UCLA Department of Statistics
Statistical Consulting Center

Basic Data Investigation

Tiffany Himmel (Head)

September 20, 2009

Outline

To follow along with the R commands, download this file:
www.stat.ucla.edu/~tiffany/bootcamp/IrisExampleCode.R

The Data Frame

```
> class(iris)
[1] "data.frame"
> names(iris)
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
[5] "Species"
> attach(iris)
```

- A data frame is the most common way to store data.
- `names` gives you the columns in the frame.
- `attach` makes the data frame columns available as vectors
- Now `species` and `iris$species`

Exploring the Data

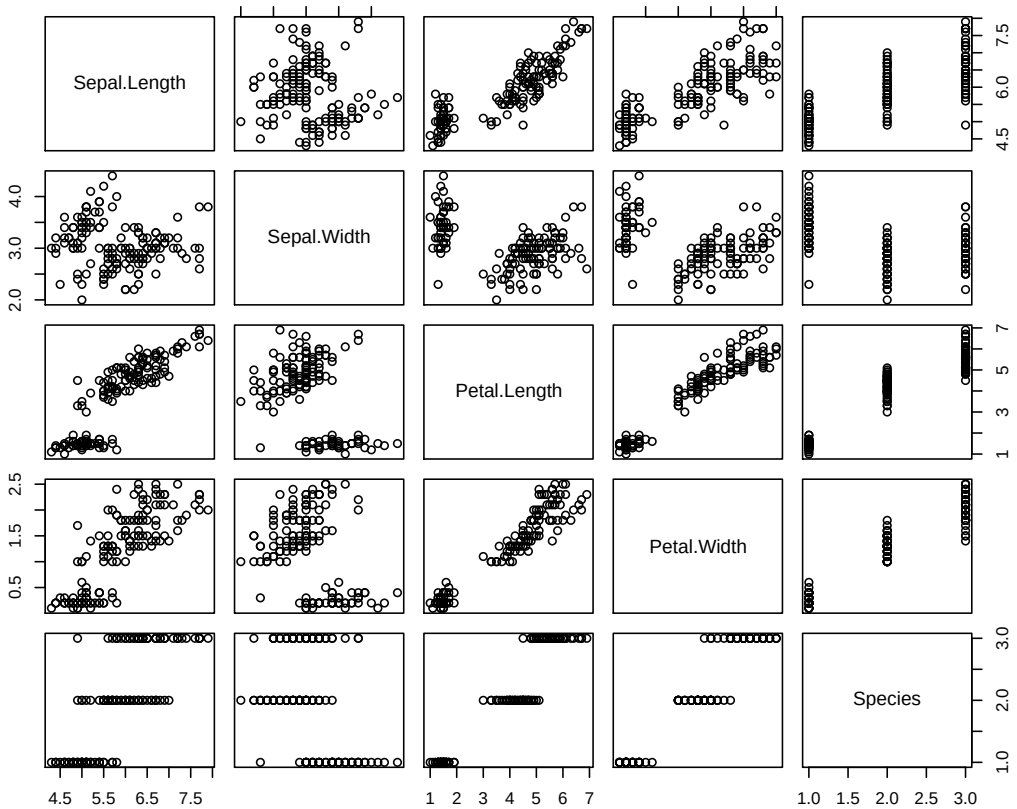
```
> summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

- This set of basic summary statistics can be very helpful.
- `species` is a factor so `summary` gives a table for it instead.

Exploring the Data

