Supplement for Lecture 10: Partitioning Variability

Load and Clean Data

Variables of Interest in `fatal` - *adj_fatal* = Number of Vehicle Fatalities Per 1,000 People - *yd* = Percent of Drivers 15 - 24

```r
data("Fatalities") # Load Data

fatal = Fatalities[,c("fatal","pop","youngdrivers")]
fatal$adj_fatal = (fatal$fatal/fatal$pop)*1000
fatal$yd=fatal$youngdrivers*100

fatal$fatal=NULL
fatal$pop=NULL
fatal$youngdrivers=NULL

head(fatal)
```

```
##   adj_fatal      yd
## 1  0.212836 21.1572
## 2  0.234848 21.0768
## 3  0.233643 21.1484
## 4  0.219348 21.1140
## 5  0.266914 21.3400
## 6  0.271859 21.5527
```

Fit Linear Regression Model

```r
#Fit Linear Regression Model
mod = lm(adj_fatal~yd,data=fatal)

#Results from t-test for slope
summary(mod)
```

```
##
## Call:
## lm(formula = adj_fatal ~ yd, data = fatal)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.119634 -0.040335 -0.007417  0.034376  0.205392
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 0.104129   0.022873   4.552 0.00000744 ***
## yd          0.005374   0.001219   4.407 0.00001414 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05551 on 334 degrees of freedom
## Multiple R-squared:  0.05495,    Adjusted R-squared:  0.05212
## F-statistic: 19.42 on 1 and 334 DF,  p-value: 0.00001414
```

```
#Results from ANOVA F-test for Effectiveness of SLR Model
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: adj_fatal
##            Df  Sum Sq  Mean Sq F value     Pr(>F)
## yd          1 0.05985 0.059853  19.422 0.00001414 ***
## Residuals 334 1.02930 0.003082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Relationship between test statistics between both tests
summary.out=summary(mod)
tstat=summary.out$coefficients[2,3]
Fstat=as.numeric(summary.out$fstatistic[1])

tstat^2
```

```
## [1] 19.42185
```

```
Fstat
```

```
## [1] 19.42185
```

Correlation Test and R^2

```
# Correlation Test (Compare p-value to previous p-values)
cor.test(x=fatal$adj_fatal,y=fatal$yd)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 4.407, df = 334, p-value = 0.00001414
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1307062 0.3330626
## sample estimates:
##       cor
## 0.2344221
```

```
cor.test(y=fatal$adj_fatal,x=fatal$yd)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 4.407, df = 334, p-value = 0.00001414
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.1307062 0.3330626
## sample estimates:
##       cor
## 0.2344221
```

```r
# R-squared
cor(x=fatal$adj_fatal,y=fatal$yd)^2
```

```
## [1] 0.05495373
```

```r
summary.out$r.squared
```

```
## [1] 0.05495373
```

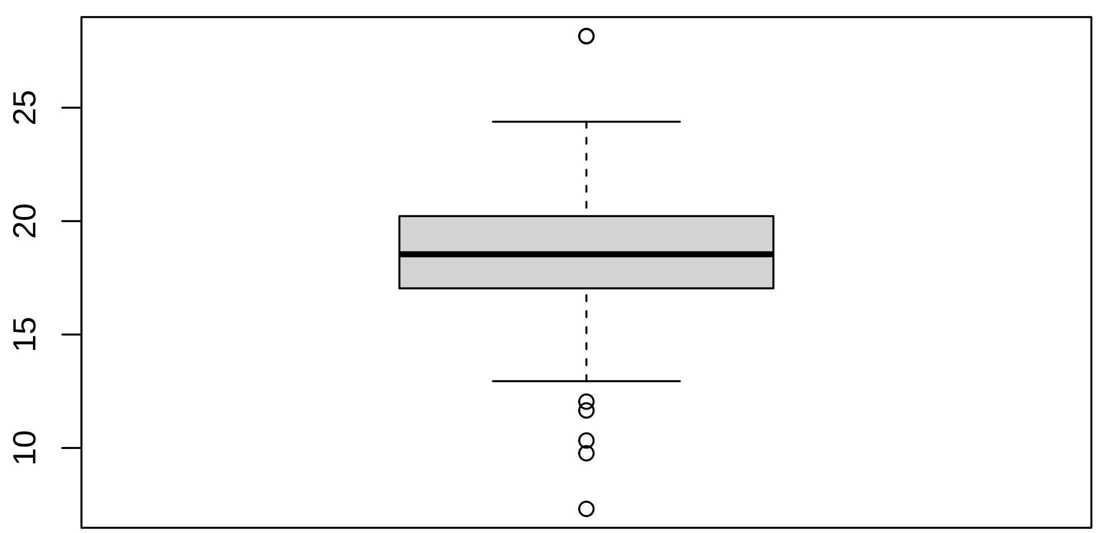
Interpretation: We are able to explain 5% of the variation in the number of fatalities per 1000 people in a state by the simple linear regression model based on the percent of young drivers (ages 15-24) in state.

