Linear Regression

Linear Regression

Key concepts: The linear regression model, least squares method, assessing the fit of a model, multiple linear regression, inference, categorical variables, modeling nonlinear relationships, model fitting, big data and regression.

Data

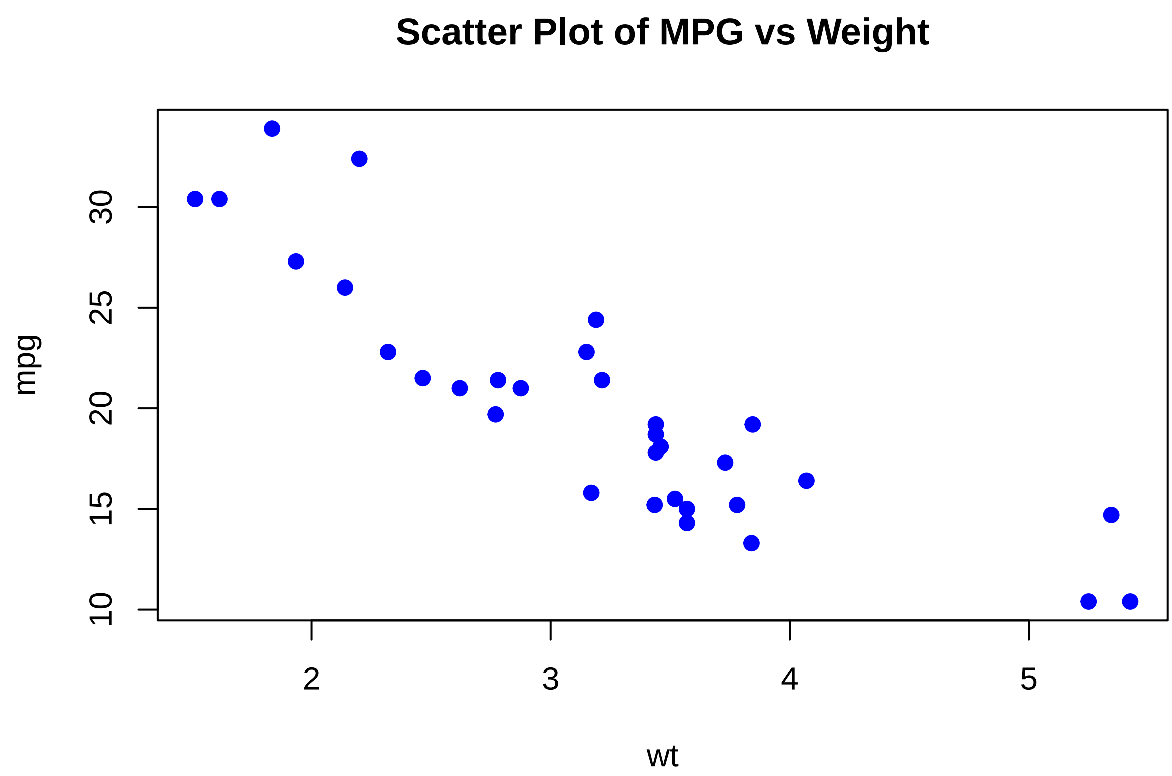
```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Simple Linear Regression

Create a scatterplot of MPG as the dependent variable and weight as the independent variable.

```
#Create scatterplot
plot(mpg~wt, data=mtcars,
     main = "Scatter Plot of MPG vs Weight",
     col ="blue",
     pch=19)
```




```
# #ggplot
# ggplot(data=mtcars) +
#   geom_point(aes(x=wt, y=mpg), color = "blue") +
#   labs(title = "Scatter Plot of MPG vs Weight")
```

Use the data to develop an estimated regression equation. Use MPG as the dependent variable and weight as the independent variable.

```
mtcars.out <- lm(mpg~wt, data=mtcars)
```

lm() is a linear model function. glm() is another function that does the same thing.

The variable before the ~ is the response (independent) variable and everything after the ~ is the predictor (dependent) variable(s).

Look at the output

```
print(mtcars.out)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Coefficients:
## (Intercept)           wt
##      37.285       -5.344
```

This gives the formula and the coefficients

Look at the summary of it as well.

