

## BIOSTAT/STAT 570: Coursework 3

To be submitted to the course canvas site by 11:59pm Monday 25th October, 2021.

1. Consider the Poisson-gamma random effects model given by

$$Y_i | \mu_i, \theta_i \sim \text{Poisson}(\mu_i \theta_i) \quad (1)$$

$$\theta_i \sim \text{Ga}(b, b), \quad (2)$$

which leads to a negative binomial marginal model with the variance a quadratic function of the mean. Design a simulation study, along the lines of that which produced Table 2.3 in the book (overdispersed Poisson example) to investigate the efficiency and robustness under:

- A Poisson model.
- A negative binomial model<sup>1</sup>.
- Quasi-likelihood with  $E[Y] = \mu$ ,  $\text{var}(Y) = \alpha\mu$ .
- Sandwich estimation.

Use a loglinear model

$$\log \mu_i = \beta_0 + \beta_1 x_i,$$

with  $x_i \sim_{iid} N(0, 1)$ , for  $i = 1, \dots, n$ , and  $\beta_0 = 0$  and  $\beta_1 = \log 2$ . You should carry out the simulation for:

- $b = 0.1, 1, 10, 1000$ .
- $n = 10, 20, 50, 100, 250$ .

Summarize what your take-away message is, after carrying out these simulations.

**Solution:** First note that the marginal density of  $Y_i | \mu_i$  is negative binomial with parameters  $b_i$  and  $\frac{\mu_i}{\mu_i + b}$ . A derivation of this result was not required on this homework, but knowing this can inform our interpretations of our results. The marginal model has

$$E(Y_i | \mu_i) = \mu_i$$

$$V(Y_i | \mu_i) = \mu_i + \frac{1}{b} \mu_i^2 = \mu_i \left( 1 + \frac{1}{b} \mu_i \right).$$

Note that the higher  $b$  is, the less overdispersion there is. When  $b$  is small, there is a lot of overdispersion and we expect the likelihood-based Poisson model (which assumes

---

<sup>1</sup>The `glm.nb` function in the `MASS` library can be used to fit this model

$V(Y_i|\mu) = \mu_i$ ) will have very poor coverage. As  $b$  increases, we expect the poisson model to attain approximately correct coverage because it gets closer and closer to a correctly specified mean/variance model. The quasipoisson model is also misspecified here; it assumes that the variance is a constant function of the mean, but in reality our variance here is a quadratic function of the mean. Once again, its performance should improve as  $b$  increases. We expect the negative binomial glm to behave very well because it is a correctly specified likelihood-based model. The sandwich estimator should be ok because it is robust to model-misspecification, but we also know that it can be unstable in small samples and so it won't be guaranteed to perform well for small  $n$ .

We carried out 1,000 simulations for each setting. When sample size was low there were occasionally convergence issues with the models since some samples had all  $y$ 's equal to 0. Those cases were labelled as not covered; if we cannot construct an interval, we have failed to cover our parameter. One could also discard those simulations but then the results reported would be based on a different distribution of the  $y$ 's (conditioned on the algorithm converging).

Figure 1 shows the coverage results. Code to reproduce the figure is given in the appendix. For each value of  $n$  and  $b$ , we generated a single  $X$  dataset and then 1000 realizations of  $\theta$  and  $Y | X, \theta$ . For each dataset, we fit the four models, build a confidence interval for each, and mark whether or not it covers the true parameter. As expected, when  $b$  is small the Poisson model has terrible coverage. Its coverage can actually get worse as  $n$  increases, because the model gets more confident in its bad answer. By the time  $b$  is large, all four models are essentially correctly specified and so they all achieve correct coverage. We see that when  $b$  is large, sandwich has the lowest coverage when  $n$  is small. This is because even though sandwich is robust to model misspecification this robustness relies on an empirical covariance matrix, which is unstable for small  $n$ .

When  $b$  is small, the quasipoisson model has incorrect coverage because the model is wrong (true mean-variance relationship is quadratic). If we extended the plot for  $b = 0.1$  to even larger values of  $n$ , we would expect Sandwich to eventually hit correct coverage but we would not expect quasipoisson to eventually hit correct coverage.

