Data Analysis in R

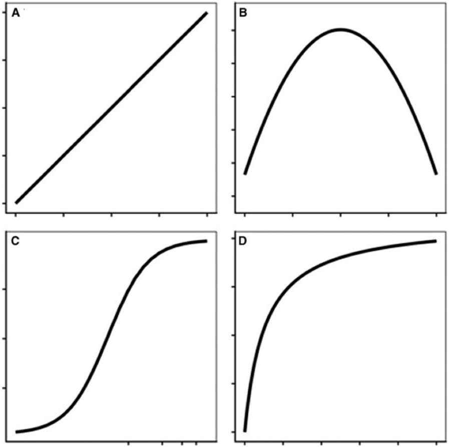
Linear Regression

Maxwel Coura Oliveira, PhD

University of Wisconsin-Madison

Common regression lines

- What does these curves is telling us?



Types of regression lines

- Nested: Models that are a particular case of each other and have identical terms, whereas one must have at least one additional term (e.g. three- and four-parameter log-logistic models)
- Non-nested: Models with different structure and parameters, such as an exponential decay and a rectangular hyperbola model.

Linear regression

- Linear regression is used to predict the value of an outcome variable Y based on one or more input predictor variables X.
- The aim is to establish a linear relationship (a mathematical formula) between the predictor variable(s) and the response variable, so that, we can use this formula to estimate the value of the response Y, when only the predictors (Xs) values are known.

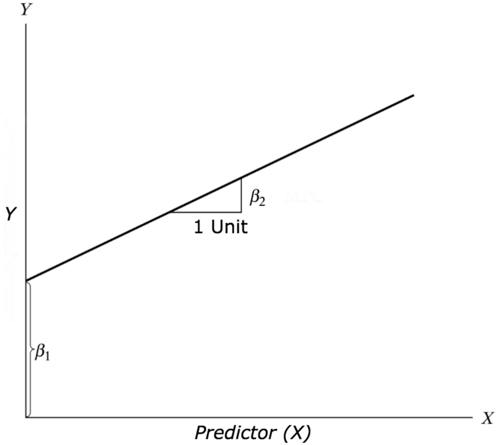
Objective

- The aim of linear regression is to model a continuous variable Y as a mathematical function of one or more X variable(s), so that we can use this regression model to predict the Y when only the X is known.

  $Y = \beta 1 + \beta 2X + \epsilon$

- where, $\beta 1$ is the intercept and $\beta 2$ is the slope. Collectively, they are called regression coefficients. ϵ is the error term, the part of Y the regression model is unable to explain.

Linear model



Example

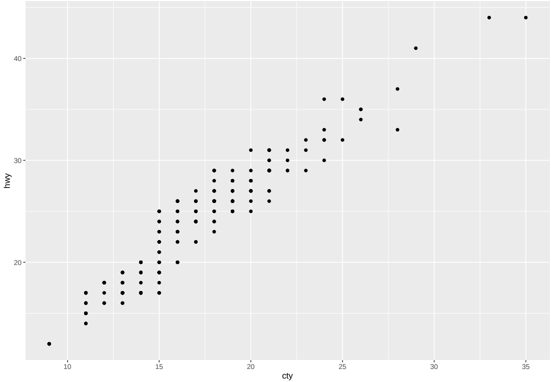
```r
sample_n(mpg, size=8)
```

```
## # A tibble: 8 x 11
##   manufacturer model  displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr>  <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 toyota       4runn~   4.7  2008     8 auto~ 4        14    17 r     suv
## 2 audi         a6 qu~   3.1  2008     6 auto~ 4        17    25 p     mids~
## 3 honda        civic    1.6  1999     4 manu~ f        28    33 r     subc~
## 4 audi         a4       2.8  1999     6 auto~ f        16    26 p     comp~
## 5 subaru       impre~   2.2  1999     4 manu~ 4        19    26 r     subc~
## 6 volkswagen   passat   1.8  1999     4 manu~ f        21    29 p     mids~
## 7 nissan       altima   2.5  2008     4 auto~ f        23    31 r     mids~
## 8 ford         explo~   4    1999     6 auto~ 4        14    17 r     suv
```

Data Visualization

- Scatter plot

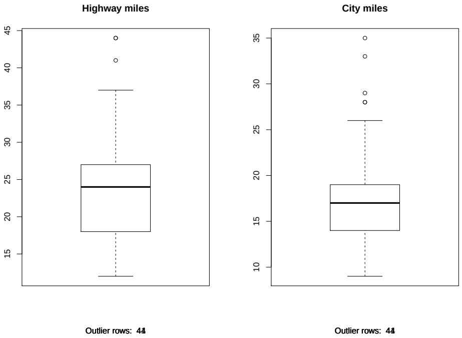
```
ggplot(mpg, aes(x=cty, y=hwy)) + geom_point()
```



