HW1

Xinwei Zhang xz2663

2/4/2018

P3

(a)

```
set.seed(1)
x = rnorm(100, mean=0, sd=1)
```

(b)

```
eps = rnorm(100, mean=0, sd=sqrt(0.25))
```

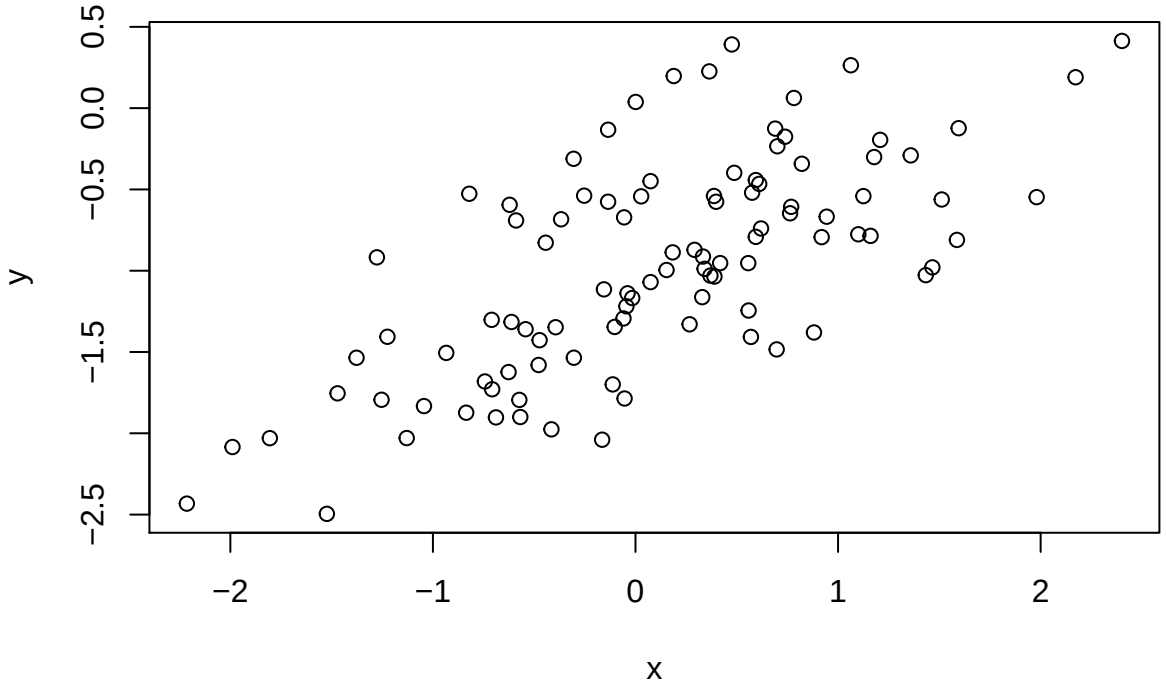
(c)

```
y = -1 + 0.5*x + eps
```

Length of y is 100. β_0 is -1, β_1 is 0.5.

(d)

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



There is a linear relationship between x and y with a slope of positive value and variance.

(e)

