pass the test;
- n_i is the number of components of type i on test

for $i = 1, 2, 3$.

Component number	$i = 1$	$i = 2$	$i = 3$
Number passing (y_i)	21	27	82
Number on test (n_i)	23	28	84

Point estimate for the system reliability:

$$\frac{21}{23} \cdot \frac{27}{28} \cdot \frac{82}{84} = \frac{1107}{1288} \cong 0.8595.$$

Application 1: Bootstrapping (continued)

Bootstrapping in reliability

Use bootstrapping to determine a 95% lower confidence bound on the system reliability for a series system of three independent components using the binary failure data (y_i, n_i), where

- y_i is the number of components of type i that pass the test;
- n_i is the number of components of type i on test

for $i = 1, 2, 3$.

Component number	$i = 1$	$i = 2$	$i = 3$
Number passing (y_i)	21	27	82
Number on test (n_i)	23	28	84

Point estimate for the system reliability:

$$\frac{21}{23} \cdot \frac{27}{28} \cdot \frac{82}{84} = \frac{1107}{1288} \cong 0.8595.$$

Application 1: Bootstrapping (continued)

Bootstrapping in reliability

Use bootstrapping to determine a 95% lower confidence bound on the system reliability for a series system of three independent components using the binary failure data (y_i, n_i), where

- y_i is the number of components of type i that pass the test;
- n_i is the number of components of type i on test

for $i = 1, 2, 3$.

Component number	$i = 1$	$i = 2$	$i = 3$
Number passing (y_i)	21	27	82
Number on test (n_i)	23	28	84

Point estimate for the system reliability:

$$\frac{21}{23} \cdot \frac{27}{28} \cdot \frac{82}{84} = \frac{1107}{1288} \cong 0.8595.$$

Application 1: Bootstrapping (continued)

```
> X1 := BinomialRV(23, 21 / 23);
> X1 := Transform(X1, [[x -> x / 23], [0, 23]]);
> X2 := BinomialRV(28, 27 / 28);
> X2 := Transform(X2, [[x -> x / 28], [0, 28]]);
> X3 := BinomialRV(84, 82 / 84);
> X3 := Transform(X3, [[x -> x / 84], [0, 84]]);
> Temp := Product(X1, X2);
> T := Product(Temp, X3);
```

- There are a possible $24 \cdot 29 \cdot 85 = 59,160$ potential mass values for T.
- Of these, only 6633 are distinct because the Product procedure combines repeated values.
- The lower 95% bootstrap confidence interval bound is the 0.05 fractile of the distribution of T, which is

$$6723/9016 \cong 0.7457$$

Application 1: Bootstrapping (continued)


```
> X1 := BinomialRV(23, 21 / 23);
> X1 := Transform(X1, [[x -> x / 23], [0, 23]]);
> X2 := BinomialRV(28, 27 / 28);
> X2 := Transform(X2, [[x -> x / 28], [0, 28]]);
> X3 := BinomialRV(84, 82 / 84);
> X3 := Transform(X3, [[x -> x / 84], [0, 84]]);
> Temp := Product(X1, X2);
> T := Product(Temp, X3);
```

- There are a possible $24 \cdot 29 \cdot 85 = 59,160$ potential mass values for T.
- Of these, only 6633 are distinct because the `Product` procedure combines repeated values.
- The lower 95% bootstrap confidence interval bound is the 0.05 fractile of the distribution of T, which is

$$6723/9016 \cong 0.7457$$

Application 1: Bootstrapping (continued)

```
> X1 := BinomialRV(23, 21 / 23);
> X1 := Transform(X1, [[x -> x / 23], [0, 23]]);
> X2 := BinomialRV(28, 27 / 28);
> X2 := Transform(X2, [[x -> x / 28], [0, 28]]);
> X3 := BinomialRV(84, 82 / 84);
> X3 := Transform(X3, [[x -> x / 84], [0, 84]]);
> Temp := Product(X1, X2);
> T := Product(Temp, X3);
```

