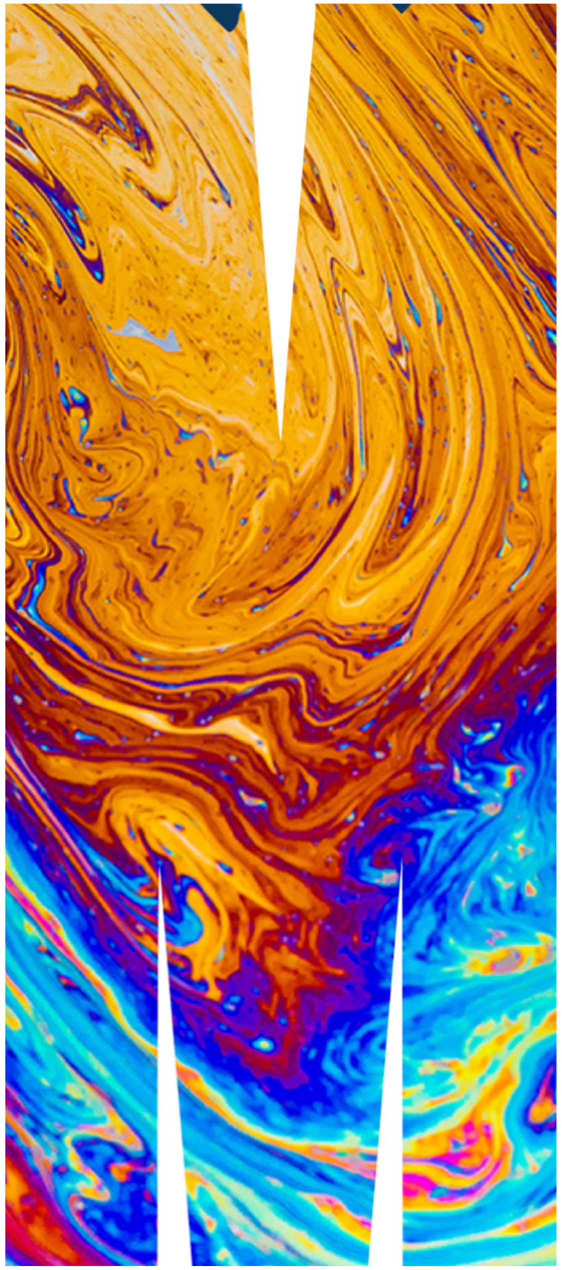
ETC5521: Exploratory Data Analysis

Sculpting data using models, checking assumptions, co-dependency and performing diagnostics

Lecturer: *Di Cook*

ETC5521.Clayton-x@monash.edu

Week 11 - Session 1



Models help focus on the structure

before focus

after focus, we can see it's a rare Eurasian Hoopoe.

Parametric regression

Parametric regression

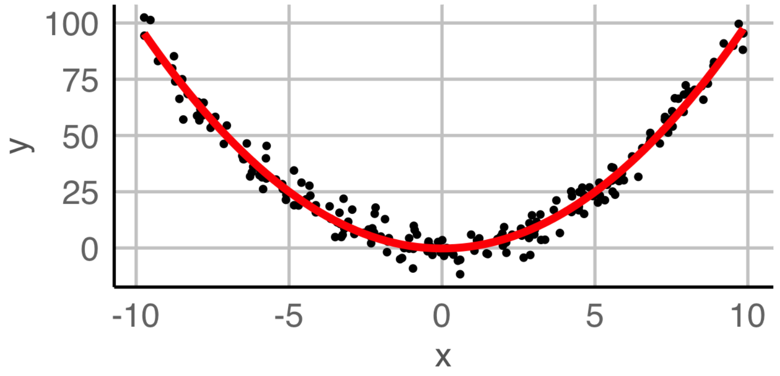
- **Parametric** means that the researcher or analyst assumes in advance that the data fits some type of distribution (e.g. the normal distribution).
- E.g. one may assume that

