My demo notebook

Andrew D. Nguyen

2025-04-19

Table of contents

Preface

Hello, welcome to my demo notebook. The purpose of this notebook is to log ideas I've explored. It'll be useful for me, and hopefully, also for you!

Part I

Causal inference section

1 Title: Causal diagram simulations

Start Date: 2025-03-23
Last modified: 2025-04-19

2 Load libraries

```r
library(tidyverse) # for data wrangling
library(gtsummary) # for formatting model outputs into a nice html table
library(DiagrammeR) # for drawing dags
library(webshot2) #to help render doc
```

3 What to condition on and what not to condition on

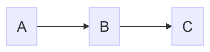
In causal inference, the building of statistical models depends on how the variables from a collected dataset relate to one another. Here, I show the arrangement of variables with a Directed Acyclic Graphs (DAG)s. The different Dags show cases where it is not appropriate to condition on certain variables.

3.1 Chain in a DAG

Here is a chain, where B is a mediator. Mediators should not be conditioned on because it will limit the association between A and C.

```
mermaid("graph LR
        A-->B
        B-->C")
```

```
file:///C:\Users\anbe6\AppData\Local\Temp\RtmpiW4ShR\file1e7847214bcf\widget1e7844344127.htm
```

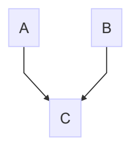

A
B
C

3.2 Colliders in a DAG

Here is a collider, where C is a collider. Colliders should not be conditioned on because there will be a spurious association between a and b.

```
mermaid("graph TD
        A-->C
        B-->C")
```

```
file:///C:\Users\anbe6\AppData\Local\Temp\RtmpiW4ShR\file1e7838d71dbe\widget1e7811736783.htm
```

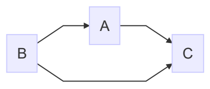

A
B
C

3.3 Confounders in a DAG

Here is a confounder, where B is a confound. Confounders SHOULD be conditioned on.

```
mermaid("graph LR
        A-->C
        B-->A
        B-->C")
```

```
file:///C:\Users\anbe6\AppData\Local\Temp\RtmpiW4ShR\file1e7841ad4224\widget1e787d2958d2.htm
```


B
A
C

4 Why we should not condition on a mediator

To determine why we should not condition on a mediator, we can simulate a chain DAG and compare a model where we condition on B vs a model without B (marginal model).

$$a \sim Normal(50, 5)$$
$$b \sim a + \epsilon$$

$$c \sim b + \epsilon$$

Random error: $\epsilon \sim Normal(0, 1)$

Note: It is expected for c and a to have 1:1 when b transmits effect.

```
# simulate mediator
#A->B->C
a<-rnorm(n=100,mean=50,sd=5)
b<-a+rnorm(n=100,mean=0,sd=1)#b is a function of a with random error centered around 0 c
c<-b+rnorm(n=100,mean=0,sd=1)

#now fit a model of c~a
mod1<-lm(c~a)
mod1|>
  tbl_regression()
```

```
ggplot(data=tibble(a,c),aes(x=a,y=c))+geom_point()+stat_smooth(method="lm")
```

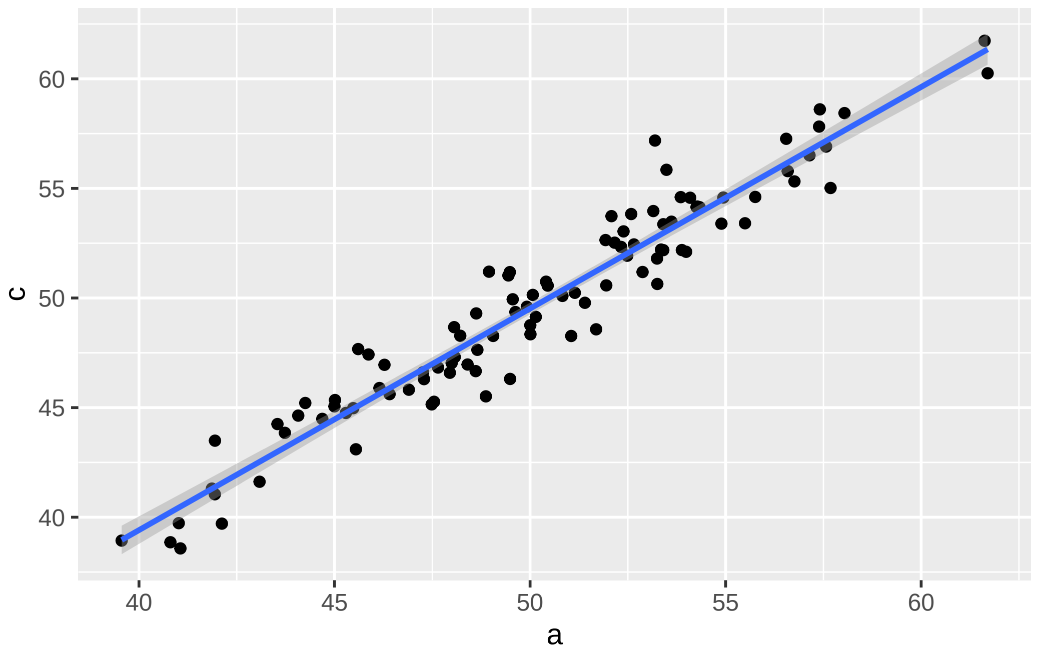
```
`geom_smooth()` using formula = 'y ~ x'
```

