Topic 4.5 Coding Categorical Predictors

Load needed packages.

```
library(Stat2Data)
library(mosaic)
```

EXAMPLE 4.9 Car prices

Load **ThreeCars2017** data from Stat2Data package and look at the structure of the data.

```
data(ThreeCars2017)
str(ThreeCars2017)
```

```
## 'data.frame':    90 obs. of  7 variables:
##  $ CarType: Factor w/ 3 levels "Accord","Maxima",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Age    : int  3 2 1 2 2 1 2 3 3 4 ...
##  $ Price  : num  15.9 16.4 18.9 16.9 20.5 19 17.5 18 13.6 12 ...
##  $ Mileage: num  17.8 19 20.9 24 24 24.2 30.1 32 34.8 35.7 ...
##  $ Mazda6 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Accord : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Maxima : int  0 0 0 0 0 0 0 0 0 0 ...
```

Use mosiac package to find the summary statistics for each car model.

```
favstats(Price~CarType,data=ThreeCars2017)
```

```
##   CarType min     Q1 median     Q3  max     mean       sd  n missing
## 1  Accord 3.5 10.025   15.6 17.750 26.8 14.27667 5.736013 30       0
## 2  Maxima 4.8 13.000   15.4 17.875 27.0 15.46333 4.563762 30       0
## 3  Mazda6 2.0  6.050   11.0 16.775 20.5 11.50333 5.679029 30       0
```

What happens if we put all three indicators in the same model?

```
mod3=lm(Price~Accord+Mazda6+Maxima,data=ThreeCars2017)
summary(mod3)
```

```
##
## Call:
## lm(formula = Price ~ Accord + Mazda6 + Maxima, data = ThreeCars2017)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7767  -3.6233   0.1367   4.4267  12.5233
##
```

```
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15.4633     0.9774  15.821  < 2e-16 ***
## Accord       -1.1867     1.3823  -0.858  0.39298
## Mazda6       -3.9600     1.3823  -2.865  0.00523 **
## Maxima           NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.354 on 87 degrees of freedom
## Multiple R-squared:  0.0904, Adjusted R-squared:  0.06949
## F-statistic: 4.323 on 2 and 87 DF,  p-value: 0.01622
```

Notice that R automatically leaves out the last indicator for Maxima. However, we shouldn't rely on R to do that for us. We can choose which indicator to omit. In the model below, let's omit Accord.

```
mod2=lm(Price~Mazda6+Maxima,data=ThreeCars2017)
summary(mod2)
```

```
##
## Call:
## lm(formula = Price ~ Mazda6 + Maxima, data = ThreeCars2017)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7767  -3.6233   0.1367   4.4267  12.5233
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.2767     0.9774  14.607   <2e-16 ***
## Mazda6       -2.7733     1.3823  -2.006   0.0479 *
## Maxima        1.1867     1.3823   0.858   0.3930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.354 on 87 degrees of freedom
## Multiple R-squared:  0.0904, Adjusted R-squared:  0.06949
## F-statistic: 4.323 on 2 and 87 DF,  p-value: 0.01622
```

The coefficients are different than when Maxima was left out, but the overall measures of fit (R^2 etc.) are the same.

