Einführung in die Angewandte Bioinformatik: Datenanalyse mit R
20.05.2010

Prof. Dr. Sven Rahmann

Funktionsaufruf mit benannten Parametern

Es ist Konvention, einer Funktion erst die nötigen Daten zu übergeben;
danach benannte Parameter, die das Verhalten der Funktion verändern.

Beispiel: Plot mit Linien und Punkten in Blau mit Titel:

```
plot(x,y, type="o", col="blue", main="Ein Funktionsplot")
```

Beispiel: Logarithmische Achsen
Achsen können mit logarithmischer Skala darzustellen.
Dies geschieht (für die y-Achse) durch Angabe des Parameters log="y".
Monome der Form $y=cx^n$ werden bei logarithmischen Achsen zu Geraden.

```
x = seq(1, 5, by=1/16)
y = 5 * x**3
plot(x,y, type="o")
plot(x,y, type="o", log="xy")  # beide Achsen logarithmisch
```

Beschreibende Statistik mit R

Sei v ein Vektor von reellen Zahlen.
Wir definieren und berechnen einige Kennzahlen.

Mittelwert	mean(v)	Durchschnitt
Varianz	var(v)	Mittlere quadrat. Abw. zum MW
Standardabweichung	sd(v)	Wurzel aus Varianz
Median	median(v)	50% der Werte kleiner, 50% größer
p-Quantil	quantile(v,p)	Anteil *p* der Werte kleiner; 1-*p* größer
p-Quantile	quantile(v,p)	*p* kann Vektor sein!
Interquartilabstand	IQR(v)	75%-Quantil – 25%-Quantil

Beschreibende Statistik mit R

5 Kennzahlen fivenum(v) # dazu ?in R fivenum lesen

- Mimimum (oder mindestens Median – 1.5*IQR)
- 25%-Quantil
- Median
- 75%-Quantil
- Maximum (oder höchstens Median + 1.5* IQR)

Boxplot boxplot(v) # Visualisierung von fivenum

Histogramm hist(v) # Anzahl der Zellen sinnvoll gewählt
hist(v,*n*) # Histogramm mit *n* Zellen

Ein Histogramm zeigt zu jedem Wert(ebereich) die Anzahl der Elemente in v an, die in diesen Bereich fallen.

Tabellen (data frames) einlesen

Wie kommen die Daten in den R-Workspace ?
- Manuelle Eingabe: langsam, umständlich
- Einlesen aus Datei: schnell, aber Format muss beachtet werden

Aktuelles Verzeichnis anzeigen / wechseln (wie `cd` in der Shell):
`getwd()` / `setwd("neuesVerzeichnis")`

Tabelle aus Datei in Variable x einlesen:
```
x = read.table(dateiname, ...)
```

Optionen dabei z.B.:
`sep = ""` oder `sep=","` oder `sep=";"` (Trennungszeichen der Elemente)
`header = FALSE` oder `header=TRUE` (erste Zeile enthält Namen?)
`col.names = vektor` (Spaltennamen explizit angeben)

Mit Tabellen (data frames) arbeiten

Eingelesen Daten `x` stehen in sog. **data frames** (Datenrahmen = Tabellen).
Jede Zeile ist ein Datensatz; jede Spalte repräsentiert ein Attribut der Daten.

Größe anzeigen lassen: `dim(x)`
Spalten haben Namen (aus Datei oder automatisch): `colnames(x)`
Spalten umbenennen: z.B. `colnames(x) = c("Anna", "Bert")`
Kurzübersicht: `str(x)`

Zugriff auf Zeile 17: `x[17,]` (Komma beachten!)
Zugriff auf mehrere Zeilen 17 bis 23: `x[17:23,]` (Komma beachten!)
Zugriff auf Spalte "Anna" als Vektor: `x$Anna` oder `x[,"Anna"]`
Auf Spalten direkt (ohne Präfix `x$`) zugreifen: `attach(x)`

Tabelle wieder speichern: `write.table(x, Dateiname, ...)`

Erste statistische Analysen mit R

Vergleich von zwei Vektoren x,y gleicher Länge

Scatterplot	plot(x,y)	Punkte (x[i],y[i])
Korrelationskoeffizient	cor(x,y)	+-1 bei linearer Abhängigkeit 0 bei Unabhängigkeit

Komplexes Beispiel einer beschreibenden Datenanalyse

Mit einer neuen Sequenziertechnologie (ABI SOLiD) wurden
kurze nicht-codierende RNA-Stücke (ncRNA reads) sequenziert.

