

FirstModelsPressuresPaco

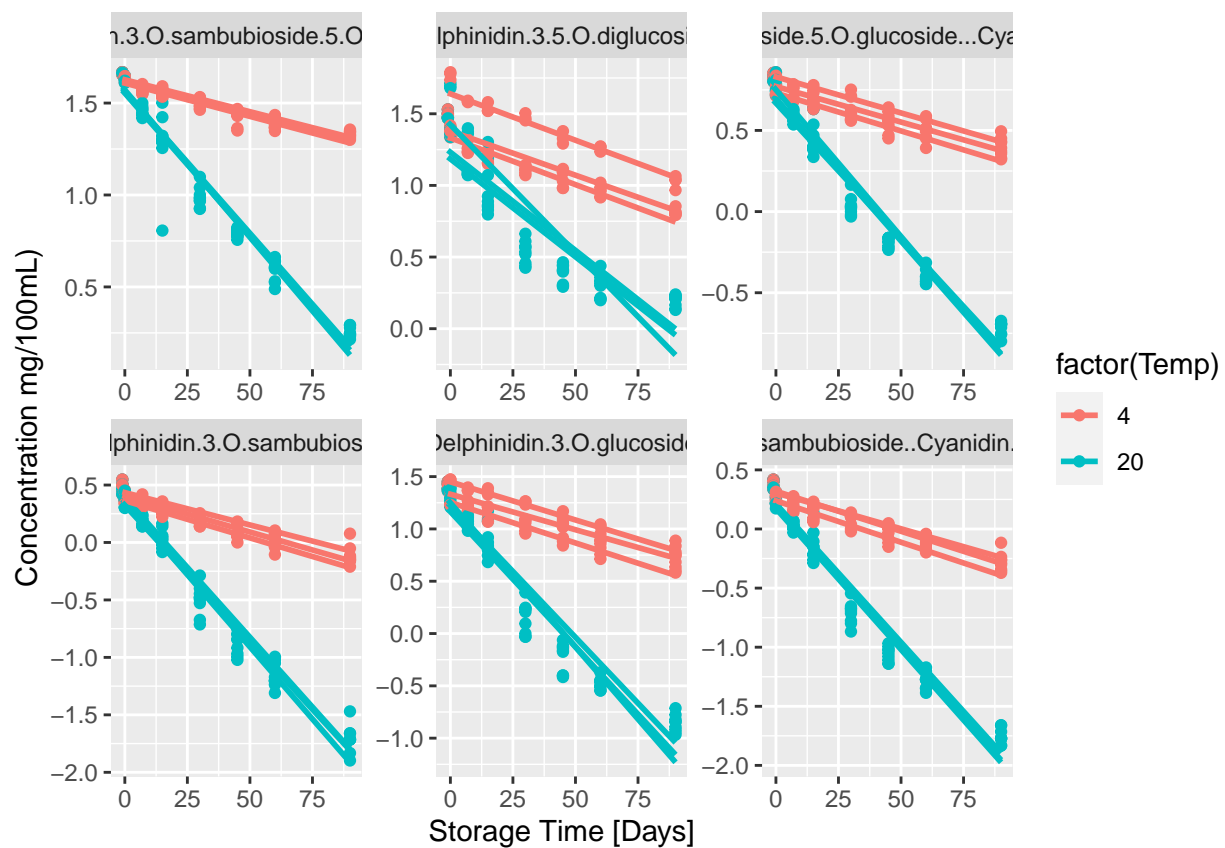
En primer lugar se:

1. Leen los datos
2. Colocan en formato de tabla larga
3. Genera la columna de temperaturas
4. Factorizan las columnas necesarias para el modelado covariante
5. Seleccionan antocianos

Para empezar, generamos una gráfica de primer orden aparente para estudiar la importancia de la temperatura

```
ggplot(anthocyanins,
  aes(x = tiempo, y = log(concentration), group = sweetener:processing:factor(Temp), color = factor(Temp))
  facet_wrap(compound ~ ., scales = "free") +
  geom_point() + geom_smooth(method = "lm", se = F) +
  xlab("Storage Time [Days]") + ylab("Concentration mg/100mL")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



De aquí podemos concluir que:

1. La temperatura de almacenamiento es el factor más importante del estudio. Las diferencias entre

colores son mucho mayores que las diferencias entre líneas del mismo color

2. Todas las antocianinas salvo Delphinidin.3.5.O.diglucoside podrían razonablemente ser modeladas con una cinética de primer orden aparentemente.

Procedemos a la primera modelización, una degradación de primer orden aparente con una concentración residual y una energía de activación, un modelo de referencia, con dependencia de la temperatura pero ni procesado ni endulzante afectan, con la siguiente distribución, derivada de la ecuación de Arrhenius y la cinética de primer orden:

$$C(t)_i \sim C_{\infty_i} + (C_{0_i} - C_{\infty_i})e^{(-e^{(lk_i - \frac{Ea_i}{R} \frac{1}{T} - \frac{1}{T_{ref}})})t}$$

Donde $C(t)_i$ es la concentración del antocianino i a tiempo t y C_{0_i} , C_{∞} , lk_i and Ea_i son la concentración inicial, concentración final, el logaritmo natural de la constante aparente de primer orden y la energía de activación para el antocianino i . La temperatura de referencia es $16^\circ C$ y la constante R es $0.008314 \frac{KJ}{Mol \cdot K}$.

