mlds401_hw2

2024-10-06

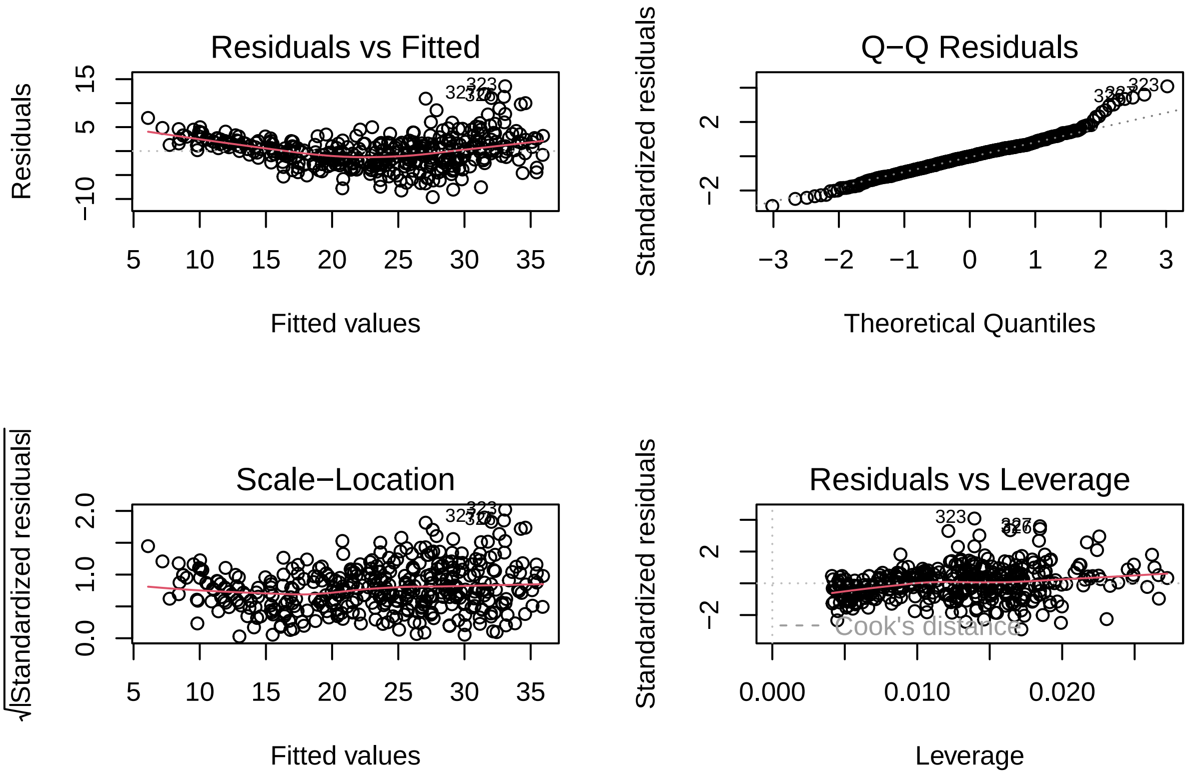
```r
library(readr)

auto <- read.table("auto.txt", sep = "", header = T)
remove <- which(is.na(auto),arr.ind = T)[,1]
auto <- auto[-remove,]
auto$origin <- factor((auto$origin), labels = c("US", "Europe", "Japan"))

table(auto$origin)
```

```
##
##     US Europe  Japan
##    245     68     79
```

```r
model1 <- lm(mpg ~ origin + weight + year, data = auto)

summary(model1)
```

```
##
## Call:
## lm(formula = mpg ~ origin + weight + year, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6025 -2.1132 -0.0206  1.7617 13.5261
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.831e+01  4.017e+00  -4.557 6.96e-06 ***
## originEurope  1.976e+00  5.180e-01   3.815 0.000158 ***
## originJapan   2.215e+00  5.188e-01   4.268 2.48e-05 ***
## weight       -5.887e-03  2.599e-04 -22.647  < 2e-16 ***
## year          7.698e-01  4.867e-02  15.818  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.337 on 387 degrees of freedom
## Multiple R-squared:  0.819,  Adjusted R-squared:  0.8172
## F-statistic: 437.9 on 4 and 387 DF,  p-value: < 2.2e-16
```

```r
# Diagnostic plots
par(mfrow=c(2,2))
plot(model1)
```



