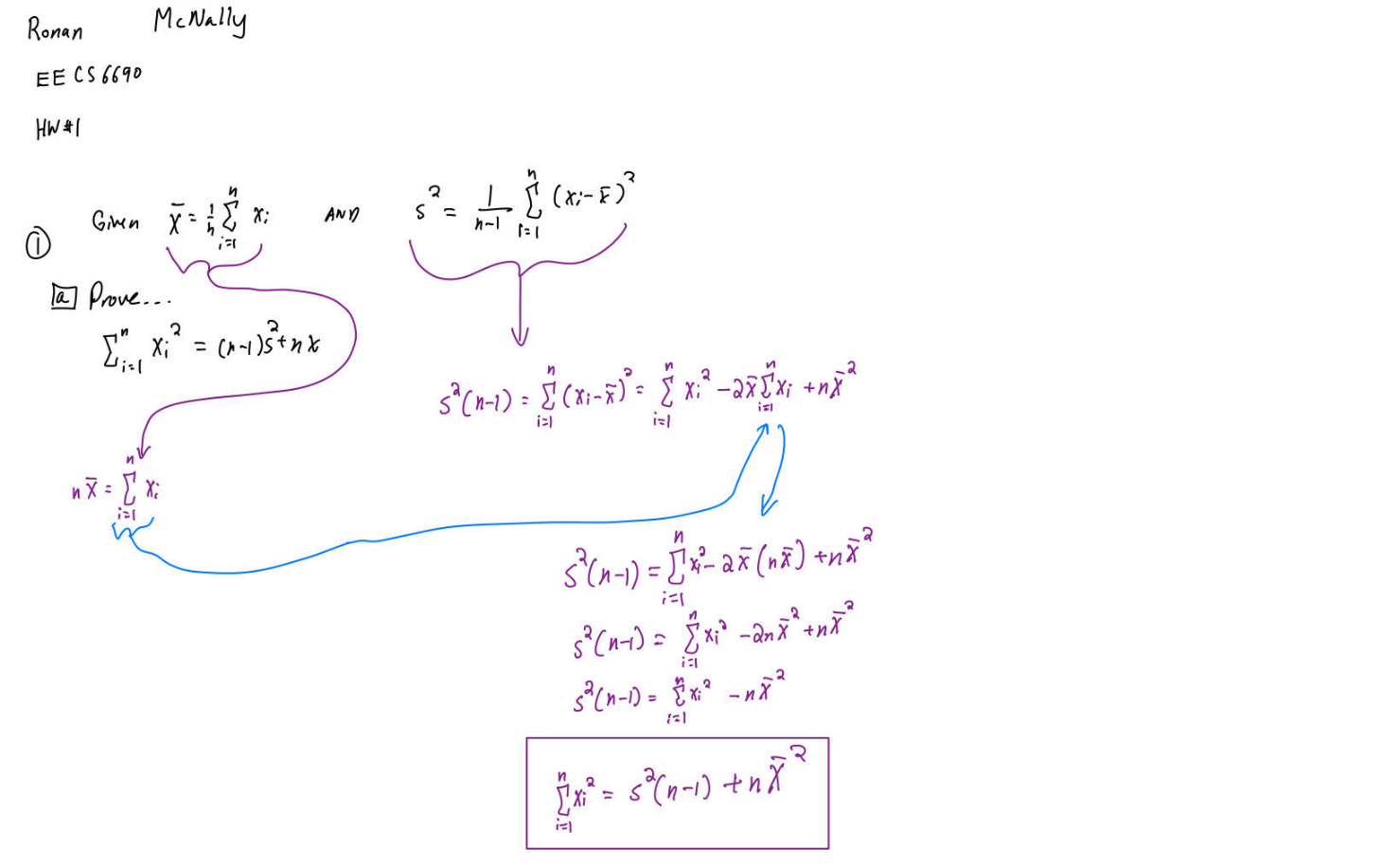
rm4064

Ronan McNally

2024-09-20

**Question 1 part (a)**



**Question 1 part (b)**

A person standing in front of a white board

Description automatically generated

**Question 1 part (c)**

A close-up of a white board with math equations

Description automatically generated

**Question 1 part (d)**

A close-up of a handwritten text

Description automatically generated

**Question 2**

A white board with black and white equations

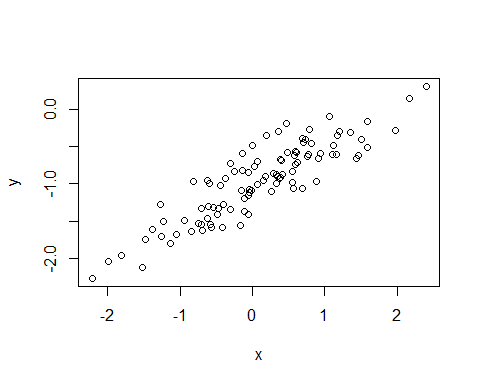
Description automatically generated with medium confidence

**Question 3**

# Q3.part(a)   
  
set.seed(1) # sets the seed for repoducible randomness  
  
x <- rnorm(100, mean = 0, sd = 1)  
  
  
# Q3.part(b)   
  
eps <- rnorm(100, mean = 0, sd = 0.25)  
  
# Q3.part(c)   
  
y <- -1 + 0.5\*x + eps  
  
length(y) # outputs length, length of vector y is 100

## [1] 100

# β0 is equal to -1, β1 is equal to 0.5 in the linear model  
  
  
  
# Q3.part(d)   
  
plot(x, y)



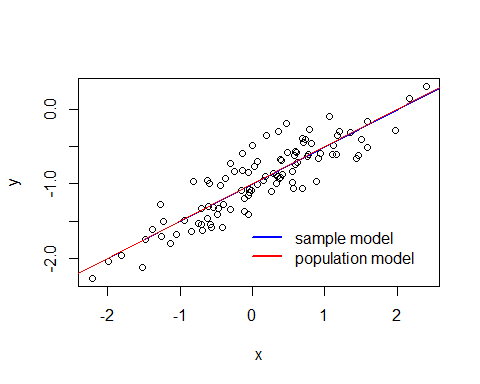
# Based off of visual appearance, it would appear y is somewhat directly proportional to x.

# Q3.part(e)  
sample\_model <- lm(y ~ x)  
  
summary(sample\_model)

##   
## 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 viewing the following summary data of the sample model, βhat0 is -1.00942 and βhat1 is 0.49973. βhat0 is slightly more than β0 and βhat1 is slightly less than β1 but both estimated betas closely approximate the true betas.

# Q3.part(f)  
  
plot(x, y)  
