R Course,
AufLand Conference

Eduard Szöcs

26. November 2015

Institute for Environmental Sciences - University of Koblenz-Landau

Intro

Short Intro: Eduard Szöcs

- PhD student @Quantitative Landscape Ecology
- Environmental Sciences & Ecotoxicology
- Research:
  - Statistical Ecology - Eco(toxico)logical Statistics
  - Effects and distribution of pesticides in freshwaters
- R-Programming:
  - R-user for 6years
  - Author/Co-Author of 3 CRAN packages (taxize, webchem, rspear)
  - Other packages on github (restax, esmisc)
  - Minor contribitions to other pkgs (e.g. vegan)

`edild.github.io` @EduardSzoecs

Short Intro: Gunnar Oehmichen

- Diploma in Environmental Sciences
- Diploma Thesis @Quantitative Landscape Ecology
- Topics: Ecology, Renewable Energies and Conservation
- Graduate Assistant @Environmental Sciences Lab: Renewable Energies
  - Automated display of Spatial Information on Renewable Energies in a WebGIS
- R-Programming:
  - R-User for 5 Years
  - Assistant in several Summerschools (@UFZ-Leipzig, @Landau)

Course Outline

- Short intro & course organisation, Software preparation
- An introduction to ggplot2.
  - Basic ggplot2 usage
  - Customization of your plots
  - Free time for specific Q.
- After 4 hours of ggplot2 you should be an expert ;)

Course material: `https://github.com/EDiLD/r_landau_2015`

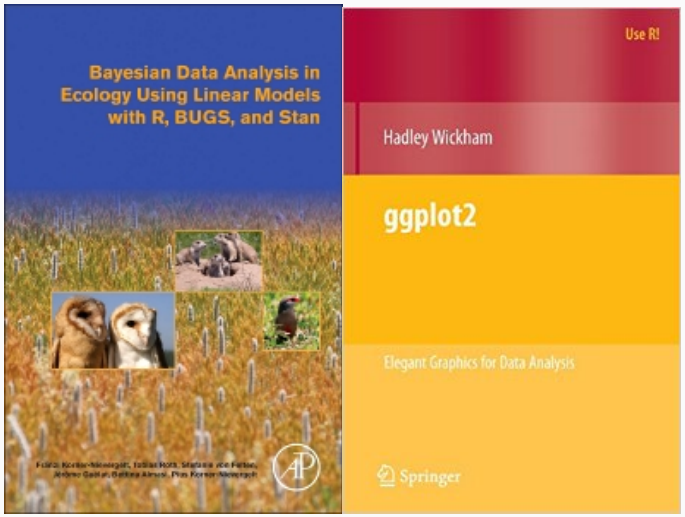
Preliminaries

- Download the course repository.
- No formal R knowledge required to follow.
- (Install RStudio and packages needed).
- Just open the '.R' files in RStudio and execute the script line by line: 'CTRL + ENTER'

Datasets

We will use 2 data sets in this course:

1. Frog monitoring in Swiss (Slides).
2. Diamond prices (Exercises).



Tip:

Don't buy the old ggplot2 book - a new one is coming soon...

```
library(blmeco)
data(frogs)
head(frogs)
```

```
##   count1 count2 elevation year fish waterarea vegetation pondid      x
## 1     16     12       380 2013    0      2500          1 400301 649750
## 2      0      0       565 2009    0       300          1 400411 647350
## 3      0      0       430 2012    0       450          1 400603 650250
## 4      0      0       500 2012    0       348          1 400608 649400
## 5      0      0       450 2012    0       200          1 400701 646700
## 6      0      0       560 2010    0        42          1 400802 646500
##        y
## 1 248850
## 2 255750
## 3 244600
## 4 243850
## 5 240750
## 6 253650
```

```
frogs$fish <- factor(frogs$fish)
frogs$vegetation <- factor(frogs$vegetation)
```

Questions:

What influences frog abundance? Can we predict frog abundance (ala SDM)?

```
library(ggplot2)
data(diamonds)
head(diamonds)
```

```
##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

```
?diamonds
```

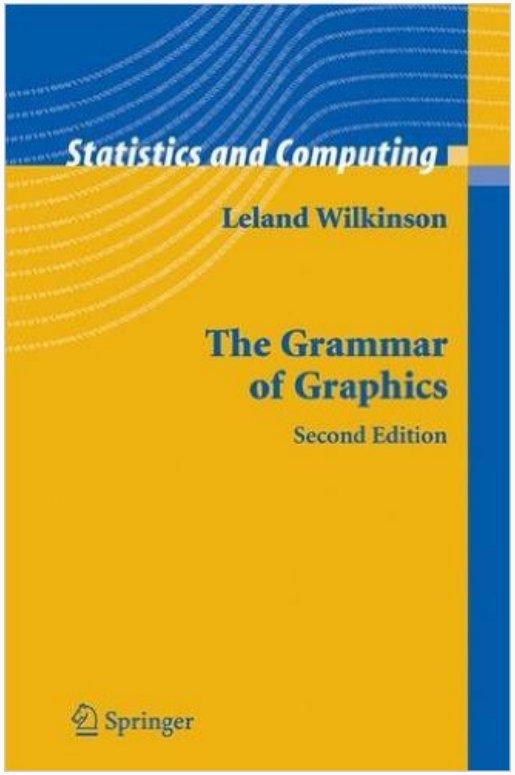
Questions:

What factors are crucial for the diamond prices?

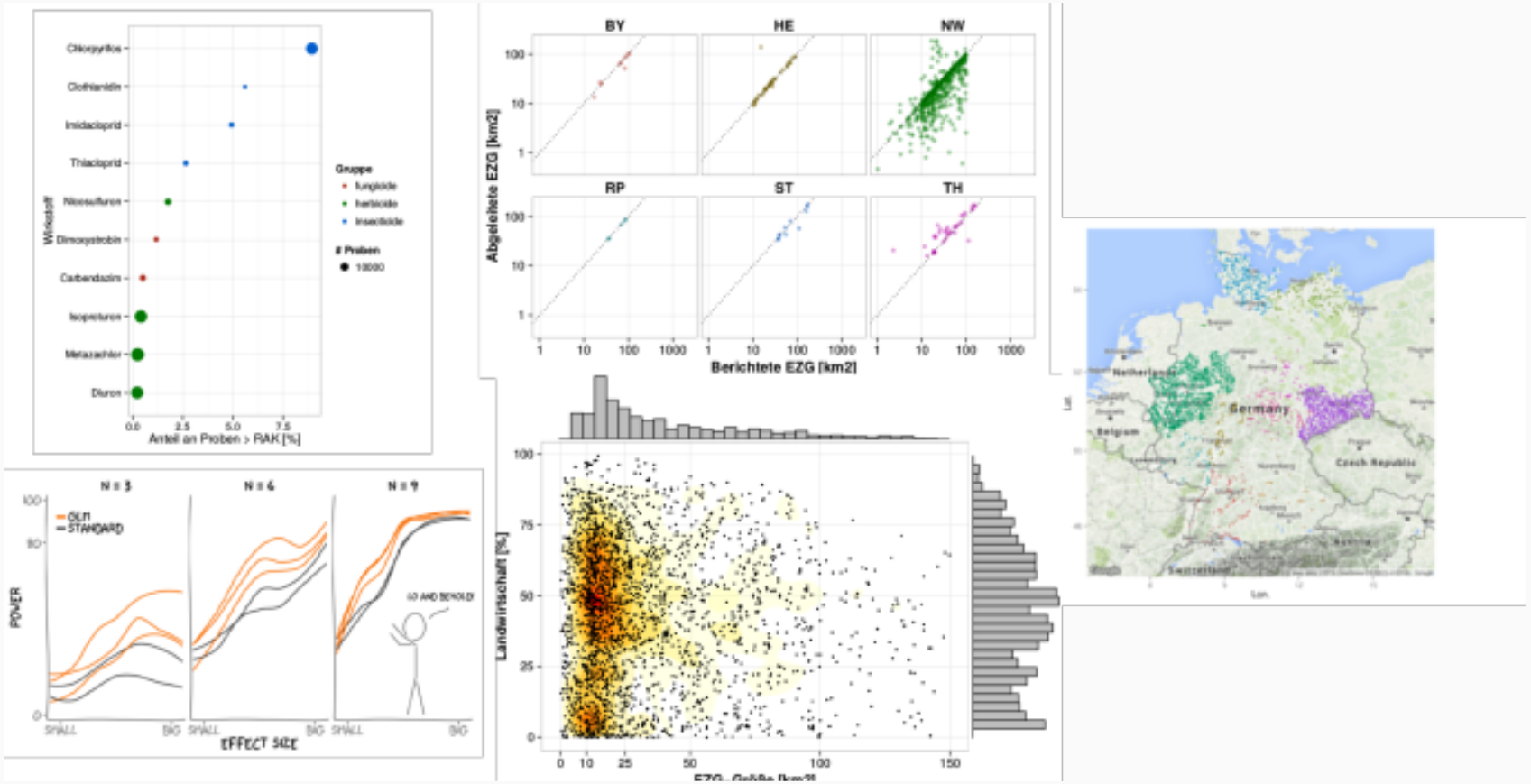
GGPLOT2

What is ggplot2?

- A graphic system for R
- gg = Grammar of Graphics (Wilkinson 2005)
- *Grammar*: A set of components that define a sentence
- ggplot defines a grammar to create plots
- Consistent, intuitive, easy to learn

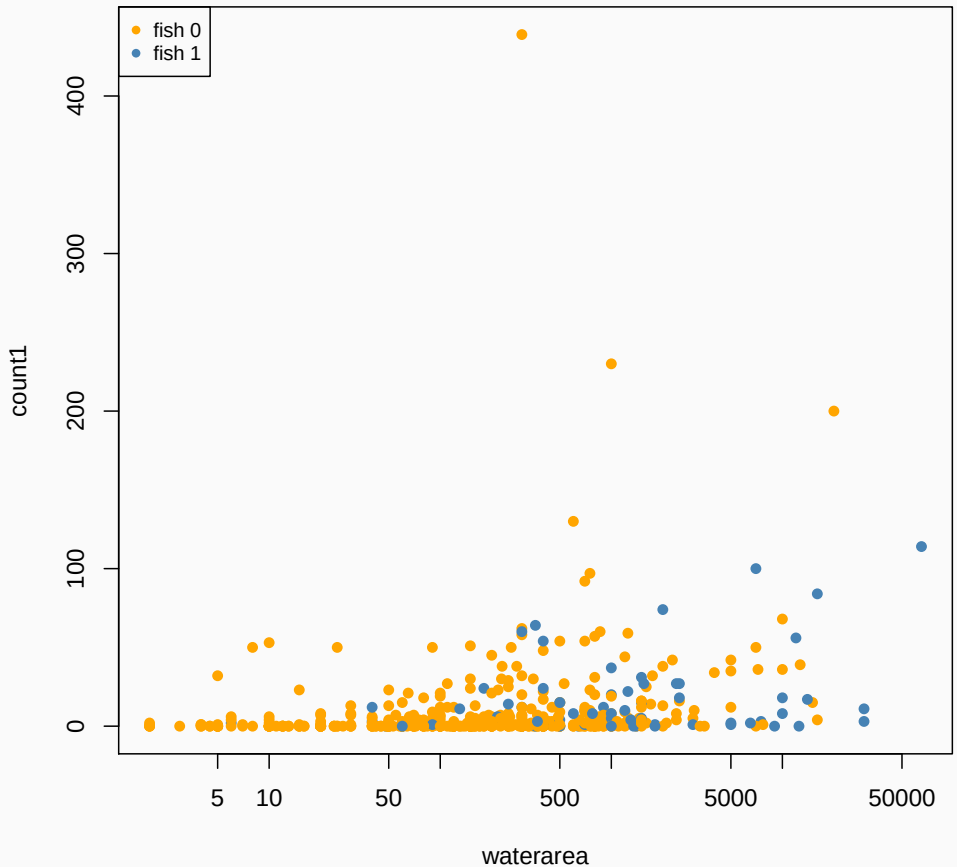


Why ggplot2?


Chlorpyrifos
Clothianidin
Imidacloprid
Thiacloprid
Nicosulfuron
Dimoxystrobin
Carbendazim
Isoproturon
Metazachlor
Diuron
Wirkstoff
Anteil an Proben > PAK [%]
Gruppe
fungicide
herbicide
insecticide
Proben
10000
BY
HE
NW
RP
ST
TH
Abgeleitete EZG [km2]
Berichtete EZG [km2]
Landwirtschaft [%]
N = 3
N = 6
N = 9
OLD
STANDARD
POWER
EFFECT SIZE
SMALL
BIG
GO AND BEHOLD
Germany
Netherlands
Belgium
Czech Republic
Switzerland

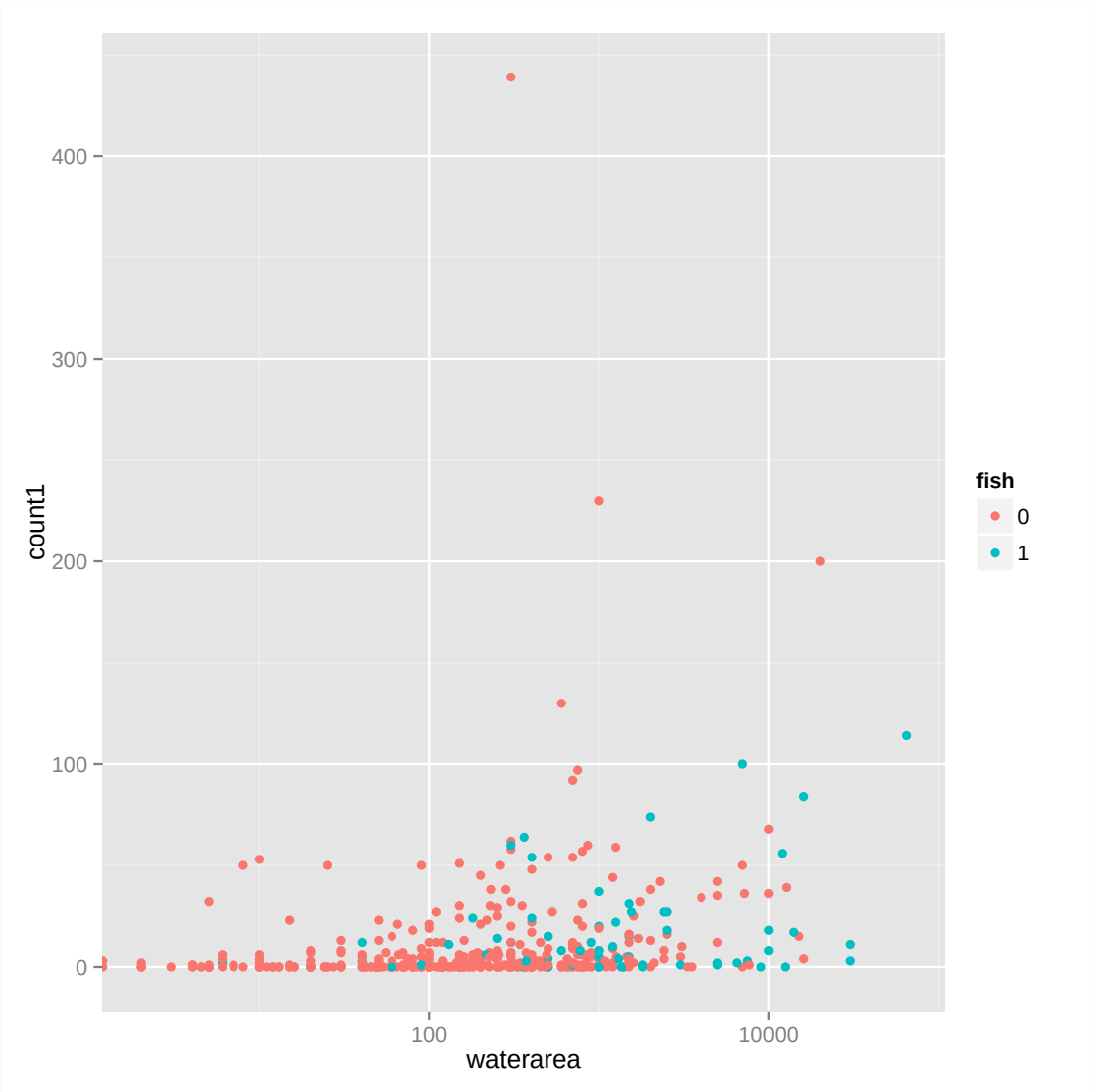
With base R

