ISLR on TV Sales

Brandon Tao

Dec 20, 2018

Sections

0. Setup
1. Read the advertising data to make a tibble
2. Scatterplots and models y = f(x) + e
3. Comparing univariate model plot
4. The regression three variable domain
5. Mulitple linear Regression
6. The Mean Squared Error

0. Setup

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------- tidyverse 1.
2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.7
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

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

```
source('hw.R')
library(rgl)

fixLights <- function(specular = gray(c(.3,.3,0))){
  clear3d(type = "lights")
  light3d(theta = -50,phi = 40,
    viewpoint.rel = TRUE, ambient = gray(.7),
    diffuse = gray(.7),specular = specular[1])

  light3d(theta = 50,phi = 40,
    viewpoint.rel = TRUE, ambient = gray(.7),
    diffuse = gray(.7),specular = specular[2])

  light3d(theta = 0,phi = -70,
    viewpoint.rel = TRUE, ambient = gray(.7),
```

```
    diffuse = gray(.7),specular = specular[3])
}
```

1. Read the advertising data to make a tibble

```
adSales <- read_csv('Advertising.csv')
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   TV = col_double(),
##   Radio = col_double(),
##   Newspaper = col_double(),
##   Sales = col_double()
## )
```

```
adSales
```

```
## # A tibble: 200 x 5
##       X1    TV Radio Newspaper Sales
##    <int> <dbl> <dbl>     <dbl> <dbl>
##  1     1 230.   37.8      69.2  22.1
##  2     2  44.5  39.3      45.1  10.4
##  3     3  17.2  45.9      69.3   9.3
##  4     4 152.   41.3      58.5  18.5
##  5     5 181.   10.8      58.4  12.9
##  6     6   8.7  48.9      75     7.2
##  7     7  57.5  32.8      23.5  11.8
##  8     8 120.   19.6      11.6  13.2
##  9     9   8.6   2.1       1     4.8
## 10    10 200.    2.6      21.2  10.6
## # ... with 190 more rows
```

```
# Remove the counting integers in column 1
adSales <- adSales[,-1]
adSales
```

```
## # A tibble: 200 x 4
##       TV Radio Newspaper Sales
##    <dbl> <dbl>     <dbl> <dbl>
##  1 230.   37.8      69.2  22.1
##  2  44.5  39.3      45.1  10.4
##  3  17.2  45.9      69.3   9.3
##  4 152.   41.3      58.5  18.5
##  5 181.   10.8      58.4  12.9
##  6   8.7  48.9      75     7.2
##  7  57.5  32.8      23.5  11.8
##  8 120.   19.6      11.6  13.2
##  9   8.6   2.1       1     4.8
```

```
## 10 200.    2.6      21.2  10.6
## # ... with 190 more rows
```

We are going fit a linear model with Sales as the dependent variable. The Unit of measure is "Thousands of Units Sold". Our linear model input variables are: Radio, TV and Newspaper expenditures.
All the units of measure "Budget in $1000s".

Input variables may be called explanatory variables, predictor, or independent variables in different contexts.

Looking at univariate summary statistics provides one way to start learning about the data.

```
summary(adSales)
```

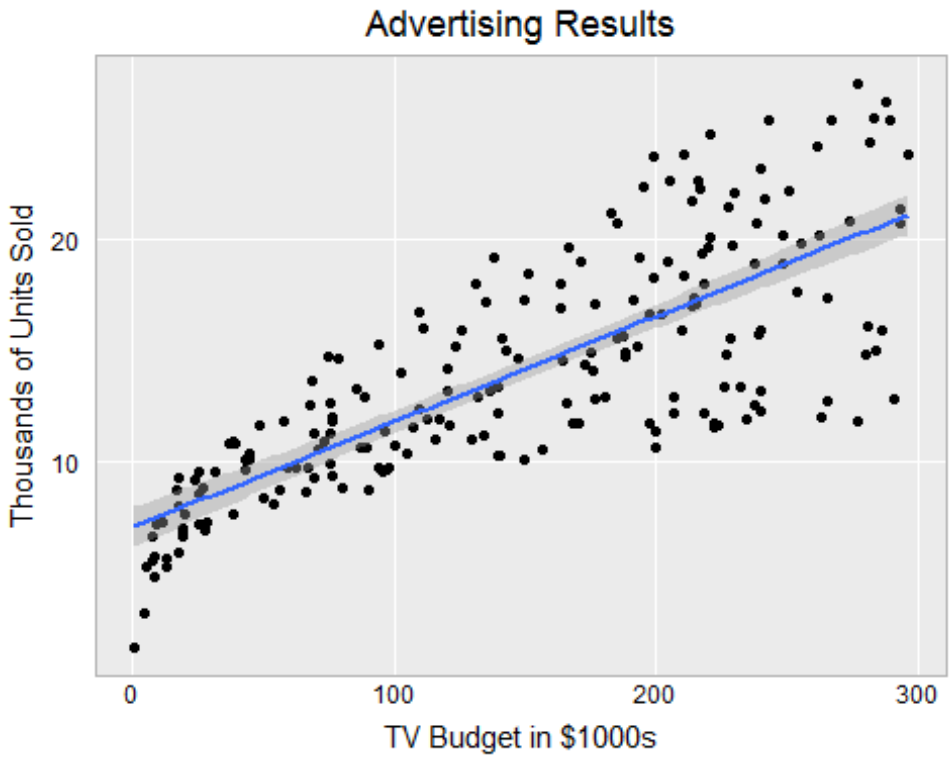
```
##        TV             Radio          Newspaper          Sales
##  Min.   :  0.70   Min.   : 0.000   Min.   :  0.30   Min.   : 1.60
##  1st Qu.: 74.38   1st Qu.: 9.975   1st Qu.: 12.75   1st Qu.:10.38
##  Median :149.75   Median :22.900   Median : 25.75   Median :12.90
##  Mean   :147.04   Mean   :23.264   Mean   : 30.55   Mean   :14.02
##  3rd Qu.:218.82   3rd Qu.:36.525   3rd Qu.: 45.10   3rd Qu.:17.40
##  Max.   :296.40   Max.   :49.600   Max.   :114.00   Max.   :27.00
```

Scatterplots are often more interesting because they can suggest functional relationships and show more of the data domain that supports the model.

