MONASH University

MONASH BUSINESS SCHOOL

Statistical Thinking (ETC2420/ETC5242)

Associate Professor Catherine Forbes

Week 9: Regression models

Learning Goals for Week 9

- Synthesise the Bayesian approach
- Compare frequentist and Bayesian inference
- Recognise when transformations may be required
- Review frequentist simple linear regression
- Diagnose problems with a regression model

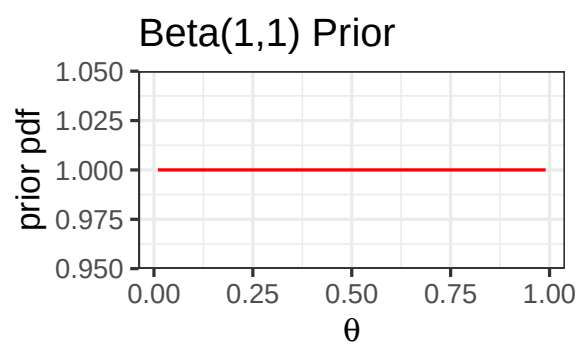
Assigned reading for Week 9:

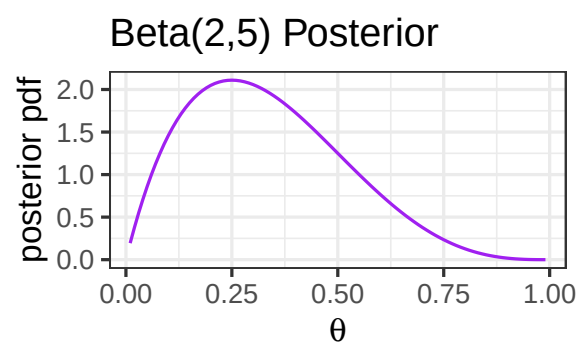
- Chapter 5 in ISRS

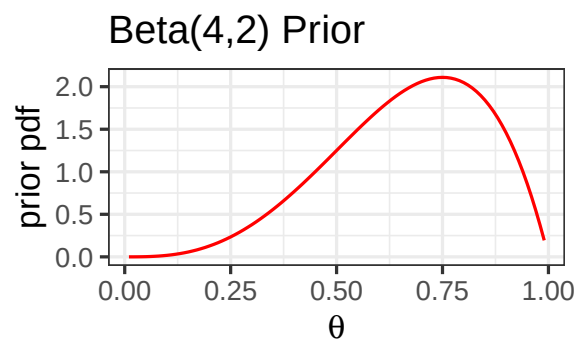
Synthesise the Bayesian approach

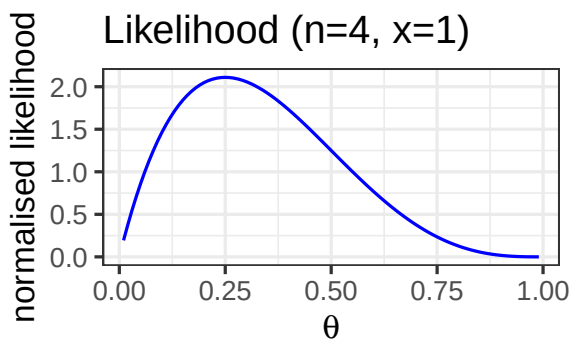
- We have discussed the idea of using subjective prior information $p(\theta)$, for $\theta \in \Theta$
- And combining this with data $X_1, X_2, \ldots, X_n \overset{i.i.d.}{\sim} F_X(x \mid \theta)$
- Specifically using conjugate prior-likelihood pairs
  - Beta-Binomial
  - Gamma-Poisson
  - Gamma-Exponential
  - Normal-Normal

