Example WIC analysis using R Markdown

by Martin Brown, Martin.Brown@state.or.us

### Introduction

This RMarkdown document provides an example application of the Waste Impact Calculator (WIC) framework. Beginning with a solid waste stream that has been characterized only in terms of weight and end-of-life disposition (e.g. “50 tons of food waste composted”, “10 tons of landfilled”), it estimates the life cycle environmental impacts associated with that waste stream. It also compares the environmental impacts associated with various *alternative* scenarios for managing that waste (e.g. preventing food waste rather than composting it).

Along the way, this document provides pointers on how to properly combine, filter, and summarize WIC’s data tables, so that the final results actually represent the scenarios you are interested in evaluating. Before programming your own analyses, you should have a clear understanding of:

* how WIC calculates impacts for individual life cycle stages (see “Prerequisites”, below); and
* how file structures and joining commands (for example *left\_join* and *full\_join* in R’s dplyr package) can create sets of impacts that represent the materials life cycle.

### Intended audience

This document is oriented towards more technical users – those familiar with statistical or database operations. This document assumes a beginner-to-intermediate familiarity with the R language, especially the packages “base,” “dplyr”, and “ggplot2”. However, once WIC’s principles are understood, the WIC framework does not specifically require R.

### Prerequisites

* “Technical overview of the Waste Impact Calculator” describes the the purpose, limitations, and basic operation of WIC. It also documents the meaning of all the fields in each of WIC’s data tables – information that will, for the most part, not be repeated here.
* Experience with R as described above.

### Conventions

* the words *mass* and *weight* are used interchangeably. The distinction between these terms is important in other fields, but not in Earthbound waste management.
* weights are in short tons

### Management question and scenarios

This document represents the analysis of a real solid waste management inquiry from an Oregon city that will go by the name of Anytown. Anytown staff want to know about the life cycle impacts of yard debris and food waste under various management scenarios.

As of this writing, Anytown’s municipal waste service includes collection of yard debris, which is taken to a site 4 miles away and composted aerobically. None of Anytown’s food waste is composted – any food waste in collections is disposed of in a landfill 178 miles away.

Anytown is concerned with the impacts of food waste in landfills. It is interested allowing people to add their food waste to the yard debris bins, but there is a complication: if food waste is added, the merged material will need to be taken to a different site for composting, 78 miles away. It is unclear *how much* food waste would be added to yard debris, and how much would stay in regular trash collection and continue being landfilled, so higher and lower estimates of food waste collection are being considered.

One alternative strategy that Anytown staff could pursue, instead of food waste composting, is a sheer reduction in the generation of food waste. Reducing food waste generation would reduce impacts “upstream,” which could powerfully reduce impacts, but anecdotal experience suggests that reducing food waste generation is not easy. [The WRAP progam in the UK reduced household food waste by 6% over 3 years](https://www.wrap.org.uk/sites/files/wrap/Courtauld_Commitment_2025_Milestone_Progress_Report.pdf), so a 6 percent reduction in generation is an optimistic scenario. A 3 percent reduction might be more realistic.

Anytown staff want to know:

* Which materials and life cycle stages represent the biggest part of this system’s associated life cycle impacts?
* Are impact reductions from composting food waste substantial?
* Does the necessity of adding transport distance undermine the benefits of composting?
* How do the benefits of food waste generation reduction compare to composting?

Accordingly, Anytown staff have created five management scenarios, for which they want to calculate life cycle impacts.

* “baseline”: Gresham’s current management of food waste, where all yard debris, but zero food waste, is composted
* “compostFW1000”: where some food waste is added to yard debris and all composting happens at a different site than baseline
* “compostFW585”: where a lesser amount of food waste is added to yard debris and all composting happens at a different site than baseline
* “reduceFW03”: no change in management sites or methods, but generation of food waste is reduced by 3 percent
* “reduceFW06”: no change in management sites or method, but generation of food waste is reduced by 6 percent.

### Outline of the analysis

The analysis will proceed in this order:

* preparing the R workspace
* loading in the two source data tables, massProfiles and impactFactors
* calculating impacts and creating the master results data table, impactsInDetail
* checking for internal consistency of impactsInDetail
* creating summary statistics and graphics using both weight and impact perspectives

### Preparing the R workspace

# checking working directory  
getwd()

## [1] "C:/Users/mbrown2/Documents/Local repositories/wic3/wic-base/exampleAnalysis"

# loading packages useful for the analysis  
library(tidyverse) # many useful functions for data management

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.6 v dplyr 1.0.4  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.3

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

## Warning: package 'dplyr' was built under R version 4.0.3

## Warning: package 'forcats' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggthemes) # some themes for plotting

## Warning: package 'ggthemes' was built under R version 4.0.3

library(scales) # useful functions for labeling charts

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(knitr) # helps generate formatted output of various kinds

## Warning: package 'knitr' was built under R version 4.0.3

library(rmarkdown) # converts RMarkdown documents to other formats  
library(viridis) # nice & accessible color schemes

## Loading required package: viridisLite

##   
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':  
##   
## viridis\_pal

library(svglite) # helps write charts to SVG files

### Loading *massProfiles* and *impactFactors*

As you recall from *Technical overview of the Waste Impact Calculator*, the massProfiles table describes waste management scenarios by listing, in detail, the mass of each waste material going to specific end-of-life dispositions (e.g. landfilling, recycling), from areas of interest (“wastesheds”), as well as (optionally) setting transport distances for those end-of-life treatments. Different waste management ideas, or “scenarios”, are expressed as different numbers of tons going to different dispositions, and (optionally) different transport distances.

Anytown’s staff has created a massProfiles table representing their five scenarios. Here we load it in to an R data frame, and print it out in a formatted table.

# loading the mass profile data into an R data frame  
massProfiles <-  
 read.csv(  
 file = "massProfiles.csv",   
 header = TRUE,   
 stringsAsFactors = FALSE   
 )  
# a formatted printout  
kable(  
 massProfiles,   
 caption="a simple massProfiles table for 5 scenarios and 2 materials"  
)

a simple massProfiles table for 5 scenarios and 2 materials

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| scenario | wasteshed | material | disposition | umbDisp | tons | miles |
| baseline | Anytown | FoodWaste | landfilling | disposal | 7669 | 178 |
| baseline | Anytown | YardDebris | composting | recovery | 9000 | 5 |
| compostFW1000 | Anytown | FoodWaste | composting | recovery | 1000 | 78 |
| compostFW1000 | Anytown | FoodWaste | landfilling | disposal | 6669 | 178 |
| compostFW1000 | Anytown | YardDebris | composting | recovery | 9000 | 78 |
| compostFW585 | Anytown | FoodWaste | composting | recovery | 585 | 78 |
| compostFW585 | Anytown | FoodWaste | landfilling | disposal | 7084 | 178 |
| compostFW585 | Anytown | YardDebris | composting | recovery | 9000 | 78 |
| reduceFW03 | Anytown | FoodWaste | landfilling | disposal | 7439 | 178 |
| reduceFW03 | Anytown | YardDebris | composting | recovery | 9000 | 5 |
| reduceFW06 | Anytown | FoodWaste | landfilling | disposal | 7209 | 178 |
| reduceFW06 | Anytown | YardDebris | composting | recovery | 9000 | 5 |

Notice how the massProfiles table, in its compact way, provides ALL the details of the management scenarios described earlier. This is the only place in the base version of WIC that the user provides input to the system!

Here we list the fields in massProfiles:

str(massProfiles)

## 'data.frame': 12 obs. of 7 variables:  
## $ scenario : chr "baseline" "baseline" "compostFW1000" "compostFW1000" ...  
## $ wasteshed : chr "Anytown" "Anytown" "Anytown" "Anytown" ...  
## $ material : chr "FoodWaste" "YardDebris" "FoodWaste" "FoodWaste" ...  
## $ disposition: chr "landfilling" "composting" "composting" "landfilling" ...  
## $ umbDisp : chr "disposal" "recovery" "recovery" "disposal" ...  
## $ tons : int 7669 9000 1000 6669 9000 585 7084 9000 7439 9000 ...  
## $ miles : int 178 5 78 178 78 78 178 78 178 5 ...