Characteristic	Beta	95% CI	p-value
a	1.0	0.95, 1.1	<0.001

Abbreviation: CI = Confidence Interval

Characteristic	Beta	95% CI	p-value
a	0.09	-0.13, 0.30	0.4
b	0.93	0.72, 1.1	<0.001

Abbreviation: CI = Confidence Interval



```r
#let's see how the estimate changes when we condition on a mediator
mod2<-lm(c~a+b)
mod2|>
  tbl_regression()
```

Conditioning on the mediator (b) reduces the causal effect of a–>c

4.1 Changing the mediator to a categorical variable to visualize

```r
# try to set b as a categorical variable
a<-rnorm(n=100,mean=50,sd=5)
b<-if_else(a<50,0,1)
c<-b+rnorm(n=100,mean=0,sd=1)
```

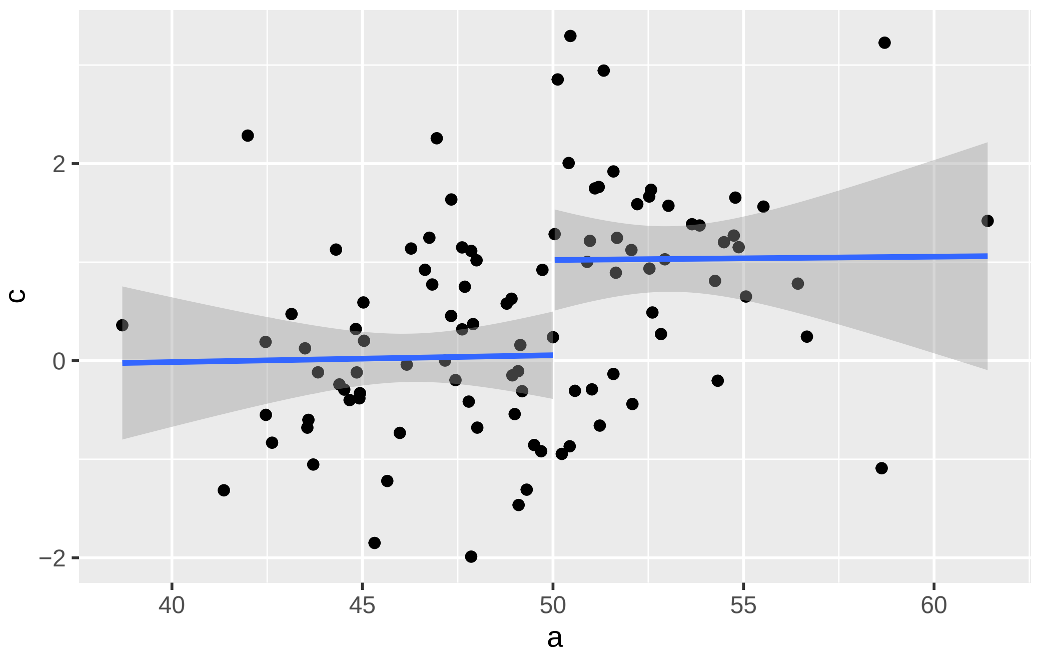
Characteristic	Beta	95% CI	p-value
a	0.01	-0.07, 0.08	0.9
b	0.97	0.30, 1.6	0.005

