On determining the post-test probability of covid-19

Sam Weisenthal (firstnamelastname@gmail.com)

Contents

Summary

Tests are crucial for controlling the covid-19 pandemic. Test result interpretation is also important. For example, a test could be negative, but covid-19 might still be present. After getting a negative rapid antigen test a few months ago, I tried to use Bayes rule, which is the standard approach, to calculate post-test probability. This was not easy. I recorded my efforts at https://discourse.datamethods.org/t/ determining-post-test-probability-of-covid-19/4723/12, and I have tried to collect my most updated thoughts

at https://github.com/samuelweisenthal/BinaxNOWpostTestProb/blob/main/covid.post.test.pdf. Difficulties included specifying my pretest probability, getting appropriate estimates for the test operating characteristics, and understanding my estimate in light of other available information, such as exposure and symptom status. Beware, as mentioned in Moons and Harrell, 2003, or Harrell's BBR textbook, there seems to be a trap: your pretest probability cannot depend on variables that are different from those collected in the study that evaluated the test. For variables x, this can be seen by examining

$$p(dz + |test-, x) = p(test - |dz+, x)p(dz + |x)/p(test - |x).$$

This trap is particularly problematic when you have reason to believe that certain variables, such as symptom status or exposure, make it more likely that you have the disease at baseline or when the study that estimates the operating characteristics conditions implicitly on covariates that are not easily reportable.

Overall, I found this to be a hard problem, based both on inherent difficulties and also on the indirect way in which I had to go about it. There may be ways to make the entire process more direct, shifting the burden of test interpretation from the person taking the test to statistical analysts who have appropriate tools and databases. For example, it might be best to maintain databases that could be used to fit models for post-test probability that are specific to location and time, as well as other important covariates. In this case, we would just need to go to a website and enter some information to get a test interpretation.

Background

Some months ago, I wanted to, given what I had available, try to determine my post-test probability of covid-19 after a negative Binax NOW rapid antigen test and a sustained, unmasked exposure. Originally, I hoped we could apply Bayes rule to some estimates from the literature. It turned out to be more complicated, and I have decided to write out my thoughts here.

Ideal test interpretation framework

This is based on Dr. Harrell's first response to my question https://discourse.datamethods.org/t/what-should-mds-in-training-know-about-medical-prediction/335/4.

The following would be ideal: every time one takes a test, one goes to a website and enters information such as age, sex, zip code, and test result. The website then calculates post-test probability based on a model for, explicitly,

$$p(dz + |test, age, sex, symptoms, location, time, onset, strain, ..).$$

The database and corresponding model would be updated in real-time based on geographically and temporally relevant statistics. Note that this post-test probability would depend very much on time, and therefore the model would have to be updated. To fit this model, we would need observational data to be collected when people report their test results (in other words, we collect information such as age, zip code, etc). We would also collect PCR confirmation. It is unclear however whether we will have enough of the variables mentioned above, which are still necessary, and therefore there would need to be ongoing studies analyzing test performance as a function of different covariates. This type of modeling does not occur usually because each model would be specific to time and location, at least. This sounds like a daunting task, therefore, to fit all of these models, but the reality is that if this is not done that way, it has to be done another way — it just shifts the responsibility of interpretation from the test manufacturer or some organization with appropriate expertise onto the person using the test.

We would also make sure to get as much information as possible. Ideally, the test result would be a continuous variable (eg, amount of viral load). This may be difficult due to at-home testing kit constraints. However, it seems that there is some cutoff above which a test is called positive. If it is possible for the tests to convey more information, such as through color or some type of numerical scale, it would lead to better estimates of the post-test probability (assuming there is no real hard cutoff - I am not sure).

To begin this document, I will consider the easier problem of estimating post-test probability given a negative test only, with no exposure. In other words, let us forget about the conditional post-test probability, $p(dz + |test-, exposure)$, and consider "$p(dz + |test-)$".

The difficulty of pre-test probability $p(dz+)$

Before doing this, however, let us consider an even simpler problem: $p(dz+)$. We will need to have some sense of this in order to estimate $p(dz + |test-)$ with the information that I have available (we would not need to estimate $p(dz+)$ if we had other data available). The quantity $P(dz+)$ is commonly called the "pretest" probability. Often, it is considered to be prevalence. However, this is a very nebulous estimand. We almost never think of a pure, unconditional probability (Moons and Harrell 2003). Also, if we want to use it to then estimate $p(dz + |test-)$, it must be somewhat constrained. I show the derivation below partly to show that $p(dz+)$ must align with $p(test - |dz+)$. Often, $p(test - |dz+)$ is conditional on things that are not mentioned explicitly. Hence, we need to estimate $p(dz+)$ conditional on covariates that were possibly not even reported in the study that evaluated the test. Hence, estimating $p(dz+)$ is quite difficult, and the estimate cannot be "what you expect your pretest probability is." It also cannot be your best guess at your probability of the disease given your current characterstics, such as symptoms, location, etc. This is a trap that is tempting to fall into. *Pretest probability cannot be estimated based on more or less information than was collected in the study that evaluated the test operating characteristics.*

In some ways, it may be best to stop here. However, I needed an answer at the time, and it is likely that many need an answer now. So, we will proceed with what we have. It is worth noting that the approach that I will take, using Bayes rule, is the standard approach in the literature.

Estimating $p(dz + |test-)$

Now that we have considered the problem of estimating $p(dz+)$, let us consider the problem of estimating $p(dz + |test-)$. We have, by Bayes rule,

$$
\begin{aligned}
p(dz + |test-) &= \frac{p(test - |dz+)}{p(test-)} p(dz+) \\
&= \frac{1 - p(test + |dz+)}{p(test - |dz-)p(dz-) + p(test - |dz+)p(dz+)} p(dz+) \\
&= \frac{1 - sens}{spec(1 - pretest) + (1 - sens)pretest} pretest.
\end{aligned}
$$

On renaming

If we can fill in the pieces of this equation, we can provide an estimate of the left hand side. Note that in the last line I rename the probabilities as sensitivity and specificity. This renaming, interestingly, can cause some difficulty; we understand the terms in the second to last line, but the renaming in the last line renames and strips the terms of their probabilistic meaning (along with removing information about what is being conditioned upon), which can lead to misuse, and may be related to the persistence of the framework for test interpreation that gives so many issues below (Moons and Harrell 2003).

On flexibility

(recent, rough thoughts:) I do wonder about the flexibility argument (https://discourse.datamethods.org/t/sensitivity-specificity-and-roc-curves-are-not-needed-for-good-medical-decision-making/1152/24). The idea is that sens/spec allow us to calculate post-test in different scenarios, where we don't have full models of post-test. It's like saying sens/spec allow us to calc post-test in population diff from derivation population, whereas PPV is tied to derivation population. This flexibility argument prompted me in large part to even pursue this problem with Bayes rule; it seemed that it was better to have a solution using the flexible procedure (based on sens/spec with Bayes rule), since not everyone (actually I don't think anyone) has access

to a fully trained model for estimating post-test probability of covid. Further, with infectious diseases, a more flexible procedure might be important; with diseases like heart disease, which have relatively constant prevalence over space and time, the direct approach might be better (ie, you can just have one model instead of a model for every location and time). However, even when it might be necessary (or even appropriate), the flexibility might come at great cost. The magnitude of this cost is still unclear to me, but the investigation below suggests that the cost might be high.

Estimates from (Bruemmer et al. 2021)

Here, I am going to use this equation to calculate the post-test probability given a negative test for a battery of rapid antigen tests. Consider numbers from the recent meta-analysis (Bruemmer et al. 2021). Note that these estimates, as far as I know, condition on nothing except the test result [and adult status? read more carefully] (I do not know how they would otherwise pool so many estimates). The validity of the estimates depend on the designs of the studies that were analyzed in the meta-analyses. Accoring to (Harrell 2020), a retrospective case-control is most appropriate. We assume that the studies used to obtain the sensitivities and specificities had a reasonable design, which is not always the case. Also, the studies generally use PCR as a gold standard, although PCR has its own imperfect operating characteristics. Nevertheless, it appears that (Bruemmer et al. 2021) is the most comprehensive study available.

Implementation

I am going to show my code to promote transparency and reproducibility. I have decided finally not however to show the plots in this document. I am doing this because, although it was a ton of work to make these plots, I am not sure actually that the world will be better off with them than it would be without them. I am not sure if they are usable due to the "trap" mentioned above. If you want to unhide the plots, feel free to do so. You can find the markdown file in my GitHub at https://github.com/samuelweisenthal/BinaxNOWpostTestProb. You are free to use and modify this code as long as you allow others to use and modify your modification of this code. Use this code at your own risk, and I will not be responsible for maintenance.