Gegeben

- FASTA-Datei mit ca. 670000 Sequenzen der Länge 35.
- FASTA-ähnliche Datei mit Qualitätswerten (-10*log10(Fehlerwahrscheinlichkeit))
  ebenfalls ca. 670000 Sequenzen à 35 Werte (.qual-Datei)

Fragestellung

Betrachte nur die Qualitätswerte, nicht die Sequenzen.
Nimmt die Qualität „hinten“ ab?
Sind Qualitätswerte an Positionen 30+ niedriger als an den ersten Positionen?

Erste Schritte – Daten anschauen

Gegeben

FASTA-ähnliche Datei mit ca 670000 x 35 Qualitätswerten (.qual-Datei)

In der Shell

```
% ls -l reads.qual          # wie groß ist die Datei?
% head -n 30 reads.qual     # erste 30 Zeilen
% tail -n 30 reads.qual     # letzte 30 Zeilen
% cat reads.qual            # ganze Datei (Unsinn!)
% more reads.qual           # durchblättern
% less reads.qual           # wie more
% wc less.qual              # Wie viele Zeilen?
```

FASTA-ähnliche Datei mit Qualitätswerten

```
...
# Title: s0329_20090331_552to561_613to614_2_552_561
>854_648_594_F3
25 27 27 2 29 30 25 3 2 27 26 27 4 5 25 27 24 2 2 28 28 31 3 3 21 26 30 3 2 27 21 26 5 5 6
>854_824_731_F3
14 20 20 23 23 14 26 20 28 26 18 26 22 26 30 14 25 24 26 28 17 28 18 28 29 7 25 11 26 27 18 21
>854_1300_825_F3
32 28 27 26 24 28 30 27 26 21 24 29 26 30 29 27 30 23 28 29 26 31 26 31 29 21 25 28 19 27 23 24
>855_103_1176_F3
31 25 27 28 32 29 28 28 19 32 30 25 23 20 19 20 21 27 29 28 19 21 26 27 26 28 28 29 19 25 5 30
>855_133_1168_F3
26 32 28 25 15 31 27 29 19 28 27 27 24 24 29 29 24 25 23 28 25 25 8 27 29 25 24 25 13 25 31 8 1
...
```

Es gibt
- Kommentarzeilen (mit #)
- Kopfzeilen (mit >)
- Datenzeilen (Zahlen)

Wir wollen nur die Qualitätswerte (670000 x 35 – Matrix) extrahieren.

Erzeugung einer Datei mit ausschließlich Qualitätswerten

```
% grep -v '[#>]' reads.qual > q.txt
% head q.txt
% wc q.txt                                    # 676773 Zeilen
```

Die Option -v bei grep invertiert die Logik und gibt die Zeilen aus,
die die Bedingung, # oder > zu enthalten, nicht erfüllen:

```
25 27 27 2 29 30 25 3 2 27 26 27 4 5 25 27 24 2 2 28 28 31 3 3 21 26 30 3 2 27 21 26 5 5 6
14 20 20 23 23 14 26 20 28 26 18 26 22 26 30 14 25 24 26 28 17 28 18 28 29 7 25 11 26 27 18 21
32 28 27 26 24 28 30 27 26 21 24 29 26 30 29 27 30 23 28 29 26 31 26 31 29 21 25 28 19 27 23 24
31 25 27 28 32 29 28 28 19 32 30 25 23 20 19 20 21 27 29 28 19 21 26 27 26 28 28 29 19 25 5 30
26 32 28 25 15 31 27 29 19 28 27 27 24 24 29 29 24 25 23 28 25 25 8 27 29 25 24 25 13 25 31 8 1
29 24 23 27 24 5 8 25 24 32 26 18 27 21 28 23 26 27 22 20 24 21 19 15 25 27 18 24 8 26 24 18 24
28 30 17 32 29 24 28 28 23 24 30 30 26 24 22 27 23 26 21 22 23 25 27 11 18 29 23 27 4 16 28 24
```