- There are a possible $24 \cdot 29 \cdot 85 = 59{,}160$ potential mass values for T.
- Of these, only 6633 are distinct because the `Product` procedure combines repeated values.
- The lower 95% bootstrap confidence interval bound is the 0.05 fractile of the distribution of T, which is

$$6723/9016 \cong 0.7457$$

Application 1: Bootstrapping (continued)


```
> X1 := BinomialRV(23, 21 / 23);
> X1 := Transform(X1, [[x -> x / 23], [0, 23]]);
> X2 := BinomialRV(28, 27 / 28);
> X2 := Transform(X2, [[x -> x / 28], [0, 28]]);
> X3 := BinomialRV(84, 82 / 84);
> X3 := Transform(X3, [[x -> x / 84], [0, 84]]);
> Temp := Product(X1, X2);
> T := Product(Temp, X3);
```

- There are a possible $24 \cdot 29 \cdot 85 = 59,160$ potential mass values for T.
- Of these, only 6633 are distinct because the `Product` procedure combines repeated values.
- The lower 95% bootstrap confidence interval bound is the 0.05 fractile of the distribution of T, which is

$$6723/9016 \cong 0.7457$$

Application 1: Bootstrapping (continued)

```
> X1 := BinomialRV(23, 21 / 23);
> X1 := Transform(X1, [[x -> x / 23], [0, 23]]);
> X2 := BinomialRV(28, 27 / 28);
> X2 := Transform(X2, [[x -> x / 28], [0, 28]]);
> X3 := BinomialRV(84, 82 / 84);
> X3 := Transform(X3, [[x -> x / 84], [0, 84]]);
> Temp := Product(X1, X2);
> T := Product(Temp, X3);
```

- There are a possible $24 \cdot 29 \cdot 85 = 59,160$ potential mass values for T.
- Of these, only 6633 are distinct because the `Product` procedure combines repeated values.
- The lower 95% bootstrap confidence interval bound is the 0.05 fractile of the distribution of T, which is

$$6723/9016 \cong 0.7457$$

Application 1: Bootstrapping (continued)

```
> X1 := BinomialRV(23, 21 / 23);
> X1 := Transform(X1, [[x -> x / 23], [0, 23]]);
> X2 := BinomialRV(28, 27 / 28);
> X2 := Transform(X2, [[x -> x / 28], [0, 28]]);
> X3 := BinomialRV(84, 82 / 84);
> X3 := Transform(X3, [[x -> x / 84], [0, 84]]);
> Temp := Product(X1, X2);
> T := Product(Temp, X3);
```

- There are a possible $24 \cdot 29 \cdot 85 = 59,160$ potential mass values for `T`.
- Of these, only 6633 are distinct because the `Product` procedure combines repeated values.
- The lower 95% bootstrap confidence interval bound is the 0.05 fractile of the distribution of `T`, which is

$$6723/9016 \cong 0.7457$$

Application 2: Kolmogorov–Smirnov test statistic

Defining formula:

$$D_n = \sup_x |F(x) - F_n(x)|$$

Computational formula:

$$D_n = \max_{i=1,2,\ldots,n} \left\{ \left| \frac{i-1}{n} - x_{(i)} \right|, \left| \frac{i}{n} - x_{(i)} \right| \right\}$$

The CDF of D_n (all parameters known case, Birnbaum, 1952):

$$P\left(D_n < \frac{1}{2n} + v\right) = n! \int_{\frac{1}{2n}-v}^{\frac{1}{2n}+v} \int_{\frac{3}{2n}-v}^{\frac{3}{2n}+v} \cdots \int_{\frac{2n-1}{2n}-v}^{\frac{2n-1}{2n}+v} g(u_1, u_2, \ldots, u_n)\, du_n \,\ldots\, du_2\, du_1$$

for $0 \le v \le \frac{2n-1}{2n}$, where

$$g(u_1, u_2, \ldots, u_n) = 1$$

for $0 \le u_1 \le u_2 \le \cdots \le u_n$.

