

Figures for ‘A Primer on Visualizations for Comparing Populations. . .’

Jerzy Wiecezorek

2021-02-06

This vignette shows how to reproduce the main figures in “A Primer on Visualizations for Comparing Populations, Including the Issue of Overlapping Confidence Intervals” (Wright, Klein, and Wiecezorek, 2019, *The American Statistician*).

Note: For this vignette itself, we automatically save the figures below using the `knitr` package with option `dev="tikz"` instead of saving them individually. In the final section of the vignette, we show an example of how to save individual plots using the `tikz()` function in the `tikzDevice` package.

Workflow to reproduce figures from the article

First, we load the package and the `TravelTime2011` dataset used in the paper. We also create string versions of our estimates and their standard errors that will print with a consistent number of digits.

```
library(RankingProject)
data(TravelTime2011)
USdata <- TravelTime2011
head(USdata)
```

##	Rank	State	Estimate.2dec	SE.2dec	Abbreviation	Region	FIPS
## 1	1	South Dakota	16.86	0.28	SD	MIDWEST	46
## 2	2	North Dakota	16.91	0.36	ND	MIDWEST	38
## 3	3	Nebraska	18.06	0.19	NE	MIDWEST	31
## 4	4	Wyoming	18.10	0.50	WY	WEST	56
## 5	5	Montana	18.18	0.32	MT	WEST	30
## 6	6	Alaska	18.39	0.33	AK	PACIFIC	2

```
