Generalized Linear Models. Uncertainty estimation

Lecture 2c

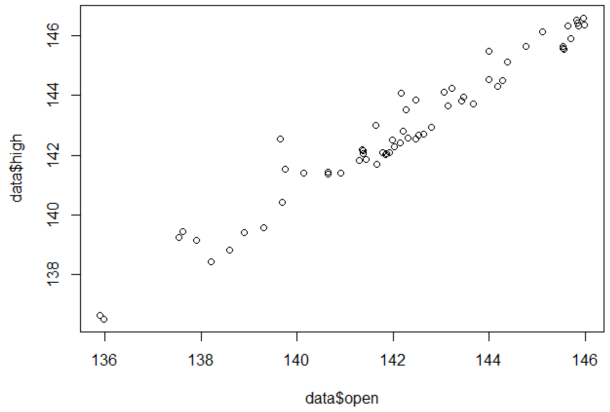
Moving beyond typical distributions

- We know how to model
  - Normally distributed targets -> linear regression
  - Bernoulli and Multinomial targets→logistic regression
  - What if target distribution is more complex?

Example 1: Daily Stock prices NASDAQ

- Open
- High (within day)

Does it seem that the error is normal here?



Example 2: Number of calls to bank

- Y=Number of calls
- X= time

Endless amount of classes → multinomial does not work… (Poisson)

Exponential family

- More advanced error distributions are sometimes needed!
- Many distributions belong to **exponential** family:
  - Normal, Exponential, Gamma, Beta, Chi-squared..
  - Bernoulli, Multinoulli, Poisson...

$$p(\boldsymbol{x}|\boldsymbol{\eta}) = h(\boldsymbol{x})g(\boldsymbol{\eta})e^{(\boldsymbol{\eta}^T u(\boldsymbol{x}))}$$

- Easy to find MLE and MAP
- Non-exponential family distributions: uniform, Student t

Example: Bernoulli

Generalized linear models

- Assume Y from the exponential family
- **Model** is $Y \sim EF(\mu, \dots),\ f(\mu) = \boldsymbol{w}^T \boldsymbol{x}$
  - Alt $\mu = f^{-1}(\boldsymbol{w}^T \boldsymbol{x})$
  - f^{-1} is activation function
  - f is link function (in principle, arbitrary)
- Arbitrary f will lead to (s – dispersion parameter)

$$p(y|w, s) = h(y, s) g(\boldsymbol{w}, \boldsymbol{x}) e^{\frac{b(\boldsymbol{w}, \boldsymbol{x}) y}{s}}$$

- If f is a canonical link, then

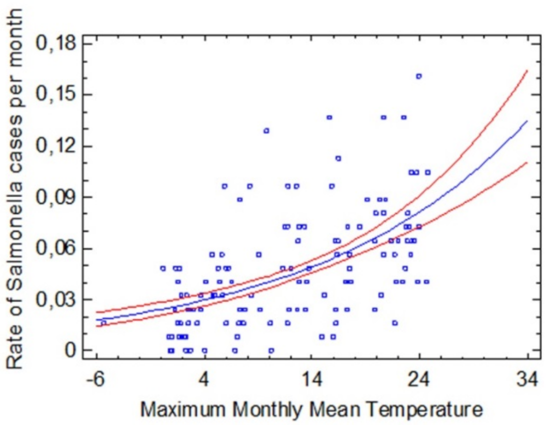
$$p(y|w, s) = h(y, s) g(\boldsymbol{w}, \boldsymbol{x}) e^{\frac{(\boldsymbol{w}^T \boldsymbol{x}) y}{s}}$$

Generalized linear models

- Canonical links are normally used
  - MLE computations simplify
  - MLE $\hat{w} = F(X^T Y)$ → computations do not depend on all data but rather a summary (sufficient statistics)→ computations speed up

Example: Poisson regression

$$f^{-1}(\mu) = e^{\mu}, Y \sim Poisson\left(e^{\boldsymbol{w}^T \boldsymbol{x}}\right)$$

