ML Basic

JEFFREY CHAN

12/23/2020

Reference website

http://www.sthda.com/english/articles/40-regression-analysis/165-linear-regression-essentials-in-r/#multiple-linear-regression

Linear Regression Essentials in R

load / install the requirement packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

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

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

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

```
theme_set(theme_bw())
```

Preparing the data

sample_n(data, col)

```
data("marketing", package = "datarium")
sample_n(marketing,3)
```

```
##   youtube facebook newspaper sales
## 1  107.64    11.88     42.84 12.72
## 2    6.48    35.88     11.28  6.36
## 3  221.88    25.20     26.40 18.60
```

```
# split the data into training and test set
set.seed(123)
```

```r
# We'll randomly split the data into training set (80% for building a
# predictive model) and test set (20% for evaluating the model).

training.samples <- marketing$sales %>%
  createDataPartition(p=0.8, list = F)
train.data <- marketing[training.samples, ]
test.data <- marketing[-training.samples, ]

# build model
model <- lm(sales ~., data = train.data)

# summarize the model
summary(model)
```

```
##
## Call:
## lm(formula = sales ~ ., data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7142  -0.9939   0.3684   1.4494   3.3619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.594142   0.420815   8.541 1.05e-14 ***
## youtube     0.044636   0.001552  28.758  < 2e-16 ***
## facebook    0.188823   0.009529  19.816  < 2e-16 ***
## newspaper   0.002840   0.006442   0.441     0.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.043 on 158 degrees of freedom
## Multiple R-squared:  0.8955, Adjusted R-squared:  0.8935
## F-statistic: 451.2 on 3 and 158 DF,  p-value: < 2.2e-16
```

```r
summary(model)$coefficient
```

```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 3.594141778 0.420814685  8.5409134 1.054115e-14
## youtube     0.044635905 0.001552128 28.7578721 1.125356e-64
## facebook    0.188823227 0.009528828 19.8159966 1.090250e-44
## newspaper   0.002839757 0.006441963  0.4408217 6.599447e-01
```

```r
# make prediction
predictions <- model %>% predict(test.data)

# model performance
# (a) Prediction error, RMSE
RMSE(predictions, test.data$sales)
```

```
## [1] 1.965508
```

```r
# (b) R-square
R2(predictions, test.data$sales)
```

```
## [1] 0.9049049
```

Simple Linear Regression

```
model_you <- lm(sales ~ youtube, data = train.data)
summary(model_you)$coef
```

```
##               Estimate  Std. Error  t value     Pr(>|t|)
## (Intercept) 8.58914961 0.616044182 13.94242 1.987874e-29
## youtube     0.04671639 0.003003398 15.55451 8.019035e-34
```

Make prediction

```
newdata <- data.frame(youtube = c(0,1000))
model_you %>% predict(newdata)
```

```
##        1        2
##  8.58915 55.30554
```

Multiple Linear Regression

```
model3 <- lm(sales ~ youtube + facebook + newspaper, data = train.data)
summary(model3)$coef
```

```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 3.594141778 0.420814685  8.5409134 1.054115e-14
## youtube     0.044635905 0.001552128 28.7578721 1.125356e-64
## facebook    0.188823227 0.009528828 19.8159966 1.090250e-44
## newspaper   0.002839757 0.006441963  0.4408217 6.599447e-01
```

Quick note, i have a lot of predictor, we can simply use ~. to include all the predictor

```
model3_1 <- lm(sales ~., data = train.data)
summary(model3_1)$coef
```

```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 3.594141778 0.420814685  8.5409134 1.054115e-14
## youtube     0.044635905 0.001552128 28.7578721 1.125356e-64
## facebook    0.188823227 0.009528828 19.8159966 1.090250e-44
## newspaper   0.002839757 0.006441963  0.4408217 6.599447e-01
```

Col1: b0 / y intercept is 3.73546064 Col2: std.error = 0.44062. this represent the accuracy of the coefficients. we always want small value for std.error Col3: T value is the t statistics estimate / std error = t value Col4: P value for the T statistics. The smaller the p value the more significant the estimate is.

Let's make prediction for the values

```
# New advertising budget
newdata <- data.frame( youtube = 2000, facebook = 1000, newspaper = 1000)

# predict sales values
model3_1 %>% predict(newdata)
```

```
##        1
## 284.5289
```

Interpretation.

Before using the model for predictions, i need to access the statsitical significance of the model. We need to apply summary(model_name)

```
summary(model)
```

```
##
```

```
## Call:
## lm(formula = sales ~ ., data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7142  -0.9939   0.3684   1.4494   3.3619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.594142   0.420815   8.541 1.05e-14 ***
## youtube     0.044636   0.001552  28.758  < 2e-16 ***
## facebook    0.188823   0.009529  19.816  < 2e-16 ***
## newspaper   0.002840   0.006442   0.441     0.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.043 on 158 degrees of freedom
## Multiple R-squared:  0.8955, Adjusted R-squared:  0.8935
## F-statistic: 451.2 on 3 and 158 DF,  p-value: < 2.2e-16
```

The residuals, should be normally distributed. Looks like our data is kind of normally distributed. by theory, mean should be zero. Q1 and Q3, min and max should be around the same with + or - sign.

coefficients, shows the paramete values and their significance. If they are significant, it will be shown with starts

RSE / R^2 / F statistics are used to check how well the model fits to our data

First step, always check the F statistics and the associated p value and the bottom of the model summary

Coefficients significance

```
summary(model)$coef
```

```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 3.594141778 0.420814685  8.5409134 1.054115e-14
## youtube     0.044635905 0.001552128 28.7578721 1.125356e-64
## facebook    0.188823227 0.009528828 19.8159966 1.090250e-44
## newspaper   0.002839757 0.006441963  0.4408217 6.599447e-01
```

Important, from the summary table , we can see that youtube, facebook advertising budget are significantly changing the sales while by newpaper, there is not much of a change.

Also, to interpret the intercept, we can say every 1000 dollar i invest in facebook advertising i will have a return of 1000* 0.19398450 = 193 sale unit. So do youtube, 1000* 0.04516611 = 45.16 sale units

Since newspaper does not affect the outcome much, lets remove it from the model

```
model4 <- lm(sales ~ youtube + facebook, data = train.data)
summary(model4)
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook, data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8127  -1.0073   0.3236   1.4643   3.3454
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.658580   0.393609   9.295   <2e-16 ***
## youtube     0.044650   0.001548  28.846   <2e-16 ***
## facebook    0.190165   0.009006  21.114   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.038 on 159 degrees of freedom
## Multiple R-squared:  0.8954, Adjusted R-squared:  0.894
## F-statistic: 680.2 on 2 and 159 DF,  p-value: < 2.2e-16
```

now our equation can be written as sales = 3.577663 + 0.045287(youtube) + 0.190299(facebook)

Model accuracy

Well, after knowing it is significant, we would like to know how well the model fits our data. This process will be referred to as the goodness of fit The overall quality of the linear regression fit can be shown by model summary RSE, R^2, adjusted R^2, F statistics

RSE / model sigma -> prediction error. it is the observed out come - predicted value (Y_i - Y bar). small rse is the best the model fits to the data. dividing the RSE by the average value of the outcome variable will give me the prediction error rate. Which should be as small as possible.

in our example, we have 1.853 for RSE.

```
(1.853/(mean(train.data$sales)))*100
```

```
## [1] 10.99196
```

It is 10.90% which is low.

R-squared and adjusted R squared R^2 is ranges between 0 and 1. Higher the R square the better the model. High R square = observed data is very close to the prediction data. So the quality of the regression line is pretty good.R^2 = pearson ^2. Here is the trick tho, if i have a lot of parameters, R^2 will increase along with it.

However, the adjusted R^2 is the correction of the for the number of parameters in the predictive model.Therefore, I should always read adjusted R^2 because it will make the correction according the to incorrect R^2 When adjusted R^2 = 0 thats mean the mdoel did not explain much about the variability in the outcome. In our outcome, adjusted R^2 is 0.9112 so it is pretty good.

Lastly, F statistics gives the overall significance of the model. it tells us whether at least one predictor variables has non zero coefficient. In a simple linear regression (one parameter), it wont be interesting because it is just a duplicated info given by the t test from the coef table.

The F statistic becomes more important once we start using multiple predicators as in multiple linear regression.

So according to what we have, our F statistics equal 825.4 with 2 parameters and 159 df, with a p-value of 2.2e^-16. which is highly significant. In order to read the p value, $p < 0.05$ will be significant.

Making predictions

Procedure to make predictions 1) predict the sales values based on new advertising budgets in the test data 2) Assess the model performance by computing: the prediction error RMSE (Root Mean Squared Error), representing the average difference between the observed known outcome values in the test data and the preidcted outcome values by the model. The lower the RMSE, the better the model. The R^2, representing the correlation between the observed outcome values and the predicted outcome values. the higher the r^2, the better the model.

```r
# make predictions
predictions <- model %>% predict(test.data)

# model performance
# (a) compute the prediction error, RMSE
RMSE(predictions, test.data$sales)
```

```
## [1] 1.965508
```

```r
R2(predictions, test.data$sales)
```

```
## [1] 0.9049049
```

```r
(RMSE(predictions, test.data$sales)/ mean(test.data$sales))*100
```

```
## [1] 11.77248
```

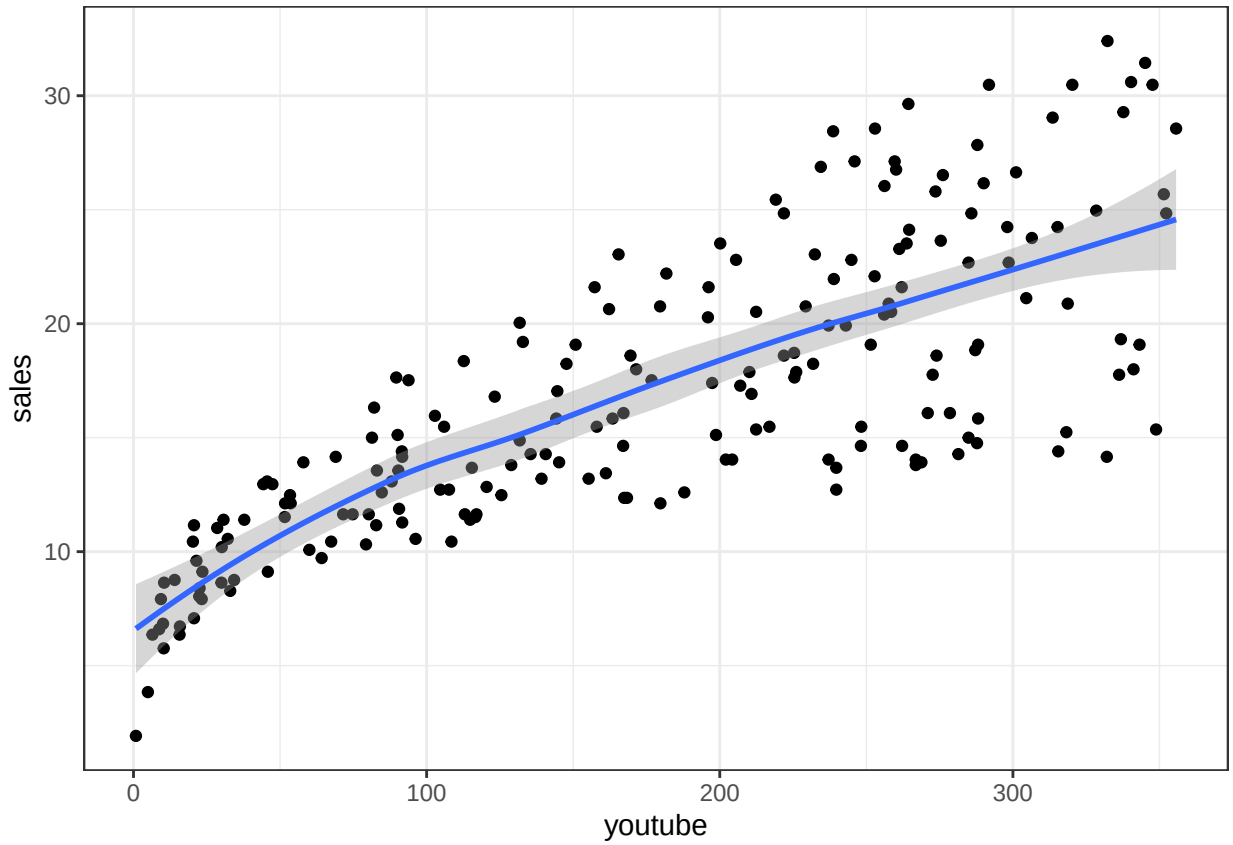
The % error is 16.39% is alright

Discussion

This section discuss basic linear regression and provides practical examples in R for computing simple and multiple linear regression model. Also, we have learnt how the accuracy of the model. The idea of linear regression is to see the predictor relationship with the response. It can be easily be visulaized by a basic plot without working a lot on lm() summary and more.

```r
ggplot(marketing, aes(youtube, sales)) +
  geom_point() +
  stat_smooth()
```

