

Aufgabe 1:

Der `cd4` Datensatz umfasst 2376 Beobachtungen der Zahl der CD4-Zellen im Blut von 369 HIV-infizierten/AIDS-kranken Männern vor und nach dem Zeitpunkt, zu dem zum ersten Mal HIV-Antikörper in ihrem Blut nachgewiesen wurden (Seroconversion). Die Zahl der CD4 -zellen dient dabei als Biomarker für den Zustand des Immunsystems. Das Hauptkenntnisinteresse liegt darin, die Form des Verlaufes des CD4-Gehalts über die Zeit zu bestimmen.

Lesen Sie die Daten ein und wandeln Sie `drug` und `ID` in Faktorvariablen um:

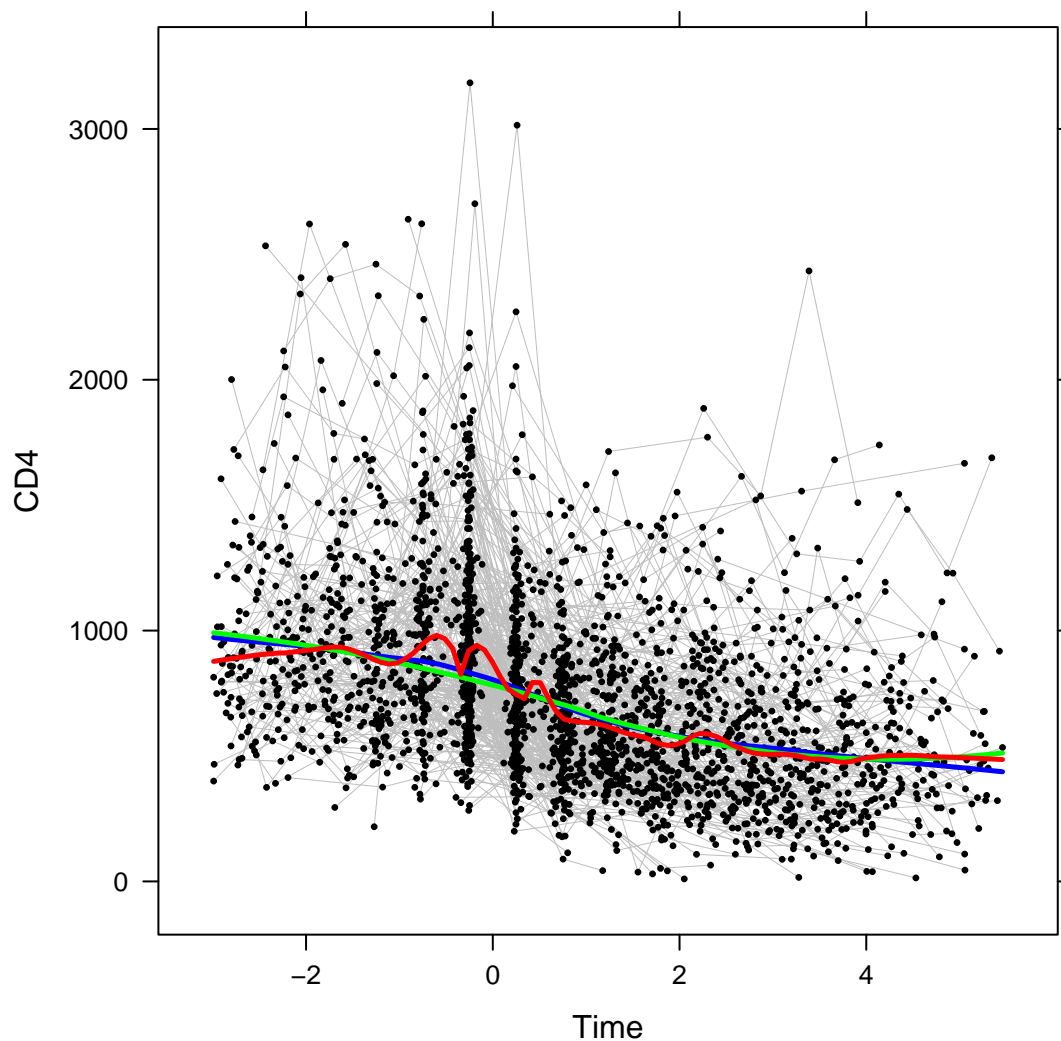
```
R>url <- "http://www.statistik.lmu.de/institut/lehrstuhl/semwiso/longitudinal_ss08/"
R>cd4 <- read.table(paste(url, "download/cd4_Diggle.txt", sep = ""),
+   header = T)
R>cd4$drug <- factor(cd4$drug, labels = c("yes", "no"))
R>cd4$ID <- factor(cd4$ID)
```

