**Aufgabe 1:**

Lesen Sie den Datensatz `nba.asc` aus dem Datenarchiv des Instituts für Statistik (`http://www.statistik.lmu.de/service/datenarchiv/nba/nba.html`) in `R` ein und überpüfen Sie den Datensatz auf eventuelle Fehler und beheben Sie diese falls vorhanden.

Exportieren Sie den Datensatz anschließend in die kontinental-europäische Version des `CSV`-Formats und kontrollieren Sie die Datei mit Excel.

Auf der Veranstaltungshomepage finden Sie zusätzlich die Dateien `nba.sav` und `nba.xlsx`. Versuchen Sie auch diese Dateien einzulesen.

Vergleichen Sie zwei der resultierenden Ergebnisse. Tipp: Eventuell müssen die Datensätze vorher sinnvoll sortiert werden.

**Lösung:**

```
> # speichern der URL in eigener Variable
> url_nba <-
+   "http://www.statistik.lmu.de/service/datenarchiv/nba/nba.asc"
> # Speichern des Datenstazes
> nba <- read.table(file = url_nba, header = TRUE)
> # Anschauen der Struktur
> str(nba)

'data.frame':        1189 obs. of  6 variables:
$ heimteam: Factor w/ 29 levels "Atlanta","Boston",..: 2 17 20 19 14 8 26 16 10
   7 ...
$ heimtore: int 98 77 103 92 94 95 99 82 96 91 ...
$ gastteam: Factor w/ 29 levels "Atlanta","Boston",..: 4 5 15 29 1 11 18 24 23
   6 ...
$ gasttore: int 107 90 111 96 81 89 107 78 85 92 ...
$ wochtag : Factor w/ 8 levels "Friday","Monday",..: 1 1 1 1 1 1 1 1 1 1 ...
$ datum : int 961101 961101 961101 961101 961101 961101 961101 961101 961101
   961101 ...
```

Die Variable `datum` könnte man in das Format Date umwandeln.

```
> # Umwandlung der Variable 'datum'
> nba <- transform(nba,
+   datum = as.Date(as.character(datum), format = "%y%m%d"))
> str(nba)

'data.frame':        1189 obs. of  6 variables:
$ heimteam: Factor w/ 29 levels "Atlanta","Boston",..: 2 17 20 19 14 8 26 16 10
   7 ...
$ heimtore: int 98 77 103 92 94 95 99 82 96 91 ...
$ gastteam: Factor w/ 29 levels "Atlanta","Boston",..: 4 5 15 29 1 11 18 24 23
   6 ...
$ gasttore: int 107 90 111 96 81 89 107 78 85 92 ...
$ wochtag : Factor w/ 8 levels "Friday","Monday",..: 1 1 1 1 1 1 1 1 1 1 ...
$ datum : Date, format: "1996-11-01" "1996-11-01" ...

> summary(nba)
```

```
     heimteam      heimtore          gastteam      gasttore
 Atlanta  : 41   Min.   : 57.00   Atlanta  : 41   Min.   : 59.00
 Boston   : 41   1st Qu.: 91.00   Boston   : 41   1st Qu.: 87.00
 Charlotte: 41   Median : 98.00   Charlotte: 41   Median : 96.00
 Chicago  : 41   Mean   : 98.08   Chicago  : 41   Mean   : 95.65
 Cleveland: 41   3rd Qu.:106.00   Cleveland: 41   3rd Qu.:104.00
 Dallas   : 41   Max.   :136.00   Dallas   : 41   Max.   :132.00
 (Other)  :943                    (Other)  :943
     wochtag        datum
 Friday   :227   Min.   :1996-11-01
 Tuesday  :200   1st Qu.:1996-12-12
 Saturday :179   Median :1997-01-25
 Wednesday:170   Mean   :1997-01-25
 Sunday   :157   3rd Qu.:1997-03-11
 Thursday :155   Max.   :1997-04-20
 (Other)  :101
```

