Regression and GLMs

HDS

2024-09-02

Pair Programming and Generalized Linear Models, Module 07

DSE5001

Created 09/02/2024 checked 01/03/2025

Name:Ryan Waterman Partner Name:

Date:2/26/2025

```
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.4     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

Data set

We've used mtcars a lot, but it's an easy set to understand, and it has both continuous variables and categories

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

We will want to create cyl,vs, am and gear as factors later in this example

```
mtcars$cyl=factor(mtcars$cyl)
mtcars$am=factor(mtcars$am,labels=c("auto","manual"))
mtcars$vs=factor(mtcars$vs,labels=c("V","inline"))
mtcars$gear=factor(mtcars$gear)
```

Repeated Use of this set

I am going to show a lot of examples of calculations all using this same data set, this saves a lot of time. This repeated use of a dataset, running a whole series of closely related analyses is not something one would do in the real world.

On targets the specific analysis used to the questions being asked and the data in use. The hodgepodge of analyses run here is just to show how they work

Linear Regression

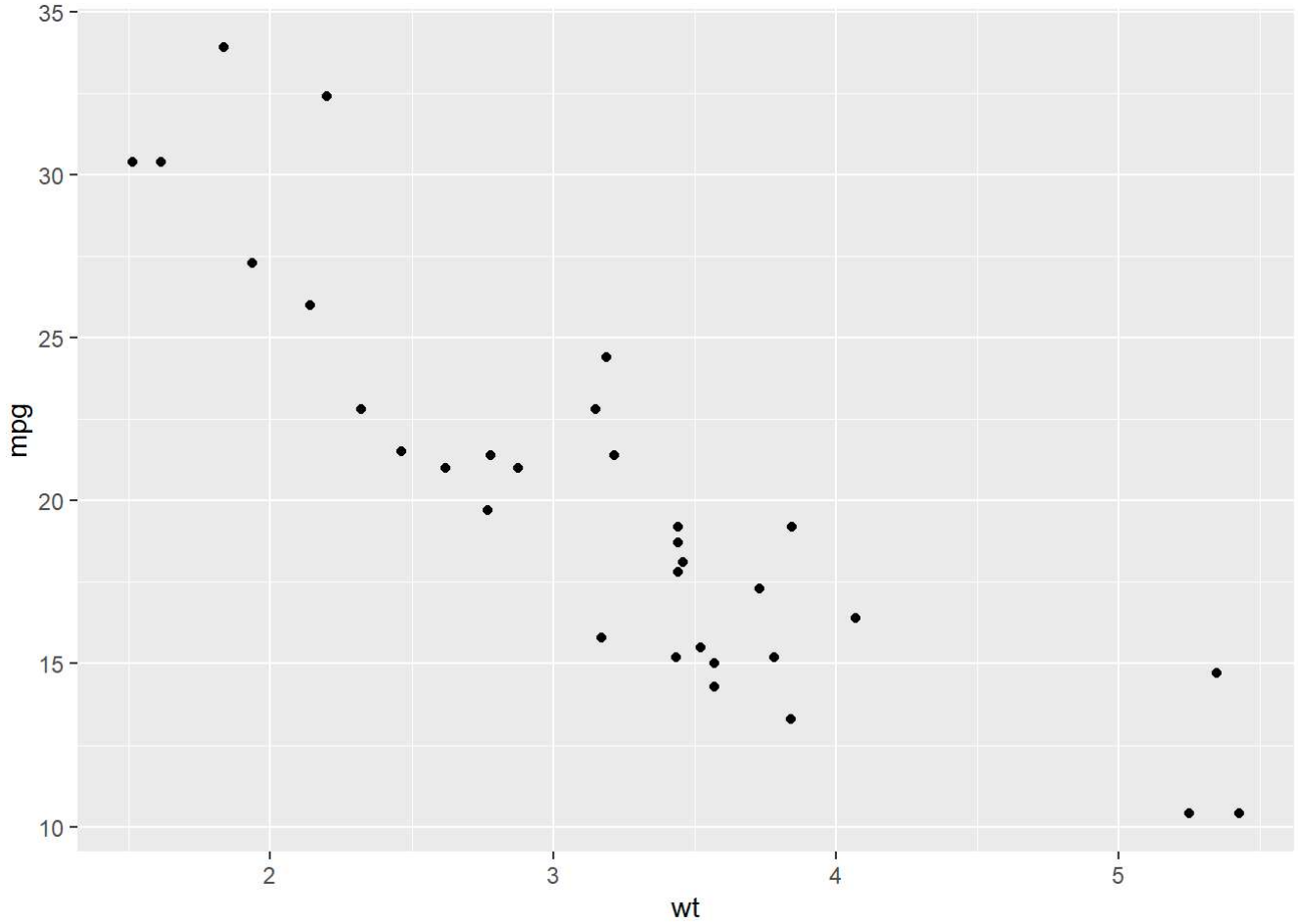
Hypothesis or Modeling Goal, *I think heavy vehicles will get poor mileage*

so mpg should correlate negatively with weight

We will

-plot mpg vs weight -fit a linear regression and look at the results -obtain the residuals -test the results for violation of the assumptions

```
ggplot(mtcars, aes(x=wt,y=mpg))+geom_point()
```



Not terrible, it looks roughly linear, although the three points on the far left look like possible outliers

Let's create the model

the statement, "mpg~wt" is an R formula, the ~ reads as "predicted by", so mpg is predicted by wt in this model, R will add the intercept as well

We are using the lm() function to run a classic regression here

```
Model_1=lm(mpg~wt,data=mtcars)

summary(Model_1)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

Summary(Model_1)

The summary tells us a lot about the model.

The F statistic is 91.38, with a p-value of 1.3e-10, so that is a large F value and the model has statistically significant explanatory power

The adjusted R^2 is 0.7446, again, substantial, the model explains about 74.5% of the variance in mpg of the car.

