Biostatistics Lecture2

Ruibin Xi

2/15/2020

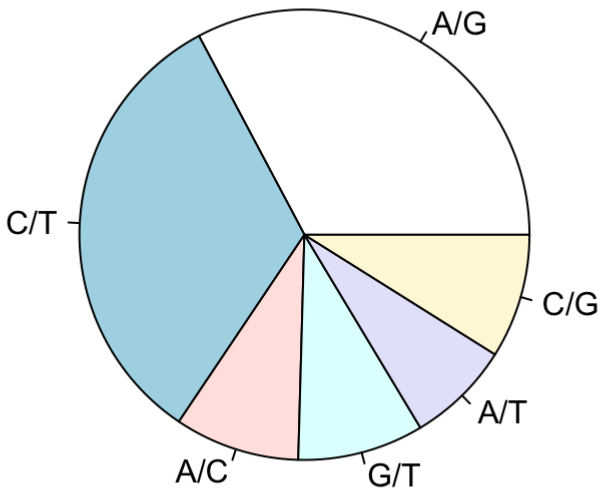
Visualize the distribution of categorical data.

We consider the SNP data. Data are from Zhao and Boerwinkle (http://genome.cshlp.org/content/12/11/1679.full (http://genome.cshlp.org/content/12/11/1679.full)). There are six possible types of SNPs. AG, GT are called transitions, and other types are called transversions.

```
AG = 844427
CT = 845441
AC = 231506
GT = 233387
AT = 192285
CG = 229857
snp = c(AG,CT,AC,GT,AT,CG)
names.snp = c("A/G","C/T","A/C","G/T","A/T","C/G")
names(snp) = names.snp
```

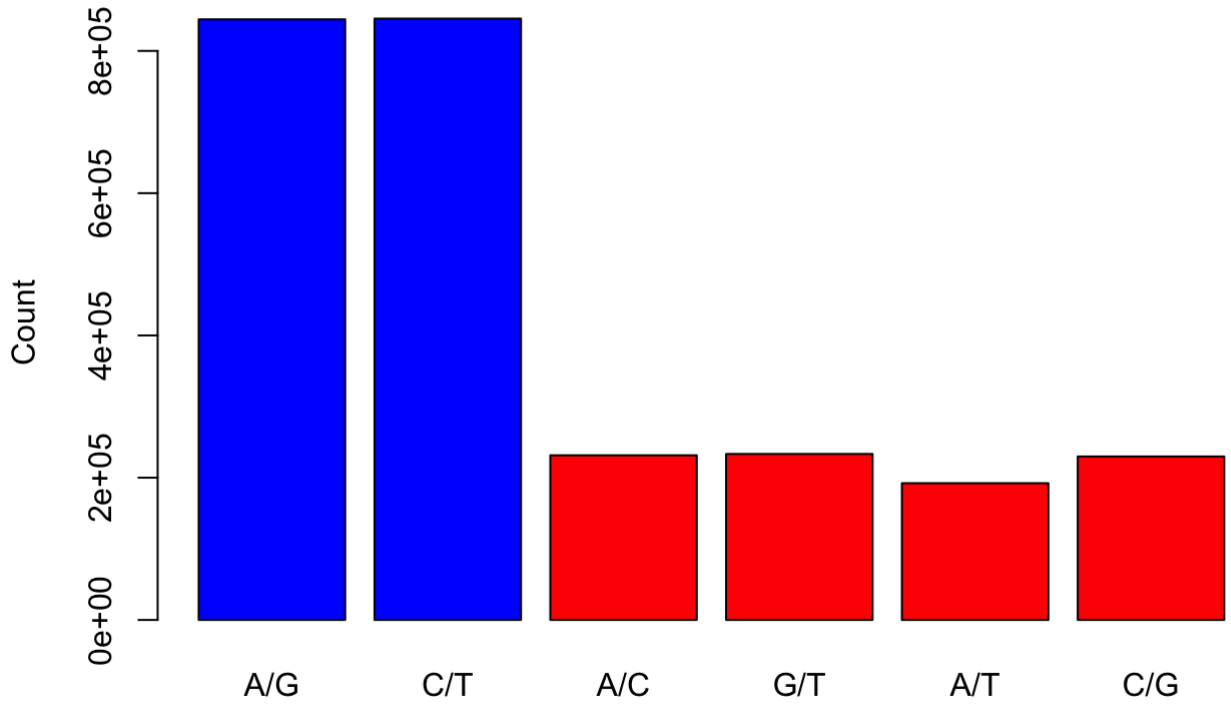
We first use the pie chart to show the distribution.

```
pie(snp)
```



Show the count of each type of SNP using boxplot

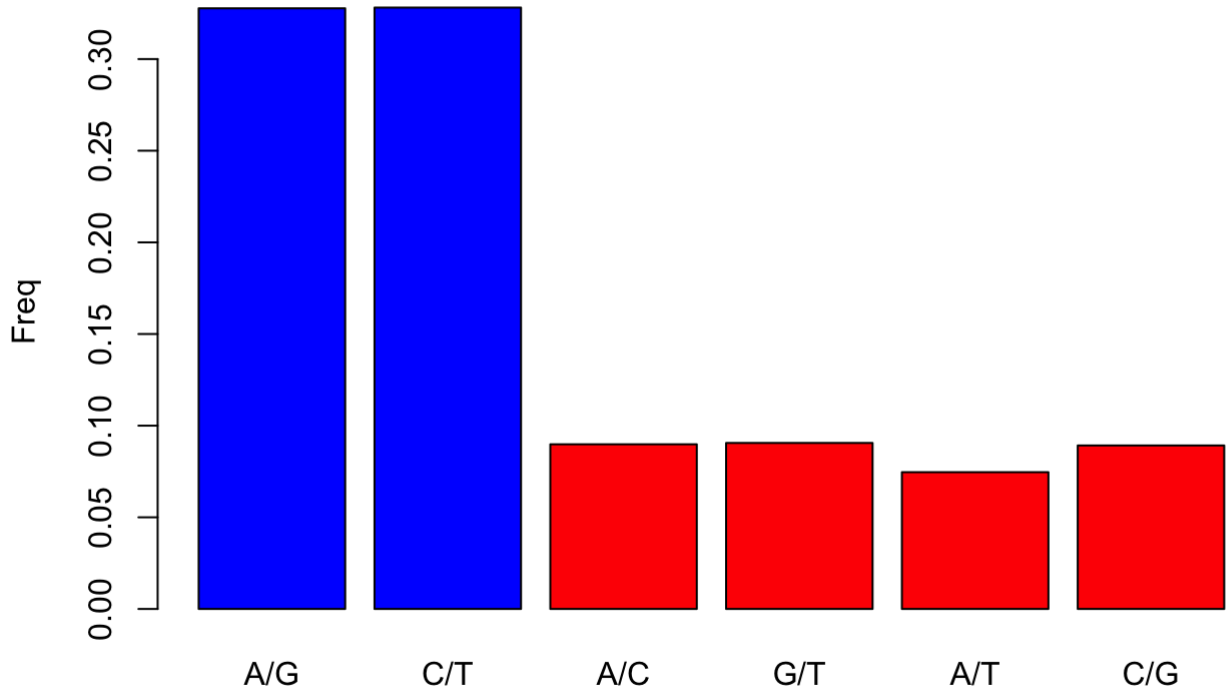
```
barplot(snp,col=c("blue","blue",rep("red",4)),ylab="Count")
```



Show the frequency of each type of SNP using boxplot

```
barplot(snp/sum(snp),col=c("blue","blue",rep("red",4)),ylab="Freq")
```

用 frequency 来画图



We may also use functions in ggplot2 to visualize.

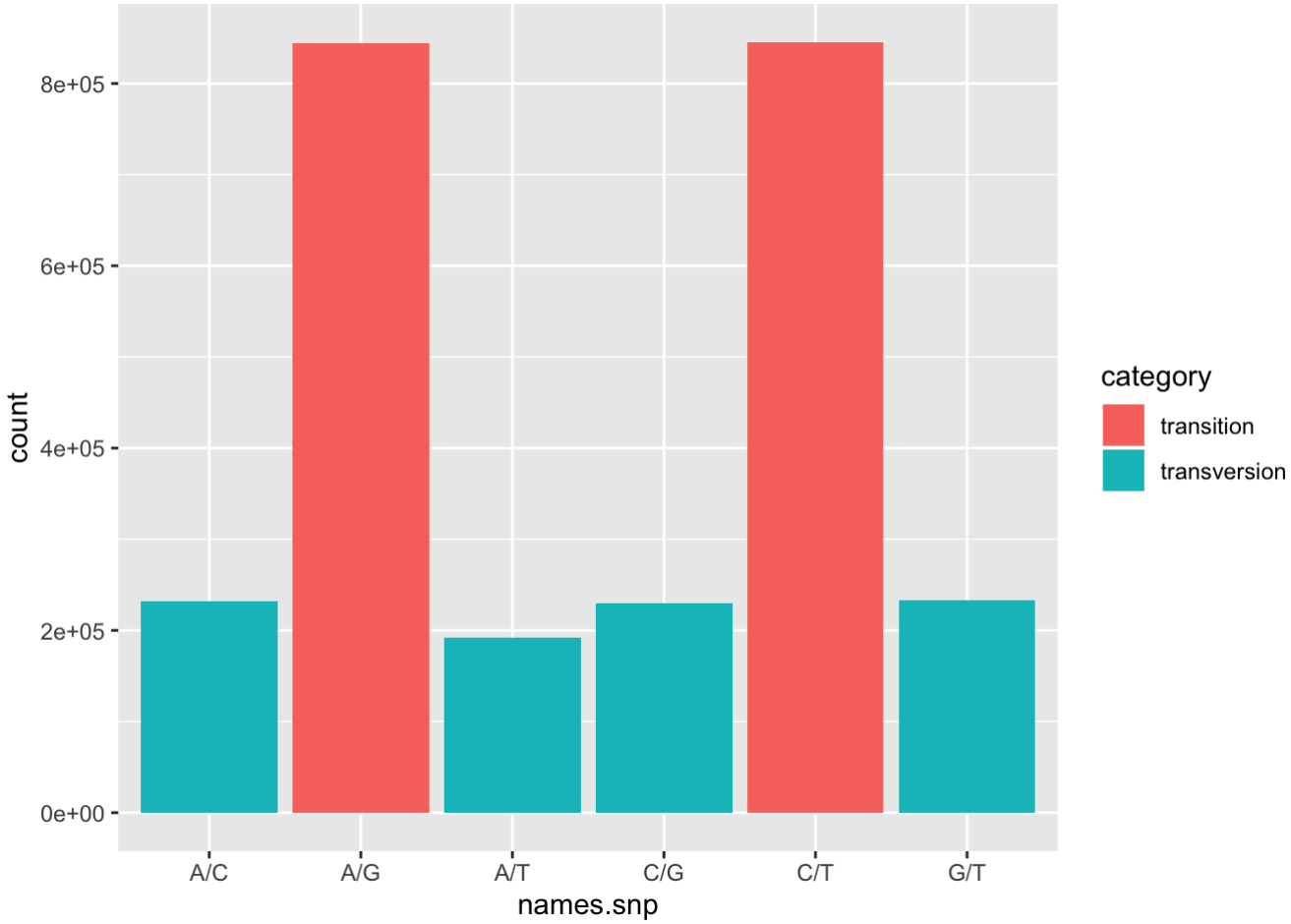
```
library(ggplot2)
library(ggridges)  画ridge图需要
library(tidyverse) 数据data.frame格式；%>%
snp.tib <- data.frame(SNPType=names.snp,snp=snp,category = c(rep("transition",2),rep(
"transversion",4))) %>% as.tibble()  定义为tibble格式
```

If directly using geom_bar(), the data will be ordered alphabetically.

按照category来填充不同颜色

```
snp.tib %>% ggplot(aes(names.snp,weight=snp,fill=category)) + geom_bar()
```

