CS 660: Mathematical Foundations for Analytics

Dr. Francis Parisi

Pace University

Spring 2017

Course Overview

1. Part I: Data Science Fundamentals
   - Data Science Concepts and Process
   - The R Language
   - Exploratory Data Analysis
   - Cleaning & Manipulating Data
   - Presenting Results
2. Part II: Graphs & Statistical Methods
   - Basic Graphics
   - Advanced Graphics
   - Probability & Statistical Methods
3. Part III: Modeling Methods
   - Model Selection and Evaluation
   - Linear and Logistic Regression
   - Unsupervised Methods
   - Advanced Modeling Methods

Linear Regression Models

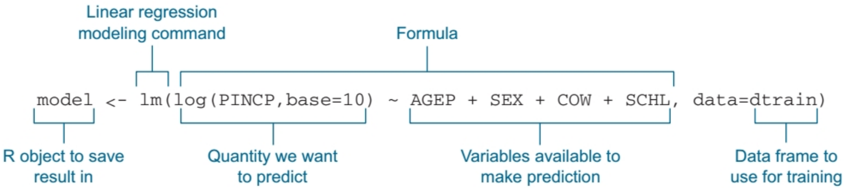
- Recall that the linear regression model has the form

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_p X_{pi} \qquad i = 1, 2 \ldots n$$

- A model with only one independent variable is a *simple* linear regression model; with more than one it's *multiple* regression
- This is the ordinary least squares (OLS) model and we find the $\hat{\beta}'s$ by minimizing the sum of squared residuals (or errors)

$$\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n} (Y_i - \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_p X_{pi})^2 = \sum_{i=1}^{n} \varepsilon_i^2$$

- Linear regression is the cornerstone prediction method
- Models the expected value of a dependent or *response* variable given a set of independent of *explanatory* variables
- Fitting a regression model in R



Linear Regression Models

Symbol	Usage
`~`	Separates response variables on the left from the explanatory variables on the right. For example, a prediction of `y` from `x`, `z`, and `w` would be coded `y ~ x + z + w`.
`+`	Separates predictor variables.
`:`	Denotes an interaction between predictor variables. A prediction of `y` from `x`, `z`, and the interaction between `x` and `z` would be coded `y ~ x + z + x:z`.
`*`	A shortcut for denoting all possible interactions. The code `y ~ x * z * w` expands to `y ~ x + z + w + x:z + x:w + z:w + x:z:w`.
`^`	Denotes interactions up to a specified degree. The code `y ~ (x + z + w)^2` expands to `y ~ x + z + w + x:z + x:w + z:w`.
`.`	A placeholder for all other variables in the data frame except the dependent variable. For example, if a data frame contained the variables `x`, `y`, `z`, and `w`, then the code `y ~ .` would expand to `y ~ x + z + w`.
`-`	A minus sign removes a variable from the equation. For example, `y ~ (x + z + w)^2 - x:w` expands to `y ~ x + z + w + x:z + z:w`.
`-1`	Suppresses the intercept. For example, the formula `y ~ x -1` fits a regression of `y` on `x`, and forces the line through the origin at `x=0`.
`I()`	Elements within the parentheses are interpreted arithmetically. For example, `y ~ x + (z + w)^2` would expand to `y ~ x + z + w + z:w`. In contrast, the code `y ~ x + I((z + w)^2)` would expand to `y ~ x + h`, where `h` is a new variable created by squaring the sum of `z` and `w`.
function	Mathematical functions can be used in formulas. For example, `log(y) ~ x + z + w` would predict `log(y)` from `x`, `z`, and `w`.

