

# Sumamry Summary

Ottavia M. Epifania

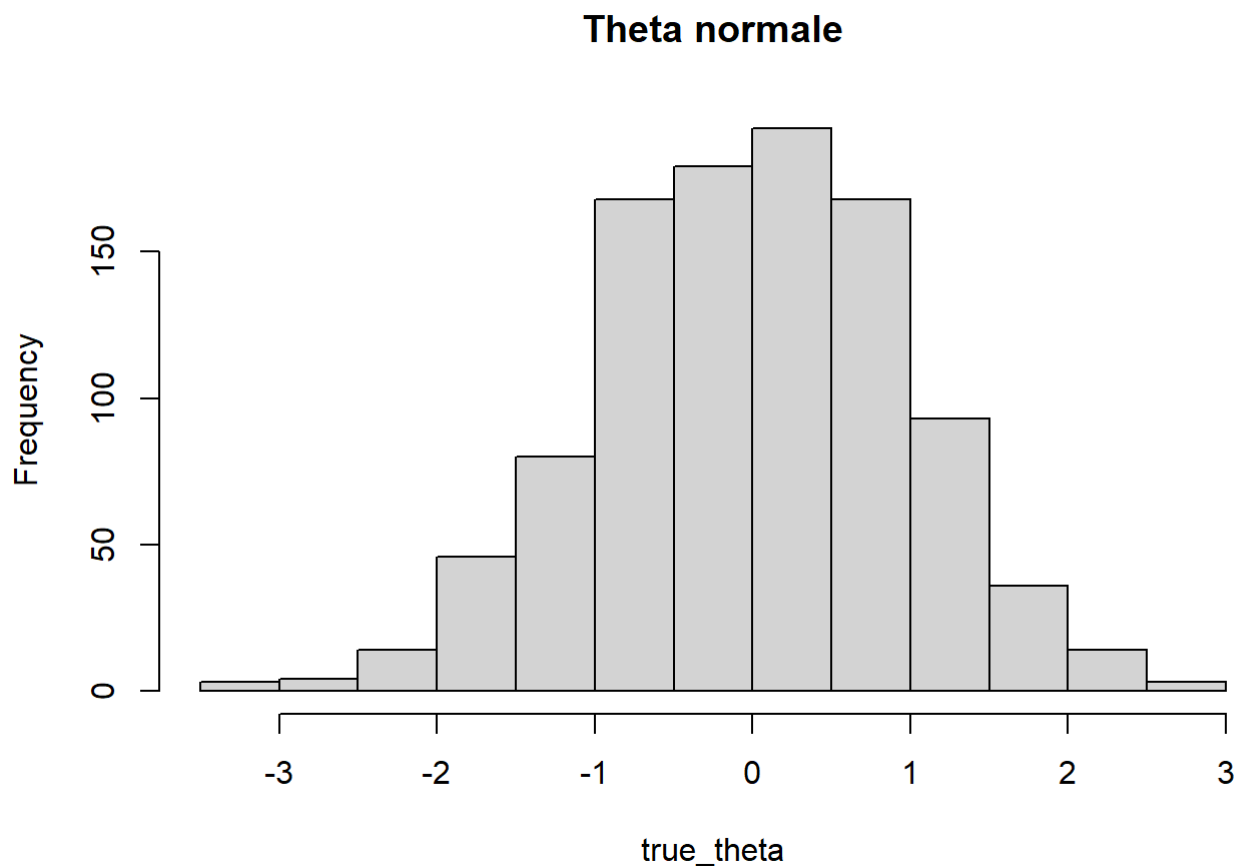
15/6/2021

## Distribuzione normale

```
set.seed(666)
N <- 1000 # number of persons
b <- runif(100, -3, 3)
a = c(runif(100, 0.4, 2))
true_theta = rnorm(N, mean = 0, sd = 1)
data <- sirt::sim.raschtype(true_theta, b = b, fixed.a = a)

diff_true <- matrix(cbind(1:length(b), b), ncol = 2)
discr_true = array(c(rep(0, length(a)), a), c(length(a),
  2, 1), dimnames = list(paste0("I", 1:length(a)),
  c("Cat0", "Cat1"), "Dim01"))

hist(true_theta, main = "Theta normale")
```



## Distribuzione Skewness “normale”

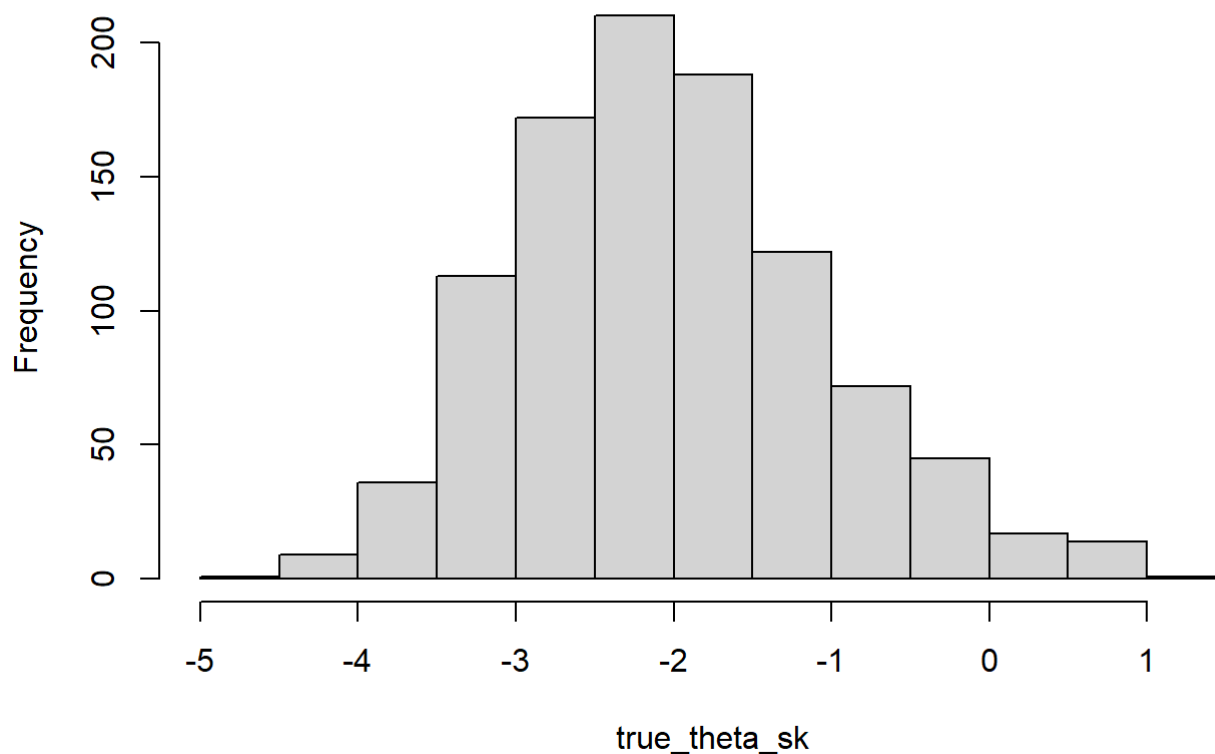
```

library(PearsonDS)
set.seed(666)
moments <- c(mean = -2, variance = 1, skewness = 0.71,
             kurtosis = 4)
true_theta_sk <- c(rpearson(1000, moments = moments))
b <- runif(100, -3, 3)
a = c(runif(100, 0.4, 2))
data_sk <- sirt::sim.raschtype(true_theta_sk, b = b,
                              fixed.a = a)

diff_true <- matrix(cbind(1:length(b), b), ncol = 2)
discr_true <- array(c(rep(0, length(a)), a), c(length(a),
        2, 1), dimnames = list(paste0("I", 1:length(a)),
        c("Cat0", "Cat1"), "Dim01"))
summary(true_theta_sk)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.569 -2.723  -2.117  -2.023  -1.431   1.010
hist(true_theta_sk, main = "Skewness normale")

```

### Skewness normale



### Skewness estrema

```

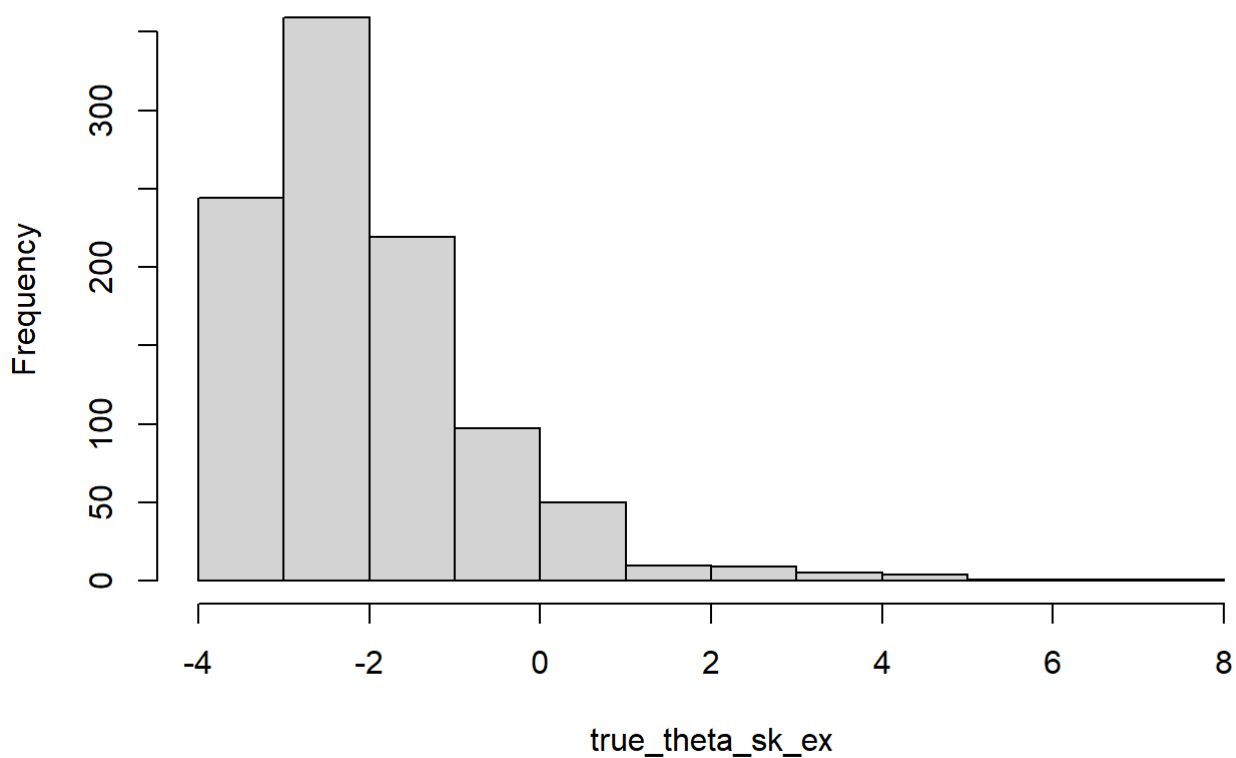
set.seed(666)
moments <- c(mean = -2, variance = 2, skewness = 1.71,
             kurtosis = 8)
true_theta_sk_ex <- c(rpearson(1000, moments = moments))
b <- runif(100, -3, 3)
a = c(runif(100, 0.4, 2))
data_sk_ex <- sirt::sim.raschtype(true_theta_sk_ex,
                                b = b, fixed.a = a)

diff_true <- matrix(cbind(1:length(b), b), ncol = 2)
discr_true <- array(c(rep(0, length(a)), a), c(length(a),
        2, 1), dimnames = list(paste0("I", 1:length(a)),
        c("Cat0", "Cat1"), "Dim01"))

summary(true_theta_sk_ex)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.850 -2.987  -2.311  -2.013  -1.413    7.265
hist(true_theta_sk_ex, main = "Skewness estrema")

