Inference with SLR Tutorial

For this tutorial, we will continue to work with the dataset `elmhurst` from the `openintro` package in R.

```
library(tidyverse)
library(openintro)
Data<-openintro::elmhurst
```

The key pieces of information are:

- A random sample of 50 students (all freshman from the 2011 class at Elmhurst College).
- Family income of the student (units are missing).
- Gift aid, in $1000s.

We want to explore how family income may be related to gift aid, in a simple linear regression framework.

Hypothesis test for β_1 (and β_0)

Applying the `summary()` function to `lm()` gives the results of hypothesis tests for β_1 and β_0:

```
##Fit a regression model
result<-lm(gift_aid~family_income, data=Data)

##look at t stats and F stat
summary(result)
```

```
##
## Call:
## lm(formula = gift_aid ~ family_income, data = Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1128  -3.6234  -0.2161   3.1587  11.5707
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.31933    1.29145  18.831  < 2e-16 ***
## family_income -0.04307    0.01081  -3.985 0.000229 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.783 on 48 degrees of freedom
## Multiple R-squared:  0.2486, Adjusted R-squared:  0.2329
## F-statistic: 15.88 on 1 and 48 DF,  p-value: 0.0002289
```

Under coefficients, we can see the results of the hypothesis tests for β_1 and β_0. Specifically, for β_1:

- $\hat{\beta}_1 =$ -0.0430717
- $se(\hat{\beta}_1) =$ 0.0108095
- the test statistic is $t =$ -3.984621
- the corresponding p-value is 2.2887345×10^{-4}

You can work out the p-value using R (slight difference due to rounding):

```r
##pvalue
2*pt(-abs(-3.985), df = 50-2)
```

```
## [1] 0.0002285996
```

Or find the critical value using R:

```r
##critical value
qt(1-0.05/2, df = 50-2)
```

```
## [1] 2.010635
```

Either way, we end up rejecting the null hypothesis. The data support the claim that there is a linear association between gift aid and family income.

Note:

- the t tests for regression coefficients are based on $H_0 : \beta_j = 0, H_a : \beta_j \neq 0$. The reported p-value is based on this set of null and alternative hypotheses. If your null and alternative hypotheses are different, you will need to compute your own test statistic and p-value.
- For SLR, the two-sided t test for β_1 gives the exact same result as the ANOVA F test. Notice the p-values are the same. The F statistic of 15.88 is the squared of the t statistic, $(-3.985)^2$.

Confidence interval for β_1 (and β_0)

To find the 95% confidence intervals for the coefficients, we use the `confint()` function:

```r
##to produce 95% CIs for all regression coefficients
confint(result,level = 0.95)
```

```
##                     2.5 %      97.5 %
## (Intercept)   21.72269421 26.91596380
## family_income -0.06480555 -0.02133775
```

The 95% CI for β_1 is (-0.0648056, -0.0213378). We have 95% confidence that for each additional thousand dollars in family income, the predicted gift aid decreases between $21.3378 and $64.8056.

Confidence interval for mean response for given x

Suppose we want a confidence interval for the average gift aid for Elmhurst College students with family income of 80 thousand dollars. We can use the `predict()` function:

```r
##to produce 95% CI for the mean response when x=80,
newdata<-data.frame(family_income=80)
predict(result,newdata,level=0.95, interval="confidence")
```

```
##       fit      lwr      upr
## 1 20.8736 19.43366 22.31353
```

The 95% CI for the mean gift aid for students with family income of 80 thousand dollars is (19.4336609, 22.3135327). We have 95% confidence the mean gift aid for students with family income of 80 thousand dollars is between $19 433.66 and $22 313.53.

Prediction interval for a response for a given x

For a prediction interval for the gift aid of an Elmhurst College student with family income of 80 thousand dollars:

```
##and the 95% PI for the response of an observation when x=80
predict(result,newdata,level=0.95, interval="prediction")
```

```
##       fit      lwr      upr
## 1 20.8736 11.15032 30.59687
```