2. Scatterplots and models y = f(x) + e

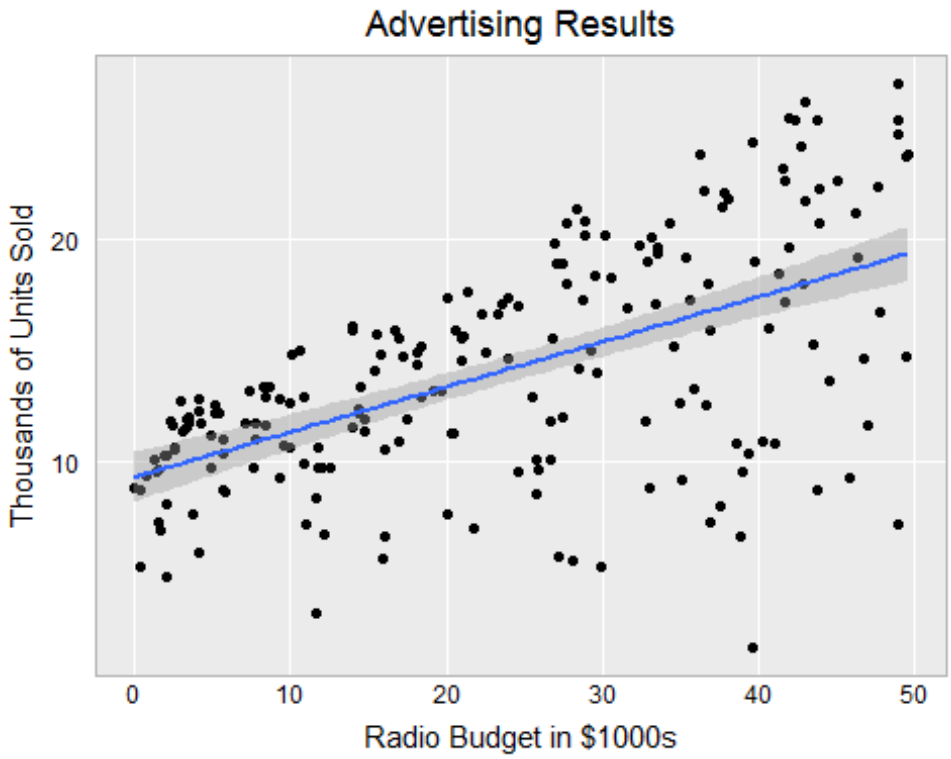
Here y is Sales and x is one of the three input variables The plots are similar to those the ISLR text, Section 2.

```
pTV <- ggplot(adSales, aes(x = TV, y = Sales)) +
  geom_point() +
  geom_smooth(method = 'lm') + hw +
  labs(x = 'TV Budget in $1000s',
    y = 'Thousands of Units Sold',
    title = 'Advertising Results')
pTV
```



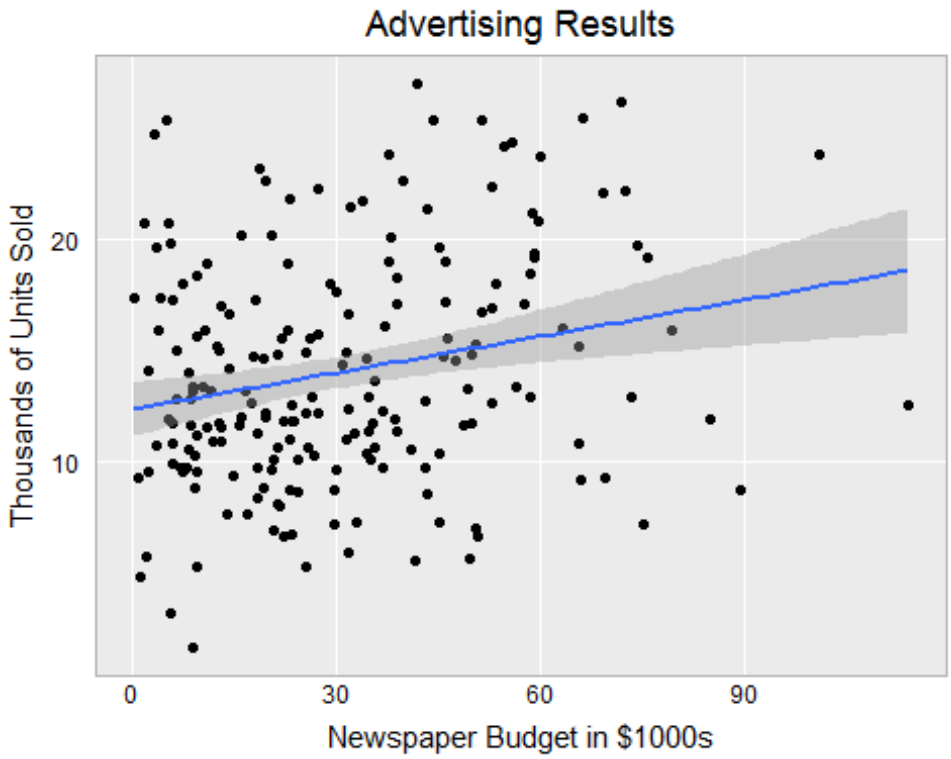
The vertical spread of the points increases from left to right. Correspondingly, the variance of the residuals about the fitted line increases going left to right. This violates the assumption that the errors are independent and Identically distributed. The Sales for the two smallest TV Budget points look anomalous.

```
pRadio <- ggplot(adSales, aes(x = Radio, y = Sales)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(x = 'Radio Budget in $1000s',
    y = 'Thousands of Units Sold',
    title = 'Advertising Results') + hw
pRadio
```



The Sales variation about the regression line also increases as the Radio Budget increases. We can see some sales values far below the regression line. From the plot above, we know two are associated with very low TV budgets. Maybe more departures from the regression line are associated with the TV budgets.

```
pNews <- ggplot(adSales, aes(x = Newspaper, y = Sales)) +
  geom_point() +
  geom_smooth(method = 'lm') +
labs(x = 'Newspaper Budget in $1000s',
  y = 'Thousands of Units Sold',
  title = 'Advertising Results') + hw
pNews
```



Here the vertical spread of points about the regression line looks fairly uniform going from left to right, up to budget of $55,000. The density of points reduces after this. There are only two points larger the $90,000 and in this part of the plot it is hard to assess the local variation in thousands of units sold.

3. Comparing univariae model plots

Showing a sequence of scatterplots with each explanatory variable on the x-axis can be suggestive about a functional relationship but can misleading in two ways.

1) The budget range on the x-axis differs the for TV, Radio, and Newspaper plots. This will make the slope appear differ even if slope were all the same in terms units sold per budget dollars spent. I this instance we can common x-axis scale because the scale units are all budget dollars.
   Often explanatory variables different units of measure.

2) The unit solds may well be some function of all three explanatory variables. Units_sold = f(TV, Radio, Newspapaer) + error

Below we will try the linear function: Units_sold = b0 + b1*TV + b2*Radio + b3*Newspaper + error.

We also try some addition linear models the may delete explanatory variables and/or include explanatory variables computed the original three explanatory variables.

Side comment: A rotating 3-D ray glyph plot can show all four variables and provide visual clues about models that may fit the data pretty well. Slow rotation provides a way to show

a 3-D scatterplots. Ray angle can encode the units sold. A ray pointing straight up can encode the maximum sales. A ray points straight down can encode the minimum sales. We can use intermediate angles that point to the right to encode intermediate values.

In ggplot graphics, we use factor levels to specify tibble rows that are to be plotted in separate panels. Then we use the facet_grid function to specify the layout of panels within the composite plot.

