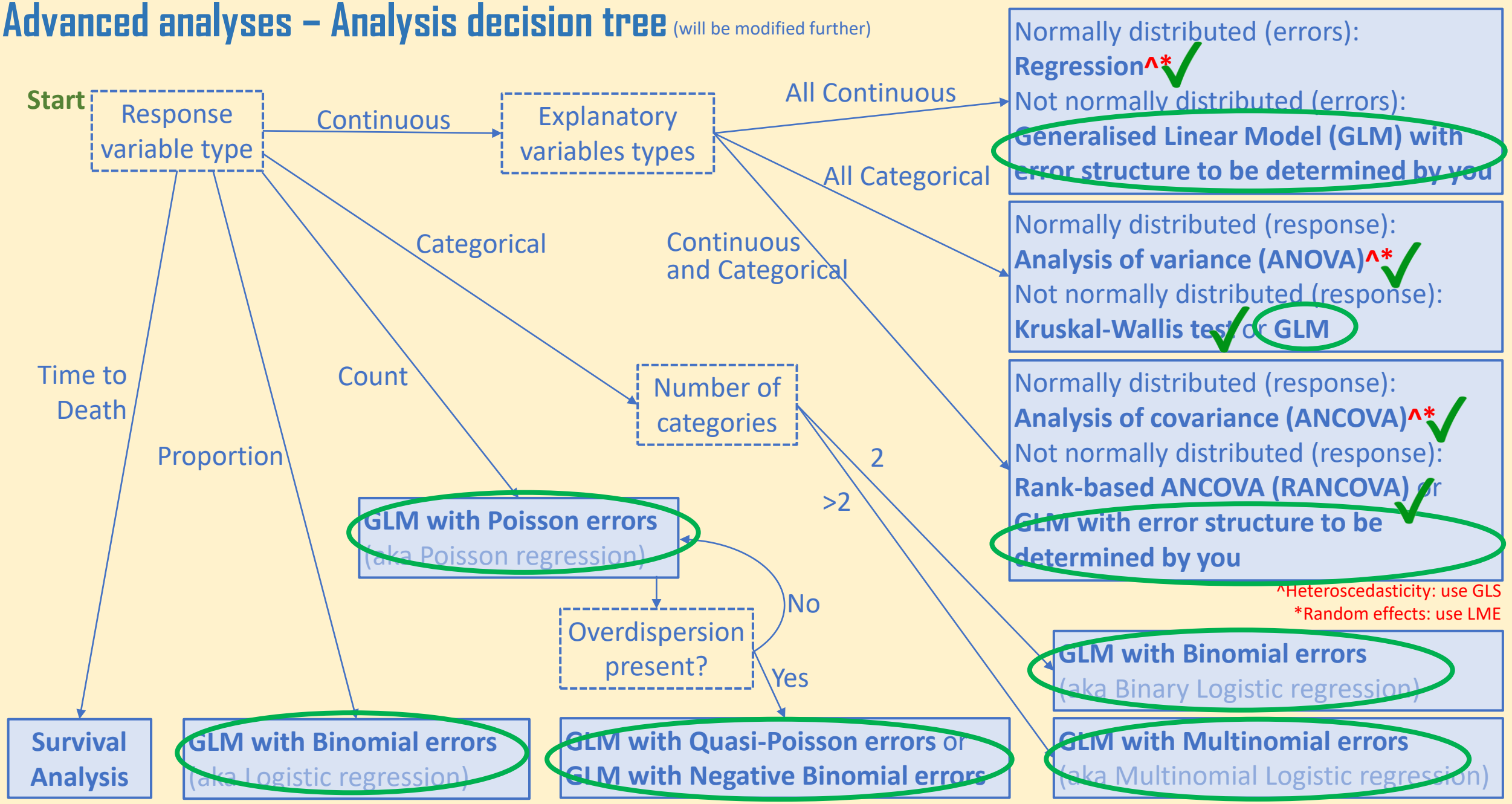
GLM

Lecture 7

LSM3257

AY22/23; Sem 2 | Ian Z.W. Chan

Advanced analyses – Analysis decision tree (will be modified further)



Summary (Learning Objectives)

Generalised Linear Models (GLM)

- Theory: Link functions, linear predictors and error distributions
    - Error distributions and variable types
    - Least Squares vs. Maximum Likelihood
- Poisson for count data
    - Quasipoisson/Negative Binomial for overdispersion
    - Simplifying, comparing, checking and interpreting models
- Binomial for proportion/categorical data (2 categories)
    - Quasibinomial for overdispersion
    - Simplifying, comparing, checking and interpreting models
- Multinomial for categorical data (>2 categories)
- Quasi as a last resort for non-normally distributed continuous data

GLM

Theoretical background

How are GLMs related to Regression, ANOVA and ANCOVA?

Another name for normally-distributed

Regression

All equivalent to

ANOVA

ANCOVA

Linear Models that have **Gaussian errors**.

<u>Assumptions</u>

Response variable: one continuous variable with normally distributed errors.

Explanatory variables: one or more continuous or categorical variables.

Homoscedasticity and normality are assumed.

But Biology is messy!: there are many types of response variables with non-normal, heteroscedastic error distributions (aka error structures).

Nelder & Wedderburn (1972) created a **<u>Generalised</u> Linear Model** (GLM) to analyse these different variables: you just need to **specify an appropriate error distribution** for the type of variable you have.

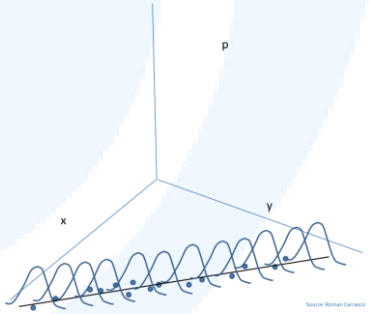
What is a GLM?

A GLM uses a **linear predictor** to model values of the response variable using a **link function and an assumed error distribution**. An example formula is shown below (note the arrows instead of an equals sign).

B) Activation function: convert the values of the linear predictor into the type of values we are expecting in our data.
Example: if our data is a proportion, we need values between 0 and 1. But we know a linear predictor can take values from -∞ to +∞. So we need a function to convert the values.

A) Linear predictor: the same thing we are familiar with in regression, ANOVA and ANCOVA: they have a constant and slopes to help explain the effects of the explanatory variables on the response variable. Note: we still want a <u>linear</u> predictor because it makes things easier to explain and understand.

$$Y_i \leftrightarrows \overbrace{\beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} \dots \beta_n X_{n,i}} + \varepsilon_i$$



C) Link function: convert values in our data to the values in the linear predictor. This is the <u>inverse</u> of the Activation function. We usually talk about Link functions rather than Activation functions.

D) Error distribution: different distributions are known to model (the errors from) different types of data better:
1) We choose a distribution based on the data we have.
2) When we choose a certain error distribution, we <u>usually</u> use a certain link function, this is called the "canonical link function".
(Note: Run `?family` in R to see the different types.)

De-mystifying "Link Functions"

What are the Activation (←) and Link (→) functions for the model below?

Left-Hand Side: our response variable

$$4 \overset{\overset{\text{* 1/2}}{\boxed{\text{Answer}}}}{\underset{\underset{\text{* 2}}{\boxed{\text{Answer}}}}{\leftrightarrows}} \underbrace{4 + 2(2)}$$

Right-Hand Side: our linear predictor

De-mystifying "Link Functions"

What are the Activation (←) and Link (→) functions for the model below?

sqrt(x)

Answer

$$3 \leftrightarrows 1 + 4(2)$$

Answer

y^2

De-mystifying "Link Functions"

What are the Activation (←) and Link (→) functions for the model below?

1/x

Answer

$$0.2 \leftrightarrows 1 + 2(2)$$

Answer

1/y

De-mystifying "Link Functions"

What are the Activation (←) and Link (→) functions for the model below?

log10X

Answer

$$2 \leftrightarrows 1 + 3(33)$$

Answer

10^y

De-mystifying "Link Functions"

What are the Activation (←) and Link (→) functions for the model below?

identity

Answer

$$13 \leftrightarrows 1 + 4(3)$$

Answer

identity

identity ... link function for linear models

Variable types and Error Distributions

Response Variable type	Error distribution	Canonical link function	Corresponding activation function
Continuous (normal)	Gaussian (aka Normal)	Identity: no conversion	Identity: no conversion
Count	Poisson	Log: $\ln(\text{count})$	e^x
Categorical / Proportion (p)	Binomial	Logit: $\ln\left(\frac{p}{1-p}\right)$ (aka "log-odds")	$\frac{1}{1+\frac{1}{e^x}}$
Time (T) to event (e.g. survival)	Exponential, Gamma	Inverse: $\frac{1}{T}$	Inverse: $\frac{1}{\mathrm{x}}$
Continuous (non-normal)	Quasi	Nil	Nil

Note: you can see what error families are available, and what link functions can be applied to each family using "?family".

How are GLM models chosen? – Maximum Likelihood Estimation

GLMs give a model that maximises the likelihood (L) of predicting the data. The exact formula is <u>different for every distribution</u>.

