EEEC E6690 hw1

Chong Hu ch3467

Sep 22, 2019

P1

(a)

$$
\begin{aligned}
(n-1)S^2 + n\bar{X}^2 &= \sum_{i=1}^{n}(X_i - \bar{X})^2 + n\bar{X}^2 \\
&= \sum_{i=1}^{n}(X_i^2 - 2X_i\bar{X} + \bar{X}^2) + n\bar{X}^2 \\
&= \sum_{i=1} X_i^2 - 2n\bar{X} + n(\bar{X})^2 + n\bar{X}^2 \\
&= \sum_{i=1} X_i^2
\end{aligned}
$$

Therefore,

$$\sum_{i=1}^{n} X_i^2 = (n-1)S^2 + n\bar{X}^2$$

(b) According to the question, we have

$$\mathrm{Var}[X_i] = \mathbb{E}[(X_i - \mathbb{E}[X_i])^2] = \sigma^2 \quad \forall i = 1, 2, 3, .., , n$$

and we can assume,

$$\mathbb{E}[X_i] = \mu$$

hence,

$$\mathrm{Var}[X_i] = \mathbb{E}[X_i^2] - \mu^2 = \sigma^2$$

Also, we have

$$
\begin{aligned}
\mathrm{Var}[\bar{X}] &= \frac{\sigma^2}{n} \\
\mathbb{E}[\bar{X}^2] &= \mathrm{Var}[\bar{X}] + \mathbb{E}[\bar{X}]^2 \\
&= \frac{\sigma^2}{n} + \mu^2
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}[S^2] &= \mathbb{E}\left[\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2\right] \\
&= \frac{1}{n-1}\mathbb{E}\left[\sum_{i=1}^{n}\left(X_i^2 - 2X_i\bar{X} + \bar{X}^2\right)\right] \\
&= \frac{1}{n-1}\mathbb{E}\left[\sum_{i=1}^{n} X_i^2 - n\bar{X}^2\right] \\
&= \frac{1}{n-1}\left(n(\sigma^2 + \mu^2) - n(\frac{\sigma^2}{n} + \mu^2)\right) \\
&= \sigma^2
\end{aligned}
$$

(c) Since X_i-s have i.i.d. normal/Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$

$$\begin{aligned}\mathrm{Cov}[\bar{X}, X_i - \bar{X}] &= \mathrm{Cov}[\bar{X}, X_i] - \mathrm{Var}[\bar{X}] \\ &= \frac{1}{n}\mathrm{Cov}[X_i + \sum_{j\neq i} X_j X_i] - \frac{\sigma^2}{n} \\ &= \frac{1}{n}\mathrm{Var}[X_i] - \frac{\sigma^2}{n} \\ &= 0\end{aligned}$$

therefore, $\bar{X}$ is independent of $X_i - \bar{X}$.

(d) Since sample variance

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

is a function of $X_i - \bar{X}$, which is independent of $\bar{X}$, sample mean is also independent of sample variance.

P2

Assume that $\bar{y} = \bar{x} = 0$, then we have,

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i)^2}{\sum_{i=1}^n y_i^2} = \frac{1}{\sum_{i=1}^n y_i^2}\sum_{i=1}^n\left(\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}x_i\right)^2 = \frac{1}{\sum_{i=1}^n y_i^2}\left(\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}\right)^2\sum_{i=1}^n x_i^2 = \frac{(\sum_{i=1}^n x_i y_i)^2}{\sum_{i=1}^n x_i^2\sum_{i=1}^n y_i^2}$$

and

$$r^2 = \frac{(\sum_{i=1}^n x_i y_i)^2}{\sum_{i=1}^n x_i^2\sum_{i=1}^n y_i^2}$$

Hence, $R^2 = r^2$

P3

simple linear regression

```
set.seed(1)
x = rnorm(100)
eps = rnorm(100, mean = 0, sd = 0.25)
y = -1 + 0.5*x + eps
```

The length of vector y is 100. In this linear model $\beta_0 = -1$ and $\beta_1 = 0.5$.

```
library(ggplot2)
scatter_fig <- ggplot(,aes(x=x, y=y)) + geom_point()
scatter_fig
```

From the scatter plot, we can see a rough line with slop 0.5. The data points in both x-direction and y-direction are centered in the middle.

```
simple_lm <-lm(y~x)
summary(simple_lm)
```

