Reproduceable Research

The following document shows how the statistical and graphics program *R* is used to analyse data for this project. Set the working directory to where the output files for "MyCarrierPlanInterpreter.java" are. These files are available at the github repository for this project. The outputs for "MyCarrierPlanInterpreter.java" can be found at https://github.com/fransbrooks/BPJ420FinalYearProject/tree/master/CarrierPlanInterpreter. For the purposes of this document an analysis of day 3's demand will be shown. The relevant directories and file names can be changed to analyse the demand for any specific day. The directory specified should contain all the files produced as output for a given day of demand data. This should result in a directory for a given day of demand containing 15 .csv files.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
```

Load the ggplot2 package for plotting data and load the xtable package to generate LaTeX code for tables.

```
library(ggplot2)
library(xtable)
```

Fleet Composition

Running-Fixed-Cost and Variable Cost

Read in the .csv file containing data about the fleet composition for running-fixed-cost and variable cost

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoRunFixAndVar <- read.csv("Day3_80_vehicleTypesFixedRunningAndVariablesumValues.csv")
FSMinfoRunFixAndVar
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 1  truck_12_1 -759298       89   60788          3164      26700
## 2 truck_3_1_0 -759298       38   65717          2931      14100
## 3   truck_3_1 -759298       22  342276         18644       9900
```

Make separate tables for each type of truck for running-fixed-cost and variable cost. Do this by subsetting the data read in above. Subset by searching for a string within the vehId column of the "FSMinfoRunFixAndVar" data.frame, in order to identify the vehicle type. The string identifying the vehicle type is searched for by making use of the grep function.

```
RunFixAndVarThreeTonners <- FSMinfoRunFixAndVar[grep("truck_3",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarSixTonners <- FSMinfoRunFixAndVar[grep("truck_6",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarSevenTonners <- FSMinfoRunFixAndVar[grep("truck_7",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarTwelveTonners <- FSMinfoRunFixAndVar[grep("truck_12",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarFifteenTonners <- FSMinfoRunFixAndVar[grep("truck_15",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarThreeTonners
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 2 truck_3_1_0 -759298       38   65717          2931      14100
## 3   truck_3_1 -759298       22  342276         18644       9900
```

Make a data.frame indicating how many vehicles of each type was specified for running-fixed-cost and variable cost.

Make a character vector containing the names of the columns for the data.frame and create an empty data.frame.

```
CollumnNames <- c("3Tonners", "6Tonners", "7Tonners", "12Tonners", "15Tonners")
TruckTotals <- data.frame(matrix(nrow=0, ncol=5))
```

Find the number of rows for each truck type from the data frames by using the length function. This will give the number of trucks for each truck type specified in the fleet composition generated by solving the FSMVRP. Append the number of vehicles specified to the data.frame.

```
TruckTotals <- rbind(c(length(RunFixAndVarThreeTonners$vehId), length(RunFixAndVarSixTonners$vehId),
    length(RunFixAndVarSevenTonners$vehId), length(RunFixAndVarTwelveTonners$vehId),
    length(RunFixAndVarFifteenTonners$vehId)))
```

Assign the column names to the data.frame "TruckTotals" in order to see how many of each vehicle type was specified in the solution to the FSMVRP. Name the row according to the cost parameters that were used in order to generate the fleet composition.

```
colnames(TruckTotals) <- CollumnNames
rownames(TruckTotals) <- "RunFixAndVar"
TruckTotals
```

```
##              3Tonners 6Tonners 7Tonners 12Tonners 15Tonners
## RunFixAndVar        2        0        0         1         0
```

Variable Cost

Read in the .csv file containing data about the fleet composition for variable cost only.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoVar <- read.csv("Day3_80_vehicleTypesOnlyVariablesumValues.csv")
```

Make separate data frames for each type of truck for variable cost only.

```
VarThreeTonners <- FSMinfoVar[grep("truck_3",FSMinfoVar$vehId),]
VarSixTonners <- FSMinfoVar[grep("truck_6",FSMinfoVar$vehId),]
VarSevenTonners <- FSMinfoVar[grep("truck_7",FSMinfoVar$vehId),]
VarTwelveTonners <- FSMinfoVar[grep("truck_12",FSMinfoVar$vehId),]
VarFifteenTonners <- FSMinfoVar[grep("truck_15",FSMinfoVar$vehId),]
```

Append the data for using variable cost only to the TruckTotals data.frame and change the row names to reflect the newly added data.

```
TruckTotals <- rbind(TruckTotals, c(length(VarThreeTonners$vehId), length(VarSixTonners$vehId),
    length(VarSevenTonners$vehId), length(VarTwelveTonners$vehId), length(VarFifteenTonners$vehId)))
rownames(TruckTotals) <- c("RunFixAndVar", "VarOnly")
```

Running-Fixed-Cost

Read in the .csv file containing data about the fleet composition for running-fixed-cost only.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoRunFix <- read.csv("Day3_80_vehicleTypesOnlyFixedRunningsumValues.csv")
```

Make separate data frames for each type of truck for fixed cost only.

```
RunFixThreeTonners <- FSMinfoRunFix[grep("truck_3",FSMinfoRunFix$vehId),]
RunFixSixTonners <- FSMinfoRunFix[grep("truck_6",FSMinfoRunFix$vehId),]
RunFixSevenTonners <- FSMinfoRunFix[grep("truck_7",FSMinfoRunFix$vehId),]
RunFixTwelveTonners <- FSMinfoRunFix[grep("truck_12",FSMinfoRunFix$vehId),]
RunFixFifteenTonners <- FSMinfoRunFix[grep("truck_15",FSMinfoRunFix$vehId),]
```

Append the data for using variable cost only to the TruckTotals data.frame.

```
TruckTotals <- rbind(TruckTotals, c(length(RunFixThreeTonners$vehId), length(RunFixSixTonners$vehId),
    length(RunFixSevenTonners$vehId), length(RunFixTwelveTonners$vehId), length(RunFixFifteenTonners$veh
rownames(TruckTotals) <- c("RunFixAndVar", "VarOnly", "RunFixOnly")
```

Fixed Cost and Variable Cost

Read in the .csv file containing data about the fleet composition for fixed cost and variable cost.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoFixAndVar <- read.csv("Day3_80_vehicleTypesAbsFixAndVarsumValues.csv")
```

Make separate data frames for each type of truck for fixed and variable cost.

```
FixAndVarThreeTonners <- FSMinfoFixAndVar[grep("truck_3",FSMinfoFixAndVar$vehId),]
FixAndVarSixTonners <- FSMinfoFixAndVar[grep("truck_6",FSMinfoFixAndVar$vehId),]
FixAndVarSevenTonners <- FSMinfoFixAndVar[grep("truck_7",FSMinfoFixAndVar$vehId),]
FixAndVarTwelveTonners <- FSMinfoFixAndVar[grep("truck_12",FSMinfoFixAndVar$vehId),]
FixAndVarFifteenTonners <- FSMinfoFixAndVar[grep("truck_15",FSMinfoFixAndVar$vehId),]
```

Append the data for using fixed and variable cost to the TruckTotals data.frame.

