lab 5: Chick Weight dataset visualisation

The aim of this lab is to understand **split-apply-combine pattern** and apply the pattern to `ChickWeight dataset`.

```
#install.packages("datasets")
library(datasets)
#data()
```

Now look at ChickWeight dataset. This gives the data from the experiment described in the lecture on "Data Wrangling".

The `ChickWeight data` frame has 578 rows and 4 columns. It has 4 variables:

- weight - a numeric value giving the body weight of the chick (gm).
- Time - the number of days since birth when the measurement was made.
- Chick - a unique identifier for the chick.
- Diet - indicating which experimental diet the chick received.

```
head(ChickWeight)
```

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

```
str(ChickWeight)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   578 obs. of  4 variables:
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Diet
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time"
##   ..$ y: chr "Body weight"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(gm)"
```

```
names(ChickWeight)
```

```
## [1] "weight" "Time"   "Chick"  "Diet"
```

It's annoying that some of the names are capitalized and some are not. We can fix this and have everything lower case:

```
chickweight <- ChickWeight
names(chickweight) <- tolower(names(chickweight))
#?tolower
str(chickweight)
```

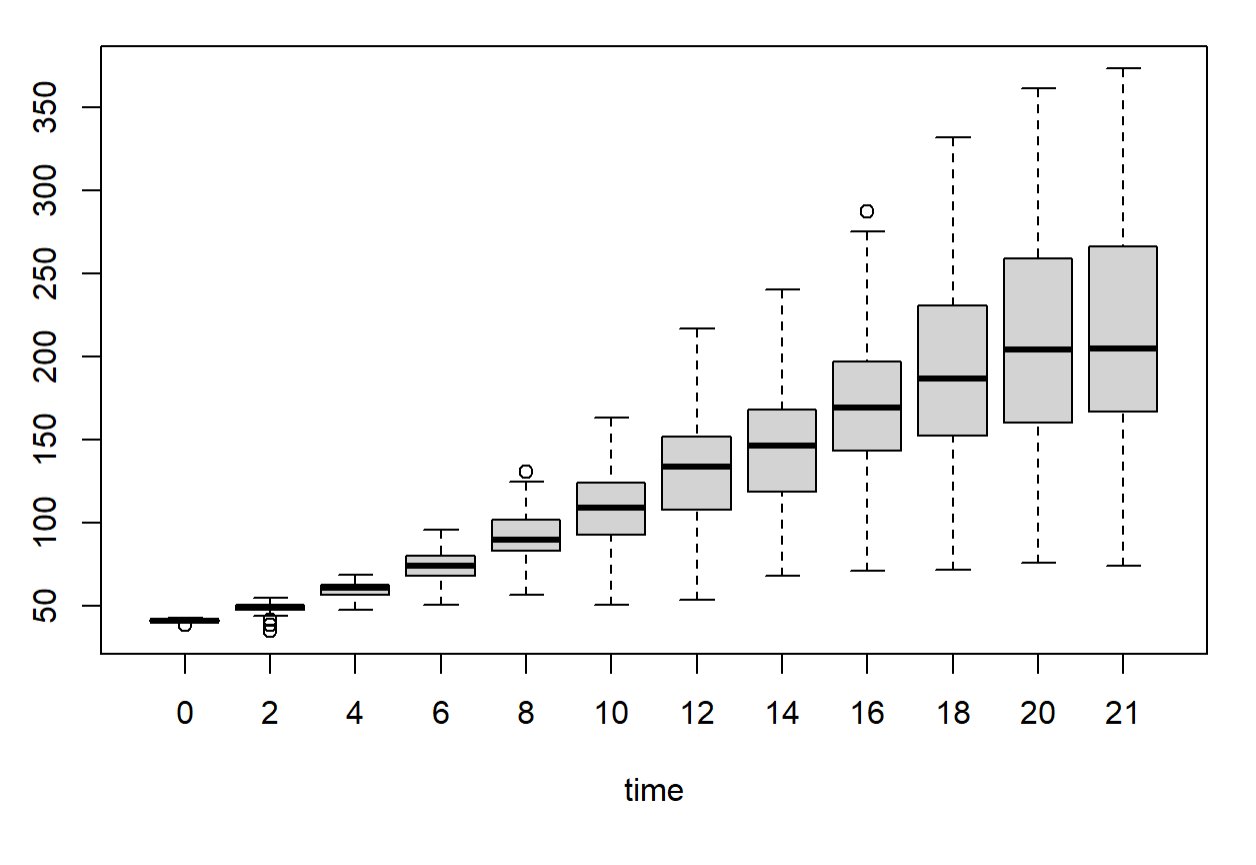
```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   578 obs. of  4 variables:
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Diet
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time"
##   ..$ y: chr "Body weight"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(gm)"
```

```
names(chickweight)
```

```
## [1] "weight" "time"   "chick"  "diet"
```