Below we use the gather() function to do the following: 1) Stack the TV, Radio, and Newspaper budget columns into a single long column. 2) Create factor that distinguishes Newspaper, TV and Radio rows. 3) Stack the values of other variables (here sales) in long columns. Yes, each sales value will appear three times.

The gather() function below makes this relatively easy as shown below. The first argument is the input tibble name. The "key" argument names the column with the panel factor The "value" argument names the stack column of selected variables The TV:Newspaper is a convenient way to specify a set of adjacent columns to be stacked.

(The column names can be specified as as individual arguments separated by columns)

The factor_key = TRUE says the column specification order such become the levels of the factor. (No alphabetic sorting)

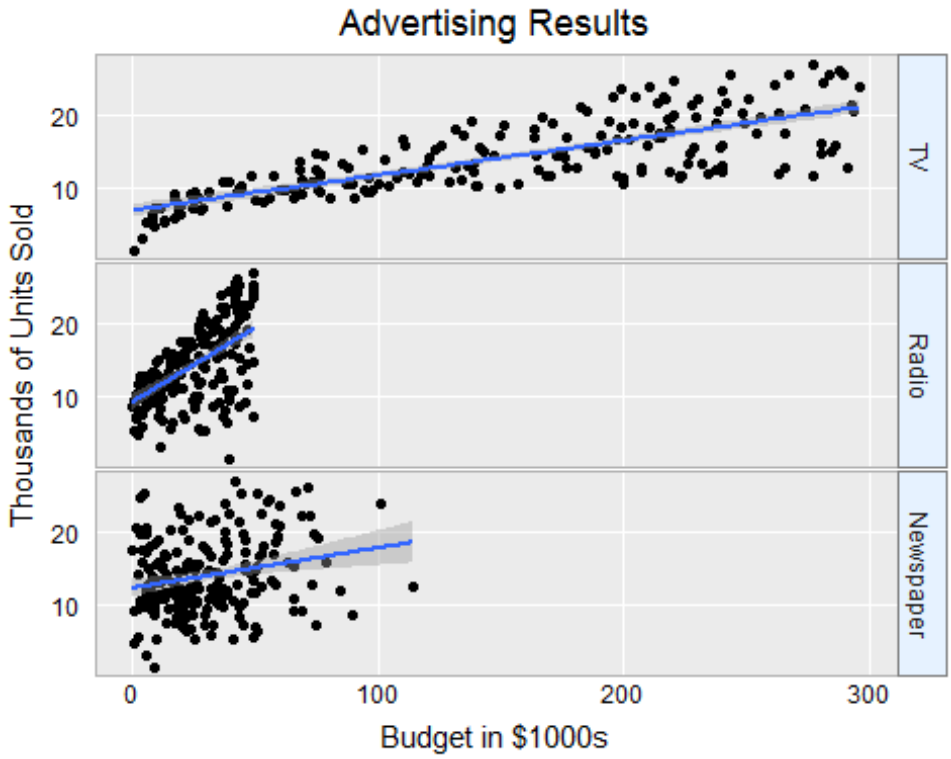
```
adSales
```

```
## # A tibble: 200 x 4
##       TV Radio Newspaper Sales
##    <dbl> <dbl>     <dbl> <dbl>
##  1 230.   37.8      69.2  22.1
##  2  44.5  39.3      45.1  10.4
##  3  17.2  45.9      69.3   9.3
##  4 152.   41.3      58.5  18.5
##  5 181.   10.8      58.4  12.9
##  6   8.7  48.9      75     7.2
##  7  57.5  32.8      23.5  11.8
##  8 120.   19.6      11.6  13.2
##  9   8.6   2.1       1     4.8
## 10 200.    2.6      21.2  10.6
## # ... with 190 more rows
```

```
adSalesG <- gather(
  adSales,
  key = "Media",
  value = "Budget",
  TV:Newspaper,
  factor_key = TRUE
)

adSalesG
```

```
## # A tibble: 600 x 3
##    Sales Media Budget
##    <dbl> <fct>  <dbl>
##  1  22.1 TV     230.
##  2  10.4 TV      44.5
##  3   9.3 TV      17.2
##  4  18.5 TV     152.
##  5  12.9 TV     181.
##  6   7.2 TV       8.7
##  7  11.8 TV      57.5
##  8  13.2 TV     120.
##  9   4.8 TV       8.6
## 10  10.6 TV     200.
## # ... with 590 more rows
```

```
ggplot(adSalesG, aes(x = Budget, y = Sales)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(x = 'Budget in $1000s',
    y = 'Thousands of Units Sold',
    title = 'Advertising Results') +
  facet_grid(Media ~ .)  + hw
```



Note that the strip labels on the right provide the media labels.

Now the x and y axes are the same scale. We clearly see that Radio budget has the steepest slope.

The right ends of the fitted blue lines seem to suggest that $50,000 for Radio advertising yielded about same number units sold as $300,000 for TV sales.
However, it was the combination of the three advertising budgets to led to the sales.

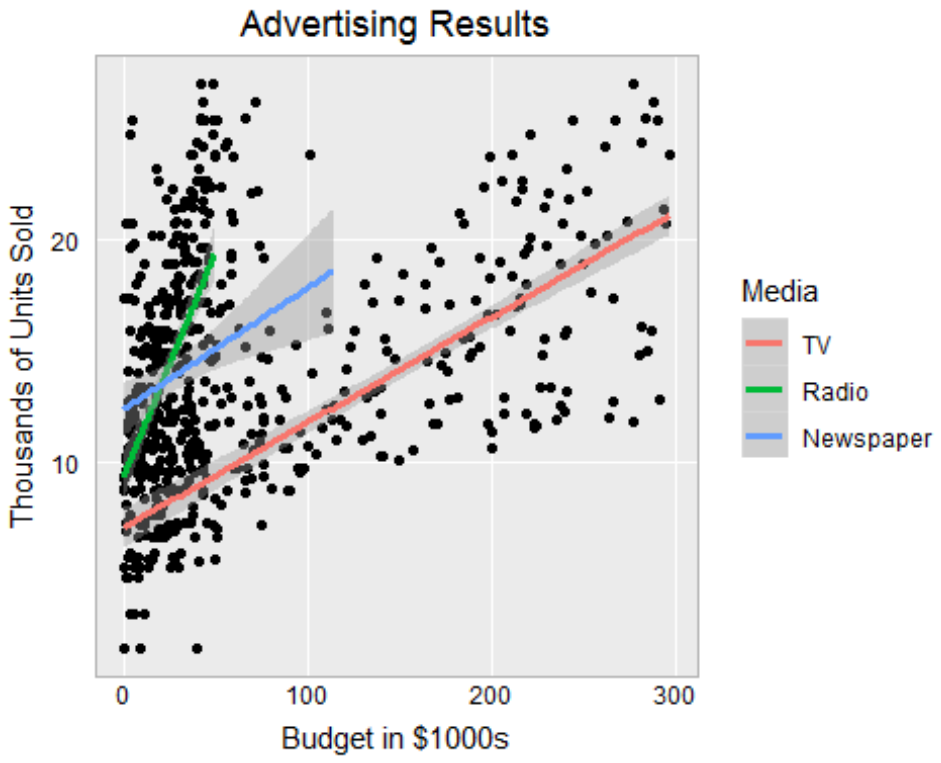
We can't really know if the units sold will increase at the same with if we just increase the Radio budget beyond the current highest amount because we don't have that data.