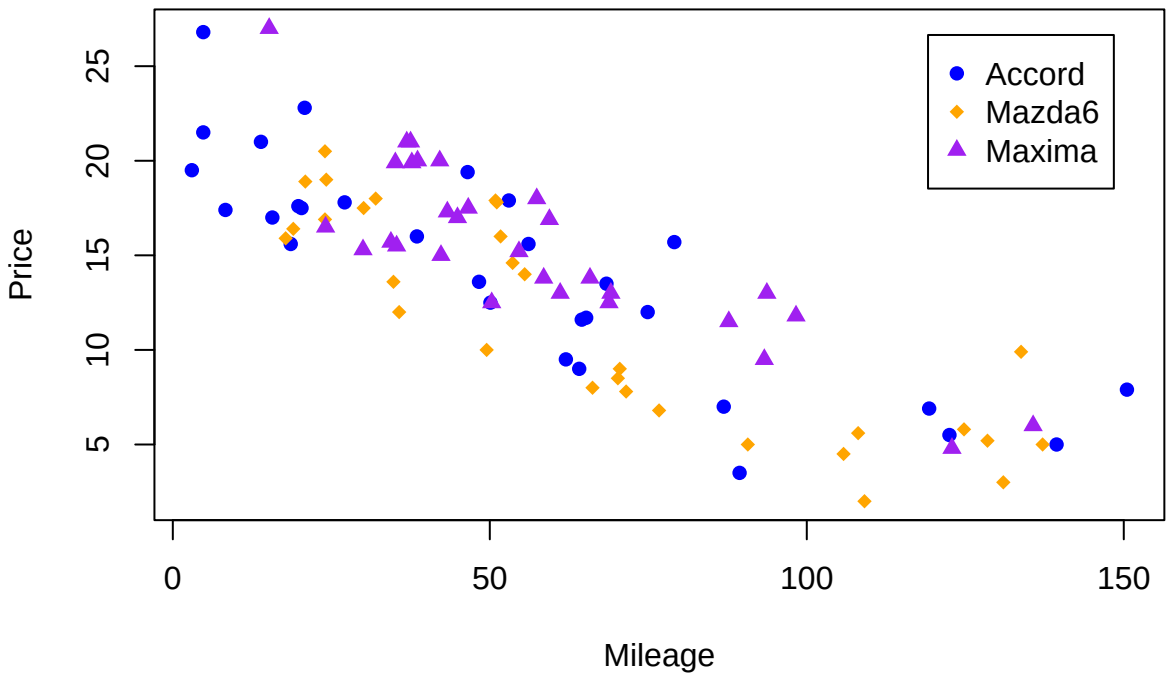
EXAMPLE 4.10 More car prices

CHOOSE

FIGURE 4.11 Price versus Mileage for three car types

Here is code that produces the scatterplot of Price versus Mileage with different symbols for the car types.

```
plot(Price ~ Mileage, type="n", data=ThreeCars2017)
points(Price ~ Mileage, pch=16, col="blue", data=filter(ThreeCars2017, CarType=="Accord"))
points(Price ~ Mileage, pch=18, col="orange", data=filter(ThreeCars2017, CarType=="Mazda6"))
points(Price ~ Mileage, pch=17, col="purple", data=filter(ThreeCars2017, CarType=="Maxima"))
legend("topright", legend=c("Accord","Mazda6","Maxima"), pch=c(16,18,17),
     col=c("blue","orange","purple"),inset=.05)
```



FIT

Fit regression model with mileage and indicators for Mazda6 and Maxima

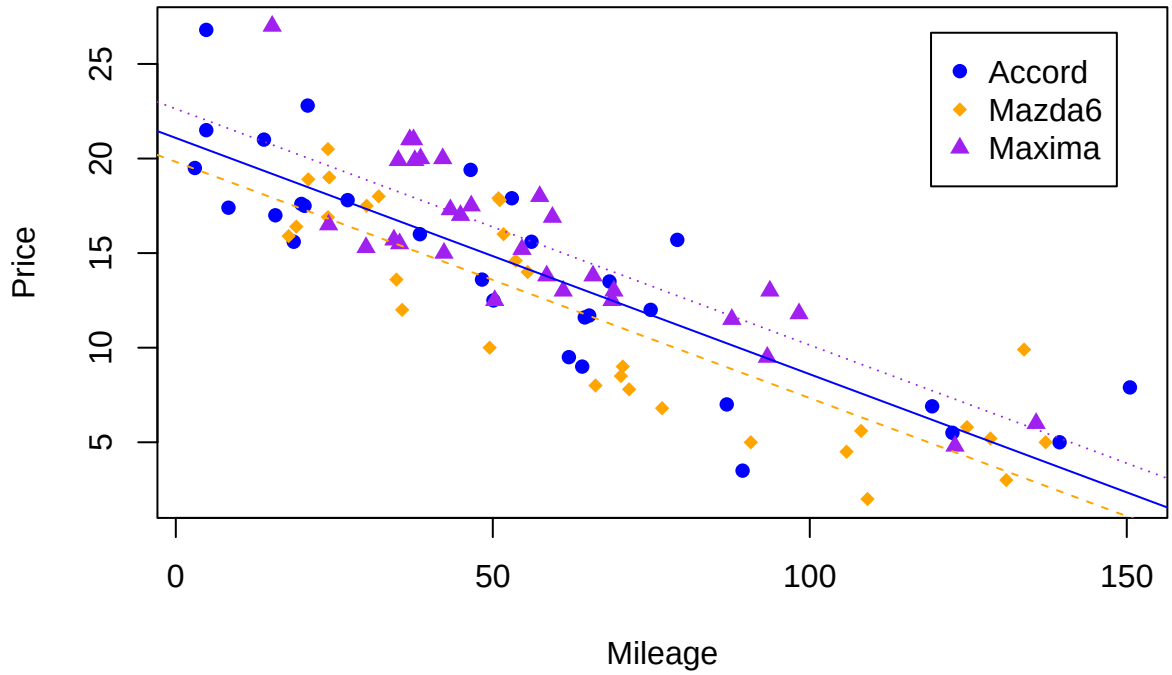
```
modMilesType=lm(Price~Mileage+Mazda6+Maxima,data=ThreeCars2017)
summary(modMilesType)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Mazda6 + Maxima, data = ThreeCars2017)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4208 -2.1225 -0.2257  1.6904  6.7866
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.087383   0.682805  30.883   <2e-16 ***
## Mileage     -0.124906   0.008252 -15.136   <2e-16 ***
## Mazda6      -1.261552   0.733145  -1.721   0.0889 .
## Maxima       1.539735   0.726685   2.119   0.0370 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.813 on 86 degrees of freedom
## Multiple R-squared:  0.7518, Adjusted R-squared:  0.7431
## F-statistic: 86.81 on 3 and 86 DF,  p-value: < 2.2e-16
```