To implement, we start with the likelihood ratio.

```r
like.rat=function(sens,spec,pretest){(1-sens)/((1-sens)*pretest+(spec)*(1-pretest))}
```

We then get that the post-test probability of disease given a negative test is likelihood ratio times the pretest probability.

```r
post.test.prob = function(sens,spec,pretest){
  like.rat(sens=sens,spec=spec,pretest=pretest)*pretest
}
```

Let us compute the post-test probability over a grid of pre-test probabilities.

```r
pre.tests = seq(0,1,0.005)
get.post.tests = function(sens,spec,pre.tests){
post.tests = c()
for(i in 1:length(pre.tests)){
  post.tests[i]=post.test.prob(sens=sens,spec=spec,pretest=pre.tests[i])
}
post.tests
}
```

Now create a function to make plots.

```r
plot.p = function(lns,post.label,xlab,
                  leg.labels,main){

lwd.point = 2
plot(c(0,0),pch=0,xlim=c(0,1),ylim=c(0,1),
```

```r
      xlab=xlab,ylab=post.label,
      lty=1,lwd=lwd.point,main=main,col='white')

xseq = seq(0,1,0.1)
yseq = seq(0,1,0.1)
grid.col=rgb(0,0,0,alpha=0.1)
for(x in xseq){
  abline(v=x,col=grid.col)
}

for(y in yseq){
  abline(h=y,col=grid.col)
}
colorBlindBlack8  <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                       "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
nl = length(lns)
for (i in 1:nl){
lines(pre.tests,lns[[i]]$m,lty=1,lwd=lwd.point,col=colorBlindBlack8[i])
lines(pre.tests,lns[[i]]$u,lty=2,col=colorBlindBlack8[i])
lines(pre.tests,lns[[i]]$l,lty=2,col=colorBlindBlack8[i])

#rgbval = paste(as.vector(col2rgb(colorBlindBlack8[i])), collapse = " ")
#lines(pre.tests,lns[[i]]$u,lty=2,col=rgb(rgbval[1],rgbval[2],rgbval[3]))
#lines(pre.tests,lns[[i]]$l,lty=2,col=rgb(rgbval[1],rgbval[2],rgbval[3]))

}


legend("topleft",legend=leg.labels,bg='white',
       lty=rep(1,nl),col=colorBlindBlack8[c(1:nl)],lwd=c(lwd.point,lwd.point))
}
```

Here I have converted the table from (Bruemmer et al. 2021) into a list.

```r
tests=list(
veritor=list(
  nm='BD Veritor',
  sens=list(m=.64,l=.49,u=.76),
  spec=list(m=1,l=.99,u=1)
),
Binax=list(
  nm='BinaxNOW',
  sens=list(m=.62,l=.48,u=.74),
  spec=list(m=1,l=1,u=1)
),
Clinitest = list(
  nm='CLINITEST',
  sens=list(m=.62,l=.47,u=.75),
  spec=list(m=.99,l=.97,u=.99)
),
Coris = list(
  nm="Coris",
  sens=list(m=.40,l=.29,u=.52),
  spec=list(m=.99,l=.95,u=1)
```

```
),
LumipulseG = list(
  nm="Lumipulse G",
  sens=list(m=.87,l=.78,u=.93),
  spec=list(m=.97,l=.89,u=.99)
),
LumiraDx = list(
  nm="LumiraDx",
  sens=list(m=.88,l=.59,u=.98),
  spec=list(m=.99,l=.96,u=1)
),
Panbio = list(
  nm="Panbio",
  sens=list(m=.72,l=.65,u=.78),
  spec=list(m=.99,l=.99,u=1)
),
Rapigen = list(
  nm="Rapigen",
  sens=list(m=.62,l=.47,u=.75),
  spec=list(m=.99,l=.94,u=1)
),
Sofia = list(
  nm="Sofia",
  sens=list(m=.77,l=.74,u=.80),
  spec=list(m=.99,l=.98,u=1)
),
StandardF = list(
  nm="StandardF",
  sens=list(m=.68,l=.56,u=.79),
  spec=list(m=.98,l=.97,u=.99)
),
StandardQ = list(
  nm="StandardQ",
  sens=list(m=.75,l=.69,u=.80),
  spec=list(m=.99,l=.98,u=1)
),
StandardQnasal = list(
  nm="StandardQnasal",
  sens=list(m=.80,l=.70,u=.87),
  spec=list(m=.99,l=.98,u=1)
)
)
```

Confidence bounds (work in progress)

We want to take into account the uncertainty in the estimates above. We need to choose which values of sensitivity and specificity will be used for the confidence bounds. I am pretty sure I am correct about sensitivity here. I am not sure about specificity, but there is very little variation in specificity over the tests anyway. Recall from above that

$$p(dz + |test-) = \frac{1 - sens}{spec(1 - pretest) + (1 - sens)pretest}pretest$$

Note that this implies that sensitivity of 1 gives 0 post-test and sensitivity of 0 gives $\frac{pretest}{spec(1-pretest)}$. So higher sensitivity at fixed specificity will decrease post-test probability of the disease being present.

Fixing sensitivity, $spec = 0$ implies $\frac{1-sens}{sens}$ and $spec = 1$ implies $\frac{(1-sens)}{(1-pretest)+(sens)(pretest)}prestest$. This is not as clear, but I guess I will assume that higher specificity will decrease post-test probability since it is in the denominator in the original expression.

Hence, our lower bound on post-test probability will use the lower confidence bound of sensitivity and the lower confidence bound of specificity (and vice versa for the upper bound on post-test probability).

```r
get.t.posts = function(tests){
test.posts=list()
for(t in tests){
  test.posts[[t$nm]]=list(
    m=get.post.tests(sens=t$sens$m,spec=t$spec$m,pre.tests),
    l=get.post.tests(sens=t$sens$l,spec=t$spec$l,pre.tests),
    u=get.post.tests(sens=t$sens$u,spec=t$spec$u,pre.tests)
    )

}
test.posts
}
```

