Guía 1 - Ejercicio 7

Bajo el mismo archivo (abalone.txt), trate de establecer una relación entre el peso total y el diámetro del espécimen. Empezar dibujando en un scatter plot ambos parámetros. Basándose en eso, considere los siguientes modelos:
a) Modelo lineal simple: $Peso = b + aDiametro$.
b) Modelo cuadrático: $Peso = c + bDiametro + Diametro^2$.
c) Modelo cúbico sin términos de orden inferior: $Peso = aDiametro^3$.
d) Modelo exponencial: $log(Peso) = b + aDiametro$.

1. Efectue en cada caso una regresión y grafique las curvas superpuestas sobre el scatter plot.
2. Identifique el "mejor" modelo mediante un esquema de training y testing set, quedándose con el que minimiza el error cuadrá tico medio en el conjunto de testing.

Aclaración: La idea es que al comenzar el ejercicio, de la muestra total, que es muy grande, elijan el 70% de los datos, llamémoslo training set. Con esa submuestra se arman los modelos. Luego, para comparar los modelos, se los aplica sobre el restante 30% de los datos, llamémoslo testing set, y calculan el error de predicción (esto es el RSS que se vió en clase pero con la muestra testing para calcular el y_ajustado). En base a esto es que se elije el mejor modelo.

Separación del dataset en train y testing

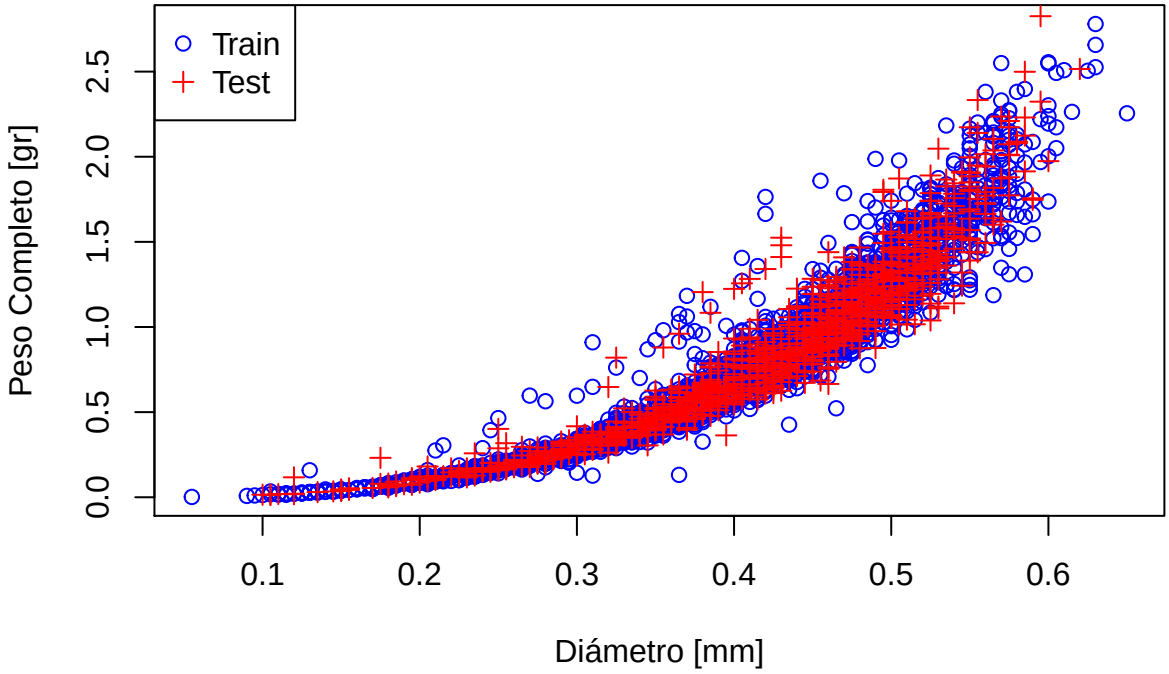
Antes de empezar a trabajar con los datos, vamos a generar un conjunto de testing y uno de training.

```
# Algunas definiciones útiles:
q <- seq(0,3,0.01) #Defino un vector de abcsisas para futuros gráficos
porc_train<-0.7 #Porcentaje del dataset que se utilizará en el training set
porc_test<-1-porc_train #Porcentaje del dataset que se utilizará en el test set
cols<-c("Sexo","Long","Diam", "Alt", "PesoComp","PesoCarne","PesoVísc","PesoCapr","Anillos") # Nombres
mydata <-read.table("abalone.txt",header = FALSE, sep = ",",col.names=cols) #Cargo los datos del datase
set.seed(123) #Para repetibilidad del experimento
tam_data<-nrow(mydata) #Cantidad de muestras en el dataset
train_ind <- sample(seq_len(tam_data), size = round(tam_data*porc_train))
train <- mydata[train_ind, ]
test <- mydata[-train_ind, ]
```