abline(sample\_model, col="blue")  
abline(-1, 0.5, col="red") #creates line equivalent to population model  
  
  
#legend("topright", legend = c("Sample Model", "Population Model"),)  
legend("bottomright", inset=.05,c("sample model","population model"), lwd=2, lty=c(1, 1), col=c("blue","red"), box.lty=0)



# Q3.part(g)  
  
polyfit\_model <- lm(y ~ x + I(x^2))  
summary((polyfit\_model))

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

anova(sample\_model, polyfit\_model)

## Analysis of Variance Table  
##   
## Model 1: y ~ x  
## Model 2: y ~ x + I(x^2)  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 98 5.6772   
## 2 97 5.5643 1 0.11291 1.9682 0.1638

# In the results from the summary of the polynomial fit, it shows a similarly significantly small p-value to that of the linear fit (2e-16) and a large F-value (which, while smaller than the linear fit, is still significant).

# Following this up with an anova test whose null hypothesis is that both models fit the data equally well, it's resulting P-value of 0.1638>0.05, therefore we do not reject the null hypothesis. Both of these results indicate a polynomial fit does not have a better fit. This is particularly underlined by the p-value for the x^2 term demonstrating weak significance (p\_I(x^2)=0.164>0.05) while the x term and intercept (the terms of the sample model) show strong significance with very small p values.

