Advanced Statistical Modelling Coursework 2

087074

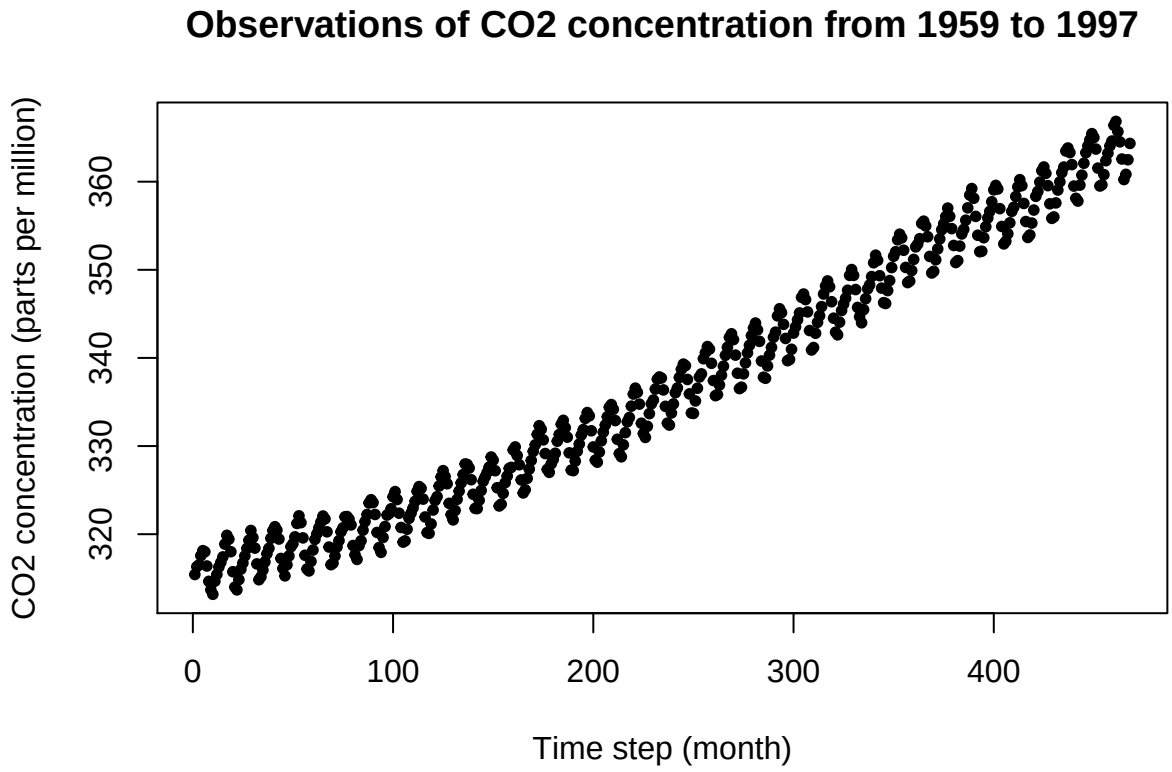
20/03/2021

Question 2

2a

A plot of $CO2$ with respect to $timeStep$ shows that CO2 concentration increases from the year 1959 to 1997. The CO2 concentration also changes in a cyclical pattern throughout each year, suggesting that there is a seasonal effect.

```
# plot CO2 with respect to time step
plot(carbonD$timeStep,carbonD$co2,pch=20,lwd=1,
     xlab="Time step (month)", ylab="CO2 concentration (parts per million)",
     main ="Observations of CO2 concentration from 1959 to 1997")
```



CO2 concentration is a continuous variable and the graph shows that the variance of CO2 concentration remains constant as time increases. Therefore, a Normal Generalised Additive Model (GAM) is a suitable model for the dataset. The model has `CO2` as the response variable and `timeStep` and `month` as covariates to account for the yearly and seasonal changes in CO2 concentration.

$$Y_i \sim N(\mu(x_i), \sigma^2) \quad (Y_i \text{ independent})$$
$$\mu(x_i) = \beta_0 + f_1(\texttt{timeStep}_i) + f_2(\texttt{month}_i)$$

2b

This GAM, `Amodel3`, is fitted to the data with enough knots k to ensure that the smooth functions have enough flexibility.

```
Amodel3 <- gam(co2~s(timeStep,k=33,bs="cs") + s(month,k=11,bs="cc"),data=carbonD,
               family=gaussian) # fit the model
```

A χ^2 goodness of fit test is performed to see if `Amodel3` fits the data well compared with the saturated model.

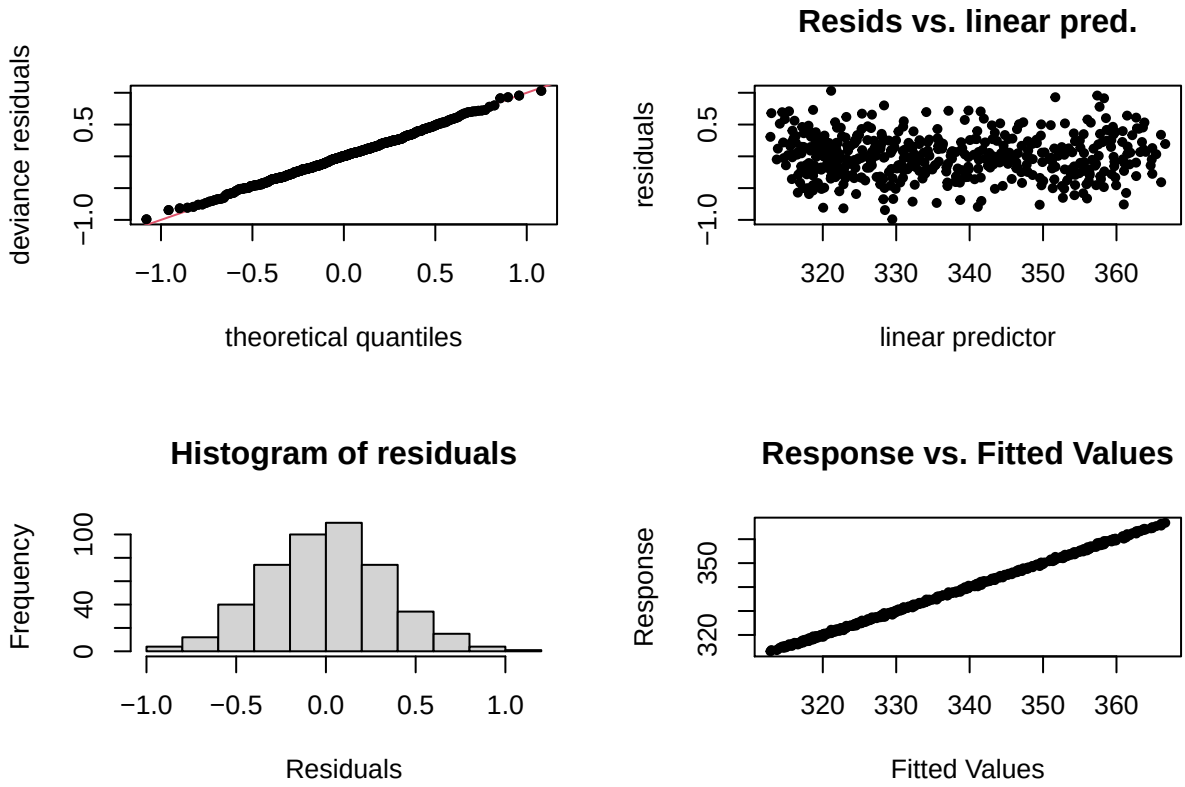
```
sc.deviance <- Amodel3$deviance/Amodel3$sig2
1-pchisq(sc.deviance,Amodel3$df.residual) # goodness of fit test
```

```
## [1] 0.490927
```

The p-value of the test is larger than 0.05, suggesting that `Amodel3` fits the data well.