```
cols <- c('orange', 'steelblue')
plot(count1 ~ waterarea, data = frogs, type = 'n', log = 'x')
with(frogs, points(waterarea, count1, col = cols[fish], pch = 16))
legend('topleft', legend = c('fish 0', 'fish 1'), pch = 16,
       col = cols,
       cex = 0.8)
```



With ggplot2

```
ggplot(frogs) +
  geom_point(aes(x = waterarea, y = count1, col = fish)) +
  scale_x_log10()
```



Why ggplot2?

base graphics

- more work / code
- legends, colors?
- defaults ok
- multipanel plots?
- can do anything
- have to know the *tricks*

ggplot2

- quick-and-dirty and complex
- Automatic legends, colors!
- nice defaults
- easy multipanel plots
- restrictions (e.g. 2nd y-axis)
- intuitive syntax

Why ggplot2?

base graphics

- more work / code
- legends, colors?
- defaults ok
- multipanel plots?
- can do anything
- have to know the *tricks*

ggplot2

- quick-and-dirty and complex
- Automatic legends, colors!
- nice defaults
- easy multipanel plots
- restrictions (e.g. 2nd y-axis)
- intuitive syntax

Tip:

If a plot is too complicated to draw with ggplot2, reconsider if it's a good representation of your data!

Terminology

```
ggplot(frogs) +
  geom_point(
    aes(x = waterarea, y = count1, col = fish)) +
  scale_x_log10() +
  theme_bw()
```

ggplot() The main function. Can specify the data set and variables **globally**.

geom A geometric object: `geom_point`, `geom_line`, `geom_text`, `geom_violin`, ...

aes aesthetics: maps a variable to the properties of a geom: `shape`, `color`, `fill`, `linetype`, transparency (`alpha`), ...

scale How are data mapped visually (`log`, `date`, `colors`, `sizes`, ...)

theme The general plot apperance (`background`, `axes`, `labels`, ...)

```
ggplot(frogs) +
  geom_point(
    aes(x = waterarea, y = count1, col = fish)) +
  scale_x_log10() +
  theme_bw()
```

ggplot() The main function. Can specify the data set and variables **globally**.

geom A geometric object: `geom_point`, `geom_line`, `geom_text`, `geom_violin`, ...

aes aesthetics: maps a variable to the properties of a geom: `shape`, `color`, `fill`, `linetype`, transparency (`alpha`), ...

scale How are data mapped visually (`log`, `date`, `colors`, `sizes`, ...)

theme The general plot apperance (`background`, `axes`, `labels`, ...)

Tip:

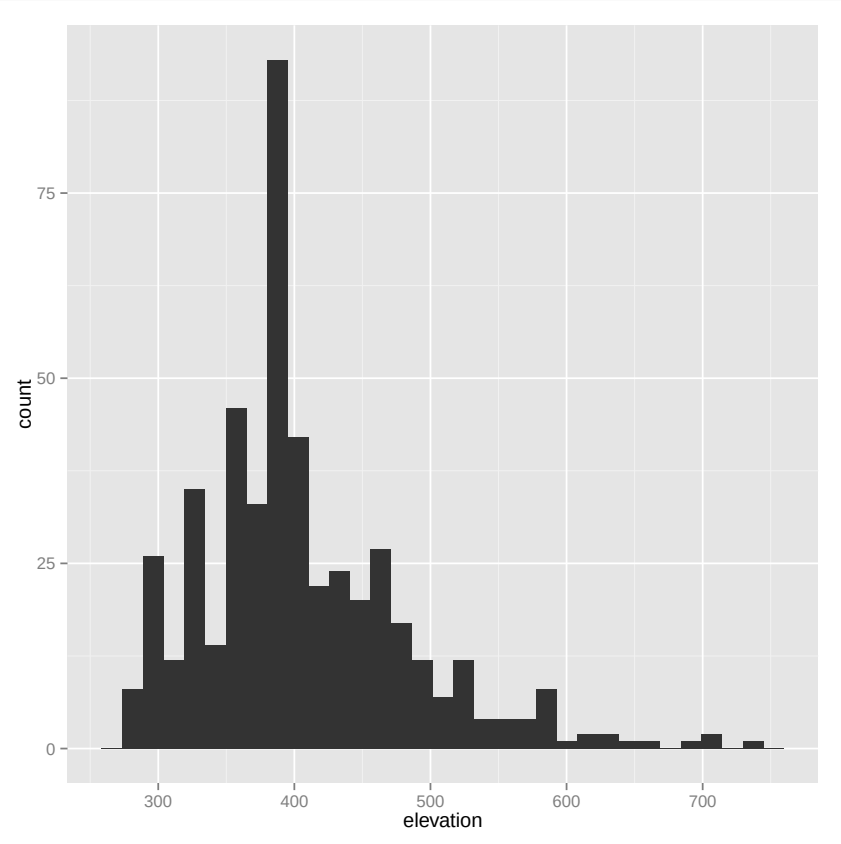
Never use `qplot()` (=quick plot)! - You won't learn the grammar...

Univariate Data

GEOM: HISTOGRAM

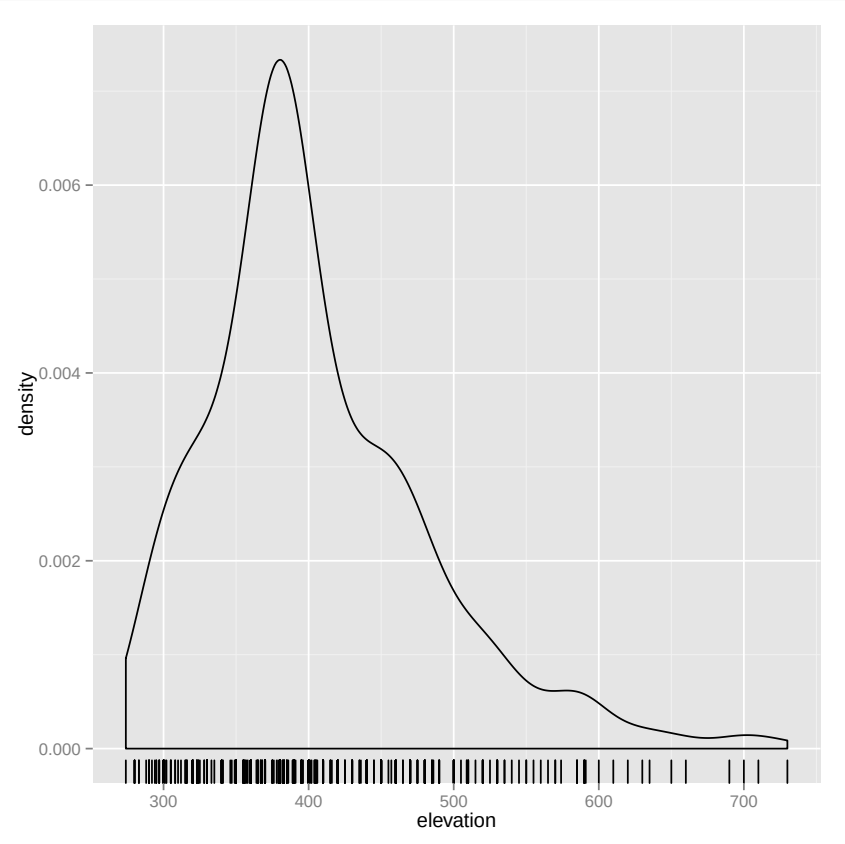
```
ggplot(frogs,                          # use the frogs
       aes(x = elevation))+   # take the variable 'elevation' from the dataset)
  geom_histogram()                     # display a histogramm

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
this.
```



We can add multiple geoms of the same variable to the plot

