2.4 Testing in MLR

Now that we understand the distribution of the LS estimators, we are ready to build hypothesis tests for the coefficients of the predictors in the model. Specifically, we will focus on

a. Testing a **single** predictor/coefficient.

b. Testing **all** predictors/coefficients.

c. Testing **multiple** (a *subset*) predictors/coefficients.

d. Other types of hypothesis tests.

2.4.1 Testing a Single Predictor

Suppose you have a p predictors in your regression model and you want to test the following hypothesis for a single predictor β_j:

$$\begin{cases} H_0 : \beta_j = c \\ H_\alpha : \beta_j \neq c \end{cases}$$

Similar to the one-dimensional case, under the null hypothesis H_0, the t-test statistic is:

$$t = \frac{\hat{\beta}_j - c}{se(\hat{\beta}_j)} = \frac{\hat{\beta}_j - c}{\hat{\sigma}\sqrt{[(\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}]_{jj}}} \sim T_{n-p}$$

and the p-value is twice the area under the curve of a T_{n-p} distribution *more extreme* than the observed statistic. The term $[(\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}]_{jj}$ is the element of the covariance matrix that corresponds to the variance of $\hat{\beta}_j$.

Birthweight Example

In `R`, we can see the test statistics and the corresponding p-values in the `summary` output. The p-value returned by the lm function command is for $c = 0$. For example, if we are interested to test the coefficient of the `Gestation` variable, the test formulates as

$$\begin{cases} H_0 : \beta_{Gestation} = 0 \\ H_\alpha : \beta_{Gestation} \neq 0 \end{cases}$$

```
summary(birthweight.mlr1)
```

```
##
## Call:
## lm(formula = Birthweight ~ ., data = birthweight2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38656 -0.26722 -0.06068  0.18271  0.60295
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.4286417  2.1583202  -1.589  0.12339
## Length       0.0257860  0.0336226   0.767  0.44954
## Headcirc     0.0850933  0.0297843   2.857  0.00798 **
## Gestation    0.0916226  0.0322518   2.841  0.00829 **
## smoker      -0.2198237  0.1728531  -1.272  0.21393
## mage        -0.0158203  0.0191605  -0.826  0.41597
## mnocig       0.0002094  0.0070011   0.030  0.97635
## mheight      0.0056438  0.0143947   0.392  0.69797
## mppwt        0.0084338  0.0116016   0.727  0.47329
## fage         0.0046535  0.0167998   0.277  0.78382
## fedyrs       0.0016448  0.0312497   0.053  0.95840
## fnocig       0.0040531  0.0041251   0.983  0.33424
## fheight     -0.0122665  0.0097854  -1.254  0.22037
## lowbwt      -0.1751779  0.2346936  -0.746  0.46164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3367 on 28 degrees of freedom
## Multiple R-squared:  0.7877, Adjusted R-squared:  0.6891
## F-statistic: 7.989 on 13 and 28 DF,  p-value: 2.432e-06
```

The test statistic that corresponds to the coefficient of the `Gestation` variable is 0.0916226 and is distributed according to a T_{28} distribution with $n-2=28$ degrees of freedom and a corresponding p-value equal to 0.00829.

Since $p = value = 0.00829 < 0.05 = \alpha$ (**Decision rule**), we reject the null hypothesis and conclude that the `Gestation` predictor is statistically significant. Since the partial slope is *positive*, there is a positive association between `Birthweight` and `Gestation` , *conditionally on all the other variables being in the model*.

R **Remark**: We can directly extract the coefficient of `Gestation` only using

```
summary(birthweight.mlr1)$coef[4,]
```

```
##    Estimate  Std. Error     t value    Pr(>|t|)
## 0.091622593 0.032251827 2.840849703 0.008294721
```

A Remark on the Degrees of Freedom of a t-test

So far, we have seen that the degrees of freedom for a single coefficient t test is $n-2$ in the *Simple Linear Regression* model and $n-p$ in the *Multiple Linear Regression*. So, the question here is how do we determine the degrees of freedom of the T distribution?

Observe that all t-tests we have encountered so far involve an estimate of the error variance σ^2. The **degrees of freedom** of a t-test are determined by the denominator of the estimated variance $\hat{\sigma}^2$.

Consider the following situations:

- **STAT 400**:
  Let $Z_1, \ldots, Z_n \sim \mathcal{N}(\theta, \sigma^2)$. We want to test whether $\boldsymbol{\theta = c}$ or not. The test statistic is

  $$\frac{\hat{\theta} - c}{se(\hat{\theta})} = \frac{\bar{Z} - c}{\sqrt{\hat{\sigma}^2 / n}} \sim T_{n-1}, \quad \text{where} \quad \hat{\sigma}^2 = \frac{\sum_i (Z_i - \bar{Z})^2}{n-1}$$

  is the estimator for the variance of the sample, i.e. σ^2. Observe that the degrees of freedom of the T distribution are the degrees of freedom associated with the variance estimator.

- **In SLR**:
  Consider now the test for the slope coefficient β_1 in SLR. We want to test whether $\beta_1 = c$ or not, so we have

$$\frac{\hat{\beta}_1 - c}{se(\hat{\beta}_1)} = \frac{\hat{\beta}_1 - c}{\sqrt{\hat{\sigma}^2 / S_{XX}}} \sim T_{n-2}, \quad \text{where} \quad \hat{\sigma}^2 = \frac{RSS}{n-2}$$

  is the estimator for the variance of the error terms, i.e. σ^2. Observe that the degrees of freedom of the T distribution are the degrees of freedom associated with the variance estimator.

- **In MLR** (with p predictors including the intercept):
  Again, we consider the test for a partial slope coefficient β_j, that is we want to test whether $\beta_j = c$ or not, so the test statistic is

$$\frac{\hat{\beta}_j - c}{se(\hat{\beta}_j)} = \frac{\hat{\beta}_j - c}{\sqrt{\hat{\sigma}^2 \ [(\mathbf{X}^T\mathbf{X})^{-1}]_{jj}}} \sim T_{n-p}, \quad \text{where} \quad \hat{\sigma}^2 = \frac{RSS}{n-p}$$

  is again the estimator for the variance of the error terms, i.e. σ^2. Observe that the degrees of freedom of the T distribution are the degrees of freedom associated with the variance estimator.