The residuals and effective degrees of freedom of `Amodel3` are displayed using `gam.check`.

```
par(mfrow=c(2,2))
gam.check(Amodel3,pch=20)
```




```
##
## Method: GCV   Optimizer: magic
```

```
## Smoothing parameter selection converged after 7 iterations.
## The RMS GCV score gradient at convergence was 1.176689e-06 .
## The Hessian was positive definite.
## Model rank =  42 / 42
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                k'   edf k-index p-value
## s(timeStep) 32.00 29.52    0.82  <2e-16 ***
## s(month)     9.00  7.82    0.61  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both of the smooth functions do not use all of their degrees of freedom, $\texttt{edf} < \texttt{k'} - 1$, thus there is enough flexibility in the smooth functions.

The QQ plot of the residuals shows that the errors of the model are normally distributed around the mean and there is no structure to the scatter plot of the residuals, with the residuals randomly scattered around the zero line. This strongly suggests that the model fits the data well.

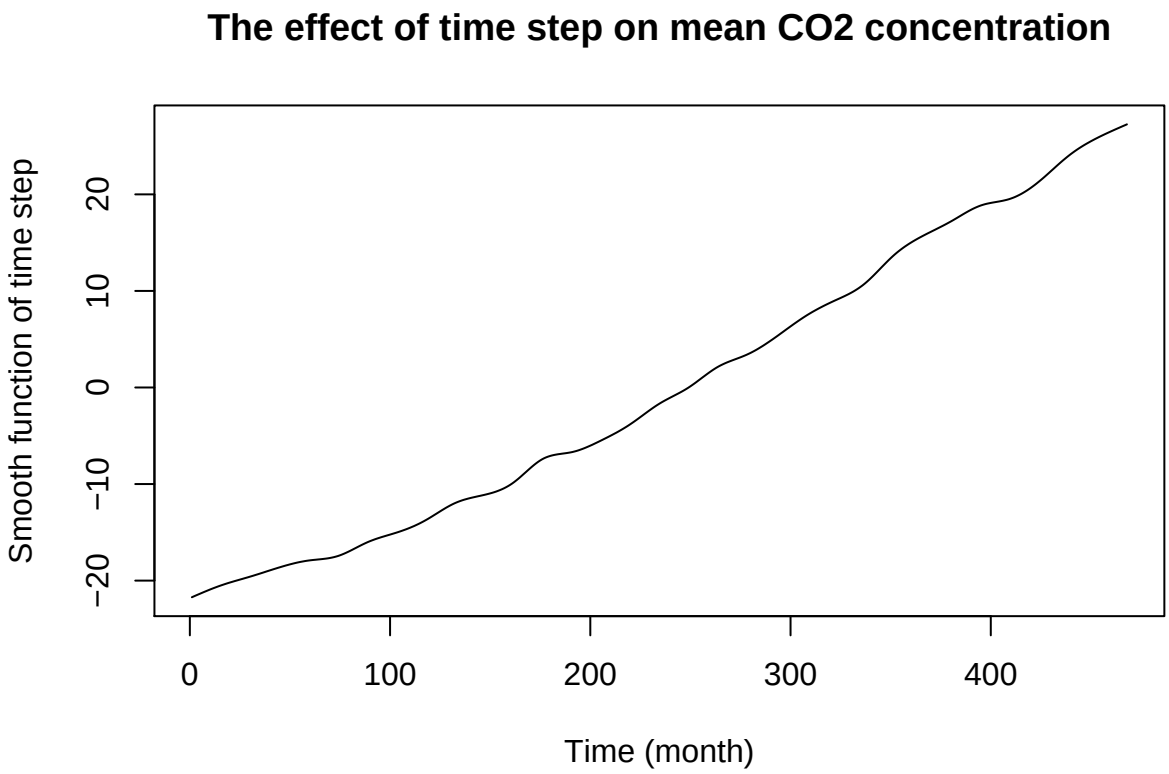
2c

The `predict` function with `type="terms"` gives the predicted values of the smooth functions, $f_1(\texttt{timeStep}_i)$ and $f_2(\texttt{month}_i)$.

```
test.data = data.frame(timeStep = carbonD$timeStep,
                       month=c(1,2,3,4,5,6,7,8,9,10,11,12)) # create a dataset
preds <- predict(Amodel3, type = 'terms') # predict the dataset
```

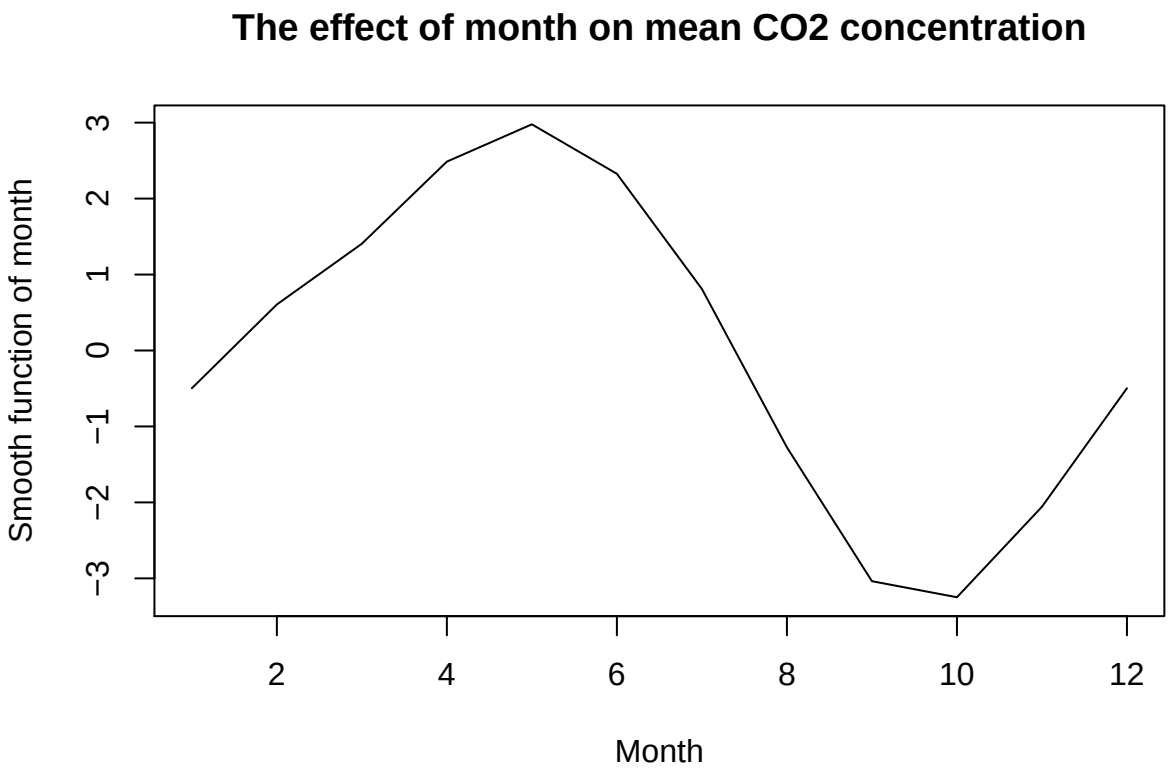
A plot of the predicted values of the smooth function $f_1(\texttt{timeStep}_i)$ shows that CO2 concentration increased linearly with respect to `timeStep`.

```
plot(carbonD$timeStep,preds[,1], type="l",
     xlab="Time (month)",ylab="Smooth function of time step",
     main="The effect of time step on mean CO2 concentration")
```



A plot of the predicted values of the smooth function $f_2(\texttt{month}_i)$ shows that CO2 concentration oscillates throughout the year.