Linear Regression Models

Function	Action
`summary()`	Displays detailed results for the fitted model
`coefficients()`	Lists the model parameters (intercept and slopes) for the fitted model
`confint()`	Provides confidence intervals for the model parameters (95% by default)
`fitted()`	Lists the predicted values in a fitted model
`residuals()`	Lists the residual values in a fitted model
`anova()`	Generates an ANOVA table for a fitted model, or an ANOVA table comparing two or more fitted models
`vcov()`	Lists the covariance matrix for model parameters
`AIC()`	Prints Akaike's Information Criterion
`plot()`	Generates diagnostic plots for evaluating the fit of a model
`predict()`	Uses a fitted model to predict response values for a new dataset

Linear Regression Models

1. Using the database `women` in the base R installation fit a regression of `weight` on `height` (don't forget to save your model)
2. Display a summary of your model, and interpret the results
3. Display the fitted values
4. Display the residuals
5. Plot `weight` (y-axis) and `height` (x-axis)

Linear Regression Models

A polynomial regression is a regression model where we use use powers of the explanatory variables

Refit the the regression as a quadratic model and interpret the results as before

```
fit2 <- lm(weight ~ height + I(height^2), data=women)
```

Note this is still a linear model even though we have a quadratic term – it's a linear combination of the $\hat{\beta}'s$

An example of a nonlinear model is

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 e^{X/\hat{\beta}_2}$$

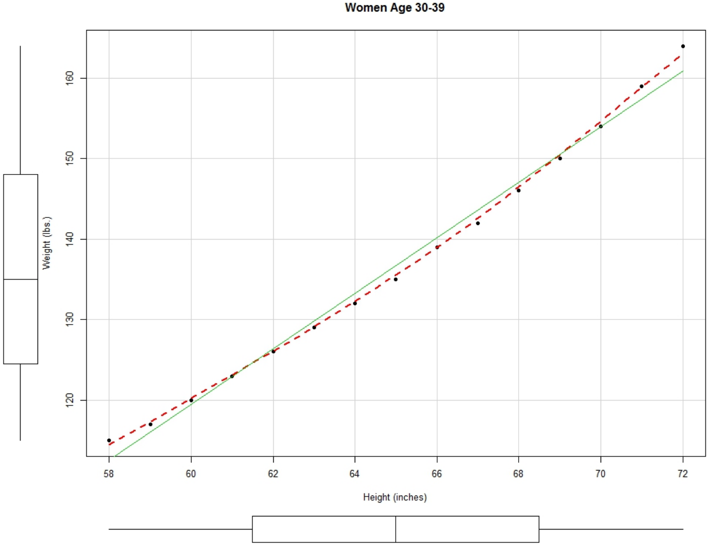
You can fit nonlinear models with `nls()`

Linear Regression Models

The `car` library has a `scatterplot()` function that can create an informative plot

```
library(car)
scatterplot(weight ~ height, data=women,
  spread=FALSE, smoother.args=list(lty=2), pch=19,
  main="Women Age 30-39",
      xlab="Height (inches)",
  ylab="Weight (lbs.)")
```

Linear Regression Models



Linear Regression Models

- What we have been discussing is *simple linear regression*
- When there are more than one independent variable we have *multiple linear regression*
- When working with several independent variables we need to check for correlation among them
- This gives insight into possible interactions and multicollinearity
- We'll us the built-in data set `state.x77`

- Since `state.x77` is a matrix we need to convert it to a data.frame
  ```
  states <- as.data.frame(state.x77[,c("Murder",
  "Population", "Illiteracy", "Income", "Frost")])
  ```
- Use `cor()` to get a correlation matrix
  ```
  cor(states)
  ```

```
           Murder Population Illiteracy Income Frost
Murder       1.00       0.34       0.70  -0.23 -0.54
Population   0.34       1.00       0.11   0.21 -0.33
Illiteracy   0.70       0.11       1.00  -0.44 -0.67
Income      -0.23       0.21      -0.44   1.00  0.23
Frost       -0.54      -0.33      -0.67   0.23  1.00
```