Good! A graphical overview of the dataset can be done:

```
boxplot(weight~time, data = chickweight,
        xlab="time", ylab="weight")
```




```
#?boxplot
```

Subset data

If we only want the weights of each chick on day 21, we can subset the data:

```
cw21 <- subset(chickweight, time == 21)
summary(cw21)
```

```
##      weight           time        chick    diet
##  Min.   : 74.0   Min.   :21   13     : 1   1:16
##  1st Qu.:167.0   1st Qu.:21   9      : 1   2:10
##  Median :205.0   Median :21   20     : 1   3:10
##  Mean   :218.7   Mean   :21   10     : 1   4: 9
##  3rd Qu.:266.0   3rd Qu.:21   17     : 1
##  Max.   :373.0   Max.   :21   19     : 1
##                               (Other):39
```

```
#?summary
#?subset
#is.na(chickweight)
#?is.na
#anyNA(chickweight, recursive = FALSE)
```

To make `cw21` we used conditional subsetting (time == 21). Available comparison operators are:

- < less than
- (> greater than)
- == equal to
- <= less than or equal to
- (>= greater than or equal to)
- != NOT equal to (! symbol indicates negation)
- is.na(x) tests if x has missing values

Logical operators to combine expressions are also available:

- & logical AND
- (| logical OR)
- ! logical NOT (negation)

Sometimes, we want to subset based on multiple variables,

for example, we canfind how many chicks on diet 1 have weights over 250 on day 21:

```
cw21.subset2 <- cw21[(cw21$diet == "1" & cw21$weight >= 250), ]
cw21.subset2
```

```
##     weight time chick diet
## 84     305   21     7    1
## 167    266   21    14    1
```

Using plyr package

Now let's use `plyr`:

```
#?plyr
#install.packages("plyr")
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.0.5
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

Let's plot the weight of each chick as a function of time.

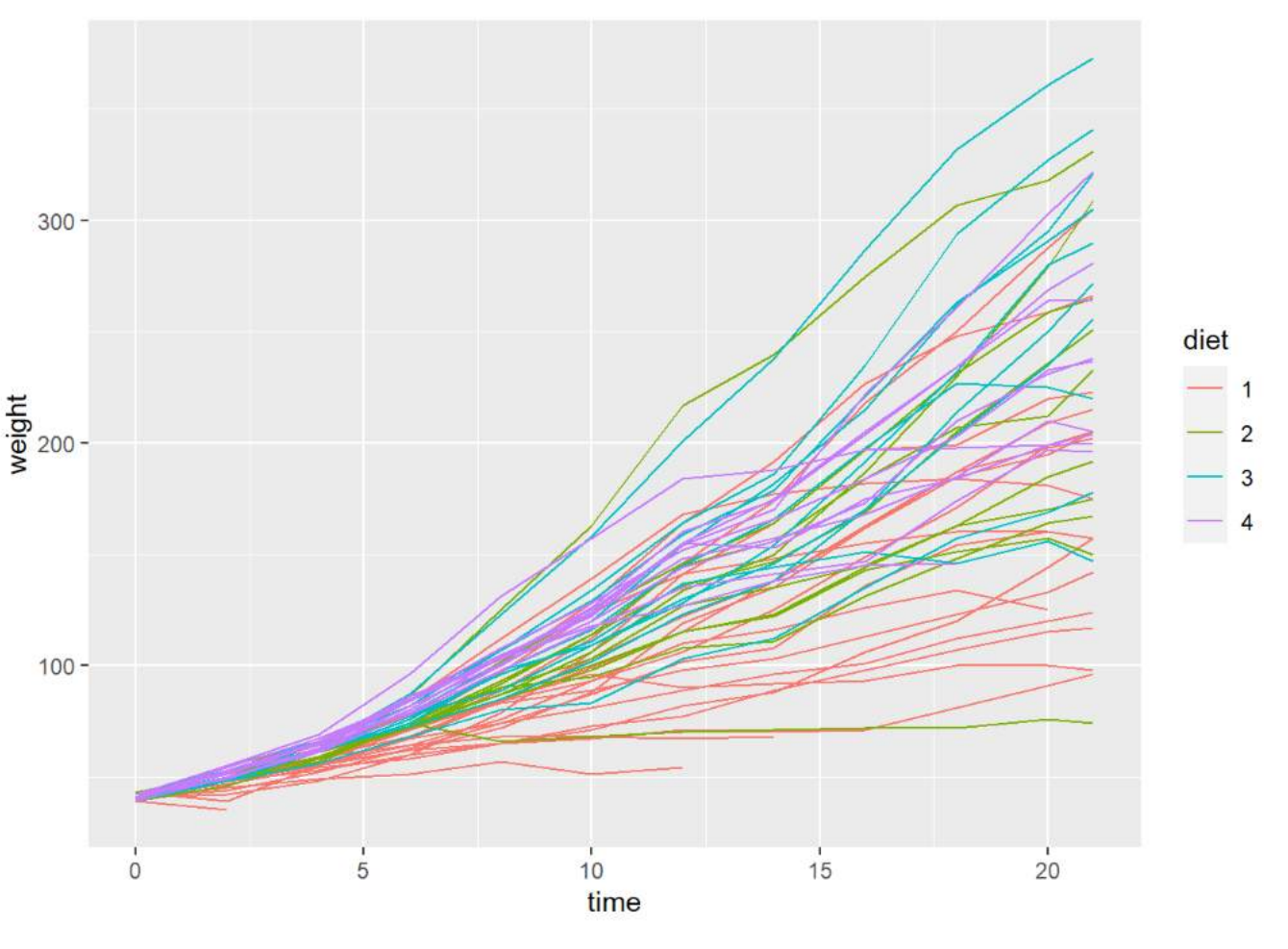
This requires a little thought.

- x will be time;
- y will be weight.

But we want a separate line for each chick - but it might be best if the lines are coloured according to diet.

There are lots of chicks so we don't want a different colour for each chick - there will be too many, and the whole point is to distinguish the effects of the diets.