Example: for a normal distribution, the **likelihood function** is:

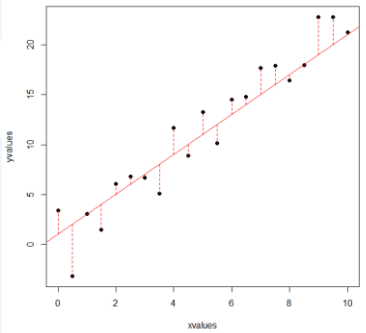
$$L = \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^N} \cdot e^{\left(\frac{-1}{2\sigma^2}\left(\underbrace{\sum_{i=1}^{N} \varepsilon^2}\right)\right)}$$

Notice that likelihood (L) is maximized when this term is minimized.

Recall that regression, ANOVA and ANCOVA use Least Squares, i.e. minimising SSE.

$$\text{SSE} = \sum_{i=1}^{N} \varepsilon^2$$

This term is actually the Sum of Squares! So with a Gaussian Distribution, maximizing Likelihood is minimizing Sum of Squares.



When (and only when) a normal distribution is used, Maximum Likelihood Estimation is equivalent to Least Squares, i.e. GLM with Gaussian errors = LM.

Residual deviance

To look at how much of our dataset is explained by our model, we calculate the **residual deviance** in the model (analogous to Residual Sum of Squares).

$$\text{Residual deviance} = 2 \cdot \left(\log L_{\text{saturated}} - \log L_{\text{fitted}}\right)$$

Likelihood of a saturated model, i.e. a model where there is one parameter for each datapoint and all the datapoints are therefore perfectly explained

The Likelihood of our model (that we just fit) that we want to calculate the Residual deviance for

Lower residual deviance is good (the model is better at predicting the data).

when comparing between the models, use anova + test = "Chisq"!!!!! or AIC

Note: This reduction in deviance between the saturated and fitted models is assumed to follow a chi-square (X^2) distribution (Wilk's Theorem), therefore we should use a **chi-square test** (using `anova(mod1,mod2,test="Chisq")`) or AIC (less confusing) when we compare different models during model simplification.

Fitting a GLM

General code:

Function to fit GLMs (from Base R)

Specifying an error distribution. We change this argument to specify different distributions. Check help(family) to see how.

```
modelObject=glm(Y~X1*X2+X3/X4,family=gaussian,data=dataset)
```

Object to save the fitted model to

Formula of response and explanatory variables. Can have interacting and nested variables.

Name of dataframe object containing all the variables to be used

The canonical link function is used by default

- You can specify a different link function, e.g.:

```
family=gaussian(link="inverse")
```

dont do this!

- To see what link functions are allowed for each distribution:

`?family` or `help(family)`

Note: I don't suggest you change the link function until you're more familiar with the theory and math behind GLMs.

```
Usage:

     family(object, ...)

     binomial(link = "logit")
     gaussian(link = "identity")
     Gamma(link = "inverse")
     inverse.gaussian(link = "1/mu^2")
     poisson(link = "log")
     quasi(link = "identity", variance = "constant")
     quasibinomial(link = "logit")
     quasipoisson(link = "log")
```

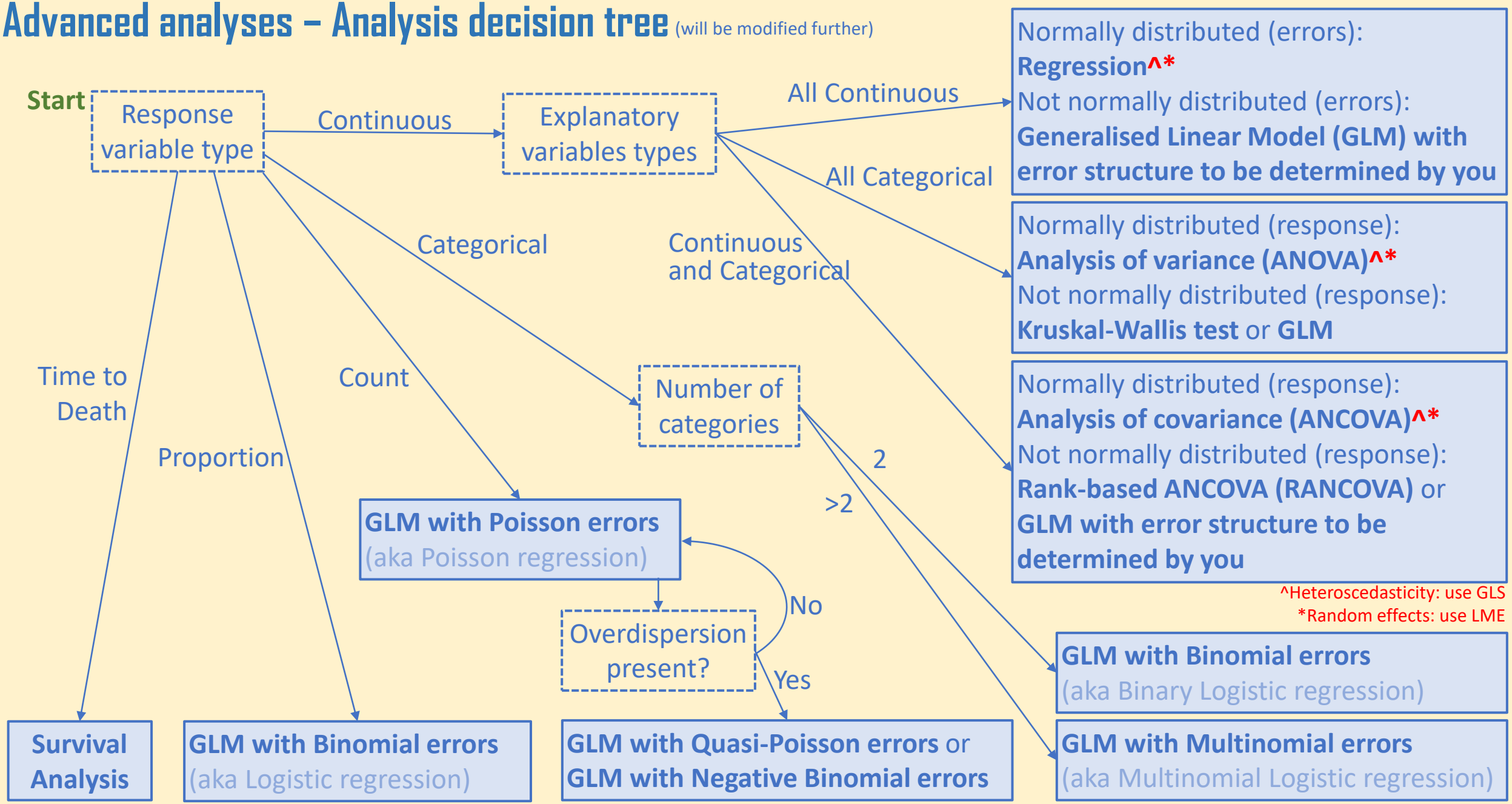
```
Arguments:

    link: a specification for the model link function.  This can be a
          name/expression, a literal character string, a length-one
          character vector, or an object of class '"link-glm"' (such as
          generated by 'make.link') provided it is not specified _via_
          one of the standard names given next.

          The 'gaussian' family accepts the links (as names)
          'identity', 'log' and 'inverse'; the 'binomial' family the
          links 'logit', 'probit', 'cauchit', (corresponding to
          logistic, normal and Cauchy CDFs respectively) 'log' and
          'cloglog' (complementary log-log); the 'Gamma' family the
          links 'inverse', 'identity' and 'log'; the 'poisson' family
          the links 'log', 'identity', and 'sqrt'; and the
          'inverse.gaussian' family the links '1/mu^2', 'inverse',
          'identity' and 'log'.

          The 'quasi' family accepts the links 'logit', 'probit',
          'cloglog', 'identity', 'inverse', 'log', '1/mu^2' and 'sqrt',
          and the function 'power' can be used to create a power link
          function.
```

Advanced analyses – Analysis decision tree (will be modified further)



Poisson GLM

Count data

When to use?

Your response variable is a count: e.g. the number of times an event happened.

- Cannot be less than zero (there is a bound, aka a limit, at zero).

- Zero is quite common.

- Variance is not constant, it increases with the mean.

