# Homework 3, STATS 315A

Stanford University, Winter 2019

*Joe Higgins*

## Question 1

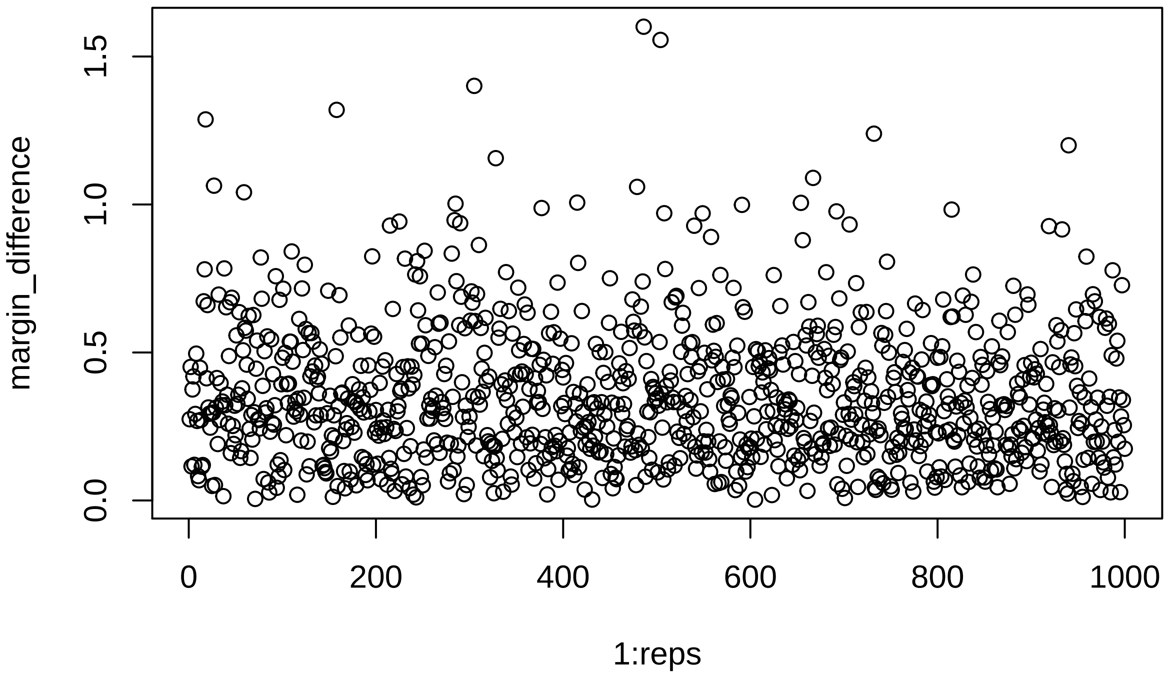
Ex. 18.9 Compare the data piling direction of Exercise 18.8 to the direction of the optimal separating hyperplane (Section 4.5.2) qualitatively. Which makes the widest margin, and why? Use a small simulation to demonstrate the difference.

```r
rm(list = ls())

#Parameters
p <- 25
n <- 20
reps <- 1000

margins <- matrix(,nrow=reps,ncol=2)
for(i in 1:reps){
  #Create covariates
  X <- matrix(mvrnorm(p*n, 0, 1), nrow=n, ncol=p)
  s <- svd(X)
  U <- s$u
  D <- diag(s$d)
  V <- s$v

  #Create response
  y <- matrix(rbinom(n, 1, .5))
  y[y == 0] <- -1

  #Create maximal data piling direction
  B_0 <- V %*% ginv(D) %*% t(U) %*% y

  #Margin from maximal data piling direction
  DP_margin <- 2/norm(B_0,"2")

  #Margin from SVM
  svmfit <- svm(
    y ~ ., data = X,
    kernel = "linear",
    cost=1000, scale = FALSE,
    type='C-classification'
  )

  beta <- drop(t(svmfit$coefs)%*%X[svmfit$index,])
  beta0 <- svmfit$rho
  f_hat <- (X %*% beta - beta0) #same as svmfit$decision.values

  SVM_margin <- 2/norm(beta,"2")

  #Save output
```

```
  margins[i,1] <- DP_margin
  margins[i,2] <- SVM_margin
}

margin_difference <- margins[,2] - margins[,1]
cat("Number of times DP margin larger than SVM margin: ",
    length(margin_difference[margin_difference < 0])
)
```

```
## Number of times DP margin larger than SVM margin:  0
```

```
plot(1:reps, margin_difference)
```