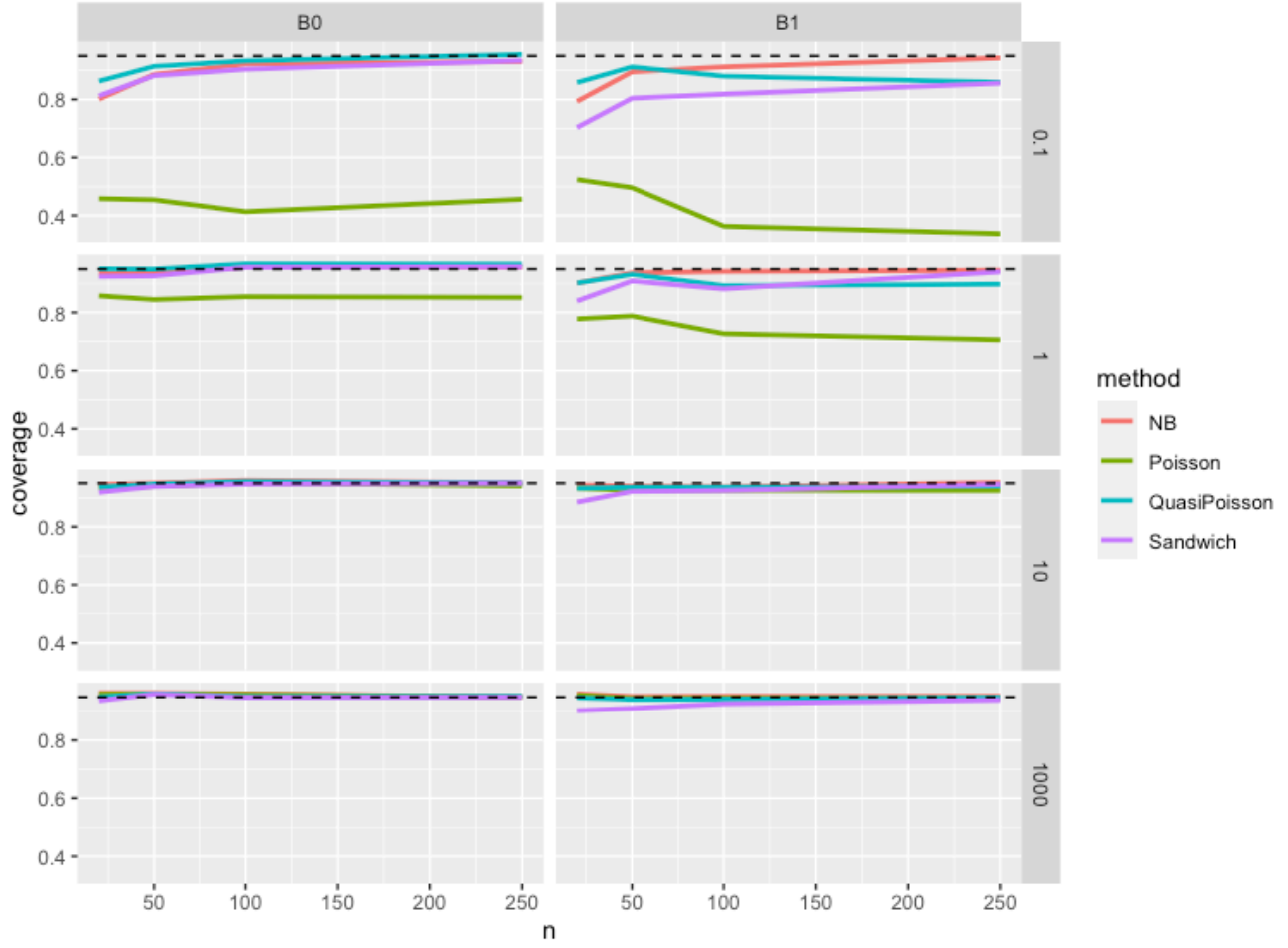


Figure 1: The rows denote the values of  $b$ .

2. The data in Table 1 contain data on a typical reliability experiment and give the failure stresses (in GPa) of four samples of carbon fibers of lengths 1, 10, 20 and 50mm.

Length (mm)	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.247	2.640	2.842	2.908	3.099	3.126	3.245	3.328	3.355	3.383	3.572	3.581	3.681
10	1.901	2.132	2.203	2.228	2.257	2.35	2.361	2.396	2.397	2.445	2.454	2.454	2.474
20	1.312	1.314	1.479	1.552	1.700	1.803	1.861	1.865	1.944	1.958	1.966	1.997	2.006
50	1.339	1.434	1.549	1.574	1.589	1.613	1.746	1.753	1.764	1.807	1.812	1.840	1.852

Table 1: Failure stress data for four groups of fibers.