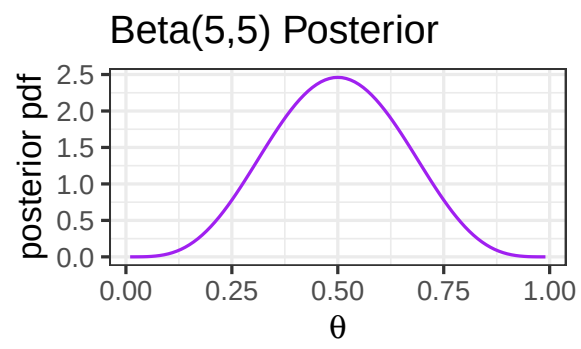
Beta-Binomial $x = 1, n = 4$

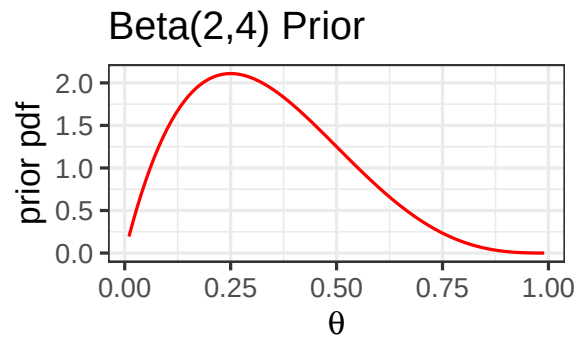


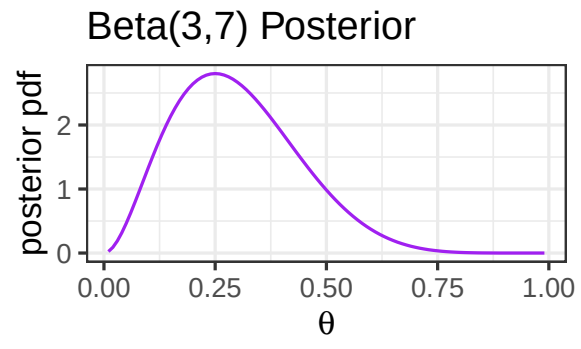




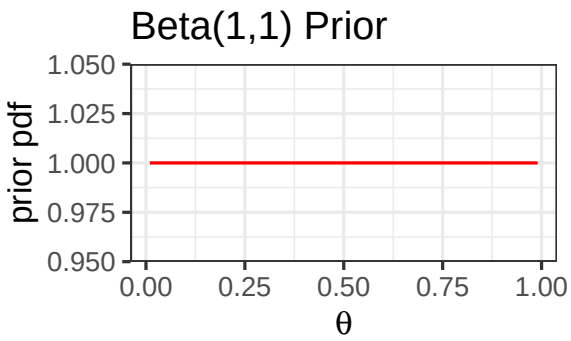


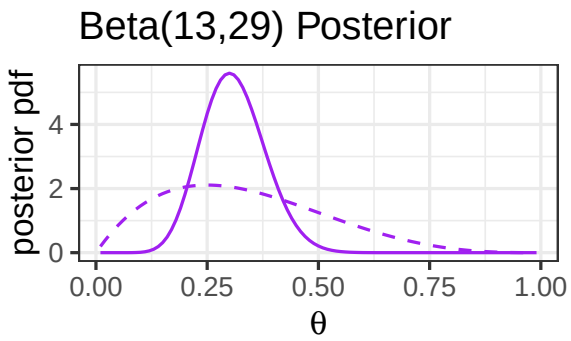


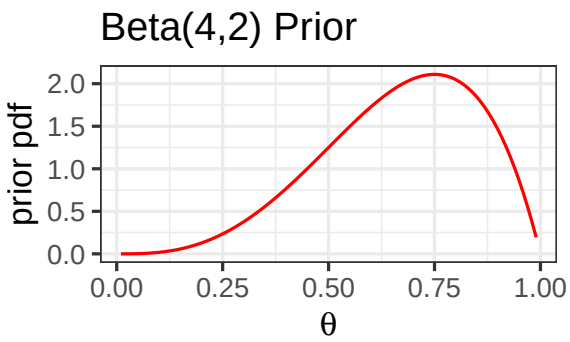


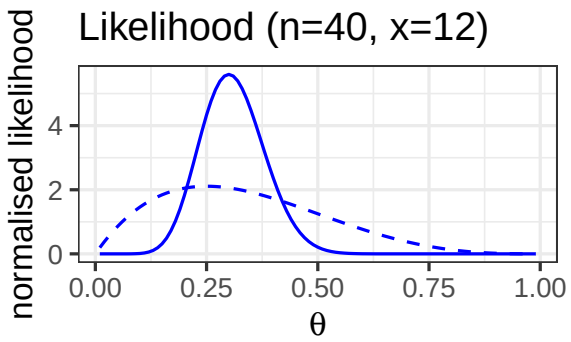


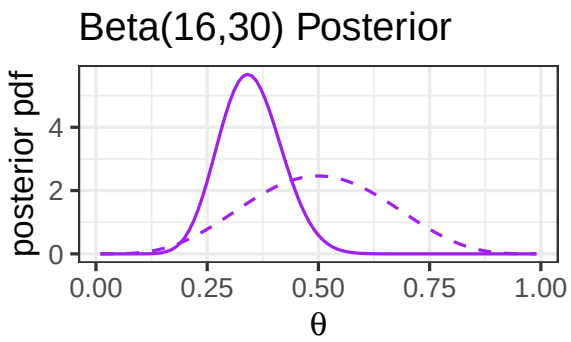
Add Beta-Binomial $x = 12, n = 40$ with same priors

