The Data Science Process

Tony Jiang

January 12, 2017

Contents

This is study notes for "Practical Data Science with R" by Nina Zumel and John Mount, Manning 2014. - The book: "Practical Data Science with R" by Nina Zumel and John Mount, Manning 2014 (book copyright Manning Publications Co., all rights reserved) - The support site: GitHub WinVector/zmPDSwR

The data science process

The roles in a data science project

Project roles

A few recurring roles in a data science project

Role	Responsibilites
Project sponsor	Represents the business interest; champions the project
Client	Represents end users' interests; domain expert

Stages of a data science project

- Define the goal
- Collect and manage data
- Build the model
- Evaluate and critique model

- Present results and document
- Deploy the model to solve the problem in the real world
- Define the goal ### Defining the goal ### Data collection and management ### Modeling

Model evaluation and critique

Presentation and documentation

Model deployment and maintenance

Setting expectations

Determining lower and upper boudns on model performance

- The null model: A lowe rbound on performance
- THe Bayes Rate: An upper bound on model performance ## Summary The data science process involves a lot of back-and-forth between the data scientist and other project stakeholders, and between the different stages of the process.

Loading data into R

Working with data from files

Workign with well-structured data from files or URLs

```r
uciCar <- read.csv("data/car.data.csv",sep=",",header= TRUE)
head(uciCar) #display first 6 rows
```

```
##   buying maint doors persons lug_boot safety rating
## 1  vhigh vhigh     2       2    small    low  unacc
## 2  vhigh vhigh     2       2    small    med  unacc
## 3  vhigh vhigh     2       2    small   high  unacc
## 4  vhigh vhigh     2       2      med    low  unacc
## 5  vhigh vhigh     2       2      med    med  unacc
## 6  vhigh vhigh     2       2      med   high  unacc
```

```r
class(uciCar)
```

```
## [1] "data.frame"
```

```r
summary(uciCar)
```

```
##    buying      maint       doors     persons     lug_boot    safety
##  high :432   high :432   2    :432   2   :576   big  :576   high:576
##  low  :432   low  :432   3    :432   4   :576   med  :576   low :576
##  med  :432   med  :432   4    :432   more:576   small:576   med :576
##  vhigh:432   vhigh:432   5more:432
##    rating
```

```
##   acc  : 384
##   good :  69
##   unacc:1210
##   vgood:  65
```

```
dim(uciCar)
```

```
## [1] 1728    7
```

Using R on less-structured data

- Transforming data in R you need "schema documentation" or "data dictionary" to decrypt troublesome data.

Read a data fro mGerman bank credit dataset

```
d <- read.table(paste('http://archive.ics.uci.edu/ml/',
   'machine-learning-databases/statlog/german/german.data',sep=''),
   stringsAsFactors=F,header=F)
print(d[1:3,])
```

```
##    V1 V2  V3  V4   V5  V6  V7 V8  V9  V10 V11  V12 V13  V14  V15 V16  V17
## 1 A11  6 A34 A43 1169 A65 A75  4 A93 A101   4 A121  67 A143 A152   2 A173
## 2 A12 48 A32 A43 5951 A61 A73  2 A92 A101   2 A121  22 A143 A152   1 A173
## 3 A14 12 A34 A46 2096 A61 A74  2 A93 A101   3 A121  49 A143 A152   1 A172
##   V18  V19  V20 V21
## 1   1 A192 A201   1
## 2   1 A191 A201   2
## 3   2 A191 A201   1
```

Change the column names to something meaningful

```
colnames(d) <- c('Status.of.existing.checking.account',
   'Duration.in.month',  'Credit.history', 'Purpose',
   'Credit.amount', 'Savings account/bonds',
   'Present.employment.since',
   'Installment.rate.in.percentage.of.disposable.income',
   'Personal.status.and.sex', 'Other.debtors/guarantors',
   'Present.residence.since', 'Property', 'Age.in.years',
   'Other.installment.plans', 'Housing',
   'Number.of.existing.credits.at.this.bank', 'Job',
   'Number.of.people.being.liable.to.provide.maintenance.for',
   'Telephone', 'foreign.worker', 'Good.Loan')
d$Good.Loan <- as.factor(ifelse(d$Good.Loan==1,'GoodLoan','BadLoan'))
print(d[1:3,])
```

```
##   Status.of.existing.checking.account Duration.in.month Credit.history
## 1                                 A11                 6            A34
## 2                                 A12                48            A32
## 3                                 A14                12            A34
##   Purpose Credit.amount Savings account/bonds Present.employment.since
```

```
## 1     A43          1169                     A65                        A75
## 2     A43          5951                     A61                        A73
## 3     A46          2096                     A61                        A74
##   Installment.rate.in.percentage.of.disposable.income
## 1                                                   4
## 2                                                   2
## 3                                                   2
##   Personal.status.and.sex Other.debtors/guarantors Present.residence.since
## 1                     A93                     A101                       4
## 2                     A92                     A101                       2
## 3                     A93                     A101                       3
##   Property Age.in.years Other.installment.plans Housing
## 1     A121           67                    A143    A152
## 2     A121           22                    A143    A152
## 3     A121           49                    A143    A152
##   Number.of.existing.credits.at.this.bank  Job
## 1                                       2 A173
## 2                                       1 A173
## 3                                       1 A172
##   Number.of.people.being.liable.to.provide.maintenance.for Telephone
## 1                                                        1      A192
## 2                                                        1      A191
## 3                                                        2      A191
##   foreign.worker Good.Loan
## 1           A201  GoodLoan
## 2           A201   BadLoan
## 3           A201  GoodLoan
```

Building a map to interprest loan use codes

```
mapping <- list(
   'A40'='car (new)',
   'A41'='car (used)',
   'A42'='furniture/equipment',
   'A43'='radio/television',
   'A44'='domestic appliances' # note that other codes are not defiend here.
   )
```

Transform the data

```
for(i in 1:(dim(d))[2]) {               # Note: 1
   if(class(d[,i])=='character') {
      d[,i] <- as.factor(as.character(mapping[d[,i]]))      # Note: 2
   }
}
table(d$Purpose,d$Good.Loan)
```

```
##
##                        BadLoan GoodLoan
##   car (new)                 89      145
##   car (used)                17       86
##   domestic appliances        4        8
##   furniture/equipment       58      123
```

```
##   NULL                      70      120
##   radio/television          62      218
```

Working with relational databases

The right way to work with data found in databases is to connect R directly to the database.

A production-size example

- United States Census 2011 national PLUMS American Communicty Survey data (www.census.gov/acs/www/data_docum
- Millions of rows
- a few gigabytes when zipped
- This size is the sweet spot for relational datbase or SQL databse. we are not forced to move into a MapReduce or database cluster to do our work.

Curating the data Staging the data into a database - H2 - SQL Screwdriver SQuirrel SQL

Exploring Data

Resist the temptation to dive into the modelig step without looking at the dataset first.

Using summary statistics to spot problems

```r
custdata<- read.csv('data/custdata.tsv',header=T,sep='\t')

summary(custdata)
```

```
##      custid          sex      is.employed         income
##  Min.   :   2068   F:440   Mode :logical   Min.   : -8700
##  1st Qu.: 345667   M:560   FALSE:73        1st Qu.: 14600
##  Median : 693403           TRUE :599       Median : 35000
##  Mean   : 698500           NA's :328       Mean   : 53505
##  3rd Qu.:1044606                           3rd Qu.: 67000
##  Max.   :1414286                           Max.   :615000
##
##              marital.stat health.ins
##  Divorced/Separated:155   Mode :logical
##  Married           :516   FALSE:159
##  Never Married     :233   TRUE :841
##  Widowed           : 96   NA's :0
##
##
##
##                          housing.type recent.move      num.vehicles
##  Homeowner free and clear    :157   Mode :logical   Min.   :0.000
##  Homeowner with mortgage/loan:412   FALSE:820       1st Qu.:1.000
##  Occupied with no rent       : 11   TRUE :124       Median :2.000
##  Rented                      :364   NA's :56        Mean   :1.916
```

```
##  NA's                        : 56                   3rd Qu.:2.000
##                                                     Max.   :6.000
##                                                     NA's   :56
##       age                state.of.res
##  Min.   :  0.0   California  :100
##  1st Qu.: 38.0   New York    : 71
##  Median : 50.0   Pennsylvania: 70
##  Mean   : 51.7   Texas       : 56
##  3rd Qu.: 64.0   Michigan    : 52
##  Max.   :146.7   Ohio        : 51
##                  (Other)     :600
```

Typical problems revealed by data summaries

- Missing values
- Invalid values and outliers
- Data range: relative because of units. (babies age in weeks vs years)
- Units

Spotting problems using graphics and visualization

```
We cannot expect a small number of numerical values to consistntly convey the wealth of information tha
```

The use of graphics to examine data is called visualization.