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



As we can expected, since we know that the Beta_1 is positive from what we have done before, we can expect that the slope is positive and the scatter dots are following the positve slope.

http://www.sthda.com/english/articles/40-regression-analysis/164-interaction-effect-in-multiple-regression-essentials/

Interaction Effect in Multiple Regression: Essentials

Interaction effects

```
# build the model
# use this:
model12 <- lm(sales ~ youtube*facebook, data = train.data)
summary(model12)
```

```
##
## Call:
## lm(formula = sales ~ youtube * facebook, data = train.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6325 -0.5051  0.2666  0.7425  1.8109
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.179e+00  3.306e-01  24.741  < 2e-16 ***
## youtube          1.859e-02  1.660e-03  11.196  < 2e-16 ***
## facebook         2.781e-02  1.016e-02   2.739  0.00687 **
## youtube:facebook 9.145e-04  4.952e-05  18.467  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.15 on 158 degrees of freedom
## Multiple R-squared:  0.9669, Adjusted R-squared:  0.9662
## F-statistic:  1537 on 3 and 158 DF,  p-value: < 2.2e-16
```

Make predictions

```
predictions12 <- model12 %>% predict(test.data)
# model performance
# (a) prediction error, RMSE
RMSE(predictions12, test.data$sales)
```

```
## [1] 1.055644
```

```
# (b) R-square
R2(predictions12, test.data$sales)
```

```
## [1] 0.9716356
```

```
# % error
(RMSE(predictions12, test.data$sales) / mean(test.data$sales))*100
```

```
## [1] 6.322813
```

Those values are pretty good. high R^2 and 10.67% error

```
summary(model12)$coef
```

```
##                     Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)      8.1786320203 3.305698e-01 24.741018 3.213476e-56
## youtube          0.0185856131 1.659981e-03 11.196278 8.614530e-22
## facebook         0.0278145861 1.015569e-02  2.738818 6.874607e-03
## youtube:facebook 0.0009145105 4.952241e-05 18.466598 2.668477e-41
```

As you can see that all of them are significant (Pr(>|t|)). so it means there is an interaction relationship between the two predictor variables(youtube and facebook advertising)

our model will be like sales = 7.89 + 0.0189949052(youtube) + 0.0330506939(facebook) + 0.0008751096 (youtube*facebook)

comparing the additive and the interaction models.

The prediction error RMESE of the interaction model is 1.721177 compared with the prediction error of the addictive model 2.642146, interaction model is lower.

R^2 of the interaction model is 0.9373706, and for the addictive model has 0.8322611. Interaction model has a better R^2

Lastly, these result suggest that the model with the int eraction term is better than the model that contains only main effects. So for this specific data, we should go for the model with the interactio model.

Discussion, after finding addictive model which is significant, we should also check if the interaction model is also significant. If they do, we should adapt the interaction model.

http://www.sthda.com/english/articles/40-regression-analysis/163-regression-with-categorical-variables-dummy-coding-essentials-in-r/

Regression with Categorical Variables: Dummy Coding Essentials in R

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

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

```
# load data
data("Salaries", package = "car")
```

```
## Warning in data("Salaries", package = "car"): data set 'Salaries' not found
```

```
# Inspect the data
sample_n(Salaries, 3)
```

```
##       rank discipline yrs.since.phd yrs.service    sex salary
## 1     Prof          A            22          15   Male 166800
## 2     Prof          A            39          36 Female 137000
## 3 AsstProf          A            11           4   Male  78785
```

Categorical variables with two levels From our data set, we would like to investigate differences i nsalaries between males and females. Based on the gender, we can say m = 1, female = 0 b0 + b1 if person is male bo if person is female. B0 is average salary among females B0 + B1 = average salary among males B1 is average difference in salary between males and females

```
mwage <- lm(salary ~ sex, data = Salaries)
summary(mwage)$coef
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 101002.41   4809.386 21.001103 2.683482e-66
## sexMale      14088.01   5064.579  2.781674 5.667107e-03
```

From the above result, average salary for female is 101002.41. Male = B0 + B1 = 101002.41 + 14088.01 = 115090.40 the P value for both sex is very significant, which is suggesting that there is a statistical evidence of a difference in average salary between the genders.

```
contrasts(Salaries$sex)
```

```
##        Male
## Female    0
## Male      1
```

I can use the function relevel() to set the baseline category to males as follow

```
Salaries <- Salaries %>%
  mutate(sex = relevel(sex, ref = "Male"))
```

```
mwage1 <- lm(salary ~ sex , data = Salaries)
summary(mwage1)$coef
```

```
##              Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 115090.42   1587.378 72.503463 2.459122e-230
## sexFemale   -14088.01   5064.579 -2.781674  5.667107e-03
```

Categorical varaibles with more than two levels

Generally, categorical variable with n levels will be tranformed in to n-1 variables each with two levels. These n-1 new varaibles contain the same info than the single variable. This recording creates a table called contrast matrix.

In salaries, data has three levels, asstprof, assocprof, and prof. These variables could be dummy coded in to two variables, one called assocprof and one prof.

if rank = assocprof, then the column assocprof would be coded with 1 and prof with 0 if rank = prof, then the col assocprof would be coded with a 0 and prof would be coded with a 1 if rank = asstprof then both cols assocprof and prof would be coded with a 0

```
res <- model.matrix(~rank, data=Salaries)
head(res[,-1])
```

```
##   rankAssocProf rankProf
## 1             0        1
## 2             0        1
## 3             0        0
## 4             0        1
## 5             0        1
## 6             1        0
```

```
head(res)
```

```
##   (Intercept) rankAssocProf rankProf
## 1           1             0        1
## 2           1             0        1
## 3           1             0        0
## 4           1             0        1
## 5           1             0        1
## 6           1             1        0
```

R will always use the first level as a reference and interpret the remaining levels relative to the first level

The following code will give you the level

```
levels(Salaries$rank)
```

```
## [1] "AsstProf"  "AssocProf" "Prof"
```

Indeed, AsstProf will be the reference level.

Also ANOVA(analyse of variance) is just a special case of linear model where the predictors are categorical variables. R understands the fact that ANOVA and regression are both examples of linear models, it lets you extract the classic ANOVA table from the regressio nmodel using the R base anova() function or the Anova() function. We generally use Anova() function because it automatically takes care of unbalanced designs.

Lets predict the salary from using a multiple regression procedure

```
mwage2 <- lm(salary ~ yrs.service + rank + discipline + sex, data = Salaries)
Anova(mwage2)
```

```
## Anova Table (Type II tests)
##
## Response: salary
##                 Sum Sq  Df  F value    Pr(>F)
## yrs.service 3.2448e+08   1   0.6324    0.4270
## rank        1.0288e+11   2 100.2572 < 2.2e-16 ***
## discipline  1.7373e+10   1  33.8582 1.235e-08 ***
## sex         7.7669e+08   1   1.5137    0.2193
## Residuals   2.0062e+11 391
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(mwage1)
```

```
## Anova Table (Type II tests)
##
## Response: salary
##               Sum Sq  Df F value   Pr(>F)
## sex       6.9800e+09   1  7.7377 0.005667 **
## Residuals 3.5632e+11 395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When we take rank, discipline, yrs of service in to consideration, the variable sex is no longer significant combined with the variation in salary between individuals.

```
summary(mwage2)
```

```
##
## Call:
## lm(formula = salary ~ yrs.service + rank + discipline + sex,
##     data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -64202 -14255  -1533  10571  99163
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   73122.92    3245.27  22.532  < 2e-16 ***
## yrs.service     -88.78     111.64  -0.795 0.426958
## rankAssocProf 14560.40    4098.32   3.553 0.000428 ***
## rankProf      49159.64    3834.49  12.820  < 2e-16 ***
## disciplineB   13473.38    2315.50   5.819 1.24e-08 ***
## sexFemale     -4771.25    3878.00  -1.230 0.219311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22650 on 391 degrees of freedom
## Multiple R-squared:  0.4478, Adjusted R-squared:  0.4407
## F-statistic: 63.41 on 5 and 391 DF,  p-value: < 2.2e-16
```

```
levels(Salaries$discipline)
```

```
## [1] "A" "B"
```

apply ??salaries and check discipline a and b is corresponding to what field from the documentation, dis A is theoretical, dis B is applied department.

For example, from dicipline B (applied departments) is significantly associated with an average increase of 13473.38 in salary compared (there is a difference) to discipline theoretical departments. So that is the reason why discipline B is significant

Nonlinear Regression Essentials in R: Polynomial and Spline Regression Models

http://www.sthda.com/english/articles/40-regression-analysis/162-nonlinear-regression-essentials-in-r-polynomial-and-spline-regression-models/

In this section, you'll learn how to compute non-linear regression models and how to compare the different models in order to choose the one that fits the best your data.

Will also be using RMSE and R2 metric to compare the different models' accuracy Recall, RMSE -> model prediction error. Thats the average difference of the observed outcome values and predicted outcome values.

R2 represents the squared correlation. How accurate is the data, if it is exactly on the regression line, that the R^2 will be 1

The best model is the model with the lowest RMSE and the highest R2

```
library(tidyverse)
library(caret)
theme_set(theme_classic())
```

Prepare the data

we will use the boston data from MASS package

```
# Load the data, if the package is not installed, instal it now and load the library
if(!require("MASS")){
  install.packages("MASS")
  library(MASS)
}
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
data("Boston", package = "MASS")
str(Boston)
```

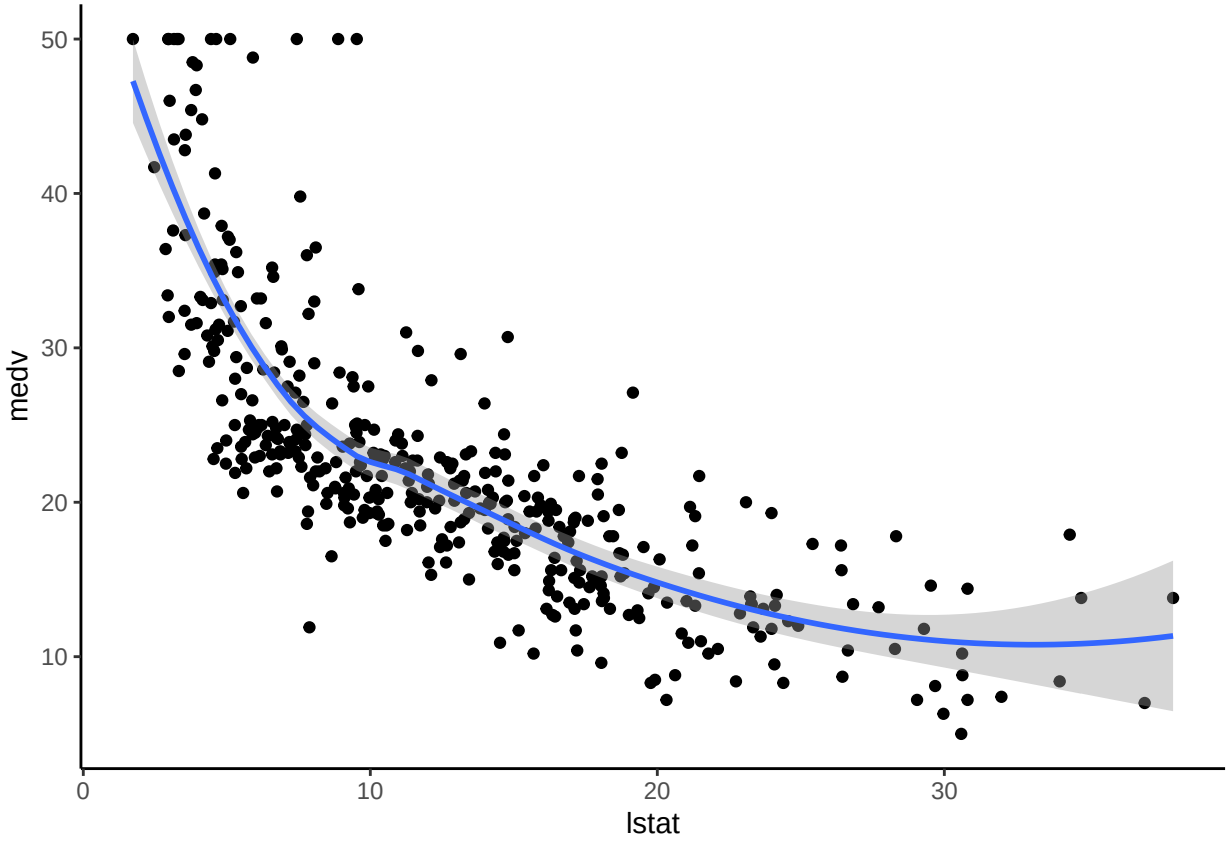
```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
# Split the data in to training and test set
set.seed(123)
ts <- Boston$medv %>%
  createDataPartition(p =0.8, list = F)
trd <- Boston[ts, ]
ted <- Boston[-ts, ]
```

Visualize the scatter plot of the medv vs lstat varaibles, both medv and lstat is from the Boston dataset use stat_smooth when i want to display the results with non standard geom