Is the TV slope larger the Newspaper slope?

For comparison proposes a superposed plot provides an alternative to a juxtaposed panel plot.

The superpose plot distinguished the Media by panel membership. Below we use the line color aesthetic in the geom_smooth to distinguish the three types of Media.

```
ggplot(adSalesG, aes(x = Budget, y = Sales)) +
  geom_point() +
  geom_smooth(method = 'lm', aes(color = Media), size = 1.2) +
  labs(x = 'Budget in $1000s',
    y = 'Thousands of Units Sold',
    title = 'Advertising Results') + hw
```



The Radio green line largest slope. It looks to me like the TV red slope is a bit smaller than the Newspaper blue line slope. Let's check.

```
lm(Sales~TV,data = adSales)
```

```
##
## Call:
## lm(formula = Sales ~ TV, data = adSales)
##
## Coefficients:
## (Intercept)           TV
##     7.03259      0.04754
```

```
lm(Sales~Newspaper,data = adSales)
```

```
##
## Call:
## lm(formula = Sales ~ Newspaper, data = adSales)
##
## Coefficients:
## (Intercept)    Newspaper
##    12.35141      0.05469
```

Why wasn't more spent for Newspaper advertising when it had the largest slope of the three univariate models? Why was so much money spent for TV advertising when the sales increase per $1000 is smallest?

Fitting a multiple linear model with all three variables tells the story.

Regression input variable domain

The domain of the input variables determines the extent to which the model output is supported by actual observations. Using input values outside the domain to obtain estimates is a kind of extrapolation.

Making decisions based extrapolation results should always be viewed as risky endeavor. Sometimes knowledge about the phenomena may suggest the little risk is minimal.

4.1 Domain 3D scatterplot

With three explanatory variables we look at the data domain points relative the range (min and max) of the three variables.

Where will the plot appear?

The plot may appear as icon on the edge of your screen and need to be opened. The window may be hidden beneath your RStudio window

```
domain <- as.matrix(adSales[, -4])
open3d(FOV = 0)  # no perspective projection
```

```
## wgl
##   1
```

```
fixLights()
plot3d(
  domain,
```

```
  type = "s",
  radius = 3.2,
  col = rgb(1, .2, .2),
  aspect = TRUE
)
```

Once the rgl plot is located, left click on the plot lower right corner and drag to change its size.

Left click on the top blue bar and drag to move the plot.

Left click in the plot and drag to rotate the cube.

Right click and move down or up to zoom in or out respectively.

Look at the 8 corners of the plot. Are there are data points near the corners? If so that reduces extrapolation concerns a little.

In linear regression, points on edges of the domain data cloud are more influential than points in the center of the cloud. If one or more of the points have anomalous dependent variable values extrapolation can still be bad.

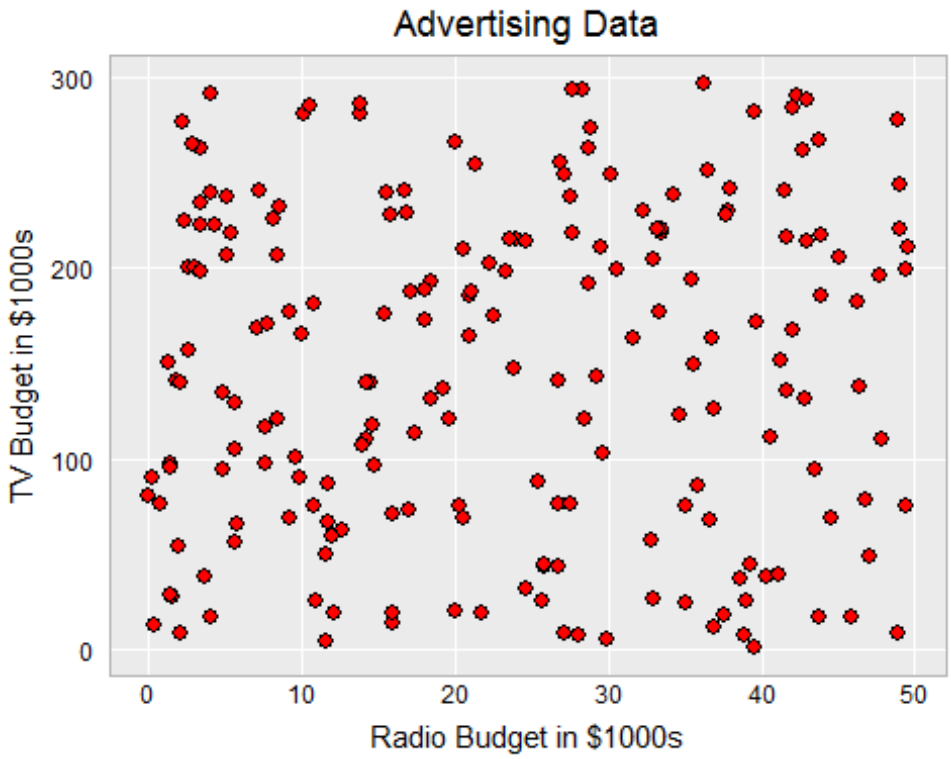
For starters we see a large empty volume around the high Newspaper, low Radio and low TV corner.

We can rotate the plot to focus 2D margin domains and look areas with little or no data. Absence near the edges correspond to cube faces. Alternativey we make three scatterplots: (TV, Radio). (TV, Newspaper), and (Radio, Newspaper).

Sometimes we find big holes in the data domain.

4.2 A look at 2D Radio and TV Domain

```
ggplot(adSales,aes(x = Radio,y = TV)) +
  geom_point(shape = 21,fill = 'red',
     color = 'black',size = 2.5) +
  labs(x = 'Radio Budget in $1000s',
       y = 'TV Budget in $1000s',
    title = 'Advertising Data') + hw
```



This looks like a reasonably good 2D domain for a linear regression model. There are points close to the corners and near the edges. There are no gaping internal gaps.

In statistics, the field of experimental design addresses the production of good regression models domains for answering questions.

4.3 Importance of the model domain

What we can learn from data is limited by the explanatory variables included and the combination provided by their values.

A series of plots and models focues on one variable can be inefficient and misleading. The combination of values can have a dramatic impact on the responds show by the dependent varibles

This inefficiency motivated the development experimental design methodology. Experimental design addresses cost efficient selection of input variables and their combinations. The application of this methodology tremendously changed industrial production. In the history of statistics R. A. Fisher became well known for his agriculture experiments and analysis methods. Edward Deming's work revolutionized the automobile production industry.

The data domain, with possible transformations, feeds into the linear model design matrix. The design matrix impacts both model coefficient and their uncertainty.

In the deep learning world a recurrent challenge is for analysts to gather data to support the learning of patterns. It is still importance to gathering data that covers the data domain of interest when possible.

