UNSW SYDNEY

CENTRE FOR BIG DATA RESEARCH IN HEALTH

HDAT9800 Visualisation and Communication of Health Data

Chapter 3 - Basic data visualisation in R, with emphasis on ggplot2 - part B

Drs James Farrow & Tim Churches

ggplot2

ggplot2 implements a layered grammar of graphics (gg=grammar of graphics)

- based on the work of Leland Wilkinson, building on the earlier work of Bertin
- **data**: variables mapped to aesthetic features of the graph.
- **geoms**: objects/shapes on the graph.
- **stats**: statistical transformations to summarise data, *e.g*. counts, mean, confidence intervals
- **scales**: mappings of aesthetics to data values; legends and axes display these
- **coordinate systems**: the plane on which data are mapped on the graphic.
- **faceting**: split the data into subsets to create multiple panels on the same graph

Data

Data for plotting with *ggplot2* tools must be stored in a *data.frame*

Objects of class *matrix* cannot be used and will need to be converted to *data.frame* before plotting

Multiple data frames may be used in one plot, which is an advantage over other graphing systems

The `ggplot()` function

All graphics begin with specifying the `ggplot()` function

Note that although the name of the package is *ggplot2*, the name of the function is `ggplot()`

In the `ggplot()` function we specify the ‘default’ dataset and map variables to aesthetics (visible aspects) of the graph

The first layer for any *ggplot2* graph is this *aesthetics* layer

We then add subsequent layers (of *geoms*, *stats*, *&c.*), producing the actual graphical elements of the plot

Aesthetics: `aes()`

Aesthetics are the visually perceivable components of the graph.

We map variables from the data frame to aesthetics using `aes()`

For example, we can map:

- which variables appear on the x-axis and y-axis.
- a classification variable to colours
- a numeric variable to the size of graphical objects such as line width or data point size

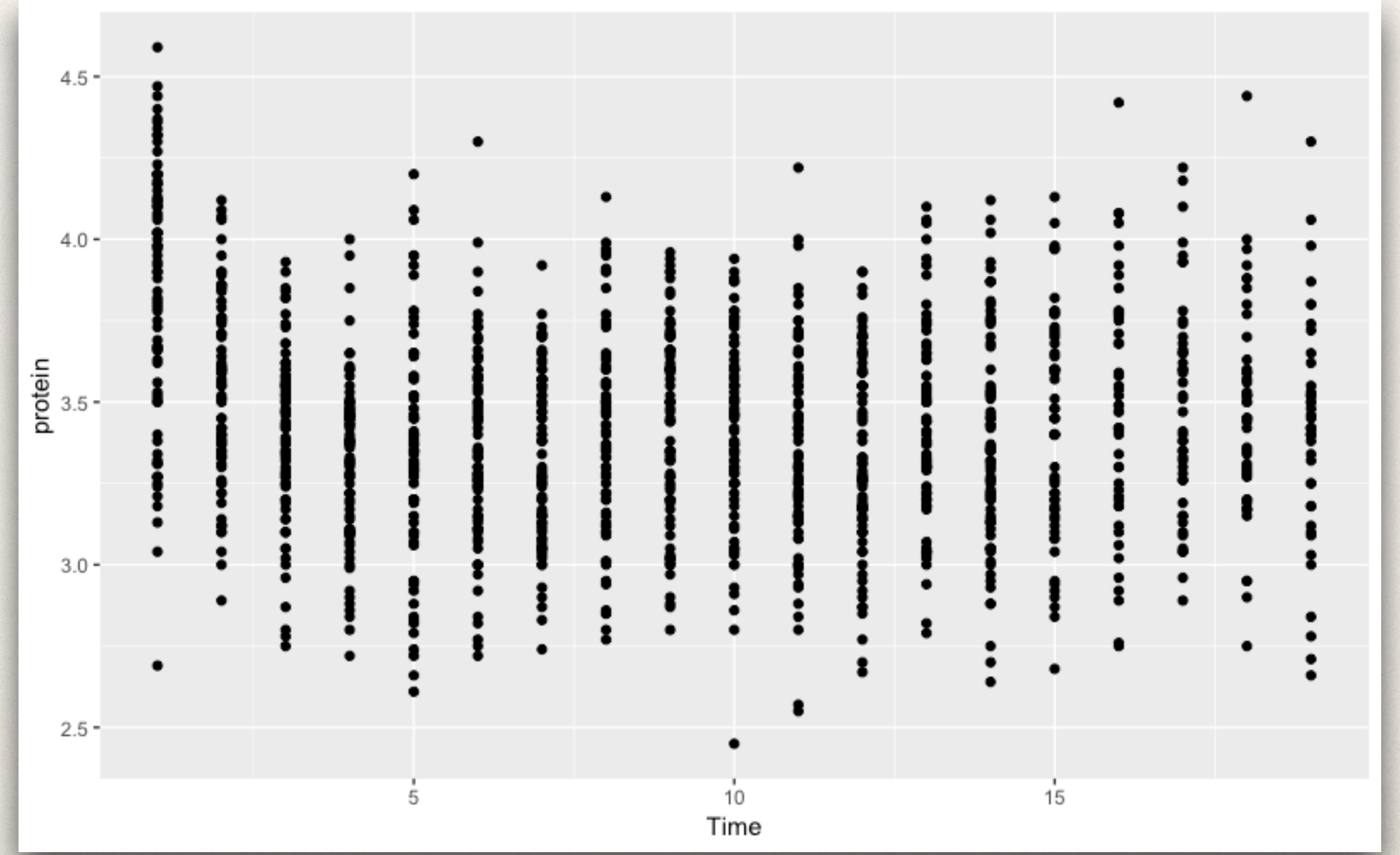
Our first plot

```
ggplot(data = Milk, aes(x=Time, y=protein)) +
   geom_point()
```

