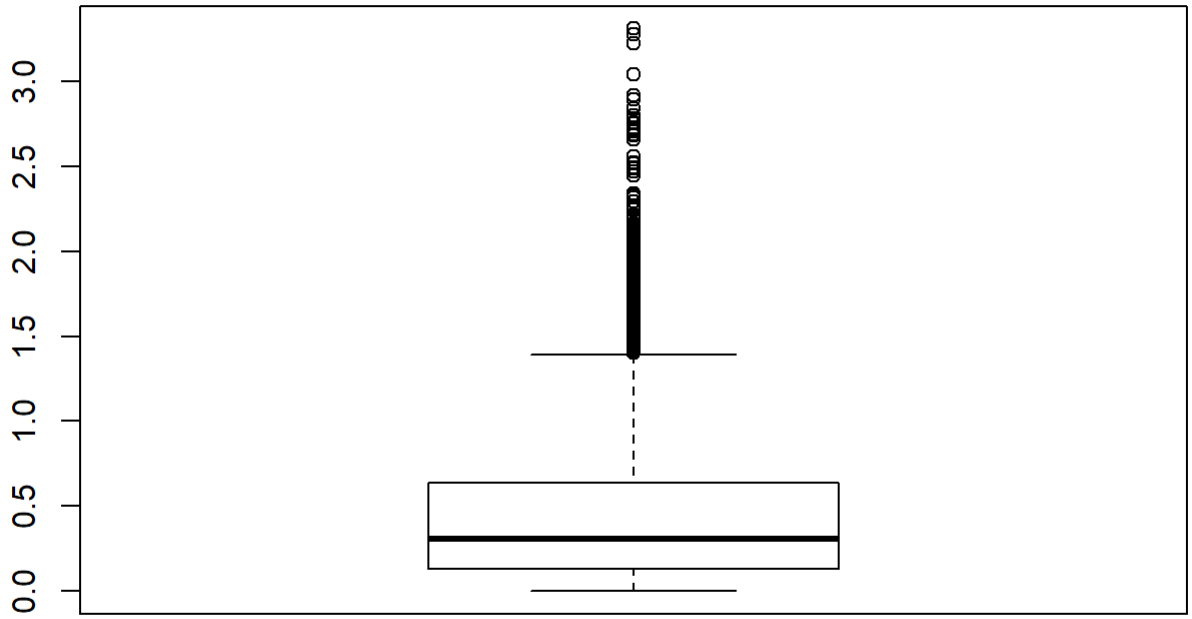
2019 Applied Qual Problem 1

Emily Shives

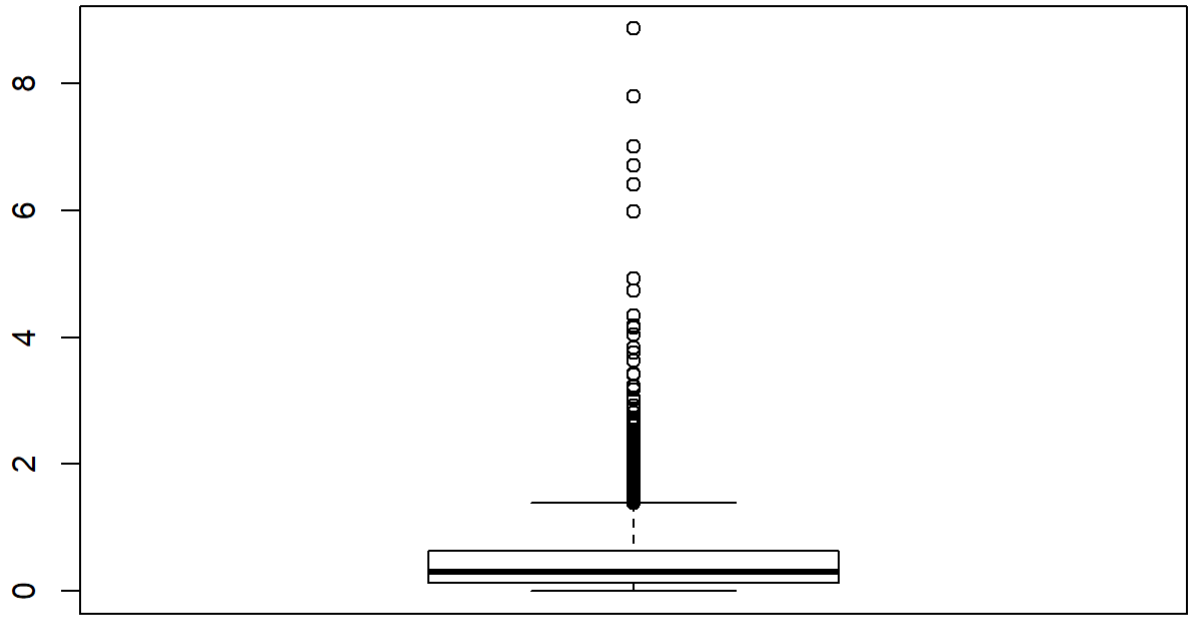
Problem 1

Part a

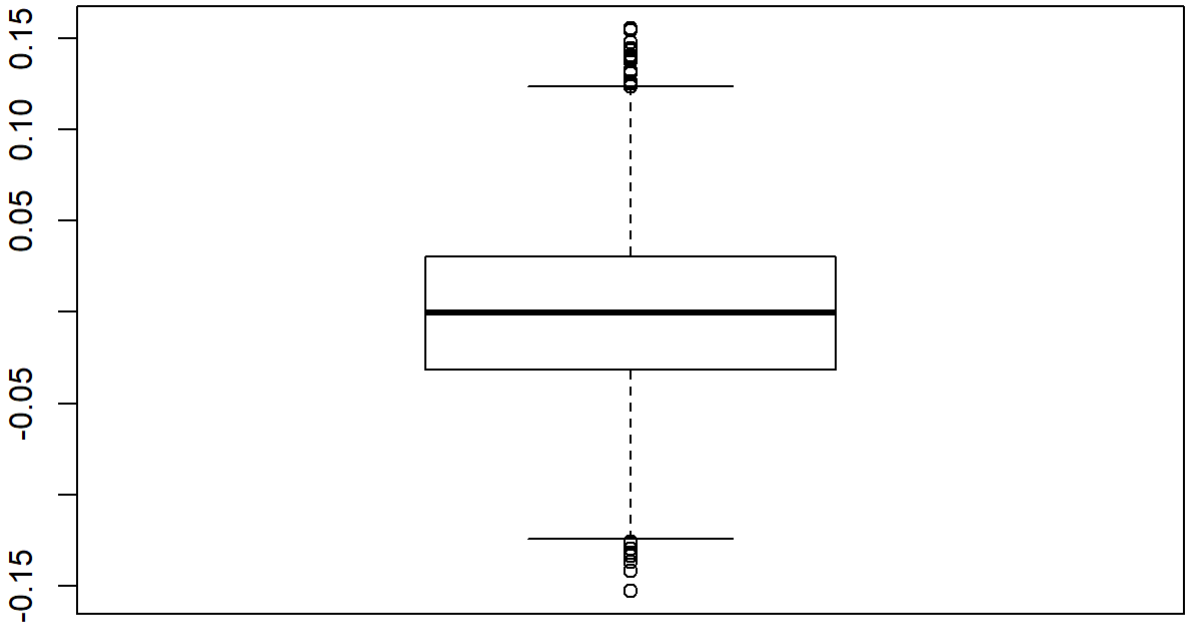
```
# data frame for coefficients and p-values
df <- data.frame(colnames(train)[-c(1,2)])
df <- df %>% add_column(y1_coef = NA, y1_lp = NA, y2_coef = NA, y2_lp = NA)
d<-dim(train)[2]
yhat1 <- rep(0,500)
yhat2 <- rep(0,500)
for (i in 3:d){
  model1 <- lm(train$ROI1 ~ train[[i]])
  p1 <- coef(summary(model1))[2, "Pr(>|t|)"]
  lp1 <- -log(p1,base=10)
  beta1 <- coef(summary(model1))[2,"Estimate"]
  df$y1_coef[i-2] <- beta1
  df$y1_lp[i-2] <- lp1
  model2 <- lm(train$ROI2 ~ train[[i]])
  p2 <- coef(summary(model2))[2, "Pr(>|t|)"]
  beta2 <- coef(summary(model2))[2,"Estimate"]
  lp2 <- -log(p2,base=10)
  df$y2_coef[i-2] <- beta2
  df$y2_lp[i-2] <- lp2
  yhat1 <- yhat1 + beta1*train[[i]]
  yhat2 <- yhat2 + beta2*train[[i]]
}

## Boxplot
boxplot(df$y1_lp)
```



