visualizations-v3

March 27, 2020

1 Visualizations, version 3

1.1 Setup packages.

[1]:
```
require(data.table)
require(magrittr)

require(ggplot2)
require(GGally)
```

```
Loading required package: data.table
Loading required package: magrittr
Loading required package: ggplot2
Loading required package: GGally
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

1.2 Read data tables.

1.2.1 2D dataset.

[2]:
```
xs2d <- fread("xs-2d-20200322a.csv")[, case:=factor(case)]
xs2d %>% dim
```

1. 121 2. 3

[3]:
```
ys2d <- fread("ys-2d-20200322a.csv")[, case:=factor(case)]
ys2d %>% dim
```

1. 2541 2. 4

[4]:
```
s2d = merge(xs2d, ys2d)
s2d %>% head
```

A data.table: 6 x 6

case <fct>	x1 <dbl>	x2 <dbl>	t <dbl>	y1 <dbl>	y2 <dbl>
969	0	0	0.0	-0.19003198	0.5144967
969	0	0	0.5	-0.14198226	0.5555341
969	0	0	1.0	-0.08952215	0.5992292
969	0	0	1.5	-0.03175619	0.6456396
969	0	0	2.0	0.03232090	0.6947990
969	0	0	2.5	0.10383499	0.7467108

1.2.2 3D dataset.

[5]:
```
xs3d <- fread("xs-3d-20200322a.csv")[, case:=factor(case)]
xs3d %>% dim
```

1. 1331 2. 4

[6]:
```
ys3d <- fread("ys-3d-20200322a.csv")[, case:=factor(case)]
ys3d %>% dim
```

1. 27951 2. 5

[7]:
```
s3d = merge(xs3d, ys3d)
s3d %>% head
```

A data.table: 6 x 8

case <fct>	x1 <dbl>	x2 <dbl>	x3 <dbl>	t <dbl>	y1 <dbl>	y2 <dbl>	y3 <dbl>
1	0	0	0	0.0	-0.190032	0.5144967	0.4093612
1	0	0	0	0.5	-0.190032	0.5144967	0.4093612
1	0	0	0	1.0	-0.190032	0.5144967	0.4093612
1	0	0	0	1.5	-0.190032	0.5144967	0.4093612
1	0	0	0	2.0	-0.190032	0.5144967	0.4093612
1	0	0	0	2.5	-0.190032	0.5144967	0.4093612

1.3 Create functions for jittering to avoid overplotting.

[8]:
```
jitter2d <- function(df)
    df[, .(
        case,
        x1=jitter(x1, factor=1),
        x2=jitter(x2, factor=1),
        t,
        y1=jitter(y1, factor=1),
        y2=jitter(y2, factor=1)
    )]
```

[9]:
```
jitter3d <- function(df)
    df[, .(
        case,
        x1=jitter(x1, factor=1),
        x2=jitter(x2, factor=1),
        x3=jitter(x3, factor=1),
```

```
        t,
        y1=jitter(y1, factor=1),
        y2=jitter(y2, factor=1),
        y3=jitter(y3, factor=1)
    )]
```

1.4 Subsets of levels.

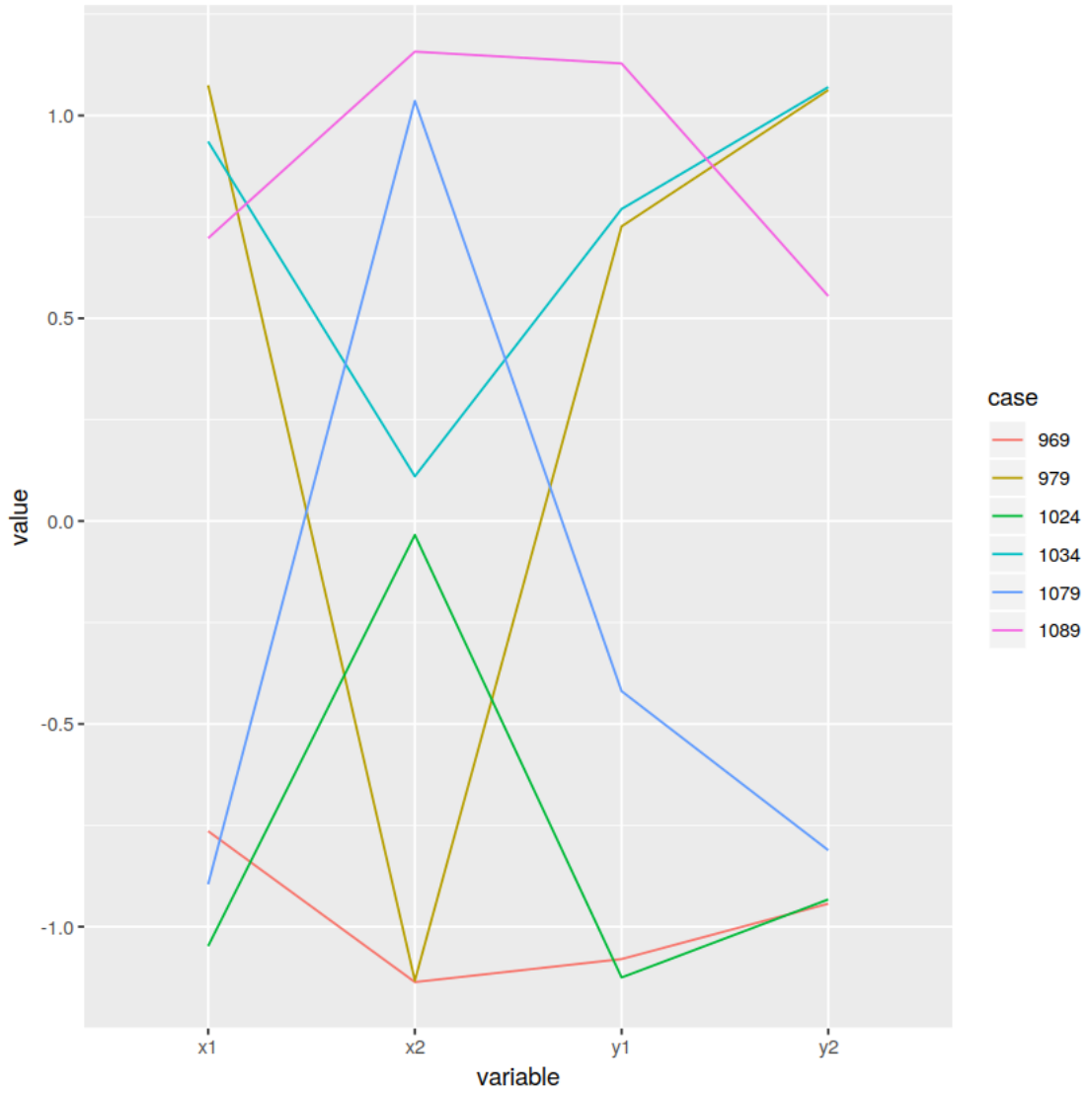
[10]:
```
levels2 <- c(0, 1)
levels3 <- c(0, 0.5, 1)
levels4 <- c(0, 0.3, 0.7, 1)
levels5 <- c(0, 0.2, 0.5, 0.8, 1)
```

1.5 Parallel coordinates plots.

1.5.1 2D dataset.