Einlesen in R

```
% R                              # Shell: Starten von R

> q = read.table('q.txt')       # R: Einlesen der Datei

> dim(q)                        # Orientierung: Größe?
[1] 676773     35

> colnames(q)                   # Namen der 35 Spalten: automatisch
 [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11"
[13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23"
[25] "V25" "V26" "V27" "V28" "V29" "V30" "V31" "V32" "V33" "V34" "V35"
```

Mittlere Qualität und Variabilität jeder Position

```
> m = mean(q)
> s = sd(q)

> plot(m, col="red", type="o")    # Plot der Mittelwerte

> X11()
> plot(s, pch="+", col="blue", type="o")
```

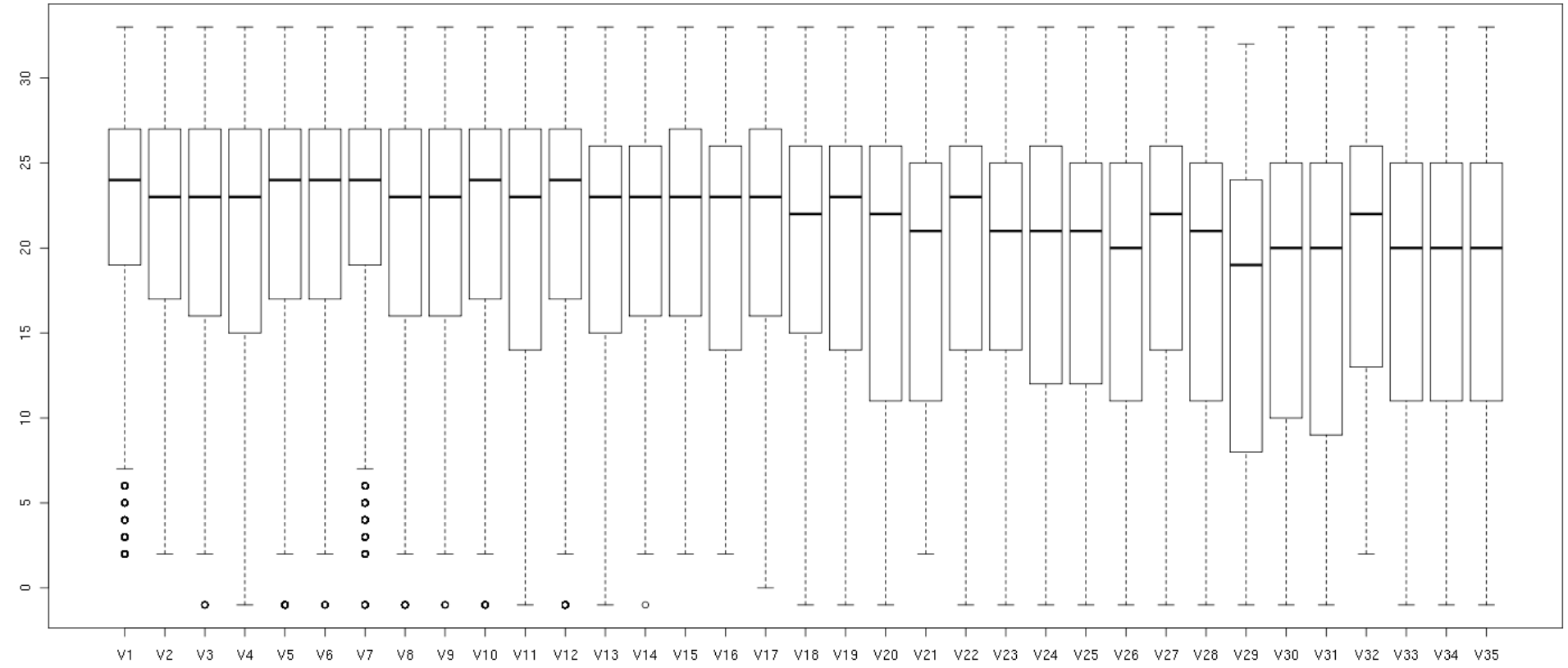
Merke

- Mit X11() kann man ein weiteres Plot-Fenster öffnen.
- Die einfachen beschreibenden Statistiken (wie `mean`, `sd`) arbeiten spaltenweise auf Tabellen (dataframes)!
- Das funktioniert leider nicht mit `median`, `IQR`.
- Aber mit `boxplot`...

