SDS358: Applied Regression Analysis

Day 9: SLR: Confidence

Dr. Michael J. Mahometa

Agenda for Today:

- Inference testing recap
- Confidence of the regression *slope*
- Confidence for the *model*
  - "Confidence" Interval
  - "Prediction"" Interval
- Tying it all together

Research Queastion:

Given that perceived social support significantly predicts life satisfaction in a sample of unemployed Spanish adults, what is the range of life satisfaction when perceived social support is equal to four?

Recap?

- Our estimate of sigma gets us estiamtes for the s.e. for both b_0 and b_1
- This gets us a way to test both simple regression parameters
- This will come in handy as we add additional independents
- F-statistic: Overall model
- t-statistic: Individual paramter = to 0
  - In the case of simple regression: does our simple slope differ from zero
- t-statistic: A Pearson Correlation compared to zero.

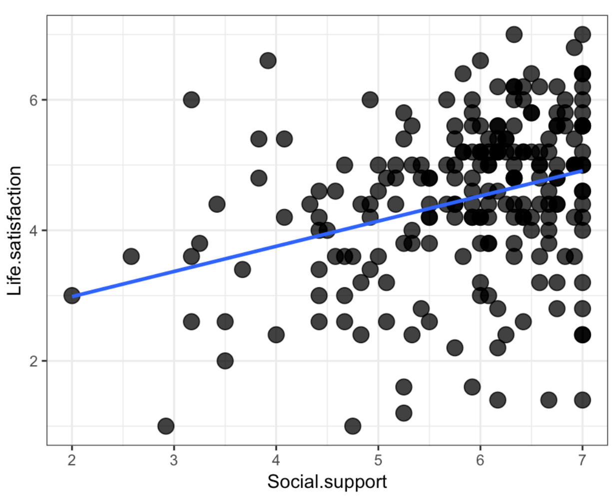
Remember:

$$\text{F-statistic} = \frac{MS_{\text{Regression}}}{MS_{\text{Error}}}$$

$$t_1 = \frac{b_1}{s.e.(b_1)}$$

Remember:

```r
library(SDSRegressionR)
simpleScatter(unemp, Social.support, Life.satisfaction, line = TRUE)
```



Remember:

```
ls_mod <- lm(Life.satisfaction ~ Social.support, unemp)
summary(ls_mod)
```

```
##
## Call:
## lm(formula = Life.satisfaction ~ Social.support, data = unemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5124 -0.5633  0.1459  0.7387  2.8750
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.21363    0.46822   4.728 4.08e-06 ***
## Social.support  0.38554    0.07896   4.883 2.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.176 on 217 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.09899,    Adjusted R-squared:  0.09484
## F-statistic: 23.84 on 1 and 217 DF,  p-value: 2.028e-06
```


But wait! There's more!

- Typically, we test a Null Hypothesis of $H_0 = 0$ for the slope. But what about *a different* slope value…can you do that? What would need to be changed?

But wait! There's more!

- We can use our knowledge (and the knowledge of the t-test for slope) to use an alternative from a "slope = 0". Say slope=0.6…

$$t_1 = \frac{b_1 - b_1^0}{s.e.(b_1)}$$

But wait! There's more!

- We can use our knowledge (and the knowledge of the t-test for slope) to use an alternative from a "slope = 0". Say slope=0.6…

$$t_1 = \frac{b_1 - b_1^0}{s.e.(b_1)}$$

$$t_1 = \frac{0.3855 - 0.6}{0.0790}$$

But wait! There's more!

- We can use our knowledge (and the knowledge of the t-test for slope) to use an alternative from a "slope = 0". Say slope=0.6…

$$t_1 = \frac{b_1 - b_1^0}{s.e.(b_1)}$$

$$t_1 = \frac{0.3855 - 0.6}{0.0790}$$

$$t_1 = -2.715$$

But wait! There's more!

- We can use our knowledge (and the knowledge of the t-test for slope) to use an alternative from a "slope = 0". Say slope=0.6…

```
t <- (0.3855 - 0.6)/0.0790
t
```

```
## [1] -2.71519
```

```
pt(abs(t), 217, lower.tail = FALSE) * 2
```

```
## [1] 0.007156942
```


But wait! There's more!

