BIOSTAT/STAT 570: Coursework 7

To be submitted to the course canvas site by 11:59pm Wednesday 8th December, 2021.

1. (a) In this question a simulation study to investigate the impact on inference of omitting covariates in logistic regression will be performed, in the situation in which the covariates are independent of the exposure of interest. Let x be the covariate of interest and z another covariate. Suppose the true (adjusted) model is $Y_i \mid x_i, z_i \sim_{iid}$ Bernoulli(p_i), with

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i + \beta_2 z_i. \tag{1}$$

A comparison with the unadjusted model $Y_i \mid x_i \sim_{iid}$ Bernoulli$(p_i^\star)$, where

$$\log\left(\frac{p_i^\star}{1-p_i^\star}\right) = \beta_0^\star + \beta_1^\star x_i, \tag{2}$$

for $i = 1, \dots, n = 1000$ will be made. Suppose x is binary with $\Pr(X = 1) = 0.5$ and $Z \sim_{iid}$ N$(0,1)$ with x and z independent. Combinations of the parameters $\beta_1 = 0.5, 1.0$ and $\beta_2 = 0.5, 1.0, 2.0, 3.0$, with $\beta_0 = -2$ in all cases, will be considered. For each combination of parameters compare the results from the two models, (1) and (2), with respect to:

i. E$[\widehat{\beta}_1]$ and E$[\widehat{\beta}_1^\star]$, as compared to β_1.

ii. The standard errors of $\widehat{\beta_1}$ and $\widehat{\beta}_1^\star$.

iii. The coverage of 95% confidence intervals for β_1 and $\beta_1^\star$.

iv. The probability of rejecting $H_0 : \beta_1 = 0$ (the power) under both models using a Wald test.

Based on the results, summarize the effect of omitting a covariate that is independent of the exposure of interest, in particular in comparison with the linear model case.

Solution: We simulate $10,000$ datasets for each parameter combination. The results for items (i)-(iv) are given in Figures 1-4.

Figure 1 shows that $E[\hat{\beta}_1] = \beta_1$ across all parameter settings. On the other hand, $E[\hat{\beta}_1^*] \neq \beta_1$. We conclude that omitting z from the model causes us to estimate a different parameter– $\beta_1^* \neq \beta_1$ and the discrepancy increases as the magnitude of the effect of z (β_2) on the outcome increases.

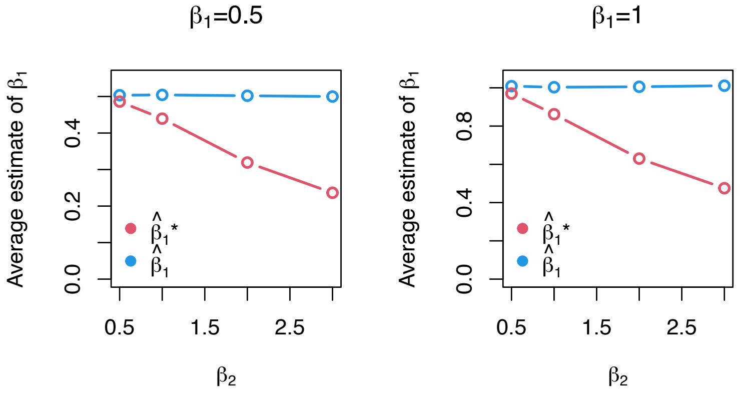

Figure 1: Bias results.

We next look at the average standard error reported by GLM for $\widehat{\beta_1}$ and $\hat{\beta}_1^\star$ under each model for each parameter setting. The results are in the top row of Figure 2. As the coefficient on the omitted variable grows, the estimated standard error for $\hat{\beta}_1$ grows while the one for $\hat{\beta}_1^*$ shrinks. These average reported standard errors from GLM closely match the empirical standard errors (computed over the 10,000 realizations) that are shown in Figure **??**. This suggests that the standard errors reported by GLM are accurate for the true sampling distributions of the coefficients.

