Chapter 7: Power law linear models

STATS 201/8

University of Auckland

Learning Outcomes

In this chapter you will learn about:

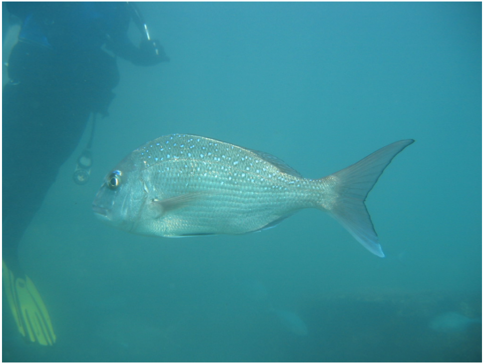
- Power law models
- How to interpret the effect of the explanatory variable
- Relevant `R`-code.

Section 7.1
Power law model example

Example – Weight of snapper as a function of length

Those of you who fish in the Hauraki Gulf will know that the minimum legal size for retaining a snapper is 30 cm. Here, we want to use snapper length to explain snapper weight, and in particular we want to estimate the weight of 30 cm snapper.

Note that this research question is highly relevant, since the relationship between length and weight is crucial for the stock assessments of snapper.



Weight of snapper as a function of length

What does our intuition tell us about the shape of the relationship between length and weight?

- Straight line?
- Quadratic?
- Exponential?
- Other?

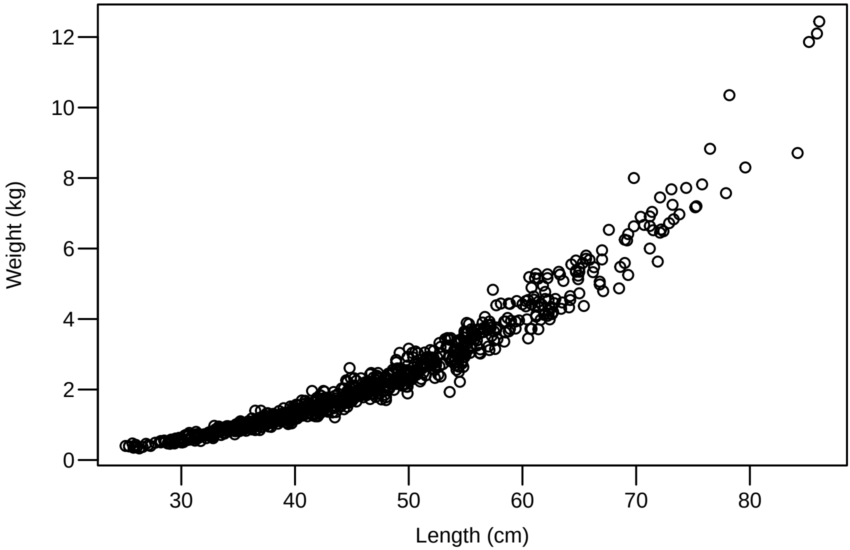
Weight of snapper as a function of length...

The data file `SnapWgt.txt` contains measurements on 844 snapper. The variables are:

`len`	fork length (cm)
`wgt`	weight (kg)

```
> Snap.df=read.table("SnapWgt.txt",header=TRUE)
> plot(wgt~len,data=Snap.df,xlab="Length (cm)",ylab="Weight (kg)")
```

Weight of snapper as a function of length...



Weight of snapper as a function of length...

Clearly there is a non-linear relationship between weight and length.

Geometry tells us that if an object changes in overall size while keeping the same shape (i.e., same ratio between height, depth and length), then its volume will increase with the 3rd power of length.

- For a cube with sides of length l, $volume = len^3$.
- For a sphere with radius r, $volume = \frac{4}{3}\pi r^3$.

That is, $volume \propto len^3$. In other words

$$volume = k_1 \times len^3$$

for some constant k_1.

Assuming weight of a solid object is proportional to its volume, we conclude

$$weight = \alpha \times len^3$$

for some constant α.

Weight of snapper as a function of length...

Those of you who have caught snapper will know that they do exhibit a small change in shape as they grow larger, so it would be better to use the model

$$\textit{weight} = \alpha \times \textit{len}^{\beta_1}$$

where β_1 is some constant that may be close to, but not necessarily equal to 3.

Taking logs gives

$$\log(\textit{weight}) = \log(\alpha) + \beta_1 \times \log(\textit{len})$$

which we can rewrite as

$$\log(\textit{weight}) = \beta_0 + \beta_1 \times \log(\textit{len}) \ .$$

Weight of snapper as a function of length...