```
p <- ggplot(chickweight,aes(x=time,y=weight,
                            colour=diet,
                            group=chick)) + geom_line()
p
```




```
#?ggplot
```

What do you see?

Let's summarise to mean weight for each diet on each day.

```
chicksummary1 <- ddply( chickweight, .(diet,time), summarise,
                        meanweight=mean(weight) )
str(chicksummary1)
```

```
## 'data.frame':    48 obs. of  3 variables:
##  $ diet      : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ time      : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ meanweight: num  41.4 47.2 56.5 66.8 79.7 ...
```

```
names( chicksummary1)
```

```
## [1] "diet"       "time"       "meanweight"
```

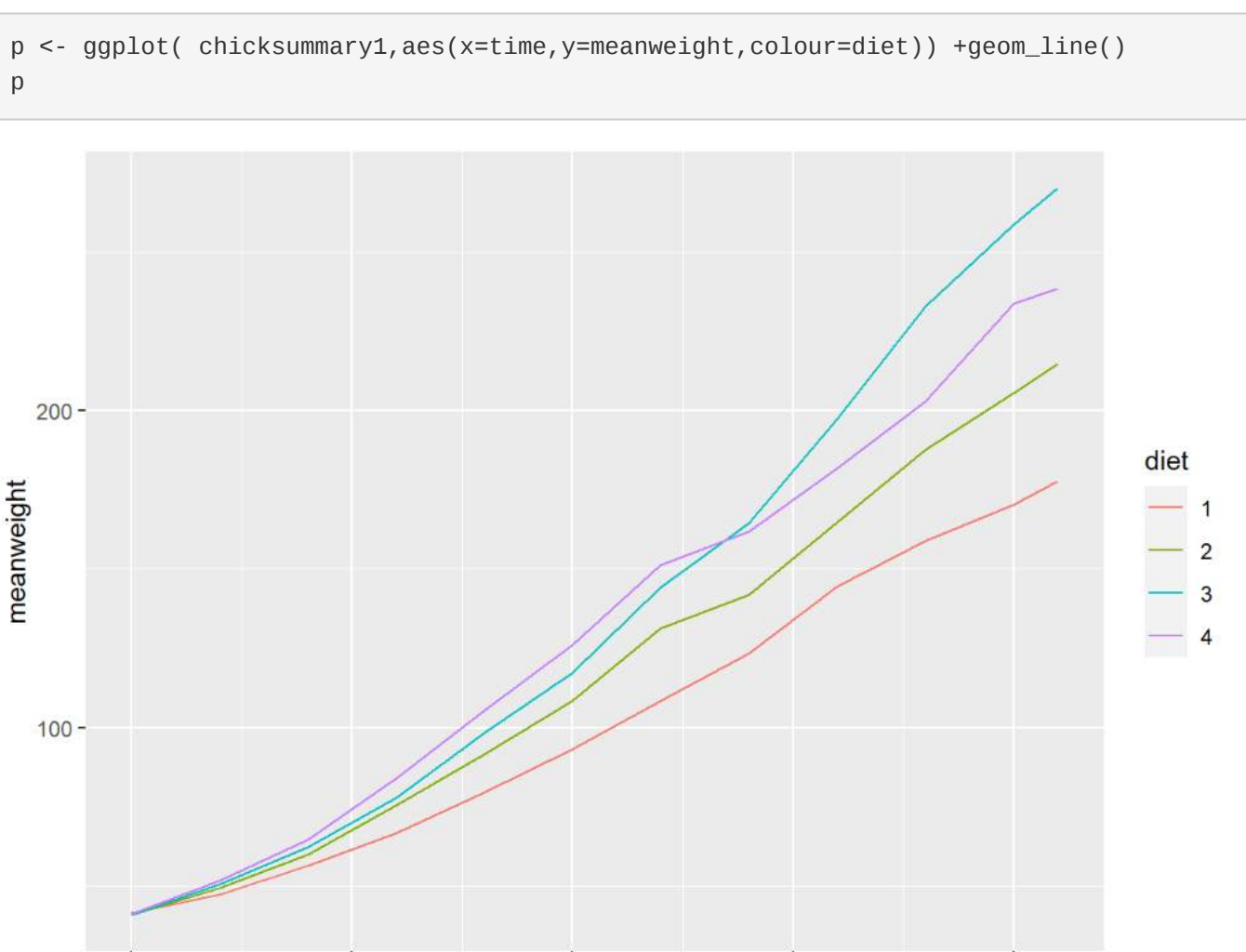
```
chicksummary1
```

```
##    diet time meanweight
## 1     1    0   41.40000
## 2     1    2   47.25000
## 3     1    4   56.47368
## 4     1    6   66.78947
## 5     1    8   79.68421
## 6     1   10   93.05263
## 7     1   12  108.52632
## 8     1   14  123.38889
## 9     1   16  144.64706
## 10    1   18  