Figure 3 shows that the coverage of confidence intervals based on $\hat{\beta}_1^*$ for the parameter β_1 gets worse as β_2 increases. We know from Figure 1 that the confidence intervals are centered at the wrong location, especially as β_2 increases. We learned in 2 that GLM computes correct standard errors under both models. Therefore, the incorrect coverage seems to be due to the confidence intervals for $\hat{\beta}^*$ not being centered at β_1, rather than due to incorrect standard errors.

Figure 4 shows that the power of the model that omits z is worse than the power of the correct model. The power of both models to detect β_1 decreases as β_2 increases because there is more noise in the data.

All together, these findings are in contrast to what happens when we use linear regression. In linear regression, omitting a covariate z that is independent from the included covariate x does not effect bias or coverage in the estimation of the coefficient on x (but it can affect power since it adds more noise to the residuals). In a GLM, when we fit the model that omits z, our estimated coefficient for x, $\hat{\beta}^*$, is unbiased for a true parameter β_1^* that is not equal to β_1.

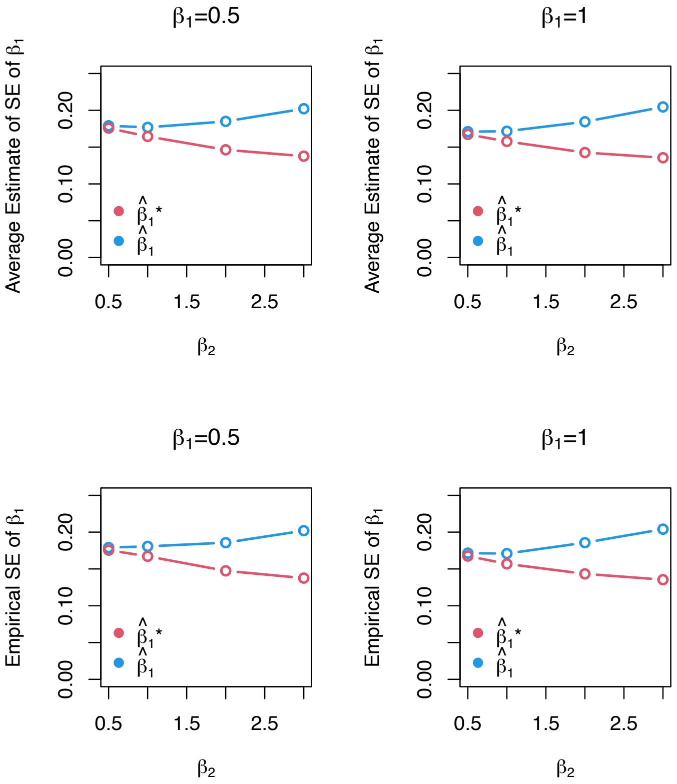

Figure 2: Top row: Average estimated standard errors, as reported by GLM. Bottom row: Empirical standard errors of the coefficients across 10,000 datasets.

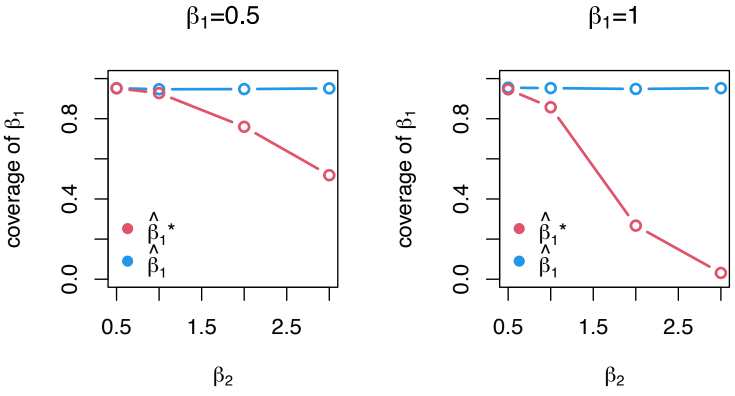

Figure 3: Coverage Results.

