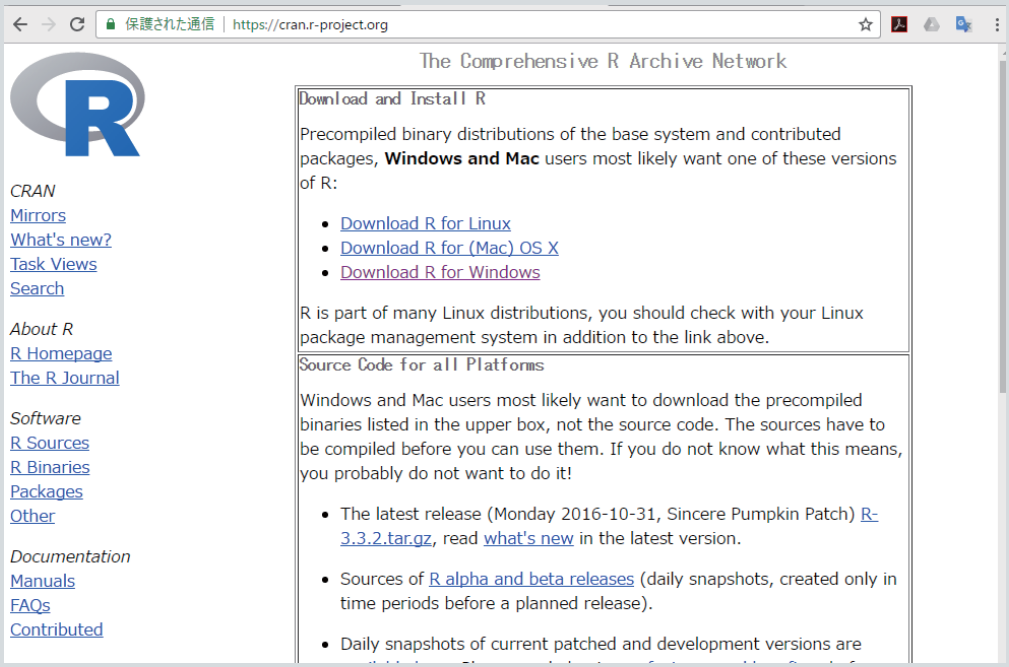
IR実務者のためのR入門

山本義郎（東海大学）
日本計算機統計学会

本日の内容

- RとRstudioのインストール
- Rを使ったら何ができるか
- Rの基本的な操作
- Rを使う環境
  - Rコマンダー（コマンドを知らなくてもOK）
  - Rstudio（Rを使いこなすなら是非）
- 実データの解析例

Rのインストール



CRAN
Rstudio　を検索

Rを使ったら何ができるか（1）

- データ解析

Excelから　Rへ

Excelが得意なこと

・集計（ピポットテーブル）
・グラフ（棒、折れ線、円グラフ）
・関数を使ったデータの変換
・フィルター
・データベース関数
・・・

Rが得意なこと

・大量データの処理
・データ解析（統計解析）
・グラフ・統計グラフ
・GIS（地理情報処理）
・プログラミング
・シミュレーション
・ルーチンワーク

Rを使ったら何ができるか（2）

- レポーティング

- WebページやPDFファイルに
- インタラクティブな機能も

EXCELよりRを使うべき理由

1. 大量データの処理
2. データ解析（統計解析）
3. グラフ・統計グラフ
4. GIS（地理情報処理）
5. プログラミング
6. シミュレーション
7. ルーチンワーク

Rを使うべき理由（1）

1. 大量データの処理

- Excelのデータも読み込める
- データベースからのデータのインポート
- データの抽出（フィルタ）などは縦横無尽

Rを使うべき理由（2）

2. データ解析（統計解析）

- 検定、分散分析、多重比較
- 統計データの視覚化（様々な統計グラフ）
- 多変量データ解析
  - 回帰分析、クラスター分析、主成分分析、、
- データマイニング
  - アソシエーションルール分析、決定木分析、、

Rを使うべき理由（3）

3. グラフ・統計グラフ

- 統計グラフ
  - 箱ひげ図、散布図行列、、、
- 統計処理のグラフ
  - 回帰直線、デンドログラム、対応分析、、、
- 大量のグラフ作成もプログラミングで容易に
- インタラクティブなグラフ作成も可能

Rを使うべき理由（4）

4. GIS（地理情報処理）

- 地理情報システム並の処理ができる
  - 塗り分け地図（コロプレスマップ）
- GoogleMapやGoogleEarthの利用も
- QGISなどのGISソフトウェアとの連携も

Rを使うべき理由（5）

5. プログラミング

- CやJavaよりも気軽にプログラミングができる
- シミュレーション
- ルーチンワークにマクロとして使える

Rを使うべき理由（6）

6. シミュレーション

- プログラミング言語でのルーチンワーク

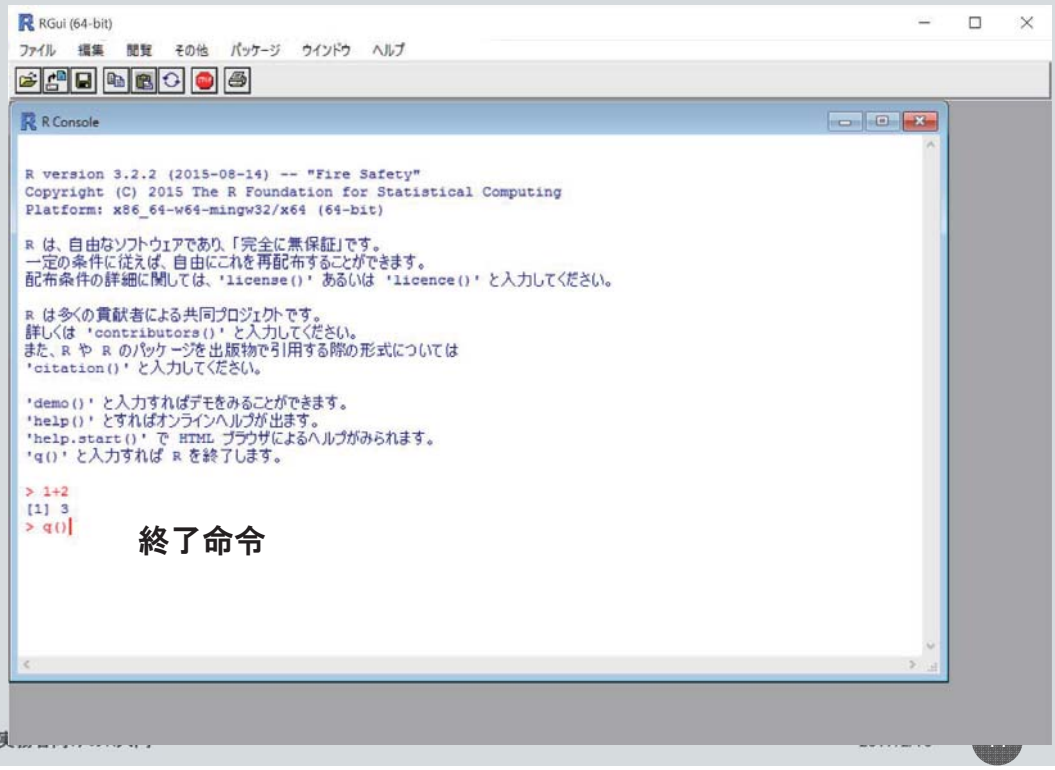
Rを使うべき理由（7）

7. ルーチンワーク

- 定形処理が得意

Rの起動と終了

スタートメニュー（デスクトップ）から、「R x64 3.3.2」を起動



終了命令

Rの基本(1) 関数電卓・グラフ

RGui (32-bit)

