Sections 1.2 and 1.3 - Assssing Conditions for a Simple Linear Model

Load needed packages.

```
library(Stat2Data)
library(e1071) #used to create a different type of probability plot with probplot() funtion
```

EXAMPLE 1.5 Accord prices (continued)

Get the dataframe with **AccordPrice** data.

```
data(AccordPrice)
str(AccordPrice)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ Age    : int  7 4 4 7 9 1 18 2 2 5 ...
##  $ Price  : num  12 17.9 15.7 12.5 9.5 21.5 3.5 22.8 26.8 13.6 ...
##  $ Mileage: num  74.9 53 79.1 50.1 62 4.8 89.4 20.8 4.8 48.3 ...
```

Find the least-squares regression line.

```
regmodel=lm(Price~Mileage, data=AccordPrice)
summary(regmodel)
```

```
##
## Call:
## lm(formula = Price ~ Mileage, data = AccordPrice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5984 -1.8169 -0.4148  1.4502  6.5655
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.8096     0.9529   21.84  < 2e-16 ***
## Mileage      -0.1198     0.0141   -8.50 3.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.085 on 28 degrees of freedom
## Multiple R-squared:  0.7207, Adjusted R-squared:  0.7107
## F-statistic: 72.25 on 1 and 28 DF,  p-value: 3.055e-09
```

The fitted values and many other statistics are stored in the regmodel object.

```
fits=regmodel$fitted.values
fits
```

```
##         1         2         3         4         5         6         7         8
## 11.835698 14.459580 11.332488 14.807034 13.381272 20.234515 10.098425 18.317524
##         9        10        11        12        13        14        15        16
## 20.234515 15.022696 15.238357 20.450177 13.129667 19.815174 17.562709 18.377430
##        17        18        19        20        21        22        23        24
## 12.614476 10.397955 13.081742  2.777915 12.997874 14.088163  4.107827 19.144227
##        25        26        27        28        29        30
## 18.581111 18.928565 16.196853 18.437336  6.516047  6.132649
```

The residuals are also stored in the regmodel object. These will be very useful in checking conditions.

```
resids=regmodel$residuals   #store the residual values in a varaible named "resids"
resids
```

```
##          1          2          3          4          5          6          7
##  0.1643021  3.4404204  4.3675123 -2.3070342 -3.8812720  1.2654845 -6.5984246
##          8          9         10         11         12         13         14
##  4.4824757  6.5654845 -1.4226957  4.1616428 -0.9501770 -4.1296669 -2.4151737
##         15         16         17         18         19         20         21
##  0.2372910 -0.8774303  0.8855244 -3.3979545 -1.4817422  5.1220854 -1.2978738
##         22         23         24         25         26         27         28
##  1.5118375  0.8921728  1.8557732 -2.9811106 -1.9285653 -0.1968528 -0.8373363
##         29         30
##  0.3839526 -0.6326491
```

EXAMPLE 1.5 Accord prices (continued)

Compute the standard error of regression (by formula)

```
n=length(AccordPrice$Mileage)
SSE=sum(resids^2)                       #Sum of squared errors
Se=sqrt(SSE/(n-2))
Se
```

```
## [1] 3.08504
```

Note that you can find this value in the regression summary, labeled as "Resiudal standard error."

The code below will generate plots showing different sorts of results for residual versus fits and normal quantile plots.

FIGURE 1.5 Residuals versus fitted values plot when linearity and constant variance conditions hold

```
set.seed(11)    # for replcating the graphs
residual=rnorm(100)
YHat=runif(100,20,50)
plot(residual~YHat,ylim=c(-3,3),cex.lab=1.5,cex.axis=1.5,cex=1.5)
abline(0,0,col="darkblue")
```

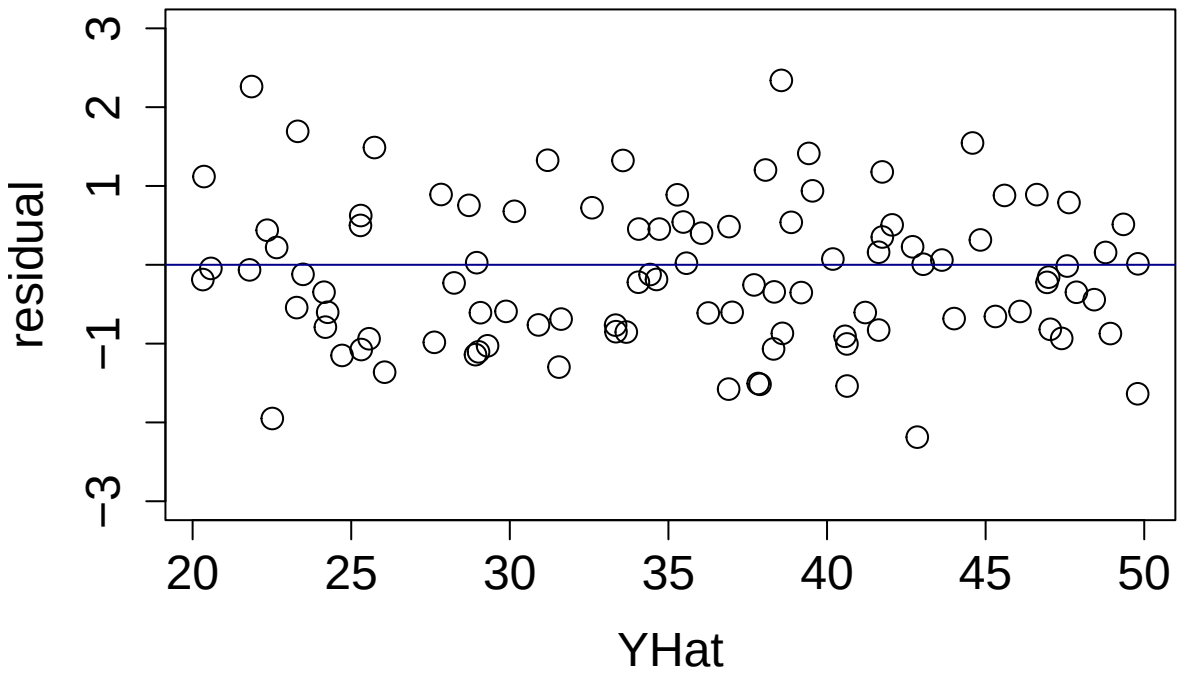


FIGURE 1.6 Residuals versus fitted values plots illustrating problems with conditions

FIGURE 1.6a - nonlinearity

```
YHat=runif(100,20,50)
newresidual=rnorm(100)+0.05*(YHat-32)^2
residual=(newresidual-mean(newresidual))/sd(newresidual)
plot(residual~YHat,cex.lab=1.5,cex.axis=1.5,cex=1.5)
abline(0,0,col="darkblue")
```

