Test your knowledge of linear regression in R

1)

a) auto ← read.csv("Auto.csv")

Str(auto)

Note that the horsepower variable is read in as a factor variable due to the presence of ?.

You need to convert this to numeric as follows.

auto$horsepower ← as.numeric(as.character(auto$horsepower))

This would lead to a reasonable model.

b) model1 ← lm(mpg ~ horsepower, data = auto)

Summary(model1)

- Yes there is a strong relationship between the predictor & response. p-value is almost zero
- Implying we can reject the null hypothesis that the corresponding $\beta = 0$.
- Since $\beta = -0.1578$ for the relation between mpg & horsepower, the relation is negative.

c) ?predict.lm

predict(model1, newdata = data.frame(horsepower = 98), interval = c("confidence"), level = 0.99)

The predicted mpg for horsepower of 98 is 24.46708 & the 99% confidence interval is [23.816, 25.117]

d) cor(auto$mpg, auto$horsepower,
use = "pairwise.complete.obs")

This helps compute correlation while dropping observations where even one entry is missing.

The correlation = -0.7784

Squaring the correlation gives the R^2 of the model

e) plot(auto$horsepower, auto$mpg)

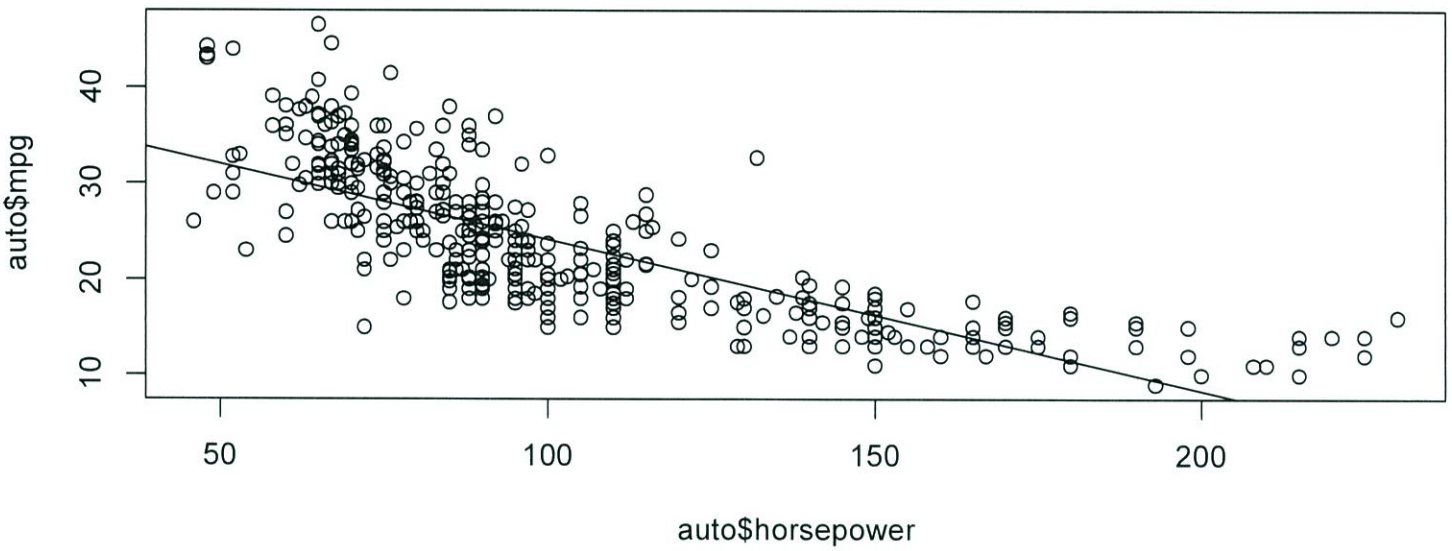
abline(model1)

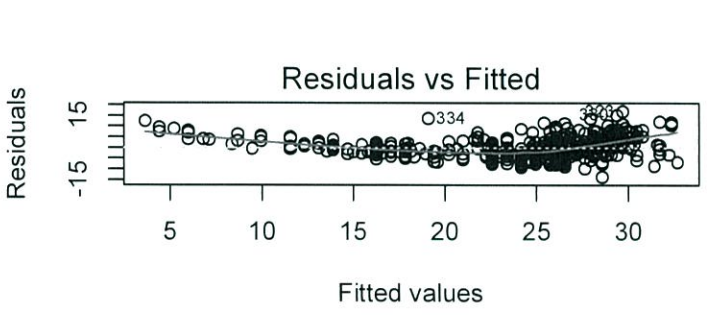
f) layout(matrix(1:4, 2, 2))

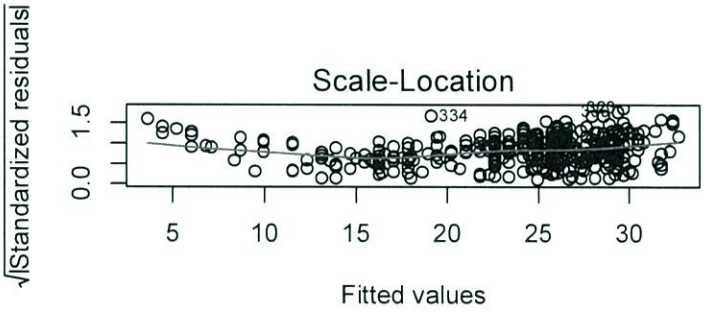
plot(model1)

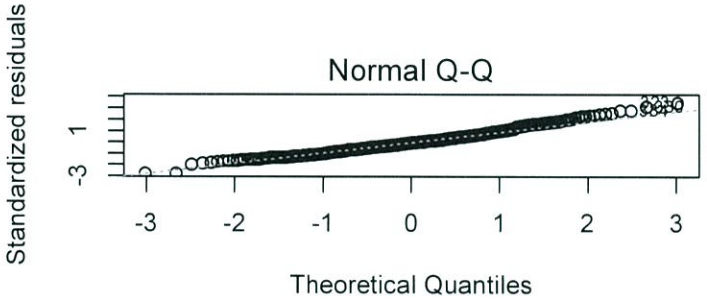
A good linear fit should see residuals randomly scattered. In this model we see that the residuals decrease & then increase as the number of fitted residuals increase. The normal Q-Q plot also shows that the distribution of the residuals is not normal at the extreme values.