```
ggplot(trd, aes(lstat,medv)) +
  geom_point() +
  stat_smooth()
```

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



It is obvious to see that the blue line is a curve

Linear Regression

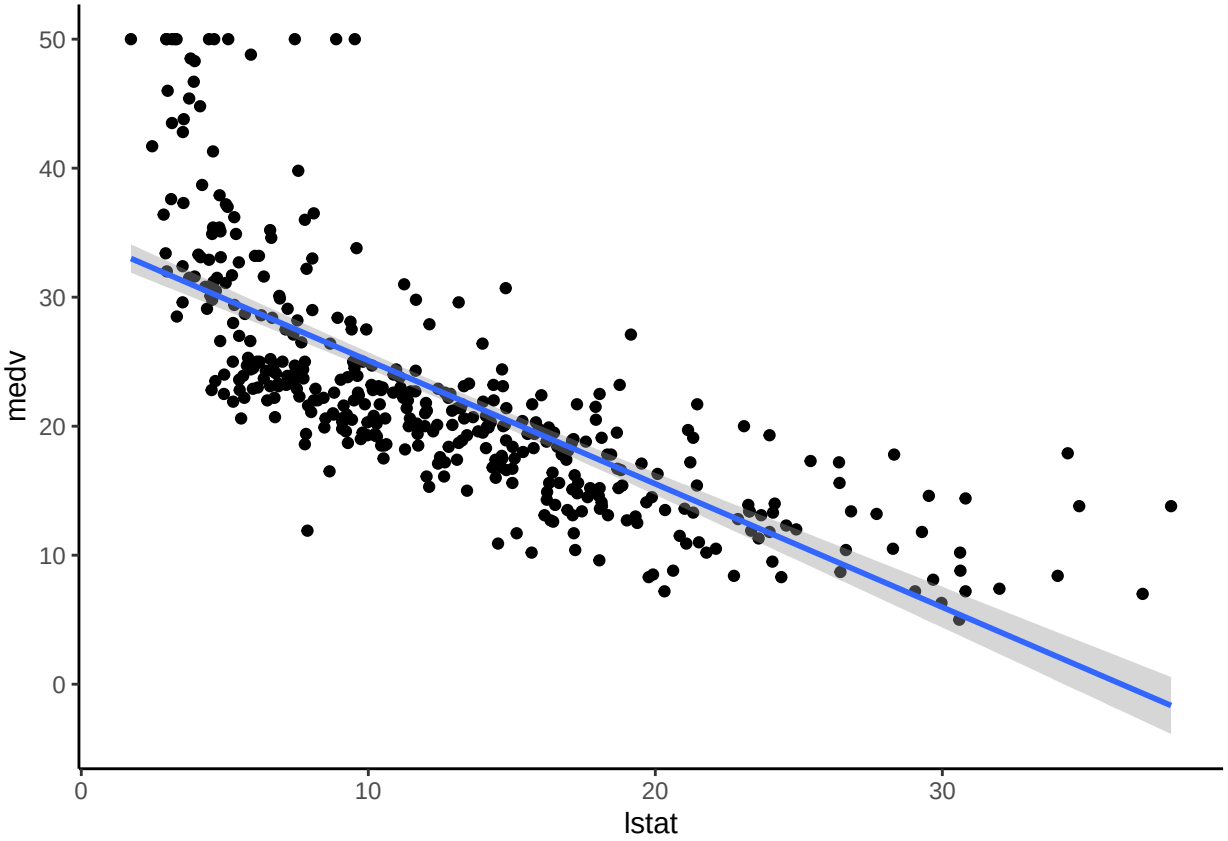
```
# Build the model
modelc <- lm(medv ~ lstat, data = trd)
# make prediction
pmodelc <- modelc %>% predict(ted)
data.frame(
  rmseL <- RMSE(pmodelc, ted$medv),
  r2L <- R2(pmodelc, ted$medv),
  perrorL <- rmseL / mean(ted$medv)
)
```

```
##   rmseL....RMSE.pmodelc..ted.medv. r2L....R2.pmodelc..ted.medv.
## 1                         6.503817                     0.513163
##   perrorL....rmseL.mean.ted.medv.
## 1                       0.2874712
```

I have a pretty high percent error 26.82%

visualize the data

```
ggplot(trd, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ x)
```



Polynomial Regression

Lets make it polynomial regression medv is the response.

we should have this following formula medv = b0 + b1 * lstat + b2*lstat^2 in r to make that ^2 we need to apply I(x^2)

```
modelc1 <- lm(medv ~ lstat + I(lstat^2), data = trd)
summary(modelc1)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = trd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.3654  -3.8250  -0.6439   2.2733  25.2922
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.573638   0.964887   44.12   <2e-16 ***
## lstat       -2.267309   0.135846  -16.69   <2e-16 ***
## I(lstat^2)   0.041198   0.004095   10.06   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.501 on 404 degrees of freedom
## Multiple R-squared:  0.6418, Adjusted R-squared:   0.64
## F-statistic: 361.9 on 2 and 404 DF,  p-value: < 2.2e-16
```

Alternative way to create 2 degree regression model

```
modelc2<- lm(medv ~ poly(lstat, 2, raw =T), data = trd)
summary(modelc2)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 2, raw = T), data = trd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.3654  -3.8250  -0.6439   2.2733  25.2922
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                42.573638   0.964887   44.12   <2e-16 ***
## poly(lstat, 2, raw = T)1 -2.267309   0.135846  -16.69   <2e-16 ***
## poly(lstat, 2, raw = T)2  0.041198   0.004095   10.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.501 on 404 degrees of freedom
## Multiple R-squared:  0.6418, Adjusted R-squared:   0.64
## F-statistic: 361.9 on 2 and 404 DF,  p-value: < 2.2e-16
```

```
modelc2p <- modelc2 %>% predict(ted)
p2 <- c(RMSE(modelc2p,ted$medv), R2(modelc2p, ted$medv), RMSE(modelc2p, ted$medv) / mean(ted$medv))
```

As you can see they are the same. intercepts and the coefficient of beta1 and beta2^2 are all significant. as well as the F statistic P value is also small. Indeed, we have an Adjusted R^2 0.6329 which is ok.

The following is the 6th order

```
modelc6 <- lm(medv ~ poly(lstat, 6, raw = T), data = trd)
summary(modelc6)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 6, raw = T), data = trd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1962  -3.1527  -0.7655   2.0404  26.7661
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 7.788e+01  6.844e+00  11.379  < 2e-16 ***
## poly(lstat, 6, raw = T)1 -1.767e+01  3.569e+00  -4.952 1.08e-06 ***
## poly(lstat, 6, raw = T)2  2.417e+00  6.779e-01   3.566 0.000407 ***
## poly(lstat, 6, raw = T)3 -1.761e-01  6.105e-02  -2.885 0.004121 **
## poly(lstat, 6, raw = T)4  6.845e-03  2.799e-03   2.446 0.014883 *
```

```
## poly(lstat, 6, raw = T)5 -1.343e-04  6.290e-05  -2.136 0.033323 *
## poly(lstat, 6, raw = T)6  1.047e-06  5.481e-07   1.910 0.056910 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.188 on 400 degrees of freedom
## Multiple R-squared:  0.6845, Adjusted R-squared:  0.6798
## F-statistic: 144.6 on 6 and 400 DF,  p-value: < 2.2e-16
```

```
modelc6p <- modelc6 %>% predict(ted)
p6 <- c(RMSE(modelc6p,ted$medv), R2(modelc6p, ted$medv), RMSE(modelc6p, ted$medv) / mean(ted$medv))
```

From this point we can see that after degree 3 it is no longer significant. Then we will just simply use degree 3 for our model

```
modelc5 <- lm(medv ~ poly(lstat, 5, raw = T), data = trd)
summary(modelc5)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 5, raw = T), data = trd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1519  -3.1235  -0.5927   2.0962  27.1286
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.765e+01  4.273e+00  15.831  < 2e-16 ***
## poly(lstat, 5, raw = T)1 -1.177e+01  1.786e+00  -6.588 1.40e-10 ***
## poly(lstat, 5, raw = T)2  1.220e+00  2.580e-01   4.727 3.16e-06 ***
## poly(lstat, 5, raw = T)3 -6.385e-02  1.644e-02  -3.884 0.000120 ***
## poly(lstat, 5, raw = T)4  1.577e-03  4.714e-04   3.345 0.000901 ***
## poly(lstat, 5, raw = T)5 -1.459e-05  4.954e-06  -2.945 0.003421 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.205 on 401 degrees of freedom
## Multiple R-squared:  0.6816, Adjusted R-squared:  0.6777
## F-statistic: 171.7 on 5 and 401 DF,  p-value: < 2.2e-16
```

```
modelc5p <- modelc5 %>% predict(ted)
p5 <- c(RMSE(modelc5p,ted$medv), R2(modelc5p, ted$medv), RMSE(modelc5p, ted$medv) / mean(ted$medv))
modelc3 <- lm(medv ~ poly(lstat, 3, raw = T), data = trd)
modelc3p <- modelc3 %>% predict(ted)
p3 <- c(RMSE(modelc3p,ted$medv), R2(modelc3p, ted$medv), RMSE(modelc3p, ted$medv) / mean(ted$medv))
modelc4 <- lm(medv ~ poly(lstat, 4, raw = T), data = trd)
modelc4p <- modelc4 %>% predict(ted)
p4 <- c(RMSE(modelc4p,ted$medv), R2(modelc4p, ted$medv), RMSE(modelc4p, ted$medv) / mean(ted$medv))
perrorp4 <- RMSE(modelc4p, ted$medv) / mean(ted$medv)
modelc7 <- lm(medv ~ poly(lstat, 7, raw = T), data = trd)
modelc7p <- modelc7 %>% predict(ted)
p7 <- c(RMSE(modelc7p,ted$medv), R2(modelc7p, ted$medv), RMSE(modelc7p, ted$medv) / mean(ted$medv))
```

Lets see the RMSE, R2 and percent error p2, p3,p4, p5, p6

```
temp <- cbind(p2,p3,p4,p5,p6,p7)
row.names(temp) <-c("RMSE", "R2", "% error")
temp
```

```
##                p2        p3        p4        p5        p6        p7
## RMSE    5.6307274 5.5007142 5.3929523 5.2703735 5.3495124 5.3569154
## R2      0.6351934 0.6521415 0.6664867 0.6829474 0.6759031 0.6753441
## % error 0.2488803 0.2431336 0.2383705 0.2329525 0.2364505 0.2367777
```

There are couple patterns are going on. First im looking at the table above, we can see that p5 is the dip of the % error. after that it bounces back and has a small changes at p7. Also, if we look at the anova(p1 : p7) the significance level stops at 4.

```
anova(modelc2,modelc3,modelc4,modelc5, modelc6, modelc7)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ poly(lstat, 2, raw = T)
## Model 2: medv ~ poly(lstat, 3, raw = T)
## Model 3: medv ~ poly(lstat, 4, raw = T)
## Model 4: medv ~ poly(lstat, 5, raw = T)
## Model 5: medv ~ poly(lstat, 6, raw = T)
## Model 6: medv ~ poly(lstat, 7, raw = T)
##   Res.Df   RSS Df Sum of Sq       F    Pr(>F)
## 1    404 12224
## 2    403 11624  1    599.27 22.2105 3.381e-06 ***
## 3    402 11100  1    524.61 19.4432 1.334e-05 ***
## 4    401 10865  1    234.93  8.7072  0.003356 **
## 5    400 10767  1     98.14  3.6375  0.057210 .
## 6    399 10766  1      1.00  0.0372  0.847094
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So i would say degree 4 is the best option.

```
modelc4 <- lm(medv ~ poly(lstat, 4, raw = T), data = trd)
modelc4p <- modelc4 %>% predict(ted)
p4 <- c(RMSE(modelc4p,ted$medv), R2(modelc4p, ted$medv), RMSE(modelc4p, ted$medv) / mean(ted$medv))
p4
```

```
## [1] 5.3929523 0.6664867 0.2383705
```

```
summary(modelc4)
```

