01_statistical_inference - Week 2

Notes on the course
https://www.coursera.org/learn/statistical-inference

Andres FR

July 2020

Introduction to variability:

Variance: $Var(X) = E[(X - \mu)^2] = E[X^2] - E[X]^2$

Standard deviation: $\sqrt{Var(X)}$

- Example: Variance of coin with p probability of heads (1).

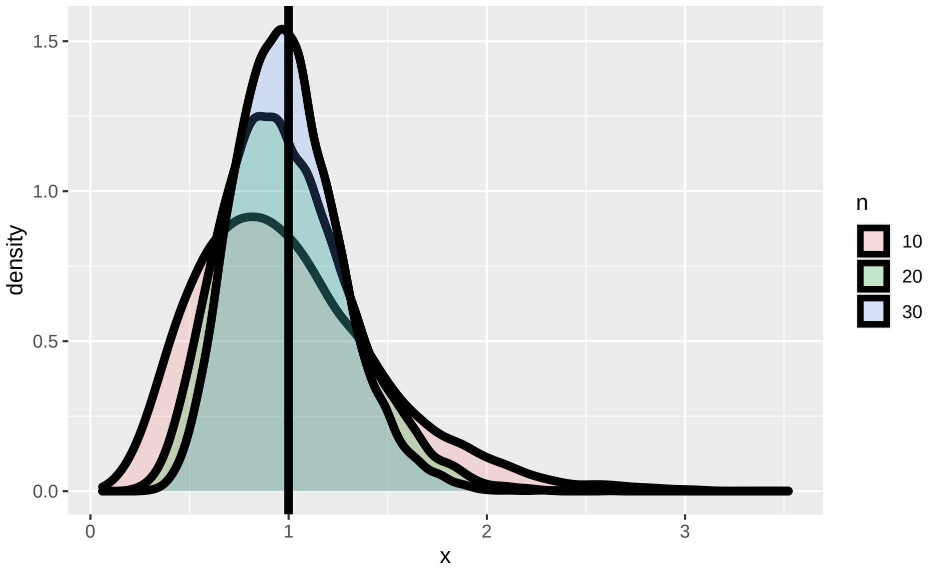
$$\begin{aligned} E[X] &= (1-p) \cdot 0 + p \cdot 1 = p \\ Var(X) &= \Big((1-p) \cdot 0^2 + p \cdot 1^2\Big) - p^2 = p - p^2 \\ &= p(1-p) \end{aligned}$$

The definition above can apply to the population variance. The **sample variance** is given as follows:

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

Variance simulation examples

But how does the *distribution* of the sample variance change with the number of samples? The following simulations, drawn from a standard normal distribution, show how the sample variance changes with no. of samples: with more samples, its expected value approaches that of the stdnormal, i.e., 1:



So the sample variance is an **unbiased estimator** of the variance. The division by $(n-1)$ is required for that. The reason is that the MSE of the samples is biased, as shown here:

$$\begin{aligned}(x_i-\bar{x}) \hat{=}& (x_i-\mu)-(\bar{x}-\mu)\\ \bar{x} :=& \frac{1}{n}\sum_i x_i\\ \frac{1}{n}\sum_i (x_i-\bar{x})^2 =& \frac{1}{n}\sum_i \{(x_i-\mu)^2+(\bar{x}-\mu)^2-2(\bar{x}-\mu)(x_i-\mu)\}\\ =& \frac{1}{n}\sum_i (x_i-\mu)^2+\frac{n}{n}(\bar{x}-\mu)^2-2(\bar{x}-\mu)\sum_i (x_i-\mu)\\ =& Var(X)+(\bar{x}-\mu)^2-2(\bar{x}-\mu)(\bar{x}-\mu)=Var(X)-(\bar{x}-\mu)^2\end{aligned}$$