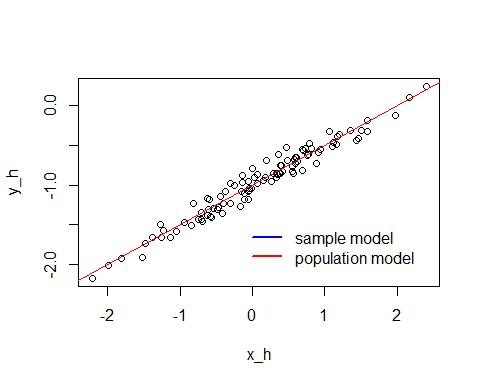
# Q3.part(h)  
  
set.seed(1)  
  
x\_h <- rnorm(100, mean = 0, sd = 1)  
  
eps\_h <- rnorm(100, mean = 0, sd = 0.1) # lowered from 0.25 to 0.1 --> point of fact: this is the stand dev term, not the variance as described in the question  
  
y\_h <- -1 + 0.5\*x\_h + eps\_h  
  
length(y\_h)

## [1] 100

plot(x\_h, y\_h)  
  
sample\_model\_h <- lm(y\_h ~ x\_h)  
  
summary(sample\_model\_h)

##   
## Call:  
## lm(formula = y\_h ~ x\_h)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.18768 -0.06138 -0.01395 0.05394 0.23462   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.003769 0.009699 -103.5 <2e-16 \*\*\*  
## x\_h 0.499894 0.010773 46.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09628 on 98 degrees of freedom  
## Multiple R-squared: 0.9565, Adjusted R-squared: 0.956   
## F-statistic: 2153 on 1 and 98 DF, p-value: < 2.2e-16

abline(sample\_model\_h, col="blue")  
abline(-1, 0.5, col="red")  
  
  
#legend("topright", legend = c("Sample Model", "Population Model"),)  
legend("bottomright", inset=.05,c("sample model","population model"), lwd=2, lty=c(1, 1), col=c("blue","red"), box.lty=0)



polyfit\_model\_h <- lm(y\_h ~ x\_h + I(x\_h^2))  
summary((polyfit\_model\_h))

##   
## Call:  
## lm(formula = y\_h ~ x\_h + I(x\_h^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.19650 -0.06254 -0.01288 0.05803 0.22700   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.994328 0.011766 -84.512 <2e-16 \*\*\*  
## x\_h 0.501716 0.010798 46.463 <2e-16 \*\*\*  
## I(x\_h^2) -0.011892 0.008477 -1.403 0.164   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0958 on 97 degrees of freedom  
## Multiple R-squared: 0.9573, Adjusted R-squared: 0.9565   
## F-statistic: 1088 on 2 and 97 DF, p-value: < 2.2e-16

anova(sample\_model\_h, polyfit\_model\_h)

## Analysis of Variance Table  
##   
## Model 1: y\_h ~ x\_h  
## Model 2: y\_h ~ x\_h + I(x\_h^2)  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 98 0.90836   
## 2 97 0.89029 1 0.018065 1.9682 0.1638

# By reducing the noise in the data, it appears that the βhat0 and βhat1 terms \*more\* closely approximate the β0 and β1 terms respectively (going from -1.00942 to -1.003769 and from 0.49973 to 0.499894 respectively) and t-values for both beta hats increase in magnitude, implying a better fit for the sample model of the population model (β0=-1, β1=0.5). The length of vector y remains the same. The scatterplot of x and y still show a direct proportional relationship but now more closely clustered about the sample model trendline. As for the quadratic fit, it appears it still remains no more an improvement of fit compared to the sample model with just x (the probability from both the summary and anova test remain the same, insignificant, values).