```
summary(mtcars.out)
```

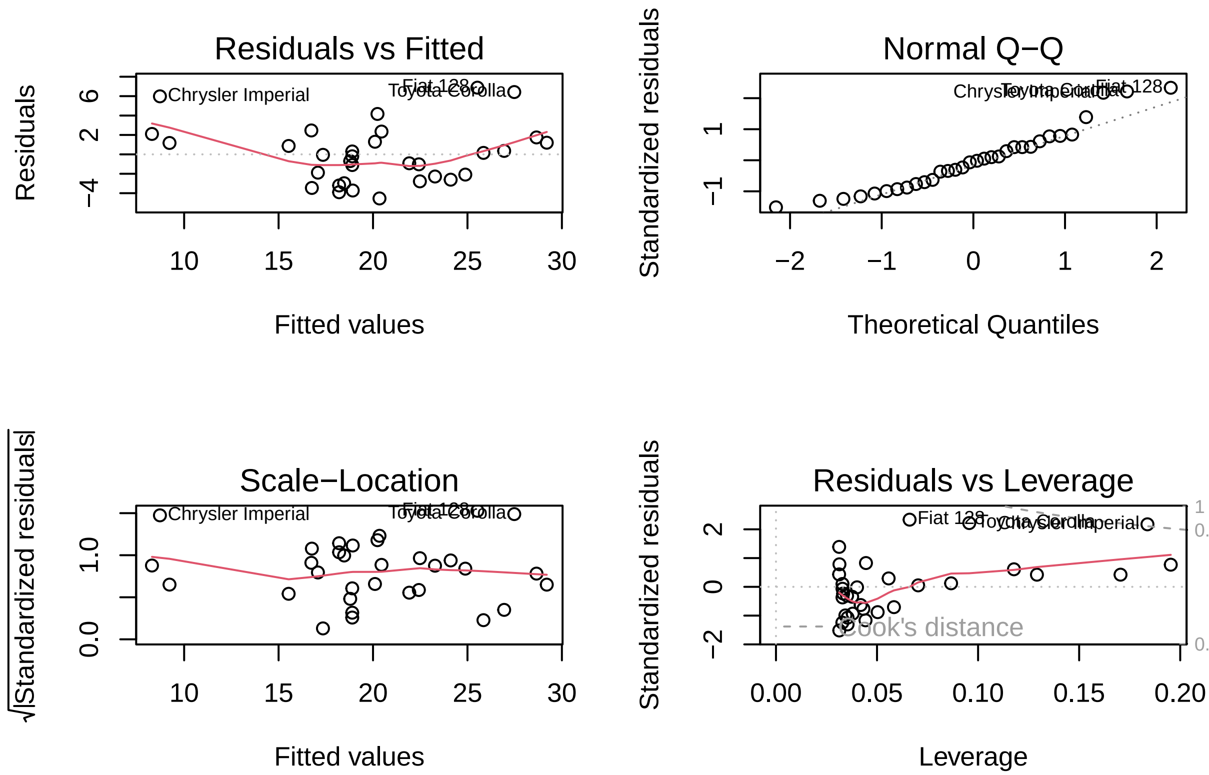
```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

This gives more detailed information.

The model is y_hat = 37.2851 - 5.3445*x, where x=weight

It is extremely important that you check the assumptions of your model before doing any inference. Though in this class the assumptions will almost always be met, in real life, they often are not. To do this, run the following code:

```
#First check assumptions before doing inference.
par(mfrow=c(2,2)) #This makes it so that you can see all 4 plots
plot(mtcars.out)
```




```r
par(mfrow=c(1,1)) #This resets the format for future plots
```

A flat horizontal line for the Residuals vs Fitted Plots indicates independence.

A a straight line for the Normal QQ Plot indicates normality.

A flat horizontal line for the Scale-Location Plot indicates equal variance.

No points over the red dotted lines indicates no influential points for the Residuals vs Leverage graph.

The summary has a lot of information that we can look at individually.

What are the estimates of the regression line?

```r
summary(mtcars.out)$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 37.285126   1.877627 19.857575 8.241799e-19
## wt          -5.344472   0.559101 -9.559044 1.293959e-10
```

How much variation is explained by x?

```r
summary(mtcars.out)$r.squared
```

```
## [1] 0.7528328
```

What is the regression formula?

```
summary(mtcars.out)$call
```

```
## lm(formula = mpg ~ wt, data = mtcars)
```

What is the standard deviation?

```
summary(mtcars.out)$sigma
```

```
## [1] 3.045882
```

Conduct a 95% confidence interval the estimates:

```
confint(mtcars.out, level=0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 33.450500 41.119753
## wt          -6.486308 -4.202635
```

Get the p-values of the estimates

```
summary(mtcars.out)$coefficients[,4]
```

```
##  (Intercept)           wt
## 8.241799e-19 1.293959e-10
```

What would you predict the mean mpg to be if the weight is 3.0?

```
predict(mtcars.out, newdata=data.frame(wt=3.0))
```

```
##        1
## 21.25171
```

You can reuse this code by changing the linear model (mtcars.out), the variable (wt) and the variable value (3.0).