```
ggplot(frogs, aes(x = elevation)) +    # plot the 'elevation' from the frogs data
  geom_density() +                        # display a density
  geom_rug()                              # display a rug
```

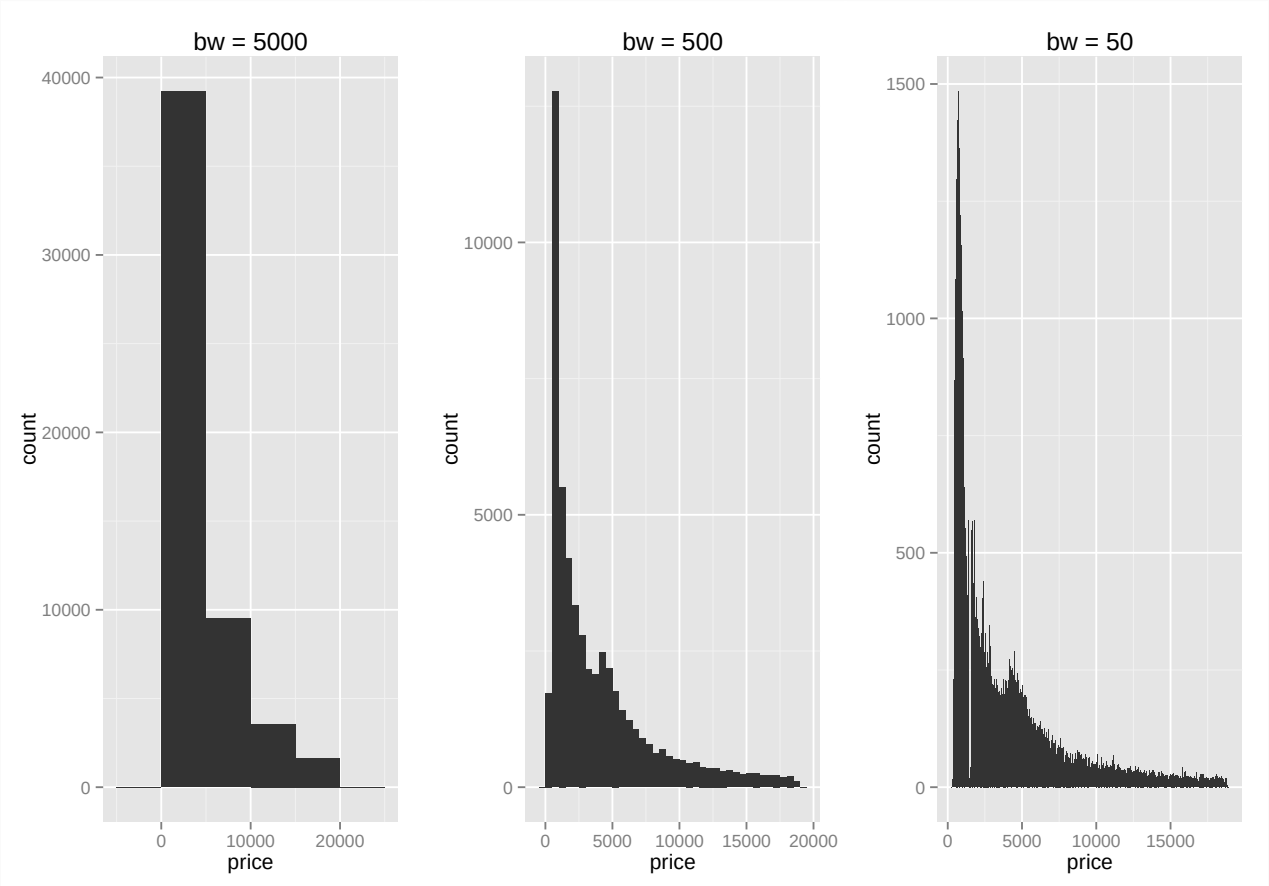


Exercise 1:

Plot a histogramm of diamond prices at different binwidths (50, 500, 5000).

How does this affect the preception of the data?

Exercise

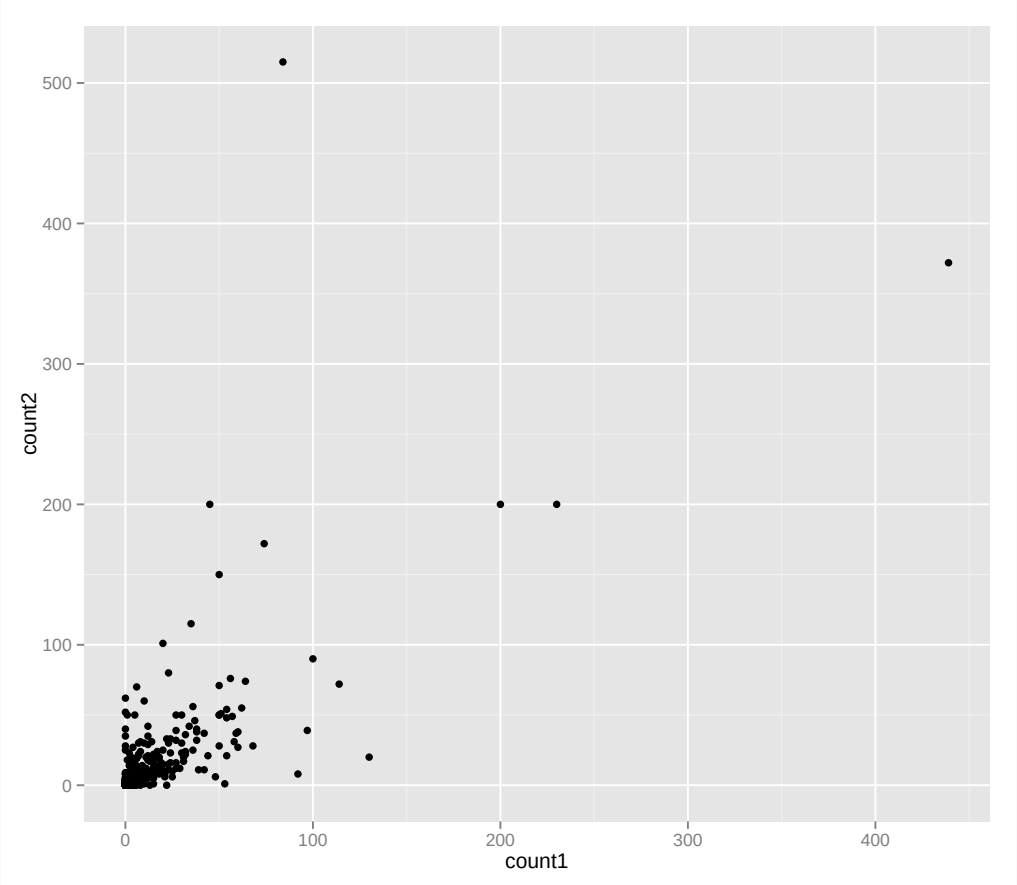

bw = 5000
bw = 500
bw = 50
count
price
0
10000
20000
30000
40000
5000
15000
500
1000
1500

Bivariate data
(cont. x cont.)

GEOM: POINT (=CONT. X CONT.)

The mother of all plots.

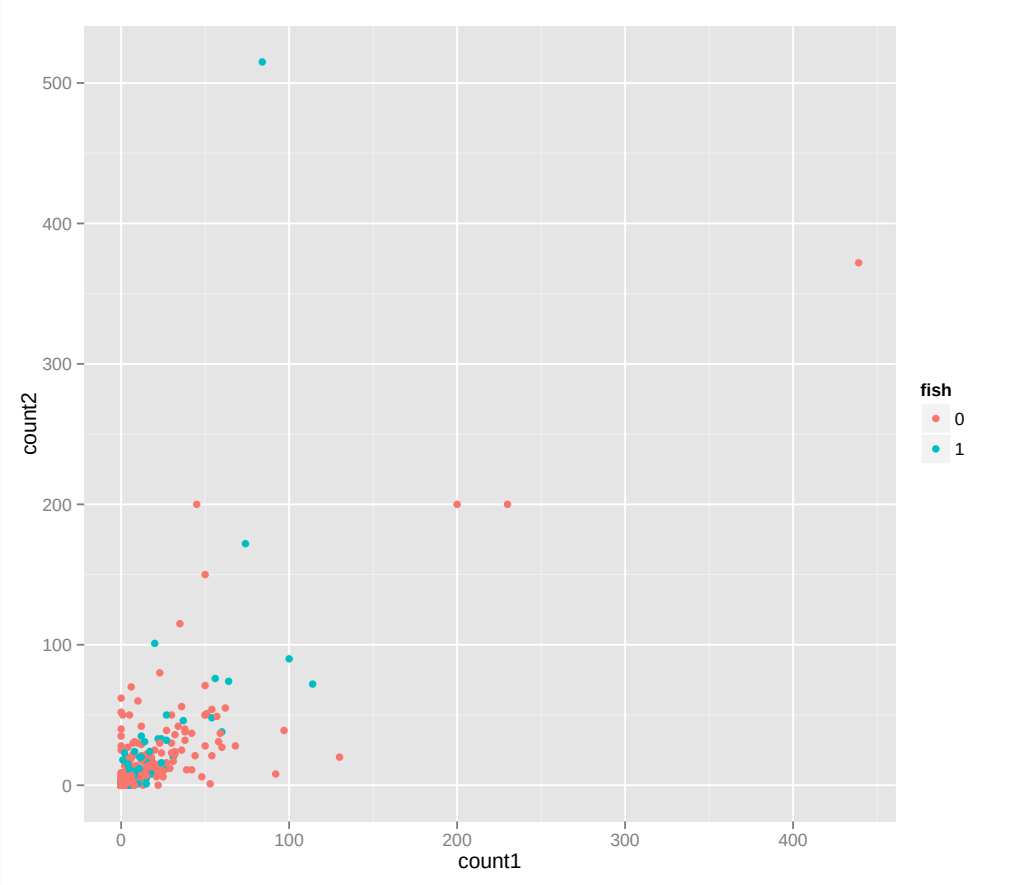
```
ggplot(frogs, aes(x = count1, y = count2)) +
  geom_point()
```



AESTHETIC: COLOR (DISCRETE)

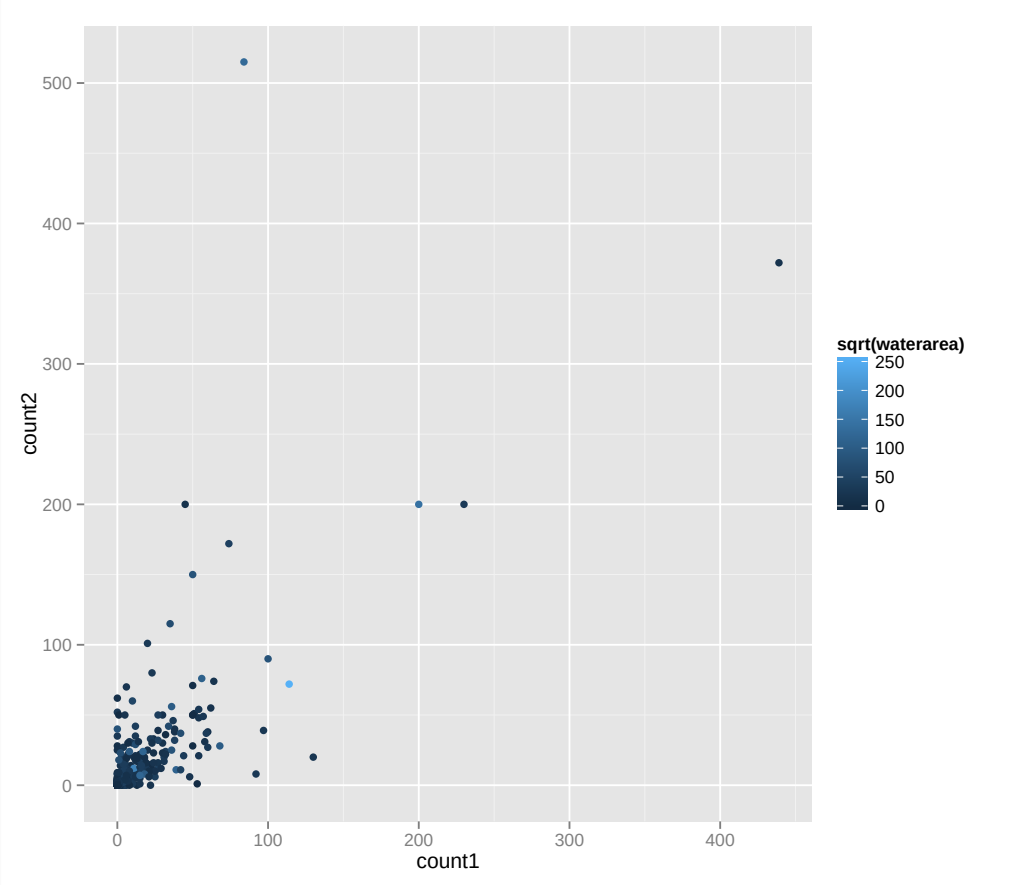
Differentiate between fish and no-fish:

```
ggplot(frogs, aes(x = count1, y = count2, col = fish)) +
  geom_point()
```



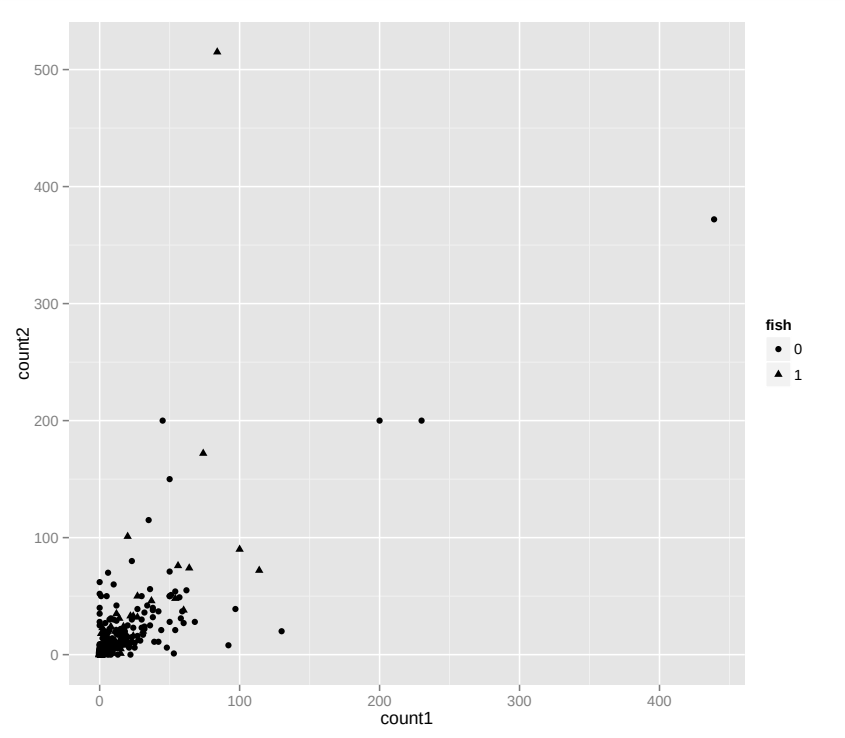
Continuous color scale:

```
ggplot(frogs, aes(x = count1, y = count2, col = sqrt(waterarea))) +
  geom_point()
```



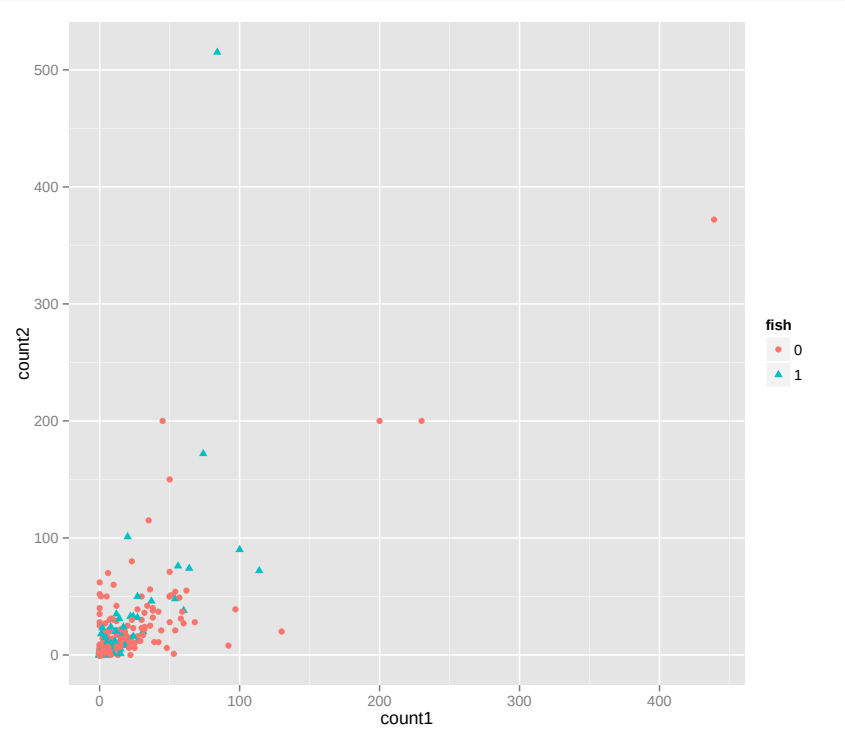
Can also differentiate by shape:

```
ggplot(frogs, aes(x = count1, y = count2, shape = fish)) +
  geom_point()
```



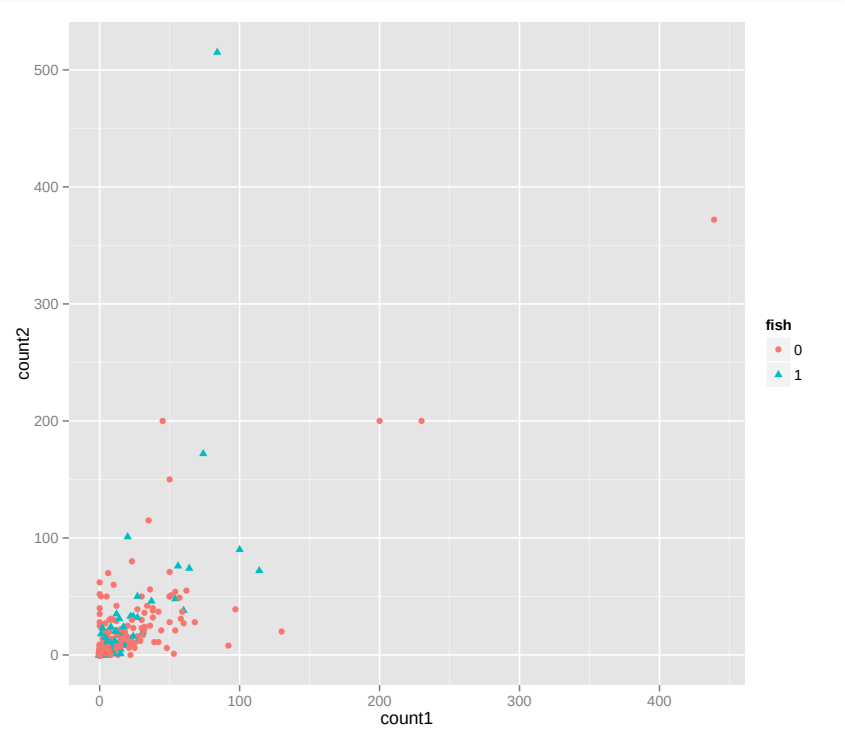
Can also differentiate by shape:

```
ggplot(frogs, aes(x = count1, y = count2, shape = fish, col = fish)) +
  geom_point()
```



Can also differentiate by shape:

```
ggplot(frogs, aes(x = count1, y = count2, shape = fish, col = fish)) +
  geom_point()
```

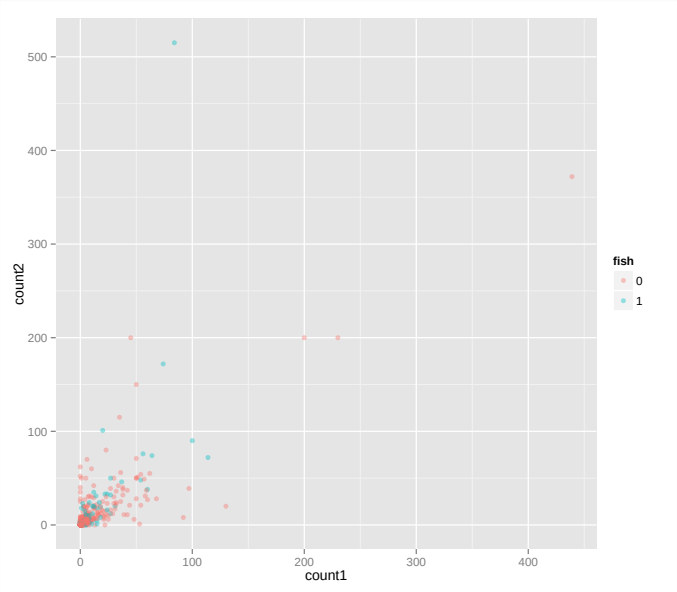


Tip:

Do not use redundant aesthetics.

AESTHETICS: ALPHA (=TRANSPARENCY)

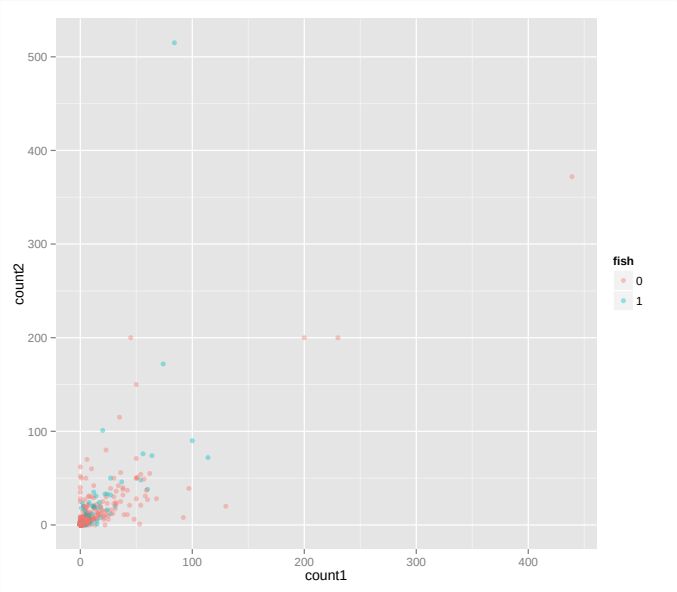
```
ggplot(frogs) +
  geom_point(aes(x = count1, y = count2, col = fish), alpha = 0.4)
```



Q: alpha is not within aes(), why?

AESTHETICS: ALPHA (=TRANSPARENCY)

```
ggplot(frogs) +
  geom_point(aes(x = count1, y = count2, col = fish), alpha = 0.4)
```

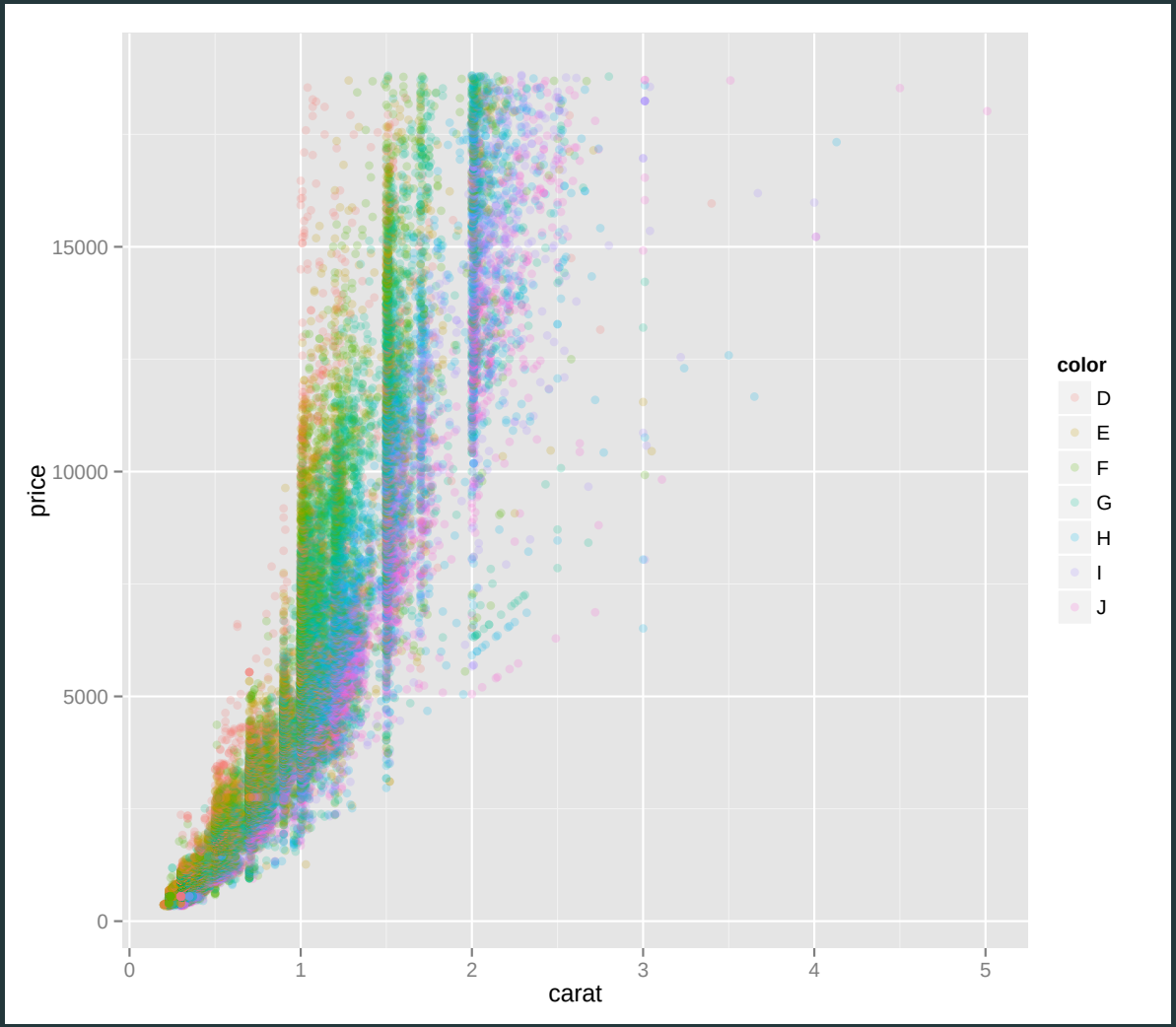


Q: alpha is not within aes(), why?

A:

Within `aes()` the aesthetic changes with the specified variable (e.g. `col = fish`). Outside of `aes` we set it to a specific value.

Exercise 2:
Create this plot:

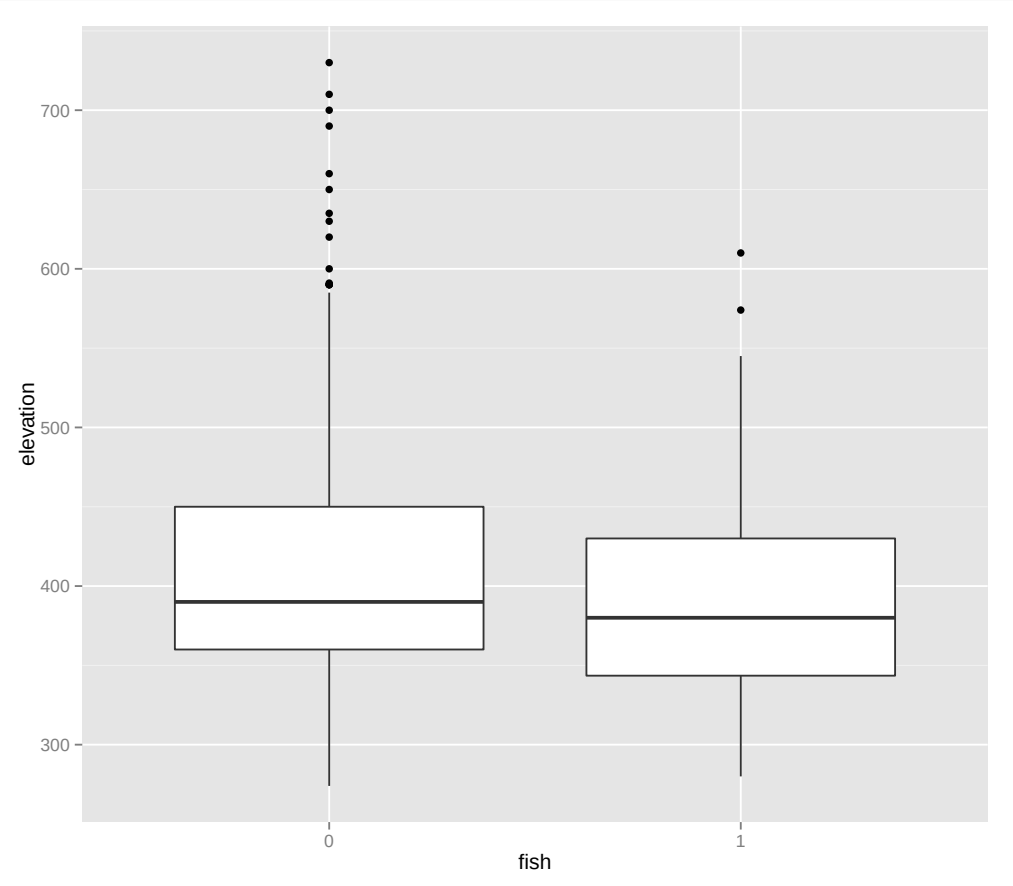


Bivariate data
(discrete x continuous)

GEOM: BOXPLOT

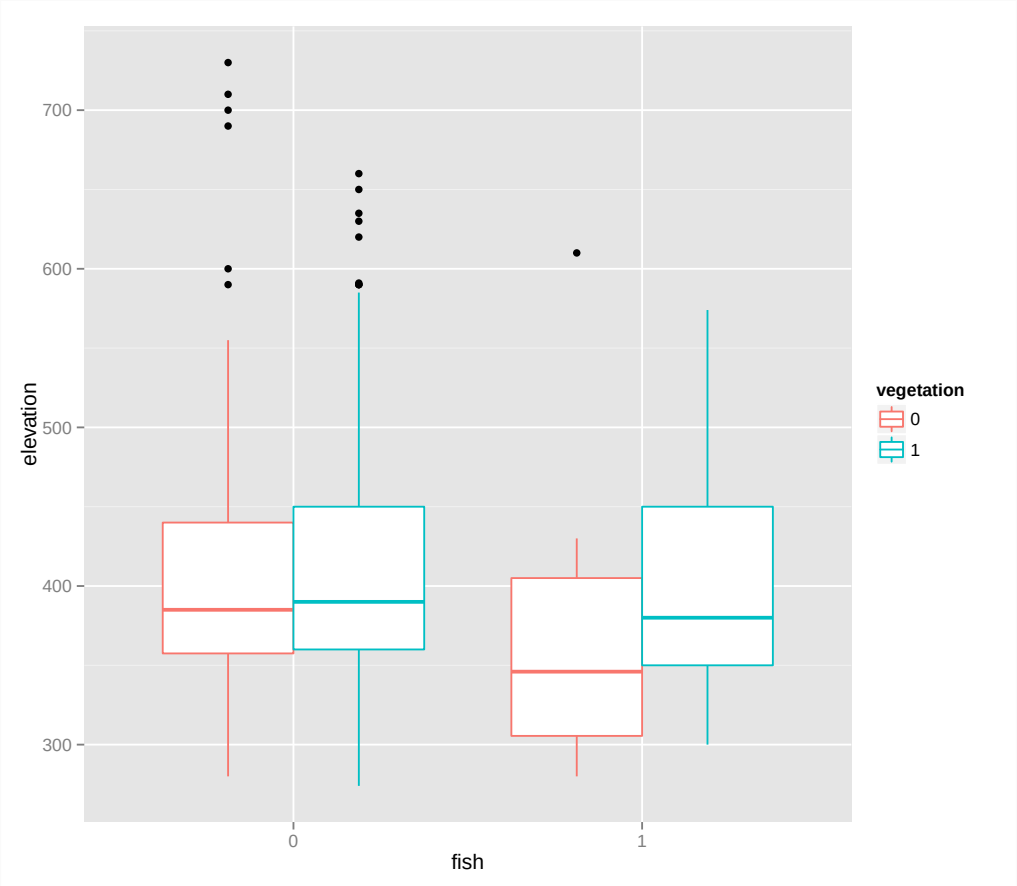
- More then 45 years (Tukey 1970)
- For discrete scales (here fish)

