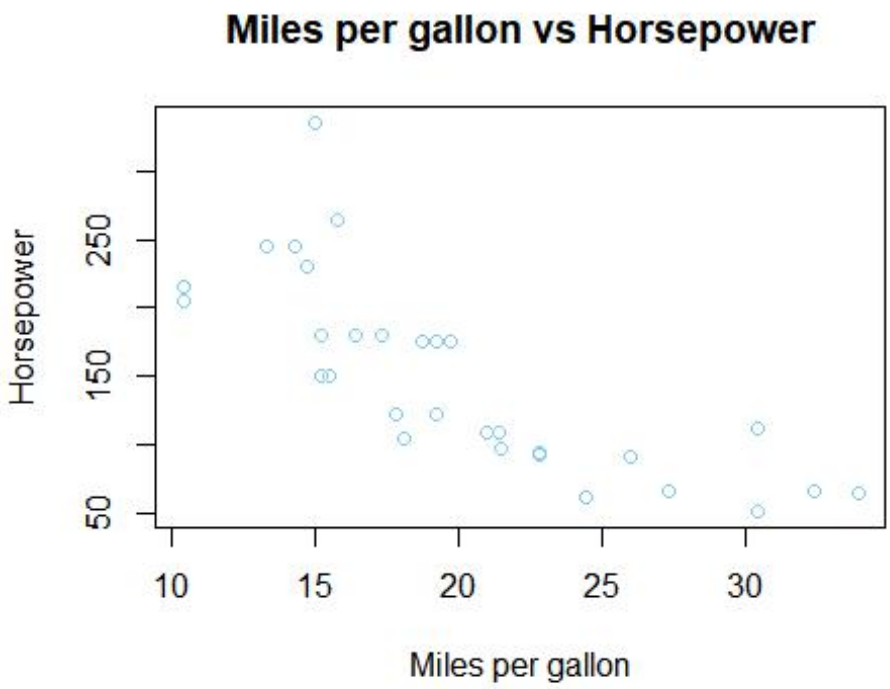
assignment–1R

2019-12-22

```
data(mtcars)
cars=mtcars
head(cars)
```

```
##                    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
```

```
plot(cars$mpg,cars$hp,type="p",col="skyblue",xlab="Miles per gallon",ylab="Ho
rsepower"
     ,main="Miles per gallon vs Horsepower")
```




```
#Simple linear model
relation=lm(cars$mpg~cars$hp)
print(relation)
```

```
##
## Call:
## lm(formula = cars$mpg ~ cars$hp)
##
## Coefficients:
## (Intercept)      cars$hp
##    30.09886     -0.06823
```

```
print(summary(relation,correlation=TRUE,symbolic.cor=TRUE))
```

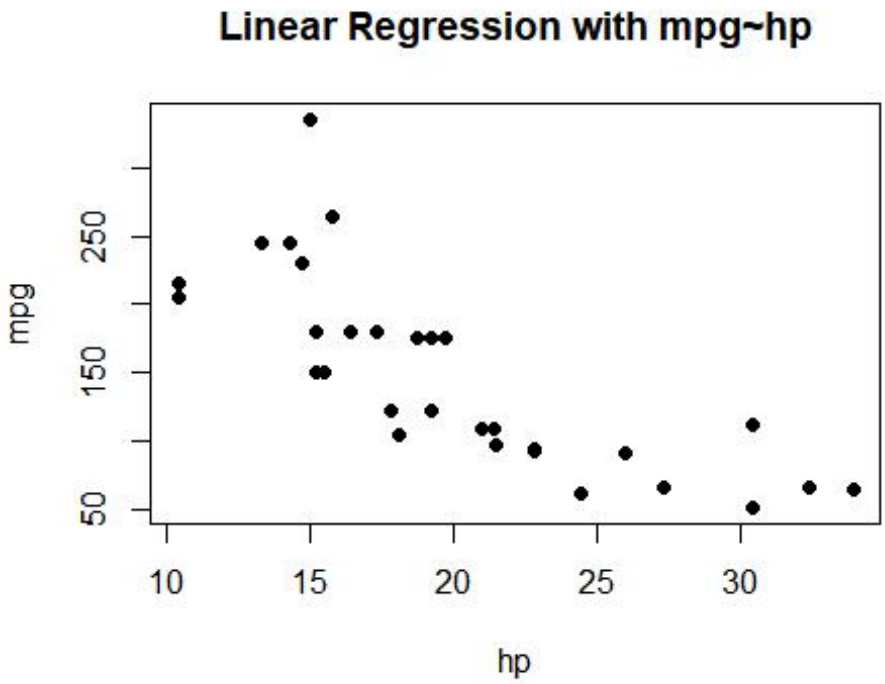
```
##
## Call:
## lm(formula = cars$mpg ~ cars$hp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7121 -2.1122 -0.8854  1.5819  8.2360
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.09886    1.63392  18.421  < 2e-16 ***
## cars$hp     -0.06823    0.01012  -6.742 1.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.863 on 30 degrees of freedom
## Multiple R-squared:  0.6024, Adjusted R-squared:  0.5892
## F-statistic: 45.46 on 1 and 30 DF,  p-value: 1.788e-07
##
## Correlation of Coefficients:
##
## (Intercept) 1
## cars$hp     * 1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