ファイル　履歴　サイズ変更　ウインドウ

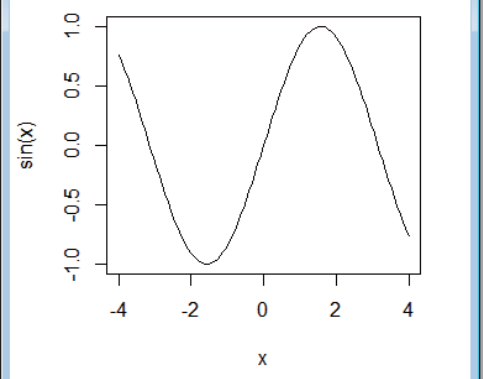
R Console

```
 'demo()' と入力すればデモをみることができます。
 'help()' とすればオンラインヘルプが出ます。
 'help.start()' で HTML ブラウザによるヘルプがみられます。
 'q()' と入力すれば R を終了します。

 [以前にセーブされたワークスペースを復帰します]

> 1+1
[1] 2
> sin(pi/2)
[1] 1
> exp(1)
[1] 2.718282
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> sin((1:10)*pi/4)
 [1]  7.071068e-01  1.000000e+00  7.071068e-01  1.224606e-16 -7.071068e-01
 [6] -1.000000e+00 -7.071068e-01 -2.449213e-16  7.071068e-01  1.000000e+00
> plot(sin((1:10)*pi/4))
> plot((1:10)*pi/4,sin((1:10)*pi/4))
> plot(sin)
> curve(sin,-4,4)
>
```

R Graphics: Device 2 (ACTIVE)



Rの基本(2)統計処理・データ解析

R Console

```
> 1+2
[1] 3
> mean(c(148, 160, 159, 153, 151, 140, 158, 137, 149, 160))
[1] 151.5
> height <- c(148, 160, 159, 153, 151, 140, 158, 137, 149, 160)
> height
 [1] 148 160 159 153 151 140 158 137 149 160
> mean(height)
[1] 151.5
> sex <- c("F","M","M","F","F","F","M","F","M","F")
> tapply(height, sex, mean)
       F        M
148.1667 156.5000
> mydata <- data.frame(height,sex)
> mydata
   height sex
1     148   F
2     160   M
3     159   M
4     153   F
5     151   F
6     140   F
7     158   M
8     137   F
9     149   M
10    160   F
> mydata$height
 [1] 148 160 159 153 151 140 158 137 149 160
> mean(mydata$height)
[1] 151.5
>
```

Rの基本(2) 統計処理・データ解析

RGui (32-bit)

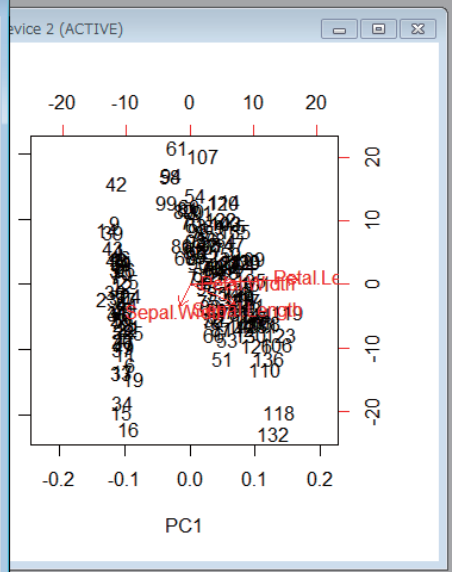
ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

R Console

```
> 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(iris)
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa     :50
 versicolor:50
 virginica :50


> iris.pca<-prcomp(iris[,1:4])
> plot(iris.pca)
> biplot(iris.pca)
> iris.pca
Standard deviations:
[1] 2.0562689 0.4926162 0.2796596 0.1543862

Rotation:
```



Rの基本(3) データの読み込み

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

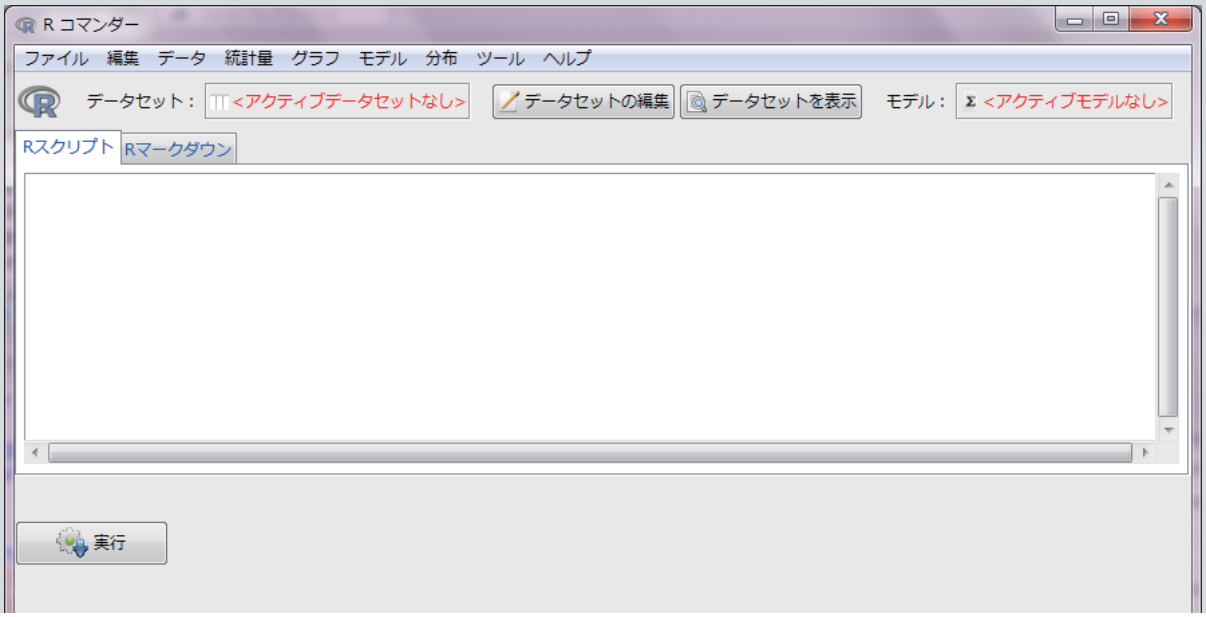
R Console

```
> hitting <- read.csv("NPB年度別打撃結果.csv")
> summary(hitting)
      年度                 球団          左右                選手名          試合数           打席数
 Min.   :2005   中日          : 674   右:4551   竹原 直隆   :  13   Min.   :  1.00   Min.   :  0.0
 1st Qu.:2008   オリックス    : 665   左:2860   長田 秀一郎:  13   1st Qu.:  9.00   1st Qu.:  0.0
 Median :2011   東北楽天      : 651   両: 222   鶴岡 一成   :  13   Median : 26.00   Median : 12.0
 Mean   :2011   読売          : 634             矢野 謙次   :  13   Mean   : 41.71   Mean   :102.1
 3rd Qu.:2014   阪神          : 630             カブレラ    :  12   3rd Qu.: 62.00   3rd Qu.:114.0
 Max.   :2016   福岡ソフトバンク: 626           ゴンザレス  :  12   Max.   :146.00   Max.   :692.0
                (Other)       :3753             (Other)     :7557
      打数             得点             安打             二塁打           三塁打           本塁打
 Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000
 1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.000
 Median : 10.00   Median :  0.00   Median :  1.00   Median : 0.000   Median : 0.0000   Median : 0.000
 Mean   : 90.39   Mean   : 10.68   Mean   : 23.44   Mean   : 3.982   Mean   : 0.3867   Mean   : 2.135
 3rd Qu.:102.00   3rd Qu.: 11.00   3rd Qu.: 23.00   3rd Qu.: 4.000   3rd Qu.: 0.0000   3rd Qu.: 1.000
 Max.   :623.00   Max.   :121.00   Max.   :216.00   Max.   :47.000   Max.   :13.0000   Max.   :60.000

      犠打             打点             盗塁             盗塁刺           犠打             犠飛
 Min.   :  0.0   Min.   :  0.00   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000   Min.   : 0.0000
 1st Qu.:  0.0   1st Qu.:  0.00   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.0000
 Median :  2.0   Median :  0.00   Median : 0.000   Median : 0.0000   Median : 0.000   Median : 0.0000
 Mean   : 34.6   Mean   : 10.19   Mean   : 1.517   Mean   : 0.6841   Mean   : 2.308   Mean   : 0.6078
 3rd Qu.: 33.0   3rd Qu.:  9.00   3rd Qu.: 1.000   3rd Qu.: 0.0000   3rd Qu.: 2.000   3rd Qu.: 0.0000
 Max.   :347.0   Max.   :147.00   Max.   :60.000   Max.   :21.0000   Max.   :62.000   Max.   :11.0000
                                                   NA's   :1         NA's   :1        NA's   :1
      四球             故意四           死球             三振           併殺打
 Min.   :  0.000   Min.   : 0.0000   Min.   : 0.000   Min.   :  0.00   Min.   : 0.000
 1st Qu.:  0.000   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.:  0.00   1st Qu.: 0.000
```

