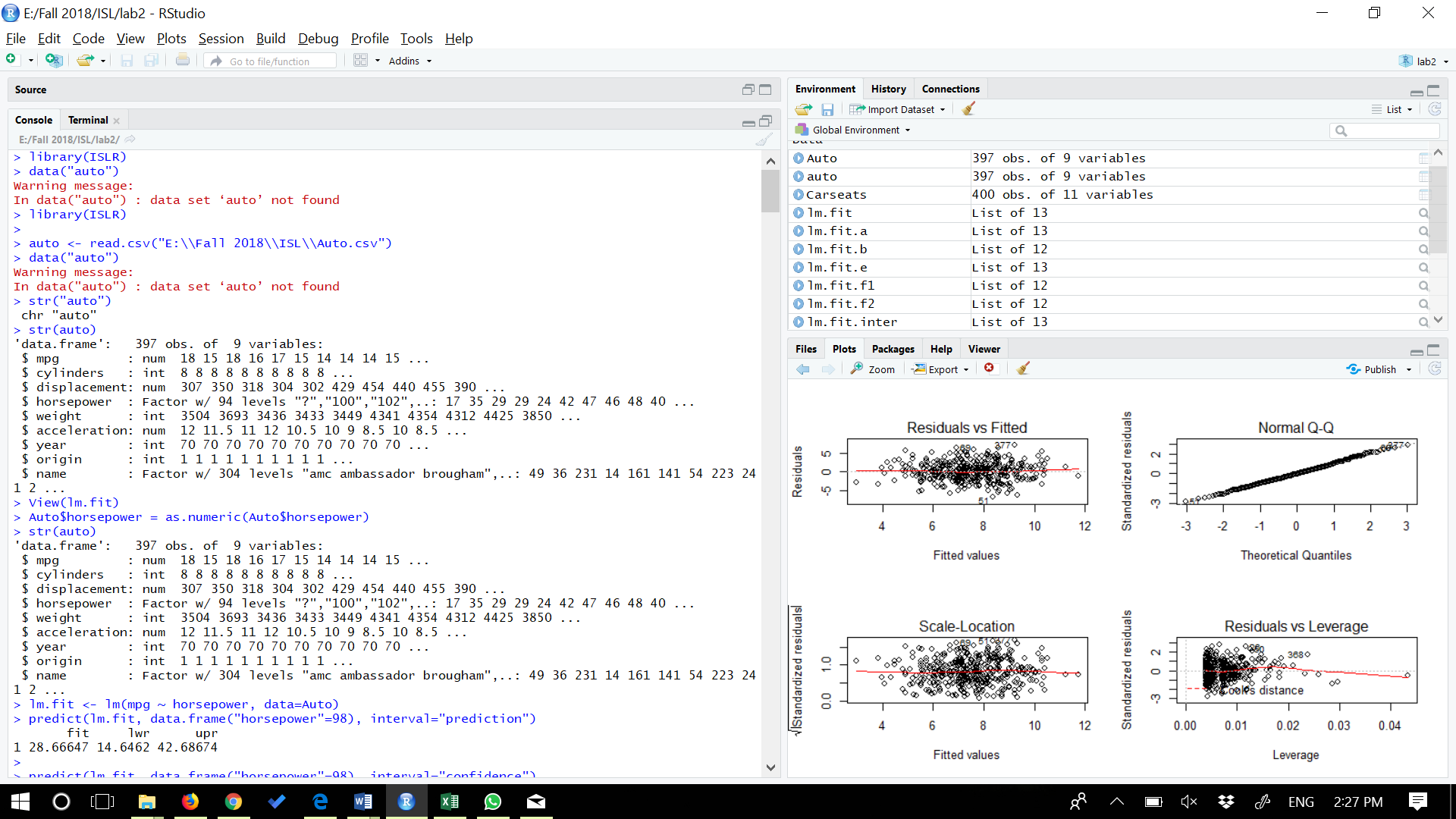
1. **Installed R**



**2.**

**#2(a)**

**#Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output**

> library(ISLR)

> data("auto")

Warning message:

In data("auto") : data set ‘auto’ not found

> library(ISLR)

>

> auto <- read.csv("E:\\Fall 2018\\ISL\\Auto.csv")

> data("auto")

Warning message:

In data("auto") : data set ‘auto’ not found

> str("auto")

chr "auto"

> str(auto)

'data.frame': 397 obs. of 9 variables:

$ mpg : num 18 15 18 16 17 15 14 14 14 15 ...

$ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...

$ displacement: num 307 350 318 304 302 429 454 440 455 390 ...

$ horsepower : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...

$ weight : int 3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...

$ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...

$ year : int 70 70 70 70 70 70 70 70 70 70 ...

$ origin : int 1 1 1 1 1 1 1 1 1 1 ...

$ name : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2 ...

