Section 3.6 Testing Subsets of Predictors

Load needed packagees.

```
library(Stat2Data)
library(mosaic)
library(ggplot2)
```

EXAMPLE 3.18 House prices: comparing models

Create a dataframe for **HousesNY** and look at the structure of the data.

```
data("HousesNY")
str(HousesNY)
```

```
## 'data.frame':    53 obs. of  5 variables:
##  $ Price: num  57.6 120 150 143 92.5 ...
##  $ Beds : int  3 6 4 3 3 2 2 4 4 3 ...
##  $ Baths: num  2 2 2 2 1 1 2 3 2.5 2 ...
##  $ Size : num  0.96 2.79 1.7 1.2 1.33 ...
##  $ Lot  : num  1.3 0.23 0.27 0.8 0.42 0.34 0.29 0.21 1 0.3 ...
```

EXAMPLE 3.18 FIT a multiple regression model with three predictors

```
model3=lm(Price~Size+Beds+Baths, data=HousesNY)
summary(model3)
```

```
##
## Call:
## lm(formula = Price ~ Size + Beds + Baths, data = HousesNY)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.361 -26.757   2.146  27.558  61.677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.929     21.566   0.970  0.33659
## Size          21.258     11.817   1.799  0.07818 .
## Beds           2.230      8.661   0.257  0.79787
## Baths         26.610      7.793   3.415  0.00129 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.88 on 49 degrees of freedom
## Multiple R-squared:  0.4066, Adjusted R-squared:  0.3702
## F-statistic: 11.19 on 3 and 49 DF,  p-value: 1.042e-05
```

```r
anova(model3)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Size       1  23407 23407.2 21.6541 2.511e-05 ***
## Beds       1    276   276.2  0.2555   0.61549
## Baths      1  12605 12605.0 11.6609   0.00129 **
## Residuals 49  52967  1081.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EXAMPLE 3.18 FIT a simple linear regression model based on Baths alone

```r
modelBaths=lm(Price~Baths, data=HousesNY)
summary(modelBaths)
```

```
##
## Call:
## lm(formula = Price ~ Baths, data = HousesNY)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -72.52 -26.20   0.30  20.30  61.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   47.069     14.648   3.213  0.00228 **
## Baths         35.815      7.453   4.806  1.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.71 on 51 degrees of freedom
## Multiple R-squared:  0.3117, Adjusted R-squared:  0.2982
## F-statistic:  23.1 on 1 and 51 DF,  p-value: 1.399e-05
```

```r
anova(modelBaths)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Baths      1  27821 27821.1  23.096 1.399e-05 ***
## Residuals 51  61434  1204.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EXAMPLE 3.18 Nested F-test

NOTE: The ANOVA tables are included in the output above just in case you want to pull out the appropriate sums of squares and degrees of freedom to check the calculations by hand. We have provided them in the text, but we will not keep including this long calculation in alternative solutions. R allows the quick code below for doing a nested F test with these two models.

```
anova(modelBaths,model3)
```

```
## Analysis of Variance Table
##
## Model 1: Price ~ Baths
## Model 2: Price ~ Size + Beds + Baths
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1     51 61434
## 2     49 52967  2    8467.2 3.9165 0.02643 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EXAMPLE 3.19 NFL winning percentage: nested F test

Create a dataframe for **NFLStandings2016** and look at the structure of the data.

```
data("NFLStandings2016")
str(NFLStandings2016)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ Team          : Factor w/ 32 levels "Arizona Cardinals",..: 20 9 16 24 2 22 26 29 12 18 ...
##  $ Wins          : int  14 13 12 12 11 11 11 10 10 10 ...
##  $ Losses        : int  2 3 4 4 5 5 5 5 6 6 ...
##  $ Ties          : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ WinPct        : num  0.875 0.813 0.75 0.75 0.688 0.688 0.688 0.656 0.625 0.625 ...
##  $ PointsFor     : int  441 421 389 416 540 310 399 354 432 363 ...
##  $ PointsAgainst: int  250 306 311 385 406 284 327 292 388 380 ...
##  $ NetPts        : int  191 115 78 31 134 26 72 62 44 -17 ...
##  $ YardsFor      : int  6179 6027 5488 5973 6653 5291 5962 5715 5901 5325 ...
##  $ YardsAgainst : int  5222 5502 5896 6002 5939 5435 5482 5099 5822 6122 ...
##  $ TDs           : int  51 49 42 47 63 36 47 37 51 45 ...
```

EXAMPLE 3.19 FIT a multiple regression model with five predictors

```
NFLStandings2016$WinPct100=NFLStandings2016$WinPct*100
NFLmodel2016five <- lm(WinPct100 ~ PointsFor + PointsAgainst + YardsFor +YardsAgainst + TDs , data=NFLS
summary(NFLmodel2016five)
```

