MSc - Data Mining

Topic 05 : Regression

Part 01 : Overview

Dr Bernard Butler and Dr Kieran Murphy

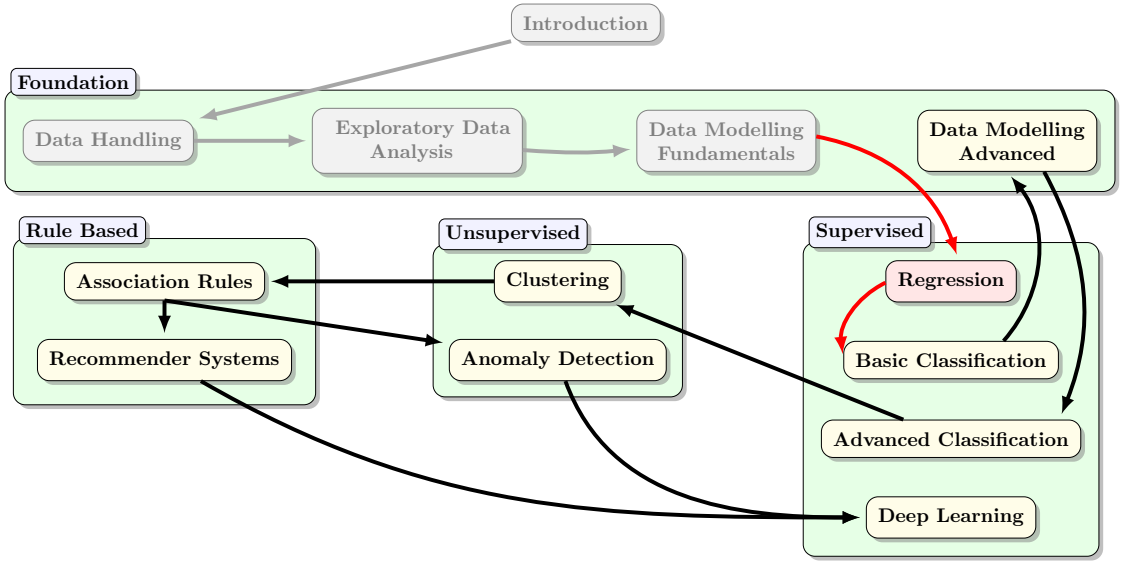
Department of Computing and Mathematics, WIT.
(bbutler@tssg.org; kmurphy@wit.ie)

Spring Semester, 2021

Outline

- Regression as a means of minimising sum of the squared errors
- Regression assumptions - what they mean, how they can be used for validation and model building
- Case studies from Advertising, Diamond sales, Credit Balance prediction

Data Mining (Week 5)



Overview — Summary

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.
- Consider how to check that the regression was successful, and make some improvements if necessary
- To provide context we will use the following datasets:
  - Generated data (various)
  - Diamond dataset: predicting diamond prices given their weights
  - Advertising dataset: predicting widgets sold based on spending in different advertising channels
  - Credit dataset: predicting credit balance using income, status, etc.

Simple Linear Regression: Background

- Linear regression was discovered by Gauss and others around 1800. The "name" came later!
- With small data sets, calculations can be done by hand, but they are tedious and error-prone.
- The goal is simple: Given a (training) set of (x, y) data where y is assumed to have a linear relationship with x
  - Find the line that is the "best fit" to that data
  - Use the specification of that line to *predict* y for the (test) x values
- Note that the "linear relationship" of y upon x is just one of the underlying assumptions
- In practice, the data does not have an exact linear relationship, but it should be "close enough"—but what does that mean?

Review: Linear combinations (scalar product)

Definition 1

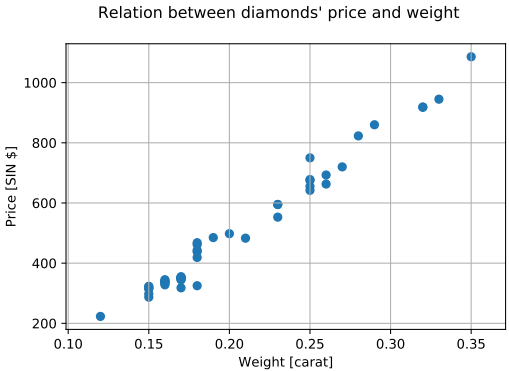
Given two vectors $\mathbf{a}$ and $\mathbf{b}$, each with n elements, the *linear combination* (c) of $\mathbf{a}$ and $\mathbf{b}$ is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

- The linear combination of 2 vectors is a scalar, which can be seen as "mixing" two vectors.
- Matrix-vector multiplication $A\mathbf{x}$ can be seen as the linear combination of each row in the matrix A with the (column) vector $\mathbf{x}$.
- Matrix-matrix multiplication AB can be seen as the linear combination of each row in the matrix A with each column in the matrix B.
- Two nonzero vectors $\mathbf{a}$ and $\mathbf{b}$ can have a scalar product that is zero if $\cos(\mathbf{a}, \mathbf{b}) = 0$, i.e., the $\mathbf{a}$ and $\mathbf{b}$ vectors are perpendicular to each other.
- Linear combinations are used for prediction from linear (regression) models.

Motivating example: Diamond data



Diamond Prices by Weight

- Given the data on the left, can we use it to predict the price of a diamond that weighs 0.22 carat?
- NB - we have not seen a diamond with that weight before in the data
- Can you think of at least 4 other factors that might affect the price?

