Porsche prices

Chris Parrish

January 7, 2016

Porsche prices

references:
- Cannon, et al., Stat2, chapter 01, examples 1.1-1.5
- Cannon, et al., Stat2, chapter 02
- Porsche

Import the data.

```
data <- read.csv("PorschePrice.csv", header=TRUE)
head(data, 3)
```

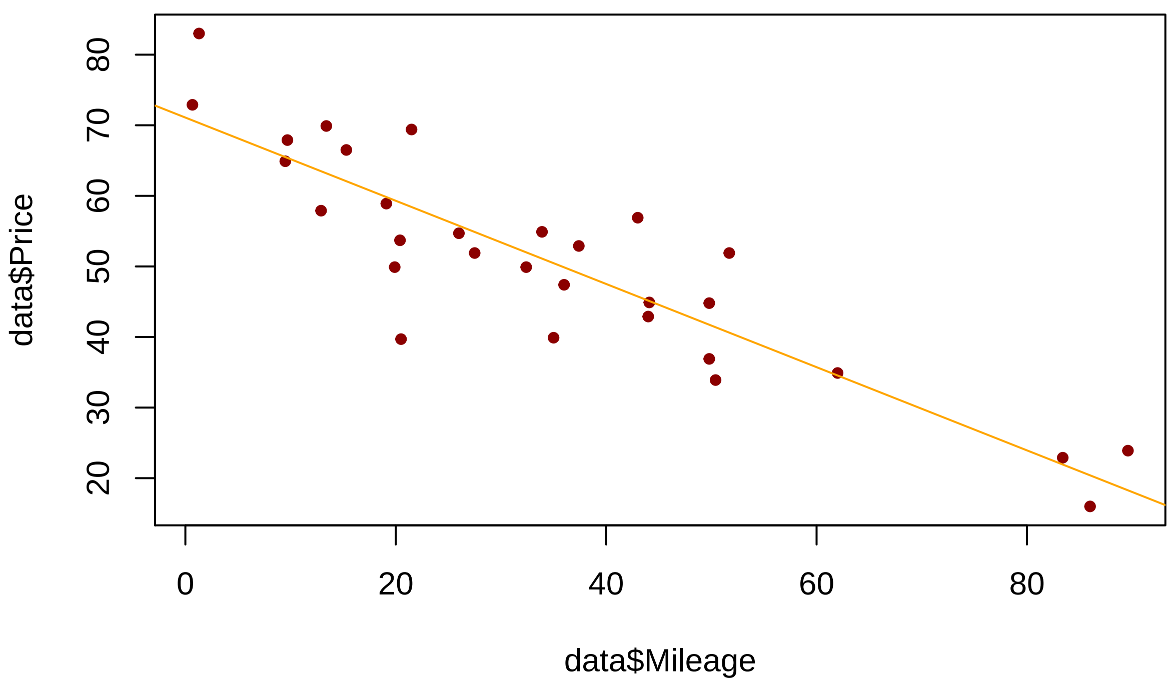
```
##   Price Age Mileage
## 1  69.4   3    21.5
## 2  56.9   3    43.0
## 3  49.9   2    19.9
```

```
dim(data)
```

```
## [1] 30  3
```

View the data.

```
plot(data$Mileage, data$Price,
     pch=20, col="darkred")
Porsche.lm <- lm(Price ~ Mileage, data=data)
abline(Porsche.lm, col="orange")
```



Linear model.

$\widehat{price} = 71.09 + \text{-}0.589\ mileage$

```
options(show.signif.stars=FALSE)
summary(Porsche.lm)
```

```
##
## Call:
## lm(formula = Price ~ Mileage, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.3077  -4.0470  -0.3945   3.8374  12.6758
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 71.09045    2.36986    30.0  < 2e-16
## Mileage     -0.58940    0.05665   -10.4 3.98e-11
##
## Residual standard error: 7.17 on 28 degrees of freedom
## Multiple R-squared:  0.7945, Adjusted R-squared:  0.7872
## F-statistic: 108.3 on 1 and 28 DF,  p-value: 3.982e-11
```

```
anova(Porsche.lm)
```