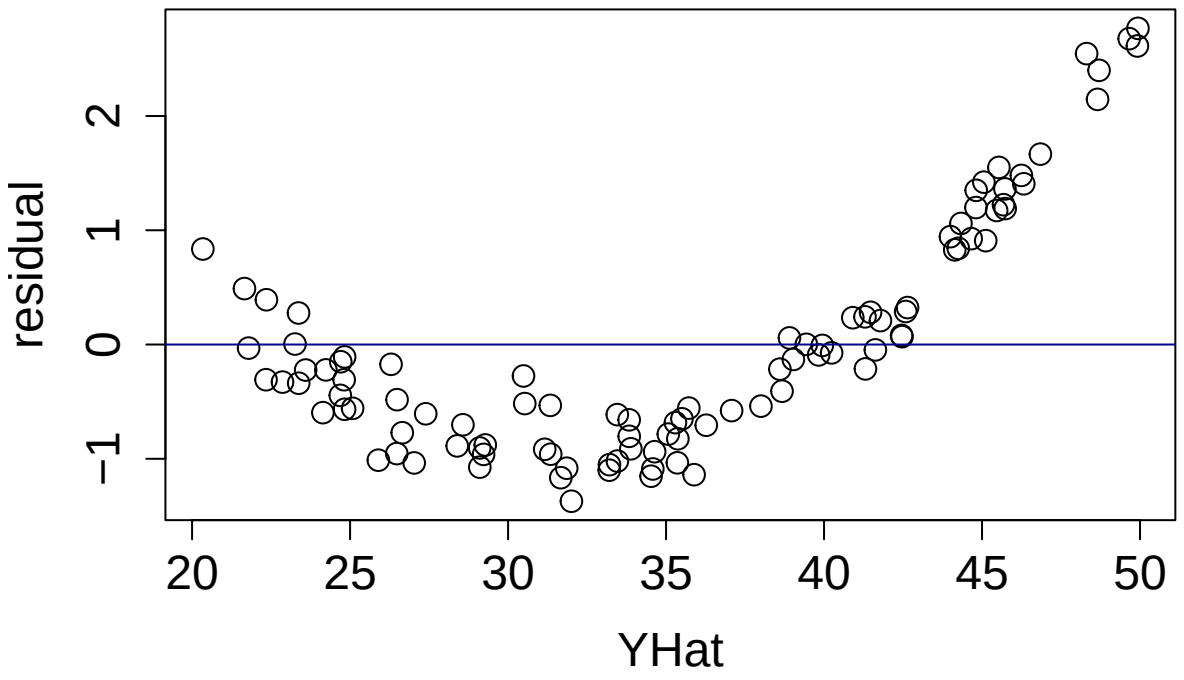


FIGURE 1.6b - nonconstant variability

```
YHat=runif(100,20,50)
newresidual=(YHat-19)*rnorm(100)
residual=(newresidual-mean(newresidual))/sd(newresidual)
plot(residual~YHat,cex.lab=1.5,cex.axis=1.5,cex=1.5,ylim=c(-3,3))
abline(0,0,col="darkblue")
```

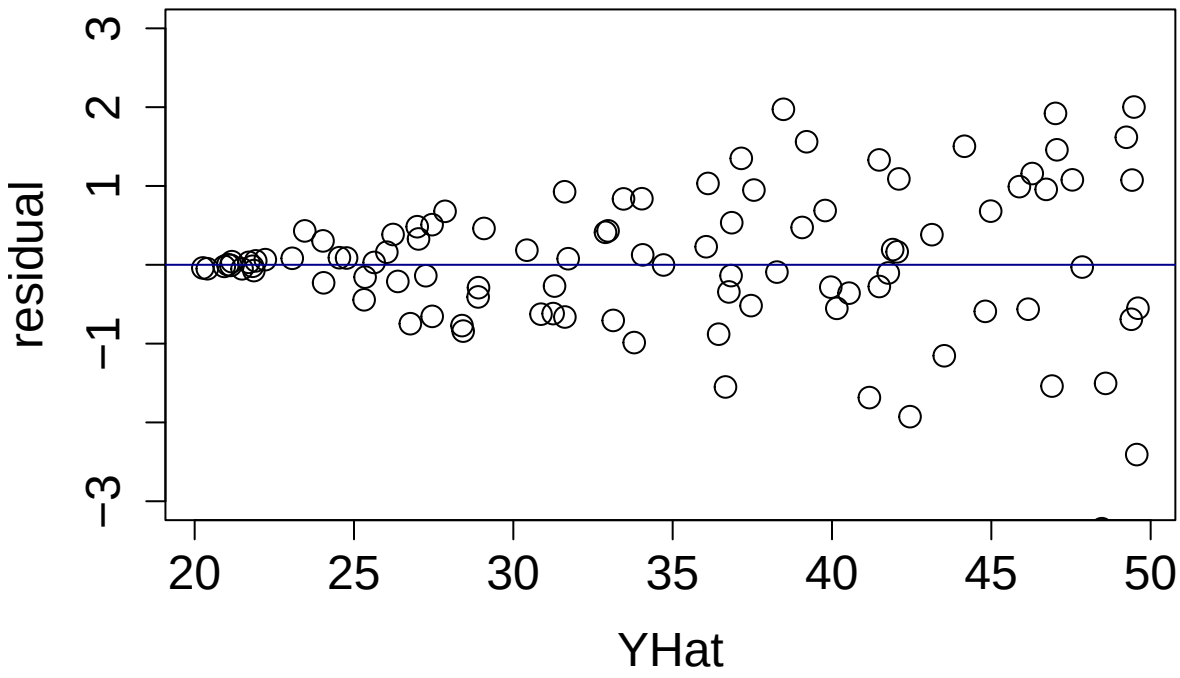


FIGURE 1.6c - Problem with linearity and constant variance This example uses the Perch data, then transforms.

```
data(Perch)
model1=lm(Weight~Length,data=Perch)
YHat=(model1$fitted+400)/50
newresidual=model1$residual
residual=(newresidual-mean(newresidual))/sd(newresidual)
plot(residual~YHat,cex.lab=1.5,cex.axis=1.5,cex=1.5,ylim=c(-3,3))
abline(0,0,col="darkblue")
```