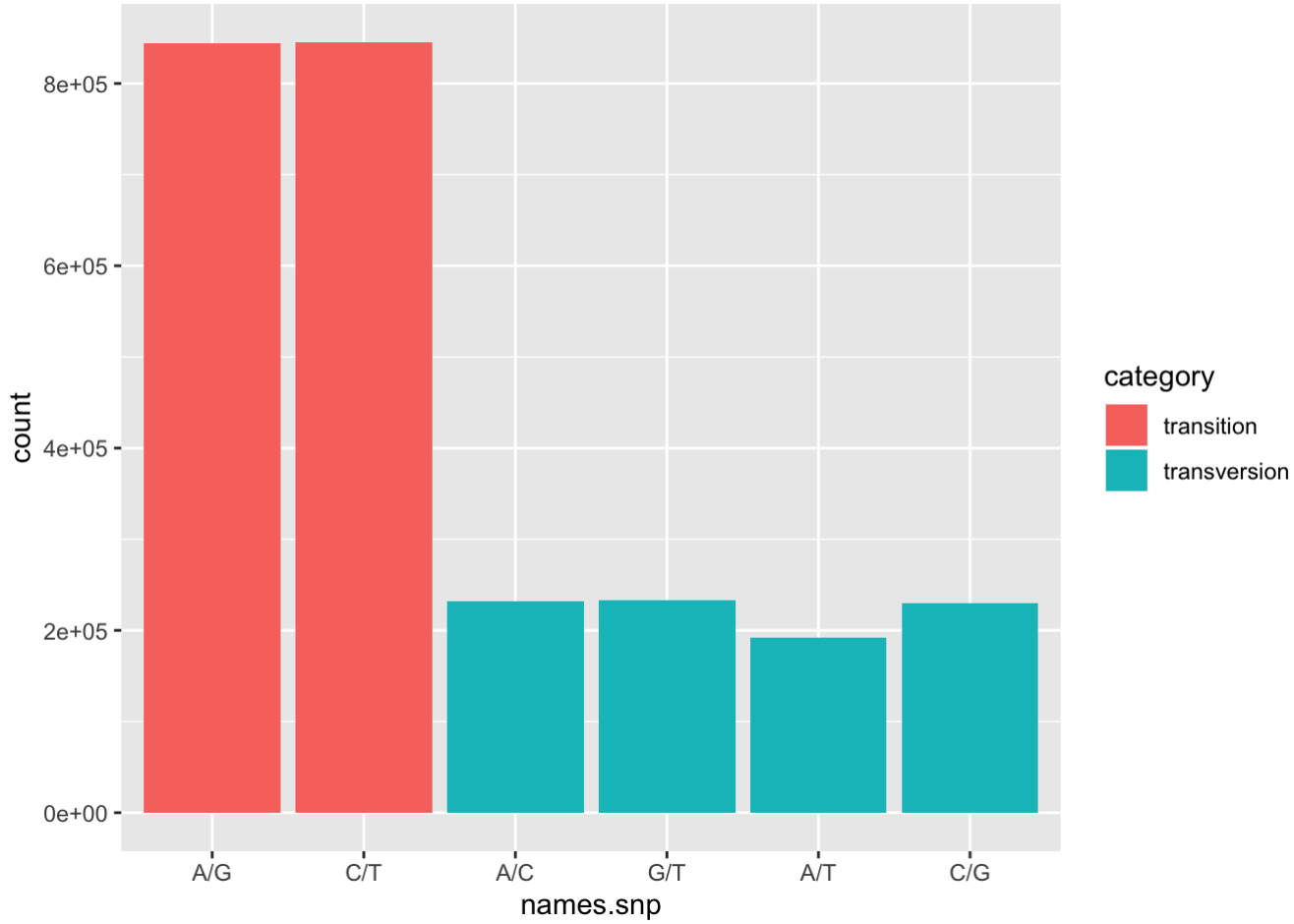
左边的数据给右边的函数，数据流，避免定义中间变量



We have to define a factor according to the given order

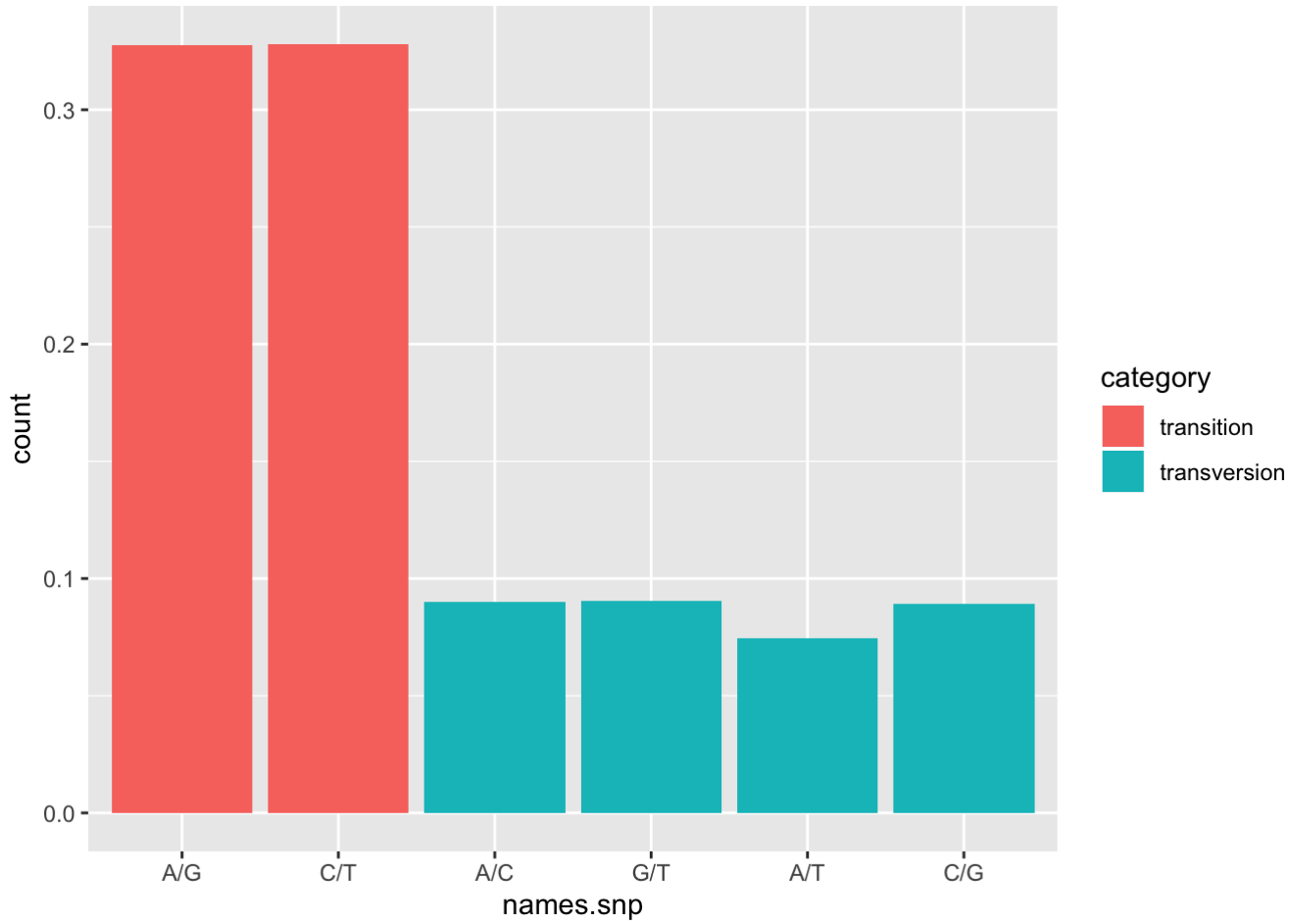
把某一列稍微改一下，factor形式省内存，但是容易有bug，所以一般读入数据时 会 StringasFastor=F

```
snp.tib %>% mutate(names.snp=factor(names.snp,levels=names.snp)) %>% ggplot(aes(name
s.snp,weight=snp,fill=category)) + geom_bar()
```




```
snp.tib %>% mutate(names.snp=factor(names.snp,levels=names.snp)) %>% ggplot(aes(name
s.snp,weight=snp/sum(snp),fill=category)) + geom_bar()
```

纵轴改为 frequency



Now we consider the Iris data

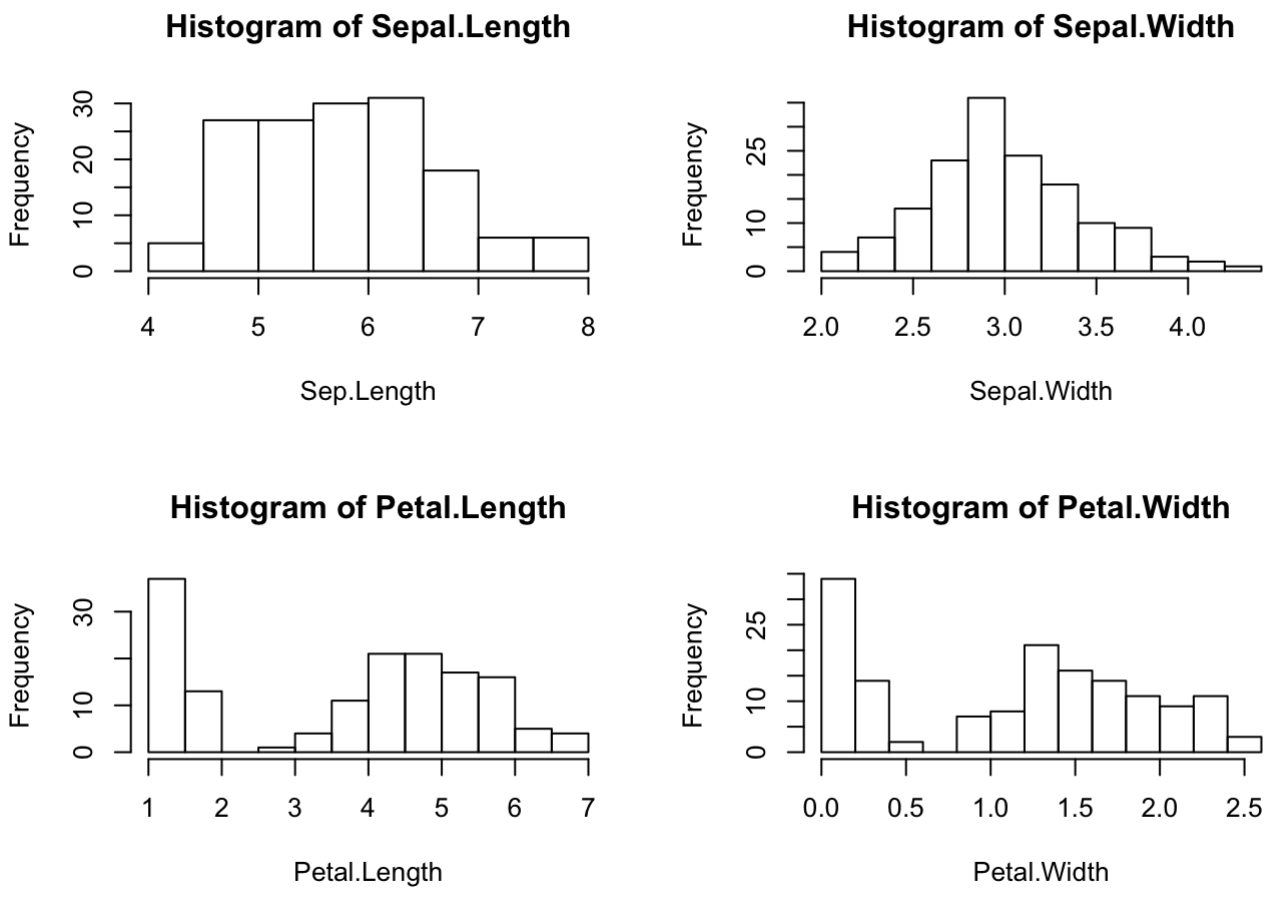
```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Plot the histogram of the four different measurements

```
par(mfrow=c(2,2))
hist(iris[,1],main="Histogram of Sepal.Length",xlab="Sep.Length")
hist(iris[,2],main="Histogram of Sepal.Width",xlab="Sepal.Width")
hist(iris[,3],main="Histogram of Petal.Length",xlab="Petal.Length")
hist(iris[,4],main="Histogram of Petal.Width",xlab="Petal.Width")
```

设定画布分成两行两列（4份）下方四个图同时展现



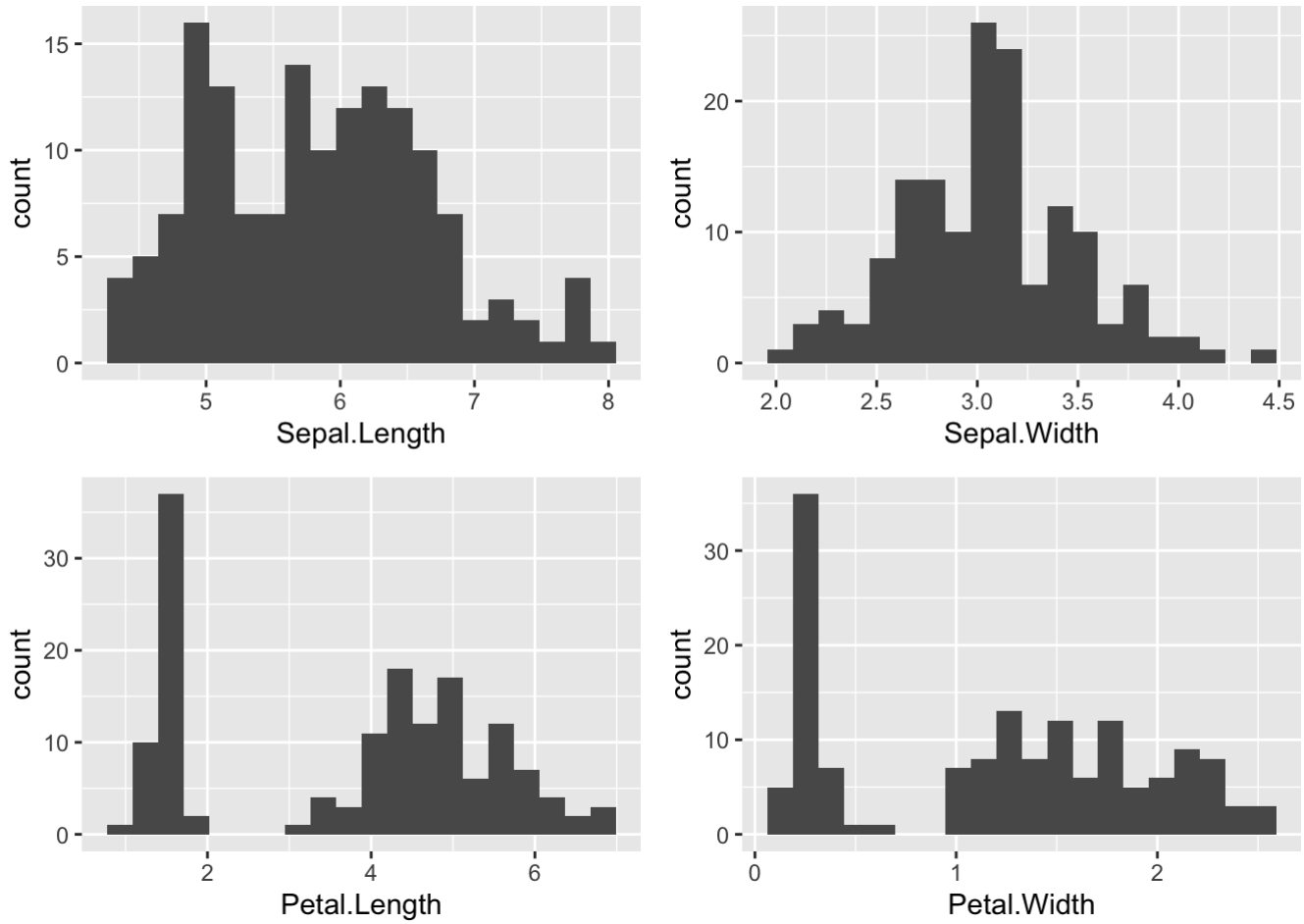
Using ggplot2. We have to use the gridExtra package to arrange the four plots.

```
library(gridExtra)
```

Now we can plot the histogram.

```
xp1 = iris %>% ggplot(aes(Sepal.Length)) + geom_histogram(bins=20)
xp2 = iris %>% ggplot(aes(Sepal.Width)) + geom_histogram(bins=20)
xp3 = iris %>% ggplot(aes(Petal.Length)) + geom_histogram(bins=20)
xp4 = iris %>% ggplot(aes(Petal.Width)) + geom_histogram(bins=20)
m1 <- grid.arrange(xp1, xp2, xp3, xp4,
             ncol = 2, nrow = 2)
```

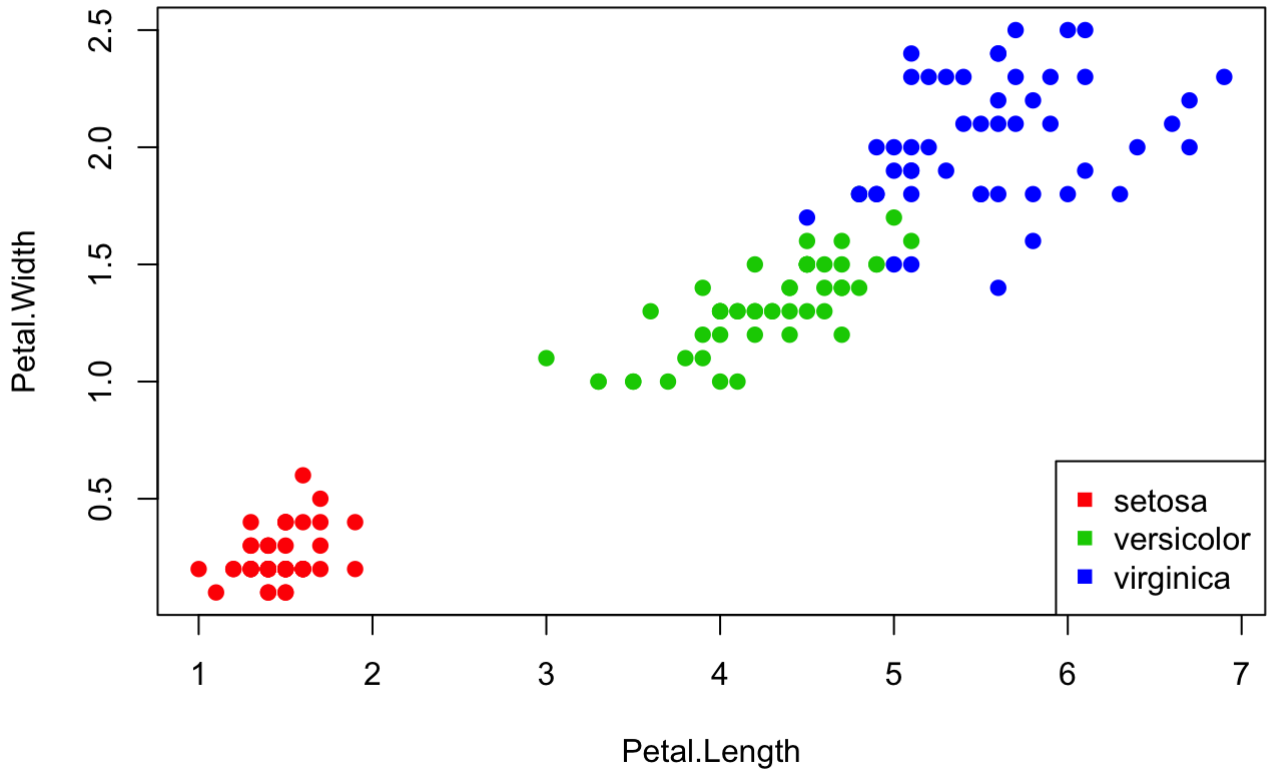
分画布位置的，功能同mfrow，需要library(gridExtra)



Now plot the scatter plot