```
> plot(iris)
```



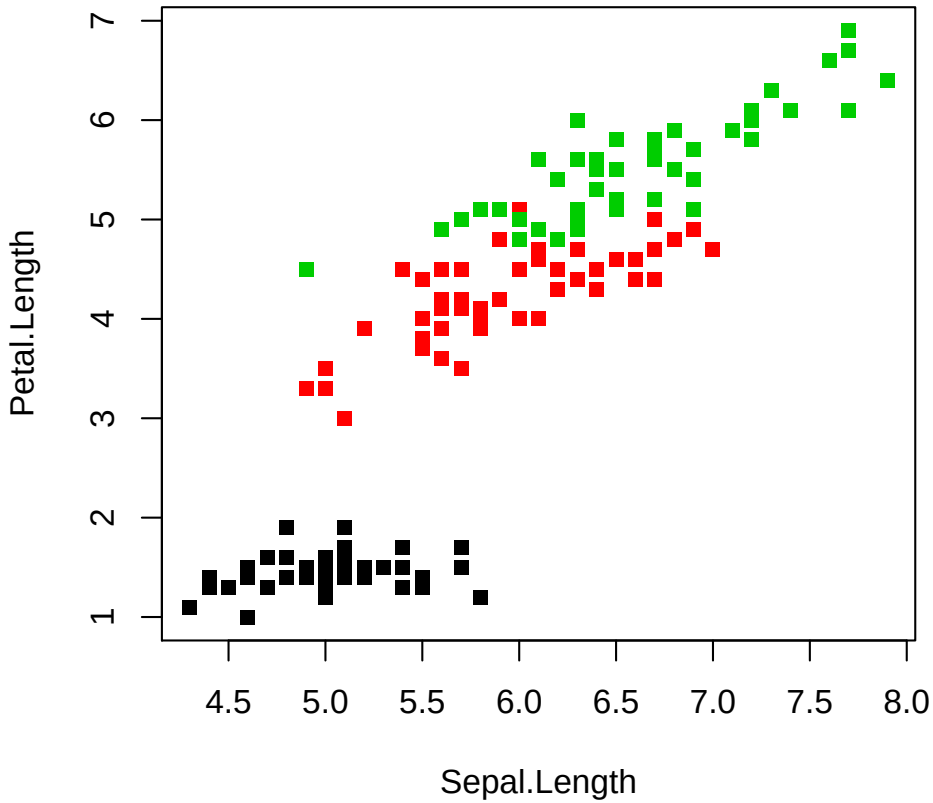
Exploring the Data

```
> plot(iris)
```

- Shows the interactions.
- It is a good exploratory plot.
- What do you see in these plots?

Exploring the Data

```
> plot(iris[c(1,3)],col=as.numeric(Species))
> class(Species)
[1] "factor"
```



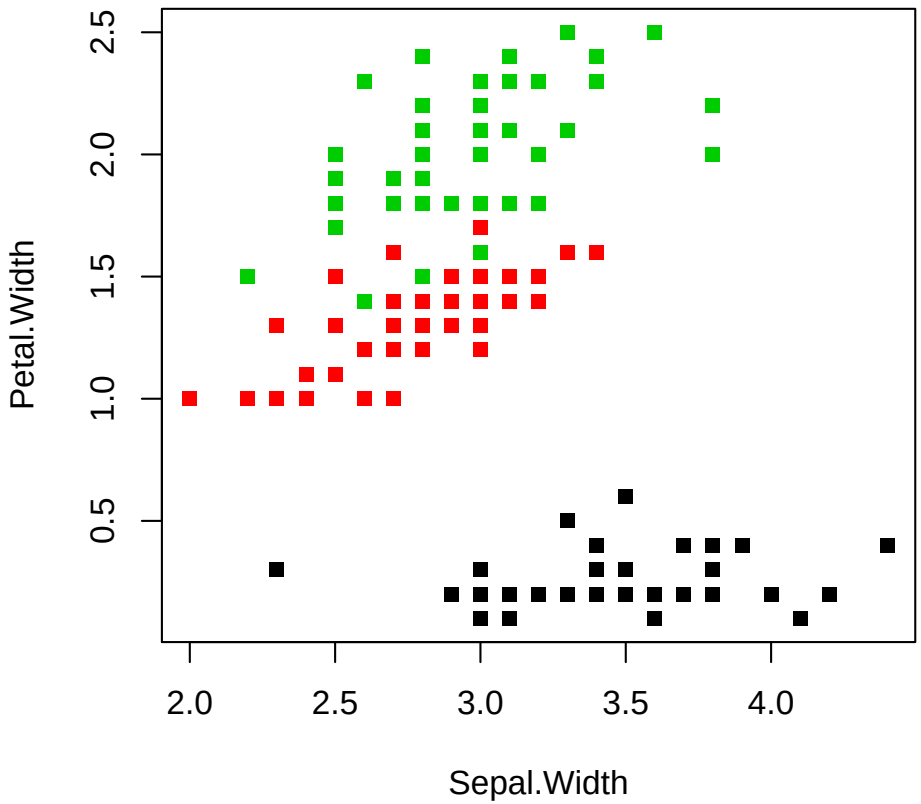
Data Subsets

```
> plot(iris[c(1,3)],col=as.numeric(Species))
> class(Species)
[1] "factor"
```

- `subset[c(1,3)]` only plots 1^{st} and 3^{rd} variable
- `as.numeric` turns factor into numbers (1, 2, 3)
- `col` option gives a color value to each data point
- We will use these two length variables in our walk through.

Data Subsets

```
> plot(iris[c(2,4)],col=as.numeric(Species))
```



- You will explore the width variables on your own later.

Data Subsets

```
> summary(Petal.Length[which(Species=="setosa")])