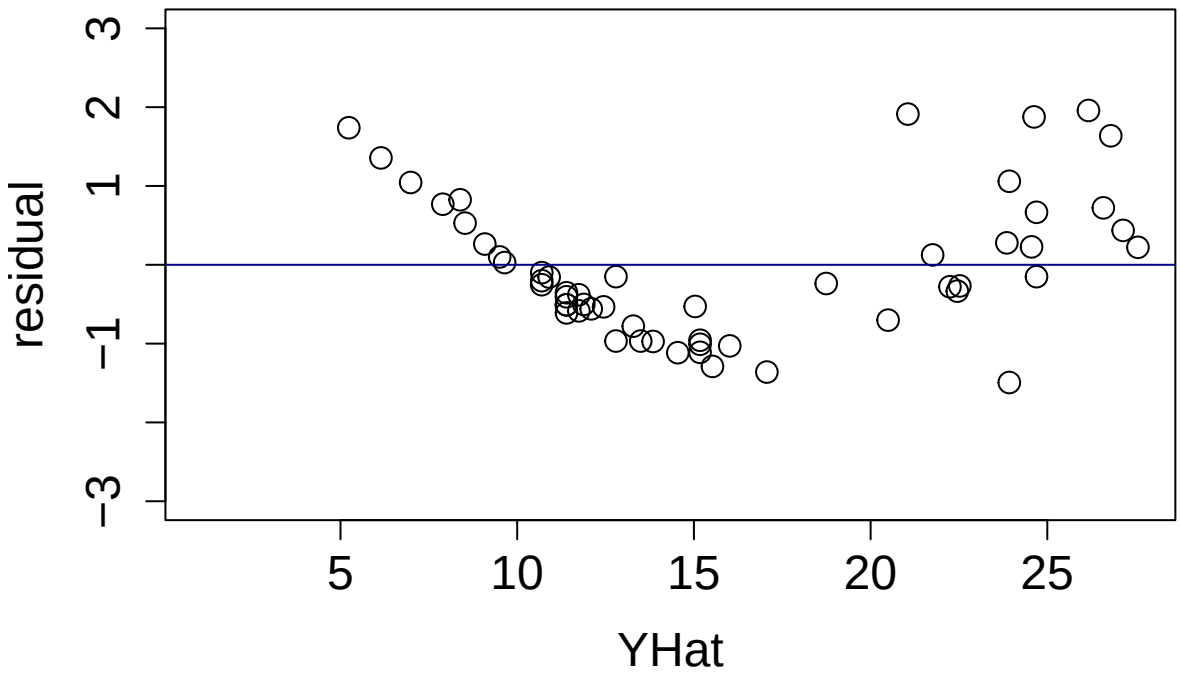


FIGURE 1.7 Examples of normal quantile plots

FIGURE 1.7a A “good” example

```
residual=rnorm(100)
qqnorm(residual,ylab="Residual",cex.lab=1.5,cex.axis=1.5,cex=1.5,main=NULL)
qqline(residual,col="darkblue")
```

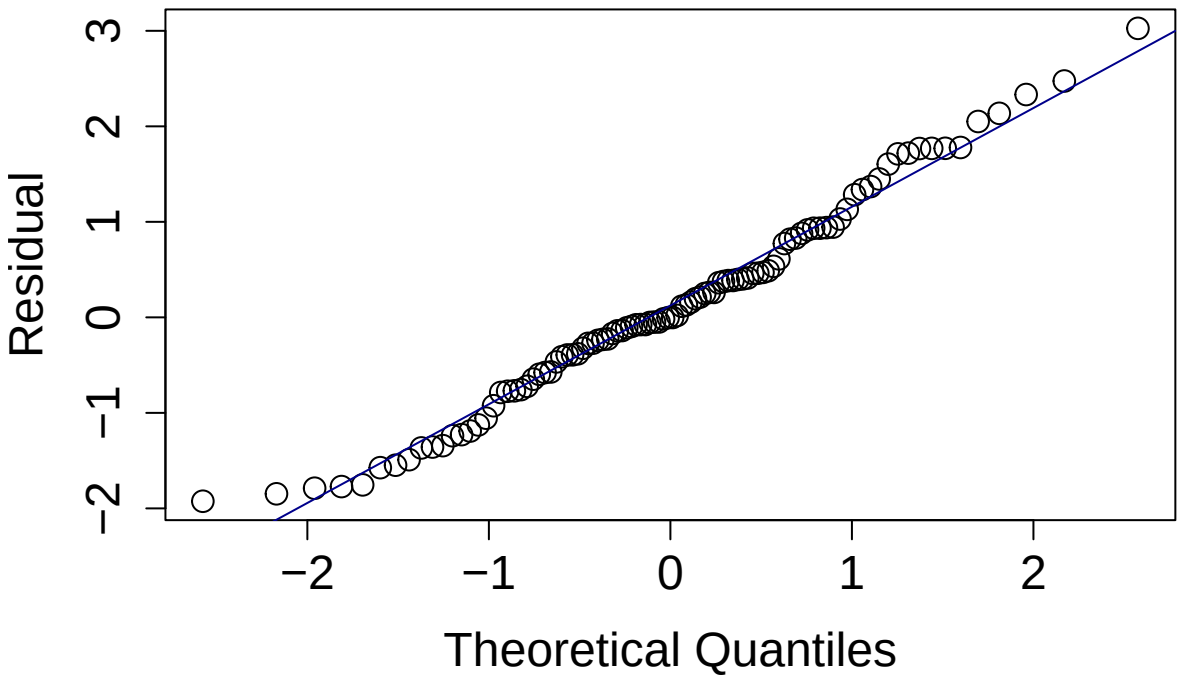


FIGURE 1.7b Skewed residuals

```
residual=rexp(100)
qqnorm(residual,ylab="Residual",cex.lab=1.5,cex.axis=1.5,cex=1.5,main=NULL)
qqline(residual,col="darkblue")
```

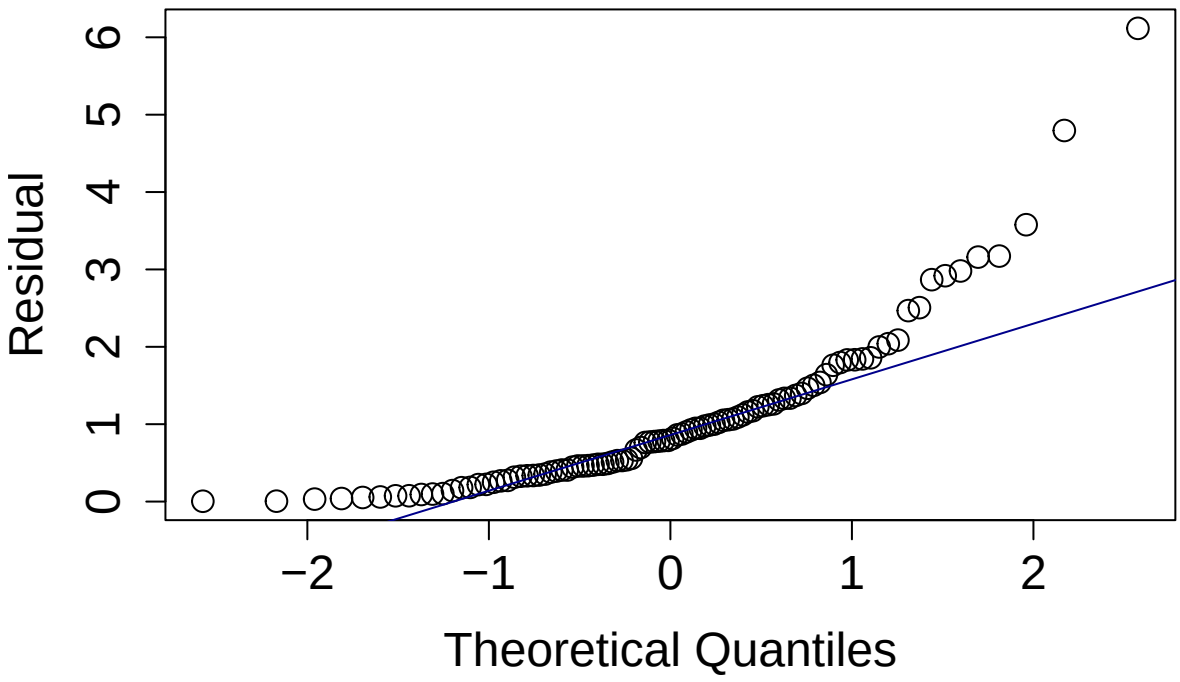


FIGURE 1.7c Long tails.

