

Table of Contents

R Script

Activity 1a

Plots

Answers to Questions

Activity 1b

Plots

Activity 1c

Plots

R Script

```
# Script for Activity 1 -- Lecture 14

##### Load data and packages

# Remove all variables from the R environment to create a fresh start
rm(list=ls())

# Set the working folder -- FILL IN THE LINE BELOW
# setwd("...")
setwd("~/Documents/SUTD/Term 5/ESA/Week 9/Lecture 14")

# Packages we are going to use in this session: ggplot2, rgl (3D scatter
plots),
# GGally (Parallel coordinate plots), fmsb (radar charts)
if(!require(ggplot2)){
  install.packages("ggplot2")
  library(ggplot2)
}
if(!require(rgl)){
  install.packages("rgl")
  library(rgl)
}
if(!require(GGally)){
  install.packages("GGally")
  library(GGally)
```

```

}

if(!require(fmsb)){
  install.packages("fmsb")
  library(fmsb)
}

# Import data from EngineDesign.csv
mydata <- read.csv(file="EngineDesign.csv")

##### Activity 1a

# Visualize the data with a scatter plot (Horsepower vs. Cost) -- FILL IN
# THE LINE BELOW
a_plot1 <- ggplot(data=mydata, mapping = aes(x=Horsepower, y=Cost)) +
  geom_point() +
  labs(title = "Activity 1a Plot 1: Horsepower vs Cost", x = "Horsepower",
y="Cost", caption="EngineDesign.csv")
a_plot1
ggsave("1a Plot1.pdf", a_plot1)

# First, we use the source() function to load ParetoSorting.R
source("ParetoSorting.R")

# Let's double-check the input (output) required (returned) by the function
ParetoSorting.R
ParetoSorting

# Prepare the input in the right format (the function requires the
# objectives to be maximized)
input <- matrix(c(mydata$Horsepower,mydata$Cost*(-1)),nrow=43,ncol=2)

# Run the function -- FILL IN THE LINE BELOW
result <- ParetoSorting(input)
result
# result
# 1 1 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1
# 0 0 0 1 1 0
# 1: dominated
# 0: not dominated (ie. pareto-efficient)

# dominated count
dominated_count <- length(which(result == 1))
dominated_count
# dominated count: 16

```

```

# not dominated count
notdominated_count <- length(which(result == 0))
notdominated_count
# non-dominated count: 39

# Let's highlight the dominated solutions on the plot -- FILL IN THE LINE
BELOW
#
# First, we create a new column in the data frame indicating which
solutions are Pareto-efficient (and which ones are not)

# function to create list stating string "dominated" or "pareto-efficient"
d_function <- function(list_input) {
  result_list <- c()
  for (i in 1:length(list_input)) {
    if (list_input[i] == 1) {
      name <- "Dominated"
    } else {
      name <- "Non-dominated"
    }
    result_list <- c(result_list, name)
  }
  return(result_list)
}

domination_list <- d_function(result)
domination_list

mydata$Dominance <- domination_list
# Then, we give a different colour to the points depending on the value in
the column "Dominance"
a_plot2 <- ggplot(data=mydata, mapping = aes(x=Horsepower, y=Cost)) +
  geom_point(aes(color=Dominance)) +
  labs(title = "Activity 1a Plot 2: Horsepower vs Cost", x = "Horsepower",
y="Cost", caption="EngineDesign.csv")
a_plot2

ggsave("1a Plot2.pdf", a_plot2)

# How to visualize the third objective?
#
# We can represent the third objective (Fuel-efficiency) with a size or
color. -- FILL IN THE LINES BELOW
# fuel efficiency by size

```

```

a_plot3 <- ggplot(data = mydata,
                    mapping = aes(x = Horsepower, y = Cost, size = Fuel.efficiency)) +
  geom_point() + # Note that we removed "size = 3"
  coord_cartesian(xlim=c(0.2,0.8), ylim=c(0,0.2)) +
  labs(title = "Activity 1a Plot 3: Horsepower vs Cost", x= "Horsepower",
       y="Cost", caption="EngineDesign.csv")
a_plot3

ggsave("1a Plot3.pdf", a_plot3)

# fuel efficiency by colour
a_plot4 <- ggplot(data = mydata,
                    mapping = aes(x = Horsepower, y = Cost, colour = Fuel.efficiency)) +
  geom_point(size = 3) +
  coord_cartesian(xlim=c(0.2,0.8), ylim=c(0,0.2)) +
  labs(title = "Activity 1a Plot 4: Horsepower vs Cost", x= "Horsepower",
       y="Cost", caption="EngineDesign.csv")
a_plot4

ggsave("1a Plot4.pdf", a_plot4)
#
# We could also use a 3D plot, for which we need the rgl package
a_plot5 <- plot3d(mydata$Horsepower, mydata$Cost, mydata$Fuel.efficiency,
type="s",
                    size=1, lit=TRUE, main = "Horsepower Vs Cost Vs Fuel-
efficiency",sub="3-D Plot")

rgl.postscript("1a Plot5.pdf", fmt = "pdf")

# Now, we can determine which solutions are Pareto-efficient. First, we use
the source() function to load ParetoSorting.R
source("ParetoSorting_adv.R")

# Let's double-check the input (output) required (returned) by the function
ParetoSorting.R
ParetoSorting_adv

# Prepare the input in the right format
input_adv <- matrix(c(mydata$Horsepower,mydata$Cost*
(-1),mydata$Fuel.efficiency),nrow=43,ncol=3)

# Run the function -- FILL IN THE LINE BELOW
result_adv <- ParetoSorting_adv(input_adv, "MAX")
result_adv

```



```

##### Activity 1b

# Visualize the data with a parallel coordinate plot (Horsepower vs Cost vs
Fuel Efficiency)
# Axes should be oriented in a specific direction, so we define a new
dataframe where the price is multiplied by -1.
# All objectives should be maximized --> the ideal solution is a line on
the top of the plot -- FILL IN THE LINE BELOW
#
# Basic parallel coordinate plot, using default settings
mydata$Flipped_Cost = mydata$Cost * (-1)
b_plot1 <- ggparcoord(data = data.frame(Horsepower = mydata[,1],
Cost=mydata[,6], Fuel.efficiency = mydata[,3]),
columns = 1:3, # a vector of variables (either names or indices)
to be axes in the plot, name wrote in line above before = sign
title = "Activity 1b Plot 1: Default Settings")
b_plot1

ggsave("1b Plot1.pdf", b_plot1)

# change scaling method
b_plot2 <- ggparcoord(data = data.frame(Horsepower = mydata[,1], Cost =
mydata[,6], Fuel.efficiency = mydata[,3]),
columns = 1:3,
title = "Activity 1b Plot 2: Scaling Method Changed",
scale = "uniminmax") # univariately, scale so the
minimum of the variable is 0, and max is 1
b_plot2

ggsave("1b Plot2.pdf", b_plot2)

# use different colours for dominated and non-dominated
b_plot3 <- ggparcoord(data = data.frame(Horsepower = mydata[,1], Cost =
mydata[,6], Fuel.efficiency = mydata[,3], Dominance = mydata[,5]),
columns = 1:3, # more than the amount of data columns
to be used as variables can be specified above
# but not used as one of the variables
on the x-axis
# can be used to group data points for
visualisation
title = "Activity 1b Plot 3: Different Colours
Depending on Domination",
scale = "uniminmax",