(a) Benutzen Sie den folgenden Code um die Verläufe der einzelnen Patienten zu plotten

```
R>library("lattice")
R>xyplot(CD4 ~ Time, data = cd4, group = ID, panel = function(x,
+   y, ...) {
+   panel.superpose(x, y, ..., col = "grey", type = "l", lwd = 0.1)
+   panel.xyplot(x, y, col = "black", type = "p", pch = 19, cex = 0.3)
+   panel.loess(x, y, lwd = 2.5, col = "blue")
+   panel.loess(x, y, lwd = 2.5, span = 1, degree = 2, col = "green")
+   panel.loess(x, y, lwd = 2.5, span = 0.1, evaluation = 100,
+     col = "red")
+ })
```

- (i) Was kontrollieren die `panel.loess`-Parameter `span` und `degree`? Was stellt also die grüne Linie dar? Überlegen sie sich wie die Schätzwerte zustande kommen. Wie ändert sich das Schätzverfahren wenn `span` nicht 1 ist?
- (ii) Untersuchen Sie wie sensibel die geglättete Kurve bezüglich Änderungen dieser beiden Parameter ist. Für sehr kleine Werte von `span` sollten sie `evaluation` entsprechend erhöhen - warum?

Lösung:



- (i) zu Parametern s. `loess`-Hilfe. Die grüne Linie stellt eine lokal gewichtete quadratische (`degree=2`) Regression mit allen Daten (`span=1`) dar, bei der die Daten mit der in der Hilfe-Datei angegebenen bikubischen Gewichtung eingehen. D.h.

$$\hat{f}(z) = \hat{\gamma}(z)_0 + \hat{\gamma}(z)_1 z + \hat{\gamma}(z)_2 z^2$$

$$\hat{\gamma}(z) = (\mathbf{X}'\mathbf{W}(z)^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}(z)\mathbf{y}$$

$$\mathbf{X} = (\mathbf{1}_n \quad \mathbf{x} \quad \mathbf{x}^2); \quad \mathbf{x} = (x_1, \dots, x_n)'$$

$$\mathbf{W}(z) = \begin{pmatrix} w(x_1, z) & & \\ & \ddots & \\ & & w(x_n, z) \end{pmatrix}$$

$$w(x_j, z) = \left(1 - \left(\frac{|x_j - z|}{d_{\max}(z)}\right)^3\right)^3$$

$$d_{\max}(z) = \max_{k=1, \dots, n} (|x_k - z|)$$

Wenn `span` nicht 1 ist, umfasst die Designmatrix nicht den gesamten Vektor x , sondern nur einen Teilvektor, der die Werte umfasst die am nächsten bei z liegen. Wenn z.B. $(x_{[1]}, \dots, x_{[100]})$ der aufsteigend geordnete Vektor ist, `span= .1` und $x_{[10]} < z < x_{[11]}$ dann geht nur die Teilvektoren $(x_{[6]}, \dots, x_{[15]})'$ und die dazugehörigen Werte von \mathbf{y} in die Schätzung $\hat{\gamma}(z)$ ein.

(ii) Zur Darstellung des Verlaufs wird die Kurve an **evaluation** äquidistanten Punkten ausgewertet (s. `?loess`) und der Verlauf zwischen 2 Punkten linear interpoliert (also man zieht halt Linien zwischen den Punkten...). Wenn die Kurve sehr unruhig ist weil `span` sehr klein ist werden bei zu wenig ausgewerteten Punkten nicht alle Buckel und Knicke der geglätteten Kurve abgebildet.