What would you predict the weight to be if the mpg is 25?

```
approx(x = mtcars.out$fitted.values, y = mtcars$wt, xout = 25)$y
```

```
## [1] 2.298661
```

A warning message will be outputted in the console because it is an approximation.

To reuse this code, change the linear model (mtcars.out), the y-variable (mtcars$wt), and the input amount (25).

Multiple Linear Regression

Use the data to develop an estimated regression equation. Use MPG as the dependent variable with weight and horsepower as the independent variable.

```r
mtcars.out2 <- lm(mpg~wt+hp, data=mtcars)
```

Use + to add more variables to the model

Look at the output

```r
print(mtcars.out2)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp, data = mtcars)
##
## Coefficients:
## (Intercept)           wt           hp
##    37.22727     -3.87783     -0.03177
```

Look at the summary of it as well. This has more information.

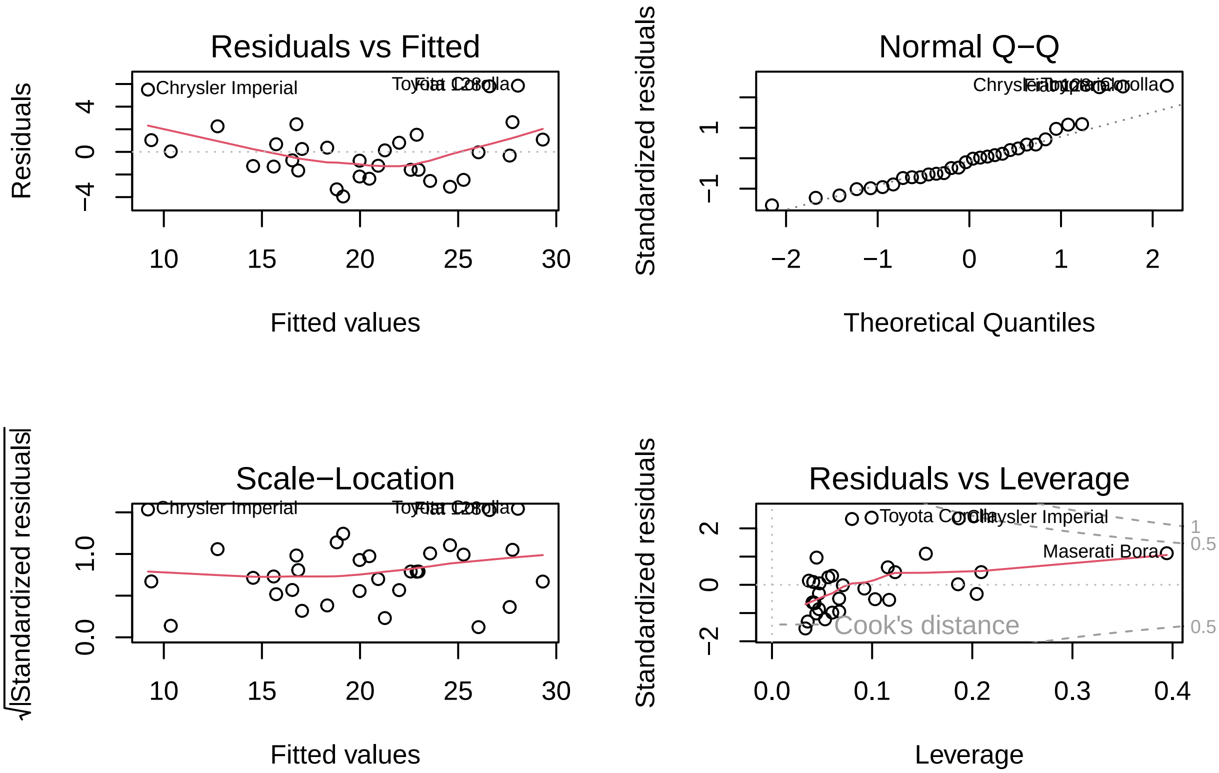
```r
summary(mtcars.out2)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## hp          -0.03177    0.00903  -3.519  0.00145 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

The model is y_hat = 37.22727 - 3.87783x1 -0.03177*x2, where x1=weight and x2=horsepower

It is extremely important that you check the assumptions of your model before doing any inference. Though in this class the assumptions will almost always be met, in real life, they often are not. To do this, run the following code:

```r
#First check assumptions before doing inference.
par(mfrow=c(2,2))
plot(mtcars.out2)
```




```
par(mfrow=c(1,1))
```

A flat horizontal line for the Residuals vs Fitted Plots indicates independence.

A a straight line for the Normal QQ Plot indicates normality.

A flat horizontal line for the Scale-Location Plot indicates equal variance.

No points over the red dotted lines indicates no influential points for the Residuals vs Leverage graph.

What are the estimates of the regression line?