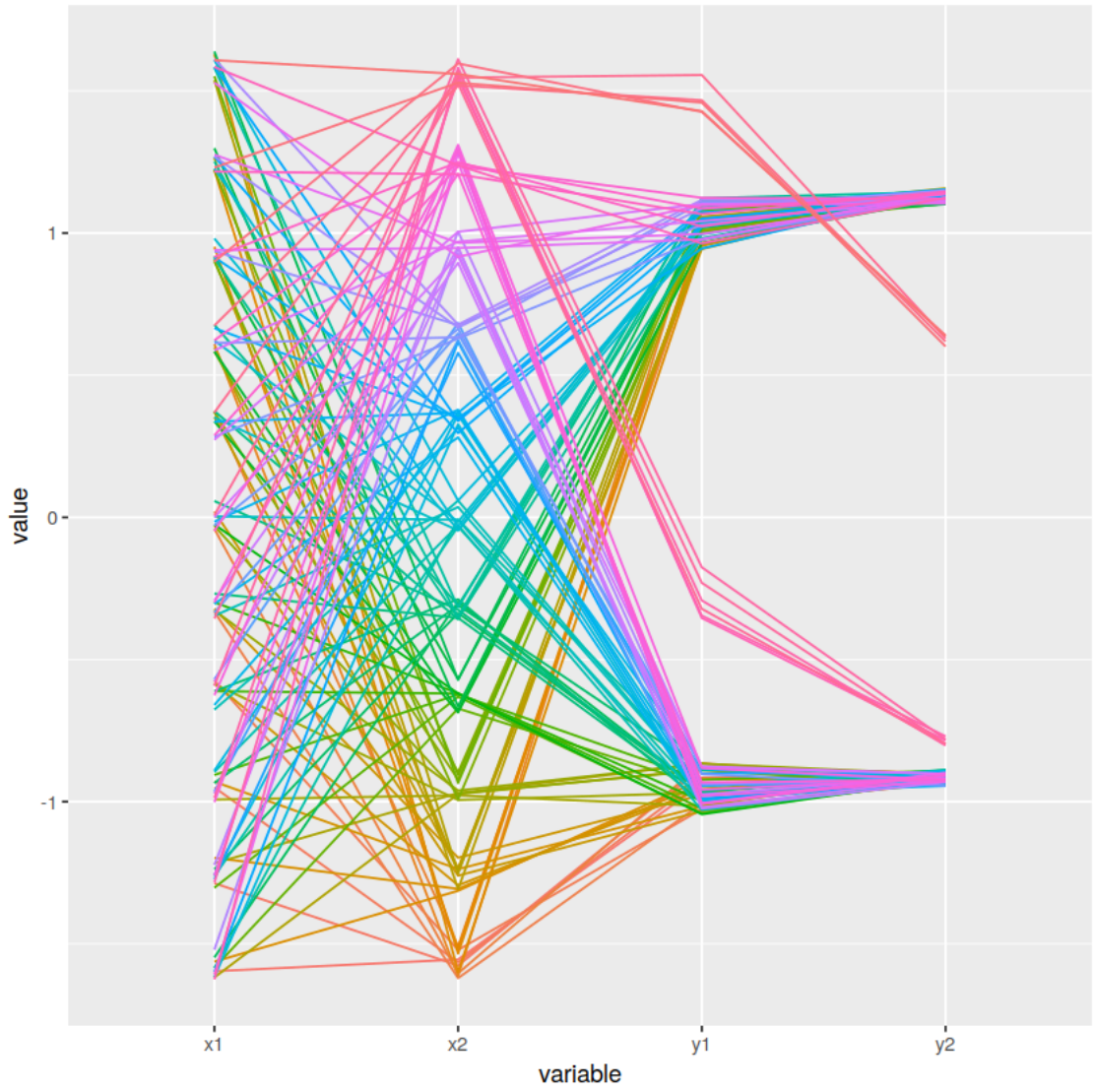
Three levels at the final time.

[11]:
```
ggparcoord(
    s2d[x1 %in% levels2 & x2 %in% levels3 & t == 10] %>% jitter2d,
    columns=c(2:3,5:6),
    groupColumn=1
)
```



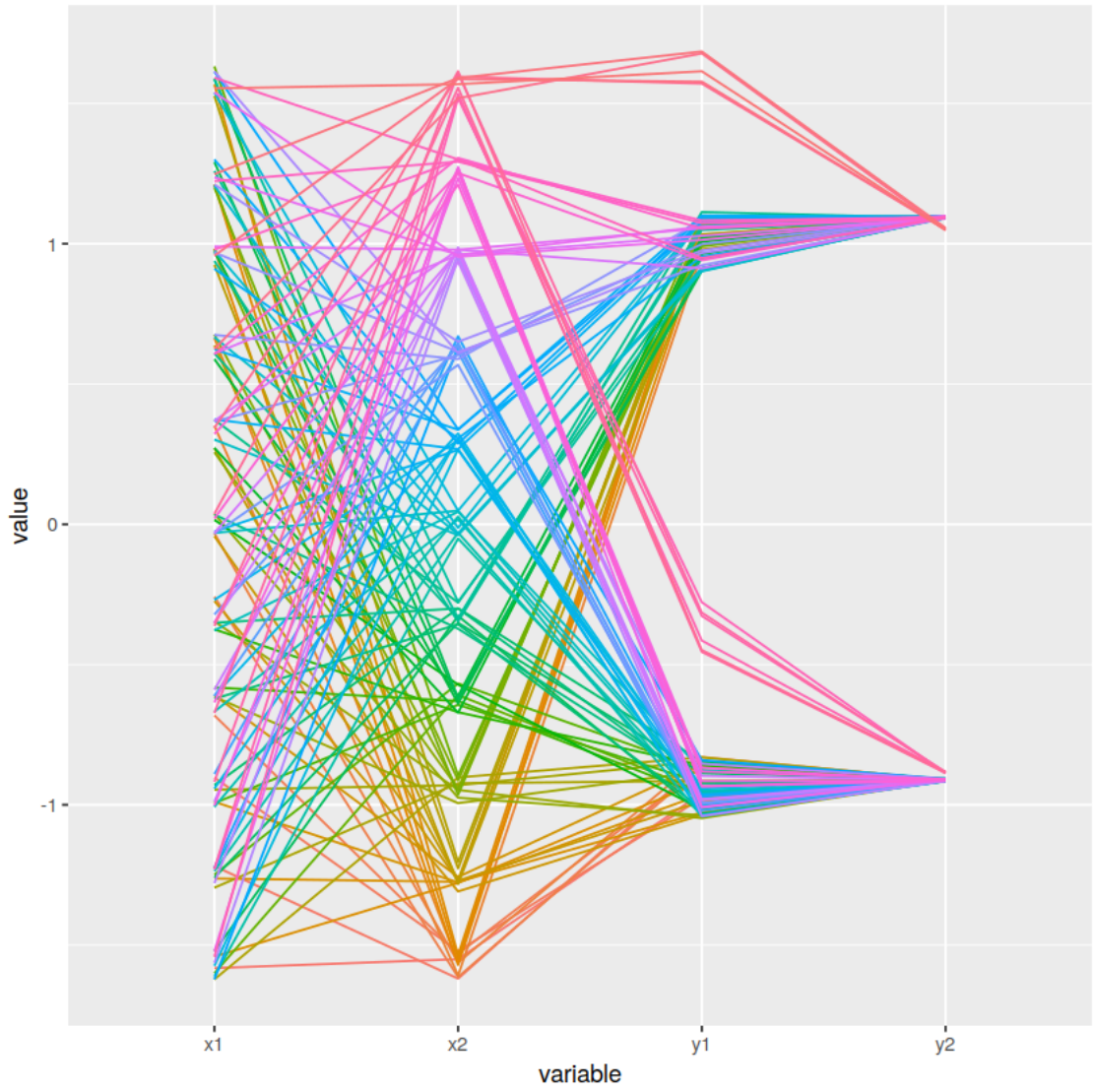
All observations at final time.

