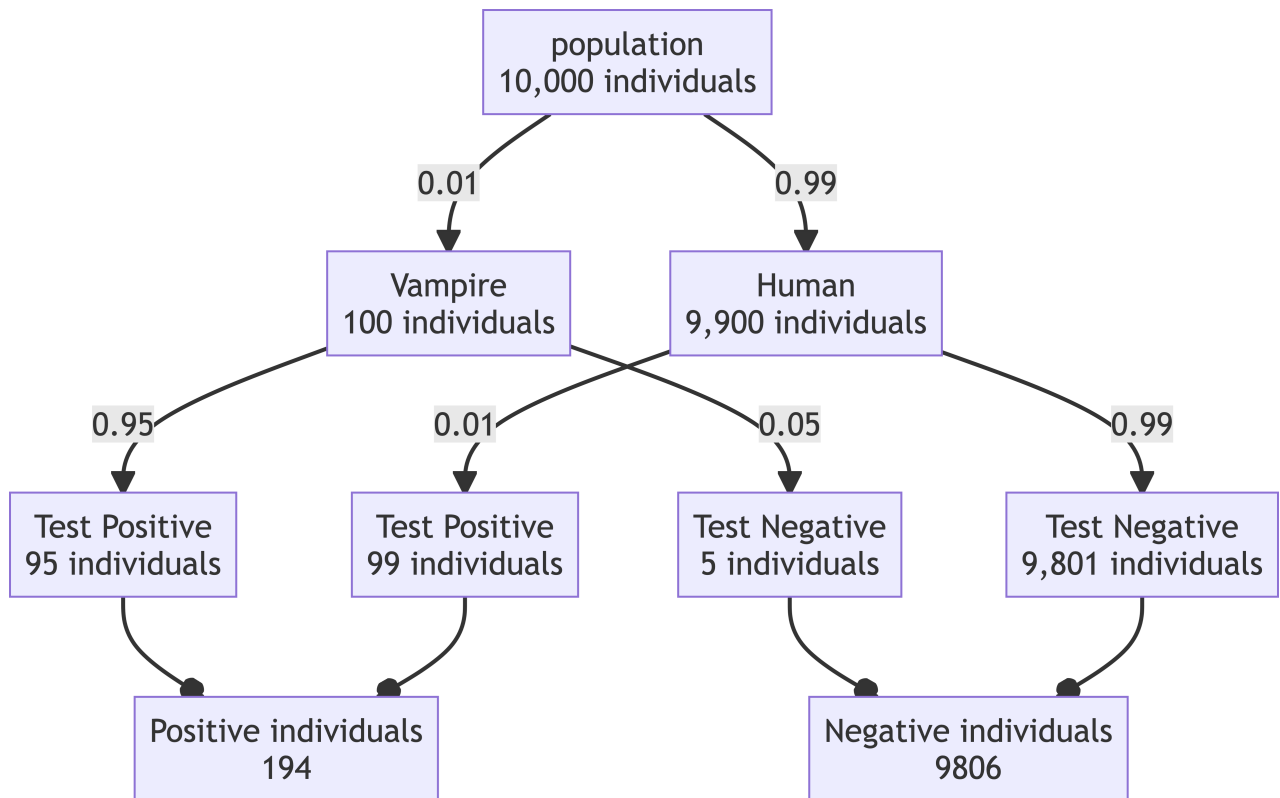# Starting Sampling

Josef Fruehwald

> **ℹ Listening**

```r
library(tidyverse)
library(patchwork)
library(here)
library(gt)
source(here("_defaults.R"))
```

Classic Base Rate Issues

```
                    ┌──────────────────────┐
                    │     population       │
                    │  10,000 individuals  │
                    └──────────────────────┘
              0.01                      0.99
       ┌──────────────────┐      ┌──────────────────┐
       │     Vampire      │      │      Human       │
       │  100 individuals │      │ 9,900 individuals│
       └──────────────────┘      └──────────────────┘
    0.95            0.01      0.05               0.99
┌──────────────┐ ┌──────────────┐ ┌──────────────┐ ┌──────────────────┐
│ Test Positive│ │ Test Positive│ │ Test Negative│ │  Test Negative   │
│95 individuals│ │99 individuals│ │5 individuals │ │9,801 individuals │
└──────────────┘ └──────────────┘ └──────────────┘ └──────────────────┘
        └────────────┐ ┌─────┘             └──────┐ ┌────────┘
          ┌────────────────────┐           ┌────────────────────┐
          │ Positive individuals│          │ Negative individuals│
          │        194          │          │        9806          │
          └────────────────────┘           └────────────────────┘
```