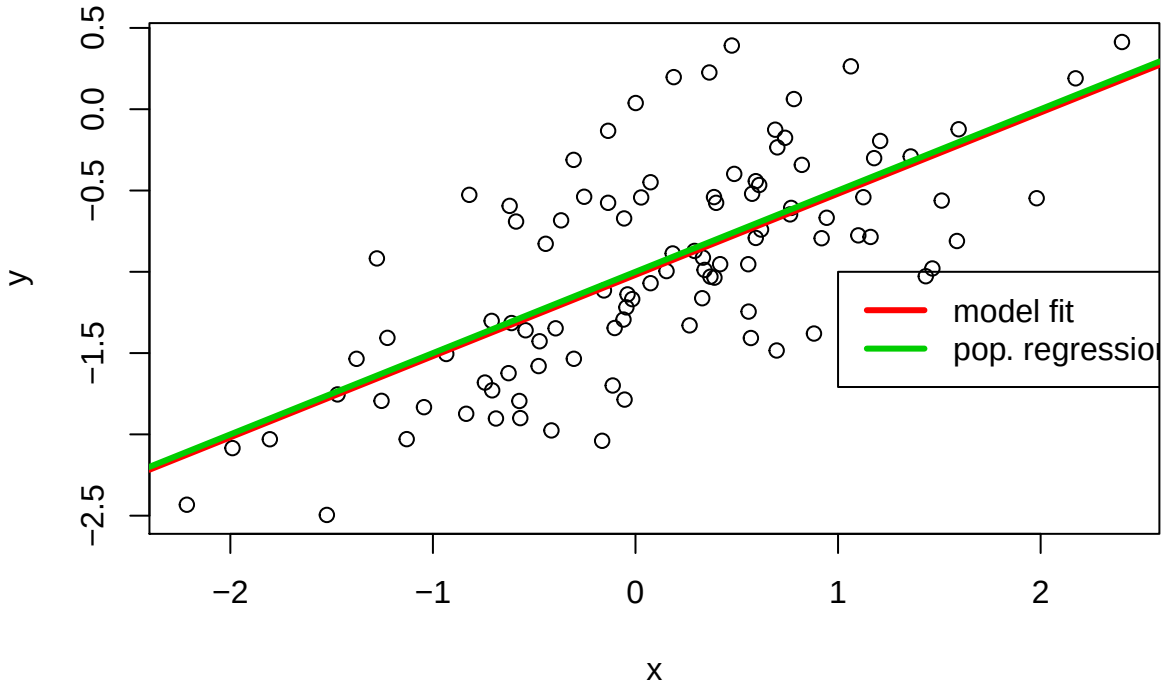
```
lm.fit = lm(y~x)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x            0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

The linear regression fits a model close to the true value of the coefficients as was constructed. The model has a large F-statistic and a near-zero p-value so the null hypothesis can be rejected.

(f)

```
plot(x,y)
abline(lm.fit, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend(-1, legend = c("model fit", "pop. regression"), col=2:3, lwd=3)
```



(g)

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

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x            0.50858    0.05399   9.420  2.4e-15 ***
## I(x^2)      -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

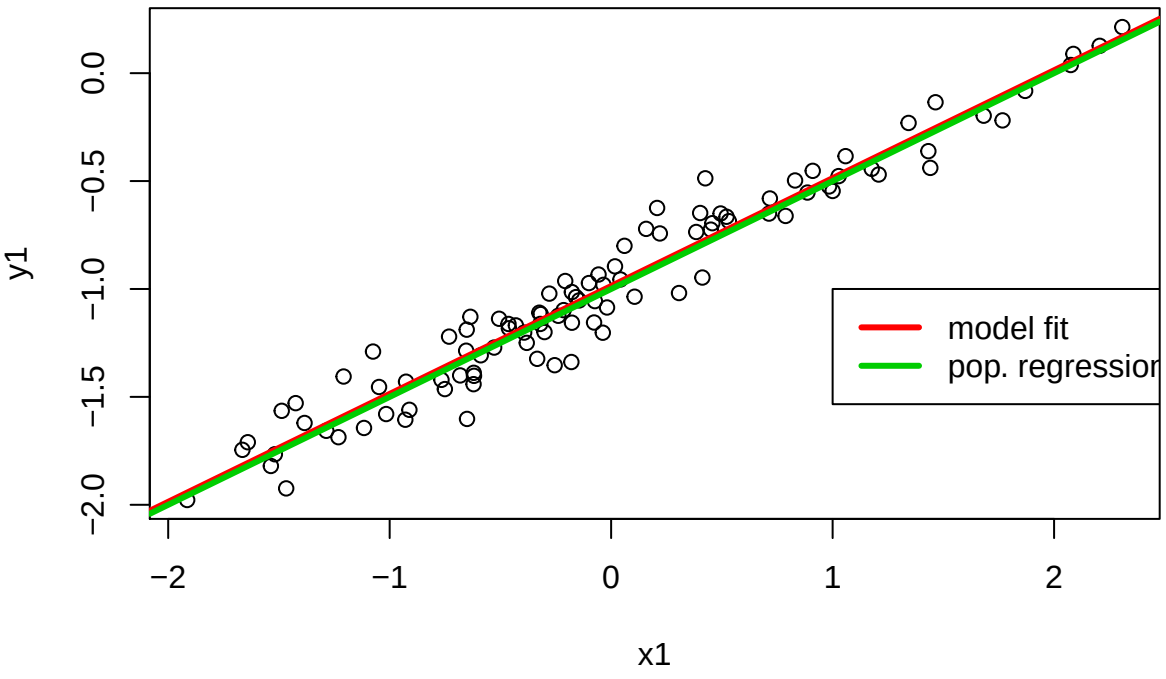
There is evidence that this model fit increased given the slight increase in R^2 and decrease in RSE. Although, the p-value suggests that there isn't a relationship between y and x^2.

(h)