[12]:
```
ggparcoord(
    s2d[t == 10] %>% jitter2d,
    columns=c(2:3,5:6),
    groupColumn=1
) + guides(color=FALSE)
```



All results at half of the final time.

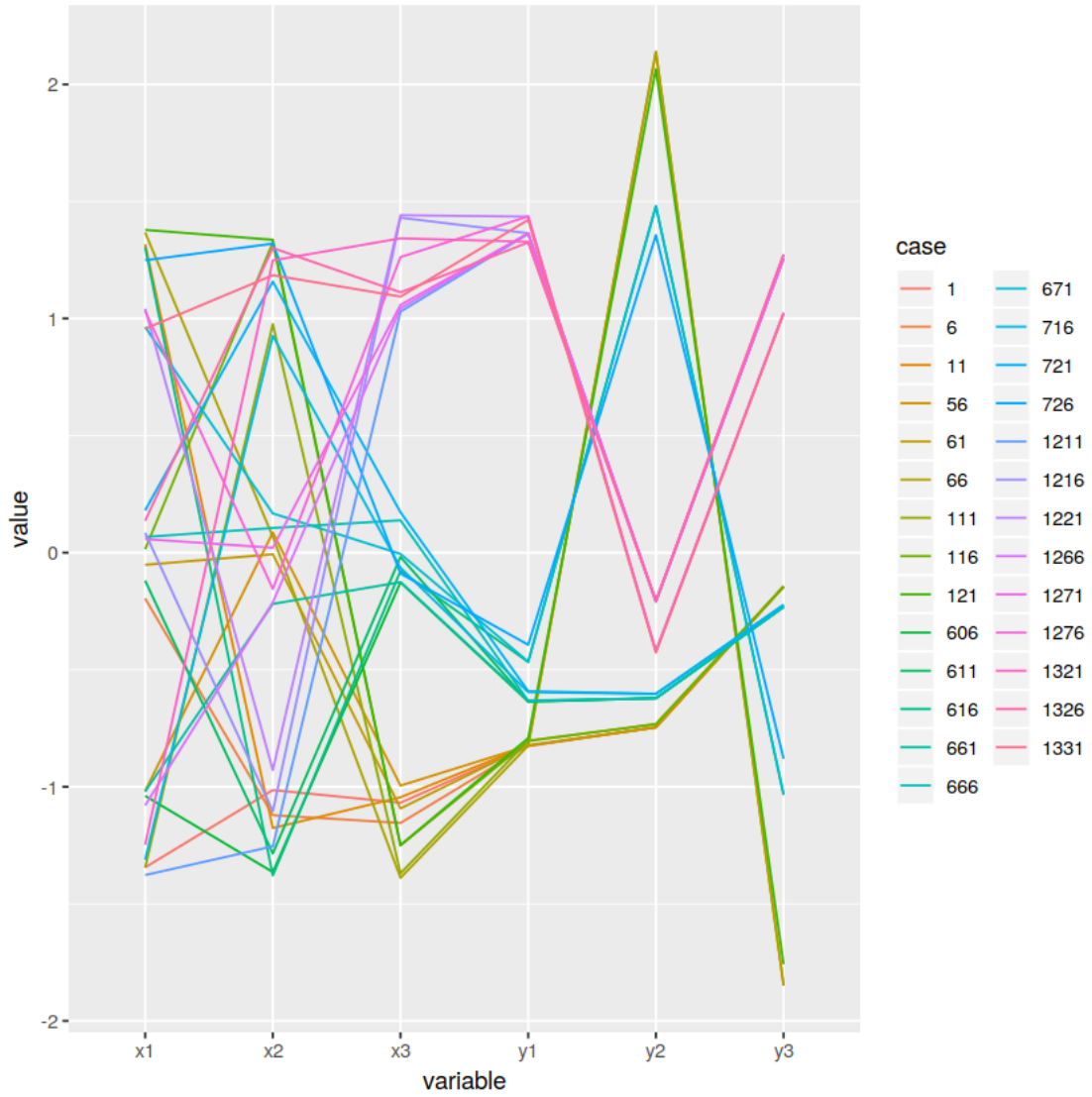
[13]:
```
ggparcoord(
    s2d[t == 5] %>% jitter2d,
    columns=c(2:3,5:6),
    groupColumn=1
) + guides(color=FALSE)
```



1.5.2 3D dataset.