```

## Skewness estrema



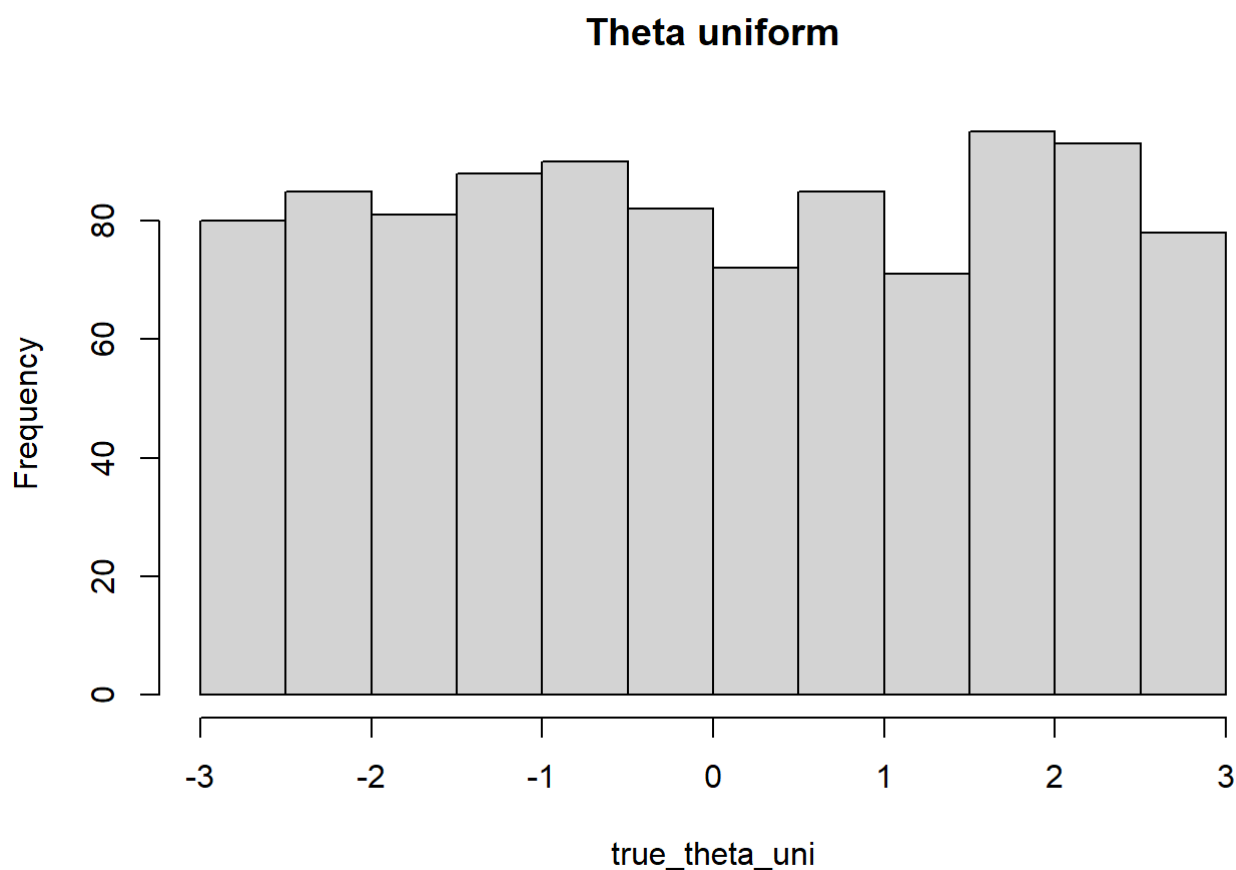
## Distribuzione uniforme

```

set.seed(666)
true_theta_uni <- c(runif(1000, min = -3, max = 3))
b <- runif(100, -3, 3)
a = c(runif(100, 0.4, 2))
diff_true <- matrix(cbind(1:length(b), b), ncol = 2)
discr_true <- array(c(rep(0, length(a)), a), c(length(a),
  2, 1), dimnames = list(paste0("I", 1:length(a)),
  c("Cat0", "Cat1"), "Dim01"))
data_uni <- sirt::sim.raschtype(true_theta_uni, b = b,
  fixed.a = a)

hist(true_theta_uni, main = "Theta uniform")

```



## Stima del modello iniziale

Modello iniziale con tutti gli item stimato come (questo è vero per tutte le distribuzioni):

```
m2p1 <- tam.mml(data, xsi.fixed = diff_true, B = discr_true)
```

## Trovo i valori theta target

SULLa base dei theta osservati

## Guided

```

ranges <- NULL
groups <- NULL

cut_value <- list()

for (i in 1:(length(num_item))) {
  ranges <- seq(min(true_theta), max(true_theta),
    length = num_item[i])
  groups <- cut(ranges, num_item[i], include.lowest = TRUE)
  cut_value[[i]] <- cut_borders(groups)
  cut_value[[i]]$mean_theta <- rowMeans(cut_value[[i]])
}

```

## Guided new

```

cut_value <- list()
# tengo solo la prima selezione di item
for (i in 1:length(num_item)) {
  cut_value[[i]] = seq(min(true_theta), max(true_theta),
    length = num_item[i])
}

```

## Cluster

```

num_clusters <- num_item
theta_mat <- matrix(true_theta, ncol = 1)
info_start <- mean(IRT.informationCurves(m2pl, theta = seq(-3,
  3, length = 1000))$test_info_curve)
# generate clusters

cluster <- list()

for (i in 1:length(num_clusters)) {
  cluster[[i]] <- kmeans(theta_mat, centers = num_clusters[i])
}

```

## Smart

Theta teorico tra -3 e 3

```

data_info_smart <- data.frame(items = 1:(ncol(data)),
  info = numeric((ncol(data))))

for (i in 1:nrow(data_info_smart)) {
  data_info_smart[i, "info"] <- mean(IRT.informationCurves(m2pl,
    theta = seq(-3, 3, length = 1000), iIndex = lab_item[i])$info_curves_item)
}

# ora scrivi il codice per la procedura iterativa
# dove dato un certo numero di item, trova il
# massimo e mano a mano toglie quel'item
filtro <- list()
data_temp <- list()
for (i in 1:length((num_item))) {
  filtro[[i]] <- data_info_smart[which(data_info_smart$info ==
    max(data_info_smart$info)), ]
  for (j in 1:(num_item[i] - 1)) {
    data_temp[[j]] <- data_info_smart[!data_info_smart$items %in%
      filtro[[i]]$items, ]
    filtro[[i]] <- rbind(filtro[[i]], data_temp[[j]][which(data_temp[[j]]$info ==
      max(data_temp[[j]]$info)), ])
  }
  names(filtro)[[i]] <- paste("number", num_item[i],
    sep = "")
}

```

## Calcolo informatività per ogni theta target

```

info_test <- NULL
temp <- list()
value <- list()
temp_data <- NULL
info_data <- NULL

for (j in 1:length(cut_value)) {
  # contiene i theta target
  value[[j]] <- cut_value[[j]][1:nrow(cut_value[[j]]),
    ]

  for (i in 1:length(lab_item)) {
    # per ognuno dei 100 item viene calcolata l'info
    # per ogni theta target
    for (m in 1:nrow(value[[j]])) {

      temp_data <- data.frame(theta_target = IRT.informationCurves(m2pl,
        theta = value[[j]][m, "mean_theta"],
        iIndex = lab_item[i])$theta, test_info = mean(IRT.informationCurves(m2pl,
        theta = value[[j]][m, "mean_theta"],
        iIndex = lab_item[i])$test_info_curve),
        item_info = mean(colSums(IRT.informationCurves(m2pl,
        theta = value[[j]][m, "mean_theta"],
        iIndex = lab_item[i])$info_curves_item)),
        item = lab_item[i], num_item = paste("number",
        nrow(value[[j]]), sep = ""))

      info_data <- rbind(info_data, temp_data) # data frame dove per ogni theta target
    } # L'info di ogni item
  }
}