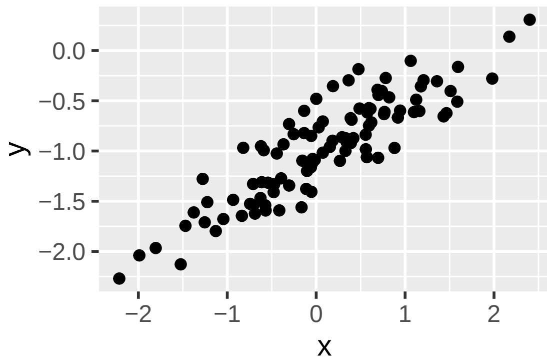

Figure 1: scatter plot of X and Y

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63   <2e-16 ***
## x            0.49973    0.02693   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

From the summary, we can clear see that $\hat{\beta}_0 =$ -1.0094232 and $\hat{\beta}_1 =$ 0.4997349. Both of them are only close to its true value. Notice that I used legend inside `ggplot2` instead of `legend()`.

```
scatter_fig2 <- scatter_fig + geom_abline( aes(intercept = simple_lm$coefficients[1],
                                              slope = simple_lm$coefficients[2],color = "red"),
                 linetype="dashed", size=1.5) +
  geom_smooth(method = "loess", aes(color = "blue"),
                 linetype="dashed", size=1.5, se = FALSE) +
  scale_colour_manual(name='Lines', labels = c("population regression", "least squares"),
                      values=c("blue", "red"))
scatter_fig2
```

```
quadratic_lm <- lm(y ~ x + I(x^2))
summary(quadratic_lm)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4913 -0.1563 -0.0322  0.1451  0.5675
```

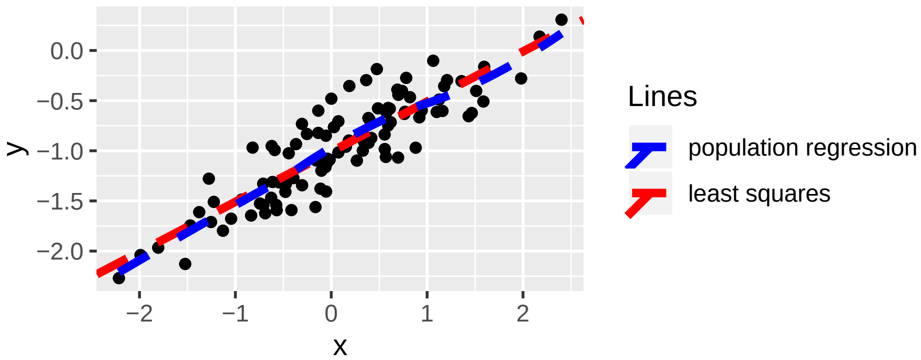

Figure 2: scatter plot of X and Y with regression lines

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98582    0.02941 -33.516   <2e-16 ***
## x            0.50429    0.02700  18.680   <2e-16 ***
## I(x^2)      -0.02973    0.02119  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2395 on 97 degrees of freedom
## Multiple R-squared:  0.7828, Adjusted R-squared:  0.7784
## F-statistic: 174.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

There is no evidence to show that quadratic term improves the model fit. Because `Pr(>|t|)` $= 0.164$, which is larger than 0.05.

simple linear regression with less noise

```
x = rnorm(100)
eps = rnorm(100, mean = 0, sd = 0.1)
y = -1 + 0.5*x + eps
```

The length of vector y is 100. In this linear model $\beta_0 = -1$ and $\beta_1 = 0.5$. Here let $X \sim \mathcal{N}(0, 0.1)$ to reduce noise.

```
library(ggplot2)
scatter_fig <- ggplot(,aes(x=x, y=y)) + geom_point()
scatter_fig
```

From the scatter plot, we can see a clear line with slop 0.5. The data points in both x-direction and y-direction are centered in the middle.

```
simple_less_lm <-lm(y~x)
summary(simple_less_lm)
```