```
summary(mtcars.out2)$coefficients
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 37.22727012 1.59878754 23.284689 2.565459e-20
## wt          -3.87783074 0.63273349 -6.128695 1.119647e-06
## hp          -0.03177295 0.00902971 -3.518712 1.451229e-03
```

How much variation is explained by x?

```
summary(mtcars.out2)$r.squared
```

```
## [1] 0.8267855
```

Conduct a 95% confidence interval the estimates:

```r
confint(mtcars.out, level=0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 33.450500 41.119753
## wt          -6.486308 -4.202635
```

Get the p-values of the estimates

```r
summary(mtcars.out2)$coefficients[,4]
```

```
##  (Intercept)           wt           hp 
## 2.565459e-20 1.119647e-06 1.451229e-03
```

What would you predict the mean mpg to be if the weight is 3.5 and horsepower of 100?

```r
predict(mtcars.out2, newdata=data.frame(wt=3.5, hp=100))
```

```
##        1 
## 20.47757
```

List the values and their variables in within the data.frame separated by commas.

Calculate the predicted price and residual for each automobile in the data

```r
predict(mtcars.out2)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive 
##           23.572329           22.583483           25.275819           21.265020 
##   Hornet Sportabout             Valiant          Duster 360           Merc 240D 
##           18.327267           20.473816           15.599042           22.887067 
##            Merc 230            Merc 280           Merc 280C          Merc 450SE 
##           21.993673           19.979460           19.979460           15.725369 
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental 
##           17.043831           16.849939           10.355205            9.362733 
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla 
##            9.192487           26.599028           29.312380           28.046209 
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28 
##           24.586441           18.811364           19.140979           14.552028 
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa 
##           16.756745           27.626653           26.037374           27.769769 
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E 
##           16.546489           20.925413           12.739477           22.983649
```

```r
resid(mtcars.out2)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive 
##         -2.57232940         -1.58348256         -2.47581872          0.13497989 
##   Hornet Sportabout             Valiant          Duster 360           Merc 240D 
##          0.37273336         -2.37381631         -1.29904236          1.51293266 
##            Merc 230            Merc 280           Merc 280C          Merc 450SE
```

```
##          0.80632669         -0.77945988         -2.17945988          0.67463146
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental
##          0.25616901         -1.64993945          0.04479541          1.03726743
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla
##          5.50751301          5.80097202          1.08761978          5.85379085
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28
##         -3.08644148         -3.31136386         -3.94097947         -1.25202805
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
##          2.44325481         -0.32665313         -0.03737415          2.63023081
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
##         -0.74648866         -1.22541324          2.26052287         -1.58364943
```

Sort the data by residuals (smallest to largest)

```
mtcars$predict <- predict(mtcars.out)
mtcars$resid <- resid(mtcars.out)
#Sort the data
library(dplyr)
mtcars %>% arrange(resid)
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
```

```
##                      predict      resid
## Ford Pantera L      20.343151 -4.5431513
## Duster 360          18.205363 -3.9053627
## AMC Javelin         18.926866 -3.7268663
## Camaro Z28          16.762355 -3.4623553
## Maserati Bora       18.205363 -3.2053627
## Dodge Challenger    18.472586 -2.9725862
## Ferrari Dino        22.480940 -2.7809399
## Toyota Corona       24.111004 -2.6110037
## Mazda RX4           23.282611 -2.2826106
## Datsun 710          24.885952 -2.0859521
## Merc 450SLC         17.083024 -1.8830236
## Merc 280C           18.900144 -1.1001440
## Volvo 142E          22.427495 -1.0274952
## Mazda RX4 Wag       21.919770 -0.9197704
## Valiant             18.793255 -0.6932545
## Hornet Sportabout   18.900144 -0.2001440
## Merc 450SL          17.350247 -0.0502472
## Porsche 914-2       25.847957  0.1520430
## Merc 280            18.900144  0.2998560
## Fiat X1-9           26.943574  0.3564263
## Merc 450SE          15.533127  0.8668731
## Cadillac Fleetwood   9.226650  1.1733496
## Lotus Europa        29.198941  1.2010593
## Hornet 4 Drive      20.102650  1.2973499
## Honda Civic         28.653805  1.7461954
## Lincoln Continental  8.296712  2.1032876
## Merc 230            20.450041  2.3499593
## Pontiac Firebird    16.735633  2.4643670
## Merc 240D           20.236262  4.1637381
## Chrysler Imperial    8.718926  5.9810744
## Toyota Corolla      27.478021  6.4219792
## Fiat 128            25.527289  6.8727113
```

Model Selection

Use the data to develop an estimated regression equation. Use MPG as the dependent variable and weight, horsepower, number of cylinders, displacement, rear axle ratio, 1/4 mile time, number of forward gears, and number of carburetors as the independent variables.

