bacs\_hw8

110071010

2024-04-13

# Question 1

We will use the interactive\_regression() function from CompStatsLib again.

**Question**

Comparing scenarios 1 and 2, which do we expect to have a stronger R^2 ?

**Answer**

Scenario 1

## 1b

**Question**

Comparing scenarios 3 and 4, which do we expect to have a stronger R^2 ?

**Answer**

Scenario 3

## 1c

**Question**

Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

**Answer**

SSE : Scenario 2 has larger SSE than Scenario 1 does

SSR : Scenario 1 has larger SSE than Scenario 2 does

SST : likely to be slightly higher in Scenario 2 due to greater overall variance

## 1d

**Question**

Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

**Answer**

SSE : Scenario 4 has larger SSE than Scenario 3 does

SSR : Scenario 3 has larger SSE than Scenario 4 does

SST : likely to be slightly higher in Scenario 4 due to greater variance

# Question 2

## 2a

Use the lm() function to estimate the regression model Salary ~ Experience + Score + Degree. Show the beta coefficients, R^2, and the first 5 values of y ($fitted.values) and ($residuals)

salary\_data <- read.table("programmer\_salaries.txt", header=TRUE, sep="\t")  
regr <- lm(Salary ~ Experience + Score + Degree, data=salary\_data)  
summary(regr)

##   
## Call:  
## lm(formula = Salary ~ Experience + Score + Degree, data = salary\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.8963 -1.7290 -0.3375 1.9699 5.0480   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.9448 7.3808 1.076 0.2977   
## Experience 1.1476 0.2976 3.856 0.0014 \*\*  
## Score 0.1969 0.0899 2.191 0.0436 \*   
## Degree 2.2804 1.9866 1.148 0.2679   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.396 on 16 degrees of freedom  
## Multiple R-squared: 0.8468, Adjusted R-squared: 0.8181   
## F-statistic: 29.48 on 3 and 16 DF, p-value: 9.417e-07

# Extract the first 5 fitted values from the model  
fitted\_values <- head(fitted(regr), 5)  
print("First 5 Fitted Values:")  
print(fitted\_values)  
  
# Extract the first 5 residuals from the model  
residuals <- head(residuals(regr), 5)  
print("First 5 Residuals:")  
print(residuals)

## [1] "First 5 Fitted Values:"  
## 1 2 3 4 5   
## 27.89626 37.95204 26.02901 32.11201 36.34251   
## [1] "First 5 Residuals:"  
## 1 2 3 4 5   
## -3.8962605 5.0479568 -2.3290112 2.1879860 -0.5425072

## 2b

Use only linear algebra and the geometric view of regression to estimate the regression yourself

### (i)(ii)

# Create an X matrix that has a first column of 1s followed by columns of the independent variables  
X <- cbind(1, salary\_data$Experience, salary\_data$Score, salary\_data$Degree)  
  
# Create a y vector with the Salary values  
y <- salary\_data$Salary

### (iii)

# Compute the beta\_hat vector of estimated regression coefficients   
beta\_hat <- solve(t(X) %\*% X) %\*% t(X) %\*% y  
beta\_hat

## [,1]  
## [1,] 7.944849  
## [2,] 1.147582  
## [3,] 0.196937  
## [4,] 2.280424

### (iv)

# Using the above, compute a y\_hat vector of estimated y values, and a res vector of residuals   
y\_hat <- X %\*% beta\_hat  
res <- y - y\_hat  
  
# Show the code and the first 5 values of y\_hat and res  
print(head(y\_hat, 5))  
print(head(res, 5))

## [,1]  
## [1,] 27.89626  
## [2,] 37.95204  
## [3,] 26.02901  
## [4,] 32.11201  
## [5,] 36.34251  
## [,1]  
## [1,] -3.8962605  
## [2,] 5.0479568  
## [3,] -2.3290112  
## [4,] 2.1879860  
## [5,] -0.5425072