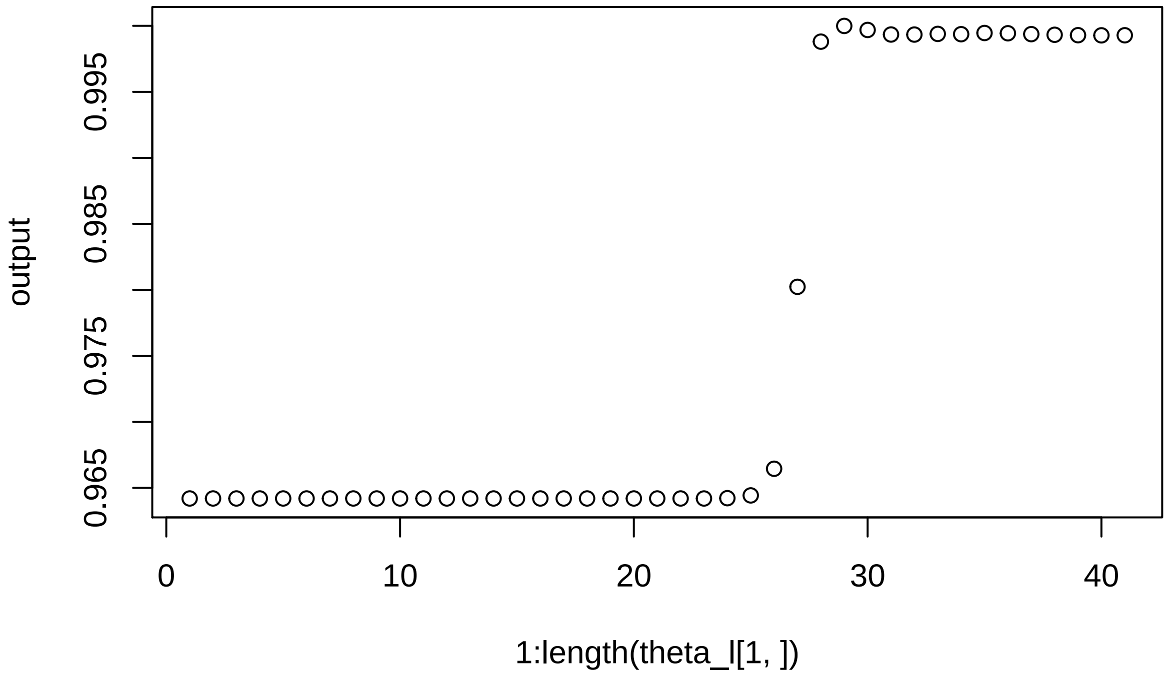
## Question 3

(e) Set up a small simulation with separable data (you don't need to have $p > n$ for this part). Fit the SVM optimal separating hyperplane (for example, use function `svm()` in package `e1071` with a large value for the cost parameter), and extract the unit vector $\beta_{svm}$ normal to the separating hyperplane. Now fit a series of ridged logistic regression models with $\lambda$ decreasing toward 0 (use a fine grid on the log scale). (The package `glmnet` might be useful here, with `alpha=0`, `standardize=FALSE`, and perhaps providing your own sequence of values for lambda). For each of your solutions, compute $\theta_\lambda = \frac{\hat{\beta}_\lambda}{||\hat{\beta}_\lambda||}$ Demonstrate empirically that as $\lambda \downarrow 0$, $|\langle\theta_\lambda, \beta_{svm}\rangle| \to 1$. What is your conclusion?

The conclusion is that as we remove the regularization penalty $\lambda$, the direction of the optimal separating hyperplane and logistic regression align. The logistic regression will approach this direction as $\lambda$ approaches zero, and as the norm of $\beta$ approaches infinity, but won't quite get there.

```r
#make it separable, but make them close! otherwise wont see convergence
rm(list = ls())
n <- 16
p <- 2
d <- 1
data_0 <- cbind(
  rep(0, n/2),
  rep(0, n/2) + runif(n/2, 0, d),
  rep(0, n/2) + runif(n/2, 0, d)
)
data_1 <- cbind(
  rep(1, n/2),
  rep(0, n/2) + runif(n/2, 0, d),
  rep(d, n/2) + runif(n/2, 0, d)
)

data <- rbind(data_0, data_1)
y <- factor(data[,1])
X <- data[,c(2:(p+1))]
X <- matrix(X,nrow=n,ncol=p)

svm_model <- svm(
  y ~ .,
  data = X,
  cost=99999,
  type='C-classification',
  kernel = 'linear',
  scale = FALSE
)

beta_svm <- drop(t(svm_model$coefs)%*%X[svm_model$index,])
beta_svm <- beta_svm/norm(beta_svm,"2")

n_lambdas <- 50
lambdas <- rep(10,n_lambdas) ^ (seq(n_lambdas:1) - rep(n_lambdas/2, n_lambdas))
lr_model <- glmnet(
  X, y, family=c("binomial"),
  alpha = 0, standardize = FALSE, lambda = lambdas
)
```