- (a) The exponential distribution  $Y \mid \lambda \sim_{iid} \text{Exponential}(\lambda)$ , is a simple model for relia-

bility data:

$$p(y|\lambda) = \lambda \exp(-\lambda y),$$

with  $\lambda, y > 0$ . The hazard function is the probability of imminent failure and is given by

$$h(y|\lambda) = \frac{p(y|\lambda)}{S(y|\lambda)},$$

where  $S(y|\lambda) = \Pr(Y > y|\lambda)$  is the probability of failure beyond  $y$ . Derive the hazard function for the exponential distribution. Suppose we have a sample  $y_1, \dots, y_n$ , of size  $n$  from an exponential distribution. Find the form of the MLE of  $\lambda$  and the asymptotic variance.

**Solution:** First, we compute the survival function

$$S(y|\lambda) = \Pr(Y > y|\lambda) = \int_y^\infty \lambda \exp(-\lambda t) dt = \exp(-\lambda y) \Big|_y^\infty = \exp(-\lambda y).$$

Then we compute the hazard function

$$h(y|\lambda) = \frac{p(y|\lambda)}{S(y|\lambda)} = \frac{\lambda \exp(-\lambda y)}{\exp(-\lambda y)} = \lambda.$$

Therefore, the exponential distribution has constant hazard  $\lambda$ . For a sample of size  $n$ , the likelihood and log likelihood functions are

$$L(\lambda) = \prod_{i=1}^n \lambda \exp(-\lambda y_i) = \lambda^n \exp\left(-\lambda \sum_{i=1}^n y_i\right) \quad \text{and} \quad l(\lambda) = n \log(\lambda) - \lambda \sum_{i=1}^n y_i.$$

Differentiating by  $\lambda$  gives us the score function, and setting it equal to 0 allows us to solve for the MLE:

$$S(\lambda) = \frac{d}{d\lambda} l(\lambda) = \frac{n}{\lambda} - \sum_{i=1}^n y_i \implies \hat{\lambda} = \frac{1}{\bar{y}}.$$

The negative expected value of the derivative of the score gives us the Information function and the asymptotic variance:

$$I(\lambda) = -\mathbf{E} \left[ \frac{d}{d\lambda} S(\lambda) \right] = -\mathbf{E} \left[ -\frac{n}{\lambda^2} \right] = \frac{n}{\lambda^2} \implies \sqrt{n}(\hat{\lambda} - \lambda) \rightarrow_d N(0, \lambda^2).$$

- (b) For each of the four groups in Table 1 estimate a separate  $\lambda$ , with an associated standard error. Examine the appropriateness of the exponential model via QQ plots.

Length (mm)	$\hat{\lambda}$	Model SE	$\hat{\alpha}$	Quasi SE	Sand SE
1	0.3170	0.0879	0.0169	0.0114	0.0110
10	0.4326	0.1200	0.0051	0.0085	0.0082
20	0.5713	0.1584	0.0212	0.0231	0.0222
50	0.5999	0.1664	0.0096	0.0163	0.0157

Table 2: MLEs and standard errors for hazard rates  $\lambda$  at each length. This table aggregates answers for parts (b), (c), and (d) of the problem.

**Solution:** Table 2 shows the MLE  $\hat{\lambda}$  along with its model-based standard error for each of the four lengths of carbon fibers. Figure 2 shows Q-Q plots for each length. From these plots, the exponential likelihood model does not seem to be a good fit. The data seem to follow a line that is much less steep than the expected line, so we have some under-dispersion.