### (v)

# Using only the results from (i) – (iv), compute SSR, SSE and SST  
SSR <- sum((y\_hat - mean(y))^2)  
SSE <- sum((y - y\_hat)^2)  
SST <- sum((y - mean(y))^2)  
  
cat("SSR:", SSR, "\n")  
cat("SSE:", SSE, "\n")  
cat("SST:", SST, "\n")

## SSR: 507.896   
## SSE: 91.88949   
## SST: 599.7855

## 2c

Compute R^2 for in two ways, and confirm you get the same results

### (i)

Use any combination of SSR, SSE, and SST

SSR / SST

## [1] 0.8467961

### (ii)

Use the squared correlation of vectors y and y\_hat

cor(y, y\_hat)^2

## [,1]  
## [1,] 0.8467961

**Observation**

Two methods yield the same result indeed.

# Question 3

We’re going to look back at the early, heady days of global car manufacturing, when American, Japanese, and European cars competed to rule the world. Take a look at the data set in file auto-data.txt. We are interested in explaining what kind of cars have higher fuel efficiency (mpg).

**mpg/** miles-per-gallon (dependent variable)  
**cylinders/** cylinders in engine  
**displacement/** size of engine  
**horsepower/** power of engine  
**weight/** weight of car  
**acceleration/** acceleration ability of car  
**model\_year/** year model was released  
**origin/** place car was designed (1: USA, 2: Europe, 3: Japan)  
**car\_name/** make and model names

auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")  
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",   
 "acceleration", "model\_year", "origin", "car\_name")

## 3a

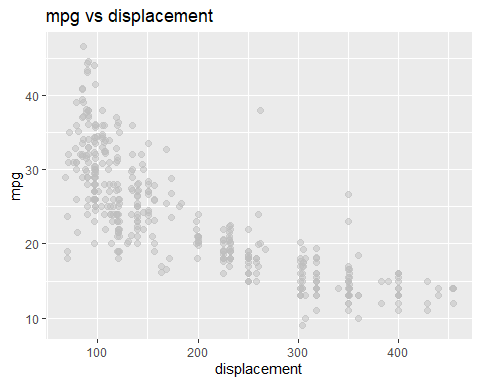
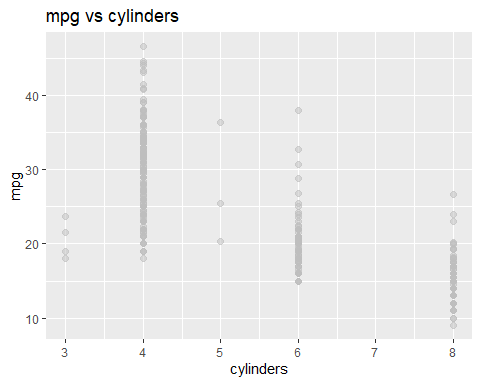
Let’s first try exploring this data and problem.