- You don't know the number of times the event did not happen (if you did, it would be proportion data).

like if you took picture every minute and can tell which sec have animals and which dont

Examples:

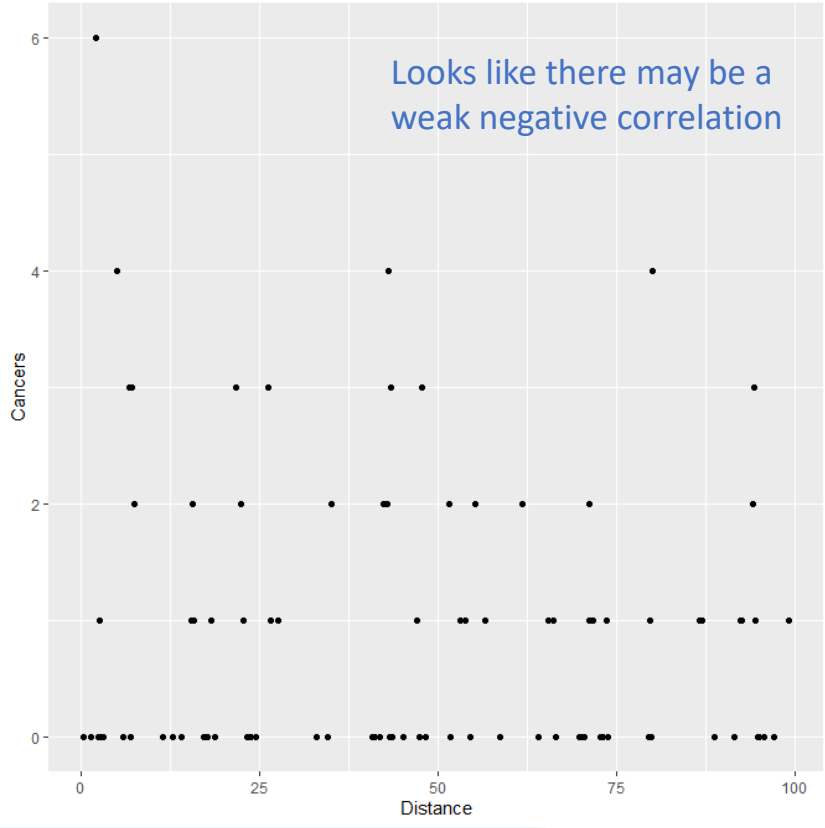
- **Number of cheetahs** observed within a nature reserve based on the size of the reserve and its connectivity to other reserves.

- **Number of cancers detected** explained by distance from the patient's home to a nuclear power plant.

Poisson Example 1: Fitting

Number of cancers detected explained by distance (km) from the patient's home to a nuclear power plant.

```
> str(d1)
'data.frame':   94 obs. of  2 variables:
 $ Cancers : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Distance: num  11.5 66.6 47.5 48.4 73.8 ...
```



#Load and visualise dataset

```
d1=read.table("clusters.txt",header=T)
str(d1)
library(ggplot2)
ggplot(d1,aes(x=Distance,y=Cancers))+geom_point()
```

#The response variable is a count, so we fit a GLM with Poisson errors

```
mod1.1=glm(Cancers~Distance,family=poisson,data=d1)
```

Poisson Example 1: Interpreting results

`summary(mod1.1)`

Distribution of the deviance for all datapoints (a bit hard to interpret because it's still in the units of the linear predictor).

```
> summary(mod1)

Call:
glm(formula = Cancers ~ Distance, family = poisson, data = d1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5504  -1.3491  -1.1553   0.3877   3.1304

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.186865   0.188728   0.990   0.3221
Distance    -0.006138   0.003667  -1.674   0.0941 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 149.48  on 93  degrees of freedom
Residual deviance: 146.64  on 92  degrees of freedom
AIC: 262.41

Number of Fisher Scoring iterations: 5
```

P-value is not significant.

The "effect size" (together with uncertainty estimate) of your explanatory variable, but **<u>in the units of the linear predictor</u>**.

if residual = df, its good
if residual > df, thats overdispersion

Number of trials required for R to numerically end up with the coefficients in your model.

total amount of variations

Deviance if it were a null model (i.e. with no explanatory variables). The residual deviance and d.f. for your model should be lower than this.

whatever is left unexplained

This is the residual deviance of your model and is used to judge whether there is overdispersion…
Residual deviance ≈ d.f.: good.
Residual deviance < d.f.: underdispersion, usually no need to correct.
Residual deviance > d.f.: overdispersion, correct for this by using quasipoisson or negative binomial.

We cannot use these results because **there is overdispersion**: specifically, Residual deviance > degrees of freedom.

Overdispersion

Guideline: **Residual Deviance:d.f. ratio < 1.5** is generally OK. Anything ≥ 1.5, you should think about correcting for overdispersion.

Overdispersion is extra unexplained variation in the response variable.

- The error distribution we use assumes a certain relationship between the variance and the mean in the data, e.g. Poisson assumes: variance = mean.

- If there is more variance than expected, we may have missed an important explanatory variable or the relationship between the explanatory and response variables is not linear.

If these are not the case, then the data may not fit the Poisson distribution we have chosen. Therefore, we can switch the error distribution to:

- **Quasipoisson**: uses quasi-likelihood which allows the variance to vary by fitting a dispersion parameter; when simplifying, use `anova(mod1,mod2,test="F")` (cannot use AIC);

OR

- **Negative binomial** (can use AIC)

Poisson Example 1: Re-Fitting with Quasipoisson errors

#Fit and view results

```
mod1.2=glm(Cancers~Distance,family=quasipoisson,data=d1)
summary(mod1.2)
```

```
> summary(mod2)

Call:
glm(formula = Cancers ~ Distance, family = quasipoisson, data = d1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5504  -1.3491  -1.1553   0.3877   3.1304

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.186865   0.235364   0.794    0.429
Distance    -0.006138   0.004573  -1.342    0.183

(Dispersion parameter for quasipoisson family taken to be 1.555271)

    Null deviance: 149.48  on 93  degrees of freedom
Residual deviance: 146.64  on 92  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

The estimated dispersion parameter: the variance was about 1.55x what was expected.

The effect is not significant. Perhaps we are missing an important explanatory variable?

Because there's no likelihood function, we cannot calculate an AIC value for quasipoisson models. That's why we have to use anova() and specify an F-test instead.

Residual deviance doesn't change but it's OK because we have accounted for it by using quasipoisson

Remember: to simplify with quasipoisson, use `anova(mod1,mod2,test="F")` not AIC.

Poisson Example 1: Interpreting results with Quasipoisson errors

#Check the link function

```
help(family)
```

```
Usage:

     family(object, ...)

     binomial(link = "logit")
     gaussian(link = "identity")
     Gamma(link = "inverse")
     inverse.gaussian(link = "1/mu^2")
     poisson(link = "log")
     quasi(link = "identity", variance = "constant")
     quasibinomial(link = "logit")
     quasipoisson(link = "log")
```

Therefore the Activation Function (to convert the linear predictor back to the original units of the y-variable) is the exponential

#Calculate number of cancers at Distance = 0 (i.e. the intercept)

```
exp(0.186865) #1.205
```

This is the value of the linear predictor when Distance = 0

#Number of cancers at Distance = 1 km

```
exp(0.186865-0.006138) #1.198
```

This is the change in the value of the linear predictor for a unit increase of Distance

```
> summary(mod2)

Call:
glm(formula = Cancers ~ Distance, family = quasipoisson, data = d1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5504  -1.3491  -1.1553   0.3877   3.1304

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.186865   0.235364   0.794    0.429
Distance    -0.006138   0.004573  -1.342    0.183

(Dispersion parameter for quasipoisson family taken to be 1.555271)

    Null deviance: 149.48  on 93  degrees of freedom
Residual deviance: 146.64  on 92  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

Because it's QUASIpoisson, there are no assumptions to check

Poisson Example 1: Re-Fitting with Negative Binomial errors

#Fit and view results

```
require(MASS)
mod1.3=glm.nb(Cancers~Distance,data=d1)
summary(mod1.3)
```

```
> summary(mod3)

Call:
glm.nb(formula = Cancers ~ Distance, data = d1, init.theta = 1.359466981,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3103  -1.1805  -1.0442   0.3065   1.9582

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.182490   0.252434   0.723    0.470
Distance    -0.006041   0.004727  -1.278    0.201

(Dispersion parameter for Negative Binomial(1.3595) family taken to be 1)

    Null deviance: 96.647  on 93  degrees of freedom
Residual deviance: 94.973  on 92  degrees of freedom
AIC: 253.19

Number of Fisher Scoring iterations: 1


              Theta:  1.359
          Std. Err.:  0.612

 2 x log-likelihood:  -247.191
```

Similar results, the effect is non-significant.

We can calculate AIC with negative binomial (that's why I personally prefer it over quasipoisson)

Residual deviance now about equal to d.f.

Note: to simplify, you can use `AIC(mod1,mod2)` or `anova(mod1,mod2,test="Chisq")` to compare models

Poisson Example 1: Interpreting results with Negative Binomial errors

#Number of cancers at Distance = 0 (i.e. the intercept)

```
exp(0.182490) #1.200
```

Conveniently provided in the summary

#Number of cancers at Distance = 1 (whatever units it is in)

```
exp(0.182490-0.006041) #1.193
```

```
> summary(mod3)

Call:
glm.nb(formula = Cancers ~ Distance, data = dl, init.theta = 1.359466981,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3103  -1.1805  -1.0442   0.3065   1.9582

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.182490   0.252434   0.723    0.470
Distance    -0.006041   0.004727  -1.278    0.201

(Dispersion parameter for Negative Binomial(1.3595) family taken to be 1)

    Null deviance: 96.647  on 93  degrees of freedom
Residual deviance: 94.973  on 92  degrees of freedom
AIC: 253.19

Number of Fisher Scoring iterations: 1


              Theta:  1.359
          Std. Err.:  0.612

 2 x log-likelihood:  -247.191
```

Results are very similar to the quasipoisson (1.205 and 1.198).

