STAT/BIOST 571: Homework 1 Solutions

Winter 2023

Problem 1 (35 points)

Investigate the impact of correlation (among observations) on the performance of linear regression. Specifically, conduct a series of simulation experiments in which there are m individuals who each have n observations. For simplicity, you may make the outcomes normally distributed with common correlation ρ (i.e. compound symmetry under a random intercept model) and assume that there is only a single independent variable. Things to consider include different values of range of m, n, and ρ values. Then assess:

1. *Type I error (you can use the usual Wald p-values): note that this model should be under the null*
2. *Bias and coverage of 95% confidence intervals under some alternative model*

Note that the mvtnorm package can be used to simulate multivariate normal data. Please summarize your findings using appropriate figures or tables with accompanying paragraph describing results. This should be similar to the simulation section of a research paper (albeit shorter and simpler).

Solution:

One of the assumptions of Linear Regression is the independence of observations, so it is of interest to investigate the impact of correlation among them with a simulation study. For the simulation setting data were generated in a compound symmetry under random intercept model setting, i.e. data for m individuals, each with n observations were generated. Outcomes were considered to be normally distributed, with common correlation ρ for the same individuals, and outcomes from different individuals were considered independent. The residuals were generated based on a standard Normal distribution, and a single covariate was considered, in which for simplicity independent and identically distributed $n \times m$ observations were generated based on a standard Normal distribution.

To verify the impact of the correlation in the performance of the linear regression it was assessed the bias and coverage of 95% confidence interval under an alternative model with intercept $\beta_0 = 5$ and slope $\beta_1 = 2$. Also, the Type I error was obtained under a null model in which both coefficients were zero. The data was simulated 1000 times using the mvrnorm function from R, and the metrics were be evaluated for each combination of m = 5, 10, 25, 50, 100, 200, 500 individuals, with n = 1, 5, 10 observations, with $\rho = 0.1$, 0.2, 0.5, 0.7, 0.9.

In Figure 1 is represented the Bias for the intercept and slope for different values of m,n and ρ. Both coefficients have similar behavior, in which for all scenarios there is not a lot of bias. The higher values of bias are when considering the higher ρ, however it is still not greater than 0.08. Thus, the correlated data does not influence much on the estimation of the coefficients, in fact, when considering more observations for each individual (increasing n) there is more data being considered, leading to smaller bias.

When considering the coverage of the 95% confidence interval, Figure 2 shows that for the β_1 is around 95% for most cases, even with high correlation between observations. Now for the β_0 there is more to observe. First, considering just the values of m we have that when considering more individuals there is a better coverage. Then, for each fixed value of ρ we have that when there is only one observation per individual then there is no correlation, so the coverage is around 95%. As more observations are considered for each individual there is a decrease in the coverage. Also, as the correlation between the observation increases there is also a decrease in coverage.