Visually checking distributions for a single variable

what is the peak value? How many peaks ? How normal is the data? How much does the data vary?

- Histogram

```r
library(ggplot2)
ggplot(custdata)+geom_histogram(aes(x=age),binwidth=5,fill="gray")
```

- Density plots

```r
library(scales)
ggplot(custdata)+geom_density(aes(x=income))+scale_x_continuous(labels=dollar)
```

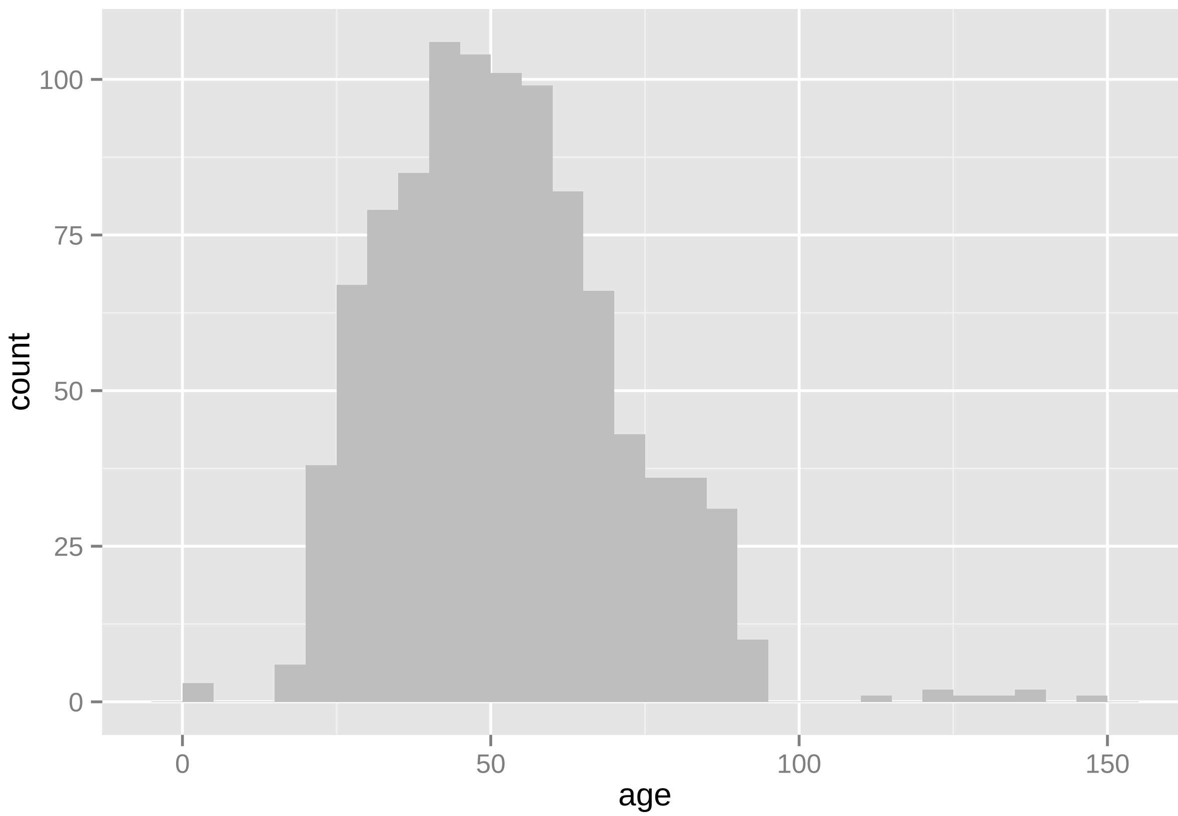
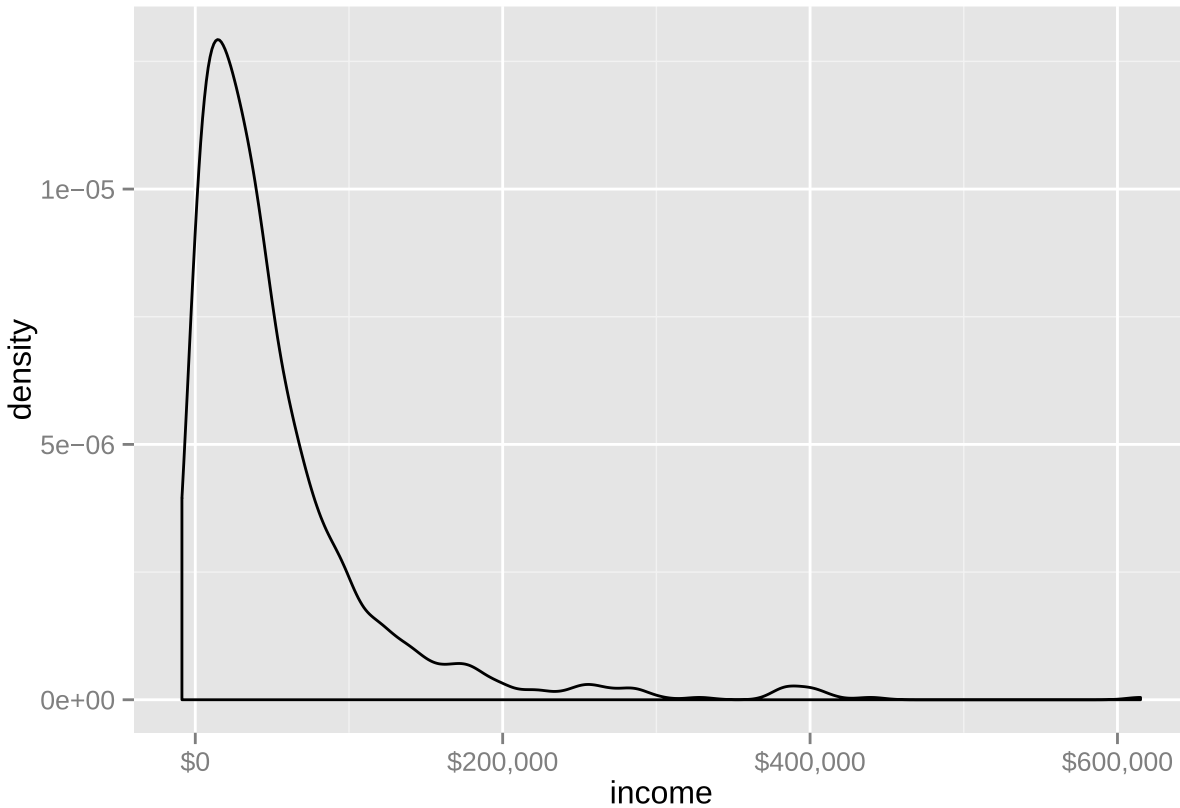


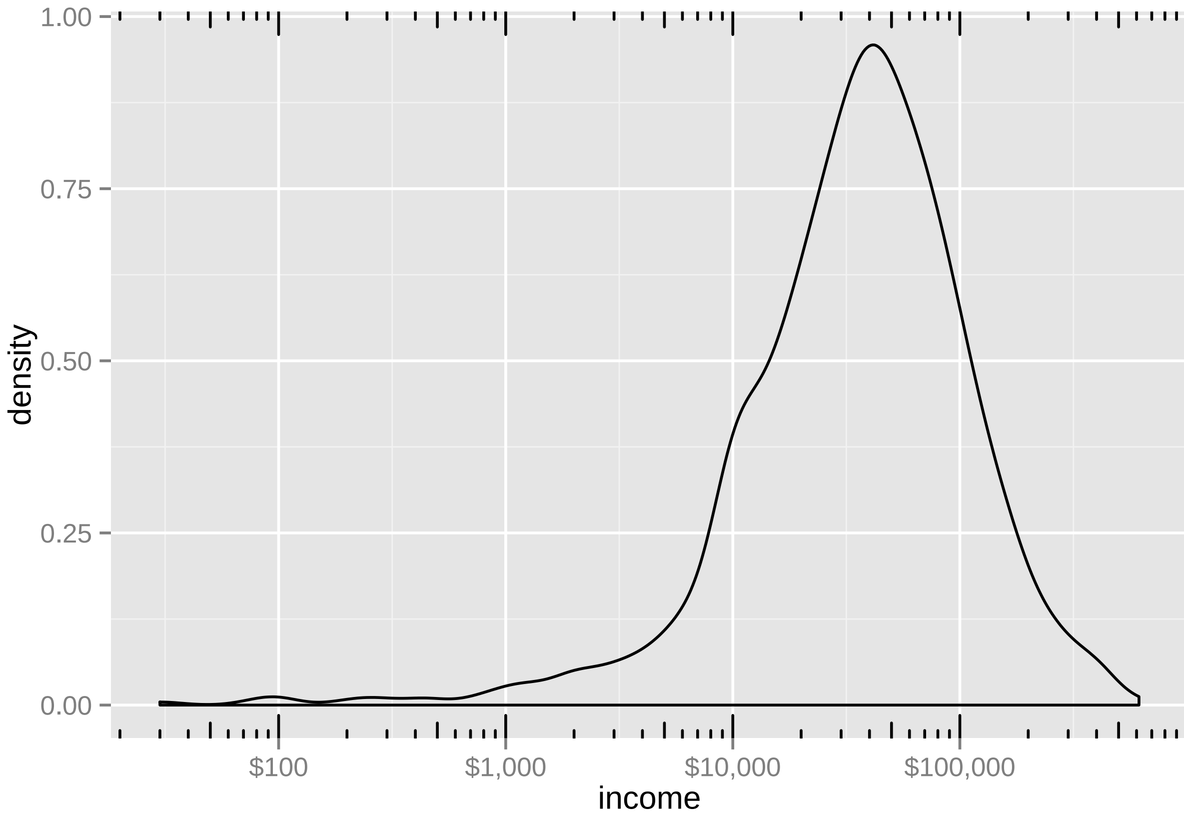
Figure 1: A histogram tells you where your data is concentrated




```
ggplot(custdata)+geom_density(aes(x=income))+scale_x_log10(breaks=c(100,1000,10000,100000),labels=dolla
```