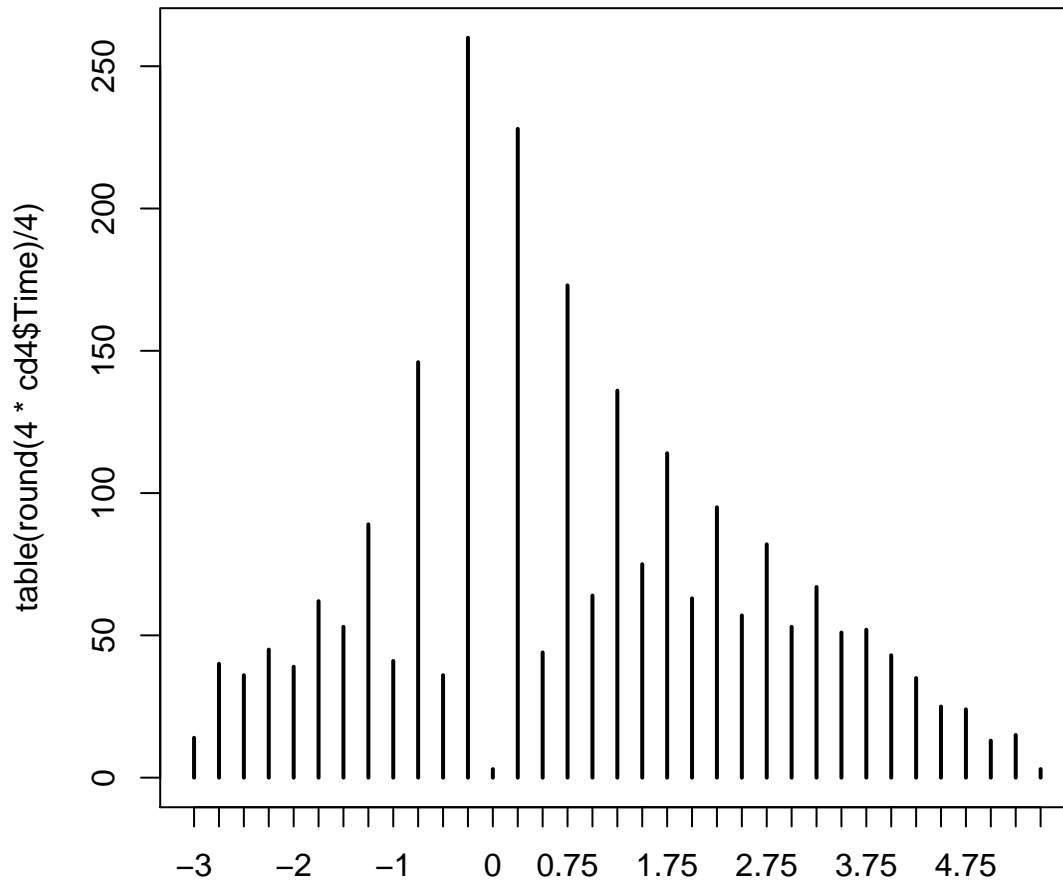
(b) Wir vergrößern die Zeitmessung um uns einen besseren Überblick über die Struktur im Datensatz zu verschaffen. Mit dem folgenden Code legen wir eine neue Variable `Time2` an, die `Time` in bestimmte vordefinierten Zeitintervalle kategorisiert:

```
R>plot(table(round(4 * cd4$Time)/4))
R>timepoints <- c(floor(min(cd4$Time)), -1.25, -0.75, -0.25, 0,
+ 0.25, 0.75, 1.25, 1.75, 2.25, 2.75, 3.25, ceiling(max(cd4$Time)))
R>cd4$Time2 <- cut(cd4$Time, breaks = timepoints)
R>(table(cd4$Time2))
```

Benutzen Sie die R-Hilfe um den obigen Code nachzuvollziehen. Überlegen Sie warum die Intervallgrenzen anders als in der Vorlesung gewählt wurden.

Lösung:

<code>(-3,-1.25]</code>	<code>(-1.25,-0.75]</code>	<code>(-0.75,-0.25]</code>	<code>(-0.25,0]</code>	<code>(0,0.25]</code>
325	164	258	116	109
<code>(0.25,0.75]</code>	<code>(0.75,1.25]</code>	<code>(1.25,1.75]</code>	<code>(1.75,2.25]</code>	<code>(2.25,2.75]</code>
247	225	192	166	151
<code>(2.75,3.25]</code>	<code>(3.25,6]</code>			
135	288			



Andere Intervallgrenzen um Struktur der Daten besser abzubilden - Intervalle so dass sie die Häufungen der Daten auf der Zeitachse abbilden. Zusätzliche Begründung s. c)(i) Nachteil: Nicht äquidistant.

- (c) Wir wollen die Korrelationen der CD4-Werte ähnlich wie auf Folie 2.23 darstellen. Dazu muss der Datensatz umstrukturiert werden:

```
R>m <- lm(CD4 ~ Time, data = cd4)
R>cd4$eps <- resid(m)
R>o <- order(cd4$Time)
R>cd4.kat <- reshape(cd4[o, ], v.names = c("CD4", "eps"), timevar = "Time2",
+   idvar = "ID", direction = "wide", drop = c("Age", "Time",
+   "cigpacks", "drug", "sexpartners", "cesd"))
R>colnames(cd4.kat)[seq(2, 24, by = 2)] <- levels(cd4$Time2)
R>colnames(cd4.kat)[seq(3, 25, by = 2)] <- paste("e.", levels(cd4$Time2),
+   sep = "")
```

- (i) Vergleichen Sie die Einträge für ID "10005" in den beiden Datensätzen. Was ist problematisch?
- (ii) Benutzen sie `spiom()` um `cd4.kat` grafisch darzustellen.