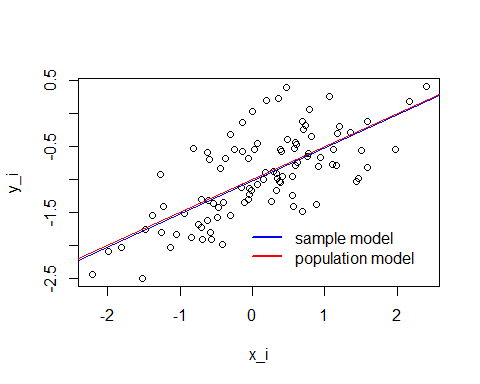
# Q3.part(i)  
  
set.seed(1)  
  
x\_i <- rnorm(100, mean = 0, sd = 1)  
  
eps\_i <- rnorm(100, mean = 0, sd = 0.5) # increased from 0.25 to 0.5 --> point of fact: this is the stand dev term, not the variance as described in the question  
  
y\_i <- -1 + 0.5\*x\_i + eps\_i  
  
length(y\_i)

## [1] 100

plot(x\_i, y\_i)  
  
sample\_model\_i <- lm(y\_i ~ x\_i)  
  
summary(sample\_model\_i)

##   
## Call:  
## lm(formula = y\_i ~ x\_i)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.93842 -0.30688 -0.06975 0.26970 1.17309   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.01885 0.04849 -21.010 < 2e-16 \*\*\*  
## x\_i 0.49947 0.05386 9.273 4.58e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4814 on 98 degrees of freedom  
## Multiple R-squared: 0.4674, Adjusted R-squared: 0.4619   
## F-statistic: 85.99 on 1 and 98 DF, p-value: 4.583e-15

abline(sample\_model\_i, col="blue")  
abline(-1, 0.5, col="red")  
  
  
#legend("topright", legend = c("Sample Model", "Population Model"),)  
legend("bottomright", inset=.05,c("sample model","population model"), lwd=2, lty=c(1, 1), col=c("blue","red"), box.lty=0)



polyfit\_model\_i <- lm(y\_i ~ x\_i + I(x\_i^2))  
summary((polyfit\_model\_i))

##   
## Call:  
## lm(formula = y\_i ~ x\_i + I(x\_i^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.98252 -0.31270 -0.06441 0.29014 1.13500   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.97164 0.05883 -16.517 < 2e-16 \*\*\*  
## x\_i 0.50858 0.05399 9.420 2.4e-15 \*\*\*  
## I(x\_i^2) -0.05946 0.04238 -1.403 0.164   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.479 on 97 degrees of freedom  
## Multiple R-squared: 0.4779, Adjusted R-squared: 0.4672   
## F-statistic: 44.4 on 2 and 97 DF, p-value: 2.038e-14

anova(sample\_model\_i, polyfit\_model\_i)

## Analysis of Variance Table  
##   
## Model 1: y\_i ~ x\_i  
## Model 2: y\_i ~ x\_i + I(x\_i^2)  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 98 22.709   
## 2 97 22.257 1 0.45163 1.9682 0.1638

# By increasing the noise in the data, it appears that the βhat0 and βhat1 terms \*less\* closely approximate the β0 and β1 terms respectively (going from -1.00942 to -1.01885 and from 0.49973 to 0.49947 respectively) and t-values for both beta hats decrease in magnitude, implying a worse fit for the sample model of the population model (β0=-1, β1=0.5). The length of vector y remains the same. The scatterplot of x and y still show a vague direct proportional relationship but now exhibits greater spread from the sample model trendline. As for the quadratic fit, it appears it still remains no more an improvement of fit compared to the sample model with just x (the significance of the x^2 factor from both the summary and anova test remain the same, insignificant, values). However, of note is that the significance of the intercept and x terms of the quadratic fit do decrease (smaller t-values and larger p values in the summary). This is to be expected as the quadratic model has shown no improvement over the linear fit and reduced "fitness" of the linear sample model becomes reflected in the summary data of the quadratic model.

# Q3.part(j)  
  
  
confint(sample\_model) # original model

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

confint(sample\_model\_h) # less noisy model

## 2.5 % 97.5 %  
## (Intercept) -1.0230161 -0.9845224  
## x\_h 0.4785159 0.5212720

confint(sample\_model\_i) # more noisy model

## 2.5 % 97.5 %  
## (Intercept) -1.1150804 -0.9226122  
## x\_i 0.3925794 0.6063602

# The confidence interval represents the range of values over which an estimate is likely to fall in. In a 95% confidence interval, this means that 95/100 times, an estimate for said parameter will fall between said interval/range. Below, the 2.5% and 97.5% columns are the lower and upper bounds on a normal distribution which represent 95% of said distribution symmetrically about the mean. As can be seen upon inspection of the confidence intervals below, as noise increases, the range of values for βhat0 (intercept) and βhat1 (x, x\_h, x\_i) become larger. This reflects the drop in precision our estimates experience as a result of noise.