As you recall, *tons* is the critical variable. This is a mass of some waste material, in short tons. All the other variables serve to identify or qualify where the *tons* came from, which disposition is being used, etc.

Note that the technical disposition of the material (landfilling or composting) is recorded independently of its legal classification (recovery or disposal) in the field umbDisp. The umbDisp field is provided as a convenience, so that users can calculate weight-based statistics such as diversion rates. However, impacts are always calculated based on the disposition name. The legal classification should have no effect on impact results.

Here are a few weight-based waste statistics for each scenario from massProfiles: the tons of waste generated, the tons of waste recovered, and the weight-based recovery rate, as a fraction.

massProfiles %>%  
 group\_by(scenario) %>%  
 summarize(  
 tonsGenerated=sum(tons),  
 recoveredTons=sum(ifelse(umbDisp=="recovery",tons,0))  
 ) %>%  
 mutate(  
 weightBasedRecoveryRate=round(recoveredTons/tonsGenerated,2)  
 ) %>%  
 kable()

|  |  |  |  |
| --- | --- | --- | --- |
| scenario | tonsGenerated | recoveredTons | weightBasedRecoveryRate |
| baseline | 16669 | 9000 | 0.54 |
| compostFW1000 | 16669 | 10000 | 0.60 |
| compostFW585 | 16669 | 9585 | 0.58 |
| reduceFW03 | 16439 | 9000 | 0.55 |
| reduceFW06 | 16209 | 9000 | 0.56 |

It appears that the scenario compostFW1000 has the highest weight-based recovery rate. Does that mean it will also represent the smallest life cycle impact? Stay tuned.

WIC’s other source data table is *impactFactors*. It contains environmental impact magnitudes for standard weights of solid waste materials, classified by disposition and life cycle stage.

Here we load the complete impactFactors table into an R data frame. Since it is thousands of records long, we print out only a small sample of it to see what it looks like.

impactFactorsAll <-  
 read.csv(  
 file = "../impactFactors/distributable/impactFactors.csv",   
 header = TRUE,   
 stringsAsFactors = FALSE #  
 )  
kable(impactFactorsAll %>% head(20))

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| material | LCstage | disposition | corporateSource | impactCategory | impactUnits | impliedMiles | impactCategoryLong | impactFactor | gabiExportDate | wicImportDate |
| AcceptedOtherSteel | endOfLife | incinerationNoER | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 85.14568 | 2020-11-06 | 2021-02-11 |
| AcceptedOtherSteel | endOfLife | landfilling | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 39.36742 | 2020-11-06 | 2021-02-11 |
| AcceptedOtherSteel | endOfLife | recyclingGeneric | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | -1455.27759 | 2020-11-06 | 2021-02-11 |
| AcceptedOtherSteel | endOfLifeTransport | incinerationNoER | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01883 | 2020-11-06 | 2021-02-11 |
| AcceptedOtherSteel | endOfLifeTransport | landfilling | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01883 | 2020-11-06 | 2021-02-11 |
| AcceptedOtherSteel | endOfLifeTransport | recyclingGeneric | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01883 | 2020-11-06 | 2021-02-11 |
| AcceptedOtherSteel | production | production | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 2213.56286 | 2020-11-06 | 2021-02-11 |
| Aluminum | endOfLife | incinerationNoER | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 142.37133 | 2020-11-06 | 2021-02-11 |
| Aluminum | endOfLife | landfilling | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 39.36742 | 2020-11-06 | 2021-02-11 |
| Aluminum | endOfLife | recyclingGeneric | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | -2267.13376 | 2020-11-06 | 2021-02-11 |
| Aluminum | endOfLifeTransport | incinerationNoER | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01883 | 2020-11-06 | 2021-02-11 |
| Aluminum | endOfLifeTransport | landfilling | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01883 | 2020-11-06 | 2021-02-11 |
| Aluminum | endOfLifeTransport | recyclingGeneric | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01883 | 2020-11-06 | 2021-02-11 |
| Aluminum | production | production | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 4724.99869 | 2020-11-06 | 2021-02-11 |
| AsepticContainers | endOfLife | incinerationER | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 210.11534 | 2020-11-06 | 2021-02-11 |
| AsepticContainers | endOfLife | landfilling | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 388.83016 | 2020-11-06 | 2021-02-11 |
| AsepticContainers | endOfLife | recyclingGeneric | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | -578.46351 | 2020-11-06 | 2021-02-11 |
| AsepticContainers | endOfLifeTransport | incinerationER | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01006 | 2020-11-06 | 2021-02-11 |
| AsepticContainers | endOfLifeTransport | landfilling | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01006 | 2020-11-06 | 2021-02-11 |
| AsepticContainers | endOfLifeTransport | recyclingGeneric | IPCC AR5 | GWP 100 | kg CO2 eq. | 180 | GWP100, excl biogenic carbon | 41.01006 | 2020-11-06 | 2021-02-11 |

Many of these records, and some of the diagnostic fields such as *corporateSource* and *gabiExportDate*, are unnecessary for the current analysis. Let’s cut impactFactors down to a manageable size – using only the materials and dispositions that we need, and only two impact categories.

# learning the materials in the massProfiles table and   
# saving them as a vector of string values  
materialNamesToUse <-  
 massProfiles %>%  
 select(material) %>%  
 distinct() %>%  
 pull(material)  
# doing the same thing for disposition names  
dispositionNamesToUse <-  
 massProfiles %>%  
 select(disposition) %>%  
 distinct() %>%  
 pull(disposition)  
# need to keep production impacts too  
dispositionNamesToUse <-  
 c("production",dispositionNamesToUse)   
  
# for the sake of brevity in printouts for this   
# example analysis,  
# limiting the impactFactors to the materials and   
# dispositions in massProfiles, and two impact   
# categories. In regular usage there is no  
# need to do such filtering -- the impactFactors   
# data frame may be left complete.  
impactFactors <-  
 impactFactorsAll %>%  
 filter(  
 material %in% materialNamesToUse &  
 disposition %in% dispositionNamesToUse &  
 (impactCategory== "GWP 100" |   
 impactCategory=="Water consumption")  
 ) %>%  
 # there are also several diagnostic columns that may be   
 # removed for the sake of this example analysis.. see the  
 # technical overview for their meaning.  
 select(  
 -corporateSource, -impactCategoryLong,   
 -gabiExportDate, -wicImportDate  
 ) %>%  
 # sorting it for easier reading  
 arrange(impactCategory, material, LCstage, disposition)  
# a formatted printout  
kable(impactFactors)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| material | LCstage | disposition | impactCategory | impactUnits | impliedMiles | impactFactor |
| FoodWaste | endOfLife | composting | GWP 100 | kg CO2 eq. | 180 | -70.38751 |
| FoodWaste | endOfLife | landfilling | GWP 100 | kg CO2 eq. | 180 | 343.67713 |
| FoodWaste | endOfLifeTransport | composting | GWP 100 | kg CO2 eq. | 180 | 41.01883 |
| FoodWaste | endOfLifeTransport | landfilling | GWP 100 | kg CO2 eq. | 180 | 41.01883 |
| FoodWaste | production | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 |
| YardDebris | endOfLife | composting | GWP 100 | kg CO2 eq. | 180 | -70.38751 |
| YardDebris | endOfLife | landfilling | GWP 100 | kg CO2 eq. | 180 | 575.69880 |
| YardDebris | endOfLifeTransport | composting | GWP 100 | kg CO2 eq. | 180 | 41.01883 |
| YardDebris | endOfLifeTransport | landfilling | GWP 100 | kg CO2 eq. | 180 | 41.01006 |
| YardDebris | production | production | GWP 100 | kg CO2 eq. | 180 | 14.18503 |
| FoodWaste | endOfLife | composting | Water consumption | kg | 180 | 158.51189 |
| FoodWaste | endOfLife | landfilling | Water consumption | kg | 180 | -250.75384 |
| FoodWaste | endOfLifeTransport | composting | Water consumption | kg | 180 | 108.31849 |
| FoodWaste | endOfLifeTransport | landfilling | Water consumption | kg | 180 | 108.31849 |
| FoodWaste | production | production | Water consumption | kg | 180 | 132157.09327 |
| YardDebris | endOfLife | composting | Water consumption | kg | 180 | 158.51189 |
| YardDebris | endOfLife | landfilling | Water consumption | kg | 180 | -507.70332 |
| YardDebris | endOfLifeTransport | composting | Water consumption | kg | 180 | 108.31849 |
| YardDebris | endOfLifeTransport | landfilling | Water consumption | kg | 180 | 108.29532 |
| YardDebris | production | production | Water consumption | kg | 180 | -1089.87390 |

As you recall from *Technical overview of the Waste Impact Calculator*, the critical field in this table is *impactFactor*. This number expresses an environmental impact for a particular mass of a particular material in a particular life cycle stage. All the other variables in each record identify or qualify the impact factor somehow – e.g. name the material, label its units, etc.