### (i)

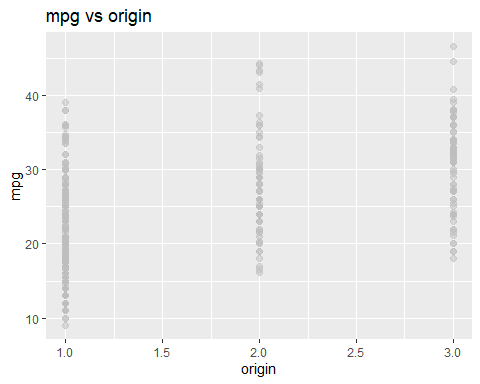
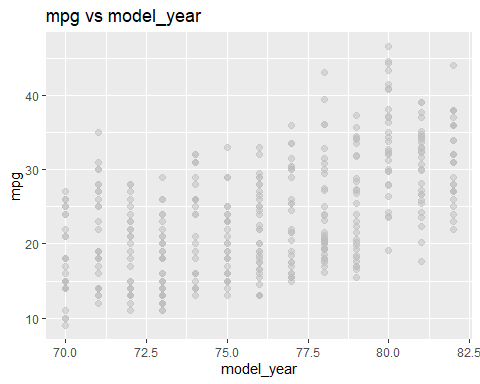
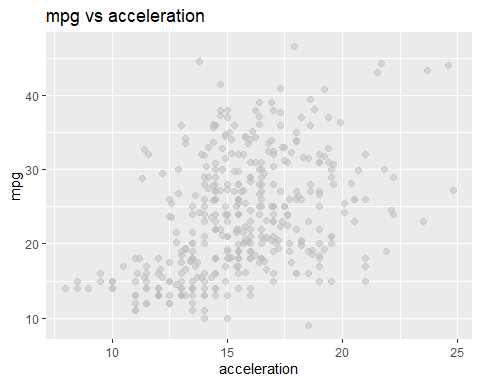
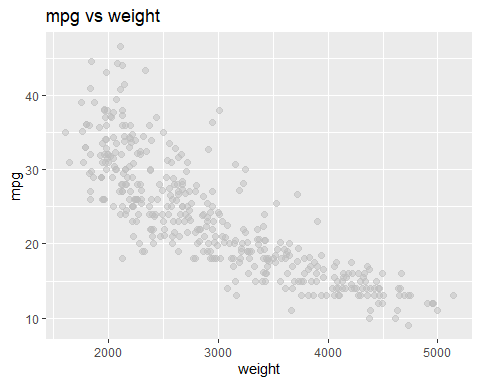
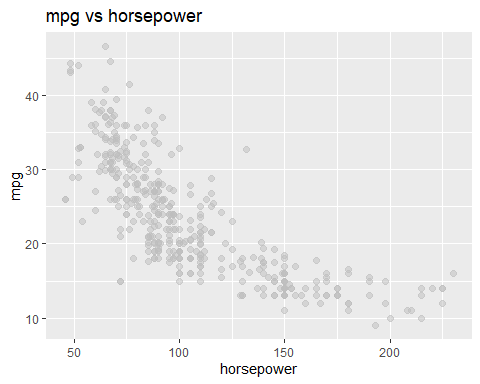
Visualize the data as you wish (report only relevant/interesting plots)

library(ggplot2)  
  
vars <- c("cylinders","displacement", "horsepower", "weight", "acceleration","model\_year","origin")  
for (var in vars) {  
 p <- ggplot(auto, aes\_string(x = var, y = "mpg")) +  
 geom\_point(alpha = 0.5, color = "grey",cex = 2) +  
 labs(title = paste("mpg vs", var), x = var, y = "mpg")  
 print(p)  
}

## Warning: `aes\_string()` was deprecated in ggplot2 3.0.0.  
## ℹ Please use tidy evaluation idioms with `aes()`.  
## ℹ See also `vignette("ggplot2-in-packages")` for more information.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



## Warning: Removed 6 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



### (ii)

Report a correlation table of all variables, rounding to two decimal places

# Compute the correlation matrix  
cor\_matrix <- cor(auto[, sapply(auto, is.numeric)], use = "pairwise.complete.obs")  
round(cor\_matrix, 2)

## mpg cylinders displacement horsepower weight acceleration  
## mpg 1.00 -0.78 -0.80 -0.78 -0.83 0.42  
## cylinders -0.78 1.00 0.95 0.84 0.90 -0.51  
## displacement -0.80 0.95 1.00 0.90 0.93 -0.54  
## horsepower -0.78 0.84 0.90 1.00 0.86 -0.69  
## weight -0.83 0.90 0.93 0.86 1.00 -0.42  
## acceleration 0.42 -0.51 -0.54 -0.69 -0.42 1.00  
## model\_year 0.58 -0.35 -0.37 -0.42 -0.31 0.29  
## origin 0.56 -0.56 -0.61 -0.46 -0.58 0.21  
## model\_year origin  
## mpg 0.58 0.56  
## cylinders -0.35 -0.56  
## displacement -0.37 -0.61  
## horsepower -0.42 -0.46  
## weight -0.31 -0.58  
## acceleration 0.29 0.21  
## model\_year 1.00 0.18  
## origin 0.18 1.00

### (iii)

**Question**

From the visualizations and correlations, which variables appear to relate to mpg?