Application 2: Kolmogorov–Smirnov test statistic (continued)

CASE I: $n = 1$

$$F_{D_1}(t) = \Pr(D_1 \leq t) = \begin{cases} 0 & t \leq \frac{1}{2} \\ 2t - 1 & \frac{1}{2} < t < 1 \\ 1 & t \geq 1 \end{cases}$$

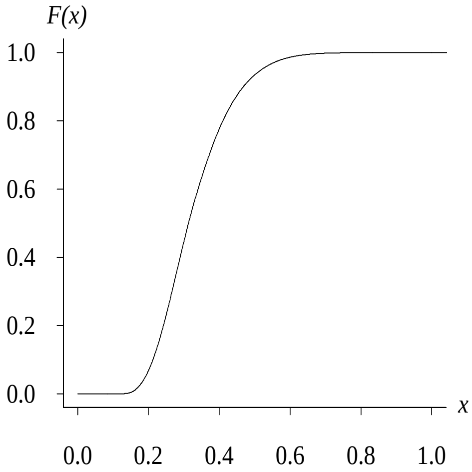
CASE II: $n = 2$

$$F_{D_2}(t) = \Pr(D_2 \leq t) = \begin{cases} 0 & t \leq \frac{1}{4} \\ 8\left(t - \frac{1}{4}\right)^2 & \frac{1}{4} < t < \frac{1}{2} \\ 1 - 2(1 - t)^2 & \frac{1}{2} < t < 1 \\ 1 & t \geq 1 \end{cases}$$

Application 2: Kolmogorov–Smirnov test statistic (continued)

Goal: `X := KSRV(n);`

CASE III: $n = 6$



Applications

Outline
Bootstrapping
Kolmogorov–Smirnov test statistic
Benford's Law

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Application 3: Benford's Law

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Application 3: Benford's Law

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Application 3: Benford's Law

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Application 3: Benford's Law

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Application 3: Benford's Law

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Benford's Law

- concerns the distribution of the leading digit in a data set
- Simon Newcomb (1881) noticed that the early pages of logarithm tables were more worn than the later pages
- if X denotes the leading digit

$$f_X(x) = P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

- $P(X = 1) = 0.301$
- $P(X = 9) = 0.0458$
- Frank Benford (1938) fit the distribution to a wide variety of data sets
- the goal: search for probability distributions satisfying Benford's Law exactly

Application 3: Benford's Law (continued)

Variate Generation

- let $X \sim$ Benford
- the cumulative distribution function of X is

$$F_X(x) = P(X = x) = \log_{10}(1 + x)$$

- let $U \sim U(0, 1)$
- a Benford random variable X is generated via

$$X \leftarrow \lfloor 10^U \rfloor$$

- this is an initial probability distribution that satisfies Benford's Law exactly: $U \sim U(0, 1)$ and $T = 10^U$ so T has probability density function

$$f_T(t) = \frac{1}{t \ln 10} \qquad 1 < t < 10$$

Application 3: Benford's Law (continued)

Variate Generation

- let $X \sim$ Benford
- the cumulative distribution function of X is

$$F_X(x) = P(X = x) = \log_{10}(1 + x)$$

- let $U \sim U(0, 1)$
- a Benford random variable X is generated via

$$X \leftarrow \lfloor 10^U \rfloor$$

- this is an initial probability distribution that satisfies Benford's Law exactly: $U \sim U(0, 1)$ and $T = 10^U$ so T has probability density function

$$f_T(t) = \frac{1}{t \ln 10} \qquad 1 < t < 10$$

Application 3: Benford's Law (continued)

Variate Generation

- let $X \sim$ Benford
- the cumulative distribution function of X is

$$F_X(x) = P(X = x) = \log_{10}(1 + x)$$

- let $U \sim U(0, 1)$
- a Benford random variable X is generated via

$$X \leftarrow \lfloor 10^U \rfloor$$

- this is an initial probability distribution that satisfies Benford's Law exactly: $U \sim U(0, 1)$ and $T = 10^U$ so T has probability density function

$$f_T(t) = \frac{1}{t \ln 10} \qquad 1 < t < 10$$

Application 3: Benford's Law (continued)