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i ,$$

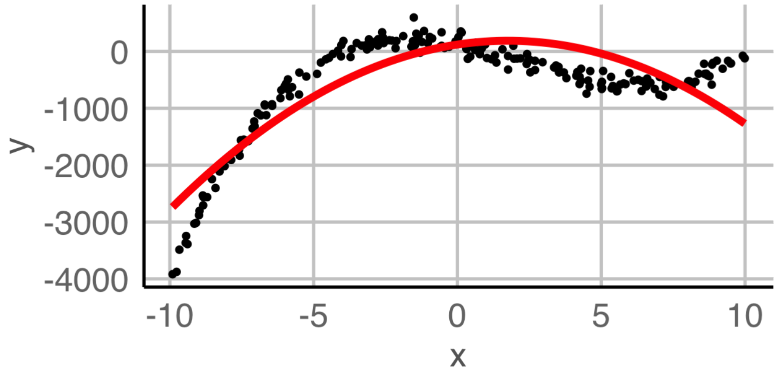
where $\epsilon_i \sim \mathrm{NID}(0, \sigma^2)$ for $i = 1, \ldots, n$,

  - $\text{red} = \text{to estimate}$
  - $\text{blue} = \text{observed}$
- Because some type of distribution is assumed in advance, parametric fitting can lead to fitting a smooth curve that misrepresents the data.

Examples



Assuming a quadratic fit:



Simulating data from parametric models

- Say a model is

$$y = x^2 + e, \qquad e \sim N(0, 2^2).$$

- Then we have

$$y \mid x \sim N(x^2, 2^2).$$

Simulating data from parametric models

- Say a model is

$$y = x^2 + e, \qquad e \sim N(0, 2^2).$$

- Then we have

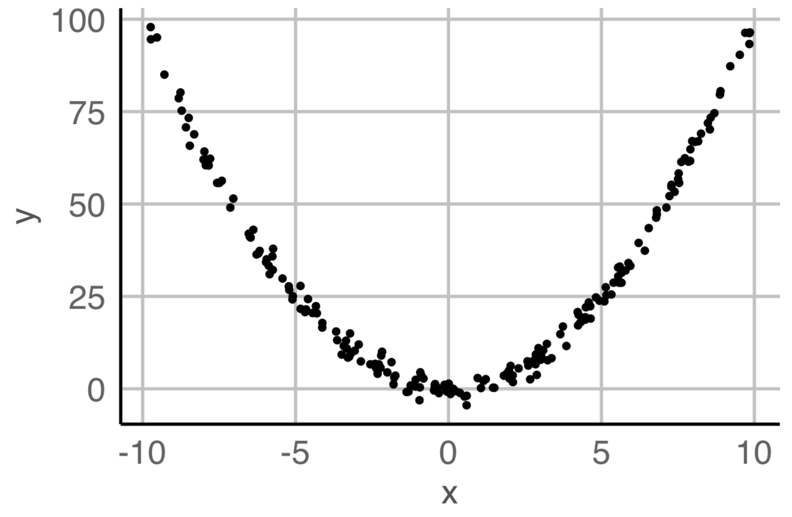
$$y \mid x \sim N(x^2, 2^2).$$

- Let's draw 200 observations from this model.
- Suppose that $x \in (-10, 10)$ and that we have uniform coverage over the support.
- The response y is generated as per above model.

```r
set.seed(1)
df <- tibble(id = 1:200) %>%
        mutate(x = runif(n(), -10, 10),
               y = x^2 + rnorm(n(), 0, 2))
```

Plotting this:

```r
ggplot(df, aes(x, y)) +
  geom_point()
```



Logistic regression

Logistic regression

- Not all parametric models assume Normally distributed errors nor continuous responses.
- Logistic regression models the relationship between a set of explanatory variables $(x_{i1}, \dots, x_{ik})$ and a set of **binary outcomes** Y_i for $i = 1, \dots, n$.
- We assume that $Y_i \sim B(1, p_i) \equiv \text{Bernoulli}(p_i)$ and the model is given by

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{i1} + \dots + \beta_k x_{ik}.$$

- Taking the exponential of both sides and rearranging we get

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \dots + \beta_k x_{ik})}}.$$

- The function $f(p) = \ln\left(\frac{p}{1 - p}\right)$ is called the **logit** function, continuous with range $(-\infty, \infty)$, and if p is the probablity of an event, $f(p)$ is the log of the odds.