This example uses MetroHealth83 data from first edition.

```
data("MetroHealth83")
model2=lm(NumMDs~NumHospitals,data=MetroHealth83)
newresidual=model2$residual
residual=(newresidual-mean(newresidual))/sd(newresidual)
qqnorm(residual,ylab="Residual",cex.lab=1.5,cex.axis=1.5,cex=1.5,main=NULL)
qqline(residual,col="darkblue")
```

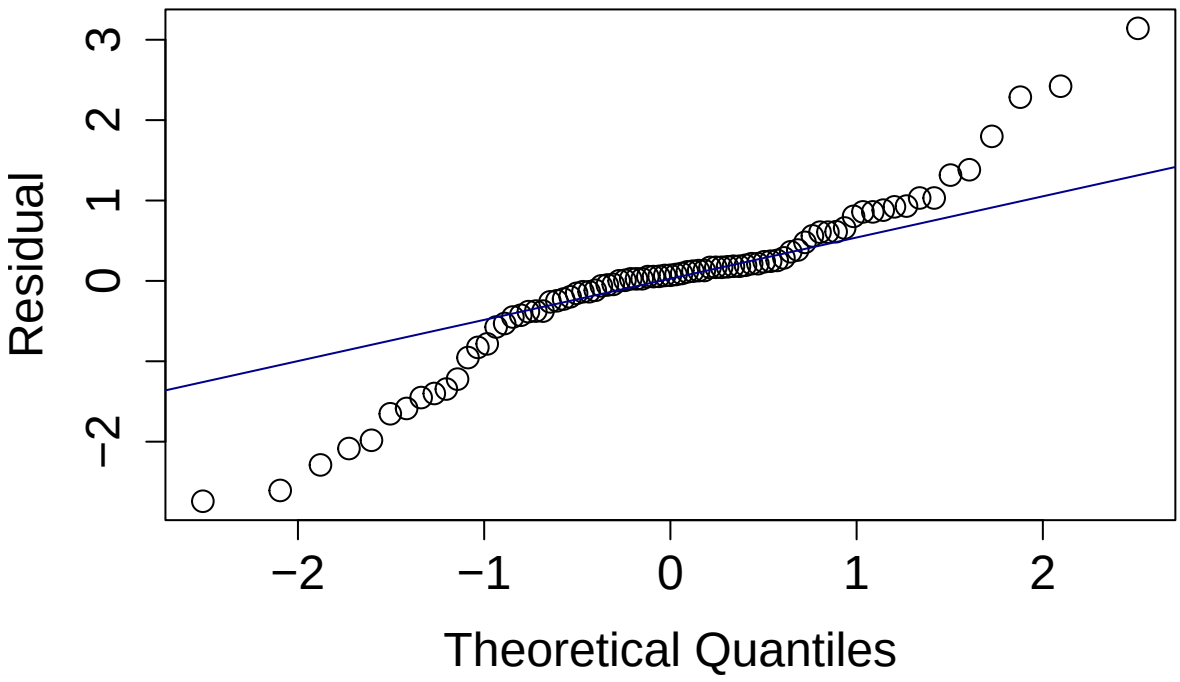


FIGURE 1.8 Examples of normal probability plots

Figure 1.8a Shows normal probability plots from Minitab, which are less common in R. Most R applications tend to use the QQ plot instead. There is a probplot function in the e1071 pacakge, which looks simialr to the Minitab plots.

```
residual=rnorm(100)
probplot(residual)
```

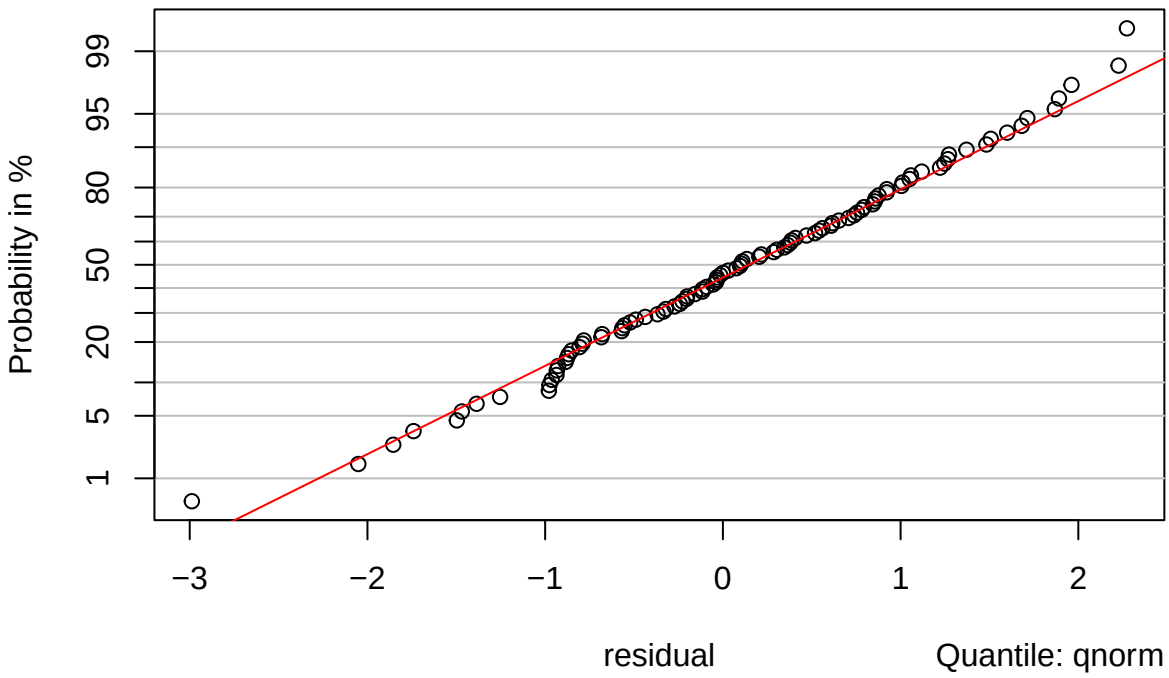
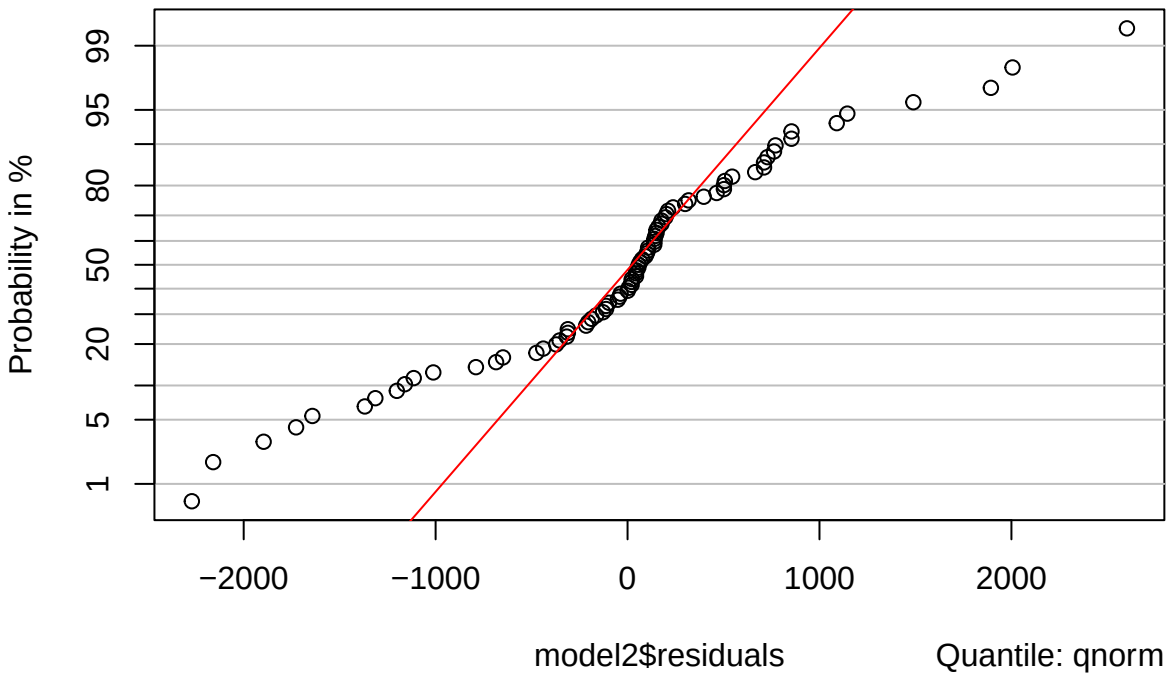