Variate Generation

- let $X \sim$ Benford
- the cumulative distribution function of X is

$$F_X(x) = P(X = x) = \log_{10}(1 + x)$$

- let $U \sim U(0, 1)$
- a Benford random variable X is generated via

$$X \leftarrow \lfloor 10^U \rfloor$$

- this is an initial probability distribution that satisfies Benford's Law exactly: $U \sim U(0, 1)$ and $T = 10^U$ so T has probability density function

$$f_T(t) = \frac{1}{t \ln 10} \qquad 1 < t < 10$$

Application 3: Benford's Law (continued)

Variate Generation

- let $X \sim$ Benford
- the cumulative distribution function of X is

$$F_X(x) = P(X = x) = \log_{10}(1 + x)$$

- let $U \sim U(0, 1)$
- a Benford random variable X is generated via

$$X \leftarrow \lfloor 10^U \rfloor$$

- this is an initial probability distribution that satisfies Benford's Law exactly: $U \sim U(0, 1)$ and $T = 10^U$ so T has probability density function

$$f_T(t) = \frac{1}{t \ln 10} \qquad 1 < t < 10$$

Application 3: Benford's Law (continued)

Variate Generation

- let $X \sim$ Benford
- the cumulative distribution function of X is

$$F_X(x) = P(X = x) = \log_{10}(1 + x)$$

- let $U \sim U(0, 1)$
- a Benford random variable X is generated via

$$X \leftarrow \lfloor 10^U \rfloor$$

- this is an initial probability distribution that satisfies Benford's Law exactly: $U \sim U(0, 1)$ and $T = 10^U$ so T has probability density function

$$f_T(t) = \frac{1}{t \ln 10} \qquad\qquad 1 < t < 10$$

Application 3: Benford's Law (continued)

More notation

- the target population of interest: a continuous random variable T with probability density function

$$f_T(t)$$

- $W = \log_{10} T$
- a random integer D denoting the order of magnitude associated with a specific value of T with probability mass function $f_D(d)$ and support $\ldots, -2, -1, 0, 1, 2, \ldots$ so that

$$10^D \le T < 10^{D+1}$$

- a random leading digit Y with support $1, 2, \ldots, 9$ with probability mass function $f_Y(y)$

Application 3: Benford's Law (continued)

More notation

- the target population of interest: a continuous random variable T with probability density function

$$f_T(t)$$

- $W = \log_{10} T$
- a random integer D denoting the order of magnitude associated with a specific value of T with probability mass function $f_D(d)$ and support $\ldots, -2, -1, 0, 1, 2, \ldots$ so that

$$10^D \le T < 10^{D+1}$$

- a random leading digit Y with support $1, 2, \ldots, 9$ with probability mass function $f_Y(y)$

Application 3: Benford's Law (continued)

More notation

- the target population of interest: a continuous random variable T with probability density function

$$f_T(t)$$

- $W = \log_{10} T$
- a random integer D denoting the order of magnitude associated with a specific value of T with probability mass function $f_D(d)$ and support $\ldots, -2, -1, 0, 1, 2, \ldots$ so that

$$10^D \leq T < 10^{D+1}$$

- a random leading digit Y with support $1, 2, \ldots, 9$ with probability mass function $f_Y(y)$

More notation

- the target population of interest: a continuous random variable T with probability density function

$$f_T(t)$$

- $W = \log_{10} T$
- a random integer D denoting the order of magnitude associated with a specific value of T with probability mass function $f_D(d)$ and support $\ldots, -2, -1, 0, 1, 2, \ldots$ so that

$$10^D \leq T < 10^{D+1}$$

- a random leading digit Y with support $1, 2, \ldots, 9$ with probability mass function $f_Y(y)$