```
set.seed(1)
eps1 = rnorm(100, 0, 0.125)
x1 = rnorm(100)
y1 = -1 + 0.5*x1 + eps1
plot(x1, y1)
lm.fit1 = lm(y1~x1)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29052 -0.07545  0.00067  0.07288  0.28664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98639    0.01129  -87.34   <2e-16 ***
## x1           0.49988    0.01184   42.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1128 on 98 degrees of freedom
## Multiple R-squared:  0.9479, Adjusted R-squared:  0.9474
## F-statistic:  1782 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
abline(lm.fit1, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend(-1, legend = c("model fit", "pop. regression"), col=2:3, lwd=3)
```



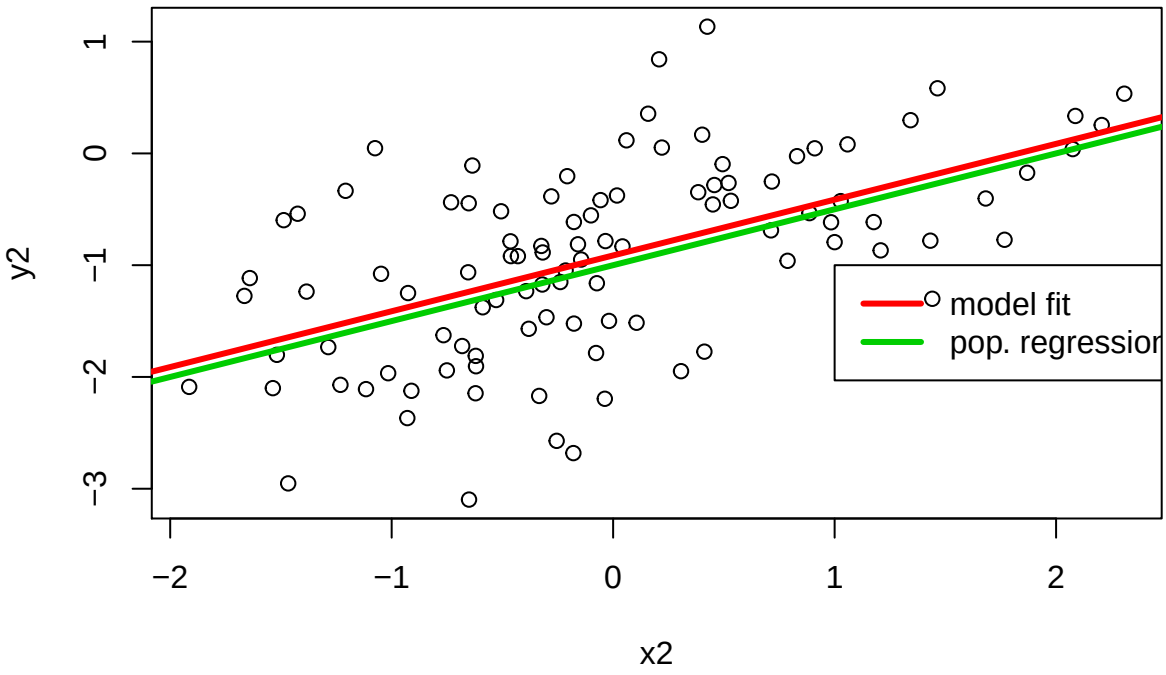
Observed RSE decreased and R^2 increased.

(i)