This creates a plot

- using the data in the data frame `Milk`
- mapping `Time` to the x axis
- mapping `protein` to the y axis
- visualising the data using a *scatterplot* using `geom_point()`

Our first plot



Some example aesthetics

Which aesthetics are required and which are allowed depends on the particular *geom* being used.

- x, y: positioning along x-axis and y-axis respectively
- colour: colour of objects; for 2-d objects, the colour of the object's outline (*cf.* fill)
- fill: fill colour of objects
- alpha: transparency of objects, a value between 0 (transparent) and 1 (opaque)
- linetype: how lines should be drawn (solid, dashed, dotted, *&c.*)
- shape: shape of markers in scatter plots
- size: how large objects appear

Geoms

The *geom* elements differ in the geometric shapes produced for the plot:

- geom_bar(): bars with bases on the x-axis
- geom_boxplot(): boxes-and-whiskers
- geom_errorbar(): T-shaped error bars
- geom_histogram(): histogram
- geom_line(): lines
- geom_point(): points (scatterplot)
- geom_ribbon(): bands spanning y-values across a range of x-values
- geom_smooth(): smoothed conditional means

Layers

Graphs in *ggplot2* are built layer-by-layer, rather than being pre-made

As we have already seen, we add more layers with the + operator.

A layer consists of graphics produced by either a *geom* or *stat* element

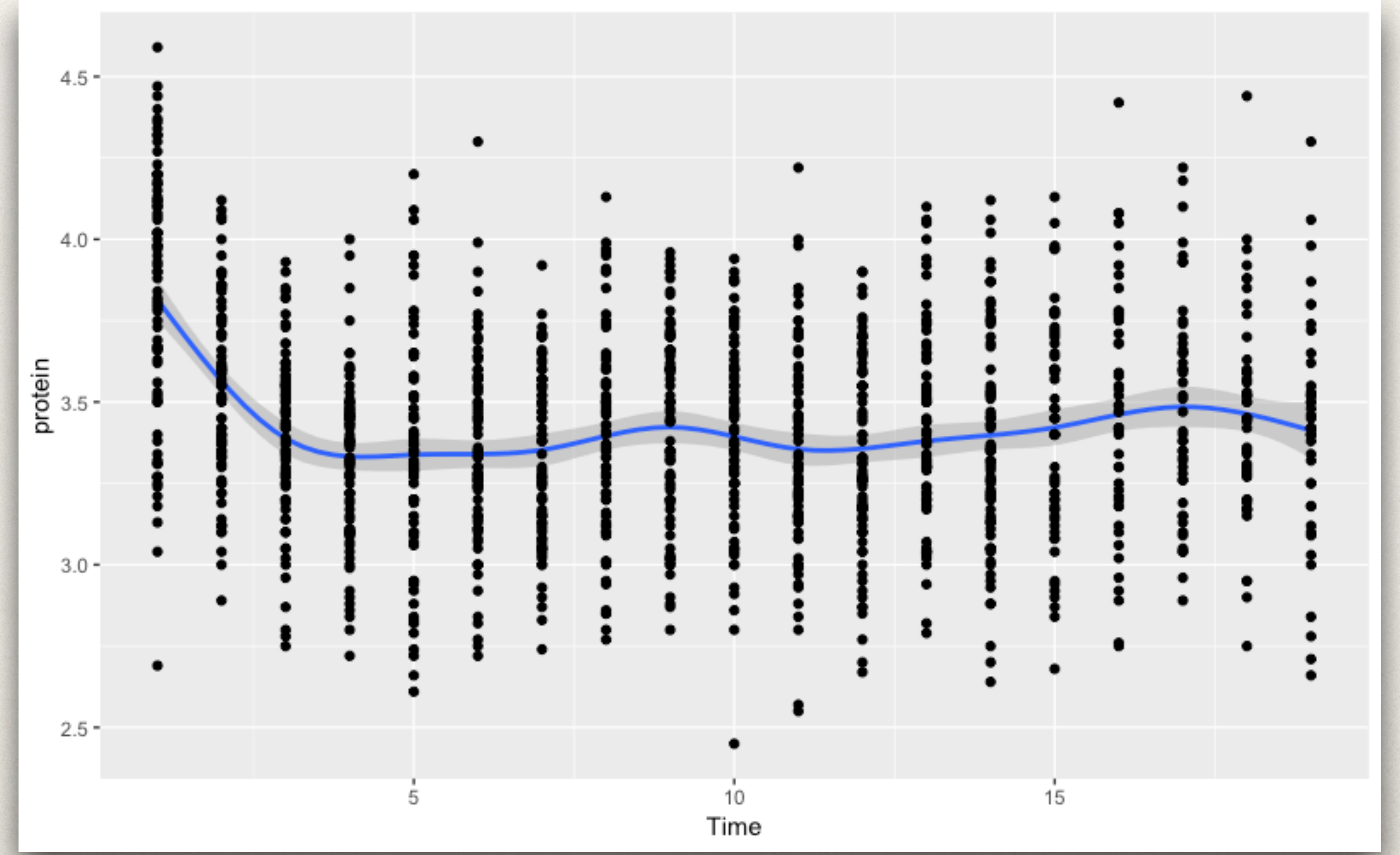
We can add layers in a virtually unrestricted way, allowing us to customise our graphs as we see fit