```
plot(iris[,3],iris[,4],pch=19,xlab="Petal.Length",ylab="Petal.Width",col=c("red","gre
en3","blue")[unclass(iris$Species)])
legend("bottomright",legend=c("setosa","versicolor","virginica"),col=c("red","green3"
,"blue"),pch=15)
```

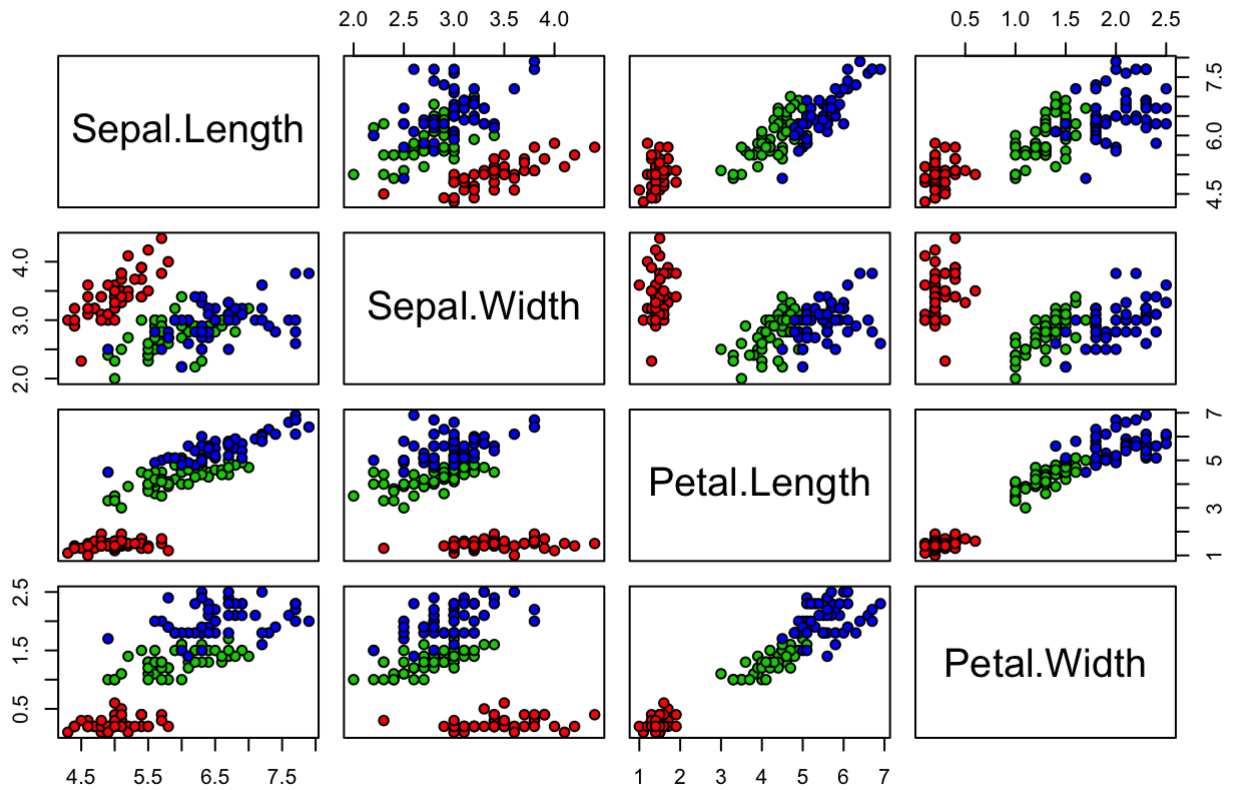
图标



The pairwise scatter plots

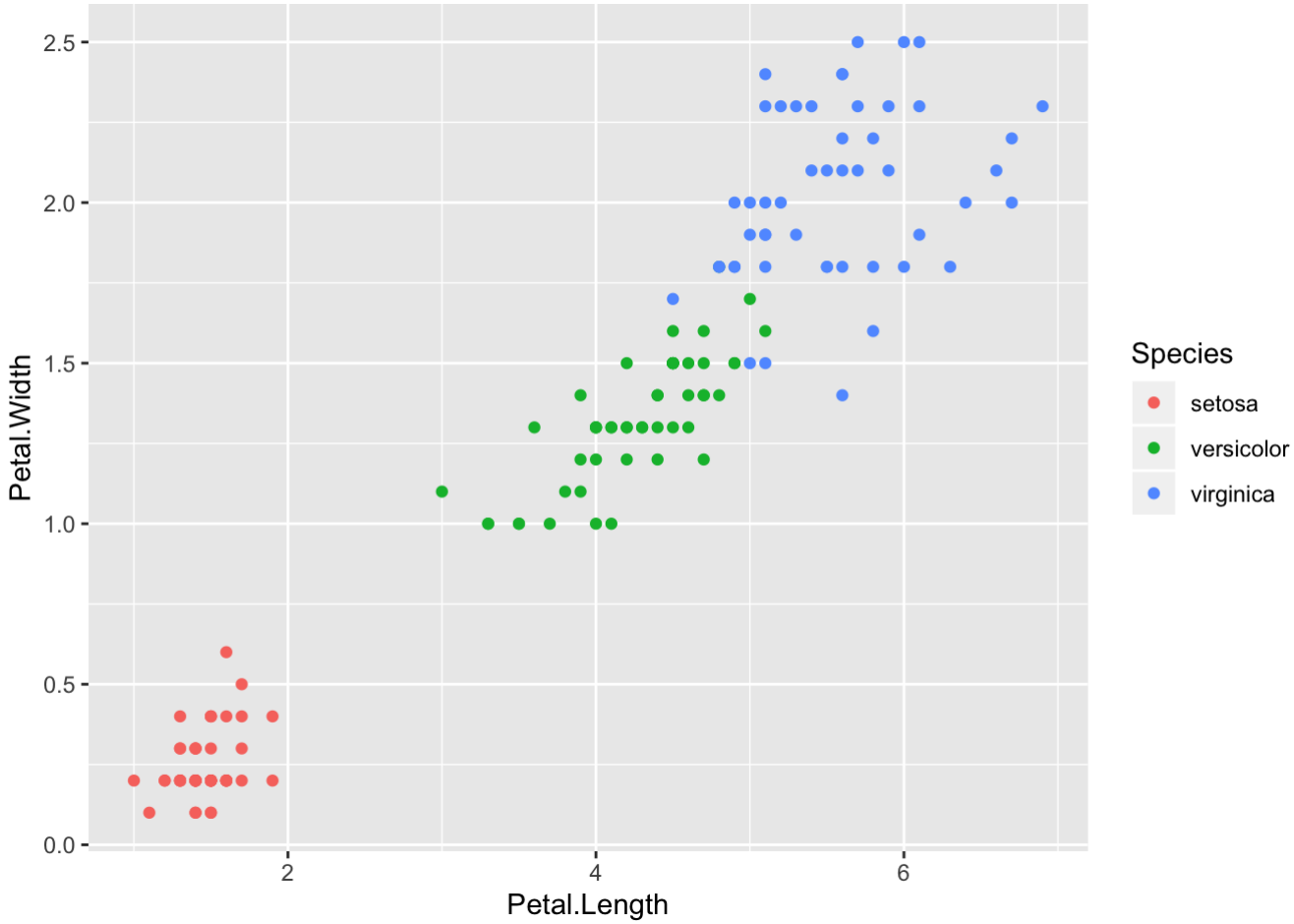
每两列来画scatter plot

```
pairs(iris[1:4],main="Iris Data (red=setosa,green=versicolor,blue=virginica)", pch=21
,
  bg=c("red","green3","blue")[unclass(iris$Species)])
```



Scatter plot by ggplot2.

```
iris %>% ggplot(aes(Petal.Length,Petal.Width,col=Species)) + geom_point()
```

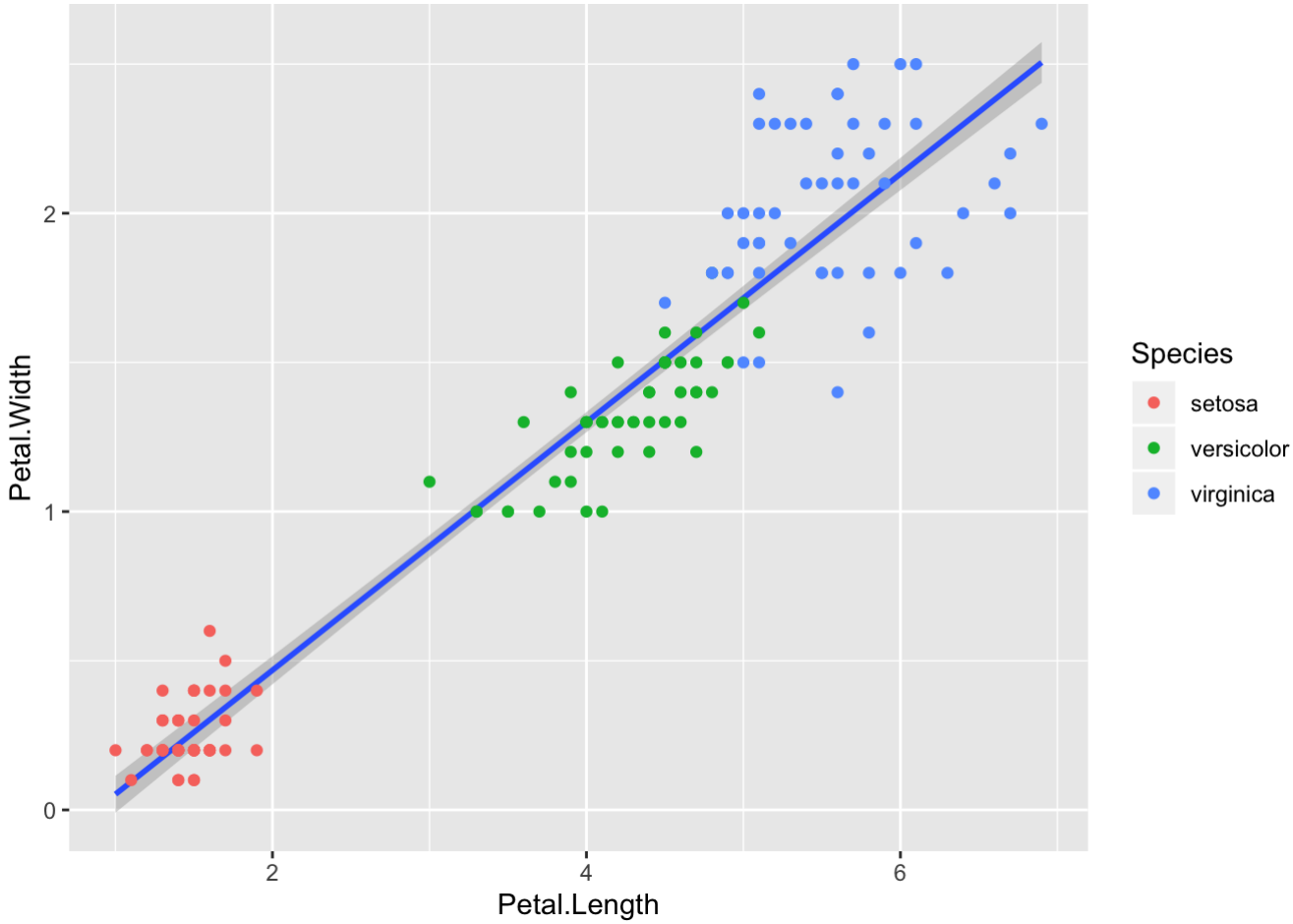


Add a regression line

```
iris %>% ggplot(aes(Petal.Length,Petal.Width)) + geom_smooth(method=lm) + geom_point
(aes(col=Species))
```

线性回归方法来把数据光滑化，同时画出回归曲线
阴影为置信度区间

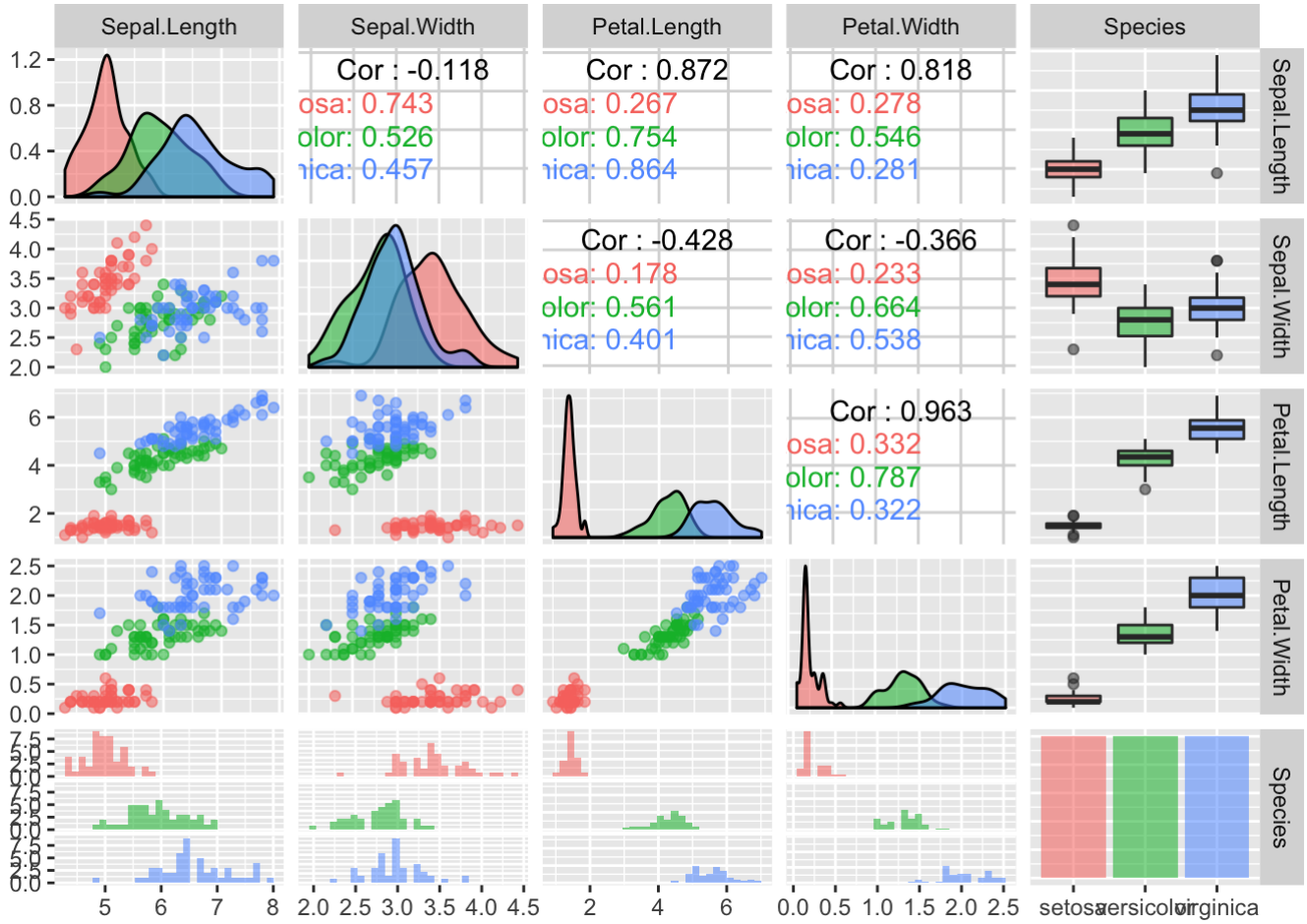
画点



There is no easy way of plotting pairwise scatterplots using ggplot2. We have to use the GGally package.

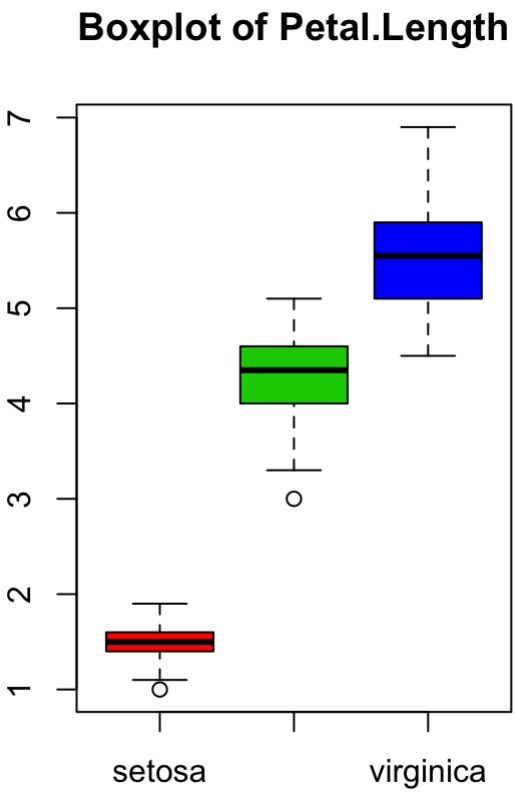
```
library(GGally)
ggpairs(iris, aes(colour = Species, alpha = 0.4))
```

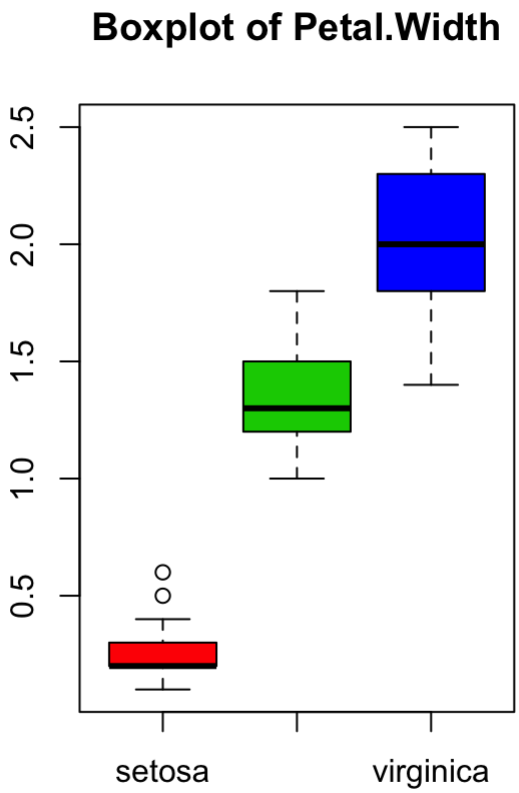
两两之间的scatter plot ， 密度函数，box plot，hist等



Now we plot the boxplots.