The impactFactors data frame should have EXACTLY one record for each combination of material, life cycle stage, disposition, and impactCategory of interest. Though impactFactor tables provided by Oregon DEQ should have this characteristic, you can check it if you desire, for example like this:

# checking for rows of impactFactors that might be duplicates  
# and printing a summary sentence  
print(  
 paste(  
 "There are ",  
 impactFactors %>%  
 group\_by(material, LCstage, disposition, impactCategory) %>%  
 summarise(myCount=n()) %>% # number of rows in each group  
 filter(myCount != 1) %>% # keep only rows where count <> 1  
 nrow(),  
 " rows in impactFactors that need to be inspected for duplicates.",  
 sep=""  
 )  
)

## `summarise()` has grouped output by 'material', 'LCstage', 'disposition'. You can override using the `.groups` argument.

## [1] "There are 0 rows in impactFactors that need to be inspected for duplicates."

### Merging the two tables to produce impactsInDetail.

massProfiles and impactsInDetail will be merged to allow us to calculate impacts, but before we do that we must address a limitation of massProfiles. So far massProfiles only includes tons of materials handled at the end-of-life phase of the life cycle. We must also account for the tons of those materials that are handled at two other life cycle stages: end-of-life transport and production.

We will add tonnages representing production here using a simple copy- and append operation. In the following code, all the cases from massProfiles are copied, labeled with a disposition (and umbDisp) of “production,” and then added back to massProfiles, creating a new data frame, massProfilesPlus.

# copy end-of-life tons and label them as production tons  
tempProductionMasses <-  
 massProfiles %>%  
 mutate(  
 disposition="production",  
 umbDisp="production",  
 miles=NA  
 )  
# add the production tons to the end-of-life tons  
massProfilesPlus <-  
 bind\_rows(  
 massProfiles,  
 tempProductionMasses  
 ) %>%  
 # sort the new, larger table  
 arrange(  
 scenario, wasteshed, material, disposition  
 )  
rm(tempProductionMasses) # remove temporary table

The resulting table, massProfilesPlus, should have exactly twice the total tonnage of massProfiles. Moreover, within each scenario, production tons should have the same sum as end-of-life tons. This can be checked…

print(  
 paste(  
 "Total tonnage in massProfiles is ",  
 sum(massProfiles$tons),  
 ".",  
 sep=""  
 )  
)

## [1] "Total tonnage in massProfiles is 82655."

print(  
 paste(  
 "Total tonnage in massProfilesPlus is ",  
 sum(massProfilesPlus$tons),  
 ".",  
 sep=""  
 )  
)

## [1] "Total tonnage in massProfilesPlus is 165310."

massProfilesPlus %>%   
 group\_by(scenario) %>%  
 summarise(  
 prodTons=sum(ifelse(umbDisp=="production",tons,0)),  
 eolTons=sum(ifelse(umbDisp!="production",tons,0))  
 ) %>%  
 print()

## # A tibble: 5 x 3  
## scenario prodTons eolTons  
## \* <chr> <dbl> <dbl>  
## 1 baseline 16669 16669  
## 2 compostFW1000 16669 16669  
## 3 compostFW585 16669 16669  
## 4 reduceFW03 16439 16439  
## 5 reduceFW06 16209 16209

The tonnages associated with end-of-life transport are still missing, but they will be generated during the following merge of massProfiles and impactFactors.

The merge is made on unique combinations of material and disposition name. However, since impactFactors has *two* life cycle stages (endOfLifeTransport and EndOfLife) in the field LCstage for each disposition name, records will effectively be added to represent endOfLifeTransport tons.

The merged file has both tons (from the massProfiles table) and impactFactor scaled to tons (from the impactFactors table), which can then be multiplied to get an impact in units of impactUnits.

Like so:

impactsInDetail <-  
 # joining all impact factors relevant to massProfiles  
 left\_join( # important: use left\_join not full\_join  
 massProfilesPlus,  
 impactFactors,  
 by = c("material", "disposition")  
 ) %>%  
 # calculating impacts with special considerations   
 # for end-of-life transport impacts  
 mutate(  
 # if miles is missing replace it with default value  
 miles = ifelse(is.na(miles), impliedMiles, miles),  
 # calculate impact   
 impact =  
 case\_when(  
 LCstage != "endOfLifeTransport" ~ tons\*impactFactor,  
 LCstage == "endOfLifeTransport" ~  
 tons\*(miles/impliedMiles)\*impactFactor  
 )  
 ) %>%  
 arrange(impactCategory, scenario, material, LCstage, disposition)

This creates a data frame, impactsInDetail, with records for each combination of scenario, wasteshed, material, LCstage, disposition, and impactCategory. A printout of this table is relatively lengthy:

kable(  
 impactsInDetail  
)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| scenario | wasteshed | material | disposition | umbDisp | tons | miles | LCstage | impactCategory | impactUnits | impliedMiles | impactFactor | impact |
| baseline | Anytown | FoodWaste | landfilling | disposal | 7669 | 178 | endOfLife | GWP 100 | kg CO2 eq. | 180 | 343.67713 | 2635659.89 |
| baseline | Anytown | FoodWaste | landfilling | disposal | 7669 | 178 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 311078.18 |
| baseline | Anytown | FoodWaste | production | production | 7669 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 13567195.48 |
| baseline | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -633487.57 |
| baseline | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 10254.71 |
| baseline | Anytown | YardDebris | production | production | 9000 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 14.18503 | 127665.26 |
| compostFW1000 | Anytown | FoodWaste | composting | recovery | 1000 | 78 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -70387.51 |
| compostFW1000 | Anytown | FoodWaste | landfilling | disposal | 6669 | 178 | endOfLife | GWP 100 | kg CO2 eq. | 180 | 343.67713 | 2291982.77 |
| compostFW1000 | Anytown | FoodWaste | composting | recovery | 1000 | 78 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 17774.83 |
| compostFW1000 | Anytown | FoodWaste | landfilling | disposal | 6669 | 178 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 270515.11 |
| compostFW1000 | Anytown | FoodWaste | production | production | 1000 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 1769095.77 |
| compostFW1000 | Anytown | FoodWaste | production | production | 6669 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 11798099.71 |
| compostFW1000 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -633487.57 |
| compostFW1000 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 159973.45 |
| compostFW1000 | Anytown | YardDebris | production | production | 9000 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 14.18503 | 127665.26 |
| compostFW585 | Anytown | FoodWaste | composting | recovery | 585 | 78 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -41176.69 |
| compostFW585 | Anytown | FoodWaste | landfilling | disposal | 7084 | 178 | endOfLife | GWP 100 | kg CO2 eq. | 180 | 343.67713 | 2434608.77 |
| compostFW585 | Anytown | FoodWaste | composting | recovery | 585 | 78 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 10398.27 |
| compostFW585 | Anytown | FoodWaste | landfilling | disposal | 7084 | 178 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 287348.78 |
| compostFW585 | Anytown | FoodWaste | production | production | 585 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 1034921.03 |
| compostFW585 | Anytown | FoodWaste | production | production | 7084 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 12532274.45 |
| compostFW585 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -633487.57 |
| compostFW585 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 159973.45 |
| compostFW585 | Anytown | YardDebris | production | production | 9000 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 14.18503 | 127665.26 |
| reduceFW03 | Anytown | FoodWaste | landfilling | disposal | 7439 | 178 | endOfLife | GWP 100 | kg CO2 eq. | 180 | 343.67713 | 2556614.16 |
| reduceFW03 | Anytown | FoodWaste | landfilling | disposal | 7439 | 178 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 301748.67 |
| reduceFW03 | Anytown | FoodWaste | production | production | 7439 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 13160303.45 |
| reduceFW03 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -633487.57 |
| reduceFW03 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 10254.71 |
| reduceFW03 | Anytown | YardDebris | production | production | 9000 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 14.18503 | 127665.26 |
| reduceFW06 | Anytown | FoodWaste | landfilling | disposal | 7209 | 178 | endOfLife | GWP 100 | kg CO2 eq. | 180 | 343.67713 | 2477568.42 |
| reduceFW06 | Anytown | FoodWaste | landfilling | disposal | 7209 | 178 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 292419.17 |
| reduceFW06 | Anytown | FoodWaste | production | production | 7209 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 1769.09577 | 12753411.43 |
| reduceFW06 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLife | GWP 100 | kg CO2 eq. | 180 | -70.38751 | -633487.57 |
| reduceFW06 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 180 | 41.01883 | 10254.71 |
| reduceFW06 | Anytown | YardDebris | production | production | 9000 | 180 | production | GWP 100 | kg CO2 eq. | 180 | 14.18503 | 127665.26 |
| baseline | Anytown | FoodWaste | landfilling | disposal | 7669 | 178 | endOfLife | Water consumption | kg | 180 | -250.75384 | -1923031.18 |
| baseline | Anytown | FoodWaste | landfilling | disposal | 7669 | 178 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 821464.55 |
| baseline | Anytown | FoodWaste | production | production | 7669 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 1013512748.29 |
| baseline | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLife | Water consumption | kg | 180 | 158.51189 | 1426607.04 |
| baseline | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 27079.62 |
| baseline | Anytown | YardDebris | production | production | 9000 | 180 | production | Water consumption | kg | 180 | -1089.87390 | -9808865.07 |
| compostFW1000 | Anytown | FoodWaste | composting | recovery | 1000 | 78 | endOfLife | Water consumption | kg | 180 | 158.51189 | 158511.89 |
| compostFW1000 | Anytown | FoodWaste | landfilling | disposal | 6669 | 178 | endOfLife | Water consumption | kg | 180 | -250.75384 | -1672277.35 |
| compostFW1000 | Anytown | FoodWaste | composting | recovery | 1000 | 78 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 46938.01 |
| compostFW1000 | Anytown | FoodWaste | landfilling | disposal | 6669 | 178 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 714349.60 |
| compostFW1000 | Anytown | FoodWaste | production | production | 1000 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 132157093.27 |
| compostFW1000 | Anytown | FoodWaste | production | production | 6669 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 881355655.02 |
| compostFW1000 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLife | Water consumption | kg | 180 | 158.51189 | 1426607.04 |
| compostFW1000 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 422442.10 |
| compostFW1000 | Anytown | YardDebris | production | production | 9000 | 180 | production | Water consumption | kg | 180 | -1089.87390 | -9808865.07 |
| compostFW585 | Anytown | FoodWaste | composting | recovery | 585 | 78 | endOfLife | Water consumption | kg | 180 | 158.51189 | 92729.46 |
| compostFW585 | Anytown | FoodWaste | landfilling | disposal | 7084 | 178 | endOfLife | Water consumption | kg | 180 | -250.75384 | -1776340.19 |
| compostFW585 | Anytown | FoodWaste | composting | recovery | 585 | 78 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 27458.74 |
| compostFW585 | Anytown | FoodWaste | landfilling | disposal | 7084 | 178 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 758802.30 |
| compostFW585 | Anytown | FoodWaste | production | production | 585 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 77311899.56 |
| compostFW585 | Anytown | FoodWaste | production | production | 7084 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 936200848.73 |
| compostFW585 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLife | Water consumption | kg | 180 | 158.51189 | 1426607.04 |
| compostFW585 | Anytown | YardDebris | composting | recovery | 9000 | 78 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 422442.10 |
| compostFW585 | Anytown | YardDebris | production | production | 9000 | 180 | production | Water consumption | kg | 180 | -1089.87390 | -9808865.07 |
| reduceFW03 | Anytown | FoodWaste | landfilling | disposal | 7439 | 178 | endOfLife | Water consumption | kg | 180 | -250.75384 | -1865357.80 |
| reduceFW03 | Anytown | FoodWaste | landfilling | disposal | 7439 | 178 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 796828.11 |
| reduceFW03 | Anytown | FoodWaste | production | production | 7439 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 983116616.84 |
| reduceFW03 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLife | Water consumption | kg | 180 | 158.51189 | 1426607.04 |
| reduceFW03 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 27079.62 |
| reduceFW03 | Anytown | YardDebris | production | production | 9000 | 180 | production | Water consumption | kg | 180 | -1089.87390 | -9808865.07 |
| reduceFW06 | Anytown | FoodWaste | landfilling | disposal | 7209 | 178 | endOfLife | Water consumption | kg | 180 | -250.75384 | -1807684.42 |
| reduceFW06 | Anytown | FoodWaste | landfilling | disposal | 7209 | 178 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 772191.67 |
| reduceFW06 | Anytown | FoodWaste | production | production | 7209 | 180 | production | Water consumption | kg | 180 | 132157.09327 | 952720485.39 |
| reduceFW06 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLife | Water consumption | kg | 180 | 158.51189 | 1426607.04 |
| reduceFW06 | Anytown | YardDebris | composting | recovery | 9000 | 5 | endOfLifeTransport | Water consumption | kg | 180 | 108.31849 | 27079.62 |
| reduceFW06 | Anytown | YardDebris | production | production | 9000 | 180 | production | Water consumption | kg | 180 | -1089.87390 | -9808865.07 |

Note that each line is labeled with the *umbDisp* from massProfiles, so distinctions can be made between recovery and disposal impacts or tonnages if desired.

### Checking the internal consistency of *impactsInDetail*

Before using the impactsInDetail table to calculate results, some basic quality checks should be performed.

For example, tonnages associated with all life cycles should have the same value within each scenario. That is, within each scenario, tons for “production” should be the same as tons for “endOfLifeTransport” and “endOfLife”. Code like the following can confirm that:

impactsInDetail %>%  
 group\_by(LCstage, scenario) %>%  
 summarise(tons=sum(tons)) %>%  
 arrange(scenario, LCstage) %>%  
 kable()

## `summarise()` has grouped output by 'LCstage'. You can override using the `.groups` argument.

|  |  |  |
| --- | --- | --- |
| LCstage | scenario | tons |
| endOfLife | baseline | 33338 |
| endOfLifeTransport | baseline | 33338 |
| production | baseline | 33338 |
| endOfLife | compostFW1000 | 33338 |
| endOfLifeTransport | compostFW1000 | 33338 |
| production | compostFW1000 | 33338 |
| endOfLife | compostFW585 | 33338 |
| endOfLifeTransport | compostFW585 | 33338 |
| production | compostFW585 | 33338 |
| endOfLife | reduceFW03 | 32878 |
| endOfLifeTransport | reduceFW03 | 32878 |
| production | reduceFW03 | 32878 |
| endOfLife | reduceFW06 | 32418 |
| endOfLifeTransport | reduceFW06 | 32418 |
| production | reduceFW06 | 32418 |

Note that tonnages in the table above are not identical to tonnages summarized earlier for the the massProfiles table. Besides the recent addition of production-related tons, and end-of-life transport tons, impactsInDetails has a complete set of tons for every impactCategory in use.

It is also valuable to check that every record has an impact factor. No impact factors should be missing, and any that are zero should be viewed with suspicion (because impactFactors of exactly zero are unlikely, and may represent a computation error or lazy assumption). In addition, impact and tons may be zero but should not be missing. These things can be checked with code like this:

impactsInDetail %>%   
 filter(is.na(impactFactor) | impactFactor==0) %>%  
 nrow()

## [1] 0

impactsInDetail %>%  
 filter(is.na(impact)) %>%  
 nrow()

## [1] 0

impactsInDetail %>%  
 filter(is.na(tons)) %>%  
 nrow()

## [1] 0

In each of these cases, the nrow() call has output 0. This means that our impactsInDetail table has passed these particular quality checks. If nrow() output >1, then it would be necessary to backtrack and correct something.