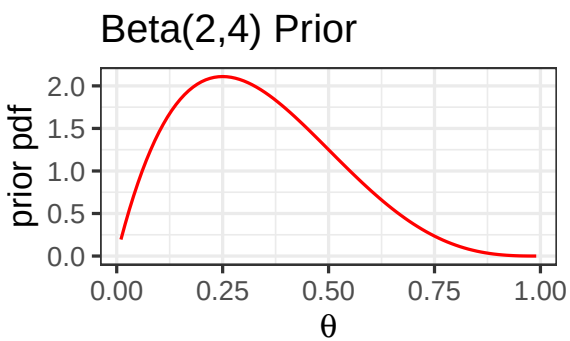


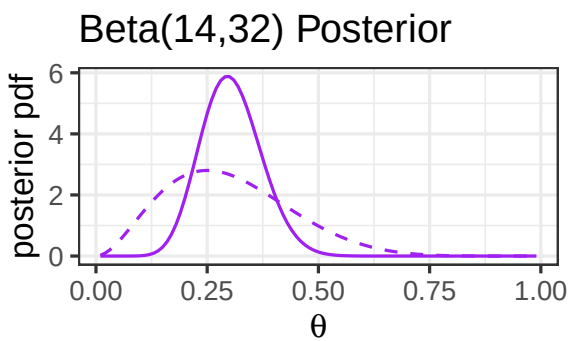




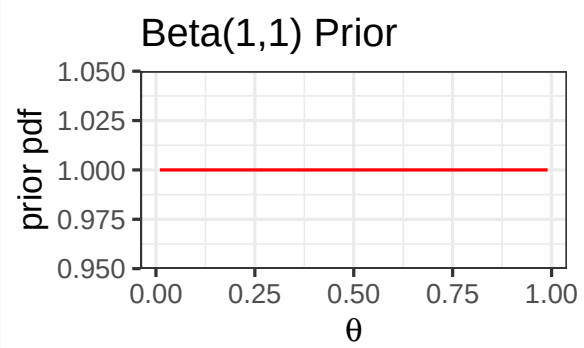


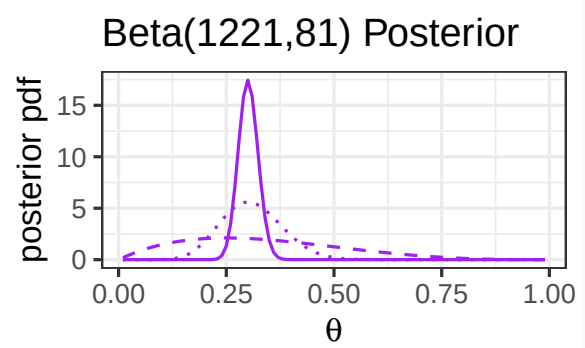


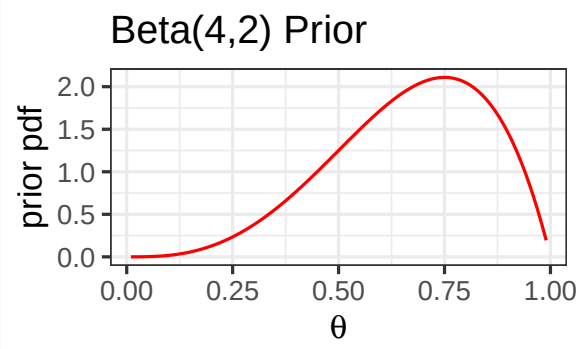


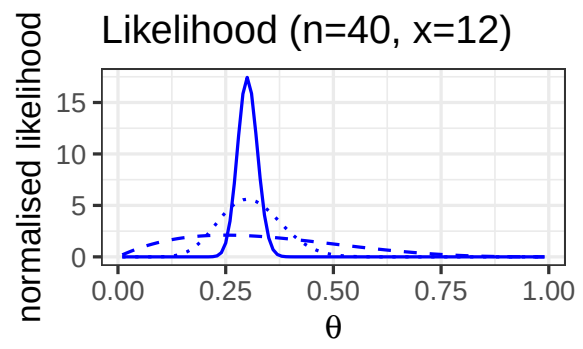


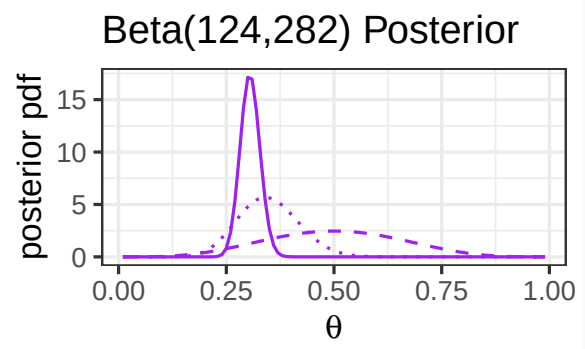
Add Beta-Binomial $x = 120, n = 400$ with same priors

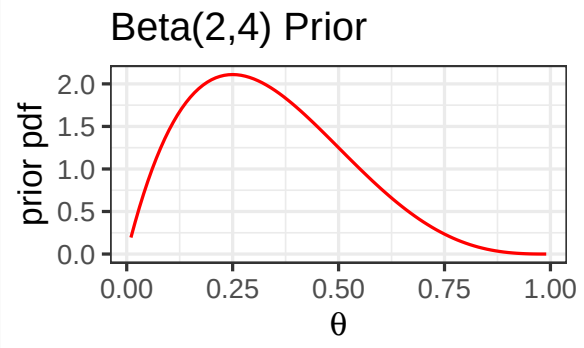


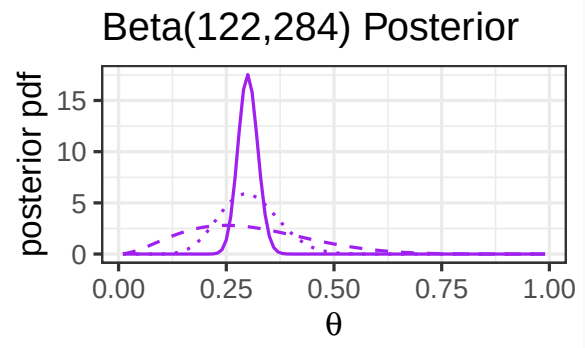












The posterior distribution

- The impact of the prior will diminish as n increases
- **Summarise the posterior distribution using point estimate**
  - Squared error loss: Use the posterior mean
  - Absolute error loss: Use the posterior median
  - Other loss function? Minimise posterior expected loss
- **Summarise the posterior distribution using interval estimate**
  - Any interval with 95% posterior probability: **95% credible interval**
  - **Shortest** interval with 95% posterior probability: **95% highest posterior density (HPD) credible interval**

Setting the prior

- There are many approaches to determining the prior
- For example, one can choose the hyper-parameters of a conjugate through constraints on:
  - the prior mean, prior variance (or other prior moments)
  - the width of a prior 90% interval (or with other probability level)
- And solve (numerically) d nonlinear equations in d unknowns
- e.g. Beta prior: choose prior mean $E[\theta \mid \alpha, \beta] = M$ and prior variance $Var(\theta \mid \alpha, \beta) = V$, then solve for α and β as function of M and V:

$$\frac{\alpha}{\alpha+\beta} = M \quad \text{and} \quad \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} = V$$

- or choose M, L, U and c such that $E[\theta \mid \alpha, \beta] = M$ and

$$\int_L^U \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}\, d\theta = c$$

Bayesian Predictions

- Next observation modelled as $X_{n+1} \sim F_{X|\theta}$, independent of $\{X_1, \ldots, X_n\}$.
- Use posterior to construct **joint** distribution of

$$f(x_{n+1}, \theta \mid x_1, \ldots, x_n) = f(x_{n+1} \mid \theta) \times f(\theta \mid x_1, \ldots, x_n)$$

- Now "marginalise out" θ to get $f(x_{n+1} \mid x_1, \ldots, x_n)$
- Produce
  - point forecasts $E[X_n + 1 \mid x_1, \ldots, x_n]$
  - interval forecasts, etc
- Unlike MLE, don't just "plug-in" $\hat{\theta}$
- Take uncertainty about θ in to account! (Prediction intervals will be wider)