```
mtcars.out3 <- lm(mpg~wt+hp+cyl+disp+drat+qsec+gear+carb, data=mtcars)
```

Look at the summary coefficients and then determines which variables are not significant. Then rerun the linear model. Use alpha=0.1

```
#Option 1
summary(mtcars.out3)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp + cyl + disp + drat + qsec + gear +
```

```
##     carb, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0230 -1.6874 -0.4109  0.9640  5.4400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.88964   17.81996   1.004   0.3259
## wt          -3.92065    1.86174  -2.106   0.0463 *
## hp          -0.02085    0.02072  -1.006   0.3248
## cyl         -0.41460    0.95765  -0.433   0.6691
## disp         0.01293    0.01758   0.736   0.4694
## drat         1.10110    1.59806   0.689   0.4977
## qsec         0.54146    0.62122   0.872   0.3924
## gear         1.23321    1.40238   0.879   0.3883
## carb        -0.25510    0.81563  -0.313   0.7573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.622 on 23 degrees of freedom
## Multiple R-squared:  0.8596, Adjusted R-squared:  0.8107
## F-statistic:  17.6 on 8 and 23 DF,  p-value: 4.226e-08
```

```r
#Variables that have a Pr(>|t|) that are greater than 0.05 are not significant and those that have a Pr
#Fit the model to the significant variables
mtcars.out4 <- lm(mpg~wt, data=mtcars)

#Option 2
#Using Code
#This code can be used again with a few edits. This is more automatic. You are not expect to know this
toselect.x <- summary(mtcars.out3)$coeff[,4] < 0.05 #p-value is adjustable, change for data file
# select significant variables
relevant.x <- names(toselect.x)[toselect.x == TRUE]
# formula with only significant variables
mod1 <- paste("mpg~",paste(relevant.x, collapse="+"),sep = "") #Change to adjust for y-variable "mpg"
mtcars.out4 <- lm(mod1,data=mtcars) #adjust for data name
```

Since only weight (wt) is a significant variable in this example, we will only use this variable. This model happens to be the same one as the first one we looked at.

Regression with Categorical Data

Create a linear model to predict mpg from weight, engine shape, and transmission type. For the engine type (am), use 0 = V-shaped, 1 = straight. For the transmission type (vs), use 0 = automatic, 1 = manual.

If the data are strings, convert them to 0 or 1:

```r
#mtcars$vs[mtcars$vs=="string1"] <- 0
#mtcars$am[mtcars$am=="string2"] <- 1
#mtcars$vs <- as.numeric(mtcars$vs)
#mtcars$am <- as.numeric(mtcars$am)
```

To reuse this code change the data (mtcars) and variables (vs and am) to match what you are trying to do. Also, change string1 and string2 to match the strings in your dataset. Note that you can run a regression model without changing the values to 0 or 1, however, it could affect your model if you expect a variable to be a 1 and it R treats it like a 0. If there are more than 2 factors, include the variable without any of the above manipulations.

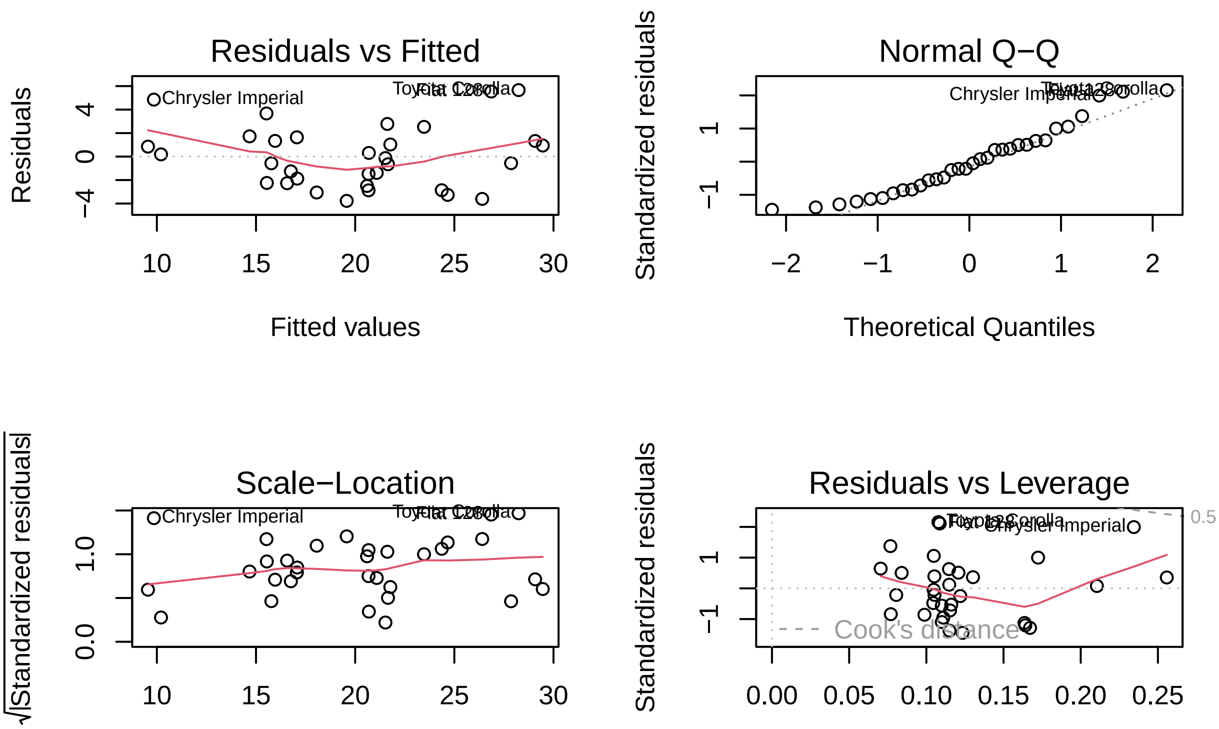
Model the data with a regression