Adding layers

```
ggplot(data = Milk, aes(x=Time, y=protein)) +
  geom_smooth() +
  geom_point()
```

Plot as before but this time with a line showing the smoothed mean

Adding layers



Stats

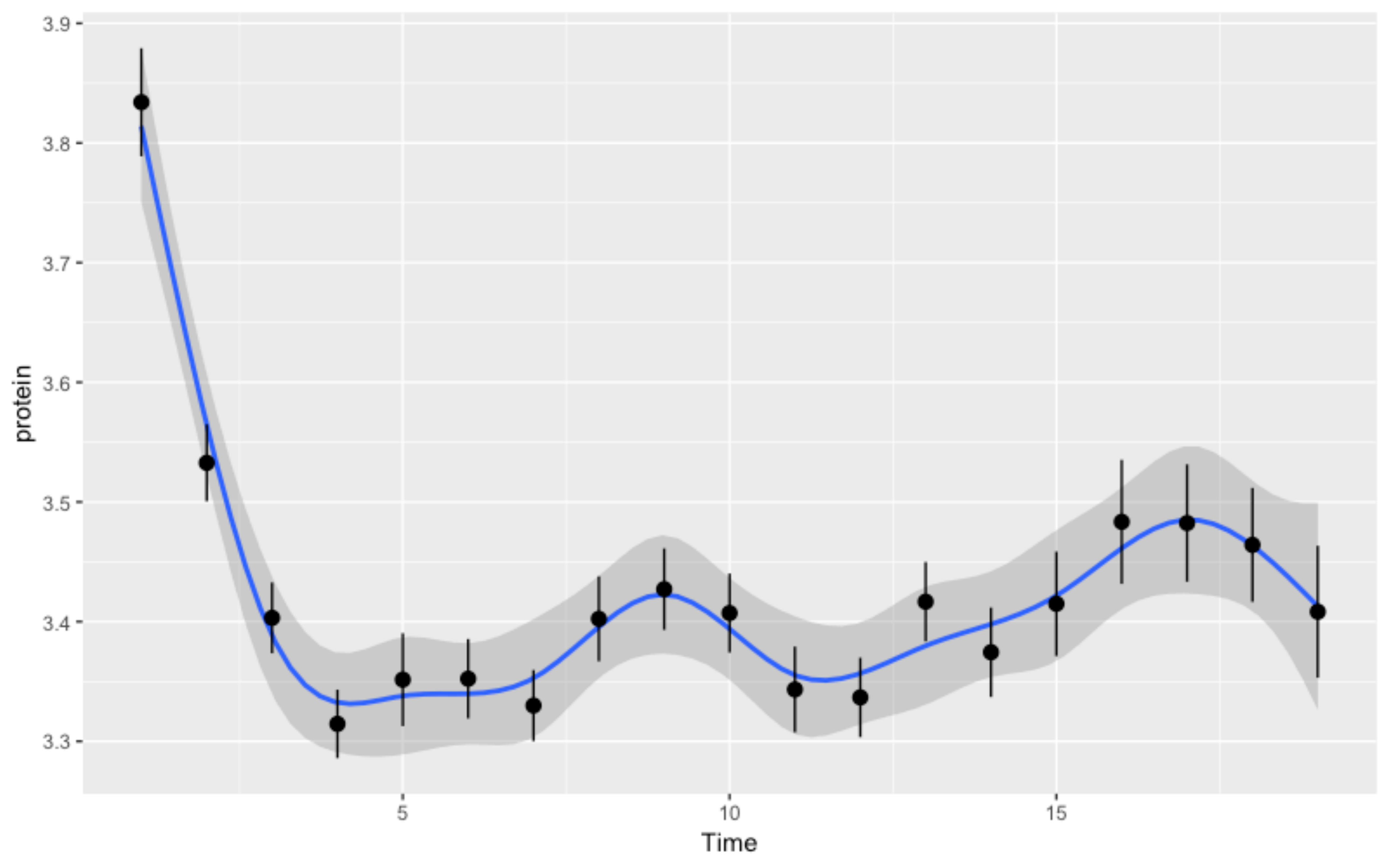
The *stat* functions statistically transform data, usually as some form of summary

- frequencies of values of a variable (histogram, bar graphs)
- a mean
- a confidence limit

Each stat function is associated with a default *geom*

```
ggplot(data = Milk, aes(x=Time, y=protein)) +
  geom_smooth() + stat_summary()
```

Stats



Scales

Scales define how aesthetic values are mapped to data values.