Multi-parameter models

- In all conjugate pair cases so far, the prior involves a **univariate** (hyper-)parameter
- e.g. **Normal-Normal** (**mean only, σ^2 assumed known**)
- $\mu \sim N(\mu_p, \tau^2)$ and $X_1, , X_2, \ldots, X_n \mid \mu \overset{i.i.d.}{\sim} N(\mu, \sigma^2)$
- $\Rightarrow \ \mu \mid \sigma^2, x_1, x_2, \ldots, x_n \sim N(\tilde{\mu}_p, \tilde{\sigma}_p^2)$,
  with $\tilde{\mu}_p = \left(\frac{n\tau^2}{\sigma^2+n\tau^2}\right)\overline{x} + \left(\frac{\sigma^2}{\sigma^2+n\tau^2}\right)\mu_p$ and $\tilde{\tau}^2 = \frac{\tau^2\sigma^2}{\sigma^2+n\tau^2}$
- In some cases, like the above, there is another parameter to include in the prior
- e.g. we want to a prior on $\theta = (\mu, \sigma^2)$

A conjugate prior for the normal likelihood

- There is a conjugate prior for the normal likelihood
- It is called a **Normal-Inverse Gamma** distribution

$$\mu \mid \sigma^2 \sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right)$$

and

$$\frac{1}{\sigma^2} \sim Gamma\left(shape = \frac{\nu_0}{2}, rate = \frac{\nu_0 g_0^2}{2}\right)$$

- We refer to the marginal distribution of σ^2 as “Inverse Gamma”

Normal-Inverse Gamma prior

- To simulate from this prior distribution R times

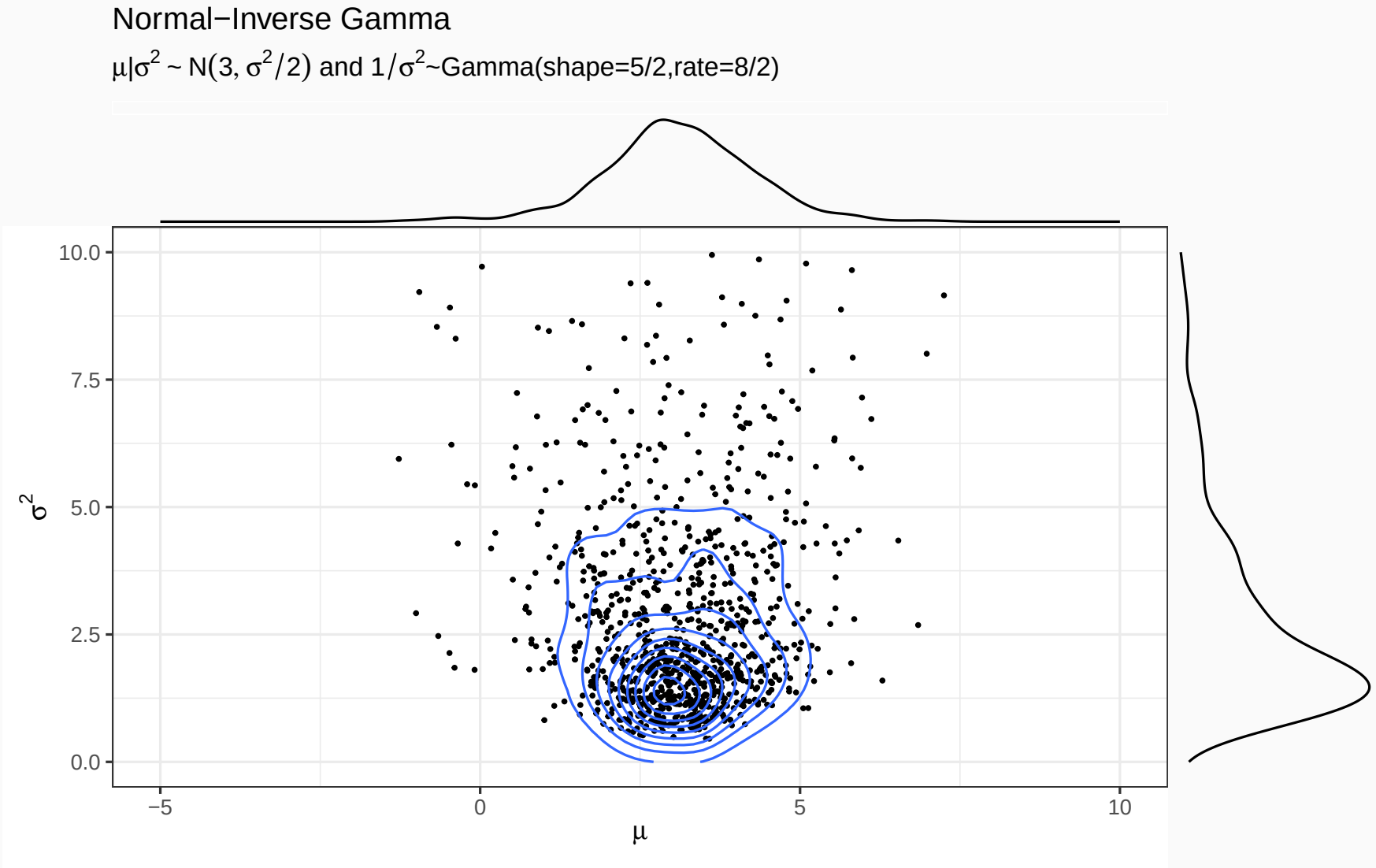
```
R <- 1000
mu_0 <- 3
kappa_0 <- 2
nu_0 <- 4
g2_0 <- 2
rate <- nu_0*g2_0/2

w <- rgamma(R, shape=nu_0/2, rate=rate)
sigsq <- 1/w

mu <- rep(0,R)
for(r in 1:R){
  mu[r] <- rnorm(1,mean=mu_0, sd=sqrt(sigsq[r]/kappa_0))
}
```

- Plot pairs $(\mu^{(r)}, \sigma^{2(r)})$, for $r = 1, 2, \ldots, R$ ("Normal-Inverse Gamma" draws)