```
confint(relation)
```

```
##                    2.5 %     97.5 %
## (Intercept) 26.76194879 33.4357723
## cars$hp     -0.08889465 -0.0475619
```

```
#Intercept:30.09886
#slope: -0.06823
#Confidence interval:97.5%
```

```
plot(cars$mpg,cars$hp,xlab="hp",ylab="mpg",pch=16,main="Linear Regression wit
h mpg~hp")
```



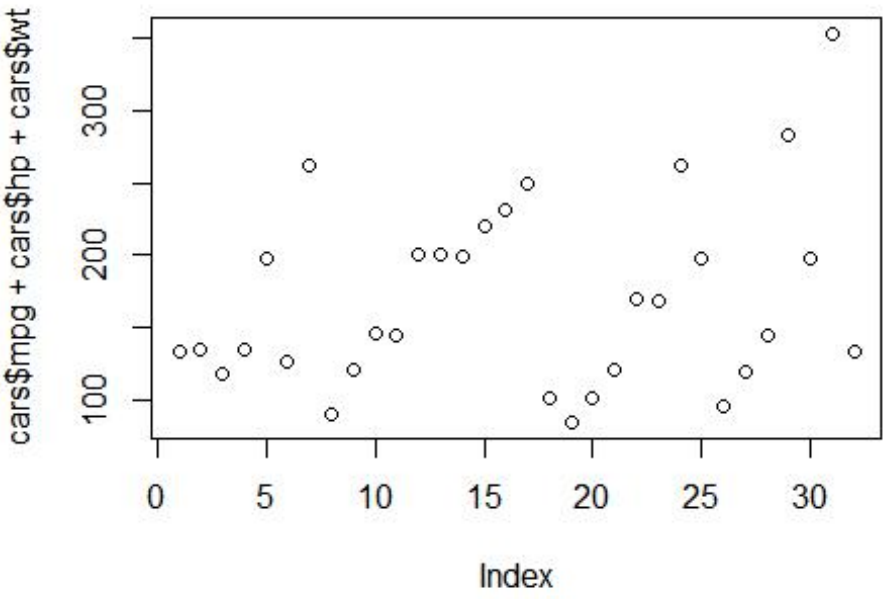

```
new=c(126,167,189,211,272,312)
new1=as.data.frame(new)
#Predict mileage
predict(relation,new1,interval="confidence")
```

```
## Warning: 'newdata' had 6 rows but variables found have 32 rows
```

```
##          fit       lwr      upr
## 1  22.593750 21.006344 24.18116
## 2  22.593750 21.006344 24.18116
## 3  23.753631 21.971485 25.53578
## 4  22.593750 21.006344 24.18116
## 5  18.158912 16.646511 19.67131
## 6  22.934891 21.295615 24.57417
## 7  13.382932 10.918575 15.84729
## 8  25.868707 23.630821 28.10659
## 9  23.617174 21.860464 25.37388
## 10 21.706782 20.228730 23.18483
## 11 21.706782 20.228730 23.18483
## 12 17.817770 16.262471 19.37307
## 13 17.817770 16.262471 19.37307
## 14 17.817770 16.262471 19.37307
## 15 16.112064 14.268892 17.95523
## 16 15.429781 13.445318 17.41424
## 17 14.406357 12.190623 16.62209
## 18 25.595794 23.421947 27.76964
## 19 26.550990 24.148024 28.95396
```