The *ggplot2* package usually allows the user to control the *scales* for each aesthetic.

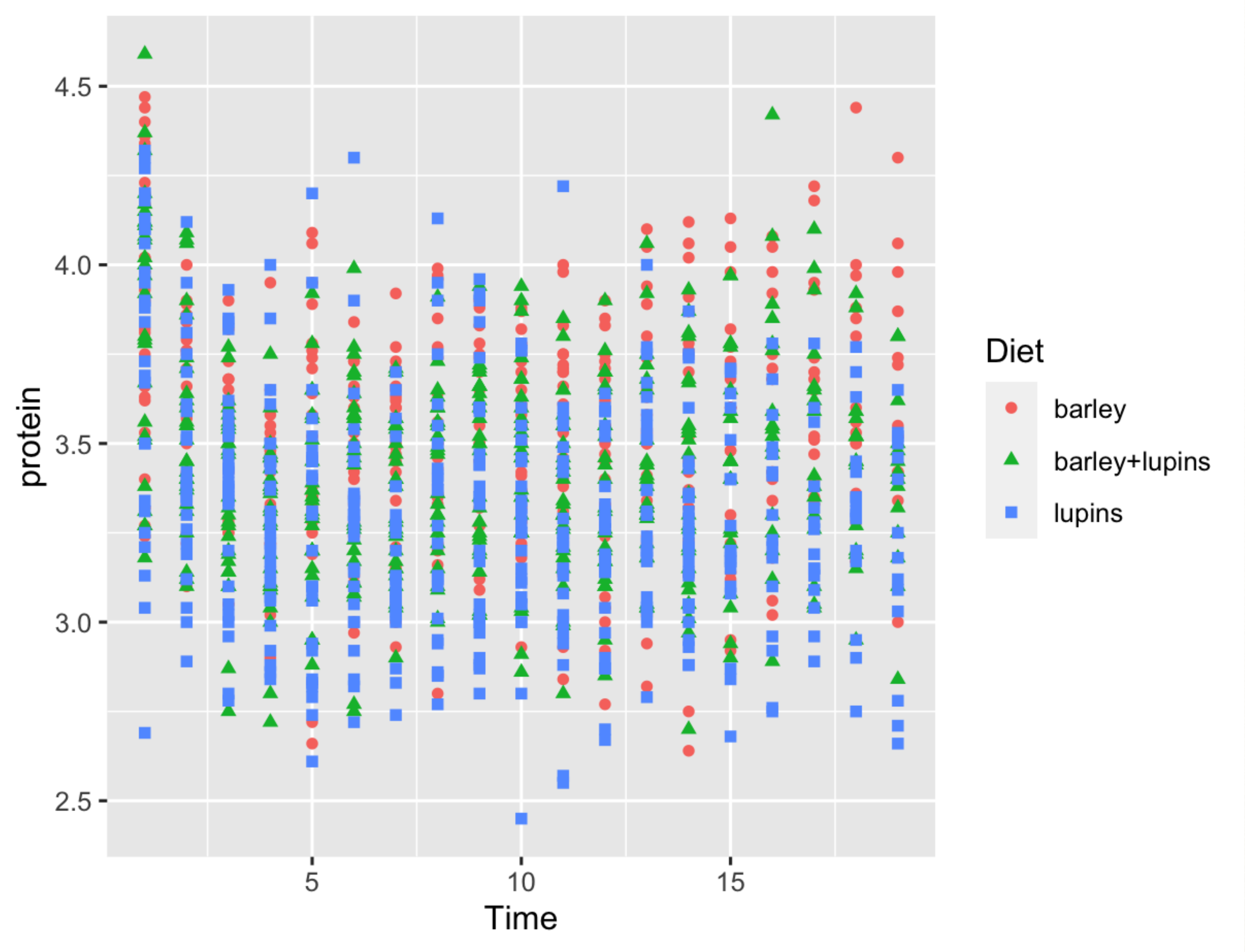
Compare:

```
ggplot(Milk, aes(x=Time, y=protein,
                 shape=Diet, color=Diet)) +
  geom_point()
```