Poisson Example 1: Diagnostics to check models

Quasipoisson: not needed (because it doesn't make any assumptions).

Negative binomial: possible (but not widely practiced).

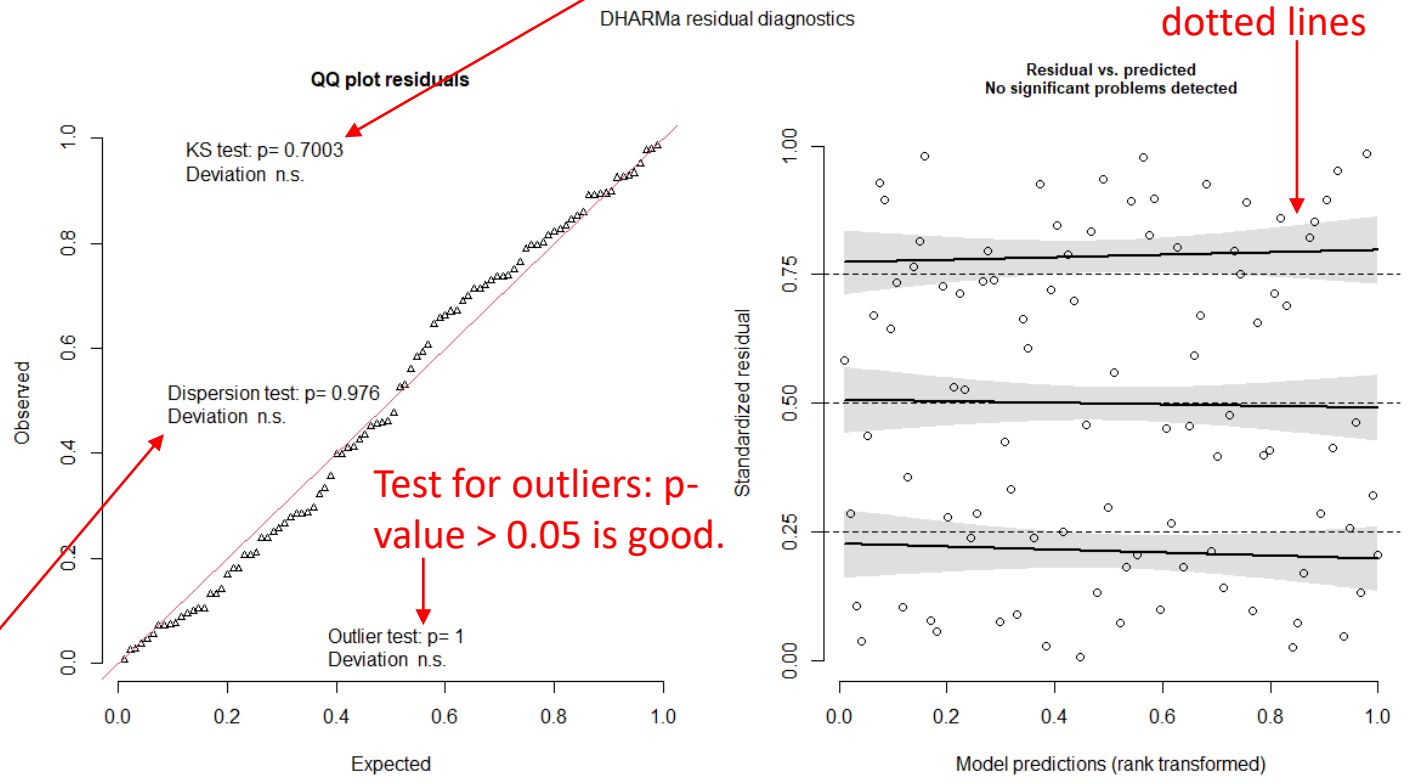
```
#Install package
require(DHARMa) #can take awhile
plot(simulateResiduals(mod1.3))
```

- Problems will be highlighted in red, so no problems here.

- If there are problems, it's not easy to solve: check whether you have left out important variables, whether the relationship is non-linear, etc.

Kolmogorov-Smirnov test: used to test whether the datapoints come from a particular distribution (p-value > 0.05 means yes – good!)

Graphical test for patterns in residuals (i.e. deviance). You want this to follow the black dotted lines



Test for outliers: p-value > 0.05 is good.

Test for overdispersion: p-value > 0.05 is good.

Poisson Example 2: Fitting

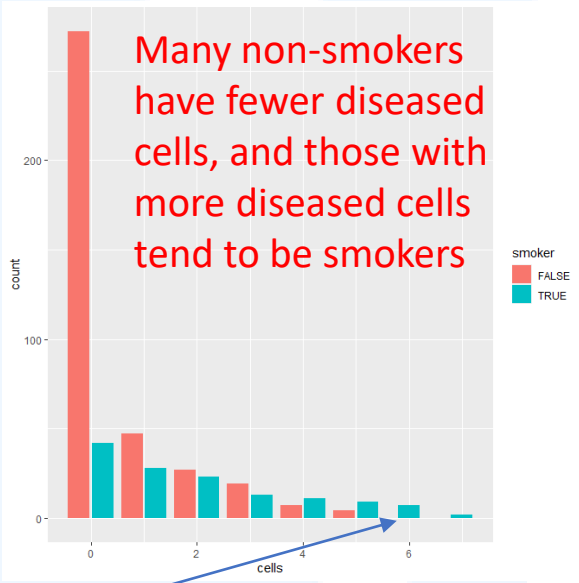
Number of diseased blood cells (count) explained by smoker status (yes/no), age (3 levels), sex (male/female) and body weight (3 levels).

```
#Load dataset
d2=read.table("cells.txt",header=T)
str(d2)
```

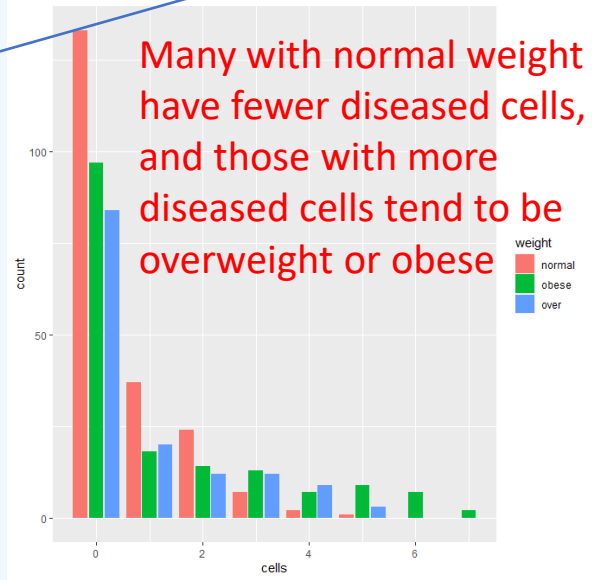
```
> str(d2)
'data.frame':   511 obs. of  5 variables:
 $ cells : int  1 0 1 1 0 2 1 0 5 1 ...
 $ smoker: logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ age   : chr  "young" "young" "young" "young" ...
 $ sex   : chr  "male" "male" "male" "male" ...
 $ weight: chr  "normal" "normal" "normal" "normal" ...
```

```
#Slightly more tricky to visualise
#Barplot of counts (but cannot view interactions)
ggplot(d2,aes(x=cells))+geom_bar(aes(fill=smoker),
position=position_dodge2(preserve="single"))
ggplot(d2,aes(x=cells))+geom_bar(aes(fill=weight),
position=position_dodge2(preserve="single"))
```

This argument positions the blue and pink bars side-by-side instead of stacked on top of each other. If there's no "2", there will be no space between bars



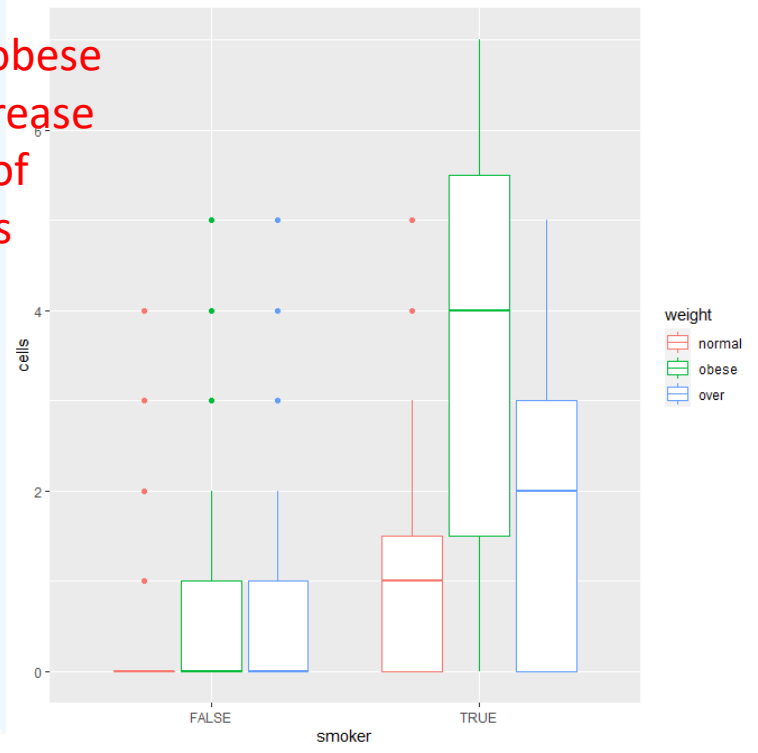
This keeps the blue bar skinny even when there is no pink bar