Random draws from a Normal-Inverse Gamma prior



Marginal prior for μ is Student-t

- The Normal-Inverse Gamma distribution for $\theta = (\mu, \sigma^2)$
- Can be shown to imply a **marginal** **Student-t** distribution for μ

$$\frac{\mu - \mu_0}{\sqrt{g_0^2/\kappa_0}} \sim t_{\nu_0}$$

- That is, the marginal distribution for μ is Student-t with
  - degrees of freedom ν_0
  - mean $E[\mu] = \mu_0$
  - variance $Var(\mu) = \frac{g_0^2}{\kappa_0} \left(\frac{\nu_0}{\nu_0 - 2} \right)$

Multi-parameter posterior

- Given random sample $x_1, \ldots, x_n$ from $N(\mu, \sigma^2)$
- The posterior distribution is also Normal-Inverse Gamma:

$$\mu \mid \sigma^2, x_1, \ldots, x_n \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right)$$

and

$$\frac{1}{\sigma^2} \mid x_1, \ldots, x_n \sim Gamma\left(shape = \frac{\nu_n}{2}, rate = \frac{\nu_n g_n^2}{2}\right)$$

- where

$$\mu_n = \left(\frac{\kappa_0}{\kappa_n}\right)\mu_0 + \left(\frac{n}{\kappa_n}\right)\bar{x}$$

$$\kappa_n = \kappa_0 + n$$

$$\nu_n = \nu_0 + n$$

$$g_n^2 = g_0^2 + (n-1)s^2 + \frac{n\kappa_0}{\kappa_n}(\bar{x} - \mu_0)^2$$

- Easy to simulate from this NIG prior to (approximate) any posterior quantities of interest, e.g.
  - point and interval predictions

Posterior inference in multi-parameter settings

- In a multi-parameter setting the posterior will be a multi-variate distribution
- $\Rightarrow$ we integrate (sum) to obtain marginal posterior distributions, e.g.
  - $\sigma^2 \mid x_1, \ldots, x_n \sim \textit{Inverse Gamma}\left(\frac{\nu_n}{2}, \frac{\nu_n g_n^2}{2}\right)$
  - $\mu \mid x_1, \ldots, x_n$ will be Student-t with
  - degrees of freedom ν_n
  - mean $E[\mu \mid x_1, \ldots, x_n] = \mu_n$
  - variance $Var(\mu \mid x_1, \ldots, x_n) = \frac{g_n^2}{\kappa_n}\left(\frac{\nu_n}{\nu_n - 2}\right)$
- If you can simulate from the joint posterior, you can approximate the features of the marginal distribution, e.g.
  - marginal means,
  - marginal variances,
  - marginal probabilities,
  - marginal credible intervals

Non-conjugate priors

- In principle we do not want to be restricted to using conjugate priors
- There are many population models where conjugate prior is not available
- $\Rightarrow$ Use advanced simulation techniques to simulate from the posterior distribution of interest
- Breakthrough technique Markov chain Monte Carlo (MCMC)
  - **dependent** posterior draws of multi-variate θ simulated as a Markov chain
  - $\theta^{(r)}$ values drawn using previous draw $\theta^{(r-1)}$
- Basic technique: (Gibbs sampling)

1. Take starting values $\theta^{(0)} = \left(\mu^{(0)}, \sigma^{2(0)}\right)$
2. Then for $r = 1, 2, \ldots, R$, simulate
   i. $\mu^{(r)} \sim \mu \mid \sigma^{2(r-1)}, x_1, \ldots x_n$
   ii. $\sigma^{2(r)} \sim \sigma^2 \mid \mu^{(r)}, x_1, \ldots, x_n$

- Breaking down the simulation into univariate draws makes it easier to sample from the posterior

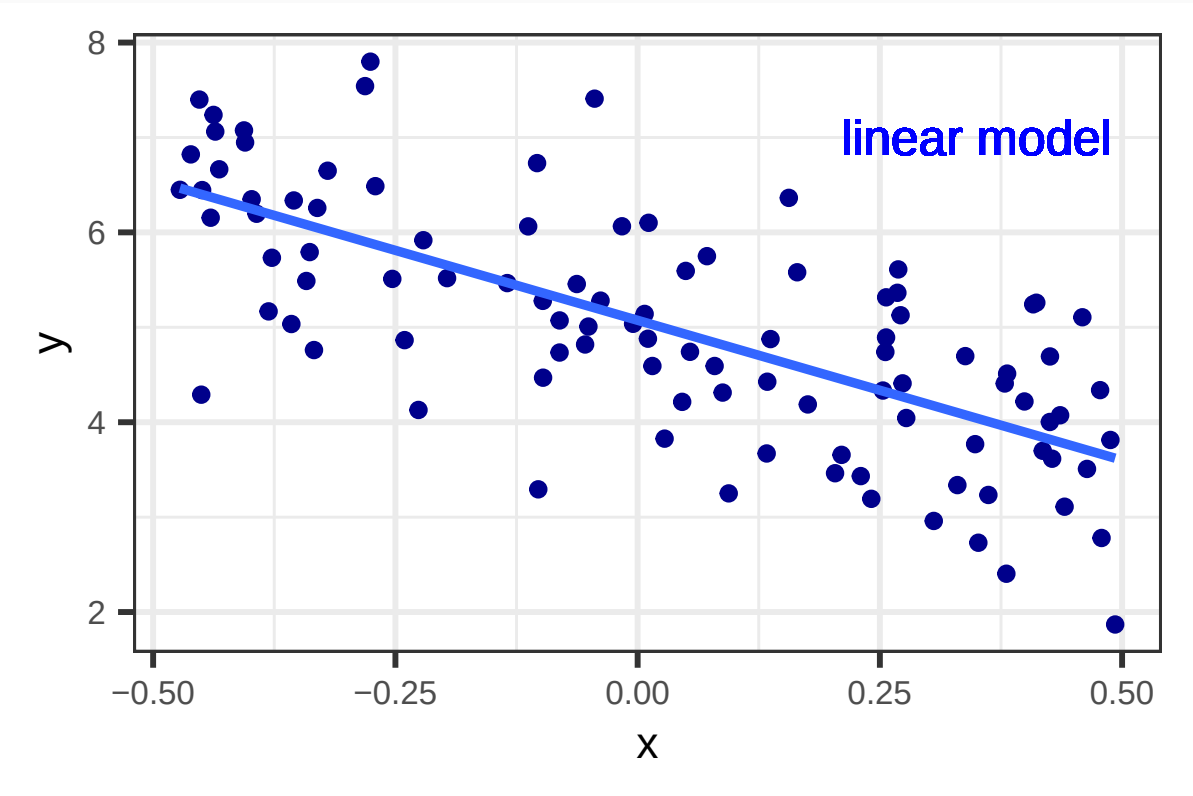
Why study Bayesian methods?

