Chapter 3 Problem 13

In this exercise, you will create some simulated data and will fit simple linear regression models to it.

*Note – When working on problems that uses randomly-generated data sets, set.seed() reproduces the exact same set of random numbers. This ensures consistent results.

```
> set.seed(1)
```

a. Using the rnorm() function, create a vector x containing 100 observations drawn from a N(0,1) distribution. This represents a feature, X.

```
> x<-rnorm(100,0,1)
```

b. Using the rnorm() function, create a vector eps containing 100 observations drawn from a N(0,0.25) distribution.

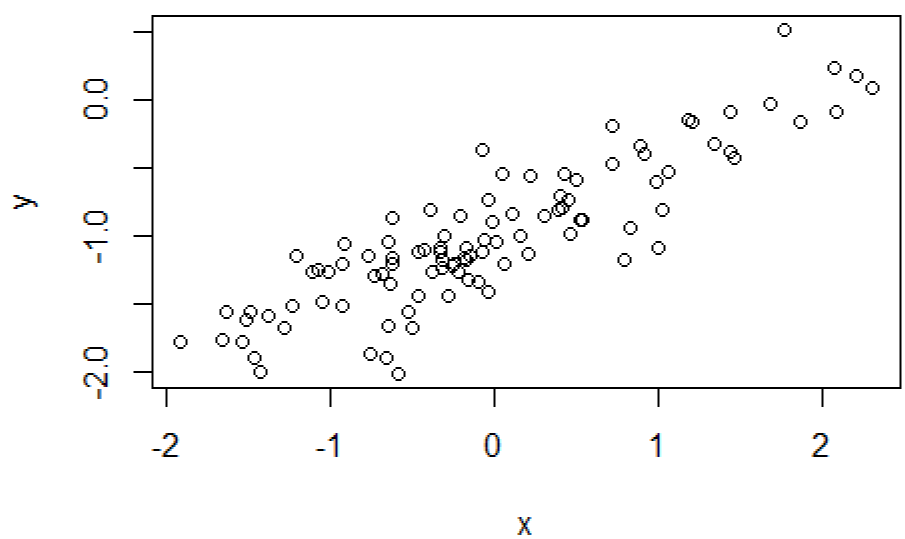
```
> eps<-rnorm(100,0,0.25)
```

c. Using x and eps, generate a vector y according to the model Y= -1 + 0.5X + E. What is the length of vector y? What are the values of β_1 and β_0 in this linear model?

```
> y<-0.5*x-1+eps
> length(y)
[1] 100
```

$\beta_1 = 0.5$ and $\beta_0 = -1$

d. Create a scatterplot displaying the relationship between x and y.



There is a positive relationship between x and y.

e. Fit a least squares model to predict y using x. Comment on the model obtained. How do the estimated coefficients compare to those in part c?

```
> lm.fit=lm(y~x)
> summary(lm.fit)
```

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.73702 -0.11537  0.00323  0.16255  0.65435 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.99309    0.02598  -38.23   <2e-16 ***
x            0.48663    0.02723   17.87   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2596 on 98 degrees of freedom
Multiple R-squared:  0.7651,	Adjusted R-squared:  0.7627 
F-statistic: 319.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

The model suggests a strong relationship between x and y. The estimated coefficients are quite close to the actual β_1 and β_0 values in part c.

f. Display the least squares and population regression line on the scatterplot in part d. (something wrong with legend function)

```
> plot(x,y)
> abline(-1,0.5,col="green",lwd=1)
> abline(lm.fit,col="red",lwd=3)
```

g. Now fit a polynomial regression model that predicts y using x and x^2. Is there evidence that the quadratic term improves the model fit?

```
> lm.fitpoly=lm(y~x+I(x^2))
> summary(lm.fitpoly)

Call:
lm(formula = y ~ x + I(x^2))

Residuals:
     Min       1Q   Median       3Q      Max 
-0.72471 -0.13441  0.01034  0.15372  0.68402 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.02386    0.03336 -30.689   <2e-16 ***
x            0.47490    0.02825  16.811   <2e-16 ***
I(x^2)       0.03334    0.02288   1.457    0.148    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2581 on 97 degrees of freedom
Multiple R-squared:  0.7702,	Adjusted R-squared:  0.7654 
F-statistic: 162.5 on 2 and 97 DF,  p-value: < 2.2e-16
```

There is no evidence suggesting that the quadratic term improves the model fit, given its high p-value.