When impactsInDetail fails such simple internal-consistency checks, it is likely to be the result of mismatches between the massProfiles and impactFactors tables. Spellings of material and disposition names must match exactly, and every field in every table (with the exception of the *miles* field) must be filled in with a reasonable value.

### Creating tabular and graphical output

##### Guidelines

The impactsInDetail data frame is the source of all future output from this analysis. Most results of interest – for example, the total waste tonnages and total impacts linked to each scenario – are simple summations of tons or impacts from impactsInDetail.

When creating results from impactsInDetail, recall that:

* there is much redundancy in this data table now: records representing every combination of scenario, wasteshed, material, LCstage, disposition, and impactCategory. So data must be filtered down to the desired specific content to avoid miscalculation.
* when tons are summed, they should be restricted to tons marked with the “endOfLife” LCstage. The tons that appear in other LCstages are redundant and only serve for the calculation of the impacts of those stages.
* furthermore, when tons are summed, they should be restricted to a single impact category (it should not matter which) – as the complete set of tonnages has been repeated for every impact category.
* impacts should be summed only within a single impactCategory – unless users are willing to create, program, and defend a method for normalizing and/or summarizing across multiple impact categories.

##### Some utility objects

For the purpose of creating charts and tables, a few miscellaneous objects could be useful:

* a plaintext list of material names, sorted in descending order of abundance. (While the current example analysis has only 2 materials, most WIC analysese will be considerably more involved.)
* a table of likely impact category labels. (Impact categories like “Energy demand” do not currently include physical units, such as “MJ” for megajoules. An impact category label would merge those for use on chart axes.)
* a graphical theme for charts
* an ordered list of scenario names

Creating those things…

# most abundant materials in the wastestream, in order  
materialSortOrder <-  
 massProfiles %>%  
 group\_by(material) %>%  
 summarise(tons=sum(tons)) %>%  
 arrange(desc(tons)) %>%  
 pull(material)  
# a table of impact categories combined with units  
# (for use in chart labels)  
impactLabels <-  
 impactFactors %>%  
 select(impactCategory, impactUnits) %>%  
 distinct() %>%  
 mutate(  
 impactLabel =   
 paste(  
 impactCategory,  
 " (",  
 impactUnits,  
 ")",  
 sep=""  
 )  
 )  
# a custom graphic theme for charts, inspired by   
# the fivethirtyeight theme  
theme\_539 <- function() {  
 theme\_fivethirtyeight() +  
 theme(  
 rect=element\_rect(fill="transparent"),  
 panel.grid = element\_blank(),  
 axis.ticks = element\_line()  
 )  
}  
  
# making an ordered list of scenarios, where "baseline" is first  
scenarioOrder <-   
 c(  
 "baseline",  
 setdiff(  
 massProfiles %>%   
 select(scenario) %>%   
 distinct() %>%   
 pull(scenario),  
 "baseline")  
 )

##### Weights of waste in each of the scenarios

Note that when weights are summed, only the “endOfLife” LCstage is used, and only a single impactCategory is used.

# summing weights by disposition for each scenario  
tempWeightData1 <-   
 impactsInDetail %>%  
 filter(  
 LCstage == "endOfLife" & impactCategory=="GWP 100"  
 ) %>% # correct set for weight calculations  
 group\_by(scenario, disposition) %>%   
 summarise(tons=sum(tons)) %>%  
 ungroup() %>%  
 filter(tons != 0) %>%  
 mutate(scenario= factor(scenario, levels=rev(scenarioOrder)))

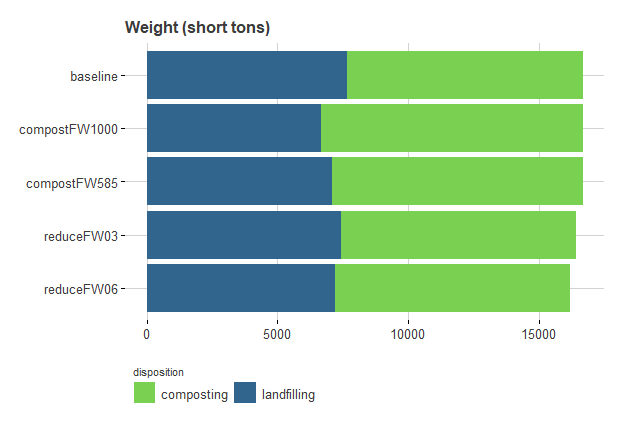
## `summarise()` has grouped output by 'scenario'. You can override using the `.groups` argument.

kable(tempWeightData1)

|  |  |  |
| --- | --- | --- |
| scenario | disposition | tons |
| baseline | composting | 9000 |
| baseline | landfilling | 7669 |
| compostFW1000 | composting | 10000 |
| compostFW1000 | landfilling | 6669 |
| compostFW585 | composting | 9585 |
| compostFW585 | landfilling | 7084 |
| reduceFW03 | composting | 9000 |
| reduceFW03 | landfilling | 7439 |
| reduceFW06 | composting | 9000 |
| reduceFW06 | landfilling | 7209 |

making that weight data into a chart…

tempWeightChart1 <-  
 ggplot()+  
 ggtitle("Weight (short tons)")+  
 theme\_539()+  
 geom\_bar(  
 data = tempWeightData1,  
 aes(x = scenario, y= tons, fill= disposition),  
 color=NA,  
 stat="identity"  
 )+  
 scale\_fill\_viridis(begin=0.32, end=0.8, discrete = TRUE,   
 direction = -1)+  
 coord\_flip()+  
 guides(fill=guide\_legend(ncol=2, title.position = "top"))+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 plot.title = element\_text(size=12),  
 legend.position="bottom",  
 legend.title = element\_text(size=8),  
 legend.justification="left"  
 )  
# printing the chart to the current device  
tempWeightChart1



# saving the chart as external files  
ggsave("chart\_output/weights.png")

## Saving 6.5 x 4.5 in image

ggsave("chart\_output/weights.svg")

## Saving 6.5 x 4.5 in image

## Warning: package 'gdtools' was built under R version 4.0.3

That chart shows that most of the scenarios are similar in terms of total weight, and management changes between scenarios are modest.

The same chart can be produced with individual materials separated.

# summing weights by disposition for each scenario  
tempWeightData2 <-   
 impactsInDetail %>%  
 filter(  
 LCstage == "endOfLife" & impactCategory=="GWP 100"  
 ) %>% # correct set for weight calculations  
 group\_by(scenario, material, disposition) %>%   
 summarise(tons=sum(tons)) %>%  
 ungroup() %>%  
 filter(tons != 0) %>%  
 mutate(scenario= factor(scenario, levels=scenarioOrder))

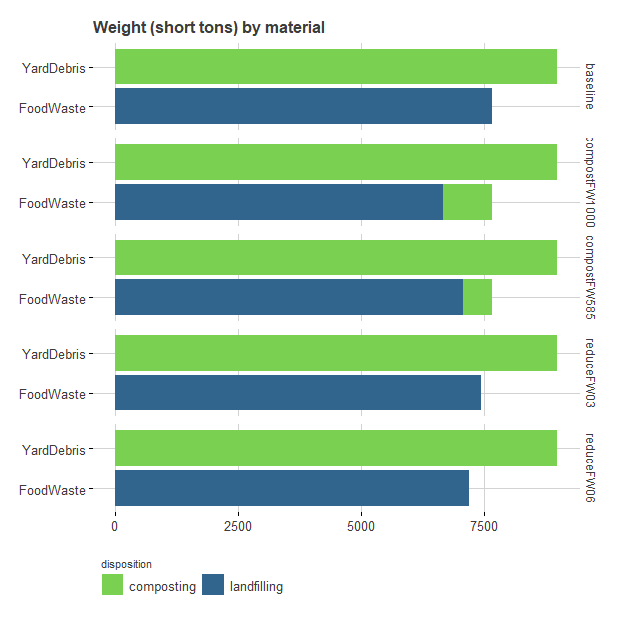
## `summarise()` has grouped output by 'scenario', 'material'. You can override using the `.groups` argument.