The formula on the previous slide specifies an assumed (i.e, expected) relationship between $\log(weight)$ and $\log(len)$ of a snapper.

Of course, snapper of a given length will have some variability in their weight, just as humans of a given height vary in their weight. So, what we are really saying is that the weight (kg) of an individual snapper of length *len* (cm) is

$$\log(weight) = \beta_0 + \beta_1 \times \log(len) + \varepsilon$$

where ε is some random variability (i.e., error around the expected value).

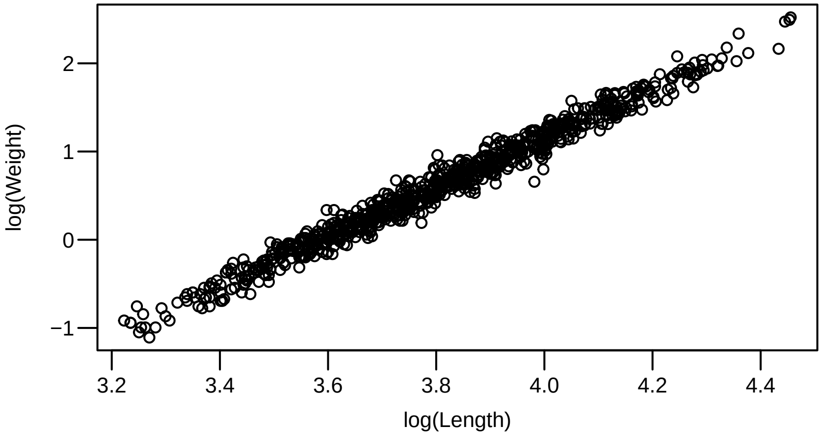
The above formula should be of very familiar form to you by now. Provided that we make the assumption that $\varepsilon \sim N(0, \sigma^2)$ then this is precisely the simple linear regression model with response variable $\log(weight)$ and explanatory variable $\log(len)$.

Weight of snapper as a function of length...

Fitting a simple linear model using log(weight) and log(length)

Let us look at the relationship between log(wgt) and log(len)

```
> plot(log(wgt)~log(len),data=Snap.df,xlab="log(Length)",ylab="log(Weight)")
```

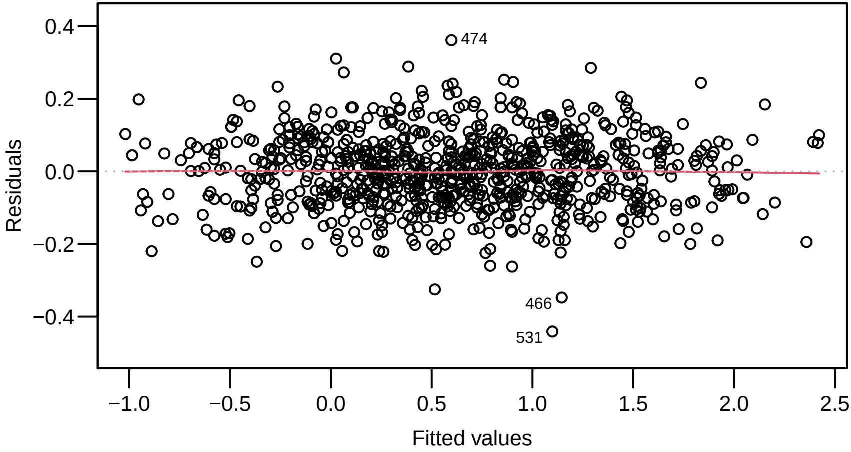


Looking good.

Weight of snapper as a function of length...

Fitting a simple linear model using log(weight) and log(length)...

```
> Snap.lm=lm(log(wgt)~log(len),data=Snap.df)
> plot(Snap.lm,which=1)
```

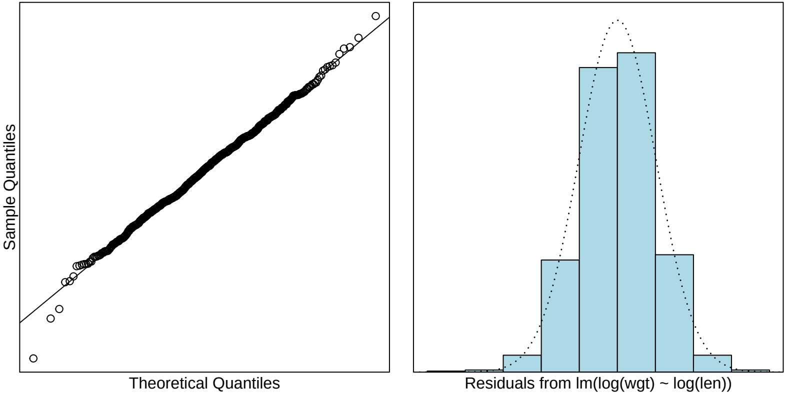


Weight of snapper as a function of length...