> View(lm.fit)

> Auto$horsepower = as.numeric(Auto$horsepower)

> str(auto)

'data.frame': 397 obs. of 9 variables:

$ mpg : num 18 15 18 16 17 15 14 14 14 15 ...

$ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...

$ displacement: num 307 350 318 304 302 429 454 440 455 390 ...

$ horsepower : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...

$ weight : int 3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...

$ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...

$ year : int 70 70 70 70 70 70 70 70 70 70 ...

$ origin : int 1 1 1 1 1 1 1 1 1 1 ...

$ name : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2 ...

> lm.fit <- lm(mpg ~ horsepower, data=Auto)

**#i. Is there a relationship between the predictor and the response?**

Yes, the coefficient p-value has a very low value.

**#ii. How strong is the relationship between the predictor and the response?**

Good evidence of relationship, R2R2 presents a value of approximately 0.6, that’s 60% of the response variance explained by the simple model.

**#iii. Is the relationship between the predictor and the response positive or negative?**

Negative, since the coefficient has a negative value.

#iv. **What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?**

> predict(lm.fit, data.frame("horsepower"=98), interval="prediction")

fit lwr upr

1 28.66647 14.6462 42.68674

>

> predict(lm.fit, data.frame("horsepower"=98), interval="confidence")

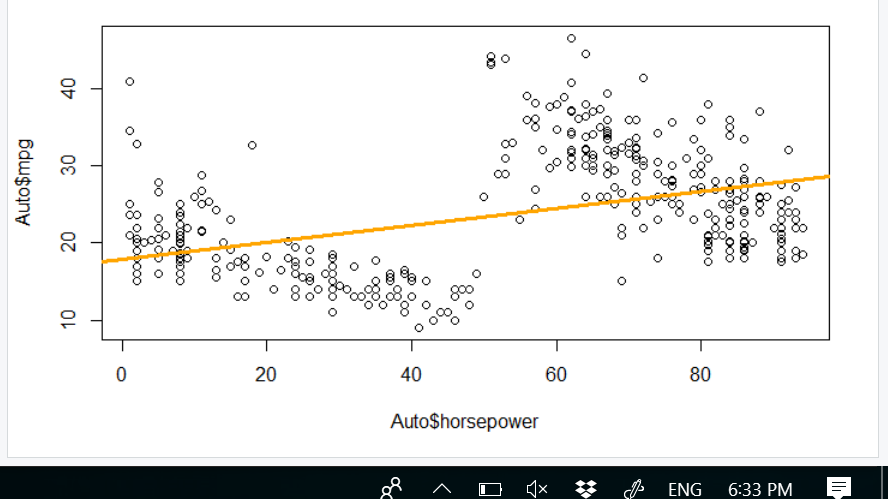
fit lwr upr

1 28.66647 27.36905 29.96389

**#2(b)**  **Plot the response and the predictor. Use the abline() function to display the least squares regression line.**

> plot(Auto$horsepower, Auto$mpg)

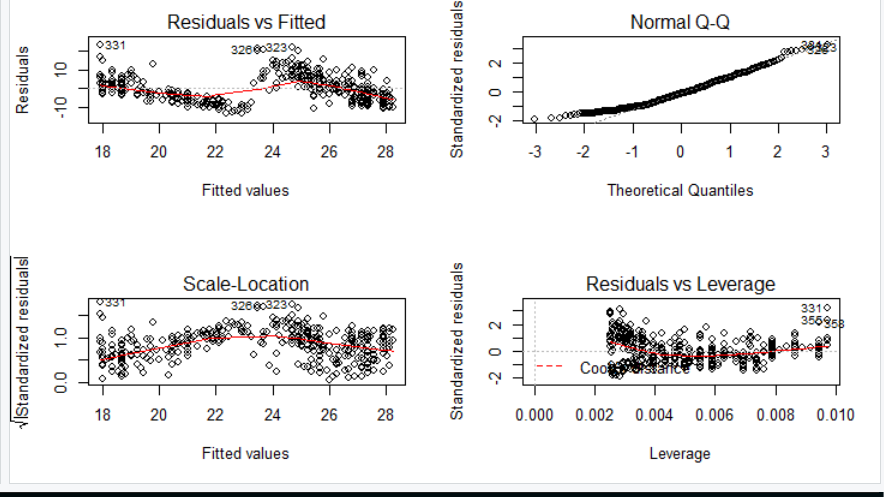
> abline(lm.fit, lwd=3, col="orange")



**#2(c) Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.**

> par(mfrow=c(2,2))

> plot(lm.fit)



> #the common problem are:

> #The Residuals vs Fitted graph appears to have a soft U-shape tendency, and as shown in the plot figure of b, the relationship between predictors and response is not so linear.

> #Analyzing the Residuals vs Fitted graph, it does NOT shows a great heteroscedasticity, which the magnitude of the residuals does not tend to increase with the fitted values.