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

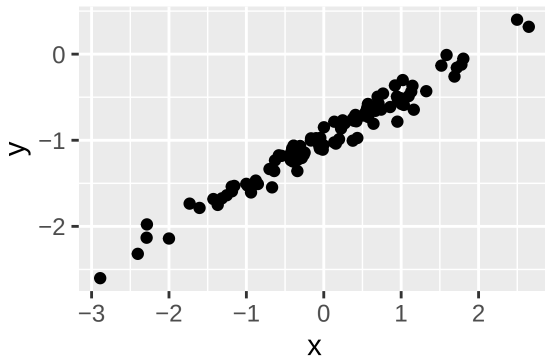

Figure 3: scatter plot of X and Y

```
##       Min        1Q    Median        3Q       Max
## -0.274179 -0.056139 -0.001749  0.067973  0.184843
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.995155   0.009910 -100.42   <2e-16 ***
## x            0.510622   0.009626   53.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09906 on 98 degrees of freedom
## Multiple R-squared:  0.9663, Adjusted R-squared:  0.966
## F-statistic:  2814 on 1 and 98 DF,  p-value: < 2.2e-16
```

From the summary, we can clear see that $\hat{\beta_0}$ = -0.995155 and $\hat{\beta_1}$ = 0.5106218. Both of them are very close to its true value. The significance is slightly more important than previous model. Notice that I used legend inside `ggplot2` instead of `legend()`. Compared with previous figure, the population regression line overlaps largely least squares line.

```r
scatter_fig2 <- scatter_fig + geom_abline( aes(intercept = simple_less_lm$coefficients[1],
                                                slope = simple_less_lm$coefficients[2],color = "red"),
                linetype="dashed", size=1.5) +
  geom_smooth(method = "loess", aes(color = "blue"),
                linetype="dashed", size=1.5, se = FALSE) +
  scale_colour_manual(name='Lines', labels = c("population regression", "least squares"),
                      values=c("blue", "red"))
scatter_fig2
```

simple linear regression with more noise

```r
x = rnorm(100)
eps = rnorm(100, mean = 0, sd = 1)
y = -1 + 0.5*x + eps
```

The length of vector y is 100. In this linear model $\beta_0 = -1$ and $\beta_1 = 0.5$. Here let $X \sim \mathcal{N}(0, 1)$ to add more noise.

```r
library(ggplot2)
scatter_fig <- ggplot(,aes(x=x, y=y)) + geom_point()
```

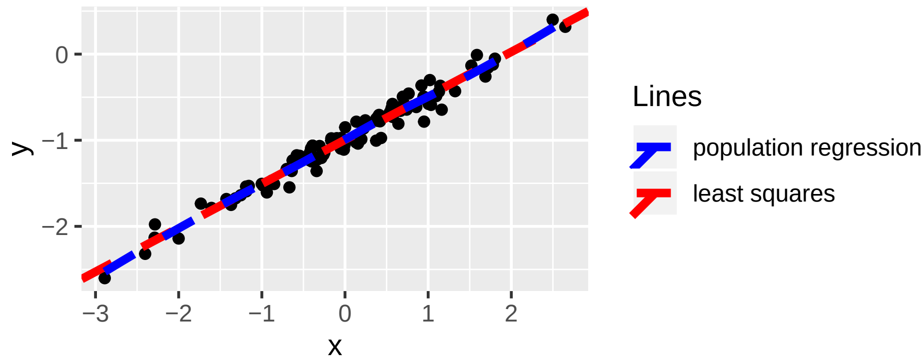

Figure 4: scatter plot of X and Y with regression lines

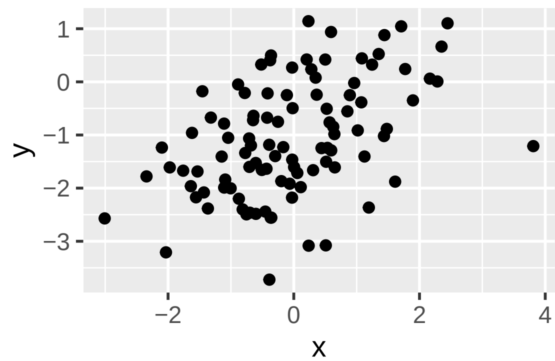

Figure 5: scatter plot of X and Y

```
scatter_fig
```

From the scatter plot, we can only see a bunch of data points. The data points in both x-direction and y-direction are centered in the middle.

```
simple_more_lm <-lm(y~x)
summary(simple_more_lm)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51014 -0.60549  0.02065  0.70483  2.08980
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.04745    0.09676 -10.825  < 2e-16 ***
## x            0.42505    0.08310   5.115 1.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9671 on 98 degrees of freedom
## Multiple R-squared:  0.2107, Adjusted R-squared:  0.2027
## F-statistic: 26.16 on 1 and 98 DF,  p-value: 1.56e-06
```

