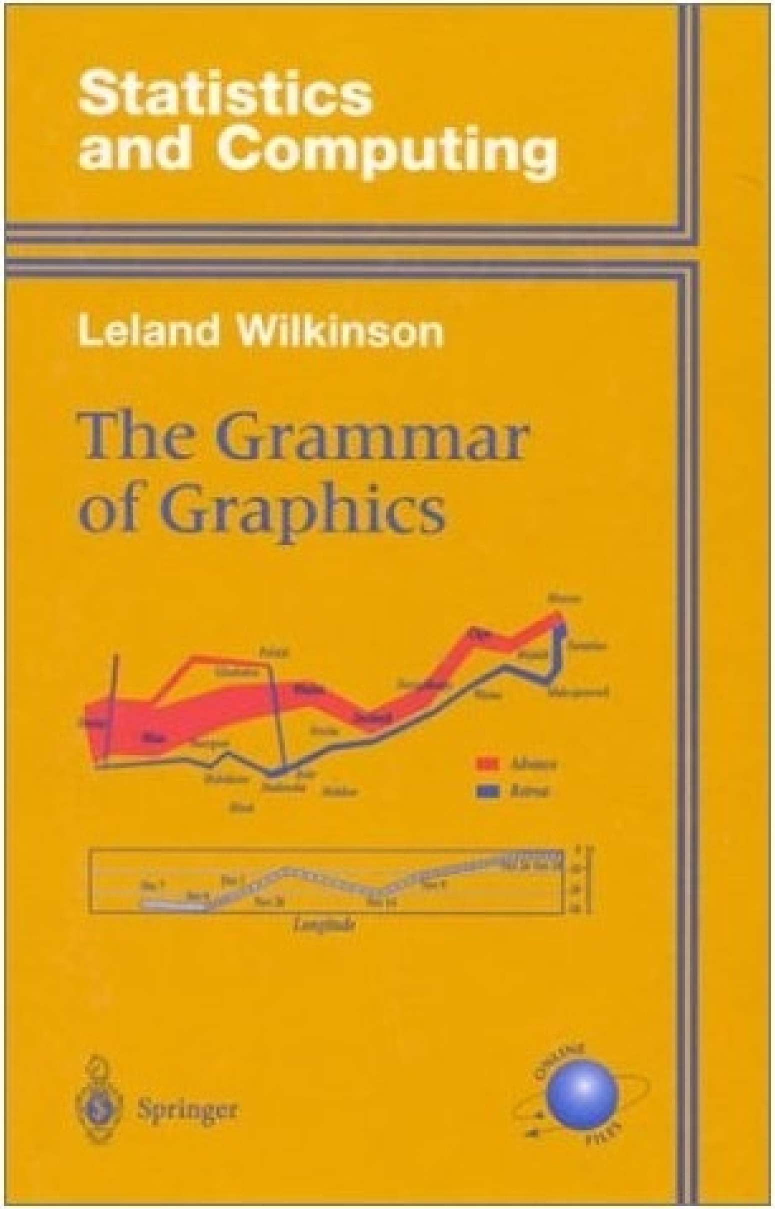
HEALTH & SAFETY LABORATORY

Easy interactive ggplots

by Richie Cotton

bitly.com/jV1NBn

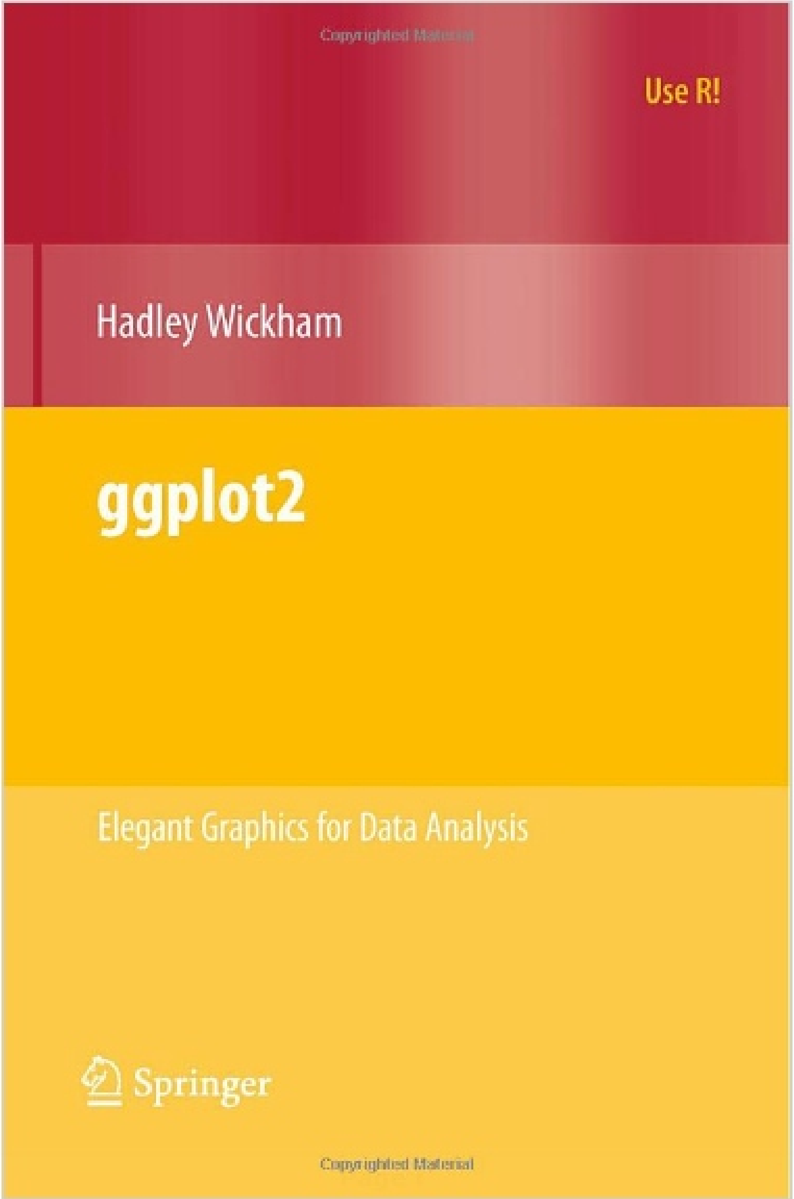


Statistics and Computing

Leland Wilkinson

The Grammar of Graphics

Springer



Use R!