# Format estimates and SEs into strings with 2 digits past the decimal
USdata$Estimate.Print = formatC(USdata$Estimate.2dec,
                                format = 'f', digits = 2)
# For SEs, also drop the leading 0
USdata$SE.Print = substring(formatC(USdata$SE.2dec,
                                    format = 'f', digits = 2),
                             first = 2)
```

Next, we set up several list-type objects to contain parameters needed for the tables and plots. As in the article, we use Colorado (CO) as the reference state. The option `tikzText=TRUE` lets us use LaTeX-style text and symbols in the figures, instead of basic R-style text.

```
# Set Colorado as the reference state
refAbbr <- "CO"
refRow <- which(USdata$Abbreviation==refAbbr)

# Set up parameter lists for table function and figure function
```

```

tableParList <- with(USdata,
  list(ranks = Rank, names = Abbreviation,
    est = Estimate.Print, se = SE.Print,
    placeType = "State", tikzText = TRUE))
plotParList <- with(USdata,
  list(est = Estimate.2dec, se = SE.2dec,
    names = Abbreviation, refName = refAbbr,
    confLevel = .90, tikzText = TRUE))

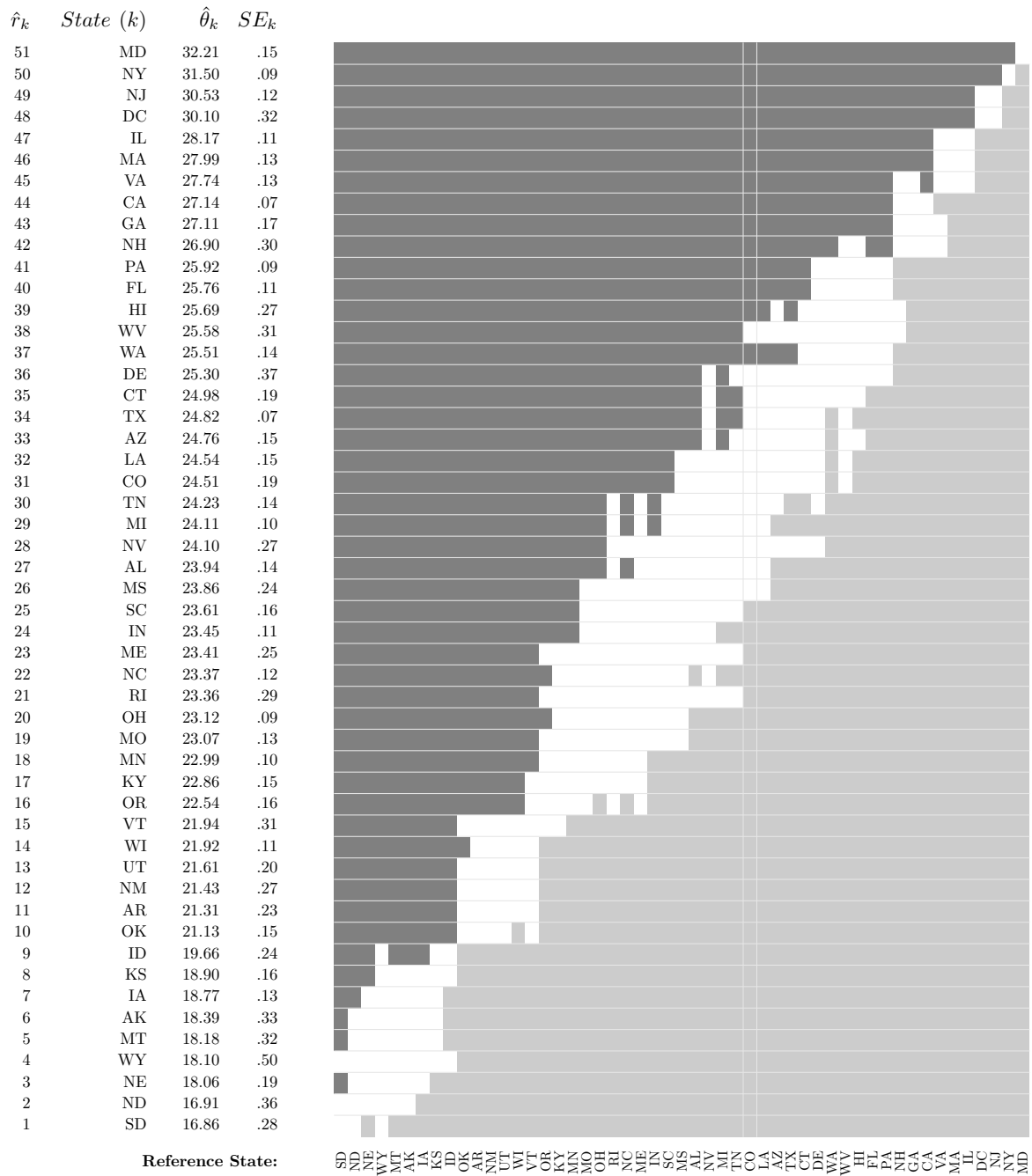
```