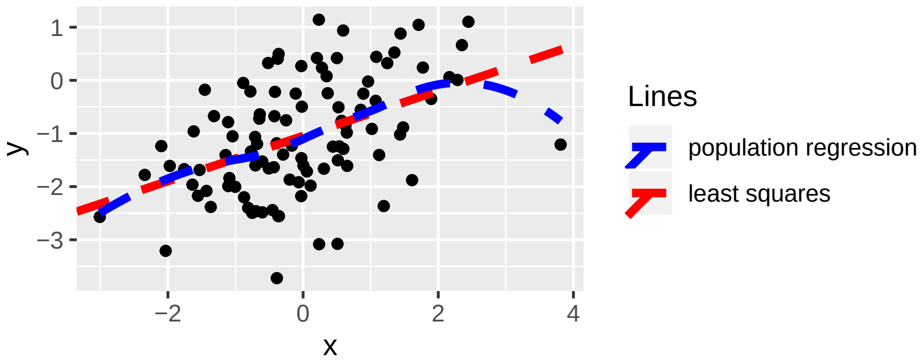

Figure 6: scatter plot of X and Y with regression lines

From the summary, we can clear see that $\hat{\beta}_0$ = -1.0474524 and $\hat{\beta}_1$ = 0.4250511. Both of them are far away from its true value. The significance is much worse than previous model. Notice that I used legend inside `ggplot2` instead of `legend()`. Compared with previous figure, the population regression line has a large difference with least squares line.

```
scatter_fig2 <- scatter_fig + geom_abline( aes(intercept = simple_more_lm$coefficients[1],
                                              slope = simple_more_lm$coefficients[2],color = "red"),
                 linetype="dashed", size=1.5) +
  geom_smooth(method = "loess", aes(color = "blue"),
                 linetype="dashed", size=1.5, se = FALSE) +
  scale_colour_manual(name='Lines', labels = c("population regression", "least squares"),
                      values=c("blue", "red"))
scatter_fig2
```

```
# "Confidence Interval for original data set"
confint(simple_lm , c("(Intercept)", "x"), level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.0575402 -0.9613061
## x            0.4462897  0.5531801
```

```
# "Confidence Interval for less noisy data set"
confint(simple_less_lm , c("(Intercept)", "x"), level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.0148210 -0.9754890
## x            0.4915195  0.5297242
```

```
# "Confidence Interval for noisier data set"
confint(simple_more_lm , c("(Intercept)", "x"), level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.2394772 -0.8554276
## x            0.2601391  0.5899632
```

Here are the confidence interval for the original data set, the noisier data set, and the less noisy data set. `(intercept)` represents $\hat{\beta}_0$ and `x` represents $\hat{\beta}_1$. While the confidence interval for original data set and less noisy data set are very close to each other, confidence interval for noiser data set are much wider due more noise.

P4

```r
adv.df = read.csv("Advertising.csv", header=T, na.string=",")
TV.lm <- lm(sales ~ TV, data = adv.df)
radio.lm <- lm(sales ~ radio, data = adv.df)
newspaper.lm <- lm(sales ~ newspaper, data = adv.df)

# "Confidence Interval for sales ~ TV"
confint(TV.lm , c("(Intercept)", "TV"), level = 0.92)
```

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

```r
# "Confidence Interval for sales ~ radio"
confint(radio.lm , c("(Intercept)", "radio"), level = 0.92)
```

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

```r
# "Confidence Interval for sales ~ newspaper"
confint(newspaper.lm , c("(Intercept)", "newspaper"), level = 0.92)
```