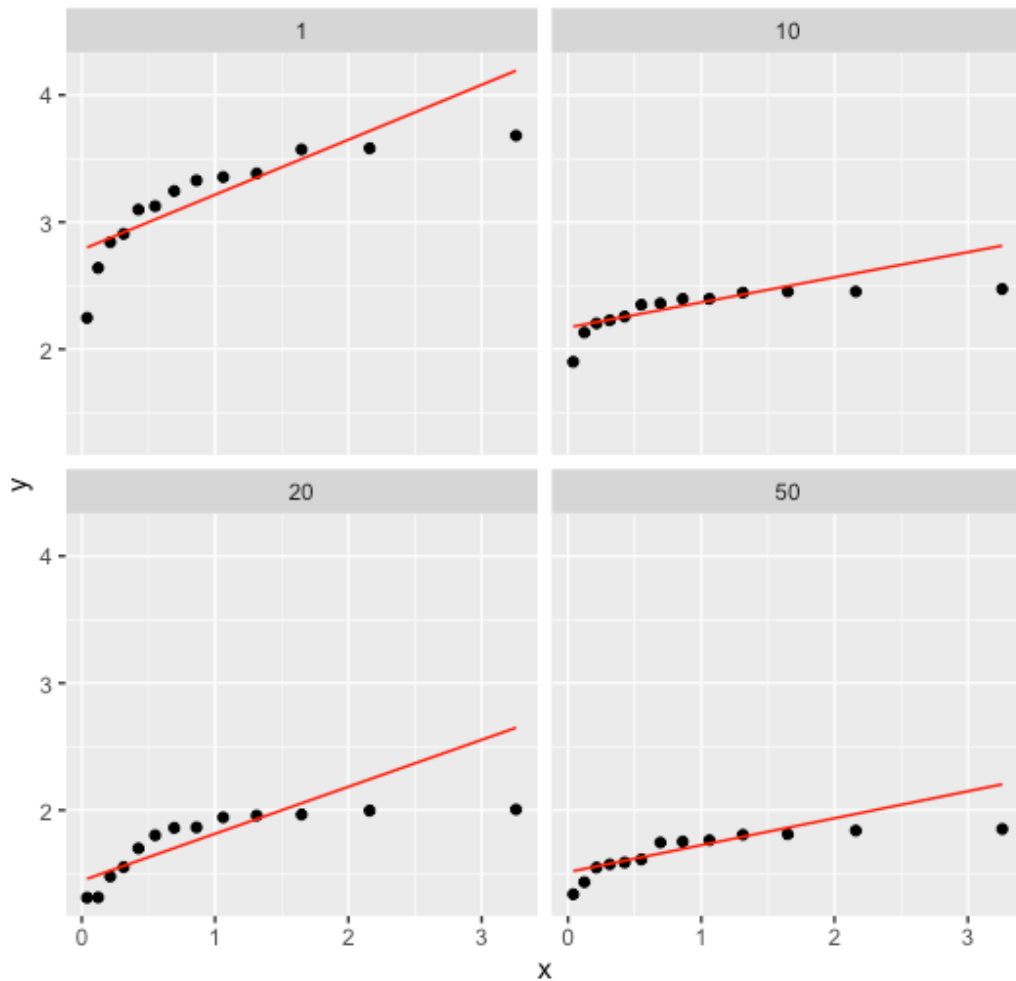


Figure 2: Q-Q plots of observations versus exponential quantiles for each length. The straight line shows the line that the data should follow if it really is exponential. This data does not appear to have come from an exponential distribution.

(c) Consider a quasi-likelihood approach to inference for  $\lambda$  under the model with

$$E[Y|\lambda] = \lambda^{-1}, \quad \text{var}(Y|\lambda) = \alpha\lambda^{-2}$$

with  $\alpha > 0$ . Suggest an estimator for  $\alpha$ . Estimate  $\lambda, \alpha$  and the standard errors, separately for each of the four groups in Table 1. What do the results suggest to you about the fit of the exponential model?

**Solution:** To estimate  $\lambda$ , we use the same MLE as under the model based approach. To estimate  $\alpha$ , we need to see how the size of our residuals compares to

the expected variance  $1/\lambda^2$  (although since we don't know  $\lambda$  we need to plug in the MLE). So we can estimate  $\alpha$  by

$$\hat{\alpha} = \frac{1}{n-1} \sum_{i=1}^n \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)} = \frac{1}{n-1} \sum_{i=1}^n \frac{(y_i - 1/\hat{\lambda})^2}{1/\hat{\lambda}^2} = \frac{1}{n-1} \sum_{i=1}^n \frac{(y_i - \bar{y})^2}{\bar{y}^2} = \frac{s^2}{\bar{y}^2},$$

so the quasi-likelihood estimate of the variance of  $\hat{\lambda}$  can be estimated by

$$\widehat{\text{var}}(\hat{\lambda}) = \hat{\alpha} \frac{\hat{\lambda}^2}{n} = \frac{s^2}{n\bar{y}^4}.$$

Table 2 shows the estimates  $\hat{\alpha}$  along with quasi-likelihood standard errors for each of the four lengths of carbon fibers. They are all quite low, confirming the underdispersion seen in the Q-Q plots.

- (d) Obtain the form of the sandwich estimate for the variance of  $\hat{\lambda}$ . Numerically evaluate sandwich standard errors for the estimate of  $\lambda$  in each of the four groups.