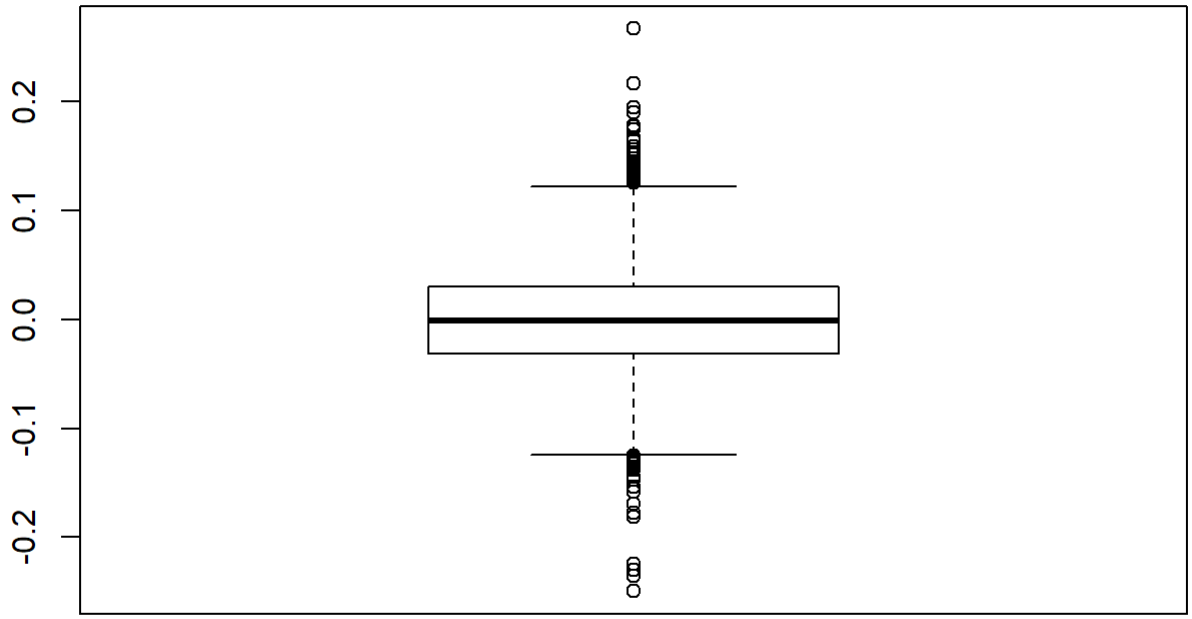

```
boxplot(df$y2_lp)
```



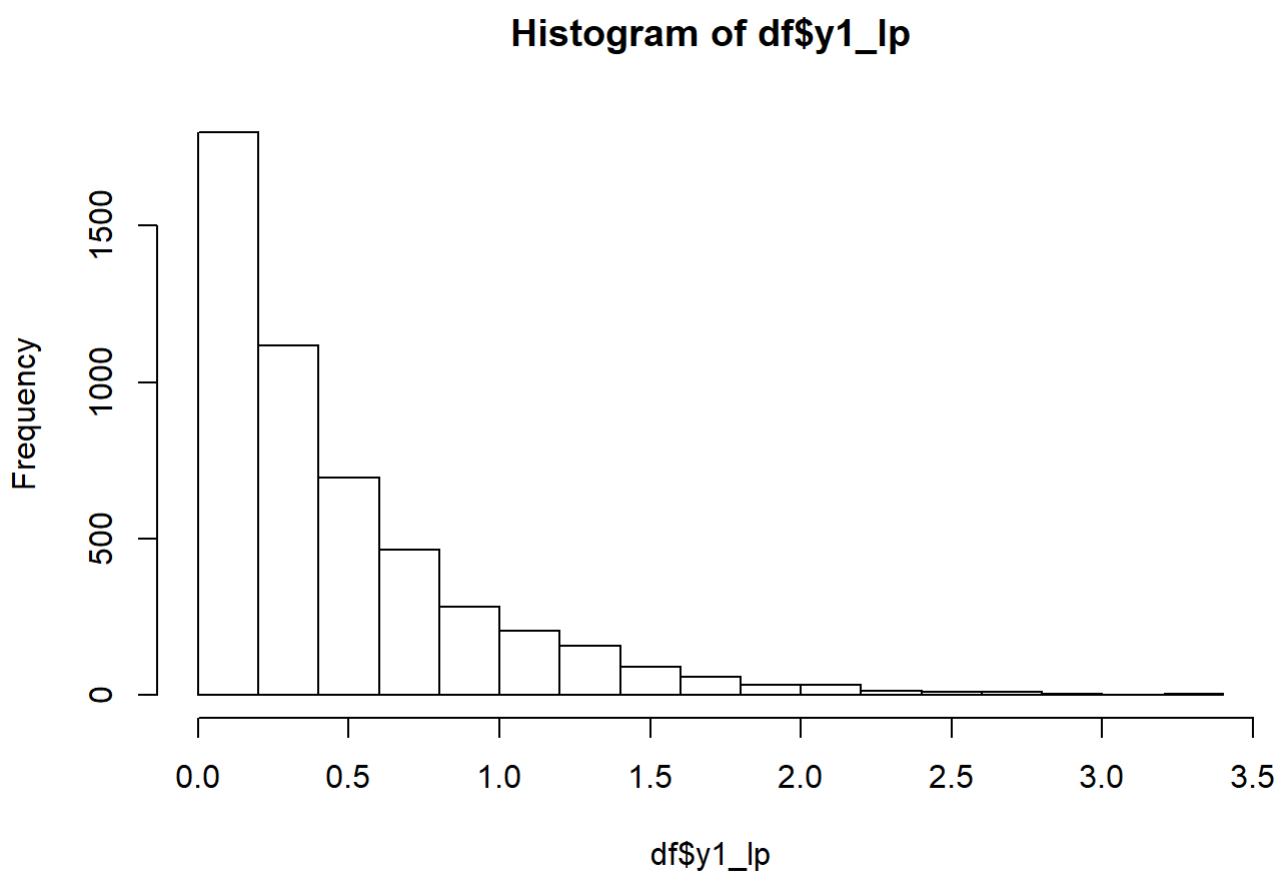

```
boxplot(df$y1_coef)
```



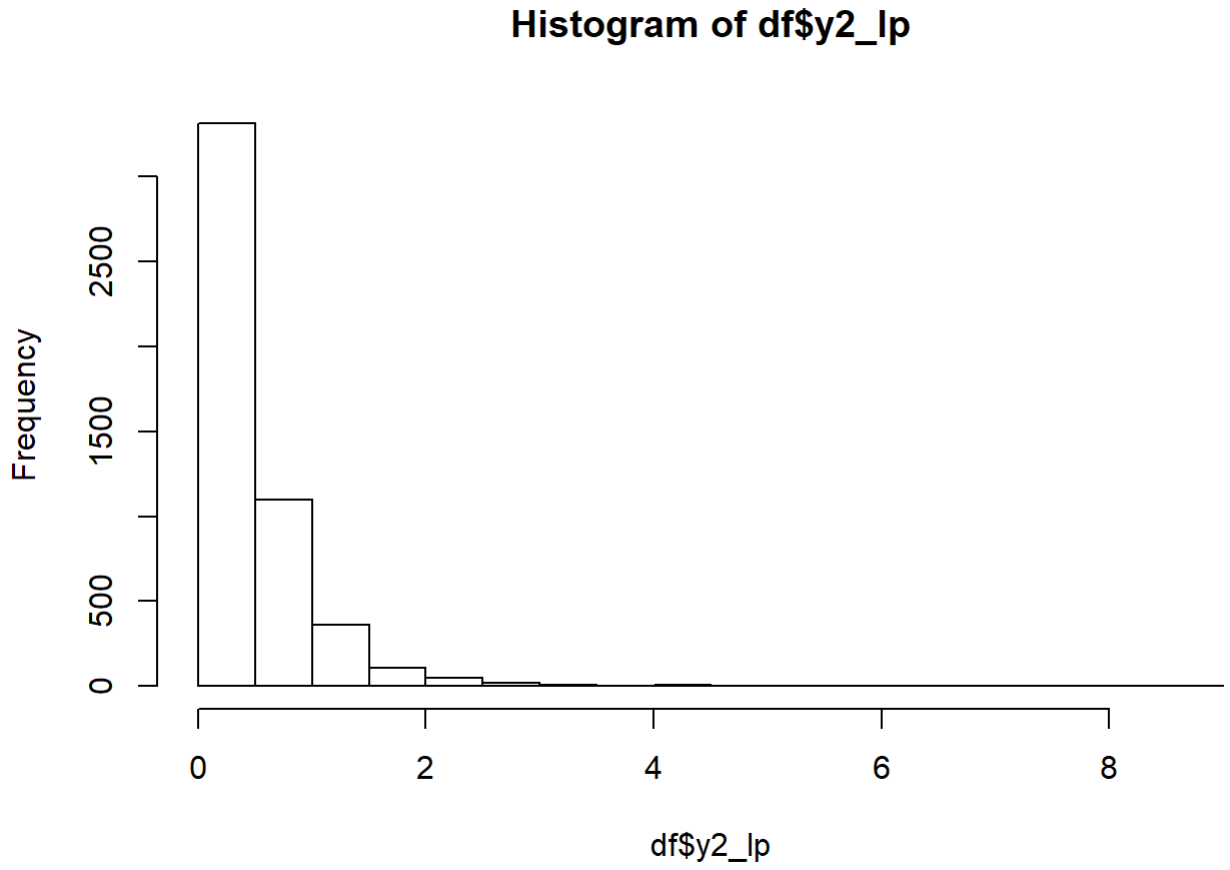

```
boxplot(df$y2_coef)