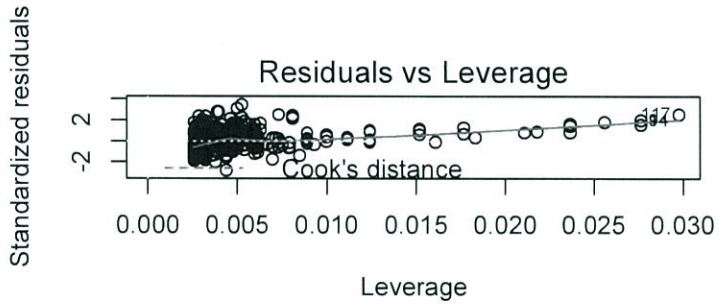
This indicates that the data might have evidence of some nonlinearities, and outliers.


auto$mpg
10
20
30
40
50
100
150
200
auto$horsepower


Residuals vs Fitted
Residuals
Fitted values
334


Scale-Location
√|Standardized residuals|
Fitted values
334


Normal Q-Q
Standardized residuals
Theoretical Quantiles


Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

2)

a) pairs(auto)

This creates a scatterplot matrix from the auto dataframe in question 1).

b) str(auto)
auto1 ← subset(auto, select = -c(name))

cor(auto1)

This drops the name variable from auto which is qualitative. Note that by default this uses every observation and hence use see NA for the correlations with horsepower which has missing entries.

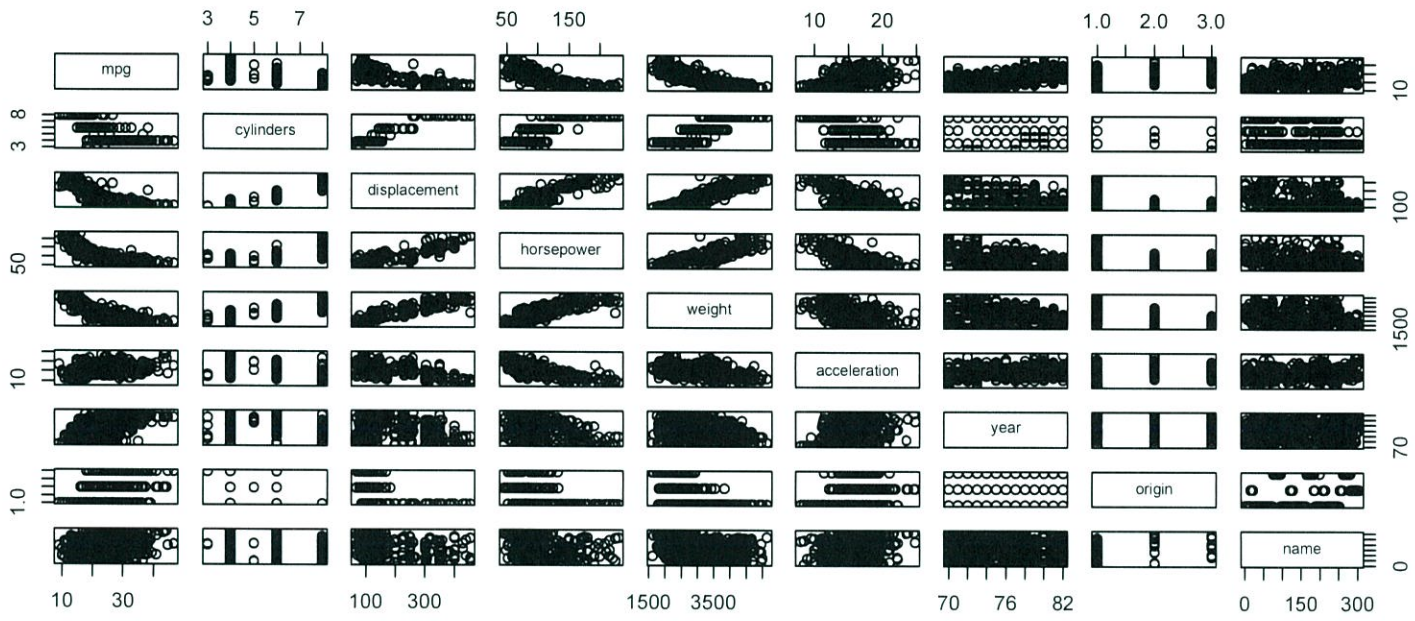
c) model2 ← lm(mpg ~ . , data = auto1)

Summary(model2)

- The p-value is very close to 0 for the multiple regression model. We can reject the null hypothesis that the β_i s are all zero.