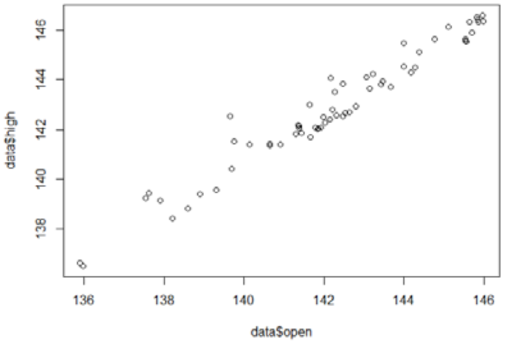


Generalized linear model: software

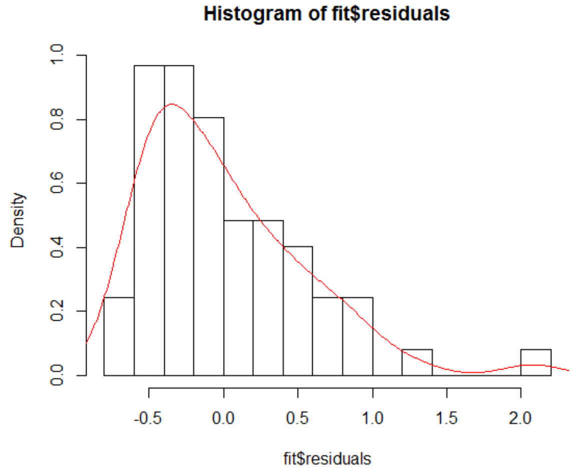
- Use **glm**(formula, family, data) in R

Example: Daily Stock prices NASDAQ

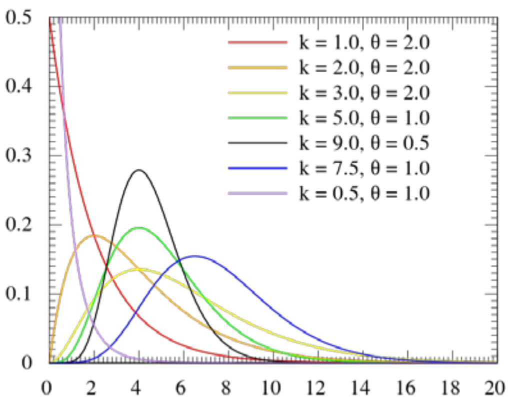
- Open
- High (within day)



1. Try to fit usual linear regression, study histogram of residuals

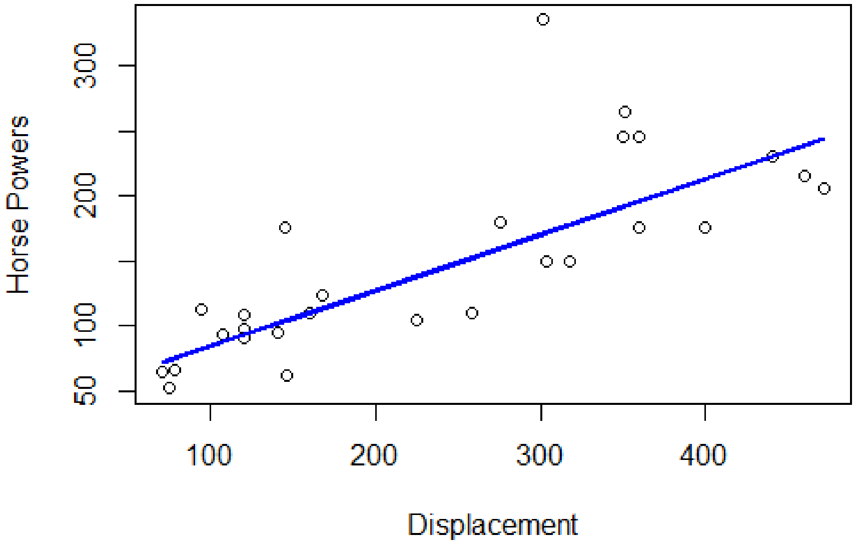


Gamma distribution: Wikipedia



Least absolute deviation regression

- Model $Y \sim Laplace(w^T X, b)$
  – Member of exponential family
- Equivalent to minimizing sum of absolute deviations