Prediction for a State Where 20% of the drivers are ages 15 to 24

```r
#Are We Extrapolating?
quantile(x=fatal$yd)
```

```
##       0%      25%      50%      75%     100%
##  7.31370 17.03733 18.53875 20.21850 28.16250
```

```r
boxplot(x=fatal$yd)
```



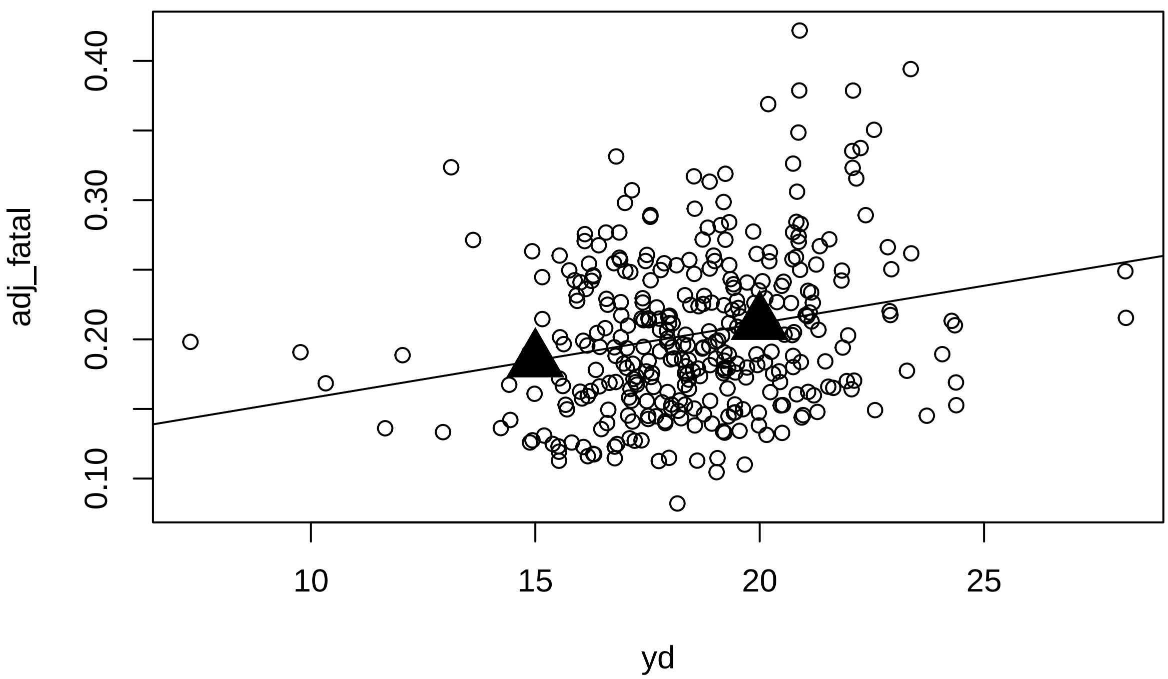
```
#Predict when yd=0.2
xstar=20
fit.yint = summary.out$coefficients[1,1]
fit.slope = summary.out$coefficients[2,1]

fit.yint+fit.slope*xstar
```

```
## [1] 0.2116054
```

```
#Alternative Way to predict when yd=20 or yd=15
unknown = data.frame(yd=c(20,15))
predict(mod,newdata=unknown)
```

```
##         1         2
## 0.2116054 0.1847363
```

```
#Plot of predictions
plot(adj_fatal~yd,data=fatal)
abline(mod)
points(x=c(20,15),y=predict(mod,newdata=unknown),pch=17,cex=3)
```



Interpretation of Prediction: If 20% of the drivers in a state were between the ages of 15 to 24, then we would predict or expect the number of vehicle fatalities to be AROUND 0.21 per 1,000 people.

Alternative Interpretation: If an infinite number of states each had 20% of their drivers between the ages of 15 and 24, the average number of vehicle fatalities per 1,000 people across all those states is ESTIMATED to be 0.21.

Confidence Interval

```
predict(mod,unknown,interval="confidence")
```

```
##         fit       lwr       upr
## 1 0.2116054 0.2047585 0.2184523
## 2 0.1847363 0.1742595 0.1952132
```

Interpretation of Confidence Interval: We are 95% confident that the average number vehicle fatalities per 1000 people in state is somewhere between 0.20 and 0.22 whenever a state's proportion of young drivers is 20%.

Prediction Interval

```
predict(mod,unknown,interval="prediction")
```

```
##         fit        lwr       upr
## 1 0.2116054 0.10219091 0.3210199
## 2 0.1847363 0.07503485 0.2944378
```

Interpretation of Prediction Interval: For a state with 20% of the drivers being young (ages 15-24), I am 95% confident that the actual number of vehicle fatalities per 1000 people in that specific state is somewhere between 0.10 and 0.32.