- The variables displacement, weight, year & origin at the 0.01 (0.1% level)

- The coefficient for year is positive & p-value is close to 0. This shows model year is positively related to mpg where every year adds 0.7507 to miles per gallon everything else staying same.


mpg
cylinders
displacement
horsepower
weight
acceleration
year
origin
name
3 5 7
50 150
10 20
1.0 2.0 3.0
3 8
50
10
1.0
10 30
100 300
1500 3500
70 76 82
0 150 300
10
100
1500
70
0

3)

a) set.seed(1)

x1 ← runif(100)

x2 ← 0.5 x1 + rnorm(100)/10

Y ← 2 + 2x1 + 0.3 x2 + rnorm(100)

The form of the linear model is:

$$Y = 2 + 2X_1 + 0.3\,X_2 + \epsilon$$

where coefficients are $\beta_0 = 2$, $\beta_1 = 2$, $\beta_2 = 0.3$.

b) cor(x1, x2) = 0.8351

plot(x1, x2)

This clearly shows a strong positive correlation between x_1, x_2 as expected from the simulation

c) model3 ← lm(y ~ x1 + x2)

summary(model3)

- From the fit we see $\hat{\beta}_1 = 1.43$, $\hat{\beta}_2 = 1$, $\hat{\beta}_0 = 2.13$.

The true values are $\beta_1 = 2$, $\beta_2 = 0.3$, $\beta_0 = 2$.

There is discrepancy in β_1, β_2 & $\hat{\beta}_1, \hat{\beta}_2$. less so in β_0 & $\hat{\beta}_0$

- We can reject the null hypothesis that $\beta_1 = 0$ at the 5% level
- We cannot reject the null hypothesis that $\beta_2 = 0$ at the 5% level

d) model 4 ← lm(Y ~ X1)

Summary(model 4)

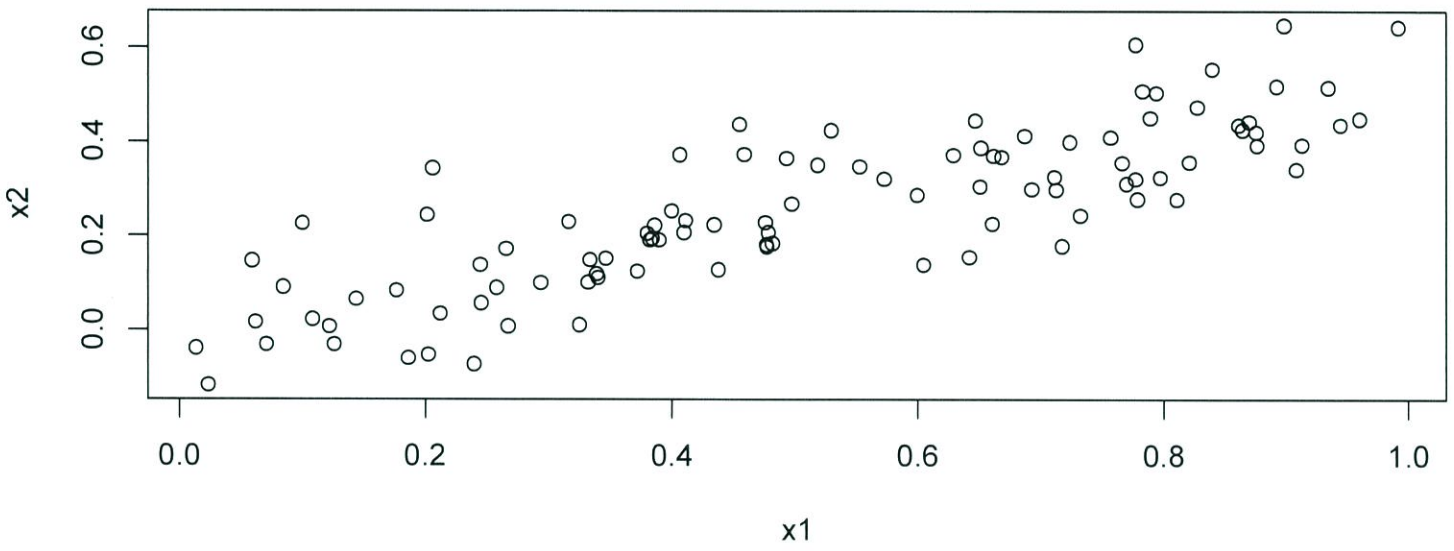
The estimated $\hat{\beta}_1 = 1.975$ is close to $\beta_1 = 2$ & we can reject the $H_0: \beta_1 = 0$ as the p-value is close to 0.

e) model 5 ← lm(Y ~ X2)

Summary(model 5)

We can reject the $H_0: \beta_2 = 0$ as p-value is very close to 0. The value $\beta_2 = 2.89$ is fairly different from 0.3

f) There is multicollinearity in the data between X1 & X2. In doing multiple regression we see this effect where one of the coefficients is possibly hard to reject $H_0: \beta_i = 0$ while we see that with a single regression with one variable, we can reject $H_0: \beta_i = 0$. This is caused due to multicollinearity there.


x2
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0
x1

4)

a) boston <- read.csv("Boston.csv")

colnames(boston)

This provides names of all columns of the dataframe

model1 <- lm(medv ~ crim, data = boston)

model2 <- lm(medv ~ zn, data = boston)

⋮

model13 <- lm(medv ~ stat, data = boston)

Use summary(model1) to summary(model13)

to verify that the p-values for all the predictors is less than 0.001

plot(boston$lstat, boston$medv)

abline(model13)

This figure helps validate this result visually.

b) modelall <- lm(medv ~ ., data = boston)

Summary(modelall)

The adjusted R^2 is 0.7338 which is larger than the adjusted R^2 from the simple regression models.