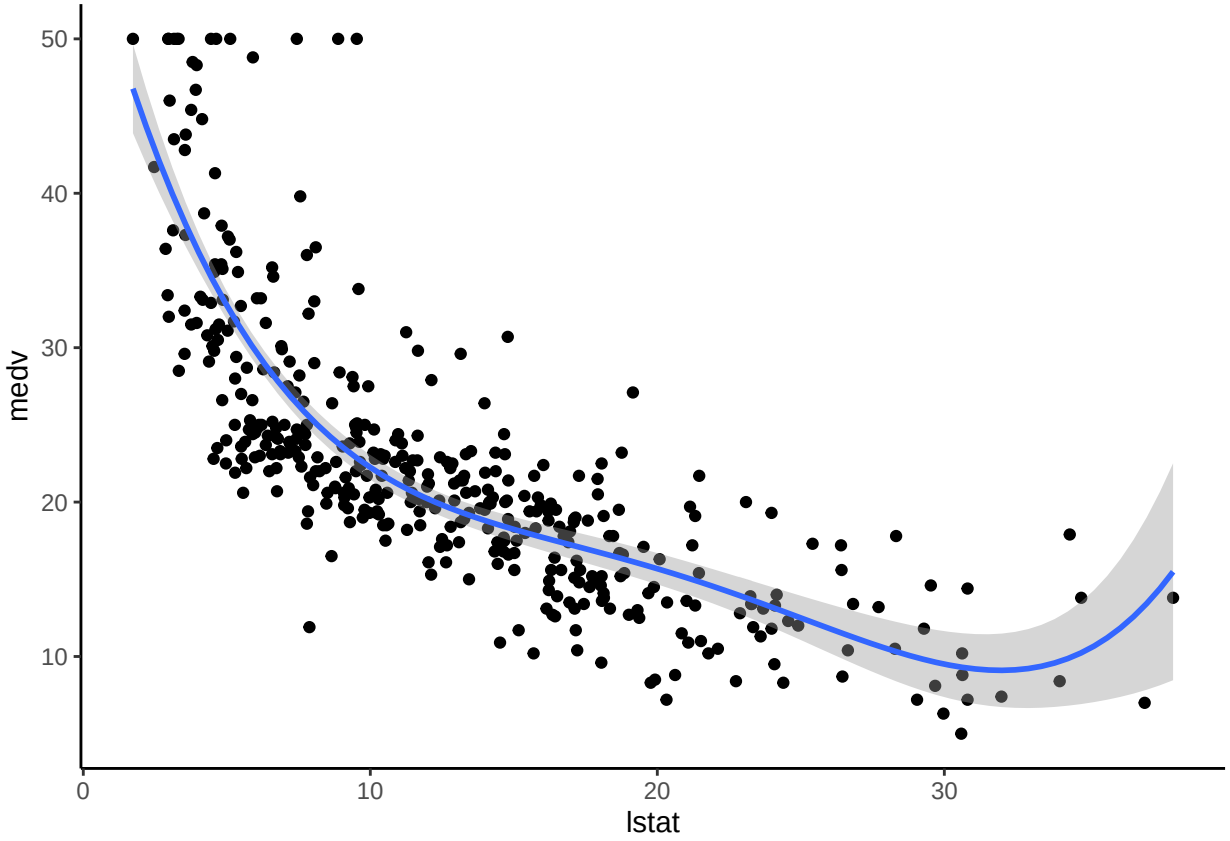
```
##
## Call:
## lm(formula = medv ~ poly(lstat, 4, raw = T), data = trd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6093  -3.1719  -0.6806   2.2075  27.1453
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.764e+01  2.615e+00  22.047  < 2e-16 ***
## poly(lstat, 4, raw = T)1 -7.098e+00  8.303e-01  -8.549 2.62e-16 ***
## poly(lstat, 4, raw = T)2  5.007e-01  8.419e-02   5.947 5.94e-09 ***
```

```
## poly(lstat, 4, raw = T)3 -1.643e-02  3.336e-03  -4.925 1.24e-06 ***
## poly(lstat, 4, raw = T)4  1.948e-04  4.468e-05   4.359 1.66e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.255 on 402 degrees of freedom
## Multiple R-squared:  0.6747, Adjusted R-squared:  0.6715
## F-statistic: 208.5 on 4 and 402 DF,  p-value: < 2.2e-16
```

The resudials distribution is alright. By theory, the median should be 0 and 1Q, 3Q, and min max should be balanced. so this is not the best model but thats all we have.

Let's visualize the data

```
ggplot(trd, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ poly(x, 4, raw = T))
```



Log Transformation When i have a non linear relationship, i can also try a log transformation of the predictor variables.

```
# build the model
modelcLog <- lm(medv ~ log(lstat), data = trd)

# make predictions
pmodelcLog <- modelcLog %>% predict(ted)

# model performance
(rmseLog <- RMSE(pmodelcLog, ted$medv))
```

```
## [1] 5.467124
```

```
(r2Log <-  R2(pmodelcLog, ted$medv))
```

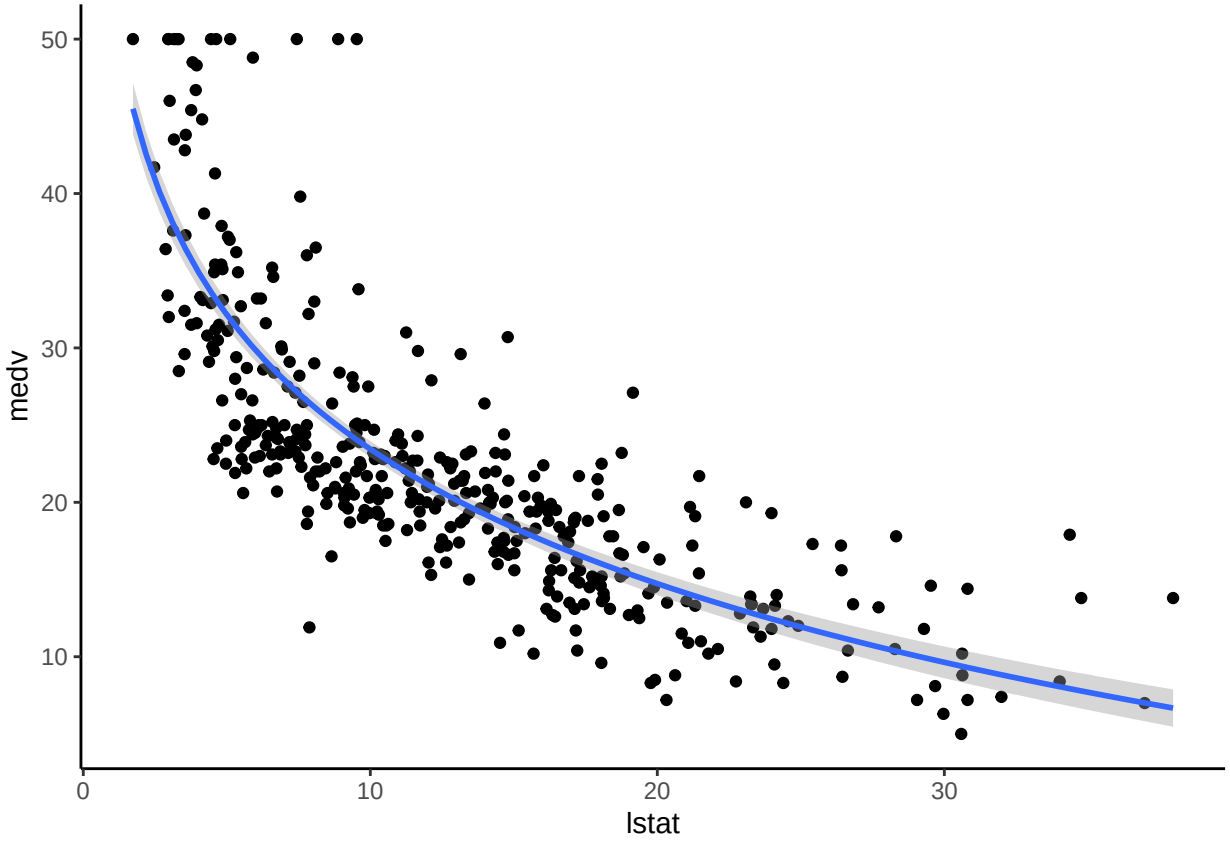
```
## [1] 0.6570091
```

```
(perrorLog <- rmseLog/ mean(ted$medv))
```

```
## [1] 0.2416489
```

visualize the data

```
ggplot(trd, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ log(x))
```



Spline regression

```
library(splines)
# splines becomes a base package so you shouldnt be install
# it if you are using R version 4.0.2
knots <- quantile(trd$lstat, p = c(.25,.5,.75))
# build model
modelcSp <- lm(medv ~ bs(lstat, knots = knots), data = trd)
# make prediction
pmodelcSp <- modelcSp %>% predict(ted)

# model performance
(rmseSp <- RMSE(pmodelcSp, ted$medv))
```

```
## [1] 5.317372
```

```
(r2Sp <-  R2(pmodelcSp, ted$medv))
```

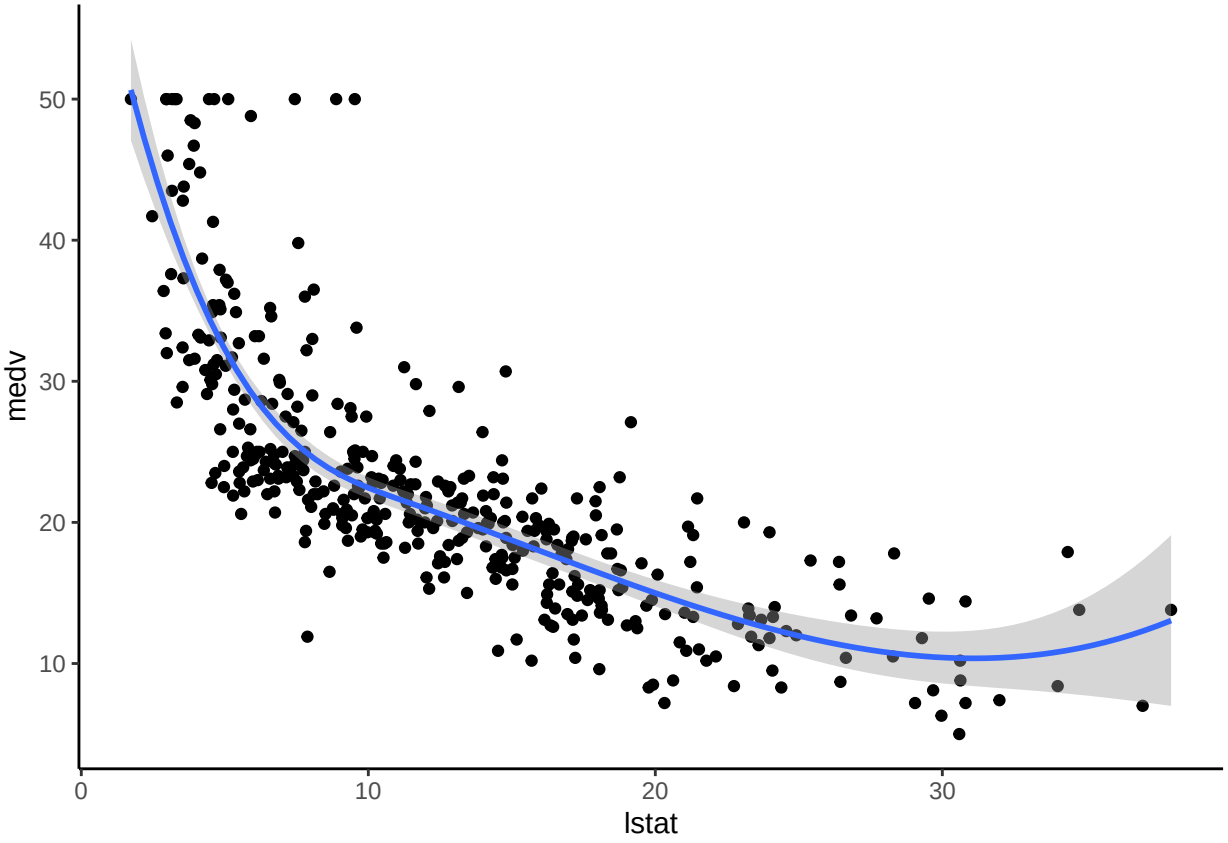
```
## [1] 0.6786367
```

```
(perrorSp <- rmseSp/ mean(ted$medv))
```

```
## [1] 0.2350299
```

Lets visualize the data

```
ggplot(trd, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ splines::bs(x, df =4))
```



Generalized additive models

Here is another problem. Once I know that the model is non linear relationship in my data, the polynomial terms might not be flixible enough to capture the relationship, and the spline terms require speccifing the knots. Therefore, we have Generalized additive models, GAM, are a technique to automatically fit a spline regression. We need mgcv package