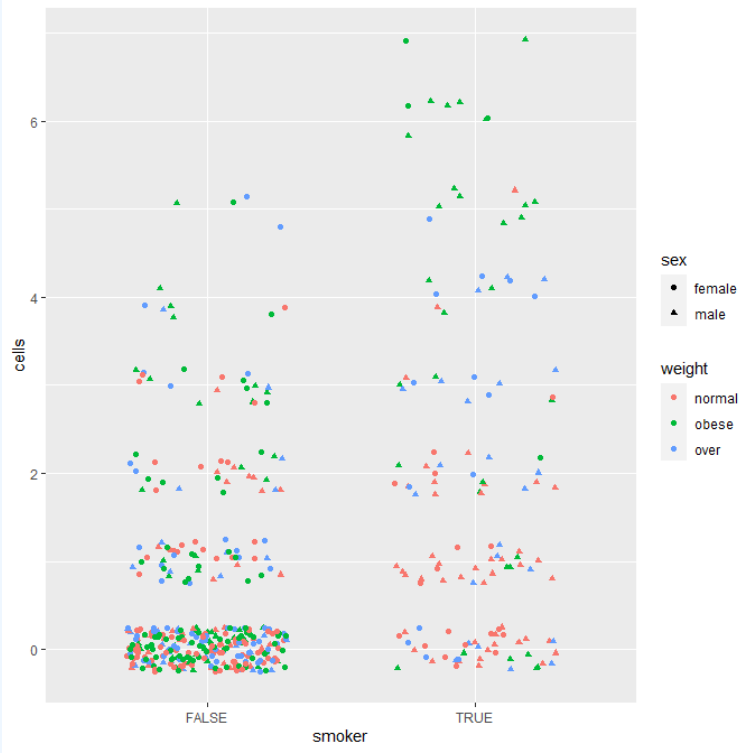


Poisson Example 2: Fitting

```
#Boxplot (not OK for publication, but OK to help you see interactions)
ggplot(d2,aes(x=smoker,y=cells))+geom_boxplot(aes(col=weight))
```

Being both a smoker and obese seems to increase the number of diseased cells





Tend to be many obese individuals near the top, especially for smokers.
No obvious pattern between males and females.

```
#Scatterplot (also just to help you see more interactions)
ggplot(d2,aes(x=smoker,y=cells))+geom_jitter(aes(col=weight),width=0.3)
```

Poisson Example 2: Fitting

```
#Fit Poisson model
mod2.1=glm(cells~smoker*sex*age*weight,family=
poisson,data=d2)
summary(mod2.1)
#Residual deviance 736.33, d.f. = 477:
overdispersion is present

#Fit Quasipoisson
mod2.2=glm(cells~smoker*sex*age*weight,family=
quasipoisson,data=d2)
summary(mod2.2)
```

```
Coefficients: (2 not defined because of singularities)
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                              -0.8329     0.4307  -1.934   0.0537 .
smokerTRUE                               -0.1787     0.8057  -0.222   0.8246
sexmale                                   0.1823     0.5831   0.313   0.7547
ageold                                   -0.1830     0.5233  -0.350   0.7267
ageyoung                                  0.1398     0.6712   0.208   0.8351
weightobese                               1.2384     0.8965   1.381   0.1678
weightover                               -0.5534     1.4284  -0.387   0.6986
smokerTRUE:sexmale                        0.8293     0.9630   0.861   0.3896
smokerTRUE:ageold                        -1.7227     2.4243  -0.711   0.4777
smokerTRUE:ageyoung                       1.1232     1.0584   1.061   0.2892
sexmale:ageold                           -0.2650     0.9445  -0.281   0.7791
sexmale:ageyoung                         -0.2776     0.9879  -0.281   0.7788
smokerTRUE:weightobese                    3.5689     1.9053   1.873   0.0617 .
smokerTRUE:weightover                     2.2581     1.8524   1.219   0.2234
sexmale:weightobese                      -1.1583     1.0493  -1.104   0.2702
sexmale:weightover                        0.7985     1.5256   0.523   0.6009
ageold:weightobese                       -0.9280     0.9687  -0.958   0.3386
ageyoung:weightobese                     -1.2384     1.7098  -0.724   0.4693
ageold:weightover                         1.0013     1.4776   0.678   0.4983
ageyoung:weightover                       0.5534     1.7980   0.308   0.7584
smokerTRUE:sexmale:ageold                 1.8342     2.1827   0.840   0.4011
smokerTRUE:sexmale:ageyoung              -0.8249     1.3558  -0.608   0.5432
smokerTRUE:sexmale:weightobese           -2.2379     1.7788  -1.258   0.2090
smokerTRUE:sexmale:weightover            -2.5033     2.1120  -1.185   0.2365
smokerTRUE:ageold:weightobese             0.8298     3.3269   0.249   0.8031
smokerTRUE:ageyoung:weightobese          -2.2108     1.0865  -2.035   0.0424 *
smokerTRUE:ageold:weightover              1.1275     1.6897   0.667   0.5049
smokerTRUE:ageyoung:weightover           -1.6156     2.2168  -0.729   0.4665
sexmale:ageold:weightobese                2.2210     1.3318   1.668   0.0960 .
sexmale:ageyoung:weightobese              2.5346     1.9488   1.301   0.1940
sexmale:ageold:weightover                -1.0641     1.9650  -0.542   0.5884
sexmale:ageyoung:weightover              -1.1087     2.1234  -0.522   0.6018
smokerTRUE:sexmale:ageold:weightobese    -1.6169     3.0561  -0.529   0.5970
smokerTRUE:sexmale:ageyoung:weightobese      NA         NA      NA       NA
smokerTRUE:sexmale:ageold:weightover         NA         NA      NA       NA
smokerTRUE:sexmale:ageyoung:weightover    2.4160     2.6846   0.900   0.3686
```

ALOT of results! Note there are some NAs because those factor level combinations have no data in them

Poisson Example 2: Simplifying

Notes on step() and stepAIC()
- step() works with Poisson, binomial and negative binomial GLMs. It does NOT work with quasipoisson or quasibinomial.
- Both step() and stepAIC() use AIC; step() is a simplified form of stepAIC() so they should give similar results.
- If step() does not work, use stepAIC() from the MASS package.

```
#Remove most complicated term first: 4-way interaction
mod2.3=update(mod2.2,~.-smoker:sex:age:weight)
#Compare using F-test (recall that AIC will not work for quasipoisson)
anova(mod2.2,mod2.3,test="F")
#No difference, so we prefer the simplified model
```

.

. (simplify manually, unfortunately step() does not work with quasipoisson)

.

```
#Final minimum adequate model
mod2.14=glm(cells~smoker*weight,family=quasipoisson,data=d2)
summary(mod2.14)
```