Birthweight Example

The *degrees of freedom* of RSS can be found in the output of the `lm` object or using the `anova` function:

```
birthweight.mlr1$df.residual
```

```
## [1] 28
```

```
anova(birthweight.mlr1)[14,1]
```

```
## [1] 28
```

2.4.2 Overall F-test in MLR

F Test in MLR

Suppose we want to conduct the following hypothesis test:

$$\begin{cases} H_0 : \beta_2 = \beta_3 = \ldots = \beta_p = 0 \\ H_a : \quad \beta_j \neq 0, \quad \text{for some } j, \, j = 2, \ldots, p \end{cases}$$

Under the Null hypothesis, the test statistic:

$$\begin{aligned} F &= \frac{FSS(X_2, \ldots, X_p)}{p-1} \div \frac{RSS(X_2, \ldots, X_p)}{n-p} \\ &= \frac{MSReg}{MSError} \sim F_{p-1,n-p} \end{aligned}$$

Large values of F lead to rejecting the H_0.

This is the **overall F test** of whether or not there is a regression relation between the response variable Y and the set of X variables. Note that the hypothesis *does not* include the intercept term. The reason is that the overall test is a test on whether the intercept-only model is more adequate compared to the full model.

Birthweight Example

In `R`, we can find the elements to perform the overall F test in the `summary` output:

```
summary(birthweight.mlr1)
```

```
##
## Call:
## lm(formula = Birthweight ~ ., data = birthweight2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38656 -0.26722 -0.06068  0.18271  0.60295
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.4286417  2.1583202  -1.589  0.12339
## Length       0.0257860  0.0336226   0.767  0.44954
## Headcirc     0.0850933  0.0297843   2.857  0.00798 **
## Gestation    0.0916226  0.0322518   2.841  0.00829 **
## smoker      -0.2198237  0.1728531  -1.272  0.21393
## mage        -0.0158203  0.0191605  -0.826  0.41597
## mnocig       0.0002094  0.0070011   0.030  0.97635
## mheight      0.0056438  0.0143947   0.392  0.69797
## mppwt        0.0084338  0.0116016   0.727  0.47329
## fage         0.0046535  0.0167998   0.277  0.78382
## fedyrs       0.0016448  0.0312497   0.053  0.95840
## fnocig       0.0040531  0.0041251   0.983  0.33424
## fheight     -0.0122665  0.0097854  -1.254  0.22037
## lowbwt      -0.1751779  0.2346936  -0.746  0.46164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3367 on 28 degrees of freedom
## Multiple R-squared:  0.7877, Adjusted R-squared:  0.6891
## F-statistic: 7.989 on 13 and 28 DF,  p-value: 2.432e-06
```

In the `Birthweight` example, we have that F is equal to **7.989** with a p-value equal to **2.432e-06** which means that the null hypothesis is rejected. This implies that **at least one** of the β coefficients is non-zero which is interpreted that the model including all predictors is "better" than the intercept-only model.

Note that the this test *does not give us any information* regarding **which** of the predictors are the insignificant ones. Additional tests are required to identify if there are any statistically insignificant predictors.

In `R` , we can also extract just the F test statistic and corresponding degrees of freedom of the numerator and denominator using:

```
summary(birthweight.mlr1)$fstatistic
```

```
##     value     numdf     dendf
##  7.989073 13.000000 28.000000
```

2.4.3 Test whether Several $\beta_k = 0$

The problem of testing for a **subset of predictors** is addressed by *comparing two regression models*. For simplicity in the notation, assume that we want to test whether the coefficients β_{p-2}, β_{p-1}, β_p are zero altogether. So, we formulate two models:

- **Full Model**: the one containing **all** available predictors:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p X_{ip} + \varepsilon_i$$

- **Reduced** Model: the one containing all the predictors **except** for the subset of predictors we want to test out:

$$y_i = \beta_1 x_{i1} + \ldots + \beta_{p-3} x_{i,p-3} + \varepsilon_i$$

Birthweight Example

Using abstract notation, the **full** model we consider can be written as:

$$y_i = \beta_0 + \beta_1 X_1 + \ldots + \beta_{14} X_{14} + \varepsilon_i$$

Assume we want to test the hypothesis whether `fedyrs` and `fage` and `fnocig` and `fheight` are statistically significant.

- The *full* model corresponds to the regression with all the variables

```
birthweight.mlr1 = lm(Birthweight~., birthweight2)
```

- The *reduced* model corresponds to the regression in which the columns that correspond to `fedyrs` , `fage` , `fnocig` , `fheight` have been removed from the design matrix. That is

$$y_i = \beta_1 X_1 + \ldots + \beta_9 X_9 + \beta_{14} X_{14} + \varepsilon_i$$

```
birthweight.mlr.red = lm(Birthweight ~ Length + Headcirc + Gestation + smoker +
  mage + mnocig + mheight + mppwt + lowbwt , birthweight2)
```

But, how can we compare the two models?

Let's start by properly formulating the corresponding hypotheses to test. All the formulations below are **equivalent**:

1. Hypothesis formulated in terms of the coefficients we want to test:

$$\begin{cases} H_0 : \beta_{p-2} = \beta_{p-1} = \beta_p = 0 \\ H_\alpha : \text{at least one of } \beta_{p-2}, \beta_{p-1}, \beta_p \text{ is not equal to } 0 \end{cases}$$

2. Hypothesis using description of the models we want to compare:

$$\begin{cases} H_0: \text{ The reduced model is adequate} \\ H_\alpha: \text{The full model is required} \end{cases}$$

3. Hypothesis formulated in terms of the models we want to compare:

$$\begin{cases} y_i = \beta_1 x_{i1} + \ldots + \beta_{p-3} x_{i(p-3)} + \varepsilon_i \text{ Reduced Model} \\ y_i = \beta_1 x_{i1} + \ldots + \beta_p X_{ip} + \varepsilon_i \text{ Full Model} \end{cases}$$

All three formulations are equivalent, and all describe the same test of whether the coefficients $\beta_{p-2} = \beta_{p-1} = \beta_p$ are **all** zero, or if there is *at least one* that is not equal to zero. The **test statistic** to test this hypothesis is the so-called **Partial F test**.