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

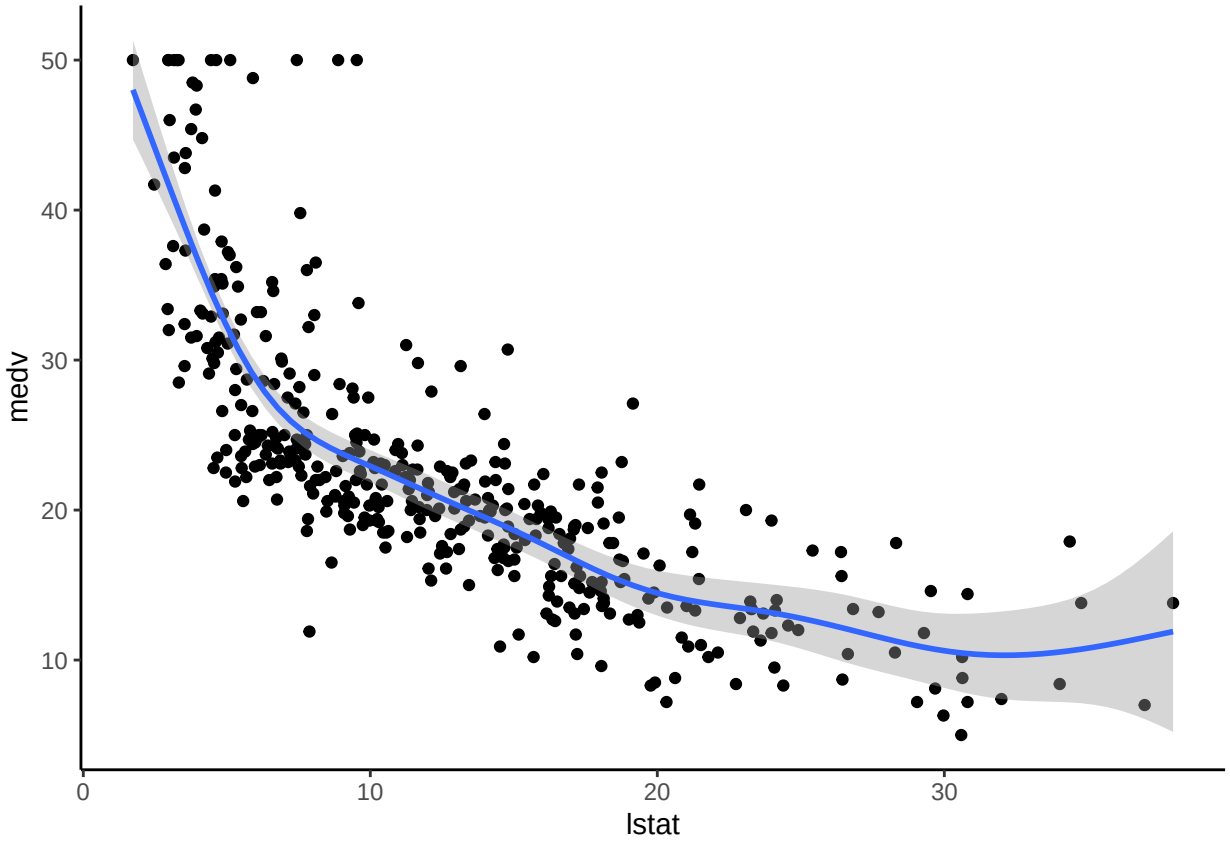
```
## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
```

```
# build the model
modelcGam <- gam(medv ~s(lstat), data = trd)
# make predictions
pmodelcGam <- modelcGam %>% predict(ted)
# Model performance
data.frame(
  rmsegam <- RMSE(pmodelcGam, ted$medv),
  r2gam <- R2(pmodelcGam, ted$medv),
  perrorGam <- rmsegam / mean(ted$medv))
```

```
##   rmsegam....RMSE.pmodelcGam..ted.medv. r2gam....R2.pmodelcGam..ted.medv.
## 1                                5.318856                          0.6760512
##   perrorGam....rmsegam.mean.ted.medv.
## 1                           0.2350954
```

Visualize the plot

```
ggplot(trd, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = gam, formula = y ~ s(x))
```



Lets compare all the models linear, p^4, log, splines, and GAM

```
(fourmodelsPercentError <- rbind(perrorL, perrorp4, perrorLog, perrorSp, perrorGam))
```

```
##                [,1]
## perrorL   0.2874712
## perrorp4  0.2383705
```

```
## perrorLog 0.2416489
## perrorSp  0.2350299
## perrorGam 0.2350954
```

Among all 5 models, the best model we have is splines. why? think of how it works. it's like integral, it breaks in to small little part and find the slope thats why the stat_smooth fits the best of the data. Therefore Sp is the best model

Last section.

Multiple Linear Regression in R

http://www.sthda.com/english/articles/40-regression-analysis/168-multiple-linear-regression-in-r/

A little bit duplicated with the first section. However, it will have a slighly more indepth about the performance and accuracy of a model

```
data("marketing", package = "datarium")
modelLS <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
summary(modelLS)
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5932  -1.0690   0.2902   1.4272   3.3951
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.526667   0.374290   9.422   <2e-16 ***
## youtube      0.045765   0.001395  32.809   <2e-16 ***
## facebook     0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.023 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

Let's interpret the data

First we might want to look at the F statistics' p value We have a pretty high significant value. This mean that, at least one of the variable is significantly related to the outcome. You can think of it is true that the parameter (youtube / facebook / newspaper) has the relationship with the response variable

You should have this question, so which one has the relationship with the response varaible?

let's take a look with the following code

```
summary(modelLS)$coef
```

```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  3.526667243 0.374289884  9.4222884 1.267295e-17
## youtube      0.045764645 0.001394897 32.8086244 1.509960e-81
## facebook     0.188530017 0.008611234 21.8934961 1.505339e-54
## newspaper   -0.001037493 0.005871010 -0.1767146 8.599151e-01
```

Look at the T statistics is checking if the b0, b1, b2, b3 is 0 or not. look at the newspaper, we have -0.001037493 as the beta3 right? and look at the t value, -0.1767146. the T value for the newspaper is pretty close to 0 and look at B3, we have -0.001037493, it simply no use at all so we can simply create another model without the news paper. but the rest, youtube, facebook have a high impact for the sales. There is a relationship with the sales. So more money to put in to youtube and facebook, we have more feedback from the sales.

Lets create another model

```
modelLSyf <- lm(sales ~ youtube + facebook, data = marketing)
summary(modelLSyf)
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5572  -1.0502   0.2906   1.4049   3.3994
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.50532    0.35339   9.919   <2e-16 ***
## youtube      0.04575    0.00139  32.909   <2e-16 ***
## facebook     0.18799    0.00804  23.382   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.018 on 197 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
## F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

95% confident interval for the coefficient

```
confint(modelLSyf)
```

```
##                 2.5 %     97.5 %
## (Intercept) 2.80841159 4.20222820
## youtube     0.04301292 0.04849671
## facebook    0.17213877 0.20384969
```

Model accuracy assessment

As we know, simple linear regression, the overall quality is based on R^2 and Residual standard error (RSE)

Remember one thing, the more parameters we have the higher the R^2. eventhough the parameter does not have much effect on the Y response, it will still in play with the R square. Thats the reason why we have adjusted R^2

Residual standard error (RSE), or sigma

The RSE estimate gives a measure of error of prediction. we want small RSE

```
sigma(modelLSyf)/mean(marketing$sales)
```

```
## [1] 0.1199045
```

This number is not to bad. so we have approximately 12% error rate.

Some cool trick, we dont have to type all the parameters. we can simply do

model <- lm(sales ~. data = marketing) that ~ means all

now if we would like to drop the newspaper from the parameter list model <- lm(sales ~. - newspaper, data = marketing)

Alternative model <- update(model, ~. - newspaper)

Predict in R: Model Predictions and Confidence Intervals

http://www.sthda.com/english/articles/40-regression-analysis/166-predict-in-r-model-predictions-and-confidence-intervals/

Outcome: predict outcome for new observations data, display confidence intervals and the predictio intervals.

```
# load the data
data("cars", package = "datasets")
# build the model
modelcar <- lm(dist ~ speed, data = cars)
modelcar
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

Prediction for new data set

```
# create data
cardata <- data.frame(speed <- c(12,19,24))
# predict data
predict(modelcar, newdata = cardata)
```

```
##        1        2        3
## 29.60981 57.13667 76.79872
```

Confidence interval

by default the confidence interval is 95%

```
predict(modelcar, newdata = cardata, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 29.60981 24.39514 34.82448
## 2 57.13667 51.82913 62.44421
## 3 76.79872 68.38765 85.20978
```

From our newly created data. Lets take 19 miles per hr as our example. Distance is in ft we can say that 19 miles per hr has an average stopping dist between 51.82913 to 62.44421. the regression line has predict the value for 19mph with a stopping dist @ 57.13667

Prediction interval prediction interval gives an uncertainty around a single value.

```
predict(modelcar, newdata = cardata, interval = "prediction")
```

```
##        fit       lwr       upr
## 1 29.60981 -1.749529  60.96915
## 2 57.13667 25.761756  88.51159
## 3 76.79872 44.752478 108.84495
```

From the table above, we can say that 95% prediction intervals tell us that with a speed of 19 mph with the stopping distance is 25.76 to 88.51. This means that, based on our model 95% of the cars with a speed of 19 mph have a stopping distance between 25.76 and 88.51

Now you should have at least one question in your mind at this point. WTH? what is the difference between prediction interval and confidence interval?

The key word for both interpreting is average.

prediction interval is predicting a single future value at that point.

confidence interval is predicting the mean.

https://stats.stackexchange.com/questions/16493/difference-between-confidence-intervals-and-prediction-intervals

More on prediction interval or confidence interval

A prediction interval reflects the uncertainty around a single value, while a confidence interval reflects the uncertainty around the mean prediction values. Thus, a prediction interval will be generally much wider than a confidence interval for the same value.

Generally, we are interested in specific individual predictions. So a prediction interval would be more appropriate. Using a confidence interval when you should be using a prediction interval will be underestimate the uncertainty in a given predicted value

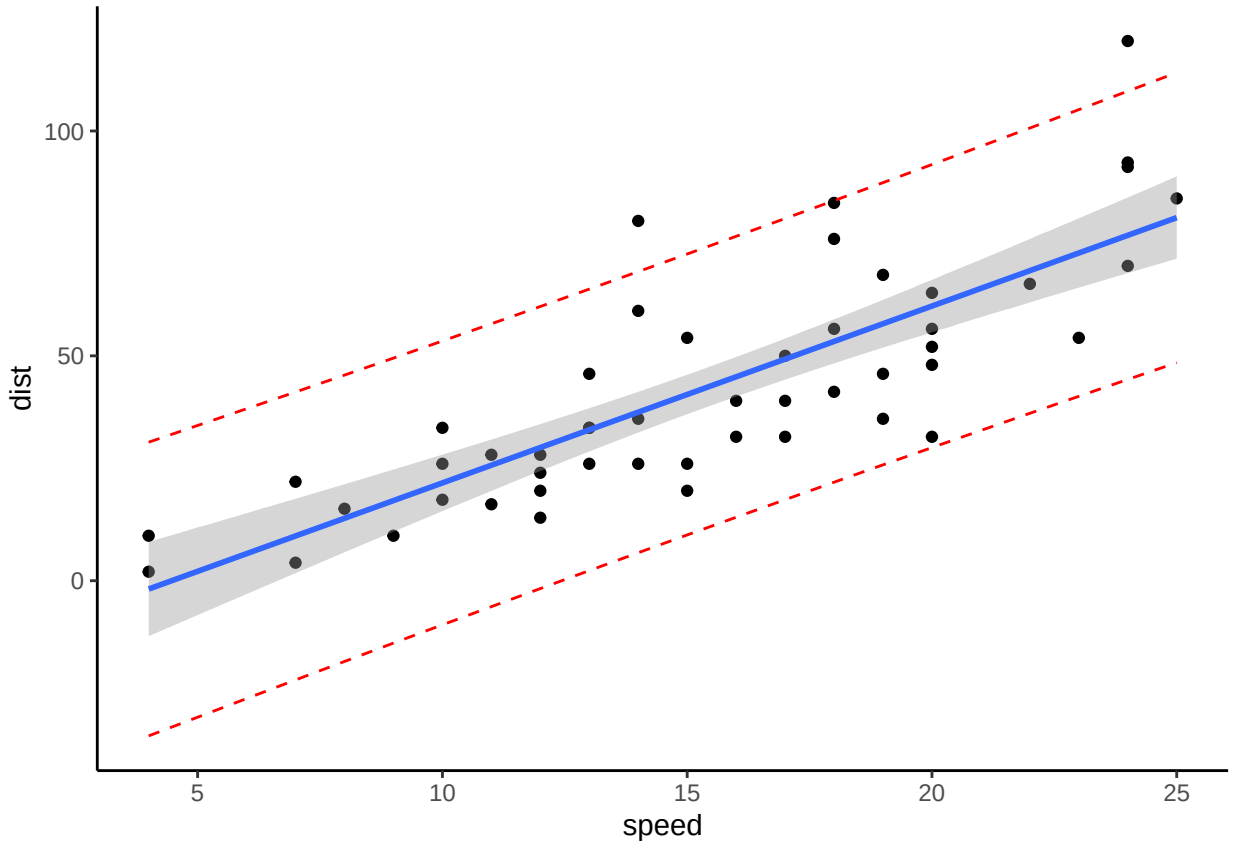
lets build the prediction band and confidence inteval

```
# build the model
data("cars", package = "datasets")
modelcar1 <- lm(dist~speed, data =cars)
# add the predictions
p.int <- predict(modelcar1, interval = "prediction")
```

```
## Warning in predict.lm(modelcar1, interval = "prediction"): predictions on current data refer to _futu
```

```
carD <- cbind(cars, p.int)
# Regression line + confidence intervals
p <- ggplot(carD, aes(speed, dist)) +
  geom_point() +
  stat_smooth(method = lm) +
  geom_line(aes(y = lwr), color = "red", linetype = "dashed") + #prediction interval
  geom_line(aes(y = upr), color = "red", linetype = "dashed") # prediction interval
p
```

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



Regression Model Diagnostics

Linear Regression Assumptions and Diagnostics in R: Essentials

http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/

By the title, you can tell we are going to check if the model works well for the data.

1) inspect the significance of the regression beta coefficients
2) R^2

Those are what we have learnt from the previous sections

This section, we will learn the additional steps to evaluate how well the model fits the data.

For example, linear regression model makes the assumption that the relationship between x and y are linear. but that might not be true. The relationship could be polynomial or logarithmic

the data might contain outlines which will affect the result.

Thats why we need to diagnose the regression model and detect potential problems and check the assumptions is met by the linear regression model

in order to do so, we will check the distribution of residuals errors, which will tell us more about my data.

Main idea for this section. - explaining residuals errors and fitted values. - present linear regression assumptions, as well as potential problems you can will tackles while performing regression analysis - We will talk about some built in diagnostic plots in R to test the assumptions about our linear regression model.

Lets get started