Linear Regression Models

- `cor()` produces a correlation matrix but doesn't tell us anything about the statistical significance of the estimates
- `cor.test()` calculates the correlation and tests for significance but only works on a pair at a time
- Years ago I created a function called `Mat.cor.test` that does what `cor.test` does and works with a matrix

Linear Regression Models

```
##  matrix cor.test
Mat.cor.test <- function(obj, alt = "two.sided", meth = "pearson")
{
  n <- dim(obj)[2]
  est <- matrix(0, nrow = n, ncol = n,
                   dimnames = list(names(obj), names(obj)))
  pval <- matrix(0, nrow = n, ncol = n,
                   dimnames = list(names(obj), names(obj)))
  for(i in 1:(n - 1)) {
    for(j in (i + 1):n) {
      temp.cor <- cor.test(obj[, i], obj[, j], alternative = alt,
                           method = meth)
      est[i, j] <- temp.cor$estimate
      pval[i, j] <- temp.cor$p.value
      est[j, i] <- temp.cor$estimate
      pval[j, i] <- temp.cor$p.value
    }
  }
  diag(est) = 1
  list(estimates = round(est, 2), p.values = zapsmall(round(pval, 2)))
}
```

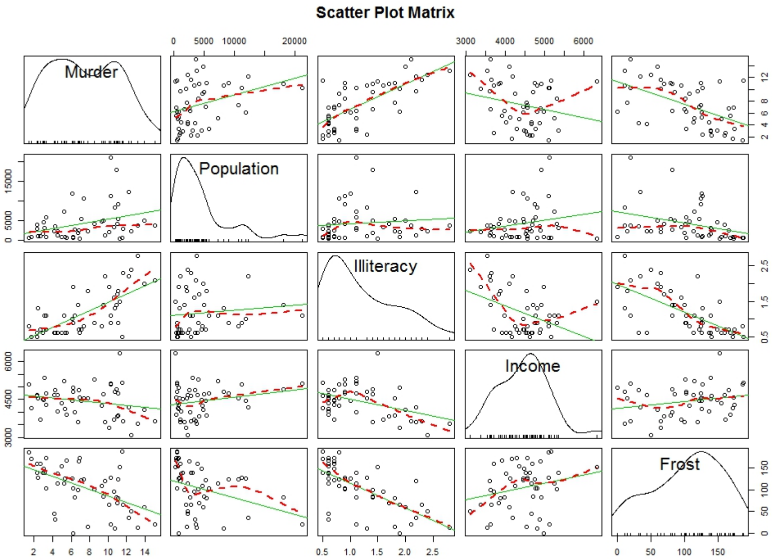
```
Mat.cor.test(states)
$estimates
            Murder Population Illiteracy Income Frost
Murder        1.00       0.34       0.70  -0.23 -0.54
Population    0.34       1.00       0.11   0.21 -0.33
Illiteracy    0.70       0.11       1.00  -0.44 -0.67
Income       -0.23       0.21      -0.44   1.00  0.23
Frost        -0.54      -0.33      -0.67   0.23  1.00

$p.values
            Murder Population Illiteracy Income Frost
Murder        0.00       0.01       0.00   0.11  0.00
Population    0.01       0.00       0.46   0.15  0.02
Illiteracy    0.00       0.46       0.00   0.00  0.00
Income        0.11       0.15       0.00   0.00  0.11
Frost         0.00       0.02       0.00   0.11  0.00
```

- As you recall from our discussions of EDA, it is important to visualize the data
- Let's turn to the `car` library again

```
library(car)
scatterplotMatrix(states, spread=FALSE,
                    smoother.args=list(lty=2),
                     main="Scatter Plot Matrix")
```