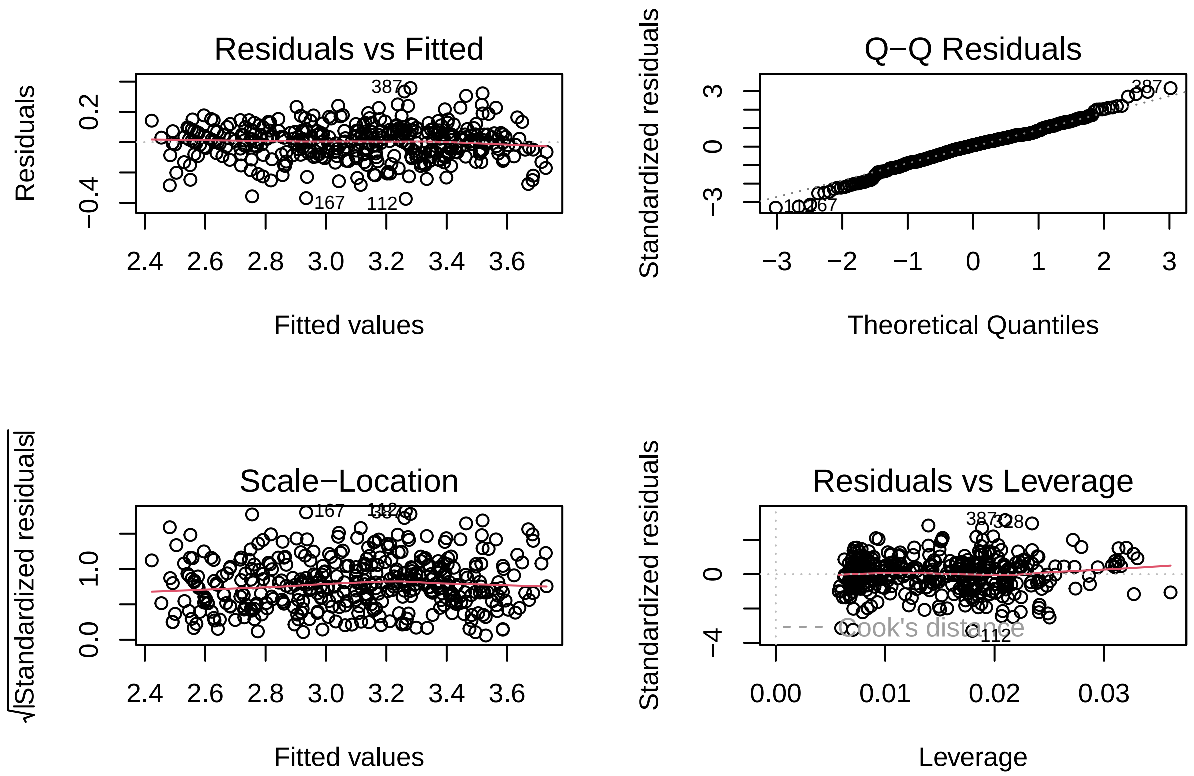
b) The key assumptions violated are non-linearity, heteroscedasticity, and some deviations from normality. The "Residuals vs Fitted" plot reveals a slight fitted curve, suggesting potential non-linearity, and the Q-Q plot indicates some deviation from normality in the right tail. Additionally, the "Scale-Location" plot suggests heteroscedasticity, with an increasing spread of residuals for larger fitted values. There are a few points of concern regarding high leverage in the "Residuals vs Leverage" plot, though no extreme outliers.

```r
auto$log_mpg <- log(auto$mpg)
auto$log_weight <- log(auto$weight)
auto$year_squared <- auto$year^2

model2 <- lm(log_mpg ~ origin + log_weight + year + year_squared, data = auto)

summary(model2)
```

```
##
## Call:
## lm(formula = log_mpg ~ origin + log_weight + year + year_squared,
##     data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37401 -0.06907  0.00861  0.06996  0.35753
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    18.5012369  2.7077793   6.833 3.26e-11 ***
## originEurope    0.0661562  0.0179622   3.683 0.000263 ***
## originJapan     0.0321082  0.0181521   1.769 0.077709 .
## log_weight     -0.8746164  0.0274593 -31.851  < 2e-16 ***
```

```
## year          -0.2569025  0.0718420  -3.576 0.000393 ***
## year_squared  0.0019113  0.0004729   4.042 6.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1142 on 386 degrees of freedom
## Multiple R-squared:  0.8886, Adjusted R-squared:  0.8871
## F-statistic: 615.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

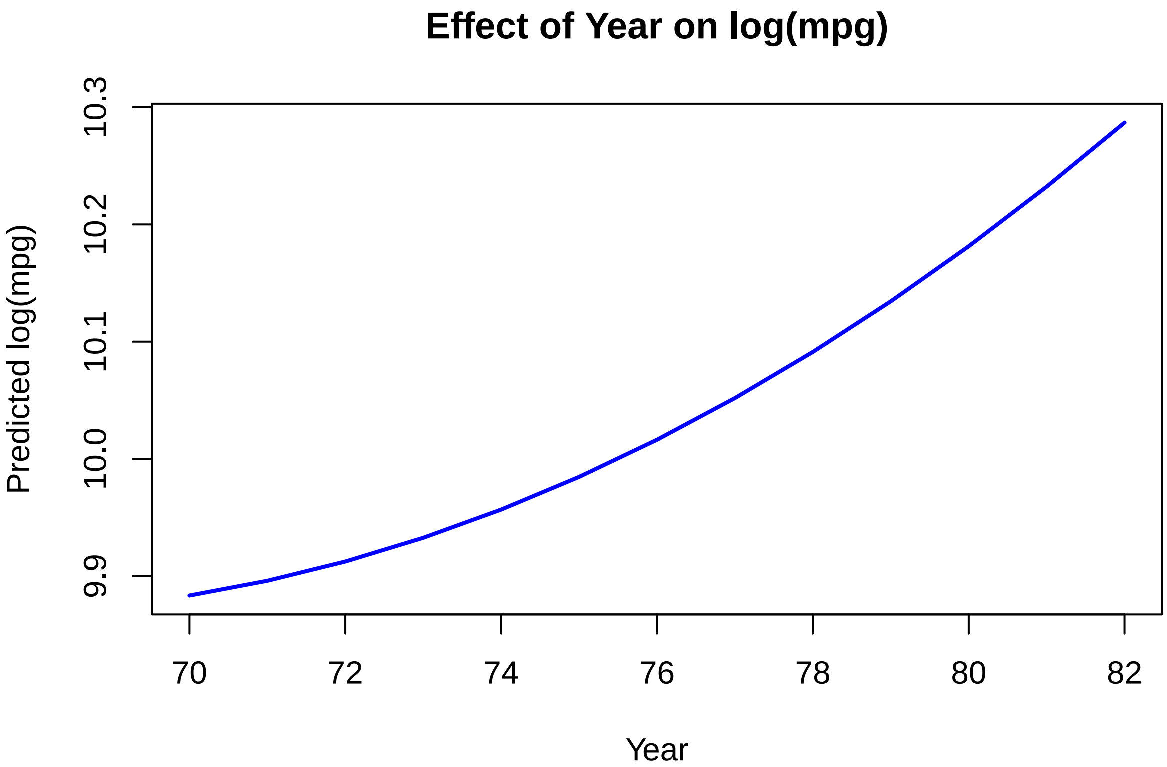
```r
# Diagnostic plots
par(mfrow=c(2,2))
plot(model2)
```



c) The model assumptions are roughly satisified. The "Residuals vs Fitted" plot shows more evenly scattered residuals, suggesting an improved linear fit. The Q-Q plot exhibits less deviation from normality, and the "Scale-Location" plot suggests a more constant variance, indicating reduced heteroscedasticity. There are still some influential points in the "Residuals vs Leverage" plot, but overall, problem (c) improve model assumptions like linearity, normality, and homoscedasticity compared to problem (b). From table summary, we can also see that the regression results in problem (c) demonstrate a better model fit compared to problem (b), with a higher adjusted R-squared (0.888 vs 0.819) and a significantly lower residual standard error (0.1142 vs 3.337), indicating more accurate predictions.