Abbreviation: CI = Confidence Interval

```
mod2.11<-lm(c~a+b)
mod2.11|>
  tbl_regression()
```

```
#grouping by b (mediator) disrupts the correlation by a nd c
ggplot(data=tibble(a,c),aes(x=a,y=c,group=factor(b)))+geom_point()+stat_smooth(method="lm")
```

```
`geom_smooth()` using formula = 'y ~ x'
```

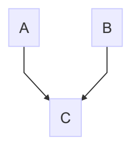


5 Why we should not condition on a collider

Let's start with a dag that illustrates a collider. The dag is A–>C and B–>C.To determine why we should not condition on a collider, we can compare a model where we do condition on C vs a model without conditioning on C. A and B are independent and should not correlate.

```
mermaid("graph TD
        A-->C
        B-->C")
```

```
file:///C:\Users\anbe6\AppData\Local\Temp\RtmpiW4ShR\file1e782cf822dc\widget1e7879276be.html
```


A
B
C

Characteristic	**Beta**	**95% CI**	**p-value**
a	0.07	-0.15, 0.28	0.6

Abbreviation: CI = Confidence Interval

Characteristic	**Beta**	**95% CI**	**p-value**
a	-0.52	-0.70, -0.34	<0.001
c	0.52	0.42, 0.61	<0.001

Abbreviation: CI = Confidence Interval

$$a \sim Normal(50, 5)$$

$$b \sim Normal(0, 5)$$

$$c \sim a + b + \epsilon$$

Random error: $\epsilon \sim Normal(0, 1)$

Note: It is expected that a and b are un-related

```
a<-rnorm(n=100,mean=50,sd=5)
b<-rnorm(n=100,mean=0,sd=5) #b is a function of a + random error
c<-a+b+rnorm(n=100,mean=0,sd=5) # c is a fucntion of a and b with random error

#fit a model between a-> b
mod3<-lm(b~a)
mod3|>
  tbl_regression()
```

```
#There is no association between A and B.

#fit a model with a collider c
mod4<-lm(b~a+c)
mod4|>
  tbl_regression()
```

```
#There is an association when conditioning on C.
```

Conditioning on a collider (c) introduces the causal effect of a–>b that is negative.

Characteristic	Beta	95% CI	p-value
a	1.1	0.86, 1.3	<0.001
b	1.1	0.89, 1.3	<0.001

Abbreviation: CI = Confidence Interval

Characteristic	Beta	95% CI	p-value
a	1.1	0.82, 1.5	<0.001

Abbreviation: CI = Confidence Interval

5.0.1 Side note: recovering effects of a and b on c

```
mod4.1<-lm(c~a+b)
mod4.1|>
  tbl_regression()
```

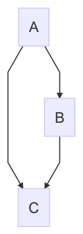
```
#now let's fit model with just one predictor alone

mod4.2<-lm(c~a)
mod4.2|>
  tbl_regression()
```

5.1 More complicated collider case

where:

```
mermaid("graph TD
        A-->B
        A-->C
        B-->C")
```

```
file:///C:\Users\anbe6\AppData\Local\Temp\RtmpiW4ShR\file1e78209a9c7\widget1e789805889.html
```


A
B
C

Characteristic	**Beta**	**95% CI**	**p-value**
a	1.0	0.84, 1.3	<0.001

Abbreviation: CI = Confidence Interval

$$a \sim Normal(50, 5)$$

$$b \sim a + \epsilon$$

$$c \sim a + b + \epsilon$$

Random error: $\epsilon \sim Normal(0, 5)$

Note: It is expected for a and b to have 1:1 relationship

```r
a<-rnorm(n=100,mean=50,sd=5)
b<-a+rnorm(n=100,mean=0,sd=5) #b is a function of a + random error
c<-a+b+rnorm(n=100,mean=0,sd=5) # c is a fucntion of a and b with random error

#fit a model between a-> b
#there is a 1:1 relationship
mod3.1<-lm(b~a)
mod3.1|>
  tbl_regression()
```

```r
#There is no association between A and B.

#fit a model with a collider c
mod4.1<-lm(b~a+c)
mod4.1|>
  tbl_regression()
```

Note that the association between a and b is negative when conditioning on c, the collider (above) and when we do it here when a and b are correlated, the correlation breaks.