**Question 4**

# Q4  
  
# first load data   
setwd("C:/Users/Owner/OneDrive - Northeastern University/Desktop/A\_Columbia/Courses/Fall2024/Machine Learning/Homework1")  
Advertising <- read.csv("Advertising.csv",header=T,na.strings="?")  
dim(Advertising)

## [1] 200 5

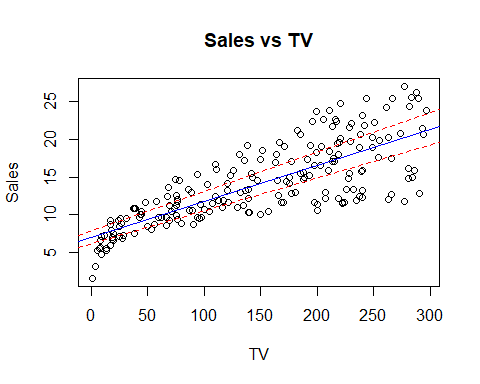
Advertising <- na.omit(Advertising)  
names(Advertising)

## [1] "X" "TV" "radio" "newspaper" "sales"

lmP4\_TVsales <- lm(Advertising$sales~Advertising$TV)  
confint(lmP4\_TVsales, level=0.92) # confidence Intervals For Model Parameters

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

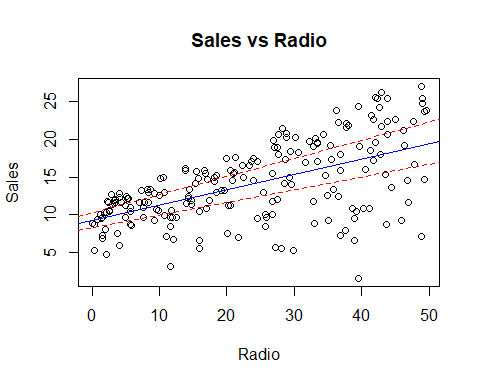
plot(Advertising$TV, Advertising$sales, xlab="TV", ylab="Sales", main = "Sales vs TV")  
abline(6.22691926, .04330989, col="red", lty=2)  
abline(7.83826784, 0.05227135, col="red", lty=2)  
abline(lmP4\_TVsales, col="blue")



lmP4\_Radiosales <- lm(Advertising$sales~Advertising$radio)  
confint(lmP4\_Radiosales, level=0.92) # confidence Intervals For Model Parameters

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

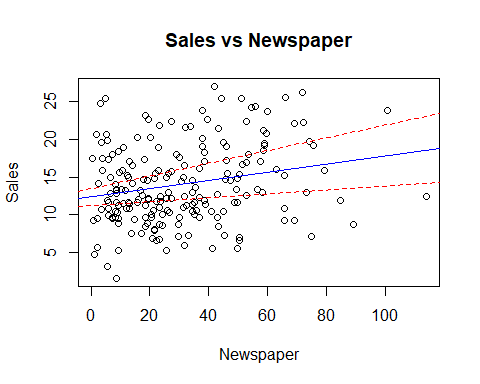
plot(Advertising$radio, Advertising$sales, xlab="Radio", ylab="Sales", main = "Sales vs Radio")  
abline(8.3210922, 0.1665776, col="red", lty=2)  
abline(10.3021840, 0.2384139, col="red", lty=2)  
abline(lmP4\_Radiosales, col="blue")



lmP4\_Newspapersales <- lm(Advertising$sales~Advertising$newspaper)  
confint(lmP4\_Newspapersales, level=0.92) # confidence Intervals For Model Parameters