Therefore,

$$E[S^2]=E[Var(X)-(\bar{x}-\mu)^2]=Var(X)-Var(\bar{X})$$

Given that we assume that the samples from X are i.i.d, and the general properties of variance, we know that $Var(\bar{X})=\frac{\sigma^2}{n}$:

$$\begin{aligned}\frac{1}{n}\sum_i (x_i-\bar{x})^2 =& Var(X)-Var(\bar{X})=Var(X)-Var(\frac{\sum_i X_i}{n})\\ =& Var(X)-\frac{\sum_i Var(X)}{n^2}=Var(X)-\frac{n\cdot Var(X)}{n^2}\\ =& Var(X)-\frac{Var(X)}{n}=\sigma^2-\frac{\sigma^2}{n}=\big(1-\frac{1}{n}\big)\sigma^2\end{aligned}$$

And therefore to "unbias" the estimator, we simply make it equal to σ^2 as follows:

$$\sigma^2 = \frac{1}{1-\frac{1}{n}} \cdot \frac{1}{n}\sum_i (x_i - \bar{x})^2 = \frac{1}{n-1}\sum_i (x_i - \bar{x})^2$$

So we see that the biased variance estimator is never bigger than the actual variance. This is understandable since any deviation between μ and $\bar{x}$ will always tend to fit the data better. Check Bessel's correction for more details: https://en.wikipedia.org/wiki/Bessel%27s_correction

---

Standard error of the mean

Recall that $E[X] = \mu$ and also that $Var(\bar{X}) = \frac{\sigma^2}{n}$. This is useful, since again we can just extract one $\bar{X}$ from our data: *we know how exactly the variance of our sample mean decreases as a function of the number of samples (and population variance).*

> A **standard error** is the standard deviation of the distribution of a statistic (like mean, regression coefficients…).
>
> The **standard error of the mean** is the square root of $Var(\bar{X})$, i.e. $\frac{\sigma}{\sqrt{n}}$.

From another angle: remember that $S^2 = \frac{\sum_{i=1}(X_i - \bar{X})^2}{n-1}$. Since we usually don't know σ, the logical estimate for the variance of the sample mean (i.e. $\frac{\sigma^2}{n}$) is $\frac{S^2}{n}$, and for the standard error is $\frac{S}{\sqrt{n}}$. **"This quantity is so important that is given a name, the sample standard error of the mean**.

Variance data example

Looking at the following dataset:

```
library(UsingR); data(father.son);
```

```
## Loading required package: MASS
```

```
## Loading required package: HistData
```

```
## Loading required package: Hmisc
```

```
## Loading required package: lattice
## Loading required package: survival
```

```
## Loading required package: Formula
```