kable(tempWeightData2)

|  |  |  |  |
| --- | --- | --- | --- |
| scenario | material | disposition | tons |
| baseline | FoodWaste | landfilling | 7669 |
| baseline | YardDebris | composting | 9000 |
| compostFW1000 | FoodWaste | composting | 1000 |
| compostFW1000 | FoodWaste | landfilling | 6669 |
| compostFW1000 | YardDebris | composting | 9000 |
| compostFW585 | FoodWaste | composting | 585 |
| compostFW585 | FoodWaste | landfilling | 7084 |
| compostFW585 | YardDebris | composting | 9000 |
| reduceFW03 | FoodWaste | landfilling | 7439 |
| reduceFW03 | YardDebris | composting | 9000 |
| reduceFW06 | FoodWaste | landfilling | 7209 |
| reduceFW06 | YardDebris | composting | 9000 |

Making a weight chart with materials separated:

tempWeightChart2 <-  
 ggplot()+  
 ggtitle("Weight (short tons) by material")+  
 theme\_539()+  
 geom\_bar(  
 data = tempWeightData2,  
 aes(x = material, y= tons, fill= disposition),  
 color=NA,  
 stat="identity"  
 )+  
 scale\_fill\_viridis(begin=0.32, end=0.8, discrete = TRUE,   
 direction = -1)+  
 coord\_flip()+  
 facet\_grid(scenario~.)+  
 guides(fill=guide\_legend(ncol=2, title.position = "top"))+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 plot.title = element\_text(size=12),  
 legend.position="bottom",  
 legend.title = element\_text(size=8),  
 legend.justification="left",  
 strip.text=element\_text(size=9)  
 )  
# printing the chart to the current device  
tempWeightChart2



# saving the chart as external files  
ggsave("chart\_output/weightsInd.png")

## Saving 6.5 x 6.5 in image

ggsave("chart\_output/weightsInd.svg")

## Saving 6.5 x 6.5 in image

This chart demonstrates that the weights of the two materials are fairly close in each scenario, though the exact amounts and the amounts composted may change.

##### Life cycle impacts for waste in each scenario

Now for comparison, let’s look at the impacts associated with those scenarios. But here we will have to deal with more voluminous output, because there are multiple impact categories.

First, sum up the impacts in similar detail to the first weight chart:

tempImpactData1 <-  
 impactsInDetail %>%  
 group\_by(scenario, impactCategory, impactUnits) %>%  
 summarise(impact=sum(impact)) %>%  
 ungroup() %>%  
 mutate(  
 scenario = factor(scenario, levels = rev(scenarioOrder)),  
 impactLabel =   
 paste(  
 impactCategory,  
 " (",  
 impactUnits,  
 ")",  
 sep=""  
 )  
 )

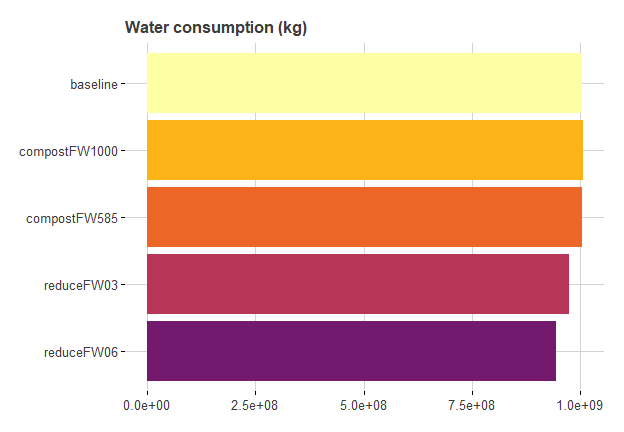
## `summarise()` has grouped output by 'scenario', 'impactCategory'. You can override using the `.groups` argument.

kable(tempImpactData1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| scenario | impactCategory | impactUnits | impact | impactLabel |
| baseline | GWP 100 | kg CO2 eq. | 16018366 | GWP 100 (kg CO2 eq.) |
| baseline | Water consumption | kg | 1004056003 | Water consumption (kg) |
| compostFW1000 | GWP 100 | kg CO2 eq. | 15731232 | GWP 100 (kg CO2 eq.) |
| compostFW1000 | Water consumption | kg | 1004800455 | Water consumption (kg) |
| compostFW585 | GWP 100 | kg CO2 eq. | 15912526 | GWP 100 (kg CO2 eq.) |
| compostFW585 | Water consumption | kg | 1004655583 | Water consumption (kg) |
| reduceFW03 | GWP 100 | kg CO2 eq. | 15523099 | GWP 100 (kg CO2 eq.) |
| reduceFW03 | Water consumption | kg | 973692909 | Water consumption (kg) |
| reduceFW06 | GWP 100 | kg CO2 eq. | 15027831 | GWP 100 (kg CO2 eq.) |
| reduceFW06 | Water consumption | kg | 943329814 | Water consumption (kg) |

For a single impactCategory, we can make an impact chart analagous to the weight chart:

# chose a single impactCategory at random  
tempImpactCategory <- sample\_n(impactLabels, 1)  
# get the impacts for that category  
tempImpactChart1 <-  
 ggplot()+  
 ggtitle(tempImpactCategory$impactLabel)+  
 theme\_539()+  
 geom\_bar(  
 data =   
 tempImpactData1 %>%   
 filter(impactCategory==tempImpactCategory$impactCategory),  
 aes(x = scenario, y= impact, fill=scenario),  
 color=NA,  
 stat="identity"  
 )+  
 scale\_fill\_viridis(begin=0.32, end=1, discrete = TRUE, option="B")+  
 coord\_flip()+  
 guides(fill=guide\_legend(ncol=2, title.position = "top"))+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 plot.title = element\_text(size=12),  
 legend.position="none",  
 legend.title = element\_text(size=8),  
 legend.justification="left"  
 )  
tempImpactChart1



ggsave("chart\_output/impacts1.png")

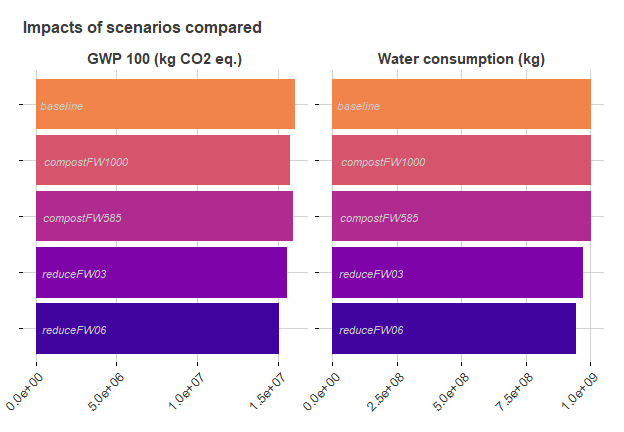
## Saving 6.5 x 4.5 in image

ggsave("chart\_output/impacts1.svg")

## Saving 6.5 x 4.5 in image

If we use a more complex layout, we can create a single image with facets expressing all of the available impact categories:

tempImpactChart1 <-  
 ggplot()+  
 ggtitle("Impacts of scenarios compared")+  
 theme\_539()+  
 geom\_bar(  
 data = tempImpactData1,  
 aes(x = scenario, y= impact, fill=scenario),  
 color=NA,  
 # size=2,  
 stat="identity"  
 )+  
 geom\_text(  
 data=tempImpactData1,  
 aes(x=scenario, y=0, label=scenario),  
 color="gray80",  
 size=3,  
 fontface="italic",  
 hjust=-0.1  
 )+  
 facet\_wrap(~impactLabel, ncol=2, scales="free")+  
 scale\_fill\_viridis(  
 begin=0.1, end=0.7, discrete = TRUE, option="C"  
 )+  
 coord\_flip()+  
# guides(fill=guide\_legend(ncol=2, title.position = "top"))+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 plot.title = element\_text(size=12),  
 legend.position="none",  
 axis.text.x=element\_text(angle=45, hjust=1),  
 axis.text.y=element\_blank(),  
 strip.text = element\_text(size=11, face="bold")  
 )  
tempImpactChart1



ggsave("chart\_output/impacts2.png")

## Saving 6.5 x 4.5 in image

ggsave("chart\_output/impacts2.svg")

## Saving 6.5 x 4.5 in image

Such side-by-side results show that management scenarios do not affect all impact categories equally. In the chart above:

* composting realistic amounts of food waste (in scenarios compostFW100 and compostFW585) did reduce GWP slightly, but very slightly.
* the composting scenarios actually increased water consumption very slightly.
* reducing food waste looked to have more of an effect, but again, it was not dramatic.