```
mtcars.out5 <- lm(mpg~wt+as.factor(vs)+am, data=mtcars)
```

Change vs to a factor so that it is treated as a qualitative variable and not a quantitative one.

First check assumptions before doing inference.

```
par(mfrow=c(2,2))
plot(mtcars.out5)
```




```
par(mfrow=c(1,1))
```

What are the estimates of the regression line?

```
summary(mtcars.out5)
```

```
##
## Call:
```

```
## lm(formula = mpg ~ wt + as.factor(vs) + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7733 -2.2519 -0.3445  1.4129  5.6594
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      30.0787     3.7480   8.025 9.71e-09 ***
## wt               -3.7845     0.8981  -4.214 0.000236 ***
## as.factor(vs)1    3.6150     1.2761   2.833 0.008454 **
## am                1.4913     1.4863   1.003 0.324262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.779 on 28 degrees of freedom
## Multiple R-squared:  0.8079, Adjusted R-squared:  0.7873
## F-statistic: 39.25 on 3 and 28 DF,  p-value: 3.659e-10
```

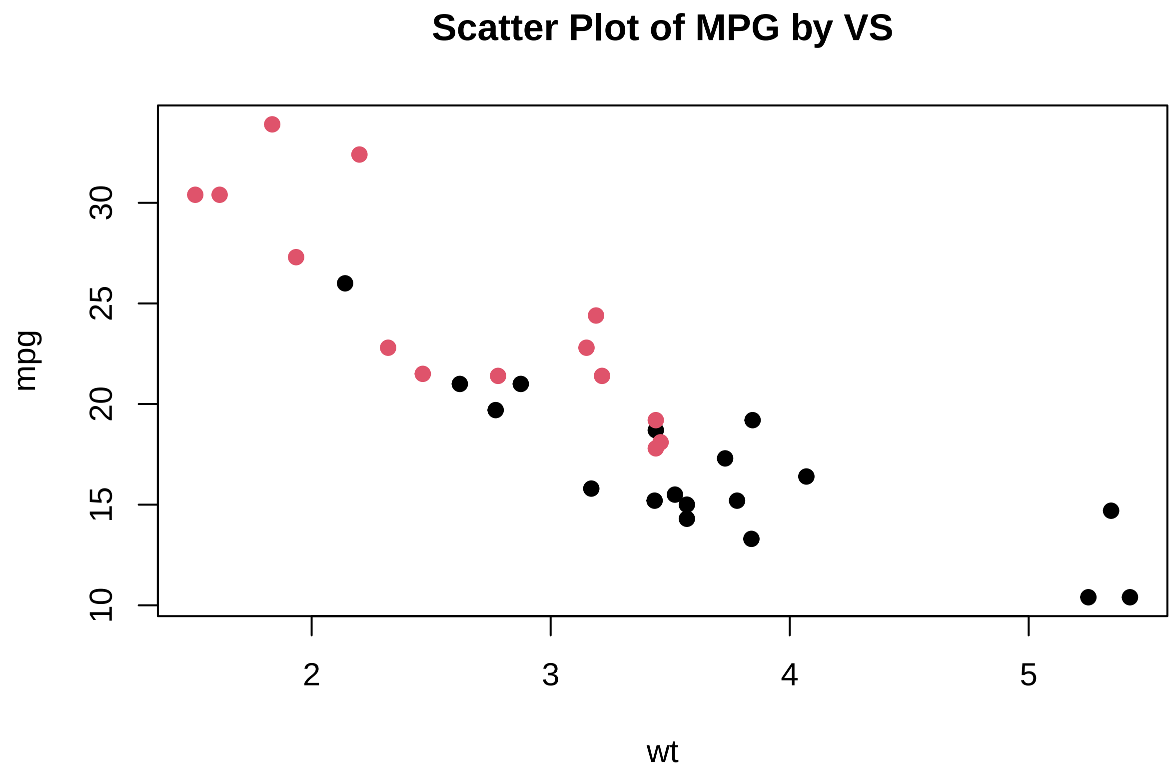
What is a 95% CI of the parameters?