```
ggplot(frogs, aes(x = fish, y = elevation)) +
  geom_boxplot()
```



AESTHTIC: COLOR (DISCRETE)

```
ggplot(frogs, aes(x = fish, y = elevation, col = vegetation)) +
  geom_boxplot()
```

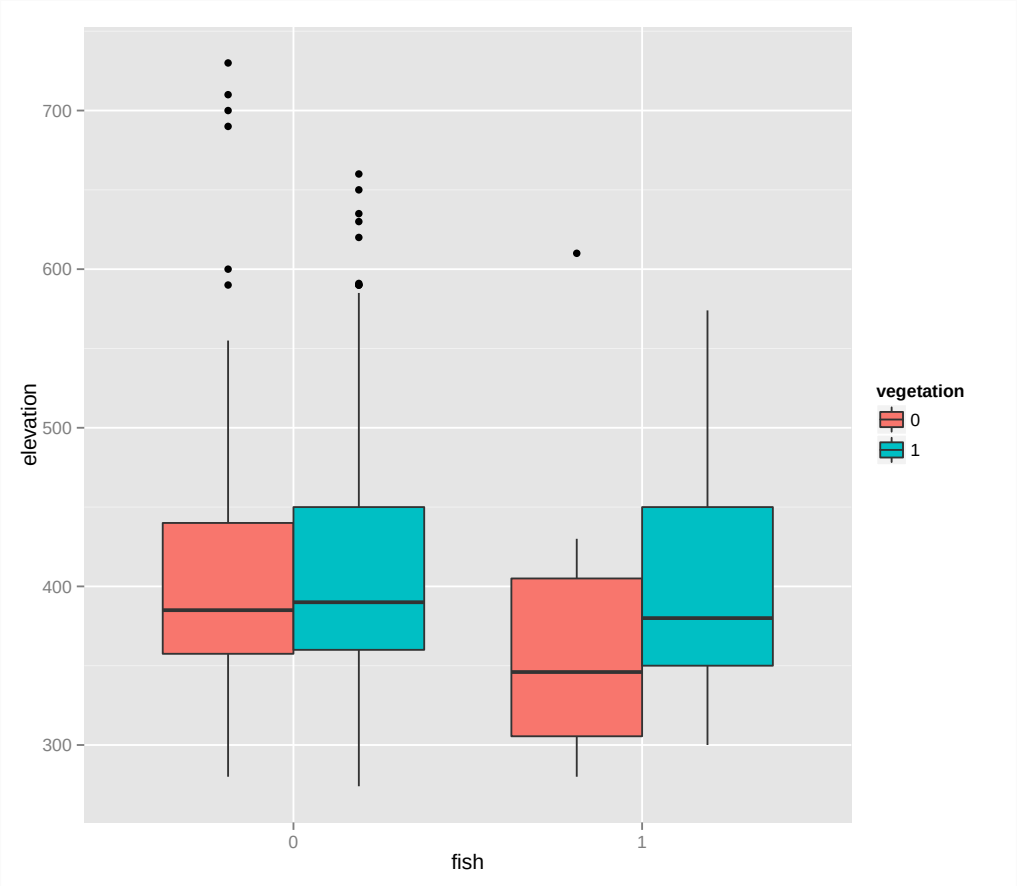


Tip:

Nice for factorial designs (e.g. easily spot interactions).

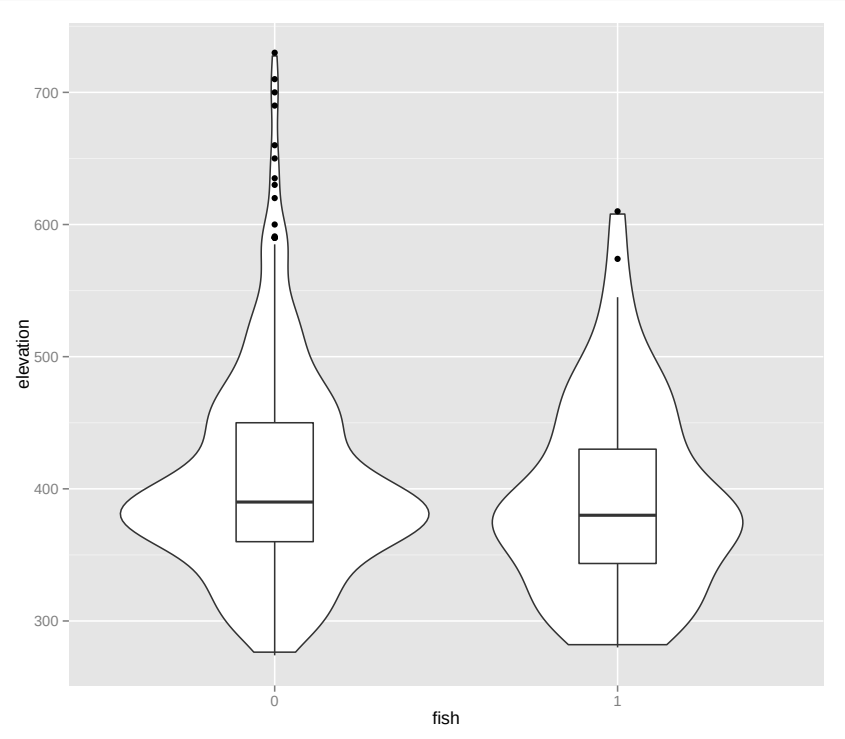
AESTHTIC: FILL (DISCRETE)