its hard to interpret the effect size of interaction terms, just show visually.

```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              -0.8712     0.1760  -4.950 1.01e-06 ***
smokerTRUE                0.8224     0.2479   3.318 0.000973 ***
weightobese               0.4993     0.2260   2.209 0.027598 *
weightover                0.2618     0.2522   1.038 0.299723
smokerTRUE:weightobese    0.8063     0.3105   2.597 0.009675 **
smokerTRUE:weightover     0.4935     0.3442   1.434 0.152226
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Poisson Example 2: Visualising results

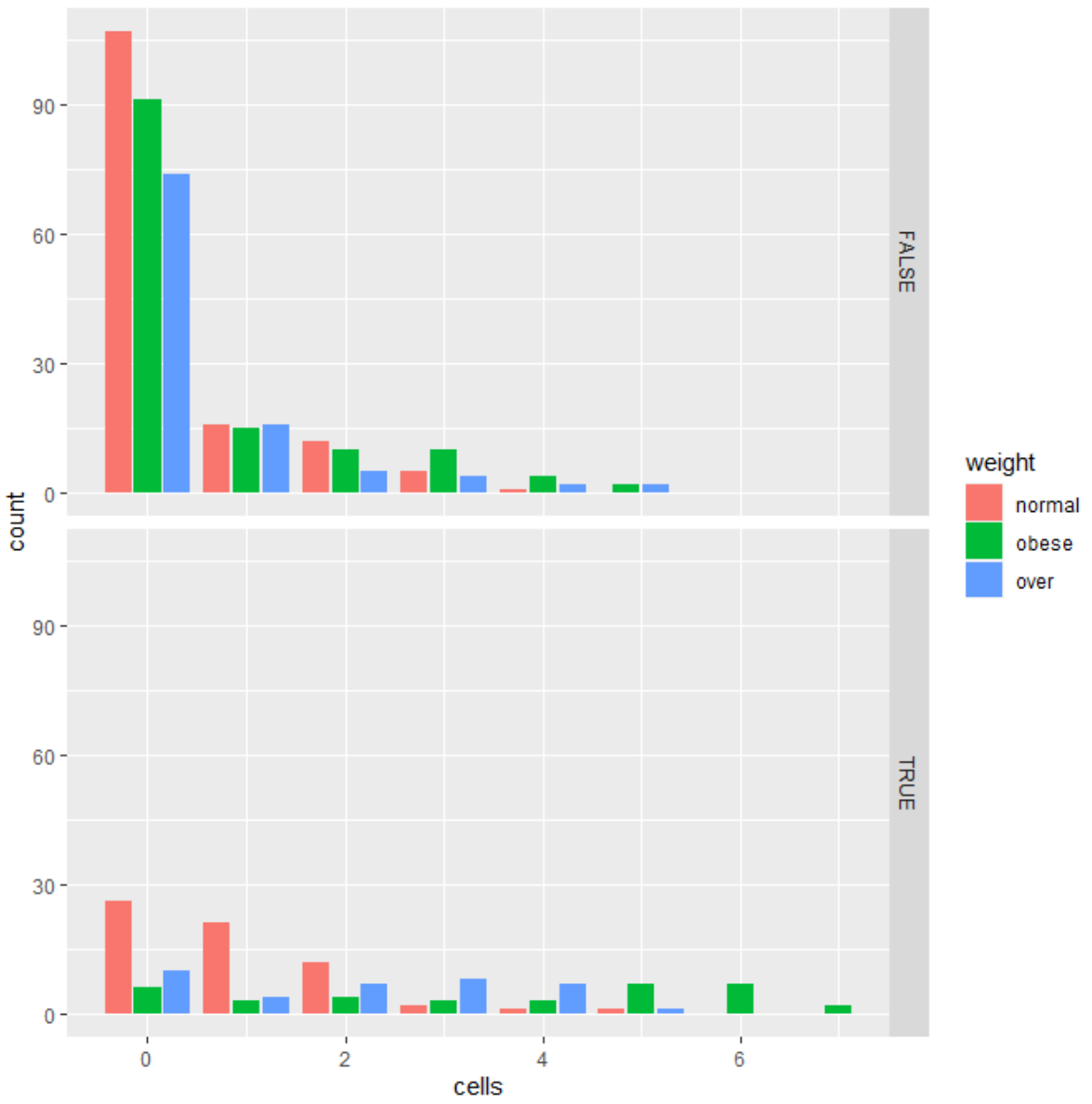
```
#Faceting by <smoker> (True or
False) to see 2-way interaction

ggplot(d2,aes(x=cells))+

geom_bar(aes(fill=weight),position=
position_dodge2(preserve="single"))+

facet_grid(smoker~.)
```

This breaks up the plot into subplots vertically (y-axis) by the levels in <smoker>

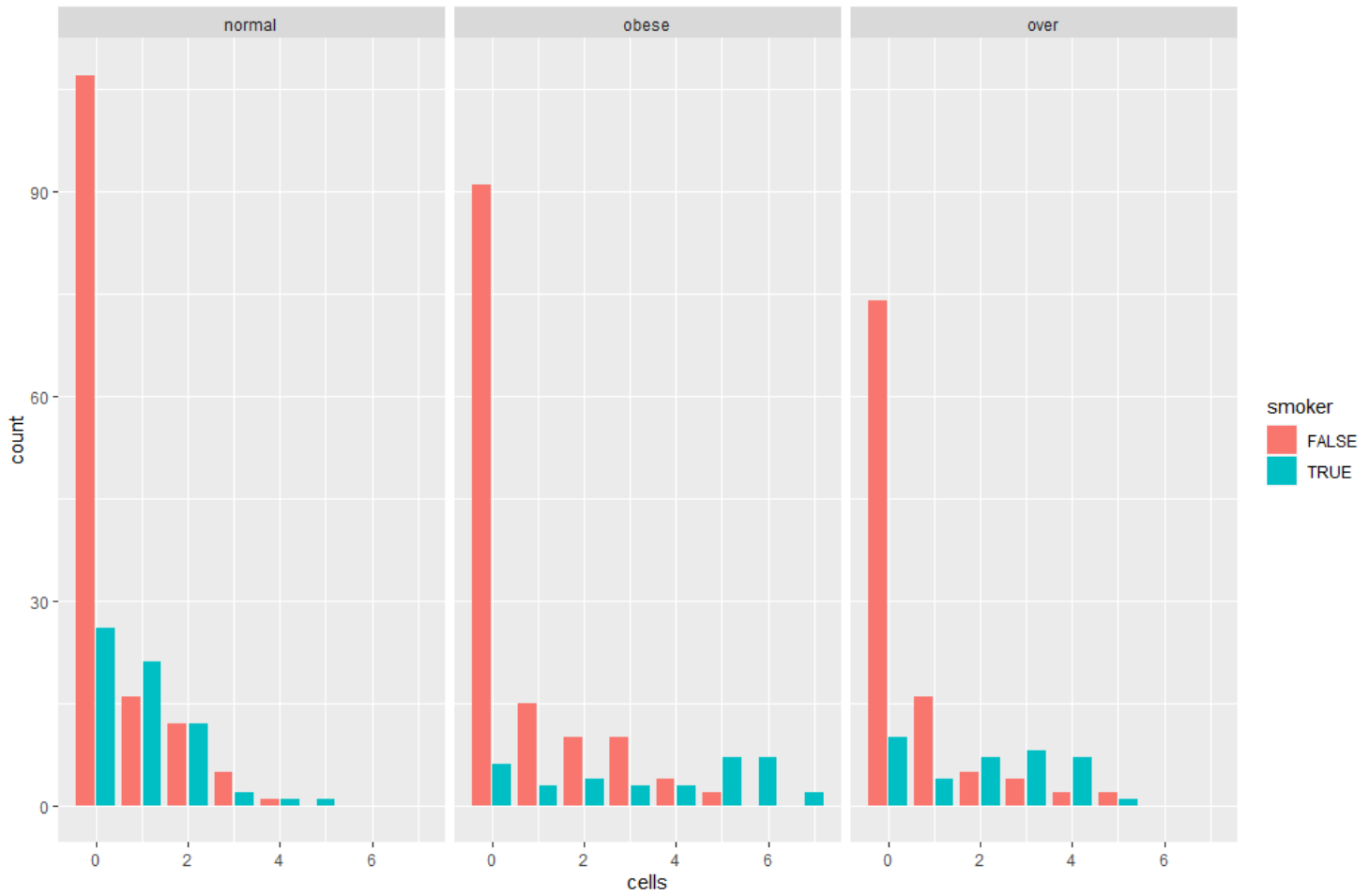


Interpretation: for non-smokers, the difference between people of normal weight and those who are obese/over is not so large; whereas for smokers, this difference is more obvious (those who are normal weight have fewer diseased cells).

Poisson Example 2: Visualising results