Data Visualization

- BoxPlot / Check for outliers

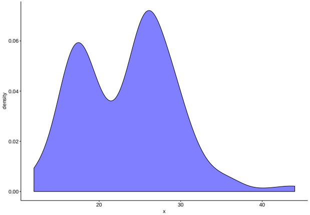
```
par(mfrow=c(1, 2))  # divide graph area in 2 columns
boxplot(mpg$hwy, main="Highway miles", sub=paste("Outlier rows: ",
     boxplot.stats(mpg$hwy)$out))  # box plot for 'speed'
boxplot(mpg$cty, main="City miles", sub=paste("Outlier rows: ",
  boxplot.stats(mpg$hwy)$out))  # box plot for 'distance'
```



Density plot

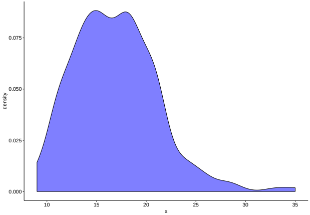
- Use package *ggpubr*
- Highway

```
ggdensity(mpg$hwy, fill="blue")
```



- City

```
ggdensity(mpg$cty, fill="blue")
```



Pearson test of normality

- Use package *nortest*

```r
pearson.test(mpg$hwy)
```

```
##
##  Pearson chi-square normality test
##
## data:  mpg$hwy
## P = 116.15, p-value < 2.2e-16
```

```r
pearson.test(mpg$cty)
```

```
##
##  Pearson chi-square normality test
##
## data:  mpg$cty
## P = 123.23, p-value < 2.2e-16
```

Data transformation

```r
# library MASS and bestNormalize
Hmil <- bestNormalize(mpg$hwy)
```

```r
Cmil <- bestNormalize(mpg$cty)
```

- Run **Hmil** and **Cmil** to investigate whether the transformation help normalizing mpg data.

Modeling linear regression

```r
#package lme4 and lmerTest
model <- lmer(hwy ~ cty + (1|year), data=mpg)
```

```r
summary(model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: hwy ~ cty + (1 | year)
##    Data: mpg
##
## REML criterion at convergence: 930.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.98036 -0.63537 -0.06445  0.68844  2.42073
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 0.07494  0.2738
##  Residual             3.03278  1.7415
## Number of obs: 234, groups:  year, 2
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   0.86738    0.50487  23.76808   1.718   0.0988 .
## cty           1.33892    0.02682 231.15811  49.921   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

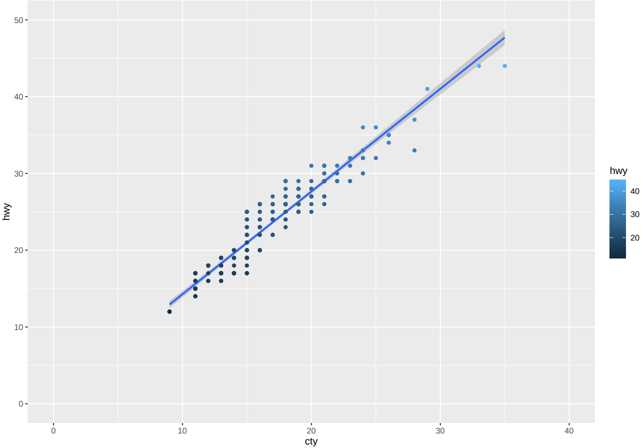
ANOVA

```r
anova(model)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)    
## cty   7558    7558     1 231.16  2492.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure

```
ggplot(mpg, aes(x=cty, y=hwy)) + geom_smooth(method="lm") +
  ylim(0, 50) + xlim(0,40) + geom_point(aes(color=hwy))
```



ANCOVA

- The analysis of covariance (ANCOVA) is used to compare two or more regression lines by testing the effect of a categorical factor on a dependent variable (y-var) while controlling for the effect of a continuous co-variable (x-var)
- Dataset Iris

