Regression session II: multiple linear regression

Warren W. Kretzschmar

2019-05-22

Contents

These lecture notes are based on and closely follow section 3.2 in *An Introduction to Statistical Learning, with applications in R* by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013, DOI: 10.1007/978-1-4614-7138-7).

1 Learning outcomes

After this session, a student should be able to:

- visualize bivariate relationships
- fit a linear regression model containing main effects
- assess the quality of the model fit
- determine if at least one predictor can predict the response
- detaermine which predictors predict the response
- asses the accuracy of predictions from the model

2 Warmup: Quiz: revisiting linear model specifications

3 Visualizing the Advertising dataset

In this session we will use the Advertising dataset. This simple dataset consists of sales data for 200 products along with the amount of money spent on TV, radio, and newspaper ads. We would like to know how best to spend advertizing money to maximize sales.

To begin with, let us familiarize ourselves with the dataset. The data are stored in the `data` subdirectory of this session directory.

```
# load the data
ads = read.csv('./data/Advertising.csv')
```

First, we check to see what columns were imported

```
head(ads)
```

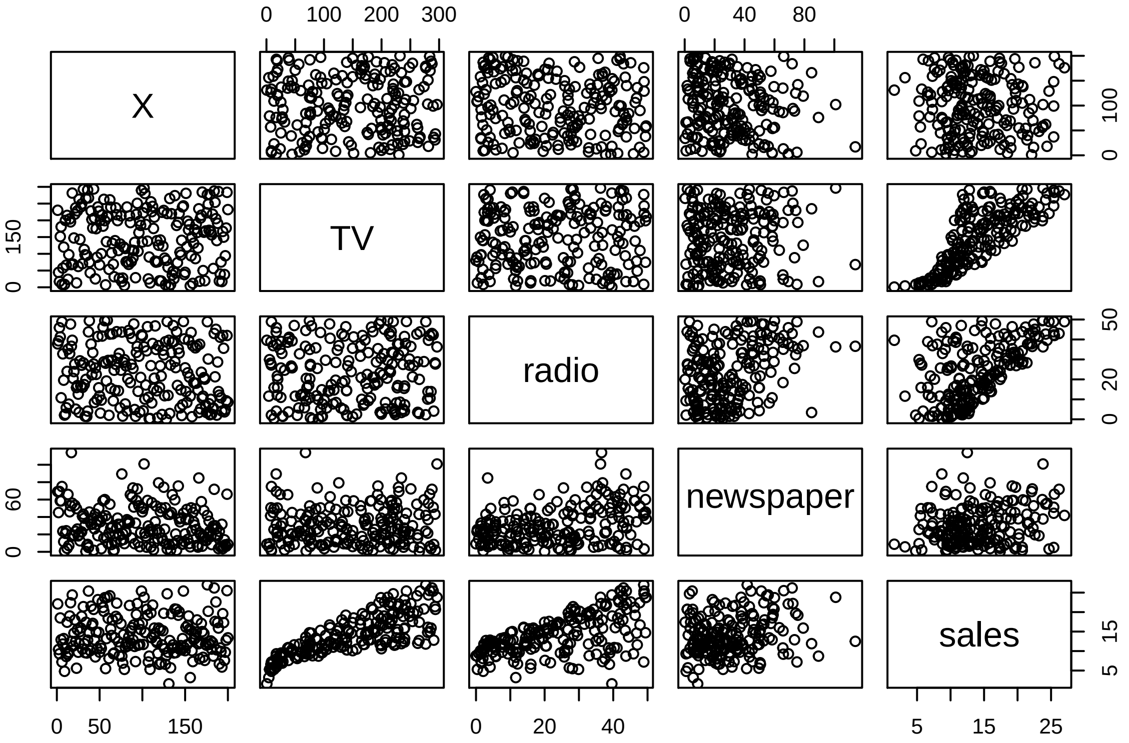
```
##   X    TV radio newspaper sales
## 1 1 230.1  37.8      69.2  22.1
## 2 2  44.5  39.3      45.1  10.4
## 3 3  17.2  45.9      69.3   9.3
## 4 4 151.5  41.3      58.5  18.5
## 5 5 180.8  10.8      58.4  12.9
## 6 6   8.7  48.9      75.0   7.2
```

It looks like a redundant column of row numbers, `X`, has made it into the table.

The other columns look like numbers. That's good.