```



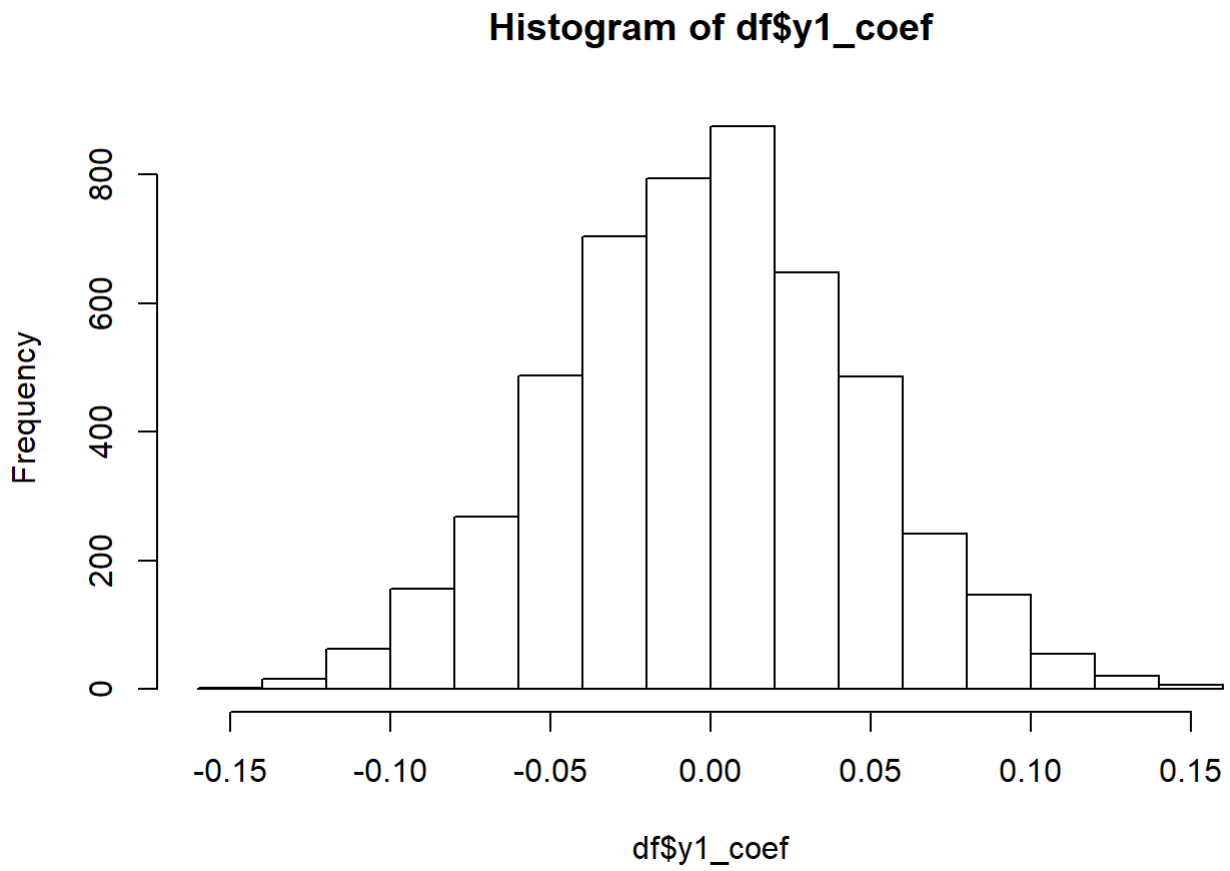

```
## Histogram
hist(df$y1_lp)
```


Histogram of df$y1_lp
Frequency
0
500
1000
1500
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
df$y1_lp