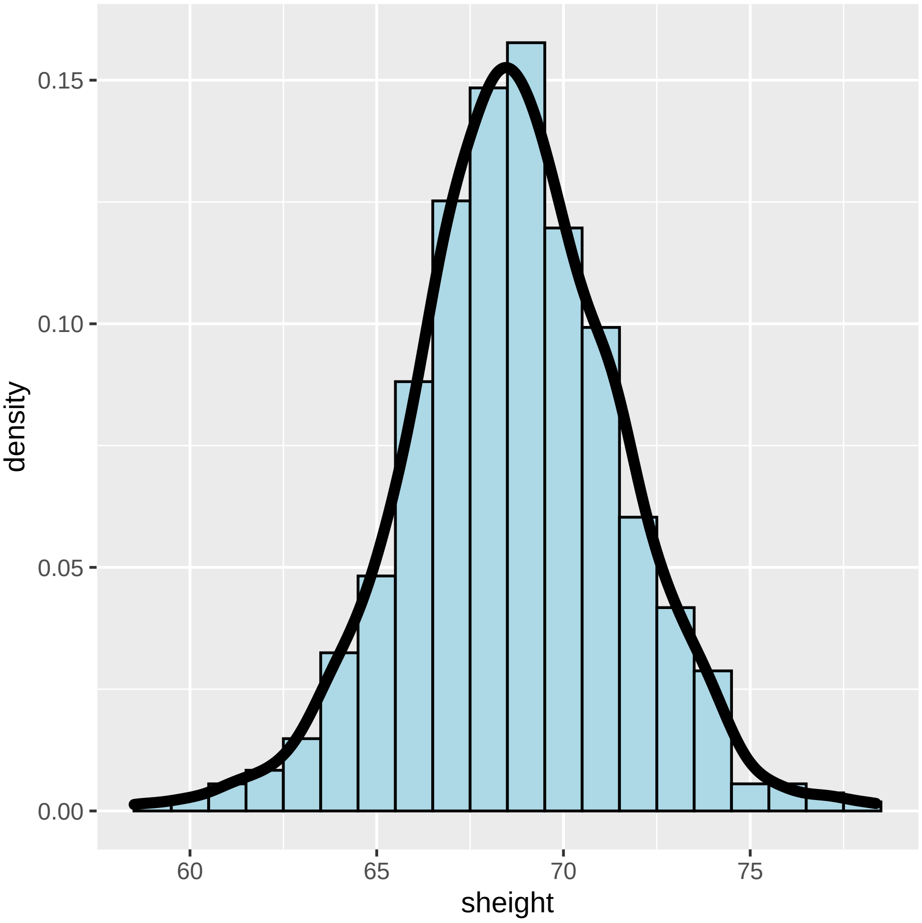
```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
##
## Attaching package: 'UsingR'
```

```
## The following object is masked from 'package:survival':
##
##     cancer
```

```
x <- father.son$sheight
n<-length(x)
```



We can compute estimators for the *variability in children heights* (e.g. the sample standard deviation), and also for the *variability of averages of children heights* (e.g. the sample standard error of the mean):

```
c(sd(x), sd(x) / sqrt(n))
```

```
## [1] 2.81470159 0.08572806
```

Binomial Distribution

- Basic explanation of Bernoulli(p) PMF, EV and Var. We usualy call the outcome 1 for success, 0 for failure.

  A Binomial RV results from summing multiple i.i.d. `Bernoulli(p)` RVs. I.e. the "total number of successes in a number of bernoulli trials.

$$Bin(n,p) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}$$

Where *n choose x* gives the total number of combinations that satisfy xs successes out of n trials in no particular order (check https://en.wikipedia.org/wiki/Pascal%27s_triangle:

$$\binom{n}{x} = \frac{n!}{x!(n-x)!} = \frac{n(n-1)(n-2)\dots(n-(k-1))}{k!}$$

As with most distributions, there are R built-ins for the binomial. Since `pbinom` is the CMF, `lower.tail` tells whether to integrate below given x or above:

```
a <- pbinom(1, size=10, prob=0.1, lower.tail=TRUE)
b <- choose(10, 0) * 0.1^0 * 0.9^10 + choose(10, 1) * 0.1^1 * 0.9^9
c(a, b)
```

```
## [1] 0.7360989 0.7360989
```

The following interactive plot allows you to observe the PMF under different n and p values:

```
library(manipulate)

binDist <- function(n, p){
  x <- 0:100
  y <- dbinom(x, n, p)
  plot(x, y, lwd=2, frame=FALSE, type="l")
}
manipulate(binDist(n, p), n=slider(1, 100, step=1),
           p=slider(0, 1, step=0.001))
```

Normal Distribution

- **Definition**: A RV X is normally distributed (i.e. $X \sim \mathcal{N}(\mu, \sigma^2)$) with mean and variance μ, σ^2 if its associated PDF is the following:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The normal with mean 0 and variance 1 is called **standard normal distribution**, usually labeled with the letter Z. All different parametrizations of $\mathcal{N}$ look identical, the only different being the values on the axes. Therefore it makes a lot of sense to talk about probabilities in terms of **standard deviations from the mean**, and refer to the standard $\mathcal{N}$.

