Biostatistics-Lecture4: Nonparametric hypothesis testing

Ruibin Xi

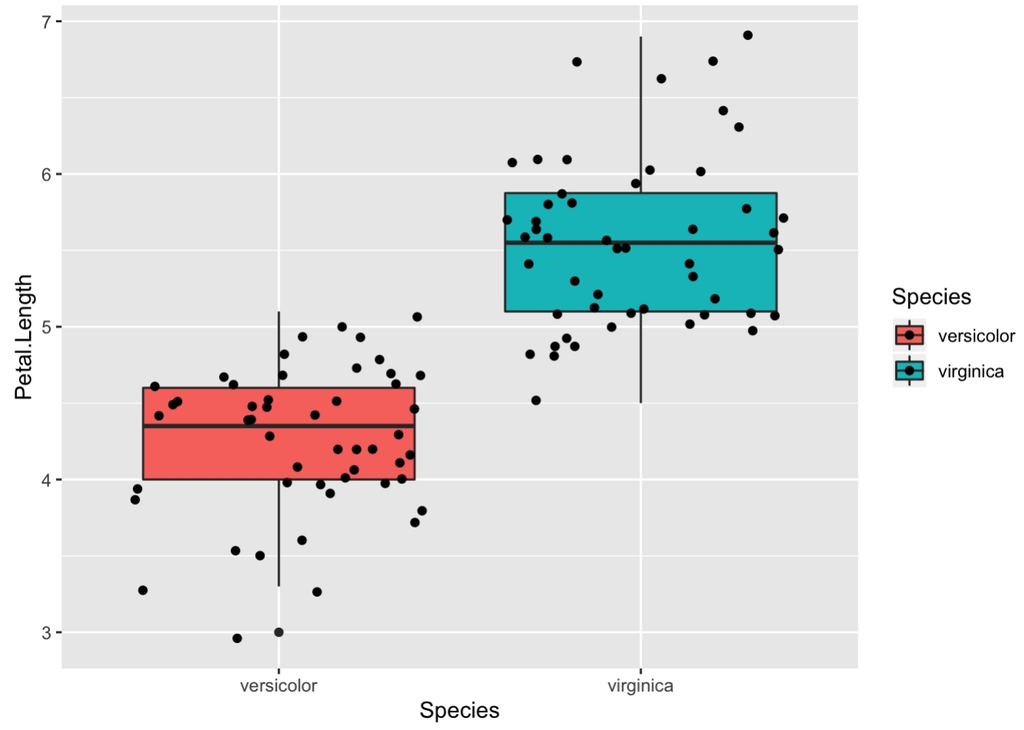
2/25/2020

```
library(ggplot2)
library(ggridges)
library(tidyverse)
library(purrr)
library(gridExtra)
library(SuppDists)
```

Nonparametric hypothesis tests

What can go wrong with parametric hypothesis tests. We consider the iris data. From the previous analysis, we know that the petal length of Virginica is larger than than of Versicolor.

```
data(iris)
iris %>% filter(Species!="setosa") %>%ggplot(aes(Species,Petal.Length,fill=Species))+geom_boxplot()+geom_jitter()
```



If we perform a two sample t-test to compare the mean petal length of Virginica and Versicolor, we get a significant p-value.

```
x <- iris %>% filter(Species=="virginica")
y <- iris %>% filter(Species=="versicolor")
t.test(x$Petal.Length,y$Petal.Length,alternative="greater")
```