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

Here are 92% confidence intervals for $\hat{\beta}_0$ and $\hat{\beta}_1$ for three linear regressions of `sales` onto `newspaper`, `TV` and `radio`. Three scatterplots with confidence interval are shown below.

```r
plot(adv.df$TV, adv.df$sales, xlab="TV", ylab="sales", pch=20)
new_TV <- seq(min(adv.df$TV), max(adv.df$TV), length.out=100)
preds <- predict(TV.lm, newdata = data.frame(TV=new_TV),
                 interval = 'confidence', level = 0.92)
polygon(c(rev(new_TV), new_TV), c(rev(preds[ ,3]), preds[ ,2]), col = 'grey80', border = NA)
abline(TV.lm, col = 'blue')
# intervals
lines(new_TV, preds[ ,3], lty = 'dashed', col = 'red')
lines(new_TV, preds[ ,2], lty = 'dashed', col = 'red')
```

```r
plot(adv.df$newspaper, adv.df$sales, xlab="newspaper", ylab="sales", pch=20)
new_newspaper <- seq(min(adv.df$newspaper), max(adv.df$newspaper), length.out=100)
preds <- predict(newspaper.lm, newdata = data.frame(newspaper=new_newspaper),
                 interval = 'confidence', level = 0.92)
polygon(c(rev(new_newspaper), new_newspaper), c(rev(preds[ ,3]), preds[ ,2]), col = 'grey80', border =
abline(newspaper.lm, col = 'blue')
# intervals
lines(new_newspaper, preds[ ,3], lty = 'dashed', col = 'red')
lines(new_newspaper, preds[ ,2], lty = 'dashed', col = 'red')
```

```r
plot(adv.df$radio, adv.df$sales, xlab="radio", ylab="sales", pch=20)
new_radio <- seq(min(adv.df$radio), max(adv.df$radio), length.out=100)
preds <- predict(radio.lm, newdata = data.frame(radio=new_radio),
                 interval = 'confidence', level = 0.95)
polygon(c(rev(new_radio), new_radio), c(rev(preds[ ,3]), preds[ ,2]), col = 'grey80', border = NA)
```