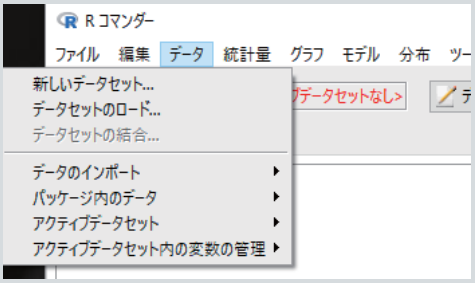
Rコマンダー

- Rcmrパッケージ
- コマンドを覚えなくてもメニューで使える

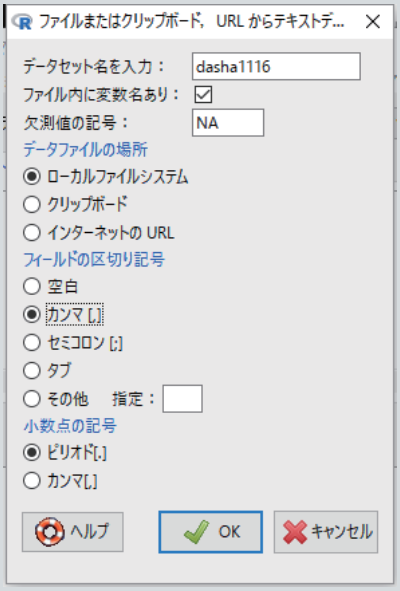


Rコマンダーでできること(1)

- データの作成

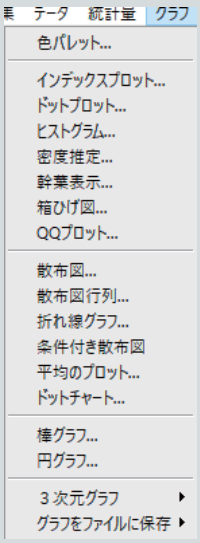


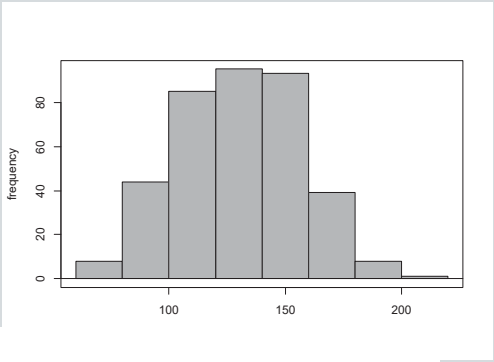
インポート



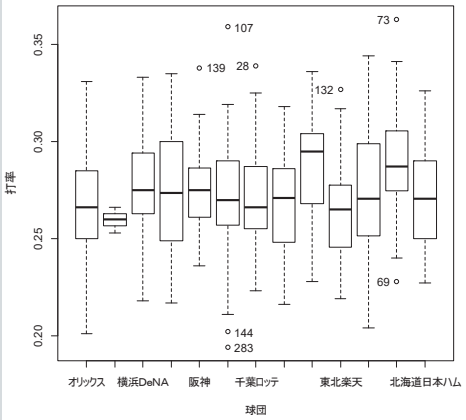
Rコマンダーでできること(2)

- グラフ





ヒストグラム

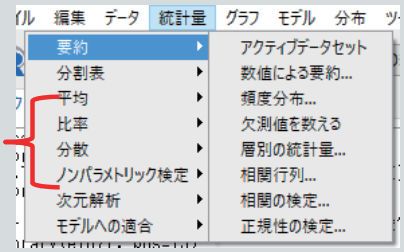


箱ひげ図

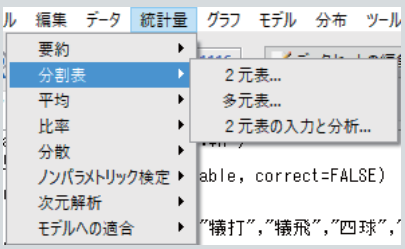
Rコマンダーでできること(3)