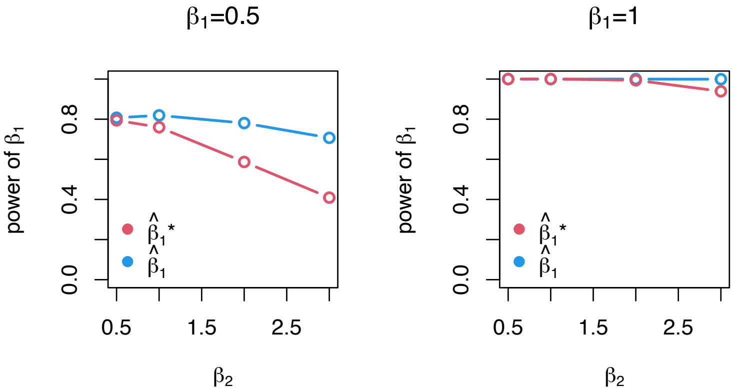

Figure 4: Power results.

2. The Pima, or Akimel O'odham, are an indigenous Native American tribe that originates from southern Arizona. In this question you will analyze data on Pima native American women who are at least 21 years of age; these data were originally from the National Institute of Diabetes and Digestive and Kidney Diseases. We will take the aim of the analysis to obtain a model to understand the association between diabetes status and the covariates in the sampled population, using binomial GLMs. The data may be found in the `mlbench` library and are called `PimaIndiansDiabetes2`.

   The variables we will examine as predictors are:

```
pregnant Number of times pregnant
glucose Plasma glucose concentration (glucose tolerance test)
mass Body mass index (weight in kg/(height in m)\^2)
pedigree Diabetes pedigree function
age Age (years)
```

   (a) We will examine models of the form

$$\begin{aligned} Y_i|p_i &\sim \text{Binomial}(1, p_i) \\ g(p_i) &= \beta_0 + \beta_1 \times \texttt{pregnant} + \beta_2 \times \texttt{glucose} + \beta_3 \times \texttt{mass} + \beta_4 \times \texttt{pedigree} + \beta_5 \times \texttt{age} \end{aligned}$$

   for $i = 1, \ldots, n$ women, and where the link function $g(\cdot)$ is one of `logit`, `probit`, `cloglog`.

   Form a new dataset containing y and the required x variables, removing the records that contain missing values.

   Solution: Code in appendix. 16 observations had missing values, so the new dataset contains 752 observations.

   (b) Fit the three binomial models that correspond to the different link functions, and give a table containing the parameter estimates along with standard errors.

   Solution: All three models are fit with GLM. The results are in Table 2b.

	Logit Estimate	Logit SE	Probit Estimate	Probit SE	Cloglog Estimate	Cloglog SE
(Intercept)	-9.32	0.74	-5.47	0.40	-6.59	0.49
pregnant	0.12	0.03	0.07	0.02	0.08	0.02
glucose	0.04	0.00	0.02	0.00	0.02	0.00
mass	0.09	0.01	0.05	0.01	0.06	0.01
pedigree	0.92	0.30	0.46	0.17	0.23	0.19
age	0.01	0.01	0.01	0.01	0.01	0.01

(c) Which link function provides the most interpretable coefficients? For this link function, provide a brief summary of your fitted model.

Solution: The Logit model is interpretable in terms of odds ratios. We interpret the intercept $exp(\beta_0)$ as the odds of having diabetes among individuals who have a value of 0 for all other covariates. As no one in the dataset has 0 age, this is not particularly useful to interpret. For β_1 through β_5, we interpret $exp(\beta_i)$ as the multiplicative increase in the odds of having diabetes that is associated with increasing covariate i by 1 unit while holding the rest constant. In the model that we fit, all covariates besides age have a statistically significant affect on the odds of having diabetes.

(d) Provide a plot showing the estimated association between diabetes prevalence and pedigree, under the three models. Provide pointwise confidence intervals to your plot, carefully explaining your method. How should one interpret this plot?

Solution: We plot pedigree vs. the predicted probability of diabetes for women with this pedigree. In order to plot these curves for the models we built above, we need to plug in values for the remaining covariates (we know from problem 1 that simply omitting these covariates is a bad idea!). I chose to plug in mean values of the other four covariates, knowing that if there is a lot of correlation between the covariates then the prediction at pedigree=3, for example, may not represent the prediction for the "typical woman" with pedigree=3.