- The probability in $(-\sigma, \sigma)$ is around 68%.
- The probability in $(-2\sigma, 2\sigma)$ is around 95%.
- The probability in $(-3\sigma, 3\sigma)$ is around 99%.
- The 10^{th} percentile is around $\mu - 1.28\sigma$ (i. e. 10% of the population is below that).
- The 5^{th} percentile is around $\mu - 1.645\sigma$
- The 2.5^{th} percentile is around $\mu - 1.96\sigma$
- The 1^{st} percentile is around $\mu - 2.33\sigma$

> **MEMORIZE:** `[68, 95, 99, 1.28, 1.645, 1.96, 2.33]`

Standarization of any RV: Given $X \sim \mathcal{N}(\mu, \sigma^2)$,

$$Z = \frac{X - \mu}{\sigma} \sim \mathcal{N}(0, 1) X = \mu + \sigma Z$$

R examples:

```r
# The 95th percentile of any normal with mu and var:
qnorm(.95, mean=0, sd=1)
```

```
## [1] 1.644854
```

```r
# The probability that any normal(mu, var) > x:
pnorm(1.645, mean=0, sd=1, lower.tail=FALSE)
```

```
## [1] 0.04998491
```

- Example: Given $X \sim \mathcal{N}(1020, 50^2)$, what is $P(x >= 1160)$?

$$P(x >= 1160) = P(z >= \frac{1160 - 1020}{50}) = P(z >= 2.8)$$

```r
pnorm(1160, mean=1020, sd=50, lower.tail=FALSE)
```

```
## [1] 0.00255513
```

```r
pnorm(2.8, mean=0, sd=1, lower.tail=FALSE)
```

```
## [1] 0.00255513
```

- Example 2: What is the 75 percentile? Here we can find the corresponding percentile on the normal, say v, and our desired result is $1020 + 50 \cdot v$.

```r
qnorm(0.75, mean=1020, sd=50)
```

```
## [1] 1053.724
```

```r
qnorm(0.75, mean=0, sd=1) * 50 + 1020
```

```
## [1] 1053.724
```

Poisson distribution

Also very important, has the following PMF:

$$P(X = x, \lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$$

> **Note**: assuming data is Poisson distributed means that the population EV and Variance are equal, which a checkable assumption.

- Some applications: Modeling count data, event-time or survival data, contigency tables (multi-dimensional feature histograms like sorting data by hair color, ethnicity…), **approximating binomials for large n and small p** (e.g. epidemiology, ecosystems… in those fields Poisson models are very extended).
- Example: modeling occurrency rates: $X \sim Poisson(\lambda t)$ where $\lambda = E[X/t]$ is the expected count per unit of time, and t is the total monitoring time: note that *lambda* **has units here**: counts *per unit time*, a very common use of this distribution.
  - The probability that 3 or fewer counts happen within 4 "time units", where the EV per time unit is 2.5 can be retrieved as follows:

```r
ppois(3, lambda=4*2.5)
```

```
## [1] 0.01033605
```

- Poisson approximation to the binomial: Assuming $X \sim Bin(n, p)$ for large n and small p, and given the fact that $\lambda = np$ due to EV definition, we obtain:

$$\begin{aligned}\lim_{n\to\infty} \binom{n}{k} p^k (1-p)^{n-k} &= \lim_{n\to\infty} \frac{n(n-1)(n-2)\dots(n-(k-1))}{k!} \left(\frac{\lambda}{n}\right)^k \left(1-\frac{\lambda}{n}\right)^{n-k} \\ &= \lim_{n\to\infty} \frac{n^k + O(n^{k-1})}{k!} \cdot \frac{\lambda^k}{n^k} \cdot \left(1-\frac{\lambda}{n}\right)^{n-k} = \lim_{n\to\infty} \frac{\lambda^k}{k!} \cdot \left(1-\frac{\lambda}{n}\right)^{n-k} \\ &= \frac{\lambda^k}{k!} \lim_{n\to\infty} \cdot \left(1-\frac{\lambda}{n}\right)^{n-k} = \frac{\lambda^k}{k!} e^{-\lambda}\end{aligned}$$