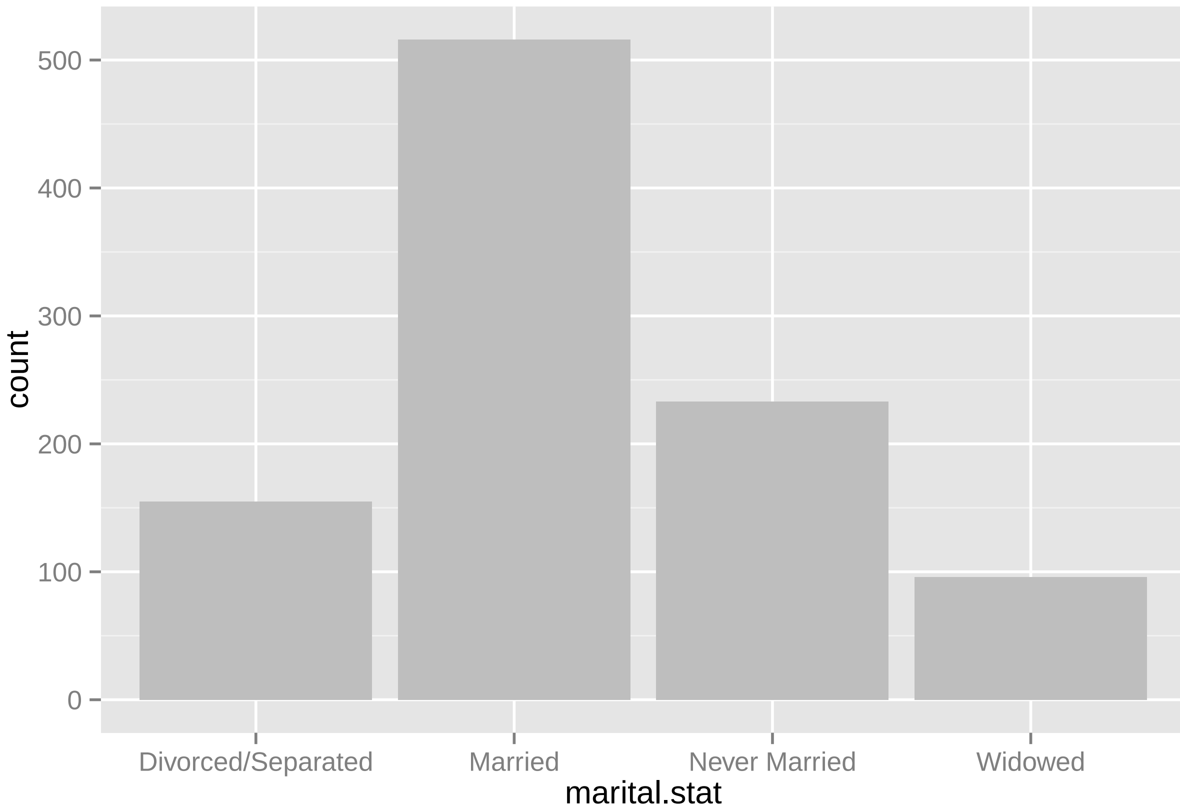
```
## Warning in scale$trans$trans(x): NaNs produced
```

```
## Warning: Removed 79 rows containing non-finite values (stat_density).
```



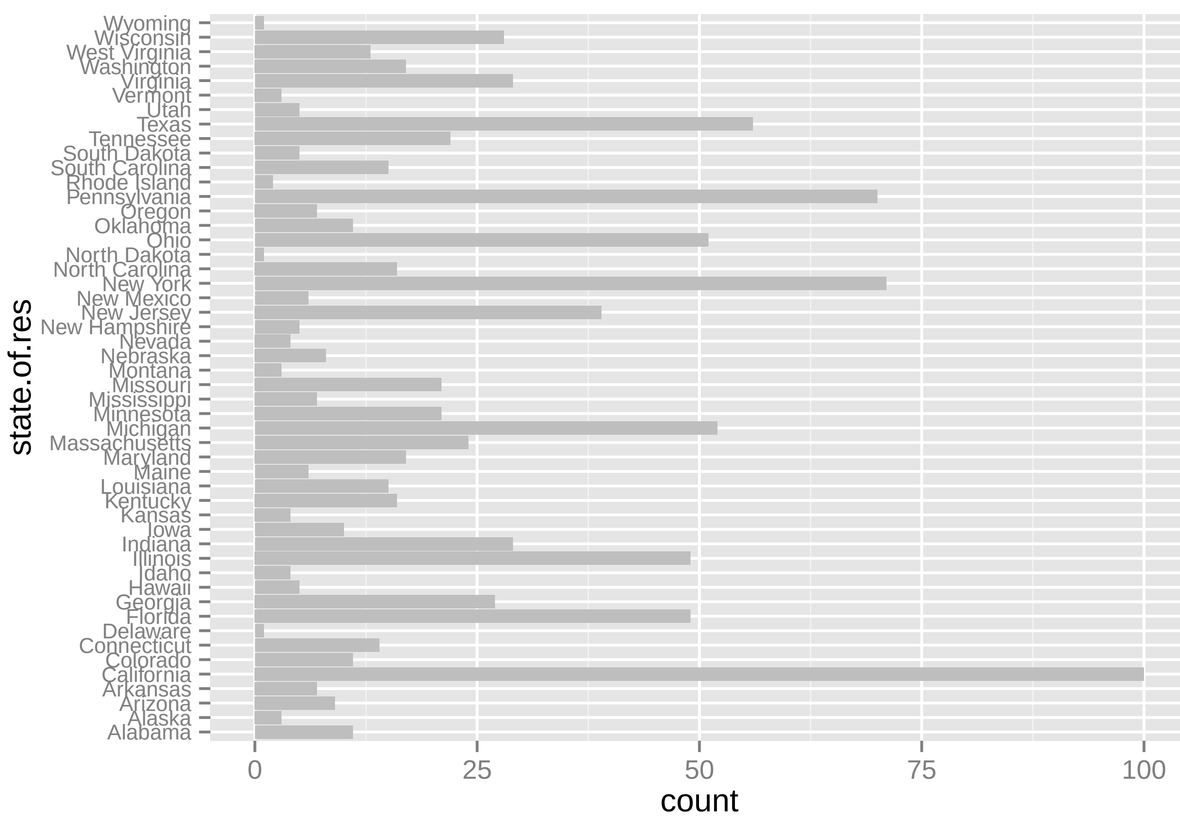
- Bar charts

```
ggplot(custdata)+geom_bar(aes(x=marital.stat),fill="gray")
```