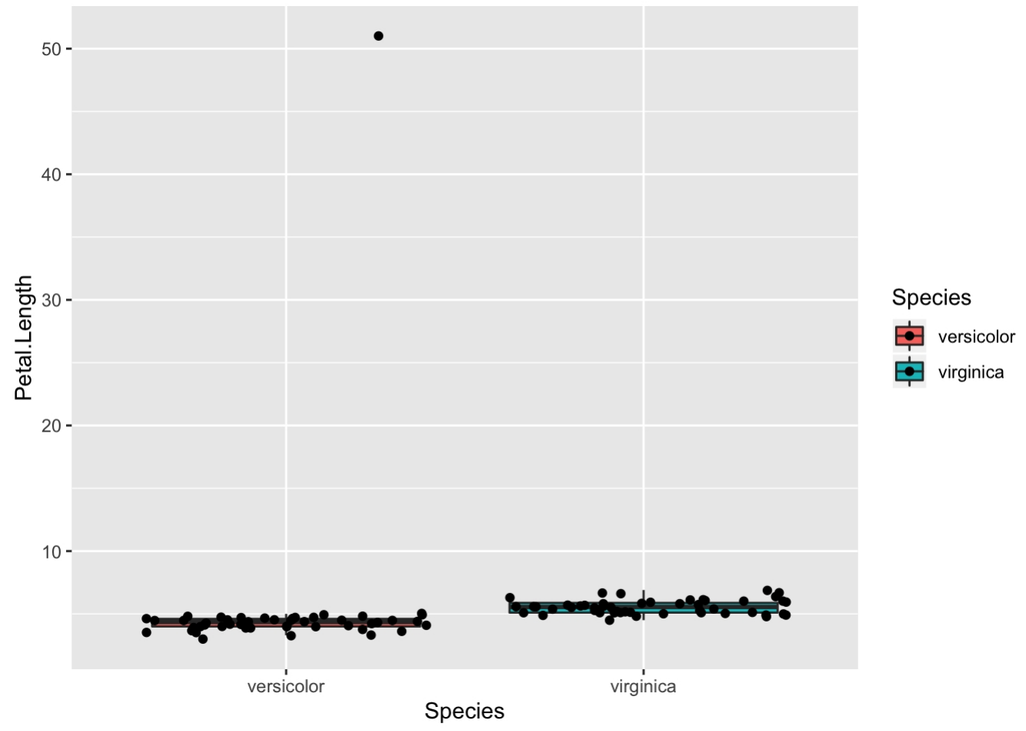
```
##
##  Welch Two Sample t-test
##
## data:  x$Petal.Length and y$Petal.Length
## t = 12.604, df = 95.57, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1.121737      Inf
## sample estimates:
## mean of x mean of y
##     5.552     4.260
```

Suppose that somehow the largest value of Versicolor petal length was mis-recorded as 51 (original was 5.1). Now the t-test gives an insignificant p-value. I have to use outlier.shape = NA to hide the outlier in the boxplot, otherwise there will be two points since we also used jitter plot.

```
max(y$Petal.Length)
```

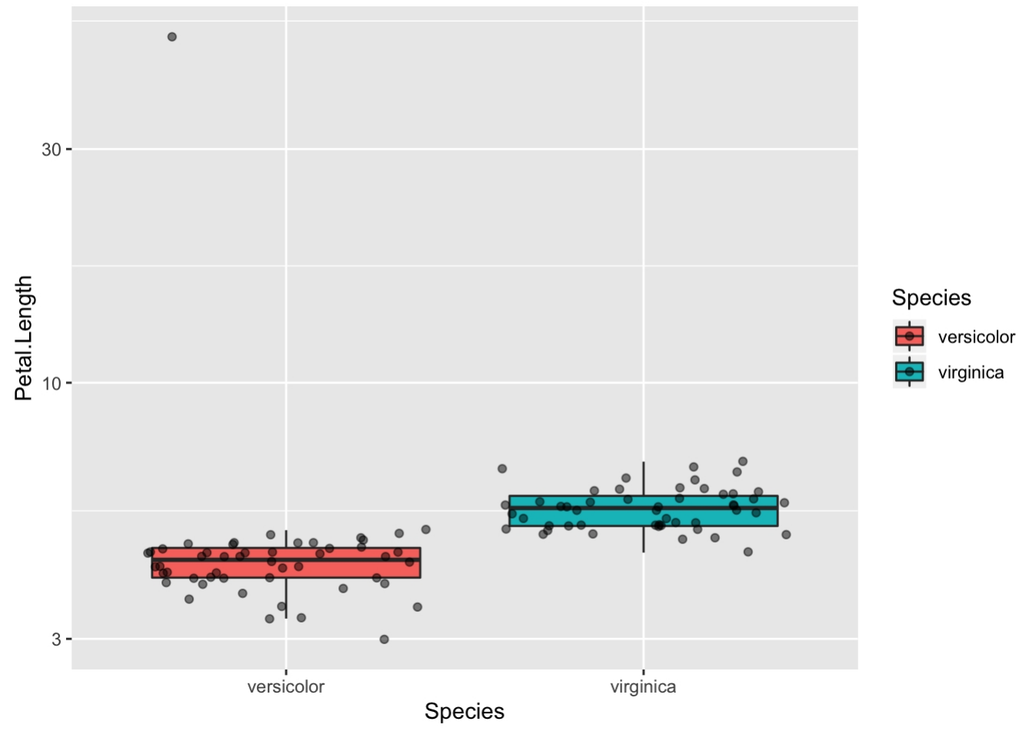
```
## [1] 5.1
```

```
y$Petal.Length[which.max(y$Petal.Length)] = 10*max(y$Petal.Length)
rbind(x,y)%>%ggplot(aes(Species,Petal.Length,fill=Species))+geom_boxplot(outlier.shape = NA)+geom_jitter()
```