158.94118
## 11    1   20  170.41176
## 12    1   21  177.75000
## 13    2    0   40.70000
## 14    2    2   49.40000
## 15    2    4   59.80000
## 16    2    6   75.40000
## 17    2    8   91.70000
## 18    2   10  108.50000
## 19    2   12  131.30000
## 20    2   14  141.90000
## 21    2   16  164.70000
## 22    2   18  187.70000
## 23    2   20  205.60000
## 24    2   21  214.70000
## 25    3    0   40.80000
## 26    3    2   50.40000
## 27    3    4   62.20000
## 28    3    6   77.90000
## 29    3    8   98.40000
## 30    3   10  117.10000
## 31    3   12  144.40000
## 32    3   14  164.50000
## 33    3   16  197.40000
## 34    3   18  233.10000
## 35    3   20  258.90000
## 36    3   21  270.30000
## 37    4    0   41.00000
## 38    4    2   51.80000
## 39    4    4   64.50000
## 40    4    6   83.90000
## 41    4    8  105.60000
## 42    4   10  126.00000
## 43    4   12  151.40000
## 44    4   14  161.80000
## 45    4   16  182.00000
## 46    4   18  202.90000
## 47    4   20  233.88889
## 48    4   21  238.55556
```

```
#?ddply
```

Now let's plot it:

```
p <- ggplot( chicksummary1,aes(x=time,y=meanweight,colour=diet)) +geom_line()
p
```

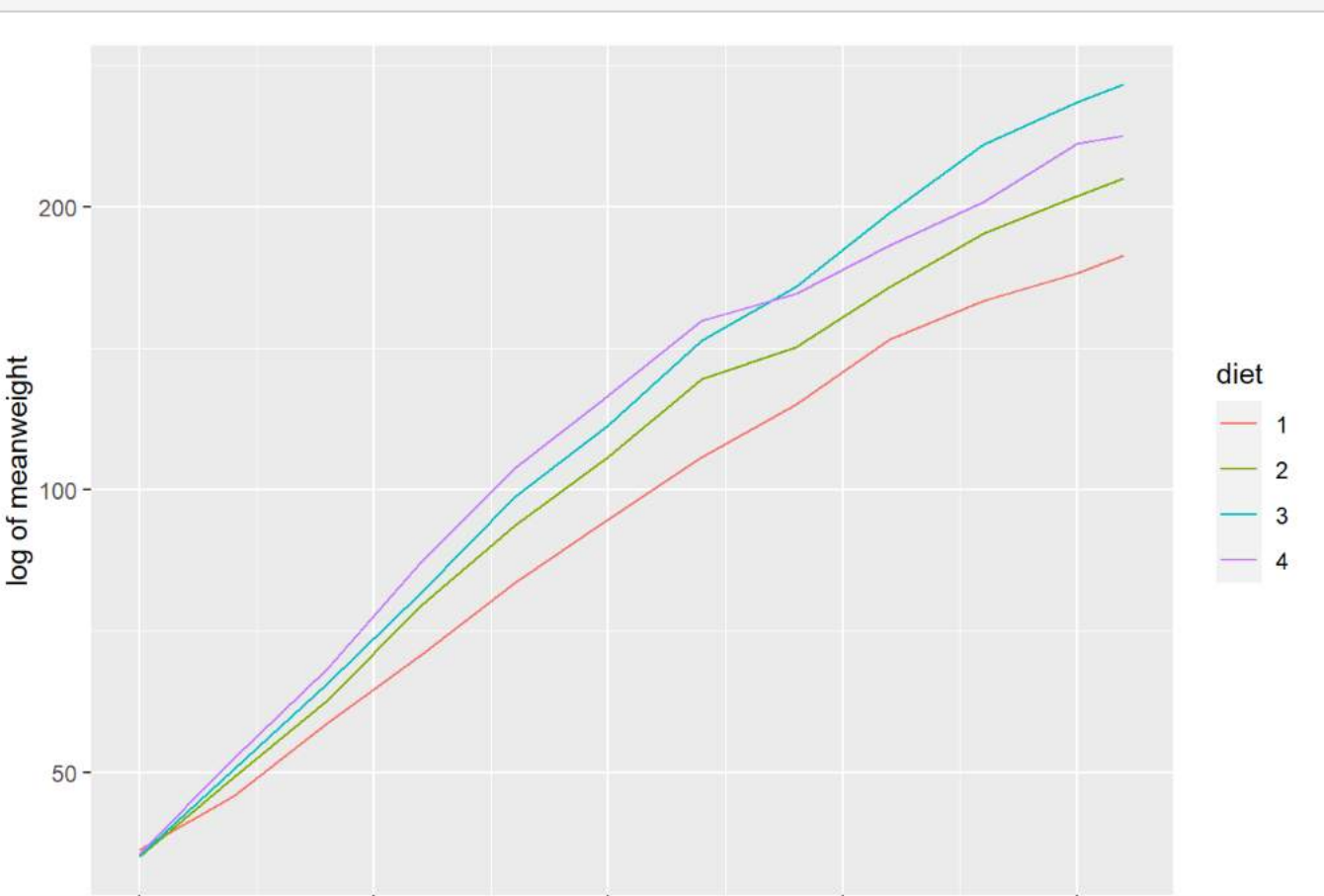


What do you see?

Try using a log scale for the y axis

- what do you see?
- Is the chick growth exponential for part of the time?
- Is this reasonable?

```
p + scale_y_log10()+ylab("log of meanweight")
```



Challenges:

1. Use `ddply` to produce other summaries.

- How many chicks were fed each diet?
- How many chicks, fed each diet, died early? Which day did they die?

How many chicks were fed each diet?