Difficulties

We will plot now the post test probabilities for each test as a function of the pretest probability and the operating characteristics above. Note that these plots may be falsely comforting. First, how do we specify the pretest probability? It's not necessarily prevalence, because prevalence is conditional upon location and time, which are conditional upon the geographic and temporal characteristics of the population. However, the sensitivity and specificity estimates likely also condition on the population characteristics, but it is not clear how they do so. Second, as I will discuss below, assuming I can specify a reasonable pretest probability, what if there are things about me that are important to take into consideration? I discuss exposure below, but how about symptom status, variant, time since symptom onset, etc? *You cannot simply change your pretest accordingly; sensitivity and specificity must also be updated.* To see this, look back at the derivation above, which shows that we must condition on the same variables for sensitivity/specificity and for the pre-test probability.
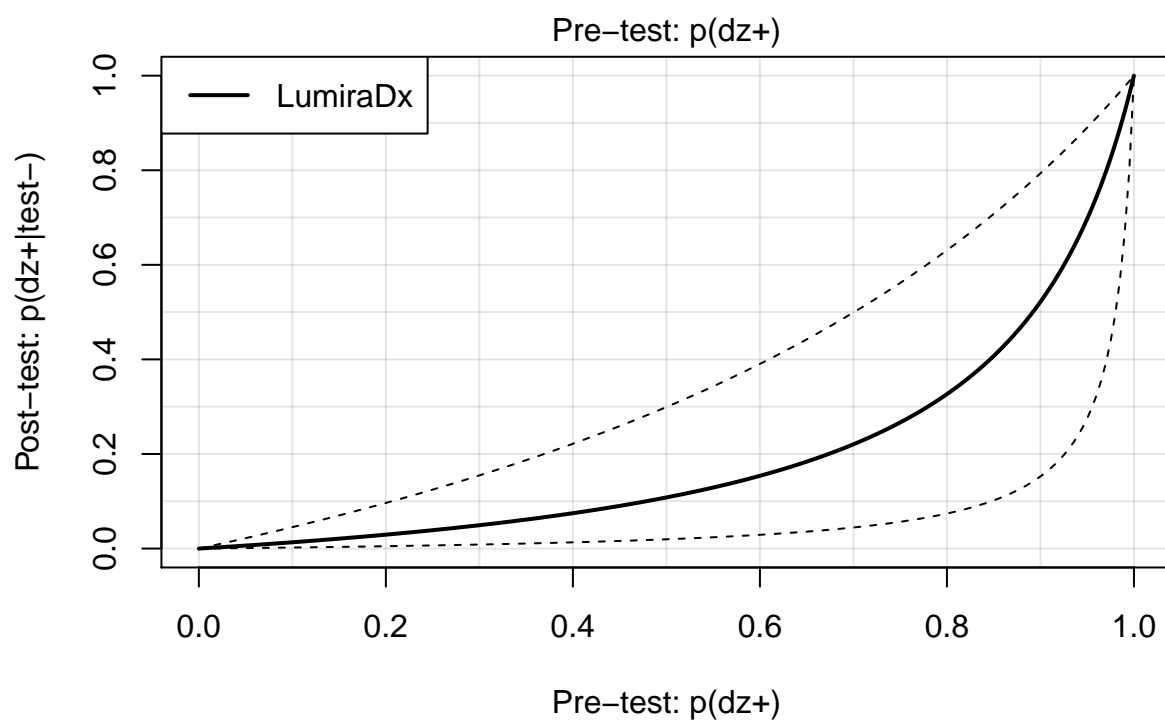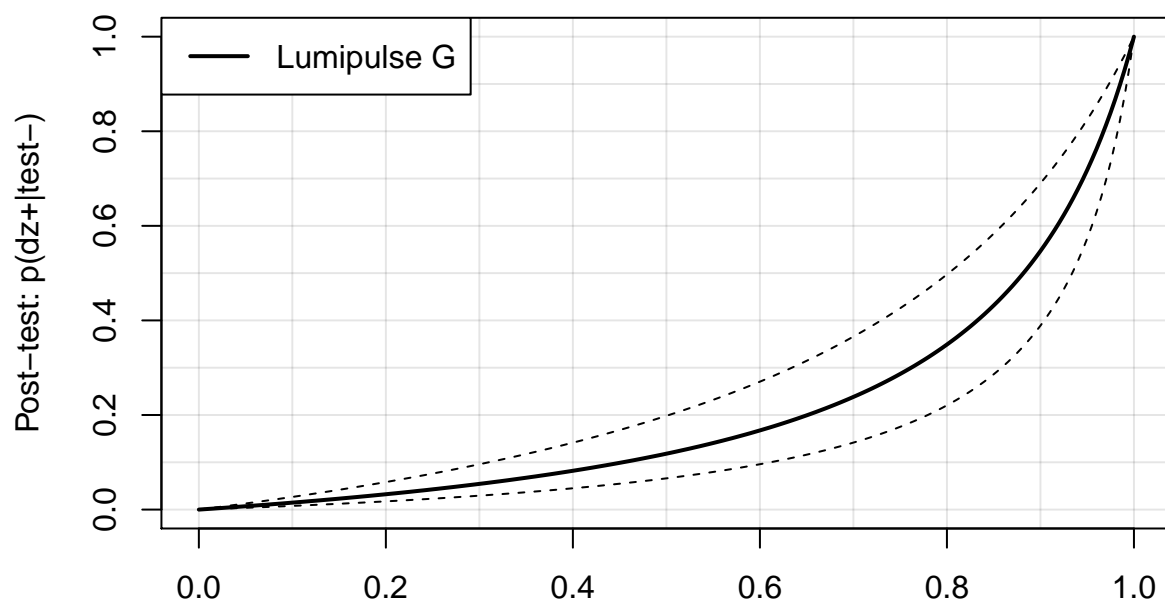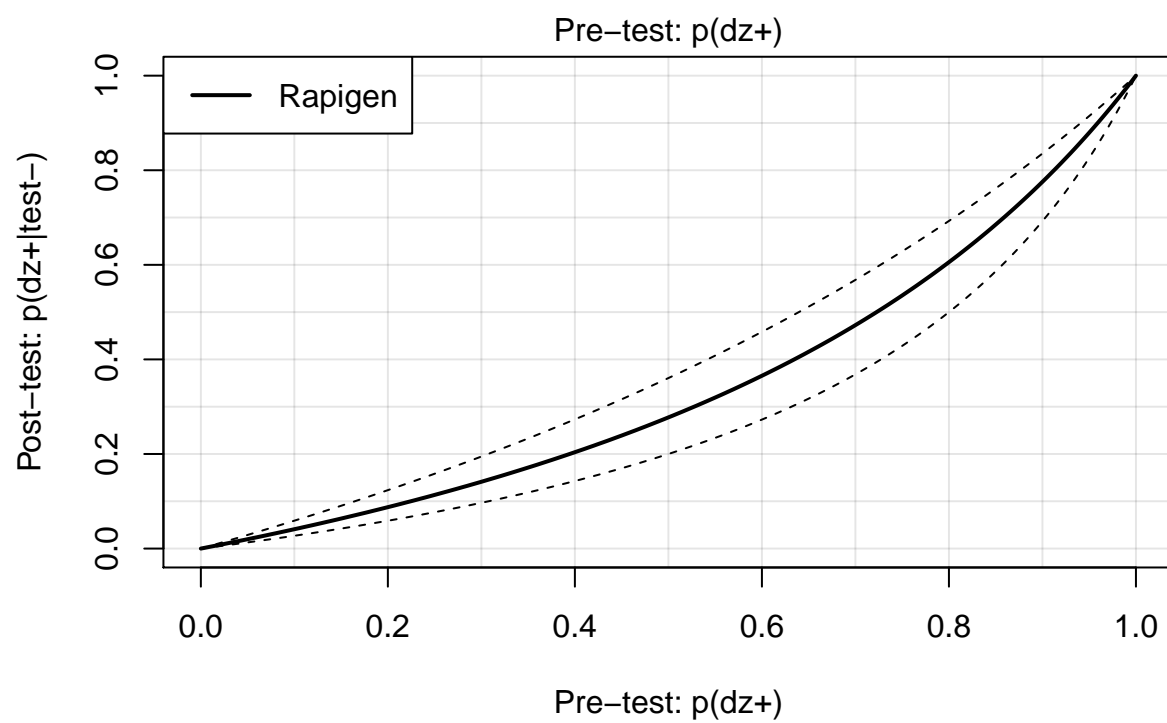
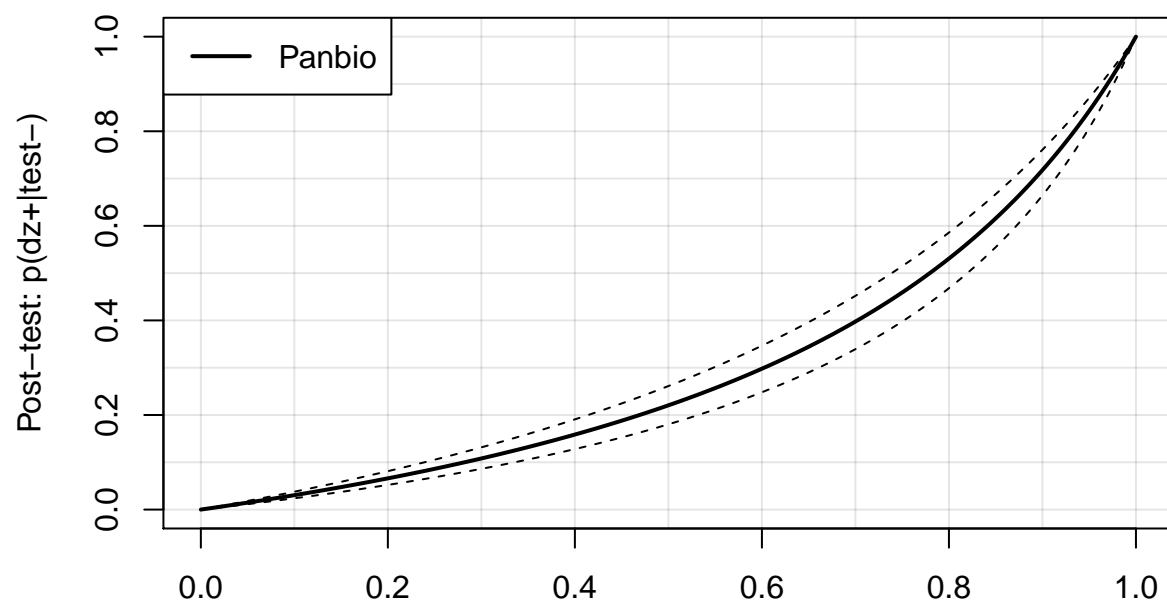## Plots using (Bruemmer et al. 2021)

```r
#```{r , fig.show='hide'}

#par(mfrow=c(12,1))
for (k in 1:12){
test.posts = get.t.posts(tests[k])
plot.p(test.posts,
       post.label="Post-test: p(dz+|test-)",xlab="Pre-test: p(dz+)",
       leg.labels=names(test.posts),main=""
       )
}
```
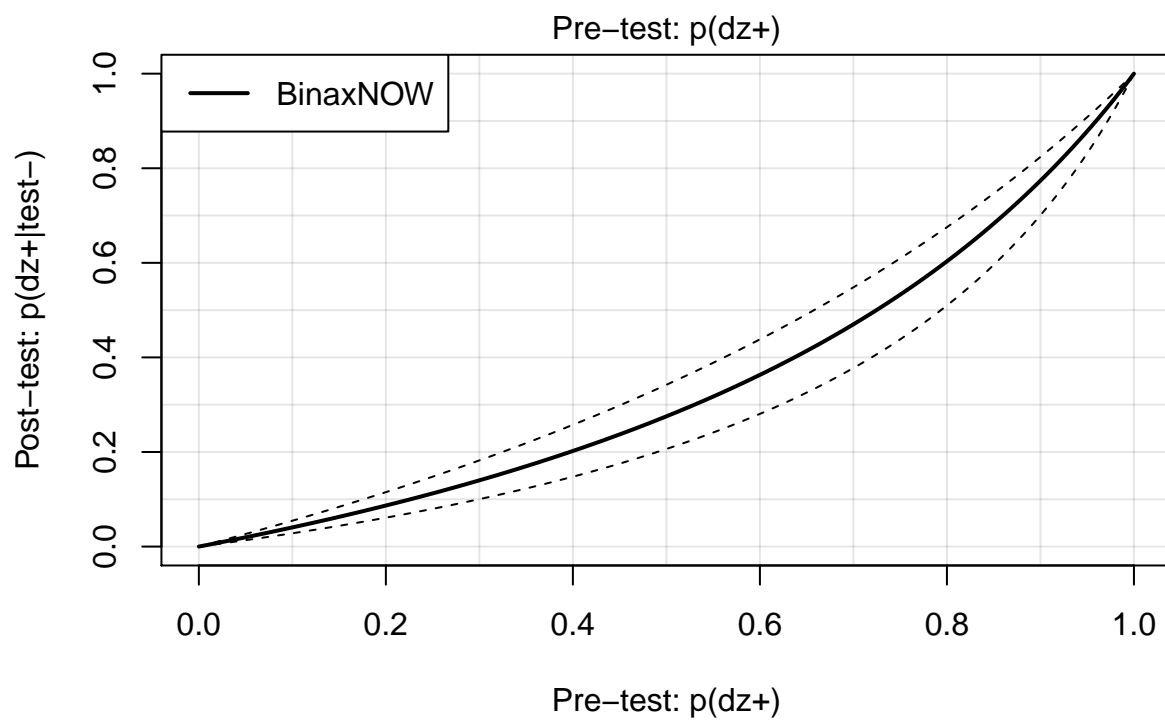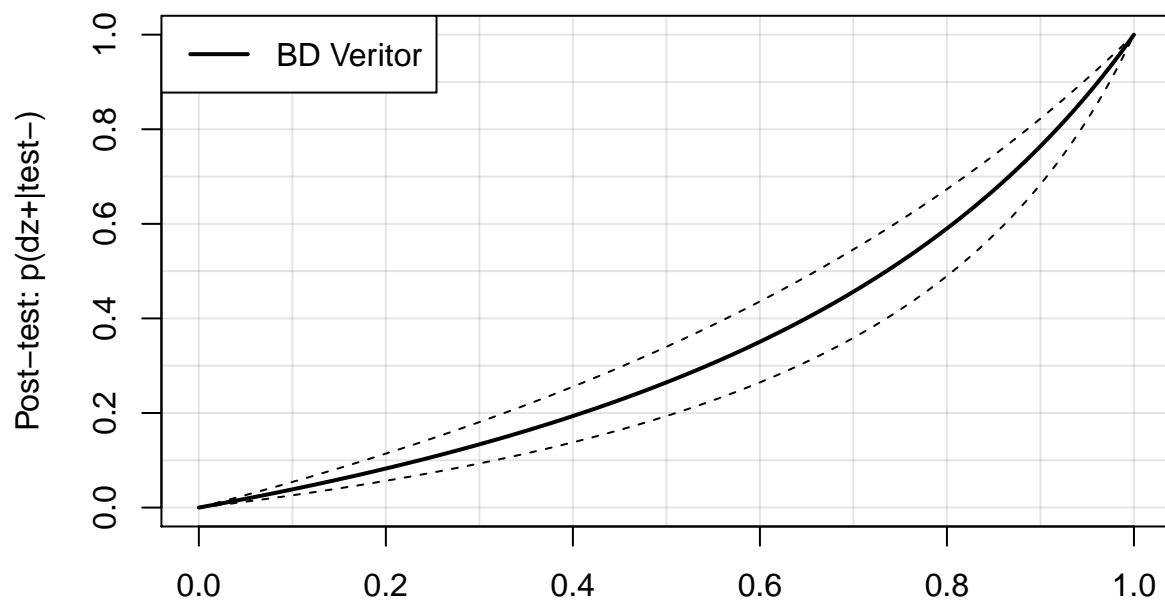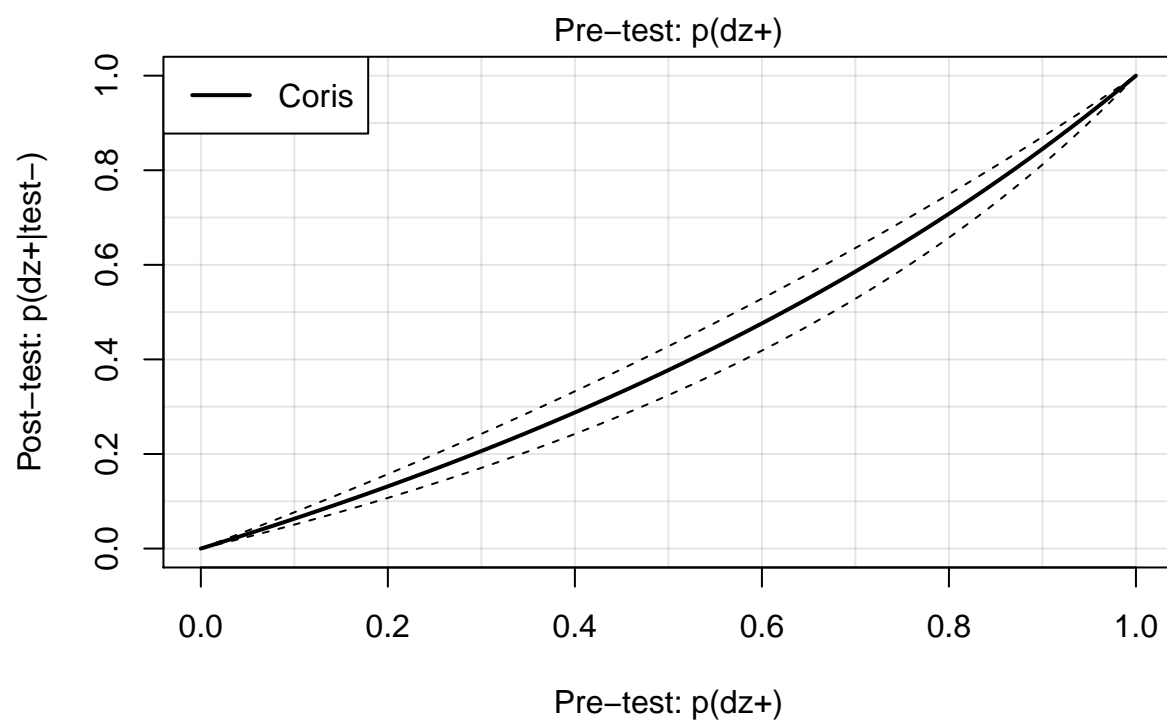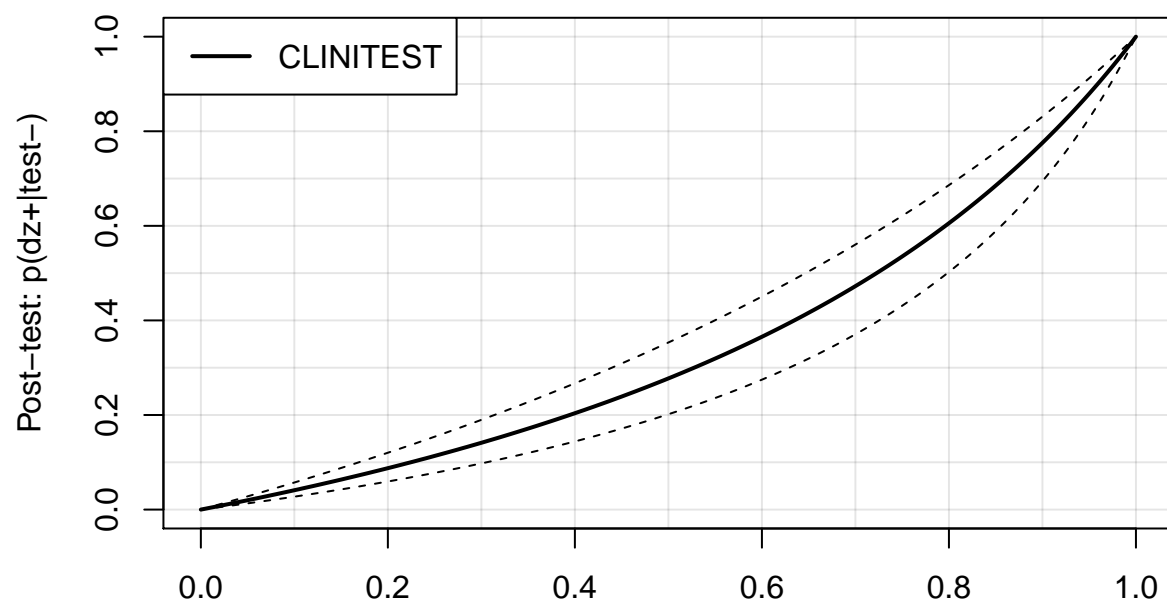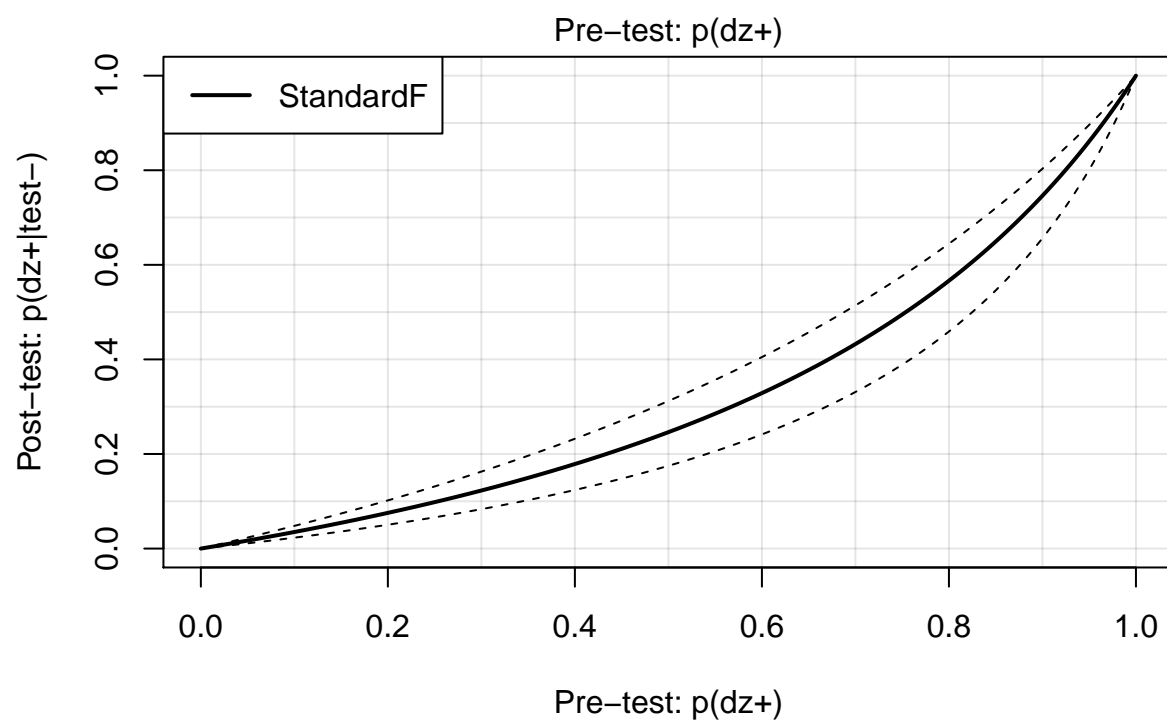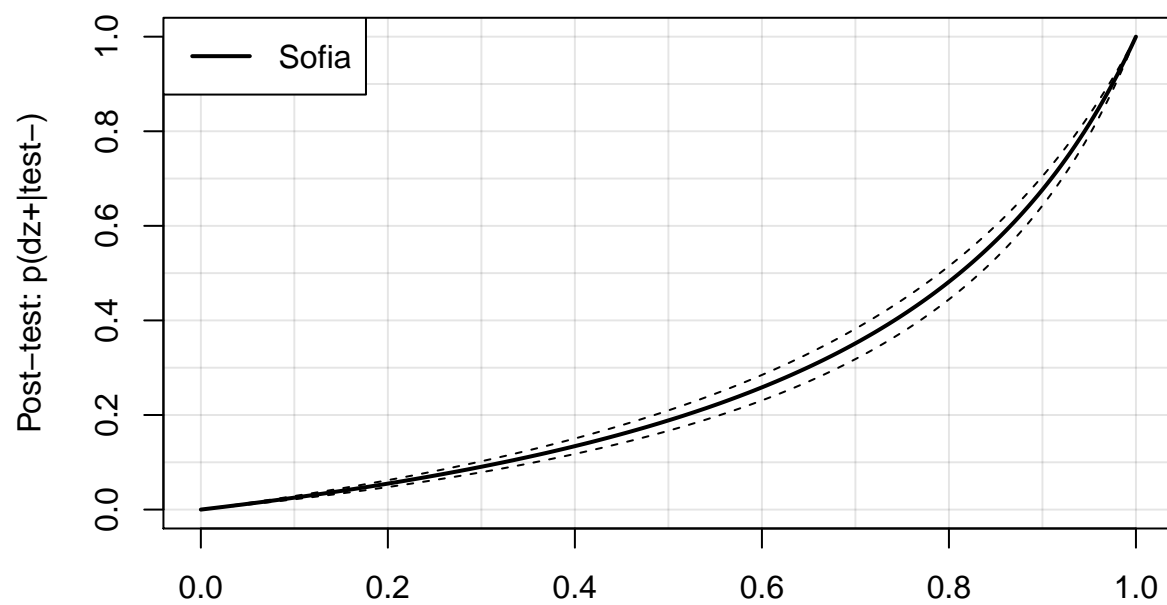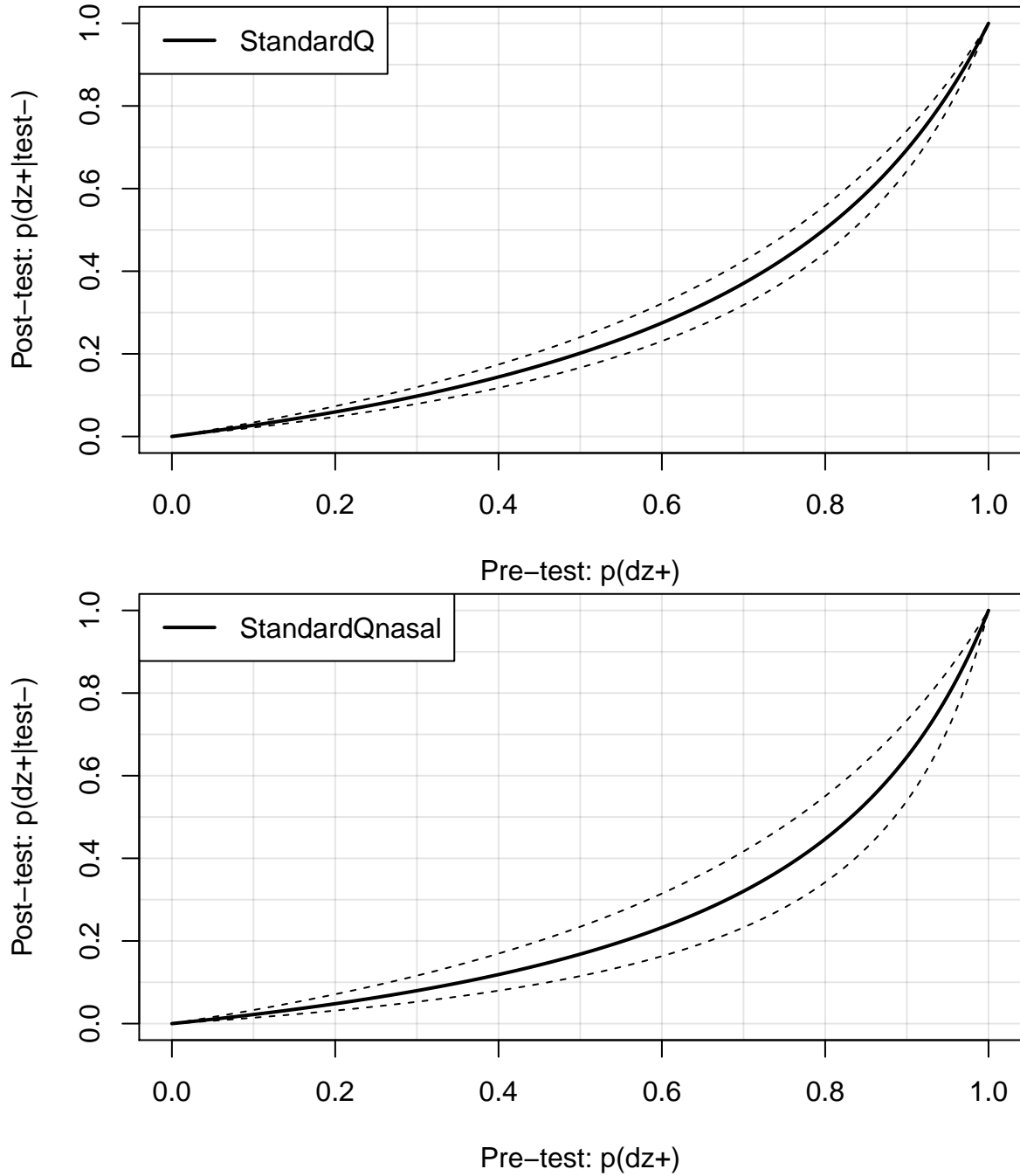
Estimating $p(dz + |test-, covariates)$ (work in progress)

