

```
library(ggplot2)
library(gridExtra)

## Loading required package: grid

opts_chunk$set(cache = TRUE, dev = "cairo_pdf", warning = FALSE, tidy = FALSE) # кэшируем в

# , out.height='3cm',out.width='3cm'
```

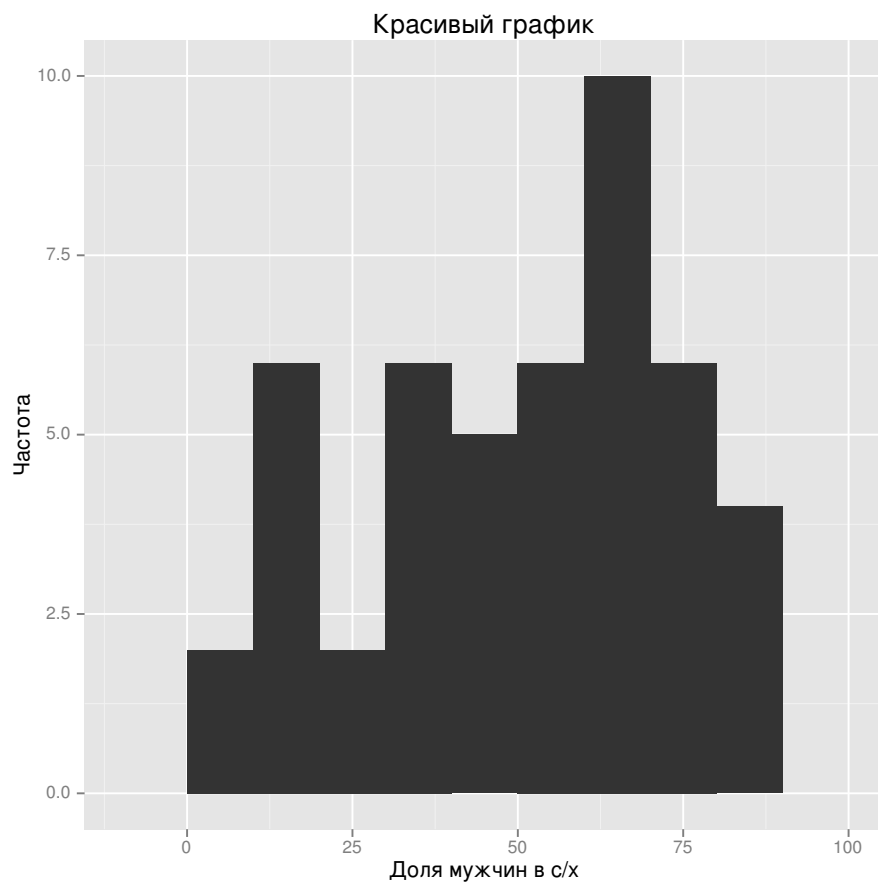
1 Одна переменная

Количественная:

```
h <- swiss
```

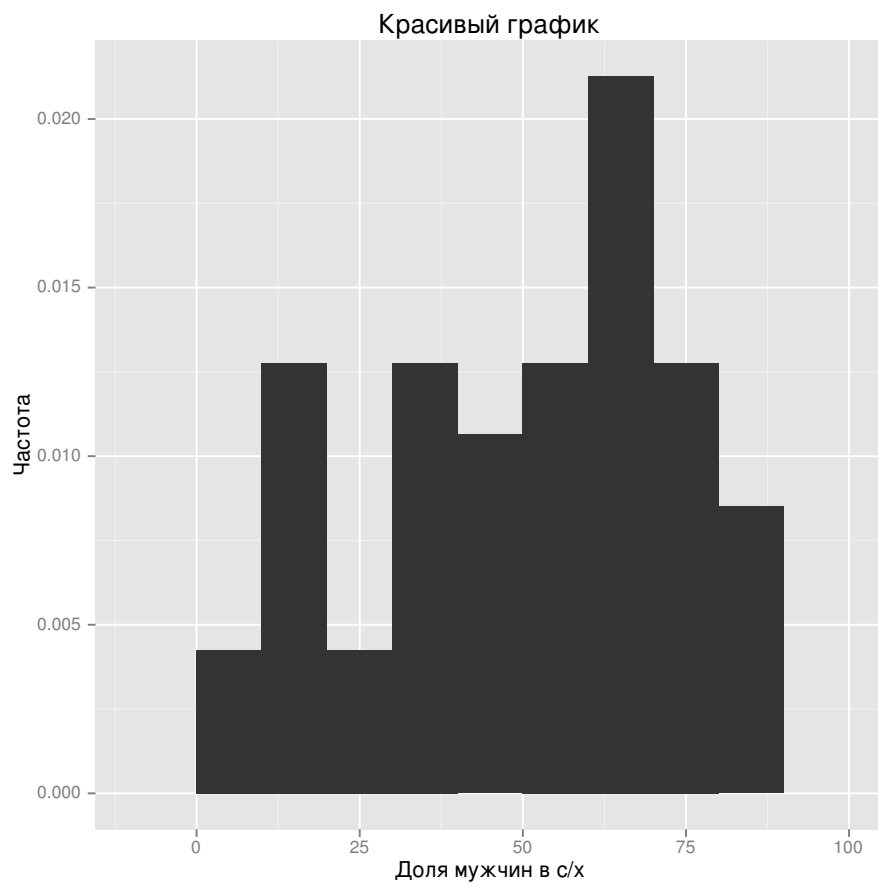
1. Гистограмма

```
p1 <- ggplot(h,aes(x=Agriculture)) +
  geom_histogram(binwidth=10) +
  xlab("Доля мужчин в с/х") +
  ylab("Частота") +
  ggtitle("Красивый график")
p1
```



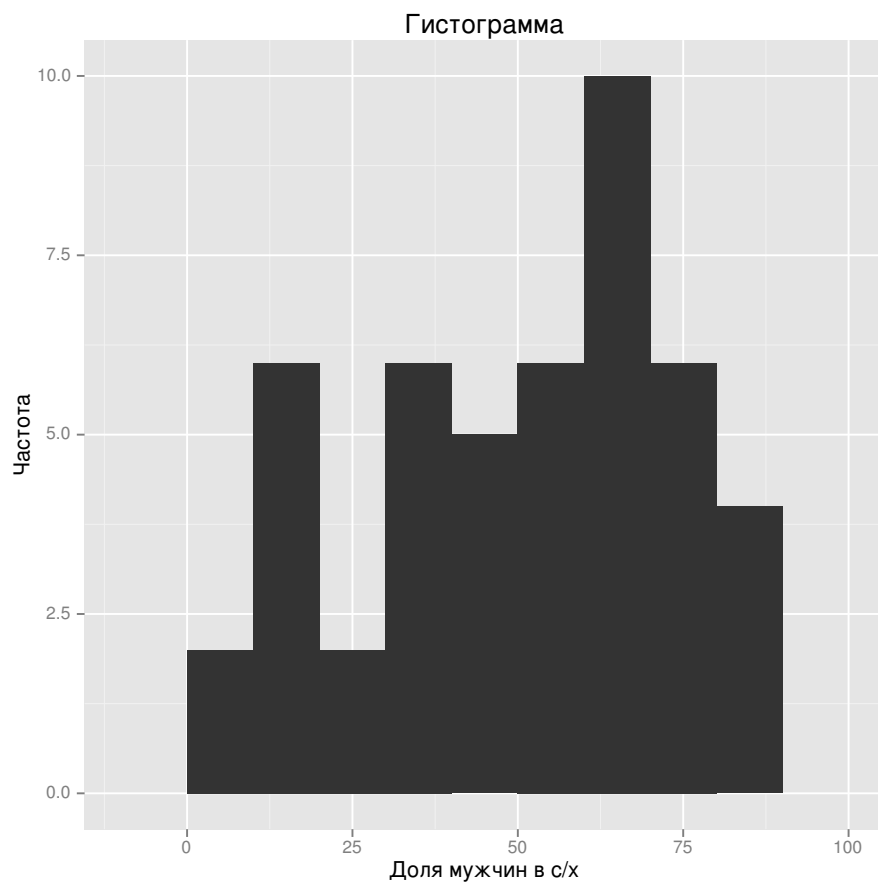
Вместо частот по вертикальной оси можно откладывать доли:

```
p2 <- ggplot(h,aes(x=Agriculture)) +
  geom_histogram(binwidth=10,aes(y=..density..)) +
  xlab("Доля мужчин в с/х") +
  ylab("Частота") +
  ggtitle("Красивый график")
p2
```



Альтернативная команда:

```
p3 <- qplot(data=h,  
  x=Agriculture, geom="histogram", binwidth=10,  
  
  main="Гистограмма",  
  xlab="Доля мужчин в с/х",  
  ylab="Частота")  
p3
```

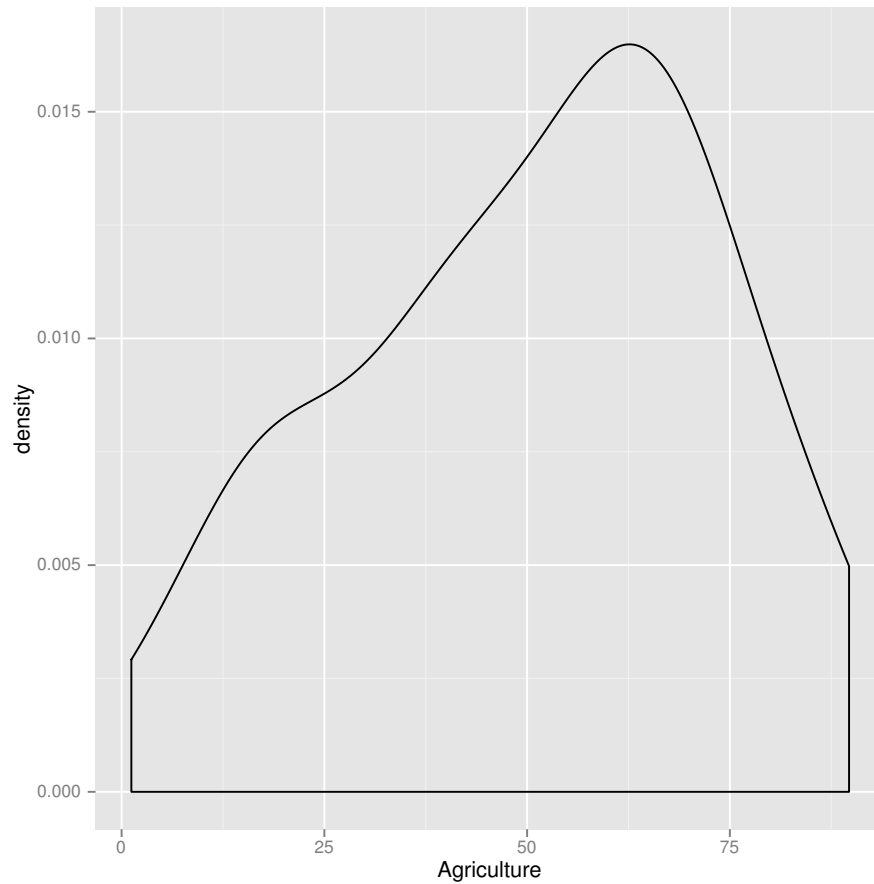


Графики рядом.

Для краткости в следующих графиках названия и подписи осей мы пропустим.

