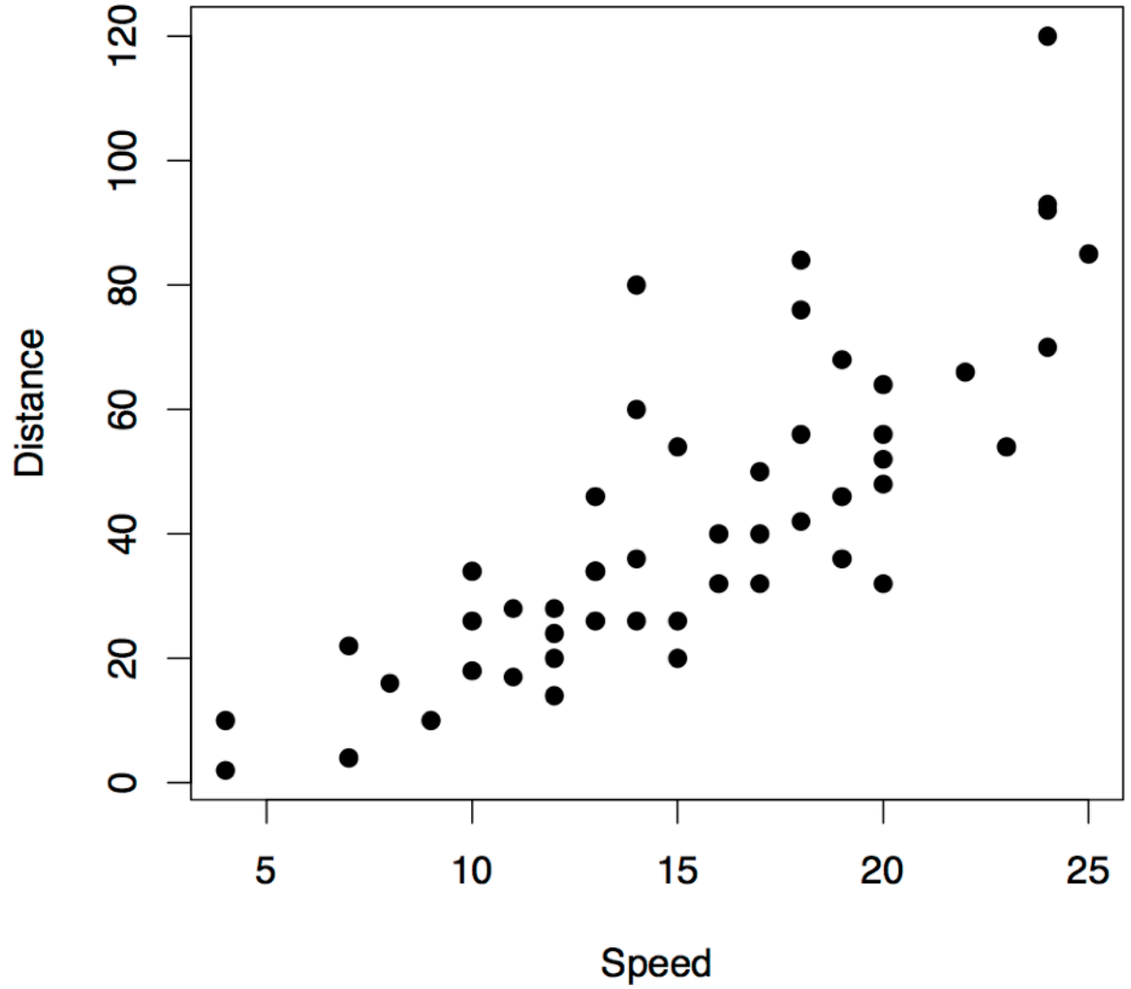
Linear Regression

Simple linear regression

$$y = \beta_0 + \beta_1 x + \epsilon$$

Example: Speed of cars and distances taken to stop (data recorded in the 1920s; 50 observations.