Asymptotics (Text)

> We can use asymptotics to help is figure out things about distributions without knowing much about them to begin with. A profound idea along these lines is the **Central Limit Theorem**. It states that the distribution of averages is often normal, even if the distribution that the data is being sampled from is very non-normal. This helps us *create robust strategies for creating statistical inferences when we're not willing to assume much about the generating mechanism of our data.*

Asymptotics and LLN

> **Asymptotics** is the term for the behavior of statistics as the sample size (or other relevant quantity) limits to infinity (or other relevant number). It is a very useful tool to explore statistics in many different contexts.

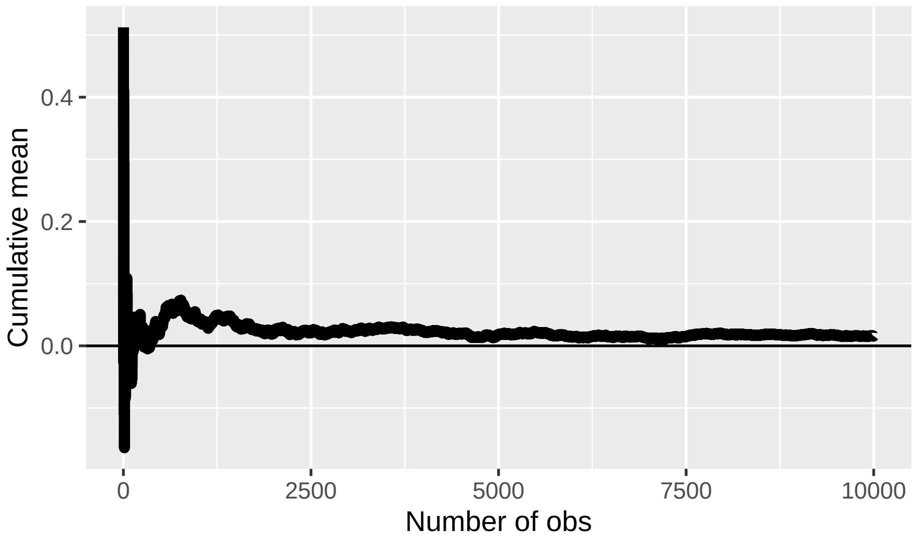
- **Law of large numbers** (LLN): The more an experiment is repeated, the closer the cumulative average gets to the EV.

We say that an estimator is consistent if it converges to what we want to estimate. The LLN states that the sample mean of iid samples is consistent for the population mean. E.g. the sample proportion of heads when flipping a coin converges to the p of that coin.

Consistency is a good property of estimators, it comes to be that *if we go through the trouble of collecting enough data, the answer is good enough.*

> **The sample variance and sample sd of iid RVs are consistent as well**

```r
n <- 10000; means <- cumsum(rnorm(n)) / (1  : n); library(ggplot2)
g <- ggplot(data.frame(x = 1 : n, y = means), aes(x = x, y = y))
g <- g + geom_hline(yintercept = 0) + geom_line(size = 2)
g <- g + labs(x = "Number of obs", y = "Cumulative mean")
g
```



The Central Limit Theorem

Perhaps the most important theorem in all statistics.