Characteristic	**Beta**	**95% CI**	**p-value**
a	0.06	-0.19, 0.31	0.6
c	0.48	0.38, 0.58	<0.001

Abbreviation: CI = Confidence Interval

6 Why we should condition on a confounder

Simulate data and compare conditioning vs not on a confound.

$$b \sim Normal(5,5)$$

$$a \sim Normal(50,5) + b + \epsilon$$

$$c \sim a + b + \epsilon$$

Random error: $\epsilon \sim Normal(0,1)$

Note: It is expected for C to have a 1:1 effect from a.

```
b<-rnorm(n=100,mean=5,sd=5)
a<-rnorm(n=100,mean=50,sd=5)+b+rnorm(n=100,mean=0,sd=1)
c<-a+b+rnorm(n=100,mean=0,sd=1)

#without conditioning
mod5<-lm(c~a)
mod5|>
  tbl_regression()
```

```
#with conditioning
mod6<-lm(c~a+b)
mod6|>
  tbl_regression()
```

Conditioning on b recovers the 1:1 relationship from a on c.

Characteristic	Beta	95% CI	p-value
a	1.5	1.4, 1.6	<0.001

Abbreviation: CI = Confidence Interval

Characteristic	**Beta**	**95% CI**	**p-value**
a	1.0	0.97, 1.0	<0.001
b	0.98	0.93, 1.0	<0.001

Abbreviation: CI = Confidence Interval

7 Summary

Drawing out a DAG for a specific case is important to determine how to analyze the data. For many observational studies, conditioning on confounders is critical and selecting the whole set of confounders is also important. Generally, baseline covariates are the confounders that is typically conditioned upon. It takes clinical expertise to decide which covariates to include.

Part II

Misc Bayesian statistics

8 Lab1: Bayesian inference with beta priors, Jingchen Hu

9 Intro

This is a lab by a professor, Jingchen Hu, which goes over Bayesian inference with beta priors.

10 Load libraries

```
library(tidyverse)
library(ProbBayes)
```

11 Posterior predictive checking

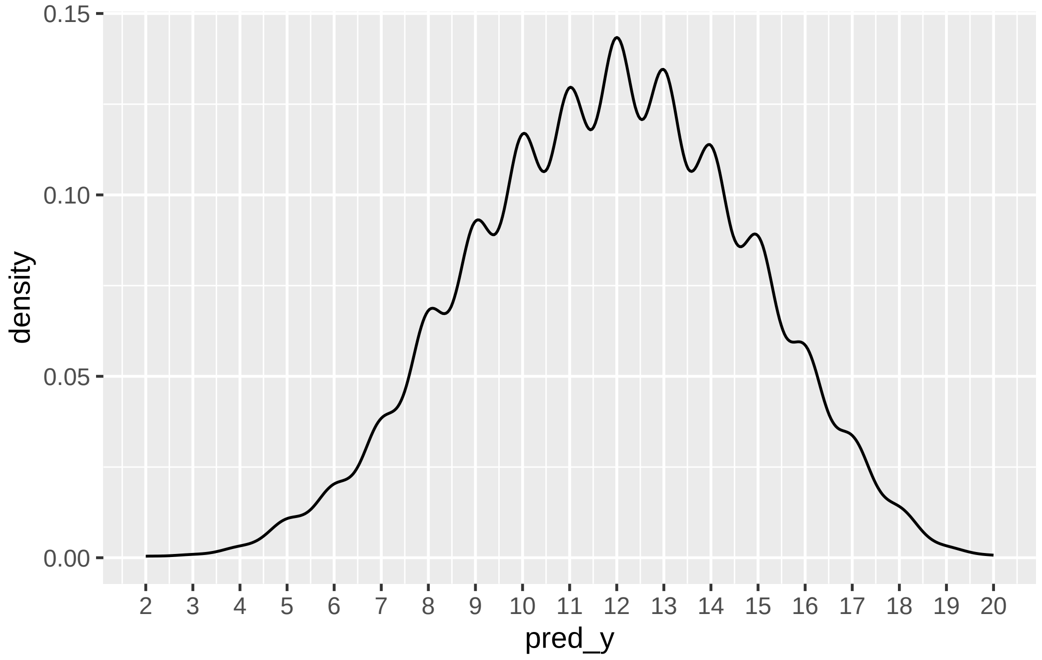
```r
S<-10000 # number of simulations
a<-3.06 # a in beta(a,b)
b<-2.56 # b in beta(a,b)
n<-20 # number of trials
y<-12 # number of successes

newy=as.data.frame(rep(NA,S))
names(newy)=c("y")

set.seed(123)
for (s in 1:S){
  pred_p_sim<-rbeta(1, a+y, b+n-y) # step 1 ; get posterior param
  pred_y_sim<-rbinom(1,n,pred_p_sim) # step 2; based on param, predict outcome-> # of succes
  newy[s,]=pred_y_sim
}
knitr::kable(head(newy))
```