Reproduce Figure 3, the “shaded columns plot”:

```

# Shaded Columns plot
plotParList$plotType <- "columns"
# Specify where to position the "Reference State:" text,
# and adjust column widths from their defaults
tableParList = c(tableParList,
  list(columnsPlotRefLine = .7, col2 = .55, col3 = .8))
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,
  tableWidthProp = 2/7, tikzText = TRUE)

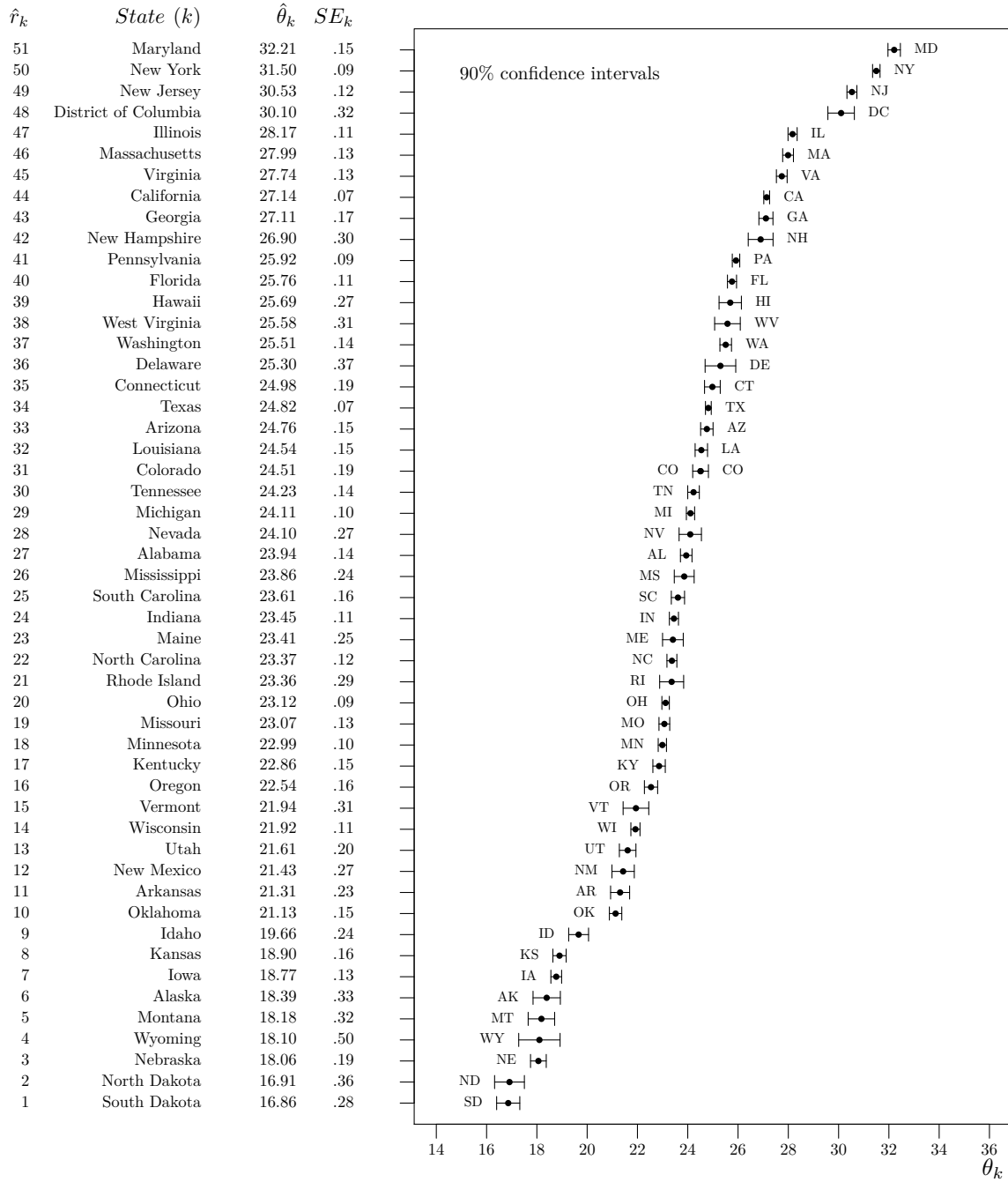
```



```
# Reset defaults for future plots
tableParList[c("columnsPlotRefLine", "col12", "col13")] <- NULL
```