```
TruckTotals <- rbind(TruckTotals, c(length(FixAndVarThreeTonners$vehId), length(FixAndVarSixTonners$veh
    length(FixAndVarSevenTonners$vehId), length(FixAndVarTwelveTonners$vehId),
    length(FixAndVarFifteenTonners$vehId)))
rownames(TruckTotals) <- c("RunFixAndVar", "VarOnly", "RunFixOnly", "FixAndVar")
```

Fixed Cost

Read in the .csv file containing data about the fleet composition for fixed cost only.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoFix <- read.csv("Day3_80_vehicleTypesOnlyAbsFixsumValues.csv")
```

Make separate data frames for each type of truck for fixed cost only.

```
FixThreeTonners <- FSMinfoFix[grep("truck_3",FSMinfoFixAndVar$vehId),]
FixSixTonners <- FSMinfoFix[grep("truck_6",FSMinfoFixAndVar$vehId),]
FixSevenTonners <- FSMinfoFix[grep("truck_7",FSMinfoFixAndVar$vehId),]
FixTwelveTonners <- FSMinfoFix[grep("truck_12",FSMinfoFixAndVar$vehId),]
FixFifteenTonners <- FSMinfoFix[grep("truck_15",FSMinfoFixAndVar$vehId),]
```

Append the data for using fixed cost only to the TruckTotals data.frame

```
TruckTotals <- rbind(TruckTotals, c(length(FixThreeTonners$vehId), length(FixSixTonners$vehId),
    length(FixSevenTonners$vehId), length(FixTwelveTonners$vehId), length(FixFifteenTonners$vehId)))
rownames(TruckTotals) <- c("Running-Fixed-Cost and Variable Cost", "Variable Cost",
    "Running-Fixed-Cost", "Fixed and Variable Cost", "Fixed Cost")
```

Finalise the TruckTotals data.frame

Add a new column which shows total number of vehicles for each cost scenario by using the rowSums function. Append the new column to the TruckTotals data.frame by using the cbind function. The TruckTotals data.frame now shows the fleet composition sing each set of cost parameters for day 3's demand data.

```
TruckTotals <- cbind(TruckTotals, rowSums(TruckTotals))
CollumnNames <- c("3Tonners", "6Tonners", "7Tonners", "12Tonners", "15Tonners",
    "Total")
colnames(TruckTotals) <- CollumnNames
TruckTotals
```

```
##                                     3Tonners 6Tonners 7Tonners 12Tonners
## Running-Fixed-Cost and Variable Cost        2        0        0         1
## Variable Cost                               0        0        0         0
## Running-Fixed-Cost                          1        2        0         0
## Fixed and Variable Cost                     2        0        0         1
## Fixed Cost                                  2        0        0         1
##                                     15Tonners Total
## Running-Fixed-Cost and Variable Cost         0     3
## Variable Cost                                3     3
## Running-Fixed-Cost                           0     3
## Fixed and Variable Cost                      0     3
## Fixed Cost                                   0     3
```

Generate LaTeXCode for Fleet Compositions

Use the xtable package to generate a table in LaTeXcode for use in LaTeXdocuments.

```
xtable(TruckTotals)
```

```
## % latex table generated in R 3.1.1 by xtable 1.7-3 package
## % Wed Aug 27 20:58:33 2014
## \begin{table}[ht]
## \centering
```

```
## \begin{tabular}{rrrrrrr}
##   \hline
##  & 3Tonners & 6Tonners & 7Tonners & 12Tonners & 15Tonners & Total \\
##   \hline
## Running-Fixed-Cost and Variable Cost & 2.00 & 0.00 & 0.00 & 1.00 & 0.00 & 3.00 \\
##   Variable Cost & 0.00 & 0.00 & 0.00 & 0.00 & 3.00 & 3.00 \\
##   Running-Fixed-Cost & 1.00 & 2.00 & 0.00 & 0.00 & 0.00 & 3.00 \\
##   Fixed and Variable Cost & 2.00 & 0.00 & 0.00 & 1.00 & 0.00 & 3.00 \\
##   Fixed Cost & 2.00 & 0.00 & 0.00 & 1.00 & 0.00 & 3.00 \\
##    \hline
## \end{tabular}
## \end{table}
```

Make Fleet Composition Plots

Plot fleet composition for running-fixed-cost and variable cost with ggplot2 package

Make the variable x equal to the column names and change the x to be of the character class. Then change x to a factor variable. Use the factor function and its arguments to preserve the order of the original CollumnNames so that fleet composition can be plotted from the smaller sized vehicles to the larger sized vehicles. If this procedure is not followed the plots will change the order of vehicles on the x-axis to alphabetical order, leading to 12-ton vehicles being plotted first instead of 3-ton vehicles being plotted first.

```
x <- CollumnNames
x <- as.character(x)
x <- factor(x, levels=unique(x))
```

Let y be equal to the first row of the TruckTotals data.frame, because it is the relevant row for plotting the fleet composition for running-fixed-cost and variable cost.

```
TruckTotals[1, ]
```

```
##  3Tonners  6Tonners  7Tonners 12Tonners 15Tonners     Total
##         2         0         0         1         0         3
```

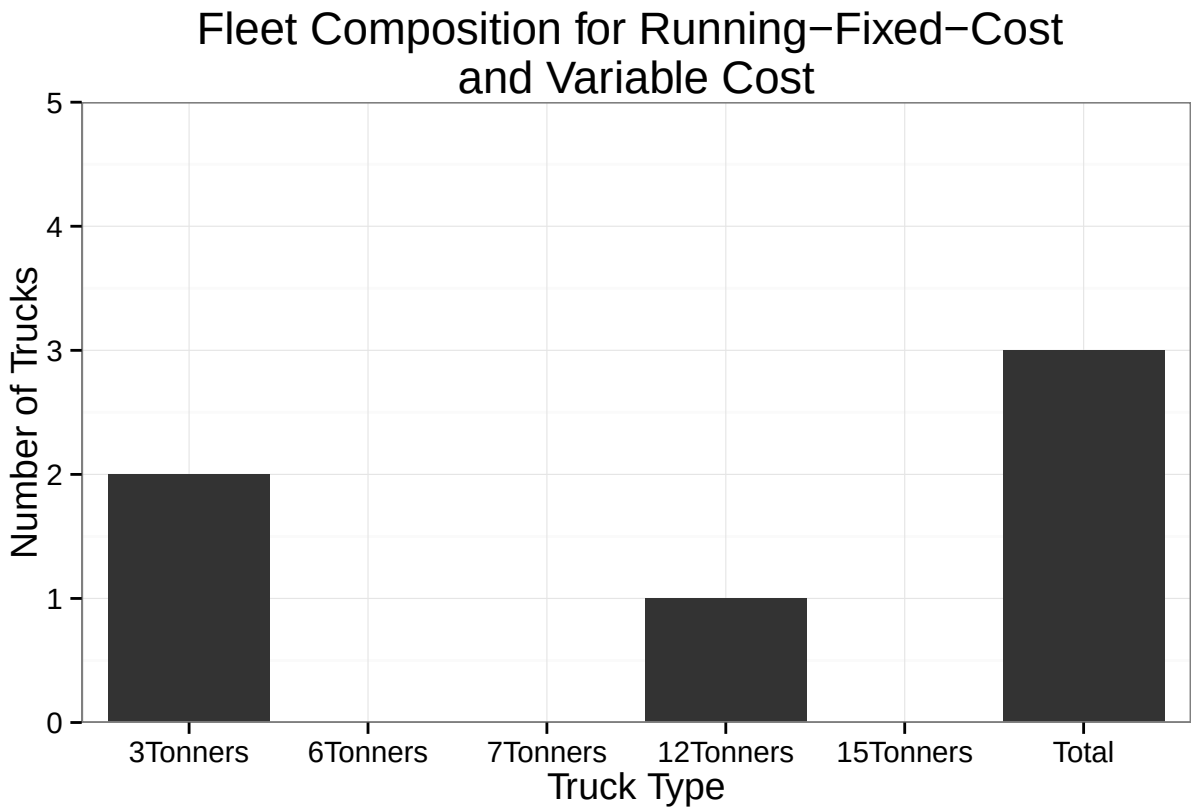
```
y <- TruckTotals[1, ]
```

Store the data of x and y in a new data.frame that will be used to generate a plot using the ggplot function from the ggplot2package

```
ggplotData <- data.frame(x,y)
```