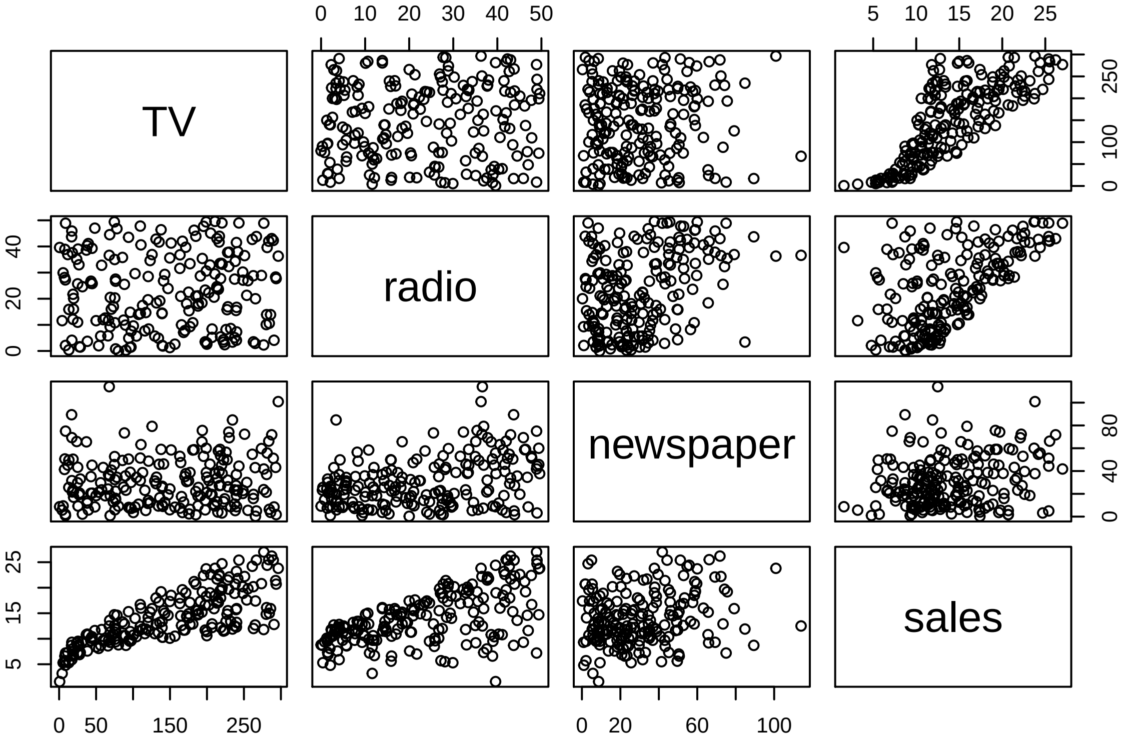
Now, let us use the `pairs` function to get a quick overview of linear relationships within the dataset.

```
# visualize all pairwise relationships
pairs(ads)
```



The pairs plot creates a scatter plot of every pair of variables in a data frame. First, to clear things up, the variable `X` is uncorrelated with the other columns and does not add anything to the dataset. We should probably remove it and replot:

```
ads = ads[-1]
pairs(ads)
```



Ah, that's better.

From the pairs plot we can see that:

1. TV expenditure appears to be correlated with sales
2. As TV expenditure goes up, the variance associated with sales increases as well
3. radio expenditure appears to be correlated with sales
4. newspaper sales do not look very correlated to sales

It looks like more than one variable could be used to predict sales. How would we handle this in the simple regression?

3.1 Excercise: fitting simple linear regressions on multivariate data

We can fit a simple linear regression for TV vs Sales this way:

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

```
## Error in eval(predvars, data, env): object 'Sales' not found
```

Oops that did not work! Let's see what's up:

```
names(ads)
```

```
## [1] "TV"        "radio"     "newspaper" "sales"
```

Ah! sales is lower case:

```
lm(sales ~ TV, data=ads)
```

```
##
## Call:
## lm(formula = sales ~ TV, data = ads)
##
```

```
## Coefficients:
## (Intercept)           TV
##     7.03259      0.04754
```

For every $1000 spent on TV ads, our average sales went up by five units. Pretty sweet! What about radio?

```
lm(sales ~ radio, data=ads)
```

```
##
## Call:
## lm(formula = sales ~ radio, data = ads)
##
## Coefficients:
## (Intercept)        radio
##      9.3116       0.2025
```

radio appears to help even more!

```
lm(sales ~ newspaper, data=ads)
```

```
##
## Call:
## lm(formula = sales ~ newspaper, data = ads)
##
## Coefficients:
## (Intercept)    newspaper
##    12.35141      0.05469
```

newspaper appears to have a similar effect to TV. But there seemed to be much more noise between sales and newspaper in the pairs plot. What's going on?

```
summary(lm(sales ~ TV, data=ads))
```

```
##
## Call:
## lm(formula = sales ~ TV, data = ads)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
summary(lm(sales ~ newspaper, data=ads))
```

```
##
## Call:
## lm(formula = sales ~ newspaper, data = ads)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2272  -3.3873  -0.8392   3.5059  12.7751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.35141    0.62142   19.88  < 2e-16 ***
## newspaper    0.05469    0.01658    3.30  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.092 on 198 degrees of freedom
## Multiple R-squared:  0.05212,    Adjusted R-squared:  0.04733
## F-statistic: 10.89 on 1 and 198 DF,  p-value: 0.001148
```

The answer is in the R^2 values: TV has a much higher R^2 than newspaper.