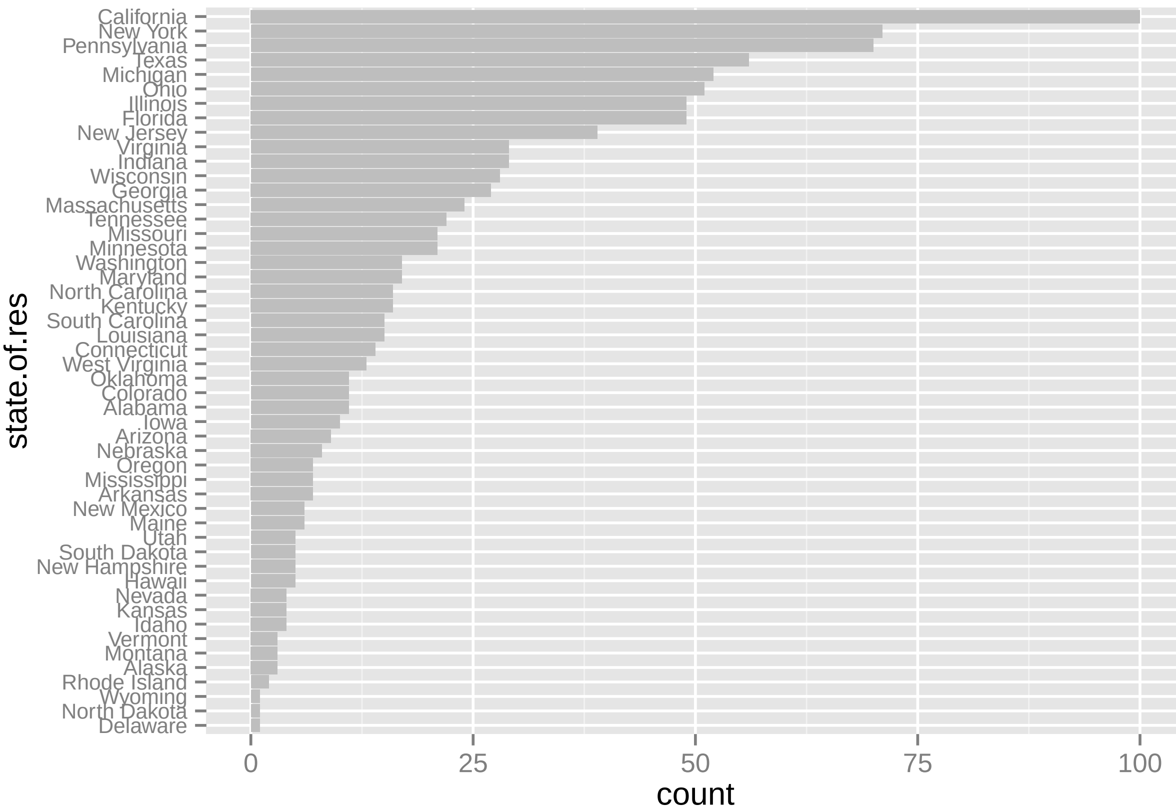
A horizontal bar chart can be easier to read when there are several categories with long names.

```
ggplot(custdata)+geom_bar(aes(x=state.of.res),fill="gray")+coord_flip()+theme(axis.text.y=element_text(
```



Cleveland recommends that the data in a bar chart be sorted, to more efficiently extract insight from the data.

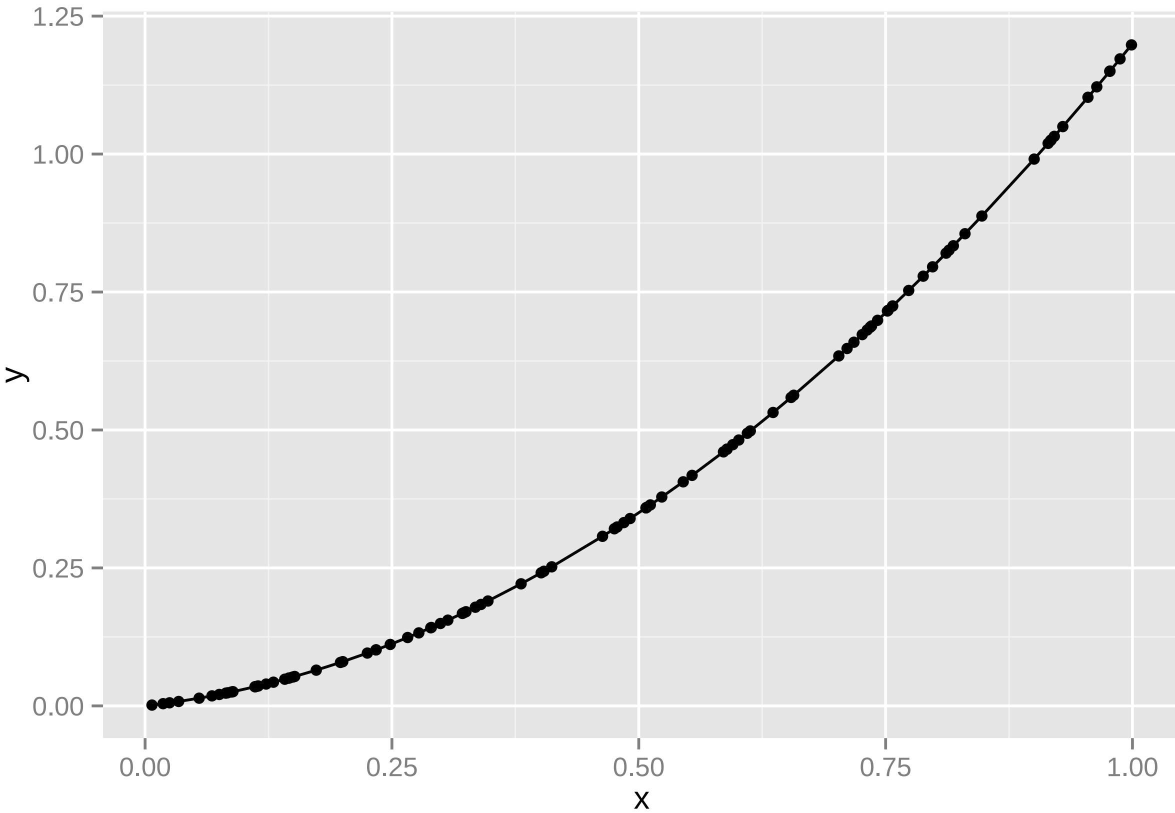
```
statesums<-table(custdata$state.of.res)
statef<-as.data.frame(statesums)
colnames(statef)<-c("state.of.res","count")
statef<-transform(statef,state.of.res=reorder(state.of.res,count))
ggplot(statef)+geom_bar(aes(x=state.of.res,y=count),stat="identity",fill="gray")+coord_flip()+theme(axi
```



Visually checking relationships between two variables

- Line Plots

```
x<-runif(100)
y<-x^2+0.2*x
ggplot(data.frame(x=x,y=y),aes(x=x,y=y))+geom_line()+geom_point()
```



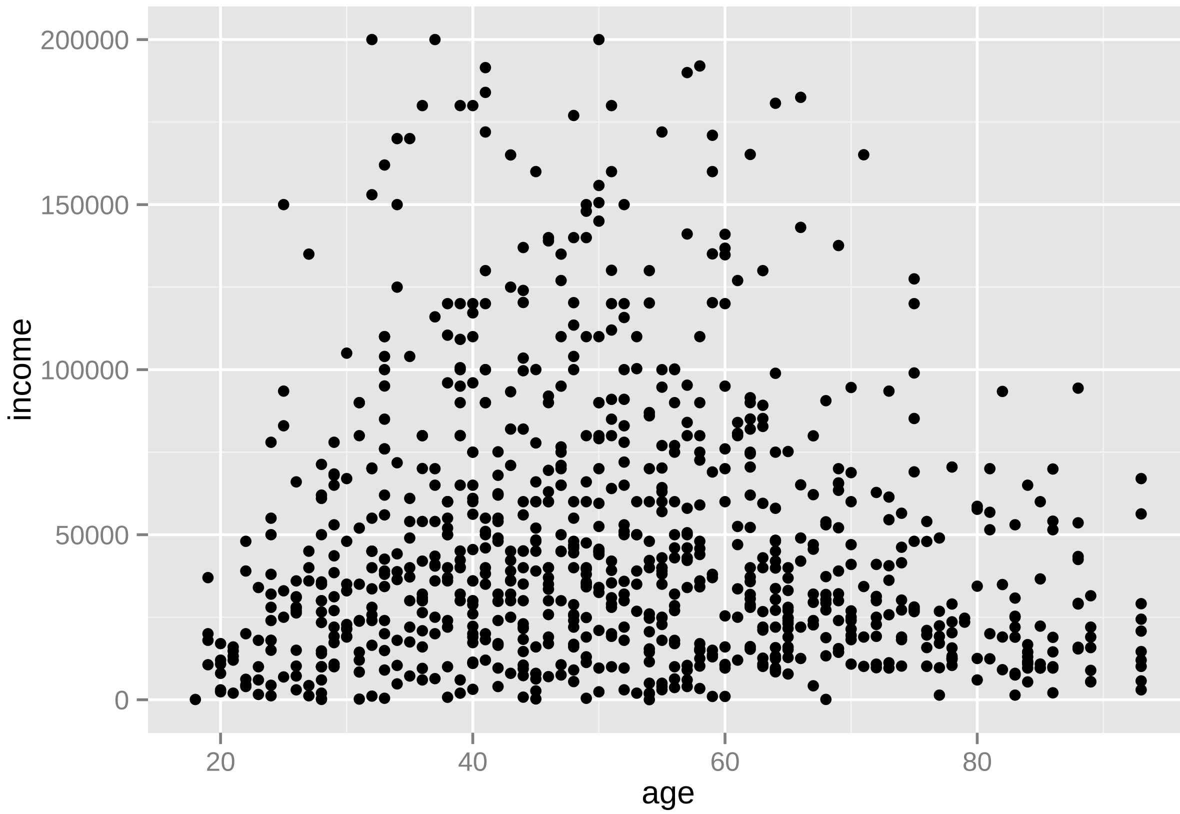
Scatter Plots and smoothing curves

```
custdata2<-subset(custdata,(custdata$age>0 & custdata$age<100 & custdata$income>0))
cor(custdata2$age,custdata2$income)
```

```
## [1] -0.02240845
```

```
ggplot(custdata2,aes(x=age,y=income))+geom_point()+ylim(0,200000)
```

```
## Warning: Removed 32 rows containing missing values (geom_point).
```