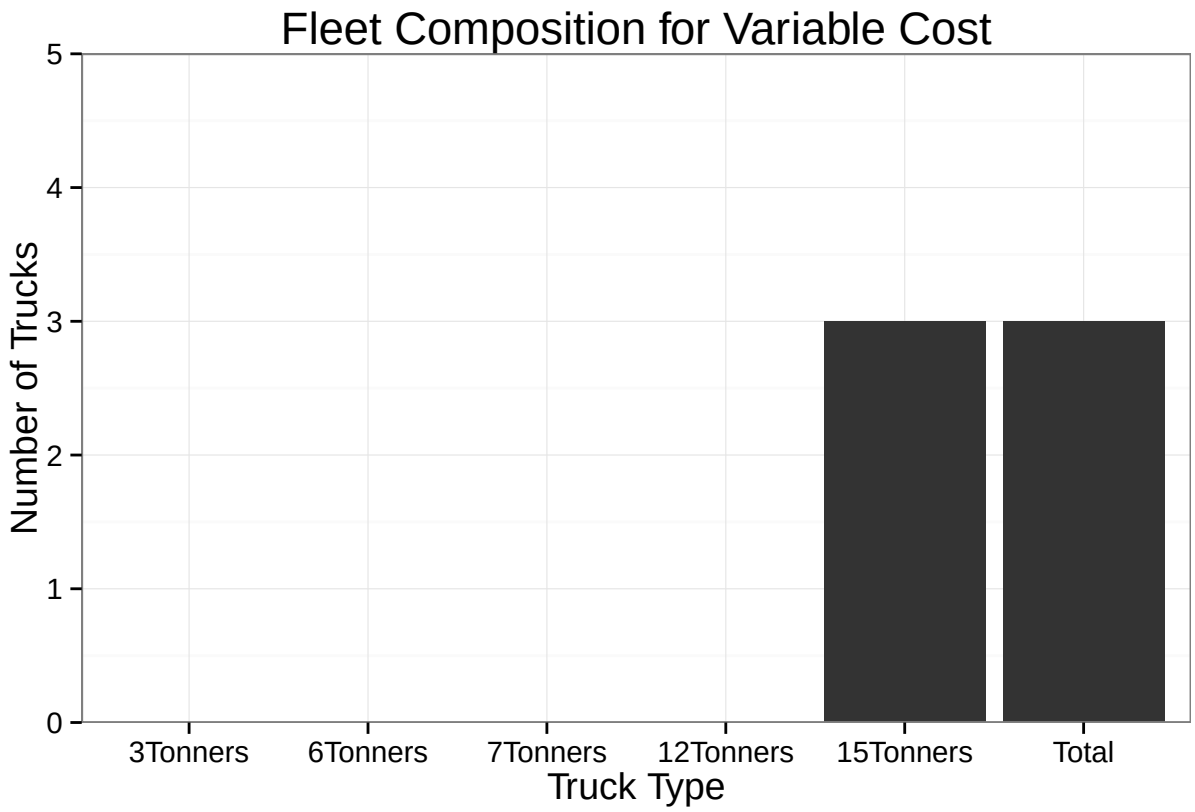
Plot the data to show the fleet composition when running-fixed-cost and variable cost input parameters are used.

```
g <- ggplot(data = ggplotData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Truck Type") + ylab("Number of Trucks")
g <- g + theme_bw(base_size = 14) + ggtitle("Fleet Composition for Running-Fixed-Cost \nand Variable Co
g1 <- g + coord_cartesian(ylim = c(0, 5)) + scale_y_continuous(breaks = seq(0,
    5, 1))
print(g1)
```



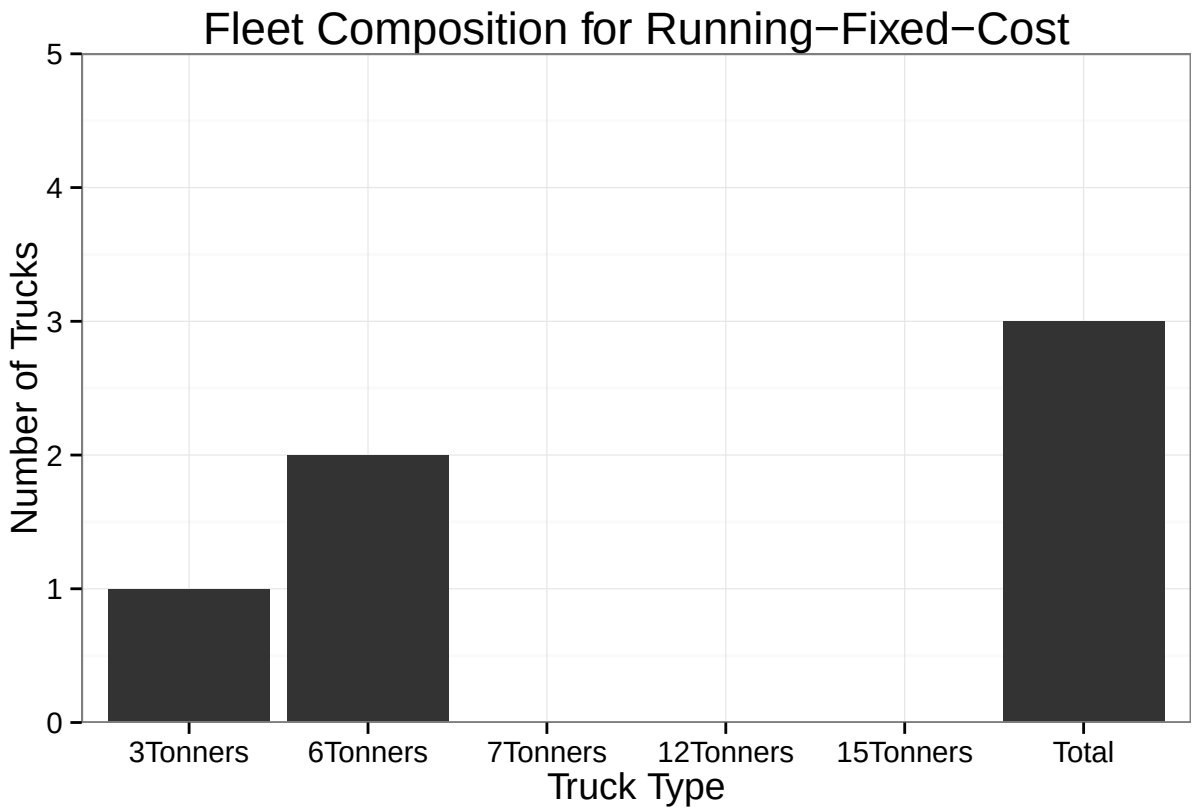
Plot fleet composition for variable cost with ggplot2 package

```
x <- CollumnNames
x <- as.character(x)
x <- factor(x, levels = unique(x))
y <- TruckTotals[2, ]
ggplotData <- data.frame(x, y)
g <- ggplot(data = ggplotData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Truck Type") + ylab("Number of Trucks")
g <- g + theme_bw(base_size = 14) + ggtitle("Fleet Composition for Variable Cost")
g2 <- g + coord_cartesian(ylim = c(0, 5)) + scale_y_continuous(breaks = seq(0,
    5, 1))
print(g2)
```