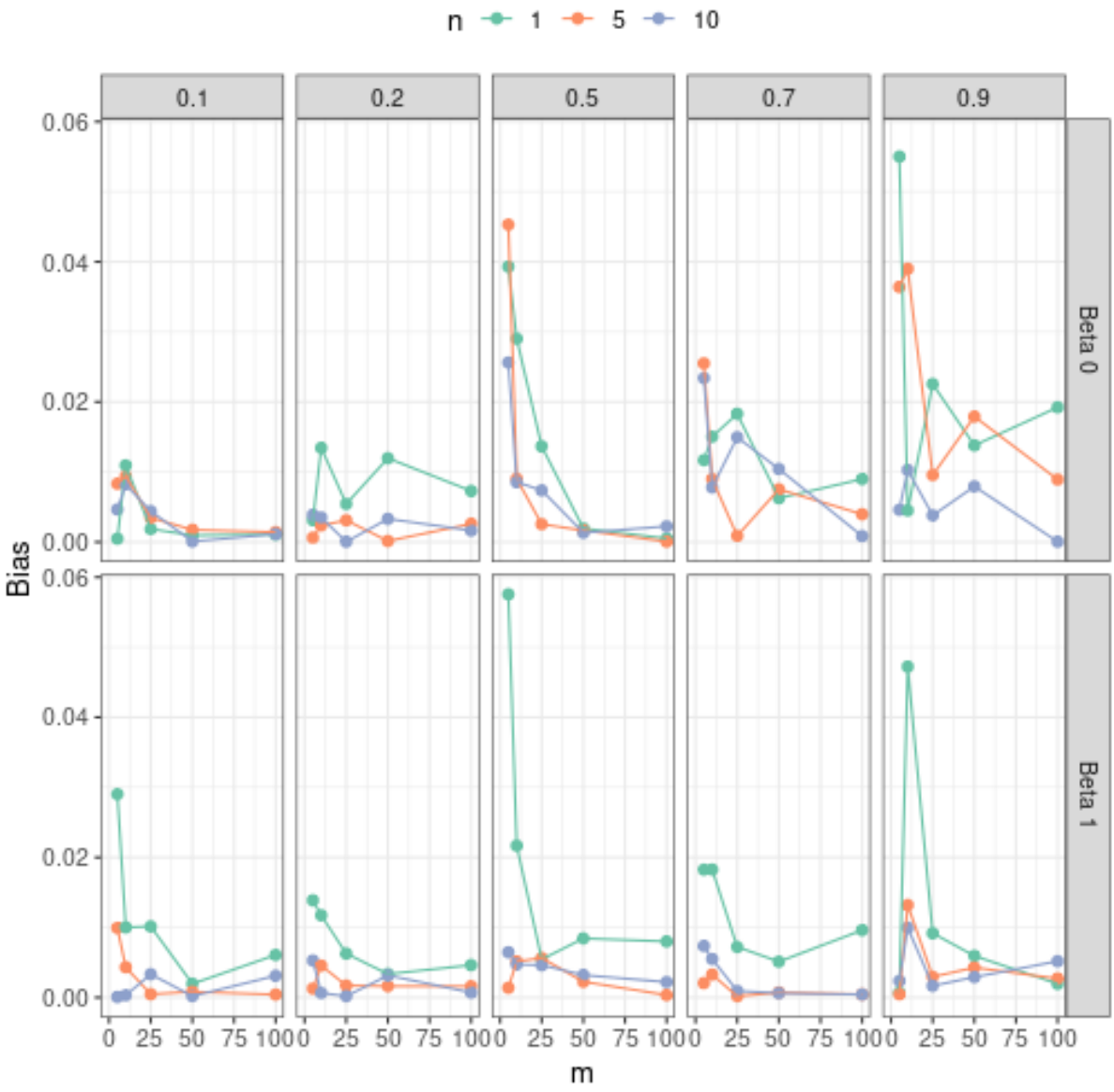

Figure 1: Bias of coefficients for each combination of parameter

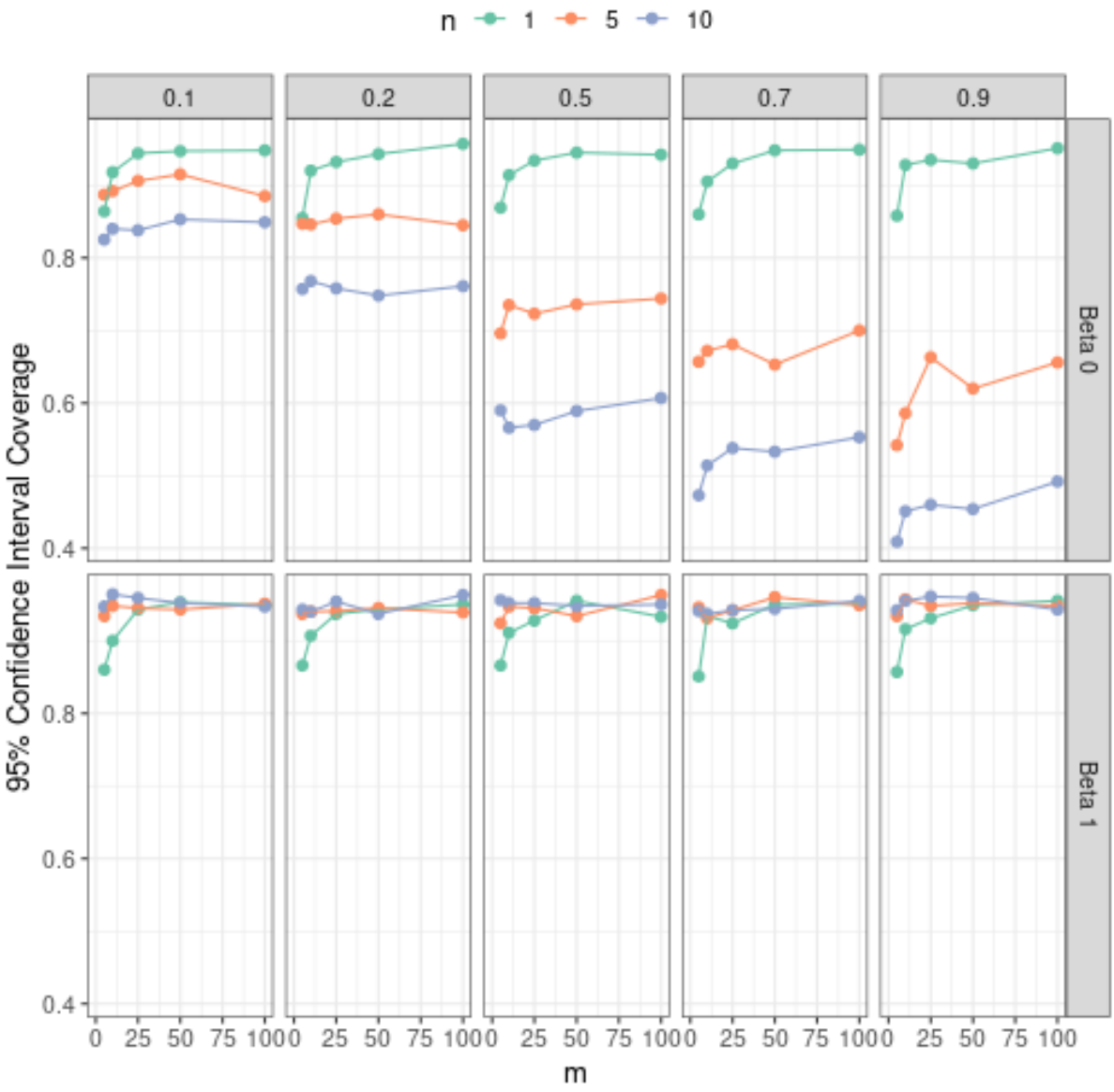

Figure 2: Coverage of 95% Confidence Interval of coefficients for each combination of parameter