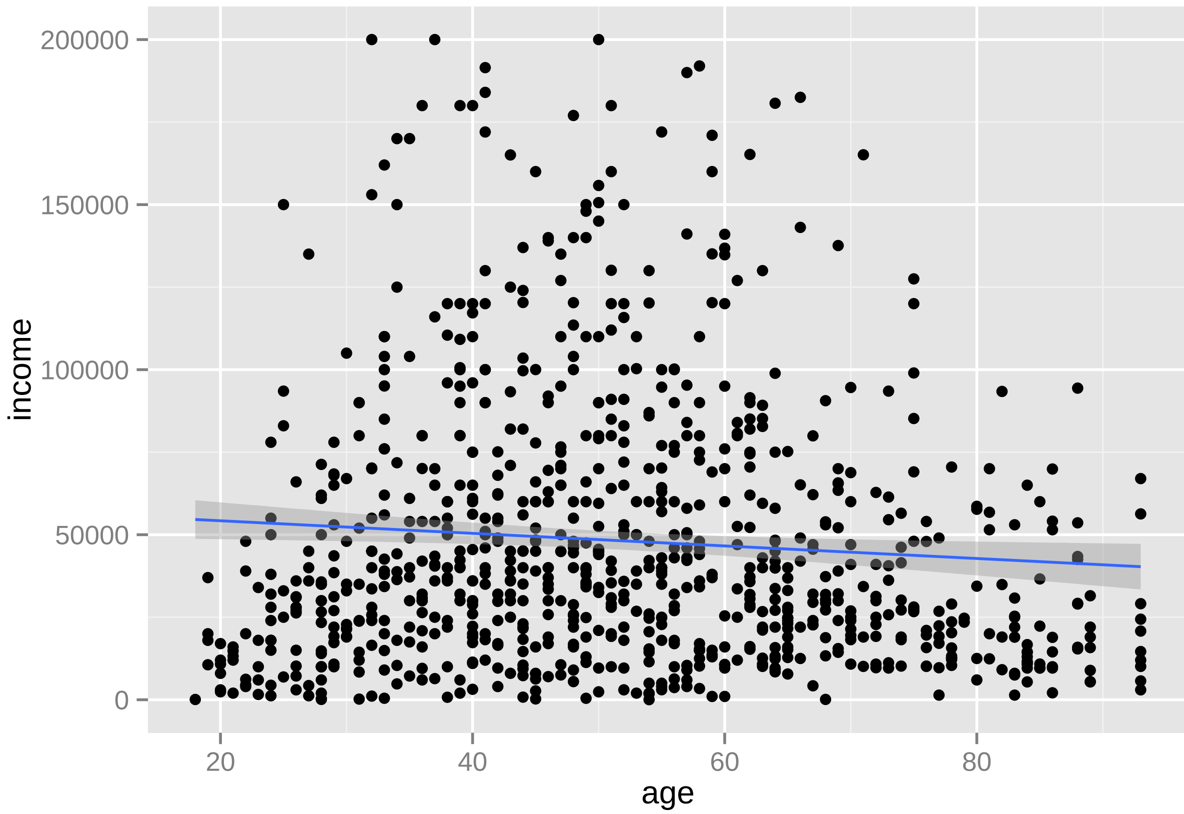


Linear fit

```
ggplot(custdata2,aes(x=age,y=income))+geom_point()+geom_smooth(method="lm")+ylim(0,200000)
```

```
## Warning: Removed 32 rows containing missing values (stat_smooth).
```

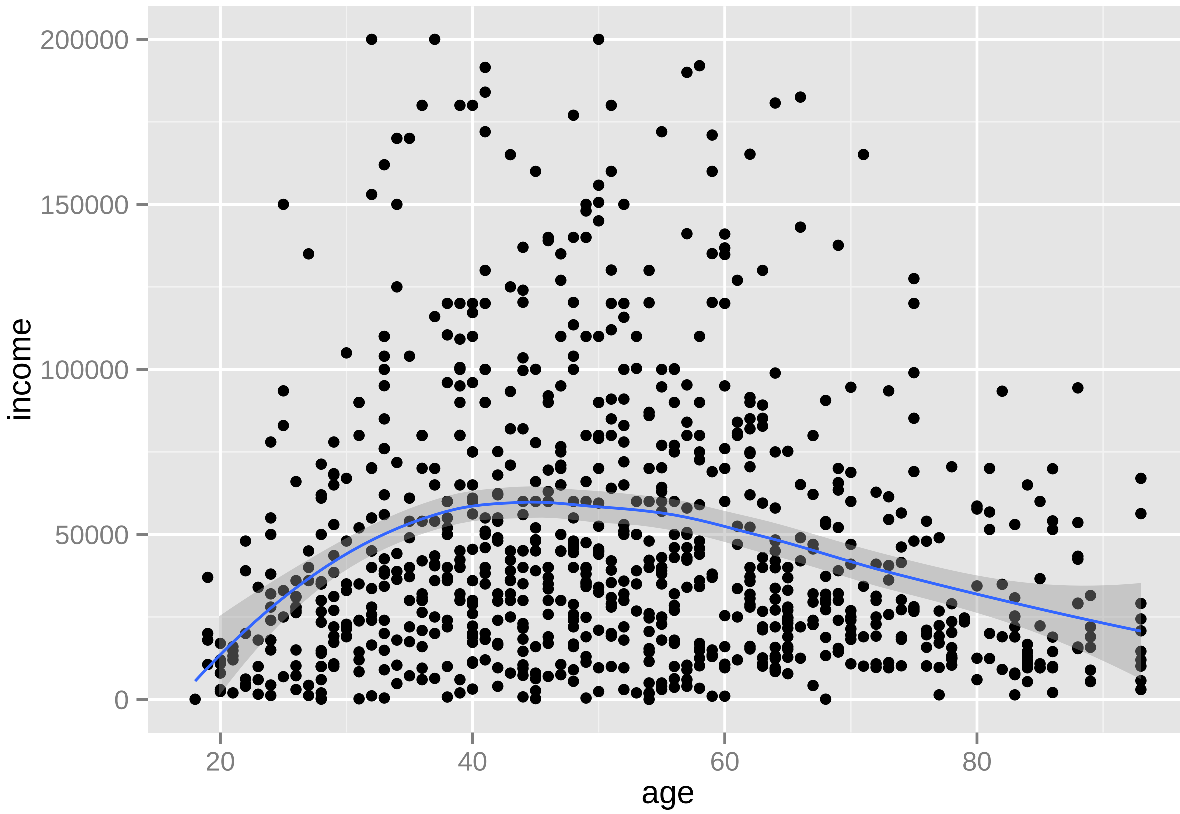
```
## Warning: Removed 32 rows containing missing values (geom_point).
```



Smoothing Curve

```
ggplot(custdata2,aes(x=age,y=income))+geom_point()+geom_smooth()+ylim(0,200000)
```