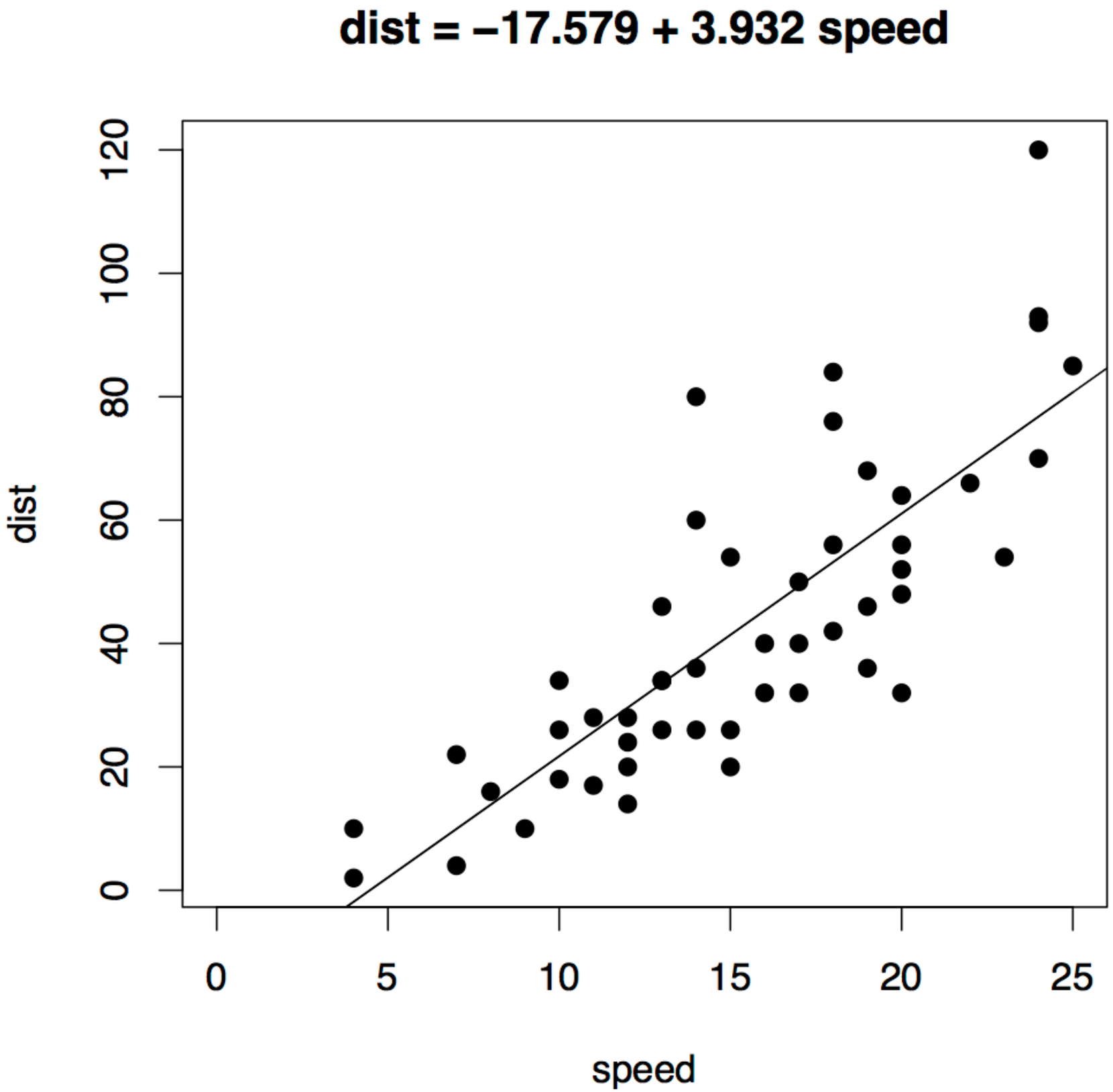
Linear Regression

```
> L1 = lm(dist ~ speed)
> print(L1)

Call:
lm(formula = dist ~ speed)

Coefficients:
(Intercept)        speed
    -17.579        3.932
```

```
> names(L1)
 [1] "coefficients"  "residuals"     "effects"        "rank"
 [5] "fitted.values" "assign"        "qr"             "df.residual"
 [9] "xlevels"       "call"          "terms"          "model"
```

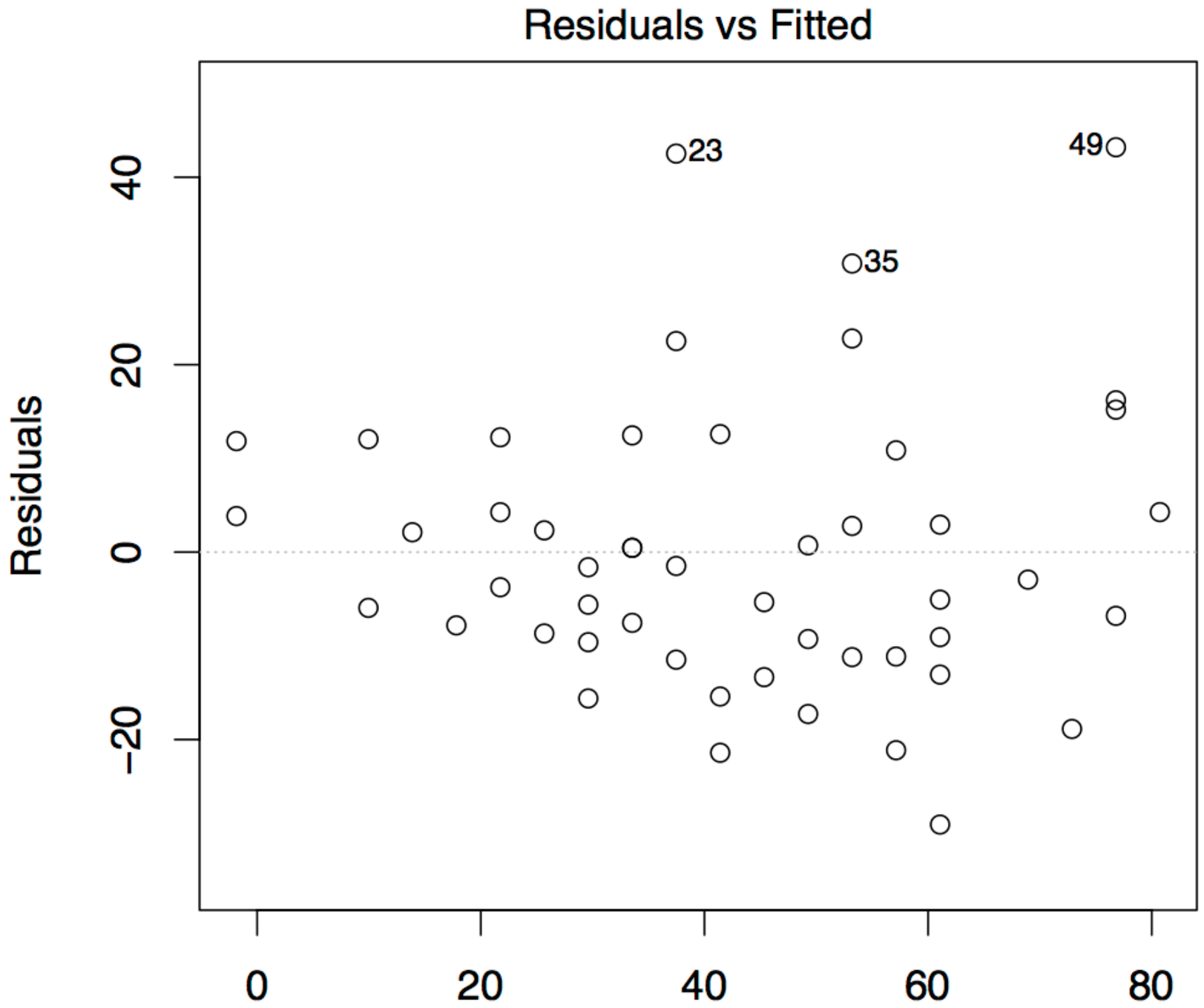
```
> plot(cars, main="dist = -17.579 + 3.932 speed",pch=19,
+ xlim=c(0, 25))
> abline(-17.579, 3.932)
> curve(-17.579 + 3.932*x, add=TRUE) #same thing
```