Ahora vamos a graficar los datos

```
# A continuación cumplo con el primer requisito del enunciado, realizar un scatter plot de Peso Complet

plot(train$Diam, train$PesoComp, main="Peso Completo vs. Diámetro", xlab="Diámetro [mm]", ylab="Peso Co
points(test$Diam, test$PesoComp,pch=3,col='red')
legend("topleft",c("Train","Test"), col=c("blue","red"), pch=c(1,3))
```



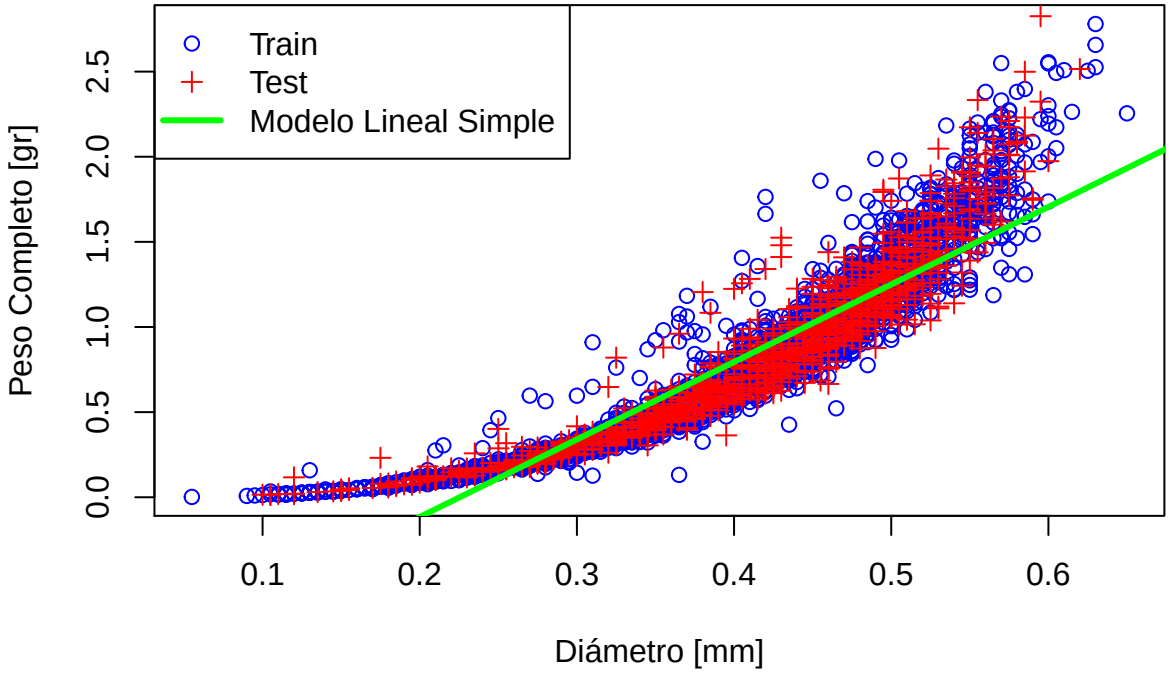
##

Regresión lineal simple:

Vamos a realizar la regresión lineal simple cumpliendo con la forma:

$$Peso = b + aDiametro$$

```
modeloA = lm( PesoComp ~ Diam, data = train) #Hago la regresión lineal simple estimando el Peso Complet
predictedA <- predict(modeloA,data.frame(Diam=q)) # Para realizar la gráfica de la estimación sobre el
plot(train$Diam, train$PesoComp, main="Peso Completo vs. Diámetro", xlab="Diámetro [mm]", ylab="Peso Co
points(test$Diam, test$PesoComp,pch=3,col='red')
lines(q,predictedA,col='green',lwd=3)
legend("topleft",c("Train","Test","Modelo Lineal Simple"), col=c("blue","red","green"), pch=c(1,3,NA),lw
```




```
summary(modeloA)
```

```
##
## Call:
## lm(formula = PesoComp ~ Diam, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56819 -0.12562 -0.03949  0.07161  0.98125
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02644    0.01443  -71.14   <2e-16 ***
## Diam         4.55298    0.03443  132.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1872 on 2922 degrees of freedom
## Multiple R-squared:  0.8569, Adjusted R-squared:  0.8568
## F-statistic: 1.749e+04 on 1 and 2922 DF,  p-value: < 2.2e-16
```

Calculemos el error de predicción para el test set:

```
predictedA <- predict(modeloA,test)
RSSA_Test=sum((predictedA - test$PesoComp)^2)
cat("El error cuadrático de predicción para el test set es:")
```

```
## El error cuadrático de predicción para el test set es:
```

```
cat(RSSA_Test)
```

