Проучвателен анализ на данни.

Проучвателният анализ на данни е процес на разглеждане на данните и търсене на подходящи статистически техники на базата, на които могат да бъдат направени статистически заключения. За едномерни данни например, можем да проверим дали данните са от наблюдения върху нормално разпределена случайна величина; дали наблюдаваното разпределение е едномодално, бимодално или многомодално; дали е асиметрично; дали е изострено. Основната статистическа техника на този етап са статистическите графики.

Нашият toolbox

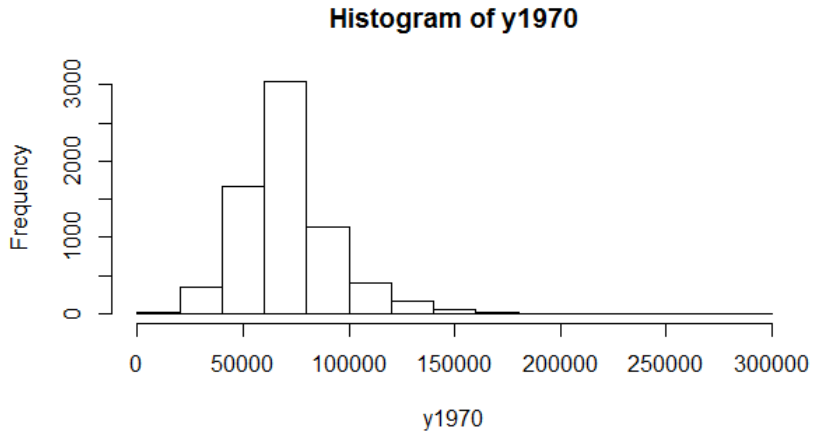
Ето и преглед на основните графични методи, които сме изучили до сега, и които могат да се използват за проучвателен анализ:

- ***barplots*** за категорийни данни;
- ***histogram, dot plots*** или ***stem and leaf*** графики, за определяне на разпределението по матриран признак
- ***boxplots*** за графично представяне на основните числови характеристики и силно отличаващи се наблюдения на количествени признаци, както и за определянето на по-тежки опашки в сравнение с нормалните.
- ***normal probability plots*** – за проверка дали наблюдаваното разпределение е приблизително нормално

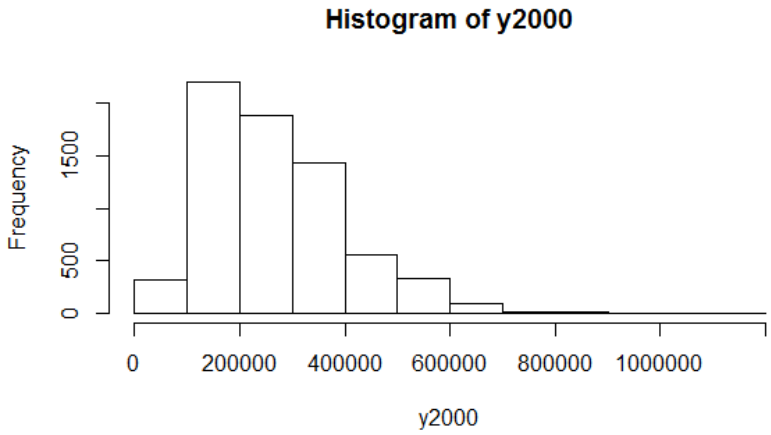
Функцията ***simple.eda*** от библиотеката ***UsingR*** изчертава и трите: хистограма, графика с мустачки и normal probability графика на вектора с данните, зададени като параметър.

Например: Да разгледаме данните homedata, от библиотеката ***UsingR,*** съдържащи оценки на жилищата от кленова (яворова, Maplewood) дървесина, NJ за годините 1970 и 2000. Каква е формата (типа) на това разпределение?

```
> data(homedata) # from simple package
> ls(homedata)
[1] "y1970" "y2000"
> attach(homedata)
> hist(y1970);
```