```
par(mfrow=c(1,2))
boxplot(Petal.Length~ Species,data=iris,col=c("red","green3","blue"),main="Boxplot of
Petal.Length")
boxplot(Petal.Width~ Species,data=iris,col=c("red","green3","blue"),main="Boxplot of
 Petal.Width")
```

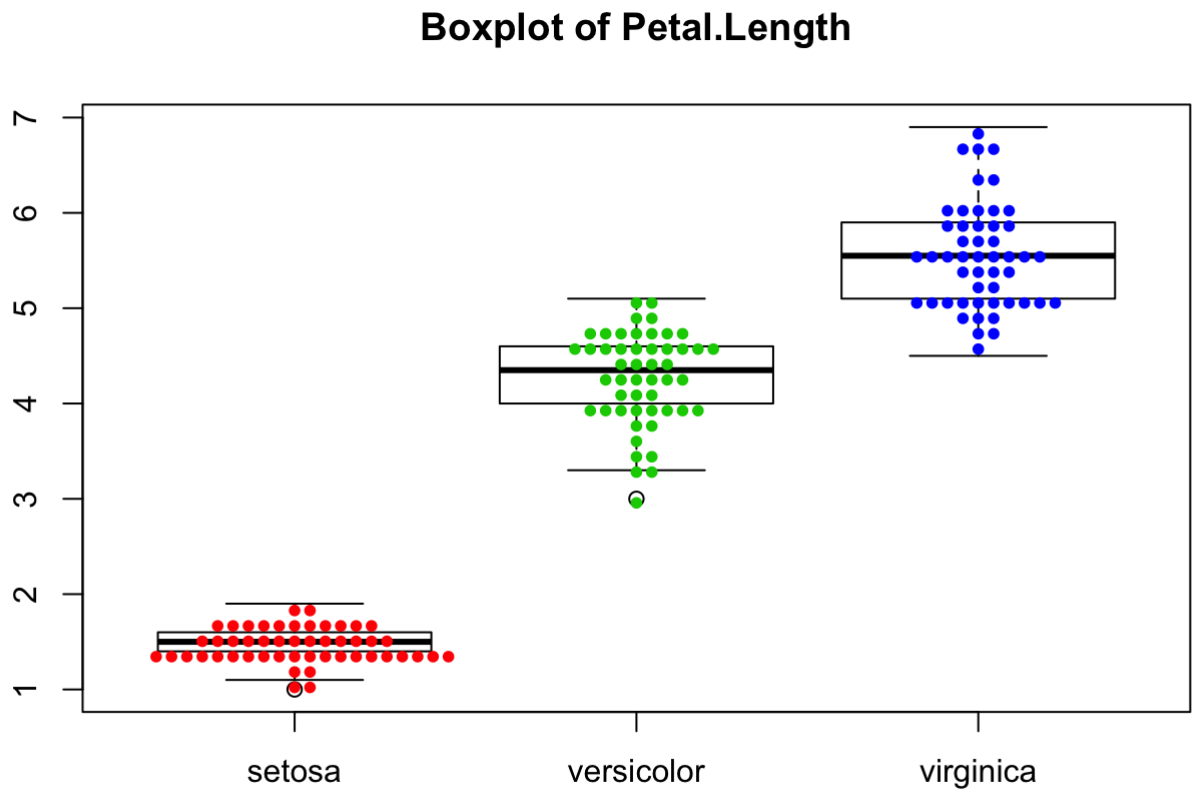




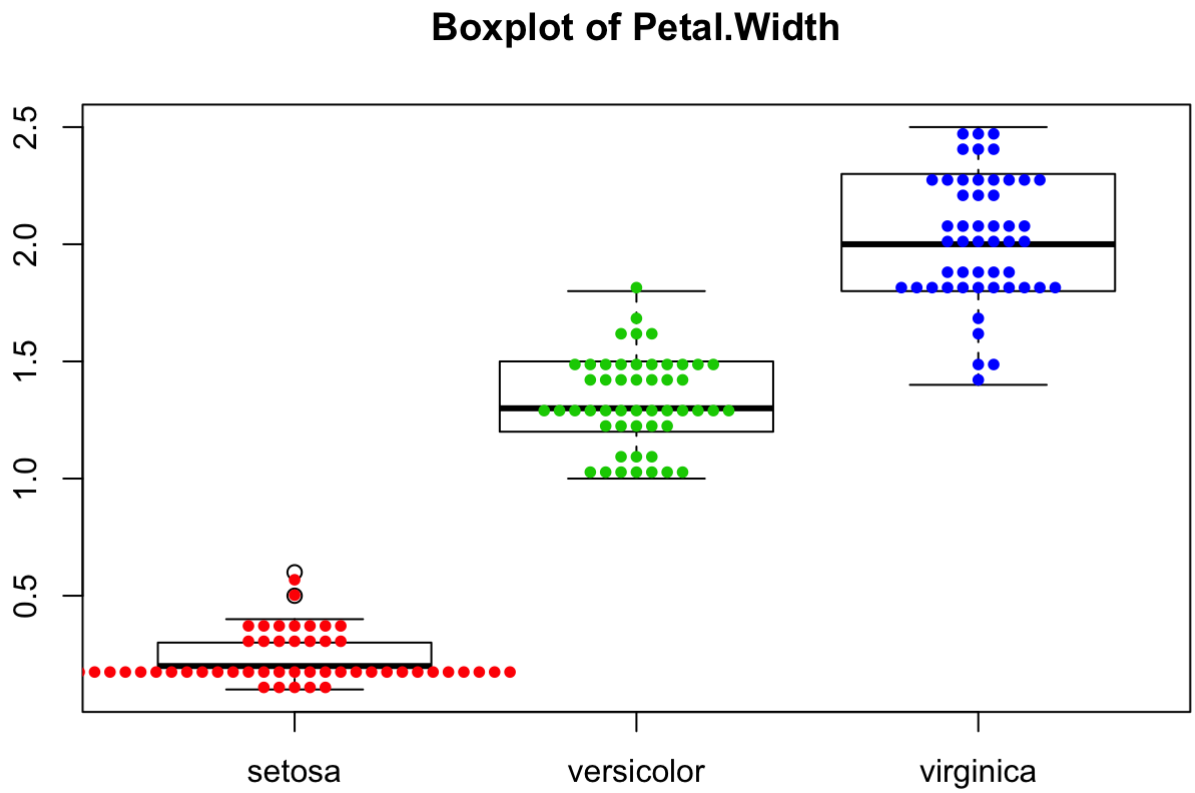
Now add the beeswarm plot. We have to use the beeswarm library.

```
library("beeswarm")
boxplot(Petal.Length~ Species,data=iris,main="Boxplot of Petal.Length")
beeswarm(Petal.Length~ Species,data=iris,col=c("red","green3","blue"),add=TRUE,pch=20
,method="square")
```

在上一张图上叠加信息

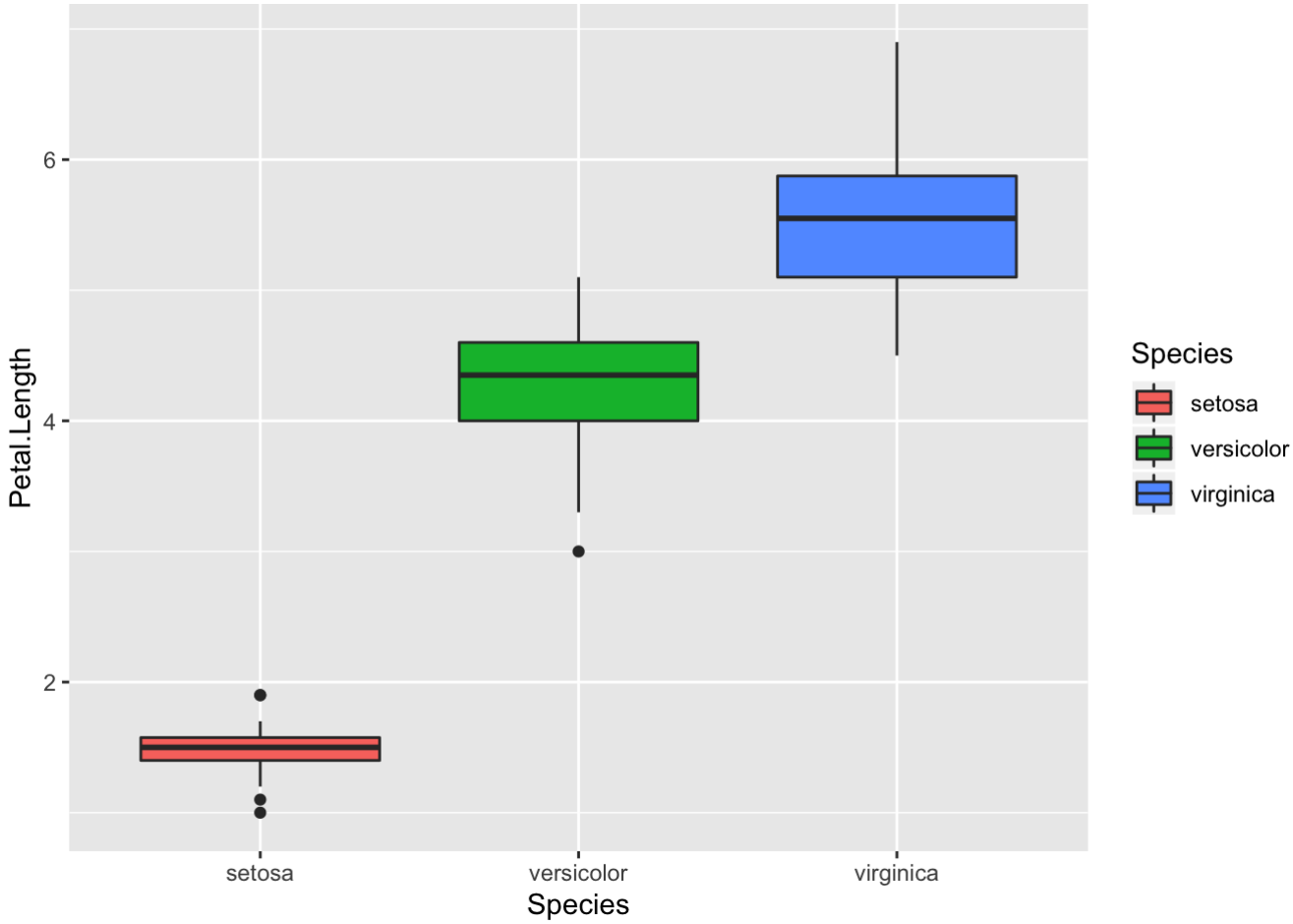



```
boxplot(Petal.Width~ Species,data=iris,main="Boxplot of Petal.Width")
beeswarm(Petal.Width~ Species,data=iris,col=c("red","green3","blue"),add=TRUE,pch=20,
method="square")
```



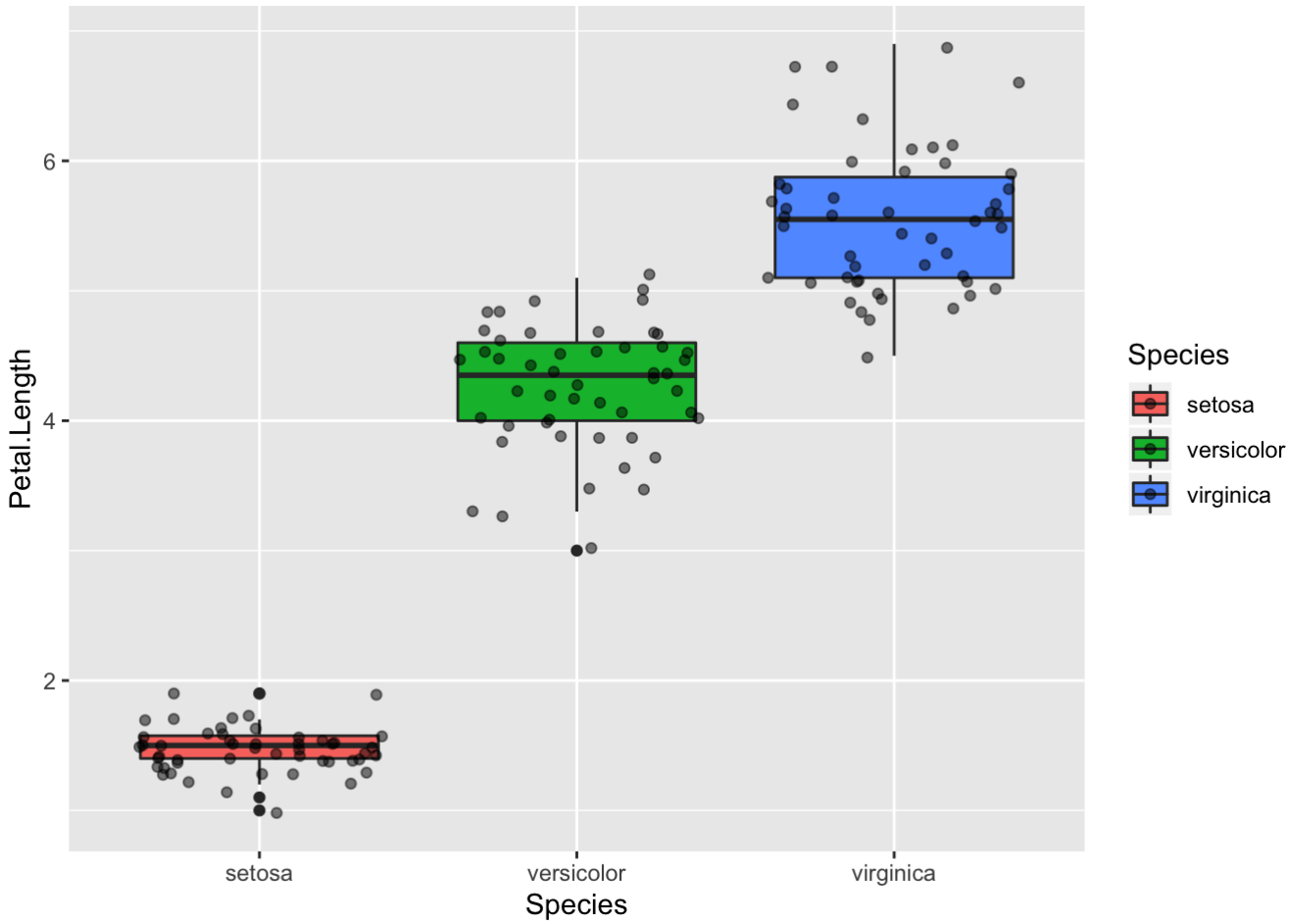
Now we use ggplot2 to plot the boxplot.