- Properties
  – Robust to outliers
  – Sensitive to changes in data
  – Multiple solutions possible

- R: package **L1pack**

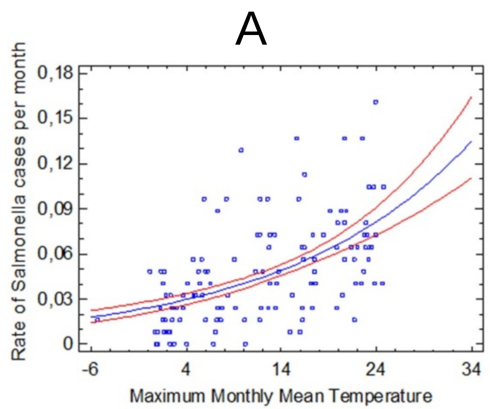
Probabilistic models

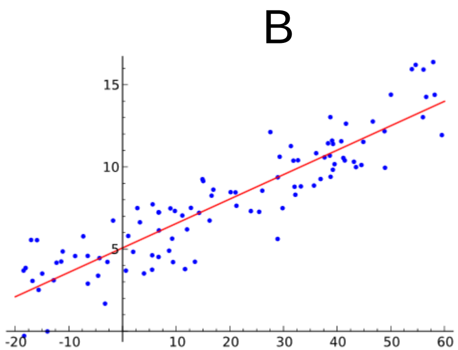
- Why it is beneficial to assume a **probabilistic** model?
- A common approach to modelling in CS and engineering:
$$y = f(x, w)$$
- f is known, w is unknown
- Fit model to data with least squares, optimization or ad hoc→ find w

Probabilistic models

Arguments against deterministic models:

- The model does not really describe actual data (error is not explained)
  - No difference between modelling data A (Poisson) and B (Normal)
  - Estimation strategy for A is not good for B
- The model typically gives a **deterministic answer**, no information about uncertainty
  - ”…The exchange rate tomorrow will be 8.22 …”





Probabilistic models

Probabilistic model

$$Y \sim Distribution(f(x,w),\theta)$$

- Data is fully explained (error as well)
- Automatic principle for finding parameters: MLE , MAP or Bayes theorem
- Automatic principle for finding uncertainty (conf. limits)
  - **Bootstrap**
  - Posterior probability
- Possibility to generate new data of the same type
  - Further testing of the model