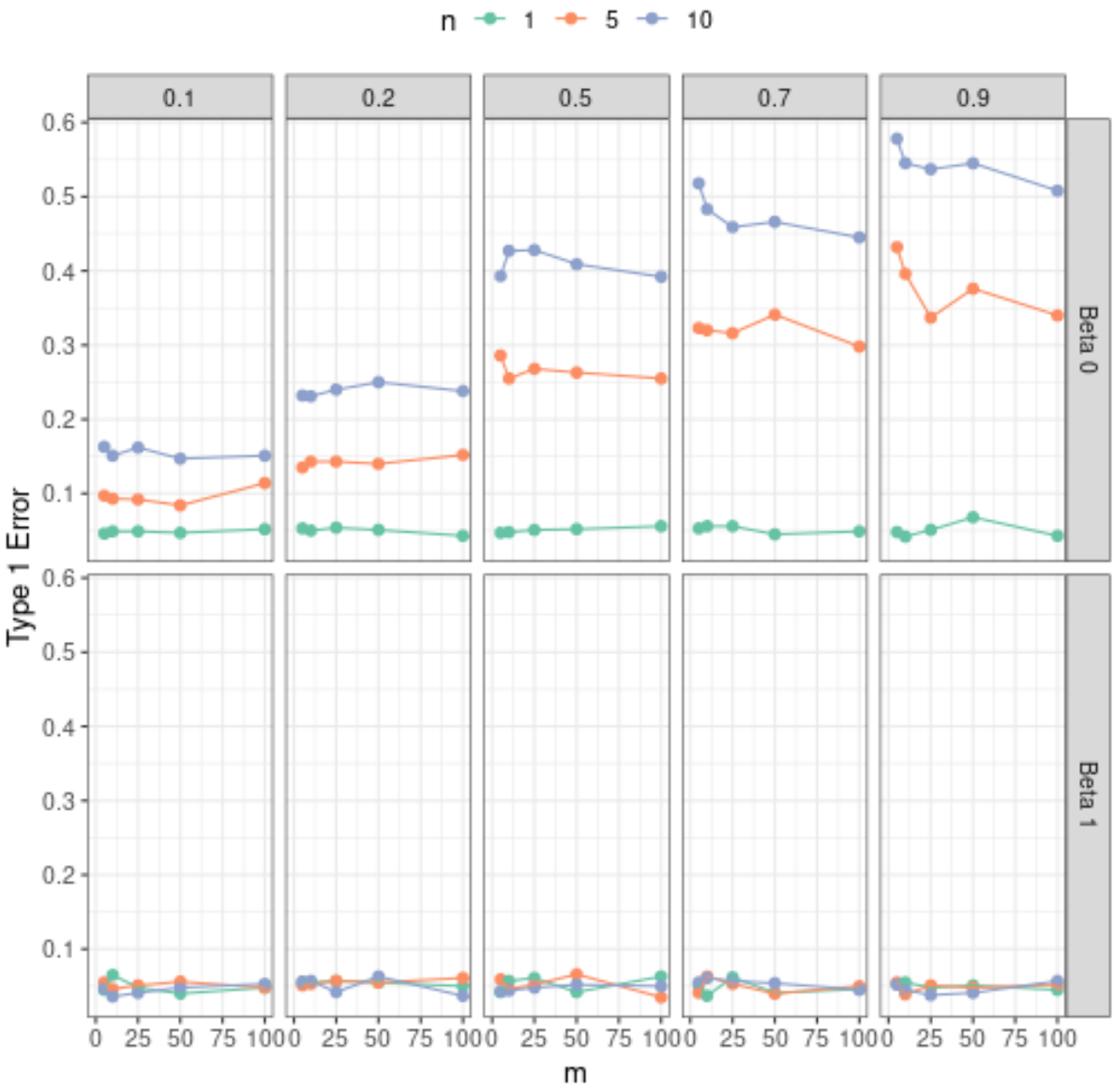

Figure 3: Type 1 Error of coefficients for each combination of parameter

Finally, for the Type 1 error, once again the metric for β_1 was not influenced by correlation on the data. For β_0 when n is one the error is close to 0.05, since the observations are independent. As there is more observations for each individual, or higher ρ the Type 1 error is higher.

In conclusion, correlation among observations does not influence much the estimates of the coefficients, however it impacts the inference of the intercept. More correlated data, either by higher covariance or including more correlated data leads to lower coverage and higher Type 1 error.