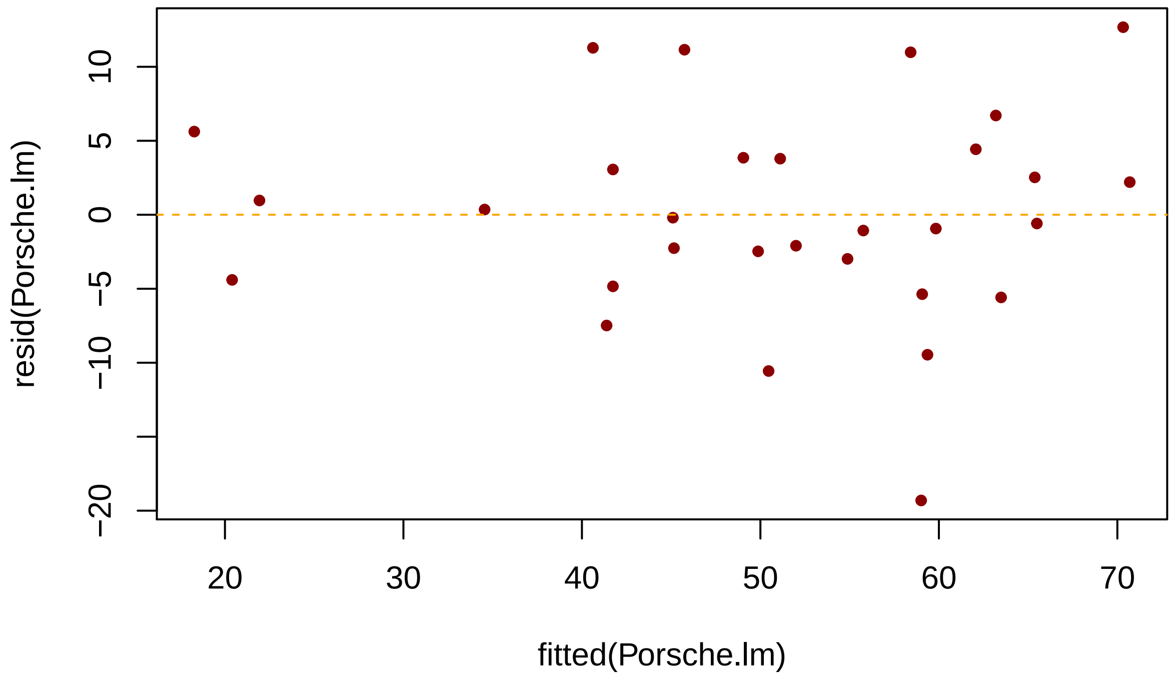
```
## Analysis of Variance Table
##
## Response: Price
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Mileage    1 5565.7  5565.7  108.25 3.982e-11
## Residuals 28 1439.6    51.4
```

Regression (= residual) standard error.

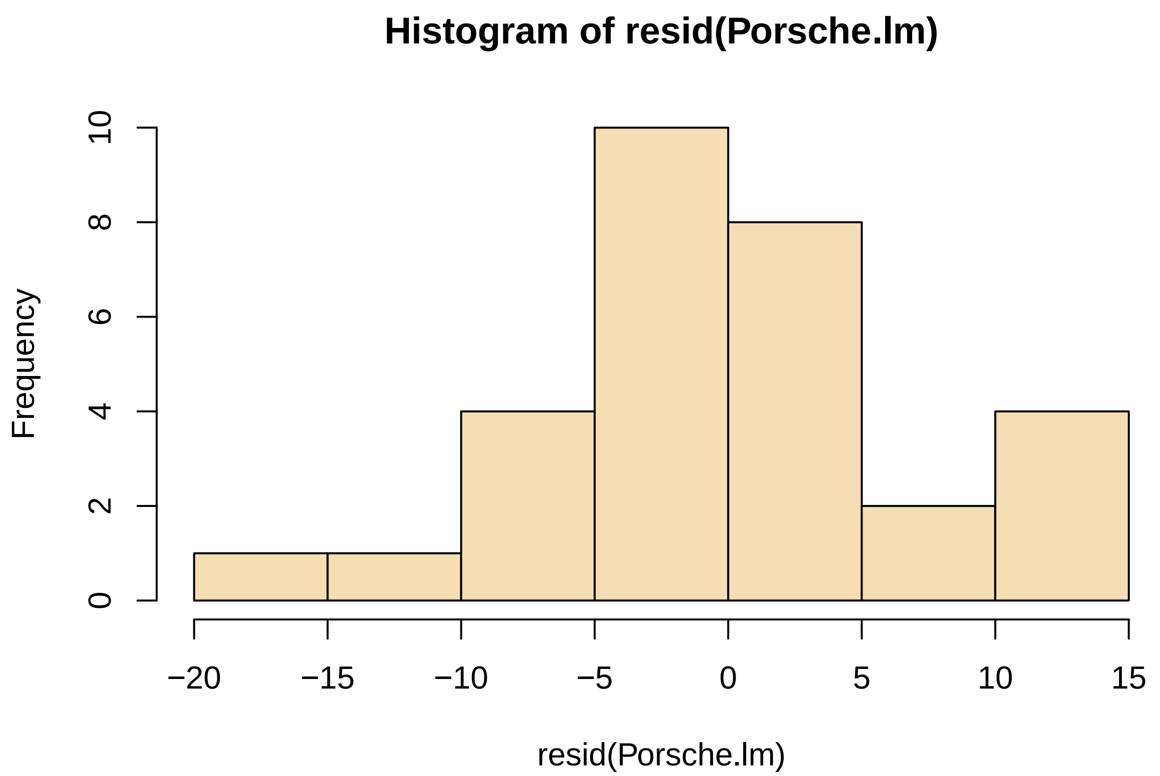
$\widehat{\sigma_e} = \sqrt{MSE} = 7.169$