Hadley Wickham

ggplot2

Elegant Graphics for Data Analysis

Springer

air concentration

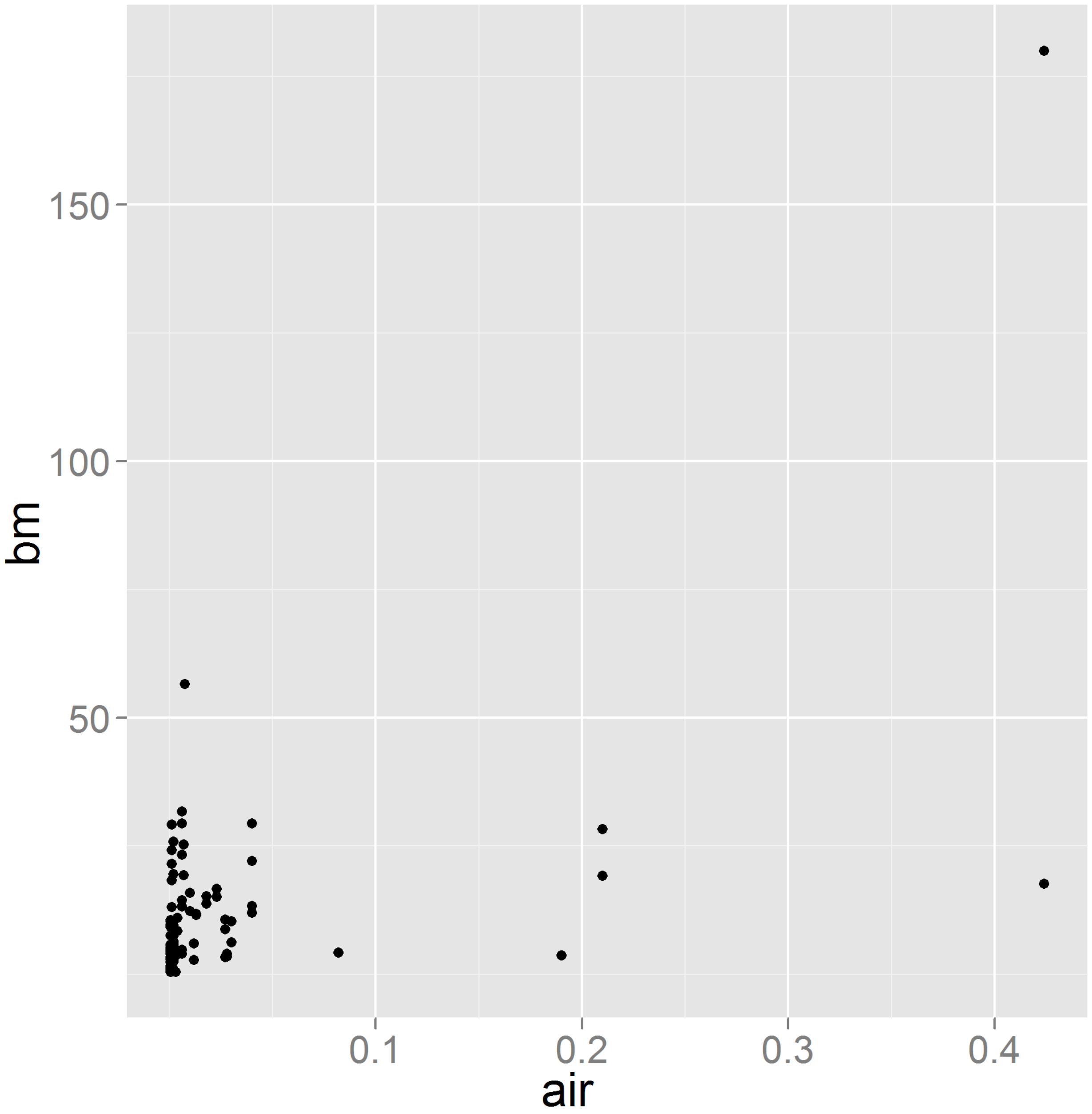
pre/post shift urine samples

type of welding

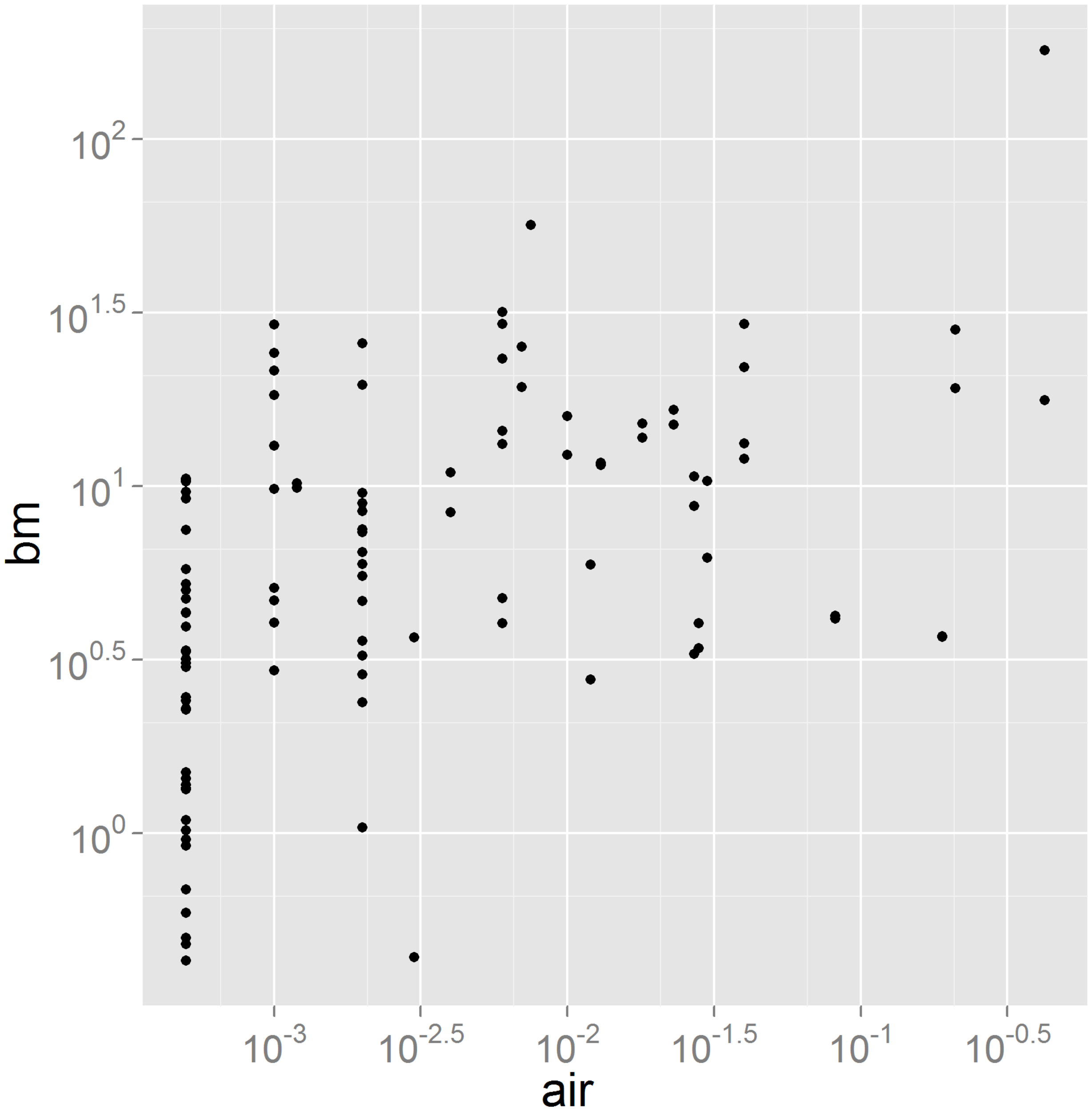
respiratory protective equipment (rpe)?

the static version

```
p <- ggplot(chromium, aes(air, bm)) +
    geom_point()
print(p)
```