Linear Regression



Linear Regression

Residuals vs. fitted values:

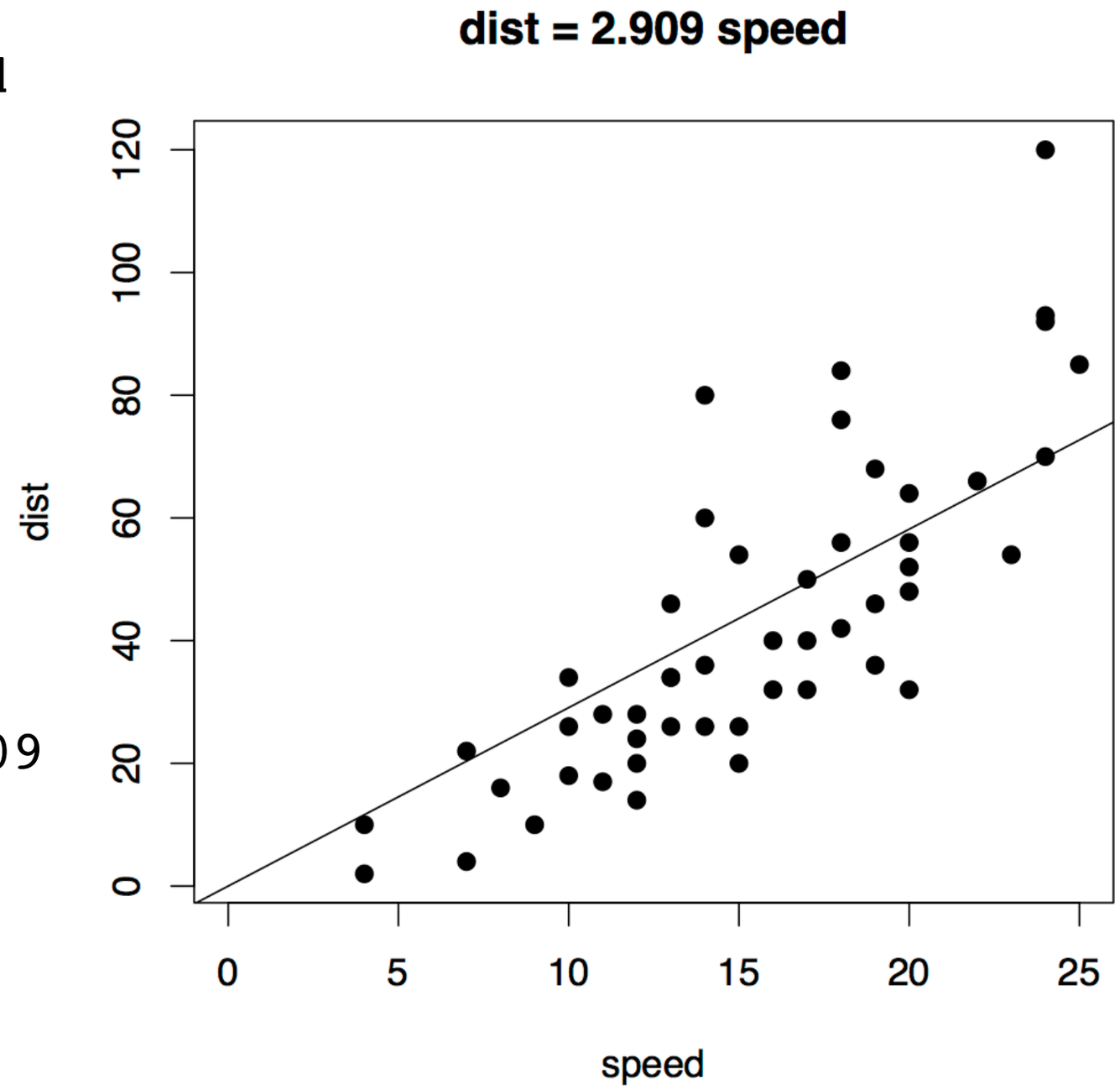


Linear Regression

Model with no intercept:

```
> L2 = lm(dist ~ 0 + speed)
> #same as L2=lm(dist ~ speed -1)
> L2
Call:
lm(formula = dist ~ 0 + speed

Coefficients:
speed
 2.909
```

```
> plot(cars, main="dist = 2.909
speed", pch=19, xlim=c(0,25))
> abline(0, L2$coeff[1])
```



Linear Regression

```
> summary(L1)

Call:
lm(formula = dist ~ speed)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Linear Regression