- (iii) Berechnen Sie die Korrelationen zwischen den Messungen in den verschiedenen Intervallen. Berechnen Sie die Korrelationen zwischen den Residuen einer Regression mit linearem Zeittrend in den verschiedenen Intervallen.

Lösung:

(i) `R>cd4[cd4$ID == "10005",]`

	Time	CD4	Age	cigpacks	drug	sexpartners	cesd	ID	Time2	eps
4	-2.729637	464	6.95	0	no	5	4	10005	(-3,-1.25]	-617.6265
5	-2.250513	845	6.95	0	no	5	-4	10005	(-3,-1.25]	-194.0077
6	-0.221766	752	6.95	0	no	5	-5	10005	(-0.25,0]	-106.5474
7	0.221766	459	6.95	0	no	5	2	10005	(0,0.25]	-360.0945
8	0.774812	181	6.95	0	no	5	-3	10005	(0.75,1.25]	-588.9002
9	1.256673	434	6.95	0	no	5	-7	10005	(1.25,1.75]	-293.0379

`R>cd4.kat[cd4.kat$ID == "10005",]`

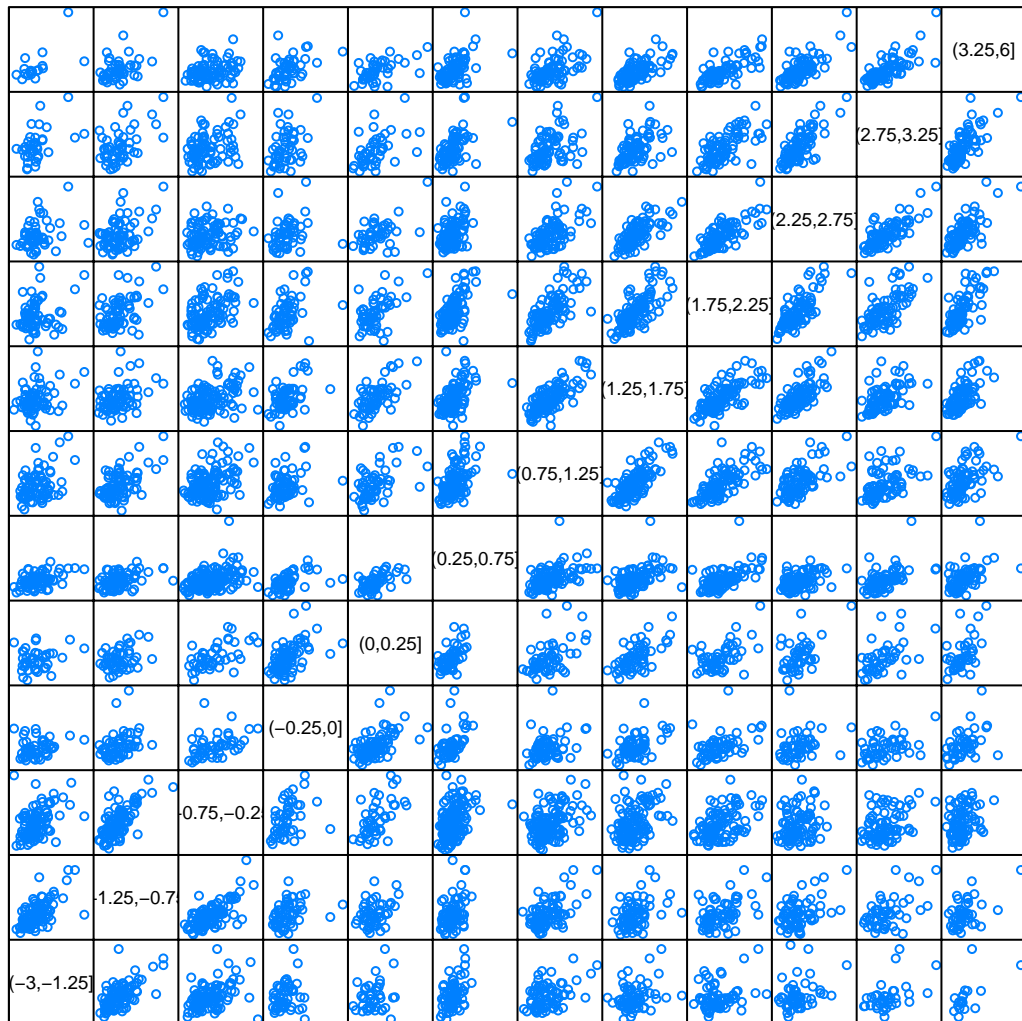
	ID	(-3,-1.25]	e.(-3,-1.25]	(-1.25,-0.75]	e.(-1.25,-0.75]	(-0.75,-0.25]	e.(-0.75,-0.25]	(-0.25,0]	e.(-0.25,0]	(0,0.25]	e.(0,0.25]	(0.25,0.75]	e.(0.25,0.75]	(0.75,1.25]	e.(0.75,1.25]	(1.25,1.75]	e.(1.25,1.75]	(1.75,2.25]	e.(1.75,2.25]	(2.25,2.75]	e.(2.25,2.75]	(2.75,3.25]	e.(2.75,3.25]	(3.25,6]	e.(3.25,6]
4	10005	464	-617.6265	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4		NA	752	-106.5474	459	-360.0945	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4		NA	181	-588.9002	434	-293.0379	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4		NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4		NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