5. Linear Regression

Below, Sales is the dependent variable. The "~" means all the other variable are explanatory variables

```
adModel1 <- lm(Sales~.,data = adSales)
adModel1

##
## Call:
## lm(formula = Sales ~ ., data = adSales)
##
## Coefficients:
## (Intercept)           TV        Radio    Newspaper
##    2.938889     0.045765     0.188530    -0.001037

summary(adModel1)

##
## Call:
## lm(formula = Sales ~ ., data = adSales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## Radio        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: 1.686 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
```

The Multiple R-squared .897 indicates that the model accounts for roughly 90% of the Sales variability about the grand mean.

The adjusted R-squared is a little smaller and includes a penalty for including more variables in a model. We can include random noise as variables in the model and improve the fit.

Assuming the standardized residuals have roughly a normal distribution we make can statistical inferences about the model. The probability of the F-statistics being so large at random is basically 0. The F-statistic compares fitting all the variables to fitting just the dependent variable mean.

The t-statistics are based individual variables improving the fit with the other variables listed in regression output already in the model. Is there strong evidence that the regression coefficient is not zero? What is the probability that improved fit is due random variation? For TV and Radio, the probability is close to 0. For Newspaper, the p value of .86 suggests that using white noise may have better chance of improving the fit.

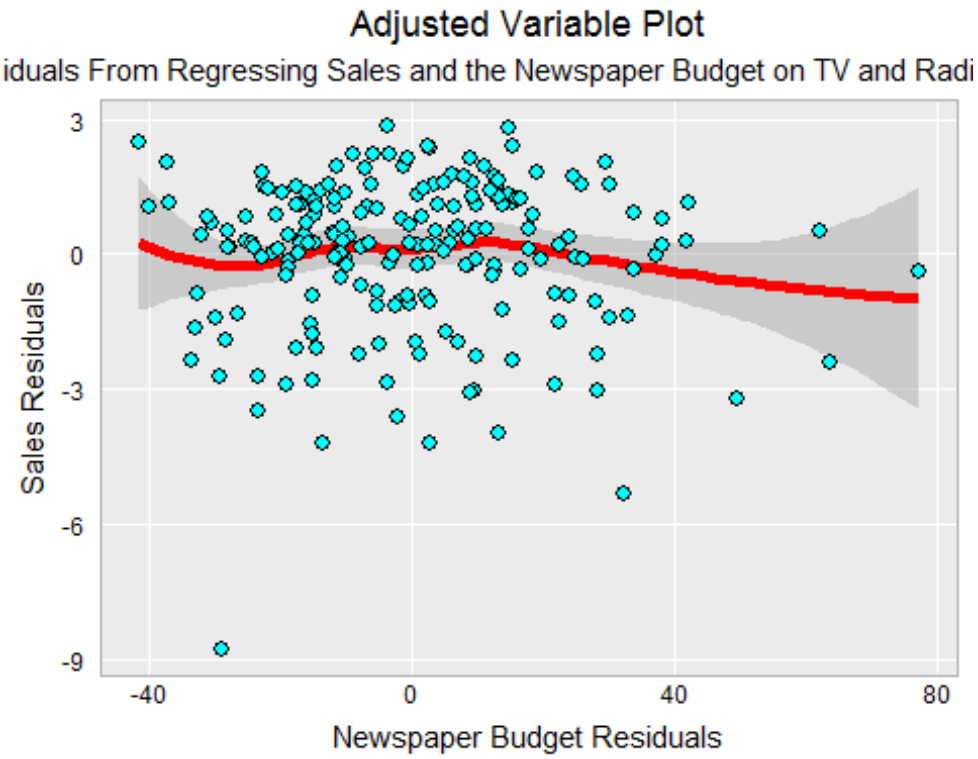
The correlation matrix shows that Newspaper budget is substantially correlated (.354) with the Radio budget

```
cor(adSales[,1:3])
```

```
##                    TV      Radio  Newspaper
## TV         1.00000000 0.05480866 0.05664787
## Radio      0.05480866 1.00000000 0.35410375
## Newspaper 0.05664787 0.35410375 1.00000000
```

In the 1 variable newspaper model, newspaper get some credit for high sales because it was high when Radio sales were high. However when TV and Radio are in the model the Newspaper budget has almost no impact on the fit.