```r
# Define a range for the year variable
year_range <- seq(min(auto$year), max(auto$year))

# Calculate log(mpg) based on the regression model
log_mpg_pred <- 18.5012369 - 0.2569025 * year_range + 0.0019113 * year_range^2

# Plot the relationship
plot(year_range, log_mpg_pred, type = "l", col = "blue", lwd = 2,
     xlab = "Year", ylab = "Predicted log(mpg)",
```

```r
    main = "Effect of Year on log(mpg)")
```




```r
# Calculate the minimum point
min_year <- 0.2569025 / (2 * 0.0019113)
min_year
```

```
## [1] 67.20622
```

d) Based on the plot, it is U-shaped curve, where the minimum point represents the year at which log(mpg) is lowest. The minimum point for year, where log(mpg) is lowest, occurs around 67.21, which is around 1967.

e) The coefficient of -0.8746164 for log(weight) means a 1% increase in weight corresponds to a 0.8746164% decrease in mpg. In contrast, the coefficient of -5.887e-03 for weight(unlogged) means that for each unit increase in weight, mpg decreases by 5.887e-03 units.

Q2-Q3 HW2

Sunday, October 6, 2024 5:17 PM

2a) $\mathbb{E}(Y_i)=\mu \quad \mathbb{V}(Y_i)=\sigma_i^2 \ (i=1,2) \quad \bar{y}_w = w_1Y_1 + w_2Y_2$

$w_1 = w \quad w_2 = 1-w$

Under what circumstance will $\bar{y}_w$ = unbiased estimator of μ

$= E$

$E(\bar{y}_w) = E(\overset{con}{w}Y_1 + (1-\overset{con}{w})Y_2)$

$= w\,E(Y_1) + (1-w)\,E(Y_2)$

$= w\mu + (1-w)\mu$

$= \mu(w+1-w) \rightarrow \mu$ = expected value of $\bar{y}_w$.

$\boxed{\text{Because } E(\bar{y}_w)=\mu,\ \bar{y}_w \text{ is an unbiased estimator of } \mu}$

b) Variance of $\bar{y}_w$

$V(\bar{y}_w) = V(wY_1 + (1-w)Y_2)$

$= w^2V(Y_1) + (1-w)^2V(Y_2)$

$= \boxed{w^2\sigma_1^2 + (1-w)^2\sigma_2^2}$

c) $\mathbb{V}(\bar{y}_w)$ minimized $\underset{=der}{}$ $w_i \propto 1/\sigma_i$

1. $\mathbb{V}(\bar{y}_w) = \underline{w^2}\sigma_1^2 + \underline{(1-w)^2}\sigma_2^2$

$\frac{d}{dw}\mathbb{V}(\bar{y}_w) = 2w\sigma_1^2 - 2(1-w)\sigma_2^2 = 0$

$2w\sigma_1^2 - 2\sigma_2^2 + 2w\sigma_2^2 = 0$

$\cancel{2}w\sigma_1^2 + \cancel{2}w\sigma_2^2 = \cancel{2}\sigma_2^2$

$w(\sigma_1^2 + \sigma_2^2) = \sigma_2^2$

$$\boxed{w_1 = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} \rightarrow \underset{w_2}{1-w} = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} \quad \frac{1}{\sigma_i^2} \text{ proportional to}}$$

3a) uncentered model: $y_1 = \beta_0 + \beta_1x_i + \beta_2x_i^2 + e_i \qquad \tilde{x}_i = x_i - \bar{x}$

centered: $y_i = \gamma_0 + \gamma_1\tilde{x}_i + \gamma_2\tilde{x}_i^2 + e_i =$

① distribute γ
② β_2 = coef x^2 β_1 = cof x $\beta_0 \neq x$

$y_i = \gamma_0 + \gamma_1(x_i - \bar{x}) + \gamma_2(x_i - \bar{x})^2 + e_i$

$y_i = \gamma_0 + \gamma_1(x_i - \bar{x}) + \gamma_2(x_i^2 - 2x_i\bar{x} + \bar{x}^2) + e_i$

$y_i = \underline{\gamma_0} + \underline{\gamma_1x_i} - \underline{\gamma_1\bar{x}} + \underline{\gamma_2x_i^2} - \underline{2\gamma_2x_i\bar{x}} + \underline{\gamma_2\bar{x}^2} + e_i$

$y_i = \underline{\gamma_2x_i^2} + \underline{(\gamma_1 - 2\gamma_2\bar{x})}x_1 + \underline{(\gamma_0 - \gamma_1\bar{x} + \gamma_2\bar{x}^2)} + e_i$

$y_1 = \underline{\beta_0} + \underline{\beta_1}x_i + \underline{\beta_2x_i^2} + e_i$

$$\boxed{\beta_2 = \gamma_2 \quad \beta_1 = \gamma_1 - 2\gamma_2\bar{x} \quad \beta_0 = \gamma_0 - \gamma_1\bar{x} + \gamma_2\bar{x}^2}$$

Question 3 parts (b) through (h)

In [1]:
```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import numpy as np
import re
```

In [2]:
```
# Load the CSV file into a DataFrame
import pandas as pd

df = pd.read_csv('auto.txt', sep='\t')

pd.set_option('display.max_rows', None)  # Display all rows
pd.set_option('display.max_columns', None)  # Display all columns

df.head()
```

Out[2]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
0	18.0 8 307.0 130.0 3504. 12...	chevrolet chevelle malibu	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	15.0 8 350.0 165.0 3693. 11...	buick skylark 320	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	18.0 8 318.0 150.0 3436. 11...	plymouth satellite	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	16.0 8 304.0 150.0 3433. 12...	amc rebel sst	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	17.0 8 302.0 140.0 3449. 10...	ford torino	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Reading the auto.txt file with relevant data cleaning steps, realised through hit-and-trial.

In [3]:

```
file_path = 'auto.txt'

with open(file_path, 'r') as file:
    lines = file.readlines()

list_of_row_data = list()
columns_comma_separated = re.sub(r'\s+', ',', lines[0])
columns_list = columns_comma_separated.split(",")[:-1]

for line in lines[1:]:

    metrics = line.split("\t")[0]
    metrics_comma_separated_string = re.sub(r'\s+', ',', metrics)
    metrics_list = [float(value) if value!= "NA" else np.nan \
                    for value in metrics_comma_separated_string.split(",")]
    last_col = line.split("\t")[1]
```

```
    # print(metrics_list)
    # print(metrics)
    # print(last_col)

    metrics_list.append(last_col)
    list_of_row_data.append(metrics_list)


df = pd.DataFrame(list_of_row_data, columns=columns_list)
```

In [4]:

```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
df.head()
```

Out[4]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	nam
0	18.0	8.0	307.0	130.0	3504.0	12.0	70.0	1.0	"chevrol chevel malibu"\
1	15.0	8.0	350.0	165.0	3693.0	11.5	70.0	1.0	"buic skylar 320"\
2	18.0	8.0	318.0	150.0	3436.0	11.0	70.0	1.0	"plymout satellite"\
3	16.0	8.0	304.0	150.0	3433.0	12.0	70.0	1.0	"amc reb sst"\
4	17.0	8.0	302.0	140.0	3449.0	10.5	70.0	1.0	"for torino"\

In [ ]:

In [ ]:

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
mpg             0
cylinders       0
displacement    0
horsepower      5
weight          0
acceleration    0
year            0
origin          0
name            0
dtype: int64
```

In [ ]:

In [6]:
```
# Create the derived variable year-squared
df['year_sq'] = df['year']**2
```

In [7]:
```
df[ [ "year", "year_sq"  ] ].head()
```

Out[7]:

	year	year_sq
0	70.0	4900.0
1	70.0	4900.0
2	70.0	4900.0
3	70.0	4900.0
4	70.0	4900.0

Creating the uncentered regression model

In [8]:
```
X_0 = df[['year' , 'year_sq']]
y_0 = df['mpg']

# Add a constant, initialized to 1, to the predictor variables.
# This will be the intercept.
X_0 = sm.add_constant(X_0)
```