Es gibt keine NAs aber einen Fehler in den Wochentagen

```
> # es liegen keine NAs vor
> any(is.na(nba))

[1] FALSE

> # Die variablen der Vereine haben gleich viele Level
> (nlevels(nba$heimteam) == nlevels(nba$gastteam))

[1] TRUE

> # und die Laenge 29 (29 Vereine laut Beschreibung)
> nlevels(nba$heimteam) == 29

[1] TRUE

> # an welchen Wochentagn wurde gespielt?, str() zeigt 8 levels
> levels(nba$wochtag)

[1] "Friday"    "Monday"    "(OT)"      "Saturday"  "Sunday"    "Thursday"
[7] "Tuesday"   "Wednesday"

> # FEHLER IM DATENSATZ!!
>
> # Welche Faelle sind betroffen?
> nba[nba$wochtag=="(OT)",] # oder

     heimteam heimtore     gastteam gasttore wochtag      datum
801  Portland       95      NewYork       96    (OT) 1997-02-26
802 Vancouver       80 L.A.Clippers       83    (OT) 1997-02-26

> subset(nba, wochtag=="(OT)")

     heimteam heimtore     gastteam gasttore wochtag      datum
801  Portland       95      NewYork       96    (OT) 1997-02-26
802 Vancouver       80 L.A.Clippers       83    (OT) 1997-02-26
```

Man hat nun die Möglichkeit entweder die 2 Zeilen zu löschen oder den Fehler zu beheben. Da man den Wochentag leicht herausfinden kann ist 2. Option sinnvoller.

```
> # Welcher Wochentag war am 26.02.1997? Vielleicht schon im Datensatz vorhanden?
> subset(nba, datum=="1997-02-26", select= wochtag) # War ein Mittwoch

      wochtag
794 Wednesday
795 Wednesday
796 Wednesday
797 Wednesday
798 Wednesday
799 Wednesday
800 Wednesday
801      (OT)
802      (OT)
```

Man muss aufpassen, da `wochtag` vom Typ **factor** ist!

```
> # erster Versuch, direkte Zuweisung von Wednesday, wo wochtag fehlerhaft ist
> nba_c1 <- nba
> nba_c1[nba_c1$wochtag=="(OT)",]$wochtag <- "Wednesday"
> subset(nba_c1, wochtag=="(OT)")

[1] heimteam heimtore gastteam gasttore wochtag  datum
<0 rows> (or 0-length row.names)

> # nimmt es aber nicht aus den Levels raus
> levels(nba_c1$wochtag)

[1] "Friday"    "Monday"    "(OT)"      "Saturday"  "Sunday"    "Thursday"
[7] "Tuesday"   "Wednesday"

> # zweiter Versuch, Aenderung der Faktorenlevel
> nba_c2 <- nba
> levels(nba_c2$wochtag)[3]<-"Wednesday"
> subset(nba_c2, wochtag=="(OT)")

[1] heimteam heimtore gastteam gasttore wochtag  datum
<0 rows> (or 0-length row.names)

> levels(nba_c2$wochtag)

[1] "Friday"    "Monday"    "Wednesday" "Saturday"  "Sunday"    "Thursday"
[7] "Tuesday"

> # Haben wir etwas geaendert fuer die anderen Faelle?
> all.equal(nba[nba$wochtag!="(OT)",], nba_c2[nba$wochtag!="(OT)",])

[1] "Component 5: Attributes: < Component 2: Lengths (8, 7) differ (string compare on first 7) >
[2] "Component 5: Attributes: < Component 2: 1 string mismatch >"

> # es hat sich nichts geaendert, ausser die Laenge des Vektors der Faktorlevel
> # alles in Ordnung also, es kann die alte Variable ueberschrieben werden
>
> nba <- nba_c2
```

Export des Datensatzes in das kontinental-europäische `CSV`-Format, d.h. `';'` statt `','` als Trennzeichen für Werte und `','` statt `'.'` als Dezimaltrennzeichen.

```
> # Daten in .csv Format expotieren
> write.csv2(nba, file="./data/nba.csv",
+   row.names = FALSE)
```

Jetzt sollen die Daten in unterschiedlichen Formaten eingelesen werden.

Das SPSS-Format `.sav`:

```
> # Daten aus einer SPSS_Datei importieren
> library("foreign")
> nba2 <- read.spss("./data/nba.sav", to.data.frame = TRUE)
```

Wenn man das Argument `to.data.frame = TRUE` nicht spezifiziert, verwendet `read.spss()` die Voreinstellnung (engl. default) und liefert als Ergebnis ein Objekt vom Typ Liste zurück.