The variables for which we can reject the $H_0: \beta_j = 0$ are crim, zn, chas, nox, rm, dis, rad, tax, ptratio, black, lstat at the 0.05 level.

c)

```
X ← c(model1$coef[2], model2$coef[2], ...,
      model13$coef[2])
Y ← modelall$coef[2:14]
plot(X, Y, main = "Coefficient relationship",
     xlab = "Simple linear regression",
     ylab = "multiple linear regression")
```

The figure seems to indicate a fairly positive relationship between the results from the simple & multiple linear regression models. The relationship seems to be linear too.

d) ? poly

```
modelpoly2 ← lm(medv ~ poly(lstat, 2, raw=TRUE), data = boston)
summary(modelpoly2)
modelpoly3 ← lm(medv ~ poly(lstat, 3, raw=TRUE), data = boston)
summary(modelpoly3)
```

⋮

```
modelpoly6 ← lm(medv ~ poly(lstat, 6, raw=TRUE), data = boston
summary(modelpoly6)
```

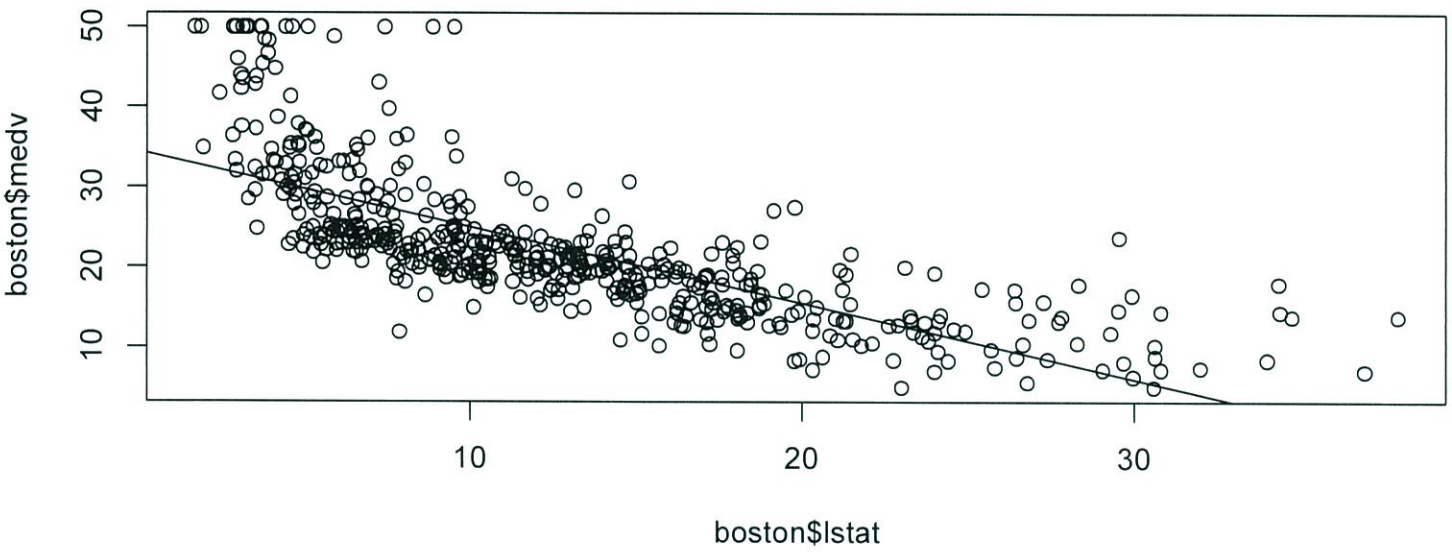
Yes adding higher degree terms helps improve the fit.

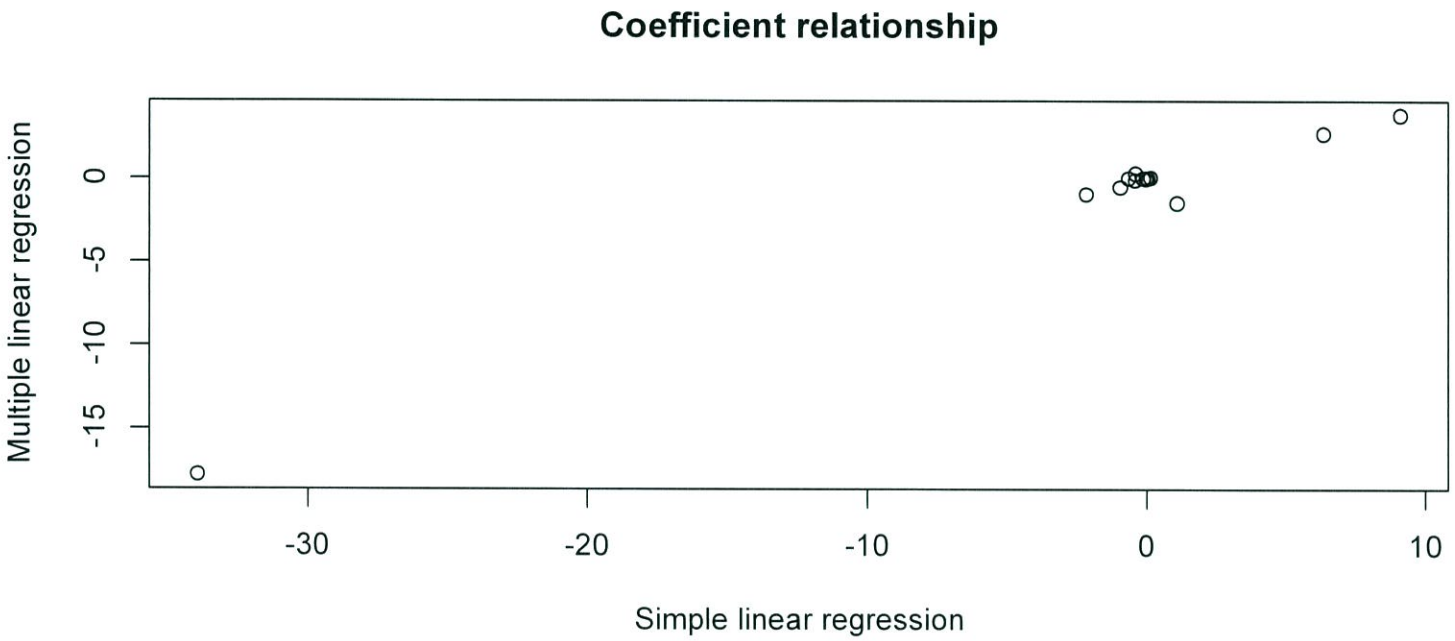
Beyond degree 5, adding additional terms does not keeps the terms significant