Simple Linear Regression: Formulation

Definition 2 (Matrix formulation)

- Given data $\{x_i, y_i\}$ where $i = 2, 3, \ldots, n$ and β_0, β_1 as the (unknown, but to be determined) *intercept* and *slope* of the regression line for this data.
- For $n = 2$ points with $x_2 \neq x_1$, this can be solved uniquely for β_0, β_1, using techniques you learnt for your Junior/Inter Cert.
- For $n > 2$ collinear points, just pick any two points and solve as before.
- General equation is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \hat{y}_i + \epsilon_i$ (data = model + error), where $\hat{y}$ is the predicted y for these values of β_0, β_1.
- Matrix form is $\mathbf{y} = X\beta$. Remember matrix-vector multiplication: inner product of i^{th} row of X times the vector β = $1 \times \beta_0 + x_i \times \beta_1 = \hat{y}_i$.
- However, we don't know β yet, nor do we know $\hat{y}_i$, so we use y_i as an estimate of $\hat{y}_i$ and solve for all data in the training set.
- So: our task is to solve the *overdetermined* (number of rows exceeds the number of columns) system of equations $\mathbf{y} = X\beta$ for β

Simple Linear Regression: Normal Equations

$$
\begin{aligned}
\mathbf{y} &\approx X\beta \\
\mathbf{y} &= X\beta + \epsilon \\
X^\mathsf{T}\mathbf{y} &= X^\mathsf{T}X\beta + X^\mathsf{T}\epsilon \\
X^\mathsf{T}\mathbf{y} &= X^\mathsf{T}X\beta
\end{aligned}
$$

because of how ϵ is defined*. Swapping sides, we have

$$
\begin{aligned}
(X^\mathsf{T}X)\beta &= X^\mathsf{T}\mathbf{y} \\
(X^\mathsf{T}X)^{-1}(X^\mathsf{T}X)\beta &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}\mathbf{y}
\end{aligned}
$$

which is equivalent to the *Normal equations*

$$\beta = (X^\mathsf{T}X)^{-1}X^\mathsf{T}\mathbf{y} \tag{1}$$

Note that everything on the right is a set of operations on the data.

Simple Linear Regression: Implementation

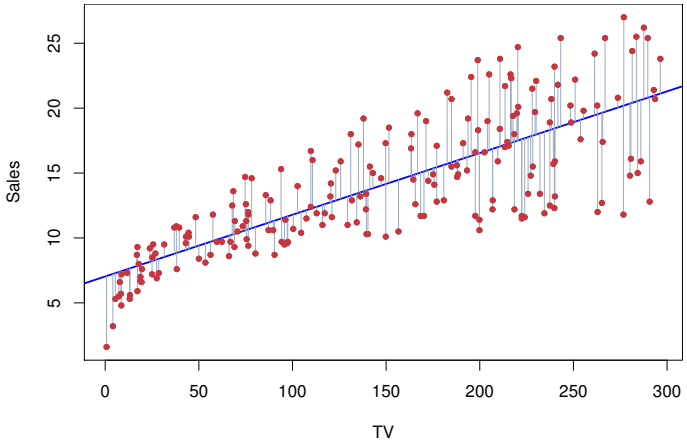
When implemented in software, the Normal equations are not used directly: better[†] algorithms are used instead, but the results are equivalent in exact arithmetic (remember: digital computers perform finite-precision arithmetic and so cannot be exact).

One option is to use statsmodels: consistent with R (separate model specification), excellent diagnostics as standard

Another option is to use sklearn: consistent with other sklearn algorithms, more controls

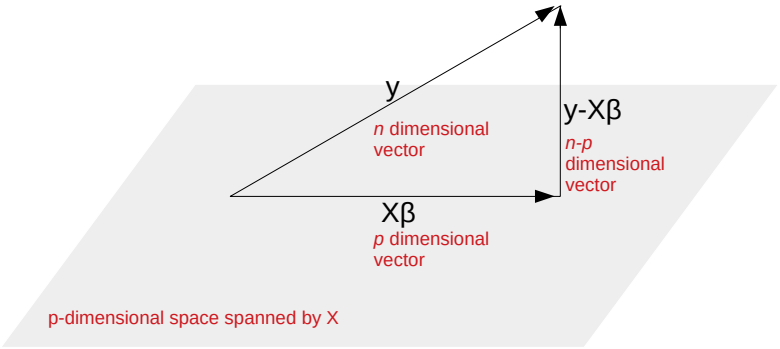
Remember: after *learning* the β parameters, it is then possible to predict $\hat{y}$ for "new" x values, using separate *prediction* function calls.

SLR: Residual Plot for the model



Source: ISLR, Fig 3.1: Advertising data with the model "Sales ∼ TV".
Note the vertical distance between the red dots (data points) **y** and the corresponding $\hat{\mathbf{y}}$ on the regression line, which is termed the *error* ϵ.

Geometrical interpretation of regression



Note that the X matrix spans the $p \times p$ space represented by the grey plane, but y has $n > p$ dimensions and so is represented by a point that lies outside the plane. When y is projected onto the X space, the projected point is $\hat{\mathbf{y}}$ and the residuals are represented by the vector $\mathbf{y} - X\beta \equiv \mathbf{y} - \hat{\mathbf{y}}$.
This decomposition of n data dimensions (observations) into p model parameters and n residuals with rank $n - p$ is helpful when interpreting regression diagnostics.

OLS and Linear Regression

Definition 3 (BLUE)

According to the Gauss-Markov theorem, *Ordinary Least Squares* (OLS), which uses the Normal equations to minimise the sum of the squares of the errors ($\|\epsilon\|_2 \equiv \sqrt{e_1^2 + e_2^2 + e_3^2 + \ldots + e_n^2}$), is the *Best, Linear, Unbiased, Estimator* of that model that can be derived from the training data, provided some reasonably loose assumptions hold.

When we discuss Bias, Variance and Irreducible Error, it is clear that low bias is not enough. OLS might be BLUE but that does not guarantee low variance, because overfitting can still be a problem.
In practice, the assumptions required for OLS to be appropriate can be stated in terms of properties of the residual vector ϵ.
In the rest of this lecture, we will generalise from Simple to Multiple Linear Regression, where $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$ and $2 \leq p \leq n$, so instead of fitting lines, we fit (hyper)planes to data.