```
#Faceting by <weight> (normal, obese or over)
ggplot(d2,aes(x=cells))+
geom_bar(aes(fill=smoker),position=position_dodge2(preserve="single"))+
facet_grid(.~weight)
```

This breaks up the plot into subplots horizontally (x-axis) by the levels in <weight>



Interpretation: for people of normal weight, both smokers and non-smokers have similar numbers of diseased cells; whereas for those who are obese/over, smokers tend to have more diseased cells.

Binomial GLM

Proportion data and Categorical data with 2 categories

When to use?

Situation 1: Your **response variable is a <u>proportion</u>**: i.e. you know the number of "successes" and the number of "failures" (previously with Poisson, you only know the number of "successes").

- Example: Number of coral colonies alive vs. dead, explained by water temperature and pollution levels on a coral reef.

Situation 2: Your **response variable is <u>categorical and can take one of two values</u>** i.e. binary (e.g. A or B, infected or not infected, male or female).

- Example: Successful or failed conservation solution, explained by funding and country.

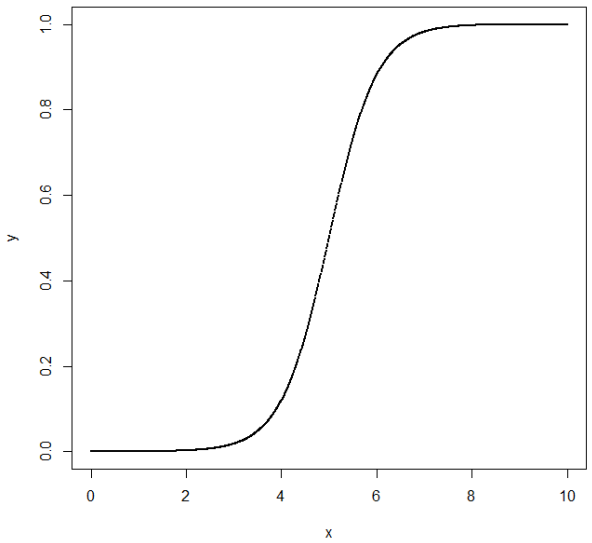
Recall Lecture 3:

- If you simply want to compare a proportion to a constant, use `binom.test()`
- If you simply want to compare two proportions to each other, use `prop.test()`

Situation 1: a binomial GLM on proportion data

Proportion data are strictly bounded above (at 1) and below (at 0). A logistic activation function is perfect because it can be made to asymptote at 0 and 1.

Logistic curve: $y = \frac{1}{1+e^{-2(x-5)}}$



Assuming *n* is sample size, *p* is the proportion of successes and *q* is the proportion of failures (note that *p* + *q* = 1):

- The mean (number of successes) = *np*.

- The variance in the binomial distribution, $s^2 = npq$. s^2 is therefore lowest (i.e. 0) when *p* = 0 or *p* = 1 and maximum when *p* = 0.5.

Note: when *n* is large and *p* is close to 0, the binomial distribution converges with the Poisson distribution (in English: if you have a large dataset and the probability of successes is quite low, you could use a Poisson instead).

Before we fit the model…

For a binomial GLM, **the response "variable" specified is an object with 2 columns**: the first column contains the number of successes, the second contains the number of failures. You have to create this object (e.g. a matrix) yourself before running the GLM.

Similar to Poisson, we need to check for overdispersion (residual dispersion > d.f.). If there is overdispersion, switch to quasibinomial (and use F-tests to simplify).

Small sample sizes (< 30) may be problematic.

The linear predictor is in logits, i.e. the log of the odds: $\ln\left(\frac{p}{q}\right)$. To convert the coefficients (z) back to a probability (p), we use the formula:

$$p = \frac{1}{1 + \frac{1}{e^z}}$$

(Sorry: Math. But this is important for reporting effect sizes.)

Binomial Example: Fitting

Predicting germination success of a parasitic plant based on its genotype <Orobanche> on the host plant <extract> (allowing the two x-variables to interact). The response variables provided are the number of successful germinations <count>, and the total number of trails for each batch <sample>.

```
#Read in the dataset
d3=read.table("germination.txt",header=T)
head(d3)
```

```
> head(d3)
  count sample Orobanche  extract
1    10     39       a75     bean
2    23     62       a75     bean
3    23     81       a75     bean
4    26     51       a75     bean
5    17     39       a75     bean
6     5      6       a75 cucumber
```

Create the “y-variable”: the first column contains the number of successes, which is <count>, and the second column contains the number of failures, which is <sample> minus <count>:

```
y=cbind(d3$count,d3$sample-d3$count)
head(y)
```

```
> head(y)
     [,1] [,2]
[1,]   10   29
[2,]   23   39
[3,]   23   58
[4,]   26   25
[5,]   17   22
[6,]    5    1
```

Binomial Example: Fitting

Fit the binomial GLM

```
mod3.1=glm(y~Orobanche*extract,family=
"binomial",data=d3)

summary(mod3.1)
```

```
Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -0.4122     0.1842  -2.238   0.0252 *
Orobanchea75                   -0.1459     0.2232  -0.654   0.5132
extractcucumber                 0.5401     0.2498   2.162   0.0306 *
Orobanchea75:extractcucumber    0.7781     0.3064   2.539   0.0111 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 33.278  on 17  degrees of freedom
AIC: 117.87
```

Looks like there's overdispersion, so we switch to quasibinomial:

```
mod3.2=glm(y~Orobanche*extract,family=
"quasibinomial",data=d3)

summary(mod3.2)#The interaction is no
longer significant
```

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -0.4122     0.2513  -1.640   0.1193
Orobanchea75                   -0.1459     0.3045  -0.479   0.6379
extractcucumber                 0.5401     0.3409   1.584   0.1315
Orobanchea75:extractcucumber    0.7781     0.4181   1.861   0.0801 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.861832)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 33.278  on 17  degrees of freedom
AIC: NA
```

No need to check assumptions because it is quasibinomial

Remember this is now OK because we have accounted for it by using quasi-likelihood

Binomial Example: Simplifying

```
#Remove interaction
mod3.3=update(mod3.2,~.-Orobanche:extract)
anova(mod3.2,mod3.3,test="F") #p-value=0.081 so we use the simpler model
summary(mod3.3)
```

Remember we now have to use F-test because it is quasi-likelihood

```
#Remove <Orobanche> (p-value=0.25)
mod3.4=update(mod3.3,~.-Orobanche)
anova(mod3.3,mod3.4,test="F") #p-value=0.25
summary(mod3.4)
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.5122     0.1531  -3.345   0.0034 **
extractcucumber  1.0574     0.2118   4.992 8.09e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 2.169821)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 42.751  on 19  degrees of freedom
AIC: NA
```

Final model: only <extract> is significant, i.e. there is a significant difference between germination rates for bean vs. cucumber extracts, but not for different genotypes

Binomial Example: Interpreting coefficients to extract effect size

We are trying to calculate the rate of germination for each of the 2 different levels of <extract>, bean and cucumber. The intercept in this case represents bean.

Recall:

$$p = \frac{1}{1 + \frac{1}{e^z}}$$

This is the *z* for beans

This is the DIFFERENCE between the *z* for beans and the *z* for cucumber

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.5122     0.1531  -3.345   0.0034 **
extractcucumber  1.0574     0.2118   4.992 8.09e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 2.169821)

    Null deviance: 98.719  on 20  degrees of freedom
Residual deviance: 42.751  on 19  degrees of freedom
AIC: NA
```

```
#Calculate germination rate for bean
1/(1+1/exp(-0.5122)) #0.3746
#Germination rate (i.e. p of success) in bean is 37.5%
```

```
#Calculate germination rate for cucumber
1/(1+1/exp(-0.5122+1.0574)) #0.6330
#Germination rate in cucumber is 63.3%
```

Note: it is also possible to use tapply() and predict() with type="response" to convert and extract the values:

tapply(predict(mod3.4,type="response"),d3$extract,mean)

```
> tapply(predict(mod3.4,type="response"),d3$extract,mean)
     bean  cucumber
0.3746835 0.6330275
```

Situation 2: a binomial GLM on categorical (binary) data

Exactly the same, except the response variable specified is a categorical variable (from the dataset) with two unique values in it

- Examples:. A and B; T and F; 1 and 2
- Best to make sure it is either a factor or chr type

Multinomial GLM

Categorical data with >2 categories

When to use?

Similar to binomial when your response variable is categorical, but here it can take >2 values.

A multinomial GLM tells you whether one combination of all the categories is different from another combination of all the categories

this distribution is significantly different from another distribution

- It only tells you whether there is an overall difference (similar to an ANOVA).

- You will then have to compare each pair of categories using a binomial GLM to see what is driving the difference (similar to pairwise t-tests).

Example:

Predict **IUCN category** of species (6 categories, from Least Concern to Extinct) based on their biogeographic <u>home region</u> and <u>body size</u>.

Multinomial Example: Fitting

Categorical variable <smoker>: shows the difference between smokers and non-smokers for obese vs. normal

Continuous variable <cells>: shows slope for obese vs. normal

Back to the <cells> dataset! Can we explain a person's **weight** by their sex, whether they smoke, how many diseased cells they have?

```
d2=read.table("cells.txt",header=T)
d2$weight=as.factor(d2$weight)
str(d2)
```

```
> str(d2)
'data.frame':   511 obs. of  6 variables:
 $ cells  : int  1 0 1 1 0 2 1 0 5 1 ...
 $ smoker : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ age    : chr  "young" "young" "young" "young" ...
 $ sex    : chr  "male" "male" "male" "male" ...
 $ weight : Factor w/ 3 levels "normal","obese",..: 1 1 1 1 1 1 1 1 1 1
```

<weight> has 3 levels, so do multinomial GLM:

```
require(nnet)
mod4.1=multinom(weight~smoker+cells+sex,data=d2)
summary(mod4.1) #only effect sizes, no p-values!
```

```
> summary(mod4.1) #no p-values!
Call:
multinom(formula = weight ~ smoker + cells + sex, data = d2)

Coefficients:
      (Intercept) smokerTRUE     cells    sexmale
obese  -0.4085892 -1.3727814 0.5184812 0.21301527
over   -0.4031678 -0.5422944 0.2989217 -0.09600557

Std. Errors:
      (Intercept) smokerTRUE      cells   sexmale
obese   0.1541833  0.3100031 0.09255907 0.2331743
over    0.1545588  0.2914103 0.09624176 0.2414315

Residual Deviance: 1066.412
AIC: 1082.412
```

These are the effect size and uncertainty for the same coefficient

As effect sizes, you can report the actual percentages in the data instead of these figures

Get overall p-value for each variable:

```
require(car)
Anova(mod4.1)
```

report p-value first
to show the effect size, report the proportion of categories when smoker is true then

```
> Anova(mod4.1)
Analysis of Deviance Table (Type II tests)

Response: weight
       LR Chisq Df Pr(>Chisq)
smoker   21.710  2   1.93e-05 ***
cells    37.258  2   8.12e-09 ***
sex       1.659  2     0.4362
```

All significant except <sex>

Note: diagnostic tests are not (yet) possible for multinomial GLM.

Multinomial Example: Comparing all levels

Compare all other levels to the reference level (by default, the first level alphabetically) using Wald tests to get p-values

```
z1=summary(mod4.1)$coefficients/summary(mod4.1)$standard.errors
p1=(1-pnorm(abs(z1),0,1))*2
```

Compares everything to "normal"

```
> p1
       (Intercept)   smokerTRUE        cells   sexmale
obese 0.008048657 9.498618e-06 2.123517e-08 0.3609558
over  0.009093785 6.275396e-02 1.896715e-03 0.6908872
```

To change to a different reference level, we create a "new" y-variable to run a new model

```
d2$weight2=relevel(d2$weight,ref="obese")
```

Create the new variable <weight2> and fit another GLM

```
mod4.2=multinom(weight2~smoker+cells+sex,data=d2)
summary(mod4.2)
z2=summary(mod4.2)$coefficients/summary(mod4.2)$standard.errors
p2=(1-pnorm(abs(z2),0,1))*2
```

```
> p2
       (Intercept)   smokerTRUE        cells   sexmale
normal 0.008048773 9.500499e-06 2.123392e-08 0.3609392
over   0.973845111 1.160963e-02 9.049725e-03 0.2142376
```

```
> str(d2)
'data.frame':   511 obs. of  6 variables:
 $ cells  : int  1 0 1 1 0 2 1 0 5 1 ...
 $ smoker : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ age    : chr  "young" "young" "young" "young" ...
 $ sex    : chr  "male" "male" "male" "male" ...
 $ weight : Factor w/ 3 levels "normal","obese",..: 1 1 1 1 1 1 1 1 1 1
 $ weight2: Factor w/ 3 levels "obese","normal",..: 2 2 2 2 2 2 2 2 2 2
```

<weight> lists "normal" first, <weight2> lists "obese" first

Note: releveling can also be used to compare different levels in lm() models.

Quasi GLM

Non-normal continuous data

When to use?

You have a **continuous response variable** (ideally not a count, proportion or time to event) that is **not normally distributed**. You don't want to/cannot transform your y-variable or use a nonparametric test, or these do not work.

"Quasi" models do not assume any error distribution at all (similar to the quasipoisson and quasibinomial), and use a "dispersion parameter" to try to account for the dispersion in the model.

Example: you want to explain **height** of a group of people using nutritional status (continuous) and sex (categorical), but when you fit the ANCOVA, the shapiro.test() shows that the data are not normally distributed. You try square root followed by log (and all other transforms) but it does not work.

WARNING: treat quasi GLMs as your last resort. They are relatively new and still not widely implemented.

Quasi Example: Fitting

Back to the <cells> dataset again!

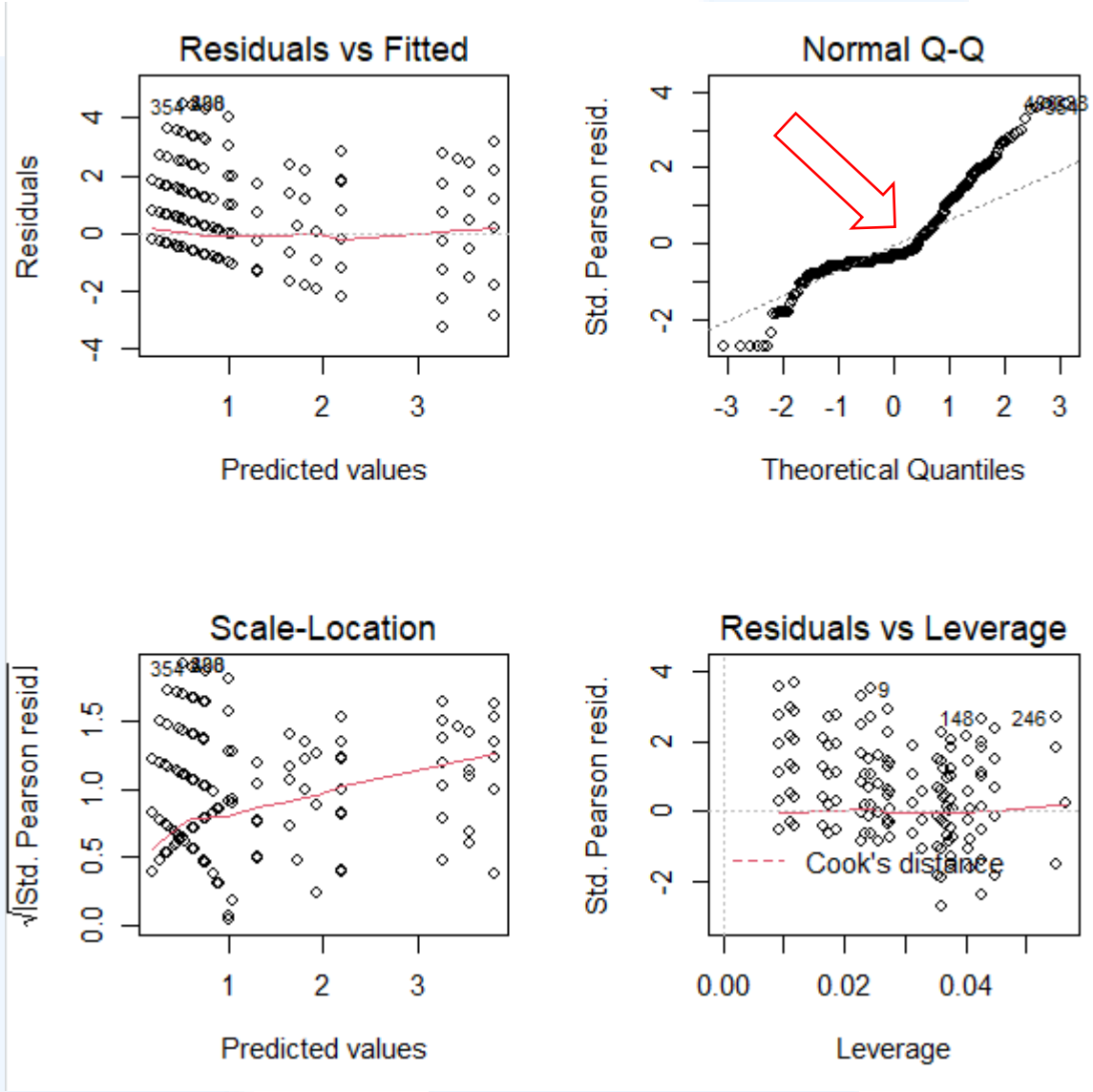
```
#Read in dataset
d2=read.table("cells.txt",header=T)
```

#Fit GLM with Gaussian distribution—this is exactly the same as running an lm() (recall that we used lm() for regression, ANOVA and ANCOVA):

```
mod5.1=glm(cells~sex/age+smoker*weight,
family="gaussian",data=d2)
```

Check assumptions:

```
par(mfrow=c(2,2))
plot(mod5.1) #clearly non-normal errors
```



Quasi Example: Fitting

Fit quasi model:

```
mod5.2=update(mod5.1,family="quasi")
summary(mod5.2)
```

```
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.27509    0.20223   1.360  0.17436
sexmale                 0.20177    0.23498   0.859  0.39094
smokerTRUE              0.55905    0.19690   2.839  0.00471 **
weightobese             0.26052    0.15435   1.688  0.09207 .
weightover              0.14933    0.16247   0.919  0.35848
sexfemale:ageold        0.07806    0.22268   0.351  0.72608
sexmale:ageold          0.26479    0.20331   1.302  0.19339
sexfemale:ageyoung      0.45928    0.27939   1.644  0.10084
sexmale:ageyoung       -0.29126    0.20112  -1.448  0.14818
smokerTRUE:weightobese  2.26286    0.29925   7.562 1.92e-13 ***
smokerTRUE:weightover   0.74502    0.30885   2.412  0.01621 *
```

Then proceed with manual simplification until you arrive at your minimum adequate model :

- In this case I would remove Answer first

Note: because this is also quasi-likelihood, there are no assumptions to check.

Summary (Learning Objectives)

Generalised Linear Models (GLM)

- Theory: Link functions, linear predictors and error distributions
    - Error distributions and variable types
    - Least Squares vs. Maximum Likelihood
- Poisson for count data
    - Quasipoisson/Negative Binomial for overdispersion
    - Simplifying, comparing, checking and interpreting models
- Binomial for proportion/categorical data (2 categories)
    - Quasibinomial for overdispersion
    - Simplifying, comparing, checking and interpreting models
- Multinomial for categorical data (>2 categories)
- Quasi as a last resort for non-normally distributed continuous data