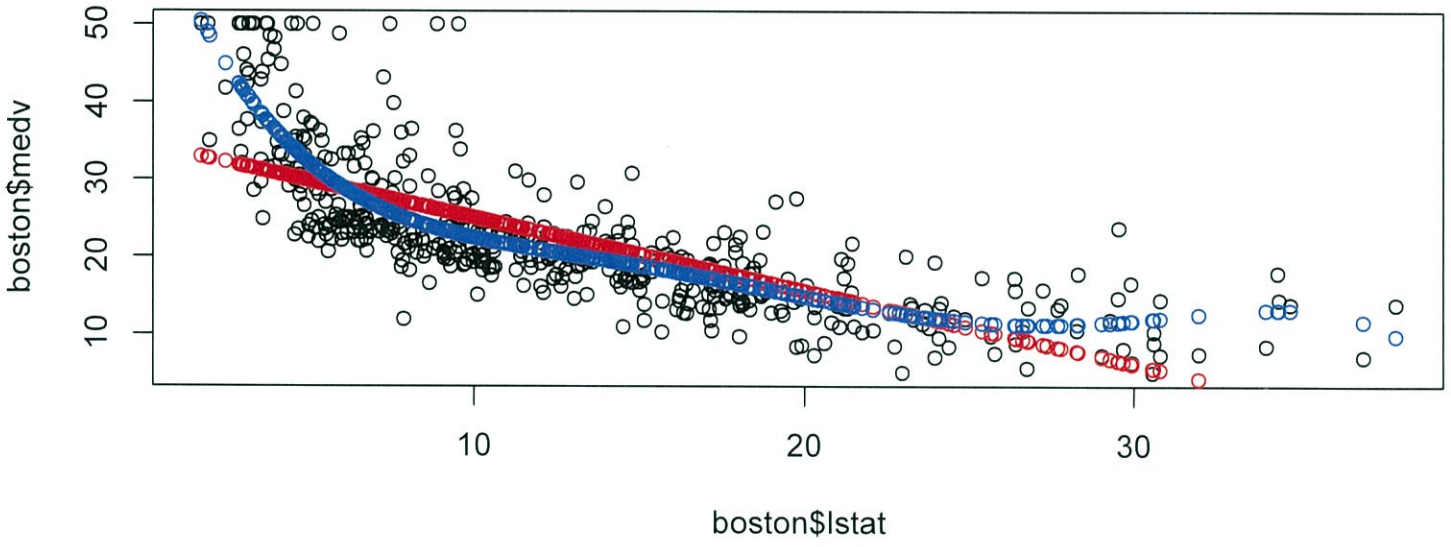
We can visualize this as follows:

```
pr1 <- predict(model13, newdata = Boston)

pr5 <- predict(modelpoly5, newdata = Boston)

plot(Boston$lstat, Boston$medv)

points(Boston$lstat, pr1, col = "red")

points(Boston$lstat, pr5, col = "blue")
```

The points command adds points to a plot.


boston$medv
50
40
30
20
10
10
20
30
boston$lstat


Coefficient relationship
Multiple linear regression
0
-5
-10
-15
-30
-20
-10
0
10
Simple linear regression


boston$medv
50
40
30
20
10
10
20
30
boston$lstat

5)

a) climate <- read.csv("climate_change.csv")

training <- subset(climate, Year <= 2006)

test <- subset(climate, Year > 2006)

model1 <- lm(Temp ~ MEI + CO2 + CH4 + N2O + CFC.11 + CFC.12 + TSI + Aerosols, data = training)

Summary(model1)

R^2 for this model is 0.7509.

b) The variables significant in this model with p-values below 0.05 are: (*** in summary)

MEI, CO2, CFC.11, CFC.12, TSI, Aerosols

c) cor(training)

N2O is highly correlated with CO2, CH4, CFC.12, Temp & quite correlated with CFC.11.

CFC.11 is fairly positively correlated with CO2, N2O & strongly correlated with CH4, CFC.12.

The results seem to indicate that all the gas concentration variables reflect human development & the variables are correlated in the dataset.