```
## 20 25.664022 23.474282 27.85376
## 21 23.480718 21.748829 25.21261
## 22 19.864619 18.468309 21.26093
## 23 19.864619 18.468309 21.26093
## 24 13.382932 10.918575 15.84729
## 25 18.158912 16.646511 19.67131
## 26 25.595794 23.421947 27.76964
## 27 23.890087 22.081920 25.69825
## 28 22.389065 20.830320 23.94781
## 29 12.086595  9.289663 14.88353
## 30 18.158912 16.646511 19.67131
## 31  7.242387  3.108308 11.37647
## 32 22.661978 21.064598 24.25936
```

```
#HP versus mpg,wt
plot(cars$mpg+cars$hp+cars$wt,type="p")
```




```
car.fit=lm(cars$mpg~cars$hp+cars$wt,data=cars)
print(car.fit)
```

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

```r
summary(car.fit,correlation=TRUE,symbolic.cor=TRUE)
```

```
##
## Call:
## lm(formula = cars$mpg ~ cars$hp + cars$wt, data = cars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## cars$hp     -0.03177    0.00903  -3.519  0.00145 **
## cars$wt     -3.87783    0.63273  -6.129 1.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
##
## Correlation of Coefficients:
##
## (Intercept) 1
## cars$hp       1
## cars$wt     , , 1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

```r
#Returns:
#The weighted residuals, the usual residuals rescaled by the square root of t
he weights specified in the call to lm.
#p x 4 matrix with columns for the estimated coefficient, its standard error,
 t-statistic and corresponding (two-sided) p-value.
#the square root of the estimated variance of the random error
#R^2, the 'fraction of variance explained by the model',
#R^2 = 1 - Sum(R[i]^2) / Sum((y[i]- y*)^2),
#where y* is the mean of y[i] if there is an intercept and zero otherwise.
#Other correlation coefficients
confint(car.fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 33.95738245 40.49715778
## cars$hp     -0.05024078 -0.01330512
## cars$wt     -5.17191604 -2.58374544
```

```r
#Intercept:37.22727
#cars$hp:-0.03177
#cars$wt:-3.87783
```

```
pred1=predict(car.fit)
pred1
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4
Drive
##           23.572329           22.583483           25.275819           21.2
65020
##   Hornet Sportabout             Valiant          Duster 360           Merc
 240D
##           18.327267           20.473816           15.599042           22.8
87067
##            Merc 230            Merc 280           Merc 280C          Merc
450SE
##           21.993673           19.979460           19.979460           15.7
25369
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Contin
ental
##           17.043831           16.849939           10.355205            9.3
62733
##   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Co
rolla
##            9.192487           26.599028           29.312380           28.0
46209
##       Toyota Corona    Dodge Challenger         AMC Javelin          Camar
o Z28
##           24.586441           18.811364           19.140979           14.5
52028
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus E
uropa
##           16.756745           27.626653           26.037374           27.7
69769
##      Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo
 142E
##           16.546489           20.925413           12.739477           22.9
83649
```

```
car.fit1=lm(cars$mpg~cars$hp+cars$wt+cars$hp*cars$wt)
print(car.fit1)
```

```
##
## Call:
## lm(formula = cars$mpg ~ cars$hp + cars$wt + cars$hp * cars$wt)
##
## Coefficients:
##     (Intercept)          cars$hp          cars$wt  cars$hp:cars$wt
##        49.80842         -0.12010         -8.21662          0.02785
```

```
confint(car.fit1)
```

```
##                         2.5 %      97.5 %
## (Intercept)      42.42359654 57.19325031
```

```
## cars$hp          -0.17069436 -0.06950982
## cars$wt         -10.81750352 -5.61574508
## cars$hp:cars$wt   0.01264983  0.04304647

#Intercept:49.8042
#Slope:-0.12010
#Returns:
#The weighted residuals, the usual residuals rescaled by the square root of t
he weights specified in the call to lm.
#p x 4 matrix with columns for the estimated coefficient, its standard error,
 t-statistic and corresponding (two-sided) p-value.
#the square root of the estimated variance of the random error
#R^2, the 'fraction of variance explained by the model',
#R^2 = 1 - Sum(R[i]^2) / Sum((y[i]- y*)^2),
#where y* is the mean of y[i] if there is an intercept and zero otherwise.
#Other correlation coefficients
summary(car.fit1,correlation=TRUE,symbolic.cor=TRUE)

##
## Call:
## lm(formula = cars$mpg ~ cars$hp + cars$wt + cars$hp * cars$wt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0632 -1.6491 -0.7362  1.4211  4.5513
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     49.80842    3.60516  13.816 5.01e-14 ***