```
plot(1:12,preds[1:12,2], type="l",
     xlab="Month",ylab="Smooth function of month",
     main="The effect of month on mean CO2 concentration")
```



2d

The GAM, `Amodel3`, can be used to predict CO2 concentrations for out-of-sample data, such as the year 1998. I generate model predictions for the years 1985 – 1988 and 95% confidence intervals for the 1998 predictions.

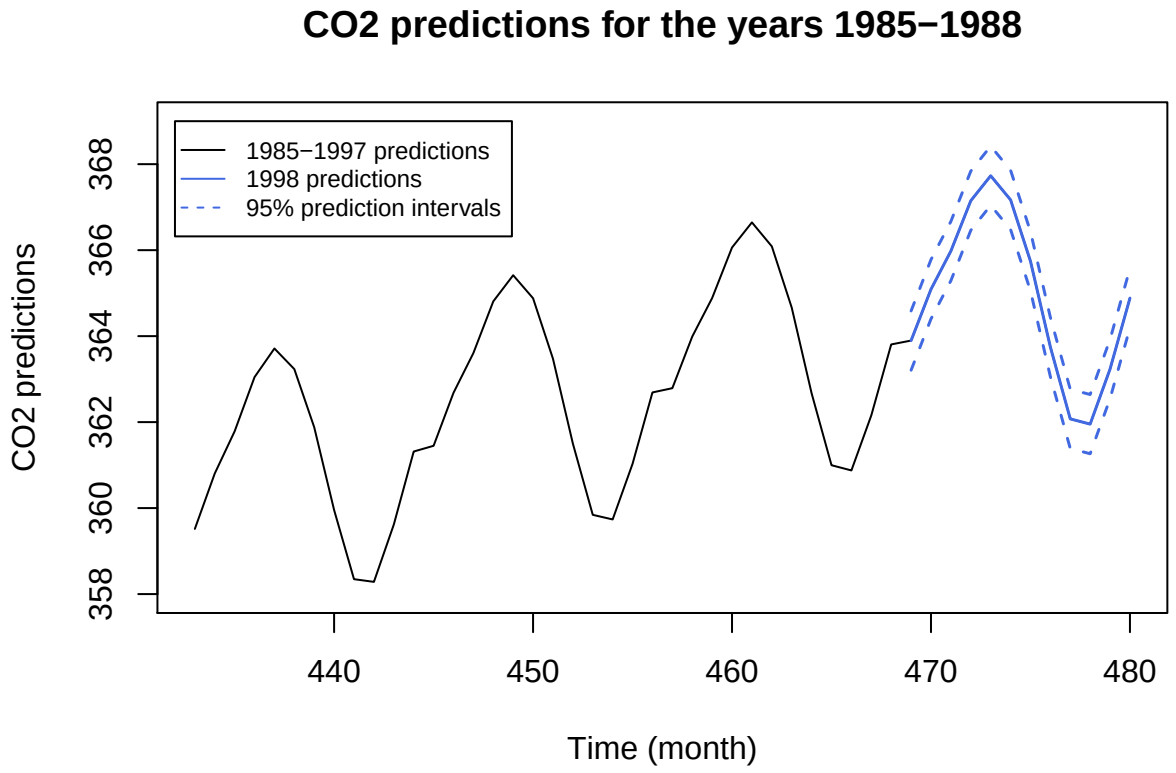
```
timeStep_before <- c(433:480) # includes years 1985-1988
data_before = data.frame(timeStep = timeStep_before,
                         month=c(1,2,3,4,5,6,7,8,9,10,11,12)) # create a dataframe
preds_before <- predict(Amodel3,newdata=data_before,se.fit=T) # predict

timeStep_1998 <- c(469:480) # the timeSteps that represent the year 1998
data_1998 = data.frame(timeStep = timeStep_1998,
                       month=c(1,2,3,4,5,6,7,8,9,10,11,12))  # create a dataframe
preds_1998 <- predict(Amodel3,newdata=data_1998,se.fit=T) # predict

sig2 <- Amodel3$sig2 # obtain variance of the model
sig <- sqrt(sig2)
upper <- qnorm(0.975,preds_1998$fit,sig) # use variance to caclculate
lower <- qnorm(0.025,preds_1998$fit,sig) # upper and lower 95% bounds
```

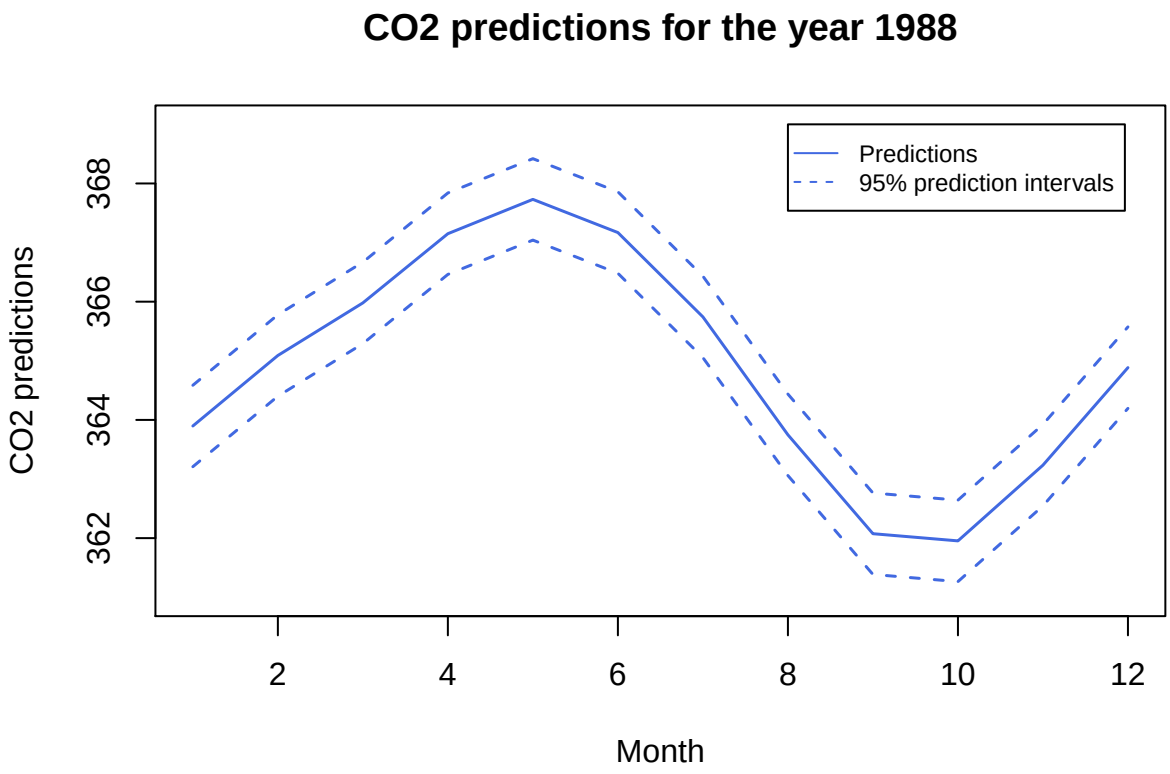
A plot showing the CO2 concentration predictions for the year 1988 following on from the within-sample predictions (1985 – 1987) shows that the model predicts a future CO2 concentration increase for the year 1988.