The b value (or intercept) is 37.3, with a standard error of 1.9, it's t-value is 19.9, so it is clearly not zero

The slope is listed as (wt), meaning it is the slope for the weight. This value is -5.3, so mpg decreases with weight as expected, the standard error in the slope (m) is about 0.56, so the 95% confidence interval would be -5.3+-1.96*0.56, rather wide, but it excludes zero, so this model is statistically meaningful

Looking at the residuals

What do we have in Model_1

```
str(Model_1)
```

```
## List of 12
##  $ coefficients : Named num [1:2] 37.29 -5.34
##   ..- attr(*, "names")= chr [1:2] "(Intercept)" "wt"
##  $ residuals    : Named num [1:32] -2.28 -0.92 -2.09 1.3 -0.2 ...
##   ..- attr(*, "names")= chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive"
...
##  $ effects      : Named num [1:32] -113.65 -29.116 -1.661 1.631 0.111 ...
##   ..- attr(*, "names")= chr [1:32] "(Intercept)" "wt" "" "" ...
##  $ rank         : int 2
##  $ fitted.values: Named num [1:32] 23.3 21.9 24.9 20.1 18.9 ...
##   ..- attr(*, "names")= chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive"
...
##  $ assign       : int [1:2] 0 1
##  $ qr           :List of 5
##   ..$ qr   : num [1:32, 1:2] -5.657 0.177 0.177 0.177 0.177 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##   .. .. ..$ : chr [1:2] "(Intercept)" "wt"
##   .. ..- attr(*, "assign")= int [1:2] 0 1
##   ..$ qraux: num [1:2] 1.18 1.05
##   ..$ pivot: int [1:2] 1 2
##   ..$ tol  : num 1e-07
##   ..$ rank : int 2
##   ..- attr(*, "class")= chr "qr"
##  $ df.residual  : int 30
##  $ xlevels      : Named list()
##  $ call         : language lm(formula = mpg ~ wt, data = mtcars)
##  $ terms        :Classes 'terms', 'formula'  language mpg ~ wt
##   .. ..- attr(*, "variables")= language list(mpg, wt)
##   .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "mpg" "wt"
##   .. .. .. ..$ : chr "wt"
##   .. ..- attr(*, "term.labels")= chr "wt"
##   .. ..- attr(*, "order")= int 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(mpg, wt)
##   .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. ..- attr(*, "names")= chr [1:2] "mpg" "wt"
##  $ model        :'data.frame':   32 obs. of  2 variables:
##   ..$ mpg: num [1:32] 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##   ..$ wt : num [1:32] 2.62 2.88 2.32 3.21 3.44 ...
##   ..- attr(*, "terms")=Classes 'terms', 'formula'  language mpg ~ wt
##   .. .. ..- attr(*, "variables")= language list(mpg, wt)
##   .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:2] "mpg" "wt"
##   .. .. .. .. ..$ : chr "wt"
##   .. .. ..- attr(*, "term.labels")= chr "wt"
##   .. .. ..- attr(*, "order")= int 1
```

```
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, "predvars")= language list(mpg, wt)
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. .. ..- attr(*, "names")= chr [1:2] "mpg" "wt"
##  - attr(*, "class")= chr "lm"
```

Okay, the residuals are in Model_1$residuals

Let's plot a histogram

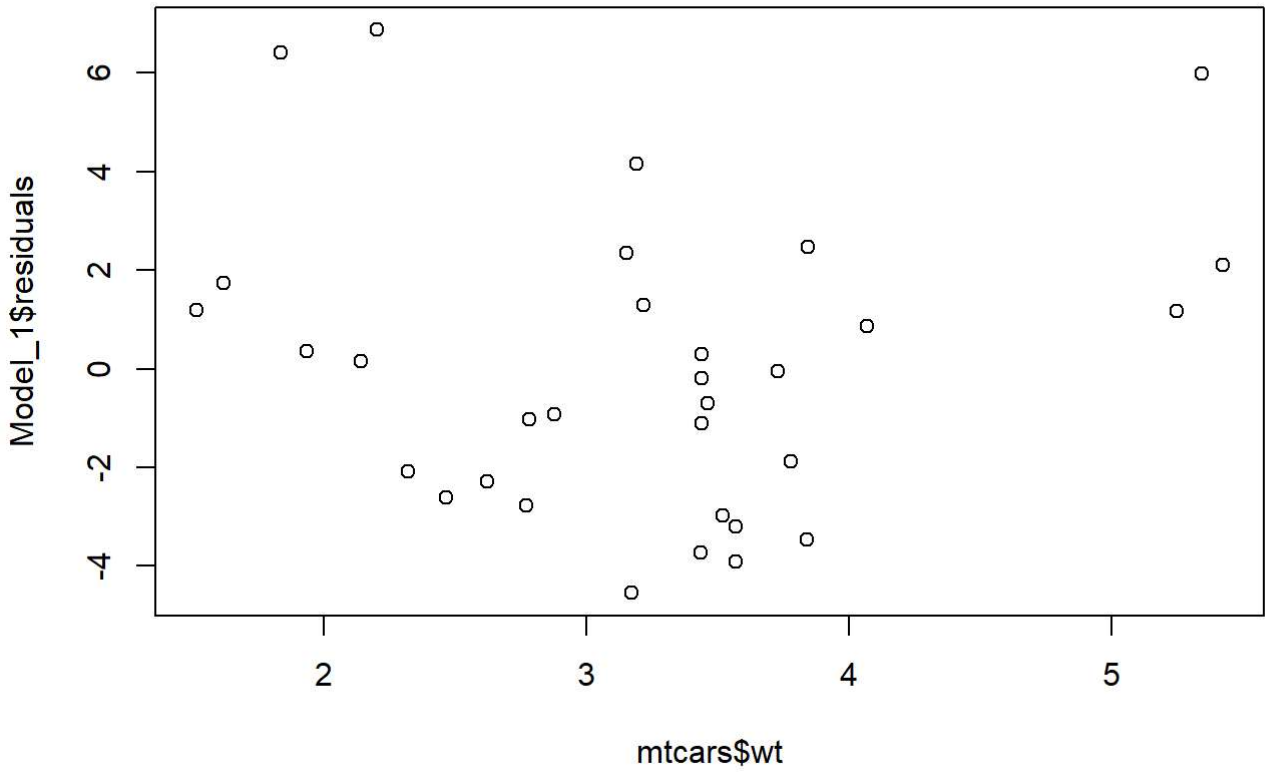
```
hist(Model_1$residuals)
```

Well,

not terrible at least, maybe slightly positive right tail?

residuals vs weight

```
plot(mtcars$wt,Model_1$residuals)
```



Neither great nor terrible, maybe slightly U shaped??