```
if(!require("broom")){
  install.packages("broom")
  library(tidyverse)
library(broom)
}
```

```
## Loading required package: broom
```

```
theme_set(theme_classic())
```

```
# load the data
data("marketing", package = "datarium")
# inspect the data
sample_n(marketing, 3)
```

```
##   youtube facebook newspaper sales
## 1   10.44    58.68     90.00  8.64
## 2  345.12    51.60     86.16 31.44
## 3  195.96    37.92     63.48 20.28
```

Build the regression model

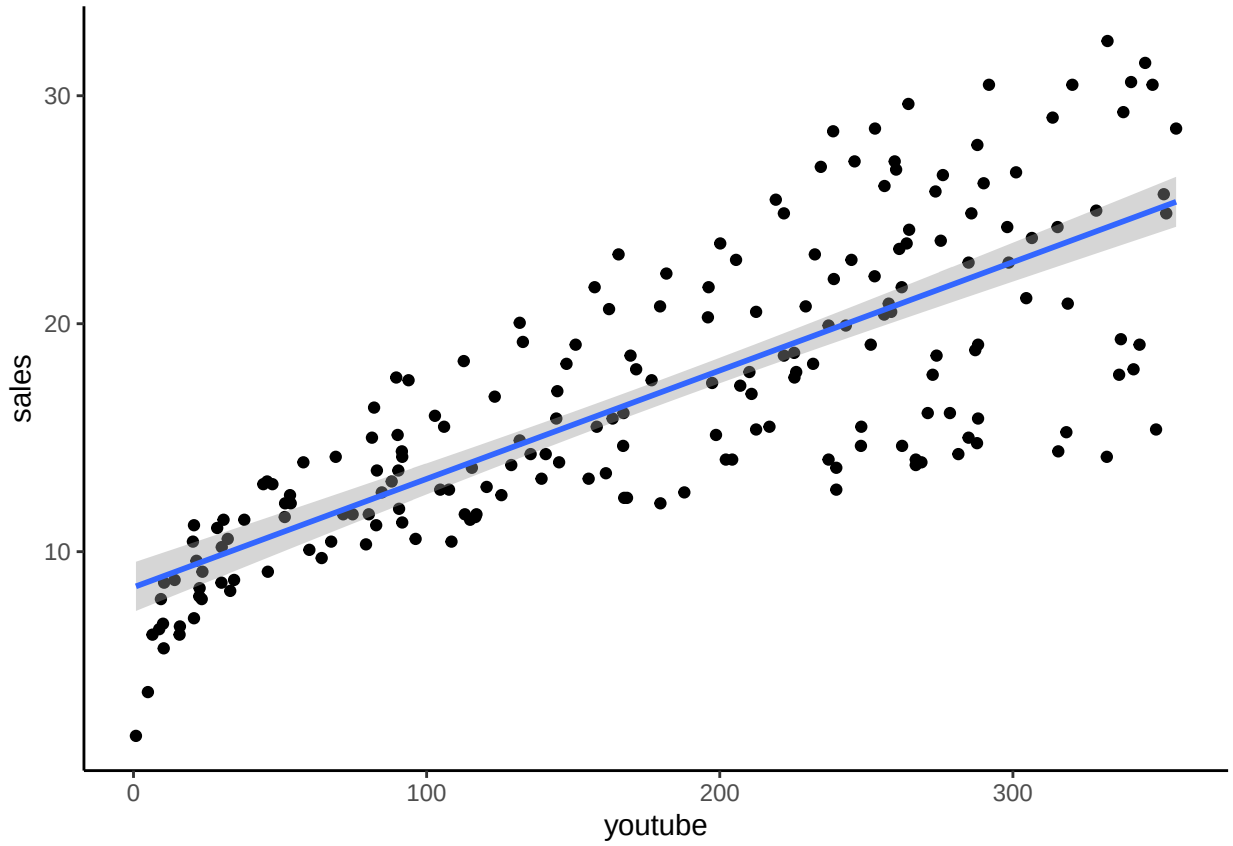
```
modelm <- lm(sales ~ youtube, data = marketing)
modelm
```

```
##
## Call:
## lm(formula = sales ~ youtube, data = marketing)
##
## Coefficients:
## (Intercept)      youtube
##     8.43911      0.04754
```

Fitted value (predicted value) is the value that is predicted from the regression line.

```
ggplot(marketing, aes(youtube, sales)) +
  geom_point() +
  stat_smooth(method = lm)
```

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



augment is from broom package

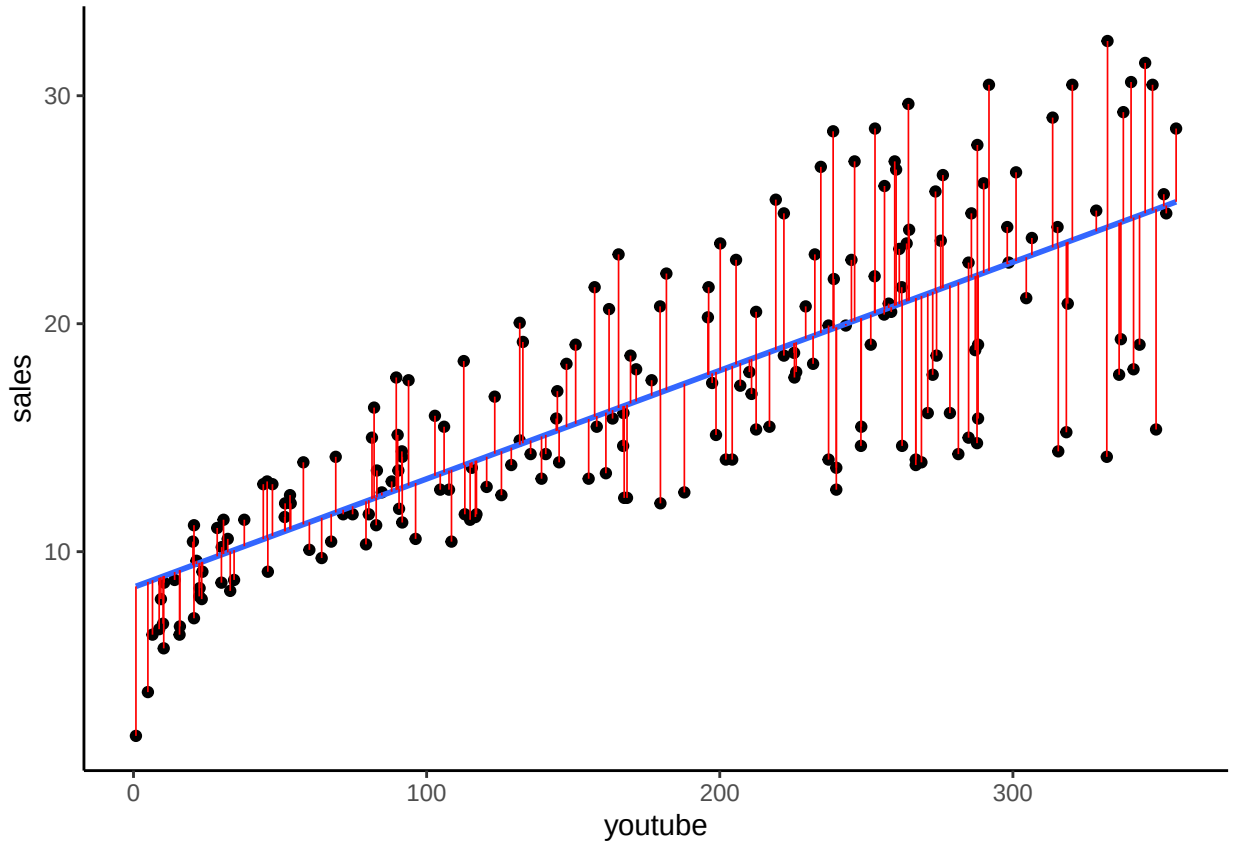
```
model.diag.metrics <- augment(modelm)
head(model.diag.metrics)
```

```
## # A tibble: 6 x 8
##   sales youtube .fitted .resid .std.resid    .hat .sigma   .cooksd
##   <dbl>   <dbl>   <dbl>  <dbl>      <dbl>   <dbl>  <dbl>     <dbl>
## 1 26.5    276.    21.6   4.96       1.27   0.00970   3.90 0.00794
## 2 12.5     53.4   11.0   1.50       0.387  0.0122    3.92 0.000920
## 3 11.2     20.6    9.42  1.74       0.449  0.0165    3.92 0.00169
## 4 22.2    182.    17.1   5.12       1.31   0.00501   3.90 0.00434
## 5 15.5    217.    18.8  -3.27      -0.839  0.00578   3.91 0.00205
## 6  8.64    10.4    8.94 -0.295     -0.0762 0.0180    3.92 0.0000534
```

Lets plot the residuals error in red

```
ggplot(model.diag.metrics, aes(youtube, sales)) +
  geom_point() +
  stat_smooth(method = lm, se = F) +
  geom_segment(aes(xend = youtube, yend = .fitted), color = "red", size =0.3)
```

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



In order to check the regression assumption, we need to see the distribution of the residuals

Linear regression makes several assumptions about the data, such as :

Linearity of the data. The relationship between the predictor (x) and the outcome (y) is assumed to be linear.

Normality of residuals. The residual errors are assumed to be normally distributed.

Homogeneity of residuals variance. The residuals are assumed to have a constant variance (homoscedasticity)

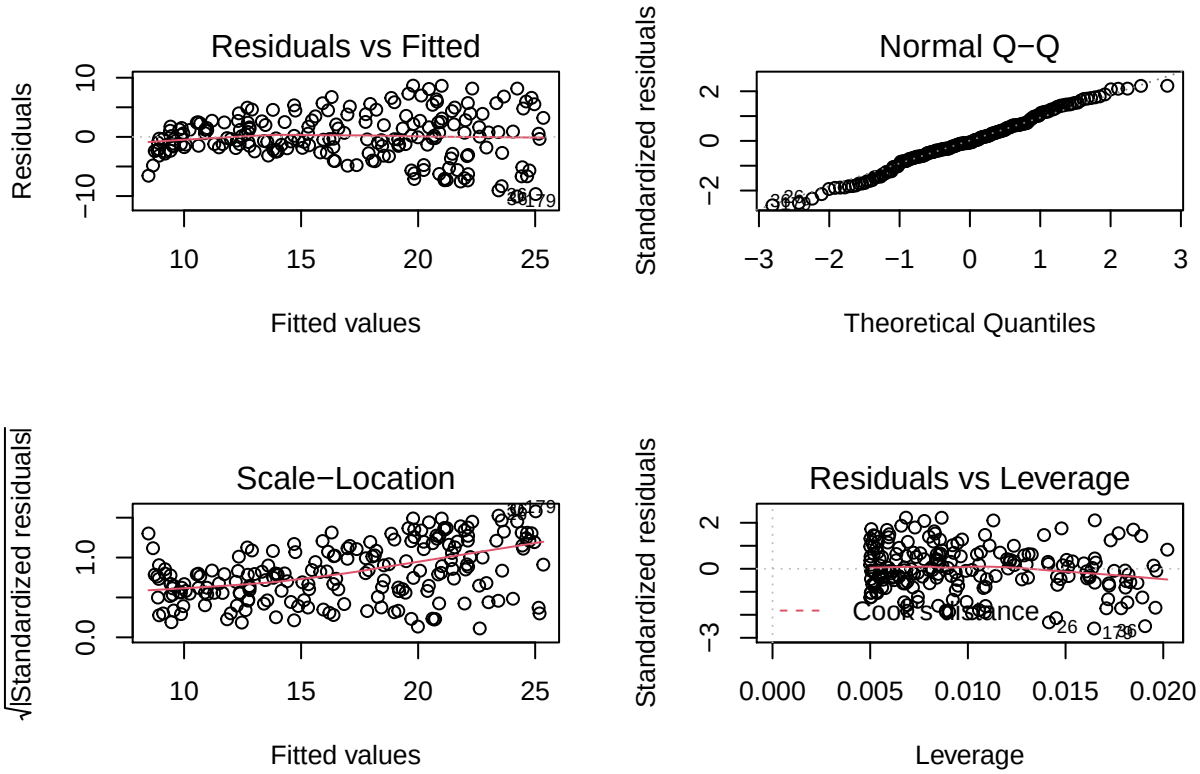
Independence of residuals error terms.

You should check whether or not these assumptions hold true. Potential problems include:

Non-linearity of the outcome - predictor relationships Heteroscedasticity: Non-constant variance of error terms. Presence of influential values in the data that can be: Outliers: extreme values in the outcome (y) variable High-leverage points: extreme values in the predictors (x) variable

Those problems can be solved by diagnostic plot