```
set.seed(1)
eps2 = rnorm(100, 0, 0.8)
x2 = rnorm(100)
y2 = -1 + 0.5*x2 + eps2
plot(x2,y2)
lm.fit2 = lm(y2~x2)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = y2 ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.85933 -0.48289  0.00429  0.46644  1.83453
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.91292    0.07228 -12.631  < 2e-16 ***
## x2           0.49925    0.07578   6.588 2.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7222 on 98 degrees of freedom
## Multiple R-squared:  0.307,  Adjusted R-squared:  0.2999
## F-statistic: 43.41 on 1 and 98 DF,  p-value: 2.235e-09
```

```
abline(lm.fit2, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend(-1, legend = c("model fit", "pop. regression"), col=2:3, lwd=3)
```



Observed RSE increased and R^2 decreased.

(j)

```
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.1150804 -0.9226122
## x            0.3925794  0.6063602
```

```
confint(lm.fit1)
```

```
##                2.5 %     97.5 %
## (Intercept) -1.008805 -0.9639819
## x1           0.476387  0.5233799
```

```
confint(lm.fit2)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.0563524 -0.7694842
## x2           0.3488766  0.6496316
```

All intervals centered around 0.5. The interval of fit1 (less noisy) is narrower than that of fit (original). The interval of fit2 (noisier) is wider than that of fit (original).

P4

```
library(ISLR)
Advertising <- read.csv("http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv",header=T,na.string=',')
#attach(Advertising)
model_newspaper = lm(sales~newspaper, data=Advertising)
model_TV = lm(sales~TV, data=Advertising)
```

```
model_radio = lm(sales~radio, data=Advertising)
confint(model_newspaper, , 0.92)
```

```
##                      4 %        96 %
## (Intercept) 11.25788302 13.44493112
## newspaper    0.02552451  0.08386169
```

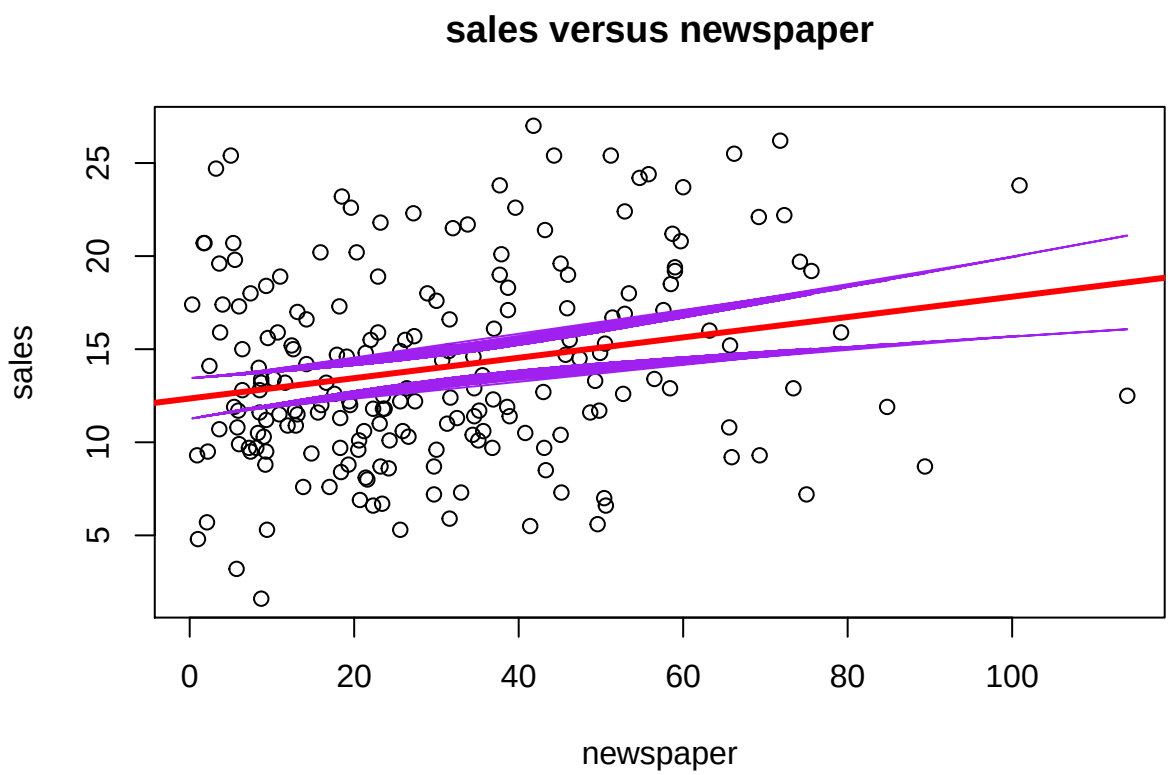
```
confint(model_TV, , 0.92)
```