Jetzt lesen wir die zuvor gepeicherte `CSV`-Datei ein. Wie auch die meisten anderen Funktionen vom Typ `read.XXX` hat die Funktion ein Argument `colClasses` mit dem festgelegt werden kann wie R die entsprechenden Spalten beim Einlesen behandelt. Gerade beim einlesen größere Datensätze ist eine explizite Angabe zu empfehlen.

```
> # Daten aus einer CSV-Datei importieren mit Angabe des Spaltenformats
> nba3 <- read.csv2("./data/nba.csv",
+   colClasses = c("factor", "numeric", "factor", "numeric",
+   "factor", "Date"))
```

Einlesen einer Excel Datei im Format `.xlsx`:

```
> # Daten aus einer .xlsx-Datei importieren
> library("xlsx")
> nba4 <- read.xlsx2("./data/nba.xlsx", sheetIndex = 1,
+   colClasses = c("character", "numeric", "character", "numeric",
+     "character", "Date"))
```

`read.xlsx2`ist ein Beispiel, wo die Konvertierung nicht für all Datentypen funktioniert

```
> head(nba4) # Das Datum ist irgendwie falsch eingelesen/konvertiert worden.

  heimteam heimtore   gastteam gasttore  wochtag      datum
1  Atlanta       78    Detroit       90 Saturday 4531-05-28
2  Atlanta       87  Cleveland       83  Tuesday 4531-06-07
3  Atlanta       85      Miami       77   Friday 4531-06-10
4  Atlanta      101  Vancouver       80  Tuesday 4531-06-21
5  Atlanta      110 Washington       81   Friday 4531-06-24
6  Atlanta      105     Boston       95  Tuesday 4531-09-06

> # Umweg ueber Umwandlung nach numeric beim Einlesen
> # und danach separat nach Date
> nba4 <- read.xlsx2("./data/nba.xlsx", sheetIndex = 1,
+   colClasses = c("character", "numeric", "character", "numeric",
+     "character", "numeric"))
> head(nba4)

  heimteam heimtore   gastteam gasttore  wochtag  datum
1  Atlanta       78    Detroit       90 Saturday 961102
2  Atlanta       87  Cleveland       83  Tuesday 961112
3  Atlanta       85      Miami       77   Friday 961115
4  Atlanta      101  Vancouver       80  Tuesday 961126
5  Atlanta      110 Washington       81   Friday 961129
6  Atlanta      105     Boston       95  Tuesday 961203
```

```
> # Manuelles umsetzen auf Date (analog zu vorher)
> nba4 <- transform(nba4,
+   datum = as.Date(as.character(datum), format = "%y%m%d"))
> head(nba4)

  heimteam heimtore   gastteam gasttore  wochtag      datum
1  Atlanta       78    Detroit       90 Saturday 1996-11-02
2  Atlanta       87  Cleveland       83  Tuesday 1996-11-12
3  Atlanta       85      Miami       77   Friday 1996-11-15
4  Atlanta      101  Vancouver       80  Tuesday 1996-11-26
5  Atlanta      110 Washington       81   Friday 1996-11-29
6  Atlanta      105     Boston       95  Tuesday 1996-12-03
```

Einlesen einer Excel Datei im Format `.xls` funktioniert nur unter Windows bzw. MacOS (gegeben die entsprechenden Treiber sind installiert). Allgemeine Hinweise zum Lesen von Excel Tabellenblättern finden Sie unter `http://cran.r-project.org/doc/manuals/R-data.html#Reading-Excel-spreadsheets`. Für solche Fälle müssen die Datenblätter einzeln im Textformat gespeichert werden.

Generell ist beim Import von Daten, die nicht im Textformat sind, der indirekte Weg über das Zwischenspeichern in einer `CSV`-Datei zu empfehlen.

Da beim Einlesen der `sav`- und `xlsx`-Datei der Fehler im Wochentag noch drinnen steckt, muss er vor dem Vergleich noch ausgebessert werden.

```
> # Fehlerkorrektur wegen falschem Faktor
>   levels(nba2$wochtag)[levels(nba2$wochtag)=="(OT)"] <- "Wednesday"
>   levels(nba4$wochtag)[levels(nba4$wochtag)=="(OT)"] <- "Wednesday"
```

Sortieren der beiden zu vergleichenden Datensätze nach Datum und Heimmannschaft

```
> # Vergleich von Excel mit Original Datensatz
> # Da eine Mannschaft nur ein Spiel pro Tag haben kann,
> # Sortierung nach Datum und Heimmannschaft
> myOrder <- with(nba, order(datum, heimteam))
> nba <- nba[myOrder, ]
> myOrder4 <- with(nba4, order(datum, heimteam))
> nba4 <- nba4[myOrder4, ]
```