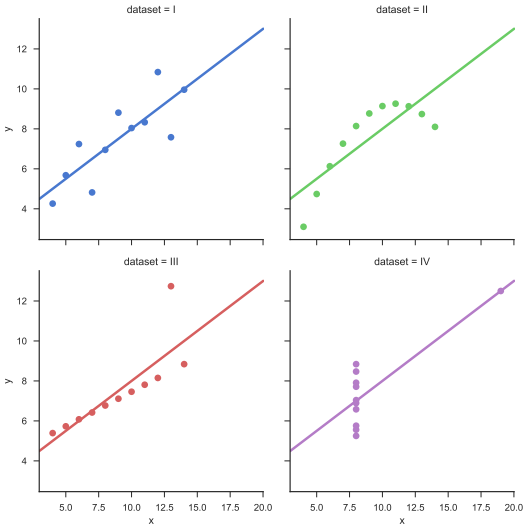
Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.
3. The predictors are not pairwise collinear, i.e., each pair of predictors β_{j_1} and β_{j_2} (associated with columns $X(:, j_1)$ and $X(:, j_2)$) have low correlation (equivalently, the inner product of $X(:, j_1)$ and $X(:, j_2)$ is far from zero).
4. There is no auto-correlation in $\mathbf{y}$.
5. The errors are *homoscedastic* (i.e., Var(ϵ) is constant over the range of $\mathbf{x}$ or $\mathbf{y}$).

Because these assumptions depend both on the data and on the model fitted to that data, it is meaningless to say that "Data set A does not satisfy the linear regression assumptions", because this observation might not apply to all formulations of all models applied to that data.
Consequently, these assumptions can be used constructively, when model building, or as checks, when validating models.

Anscombe's quartet (1973)



Francis Anscombe devised 4 data sets to show different forms of misalignment between data and models. Sets I,II,III share the same x values. All 4 sets share approximately the same descriptive statistics (mean and variance), but little else is common to all 4!

Only I appears suited as it stands. The other data sets require some work, particularly IV.

What do you think needs to be done for each data set?

Common Cost Functions in Regression Models

Remember: we are trying to minimise a loss function based on the error.

Measure	Definition	Purpose
Mean square error (MSE)	$\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{m}$	Mathematically tractable but places greater emphasise on observations with large error
Root mean square error (RMSE)	$\sqrt{\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{m}}$	Has same units as data
Mean absolute error (RMAE)	$\frac{\lvert p_1-a_1\rvert+\cdots+\lvert p_m-a_m\rvert}{m}$	Does not overemphasise observations with large error (like MSE)
Relative square error (RSE)	$\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{(p_1-\bar{a})^2+\cdots+(p_m-\bar{a})^2}$	Relative metric compares the error in the predictions with errors in the simplest model possible (a model just always predicting the average value of y)
Root Relative square error (RRSE)	$\sqrt{\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{(p_1-\bar{a})^2+\cdots+(p_m-\bar{a})^2}}$	
Relative absolute error (RAE)	$\frac{\lvert p_1-a_1\rvert+\cdots+\lvert p_m-a_m\rvert}{\lvert p_1-\bar{a}\rvert+\cdots+\lvert p_m-\bar{a}\rvert}$	

where a_j is the actual value, p_j is the predicted value, m is the number of observations, and $\bar{a}$ represents the mean of the a_j.

Choices of Vector norms

Definition 5 (Manhattan norm)

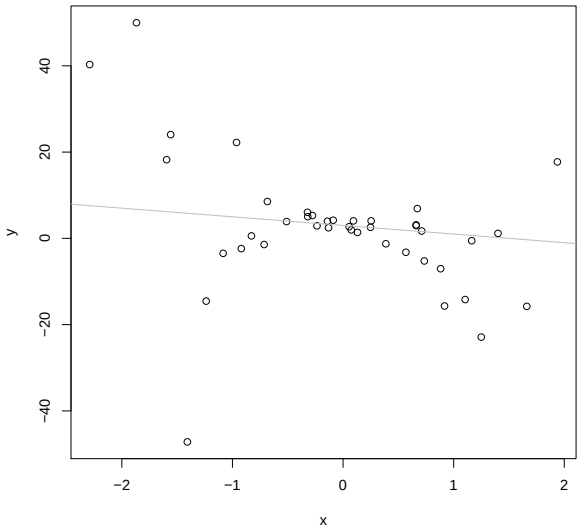
$\ell_1(\ldots) = \|\ldots\|_1$ is the *Manhattan* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Then $\ell_1(\ldots) = \|\ldots\|_1 = |x_1| + |x_2| + \ldots + |x_m|$ is the *Manhattan* distance of $\mathbf{x}$ from the origin. Think of having to *walk* from one junction in Manhattan to another, the distance is the difference in the Street numbers plus the difference in the Avenue numbers.

Definition 6 (Euclidean norm)

$\ell_2(\ldots) = \|\ldots\|_2$ is the *Euclidean* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Then $\ell_2(\ldots) = \|\ldots\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_m^2}$ is the *Euclidean* distance of $\mathbf{x}$ from the origin. Think of being able to *fly* over all the buildings using the shortest route (think: Pythagoras theorem!) from one junction in Manhattan to another.