Linear Regression Models



Linear Regression Models

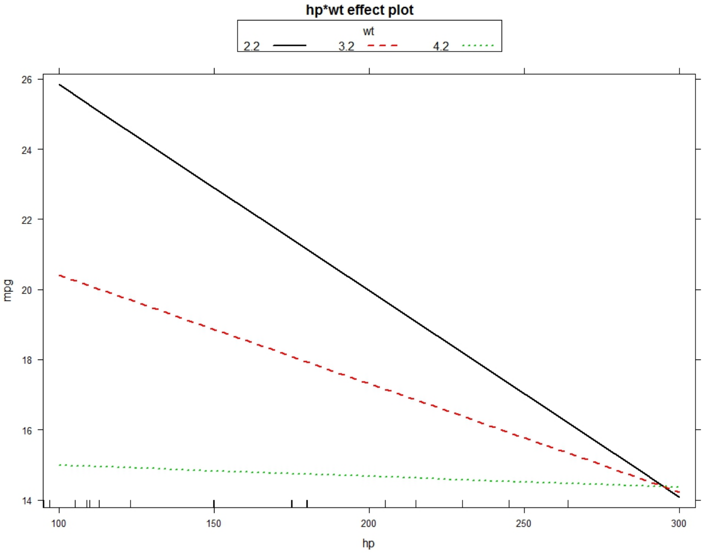
- Fit a multiple regression using the `states` data frame, with `Murder` as the dependent variable...
- `fit <- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)`
- `summary(fit)`
- `anova(fit)`
- This model shows use the main effects of each of the variables on the murder rate
- We should look at the interaction between variables when we fit regression models

- Let's look at another model

```
fit <- lm(mpg   hp + wt + hp:wt, data=mtcars)
```

- When we fit a model with interactions we can visualize the interaction terms to learn more about the relationship
- We can use the effects package to help out

```
library(effects)
plot(effect("hp:wt", fit,,
        list(wt=c(2.2,3.2,4.2))),
        multiline=TRUE)
```


hp*wt effect plot
wt
2.2
3.2
4.2
26
24
22
20
18
16
14
mpg
100
150
200
250
300
hp

Linear Regression Models

After fitting a regression model it's important to look at diagnostics to check the assumptions

1. *Normality*: the residuals should be normally distributed with a mean of 0 and variance σ^2
2. *Independence*: the dependent variable values are independent
3. *Linearity*: the dependent variable is linearly related to the independent variables
4. *Homoscedasticity*: the variance of the residuals is constant and does not vary with the dependent variable

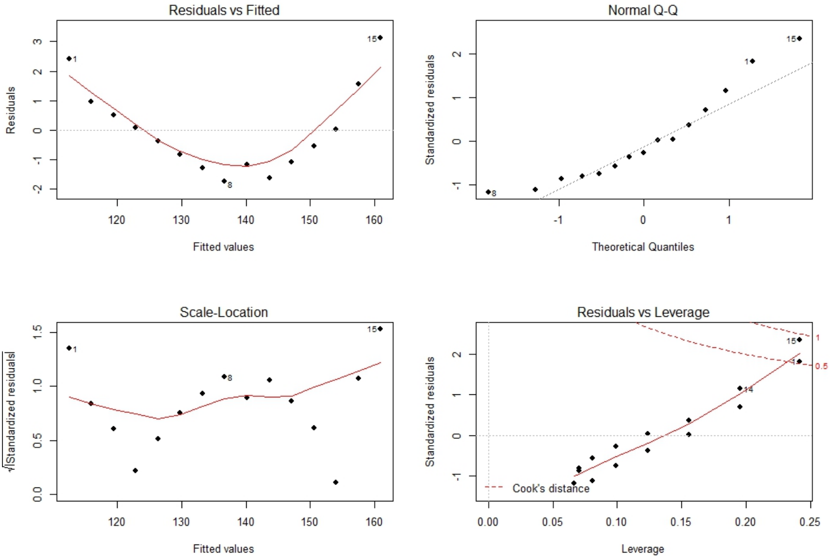
We also look for outliers, high-leverage observations, and influential observations (*Cook's distance*)