```
iris %>% ggplot(aes(Species,Petal.Length,fill=Species)) + geom_boxplot()
```



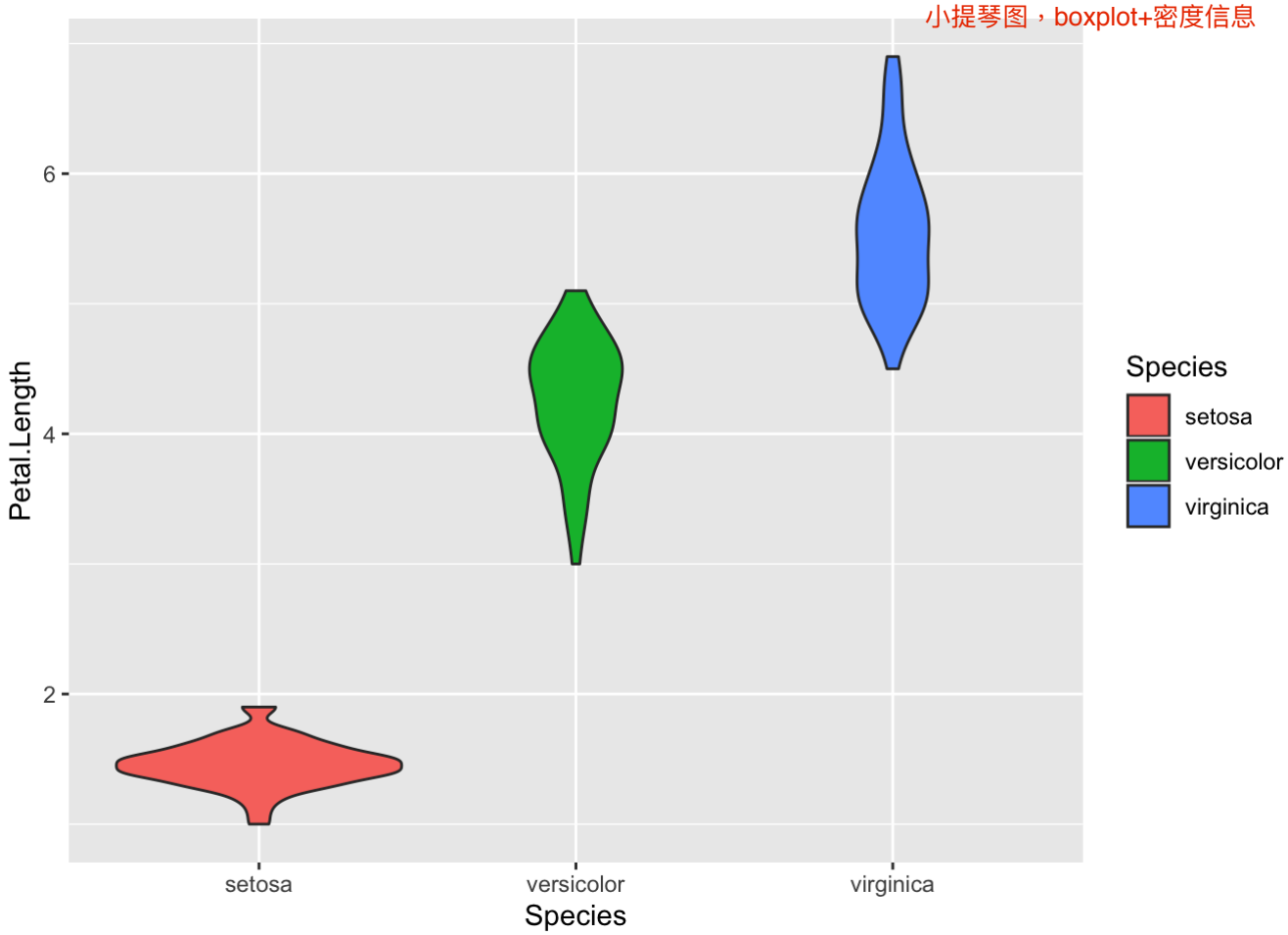

```
iris %>% ggplot(aes(Species,Petal.Length,fill=Species)) + geom_boxplot() + geom_jitte
r(alpha=0.5)
```

透明度，密度越小越透明



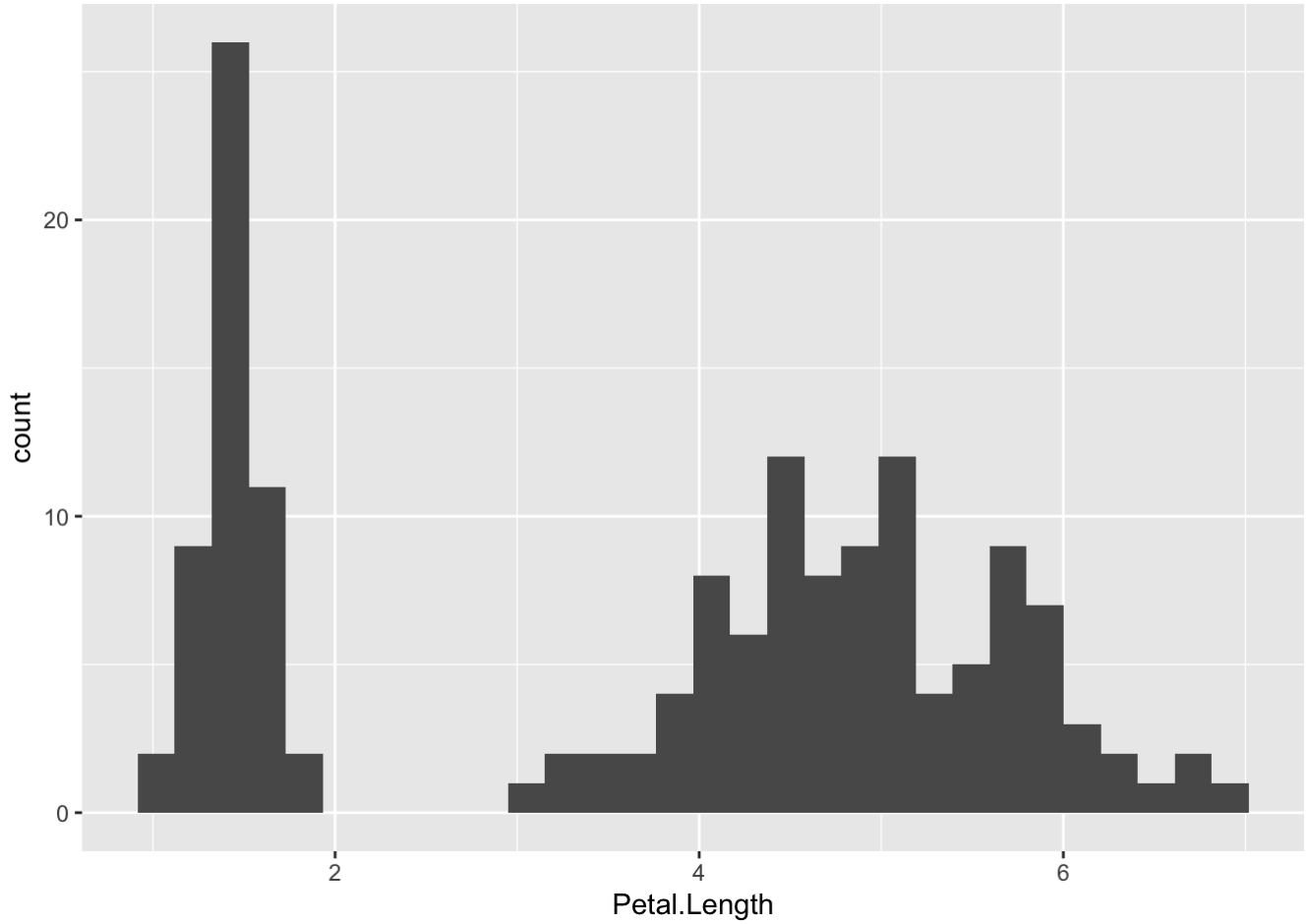
```
iris %>% ggplot(aes(Species,Petal.Length,fill=Species)) + geom_violin()
```

小提琴图，boxplot+密度信息

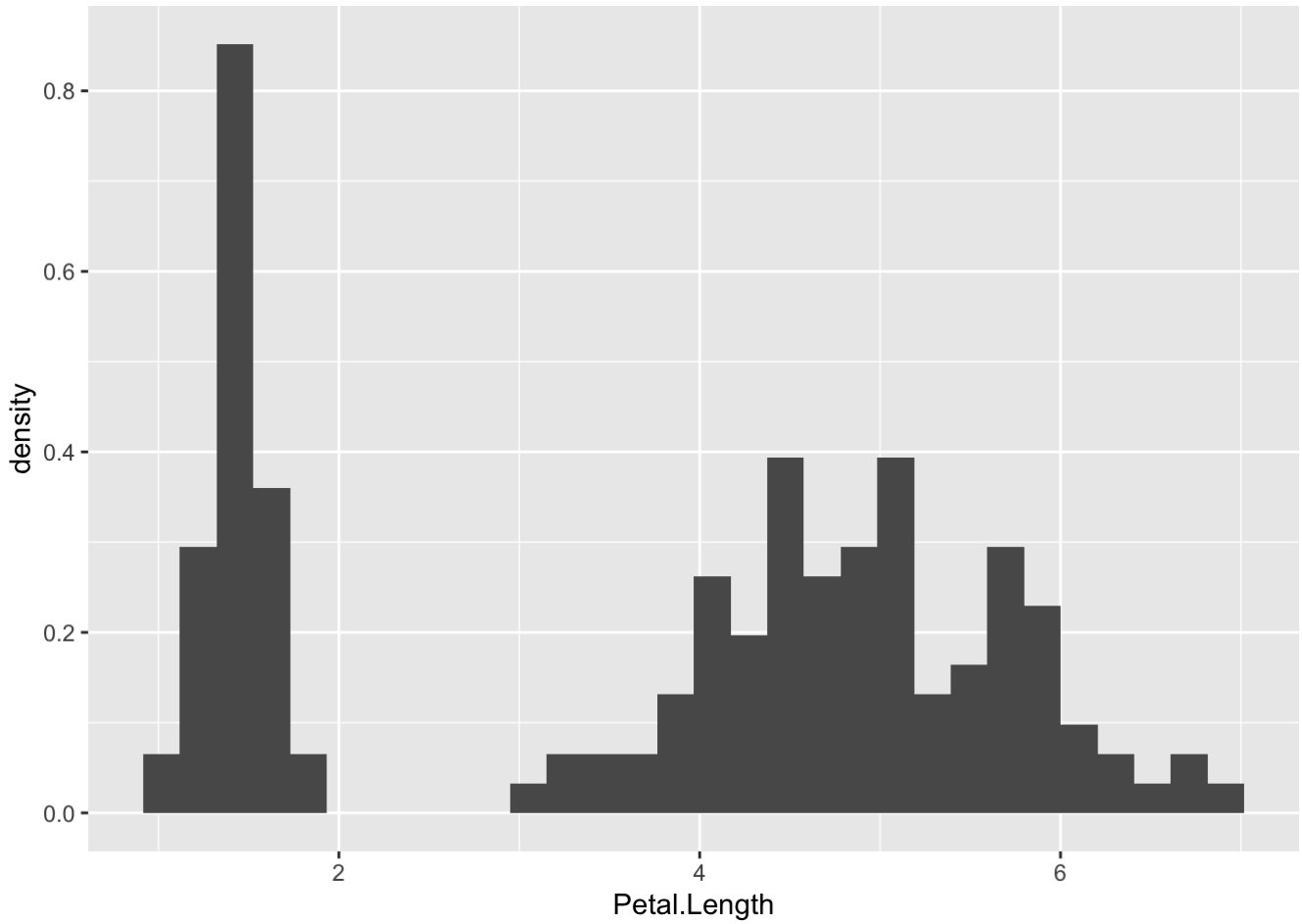


Now we plot the density

```
iris %>% ggplot(aes(Petal.Length)) + geom_histogram()
```

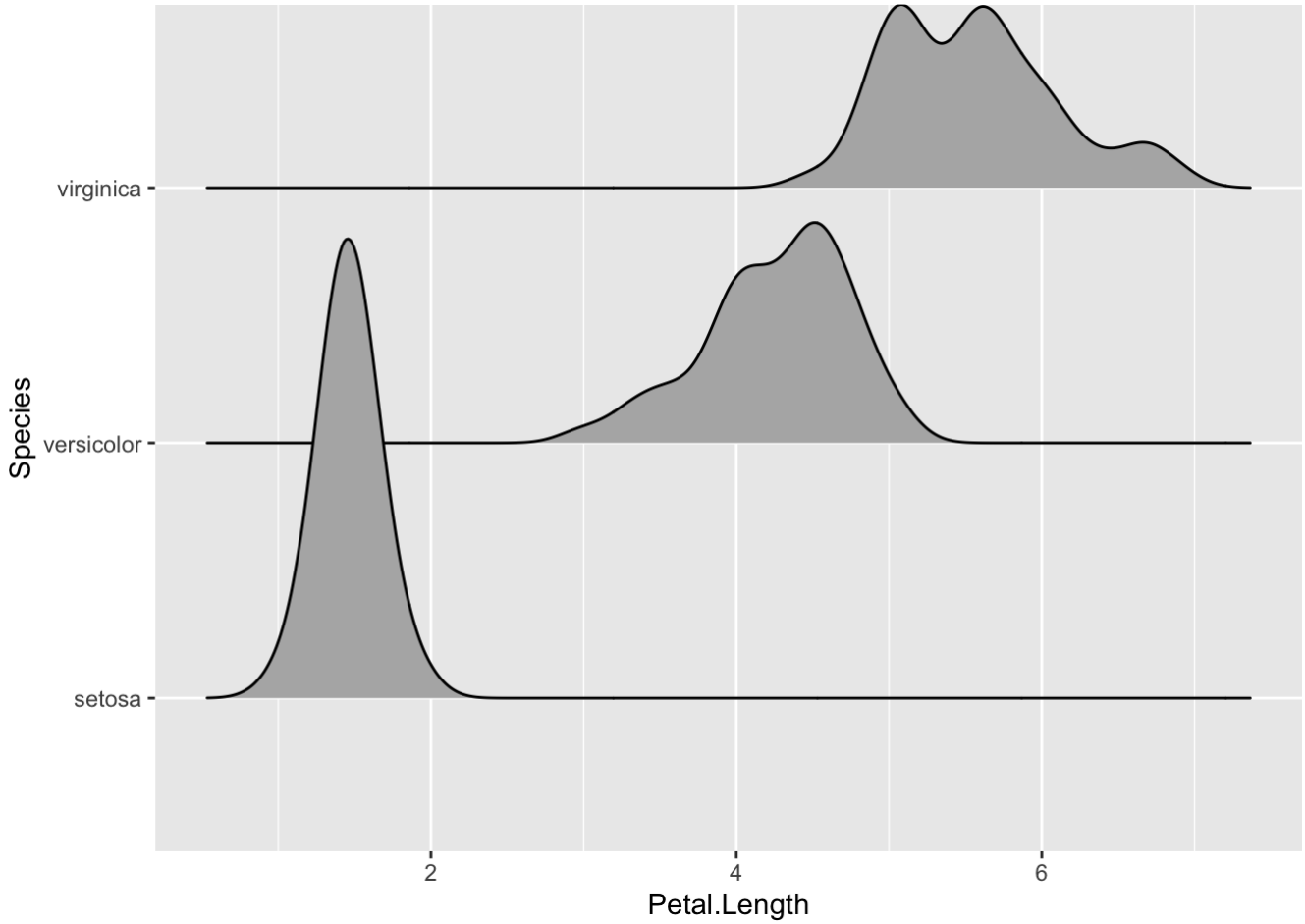