```
par(mfrow = c(2,2))
plot(modelm)
```



How to read it?

https://data.library.virginia.edu/diagnostic-plots/

Residuals vs Fitted. Used to check the linear relationship assumptions. A horizontal line, without distinct patterns is an indication for a linear relationship, what is good.

Normal Q-Q. Used to examine whether the residuals are normally distributed. It's good if residuals points follow the straight dashed line.

Scale-Location (or Spread-Location). Used to check the homogeneity of variance of the residuals (homoscedasticity). Horizontal line with equally spread points is a good indication of homoscedasticity. This is not the case in our example, where we have a heteroscedasticity problem.

Residuals vs Leverage. Used to identify influential cases, that is extreme values that might influence the regression results when included or excluded from the analysis. This plot will be described further in the next sections.

Lets create a better table for model.diag.metrics

```
names(model.diag.metrics)
```

```
## [1] "sales"      "youtube"    ".fitted"    ".resid"     ".std.resid"
## [6] ".hat"       ".sigma"     ".cooksd"
```

Before and after

```
index <- c(1:nrow(model.diag.metrics))
model.diag.metrics1 <- data.frame(index,model.diag.metrics[ , c(-7)])
names(model.diag.metrics1)
```

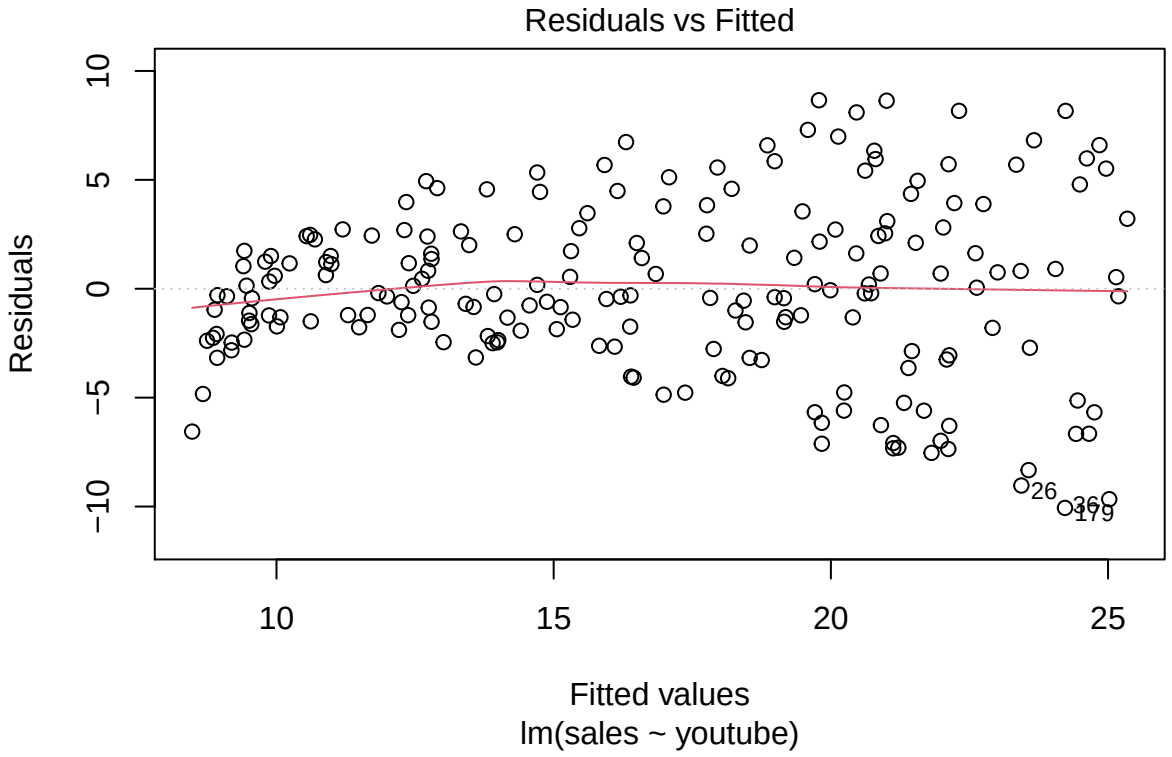
```
## [1] "index"      "sales"      "youtube"    ".fitted"    ".resid"
## [6] ".std.resid" ".hat"       ".cooksd"
```

fitted values = predicted value from the regression line .resid = residual errors .hat = outliers .std.resid =

detect outliers, extreme values -> standardized residuals = residuals / standard errors .cooksd = Cook's distance -> detect influential values outliers or high leverage point $

Let's look at the plot one by one

```
plot(modelm,1)
```



Good plot would be show no fitted pattern, which means the red line should be a horizontal line at 0. If the red line is not at 0 or around 0 there might be a problem with our linear model, which means we need to use non linear model. In our example, plot 1 residuals vs fitted, there is no pattern in the residual plot, so we can assume it is a linear relationship sales ~ youtube.

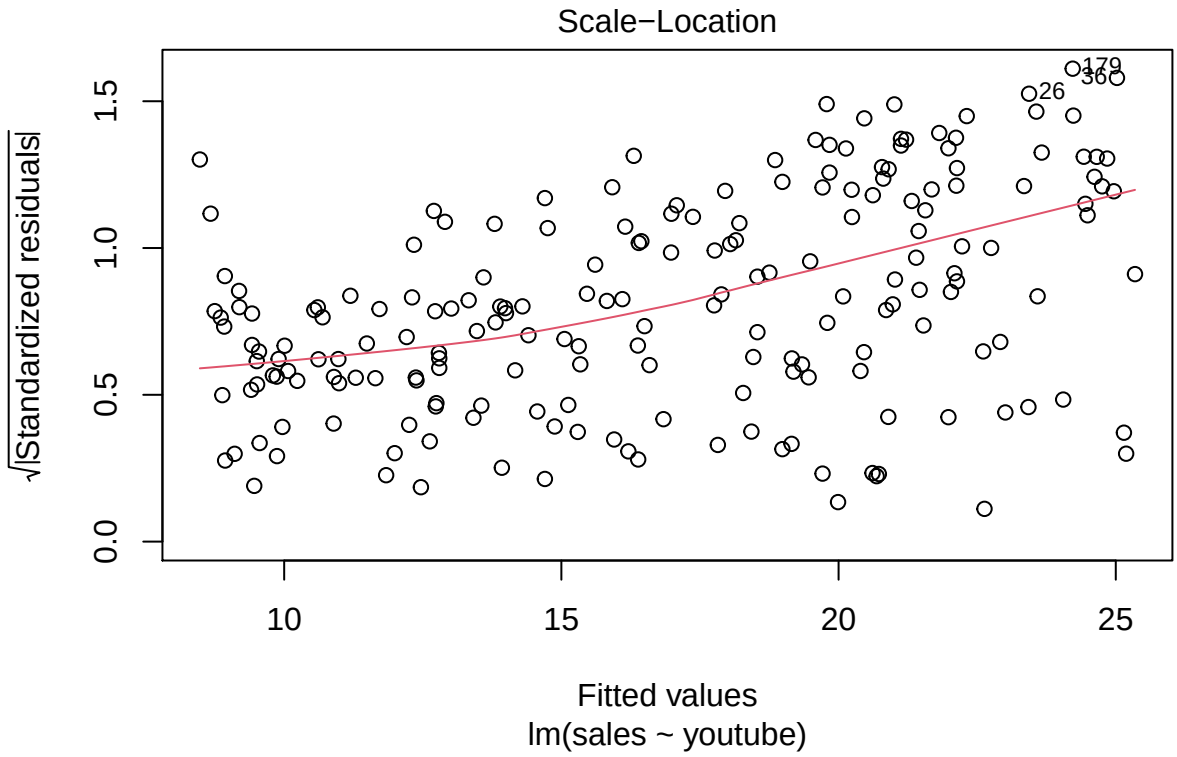
if the residual plot is non linear relationship, you need to transfrom your data ith log(x), sqrt(x), x^2 in the regression and so on.

You might ask how do you know if it is non linear? well it will be obvious that red line is something else like quadratic or something. reference link: non linear https://i0.wp.com/blogs.sas.com/content/iml/files/2019/06/residsmooth1.png?ssl=1

Homogeneity of variance

This assumption can be checked by examining the scale - location plot, also known as the spread- location plot

```
plot(modelm,3)
```



Good plot for scale - location is the data set is close to gether and close the the line. since this plot is spreading wider when x increases, so this is not a good plot.

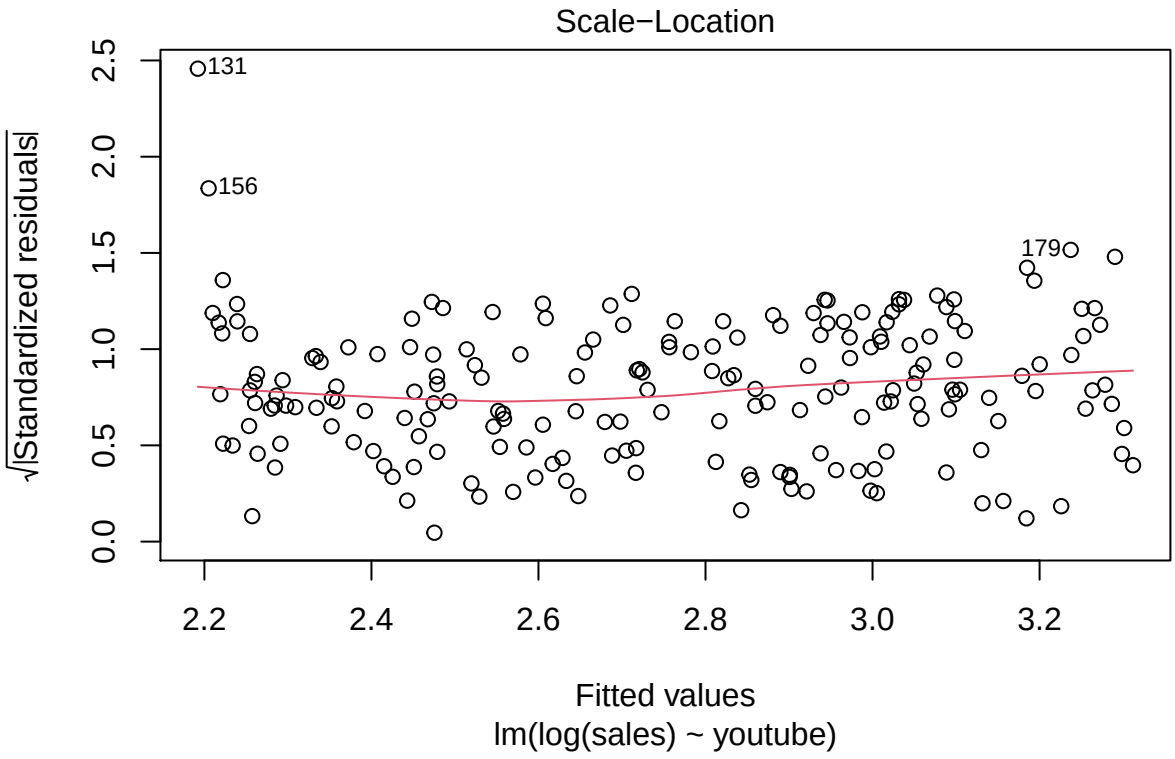
So i can say not a constant variances in the residuals error (heterosedasticity)

Remember, by theory epsilon has mean zero and the variance is constant.

a possible solution to reduce the variance is to use log or square root transformation for variable y

lets try again by transforming the plot

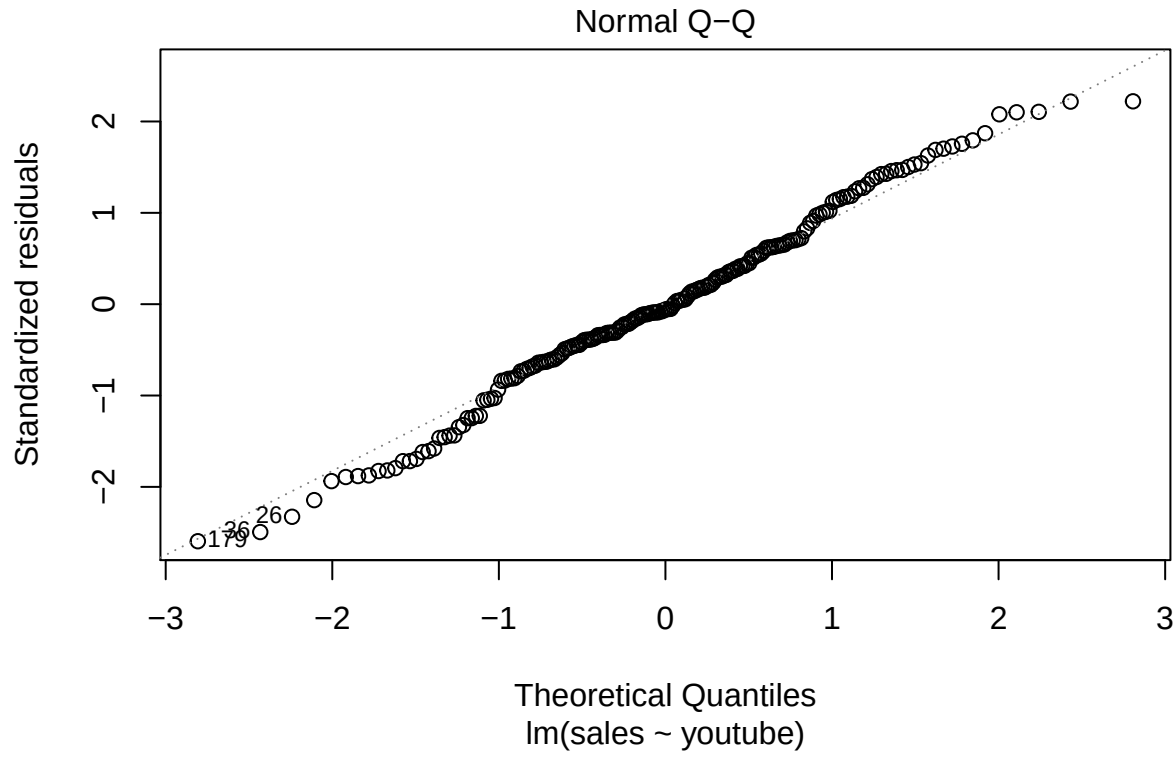
```
modelmLog <- lm(log(sales) ~ youtube, data = marketing)
plot(modelmLog,3)
```



This plot looks a lot better

Normality of Residuals this is also called QQ plot. This plot can check the normality assumption. The normal probability plot of residuals should approximately follow a straight line like below.