In order to add confidence bands around this prediction line, there are two options. We could use the Multivariate Delta method, or we could use the computational alternative suggested on slide 117 of the Lecture 7 notes. For the computational alternative, we do the following.

- Fit our GLM and obtain the asymptotic covariance matrix V using `vcov()`.
- Simulate $\hat{\beta}^i$ for $i = 1, \ldots, 10,000$ by drawing them from $N_6(\hat{\beta}, V)$.
- For each $\hat{\beta}^i$, for several different values of `pedigree`, p, compute

$$f(\tilde{\beta}^i, p) = g^{-1}\left(\bar{X}^p \tilde{\beta}^i\right),$$

  where $\bar{X}_p$ contains the average values of `age`, `mass`, etc., but has the value p plugged in for pedigree.
- Find the 5th and 95th percentiles of this empirical distribution for each value of p. These give confidence bands for our plot at `pedigree=`p.

The plot that we obtain from following this method is given in Figure **??**. We see that uncertainty is higher for higher values of pedigree on this transformed scale.

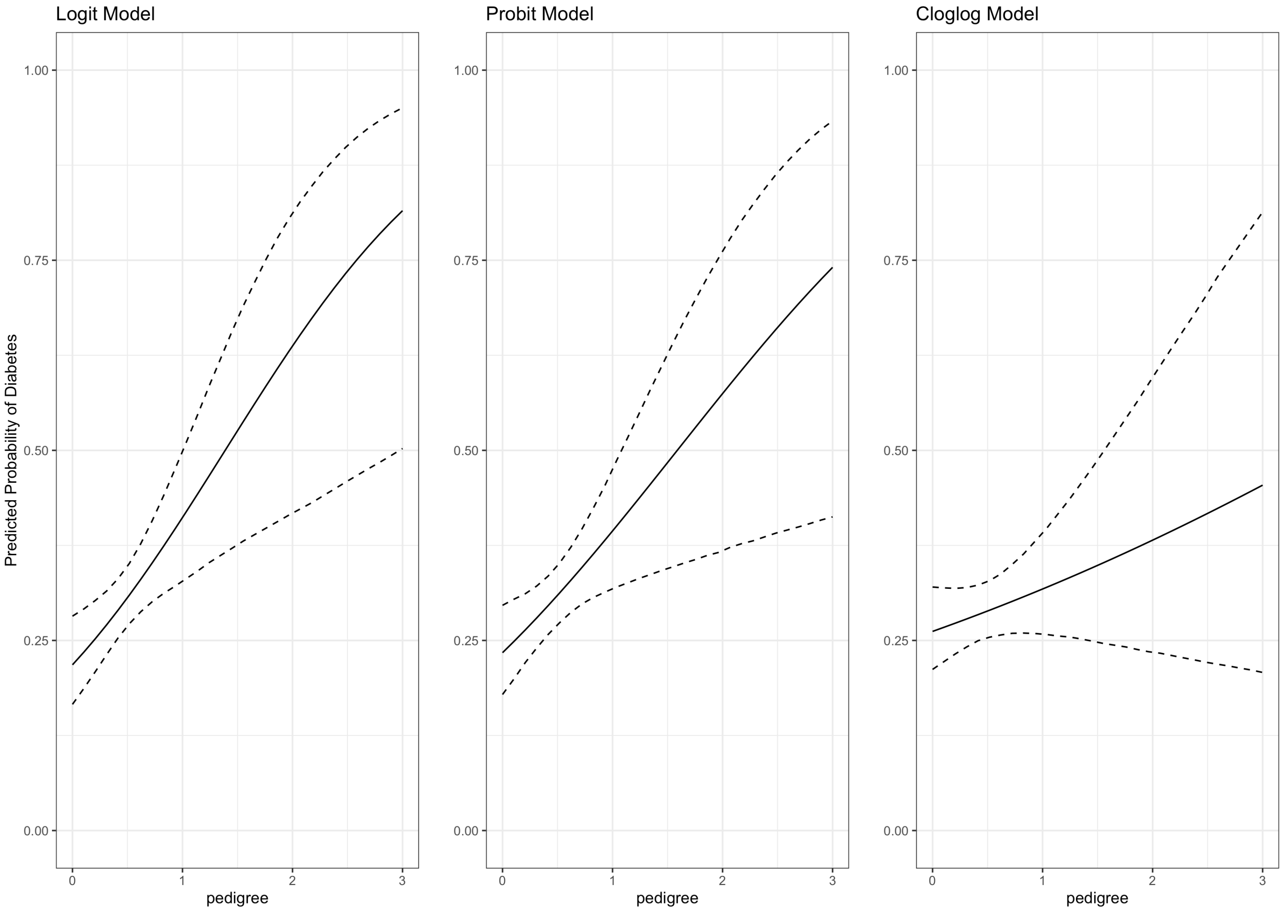

Figure 5: Pointwise predictions under three models with confidence bands added.

(e) Suppose the aim of the analysis was prediction, rather than understanding associations. How would this affect the way you carry out your analysis, and how would you assess the success of a model? There is no need to do any analyses here, I just want to see an outline of what you would do.

Solution: If we only cared about prediction, we would not need our link function to provide interpretable parameter estimates. Therefore, we would want to pick the link function that is capable of achieving the highest predictive accuracy. We would also want to consider possibly adding or removing covariates and interaction terms from the model, looking for the combination that would achieve the highest predictive accuracy. To figure out which link function and which combination of covariates achieves the highest predictive accuracy, we would want to use cross validation to ensure that we are not just picking a model that overfits to the training data.