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.400   1.500   1.462   1.575   1.900
```

- `which` is one of the most useful functions in R.
- `which` returns the locations in the vector for which the expression evaluates to `TRUE`.

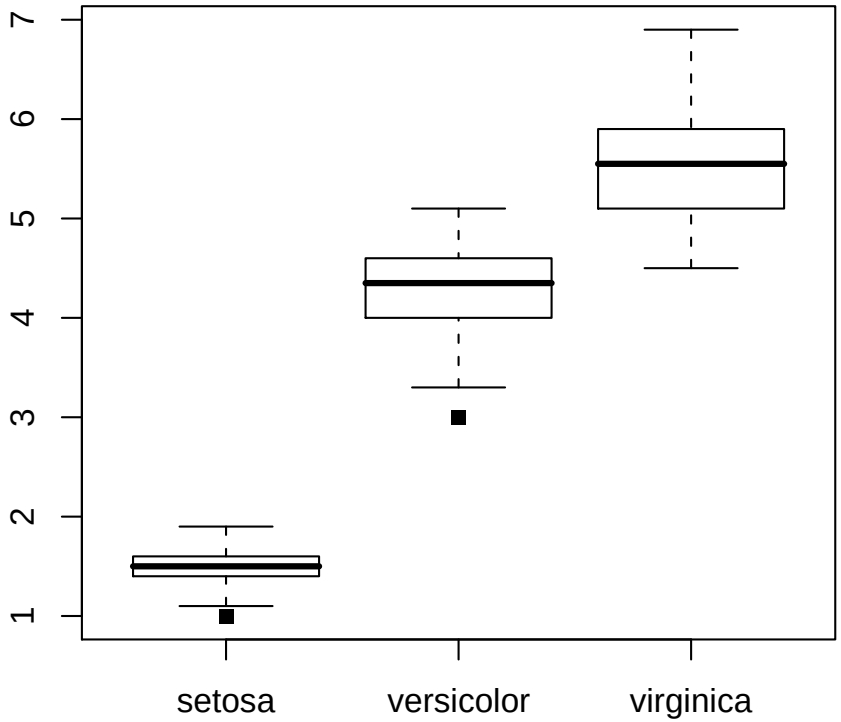
Data Subsets

```
> Petal.Length.V<-Petal.Length[-which(Species=="setosa")]
> Sepal.Length.V<-Sepal.Length[-which(Species=="setosa")]
```

- Now let's look at the non-`"setosa"` values.
- The minus `-` in `subset` removes those entries.

The Linear Model

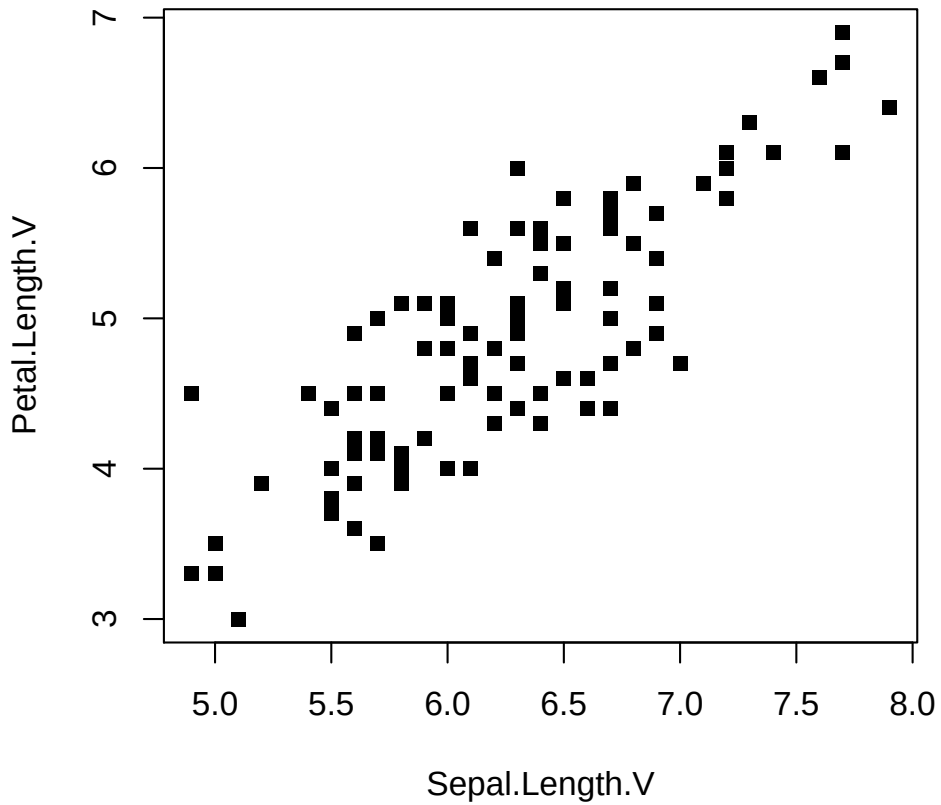
```
> boxplot(boxplot(Petal.Length~Species))
```



- `y~x` is called a "formula" in R.

The Linear Model

