STA206_hw5

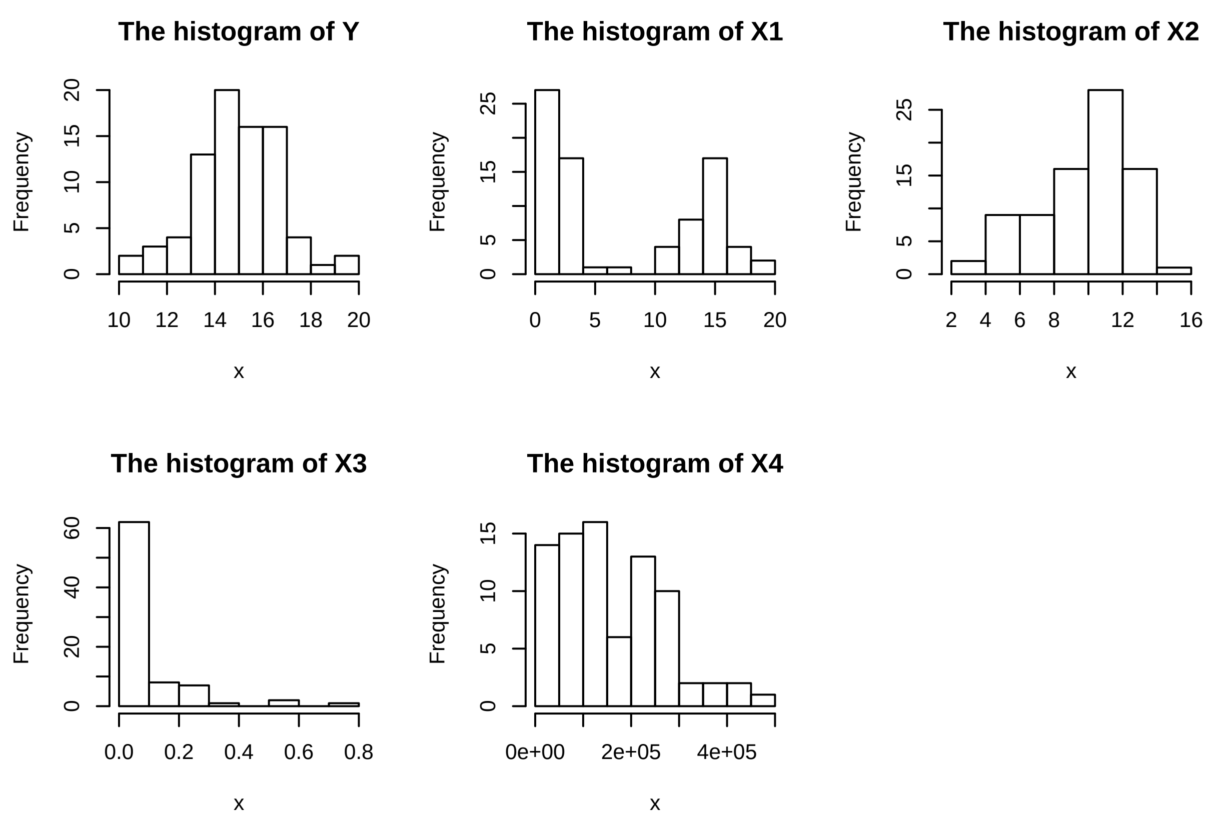
Zhen Zhang

November 7, 2015

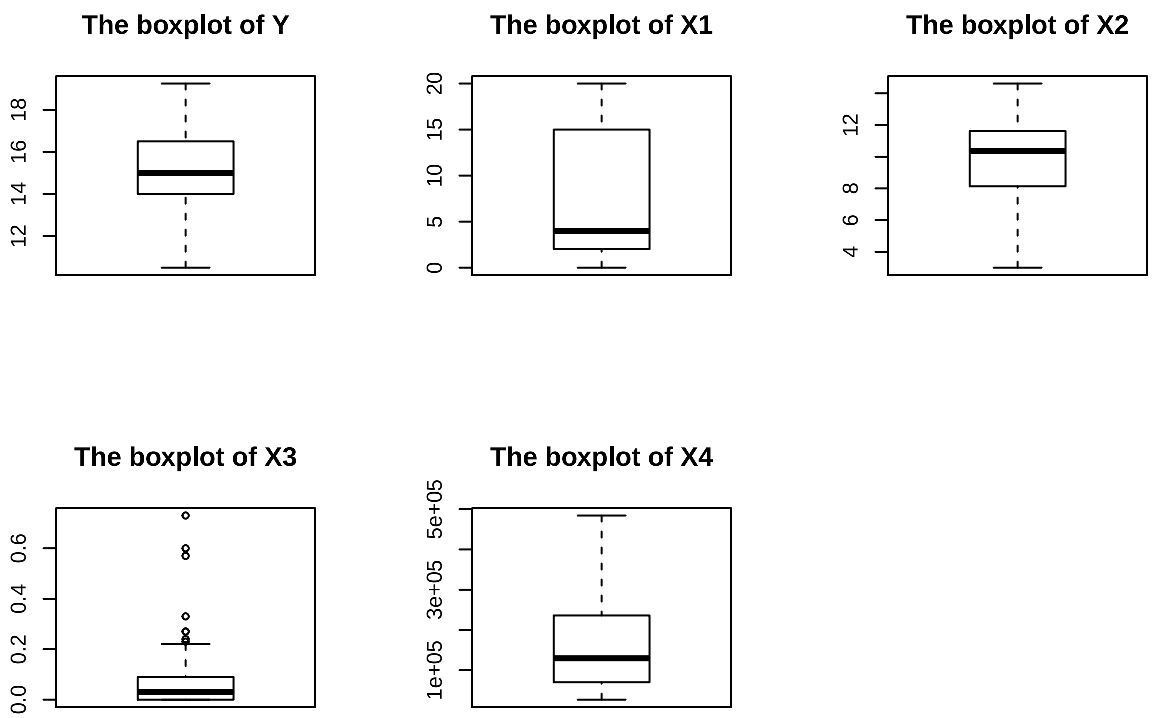
Problem 2

(a)

```
property <- read.table("STA206_hw4_data_property.txt")
names(property) <- c("Y", paste0("X", 1:4))
par(mfrow = c(2,3))
plot_hist <- lapply(1:5, function(i) {
  x <- property[, i]
  hist(x, main = paste0("The histogram of ", names(property)[i]))
})
```




```
par(mfrow = c(2,3))
plot_boxplot <- lapply(1:5, function(i) {
  x <- property[, i]
  boxplot(x, main = paste0("The boxplot of ", names(property)[i]))
})
```



The distributions for each variable are different. For X_1, it has the lowest value at the middle, X_2 and Y are all almost normal distributions, X_3 is almost monotonically decreasing, and has many outliers.

The scales are different for different variables. For instance, X_4 is much larger than the others.

(b)

The sample mean and sample standard deviation of each variable:

```
(property_mean <- apply(property, 2, mean))
```

```
##            Y           X1           X2           X3           X4
## 1.513889e+01 7.864198e+00 9.688148e+00 8.098765e-02 1.606333e+05
```