- Modern computation opens up many possibilities not previously available
- Theoretically Bayesian methods will always out-perform frequentist method
- In many practical problems, specific loss function can be constructed and incorporated into the analysis
- For more complex models beyond simple univariate random sample, we have more complex models
  - heterogeneous populations
  - high dimensional data
  - time series data
- Many machine learning methods exploit ideas from Bayesian inference, e.g.
  - add a little bias to reduce MSE
  - average over predictions based on uncertain parameter estimate

Simple linear regression model

- Simple linear regression uses a line to predict value of y_i for a given value of x_i
- Explains how response variable, y, changes (linearly) in relation to explanatory variable, x, on average.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$



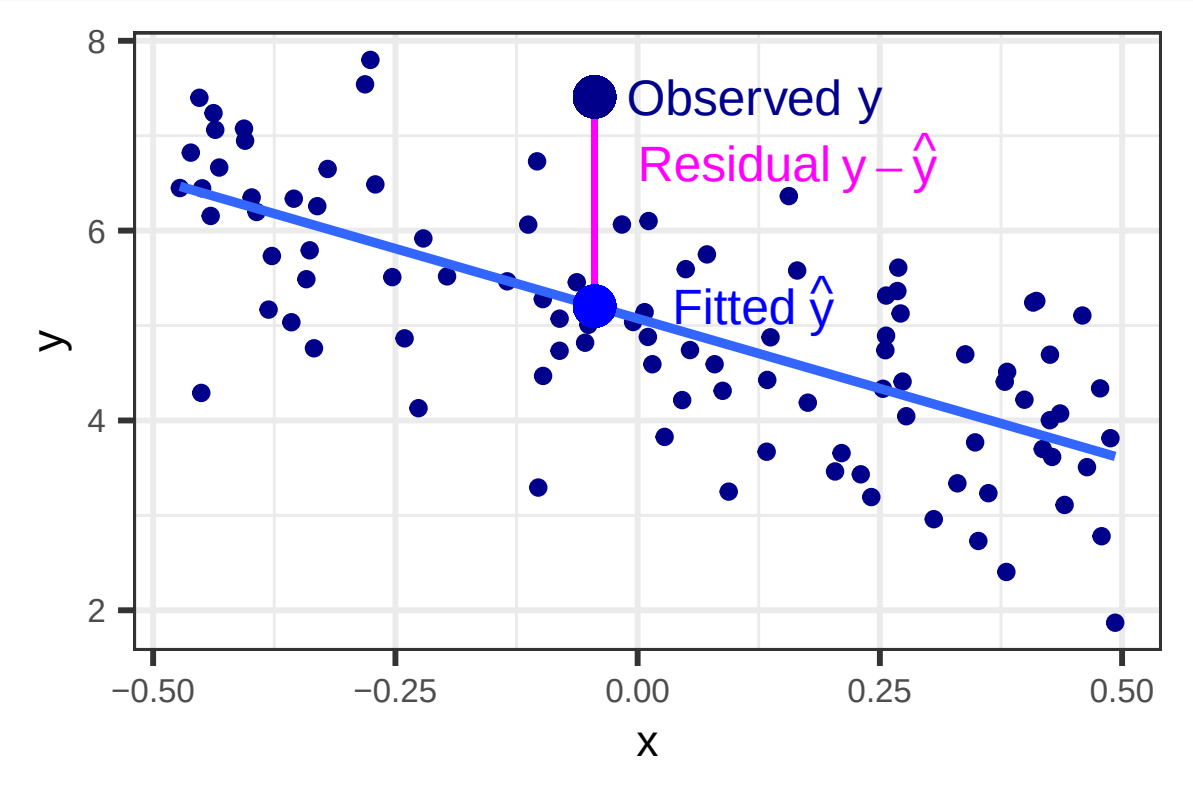
Fitting a regression model using least squares

- Minimise the sum of squared residuals produces the best fitting line
- i.e. Minimise $\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$
- This is **Ordinary least squares (OLS)**

- Fitted line has smallest average **vertical squared distance**, at available observed points

- **Observed** values y_i are points on plot
- **Fitted** (or **Predicted**) values $\hat{y}_i = b_0 + b_1 x_i$ are values that lie on the regression line

Fitting a regression model using least squares



Parameter interpretation

- **Line of best fit**: $\hat{y} = b_0 + b_1 x$, for any value of x
- b_0 is the **y-intercept** of the fitted line with y-axis
- b_1 is the **slope** of the fitted line

Slope coefficient of fitted regression line satisfies

$$b_1 = r\frac{s_y}{s_x}$$

- s_x is sample standard deviation of x_i's
- s_y is sample standard deviation of y_i's
- r is sample correlation, found using x_i and y_i pairs

Given sample means $\bar{x}, \bar{y}$, fitted regression line **y-intercept** coefficient is

$$b_0 = \bar{y} - b_1\bar{x}$$

Does the point $\bar{x}, \bar{y}$ lie on the regression line?