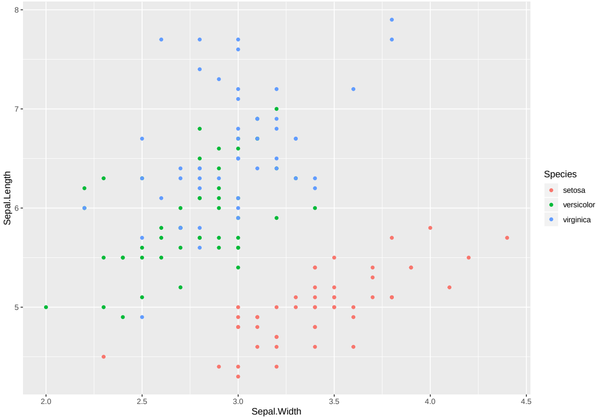
```
#Package tidyverse
sample_n(iris, size=5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1          4.9         3.1          1.5         0.2     setosa
## 2          6.8         2.8          4.8         1.4 versicolor
## 3          5.2         3.4          1.4         0.2     setosa
## 4          5.8         2.8          5.1         2.4  virginica
## 5          6.1         2.8          4.0         1.3 versicolor
```

Plot raw data

- I am interesting in the sepal length and sepal width relationship.

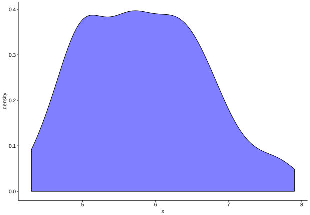
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, color=Species)
  geom_point()
```



Checking normality

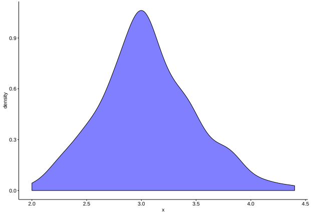
- Use package *ggpubr*
- Sepal length

```
ggdensity(iris$Sepal.Length, fill="blue")
```



- Sepal width

```
ggdensity(iris$Sepal.Width, fill="blue")
```



Modeling

- Note I use *lm* function because there is no random effects in my dataset.

```
data <- iris
Model <- lm(Sepal.Length~Sepal.Width * Species, data=data)
summary(Model)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width * Species, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26067 -0.25861 -0.03305  0.18929  1.44917
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.6390     0.5715   4.618 8.53e-06 ***
## Sepal.Width                     0.6905     0.1657   4.166 5.31e-05 ***
## Speciesversicolor               0.9007     0.7988   1.128    0.261
## Speciesvirginica                1.2678     0.8162   1.553    0.123
## Sepal.Width:Speciesversicolor   0.1746     0.2599   0.672    0.503
## Sepal.Width:Speciesvirginica    0.2110     0.2558   0.825    0.411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4397 on 144 degrees of freedom
## Multiple R-squared:  0.7274, Adjusted R-squared:  0.718
## F-statistic: 76.87 on 5 and 144 DF,  p-value: < 2.2e-16
```

ANOVA

```r
anova(Model)
```

```
## Analysis of Variance Table
##
## Response: Sepal.Length
##                      Df Sum Sq Mean Sq  F value    Pr(>F)
## Sepal.Width           1  1.412   1.412   7.3030  0.007712 **
## Species               2 72.752  36.376 188.1091 < 2.2e-16 ***
## Sepal.Width:Species   2  0.157   0.079   0.4064  0.666777
## Residuals           144 27.846   0.193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- What is ANOVA telling us?

Extracting slopes

- You may want to compare slopes

```
m.lst <- emtrends(Model, "Species", var="Sepal.Width")
m.lst # list the slope values
```

```
##  Species    Sepal.Width.trend    SE  df lower.CL upper.CL
##  setosa                 0.690 0.166 144    0.363     1.02
##  versicolor             0.865 0.200 144    0.469     1.26
##  virginica              0.902 0.195 144    0.517     1.29
##
## Confidence level used: 0.95
```

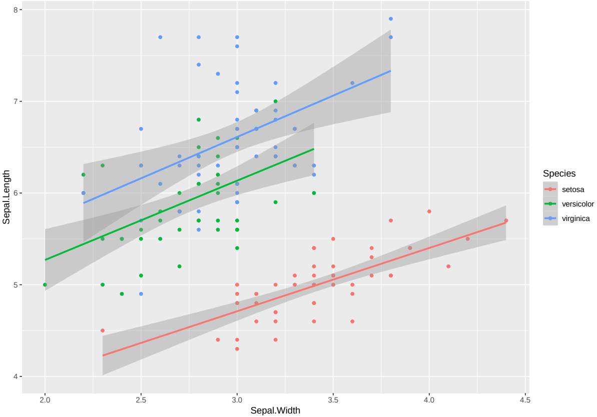
Comparing slopes

```r
pairs(m.lst)
```

```
##  contrast                 estimate    SE  df t.ratio p.value
##  setosa - versicolor       -0.1746 0.260 144 -0.672  0.7803
##  setosa - virginica        -0.2110 0.256 144 -0.825  0.6880
##  versicolor - virginica    -0.0365 0.279 144 -0.131  0.9907
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Plotting Model

```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length,
                 color=Species)) +
   geom_point() + geom_smooth(method="lm")
```



Fitting a more parsimonious model

- What is the difference between the Model (previous model) and Model2?