```
> SSxx=sum((speed-mean(speed))^2)
> SSyy=sum((dist-mean(dist))^2)
> SSxy=sum((speed-mean(speed))*(dist-mean(dist)))
> beta1_hat=SSxy/SSxx
> beta0_hat=mean(dist)-beta1_hat*mean(speed)
> beta0_hat
[1] -17.57909
> beta1_hat
[1] 3.932409
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601    0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Linear Regression

```
> SSE=sum((L1$residuals)^2)
> sigma2_hat=SSE/(length(dist)-2)
> sigma_hat=sqrt(sigma2_hat)
> sigma_hat
[1] 15.37959

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```
> t_statistic=beta1_hat/(sigma_hat/sqrt(SSxx))
> t_statistic
[1] 9.46399

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
```

Linear Regression

```
> R2=1-(SSE/SSyy)
> R2
[1] 0.6510794

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Prediction

```
> new=data.frame(speed=c(7,12))
> ynew_hat=L1$coef[1]+L1$coef[2]*new
> ynew_hat
      speed
1  9.947766
2 29.609810
```

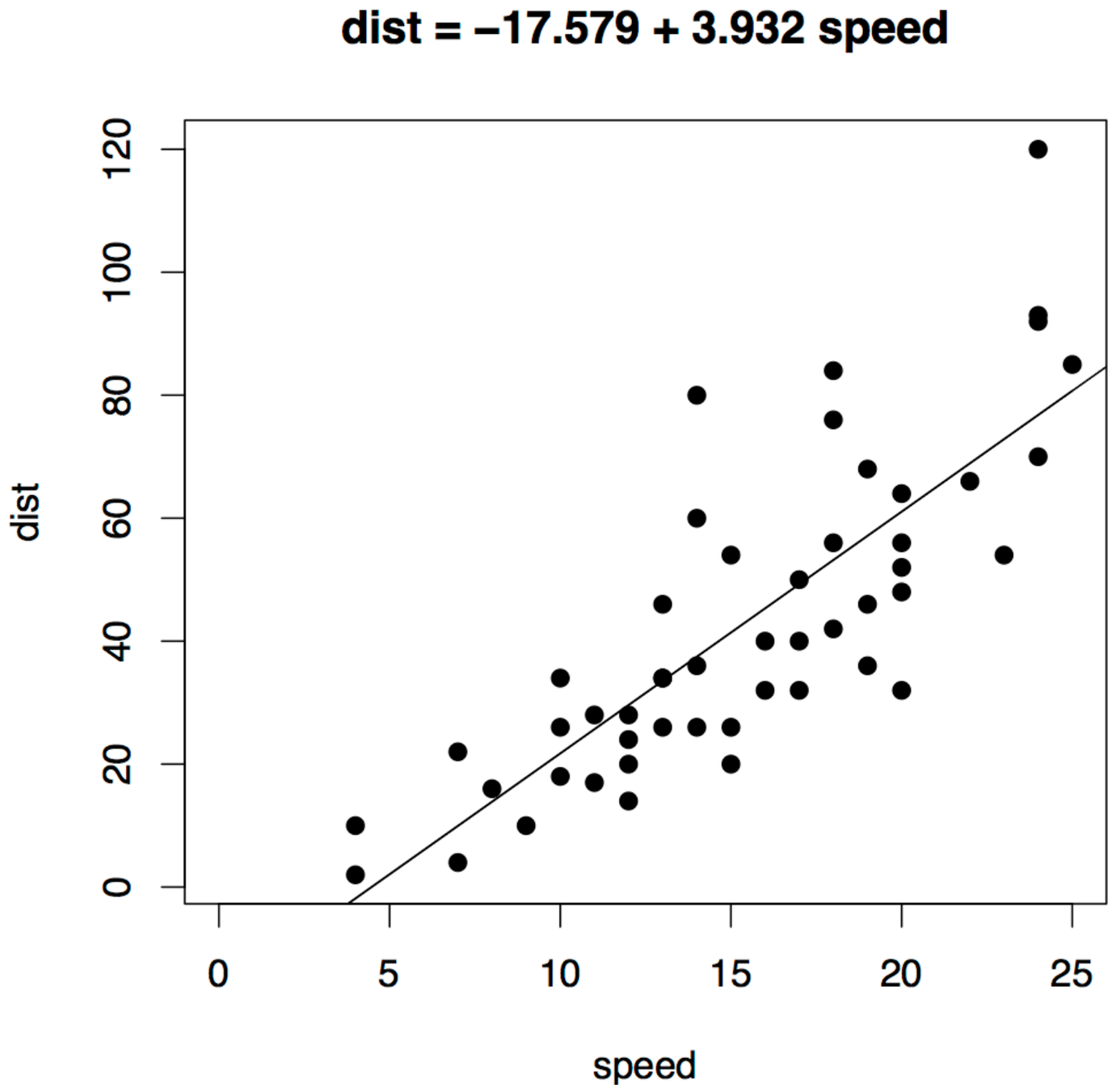
Linear Regression

```
> ynew_hat
      speed
1  9.947766
2 29.609810
> bound=qt(0.975,df=48)*sigma_hat*sqrt(1/50+(new-mean(speed))^2/
SSxx)
> low_ci=ynew_hat-bound
> up_ci=ynew_hat+bound
>
> low_ci
      speed
1  1.678977
2 24.395138
> up_ci
     speed
1 18.21656
2 34.82448
>
> predict(L1,new,interval="confidence")
       fit       lwr      upr
1  9.947766  1.678977 18.21656
2 29.609810 24.395138 34.82448
```

Linear Regression

Note that:

```
> up_ci-low_ci
      speed
1 16.53758
2 10.42935
```

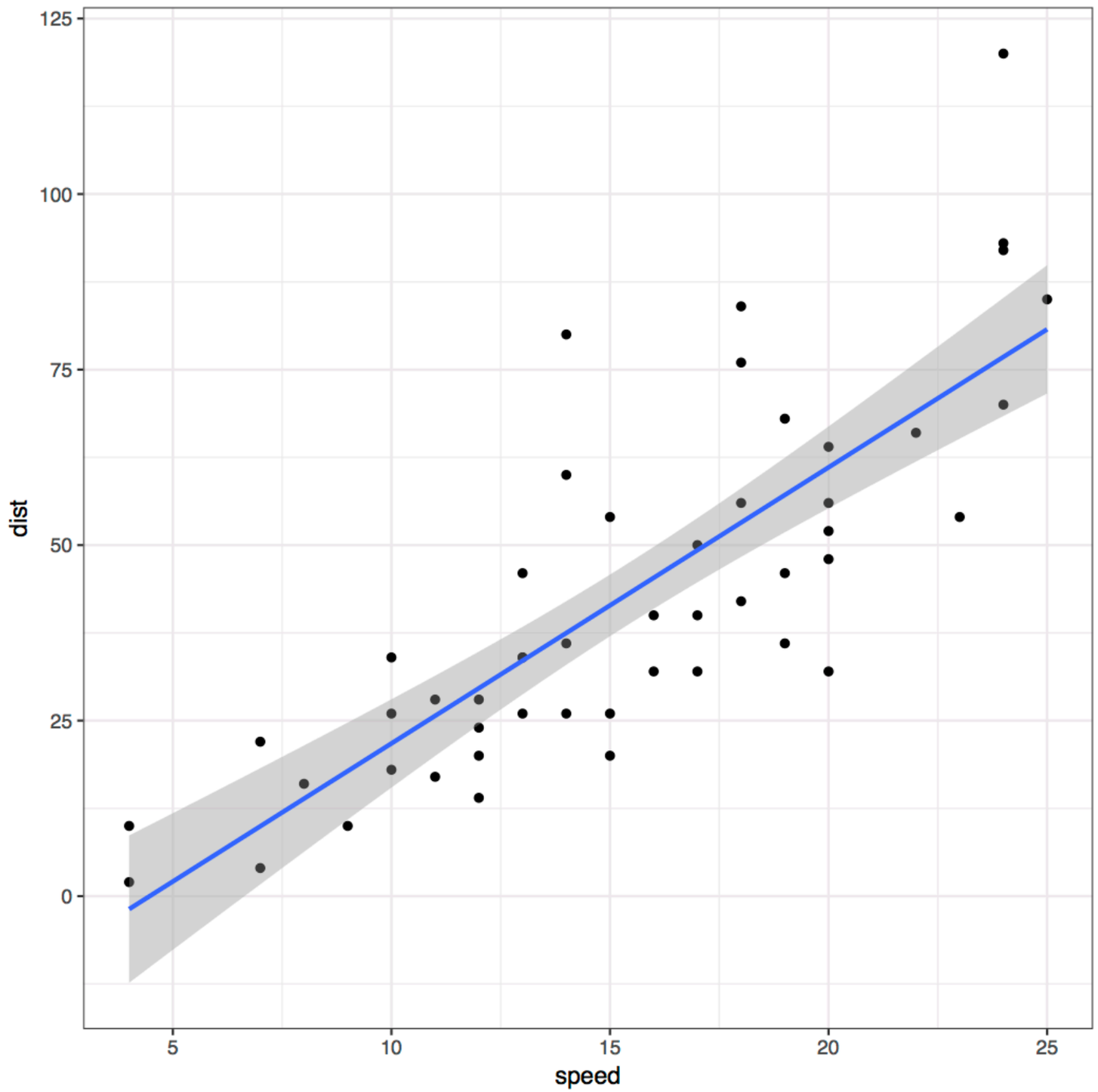


Linear Regression

```
> predict(L1,new,interval="prediction")
        fit         lwr      upr
1  9.947766 -22.061423 41.95696
2 29.609810  -1.749529 60.96915
>
> bound_predict=qt(0.975,df=48)*sigma_hat*
sqrt(1+1/50+(new-mean(speed))^2/SSxx)
>
> ynew_hat-bound_predict
      speed
1 -22.061423
2  -1.749529
> ynew_hat+bound_predict
    speed
1 41.95696
2 60.96915
```

Linear Regression

