Sample Analysis and Visualizations

Irfan Kanat

12/20/2015

Introducing the Dataset

In this document we will analyze the Motor Trends data. The dataset was compiled from 1974 issues of Motor Trends magazine and is included with R Base package.

Let us start with loading the dataset.

```
data(mtcars)
```

As we learned in the section on packages, you can querry the documentation for almost anything. Including the datasets included in packages. The document includes descriptions of the variables.

```
?mtcars
```

Let us get a sense of the data.

```
# A summary of variables
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
# Correlation table for first 4 variables (due to space concerns)
cor(mtcars[,1:4])
```

```
##             mpg        cyl       disp         hp
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475
## disp -0.8475514  0.9020329  1.0000000  0.7909486
## hp   -0.7761684  0.8324475  0.7909486  1.0000000
```

```
# bivariate comparisons of categorical variables
table(mtcars[,c("am","cyl")])
```

```
##    cyl
## am   4  6  8
##   0  3  4 12
##   1  8  3  2
```

```
# The histogram below should reflect these figures.
table(mtcars$gear)
```

```
## 
##  3  4  5 
## 15 12  5
```

Plotting with qplot()

R has long been known for its extensive visualization capabilities. The number of packages that handle visualizations are many, yet ggplot2 shines among them all. Today I will focus on ggplot2 and discuss plotting histograms and scatter plots with qplot. I will focus mostly on qplot() function, and discuss ggplot structure only briefly.

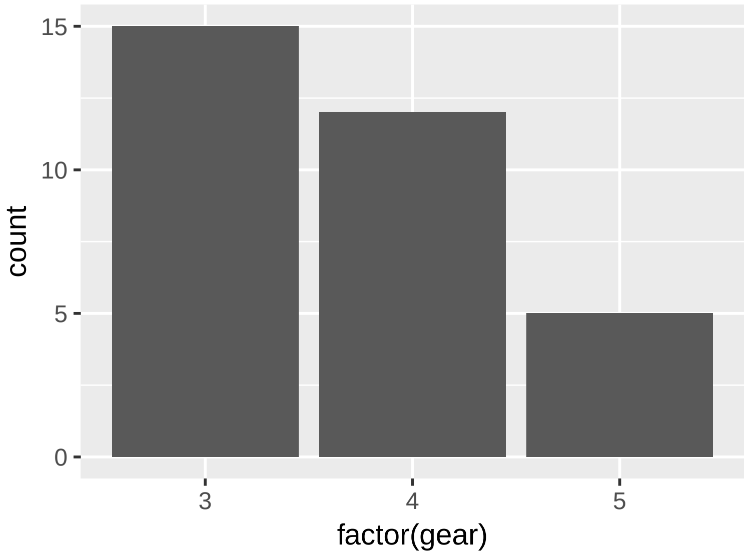
Now we can get to the fun part. qplot simplifies the ggplot functionality by automating most common tasks. We will use qplot for most common plots.

```
# Load the ggplot package
library(ggplot2)
# Review function syntax
?qplot
```

Histogram

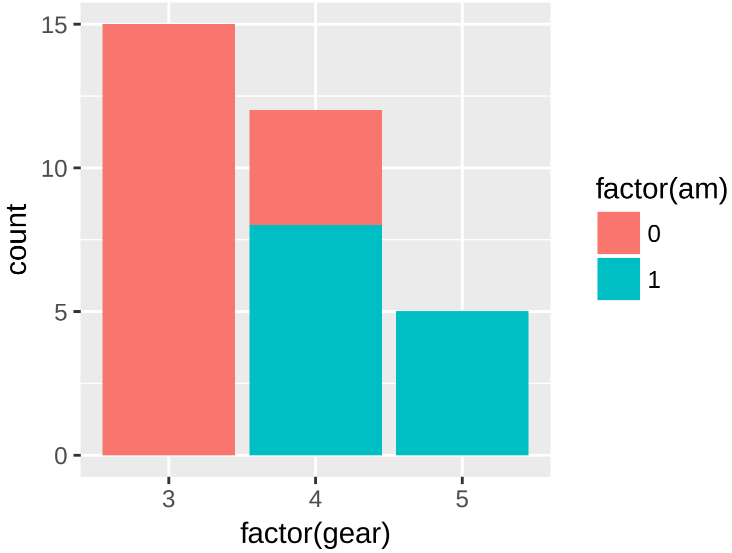
You would use a histogram when you are interested in frequencies of certain categories, like number of people with different eye colors. Note that we specify a single variable.

```
# Let us report the number of cars with differing number of front gears
qplot(factor(gear), data=mtcars, geom="bar") # used factor to declare categorical
```



If we want to get fancy and want to report across two categorical variables we can color the bars based on another variable.