```
ant.fm0<-gnls(concentration~Cinf+(C0-Cinf)*exp(-exp(lk-Ea/8.314e-3*(1/(Temp+273)-1/(16+273))))*tiempo),
  data=anthocyanins,
  param=list(C0~compound,
    Cinf~compound,
    lk~compound,
    Ea~compound),
  start=c(C0=c(1,rep(0.001,5)),
    Cinf=c(0,rep(0.001,5)),
    lk=c(0.01,rep(0.001,5)),
    Ea=c(10,rep(0.001,5)))
)
summary(ant.fm0,cor=F)
```

Modelización de la co-varianza

Con el diseño experimental se pueden estimar los efectos principales de las variables endulzante y procesado para compuesto. La estimación de interacciones y los términos de orden superior requeriría completar este diseño.

En este modelo endulzante y procesado son efectos iguales para todos los compuestos.

```
ant.fm1<-gnls(concentration~Cinf+(C0-Cinf)*exp(-exp(lk-Ea/8.314e-3*(1/(Temp+273)-1/(16+273))))*tiempo),
  data=anthocyanins,
  param=list(C0~compound+sweetener+processing,
    Cinf~compound+sweetener+processing,
    lk~compound+sweetener+processing,
    Ea~1),
  start=c(C0=c(coef(ant.fm0)[1:6],rep(0.001,3)),
    Cinf=c(coef(ant.fm0)[7:12],rep(0.001,3)),
    lk=c(coef(ant.fm0)[13:18],rep(0.001,3)),
    Ea=c(coef(ant.fm0)[19])
  )
summary(ant.fm1)
```

Ahora intentaremos estimar los diferentes coeficientes de constantes entre diferentes componentes y condiciones de procesamiento

```
ant.fm2<-gnls(concentration~Cinf+(C0-Cinf)*exp(-exp(lk-Ea/8.314e-3*(1/(Temp+273)-1/(16+273))))*tiempo),
  data=anthocyanins,
  param=list(C0~compound+sweetener+processing,
    Cinf~compound+sweetener+processing,
```

```

        lk~compound+sweetener+compound:processing,
        Ea~compound),
    start=c(C0=c(coef(ant.fm0)[1:6],rep(0.001,3)),
            Cinf=c(coef(ant.fm0)[7:12],rep(0.001,3)),
            lk=c(coef(ant.fm0)[13:18],rep(0.001,13)),
            Ea=c(coef(ant.fm0)[19],rep(0.001,5))
    )
)
summary(ant.fm2)

```

Obtenemos aproximadamente cuatro parámetros significativos. Ea es solo diferente en D.3.5.0.diglucosido

Realizamos un análisis anova para los resultados de los modelos:

```
anova(ant.fm0,ant.fm1,ant.fm2)
```

```
##           Model df          AIC          BIC    logLik    Test  L.Ratio p-value
## ant.fm0      1 25    71.25632  190.29564 -10.62816
## ant.fm1      2 29 -177.22100 -39.13539  117.61050 1 vs 2 256.4773 <.0001
## ant.fm2      3 44 -291.24731 -81.73811  189.62366 2 vs 3 144.0263 <.0001
```

Las mejoras en los modelos son muy significativas y mejoran el ruido

Realizamos un cuarto modelo, con los coeficientes de los factores en todas las variables

```

ant.fm3<-gnls(concentration~Cinf+(C0-Cinf)*exp(-exp(lk-Ea/8.314e-3*(1/(Temp+273)-1/(16+273))))*tiempo),
              data=anthocyanins,
              param=list(C0~compound+compound:sweetener+compound:processing,
                          Cinf~compound+compound:sweetener+compound:processing,
                          lk~compound+compound:sweetener+compound:processing,
                          Ea~compound),
              start=c(C0=c(coef(ant.fm0)[1:6],rep(0.001,18)),
                      Cinf=c(coef(ant.fm0)[7:12],rep(0.001,18)),
                      lk=c(coef(ant.fm0)[13:18],rep(0.001,18)),
                      Ea=c(coef(ant.fm0)[13:18])
              )
)
screenreg(ant.fm3,single.row=T,ci.force=T)

```

De nuevo, evaluamos los modelos con anova, y miramos el r2 del cuarto modelo