```

```

groupColumn = "Dominance") # a single variable to
group (colour) by

b_plot3

ggsave("1b Plot3.pdf", b_plot3)

# ADVANCED
# utilise ggplot2 aes to switch to thicker lines
b_plot4 <- ggparcoord(data = data.frame(Horsepower = mydata[,1], Cost =
mydata[,6], Fuel.efficiency = mydata[,3], Dominance = mydata[,5]),
columns = 1:3,
title = "Activity 1b Plot 4: Using ggplot2 to switch
to thicker lines",
scale = "uniminmax",
groupColumn = "Dominance",
mapping = ggplot2::aes(size = Dominance)) #aes string
to pass to ggplot object

b_plot4

ggsave("1b Plot4.pdf", b_plot4)

##### Activity 1c

# First, we visualize only one solution. For example, we visualize
Alternative 1.
# mydata_ <-
data.frame(Horsepower=input[1,1],Cost=input[1,2],Fuel.efficiency=input[1,3]
)
mydata_ <- data.frame(Horsepower = mydata[,1], Cost = mydata[, 6],
Fuel.efficiency = mydata[, 3])

# To use the fmsb package, we have to add 2 lines to the dataframe: the max
and min of each topic to show on the plot!
mydata_ = rbind(c(1,0,1),c(0,-1,0),mydata_) # -- FILL IN THE LINE BELOW
c_plot1 <- radarchart(df=mydata_,
title = "Activity 1c Plot 1: Alternative Solution 1")

c_plot1

legend(1,1,lty=c(1,1),lwd=c(2.5,2.5),col=c("black"),c("Alternative 1"), cex
= 0.4)

```

```

# Now, let's compare the first two solutions -- FILL IN THE LINE BELOW
# mydata_ <- ...
mydata_1 <- data.frame(Horsepower = mydata[1:2,1], Cost = mydata[1:2, 6],
Fuel.efficiency = mydata[1:2, 3])
mydata_1 = rbind(c(1,0,1),c(0,-1,0),mydata_1)
radarchart(df = mydata_1,
           title = "Activity 1c Plot 2: Alternative Solutions 1 and 2")
legend(1,1,lty=c(1,1),lwd=c(2.5,2.5),col=c("black","red"),c("Alternative
1","Alternative 2"),cex=0.4)
# You can notice that the second solution has less Horsepower but a better
Cost ... it is still Pareto efficient

# Finally, let's compare a non-efficient (8) solution against an efficient
one (25) -- FILL IN THE LINE BELOW
# mydata_ <- ...
mydata_2 <- data.frame(Horsepower = c(mydata[8, 1], mydata[25,1]), Cost =
c(mydata[8,6,], mydata[25, 6]), Fuel.efficiency = c(mydata[8, 3],
mydata[25, 3]))

mydata_2 = rbind(c(1,0,1),c(0,-1,0),mydata_2)
radarchart(df = mydata_2,
           title = "Activity 1c Plot 3: Alternative Solutions 8 and 25")
legend(1,1,lty=c(1,1),lwd=c(2.5,2.5),col=c("black","red"),c("Alternative
8","Alternative 25"),cex=0.4)
# Solution 8 is slightly worse with respect to all objectives!
input[8,]
input[25,]

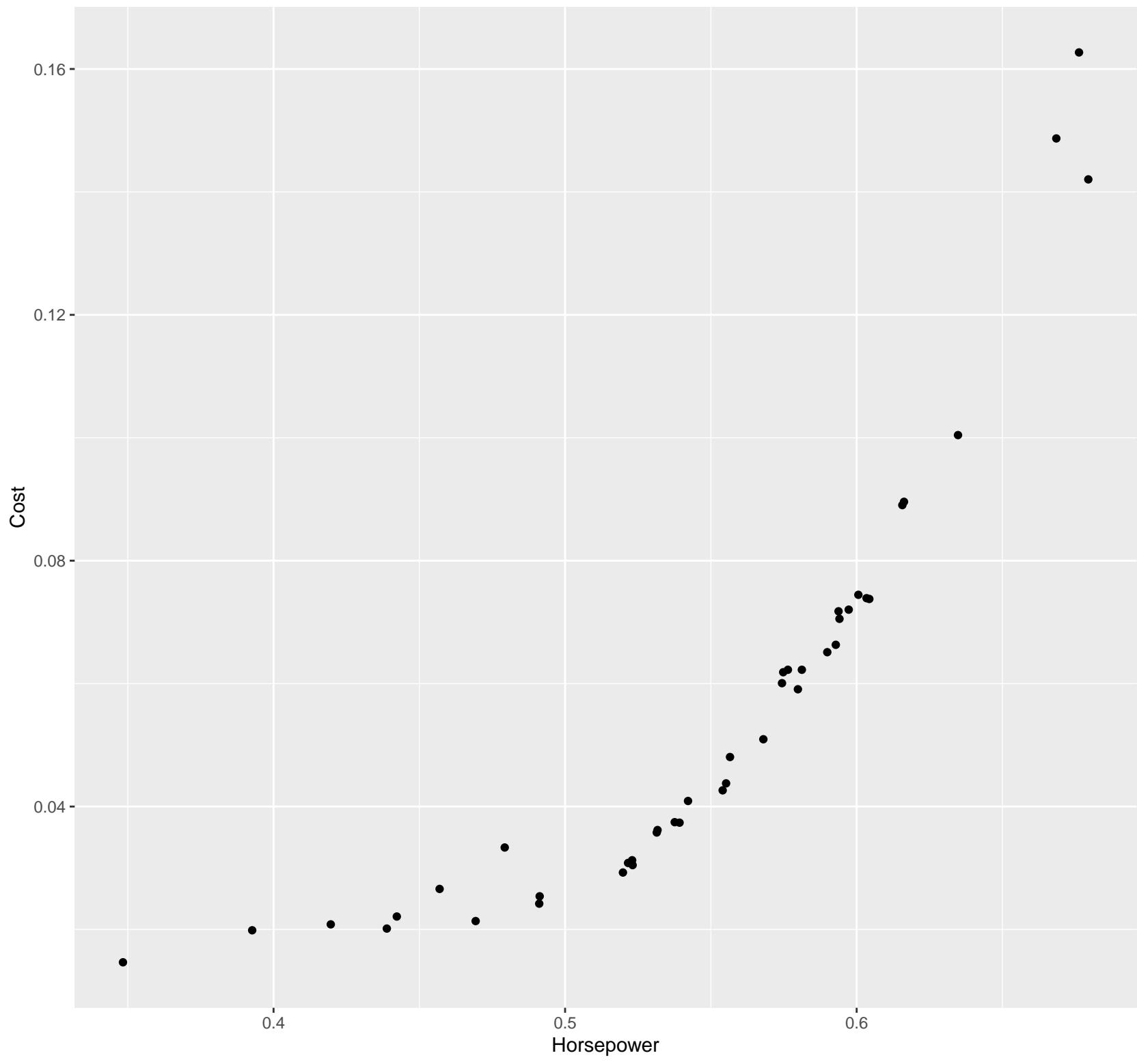
mydata[8,]
mydata[25,]