FIGURE 1.8b A Minitab-like normal plot with problems

```
#using the residuals from Figure 1.7c
probplot(model2$residuals)
```



EXAMPLE 1.6 Accord prices-checking conditions

Add the regression line to the scatterplot.

```
plot(Price~Mileage, data=AccordPrice)
abline(regmodel)
```

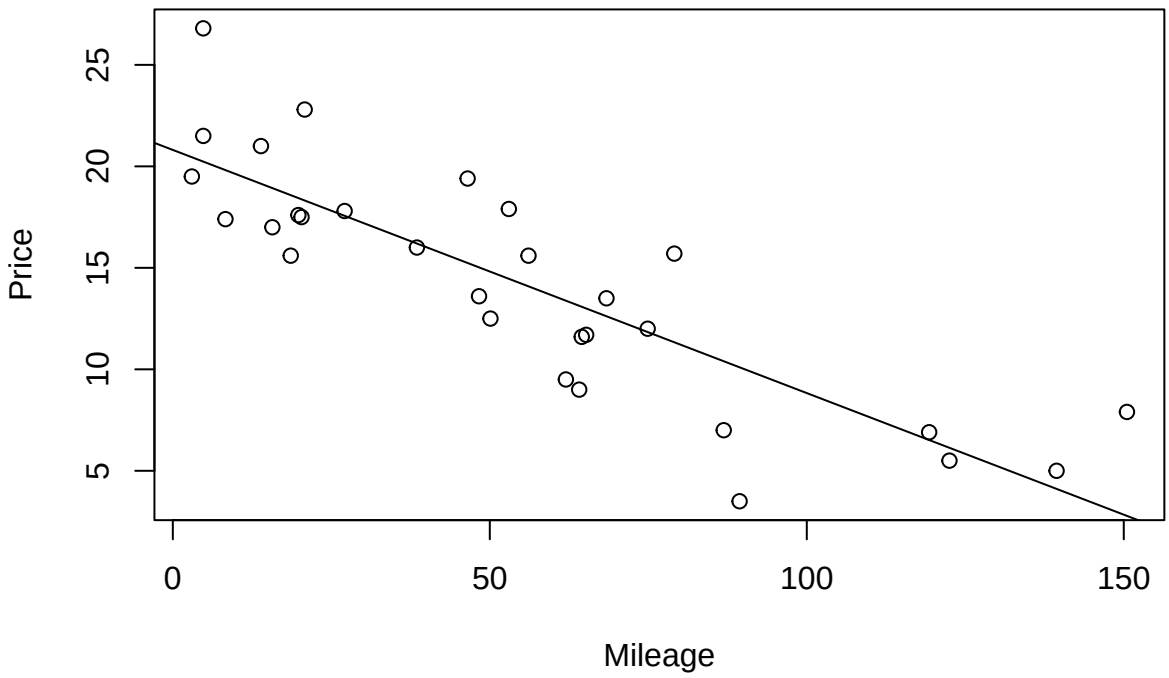


FIGURE 1.9 Plot of Accord residuals versus fitted values

Scatterplot of residuals versus fits (with horizontal line at zero for reference)

```
plot(resids~fits)
abline(0,0)
```