Representation of data for binary outcomes

Data:

```
mock_df
```

```
## # A tibble: 18 × 5
##    Patient Smoker Sex    Cancer CancerBinary
##    <fct>   <fct>  <fct>  <fct>         <dbl>
##  1 1       Yes    Female No                0
##  2 2       Yes    Male   No                0
##  3 3       No     Female Yes               1
##  4 4       Yes    Male   No                0
##  5 5       Yes    Female Yes               1
##  6 6       No     Female No                0
##  7 7       Yes    Female Yes               1
##  8 8       No     Female No                0
##  9 9       No     Female No                0
## 10 10      No     Male   No                0
## 11 11      Yes    Male   No                0
## 12 12      Yes    Female Yes               1
## 13 13      Yes    Male   No                0
## 14 14      Yes    Female No                0
## 15 15      No     Male   Yes               1
## 16 16      No     Female Yes               1
## 17 17      No     Male   No                0
## 18 18      No     Male   Yes               1
```

Summarised data:

```
mock_sumdf
```

```
## # A tibble: 4 × 4
## # Groups:   Smoker [2]
##   Smoker Sex    Cancer Total
##   <fct>  <fct>   <int> <int>
## 1 No     Female      2     5
## 2 No     Male        2     4
## 3 Yes    Female      3     5
## 4 Yes    Male        0     4
```

- The summarised data here give the same information as the original data, except you lost the patient number
- Note the sample size, n, is larger than the number of rows in the summarised data

Logistic regression in R

- Fitting logistic regression models in R depend on the form of input data

```
glm(Cancer ~ Smoker + Sex,
    family = binomial(link = "logit"),
    data = mock_df)

##
## Call:  glm(formula = Cancer ~ Smoker + Sex, family = bi
##     data = mock_df)
##
## Coefficients:
## (Intercept)    SmokerYes      SexMale
##      0.2517      -0.5034      -1.1145
##
## Degrees of Freedom: 17 Total (i.e. Null);  15 Residual
## Null Deviance:        24.06
## Residual Deviance: 22.61     AIC: 28.61
```

```
glm(cbind(Cancer, Total - Cancer) ~ Smoker + Sex,
    family = binomial(link = "logit"),
    data = mock_sumdf)

##
## Call:  glm(formula = cbind(Cancer, Total - Cancer) ~ Smok
##     data = mock_sumdf)
##
## Coefficients:
## (Intercept)    SmokerYes      SexMale
##      0.2517      -0.5034      -1.1145
##
## Degrees of Freedom: 3 Total (i.e. Null);  1 Residual
## Null Deviance:        5.052
## Residual Deviance: 3.604     AIC: 15.82
```

Simulating from a logistic regression model Part 1

- Let's suppose that the probability of having cancer are the following:
  - 0.075 for women smokers
  - 0.045 for men smokers
  - 0.005 for women non-smokers
  - 0.003 for men non-smokers
- We'll sample 500 people for each group
- Remember that under the logistic regression model, we assumed that $Y_i \sim B(1, p_i)$

```
df <- tibble(id = 1:2000) %>%
  mutate(Smoker = rep(c("Yes", "No"), each = n() / 2),
         Sex = rep(c("Female", "Male"), times = n() / 2)) %>%
  rowwise() %>%
  mutate(CancerBinary =
          case_when(Smoker=="Yes" & Sex=="Female" ~ rbinom(1, 1, 0.075)
                    Smoker=="Yes" & Sex=="Male" ~ rbinom(1, 1, 0.045),
                    Smoker=="No" & Sex=="Female" ~ rbinom(1, 1, 0.005),
                    Smoker=="No" & Sex=="Male" ~ rbinom(1, 1, 0.003)),
         Cancer = ifelse(CancerBinary, "Yes", "No"))

df %>%
  filter(Cancer=="Yes")
```

```
## # A tibble: 53 × 5
## # Rowwise:
##       id Smoker Sex    CancerBinary Cancer
##    <int> <chr>  <chr>         <int> <chr>
##  1    27 Yes    Female            1 Yes
##  2    32 Yes    Male              1 Yes
##  3    47 Yes    Female            1 Yes
##  4    83 Yes    Female            1 Yes
##  5   129 Yes    Female            1 Yes
##  6   136 Yes    Male              1 Yes
##  7   149 Yes    Female            1 Yes
##  8   218 Yes    Male              1 Yes
##  9   245 Yes    Female            1 Yes
```