Offenbar gibt keinen Unterschied in den Werten sondern nur in den Attributen. Der zweite Ausdruck verzichtet auf den Vergleich der Attribute.

```
> all.equal(target = nba,
+           current = nba4)

[1] "Attributes: < Component 2: Mean relative difference: 0.6654624 >"

> # Laesst man nun die Attribute weg ...
> all.equal(target = nba,
+           current = nba4,
+           check.attributes = FALSE)

[1] TRUE
```

**Aufgabe 2:**

Überprüfen Sie, ob jede Mannschaft die gleiche Anzahl von Heim- und Auswärtsspielen absolviert hat!

**Lösung:**

```
> with(nba, table(heimteam))

heimteam
     Atlanta       Boston    Charlotte      Chicago    Cleveland       Dallas
          41           41           41           41           41           41
      Denver      Detroit  GoldenState      Houston      Indiana L.A.Clippers
          41           41           41           41           41           41
  L.A.Lakers        Miami    Milwaukee    Minnesota    NewJersey      NewYork
          41           41           41           41           41           41
     Orlando Philadelphia      Phoenix     Portland   Sacramento   SanAntonio
          41           41           41           41           41           41
     Seattle      Toronto         Utah    Vancouver   Washington
          41           41           41           41           41

> with(nba, table(gastteam))

gastteam
     Atlanta       Boston    Charlotte      Chicago    Cleveland       Dallas
          41           41           41           41           41           41
      Denver      Detroit  GoldenState      Houston      Indiana L.A.Clippers
          41           41           41           41           41           41
  L.A.Lakers        Miami    Milwaukee    Minnesota    NewJersey      NewYork
          41           41           41           41           41           41
     Orlando Philadelphia      Phoenix     Portland   Sacramento   SanAntonio
          41           41           41           41           41           41
     Seattle      Toronto         Utah    Vancouver   Washington
          41           41           41           41           41
```

**Aufgabe 3:**

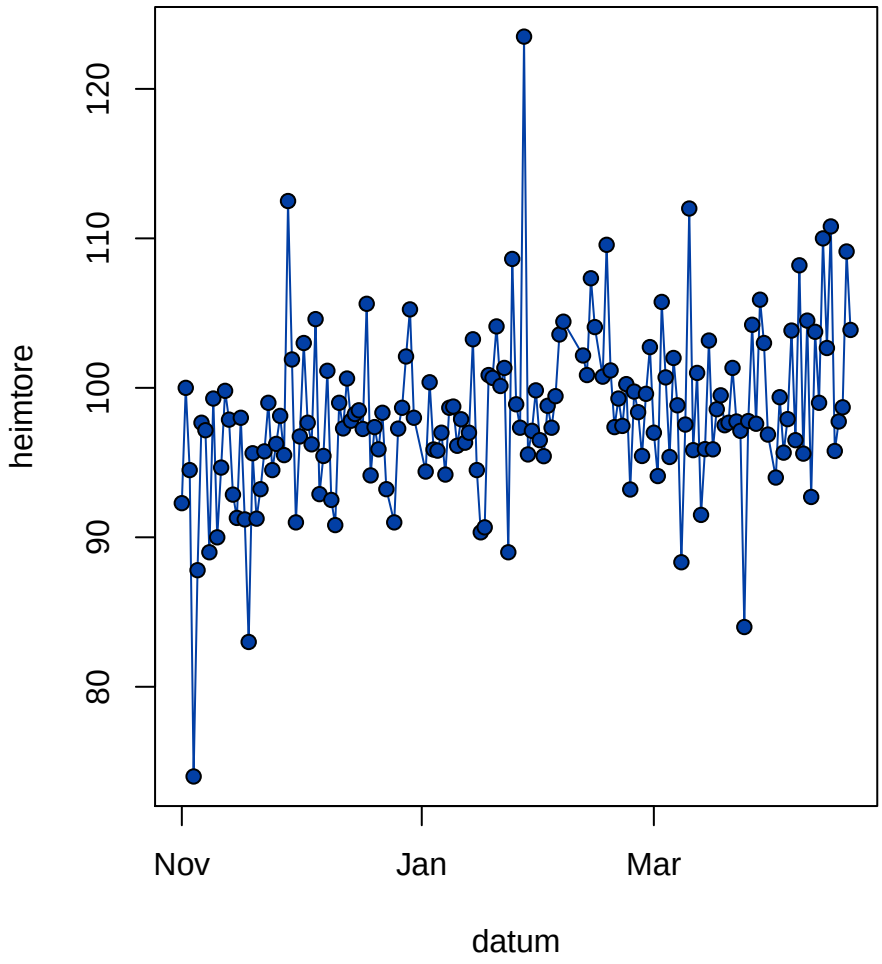
Stellen Sie fest, ob es einen zeitlichen Trend für die mittlere Anzahl der Heim- bzw. Auswärtspunkte gibt!