- *Or* we can use some help from another r package….

```
library(car)
linearHypothesis(ls_mod, c("Social.support = 0.6"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## Social.support = 0.6
##
## Model 1: restricted model
## Model 2: Life.satisfaction ~ Social.support
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    218 310.37
## 2    217 300.17  1    10.204 7.3769 0.007139 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On to Confidence

- Remember the *Confidence Interval* for a single value (Intro Stats):

$$\bar{x} \pm t_{(n-1,\alpha/2)} \times SE_{\bar{x}}$$

- How would this equation look if we wanted confidence of the *Slope* of a Simple Linear Regression model?

Confidence

- Now that we have the $s.e.(b_0)$ and $s.e.(b_1)$, we can also come up with confidence interevals.

$$b_1 \pm t_{(n-p,\alpha/2)} \times s.e(b_1)$$

Confidence

- Now that we have the $s.e.(b_0)$ and $s.e.(b_1)$, we can also come up with confidence interevals.

```
0.3855 + abs(qt(.025, 217)) * 0.0790
```

```
## [1] 0.5412055
```

```
0.3855 - abs(qt(.025, 217)) * 0.0790
```

```
## [1] 0.2297945
```


Confidence

- Confidence for the slope: "I am 95% confident that the true slope of the underlying population from which this sample comes, lies between these two values."
- If it catches zero, then the slope is not significant.

Confidence

- Let's see this in action…
- RStudio

Confidence

- Notice: for the multiple slopes, the *predicted* values were tighter in the center of the graph, and a bit larger at the ends…

Confidence

- We can ask R to give us the confidence interval of the slope in the model:

```
ls_mod <- lm(Life.satisfaction ~ Social.support, unemp)
summary(ls_mod)
```

```
##
## Call:
## lm(formula = Life.satisfaction ~ Social.support, data = unemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5124 -0.5633  0.1459  0.7387  2.8750
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.21363    0.46822   4.728 4.08e-06 ***
## Social.support  0.38554    0.07896   4.883 2.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.176 on 217 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.09899,    Adjusted R-squared:  0.09484
## F-statistic: 23.84 on 1 and 217 DF,  p-value: 2.028e-06
```


Confidence

- We can ask R to give us the confidence interval of the slope in the model:

```
confint(ls_mod)
```

```
##                    2.5 %   97.5 %
## (Intercept)    1.2907843 3.136483
## Social.support 0.2299132 0.541167
```

"I am 95% confident that the true slope of the underlying population from which this sample comes, lies between these two values."

What does this tell us…

- The sample slopes will all be different from one another.
- The "predition" of y is better a the mean of x, less so at the ends of the distribution of x.

Two kinds of "confidence" for the regression model

- When it comes to using the model for prediction purposes, we have a choice:
  - Predict a single new observation (y) at value x.
  - Predict the mean of y, given a value of x.

Two kinds of "confidence" for the regression model

- Single observation: prediction interval
- Mean observation: confidence interval

The Prediction Interval

- Used to predict the value of an *individual* y value for a given x (x_0).

$$s.e.(\hat{y}_0) = \hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\Sigma(x - \bar{x})^2}}$$

The Confidence Interval

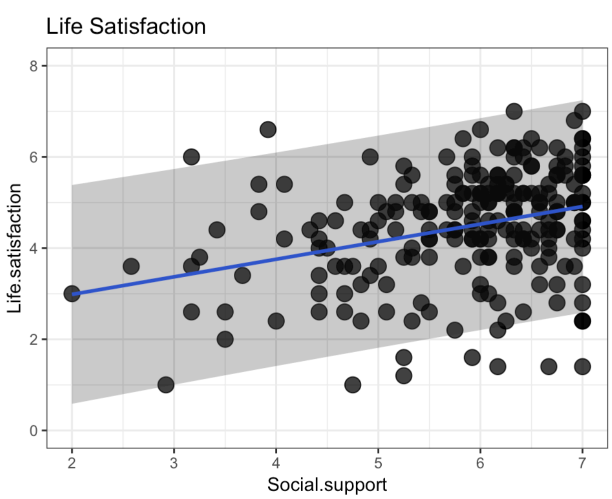
- *Not* the same as the Confidence Interval for the slope
- Used to predict the value of a *mean* y value for a given x (x_0).

$$s.e.(\hat{\mu}_0) = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\Sigma(x - \bar{x})^2}}$$

The Prediction Interval

In R

- Visualize it first:



The Prediction Interval

In R