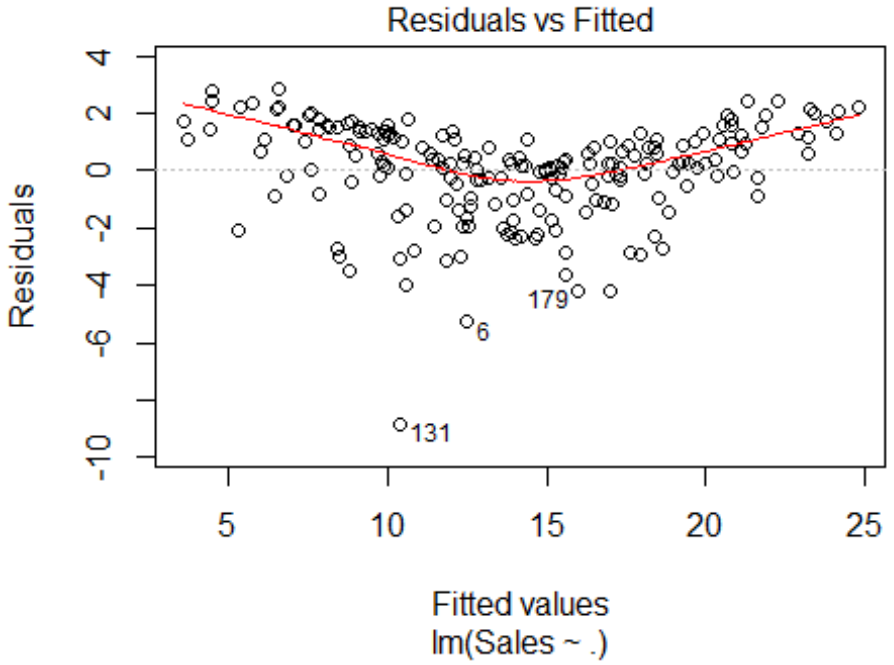
```
Res_Sales <- residuals(lm(Sales~TV+Radio,adSales))
Res_News <- residuals(lm(Newspaper~TV + Radio,adSales))
ResRes <- tibble(Res_Sales, Res_News)
subtitletxt <- paste("Residuals From Regressing",
  "Sales and the Newspaper Budget on TV and Radio Budgets")

ggplot(ResRes, aes(x = Res_News, y = Res_Sales)) +
  geom_smooth(size = 2, color = 'red') +
  geom_point(shape = 21, fill = 'cyan',
    color = "black", size = 2.5) +
  labs(x = "Newspaper Budget Residuals",
     y = "Sales Residuals", title = "Adjusted Variable Plot",
     subtitle = subtitletxt) + hw

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

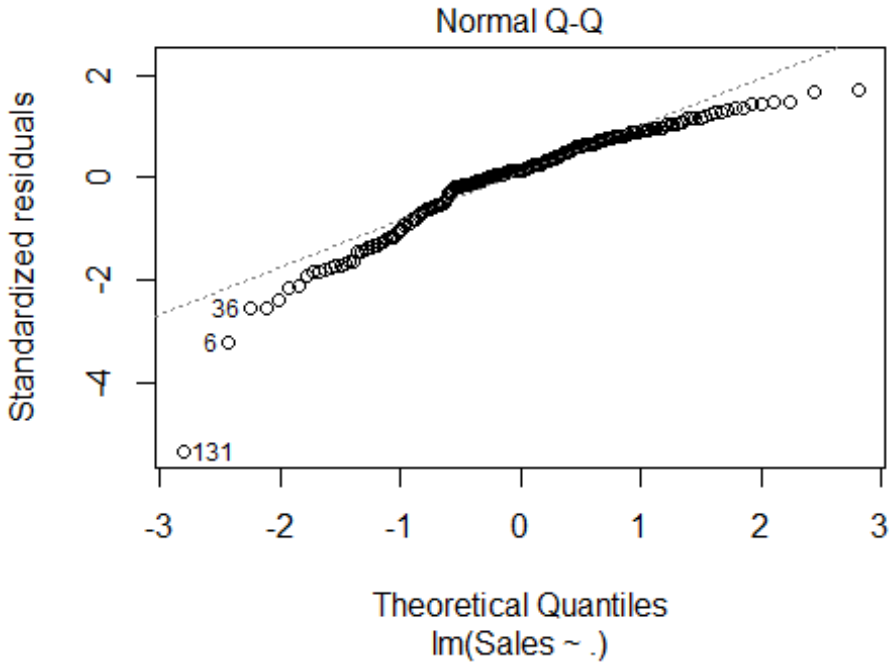


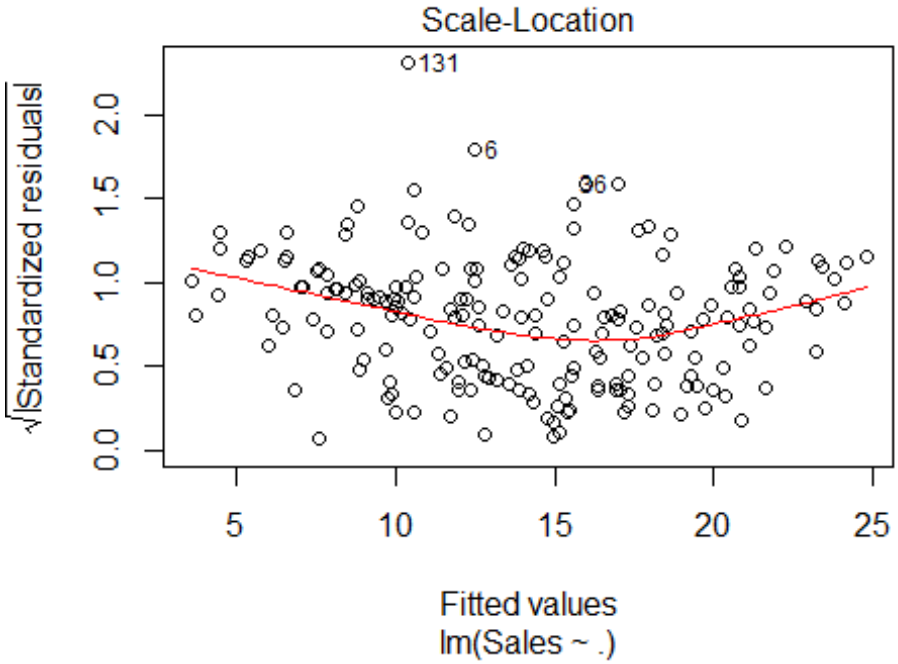
The unique contribution of the Newspaper budget to model domain has no relationship to Sales residual from fitting a TV+Radio model.

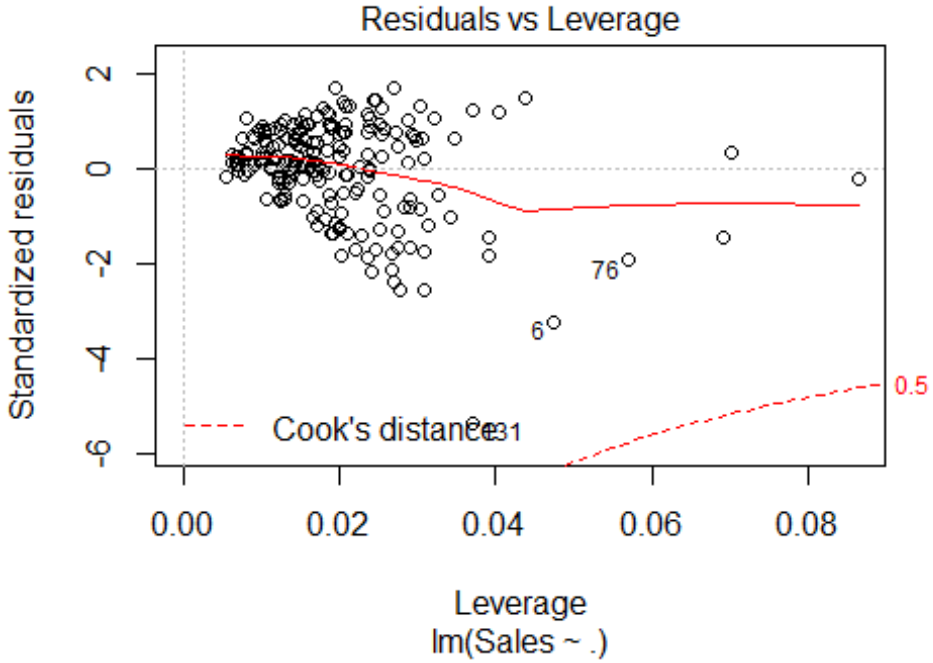
5.1 Regression diagnostics plots

```
plot(adModel1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Sales ~ .)
179
6
131


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ .)
36
6
131





As indicated in the console, after clicking in the console, hit Return to see the next plot in the set of 4 regression, diagnostics plots.

Residual versus fitted values plot To match the assumption that the model errors are independent and identically distributed normal random variables residuals centered roughly centers about the line y = 0. The red line smooth of the residuals around shows curvature. This violates the model assumption need justify making statistical inference model and it coefficients.

In this plot, the points numbered 131, 6 and 179 are low value outliers. Removing them will likely reduce the bend in the red line. We might consider deleting the points if we have a good reason to think that one or more of their value are flawed. Of course then we should also wonder there other flawed values are that don't stand out as outliers.

When residuals are plotted against a variable and the smooth looks like a parabola, including the square of variable's values in the model may yield a better fitting model. Here any variable highly correlated to the fitted values will likely be helpful.