Note that the correlation between Temp & N2O is 0.728 & Temp & CFC.11 is 0.40 (fairly high)

d) model 2 ← lm (Temp ~ MEI + TSI + Aerosols + N2O, data = training)

Summary (model 2)

The coefficient for N2O in this reduced model is 0.0253. The variable is also very significant as compared to the model with all variables in it. By comparing the R^2 & adjusted R^2, we also see the model does not lose a lot of explanatory power while variables are reduced. This is typical of models where the independent variables are highly correlated with each other.

e) ? step

model 3 ← step (model 1)

Summary (model 3)

R^2 is 0.7508 which is slightly lower than full model but adjusted R^2 is 0.7445 & higher. CH4 is eliminated in this model to get a better fit but with fewer predictors.

f) pred ← predict(model3, newdata = test)

SSE ← sum((test$Temp - pred)^2)

SST ← sum((test$Temp - mean(training$Temp))^2)

testR2 ← 1 - SSE/SST

The test R2 value is 0.6286051

Note that with the full model, you get

test R^2 = 0.6274

6)

a) wine ← read.csv("wine data.csv")

str(wine)

wine$age91 ← 1991 - wine$vintage

wine$age92 ← 1992 - wine$vintage

mean(subset(wine$price91, wine$age91 >= 15))

The average price of wine that were 15 years or older at the 1991 auction = 96.435 $

b) mean(subset(wine$price91, wine$harvrain < mean(wine$harvrain) & wine$tempdiff < mean(wine$tempdiff)))

The average price in 1991 when harvest rain & temperature difference was below average = 72.867 $

c) train ← subset(wine, vintage ≤ 1981)

model1 ← lm(log(price91) ~ age91, data = train)

summary(model1)

R^2 for this model = 0.6675

d) confint(model1, level = 0.99)

β_0 (for intercept) [3.159, 3.98]

β_1 (for age91) [0.022, 0.062]

e) test ← subset(wine, vintage ≥ 1982)

predtest ← predict(model1, newdata = test)

SSE ← sum((log(test$price91) − predtest)^2)

SST ← sum((log(test$price91) − mean(log(train$price91)))^2)

testR2 ← 1 − SSE/SST

Test R^2 = 0.9213742

f) In comparison to the results for the Bordeaux wine data, the traning (model) R^2 and test R^2 is higher for this new dataset. This seems to indicate that the variation in the prices of the wine in the data set is explained much more by the age of the wines in comparison to the Bordeaux dataset.

g) model2 ← lm(log(price91) ~ temp + harvrain + winrain + tempdiff + age91, data = train)

Summary(model2)

R^2 for this model = 0.7938

h) With only the age variable, adjusted $R^2 = 0.65$

With all the variables, adjusted $R^2 = 0.7145$

This seems to indicate that the latter model with more variables included is preferred.

i) The result indicates that lesser the harvest rain, the better is the price & quality of the wine. This is because the corresponding $\beta = -0.002$ & is significant at the 0.1 level.

All other statements appear to be false.

j) The least significant variables are wrain & tempdiff with pvalues 0.53 & 0.416 respectively

```
model3 <- lm(log(Price91) ~ temp + age91 +
              hrain, data = train)

summary(model3)
```

$$\log(\text{Price91}) = 1.80 + 0.0975\,\text{temp} + 0.0456\,\text{age91} - 0.0019\,\text{hrain}$$

k) In the training set, adjusted R^2 for this model is 0.73 while for model 2, adjusted $R^2 = 0.7145$

In this case, the new model 3 is preferred to model 2.

l) model 4 ← lm(log(price92) ~ temp + hrain + age92, data = train)

Summary (model 4)

R^2 for this model is 0.5834

m) The p-value for hrain = 0.32. Hence we cannot reject the null hypothesis that the β coefficient for hrain = 0.

n) The best explanation seems to be that we can check for consistency of the effect of weather variables & age by looking at the sign of the estimated coefficients.

o) Clearly dropping missing entries is reliable. However if there are many missing entries then, this implies we can loose a lot of data.

7)

a) batters ← read.csv("batters.csv")

str(batters)

which.max(batters$salary)

This indicates the batter at row 1159 made the maximum salary.

batters[1159,]

This batter's name is coded as giambja01.

b) max(batters$salary[batters$yearID == 2006]) / min(batters$salary[batters$yearID == 2006])

The number is 61.65. This indicates the top paid player made 61 times what the least paid player made.

c) sort(tapply(batters$salary[batters$yearID == 1996]

d) batters$teamID[batters$yearID == 1996], sum))

This indicates that OAK (Oakland Athletics) had the smallest payroll while NYA (New York Yankees) had the highest payroll.

e) hist(batters$salary)

Most of the salaries are small with a relatively small number of much large salaries (right-skewed).

f) model1 ← lm(log(salary) ~ R, data = batters)

summary(model1)

The model fit gives:

$$\log(\text{Salary}) = 13.41 + 0.0149\,R$$

When $R = 0$, $\log(\text{Salary}) = 13.41$

g) mean(batters$salary[batters$R == 0], na.rm = TRUE)

Actual average in the data set with $R = 0$ for the logarithm of salary is 13.60

h) The values are close. Note that if you only regress log(salary) with a constant, the best fit would be the mean. This is the reason for this result.

i) $\log(\text{salary}) = \beta_0 + \beta_1 R$

$\therefore \quad \text{Salary} = e^{\beta_0 + \beta_1 R}$

If runs increases by 1, we have New salary = Old salary e^{β}

j)

batters$OBP ← (batters$H + batters$BB + batters$HBP) /
(batters$AB + batters$BB + batters$HBP
+ batters$SF)

batters$SLG ← (batters$H + batters$X2B + 2 batters$X3B
+3batters$HR) / (batters$AB)

mean(batters$OBP[, na.rm = TRUE)
([batters$yearID] == 2006)

The average on base percentage 0.2707
in the 2006 season was:

k) t.test(batters$SLG[batters$yearID == 1996],
batters$SLG[batters$yearID == 2006])

p-value of test = 0.4045.