```

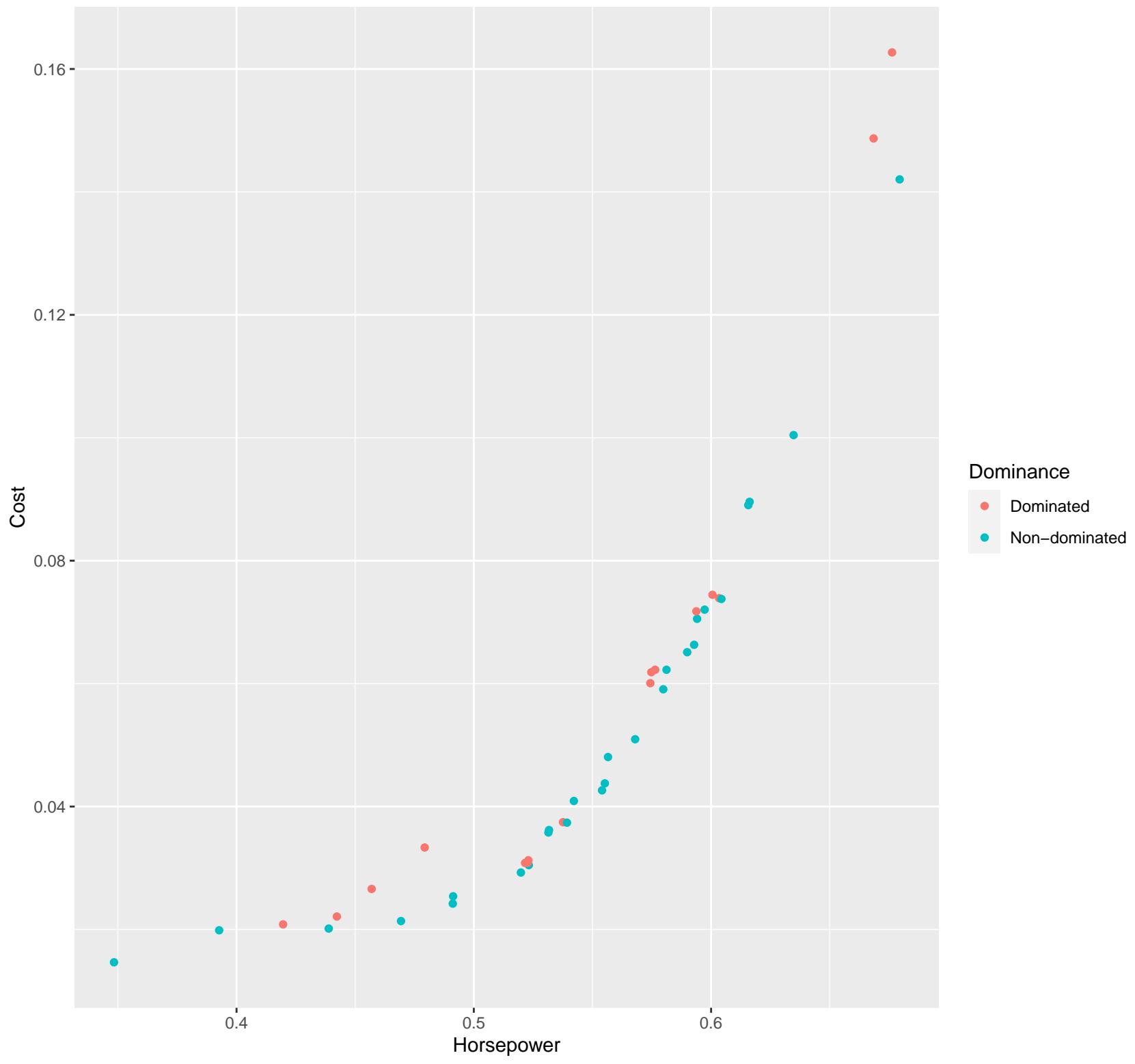
Activity 1a

Plots

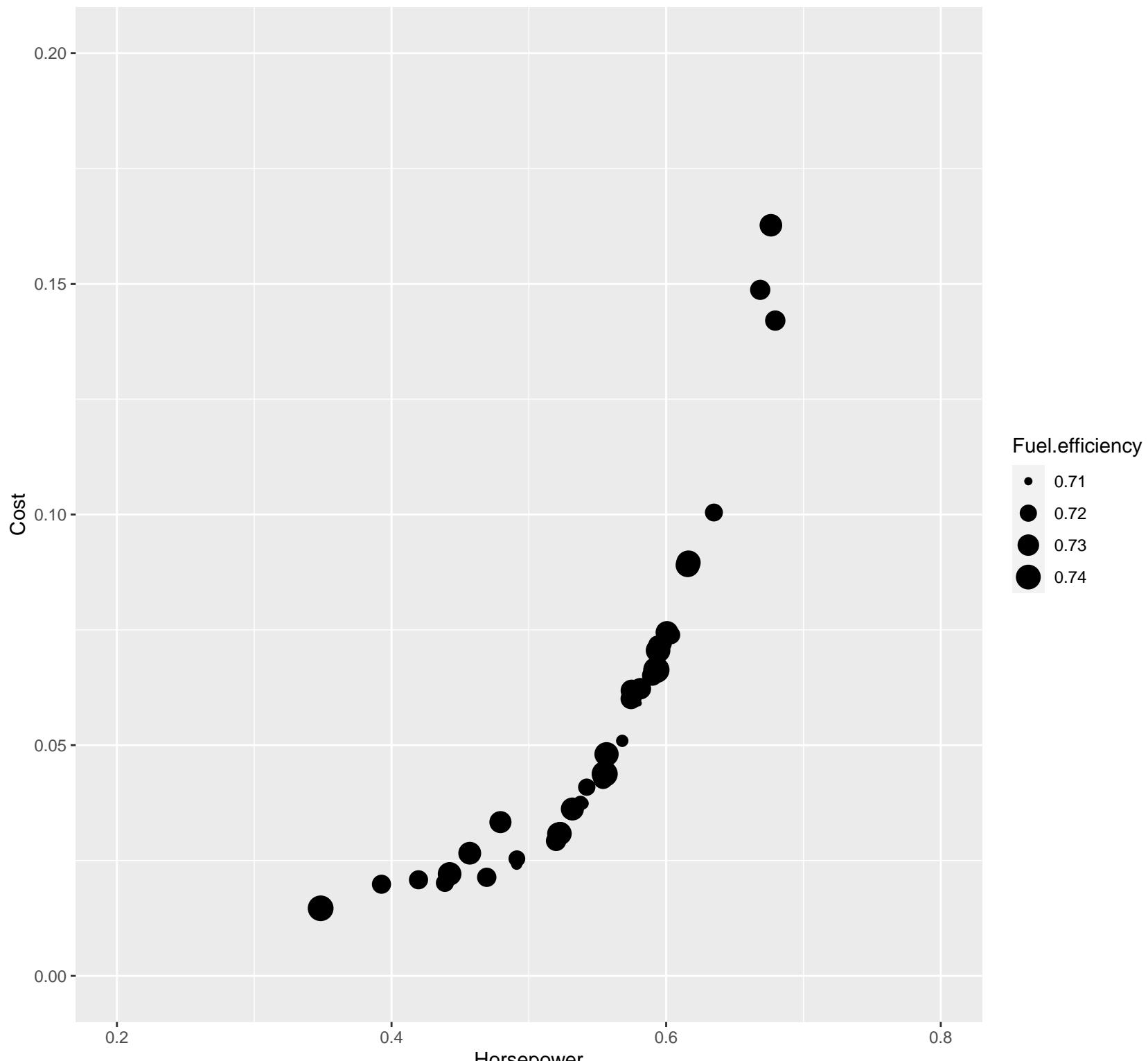
Activity 1a Plot 1: Horsepower vs Cost