```
ggplot(frogs, aes(x = fish, y = elevation, fill = vegetation)) +
  geom_boxplot()
```



The Violin plot is a powerful alterative/addition:

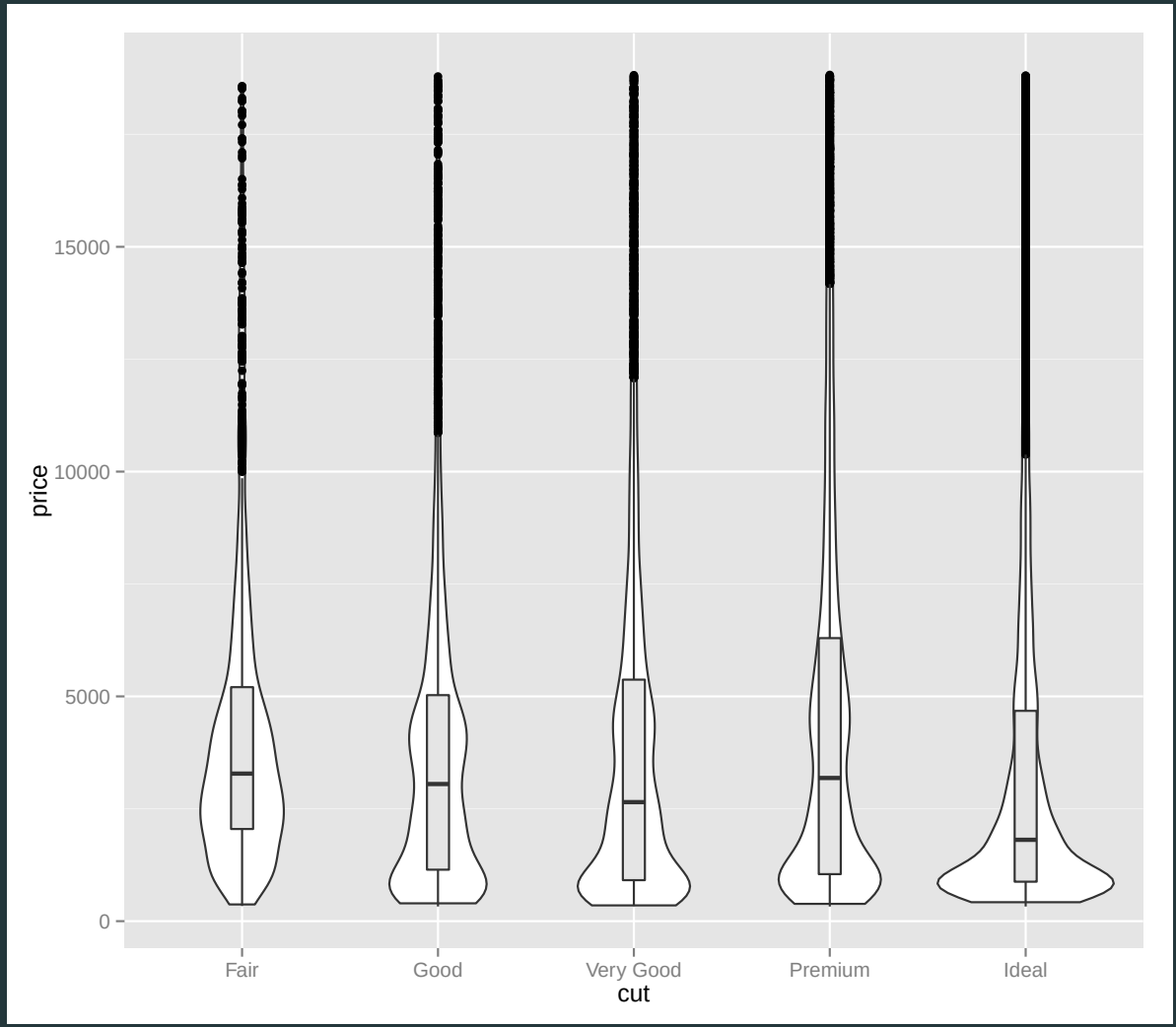
```
ggplot(frogs, aes(x = fish, y = elevation)) +
  geom_violin() +
  geom_boxplot(width = 0.3)
```



Tip:

The order of geoms is important! Uppermost layer is the last geom specified.

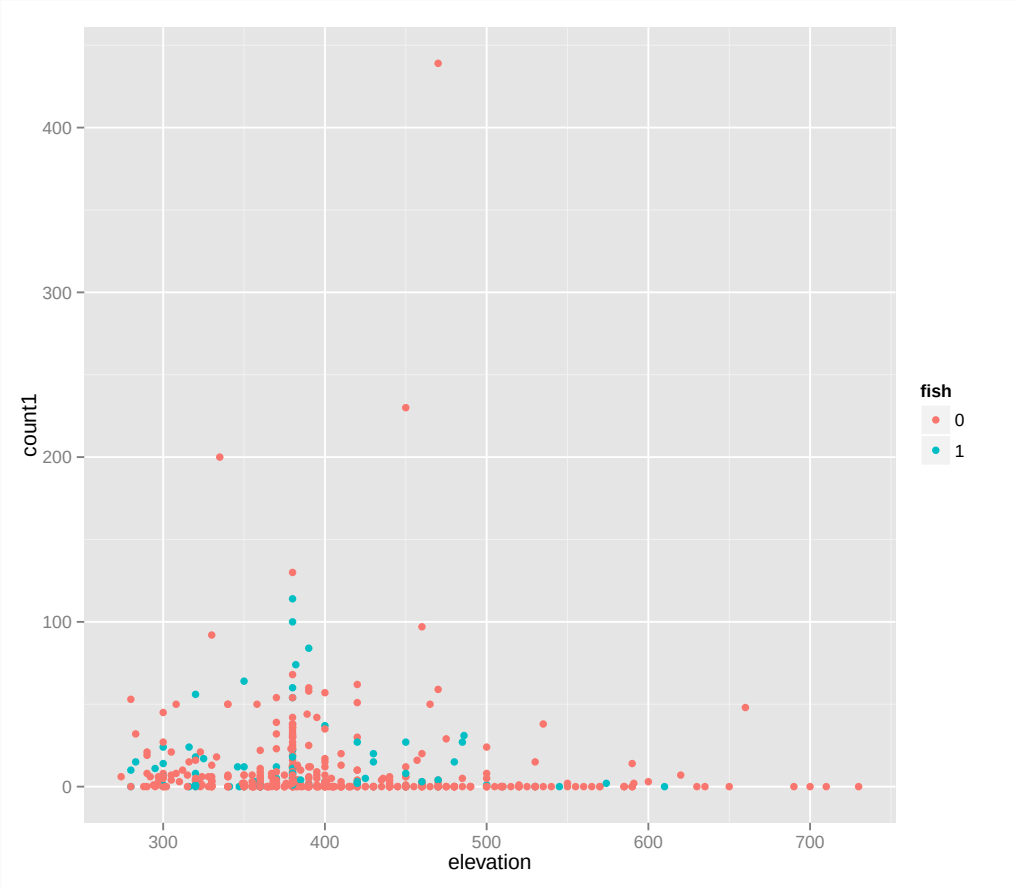
Exercise 3:
Create this plot:



Multivariate data
(Facets)

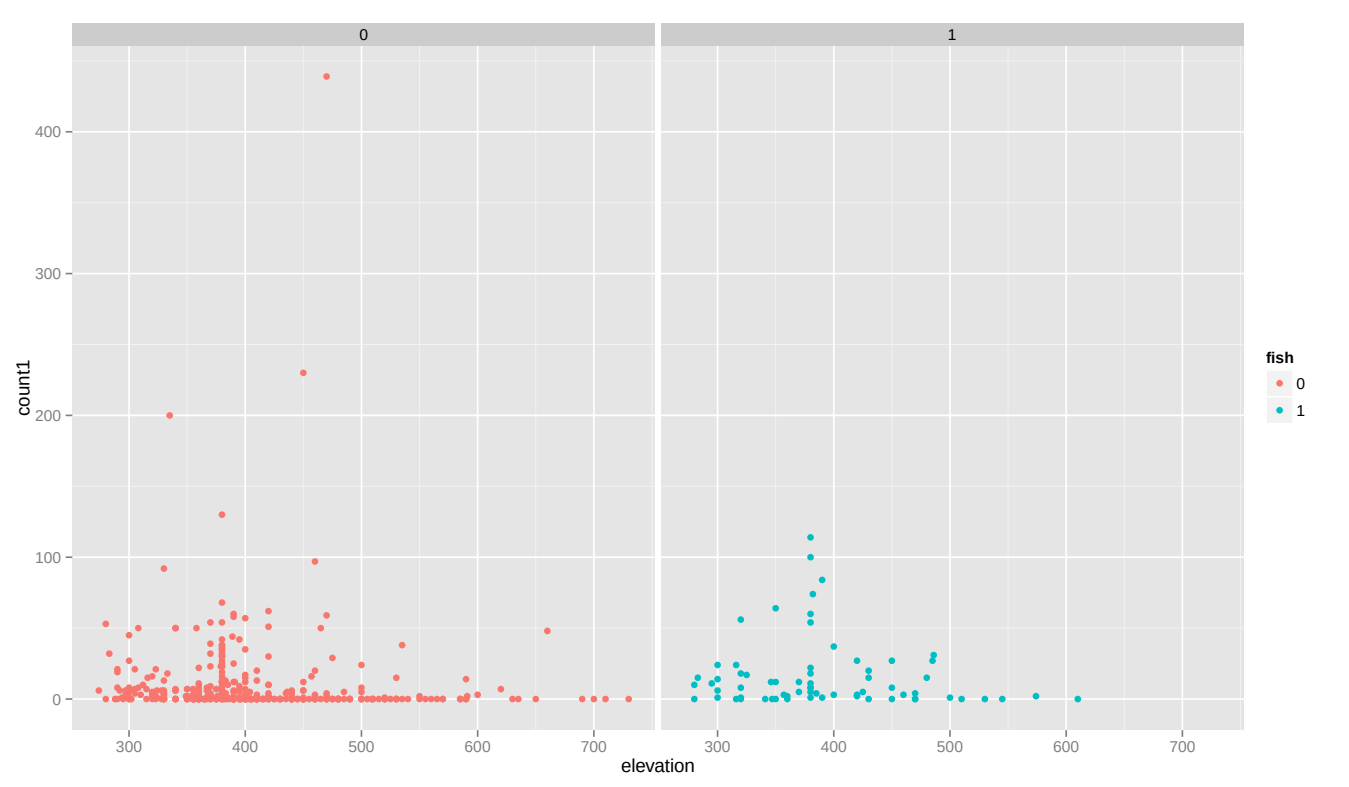
The different relationships between *fish* and *no fish* are hard to spot:

```
ggplot(frogs, aes(x = elevation, y = count1, col = fish)) +
  geom_point()
```



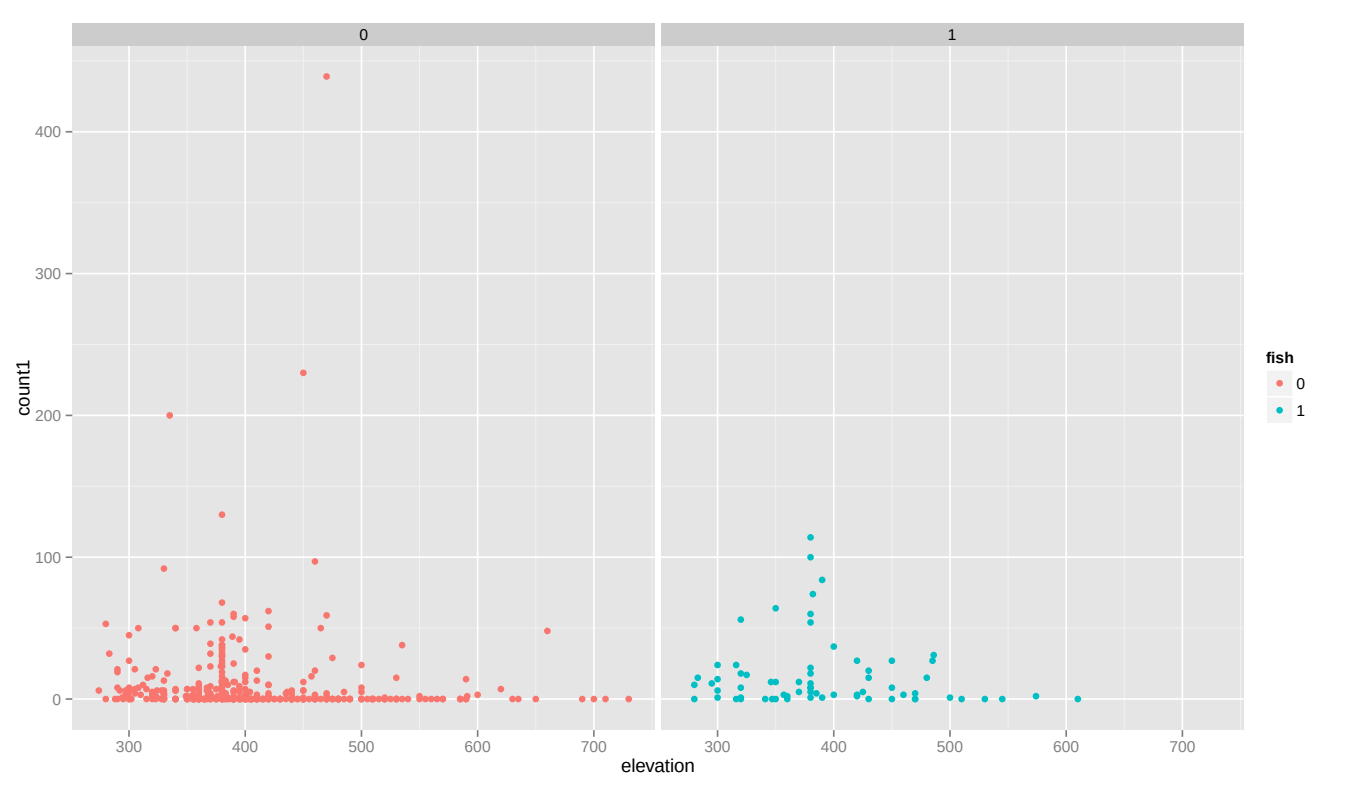
facets: Split the plots by groups

```
ggplot(frogs, aes(x = elevation, y = count1, col = fish)) +
  geom_point() +
  facet_wrap(~fish)
```



facets: Split the plots by groups