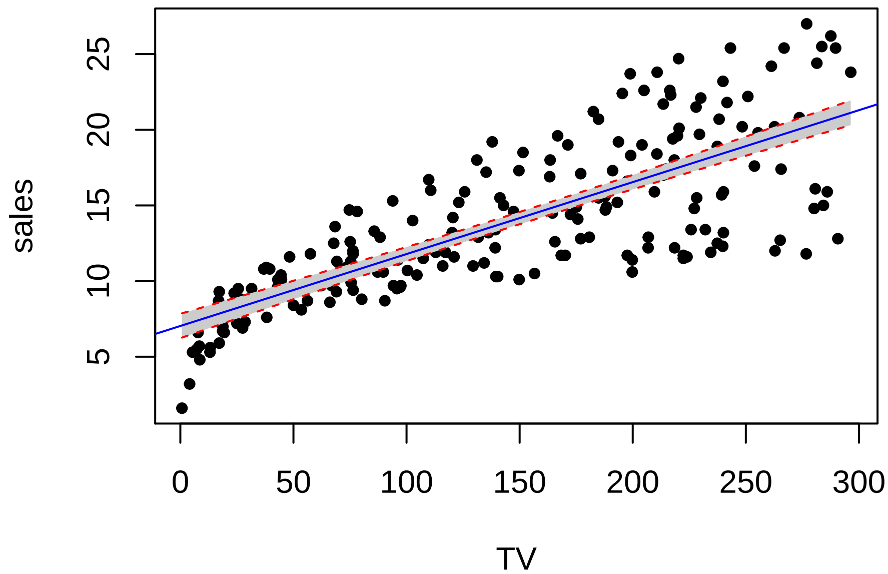

Figure 7: confidence interval for `sales` ~ `TV`

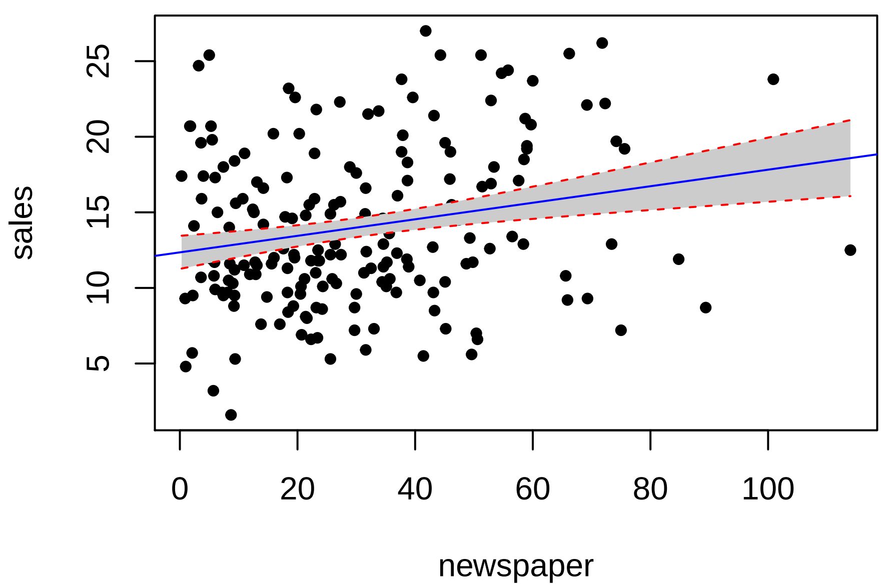

Figure 8: confidence interval for `sales` ~ `newspaper`

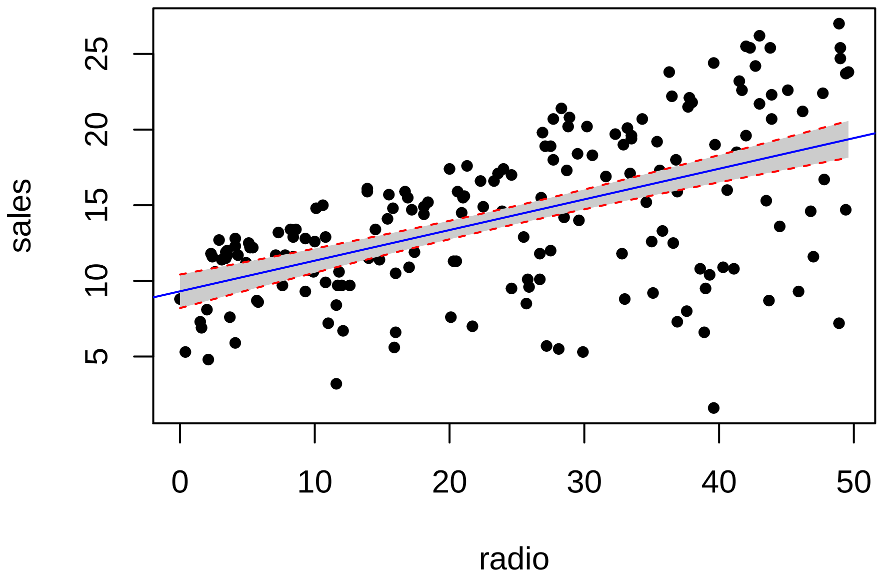

Figure 9: confidence interval for `sales` ~ `radio`

```
abline(radio.lm, col = 'blue')
# intervals
lines(new_radio, preds[ ,3], lty = 'dashed', col = 'red')
lines(new_radio, preds[ ,2], lty = 'dashed', col = 'red')
```

P5

```
Auto.df <- read.csv("Auto.csv", header=T, na.strings="?")
Auto.df <- na.omit(Auto.df)
```

```
pairs(Auto.df[,1:9])
```

Here is the correlation matrix between the variables exclude `name`

```
cor(Auto.df[,1:8])
```

```
##                     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
```