Problem 2 (35 points)

Repeat the previous exercise except using a dichotomous outcome and using logistic regression. Note that instead of ρ you may use the probability that any pair of variables are equal to 1 or any other analogous quantity (e.g. ORs). Several different packages can be used to generate multivariate dichotomous outcomes

Solution:

We consider a randomized trial where we would like to estimate and test the treatment effect. More specifically, let there be an even number of m individuals, equally divided to the treatment (trt = 1) and control (trt = 0) group. Each individual was observed n times for an adverse clinical event, with Y = 1 and 0 representing event and non-event, respectively. The data generating mechanism is given by

$$\Pr(Y_{ij} = 1) := p_{ij} = \beta_0 + trt_i\beta, \quad i = 1, ..., m, j = 1, ..., n$$

$$\text{Corr}(Y_{ij}, Y_{i',j'}) = \begin{cases} \rho & \text{if } i = i', \\ 0 & \text{if } i \neq i'. \end{cases}$$

We use the bindata R package to generate correlated binary random variables, which is based on thresholding correlated multivariate Gaussian random variables (essentially a Gaussian copula). It allows direct specification of marginal event probabilities (our p_{ij}'s) and binary correlations (our $\text{Corr}(Y_{ij}, Y_{i'j'})$). Note that not all simulation methods/R packages correspond to the exact same quantities, e.g. simulation based on Gaussian random effects corresponds to conditional rather than marginal event probabilities, and some Gaussian copula-based R packages ask for Gaussian rather than binary correlations.

Two scenarios are investigated based on 200 replicates each:

1. The null $\beta_0 = 0$, $\beta = 0$ under which we assess the type I error rate, i.e. proportion of times where the hypothesis $\beta = 0$ is rejected;
2. An alternative model $\beta_0 = 0$, $\beta = -1$ under which we assess the bias and coverage (proportion of times when the 95% confidence interval of β contains the true β).

and we consider settings where m = 10, 50, 100, n = 3,10,20, ρ = 0,0.1,0.2,. . . ,0.9.

Figure 4 shows how the type I error rate varies with different m, n and ρ under the null. Each panel corresponds to a cluster size n. We observe that:

- When there is no correlation, logistic regression achieves the nominal 0.05 type I error rate as expected. The type I error rate grows as ρ increases;
- When correlation is present, the type I error rate is higher with larger cluster size n. This makes sense because we have independent clusters, and a small cluster size implies that the impact of within-cluster correlation is relatively small;
- We do not see a clear pattern for the relationship between type I error rate and m. For a very small m the result may be less stable, but once we have a moderate sample size, our estimates converge reasonably well to the target (no matter it is close to the truth or not).

Figures 5 and 6 show the bias and coverage with different m,n,ρ under the alternative model. We observe that

- When m is sufficiently large, the bias of $\hat{\beta}$ remains close to 0 for all n as ρ increases. The magnitude of bias increases with ρ for small m, because larger correlation translates to a smaller effective sample size;
- Logistic regression achieves 95% coverage when $\rho = 0$. Ignoring the correlation leads to invalid inference of β, as reflected by decreasing coverage as ρ increases. Similar to our observation with type I error rate, the trend remains similar for different m as long as it is not too small. The impact of correlation on the inference of β is more obvious when the cluster size n is large.