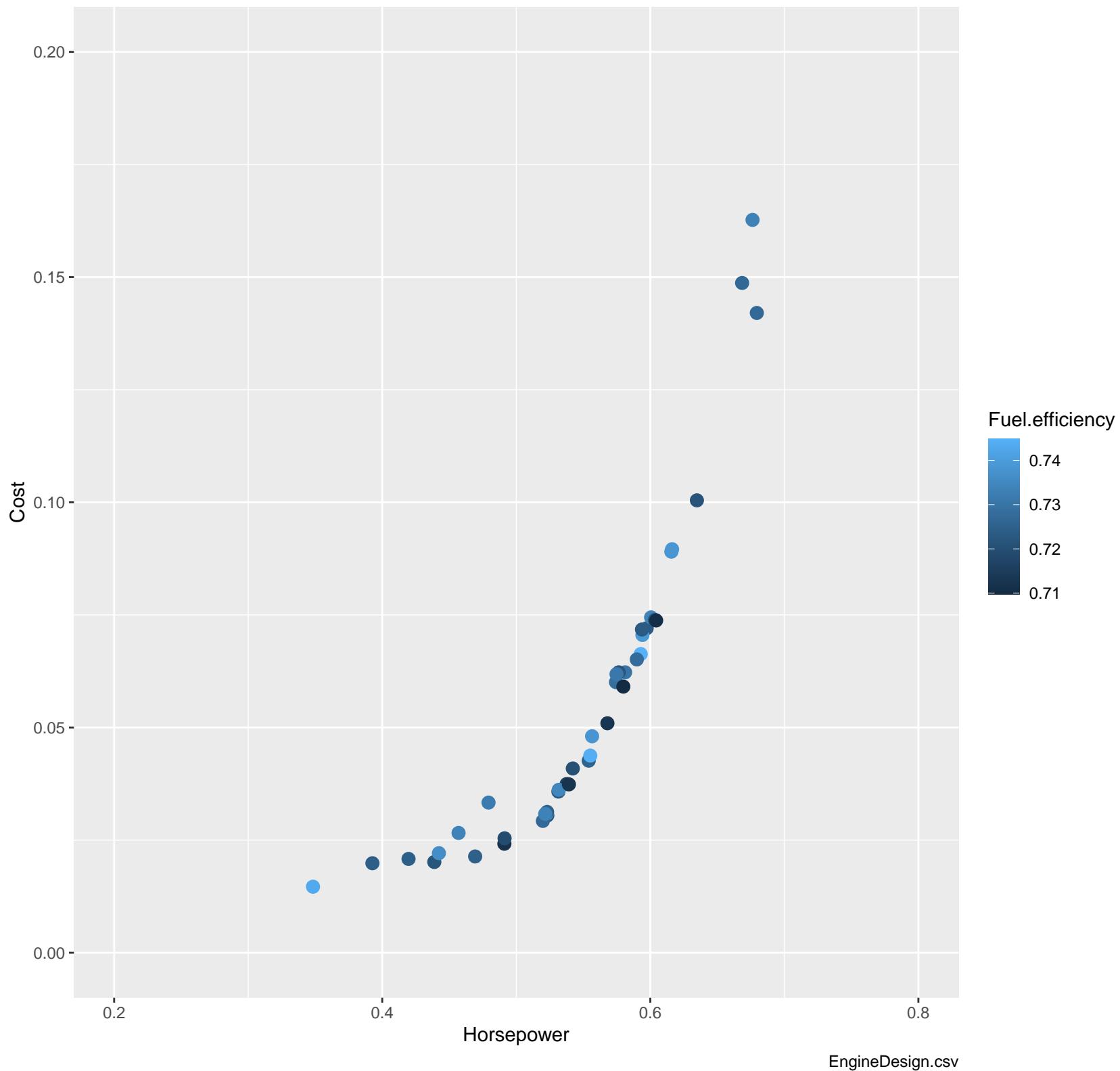
Activity 1a Plot 2: Horsepower vs Cost



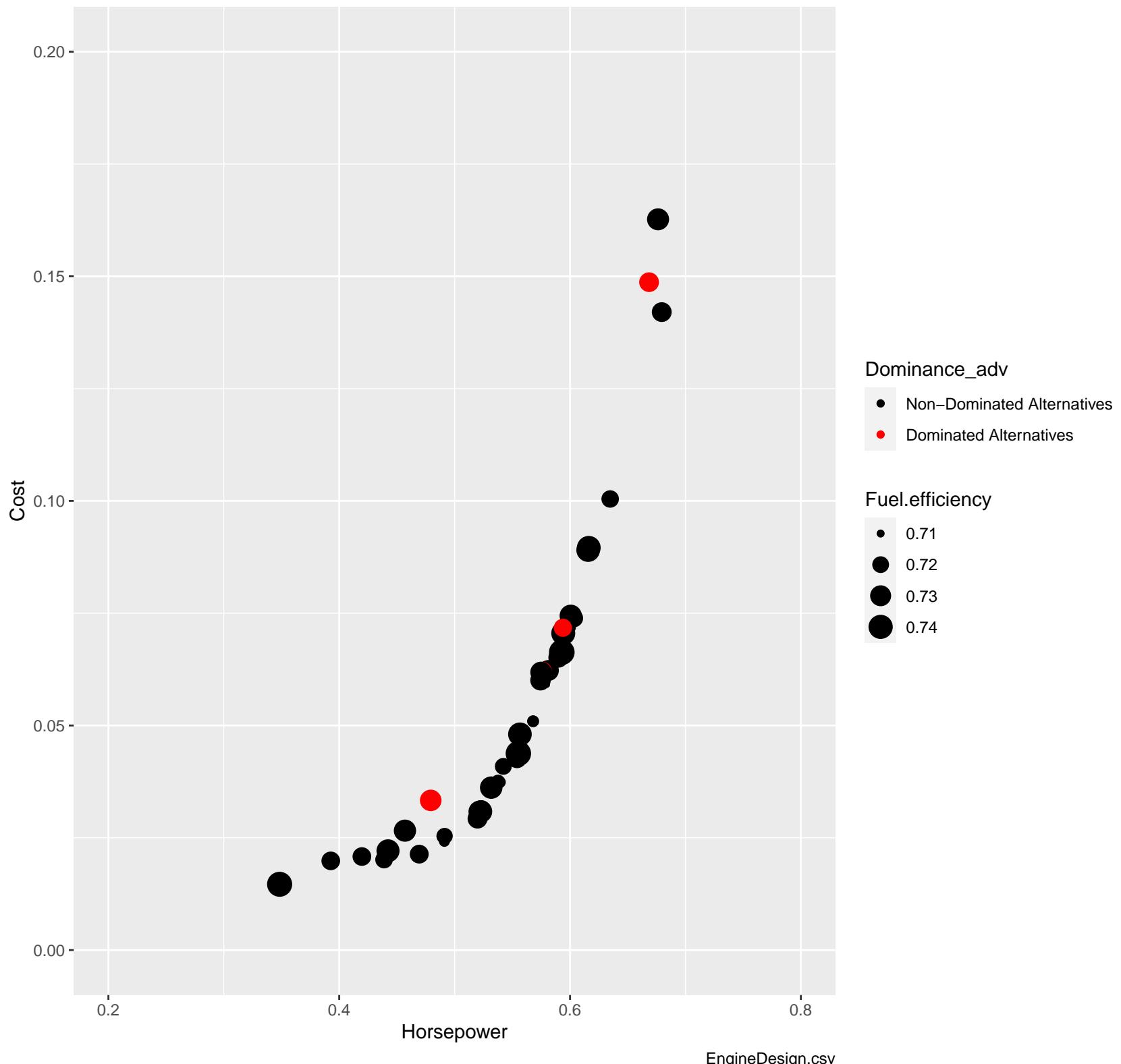
Activity 1a Plot 3: Horsepower vs Cost