```
##
## Call:
## lm(formula = WinPct100 ~ PointsFor + PointsAgainst + YardsFor +
##     YardsAgainst + TDs, data = NFLStandings2016)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.654  -5.308   0.513   5.956  18.759
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   48.737059  31.602990   1.542   0.1351
## PointsFor      0.157242   0.114616   1.372   0.1818
## PointsAgainst -0.310314   0.041177  -7.536 5.32e-08 ***
## YardsFor      -0.003019   0.006294  -0.480   0.6355
## YardsAgainst   0.012357   0.005465   2.261   0.0324 *
## TDs            0.115525   0.737425   0.157   0.8767
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.237 on 26 degrees of freedom
## Multiple R-squared:  0.8214, Adjusted R-squared:  0.787
## F-statistic: 23.91 on 5 and 26 DF,  p-value: 5.818e-09
```

```
anova(NFLmodel2016five)
```

```
## Analysis of Variance Table
##
## Response: WinPct100
##               Df Sum Sq Mean Sq F value    Pr(>F)
## PointsFor      1 4126.2  4126.2 48.3621 2.202e-07 ***
## PointsAgainst  1 5588.4  5588.4 65.5007 1.422e-08 ***
## YardsFor       1   23.7    23.7  0.2775    0.6028
## YardsAgainst   1  458.3   458.3  5.3716    0.0286 *
## TDs            1    2.1     2.1  0.0245    0.8767
## Residuals     26 2218.3    85.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EXAMPLE 3.19 FIT a reduced model with only PointsFor and PointsAgainst

```
NFLmodel2016reduced <- lm(WinPct100 ~ PointsAgainst + YardsAgainst , data=NFLStandings2016)
summary(NFLmodel2016reduced)
```

```
##
## Call:
## lm(formula = WinPct100 ~ PointsAgainst + YardsAgainst, data = NFLStandings2016)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.3268  -6.8885   0.1541   8.1946  26.5090
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   59.394651  32.537755   1.825  0.07826 .
## PointsAgainst -0.359417   0.056932  -6.313 6.77e-07 ***
## YardsAgainst   0.021690   0.007387   2.936  0.00644 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.24 on 29 degrees of freedom
## Multiple R-squared:  0.5904, Adjusted R-squared:  0.5622
## F-statistic:  20.9 on 2 and 29 DF,  p-value: 2.394e-06
```

```
anova(NFLmodel2016reduced)
```

```
## Analysis of Variance Table
##
## Response: WinPct100
##               Df Sum Sq Mean Sq F value    Pr(>F)
## PointsAgainst  1 5818.9  5818.9 33.1795 3.081e-06 ***
## YardsAgainst   1 1512.0  1512.0  8.6215  0.006443 **
## Residuals     29 5085.9   175.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EXAMPLE 3.19 USE the nested F test to compare the full and reduced models

```
anova(NFLmodel2016reduced, NFLmodel2016five)
```

```
## Analysis of Variance Table
##
## Model 1: WinPct100 ~ PointsAgainst + YardsAgainst
## Model 2: WinPct100 ~ PointsFor + PointsAgainst + YardsFor + YardsAgainst +
##     TDs
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     29 5085.9
## 2     26 2218.3  3    2867.7 11.204 6.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EXAMPLE 3.20 Perch weights revisited

Create a dataframe for **Perch** and look at the structure of the data.

```
data("Perch")
str(Perch)
```

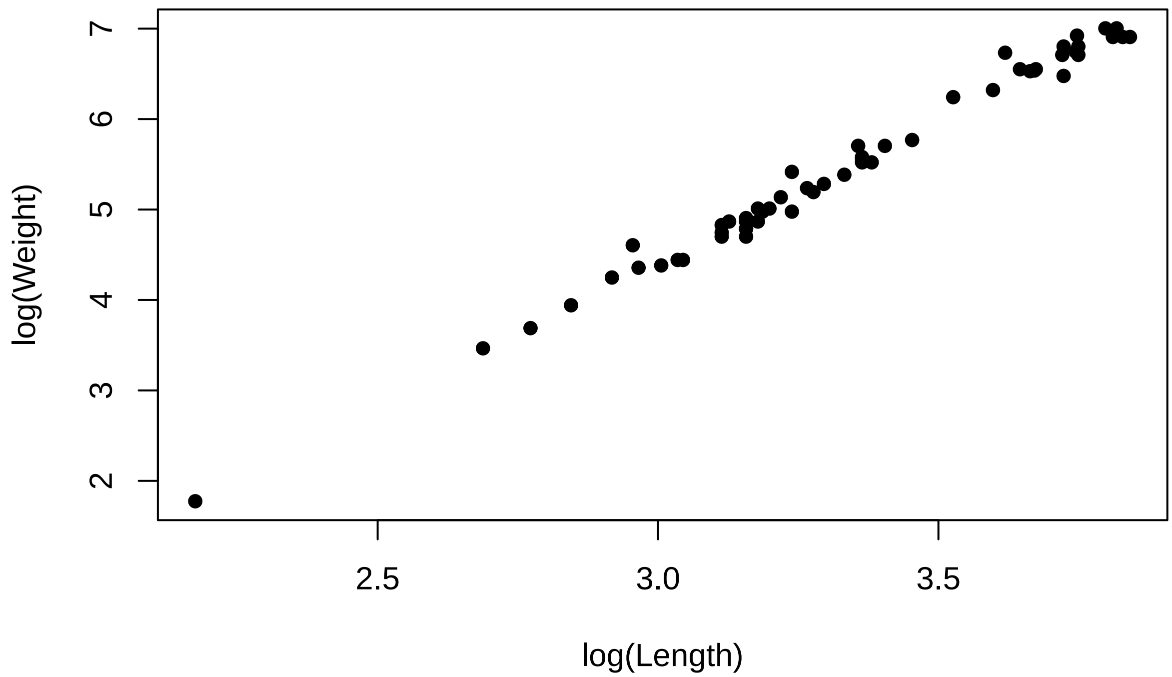
```
## 'data.frame':    56 obs. of  4 variables:
##  $ Obs   : int  104 105 106 107 108 109 110 111 112 113 ...
##  $ Weight: num  5.9 32 40 51.5 70 100 78 80 85 85 ...
##  $ Length: num  8.8 14.7 16 17.2 18.5 19.2 19.4 20.2 20.8 21 ...
##  $ Width : num  1.4 2 2.4 2.6 2.9 3.3 3.1 3.1 3 2.8 ...
```