> #Seen the Scale-Location graph, it is pointed some possible outliers, but checking the picture they don’t seem real outliers. I will get the ISLR reference and use the studentized residuals and observe which ones are greater than 3.

> which(rstudent(lm.fit)>3)

323 331

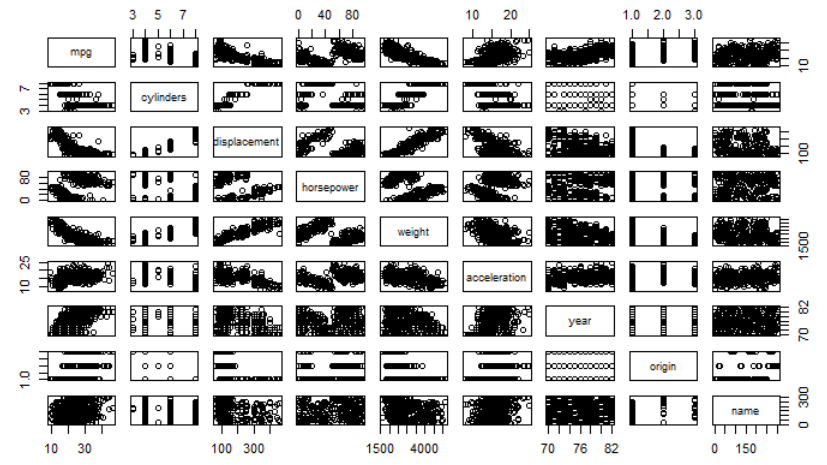
323 331

**3.**

**#3(a)** **Produce a scatterplot matrix which includes all of the variables in the data set.**

> #3. Use of multiple regression

> pairs(auto)



**> #3(b) Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable which is qualitative.**

> names(auto)

[1] "mpg" "cylinders" "displacement" "horsepower" "weight" "acceleration"

[7] "year" "origin" "name"

> cor(auto[1:8])

Error in cor(auto[1:8]) : 'x' must be numeric

> cor(Auto[, !(names(Auto)=="name")])

mpg cylinders displacement horsepower weight acceleration year

mpg 1.0000000 -0.7762599 -0.8044430 0.4228227 -0.8317389 0.4222974 0.5814695

cylinders -0.7762599 1.0000000 0.9509199 -0.5466585 0.8970169 -0.5040606 -0.3467172

displacement -0.8044430 0.9509199 1.0000000 -0.4820705 0.9331044 -0.5441618 -0.3698041

horsepower 0.4228227 -0.5466585 -0.4820705 1.0000000 -0.4821507 0.2662877 0.1274167

weight -0.8317389 0.8970169 0.9331044 -0.4821507 1.0000000 -0.4195023 -0.3079004

acceleration 0.4222974 -0.5040606 -0.5441618 0.2662877 -0.4195023 1.0000000 0.2829009

year 0.5814695 -0.3467172 -0.3698041 0.1274167 -0.3079004 0.2829009 1.0000000

origin 0.5636979 -0.5649716 -0.6106643 0.2973734 -0.5812652 0.2100836 0.1843141

origin

mpg 0.5636979

cylinders -0.5649716

displacement -0.6106643

horsepower 0.2973734

weight -0.5812652

acceleration 0.2100836

year 0.1843141

origin 1.0000000

> cor(auto[, !(names(auto)=="name")])

Error in cor(auto[, !(names(auto) == "name")]) : 'x' must be numeric

> cor(Auto[1:8])

mpg cylinders displacement horsepower weight acceleration year

mpg 1.0000000 -0.7762599 -0.8044430 0.4228227 -0.8317389 0.4222974 0.5814695

cylinders -0.7762599 1.0000000 0.9509199 -0.5466585 0.8970169 -0.5040606 -0.3467172

displacement -0.8044430 0.9509199 1.0000000 -0.4820705 0.9331044 -0.5441618 -0.3698041

horsepower 0.4228227 -0.5466585 -0.4820705 1.0000000 -0.4821507 0.2662877 0.1274167

weight -0.8317389 0.8970169 0.9331044 -0.4821507 1.0000000 -0.4195023 -0.3079004

acceleration 0.4222974 -0.5040606 -0.5441618 0.2662877 -0.4195023 1.0000000 0.2829009

year 0.5814695 -0.3467172 -0.3698041 0.1274167 -0.3079004 0.2829009 1.0000000

origin 0.5636979 -0.5649716 -0.6106643 0.2973734 -0.5812652 0.2100836 0.1843141

origin

mpg 0.5636979

cylinders -0.5649716

displacement -0.6106643

horsepower 0.2973734

weight -0.5812652

acceleration 0.2100836

year 0.1843141

origin 1.0000000

**> #3(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output. For instance:**

> lm.fit <- lm(mpg ~ .-name, data=Auto)

> summary(lm.fit)

Call:

lm(formula = mpg ~ . - name, data = Auto)