Applications

Outline
Bootstrapping
Kolmogorov–Smirnov test statistic
Benford's Law

More notation

- the target population of interest: a continuous random variable T with probability density function
$$f_T(t)$$
- $W = \log_{10} T$
- a random integer D denoting the order of magnitude associated with a specific value of T with probability mass function $f_D(d)$ and support $\ldots, -2, -1, 0, 1, 2, \ldots$ so that
$$10^D \le T < 10^{D+1}$$
- a random leading digit Y with support $1, 2, \ldots, 9$ with probability mass function $f_Y(y)$

Application 3: Benford's Law (continued)

Example 1

Let $W \sim U(0, 2)$
$T = 10^W$ has probability density function

$$f_T(t) = \frac{1}{2t \ln 10} \qquad 1 < t < 100$$

Since D has probability mass function

$$f_D(0) = P(1 < T < 10) = 1/2$$

$$f_D(1) = P(10 < T < 100) = 1/2$$

Last step: calculate the probability mass function of the leading digit Y

Application 3: Benford's Law (continued)

Example 1

Let $W \sim U(0, 2)$

$T = 10^W$ has probability density function

$$f_T(t) = \frac{1}{2t \ln 10} \qquad 1 < t < 100$$

Since D has probability mass function

$$f_D(0) = P(1 < T < 10) = 1/2$$

$$f_D(1) = P(10 < T < 100) = 1/2$$

Last step: calculate the probability mass function of the leading digit Y

Application 3: Benford's Law (continued)

Example 1

Let $W \sim U(0, 2)$

$T = 10^W$ has probability density function

$$f_T(t) = \frac{1}{2t \ln 10} \qquad 1 < t < 100$$

Since D has probability mass function

$$f_D(0) = P(1 < T < 10) = 1/2$$

$$f_D(1) = P(10 < T < 100) = 1/2$$

Last step: calculate the probability mass function of the leading digit Y

Application 3: Benford's Law (continued)

Example 1

Let $W \sim U(0, 2)$

$T = 10^W$ has probability density function

$$f_T(t) = \frac{1}{2t \ln 10} \qquad 1 < t < 100$$

Since D has probability mass function

$$f_D(0) = P(1 < T < 10) = 1/2$$

$$f_D(1) = P(10 < T < 100) = 1/2$$

Last step: calculate the probability mass function of the leading digit Y

Application 3: Benford's Law (continued)

Example 1

Let $W \sim U(0, 2)$

$T = 10^W$ has probability density function

$$f_T(t) = \frac{1}{2t \ln 10} \qquad 1 < t < 100$$

Since D has probability mass function

$$f_D(0) = P(1 < T < 10) = 1/2$$

$$f_D(1) = P(10 < T < 100) = 1/2$$

Last step: calculate the probability mass function of the leading digit Y

Application 3: Benford's Law (continued)

Example 1 (continued) Y has probability mass function

$$
\begin{aligned}
f_Y(y) &= \Pr(Y = y) \\
&= \Pr(y < T < y+1) + \Pr(10y < T < 10(y+1)) \\
&= \int_y^{y+1} f_T(t)\,dt + \int_{10y}^{10(y+1)} f_T(t)\,dt \\
&= \int_y^{y+1} \frac{1}{2t\ln 10}\,dt + \int_{10y}^{10(y+1)} \frac{1}{2t\ln 10}\,dt \\
&= \frac{1}{2\ln 10}\left[\ln(y+1) - \ln y + \ln(10(y+1)) - \ln(10y)\right] \\
&= \log_{10}\left(\frac{y+1}{y}\right) \qquad y = 1, 2, \ldots, 9
\end{aligned}
$$

which satisfies Benford's Law exactly

Application 3: Benford's Law (continued)