Simulating from a logistic regression model Part 2

- At times, you may want to **simulate the summary data directly** instead of the individual data
- Recall that if $Y_i \sim B(1, p)$ for $i = 1, \dots k$ and Y_is are independent,

$$S = Y_1 + Y_2 + \dots + Y_k \sim B(k, p)$$

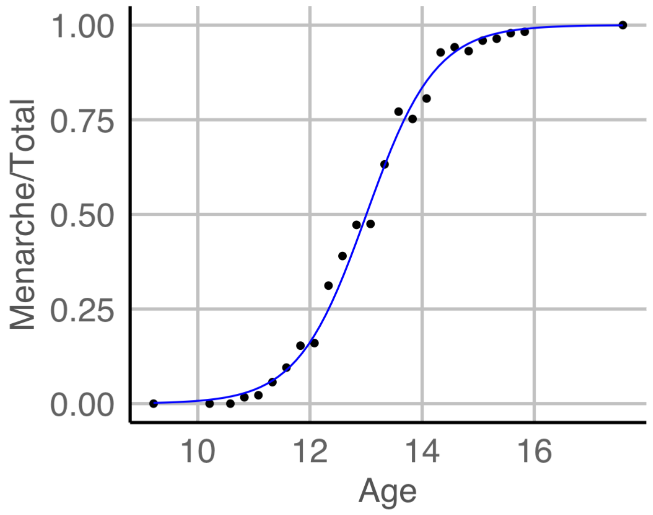
```
expand_grid(Smoker = c("Yes", "No"), Sex = c("Female", "Male")) %>%
  rowwise() %>%
  mutate(Cancer =
           case_when(Smoker=="Yes" & Sex=="Female" ~ rbinom(1, 500, 0.07
                     Smoker=="Yes" & Sex=="Male" ~ rbinom(1, 500, 0.045)
                     Smoker=="No" & Sex=="Female" ~ rbinom(1, 500, 0.005
                     Smoker=="No" & Sex=="Male" ~ rbinom(1, 500, 0.003))
         Total = 500)
```

```
## # A tibble: 4 × 4
## # Rowwise:
##   Smoker Sex    Cancer Total
##   <chr>  <chr>   <int> <dbl>
## 1 Yes    Female     35   500
## 2 Yes    Male       23   500
## 3 No     Female      0   500
## 4 No     Male        3   500
```

Case study 1 Menarche

- In 1965, the average age of 25 homogeneous groups of girls was recorded along with the number of girls who have reached menarche out of the total in each group.

data R



Simulating data from a fitted logistic regression model Part 1

- Suppose we want to simulate from the fitted model

- We first fit the fitted model

```
fit1 <-
  glm(cbind(Menarche, Total - Menarche) ~ Age,
      family = "binomial",
      data = menarche)
(beta <- coef(fit1))

## (Intercept)         Age
##  -21.226395    1.631968
```

- The fitted regression model is given as:

$$\text{logit}(\hat{p_i}) = \hat{\beta_0} + \hat{\beta_1} x_{i1} .$$

- Rearranging we get

$$\hat{p_i} = \frac{1}{1 + e^{-(\hat{\beta_0} + \hat{\beta_1} x_{i1})}} .$$

- Simulating from first principles:

```
menarche %>%
  rowwise() %>%
  mutate(
    phat = 1/(1 + exp(-(beta[1] + beta[2] * Age))),
    simMenarche = rbinom(1, Total, phat))

## # A tibble: 25 × 5
## # Rowwise:
##      Age Total Menarche    phat simMenarche
##    <dbl> <dbl>    <dbl>   <dbl>       <int>
##  1  9.21   376        0 0.00203           1
##  2 10.2    200        0 0.0103            3
##  3 10.6     93        0 0.0187            2
##  4 10.8    120        2 0.0279            3
##  5 11.1     90        2 0.0413            1
##  6 11.3     88        5 0.0609            6
##  7 11.6    105       10 0.0888            9
##  8 11.8    111       17 0.128            12
##  9 12.1    100       16 0.181            17
## 10 12.3     93       29 0.249            23
## # ℹ 15 more rows
```

Simulating data from a fitted logistic regression model Part 2

- An easier way to do this is to use the `simulate` function which works for many model objects in R
- Below it's simulating 3 sets of responses (i.e. counts of "success" and "failure" events) from `fit1` logistic model object