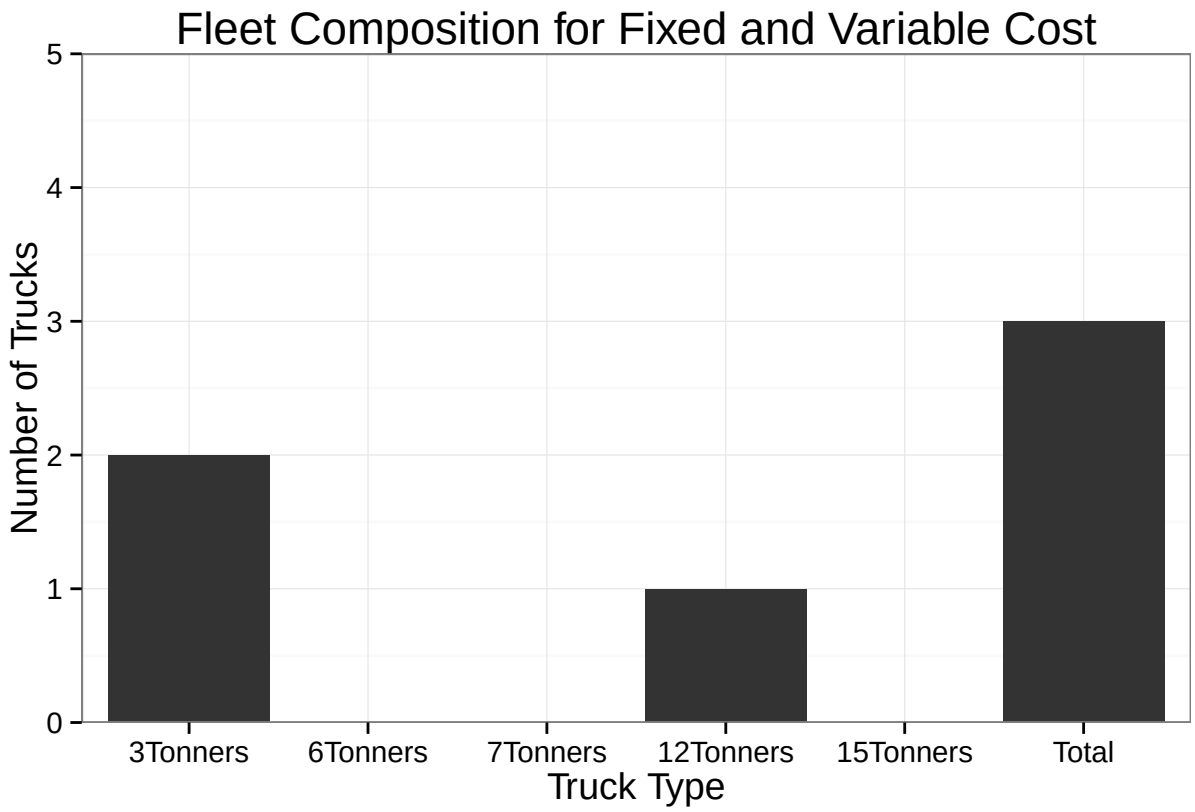
Plot fleet composition for running-fixed-cost with ggplot2 package

```
x <- CollumnNames
x <- as.character(x)
x <- factor(x, levels = unique(x))
y <- TruckTotals[3, ]
ggplotData <- data.frame(x, y)
g <- ggplot(data = ggplotData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Truck Type") + ylab("Number of Trucks")
g <- g + theme_bw(base_size = 14) + ggtitle("Fleet Composition for Running-Fixed-Cost")
g3 <- g + coord_cartesian(ylim = c(0, 5)) + scale_y_continuous(breaks = seq(0,
    5, 1))
print(g3)
```



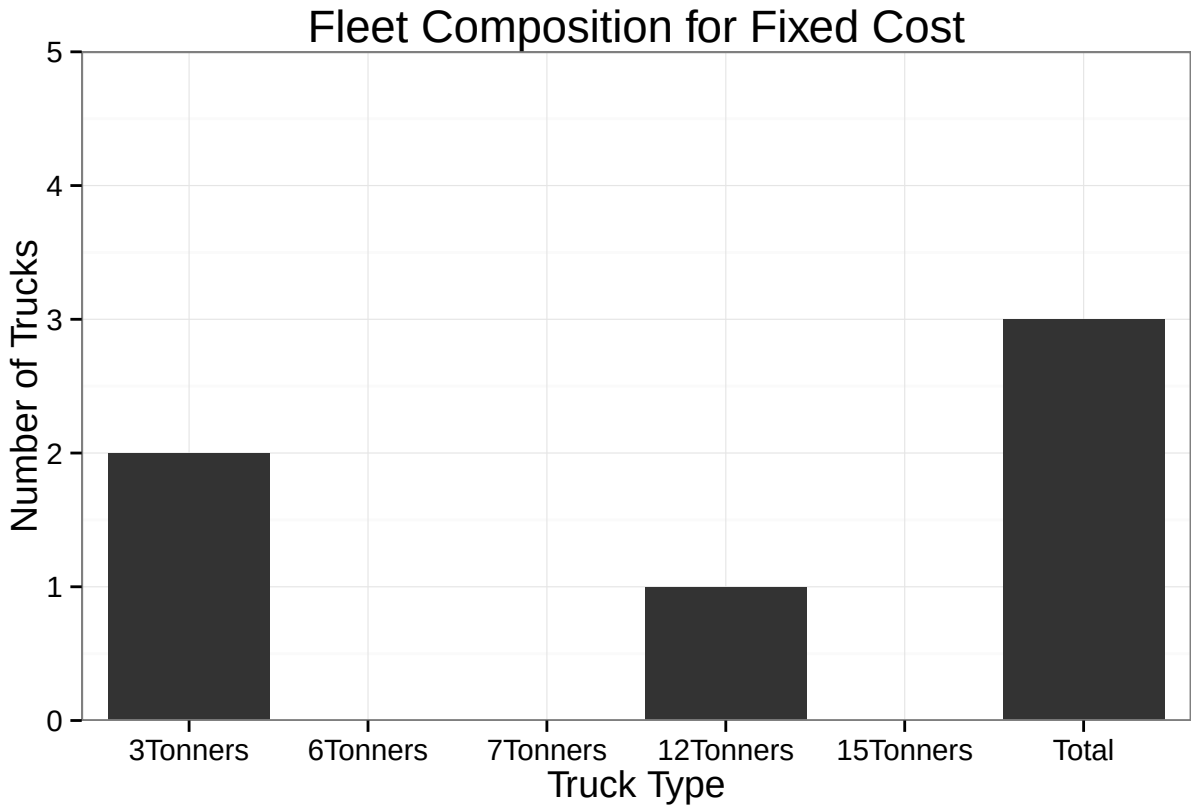
Plot fleet composition for fixed and variable cost with ggplot2 package

```
x <- CollumnNames
x <- as.character(x)
x <- factor(x, levels = unique(x))
y <- TruckTotals[4, ]
ggplotData <- data.frame(x, y)
g <- ggplot(data = ggplotData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Truck Type") + ylab("Number of Trucks")
g <- g + theme_bw(base_size = 14) + ggtitle("Fleet Composition for Fixed and Variable Cost")
g4 <- g + coord_cartesian(ylim = c(0, 5)) + scale_y_continuous(breaks = seq(0,
    5, 1))
print(g4)
```



Plot fleet composition for fixed cost with ggplot2 package

```
x <- CollumnNames
x <- as.character(x)
x <- factor(x, levels = unique(x))
y <- TruckTotals[5, ]
ggplotData <- data.frame(x, y)
g <- ggplot(data = ggplotData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Truck Type") + ylab("Number of Trucks")
g <- g + theme_bw(base_size = 14) + ggtitle("Fleet Composition for Fixed Cost")
g5 <- g + coord_cartesian(ylim = c(0, 5)) + scale_y_continuous(breaks = seq(0,
    5, 1))
print(g5)
```



Define the multiplot function

The multiplot function was obtained from http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/. It enables one to plot multiple plots generated with the ggplot2 package in one plot.

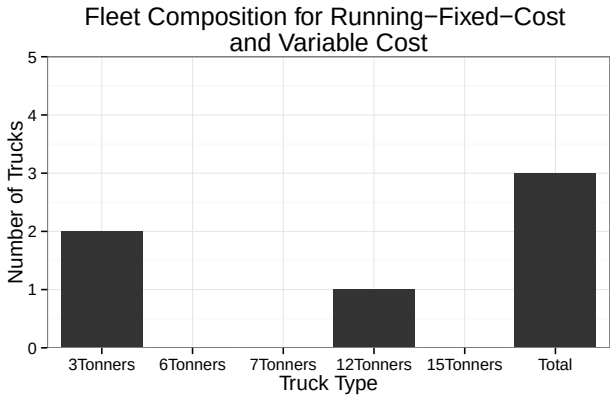
```
multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL) {
    require(grid)

    # Make a list from the ... arguments and plotlist
    plots <- c(list(...), plotlist)

    numPlots = length(plots)

    # If layout is NULL, then use 'cols' to determine layout
    if (is.null(layout)) {
        # Make the panel ncol: Number of columns of plots nrow: Number of rows
        # needed, calculated from # of cols
        layout <- matrix(seq(1, cols * ceiling(numPlots/cols)), ncol = cols,
            nrow = ceiling(numPlots/cols))
    }

    if (numPlots == 1) {
        print(plots[[1]])

    } else {
        # Set up the page
        grid.newpage()
```