Activity 1a Plot 4: Horsepower vs Cost



Activity 1a Plot 6



Answers to Questions

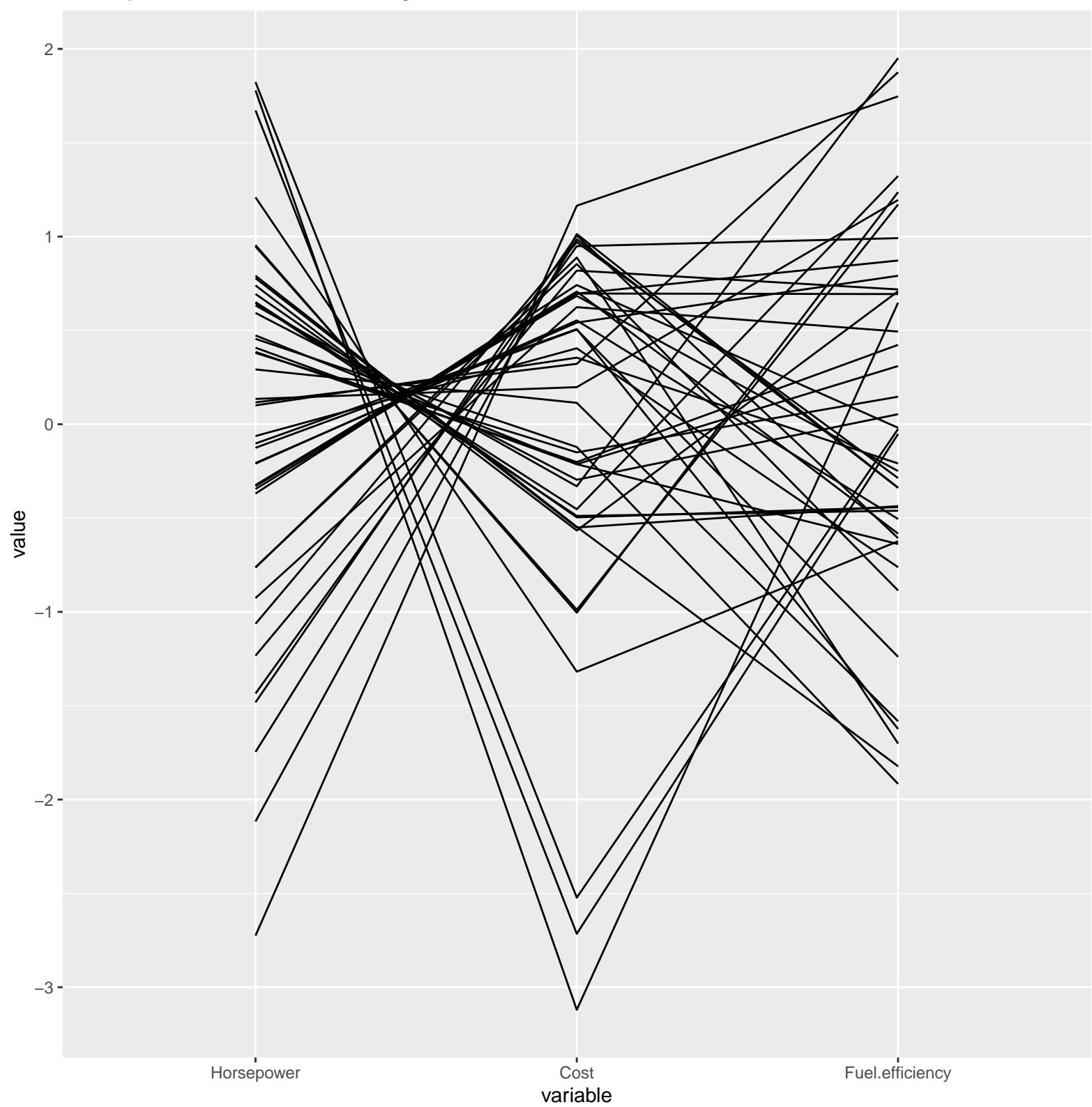
Number of alternative solutions dominated