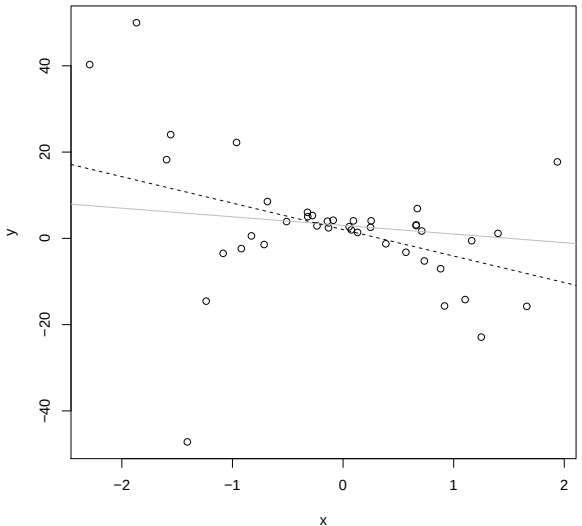
The Euclidean norm is very common, but the Manhattan norm is gaining popularity, because it is robust to outliers and computers are becoming powerful enough. However we generally use Euclidean norm in thisn module.

Case Study 1: Heteroscedasticity - Step 1



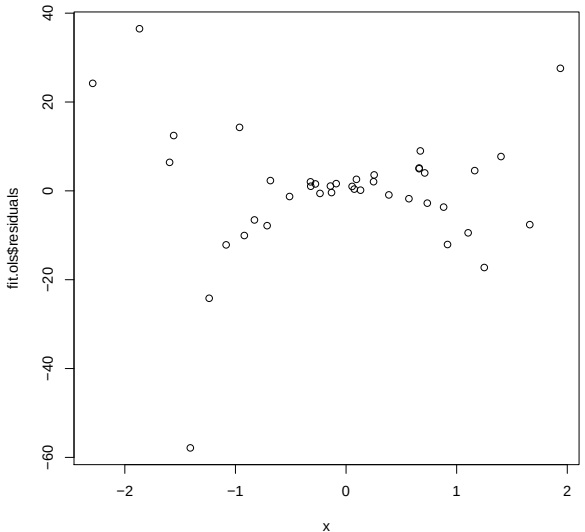
I generated 41 x, y points based on $y = 3 - 2x$, but with added errors that increase away from $x = 0$. The plot shows the line with $\beta = (3, -2)$ in grey.

Case Study 1: Heteroscedasticity - Step 2



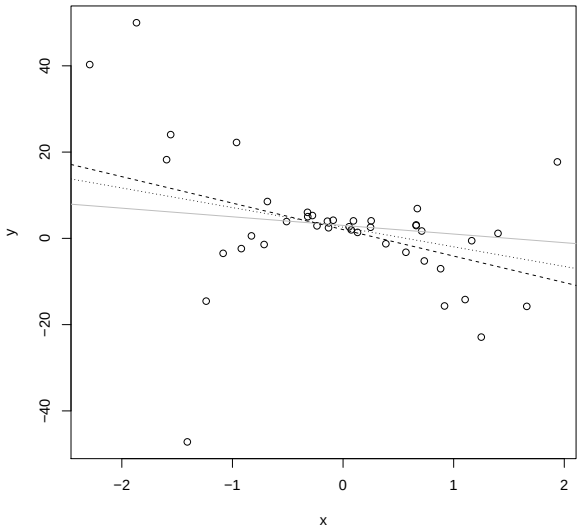
In this plot I added the OLS fit as a dashed line. Note that the parameters of the fit are quite different: $\beta_{OLS} \approx (2, -6)$.

Case Study 1: Heteroscedasticity - Step 3



This plot shows how the OLS residuals ϵ_{OLS} increase rapidly away from 0, as expected (since this was how the data was generated).

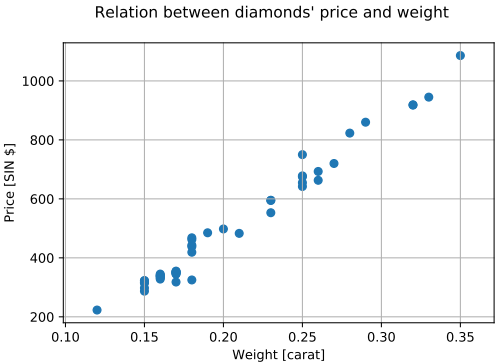
Case Study 1: Heteroscedasticity - Step 4

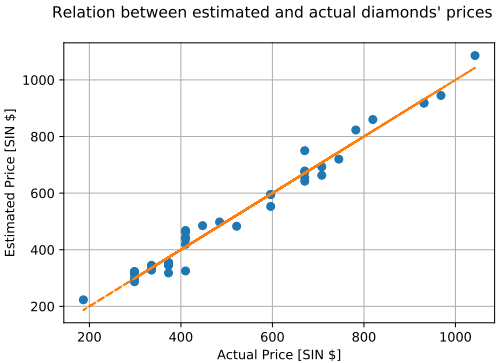


By inspecting the previous residual plot I estimated a weighting function so that the residuals would be "more constant". When this was used to scale the residuals, the resulting Weighted Least Squares estimates were $\beta \approx (2.6, -4.5)$ (shown as a dotted line) and hence closer to the "true" β.

Can you see a problem with finding the weights?

Case Study 2: Diamonds - Check relationship





Clearly there is a linear relationship between a diamond's weight (in carats) and its price (in Singapore dollars, as here). So that is one assumption satisfied!

Sometimes the dependent variable has a linear dependence on a function of an attribute. Example functions include log, exp, sqrt, polynomial, etc. Even if the function is nonlinear in the attribute, that does not matter, as long as the model is linear in the regression parameters β.