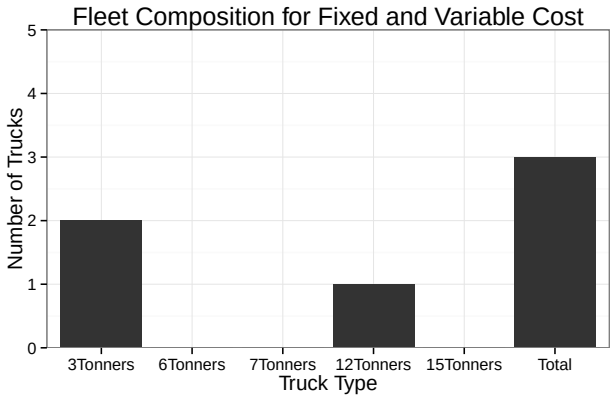
```
        pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

        # Make each plot, in the correct location
        for (i in 1:numPlots) {
            # Get the i,j matrix positions of the regions that contain this subplot
            matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

            print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row, layout.pos.col = matchidx$col
        }
    }
}
```

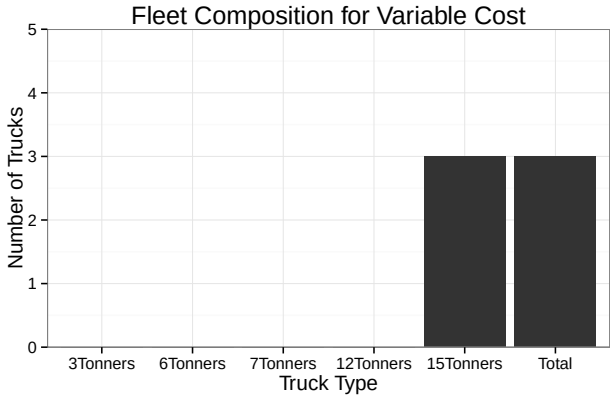
Plot the fleet compositions for various cost parameters

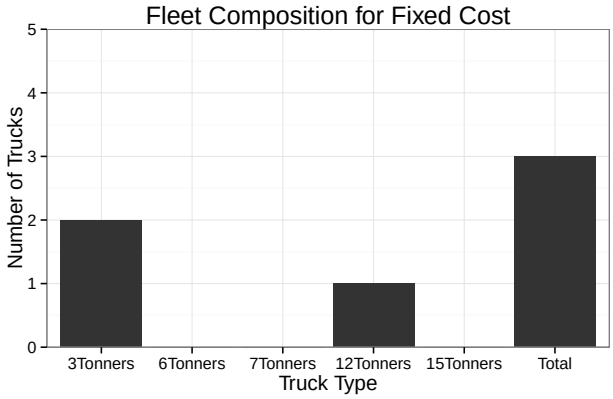
```
AllFC <- multiplot(g1, g2, g3, g4, g5, cols = 2)
```

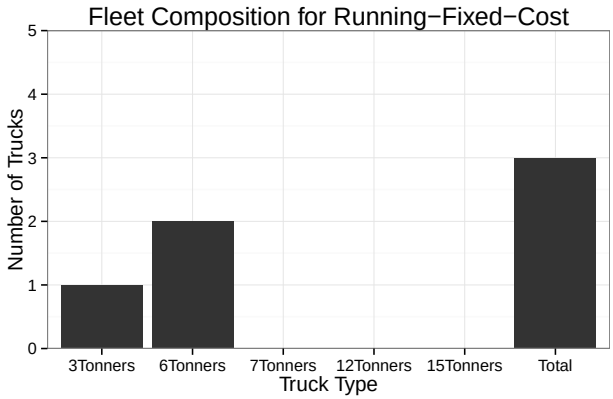
```
## Loading required package: grid
```











Total Cost

Create a data.frame which contains the total cost value for each model run. The character vector x specifies the the cost parameters used.

```
x <- c("Fixed Cost", "Fixed and Variable Cost", "Running-Fixed-Cost", "Running-Fixed-Cost and Variable C
    "Variable Cost")
```

y is a vector which shows the total cost for each set of cost parameters. The value extracted from the data frames containing the cost is in cent. It is divided by 100 to obtain the value in Rand. These values are also multiplied by a factor -1, because the output from "MyCarrierPlanInterpreter.java" gives a negative value for the cost, which is called "score" in the .csv file produced as output by "MyCarrierPlanInterpreter.java". To

obtain the cost only the first row of each data frame's score value is extracted, because each row gives the same value, which is the total cost.

```
FSMinfoFix
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 1   truck_6_1 -115825       53  362183         20329      16200
## 2 truck_3_1_0 -115825       31  412884         19722       9300
## 3 truck_3_1_1 -115825       30  388398         19400       9000
## 4   truck_3_1 -115825       54  413257         19754      16200
```

```
FSMinfoFixAndVar
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 1  truck_12_1 -559070      101   64020          3008      30900
## 2 truck_3_1_0 -559070       51  215192          8813      15300
## 3   truck_3_1 -559070       15  160805         11920       4500
```

```
FSMinfoRunFix
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 1   truck_6_1 -274124       90   81449          4157      30000
## 2   truck_3_1 -274124       15  160805         11920       4500
## 3 truck_6_1_0 -274124       54  193959          7771      16200
```

```
FSMinfoRunFixAndVar
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 1  truck_12_1 -759298       89   60788          3164      26700
## 2 truck_3_1_0 -759298       38   65717          2931      14100
## 3   truck_3_1 -759298       22  342276         18644       9900
```

```
FSMinfoVar
```

```
##          vehId   score numOfAct totDist totTravelTime totActTime
## 1 truck_15_1_0 -415943       58  194085          8031      17400
## 2   truck_15_1 -415943       94   71048          3306      28800
## 3 truck_15_1_1 -415943       15  160805         11920       4500
```

```
y <- c(FSMinfoFix$score[1]/100, FSMinfoFixAndVar$score[1]/100, FSMinfoRunFix$score[1]/100,
    FSMinfoRunFixAndVar$score[1]/100, FSMinfoVar$score[1]/100) * -1
y
```

```
## [1] 1158 5591 2741 7593 4159
```

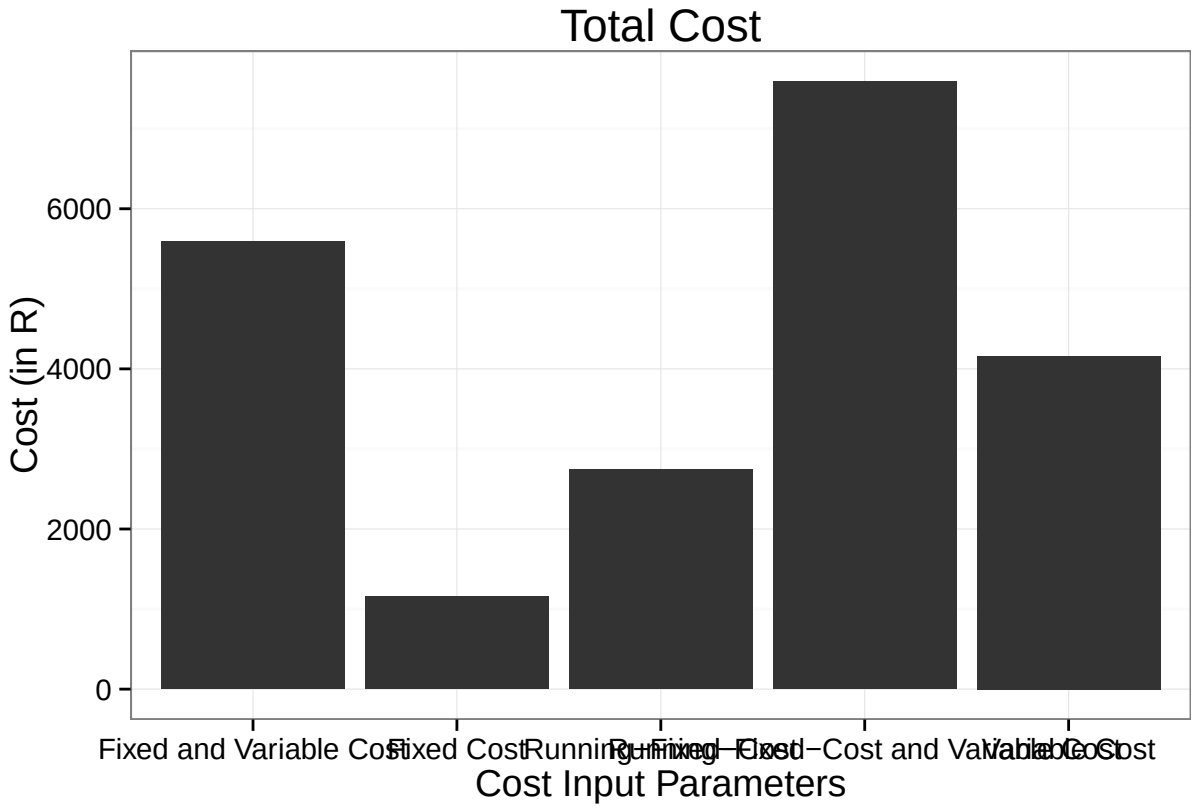
The total cost data is stored in a new data.frame

```
TotCostData <- data.frame(x,y)
TotCostData
```

```
##                                     x    y
## 1                         Fixed Cost 1158
## 2             Fixed and Variable Cost 5591
## 3                  Running-Fixed-Cost 2741
## 4 Running-Fixed-Cost and Variable Cost 7593
## 5                       Variable Cost 4159
```