FIGURE 4.12 Price versus Mileage with equal slope fits

This plot adds the regression lines to the previous figure.

```
plot(Price ~ Mileage, type="n", data=ThreeCars2017)
points(Price ~ Mileage, pch=16, col="blue", data=filter(ThreeCars2017, CarType=="Accord"))
points(Price ~ Mileage, pch=18, col="orange", data=filter(ThreeCars2017, CarType=="Mazda6"))
points(Price ~ Mileage, pch=17, col="purple", data=filter(ThreeCars2017, CarType=="Maxima"))
legend("topright", legend=c("Accord","Mazda6","Maxima"), pch=c(16,18,17),
     col=c("blue","orange","purple"),inset=.05)
abline(21.087,-0.1249,col="blue",lty=1)
abline(21.087-1.26155,-0.1249,col="orange",lty=2)
abline(21.087+1.5397,-0.1249,col="purple",lty=3)
```



ASSESS

Here is one way ("brute force" with indicator variables) to fit the model with separate lines for each type. Another approach, which you may prefer, is shown below in the alternative solutions section.

```
mod3slopes=lm(Price~Mileage+Mazda6+Maxima+Mileage*Mazda6+Mileage*Maxima,data=ThreeCars2017)
summary(mod3slopes)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Mazda6 + Maxima + Mileage * Mazda6 +
##     Mileage * Maxima, data = ThreeCars2017)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5984 -2.0047 -0.1778  1.8321  6.7536
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20.809613   0.876372  23.745  < 2e-16 ***
## Mileage        -0.119812   0.012964  -9.242 1.93e-14 ***
## Mazda6         -1.016487   1.355525  -0.750   0.4554
## Maxima          2.461613   1.467904   1.677   0.0973 .
## Mileage:Mazda6 -0.004603   0.018668  -0.247   0.8058
## Mileage:Maxima -0.016325   0.022540  -0.724   0.4709
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.837 on 84 degrees of freedom
## Multiple R-squared:  0.7533, Adjusted R-squared:  0.7386
## F-statistic:  51.3 on 5 and 84 DF,  p-value: < 2.2e-16
```

FIGURE 4.13 Price versus Mileage with different linear fits

To show the regression lines with difference slopes on the scatterplot, we can modify the code for Figure 4.12 using the regression output from the interaction model. Note that this is a bit tedious, but it helps you understand exactly what is being added to the plot. We show an easier way to create this plot below.

```
plot(Price ~ Mileage, type="n", data=ThreeCars2017)
points(Price ~ Mileage, pch=16, col="blue", data=filter(ThreeCars2017, CarType=="Accord"))
points(Price ~ Mileage, pch=18, col="orange", data=filter(ThreeCars2017, CarType=="Mazda6"))
points(Price ~ Mileage, pch=17, col="purple", data=filter(ThreeCars2017, CarType=="Maxima"))
legend("topright", legend=c("Accord","Mazda6","Maxima"), pch=c(16,18,17),
     col=c("blue","orange","purple"),inset=.05)
abline(20.8096,-0.1198,col="blue",lty=1)
abline(20.8096-1.0165,-0.1198-0.0046,col="orange",lty=2)
abline(20.8096+2.4616,-0.1198-0.016325,col="purple",lty=3)
```