- 記述統計・検定

検定



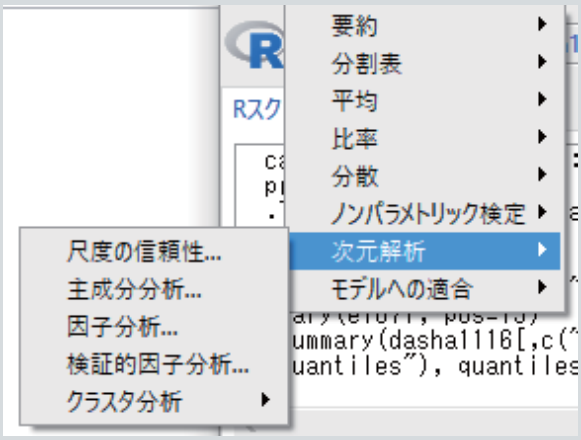
量的変数の要約



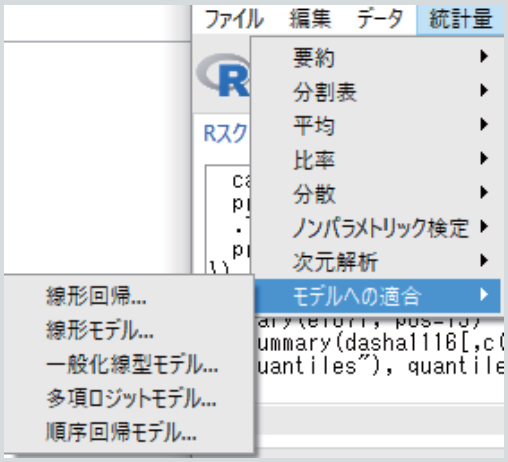
質的変数の要約

Rコマンダーでできること(4)

- 多変量データ解析



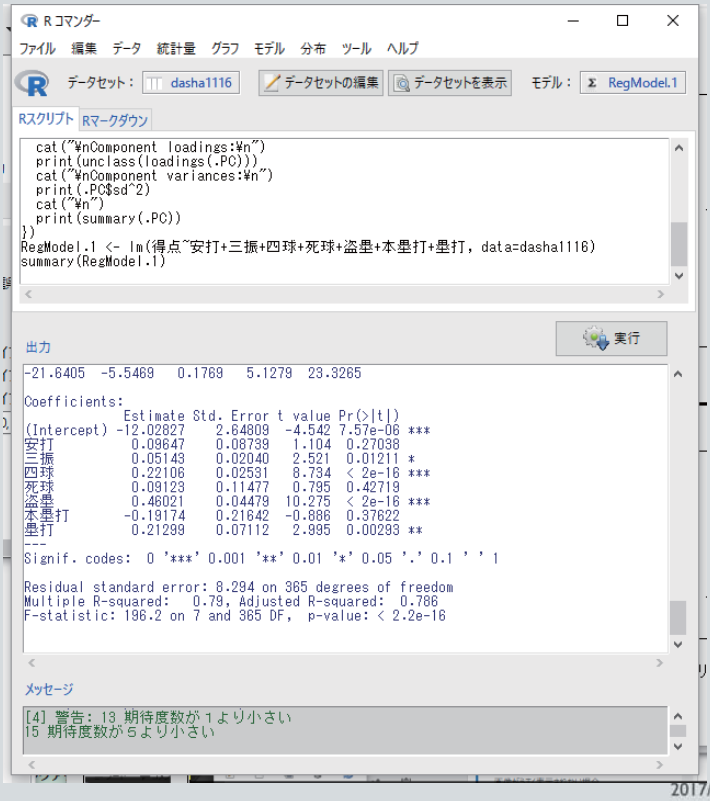
分類（クラスタリング）
次元縮約



予測

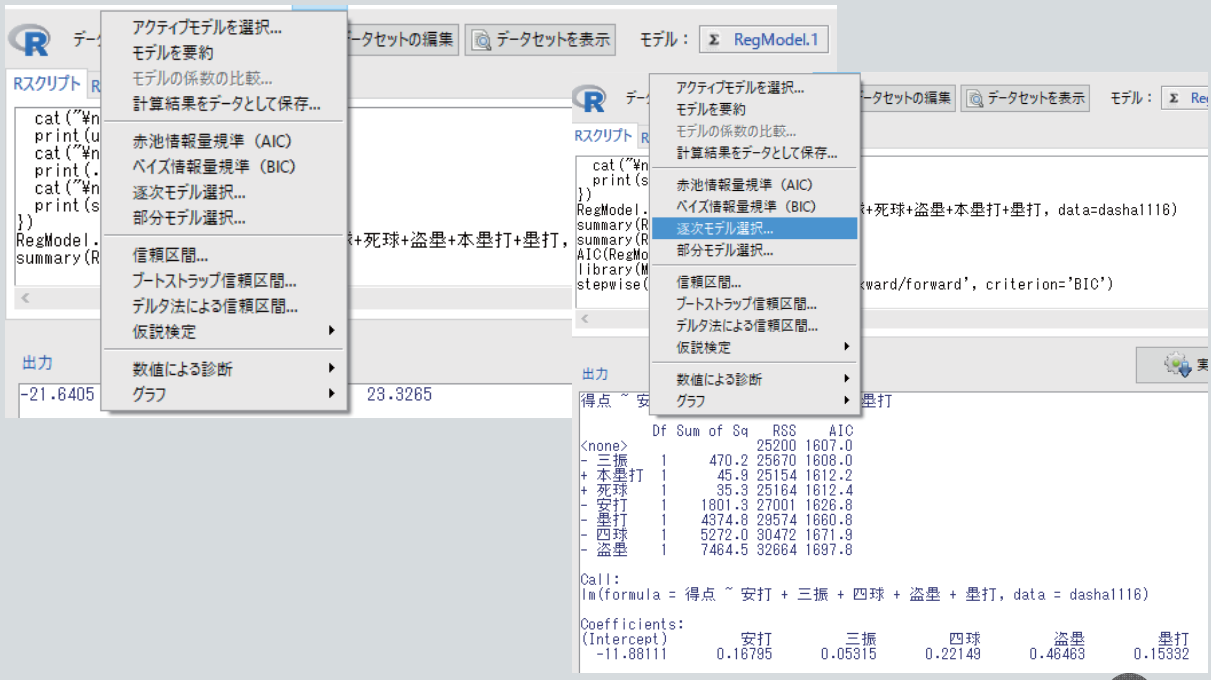
Rコマンダーでできること(4)

- 線形重回帰



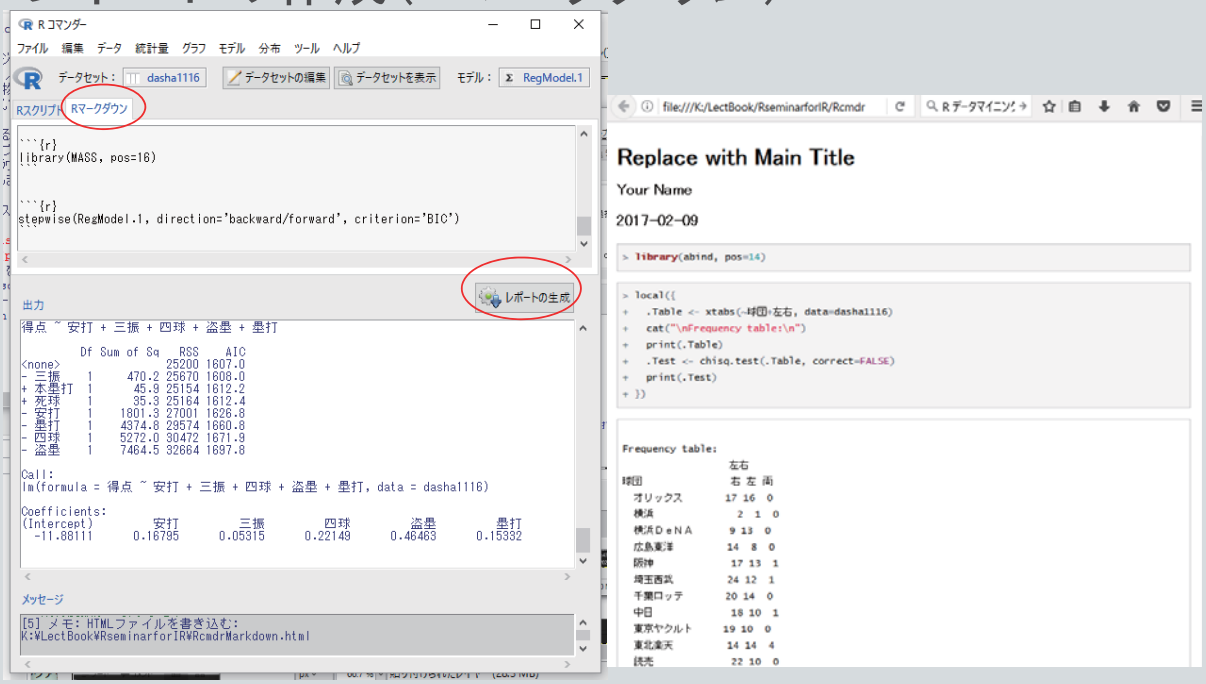
Rコマンダーでできること(4)