y
14
13
8
12
14
5

```r
#how i would write the simluation

dat<-tibble(pred_p=rbeta(S,a+y,b+n-y))|>
  rowwise()|>
  mutate(pred_y=rbinom(1,n,pred_p))

sum(dat$pred_y>=5&dat$pred_y<=15)/S
```

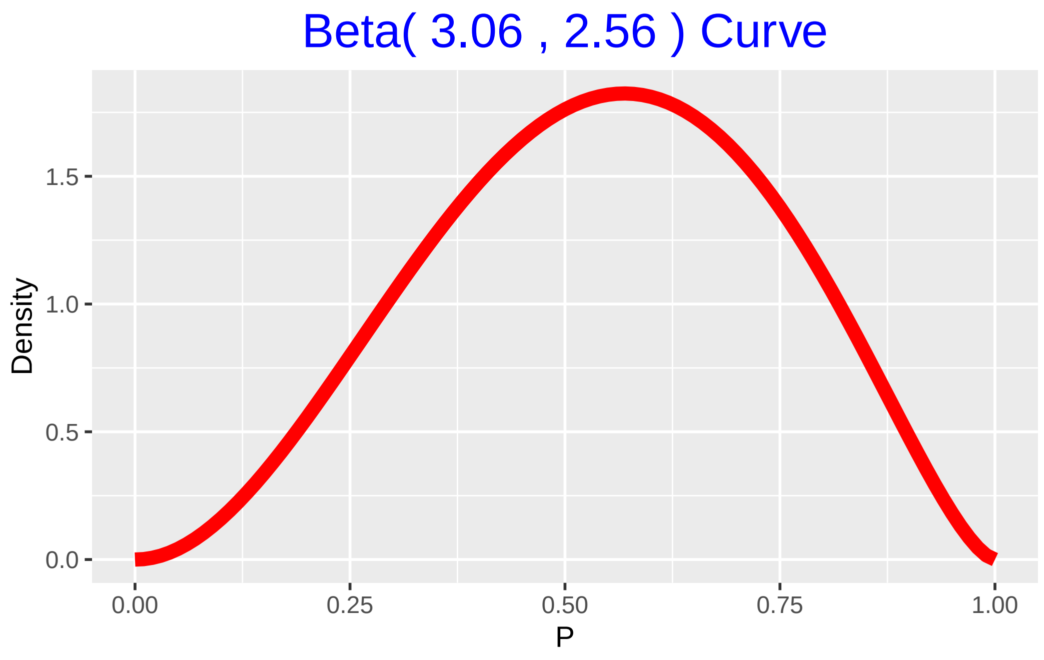
```
[1] 0.8943
```

```
#dat$pred_y<-rbinom(1000,n,dat$pred_p)
ggplot(data=dat,aes(pred_y))+geom_density()+scale_x_continuous(breaks=seq(0,20,1),labels=seq
```