**Answer**

displacement, horsepower and weight

### (iv)

**Question**

Which relationships might not be linear?

**Answer**

displacement, horsepower, and weight

### (v)

**Question**

Are there any pairs of independent variables that are highly correlated (r > 0.7)?

**Answer**

cylinders, displacement, horsepower, and weight

## 3b

Let’s create a linear regression model where mpg is dependent upon all other suitable variables

### (i)

**Question**

Which independent variables have a ‘significant’ relationship with mpg at 1% significance?

# Fit the linear regression model  
regr <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model\_year + factor(origin), data = auto)  
summary(regr)

##   
## Call:  
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +   
## acceleration + model\_year + factor(origin), data = auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.0095 -2.0785 -0.0982 1.9856 13.3608   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.795e+01 4.677e+00 -3.839 0.000145 \*\*\*  
## cylinders -4.897e-01 3.212e-01 -1.524 0.128215   
## displacement 2.398e-02 7.653e-03 3.133 0.001863 \*\*   
## horsepower -1.818e-02 1.371e-02 -1.326 0.185488   
## weight -6.710e-03 6.551e-04 -10.243 < 2e-16 \*\*\*  
## acceleration 7.910e-02 9.822e-02 0.805 0.421101   
## model\_year 7.770e-01 5.178e-02 15.005 < 2e-16 \*\*\*  
## factor(origin)2 2.630e+00 5.664e-01 4.643 4.72e-06 \*\*\*  
## factor(origin)3 2.853e+00 5.527e-01 5.162 3.93e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.307 on 383 degrees of freedom  
## Multiple R-squared: 0.8242, Adjusted R-squared: 0.8205   
## F-statistic: 224.5 on 8 and 383 DF, p-value: < 2.2e-16

**Answer**

weight, model\_year

### (ii)

**Question**

Looking at the coefficients, is it possible to determine which independent variables are the most effective at increasing mpg? If so, which ones, and if not, why not? (hint: units!)

**Answer**

No. The data are comparable only when they are standardized, or we are simply juggling with different units.

## 3c

Let’s try to resolve some of the issues with our regression model above.

### (i)

**Question**

Create fully standardized regression results: are these slopes easier to compare?

auto\_std <- data.frame(cbind(scale(auto[,1:7]), origin = auto[,8]))  
regr\_std <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model\_year + factor(origin), data = auto\_std)  
summary(regr\_std)

##   
## Call:  
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +   
## acceleration + model\_year + factor(origin), data = auto\_std)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.15270 -0.26593 -0.01257 0.25404 1.70942   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.13323 0.03174 -4.198 3.35e-05 \*\*\*  
## cylinders -0.10658 0.06991 -1.524 0.12821   
## displacement 0.31989 0.10210 3.133 0.00186 \*\*   
## horsepower -0.08955 0.06751 -1.326 0.18549   
## weight -0.72705 0.07098 -10.243 < 2e-16 \*\*\*  
## acceleration 0.02791 0.03465 0.805 0.42110   
## model\_year 0.36760 0.02450 15.005 < 2e-16 \*\*\*  
## factor(origin)2 0.33649 0.07247 4.643 4.72e-06 \*\*\*  
## factor(origin)3 0.36505 0.07072 5.162 3.93e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.423 on 383 degrees of freedom  
## Multiple R-squared: 0.8242, Adjusted R-squared: 0.8205   
## F-statistic: 224.5 on 8 and 383 DF, p-value: < 2.2e-16

**Answer**

Yes. All variables get uniform units, so they become comparable.

### (ii)

**Question**

Regress mpg over each nonsignificant independent variable, individually. Which ones become significant when we regress mpg over them individually?