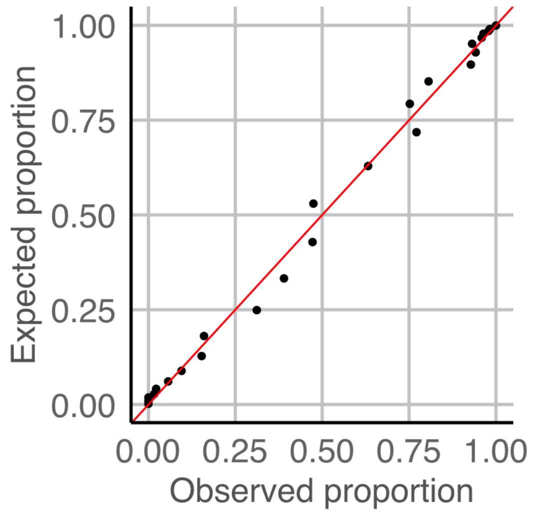
```
simulate(fit1, nsim = 3)

##    sim_1.Menarche sim_1.V2 sim_2.Menarche sim_2.V2 sim_3.Menarche sim_3.V2
## 1               0      376              0      376              0      376
## 2               2      198              1      199              0      200
## 3               4       89              0       93              3       90
## 4               4      116              2      118              6      114
## 5               8       82              5       85              3       87
## 6               6       82              3       85              6       82
## 7              14       91              7       98              5      100
## 8              13       98             14       97             16       95
## 9              21       79             18       82             20       80
## 10             25       68             21       72             27       66
## 11             47       53             32       68             33       67
## 12             37       71             43       65             41       67
## 13             47       52             54       45             59       40
```

Diagnostics for logistic regression models

- One diagnostic is to compare the observed and expected proportions under the logistic regression fit.

```
df1 <- menarche %>%
  mutate(
    pexp = 1/(1 + exp(-(beta[1] + beta[2] * Age))),
    pobs = Menarche / Total)
```



Diagnostics for logistic regression models

- Goodness-of-fit type test is used commonly to assess the fit as well.
- E.g. Hosmer–Lemeshow test, where test statistic is given as

$$H = \sum_{i=1}^{r} \left(\frac{(O_{1i} - E_{1g})^2}{E_{1i}} + \frac{(O_{0i} - E_{0g})^2}{E_{0i}} \right)$$

where O_{1i} (E_{1i}) and O_{0i} (E_{0i}) are observed (expected) frequencies for successful and non-successful events for group i, respectively.

```
vcdExtra::HLtest(fit1)

## Hosmer and Lemeshow Goodness-of-Fit Test
##
## Call:
## glm(formula = cbind(Menarche, Total - Menarche) ~ Age, family = "binomial",
##     data = menarche)
##  ChiSquare df   P_value
##  0.1088745  8 0.9999996
```

Diagnostics for linear models

Assumptions for linear models

For $i \in \{1, \ldots, n\}$,

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \epsilon_i,$$

where $\epsilon_i \sim \text{NID}(0, \sigma^2)$ or in matrix format,

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \text{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

where

- `$\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$`,
- `$\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)^\top$`,
- `$\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^\top$`, and
- `$\mathbf{X} = \begin{bmatrix} \mathbf{1}_n & \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_k \end{bmatrix}$`, where
- `$\boldsymbol{x}_j = (x_{1j}, \ldots, x_{nj})^\top$` for `$j \in \{1, \ldots, k\}$`

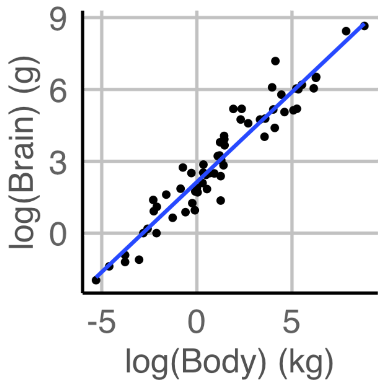
This means that we assume

1. $E(\epsilon_i) = 0$ for $i \in \{1, \ldots, n\}$.
2. $\epsilon_1, \ldots, \epsilon_n$ are independent.
3. $Var(\epsilon_i) = \sigma^2$ for $i \in \{1, \ldots, n\}$ (i.e. homogeneity).
4. $\epsilon_1, \ldots, \epsilon_n$ are normally distributed.

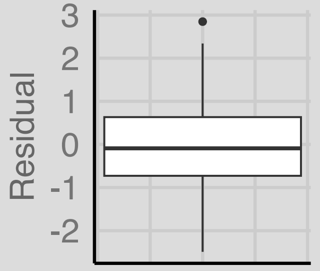
So how do we check it?

Model diagnostics for linear models

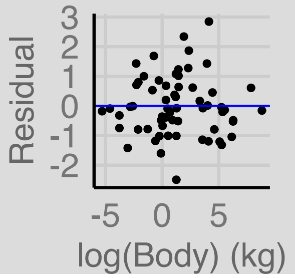
Plot Y_i vs x_i to see if there is ≈ a linear relationship between Y and x.