> **CLT intuition**: The distribution of averages of iid variables (properly normalized) becomes that of a standard normal as the sample size increases. As already seen, the EV is μ and sd is $\frac{\sigma}{\sqrt{n}}$

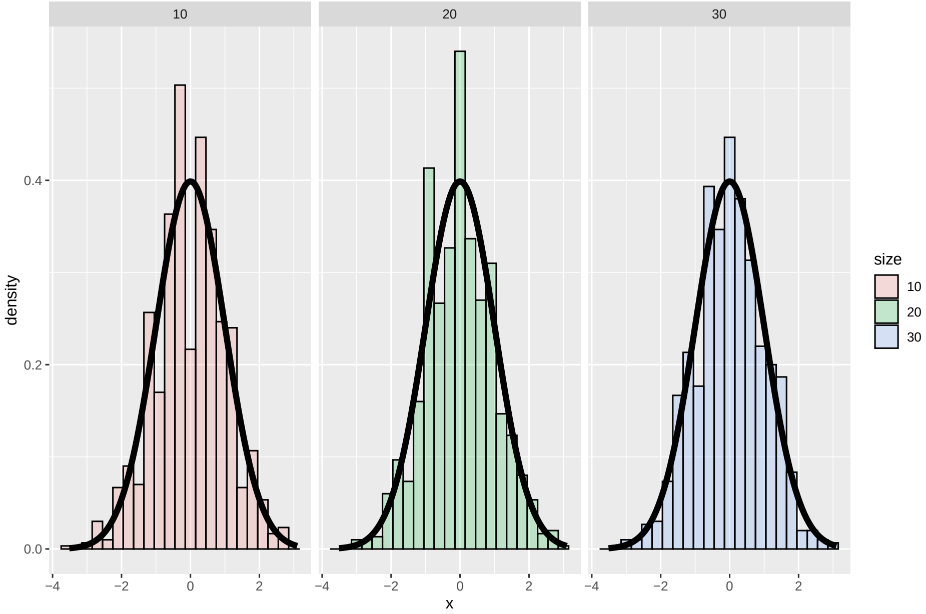
More details: https://en.wikipedia.org/wiki/Central_limit_theorem

Examples: From our calculations before, we know that the mean value of such averaged distribution will be the population's EV, and the standard deviation will be $\frac{\sigma}{\sqrt{n}}$.

So, for a coin $X \sim Bernoulli(p)$ with standard error $\sqrt{\frac{p(1-p)}{n}}$, if we count the sample proportion $\hat{p}$ (which is a RV itself), we have that the following should follow a standard normal distribution:

$$\frac{\sqrt{n}(\hat{p} - p)}{\sqrt{p(1-p)}}$$

The following is an analogous example with a die:



Also note that the fact that it converges in the limit doesn't tell how fast it converges: **uniform distributions converge faster than highly skewed ones**.

Asymptotics and confidence intervals

So the CLT tells us that $\bar{X}$ is *approximately* normal with EV μ and sd $\frac{\sigma}{\sqrt{n}}$. We can use this information to develop confidence intervals for our estimation of the means. We see that, **for any required confidence, the size of the interval decreases when n increases**. Thus, sampling more allows us to be more precise. E.g. $\bar{X} \pm \frac{1.96\sigma}{\sqrt{n}}$ is called **the 95% interval for μ**, this means that if we were to get repeated average samples from a population, 95% of the extracted intervals would contain the population mean.

Example: Confidence interval for a coin flip: Given our desired normal quantile $z_{1-\alpha/2}$:

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}}$$

> In the cases where we don't know p, we can replace it with $\hat{p}$ yielding the **Wald confidence interval**.

For fair coins, it turns out that this confidence interval is largest when $p = 0.5$. This has a very convenient approximation for 95% intervals: if we set $z = 2$, the (approx. 95%) interval for $p = 0.5$ ends up being $\hat{p} \pm \frac{1}{\sqrt{n}}$ which can be handy for quick calculations. So we can roughly see that 100 samples will provide plus minus 10%, if we want 1% we need around 10000, 1 million for 0.1%, etc.

Example: Binomial CI: Assuming a poll of 100 where 56 are in favor, the 95% confidence interval assuming the $\hat{p} \pm \frac{1}{\sqrt{n}}$ rule is:

```
.56 + c(-1, 1) * qnorm(.975) * sqrt(.56 * .44 / 100)
```

```
## [1] 0.4627099 0.6572901
```

> We can also use the `test` functions from R (dollar sign is an accessor by name). This binomial test with 56 successes out of 100 trials returns a confidence interval guaranteed to be of 0.95 or higher, regardless of the sample size n. We observe that it returns a very similar interval to our back-of-the envelope method. "This so-called **exact procedures** are a nice complement to large sample procedures, tend to involve computations that cannot be done by hand, are conservative but nonetheless don't rely on the CLT".