2.4.4 Partial F-Test

In general, consider the following **partition** of the design matrix into two sub-matrices $\mathbf{X_1}$ and $\mathbf{X_2}$, that is

$$\mathbf{X}_{n\times p} = (\mathbf{X}_{1,n\times(p-q)}, \mathbf{X}_{2,n\times q})$$

We assume that the sub-matrix $\mathbf{X_1}$ contains $p-q$ columns/predictors, while $\mathbf{X_2}$ contains the remaining q columns. The corresponding **partition** of the parameter vector is:

$$\beta^T = (\beta_1^T, \beta_2^T)$$

where β_1 is $(p-q)\times 1$ and β_2 is $q \times 1$.

Using this notation, we can also formulated the hypothesis test as follows:

$$\begin{cases} H_0: \beta_2 = \mathbf{0}, \text{ i.e., } \mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{error} \\ H_\alpha: \beta_2 \neq \mathbf{0}, \text{ i.e., } \mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2 + \mathbf{error} \end{cases}$$

where β_1 and β_2 here denote the two sub-vector (partitions) of the original β vector.

To test this hypothesis, the **test statistic** is:

Partial F Test

$$F = \frac{(RSS_0 - RSS_\alpha)/q}{RSS_\alpha/(n-p)} \sim F_{q,n-p}$$

where

- RRS_0 is the Residual Sum of Squares for the model under the H_0 and
- RRS_α is the Residual Sum of squares for the model under the H_α.
- q is the number of predictors we want to test and is equal to

$$df_0 - df_\alpha = n - p - (n - p - q) = q$$

- $n - p$ is the number of residual degrees of freedom of the model under the H_0.

Note here that *the model under the H_0 is a **subset** of the model under the alternative*. This F test rejects the H_0, if the F test statistic is *large*, that is when the variation missed by the reduced model, when being compared with the error variance, is significantly large[11].

The intuition behind this test is that the numerator is the variation in the data *not explained by the reduced model, but explained by the full model*, while the Denominator is variation in the data *not explained by the full model*, which is used to estimate the error variance.

Birthweight Example

In `R`, the partial F test calculation is done using the `anova` function. So, if we want to "test out" the variables `fedyrs`, `fage`, `fnocig`, and `fheight` we have:

```
# Full Model (model under H_a)
birthweight.mlr1 = lm(Birthweight~., birthweight2) # as before

# Reduced Model (model under H_0)
birthweight.mlr.red = lm(Birthweight ~ Length + Headcirc + Gestation + smoker + m

# Partial F Test
anova(birthweight.mlr.red, birthweight.mlr1)
```

```
## Analysis of Variance Table
##
## Model 1: Birthweight ~ Length + Headcirc + Gestation + smoker + mage +
##     mnocig + mheight + mppwt + lowbwt
## Model 2: Birthweight ~ Length + Headcirc + Gestation + smoker + mage +
##     mnocig + mheight + mppwt + fage + fedyrs + fnocig + fheight +
##     lowbwt
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1     32 3.4645
## 2     28 3.1751  4    0.2894 0.638 0.6397
```

The value of the partial F test statistic is **0.638** and the corresponding p-value is **0.6397**, which means that we *fail to reject the null*, and therefore *the reduced model is adequate*. This implies that if we were to remove the variables `fedyrs` , `fage` , `fnocig` , and `fheight` from the model, then that would not significantly affect the fit of the regression.

Remark: Observe that in the `anova` function we first input the *reduced* model and then the *full*, and conceptually this corresponds to *model under the null* first and *model under the alternative* second. In practice, if we were to change the order of the two models, then `R` would still perform the same hypothesis test of full/reduced. The output would be the same except for the sign of the F test statistic that would be the opposite.

We can also perform this partial F test *"by hand"* as follows:

```
rss.full1 = sum(birthweight.mlr1$res^2) # by calculating RSS using the residuals
# or alternatively
rss.full2 = deviance(birthweight.mlr1)  # using the deviance function to get the

# Similarly for the reduced model
rss.reduced1 = sum(birthweight.mlr.red$res^2)
rss.reduced2 = deviance(birthweight.mlr.red)

# Once we have the RSS values (one way or another) we can formulated the F test s

Fstat = (rss.reduced1 - rss.full1)/4/(rss.full1/28)
Fstat
```

```
## [1] 0.6380294
```

```
# The p-value is obtained using the pf function in R
1-pf(Fstat, 4, 28)
```

```
## [1] 0.6397153
```

As you can see the conclusion is exactly the same (up to rounding errors) as the one obtained using the `anova` function.

Below, we summarize all the tests we discussed using the Partial F tests:

1. **Testing all predictors**: (The default F-test returned by the function `lm`):

$$\begin{cases} H_0 : \mathbf{y} = \mathbf{1}_n\alpha + \textbf{error} \\ H_\alpha : \mathbf{y} = \mathbf{X}_{n\times p}\beta + \textbf{error} \end{cases}$$

2. **Testing one-predictor**:

$$\begin{cases} H_0 : \mathbf{y} = \mathbf{X}[,-\mathbf{j}]_{n\times(p-1)}\alpha + \mathbf{error} \\ H_\alpha : \mathbf{y} = \mathbf{X}_{n\times p}\beta + \mathbf{error} \end{cases}$$

where $\mathbf{X}[,-\mathbf{j}]$ is the desing matrix $\mathbf{X}$ without the j-th column, and α is $(p-1)\times 1$.

3. **Testing a subset of predictors**:

$$\begin{cases} H_0 : \mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{error} \\ H_\alpha : \mathbf{y} = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2 + \mathbf{error} \end{cases}$$

where $\mathbf{(X_1, X_2)}$ is a partition of matrix $\mathbf{X}$.

2.4.5 Testing a sub-space of predictors

Assume now that we want to test a sub-space of the predictors that does not fall in any of the categories above.