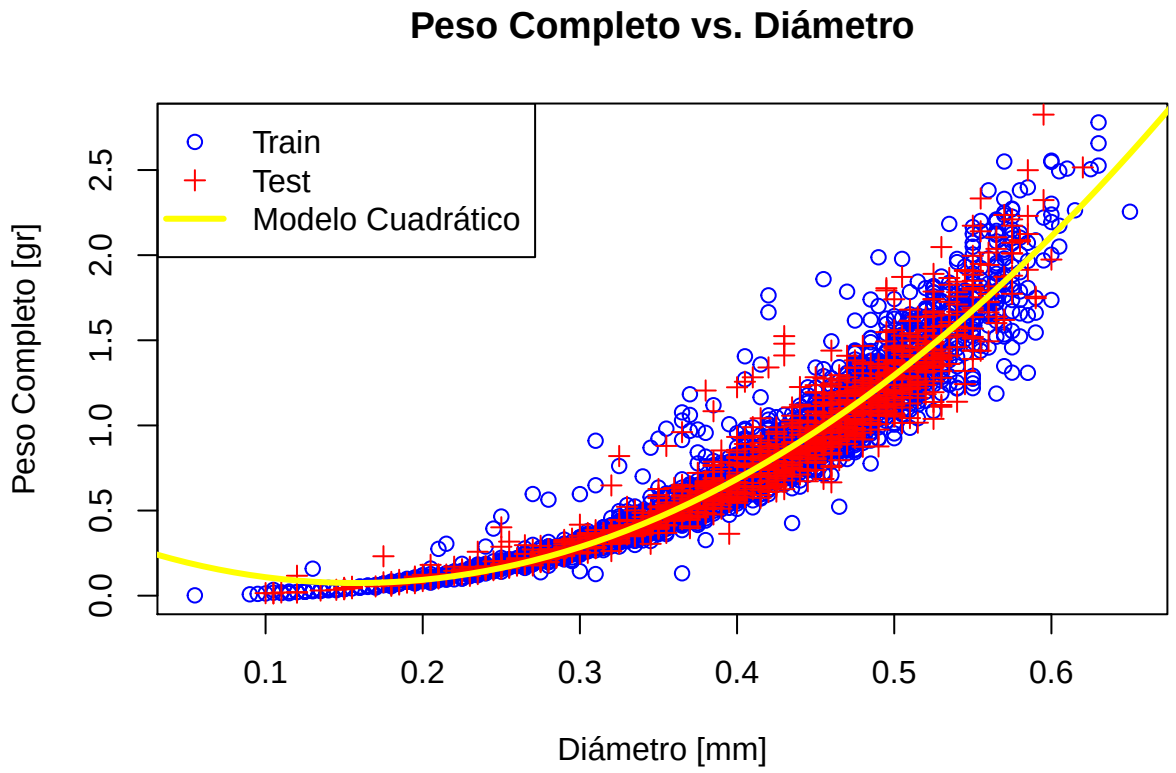
```
## 41.76318
```

Modelo Cuadrático

Aplicando el modelo cuadrático al train set, nos queda:

$$Peso = c + bDiametro + Diametro^2$$

```
modeloB = lm( PesoComp ~ poly(Diam,2), data = train) #Entreno el modelo cuadrático
predictedB <- predict(modeloB,data.frame(Diam=q)) # Para realizar la gráfica de la estimación sobre el
plot(train$Diam, train$PesoComp, main="Peso Completo vs. Diámetro", xlab="Diámetro [mm]", ylab="Peso Co
points(test$Diam, test$PesoComp,pch=3,col='red')
lines(q,predictedB,col='yellow',lwd=3)
legend("topleft",c("Train","Test","Modelo Cuadrático"), col=c("blue","red","yellow"), pch=c(1,3,NA),lwd
```




```
summary(modeloB)
```

```
##
## Call:
## lm(formula = PesoComp ~ poly(Diam, 2), data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66565 -0.06374 -0.00625  0.04596  0.97328
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.826076   0.002448  337.51   <2e-16 ***
## poly(Diam, 2)1 24.759734   0.132349  187.08   <2e-16 ***
## poly(Diam, 2)2  7.159033   0.132349   54.09   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1323 on 2921 degrees of freedom
## Multiple R-squared:  0.9285, Adjusted R-squared:  0.9284
## F-statistic: 1.896e+04 on 2 and 2921 DF,  p-value: < 2.2e-16
```

Calculemos el error de predicción para el test set:

```
predictedB <- predict(modeloB,test)
RSSB_Test=sum((predictedB - test$PesoComp)^2)
cat("El error cuadrático de predicción para el test set es:")
```

```
## El error cuadrático de predicción para el test set es:
```

```
cat(RSSB_Test)
```