```
ggplot(frogs, aes(x = elevation, y = count1, col = fish)) +
  geom_point() +
  facet_wrap(~fish)
```



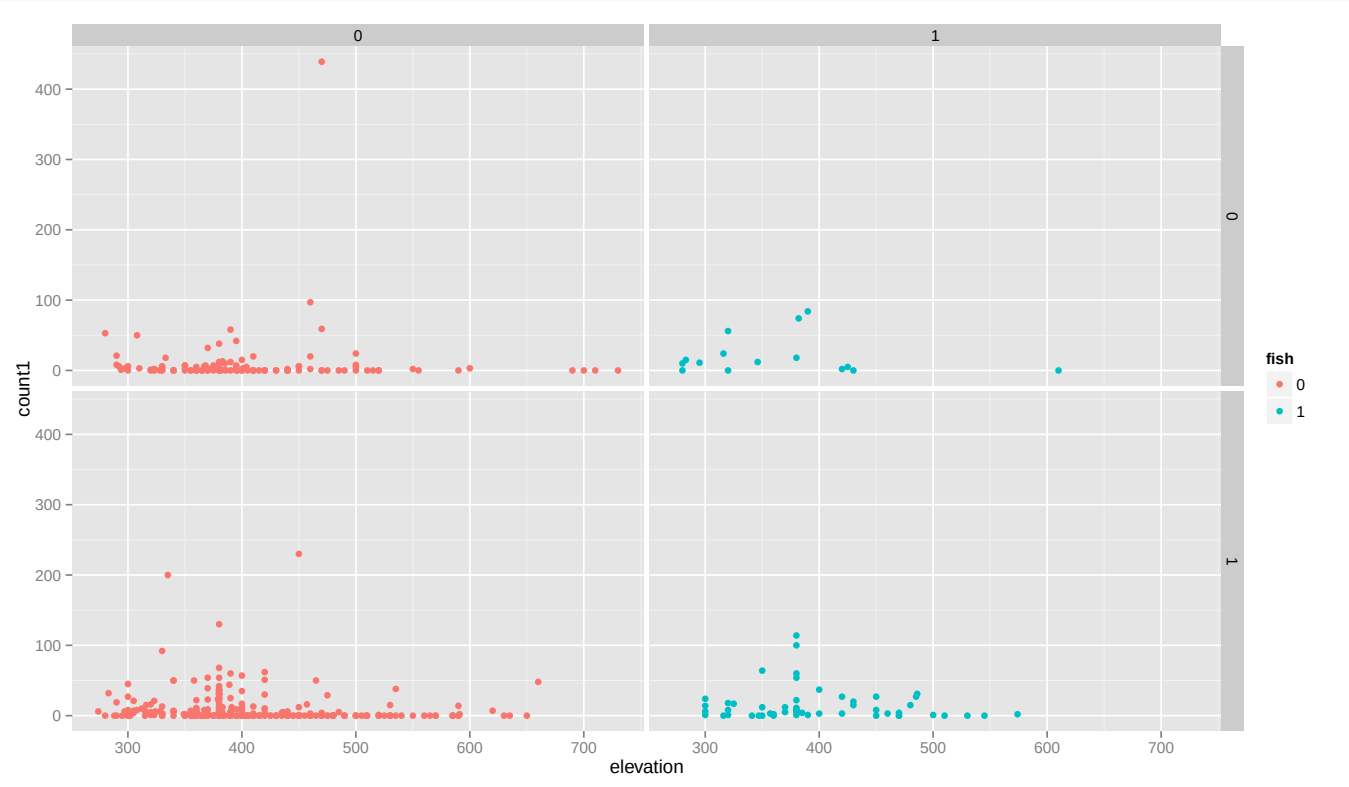
Tip:

The color here is redundant!

facets: Split the plots by groups

- `facet_wrap()` wraps into a rectagular layout
- `facet_grid(rows ~cols)` wraps into rows and columns

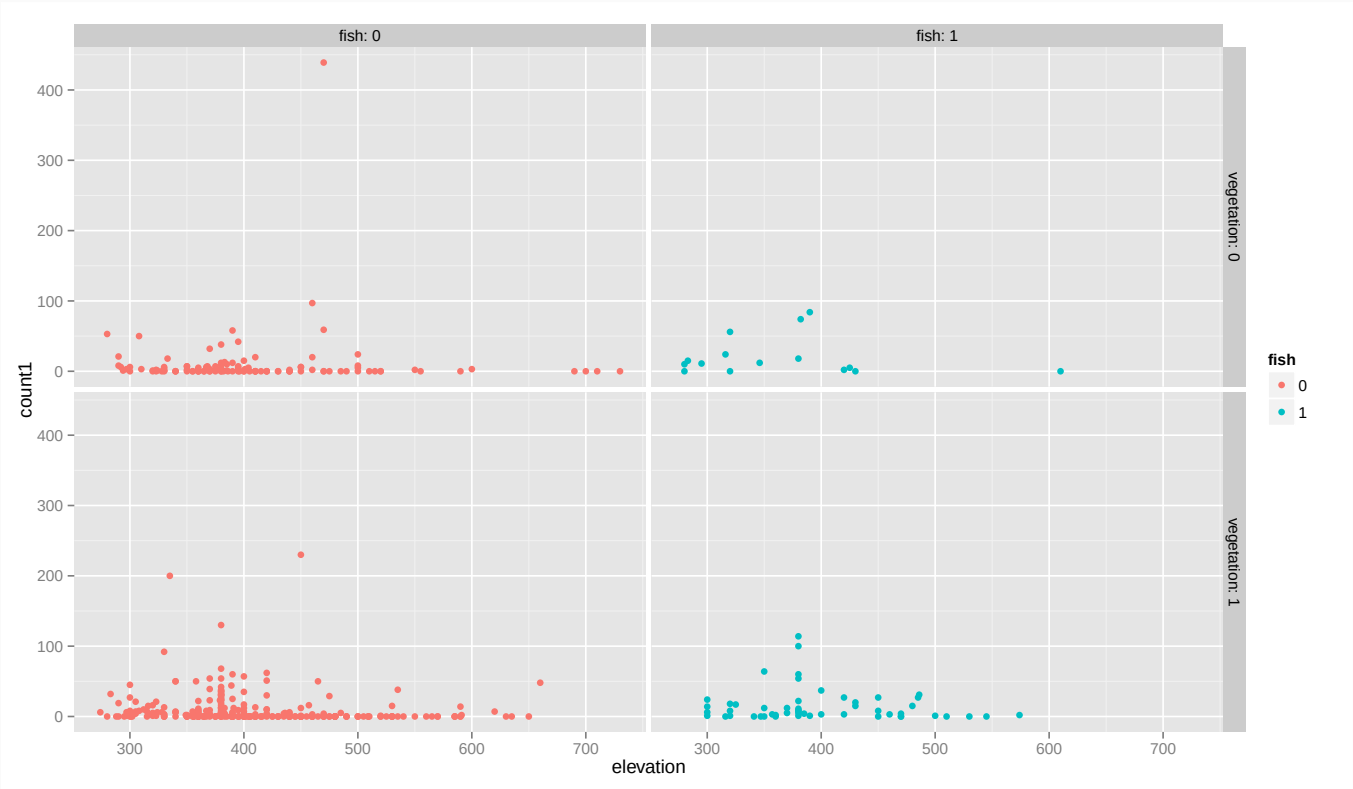
```
ggplot(frogs, aes(x = elevation, y = count1, col = fish)) +
  geom_point() +
  facet_grid(vegetation~fish)
```



facets: Split the plots by groups

- add text to strip to identify variables with `labeller`

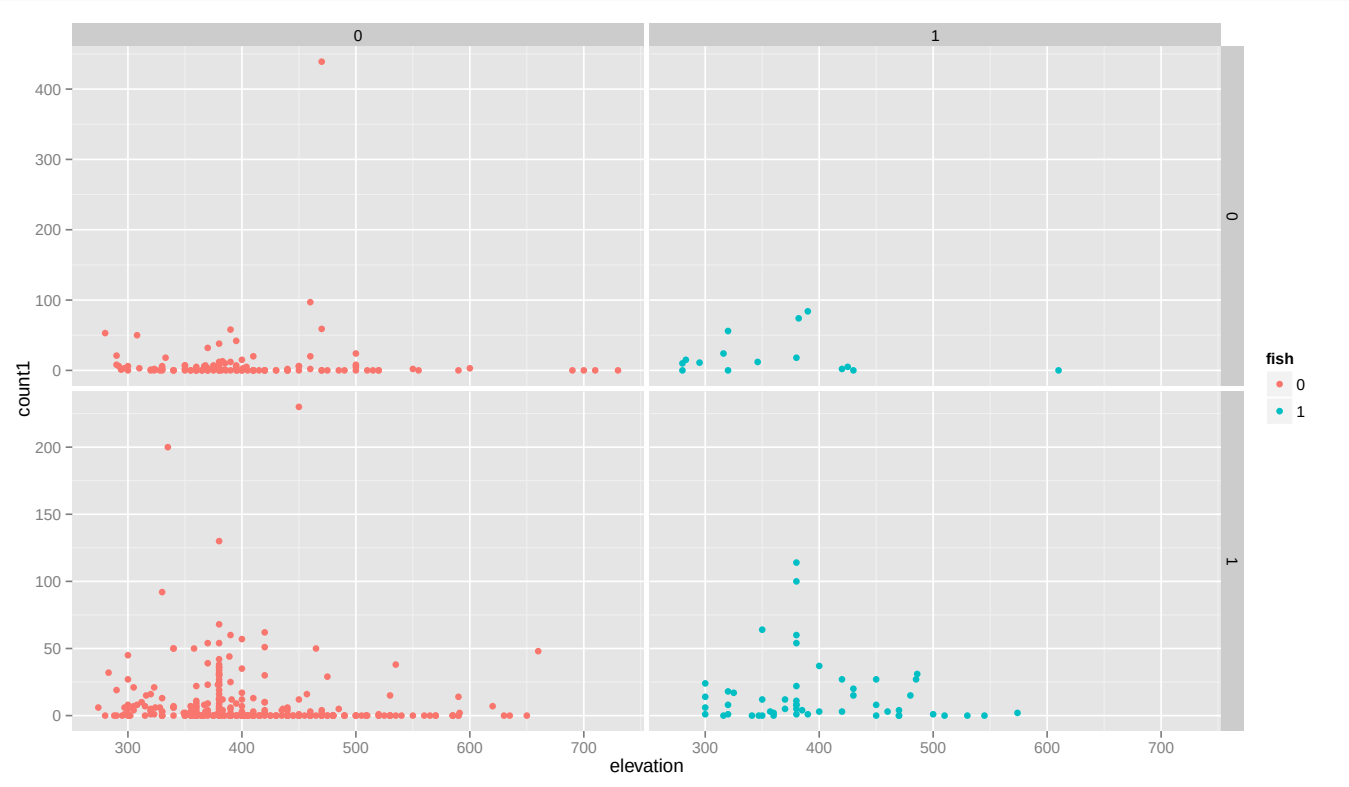
```
ggplot(frogs, aes(x = elevation, y = count1, col = fish)) +
  geom_point() +
  facet_grid(vegetation~fish, labeller = label_both)
```



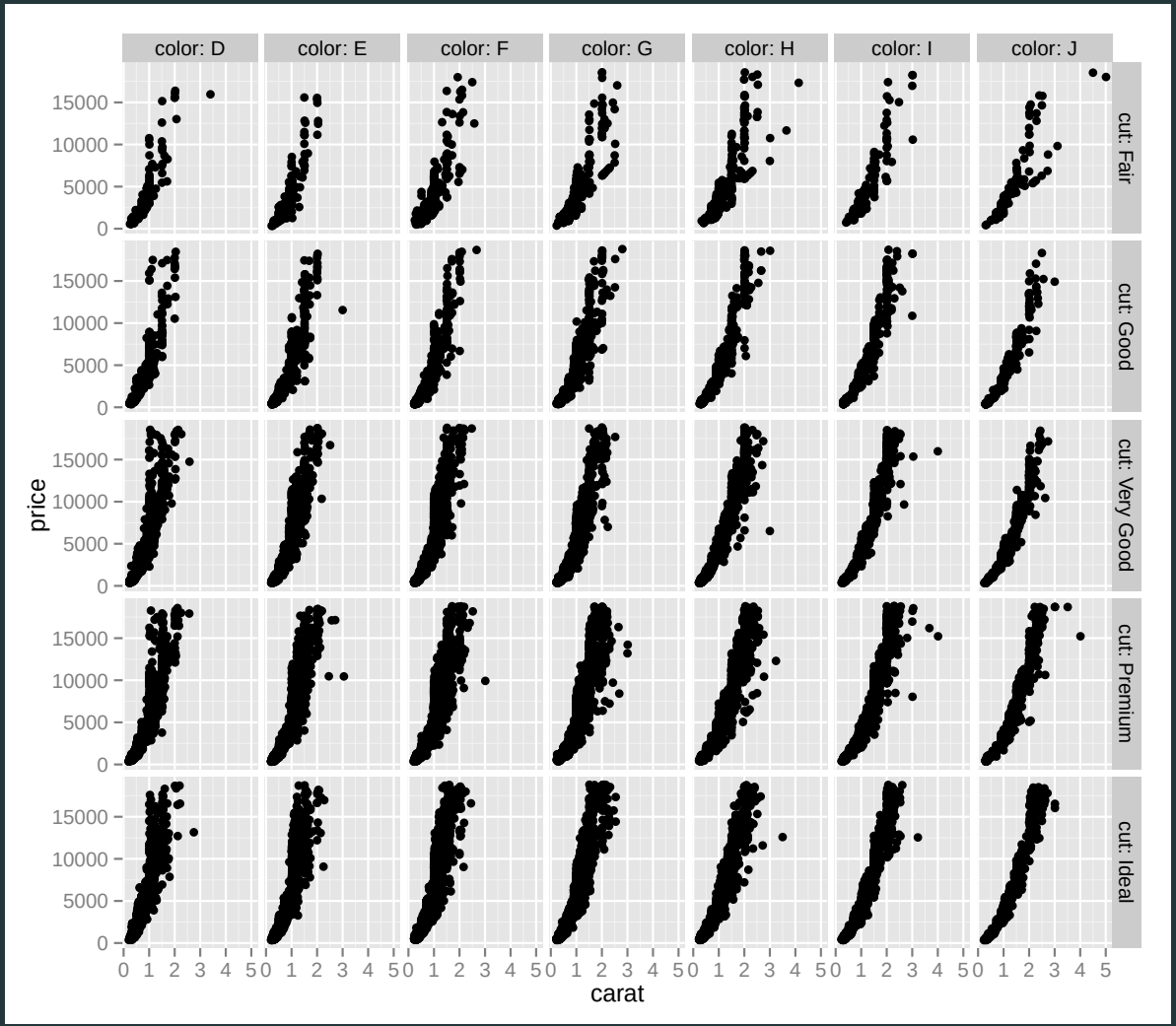
facets: Split the plots by groups

- Both rows have the same scale.
- use `scales = "free" or "frex_x", "free_y"` to adjust