```
(property_sd <- apply(property, 2, sd))
```

```
##            Y           X1           X2           X3           X4
## 1.719584e+00 6.632784e+00 2.583169e+00 1.345512e-01 1.090990e+05
```

The transformation is:

```
property_tran <- sapply(1:5, function(i) {
  (property[, i] - property_mean[i]) / (sqrt(nrow(property) - 1) * property_sd[i])
})
property_tran <- as.data.frame(property_tran)
colnames(property_tran) <- c("Y", paste0("X", 1:4))
```

The tranformed sample mean and sample standard deviation of each variable:

```
(property_tran_mean <- apply(property_tran, 2, mean))
```

```
##             Y            X1            X2            X3            X4
## -2.573307e-17 -5.078600e-18  5.541478e-18 -1.233447e-18  1.426588e-17
```

```
(property_tran_sd <- apply(property_tran, 2, sd))
```

```
##         Y        X1        X2        X3        X4
## 0.1118034 0.1118034 0.1118034 0.1118034 0.1118034
```

So the mean for the transformed ones are 0, and standard error is $\frac{1}{\sqrt{81-1}} = 0.1118034$

(c)

The model is:

$$Y^* = \beta_0 + \beta_1 X_1^* + \beta_2 X_2^* + \beta_3 X_3^* + \beta_4 X_4^*$$