Residuals:

Min 1Q Median 3Q Max

-9.629 -2.034 -0.046 1.801 13.010

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -2.128e+01 4.259e+00 -4.998 8.78e-07 \*\*\*

cylinders -2.927e-01 3.382e-01 -0.865 0.3874

displacement 1.603e-02 7.284e-03 2.201 0.0283 \*

horsepower 7.942e-03 6.809e-03 1.166 0.2442

weight -6.870e-03 5.799e-04 -11.846 < 2e-16 \*\*\*

acceleration 1.539e-01 7.750e-02 1.986 0.0477 \*

year 7.734e-01 4.939e-02 15.661 < 2e-16 \*\*\*

origin 1.346e+00 2.691e-01 5.004 8.52e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.331 on 389 degrees of freedom

Multiple R-squared: 0.822, Adjusted R-squared: 0.8188

F-statistic: 256.7 on 7 and 389 DF, p-value: < 2.2e-16

> **#i.** Is there a relationship between the predictors and the response?

> #The p-value corresponding to the F-statistic is 2.037105910^{-139}, this indicates a clear evidence of a relationship between “mpg” and the other predictors

> **#ii**. Which predictors appear to have a statistically significant relationship to the response ?

> #The origin, the year and the cylinders.

> **#iii**. What does the coefficient for the year variable suggest?

> #It suggest that, for each additional year, more 0.75 miles per galon is possible for each car

>

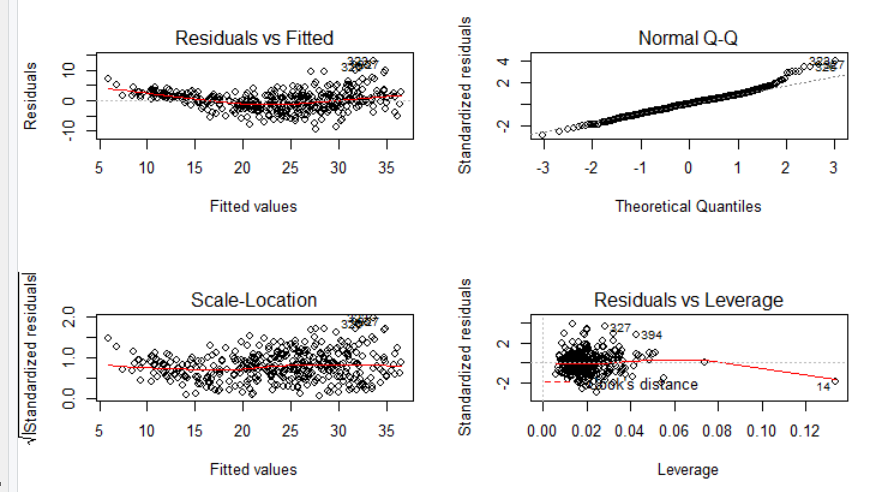
>

**> #3(d) Use the plot() function to produce diagnostic plots of the linear regression ﬁt. Comment on any problems you see with the ﬁt. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?**

> par(mfrow=c(2,2)

+ )

> plot(lm.fit)



> #the plot of residuals versus fitted values indicates the presence of mild non linearity in the data. The plot of standardized residuals versus leverage indicates the presence of a few outliers (higher than 2 or lower than -2) and one high leverage point (point 14).

>

**>**

**> #3(e) Use the ∗ and : symbols to ﬁt linear regression models with interaction eﬀects. Do any interactions appear to be statistically signiﬁcant?**

> lm.fit.inter = lm(mpg ~ (.-name)\*(.-name), data=Auto)

>

> summary(lm.fit.inter)

Call:

lm(formula = mpg ~ (. - name) \* (. - name), data = Auto)

Residuals:

Min 1Q Median 3Q Max

-8.262 -1.554 0.073 1.306 12.360

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 7.563e+01 4.421e+01 1.710 0.088019 .

cylinders 9.271e+00 7.736e+00 1.198 0.231526

displacement -3.392e-01 1.710e-01 -1.983 0.048061 \*

horsepower 2.066e-01 1.433e-01 1.442 0.150225

weight 1.692e-03 1.476e-02 0.115 0.908770

acceleration -8.338e+00 1.655e+00 -5.039 7.37e-07 \*\*\*

year 1.425e-01 5.413e-01 0.263 0.792474

origin -2.024e+01 6.798e+00 -2.978 0.003095 \*\*

cylinders:displacement -4.582e-03 5.500e-03 -0.833 0.405308

cylinders:horsepower 8.186e-03 1.136e-02 0.721 0.471490

cylinders:weight 5.995e-04 7.383e-04 0.812 0.417263

cylinders:acceleration 1.790e-01 1.315e-01 1.361 0.174199

cylinders:year -1.764e-01 1.025e-01 -1.721 0.086094 .