```

## Calcolo info massima di ogni item per uno specifico theta target

```

temp_data <- NULL
temp_maxrange <- NULL
temp <- NULL
max_temp <- NULL

for (i in 1:length(unique(info_data$num_item))) {
  temp_data <- info_data[info_data$num_item %in%
    unique(info_data$num_item)[i], ]
  temp_maxrange <- aggregate(test_info ~ item + theta_target,
    data = temp_data, max)
  temp_maxrange$range_name <- unique(temp_data$num_item) # trova l'item maggiormente infor
mativo
  # per ogni theta target toglie l'item e il theta e
  # ricomincia da capo
  for (j in 1:length(unique(temp_maxrange$theta_target))) {
    temp <- temp_maxrange[which(temp_maxrange$test_info ==
      max(temp_maxrange$test_info)), ]
    temp_maxrange <- temp_maxrange[which(temp_maxrange$item !=
      temp$item & temp_maxrange$theta_target !=
      temp$theta_target), ]
    max_temp <- rbind(max_temp, temp)
  }
}

```

Stimo in maniera ricorsiva il modello selezionando gli item trovati al punto precedente

Parametri degli item liberi

```

out_range <- list()
model_out_range <- list()
info_out_range <- list()

for (i in 1:length(unique(max_temp$range_name))) {
  out_range[[i]] <- data[, c(max_temp[max_temp$range_name %in%unique(max_temp$range_name)
[i],
                                "item"])]
  model_out_range[[i]] <- tam.mml.2pl(out_range[[i]]) # stimo il modello
  info_out_range[[i]] <- IRT.informationCurves(model_out_range[[i]], # calcolo info del mod
ello
                                theta = seq(-3, 3, length = 1000))
  names(info_out_range)[i] <- unique(max_temp$range_name)[i]
}

```

Parametri degli item vincolati



```

out_range_theta <- list()
model_out_range_theta <- list()
info_out_range_theta <- list()

for (i in 1:length(unique(max_temp$range_name))) {
  out_range_theta[[i]] <- data[, c(max_temp[max_temp$range_name %in% unique(max_temp$range_n
ame)][i],
                                "item"])]
  model_out_range_theta[[i]] <- tam.mml(out_range_theta[[i]], # stimo il modello tenendo
                                # gli item fissi ma selezionando
                                # solo i parametri degli item selezionati
                                xsi.fixed =
                                cbind(1:ncol(out_range_theta[[i]]),
                                      diff_true[as.integer(gsub("I00|I0|I",
                                                                '',
                                                                colnames(out_range_theta[[i]]))),
2]),
                                B =
                                array(c(rep(0, ncol(out_range_theta[[i]])),
                                      discr_true[,2,][as.integer(gsub("I00|I0|
I",
                                                                "",
                                                                colnames(out_range_theta
[[i]])))]),
                                c(ncol(out_range_theta[[i]]),2,1),
                                dimnames = list(colnames(out_range_theta
[[i]]),
                                                c("Cat0", "Cat1"),
                                                "Dim01")))
  info_out_range_theta[[i]] <- IRT.informationCurves(model_out_range_theta[[i]],
                                theta = seq(-3, 3, length = 1000))
  names(info_out_range_theta)[[i]] <- unique(max_temp$range_name)[i]
}

```

## Calcolo info nuovi modelli e reliability

(Stesso codice per i modelli stimati tenendo fissi i parametri degli item)

```

info_summary_range <- NULL
temp <- NULL
for(i in 1:length(info_out_range)) {
  temp <- data.frame(info_test = mean(info_out_range[[i]]$test_info_curve),

                    range_name = names(info_out_range)[[i]],
                    item = paste(colnames(out_range[[i]]), collapse = ","))

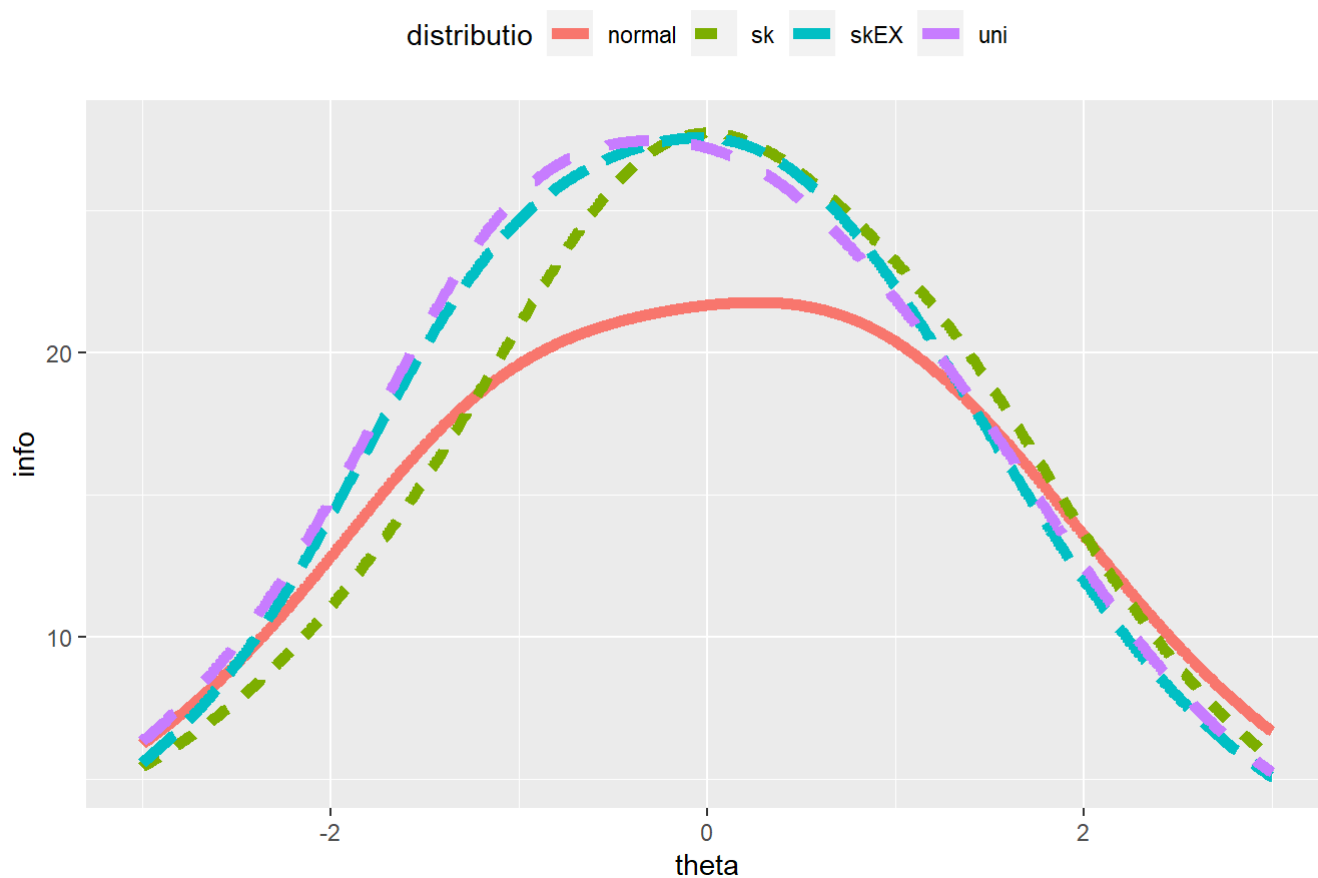
  info_summary_range <- rbind(info_summary_range,
                              temp)
}

info_summary_range$rel <- 1 - (1/sqrt(info_summary_range$info_test))^2
info_summary_range <- rbind(info_summary_range,
                            data.frame(info_test = sum(info_start),
                                      range_name = "all",
                                      item = "all",
                                      rel = 1 - (1/sqrt(info_start))^2))
info_summary_range$selection <- "guided"