Fitting a simple linear model using log(weight) and log(length)...

Check the Normality assumption.

```
> normcheck(Snap.lm)
```

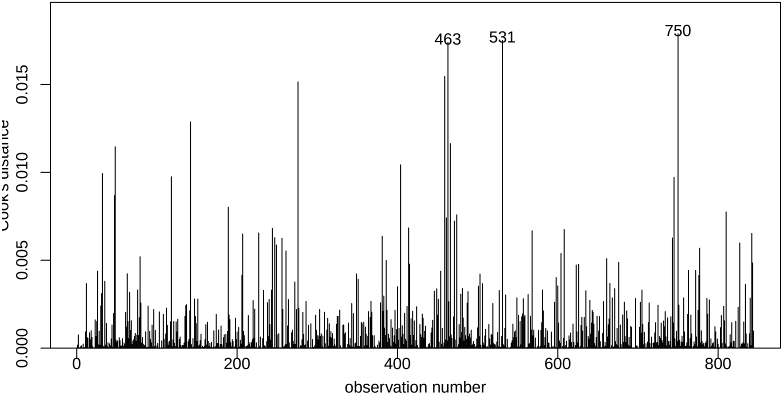


Weight of snapper as a function of length...

Fitting a simple linear model using log(weight) and log(length)...

Check for influential observations.

```
> cooks20x(Snap.lm)
```



Weight of snapper as a function of length...

Making inference

We can trust the fitted model.

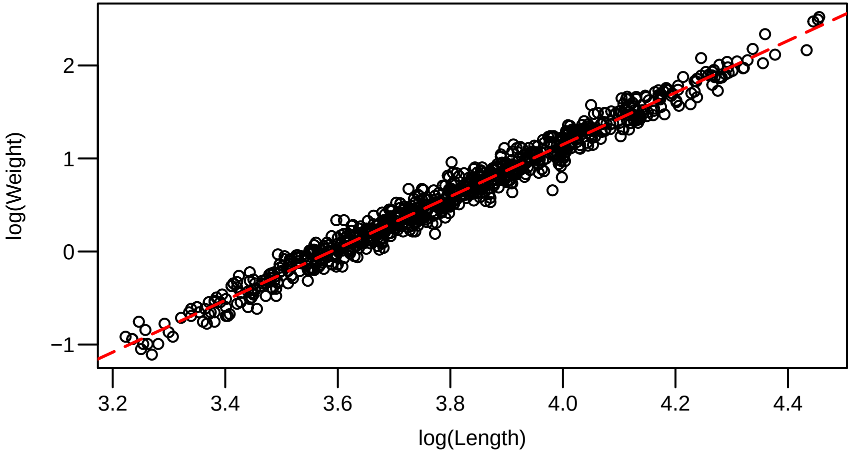
```
> summary(Snap.lm)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.01416    0.05602  -178.7   <2e-16 ***
log(len)      2.79104    0.01469   190.0   <2e-16 ***
---
Residual standard error: 0.1012 on 842 degrees of freedom
Multiple R-squared:  0.9772,Adjusted R-squared:  0.9772
F-statistic: 3.609e+04 on 1 and 842 DF,  p-value: < 2.2e-16
> confint(Snap.lm)
                 2.5 %    97.5 %
(Intercept) -10.124126 -9.904199
log(len)      2.762204  2.819879
```

Weight of snapper as a function of length. . .

The fitted line on the log scale

```
> plot(log(wgt)~log(len),data=Snap.df,xlab="log(Length)",ylab="log(Weight)")
> abline(coef(Snap.lm),lty=5, col="red")
```



Weight of snapper as a function of length...