Example 1 (continued) Y has probability mass function

$$\begin{aligned}
f_Y(y) &= \Pr(Y = y) \\
&= \Pr(y < T < y+1) + \Pr(10y < T < 10(y+1)) \\
&= \int_y^{y+1} f_T(t)\,dt + \int_{10y}^{10(y+1)} f_T(t)\,dt \\
&= \int_y^{y+1} \frac{1}{2t\ln 10}\,dt + \int_{10y}^{10(y+1)} \frac{1}{2t\ln 10}\,dt \\
&= \frac{1}{2\ln 10}\left[\ln(y+1) - \ln y + \ln(10(y+1)) - \ln(10y)\right] \\
&= \log_{10}\left(\frac{y+1}{y}\right) \qquad y = 1, 2, \ldots, 9
\end{aligned}$$

which satisfies Benford's Law exactly

Application 3: Benford's Law (continued)

Example 2 Let $W \sim \mathit{Triangular}(0, 1, 2)$

$$f_W(w) = \begin{cases} w & 0 < w < 1 \\ 2 - w & 1 \le w < 2 \end{cases}$$

So $T = 10^W$ has a piecewise probability density function

$$f_T(t) = \begin{cases} \dfrac{\ln t}{t(\ln 10)^2} & 1 < t < 10 \\ \dfrac{2\ln 10 - \ln t}{t(\ln 10)^2} & 10 \le t < 100 \end{cases}$$

via the APPL statements

```
> W := TriangularRV(0, 1, 2);
> g := [[w -> 10 ^ w], [0, 2]];
> T := Transform(W, g);
```

Application 3: Benford's Law (continued)

Example 2 Let $W \sim \mathit{Triangular}(0, 1, 2)$

$$f_W(w) = \begin{cases} w & 0 < w < 1 \\ 2 - w & 1 \le w < 2 \end{cases}$$

So $T = 10^W$ has a piecewise probability density function

$$f_T(t) = \begin{cases} \dfrac{\ln t}{t(\ln 10)^2} & 1 < t < 10 \\ \dfrac{2\ln 10 - \ln t}{t(\ln 10)^2} & 10 \le t < 100 \end{cases}$$

via the APPL statements

```
> W := TriangularRV(0, 1, 2);
> g := [[w -> 10 ^ w], [0, 2]];
> T := Transform(W, g);
```

Application 3: Benford's Law (continued)

Example 2 Let $W \sim \textit{Triangular}(0, 1, 2)$

$$f_W(w) = \begin{cases} w & 0 < w < 1 \\ 2 - w & 1 \le w < 2 \end{cases}$$

So $T = 10^W$ has a piecewise probability density function

$$f_T(t) = \begin{cases} \dfrac{\ln t}{t(\ln 10)^2} & 1 < t < 10 \\ \dfrac{2\ln 10 - \ln t}{t(\ln 10)^2} & 10 \le t < 100 \end{cases}$$

via the APPL statements

```
> W := TriangularRV(0, 1, 2);
> g := [[w -> 10 ^  w], [0, 2]];
> T := Transform(W, g);
```

Application 3: Benford's Law (continued)

Example 2 (continued)

The probability mass function of Y is

$$f_Y(y) = \log_{10}\left(\frac{y+1}{y}\right) \qquad y = 1, 2, \ldots, 9$$

which satisfies Benford's Law.

The calculation was performed via the APPL statements

```
> W := TriangularRV(0, 1, 2);
> Z := Benford(W);
```

Application 3: Benford's Law (continued)

Example 2 (continued)

The probability mass function of Y is

$$f_Y(y) = \log_{10}\left(\frac{y+1}{y}\right) \qquad y = 1, 2, \ldots, 9$$

which satisfies Benford's Law.

The calculation was performed via the APPL statements

```
> W := TriangularRV(0, 1, 2);
> Z := Benford(W);
```

Application 3: Benford's Law (continued)

Example 2 (continued)

The probability mass function of Y is

$$f_Y(y) = \log_{10}\left(\frac{y+1}{y}\right) \qquad y = 1, 2, \ldots, 9$$

which satisfies Benford's Law.

The calculation was performed via the APPL statements

```
> W := TriangularRV(0, 1, 2);
> Z := Benford(W);
```