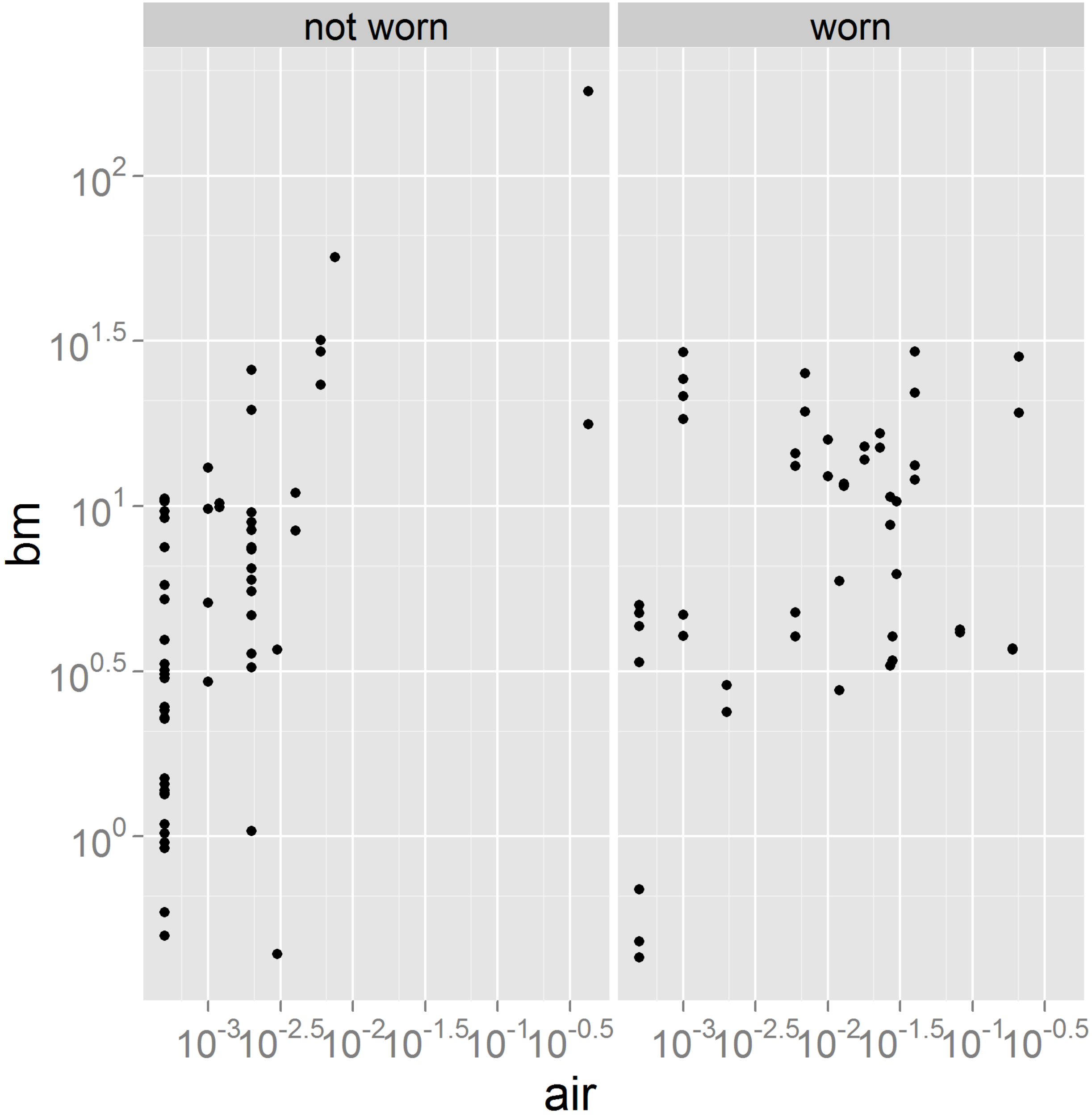

150
100
50
bm
0.1
0.2
0.3
0.4
air

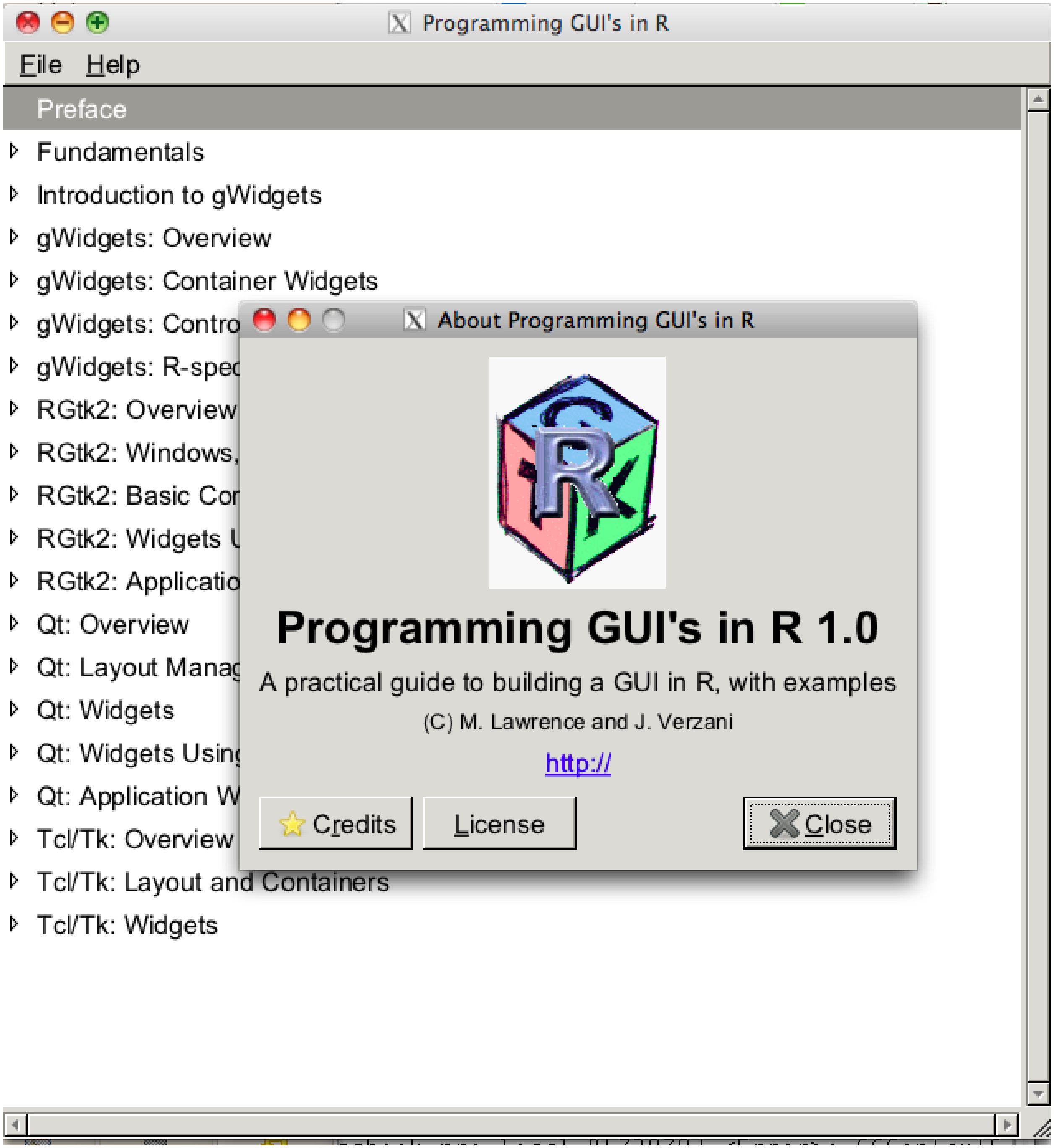
```
p <- ggplot(chromium, aes(air, bm)) +
    geom_point() +
    scale_x_continuous(trans=“log10”) +
    scale_y_continuous(trans=“log10”)
print(p)
```



10^{2}
$10^{1.5}$
10^{1}
$10^{0.5}$
10^{0}
bm
10^{-3}
$10^{-2.5}$
10^{-2}
$10^{-1.5}$
10^{-1}
$10^{-0.5}$
air

```
p <- ggplot(chromium, aes(air, bm)) +
    geom_point() +
    scale_x_continuous(trans=“log10”) +
    scale_y_continuous(trans=“log10”) +
    facet_wrap(~ rpe)
print(p)
```


not worn
worn
bm
10^2
$10^{1.5}$
10^1
$10^{0.5}$
10^0
air
10^{-3}
$10^{-2.5}$
10^{-2}
$10^{-1.5}$
10^{-1}
$10^{-0.5}$
10^{-3}
$10^{-2.5}$
10^{-2}
$10^{-1.5}$
10^{-1}
$10^{-0.5}$