```
anova(ant.fm0,ant.fm1,ant.fm2,ant.fm3)
```

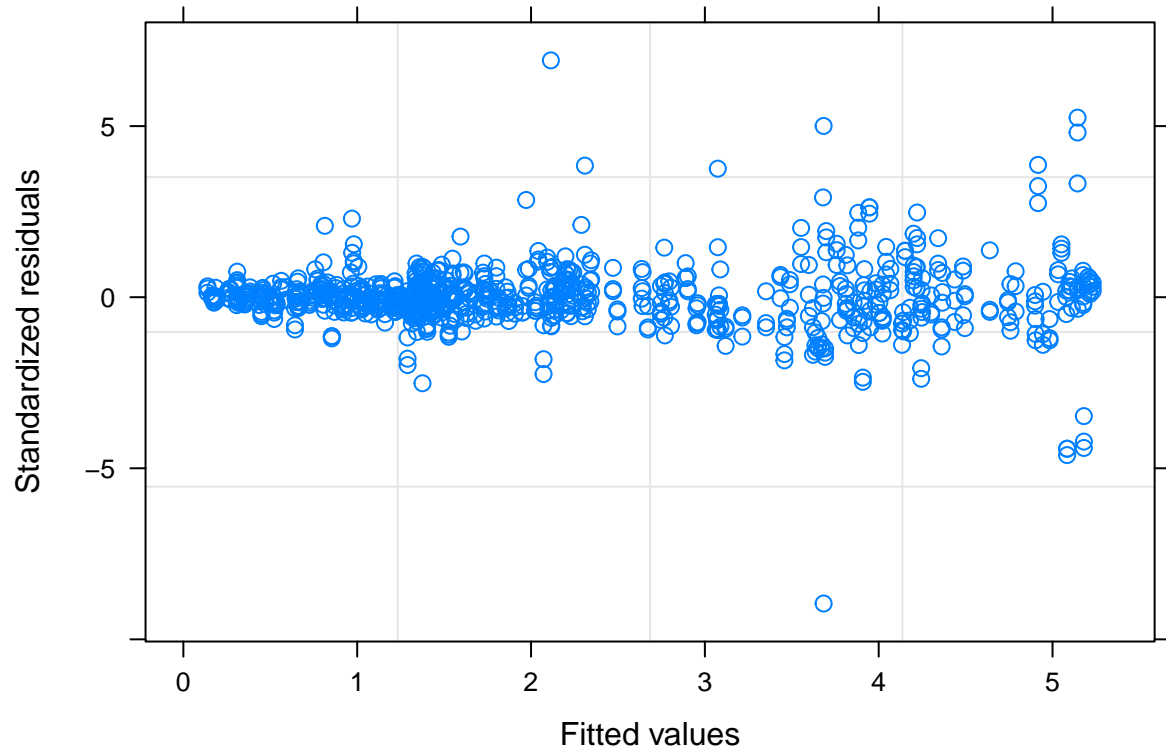
```
##           Model df          AIC          BIC    logLik    Test  L.Ratio p-value
## ant.fm0      1 25    71.2563  190.29564 -10.6282
## ant.fm1      2 29 -177.2210 -39.13539  117.6105 1 vs 2 256.4773 <.0001
## ant.fm2      3 44 -291.2473 -81.73811  189.6237 2 vs 3 144.0263 <.0001
## ant.fm3      4 79 -625.5766 -249.41238  391.7883 3 vs 4 404.3293 <.0001
```