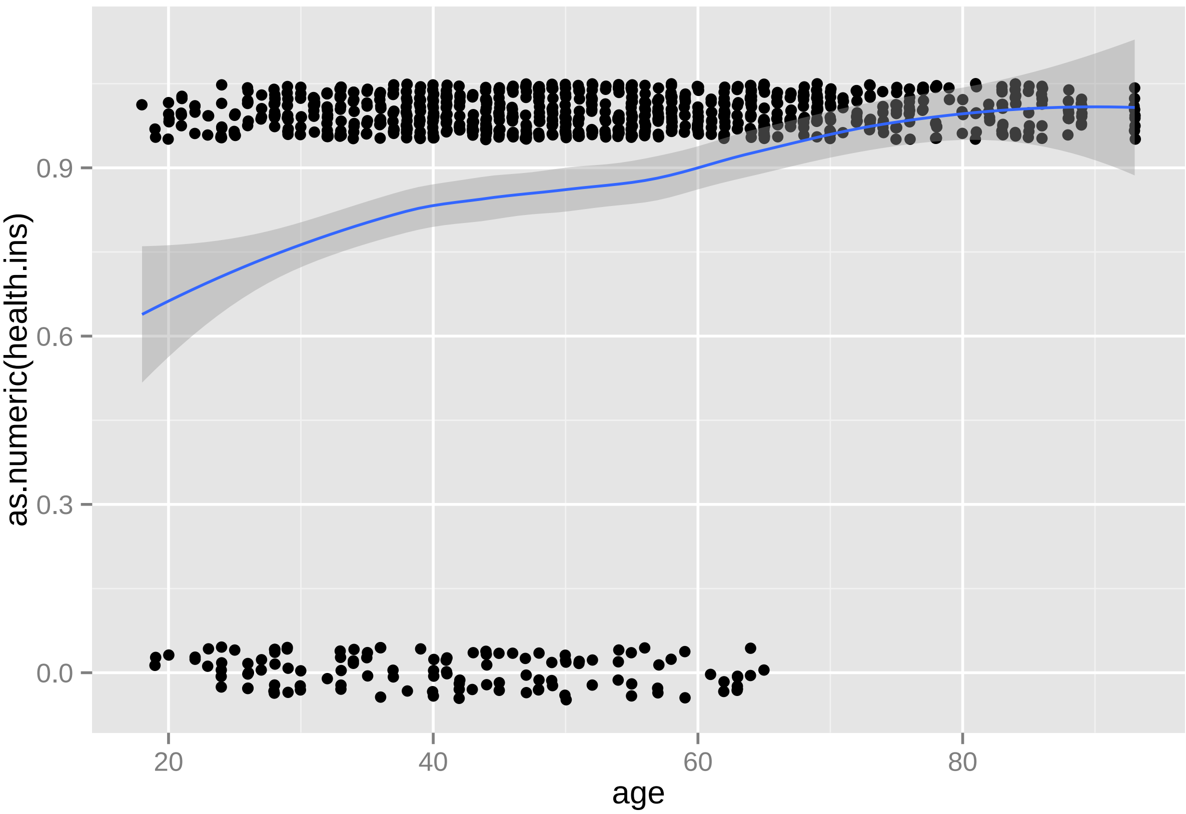
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to cl

## Warning: Removed 32 rows containing missing values (stat_smooth).

## Warning: Removed 32 rows containing missing values (geom_point).
```



Distribution of customers with health insurance, as a function of age

```
ggplot(custdata2,aes(x=age,y=as.numeric(health.ins)))+geom_point(position=position_jitter(w=0.05,h=0.05)
```

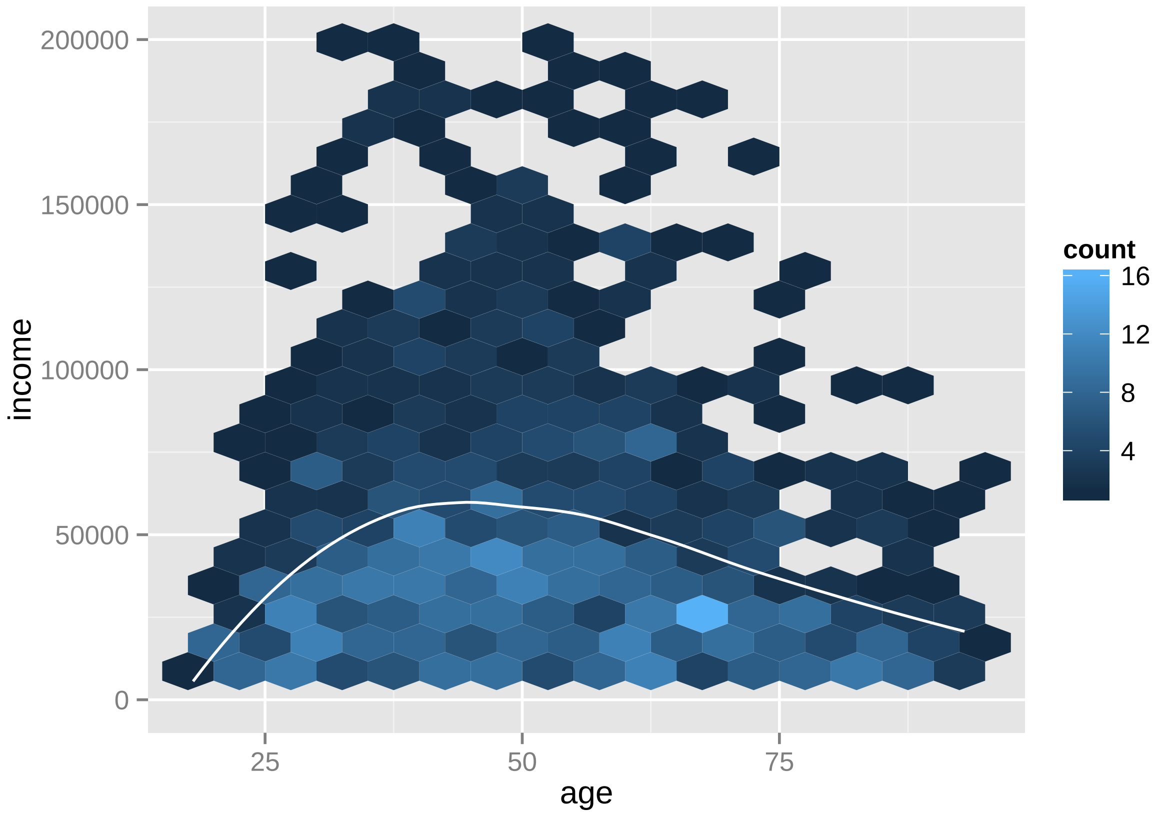
```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to cl
```



- Hexbin Plots

```
library(hexbin)
ggplot(custdata2,aes(x=age,y=income))+geom_hex(binwidth=c(5,10000))+geom_smooth(color="white",se=F)+ylim
```

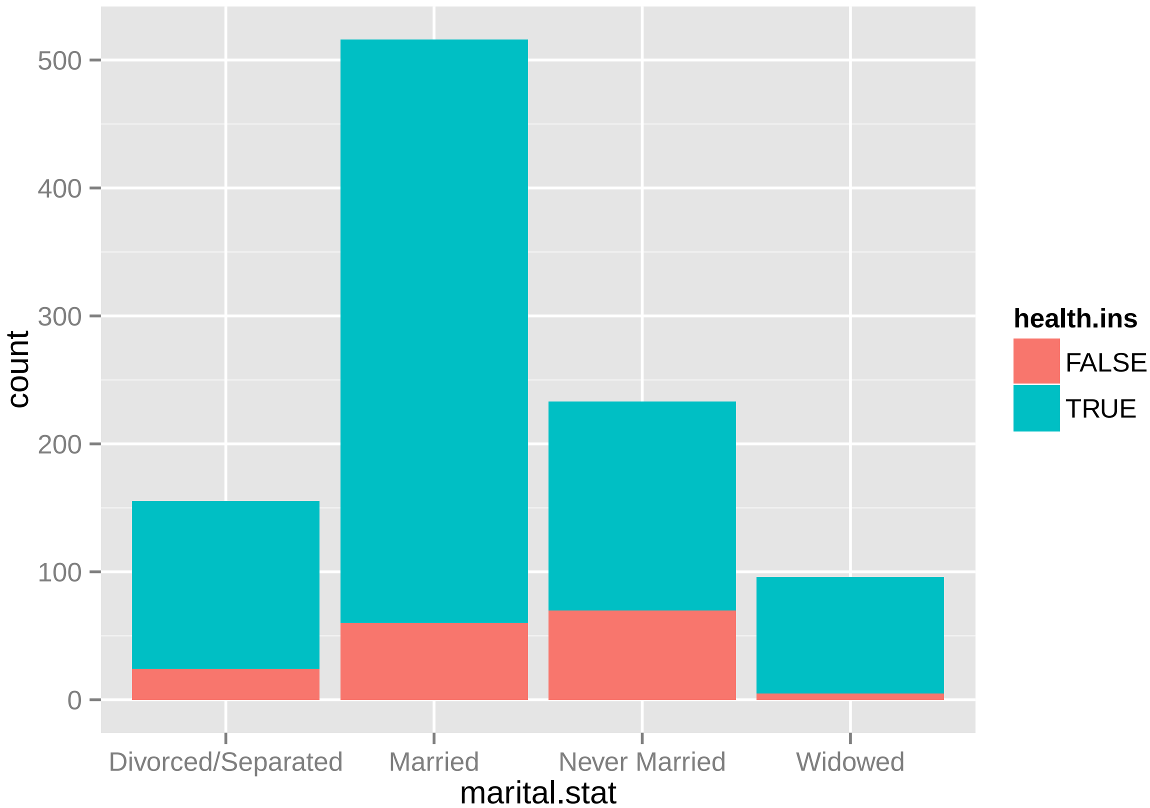
```
## Warning: Removed 32 rows containing missing values (stat_hexbin).