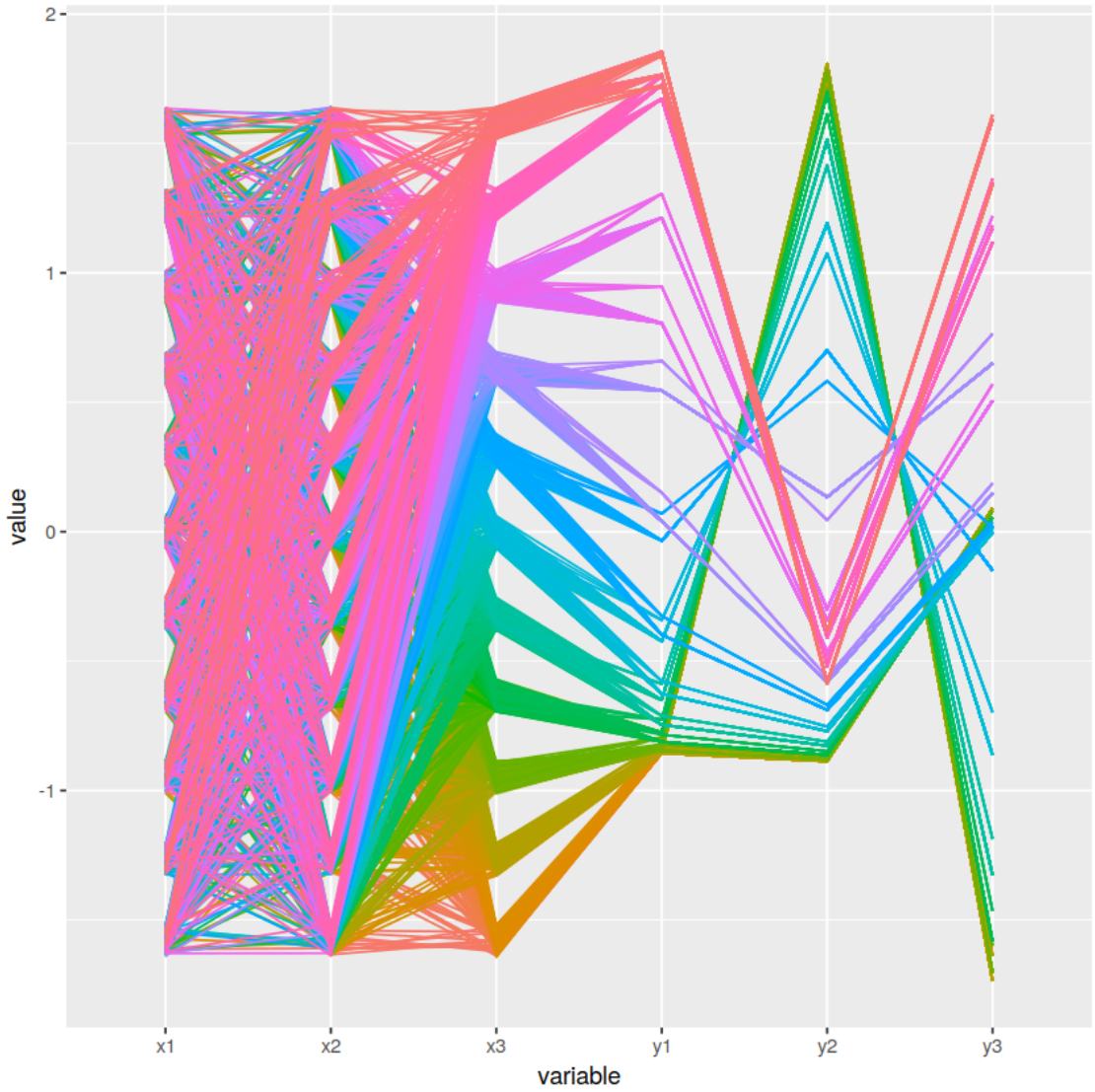
Three levels at the final time.

[14]:
```
ggparcoord(
    s3d[x1 %in% levels3 & x2 %in% levels3 & x3 %in% levels3 & t == 10] %>%␣
 ↪jitter3d,
    columns=c(2:4,6:8),
    groupColumn=1
)
```



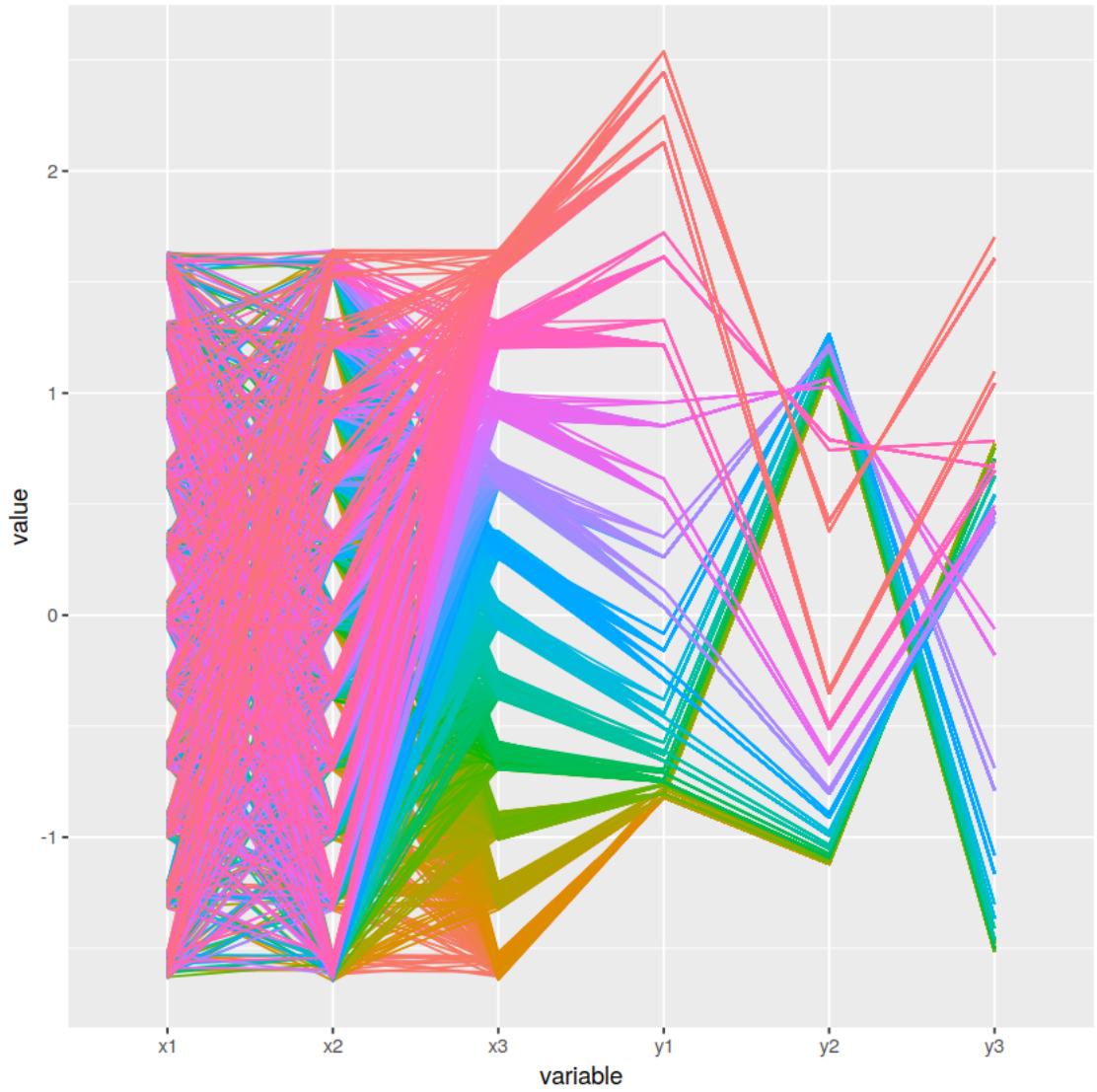
All observations at final time.

[15]:
```
ggparcoord(
    s3d[t == 10] %>% jitter3d,
    columns=c(2:4,6:8),
    groupColumn=1
) + guides(color=FALSE)
```



All results at half of the final time.