Plot of the base rate vs P(vampire | positive test)

```r
tibble(
  # might as well get logarithmic
  base_rate = 10^(seq(-3, -1, length = 20)),
  vamp_and_pos = base_rate * 0.95,
  vamp_and_neg = base_rate * 0.05,
  human_and_pos = (1-base_rate) * 0.01,
  human_and_neg = (1-base_rate) * 0.99,
  p_vamp_pos = vamp_and_pos/(vamp_and_pos + human_and_pos),
  p_hum_neg = human_and_neg/(vamp_and_neg + human_and_neg)
) -> test_metrics

test_metrics |>
  ggplot(aes(base_rate, p_vamp_pos))+
```

```
geom_point(color = "steelblue",
          size = 3)+
geom_line(color = "steelblue",
         linewidth = 1)+
scale_x_log10()+
ylim(0,1)+
labs(x = "P(vampire)",
     y = "P(vampire | positive)",
     subtitle = "P(positive | vampire) = 0.95\nP(positive | human) = 0.01",
     title = "Positive Predictive Value") +
theme(plot.subtitle = element_text(size = 12))
```
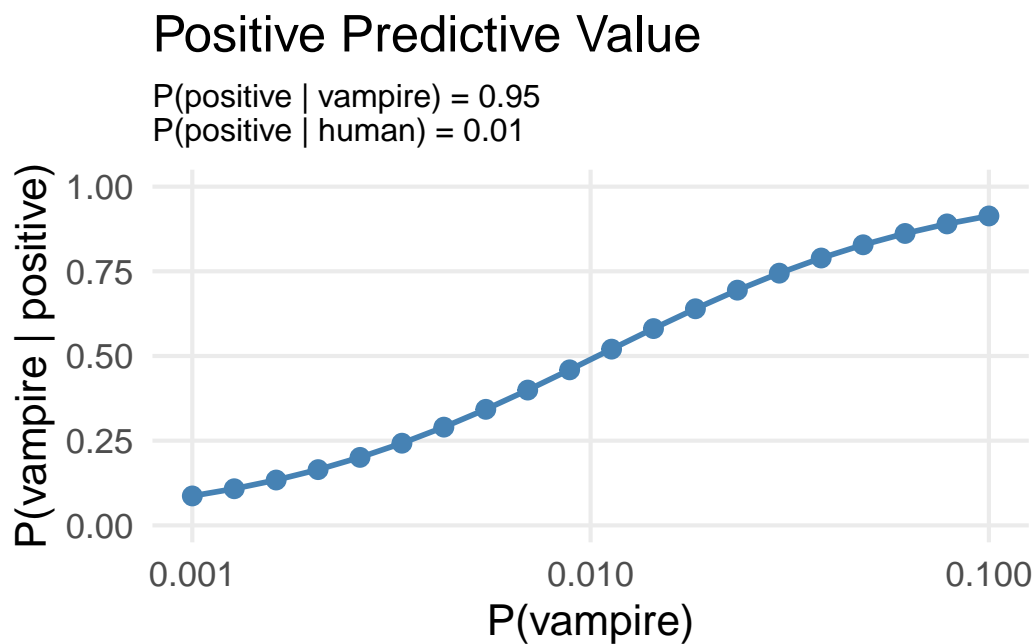
Positive Predictive Value

P(positive | vampire) = 0.95
P(positive | human) = 0.01

Figure 1: Probability someone *is* a vampire, given that they tested positive, relative to the base rate of being a vampire

```
test_metrics |>
  ggplot(aes(base_rate, p_hum_neg))+
    geom_point(color = "steelblue",
              size = 3)+
    geom_line(color = "steelblue",
             linewidth = 1)+
```

```
    scale_x_log10()+
    labs(x = "P(vampire)",
        y = "P(human | negative)",
        subtitle = "P(positive | vampire) = 0.95\nP(positive | human) = 0.01",
        title = "Negative Predictive Value") +
    theme(plot.subtitle = element_text(size = 12))
```

