Multiple Linear Regression Application in R

```
library(car)
library(ggplot2)
library(mosaicData)
library(dplyr)
library(psych)
```

Multiple Regression

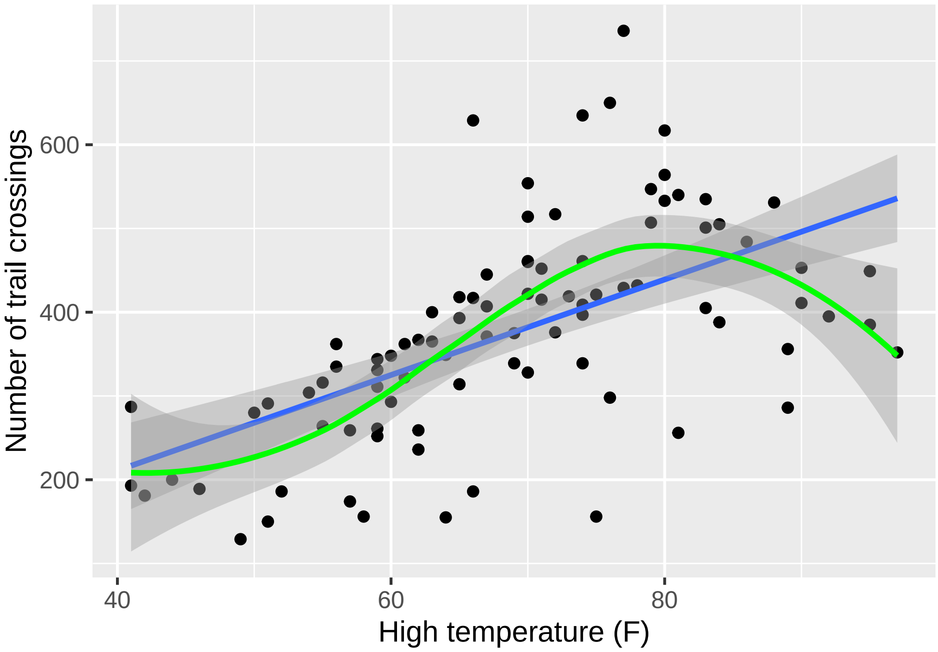
The `mosaicData` package contains a data set of the number of rail trail users over a 90 day period and the temperature, precipitation, cloud cover, day of the week, and a few other explanatory variables. The data set is called `RailTrail`

```
head(RailTrail)
```

```
##   hightemp lowtemp avgtemp spring summer fall cloudcover precip volume weekday
## 1       83      50    66.5      0      1    0        7.6   0.00    501    TRUE
## 2       73      49    61.0      0      1    0        6.3   0.29    419    TRUE
## 3       74      52    63.0      1      0    0        7.5   0.32    397    TRUE
## 4       95      61    78.0      0      1    0        2.6   0.00    385   FALSE
## 5       44      52    48.0      1      0    0       10.0   0.14    200    TRUE
## 6       69      54    61.5      1      0    0        6.6   0.02    375    TRUE
##   dayType
## 1 weekday
## 2 weekday
## 3 weekday
## 4 weekend
## 5 weekday
## 6 weekday
```

```
ggplot(data = RailTrail, aes(x = hightemp, y = volume)) +
  geom_point() +
  stat_smooth(method = lm) +
  stat_smooth(method = loess, color = "green") +
  ylab("Number of trail crossings") + xlab("High temperature (F)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Let's fit a model in which we predict the number of users (Y) conditional on the high temperature `hightemp` and the precipitation `precip`, (X_1 and X_2, respectively).

```
mod.rail <- lm(volume ~ hightemp + precip, data = RailTrail)
coef(mod.rail)
```

```
## (Intercept)    hightemp      precip
##  -31.519670    6.117745 -153.260844
```

The fitted least-squares regression equation is

$$\widehat{Volume} = -31.52 + 6.12 \times HighTemp - 153.26 \times Precip$$

For each additional degree in temperature, we expect an additional 6.1 riders on the rail trail, after controlling for the amount of precipitation. Controlling for temperature, an inch of rainfall is associated with a drop in ridership of about 153.

For the intercept, the interpretation is that if the high temperature was 0 degrees Fahrenheit and precipitation was 0, then the estimated ridership would be about -32 riders, which is non-sensical because we cannot have negative riders. Technically, the outcome is a count variable, which is more appropriately modeled using Poisson Regression. By doing so, we would not obtain predicted negative counts so we will return to this example later in the semester. In addition, a high temperature of 0 is far outside the range of the observed high temperatures (41-97 degrees).