2. Оценка функции плотности

```
ggplot(h,aes(x=Agriculture)) +  
  geom_density()
```



Опции `geom_density()`:

`adjust` — корректирующий множитель для ширины окна сглаживания

3. Beanplot

4. Добавим функцию на график

5. QQ плот

Качественная:

1. Гистограмма

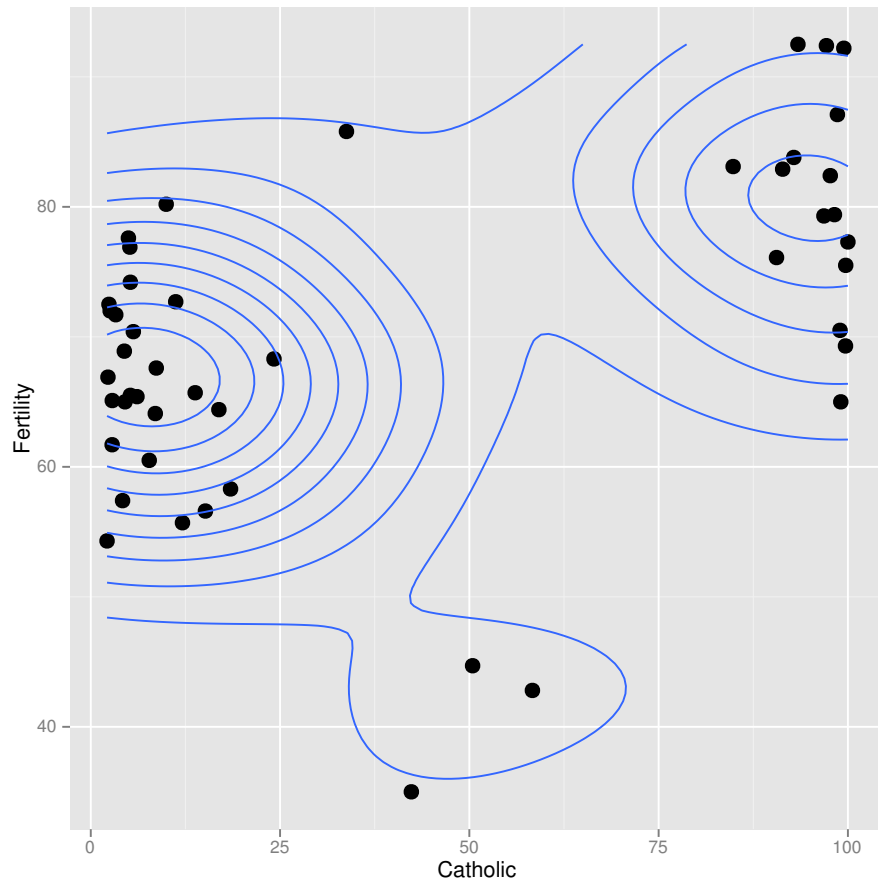
Если предварительные подсчёты уже проведены...

2 Две переменных

Количественная - количественная.

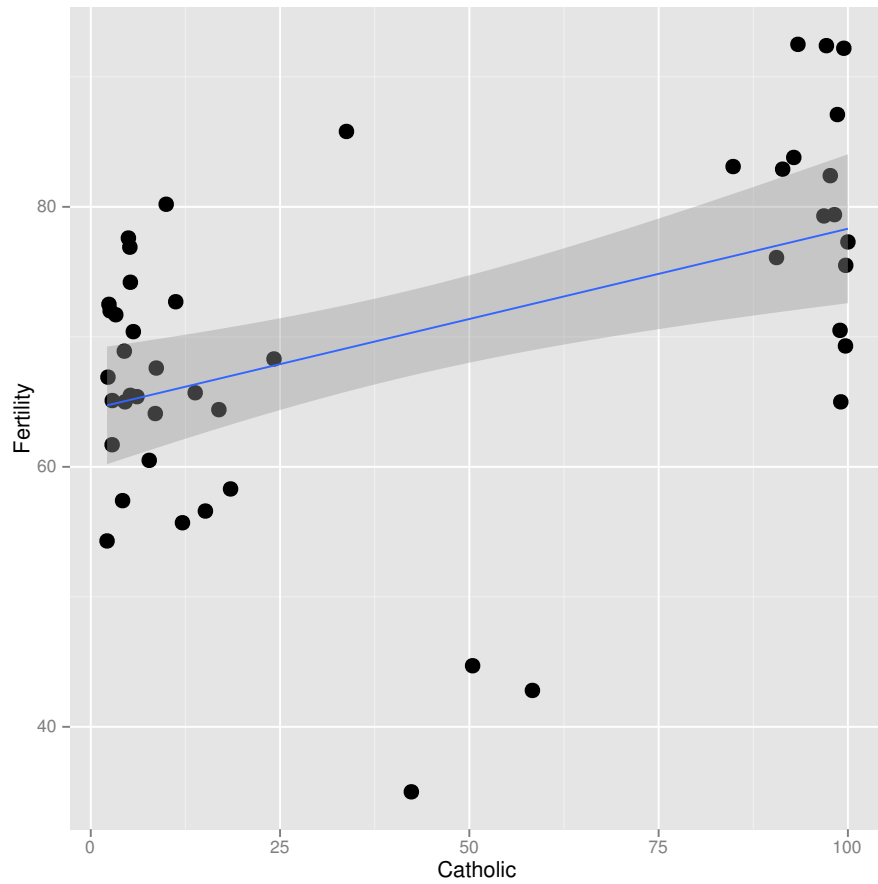
1. Диаграмма рассеяния

```
h <- swiss
ggplot(h, aes(x=Catholic, y=Fertility)) + geom_point(size=4) + geom_density2d()
```



2. Диаграмма рассеяния с добавкой линии регрессии или линии loess

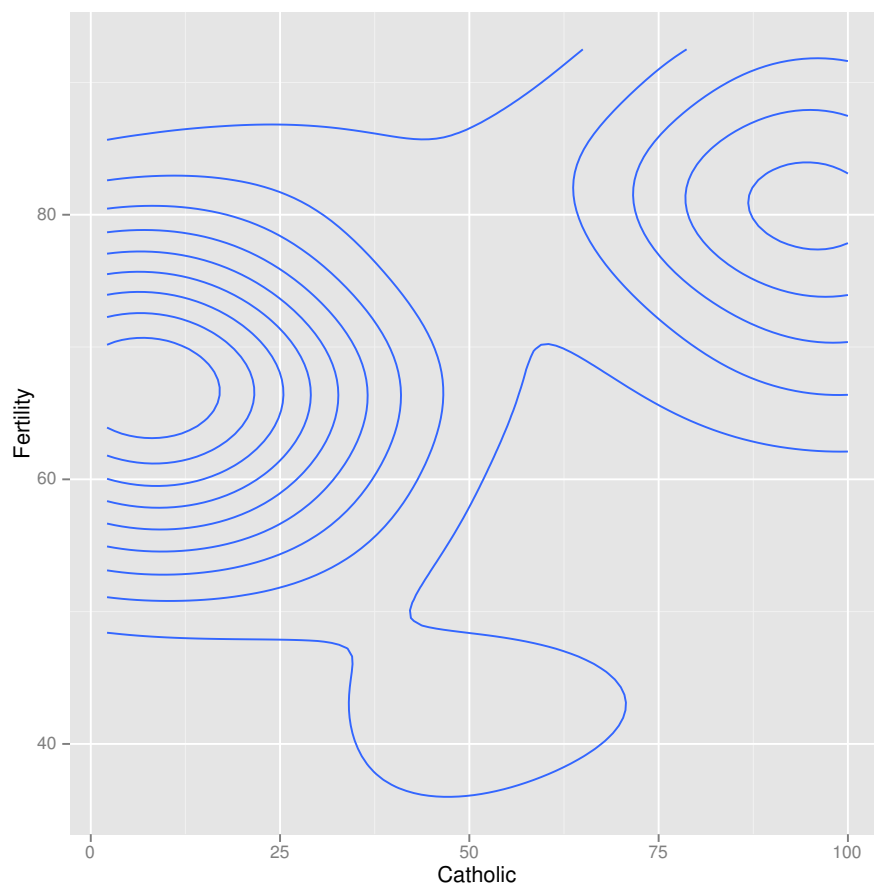
```
h <- swiss
ggplot(h,aes(x=Catholic,y=Fertility))+geom_point(size=4)+
  stat_smooth(method="lm")
```



Можно добавить сглаживание `method="loess"`. Если нужна только линия без стандартных ошибок, то добавляют опцию `se=FALSE`.

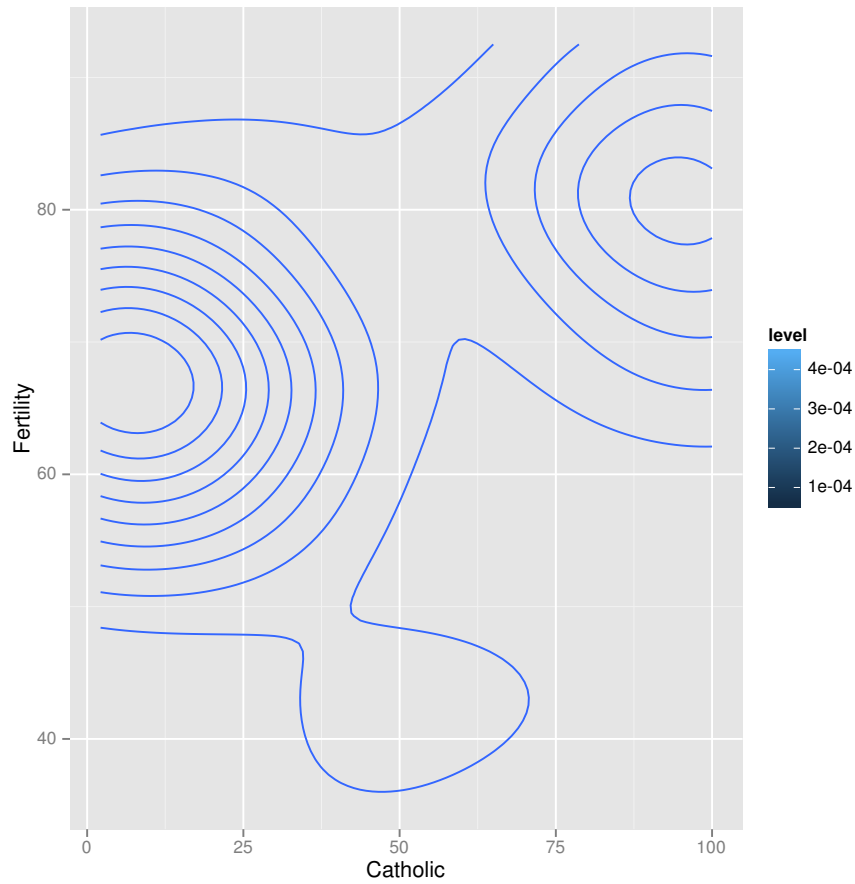
3. Линии уровня двумерной функции плотности

```
h <- swiss
ggplot(h,aes(x=Catholic,y=Fertility))+geom_density2d()
```



Вариант с закраской

```
h <- swiss  
ggplot(h, aes(x=Catholic, y=Fertility)) + geom_density2d(aes(fill=..level..))
```

Нюансы:

1. Слишком много наблюдений
2. Мало значений у количественной переменной

Оформление:

Подписи к осям

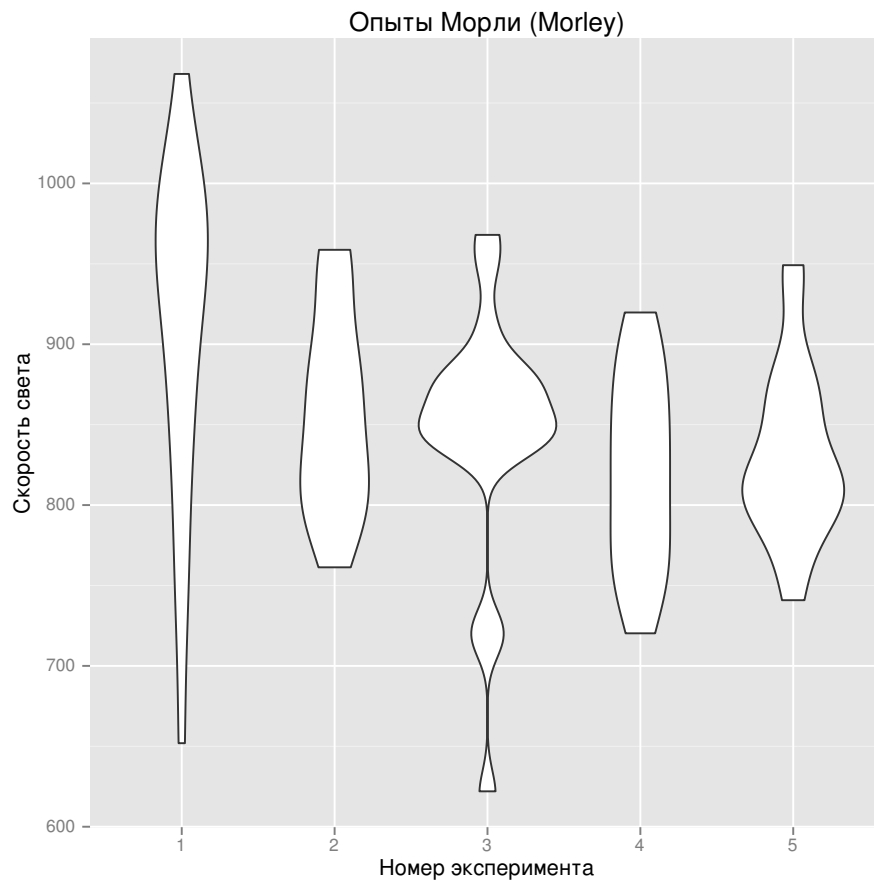
Количественная - качественная

1. Несколько беанплов

```
h <- morley
```

```
h$Expt <- factor(h$Expt)
```

```
ggplot(h, aes(x=Expt, y=Speed))+geom_violin()+xlab("Номер эксперимента")+ylab("Скорость света")
```