```
> plot(Petal.Length.V~Sepal.Length.V)
```



- Plot has many capabilities:
- `plot(x, y)` or `plot(y~x)` or `plot(object)`

The Linear Model

```
# Using the subsets we build
> l<-lm(Petal.Length.V~Sepal.Length.V)
# Letting lm do it for us
> l<-lm(Petal.Length~Sepal.Length,subset=which(Species!="setosa"))
```

- `lm` stands for linear model.
- Using the formula `Petal.Length~Sepal.Length` fits:

$$PL_i = \text{Intercept} + SL_i \cdot \text{slope} + \epsilon_i$$

- We solve for the intercept and slope
- ϵ_i is assumed to be $\epsilon_i \sim N(0, 1)$

The Linear Model

```
> summary(l)

Call:
lm(formula = Petal.Length ~ Sepal.Length, subset = which(Species !=
    "setosa"))

Residuals:
     Min       1Q   Median       3Q      Max
-0.96754 -0.32448 -0.03883  0.32768  1.05479

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.55571    0.44366  -3.507 0.000687 ***
Sepal.Length  1.03189    0.07046  14.645  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.4647 on 98 degrees of freedom
Multiple R-squared: 0.6864, Adjusted R-squared: 0.6832
F-statistic: 214.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

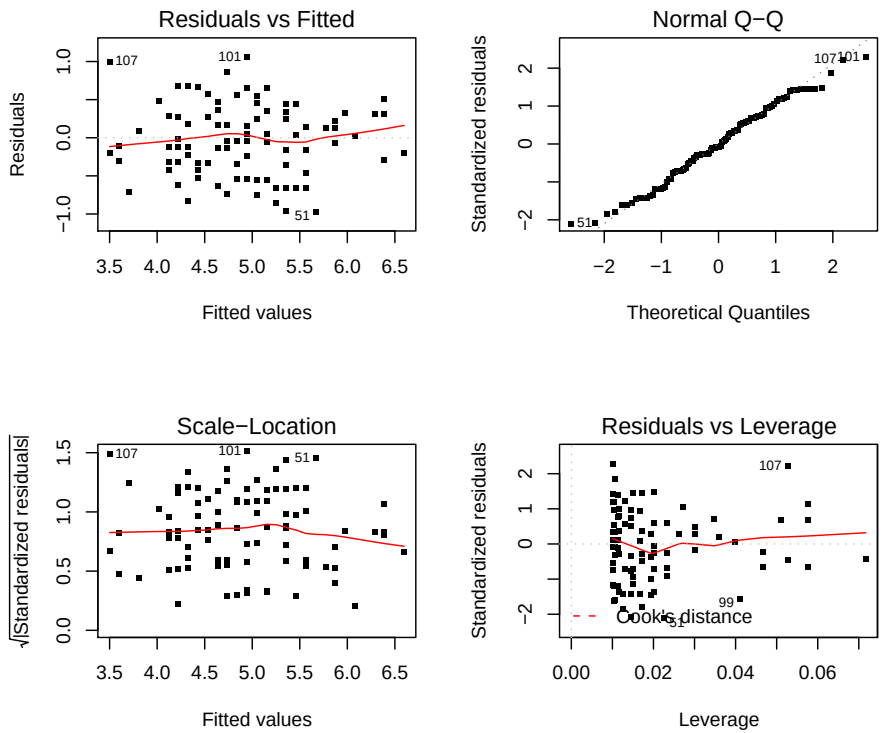
The Linear Model

```
> names(l)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
> names(summary(l))
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

- Almost all objects in R have names.
- `l$coefficients` will give a vector of intercept and slope.
- A very helpful way to get values out of a fit.

The Linear Model