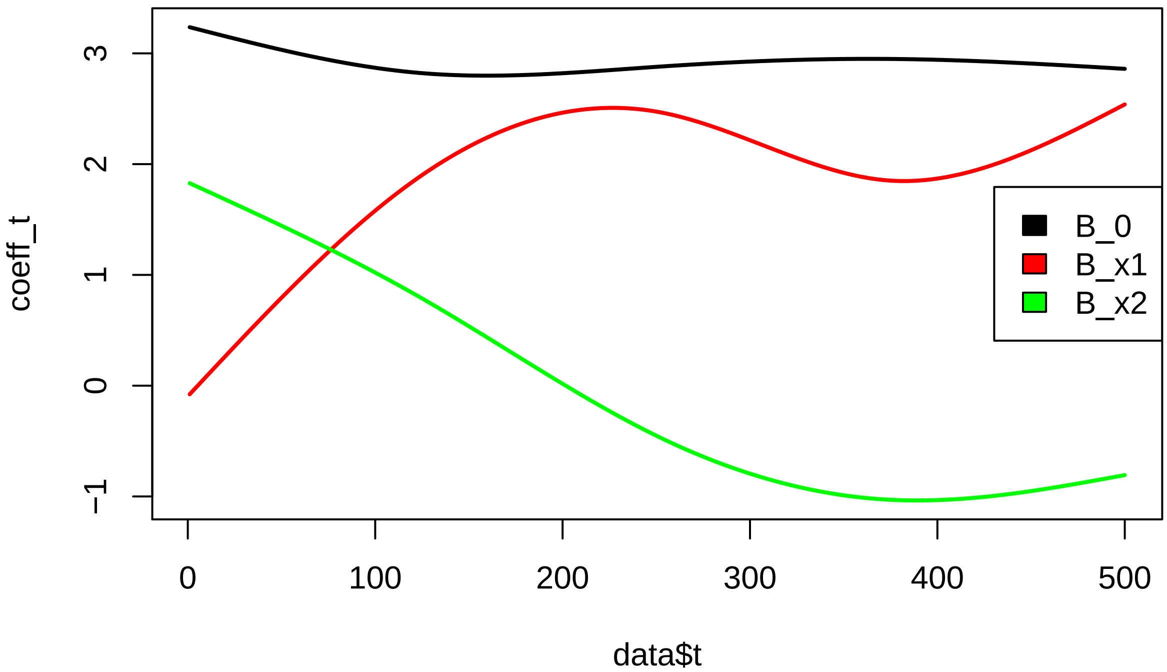
```r
theta_l <- apply(lr_model$beta, 2, function(x) x / norm(x,"2"))
output <- apply(theta_l, 2, function(x) abs(x %*% beta_svm))
plot(1:length(theta_l[1,]), output)
```



## Question 5

(b) Read in the data in `vcdata.csv`. There is a two-column $x$ and a time variable $t$. Fit the varying coefficient model, and plot the three coefficient (functions) versus time (on the same plot using matplot).