```
confint(mtcars.out5, level=0.95)
```

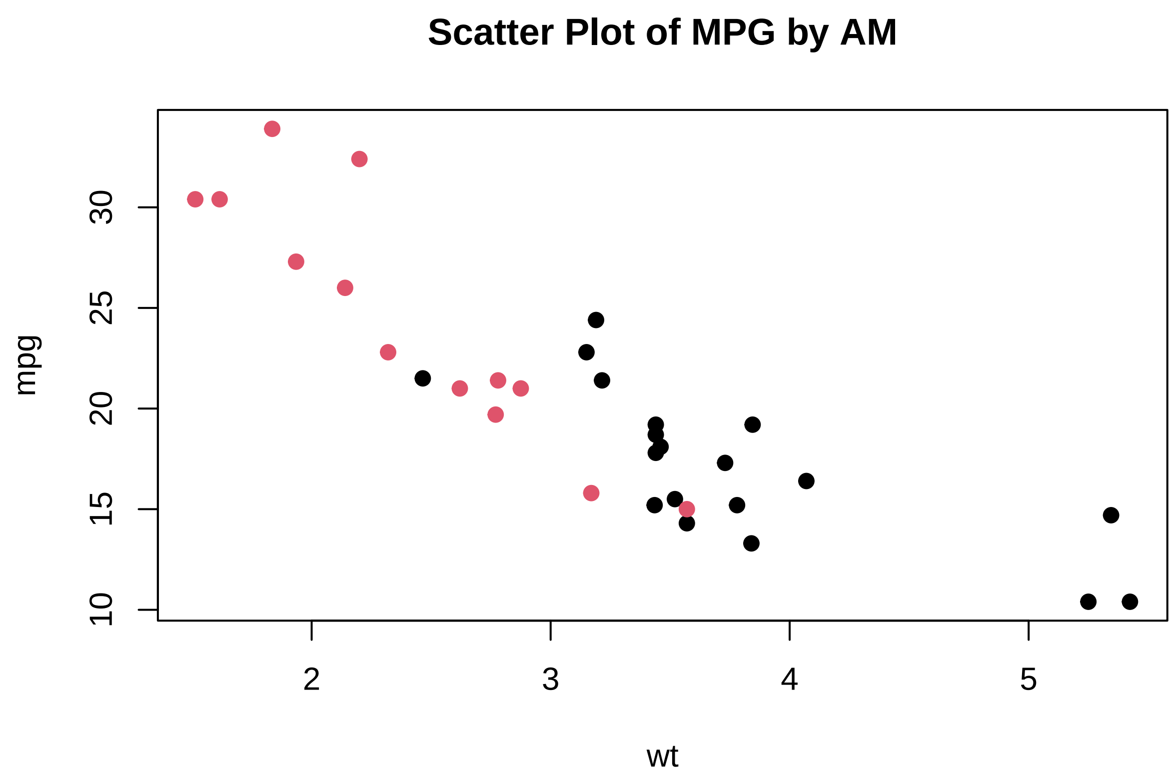
```
##                     2.5 %    97.5 %
## (Intercept)     22.401340 37.756104
## wt              -5.624187 -1.944722
## as.factor(vs)1   1.001166  6.228914
## am              -1.553193  4.535883
```

Using the first linear model mtcars.out <- lm(mpg~wt, data=mtcars), plot mpg as a function of weight using vs and am as different colors.