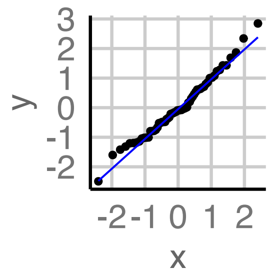
A boxplot of the residuals R_i to check for symmetry.



To check the homoscedasticity assumption, plot R_i vs x_i. There should be no obvious patterns.



A normal Q-Q plot, i.e. a plot of the ordered residuals vs $\Phi^{-1}\left(\frac{i}{n+1}\right)$.



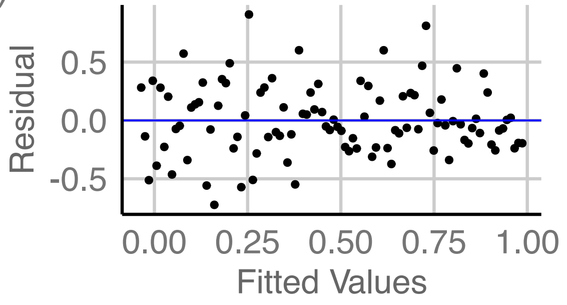
Assessing (A1) $E(\epsilon_i) = 0$ for $i = 1, \ldots, n$

- It is a property of the least squares method that

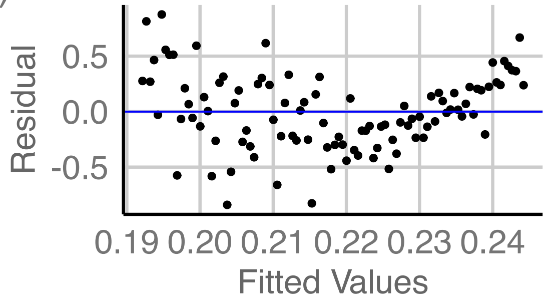
$$\sum_{i=1}^{n} R_i = 0, \quad \text{so} \quad \bar{R}_i = 0$$

  for $R_i = Y_i - \hat{Y_i}$, hence (A1) will always appear valid "overall".
- Trend in residual versus fitted values or covariate can indicate "local" failure of (A1).
- What do you conclude from the following plots?

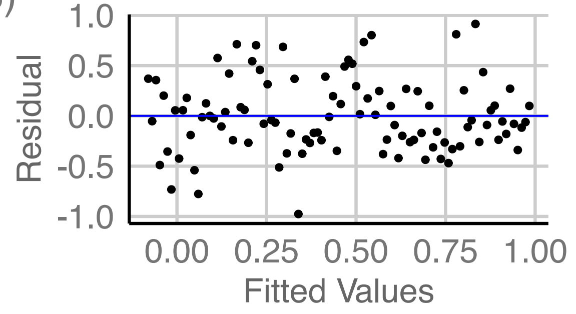
(1)



(2)



(3)



Assessing (A2)-(A3)

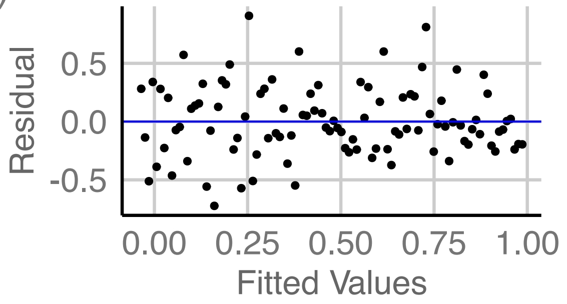
(A2) $\epsilon_1, \ldots, \epsilon_n$ are independent

- If (A2) is correct, then residuals should appear randomly scattered about zero if plotted against fitted values or covariate.
- Long sequences of positive residuals followed by sequences of negative residuals in R_i vs x_i plot suggests that the error terms are not independent.

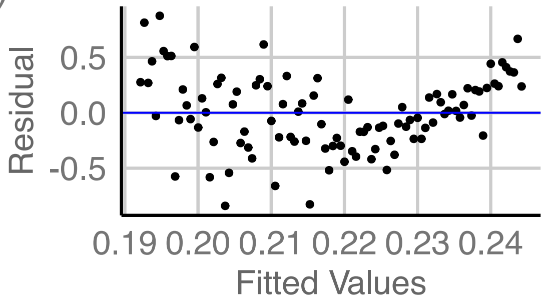
(A3) $Var(\epsilon_i) = \sigma^2$ for $i = 1, \ldots, n$

- If (A3) holds then the spread of the residuals should be roughly the same across the fitted values or covariate.