```
Model2 <- lm(Sepal.Length ~ Sepal.Width + Species, data=iris)
summary(Model2)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width + Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30711 -0.25713 -0.05325  0.19542  1.41253
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.2514     0.3698   6.089 9.57e-09 ***
## Sepal.Width         0.8036     0.1063   7.557 4.19e-12 ***
## Speciesversicolor   1.4587     0.1121  13.012  < 2e-16 ***
## Speciesvirginica    1.9468     0.1000  19.465  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

ANOVA

```r
anova(Model2)
```

```
## Analysis of Variance Table
##
## Response: Sepal.Length
##              Df Sum Sq Mean Sq  F value  Pr(>F)
## Sepal.Width   1  1.412   1.412   7.3628 0.00746 **
## Species       2 72.752  36.376 189.6512 < 2e-16 ***
## Residuals   146 28.004   0.192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA Test

- Comparison between Model and Model2

```
anova(Model, Model2, test="F")
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Sepal.Width * Species
## Model 2: Sepal.Length ~ Sepal.Width + Species
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    144 27.846
## 2    146 28.004 -2  -0.15719 0.4064 0.6668
```

- The *anova()* command clearly shows that removing the interaction does not affect the fit of the model (F=0.4064, *P*-value=0.6668)

Extracting slopes

```r
m.lst <- emtrends(Model2, "Species", var="Sepal.Width")
m.lst
```

```
##  Species    Sepal.Width.trend    SE  df lower.CL upper.CL
##  setosa                 0.804 0.106 146    0.593     1.01
##  versicolor             0.804 0.106 146    0.593     1.01
##  virginica              0.804 0.106 146    0.593     1.01
##
## Confidence level used: 0.95
```

Fitting a more parsimonious model

- What is the difference between the Model2 (previous model) and Model3?

```r
Model3 <- lm(Sepal.Length~Sepal.Width, data=iris)
summary(Model3)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5561 -0.6333 -0.1120  0.5579  2.2226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5262     0.4789   13.63   <2e-16 ***
## Sepal.Width  -0.2234     0.1551   -1.44    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8251 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

ANOVA test

```r
anova(Model2, Model3)
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Sepal.Width + Species
## Model 2: Sepal.Length ~ Sepal.Width
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    146  28.004
## 2    148 100.756 -2   -72.752 189.65 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

- The *anova()* command clearly shows that removing species strongly affect the fit of the model (F=189.65 , *P*-value=< 2.2e-16)

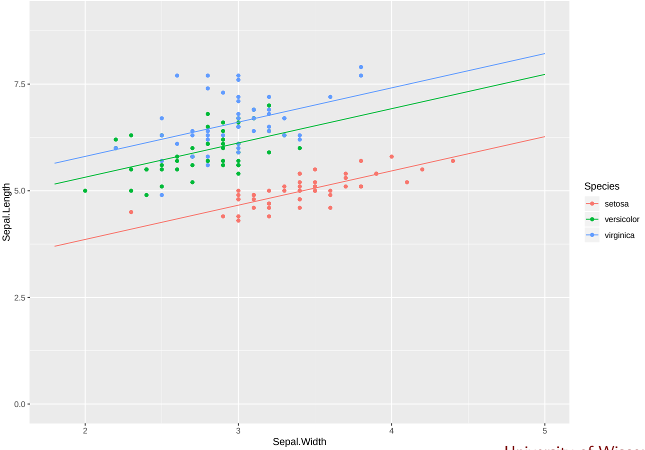
Fit the Model2

- Creating new data frames to fit Model 2 in a ggplot2 figure.

```r
newdata <- expand.grid(Sepal.Width=seq(1.8, 5, length=5))
newdata1 <- data.frame(Species =c("versicolor"), newdata)
newdata2 <- data.frame(Species =c("setosa"), newdata)
newdata3 <- data.frame(Species =c("virginica"), newdata)

nd=rbind(newdata1, newdata2, newdata3)

pm <- predict(Model2, newdata=nd, interval="confidence")

nd$p <- pm[,1]
nd$pmin <- pm[,2] # conf interval
nd$pmax <- pm[,3] # conf interval
```

Figure

```
ggplot(data = iris, aes(x = Sepal.Width, y = Sepal.Length, col
  geom_point() +
  geom_line(data=nd, aes(x=Sepal.Width, y=p)) + ylim(0,9)
```


Sepal.Length
Sepal.Width
Species
setosa
versicolor
virginica