- 線形重回帰モデル



Rコマンダーでできること(5)

- レポートの作成(Rマークダウン)



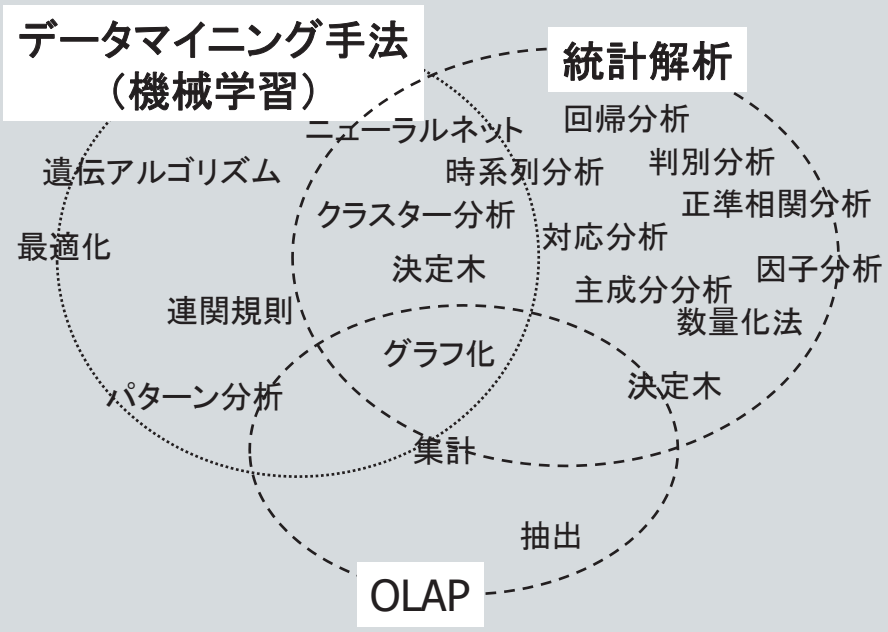
データマイニング（目的と手法）

目的	手法	適用例(マーケティング)
予測 （判別予測 と数値予測）	・決定木 ・ニューラルネット ・ロジスティック回帰 ・重回帰分析（線形・非線形） ・判別分析	・スコアリング ・チャーン ・不正発見 ・リスク管理
分類 セグメンテーション	・自己組織化マップ ・クラスター分析 ・ニューラルネット ・主成分分析 ・コレスポンデンス分析 ・決定木	・優良顧客の属性 ・市場セグメンテーション
関連性の発見 （リンク分析）	・連関規則 ・時系列パターン分析 ・主成分分析 ・コレスポンデンス分析	・マーケットバスケット分析

データマイニング・統計手法・OLAP

	データマイニング	統計解析	OLAP
分析目的	探索的解析 目的志向解析	目的志向解析 （探索的解析）	履歴データ
分析プロセス	パターンの発見と検証	仮説検証	主観に基づく集計
難易度	簡易に実行可能だが、各技術の基本的知識が必要。 （半自動）	各手法の基本的知識および統計に関する知識が必要。 （対話的） モデルが扱い易い	容易（対話的）
手法	機械学習（NN） リンク分析 グラフ化	多変量解析 グラフ化	集計 抽出