In [9]:
```
# Fit the linear regression model
model = sm.OLS(y_0, X_0).fit()

# Print the summary table of the regression model
print(model.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.369
Model:                            OLS   Adj. R-squared:                  0.366
Method:                 Least Squares   F-statistic:                     115.4
Date:                Tue, 08 Oct 2024   Prob (F-statistic):           3.61e-40
Time:                        10:30:58   Log-Likelihood:                -1288.1
No. Observations:                 397   AIC:                             2582.
Df Residuals:                     394   BIC:                             2594.
Df Model:                           2                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        577.2523    146.671      3.936      0.000     288.896     865.609
year         -15.8409      3.865     -4.098      0.000     -23.440      -8.242
year_sq        0.1123      0.025      4.419      0.000       0.062       0.162
==============================================================================
Omnibus:                       21.346   Durbin-Watson:                   0.809
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.130
Skew:                           0.446   Prob(JB):                     0.000116
Kurtosis:                       2.450   Cond. No.                     2.73e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 2.73e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:

In [10]:
```
correlation_coefficient = df['year'].corr(df['year_sq'])
correlation_coefficient
```

Out[10]:
```
0.999759011614808
```

In [11]:
```
year_mean = np.mean(df['year'])
year_mean
```

Out[11]:
```
75.99496221662469
```

Creating centered year

In [12]:
```