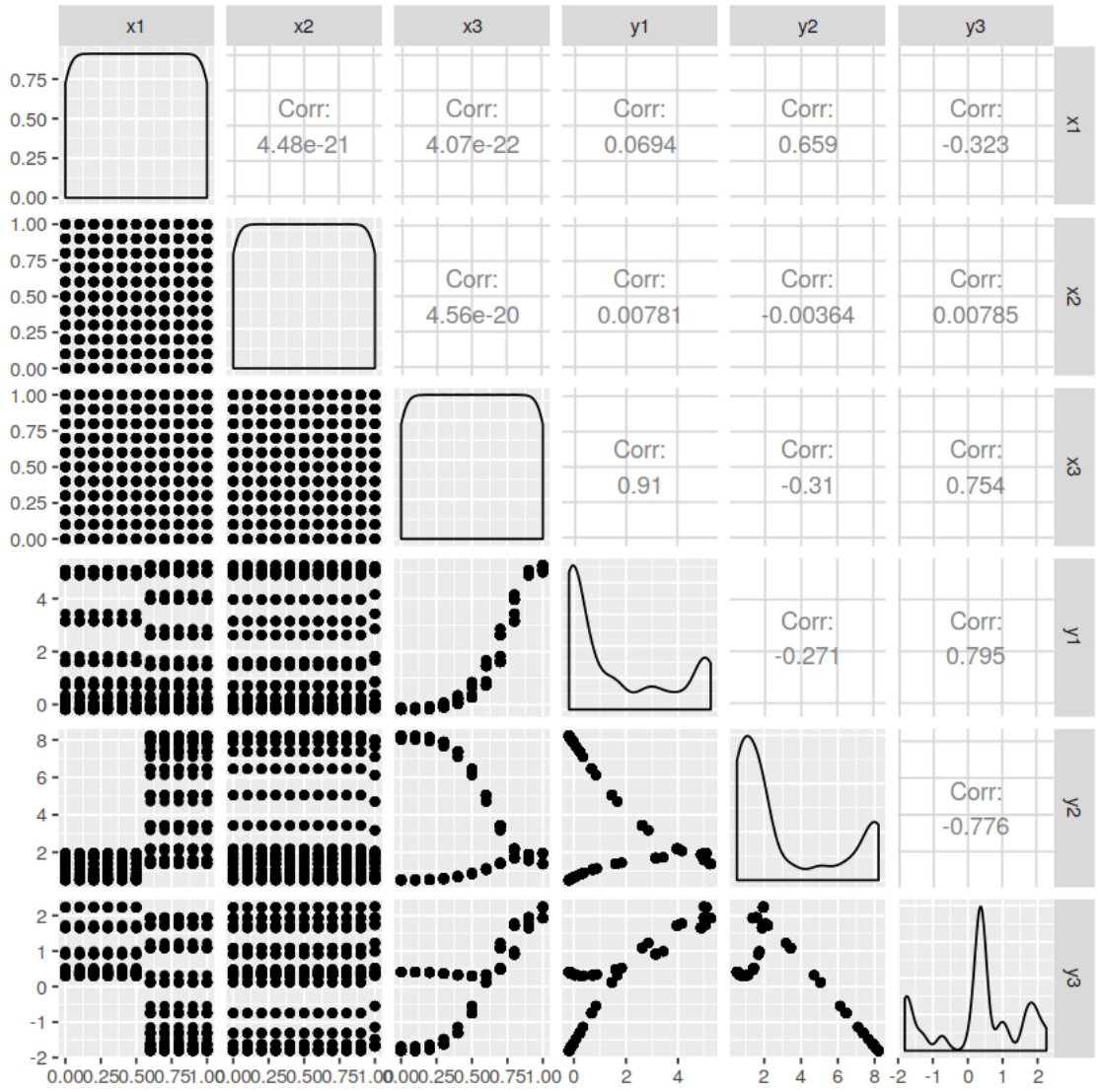
[16]:
```
ggparcoord(
    s3d[t == 5] %>% jitter3d,
    columns=c(2:4,6:8),
    groupColumn=1
) + guides(color=FALSE)
```



1.6 Scatterplot matrices.