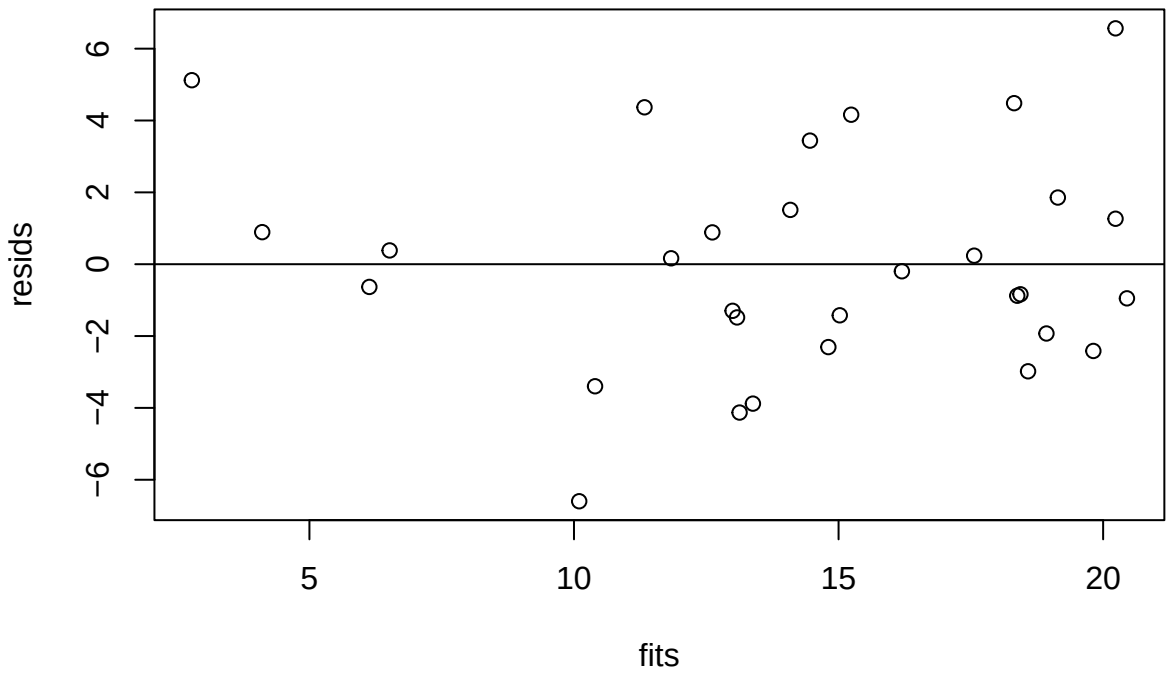


FIGURE 1.10 Histogram of Accord residuals

```
hist(resids)
```

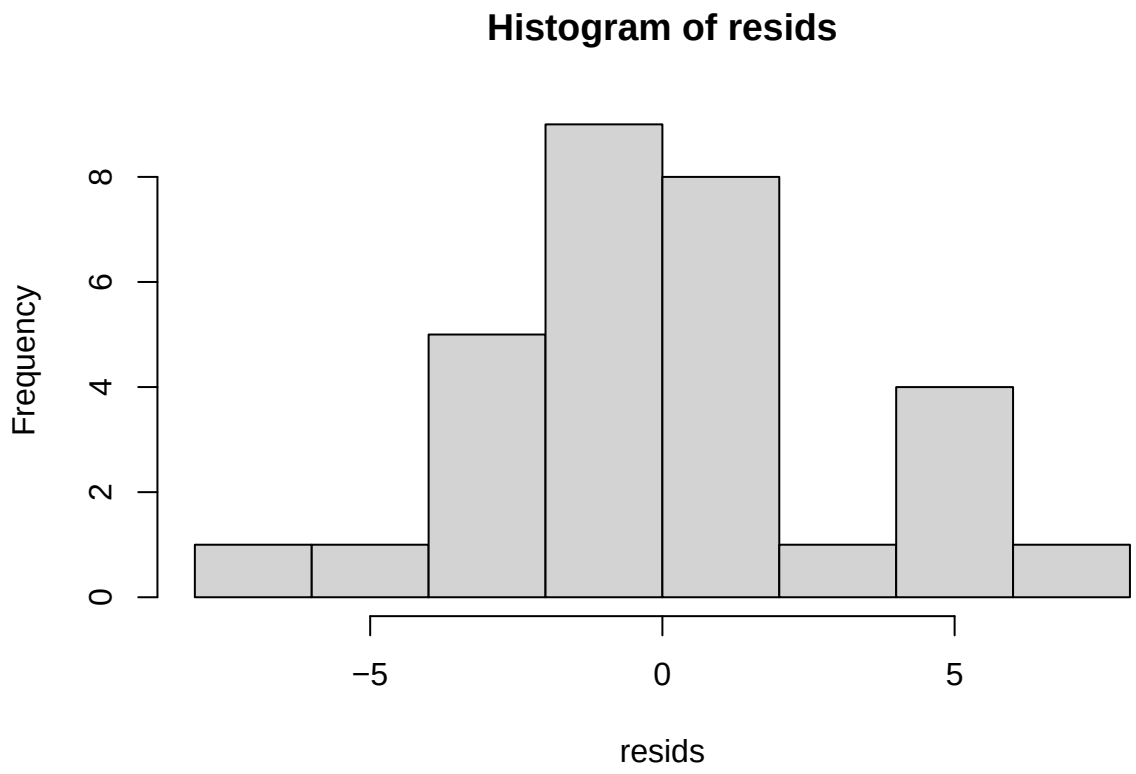
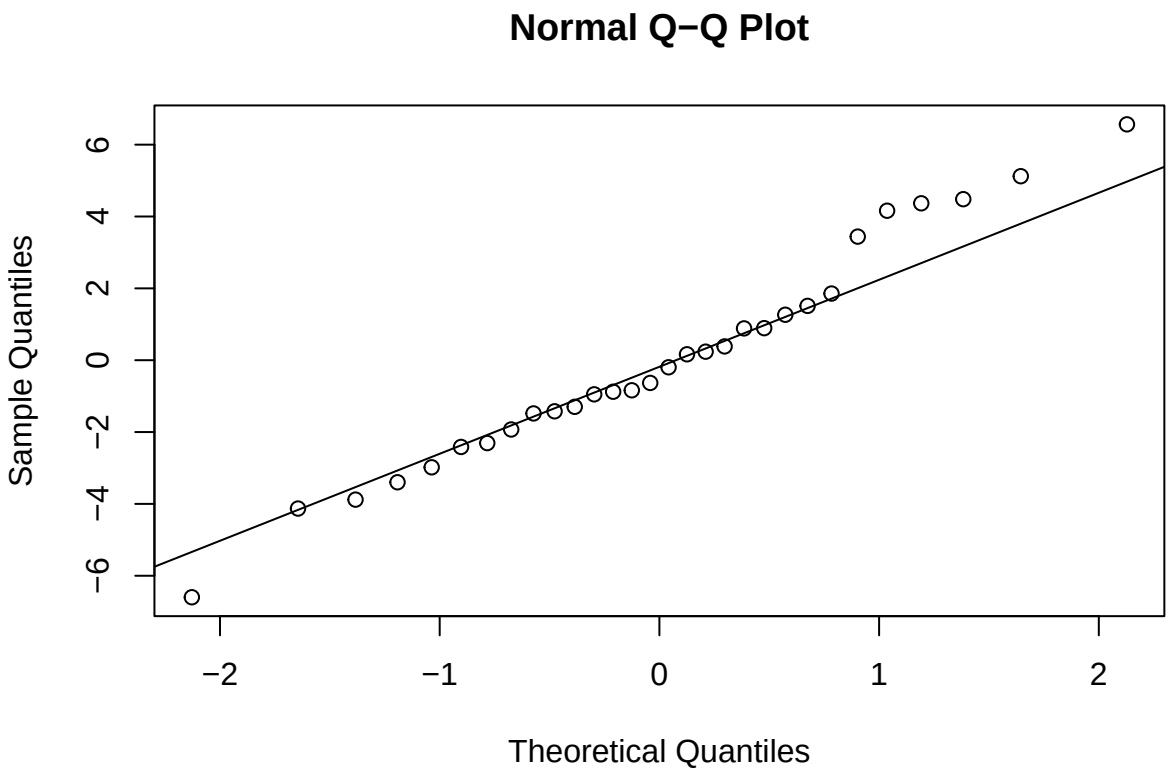