df['year_centered'] = df['year'] - year_mean
df['year_centered'].head()
```

Out[12]:
```
0   -5.994962
1   -5.994962
2   -5.994962
3   -5.994962
4   -5.994962
Name: year_centered, dtype: float64
```

In [ ]:

In [13]:
```
# Creating derived variable for centered year squared
df['year_centered_sq'] = df['year_centered']**2
df[ ['year_centered_sq', 'year_centered'] ].head()
```

Out[13]:

	year_centered_sq	year_centered
0	35.939572	-5.994962
1	35.939572	-5.994962
2	35.939572	-5.994962
3	35.939572	-5.994962
4	35.939572	-5.994962

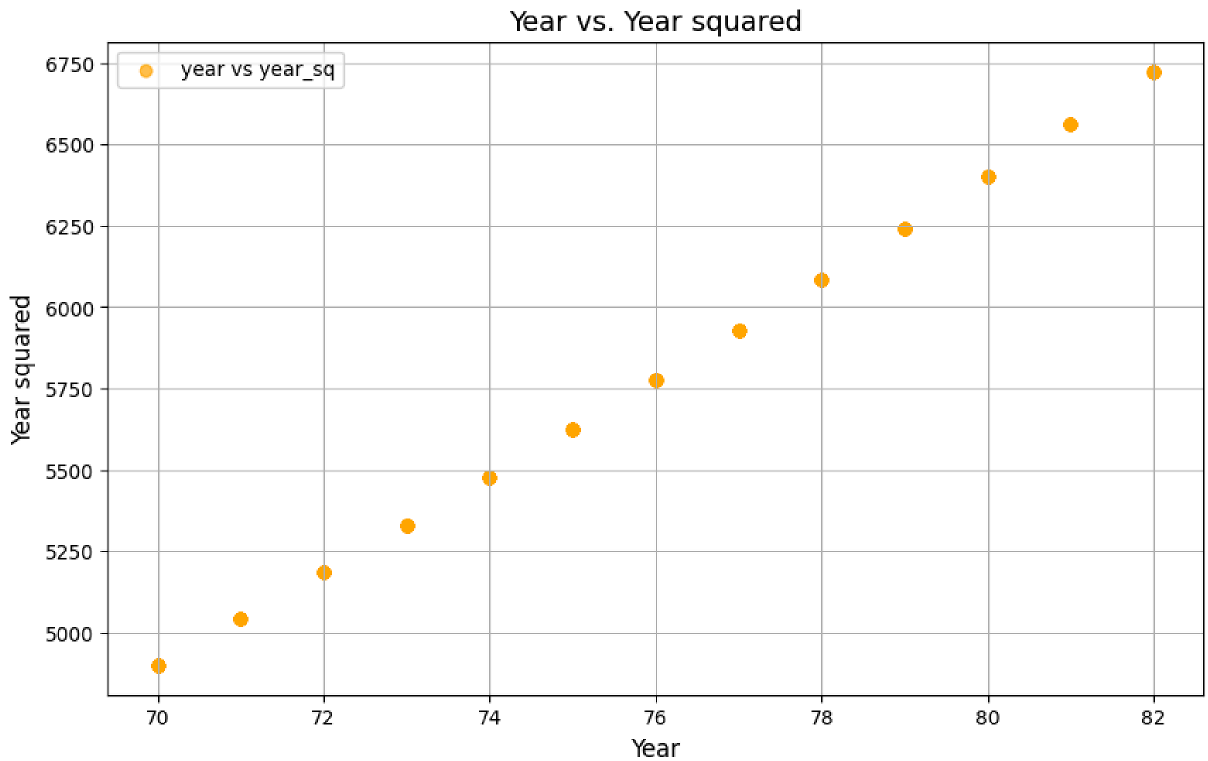
Calculating correlation

In [14]:
```
correlation_coefficient = df['year_centered'].corr(df['year_centered_sq'])
correlation_coefficient
```

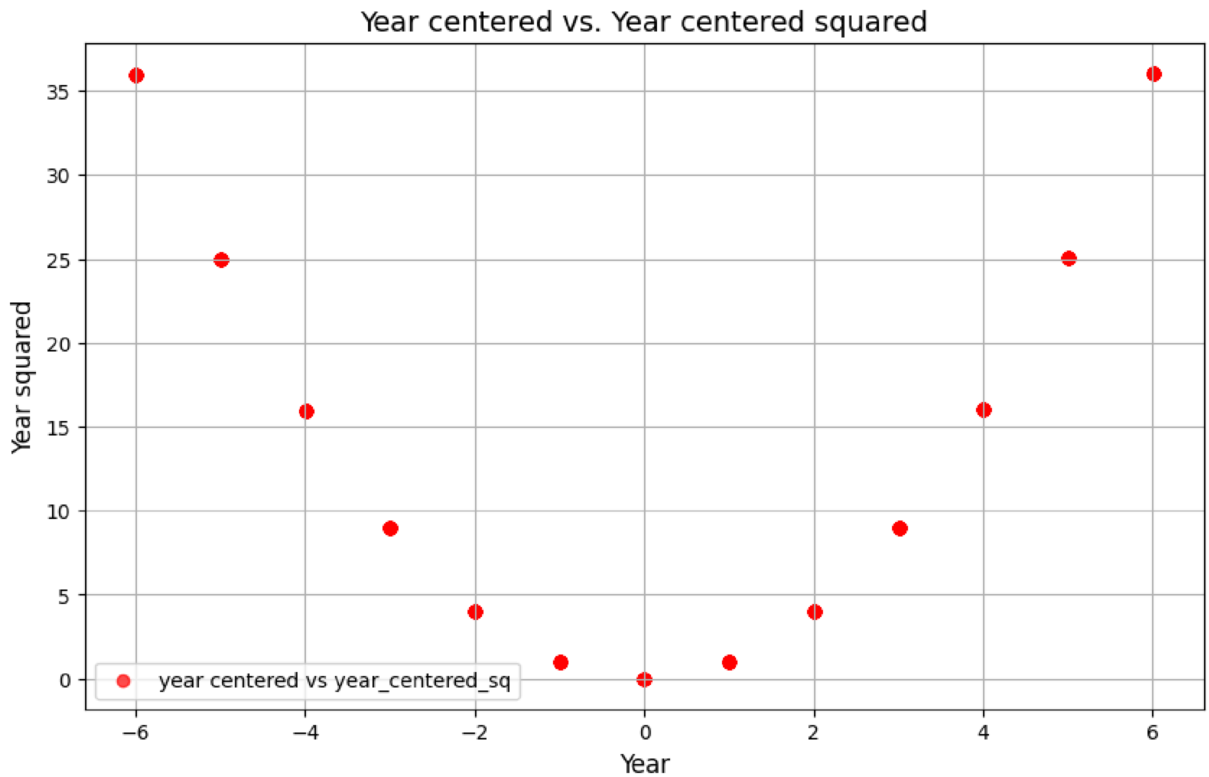
Out[14]:
```
0.014413995959565213
```

Creating required scatter plots

In [15]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['year'], df['year_sq'], color='orange',
            label='year vs year_sq', alpha=0.7)
# plt.scatter(df['year_centere'], df['year_centered_sq'], color='orange', label='Ap
plt.title('Year vs. Year squared', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Year squared', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```



In [16]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['year_centered'], df['year_centered_sq'],
            color='red', label='year centered vs year_centered_sq',
            alpha=0.7)
# plt.scatter(df['year_centere'], df['year_centered_sq'], color='orange', label='Ap
plt.title('Year centered vs. Year centered squared', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Year squared', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```



Creating the centered model

In [17]:
```
X_1 = df[['year_centered' , 'year_centered_sq']]
y_1 = df['mpg']