Let us consider the general problem of conditioning on other covariates. This is still a work in progress. Note that I was originally interested in estimating probability of disease given a negative test and an expsoure, but that the post-test estimates above do not take into account the extra variable that we call "exposure." We have therefore the issue where we do not know the operating characteristics conditional on information that we have in hand. This is a more general problem. Exposure is a complicated variable and not readily collected. Before thinking about exposure, let us consider a different variable: symptoms.. This will give us insight into what happens when we ignore covariates that are available to us. I am including symptom status and "adult" as done in a study that measures these things, (Prince-Guerra et al. 2021), but this is more for

the sake of illustration than to give an accurate estimate.

Note:

$$p(dz + |test-, sx) = \frac{p(test - |dz+, sx)}{p(test - |sx)} p(dz + |sx)$$

$$= \frac{1 - p(test + |dz+, sx)}{p(test - |dz-, sx)p(dz - |sx) + p(test - |dz+, sx)p(dz + |sx)} p(dz + |sx)$$

$$= \frac{1 - sens}{spec(1 - pretest) + (1 - sens)pretest} pretest$$

Aside: will there be issues with the observational nature of the data? note that, just focusing on $p(dz+|test, sx)$, we are really interested in the test effect. Let us therefore consider how this would be estimated in general using the direct, ideal method. First, if the data is observational and there is an unmeasured variable that affects whether someone takes a test and also whether they have the disease (confounding), the effect estimate may be incorrect. Generally, we would condition on the confounder to remove this issue. I am still trying to think more about how this relates to the discussion on conditioning on the necessary set of variables. I think it is different, since we can defend $p(dz + |test)$, if it is tailored enough to our covariates to inform decision-making, but we can't really defend an unadjusted estimate of, e.g., a treatment effect. This actually seems to be the tension here. In some cases, $p(dz + |test-)$, $p(dz + |test-)$, is good enough, even though ideally we would have conditional $p(dz + |test-, age, sex, location, time, incubation, \dots)$. Especially, $p(dz + |test-)$ is good enough if somehow the variables we do not condition on in the $p(dz + |test-)$ estimate sort of cancel each other out. I think therefore that this is really mostly just about whether we want a more tailored estimate or not, there is not a causal question here. I would be interested to hear other perspectives.