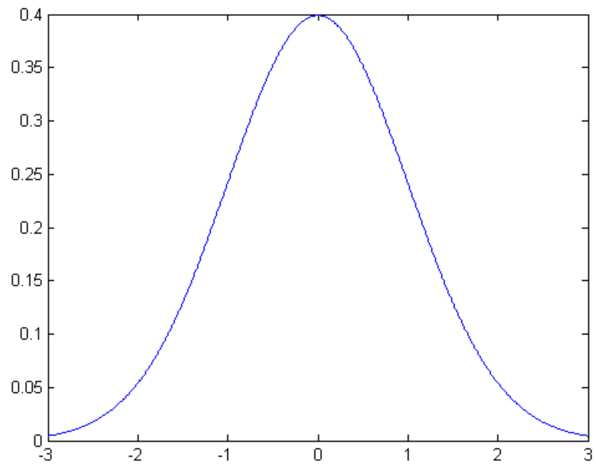
Uncertainty estimation

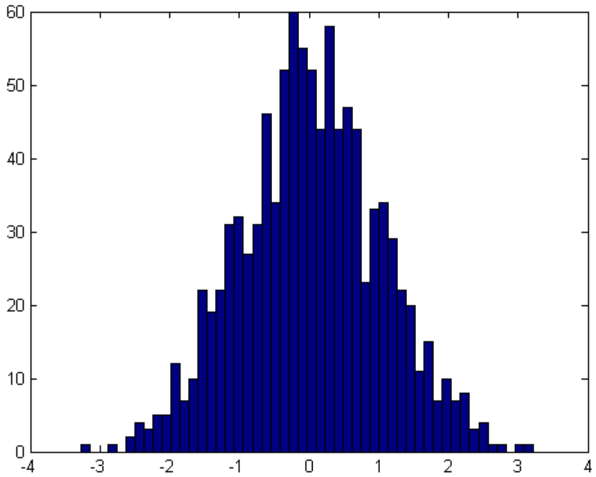
- Given estimator $\hat{f} = \hat{f}(x, D)$ (or $\widehat{\boldsymbol{\alpha}} = \delta(D)$), how to estimate the uncertainty?

- Answer 1: if the distribution for data *D* is given, compute analytically the distribution for the estimator→ derive confidence limits
  - Often difficult
  - Example: In simple linear regression, $\widehat{\alpha}$ follows *t* distribution

- Answer 2: Use **bootstrap**

The bootstrap: general principle



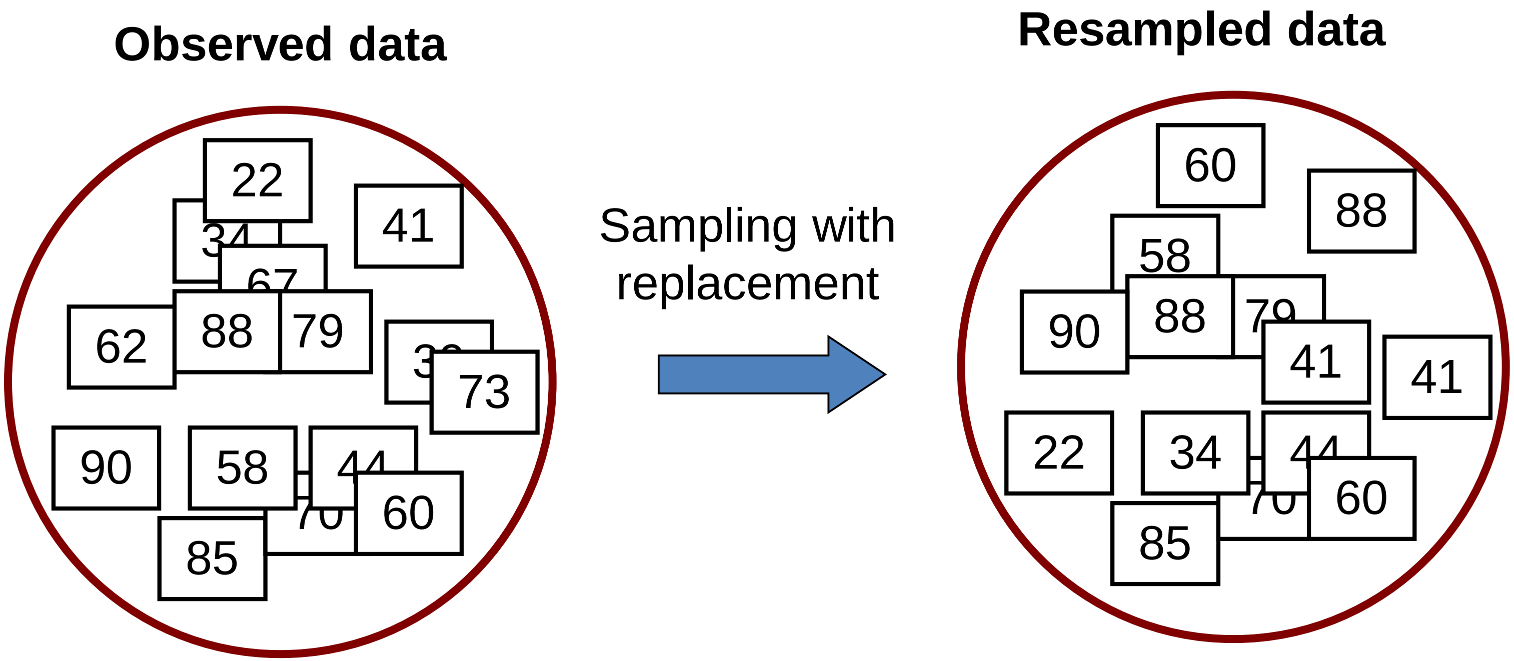


We want to determine uncertainty of $\hat{f}(D, X)$

1. Generate many different D_i from their distribution
2. Use histogram of $\hat{f}(D_i, X)$ to determine confidence limits→ unfortunately can not be done (distr of *D is often unknown)*