There is a statistical test for normality, we can see if we can reject the null hypothesis of normality, so we are asking if there is strong evidence the residuals are not normal

THis is a test called the Wilks-Shapiro test of normality

```
shapiro.test(Model_1$residuals)
```

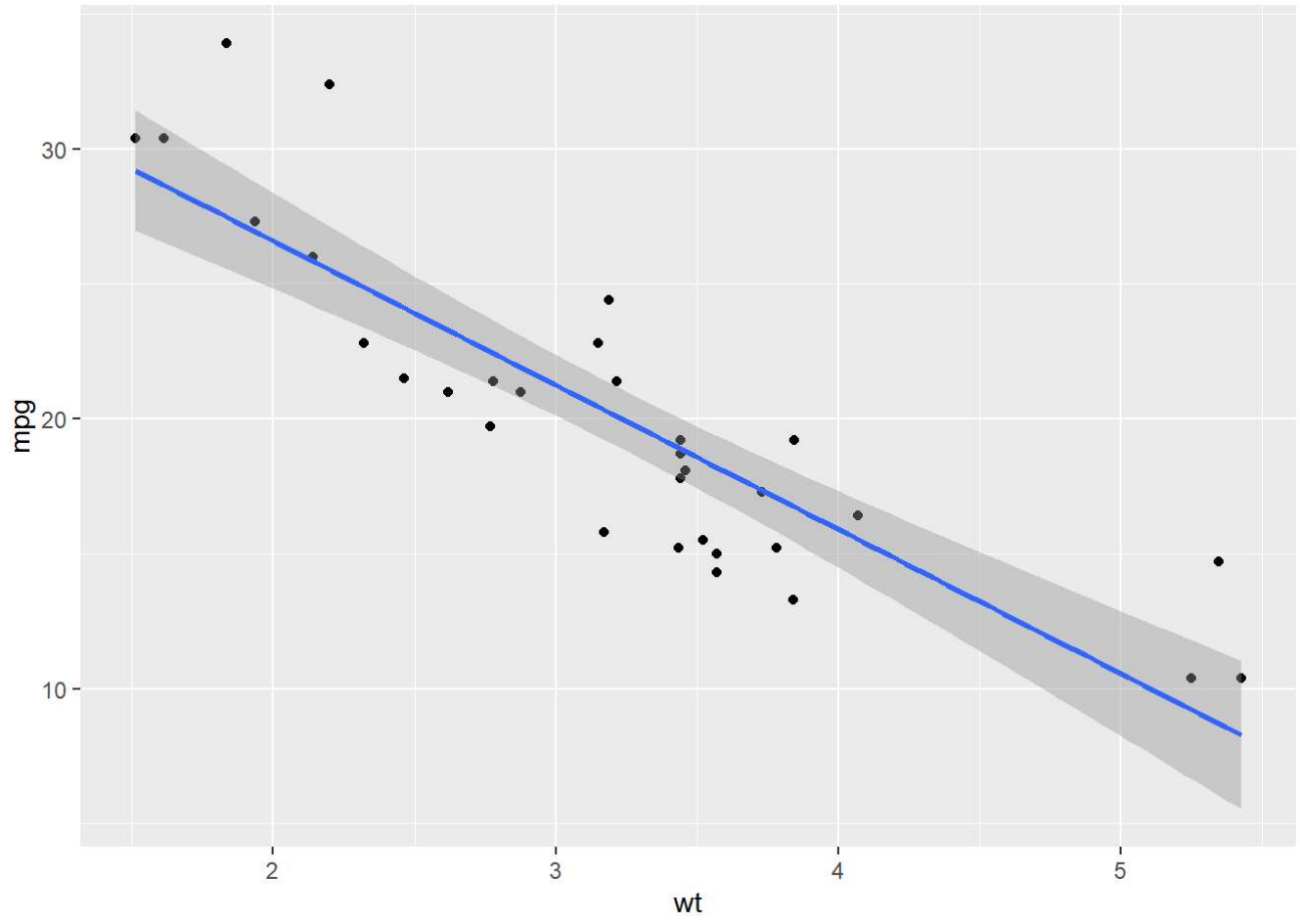
```
##
##  Shapiro-Wilk normality test
##
## data:  Model_1$residuals
## W = 0.94508, p-value = 0.1044
```

We cannot convincingly reject the null that the residuals are normal.

Ggplot will show us the regression model with a confidence interval

```
ggplot(mtcars,aes(x=wt,y=mpg)) +geom_point()+geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Question/Action

Build a regression model

We will use the lime data set