We can fit a linear model for each predictor separately, but we are left with two problems:

1. How would be combine the three models into a single model to create a single prediction for Sales?
2. The pairs plot shows us that the predictors are correlated. The simple linear regression fits ignore all other predictors, and this can lead to incorrect predictions.

This is where multiple linear regression comes in. It allows us to create a single model for predicting sales from multiple predictors, and it allows us to create a model that takes correlation between predictors into account.

4 Multiple linear regression model specifications

A multiple linear regression model that incorporates p predictors can be expressed as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_p + \epsilon$$

where Y is the response variable, X_j is the j^{th} predictor, and β_j is the j^{th} model coefficient. β_j can be interpreted as the average increase in Y for one unit increase in X_j while holding all other predictors fixed.

For the `Advertising` dataset we can express a regression model as:

$$Sales = \beta_0 + \beta_1 newspaper + \beta_2 radio + \beta_3 TV + \epsilon$$

5 Estimating regression coefficients

We can estimate the regression coefficients $\hat{\beta}_j$ from the data in the same manner as for simple linear regression. We can then use $\hat{\beta}_j$ to make predictions $\hat{y}$ using the formula:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p$$

As in simple linear regression, we choose β_j such that we minimize the residual sum of squares:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

, which is equivalent to

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - ... - \hat{\beta}_p x_{ip})^2$$

The mathematical formulas for estimating the model coefficients in multiple linear regression work similarly to the formulas for simple linear regression. However, they are more complex and require some linear algebra to understand, so we will skip those formulas here.

5.1 Quiz: What was y_i again?

5.2 Exercise: Fitting a multiple linear regression model

Above, we fit a simple linear regression model to the Advertising dataset. We had to fit each predictor separately. Here we will fit a joint model:

```
summary(lm(sales ~ TV + newspaper + radio, data=ads))
```

```
##
## Call:
## lm(formula = sales ~ TV + newspaper + radio, data = ads)
##
## 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 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## radio        0.188530   0.008611  21.893   <2e-16 ***
## ---
## 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
```

We note that the coefficient estimates have changed! TV looks similar to what we saw in simple linear regression, but newspaper and radio have both decreased. This is expected when predictors are correlated. The multiple linear regression fit takes the correlation between predictors into account. This fit states that once we take TV and radio into account, newspaper ads do not increase sales.

6 Questions we can answer with a multiple linear regression

Now that we have a better sense of what a multiple linear regression is, let us focus on the questions we might want to answer with with the help of such a regression:

1. Can any of the predictors predict response?
2. If so, to what degree are the predictors important?
3. How well does the model explain the data?
4. What is the model's prediction accuracy?

6.1 Can any of the predictors predict response?

The hypothesis that no predictor predicts response can be coded as the null hypothesis

$$H_0 : \beta_0 = \beta_1 = ... = \beta_p = 0$$

The alternate hypothesis is then

$$H_A : \text{at least one } \beta_j \text{ is not } 0$$

We can use the F statistic to test the null hypothesis. The F-test (named in honor of R. A. Fisher) uses the ratio:

$$F = \frac{\text{explained variance}}{\text{unexplained variance}}$$

In a model in which our assumptions hold, and for which our null hypothesis is true, we expect the explained and unexplained variances to be equal. Therefore, under the null hypothesis, the F-statistic should be around 1.

We can estimate the variance explained by our model as

$$\frac{TSS - RSS}{p}$$

For any model, as we increase the number of predictors the amount of variance that we explain with our model increases. The denominator, p, corrects for this.

We can estimate the unexplained variance as

$$\frac{RSS}{(n - p - 1)}$$

As with the explained variance above, p in this denominator corrects for how unexplained variance decreases as we add more predictors. We can also see that as our sample size, n, increases, our RSS increases as well.

This leads to the following equation for calculating the F-statistic for a linear model:

$$F = \frac{(TSS - RSS)}{p} \bigg/ \frac{RSS}{(n - p - 1)}$$

If our null hypothesis is not true, then, assuming that the TSS remains constant, we can expect the RSS to become smaller compared to the TSS, and for the ratio to increase. Therefore, the larger the F-statistic, the more evidence there is for rejecting the null hypothesis.

`R` will happily calculate the F-statistic of a linear model fit for us. One way to get the F statistic is using the `summary()` function:

```
summary(lm(sales ~ TV + newspaper + radio, data=ads))
```

```
##
## Call:
## lm(formula = sales ~ TV + newspaper + radio, data = ads)
##
## 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 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## radio        0.188530   0.008611  21.893   <2e-16 ***
## ---
## 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
```

In this case, the F-statistic is 570. That looks like a lot, and the low p-value confirms that it is.

6.1.1 Excercise: playing around with n

We can get an idea of how sample size changes the F-statistic and the p-value associated with the statistic.