```
iris %>% ggplot(aes(Petal.Length,stat(density))) + geom_histogram()
```



The ridges plot.

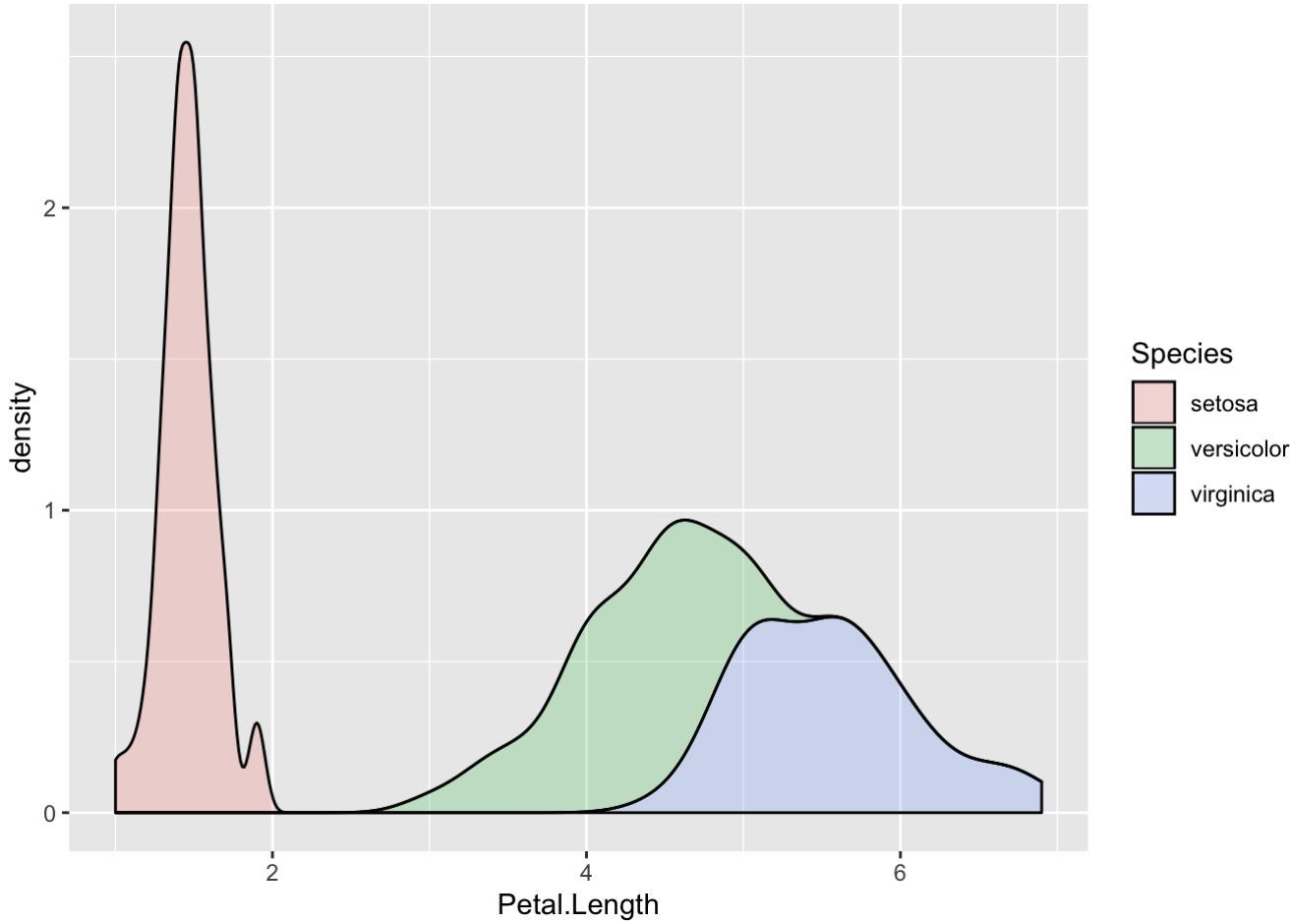
```
iris %>% ggplot(aes(Petal.Length,Species)) + geom_density_ridges()
```



Stacking the densities on top of each other.

```
iris %>% ggplot(aes(Petal.Length,fill=Species)) + geom_density(alpha = 0.2,  position
= "stack")
```

ridge plot，画在同一个图上，且区分颜色



Now we consider the QQ plot. We use the BMI data from the MASS package

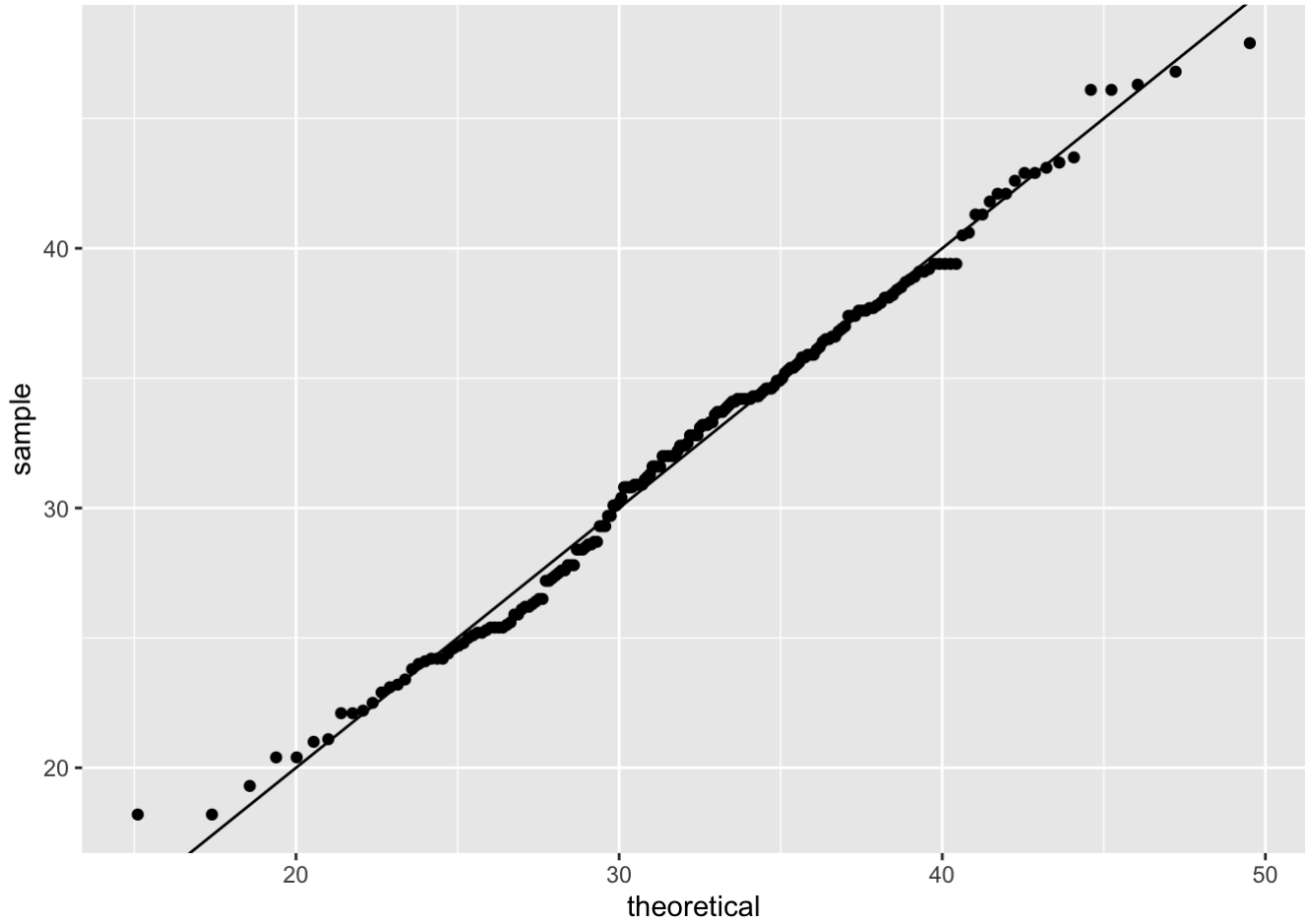
```
library(MASS)
data(Pima.tr)
```

We use the ggplot2 to plot the QQ plot

qqplot需要先算出平均值和sd参数，这里把参数存在params变量里

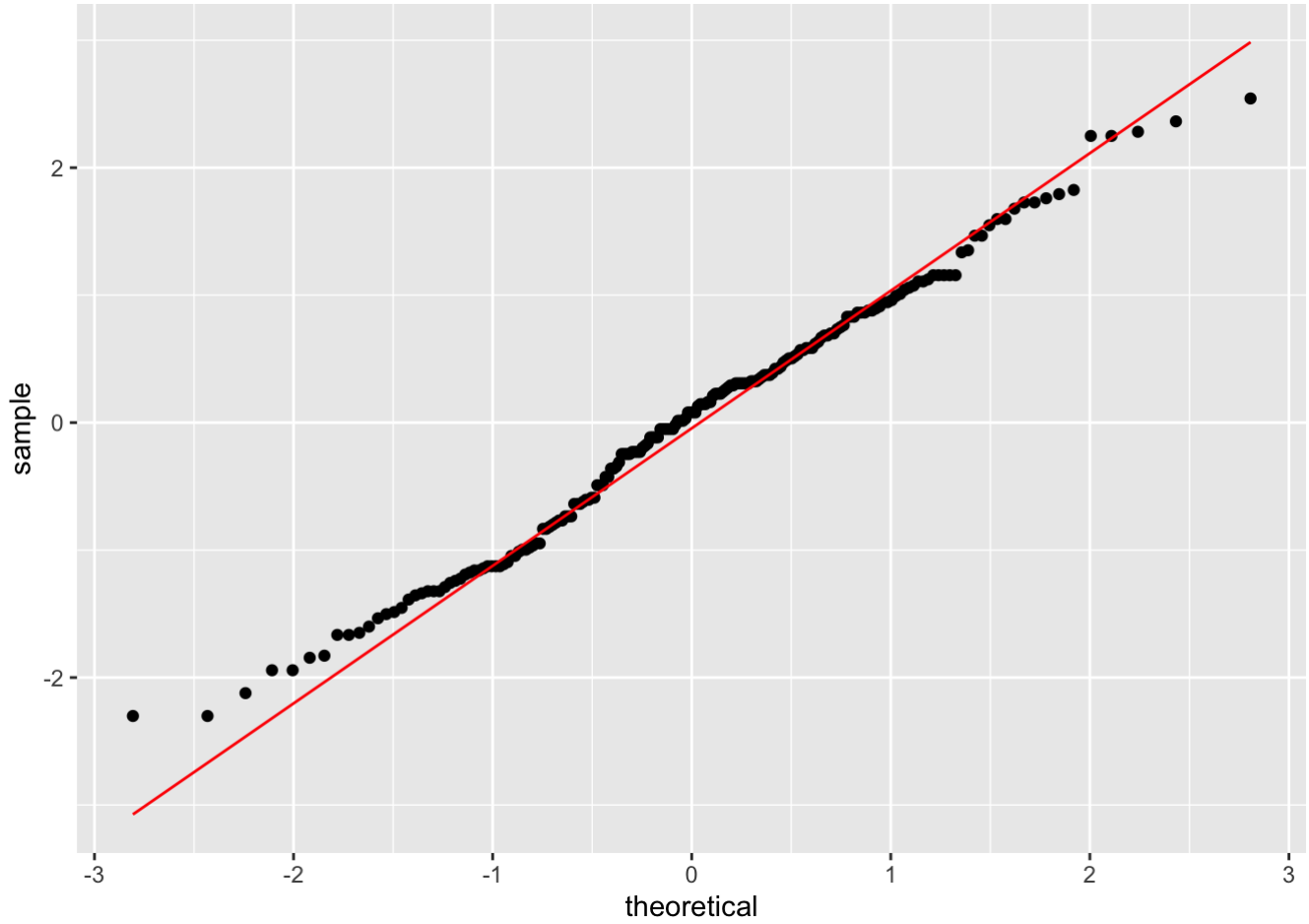
```
params = Pima.tr %>% summarise(mean= mean(bmi),sd = sd(bmi))
Pima.tr %>% ggplot(aes(sample=bmi)) + geom_qq(dparams=params) + geom_abline()
```

qqplot



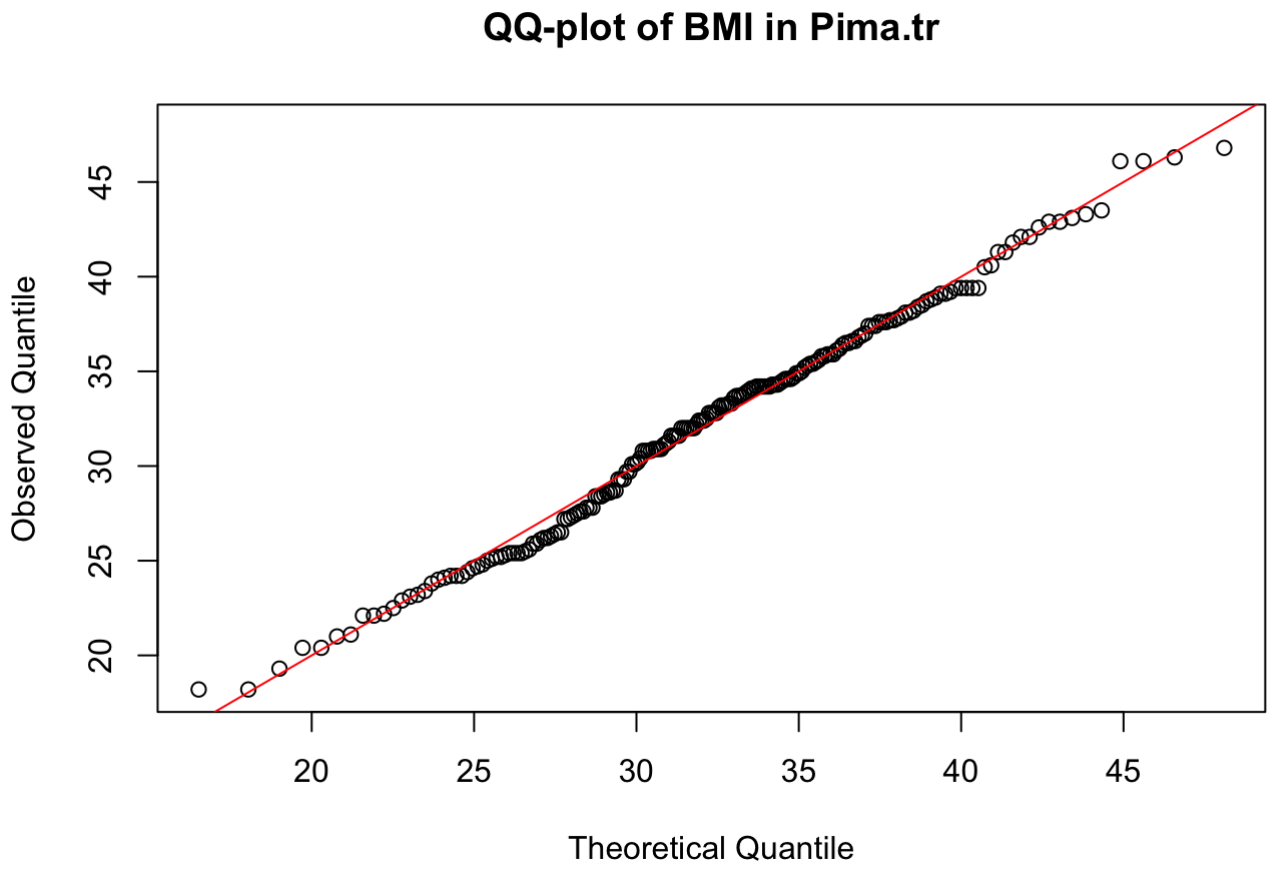
Or scale the data first and use the standard norml