## cars$hp         -0.12010    0.02470  -4.863 4.04e-05 ***
## cars$wt         -8.21662    1.26971  -6.471 5.20e-07 ***
## cars$hp:cars$wt  0.02785    0.00742   3.753 0.000811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.153 on 28 degrees of freedom
## Multiple R-squared:  0.8848, Adjusted R-squared:  0.8724
## F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13
##
## Correlation of Coefficients:
##
## (Intercept)     1
## cars$hp         + 1
## cars$wt         B , 1
## cars$hp:cars$wt * B * 1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1

pred2=predict(car.fit1)
pred2
```

```
##        1        2        3        4        5        6        7        8
## 23.09547 21.78138 25.58488 20.02924 17.28996 18.88542 15.40745 21.65887
##        9       10       11       12       13       14       15       16
## 20.84992 18.55379 18.55379 15.14994 16.23929 16.07909 12.02179 11.89490
##       17       18       19       20       21       22       23       24
## 12.50221 27.84866 32.63195 30.24587 24.56317 17.57441 17.91776 15.03111
##       25       26       27       28       29       30       31       32
## 15.93596 29.53900 26.71871 28.56630 15.36033 19.52990 13.54587 22.31363
```

```
hp1=c(126,167,189,211,272,312)
wt1=c(1.5,2.2,2.9,3.2,3.8,4.2)
new2=c(hp1,wt1)
new3=as.data.frame(new2)
predict(car.fit,new3,interval="prediction")
```

```
## Warning: 'newdata' had 12 rows but variables found have 32 rows
```

```
##          fit       lwr      upr
## 1  23.572329 18.152053 28.99261
## 2  22.583483 17.173052 27.99391
## 3  25.275819 19.814373 30.73726
## 4  21.265020 15.836451 26.69359
## 5  18.327267 12.926266 23.72827
## 6  20.473816 14.994498 25.95313
## 7  15.599042  9.993171 21.20491
## 8  22.887067 17.284608 28.48953
## 9  21.993673 16.532703 27.45464
## 10 19.979460 14.552353 25.40657
## 11 19.979460 14.552353 25.40657
## 12 15.725369 10.274249 21.17649
## 13 17.043831 11.631452 22.45621
## 14 16.849939 11.434088 22.26579
## 15 10.355205  4.579380 16.13103
## 16  9.362733  3.530661 15.19480
## 17  9.192487  3.414919 14.97006
## 18 26.599028 21.087052 32.11100
## 19 29.312380 23.691616 34.93314
## 20 28.046209 22.484454 33.60796
## 21 24.586441 19.143493 30.02939
## 22 18.811364 13.413350 24.20938
## 23 19.140979 13.749010 24.53295
## 24 14.552028  8.981473 20.12258
## 25 16.756745 11.335804 22.17769
## 26 27.626653 22.083269 33.17004
## 27 26.037374 20.548743 31.52601
## 28 27.769769 22.072733 33.46681
## 29 16.546489 10.725413 22.36756
## 30 20.925413 15.446364 26.40446
## 31 12.739477  6.476550 19.00240
## 32 22.983649 17.570829 28.39647
```

```r
predict(car.fit,new3,interval="confidence")
```

```
## Warning: 'newdata' had 12 rows but variables found have 32 rows
```

```
##          fit       lwr      upr
## 1  23.572329 22.456232 24.68843
## 2  22.583483 21.516224 23.65074
## 3  25.275819 23.974405 26.57723
## 4  21.265020 20.109318 22.42072
## 5  18.327267 17.308891 19.34564
## 6  20.473816 19.099332 21.84830
## 7  15.599042 13.784634 17.41345
## 8  22.887067 21.083228 24.69091
## 9  21.993673 20.694257 23.29309
## 10 19.979460 18.830645 21.12827
## 11 19.979460 18.830645 21.12827
## 12 15.725369 14.467994 16.98274
## 13 17.043831 15.966740 18.12092
## 14 16.849939 15.755529 17.94435
## 15 10.355205  8.069065 12.64134
## 16  9.362733  6.937998 11.78747
## 17  9.192487  6.901946 11.48303
## 18 26.599028 25.099641 28.09842
## 19 29.312380 27.452469 31.17229
## 20 28.046209 26.373069 29.71935
## 21 24.586441 23.364981 25.80790
## 22 18.811364 17.808947 19.81378
## 23 19.140979 18.171643 20.11032
## 24 14.552028 12.849864 16.25419
## 25 16.756745 15.637422 17.87607
## 26 27.626653 26.015632 29.23767
## 27 26.037374 24.626220 27.44853
## 28 27.769769 25.690706 29.84883
## 29 16.546489 14.148322 18.94466
## 30 20.925413 19.551999 22.29883
## 31 12.739477  9.409231 16.06972
## 32 22.983649 21.904339 24.06296
```

```r
predict(car.fit1,new3,interval="prediction")
```

```
## Warning: 'newdata' had 12 rows but variables found have 32 rows
```

```
##         fit       lwr      upr
## 1  23.09547 18.581687 27.60926
## 2  21.78138 17.262031 26.30072
## 3  25.58488 21.041239 30.12852
## 4  20.02924 15.465956 24.59253
## 5  17.28996 12.764155 21.81576
## 6  18.88542 14.248302 23.52253
## 7  15.40745 10.745695 20.06920
## 8  21.65887 16.953143 26.36459
```

```
## 9  20.84992 16.267096 25.43274
## 10 18.55379 13.975233 23.13234
## 11 18.55379 13.975233 23.13234
## 12 15.14994 10.607150 19.69273
## 13 16.23929 11.718200 20.76037
## 14 16.07909 11.556876 20.60130
## 15 12.02179  7.134532 16.90904
## 16 11.89490  6.853171 16.93663
## 17 12.50221  7.370479 17.63393
## 18 27.84866 23.215676 32.48165
## 19 32.63195 27.620095 37.64381
## 20 30.24587 25.468673 35.02307
## 21 24.56317 20.038028 29.08832
## 22 17.57441 13.036154 22.11268
## 23 17.91776 13.385583 22.44994
## 24 15.03111 10.392521 19.66970
## 25 15.93596 11.406922 20.46500
## 26 29.53900 24.813678 34.26433
## 27 26.71871 22.140473 31.29694
## 28 28.56630 23.810027 33.32258
## 29 15.36033 10.477733 20.24293
## 30 19.52990 14.911525 24.14827
## 31 13.54587  8.320464 18.77127
## 32 22.31363 17.798713 26.82854
```

```
predict(car.fit1,new3,interval="confidence")
```