In log10 scale

```
rbind(x,y)%>%ggplot(aes(Species,Petal.Length,fill=Species))+geom_boxplot(outlier.shape = NA)+geom_jitter(alpha=0.5)+scale_y_log10()
```




```
t.test(x$Petal.Length,y$Petal.Length,alternative="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  x$Petal.Length and y$Petal.Length
## t = 0.39762, df = 49.679, p-value = 0.3463
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -1.202529       Inf
## sample estimates:
## mean of x mean of y
##     5.552     5.178
```

万一有一个点记录错误，
t-test结果影响很大

If we instead use nonparametric test, we still get a significant result.

```
wilcox.test(x$Petal.Length,y$Petal.Length,alternative="greater")
```

默认unpaired

```
##
##  Wilcoxon rank sum test with continuity correction
##
```

```
## data:  x$Petal.Length and y$Petal.Length
## W = 2418, p-value = 3.939e-16
## alternative hypothesis: true location shift is greater than 0
```

规避了outier的影响

If the distribution deviates from normal distribution, it seems that t-test often can largely control the false positives, but its power is significantly influenced. Let's perform simulations to show this. First consider the NULL hypothese setting.

```
n=200
set.seed(3332)
alpha = 0.05
pvalue.t = c()
pvalue.w = c()
for(i in 1:100){
  x = rcauchy(n)
  y = rcauchy(n)
  rslt.t = t.test(x,y)
  rslt.w = wilcox.test(x,y)
  pvalue.t[i] = rslt.t$p.value
  pvalue.w[i] = rslt.w$p.value
  }
```

The type I error rate of t-test

```
sum(pvalue.t<alpha)/100
```

```
## [1] 0.02
```

The type I error rate of Wilcoxon test

```
sum(pvalue.w<alpha)/100
```

```
## [1] 0.06
```

Now we consider the alternative hypothesis.

```
set.seed(3332)
pvalue.t = c()
pvalue.w = c()
for(i in 1:100){
  x = rcauchy(n)
  y = rcauchy(n,location=2)
  rslt.t = t.test(x,y)
  rslt.w = wilcox.test(x,y)
  pvalue.t[i] = rslt.t$p.value
  pvalue.w[i] = rslt.w$p.value
  }
```

The power of t-test

```
sum(pvalue.t<alpha)/100
```

```
## [1] 0.23
```

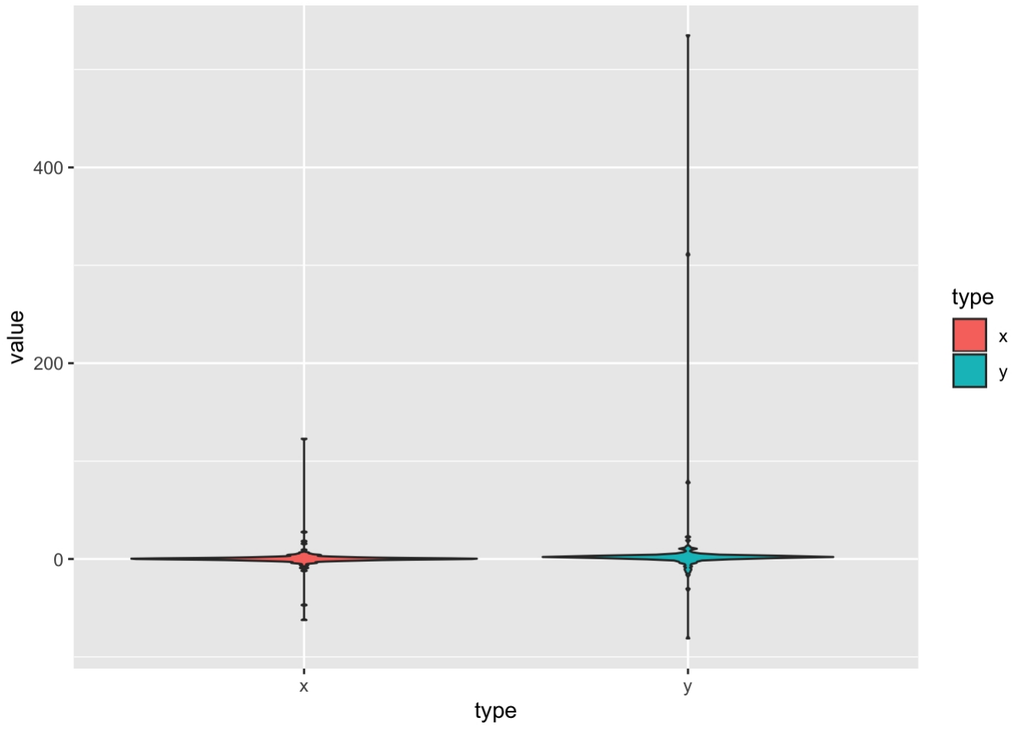
The power of Wilcoxon test

```
sum(pvalue.w<alpha)/100
```

```
## [1] 1
```