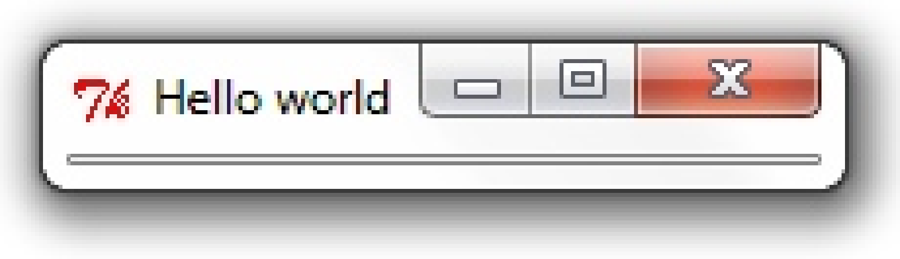
Programming GUI's in R 1.0

A practical guide to building a GUI in R, with examples

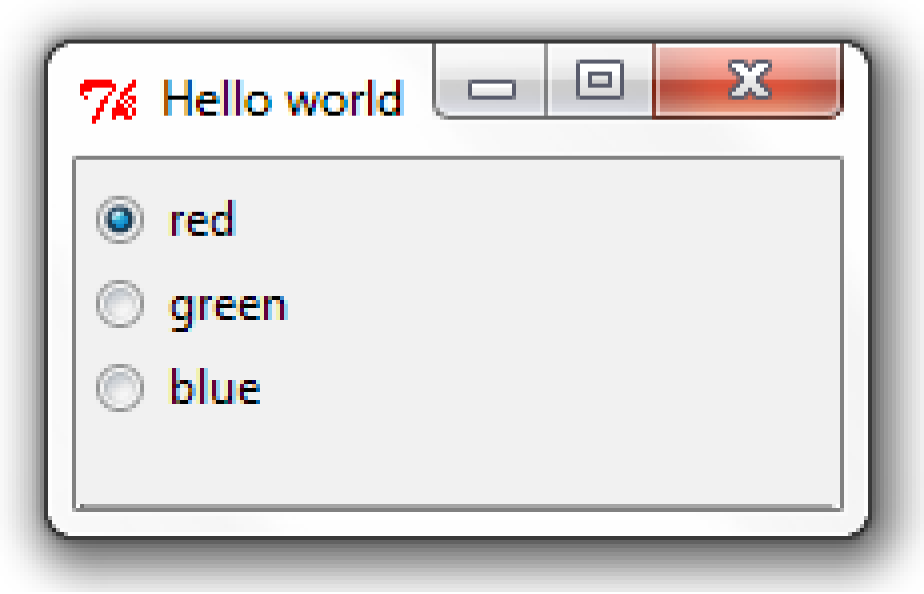
(C) M. Lawrence and J. Verzani

http://

```
win <- gwindow("Hello world")
```



```
win <- gwindow("Hello world")
rad <- gradio(
  c("red", "green", "blue"),
  container = win
)
```



Hello world

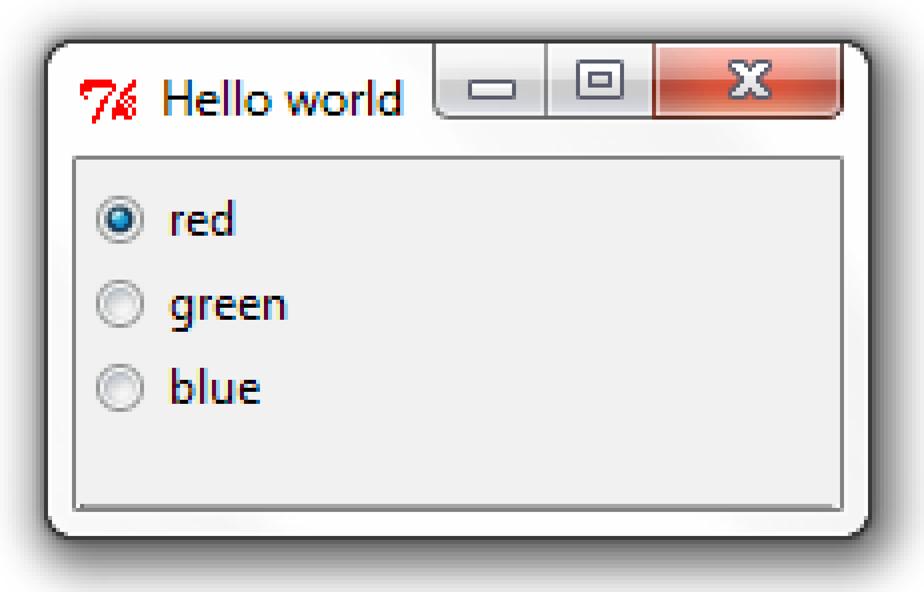
● red

○ green

○ blue

```
svalue(rad)                      #red
svalue(rad, index = TRUE) #1

svalue(rad) <- "green"
svalue(rad, index = TRUE) <- 2
```

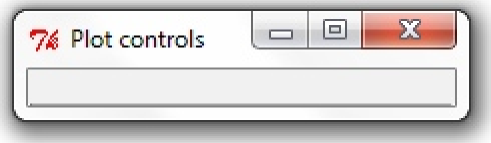


● red
○ green
○ blue

```
win <- gwindow("Hello world")
rad <- gradio(
  c("red", "green", "blue"),
  container = win,
  handler = function(h, ...)
  {
    cat("You chose", svalue(h$obj))
  }
)
```

creating the plot controls

```
win_ctrls <- gwindow("Plot controls")
grp_ctrls <- ggroup(
  container = win_ctrls,
  horizontal = FALSE
)
```