Если категорий мало:

1. Наложенные гистограммы
2. Чередующиеся гистограммы
3. Stacked гистограммы
4. Наложенные функции плотности
- 4.5. Сглаженные условные вероятности
5. Гистограммы на отдельных гранях (срезах, facet) графика

Накладываем на них оценку функции плотности или гипотетическую

плотность

Качественная - качественная

1. Мозаичный график
2. Stacked histogram

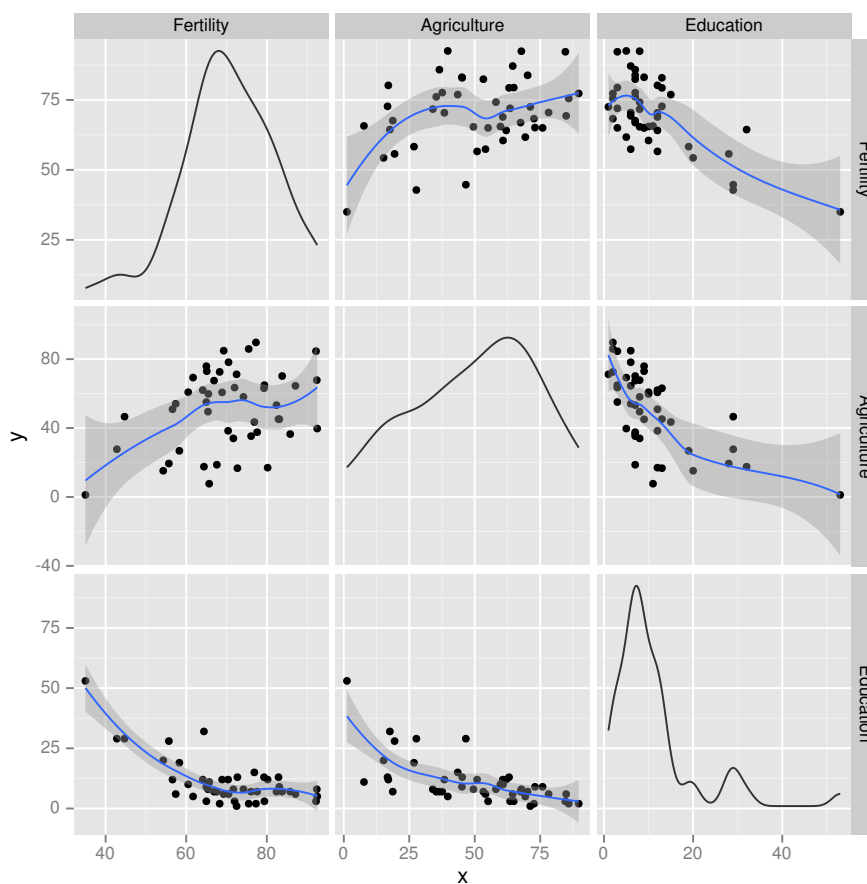
3 Много переменных

Количественная, количественная, качественная

1. Диаграмма рассеяния с разным типом (цветом) точек
 2. Несколько диаграмм рассеяния на границах графика
- Много количественных
1. plotmatrix

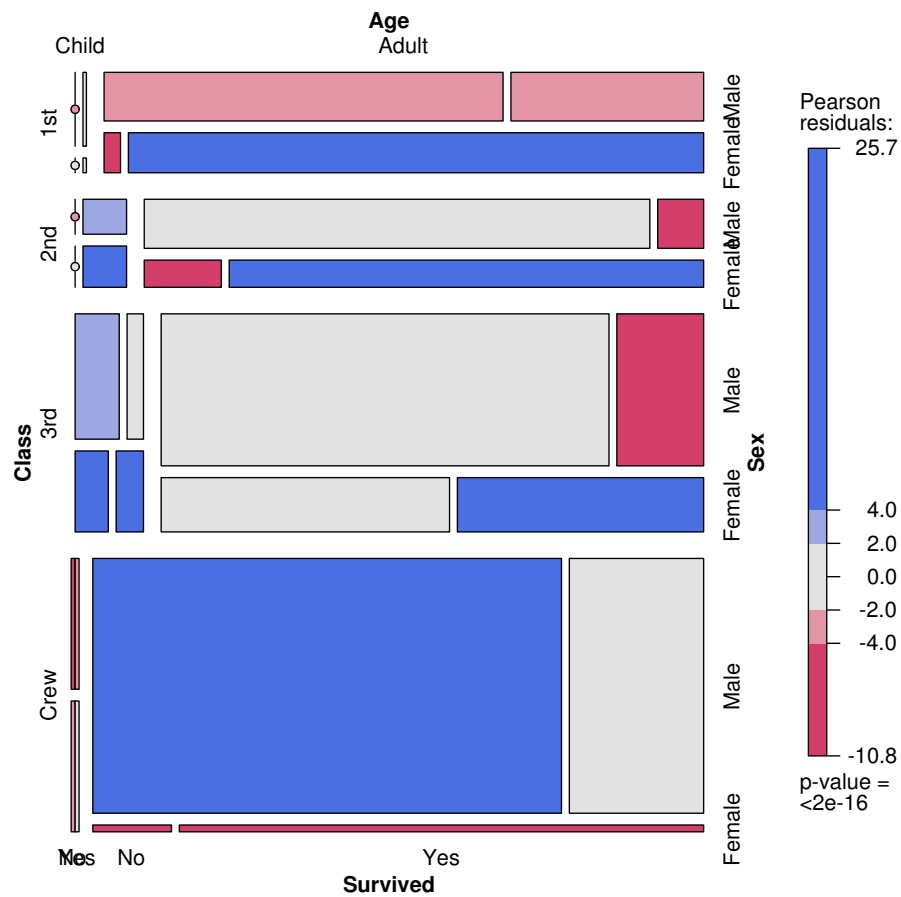
```
h <- swiss
plotmatrix(h[,c("Fertility", "Agriculture", "Education")]) + stat_smooth(method="loess")

## This function is deprecated. For a replacement, see the ggpairs
function in the GGally package. (Deprecated; last used in version 0.9.2)
```



- Много качественных
1. мозаичный график

```
library(vcd)
titan <- Titanic
mosaic(~Class+Age+Sex+Survived, data=titan, shade=TRUE, legend=TRUE)
```



2. Грани графика + Stacked histogram

Размещение нескольких графиков рядом.

1. Диаграмма рассеяния + гистограммы по краям

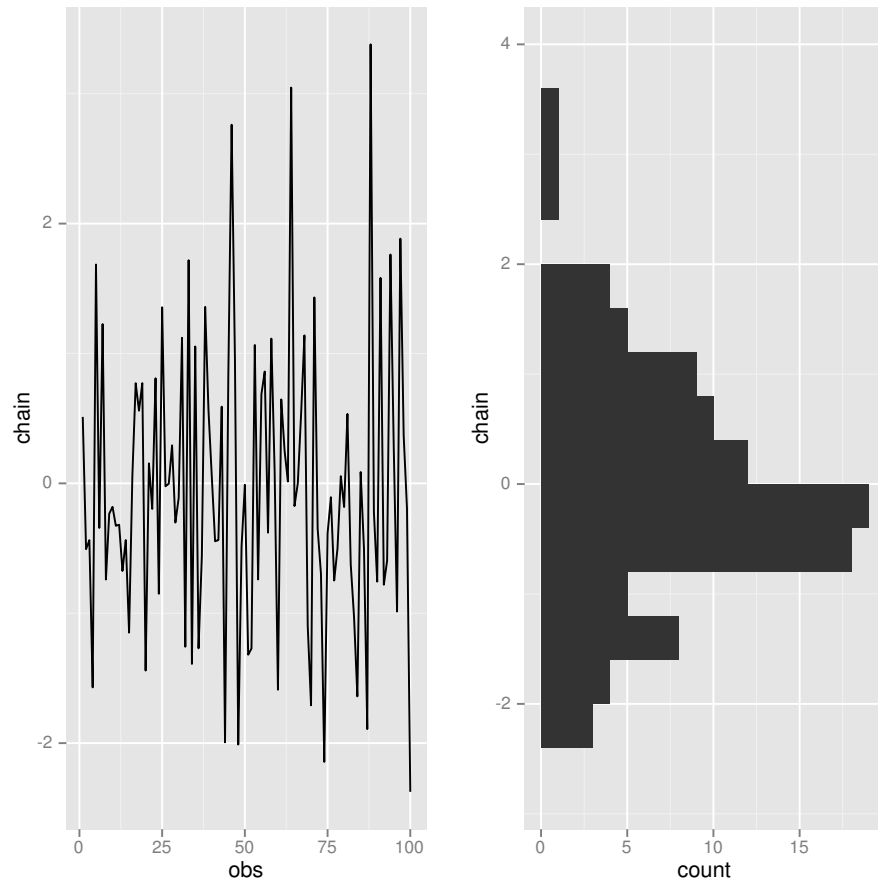
2. Временной ряд + гистограмма сбоку (для mcmc)

К примеру мы сгенерировали марковскую цепь и хотим посмотреть, а не сошлась ли она часом к нужному нам распределению...

```
library(gridExtra)
h <- data.frame(chain = rnorm(100), obs=1:100)

p1 <- ggplot(h, aes(x=obs, y=chain))+geom_line()
p2 <- ggplot(h, aes(x=chain))+geom_histogram(binwidth=0.4)+coord_flip()

p <- arrangeGrob(p1, p2, nrow=1)
p
```



4 Карты

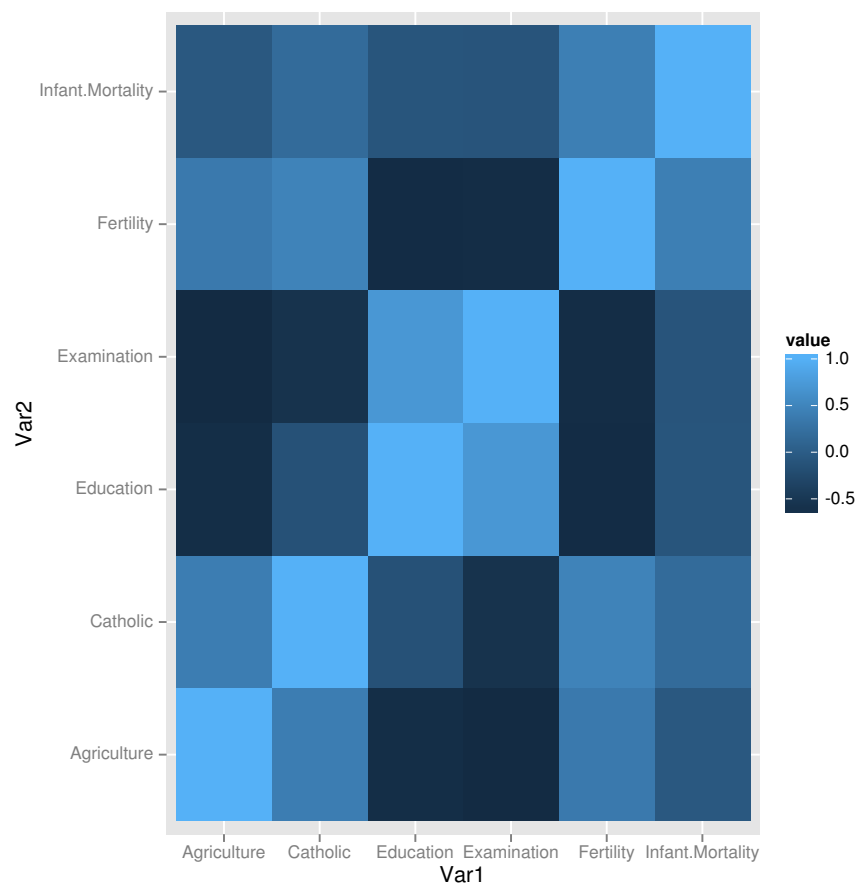
5 Часто встречаются

1. Визуализация корреляционной матрицы

```
library(reshape2)

h <- swiss
cor.mat <- cor(h)
cor.mat2 <- melt(cor.mat)

ggplot(cor.mat2, aes(x=Var1, y=Var2, fill=value)) + geom_raster()
```