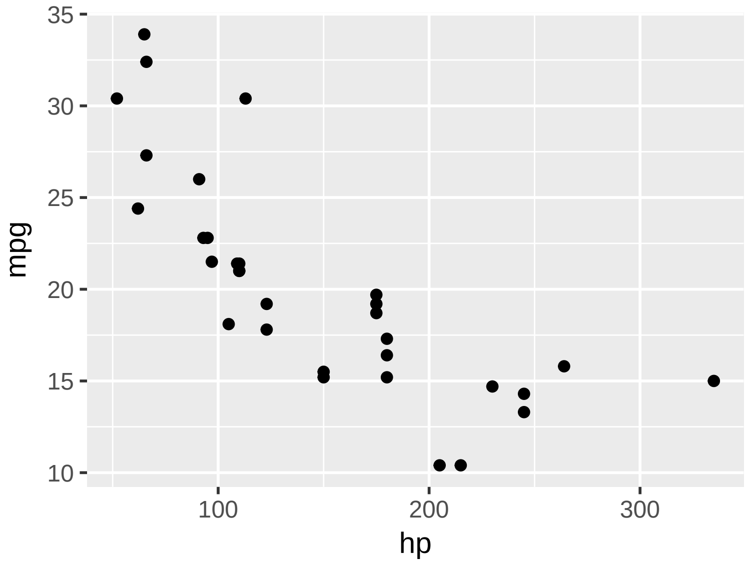
```
qplot(factor(gear), data=mtcars,  fill=factor(am), geom="bar") # used factor to declare categorical
```



Scatter Plots

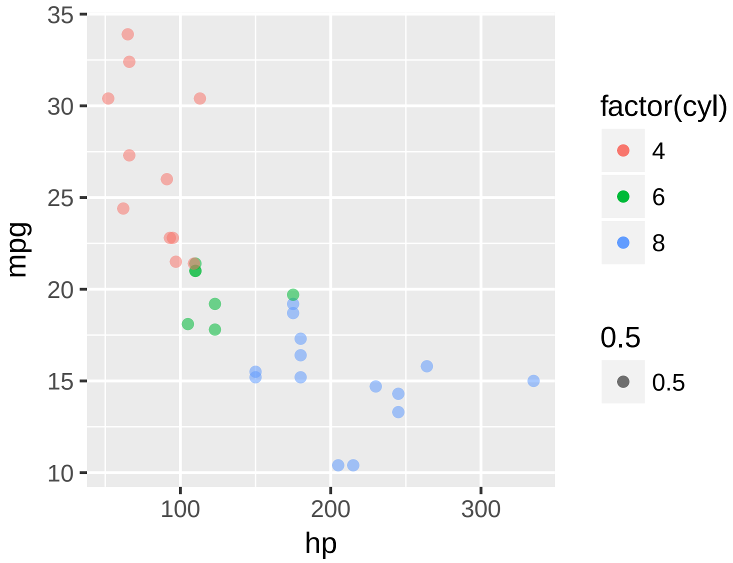
If you are interested in the relationship between two continuous variables, you can use scatter plots.

```
qplot(hp, mpg, data=mtcars)
```



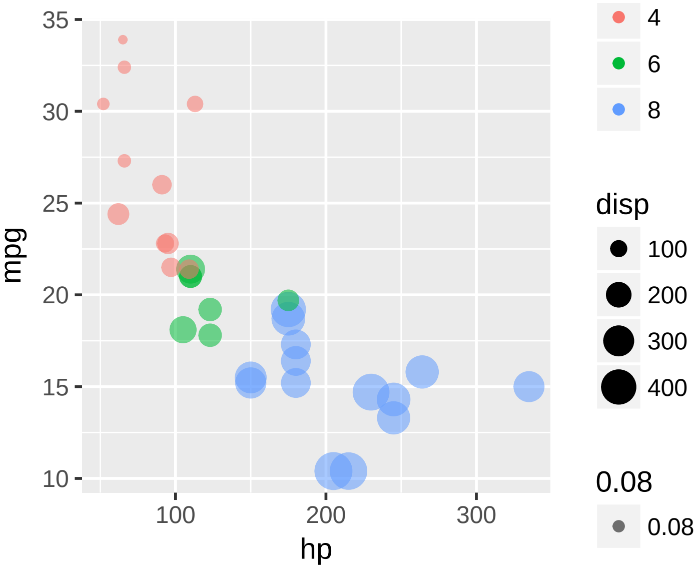
Let us impose an additional factor into the plot. Let us color the dots by the number of cylinders.

```
qplot(hp, mpg, data=mtcars, color=factor(cyl), alpha=.5)
```



Size of dots dependent on a continuous variable (displacement).