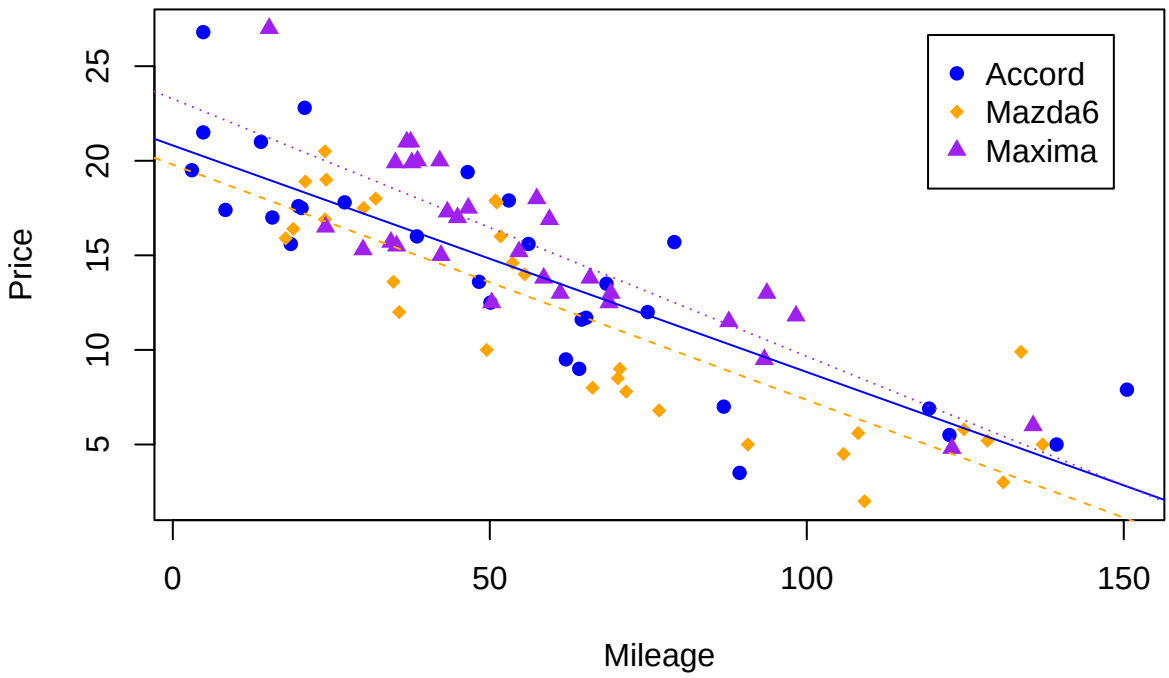


TABLE 4.2 Several models for predicting car price

We need to run several regressions to get the information for this table.

```
mod1=lm(Price~Mileage,data=ThreeCars2017)                    #just mileage
mod2=lm(Price~CarType,data=ThreeCars2017)                    #just car types
modreduced=lm(Price~Mileage+CarType,data=ThreeCars2017)        #parallel lines
modfull=lm(Price~Mileage*CarType,data=ThreeCars2017)         #3 different lines
summary(mod1)
```

```
##
## Call:
## lm(formula = Price ~ Mileage, data = ThreeCars2017)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4205 -2.1502  0.0334  1.8751  7.5806
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.365247   0.609265   35.07   <2e-16 ***
## Mileage     -0.128018   0.008741  -14.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.01 on 88 degrees of freedom
```

```
## Multiple R-squared:  0.7091, Adjusted R-squared:  0.7058
## F-statistic: 214.5 on 1 and 88 DF,  p-value: < 2.2e-16
```

```
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## Mileage    1 1943.81 1943.81   214.5 < 2.2e-16 ***
## Residuals 88  797.46    9.06
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = Price ~ CarType, data = ThreeCars2017)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7767  -3.6233   0.1367   4.4267  12.5233
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    14.2767     0.9774  14.607   <2e-16 ***
## CarTypeMaxima   1.1867     1.3823   0.858   0.3930
## CarTypeMazda6  -2.7733     1.3823  -2.006   0.0479 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.354 on 87 degrees of freedom
## Multiple R-squared:  0.0904, Adjusted R-squared:  0.06949
## F-statistic: 4.323 on 2 and 87 DF,  p-value: 0.01622
```

```
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## CarType    2  247.81  123.91  4.3232 0.01622 *
## Residuals 87 2493.45   28.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(modreduced)
```