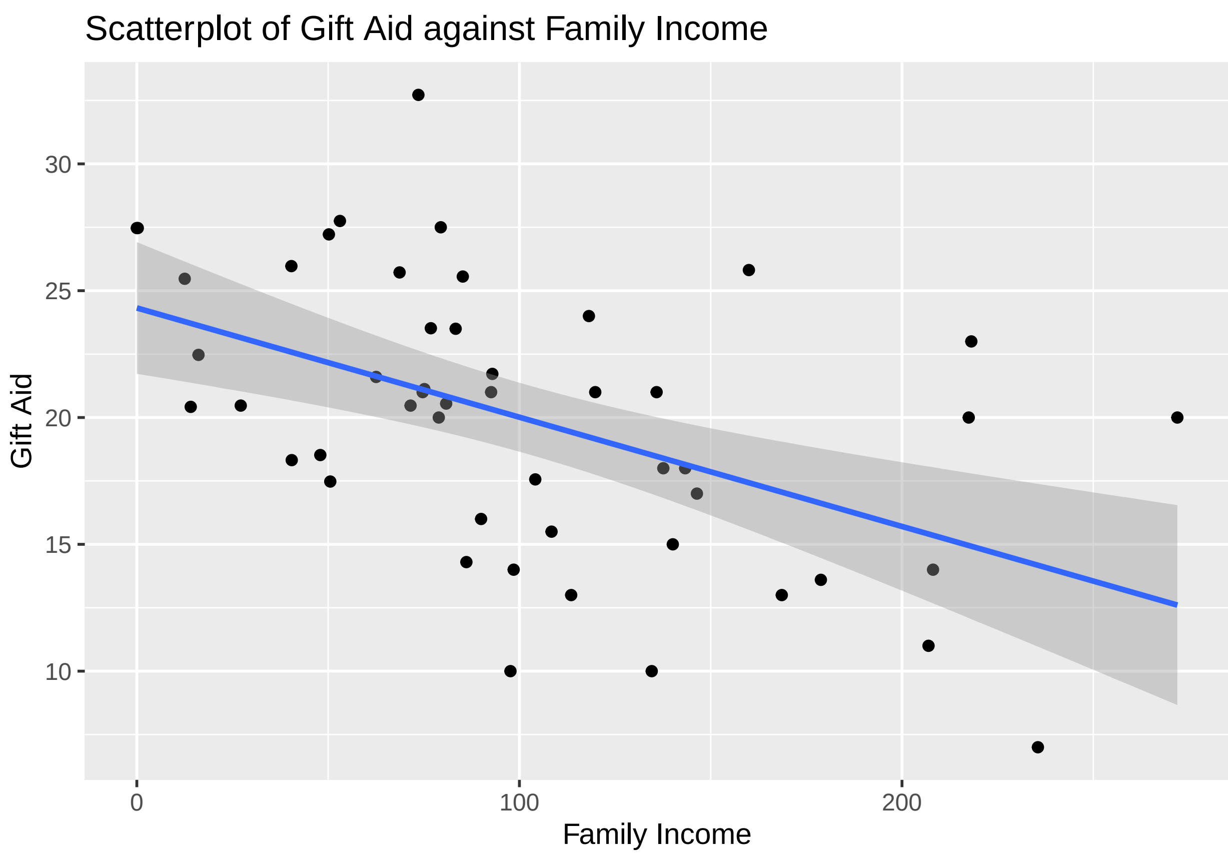
We have 95% confidence that for an Elmhurst College student with family income of 80, this student's gift aid is between \$11 150.32 and \$30 596.87.

Visualization of CI for mean response given x and PI of response given x

When using the `ggplot()` function to create a scatterplot, we can overlay the SLR equation by adding a layer via `geom_smooth(method = lm)`. By default, the CI for the mean response for each value of the predictor gets overlaid as well. In the previous tutorial, we removed this by adding `se=FALSE` inside `geom_smooth()`:

```
##regular scatterplot
##with regression line overlaid, and bounds of CI for mean y
ggplot2::ggplot(Data, aes(x=family_income, y=gift_aid))+
  geom_point() +
  geom_smooth(method=lm)+
  labs(x="Family Income",
       y="Gift Aid",
       title="Scatterplot of Gift Aid against Family Income")
```

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



Overlaying prediction intervals require a bit more work. We need to compute the lower and upper bounds of the PI for each value of the predictor:

```
##find PIs for each observation
preds <- predict(result, interval="prediction")
```

```
## Warning in predict.lm(result, interval = "prediction"): predictions on current data refer to _future_
```