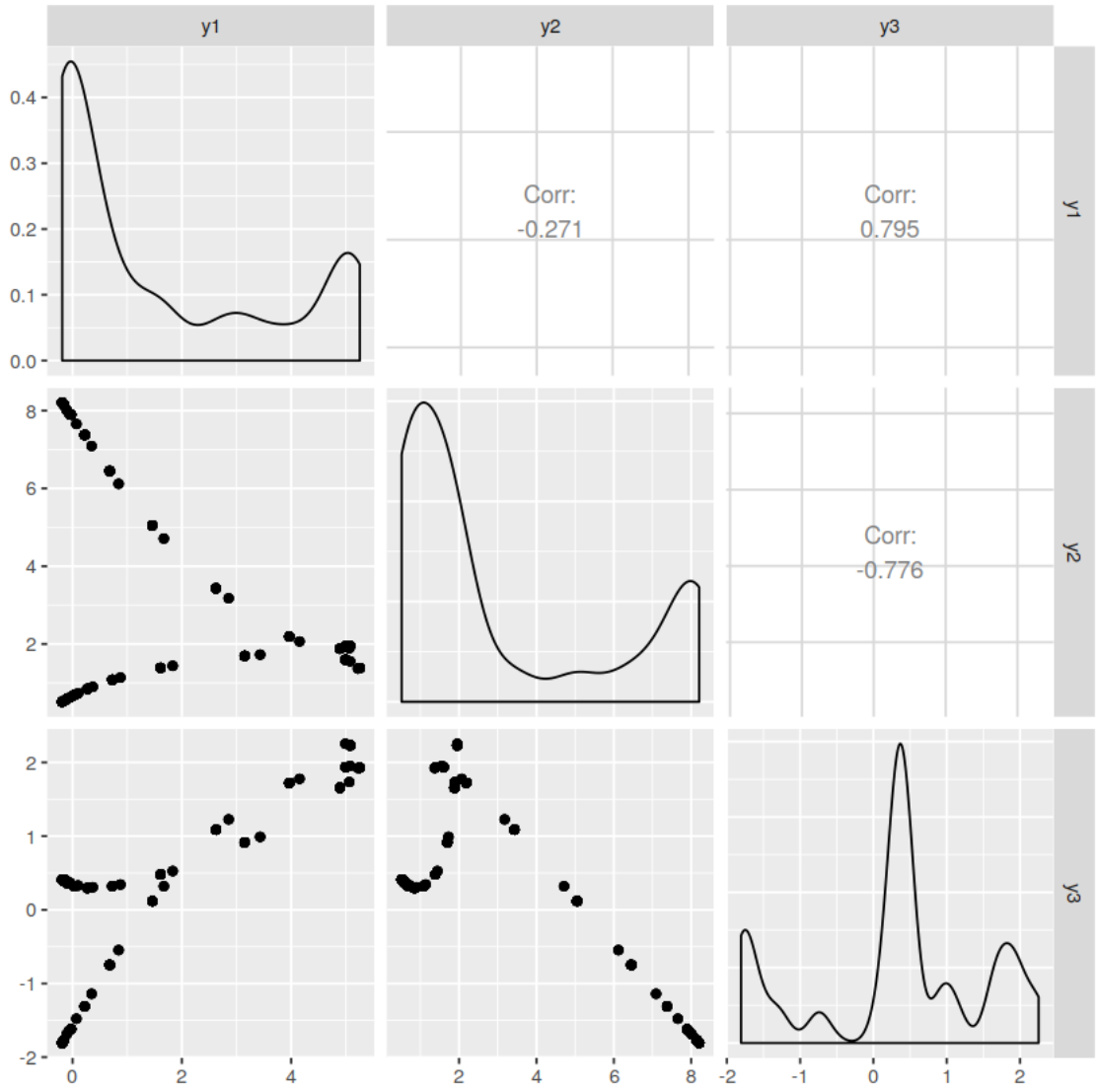
1.6.1 Independent and dependent variables at final time.

[17]:
```
ggpairs(
    s3d[t == 10],
    columns=c(2:4,6:8)
)
```



1.6.2 Dependent variables at final time.

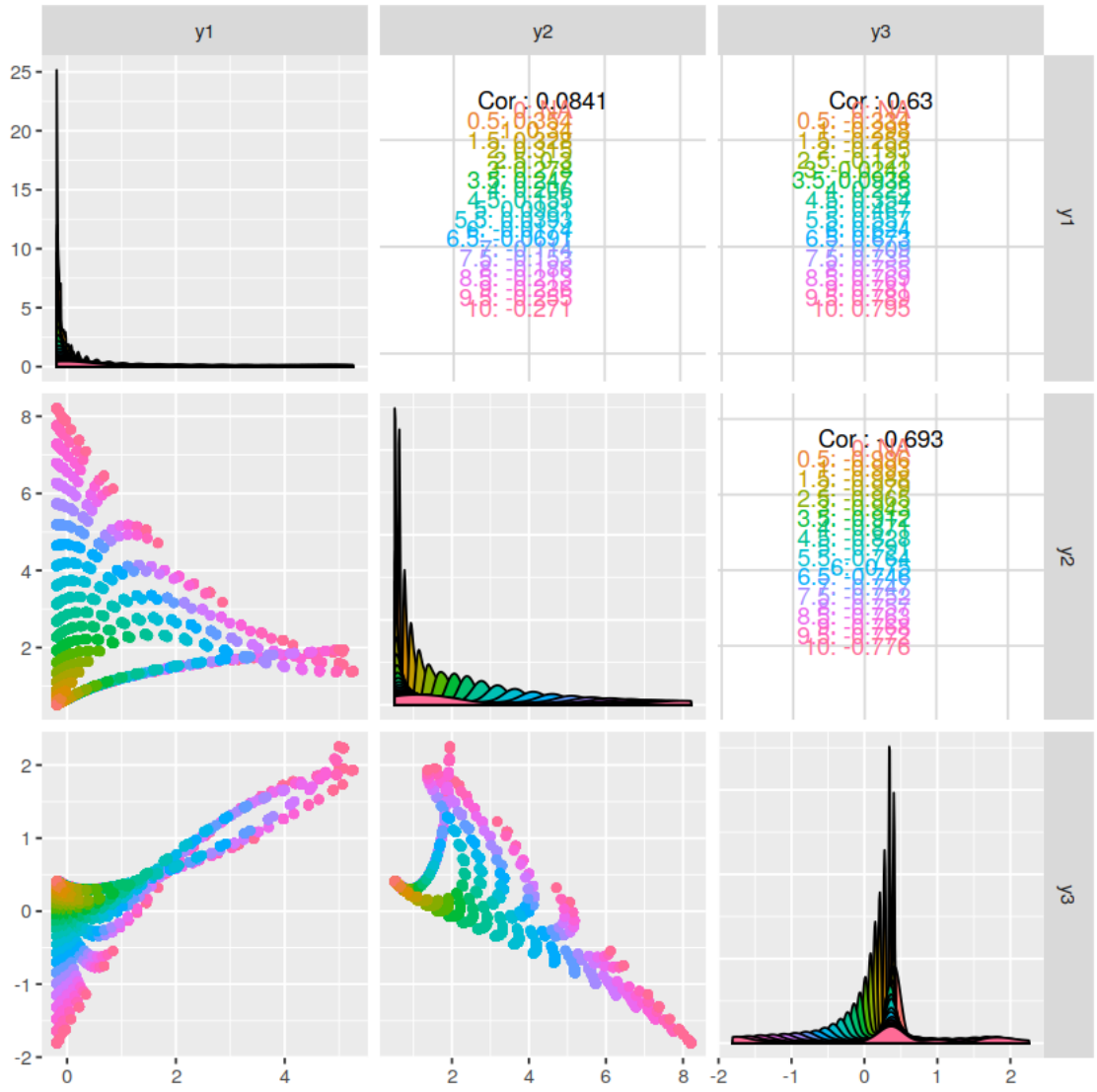
[18]:
```
ggpairs(
    s3d[t == 10],
    columns=6:8
)
```



1.6.3 Dependent variables at all times.

[19]:
```