**Solutions:** To compute the sandwich estimate we use the estimating function  $G(\lambda, y_i) = 1/\lambda - y_i$  to obtain

$$\begin{aligned} \hat{\mathbf{A}} &= \frac{1}{n} \sum_{i=1}^n \frac{d}{d\lambda} G(\hat{\lambda}, y_i) = -\frac{1}{n} \sum_{i=1}^n -\frac{1}{\hat{\lambda}^2} = -\bar{y}^2 \\ \hat{\mathbf{B}} &= \frac{1}{n} \sum_{i=1}^n G(\hat{\lambda}, y_i)^2 = \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{\hat{\lambda}} - y_i \right)^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 = \frac{n-1}{n} \times s^2. \end{aligned}$$

Then our variance estimate is

$$\widehat{\text{var}}(\hat{\lambda}) = \hat{\mathbf{A}}^{-1} \hat{\mathbf{B}} \hat{\mathbf{A}}^{\top-1} / n = \frac{n-1}{n} \times \frac{s^2}{n\bar{y}^4}.$$

Table 2 shows the sandwich standard error estimates for each of the four lengths of carbon fibers. The differ from the quasi-likelihood standard errors only by a factor of  $(n-1)/n$ .

- (e) The Weibull distribution is a common model for survival or reliability data:  $Y \mid \eta, \alpha \sim_{iid} \text{Weibull}(\eta, \alpha)$ , with  $\eta > 0$ , and  $\alpha > 0$ . The random variable  $Y$  has a Weibull distribution if its density can be written in the form

$$p(y \mid \eta, \alpha) = \eta \alpha^{-\eta} y^{\eta-1} \exp \left[ - \left( \frac{y}{\alpha} \right)^\eta \right].$$

Find the mean, variance and hazard function of a Weibull distribution. For what value of the parameters does the exponential distribution result?

**Solution:** We can integrate the Weibull density to obtain

$$\begin{aligned}
E[Y|\eta, \alpha] &= \int_0^\infty \eta \alpha^{-\eta} y^\eta \exp\left[-\left(\frac{y}{\alpha}\right)^\eta\right] dy = \int_0^\infty \eta \left(\frac{y}{\alpha}\right)^\eta \exp\left[-\left(\frac{y}{\alpha}\right)^\eta\right] dy \\
&= \alpha \int_0^\infty t^{1/\eta} \exp(-t) dt, \quad \text{substituting } t = \left(\frac{y}{\alpha}\right)^\eta, dt = \frac{\eta}{\alpha} \left(\frac{y}{\alpha}\right)^{\eta-1} dy, \\
&= \alpha \Gamma(1 + 1/\eta), \\
E[Y^2|\eta, \alpha] &= \int_0^\infty \eta \alpha^{-\eta} y^{\eta+1} \exp\left[-\left(\frac{y}{\alpha}\right)^\eta\right] dy = \int_0^\infty \alpha \eta \left(\frac{y}{\alpha}\right)^{\eta+1} \exp\left[-\left(\frac{y}{\alpha}\right)^\eta\right] dy \\
&= \alpha^2 \int_0^\infty t^{2/\eta} \exp(-t) dt, \quad \text{substituting } t = \left(\frac{y}{\alpha}\right)^\eta, dt = \frac{\eta}{\alpha} \left(\frac{y}{\alpha}\right)^{\eta-1} dy, \\
&= \alpha^2 \Gamma(1 + 2/\eta), \\
\text{var}(Y|\eta, \alpha) &= E[Y^2|\eta, \alpha] - E[Y|\eta, \alpha]^2 = \alpha^2 [\Gamma(1 + 2/\eta) - \Gamma(1 + 1/\eta)^2] \\
S(y|\eta, \alpha) &= \int_y^\infty \eta \alpha^{-\eta} x^{\eta-1} \exp\left[-\left(\frac{x}{\alpha}\right)^\eta\right] dx = \int_y^\infty \frac{\eta}{\alpha} \left(\frac{x}{\alpha}\right)^{\eta-1} \exp\left[-\left(\frac{x}{\alpha}\right)^\eta\right] dx \\
&= \int_{\left(\frac{y}{\alpha}\right)^\eta}^\infty \exp(-t) dt, \quad \text{substituting } t = \left(\frac{x}{\alpha}\right)^\eta, dt = \frac{\eta}{\alpha} \left(\frac{x}{\alpha}\right)^{\eta-1} dx, \\
&= \exp\left[-\left(\frac{y}{\alpha}\right)^\eta\right], \\
h(y|\eta, \alpha) &= \frac{p(y|\eta, \alpha)}{S(y|\eta, \alpha)} = \eta \alpha^{-\eta} y^{\eta-1}.
\end{aligned}$$

When the parameter  $\eta = 1$ , the Weibull distribution reduces to the exponential distribution with rate  $\lambda = 1/\alpha$ .

- (f) Is the Weibull distribution with unknown parameters  $\eta, \alpha$  a member of the exponential family? What are the implications for inference?