Estimates from (Prince-Guerra et al. 2021):

Assuming that the study design of (Prince-Guerra et al. 2021) allows for valid estimation, we can use their point estimates and 95% confidence intervals for

$$p(test + |dz+, sx+, ad+) \approx 0.64 \ (0.57, 0.71)$$

and

$$p(test + |dz+, sx-, ad+) \approx 0.36 \ (0.27, 0.45).$$

Also,

$$p(dz - |test-, sx-, ad+) \approx 1(.99, 1)$$

and

$$p(dz - |test-, sx+, ad+) \approx 1(1, 1).$$

We can then plug these estimates into the equation above. Consider now what happens in the situation in which we did not measure symptom status:

$$p(dz + |test-) = (dz + |test-, sx+)p(sx + |test-) + p(dz + |test-, sx-)p(sx - |test-).$$

If we blindly compute the left hand side above, we will be estimating our post test probability ignoring symptoms, and therefore underestimate our risk. With the equation above, this is very clear that this should be the case. However, the mechanical use of Bayes rule to estimate post-test probability often leads us to ignore this fact, surprisingly. The same would be true for ignoring exposure; it may be the case that exposure is even more important than symptoms. This may also not be the case. Maybe, knowing that symptoms is more important than exposure, we can safely ignore exposure. Only gathering the appropriate data can resolve this. Possibly if the variables we do not condition on have equally opposing effect, we may be able to ignore them. We really though have to gather data to see if this is the case.

Other variables that might be relevant:

As I said above, we can only know how necessary it is to condition on additional covariates by collecting more data. However, one can also use previous studies to detect variables that might be relevant, but are ignored. I am concerned with the omission of gender, which appears to be possibly correlated with viral load (Mahallawi et al. 2021), assuming viral load and antigen presence are essentially the same. I think age should be treated as a continuous variable. It seems that age should correlate with viral load; this was not supported by (Mahallawi et al. 2021), but it might be supported by (Euser et al. 2021).

We need to condition on anything that leads to different antigen levels in different people. If the antigens are excreted or metabolized, we would need to take into account liver and kidney status. Of course, this should depend on immune system function, which will vary along with comorbidities and medications. Antigen level also probably depends on the covid strain. It appears that Table 2 in (Prince-Guerra et al. 2021) provides some interesting information that can help us here (there are counts of eg false positive and negative rates conditioned on sex, ethnicity, etc; unfortunately not jointly - I will have to look for the raw data). I am just eyeballing it, and it appears that the percentage of FN differs by sex, ethnicity, race, age, symptoms, days from symptom onset, and known exposure (!!).

As I mentioned before, the pretest probability is complex. It becomes even more complex when we wish to estimate it conditional on a lot of background variables. I am also unsure how one should estimate $p(dz+|sx)$. We can get this from the (Prince-Guerra et al. 2021) cohort, but this depends on things like location, time of year, and lockdown status, now in interaction with symptoms. Some of these have changed a lot since the study was conducted. One can avoid having to think about the latter if things are done with updated, location-specific databases, as mentioned above. It really should not be the responsibility of the test taker to think about the pretest probability, especially because it depends on the covariates chosen in the study that evaluates the test. It is unreasonable to expect that someone can guess, without data, the value of their conditional pre-test probability.

Comparing $p(dz+|test-, sx)$ from (Prince-Guerra et al. 2021) with $p(dz+|test-)$ (Bruemmer et al. 2021)

We can compare $p(dz+|test-, sx, adult)$ from (Prince-Guerra et al. 2021) with $p(dz|test-)$ from (Bruemmer et al. 2021). First, let us record the data from (Prince-Guerra et al. 2021).

```r
post.tests.6=get.post.tests(sens=0.64,spec=1,pre.tests)
post.tests.6l=get.post.tests(sens=0.57,spec=1,pre.tests)
post.tests.6u=get.post.tests(sens=0.71,spec=1,pre.tests)

l.6 = list(m=post.tests.6,l=post.tests.6l,u=post.tests.6u)

post.tests.36=get.post.tests(sens=0.36,spec=1,pre.tests)
post.tests.36u = get.post.tests(sens=.45,spec=.99,pre.tests)
#post.tests.36l = get.post.tests(sens=.27, spec=1,pre.tests)
post.tests.36l = get.post.tests(sens=.27, spec=1,pre.tests)

l.36 = list(m=post.tests.36,l=post.tests.36l,u=post.tests.36u)

# Prince-Guerra 2021

lns = list(l.6,l.36)
```
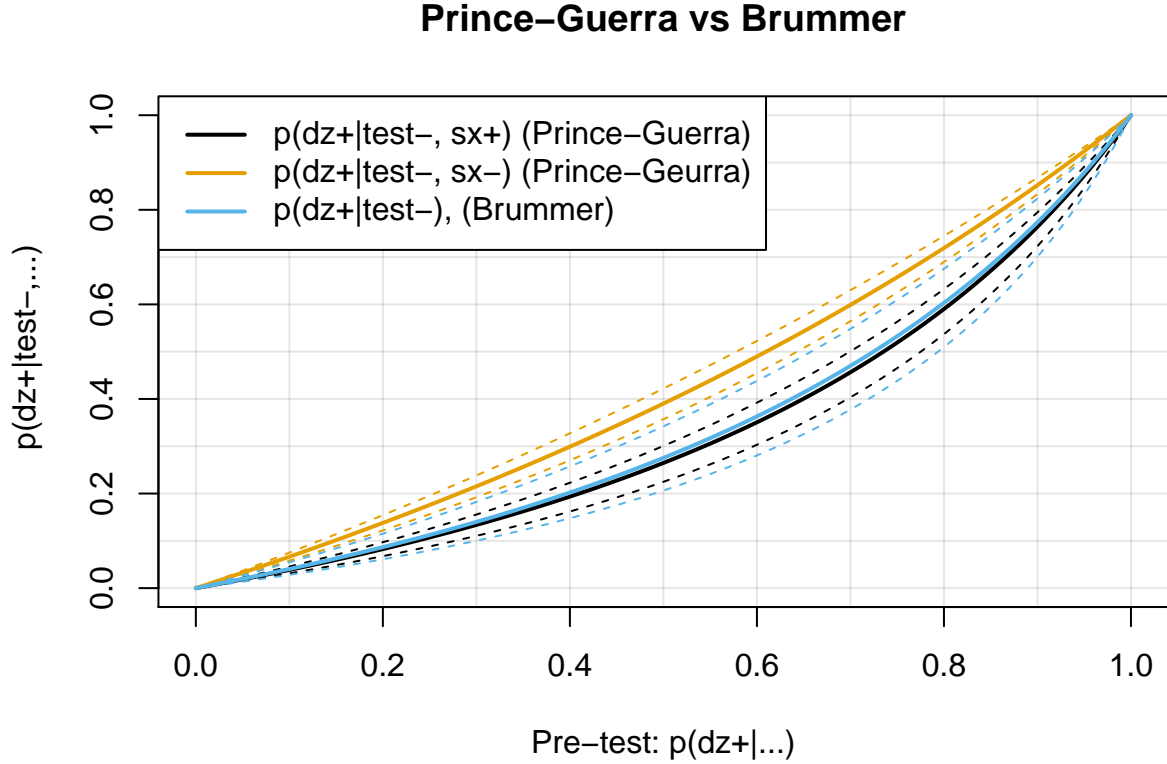
Now, let us plot.

```r
#```{r ,fig.show='hide'}

pts=get.t.posts(tests)
#pts$BinaxNOW
```

```
lns = list(1.6,1.36,pts$BinaxNOW)


plot.p(lns,
       post.label=" p(dz+|test-,...)",xlab="Pre-test: p(dz+|...)",
       leg.labels=c(TeX("p(dz+|test-,sx+) (Prince-Guerra)"),
                    TeX("p(dz+|test-,sx-) (Prince-Geurra)"),"p(dz+|test-), (Brummer)"),
       main="Prince-Guerra vs Brummer"
       )
```


Prince–Guerra vs Brummer

On which bound $p(dz+|test-)$ should favor

We expected that the $p(dz+|test-)$ would be between the two conditionals; going back to α above, which combines the two conditional distributions, we see.

$$p(dz+|test-) = p(dz+|test-,sx+)p(sx+|test-) + p(dz+|test-,sx-)p(sx-|test-)$$
$$= p(dz+|test-,sx+)p(sx+|test-) + p(dz+|test-,sx-)(1 - p(sx+|test-))$$
$$= p(dz+|test-,sx+)\alpha + p(dz+|test-,sx-)(1 - \alpha),$$

where $\alpha \in [0, 1]$.

Hence, the difference between the $p(dz+|test-)$ estimate and your conditional estimate will vary depending on your covariates. In other words, for someone who is symptomatic, it might not be a big deal to use the $p(dz+|test-)$ estimate. For someone who is asymptomatic, this might not be the case. I am fascinated that the $p(dz+|test-)$ seems to favor the symptomatic curve; I would have expected the opposite. I am going to explore this a bit more with a simulation.