```
##                    4 %       96 %
## (Intercept) 6.22691926 7.83826784
## TV          0.04280193 0.05227135
```

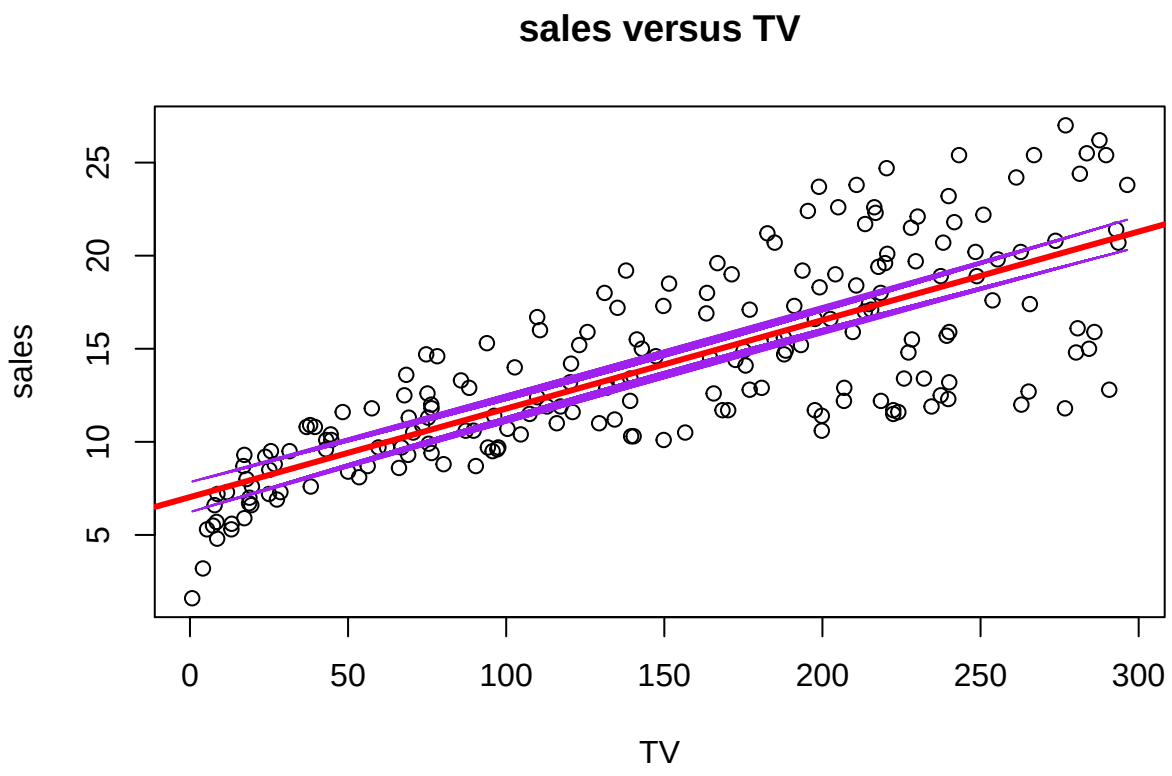
```
confint(model_radio, , 0.92)
```