cylinders:origin 1.506e-01 5.935e-01 0.254 0.799826

displacement:horsepower -2.255e-04 2.970e-04 -0.759 0.448209

displacement:weight 1.939e-05 7.875e-06 2.463 0.014235 \*

displacement:acceleration -3.368e-03 2.698e-03 -1.248 0.212707

displacement:year 4.403e-03 2.312e-03 1.905 0.057614 .

displacement:origin 2.733e-02 2.100e-02 1.302 0.193852

horsepower:weight 5.615e-06 2.322e-05 0.242 0.809080

horsepower:acceleration -3.203e-03 3.483e-03 -0.919 0.358479

horsepower:year -2.275e-03 1.785e-03 -1.275 0.203111

horsepower:origin -3.181e-03 1.333e-02 -0.239 0.811472

weight:acceleration 2.419e-04 1.957e-04 1.236 0.217252

weight:year -2.607e-04 1.734e-04 -1.504 0.133495

weight:origin -2.628e-04 1.314e-03 -0.200 0.841549

acceleration:year 9.102e-02 1.955e-02 4.655 4.53e-06 \*\*\*

acceleration:origin 4.806e-01 1.223e-01 3.930 0.000102 \*\*\*

year:origin 1.295e-01 7.043e-02 1.839 0.066721 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.754 on 368 degrees of freedom

Multiple R-squared: 0.8849, Adjusted R-squared: 0.8762

F-statistic: 101.1 on 28 and 368 DF, p-value: < 2.2e-16

> #The model at all had an improvement in R2 from 0.82 to almost 0.89, maybe it can be overfitting, though the interactive term most significant was acceleration:origin with a good coefficient in comparison with the main terms and a small p-value, validating the coefficient.

>

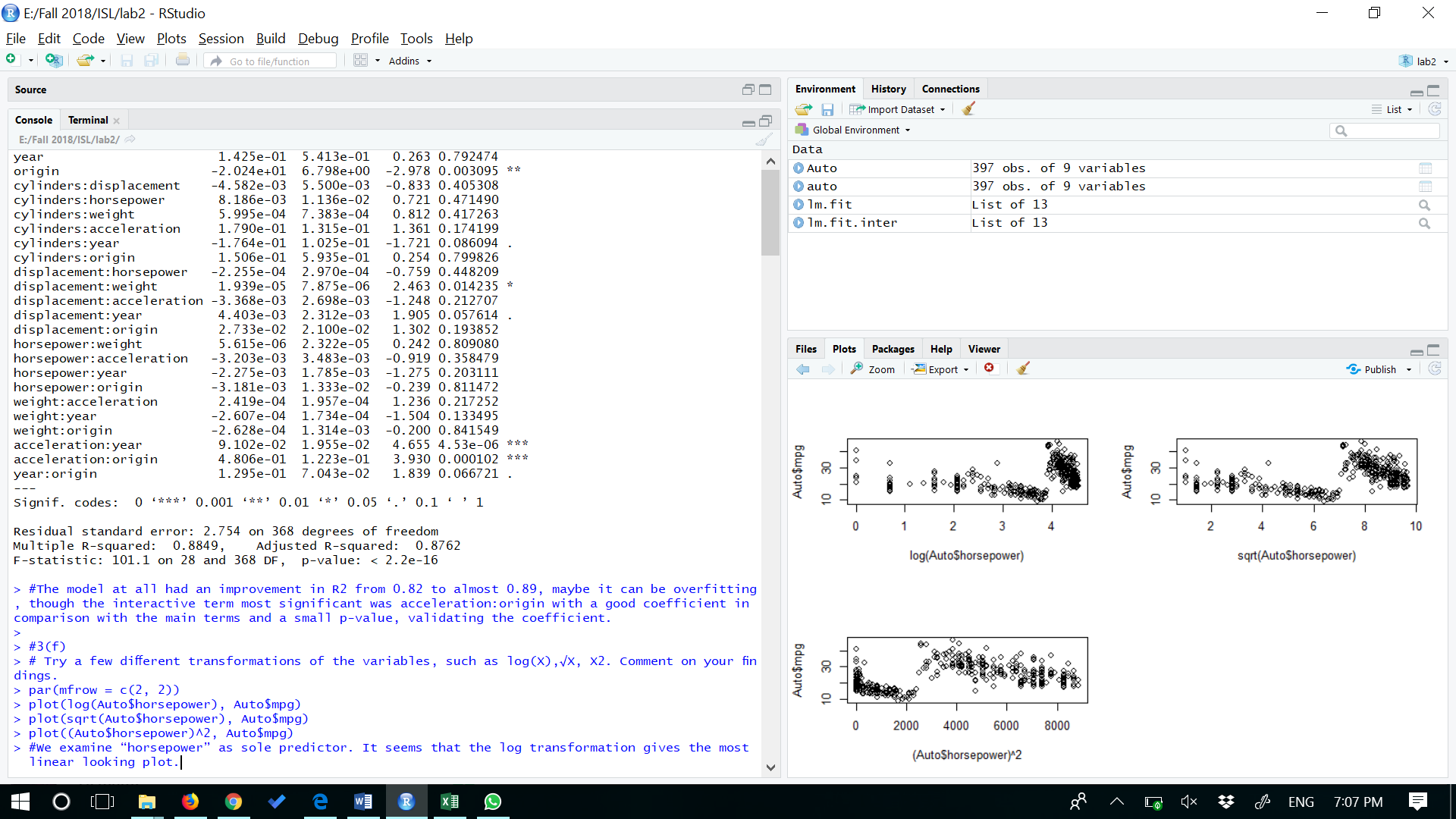
**> #3(f) Try a few diﬀerent transformations of the variables, such as log(X),√X, X2. Comment on your ﬁndings.**

