Section 3.1 Multiple Linear Regression

Load needed packages.

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

Create a dataframe for **NFLStandings2016** data 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 3.1 NFL winning percentage

FIGURE 3.1 Scatterplot matrix for NFL winning percentages and scoring

We will use only columns 5, 6, and 7 for this scatterplot matrix. You can also create a scatterplot matrix for all variables in a dataframe by specifying only the dataframe name without any column references.

```
pairs(NFLStandings2016[c(5:7)],pch=16)
```

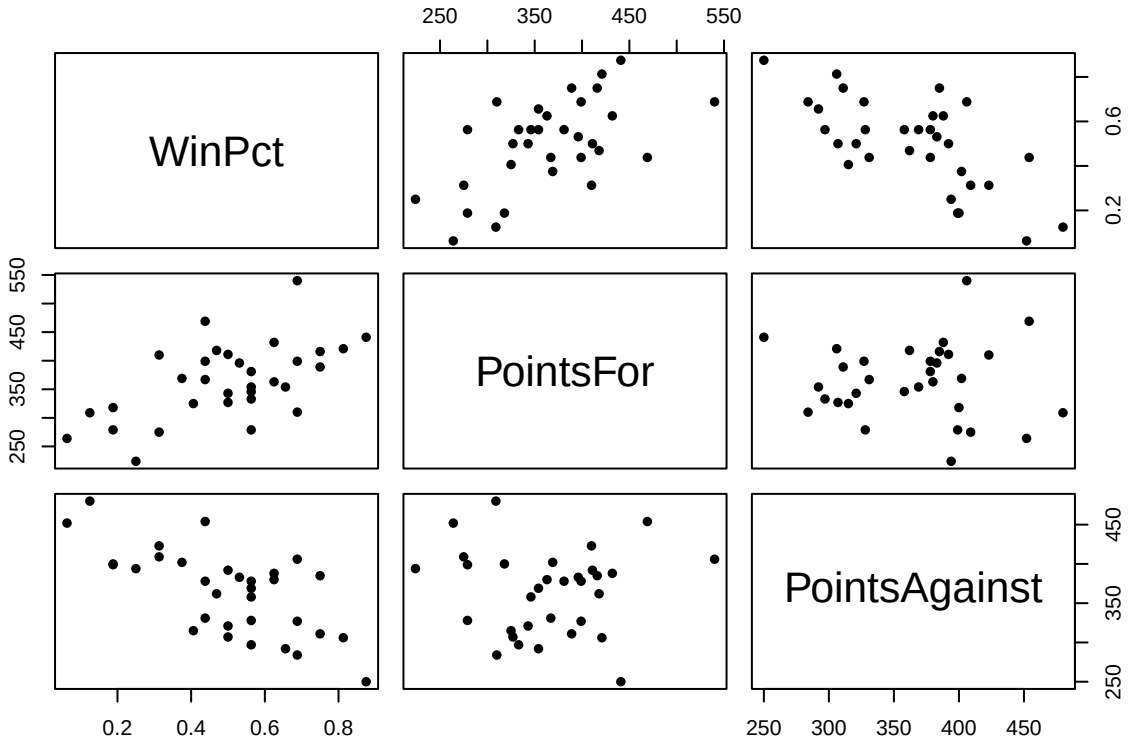


TABLE 3.1 Records and points for NFL teams in 2016 season (only the first six rows)

```
head(NFLStandings2016)
```

```
##                  Team Wins Losses Ties WinPct PointsFor PointsAgainst NetPts
## 1 New England Patriots   14      2    0  0.875       441           250    191
## 2       Dallas Cowboys   13      3    0  0.813       421           306    115
## 3   Kansas City Chiefs   12      4    0  0.750       389           311     78
## 4      Oakland Raiders   12      4    0  0.750       416           385     31
## 5      Atlanta Falcons   11      5    0  0.688       540           406    134
## 6      New York Giants   11      5    0  0.688       310           284     26
##   YardsFor YardsAgainst TDs
## 1     6179         5222  51
## 2     6027         5502  49
## 3     5488         5896  42
## 4     5973         6002  47
## 5     6653         5939  63
## 6     5291         5435  36
```