For example, assume that we want to test

$$\begin{cases} H_0 : \beta_2 = \beta_3 \\ H_\alpha : \beta_2 \neq \beta_3 \end{cases} \Leftrightarrow \begin{cases} H_0 : y_i = \beta_1 x_{i1} + \beta_c(x_{i2} + x_{i3}) + \ldots + \beta_p x_{ip} + \varepsilon_i \\ H_\alpha : y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \beta_p x_{ip} + \varepsilon_i \end{cases}$$

Using a general formulation, the hypothesis test above is a special case of the following:

$$\begin{cases} H_0 : \mathbf{y} = \mathbf{X}_1\gamma + \mathbf{error} \\ H_\alpha : \mathbf{y} = \mathbf{X}\beta + \mathbf{error} \end{cases}$$

where $\mathbf{X}_1$ is a $n \times (p-1)$ matrix that is almost the same as $\mathbf{X}$, but replaces the 2nd and 3rd columns of $\mathbf{X}$ by their sum, and γ is $(p-1)\times 1$.

In this case, we can still perform the *partial F-test* between the full (model under H_α) and the reduced (model under the H_0) models, since the model under the H_0 is a subset of the model under the H_α.

Birthweight Example

Assume that we want to test the following hypothesis

$$\begin{cases} H_0 : \beta_{fage} = \beta_{fheight} \\ H_\alpha : \beta_{fage} \neq \beta_{fheight} \end{cases}$$

This means that the reduced model under the H_0 is the following:

$$y_i = \beta_1 x_{i1} + \ldots + \beta_9 x_{i9} + \beta_{11} x_{i11} + \beta_{12} x_{i12} + \beta_c (x_{i10} + x_{i13}) + \beta_{14} x_{i14}$$

```
# Full Model
birthweight.mlr1 = lm(Birthweight~., birthweight2)

# Reduced Model
birthweight.mlr.red2 = lm(Birthweight ~ Length + Headcirc + Gestation + smoker +
  mage + mnocig + mheight + mppwt + fedyrs + fnocig + lowbwt  + I(fage+fheight),
summary(birthweight.mlr.red2)
```

```
##
## Call:
## lm(formula = Birthweight ~ Length + Headcirc + Gestation + smoker +
##     mage + mnocig + mheight + mppwt + fedyrs + fnocig + lowbwt +
##     I(fage + fheight), data = birthweight2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40548 -0.23232 -0.05593  0.18954  0.55503
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.2475982  2.1465079  -1.513  0.14111
## Length             0.0230032  0.0334420   0.688  0.49701
## Headcirc           0.0942470  0.0281469   3.348  0.00226 **
## Gestation          0.0905179  0.0321796   2.813  0.00872 **
## smoker            -0.2027740  0.1716499  -1.181  0.24707
## mage              -0.0020825  0.0125963  -0.165  0.86983
## mnocig             0.0001155  0.0069893   0.017  0.98693
## mheight           -0.0001072  0.0130469  -0.008  0.99350
## mppwt              0.0109846  0.0112704   0.975  0.33780
## fedyrs            -0.0053502  0.0303263  -0.176  0.86119
## fnocig             0.0039052  0.0041156   0.949  0.35053
## lowbwt            -0.2352302  0.2257090  -1.042  0.30594
## I(fage + fheight) -0.0088077  0.0090725  -0.971  0.33966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3362 on 29 degrees of freedom
## Multiple R-squared:  0.7808, Adjusted R-squared:  0.6901
## F-statistic: 8.607 on 12 and 29 DF,  p-value: 1.188e-06
```

Using `I(fage+fheight)` we tell `R` to treat the variables as a single variable. This is equivalent to computing a new variable equal to the sum of the other two and using that in the model In the `summary` output observe that there is a single coefficient estimated for `fage` and `fheight` which is equal to **-0.0088077.**

We now perform the *partial F test* using the `anova()` function:

```
anova(birthweight.mlr.red2, birthweight.mlr1)
```

```
## Analysis of Variance Table
##
## Model 1: Birthweight ~ Length + Headcirc + Gestation + smoker + mage +
##     mnocig + mheight + mppwt + fedyrs + fnocig + lowbwt + I(fage +
##     fheight)
## Model 2: Birthweight ~ Length + Headcirc + Gestation + smoker + mage +
##     mnocig + mheight + mppwt + fage + fedyrs + fnocig + fheight +
##     lowbwt
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     29 3.2780
## 2     28 3.1751  1   0.10291 0.9076 0.3489
```

The p-value of the *partial F-test* is equal to **0.3489** which means that we *fail to reject the null* and conclude that indeed $\beta_2 = \beta_3$.

Another example is when we want to test the following hypothesis:

$$\begin{cases} H_0 : \beta_2 = c \text{ and } \beta_3 = d \\ H_\alpha : \beta_2 \neq c \text{ or } \beta_3 \neq d \end{cases}$$

where c, d are known constants.

In this case, we just need to observe that the model under the H_α is the following:

$$H_0 : y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \varepsilon_i$$

while the model under the H_0 can be written as:

$$H_\alpha : y_i = \beta_1 x_{i1} + \beta_2\, x_{i2} + \beta_3\, x_{i3} \ldots + \beta_p x_{ip} + \varepsilon_i$$

which is equivalent to

$$\begin{aligned} y_i = \beta_1 x_{i1} + c\, x_{i2} + d\, x_{i3} + \ldots + \beta_p x_{ip} + \varepsilon_i \ \Leftrightarrow \\ y_i - c\, x_{i2} - d\, x_{i3} = \beta_1 x_{i1} + \beta_4 x_{i4} + \ldots + \beta_p x_{ip} + \varepsilon_i \end{aligned}$$

So, to test the hypothesis, we fit both models and use the **General Linear Test** statistic F

$$F = \frac{(RSS_0 - RSS_\alpha)/2}{RSS_\alpha/n-p} \sim F_{2,n-p}$$

Note that in this case the reduced model has a different response than the model full model, which means that we cannot use the `anova()` function in `R` and we need to perform the hypothesis test "by hand".

Birthweight Example

Assume now that we want to test the following hypothesis

$$\begin{cases} H_0 : \beta_{Gestation} = 0.1 \text{ and } \beta_{Length} = 0.05 \\ H_\alpha : \beta_{Gestation} \neq 0.1 \text{ or } \beta_{Length} \neq 0.05 \end{cases}$$

Here we need to first transform the response and then regress the new response with respect to the remaining parameters to get the "reduced" model. The full model always remains the same.

```
# Full Model
birthweight.mlr1 = lm(Birthweight~., birthweight2)

# "Reduced" Model
birthweight3 = birthweight2
birthweight3$new.Birthweight = birthweight3$Birthweight - 0.1*birthweight3$Gestat

birthweight.mlr.red3 = lm(new.Birthweight ~ Headcirc + smoker +  mage + mnocig +

# General Linear F Test
rss.full3 = deviance(birthweight.mlr1)
rss.red3 = deviance(birthweight.mlr.red3)

Fstat = (rss.red3 - rss.full3)/2/(rss.full3/28)

# The p-value is obtained using the pf function in R
1 - pf(Fstat, 2, 28)
```

```
## [1] 0.5856135
```

The p-value of this test is **0.59** which is greater than α and leads us to fail to reject the H_0.