```
chicksummary2a <- ddply(chickweight, .(diet), summarise,
                        numberofchick=max(chick))
chicksummary2a
```

```
##   diet numberofchick
## 1    1             7
## 2    2            21
## 3    3            35
## 4    4            48
```

How many chicks, fed each diet, died early? Which day did they die?

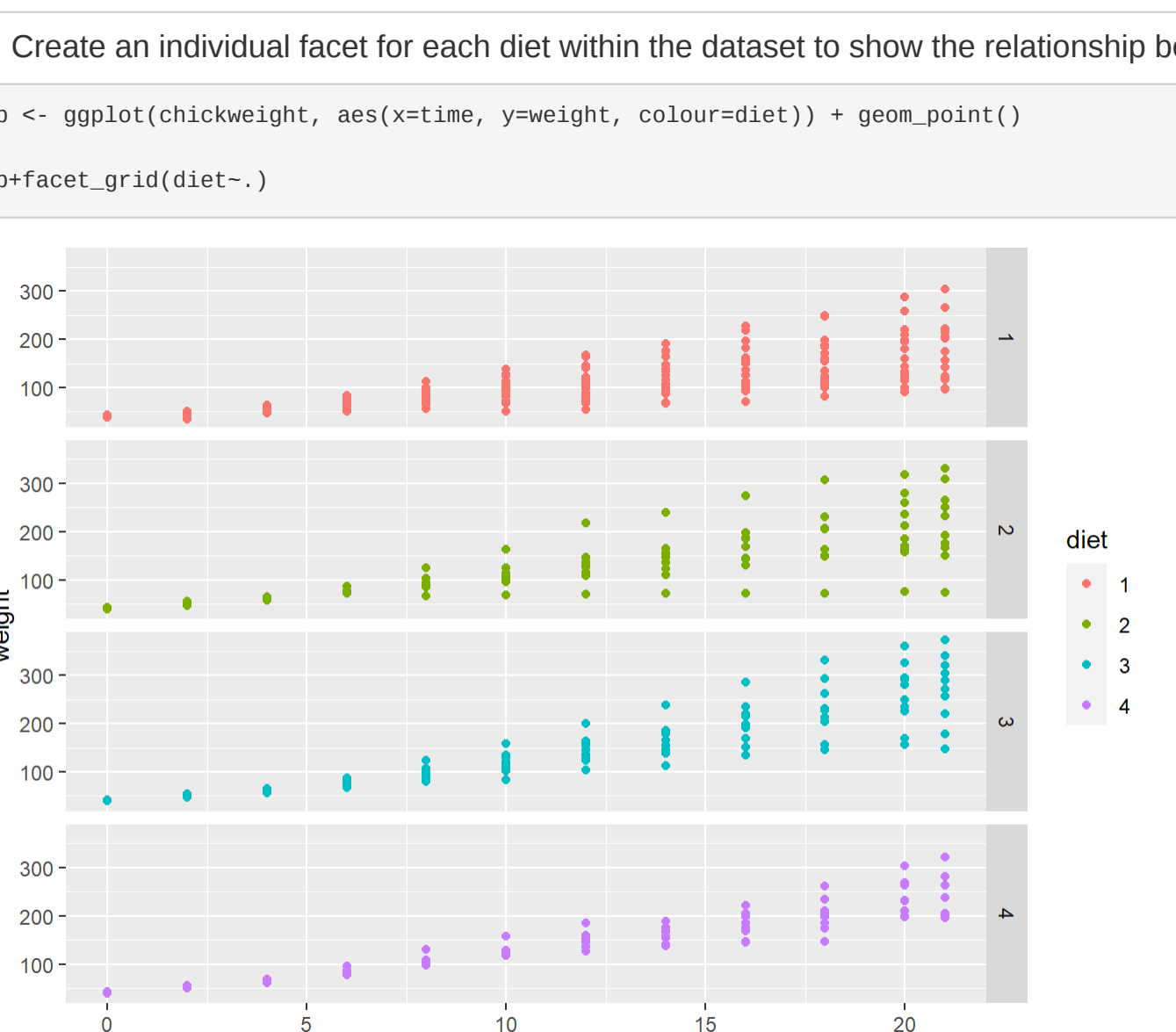
```
chicksummary1b <- ddply( chickweight, .(diet), summarise,
            lastday=max(time),numberofchick=max(chick))
chicksummary1b
```

```
##   diet lastday numberofchick
## 1    1      21             7
## 2    2      21            21
## 3    3      21            35
## 4    4      21            48
```

2. Create an individual facet for each diet within the dataset to show the relationship between times and weights.