We will make our variables $Z \sim Bern(expit(\beta X))$.

```
expit=function(x){exp(x)/(1+exp(x))}
#plot(-10:10,expit(-10:10))
```

Set the seed and sample size.

```
set.seed(1)
n=50000
```

Set prevalence, pdz.

```
pdz=.2
```

Let $dz \sim Bern(pdz), sx \sim Bern(expit(-2 + (2 + 2)dz))$, so we generally don't have symptoms unless disease is present. Let $ts \sim Bern(expit(-3 + (3 + 0.5)dz + 1.5(sx)(dz)))$, so we tend not to have positive tests unless there is disease, and the presence of symptoms along with the disease increases the probability.

```
dz = rbinom(n,1,pdz)
sx=rbinom(n,1,expit(-2+(2+2)*dz))
#antigen = rnorm(n,mean=-3+sx*.1+dz*1) #todo: explore cutoff
ts = rbinom(n,1,expit(-3+(3+0.5)*dz+1.5*sx*dz))
df = data.frame(dz=dz,sx=sx,ts=ts)
```

For good measure, let's calculate various statistics. Let's start with prevalences and conditional prevalences.

```
print(c("p(dz+)",pdz))
```

```
## [1] "p(dz+)" "0.2"
```

```
pdzpsxn = dim(df[(df$dz==1)&(df$sx==0),])[1]/dim(df[(df$sx==0),])[1]
pdzpsxp = dim(df[(df$dz==1)&(df$sx==1),])[1]/dim(df[(df$sx==1),])[1]
print(c("p(dz+|sx-)",pdzpsxn))
```

```
## [1] "p(dz+|sx-)"          "0.0330934856640668"
```

```
print(c("p(dz+|sx+)",pdzpsxp))
```

```
## [1] "p(dz+|sx+)"          "0.646305563732705"
```

Calculate the $p(dz + |test-)$ sensitivity and specificity.

```
ptspdzp=dim(df[(df$dz==1)&(df$ts==1),])[1]/dim(df[(df$dz==1),])[1]
ptsndzn=dim(df[(df$dz==0)&(df$ts==0),])[1]/dim(df[(df$dz==0),])[1]
print(c("marg sens: p(test+|dz+)",round(ptspdzp,2)))
```

```
## [1] "marg sens: p(test+|dz+)" "0.85"
```

```
print(c("marg spec: p(test-|dz-)",round(ptsndzn,2)))
```

```
## [1] "marg spec: p(test-|dz-)" "0.95"
```

```
pdzptsn = dim(df[(df$dz==1)&(df$ts==0),])[1]/dim(df[(df$ts==0),])[1]
print(c("p(dz+|test-)",pdzptsn))
```

```
## [1] "p(dz+|test-)"          "0.0374678306504516"
```

Calculate the conditional sensitivities and specificities. First, let us consider what happens given symptoms.

```
ptspdzpsxp = dim(df[(df$ts==1)&(df$dz==1)&(df$sx==1),])[1]/dim(df[(df$dz==1)&(df$sx==1),])[1]
ptsndznsxp = dim(df[(df$ts==0)&(df$dz==0)&(df$sx==1),])[1]/dim(df[(df$dz==0)&(df$sx==1),])[1]
print(c("Cond sens, sx-: p(test+|dz+,sx+)",round(ptspdzpsxp,2)))
```

```
## [1] "Cond sens, sx-: p(test+|dz+,sx+)" "0.88"
```

```r
print(c("Cond sens, sx-:p(test-|dz-,sx+)",round(ptsndznsxp,2)))
```

```
## [1] "Cond sens, sx-:p(test-|dz-,sx+)" "0.96"
```

```r
pdzptsnsxp = dim(df[(df$ts==0)&(df$dz==1)&(df$sx==1),])[1]/dim(df[(df$ts==0)&(df$sx==1),])[1]
print(c("p(dz+|test-,sx+)",pdzptsnsxp))
```

```
## [1] "p(dz+|test-,sx+)"  "0.180345380096137"
```

Now let us consider what happens with no symptoms.

```r
ptspdzpsxn = dim(df[(df$ts==1)&(df$dz==1)&(df$sx==0),])[1]/dim(df[(df$dz==1)&(df$sx==0),])[1]
ptsndznsxn = dim(df[(df$ts==0)&(df$dz==0)&(df$sx==0),])[1]/dim(df[(df$dz==0)&(df$sx==0),])[1]
print(c("Cond spec, sx+: p(test+|dz+,sx-)",round(ptspdzpsxn,2)))
```

```
## [1] "Cond spec, sx+: p(test+|dz+,sx-)" "0.61"
```

```r
print(c("Cond spec, sx+: p(test+|dz-,sx-)",round(ptsndznsxn,2)))
```

```
## [1] "Cond spec, sx+: p(test+|dz-,sx-)" "0.95"
```

```r
pdzptsnsxn = dim(df[(df$ts==0)&(df$dz==1)&(df$sx==0),])[1]/dim(df[(df$ts==0)&(df$sx==0),])[1]
print(c("p(dz+|test-,sx-)",pdzptsnsxn))
```

```
## [1] "p(dz+|test-,sx-)"   "0.0138754152335597"
```

Now look at $P(sx|ts)$.

```r
psxptsn = dim(df[(df$sx==1)&(df$ts==0),])[1]/dim(df[(df$ts==0),])[1]
psxntsn = dim(df[(df$sx==0)&(df$ts==0),])[1]/dim(df[(df$ts==0),])[1]
print(c("P(sx+|test-)",round(psxptsn,2)))
```

```
## [1] "P(sx+|test-)" "0.14"
```

```r
print(c("P(sx-|test-)",round(psxntsn,2)))
```

```
## [1] "P(sx-|test-)" "0.86"
```

Finally, let's check that our estimates using Bayes rule correspond to our estimates without it. Note how much simpler the former is!

```r
print(c("p(dz+|test-)",pdzptsn))
```

```
## [1] "p(dz+|test-)"        "0.0374678306504516"
```

```r
print(c("p(dz+|test-,sx+)p(sx+|test-)+p(dz+|test-,sx-)p(sx-|test-)",
        pdzptsnsxp*psxptsn + pdzptsnsxn*psxntsn))
```

```
## [1] "p(dz+|test-,sx+)p(sx+|test-)+p(dz+|test-,sx-)p(sx-|test-)"
## [2] "0.0374678306504516"
```

```r
print(c("p(dz+|test-,sx+)",pdzptsnsxp))
```

```
## [1] "p(dz+|test-,sx+)"  "0.180345380096137"
```

```r
print(c("p(dz+|test-,sx+)",get.post.tests(sens=ptspdzpsxp,
                                          spec=ptsndznsxp,pre.tests=c(pdzpsxp))))
```

```
## [1] "p(dz+|test-,sx+)"  "0.180345380096137"
```

```r
#get.post.tests()

print(c("p(dz+|test-,sx-)",pdzptsnsxn))
```

```
## [1] "p(dz+|test-,sx-)"    "0.0138754152335597"
```

```r
print(c("p(dz+|test-,sx-)",get.post.tests(sens=ptspdzpsxn,
                                          spec=ptsndznsxn,pre.tests=c(pdzpsxn))))
```

```
## [1] "p(dz+|test-,sx-)"    "0.0138754152335597"
```

```r
#cbind(pts$`Conditional sx+`$m,pts$`Conditional sx-`$m,pts$$p(dz+|test-)$$m,pre.tests)[21,]
#cbind(pdzptsnsxp,pdzptsnsxn)
```