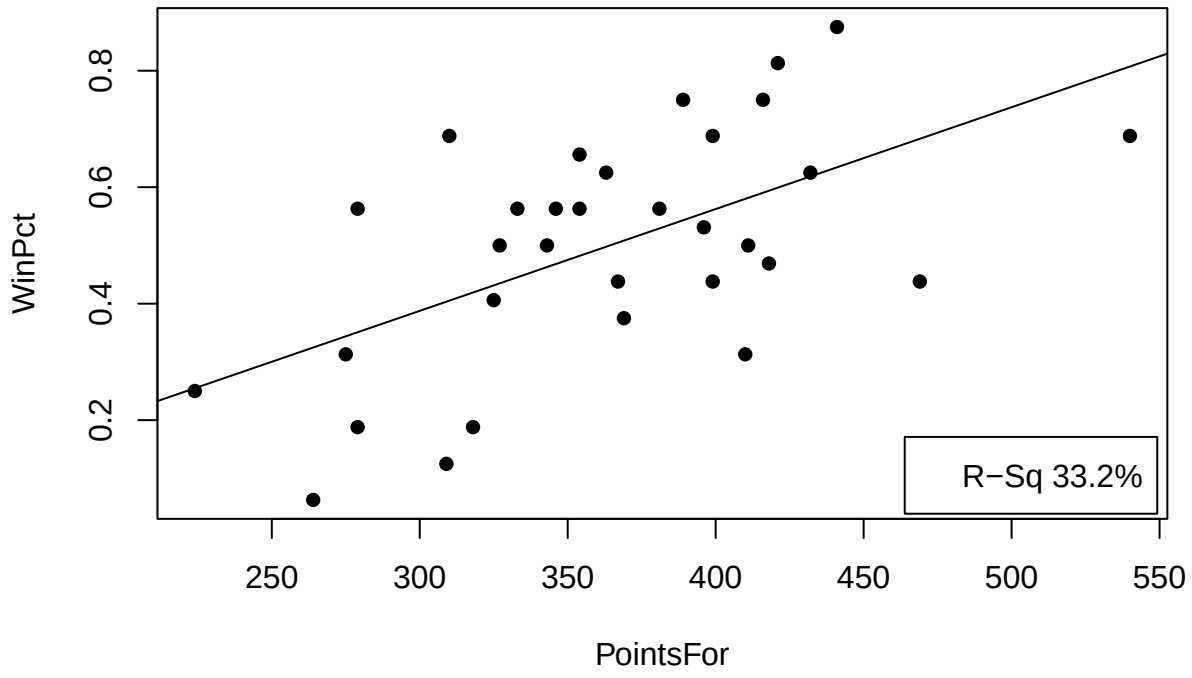
FIGURE 3.2 Linear regressions to predict NFL winning percentage

(a) Using points scored

```r
plot(WinPct~PointsFor, pch=16, main="", data=NFLStandings2016)
summary(NFLmod1 <- lm(WinPct~PointsFor, data=NFLStandings2016))
```

```
##
## Call:
## lm(formula = WinPct ~ PointsFor, data = NFLStandings2016)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.278389 -0.123281  0.000895  0.127225  0.282863
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1368044  0.1674677  -0.817 0.420426
## PointsFor    0.0017482  0.0004524   3.864 0.000554 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1662 on 30 degrees of freedom
## Multiple R-squared:  0.3323, Adjusted R-squared:   0.31
## F-statistic: 14.93 on 1 and 30 DF,  p-value: 0.0005541
```

```r
abline(NFLmod1)
legend("bottomright", legend=c("R-Sq 33.2%"), inset=.01)
```