- Horsepower vs Cost: 16
- Horsepower vs Cost vs Fuel-Efficiency: 4

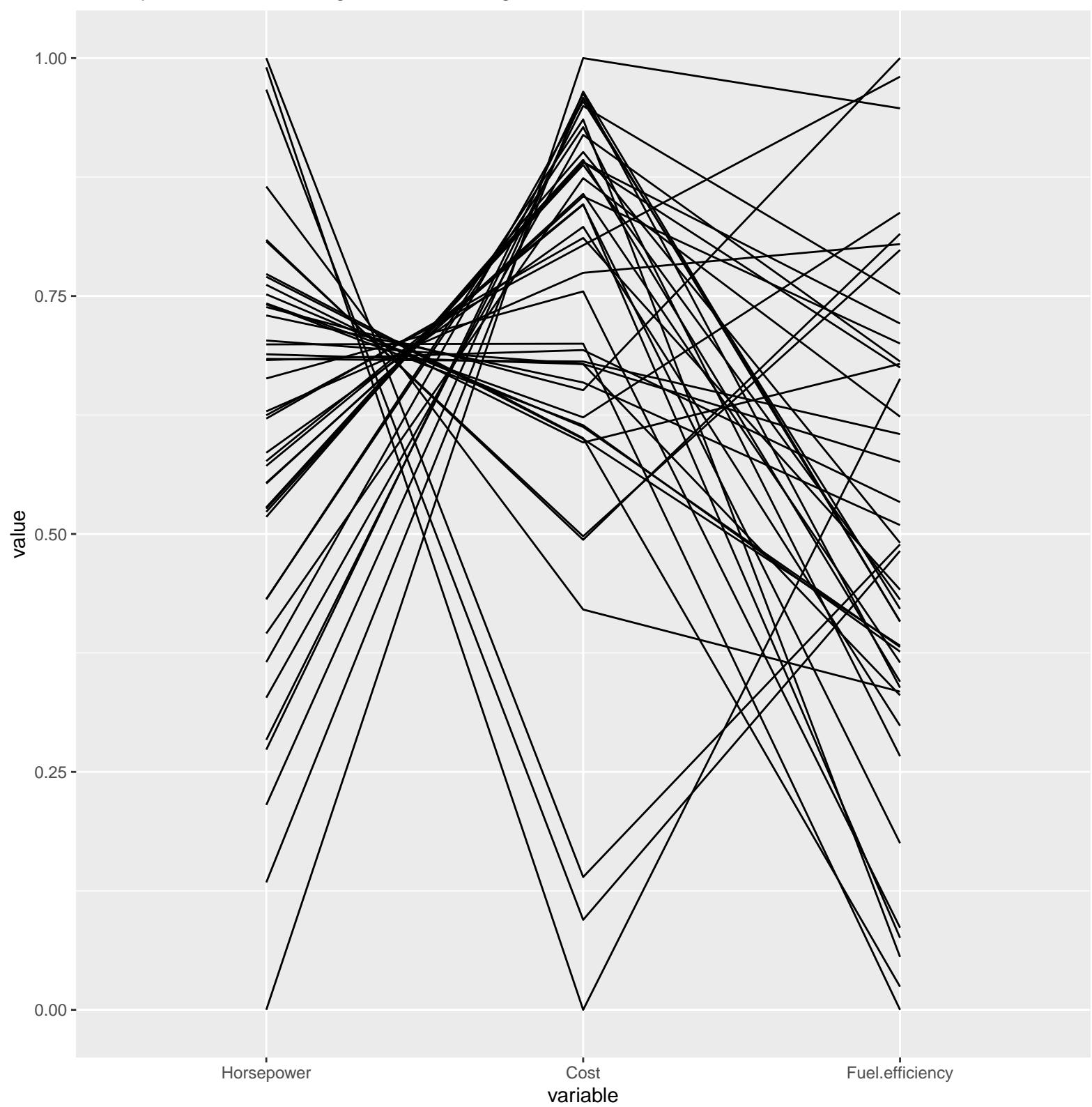
Activity 1b

Plots