Instead: Generate many different D_i^* from the empirical distribution (histogram)

Nonparametric bootstrap



$\bar{x}$

$\bar{x}_1^*, \bar{x}_2^*, \ldots, \bar{x}_N^*$

Nonparametric bootstrap

Given estimator $\widehat{w} = \hat{f}(D)$

Assume $X \sim F(X, w)$, F and w are unknown

1. Estimate $\widehat{w}$ from data **D=**$(X_1,\ldots X_n)$
2. Generate $\mathbf{D_1}$ **=**$(X^*_1,\ldots X^*_n)$ by sampling with replacement
3. Repeat step 2 B times
4. The distribution of w is given by $\hat{f}(D_1), \ldots \hat{f}(D_B)$

Nonparametric bootstrap can be applied to any deterministic estimator, distribution-free

Parametric bootstrap

Given estimator $\hat{w} = \hat{f}(D)$

Assume $X \sim F(X, w)$, F is known and w is unknown

1. Estimate $\hat{w}$ from data **D=**(X_1,…X_n)
2. Generate $\mathbf{D_1}$ **=**(X^*_1,…X^*_n) by generating from $F(X, \hat{w})$
3. Repeat step 2 B times
4. The distribution of w is given by $\hat{f}(D_1), \ldots \hat{f}(D_B)$

Parametric bootstrap is **more** precise if the distribution form is correct

Uncertainty estimation

1. Get D_1 , … D_B by bootstrap
2. Use $\hat{f}(D_1)$, … $\hat{f}(D_B)$ to estimate the uncertainty
   – Boostrap percentile
   – Bootstrap Bca
   – …

- Bootstrap works for all distribution types
- Can be bad accuracy for small data sets $n < 40$ (empirical is far from true)
- Parametric bootstrap works even for small samples

Bootstrap confidence intervals

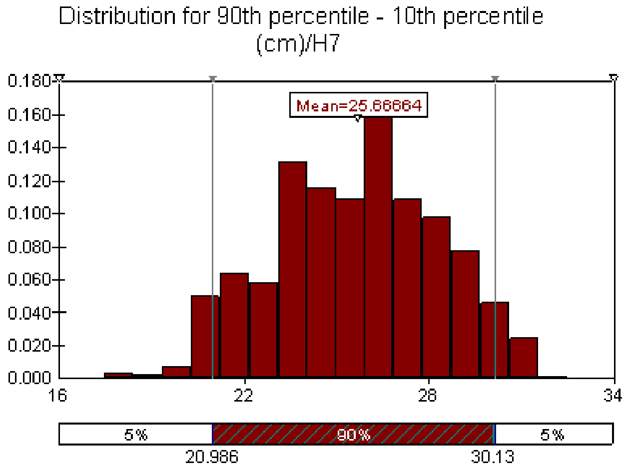
- To estimate 100(1-α) confidence interval for w