**Solution:** If  $\eta$  and  $\alpha$  are unknown (which is the case here), we don't have an exponential family and our likelihood is not of linear exponential form. Then, results about consistency of our estimators do not follow as easily. Perhaps most importantly, we are not guaranteed consistency under mean model misspecification as we are in the exponential family case. Also, asymptotic inference may not be accurate for as small of  $n$  as it is for an exponential family because we don't have linearity of the score with respect to  $y$ . Further, there is no guarantee that the likelihood surface is well-behaved, as it is in an exponential family (having a well-behaved likelihood surface makes computation easier).

- (g) For the Weibull model and a random sample of size  $n$  obtain: the log-likelihood, the score and the observed information matrix.



**Solution:** For a random sample of size  $n$ , we have

$$\begin{aligned}
L(\eta, \alpha) &= \prod_{i=1}^n p(y_i|\eta, \alpha) = \eta^n \alpha^{-n\eta} \left( \prod_{i=1}^n y_i \right)^{\eta-1} \exp \left[ - \sum_{i=1}^n \left( \frac{y_i}{\alpha} \right)^\eta \right], \\
l(\eta, \alpha) &= n \log(\eta) - n\eta \log(\alpha) + (\eta - 1) \sum_{i=1}^n \log(y_i) - \sum_{i=1}^n \left( \frac{y_i}{\alpha} \right)^\eta, \\
S_\eta(\eta, \alpha) &= \frac{\partial}{\partial \eta} l(\eta, \alpha) = \frac{n}{\eta} - n \log(\alpha) + \sum_{i=1}^n \log(y_i) - \sum_{i=1}^n \log \left( \frac{y_i}{\alpha} \right) \left( \frac{y_i}{\alpha} \right)^\eta, \\
S_\alpha(\eta, \alpha) &= \frac{\partial}{\partial \alpha} l(\eta, \alpha) = -\frac{n\eta}{\alpha} + \frac{\eta}{\alpha^{\eta+1}} \sum_{i=1}^n y_i^\eta, \\
I_{\eta^2}^{(obs)}(\eta, \alpha) &= -\frac{\partial}{\partial \eta} S_\eta(\eta, \alpha) = \frac{n}{\eta^2} + \sum_{i=1}^n \log^2 \left( \frac{y_i}{\alpha} \right) \left( \frac{y_i}{\alpha} \right)^\eta, \\
I_{\eta\alpha}^{(obs)}(\eta, \alpha) &= -\frac{\partial}{\partial \alpha} S_\eta(\eta, \alpha) = \frac{n}{\alpha} - \frac{\eta}{\alpha} \sum_{i=1}^n \log \left( \frac{y_i}{\alpha} \right) \left( \frac{y_i}{\alpha} \right)^\eta - \frac{1}{\alpha} \sum_{i=1}^n \left( \frac{y_i}{\alpha} \right)^\eta, \\
I_{\alpha^2}^{(obs)}(\eta, \alpha) &= -\frac{\partial}{\partial \alpha} S_\alpha(\eta, \alpha) = -\frac{n\eta}{\alpha^2} + \frac{\eta(\eta+1)}{\alpha^{\eta+2}} \sum_{i=1}^n y_i^\eta = -\frac{n\eta}{\alpha^2} + \frac{\eta(\eta+1)}{\alpha^2} \sum_{i=1}^n \left( \frac{y_i}{\alpha} \right)^\eta.
\end{aligned}$$

- (h) Solve the score equations in order to obtain the maximum likelihood estimators (MLEs). You should obtain a single equation that needs to be numerically solved.

**Solution:** Setting  $S(\eta, \alpha) = 0$ , we first get  $\alpha$  as a function of  $\eta$

$$S_\alpha(\eta, \alpha) = 0 \implies \frac{\eta}{\alpha^{\eta+1}} \sum_{i=1}^n y_i^\eta = \frac{n\eta}{\alpha} \implies \alpha(\eta) = \left( \frac{1}{n} \sum_{i=1}^n y_i^\eta \right)^{1/\eta}.$$

Then we can find the MLE by solving  $S_\eta(\eta, \alpha(\eta)) = 0$  numerically as a function of just  $\eta$ .

- (i) Obtain the MLEs and standard errors for the parameters of the Weibull model, for each of the groups in Table 1.

**Solution:** Table 3 shows the estimates  $\hat{\eta}$  and  $\hat{\alpha}$  along with model-based standard errors for each of the four lengths of carbon fibers. The standard errors were estimated using the observed information.