Plotting the total cost data

```
g <- ggplot(data = TotCostData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Cost Input Parameters") + ylab("Cost (in R)")
g <- g + theme_bw(base_size = 14) + ggtitle("Total Cost")
print(g)
```



Generating a table in LaTeXcode for total cost data

```
xtable(TotCostData)
```

```
## % latex table generated in R 3.1.1 by xtable 1.7-3 package
## % Wed Aug 27 20:58:36 2014
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlr}
##   \hline
##  & x & y \\
##   \hline
## 1 & Fixed Cost & 1158.25 \\
##   2 & Fixed and Variable Cost & 5590.70 \\
##   3 & Running-Fixed-Cost & 2741.24 \\
##   4 & Running-Fixed-Cost and Variable Cost & 7592.98 \\
##   5 & Variable Cost & 4159.43 \\
```

```
##    \hline
## \end{tabular}
## \end{table}
```

Distance Travelled

Running-Fixed-Cost and Variable Cost

The information from for the running-fixed-cost model is used.

```
FSMinfoRunFixAndVar
```

```
##         vehId   score numOfAct totDist totTravelTime totActTime
## 1  truck_12_1 -759298       89   60788          3164      26700
## 2 truck_3_1_0 -759298       38   65717          2931      14100
## 3   truck_3_1 -759298       22  342276         18644       9900
```

This is the information that was read in from the .csv file. Let x be a vector that contains all the vehicles of the fleet specified by the solution to the FSMVRP.

```
x <- FSMinfoRunFixAndVar$vehId
x
```

```
## [1] truck_12_1  truck_3_1_0 truck_3_1
## Levels: truck_12_1 truck_3_1 truck_3_1_0
```

Let y be a vector containing the total distance that each vehicle has travelled.

```
y <- FSMinfoRunFixAndVar$totDist
```

Convert the distance from meters to *km*

```
y <- y/1000
y
```

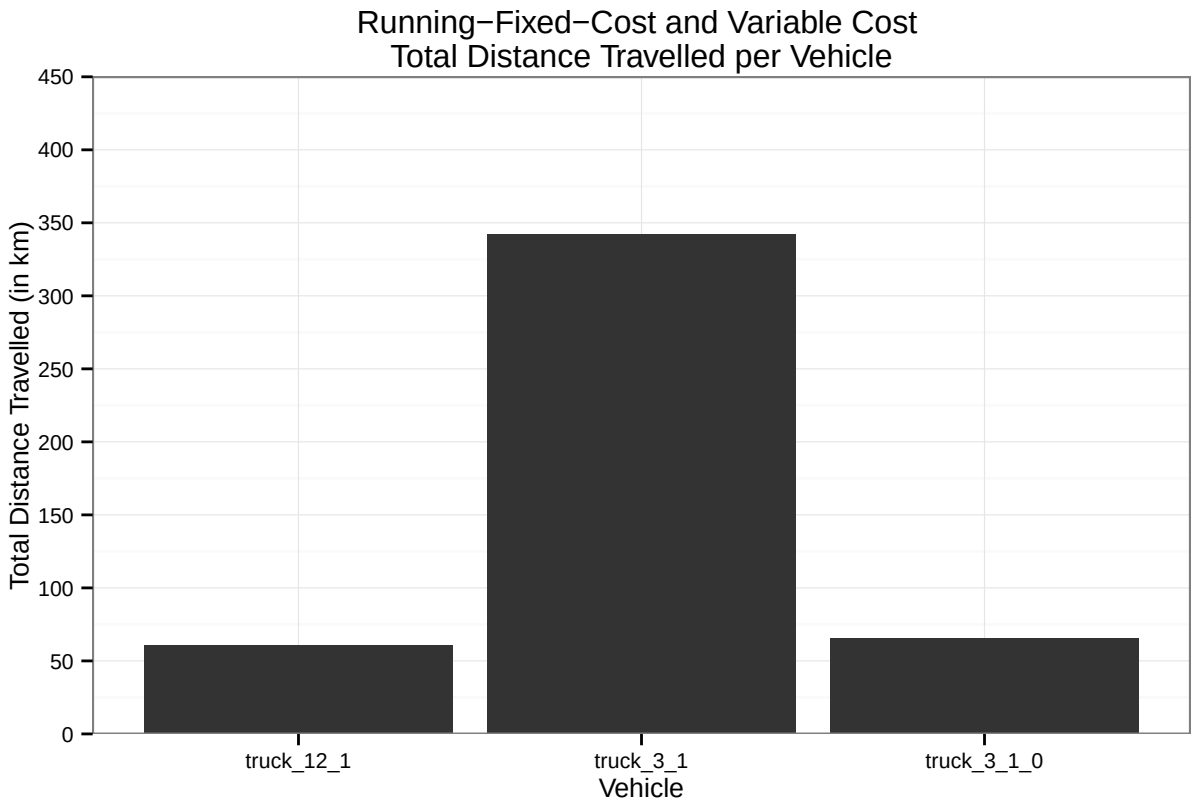
```
## [1]  60.79  65.72 342.28
```

Put the data for x and y in a data.frame called DistanceData so it can be plotted.

```
DistanceData <- data.frame(x,y)
```

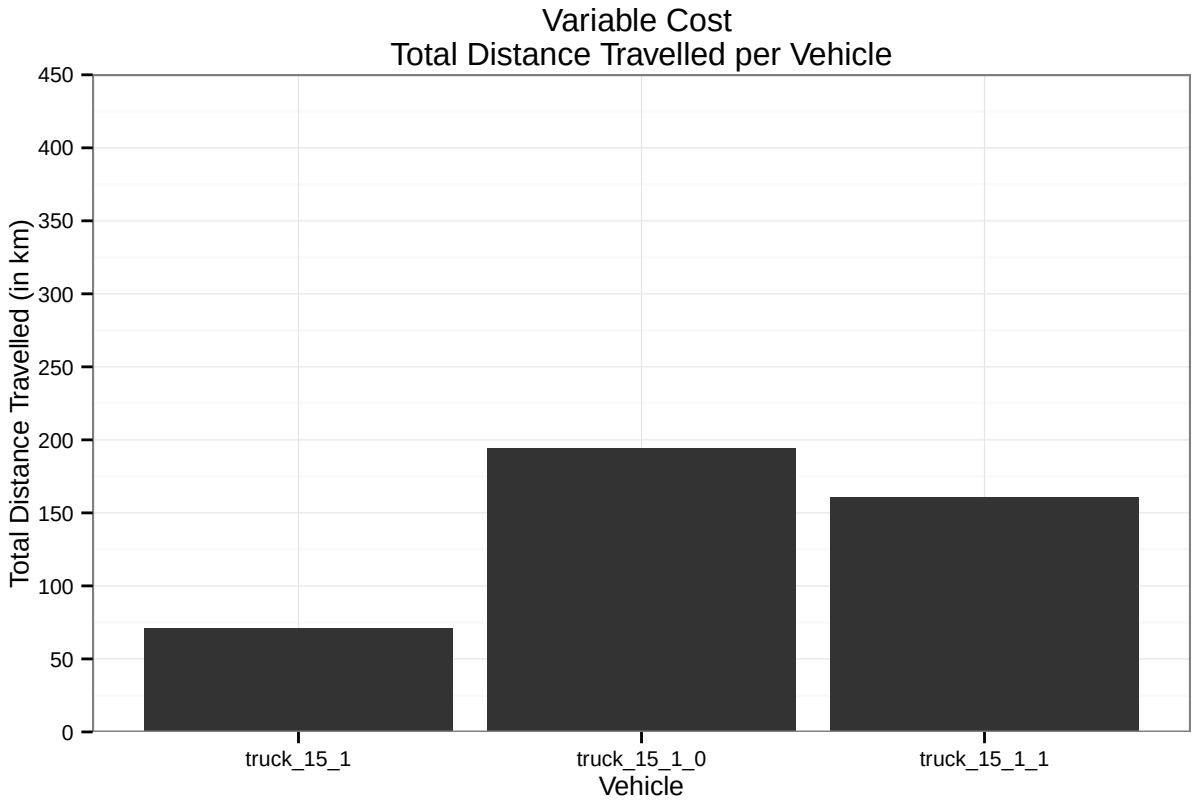
Plot the total distance travelled by each vehicle against that vehicle's name.

```
g <- ggplot(data = DistanceData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Vehicle") + ylab("Total Distance Travelled (in km)")
gD1 <- g + ggtitle("Running-Fixed-Cost and Variable Cost \nTotal Distance Travelled per Vehicle") +
    theme_bw(base_size = 10)
gD1 <- gD1 + coord_cartesian(ylim = c(0, 450)) + scale_y_continuous(breaks = seq(0,
    450, 50))
print(gD1)
```