We can use `sample()` to reduce the sample size:

```
?sample
```

We want to be sure to sample without replacement, and the help page tells us that this is the default

```
summary(lm(sales ~ TV + newspaper + radio, data=ads[sample(nrow(ads), 200),]))
```

```
##
## Call:
## lm(formula = sales ~ TV + newspaper + radio, data = ads[sample(nrow(ads),
##     200), ])
##
## 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 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## radio        0.188530   0.008611  21.893   <2e-16 ***
## ---
## 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
```

```
summary(lm(sales ~ TV + newspaper + radio, data=ads[sample(nrow(ads), 50),]))
```

```
##
## Call:
```

```
## lm(formula = sales ~ TV + newspaper + radio, data = ads[sample(nrow(ads),
##     50), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6372 -0.3573  0.3836  1.3161  2.2505
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.156855   0.739660   2.916  0.00546 **
## TV          0.050326   0.003186  15.796  < 2e-16 ***
## newspaper   0.006358   0.011118   0.572  0.57022
## radio       0.176510   0.019457   9.072 8.21e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.911 on 46 degrees of freedom
## Multiple R-squared:  0.8867, Adjusted R-squared:  0.8793
## F-statistic:   120 on 3 and 46 DF,  p-value: < 2.2e-16
```

```
summary(lm(sales ~ TV + newspaper + radio, data=ads[sample(nrow(ads), 10),]))
```

```
##
## Call:
## lm(formula = sales ~ TV + newspaper + radio, data = ads[sample(nrow(ads),
##     10), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89569 -0.22713  0.08619  0.68410  3.10750
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.157084   2.968610  -0.390  0.71016
## TV           0.053850   0.007841   6.867  0.00047 ***
## newspaper   -0.027562   0.046557  -0.592  0.57547
## radio        0.298904   0.103261   2.895  0.02753 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.02 on 6 degrees of freedom
## Multiple R-squared:  0.8883, Adjusted R-squared:  0.8325
## F-statistic: 15.91 on 3 and 6 DF,  p-value: 0.002915
```

It looks like the signal is very strong, and we can still reject our null hypothesis with 10 data points.

The model summaries have p-values next to each model coefficient. These p-values are based on the t-statistic and are calculated in the same way as for simple linear regression (see previous section). Why can't we just use these p-values instead of the F-statistic to determine if any coefficients are non-zero? Without correction for multiple testing we will get false positives at the probability of α per test. If we do correct for multiple testing, then the F-test is more powerful and/or requires fewer assumptions about the correlation structure of the t-statistics.

6.2 Which predictors predict response?

Usually, we will not want to include all possible predictors in our model. But how can we choose the predictors? We could use the t-test, but then we will have to deal with false positives. The process of choosing predictors in this way is called *variable selection*. There is a substantial body of literature on variable selection in linear models.

Out of all possible models, we would like to choose the one that is closest to the true model. We face two problems in this endeavor:

1. Determining which model fits best while correcting for multiple testing and overfitting
2. The number of models that can be created from a subset of p predictors is 2^p, and this number can get large very quickly.

For point one there are several statistics that can be used. These statistics include *Mallow's C*, *Akaike's Information Criterion* (AIC), and the *Bayesian Information Criterion* (BIC). These statistics balance the number of predictors used against the amount of variance explained by a model.

For point two we can use a step-wise selection process in which we add, remove, or add and remove predictors in order to minimize an information criterion. This approach does not explore the full model space, but in practice it can be quite useful.

6.2.1 Exercise: Let's select predictors in a model

The `stepAIC()` function can be used to perform step-wise model selection in `R`. By default it uses AIC, but other statistics are also supported.

We can start with the full model:

```r
library(MASS)
full = lm(sales ~ TV + newspaper + radio, data=ads)
summary(stepAIC(full))
```

```
## Start:  AIC=212.79
## sales ~ TV + newspaper + radio
##
##               Df Sum of Sq    RSS    AIC
## - newspaper  1      0.09  556.9 210.82
## <none>                     556.8 212.79
## - radio      1   1361.74 1918.6 458.20
## - TV         1   3058.01 3614.8 584.90
##
## Step:  AIC=210.82
## sales ~ TV + radio
##
##         Df Sum of Sq    RSS    AIC
## <none>                556.9 210.82
## - radio  1    1545.6 2102.5 474.52
## - TV     1    3061.6 3618.5 583.10
```

```
##
## Call:
## lm(formula = sales ~ TV + radio, data = ads)
##
## 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
```

This returns a model in which newspaper (unsurprisingly) has been removed.