Fit the model:

```
fit1 <- lm(Y ~ ., data = property_tran)
summary(fit1)
```

```
##
## Call:
## lm(formula = Y ~ ., data = property_tran)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.207223 -0.038429 -0.005914  0.036276  0.191422
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.178e-17  8.213e-03   0.000     1.00
## X1          -5.479e-01  8.232e-02  -6.655 3.89e-09 ***
## X2           4.236e-01  9.490e-02   4.464 2.75e-05 ***
## X3           4.846e-02  8.504e-02   0.570     0.57
## X4           5.028e-01  8.786e-02   5.722 1.98e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07392 on 76 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5629
## F-statistic: 26.76 on 4 and 76 DF,  p-value: 7.272e-14
```

To compare easily, I also fit the howework 4, problem 4 model here:

```r
fit2 <- lm(Y ~ ., data = property)
summary(fit2)
```

```
##
## Call:
## lm(formula = Y ~ ., data = property)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1872 -0.5911 -0.0910  0.5579  2.9441
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.220e+01  5.780e-01  21.110  < 2e-16 ***
## X1          -1.420e-01  2.134e-02  -6.655 3.89e-09 ***
## X2           2.820e-01  6.317e-02   4.464 2.75e-05 ***
## X3           6.193e-01  1.087e+00   0.570     0.57
## X4           7.924e-06  1.385e-06   5.722 1.98e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 76 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5629
## F-statistic: 26.76 on 4 and 76 DF,  p-value: 7.272e-14
```

The intercept is 0.

```r
# coefficients convert to the original model
fit1$coefficients[2:5] * property_sd[1] / property_sd[2:5]
```

```
##            X1            X2            X3            X4
## -1.420336e-01  2.820165e-01  6.193435e-01  7.924302e-06
```

The results are identical with that of howework 4, problem 4:

```r
# coefficients of the original model
fit2$coefficients[2:5]
```

```
##            X1            X2            X3            X4
## -1.420336e-01  2.820165e-01  6.193435e-01  7.924302e-06
```

(d)

The standard errors in the standardized model is:

```r
summary(fit1)$coefficients[2:5, 2]
```

```
##         X1         X2         X3         X4
## 0.08232278 0.09489786 0.08503913 0.08785704
```

Now convert it to the original models standard errors, and compare them with that of the original models:

```r
# convert to the original model, sd
summary(fit1)$coefficients[2:5, 2] * property_sd[1] / property_sd[2:5]
```

```
##           X1           X2           X3           X4
## 2.134261e-02 6.317235e-02 1.086813e+00 1.384775e-06
```

```r
# original model, sd
summary(fit2)$coefficients[2:5, 2]
```

```
##           X1           X2           X3           X4
## 2.134261e-02 6.317235e-02 1.086813e+00 1.384775e-06
```

(e)

The SSE for the standardized model is $0.07392^2 * 76 = 0.41525$, and SSTO is 0.41525 / (1 - 0.5847) = 0.9998796, SSR is SSTO - SSE = 0.5846296.

The SSE for the original model is $1.137^2 * 76 = 98.231$, and SSTO is 98.231 / (1 - 0.5847) = 236.5302, SSR is SSTO - SSE = 138.2992.

The relationship:

$$\sigma^* = \frac{\sigma}{\sqrt{n-1} * S_Y}$$

So

$$SSE^* = \frac{SSE}{(n-1) * S_Y^2}$$

In this question, if we convert SSE to the original model from the standardized model,

```r
0.41525 * (nrow(property) - 1) * property_sd[1]^2
```

```
##       Y
## 98.2305
```

which is identical to the original model's SSE.

(f)

The r square for the standardized model is and the original model are:

```r
sapply(c('r.squared', 'adj.r.squared'), function(x) summary(fit1)[[x]])
```

```
##     r.squared adj.r.squared
##     0.5847496     0.5628943
```

```r
sapply(c('r.squared', 'adj.r.squared'), function(x) summary(fit2)[[x]])
```

```
##     r.squared adj.r.squared
##     0.5847496     0.5628943
```

They are the same.