vs

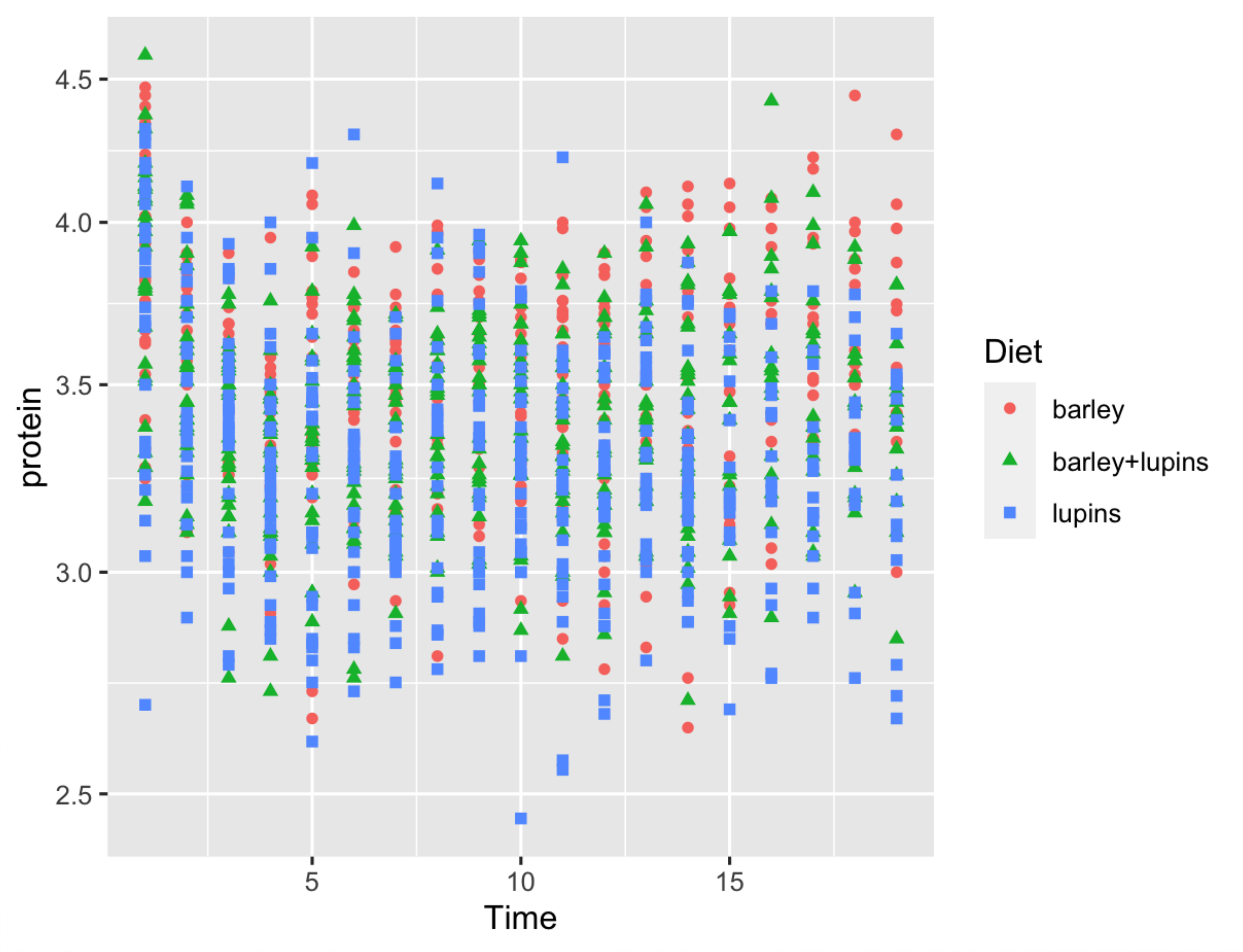
```
ggplot(Milk, aes(x=Time, y=protein,
                 shape=Diet, color=Diet)) +
  geom_point() + scale)y_log10()
```

Scales



4.5
4.0
3.5
3.0
2.5
protein
5
10
15
Time
Diet
● barley
▲ barley+lupins
■ lupins

Scales


4.5
4.0
3.5
3.0
2.5
protein
5
10
15
Time
Diet
● barley
▲ barley+lupins
■ lupins

Guides

Guides (axes and legends) visualise a *scale*, displaying data values and their matching aesthetic values