Or we can start with an empty model:

```
empty = lm(sales ~ 1, data=ads)
summary(stepAIC(empty, scope=sales ~ TV + newspaper + radio))
```

```
## Start:  AIC=661.8
## sales ~ 1
##
##             Df Sum of Sq    RSS    AIC
## + TV         1    3314.6 2102.5 474.52
## + radio      1    1798.7 3618.5 583.10
## + newspaper  1     282.3 5134.8 653.10
## <none>                   5417.1 661.80
##
## Step:  AIC=474.52
## sales ~ TV
##
##             Df Sum of Sq    RSS    AIC
## + radio      1    1545.6  556.9 210.82
## + newspaper  1     184.0 1918.6 458.20
## <none>                   2102.5 474.52
## - TV         1    3314.6 5417.1 661.80
##
## Step:  AIC=210.82
## sales ~ TV + radio
##
##             Df Sum of Sq    RSS    AIC
## <none>                    556.9 210.82
## + newspaper  1      0.09  556.8 212.79
## - radio      1   1545.62 2102.5 474.52
## - TV         1   3061.57 3618.5 583.10
```

```
##
## Call:
## lm(formula = sales ~ TV + radio, data = ads)
##
## 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
```

And in this case we get the same model as above.

6.3 How well does the model explain the data?

As in simple linear regression, two important statistics for assessing the quality of the model fit are R^2 and RSE.

6.3.1 R^2

- In simple linear linear regression: $R^2 = \text{Cor}(X, Y)^2$
- In multiple linear regression: $R^2 = \text{Cor}(Y, \hat{Y})$

Let's see what the R^2 of our models look like:

```
summary(lm(sales ~ TV , data=ads))$r.squared
```

```
## [1] 0.6118751
```

```
summary(lm(sales ~ TV + radio, data=ads))$r.squared
```

```
## [1] 0.8971943
```

```
summary(lm(sales ~ TV + radio + newspaper, data=ads))$r.squared
```

```
## [1] 0.8972106
```

As we add predictors R^2 always increases. When the predictors explain no variance, then $R^2 = 0$. When they explain all variance, then $R^2 = 1$.

6.3.2 RSE

- In simple linear regression we learned: RSE $= \sqrt{RSS/(n-2)}$
- In multiple linear regression: RSE $= \sqrt{RSS/(n-p-1)}$

Let's see what the RSE of our models look like:

```
summary(lm(sales ~ TV , data=ads))$sigma
```

```
## [1] 3.258656
```

```
summary(lm(sales ~ TV + radio, data=ads))$sigma
```

```
## [1] 1.681361
```

```
summary(lm(sales ~ TV + radio + newspaper, data=ads))$sigma
```

```
## [1] 1.68551
```

- The model that includes `newspaper` has a higher RSE than the model with only `TV` and `radio`
- This is because RSE depends on p

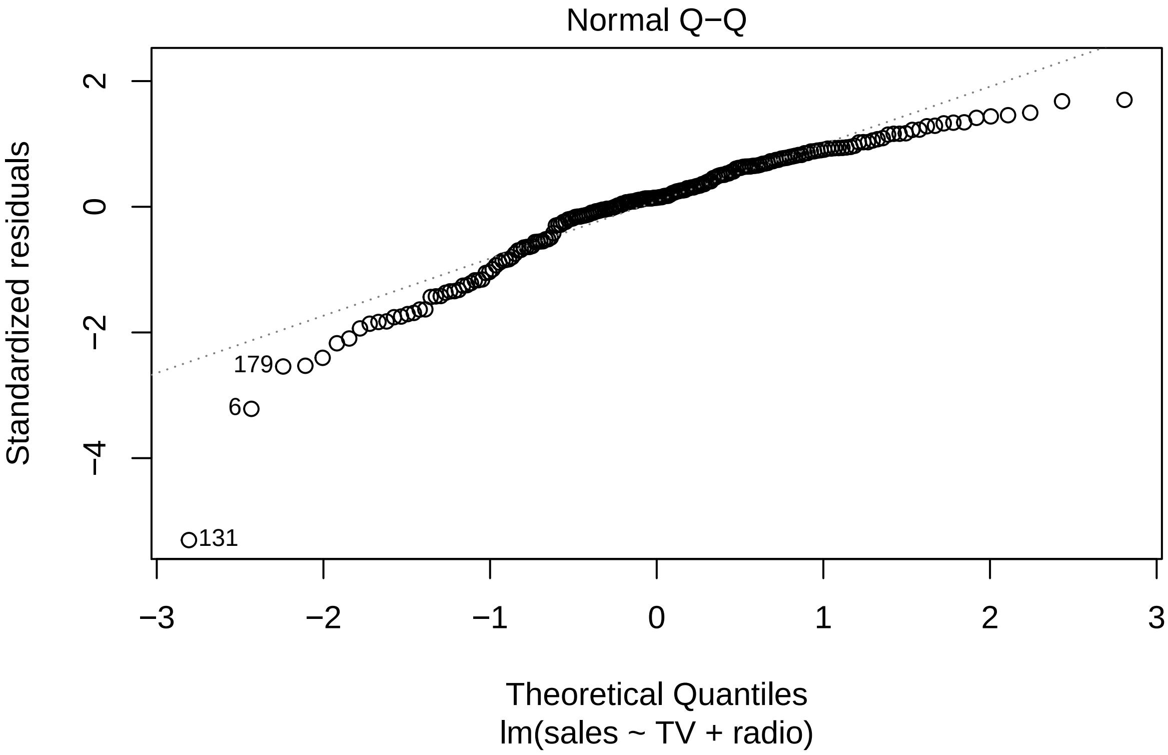
6.3.3 Visual inspection

- Plotting the data can reveal problems that are not evident from metrics

The model that includes `TV` and `radio` has some issues. Let's see if we can discover them by visualization.