Length (mm)	$\hat{\eta}$	$\widehat{SE}(\hat{\eta})$	$\hat{\alpha}$	$\widehat{SE}(\hat{\alpha})$
1	10.334	2.350	3.319	0.093
10	21.345	5.054	2.377	0.032
20	9.636	2.311	1.852	0.056
50	13.829	3.213	1.735	0.037

Table 3: MLEs and standard errors for the Weibull parameters  $\eta$  and  $\alpha$  at each length.

## A Code for all of problem 1 (simulation study)

```

### True data generating process
true_data <- function(x, b, n) {
  B0 = 0
  B1 = log(2)
  logmu <- B0+B1*x
  mu <- exp(logmu)
  theta <- rgamma(n,shape=b,rate=b)
  y <- sapply(mu*theta, function(lambda) rpois(1, lambda))
  return(data.frame(x=x, y=y))
}

### For a given b,n: run the experiment.
### Either coverage or length experiment.
### Fixed X!! So use same one for all trials
run_sim <- function(b,n, nTrials) {
  set.seed(1)
  x <- rnorm(n)
  resB0 <- replicate(nTrials, fit_mods(x, b,n,1))
  resB1 <- replicate(nTrials, fit_mods(x, b,n,2))

  beta0 <- apply(resB0, 1, mean, na.rm=TRUE)
  beta1 <- apply(resB1, 1, mean, na.rm=TRUE)

  return(list(beta0 = beta0, beta1 = beta1))
}

```

```

### For a given b,n, complete one trial
### Meaning it generates 1 true dataset given b and n
### and then builds 4 models with corresponding CI.
fit_mods <- function(x, b,n, index) {

  dat <- true_data(x, b,n)

  ##### If a model has convergence issues, we say that it did not cover the correct interval
  try1 <- try({
    mod1 <- glm(y~x, data=dat, family="poisson")
    poissint <- c(mod1$coef[index] - sqrt(vcov(mod1)[index, index])*1.96 , mod1$coef[index]

    resid1 <- (dat$y-mod1$fit)/sqrt(mod1$fit)
    Dmat <- matrix(0,ncol=2,nrow=length(dat$y))
    Dmat <- cbind( mod1$fit, mod1$fit*dat$x)
    V <- diag(mod1$fit)
    invinf <- solve( t(Dmat) %*% solve(V) %*% Dmat )
    ingred <- diag(resid1^2/mod1$fit)
    lambhat <- matrix(0,ncol=2,nrow=2)
    lambhat <- t(Dmat) %*% ingred %*% Dmat
    sandcov <- invinf %*% lambhat %*% invinf
    sandint <- c(mod1$coef[index] - sqrt(sandcov[index, index])*1.96 ,
      mod1$coef[index] + sqrt(sandcov[index, index])*1.96)
  }, silent=TRUE)

  if (class(try1)=="try-error") {
    poissint <- c(5,5)
    sandint <- c(5,5)
  }

  try2 <- try({
    mod2 <- glm(y~x, data=dat, family="quasipoisson")
    qpoissint <- c(mod2$coef[index] - sqrt(vcov(mod2)[index, index])*1.96 ,
      mod2$coef[index] + sqrt(vcov(mod2)[index, index])*1.96)
  }, silent=TRUE)

  if (class(try2)=="try-error") {
    qpoissint <- c(5,5)
  }

```

```

try3 <- try({
  mod3 <- MASS::glm.nb(y~x, data=dat)
  nbint<- c(mod3$coef[index] - sqrt(vcov(mod3)[index, index])*1.96 ,
    mod3$coef[index] + sqrt(vcov(mod3)[index, index])*1.96)
}, silent=TRUE)

if (class(try3)=="try-error") {
  nbint <- c(5,5)
}

ints <- rbind(poissint, qpoissint, nbint, sandint)

if (index==1) {
  success <- ((0 > ints[,1]) & 0 < ints[,2])
} else {
  success <- ((log(2) > ints[,1]) & log(2) < ints[,2])
}
names(success) <- c("Pois", "QPois", "NB", "Sand")
return(success)
}

#### Actually run experiment
B <- c(0.1, 1, 10, 1000)
N <- c(20, 50, 100, 250)
var <- c("B0", "B1")
params <- expand.grid(B, N, var)

results <- cbind(params, 0,0,0,0)
names(results) <- c("b", "n", "var", "Pois", "QPois", "NB", "SAND")

for (p in 1:16) {
  print(p)
  cov <- run_sim(params[p,1], params[p,2], 1000)
  results[p, 4:7] <- cov$beta0
  results[p+16, 4:7] <- cov$beta1
}

#Interpret results and build plots