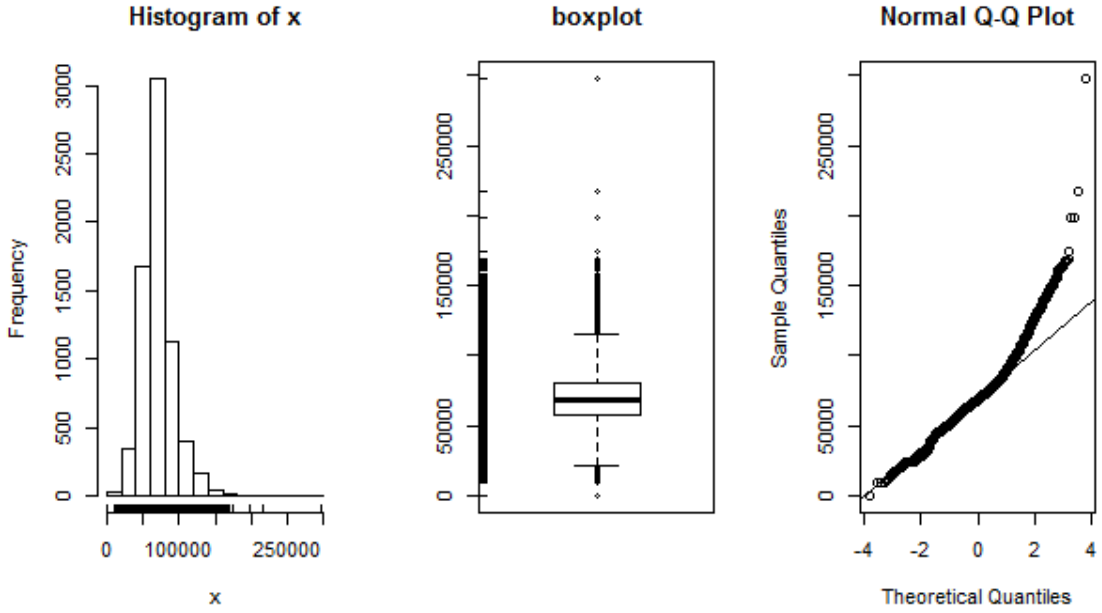

```
> hist(y2000)
```



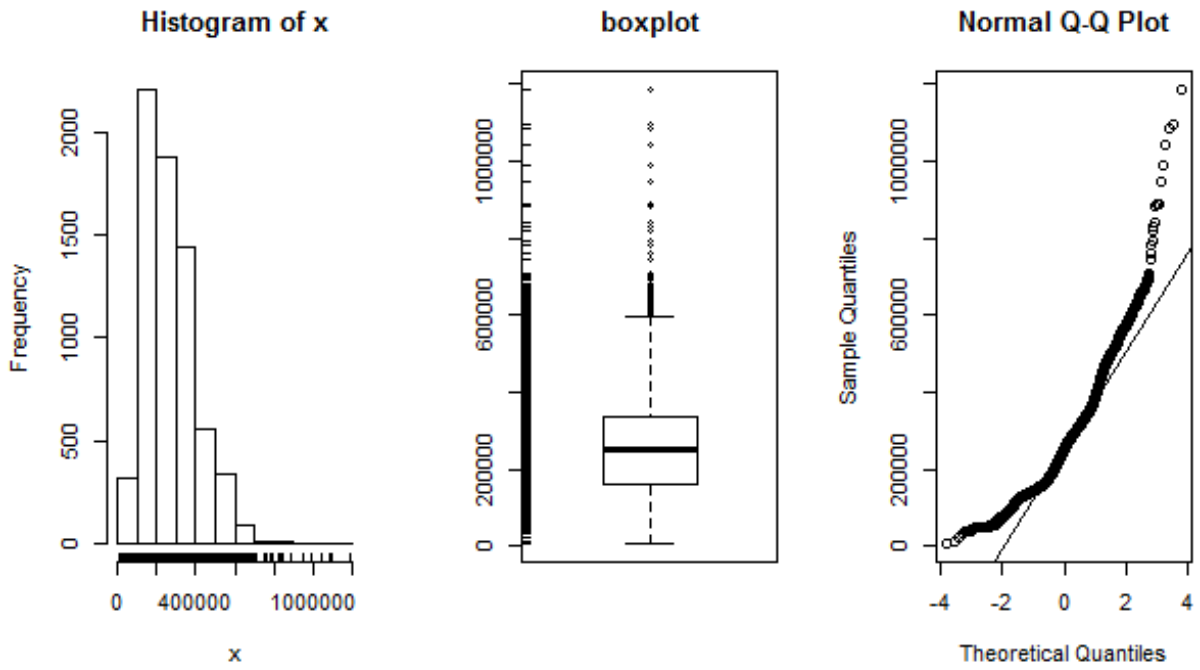
Очевидно се наблюдава поскъпване на жилищата.