```
Pima.tr %>% ggplot(aes(sample=scale(bmi))) + geom_qq() + geom_qq_line(col="red")
```



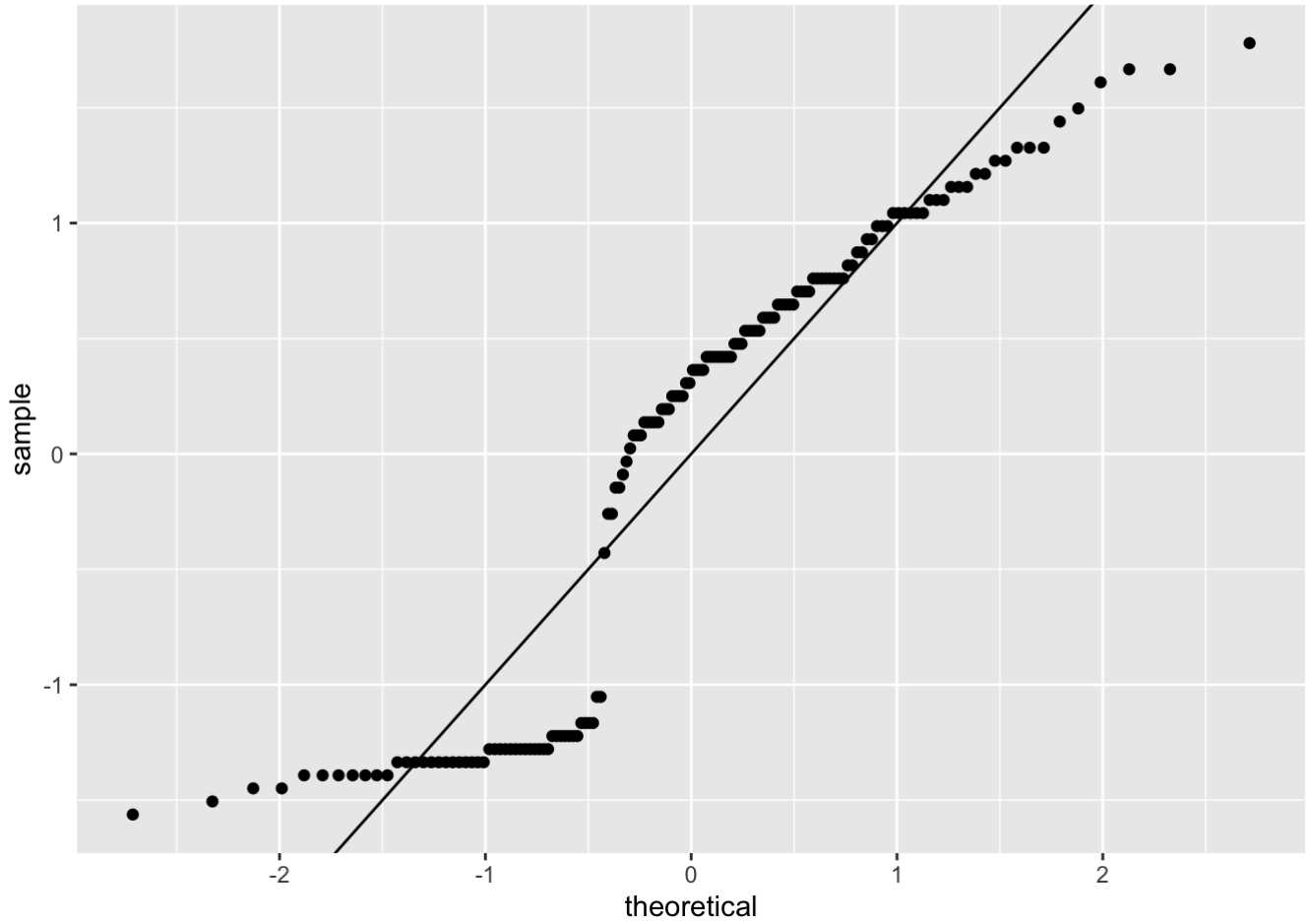
We could also use base R functions to do so.

```
m1 = mean(Pima.tr$bmi)
sd = sd(Pima.tr$bmi)
prb = ((1:nrow(Pima.tr)))/nrow(Pima.tr)
q.n = qnorm(prb,mean=m1,sd=sd)
plot(q.n,sort(Pima.tr$bmi),ylab="Observed Quantile",xlab="Theoretical Quantile",main=
"QQ-plot of BMI in Pima.tr")
abline(a=0,b=1,col="red",lwd=1)
```



The Iris data.

```
iris %>% ggplot(aes(sample=scale(Petal.Length))) + geom_qq() + geom_abline()
```



We now consider the heat maps

This code is modified from codes in “Introduction to Data Science” Consider the infectious data available in the dslabs

```
library(dslabs)
library(RColorBrewer) 颜色方案
library(tidyverse)
data(us_contagious_diseases)
names(us_contagious_diseases)
```

```
## [1] "disease"         "state"           "year"            "weeks_reporting"
## [5] "count"           "population"
```

First remove Alaska and Hawaii since they only became US states in the late 1950s. Reorder the data according to state and year.

```
the_disease <- "Measles"
dat <- us_contagious_diseases %>%
  filter(!state%in%c("Hawaii","Alaska") & disease == the_disease) %>%
  mutate(rate = count / population * 10000 * 52 / weeks_reporting) %>%
  mutate(state = reorder(state, rate))
```

数据预处理，取出想要的数

数数 并 排序

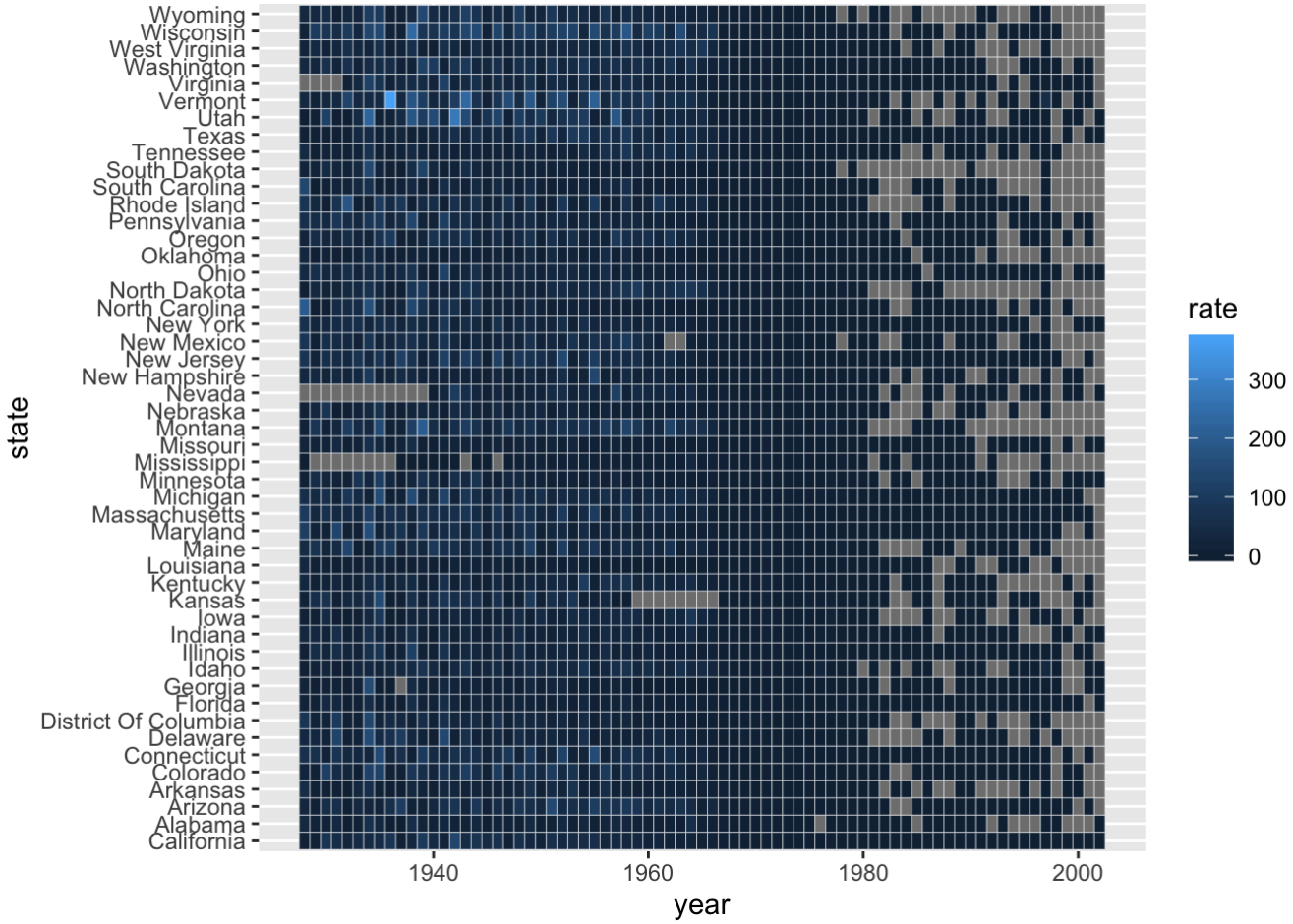
Now we can plot the heat map.

```
dat %>% ggplot(aes(year, state, fill = rate)) +
  geom_tile(color = "white")
```

fill 颜色

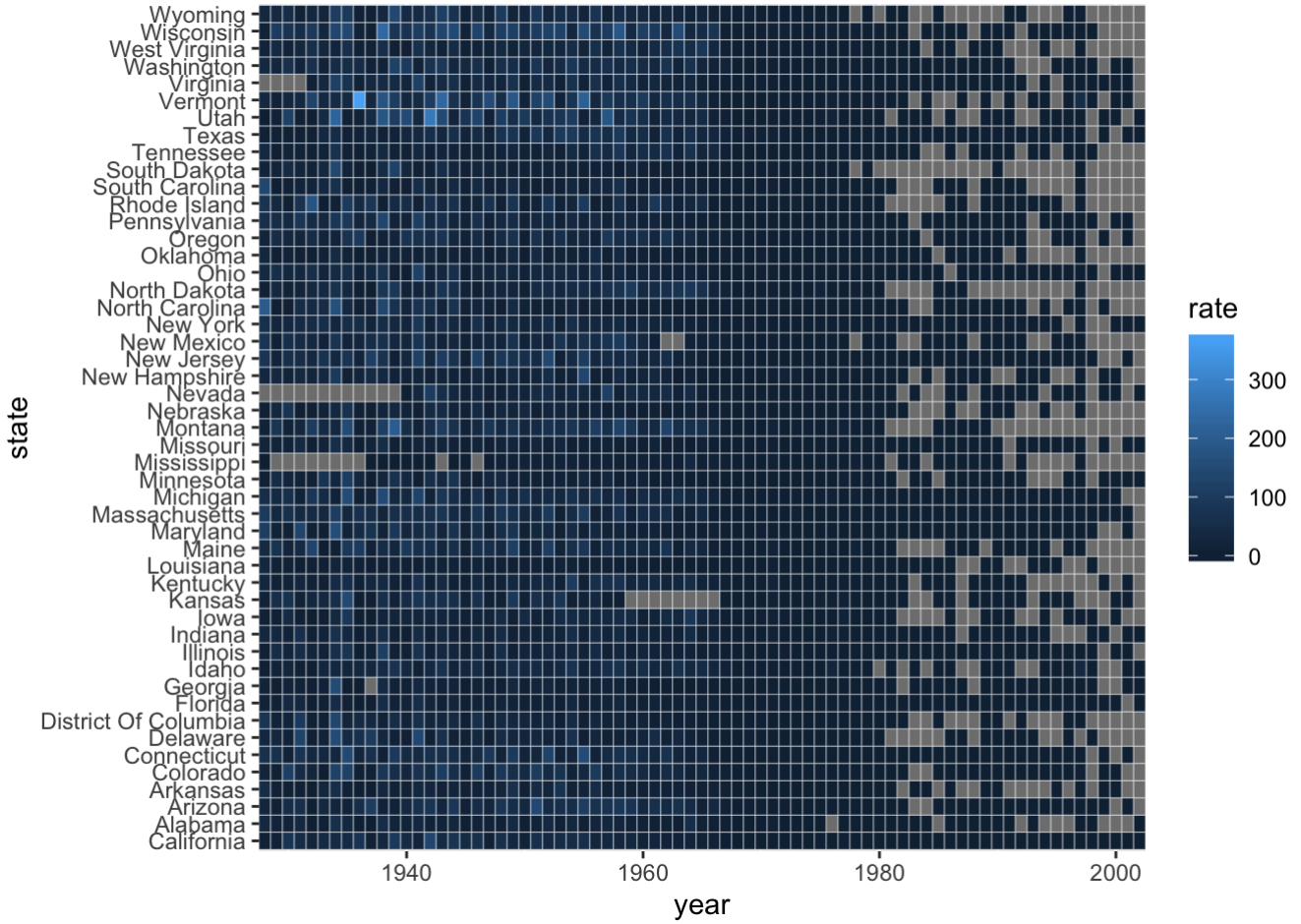
heatmap

背景色白色，灰色是NA，missing data



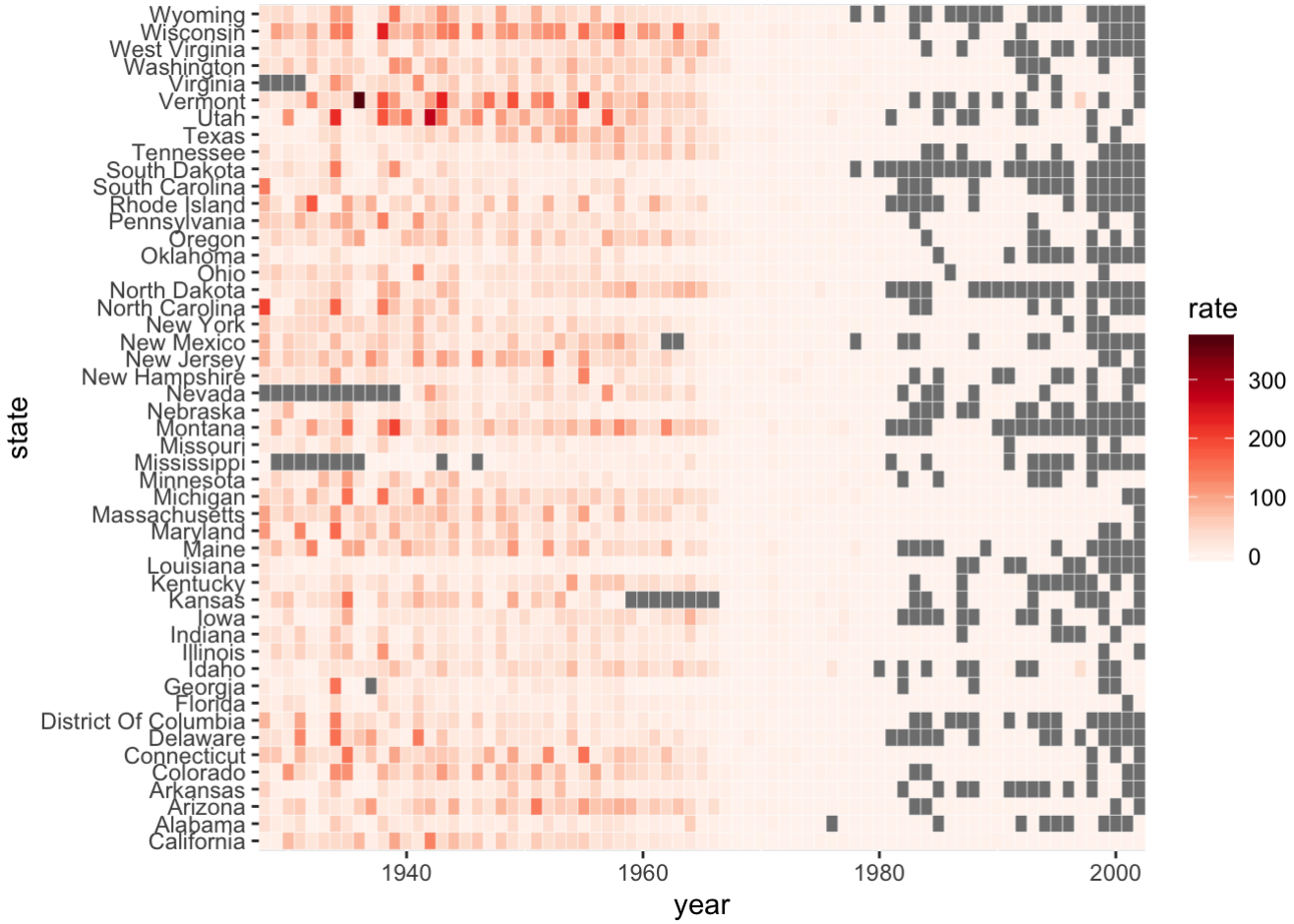
We do not want the margins on both sides of the x-axis.