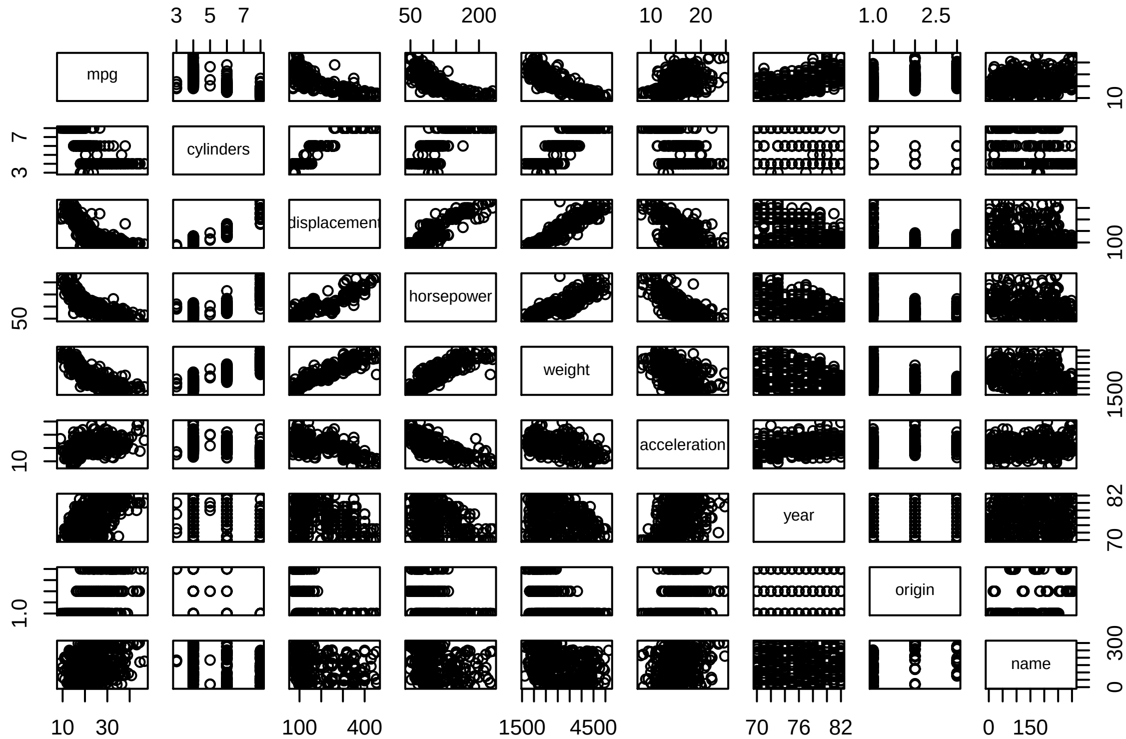

Figure 10: scatter matrix for Auto data set

```
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

```r
Auto.lm <- lm(mpg ~ cylinders + displacement + horsepower +
                weight + acceleration + year + origin,  data = Auto.df)
summary(Auto.lm)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = Auto.df)
##
## 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
```

Since the `Adjusted R-squared` term is 0.8182, the linear model could fit the relationship between predictors and response. From the summary of linear regression model using all numeric variables, we can clearly see that it is evident that `cylinders`, `horsepower` and `acceleration` are not related with `mpg`. The other variables are related with response. According to coefficients in summary, predictors `displacement`, `year` and `origin` have a positive correlation with `mpg`, while `weight` have a negative correlation with `mpg`. This result makes common sense, since as the car becomes heavier, mpg will decrease. And as time flies, the car also becomes more powerful and mpg also increases.

Since some of predictors don't have a statistically significant relationship to the response. Therefore, I will try a few different transformations of the variables.

```
Auto1.lm <- lm(mpg ~ cylinders + displacement + sqrt(horsepower) +
                 weight + acceleration + year + origin,  data = Auto.df)
summary(Auto1.lm)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + sqrt(horsepower) +
##     weight + acceleration + year + origin, data = Auto.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5240 -1.9910 -0.1687  1.8181 12.9211
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -6.0373910  5.5460041  -1.089 0.277012
## cylinders        -0.5222540  0.3166839  -1.649 0.099938 .
## displacement      0.0220542  0.0071987   3.064 0.002341 **
## sqrt(horsepower) -1.1434906  0.3113771  -3.672 0.000274 ***
## weight           -0.0054593  0.0006842  -7.979 1.72e-14 ***
## acceleration     -0.1021239  0.1038565  -0.983 0.326070
## year              0.7240379  0.0501791  14.429  < 2e-16 ***
## origin            1.5173206  0.2703470   5.612 3.83e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.277 on 384 degrees of freedom
## Multiple R-squared:  0.8269, Adjusted R-squared:  0.8237
## F-statistic:   262 on 7 and 384 DF,  p-value: < 2.2e-16
```

Here I used `sqrt(horsepower)` instead of original `horsepower`. From the summary, we can see that the new variable has a significant relationship to the response.