⇒ Falls für ein Subjekt mehrere Messungen im selben Intervall vorliegen wird nur die erste davon in den “wide”-Datensatz übernommen. Ein Grund mehr feinere Intervalle zu benutzen (s.a. b)).

(ii,iii) `R>trellis.par.set(list(add.text = list(cex = 0.6)))`
`R>print(splom(cd4.kat[, seq(2, 24, by = 2)], pscale = 0, cex = 0.5))`
`R>round(cor(cd4.kat[, seq(2, 24, by = 2)], use = "pairwise"), 2)`

	(-3,-1.25]	(-1.25,-0.75]	(-0.75,-0.25]	(-0.25,0]	(0,0.25]
(-3,-1.25]	1.00	0.65	0.40	0.18	0.22
(-1.25,-0.75]	0.65	1.00	0.69	0.36	0.35
(-0.75,-0.25]	0.40	0.69	1.00	0.42	0.59
(-0.25,0]	0.18	0.36	0.42	1.00	0.55
(0,0.25]	0.22	0.35	0.59	0.55	1.00
(0.25,0.75]	0.47	0.44	0.40	0.49	0.58
(0.75,1.25]	0.42	0.58	0.45	0.33	0.55
(1.25,1.75]	0.32	0.41	0.42	0.50	0.66
(1.75,2.25]	0.24	0.50	0.49	0.44	0.54
(2.25,2.75]	0.30	0.45	0.22	0.23	0.54
(2.75,3.25]	0.49	0.54	0.36	0.45	0.51
(3.25,6]	0.61	0.54	0.31	0.48	0.57
	(0.25,0.75]	(0.75,1.25]	(1.25,1.75]	(1.75,2.25]	(2.25,2.75]
(-3,-1.25]	0.47	0.42	0.32	0.24	0.30
(-1.25,-0.75]	0.44	0.58	0.41	0.50	0.45
(-0.75,-0.25]	0.40	0.45	0.42	0.49	0.22
(-0.25,0]	0.49	0.33	0.50	0.44	0.23
(0,0.25]	0.58	0.55	0.66	0.54	0.54

(0.25,0.75]	1.00	0.52	0.52	0.59	0.42
(0.75,1.25]	0.52	1.00	0.78	0.73	0.62
(1.25,1.75]	0.52	0.78	1.00	0.74	0.73
(1.75,2.25]	0.59	0.73	0.74	1.00	0.77
(2.25,2.75]	0.42	0.62	0.73	0.77	1.00
(2.75,3.25]	0.54	0.51	0.53	0.68	0.77
(3.25,6]	0.54	0.59	0.66	0.74	0.75
	(2.75,3.25]	(3.25,6]			
(-3,-1.25]	0.49	0.61			
(-1.25,-0.75]	0.54	0.54			
(-0.75,-0.25]	0.36	0.31			
(-0.25,0]	0.45	0.48			
(0,0.25]	0.51	0.57			
(0.25,0.75]	0.54	0.54			
(0.75,1.25]	0.51	0.59			
(1.25,1.75]	0.53	0.66			
(1.75,2.25]	0.68	0.74			
(2.25,2.75]	0.77	0.75			
(2.75,3.25]	1.00	0.76			
(3.25,6]	0.76	1.00			