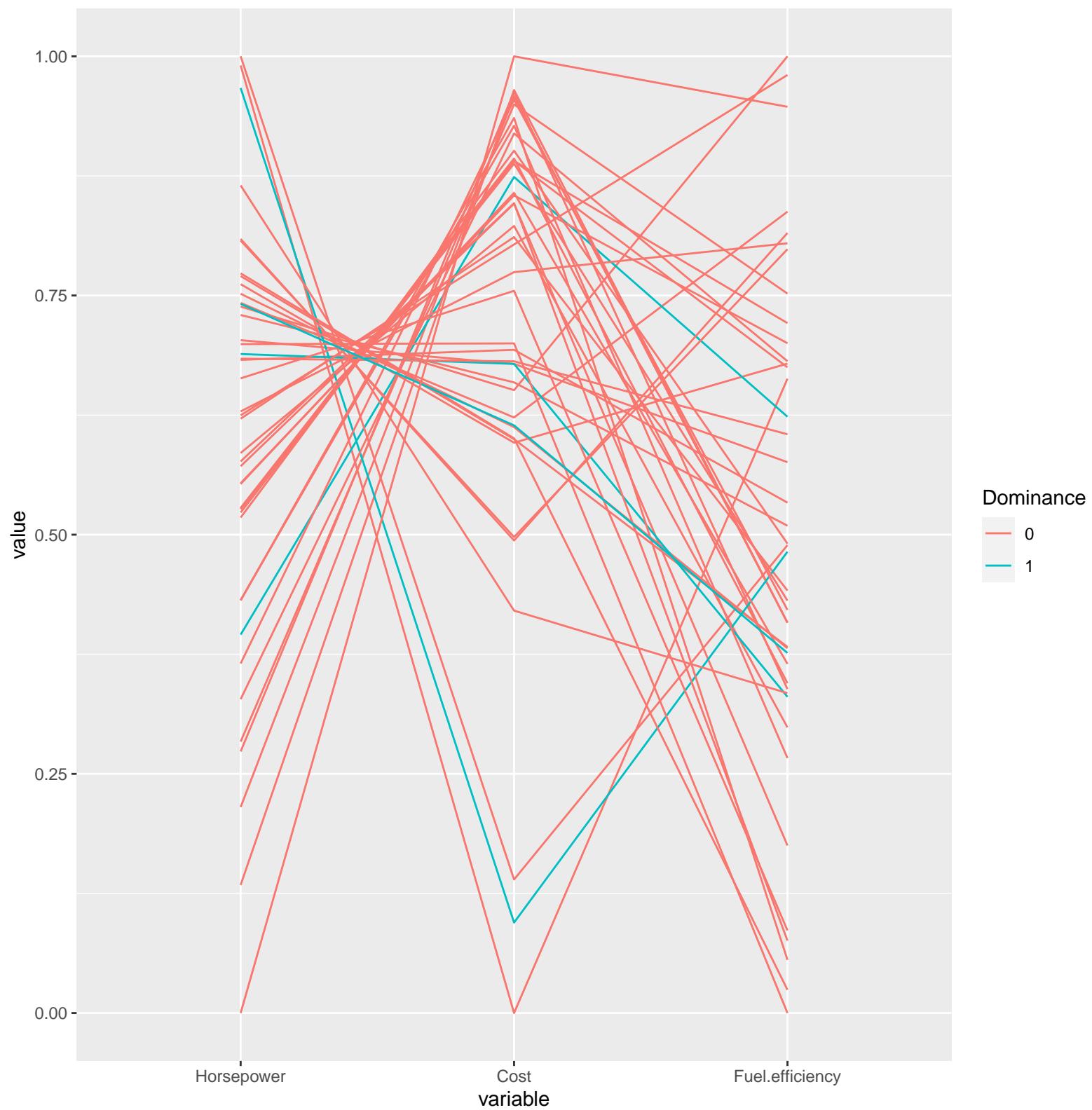
Activity 1b Plot 1: Default Settings



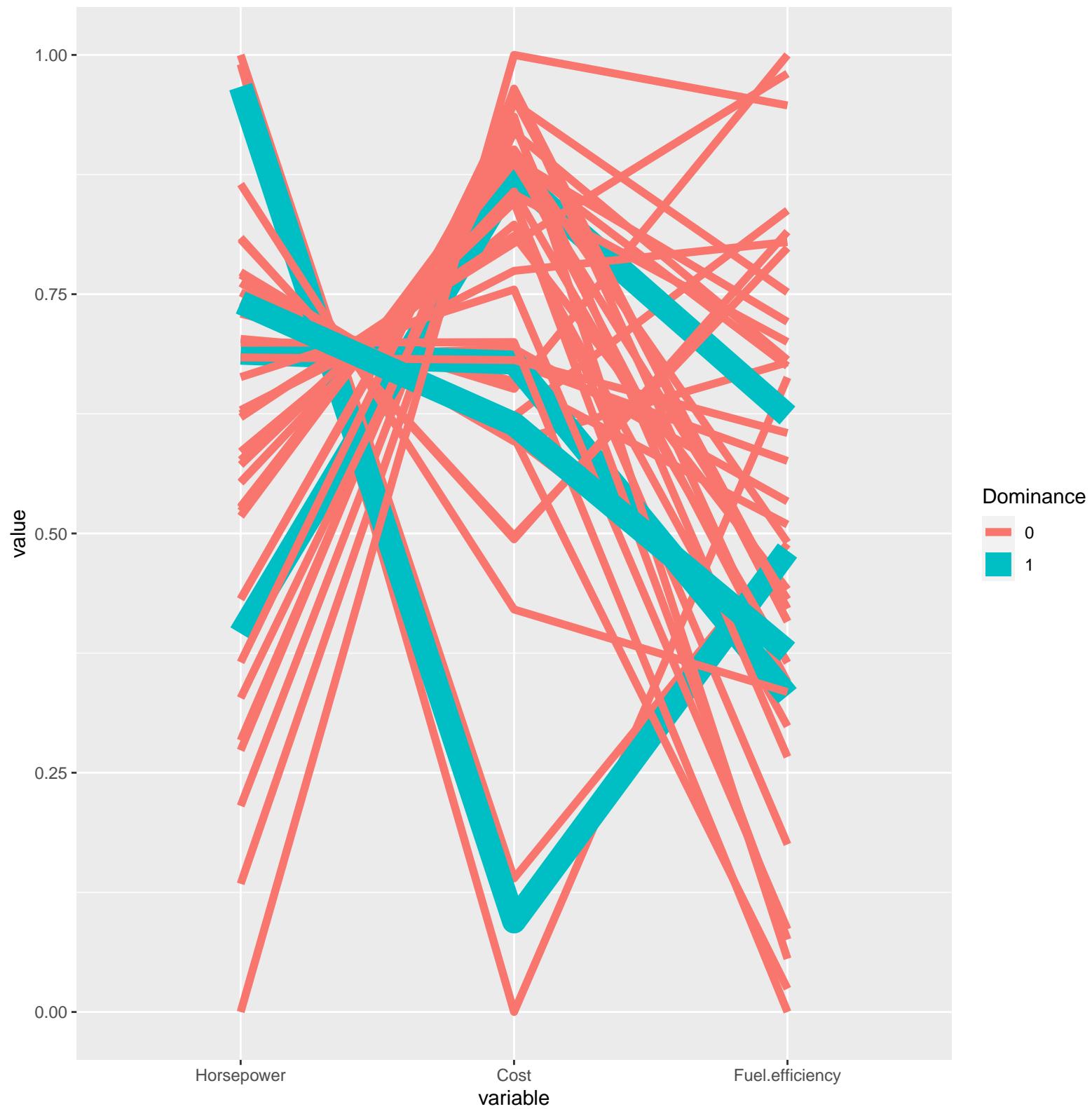
Activity 1b Plot 2: Scaling Method Changed