Let's start with my favorite quality assessment plot: the quantiles plot:

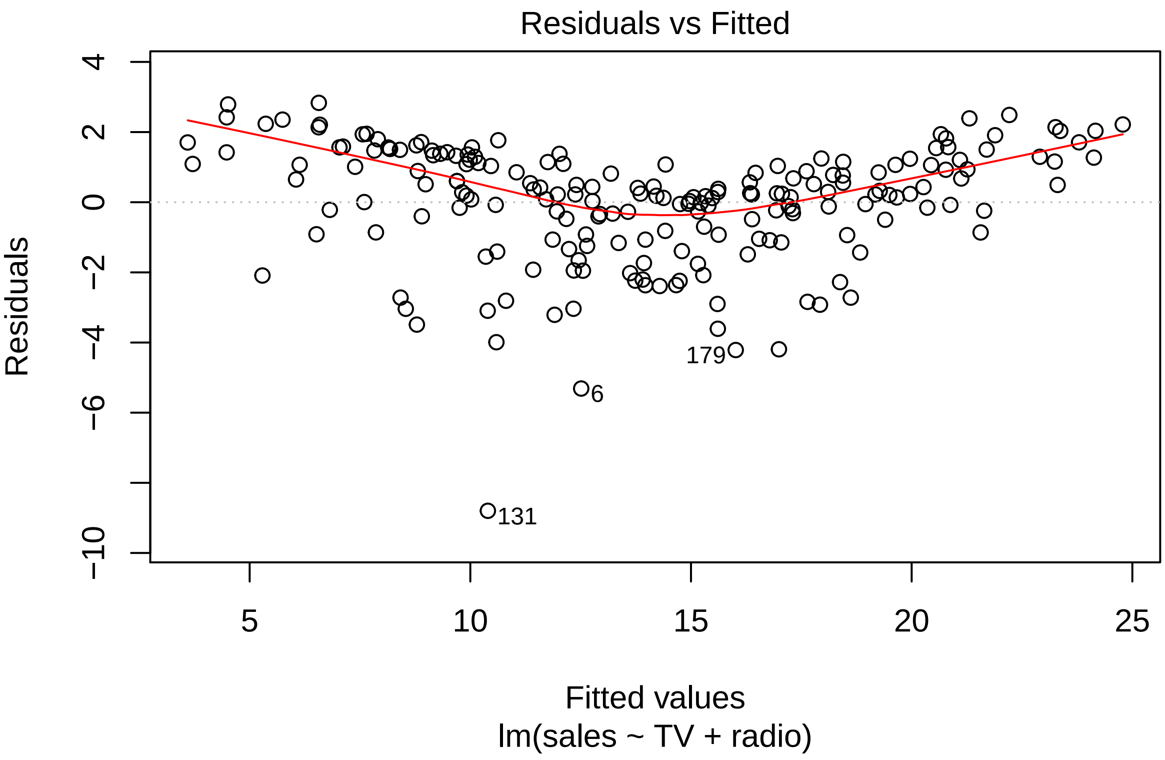
```
fit = lm(sales ~ TV + radio, data=ads)
plot(fit, which=2)
```



This is not what I would expect normally distributed residuals to look like!

Let's see if we can get a better idea of where things are going wrong. Let's try and plot $\hat{Y}$ (fitted values) versus the residuals:

```
# NB: predict()
plot(fit, which=1)
```