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

```
## lm(formula = Price ~ Mileage + CarType, data = ThreeCars2017)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4208 -2.1225 -0.2257  1.6904  6.7866
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.087383   0.682805  30.883   <2e-16 ***
## Mileage       -0.124906   0.008252 -15.136   <2e-16 ***
## CarTypeMaxima  1.539735   0.726685   2.119   0.0370 *
## CarTypeMazda6 -1.261552   0.733145  -1.721   0.0889 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.813 on 86 degrees of freedom
## Multiple R-squared:  0.7518, Adjusted R-squared:  0.7431
## F-statistic: 86.81 on 3 and 86 DF,  p-value: < 2.2e-16
```

```
anova(modreduced)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## Mileage    1 1943.81 1943.81 245.6506 < 2.2e-16 ***
## CarType    2  116.95   58.47   7.3896  0.001093 **
## Residuals 86  680.51    7.91
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(modfull)
```

```
##
## Call:
## lm(formula = Price ~ Mileage * CarType, data = ThreeCars2017)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5984 -2.0047 -0.1778  1.8321  6.7536
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             20.809613   0.876372  23.745  < 2e-16 ***
## Mileage                 -0.119812   0.012964  -9.242 1.93e-14 ***
## CarTypeMaxima            2.461613   1.467904   1.677   0.0973 .
## CarTypeMazda6           -1.016487   1.355525  -0.750   0.4554
## Mileage:CarTypeMaxima -0.016325   0.022540  -0.724   0.4709
## Mileage:CarTypeMazda6 -0.004603   0.018668  -0.247   0.8058
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.837 on 84 degrees of freedom
```

```
## Multiple R-squared:  0.7533, Adjusted R-squared:  0.7386
## F-statistic:  51.3 on 5 and 84 DF,  p-value: < 2.2e-16
```

```
anova(modfull)
```

```
## Analysis of Variance Table
##
## Response: Price
##                  Df  Sum Sq Mean Sq  F value    Pr(>F)
## Mileage           1 1943.81 1943.81 241.4423 < 2.2e-16 ***
## CarType           2  116.95   58.47   7.2630  0.001232 **
## Mileage:CarType   2    4.24    2.12   0.2634  0.769097
## Residuals        84  676.27    8.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nested F test to see if we need the diffent slopes.

Full model = mod5 above.

Reduced model = mod3 above.

```
anova(modreduced,modfull)
```

```
## Analysis of Variance Table
##
## Model 1: Price ~ Mileage + CarType
## Model 2: Price ~ Mileage * CarType
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     86 680.51
## 2     84 676.27  2    4.2405 0.2634 0.7691
```

FIGURE 4.14 Residual plots for Price model based on Mileage, Mazda6, and Maxima

These plots use the residuals for the reduced model (modreduced) above.

FIGURE 4.14a - Studentized residuals versus Fits