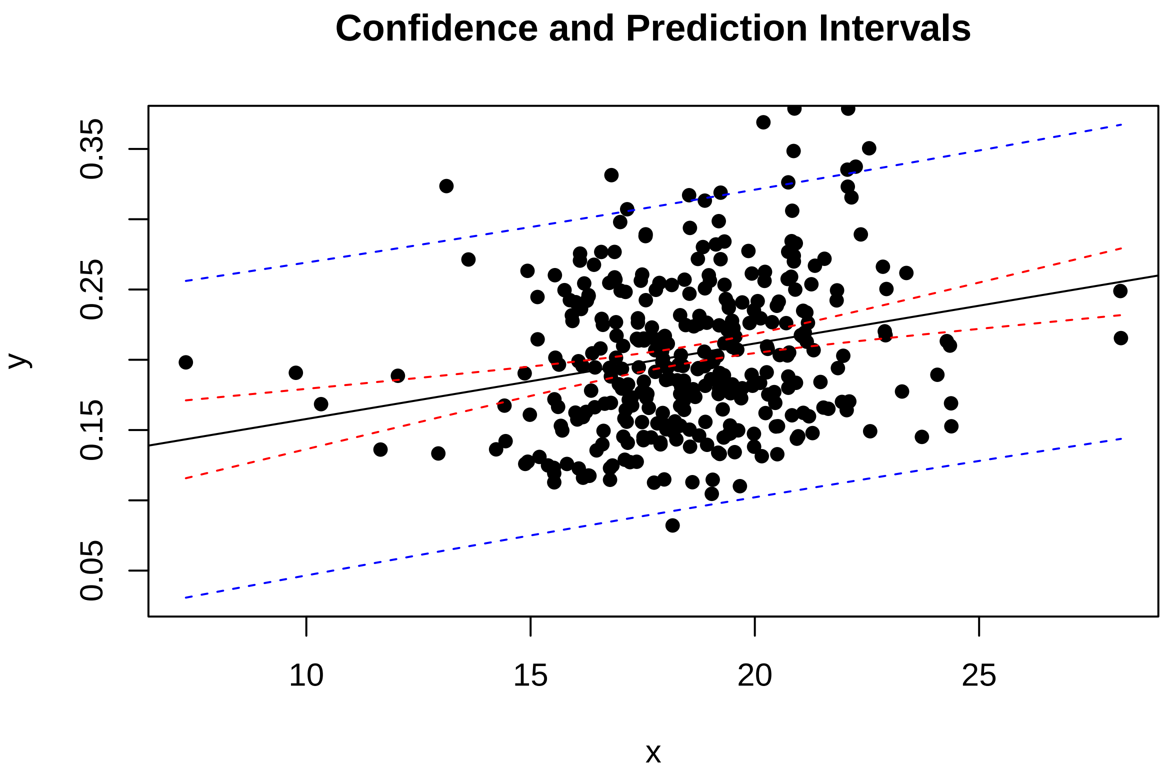
Comparison of Both Intervals

```
CIPIPlot
```

```
## function (x, y, level = 0.95, xname = "x", yname = "y")
## {
##     mymodel = lm(y ~ x)
##     n = length(x)
##     tstar = qt(1 - (1 - level)/2, n - 2)
##     xbar = mean(x)
##     ssx = sum((x - xbar)^2)
##     b0 = mymodel$coeff[1]
##     b1 = mymodel$coeff[2]
##     Se = summary(mymodel)$sigma
##     y1 = b0 + b1 * min(x) - tstar * Se * sqrt(1 + 1/n + (min(x) -
##         xbar)^2/ssx)
##     y2 = b0 + b1 * max(x) + tstar * Se * sqrt(1 + 1/n + (max(x) -
##         xbar)^2/ssx)
##     plot(y ~ x, main = "Confidence and Prediction Intervals",
##         pch = 16, ylim = sort(c(y1, y2)), xlab = xname, ylab = yname)
##     abline(mymodel)
##     curve(b0 + b1 * x + tstar * Se * sqrt(1/n + (x - xbar)^2/ssx),
##         add = T, lty = 2, col = "red", lwd = 1)
##     curve(b0 + b1 * x - tstar * Se * sqrt(1/n + (x - xbar)^2/ssx),
##         add = T, lty = 2, col = "red", lwd = 1)
##     curve(b0 + b1 * x + tstar * Se * sqrt(1 + 1/n + (x - xbar)^2/ssx),
##         add = T, lty = 2, col = "blue", lwd = 1)
##     curve(b0 + b1 * x - tstar * Se * sqrt(1 + 1/n + (x - xbar)^2/ssx),
##         add = T, lty = 2, col = "blue", lwd = 1)
```

```
## }
```

```r
CIPIPlot(x=fatal$yd, y=fatal$adj_fatal)
```



The red dashed lines represents are uncertainty in the fitted line itself. We are 95% confident that the fitted line to the population lies somewhere in this interval. This is based on our assumptions for the linear regression model to be reasonable.

The blue dashed lines represent are uncertainty in using the line to predict. We are 95% confident that a prediction of y for any x value is somewhere in this interval. If are the assumptions of the linear regression model are reasonable, you would expect approximately 95% of the data points in the population to lie within the blue interval.