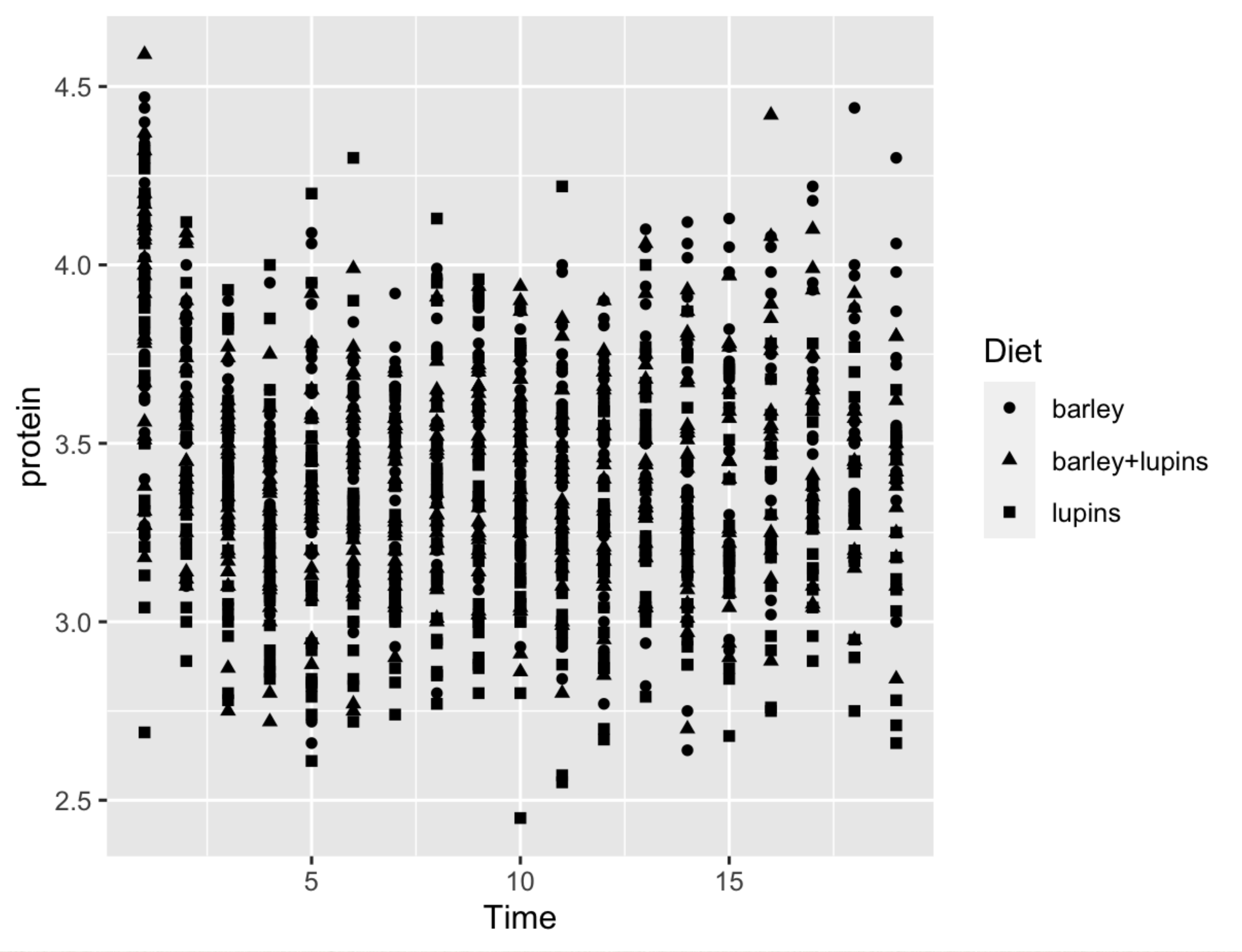
The x-axis, a *guide*, visualises the mapping of data values to position along the x-axis

A colour scale *guide* (legend) displays colour mappings to data values

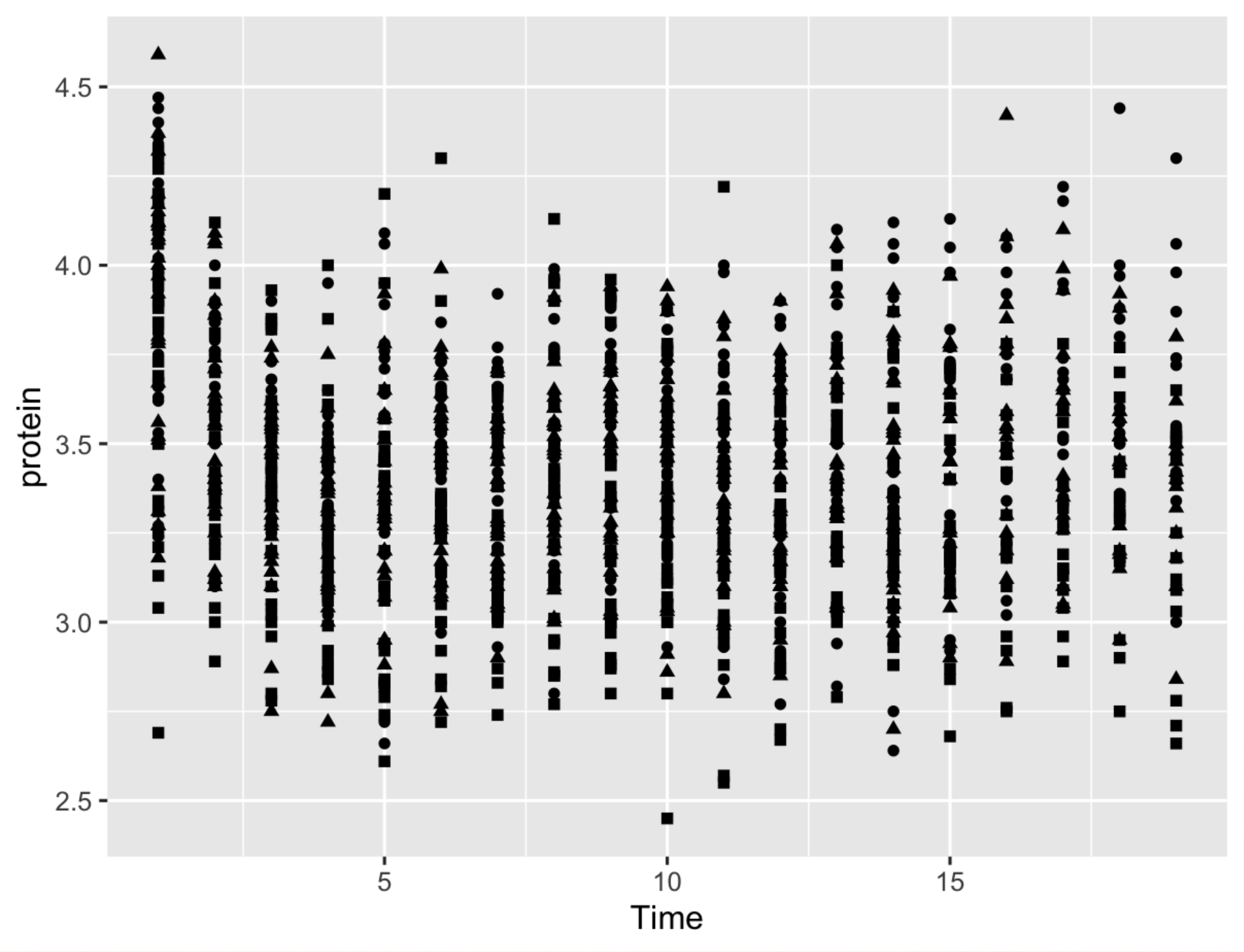
The `guides()` function sets and removes *guides* for each *scale*

```
ggplot(Milk, aes(x=Time, y=protein, shape=Diet)) +
  geom_point() + guides(shape="none")
```

```
ggplot(Milk, aes(x=Time, y=protein, shape=Diet)) +
  geom_point()
```



```
ggplot(Milk, aes(x=Time, y=protein, shape=Diet)) +
  geom_point() + guides(shape="none")
```