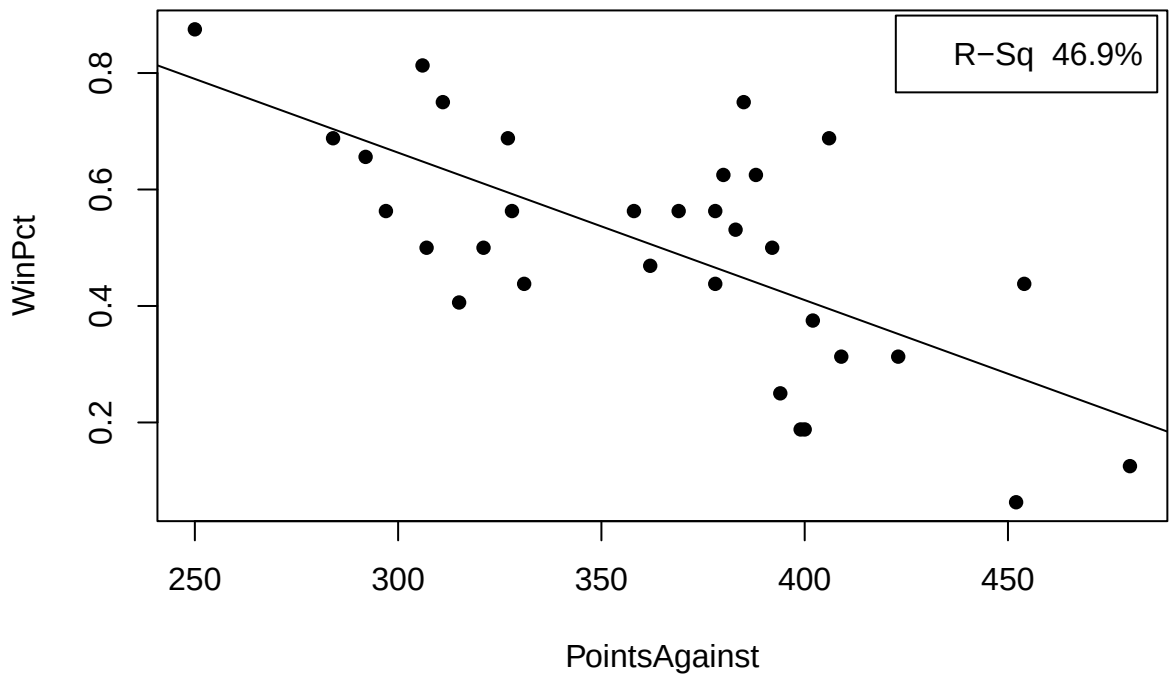


(b) Using points allowed

```
plot(WinPct~PointsAgainst, pch=16, main="", data=NFLStandings2016)
summary(NFLmod2 <- lm(WinPct~PointsAgainst, data=NFLStandings2016))
```

```
##
## Call:
## lm(formula = WinPct ~ PointsAgainst, data = NFLStandings2016)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22468 -0.10845 -0.02769  0.09408  0.30188
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.4227475  0.1812505   7.850 9.26e-09 ***
## PointsAgainst -0.0025315  0.0004922  -5.144 1.55e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1483 on 30 degrees of freedom
## Multiple R-squared:  0.4686, Adjusted R-squared:  0.4509
## F-statistic: 26.46 on 1 and 30 DF,  p-value: 1.552e-05
```

```
abline(NFLmod2)
legend("topright", legend=c("R-Sq  46.9%"), inset=.01)
```



EXAMPLE 3.2 NFL winning percentage: fitting the model

FIGURE 3.3 Computer output for a multiple regression

```
NFLmodel2016mr <- lm(WinPct ~ PointsFor + PointsAgainst , data=NFLStandings2016)
summary(NFLmodel2016mr)
```

```
##
## Call:
## lm(formula = WinPct ~ PointsFor + PointsAgainst, data = NFLStandings2016)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.149898 -0.073482 -0.006821  0.072569  0.213189
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.7853698  0.1537422   5.108 1.88e-05 ***
## PointsFor      0.0016992  0.0002628   6.466 4.48e-07 ***
## PointsAgainst -0.0024816  0.0003204  -7.744 1.54e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09653 on 29 degrees of freedom
## Multiple R-squared:  0.7824, Adjusted R-squared:  0.7674
## F-statistic: 52.13 on 2 and 29 DF,  p-value: 2.495e-10
```

Predict the winning percentage for the Green Bay Packers, who scored 432 points while allowing 388 points during the 2016 regular season.

```
newx=data.frame(PointsFor=432, PointsAgainst=388)
fit=predict(NFLmodel2016mr,newdata=newx)
fit
```

```
##         1
## 0.5565525
```

Find the residuals for the Green Bay Packers, who were 10-6 during the 2016 regular season.

```
ResidGBPackers=(10/16-fit)
ResidGBPackers
```

```
##          1
## 0.06844755
```

The standard error of the multiple regression model is shown in the original summary of the model. Residual standard error: 0.09653.

Alternative Solutions

There are also a couple of ways to find the quantities needed to compute the standard error of the regression model using anova( ).

```
anovaNFL2016=anova(NFLmodel2016mr)
anovaNFL2016
```

```
## Analysis of Variance Table
##
## Response: WinPct
##               Df  Sum Sq Mean Sq F value    Pr(>F)
## PointsFor      1 0.41262 0.41262  44.280 2.692e-07 ***
## PointsAgainst  1 0.55884 0.55884  59.972 1.537e-08 ***
## Residuals     29 0.27023 0.00932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can extract the SSE and error df from the anova object

```
SSE=anovaNFL2016$'Sum Sq'[3]
SSE
```

```
## [1] 0.2702337
```

```
df=anovaNFL2016$'Df'[3]
df
```

```
## [1] 29
```

```
se_mrm=sqrt(SSE/df)
se_mrm
```

```
## [1] 0.09653188
```

OR we can pull out the MSE and take the square root to find the standard error of the multiple regression model.

```
MSE=anovaNFL2016$'Mean Sq'[3]
sqrt(MSE)
```

```
## [1] 0.09653188
```