> par(mfrow = c(2, 2))

> plot(log(Auto$horsepower), Auto$mpg)

> plot(sqrt(Auto$horsepower), Auto$mpg)

> plot((Auto$horsepower)^2, Auto$mpg)



> #We examine “horsepower” as sole predictor. It seems that the log transformation gives the most linear looking plot.

**4.**

**> #4(a) Fit a multiple regression model to predict**Sales**using**Price**,**Urban**, and**US**.**

> data("Carseats")

> > lm.fit.a <- lm(Sales ~ Price + Urban + US, data=Carseats)

Error: unexpected '>' in ">"

> > summary(lm.fit.a)

Error: unexpected '>' in ">"

>

> data("Carseats")

> > lm.fit.a <- lm(Sales ~ Price + Urban + US, data=Carseats)

Error: unexpected '>' in ">"

> > summary(lm.fit.a)

Error: unexpected '>' in ">"

>

> data("Carseats")

> lm.fit.a <- lm(Sales ~ Price + Urban + US, data=Carseats)

> summary(lm.fit.a)

Call:

lm(formula = Sales ~ Price + Urban + US, data = Carseats)

Residuals:

Min 1Q Median 3Q Max

-6.9206 -1.6220 -0.0564 1.5786 7.0581

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 13.043469 0.651012 20.036 < 2e-16 \*\*\*

Price -0.054459 0.005242 -10.389 < 2e-16 \*\*\*

UrbanYes -0.021916 0.271650 -0.081 0.936

USYes 1.200573 0.259042 4.635 4.86e-06 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.472 on 396 degrees of freedom

Multiple R-squared: 0.2393, Adjusted R-squared: 0.2335

F-statistic: 41.52 on 3 and 396 DF, p-value: < 2.2e-16

>

**> #4(b) Provide an interpretation of each coefficient in the model. Be careful - some of the variables in the model are qualitative!**

**> #checking qualitative and quantitative variables**

> attach(Carseats)

> str(data.frame(Price, Urban, US))

'data.frame': 400 obs. of 3 variables:

$ Price: num 120 83 80 97 128 72 108 120 124 124 ...

$ Urban: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...

$ US : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...

> #The Urband and US are qualitative

> attach(Carseats)

The following objects are masked from Carseats (pos = 3):

Advertising, Age, CompPrice, Education, Income, Population, Price, Sales,

ShelveLoc, Urban, US

> contrasts(Urban)

Yes

No 0

Yes 1

> contrasts(US)

Yes

No 0

Yes 1

> #By analyzing the coefficients, the Urban has a very high p-value, so it doesn’t prove any evidence of relevance for Sales. The US indicates a strong influence in the model and assigns more 1.2 thousands sales units for each US location. The Price coefficient has a negative relationship with Sales

>

**> #4(c) Write out the model in the equation form, being careful to handle the qualitative variables properly.**

>

> #Sales=13.0434689+(−0.0544588)×Price+(−0.0219162)×Urban+(1.2005727)×US+ε

> If store I sin urban, then urban=1 else 0

Error: unexpected symbol in "If store"

> If stire is in US , then us=1 else 0

Error: unexpected symbol in "If stire"

>

> #If store I sin urban, then urban=1 else 0

> #

> #V

> #If stire is in US , then us=1 else 0

>

>

**> #4(d) For which of the predictors can you reject the null hypothesis H0:βj=0?**

> #We can reject the null hypothesis for price and Us variable

>

>

> **#4(e) On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.**

>

> lm.fit.e <- lm(Sales ~ Price + US, data=Carseats)

> summary(lm.fit.e)

Call:

lm(formula = Sales ~ Price + US, data = Carseats)

Residuals:

Min 1Q Median 3Q Max

-6.9269 -1.6286 -0.0574 1.5766 7.0515

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 13.03079 0.63098 20.652 < 2e-16 \*\*\*

Price -0.05448 0.00523 -10.416 < 2e-16 \*\*\*

USYes 1.19964 0.25846 4.641 4.71e-06 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.469 on 397 degrees of freedom

Multiple R-squared: 0.2393, Adjusted R-squared: 0.2354

F-statistic: 62.43 on 2 and 397 DF, p-value: < 2.2e-16

>

> **#4(f) How well do the models in (a) and (e) fit the data?**

> anova(lm.fit.a, lm.fit.e)

Analysis of Variance Table

Model 1: Sales ~ Price + Urban + US

Model 2: Sales ~ Price + US

Res.Df RSS Df Sum of Sq F Pr(>F)

1 396 2420.8

2 397 2420.9 -1 -0.03979 0.0065 0.9357

> #Both appears identically, and the p-value of F-statistic doesn’t present evidence of differiation.