A Code for Problem 1

```
# Question 2
do.one <- function(b0,b1,b2,n=1000){
  x <- rbinom(n,1,prob=0.5)
  z <- rnorm(n)
  mu <- b0 + b1*x + b2*z
  p <- exp(mu)/(1+exp(mu))
  y <- rbinom(n,1,p)
  m1 <- glm(y~1+x+z,family = "binomial")
  m2 <- glm(y~1+x, family = "binomial")
  b.est1 <- m1$coefficients[2]
  b.est2 <- m2$coefficients[2]
  se1 <- sqrt(diag(vcov(m1)))[2]
  se2 <- sqrt(diag(vcov(m2)))[2]
  ci1 <- b.est1 + se1 %o% c(qnorm(0.025), qnorm(0.975))
  ci2 <- b.est2 + se2 %o% c(qnorm(0.025), qnorm(0.975))
  cover1 <- b1 >= ci1[1] & b1 <= ci1[2]
  cover2 <- b1 >= ci2[1] & b1 <= ci2[2]
  rej1 <- 0 < ci1[1] | 0 > ci1[2]
  rej2 <- 0 < ci2[1] | 0 > ci2[2]
  return(c(b.est1=b.est1, b.est2=b.est2, se1=se1, se2=se2,cover1= cover1,
       cover2=cover2, rej1=rej1, rej2=rej2))
}

init1 <- c(0.5,1)
init2 <- c(0.5,1,2,3)
result <- NULL
betterSEs <- NULL
for(b1 in init1){
  for(b2 in init2){
    temp <- replicate(10000, do.one(-2, b1, b2))
    betterSEs <- cbind(betterSEs, apply(temp,1,sd))
    result <- cbind(result, apply(temp,1,mean))
    cat(".")
  }
}

# (a)
tbl1 <- result[1:2,]
library(xtable)
xtable(tbl1, digits = 3)

pdf("f2a.pdf",width = 6,height = 3.5)
par(mfrow=c(1,2))
for(i in 1:2){
  plot(x=init2, y=tbl1[1,(i*4-3):(i*4)],main=substitute(paste(beta[1],"=",xxx),
       list(xxx=init1[i])),ylim=c(0,init1[i]+0.05),
       col=4,type="b",lwd=2,lty=1, xlab=expression(beta[2]),
       ylab=expression(paste("Average estimate of ",beta[1])))
  lines(x=init2,y=tbl1[2,(i*4-3):(i*4)],type="b",col=2,lwd=2,lty=1)
  legend(x="bottomleft",pch=c(19,19),col=c(2,4),
         c(expression(paste(hat(beta)[1],"*")),
           expression(paste(hat(beta)[1]))),
         bty="n")
}
dev.off()

#(b)
tbl1 <- result[3:4,]
xtable(tbl1, digits = 3)
pdf("f2b.pdf",width = 6,height = 3.5)
par(mfrow=c(1,2))
```

```
for(i in 1:2){
  plot(x=init2, y=tbl1[1,(i*4-3):(i*4)],main=substitute(paste(beta[1],"=",xxx),
       list(xxx=init1[i])),ylim=c(0,0.2+0.05),
       col=4,type="b",lwd=2,lty=1, xlab=expression(beta[2]),
       ylab=expression(paste("Average Estimate of SE of ",beta[1])))