Linear Regression Models

Let's return to our earlier model of regessing weight on height

```
fit <- lm(weight ~ height, data=women)
par(mfrow=c(2,2))
plot(fit)
par(mfrow=c(1,1)) ## return parameters to normal
```

Linear Regression Models



Linear Regression Models

Look again at the `states` model

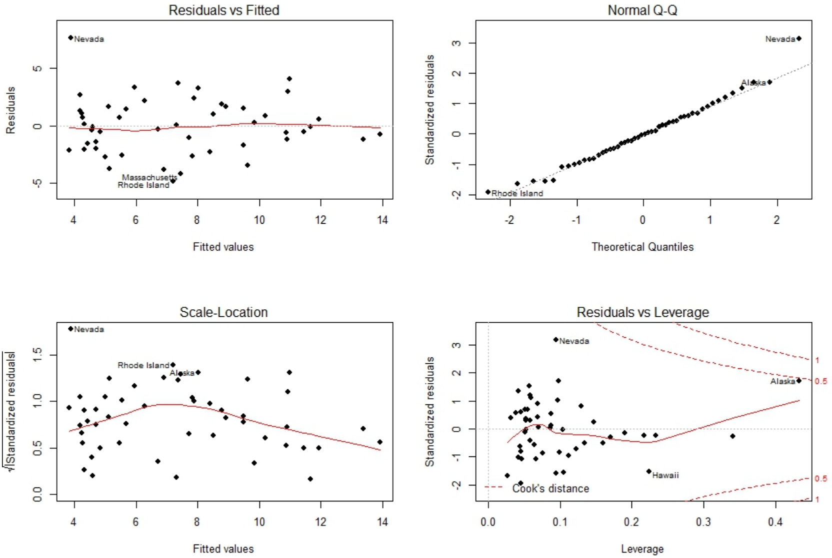
```
fit <- lm(Murder   Population + Illiteracy + Income
+ Frost, data=states)

par(mfrow=c(2,2))

plot(fit)

par(mfrow=c(1,1))
```

Linear Regression Models



Linear Regression Models

The `car` package provides several additional functions that enhance model evaluation

Function	Purpose
`qqPlot()`	Quantile comparisons plot
`durbinWatsonTest()`	Durbin–Watson test for autocorrelated errors
`crPlots()`	Component plus residual plots
`ncvTest()`	Score test for nonconstant error variance
`spreadLevelPlot()`	Spread-level plots
`outlierTest()`	Bonferroni outlier test
`avPlots()`	Added variable plots
`influencePlot()`	Regression influence plots
`scatterplot()`	Enhanced scatter plots
`scatterplotMatrix()`	Enhanced scatter plot matrixes
`vif()`	Variance inflation factors

Linear Regression Models

In-class practice

1. Fit a multiple regression model using the `Auto` data used before
2. If you need to download the data again...
   `AutoLink <-`
   `"http://www-bcf.usc.edu/ gareth/ISL/Auto.csv"`
3. Make `origin` a factor and be sure `horsepower` is numeric
4. Regress `mpg` on `cylinders, displacement, horsepower, weight, acceleration` and `origin`
5. Change your plot to produce four graphs, two by two
6. Revise the model based on the output from summary()
7. Interpret your final model

Linear Regression Models

- Multicollinearity is when two or more independent variables are highly correlated
- Results in large confidence intervals for the parameter estimates (uncertainty)
- Makes interpretation of of the parameters difficult
- May show variables are not significant when they are
- May switch the sign of some parameters
- Variance Inflation Factor (VIF) is a measure to detect multicollinearity – `vif()` in the `car` library