Activity 1b Plot 3: Different Colours Depending on Domination



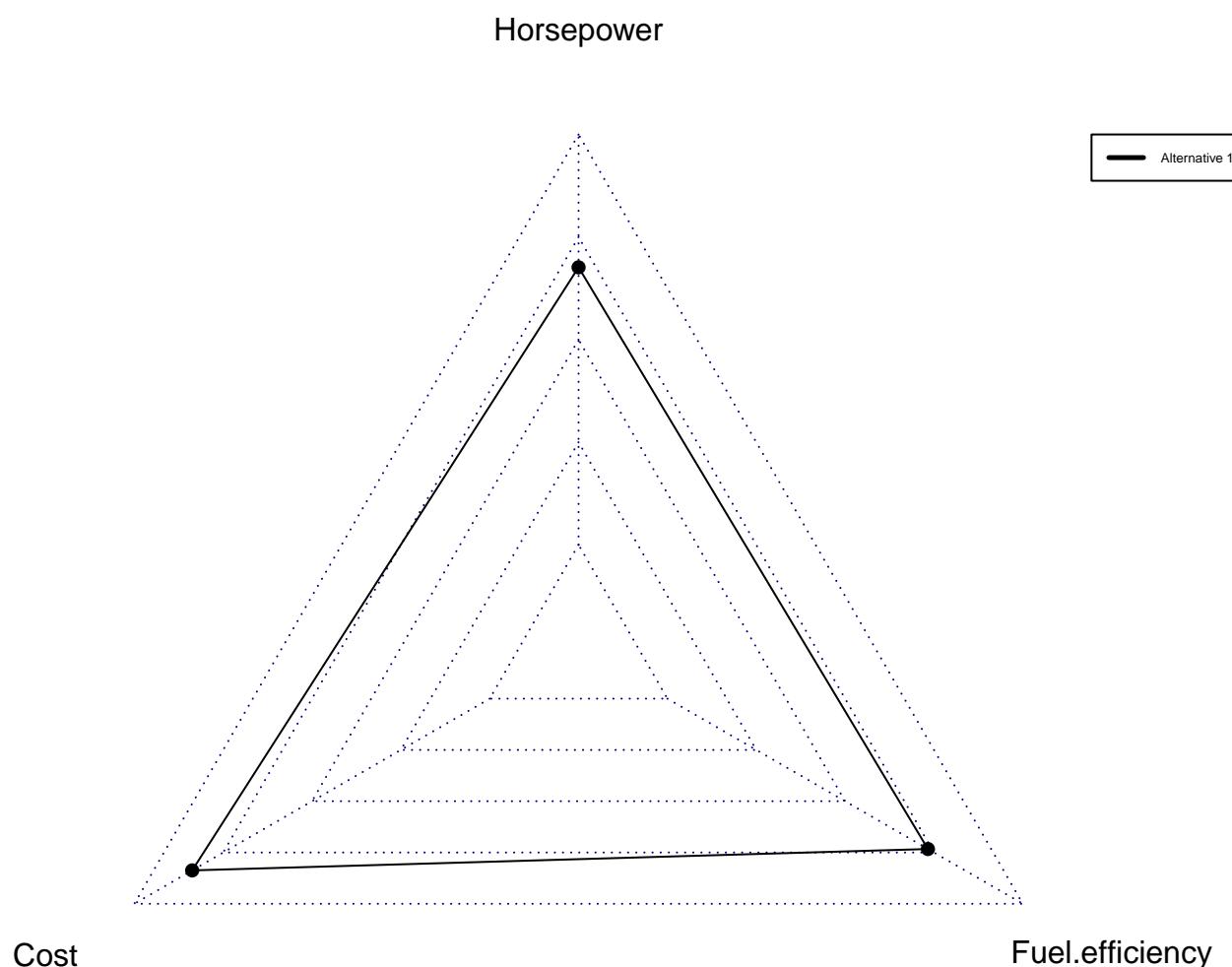
Activity 1b Plot 4: Using ggplot2 to switch to thicker lines



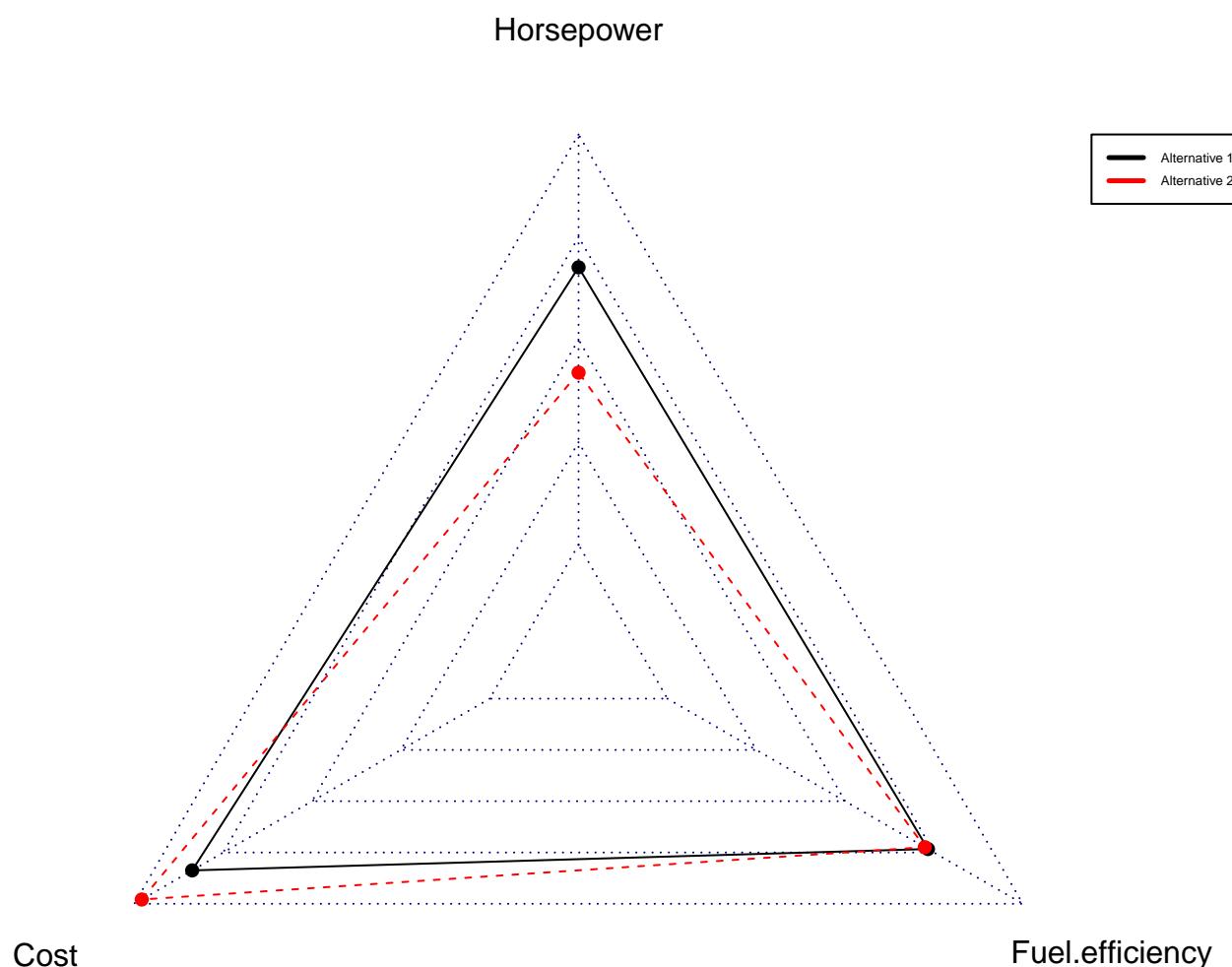
Activity 1c

Plots

Activity 1c Plot 1: Alternative Solution 1



Activity 1c Plot 2: Alternative Solutions 1 and 2



Activity 1c Plot 3: Alternative Solutions 8 and 25