```r
plot(timeStep_before, preds_before$fit, type="l", ylim=c(358,369),
     xlab="Time (month)", ylab = "CO2 predictions",
     main = "CO2 predictions for the years 1985-1988") # 1995-1997
legend(x=432,y=369, legend=c("1985-1997 predictions", "1998 predictions",
                             "95% prediction intervals"), pch=c(-1,-1,-1),
       lty=c(1,1,2), col=c('black','royalblue', 'royalblue'),
       lwd=c(1,1,1), cex = 0.75) # add a legend
lines(timeStep_1998, preds_1998$fit,col="royalblue",lwd=1.5,lty=1) # 1988
lines(c(469:480),upper,col="royalblue",lwd=1.5,lty=2) # 95% upper confidence bound
lines(c(469:480),lower,col="royalblue",lwd=1.5,lty=2) # 95% lower confidence bound
```



A plot only containing the predictions of the year 1998, along with it's 95 confidence intervals shows that the model predicts a continuation of the seasonal changes in CO2 concentration.

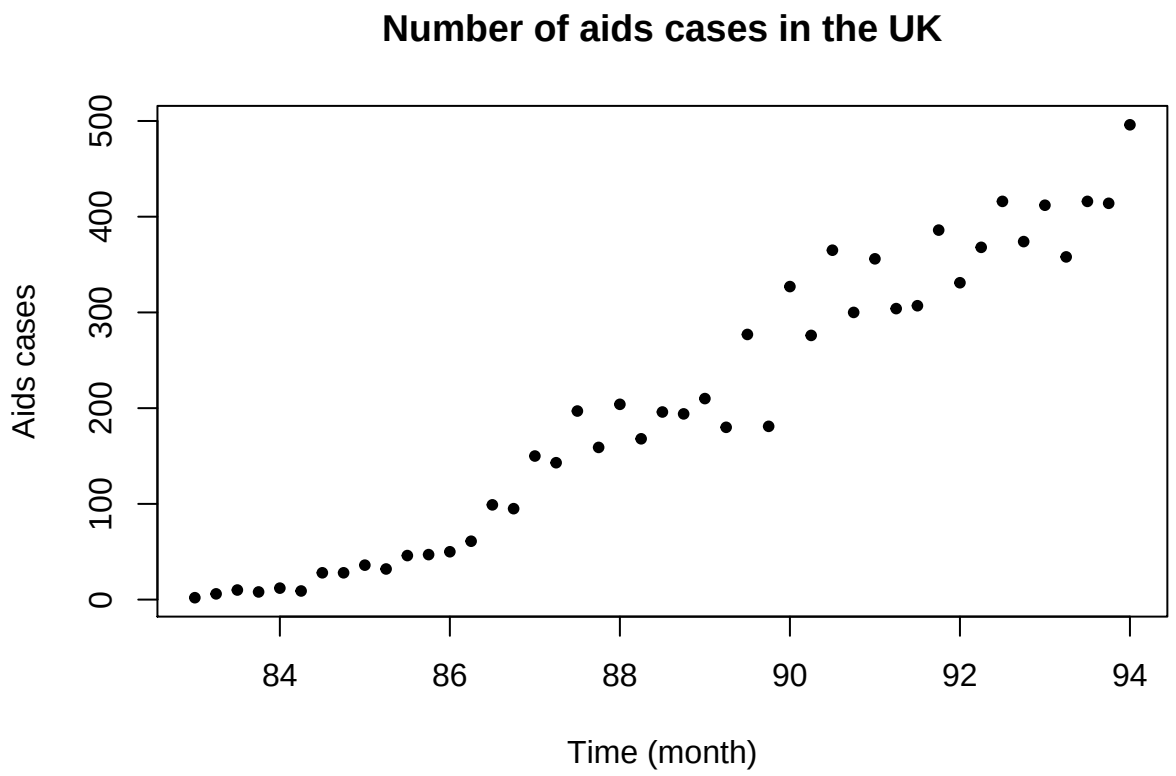
```r
plot(1:12, preds_1998$fit,type="l",col="royalblue",lwd=1.5,lty=1,
     ylim=c(361,369), xlab="Month", ylab = "CO2 predictions",
     main = "CO2 predictions for the year 1988") # 1988
legend(x=8,y=369, legend=c("Predictions", "95% prediction intervals"),
       pch=c(-1,-1), lty=c(1,2), col=c('royalblue', 'royalblue'),
       lwd=c(1,1), cex = 0.75)
lines(1:12,upper,col="royalblue",lwd=1.5,lty=2) # 95% upper confidence bound
lines(1:12,lower,col="royalblue",lwd=1.5,lty=2) # 95% lower confidence bound
```


CO2 predictions for the year 1988
Predictions
95% prediction intervals
CO2 predictions
362
364
366
368
2
4
6
8
10
12
Month

Question 3

A plot of `cases` with respect to `date` shows that the mean and variance of aids cases increases over time.

```
plot(aids$date,aids$cases,pch=20,lwd=1,xlab="Time (month)",ylab="Aids cases",
     main="Number of aids cases in the UK")
```



Furthermore, the response variable (number of aids cases) is count data. Therefore, a Poisson GAM is a suitable model for the dataset.

```
# fit a Poisson model to the data
aidsmodel2 <- gam(cases~s(date,k=17,bs="cs"),data=aids,
                  family=poisson(link="identity"))
```

A χ^2 goodness of fit test is performed to see if `aidsmodel2` fits the data as well as the saturated model.

```
sc.deviance <- aidsmodel2$deviance/aidsmodel2$sig2
1-pchisq(sc.deviance,aidsmodel2$df.residual) # goodness of fit test
```

```
## [1] 4.305555e-10
```

The p-value of the χ^2 goodness of fit test is larger than 0.05, suggesting that the Poisson GAM does not fit the data well. This is most likely due to the model not being able to capture the variance in the data.

The Negative Binomial distribution is similar to the Poisson distribution, but with a dispersion parameter that is free to vary. Therefore, I fit a Negative Binomial GAM, `aidsmodel3` to the data.

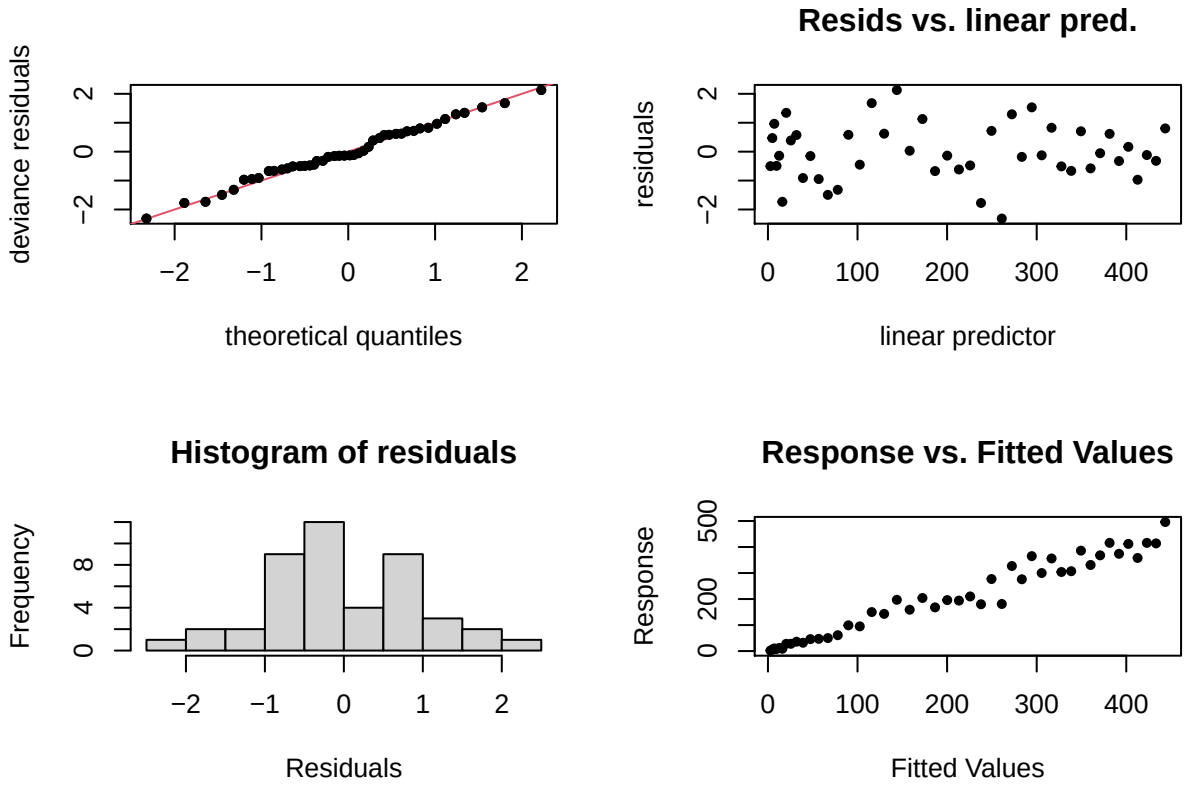
```r
# fit a Negative Binomial model to the data
aidsmodel3 <- gam(cases~s(date,k=7,bs="cs"),data=aids,
                  family=nb(theta = NULL, link = "identity"))
sc.deviance <- aidsmodel3$deviance/aidsmodel3$sig2
1-pchisq(sc.deviance,aidsmodel3$df.residual) # goodness of fit test
```

```
## [1] 0.4273337
```

The p-value of the test is larger than 0.05, suggesting that the `aidsmodel3` fits the data well compared to the saturated model.

The residuals and effective degrees of freedom of `aidsmodel3` are displayed using `gam.check`.