```

```

reshapeRes <- results %>% reshape(varying=list(4:7), direction="long") %>%
  mutate(method=c("Poisson", "QuasiPoisson", "NB", "Sandwich")[time])
names(reshapeRes)[5] <- "coverage"

```

```

ggplot(data=reshapeRes, aes(x=n, y=coverage, col=method))+geom_line(lwd=1)+
  facet_grid(cols=vars(var), rows=vars(b))+
  geom_hline(yintercept=0.95, col="black", lty=2)

```

## B Code for all of problem 2.

```

length1 <- c(2.247,2.640,2.842,2.908,3.099,3.126,3.245,3.328,3.355,
  3.383,3.572,3.581,3.681)
length10 <- c(1.901,2.132,2.203,2.228,2.257,2.35,2.361,2.396,2.397,
  2.445,2.454,2.454, 2.474)
length20 <- c(1.312,1.314,1.479,1.552,1.700,1.803,1.861,1.865,1.944,1.958,
  1.966,1.997,2.006)
length50 <- c(1.339,1.434,1.549,1.574,1.589,1.613,1.746,1.753,1.764, 1.807,
  1.812,1.840,1.852)

```

### Part B

```

MLEs <- c(1/mean(length1), 1/mean(length10), 1/mean(length20),1/mean(length50))

```

```

n <- 13

```

```

modelSEs <- MLEs/sqrt(13)

```

```

lengthdat <- data.frame(cbind(c(length1,length10,length20,length50), rep(c(1,10,20,50),
  each=13)))

```

```

names(lengthdat) <- c("x", "length")

```

```

ggplot(data=lengthdat, aes(sample=x))+
  geom_qq(distribution="qexp")+
  geom_qq_line(distribution = "qexp", col="red")+
  facet_wrap(vars(length))

```

MLEs

modelSEs

### Part C

```
alphas <- lengthdat %>% group_by(length) %>%  
  summarize(s2 = sd(x)^2, means2=mean(x)^2) %>%  
  mutate(alpha = s2/means2) %>%  
  dplyr::select(alpha)
```

```
quasiSEs <- sqrt(alphas$alpha)*modelSEs
```

### Part D

```
sandwichSEs <- sqrt((n-1)/n)*quasiSEs
```

#### Compile results A-D

```
cbind(MLEs, modelSEs, alphas$alpha, quasiSEs, sandwichSEs)
```

#### Parts E-G

```
weibullScore <- function(eta,y) {  
  n <- length(y)  
  alpha = ((1/n*sum(y^eta))^(1/eta))  
  n/eta - n*log(alpha) + sum(log(y)) - sum(log(y/alpha)*(y/alpha)^eta)  
}
```

```
alpha<- function(eta,y) {  
  mean(y^eta)^(1/eta)  
}
```

```
etahat1 <- uniroot(weibullScore, c(8,12), y=length1, maxiter=5000)$root  
etahat10 <- uniroot(weibullScore, c(8,25), y=length10, maxiter=5000)$root  
etahat20 <- uniroot(weibullScore, c(6,25), y=length20, maxiter=5000)$root  
etahat50 <- uniroot(weibullScore, c(0.01,540), y=length50, maxiter=5000)$root
```

```
alphahat1 <- alpha(etahat1, length1)  
alphahat10 <- alpha(etahat10, length10)
```

```

alphahat20 <- alpha(etahat20, length20)
alphahat50 <- alpha(etahat50, length50)

info <- function(y, alpha, eta) {
  n <- length(y)
  matrix(c(n/eta^2 + sum(log(y/alpha)^2*(y/alpha)^eta),
           n/alpha - eta/alpha*sum(log(y/alpha)*(y/alpha)^eta) -
           1/alpha*sum((y/alpha)^eta),
           n/alpha - eta/alpha*sum(log(y/alpha)*(y/alpha)^eta) -
           1/alpha*sum((y/alpha)^eta),
           eta*(eta+1)/alpha^(2+eta)*sum(y^eta) - eta*n/alpha^2), 2, 2)
}

res <- rbind(
  c(1, etahat1, alphahat1, sqrt(diag(solve(info(length1, alphahat1, etahat1))))),
  c(10, etahat10, alphahat10, sqrt(diag(solve(info(length10, alphahat10, etahat10))))),
  c(20, etahat20, alphahat20, sqrt(diag(solve(info(length20, alphahat20, etahat20))))),
  c(50, etahat50, alphahat50, sqrt(diag(solve(info(length50, alphahat50, etahat50))))
)
colnames(res) <- c("length", "eta-hat", "alpha-hat", "se(etahat)", "se(alphahat)")
res

```