```
#Create scatter plots that color based on transmission type (vs)
plot(mpg~wt, data=mtcars,
     main = "Scatter Plot of MPG by VS",
     col = as.factor(mtcars$vs),
     pch = 19)
```




```
#Create scatter plots that color based on engine type (am)
plot(mpg~wt, data=mtcars,
     main = "Scatter Plot of MPG by AM",
     col = as.factor(mtcars$am),
     pch = 19)
```



Quadratic Regression

Create a quadratic regression predicting mpg from weight and weight squared.

Create a new variable that is a value squared

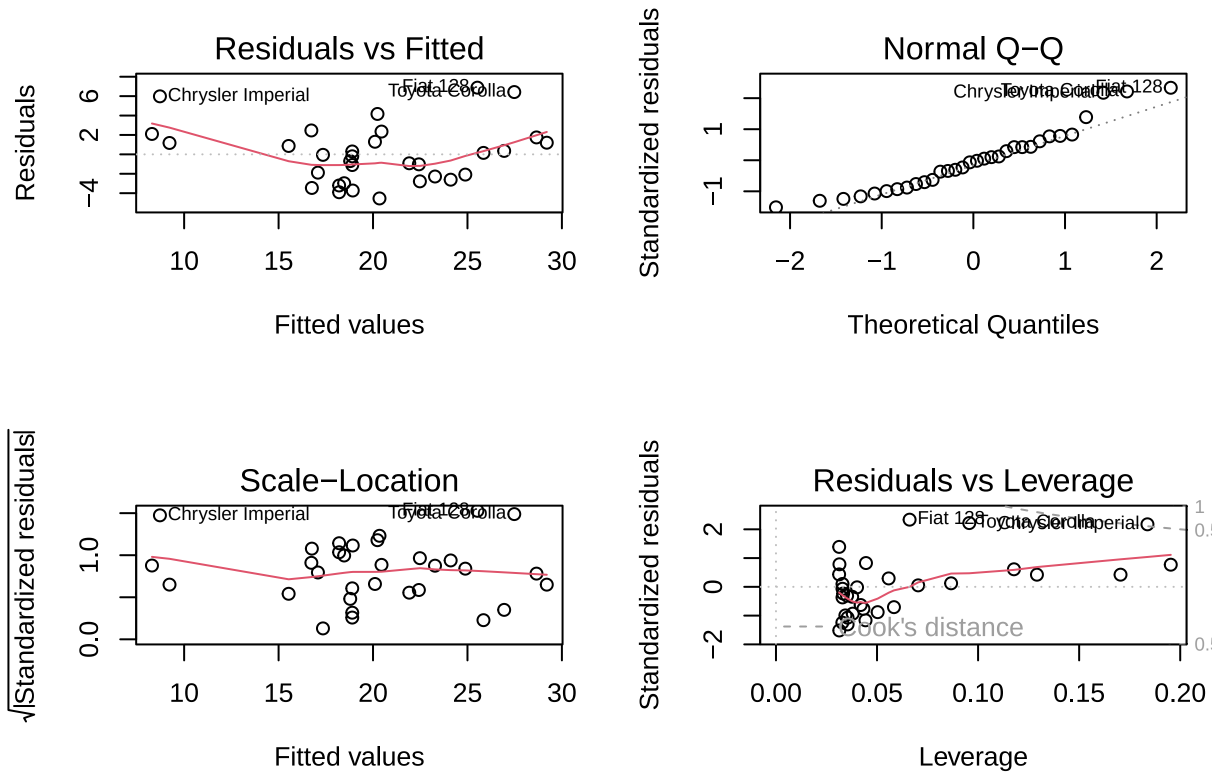
```
mtcars$wt_2 <- mtcars$wt^2
```

Model the quadratic regression

```
mtcars.out6 <- lm(mpg~wt+wt_2, data=mtcars)
```

First check assumptions before doing inference.

```
par(mfrow=c(2,2))
plot(mtcars.out)
```




```
par(mfrow=c(1,1))
```

What are the estimates of the regression line?

```
summary(mtcars.out5)
```

```
##
## Call:
## lm(formula = mpg ~ wt + as.factor(vs) + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7733 -2.2519 -0.3445  1.4129  5.6594
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     30.0787     3.7480   8.025 9.71e-09 ***
## wt              -3.7845     0.8981  -4.214 0.000236 ***
## as.factor(vs)1   3.6150     1.2761   2.833 0.008454 **
## am               1.4913     1.4863   1.003 0.324262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.779 on 28 degrees of freedom
## Multiple R-squared:  0.8079, Adjusted R-squared:  0.7873
## F-statistic: 39.25 on 3 and 28 DF,  p-value: 3.659e-10
```

What is a 95% CI of the parameters?

```
confint(mtcars.out6, level=0.95)
```

```
##                   2.5 %    97.5 %
## (Intercept)  41.3177599 58.543862
## wt          -18.5220551 -8.238619
## wt_2          0.4359382  1.906236
```

Check if the quadratic linear model is significantly different than a simple linear model. Use alpha=0.05

```
anova(mtcars.out6,mtcars.out)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + wt_2
## Model 2: mpg ~ wt
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     29 203.75
## 2     30 278.32 -1   -74.576 10.615 0.00286 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value is less alpha, we conclude that the two models are statistically significant. If both models satisfy the assumptions of a linear model, then we are usually want the model with the larger R-squared, given that it fits the model well.