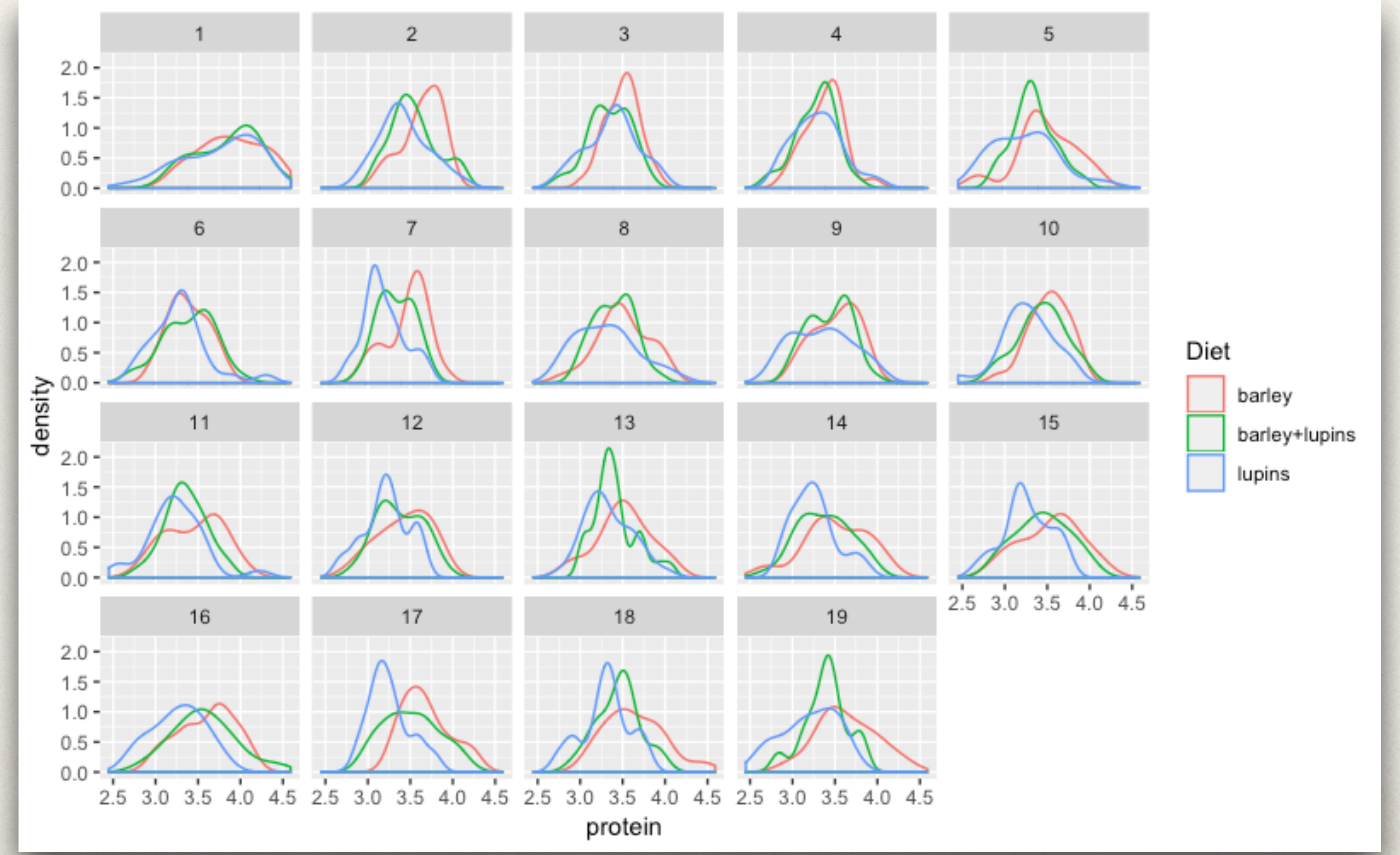
Facetting

We can split plots into small multiples (panels) with the faceting functions `facet_wrap()` and `facet_grid()`