**Observation**

According to the summary table above, three variables (cylinders, horsepower, and acceleration) are not significant as their p-value are greater than 0.01.

regr\_cylinders <- lm(mpg ~ cylinders, data = auto\_std)  
summary(regr\_cylinders)

##   
## Call:  
## lm(formula = mpg ~ cylinders, data = auto\_std)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.82455 -0.43297 -0.08288 0.32674 2.29046   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.834e-15 3.169e-02 0.00 1   
## cylinders -7.754e-01 3.173e-02 -24.43 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6323 on 396 degrees of freedom  
## Multiple R-squared: 0.6012, Adjusted R-squared: 0.6002   
## F-statistic: 597.1 on 1 and 396 DF, p-value: < 2.2e-16

regr\_horsepower <- lm(mpg ~ horsepower, data = auto\_std)  
summary(regr\_horsepower)

##   
## Call:  
## lm(formula = mpg ~ horsepower, data = auto\_std)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.73632 -0.41699 -0.04395 0.35351 2.16531   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.008784 0.031701 -0.277 0.782   
## horsepower -0.777334 0.031742 -24.489 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6277 on 390 degrees of freedom  
## (因為不存在，6 個觀察量被刪除了)  
## Multiple R-squared: 0.6059, Adjusted R-squared: 0.6049   
## F-statistic: 599.7 on 1 and 390 DF, p-value: < 2.2e-16

regr\_acceleration <- lm(mpg ~ acceleration, data = auto\_std)  
summary(regr\_acceleration)

##   
## Call:  
## lm(formula = mpg ~ acceleration, data = auto\_std)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.3039 -0.7210 -0.1589 0.6087 2.9672   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.004e-16 4.554e-02 0.000 1   
## acceleration 4.203e-01 4.560e-02 9.217 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9085 on 396 degrees of freedom  
## Multiple R-squared: 0.1766, Adjusted R-squared: 0.1746   
## F-statistic: 84.96 on 1 and 396 DF, p-value: < 2.2e-16

### (iii)

**Question**

Plot the distribution of the residuals: are they normally distributed and centered around zero?  
(get the residuals of a fitted linear model, e.g. regr <- lm(…), using regr$residuals

residual1 <- data.frame(residual = regr\_std$residuals)  
residual2 <- data.frame(residual = regr\_cylinders$residuals)  
residual3 <- data.frame(residual = regr\_horsepower$residuals)  
residual4 <- data.frame(residual = regr\_acceleration$residuals)

plot1 <- ggplot(x = residual1)+  
 geom\_histogram(aes(y = after\_stat(density)), color = "grey" , bins = 40)+  
 labs(title = "STD Residuals Histogram", x = "residuals", y = "frequency")+  
 geom\_density(aes(x = residual1), color = "darkgrey", lwd = 1)  
plot2 <- ggplot(x = residual2)+  
 geom\_histogram(aes(y = after\_stat(density)), color = "grey" , bins = 40)+  
 labs(title = "CYLINDERS Residuals Histogram", x = "residuals", y = "frequency")+  
 geom\_density(aes(x = residual2), color = "darkgrey", lwd = 1)  
plot3 <- ggplot(x = residual3)+  
 geom\_histogram(aes(y = after\_stat(density)), color = "grey" , bins = 40)+  
 labs(title = "HORSEPOWER Residuals Histogram", x = "residuals", y = "frequency")+  
 geom\_density(aes(x = residual3), color = "darkgrey", lwd = 1)  
plot4 <- ggplot(x = residual4)+  
 geom\_histogram(aes(y = after\_stat(density)), color = "grey" , bins = 40)+  
 labs(title = "ACCELERATION Residuals Histogram", x = "residuals", y = "frequency")+  
 geom\_density(aes(x = residual4), color = "darkgrey", lwd = 1)

**一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述**

**Answer**

According to the graphs presented above, the residuals are clearly normally distributed and centered around zero as well.