# Add a constant, initialized to 1, to the predictor variables.
# This will be the intercept.
X_1 = sm.add_constant(X_1)
```

In [ ]:
```
```

In [18]:
```
# Fit the linear regression model
model = sm.OLS(y_1, X_1).fit()

# Print the summary table of the regression model
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.369
Model:                            OLS   Adj. R-squared:                  0.366
Method:                 Least Squares   F-statistic:                     115.4
Date:                Tue, 08 Oct 2024   Prob (F-statistic):           3.61e-40
Time:                        10:30:59   Log-Likelihood:                -1288.1
No. Observations:                 397   AIC:                             2582.
Df Residuals:                     394   BIC:                             2594.
Df Model:                           2
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               21.9906      0.466     47.214      0.000      21.075      22.906
year_centered        1.2278      0.085     14.469      0.000       1.061       1.395
year_centered_sq     0.1123      0.025      4.419      0.000       0.062       0.162
==============================================================================
Omnibus:                       21.346   Durbin-Watson:                   0.809
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.130
Skew:                           0.446   Prob(JB):                     0.000116
Kurtosis:                       2.450   Cond. No.                         27.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

In [ ]:

As can be observed ->

beta2 = gamma2 i.e. beta2 = 0.1123

evaluating ->

- **beta1 = gamma1 - 2 * gamma2 * year_mean**

----

gamma2 = 0.1123

gamma1 = 1.2278

year_mean = 75.99

In [19]:

```
gamma1 = 1.2278
gamma2 = 0.1123
year_mean = 75.99

# b2= gamma2
# b1 = gamma1 - 2*gamma2

beta2 = gamma1 - 2*gamma2*year_mean
print(beta2, round(beta2,2))
```

```
-15.839553999999998 -15.84
```

As can be seen, the values derived for beta1 and beta2 in terms of gamma1 and gamma2 in the first part of the problem have been verified by the Betas and Gammas produced by the OLS models for un-centered and centered model.

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

Hw2_Q4

Hongkai Lou

2024-10-04

```r
part <- read.csv('part.csv')
head(part)
```

```
##   tx wc          x  y
## 1  1  4   6.854164  3
## 2  1 32  29.893616  2
## 3  0  0 108.425476  0
## 4  1 27  63.583313  0
## 5  0  0  54.541656 84
## 6  0  0  23.914886  5
```

```r
dim(part)
```

```
## [1] 14178     4
```

```r
summary(part)
```

```
##        tx             wc                x                  y
##  Min.   :0.0   Min.   :  0.000   Min.   :  0.0233   Min.   :   0.0
##  1st Qu.:0.0   1st Qu.:  0.000   1st Qu.:  7.5445   1st Qu.:   0.0
##  Median :0.5   Median :  0.500   Median : 20.3936   Median :   6.0
##  Mean   :0.5   Mean   :  8.626   Mean   : 36.2646   Mean   :  55.1
##  3rd Qu.:1.0   3rd Qu.: 13.000   3rd Qu.: 46.7487   3rd Qu.:  45.0
##  Max.   :1.0   Max.   :189.000   Max.   :489.1250   Max.   :8188.0
```