```
p <- ggplot(chickweight, aes(x=time, y=weight, colour=diet)) + geom_point()

p+facet_grid(diet~.)
```



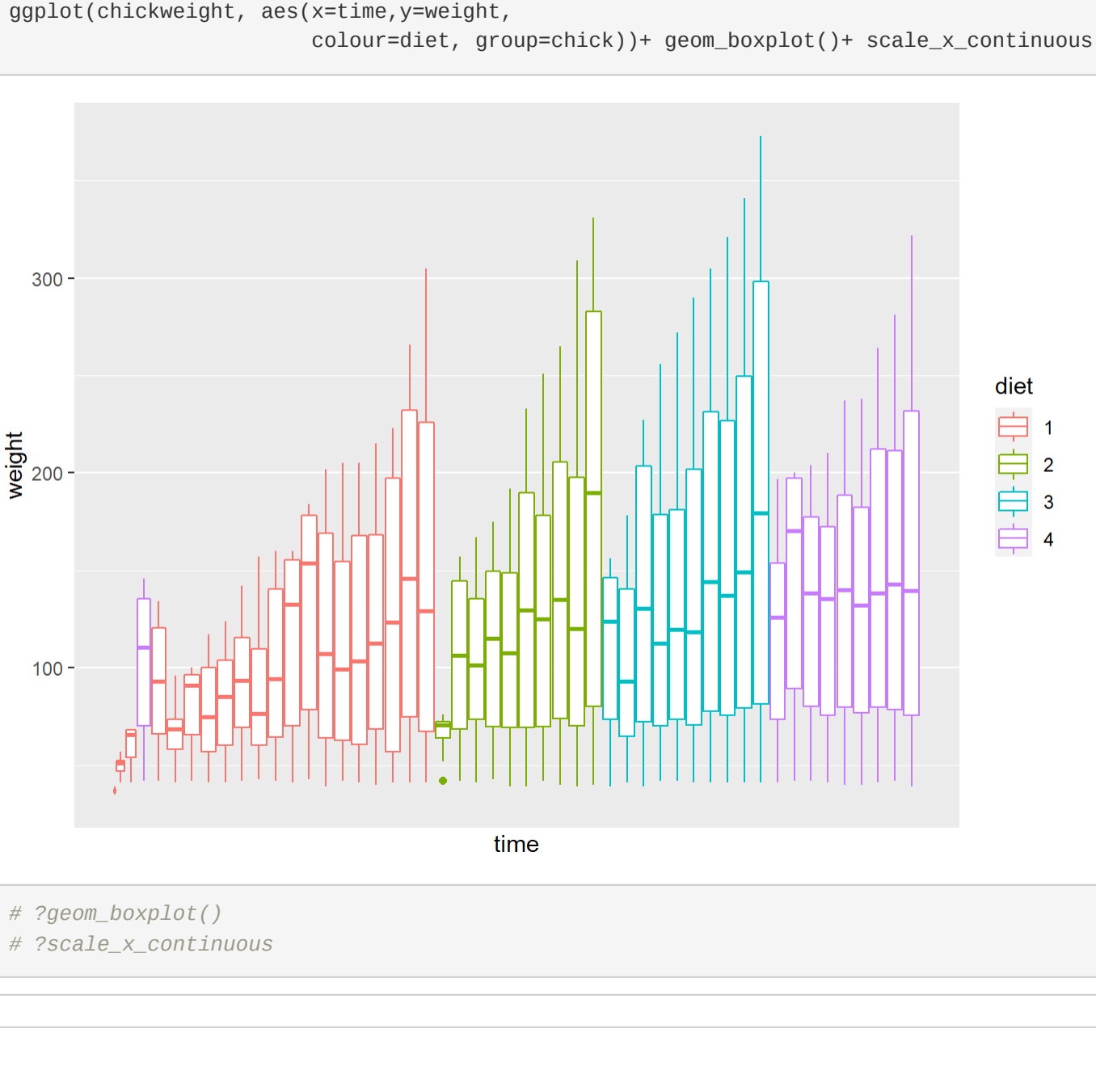

```
# ?geom_point()
# ?facet_grid
```

3. Do the line plots we have done show that one diet is better than another?

(Hint: they don't ! Not quite.) What would be a better plot to compare how heavy chicks raised on different diets became?

(Hint: boxplots. Look up notched boxplots on page 133 in the "R Graphics Cookbook" by Winston Chang.)

```
ggplot(chickweight, aes(x=time,y=weight,
                        colour=diet, group=chick))+ geom_boxplot()+ scale_x_continuous(breaks = NULL)
```




```
# ?geom_boxplot()
# ?scale_x_continuous
```