```
##                   4 %       96 %
## (Intercept) 8.3210922 10.3021840
## radio       0.1665776  0.2384139
```

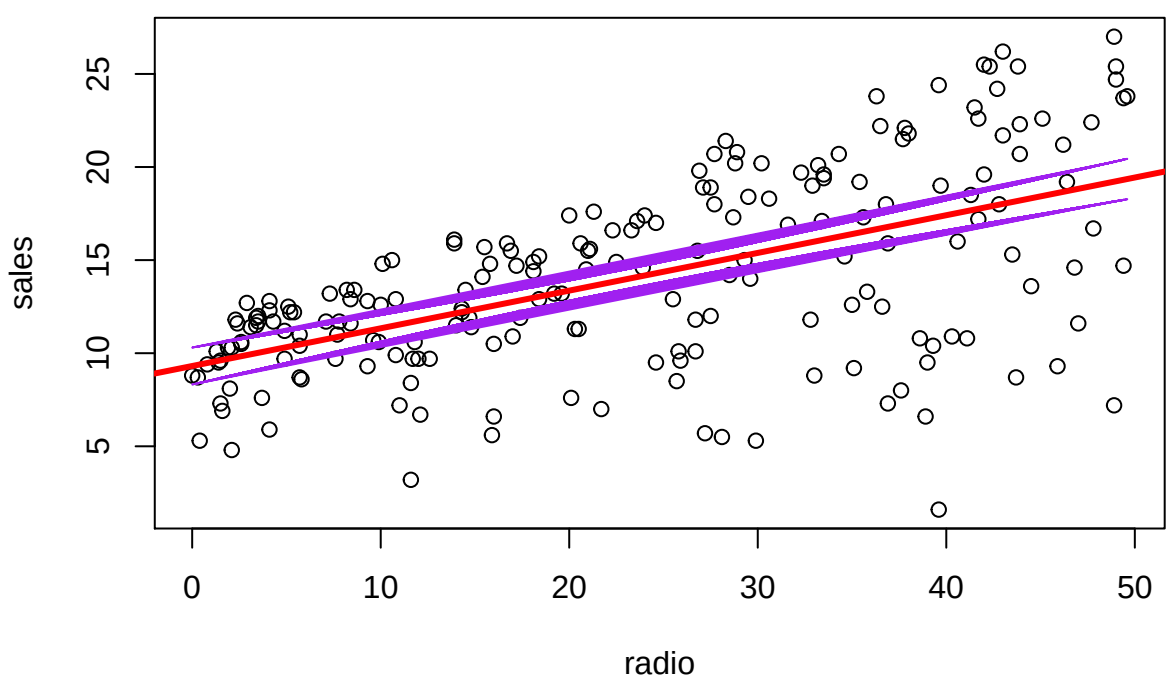
```
plot(Advertising$newspaper,Advertising$sales, xlab = "newspaper", ylab = "sales", main = "sales versus n
abline(model_newspaper, lwd=3, col=2)
confint_newspaper = predict(model_newspaper, newspaper = data.frame(Advertising$newspaper), interval =
lines(Advertising$newspaper, confint_newspaper[,2], col="purple")
lines(Advertising$newspaper, confint_newspaper[,3], col="purple")
```




