1.

$L(x) = \Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2 = \Sigma_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$
$\frac{\partial L}{\partial \beta_0} = 0$
$\Rightarrow -2\Sigma_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0 \Rightarrow \Sigma_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0$
$\Rightarrow n\bar{y} - n\beta_0 - n\beta_1\bar{x} = 0 \Rightarrow \bar{y} - \beta_0 - \beta_1\bar{x} = 0$
$\Rightarrow \beta_0 = \bar{y} - \beta_1\bar{x}$

$\frac{\partial L}{\partial \beta_1} = 0$
$\Rightarrow -2\Sigma_{i=1}^{n}x_i(y_i - \beta_0 - \beta_1 x_i) = 0 \Rightarrow \Sigma_{i=1}^{n}(x_i y_i - \beta_0 x_i - \beta_1 x_1^2) = 0$
$\Rightarrow n\overline{xy} - n\beta_0\bar{x} - n\beta_1\overline{x^2} = 0 \Rightarrow \overline{xy} - \beta_0\bar{x} - \beta_1\overline{x^2} = 0$
$\Rightarrow \overline{xy} - (\bar{y} - \beta_1\bar{x})\bar{x} - \beta_1\overline{x^2} = 0 \Rightarrow \overline{xy} - \bar{x}\bar{y} = \beta_1\overline{x^2} - \beta_1\bar{x}^2 \Rightarrow \overline{xy} - \bar{x}\bar{y} = \beta_1\left(\overline{x^2} - \bar{x}^2\right)$
$\Rightarrow \beta_1 = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} = \frac{\Sigma_{i=1}^{n}x_i(y_i - \bar{y})}{\Sigma_{i=1}^{n}x_i(x_i - \bar{x})} = \frac{\Sigma_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}$

Sorry, forgot to add hat symbol to each parameter.

2.

a.

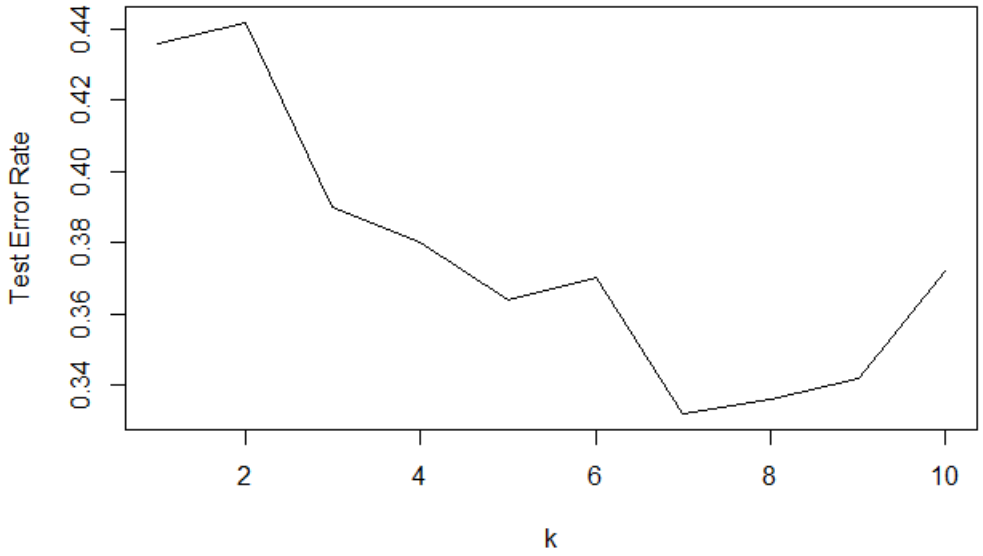
i. We expect the overall Bayes error rate to be the same for the train and test sets, knowing the true conditional distribution of $Y|X$, expressed as $P(Y = 1|X_1, X_2) = \Phi(0.5X_1 - 0.4X_2)$. The Bayes error rate (BER) for a specific observation (x_1, x_2) is computed by $1 - \max_j P(Y = j|X_1 = x_1, X_2 = x_2)$. This means that computing the BER is deterministic and produces the same constant regardless of set of observations (x_1, x_2) becomes shuffled into from train-test-split of the original data. An arbitrary observation also possesses a 50%-50% probability of being in the train or test set, as both sets have equal size. When computing the overall BER with $1 - E\left[\max_j P(Y = j|X_1, X_2)\right]$, the individual contribution of an arbitrary observation to 2nd term (expected value) for either the train or test set is therefore 50%-50%. Therefore, in theory we expect both sets to be equal in overall BER. Running this question's code cell produces overall BERs of 0.3207711 and 0.318458 for the train and test sets, respectively.