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

plot(Advertising$newspaper, Advertising$sales, xlab="Newspaper", ylab="Sales", main = "Sales vs Newspaper")  
abline(11.25788302 , 0.02552451 , col="red", lty=2)  
abline(13.44493112, 0.08386169, col="red", lty=2)  
abline(lmP4\_Newspapersales, col="blue")



**Question 5**

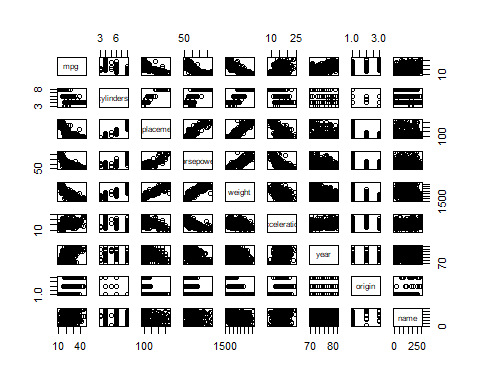
# first load data  
  
# first load data   
Auto <- read.csv("Auto.csv",header=T,na.strings="?")  
dim(Auto)

## [1] 397 9

Auto <- na.omit(Auto)  
names(Auto)

## [1] "mpg" "cylinders" "displacement" "horsepower" "weight"   
## [6] "acceleration" "year" "origin" "name"

# Q5.part(a)  
  
plot(Auto)



# Q5.part(b)  
  
cor(Auto[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  
## acceleration 1.0000000 0.2903161 0.2127458  
## year 0.2903161 1.0000000 0.1815277  
## origin 0.2127458 0.1815277 1.0000000

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

# Q5.part(c)  
  
lmP5 <- lm(Auto$mpg ~ Auto$cylinders+Auto$displacement+Auto$horsepower+Auto$weight+Auto$acceleration+Auto$year+Auto$origin)  
  
summary(lmP5)

##   
## Call:  
## lm(formula = Auto$mpg ~ Auto$cylinders + Auto$displacement +   
## Auto$horsepower + Auto$weight + Auto$acceleration + Auto$year +   
## Auto$origin)  
##   
## 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 \*\*\*  
## Auto$cylinders -0.493376 0.323282 -1.526 0.12780   
## Auto$displacement 0.019896 0.007515 2.647 0.00844 \*\*   
## Auto$horsepower -0.016951 0.013787 -1.230 0.21963   
## Auto$weight -0.006474 0.000652 -9.929 < 2e-16 \*\*\*  
## Auto$acceleration 0.080576 0.098845 0.815 0.41548   
## Auto$year 0.750773 0.050973 14.729 < 2e-16 \*\*\*  
## Auto$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

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

# (ci) Yes, assuming an alpha=0.05, there is an apparent relationship between the predictors and the response  
  
# (cii) The predictors "year" and "weight" appear to have the most statistically significant relationship to the response. This is followed by "origin." Still significantly significant but by a much smaller margin are "cylinders" and "displacement." The remainder are statistically insignificant as indicators.  
  
# (ciii) The coefficient of "year" is 0.750773. This suggests a direct proportional relationship between mpg and year of the car such that around every 4 improvements in car design and manufacturing mean you can travel 3 miles further for the same gallon of gas.  
  
  
# Q5.part(d)  
  
lmP5\_log <- lm(Auto$mpg ~ log(Auto$cylinders)+log(Auto$displacement)+log(Auto$horsepower)+log(Auto$weight)+log(Auto$acceleration)+log(Auto$year)+log(Auto$origin))  
  
summary(lmP5\_log)

##   
## Call:  
## lm(formula = Auto$mpg ~ log(Auto$cylinders) + log(Auto$displacement) +   
## log(Auto$horsepower) + log(Auto$weight) + log(Auto$acceleration) +   
## log(Auto$year) + log(Auto$origin))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.5987 -1.8172 -0.0181 1.5906 12.8132   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -66.5643 17.5053 -3.803 0.000167 \*\*\*  
## log(Auto$cylinders) 1.4818 1.6589 0.893 0.372273   
## log(Auto$displacement) -1.0551 1.5385 -0.686 0.493230   
## log(Auto$horsepower) -6.9657 1.5569 -4.474 1.01e-05 \*\*\*  
## log(Auto$weight) -12.5728 2.2251 -5.650 3.12e-08 \*\*\*  
## log(Auto$acceleration) -4.9831 1.6078 -3.099 0.002082 \*\*   
## log(Auto$year) 54.9857 3.5555 15.465 < 2e-16 \*\*\*  
## log(Auto$origin) 1.5822 0.5083 3.113 0.001991 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.069 on 384 degrees of freedom  
## Multiple R-squared: 0.8482, Adjusted R-squared: 0.8454   
## F-statistic: 306.5 on 7 and 384 DF, p-value: < 2.2e-16