Problem 3

(a)

The correlation matrix of $\mathbf{r}_{xx}$ is:

```r
(XX <- cor(property_tran[2:5]))
```

```
##            X1         X2          X3         X4
## X1  1.0000000  0.3888264 -0.25266347 0.28858350
## X2  0.3888264  1.0000000 -0.37976174 0.44069713
## X3 -0.2526635 -0.3797617  1.00000000 0.08061073
## X4  0.2885835  0.4406971  0.08061073 1.00000000
```

The correlation matrix of $\mathbf{r}_{xy}$ is:

```r
(XY <- cor(property_tran[2:5], property_tran[1]))
```

```
##              Y
## X1 -0.25028456
## X2  0.41378716
## X3  0.06652647
## X4  0.53526237
```

Now calculate $X^{'}X$ and $X^{'}Y$:

```r
t(as.matrix(property_tran[2:5])) %*% as.matrix(property_tran[2:5])
```

```
##            X1         X2          X3         X4
## X1  1.0000000  0.3888264 -0.25266347 0.28858350
## X2  0.3888264  1.0000000 -0.37976174 0.44069713
## X3 -0.2526635 -0.3797617  1.00000000 0.08061073
## X4  0.2885835  0.4406971  0.08061073 1.00000000
```

```r
t(as.matrix(property_tran[2:5])) %*% as.matrix(property_tran[1])
```

```
##              Y
## X1 -0.25028456
## X2  0.41378716
## X3  0.06652647
## X4  0.53526237
```

Clearly, $X^{'}X = \mathbf{r}_{xx}$, $X^{'}Y = \mathbf{r}_{xy}$

(b)

$\mathbf{r}_{xx}^{-1}$ is:

```
(XXInv <- solve(XX))
```

```
##            X1         X2         X3         X4
## X1  1.2403482 -0.2870567  0.2244927 -0.2495354
## X2 -0.2870567  1.6482246  0.6092380 -0.6926391
## X3  0.2244927  0.6092380  1.3235525 -0.4399669
## X4 -0.2495354 -0.6926391 -0.4399669  1.4127219
```

So VIF_k are:

```
(VIF <- sapply(1:4, function(i) XXInv[i, i]))
```

```
## [1] 1.240348 1.648225 1.323552 1.412722
```

Now regress X_k on the other X_i $(1 <= i! = k <= 4)$, and get their r squared, use them to calculate VIF_k, then combine together:

```
(VIF2 <- sapply(1:4, function(i) {
  fit <- lm(as.formula(paste(paste0("X",i), '~', '.')), data = property[, -1])
  1 / (1 - summary(fit)$r.squared)
}))
```

```
## [1] 1.240348 1.648225 1.323552 1.412722
```

They are identical. So, the degree of multicollinearity is not very large, based on VIF value.

(c)

The regression model for relating Y to X_4 is:

```
fit3 <- lm(Y ~ X4, data = property)
summary(fit3)
```

```
##
## Call:
## lm(formula = Y ~ X4, data = property)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1390 -0.7930  0.2890  0.9653  3.4415
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.378e+01  2.903e-01  47.482  < 2e-16 ***
```

```
## X4          8.437e-06  1.498e-06   5.632 2.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.462 on 79 degrees of freedom
## Multiple R-squared:  0.2865, Adjusted R-squared:  0.2775
## F-statistic: 31.72 on 1 and 79 DF,  p-value: 2.628e-07
```

The regression model for relating Y to X_3 and X_4 is:

```
fit4 <- lm(Y ~ X3 + X4, data = property)
summary(fit4)
```

```
##
## Call:
## lm(formula = Y ~ X3 + X4, data = property)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1886 -0.7879  0.3140  0.9820  3.4021
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.376e+01  3.027e-01  45.469  < 2e-16 ***
## X3          3.007e-01  1.226e+00   0.245    0.807
## X4          8.407e-06  1.512e-06   5.561 3.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.47 on 78 degrees of freedom
## Multiple R-squared:  0.2871, Adjusted R-squared:  0.2688
## F-statistic:  15.7 on 2 and 78 DF,  p-value: 1.859e-06
```

