Question 1:

a) The "test error criterion" aims to adjust the bias and variance due to overfitting and generate the testing error indirectly by deal with the fact of underestimation, when the model size is known.

- works well for simple models such as linear-regression.
- more interpretability with the adjustments on training error.
- closed-form criterion: can be computed efficiently and easy to apply
- sometimes might perform poorly if the adjustment made for training error is far away from the true data set or true data distribution

The "cross-validation" aims to directly derive the test error by randomly dividing a data-set into $k = n$ parts, with less assumptions needed for the true model, and it is applicable when p (the number of predictors) and $\hat{\sigma}^2$ is unknown.

- apply for a wider range of models such as complicated non-linear models.
- provide a direct estimate of test error instead of approximation.
- no need for known value of $\hat{\sigma}^2$ or error variance or degree of freedom.
- generalize easily to bigger and complex models.
- results might be highly-variable because it depends on which part of the training data has been included and which part has not been included, and it also produce overestimation because we are using a small part of data to estimate the whole set of data.
- it can be expensive or time-consuming with a large n and large k-part.

Thus, when we have a relatively-simple linear model (with less predictors and a known model size $n = xx$), with known or calculatable $\hat{\sigma}^2$, and we want to interpret it clearly, we may choose (i) test-error criterion, vice-versa.
Cross-validation might not work well for a large dataset, so we can choose AIC/BIC for a very big n.

b) $AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$, where n = number of observations

$BIC = \frac{1}{n\hat{\sigma}^2}(RSS + \log(n)\, d\hat{\sigma}^2)$

d = number of predictors

$\hat{\sigma}^2$ = the variance of the ε associated with each response

The BIC replace 2 with $\log(n)$,
thus $\log(n) > 2$ when $n > 7$, therefore BIC places a heavier penalty on models with a bigger n, resulting in a model with less predictors.

↓ Therefore, we choose AIC if we would like to produce a more accurately designated model with adequate predictors, and we may choose BIC if we want to find a more simpler and more-interpretable model, which might result in a less accurate (more bias) model selection with smaller variance.

c) $C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$, the smaller the value of C_p, the smaller the value of test error.

It penalize heavier with a bigger d or a bigger $\hat{\sigma}^2$ is because the model becomes more complex if we add more predictors in and so the training error will decrease which might underestimate a the true testing error and the model might involve in the problem of overfitting if the irreducible error $\hat{\sigma}^2$ is big, which also requires a greater variance adjustment in the bias-variance trade-off.

d) From part (b) we know that BIC will results in a model with less predictors, thus, for BIC, it may has a smaller value of $\text{Var}\hat{f}(x)$ but a bigger value of $\text{Bias}\hat{f}(x)$ because it is simpler with less predictors, avoiding the problem of overfitting and may also apply to other new data sets while can generate bigger bias due to lack of precision, vice-versa.
Thus $\hat{f}_1(x) \rightarrow$ larger $\text{Var}\hat{f}_1(x)$, smaller $\text{Bias}\hat{f}_1(x)$
$\hat{f}_2(x) \rightarrow$ large $\text{Bias}\hat{f}_2(x)$, bigger $\text{Var}\hat{f}_1(x)$.

Question 2:

a) For model 1, we need 3 parameters,
model 2, we need 5 parameters
model 3, ... 7 parameters
model 4, ... 9 parameters
} For a polynomial model of degree M, we might need $2M+1$ parameters.

b). The bias is biggest with model of degree 1, and substancially decreases with model of degree 2 and degree 3, because black lines (fitted mode) move closer and shaped similar toward the purple line (BIC model).

The bias is almost the same with models of degree 3 and degree 4.

The variance is smallest with linear model of degree 1, and increases with more parameters involved in and with the increase of complexity of the models, the model with higher variance (degree 3 or 4) may not easily to a new data set.

c) The order of the true optimal model = 4,
the order of the estimated optimal model = 3.

The black line is lower than the orange curve means that we overestimate the test-error, the reason is 1) randomly split the data into K=10 folds introducing bias for test error estimation 2) the training set we use for estimation is smaller than the full data set (because we split the full data into several parts of training data)

d) i) The number of parameters in this model.
If we increase the number of predictors, the model's complexity will increase and become less interpretable.

ii) The terms of interactions between two variables that have multicollinearity, we add an interaction to capture the correlation between predictors, and the model complexity will also increase if there are more interaction terms.
The variance for both model will increase while the bais reduces.