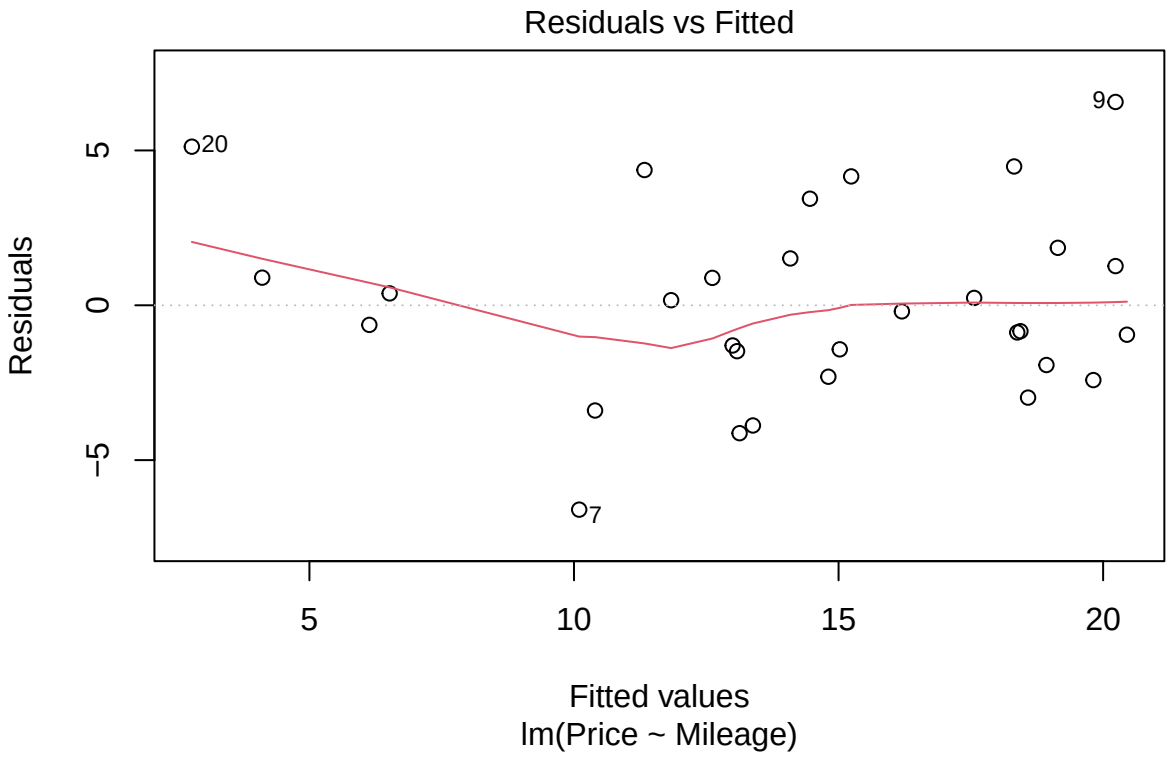
FIGURE 1.11 Normal quantile plot of residuals for Accord data