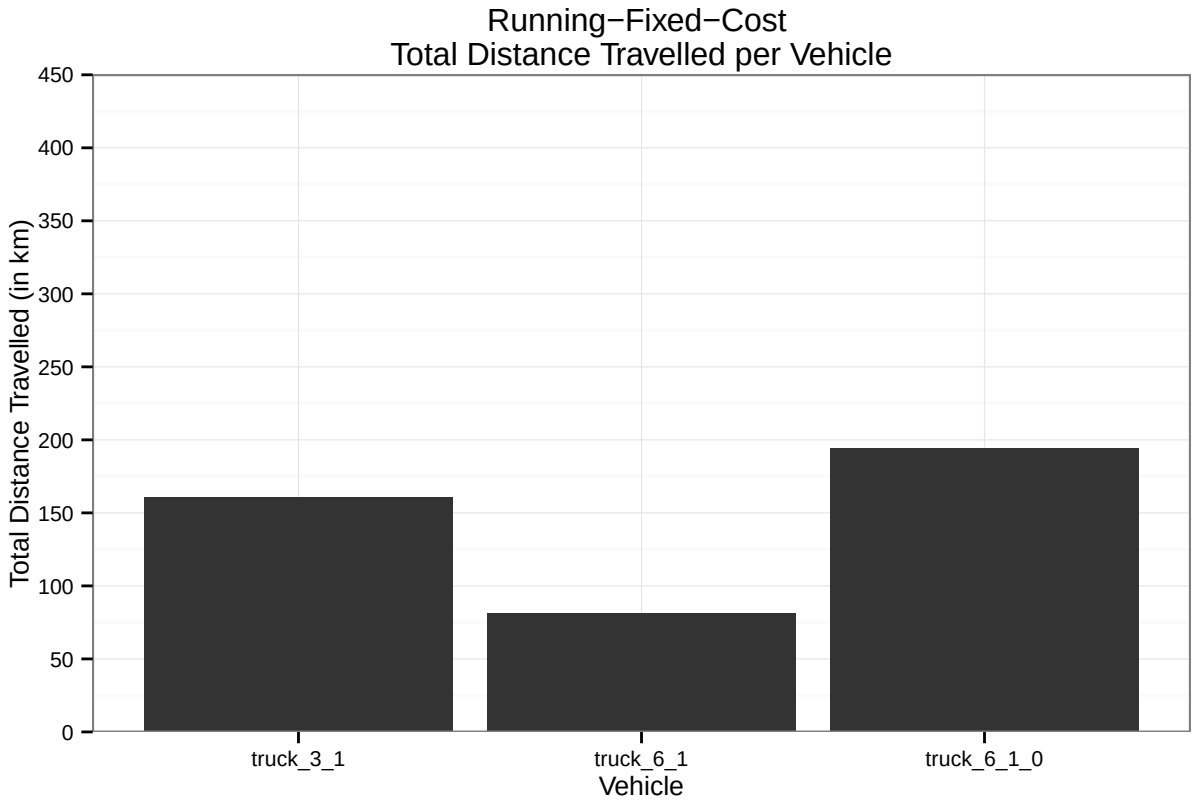
Variable Cost

```
x <- FSMinfoVar$vehId
y <- FSMinfoVar$totDist
y <- y/1000
DistanceData <- data.frame(x, y)
g <- ggplot(data = DistanceData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Vehicle") + ylab("Total Distance Travelled (in km)")
gD2 <- g + ggtitle("Variable Cost \nTotal Distance Travelled per Vehicle") +
    theme_bw(base_size = 10)
gD2 <- gD2 + coord_cartesian(ylim = c(0, 450)) + scale_y_continuous(breaks = seq(0,
    450, 50))
print(gD2)
```



Running-Fixed-Cost

```
x <- FSMinfoRunFix$vehId
y <- FSMinfoRunFix$totDist
y <- y/1000
DistanceData <- data.frame(x, y)
g <- ggplot(data = DistanceData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Vehicle") + ylab("Total Distance Travelled (in km)")
gD3 <- g + ggtitle("Running-Fixed-Cost \nTotal Distance Travelled per Vehicle") +
    theme_bw(base_size = 10)
gD3 <- gD3 + coord_cartesian(ylim = c(0, 450)) + scale_y_continuous(breaks = seq(0,
    450, 50))
print(gD3)
```



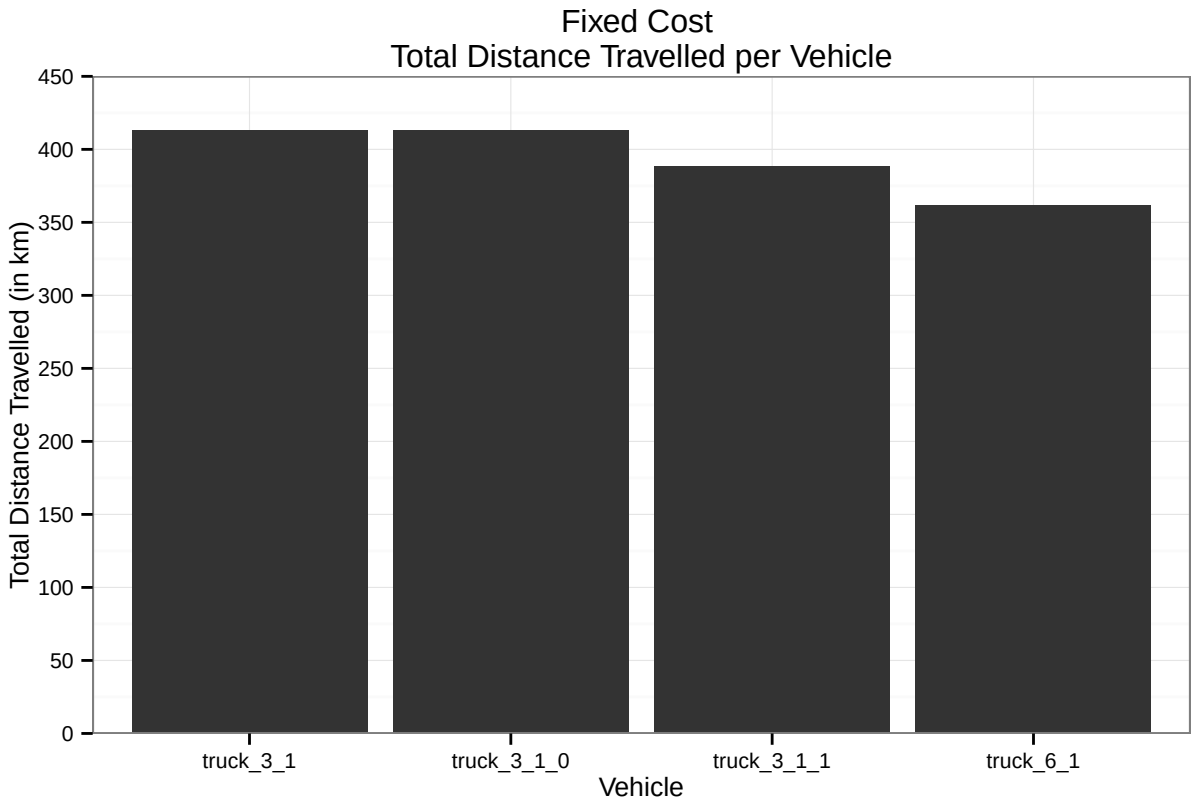
Fixed and Variable Cost

```
x <- FSMinfoFixAndVar$vehId
y <- FSMinfoFixAndVar$totDist
y <- y/1000  ##Convert y from meters to kilometres
DistanceData <- data.frame(x, y)
g <- ggplot(data = DistanceData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Vehicle") + ylab("Total Distance Travelled (in km)")
gD4 <- g + ggtitle("Fixed and Variable Cost \nTotal Distance Travelled per Vehicle") +
    theme_bw(base_size = 10)
gD4 <- gD4 + coord_cartesian(ylim = c(0, 450)) + scale_y_continuous(breaks = seq(0,
    450, 50))
pdf("VehicleDistanceFixAndVar.pdf")
print(gD4)
```