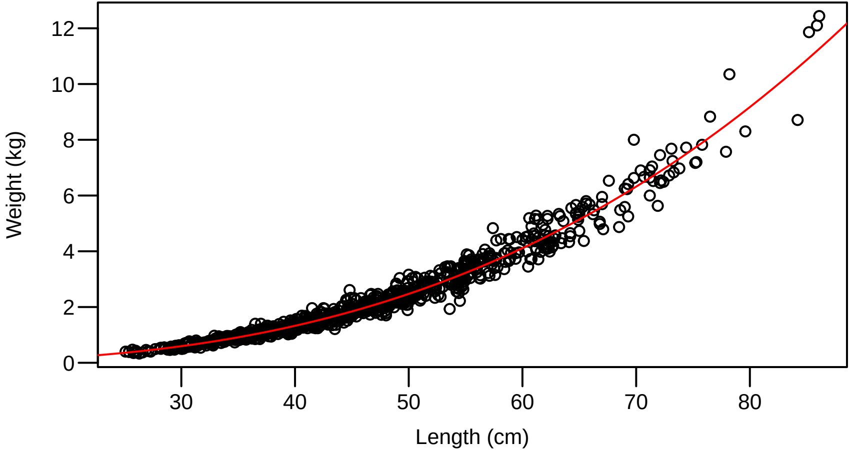
The fitted line on the log scale...

Let us redo the plot on the raw scale (rather than log scale):

```
> plot(wgt~len, data = Snap.df)
> pred.df = data.frame(len = 20:90)
> Snap.pred = exp(predict(Snap.lm, pred.df))
> lines(pred.df$len, Snap.pred, col="red")
```

Weight of snapper as a function of length. . .

The fitted line on the raw scale



Weight of snapper as a function of length. . .

Estimated weight of a 30cm snapper

Recall that we wanted to estimate the weight of 30 cm snapper. Since the linear model is fitted to log(wgt), we must back-transform, and are making inference about median weight.

```
> Pred.df=data.frame(len=30)
> exp(predict(Snap.lm,Pred.df,interval="confidence"))
        fit       lwr       upr
1 0.5937602 0.5857844 0.6018445
```

That is, we estimate 30 cm snapper to have median weight between 586 and 602 grams.

Note: If the research question had asked up to *predict* the weight of a 30 cm snapper then we would use

```
> exp(predict(Snap.lm,Pred.df,interval="prediction"))
        fit       lwr       upr
1 0.5937602 0.4865954 0.7245262
```

We predict a 30 cm snapper to weigh between about 487 and 725 grams.

Weight of snapper as a function of length...

Testing $H_0 : \beta_1 = 3$

A few slides earlier we deduced that the power coefficient β_1 should be close to, though not necessarily equal to 3.

Let us examine this formally by testing the null hypothesis $H_0 : \beta_1 = 3$.

Question 1 Is this hypothesis rejected at the 5% level? (Hint: the answer can be worked out from output already seen)

Question 2 What is the P-value for $H_0 : \beta_1 = 3$? (This takes a bit more work).

What is the P-value?

```
> beta1 = coef(Snap.lm)[2]
> seBeta1 = summary(Snap.lm)$coefficients[2,2]
> hyp = 3
> tstat = (beta1 - hyp)/seBeta1
> tstat
 log(len)
-14.22257
> pval = 2 * (1 - pt(abs(tstat), df = nrow(Snap.df) - 2))
> pval
log(len)
       0
```

The P-value is so small that it been rounded down to zero.

If we really want to see the P-value then we can get the pt function to do the probability calculation on the log scale, and then exponentiate:[1]

```
> logP = log(2)+pt(-abs(tstat), df = nrow(Snap.df) - 2, log.p = T)
> exp(logP)
    log(len)
2.677086e-41
```

We see that the P-value() is in the order of 10^{-41}, i.e., virtually zero.[2]