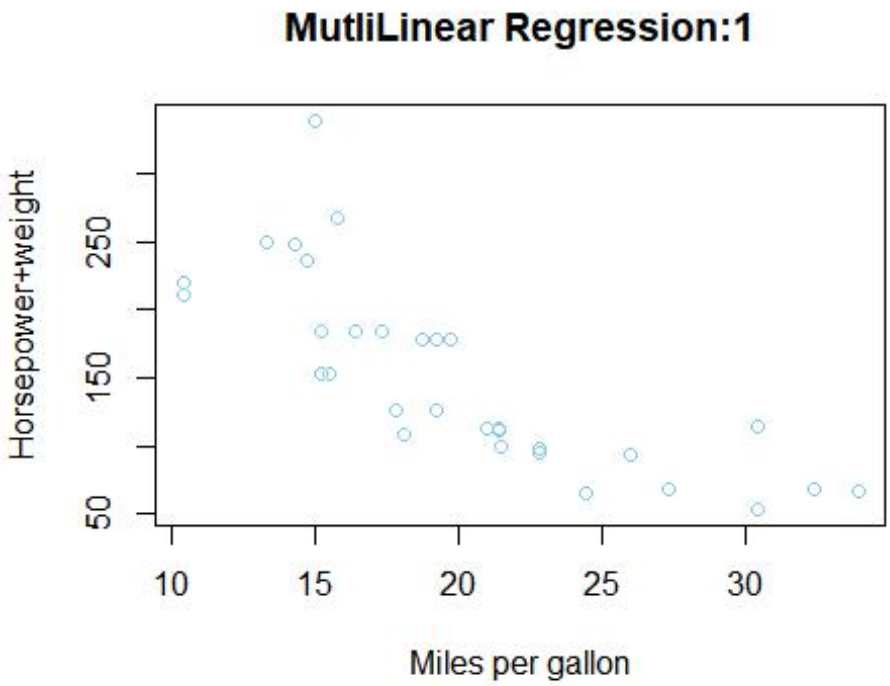
```
## Warning: 'newdata' had 12 rows but variables found have 32 rows
```

```
##         fit       lwr      upr
## 1  23.09547 22.131774 24.05917
## 2  21.78138 20.791974 22.77078
## 3  25.58488 24.489847 26.67991
## 4  20.02924 18.855346 21.20314
## 5  17.28996 16.271471 18.30844
## 6  18.88542 17.451097 20.31974
## 7  15.40745 13.895373 16.91952
## 8  21.65887 20.016220 23.30152
## 9  20.84992 19.602242 22.09759
## 10 18.55379 17.321878 19.78570
## 11 18.55379 17.321878 19.78570
## 12 15.14994 14.058437 16.24144
## 13 16.23929 15.241961 17.23661
## 14 16.07909 15.076670 17.08150
## 15 12.02179  9.914727 14.12885
## 16 11.89490  9.450830 14.33897
## 17 12.50221  9.877502 15.12691
## 18 27.84866 26.427743 29.26958
## 19 32.63195 30.250111 35.01380
## 20 30.24587 28.408467 32.08327
```

```
## 21 24.56317 23.547603 25.57874
## 22 17.57441 16.501916 18.64691
## 23 17.91776 16.871290 18.96424
## 24 15.03111 13.592025 16.47020
## 25 15.93596 14.903181 16.96874
## 26 29.53900 27.841019 31.23699
## 27 26.71871 25.487991 27.94942
## 28 28.56630 26.783998 30.34861
## 29 15.36033 13.264095 17.45657
## 30 19.52990 18.157379 20.90242
## 31 13.54587 10.742427 16.34931
## 32 22.31363 21.344667 23.28258
```