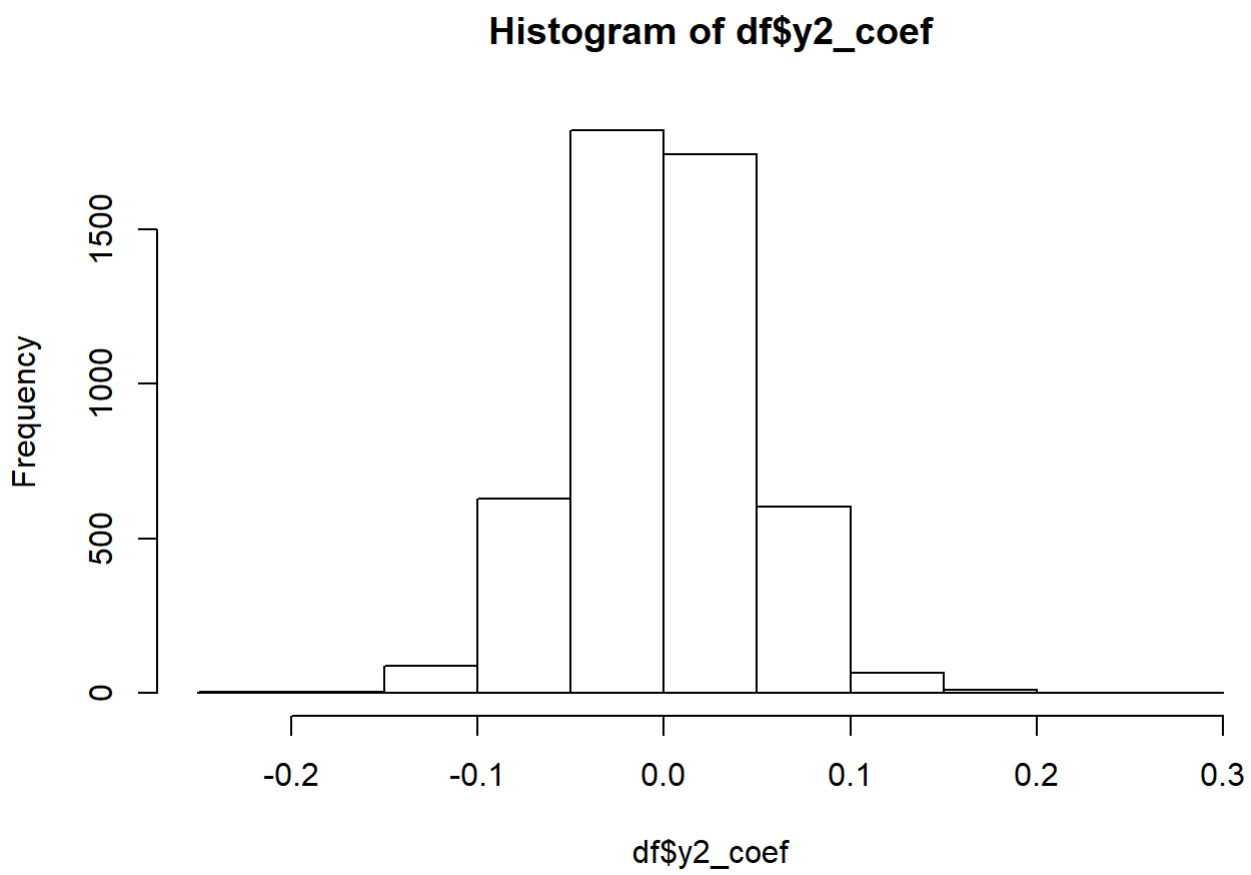
```
hist(df$y2_lp)
```




```
hist(df$y1_coef)
```




```
hist(df$y2_coef)
```




```
## Number of significant covariates
sum(df$y1_lp>-log(0.05,10)) # I think this is the response with a dense signal
```

```
## [1] 313
```

```
sum(df$y2_lp>-log(0.05,10)) # I think this is the response with a sparse signal
```

```
## [1] 293
```

Part b

```
# yhat1 <- as.matrix(train2)%*%as.matrix(df$y1_coef)
# yhat2 <- as.matrix(train2)%*%as.matrix(df$y2_coef)
summary(lm(train$ROI1~yhat1))$r.squared
```

```
## [1] 0.9206125
```

```
summary(lm(train$ROI2~yhat2))$r.squared
```

```
## [1] 0.9142741
```

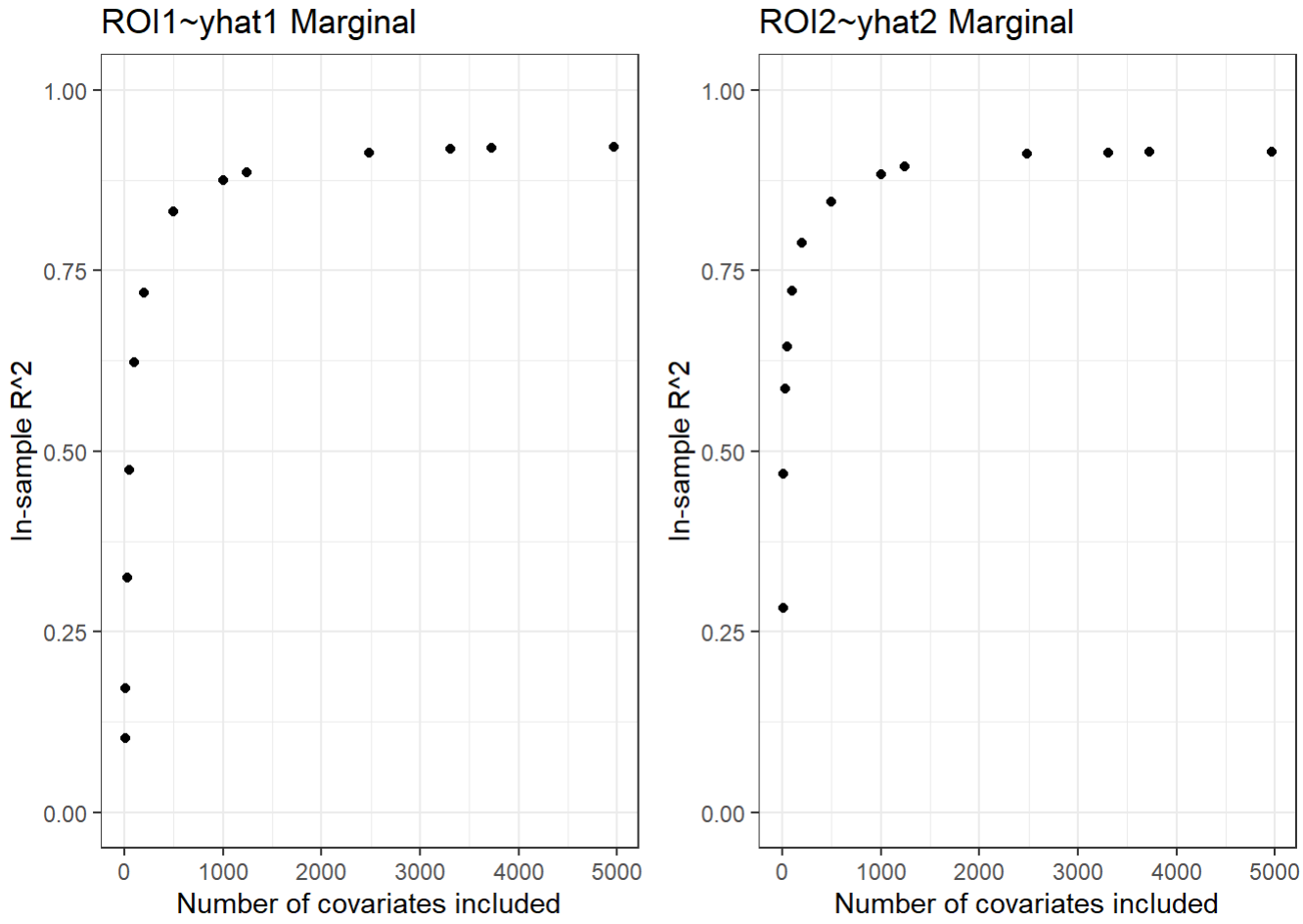
ROI1 has higher R^2 from $\hat{y}_1$ than ROI2 does, which makes sense if I know that I have a dense signal with ROI1.

Part c

```
## sorting
o1 <- order(df$y1_lp, decreasing = TRUE)
ordering1 <- df[o1,1] # not sure if this is what I want for ordering
o2 <- order(df$y2_lp, decreasing = TRUE)
ordering2 <- df[o2,1]
```

Subset different numbers of covariates

```
train2 <- train[,-c(1,2)]
n <- c(d-2,floor((d-2)*3/4),floor((d-2)*2/3),floor((d-2)/2),floor((d-2)/4),1000,500,200,100,50,2
5,10,5)
sub_Rsq1 <- rep(0,length(n))
sub_Rsq2 <- rep(0,length(n))
for (i in 1:length(n)){
  subset1 <- o1[1:n[i]]
  sub_yhat1 <- as.matrix(train2[,subset1])%*%as.matrix(df$y1_coef[subset1])
  fit1 <- lm(train$ROI1~sub_yhat1)
  sub_Rsq1[i] <- summary(fit1)$r.squared
  subset2 <- o2[1:n[i]]
  sub_yhat2 <- as.matrix(train2[,subset2])%*%as.matrix(df$y2_coef[subset2])
  fit2 <- lm(train$ROI2~sub_yhat2)
  sub_Rsq2[i] <- summary(fit2)$r.squared
}
g1<- ggplot()+
  geom_point(aes(x=n,y=sub_Rsq1))+
  theme_bw()+
  labs(x="Number of covariates included",y="In-sample R^2",title="ROI1~yhat1 Marginal")+
  ylim(0,1)
g2<- ggplot()+
  geom_point(aes(x=n,y=sub_Rsq2))+
  theme_bw()+
  labs(x="Number of covariates included",y="In-sample R^2",title="ROI2~yhat2 Marginal")+
  ylim(0,1)
grid.arrange(g1, g2, nrow = 1)
```