Negative Predictive Value

P(positive | vampire) = 0.95
P(positive | human) = 0.01
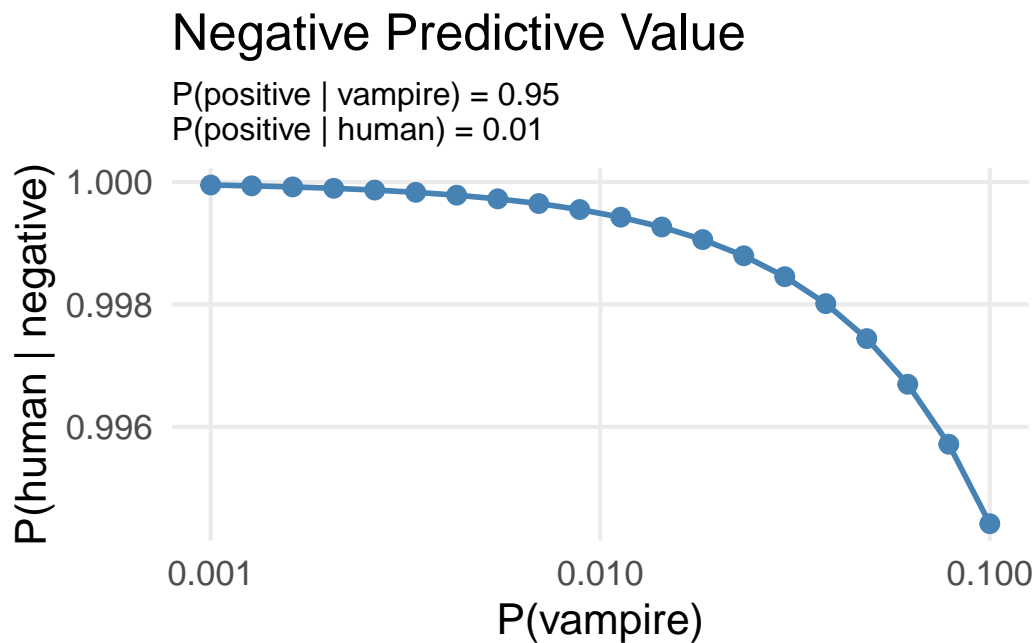
Figure 2: Probability of being a human given a negative test, relative to the base rate of being a vampire.

Tibble grid sampling

```
grid <- tibble(
  prob = seq(0.0001, 0.9999, length = 5000),
  prior_unstd = case_when(
    prob < 0.5 ~ 0,
    .default = 1
  ),
  prior_std = prior_unstd/sum(prior_unstd),
  data = dbinom(6, size = 9, prob = prob),
```

```
    posterior_unstd = prior_std * data,
    posterior = posterior_unstd / sum(posterior_unstd)
)

grid |>
  ggplot(aes(prob, prior_std))+
    geom_line()+
    labs(y = "prior density",
         title = "Prior") ->
  prior_plot

grid |>
  ggplot(aes(prob, data))+
    geom_line()+
    labs(y = "data density",
         title = "Data") ->
  data_plot

grid |>
  ggplot(aes(prob, posterior))+
    geom_line() +
    labs(y = "posterior density",
         title = "Posterior") ->
  posterior_plot

prior_plot | data_plot | posterior_plot
```
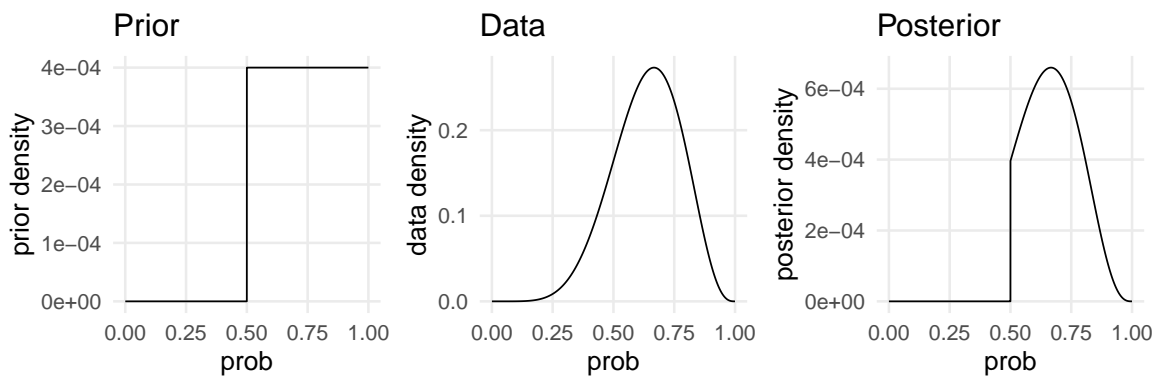