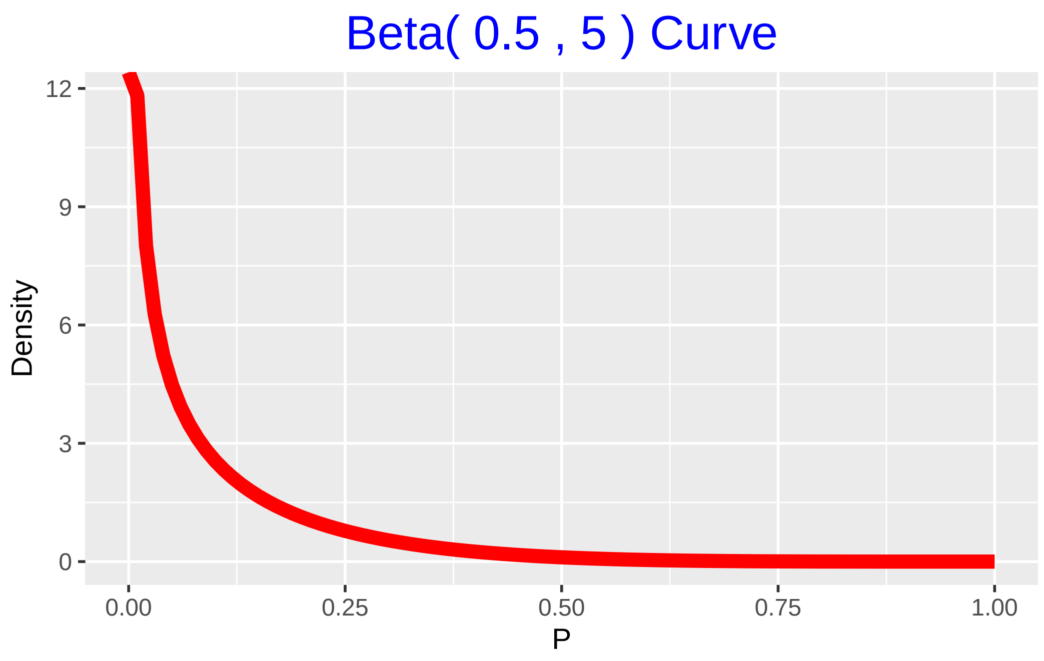


12 Let's try to simulate a situation with mismatched prior with the data

```
beta_draw(c(3.06,2.56)) #prior fromp revious section
```



```
beta_draw(c(0.5,5)) #this looks liek a good prior to mess up the data
```



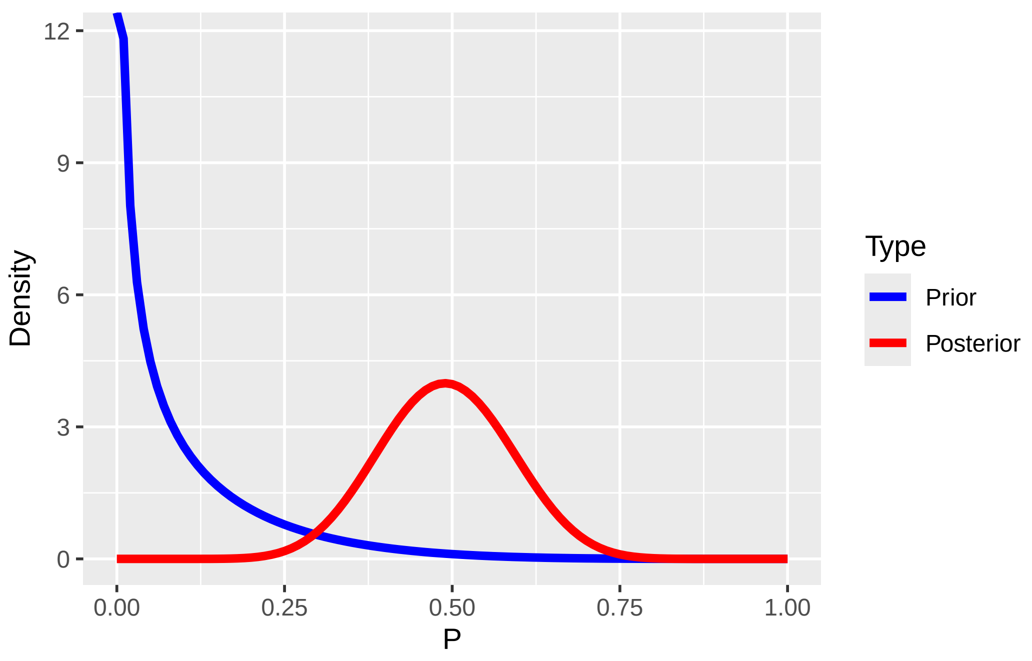
```
s<-10000
n<-20 # trials
y<-12 #successes
a<-.5
b<-5

dat2<-tibble(pred_p=rbeta(S,a+y,b+n-y))|>
  rowwise()|>
  mutate(pred_y=rbinom(1,n,pred_p))

# model check : how often pr(y > ypred|y)
sum(y>dat2$pred_y)/S # how often collected data above posterior prediction
```

```
[1] 0.7158
```

```
1-sum(y>dat2$pred_y)/S #how often collected data below posterior prediction
```

```
[1] 0.2842
```