```

## TIF con tutti gli item

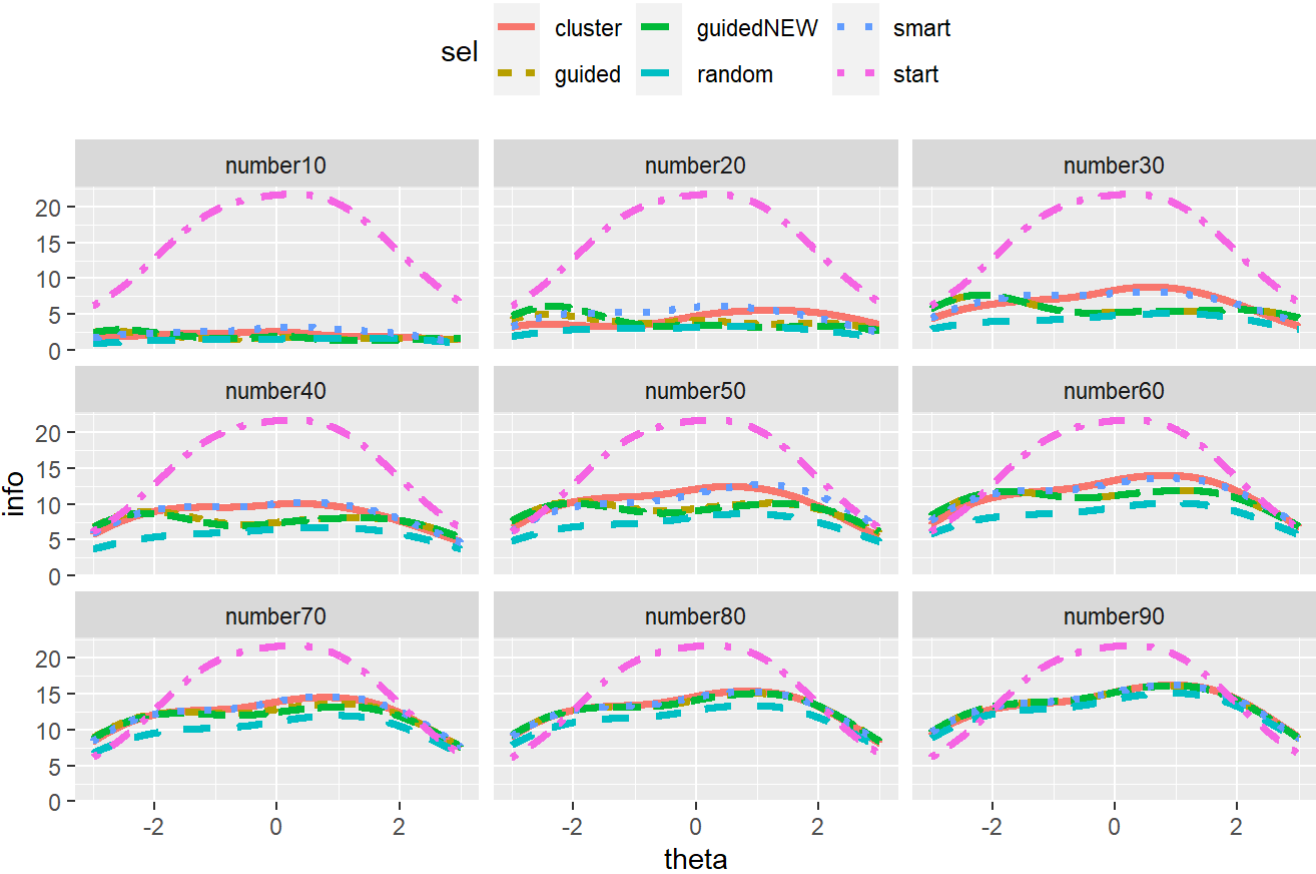
TIF-All item



## Tif gruppi di item

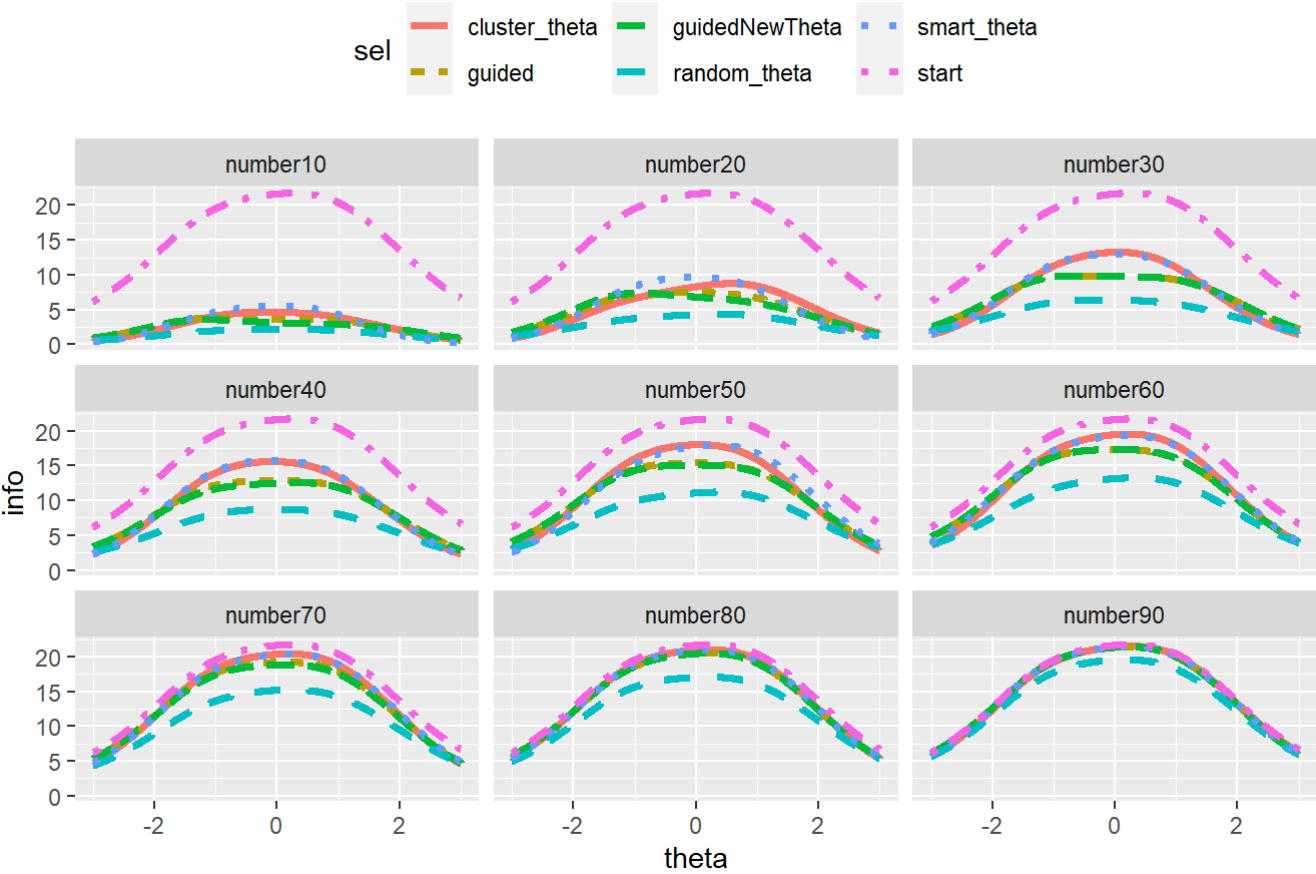
Normale Parametri liberi

Normale Parametri liberi



Normale parametri fissi:

Normale Parametri fissi

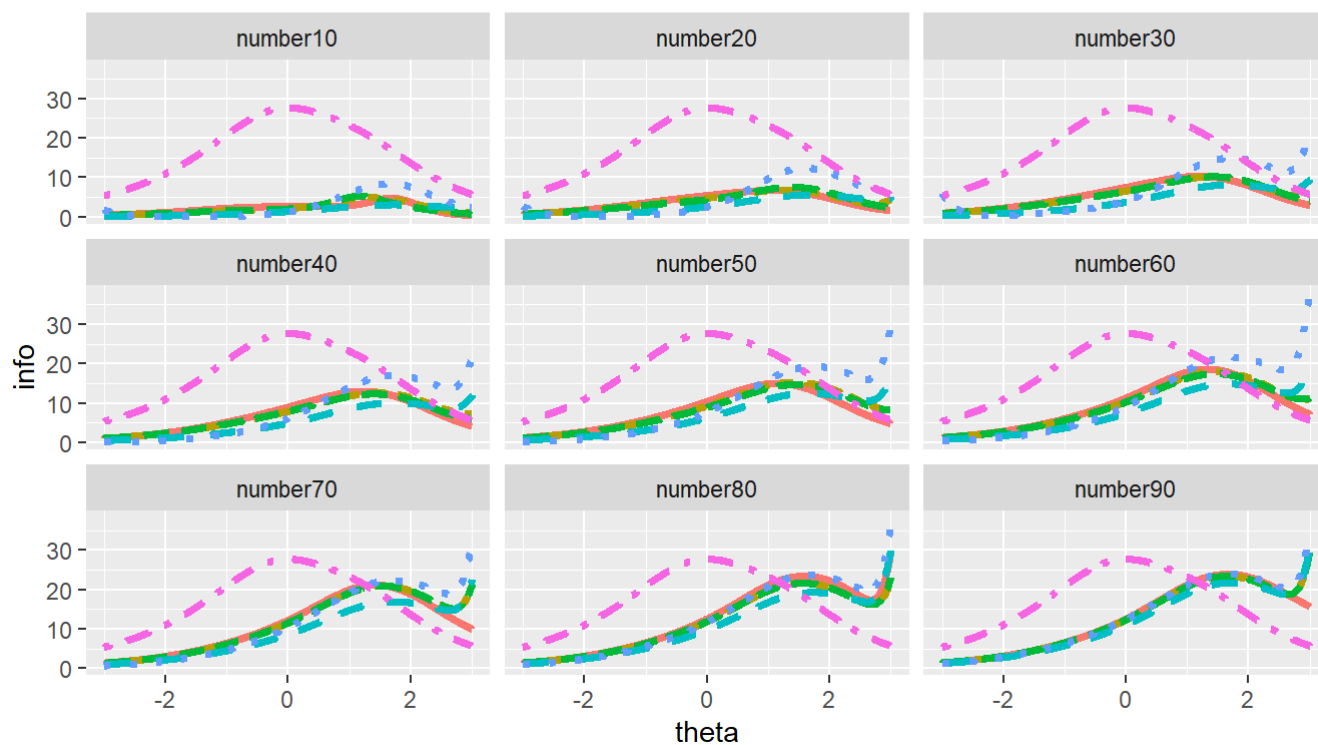


Skewness parametri liberi

## Sk Parametri liberi

sel

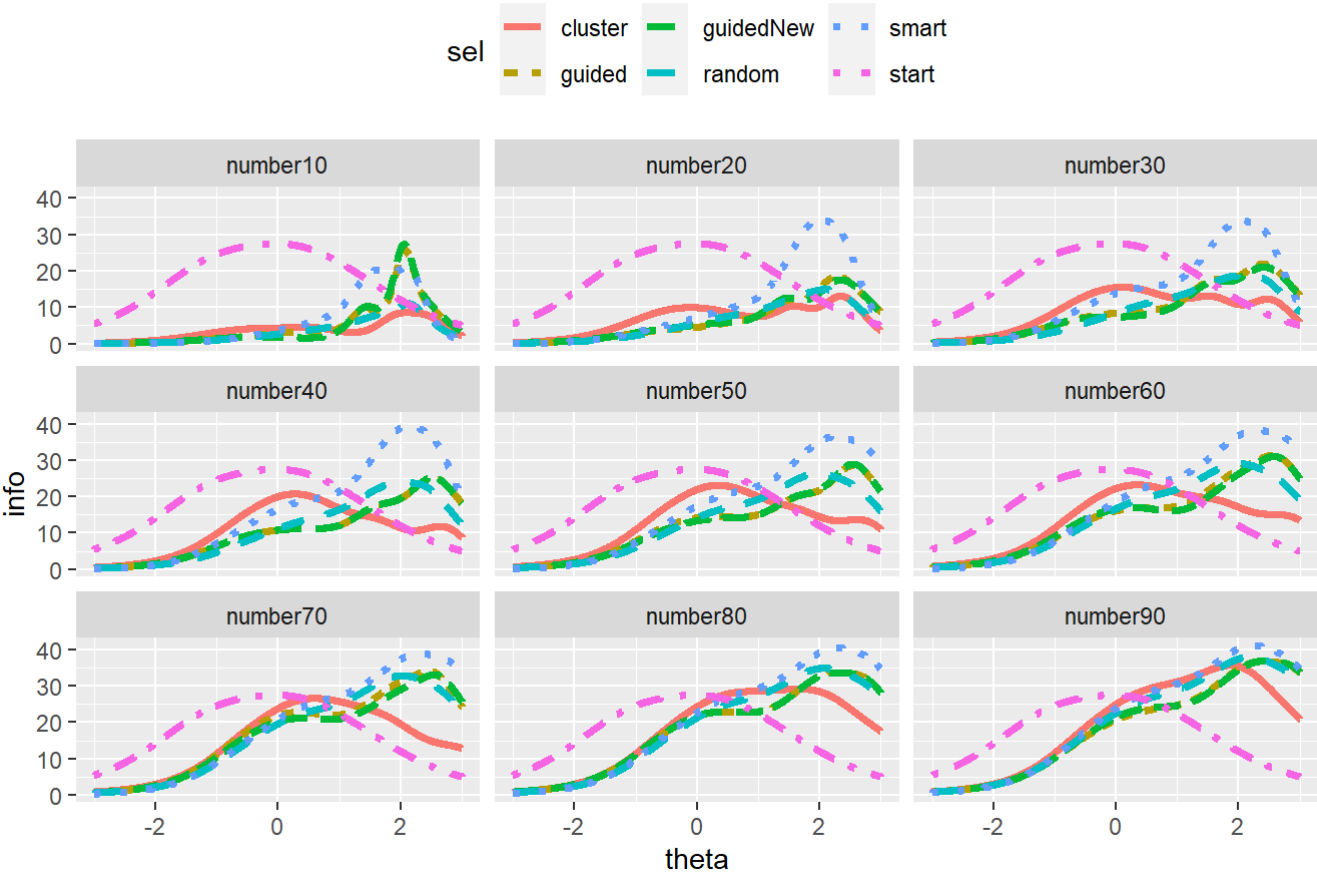
|         |           |       |
|---------|-----------|-------|
| cluster | guidedNew | smart |
| guided  | random    | start |



## Parametri liberi SK estrema

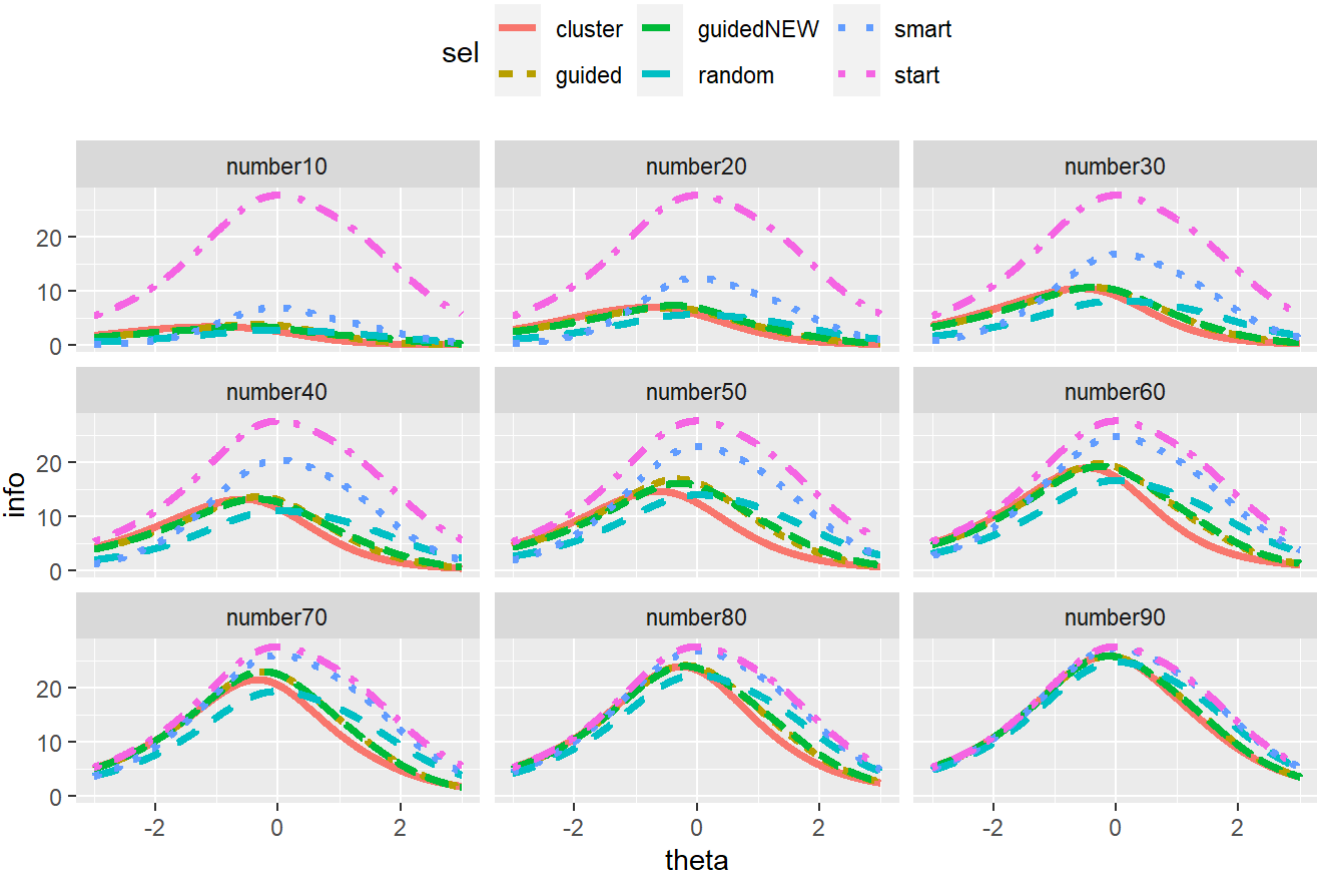
```
ggplot(temp_graph_sk_ex, aes(x = theta, y = info, group = sel,
  col = sel)) + geom_line(aes(linetype = sel), lwd = 1.4) +
  facet_wrap(~num_item) + theme(legend.position = "top") +
  ggtitle("Sk estrema Parametri liberi")
```