Сега да изчертаем едновременно хистограма, графика с мустачки и normal probability графика на вектора с данните от тези два вектора, като използваме функцията ***simple.eda***.

```
> simple.eda(y1970);
```



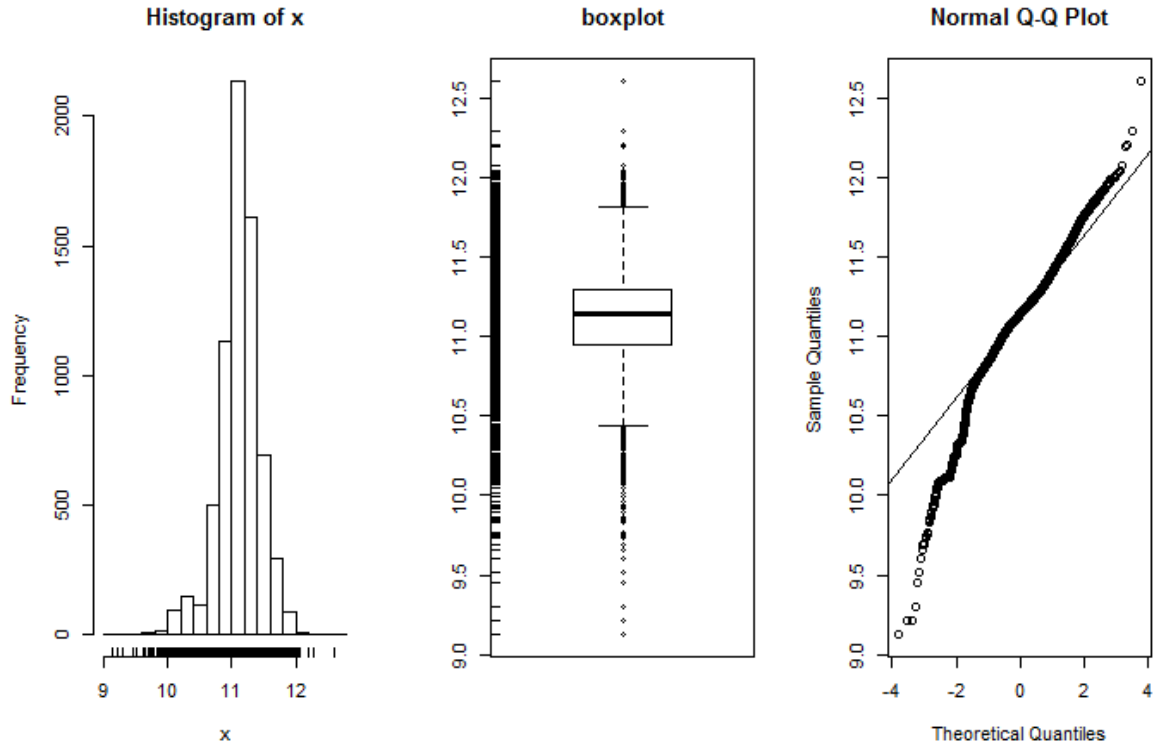

```
> simple.eda(y2000)
```



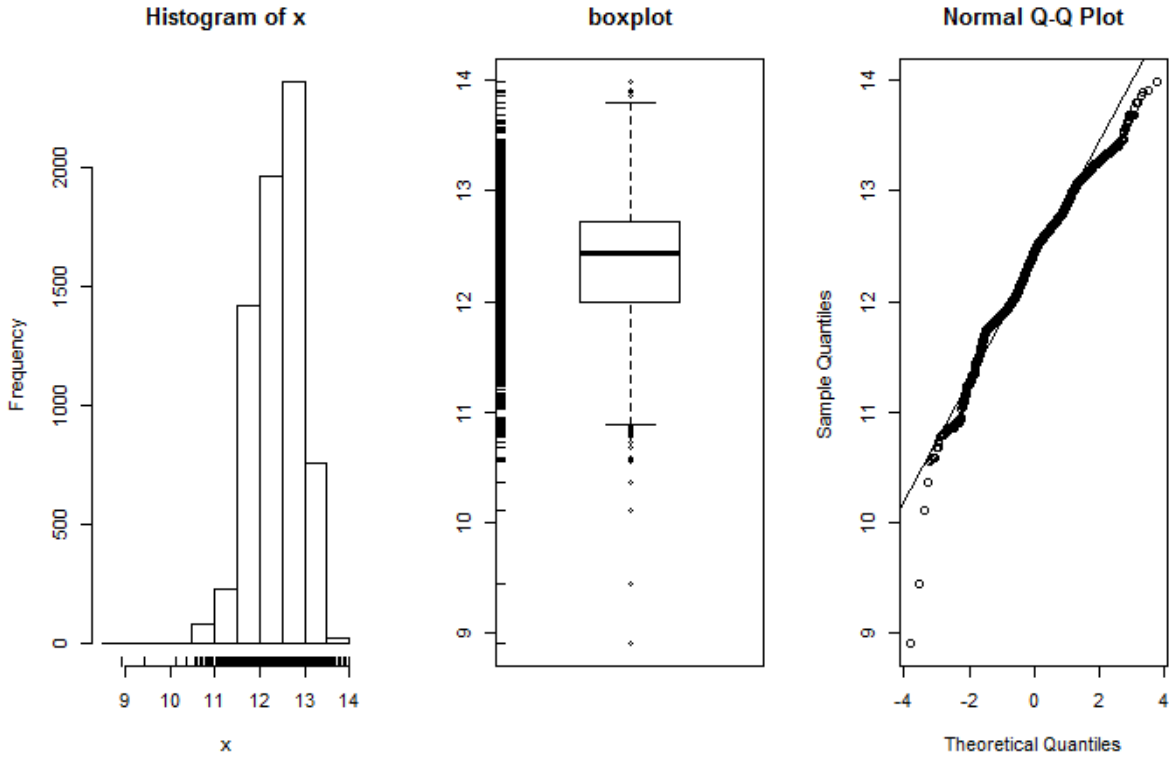
Виждаме, че разпределението на тази величина през 1970 е по-близко до нормалното, в сравнение с това през 2000. През двете години, обаче наблюдаваните величини не са нормално разпределени, защото графиките от normal qq-plot показват, че точките не лежат върху ъглополовящата на първи и трети квадрант. И в двата случая имаме по-тежка дясна опашка. Асиметрията е дясна, защото имаме струпване на наблюденията около стойностите, които са по-малки от средното аритметично. В такива случаи обикновено се прави логаритмична трансформация, за да се направи разпределението на новата величина малко по-симетрично и по-лесно за анализ. Всъщност логаритмичната трансформация измества тежестта на разпределението от малките към по-големите стойности. При това трябва да се внимава, защото както знаем, можем да логаритмуваме само положителни числа.

Сега да изчертаем аналогични графики за логаритмите на цените през двете години поотделно.

```
> simple.eda(log(y1970[y1970>0]));
```




```
> simple.eda(log(y2000[y2000>0]));
```



