Assignment2_Oct4

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(lpSolveAPI)
Weigelt_linear.lp <- make.lp(11,9)

# obj function-max

set.objfn(Weigelt_linear.lp, c(420,360,300,420,360,300,420,360,300))
lp.control(Weigelt_linear.lp,sense='max')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
```

```
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```r
# storage
set.row(Weigelt_linear.lp, 1, c(20, 15, 12), indices = c(1, 2, 3))
set.row(Weigelt_linear.lp, 2, c(20, 15, 12), indices = c(4, 5, 6))
set.row(Weigelt_linear.lp, 3, c(20, 15, 12), indices = c(7, 8, 9))


# excess caapcity
set.row(Weigelt_linear.lp, 4, c(1, 1, 1), indices = c(1, 2, 3))
set.row(Weigelt_linear.lp, 5, c(1, 1, 1), indices = c(4, 5, 6))
set.row(Weigelt_linear.lp, 6, c(1, 1, 1), indices = c(7, 8, 9))

#sales
set.row(Weigelt_linear.lp, 7, c(1, 1, 1), indices = c(1, 4, 7))
set.row(Weigelt_linear.lp, 8, c(1, 1, 1), indices = c(2, 5, 8))
set.row(Weigelt_linear.lp, 9, c(1, 1, 1), indices = c(3, 6, 9))
```

```
#Percent

set.row(Weigelt_linear.lp,10, c(0.0013,0.0013,0.0013,-0.0011,-0.0011,-0.0011),indices = c(1,2,3,4,5,6))
set.row(Weigelt_linear.lp,11, c(0.0013,0.0013,0.0013,-0.0022,-0.0022,-0.0022),indices = c(1,2,3,7,8,9))

# rhs and signs
rhs <- c(13000, 12000, 5000, 750, 900, 450, 900, 1200, 750,0,0)
set.rhs(Weigelt_linear.lp, rhs)
set.constr.type(Weigelt_linear.lp, c("<=","<=","<=","<=","<=","<=","<=","<=","<=","=","="))


# names
varname <- c("P1L", "P1M", "P1S", "P2L", "P2M", "P2S","P3L", "P3M","P3S")
constrname <- c("Storage1","Storage2","Storage3","EC1", "EC2", "EC3", "Sales1", "Sales2","Sales3","Perc


Weigelt_linear.lp
```

```
## Model name:
##   a linear program with 9 decision variables and 11 constraints
```

```
solve(Weigelt_linear.lp)
```

```
## [1] 0
```

```
get.objective(Weigelt_linear.lp)
```

```
## [1] 697923.1
```

```
solution <- data.frame(varname, get.variables(Weigelt_linear.lp))
colnames(solution) <- c("variable", "value")
solution
```

```
##   variable    value
## 1      P1L 484.6154
## 2      P1M 220.5128
## 3      P1S   0.0000
## 4      P2L   0.0000
## 5      P2M 666.6667
## 6      P2S 166.6667
## 7      P3L   0.0000
## 8      P3M   0.0000
## 9      P3S 416.6667
```

Alternative method

You can also embed plots, for example:

```
#Alternative method
library(lpSolveAPI)
Weigelt.lp.alt <- read.lp("Weigelt.lp")
Weigelt.lp.alt
```

```
## Model name:
##   a linear program with 9 decision variables and 11 constraints
```

```
solve(Weigelt.lp.alt)
```

```
## [1] 0
```

```
get.variables(Weigelt.lp.alt)
```

```
## [1] 484.6154 220.5128   0.0000   0.0000 666.6667 166.6667   0.0000   0.0000
## [9] 416.6667
```

```
get.objective(Weigelt.lp.alt)
```

```
## [1] 697923.1
```

```
get.constraints(Weigelt.lp.alt)
```

```
##  [1] 13000.0000 12000.0000  5000.0000   705.1282   833.3333   416.6667
##  [7]   484.6154   887.1795   583.3333     0.0000     0.0000
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.