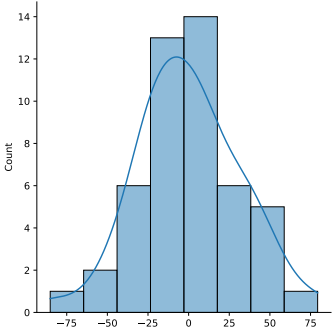
Case Study 2: Diamonds - Check residual distribution

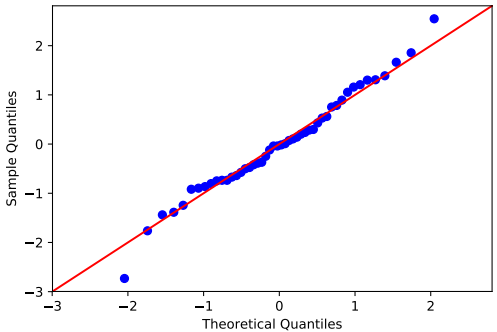
residHist

```
import seaborn as sns
resFig = "res/residHist.pdf"
sns_plot = sns.displot(x = residuals, kde=True)
sns_plot.savefig(resFig)
```

residualsQQ

```
resFig = "residualsqq.pdf"
fig = sm.qqplot(residuals, fit=True, line = '45')
fig.savefig(resFig)
```





Both diagnostic plots indicate the residuals are reasonably close to Normal distribution centred on 0. The qqplot is perhaps more informative. Looking good so far!

Is the standardised residual distribution heavy-tailed or light-tailed relative to the Normal distribution? Any other features?

Case Study 2: Diamonds - model summary

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Sun, 11 Feb 2018	**Prob (F-statistic):**	6.75e-40
Time:	16:22:40	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

smSumm

```
1 simpleModel.summary()
```

The output from Python's statsmodels.summary() call has lots of information!

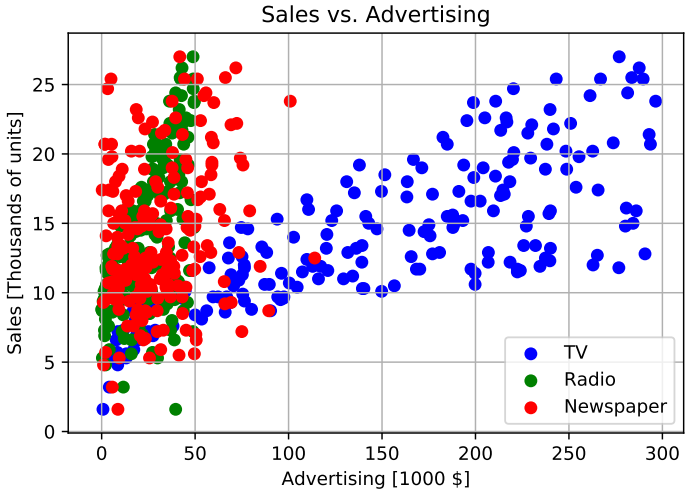
- **How much of the variability of the data is explained by the model?**
- **What is the probability that such data arose if price does not increase with weight?**
- **Explain the degrees of freedom in the table**
- **What scores indicate that the distribution of the residuals is Normal?**

Case Study 3: Advertising: Data and Hypotheses

	TV	Radio	Newspaper	Sales
1	230.1	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	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

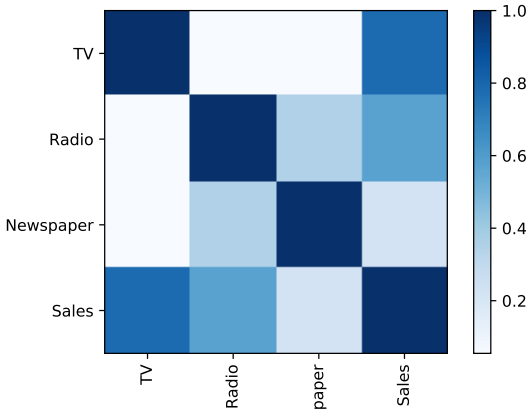
In this data set, the sales figure captures thousands of widgets of a particular type wss sold in a year. Newspaper, Radio and TV represent the annual spend per widget type on the associated advertising channel. The hypothesis is that spend on advertising is a good predictor of sales performance. Since marketing budgets are limited, where should the adverts be placed for maximum sales?

Case Study 3: Advertising: Looking at the data



Which of the advertising channels appear to have a linear relationship with Sales?

Case Study 3: Advertising: Collinearity?



Correlation matrix can indicate which attributes should participate in the model as predictors.

A good predictor should have a high correlation with the dependent variable (Sales in this case) and should have low correlation with other candidate predictors.

What are expected to be good predictors for this data?

Sidebar: specifying models

The statsmodels way

- The dataframe contains the observed variables
- The model is specified separately
- Easier to change the model when experimenting

The sklearn way

- The dataframe contains the (computed) features
- The model is defined implicitly
- Standard interface across all sklearn

statsmodels models look like "Sales $\sim$ TV * Radio + poly(Newspaper,2)"; notation came from applied statistics community.
statsmodels offers its own plotting (like seaborn but not as good). Model summary is very convenient.
sklearn exposes more of the details (e.g., choice of algorithm and configuration parameters).
Both statsmodels and sklearn use the same libraries (scipy, numpy, etc.) underneath.

Case Study 3: Advertising: Model Building

- Start from a "full model" and prune, versus from an "empty model" and add
- We choose the latter, as it is often easier to avoid overfitting

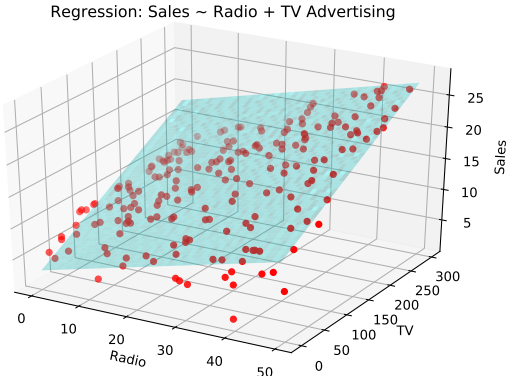
Example 7 (Forward Selection for Advertising Data)

Define: regression model score: (adjusted) R^2 for a given model.