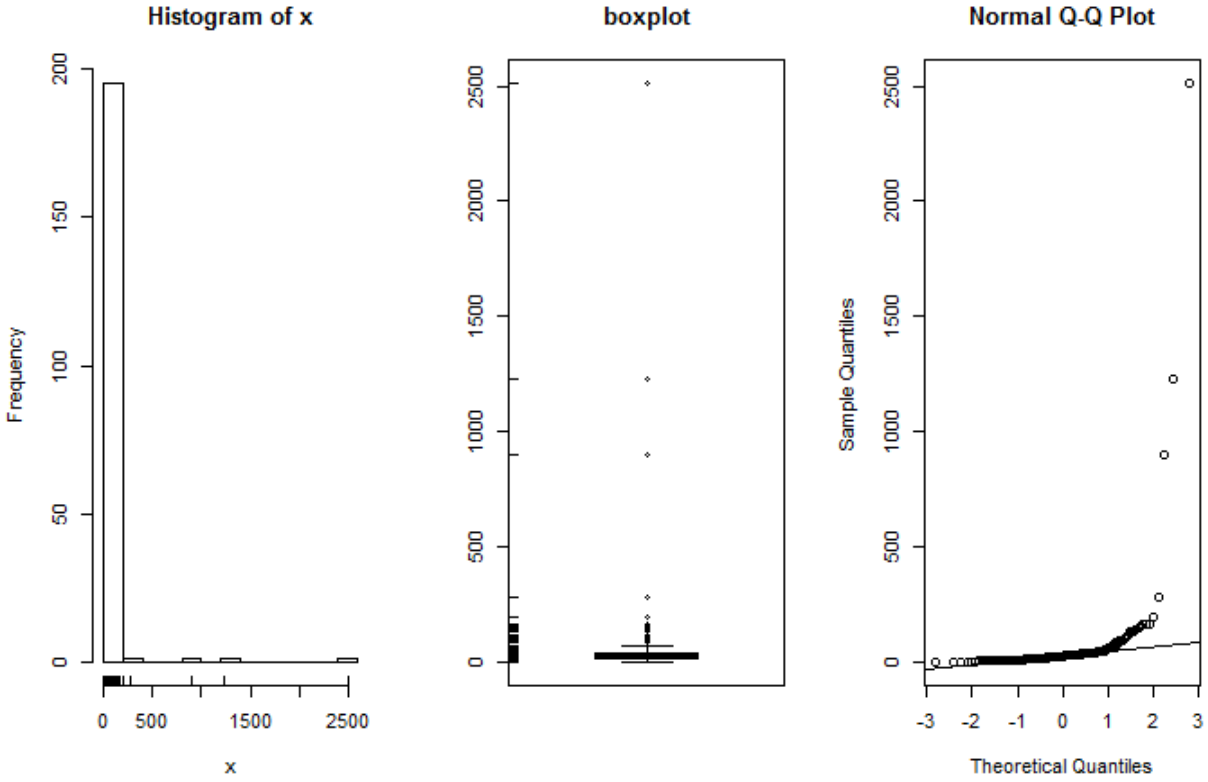

```
> detach(homedata)
```

Виждаме, че разпределенията на логаритмуваните данни са много по-симетрични и разпределението им е по-близко до нормалното, но все още тяхното разпределение не е нормално. [1]

Пример: Разгледайте данните за общите директни компенсации за CEO (Chief Executive Officer - Главен изпълнителен директор) в 200 големи публични търговски компении в U.S за 2000г. (в по $100 000). Т.е. това са добавки към заплатата. Те са намират в множеството exec.pay. Какво можете да кажете за разпределението на наблюдаваната величина?

Отг. Чрез ***simple.eda*** получаваме

```
> data(exec.pay)
> simple.eda(exec.pay)
```