Sk estrema Parametri liberi



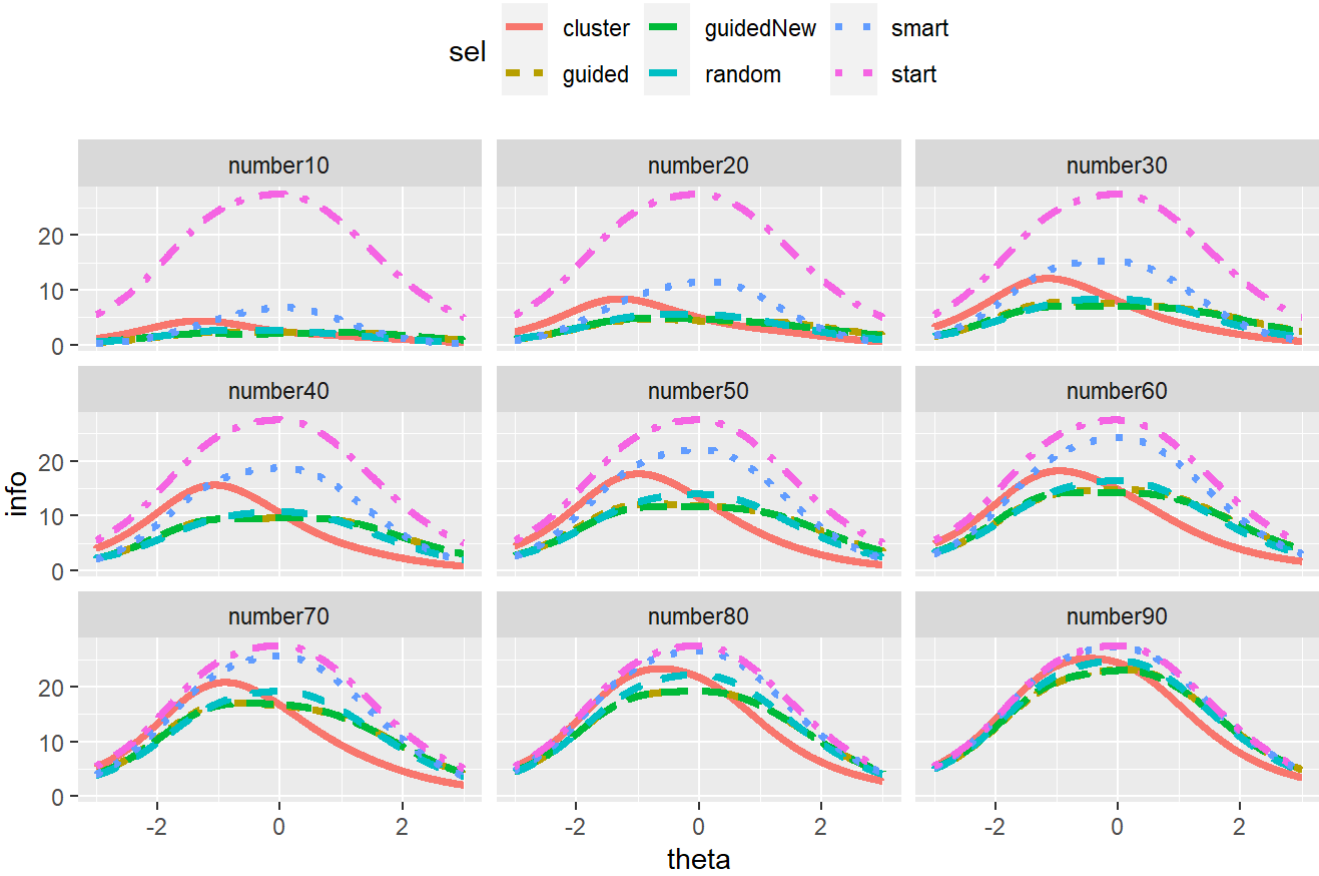
Parametri fissi skenweness

Sk estrema Parametri fissi



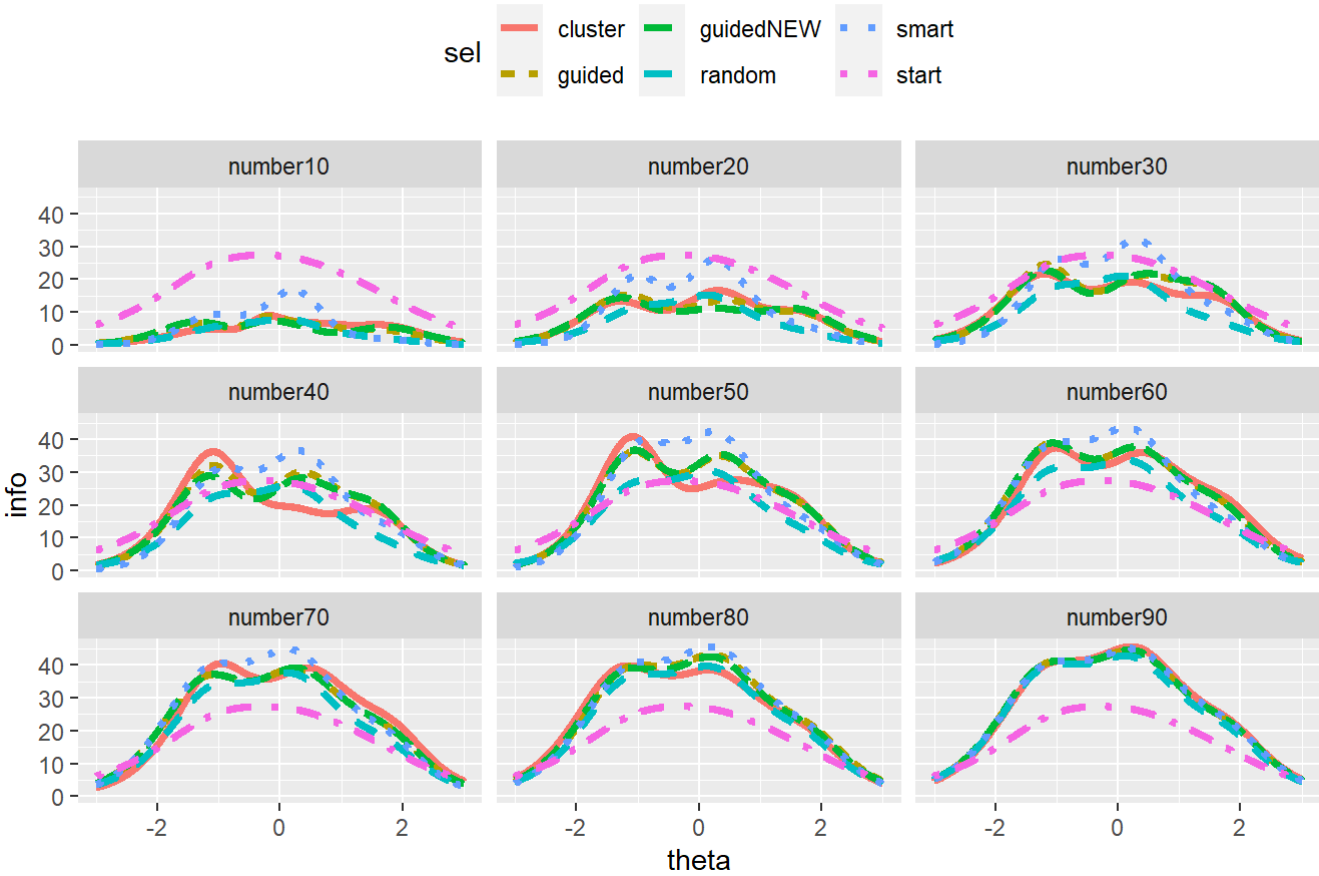
Parametri fissi skewness estrema

Sk Parametri fissi



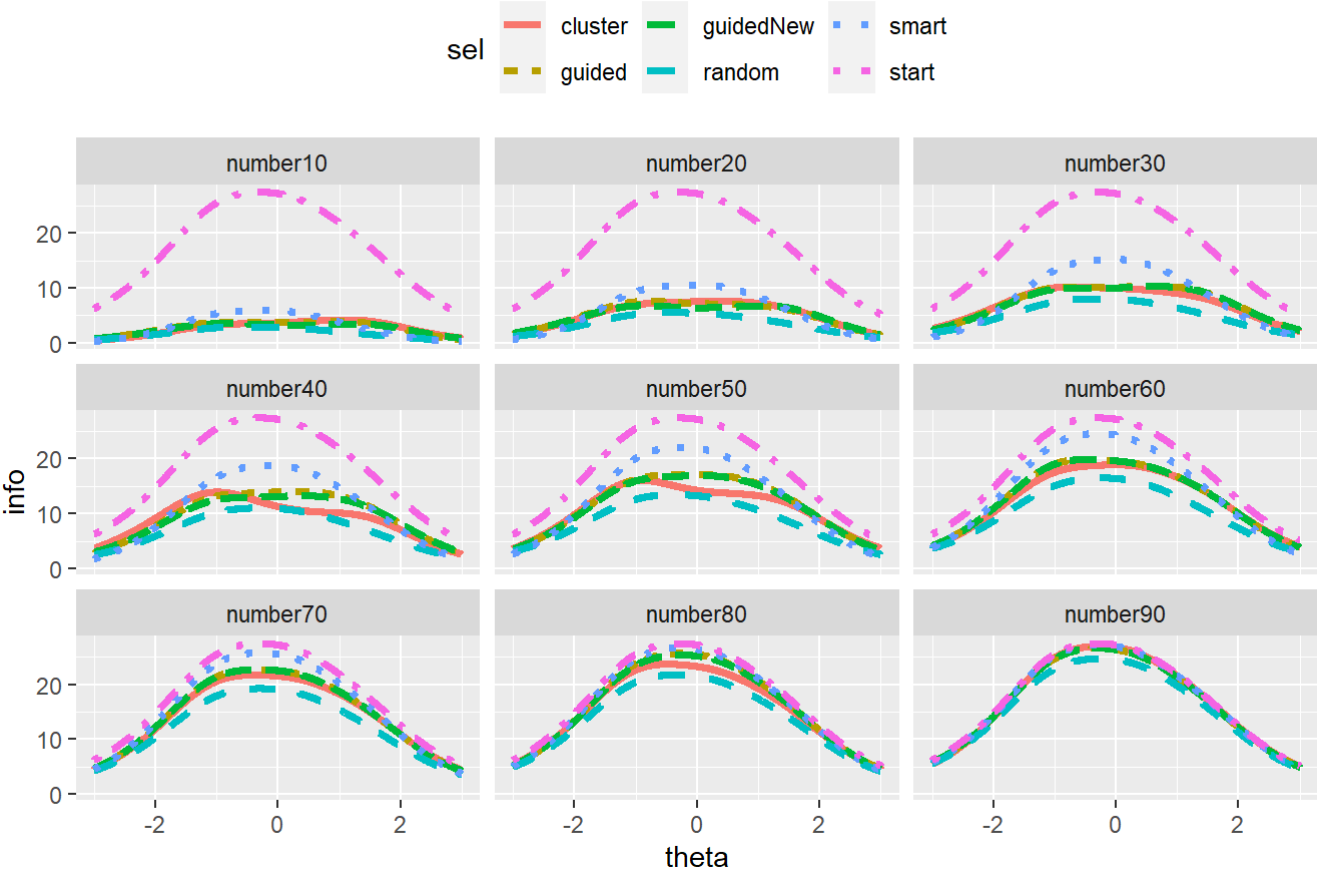
Uniforme parametri liberi

Uniforme Parametri liberi

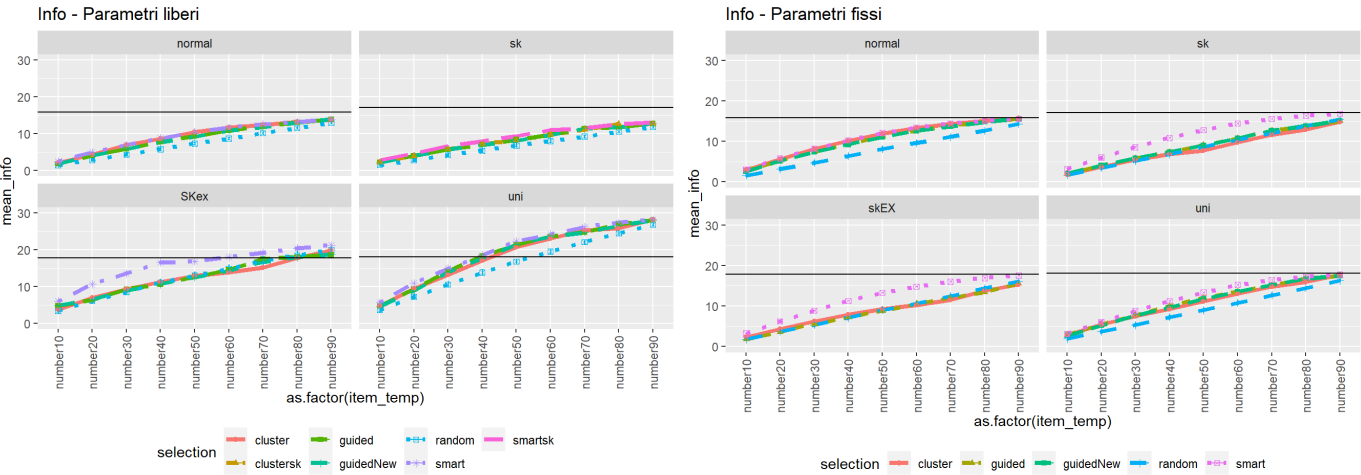


Parametri fissi uniforme

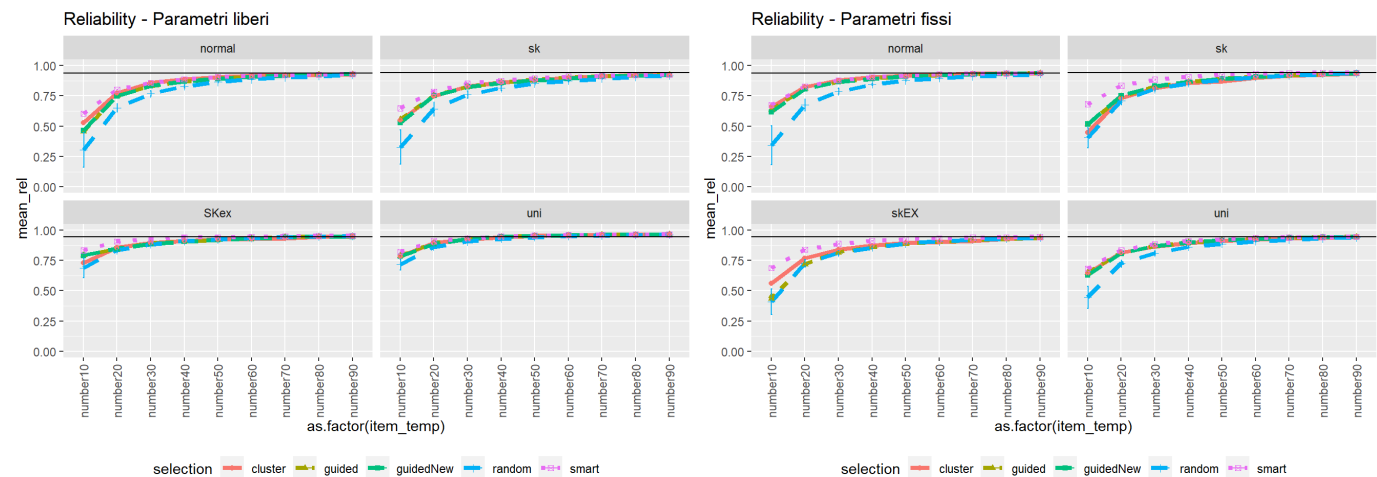
Uniforme Parametri fissi



Informazione