Reproduce Figure 1, the plot of individual 90% confidence intervals (CIs):

```
# For all remaining figures,  
# table will show full state names instead of abbreviations  
tableParList$names <- USdata$State  
  
# Individual CIs  
plotParList$plotType <- "individual"  
plotParList$cex <- 0.6  
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,  
                  tikzText = TRUE)
```



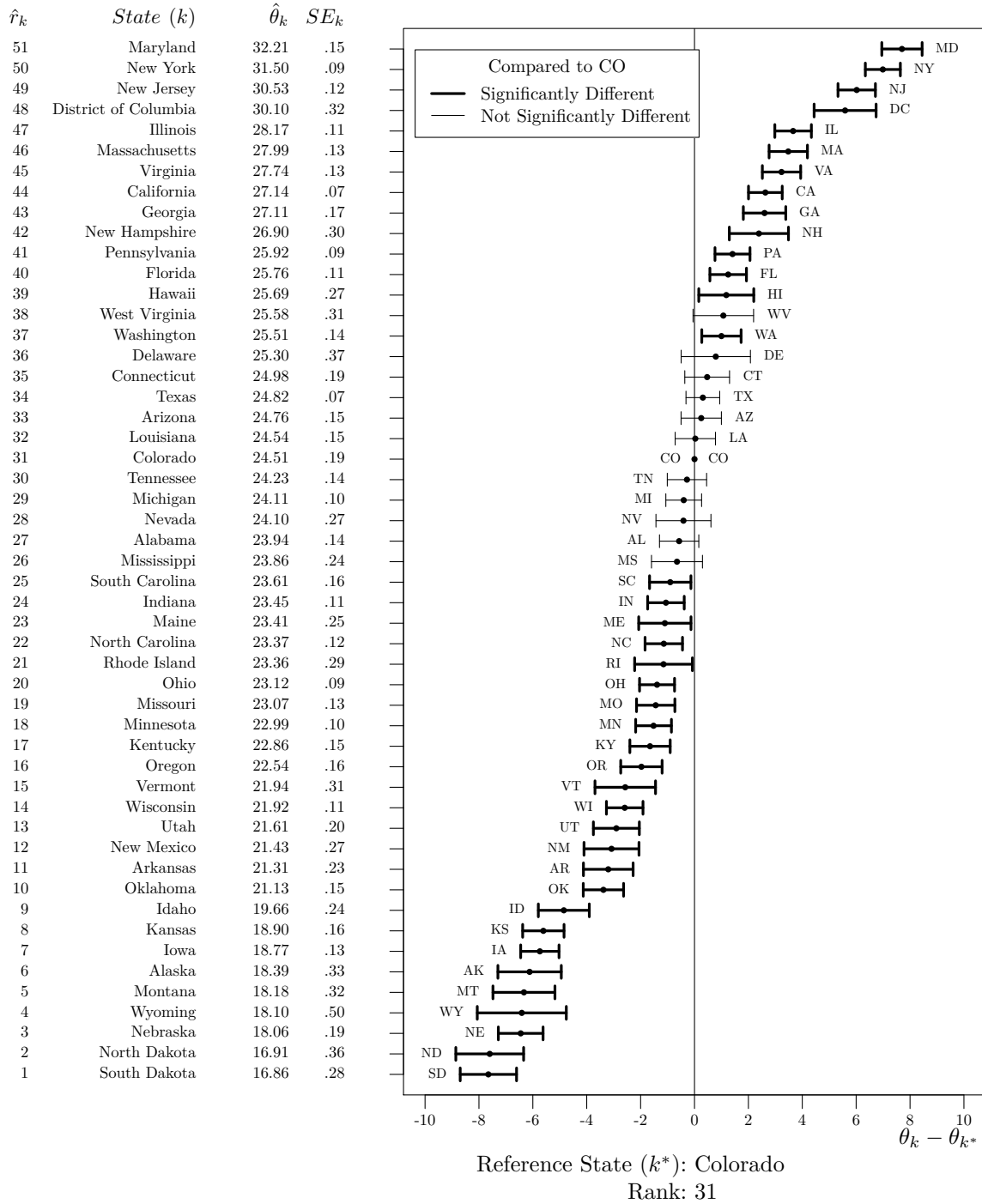