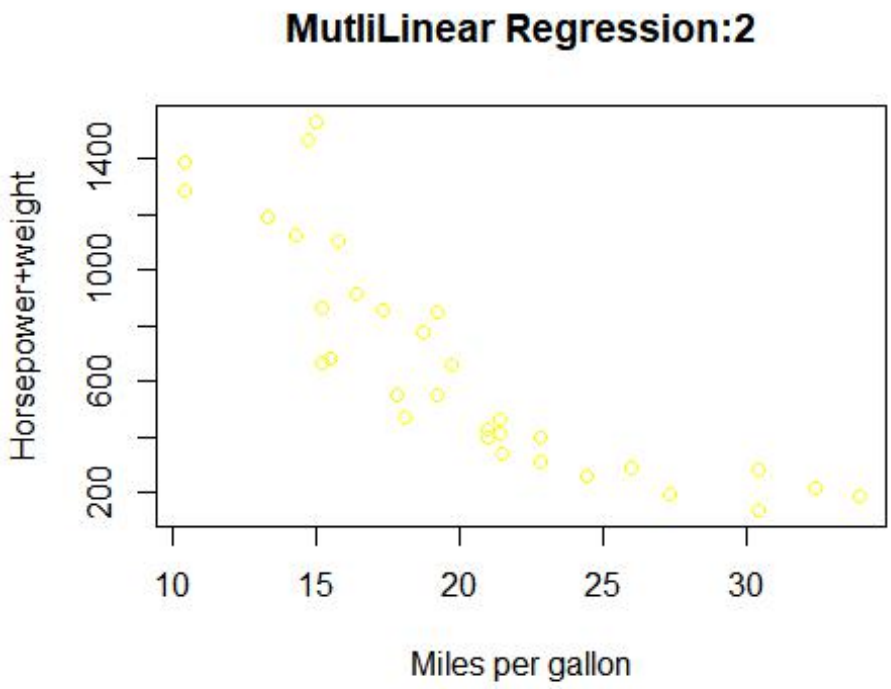
```
plot(cars$mpg,cars$hp+cars$wt,type="p",col="skyblue",xlab="Miles per gallon",
ylab="Horsepower+weight"
     ,main="MutliLinear Regression:1")
abline(car.fit,col="skyblue")
```

```
## Warning in abline(car.fit, col = "skyblue"): only using the first two of 3
## regression coefficients
```




```
plot(cars$mpg,cars$hp+cars$wt+cars$hp*cars$wt,type="p",col="yellow",xlab="Mil
es per gallon",ylab="Horsepower+weight"
     ,main="MutliLinear Regression:2")
abline(car.fit1,col="Red")
```

```
## Warning in abline(car.fit1, col = "Red"): only using the first two of 4
## regression coefficients
```




```
#Multilinear model 1 performs better than 2 as confidence interval is shorter
#Both models,due to low pvalues,need improvement in terms of parameters consi
dered.
```