Remarks

1. For testing whether a single β_k equals zero, two equivalent test statistics are available: the t test statistic and the F linear test statistic. When testing whether several β_k are equal to zero, only the general linear test statistic F is available.
2. The general linear test statistic for testing whether several X variables can be dropped from the general linear regression model can be expressed in terms of the coefficients of multiple determination for the full and reduced models. Denoting these by R_F^2 and R_R^2

respectively, we have:

$$F = \frac{R_F^2 - R_R^2}{df_R - df_F} \div \frac{1 - R_F^2}{df_F}$$

Note that this test statistic is not appropriate when the full and reduced regression models **do not** contain the intercept term β_0.

2.4.6 Confidence Intervals for β

We first start with interval estimators for the model parameters. What we have proved for $\hat{\beta}$ is summarized below:

$$\hat{\beta} = (\mathbf{X^T X})^{-1}\mathbf{X^T Y} \sim \mathcal{N}_p\left(\beta, \sigma^2 (\mathbf{X^T X})^{-1}\right)$$

Therefore, we have

(1-α)100% Confidence Interval for β_j

$$\left(\hat{\beta}_j \pm T_{n-p}(\alpha/2)\, se(\hat{\beta}_j)\right) = \left(\hat{\beta}_j \pm T_{n-p}(\alpha/2)\, \hat{\sigma}\sqrt{\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}}\right)$$

where $T_{n-p}(\alpha/2)$ is the $(1-\alpha/2)$ percentile of the student T distribution with $(n-p)$ degrees of freedom.

In `R`, we can obtain the confidence interval for β_j using the function `confint()` or by extracting the appropriate elements from the `summary()` function. Here is an example:

Birthweight Example

```
# birthweight.mlr1 is the full fitted model
confint(birthweight.mlr1, level=0.95)
```

```
##                    2.5 %      97.5 %
## (Intercept) -7.849760141 0.992476740
## Length      -0.043086734 0.094658710
## Headcirc     0.024082882 0.146103714
## Gestation    0.025557720 0.157687466
## smoker      -0.573897267 0.134249906
## mage        -0.055068892 0.023428235
## mnocig      -0.014131807 0.014550565
## mheight     -0.023842476 0.035130009
## mppwt       -0.015330916 0.032198531
## fage        -0.029759433 0.039066340
## fedyrs      -0.062367251 0.065656774
## fnocig      -0.004396726 0.012502884
## fheight     -0.032311011 0.007778007
## lowbwt      -0.655925999 0.305570152
```

The default level of confidence is 0.95, but if we want to change it this happends using the `level` argument. If we want the confidence interval for a *specific* predictor, e.g. Gestation, then we need to extract the appropriate line from the produced `data.frame` as follows:

```
confint(birthweight.mlr1)[4,]
```

```
##      2.5 %     97.5 %
## 0.02555772 0.15768747
```

Now, we will explain how we can obtain all arguments in the CI formula from relevant `R` output. As an example, we will use the `Gestation` variable that corresponds to β_4:

```
# Focus on the 4th line in the summary output
summary(birthweight.mlr1)$coef[4,]
```

```
##    Estimate  Std. Error     t value    Pr(>|t|)
## 0.091622593 0.032251827 2.840849703 0.008294721
```

Here we can see that

$$\hat{\beta}_4 = 0.092, \;\; s.e(\hat{\beta}_4) = 0.032$$

The critical value is $T_{28}(0.05/2)$

```
qt(0.975, 28)
```

```
## [1] 2.048407
```

So, a *95% confidence interval* for β_4 is

$$\beta_4 \in \left(\hat{\beta}_4 \pm T_{28}(0.05/2)s.e(\hat{\beta}_4)\right) = (0.092 \pm (2.04)(0.032)) = (0.026, 0.157)$$

2.4.7 Confidence Region for the β vector

Just as we can use estimated standard errors and test statistics form confidence intervals for a single parameter, we can also obtain a $(1-\alpha) \times 100\%$ confidence **region** for the entire vector β.

Recall that

$$\hat{\beta} = (\mathbf{X^T X})^{-1}\mathbf{X^T Y} \sim \mathcal{N}_p\left(\beta, \sigma^2(\mathbf{X^T X})^{-1}\right) \;\Leftrightarrow\; \beta - \hat{\beta} \sim \mathcal{N}_p\left(\mathbf{0}, \sigma^2(\mathbf{X^T X})^{-1}\right)$$

Thus, the quadratic form:

$$\frac{(\beta - \hat{\beta})^T\; \mathbf{X^T X}\; (\beta - \hat{\beta})}{p\,\hat{\sigma}^2} \sim F_{p,n-p}$$

Based on that, we can define

$(1-\alpha) \times 100\%$ Confidence Region for β

It is defined to be *all the points in the following ellipsoid*

$$\frac{(\beta-\hat{\beta})^T\,\mathbf{X^T X}\,(\beta-\hat{\beta})}{p\,\hat{\sigma}^2} < F(\alpha; p, n-p)$$

where $F(\alpha; p, n-p)$ is defined to be the point such that:

$$\mathbb{P}\left(F_{p,n-p} > F(\alpha; p, n-p)\right) = \alpha$$

Let's construct a confidence region for β in the `Birthweight` example:

Birthweight Example

The confidence regions have the shape of an ellipsoid. In dimensions higher than 2, we cannot visualize the result, so for *illustration purposes* we will show the confidence region for two βs. Consider `Gestation` and `Headcirc` as an example.

For two βs, the ellipse can be constructed using the command `ellipse` from the `ellipse` package.