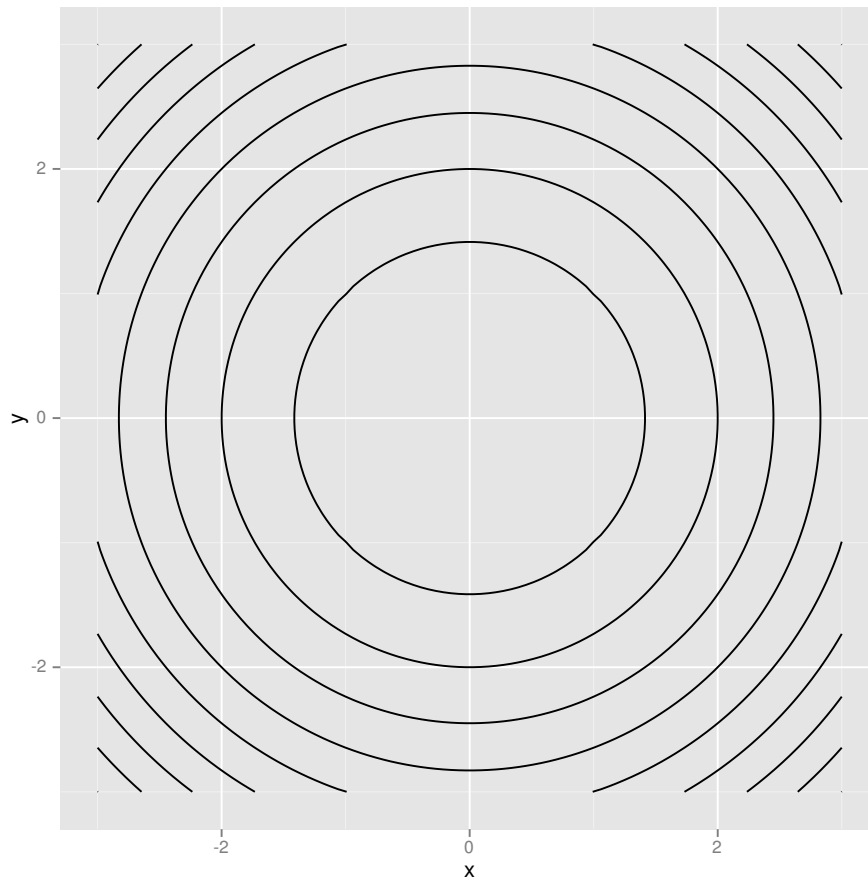
2. Построение линий уровня функции двух переменных

```
my_f <- function(x,y) x^2+y^2

x <- seq(-3,3,length.out=100)
y <- seq(-3,3,length.out=100)
h <- data.frame(x=rep(x,times=100),y=rep(y,each=100))

h$z <- my_f(h$x,h$y)

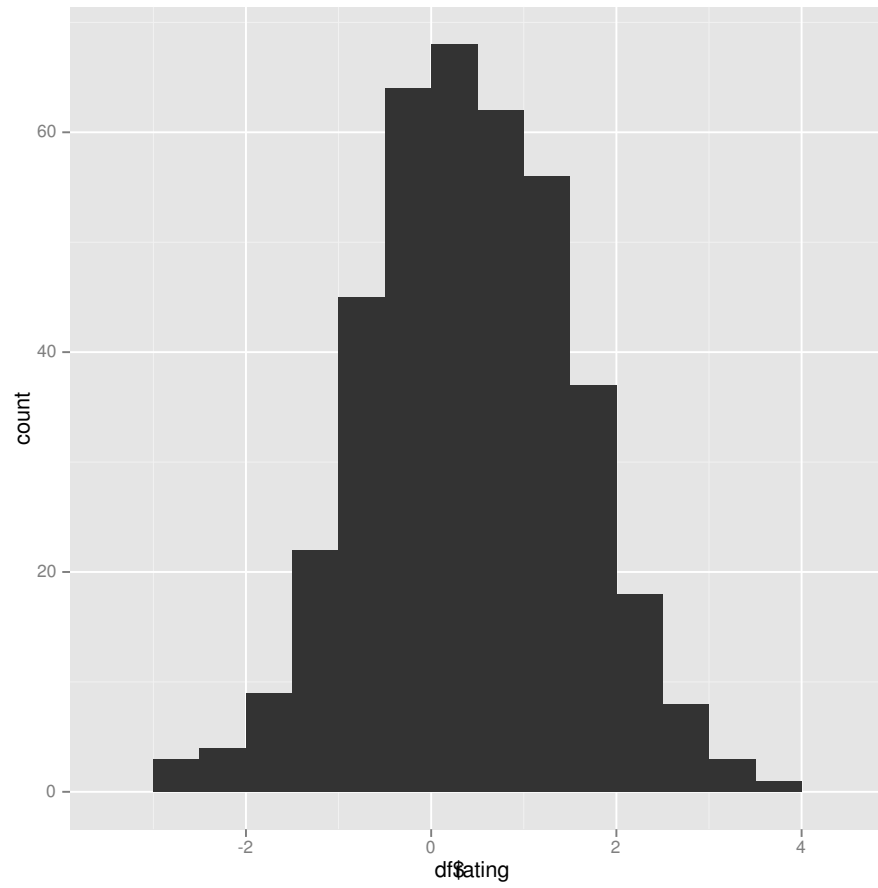
ggplot(h, aes(x=x, y=y, z=z)) + stat_contour()
```



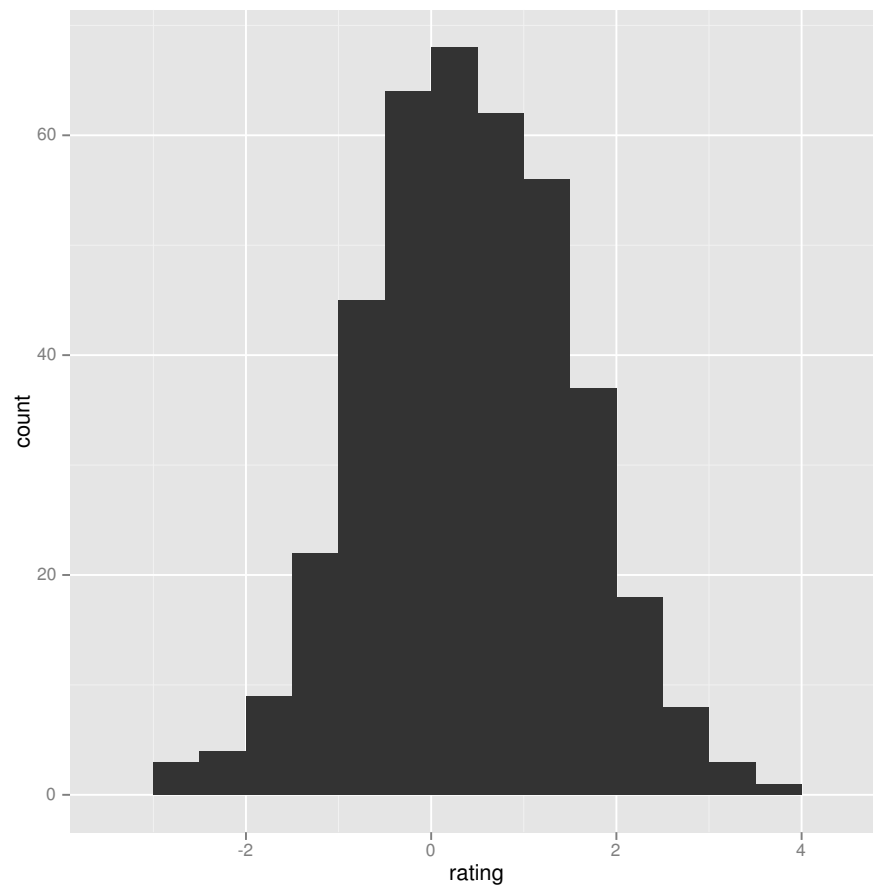
<http://wiki.stdout.org/rcookbook/Graphs/>

```
set.seed(1234)
df <- data.frame(cond = factor( rep(c("A","B"), each=200) ),
                  rating = c(rnorm(200),rnorm(200, mean=.8)))
```