```
> ggplot(cars,aes(x=speed,y=dist)) + theme_bw()+ geom_point()
+ geom_smooth(method="lm")
```



Multiple Regression

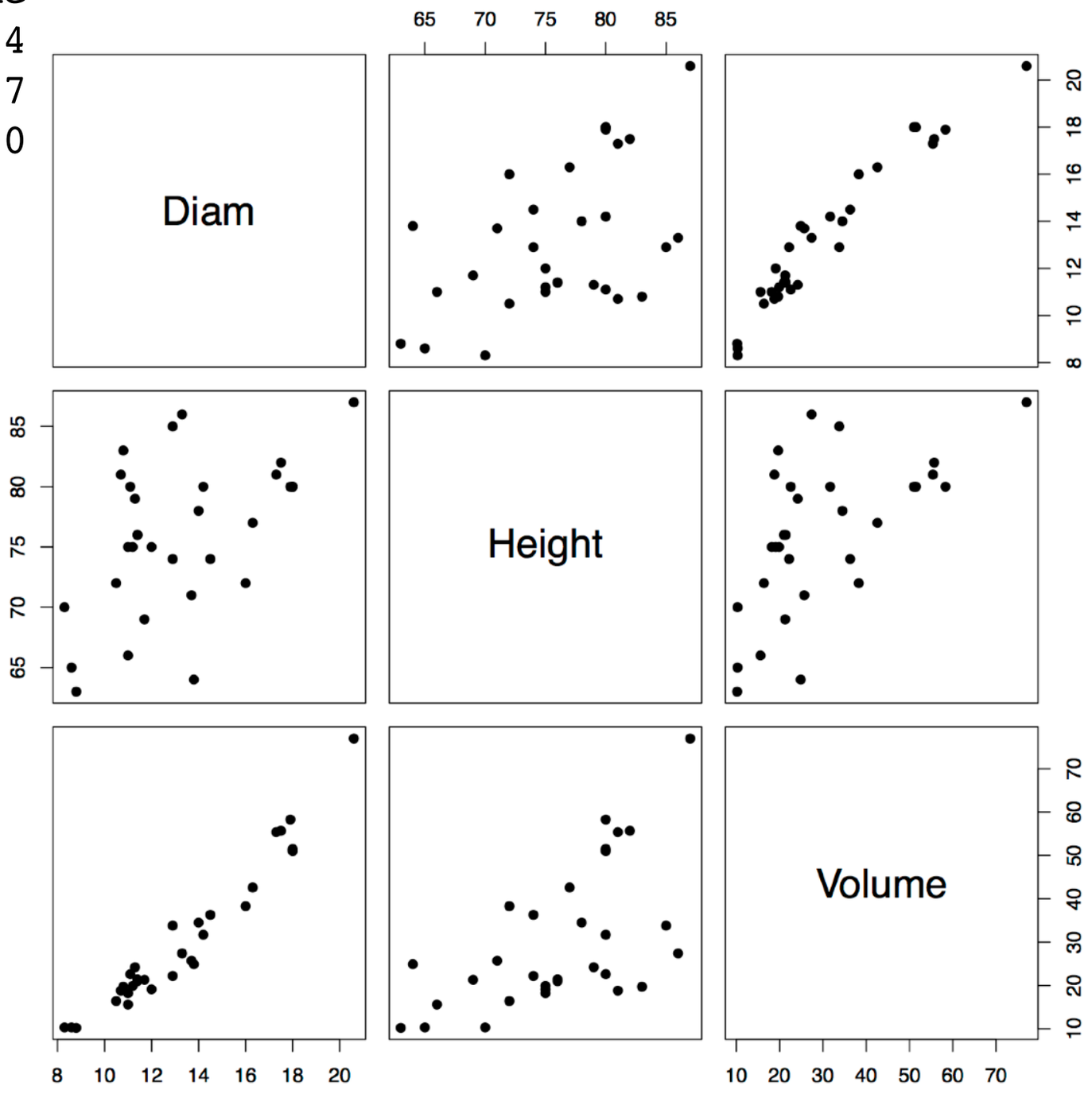
Example, Trees Data:

- Measurements of girth, height and volume of timber in 31 felled black cherry trees. Girth: diameter of the tree (in inches) measured at 4 ft 6 in above the ground

Multiple Regression

```
> cor(Trees)
             Diam    Height    Volume
Diam   1.0000000 0.5192801 0.9671194
Height 0.5192801 1.0000000 0.5982497
Volume 0.9671194 0.5982497 1.0000000
```

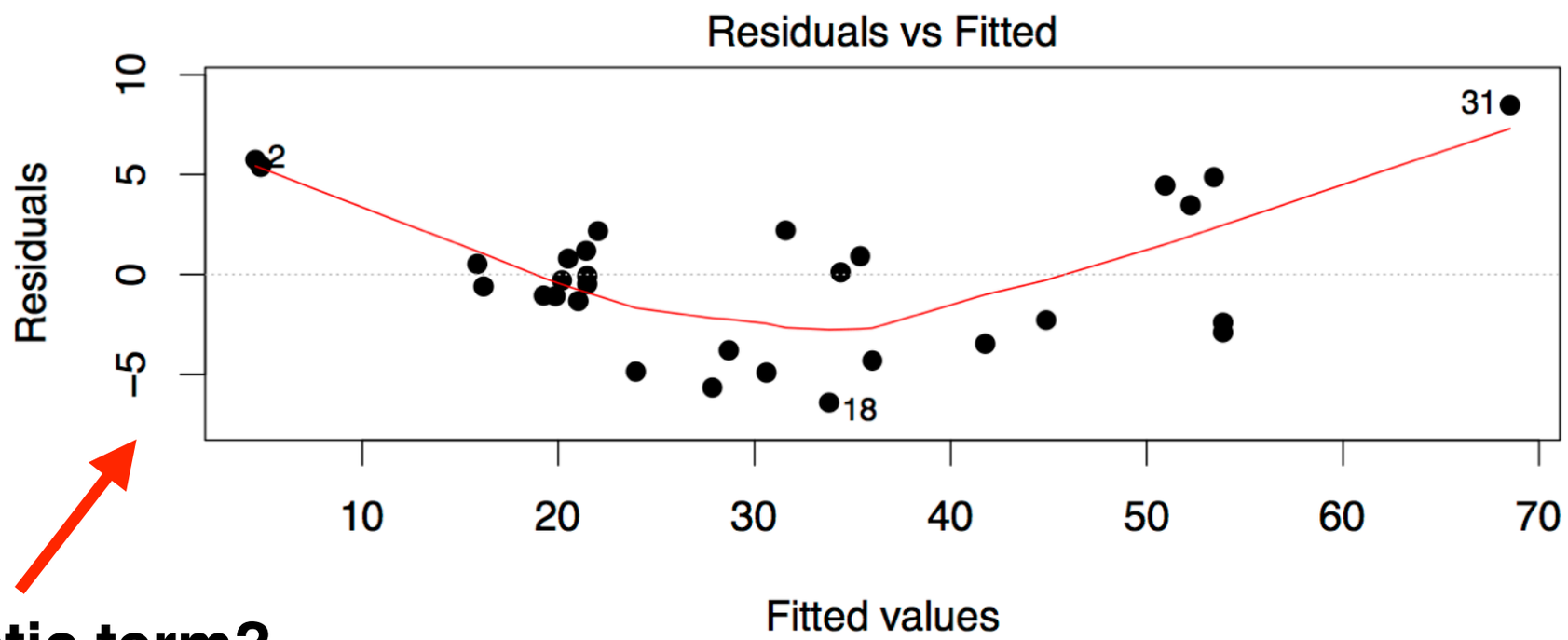
```
> M1 = lm(Volume ~ Diam)
> print(M1)