ggpairs(s3d, columns=6:8, mapping=aes(color=factor(t)))
```

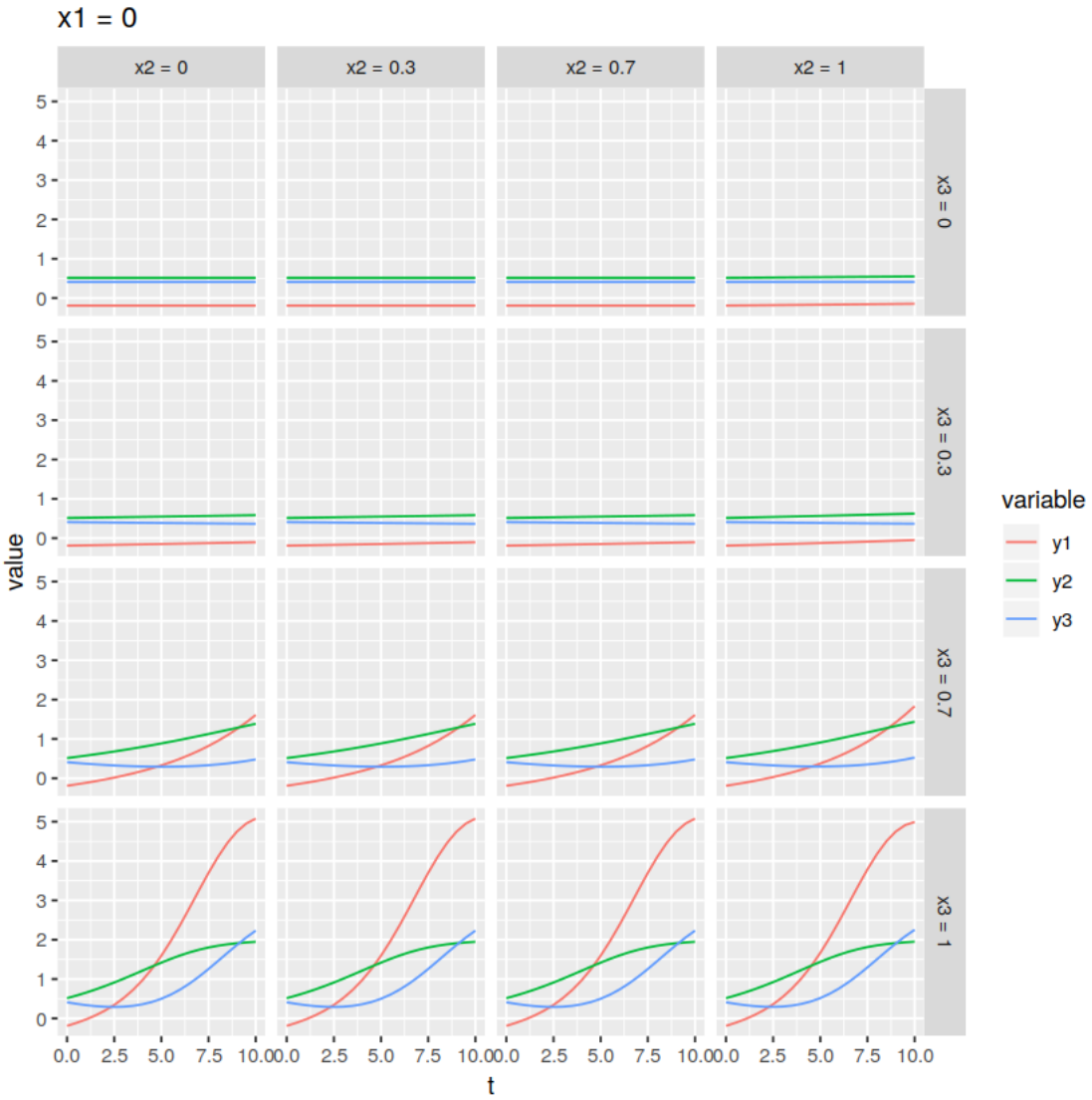
```
Warning message in cor(x, y, method = method, use = use):
the standard deviation is zeroWarning message in cor(x, y, method = method,
use = use):
the standard deviation is zeroWarning message in cor(x, y, method = method,
use = use):
```

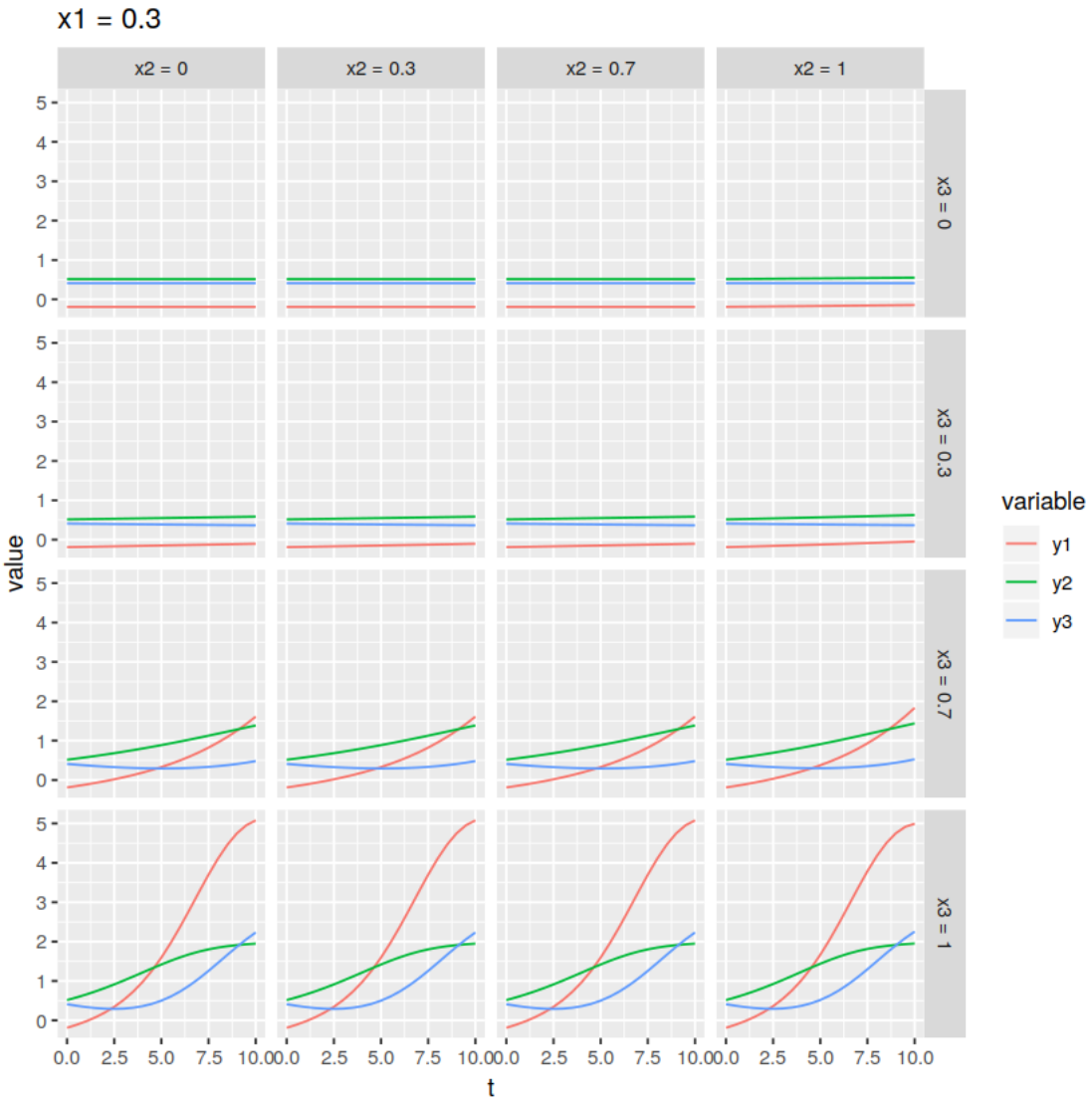