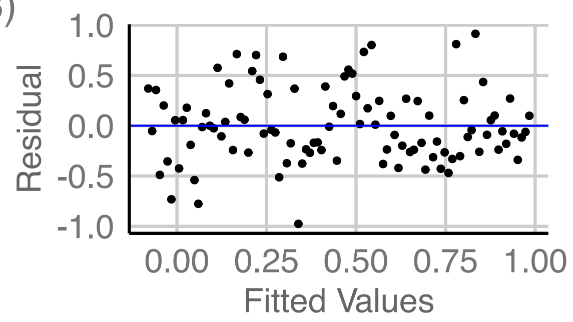
(1)



(2)



(3)



Assessing (A4) $\epsilon_1, \ldots, \epsilon_n$ are normally distributed

Q-Q Plots

- The function `qqnorm(x)` produces a Q-Q plot of the ordered vector `x` against the quantiles of the normal distribution.
- The n chosen normal quantiles $\Phi^{-1}\left(\frac{i}{n+1}\right)$ are easy to calculate but more sophisticated ways exist:
  - $\frac{i}{n+1} \mapsto \frac{i-3/8}{n+1/4}$, default in `qqnorm`.
  - $\frac{i}{n+1} \mapsto \frac{i-1/3}{n+1/3}$, recommended by Hyndman and Fan (1996).

In R

```
fit <- lm(y ~ x)
```

By "hand"

```
plot(qnorm((1:n) / (n + 1)), sort(resid(fit)))
```

By `base`

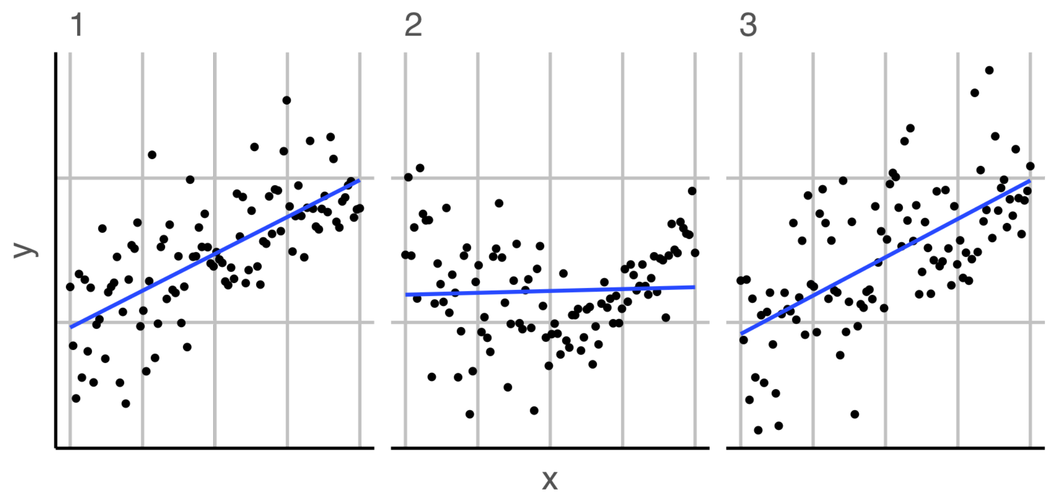
```
qqnorm(resid(fit))
qqline(resid(fit))
```