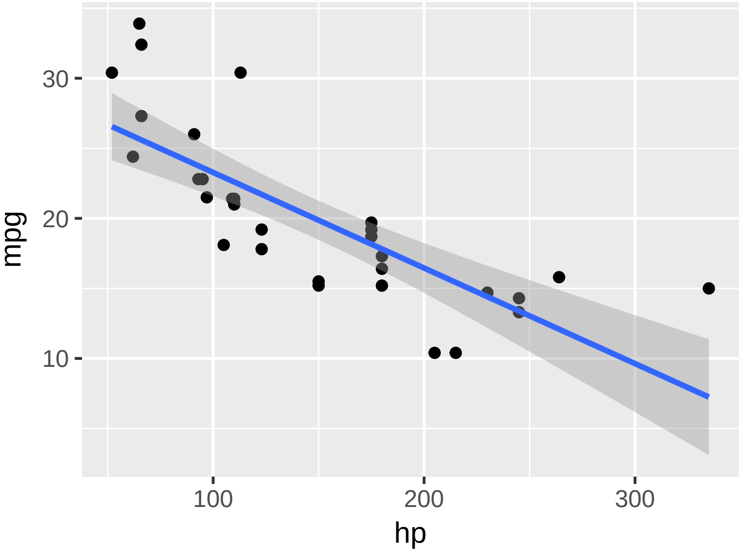
```
qplot(hp, mpg, data=mtcars, color=factor(cyl), size=disp, alpha=.08)
```



Let us fit a regression line. This is where things start to get a bit ggplotty.

```
qplot(hp, mpg, data=mtcars) +
  geom_smooth(method=lm, sd=F)
```

```
## Warning: Ignoring unknown parameters: sd
```

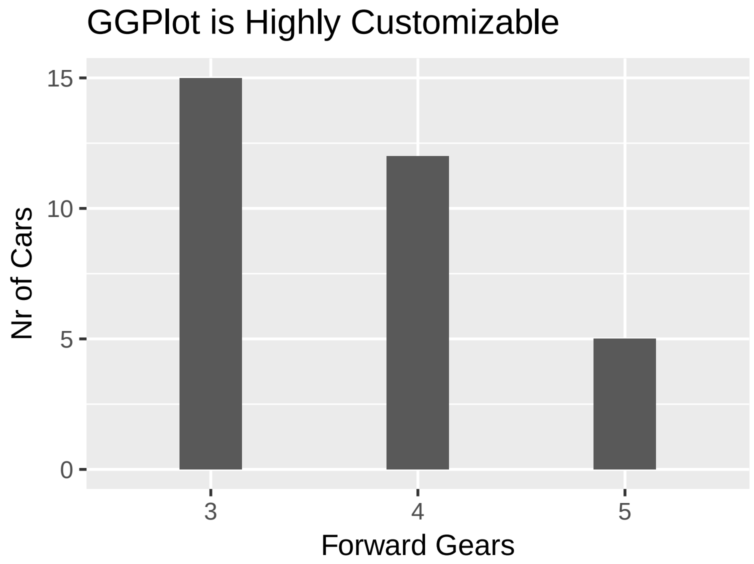


ggplot

qplot provides a convenient command for plotting. While qplot would address 90% of your plotting needs. ggplot is way more than qplot, it is almost a different language just for plotting. The intricacies may be hard to learn and is clearly beyond the scope of this workshop. I am providing ggplot code below to achieve the same results as the qplot, so the attendees can get a sense of what ggplot is really about.