This seems to indicate that there is not enough evidence to reject the null hypothesis that the average slugging percentage of the 1996 & 2006 seasons are the same.

l) model2 ← lm(log(salary) ~ OBP + SLG,
data = subset(batters, yearID == 1996 & batters$AB >= 13
Summary(model2)

Adjusted R^2 = 0.2589

m) Since all the p-values are very close to 0, there is enough evidence to indicate that we can reject each of the null hypothesis that

$H_0 : \beta_j = 0$.

n) model 3 <- lm(log(salary) ~ OBP + SLG, data = subset(batters, yearID == 2006 & batters$AB >= 130))

Summary(model 3)

Adjusted $R^2 = 0.1164$

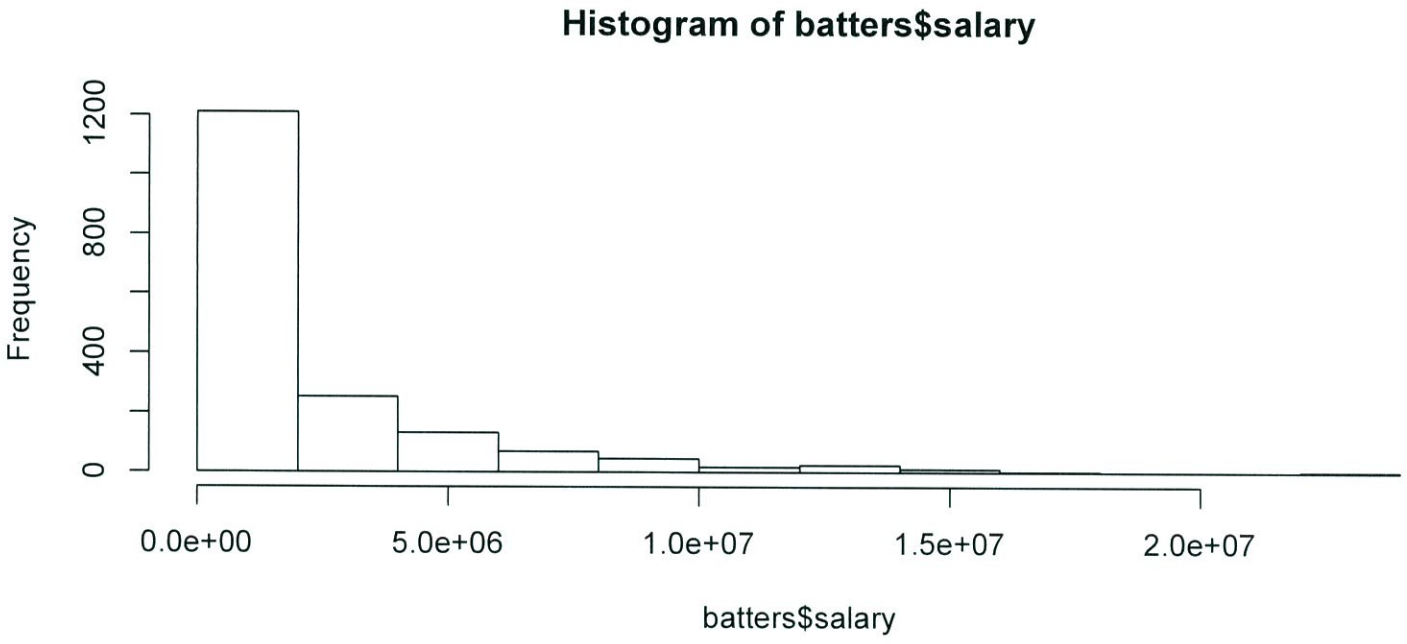
o) In model 2, (1996) $\beta_{OBP} = 4.87$, $\beta_{SLG} = 5.46$

Both are significant in a statistical sense

In model 3, (2006) $\beta_{OBP} = 6.64$, $\beta_{SLG} = 2.90$

Both are statistically significant.

The market undervalued OBP compared to SLG in 1996 ($4.87 < 5.46$) before Moneyball was published. This has been corrected since in 2006 ($6.64 > 2.90$)