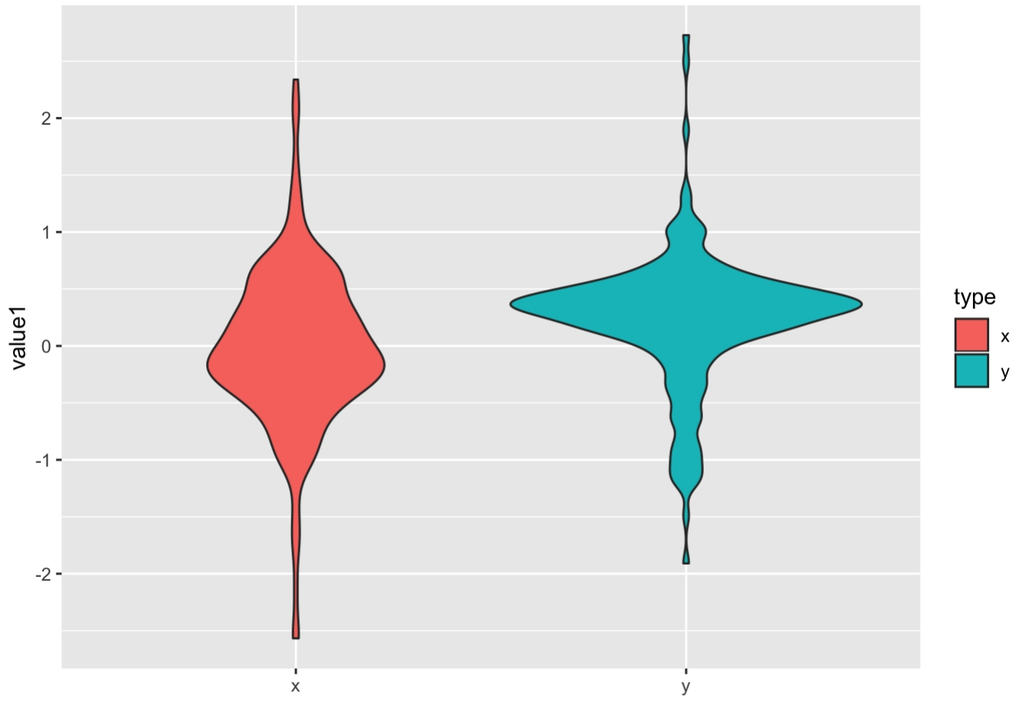
Let’s look at the data of the last simulation

```
data.frame(value=c(x,y),type=c(rep("x",n),rep("y",n))) %>% ggplot(aes(type,value,fill=type))+geom_violin()
```



It is is a bit difficult to see the difference, let’s make a transformation

```
data.frame(value=c(x,y),type=c(rep("x",n),rep("y",n))) %>% mutate(value1=sign(value)*log10(abs(value))) %>% ggplot(aes(type,value1,fill=type))+geom_violin()
```



The pvalue of the t test is

```
pvalue.t[100]
```

```
## [1] 0.1014825
```

The pvalue of the Wilcoxon test is

```
pvalue.w[100]
```

```
## [1] 5.545308e-16
```