```
# For an x = to the mean
mean(unemp$Social.support, na.rm=TRUE)
```

```
## [1] 5.841136
```

```
mn_SS <- data.frame(Social.support = mean(unemp$Social.support, na.rm=TRUE))
predict(ls_mod, mn_SS, interval="prediction")
```

```
##        fit      lwr      upr
## 1 4.465626 2.142253 6.788999
```

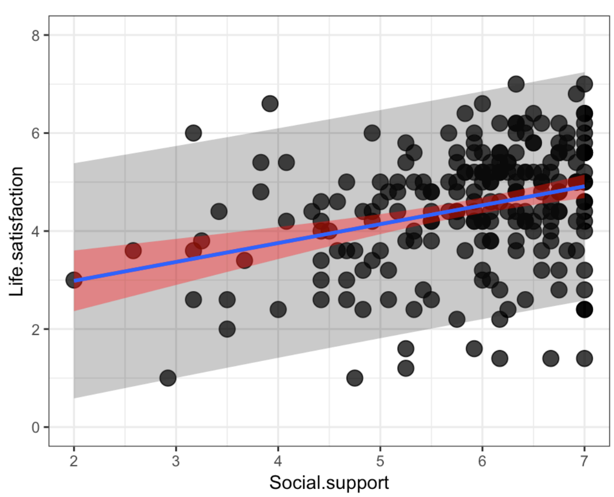
```
# For an x = to 4
nw_SS <- data.frame(Social.support = 4)
predict(ls_mod, nw_SS, interval="prediction")
```

```
##        fit      lwr     upr
## 1 3.755794 1.414769 6.09682
```


The Confidence Interval

In R

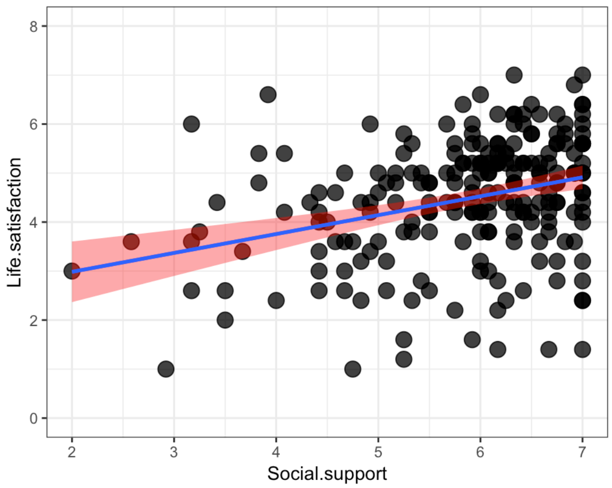
- Visualize it first:



The Confidence Interval

In R

- Visualize it first:



The Confidence Interval

In R

```
# For an x = to the mean
mn_SS <- data.frame(Social.support = mean(unemp$Social.support, na.rm=TRUE))
predict(ls_mod, mn_SS, interval="confidence")
```

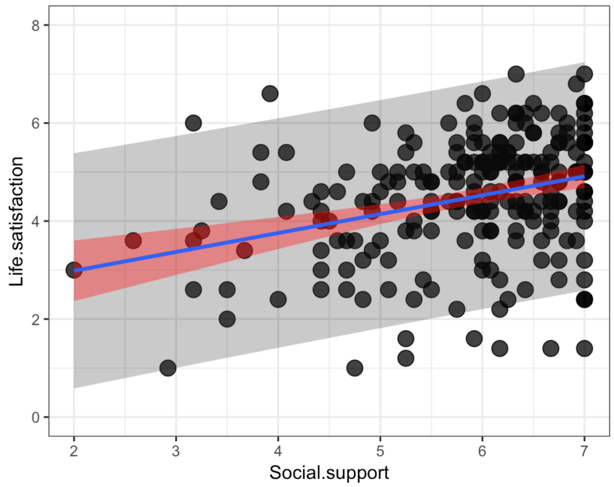
```
##        fit      lwr      upr
## 1 4.465626 4.308984 4.622268
```