Fixed Cost

```
x <- FSMinfoFix$vehId
y <- FSMinfoFix$totDist
y <- y/1000  ##Convert y from meters to kilometres
DistanceData <- data.frame(x, y)
g <- ggplot(data = DistanceData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Vehicle") + ylab("Total Distance Travelled (in km)")
gD5 <- g + ggtitle("Fixed Cost \nTotal Distance Travelled per Vehicle") + theme_bw(base_size = 10)
```

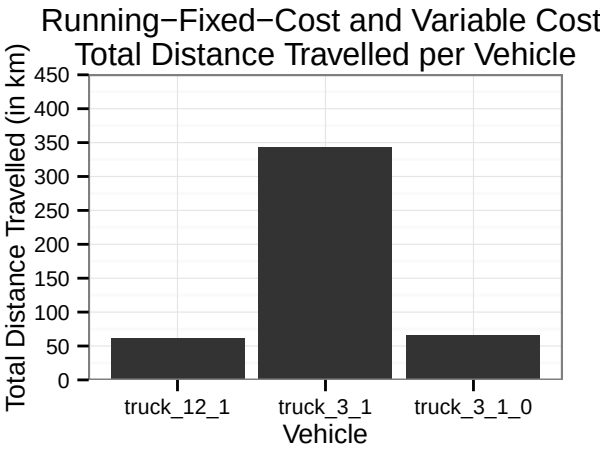
```
gD5 <- gD5 + coord_cartesian(ylim = c(0, 450)) + scale_y_continuous(breaks = seq(0,
    450, 50))
print(gD5)
```

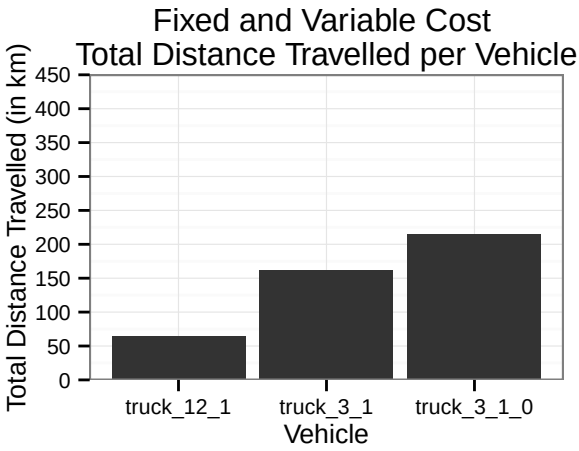


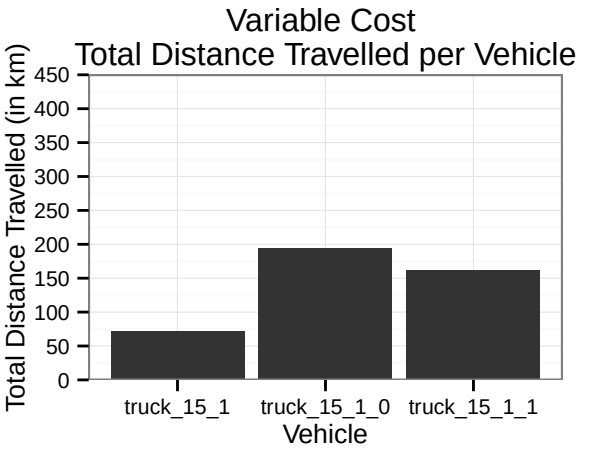
Combining Distance Plots

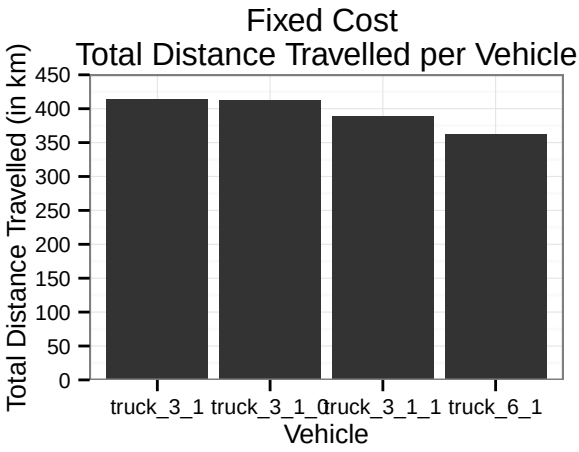
Plot distance travelled per vehicle for different cost parameters using the multiplot function.

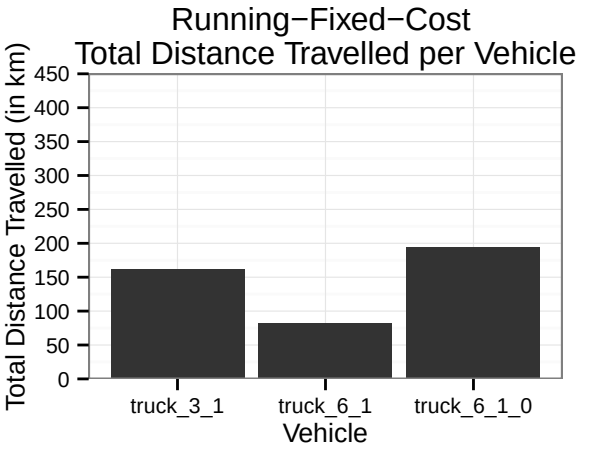
```
Alldist <- multiplot(gD1, gD2, gD3, gD4, gD5, cols = 2)
```











Total Distance for All Trucks

Plot total distance travelled for each set of cost parameters

Let x be a character vector containing all the sets of cost inputs.

```
x <- c("Fixed Cost", "Fixed and Variable Cost", "Running-Fixed-Cost", "Running-Fixed-Cost and Variable C
    "Variable Cost")
```

Let y be a numeric vector giving the total distance travelled by all vehicles for each set of cost parameters. The sum function sums the distance travelled by all the vehicles from the various data frames that contain the information for a single model run. Each value is divided by 1000 to convert from meters to *km*.

```
y <- c(sum(FSMinfoFix$totDist)/1000, sum(FSMinfoFixAndVar$totDist)/1000, sum(FSMinfoRunFix$totDist)/100
    sum(FSMinfoRunFixAndVar$totDist)/1000, sum(FSMinfoVar$totDist)/1000)
TotDistData <- data.frame(x, y)
g <- ggplot(data = TotDistData, aes(x = x, y = y)) + geom_bar(stat = "identity") +
    xlab("Cost Parameters") + ylab("Total Distance (in km)")
g <- g + theme_bw(base_size = 14) + ggtitle("Total Distance")
print(g)
```