>

**> #4(g) Using the model from (e), obtain 95% confidence intervals for the coefficients(s).**

> confint(lm.fit.e)

2.5 % 97.5 %

(Intercept) 11.79032020 14.27126531

Price -0.06475984 -0.04419543

USYes 0.69151957 1.70776632

>

**> #4(h) Is there evidence of outliers or high leverage observations in the model from (e)?**

>

>

> par(mfrow=c(2,2))

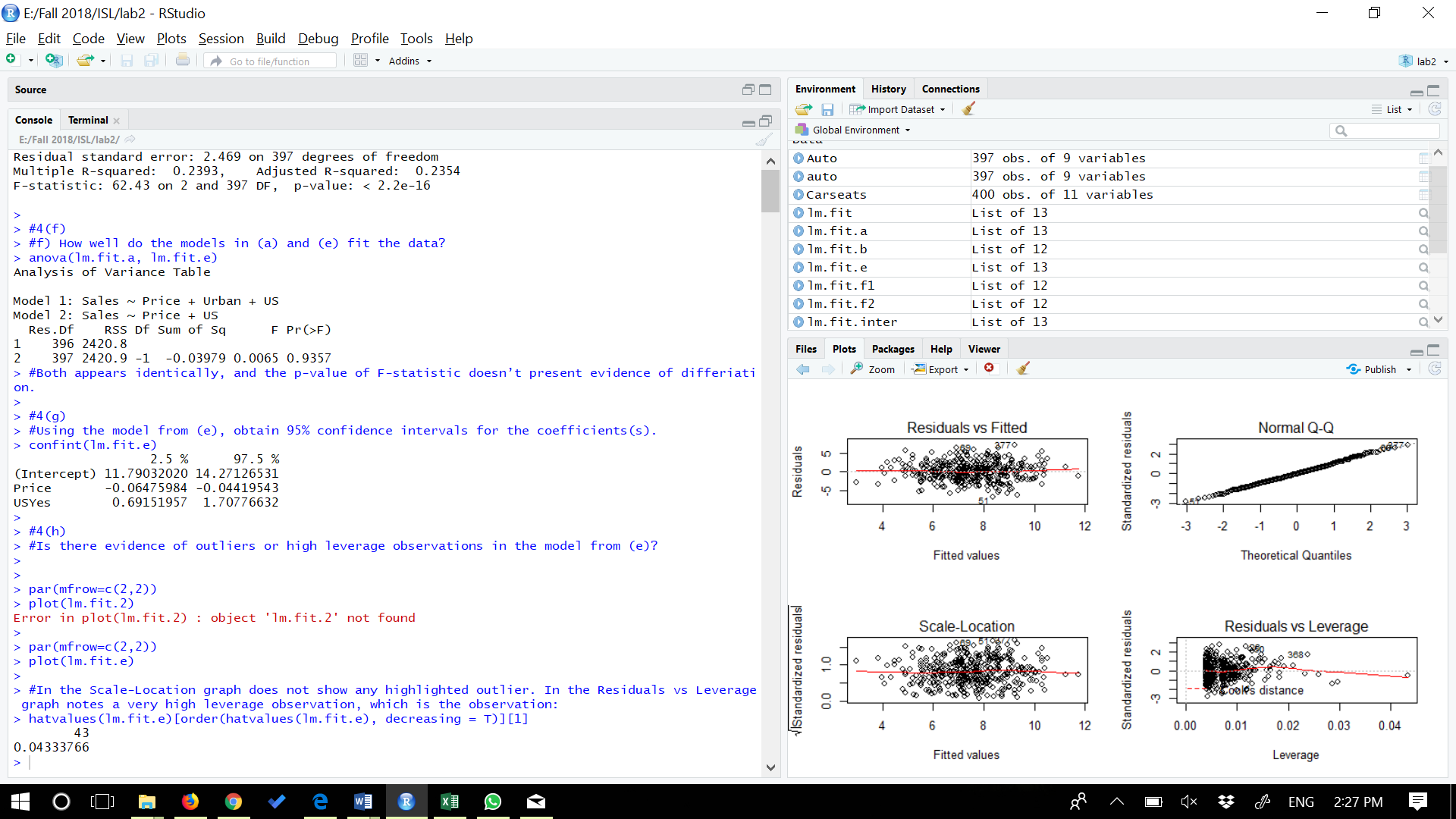
> plot(lm.fit.2)