Standard errors

- We have estimated β_0 and β_1 using b_0 and b_1, respectively
- What are the (estimated) **standard errors** for b_0 and b_1 in **hypothetical repeated samples**?

$$SE(b_0) = \sqrt{\frac{MSE \sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

and

$$SE(b_1) = \sqrt{\frac{MSE}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

where

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(n-2)} = \frac{\sum_{i=1}^{n} e_i^2}{(n-2)}$$

Simple linear regression using maximum likelihood estimation

- Simple linear regression (SLR) uses only a single regressor
- The SLR model for observation i is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

- If we assume:
  - $\varepsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$ and x_i's are fixed
- Then, the **likelihood function** is

$$L(\beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2\right\}$$

- And **2 times the log-likelihood** is

$$2l(\beta_0, \beta_1, \sigma^2) = -n\ln(2\pi) - n\log(\sigma^2) - \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

- This is **maximised** at the OLS estimator, with $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n}$
- (We typically use **MSE** based on $(n-2)$ rather than n when estimating σ^2)

Multiple linear regression using maximum likelihood estimation

- **Multiple linear regression** (or just linear regression) uses more than regressor
  - We will assume there are p regressors, including the intercept
- Linear regression model for observation i is

$$y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_{p-1} x_{p-1,i} + \varepsilon_i$$

- Assuming $\varepsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$ and $x_{k,i}$'s are fixed
- Then, the **likelihood function** is

$$L(\beta_0, \beta_1, ..., \beta_{p-1}, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{ -\frac{1}{2\sigma^2} (y_i - \beta_0 - \beta_1 x_{1,i} - \cdots - \beta_{p-1} x_{p-1,i})^2 \right.$$

- And **2 times the log-likelihood** is

$$2l(\beta_0, \beta_1, ..., \beta_{p-1}, \sigma^2) = -n \ln(2\pi) - n \log(\sigma^2) - \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1,i} - \cdots - \beta_{p-1} x_{p-1,i})^2$$

- This is **maximised** at the OLS estimator, with $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n}$
- (We typically use MSE based on $(n - p)$ rather than n when estimating σ^2)

R-squared for goodness of fit

- "R-squared" (R^2) is the **proportion of variation** in the observed y_i's **explained** by the regression line.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{j=1}^{n} (y_j - \bar{y})^2} = \frac{SSR}{SSTo} = 1 - \frac{SSE}{SSTo}$$

where

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \qquad \text{Regression sum of squares}$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad \text{Error sum of squares}$$

$$SSTo = \sum_{i=1}^{n} (y_i - \bar{y})^2 \qquad \text{Total sum of squares}$$

R-squared for goodness of fit

- In general, $\hat{y}_i = b_0 + b_1 x_{1,i} + \cdots + b_{p-1} x_{p-1,i}$
- The b_i coefficients are the OLS estimators of the corresponding β_i unknowns

- R^2 is just one available numerical summary measure of model fit

- Note that R-squared will **never decrease** when additional regressors are added
- So **R-squared is only good** for comparing regressions
  - For the same response variable y
  - And for models with the **same number** of regressors (predictors)

CLT-based tests and confidence intervals

- Use the **lm()** function in R for estimated coefficients and their (estimated) standard errors

Due to the availability of an appropriate CLT result

- Can undertake **hypothesis test** for individual regression coefficient β_k
- Can construct **confidence interval** for individual regression coefficient β_k
- for for any $k = 0, ..., p-1$.

CLT-based hypothesis tests

$$H_0 : \beta_k = 0 \;\; vs \;\; H_1 : \beta_k \neq 0$$

- Under H_0, $\frac{b_k}{s(b_k)}$ has (approximately) a t_{n-p} distribution

CLT-based confidence intervals

A $(1-\alpha) \times 100\%$ Confidence interval for β_k is given by:

$$b_k \pm t_{\alpha/2, n-p} SE(b_k)$$

Bootstrap-based confidence intervals for regression

Bootstrap-based CI for a regression coefficient

1. Create an $(R \times p)$ matrix to store all regression coefficients from each bootstrap sample
   - R rows, one for each for bootstrap sample
   - p columns for number of regression coefficients in model
2. Repeat for each bootstrap replication
   - Sample **rows** of the data frame **with replacement**
   - **Fit the regression model** for each bootstrap sample
   - **Save all regression coefficients** in a row of the storage matrix
3. Compute bootstrap-based confidence interval for β_k
   - Select the 2.5% and 97.5% quantiles of the column $(k + 1)$ corresponding to β_k
   - (These are the end points of the bootstrap-CI for β_k)

Can use for each $k = 0, 1, 2, ..., p - 1$

Permutation tests for regression

We used a **permutation test** previously (with two independent samples) to formally decide if

- two groups have the same mean
- two groups have the same proportion

- The idea was to **break** the connection between group and promotion outcome
- To **force null hypothesis** (H_0 : no difference between groups) **to hold**
- And generate an approximate **sampling distribution of the test statistic** $\bar{X}_1 - \bar{X}_2$

For a **regression**, we test $H_0 : \beta_k = 0$ vs $H_1 : \beta_k \neq 0$

- For any $k = 1, 2, ..., p - 1$ (note no testing for β_0)
- we **need to break any existing association between regressor x_k and y in our sample**
- We do this via permutations (shuffling) the values of x_k over different observations

Permutation-based hypothesis tests for regression

Procedure for coefficient β_k ($k > 1$) based on R permuted samples

Want to test, **for some** $k = 1, 2, ..., p-1$ **(but not for** $k = 0$**)**,

$$H_0 : \beta_k = 0 \ \ vs \ \ H_1 : \beta_k \neq 0$$

1. Create an $(R \times 1)$ **vector** to store all b_k regression coefficients from each permutation sample

2. Repeat for each permutation replication
   - Permute column of tibble containing regressor x_k **only** - keep all other rows of the data frame in order
   - **Fit the regression model** to the permuted data frame
   - **Save** b_k in the i^{th} entry of the storage vector

3. Plot a histogram of the permutation-generated b_k values
   - Draw a vertical red-line correpsonding to the data-based b_k value
   - Compute percentage of permutation-generated b_k values exceeds the data-based b_k value
   - (Can do one-sided or two-sided tests)