```r
par(mfrow=c(2,2))
gam.check(aidsmodel3,pch=20)
```




```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [1.963913e-06,2.153398e-05]
## (score 207.4925 & scale 1).
## Hessian positive definite, eigenvalue range [2.057985,9.550735].
## Model rank =  7 / 7
##
```

```
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'  edf k-index p-value
## s(date) 6.00 4.15    1.15    0.77
```

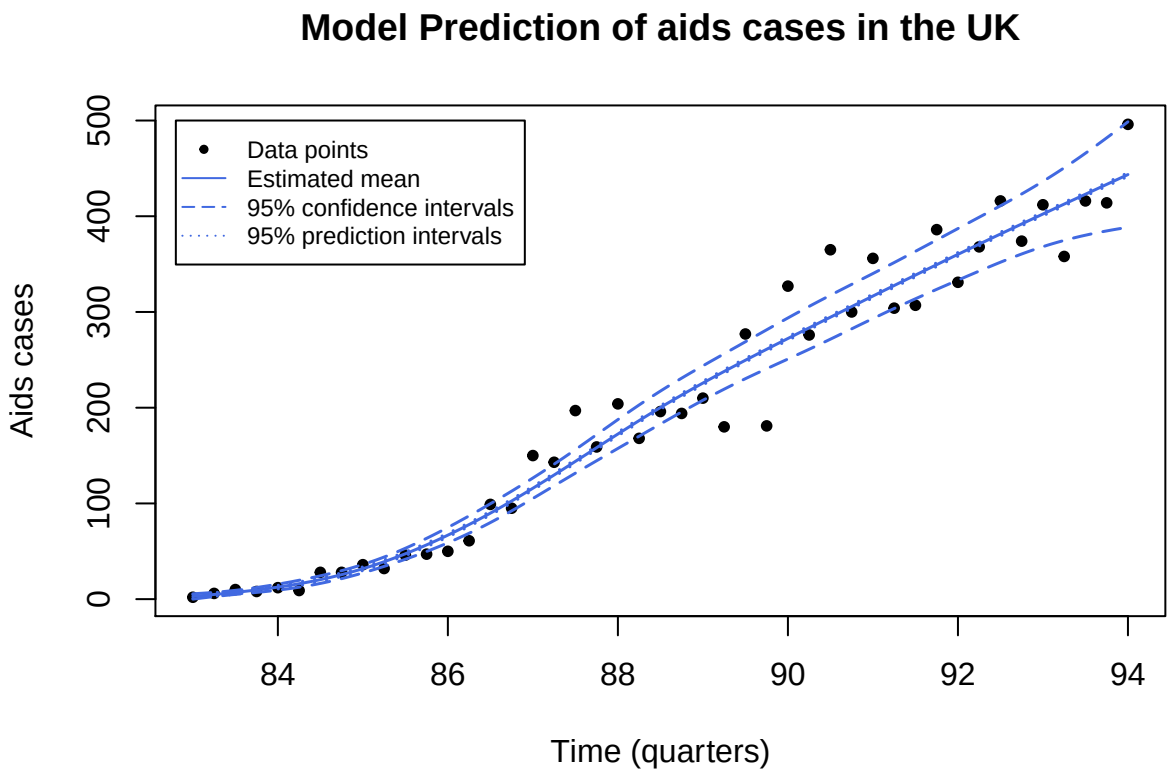
The smooth function does not use all of its degrees of freedom, `edf` $<$ `k'` -1, thus there is enough flexibility in the smooth function.

The QQ plot of the residuals shows that the errors of the model are normally distributed around the mean and there is no structure to the scatter plot of the residuals, with the residuals randomly scattered around the zero line. This strongly suggests that the model fits the data well.

3b

This produces a graph of the estimated mean of `aidsmodel3` as well as 95% confidence and prediction intervals.

```
# predictions for Negative Binomial
xx <- seq(min(aids$date),max(aids$date),length=200)
preds <- predict(aidsmodel3,newdata=data.frame(date=xx),se.fit=T)
# confidence intervals of the mean
plot(aids$date,aids$cases,pch=20,xlab="Time (quarters)",ylab="Aids cases",
     main="Model Prediction of aids cases in the UK")
lines(xx,preds$fit,col="royalblue",lwd=1.5)
lines(xx,preds$fit+1.96*preds$se.fit,col="royalblue",lwd=1.5,lty=5)
lines(xx,preds$fit-1.96*preds$se.fit,col="royalblue",lwd=1.5,lty=5)
## prediction intervals using plug it prediction
sig2 <- aidsmodel3$sig2
sig <- sqrt(sig2)
upper <- qnorm(0.975,preds$fit,sig)
lower <- qnorm(0.025,preds$fit,sig)
lines(xx,lower,col="royalblue",lwd=1.5,lty=3)
lines(xx,upper,col="royalblue",lwd=1.5,lty=3)
# add a legend
legend(x=82.8,y=500, legend=c("Data points","Estimated mean",
                              "95% confidence intervals",
                              "95% prediction intervals"),
       pch=c(20,-1,-1,-1), lty=c(-1,1,5,3), col=c('black','royalblue','royalblue',
                                                  'royalblue'), lwd=c(-1,1,1,1),
       cex = 0.75)
```



The graph shows that the mean, along with the 95% confidence intervals, closely follows the trend of the data. The majority of the data falls within the 95% prediction intervals, however there are a few too many data points outside the interval. Overall, the graph suggests that the model fits the data well.

Question 6

6a

The response variable `yeild`, $Y_{i,j}$, in the dataset `penicillin` can be modeled by a Normal Generelised Linear Mixed Model (GLMM) with `treat` as a fixed effect and `blend` as a random effect.

$$\begin{aligned} Y_{i,j}|\gamma_j &\sim N(\mu_{i,j}, \sigma_y^2) \quad (Y_{i,j} \text{ independent}) \\ \mu_j &= \beta_0 + \beta_1 \texttt{treatB}_i + \beta_2 \texttt{treatC}_i + \beta_3 \texttt{treatD}_i + \gamma_j \\ \gamma_j &\sim N(0, \sigma_\gamma^2) \end{aligned}$$

This GLMM, `model1`, is fitted to the data.

```
# fit a model with random and fixed effects
model1 <- lmer(yield~treat+(1|blend),data=penicillin,REML=F)
```

A GLMM that doesn't contain the fixed effects, `model2`, is fitted to the data and a likelihood ratio test of `model1` and `model2` is performed to test the significance of the fixed effects.

```
# fit a model without fixed effects
model2 <- lmer(yield~(1|blend),data=penicillin,REML=F)

# likelihood ratio test
ll2 <- logLik(model2)
ll1 <- logLik(model1)
LRT_fixed <- -2*(ll2-ll1)
LRT_fixed <- as.numeric(LRT_fixed)
## 4 treatments so the models differ by 3 parameters
1 - pchisq(LRT_fixed,3)
```

```
## [1] 0.2563946
```

The p-value of the test is larger than 0.05, therefore the null hypothesis that the models are equivalent can be accepted and the fixed effects are deemed to be insignificant.

A GLMM that doesn't contain the random effects, `model3`, is fitted to the data and a likelihood ratio test of `model1` and `model3` is performed to test the significance of the random effects.

```
# fit a model without random effects
model3 <- glm(yield~treat,data=penicillin)
# likelihood ratio test
ll3 <- logLik(model3)
ll1 <- logLik(model1)
LRT_random <- -2*(ll3-ll1)
LRT_random <- as.numeric(LRT_random)
# separated by the variance of the random effects
1 - pchisq(LRT_random,1)
```

```
## [1] 0.06311285
```

The p-value of the test is larger than 0.05, therefore the null hypothesis that the models are equivalent can be accepted and the random effects are deemed to be insignificant.

6b

The likelihood ratio test operates under the assumption that the parameters under the null hypothesis are not on the boundary of the parameter space. I have tested the hypotheses that the variance of the random effect is are zero $H_0 : \sigma^2_\gamma = 0$ and that the the fixed effects are zero $\beta_i = 0$, both of which test if paramters are on the boundary of the parameter space.

Parametric bootstrapping is a more accurate method to estimate p-values when testing the significance of fixed and random effects in a GLMM. This code implements parametric bootstrapping to estimate the p-values of the fixed and random effects of `model1`.