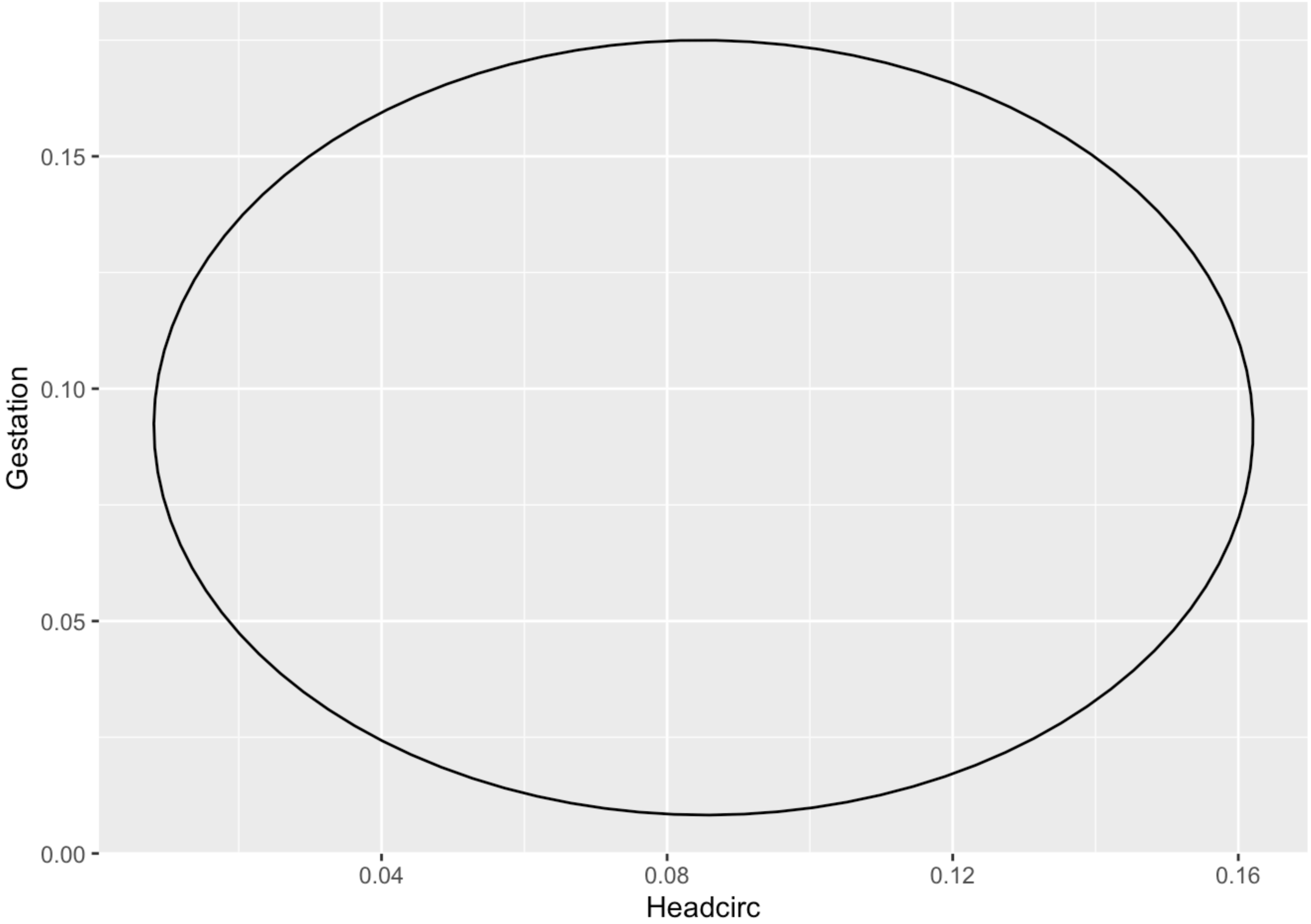
```
# Load the appropriate packages
library(ellipse)
library(ggplot2)

# Construct a data.frame to store the points necessary to draw the ellipsoid
# The default confidence level is 95%
CR = ellipse(birthweight.mlr1, c(3,4), level=0.95)
```

The `CR` object is an `matrix` (not a `data.frame`) with two columns (one for `Headcirc` and one for `Gestation`) and 100 rows -which are the one hundred points that are needed by default to construct an ellipsis in `R` . Now, we need to plot this points using `ggplot` .

```
# In order for the `ggplot` function to work, we first need CR to be a data.frame
CR = as.data.frame(CR)

CR.region1 = ggplot(data=CR, aes(x=Headcirc, y=Gestation))  +
   geom_path(aes(x=Headcirc, y=Gestation))
plot(CR.region1)
```