Residuals.

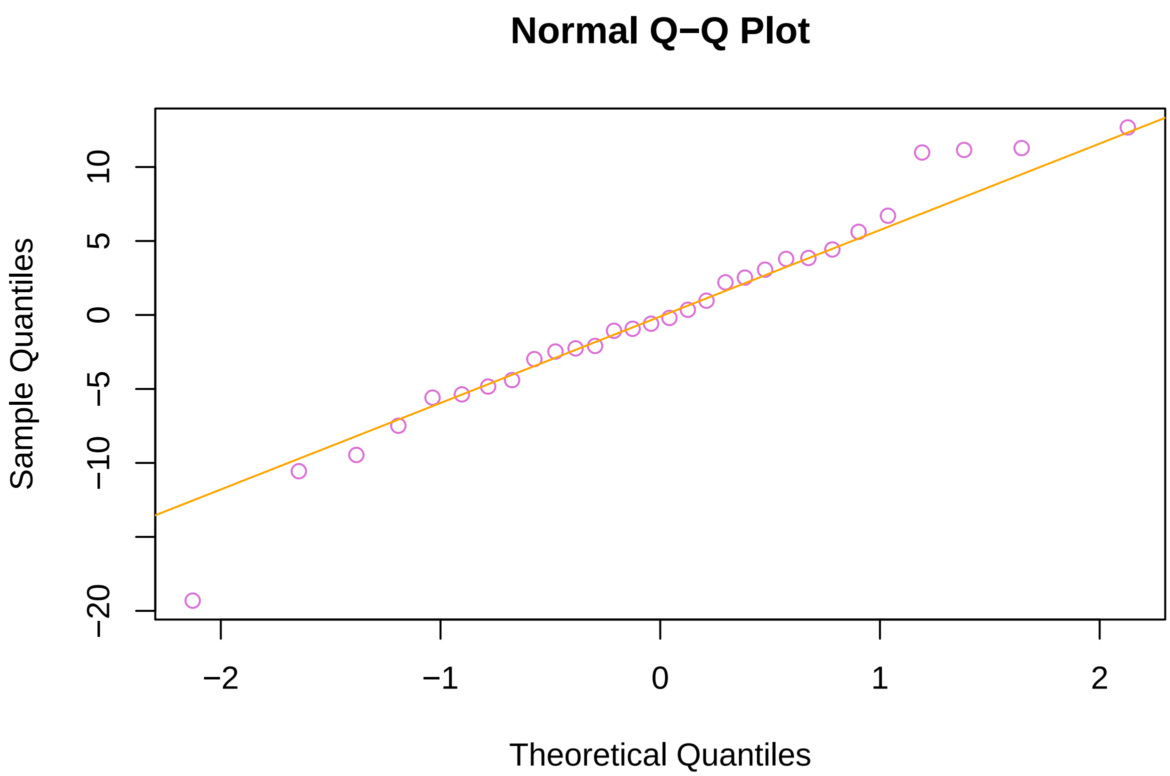
```
plot(fitted(Porsche.lm), resid(Porsche.lm),
     pch=20, col="darkred")
abline(h=0, col="orange", lty="dashed")
```




```r
hist(resid(Porsche.lm), col="wheat")
```




```r
qqnorm(resid(Porsche.lm),
       col="orchid")
qqline(resid(Porsche.lm), col="orange")
```