Reproduce Figure 4, the plot of demi-Bonferroni-corrected 90% CIs for the difference between the reference state Colorado and all other states:

```
# CIs for differences from ref
plotParList$plotType <- "difference"
```

```

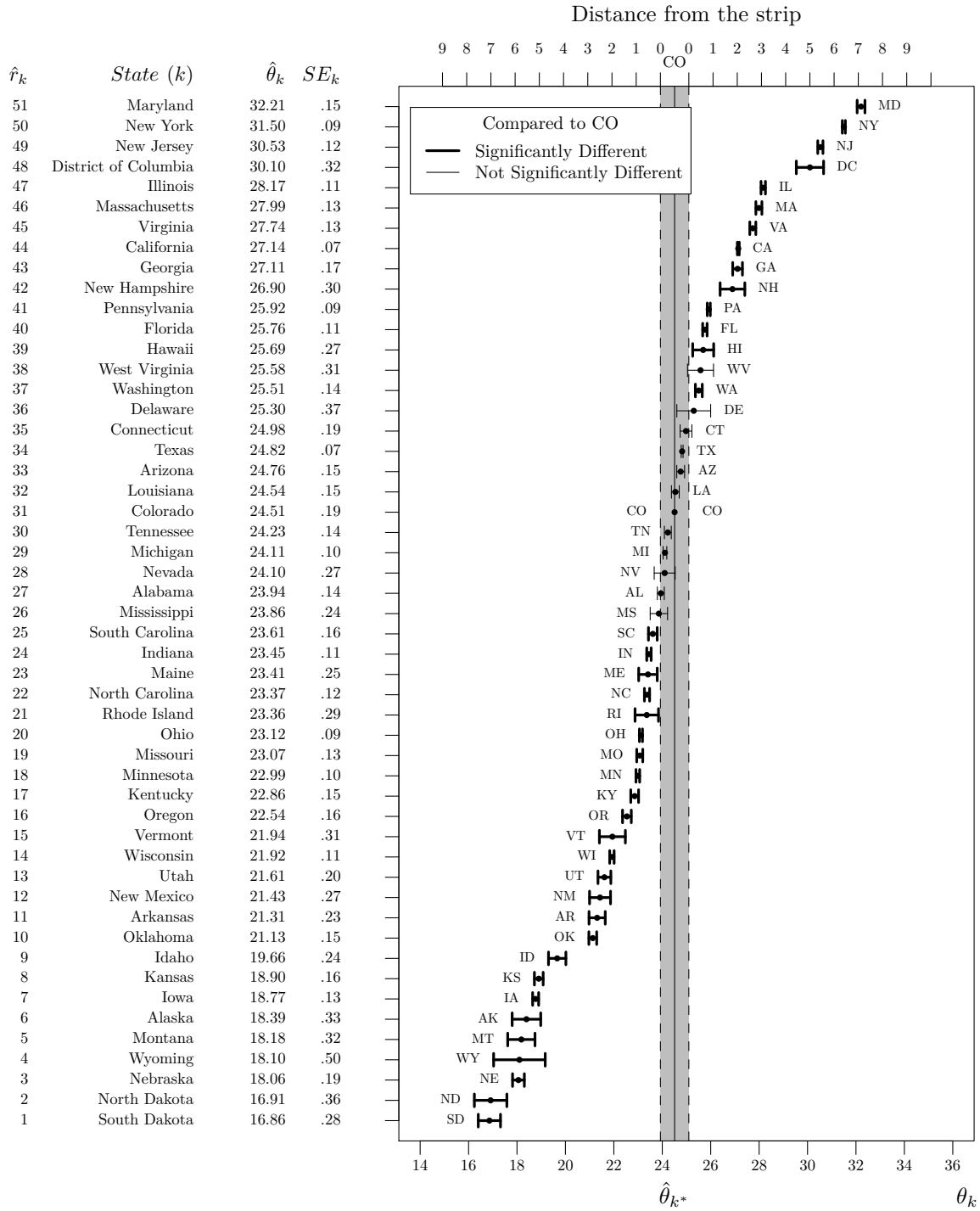
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,
  annotRefName = USdata$State[refRow],
  annotRefRank = USdata$Rank[refRow],
  tikzText = TRUE)