**Lösung:**

```
> mhtrunde <- aggregate(heimtore ~ datum, FUN = mean, data=nba)
> mgtrunde <- aggregate(gasttore ~ datum, FUN = mean, data=nba)
> # wann wurde gespielt?
> range(mhtrunde$datum)

[1] "1996-11-01" "1997-04-20"

> library("colorspace")
> torartcol <- diverge_hcl(2)
> # einfache Grafik
> plot(mhtrunde, type="l", col=torartcol[1])
> # Punkte einzeichnen, damit man etwas sieht.
> points(mhtrunde, bg=torartcol[1], pch=21)
> plot(mgtrunde, type="l", col=torartcol[2])
> # Punkte einzeichnen, damit man etwas sieht.
> points(mgtrunde, bg=torartcol[2], pch=21)
> # Durch Konfiguration wird schoener
> plot(mhtrunde, type = "n", ann = FALSE, ylim = c(70, 130), xlim =
+   as.Date(c("1996-11-01","1997-05-01")))
> lines(mhtrunde, col=torartcol[1])
> points(mhtrunde, bg=torartcol[1], pch=21)
> title(main = "Mittlere Anzahl von Punkten pro Spieltag", ylab =
+   "Mittlere Anzahl Punkten", xlab = "Spieltag")
> plot(mgtrunde, type = "n", ann = FALSE, ylim = c(70, 130), xlim =
+   as.Date(c("1996-11-01","1997-05-01")))
> lines(mgtrunde, col = torartcol[2])
> points(mgtrunde, bg = torartcol[2], pch = 21)
> title(main = "Mittlere Anzahl von Gastpunkten pro Spieltag", ylab =
+   "Mittlere Anzahl Gastpunkten", xlab = "Spieltag")
```


heimtore
80
90
100
110
120
Nov
Jan
Mar
datum

**Aufgabe 4:**

Welche Mannschaften haben mehr als 4200 Heimpunkte erzielt? Welche Mannschaften haben mehr als 4200 Auswärtspunkte geholt? Wie haben diese Mannschaften im direkten Vergleich gespielt?

**Lösung:**

```
> shtteam <- aggregate(heimtore ~ heimteam, FUN = sum, data = nba)
> A <- subset(shtteam, subset = heimtore > 4200, select = heimteam)
> A

   heimteam
4   Chicago
10  Houston
21  Phoenix
27     Utah

> sgtteam <- aggregate(gasttore ~ gastteam, FUN = sum, data = nba)
> B <- subset(sgtteam, subset = gasttore > 4200, select = gastteam)
> B

   gastteam
27     Utah

> subset(nba, subset = ((heimteam %in% A$heimteam | heimteam %in%
+   B$gastteam) & (gastteam %in% B$gastteam | gastteam %in% A$heimteam)))

     heimteam heimtore gastteam gasttore   wochtag      datum
22    Phoenix       95  Houston      110  Saturday 1996-11-02
29       Utah       72  Houston       75    Monday 1996-11-04
68    Houston       91     Utah       85  Saturday 1996-11-09
76    Chicago       97  Phoenix       79    Monday 1996-11-11
143   Phoenix       99  Chicago      113 Wednesday 1996-11-20
151   Houston      115  Phoenix      105  Thursday 1996-11-21
166      Utah      105  Chicago      100  Saturday 1996-11-23
295      Utah       87  Phoenix       95  Thursday 1996-12-12
454   Chicago      102     Utah       89    Monday 1997-01-06
494   Chicago      110  Houston       86  Saturday 1997-01-11
527      Utah       95  Phoenix       91  Thursday 1997-01-16
547   Houston      102  Chicago       86    Sunday 1997-01-19
572   Phoenix       99     Utah      111 Wednesday 1997-01-22
597   Houston      100     Utah      105  Saturday 1997-01-25
978   Houston       99  Phoenix      104  Saturday 1997-03-22
1053  Phoenix      109  Houston       96 Wednesday 1997-04-02
1120     Utah      104  Houston       83    Friday 1997-04-11
1146  Phoenix      122     Utah      127   Tuesday 1997-04-15
```