  lines(x=init2,y=tbl1[2,(i*4-3):(i*4)],type="b",col=2,lwd=2,lty=1)
  legend(x="bottomleft",pch=c(19,19),col=c(2,4),
         c(expression(paste(hat(beta)[1],"*")),
           expression(paste(hat(beta)[1]))),
         bty="n")
}
dev.off()


#### EMPIRICAL SEs.
tbl1 <- betterSEs[1:2,]
pdf("f2b2.pdf",width = 6,height = 3.5)
par(mfrow=c(1,2))
for(i in 1:2){
  plot(x=init2, y=tbl1[1,(i*4-3):(i*4)],main=substitute(paste(beta[1],"=",xxx),
       list(xxx=init1[i])),ylim=c(0,0.2+0.05),
       col=4,type="b",lwd=2,lty=1, xlab=expression(beta[2]),
       ylab=expression(paste("Empirical SE of ",beta[1])))
  lines(x=init2,y=tbl1[2,(i*4-3):(i*4)],type="b",col=2,lwd=2,lty=1)
  legend(x="bottomleft",pch=c(19,19),col=c(2,4),
         c(expression(paste(hat(beta)[1],"*")),
           expression(paste(hat(beta)[1]))),
         bty="n")
}
dev.off()

#(c)
tbl1 <- result[5:6,]
xtable(tbl1, digits = 3)
pdf("f2c.pdf",width = 6,height = 3.5)
par(mfrow=c(1,2))
for(i in 1:2){
  plot(x=init2, y=tbl1[1,(i*4-3):(i*4)],main=substitute(paste(beta[1],"=",xxx),
       list(xxx=init1[i])),ylim=c(0,1),
       col=4,type="b",lwd=2,lty=1, xlab=expression(beta[2]),
       ylab=expression(paste("coverage of ",beta[1])))
  lines(x=init2,y=tbl1[2,(i*4-3):(i*4)],type="b",col=2,lwd=2,lty=1)
  legend(x="bottomleft",pch=c(19,19),col=c(2,4),
         c(expression(paste(hat(beta)[1],"*")),
          expression(paste(hat(beta)[1]))),
         bty="n")
}
dev.off()

# (d)
tbl1 <- result[7:8,]
xtable(tbl1, digits = 3)
pdf("f2d.pdf",width = 6,height = 3.5)
par(mfrow=c(1,2))
for(i in 1:2){
  plot(x=init2, y=tbl1[1,(i*4-3):(i*4)],
       main=substitute(paste(beta[1],"=",xxx),list(xxx=init1[i])),ylim=c(0,1),
       col=4,type="b",lwd=2,lty=1, xlab=expression(beta[2]),
        ylab=expression(paste("power of ",beta[1])))
  lines(x=init2,y=tbl1[2,(i*4-3):(i*4)],type="b",col=2,lwd=2,lty=1)
  legend(x="bottomleft",pch=c(19,19),col=c(2,4),
         c(expression(paste(hat(beta)[1],"*")),
           expression(paste(hat(beta)[1]))),
         bty="n")
}
dev.off()
```