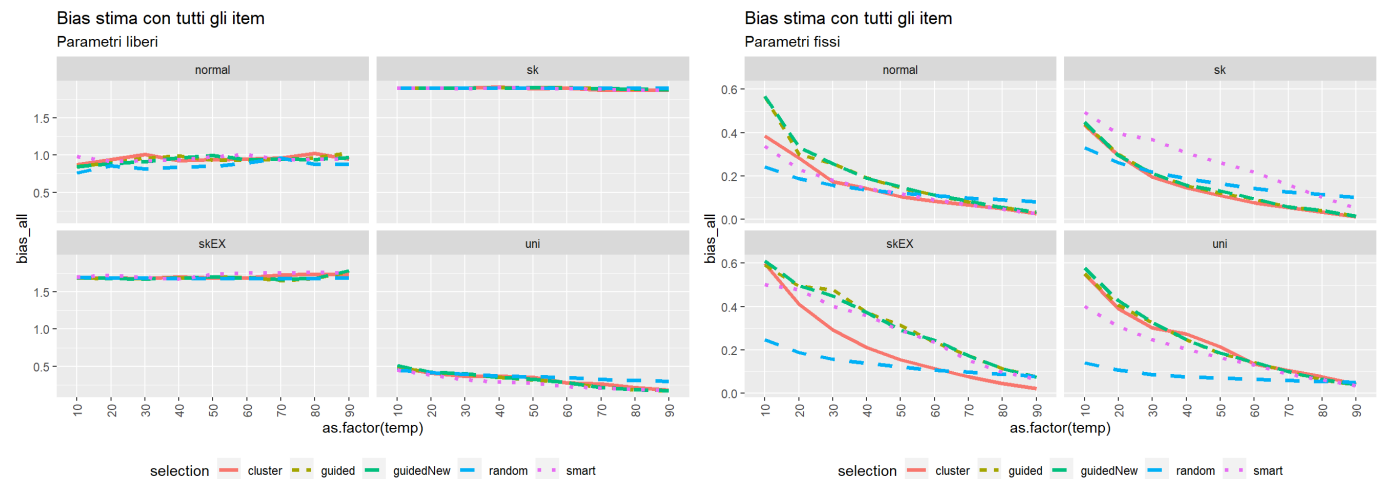


Reliability



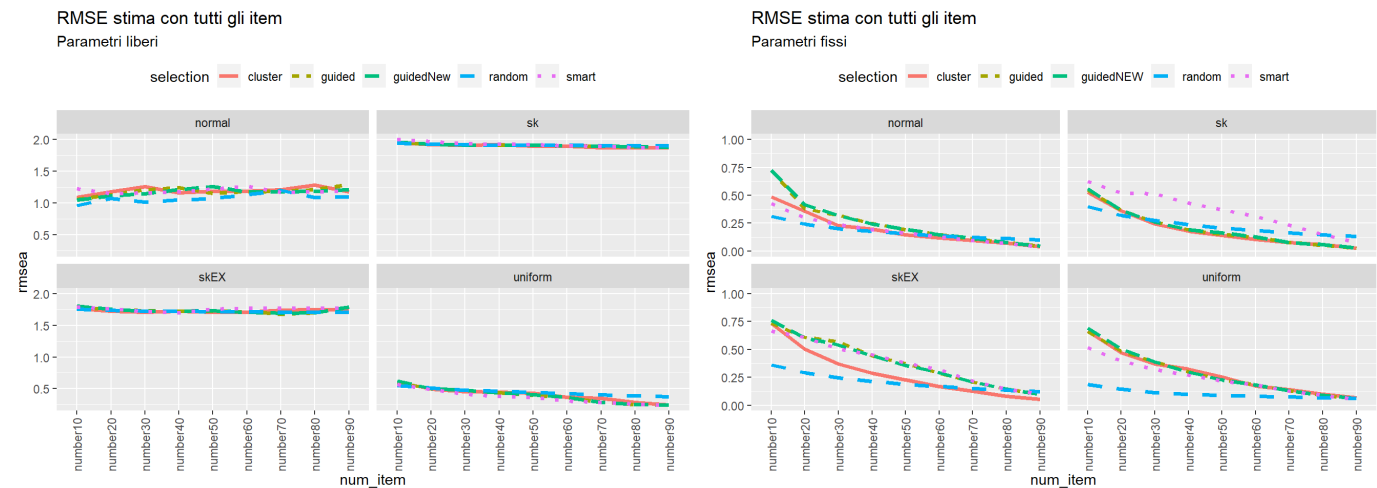
# Bias assoluto

La differenza è calcolata tra i  $\theta$  calcolati per ognuno dei diversi nuovi nuovi test e i  $\theta$  stimati sul modello con tutti gli item.



# RMSE

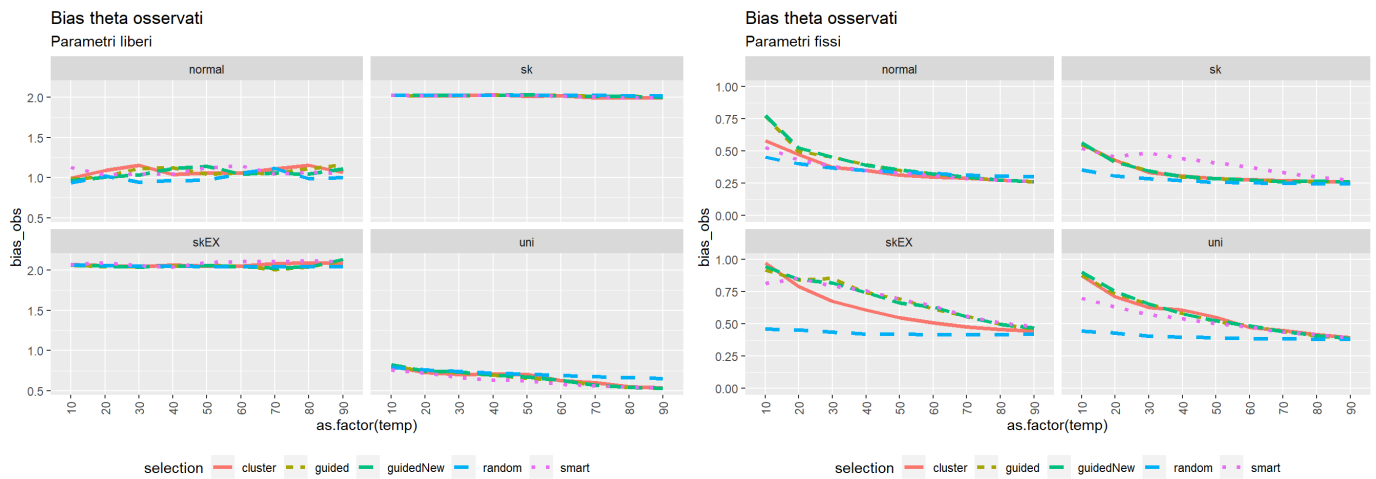
La differenza è calcolata tra i  $\theta$  calcolati per ognuno dei diversi nuovi nuovi test e i  $\theta$  stimati sul modello con tutti gli item.