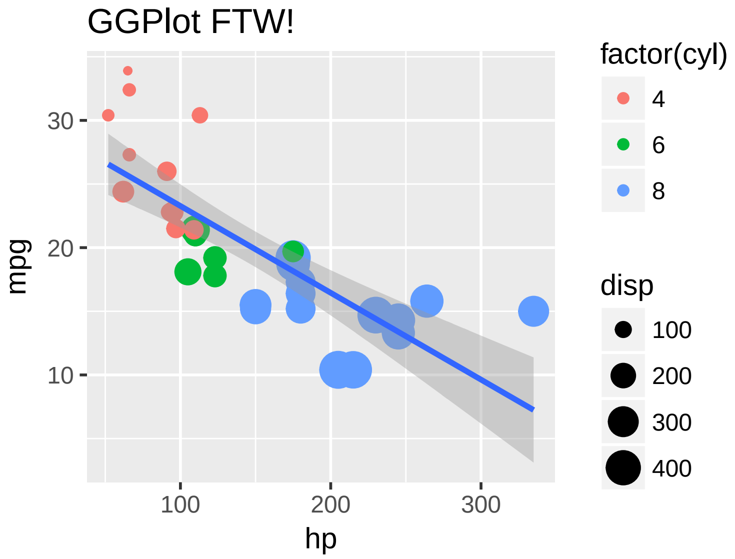
Histogram

```
# Initialize the plot with variables of interest
ggplot(mtcars, aes(factor(gear))) +
# Instruct ggplot to plot bars of width .3
  geom_bar(width=0.3) +
  ggtitle('GGPlot is Highly Customizable') +
  xlab('Forward Gears') +
  ylab('Nr of Cars')
```



Scatter Plot

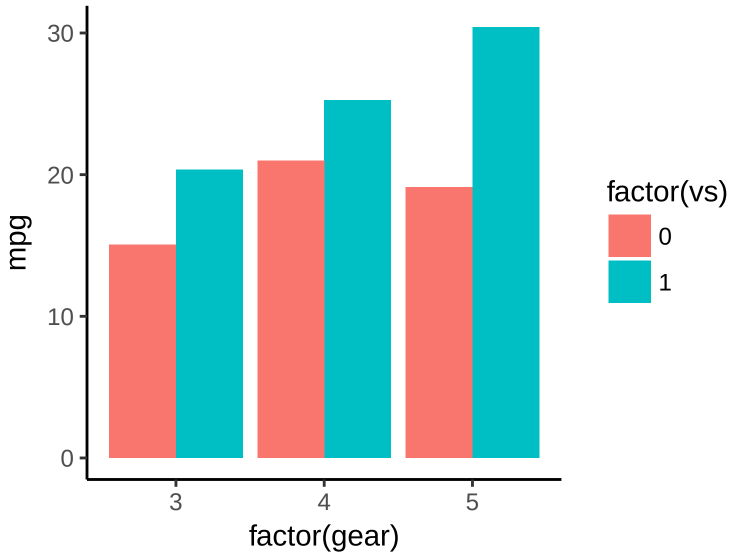
```
ggplot(mtcars, aes(x=hp, y=mpg)) +
  geom_point(aes(color=factor(cyl), size=disp)) + # For scatter plot
    geom_smooth(method=lm) + # Add a regression line
  ggtitle('GGPlot FTW!') # Add a title
```



Bar Charts

You use bar charts when you want to visualize the relationship of a continuous variable over a categorical variable (eg. gender-height). Here I plot mean mpg over two categorical variables.

```r
ggplot(mtcars,aes(x=factor(gear),y=mpg,fill=factor(vs)), color=factor(vs)) +
  stat_summary(fun.y=mean, position=position_dodge(), geom="bar") +
  theme_classic() # BONUS: Nicer looking
```



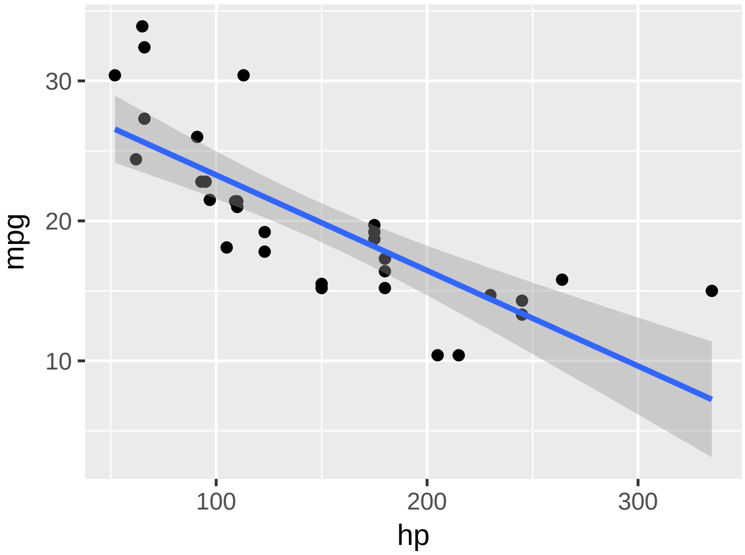
Statistical Models

In this section, I will try to provide an introduction to using two simple statistical models in R: regression and logistic regression.