```
> par(mfrow=c(2,2))
> plot(l))
> par(mfrow=c(1,1))
```



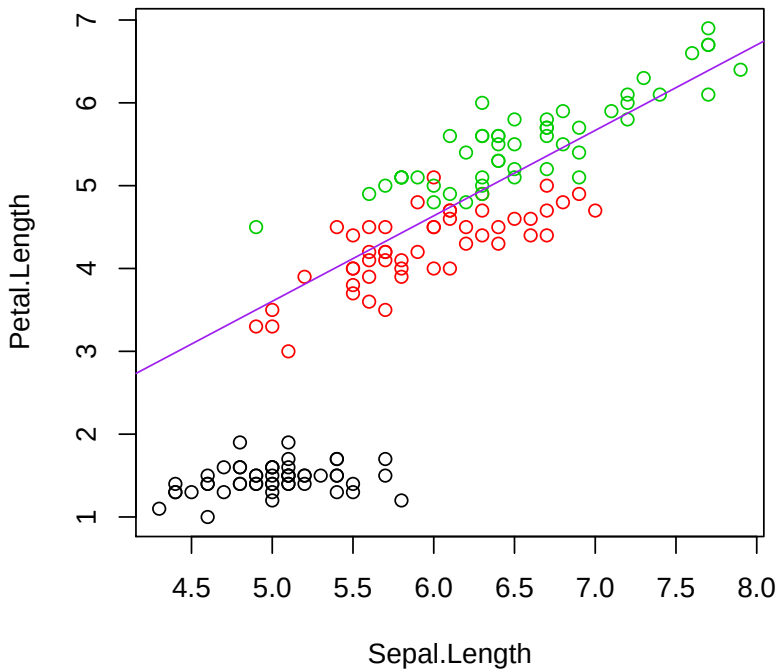
The Linear Model

```
> par(mfrow=c(2,2))
> plot(l))
> par(mfrow=c(1,1))
```

- These four plots are the default plots we use to examine linear models.
  - Residuals plot shows no pattern
  - Normal QQ plot is linear
- Use `par(mfrow=c(2,2))` to show all 4 plots at once instead of one at a time.
- Don't forget to go turn this off with `par(mfrow=c(1,1))`.

The Linear Model

```
> plot(Petal.Length~Sepal.Length,main="Iris Length Data",
col=as.numeric(Species))
> abline(coef(l),col="purple")
```



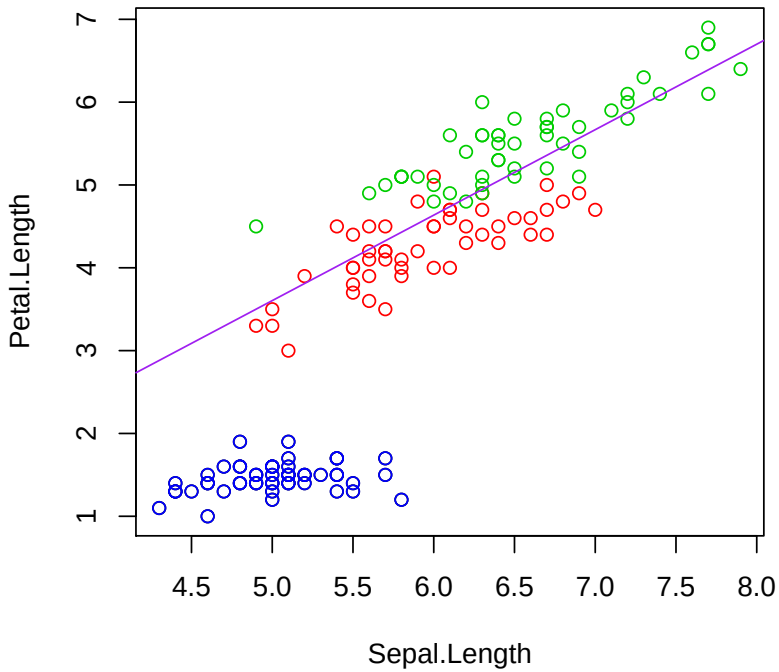
The Linear Model

```
> plot(Petal.Length~Sepal.Length,main="Iris Length Data",
col=as.numeric(Species))
> abline(coef(l),col="purple")
```

- `coef(l)` gives the intercept and slope calculated for `l`.
- `abiline` adds the fitted line to the current plot.

The Linear Model

```
> setosa.rows<-which(Species=="setosa")
> points(Sepal.Length[setosa.rows],Petal.Length[setosa.rows],col="blue")
```



The Linear Model

```
> setosa.rows<-which(Species=="setosa")
> points(Sepal.Length[setosa.rows],Petal.Length[setosa.rows],col="blue")
```

- We can also hold onto the `setosa` rows.
- `points` adds points on top of the current plot like `abiline` did for the fit line.

On Your Own

Things to try:

- Explore other variables.
- Compare linear models.
- Use subsetting any other methods you've learned today.

Want more colors/models/options?

- Try `help(lm)`, `help(plot)`
- Try googling: "R help *subject*"