(a) and (b)

```r
lm1 <- lm(log(y+1)~log(x+1)+tx, data = part )
lm2 <- lm(log(y+1)~log(x+1)+tx+log(wc+1), data = part)
summary(lm1)
```

```
##
## Call:
## lm(formula = log(y + 1) ~ log(x + 1) + tx, data = part)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6845 -1.2918 -0.0937  1.3063  6.1629
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.31705    0.04155  -7.631 2.47e-14 ***
## log(x + 1)   0.80318    0.01205  66.657  < 2e-16 ***
## tx           0.24438    0.02845   8.591  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.693 on 14175 degrees of freedom
## Multiple R-squared:  0.2406, Adjusted R-squared:  0.2405
## F-statistic:  2246 on 2 and 14175 DF,  p-value: < 2.2e-16
```

```r
summary(lm2)
```

```
##
## Call:
## lm(formula = log(y + 1) ~ log(x + 1) + tx + log(wc + 1), data = part)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6819 -1.2885 -0.0959  1.2999  6.1015
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.30823    0.04166  -7.398 1.46e-13 ***
## log(x + 1)   0.80026    0.01209  66.168  < 2e-16 ***
## tx           0.05039    0.07657   0.658  0.51053
## log(wc + 1)  0.07382    0.02706   2.729  0.00637 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.693 on 14174 degrees of freedom
## Multiple R-squared:  0.241,  Adjusted R-squared:  0.2409
## F-statistic:  1500 on 3 and 14174 DF,  p-value: < 2.2e-16
```

(c) Our Null Hypothesis is $B_2 = 0$, where participation does not have a significant effect on spending

Our Alternative Hypothesis is $B_2 \neq 0$, where participation has a significant effect on spending

Based on Model 1, the t-value of tx (participation) is 8.591, with the corresponding p-value less than 2e-16. our p-value < 2e-16. So that means participation has a significant effect on future spending. The estimate is 0.244, meaning that on average, if the customer participated, the amount spent by each customer in the week following the contest increase by 0.244.

(d) Approximately 1.28 greater.

```r
exp(0.24438)
```

```
## [1] 1.276829
```

(e)

```r
summary(lm2)
```

```
##
## Call:
## lm(formula = log(y + 1) ~ log(x + 1) + tx + log(wc + 1), data = part)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6819 -1.2885 -0.0959  1.2999  6.1015
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.30823    0.04166  -7.398 1.46e-13 ***
```

```
## log(x + 1)   0.80026    0.01209  66.168  < 2e-16 ***
## tx           0.05039    0.07657   0.658  0.51053
## log(wc + 1)  0.07382    0.02706   2.729  0.00637 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.693 on 14174 degrees of freedom
## Multiple R-squared:  0.241,  Adjusted R-squared:  0.2409
## F-statistic:  1500 on 3 and 14174 DF,  p-value: < 2.2e-16
```

We can imagine the value of tx will be highly impacted in model 2 because tx itself is an indicator that tx = (wc>0). When we include log(wc+1) in the model, it will have similar or more dramatic impact on the model than its indicator variable. log(wc+1) will be equal to 0 when wc = 0, and thus it can include more information and explain more variance, thus the effect of tx variable will be reduced.

(f) We see participation(tx) loses significance. Unlike model 1, participation has significant effect, once cognitive elaboration is included, effect of participation diminishes, with large p-value and will not pass the hypothesis test ($0.51 > 0.05$) at the 5% level. Whereas word count, with a coefficient of 0.07382, is more significant, affects the spending of customer in post-contest period, and will pass the hypothesis test at 5% level ($0.00637 < 0.05$).

(g): This suggests customers who put more cognitive effort into the entries tend to spend more in post-contest period. If the customer participate, and has high word count, the customer will have high cognitive elaboration, meaning that they are motivated and willing to spend more in the future.

(h) The results suggest that when designing future social media contests, the company should focus on encouraging deeper cognitive engagement rather than simple participation or writing just one word. Word count, a measure of engagement, was a strong predictor of future spending, while participation alone was not significant. Higher word count meaning more cognitive elaboration. Therefore, contests should require more thoughtful or detailed submissions, such as essays or creative content, to drive post-contest spending. Additionally, the company should target loyal customers, as their pre-contest spending is a good indicator of future behavior. Incentivizing higher levels of engagement through skill-based or interactive contests could further enhance future spending.