Call:
lm(formula = Volume ~ Diam)

Coefficients:
(Intercept)         Diam
    -36.943        5.066
```



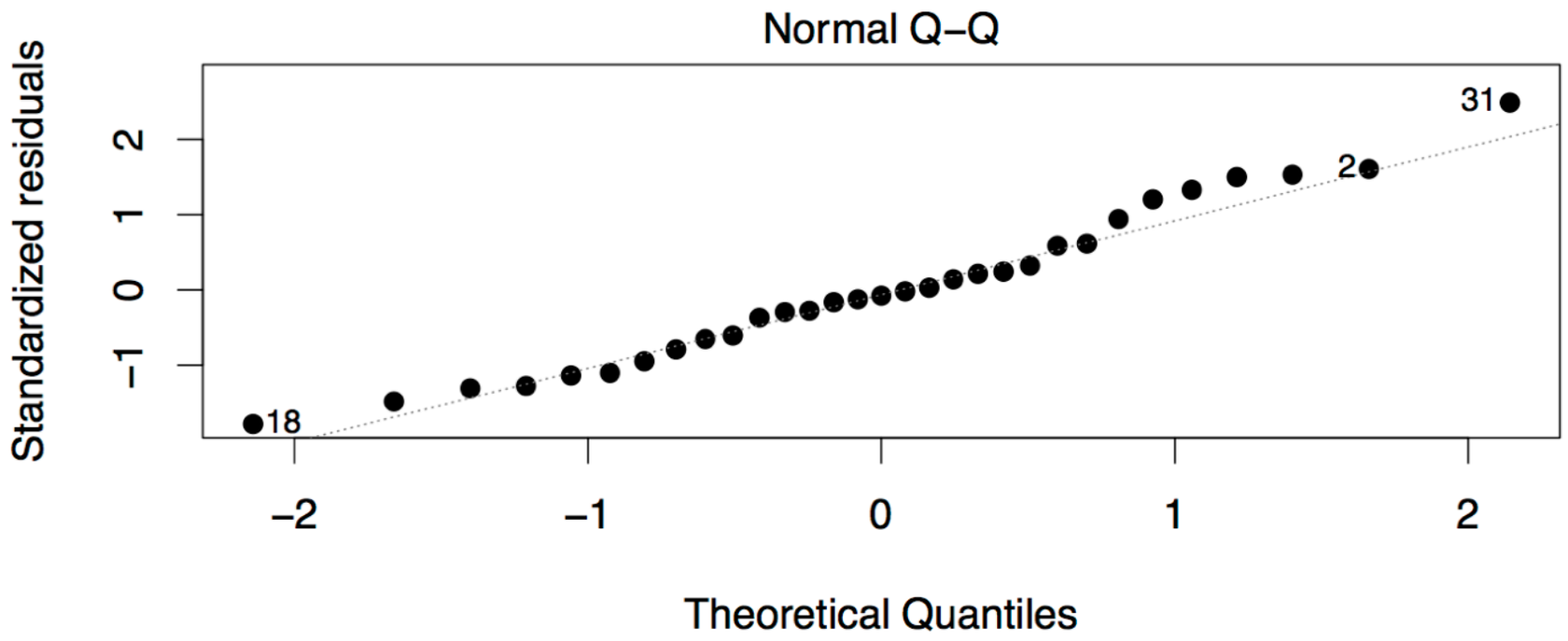
Multiple Regression

```
> M2 = lm(Volume ~ Diam + Height)
> summary(M2)

Call:
lm(formula = Volume ~ Diam + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4065 -2.6493 -0.2876  2.2003  8.4847

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
Diam          4.7082     0.2643  17.816  < 2e-16 ***
Height        0.3393     0.1302   2.607   0.0145 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.882 on 28 degrees of freedom
Multiple R-squared:  0.948,   Adjusted R-squared:  0.9442
F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