Prediction.

```
new.data <- data.frame(Mileage=50)
predict(Porsche.lm, new.data)
```

```
##        1
## 41.62041
```

T-test for slope of simple linear model.

$$t = \frac{\hat{\beta}_1}{SE_{\hat{\beta}_1}}, \quad df = n - 2$$

```
summary(Porsche.lm)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 71.0904527 2.36985805  29.99777 7.872391e-23
## Mileage     -0.5894009 0.05664847 -10.40453 3.981734e-11
```

CI for slope of simple linear model.

$$\hat{\beta}_1 \pm t^* \cdot SE_{\hat{\beta}_1}$$

```r
beta.hat <- -0.5894009
alpha <- 0.05
n <- 30
t.star <- qt(c(alpha/2, 1 - alpha/2), df=n-2)
se <- 0.05664847
ci <- beta.hat + t.star * se
ci
```

```
## [1] -0.7054400 -0.4733618
```

ANOVA test for simple linear regression.

$$F = \frac{MSModel}{MSE}, \quad df_1 = 1, \quad df_2 = n - 2$$

```r
anova(Porsche.lm)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Mileage    1 5565.7  5565.7  108.25 3.982e-11
## Residuals 28 1439.6    51.4
```

Coefficient of determination.

$$r^2 = \frac{SSModel}{SSE}$$

```r
summary(Porsche.lm)$r.squared
```

```
## [1] 0.794502
```

Inference for correlation.

$$\hat{\beta}_1 = r \cdot \frac{s_Y}{s_X}$$

```r
r <- with(data,
          cor(Price, Mileage))
r
```

```
## [1] -0.8913484
```

```r
r^2
```

```
## [1] 0.794502
```

T-test for correlation.

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}, \quad df = n - 2$$

```r
t <- r * sqrt(n - 2) / sqrt(1 - r^2)
t
```

```
## [1] -10.40453
```

CI for simple linear regression response.

$$\hat{y} \pm t^* \cdot SE_{\hat{\mu}}$$

$$SE_{\hat{\mu}} = \hat{\sigma}_\epsilon \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum (x - \bar{x})^2}}$$

```r
new.data <- data.frame(Mileage=50)
predict(Porsche.lm, new.data, interval="confidence")
```

```
##        fit      lwr      upr
## 1 41.62041 38.41535 44.82546
```

PI for simple linear regression response.

$$\hat{y} \pm t^* \cdot SE_{\hat{y}}$$

$$SE_{\hat{y}} = \hat{\sigma}_\epsilon \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum (x - \bar{x})^2}}$$

```r
new.data <- data.frame(Mileage=50)
predict(Porsche.lm, new.data, interval="prediction")
```

```
##        fit      lwr     upr
## 1 41.62041 26.58711 56.6537
```

Illustration.

Function `predict.plots` from Cannon, et al., R Student Manual (slightly modified).

```r
predict.plots <- function(x, y, xlab, ylab, conf.level=.95) {
  # x = explanatory variable;
  # y = response variable.
  model <- lm(y~x)
  new <- seq(min(x), max(x), length=101)
  CI <- predict(model, list(x = new), int="confidence", level=conf.level)
  PI <- predict(model, list(x = new), int="prediction", level=conf.level)
  plot(x, y, ylim=range(y, PI[, 3]),
       las=1, pch=20, col="darkred",
       xlab=xlab, ylab=ylab)
  abline(model, col="orange")                       # to obtain solid regression line
  points(new, CI[ , 2], type="l", col=2, lty=2)
  points(new, CI[ , 3], type="l", col=2, lty=2)
```

```r
  points(new, PI[ , 2], type="l", col=3, lty=3)
  points(new, PI[ , 3], type="l", col=3, lty=3)
  legend(x="topright", legend=c("Regression", "95% CI", "95% PI"),
         lty=1:3, col=c("orange", 2, 3))
}
```

```r
with(data,
     predict.plots(Mileage, Price,
                   "Mileage", "Price"))
```