The estimated regression coefficients of X_4 in these two models are all $8.407 * 10^{-6}$.

```
anova(fit3)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## X4         1  67.775  67.775  31.723 2.628e-07 ***
## Residuals 79 168.782   2.136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit4)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
```

```
## X3          1   1.047   1.047  0.4842    0.4886
## X4          1  66.858  66.858 30.9213 3.626e-07 ***
## Residuals 78 168.652   2.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So $SSR(X_4) = 67.775$ and $SSR(X_4|X_3) = 66.858$, almost identical. This means that X_4 is uncorrelated with X_3. This can be seen when add X_3 into the regreesion model, but the coefficient of X_4 does not change.

(d)

The regression model for relating Y to X_2 is:

```r
fit5 <- lm(Y ~ X2, data = property)
summary(fit5)
```

```
##
## Call:
## lm(formula = Y ~ X2, data = property)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5733 -0.9093 -0.1559  0.8290  4.7607
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.47026    0.68337   18.25  < 2e-16 ***
## X2           0.27545    0.06818    4.04 0.000123 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.575 on 79 degrees of freedom
## Multiple R-squared:  0.1712, Adjusted R-squared:  0.1607
## F-statistic: 16.32 on 1 and 79 DF,  p-value: 0.0001231
```

The regression model for relating Y to X_2 and X_4 is:

```r
fit6 <- lm(Y ~ X4 + X2, data = property)
summary(fit6)
```

```
##
## Call:
## lm(formula = Y ~ X4 + X2, data = property)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1949 -0.8982  0.2719  1.0263  3.9098
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.261e+01  6.211e-01  20.296  < 2e-16 ***
## X4          6.903e-06  1.632e-06   4.229 6.34e-05 ***
```

```
## X2          1.470e-01  6.895e-02   2.132   0.0362 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.43 on 78 degrees of freedom
## Multiple R-squared:  0.3258, Adjusted R-squared:  0.3085
## F-statistic: 18.84 on 2 and 78 DF,  p-value: 2.105e-07
```

The estimated regression coefficients of X_2 in the first model is 0.27545, in the second model is 0.1470 The second one is smaller, almost half of that of the first model.

```r
anova(fit5)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## X2         1  40.503  40.503  16.321 0.0001231 ***
## Residuals 79 196.054   2.482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit6)
```

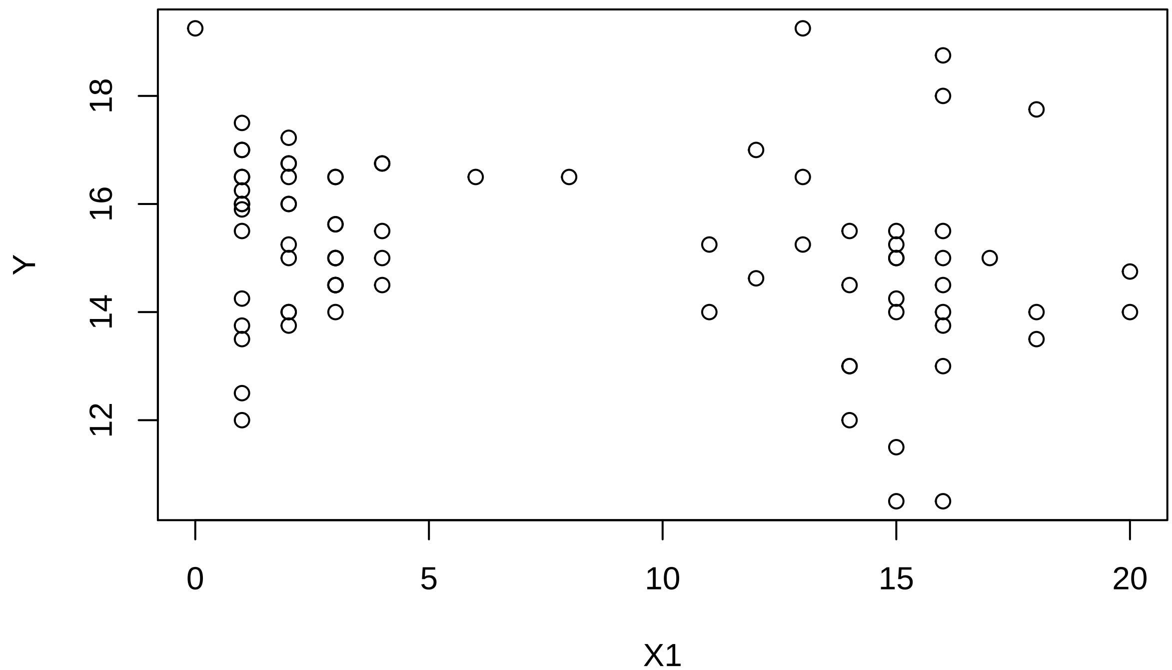
```
## Analysis of Variance Table
##
## Response: Y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## X4         1  67.775  67.775 33.1457 1.611e-07 ***
## X2         1   9.291   9.291  4.5438   0.03619 *
## Residuals 78 159.491   2.045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So $SSR(X_2) = 40.503$ and $SSR(X_2|X_4) = 9.291$, the second one much smaller. This means that X_2 and X_4 have high collinearity.