Bootstrap percentile method

1. Using bootstrap, compute $\hat{f}(D_1), \ldots \hat{f}(D_B)$, sort in ascending order, get $w_1 \ldots w_B$
2. Define A_1=ceil(B α/2), A_2=floor(B-B α/2)
3. Confidence interval is given by

$$\left(w_{A_1}, w_{A_2}\right)$$

Look at the plot…



Bootstrap: regression context

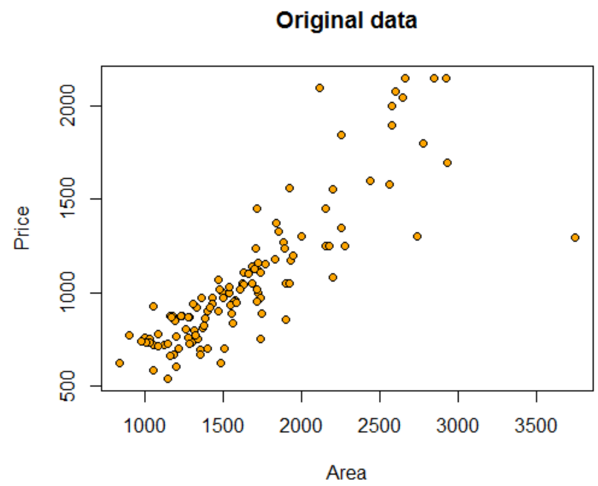
- Model $Y \sim F(X, w)$
- Data $\mathrm{D} = \{(Y_i, X_i), i = 1, \ldots, n\}$
- Idea: produce several bootstrap sets that are similar to D

Nonparametric bootstrap:

1. Using observation set **D**, sample **pairs** (X_i, Y_i) with replacement and get bootstrap sample $\mathbf{D_1}$
2. Repeat step 1 B times→ get $\mathbf{D_{1,\ldots}}$ $\mathbf{D_B}$

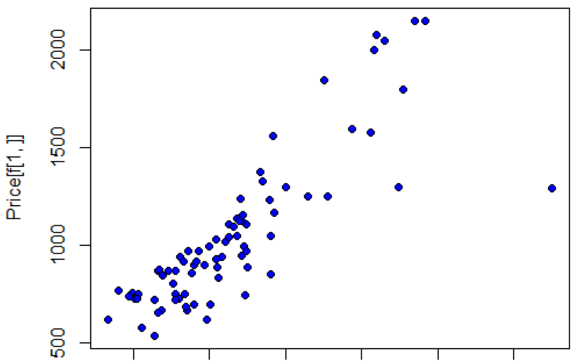
Uncertainty estimation

Example: Albuquerque dataset:
Y=Price of House
X=Area (sqft)