Histogram of batters$salary
Frequency
0
400
800
1200
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
batters$salary

```
#1a)
auto <- read.csv("Auto.csv")
str(auto)
auto$horsepower <- as.numeric(as.character(auto$horsepower))

#1b)
model1 <- lm(mpg~horsepower, data = auto)
summary(model1)

#1c)
?predict.lm
predict(model1,newdata = data.frame(horsepower = 98), interval = c("confidence"),
level = .99)

#1d)
cor(auto$mpg, auto$horsepower, use = "pairwise.complete.obs")
cor(auto$mpg, auto$horsepower, use = "pairwise.complete.obs") ^2

#1e)
plot(auto$horsepower, auto$mpg)
abline(model1)

#1f)
layout(matrix(1:4,2,2))
plot(model1)

#2a)
pairs(auto)

#2b)
auto1 <- subset(auto, select = -c(name))
cor(auto1)

#2c)
model2 <- lm(mpg~., data = auto1)
summary(model2)

#3a)
set.seed(1)
x1 <- runif(100)
x2 <- .5*x1 + rnorm(100)/10
y <- 2 + 2*x1 + .3*x2 + rnorm(100)

#3b)
cor(x1,x2)
plot(x1, x2)

#3c)
model3 <- lm(y~x1+x2)
summary(model3)

#3d)
model4 <- lm(y~x1)
summary(model4)

#3e)
model5 <- lm(y~x2)
summary(model5)

#4a)
boston <- read.csv("Boston.csv")
colnames(boston)
model1 <- lm(medv~crim, data=boston)
model2 <- lm(medv~zn, data=boston)
model3 <- lm(medv~indus, data=boston)
model4 <- lm(medv~chas, data=boston)
model5 <- lm(medv~nox, data=boston)
model6 <- lm(medv~rm, data=boston)
```

```
model7 <- lm(medv~age, data=boston)
model8 <- lm(medv~dis, data=boston)
model9 <- lm(medv~rad, data=boston)
model10 <- lm(medv~tax, data=boston)
model11 <- lm(medv~ptratio, data=boston)
model12 <- lm(medv~black, data=boston)
model13 <- lm(medv~lstat, data=boston)
summary(model13)
plot(boston$lstat, boston$medv)
abline(model13)

#4b)
modelall<- lm(medv~., data=boston)
summary(modelall)

#4c)
x <- c(model1$coef[2], model2$coef[2], model3$coef[2], model4$coef[2],
model5$coef[2], model6$coef[2], model7$coef[2], model8$coef[2], model9$coef[2],
model10$coef[2], model11$coef[2], model12$coef[2], model13$coef[2])
y <- modelall$coef[2:14]
plot(x, y, main = "Coefficient relationship", xlab = "Simple linear regression",
ylab = "Multiple linear regression")

#4d)
modelpoly2 <- lm(medv~poly(lstat,2,raw=TRUE), data = boston)
summary(modelpoly2)
modelpoly3 <- lm(medv~poly(lstat,3,raw=TRUE), data = boston)
summary(modelpoly3)
modelpoly4 <- lm(medv~poly(lstat,4,raw=TRUE), data = boston)
summary(modelpoly4)
modelpoly5 <- lm(medv~poly(lstat,5,raw=TRUE), data = boston)
summary(modelpoly5)
modelpoly6 <- lm(medv~poly(lstat,6,raw=TRUE), data = boston)
summary(modelpoly6)

pr1 <- predict(model13,newdata=boston)
pr5 <- predict(modelpoly5,newdata=boston)
plot(boston$lstat,boston$medv)
points(boston$lstat,pr1,col="red")
points(boston$lstat,pr5,col="blue")

#5a)
climate <- read.csv("climate_change.csv")
training <- subset(climate, climate$Year <= 2006)
test <- subset(climate, Year > 2006)
model1 <- lm(Temp~MEI+CO2+CH4+N2O+CFC.11+CFC.12+TSI+Aerosols, data = training)
summary(model1)

#5c)
cor(training))

#5d)
model2 <- lm(Temp~MEI+TSI+Aerosols+N2O, data = training)
summary(model2)

#5e)
model3 <- step(model1)
summary(model3)

#5f)
pred <- predict(model3, newdata = test)
sse <- sum((test$Temp-pred)^2)
sst <- sum((test$Temp-mean(training$Temp))^2)
testR2 <- 1-sse/sst
testR2

#6a)
wine<-read.csv("winedata.csv")
```

```
str(wine)
wine$age91<-1991-wine$vintage
wine$age92<-1992-wine$vintage
mean(subset(wine$price91,wine$age91>=15))

#6b)
mean(subset(wine$price91,wine$hrain<mean(wine$hrain)&wine$tempdiff<mean(wine$temp
diff)))

#6c)
train<-subset(wine,vintage<=1981)
model1<-lm(log(price91)~age91,data=train)
summary(model1)

#6d)
confint(model1,  level = 0.99)

#6e)
test<-subset(wine,vintage>=1982)
predtest<-predict(model1,newdata=test)
sse<-sum((log(test$price91)-predtest)^2)
sst<-sum((log(test$price91)-mean(log(train$price91)))^2)
testR2<- 1-sse/sst
testR2

#6g)
model2<-lm(log(price91)~temp+hrain+wrain+tempdiff+age91,data=train)
summary(model2)

#6j)
model3<-lm(log(price91)~temp+hrain+age91,data=train)
summary(model3)

#6l)
model4<-lm(log(price92)~temp+hrain+age92,data=train)
summary(model4)

#7a)
batters <- read.csv("batters.csv")
str(batters)
which.max(batters$salary)
batters[1159,]

#7b)
max(batters$salary[batters$yearID==2006])/min(batters$salary[batters$yearID==2006
])

#7cd)
sort(tapply(batters$salary[batters$yearID==1996],batters$teamID[batters$yearID==1
996],sum))

#7f)
model1 <- lm(log(salary)~R,data=batters)
summary(model1)

#7g)
mean(log(batters$salary[batters$R==0]),na.rm=TRUE)

#7j)
batters$OBP <-
(batters$H+batters$BB+batters$HBP)/(batters$AB+batters$BB+batters$HBP+batters$SF)
batters$SLG <- (batters$H+batters$X2B+2*batters$X3B+3*batters$HR)/(batters$AB)
mean(batters$OBP[batters$yearID==2006],na.rm=TRUE)

#7k)
t.test(batters$SLG[batters$yearID==1996],batters$SLG[batters$yearID==2006])

#7l)
```

```
model2 <-
lm(log(salary)~OBP+SLG,data=subset(batters,batters$yearID==1996&batters$AB>=130))

#7n)
model3 <-
lm(log(salary)~OBP+SLG,data=subset(batters,batters$yearID==2006&batters$AB>=130))
summary(model3)
```