```
library(GLMsData)
data(lime)
head(lime)
```

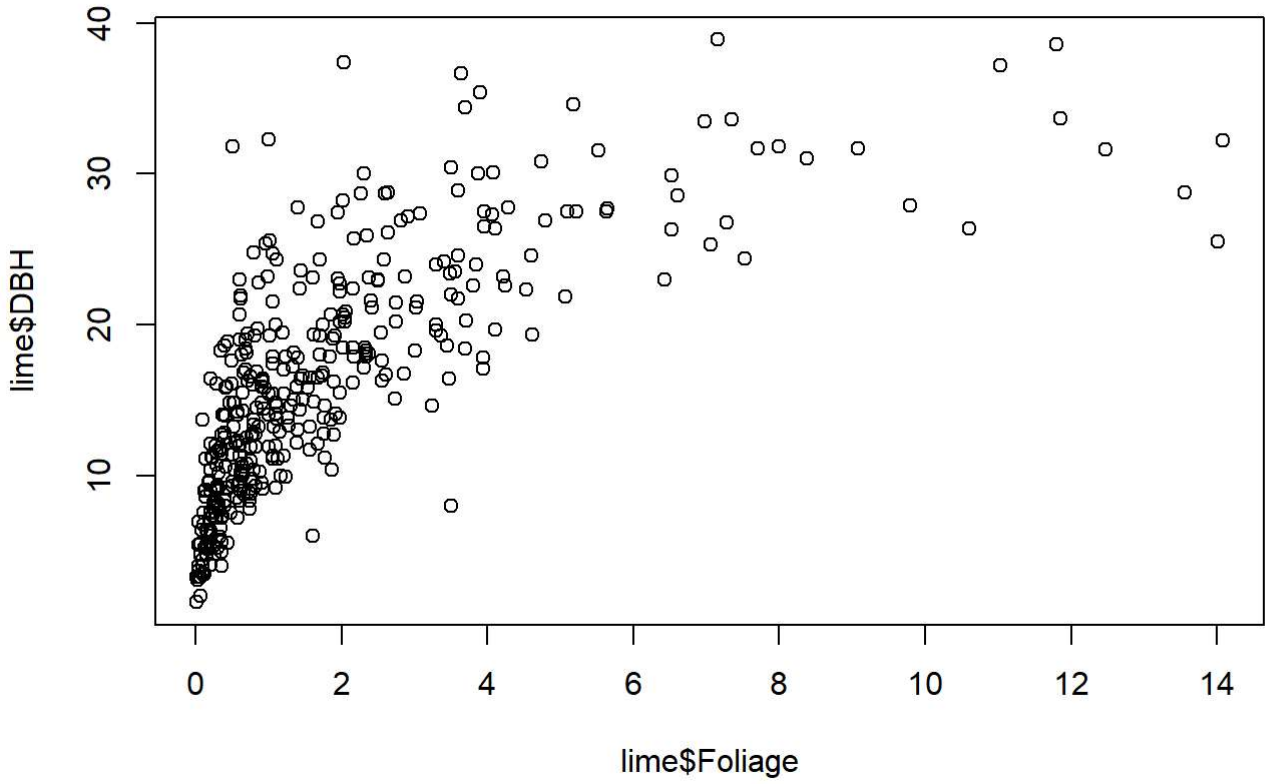
```
##   Foliage  DBH Age  Origin
## 1     0.1  4.0  38 Natural
## 2     0.2  6.0  38 Natural
## 3     0.4  8.0  46 Natural
## 4     0.6  9.6  44 Natural
## 5     0.6 11.3  60 Natural
## 6     0.8 13.7  56 Natural
```

Foliage- biomass of foliage DBH-tree diameter Age-age of the tree Origin- origin of the tree, coppice, Natural, Planted

Do the following:

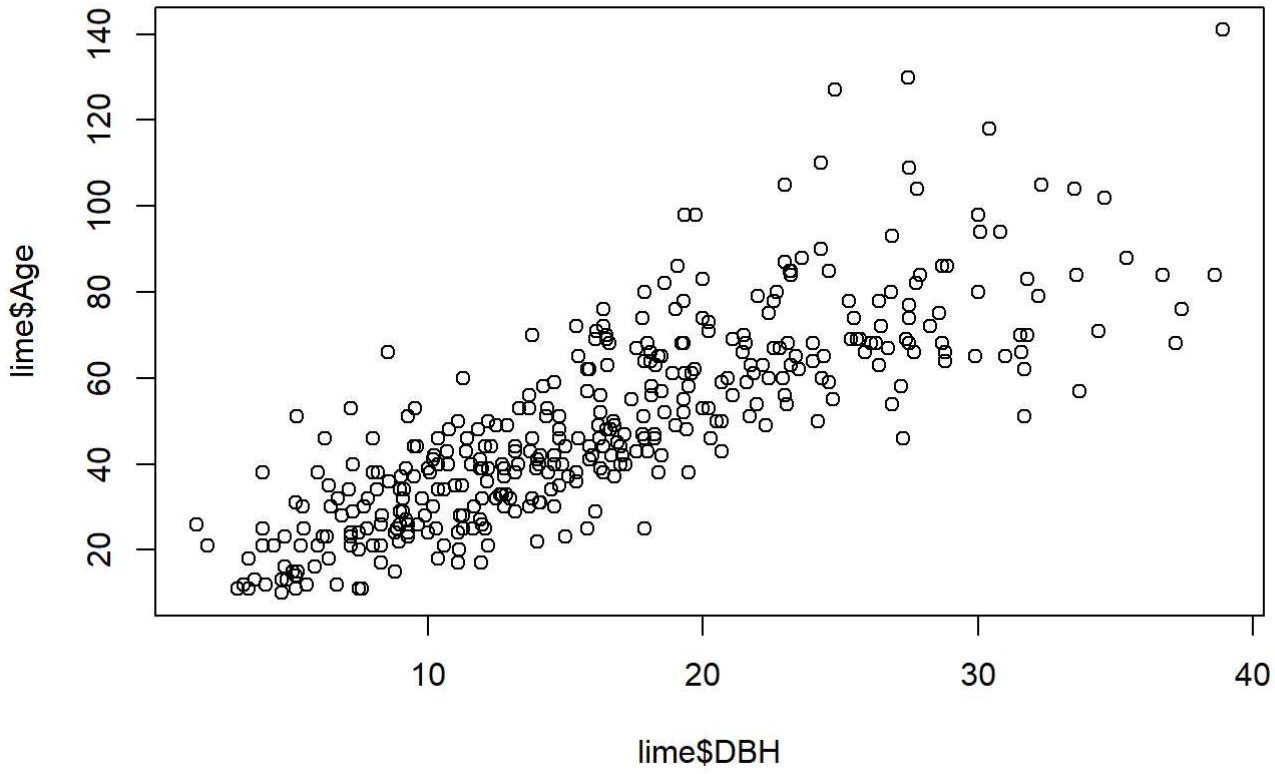
-Plot Foliage vs DBH

```
plot(lime$Foliage, lime$DBH)
```



-Plot DBH vs Age

```
plot(lime$DBH, lime$Age)
```



Which looks more like to have a strong linear relationship?

DBH vs. Age

Create a linear model of Foliage~DBH, or DBH~Age

```
model=lm(DBH~Age, lime)
summary(model)
```

```
##
## Call:
## lm(formula = DBH ~ Age, data = lime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0350  -3.0083  -0.1627   2.5708  15.7440
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5785     0.5478   4.707 3.52e-06 ***
## Age           0.2776     0.0100  27.761  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.582 on 383 degrees of freedom
## Multiple R-squared:  0.668,  Adjusted R-squared:  0.6671
## F-statistic: 770.7 on 1 and 383 DF,  p-value: < 2.2e-16
```

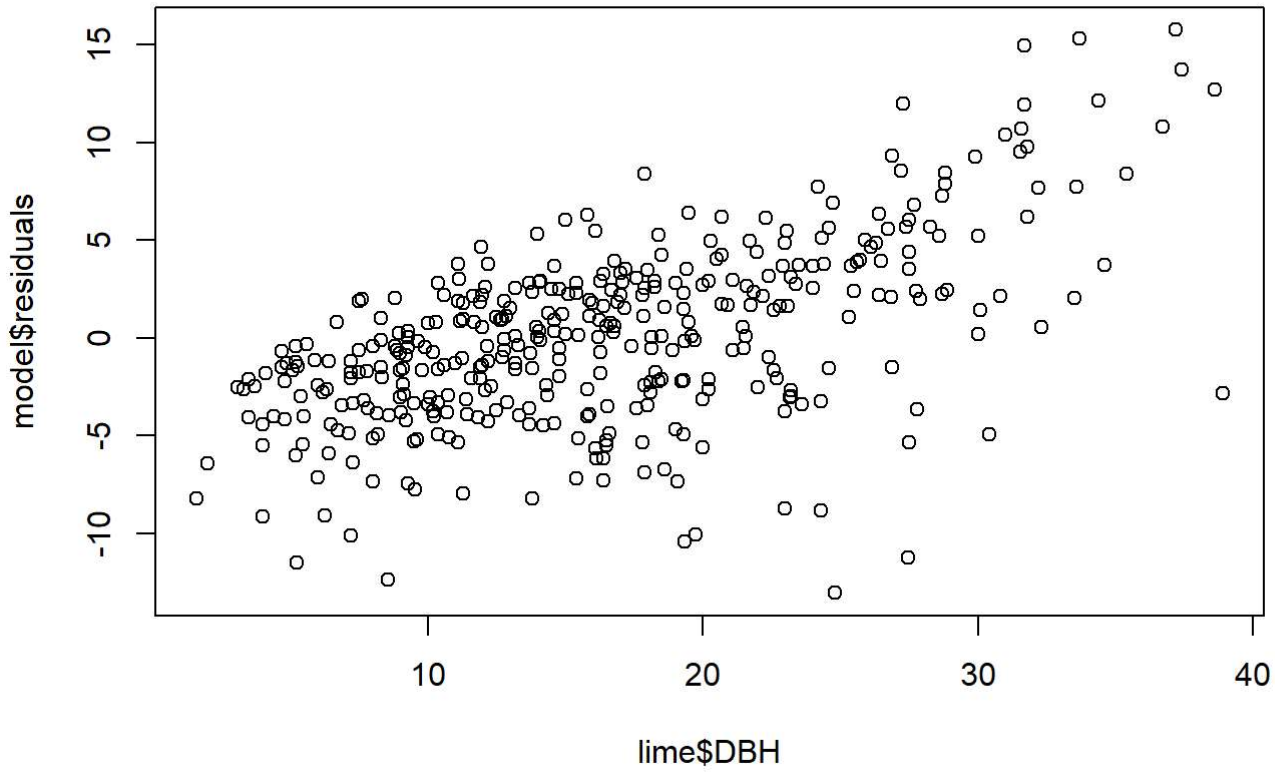
Get the statistics and explain what they mean

An R-squared of 0.668 suggests that the linear model fits the data moderately well. The small p-value indicates a rejection of the null hypothesis, and the accompanying large F-value shows that the relationship between the variables is strong.

Look at the residuals, plot a histogram and residuals vs x. Run the Wilk's Shapiro test