```
# For an x = to 4
nw_SS <- data.frame(Social.support = 4)
predict(ls_mod, nw_SS, interval="confidence")
```

```
##        fit      lwr      upr
## 1 3.755794 3.428873 4.082715
```

Two Possible Intervals

- Visual Representation:



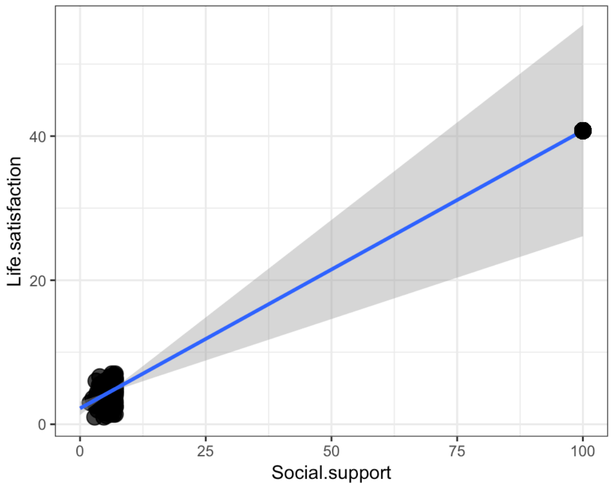
Riddle me this….

```
# For an x = 10.00
nw_SS <- data.frame(Social.support = 100.00)
predict(ls_mod, nw_SS, interval="confidence")
```

```
##        fit      lwr      upr
## 1 40.76764 26.11357 55.42172
```

Use with caution

- Social Support x=100?



Use with caution

- Confidence intervals around the slope (Good to use)
  - 95% confident the *true* population slope lies between these two values
- Confidence intervals for model prediction
  - 95% confident that the true $\hat{y}$ (for a given x (x_i)) lies between these two values
- Prediction intervals for the model prediction
  - 95% confident that the a new *single observation* of y (for a given x (x_i) lies between these two values

Use with caution

- *BUT,* we can only use the model prediction intervals for data of (x_i) that is in the *range of x in the model.*
- Extrapolation past the data points of the model = **BAD**

Recap Simple Linear Regression

- SLR extends Pearson (both are based on a linear relationship).
  - (r) tells us about the strength of the relationship
  - SLR tells use two things:
    - Overall model fit (explained variance) with an F-statistic
    - Simple slope test (deviation from zero) with a t-statistic
  - SLR shows us the scale change in y given a unit change in x (puts context, where Pearson is "scale free")

By the way…

- They all match in SLR:

```
cor(select(unemp, Social.support, Life.satisfaction),
         use="pairwise.complete.obs")
```

```
##                   Social.support Life.satisfaction
## Social.support         1.0000000         0.3146274
## Life.satisfaction      0.3146274         1.0000000
```


By the way…

- They all match in SLR:

```
summary(ls_mod)
```

```
##
## Call:
## lm(formula = Life.satisfaction ~ Social.support, data = unemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5124 -0.5633  0.1459  0.7387  2.8750
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.21363    0.46822   4.728 4.08e-06 ***
## Social.support  0.38554    0.07896   4.883 2.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.176 on 217 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.09899,    Adjusted R-squared:  0.09484
## F-statistic: 23.84 on 1 and 217 DF,  p-value: 2.028e-06
```

AND if we *scale*…

```
unemp$z_SS <- scale(unemp$Social.support)
unemp$z_LS <- scale(unemp$Life.satisfaction)

ls_mod_std <- lm(z_LS ~ z_SS, data=unemp)
round(ls_mod_std$coefficients, 4)
```

```
## (Intercept)        z_SS
##      0.0031      0.3143
```


Summary

- Running an SLR:
  1. Possible correlation matrix
  2. SLR initial model
  3. Look for outliers (Cook's D is arguably the most important)
  4. Re-run without influential points
  5. Report
     1. Removal of outliers
     2. Overall model F
     3. Slope interpretation (with c.i.)