データマイニング・統計手法・OLAP

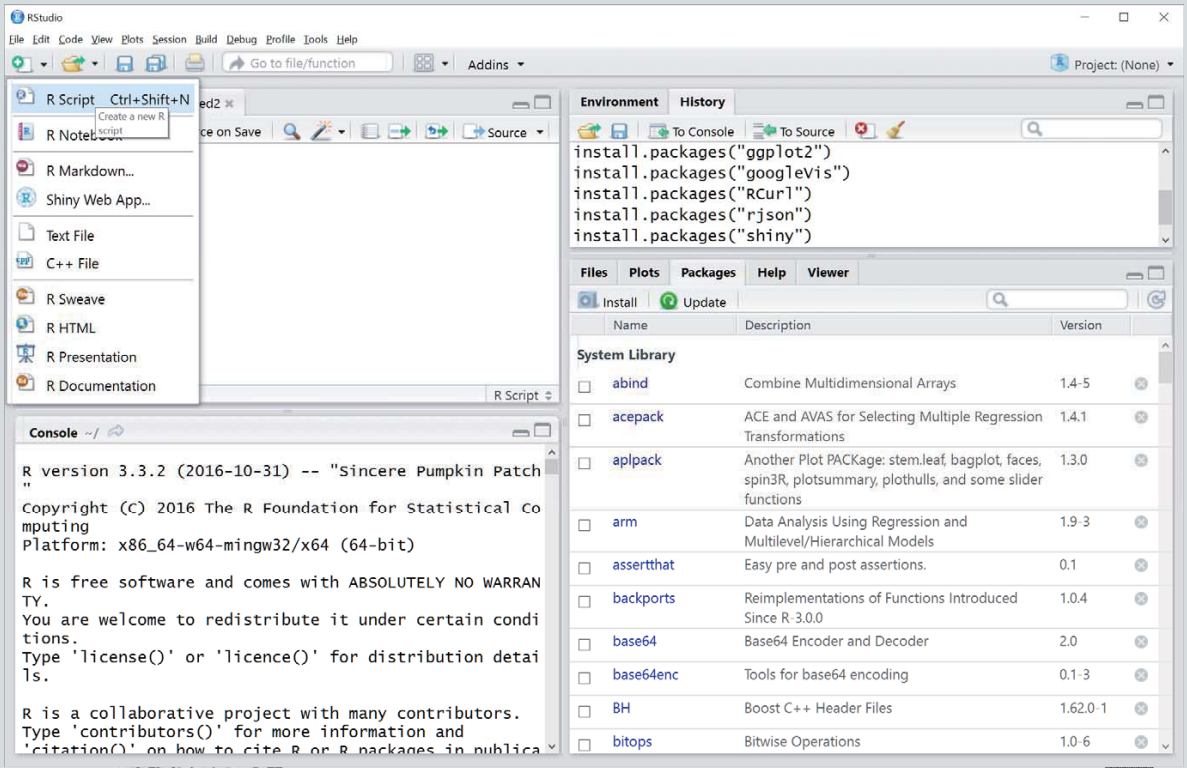


Rでのデータマイニング

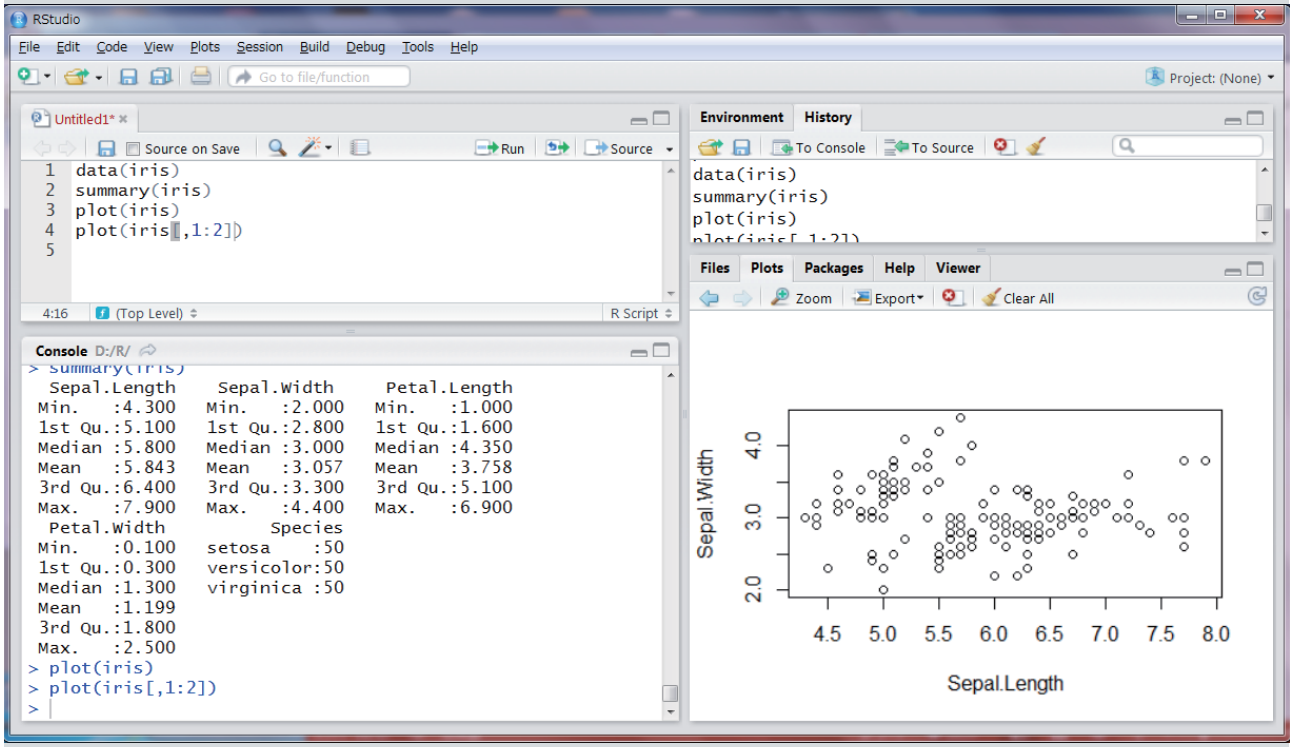
RSTUDIOを使う

- Rstudioの長所
  - 作業の記録(プログラムの編集)
  - グラフ
    - 複数参照できる
    - インタラクティブ
  - ドキュメンテーション機能が充実

RSTUDIOの起動



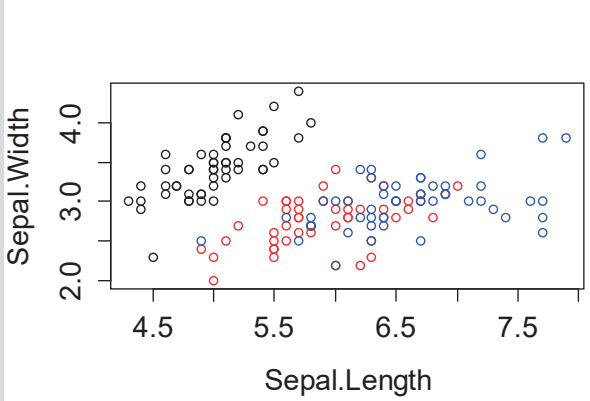
RSTUDIOでの作業

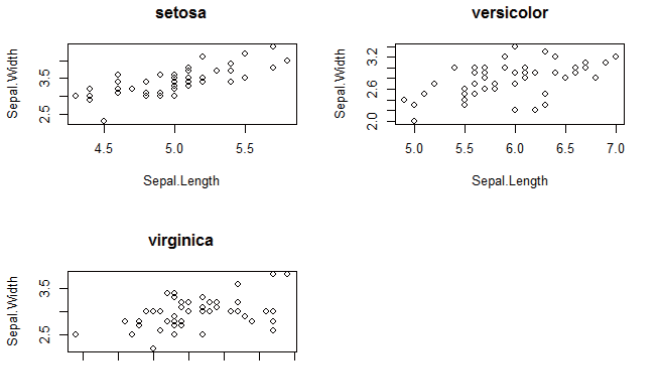


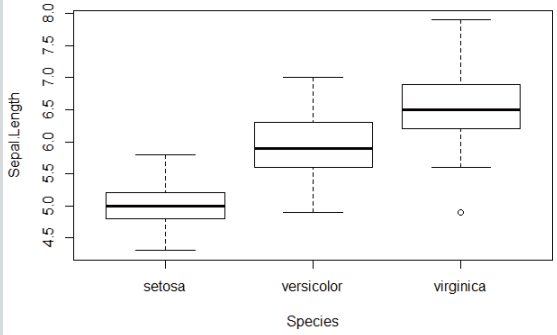
Rでのグラフ描画

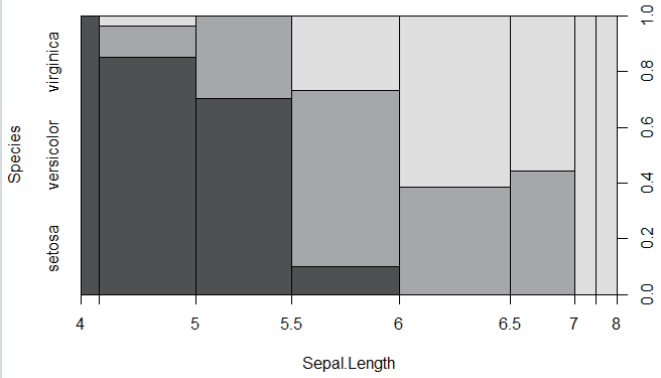
```
plot(iris[,1:2],type="n")
points(iris[iris$Species=="setosa",1:2])
points(iris[iris$Species=="versicolor",1:2],col="red")
points(iris[iris$Species=="virginica",1:2],col="blue")
plot(Sepal.Width~Sepal.Length,data=iris)
plot(Sepal.Width~Sepal.Length,data=iris,col=Species)
legend("topright",legend=levels(iris$Species),pch=1,col=1:3)
plot(Sepal.Width~Sepal.Length,data=iris,col=Species)
legend(x=6.8,y=4.5,legend=levels(iris$Species),pch=1,col=1:3)
par(mfrow=c(2,2))
plot(iris[iris$Species=="setosa",1:2],main="setosa")
plot(iris[iris$Species=="versicolor",1:2],main="versicolor")
plot(iris[iris$Species=="virginica",1:2],main="virginica")
par(mfrow=c(1,1))
plot(Sepal.Length~Species,data=iris)
```