To find out how this happened, let’s return to the original questions that Anytown’s staff had.

##### Question: which materials and life cycle stages represent the biggest part of this system’s associated life cycle impacts?

To answer this, we’ll make a chart comparing weights and impacts of individual materials, just for a single scenario… “compostFW1000”, which represents a relatively large addition of food waste to yard debris composting.

Unfortunately this takes a bit more coding than the previous charts.

tempWeightData4 <-  
 impactsInDetail %>%  
 filter(  
 LCstage == "endOfLife" & impactCategory=="GWP 100"   
 ) %>% # correct set for weight calculations  
 group\_by(wasteshed, scenario, material, umbDisp) %>%   
 summarise(tons=sum(tons)) %>%  
 ungroup() %>%  
 filter(tons != 0) %>%  
 mutate(scenario= factor(scenario, levels=rev(scenarioOrder))) %>%  
 mutate(recovTons=ifelse(umbDisp=="recovery",tons,0)) %>%  
 mutate(material=factor(material, levels=materialSortOrder))

## `summarise()` has grouped output by 'wasteshed', 'scenario', 'material'. You can override using the `.groups` argument.

# creating a data file that expresses weight-based  
# recovery rate for each scenario  
tempWeightData4a <-  
 tempWeightData4 %>%  
 group\_by(wasteshed, scenario, material) %>%  
 summarise(  
 recovTons=sum(recovTons),  
 tons=sum(tons)  
 ) %>%  
 ungroup() %>%  
 mutate(recovRate=recovTons/tons)

## `summarise()` has grouped output by 'wasteshed', 'scenario'. You can override using the `.groups` argument.

# total weight for the scenario  
tempWeightData4b <-  
 tempWeightData4a %>%  
 group\_by(wasteshed, scenario) %>%  
 summarise(allTons=sum(tons)) %>%  
 ungroup()

## `summarise()` has grouped output by 'wasteshed'. You can override using the `.groups` argument.

# combining that total  
tempWeightData4c <-  
 full\_join(  
 tempWeightData4a,  
 tempWeightData4b,  
 by=c("wasteshed", "scenario")  
 ) %>%  
 mutate(pctTons=tons/allTons)  
  
# adding impact categories  
tempWeightData4d <-  
 left\_join(  
 tempWeightData4c %>% mutate(dummy=1),  
 impactLabels %>% mutate(dummy=1),  
 by="dummy"  
 ) %>%  
 select(-dummy)

Now processing the impact data into parallel form…

tempImpactData4 <-  
 impactsInDetail %>%  
 group\_by(  
 wasteshed,  
 scenario,   
 material,  
 LCstage,   
 impactCategory, impactUnits  
 ) %>%  
 summarise(impact=sum(impact)) %>%  
 ungroup() %>%  
 mutate(  
 scenario = factor(scenario, levels = rev(scenarioOrder)),  
 impactLabel =   
 paste(  
 impactCategory,  
 " (",  
 impactUnits,  
 ")",  
 sep=""  
 ),  
 LCstage=  
 factor(  
 LCstage,  
 levels=c("production","endOfLife","endOfLifeTransport")  
 )  
 )

## `summarise()` has grouped output by 'wasteshed', 'scenario', 'material', 'LCstage', 'impactCategory'. You can override using the `.groups` argument.

tempImpactData4a <-  
 tempImpactData4 %>%  
 group\_by(  
 wasteshed,   
 scenario,   
 material,   
 impactCategory,   
 impactUnits,   
 impactLabel  
 ) %>%  
 summarise(impact=sum(impact)) %>%  
 ungroup()

## `summarise()` has grouped output by 'wasteshed', 'scenario', 'material', 'impactCategory', 'impactUnits'. You can override using the `.groups` argument.

tempImpactData4b <-  
 tempImpactData4a %>%  
 group\_by(  
 wasteshed, scenario, impactCategory, impactUnits, impactLabel  
 ) %>%  
 summarise(allImpact=sum(impact)) %>%  
 ungroup()

## `summarise()` has grouped output by 'wasteshed', 'scenario', 'impactCategory', 'impactUnits'. You can override using the `.groups` argument.

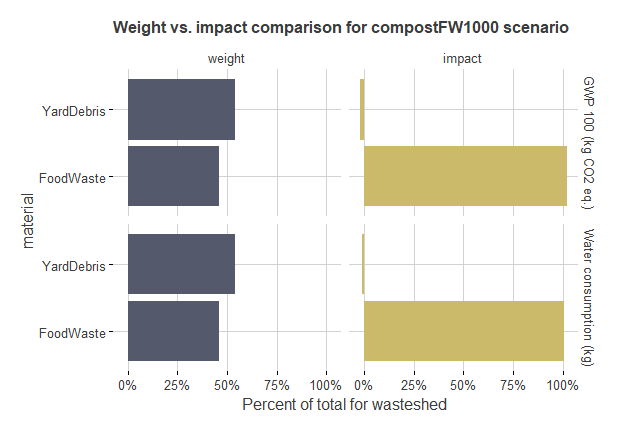
tempImpactData4c <-  
 full\_join(  
 tempImpactData4a,  
 tempImpactData4b,  
 by=c(  
 "wasteshed", "scenario", "impactCategory",  
 "impactUnits", "impactLabel"  
 )  
 ) %>%  
 mutate(pctImpact=impact/allImpact)

now lining those percentages up into a long, skinny file

weightImpactComparisonData1 <-  
 bind\_rows(  
 tempWeightData4d %>%  
 select(scenario, wasteshed, material, pctTons, impactLabel) %>%  
 rename(pctTotal=pctTons) %>%  
 mutate(dataType="weight"),  
 tempImpactData4c %>%  
 select(scenario, wasteshed, material, pctImpact, impactLabel) %>%  
 rename(pctTotal=pctImpact) %>%  
 mutate(dataType="impact")  
 ) %>%  
 arrange(wasteshed, scenario, material, impactLabel)

For the chart I’ll separate impact categories.

weightImpactComparisonData2 <-  
 filter(  
 weightImpactComparisonData1,  
 scenario=="compostFW1000"  
 ) %>%  
 mutate(dataType=factor(dataType, levels=c("weight", "impact")))  
ggplot()+  
 ggtitle("Weight vs. impact comparison for compostFW1000 scenario")+  
 theme\_539()+  
 geom\_bar(  
 data=weightImpactComparisonData2,  
 aes(x=material, y=pctTotal, fill=dataType),  
 stat="identity",  
 color=NA  
 )+  
 scale\_y\_continuous(  
 name="Percent of total for wasteshed",  
 labels=percent  
 )+  
 scale\_fill\_viridis(begin = 0.32, end=0.8,   
 discrete=TRUE,  
 option="cividis",  
 direction=1)+  
 facet\_grid(impactLabel~dataType)+  
 coord\_flip()+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 axis.ticks=element\_line(),  
 legend.position="none",  
 axis.title=element\_text(),  
 plot.title=element\_text(size=12)  
 )



ggsave("chart\_output/weight\_impact\_comparison.png")

## Saving 6.5 x 4.5 in image

ggsave("chart\_output/weight\_impact\_comparison.svg")

## Saving 6.5 x 4.5 in image

What this shows is that one material is dominating life cycle impacts: food waste. Despite the fact that yard debris is the majority of weight in the system, it contributes very little to the total impact.