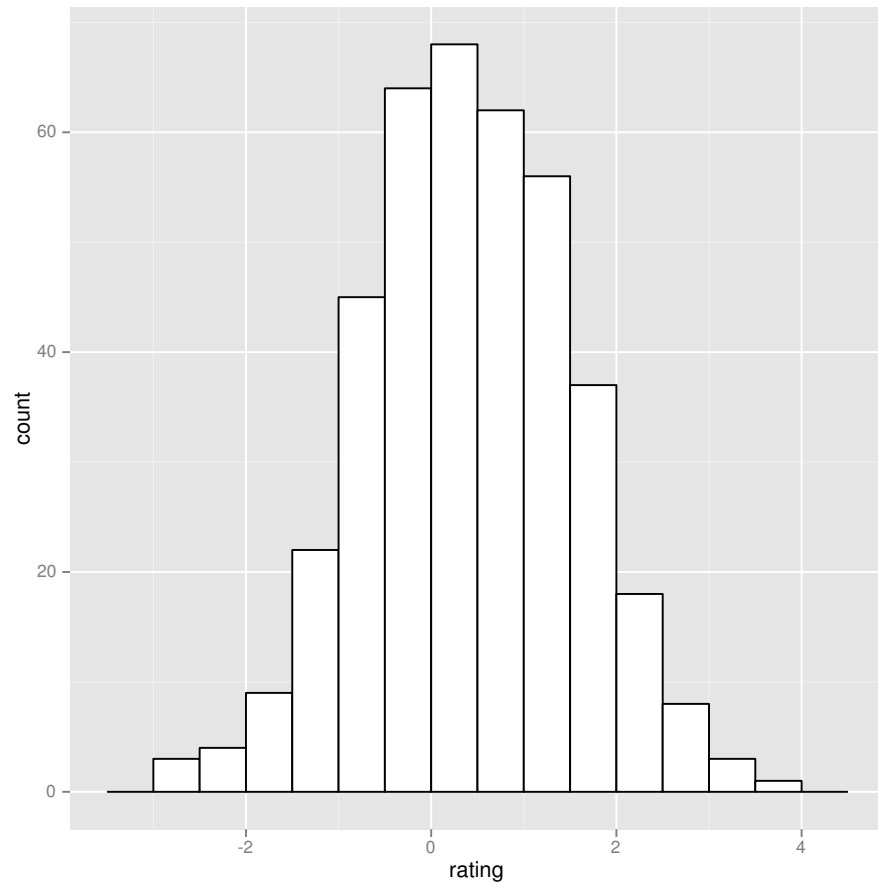
```
qplot(df$rating, binwidth=.5)
```



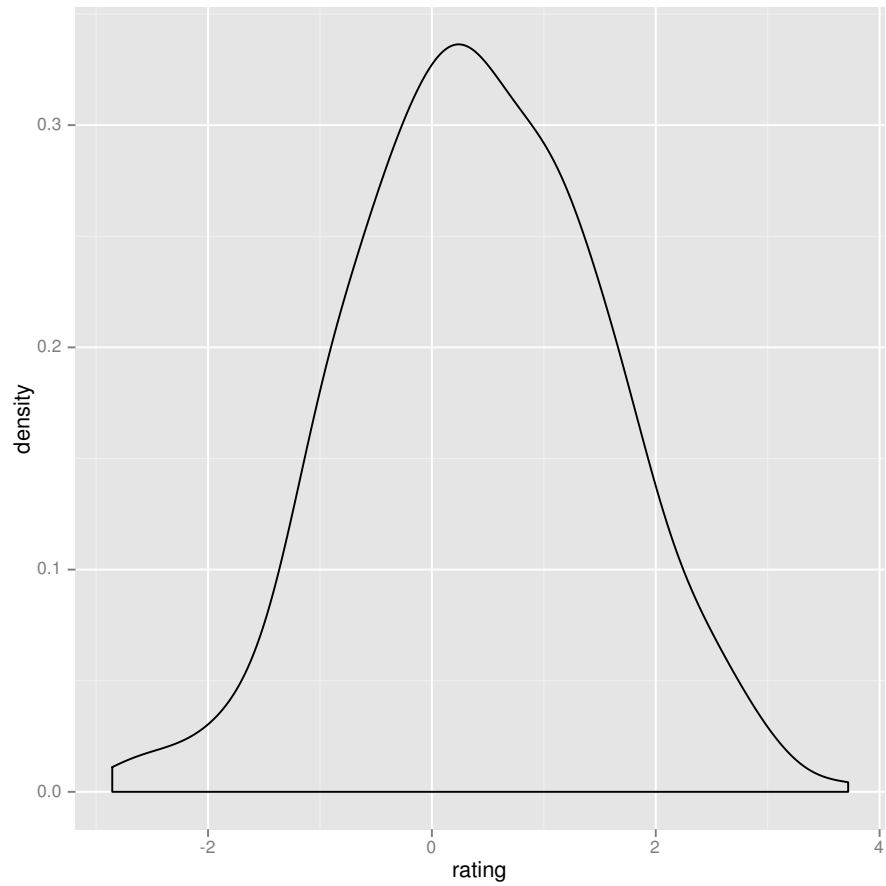
```
ggplot(df, aes(x=rating)) + geom_histogram(binwidth=.5)
```

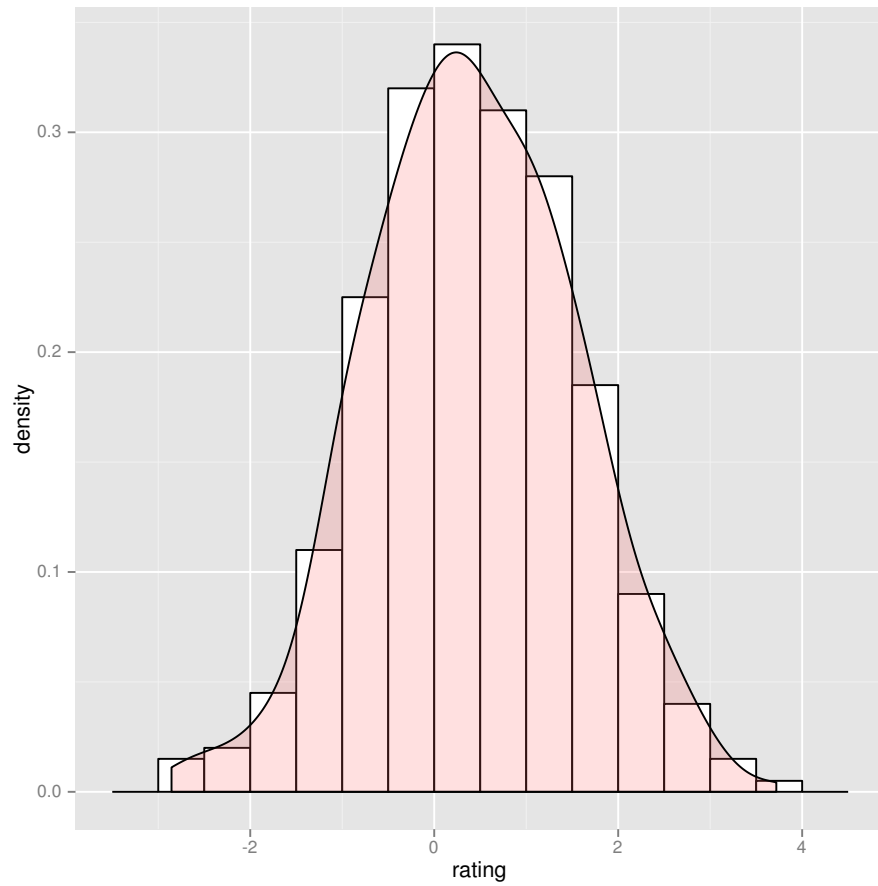
```
# Draw with black outline, white fill  
ggplot(df, aes(x=rating)) + geom_histogram(binwidth=.5, colour="black", fill="white")
```



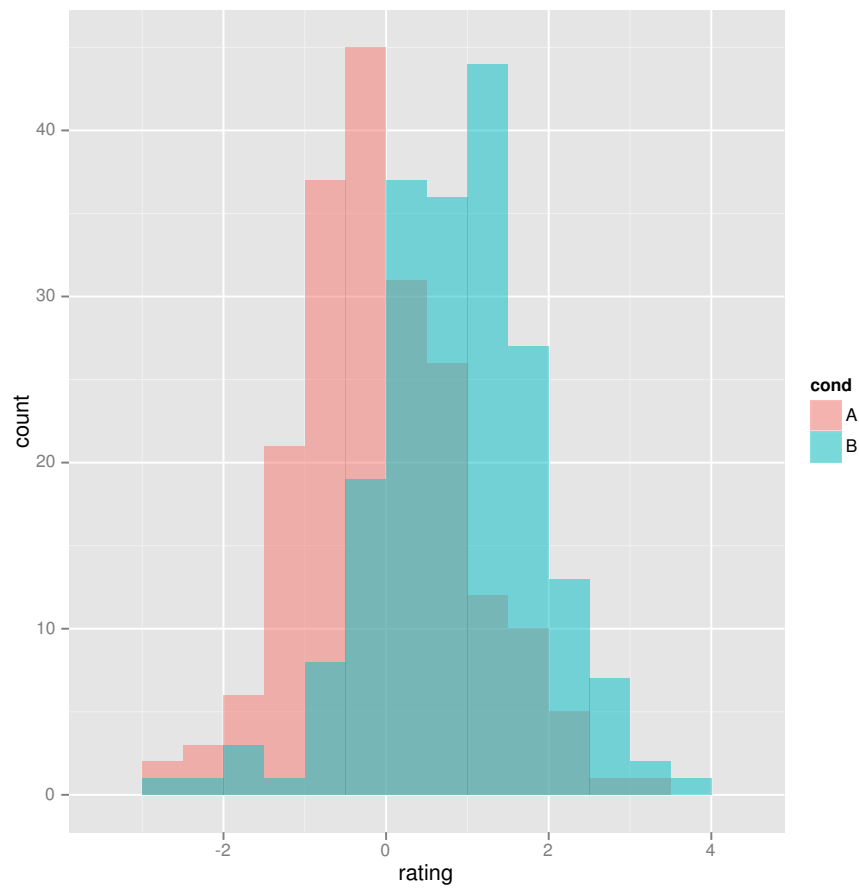
```
# Density curve  
ggplot(df, aes(x=rating)) + geom_density()
```



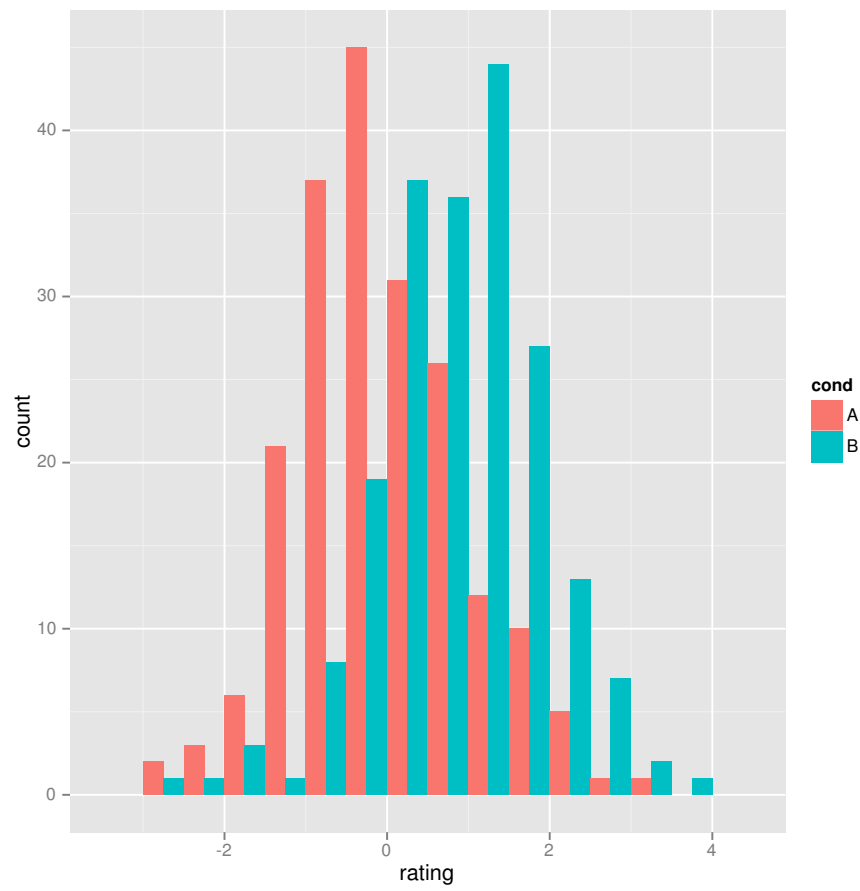
```
# Histogram overlaid with kernel density curve
ggplot(df, aes(x=rating)) +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of count on y-axis
                 binwidth=.5,
                 colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666") # Overlay with transparent density plot
```



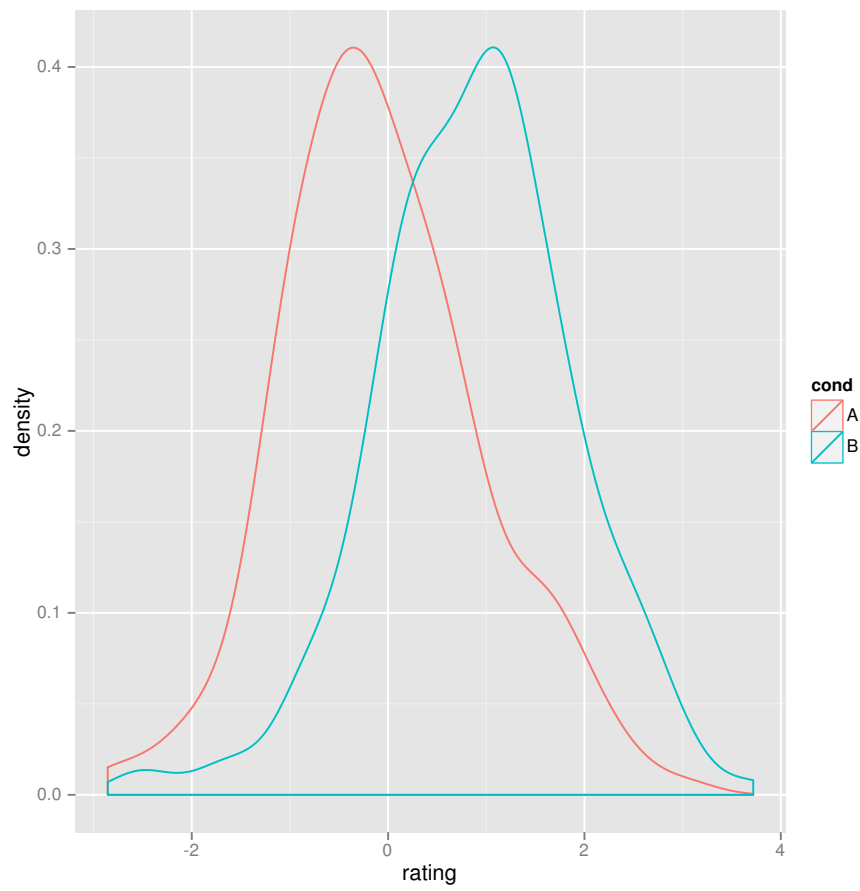
```
# Overlaid histograms  
ggplot(df, aes(x=rating, fill=cond)) + geom_histogram(binwidth=.5, alpha=.5, position="ident
```