```
n_fixed <- 1000 # number of iterations
sim_LRT_fixed <- 1:n_fixed # vector to store LRT values
Dat <- simulate(model2,n_fixed) # Simulate n_fixed data sets from the smaller model
for(i in 1:n_fixed){
  # fit the models to the simulated data
  Mod2 <- lmer(Dat[,i]~(1|blend),data=penicillin,REML=F)
  Mod1 <- lmer(Dat[,i]~treat+(1|blend),data=penicillin,REML=F)
  sim_LRT_fixed[i] <- -2*(logLik(Mod2)-logLik(Mod1)) # Calculate and store LRT
}
n_random <- 1000 # number of iterations
sim_LRT_random <- 1:n_random # vector to store LRT values
Dat <- simulate(model3,n_random) ### Simulate n_fixed data sets from the smaller model
for(i in 1:n_random){
  # fit the models to the simulated data
  Mod3 <- glm(Dat[,i]~treat,data=penicillin)
  Mod1 <- lmer(Dat[,i]~treat+(1|blend),data=penicillin,REML=F)
  sim_LRT_random[i] <- -2*(logLik(Mod3)-logLik(Mod1)) # Calculate and store LRT
}
# display p-values
print(paste('The p-value of the fixed effects is',mean(sim_LRT_fixed>LRT_fixed)))
```

```
## [1] "The p-value of the fixed effects is 0.341"
```

```
print(paste('The p-value of the random effects is',mean(sim_LRT_random>LRT_random)))
```

```
## [1] "The p-value of the random effects is 0.044"
```

The p-value of the fixed effects using parametric bootstrapping and the likelihood ratio test is larger than 0.05, therefore we can be confident that the fixed effects are not significant.

The p-value for the random effects has decreased using parametric bootstrapping, this is because the usual likelihood ratio test tends to be a conservative method of calculating p-values. The p-value for the random effects is less than 0.05, therefore the null hypothesis can be rejected and we accept the alternative hypothesis that the random effects are significant.

Question 7

7a

A Normal Generalised Linear Model (GLM), `model1`, is fitted to the dataset `pupils` with `test` as the response variable, `IQ` and `ses` as covariates and `Class` as a factor. The model summary shows the significance of the covariates.

```
# fit the glm model to the data
model1 <- glm(test~IQ+ses+Class,data=pupils,family=gaussian(link="identity"))
summary(model1)$coefficients[2:3,] # display covariates
```

```
##      Estimate Std. Error  t value      Pr(>|t|)
## IQ  2.1951549 0.07268296 30.20178 2.202719e-167
## ses 0.1669151 0.01542594 10.82041  1.313781e-26
```

The t-values of the continuous covariates `IQ` and `ses` are both lower than 0.05. Therefore, the effects of `IQ` and `ses` are significant.

A GLM that doesn't contain the factor `Class`, `model2`, is fitted to the data and a likelihood ratio test of `model1` and `model2` is performed to test the significance of the factor `Class`.

```
model2 <- glm(test~IQ+ses,data=pupils,family=gaussian(link="identity"))
ll1 <- logLik(model1)
ll2 <- logLik(model2)
LRT_random <- -2*(ll1-ll2)
LRT_random <- as.numeric(LRT_random)
1 - pchisq(LRT_random,130)
```

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

The p-value of the test is larger than 0.05, therefore the null hypothesis that the models are equivalent can be accepted and the factor `Class` is deemed to be insignificant.

7bi

The variable `Class` contains a random sample of school classes in the Netherlands. Therefore, `Class` can be treated as a random effect so that inference about the the language scores for pupils throughout the Netherlands can be understood.

The number of covariates to model `Class` as a factor is very large. Therefore, `Class` can be treated as a random to reduce the effective number of parameters in the model.

7bii

The response variable `Class`, $Y_{i,j}$, can be modeled by a Normal Generelised Linear Mixed Model (GLMM) with `IQ` and `ses` as the fixed effects and `Class` as the random effects.

$$
\begin{aligned}
Y_{i,j}|\gamma_j &\sim N(\mu_{i,j}, \sigma_y^2) \quad (Y_{i,j} \text{ independent}) \\
\mu_j &= \beta_0 + \beta_1 \texttt{IQ}_i + \beta_2 \texttt{ses}_i + \gamma_j \\
\gamma_j &\sim N(0, \sigma_\gamma^2)
\end{aligned}
$$

7biii

This GLMM, `model3`, is fitted to the data and a model summary is obtained.

```
# git the GLMM model to the data
model3 <- lmer(test~IQ+ses+(1|Class),data=pupils)
summary(model3) # model summary
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: test ~ IQ + ses + (1 | Class)
##    Data: pupils
##
## REML criterion at convergence: 15140.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0086 -0.6609  0.0571  0.7075  3.0931
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Class    (Intercept)  9.212   3.035
##  Residual             40.049   6.328
## Number of obs: 2287, groups:  Class, 131
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  9.41092    0.87139   10.80
## IQ           2.25325    0.07138   31.57
## ses          0.16538    0.01479   11.18
##
## Correlation of Fixed Effects:
##     (Intr) IQ
## IQ  -0.827
## ses -0.181 -0.293
```

The t-values of the fixed effect covariates are larger than 1.96 and are therefore significant at the 5% level. This conclusion is based on the assumption that β_1 and β_2 are normally distributed around the mean.

7biv

The between-class variance σ^2_γ of `model3`, is

```
between_var <- as.numeric(VarCorr(model3)$Class)
between_var
```

```
## [1] 9.212033
```

The within-class variance σ^2_y of `model3`, is

```r
within_var <- summary(model3)$sigma^2
within_var
```

```
## [1] 40.04893
```

The marginal variance of the response of a GLMM is equal to the sum of the between-class and the within-class variance.

$$\text{var}[Y_{i,j}] = \sigma_\gamma^2 + \sigma_y^2$$

```r
total_var <- between_var + within_var # marginal variance of model3
total_var
```

```
## [1] 49.26097
```

The variance of the GLM, `model1`, (created in part a) has a variance of

```r
summary(model1)$dispersion # marginal variance of model1
```

```
## [1] 39.98367
```

The variance of the GLM, `model1`, is equal to the within-class variance σ_y^2 of the GLMM, `model2`. The random effects in the GLMM add extra variance, σ_γ^2 to the model.

7bv

A GLM, `model2`, that doesn't contain the factor random effects of `Class` is fitted to the data and a likelihood ratio test of `model3` and `model2` is performed to test the significance of the factor `Class`.

```r
# fit a glm to the data
model2 <- glm(test~IQ+ses,data=pupils,family=gaussian(link="identity"))
# likelihood ratio test
ll2 <- logLik(model2)
ll3 <- logLik(model3)
LRT <- -2*(ll2-ll3)
LRT <- as.numeric(LRT)
# separated by the variance of the random effects
1-pchisq(LRT,1) # chai-squared goodness of fit test with 1 df
```

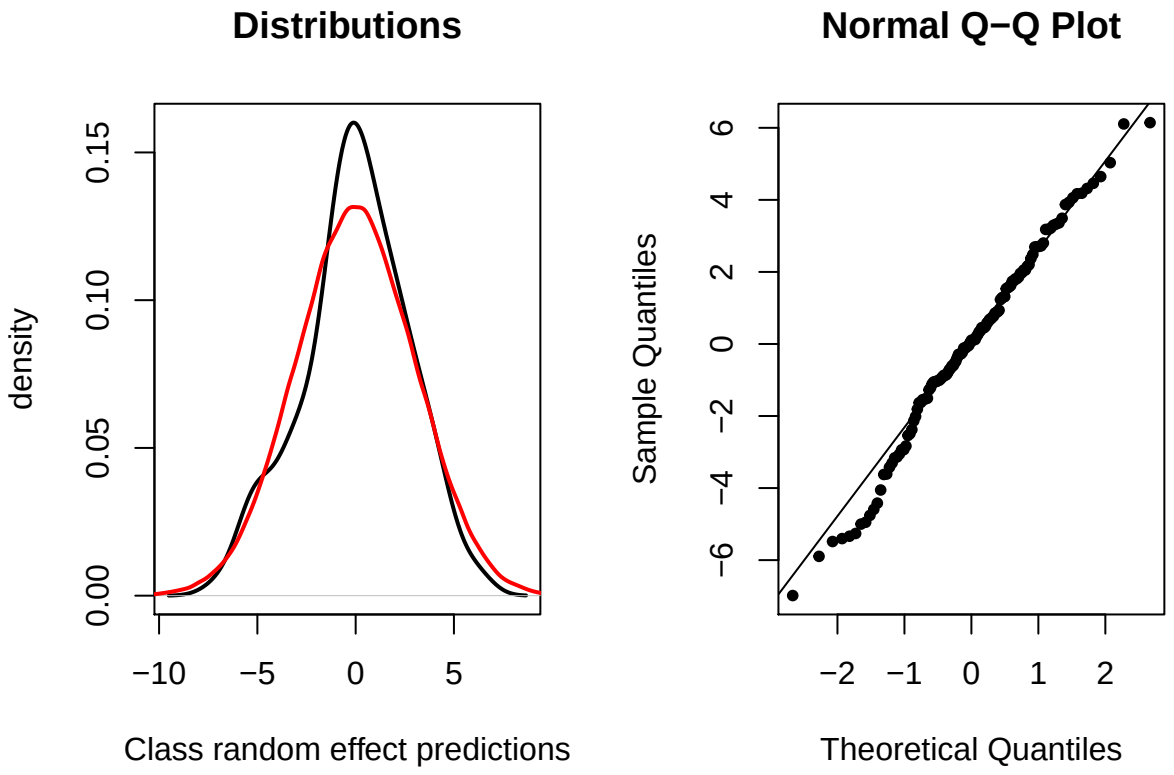
```
## [1] 0
```

The p-value of the test is less than 0.05, therefore the null hypothesis that the models are equivalent is rejected and the random effect of `CLass` is deemed to be significant, hence $sigma_\gamma^2 \neq 0$.

7bvi

The predicted random effects of `model3` are calculated and compared with a Normal distribution.