```



Reproduce Figure 7, the plot of demi-Bonferroni-corrected 90% “comparison intervals” (based on Almond et al., 2000) for comparing the reference state Colorado to all other states:

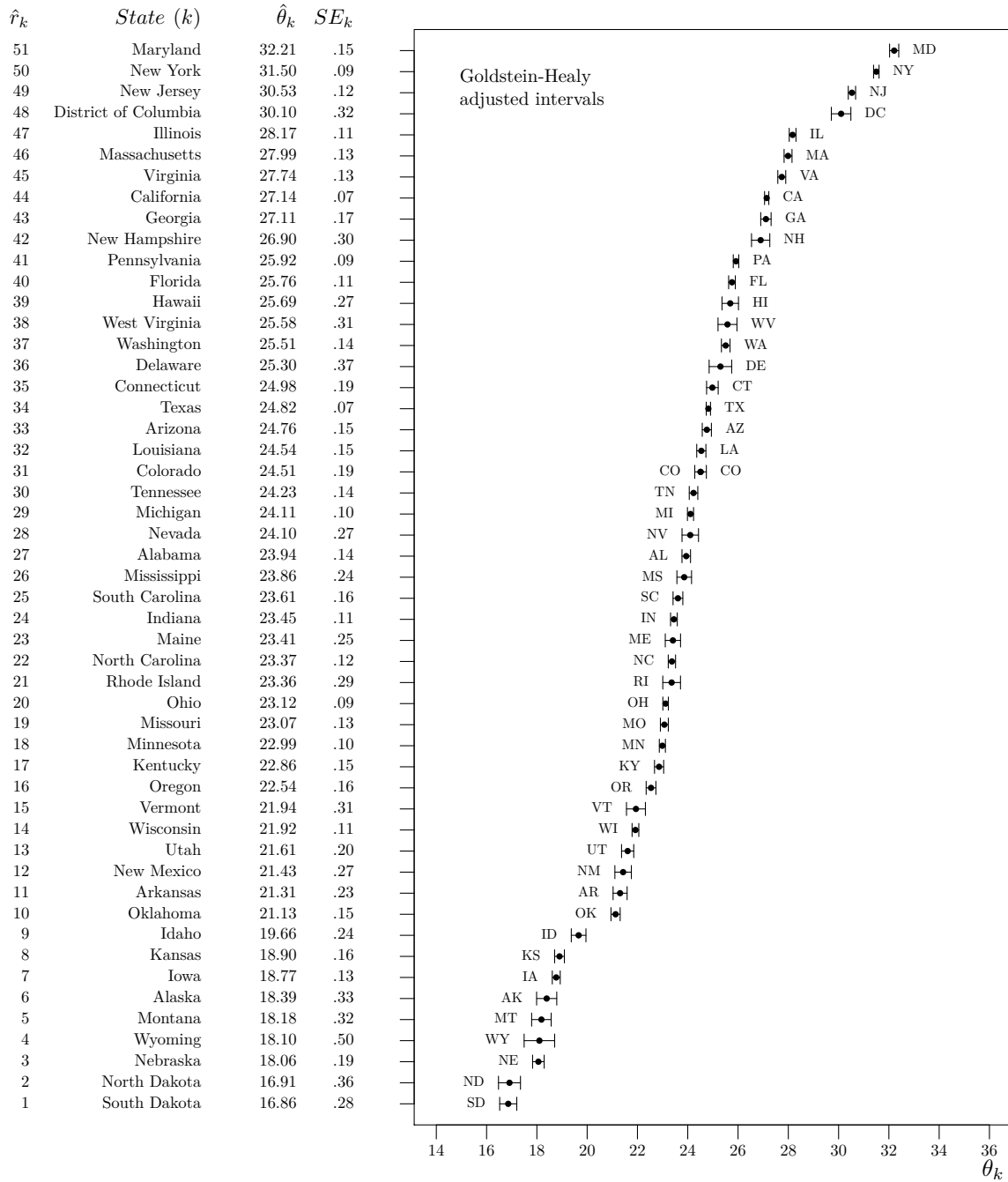
```
# Comparison intervals
plotParList$plotType <- "comparison"
plotParList$thetaLine <- 1.5
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,
                  tikzText = TRUE)
```



```
plotParList$thetaLine <- NULL
```