1. Fit "Sales $\sim$ Newspaper", "Sales $\sim$ Radio", "Sales $\sim$ TV" and calculate their R^2 values.
2. Choose the best (largest R^2) single-term model ("Sales $\sim$ TV" in this case) with $R^2 = 0.61$.
3. Fit "Sales $\sim$ TV + Newspaper" and "Sales $\sim$ TV + Radio" and choose the best by R^2 score, which is "Sales $\sim$ TV + Radio" with $R^2 = 0.9$, which is significantly better.
4. Fit "Sales $\sim$ TV + Radio + Newspaper". Its R^2 score is also 0.9, so we favour the existing simpler two-term model (Occam's Razor: other things being equal, choose the simplest model.).

So our preferred model is "Sales $\sim$ TV + Radio" with adjusted $R^2 = 0.9$.

Case Study 3: Advertising: Viewing the Model



Since this two-term model ignores the contribution of the newspaper channel, the Newspaper spend as a contribution to Sales is just another component of the unmodelled (and apparently random) contribution to Sales.

However, the result is a model where every term is highly significant and the model "explains" 90% of the variance of the data, which is high for an observational study. **Why? Can we do better?**

Case Study 3: Advertising: Interactions; Interpretation

- Trying powers of the Radio and TV greater than 1 did not offer much more.
- However, by adding the TV,Radio interaction so that the model became "Sales $\sim$ TV + Radio + TV:Radio" or equivalently "Sales $\sim$ TV * Radio", R^2 increased to 0.97 from 0.9, which is a significant improvement.
- All β terms have $t-$statistic significance of approximately 0.001 which is extremely significant.
- $\beta_0 = 6.75$, $\beta_{\text{TV}} = 0.019$, $\beta_{\text{Radio}} = 0.029$ and $\beta_{\text{TV:Radio}} = 0.001$, indicating that there is a favourable relationship between TV and Radio advertising ($\beta_{\text{TV:Radio}} > 0$), and that additional spending on Radio results in more Sales than the same spending on TV ($\beta_{\text{Radio}} > \beta_{\text{TV}}$).
- Spending on Newspaper advertising should be discontinued as its contribution to Sales is insignificant (indistinguishable from random noise).

Case Study 4: Credit balances - overview

Introducing

- the `sklearn` approach to regression (we used `statsmodels` with the Diamonds and Advertising data)
- non-numeric explanatory variables like gender and ethnicity
- more advanced regression modelling, e.g., handling correlated variables

Case Study 4: Credit balances - introduction

	Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
1	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333
2	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903
3	104.593	7075	514	4	71	11	Male	No	No	Asian	580
4	148.924	9504	681	3	36	11	Female	No	No	Asian	964
5	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

Note the presence of some categorical attributes (Gender, Student, Married, Ethnicity). These can participate in linear regression models to predict a numeric response, but must be coded first.
For example, Gender can become an indicator (0,1)-valued variable of the form IsFemale.
Ethnicity has 3 levels and is replaced by 3-1=2 indicator variables.

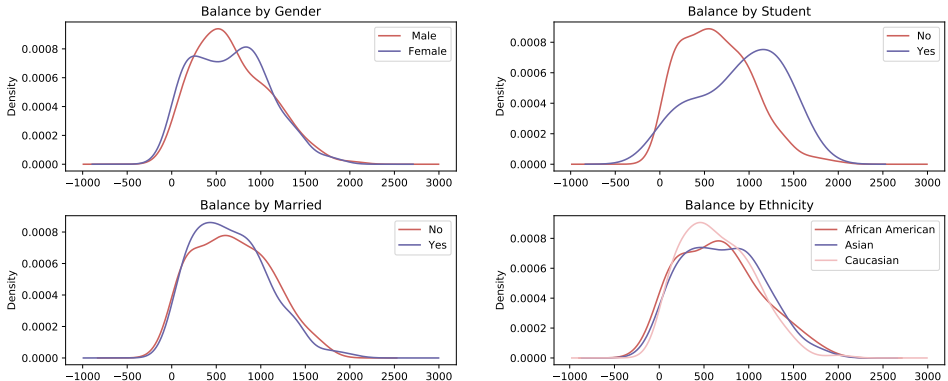
Case Study 4: Credit balances - Removing Data

- the purpose of the analysis is to predict credit balances.
- Basic exploratory data techniques (histograms) soon indicated that there were 2 cohorts
  1. those who do not use their cards and/or clear their balance each month
  2. those who use their cards and have nonzero balances
- Removing data relating to the first cohort meant that the remaining data looked more cohesive and also made linear regression easier
- Take-away: look for inconsistent subsets in the data, remove them if possible

Case Study 4: Credit balances - Removing correlated attribute

- Correlations between predictors are relatively high, but that between “Limit” and “Rating” is 1
- Generally, customers with a high rating are allowed to have high credit limits
- Conversely, customers will not be allowed high credit limits unless they have a high credit rating
- “Limit” was removed from the data used for analysis
- Take-away: remove correlated attributes, because they increase the standard error (hence variance) and make the solver’s job much more difficult

Case Study 4: Credit balances - Contribution of Categorical Variables



Which of these categorical attributes has a significant effect on Balance?

Case Study 4: Credit balances - Model building

- Using forward selection as before, the best model was found to be "Balance $\sim$ poly(Income,2) + Rating + Age + Student + Income:Rating"
- Could also use Backward Elimination to prune from a complex model
- For this data, high correlations between features can cause difficulties - we need techniques to handle this

Difficulties caused by correlated features

The Problem : Several features are highly correlated, so the solver has difficulty assigning an importance independently to each.

How it shows up : The condition score is large and several model coefficients take large values with opposite signs. Sometimes the solver gives up.