Let us consider t distribution. First consider the NULL hypothese setting.

```
n = 50
set.seed(3332)
alpha = 0.05
pvalue.t = c()
pvalue.w = c()
for(i in 1:100){
  x = rt(n,df=1)
```

```
  y = rt(n,df=2)
  rslt.t = t.test(x,y)
  rslt.w = wilcox.test(x,y,exact=TRUE)
  pvalue.t[i] = rslt.t$p.value
  pvalue.w[i] = rslt.w$p.value
  }
```

The type I error rate of t-test

```
sum(pvalue.t<alpha)/100
```

```
## [1] 0.02
```

The type I error rate of Wilcoxon test

```
sum(pvalue.w<alpha)/100
```

```
## [1] 0.03
```

Now we consider the alternative hypothesis.

```
set.seed(3332)
pvalue.t = c()
pvalue.w = c()
for(i in 1:100){
  x = rt(n,df=1.5)
  y = rt(n,df=2) + 1
  rslt.t = t.test(x,y)
  rslt.w = wilcox.test(x,y,exact=TRUE)
  pvalue.t[i] = rslt.t$p.value
  pvalue.w[i] = rslt.w$p.value
  }
```

rt() 模拟t分布，自由度为1.5 和 2 的数据

The power of t-test

```
sum(pvalue.t<alpha)/100
```

```
## [1] 0.32
```

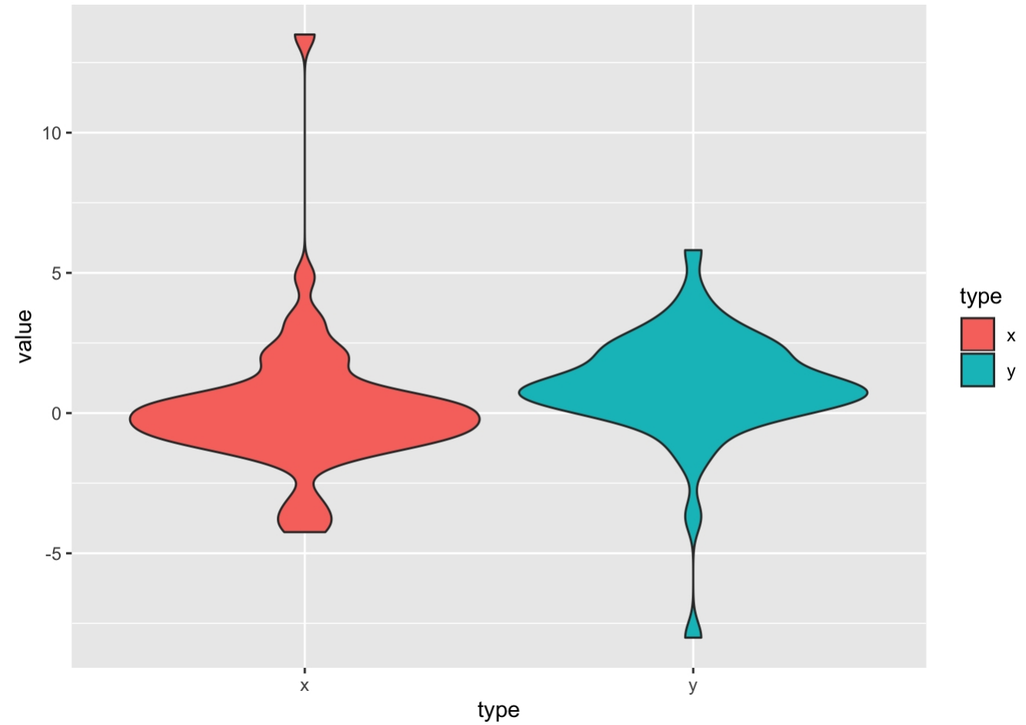
The power of Wilcoxon test

```
sum(pvalue.w<alpha)/100
```

```
## [1] 0.89
```

Let’s look at the data of the last simulation

```
data.frame(value=c(x,y),type=c(rep("x",n),rep("y",n))) %>% ggplot(aes(type,value,fill=type))+geom_violin()
```



The pvalue of the t test is

```
pvalue.t[100]
```

```
## [1] 0.1073579
```

The pvalue of the Wilcoxon test is

```
pvalue.w[100]
```

```
## [1] 0.000211853
```

Fisher’s Tea Drinker data.