Multiple Regression



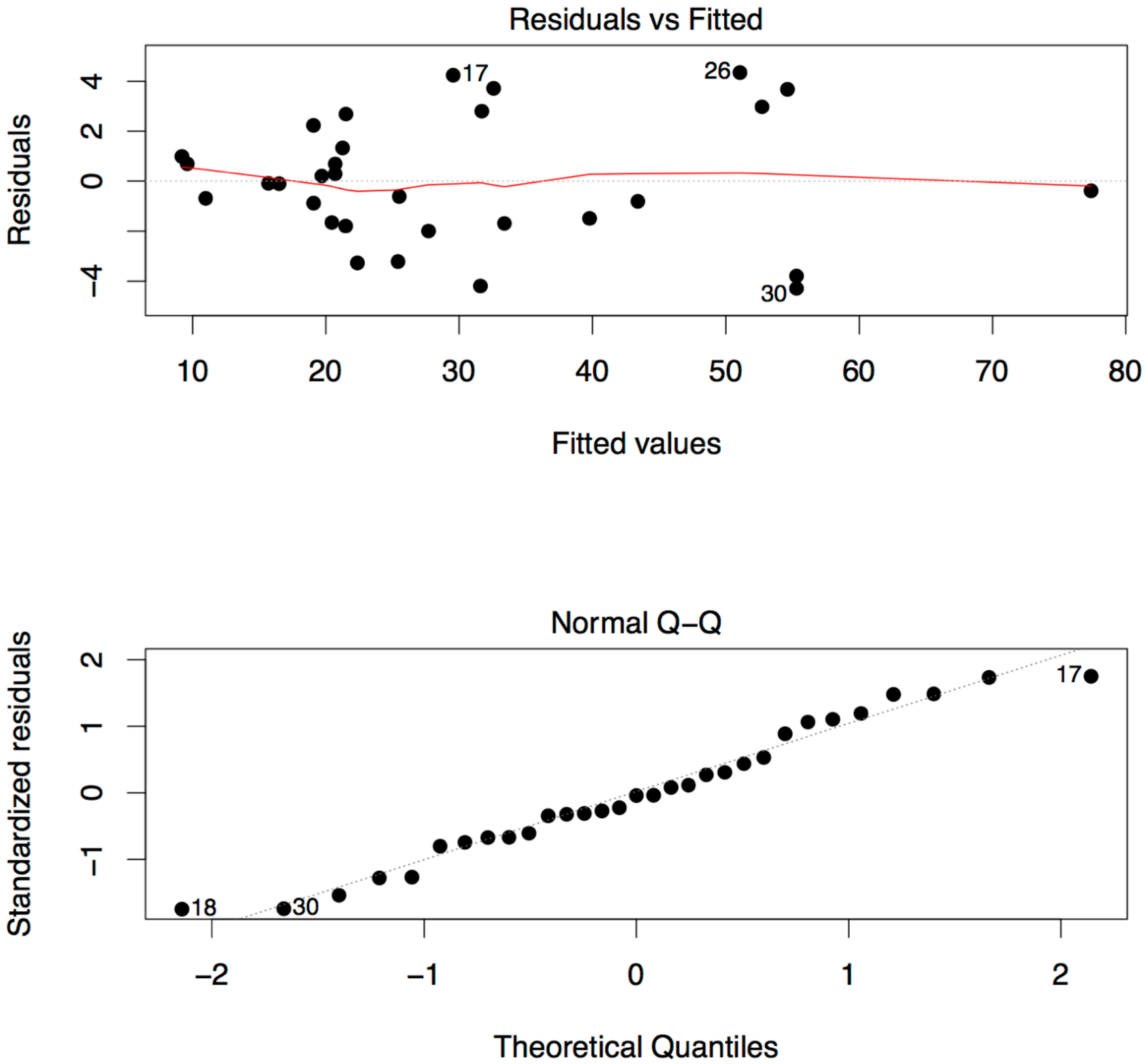
Quadratic term?



Multiple Regression

```
> M3 = lm(Volume ~ Diam + I(Diam^2) + Height)
> summary(M3)

Call:
lm(formula = Volume ~ Diam + I(Diam^2) + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2928 -1.6693 -0.1018  1.7851  4.3489

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.92041   10.07911  -0.984 0.333729
Diam        -2.88508    1.30985  -2.203 0.036343 *
I(Diam^2)    0.26862    0.04590   5.852 3.13e-06 ***
Height       0.37639    0.08823   4.266 0.000218 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.625 on 27 degrees of freedom
Multiple R-squared:  0.9771, Adjusted R-squared:  0.9745
F-statistic: 383.2 on 3 and 27 DF,  p-value: < 2.2e-16
```

Multiple Regression



Multiple Regression

```
> anova(M1)
Analysis of Variance Table
```

Compares model with diam to model with intercept

```
Response: Volume
          Df Sum Sq Mean Sq F value    Pr(>F)
Diam       1 7581.8  7581.8  419.36 < 2.2e-16 ***
Residuals 29  524.3    18.1
```

```
> anova(M2)
Analysis of Variance Table
```

Compares model with diam + height to model with diam

```
Response: Volume
          Df Sum Sq Mean Sq  F value  Pr(>F)
Diam       1 7581.8  7581.8 503.1503 < 2e-16 ***
Height     1  102.4   102.4   6.7943 0.01449 *
Residuals 28  421.9    15.1
```

$$F = \frac{(524.3 - 421.92)/1}{421.92/28} = \frac{102.38}{15.07} = 6.7943$$

Multiple Regression

Model comparison

```
> anova(M1, M2, M3)
Analysis of Variance Table

Model 1: Volume ~ Diam
Model 2: Volume ~ Diam + Height
Model 3: Volume ~ Diam + I(Diam^2) + Height
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     29 524.30
2     28 421.92  1    102.38 14.861 0.0006487 ***   M2 vs M1
3     27 186.01  1    235.91 34.243  3.13e-06 ***   M3 vs M2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Multiple Regression

- Prediction:

```
> diameter = 16
> height = seq(65, 70, 1)
> new = data.frame(Diam=diameter, Height=height)
> predict(M3, newdata=new, interval="conf")
       fit      lwr      upr
1 37.15085 34.21855 40.08315
2 37.52724 34.75160 40.30287
3 37.90362 35.28150 40.52574
4 38.28001 35.80768 40.75234
5 38.65640 36.32942 40.98338
6 39.03278 36.84581 41.21975
```