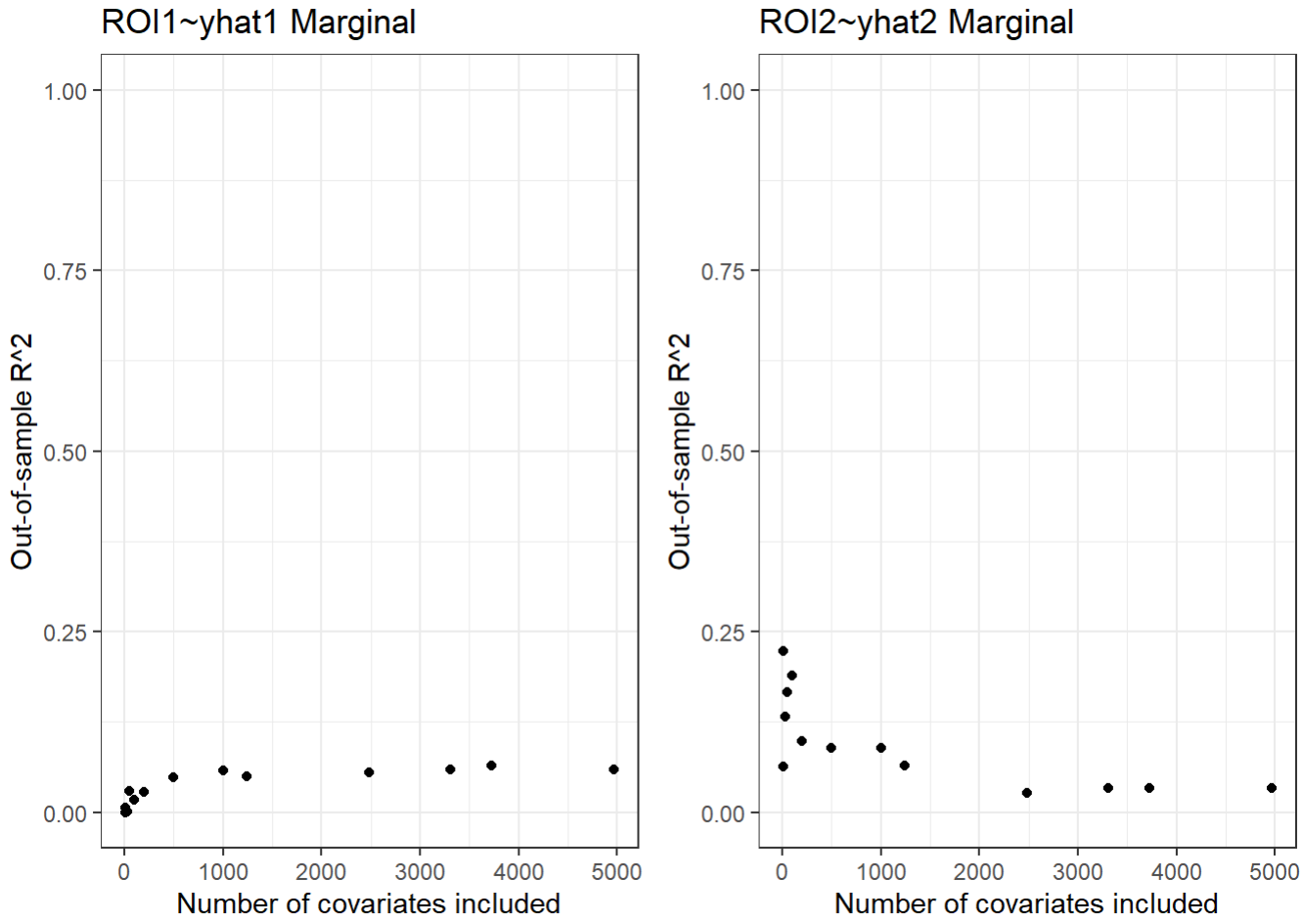
Thresholds are by p-value, with the most significant p covariates included in calculating $\hat{y}$.

Part d

```
test <- read.csv("C:/Users/nick work s-pro4/Documents/Emily/UNC/OldPhDExams/Applied/qual_2019_test_data_0.csv", header = TRUE)
```

```
test2 <- test[,-c(1,2)]
yhat1_test <- as.matrix(test2)%*%as.matrix(df$y1_coef)
yhat2_test <- as.matrix(test2)%*%as.matrix(df$y2_coef)
fit1 <- lm(test$ROI1~yhat1_test)
out_Rsq1 <- summary(fit1)$r.squared

out_Rsq1 <- rep(0,length(n))
out_Rsq2 <- rep(0,length(n))
for (i in 1:length(n)){
  subset1 <- o1[1:n[i]]
  sub_yhat1_test <- as.matrix(test2[,subset1])%*%as.matrix(df$y1_coef[subset1])
  fit1 <- lm(test$ROI1~sub_yhat1_test)
  out_Rsq1[i] <- summary(fit1)$r.squared
  subset2 <- o2[1:n[i]]
  sub_yhat2_test <- as.matrix(test2[,subset2])%*%as.matrix(df$y2_coef[subset2])
  fit2 <- lm(test$ROI2~sub_yhat2_test)
  out_Rsq2[i] <- summary(fit2)$r.squared
}
```

```
g1<- ggplot()+
  geom_point(aes(x=n,y=out_Rsq1))+
  theme_bw()+
  labs(x="Number of covariates included",y="Out-of-sample R^2",title="ROI1~yhat1 Marginal")+
  ylim(0,1)
g2<- ggplot()+
  geom_point(aes(x=n,y=out_Rsq2))+
  theme_bw()+
  labs(x="Number of covariates included",y="Out-of-sample R^2",title="ROI2~yhat2 Marginal")+
  ylim(0,1)
grid.arrange(g1, g2, nrow = 1)
```



Of course the out of sample R^2 values are much smaller than the in-sample R^2 values, because the models were fitted using the training data, and in particular there is overfitting in the models with larger numbers of covariates.

The out of sample R^2 pattern is not the same for y_1 and y_2. Since the y_2 signal is actually sparse, just using the few covariates that have large p-values in the training data actually results in a better fit than using more covariates, which may not actually be related to the response.

Part e

```
# Ridge regression
lambdas <- 10^seq(3, -2, by = -.1) # not necessary to specify, glmnet will pick a set of lambdas
for you
# ROI1
y <- as.vector(train$ROI1)
x <- data.matrix(train2)
fit1R <- glmnet(x, y, alpha = 0, lambda=lambdas)
cv_fit <- cv.glmnet(x, y, alpha = 0, lambda = lambdas)
opt_lambda <- cv_fit$lambda.min
#fit1R <- glmnet(x, y, alpha = 0, lambda = opt_lambda)
y_predicted <- predict(fit1R, s = opt_lambda, newx = x)
coef_1 <- predict(fit1R, s = opt_lambda, newx = x, type = "coef")[-c(1),]

l1 <- lm(y ~ y_predicted)
summary(l1)$r.squared
```

```
## [1] 0.9979068
```