```
summary(RailTrail)
```

```
##     hightemp        lowtemp         avgtemp          spring
##  Min.   :41.00   Min.   :19.00   Min.   :33.00   Min.   :0.0000
##  1st Qu.:59.25   1st Qu.:38.00   1st Qu.:48.62   1st Qu.:0.0000
##  Median :69.50   Median :44.50   Median :55.25   Median :1.0000
##  Mean   :68.83   Mean   :46.03   Mean   :57.43   Mean   :0.5889
##  3rd Qu.:77.75   3rd Qu.:53.75   3rd Qu.:64.50   3rd Qu.:1.0000
##  Max.   :97.00   Max.   :72.00   Max.   :84.00   Max.   :1.0000
##      summer            fall          cloudcover         precip
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 3.650   1st Qu.:0.00000
##  Median :0.0000   Median :0.0000   Median : 6.400   Median :0.00000
##  Mean   :0.2778   Mean   :0.1333   Mean   : 5.807   Mean   :0.09256
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 8.475   3rd Qu.:0.02000
##  Max.   :1.0000   Max.   :1.0000   Max.   :10.000   Max.   :1.49000
##      volume       weekday          dayType
##  Min.   :129.0   Mode :logical   Length:90
##  1st Qu.:291.5   FALSE:28        Class :character
##  Median :373.0   TRUE :62        Mode  :character
##  Mean   :375.4
##  3rd Qu.:451.2
##  Max.   :736.0
```

Precipitation has a larger coefficient. Does that mean it is more important than high temperature? Not necessarily, because they are on different scales.

We can standardize the variables to put them on the same scale. To do that we are going to use the `mutate` function from the package `dplyr`.

```
sRailTrail <- mutate(RailTrail,
                     s.hightemp = (hightemp-mean(hightemp))/sd(hightemp),
                     s.precip = (precip-mean(precip))/sd(precip),
                     s.volume = (volume-mean(volume))/sd(volume))
```

Standardized variables ought to have a mean of 0 and standard deviation of 1. So check to confirm.

```
summary(sRailTrail)
```

```
##     hightemp        lowtemp         avgtemp          spring
##  Min.   :41.00   Min.   :19.00   Min.   :33.00   Min.   :0.0000
##  1st Qu.:59.25   1st Qu.:38.00   1st Qu.:48.62   1st Qu.:0.0000
##  Median :69.50   Median :44.50   Median :55.25   Median :1.0000
##  Mean   :68.83   Mean   :46.03   Mean   :57.43   Mean   :0.5889
##  3rd Qu.:77.75   3rd Qu.:53.75   3rd Qu.:64.50   3rd Qu.:1.0000
##  Max.   :97.00   Max.   :72.00   Max.   :84.00   Max.   :1.0000
##      summer            fall          cloudcover         precip
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 3.650   1st Qu.:0.00000
##  Median :0.0000   Median :0.0000   Median : 6.400   Median :0.00000
##  Mean   :0.2778   Mean   :0.1333   Mean   : 5.807   Mean   :0.09256
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 8.475   3rd Qu.:0.02000
##  Max.   :1.0000   Max.   :1.0000   Max.   :10.000   Max.   :1.49000
##      volume       weekday          dayType            s.hightemp
##  Min.   :129.0   Mode :logical   Length:90          Min.   :-2.13723
##  1st Qu.:291.5   FALSE:28        Class :character   1st Qu.:-0.73587
##  Median :373.0   TRUE :62        Mode  :character   Median : 0.05119
##  Mean   :375.4                                      Mean   : 0.00000
##  3rd Qu.:451.2                                      3rd Qu.: 0.68468
##  Max.   :736.0                                      Max.   : 2.16282
##     s.precip          s.volume
##  Min.   :-0.3518   Min.   :-1.93312
##  1st Qu.:-0.3518   1st Qu.:-0.65823
##  Median :-0.3518   Median :-0.01883
##  Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.:-0.2758   3rd Qu.: 0.59508
##  Max.   : 5.3116   Max.   : 2.82907
```

The `summary()` function does not give the standard deviation. There is a function called `describe()` in the `psych` package that gives many more descriptives. First, load the `psych` package.