```
dat %>% ggplot(aes(year, state, fill = rate)) +
  geom_tile(color = "white") + scale_x_continuous(expand=c(0,0))
```



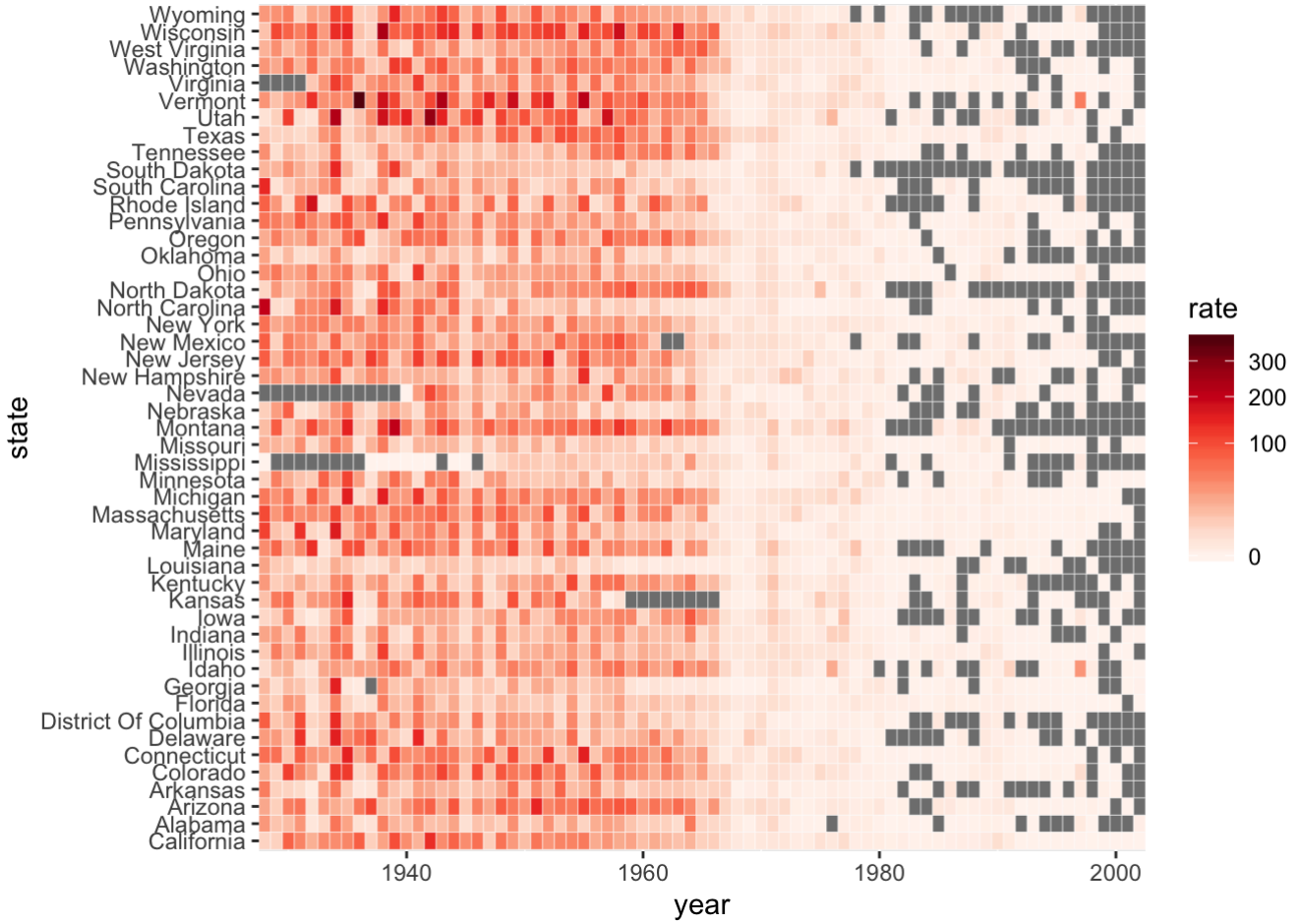
Use a different color encoding.

```
dat %>% ggplot(aes(year, state, fill = rate)) +
  geom_tile(color = "white") + scale_x_continuous(expand=c(0,0)) + scale_fill_gradien
tn(colors = brewer.pal(9, "Reds"))
```



Now transform the data using square root to better distinguish smaller values.

```
dat %>% ggplot(aes(year, state, fill = rate)) +
  geom_tile(color = "white") + scale_x_continuous(expand=c(0,0)) + scale_fill_gradien
tn(colors = brewer.pal(9, "Reds"), trans = "sqrt")
```



Draw a line to distinguish the year that vaccine was introduced.

fill 颜色

去掉左右两边空白区域

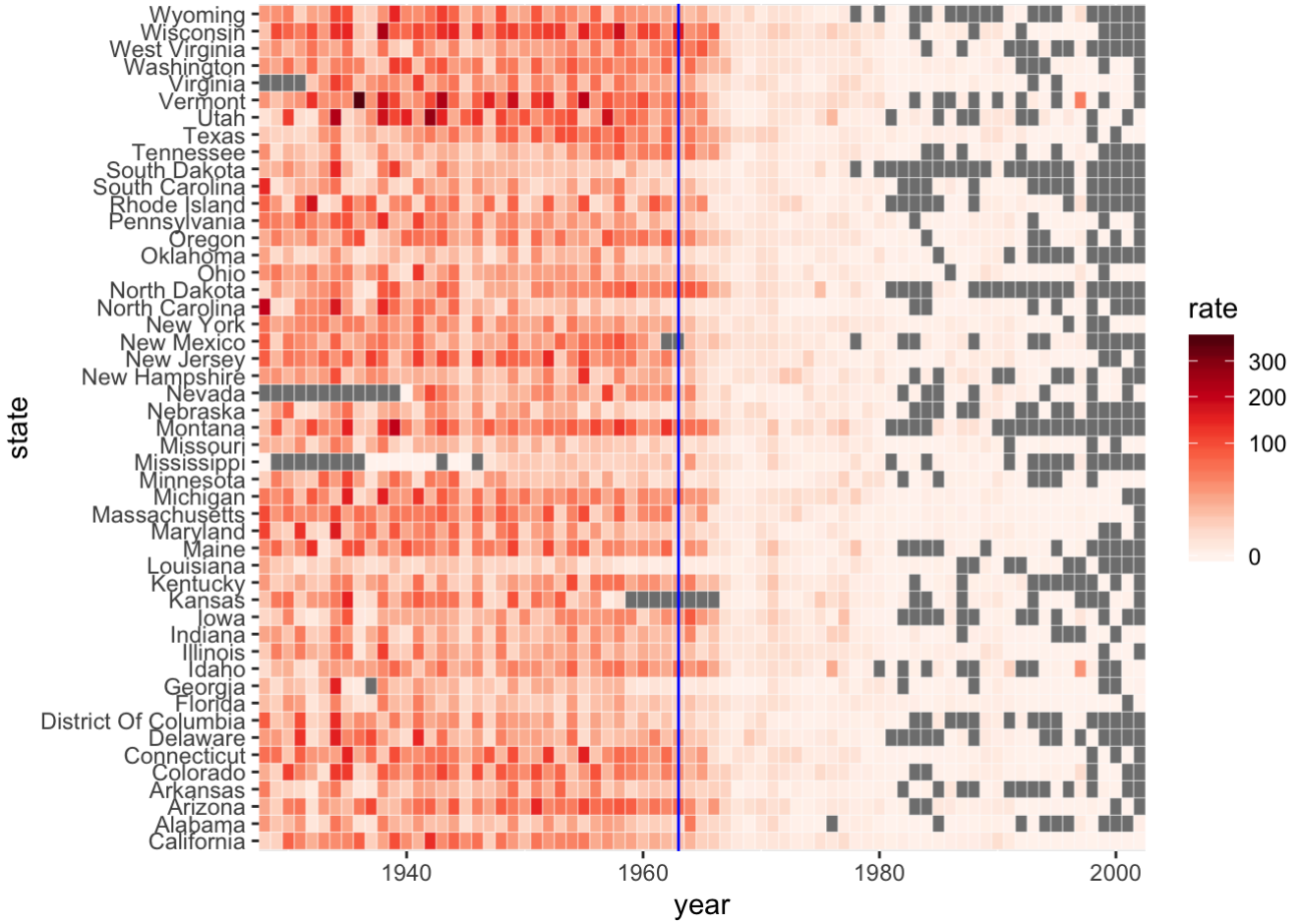
整体配色方案变为红色

heatmap
白色背景
灰NA

```
dat %>% ggplot(aes(year, state, fill = rate)) +
 geom_tile(color = "white") + scale_x_continuous(expand=c(0,0)) + scale_fill_gradien
tn(colors = brewer.pal(9, "Reds"), trans = "sqrt") + geom_vline(xintercept=1963, col
 = "blue")
```

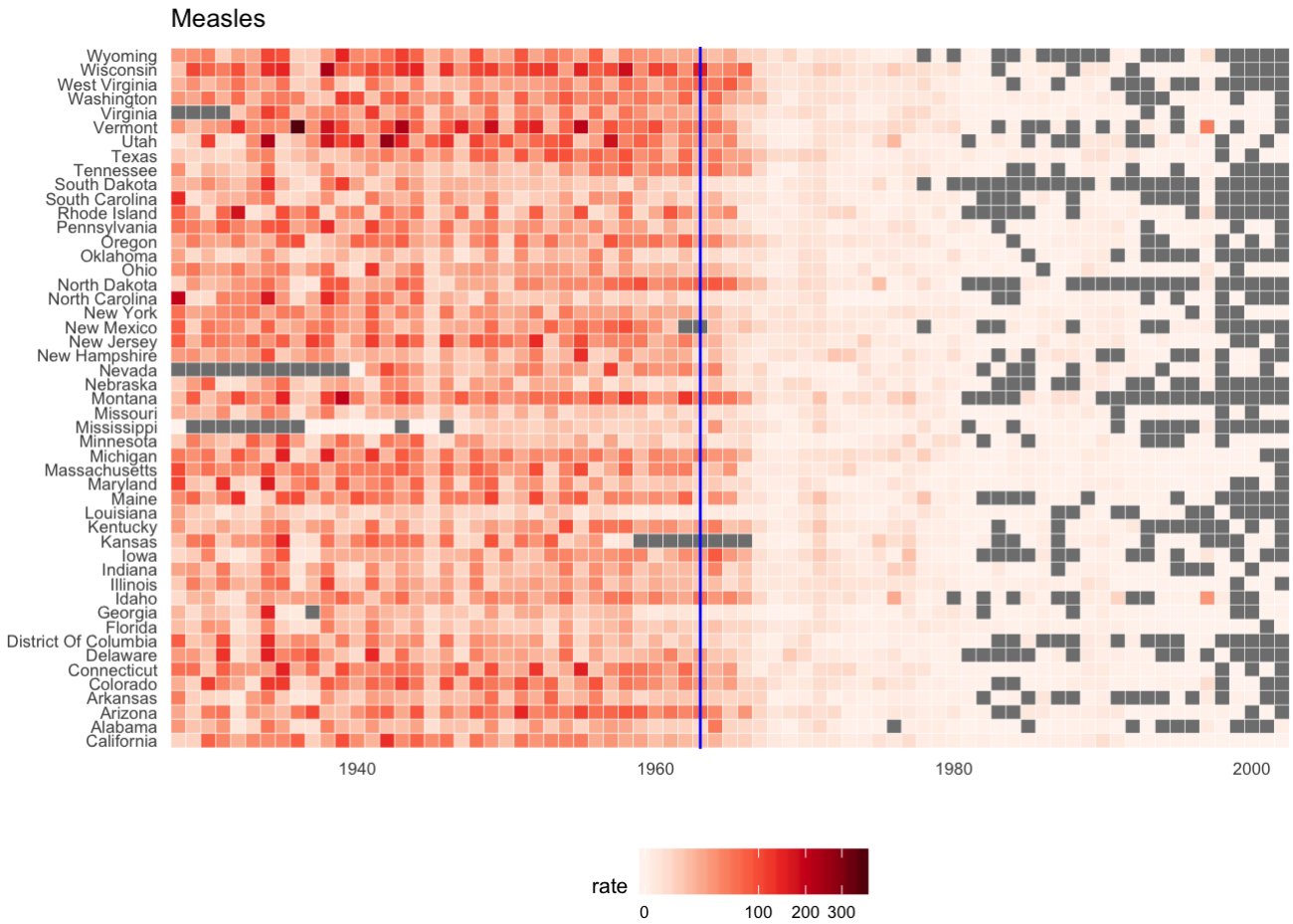
原始数据开根号再进行画图（但显示依然是原始数据），减少数据range，更好区分数据

画辅助线



Add a theme and a tile. Remove the xlab and ylab texts.

```
dat %>% ggplot(aes(year, state, fill = rate)) +
  geom_tile(color = "white") + scale_x_continuous(expand=c(0,0)) + scale_fill_gradien
tn(colors = brewer.pal(9, "Reds"), trans = "sqrt") + geom_vline(xintercept=1963, col
 = "blue") + theme_minimal() +
  theme(panel.grid = element_blank(),
        legend.position="bottom",       改变legend大小位置
        text = element_text(size = 8)) +
 ggtitle(the_disease) + ylab("") + xlab("") 去掉横轴纵轴
```



R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.