```
r2(ant.fm3)
```

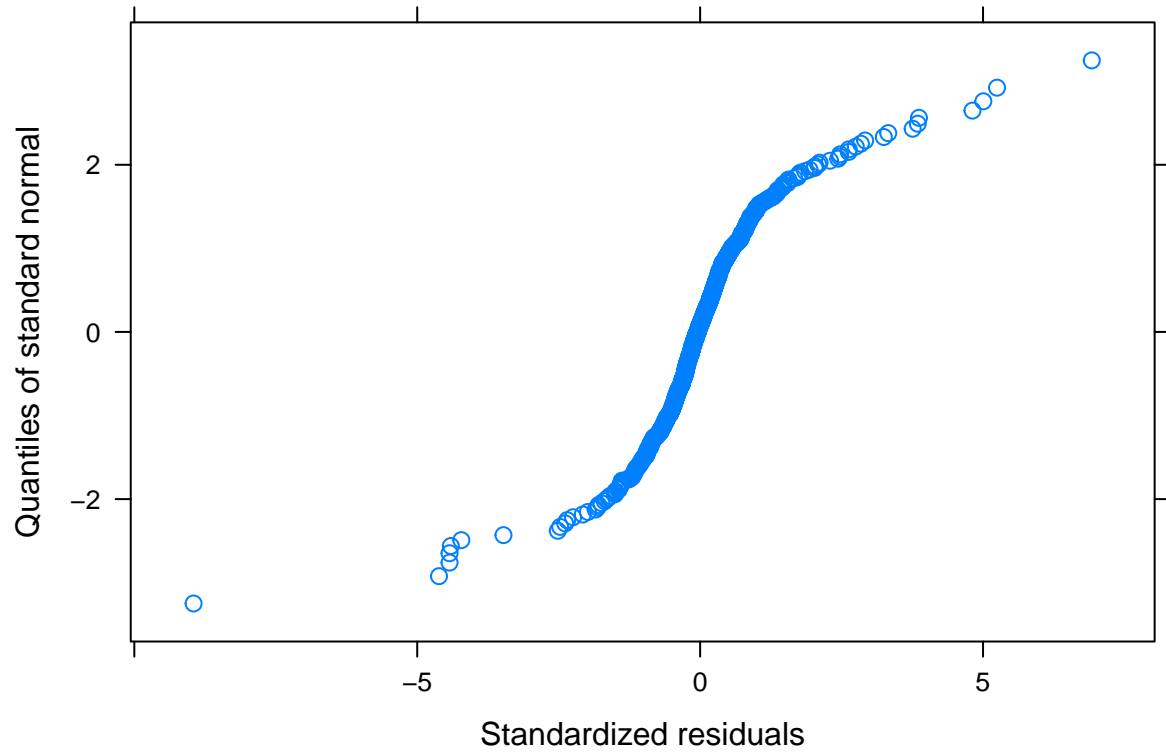
```
## [1] 0.9886055 0.9874892
```

Realizamos gráficas de diagnóstico para el cuarto modelo

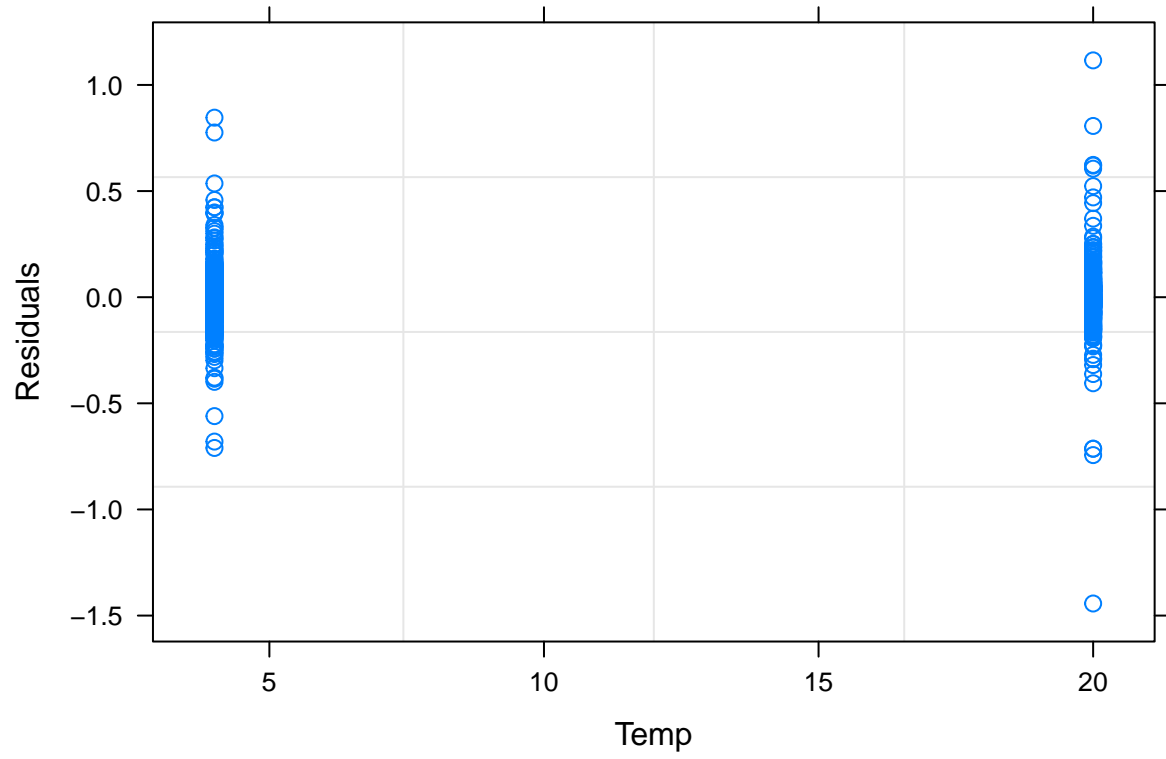
```
plot(ant.fm3)
```