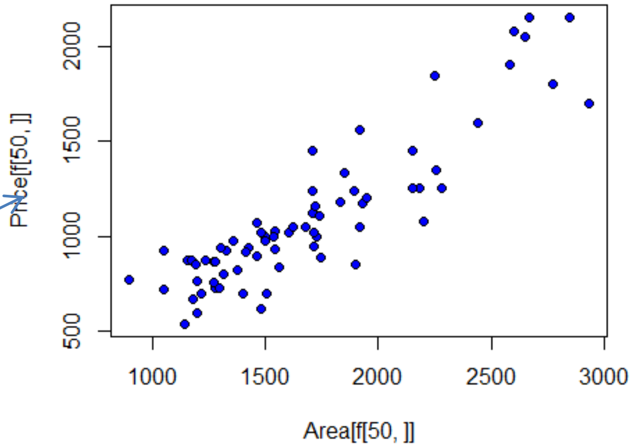


We sample data index, from {1…N}

D_1



D_{50}



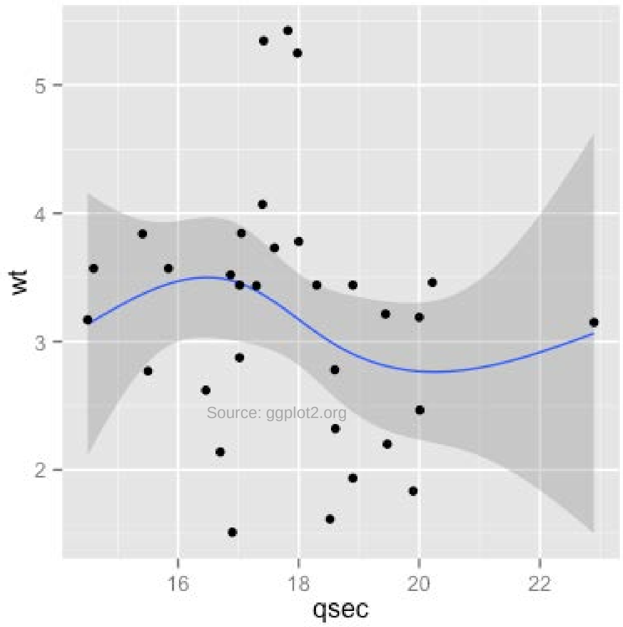
Bootstrap: regression context

Parametric bootstrap

1. Fit a model to D → get $\hat{w}(D)$.
2. Set $X_i^* = X_i$, generate $Y_i^* \sim F(X_i, \hat{w})$.
3. $D_i = \{(X_i^*, Y_i^*), i = 1, \dots, n\}$
4. Repeat step 2 B times

Confindence intervals in regression

- Given $Y \sim Distribution(y|x, w), EY|X = \mu|x = f(x, w)$
  – Example: $Y \sim N(w^T x, \sigma^2), \mu|x = f(x, w) = w^T x$
- Estimate intervals for $\mu|x = f(x, w)$ for many X, combine in a **confidence band**
- What is estimator?
  – $\mu|x = f(x, w)$



Confindence intervals in regression

Estimation

1. Compute $D_1, \ldots D_B$ using a bootstrap
2. Fit model to $D_1, \ldots D_B$ →estimate $\widehat{w}_1, \ldots \widehat{w}_B$
3. For a given X, compute $f(X, \widehat{w}_1), \ldots f(X, \widehat{w}_B)$ and estimate confidence interval by (percentile method)
4. Combine confidence intervals in a band

Bootstrap: R

- Package **boot**
  - **Functions:**
    - boot()
    - boot.ci() – 1 parameter
    - envelope() – many parameters
- Random random generation for parametic bootstrap:
  - Rnorm()
  - Runif()
  - …

boot(data, statistic, R, sim = "ordinary", ran.gen = function(d, p) d, mle = NULL,…)

Bootstrap: R

Nonparametric bootstrap:

- Write a function *statistic* that depends on *dataframe* and *index* and returns the estimator

```
library(boot)
data2=data[order(data$Area),]#reordering data according to Area

# computing bootstrap samples
f=function(data, ind){
  data1=data[ind,]# extract bootstrap sample
  res=lm(Price~Area, data=data1) #fit linear model
  #predict values for all Area values from the original data
  priceP=predict(res,newdata=data2)
  return(priceP)
}
res=boot(data2, f, R=1000) #make bootstrap
```

Bootstrap: R

Parametric bootstrap:

- Compute value *mle* that estimates model parameters from the data
- Write function *ran.gen* that depends on *data* and *mle* and which generates new data
- Write function *statistic* that depend on *data* which will be generated by *ran.gen* and should return the estimator

Bootstrap

```
mle=lm(Price~Area, data=data2)

rng=function(data, mle) {
  data1=data.frame(Price=data$Price, Area=data$Area)
  n=length(data$Price)
#generate new Price
  data1$Price=rnorm(n,predict(mle, newdata=data1),sd(mle$residuals))
  return(data1)
}

f1=function(data1){
  res=lm(Price~Area, data=data1) #fit linear model
  #predict values for all Area values from the original data
  priceP=predict(res,newdata=data2)
  return(priceP)
}

res=boot(data2, statistic=f1, R=1000, mle=mle,ran.gen=rng, sim="parametric")
```