**Aufgabe 5:**

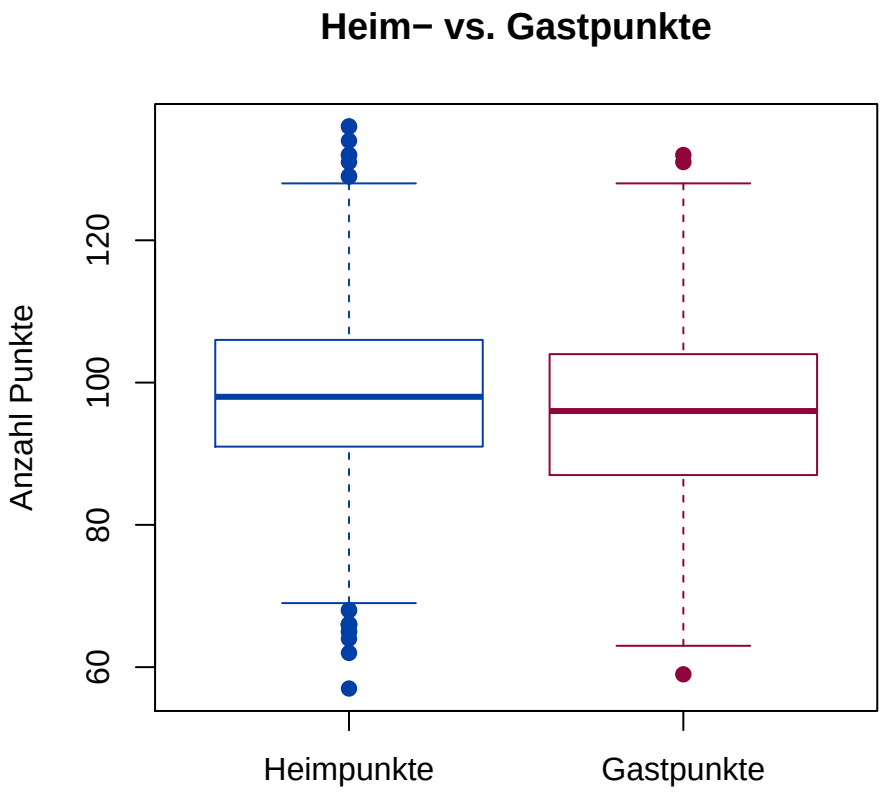
Erstellen Sie Boxplots für Anzahl der Punkte gruppiert nach Heim- bzw. Gastpunkten. Berücksichtigen Sie in einem weiteren Plot zusätzlich den Wochentag. (Achtung: Die Daten müssen zunächst umstrukturiert werden!)

**Lösung:**

```
> library("reshape")
> nba_long <- melt(nba,
+              measure.vars = c("gasttore", "heimtore"),
+              variable_name = "Punktart")
> str(nba_long)

'data.frame':        2378 obs. of  6 variables:
$ heimteam: Factor w/ 29 levels "Atlanta","Boston",..: 2 7 8 9 10 13 14 16 17
   19 ...
$ gastteam: Factor w/ 29 levels "Atlanta","Boston",..: 4 6 11 12 23 21 1 24 5
   29 ...
$ wochtag : Factor w/ 7 levels "Friday","Monday",..: 1 1 1 1 1 1 1 1 1 ...
$ datum : Date, format: "1996-11-01" "1996-11-01" ...
$ Punktart: Factor w/ 2 levels "gasttore","heimtore": 1 1 1 1 1 1 1 1 1 ...
$ value : int 107 92 89 97 85 82 81 78 90 96 ...

> ## Bei der Variablen 'Punktart' soll das level 'heimtore' die erste
> ## Kategorie sein!
> nba_long <- transform(nba_long, Punktart = relevel(Punktart, ref =
+   "heimtore"))
> boxplot(value ~ Punktart, data = nba_long, main = "Heim- vs. Gastpunkte",
+   ylab = "Anzahl Punkte", names = c("Heimpunkte", "Gastpunkte"), border =
+   torartcol, pars = list(outbg = torartcol, outpch = 21))
```