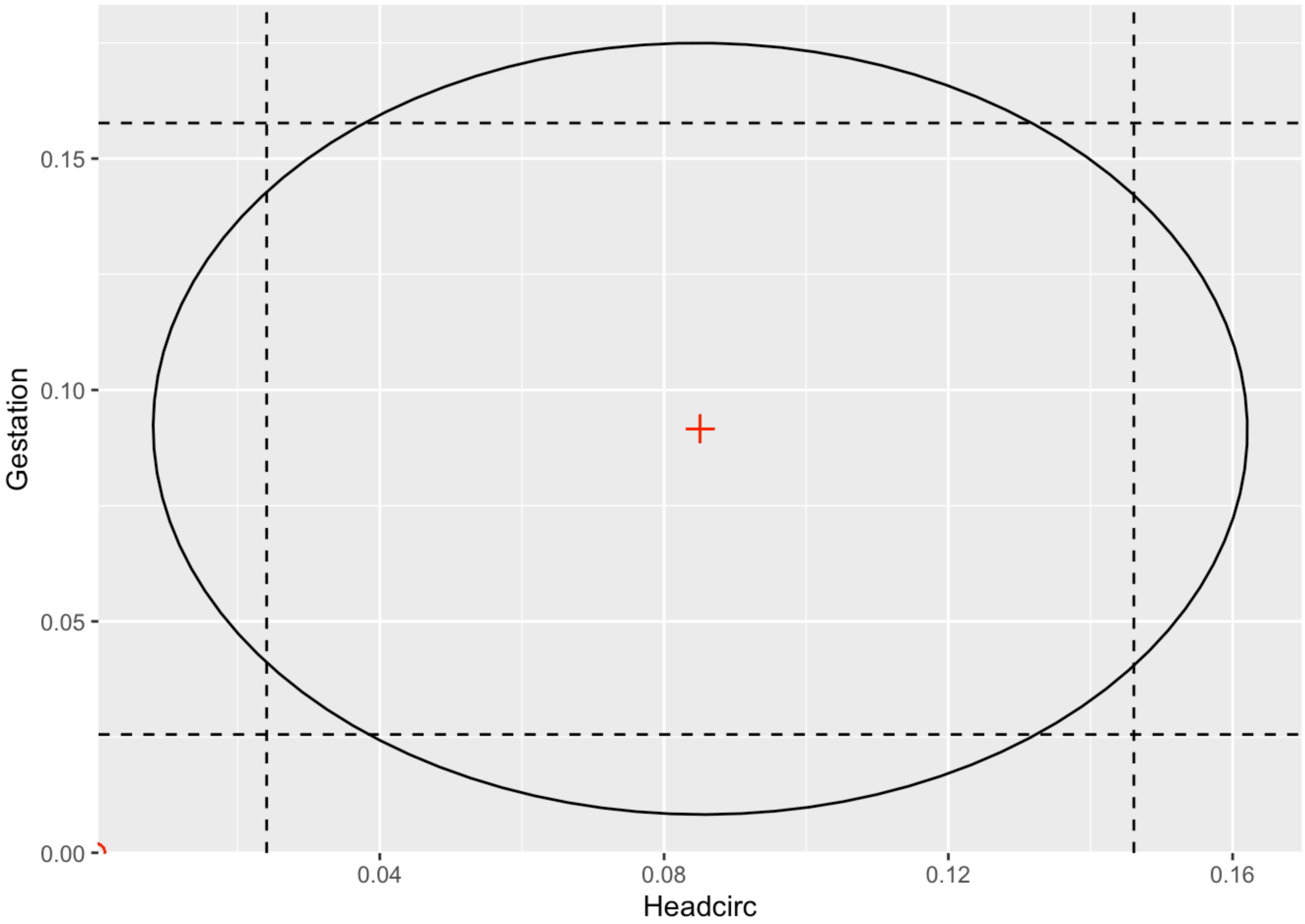


In `ggplot` , the first part `ggplot(data=CR, aes(x=Headcirc, y=Gestation))` creates the two axis and we choose `Headcirc` to be on the x axis and `Gestation` to be on the y axis. The `geom_path` connects the observations in the order in which they appear in the data (and understands them as pairs of $(x, y) = (Headcirc, Gestation)$. Here we do *not* use `geom_line` (that we have used when plotted regression lines), because `geom_line` connects the data in order of the variable on the x-axis.

We can now enhance our plot, by adding other useful features, such as the (0,0) point, the $(\hat{\beta}_{Circ}, \hat{\beta}_{Gestation})$ and the corresponding **individual** 95% confidence intervals for β_{Circ} and $\beta_{Gestation}$:

```
CR.region2 = ggplot(data=CR, aes(x=Headcirc, y=Gestation))  +
   geom_path(aes(x=Headcirc, y=Gestation)) +
   geom_point(x=coef(birthweight.mlr1)[3], y=coef(birthweight.mlr1)[4], shape=3,
  geom_point(x=0, y=0, shape=1, size=3, colour='red') + # this is the origin
  geom_hline(yintercept = confint(birthweight.mlr1, 'Gestation'), lty=2)+ # indiv
  geom_vline(xintercept = confint(birthweight.mlr1, 'Headcirc'), lty=2) # indivic

plot(CR.region2)
```



Interpretation of the Plot

Similar to a confidence interval, we can use a confidence region to test whether two LS coefficients are zero (**simultaneously**) or not, i.e.

$$\begin{cases} H_0: \beta_{Gestation} = \beta_{Headcirc} = 0 \\ H_\alpha: \text{not both } \beta_{Gestation} \text{ and } \beta_{Headcirc} \text{ equal to } 0 \end{cases}$$

using an F-test. Instead of performing this F test "by hand", we can use the duality of confidence region/interval and hypothesis tests and *we can test the hypothesis based on whether the **point (0,0)** is inside the ellipsoid or not*. If the point (0,0) is inside the confidence region, then we fail to reject the null, otherwise we reject it.

In this case, we observe that **(0,0)** is **not** included in the ellipsoid, so we reject the null and conclude that at least one of the `Headcirc` , `Gestation` variables is statistically significant, given that all other variables are in the model.

Now, the question that arises is the following:
Can we test the following hypothesis for a **single** β_j *using the ellipsoid*? The answer is **not quite**. For example, consider

$$\begin{cases} H_0: \beta_{Gestation} = 0 \\ H_\alpha: \beta_{Gestation} \neq 0 \end{cases}$$

The 95% confidence interval that corresponds to the **individual** hypothesis test for $\beta_{Gestation}$ is computed as

$$\beta_{Gestation} \in (0.026, 0.157)$$

and is depicted on our plot by the *dashed horizontal lines*. Since the **0 point** is not included in the interval, we can say that we *reject the null hypothesis* and conclude that `Gestation` is a statistically significant variable in our model (of course given that all variables are in the model).

Similarly, we can perform an **individual** hypothesis test for the `Headcirc` variable, look at the vertical dashed lines and conclude that since **0** is not included in that interval either, the `Headcirc` variable is a statistically significant variable in our model (of course given that all variables are in the model).

But, if we look at the ellipsoid, and the projections of the ellipsoid on the “x” and “y” axes, we will observe that the intervals for each of the β’s (obtained by projecting the ellipsoid on the two axes) are **wider** than the *individual* confidence intervals. This difference (in some cases) might lead to different conclusions on whether 0 is included in the interval or not.

So, why do we observe this difference and which interval should we use to draw conclusions?

The confidence intervals obtained by projecting the ellipsoid on the axes, are **joint/simultaneous** confidence intervals that correspond to the hypothesis of testing whether **both** β_js are zero or *at least one non-zero*. Our confidence level is 95% for the family of intervals - not each component’s interval. For example, we say that $\beta_1 \in (\ldots,\ldots) \text{ and } \beta_2 \in (\ldots,\ldots)\ldots$. So, the intervals we obtain by projecting the ellipsoid, do not correspond to a 95% confidence level for *each* individual β_j. In fact one can show that they produce individual CIs with **higher** confidence levels, and this is why those intervals are typically **wider** than the ones we construct when we focus on a single coefficient β_j.