Boxplot der Qualität jeder Position

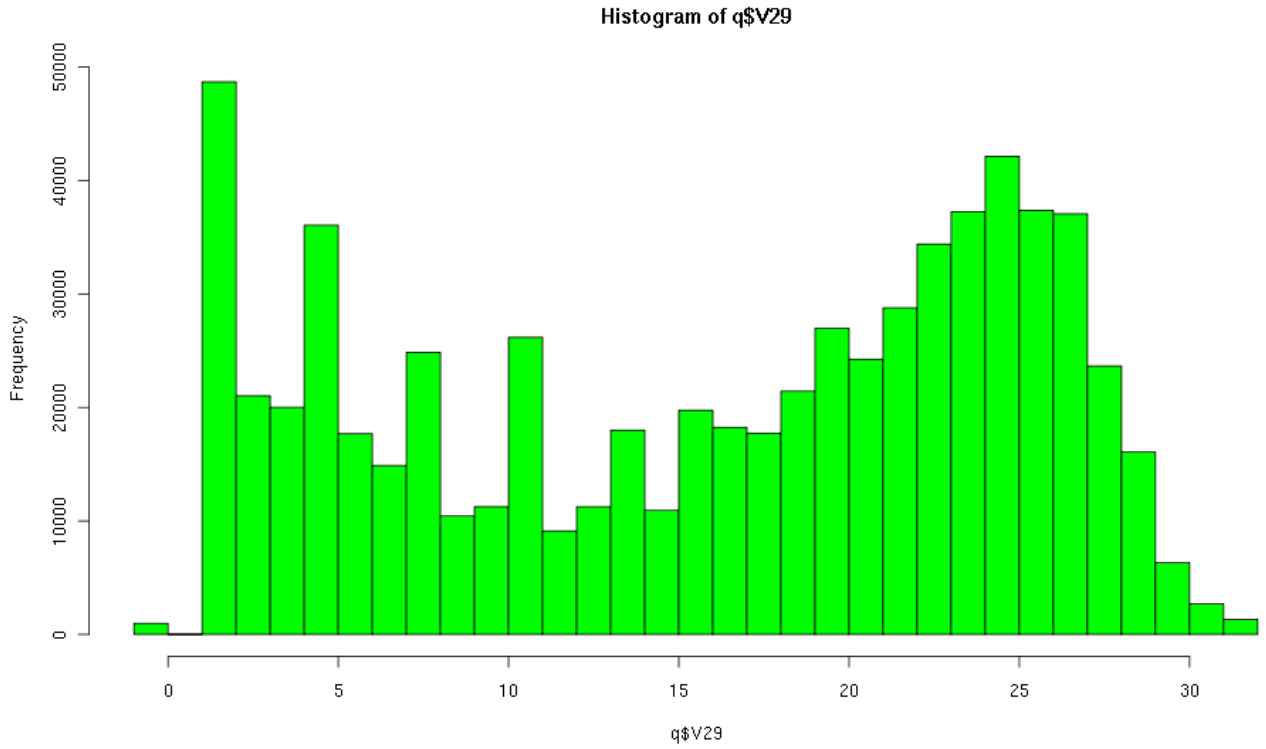
```
> boxplot(q)          # dauert...
```

Position 29 fällt auf.



Qualitätshistogram (Position 29)

```
> attach(q)                      # spart Tipparbeit
> min(V29)                       # min. Qualität an Position 29
> max(V29)                       # max. Qualität an Position 29
> hist(V29,35,col="green")       # Histogramm
```



Korrelieren Mittelwert und Standardabweichung?

```
> plot(m,s, col="blue")     # Scatterplot m gegen s
> cor(m,s)                  # Tendenz?
```