Plot 2: the Normal Q-Q plot. We see the outliers and a thick left tail. That is, points on the left are far below the reference line. The residuals do have an approximately normal distribution. Statistical inference (hypothesis tests and confidence intervals) for the model as a whole and for the individual terms are not justified.

The right tail is thin. The right-side points are on the center-of-the-plot side of the reference line. Thin tails are of less concerned in linear regression.

Plot 3, the scale-location plot The y-axis is the square root of the absolute standardized residuals. The regression residuals have covariance matrix that is based on the design matrix. In general the correlations are ignorable. The variances are not. Standardized residual have been divide by their estimated standard deviations.

The absolute value transformation puts all the negative residuals on the positive side of the zero.

The square-root transformation helps balance the small and large absolute residuals. The red smooth line should be y = 1. The curved line means the residual don't have the same variance. Our independent identically distributed errors assumption fails in terms of the mean (Plot 1) and the variance (Plot 3)

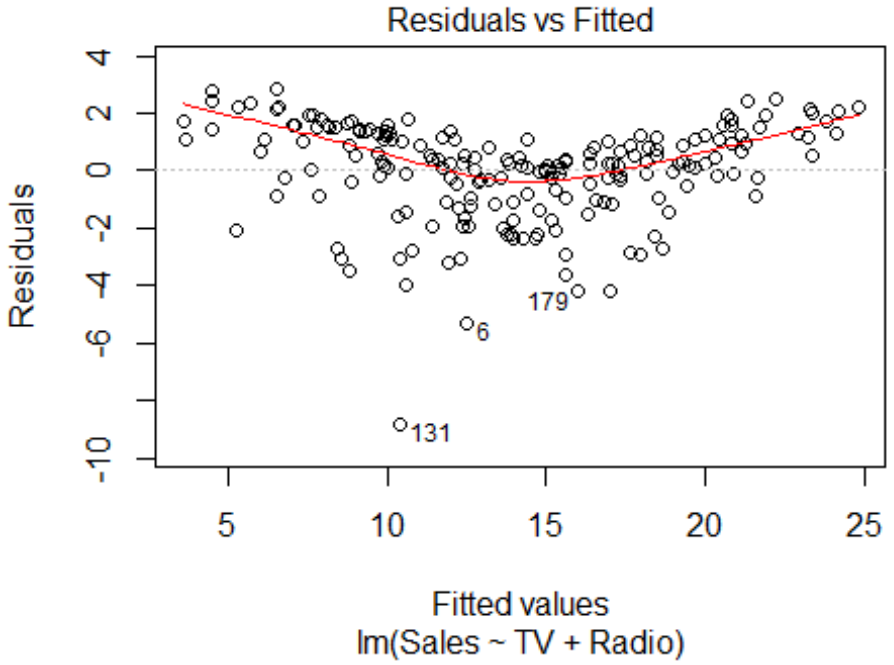
Plot 4 Standardized Residuals versus Leverage

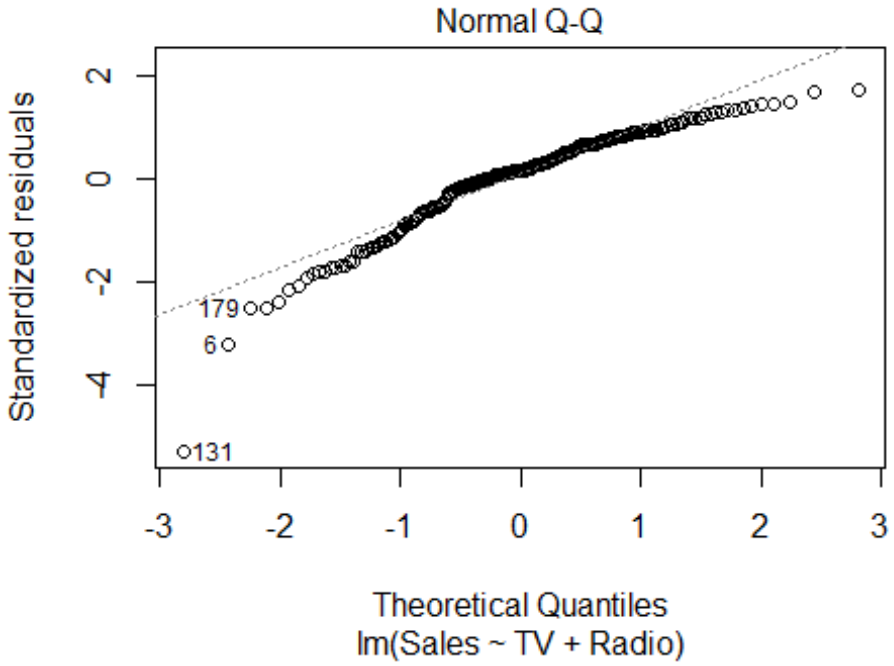
The high leverage points are on the far right. The influence of a depends on its leverage and have far it would be from the regression line if it were omitted. Points 131 and 6 are high influence points. They have substantial leverage and large standardized residuals.