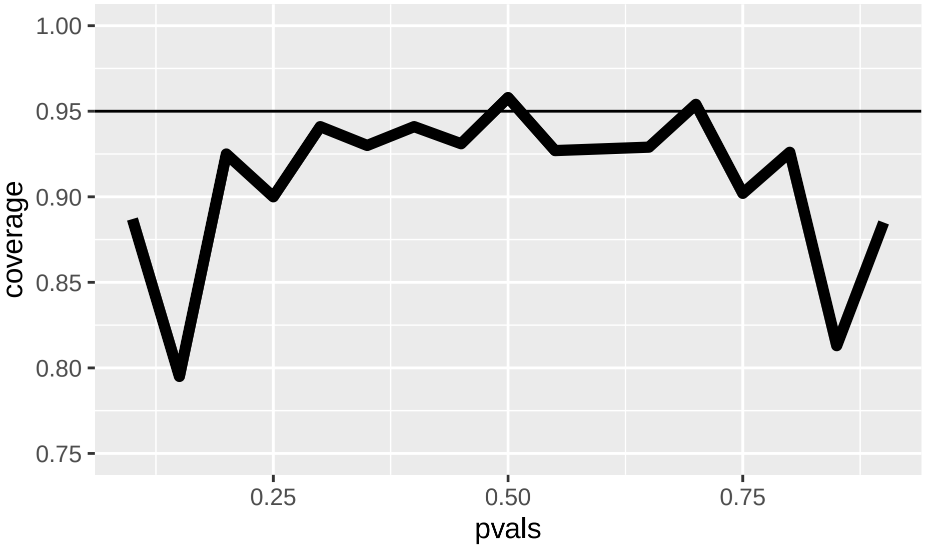
```
binom.test(56, 100)$conf.int
```

```
## [1] 0.4571875 0.6591640
## attr(,"conf.level")
## [1] 0.95
```

Check the docs: https://www.rdocumentation.org/packages/mosaic/versions/1.7.0/topics/binom.test

Simulation: We will flip a coin with different p probabilities, and check how often our Wald interval contains the actual p. Observe the result:

```
n <- 20; pvals <- seq(.1, .9, by = .05); nosim <- 1000
coverage <- sapply(pvals, function(p){
  phats <- rbinom(nosim, prob = p, size = n) / n
  ll <- phats - qnorm(.975) * sqrt(phats * (1 - phats) / n)
  ul <- phats + qnorm(.975) * sqrt(phats * (1 - phats) / n)
  mean(ll < p & ul > p)
})
```



How can it be that our 95% interval did not do so well for extreme p coins? the reason, as we mentioned before, is that the CLT doesn't say how fast do averages converge to a normal: in fact, the more extreme our p the slower the convergence, and our Wald interval assumes normality. Hence the error.

How to fix it? Note that increasing sample size can work well. But there is another quickfix, called the **Agresti/Coull interval**: Add two successes and failures (i.e. `x+=2, n+=4`), and repeat procedure as normal but with these artificially updated values. The result is a wider interval, but with much better guarantees for the interval. It can be considered "a little conservative". Nonetheless, **categorical recommendation:** The Agresti/Coull should be generally used instead of the Wald, in general.

Poisson CI example

- A nuclear pump failed 5 times in 94.32 days. Give a 95% CI for the failure rate per day.