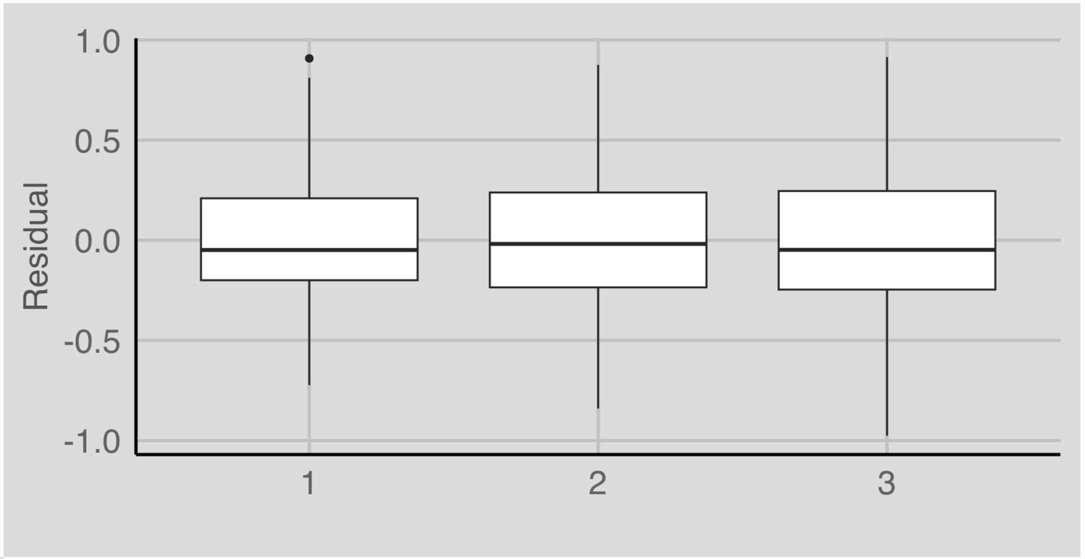
By `ggplot2`

```
data.frame(residual = resid(fit)) %>%
  ggplot(aes(sample = residual)) +
  stat_qq() + stat_qq_line(color="blue")
```

Reference: Hyndman and Fan (1996). *Sample quantiles in statistical packages*, American Statistician, 50, 361--365.

Examining simulated data

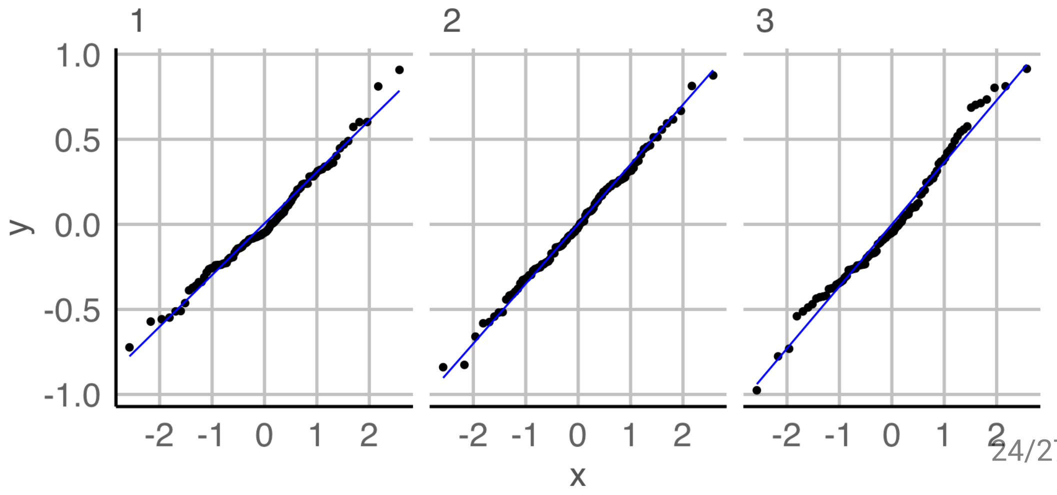




Simulation scheme

```
n <- 100
x <- seq(0, 1, length.out = n)
y1 <- x + rnorm(n) / 3                       #  Linear
y2 <- 3 * (x - 0.5) ^ 2 +
  c(rnorm(n / 2)/3, rnorm(n / 2)/6)         #  Quadratic
y3 <- -0.25 * sin(20 * x - 0.2) +
  x + rnorm(n) / 3                           #  Non-linear

M1 <- lm(y1 ~ x); M2 <- lm(y2 ~ x); M3 <- lm(y3 ~ x)
```



Take away messages

- Parametric models assume some distribution in advance
- Logistic models can be used to model explanatory variables with binary outcomes
- You should be able to simulate from parametric models
- You can perform basic model diagnostics
- You can use simulation to analyse model properties

Resources and Acknowledgement

- These slides were originally created by Dr Emi Tanaka, and modified by Dr Michael Lydeamore.
- Some of these slides were inspired by STAT3012 Applied Linear Models at The University of Sydney by Prof Samuel Muller
- Cook & Weisberg (1994) "An Introduction to Regression Graphics"
- Data coding using `tidyverse` suite of R packages
- Slides constructed with `xaringan`, remark.js, `knitr`, and R Markdown.

Lecturer: *Di Cook*

ETC5521.Clayton-x@monash.edu

Week 11 - Session 1