To conclude this section, we replicate the plot in the Lectures that includes three ellipses corresponding to 3 different confidence levels: 95%, 99%, 99.8% using the code below:

R Code Explanation

First, we construct the ellipsoids for 3 different *levels*: 95% (default), 99% and 99.8%. The example is shown for `Gestation` and `Headcirc`.

```
CR95 = ellipse(birthweight.mlr1, c(3,4)) # the 95% is the default
CR99 = ellipse(birthweight.mlr1, c(3,4), level=0.99)
CR998 = ellipse(birthweight.mlr1, c(3,4), level=0.998)
```

Next, we combine the three CR variables in a new one, `newCR` where the first rows correspond to CR95, the next to CR99, and the last to CR998.

```
newCR = rbind(CR95, CR99, CR998);
```

This object is a `matrix`, so we make it a `data.frame` so that it can be read by the `ggplot()` function, and we add names to the columns:

```r
newCR = data.frame(newCR);
names(newCR) = c("Headcirc","Gestation");
```

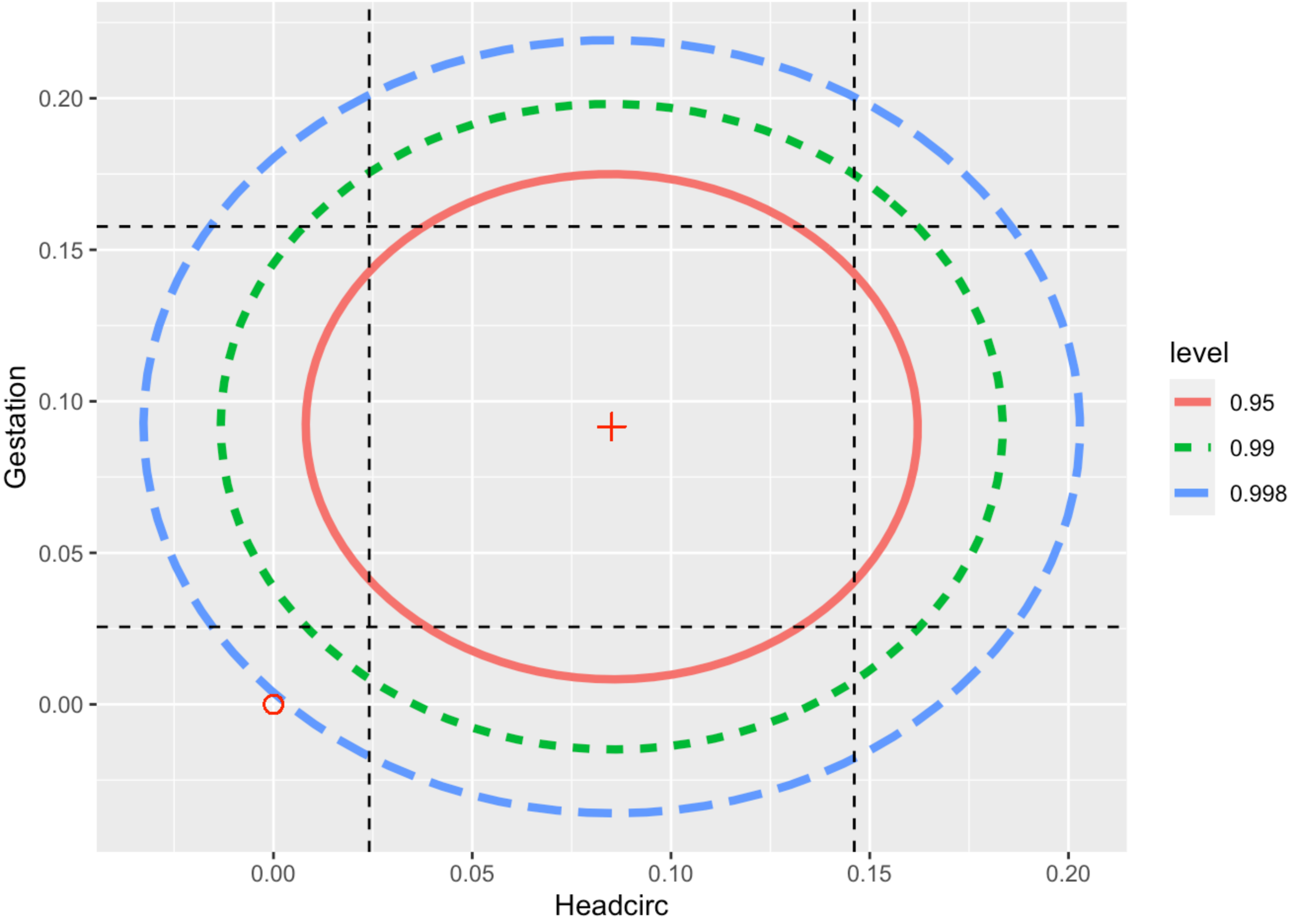
For plotting purposes, we need to add a column that indicates the confidence level for each value called `level` :

```r
newCR[, 'level']=as.factor(c(rep(0.95, dim(CR95)[1]),
                              rep(0.99, dim(CR99)[1]),
                              rep(0.998, dim(CR998)[1])));
```

Now, we are ready to call `ggplot()`

```r
CR.region3 = ggplot(data=newCR, aes(x=Headcirc, y=Gestation, colour=level)) +
  geom_path(aes(linetype=level), size=1.5) + # this will create different lines b
  geom_point(x=coef(birthweight.mlr1)[3], y=coef(birthweight.mlr1)[4], shape=3, s
  geom_point(x=0, y=0, shape=1, size=3, colour='red') + # add the origin
  geom_hline(yintercept = confint(birthweight.mlr1, 'Gestation'), lty=2)+ # add t
  geom_vline(xintercept = confint(birthweight.mlr1, 'Headcirc'), lty=2) # add the
```

```r
 plot(CR.region3)
```


Gestation
0.20
0.15
0.10
0.05
0.00
0.00
0.05
0.10
0.15
0.20
Headcirc
level
0.95
0.99
0.998