Error in plot(lm.fit.2) : object 'lm.fit.2' not found

>

> par(mfrow=c(2,2))

> plot(lm.fit.e)



>

> #In the Scale-Location graph does not show any highlighted outlier. In the Residuals vs Leverage graph notes a very high leverage observation, which is the observation:

> hatvalues(lm.fit.e)[order(hatvalues(lm.fit.e), decreasing = T)][1]

43

0.04333766

**5.**

|  |
| --- |
| >  **> #5 In this problem we will investigate the t-statistic for the null hypothesis H0:β=0 in simple linear regression without an intercept. To begin, we generate a predictor x and a response y as follows.**  **> set.seed(1)**  **> x=rnorm(100)**  **> y=2\*x+rnorm(100)**  **>**  **> #5(a) Perform a simple linear regression of y onto x, without an intercept. Report the coeﬃcient estimate ˆ β, the standard error of this coeﬃcient estimate, and the tstatistic and p-value associated with the null hypothesis H0 : β = 0. Comment on these results. (You can perform regression without an intercept using the command lm(y ∼ x + 0).)**  > lm.fit.a <- lm(y ~ x+0)  > summary(lm.fit.a)$coefficients  Estimate Std. Error t value Pr(>|t|)  x 1.993876 0.1064767 18.72593 2.642197e-34  > lm.fit.a <- lm(y ~ x+0)  > summary(lm.fit.a)  Call:  lm(formula = y ~ x + 0)  Residuals:  Min 1Q Median 3Q Max  -1.9154 -0.6472 -0.1771 0.5056 2.3109  Coefficients:  Estimate Std. Error t value Pr(>|t|)  x 1.9939 0.1065 18.73 <2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.9586 on 99 degrees of freedom  Multiple R-squared: 0.7798, Adjusted R-squared: 0.7776  F-statistic: 350.7 on 1 and 99 DF, p-value: < 2.2e-16  > #According to the summary above, we have a value of 1.9939 for β^, a value of 0.1065 for the standard error, a value of 18.73 for the t-statistic and a value of 2.642196910^{-34} for the p-value. The small p-value allows us to reject H0.  >  **> #5(b) Now perform a simple linear regression of x onto y without an intercept, and report the coeﬃcient estimate, its standard error, and the corresponding t-statistic and p-values associated with the null hypothesis H0 : β = 0. Comment on these results.**  > lm.fit.b <- lm(x ~ y+0)  > summary(lm.fit.b)$coefficients  Estimate Std. Error t value Pr(>|t|)  y 0.3911145 0.02088625 18.72593 2.642197e-34  > #According to the summary above, we have a value of 0.3911145 for β^, a value of 0.02088625 for the standard error, a value of 18.72593 for the t-statistic and a value of 2.642196910^{-34} for the p-value. The small p-value allows us to reject H0.  >  > **#5(c) What is the relationship between the results obtained in (a) and (b)?**  > #We obtain the same value for the t-statistic and consequently the same value for the corresponding p-value. Both results in (a) and (b) reflect the same line created in (a)  >  **> #5(d)**      > #lets conform by R  > t\_value = (sqrt(length(x)-1)\*sum(x\*y))/sqrt(sum(y^2)\*sum(x^2) - (sum(x\*y))^2)  > summary(t\_value)  Min. 1st Qu. Median Mean 3rd Qu. Max.  18.73 18.73 18.73 18.73 18.73 18.73  > n <- length(x)  > t <- sqrt(n - 1)\*(x %\*% y)/sqrt(sum(x^2) \* sum(y^2) - (x %\*% y)^2)  > as.numeric(t)  [1] 18.72593  > as.numeric((t\_value))  [1] 18.72593  > #We may see that the t above is exactly the t-statistic given in the summary of “fit.b”.  >  >  > **#5(e) Using the results from (d), argue that the t-statistic for the regression of y onto x is the same as the t-statistic for the regression of x onto y**.  > #if we take only the formula of β^ and SE(β^), the ratio of them will be the same indepedently if we do regression onto x or y.  >  **>**  **> #5(f) In R, show that when regression is performed with an intercept, the t-statistic for H0 : β1 = 0. is the same for the regression of y onto x as it is for the regression of x onto y.**  >  > lm.fit.f1 <- lm(x ~ y)  > summary(lm.fit.f1)$coefficients[2,3]  [1] 18.5556  > #th above is the regression of x onto y  > #regression y onto x  > lm.fit.f2 <- lm(y ~ x)  > summary(lm.fit.f2)$coefficients[2,3]  [1] 18.5556  > # we can see both t values are equal |
|  |
| |  | | --- | |  | |