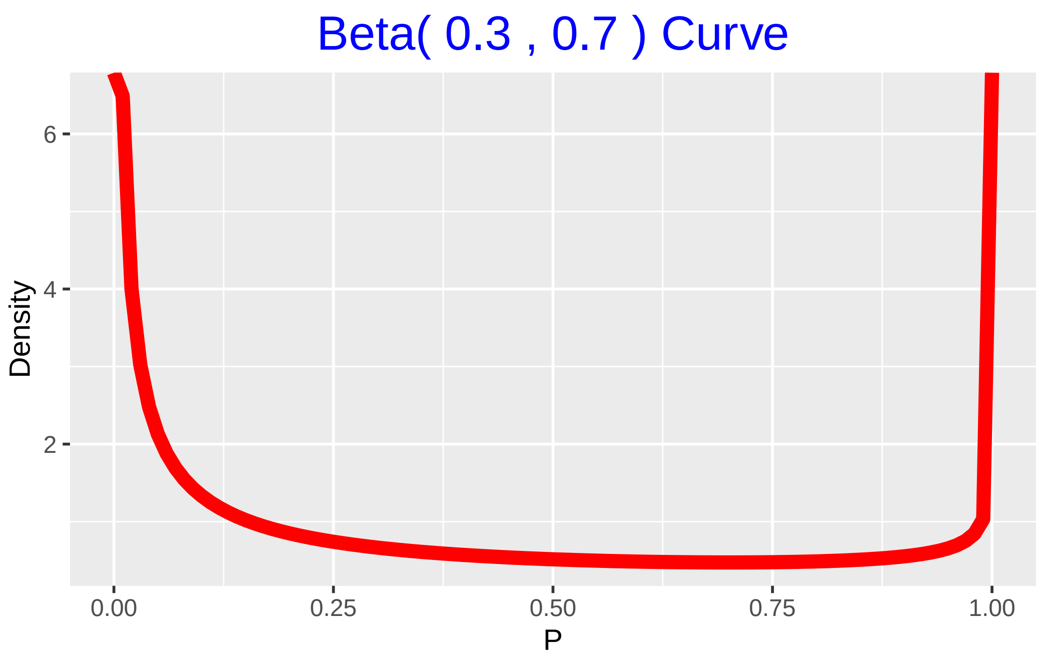
```
#draw posterior
beta_prior_post(c(.5,5),c(a+y,b+n-y))
```



12
9
6
3
0
Density
0.00
0.25
0.50
0.75
1.00
P
Type
Prior
Posterior

13 Session info

```r
beta_draw(c(.3,.7))
```



```r
sessionInfo()
```

```
R version 4.5.0 (2025-04-11 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 26100)

Matrix products: default
  LAPACK version 3.12.1

locale:
[1] LC_COLLATE=English_United States.utf8
```

```
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] ProbBayes_1.1     shiny_1.10.0      gridExtra_2.3     LearnBayes_2.15.1
 [5] lubridate_1.9.4   forcats_1.0.0     stringr_1.5.1     dplyr_1.1.4
 [9] purrr_1.0.4       readr_2.1.5       tidyr_1.3.1       tibble_3.2.1
[13] ggplot2_3.5.2     tidyverse_2.0.0

loaded via a namespace (and not attached):
 [1] generics_0.1.3    stringi_1.8.7     hms_1.1.3         digest_0.6.37
 [5] magrittr_2.0.3    evaluate_1.0.3    grid_4.5.0        timechange_0.3.0
 [9] fastmap_1.2.0     jsonlite_2.0.0    tinytex_0.57      promises_1.3.2
[13] scales_1.3.0      cli_3.6.4         rlang_1.1.6       munsell_0.5.1
[17] withr_3.0.2       yaml_2.3.10       tools_4.5.0       tzdb_0.5.0
[21] colorspace_2.1-1  httpuv_1.6.16     vctrs_0.6.5       R6_2.6.1
[25] mime_0.13         lifecycle_1.0.4   pkgconfig_2.0.3   pillar_1.10.2
[29] later_1.4.2       gtable_0.3.6      glue_1.8.0        Rcpp_1.0.14
[33] xfun_0.52         tidyselect_1.2.1  knitr_1.50        farver_2.1.2
[37] xtable_1.8-4      htmltools_0.5.8.1 rmarkdown_2.29    labeling_0.4.3
[41] compiler_4.5.0
```