Solution options :

1. Remove selected features from the model (simple, does not always work and requires care)
2. Use *regularisation*, to "penalise" large model coefficients (solve a related problem with a different loss function)
3. Use *dimensionality reduction* (linear PCA) to derive an uncorrelated subset of the features with least loss in explanatory power (principal components can be opaque)

Regularisation introduction

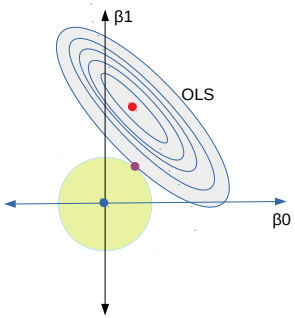
Add *regularisation* constraints to make the model work: $\min_\beta \|\epsilon\|_2^2 + \lambda p(\beta)$

- Options are
  1. *Ridge Regression* where the penalty term takes the form $p(\beta) = \|\beta\|_2$
  2. *Lasso* where the penalty term takes the form $p(\beta) = \|\beta\|_1$
- Regularisation has a metaparameter λ - the challenge is to choose a suitable value
  - if too large: tries less to match the data, increases the bias
  - if too small: tries too hard to match the data so $\beta \to \infty$ and increases the variance

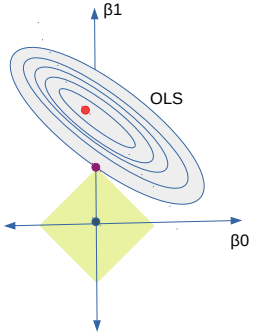
Ridge vs Lasso Regression

Because lasso regression favours the “corners” in parameter space, it tends to set some parameter values to 0 (essentially dropping the associated features). This has the added benefit of making the model smaller and easier to interpret.

Ridge Regression



Lasso Regression



Case Study 4: Credit balances - Regularisation - Searching for λ

1. Choose a set of candidate λ values
2. For each candidate λ, use K-fold cross-validation on data subsets to estimate the prediction error for the regularised fit with that λ
3. Choose the λ for which the expected error is least
4. Now fit all the training data again with this choice of λ

Note that lasso (but not ridge regression) can set particular β_j to 0 (effectively removing them from the model), so it operates more like backwards elimination in terms of creating a more frugal model having fewer terms.
Ridge regression downweights certain terms but does not set them to zero. However, it can be more performant.

Attribute independence in Multivariate Data

Definition 8 (Covariance)

$\sigma_{12} = E[(X_1 - \mu_1)(X_2 - \mu_2)]$. In words, for two attributes X_1 and X_2, with means μ_1 and μ_2, respectively, σ_{12} is a measure of the the linear dependence between them. If they are independent, we can show that $\sigma_{12} = 0$.

Definition 9 ((Variance-)Covariance Matrix)

When there are n numeric attributes, there are $n \times n$ pairs of covariances $\sigma_{ij}, i = 1, \ldots, n; j = 1, \ldots, n$. The resulting covariance matrix is symmetric and diagonally dominant. This matrix captures the covariance structure of the set of n attributes $\{X_i\}$.

Sometimes it is convenient to work with the correlation matrix, which is a scaled version pf the covariance matrix, with elements $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$, which is scaled so that all the diagonal elements are 1 and the off diagonal elements satisfy $-1 < \rho_{ij} < 1$. If two attributes are highly correlated, adding the second into the model does not increase the explanatory power of the model. Therefore, it pays to determine the covariance matrix from the data before building any models.

Multivariate data with correlated measurements

Example 10 (Measles cases, by city, per week from 1948–1985)

This data spans the period before and after the introduction of vaccination for measles (during the mid 1960s). Measles cases are recorded per week in 7 English cities. Although the cities are not adjacent, it is likely that there will be some spatial autocorrelation. Also, by the nature of disease outbreaks, there will also be some temporal autocorrelation per city.

Row ID	Date	London	Bristol	Liverp...	Manch...	Newc...	Birmin...	Sheffield
Row0	1948-01-10	?	3	40	22	58	78	9
Row1	1948-01-17	240	4	51	19	52	84	11
Row2	1948-01-24	284	3	54	23	34	65	11
Row3	1948-01-31	340	5	54	31	25	106	4
Row4	1948-02-07	511	1	89	66	27	142	7
Row5	1948-02-14	649	3	73	60	47	143	3
Row6	1948-02-21	766	13	169	87	46	191	6
Row7	1948-02-28	932	5	212	61	66	208	9
Row8	1948-03-06	1303	4	283	79	57	290	7
Row9	1948-03-13	1257	15	285	56	82	310	10
Row10	1948-03-20	1716	9	279	85	92	425	5

Removing redundant attribues, based on correlation filters

Row ID	London	Bristol	Liverp...	Manch...	Newc...	Sheffield
London	1	0.474	0.295	0.52	0.52	0.539
Bristol	0.474	1	0.228	0.438	0.374	0.68
Liverpool	0.295	0.228	1	0.432	0.482	0.33
Manchester	0.52	0.438	0.432	1	0.554	0.523
Newcastle	0.52	0.374	0.482	0.554	1	0.536
Birmingham	0.707	0.547	0.365	0.472	0.646	0.691
Sheffield	0.539	0.68	0.33	0.523	0.536	1

The Pearson product-moment correlation matrix was computed and filtered on the critical value $\rho^{(\text{crit})} = 0.7$. Because the correlation between London and Birmingham exceeded 0.7, one of them was removed (Birmingham) so its column was omitted from the filtered correlation matrix above.

Note that the correlation matrix based on rank (e.g., Kendall's τ) is different and the linear dependence between London and Birmingham data was much weaker in that case.

Working with high-dimensional data

Definition 11 (Curse of Dimensionality)

High dimensions do not just require more computing resources and make interpretation more difficult. They also make it more difficult to capture data that samples very high dimensional spaces efficiently. It can be shown that, as the dimension d increases, most of the volume of a hypercube is near the corners, not near the centre, where data might be easiest to collect. This makes estimating parameters much more difficult.