```
plot(rstudent(modreduced)~modreduced$fitted.values,ylab="Studnetized Residuals",xlab="Predicted Price",
abline(0,0)
```

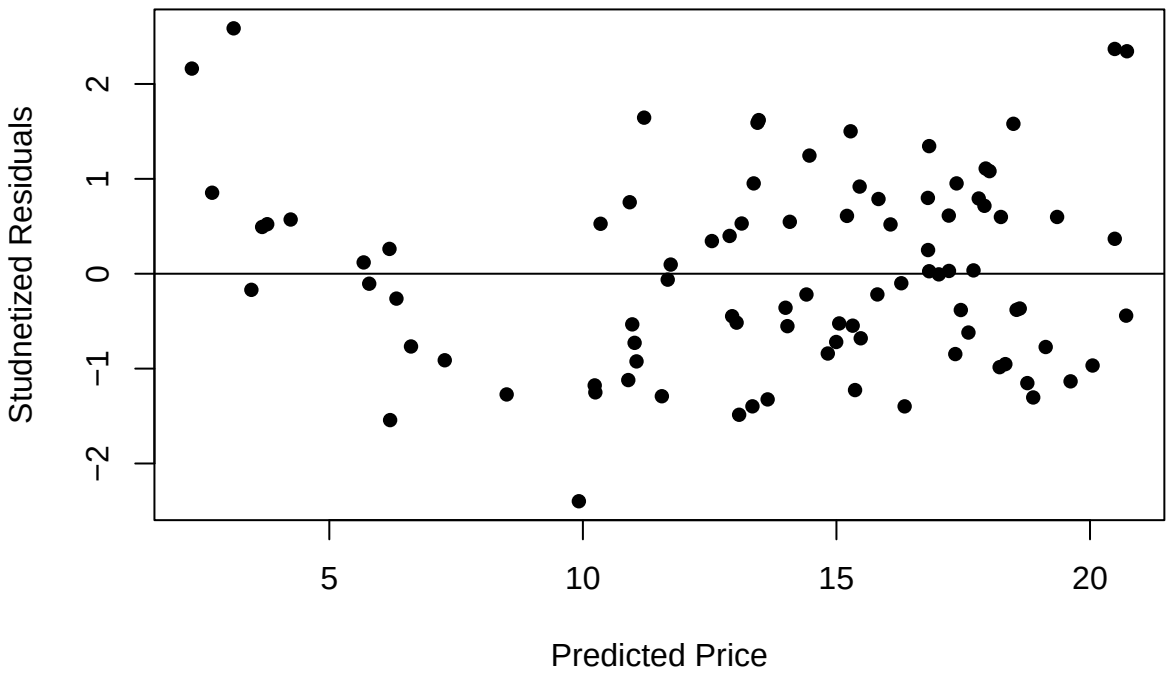
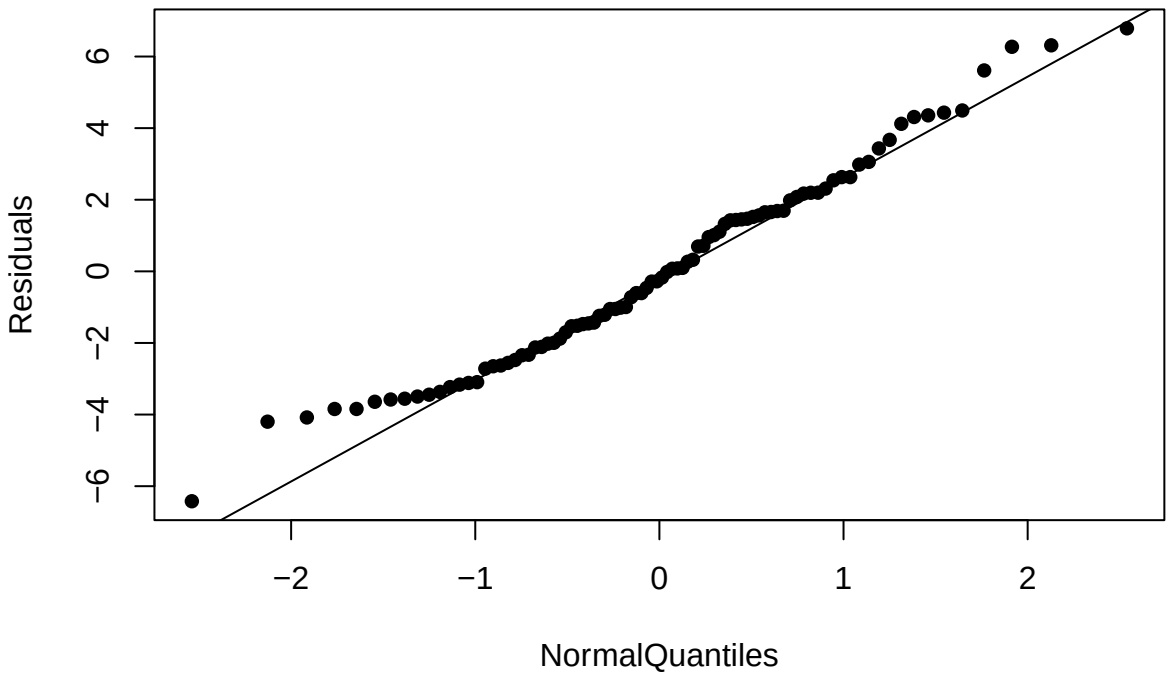


FIGURE 4.14b Normal quantile plot for residuals

Note: The figure in the text omits the line.

```
qqnorm(modreduced$residuals,ylab="Residuals",main="",xlab="NormalQuantiles", pch=16)
qqline(modreduced$residuals)
```



USE

Find predicted values and intervals for each type of car when Mileage = 50.

```
newdata=data.frame(CarType=c("Accord","Mazda6","Maxima"),Mileage=c(50,50,50))
predict(modreduced,newdata,interval="confidence")
```

```
##        fit      lwr      upr
## 1 14.84208 13.81842 15.86573
## 2 13.58052 12.52374 14.63730
## 3 16.38181 15.35375 17.40987
```

```
predict(modreduced,newdata,interval="prediction")
```

```
##        fit      lwr      upr
## 1 14.84208  9.15712 20.52703
## 2 13.58052  7.88951 19.27154
## 3 16.38181 10.69606 22.06756
```

Alternative solutions to the approaches above.

Fitting the model for Example 4.10 without using indicator variables. R creates them automatically from a variable that is a factor of categories.