changing scales

changing facets

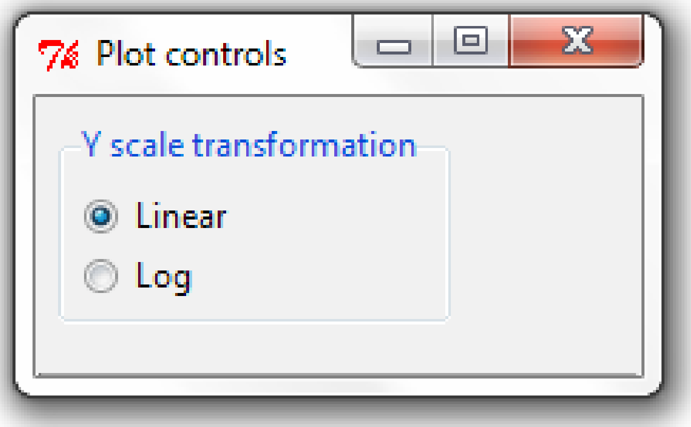
changing the dataset

changing the plot title

changing geoms

```
available_scales <- c(
  Linear = "identity",
  Log    = "log10"
)
```

```
frm_scale_trans_y <- gframe(
  "Y scale transformation",
  container = grp_ctrls,
  expand = TRUE
)
```



Plot controls

Y scale transformation

● Linear

○ Log

```
rad_scale_trans_y <- gradio(
  names(available_scales),
  container = frm_scale_trans_y,
  handler = function(h, ...)
  {
    scale_trans_y <-
      available_scales[svalue(h$obj)]
    p <<- p +
      scale_y_continuous(
        trans = scale_trans_y
      )
    print(p)
  }
)
```

```
rad_scale_trans_y <- gradio(
  names(available_scales),
  container = frm_scale_trans_y,
  handler = function(h, ...)
  {
    scale_trans_y <-
      available_scales[svalue(h$obj)]
    p <<- p +
      scale_y_continuous(
        trans = scale_trans_y
      )
    print(p)
  }
)
```

```
rad_scale_trans_y <- gradio(
  names(available_scales),
  container = frm_scale_trans_y,
  handler = function(h, ...)
  {
    scale_trans_y <-
      available_scales[svalue(h$obj)]
    p <<- p +
      scale_y_continuous(
        trans = scale_trans_y
      )
    print(p)
  }
)
```

```
rad_scale_trans_y <- gradio(
  names(available_scales),
  container = frm_scale_trans_y,
  handler = function(h, ...)
  {
    scale_trans_y <-
      available_scales[svalue(h$obj)]
    p <<- p +
      scale_y_continuous(
        trans = scale_trans_y
      )
    print(p)
  }
)
```

```
rad_scale_trans_y <- gradio(
  names(available_scales),
  container = frm_scale_trans_y,
  handler = function(h, ...)
  {
    scale_trans_y <-
      available_scales[svalue(h$obj)]
    p <<- p +
      scale_y_continuous(
        trans = scale_trans_y
      )
    print(p)
  }
)
```

```
rad_scale_trans_y <- gradio(
  names(available_scales),
  container = frm_scale_trans_y,
  handler = function(h, ...)
  {
    scale_trans_y <-
      available_scales[svalue(h$obj)]
    p <<- p +
      scale_y_continuous(
        trans = scale_trans_y
      )
    print(p)
  }
)
```

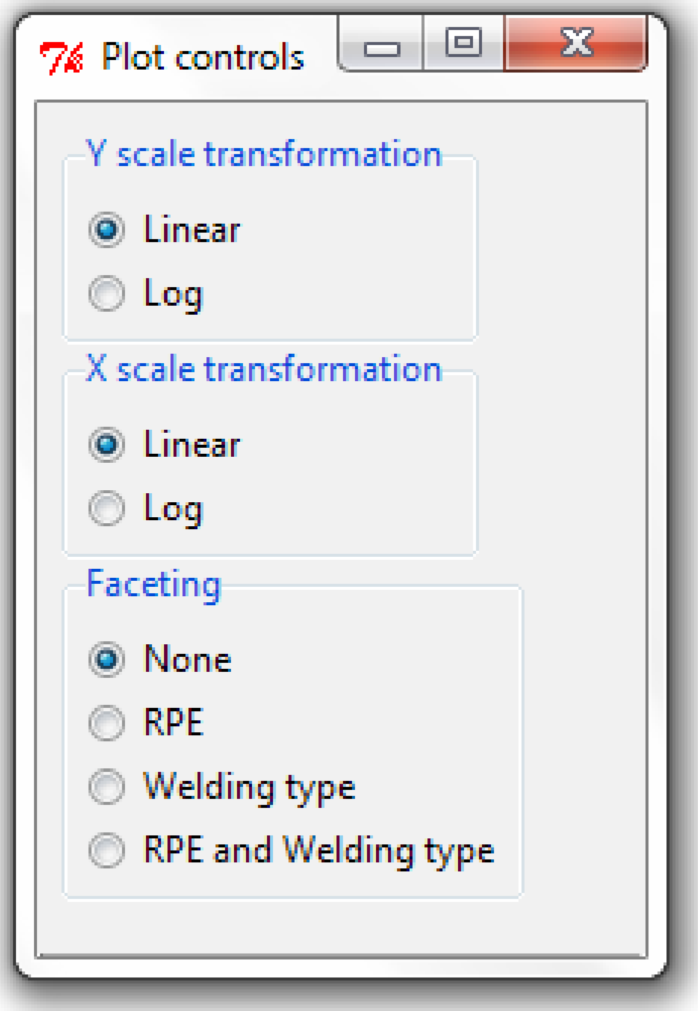
changing scales

changing facets

changing the dataset

changing the plot title

changing geoms



Plot controls

Y scale transformation

● Linear

○ Log

X scale transformation

● Linear

○ Log

Faceting

● None

○ RPE

○ Welding type

○ RPE and Welding type

```
facet_choices <- list(
  None = . ~ .,
  RPE = . ~ rpe,
  "Welding type" = . ~ welding.type,
  "RPE and Welding type" =
    rpe ~ welding.type
)
```

```
rad_facets <- gradio(
  names(facet_choices),
  container = grp_ctrls,
  handler = function(h, ...)
  {
    facet_formula <-
      facet_choices[[svalue(h$obj)]]
    p <<- p +
      facet_grid(facet_formula)
    print(p)
  }
)
```