# Bias assoluto valori osservati

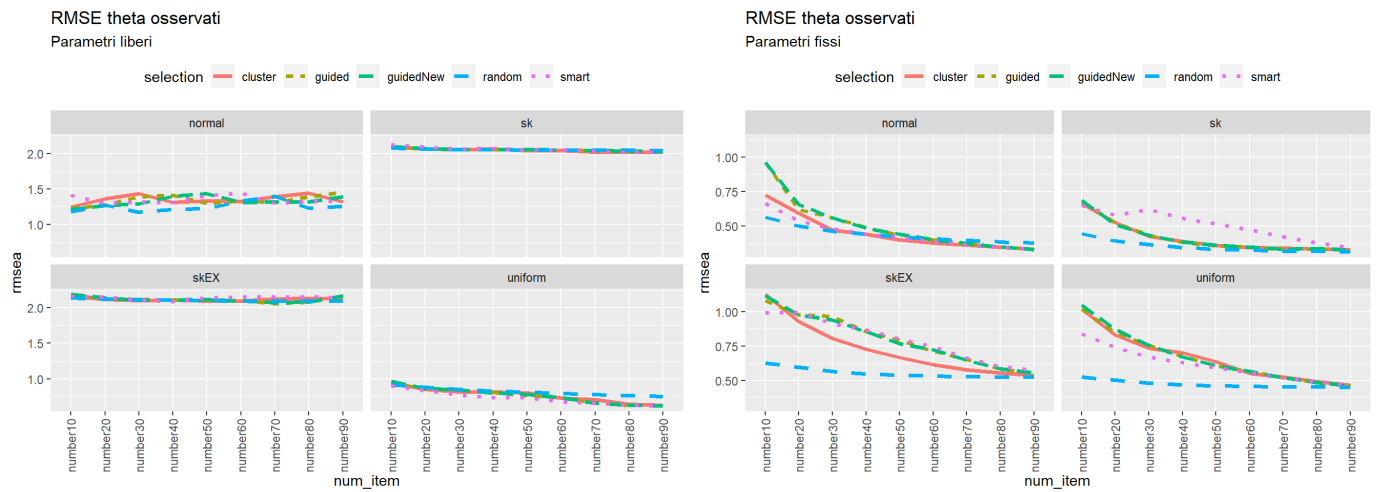


La differenza è calcolata tra i  $\theta$  calcolati per ognuno dei diversi nuovi nuovi test e i  $\theta$  “veri” (quelli simulati all’inizio, che quindi possono avere distribuzione normale, con skewness o uniforme).



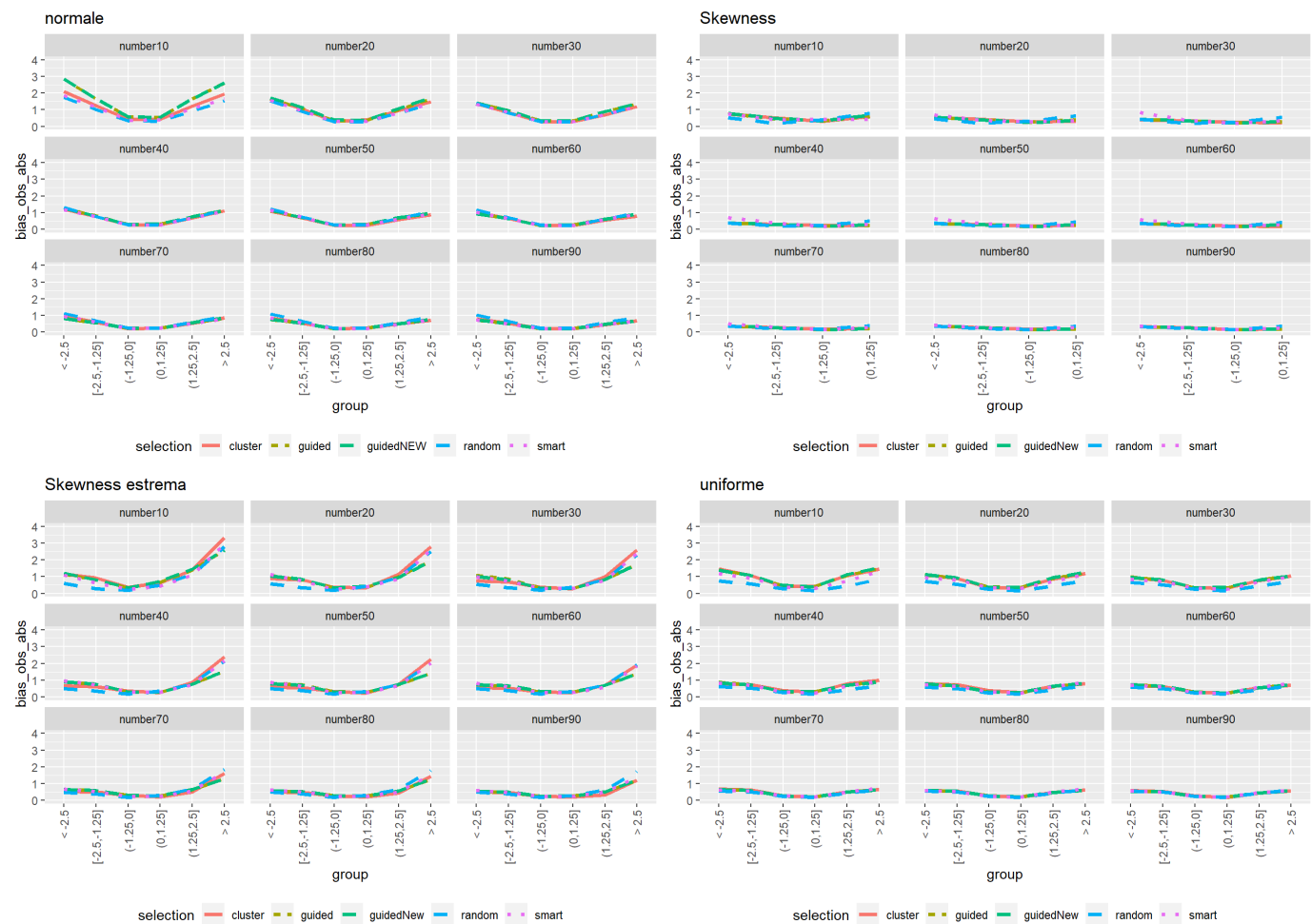
## RMSE valori osservati

La differenza è calcolata tra i  $\theta$  calcolati per ognuno dei diversi nuovi nuovi test e i  $\theta$  “veri” (quelli simulati all’inizio, che quindi possono avere distribuzione normale, con skewness o uniforme).



## Bias assoluto per gruppi di theta

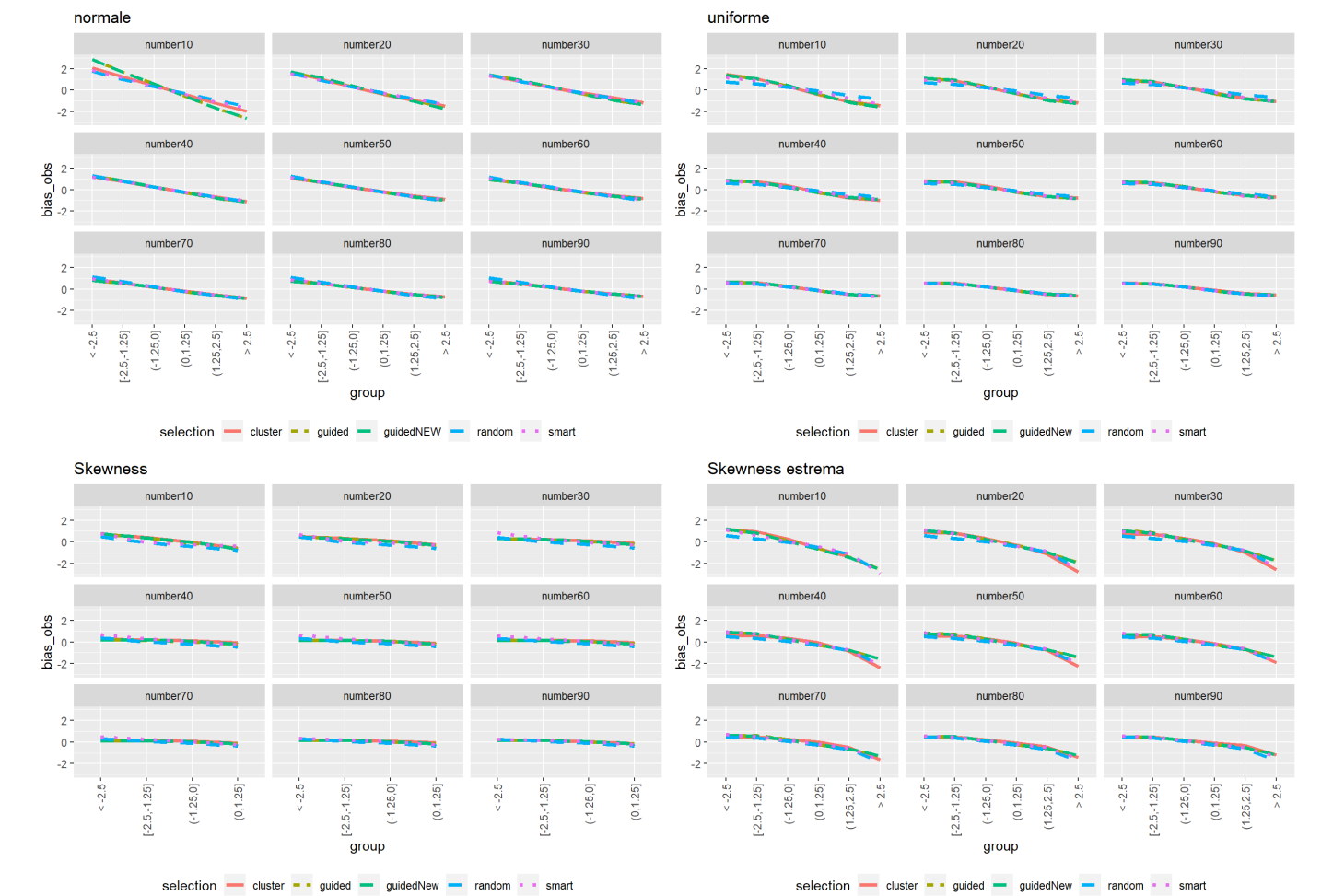
La differenza è calcolata tra i  $\theta$  calcolati per ognuno dei diversi nuovi nuovi test e i  $\theta$  “veri” (quelli simulati all’inizio, che quindi possono avere distribuzione normale, con skewness o uniforme).



# Bias per gruppi di theta

La differenza è calcolata tra i  $\theta$  calcolati per ognuno dei diversi nuovi nuovi test e i  $\theta$  “veri” (quelli simulati all’inizio, che quindi possono avere distribuzione normale, con skewness o uniforme).

A differenza del punto precedente, il bias non ha il valore assoluto. Essendo  $\hat{\theta} - \theta$ , valori positivi indicano sovrastima, valori negativi indicano sottostima.



# RMSE per gruppi di theta