```
modMilesTypea=lm(Price~Mileage+CarType,data=ThreeCars2017)
summary(modMilesTypea)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + CarType, data = ThreeCars2017)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4208 -2.1225 -0.2257  1.6904  6.7866
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.087383   0.682805  30.883   <2e-16 ***
## Mileage       -0.124906   0.008252 -15.136   <2e-16 ***
## CarTypeMaxima  1.539735   0.726685   2.119   0.0370 *
## CarTypeMazda6 -1.261552   0.733145  -1.721   0.0889 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.813 on 86 degrees of freedom
## Multiple R-squared:  0.7518, Adjusted R-squared:  0.7431
## F-statistic: 86.81 on 3 and 86 DF,  p-value: < 2.2e-16
```

Here is another way to fit different models for each car type that uses the CarType variable as a factor and automatically generates the interaction terms.

```
mod3slopesa=lm(Price~Mileage*CarType,data=ThreeCars2017)
summary(mod3slopesa)
```

```
##
## Call:
## lm(formula = Price ~ Mileage * CarType, data = ThreeCars2017)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5984 -2.0047 -0.1778  1.8321  6.7536
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           20.809613   0.876372  23.745  < 2e-16 ***
## Mileage               -0.119812   0.012964  -9.242 1.93e-14 ***
## CarTypeMaxima          2.461613   1.467904   1.677   0.0973 .
## CarTypeMazda6         -1.016487   1.355525  -0.750   0.4554
## Mileage:CarTypeMaxima -0.016325   0.022540  -0.724   0.4709
## Mileage:CarTypeMazda6 -0.004603   0.018668  -0.247   0.8058
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.837 on 84 degrees of freedom
## Multiple R-squared:  0.7533, Adjusted R-squared:  0.7386
## F-statistic:  51.3 on 5 and 84 DF,  p-value: < 2.2e-16
```

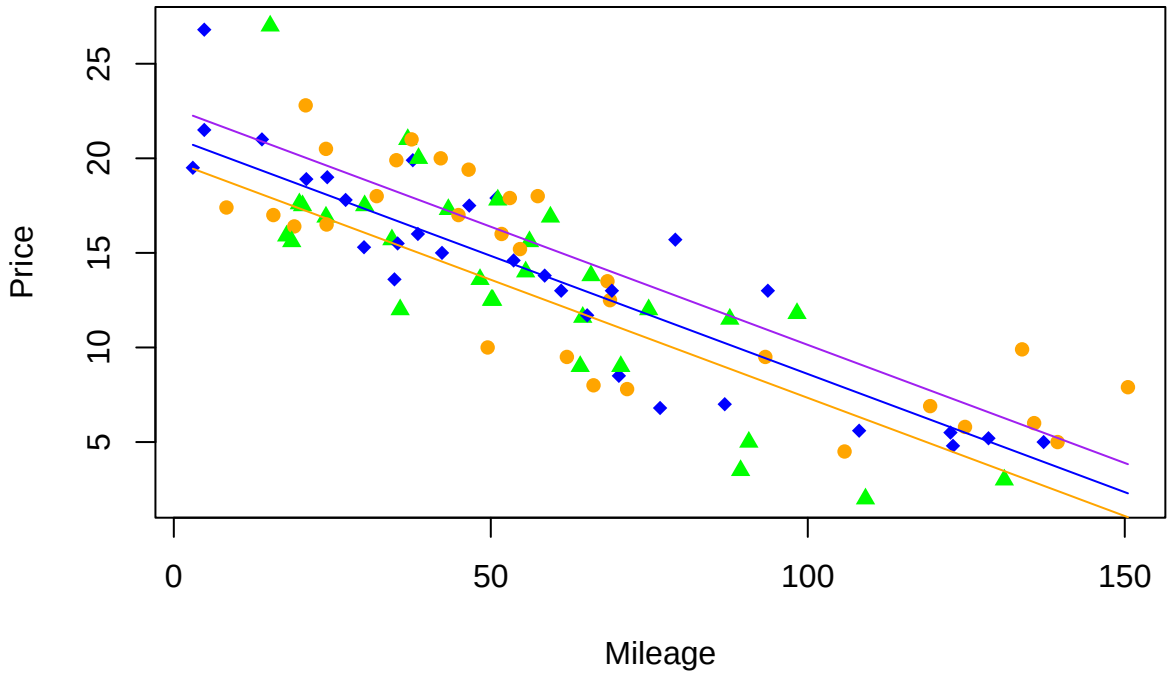
Here is another approach to Figures 4.12 and 4.13 that uses the TeachingDemos package:

First the parallel lines