Regression

If your dependent variable is continuous you can simply use regression. Regression is a good option if you are interested in explaining the role of each factor in determining the outcome. What regression does can be explained as fitting a line that best explains the data.

```r
ggplot(mtcars, aes(x=hp, y=mpg)) +
  geom_point() + # For scatter plot
    geom_smooth(method=lm)
```



The line in the scatter plot shows you the relationship between horse power and miles per galon. As you can see each additional unit of horse power reduces the mpg by an amount equal to the slope of the regression line.

For this demonstration, I will use the same Motor Trends dataset I used in Visualization section.

```
data(mtcars) # Get the data
?mtcars # Help on dataset
```

We will use lm() function to fit regular regression.

```
?lm
```

Below I declare a model where I use horse power, cylinders, and transmission type to estimate gas milage. Pay attention to model specification:

`mpg ~ hp + cyl + am`

Here the left hand side of the tilde is the dependent variable. and the right hand side has all the predictors we use separated by plus signs.

```
# Fit
reg_0 <- lm( mpg ~ hp + cyl + am, data = mtcars)
summary(reg_0)
```

```
##
## Call:
## lm(formula = mpg ~ hp + cyl + am, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.864 -1.811 -0.158  1.492  6.013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.88834    2.78422  11.094 9.27e-12 ***
## hp          -0.03688    0.01452  -2.540  0.01693 *
## cyl         -1.12721    0.63417  -1.777  0.08636 .
## am           3.90428    1.29659   3.011  0.00546 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.807 on 28 degrees of freedom
## Multiple R-squared:  0.8041, Adjusted R-squared:  0.7831
## F-statistic: 38.32 on 3 and 28 DF,  p-value: 4.791e-10
```

Look at the R-squared value to see how much variance is explained by the model, the more the better.

You can access estimated values as follows. I used a head function to limit the output.

```
head(reg_0$fitted.values)
```

```
##         Mazda RX4     Mazda RX4 Wag        Datsun 710    Hornet 4 Drive
##          23.97302          23.97302          26.85433          20.06874
## Hornet Sportabout           Valiant
##          15.41740          20.25312
```

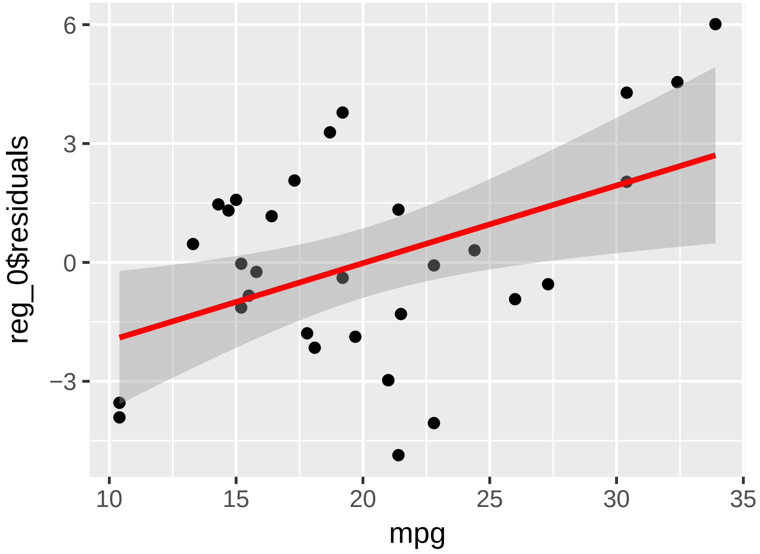
You can use the fitted model to predict new datasets. Here I am modifying Datsun710 to see how the gas milage may have been influenced if the car was automatic instead of manual transmission.

```
newCar <- mtcars[3,] # 3rd observation is Datsun 710
newCar$am <- 0 # What if it was automatic?
predict(reg_0, newdata = newCar) # Estimate went down by 4 miles
```

```
## Datsun 710
##   22.95005
```