Rでのグラフ描画





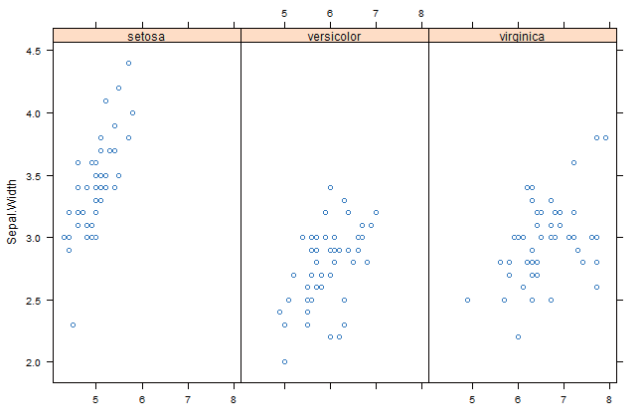


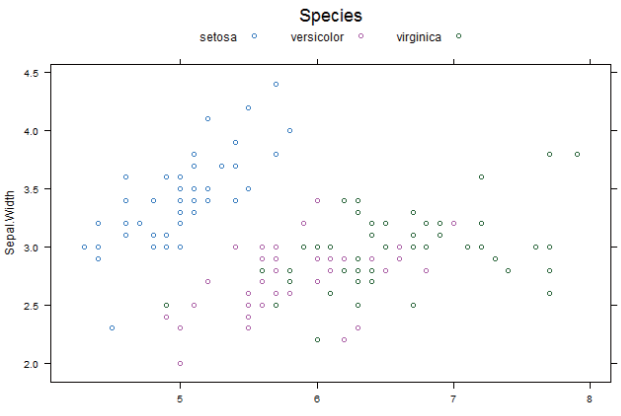


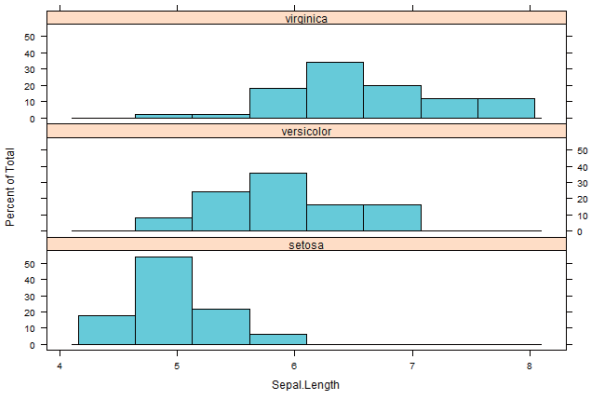
LATTICEグラフ

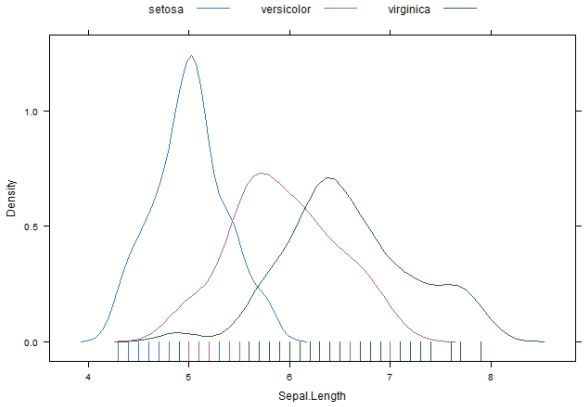
```
xyplot(Sepal.Width~Sepal.Length,data=iris)
xyplot(Sepal.Width~Sepal.Length|Species,data=iris)
xyplot(Sepal.Width~Sepal.Length,group=Species,data=iris)
xyplot(Sepal.Width~Sepal.Length,group=Species,data=iris,auto.key=list(title="Species",column
=3))
xyp.iris2 <- xyplot(Sepal.Width~Sepal.Length,group=Species,data=iris)
update(xyp.iris2,auto.key=list(title="Species",column=3))
histogram(~Sepal.Length,data=iris)
histogram(~Sepal.Length|Species,data=iris)
histogram(~Sepal.Length|Species,data=iris,layout=c(1,3))
densityplot(~Sepal.Length,data=iris)
densityplot(~Sepal.Length|Species,data=iris,plot.points="rug",layout=c(1,3))
densityplot(~Sepal.Length,group=Species,data=iris,plot.points="rug",auto.key=list(column=3))
bwplot(Species~Sepal.Length,data=iris)
bwplot(Sepal.Length~Species,data=iris)
stripplot(Sepal.Length~Species,data=iris)
stripplot(Sepal.Length~Species,data=iris,jitter.data=TRUE)
```