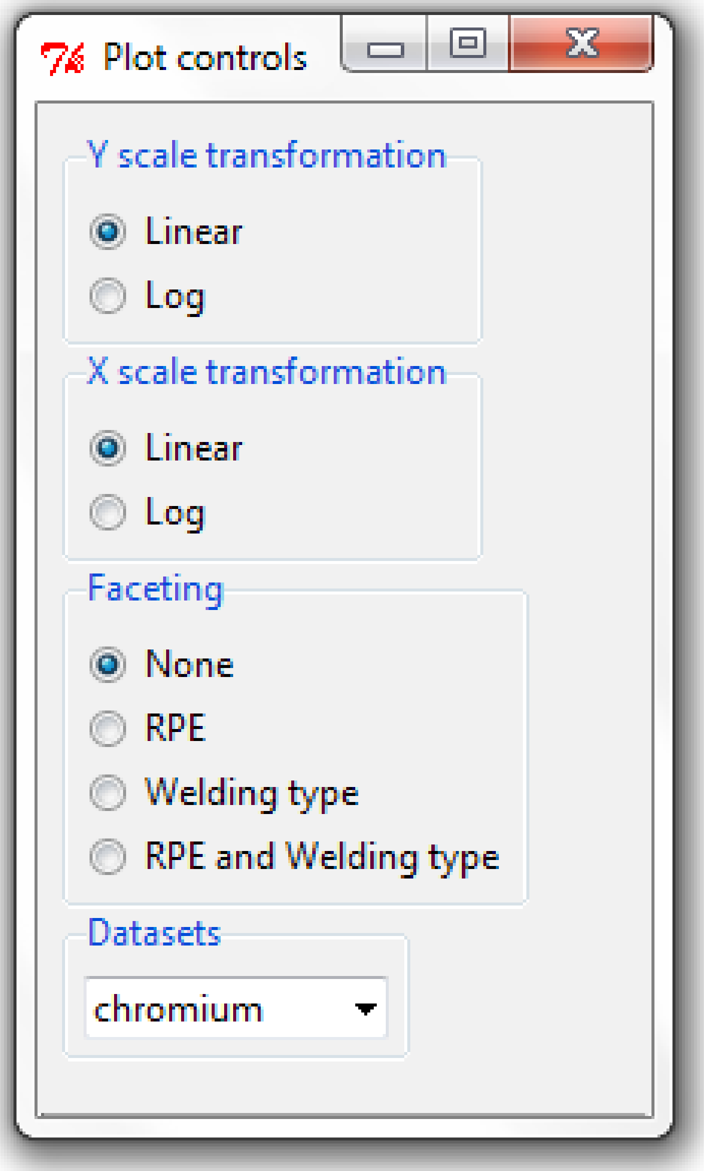
changing scales

changing facets

changing the dataset

changing the plot title

changing geoms



Plot controls

Y scale transformation

● Linear

○ Log

X scale transformation

● Linear

○ Log

Faceting

● None

○ RPE

○ Welding type

○ RPE and Welding type

Datasets

chromium ▾

```
cmb_datasets <- gcombobox(
  c("chromium", "nickel"),
  container = grp_ctrls,
  handler = function(h, ...)
  {
    p <<- p %+% get(svalue(h$obj))
    print(p)
  }
)
```

```
cmb_datasets <- gcombobox(
  c("chromium", "nickel"),
  container = grp_ctrls,
  handler = function(h, ...)
  {
    p <<- p %+% get(svalue(h$obj))
    print(p)
  }
)
```

```
cmb_datasets <- gcombobox(
  c("chromium", "nickel"),
  container = grp_ctrls,
  handler = function(h, ...)
  {
    p <<- p %+% get(svalue(h$obj))
    print(p)
  }
)
```