```
describe(sRailTrail[ ,12:14])
```

```
##            vars  n mean sd median trimmed  mad   min  max range skew kurtosis
## s.hightemp    1 90    0  1   0.05    0.00 1.08 -2.14 2.16  4.30 0.00    -0.52
## s.precip      2 90    0  1  -0.35   -0.25 0.00 -0.35 5.31  5.66 4.02    16.72
## s.volume      3 90    0  1  -0.02   -0.02 0.93 -1.93 2.83  4.76 0.23    -0.20
##              se
## s.hightemp 0.11
## s.precip   0.11
## s.volume   0.11
```

```
# does the same thing
describe(sRailTrail[ ,c("s.hightemp","s.precip","s.volume")])
```

```
##            vars  n mean sd median trimmed  mad   min  max range skew kurtosis
## s.hightemp    1 90    0  1   0.05    0.00 1.08 -2.14 2.16  4.30 0.00    -0.52
## s.precip      2 90    0  1  -0.35   -0.25 0.00 -0.35 5.31  5.66 4.02    16.72
## s.volume      3 90    0  1  -0.02   -0.02 0.93 -1.93 2.83  4.76 0.23    -0.20
##              se
## s.hightemp 0.11
## s.precip   0.11
## s.volume   0.11
```

```
mod.s.rail <- lm(s.volume ~ s.hightemp + s.precip, data = sRailTrail)
coef(mod.s.rail)
```

```
##   (Intercept)    s.hightemp      s.precip
## -9.090237e-17  6.250618e-01 -3.163405e-01
```

```
summary(mod.s.rail)
```

```
##
## Call:
## lm(formula = s.volume ~ s.hightemp + s.precip, data = sRailTrail)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.12856 -0.44362  0.04641  0.38413  2.32581
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.090e-17  7.995e-02   0.000  1.00000
## s.hightemp   6.251e-01  8.113e-02   7.704 1.97e-11 ***
## s.precip    -3.163e-01  8.113e-02  -3.899  0.00019 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7585 on 87 degrees of freedom
## Multiple R-squared:  0.4377, Adjusted R-squared:  0.4247
## F-statistic: 33.85 on 2 and 87 DF,  p-value: 1.334e-11
```

So one standard deviation increase in the high temperature (holding constant precipitation), is associated with 0.625 standard deviations increase in volume and one standard deviation increase in precipitation (holding constant high temperature), is associated with 0.316 standard deviations *decrease* in volume.

You can suppress the scientific notation using `options(scipen=20)`.

```
options(scipen=20)
summary(mod.s.rail)
```

```
## 
## Call:
## lm(formula = s.volume ~ s.hightemp + s.precip, data = sRailTrail)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.12856 -0.44362  0.04641  0.38413  2.32581 
## 
## Coefficients:
##                           Estimate             Std. Error t value
## (Intercept) -0.0000000000000000909  0.0799493961567915024   0.000
## s.hightemp   0.6250618461362794642  0.0811324862538072566   7.704
## s.precip    -0.3163405019709329968  0.0811324862538072705  -3.899
##                    Pr(>|t|)    
## (Intercept)         1.00000    
## s.hightemp  0.0000000000197 ***
## s.precip            0.00019 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.7585 on 87 degrees of freedom
## Multiple R-squared:  0.4377, Adjusted R-squared:  0.4247 
## F-statistic: 33.85 on 2 and 87 DF,  p-value: 0.00000000001334
```

Note that R^2 does not change.

```
options(scipen = NULL) # Revert back to default scientific notation option
summary(mod.rail)
```

```
## 
## Call:
## lm(formula = volume ~ hightemp + precip, data = RailTrail)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -271.311  -56.545    5.915   48.962  296.453 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -31.5197    55.2383  -0.571  0.56973    
## hightemp       6.1177     0.7941   7.704 1.97e-11 ***
## precip      -153.2608    39.3071  -3.899  0.00019 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 96.68 on 87 degrees of freedom
## Multiple R-squared:  0.4377, Adjusted R-squared:  0.4247 
## F-statistic: 33.85 on 2 and 87 DF,  p-value: 1.334e-11
```

Computing sum of squares

Let's compute RSS and TSS for the rail trail regression. First, RSS comes from our original model.