Part III

Misc Frequentist statistics

14 Simulating 95% confidence intervals

15 Load Libraries

```r
library(tidyverse)
```

16 Simulating 95% frequentist confidence intervals

A good explanation here:

A 95% confidence interval is constructed such that if the model assumptions are correct and if you were to hypothetically repeat the experiment or sampling many many times, 95% of the intervals constructed would contain the true value of the parameter.

My own words: **The 95% confidence interval is when the true parameter is contained within the interval 95% of the time from constructing the 95% confidence interval from repeated experiments under the assumption of a correct model.**

Let's gain intuition by what this means:

1. Simulate data and do it a bunch of times

2. Then calculate 95% confidence interval with say a t-test
3. Determine how many times the true parameter (which we set in step 1) is in between the confidence intervals

```r
#1) simulate data
sim<-10000
#dataset size
n<-100
# sampel data with mean 10, sd =1
x<-rnorm(n,mean=10,sd=1)
#fit t.test ; grab lower and upper confidence interval
#as.vector(c(t.test(x)$conf.int,t.test(x)$estimate))


## now simulate across sim

#for loop is prob best
#prep dataset
#d<-tibble(lower=rep(0,sim),upper=rep(0,sim),mean=rep(0,sim))
```

```r
d<-array(0,dim=c(sim,2))

for (i in 1:sim){
  x<-rnorm(n,mean=10,sd=1)
  d[i,]<-as.vector(c(t.test(x)$conf.int))
}


#head(d)
d<-data.frame(d)
names(d)<-c("lower","upper")
knitr::kable(head(d))
```

lower	upper
9.890115	10.26806
9.622940	10.02791
9.901198	10.32905
9.848092	10.25550
9.631622	10.05999
9.677642	10.11814

```r
#count how many times the lower and upper confidence interval is below true value of 10
d<-d|>
  mutate(out=1*(lower<10 & upper>10))
mean(d$out)
```

```
[1] 0.948
```

```r
#cases where confidence interval is does not include true parameter
d|>
  filter(out==0)|>
  head()
```

```
      lower     upper out
1 10.003804 10.371640   0
2  9.601640  9.969193   0
3 10.055226 10.434564   0
4 10.029652 10.422496   0
5  9.553648  9.940772   0
6  9.627342  9.993598   0
```

Additional notes:

Cementing interpretation: When you have a single 95% CI on a single sample, it doesn't mean, that the population mean belongs to this particular interval with a particular probability. If you were to repeat the experiment many many times and calculate this interval on each fo the samples, then 95% of the repeated samples would have the true population mean.

17 Session info

```r
sessionInfo()
```

```
R version 4.5.0 (2025-04-11 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 26100)

Matrix products: default
  LAPACK version 3.12.1

locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] lubridate_1.9.4 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
 [5] purrr_1.0.4     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
 [9] ggplot2_3.5.2   tidyverse_2.0.0

loaded via a namespace (and not attached):
 [1] gtable_0.3.6      jsonlite_2.0.0    compiler_4.5.0    tidyselect_1.2.1
 [5] tinytex_0.57      scales_1.3.0      yaml_2.3.10       fastmap_1.2.0
 [9] R6_2.6.1          generics_0.1.3    knitr_1.50        munsell_0.5.1
[13] pillar_1.10.2     tzdb_0.5.0        rlang_1.1.6       stringi_1.8.7
[17] xfun_0.52         timechange_0.3.0  cli_3.6.4         withr_3.0.2
```

```
[21] magrittr_2.0.3    digest_0.6.37     grid_4.5.0        hms_1.1.3
[25] lifecycle_1.0.4   vctrs_0.6.5       evaluate_1.0.3    glue_1.8.0
[29] colorspace_2.1-1  rmarkdown_2.29    tools_4.5.0       pkgconfig_2.0.3
[33] htmltools_0.5.8.1
```

Part IV

Function-valued traits

References