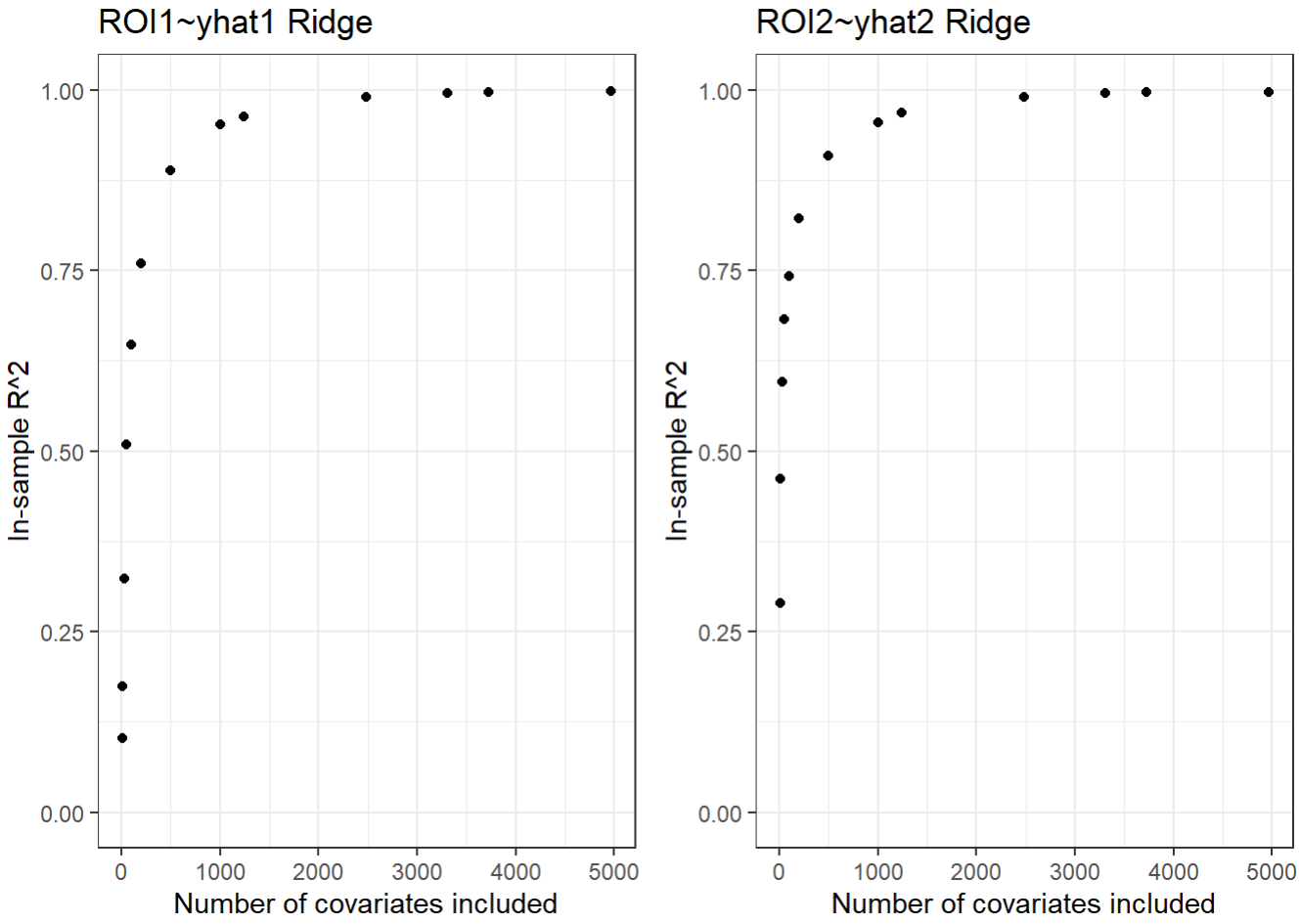
```
#ROI2
y <- as.vector(train$ROI2)
fit2R <- glmnet(x, y, alpha = 0, lambda=lambdas)
cv_fit <- cv.glmnet(x, y, alpha = 0, lambda = lambdas)
opt_lambda <- cv_fit$lambda.min
y_predicted <- predict(fit2R, s = opt_lambda, newx = x)
coef_2 <- predict(fit2R, s = opt_lambda, newx = x, type = "coef")[-c(1),]

l2 <- lm(y ~ y_predicted)
summary(l2)$r.squared
```

```
## [1] 0.99671
```

```
# Ordering by absolute value of coefficients
o1 <- order(abs(coef_1), decreasing = TRUE)
ordering1 <- df[o1,1]
o2 <- order(abs(coef_2), decreasing = TRUE)
ordering2 <- df[o2,1]
```

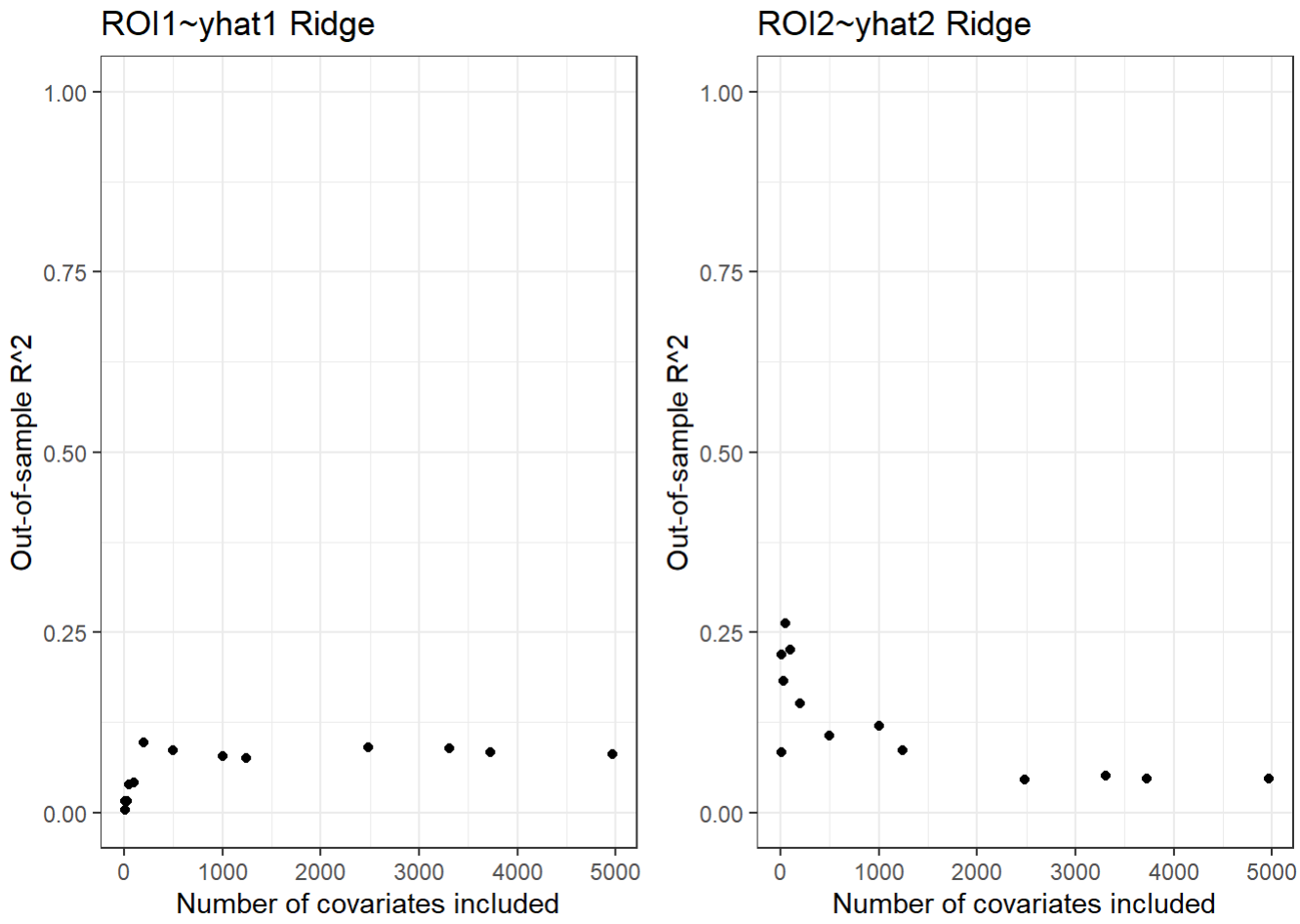
```
# Different subsets
sub_Rsq1 <- rep(0,length(n))
sub_Rsq2 <- rep(0,length(n))
y1 <- as.vector(train$ROI1)
y2 <- as.vector(train$ROI2)
x <- data.matrix(train2)
for (i in 1:length(n)){
  subset1 <- o1[1:n[i]]
  sub_yhat1 <- as.matrix(train2[,subset1])%*%as.matrix(coef_1[subset1])
  l1 <- lm(y1 ~ sub_yhat1)
  sub_Rsq1[i] <- summary(l1)$r.squared
  subset2 <- o2[1:n[i]]
  sub_yhat2 <- as.matrix(train2[,subset2])%*%as.matrix(coef_2[subset2])
  l2 <- lm(y2 ~ sub_yhat2)
  sub_Rsq2[i] <- summary(l2)$r.squared
}
```

```
# Plot the R^2 for the different subsets
g1<- ggplot()+
  geom_point(aes(x=n,y=sub_Rsq1))+
  theme_bw()+
  labs(x="Number of covariates included",y="In-sample R^2",title="ROI1~yhat1 Ridge")+
  ylim(0,1)
g2<- ggplot()+
  geom_point(aes(x=n,y=sub_Rsq2))+
  theme_bw()+
  labs(x="Number of covariates included",y="In-sample R^2",title="ROI2~yhat2 Ridge")+
  ylim(0,1)
grid.arrange(g1, g2, nrow = 1)
```



```
## Out of sample R^2 based on Ridge Regression for different subset sizes
out_Rsq1 <- rep(0,length(n))
out_Rsq2 <- rep(0,length(n))
for (i in 1:length(n)){
  subset1 <- o1[1:n[i]]
  sub_yhat1_test <- as.matrix(test2[,subset1])%*%as.matrix(coef_1[subset1])
  fit1 <- lm(test$ROI1~sub_yhat1_test)
  out_Rsq1[i] <- summary(fit1)$r.squared
  subset2 <- o2[1:n[i]]
  sub_yhat2_test <- as.matrix(test2[,subset2])%*%as.matrix(coef_2[subset2])
  fit2 <- lm(test$ROI2~sub_yhat2_test)
  out_Rsq2[i] <- summary(fit2)$r.squared
}
```

```
## Graphing out of sample R^2 for different subset sizes based on Ridge Regression
g1<- ggplot()+
  geom_point(aes(x=n,y=out_Rsq1))+
  theme_bw()+
  labs(x="Number of covariates included",y="Out-of-sample R^2",title="ROI1~yhat1 Ridge")+
  ylim(0,1)
g2<- ggplot()+
  geom_point(aes(x=n,y=out_Rsq2))+
  theme_bw()+
  labs(x="Number of covariates included",y="Out-of-sample R^2",title="ROI2~yhat2 Ridge")+
  ylim(0,1)
grid.arrange(g1, g2, nrow = 1)
```