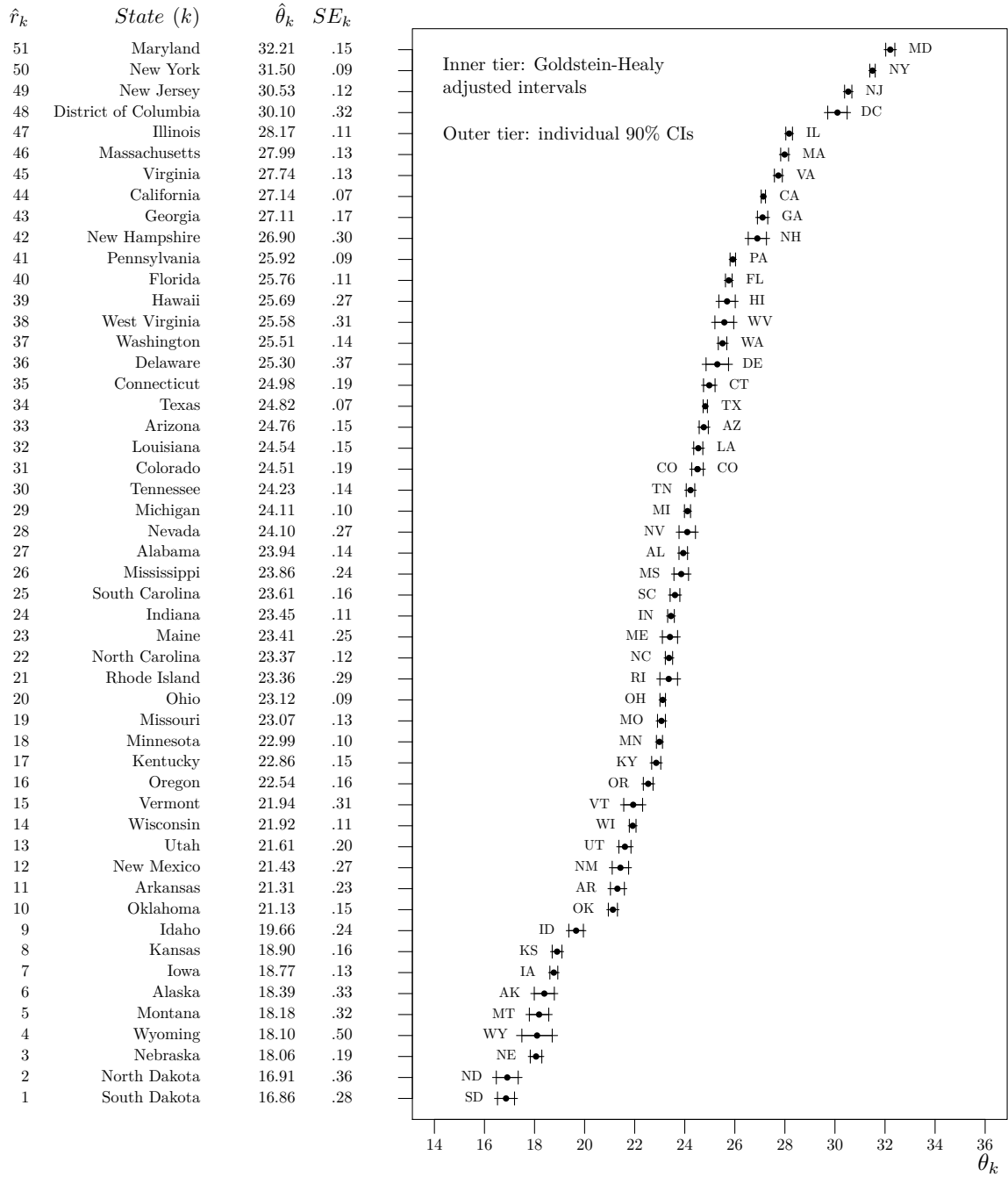
Reproduce Figure 10, the plot of Goldstein-Healy-adjusted “90%” CIs (based on Goldstein and Healy, 1995), which are in fact 77.49% CIs as chosen to achieve an “average significance level” of $\alpha = 0.10$:


```
# Goldstein-Healy adjusted CIs
plotParList$plotType <- "individual"
plotParList$GH <- TRUE
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,
                  tikzText = TRUE)
```