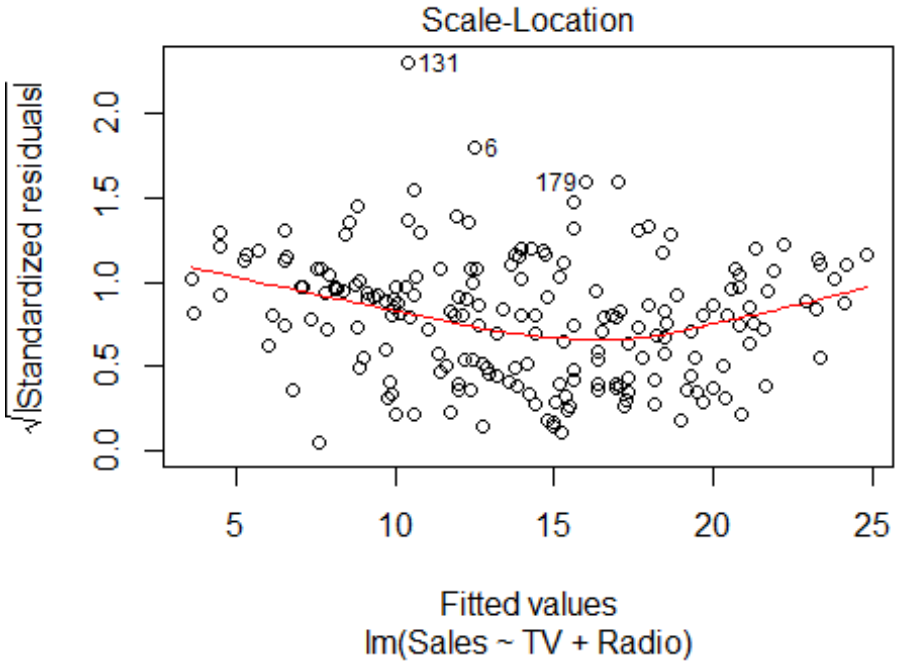
5.2 A TV and Radio model

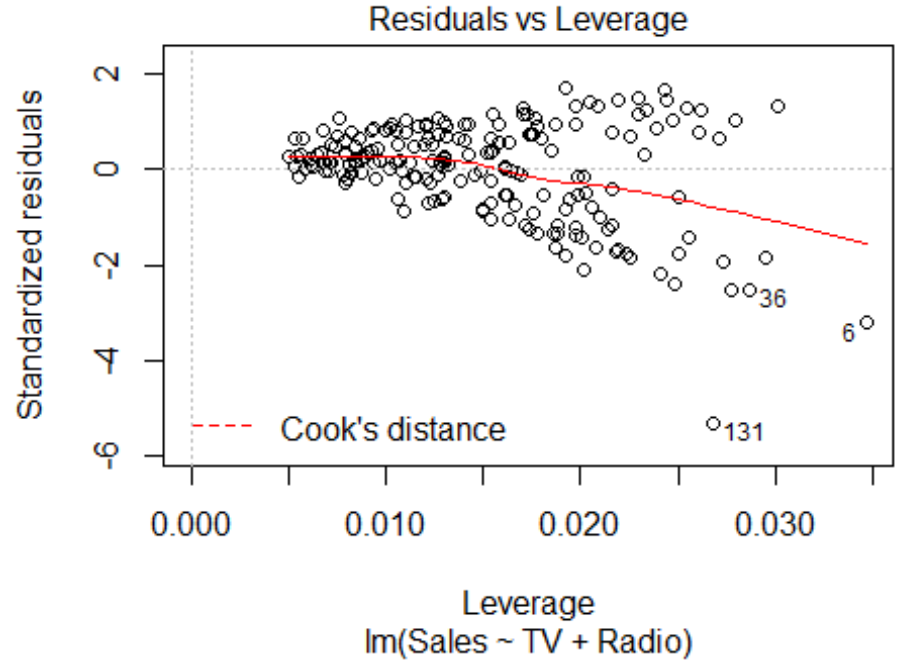
```
adModel2 <- lm(Sales~TV+Radio, data = adSales)
summary(adModel2)

##
## Call:
## lm(formula = Sales ~ TV + Radio, data = adSales)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -8.7977 -0.8752  0.2422  1.1708  2.8328 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.92110    0.29449   9.919   <2e-16 ***
## TV           0.04575    0.00139  32.909   <2e-16 ***
## Radio        0.18799    0.00804  23.382   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 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

plot(adModel2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Sales ~ TV + Radio)
179
6
131


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ TV + Radio)
179
6
131





Dropping the Newspape term didn't change much.

5.3 Adding an interaction terms for TV and Radio

In the R linear model syntax TV:Radio is an interaction term This multiples the TV and Radio vectors and includes the resulting vector in the model.

TV*Radio is interpreted as TV + Radio + TV:Radio

There are three different ways to specify the same model

This result is the same

```
adModel3a <- lm(Sales~ TV + Radio + TV:Radio, adSales)
summary(adModel3a)

##
## Call:
## lm(formula = Sales ~ TV + Radio + TV:Radio, data = adSales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3366 -0.4028  0.1831  0.5948  1.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.750e+00  2.479e-01  27.233   <2e-16 ***
## TV          1.910e-02  1.504e-03  12.699   <2e-16 ***
## Radio       2.886e-02  8.905e-03   3.241   0.0014 **
## TV:Radio    1.086e-03  5.242e-05  20.727   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9435 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

Results of direct mathematical operations on variables need to be surrounded by I() This result is the same adModel3b <- lm(Sales~TV + Radio + I(TV*Radio), adSales) summary(adModel3b)

We can look at a 3D scatterplot of Sales versus Radio and TV budgets. With the plot rotated so is Sales is pointing upward we can see noticeable downward bend when TV Sales is high and Radio sales is low. A few low points wtih TV Sales is very low and and Radio sales is high. The two variables have a positive interaction.

Noet: With rgl we could make all 3d plots like those in the ILSR text.

```
# Omit Newspaper and show Sales
domain <- as.matrix(adSales[, -3])
open3d(FOV = 0)  # no perspective projection

## wgl
##   2
```

```
fixLights()
plot3d(
  domain,
  type = "s",
  radius = 3.0,
  col = rgb(1, .2, .2),
  aspect = TRUE
)
```

5.4 Adding a square term

We can square the TV budget vector and include it in the model

```
adModel4 <- lm(Sales~ TV*Radio+I(TV^2), adSales)
summary(adModel4)
```

```
##
## Call:
## lm(formula = Sales ~ TV * Radio + I(TV^2), data = adSales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9949 -0.2969 -0.0066  0.3798  1.1686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.137e+00  1.927e-01  26.663  < 2e-16 ***
## TV           5.092e-02  2.232e-03  22.810  < 2e-16 ***
## Radio        3.516e-02  5.901e-03   5.959 1.17e-08 ***
## I(TV^2)     -1.097e-04  6.893e-06 -15.920  < 2e-16 ***
## TV:Radio     1.077e-03  3.466e-05  31.061  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6238 on 195 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.9857
## F-statistic:  3432 on 4 and 195 DF,  p-value: < 2.2e-16
```

There is almost no variance left to explain. Is this data real or was in generated with two outliers included.

The two outliers are still present on the left but the curvature in the residuals has been reduced.

5.5 Specifying a quadratic response surface

The polym() function can be used specify quadratic response surface.

```
adModel5 <- lm(Sales~polym(TV,Radio,degree=2),adSales)
summary(adModel5)
```

```
##
## Call:
## lm(formula = Sales ~ polym(TV, Radio, degree = 2), data = adSales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0027 -0.2859 -0.0062  0.3829  1.2100
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      13.94780    0.04422 315.449   <2e-16 ***
## polym(TV, Radio, degree = 2)1.0  53.71298    0.62764  85.579   <2e-16 ***
## polym(TV, Radio, degree = 2)2.0  -9.99022    0.62778 -15.914   <2e-16 ***
## polym(TV, Radio, degree = 2)0.1  40.52042    0.62857  64.464   <2e-16 ***
## polym(TV, Radio, degree = 2)1.1 272.58577    8.82370  30.892   <2e-16 ***
## polym(TV, Radio, degree = 2)0.2   0.49390    0.62608   0.789    0.431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6244 on 194 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.9857
## F-statistic:  2740 on 5 and 194 DF,  p-value: < 2.2e-16
```

6. The Mean Squared Error

The Q-Q plot discourages us from makingclaims based on test statistics p-values because the residual distribution does not support distribution assumptions upon which they are made.

However, that does prohibit us from comparing the accuracy linear regression models to each other or to other models.

We often compare models based on criteria such as the mean squared error.

```
MSEmodel4 <- mean((adSales$Sales - fitted(adModel4))^2)
MSEmodel4
```

```
## [1] 0.3793544
```

We are interested in the accuracy of predictions that we obtain when we apply our method to previously unseen test data. In the context of new data, our data set becomes a training data set and our MSE is a training set MSE.

We expect the training data “sample” to differ somewhat from the population of possible future data sets assessing the same phenomena. We may overfit details of the training set that are atypical or simply random variation.