```
ggplot(frogs, aes(x = elevation, y = count1, col = fish)) +
  geom_point() +
  facet_grid(vegetation~fish,
             scales = 'free_y')
```



Exercise 4:
Create this plot:

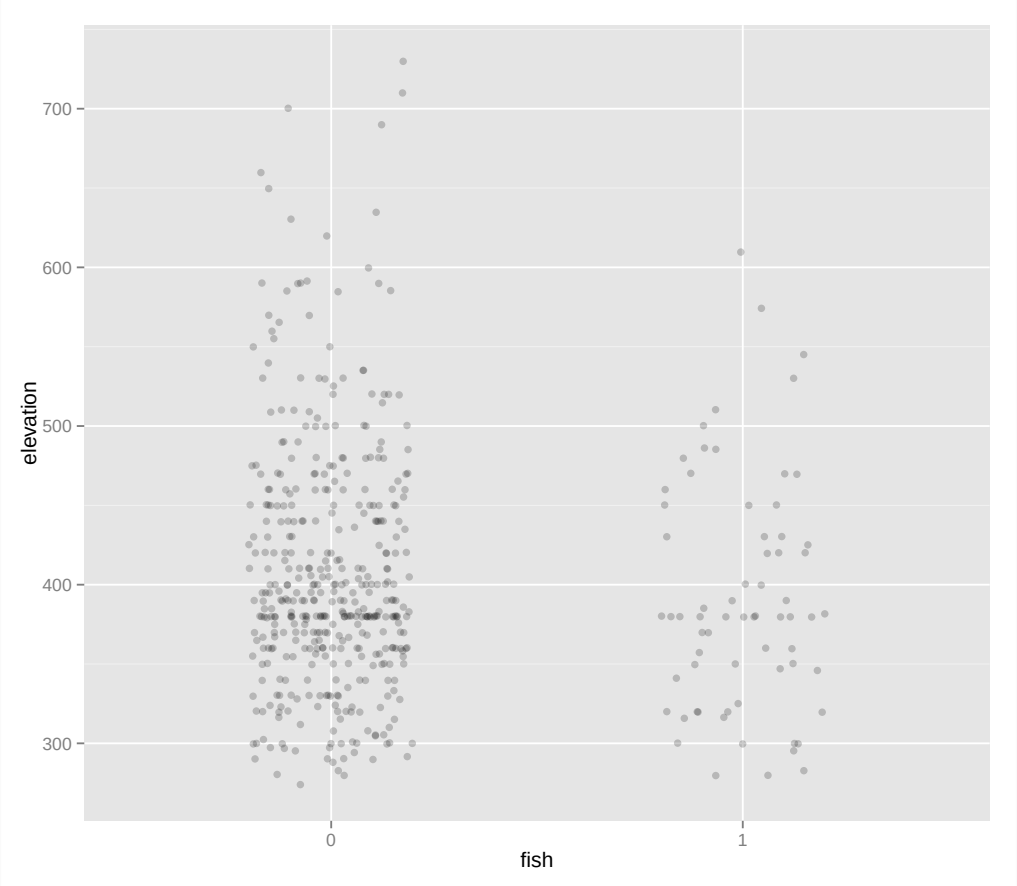


Aggregated data
(Visualize errors)

Aggregate data

- Often we want to show aggregates of our data (e.g. means and CIs)

```
ggplot(frogs, aes(x = fish, y = elevation)) +
  geom_point(position = position_jitter(width = 0.2), alpha = 0.2)
```



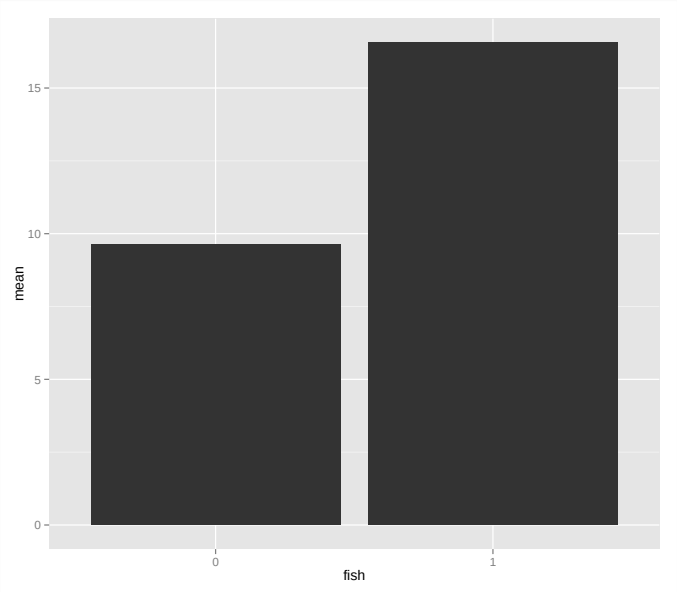
Aggregate data

- Aggregate first, then plot!
- A lot of possibilities to aggregate (`plyr`, `dplyr`, `data.table`, `sqldf`, `ave()`, `doBy`, ...)
- I use `plyr` solutions here.

```
require(plyr)
mean_ci <- ddply(frogs, .(fish), summarise,
      mean = mean(elevation),
      err = qnorm(0.975) * sd(elevation) / sqrt(length(elevation)))
mean_ci
```

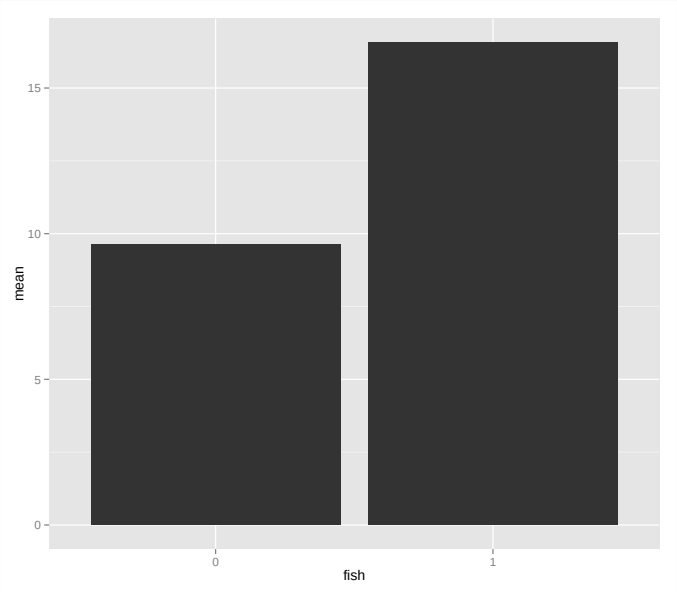
```
##   fish     mean       err
## 1    0 407.5990  7.509866
## 2    1 391.6119 17.560849
```

```
ggplot(mean_ci, aes(x = fish, y = mean)) +
  geom_bar(stat = 'identity')
```



- We need to set `stat = 'identity'` so that the heights of the bars represent the mean.
- Otherwise `geom_bar` tries to transform the data (height = number of cases per group)

```
ggplot(mean_ci, aes(x = fish, y = mean)) +
  geom_bar(stat = 'identity')
```



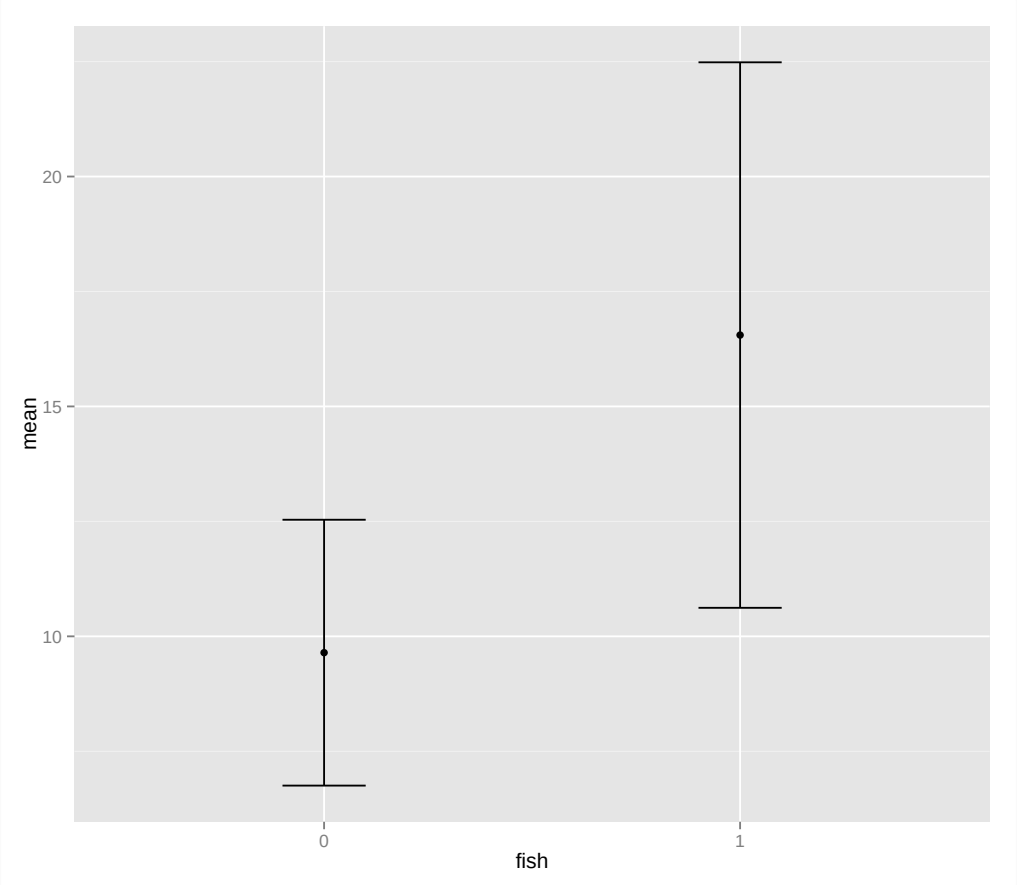
- We need to set `stat = 'identity'` so that the heights of the bars represent the mean.
- Otherwise `geom_bar` tries to transform the data (height = number of cases per group)

Tip:

Don't use bars to display means. Bad data-to-ink ratio (Tufte)!

GEOM: ERRORBARS

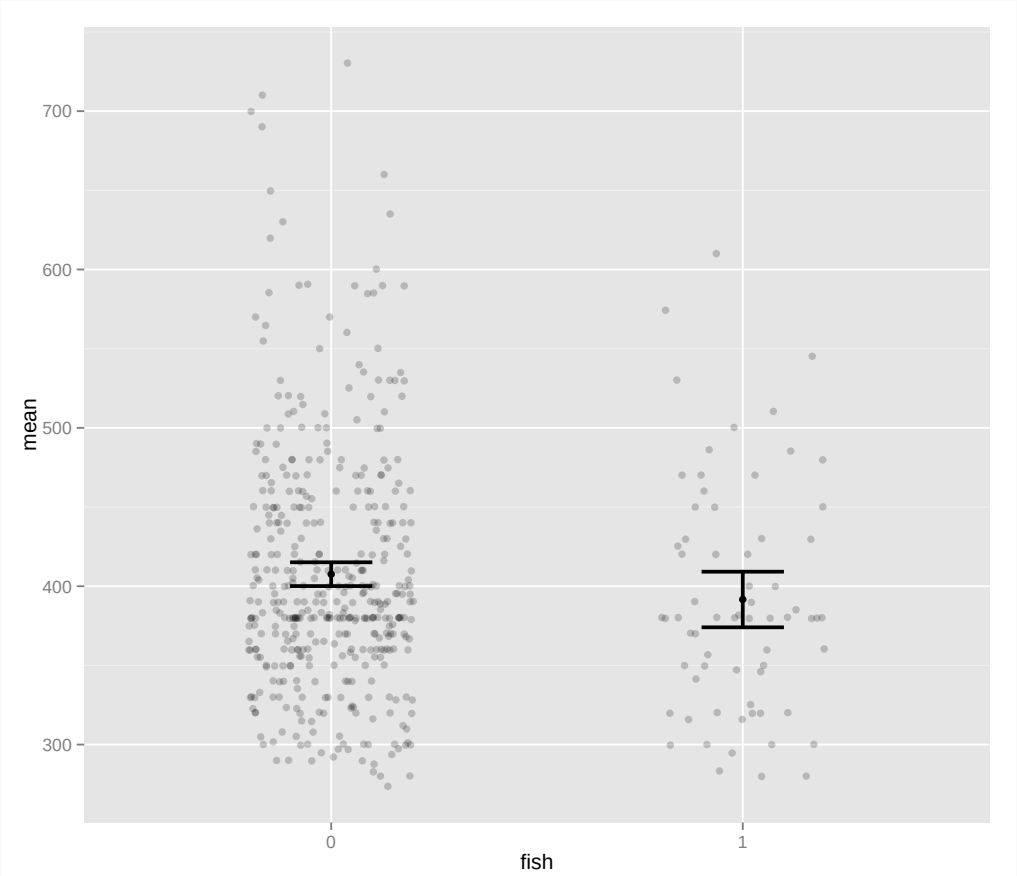
```
ggplot(data = mean_ci, aes(x = fish, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = mean - err, ymax = mean + err), width = 0.2)
```



- points are enough to represent the means
- `width = 0.2` fo smaller whisker (default = 1)

Combining different datasets into on plot

```
ggplot() +
  geom_point(data = mean_ci, aes(x = fish, y = mean)) +
  geom_errorbar(data = mean_ci, aes(x = fish, y = mean,
                                    ymin = mean - err, ymax = mean + err),
                width = 0.2, size = 1) +
  geom_point(data = frogs, aes(x = fish, y = elevation),
             position = position_jitter(width = 0.2),
             alpha = 0.2)
```



Combining different datasets into on plot

- Because we have multiple datasets, we cannot pass them globally in `ggplot()`
- specify data and aesthetics for each layer / geom

Combining different datasets into on plot

- Because we have multiple datasets, we cannot pass them globally in `ggplot()`
- specify data and aesthetics for each layer / geom

Tip:

Plotting the raw data is a good idea (spot errors, make data available, etc...)

MAPS

Customization and Prettification