One way to see how your model did is to plot residuals. Ideally the residuals should be close to 0 and randomly distributed. If you see a pattern, it indicates misspecification.

```
library(ggplot2)
# Plot the fitted values against real values
qplot(data=mtcars, x = mpg, y = reg_0$residuals) +
  stat_smooth(method = "lm", col = "red")
```



```
## Diagnostics

# Normality (p<0.05 indicates NN)
shapiro.test(reg_0$residuals)
```

```
##
```

```
##  Shapiro-Wilk normality test
## 
## data:  reg_0$residuals
## W = 0.98366, p-value = 0.8961
```

```r
# Multicollienarity (vif>10 indicates MC)
library(car)
vif(reg_0)
```

```
##       hp      cyl       am 
## 3.900138 5.048209 1.647399 
```

```r
# Homoscedasticity (p<0.05 indicates Heteroscedasticity)
ncvTest(reg_0)
```

```
## Non-constant Variance Score Test 
## Variance formula: ~ fitted.values 
## Chisquare = 3.744471    Df = 1     p = 0.05298249
```

Comparing models. If you are using the same dataset, and just adding or removing variables to a model. You can compare models with a likelihood ratio test or an F test. Anova facilitates comparison of simple regression models.

```r
# Add variable wt
reg_1 <- lm( mpg ~ hp + cyl + am + wt, mtcars)

# Aikikae Information Criteria
# AIC lower the better
AIC(reg_0)
```

```
## [1] 162.5849
```

```r
AIC(reg_1)
```

```
## [1] 156.2536
```

```r
# Compare
anova(reg_0, reg_1) # models are significantly different
```

```
## Analysis of Variance Table
## 
## Model 1: mpg ~ hp + cyl + am
## Model 2: mpg ~ hp + cyl + am + wt
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)   
## 1     28 220.55                                
## 2     27 170.00  1    50.555 8.0295 0.008603 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Logistic Regression

Let us change gears and try to predict a binary variable. For this purpose we will use the logistic regression with a binomial link function. The model estimates the probability of Y=1. Using a linear regression model is inadvisable as the probability is constrained between 0 and 1. Thus we use a link function to transform regression to be limited between 0 and 1.

Let us stick to the mtcars dataset and try to figure out if a car is automatic or manual based on predictors.

We will use glm function.

```
?glm
```

Let us fit the model

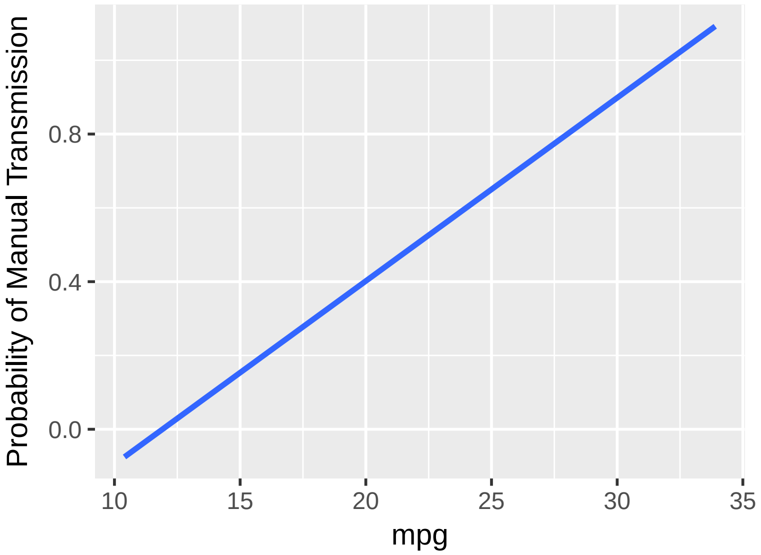
```
logit_2 <- glm(am ~ mpg + drat + cyl, data = mtcars, family='binomial')
summary(logit_2)
```

```
##
## Call:
## glm(formula = am ~ mpg + drat + cyl, family = "binomial", data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.58367  -0.31020  -0.03757   0.17972   1.75395
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -49.4548    24.1280  -2.050   0.0404 *
## mpg           0.6378     0.4266   1.495   0.1349
## drat          7.2595     3.2702   2.220   0.0264 *
## cyl           1.6115     1.0801   1.492   0.1357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.23  on 31  degrees of freedom
## Residual deviance: 17.03  on 28  degrees of freedom
## AIC: 25.03
##
## Number of Fisher Scoring iterations: 7
```

Visualize the results.