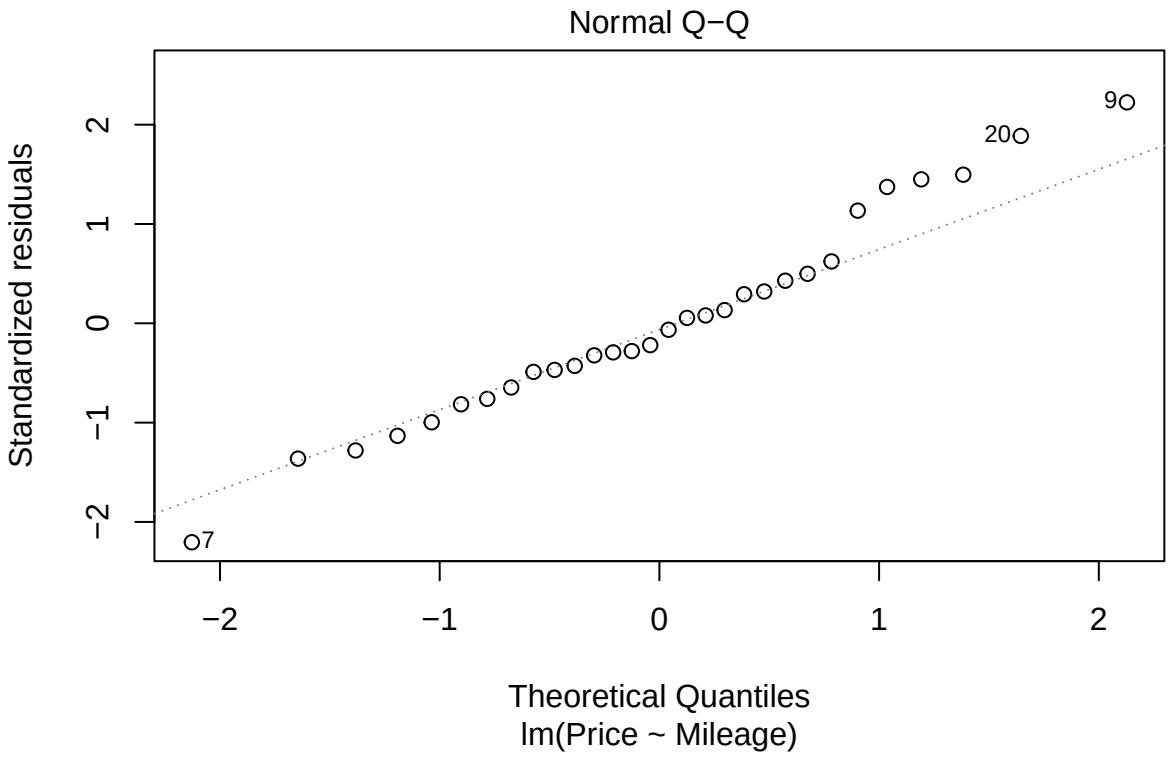
```
qqnorm(resids)
qqline(resids)
```

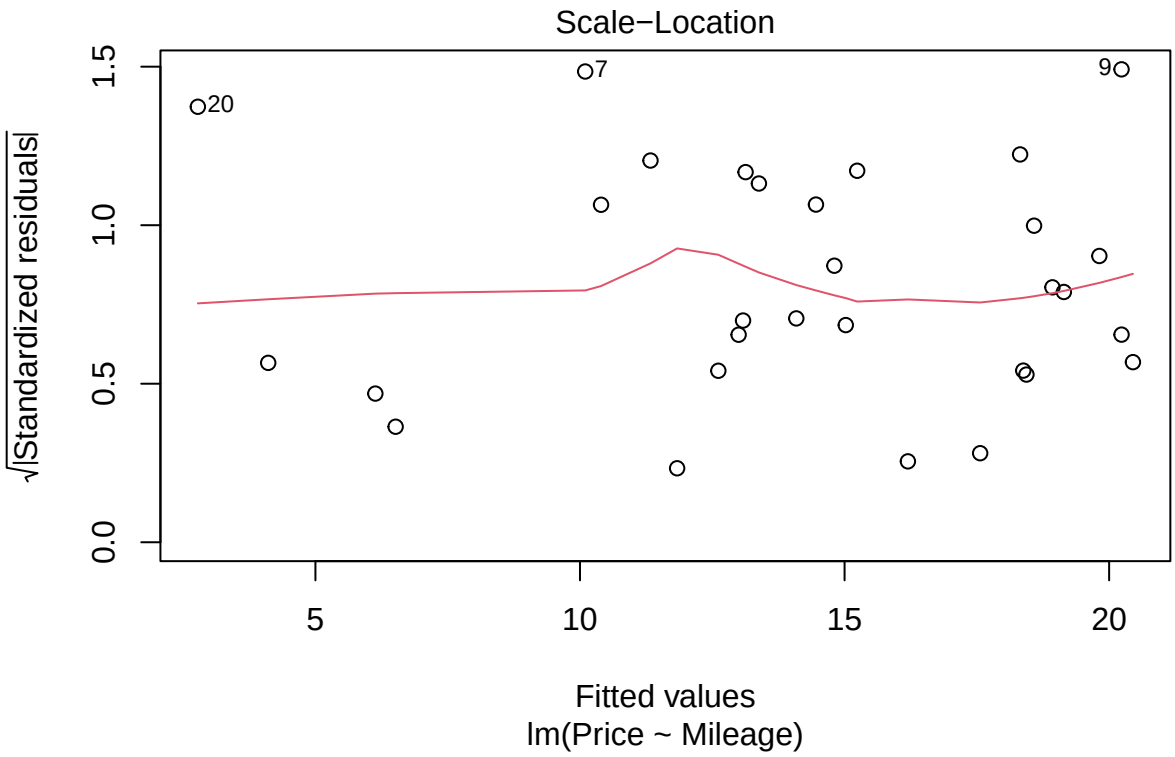


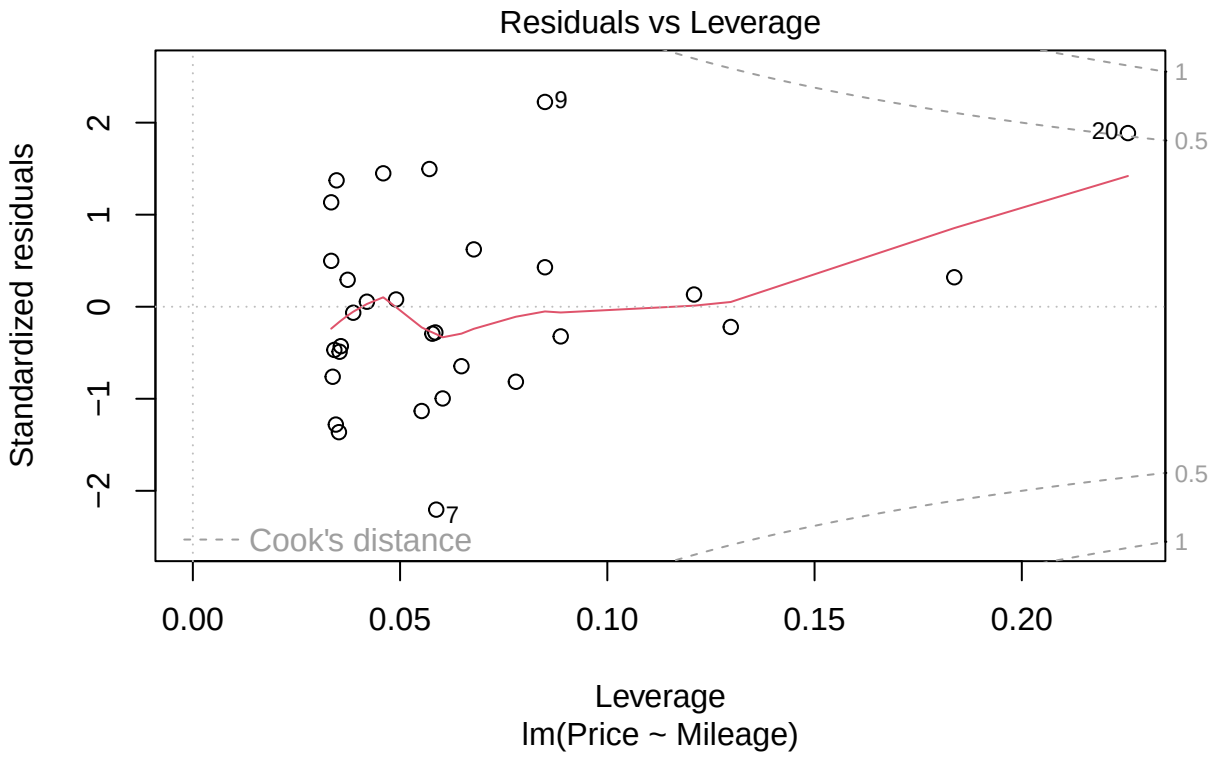
We can also use the plot command on a regression model to generate several diagnostic plots

```
plot(regmodel)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Price ~ Mileage)
20
9
7
5
0
−5
5
10
15
20


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Price ~ Mileage)
9
20
7
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
20
7
9
5
10
15
20
Fitted values
lm(Price ~ Mileage)



Alternative Solutions

Note: Several of the residual plots from plot(regmodel) use standardized residuals which aren't introduced until Topic 4.3. These are easy to find directly from a model. They can be substituted for ordinary residuals in any of the residual plots.

```
stdres=rstandard(regmodel)
stdres
```

```
##           1           2           3           4           5           6
##  0.05441255  1.13428779  1.44938964 -0.76075840 -1.28037430  0.42881893
##           7           8           9          10          11          12
## -2.20455072  1.49630214  2.22476376 -0.46923967  1.37299176 -0.32264606
##          13          14          15          16          17          18
## -1.36284362 -0.81528261  0.07887481 -0.29300658  0.29255382 -1.13315495
##          19          20          21          22          23          24
## -0.48903549  1.88671332 -0.42841842  0.49844532  0.32008447  0.62302677
##          25          26          27          28          29          30
## -0.99682141 -0.64643049 -0.06508041 -0.27972455  0.13273991 -0.21983069
```

```
studres=rstudent(regmodel)
studres
```

```
##           1           2           3           4           5           6
##  0.05343489  1.14035646  1.47986901 -0.75489238 -1.29580855  0.42248141
##           7           8           9          10          11          12
## -2.38133618  1.53186024  2.40783267 -0.46260672  1.39606514 -0.31742276
##          13          14          15          16          17          18
## -1.38501043 -0.81026676  0.07746213 -0.28816888  0.28772223 -1.13916279
##          19          20          21          22          23          24
## -0.48228741  1.98305344 -0.42208421  0.49164968  0.31489335  0.61608542
##          25          26          27          28          29          30
## -0.99670426 -0.63957257 -0.06391253 -0.27506866  0.13038904 -0.21605598
```

Plot standardized residuals, instead of residuals, against fitted values.

```
plot(stdres~regmodel$fitted.values,ylim=c(-3,3),cex.lab=1.5,cex.axis=1.5,cex=1.5)
abline(0,0,col="darkblue")
```