We will assume Poisson dist. and use our "estimate plus minus quantile times SEest" model, Wald variant:

$$\begin{aligned}
X \sim & Poisson(\lambda t)\\
E[X] = Var(X) = & \lambda/t\\
t = & day\\
z = & 1.96\\
n = t = & 94.32\\
\hat{\lambda} = X/t = & 5/94.32 \approx 0.053\\
Var(\hat{\lambda}) = & \hat{\lambda} \pm z \cdot \sqrt{\frac{Var(X)}{n}} \approx 0.053 \pm 1.96\sqrt{\frac{0.053}{94.32}} = (0.065, 0.099)
\end{aligned}$$

Note that EV and Var are frequencies and have a unit. We replaced the unknown population values with our empirical ones, and obtained the standard Wald interval.

Note also that, for Poisson, the unbiased estimator is just bad: https://en.wikipedia.org/wiki/Bias_of_an_estimator. See also attached paper.

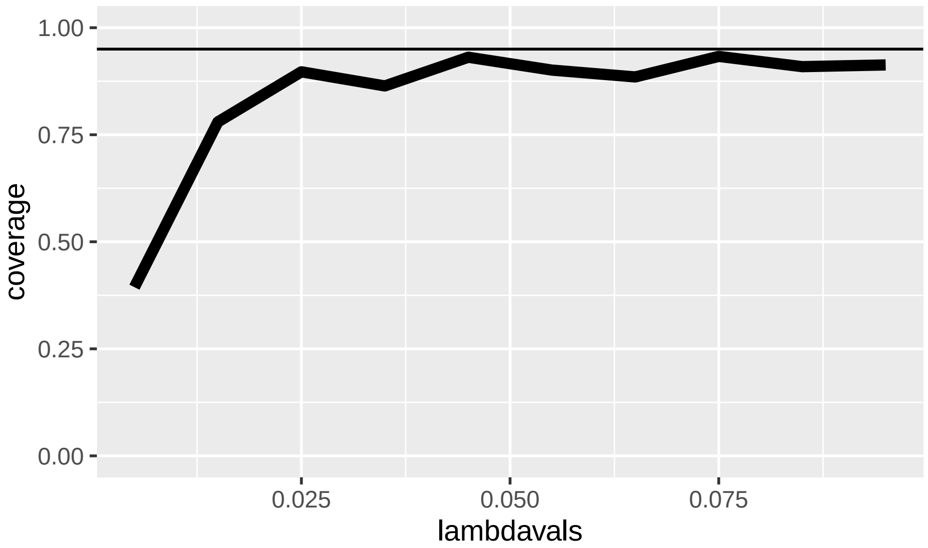
Also note that, like for the Binomial, we also have a `poisson.test` function that retrieves us intervals with guaranteed **at least** the required confidence, and doesn't depend on simulation or CLT:

```
poisson.test(5, 94.32, conf.level=0.95)
```

```
##
##  Exact Poisson test
##
## data:  5 time base: 94.32
## number of events = 5, time base = 94.32, p-value < 2.2e-16
## alternative hypothesis: true event rate is not equal to 1
## 95 percent confidence interval:
##  0.01721254 0.12371005
## sample estimates:
## event rate
## 0.05301103
```

We see that the guaranteed interval is much wider than our Wald interval. Why is that? This is how the Wald CI performs in a simulation with similar parameters:

```
lambdavals <- seq(0.005, 0.10, by = .01); nosim <- 1000
t <- 100
coverage <- sapply(lambdavals, function(lambda){
  lhats <- rpois(nosim, lambda = lambda * t) / t
  ll <- lhats - qnorm(.975) * sqrt(lhats / t)
  ul <- lhats + qnorm(.975) * sqrt(lhats / t)
  mean(ll < lambda & ul > lambda)
})
```



We observe that as the frequency increases, the coverage improves. Low lambdas have very poor coverage due to the fewer events thus less information to infer. Increasing the recording time improves this.

> Summary: **The Poisson and binomial case have exact procedures to obtain confidence intervals**, but a quick filx for small sample size binomial calculations is to add 2 successes and 2 failures (Agresti/Coull).