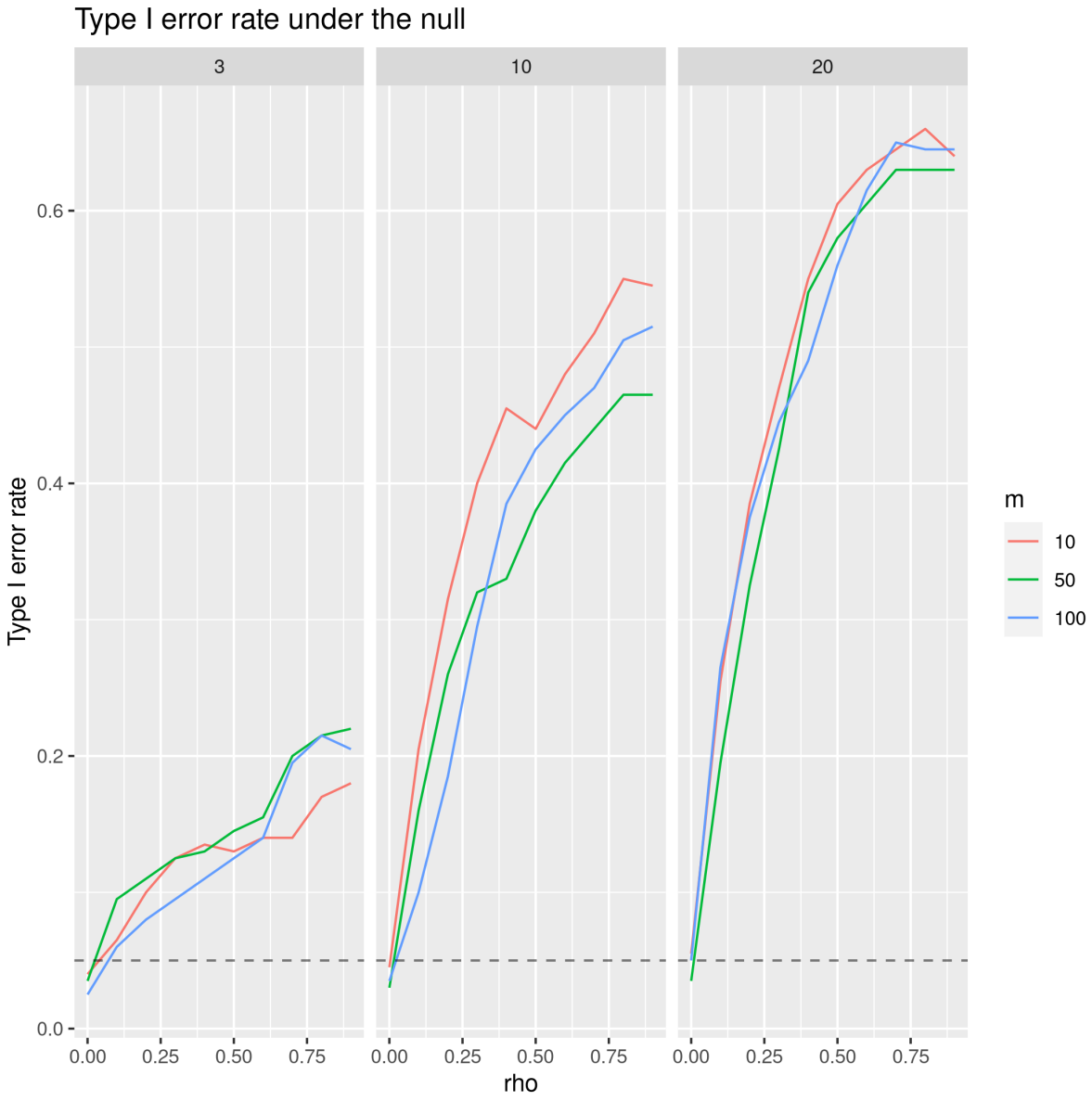

Figure 4: Type I error rate under the null

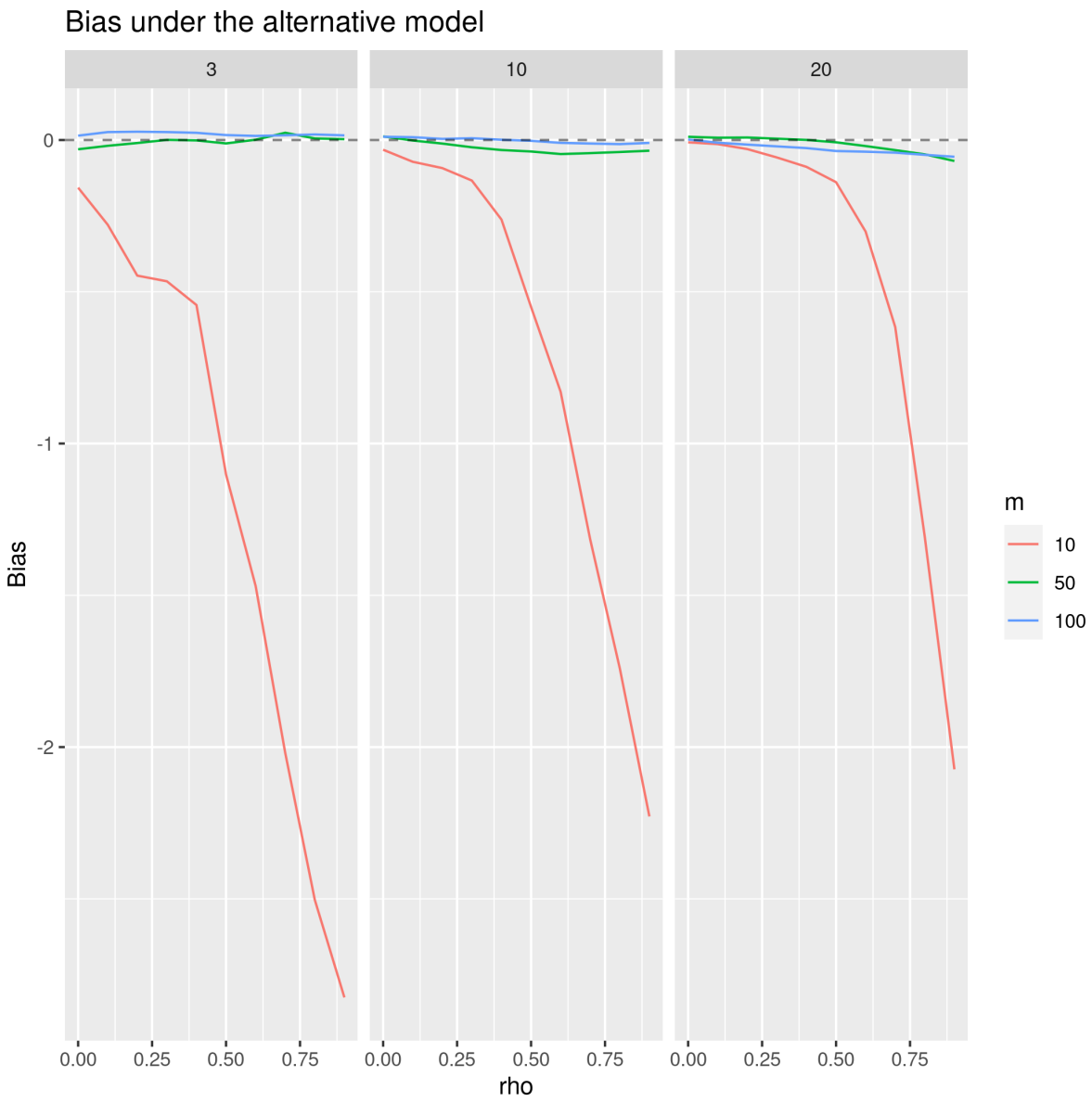

Figure 5: Bias of under the alternative model

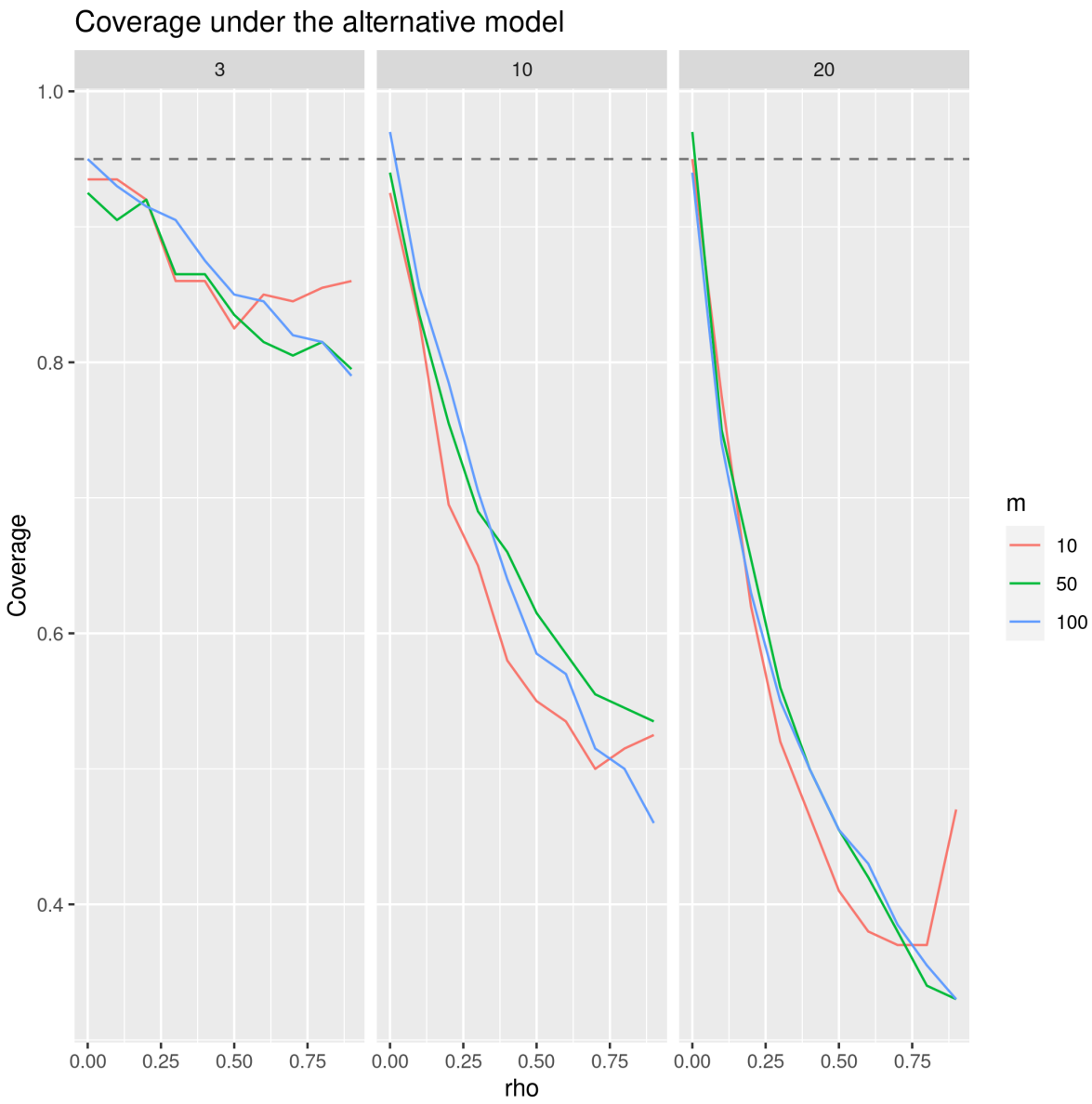

Figure 6: 95 % coverage under the alternative model

Problem 3 (30 points)

Sample discussion:

Practicality: Using the averaged oucomes is much simpler to implement since we end up using the methods from 570. Also, there is no need to consider the random effect terms, from the estimation is more stable with less assumptions. However, with an incomplete data, the subjects with missing values will have to be dropped. Or, the model needs to consider the missingness, which would add onto the complexity of the analysis. Moreover, we cannot estimate the trend along the time points if we aggregate the outcomes repeatedly measured over time elapse.

Validity: First, we will still obtain an unbiased estimate. Given that there is very small within-subject correlation, the variability of the random effect term will be negligible. In such setting, averaging the outcomes will be valid, since using an approach for independent data sufficiently reflects the true underlying trend. However, when there exists within-subject correlation, the fit may accompany high standard error which leads to loss in efficiency. Also, with incomplete data, we would leave out subjects with missingness. This could lead to selection bias, which makes the analysis less valid.

Power: We might end up losing power from the reduced number of observations. In order to minimize the loss, there needs to be large number of subjects in the study. Also, as discussed in Validity, ignoring the presence of the random effects may lead to increased standard error which leads to smaller power.

This type of analysis is appropriate to use when we have complete data; there is high signal to noise ratio; and the data consists of a large number of subjects. The analysis will be nearly optimal when there is very small within-subject correlation. In such setting, the variance of the random effect term will be negligible, and considering one less parameter may lead to a more efficient estimate.