```
plot(Advertising$TV,Advertising$sales, xlab = "TV", ylab = "sales", main = "sales versus TV")
abline(model_TV, lwd=3, col=2)
confint_TV = predict(model_TV, TV = data.frame(Advertising$TV), interval = "confidence", level=0.92)
lines(Advertising$TV, confint_TV[,2], col="purple")
lines(Advertising$TV, confint_TV[,3], col="purple")
```



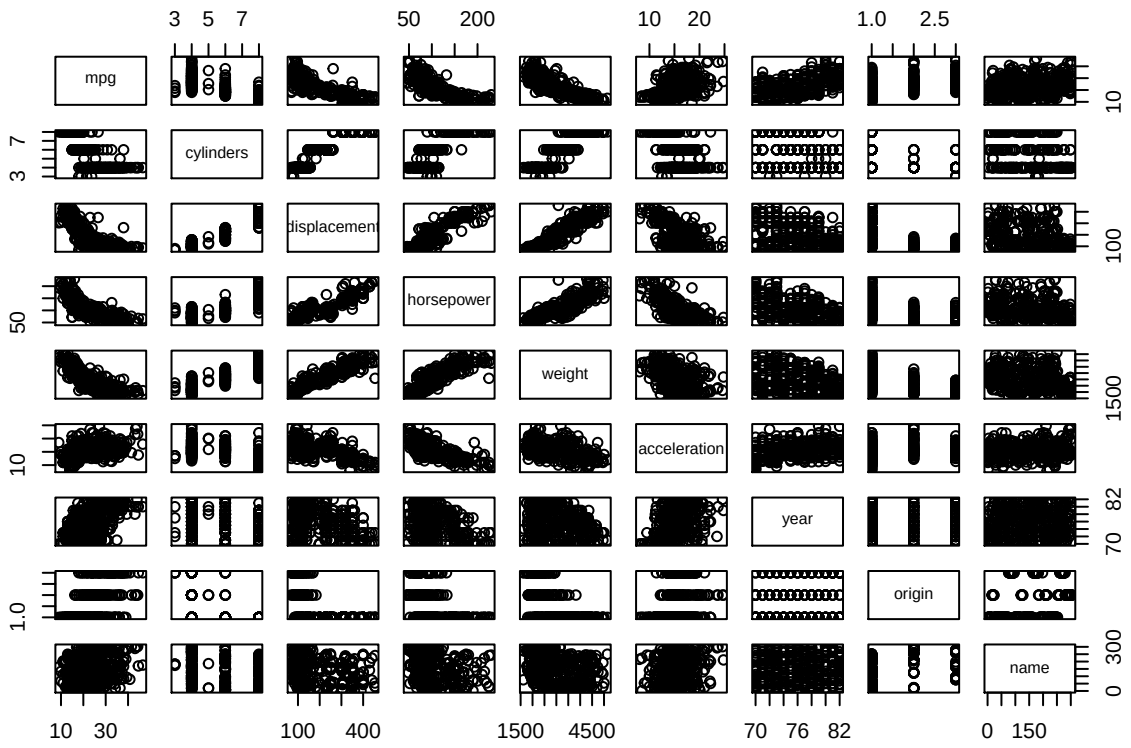

```
plot(Advertising$radio,Advertising$sales, xlab = "radio", ylab = "sales", main = "sales versus radio")
abline(model_radio, lwd=3, col=2)
confint_radio = predict(model_radio, radio = data.frame(Advertising$radio), interval = "confidence", lev
lines(Advertising$radio, confint_radio[,2], col="purple")
lines(Advertising$radio, confint_radio[,3], col="purple")
```



P5

(a)

```
pairs(Auto)
```



(b)

```
cor(subset(Auto, select=-name))
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

(c)

```
lm.fit1 = lm(mpg~.-name, data=Auto)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

(i)

There is a relationship between the predictors and the response by testing the null hypothesis of whether all the regression coeffcients are zero. F statistic is large with small p value, which is againest the null hypothesis.

(ii)

Displacement, weight, year and origin have significant relationship to the response.

(iii)

The coefficient for year, 0.750773, suggests that for every year, mpg increases by 0.750773. Cars become more fuel efficient every year.

(d)

```
lm.fit2 = lm(mpg~log(weight)+sqrt(horsepower)+I(acceleration^2), data=Auto)
summary(lm.fit2)
```

```
##
## Call:
```

```
## lm(formula = mpg ~ log(weight) + sqrt(horsepower) + I(acceleration^2),
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1438  -2.5491  -0.4078   2.1189  15.6233
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       166.362313  10.374037  16.036  < 2e-16 ***
## log(weight)       -16.219953   1.709524  -9.488  < 2e-16 ***
## sqrt(horsepower)   -1.340211   0.332448  -4.031 6.68e-05 ***
## I(acceleration^2)  -0.001329   0.003620  -0.367    0.714
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.045 on 388 degrees of freedom
## Multiple R-squared:  0.7335, Adjusted R-squared:  0.7314
## F-statistic: 355.9 on 3 and 388 DF,  p-value: < 2.2e-16
```

From p values, log(weight) and sqrt(horsepower) have significant relationship to mpg.