Uncertainty estimation: R

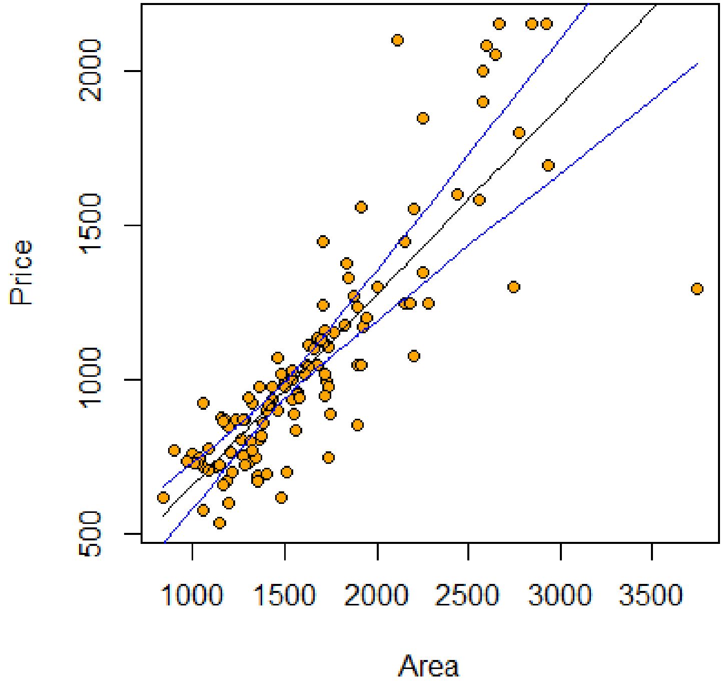
- Bootstrap cofidence bands for linear model

```
e=envelope(res) #compute confidence bands

fit=lm(Price~Area, data=data2)
priceP=predict(fit)

plot(Area, Price, pch=21, bg="orange")
points(data2$Area,priceP,type="l") #plot fitted line

#plot cofidence bands
points(data2$Area,e$point[2,], type="l", col="blue")
points(data2$Area,e$point[1,], type="l", col="blue")
```



Prediction bands

- Confidence interval for Y|X= interval for mean $EY|X$
- Prediction interval for Y|X= interval for $Y|X$

$$Y \sim Distribution(x, w)$$

Prediction band for parametric bootstrap

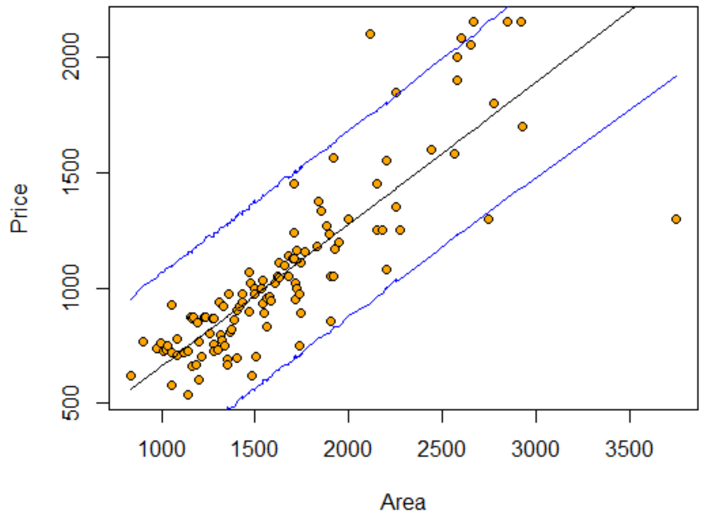
1. Run parametric bootstrap and get $D_1, \ldots D_B$
2. Fit the model to the data and get $\widehat{w}(D_1), \ldots \widehat{w}(D_B)$
3. For each X, generate from $Distribution\big(X, \widehat{w}(D_1)\big), \ldots$ $Distribution\big(X, \widehat{w}(D_B)\big)$ and apply percentile method
4. Connect the intervals→get the band

Estimation of the model quality

Example: parametric bootstrap

```
mle=lm(Price~Area, data=data2)

f1=function(data1){
  res=lm(Price~Area, data=data1) #fit
linear model
  #predict values for all Area values
from the original data
  priceP=predict(res,newdata=data2)
  n=length(data2$Price)
  predictedP=rnorm(n,priceP,
sd(mle$residuals))
  return(predictedP)
}
res=boot(data2, statistic=f1, R=10000,
mle=mle,ran.gen=rng, sim="parametric")
```



Why wider band?