lmP5\_sqrt <- lm(Auto$mpg ~ sqrt(Auto$cylinders)+sqrt(Auto$displacement)+sqrt(Auto$horsepower)+sqrt(Auto$weight)+sqrt(Auto$acceleration)+sqrt(Auto$year)+sqrt(Auto$origin))  
  
summary(lmP5\_sqrt)

##   
## Call:  
## lm(formula = Auto$mpg ~ sqrt(Auto$cylinders) + sqrt(Auto$displacement) +   
## sqrt(Auto$horsepower) + sqrt(Auto$weight) + sqrt(Auto$acceleration) +   
## sqrt(Auto$year) + sqrt(Auto$origin))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.5250 -1.9822 -0.1111 1.7347 13.0681   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -49.79814 9.17832 -5.426 1.02e-07 \*\*\*  
## sqrt(Auto$cylinders) -0.23699 1.53753 -0.154 0.8776   
## sqrt(Auto$displacement) 0.22580 0.22940 0.984 0.3256   
## sqrt(Auto$horsepower) -0.77976 0.30788 -2.533 0.0117 \*   
## sqrt(Auto$weight) -0.62172 0.07898 -7.872 3.59e-14 \*\*\*  
## sqrt(Auto$acceleration) -0.82529 0.83443 -0.989 0.3233   
## sqrt(Auto$year) 12.79030 0.85891 14.891 < 2e-16 \*\*\*  
## sqrt(Auto$origin) 3.26036 0.76767 4.247 2.72e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.21 on 384 degrees of freedom  
## Multiple R-squared: 0.8338, Adjusted R-squared: 0.8308   
## F-statistic: 275.3 on 7 and 384 DF, p-value: < 2.2e-16

lmP5\_sqr <- lm(Auto$mpg ~ (Auto$cylinders)^2+(Auto$displacement)^2+(Auto$horsepower)^2+(Auto$weight)^2+(Auto$acceleration)^2+(Auto$year)^2+(Auto$origin)^2)  
  
summary(lmP5\_sqr)

##   
## Call:  
## lm(formula = Auto$mpg ~ (Auto$cylinders)^2 + (Auto$displacement)^2 +   
## (Auto$horsepower)^2 + (Auto$weight)^2 + (Auto$acceleration)^2 +   
## (Auto$year)^2 + (Auto$origin)^2)  
##   
## 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 \*\*\*  
## Auto$cylinders -0.493376 0.323282 -1.526 0.12780   
## Auto$displacement 0.019896 0.007515 2.647 0.00844 \*\*   
## Auto$horsepower -0.016951 0.013787 -1.230 0.21963   
## Auto$weight -0.006474 0.000652 -9.929 < 2e-16 \*\*\*  
## Auto$acceleration 0.080576 0.098845 0.815 0.41548   
## Auto$year 0.750773 0.050973 14.729 < 2e-16 \*\*\*  
## Auto$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

# RESULTS OF TRANSFORMS: Across all transforms, year and weight remain highly statistically significant as an indicator. When log transform is applied, acceleration and horsepower becomes statistically significant indicators while origin weakens in significance and displacement becomes insignificant. Under the sqrt() transform of the variables, displacement becomes statistically insignificant while horsepower becomes (weakly) statistically significant. Finally, no change in significants appears to occur for any of the indicators upon squaring them.

**Question 6**

**A math equations and formulas on a white background

Description automatically generated**

**Question 7**

**A white board with text and numbers

Description automatically generated**