The results of Ridge regression are similar in pattern to the results of the marginal estimators, although the values of R^2 are somewhat higher for Ridge regression.

Part f

Below I run LASSO to generate similar results.

```
# LASSO regression
set.seed(100)
lambdas <- 10^seq(3, -2, by = -.1) # not necessary to specify, glmnet will pick a set of lambdas
for you
# ROI1
y <- as.vector(train$ROI1)
x <- data.matrix(train2)
fit1R <- glmnet(x, y, alpha = 1, lambda=lambdas)
cv_fit <- cv.glmnet(x, y, alpha = 1, lambda = lambdas)
opt_lambda <- cv_fit$lambda.min
#fit1R <- glmnet(x, y, alpha = 0, lambda = opt_lambda)
y_predicted <- predict(fit1R, newx = x)
coef_1 <- predict(fit1R, s = opt_lambda, newx = x, type = "coef")[-c(1),]

l1 <- lm(y ~ y_predicted)
summary(l1)$r.squared
```

```
## [1] 0.9990286
```

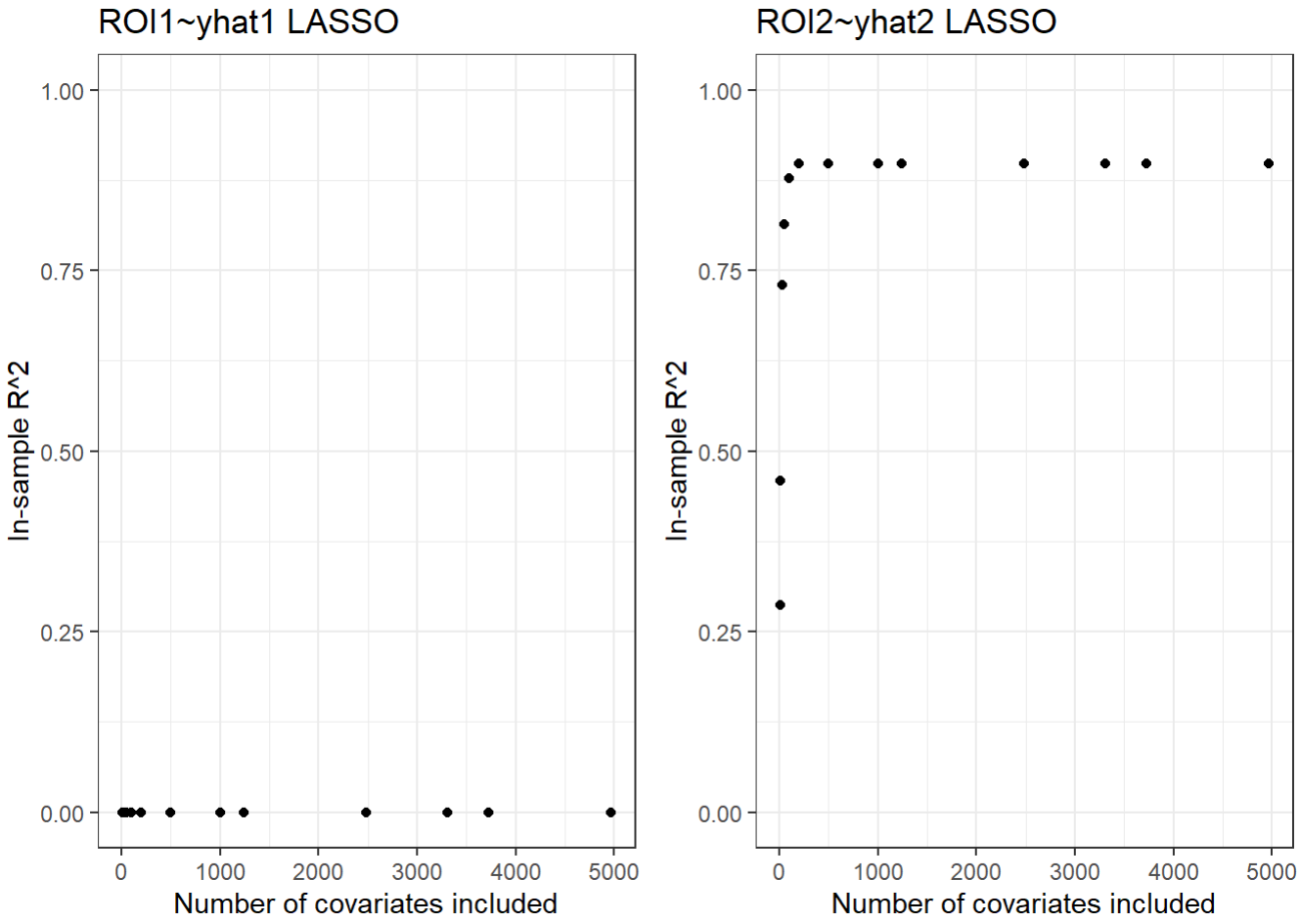
```
#ROI2
y <- as.vector(train$ROI2)
fit2R <- glmnet(x, y, alpha = 1, lambda=lambdas)
cv_fit <- cv.glmnet(x, y, alpha = 1, lambda = lambdas)
opt_lambda <- cv_fit$lambda.min
y_predicted <- predict(fit2R, s = opt_lambda, newx = x)
coef_2 <- predict(fit2R, s = opt_lambda, newx = x, type = "coef")[-c(1),]

l2 <- lm(y ~ y_predicted)
summary(l2)$r.squared
```

```
## [1] 0.8974341
```

```
# Ordering by absolute value of coefficients
o1 <- order(abs(coef_1), decreasing = TRUE)
ordering1 <- df[o1,1]
o2 <- order(abs(coef_2), decreasing = TRUE)
ordering2 <- df[o2,1]
```