LATTICEグラフ





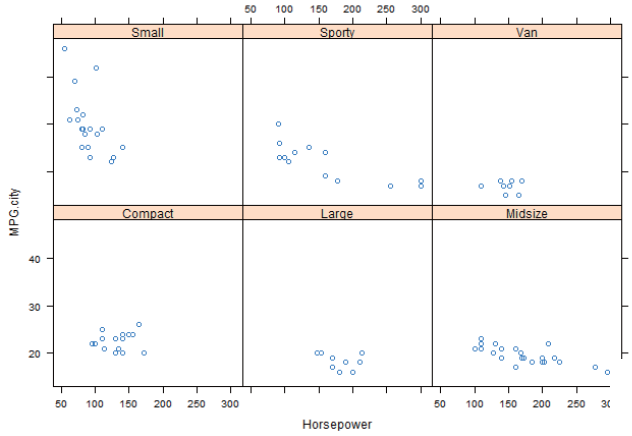


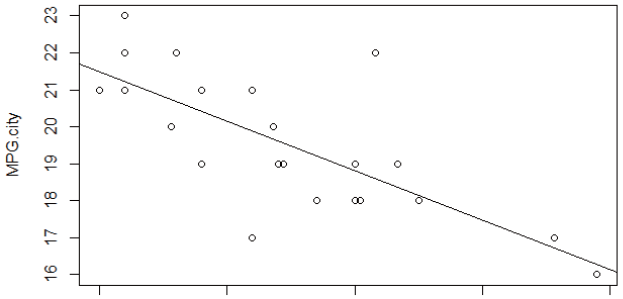


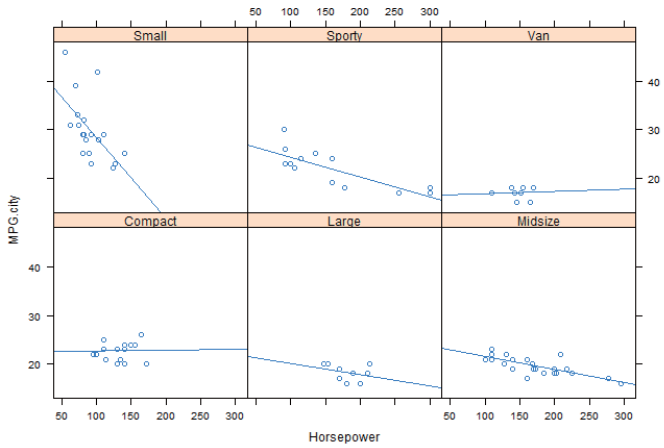
データの抽出利用

```
data(Cars93, package="MASS")
names(Cars93)
xyplot(MPG.city~Horsepower,data=Cars93)
xyplot(MPG.city~Horsepower|Type,data=Cars93)
Cars93[Cars93$Type=="Midsize",c(7,13)]
plot(Cars93[Cars93$Type=="Midsize",c(7,13)])
subset(Cars93,Type=="Midsize",c(Horsepower,MPG.city))
with(subset(Cars93,Type=="Midsize",c(Horsepower,MPG.city)),
     plot(MPG.city~Horsepower))
with(subset(Cars93,Type=="Midsize",c(Horsepower,MPG.city)),
     {plot(MPG.city~Horsepower)
      lm.mid<-lm(MPG.city~Horsepower)
      summary(lm.mid)
      })
Cars93mid<-subset(Cars93,Type=="Midsize",c(Horsepower,MPG.city))
plot(MPG.city~Horsepower,data=Cars93mid)
lm.mid<-lm(MPG.city~Horsepower,data=Cars93mid)
summary(lm.mid)
```

データの抽出利用



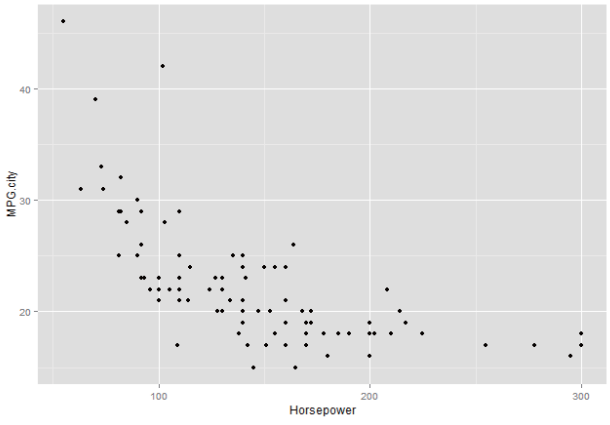


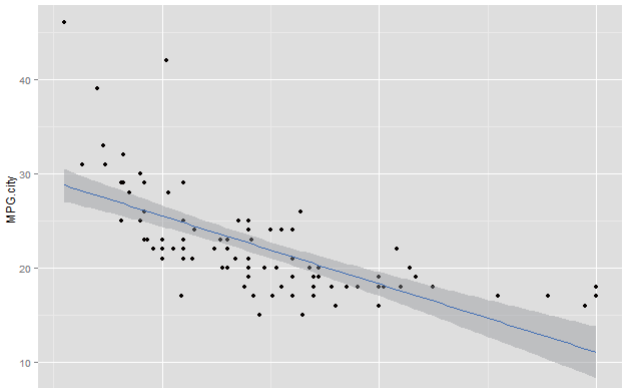


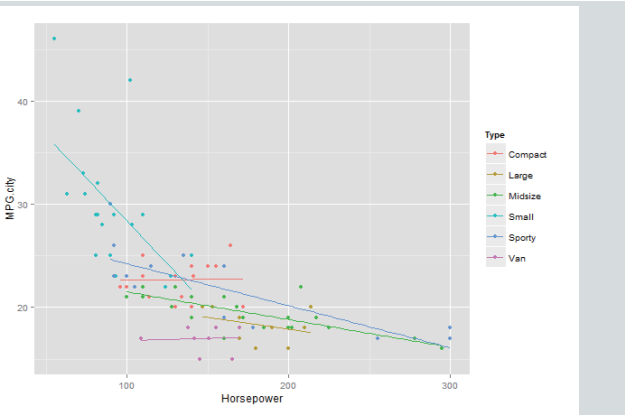
GGPLOT2グラフ

```
library(ggplot2)
ggplot(Cars93, aes(x=Horsepower,y=MPG.city))+geom_point()
gp<-ggplot(Cars93, aes(x=Horsepower,y=MPG.city))
gp+geom_point()
gp+geom_point()+stat_smooth(method=lm)
gp+geom_point()+stat_smooth(method=lm,se=FALSE)
ggplot(Cars93, aes(x=Horsepower,y=MPG.city,colour=Type))+geom_point()
gpType<-ggplot(Cars93, aes(x=Horsepower,y=MPG.city,colour=Type))
gpType+geom_point()+stat_smooth(method=lm,se=FALSE)
gpType2<-ggplot(Cars93, aes(x=Horsepower,y=MPG.city))
gpType2+geom_point()+stat_smooth(method=lm,se=FALSE)+facet_grid(Type~.)
gpType2+geom_point()+stat_smooth(method=lm,se=FALSE)+facet_wrap(~Type)
gpType2+geom_point()+stat_smooth(method=lm,se=FALSE)+facet_wrap(~Type,ncol=2)
ggplot(Cars93,aes(x=Horsepower,colour=Type,fill=Type))+geom_histogram()
ggplot(Cars93,aes(x=Horsepower,colour=Type,fill=Type))+geom_histogram(position="identity",alpha=0.4)
ggplot(Cars93,aes(x=Horsepower,colour=Type,fill=Type))+geom_density(alpha=0.4)
ggplot(Cars93,aes(x=Horsepower))+geom_histogram()+facet_grid(Type~.)
ggplot(Cars93,aes(x=Horsepower))+geom_density()+facet_grid(Type~.)
```

GGPLOT2グラフ







DPLYRによるデータの抽出

```
head(select(Cars93,Horsepower,MPG.city,Type))
head(filter(Cars93,Type=="Midsize"))
Cars93 %>%
 select(Horsepower,MPG.city,Type) %>%
 filter(Type=="Midsize") %>%
 ggplot(aes(x=Horsepower,y=MPG.city))+geom_point()+stat_smooth(method=lm)
Cars93mid <- Cars93 %>%
  select(Horsepower,MPG.city,Type) %>%
  filter(Type=="Midsize")
ggplot(Cars93mid,aes(x=Horsepower,y=MPG.city))+geom_point()+stat_smooth(method=lm)
lm.Mid<-lm(MPG.city~Horsepower,data=Cars93mid)
summary(lm.Mid)
head(mutate(Cars93mid,var1=MPG.city/Horsepower))
head(arrange(Cars93mid,Horsepower))  #desc(Horsepower)
summarize(Cars93mid,n=n(),m.Hp=mean(Horsepower),s.Hp=sd(Horsepower))
grouped.Cars93 <- group_by(Cars93,Type)
summarize(grouped.Cars93,n=n(),m.Hp=mean(Horsepower),s.Hp=sd(Horsepower))
```

実データの解析例

- プロ野球の打者の成績
  - Excelデータ
    - フィルタ、新たな変数の作成
    - Rでのコマンドによる作業
  - 毎年のデータの読み込みと結合
  - Rstudioの機能
    - manipulateパッケージ
    - R ドキュメンテーション
    - Shiny