Figure 3: Prior, Data, Posterior

5

Sampling from the posterior

```r
grid |>
  sample_n(size = 1e4,
           replace = T,
           weight = posterior)->
  posterior_samples
```

```r
renv::install("tidybayes")
```

```r
library(tidybayes)
```

```r
posterior_samples |>
  ggplot(aes(prob))+
    stat_halfeye(point_interval = median_hdci)+
    geom_line(aes(y = posterior/max(posterior)),
              linetype = 2)+
    xlim(0,1)+
    theme(
      axis.title.y = element_blank(),
      axis.text.y = element_blank(),
      panel.grid.major.y = element_blank()
    )
```
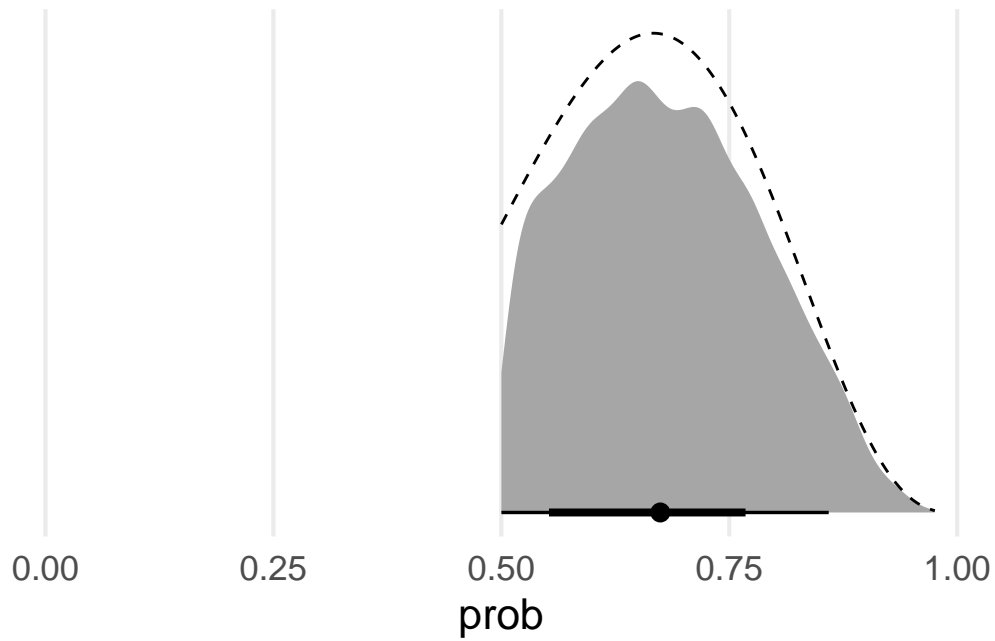
```
posterior_samples |>
  median_hdci(prob, .width = c(0.5, 0.95)) |>
  gt() |>
  fmt_number(decimals = 2)
```

prob	.lower	.upper	.width	.point	.interval
0.67	0.58	0.74	0.50	median	hdci
0.67	0.50	0.86	0.95	median	hdci

BRMS

```
library(brms)
```

```
tibble(
  water = 6,
  samples = 9
)->
  water_to_model
```

```r
water_form <- bf(
    water | trials(samples) ~ 1,
    family = binomial(link = "identity")
)
```

```r
brm(
    water | trials(samples) ~ 1,
    data = water_to_model,
    family = binomial(link = "identity"),
    prior(beta(1, 1), class = Intercept, ub = 1, lb = 0),
    file_refit = "on_change",
    file = "water_fit.rds"
) ->
    water_model
```

```r
water_model
```

```
 Family: binomial
  Links: mu = identity
Formula: water | trials(samples) ~ 1
   Data: water_to_model (Number of observations: 1)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.64      0.14     0.35     0.88 1.00     1598     1945

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
library(gtsummary)
```

```r
water_model |>
    gtsummary::tbl_regression(intercept = T)
```

Characteristic	Beta	95% CI
(Intercept)	0.64	0.35, 0.88

```r
fs::dir_create("tables")
```

```r
water_model |>
  spread_draws(b_Intercept) |>
  summarise(Intercept = list(b_Intercept)) |>
  gt() |>
  gtExtras::gt_plt_dist(Intercept)  -> modtable

modtable |>
  gtsave("tables/model.png")
```