```
rss.m1 <- sum(residuals(mod.rail)^2)
rss.m1
```

```
## [1] 813125.7
```

Next, we need to find the TSS, which we can obtain by fitting a "null" or "intercept-only" model and computing the RSS for that model. The RSS from this null model is the TSS.

```
m0 <- lm(volume ~ 1, data = RailTrail)
rss.m0 <- sum(residuals(m0)^2)
tss <- rss.m0
tss
```

```
## [1] 1445958
```

The $RegSS$ is then `tss-rss.m1`, and R^2 is `(tss-rss.m1)/tss`.

```
tss-rss.m1
```

```
## [1] 632831.9
```

```
(tss-rss.m1)/tss
```

```
## [1] 0.4376559
```

which is what we obtained from our earlier output.

```
summary(mod.rail)
```

```
##
## Call:
## lm(formula = volume ~ hightemp + precip, data = RailTrail)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -271.311  -56.545    5.915   48.962  296.453
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -31.5197    55.2383  -0.571  0.56973
## hightemp       6.1177     0.7941   7.704 1.97e-11 ***
## precip      -153.2608    39.3071  -3.899  0.00019 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.68 on 87 degrees of freedom
## Multiple R-squared:  0.4377, Adjusted R-squared:  0.4247
## F-statistic: 33.85 on 2 and 87 DF,  p-value: 1.334e-11
```

The multiple correlation coefficient is the simple correlation between the observed Y values and the fitted values, $\widehat{Y}$,

```
yhat <- fitted(mod.rail)
cor(RailTrail$volume, yhat)
```

```
## [1] 0.6615557
```

and is the same as the positive square root of R^2

```
sqrt(0.4377)
```

```
## [1] 0.661589
```

Hypothesis testing and confidence intervals

```
summary(mod.rail)
```

```
##
## Call:
## lm(formula = volume ~ hightemp + precip, data = RailTrail)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -271.311  -56.545    5.915   48.962  296.453
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -31.5197    55.2383  -0.571  0.56973
## hightemp       6.1177     0.7941   7.704 1.97e-11 ***
## precip      -153.2608    39.3071  -3.899  0.00019 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.68 on 87 degrees of freedom
## Multiple R-squared:  0.4377, Adjusted R-squared:  0.4247
## F-statistic: 33.85 on 2 and 87 DF,  p-value: 1.334e-11
```

The above results show that the `hightemp` estimate is statistically significantly different from 0, $t(87) = 7.704, p < .001$, as is the `precip` estimate, $t(87) = -3.899, p < .001$. The intercept estimate is not statistically different from 0, $t(87) = -0.571, p = 0.570$.

Confidence intervals for the estimates can be obtained using the `confint` function.

```
confint(mod.rail)
```

```
##                   2.5 %     97.5 %
## (Intercept) -141.311860  78.272520
## hightemp       4.539429   7.696062
## precip      -231.387996 -75.133691
```

So 95 out of 100 repeated samples, we would expect the `hightemp` estimate to lie between 4.539 and 7.696 and the `precip` estimate to lie between -231.39 and -75.13. Because the intervals do not include 0, we can conclude that these estimates are statistically significantly different from 0 at $\alpha = 0.05$. The confidence interval for the intercept does include 0 and matches the conclusion above that the intercept estimate is not significantly different from 0 at $\alpha = 0.05$.

An F-test for the omnibus null hypothesis that all of the slopes are zero can be calculated from the analysis of variance for the regression:

$$F_0 = \frac{RegSS/k}{RSS/(n-k-1)}$$

The omnibus F-statistic has k and $n-k-1$ degrees of freedom.

In this case, $F(2, 87) = 33.85, p < .001$.

Incremental F tests

There is also an F-test for the hypothesis that a subset of q slope coefficients is zero, based on a comparison of the $RegSS$ for the full regression model (model 1) and for a reduced model (model 0) that deletes the explanatory variables in the null hypothesis:

$$F_0 = \frac{(RegSS_1 - RegSS_0)/q}{RSS_1/(n-k-1)}$$

This incremental F-statistic has q and $n-k-1$ degrees of freedom.

Returning to the rail trail data, let's now add `cloudcover` and `lowtemp` as predictors.