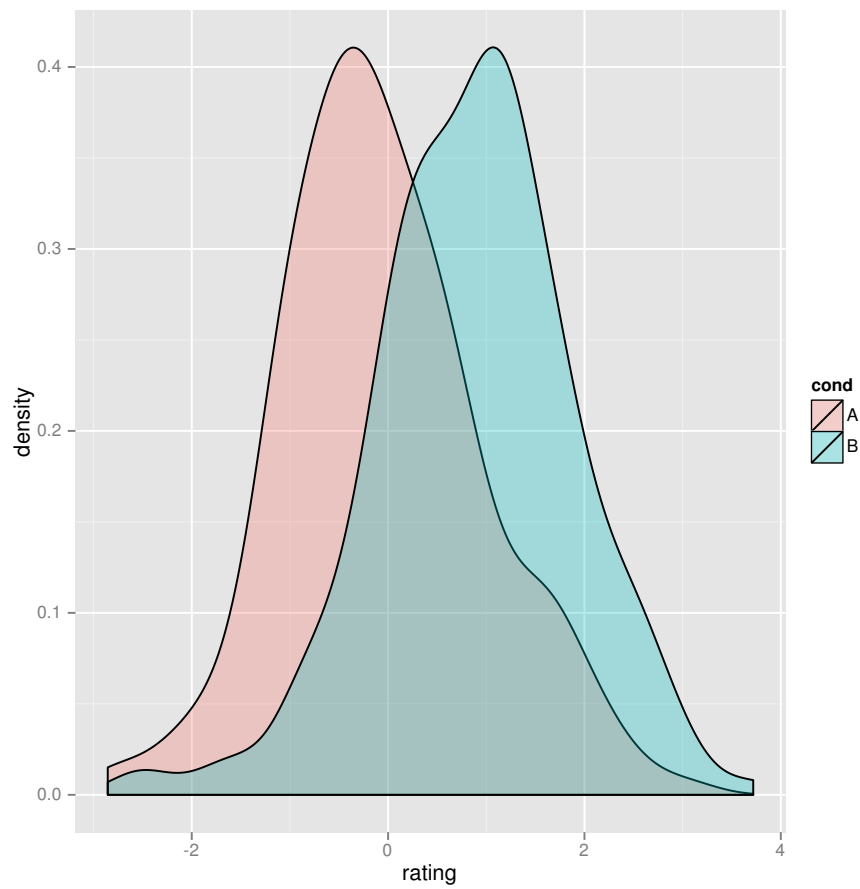
```
# Interleaved histograms  
ggplot(df, aes(x=rating, fill=cond)) + geom_histogram(binwidth=.5, position="dodge")
```



```
# Density plots  
ggplot(df, aes(x=rating, colour=cond)) + geom_density()
```

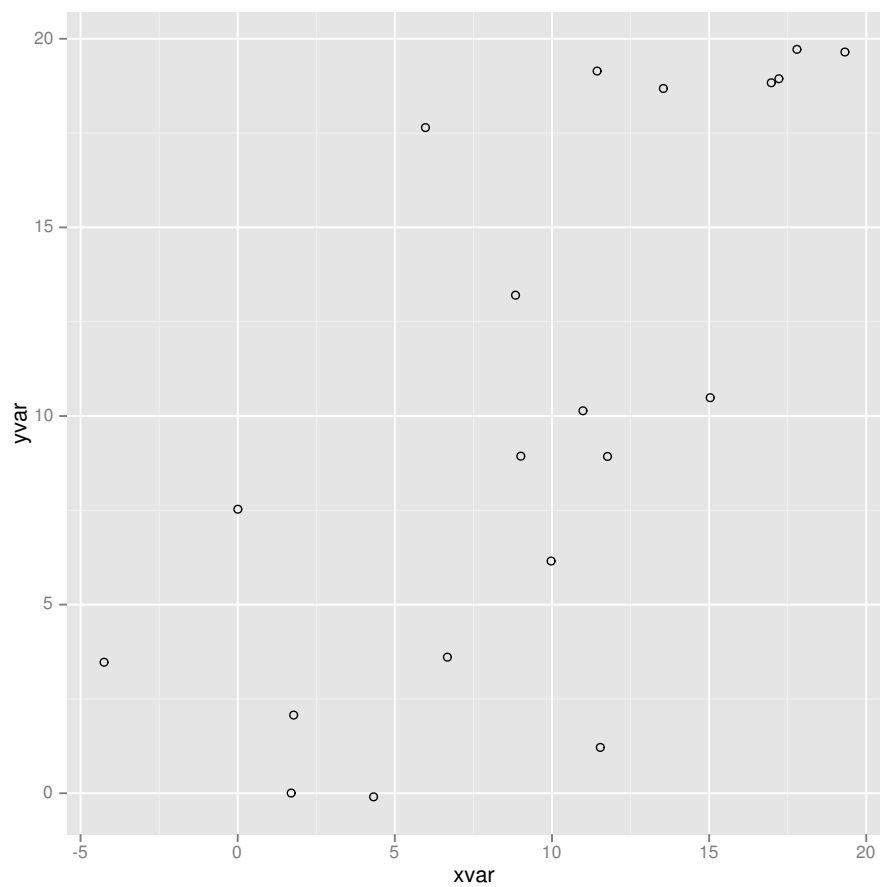


```
# Density plots with semi-transparent fill  
ggplot(df, aes(x=rating, fill=cond)) + geom_density(alpha=.3)
```

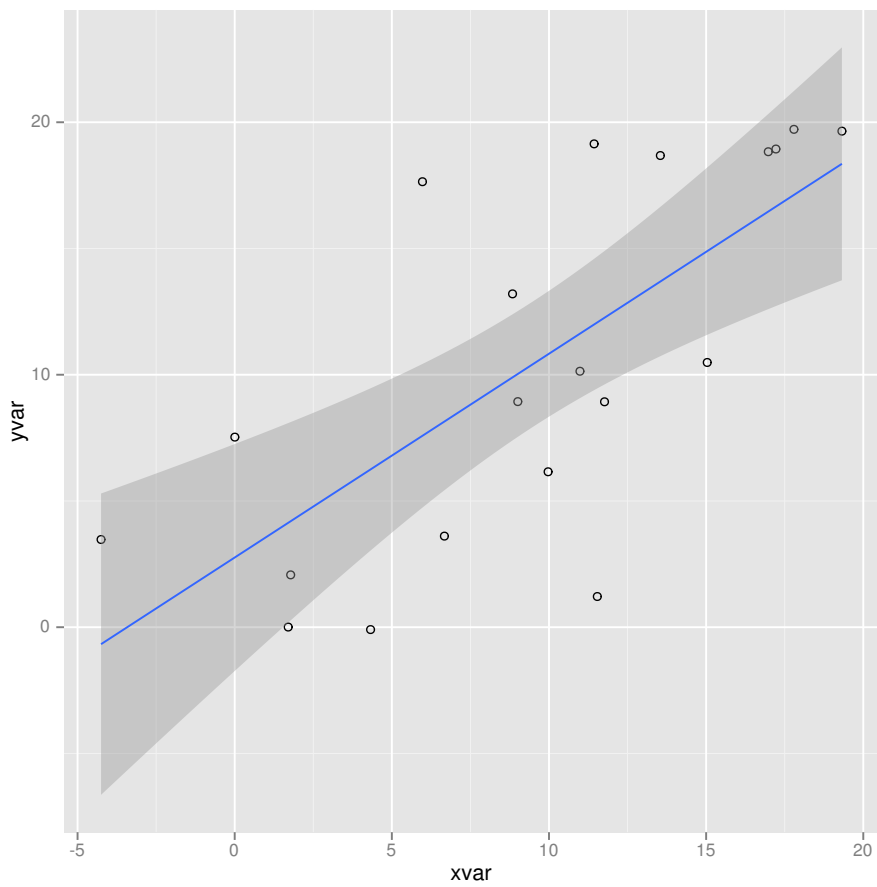


```
set.seed(955)
# Make some noisily increasing data
dat <- data.frame(cond = rep(c("A", "B"), each=10),
                  xvar = 1:20 + rnorm(20,sd=3),
                  yvar = 1:20 + rnorm(20,sd=3))
```

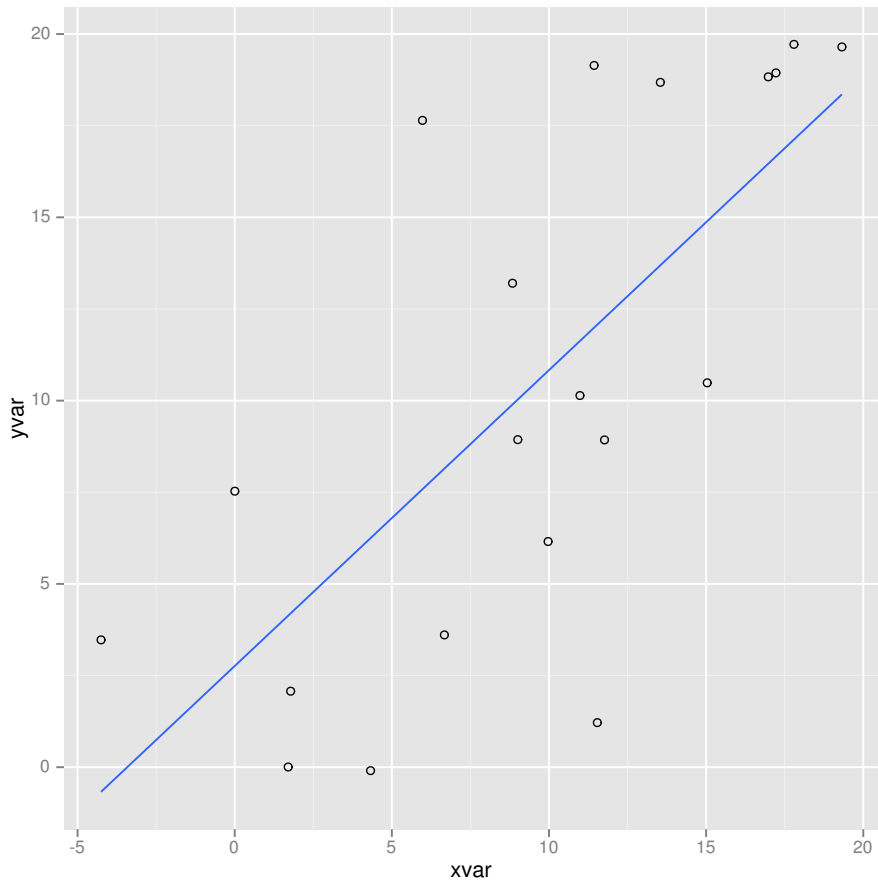
```
ggplot(dat, aes(x=xvar, y=yvar)) +
  geom_point(shape=1)      # Use hollow circles
```