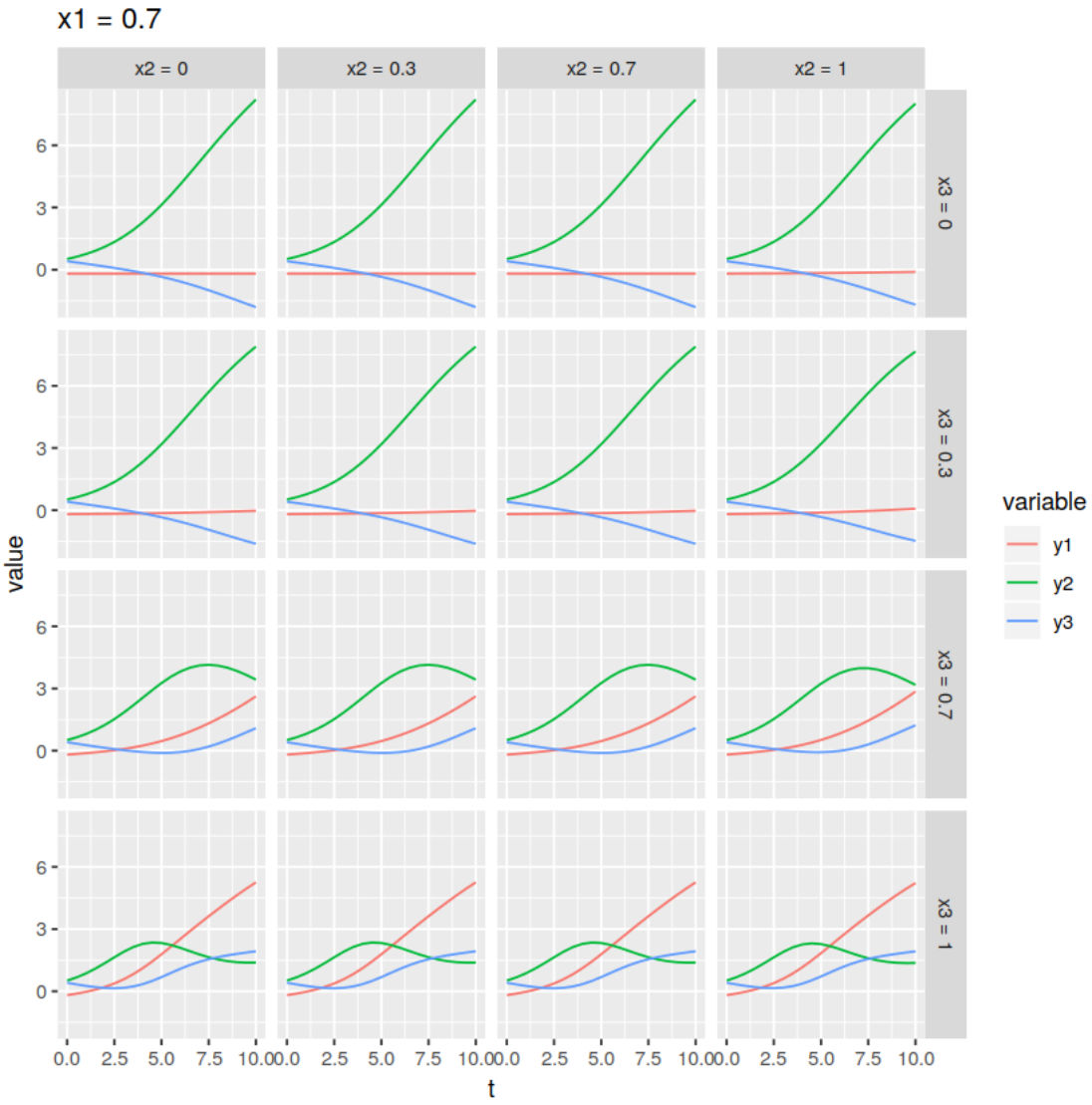
1.7 Timeseries plots conditioned by independent variables.

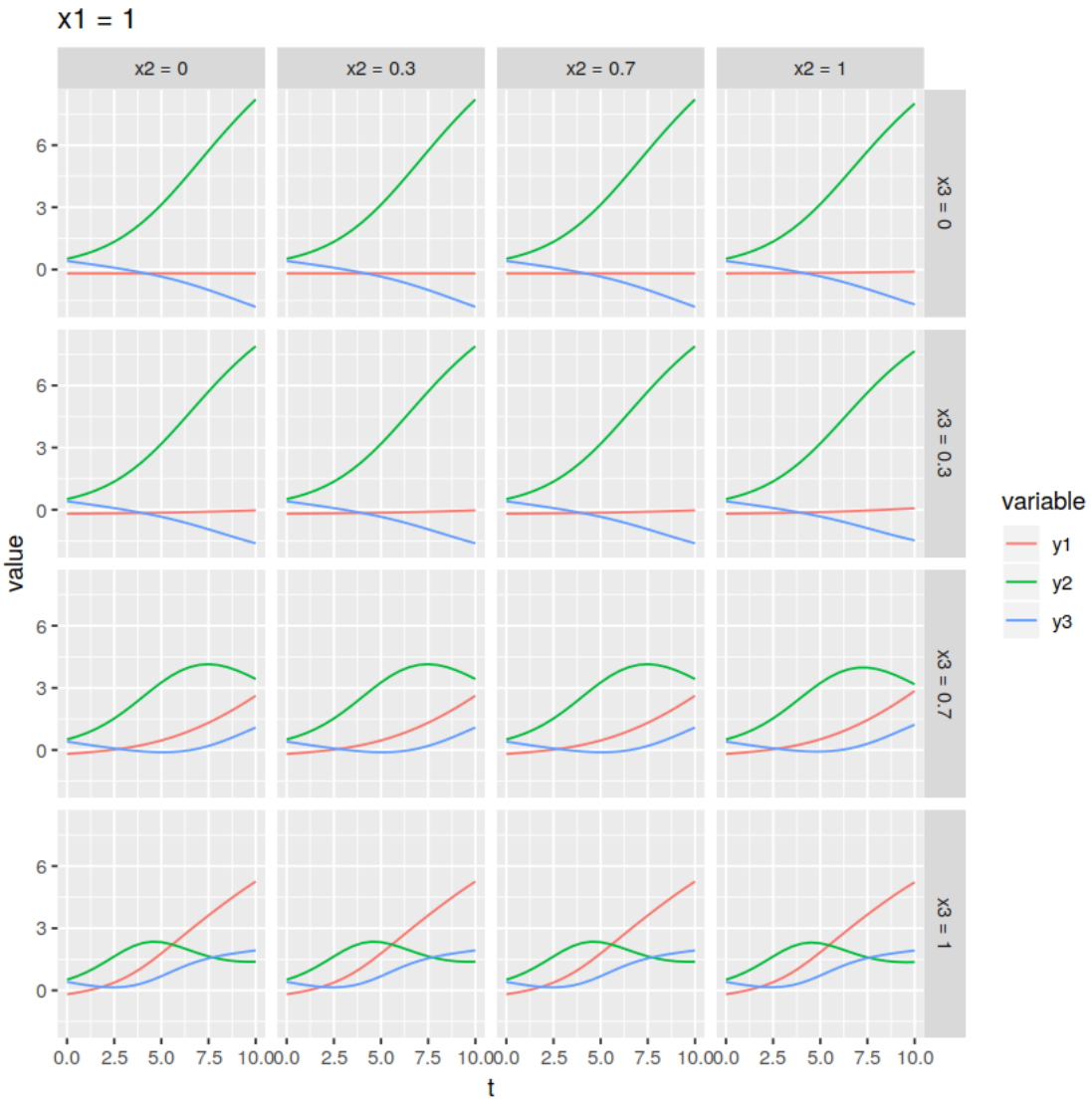
[20]:
```
x2.lab <- paste("x2 =", levels4)
names(x2.lab) <- levels4
x3.lab <- paste("x3 =", levels4)
names(x3.lab) <- levels4

for (x1.value in levels4) {
    g <- ggplot(
        melt(
            s3d[x1 == x1.value & x2 %in% levels4 & x3 %in% levels4],
            id.vars=c("case", "t", "x1", "x2", "x3")
```

```
        ),
        aes(x=t, y=value, color=variable)
    ) +
        facet_grid(
            x3 ~ x2,
            labeller=labeller(x2=x2.lab, x3=x3.lab)) +
        ggtitle(paste("x1 =", x1.value)) +
        geom_line()
    print(g)
}
```




x1 = 0.3
x2 = 0
x2 = 0.3
x2 = 0.7
x2 = 1
x3 = 0
x3 = 0.3
x3 = 0.7
x3 = 1
value
t
variable
y1
y2
y3


x1 = 0.7
x2 = 0
x2 = 0.3
x2 = 0.7
x2 = 1
x3 = 0
x3 = 0.3
x3 = 0.7
x3 = 1
value
t
variable
y1
y2
y3


x1 = 1
x2 = 0
x2 = 0.3
x2 = 0.7
x2 = 1
x3 = 0
x3 = 0.3
x3 = 0.7
x3 = 1
value
t
variable
y1
y2
y3