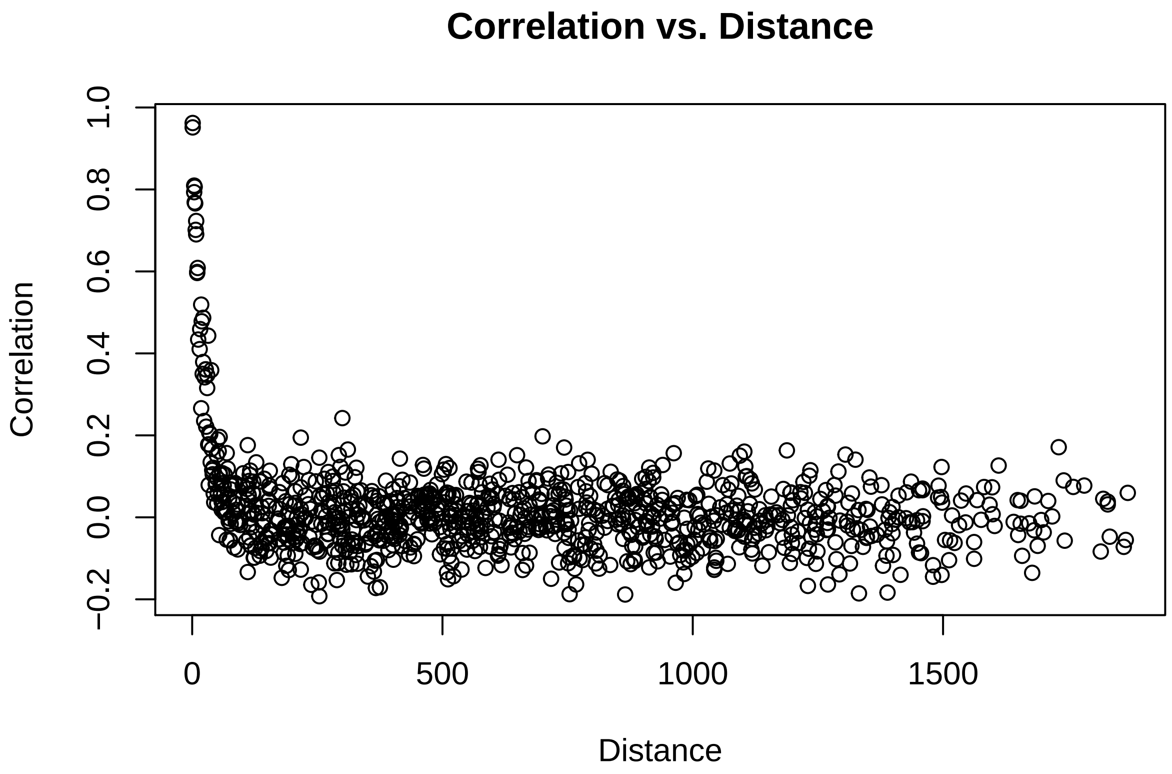
```r
# read training data
con = url("http://www.[redacted].csv")
train = read.csv(con)

# reorganize nicely
p = dim(train)[2]-1
n = dim(train)[1]
X = as.matrix(train[,1:p],nrow=n)
Y = as.numeric(train[,p+1])

# choose 1000 random pairs
indexes = sample(1:p, 2000, TRUE)
M = matrix(nrow = 2, ncol = 1000)
for (i in 1:1000)
{
  M[1,i] = abs(indexes[i]-indexes[i+1000])
  M[2,i] = cor(X[,indexes[i]],X[,indexes[i+1000]])
}
plot(M[1,],M[2,], main = "Correlation vs. Distance", xlab = "Distance", ylab = "Correlation" )
```



Question 1

a

We conclude that the correlation structure is based on distance between the columns. Two adjacent columns are highly correlated and the correlation decreases drastically as they are farther apart.\ This case has a resemblance to the GWAS data, where we assume close (or “in the same neighborhood”) columns are highly correlated and far columns are independent. It seems that this case doesn't comply with compressed sensing assumptions because in the latter case we assume low correlation between all columns.