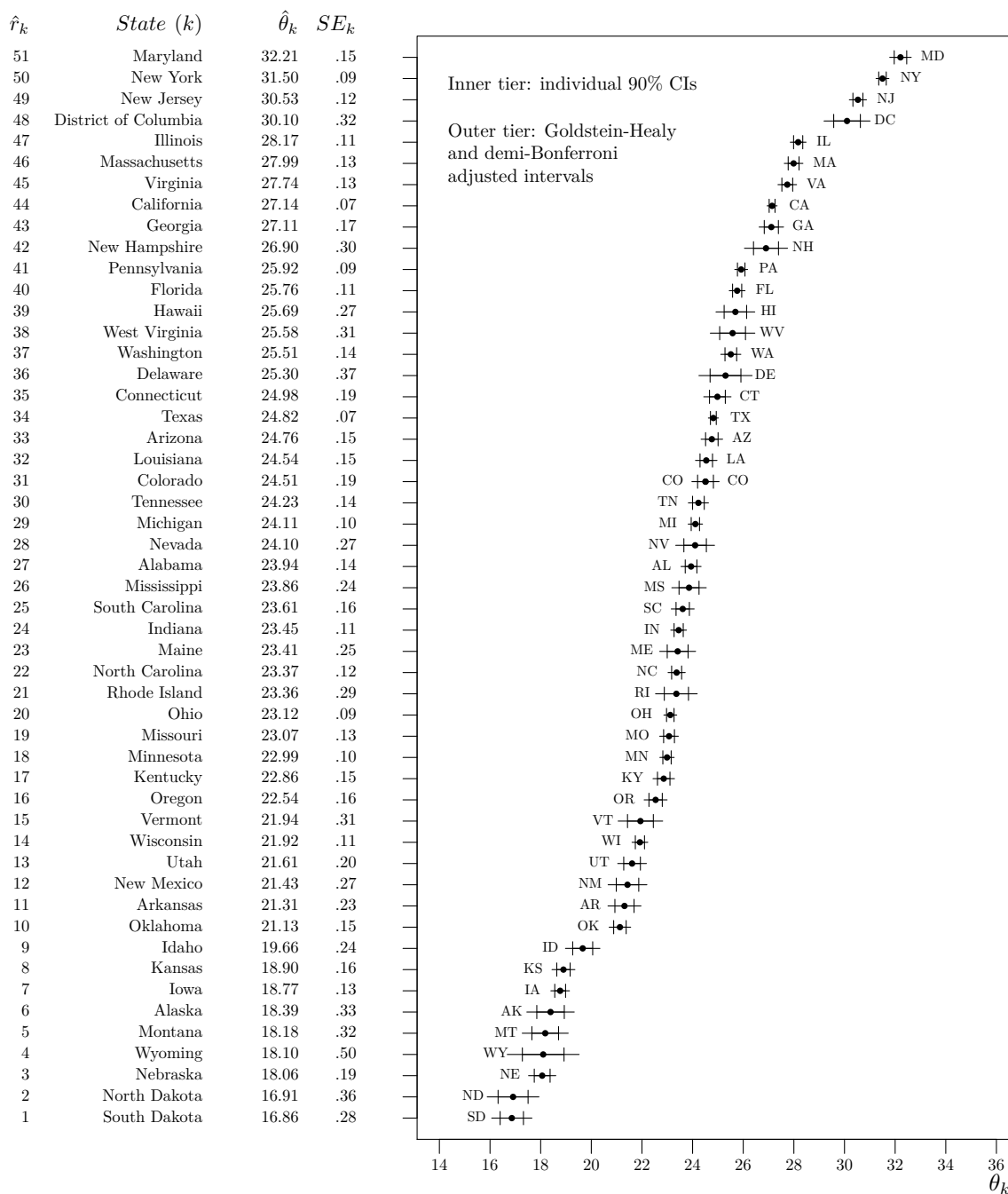
Reproduce Figure 11, the plot of two-tiered error bars, where the inner tier are the same Goldstein-Healy-adjusted “90%” CIs from Figure 10, and the outer tier are the same individual 90% CIs from Figure 1:

```
# Double-tiered GH plot:
# inner tiers are GH CIs,
# outer tiers are usual 90% CIs
plotParList$tiers <- 2
# Legend auto-positioning is poor with line breaks in legend text;
# we can improve it by controlling (X,Y) manually
plotParList$legendX <- 13
plotParList$legendY <- 52
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,
                  tikzText = TRUE)
```



Reproduce Figure 12, another plot of two-tiered error bars, where now the inner tier are the same individual 90% CIs from Figure 1, and the outer tier are demi-Bonferroni-corrected Goldstein-Healy-adjusted “90%” CIs (in fact 99.55% CIs):

```
# Double-tiered GH + Bonferroni plot:  
# inner tiers are usual 90% CIs,  
# outer tiers are 50-way demi-Bonferroni-corrected GH CIs  
plotParList$Bonferroni <- "demi"  
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,  
                  tikzText = TRUE)
```



Using `tikzDevice::tikz()` to save individual figures

For this vignette, the figures above were automatically converted to PDF format using `knitr` with chunk option `dev="tikz"`. When not using `knitr`, we may prefer to save plots one at a time “manually.” To do this,

we can explicitly call the `tikz()` function from the `tikzDevice` package, as in the following example code. The `tikz()` function works much like `pdf()` or `png()` and other standard functions for saving plots from R scripts. We must remember to call `dev.off()` after the plotting function runs, to let R know the plot is ready to be saved.

Using `tikz()` will create and save a `.tex` file. To convert this to a figure, we can:

- compile it into a standalone PDF separately; or
- use R's `tools::texi2pdf()` which compiles the PDF and saves it in the current working directory; or
- set `standAlone=FALSE` below, then copy-paste the contents of the saved `.tex` file directly into a larger `.tex` document.

```
# Not run:
library(tikzDevice)
tikz("/path/to/my/file.tex", standAlone = TRUE, width = 6.5, height = 8)
RankPlotWithTable(tableParList = tableParList, plotParList = plotParList,
                  tikzText = TRUE)
dev.off()
tools::texi2pdf("/path/to/my/file.tex")
```