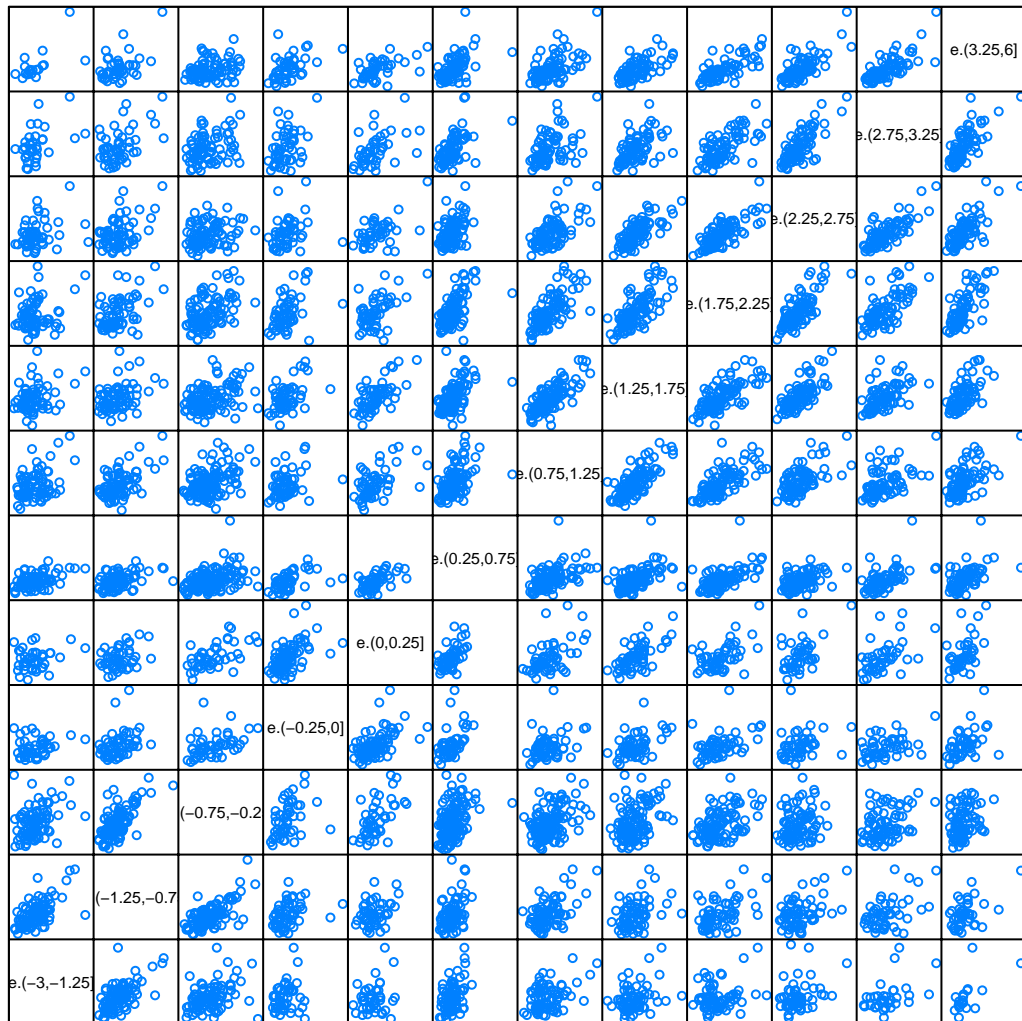


Scatter Plot Matrix

```
R>trellis.par.set(list(add.text = list(cex = 0.5)))
R>print(splom(cd4.kat[, seq(3, 25, by = 2)], pscale = 0, cex = 0.5))
```

```
R>round(cor(cd4.kat[, seq(3, 25, by = 2)], use = "pairwise"), 2)
```