Problem 4

(a)

```r
with(property, plot(X1, Y))
```



Y distributes at the two sides of X_1.

(b)

```
# center X1
property$X1_center <- property$X1 - mean(property$X1)
fit8 <- lm(Y ~ X1_center + X2 + X4 + I(X1_center^2), data = property)
summary(fit8)
```

```
##
## Call:
## lm(formula = Y ~ X1_center + X2 + X4 + I(X1_center^2), data = property)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89596 -0.62547 -0.08907  0.62793  2.68309
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.019e+01  6.709e-01  15.188  < 2e-16 ***
## X1_center      -1.818e-01  2.551e-02  -7.125 5.10e-10 ***
## X2              3.140e-01  5.880e-02   5.340 9.33e-07 ***
## X4              8.046e-06  1.267e-06   6.351 1.42e-08 ***
## I(X1_center^2)  1.415e-02  5.821e-03   2.431   0.0174 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.097 on 76 degrees of freedom
## Multiple R-squared:  0.6131, Adjusted R-squared:  0.5927
## F-statistic:  30.1 on 4 and 76 DF,  p-value: 5.203e-15
```

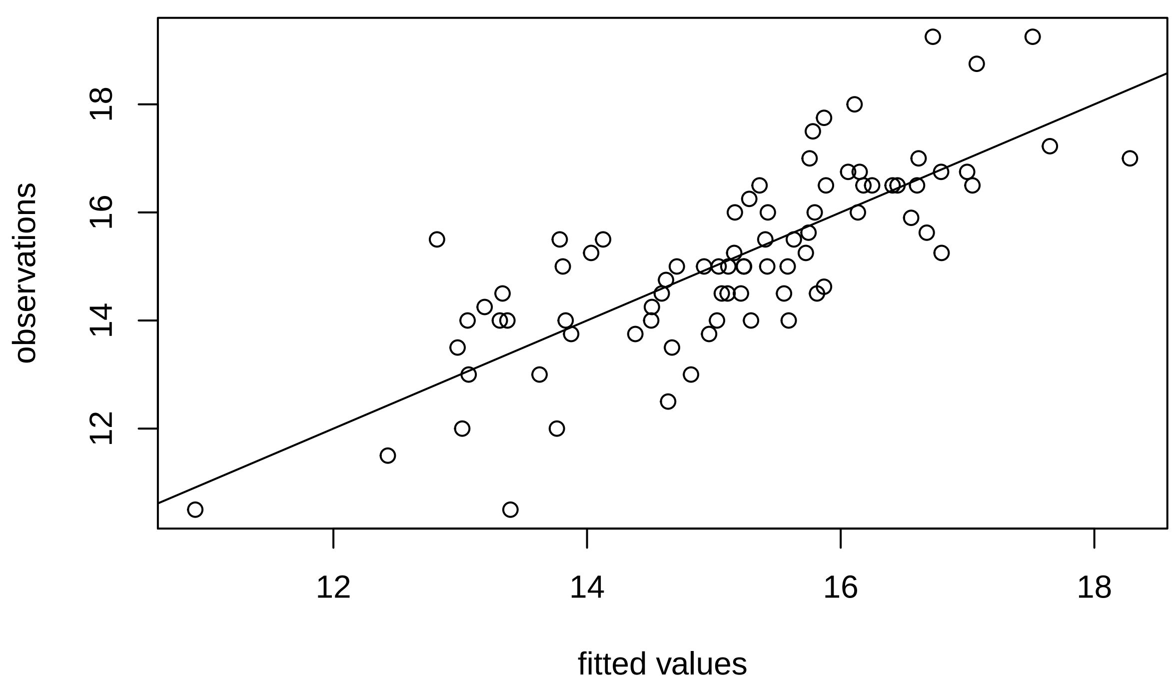
Model equation: $Y_i = \beta_0 + \beta_1 \tilde{X_{i1}} + \beta_2 X_{i2} + \beta_3 X_{i4} + \beta_4 \tilde{X_{i1}^2} + \epsilon$