```r
par(mfrow=c(1, 2))
# calculate the standard deviation of the random effects in the model
between_var <- as.numeric(VarCorr(model3)$Class)
between_std <- sqrt(between_var)
# obtain the random effects of the model
class_ran <- ranef(model3)$Class[,1]
# plot the random effects
plot(density(class_ran),lwd=2,xlab="Class random effect predictions",
     ylab="density", main = "Distributions")
## add theoretical Gaussian distribution with sd = between_std
lines(density(rnorm(100000,0,between_std)),col="red",lwd=2)
qqnorm(class_ran,pch=20)
qqline(class_ran)
```



The plots show that the predicted random effects are roughly normally distributed. There is a slight deviation from the Normal distribution in the lower tail.

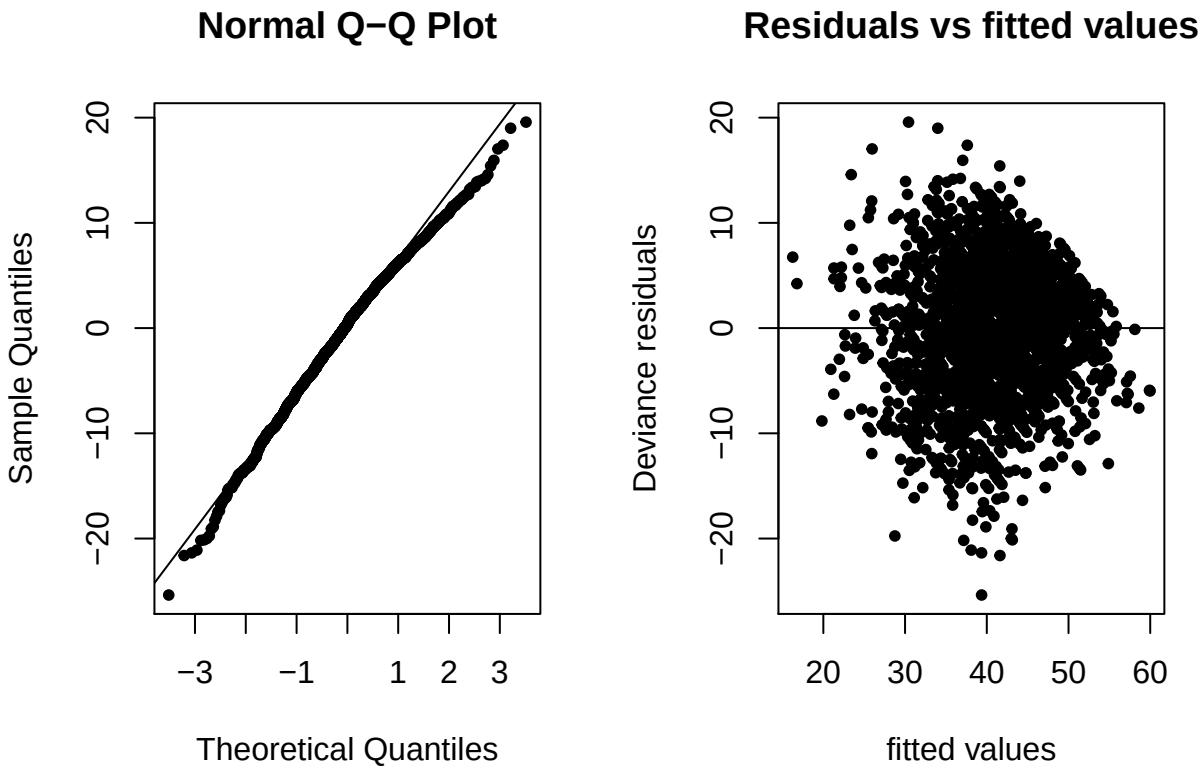
7bvii

The predictions of `model3` $\hat{y}$ can be compared with the true values y to obtain the residuals $y - \hat{y}$. A QQ plot and a scatter plot of the residuals are produced.

```r
par(mfrow=c(1, 2))
```

```r
# produce a QQ plot of the residuals
resids <- residuals(model3) # residuals of model3
qqnorm(resids,pch=20)
qqline(resids)

fitted <- predict(model3) # fitted values of model3
# residual plot
plot(fitted,resids,ylab="Deviance residuals", xlab="fitted values",
     main="Residuals vs fitted values", pch=20)
abline(h=0)
```



The QQ plot shows that the residuals are Normally distributed, with a deviation in both of the tails. A scatter plot of the residuals with respect to the fitted values shows that the variance of the residuals decreases as y increases.

7ci

The GLMM fitted in biii, `model3`, treats the covariate `IQ` as constant across classes. However, there may be class level variables that have an effect on how `IQ` relates to the test results in each class. To account for this, I fit a model, `model4`, to the data that treats `IQ` as a random effect. As `IQ` is now allowed to vary within classes, it will have a larger effect on `test`.

```r
# fit model4 using maximum likelihood
model4 <- lmer(test~IQ+ses+(1+IQ|Class),data=pupils,REML=F) # IQ varies with class
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0237085 (tol = 0.002, component 1)
```

7cii

A likelihood ratio test of `model3` and `model4` is performed to test the significance of the random slope of `IQ`

```
# likelihood ratio test
ll3 <- logLik(model3)
ll4 <- logLik(model4)
LRT <- -2*(ll3-ll4)
LRT <- as.numeric(LRT)
# There are two parameter that separate the models
1-pchisq(LRT,2) # chai-squared goodness of fit test with 2 df
```

```
## [1] 1.658521e-06
```

The p-value of the test is less than 0.05, therefore the null hypothesis that the models are equivalent is rejected and the random effect of `IQ` is deemed to be significant.

Quesion 9

9a

The variable `municipality` contains a random sample of municipalities in Spain. Therefore, `municipality` can be treated as a random effect so that inference about the the hip fracture rate in the elderly throughout Spain can be understood.

Treating `municipality` as a random effect incorporates unobserved factors into the model by including random intercepts for individual observations. This allows the model to handle overdispersion in the data.

9b

The response variable, `Nfract`, is the count of the number of fractures and can be suitably modeled by a Poisson GLMM with `ses` and `sex` as the fixed effects and `municipality` as the random effects. An offset of log(`Npop`) is required to obtain the rate of hip fractures per 1000 people.

$$
\begin{aligned}
Y_{i,j}|\gamma_j &\sim Pois(\lambda_{i,j}) \quad (Y_{i,j} \text{ independent}) \\
\log(\lambda_j) &= \beta_0 + \beta_1 \texttt{sex2}_i + \beta_2 \texttt{ses2}_i + \beta_3 \texttt{ses3}_i + \gamma_j + \log(\texttt{Npop}_i) \\
\gamma_j &\sim N(0, \sigma^2_\gamma)
\end{aligned}
$$

9c

This model, `model1`, is fitted to the data and a model summary is produced.