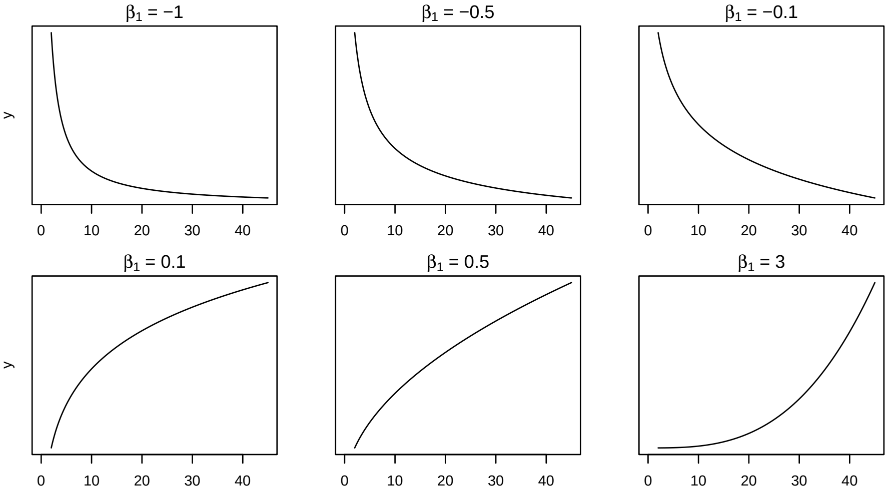
[1]Not examinable

[2]By way of comparison, the earth has about 10^{50} atoms.

Section 7.2
Power law curves and their interpretation

Power law relationships

Other examples of power curves



In general, a power law model could be used whenever one has good reason to believe that the relationship between x and y takes any of these forms.

Power relationships...

Interpretation of power curves

Recall, we are fitting the model $\log(y) = \beta_0 + \beta_1 \log(x)$, which is equivalent to $y = e^{\beta_0} x^{\beta_1} = \alpha x^{\beta_1}$.

For the snapper example the confidence interval for β_1 is

```
> confint(Snap.lm)[2,]
   2.5 %   97.5 %
2.762204 2.819879
```

One way to interpret this is to say that a 1% increase in the $x = len$ value results in a 2.76% to a 2.82% increase in the median value of $y = wgt$.

Explanation: Increasing x by 1% is the same as multiplying x by 1.01. So the relative change in y is:

$$\Delta y = \frac{\alpha(1.01x)^{\beta_1}}{\alpha x^{\beta_1}} = \frac{\alpha x^{\beta_1}(1.01)^{\beta_1}}{\alpha x^{\beta_1}} = 1.01^{\beta_1}$$

$1.01^{\beta_1} \approx 1 + \beta_1 \times .01$ for reasonable values of β_1 (Taylor's series). So, a 1% increase in x results in an approximate relative increase in y of $\Delta y \approx \beta_1 \times .01$ or an increase of β_1%

Power relationships...

Interpretation of power curves...

Alternatively, it might be more meaningful to quantify the change in the median of y arising from a 50% increase in x, or perhaps a doubling of x, say 50% increase in x: Then y changes from αx^{β_1} to

$$\alpha(1.5x)^{\beta_1} = \alpha x^{\beta_1} 1.5^{\beta_1}$$

i.e., the median of y gets multiplied by 1.5^{β_1}.

Doubling in x: Then y changes from αx^{β_1} to

$$\alpha(2x)^{\beta_1} = \alpha x^{\beta_1} 2^{\beta_1}$$

i.e., the median of y gets multiplied by 2^{β_1}.

Power relationships. . .

Interpretation of power curves. . .

```
> 1.5^confint(Snap.lm)[2,]
   2.5 %   97.5 %
3.064785 3.137300
```

That is, increasing length by 50% corresponds to an increase in median snapper weight between 206% and 214%[3]. [4]

```
> 2^confint(Snap.lm)[2,]
   2.5 %   97.5 %
6.784319 7.061030
```

That is, doubling length corresponds to an increase in median weight between 578% and 606%.

---

[3]These percentages are given by `100*(1.5^confint(Snap.lm)[2,]-1)`.

[4]One could also say that median snapper weight is between 2.06 and 2.14 times higher. Be careful **not** to say that it is between 3.06 and 3.14 times higher (since it actually multiplies by between 3.06 and 3.14). It is probably safest to talk about % change.

Section 7.3
Relevant R-code

Most of the R-code you need for this chapter

When your response variable is right skew and you have a good reason to believe the underlying relationship follows a power relationship then try taking logs of both y and x.

```
> Snap.lm=lm(log(wgt)~log(len),data=Snap.df)
```

We state the effect as the % change in the median of y for a given % change in x.

The confidence interval for β_1 is

```
> confint(Snap.lm)[2,]
```

and is the (approximate) % change in the median y for a 1% increase in x.

In general, for a z% increase in x, the multiplier for the median of y is

```
> (1+z/100)^confint(Snap.lm)[2,]
```

or alternatively, the percentage change in the median of y is

```
> 100*((1+z/100)^confint(Snap.lm)[2,]-1)
```