```
hist(model$residuals)
```


```
plot(lime$DBH,model$residuals)
```




```
shapiro.test(model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals
## W = 0.98486, p-value = 0.0004683
```

Explain what you see- I'm looking for a fair amount of detail in this discussion.

The Shapiro-Wilk normality test is used to test if the input sample is normal. In this case, we are testing that the residuals of the linear model are normal. A small p-value indicates the null hypothesis is rejected, therefore the residuals are not normal, even though the W statistic is nearly 1. The W statistic measures how well a model conforms to the normal distribution, but the p-value takes precedent, and it suggests that the null hypothesis is rejected.

General Linear Models

We will run the same analysis using a different function, a generalized linear model (glmer)

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

In practice, I don't bother with lm(), I just go directly to glm() or similar models

```
Model_2=glm(mpg~wt,data=mtcars)

summary(Model_2)
```

```
##
## Call:
## glm(formula = mpg ~ wt, data = mtcars)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 9.277398)
##
##     Null deviance: 1126.05  on 31  degrees of freedom
## Residual deviance:  278.32  on 30  degrees of freedom
## AIC: 166.03
##
## Number of Fisher Scoring iterations: 2
```

This looks much like the set of results we got from lm(), but it doesn't have the F or R^2, but it does have the AIC score

We can use the anova function to get more information (like the F score for the model)

```
anova(Model_2,test=c("F"))
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: mpg
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev      F    Pr(>F)
## NULL                    31    1126.05
## wt    1   847.73        30     278.32 91.375 1.294e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that we get an F score for the wt as a predictor, as well as the related pvalue

We can calculate a R^2 for the model, this is McFadden's R-squared

```
with(summary(Model_2),1-deviance/null.deviance)
```

```
## [1] 0.7528328
```

Which is the same Rsquare we saw for the LM (but not the adjusted R^2)

Multiple Continuous predictors

Okay lets use wt,hp and disp (engine displacement) to predict mpg

the formula is mpg~wt+hp+disp, meaning we want to use all three predictors

```
Model_3=glm(mpg~hp+wt+disp,data=mtcars)
summary(Model_3)
```

```
##
## Call:
## glm(formula = mpg ~ hp + wt + disp, data = mtcars)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.105505   2.110815  17.579  < 2e-16 ***
## hp          -0.031157   0.011436  -2.724  0.01097 *
## wt          -3.800891   1.066191  -3.565  0.00133 **
## disp        -0.000937   0.010350  -0.091  0.92851
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6.963953)
##
##     Null deviance: 1126.05  on 31  degrees of freedom
## Residual deviance:  194.99  on 28  degrees of freedom
## AIC: 158.64
##
## Number of Fisher Scoring iterations: 2
```

when we look at this set of results, the t value for disp is-0.091, with a p of 0.928, so when we already have hp and wt as predictors, disp doesn't help any. The other two predictor are both significant

```
anova(Model_3,test="F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: mpg
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                    31    1126.05
## hp    1   678.37        30     447.67 97.4120 1.285e-10 ***
## wt    1   252.63        29     195.05 36.2763 1.720e-06 ***
## disp  1     0.06        28     194.99  0.0082    0.9285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows use the F values for each predictor, again disp is not a useful predictor

```
with(summary(Model_3),1-deviance/null.deviance)
```

```
## [1] 0.8268361
```

Using 3 predictors, we are now predicting 82.6% of the variance, an improvement over the roughly 75% with one predictor

Using only categorical predictors

Let's create a table first