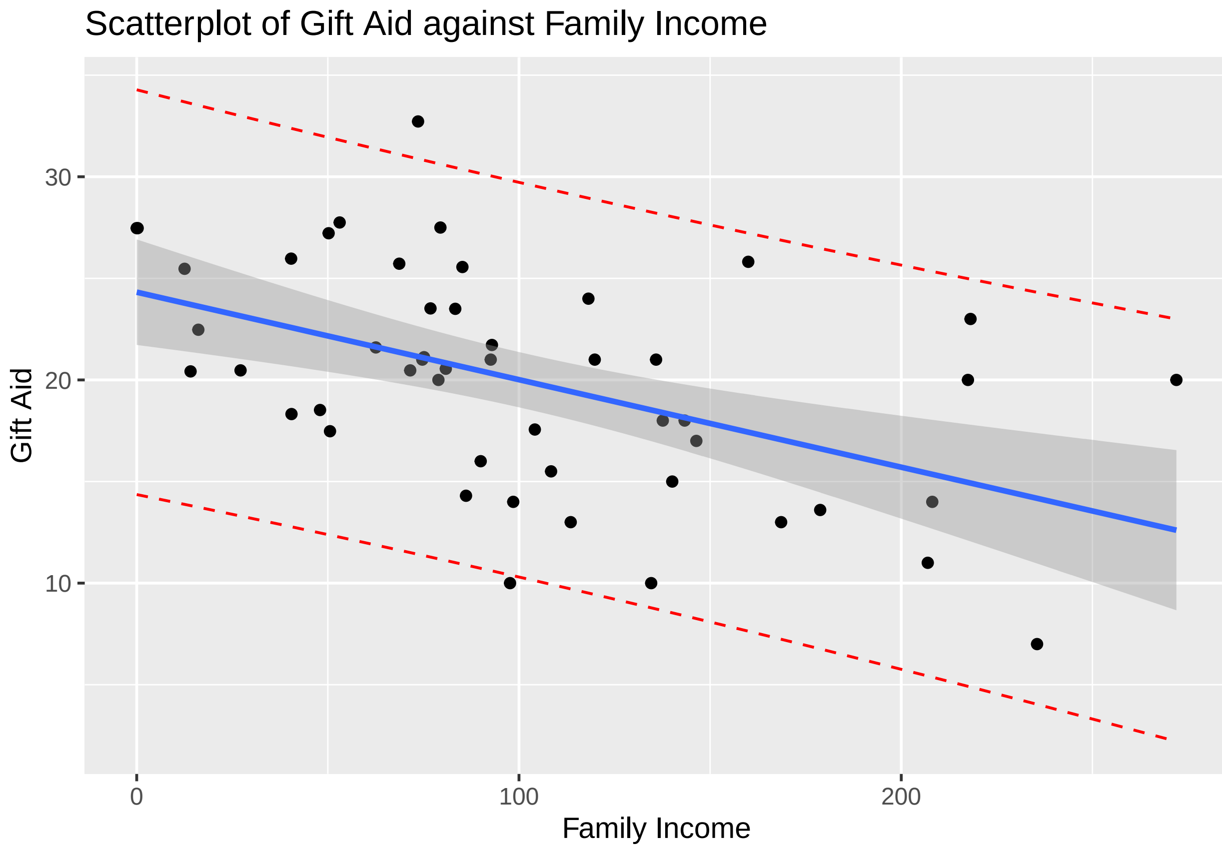
Previously, when we used the `predict()` function, we provided the numerical value of x to make a prediction on. If this is not supplied, the function will use all the current values of x to make predictions, and will actually print out a warning message. For our purpose, this is not an issue since this is exactly what we want.

We then add `preds` to the data frame in order to overlay the lower and upper bounds on the scatterplot, by adding extra layers via `geom_line()` in the `ggplot()` function:

```
##add preds to data frame
Data<-data.frame(Data,preds)

##overlay PIs via geom_line()
ggplot2::ggplot(Data, aes(x=family_income, y=gift_aid))+
  geom_point() +
  geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
  geom_line(aes(y=upr), color = "red", linetype = "dashed")+
  geom_smooth(method=lm)+
  labs(x="Family Income",
       y="Gift Aid",
       title="Scatterplot of Gift Aid against Family Income")
```

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



As mentioned in the notes, the CI captures the location of the regression line, whereas the PI captures the data points.