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

## Warning: Removed 32 rows containing missing values (stat_smooth).
```



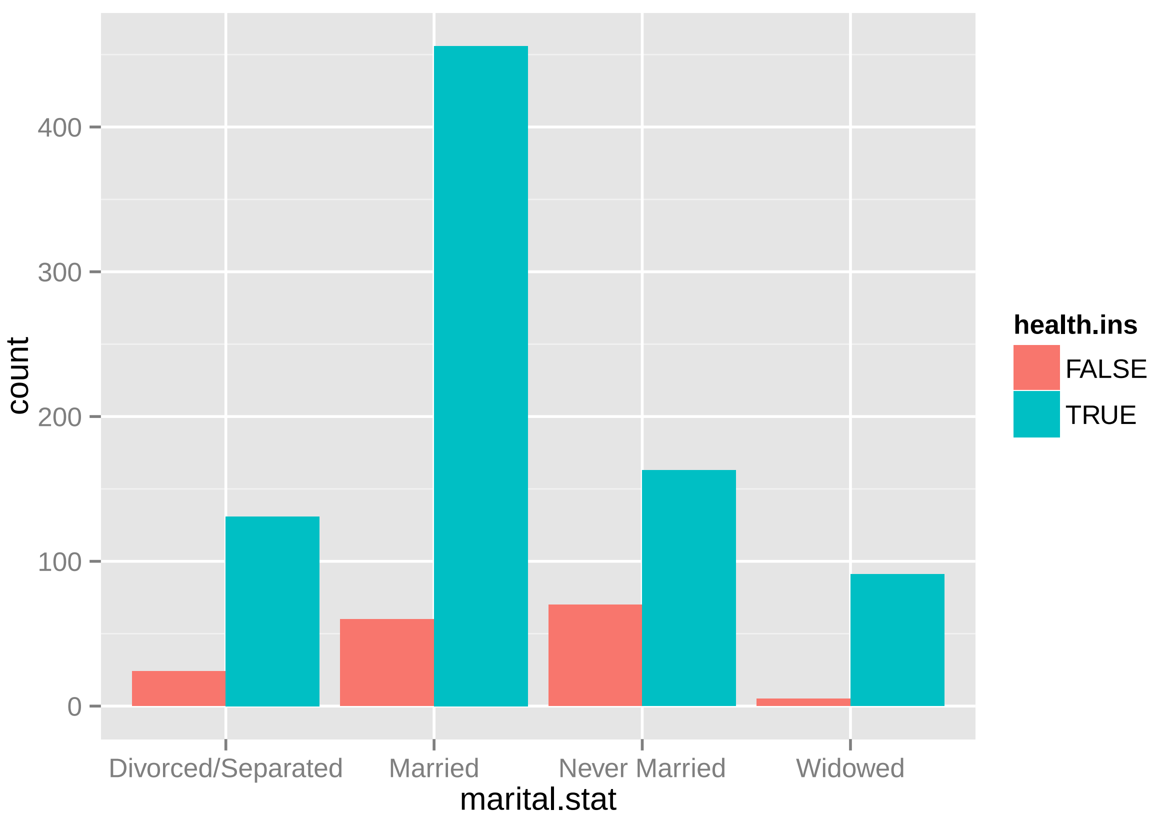
- Bar charts for two categorical variables

Relationship between marital status and the probability of health insurance coverage.

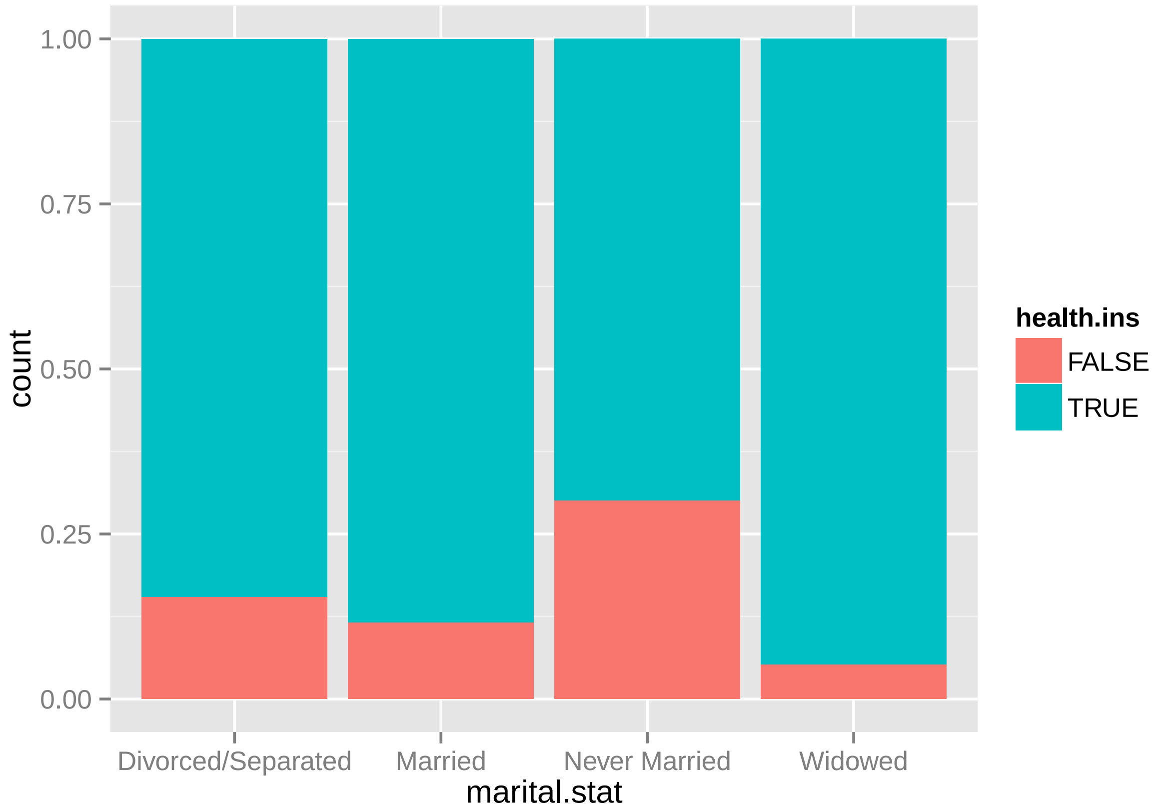
```
# bar chart
ggplot(custdata)+geom_bar(aes(x=marital.stat,fill=health.ins))
```




```
ggplot(custdata)+geom_bar(aes(x=marital.stat,fill=health.ins),position="dodge")
```

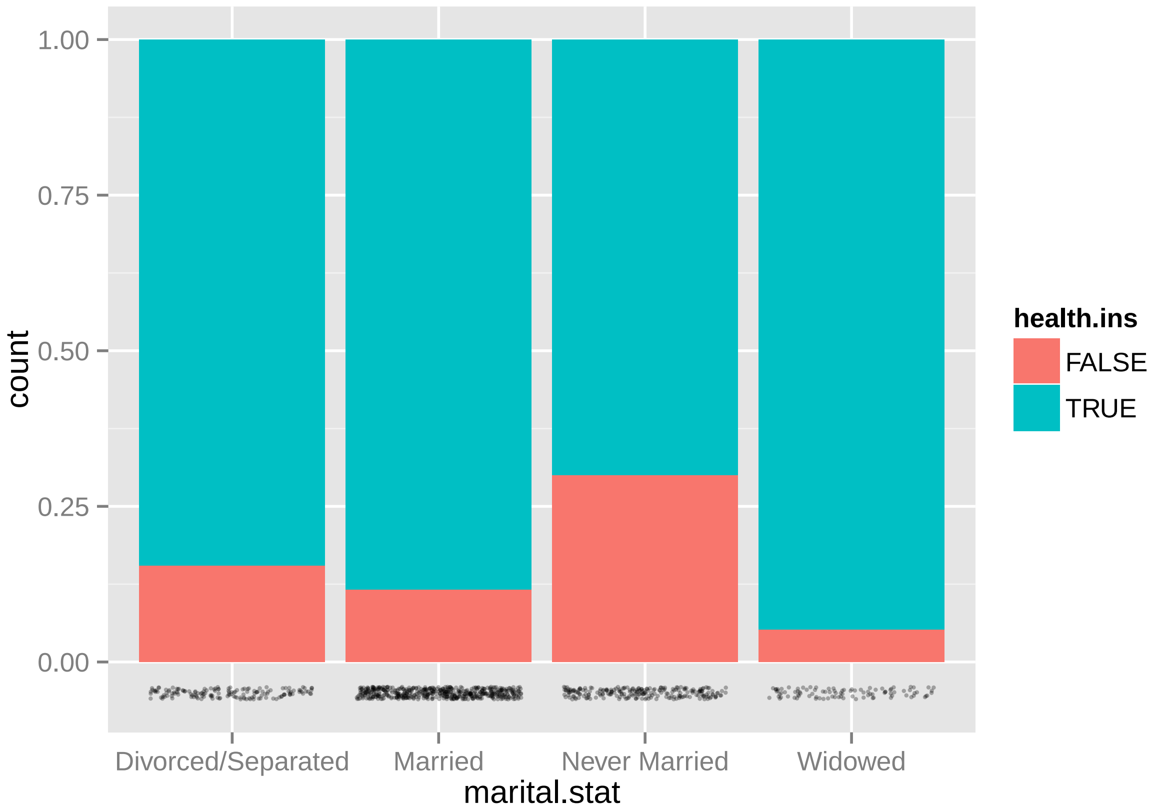