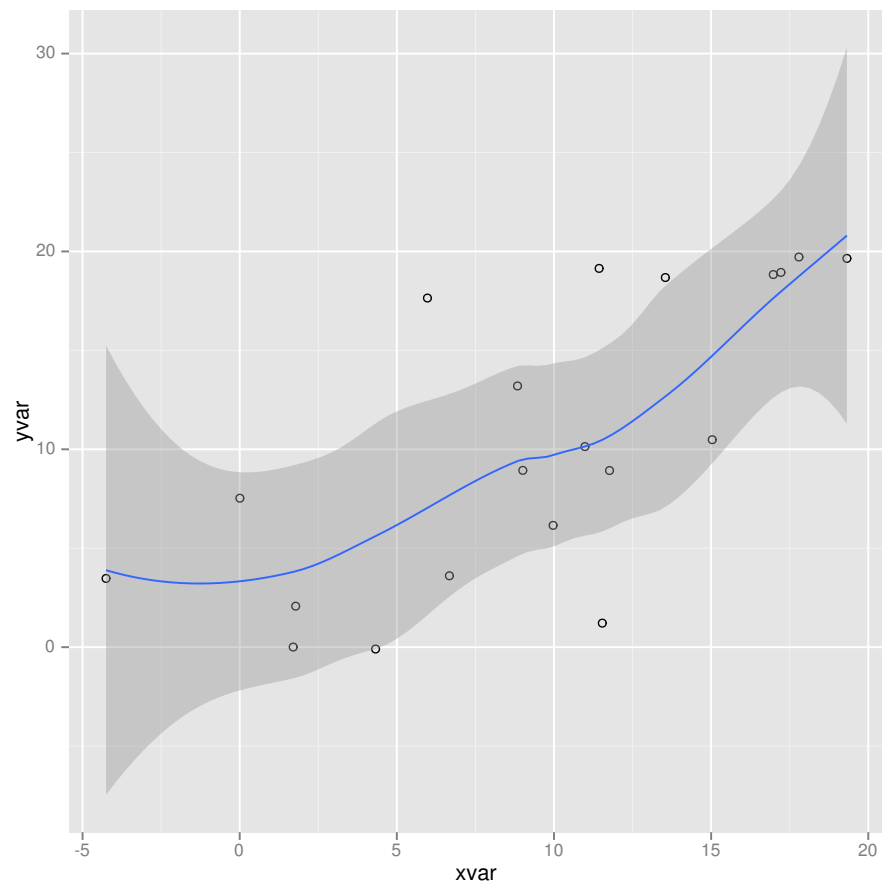
```
ggplot(dat, aes(x=xvar, y=yvar)) +  
  geom_point(shape=1) +      # Use hollow circles  
  geom_smooth(method=lm)     # Add linear regression line
```



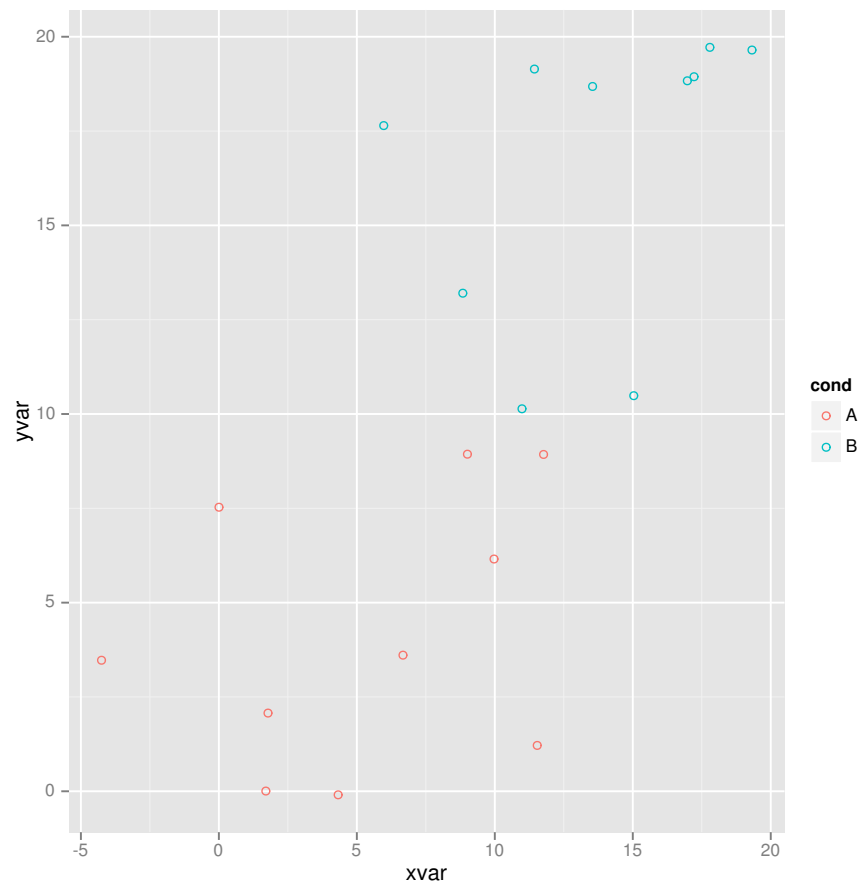
```
# (by default includes 95% confidence region)  
ggplot(dat, aes(x=xvar, y=yvar)) +  
  geom_point(shape=1) + # Use hollow circles  
  geom_smooth(method=lm, # Add linear regression line  
              se=FALSE) # Don't add shaded confidence region
```



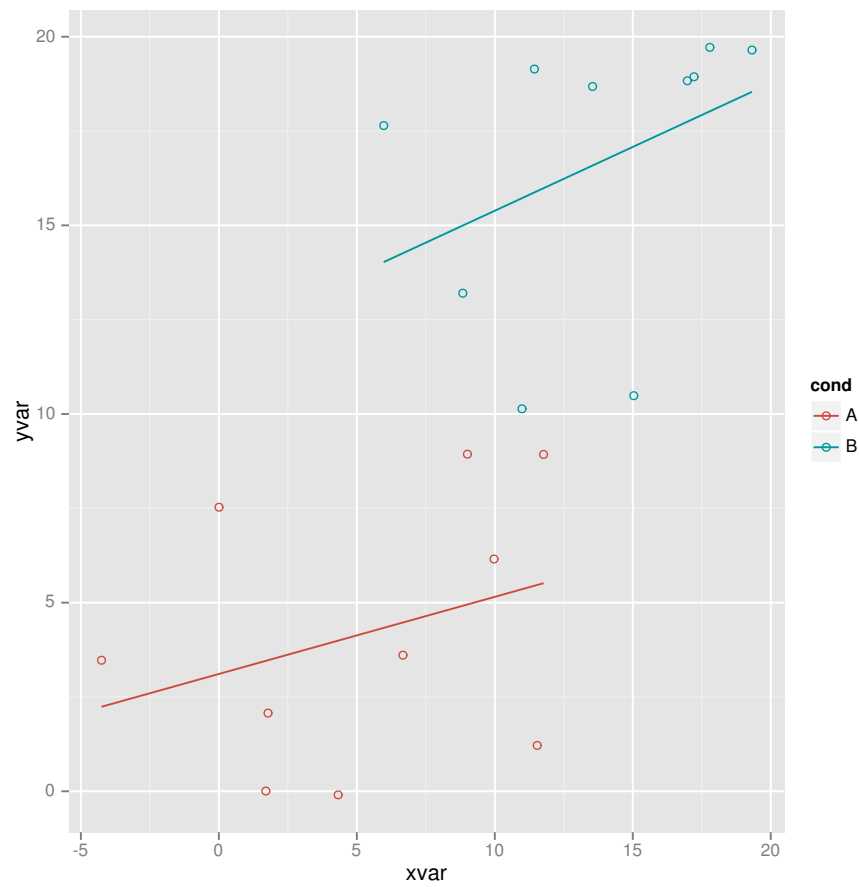
```
ggplot(dat, aes(x=xvar, y=yvar)) +  
  geom_point(shape=1) +      # Use hollow circles  
  geom_smooth()              # Add a loess smoothed fit curve with confidence region  
  
## geom_smooth: method="auto" and size of largest group is <1000, so  
## using loess. Use 'method = x' to change the smoothing method.
```



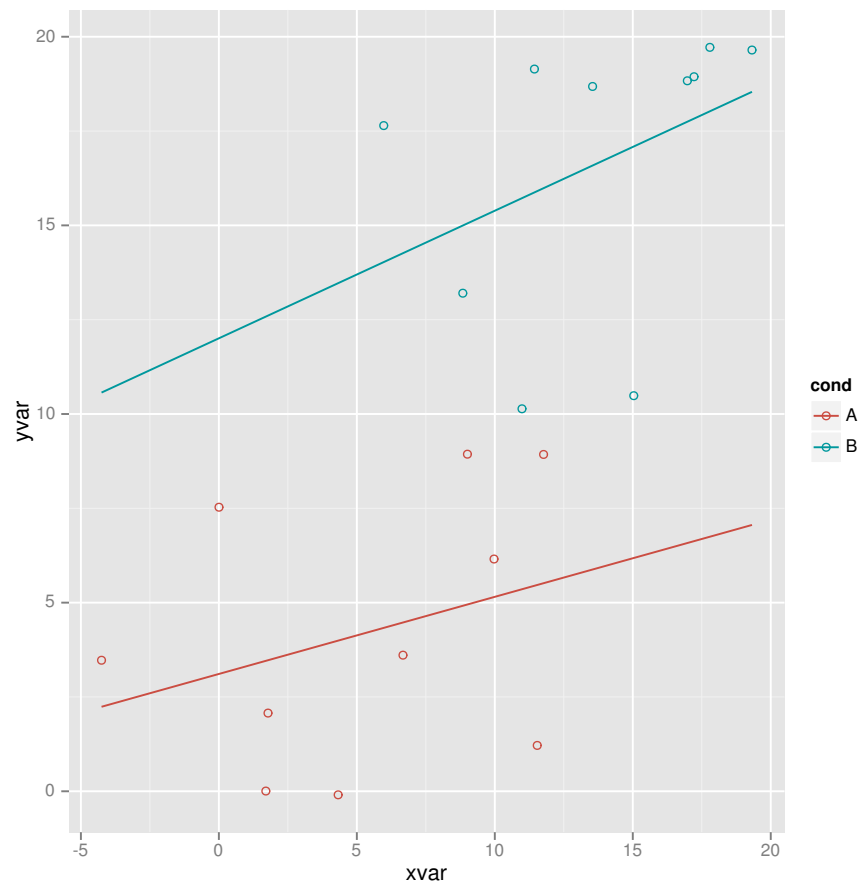
```
# Set color by cond  
ggplot(dat, aes(x=xvar, y=yvar, color=cond)) + geom_point(shape=1)
```



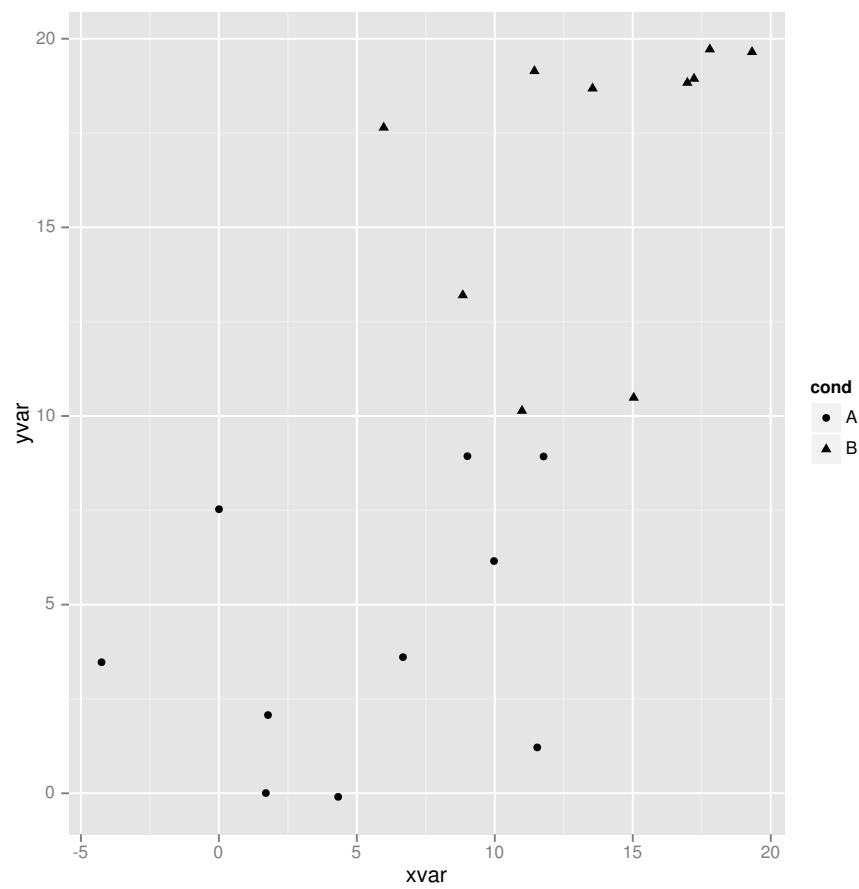
```
# Same, but with different colors and add regression lines
ggplot(dat, aes(x=xvar, y=yvar, color=cond)) + geom_point(shape=1) +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method=lm,   # Add linear regression lines
              se=FALSE)    # Don't add shaded confidence region
```