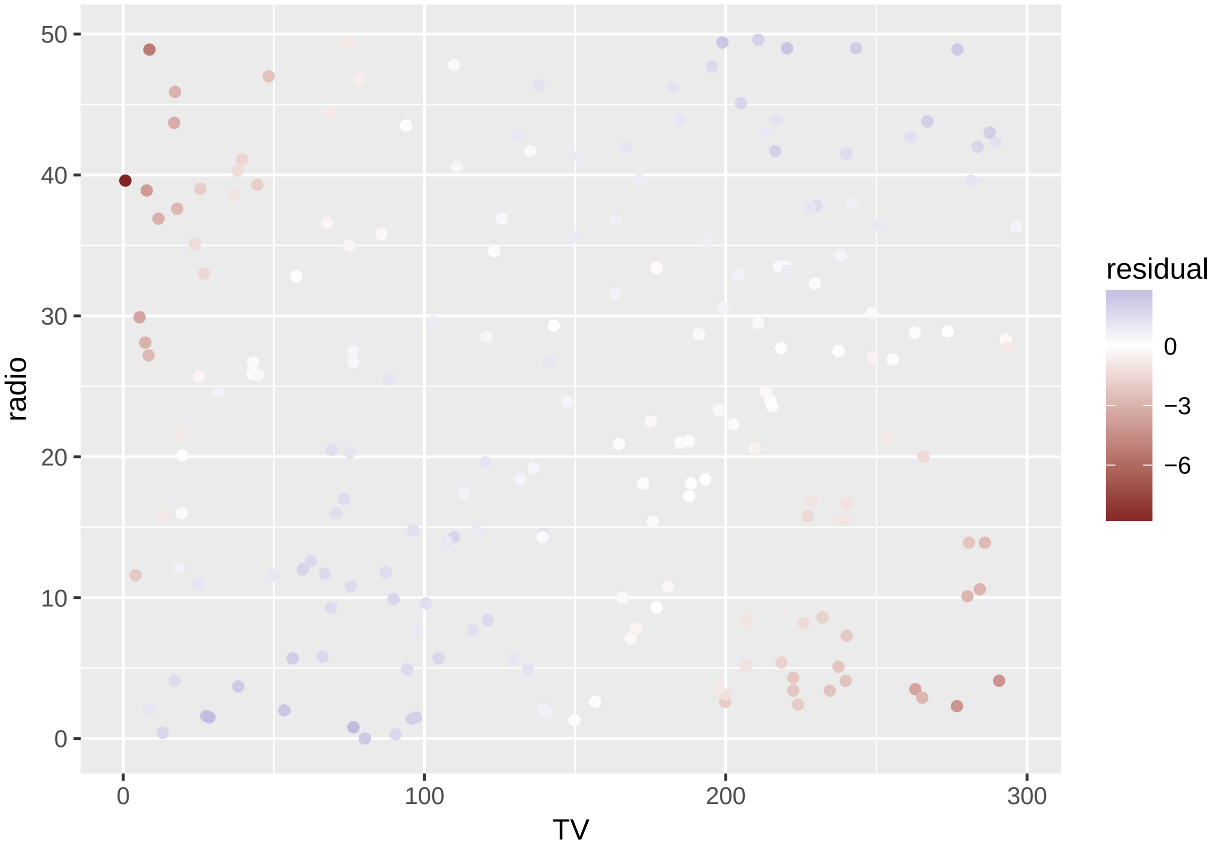


We see a dip in residuals and an increase in variance towards the middle of the range of fitted values. It looks like our model is not adequate.

In fact, this model fails to capture a synergy between TV and radio advertizing:

```r
# Here we rely on resid() returning the residuals in the same order as the ads data.frame
plot_dat = data.frame(TV=ads$TV, radio=ads$radio, residual=resid(fit))

library(ggplot2)
ggplot(aes(x=TV, y=radio, color=residual), data=plot_dat) + geom_point() +
  scale_color_gradient2()
```



- The model overestimates sales generated from investment in a single ads platform (top left and bottom right)
- The model underestimates sales generated from investment in both add platforms (top right and bottom left)

We can model this synergy as an "interaction term". Unfortunately, interaction terms are beyond the scope of this session. See further reading for more on interaction terms.

6.4 What is the model's prediction accuracy?

- Previously: we can make predictions from our model using the `predict()` function.

```
# making a prediction on a new data point
predict(fit, newdata=data.frame(TV=100, radio=30))
```

```
##        1
## 13.13641
```

There are three sources of error when making predictions from a linear model

1. *reducible error*: A result of the difference between the estimates

   $$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + ... + \hat{\beta}_p X_p$$

   and the *true population regression plane*

   $$f(x) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

   We use *confidence intervals* to estimate this error.

2. *model bias*: When our model differs from the true model (see for example the previous section)
3. *irreducible error*: The random noise that is part of our system, ϵ. We can use *prediction intervals* to estimate this error.

If we assume that we have the correct model, then we can ask two kinds of questions:

1. How far is $\hat{Y}$ from $f(x)$?
    - We use *confidence intervals* to talk about how our estimate of average sales will differ from the true average of sales
2. How far is any one prediction from its true value?
    - For this, we use *prediction intervals*

Prediction intervals are always larger than confidence intervals because prediction intervals quantify both the reducible and irreducible error.