```r
ggplot(custdata)+geom_bar(aes(x=marital.stat,fill=health.ins),position="fill")
```



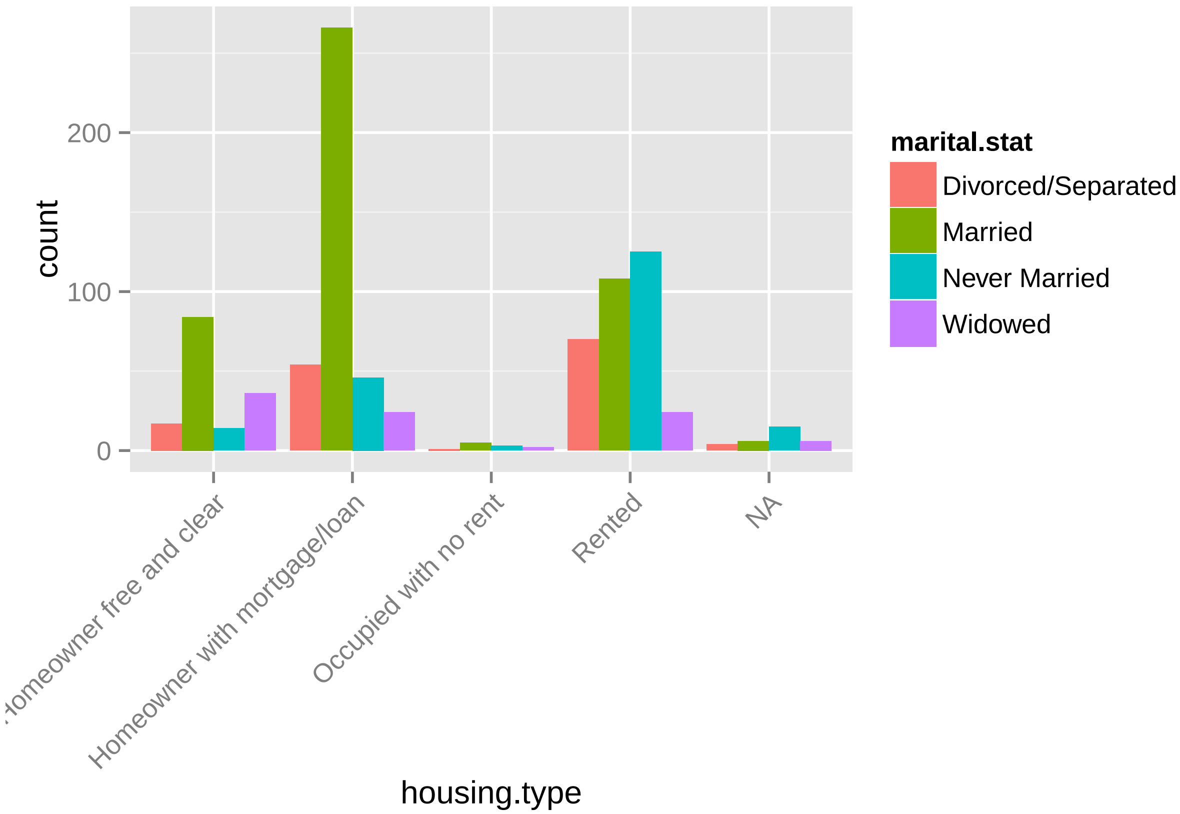
add a rug

```r
ggplot(custdata,aes(x=marital.stat))+geom_bar(aes(fill=health.ins),position="fill")+geom_point(aes(y=-0
```

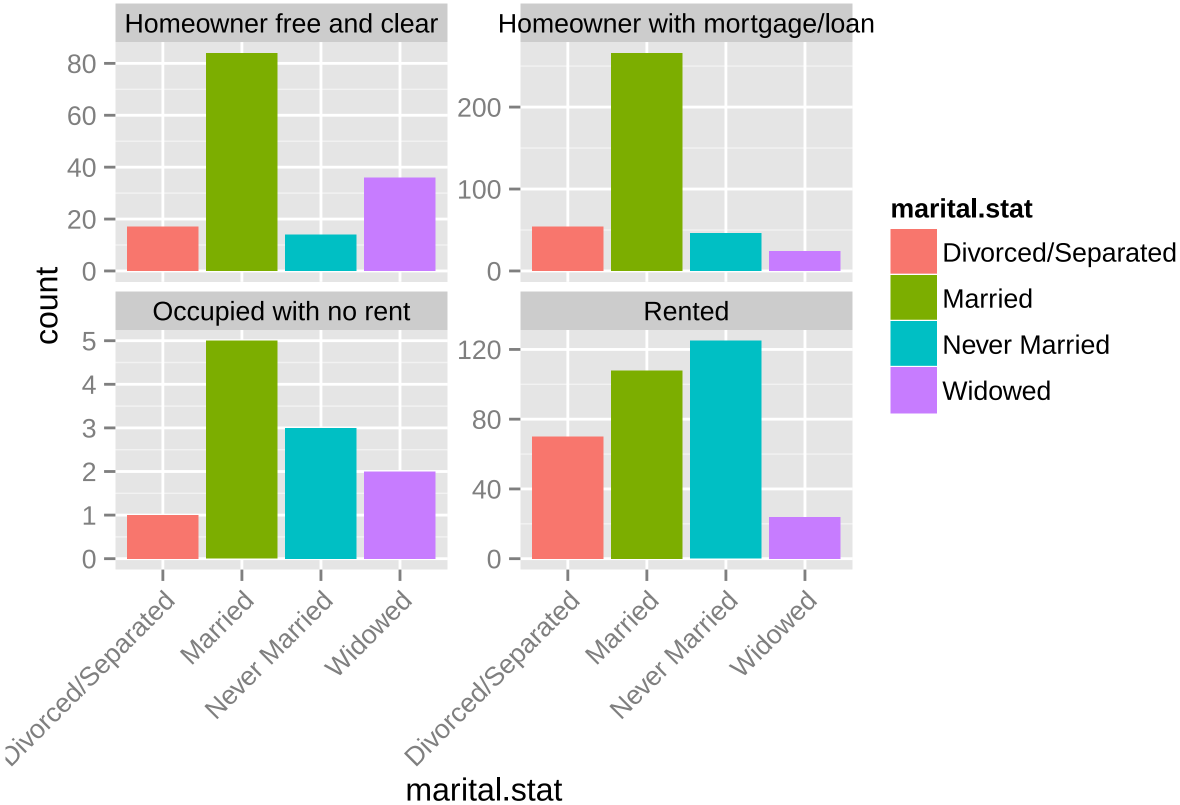


Facetd

```
ggplot(custdata2)+geom_bar(aes(x=housing.type,fill=marital.stat),position="dodge")+theme(axis.text.x=ele
```




```
ggplot(subset(custdata2,housing.type!="NA"))+geom_bar(aes(x=marital.stat,fill=marital.stat),position="de
```



Summary

- Take the time to examine your data before diving into the modeling
- The summary command helps you spot issues with data range, units, data type, and missing or invalid values
- Visualization additionally gives you a sense of data distribution and relationships among variables
- Visualization is an iterative process and helps answer questions about the data. Time spent here is time not wasted during the modeling process.