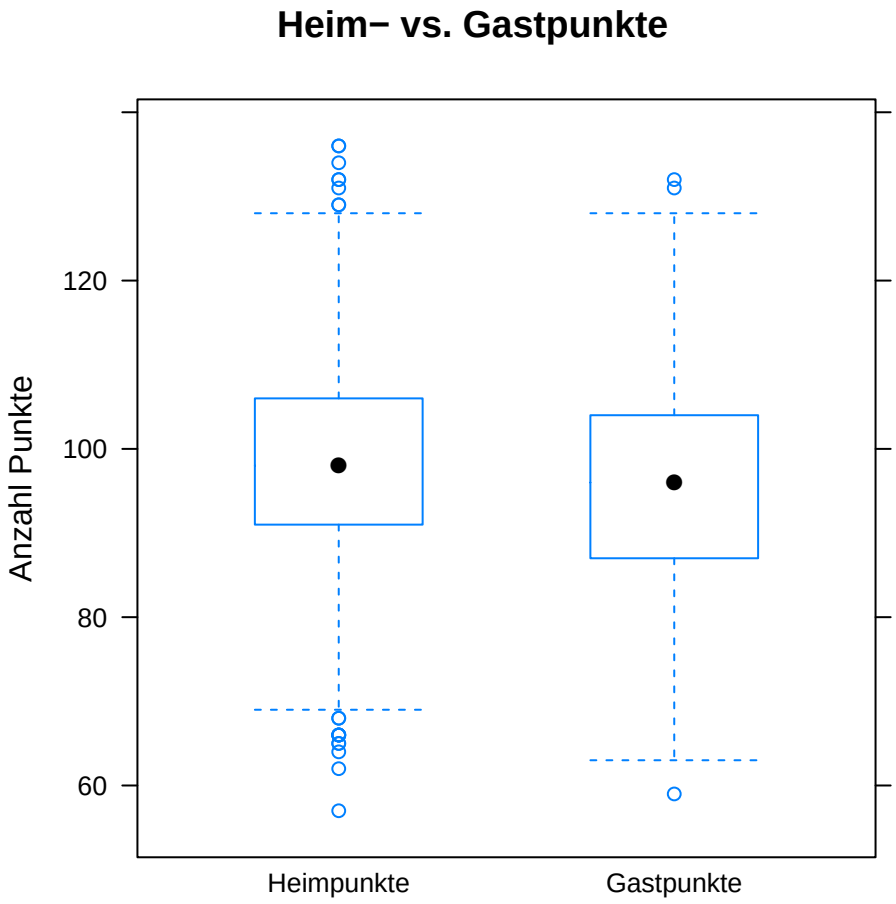

Nach dem Umstrukturieren der Daten hätten wir uns auch mit Aufgabe 2 leichter getan:

```
> xtabs(~ heimteam + Punktart, data =  nba_long)
```

```
              Punktart
heimteam       heimtore gasttore
  Atlanta             41       41
  Boston              41       41
  Charlotte           41       41
  Chicago             41       41
  Cleveland           41       41
  Dallas              41       41
  Denver              41       41
  Detroit             41       41
  GoldenState         41       41
  Houston             41       41
  Indiana             41       41
  L.A.Clippers        41       41
  L.A.Lakers          41       41
  Miami               41       41
  Milwaukee           41       41
  Minnesota           41       41
  NewJersey           41       41
  NewYork             41       41
  Orlando             41       41
  Philadelphia        41       41
  Phoenix             41       41
  Portland            41       41
  Sacramento          41       41
  SanAntonio          41       41
```

```
  Seattle             41       41
  Toronto             41       41
  Utah                41       41
  Vancouver           41       41
  Washington          41       41
```

```
> library("lattice")
> bwplot(value ~ Punktart, data=nba_long, main = "Heim- vs. Gastpunkte",
+   ylab = "Anzahl Tore", xlim = c("Heimpunkte", "Gastpunkte"))
```




```
> bwplot(value ~ Punktart | wochtag, data=nba_long, main =
+   "Heim- vs. Gastpunkte nach Wochentag", ylab = "Anzahl Punkte",
+   xlim = c("Heimpunkte", "Gastpunkte"))
```

## Heim− vs. Gastpunkte nach Wochentag