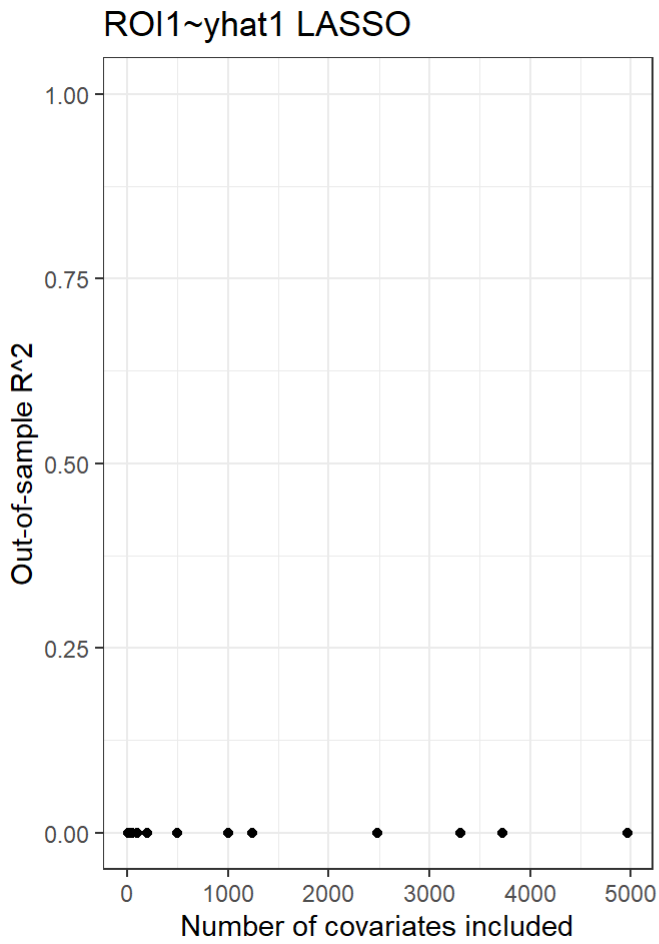
```
# Different subsets
sub_Rsq1 <- rep(0,length(n))
sub_Rsq2 <- rep(0,length(n))
y1 <- as.vector(train$ROI1)
y2 <- as.vector(train$ROI2)
x <- data.matrix(train2)
for (i in 1:length(n)){
  subset1 <- o1[1:n[i]]
  sub_yhat1 <- as.matrix(train2[,subset1])%*%as.matrix(coef_1[subset1])
  l1 <- lm(y1 ~ sub_yhat1)
  sub_Rsq1[i] <- summary(l1)$r.squared
  subset2 <- o2[1:n[i]]
  sub_yhat2 <- as.matrix(train2[,subset2])%*%as.matrix(coef_2[subset2])
  l2 <- lm(y2 ~ sub_yhat2)
  sub_Rsq2[i] <- summary(l2)$r.squared
}
```

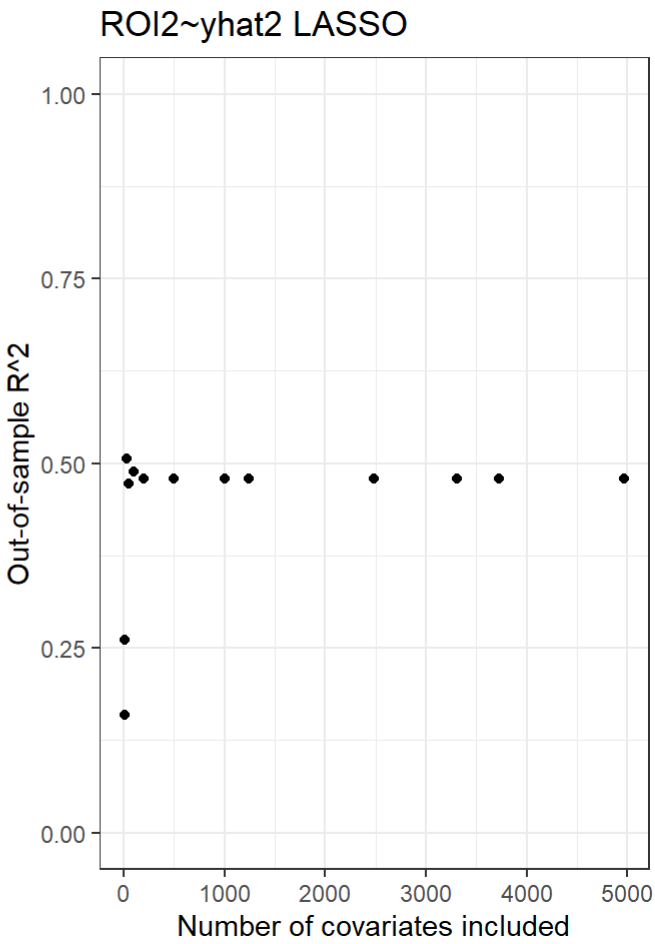
```
# Plot the R^2 for the different subsets
g1<- ggplot()+
  geom_point(aes(x=n,y=sub_Rsq1))+
  theme_bw()+
  labs(x="Number of covariates included",y="In-sample R^2",title="ROI1~yhat1 LASSO")+
  ylim(0,1)
g2<- ggplot()+
  geom_point(aes(x=n,y=sub_Rsq2))+
  theme_bw()+
  labs(x="Number of covariates included",y="In-sample R^2",title="ROI2~yhat2 LASSO")+
  ylim(0,1)
grid.arrange(g1, g2, nrow = 1)
```



```
## Out of sample R^2 based on LASSO Regression for different subset sizes
out_Rsq1 <- rep(0,length(n))
out_Rsq2 <- rep(0,length(n))
for (i in 1:length(n)){
  subset1 <- o1[1:n[i]]
  sub_yhat1_test <- as.matrix(test2[,subset1])%*%as.matrix(coef_1[subset1])
  fit1 <- lm(test$ROI1~sub_yhat1_test)
  out_Rsq1[i] <- summary(fit1)$r.squared
  subset2 <- o2[1:n[i]]
  sub_yhat2_test <- as.matrix(test2[,subset2])%*%as.matrix(coef_2[subset2])
  fit2 <- lm(test$ROI2~sub_yhat2_test)
  out_Rsq2[i] <- summary(fit2)$r.squared
}
```

```
## Graphing out of sample R^2 for different subset sizes based on LASSO Regression
g1<- ggplot()+
  geom_point(aes(x=n,y=out_Rsq1))+
  theme_bw()+
  labs(x="Number of covariates included",y="Out-of-sample R^2",title="ROI1~yhat1 LASSO")+
  ylim(0,1)
g2<- ggplot()+
  geom_point(aes(x=n,y=out_Rsq2))+
  theme_bw()+
  labs(x="Number of covariates included",y="Out-of-sample R^2",title="ROI2~yhat2 LASSO")+
  ylim(0,1)
grid.arrange(g1, g2, nrow = 1)
```


ROI1~yhat1 LASSO
Out-of-sample R^2
1.00
0.75
0.50
0.25
0.00
0
1000
2000
3000
4000
5000
Number of covariates included


ROI2~yhat2 LASSO
Out-of-sample R^2
1.00
0.75
0.50
0.25
0.00
0
1000
2000
3000
4000
5000
Number of covariates included