```
plot(Price ~ Mileage, pch=c(17,16,18), col=c("green","orange","blue"), data=ThreeCars2017)
library(TeachingDemos)
```

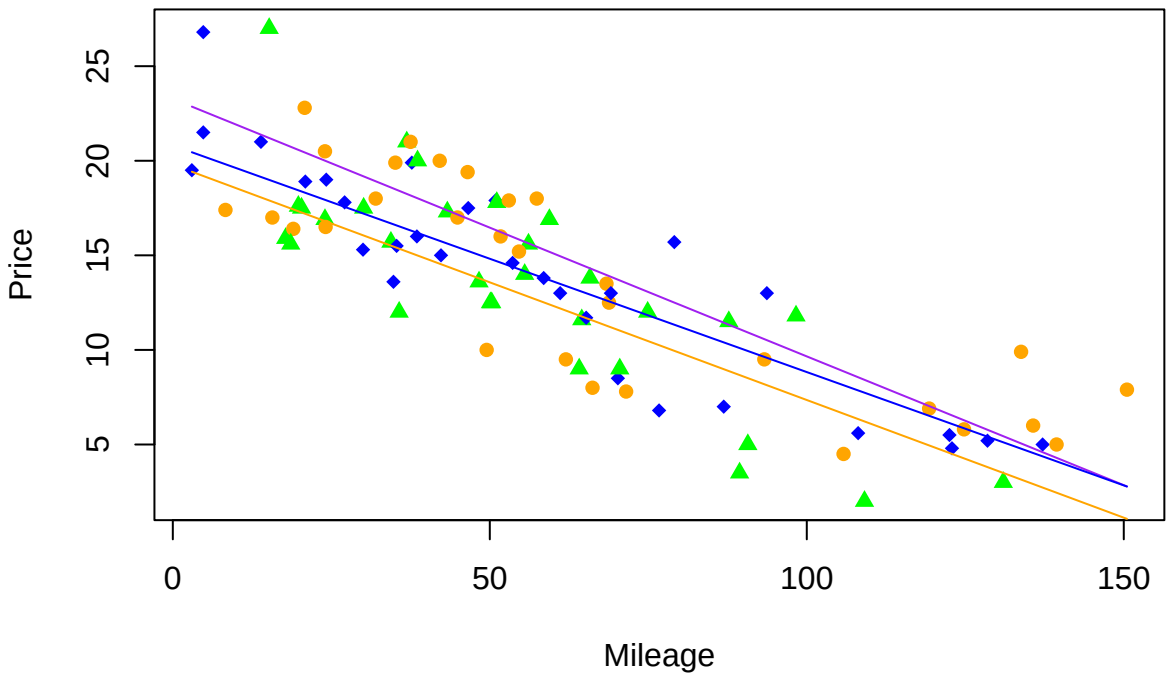
```
## Warning: package 'TeachingDemos' was built under R version 4.2.3
```

```
Predict.Plot(modMilesTypea,pred.var="Mileage",CarType="Mazda6",plot.args=list(col="orange"),add=TRUE)
Predict.Plot(modMilesTypea,pred.var="Mileage",CarType="Maxima",plot.args=list(col="purple"),add=TRUE)
Predict.Plot(modMilesTypea,pred.var="Mileage",CarType="Accord",plot.args=list(col="blue"),add=TRUE)
```



Then with different slopes (again, using TeachingDemos package)

```
plot(Price ~ Mileage, pch=c(17,16,18), col=c("green","orange","blue"), data=ThreeCars2017)
Predict.Plot(mod3slopesa,pred.var="Mileage",CarType="Mazda6",plot.args=list(col="orange"),add=TRUE)
Predict.Plot(mod3slopesa,pred.var="Mileage",CarType="Maxima",plot.args=list(col="purple"),add=TRUE)
Predict.Plot(mod3slopesa,pred.var="Mileage",CarType="Accord",plot.args=list(col="blue"),add=TRUE)
```


Price
5
10
15
20
25
0
50
100
150
Mileage