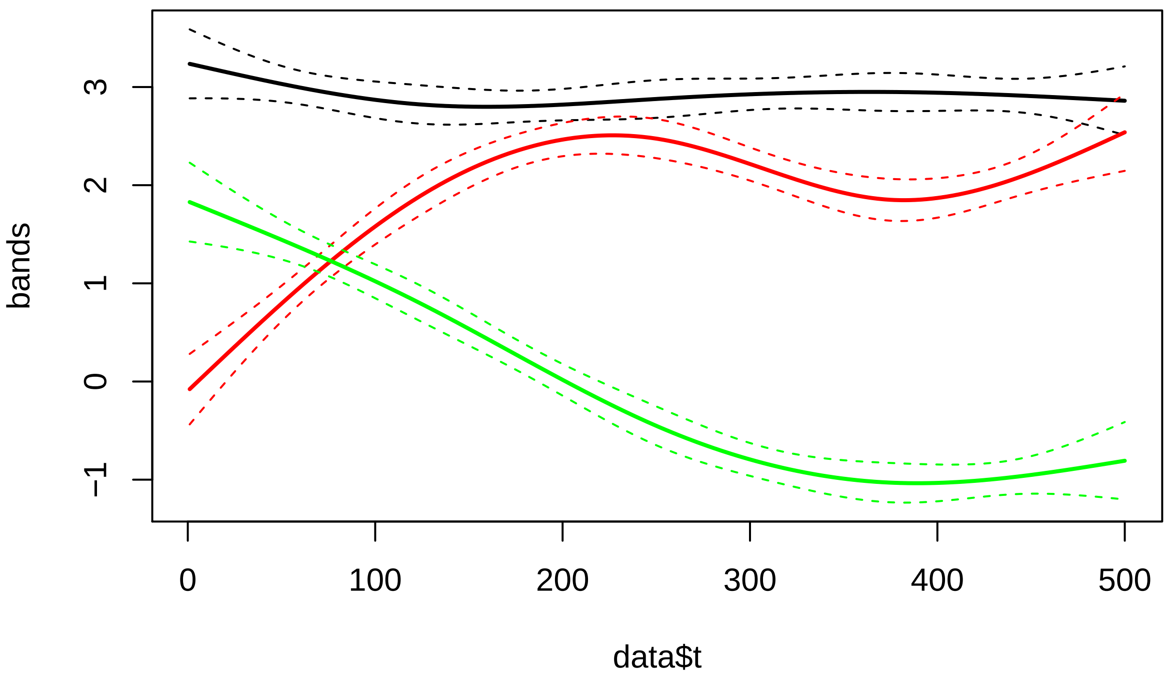
```
#Setup, read data
rm(list = ls())
data <- read.csv("vcdata.csv")

#User parameters
df <- 5

#Design data
column_names <- c('intcp', 'x1', 'x2')
p <- length(column_names)
n <- nrow(data)

t_splines <- ns(data$t, df = df, intercept = TRUE)
intcp <- rep(1, nrow(data))
X <- data.frame(matrix(c(intcp, data$x.1, data$x.2), nrow=n, ncol=p))
colnames(X) <- column_names

#Fit model, aggregate parameters, make predictions
model <- lm(data$y ~ t_splines:(intcp + x1 + x2) - 1, data=X)
bint_t <- t_splines[,] %*% model$coefficients[0*df+1:df]
b1_t <-   t_splines[,] %*% model$coefficients[1*df+1:df]
b2_t <-   t_splines[,] %*% model$coefficients[2*df+1:df]
y_hat <- bint_t + (b1_t * data$x.1) + (b2_t * data$x.2)

#Plot aggregate parameters as function of t
coeff_t <- matrix(c(bint_t, b1_t, b2_t), nrow=nrow(data),ncol=p)
matplot(
  data$t, coeff_t,
  type = "l",
  lty = c('solid','solid','solid'),
  lwd = c(2,2,2),
  col = c('black', 'red', 'green')
)
legend(
  'right',
  legend = c('B_0','B_x1','B_x2'),
  col = c('black', 'red', 'green'),
  fill = c('black', 'red', 'green')
)
```



coeff_t
3
2
1
0
−1
0
100
200
300
400
500
data$t
B_0
B_x1
B_x2

(c) Compute the pointwise standard errors for each of these, and in- clude a standard-error band (upper and lower) for each function.

```
#Create confidence intervals for each
bands <- c()
for(i in 1:ncol(X)){
  #Make synthetic data, all 1s for the coefficient we want to calculate
  synth_X <- matrix(0,nrow=nrow(data),ncol=p)
  synth_X[,i] <- 1
  colnames(synth_X) <- column_names

  #Make "predictions" of B_i, with confidence bands
  bands_i <- predict(model, data.frame(synth_X), interval = 'confidence')
  bands <- cbind(bands,bands_i)
}

#Plot confidence bands of each coefficient
matplot(
  data$t, bands,
  type = "l",
  lty = c('solid','dashed','dashed','solid','dashed','dashed','solid','dashed','dashed'),
  lwd = c(2,1,1,2,1,1,2,1,1),
  col = c('black','black','black','red','red','red','green','green','green')
)
```




(d) Is this an interaction model? If so, what order.

Yes this is an interaction model. It is a second-order interaction model, since the covariates of the model are based on two vectors multiplied element-wise on each other.