	e. (-3,-1.25]	e. (-1.25,-0.75]	e. (-0.75,-0.25]	e. (-0.25,0]
e. (-3,-1.25]	1.00	0.64	0.38	0.20
e. (-1.25,-0.75]	0.64	1.00	0.69	0.36
e. (-0.75,-0.25]	0.38	0.69	1.00	0.42
e. (-0.25,0]	0.20	0.36	0.42	1.00
e. (0,0.25]	0.23	0.36	0.59	0.55
e. (0.25,0.75]	0.45	0.45	0.41	0.50
e. (0.75,1.25]	0.41	0.57	0.45	0.34
e. (1.25,1.75]	0.31	0.40	0.42	0.49
e. (1.75,2.25]	0.23	0.49	0.48	0.44
e. (2.25,2.75]	0.29	0.44	0.22	0.23
e. (2.75,3.25]	0.47	0.52	0.35	0.45
e. (3.25,6]	0.62	0.52	0.29	0.49
	e. (0,0.25]	e. (0.25,0.75]	e. (0.75,1.25]	e. (1.25,1.75]
e. (-3,-1.25]	0.23	0.45	0.41	0.31
e. (-1.25,-0.75]	0.36	0.45	0.57	0.40
e. (-0.75,-0.25]	0.59	0.41	0.45	0.42
e. (-0.25,0]	0.55	0.50	0.34	0.49
e. (0,0.25]	1.00	0.59	0.55	0.66
e. (0.25,0.75]	0.59	1.00	0.52	0.53
e. (0.75,1.25]	0.55	0.52	1.00	0.78
e. (1.25,1.75]	0.66	0.53	0.78	1.00
e. (1.75,2.25]	0.54	0.60	0.73	0.74
e. (2.25,2.75]	0.55	0.44	0.63	0.74
e. (2.75,3.25]	0.51	0.56	0.50	0.53
e. (3.25,6]	0.57	0.55	0.59	0.66
	e. (1.75,2.25]	e. (2.25,2.75]	e. (2.75,3.25]	e. (3.25,6]
e. (-3,-1.25]	0.23	0.29	0.47	0.62
e. (-1.25,-0.75]	0.49	0.44	0.52	0.52
e. (-0.75,-0.25]	0.48	0.22	0.35	0.29
e. (-0.25,0]	0.44	0.23	0.45	0.49
e. (0,0.25]	0.54	0.55	0.51	0.57
e. (0.25,0.75]	0.60	0.44	0.56	0.55
e. (0.75,1.25]	0.73	0.63	0.50	0.59
e. (1.25,1.75]	0.74	0.74	0.53	0.66
e. (1.75,2.25]	1.00	0.77	0.68	0.73
e. (2.25,2.75]	0.77	1.00	0.78	0.75
e. (2.75,3.25]	0.68	0.78	1.00	0.75
e. (3.25,6]	0.73	0.75	0.75	1.00



Scatter Plot Matrix

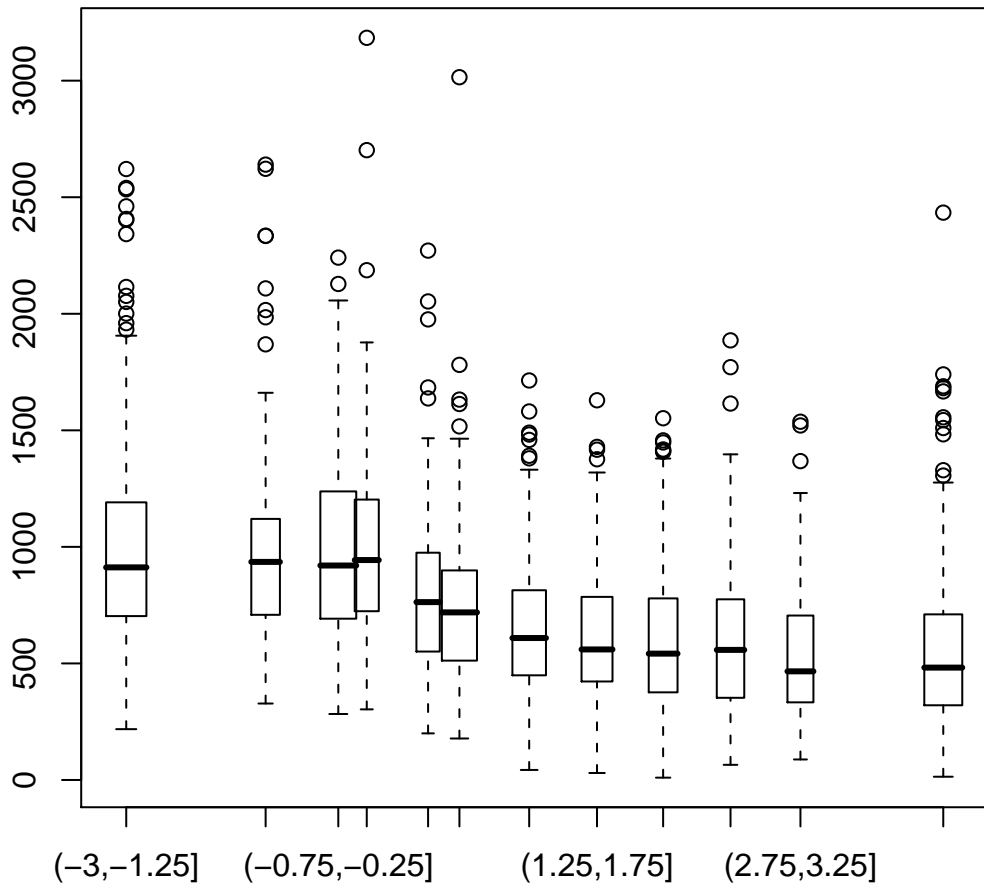
Da in den Daten kein starker (linearer) zeitlicher Trend vorliegt macht es hier wenig Unterschied wenn man die beob. Werte statt der Residuen betrachtet.