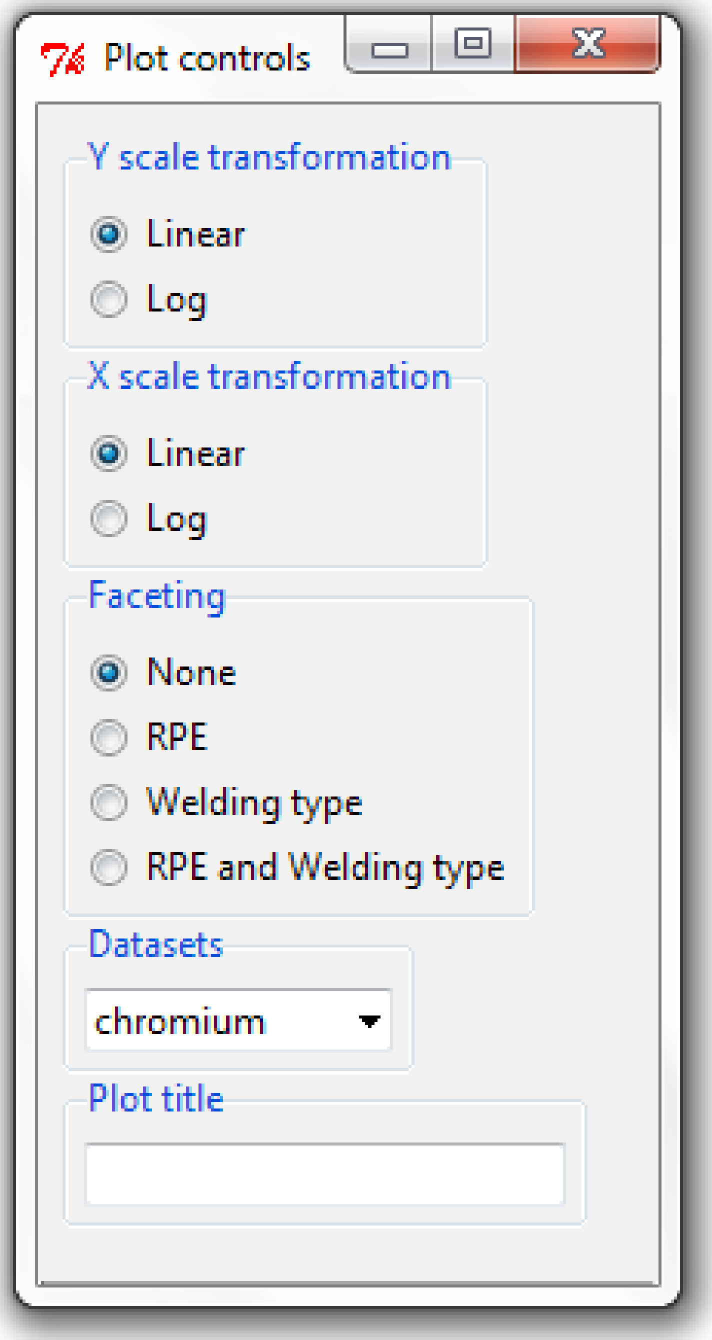
changing scales

changing facets

changing the dataset

changing the plot title

changing geoms



Plot controls

Y scale transformation

● Linear
○ Log

X scale transformation

● Linear
○ Log

Faceting

● None
○ RPE
○ Welding type
○ RPE and Welding type

Datasets

chromium ▾

Plot title

```
txt_title <- gedit(
  p$options$title,
  container = grp_ctrls,
  handler = function(h, ...)
  {
    p <<- p + opts(
      title = svalue(txt_title)
    )
    print(p)
  }
)
```

```
txt_title <- gedit(
  p$options$title,
  container = grp_ctrls,
  handler = function(h, ...)
  {
    p <<- p + opts(
      title = svalue(txt_title)
    )
    print(p)
  }
)
```

```r
txt_title <- gedit(
  p$options$title,
  container = grp_ctrls,
  handler = function(h, ...)
  {
    p <<- p + opts(
      title = svalue(txt_title)
    )
    print(p)
  }
)
```

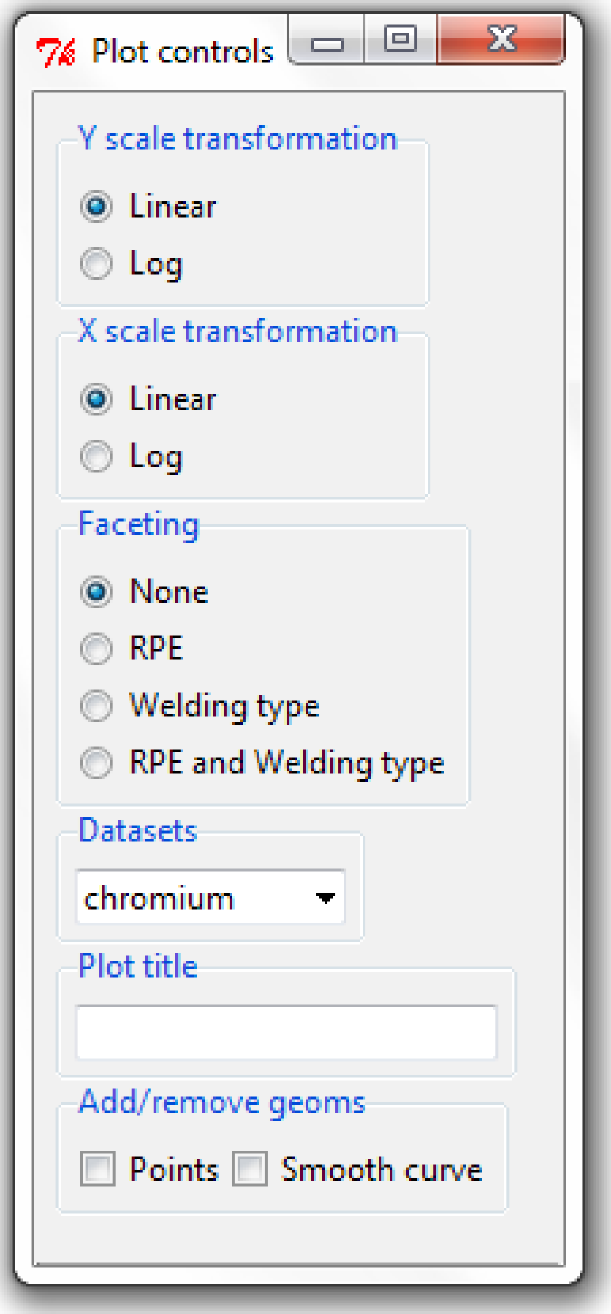
changing scales

changing facets

changing the dataset

changing the plot title

changing geoms



Plot controls

Y scale transformation

● Linear
○ Log

X scale transformation

● Linear
○ Log

Faceting

● None
○ RPE
○ Welding type
○ RPE and Welding type

Datasets

chromium

Plot title

Add/remove geoms

□ Points □ Smooth curve

```
p <- ggplot(chromium, aes(air, bm)) +
  geom_blank()
```

```
p <- ggplot(chromium, aes(air, bm)) +
  geom_blank()

print_p <- function()
{
  pp <- get("p", envir = globalenv())
  if(svalue(chk_points))
    pp <- pp + geom_point()
  if(svalue(chk_smooth))
    pp <- pp + geom_smooth()
  print(pp)
}
```

```r
chk_points <- gcheckbox(
  "Points",
  container = grp_ctrls,
  handler = function(h, ...) print_p()
)

chk_smooth <- gcheckbox(
  "Smooth curve",
  container = grp_ctrls,
  handler = function(h, ...) print_p()
)
```

what else can we do?

but what can't we do?

thanks for listening