A British woman claimed to be able to distinguish whether milk or tea was added to the cup first. To test, she was given 8 cups of tea, in four of which milk was added first. The null hypothesis is that there is no association between the true order of pouring and the woman's guess, the alternative that there is a positive association (that the odds ratio is greater than 1).

```
TeaTasting <-
matrix(c(3, 1, 1, 3),
       nrow = 2,
       dimnames = list(Guess = c("Milk", "Tea"),
                       Truth = c("Milk", "Tea")))
TeaTasting
```

造数据

```
##       Truth
## Guess  Milk Tea
##   Milk    3   1
##   Tea     1   3
```

```
fisher.test(TeaTasting, alternative = "greater")
```

H1:大于0

```
##
##  Fisher's Exact Test for Count Data
##
## data:  TeaTasting
## p-value = 0.2429
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  0.3135693       Inf
## sample estimates:
## odds ratio
##   6.408309
```

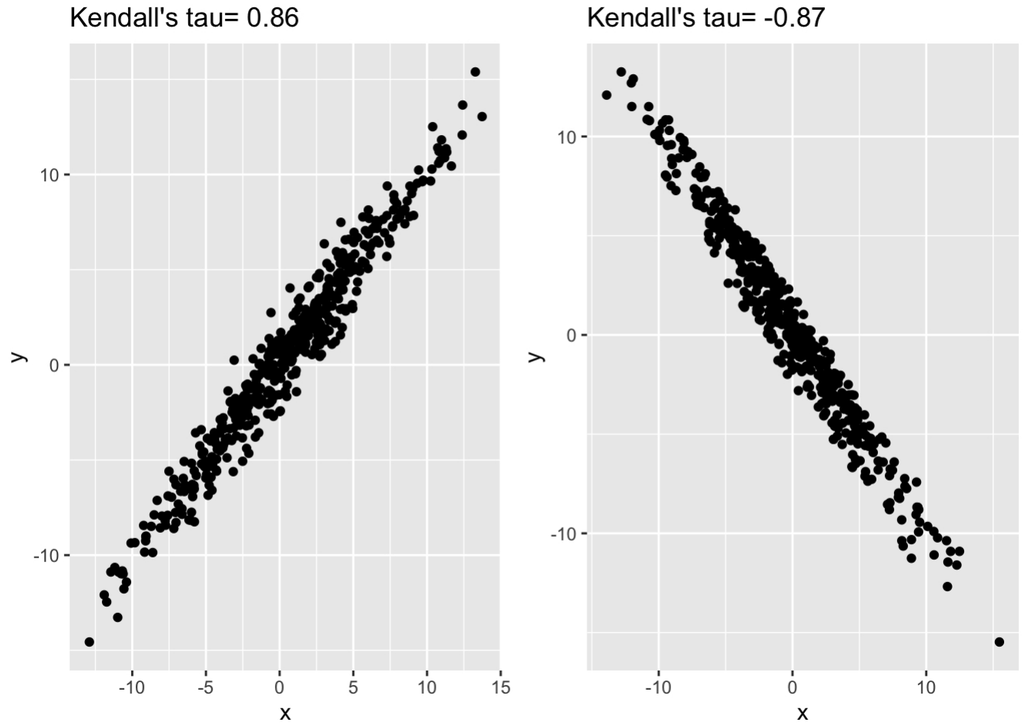
Now we consider the nonparametric correlations.

```
set.seed(12345)
n = 500
dta1 = data.frame(x=rnorm(n,sd=5)) %>% mutate(y=x+rnorm(100))
dta2 = data.frame(x=rnorm(n,sd=5)) %>% mutate(y= -x+rnorm(100))
bp1 <- dta1 %>% ggplot(aes(x,y))+geom_point()
bp2 <- dta2 %>% ggplot(aes(x,y))+geom_point()
```

模拟线性单调关系数据

```
dta1.cor.kendall = cor(dta1$x,dta1$y,method="kendal")
dta2.cor.kendall = cor(dta2$x,dta2$y,method="kendal")
```

```
m <- grid.arrange(bp1 + ggtitle(paste("Kendall's tau=",format(dta1.cor.kendall,digits=2))), bp2+ggtitle(paste("Kendall's tau=",format(dta2.cor.kendall,digits=2))),n
```