The fitted model is: $\hat{Y_i} = 10.19 - 0.1818 * \tilde{X_{i1}} + 0.314 * X_{i2} + 8.046 * 10^{-6} * X_{i4} + 0.01415 * \tilde{X_{i1}^2}$

In terms of the original age of property X_1: $\hat{Y_i} = 10.19 - 0.1818 * (X_1 - \bar{X_1}) + 0.314 * X_2 + 8.046 * 10^{-6} * X_4 + 0.01415 * (X_1 - \bar{X_1})^2$, which is $\hat{Y_i} = 20.5 - 0.106 * X_1 + 0.314 * X_2 + 8.046 * 10^{-6} * X_4 + 0.01415 * X_1^2$

The observations Y against the fitted values $\hat{Y}$ plot:

```r
plot(fit8$fitted.values, property$Y, xlab = 'fitted values', ylab = 'observations')
abline(0, 1)
```



The points scatter evenly along both sides of the regression line, which means this line is a good estimation.

(c)

The model 2 from Homework 4:

```r
fit9 <- lm(Y ~ X1 + X2 + X4, data = property)
summary(fit9)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X4, data = property)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0620 -0.6437 -0.1013  0.5672  2.9583
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.237e+01  4.928e-01  25.100  < 2e-16 ***
## X1          -1.442e-01  2.092e-02  -6.891 1.33e-09 ***
## X2           2.672e-01  5.729e-02   4.663 1.29e-05 ***
## X4           8.178e-06  1.305e-06   6.265 1.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.132 on 77 degrees of freedom
## Multiple R-squared:  0.583,  Adjusted R-squared:  0.5667
## F-statistic: 35.88 on 3 and 77 DF,  p-value: 1.295e-14
```

Now compare their r square:

```r
sapply(list(fit8 = fit8, fit9 = fit9), function(x) {
  sapply(c('r.squared', 'adj.r.squared'), function(y) summary(x)[[y]])
})
```

```
##                    fit8      fit9
## r.squared     0.6130541 0.5829752
## adj.r.squared 0.5926885 0.5667275
```

Here we can see the r square and adjusted r square of the above model is greater than the model 2 in homework 4. This means the above model is better.

(d)

Model equation: $Y = \beta_0 + \beta_1 \tilde{X}_1 + \beta_2 X_2 + \beta_3 X_4 + \beta_4 \tilde{X}_1^2$

Null & alternative hypotheses: $H_0 : \beta_4 = 0$ vs $H_1 : \beta_4 \neq 0$

Test statistic: t test, with $T^* = \frac{\hat{\beta}_4}{se(\hat{\beta}_4)} = 2.431$

Null distribution: under H_0, $T^* \sim t_{(0.975,76)} = 1.991673$

Decision rule: reject H_0 if $|T^*| > t_{(0.975,76)}$

Conclusion: since $T^* = 2.431 > t_{(0.975,76)} = 1.991673$, then reject H_0, meaning $\tilde{X}_1$ cannot be dropped from the model at level 0.05.

(e)

```r
newdata = data.frame(X1 = 4, X2 = 10, X4 = 80000, X1_center = 4 - mean(property$X1))
lapply(list(fit8 = fit8, fit9 = fit9), function(x) {
  predict(x, newdata, interval = "prediction", level = 0.99)
})
```

```
## $fit8
##        fit      lwr      upr
## 1 14.88699 11.93875 17.83524
##
## $fit9
##        fit      lwr      upr
## 1 15.11985 12.09134 18.14836
```

So the predicted value under the model in this problem is 11.93875, and the prediction interval is smaller in the model in this problem, compared with the model 2 in homework 4.