Appendix: R code

Problem 1

```
library(ggplot2)
library(dplyr)
library(tidyr)

params <- expand.grid(m = c(5,10,25,50,100,200), # individuals
                      n = c(1,5,10), # observations per individual
                      rho = c(0.1,0.2,0.5,0.7,0.9), # outcome correlation
                      sigma2 = 1, # error variance
                      beta0 = 5,
                      beta1 = 2 # true beta
)
nrep <- 1000

simulation <- do.call(rbind, lapply(c(1:nrow(params)), function(i){
  m <- params$m[i]
  n <- params$n[i]
  rho <- params$rho[i]
  sigma2 <- params$sigma2[i]
  beta0 <- params$beta0[i]
  beta1 <- params$beta1[i]

  #Getting value of theta
  theta <- (rho*sigma2)/(1-rho)

  #Covariance matrix for each individual
```

```
  indmat <- matrix(rep(theta, n*n), ncol = n)
  diag(indmat) <- rep(theta+sigma2, n)

  #Covariance matrix for all individuals
  covmat <- kronecker(diag(m), indmat)
  ## This was originally rept, and I changed it to nrep
  do.call(rbind, lapply(c(1:nrep), function(nrep){

    #Generating data
    random <- MASS::mvrnorm(1, mu = rep(0,n*m), Sigma = covmat)
    x <- rnorm(n*m, mean = 0, sd = 1)

    #Under null of beta=0
    ynull <-  random
    modnull <- lm(ynull ~ x)

    #Under alternative
    yalt <-  beta0 + x*beta1 + random
    modalt <- lm(yalt ~ x)

    data.frame(m = m, n = n, rho = rho, sigma2 = sigma2,
               truebeta0 = beta0, truebeta1 = beta1,
               beta_0 = summary(modalt)$coefficients[1,1],
               sd_0 = summary(modalt)$coefficients[1,2],
               beta_1 = summary(modalt)$coefficients[2,1],
               sd_1 = summary(modalt)$coefficients[2,2],
               t1err_0 = ifelse(summary(modnull)$coefficients[1,4] < 0.05, 1, 0),
               t1err_1 = ifelse(summary(modnull)$coefficients[2,4] < 0.05, 1, 0))
  }))

}))

results_simulation <- simulation %>% na.omit %>%
  mutate_all(as.numeric) %>%
  mutate(liminf0 = beta_0 - qnorm(0.975)* sd_0,
         limsup0 = beta_0 + qnorm(0.975)* sd_0,
         liminf1 = beta_1 - qnorm(0.975)* sd_1,
         limsup1 = beta_1 + qnorm(0.975)* sd_1) %>%
  mutate(coverage0 = ifelse(liminf0 < truebeta0 & limsup0 > truebeta0, 1, 0),
         coverage1 = ifelse(liminf1 < truebeta1 & limsup1 > truebeta1, 1, 0)) %>%
  group_by(m, n, rho, sigma2, truebeta0, truebeta1) %>%
  summarise(beta_0 = mean(beta_0),
            sd_0 = mean(sd_0),
            coverage_0 = mean(coverage0),
            t1err_0 = mean(t1err_0),
            beta_1 = mean(beta_1),
            sd_1 = mean(sd_1),
            coverage_1 = mean(coverage1),
            t1err_1 = mean(t1err_1))

#Problem1_Bias
results_simulation %>% mutate(bias_0 = abs(beta_0 - as.numeric(truebeta0)),
                              bias_1 = abs(beta_1 - as.numeric(truebeta1))) %>%
```

```
  select(!c(beta_0, sd_0, beta_1, sd_1)) %>%
  pivot_longer(coverage_0:bias_1, names_to = c("Metric","Coef"), values_to = "Values",
               names_sep = "_") %>%
  mutate(Metric = factor(Metric, levels = c("bias","coverage","t1err"),
                         labels = c("Bias","Coverage","Type 1 Error")),
         Coef = factor(Coef, levels = c(0,1), labels = c("Beta 0","Beta 1")),
         n = as.factor(n)) %>% filter(Metric == "Bias") %>%
  ggplot(aes(x = m, y = Values, col = n)) +
  geom_point(size=2)+ geom_line()+
  scale_colour_manual(values = palette.colors(n=3,palette = "Set2"))+
  scale_x_continuous(limits = c(0,100))+
  facet_grid(Coef~rho)+ labs(y = "Bias")+
  theme_bw()+ theme(legend.position = "top",
                    text = element_text(size=13))

#Problem1_Coverage
results_simulation %>% mutate(bias_0 = abs(beta_0 - as.numeric(truebeta0)),
                              bias_1 = abs(beta_1 - as.numeric(truebeta1))) %>%
  select(!c(beta_0, sd_0, beta_1, sd_1)) %>%
  pivot_longer(coverage_0:bias_1, names_to = c("Metric","Coef"), values_to = "Values",
               names_sep = "_") %>%
  mutate(Metric = factor(Metric, levels = c("bias","coverage","t1err"),
                         labels = c("Bias","Coverage","Type 1 Error")),
         Coef = factor(Coef, levels = c(0,1), labels = c("Beta 0","Beta 1")),
         n = as.factor(n)) %>% filter(Metric == "Coverage") %>%
  ggplot(aes(x = m, y = Values, col = n)) +
  geom_point(size=2)+ geom_line()+
  scale_colour_manual(values = palette.colors(n=3,palette = "Set2"))+
  scale_x_continuous(limits = c(0,100))+
  facet_grid(Coef~rho)+ labs(y = "95% Confidence Interval Coverage")+
  theme_bw()+ theme(legend.position = "top",
                    text = element_text(size=13))
#Problem1_T1Err
results_simulation %>% mutate(bias_0 = abs(beta_0 - as.numeric(truebeta0)),
                              bias_1 = abs(beta_1 - as.numeric(truebeta1))) %>%
  select(!c(beta_0, sd_0, beta_1, sd_1)) %>%
  pivot_longer(coverage_0:bias_1, names_to = c("Metric","Coef"), values_to = "Values",
               names_sep = "_") %>%
  mutate(Metric = factor(Metric, levels = c("bias","coverage","t1err"),
                         labels = c("Bias","Coverage","Type 1 Error")),
         Coef = factor(Coef, levels = c(0,1), labels = c("Beta 0","Beta 1")),
         n = as.factor(n)) %>% filter(Metric == "Type 1 Error") %>%
  ggplot(aes(x = m, y = Values, col = n)) +
  geom_point(size=2)+ geom_line()+
  scale_colour_manual(values = palette.colors(n=3,palette = "Set2"))+
  scale_x_continuous(limits = c(0,100))+
  facet_grid(Coef~rho)+ labs(y = "Type 1 Error")+
  theme_bw()+ theme(legend.position = "top",
                    text = element_text(size=13))
```