Let's try and calculate the confidence and prediction intervals around $\hat{Y}$ of our (dubious) model fit:

```
preds = as.data.frame(predict(fit, interval="confidence"))
head(preds)
```

```
##        fit      lwr      upr
## 1 20.55546 20.16278 20.94815
## 2 12.34536 11.89093 12.79979
## 3 12.33702 11.76734 12.90670
## 4 17.61712 17.24763 17.98660
## 5 13.22391 12.90049 13.54733
## 6 12.51208 11.89485 13.12932
```

```
preds2 = as.data.frame(predict(fit, interval="prediction"))
```

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

```
head(preds2)
```

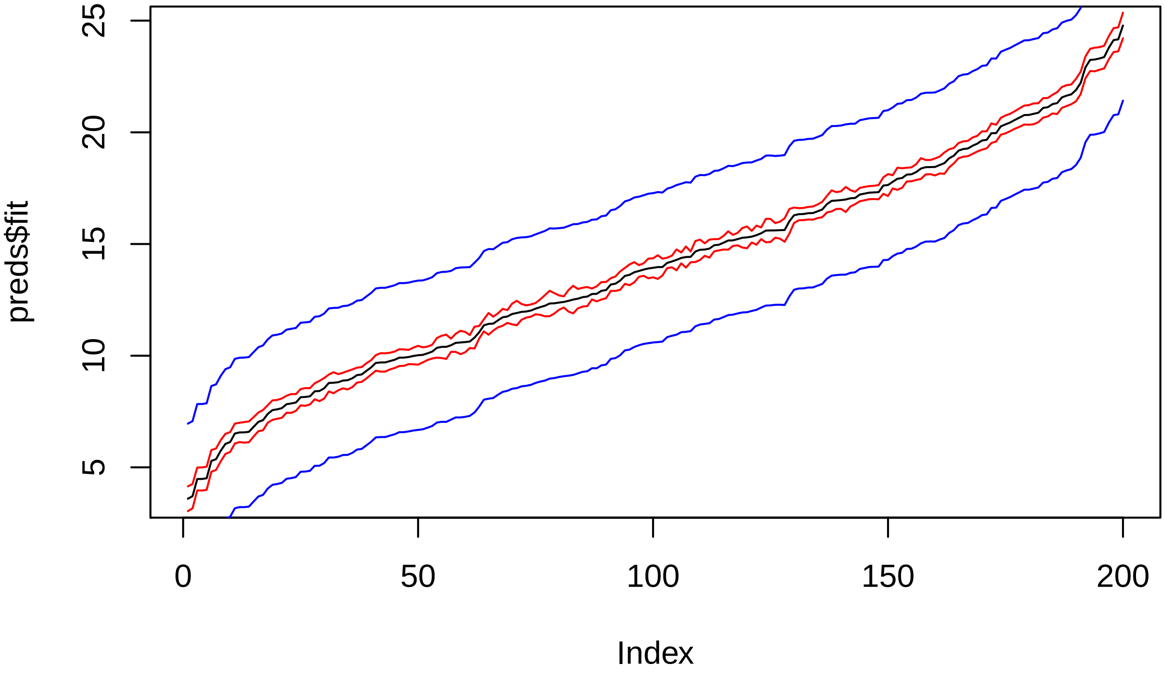
```
##        fit       lwr      upr
## 1 20.55546 17.216516 23.89441
## 2 12.34536  8.998591 15.69213
## 3 12.33702  8.972659 15.70138
## 4 17.61712 14.280817 20.95341
## 5 13.22391  9.892396 16.55542
## 6 12.51208  9.139348 15.88482
```

```
preds$lwr_pred = preds2$lwr
preds$upr_pred = preds2$upr

preds = preds[order(preds$fit),]
preds$index = 1:nrow(preds)
head(preds)
```

```
##          fit      lwr      upr  lwr_pred upr_pred index
## 109 3.595686 3.041903 4.149468 0.2339824 6.957389     1
## 9   3.709379 3.163756 4.255002 0.3490103 7.069748     2
## 193 4.478859 3.966817 4.990901 1.1237791 7.833939     3
## 77  4.480148 3.962410 4.997886 1.1241941 7.836102     4
## 92  4.511679 3.994733 5.028625 1.1558471 7.867511     5
## 156 5.289428 4.804825 5.774031 1.9384257 8.640430     6
```

```
plot(preds$fit, type='l')
lines(preds$index, preds$upr, col='red')
lines(preds$index, preds$lwr, col='red')
lines(preds$index, preds$upr_pred, col='blue')
lines(preds$index, preds$lwr_pred, col='blue')
```



We can see that the prediction intervals are larger than the confidence intervals. However, neither the confidence intervals nor the prediction intervals are valid here.

Caveats:

- Our model does not fit the data well, and so we are also dealing with model bias. The confidence interval calculations assume a good model fit, which is clearly not the case here.
- Prediction intervals are, of course, for new data. Being able to calculate prediction accuracy on new data from training data is a substantial upside of linear models (provided their assumptions are met)

7 Further reading

There is much more to linear regression. Section 3.3 of An Introduction to Statistical Learning is worth a read before you start fitting linear models to your data. That section discusses the following topics:

- Qualitative predictors
- Interaction terms
- Non-linear transformation of the predictors
- Potential problems: non-linearity, collinearity, outliers, heteroskedasticity
- Logistic regression

The R builtin functions for visualization are sometimes not as helpful for quickly looking at data in many different ways. I find the `R` library `ggplot2` very useful in such cases.