The proof is based on the fact that, as the dimension d tends to infinity, the *ratio* of the volume of the maximum hypersphere inscribed inside the hypercube of the same dimension, tends to 0. Thus there are good reasons to prefer low dimension approximations to high dimensional space.

In 2D: imagine the largest circle fitting inside a square; ratio is $\frac{\pi r^2}{4r^2} = \frac{\pi}{4} \approx 0.79$.

In 3D: imagine the largest sphere fitting inside a cube; ratio is $\frac{(4/3)\pi r^3}{8r^3} = \frac{\pi}{6} \approx 0.52$.

More generally

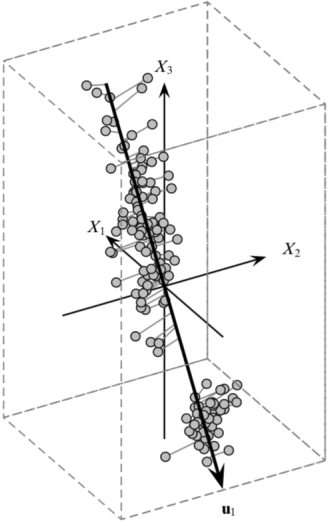
$$\frac{V_{\text{hypersphere}}}{V_{\text{hypercube}}} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \to 0 \text{ when } d \to \infty$$

Feature reduction

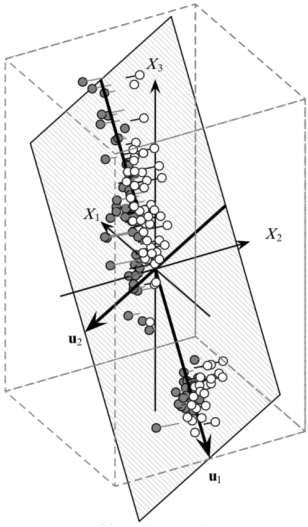
- Sometimes it is possible to use intuition to reduce the dimension, by omitting selected attributes.
- Another possibility is to look for groups of correlated attributes (c.f., *mediation*), such as the London and Birmingham measles cases above, and just choose 1 of these.
- More generally, there are techniques that search for a subspace with specified dimension d' of the attributes that captures most of the variance of the full set of attributes having dimension d, where $d' < d$ (often $d' \ll d$).
- The best known of these techniques is *Principal Components Analysis* (PCA).

PCA visualisation

Mapping to 1 Principal Component



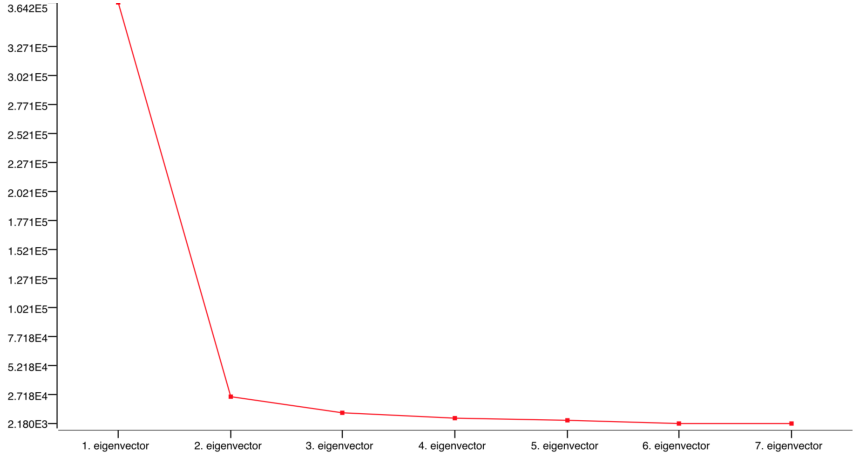
Mapping to 2 Principal Components



PCA interpretation

- Although the data has dimension $d = 3$, it is possible to find the line (on the left; $d = 1$) and plane (on the right, $d = 2$) which retain most of the variance of the data after it has been projected onto this lower dimensional subspace.
- First compute the transformations needed to align the data with the selected subspace.
- Can then project other data, e.g., test data, onto the subspace that was derived with the original, training data.
- Apply the inverse projection to the data, restoring it to its original orientation. However, because of the use of projections, it is not the same as the original data - the round-trip is “lossy”. However, it is helpful to interpret the results in terms of the original attributes.

PCA example



As can be seen from the *scree plot* above, the first eigenvalue captures the bulk of the variance. Arguably, instead of requiring 7 attributes, a single attribute, which is a transformation of the other 7, is sufficient. You could interpret that attribute as representing the measles outbreaks in an archetypal English city...

This youtube video describes PCA concepts well.

Diagnostic plots, statistics and worked examples

See notebooks accompanying this weeks lecture notes.

Review and summary

- Linear regression is one of the foundations of data mining
- It has two phases, of which the first (learning) is generally the most challenging
- It has many variants, so is quite flexible, but flexibility can be abused!
- Careful validation and model building is essential for success - it is an extension of the exploratory work done earlier in the process
- In machine learning, prediction error is the main focus, but you need to be aware of other considerations such as
  1. model parsimony (keep them as small as possible!): faster at both training and evaluation time
  2. the bias-variance dilemma: avoid overfitting and underfitting - remember, your model needs to generalise well from the training to the test set
  3. model interpretability: some models are easier to understand because the terms in the model represent concepts from the domain the data is from

Some Additional Resources

- Book: Introduction to Statistical Learning with R (2013) by James, Gareth and Witten, Daniela and Hastie, Trevor and Tibshirani, Robert.

  I strongly recommend that you read Chapter 3 of the book, as it is very well written and available online for free.

- Kaggle notebooks relating to the datasets addressed this week. There are many, but searching Kaggle should provide nice examples of data mining in action.