```
model1 <- glmer(Nfract~offset(I(log(Npop)))+sex+ses+(1|municipality),
                data=hip,family=poisson(link="log")) # fit the GLMM
summary(model1) # model summary
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Nfract ~ offset(I(log(Npop))) + sex + ses + (1 | municipality)
##    Data: hip
##
##      AIC      BIC   logLik deviance df.resid
##  29512.4  29546.8 -14751.2  29502.4     7223
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.3230 -0.7754 -0.0956  0.7154  5.3739
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  municipality (Intercept) 0.04975  0.2231
## Number of obs: 7228, groups:  municipality, 278
##
## Fixed effects:
##             Estimate Std. Error  z value Pr(>|z|)
```

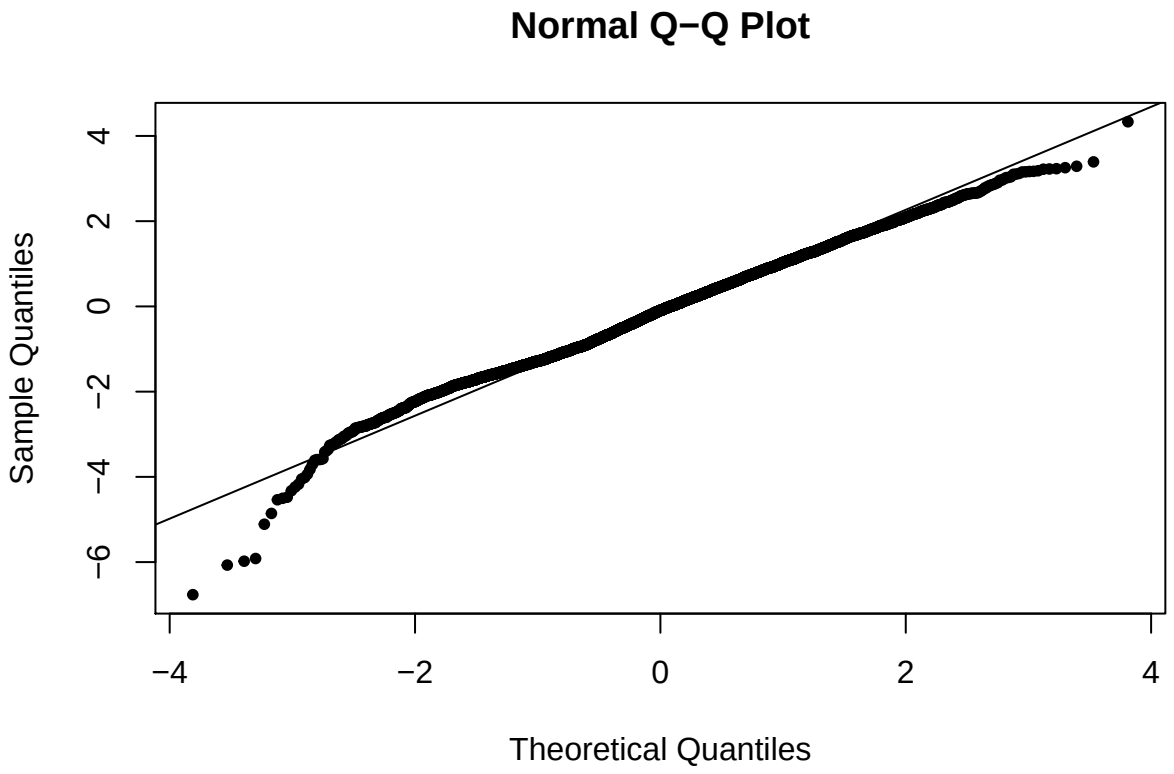
```
## (Intercept) -4.56923    0.02717 -168.143  < 2e-16 ***
## sex2         0.74854    0.01247   60.051  < 2e-16 ***
## ses2        -0.09401    0.03576   -2.629  0.00856 **
## ses3        -0.01831    0.03728   -0.491  0.62328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr) sex2   ses2
## sex2 -0.372
## ses2 -0.655  0.005
## ses3 -0.628  0.005  0.476
```

The z-values of the fixed effects are less than 0.05 for `ses2` and `sex2`, therefore these fixed effects are significant. The z-value of `ses3` is not larger than 0.05, therefore this fixed effect is not significant.

9d

A QQ plot of the residuals of `model1` is produced.

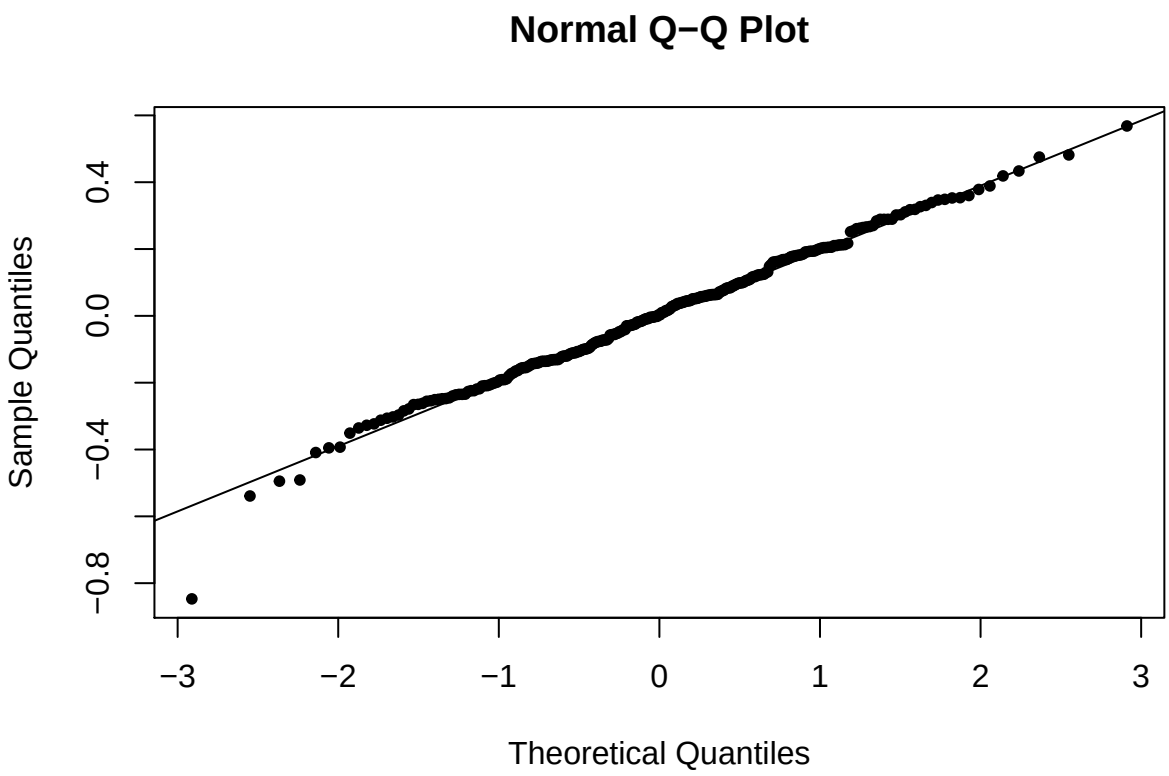
```
resids <- residuals(model1) # obtain residuals of the model
qqnorm(resids,pch=20) # QQ plot of the residuals
qqline(resids)
```



The QQ plot shows that the residuals are Normally distributed, with deviation in both of the tails.

A QQ plot of the random effects of `model1`, γ_j is produced.

```
municipality_ran <- ranef(model1)$municipality[,1] # the random effects of the model
qqnorm(municipality_ran,pch=20) # produce a QQ plot of the random effects
qqline(municipality_ran)
```



The QQ plot shows that they are Normally distributed, with a deviation in the lower tail.

9e

A GLM, `model0`, that doesn't contain the random effects of `municipality` is fitted to the data and a likelihood ratio test of `model1` and `model0` is performed to test the significance of the random effects of `municipality`.

```
# fit a glm without the random effects
model0 <- glm(Nfract~offset(I(log(Npop)))+sex+ses, data=hip,
              family=poisson(link="log"))
# likelihood ratio test
ll2 <- logLik(model0)
ll3 <- logLik(model1)
LRT <- -2*(ll2-ll3)
LRT <- as.numeric(LRT)
```

```r
# 1 paramter seperating the models (variance of random effects)
1 - pchisq(LRT,1) # chai-squared goodness of fit test
```

```
## [1] 0
```

The p-value of the test is less than 0.05, therefore the null hypothesis that the models are equivalent is rejected and the random effect of `municipality` is deemed to be significant.