So, everything seems to work out. Now let us plot a similar diagram to the one above.

```r
tests=list(
condsxp=list(
  nm='p(dz+|test-, sx+)',
  sens=list(m=ptspdzpsxp,l=ptspdzpsxp,u=ptspdzpsxp),
  spec=list(m=ptsndznsxp,l=ptsndznsxp,u=ptsndznsxp)
  ),
marg=list(
  nm='p(dz+|test-)',
  sens=list(m=ptspdzp,l=ptspdzp,u=ptspdzp),
  spec=list(m=ptsndzn,l=ptsndzn,u=ptsndzn)
),
condsxn=list(
  nm='p(dz+|test-, sx-)',
  sens=list(m=ptspdzpsxn,l=ptspdzpsxn,u=ptspdzpsxn),
  spec=list(m=ptsndznsxn,l=ptsndznsxn,u=ptsndznsxn)
  )
)

pts=get.t.posts(tests)

plot.p(pts,
       post.label="p(dz+|test-,...)",xlab="Pre-test: p(dz+|...)",
       leg.labels=c(TeX("p(dz+|test-,sx+)"),"p(dz+|test-)",
                    TeX("p(dz+|test-,sx-)")),
       main="Simulation of p(dz+|test-,sx+) vs. p(dz+|test-)"
       )
colorBlindBlack8  <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                       "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

points(pdz,pdzptsnsxp,col=colorBlindBlack8[1],pch=19)
points(pdzpsxp,pdzptsnsxp,col=colorBlindBlack8[1],pch=19)
points(pdz,pdzptsn,col=colorBlindBlack8[2],pch=19)
points(pdz,pdzptsnsxn,col=colorBlindBlack8[3],pch=19)
points(pdzpsxn,pdzptsnsxn,col=colorBlindBlack8[3],pch=19)
```
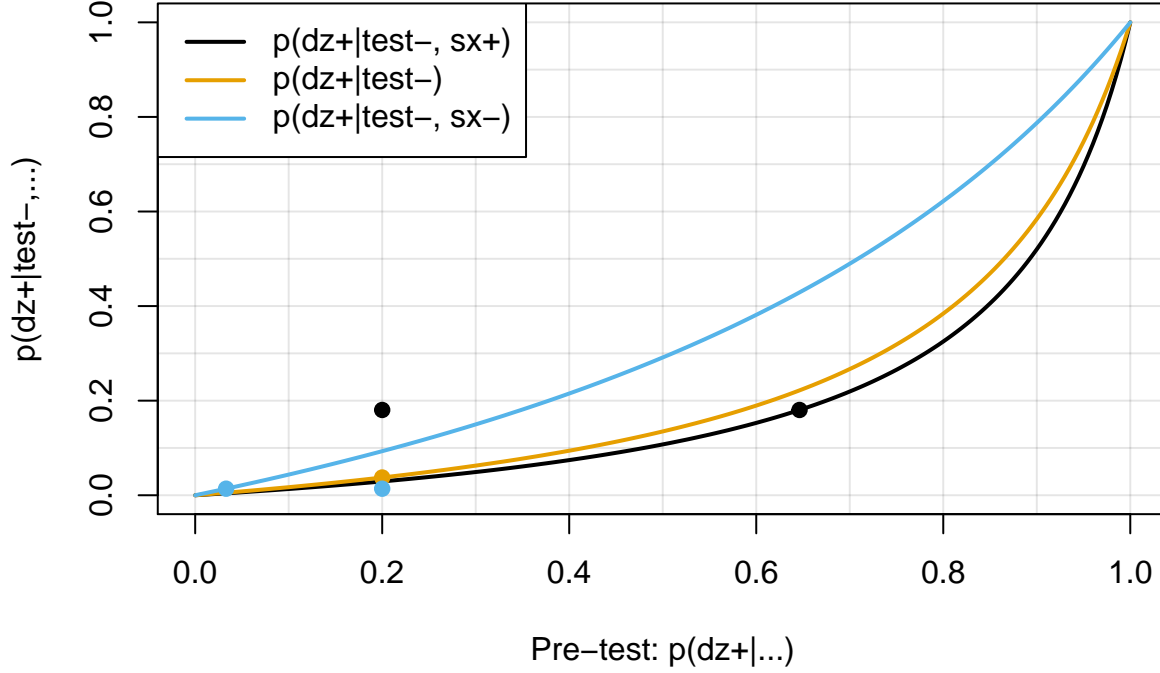
Simulation of p(dz+|test−,sx+) vs. p(dz+|test−)

This corresponds to what we see with $p(dz+|test-,sx)$ from (Prince-Guerra et al. 2021) with $p(dz+|test-)$ (Bruemmer et al. 2021). The implication is that if there is a mismatch between the covariates of the study and our own covariates, we can get into a lot of trouble. Consider the lines above and below the point for $p(dz+|test-)$ at the true prevalence of $p(dz+) = 0.2$. We have that $p(dz+|test-)$ is closer to $p(dz+|test-,sx+)$. However, consider the points at $p(dz+)$, which correpond to $p(dz+|test-,sx)$ at $p(dz+|sx)$. We have that $p(dz+|test-)$ is closer to $p(dz+|test-,sx-)$! The latter is correct.

Bound for post-test probability given exposure

Recall that my original question was whether I had the disease given a sustained exposure. Unfortunately, (Prince-Guerra et al. 2021) does not give us operating characteristics that depend on exposure (although their data do suggest to us that it is relevant). However, the following section is just some musings on possibly being able to make a statement like "disease probabilty is greater than this value, even though we know it is not this value precisely." We can say, for example, $p(dz+|test-) \leq p(dz+|test-,exposure+)$, which is based on the fact that a sustained exposure increases probability of disease.

We can also somewhat more circuitously write (just to see the term $p(dz+|sx,ad,test-)$ explicitly within $p(dz+|test-,sx,ad,exposure+)$)

$$p(dz+|test-,sx,ad,exposure+) = \frac{p(exposure+|test-,dz+,sx,ad)}{p(exposure+|test-,sx,ad)}p(dz+|sx,ad,test-).$$

We see that if

$$\frac{p(exposure+|test-,dz+,sx,ad)}{p(exposure+|test-,sx,ad)} > 1,$$

which is the case when the event "exposure" increases the probability of the disease (I assume this is true), then

$$p(dz + |sx, ad, test-),$$

which we can obtain from (Prince-Guerra et al. 2021), is a lower bound on

$$p(dz + |exposure+, sx, ad, test-).$$

So, even though the estimates in (Prince-Guerra et al. 2021) do not apply to our situation directly, if we were to assume that other variables, besides those concerning age cutoff, symptom status, test result, and exposure, are negligible, our post-test probability estimate can still be considered a lower bound, which still gives us some information, and also shows us, even with our strong assumptions, how little we actually know.

On the test cutoff (if there is a cutoff)

It is not clear this is how it works, but, in general, test cutoffs have highly significant implications. If indeed there is a cutoff, what is the reward function that is being optimized? It appears that these tests were designed to minimize false positive results. However, that is not, in general, always a good idea. Decreasing false positives (e.g., by setting a high cutoff) also increases false negatives. In general, for someone who works in a highly populated area, or with vulnerable populations, a false negative is worse than a false positive. Therefore, it should be a priority not to include a threshold in the test, if possible.

As a thought experiment, consider a world in which people do not interpret tests, and just act according to the test result. In other words, if I test negative, I assume I have no disease, and if I test positive, I assume I have the disease. In this case, I believe (not sure) that the cutoff will set the reward that this society achieves entirely. In other words, it is important to interpret a test in order to avoid having your reward function be set externally by, e.g., a test cutoff. As we have seen, interpretation is complicated. Therefore, it is important that the test manufacterers avoid cutoffs. This could be done by providing a number rather than a binary test result, and I realize that this may be a difficult engineering problem.

Serial testing

I have also done some more thinking on serial testing - my current thinking (maybe this is not correct, I need to still write it out here) is that **if the tests are independent**, you can essentially treat the post-test probability from the first test as the pre-test probability for the second. If this is the case, then two tests taken, premeditated, in sequence, will perform like a better test. If this is the case, using the plot, you can just start with a pretest and update it, and then make your next pretest your post-test from the first test.

Generally, though, it is also advised to take the two tests e.g. 24 hours apart to see if the viral load increases during that time. I am not sure that two tests that are taken like this are still independent.

Viral load

I mentioned viral load above. See also e.g. (Sethuraman, Jeremiah, and Ryo 2020). Viral load (obviously) affects sensitivity and specificity. It would be important to measure the time of supposed exposure in order to take this into account $p(dz + |..., exposure\ time, ...)$.

Positive tests

Note that I am focusing only on the negative test case, although you could do the same for positive tests (I said originally that this was a non-issue, but I should not have — you can, of course, have a positive test

with no disease, and I should repeat the analysis above for that case).

Code link

Code: https://github.com/samuelweisenthal/BinaxNOWpostTestProb

References

Bruemmer, Lukas E, Stephan Katzenschlager, Mary Gaeddert, Christian Erdmann, Stephani Schmitz, Marc Bota, Maurizio Grilli, et al. 2021. "The Accuracy of Novel Antigen Rapid Diagnostics for Sars-Cov-2: A Living Systematic Review and Meta-Analysis." *medRxiv*. Cold Spring Harbor Laboratory Press.

Euser, Sjoerd, Sem Aronson, Irene Manders, Steven van Lelyveld, Bjorn Herpers, Jan Sinnige, Jayant Kalpoe, et al. 2021. "SARS-Cov-2 Viral Load Distribution Reveals That Viral Loads Increase with Age: A Retrospective Cross-Sectional Cohort Study." *medRxiv*. Cold Spring Harbor Laboratory Press.

Harrell, Frank E. 2020. https://www.fharrell.com/post/backwards-probs/.

Mahallawi, Waleed H, Ali Dakhilallah Alsamiri, Alaa Faisal Dabbour, Hamdah Alsaeedi, and Abdulmohsen H Al-Zalabani. 2021. "Association of Viral Load in Sars-Cov-2 Patients with Age and Gender." *Frontiers in Medicine* 8. Frontiers: 39.

Moons, Karel GM, and Frank E Harrell. 2003. "Sensitivity and Specificity Should Be de-Emphasized in Diagnostic Accuracy Studies." *Academic Radiology* 10 (6). Elsevier: 670–72.

Prince-Guerra, Jessica L, Olivia Almendares, Leisha D Nolen, Jayleen KL Gunn, Ariella P Dale, Sean A Buono, Molly Deutsch-Feldman, et al. 2021. "Evaluation of Abbott Binaxnow Rapid Antigen Test for Sars-Cov-2 Infection at Two Community-Based Testing Sites—Pima County, Arizona, November 3–17, 2020." *Morbidity and Mortality Weekly Report* 70 (3). Centers for Disease Control; Prevention: 100.

Sethuraman, Nandini, Sundararaj Stanleyraj Jeremiah, and Akihide Ryo. 2020. "Interpreting Diagnostic Tests for Sars-Cov-2." *Jama* 323 (22). American Medical Association: 2249–51.