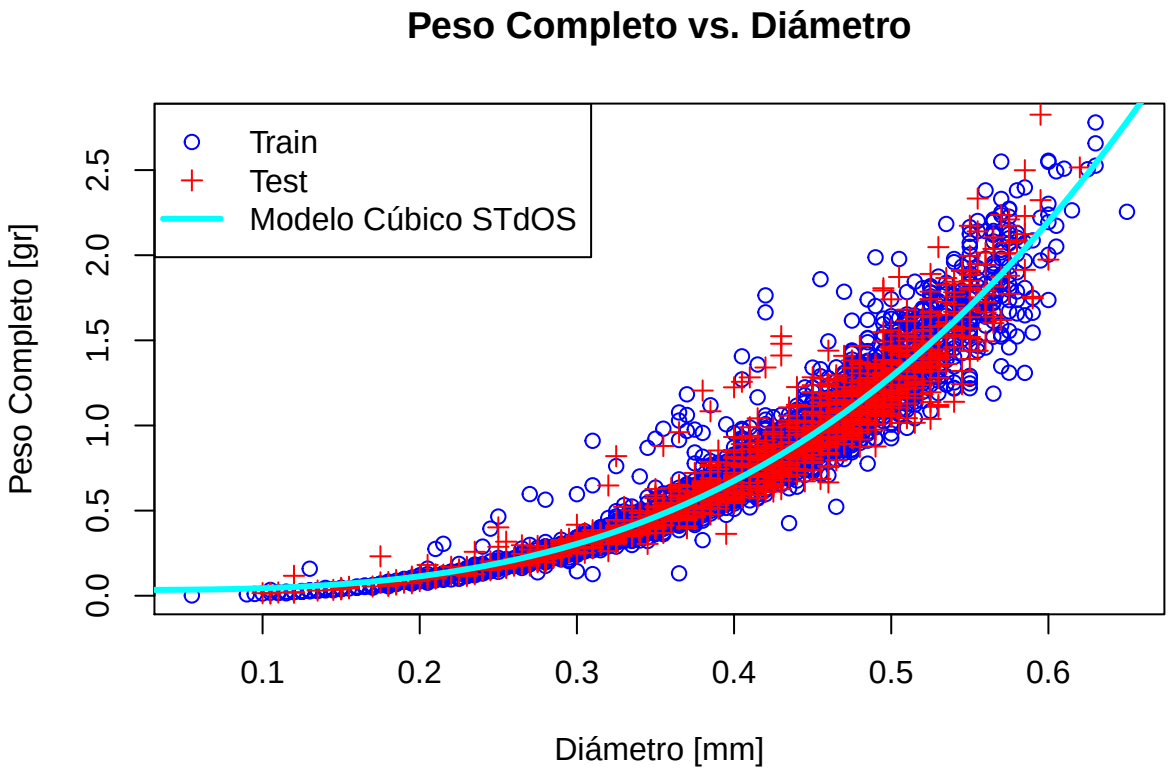
```
## 22.39172
```

Modelo cúbico sin términos de orden superior

Aplicando el modelo cúbico sin términos de orden superior al train set, nos queda:

$$Peso = aDiametro^3$$

```
modeloC = lm( PesoComp ~ I(Diam^3), data = train) #Entreno el modelo cúbico sin términos de orden super
predictedC <- predict(modeloC,data.frame(Diam=q)) # Para realizar la gráfica de la estimación sobre el
plot(train$Diam, train$PesoComp, main="Peso Completo vs. Diámetro", xlab="Diámetro [mm]", ylab="Peso Co
points(test$Diam, test$PesoComp,pch=3,col='red')
lines(q,predictedC,col='cyan',lwd=3)
legend("topleft",c("Train","Test","Modelo Cúbico STdOS"), col=c("blue","red","cyan"), pch=c(1,3,NA),lwd=
```



```r
summary(modeloC)
```

```
##
## Call:
## lm(formula = PesoComp ~ I(Diam^3), data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72975 -0.05807 -0.01516  0.04883  0.98800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03417    0.00473   7.223 6.46e-13 ***
## I(Diam^3)   10.01285    0.05124 195.423  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1319 on 2922 degrees of freedom
## Multiple R-squared:  0.9289, Adjusted R-squared:  0.9289
## F-statistic: 3.819e+04 on 1 and 2922 DF,  p-value: < 2.2e-16
```

Calculemos el error de predicción para el test set:

```r
predictedC <- predict(modeloC,test)
RSSC_Test=sum((predictedC - test$PesoComp)^2)
cat("El error cuadrático de predicción para el test set es:")
```

```
## El error cuadrático de predicción para el test set es:
```

```r
cat(RSSC_Test)
```