```
mtcars %>% group_by(am,cyl,vs) %>% summarise("Mean Mpg"=mean(mpg))
```

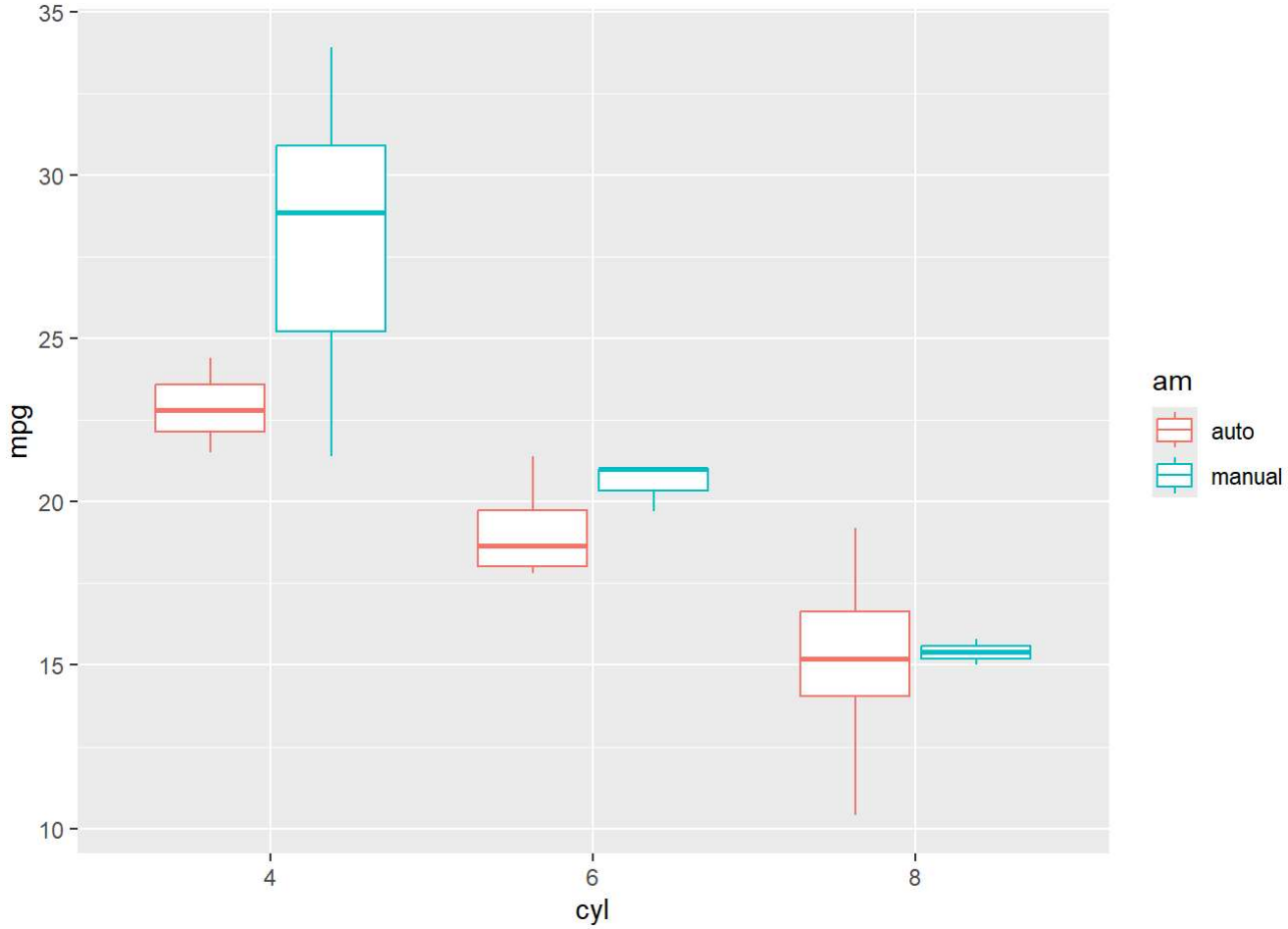
```
## `summarise()` has grouped output by 'am', 'cyl'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 7 × 4
## # Groups:   am, cyl [6]
##   am     cyl   vs     `Mean Mpg`
##   <fct>  <fct> <fct>       <dbl>
## 1 auto   4     inline       22.9
## 2 auto   6     inline       19.1
## 3 auto   8     V            15.0
## 4 manual 4     V            26
## 5 manual 4     inline       28.4
## 6 manual 6     V            20.6
## 7 manual 8     V            15.4
```

the variable vs doesn't look all that helpful,but we'll see

boxplots

```
ggplot(mtcars,aes(x=cyl,y=mpg,color=am))+geom_boxplot()
```



we will predict mpg using am, cyl, vs

```
Model_4=glm(mpg~am+cyl+vs,data=mtcars)
summary(Model_4)
```

```
##
## Call:
## glm(formula = mpg ~ am + cyl + vs, data = mtcars)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.809      2.928   7.789 2.24e-08 ***
## ammanual       3.165      1.528   2.071  0.04805 *
## cyl6          -5.399      1.837  -2.938  0.00668 **
## cyl8          -8.161      2.892  -2.822  0.00884 **
## vsinline       1.708      2.235   0.764  0.45135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 9.58871)
##
##     Null deviance: 1126.0  on 31  degrees of freedom
## Residual deviance:  258.9  on 27  degrees of freedom
## AIC: 169.71
##
## Number of Fisher Scoring iterations: 2
```

Looking at this

am manual is significant, as are the cylinder categories

note am auto is the default as is cyl=4 so a 4 cylinder auto has mpg=(Intercept)=22.8

A 4 cylinder with manual would have mpg=(Intercept)+(ammanual)=22.8+3.165= 25.9 mpg

vs is not significant (vs indicates a v piston configuration vs an inline engine)

```
anova(Model_4,test="F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: mpg
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                    31     1126.0
## am    1   405.15        30      720.9 42.2529 5.727e-07 ***
## cyl   2   456.40        28      264.5 23.7989 1.101e-06 ***
## vs    1     5.60        27      258.9  0.5841    0.4513
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows an F for each variable am and cyl are statistically meaning ful, vs is not

We can get the R^2

```
with(summary(Model_4),1-deviance/null.deviance)
```

```
## [1] 0.770085
```

GLM with both continuous variables and factors

We will use mpg predicted by hp, wt (continuous) and am,cyl (factors/categories)

```
Model_5=glm(mpg~am+cyl+hp+wt,data=mtcars)
summary(Model_5)
```

```
##
## Call:
## glm(formula = mpg ~ am + cyl + hp + wt, data = mtcars)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.70832    2.60489  12.940 7.73e-13 ***
## ammanual     1.80921    1.39630   1.296  0.20646
## cyl6        -3.03134    1.40728  -2.154  0.04068 *
## cyl8        -2.16368    2.28425  -0.947  0.35225
## hp          -0.03211    0.01369  -2.345  0.02693 *
## wt          -2.49683    0.88559  -2.819  0.00908 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 5.808677)
##
##     Null deviance: 1126.05  on 31  degrees of freedom
## Residual deviance:  151.03  on 26  degrees of freedom
## AIC: 154.47
##
## Number of Fisher Scoring iterations: 2
```

It now appears the category cyl=8 is not helpful, nor is am, due to the other predictors

