

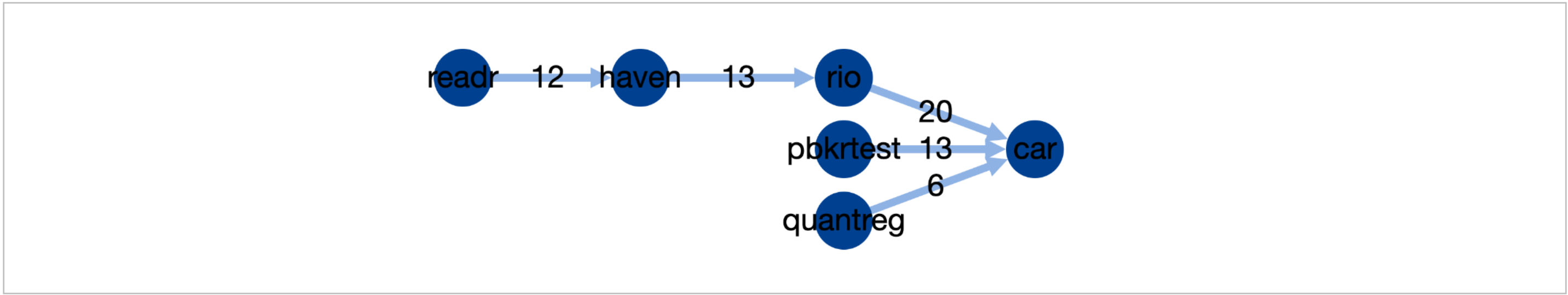
A

Package	Repository	Number of strong dependency packages	Number of all dependency packages	Number of parent packages	Max heaviness from parent packages	Heaviness on child packages	Number of child packages	Heaviness on downstream packages	Number of downstream packages
Rcmdr	CRAN	130	150	14	35	95.9	47	95.5	48
Seurat	CRAN	140	252	50	reducible 12	84.7	31	82.5	33
RTCGA	Bioconductor	134	195	15	67	135.0	9	135.0	9
lumi	Bioconductor	159	167	19	51	111.6	13	82.0	18
brms	CRAN	181	230	25	reducible 50	125.8	8	125.8	8
minfi	Bioconductor	138	159	34	25	57.7	38	66.5	59
taxize	CRAN	127	145	24	55	77.8	14	78.3	15
car	CRAN	83	108	15	20	45.9	183	42.7	587
survminer	CRAN	122	156	18	47	61.6	24	63.1	33
tidyverse	CRAN	106	118	29	reducible 17	49.3	74	47.5	86
smacof	CRAN	131	140	17	reducible 33	89.1	9	81.9	13
devtools	CRAN	76	117	23	12	46.8	77	44.8	90
caret	CRAN	80	210	16	reducible 26	42.6	177	41.9	197
GenomicScores	Bioconductor	98	153	17	26	55.6	25	56.0	26
FactoMineR	CRAN	112	148	15	55	46.4	51	38.2	85
ggpubr	CRAN	103	103	19	reducible 52	42.1	111	37.2	185
WGCNA	CRAN	108	113	17	reducible 47	50.4	32	49.6	33

B

Upstream package	Dependency path	Dependency depth	Heaviness from upstream on car
rio	rio → car 20	2	20
pbkrtest	pbkrtest → car 13	2	13
haven	haven → rio → car 13   20	3	9
readr	readr → haven → rio → car 12   13   20	4	7
quantreg	quantreg → car 6	2	6

Dependency paths in the table are represented as the following graph with 6 nodes and 8 edges. [Reset graph](#) | [Vertical layout](#)



C

Adjust range of dependency depth: 

1

5

Depth = 1: 182 paths

Depth = 2: 219 paths

Depth = 3: 152 paths

Depth = 4: 36 paths

Depth = 5: 12 paths

Downstream package	Dependency path	Dependency depth	Heaviness of car on downstream package
DJL	car → DJL 84	2	84
DistatisR	car → DistatisR 84	2	84
GEWIST	car → GEWIST 84	2	84
GrimR	car → GrimR 84	2	84
GroupComparisons	car → GroupComparisons 84	2	84

D

Dependency paths in the table are represented as the following graph with 585 nodes and 601 edges. [Reset graph](#)

**Green dots:** Groups of leave nodes (number > 1) in the complete graph that connect to their parents nodes. It is for reducing the size of original graph.

**Red arrows:** Edges with high betweenness. Value on a edge is the heaviness from a parent on its child package.