FIGURE 3.31 Individual predictors for perch weights

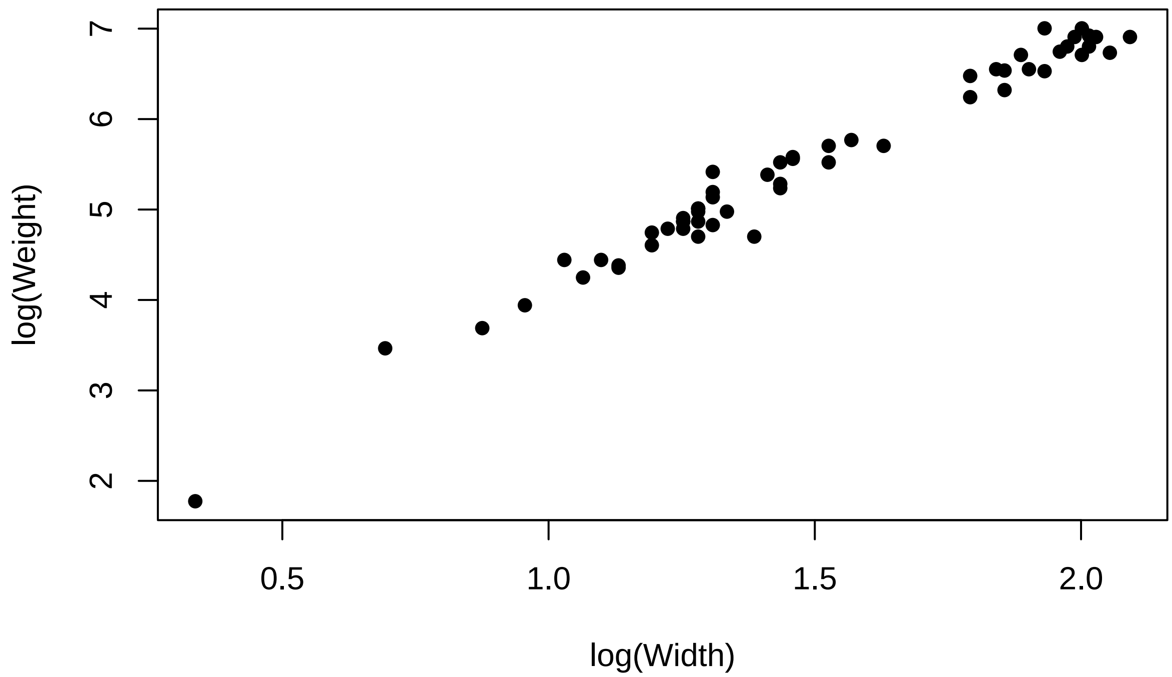
Create log transformations and construct plots

```
Perch$LogWeight=log(Perch$Weight)
Perch$LogLength=log(Perch$Length)
Perch$LogWidth=log(Perch$Width)

plot(log(Weight)~log(Length),data=Perch,pch=16)
```




```
plot(log(Weight)~log(Width),data=Perch,pch=16)
```



EXAMPLE 3.20 FIT the model on log scale

```
modlog=lm(log(Weight)~log(Length)+log(Width),data=Perch)
summary(modlog)
```

```
##
## Call:
## lm(formula = log(Weight) ~ log(Length) + log(Width), data = Perch)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.289703 -0.044798  0.003553  0.049611  0.313656
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.1509     0.4090  -7.704 3.33e-10 ***
## log(Length)   2.1993     0.1979  11.111 1.86e-15 ***
## log(Width)    0.8642     0.1743   4.959 7.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09651 on 53 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9922
## F-statistic:  3496 on 2 and 53 DF,  p-value: < 2.2e-16
```

FIGURE 3.32 Residual plot of perch model that uses logs

```r
plot(modlog$residuals~modlog$fitted,xlab="Fitted Values",ylab="Residuals",pch=16)
abline(0,0)
```