```
ggplot(mtcars, aes(x = mpg, y = am)) +
    stat_smooth(method="glm", family="binomial", se=FALSE)+
# Bonus: rename the y axis label
        ylab('Probability of Manual Transmission')
```

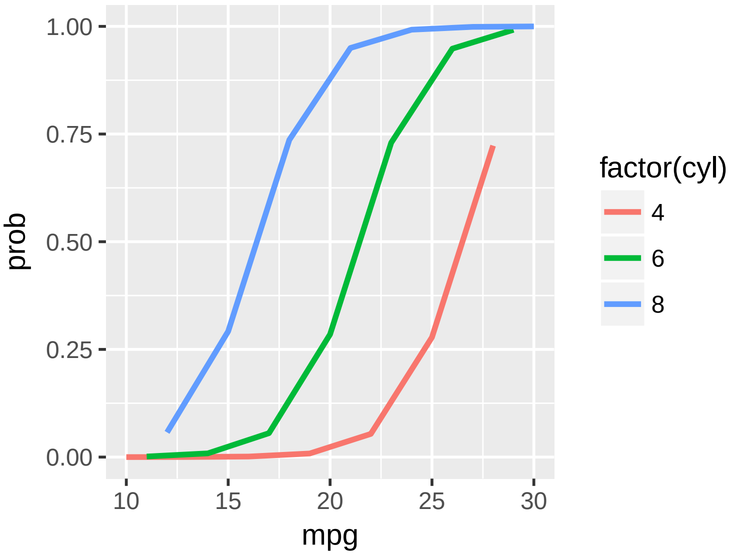
```
## Warning: Ignoring unknown parameters: family
```



How about plotting results for number of cylinders? We will need to process the data a little bit.

```
# Create a new dataset with varying number of cylinders and other variables fixed at mean levels.
mtcars2<-data.frame(mpg = rep(10:30, 3),drat = mean(mtcars$drat), disp = mean(mtcars$disp), cyl = rep(c
# Predict probability of new data
mtcars2$prob<-predict(logit_2, newdata=mtcars2, type = "response")

# Plot the results
ggplot(mtcars2, aes(x=mpg, y=prob)) +
  geom_line(aes(colour = factor(cyl)), size = 1)
```



Diagnostics with logistic regression.

```
# Let us compare predicted values to real values
mtcars$prob <- predict(logit_2, type="response")
# Prevalence of Manual Transmission
mean(mtcars$am)
```

```
## [1] 0.40625
```

```
# Create predict variable
mtcars$pred <- 0
# If probability is greater than .6 (1-prevalence), set prediction to 1
mtcars[mtcars$prob>.6, 'pred'] <- 1

# Predictions versus reality
table(mtcars[,c("am", "pred")])
```

```
##    pred
## am   0  1
##   0 18  1
##   1  3 10
```

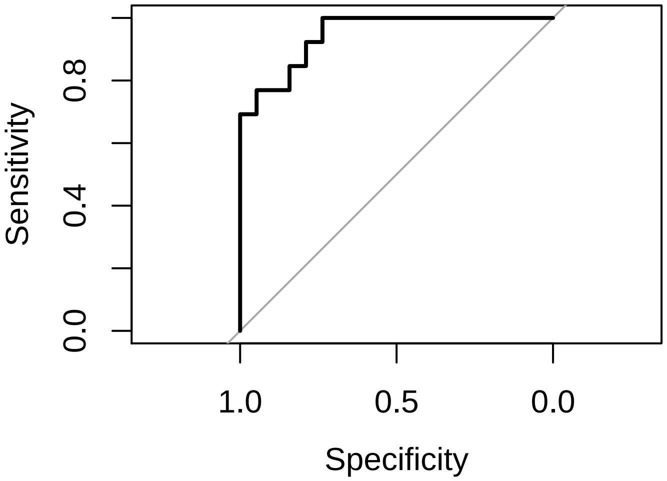
```
## ROC CURVE
# Load the necessary library
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Calculate the ROC curve using the predicted probability vs actual values
logit_2_roc <- roc(am~prob, mtcars)
# Plot ROC curve
plot(logit_2_roc)
```


Sensitivity
Specificity