```
plot(modelm, 2)
```



Outliers and high Leverage points

The outlier will directly affect the RSE because it is so far away from the regression line. Outlier can be identified by examining the standardized residual (or studentized residual), which is the residual divided by the estimated standard error.

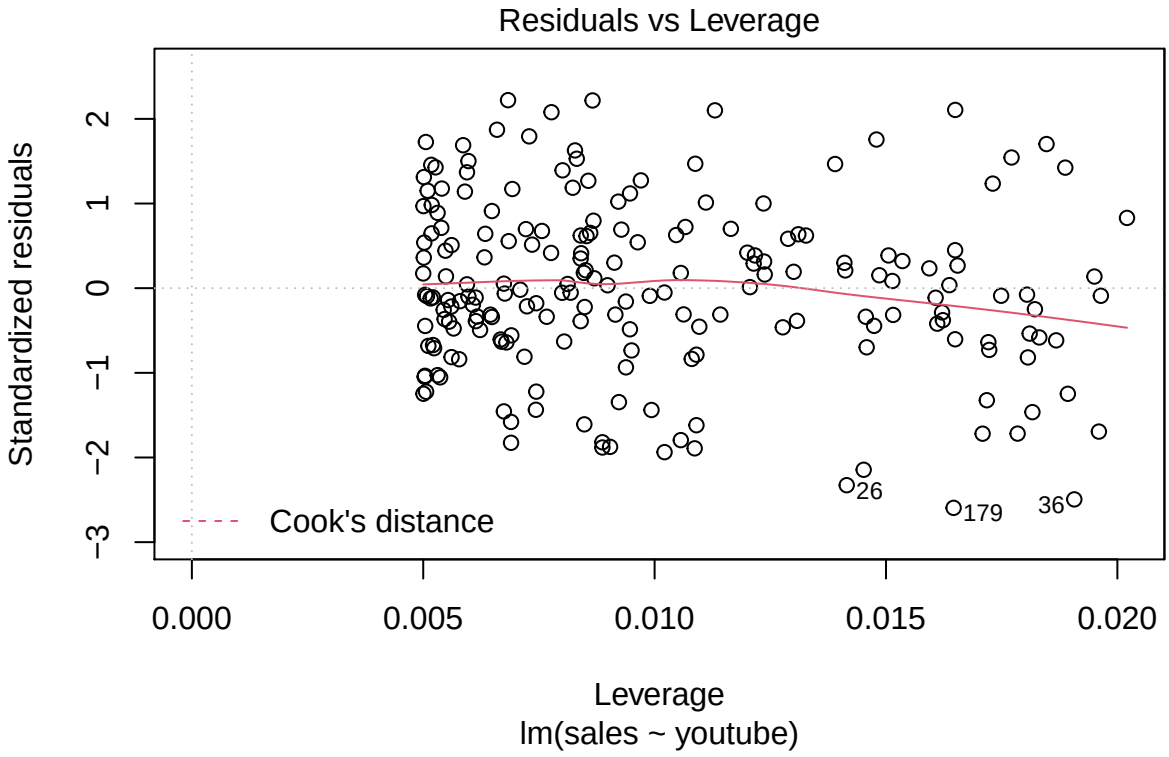
Observations whose standardized residuals are greater than 3 in absolute value are possible outliers (James et al. 2014).

High leverage points (hat value)

A value of this statistic above 2(p + 1)/n indicates an observation with high leverage (P. Bruce and Bruce 2017); where, p is the number of predictors and n is the number of observations.

Residuals vs Leverage plot

```
plot(modelm, 5)
```



The plot will highlight the most extreme points. 26, 179, 36. as you can see no outliers that have exceed 3 standard deviations, which is a good plot.

Influential values

Not all outliers (extreme data points) are influential in linear regression analysis

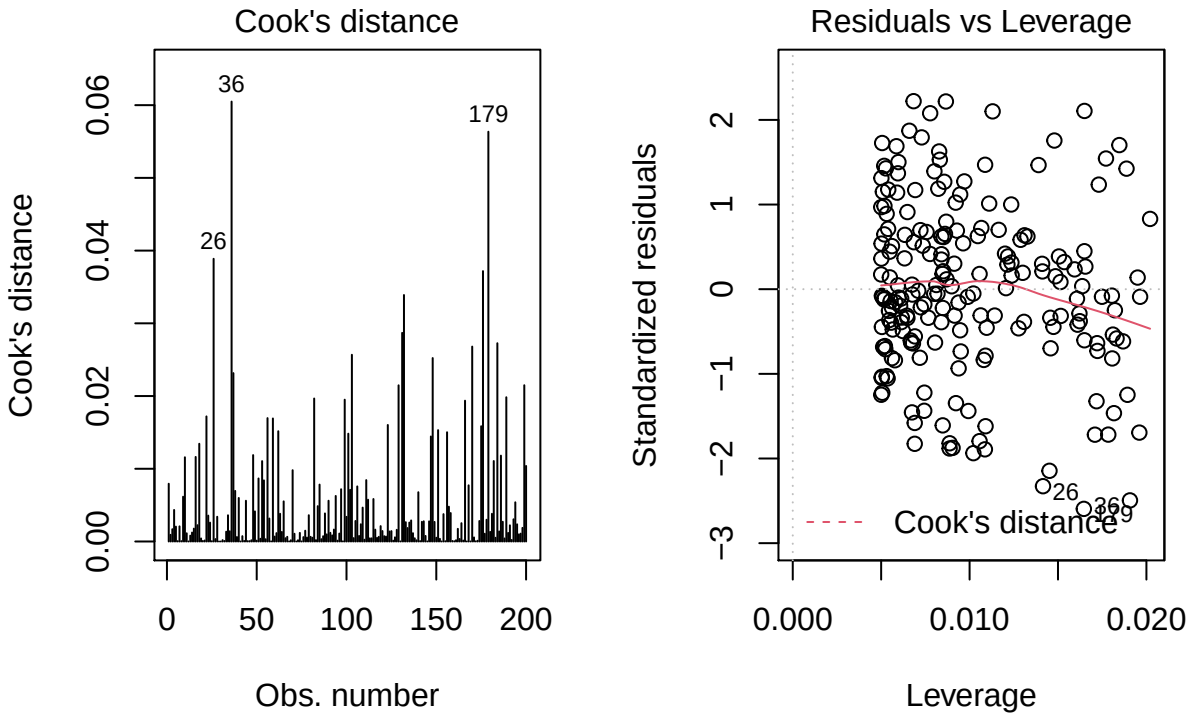
We use Cook's distance to determine the influence of a value. This metric defines influence as a combination of leverage and residual size.

High influence if Cook's distance exceeds 4/ (n - p - 1) n = # of observations p = number of predictor variables.

again residuals vs leverage plot can help us to find influential observations. outlying values are generally located at the upper right corner or lower right corner.

Those corners will have direct influential against a regression line.

```
par(mfrow=c(1,2))
# Cook's distance
plot(modelm, 4)
# Residuals vs Leverage
plot(modelm, 5)
```



Now we have been talking about Cook's distance for the last 10 mins, so how do i know if the outliers are influencing the regression line?

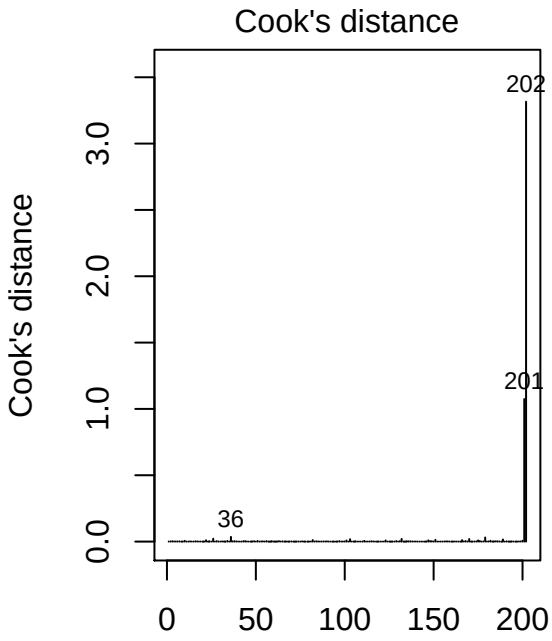
First from the Residuals vs Leverage plot, i am not seeing any red dashed line, so that means my outliers is not affecting the regression line a lot. If we see the dashed line thats mean we are close to the Cook's distance line which means it is some how affecting the regression line seriously. but usually if the outliers are within the Cook's distance we are all good.

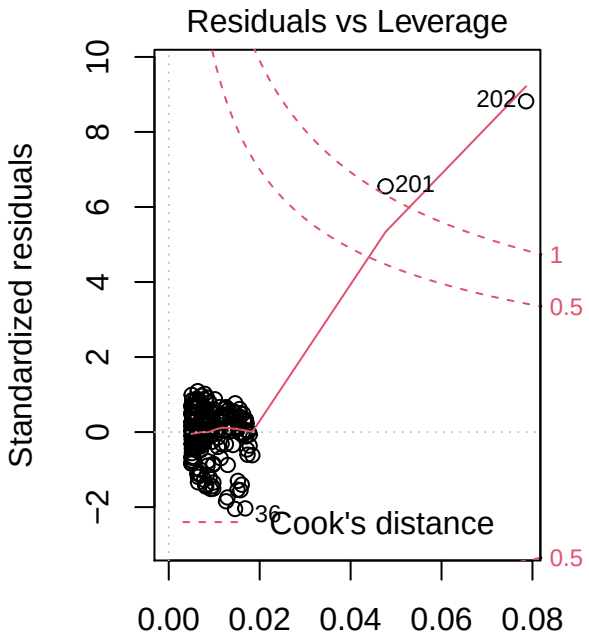
By default, only top 3 most extreme values are labeled on Cook's distance plot, if I want to add more i can use the following code plot(modelm, 4, id.n = 5)

if i would like to access those distance later, i can use the following code model.diag.metrics %?% top_n(3, wt = .cooksd)

Lets create something that is outside the Cook's distance

```
dfcook <- data.frame(
  x <- c(marketing$youtube, 500,600),
  y <- c(marketing$sales, 80,100)
)
modelm2 <- lm(y~x, dfcook)
par(mfrow = c(1,2))
# Cook's distance
plot(modelm2, 4)
# Residuals vs Leverage
plot(modelm2, 5)
```





As you can see from the Residuals vs Leverage, you can see there are 2 outliers are outside of the dashed line (Cook's distance) those are the outliers that are affecting the regression line directly. In the other words, those data are outside of the cook's distance would simply mean they have a high cook's value.

The plot identified the influential observation as #201 and #202. If you exclude these points from the analysis, the slope coefficient changes from 0.06 to 0.04 and R2 from 0.5 to 0.6. Pretty big impact!

There are a lot of concepts that must be undersood, so take your time to go thru it again.

Discussion. This section describes linear regression assumptions and how to diagnose the potential problems in the model You must visualizing the residuals and the patterns in residuals is not a stop signal. your current regression model might not be the best way to udnerstand your data.

Here are the potential problems: non - linear relationships between the outcome and the predictor variables. When facing to this problem, one solution is to include a quadratic term, such as polynomial terms or log transformation. See Chapter

Existence of important variables that you left out from your model. Other variables you didn't include (e.g., age or gender) may play an important role in your model and data.

Presence of outliers. If you believe that an outliers have occurred due to an error in data collection and entry, then one solution is to simply remove the concerned observation.