Problem 2

```
library(bindata)
library(parallel)
library(dplyr)
library(ggplot2)

sim.one = function(m, n, rho, b, seed){
  set.seed(seed)
  X.trt = cbind(rep(1,n), rep(1,n))
  X.ctrl = cbind(rep(1,n), rep(0,n))
  p.trt = exp(X.trt%*%b) / (1 + exp(X.trt%*%b))
  p.ctrl = exp(X.ctrl%*%b) / (1 + exp(X.ctrl%*%b))
  corr = matrix(rho, n, n) + diag(1-rho, n, n)
  y.trt = rmvbin(m/2, margprob = p.trt, bincorr = corr)
  y.ctrl = rmvbin(m/2, margprob = p.ctrl, bincorr = corr)

  y = c(t(rbind(y.trt, y.ctrl)))
  X = rep(c(1,0), each = n*m/2)
  mod = glm(y ~ X, family = 'binomial')
  t1e = bias = cover = NA
  ci = confint.default((mod), level=0.95)
  if (b[2] == 0){
    t1e = ci[2,1] > 0 | ci[2,2] < 0
  } else{
    bias = mod$coefficients[2] - b[2]
    cover = ci[2,1] < b[2] & ci[2,2] >= b[2]
  }
  return(list(t1e = t1e, bias = bias, cover = cover))

}

m = c(10, 50, 100)
n = c(3, 10, 20)
rho = seq(0, .9, .1)
seed = 1:200
pars = expand.grid(m, n, rho, seed)

# if parallel computing and/or the number of threads is not supported on your computer,
# you could replace mcmapply with mapply and/or reduce mc.cores
rslt.null = mcmapply(sim.one, m = pars[,1], n = pars[,2],
                     rho = pars[,3], seed = pars[,4],
                     MoreArgs = list(b = rep(0, 2)), mc.cores = 16)

rslt.alt = mcmapply(sim.one, m = pars[,1], n = pars[,2],
                    rho = pars[,3], seed = pars[,4],
                    MoreArgs = list(b = c(0, -1)), mc.cores = 16)

summary.null = cbind(pars, unlist(rslt.null[1,]))

colnames(summary.null) = c("m", "n", "rho", "seed", "t1e")

summary.null = summary.null %>% group_by(m, n, rho) %>%
  summarize(t1er = mean(t1e), valid_sims = n()) %>%
  mutate(m = as.factor(m)) %>%
```

```
  filter(valid_sims == length(seed))

ggplot(data = summary.null, aes(x = rho, y = t1er)) +
  geom_line(aes(color = m)) + facet_wrap(~n) +
  geom_hline(yintercept = .05, alpha = .5, linetype = "dashed") +
  labs(y = "Type I error rate", title = "Type I error rate under the null")

summary.alt = cbind(pars, unlist(rslt.alt[2,]), unlist(rslt.alt[3,]))
summary.alt = summary.alt[complete.cases(summary.alt), ]
colnames(summary.alt) = c("m", "n", "rho", "seed", "bias_each", "cover_each")

summary.alt = summary.alt %>% group_by(m, n, rho) %>%
  summarize(bias = mean(bias_each), coverage = mean(cover_each),
            valid_sims = n()) %>% mutate(m = as.factor(m)) %>%
  filter(valid_sims == length(seed))

ggplot(data = summary.alt, aes(x = rho, y = bias)) +
  geom_line(aes(color = m)) + facet_wrap(~n) +
  geom_hline(yintercept = 0, alpha = .5, linetype = "dashed") +
  labs(y = "Bias", title = "Bias under the alternative model")

ggplot(data = summary.alt, aes(x = rho, y = coverage)) +
  geom_line(aes(color = m)) + facet_wrap(~n) +
  geom_hline(yintercept = .95, alpha = .5, linetype = "dashed") +
  labs(y = "Coverage", title = "Coverage under the alternative model")
```