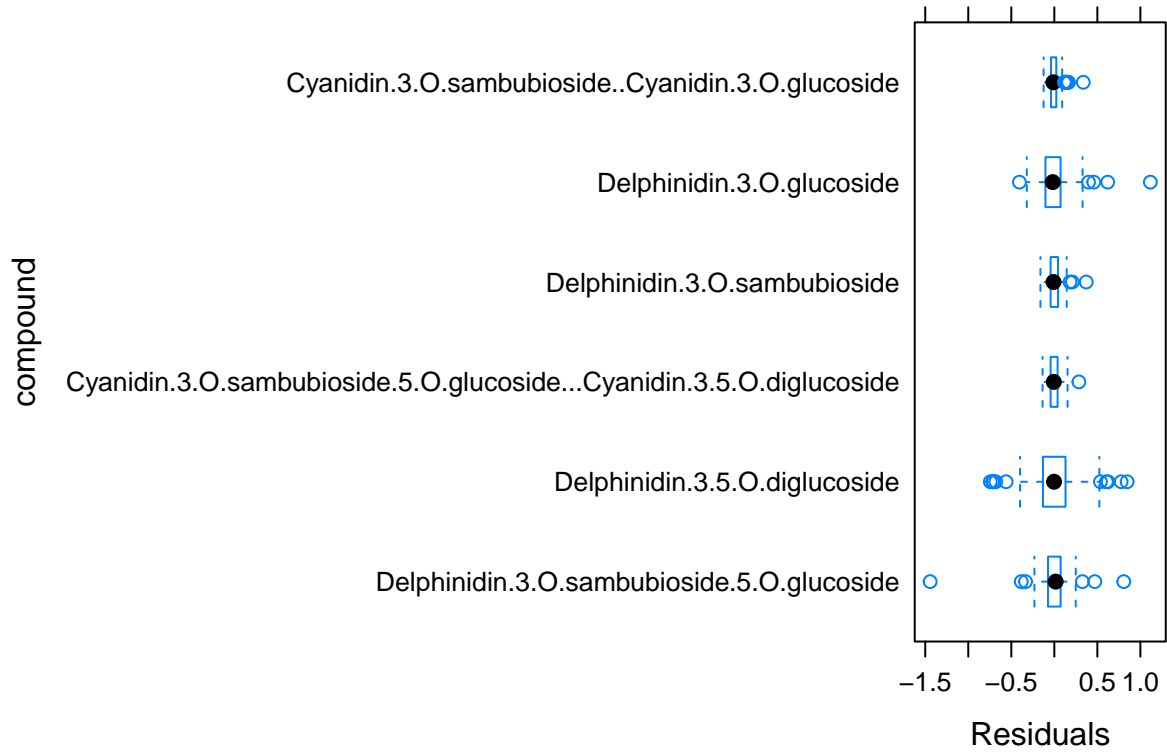
```
qqnorm(ant.fm3)
```



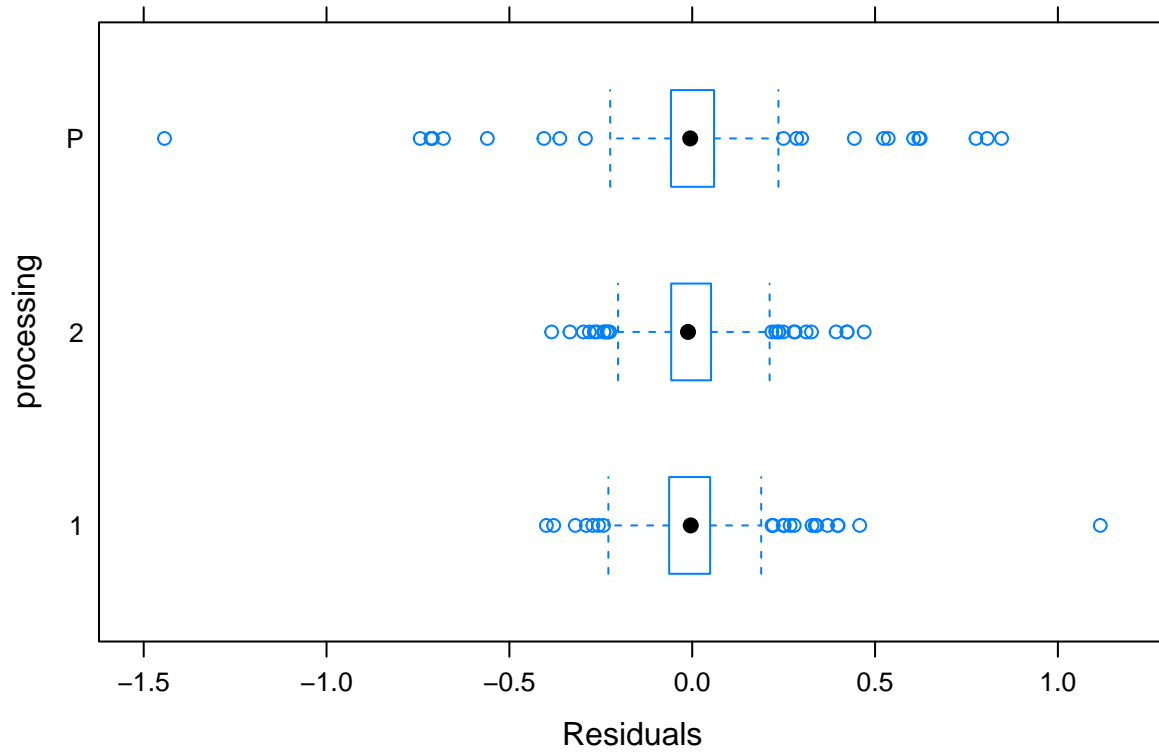
```
plot(ant.fm3,resid(.)~Temp)
```



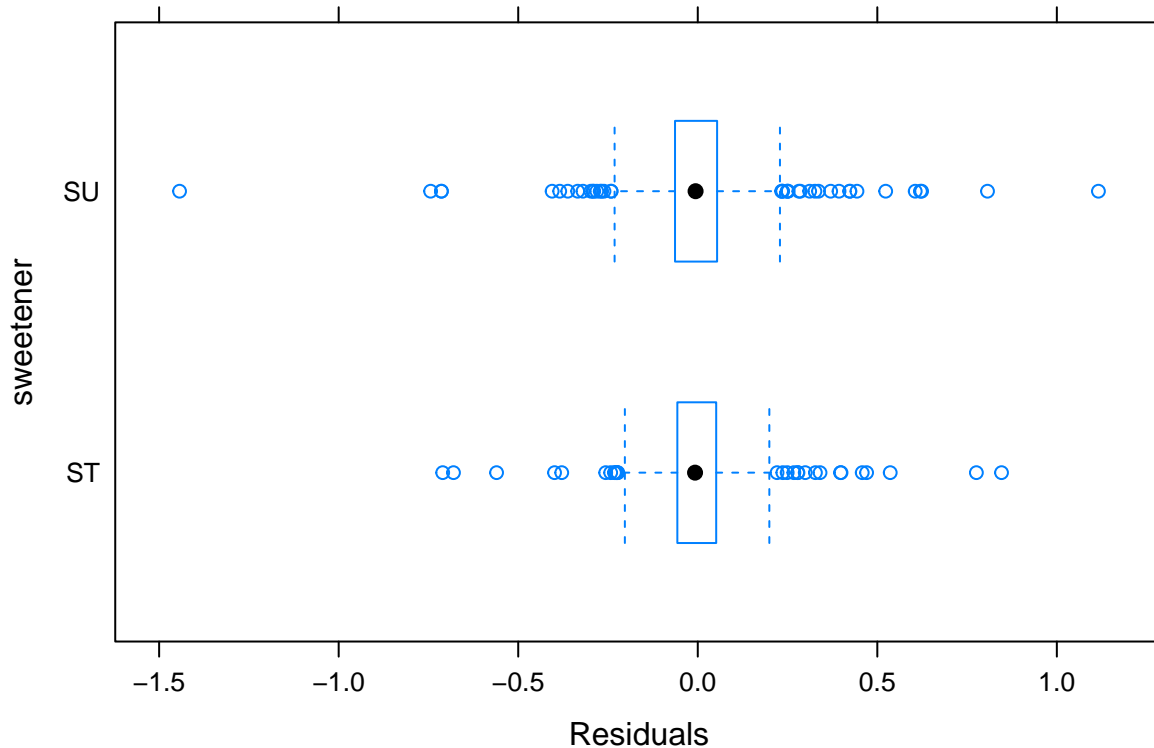
```
plot(ant.fm3,compound~resid(.))
```