Remember, the predictors are all correlated, so when we combine many of them in a model, only the most effective are typically useful

```
anova(Model_5,test="F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: mpg
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev      F    Pr(>F)
## NULL                    31    1126.05
## am    1   405.15        30     720.90 69.749 7.823e-09 ***
## cyl   2   456.40        28     264.50 39.286 1.388e-08 ***
## hp    1    67.30        27     197.20 11.585  0.002164 **
## wt    1    46.17        26     151.03  7.949  0.009081 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interesting, the F score indicates all the predictors are effective.

I'd go with the F-test, rather than the t

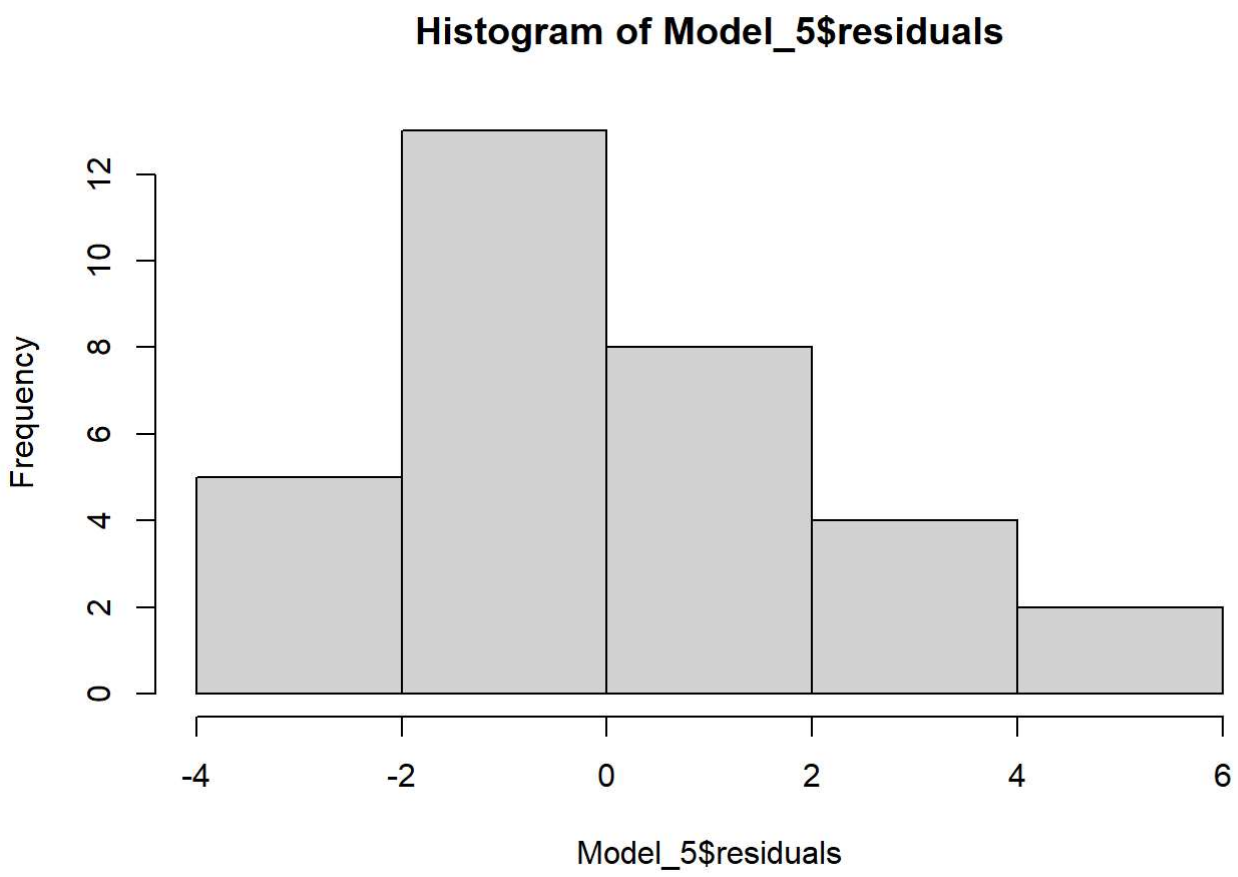
```
with(summary(Model_5),1-deviance/null.deviance)
```

```
## [1] 0.8658799
```

With a lot of predictors in the model, we are up to 87% prediction

We should look at the residuals

```
hist(Model_5$residuals)
```




```
shapiro.test(Model_5$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Model_5$residuals
## W = 0.96807, p-value = 0.4479
```

There is no strong evidence that the residuals are non-normal.

There is a risk here, we have failed to reject a null, which is always a weak argument.

Question/Action

Load the lungcap data

This is measured lung capacity data among children

```
data(lungcap)

head(lungcap)
```

```
##   Age   FEV Ht Gender Smoke
## 1   3 1.072 46      F     0
## 2   4 0.839 48      F     0
## 3   4 1.102 48      F     0
## 4   4 1.389 48      F     0
## 5   4 1.577 49      F     0
## 6   4 1.418 49      F     0
```

-What do you think the dominant factor determining lung capacity will be?

Based on the available variables, I would estimate height to the strongest factor influencing lung capacity, which will affect the lung volume based on overall size. This age range seems to span pre and post puberty, so there will be dramatic changes in the height of the subject over time based on age as well, but age is not a perfect predictor of height.

-Would the same factor dominate among adults?

No, I would expect smoker status to be the primary factor here. Adults are likely to have been smoking for much longer, and their health would have had more time to decline as a result. Also, age/height are now worse predictive factors because there could be very healthy or very unhealthy individuals in the same age/height range.

Age- subject age FEV- Forced expelled volume, a measure of lung capacity Ht- height in inches Gender- Smoke,- 0 is non smokers, 1 is smokers

Build a GLM of FEV ~ age, ht,gender and smoke

```
model_lungcap=glm(FEV~Age + Ht + Gender + Smoke, data=lungcap)
summary(model_lungcap)
```

```
##
## Call:
## glm(formula = FEV ~ Age + Ht + Gender + Smoke, data = lungcap)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.456974   0.222839 -20.001  < 2e-16 ***
## Age          0.065509   0.009489   6.904 1.21e-11 ***
## Ht           0.104199   0.004758  21.901  < 2e-16 ***
## GenderM      0.157103   0.033207   4.731 2.74e-06 ***
## Smoke       -0.087246   0.059254  -1.472    0.141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1699223)
##
##     Null deviance: 490.92  on 653  degrees of freedom
## Residual deviance: 110.28  on 649  degrees of freedom
## AIC: 703.79
##
## Number of Fisher Scoring iterations: 2
```

Get the stats and explain them

The strongest predictive factor appears to be height, based on the high t-value and low p-value. The next strongest predictors are age and gender (male), respectively, which are also strong predictive factors for height. Interestingly, smoking is not a strong predictor of FEV, but it is negatively correlated, based on the negative slope.

By how much does smoke decrease lung capacity?

Smoking decreases FEV by about 0.087. The smoke variable is comprised of just 0 and 1, therefore the negative slope is the change in FEV between non-smoker and smoker.

Logistic Regression

Can we predict whether the car has an automatic or manual transmission?

This is a categorical prediction, a probability that the car has an automatic or manual transmission

Most predictions are either regressions or categorizations, this holds true for ML models as well.

We will try to predict am using mpg, hp, cyl