B Code for Problem 2

```
library(mlbench)
data(PimaIndiansDiabetes2)
head(PimaIndiansDiabetes2)
Pima2 <- PimaIndiansDiabetes2[,-c(3,4,5)]
Pima2NA <- na.omit(Pima2)
modLR <- glm(diabetes˜., data = Pima2NA, family = "binomial")
modPL <- glm(diabetes˜., data = Pima2NA, family = binomial(link="probit"))
modCLL<- glm(diabetes˜., data = Pima2NA, family = binomial(link = "cloglog"))
```

```
coeff_res <- cbind(modLR$coefficients, sqrt(diag(vcov(modLR))),
      modPL$coefficients, sqrt(diag(vcov(modPL))),
      modCLL$coefficients, sqrt(diag(vcov(modCLL))))
xtable(coeff_res)


#### Make a test dataset to evaluate predictions on.
num = 30
test_pedigrees <- seq(0,3,length.out=num)
test_data <- apply(Pima2NA[,1:5],2,mean)
test_data <- data.frame(matrix(rep(test_data, each=num), nrow=num))
names(test_data) <- names(Pima2NA)[1:5]
test_data$pedigree <- test_pedigrees

##### Get point estimates
test_data$predsLR <- predict.glm(modLR, newdata=test_data, type="response")
test_data$predsPL <- predict.glm(modPL, newdata=test_data, type="response")
test_data$predsCLL <- predict.glm(modCLL, newdata=test_data, type="response")


##### Carry out the computational approach!!! Get samples of beta from the easy
### asymptotic distribution of Beta
beta_samps_LR <- MASS::mvrnorm(5000, mu=modLR$coefficients, Sigma=vcov(modLR))
beta_samps_PL <- MASS::mvrnorm(5000, mu=modPL$coefficients, Sigma=vcov(modPL))
beta_samps_CLL <- MASS::mvrnorm(5000, mu=modCLL$coefficients, Sigma=vcov(modCLL))

gLR <- function(beta, x) {
  exp(beta%*%x)/(1+exp(beta%*%x))
}

gPL <- function(beta, x) {
  pnorm(beta%*%x)
}

gCLL <- function(beta, x) {
  1-exp(-exp(beta%*%x))
}

### For each test data point and each model, transform the Beta sampling distributions
#### into predicted sampling distributions.
```

```
for (i in 1:NROW(test_data)) {
  x <- as.numeric(c(1,test_data[i,1:5]))
  distLR <- apply(beta_samps_LR, 1, function(u) gLR(x,u))
  distPL <- apply(beta_samps_PL, 1, function(u) gPL(x,u))
  distCLL <- apply(beta_samps_CLL, 1, function(u) gCLL(x,u))
  test_data$lowerLR[i] <- quantile(distLR, 0.025)
  test_data$upperLR[i] <- quantile(distLR, 0.975)
  test_data$lowerPL[i] <- quantile(distPL, 0.025)
  test_data$upperPL[i] <- quantile(distPL, 0.975)
  test_data$lowerCLL[i] <- quantile(distCLL, 0.025)
  test_data$upperCLL[i] <- quantile(distCLL, 0.975)
}

library(ggplot2)
library(patchwork)
p1 <- ggplot(data=test_data)+
  geom_line(aes(x=pedigree, y=predsLR))+
  geom_line(aes(x=pedigree, y=lowerLR), lty=2)+
  geom_line(aes(x=pedigree, y=upperLR), lty=2)+
  ggtitle("Logit Model")+theme_bw()+
  ylab("Predicted Probability of Diabetes")+
  ylim(0,1)
p2 <- ggplot(data=test_data)+
  geom_line(aes(x=pedigree, y=predsPL))+
  geom_line(aes(x=pedigree, y=lowerPL), lty=2)+
  geom_line(aes(x=pedigree, y=upperPL), lty=2)+
  ggtitle("Probit Model")+theme_bw()+
  ylab("")+ylim(0,1)
p3 <-  ggplot(data=test_data)+
  geom_line(aes(x=pedigree, y=predsCLL))+
  geom_line(aes(x=pedigree, y=lowerCLL), lty=2)+
  geom_line(aes(x=pedigree, y=upperCLL), lty=2)+
  ggtitle("Cloglog Model")+theme_bw()+
  ylab("")+ylim(0,1)
p1+p2+p3
ggsave("pointwise.png")
```