```
plot(ant.fm3,processing~resid(.))
```



```
plot(ant.fm3,sweetener~resid(.))
```

Gráfica 1, Residuos vs Ajustes: Se confirma que los residuos se distribuyen aleatoriamente y con varianza constante.

Gráfica 2, probabilidad normal de residuos: Se confirma que los residuos siguen una probabilidad normal

Gráficas 3,4,5 y 6, residuos vs variables: Las variables tienen importancia en el modelo al observar patrones no aleatorios

Creamos una tabla a partir de los resultados, la guardamos como “anthocyanins.html” y generamos gráficas para evaluar las predicciones del cuarto modelo, viéndose la predicción del ajuste en las líneas y los datos del experimento en puntos:

```
list.of.compound.fit<-list()
list.of.r2adj<-list()
for (i in levels(factor(anthocyanins$compound))) {
  #print(i)
  list.of.compound.fit[[i]]<-gnls(concentration~Cinf+(C0-Cinf)*exp(-exp(lk-Ea/8.314e-3*(1/(Temp+273))-1/
    data=subset(anthocyanins,compound==i),
    param=list(C0~sweetener+processing,
              Cinf~sweetener+processing,
              lk~sweetener+processing,
              Ea~1),
    start=c(C0=c(5.05,rep(0.001,3)),
            Cinf=c(0,rep(0.001,3)),
            lk=c(-4.38,rep(0.001,3)),
            Ea=c(67)
  )
}
```