ii. Assuming we do not know the true posterior conditional probability of $Y \mid X$ mentioned in i., we can employ a naïve bayes classifier. From executing the corresponding cell, the overall BERs were 0.338 and 0.314 for the train and test sets, respectively.

iii. This time, a KNN classifier is employed with $k = 3$. Note that all KNN classifiers in the code center and scale each predictor column before classification. The test error rate is 0.39.

iv. Since the test error rate for the KNN classifier in iii. (0.39) is noticeably higher than the test error rate for naïve bayes classifier (0.318458), the value $k = 3$ is not the best parameter choice.

v. Several contiguous candidates in the vicinity of the current value for k have been plotted, with y-axis being the test error rate and x-axis being k, the number of neighbors.



The optimal value in the domain of the plot seems to be $k = 7$ with a test error rate of approximately 0.33. However, a gridsearch hyperparameter tuning scheme was done with a domain of $k \in \{1,2, \dots, 100\}$. The optimal value now is $k = 69$ with a test error rate of 0.328. For practicality, setting k to either value or any other value within a miniscule gap for the test error rate would not be too different.

vi. KNN does an excellent role of approximating naïve bayes for classification. From the theoretical side, the posterior conditional probability of $P(Y = j|X_1 = x_1, X_2 = x_2)$ can be effectively estimated with $P(Y = j|\widehat{X_1 = x_1}, X_2 = x_2) = \frac{1}{k}\Sigma_{i \in N_0} I(y_i = j)$, where N_0 denotes the set of k nearest neighbors to (x_1, x_2), and I is an indicator variable that is 1 if class j was predicted correctly and 0 otherwise. The RHS of the equation is essentially the KNN algorithm expressed formulaically. Furthermore, KNN has a tunable parameter while naïve bayes lacks any when using the most fundamental versions of each classifier. This is a key advantage for KNN.

vii. Using X_1, X_2, and 20 predictors from xrandom together under a KNN classifier with $k = 40$ yielded a test error rate of 0.408.

viii. Appending more predictors does not necessarily increase the accuracy (or decrease the error rate). In this case, this is especially true by appending predictors that hold random values from a distributions, which give no extra information that intuitively helps with the classification.

3.

a.

i. Although numerical in name, Year is better used as a categorical predictor (factor in R) for several reasons. First, specific integer values for Year occur repeatedly across tuples of different

values for Lag1 – Lag5 and Volume. Therefore, these tuples can be "binned" into specific years. Second, the range and variance of the outcome, Today, varies drastically from year to year when plotted by Year. It makes more sense to predict the value of Today within the context of each year separately than together. In the full model for multiple regression, the estimated coefficient for Year would point out the average difference in outcome (Today) from a specific year to the base (2001).

ii. The hypothesis test uses the null hypothesis $H_0: \beta_0 = \beta_1 = \cdots = \beta_7$. Using the *summary* function in R, an F-statistic of 0.99378 with a p-value of 0.9122 are yielded. The p-value is no where near the commonly used thresholds of 0.10, 0.05, or smaller to confidently reject H_0. Furthermore, the F-statistic being very close to 1 suggests little difference in sum of squared errors between the full model and reduced model (see code for predictors in each model). Therefore, we cannot reject the null hypothesis and are inconclusive as to whether any covariate is predictive of the outcome.

iii. Using polynomial regression, even while omitting Lag2-4, performed better than the linear regression in ii. This time, an F-statistic of 0.9234 and p-value of 0.477 was produced, both smaller than their linear counterparts.

b.

i. The best performing KNN classifier had $k = 70$ with a test error rate of 0.1426508.

ii. Covariates are not very predicative when comparing the linear regression and KNN classifier. Simply tuning KNN led to dramatic decrease in test error rate unseen in part a.

4.

i. Yes. The expected value of an arbitrary observation is the mean, which can be estimated by the sample mean.

ii. As the sample size *n* increases, there are more and more evidence to back up a confidence interval. Therefore, the CI in part I goes to zero.