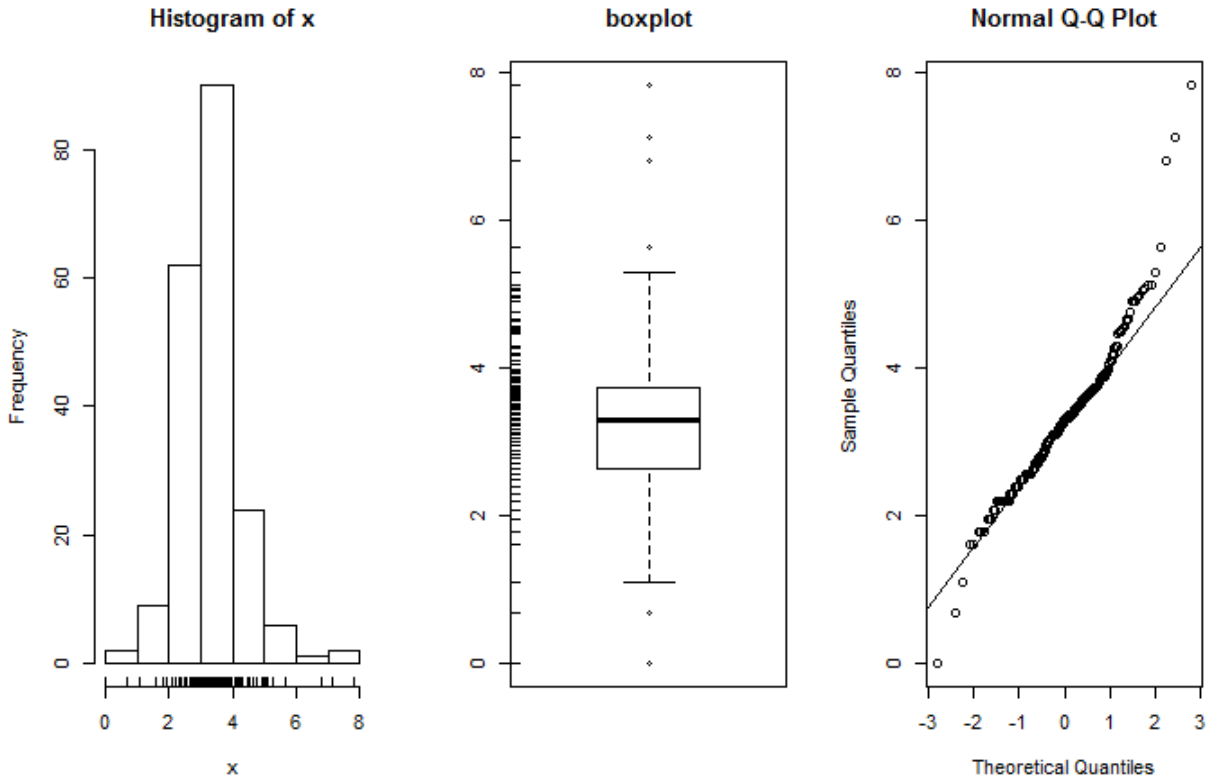


Наблюдаваме силна концентрация върху малките стойности, т.е. силна дясна асиметрия. Дясна, защото коефициентът на асиметрия ще е положителен. Т.е. имаме много на брой малки плащания и съвсем малък относителен дял на силно отличаващи се плащания. Отново опитваме с логаритмуване да симетризираме разпределението си. Т.к. някои CEO плащания са нули (тези, които по принцип си имат големи заплати нямат нужда от компенсации) и не можем да ги логаритмуваме, разглеждаме само тези, които са по-големи от нула.

```
> l = log(exec.pay[exec.pay > 0])
> simple.eda(l)
```

---

[1] От познатите ни разпределения няма такова, което по-добре да описва данните. Трябва да се търси трансформация на данните, но това е изследователска работа и резултата е достоен за статия.



Разпределението на логаритъма на плащанията е значително по-симетрично.

Разпределения на величини, чиито логаритми са нормални се наричат логаритмично нормални разпределения. Т.е. наблюдаваната величина има приблизително логаритмично нормални разпределение.

Пример: Множеството от данни ewr съдържа времената за влизане и излизане на такситата от аерогара Newark. Характеризирайте разпределенията на наблюдаваните величини.

```
> data(ewr)
```

Разглеждаме имената на данните. Командата ***ls*** ги подрежда автоматично по азбучен ред, а не по реда в колоните.

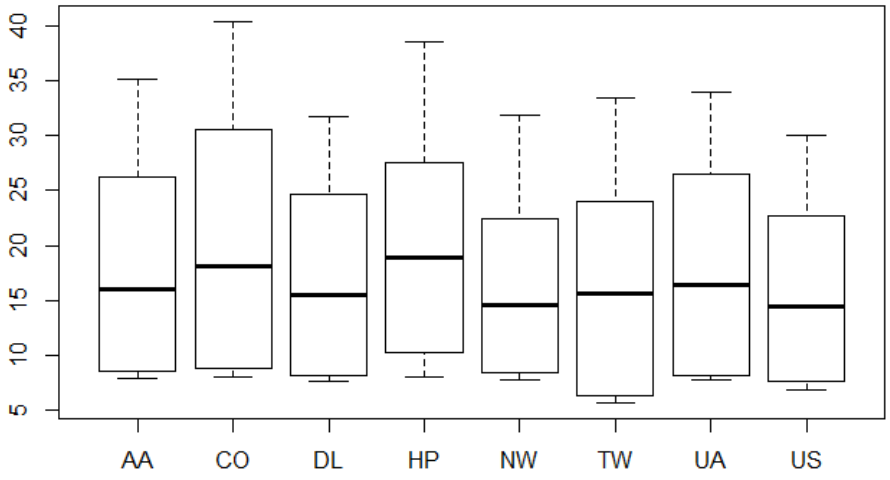
```
> ls(ewr)
 [1] "AA"     "CO"     "DL"     "HP"     "inorout" "Month"   "NW"
 [8] "TW"     "UA"     "US"     "Year"
> head(ewr)
  Year Month  AA  CO  DL   HP  NW   TW  UA  US inorout
1 2000   Nov 8.6 8.3 8.6 10.4 8.1  9.1 8.4 7.6      in
2 2000   Oct 8.5 8.0 8.4 11.2 8.2  8.5 8.5 7.8      in
3 2000   Sep 8.1 8.5 8.4 10.2 8.3  8.6 8.2 7.6      in
4 2000   Aug 8.9 9.1 9.2 14.5 9.0 10.3 9.2 8.7      in
5 2000   Jul 8.3 8.9 8.2 11.5 8.8  9.1 9.2 8.2      in
6 2000   Jun 8.8 9.0 8.8 14.9 8.4 10.8 8.9 8.3      in
```