```
Model_lo=glm(am~mpg+cyl,family=binomial(link='logit'),data=mtcars)
summary(Model_lo)
```

```
##
## Call:
## glm(formula = am ~ mpg + cyl, family = binomial(link = "logit"),
##     data = mtcars)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.3421     5.1310  -1.626   0.1040
## mpg           0.3699     0.2113   1.751   0.0799 .
## cyl6          0.7316     1.4136   0.518   0.6048
## cyl8          0.7016     1.9562   0.359   0.7198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 31  degrees of freedom
## Residual deviance: 29.404  on 28  degrees of freedom
## AIC: 37.404
##
## Number of Fisher Scoring iterations: 5
```

Looking at this set of results, the model does not work, we have no non-zero parameter values

Surviving the titanic, a logistic model

We will try the titanic data set, looking at classifying passengers as surviving or not

```
library("titanic")
data(titanic_train)
head(titanic_train)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age SibSp Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

what have we got

```
str(titanic_train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs T
hayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
titanic_train$Survived=factor(titanic_train$Survived,labels=c("no","yes"))
titanic_train$Pclass=factor(titanic_train$Pclass)
titanic_train$Sex=factor(titanic_train$Sex)
```

Let's predict survival using only Pclass and Sex

We could do more with this, but this is an example

```
Model_titanic=glm(Survived~Pclass+Sex,family=binomial(link='logit'),data=titanic_train)
summary(Model_titanic)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex, family = binomial(link = "logit"),
##     data = titanic_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.2971     0.2190  10.490  < 2e-16 ***
## Pclass2      -0.8380     0.2447  -3.424 0.000618 ***
## Pclass3      -1.9055     0.2141  -8.898  < 2e-16 ***
## Sexmale      -2.6419     0.1841 -14.351  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  826.89  on 887  degrees of freedom
## AIC: 834.89
##
## Number of Fisher Scoring iterations: 4
```

Okay, in this model, all the predictors are meaningful

Let's figure out how accurate the logistic regression is

```
# get the predictions of the probability of survival for each passenger

y_pred=predict.glm(object=Model_titanic,newdata=titanic_train,type="response")

# to binarize the answer into a yes/no prediction, set all values above 0.5 to 1
# and all values below to 0

y_pred=(y_pred>0.5)*1

#now convert to a factor so we can compare

y_pred=factor(y_pred,labels=c("no","yes"))

y_pred[1:10]
```

```
##   1   2   3   4   5   6   7   8   9  10
##  no yes yes yes  no  no  no  no yes yes
## Levels: no yes
```

```
# sum up the number of cases where the prediction is correct
# and divide the the size of the data set to get the rate of correct predictions

sum(y_pred==titanic_train$Survived)/dim(titanic_train)[1]
```

```
## [1] 0.7867565
```

confusion matrix

We can show the results using a confusion matrix

This will show us the nature of the errors, and a number of other statistics, including a confidence interval on the accuracy

```
library("caret")
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
confusionMatrix(y_pred,titanic_train$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  468 109
##        yes  81 233
##
##                Accuracy : 0.7868
##                  95% CI : (0.7584, 0.8132)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5421
##
##  Mcnemar's Test P-Value : 0.05014
##
##             Sensitivity : 0.8525
##             Specificity : 0.6813
##          Pos Pred Value : 0.8111
##          Neg Pred Value : 0.7420
##              Prevalence : 0.6162
##          Detection Rate : 0.5253
##    Detection Prevalence : 0.6476
##       Balanced Accuracy : 0.7669
##
##        'Positive' Class : no
##
```

Question/Action

We will look at predicting loan defaults using data from the ISLR package

```
library("ISLR")
(default_tib <- as_tibble(ISLR::Default))
```

```
## # A tibble: 10,000 × 4
##    default student balance income
##    <fct>   <fct>     <dbl>  <dbl>
##  1 No      No         730. 44362.
##  2 No      Yes        817. 12106.
##  3 No      No        1074. 31767.
##  4 No      No         529. 35704.
##  5 No      No         786. 38463.
##  6 No      Yes        920.  7492.
##  7 No      No         826. 24905.
##  8 No      Yes        809. 17600.
##  9 No      No        1161. 37469.
## 10 No      No           0  29275.
## # ℹ 9,990 more rows
```

```
head(default_tib)
```

```
## # A tibble: 6 × 4
##   default student balance income
##   <fct>   <fct>     <dbl>  <dbl>
## 1 No      No         730. 44362.
## 2 No      Yes        817. 12106.
## 3 No      No        1074. 31767.
## 4 No      No         529. 35704.
## 5 No      No         786. 38463.
## 6 No      Yes        920.  7492.
```

```
default_tib$default=factor(default_tib$default,labels=c("no","yes"))
```

Build a logistic regression model to predict default using student, balance and income

```
default_tib$default = factor(default_tib$default)
model_default = glm(default~student + balance + income, family=binomial(link = "logit"), data=de
fault_tib)
summary(model_default)
```

```
##
## Call:
## glm(formula = default ~ student + balance + income, family = binomial(link = "logit"),
##     data = default_tib)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

Determine the accuracy rate of the logistic regression model, create a confusion matrix

```
# get the predictions of the probability of default

y_pred=predict.glm(object=model_default,newdata=default_tib,type="response", levels=levels(defau
lt_tib$default))

# to binarize the answer into a yes/no prediction, set all values above 0.5 to 1
# and all values below to 0

y_pred=(y_pred>0.5)*1

#now convert to a factor so we can compare

y_pred=factor(y_pred,labels=c("no","yes"))

y_pred[1:10]
```

```
##  1  2  3  4  5  6  7  8  9 10
## no no no no no no no no no no
## Levels: no yes
```

```
# sum up the number of cases where the prediction is correct
# and divide the the size of the data set to get the rate of correct predictions

sum(y_pred==default_tib$default)/dim(default_tib)[1]
```

```
## [1] 0.9732
```

```
confusionMatrix(y_pred,default_tib$default)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  9627  228
##        yes   40  105
##
##                Accuracy : 0.9732
##                  95% CI : (0.9698, 0.9763)
##     No Information Rate : 0.9667
##     P-Value [Acc > NIR] : 0.0001044
##
##                   Kappa : 0.4278
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9959
##             Specificity : 0.3153
##          Pos Pred Value : 0.9769
##          Neg Pred Value : 0.7241
##              Prevalence : 0.9667
##          Detection Rate : 0.9627
##    Detection Prevalence : 0.9855
##       Balanced Accuracy : 0.6556
##
##        'Positive' Class : no
##
```