b

```
library(tictoc)

# GWAS-like marginal regression
tic(msg = "Time to complete GWAS-like marginal regression model")
R2vec = 1:p
for (i in 1:p)
{
  mod1 = lm(Y~X[,i])
  R2vec[i] = summary(mod1)$r.squared
}
best1 = 1:4
orderedVec = order(R2vec, decreasing = TRUE)
for (i in 1:4)
{
  best1[i] = orderedVec[1]
  orderedVec = orderedVec[orderedVec > orderedVec[1]+70 | orderedVec < orderedVec[1]-70]
}
mod1 = lm(Y~X[,best1[1]]+X[,best1[2]]+X[,best1[3]]+X[,best1[4]])
cat("Variables' indexes used: ",best1)
```

```
## Variables' indexes used:  49 350 275 501
```

```
summary(mod1)
```

```
##
## Call:
## lm(formula = Y ~ X[, best1[1]] + X[, best1[2]] + X[, best1[3]] +
##     X[, best1[4]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8399 -0.6805  0.1099  0.7518  2.4860
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.03266    0.07620   0.429    0.669
## X[, best1[1]]  1.01289    0.08136  12.449   <2e-16 ***
## X[, best1[2]]  0.92362    0.07467  12.370   <2e-16 ***
## X[, best1[3]]  1.01323    0.07562  13.398   <2e-16 ***
## X[, best1[4]]  0.85792    0.07490  11.454   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 195 degrees of freedom
## Multiple R-squared:  0.768,	Adjusted R-squared:  0.7632
## F-statistic: 161.3 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
toc()
```

```
## Time to complete GWAS-like marginal regression model: 1.735 sec elapsed
```

```r
# Relaxed Lasso
library(lars)
```

```
## Loaded lars 1.2
```

```r
tic(msg = "Time to complete Relaxed Lasso model")
mod2 = lars(x=X,y=Y,type="lasso",use.Gram=FALSE)
best2 = order(mod2$beta[5,],decreasing = TRUE)[1:4]
mod2 = lm(Y~X[,best2[1]]+X[,best2[2]]+X[,best2[3]]+X[,best2[4]])
cat("Variables' indexes used: ",best2)
```

```
## Variables' indexes used:  49 350 275 501
```

```r
summary(mod2)
```

```
##
## Call:
## lm(formula = Y ~ X[, best2[1]] + X[, best2[2]] + X[, best2[3]] +
##     X[, best2[4]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8399 -0.6805  0.1099  0.7518  2.4860
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.03266    0.07620   0.429    0.669
## X[, best2[1]]  1.01289    0.08136  12.449   <2e-16 ***
## X[, best2[2]]  0.92362    0.07467  12.370   <2e-16 ***
## X[, best2[3]]  1.01323    0.07562  13.398   <2e-16 ***
## X[, best2[4]]  0.85792    0.07490  11.454   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 195 degrees of freedom
## Multiple R-squared:  0.768,  Adjusted R-squared:  0.7632
## F-statistic: 161.3 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
toc()
```

```
## Time to complete Relaxed Lasso model: 4.206 sec elapsed
```

```r
# L0 variable selection
library(leaps)
tic(msg = "Time to complete L0 variable selection model")
best3 = c()
```

```r
for (i in 1:19)
{
  mod3 = regsubsets(x=X[,((i-1)*100+1):((i-1)*100+200)],y=Y,nvmax=4,really.big=T)
  best3 = unique(append(best3,(((i-1)*100):((i-1)*100+200))[summary(mod3)$which[4,]][-1]))
}
mod3 = regsubsets(x=X[,best3],y=Y,nvmax=4,really.big=T)
#tmp = summary(mod3)$which[4,]
best3 = best3[summary(mod3)$which[4,][-1]]
mod3 = lm(Y~X[,best3[1]]+X[,best3[2]]+X[,best3[3]]+X[,best3[4]])
cat("Variables' indexes used: ",best3)
```

```
## Variables' indexes used:  49 275 350 500
```

```r
summary(mod3)
```

```
##
## Call:
## lm(formula = Y ~ X[, best3[1]] + X[, best3[2]] + X[, best3[3]] +
##     X[, best3[4]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.82733 -0.68991  0.05756  0.66698  2.32166
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.03828    0.07456   0.513    0.608
## X[, best3[1]] 1.01796    0.07962  12.785   <2e-16 ***
## X[, best3[2]] 1.03561    0.07397  14.001   <2e-16 ***
## X[, best3[3]] 0.93558    0.07307  12.803   <2e-16 ***
## X[, best3[4]] 0.87265    0.07231  12.069   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.033 on 195 degrees of freedom
## Multiple R-squared:  0.7778, Adjusted R-squared:  0.7733
## F-statistic: 170.7 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
toc()
```

```
## Time to complete L0 variable selection model: 289.758 sec elapsed
```

c

```r
# read test data
con = url("http://www.[redacted].csv")
test = read.csv(con)

# reorganize nicely
p = dim(test)[2]-1
```

```
n = dim(test)[1]
X = as.matrix(test[,1:p],nrow=n)
Y = as.numeric(test[,p+1])

cat("The MSE of GWAS-like marginal regression:",mean((Y-predict(mod1,data.frame(X)))^2))
```

```
## The MSE of GWAS-like marginal regression: 1.147111
```

```
cat("The MSE of Relaxed Lasso:",mean((Y-predict(mod2,data.frame(X)))^2))
```

```
## The MSE of Relaxed Lasso: 1.147111
```

```
cat("The MSE of L0 variable selection:",mean((Y-predict(mod3,data.frame(X)))^2))
```

```
## The MSE of L0 variable selection: 1.040422
```

We can see that the performances of the models are very similar, which is not surprising because the first 2 models has chosen the same features and the third chose only one feature that is different but is adjacent so highly correlated. The main difference between the models is the running time that is much longer in the third model, L0 variable selection (taking few minutes instead of few seconds).

d

We have noticed that all the approaches had chosen the same, or almost the same, variables, but our confident in those specific features is not high. Because of the high correlation between adjacent columns, small changes in the Y (due to noise) can cause us to confuse between near columns. We have seen that the third model for example did chose one different variable and indeed it was an adjacent column. This is exactly why the compressed sensing assumptions, which are not valid here, are important.