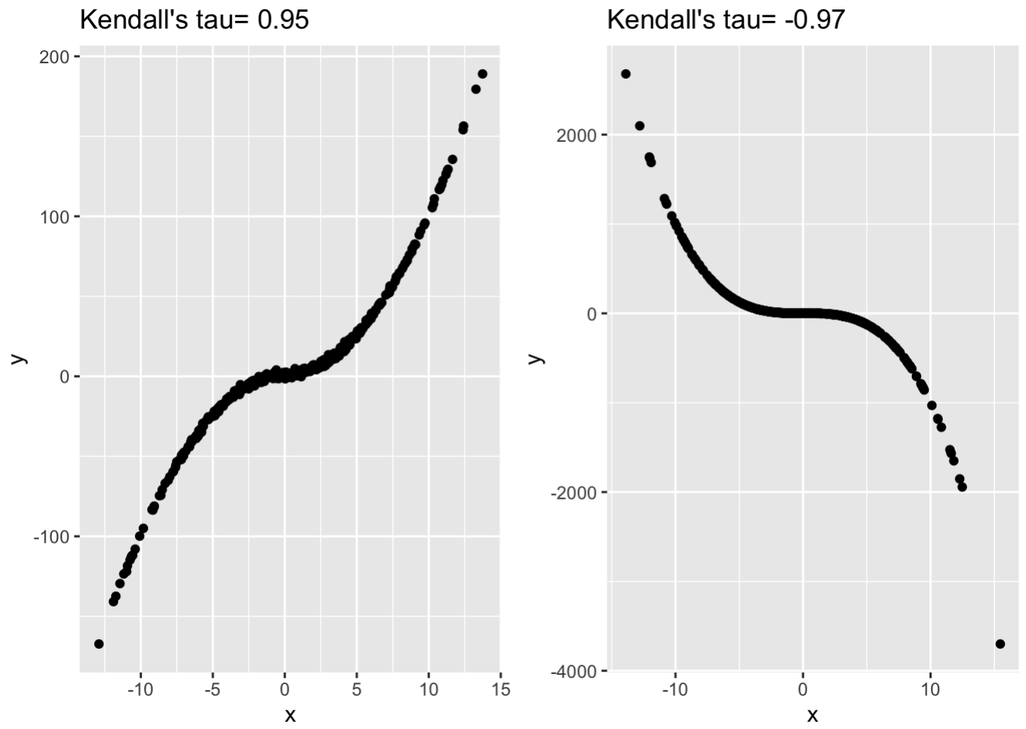
Nonlinear relationship.

```
set.seed(12345)
n = 500
dta1 = data.frame(x=rnorm(n,sd=5)) %>% mutate(y=sign(x)*x^2+1+rnorm(100))
dta2 = data.frame(x=rnorm(n,sd=5)) %>% mutate(y= -x^3+1+rnorm(100))
bp1 <- dta1 %>% ggplot(aes(x,y))+geom_point()
bp2 <- dta2 %>% ggplot(aes(x,y))+geom_point()
```

模拟非线性的单调关系数据

```
dta1.cor.kendall = cor(dta1$x,dta1$y,method="kendal")
dta2.cor.kendall = cor(dta2$x,dta2$y,method="kendal")
```

```
m <- grid.arrange(bp1 + ggtitle(paste("Kendall's tau=",format(dta1.cor.kendall,digits=2))), bp2+ggtitle(paste("Kendall's tau=",format(dta2.cor.kendall,digits=2))),n
```



An Real example

We consider the drug test experiments in Yin et al. 2020. For each patient, we have to evaluations. One is the pathological response evaluated using the Miller & Payne (M&P) classification system. Another is the drug test resulst based on the cancer cells cultured from the tumors of patients