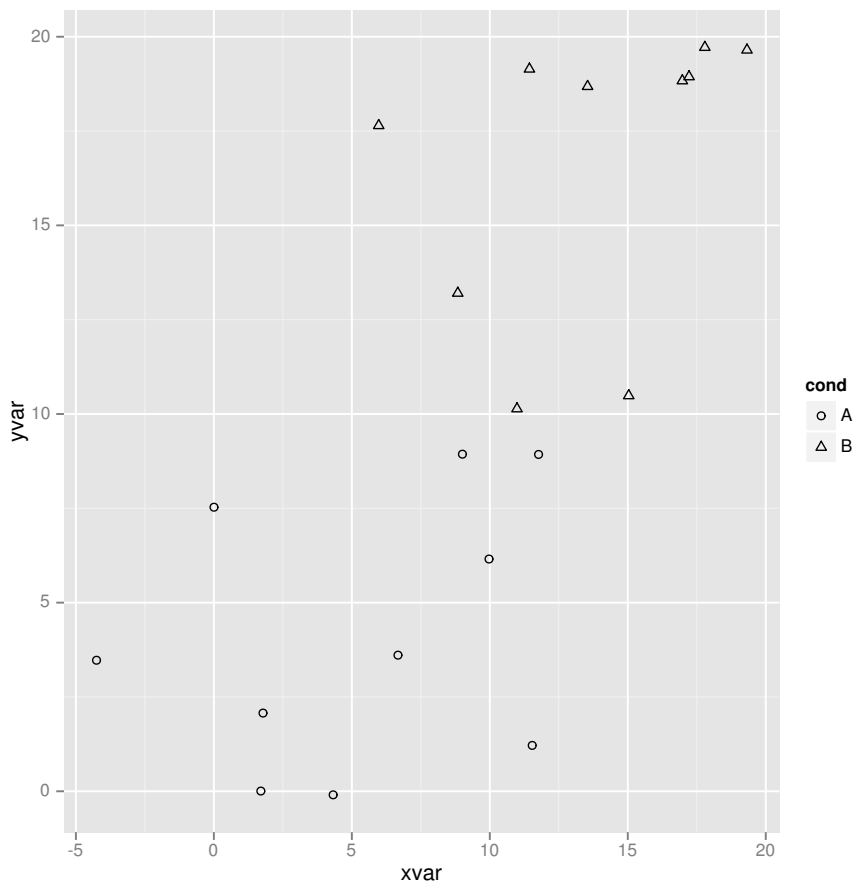
```
# Extend the regression lines beyond the domain of the data
ggplot(dat, aes(x=xvar, y=yvar, color=cond)) + geom_point(shape=1) +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method=lm, # Add linear regression lines
              se=FALSE, # Don't add shaded confidence region
              fullrange=T) # Extend regression lines
```



```
# Set shape by cond
ggplot(dat, aes(x=xvar, y=yvar, shape=cond)) + geom_point()
```



```
# Same, but with different shapes
ggplot(dat, aes(x=xvar, y=yvar, shape=cond)) + geom_point() +
  scale_shape_manual(values=c(1,2)) # Use a hollow circle and triangle
```

```
# This example uses the ChickWeight dataset, which comes with ggplot2
# First plot
p1 <-
  ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet, group=Chick)) +
  geom_line() +
  ggtitle("Growth curve for individual chicks")

# Second plot
p2 <-
  ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet)) +
  geom_point(alpha=.3) +
  geom_smooth(alpha=.2, size=1) +
  ggtitle("Fitted growth curve per diet")

# Third plot
```

```

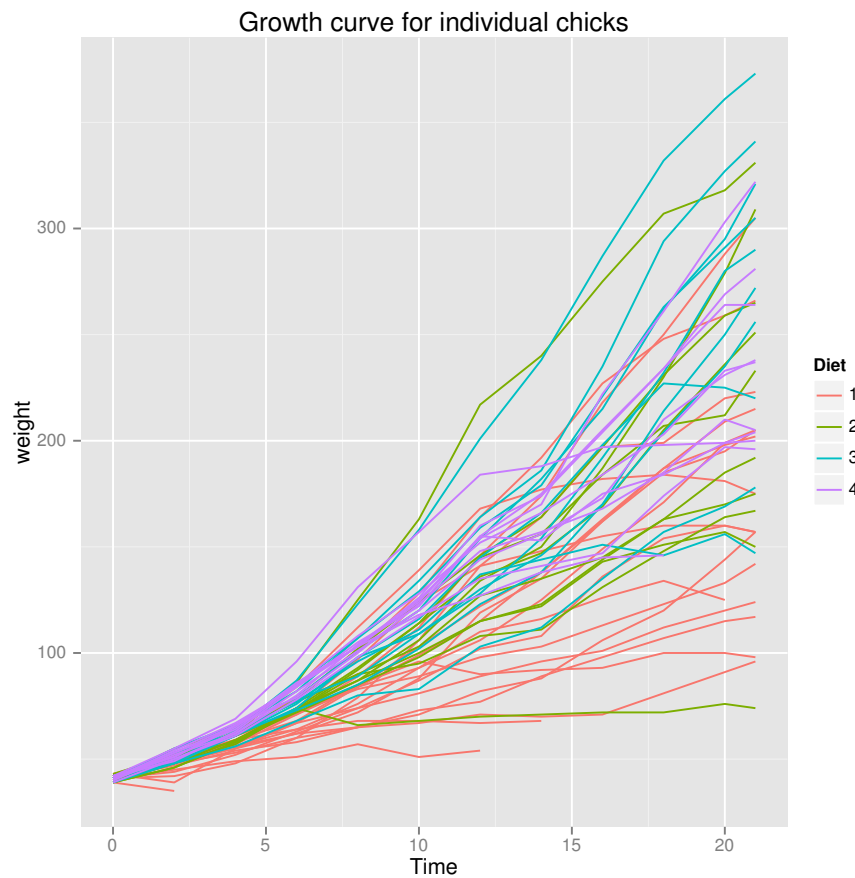
p3 <-
  ggplot(subset(ChickWeight, Time==21), aes(x=weight, colour=Diet)) +
  geom_density() +
  ggtitle("Final weight, by diet")

# Fourth plot
p4 <-
  ggplot(subset(ChickWeight, Time==21), aes(x=weight, fill=Diet)) +
  geom_histogram(colour="black", binwidth=50) +
  facet_grid(Diet ~ .) +
  ggtitle("Final weight, by diet") +
  theme(legend.position="none") # No legend (redundant in this graph)

par(mfrow=c(2,2)) # arrange in 2 rows and 2 cols

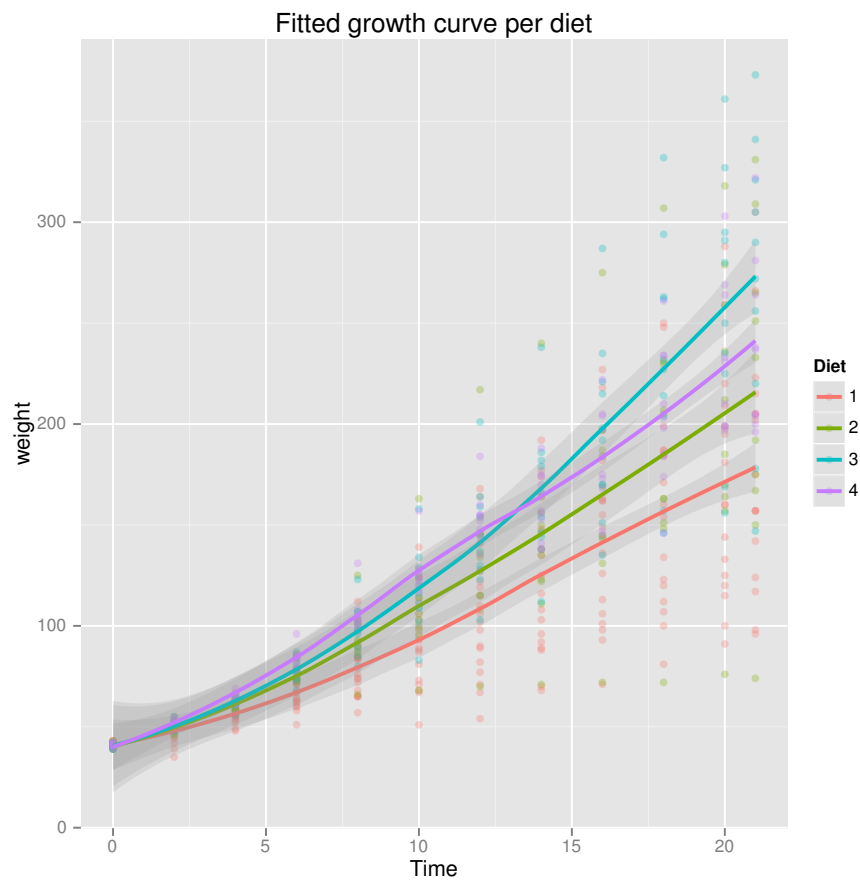
p1

```

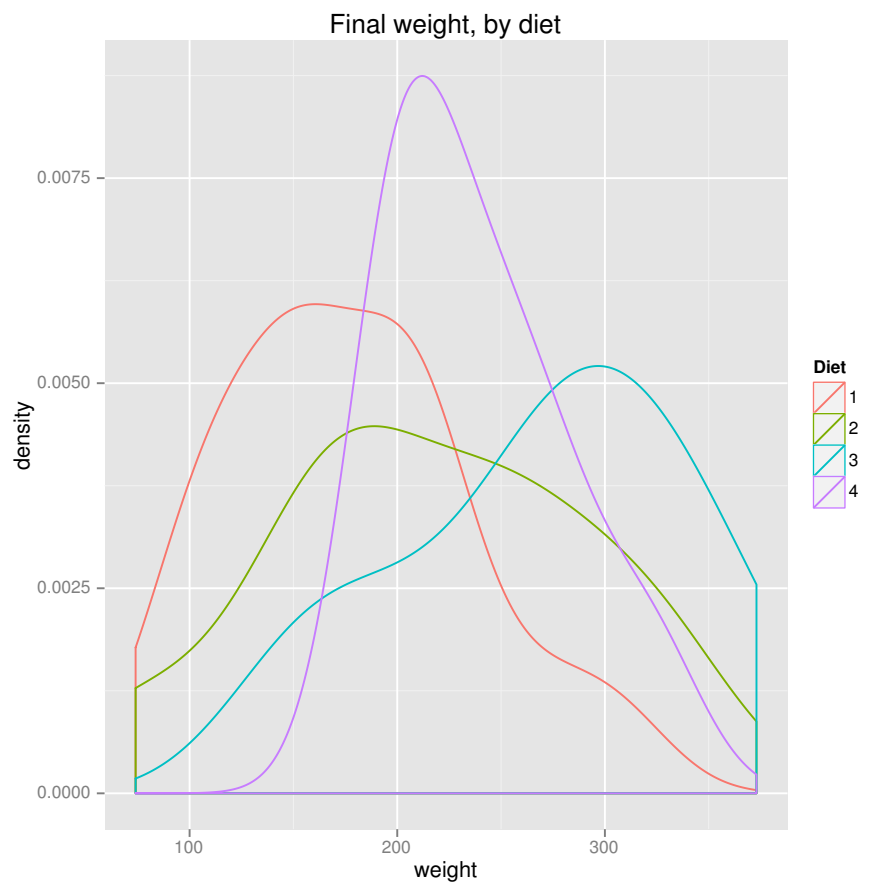


p2

```
## geom_smooth: method="auto" and size of largest group is <1000, so  
using loess. Use 'method = x' to change the smoothing method.
```



p3



p4