За да изведем имената, в реда в който е в колоните използваме функцията ***names***.

```
> names(ewr)
[1] "Year"   "Month"  "AA"     "CO"     "DL"     "HP"     "NW"
```

[8] "TW" "UA" "US" "inorout"

```
> ewr.actual = ewr[ , 3:10]        # вземаме само тези, които ни трябват.
> boxplot(ewr.actual)
```

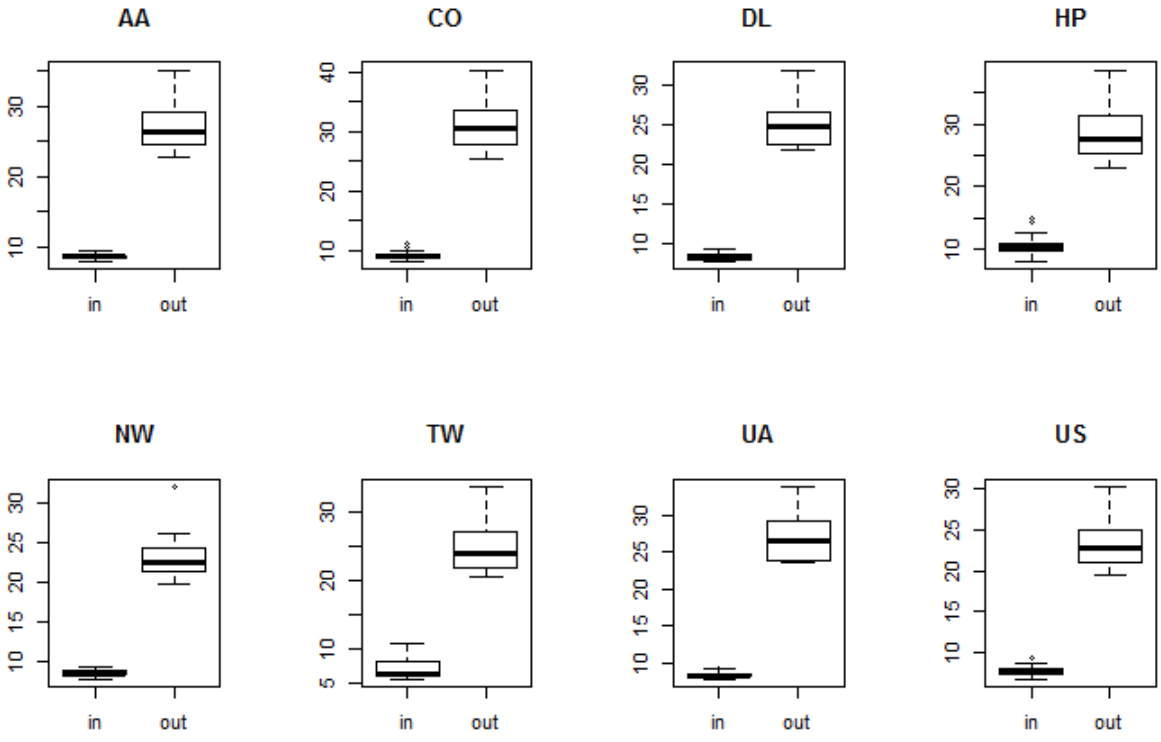


Всички наблюдавани разпределения изглеждат асиметрични. Нека сега да ги разделим, според това дали се отнасят за влизане или за излизане.

```
> par(mfrow = c(2, 4))                    # 2 реда 4 колони
> attach(ewr)
> airnames = names(ewr)
```

В следващия ред, за да разделим анализите по подгрупи, използваме формула, като независимата променлива, която стои след знака „~“ трябва да е фактор. За да я превърнем в такава използваме функцията ***as.factor.*** Друг интересен момент е, че сме извели заглавията автоматично, чрез написване на променливата, която ги съдържа.

```
> for(i in 3:10)
{
boxplot(ewr[,i] ~ as.factor(inorout), main = airnames[i])
}
> detach(ewr)
```



Наблюдаваме, че времената за излизане са много по-дълги от времената за влизане в летището.

```
> par(mfrow = c(1, 1))      # връща графичния прозорец в предишния вид за
                            # изчертаване само на една графика.
```

Нататък можем да извършим проверка на хипотези, които ще разгледаме по-нататък.