```
list.of.r2adj[[i]]<-round(r2(list.of.compound.fit[[i]])[2],3)
}  
  
#screenreg(list.of.compound.fit,single.row=T,ci.force=T)
```

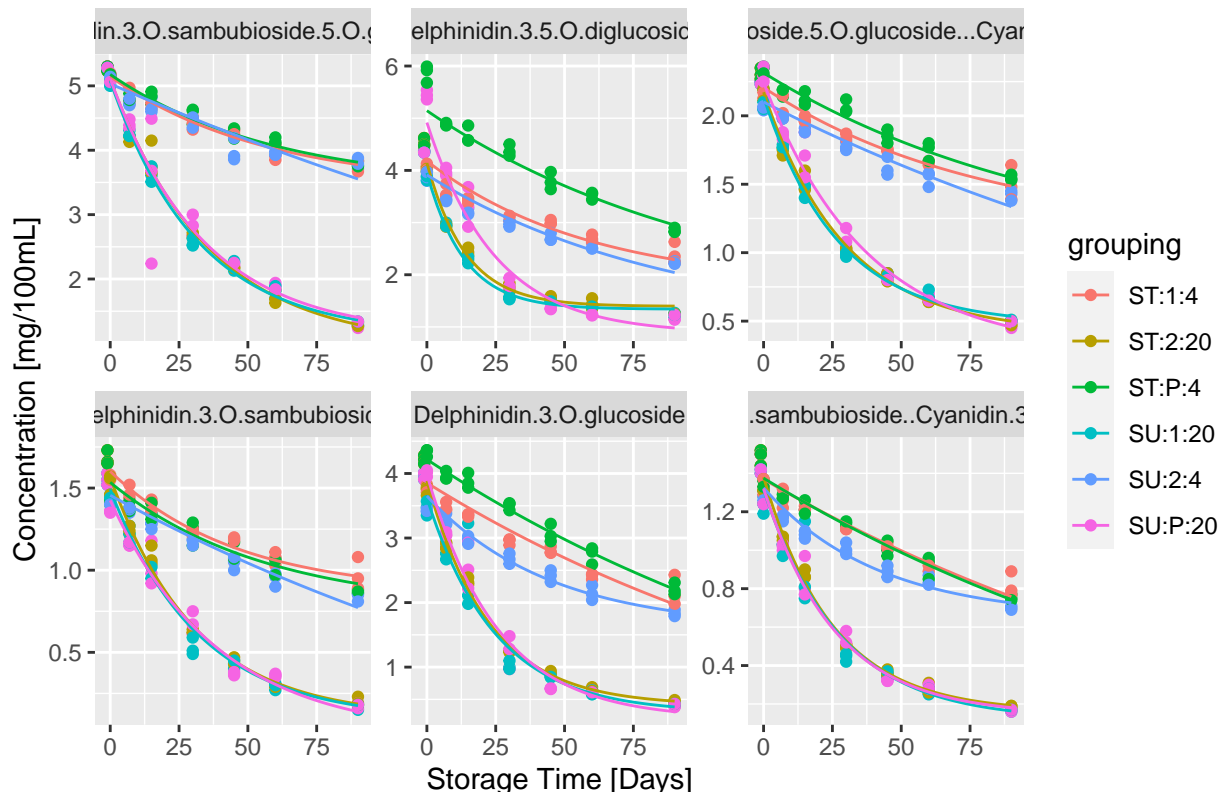
```
## The table was written to the file 'anthocyanins.html'.
```

```

anthocyanins$grouping<-with(anthocyanins,sweetener:processing:factor(Temp))
ant.pred<-expand.grid(tiempo=seq(0,90,length=50),
                      compound=levels(factor(anthocyanins$compound)),
                      grouping=levels(factor(anthocyanins$grouping))
                      )
ant.pred$sweetener<-factor(with(ant.pred,substr(grouping,0,2)))
ant.pred$processing<-factor(with(ant.pred,substr(grouping,4,4)))
ant.pred$Temp<-as.numeric(with(ant.pred,substr(grouping,6,8)))
ant.pred$concentration<-predict(ant.fm3,newdata = ant.pred)
ggplot(anthocyanins,
      aes(x = tiempo, y = concentration, col =grouping)) +
  facet_wrap(compound~.,scales="free")+
  geom_point()+geom_line(data=ant.pred,aes(x=tiempo,y=concentration,col=grouping))+
  ggtitle("Degradation kinetic modelled from experimental (lines) vs experimental (dots)")+
  xlab("Storage Time [Days]")+ylab("Concentration [mg/100mL]")

```

Degradation kinetic modelled from experimental (lines) vs experimental (dots)