Linear Regression Models

The math behind the R output

- Standard error of the estimates is the square root of the variance

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\overline{X}^2}{\sum_{i=1}^{n} (X_i - \overline{X})^2} \right]$$

  where σ^2 is the variance of the residuals

$$\hat{\sigma}^2 = \frac{RSS}{n - p - 1}$$

  recall

$$RSS = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \sum \varepsilon^2$$

Linear Regression Models

$$SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(X_i - \overline{X})^2}$$

when there's only one independent variable, and in general

$$\mathbf{SE}(\beta)^{\mathbf{2}} = \sigma^2(\mathbf{X'X})^{-1}$$

or

$$\mathbf{SE}(\beta) = \sqrt{\sigma^2(\mathbf{X'X})^{-1}}$$

where $\mathbf{X}$ is the model matrix

Linear Regression Models

- For regression models $H_0 : \beta_i = 0$
- The t-statistic is

$$t = \frac{\hat{\beta}_i - \beta_i}{SE(\beta_i)}$$

or

$$t = \frac{\hat{\beta}_i}{SE(\beta_i)}$$

Linear Regression Models

- Total sum of squares

$$SSTO = \sum (Y_i - \overline{Y})^2$$

- Regression sum of squares

$$SSR = \sum (\hat{Y}_i - \overline{Y})^2$$

- Error sum of squares

$$SSE = \sum (Y_i - \hat{Y})^2 = RSS$$

- The residual standard error is

$$RSE = \sqrt{\frac{SSE}{n - p - 1}}$$

Linear Regression Models

- Regression Mean Square

$$MSR = \frac{SSR}{p}$$

where p is the number of independent variables in the regression model

- Error Mean Square

$$MSE = \frac{SSE}{n - p - 1}$$

- F-statistic

$$F = \frac{MSR}{MSE}$$

Linear Regression Models

- R^2 is the proportion of variance explained by the model
- R^2 always takes a value between 0 and 1
- R^2 is independent of the scale of the dependent variable Y

$$R^2 = \frac{SSTO - SSE}{SSTO} = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO}$$

and adjusted R^2

$$R^2_{adj} = 1 - \left(\frac{n-1}{n-p-1}\right)(1 - R^2)$$

Linear Regression Models

After fitting a regression model and interpreting the output, it may be necessary to make revisions

- Delete observations
- Transform variables
- Add or delete variables
- Try another regression modeling approach

Finding the “best” model

- Comparing models using `anova` – based on the F for a reduced model vs. a full model

```
> anova(fit.02b, fit.02)
Analysis of Variance Table

Model 1: Murder ~ Population + Illiteracy
Model 2: Murder ~ Population + Illiteracy + Income + Frost
  Res.Df   RSS Df Sum of Sq     F Pr(>F)
1     47 289.2
2     45 289.2  2   0.07851 0.006  0.994
```

- and with `AIC`

```
> AIC(fit.02b, fit.02)
        df     AIC
fit.02b  4 237.657
fit.02   6 241.643
```

Linear Regression Models

- Variable selection using step-wise regression, and all subsets regression
- Assessing how well a model works in the real world
  - Cross-validation
    - ★ A portion of the data is selected as the training sample, and a portion is selected as the hold-out sample
    - ★ A regression equation is developed on the training sample and then applied to the hold-out sample
    - ★ The performance on this sample is a more accurate estimate of the operating characteristics of the model with new data
  - Relative Importance
    - ★ Which variables are most important in predicting the outcome?
    - ★ Standardize your data before fitting the regression
    - ★ The coefficients measure how many standard deviations your dependent variable changes with a 1 standard deviation of your independent variable
    - ★ Puts the effects of each variable on the same footing, and comparable

References

James, G., Hastie, T., Witten, D. and Tibshirani, R. (2013).
An Introduction to Statistical Learning with Applications in R.
Springer, second edition.
6th Printing 2015.

Kabacoff, R. I. (2015).
R in Action.
Manning, Shelter Island, NY, second edition.

Lander, J. P. (2014).
R for Everyone.
Addison-Wesley, Upper Saddle River.

Zumel, N. and Mount, J. (2014).
Practical Data Science with R.
Manning, Shelter Island, NY, second edition.