```
Auto2.lm <- lm(mpg ~ cylinders + displacement + sqrt(horsepower) +
                 weight + log(acceleration) + year + origin,  data = Auto.df)
summary(Auto2.lm)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + sqrt(horsepower) +
```

```
##     weight + log(acceleration) + year + origin, data = Auto.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7023 -2.0777 -0.1715  1.7945 12.9667
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.9841661  7.6330267   0.391   0.6960
## cylinders         -0.5052647  0.3154532  -1.602   0.1100
## displacement       0.0197672  0.0072902   2.711   0.0070 **
## sqrt(horsepower)  -1.3654827  0.3152258  -4.332 1.89e-05 ***
## weight            -0.0050067  0.0007057  -7.095 6.29e-12 ***
## log(acceleration) -3.3303267  1.6850727  -1.976   0.0488 *
## year               0.7205965  0.0499355  14.431  < 2e-16 ***
## origin             1.5009117  0.2694841   5.570 4.81e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.265 on 384 degrees of freedom
## Multiple R-squared:  0.8282, Adjusted R-squared:  0.825
## F-statistic: 264.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

And I also switch to use `log(acceleration)` instead of `acceleration`, the significance of that term is a little improved. But overall `Adjusted R-squared` only improved little.

```
Auto3.lm <- lm(log(mpg) ~ displacement + sqrt(horsepower) + weight +
                 log(acceleration) + year + origin,  data = Auto.df)
summary(Auto3.lm)
```

```
##
## Call:
## lm(formula = log(mpg) ~ displacement + sqrt(horsepower) + weight +
##     log(acceleration) + year + origin, data = Auto.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41972 -0.06686 -0.00329  0.06849  0.35582
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.464e+00  2.736e-01   9.005  < 2e-16 ***
## displacement       1.248e-04  1.976e-04   0.632  0.52799
## sqrt(horsepower)  -5.711e-02  1.133e-02  -5.041 7.16e-07 ***
## weight            -2.144e-04  2.531e-05  -8.470 5.24e-16 ***
## log(acceleration) -1.586e-01  6.061e-02  -2.617  0.00921 **
## year               2.889e-02  1.796e-03  16.081  < 2e-16 ***
## origin             3.873e-02  9.660e-03   4.009 7.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1174 on 385 degrees of freedom
## Multiple R-squared:  0.8825, Adjusted R-squared:  0.8807
## F-statistic: 482.1 on 6 and 385 DF,  p-value: < 2.2e-16
```

Also, I tried to use transformation on the response. I used `log(mpg)` to subsitute `mpg`. Athough the `displacement` term becomes not significant again, overall `Adjusted R-squared` is improved.

P6

$$
\begin{aligned}
\hat{\beta}_1 &= \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \\
&= \frac{20 \cdot 216.6 - 8.552 \cdot 398.2}{20 \cdot 5.196 - (8.552)^2} \\
&= 30.10
\end{aligned}
$$

$$
\begin{aligned}
\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\
&= 7.04
\end{aligned}
$$

$$
\begin{aligned}
\hat{\sigma}^2 &= \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\
&= \frac{1}{n-2}\sum_{i=1}^{n} y_i^2 + \hat{\beta}_0^{\ 2} + \hat{\beta}_1^{\ 2} x_i^2 - 2\hat{\beta}_0 y_i - 2\hat{\beta}_1 x_i y_i + 2\hat{\beta}_0\hat{\beta}_1 x_1 \\
&= \frac{1}{18}(9356 + 20 \cdot 7.04^2 + 30.10^2 \cdot 5.196 - 2 \cdot 7.04 \cdot 398.2 - 2 \cdot 30.10 \cdot 216.6 + 2 \cdot 7.04 \cdot 30.10 \cdot 8.552) \\
&= 1.8023
\end{aligned}
$$

$$
\hat{y}_{x=0.5} = \hat{\beta}_0 + \hat{\beta}_1 x = 22.09
$$

$$
R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{32.44}{1427.84} = 0.977
$$

P7

Null hypothesis:

$$
\mathcal{H}_0 = \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 0
$$

Here we have

$$
k = 6 \quad n = 45
$$

$$
\text{TSS} = 11.62 \quad \text{RSS} = 8.95
$$

$$
\mathcal{F}_{k,n-k-1} = \frac{(\text{TSS} - \text{RSS})/k}{\text{RSS}/(n-k-1)} = 1.889
$$

```r
# p-value
1-pf(1.889385,6,45-6-1)
```

```
## [1] 0.1079353
```

Also, at $\alpha = 0.05$, we have

$$f_{\alpha,k,n-k-1} = 2.349027$$

Therefore, $\mathcal{F}_{k,n-k-1} < f_{\alpha,k,n-k-1}$, then $\mathcal{H}_0$ cannot be rejected, which means there is reason to believe that the regression is not significant.