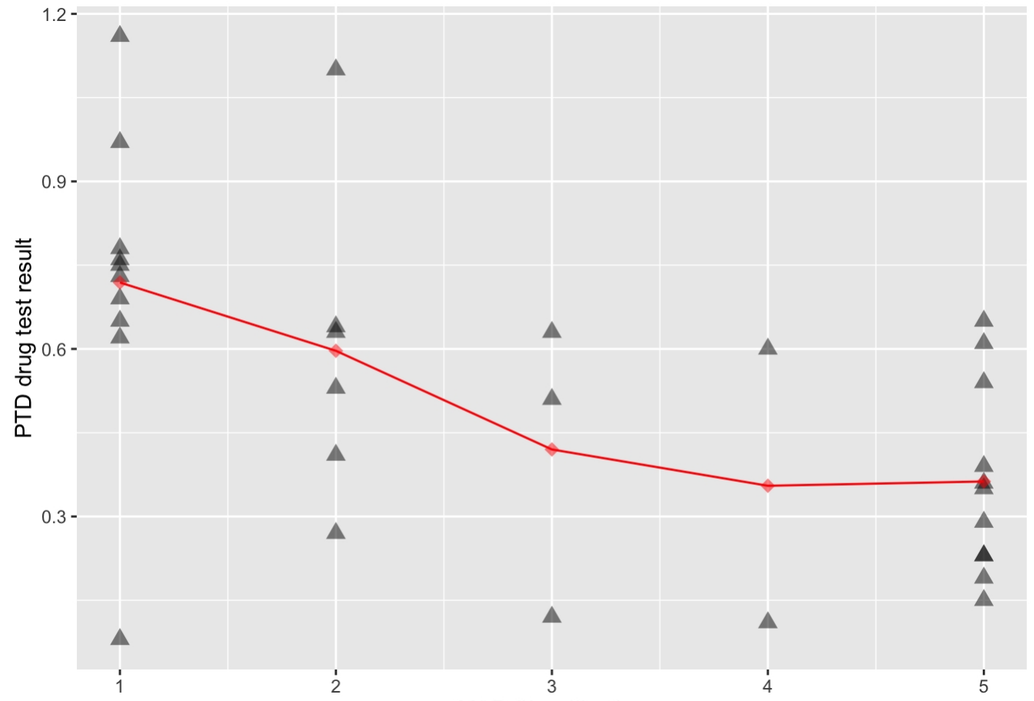
```
MP=c(1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,4,4,5,5,5,5,5,5,5,5,5,5)
test.rslt = c(1.16,0.97,0.78,0.76,0.75,0.73,0.69,0.65,0.62,0.08,1.1,0.64,0.63,0.53,0.41,0.27,0.63,0.51,0.12,0.6,0.11,0.65,0.61,0.54,0.39,0.36,0.35,0.29,0.23,0.23,0.
drugtest = data.frame(MP=MP,test=test.rslt)
```

We calculate a mean test value for each MP level.

```
drugtestMean <- drugtest %>%group_by(MP) %>% summarize(testmean = mean(test))
```

We first plot the

```
size = 3
bp1 <- drugtest %>% ggplot(aes(y=test,x=MP)) + geom_point(alpha=0.5,shape=17,size=size)
bp1 <- bp1 + geom_point(mapping=aes(y=testmean,x=MP),data=drugtestMean,alpha=0.5,shape=18,col="red",size=size) + geom_line(mapping=aes(y=testmean,x=MP),data=drugtes
bp1 + ylab("PTD drug test result") + xlab("M&P Classification")
```



M&P Classification

We calculate the Kendall's tau and its p-value

```
cor.kenall = cor(drugtest$MP,drugtest$test,method="kendall")
cor.kenall
```

```
## [1] -0.4662318
```

We could use the cor.test function to get the p-value

```
cor.test(drugtest$test,drugtest$MP,method="kendall",alternative="less",exact=TRUE)
```

```
##
##  Kendall's rank correlation tau
##
## data:  drugtest$test and drugtest$MP
## z = -3.4085, p-value = 0.0003265
## alternative hypothesis: true tau is less than 0
## sample estimates:
##       tau
## -0.4662318
```

Alternatively, the one-sided p-value can be obtained using the function pKendall. Note that this p-value and the above p-value are different. This is because pKendall and cor.test use different algorithms to calculate the approximation of the probability. It seems pKendall gives more accurate p-value.

```
pKendall(cor.kenall,N=nrow(drugtest),lower.tail=TRUE)
```

这个算kenall p值会更准确

```
## [1] 5.18113e-05
```

Similary, we can also consider the Spearman's rho correlation.

```
cor.spearman = cor(drugtest$MP,drugtest$test,method="spearman")
cor.spearman
```

```
## [1] -0.5979665
```

The p-value by the cor.test.

```
cor.test(drugtest$test,drugtest$MP,method="spearman",alternative="less",exact=TRUE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  drugtest$test and drugtest$MP
## S = 8718.5, p-value = 0.0001506
## alternative hypothesis: true rho is less than 0
## sample estimates:
##       rho
## -0.5979665
```

算法不一样导致p值结果不一样，但差别一般不会太大

Alternaviely, using pSpearman

```
pSpearman(cor.spearman,r=nrow(drugtest),lower.tail=TRUE)
```

```
## [1] 0.000149836
```

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.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.