##Question 3:

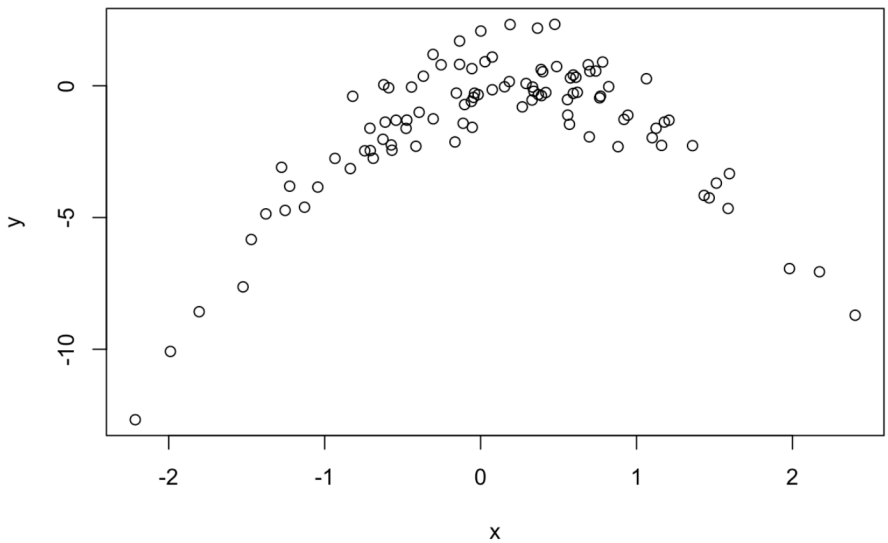
a. Generate a simulated data set as follows: In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

```
set.seed(1)
x=rnorm(100)
y=x-2*x^2+rnorm(100)
```

The n(the number of observations) is 100, and the p(the number of predictors) is 2, and the model can be written as $Y = X - 2X^2 + \epsilon$, where ϵ for N(0, 1)

b. Create a scatterplot of X against Y . Comment on what you find.

```
plot(x, y)
```



The relationship between X and Y apparently is not linear.

c. Set a random seed, and then compute the LOOCV errors that result from fitting the following four models using least squares:

i. Y = β0 + β1X + ε

```
library(boot)
set.seed(1)
Data <- data.frame(x, y)
fitted_glm_1 <- glm(y ~ x)
cv.glm(Data, fitted_glm_1)$delta[1]
```

```
## [1] 7.288162
```

ii. Y = β0 + β1X + β2X2 + ε

```
fitted_glm_2 <- glm(y ~ poly(x, 2))
cv.glm(Data, fitted_glm_2)$delta[1]
```

```
## [1] 0.9374236
```

iii. Y = β0 +β1X +β2X2 +β3X3 +ε

```
fitted_glm_3 <- glm(y ~ poly(x, 3))
cv.glm(Data, fitted_glm_3)$delta[1]
```

```
## [1] 0.9566218
```

iv. $Y = \beta0 + \beta1X + \beta2X2 + \beta3X3 + \beta4X4 + \varepsilon$.

```
fitted_glm_4 <- glm(y ~ poly(x, 4))
cv.glm(Data, fitted_glm_4)$delta[1]
```

```
## [1] 0.9539049
```

d. Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why?

```
set.seed(10)
new_fitted_glm_1 <- glm(y ~ x)
cv.glm(Data, new_fitted_glm_1)$delta[1]
```

```
## [1] 7.288162
```

```
new_fitted_glm_2 <- glm(y ~ poly(x, 2))
cv.glm(Data, new_fitted_glm_2)$delta[1]
```

```
## [1] 0.9374236
```

```
new_fitted_glm_3 <- glm(y ~ poly(x, 3))
cv.glm(Data, new_fitted_glm_3)$delta[1]
```

```
## [1] 0.9566218
```

```
new_fitted_glm_4 <- glm(y ~ poly(x, 4))
cv.glm(Data, new_fitted_glm_4)$delta[1]
```

```
## [1] 0.9539049
```

The results that I have is identical with what I got in part c using another random seed, and the reason is that LOOCV evaluates n folds of a single observation.

e. Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer. The second model in part(c) had the smallest LOOCV error, which correspond with my expectation because as we seen in part(b) which is clear that the relationship between X and Y is quadratic so the estimation is appropriate.

Question 4 :

a) The model with best subset selection will generate the smallest training RSS because it take into accounts of every possible combination of predictors, and it go through 2^p (where p is the number of variables considered in the data set) combinations of predictor, then it gives out the model that contain the highest adjusted R-squared. (with smallest RSS)

b) When n (the number of observations) $> p$ (the number of predictors), the backward stepwise selection will generate a smaller testing RSS because the best subset selection will overfit the model with higher variance because it might include many predictors to guarantee the highest adjusted R-squared.

When $n < p$, the forward stepwise selection will generate a smaller testing RSS because it is the only viable method when there are lots of parameters should be considered in the model, and it is time-saving and an easier alternative of best subset selection.

Overall, it is hard to determine because sometimes when there is no such big differences between n and p, the backward and forward models might perform the same because they are all trained on the training data.

c) False or True

i) T

ii) T

iii) F

iv) F

v) F