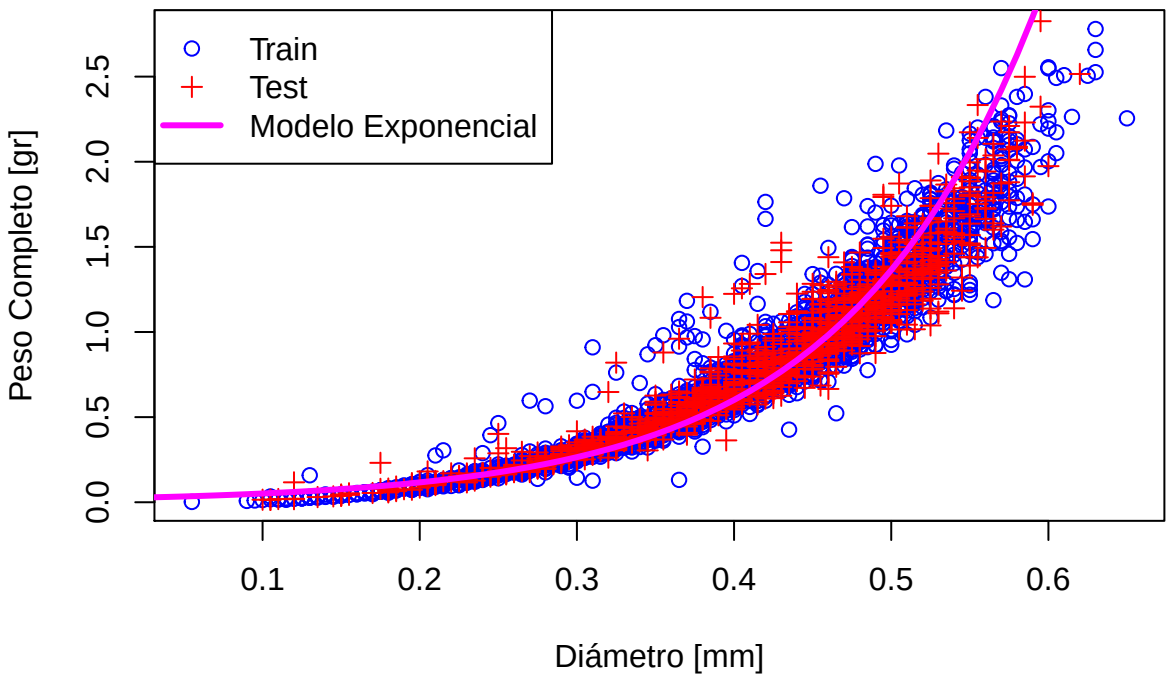
```
## 21.9383
```

Modelo exponencial

Aplicando el modelo exponencial al train set, nos queda:

$$log(Peso) = b + aDiametro$$

```r
modeloD = lm( log(PesoComp) ~ Diam, data = train) #Entreno el modelo exponencial
predictedD <- exp(predict(modeloD,data.frame(Diam=q))) # Para realizar la gráfica de la estimación sobr
plot(train$Diam, train$PesoComp, main="Peso Completo vs. Diámetro", xlab="Diámetro [mm]", ylab="Peso Co
points(test$Diam, test$PesoComp,pch=3,col='red')
lines(q,predictedD,col='magenta',lwd=3)
legend("topleft",c("Train","Test","Modelo Exponencial"), col=c("blue","red","magenta"), pch=c(1,3,NA),l
```




```
summary(modeloD)
```

```
##
## Call:
## lm(formula = log(PesoComp) ~ Diam, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.87800 -0.09943  0.01574  0.12273  1.15249
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.78735    0.01770  -214.0   <2e-16 ***
## Diam         8.19532    0.04223   194.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2297 on 2922 degrees of freedom
## Multiple R-squared:  0.928,  Adjusted R-squared:  0.928
## F-statistic: 3.766e+04 on 1 and 2922 DF,  p-value: < 2.2e-16
```

Calculemos el error de predicción para el test set:

```
predictedD <- exp(predict(modeloD,test))
RSSD_Test=sum((predictedD - test$PesoComp)^2)
cat("El error cuadrático de predicción para el test set es:")
```

```
## El error cuadrático de predicción para el test set es:
```

```
cat(RSSD_Test)
```

```
## 45.56018
```

Resultados y conclusiones

Para los cuatro modelos los RSS obtenidos fueron:

- Lineal simple: 41.76318
- Cuadrático: 22.39172
- Cúbico sin términos de orden superior: 21.9383
- Exponencial: 45.56018

El modelo que cumple con el requisito pedido en el punto 2 es el modelo cúbico sin términos de orden superior, es decir, el modelo C.
Con respecto a este resultado, surge un tema mencionado en las primeras clases, que es la interpretabilidad del modelo. Siendo que en el modelo de menor error el peso depende del cubo del diámetro, podríamos interpretar lo siguiente:

"Si suponemos que la densidad de un abalone se mantiene constante entre uno y otro, el peso dependería linealmente del volúmen, el cual depende del cubo de las dimensiones lineales, en nuestro caso, el diámetro".