Residual plots

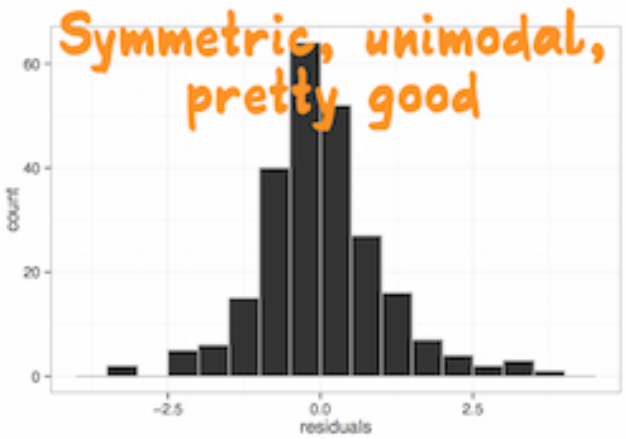
Check your residuals using visualisation techniques

Critical plots to assess model fit include

- Histogram of residuals
  - for a good fit the shape should be relatively **symmetric and bell-shaped**
- Do a QQplot of **theoretical normal quantiles** against **residuals**
  - (“Normal probability plot of the residuals”)
- Plot the residuals against **fitted values**
- Plot the residuals against available regressors (any x’s included or not included)
  - a good fit means should there should not be any obvious patterns

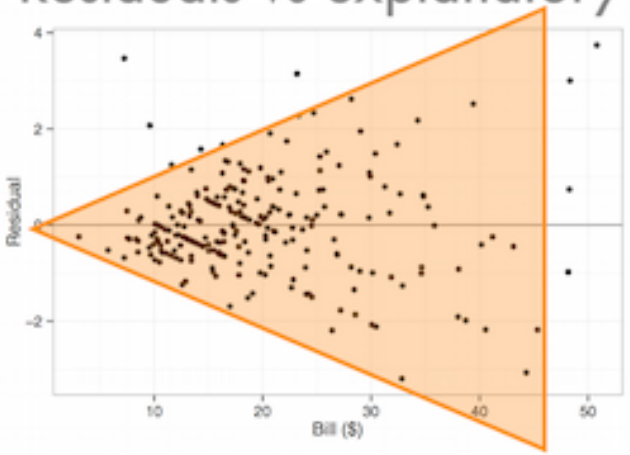
Residual plots to check model fit - what to look for

Histogram of residuals



Normal probability plot

Residuals vs explanatory variable



Plot exhibits heteroskedasticity, suggests that tip variability depends size of the bill.

What if residual plots show a problem?

- Consider possible **need to transform** y using logarithm or other function
  - Shift values first, then take logarithm to avoid log of a negative number
  - Other transformations are possible (e.g. power transform y^c or y^{-c})
- Consider **adding other regressors**
- Consider **alternative loss function** (e.g. “Weighted least squares”) for selecting parameters
  - May be equivalent to assuming different error distribution
- Consider if you have any **influential observations**
  - Check **Leverage** and **Cook’s D** (See below)

Leverage

h_{ii} is the i^{th} diagonal element of the **hat matrix** H:

$$H = X(X^T X)^{-1} X^T$$

where X is the **design matrix** containing all of the regressors

$$\text{SLR: } X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \qquad \text{general LR: } X = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{p-1,1} \\ 1 & x_{1,2} & \cdots & x_{p-1,2} \\ \vdots & \vdots & & \\ 1 & x_{1,n} & \cdots & x_{p-1n1} \end{bmatrix}$$

- Intuitively, observations far from $\bar{x}$ will have higher **leverage**
- $\Rightarrow$ They have **greater influence on the fitted regression function**
- $\Rightarrow$ Changing their y value a little can **substantially effect** the fitted line

About that hat matrix. . .

Where does the hat matrix H come from?

In general (multiple) linear regression, using vector notation, we have

$$Y = X\beta + \varepsilon$$

where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{p-1,1} \\ 1 & x_{1,2} & \cdots & x_{p-1,2} \\ \vdots & \vdots & & \\ 1 & x_{1,n} & \cdots & x_{p-1n1} \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

- The OLS estimator is $\hat{\beta} = (X'X)^{-1}X'Y$, and predictions at the observed X is given by
$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y = HY$$
- Notice that $\hat{Y} = HY$. This is why H is called the "hat" matrix!

Cook's D

- Another **influence measure** for observations that uses the response variable

$$D_i = \frac{e_i^2}{pMSE} \frac{h_{ii}}{(1-h_{ii})^2}$$

- e_i is the i^{th} residual
- $p =$ number of explanatory variables (regressors, including the intercept)
- MSE is the mean squared error of the linear model ($MSE = SSE/(n-p)$)
- As a **rule of thumb** check any point with Cook's D value greater than $2p/n$ (same as for leverage)

Leave One Out Cross Validation (LOOCV)

- **LOOCV is a method for validating a model**
- **Leverage** is related to **LOOCV** for regression models

$$LOOCV = \frac{1}{n}\sum_{i=1}^{n} e_{[i]}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_{[i]})^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{e_i}{1-h_{ii}}\right)^2$$

Here

- $e_{[i]} = y_i - \hat{y}_{[i]}$ is the i^{th} **case-deleted residual**
- $\hat{y}_{[i]}$ is the **predicted** value for the i^{th} observation
  - using model estimated with the i^{th} case deleted
- e_i is the **OLS residual** based on all of the data, and
- h_{ii} is the i^{th} **leverage** value from the OLS fit
- $\Rightarrow$ This means we can calculate **LOOCV** without fitting all n models!
  - (rather than fitting the n different regressions that leave out just one observation)

How to get all this out of R?

- Fit models using the *lm*() function
- Use *summary*() to extract from fitted results
  - **e.g. MSE, regression coefficients and standard errors, t-stats and MSE**
- Use the **broom** package to *augment*() your tibble with fitted values, leverage, Cook's D
  - Other useful broom package functions: *tidy*() and *glanc*() to organise model output
- Sort tibble using *dplyr::arrange*()