```
ggsave(filename="Figure1.pdf")
```

Saving 6.5 x 4.5 in image

Podemos concluir que este cuarto modelo ajusta bastante bien la degradación del compuesto respecto a los datos experimentales salvo algún punto marginal, y que puede emplearse para realizar predicciones bastante ajustadas en el caso de los antocianos.

Por tanto, generamos una grilla de datos para simular las condiciones en las que carecemos de información, realizamos la predicción y representamos cada condición a todos los compuestos:

```
ant.simul2<-expand.grid(tiempo=seq(0,90,length=30),
```

```

Temp = c(20,4),
compound=levels(factor(anthocyanins$compound)),
grouping=levels(factor(anthocyanins$grouping))
)

```

```

ant.simul2$sweetener<-factor(with(ant.simul2,substr(grouping,0,2)))
ant.simul2$processing<-factor(with(ant.simul2,substr(grouping,4,4)))
ant.simul2$grouping<-with(ant.simul2,sweetener:processing:factor(Temp))

```

```

ant.simul2$prediction<-predict(ant.fm3,newdata = ant.simul2)

```

```

ggplot(ant.simul2,
  aes(x = tiempo, y = prediction, col = factor(Temp))) +
  facet_grid(processing+sweetener~factor(compound),scales="free")+
  geom_point()+geom_line()+
  xlab("Storage Time [Days]")+ylab("Concentration [mg/100mL]")

```

Vemos que las predicciones se comportan de forma acorde a lo esperado según las muestras experimentales.

Ahora procedemos a representar cada una de las condiciones separando las temperaturas, para comparar los datos experimentales con los simulados

```

# ST:1

```

```

ggplot(data = subset(ant.simul2,grouping == "ST:1:20"), aes(x=tiempo, y = prediction, col = factor(Temp))) +
  facet_wrap(compound~.,scales="free")+
  geom_point()+ geom_point(data=subset(anthocyanins,grouping == "ST:1:4"), aes(x=tiempo, y=concentration)) +
  ggtitle("ST:1 20° modelled | 4° experimental") + xlab("Storage Time [Days]")+ylab("Concentration [mg/100mL]")

```

```

# # ST:2

```

```

# ggplot(data = ant.simul2 %>% filter(grouping == "ST:2:4"), aes(x=tiempo, y = prediction, col = factor(Temp))) +
#   facet_wrap(compound~.,scales="free")+
#   geom_point()+geom_point(data=anthocyanins %>% filter(grouping == "ST:2:20"), aes(x=tiempo, y=concentration)) +
#   ggtitle("SU:1 20° experimental | 4° modelled")+ xlab("Storage Time [Days]")+ylab("Concentration [mg/100mL]")
#

```

```

# # ST:P

```

```

# ggplot(data = ant.simul2 %>% filter(grouping == "ST:P:20"), aes(x=tiempo, y = prediction, col = factor(Temp))) +
#   facet_wrap(compound~.,scales="free")+
#   geom_point()+geom_point(data=anthocyanins %>% filter(grouping == "ST:P:4"), aes(x=tiempo, y=concentration)) +
#   ggtitle("ST:P 20° modelled | 4° experimental")+ xlab("Storage Time [Days]")+ylab("Concentration [mg/100mL]")
#

```

```

# # SU:1

```

```

# ggplot(data = ant.simul2 %>% filter(grouping == "SU:1:4"), aes(x=tiempo, y = prediction, col = factor(Temp))) +
#   facet_wrap(compound~.,scales="free")+
#   geom_point()+geom_point(data=anthocyanins %>% filter(grouping == "SU:1:20"), aes(x=tiempo, y=concentration)) +
#   ggtitle("SU:1 20° experimental | 4° modelled")+ xlab("Storage Time [Days]")+ylab("Concentration [mg/100mL]")
#

```

```

# # SU:2

```

```

# ggplot(data = ant.simul2 %>% filter(grouping == "SU:2:20"), aes(x=tiempo, y =prediction, col = factor(Temp))) +
#   facet_wrap(compound~.,scales="free")+

```

```

#   geom_point()+geom_point(data=anthocyanins %>% filter(grouping == "SU:2:4"), aes(x=tiempo, y=concentracion))
#   ggtitle("SU:2 20° modelled | 4° experimental")+ xlab("Storage Time [Days]")+ylab("Concentration [mg/L]")
#
# # SU:P
#
# ggplot(data = ant.simul2 %>% filter(grouping == "SU:P:4"), aes(x=tiempo, y = prediction, col = factor(grouping)))
#   facet_wrap(compound~., scales="free")+
#   geom_point()+geom_point(data=anthocyanins %>% filter(grouping == "SU:P:20"), aes(x=tiempo, y=concentracion))
#   ggtitle("SU:P 20° experimental | 4° modelled")+ xlab("Storage Time [Days]")+ylab("Concentration [mg/L]")

```