- (d) Stellen sie auf Basis der vergrößerten Zeitpunkte den Verlauf der Messungen und der Varianz der Messungen über die Zeit dar. Benutzen sie `tapply()` um die Varianzen in den Intervallen zu bestimmen. Warum ist es hier besser den Datensatz im "long"-Format zu benutzen?

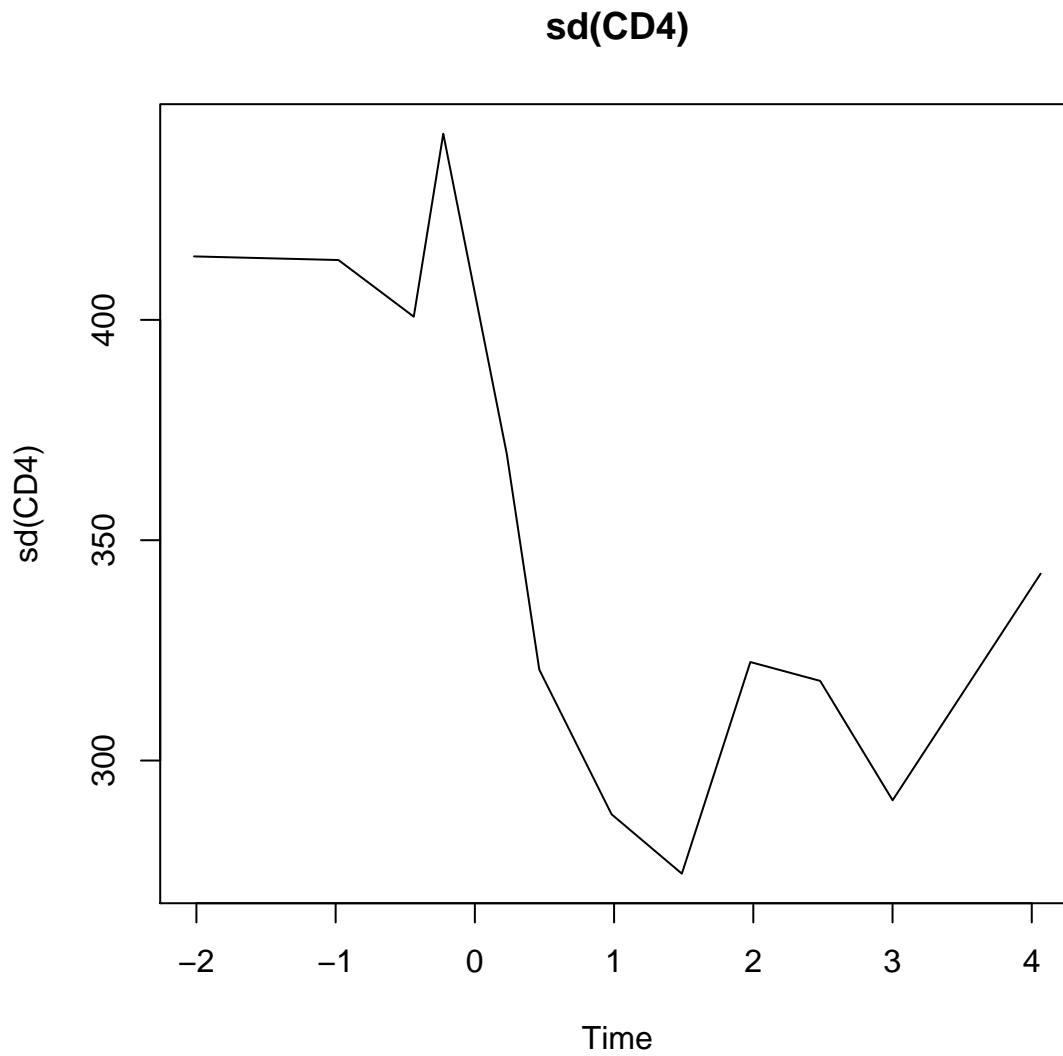
Lösung:

```
R>int.means <- with(cd4, tapply(Time, Time2, mean))
R>boxplot(CD4 ~ Time2, data = cd4, at = int.means, varwidth = T,
+        xlim = c(-2.1, 4.2), boxwex = 0.3, axis.cex = 0.5, main = "CD4")
```

CD4



```
R>var.time2 <- with(cd4, tapply(CD4, Time2, var))
R>plot(int.means, sqrt(var.time2), type = "l", main = "sd(CD4)",
+      xlab = "Time", ylab = "sd(CD4)")
```



Es ist hier besser den Datensatz im “long”-Format zu benutzen, weil dieser tatsächlich ALLE Beobachtungen enthält, siehe c)(i).