```
full <- lm(volume ~ hightemp + lowtemp + precip + cloudcover, data = RailTrail)
summary(full)
```

```
##
## Call:
## lm(formula = volume ~ hightemp + lowtemp + precip + cloudcover,
##     data = RailTrail)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -269.447  -37.449    4.186   41.178  299.266
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.308     59.796   0.590   0.5564
## hightemp       6.571      1.153   5.699  1.7e-07 ***
## lowtemp       -1.290      1.387  -0.930   0.3551
## precip      -100.616     42.064  -2.392   0.0190 *
## cloudcover    -7.501      3.851  -1.948   0.0547 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.2 on 85 degrees of freedom
## Multiple R-squared:  0.4894, Adjusted R-squared:  0.4654
## F-statistic: 20.37 on 4 and 85 DF,  p-value: 8.537e-12
```

Suppose we want to test the hypothesis that $H_0 : \beta_1 = \beta_2 = 0$. Then we can fit a reduced model that does not include these predictors.

```
reduced <- lm(volume ~ precip + cloudcover, data = RailTrail)
summary(reduced)
```

```
##
## Call:
## lm(formula = volume ~ precip + cloudcover, data = RailTrail)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -219.47  -79.93   -0.37   64.59  345.06
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   456.98      26.24  17.415  < 2e-16 ***
## precip        -52.79      51.51  -1.025  0.30827
## cloudcover    -13.21       4.20  -3.145  0.00228 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118.8 on 87 degrees of freedom
## Multiple R-squared:  0.1506, Adjusted R-squared:  0.131
## F-statistic:  7.71 on 2 and 87 DF,  p-value: 0.0008269
```

```
anova(reduced, full)
```

```
## Analysis of Variance Table
##
## Model 1: volume ~ precip + cloudcover
## Model 2: volume ~ hightemp + lowtemp + precip + cloudcover
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     87 1228266
## 2     85  738282  2    489984 28.206 4.023e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So `hightemp` and `lowtemp` significantly add to the prediction of trail users above and beyond `precip` and `cloudcover`, $F(2, 85) = 28.206, p < .001$. In other words, we reject $H_0 : \beta_1 = \beta_2 = 0$.

Summary

In simple linear regression, the least-squares coefficients are given by

$$\widehat{\beta_0} = \overline{Y} - \widehat{\beta_1 \overline{X}}$$

$$\widehat{\beta_1} = \frac{cov_{XY}}{s_X^2} = \frac{\sum(X_i - \overline{X})(Y_i - \overline{Y})}{\sum(X_i - \overline{X})^2}$$

The least-squares coefficients in multiple linear regression are found by solving the normal equations for the intercept β_0 and the slope coefficients $\beta_1, \beta_2, \ldots, \beta_k$.

The least-squares residuals, ϵ, are uncorrelated with the fitted values, $\widehat{Y}$, and with the explanatory variables, $X_1, X_2, \ldots, X_k$.

The linear regression decomposes the variation in Y into 'explained' and 'unexplained' components: $TSS = RegSS + RSS$.

The standard error of the regression,

$$S_\epsilon = \sqrt{\frac{\sum \epsilon_i^2}{n - k - 1}}$$

gives the 'average' size of the regression residuals.

The squared multiple correlation,

$$R^2 \equiv \frac{RegSS}{TSS}$$

indicates the proportion of the variation in Y that is captured by its linear regression on the X's.

Standard statistical inference for least-squares regression analysis is based upon the statistical model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_k X_{ik} + \epsilon_i$$

The key assumptions of the model include linearity, constant variance, normality, and independence.

The X-values are either fixed or, if random, are assumed to be independent of the errors.

Under these assumptions, or particular subsets of them, the least-squares coefficients have certain desirable properties as estimators of the population regression coefficients.

The estimated error of the slope coefficient $\widehat{\beta}_1$ in simple regression is

$$SE(\widehat{\beta}_1) = \frac{\sigma_\epsilon}{\sqrt{\sum (X_i - \overline{X})^2}}$$

The standard error of the slope coefficient $\widehat{\beta}_j$ in multiple regression is

$$SE(\widehat{\beta}_j) = \frac{1}{\sqrt{1 - R_j^2}} \times \frac{\sigma_\epsilon}{\sqrt{\sum (X_{ij} - \overline{X}_j)^2}}$$

In both cases, these standard errors can be used in confidence intervals and hypothesis tests for the corresponding population slope coefficients.

By rescaling regression coefficients in relation to a measure of variation (i.e., standard deviation), standardized regression coefficients permit a limited comparison of the relative impact of incommensurable explanatory variables.