The resulting graph shows how each plot varies along the faceting variable(s)

```
ggplot(Milk, aes(x=protein, color=Diet)) +
  geom_density() +
  facet_wrap(. ~ Time)
```

Facetting



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
density
protein
2.0
1.5
1.0
0.5
0.0
2.5
3.0
3.5
4.0
4.5
Diet
barley
barley+lupins
lupins

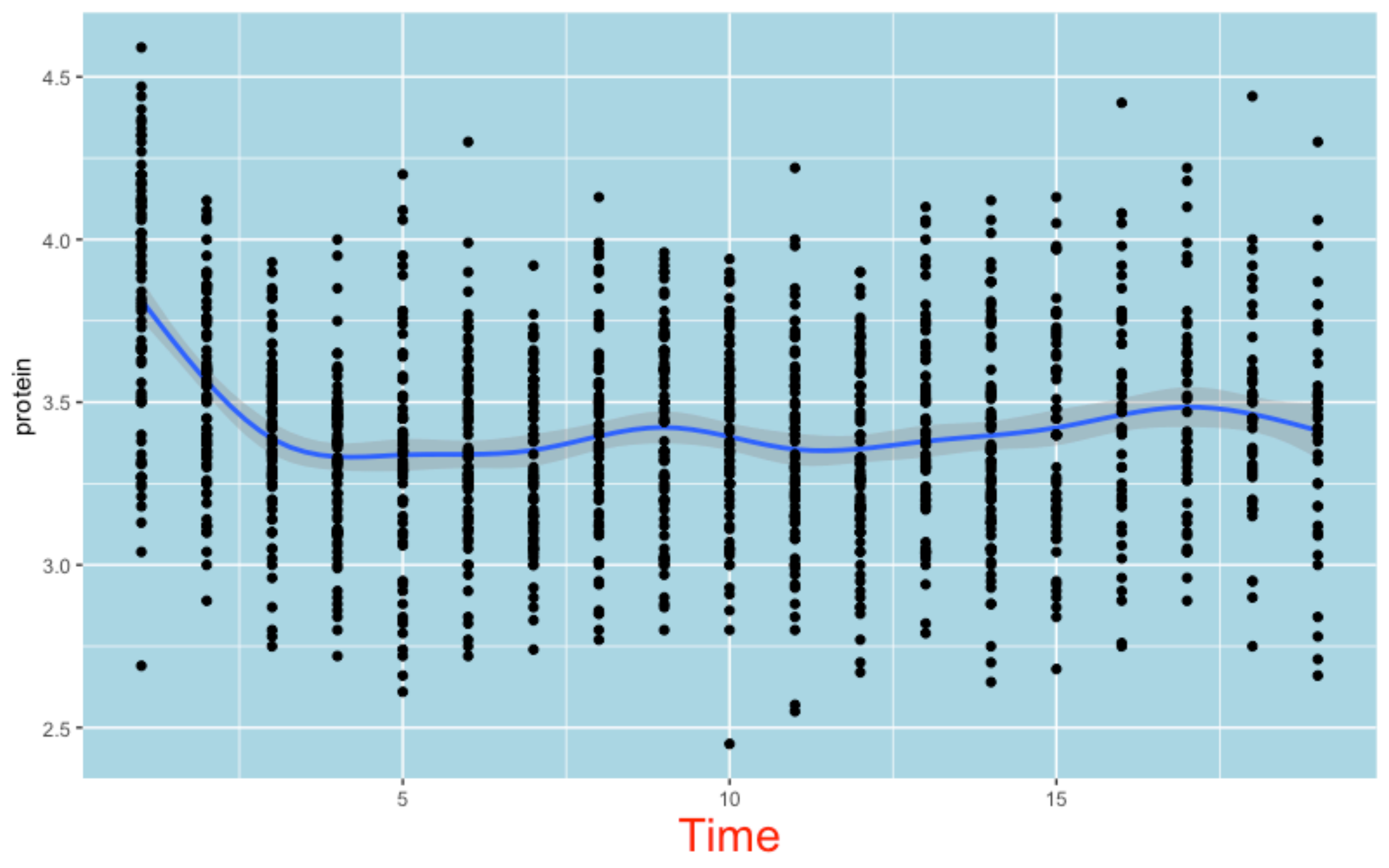
Themes

Themes control elements of the graph not related to the data

- the background colour
- the size of fonts
- gridlines
- the colour of labels

```
ggplot(Milk, aes(x=Time, y=protein)) +
  geom_smooth() + geom_point() +
  theme(panel.background=element_rect(fill=“lightblue”)) +
  theme(axis.title.x=element_text(size=20, color=“red"))
```

Themes



Chapter 3 Assessment

Will be announced through the course site

A simple *ggplot2* task as a markdown document with embedded plots