##### Question: Are impact reductions from composting food waste substantial?

##### Question: How much do transport distances affect total results?

##### Question: How do the benefits of food waste generation reduction compare to composting?

We can answer these questions from Anytown’s staff by plotting the contributions of individual life cycle stages to total impacts for each material.

# calculating detailed impacts by scenario, material, and LCstage  
tempImpactData5 <-  
 impactsInDetail %>%  
 group\_by(  
 scenario, material, LCstage, impactCategory, impactUnits  
 ) %>%  
 summarise(impact=sum(impact)) %>%  
 ungroup() %>%  
 mutate(  
 scenario = factor(scenario, levels = rev(scenarioOrder)),  
 impactLabel =   
 paste(  
 impactCategory,  
 " (",  
 impactUnits,  
 ")",  
 sep=""  
 ),  
 LCstage=  
 factor(  
 LCstage,  
 levels=rev(c("production","endOfLife","endOfLifeTransport"))  
 )  
 )

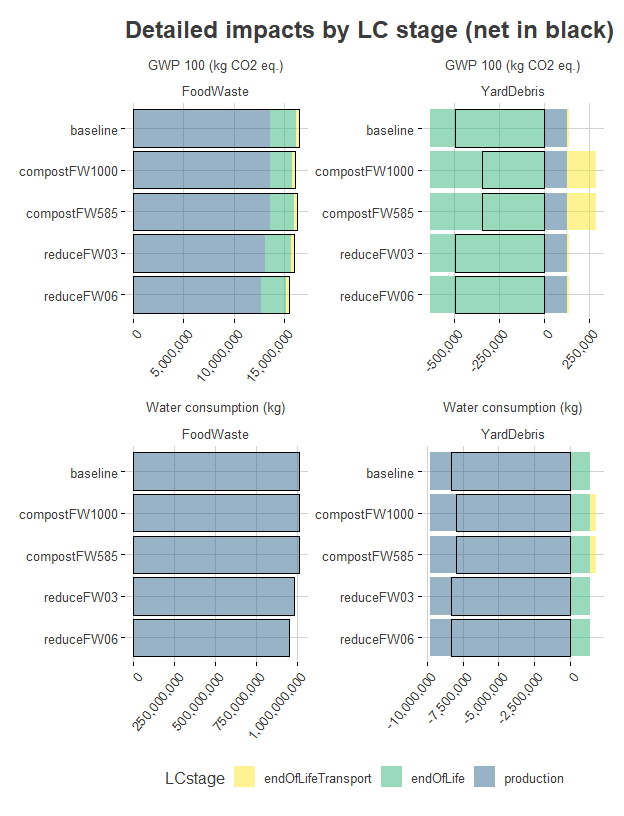
## `summarise()` has grouped output by 'scenario', 'material', 'LCstage', 'impactCategory'. You can override using the `.groups` argument.

# calculating the net totals by scenario and material  
tempImpactData5b <-  
 tempImpactData5 %>%  
 group\_by(scenario, material, impactCategory, impactUnits,  
 impactLabel) %>%  
 summarize(impact=sum(impact))

## `summarise()` has grouped output by 'scenario', 'material', 'impactCategory', 'impactUnits'. You can override using the `.groups` argument.

Now making the chart…

tempImpactChart5 <-  
 ggplot()+  
 ggtitle("Detailed impacts by LC stage (net in black)")+  
 theme\_539()+  
 geom\_bar(  
 data=tempImpactData5,  
 aes(x=scenario, y=impact, fill=LCstage),  
 color=NA,  
 stat="identity",  
 alpha=0.5  
 )+  
 geom\_bar(  
 data=tempImpactData5b,  
 aes(x=scenario, y=impact),  
 stat="identity",  
 fill=NA,  
 color="black"  
 )+  
 scale\_y\_continuous(labels=comma)+  
 coord\_flip()+  
 facet\_wrap(facets = c("impactLabel","material"), scales="free")+  
 scale\_fill\_viridis(begin=0.32, end=1, discrete = TRUE,   
 direction = -1)+  
 theme(  
 axis.text.x=element\_text(angle=50, hjust=1, vjust=1),  
 panel.background=element\_blank(),  
 panel.grid = element\_blank()  
 )  
tempImpactChart5

 A careful examination of this chart answers most of the questions from Anytown’s staff.

* For both water consumption and GWP 100, the scales of impacts associated with food waste are orders of magnitude larger. Food waste will dominate total impacts of the system, no matter what is done with yard debris.
* Considered alone, composting yard debris does seem to represent a net environmental benefit – see its negative “endOfLife” impact values for GWP 100. “Producing” yard debris is also related to water savings. But the scale of these benefits are utterly swamped by the high production impacts of food waste.
* Composting food waste does reduce the impacts of food waste compared to baseline, but not dramatically. It also increases the transportation impacts of yard debris, so that the net benefit of the composting scenarios is very small.
* Since food waste impacts are dominated by production impacts, the most powerful intervention is anything that reduces food waste generation.

##### Detailing weight by umbrella disposition

The charts above don’t have a lot of detail about waste management. Let’s redo the weight chart so that it characterizes the weight-based recovery rate in each scenario. For that, we sum weights by scenario and umbDisp:

# summing weights by umbDisp for each scenario  
tempWeightData2 <-   
 impactsInDetail %>%  
 filter(  
 LCstage == "endOfLife" & impactCategory == "Energy demand"  
 ) %>% # correct set for weight calculations  
 group\_by(scenario, umbDisp) %>%   
 summarise(tons=sum(tons)) %>%  
 ungroup() %>%  
 filter(tons != 0) %>%  
 mutate(scenario= factor(scenario, levels=rev(scenarioOrder))) %>%  
 group\_by(scenario) %>%  
 mutate(recovTons=ifelse(umbDisp=="recovery",tons,0)) %>%  
 ungroup()

## `summarise()` has grouped output by 'scenario'. You can override using the `.groups` argument.

kable(tempWeightData2)

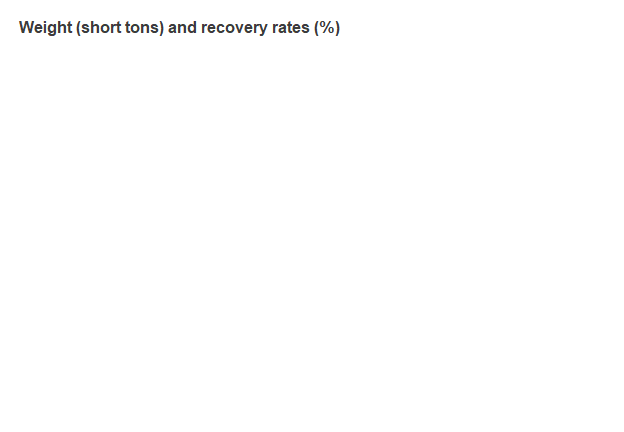
|  |  |  |  |
| --- | --- | --- | --- |
| scenario | umbDisp | tons | recovTons |

# creating a data file that expresses weight-based  
# recovery rate for each scenario  
tempWeightData2a <-  
 tempWeightData2 %>%  
 group\_by(scenario) %>%  
 summarise(  
 recovTons=sum(recovTons),  
 tons=sum(tons)  
 ) %>%  
 ungroup() %>%  
 mutate(recovRate=recovTons/tons)  
kable(tempWeightData2a)

|  |  |  |  |
| --- | --- | --- | --- |
| scenario | recovTons | tons | recovRate |

making that weight data into a chart…

tempWeightChart2 <-  
 ggplot()+  
 ggtitle("Weight (short tons) and recovery rates (%)")+  
 theme\_539()+  
 geom\_bar(  
 data = tempWeightData2,  
 aes(x = scenario, y= tons, fill= umbDisp),  
 color=NA,  
 stat="identity"  
 )+  
 geom\_text(  
 data=tempWeightData2a,  
 aes(x=scenario, y=tons, label=percent(recovRate)),  
 hjust=1.1,  
 color="white"  
 )+  
 scale\_fill\_viridis(  
 begin=0.32, end=0.8, option="D", discrete = TRUE  
 )+  
 coord\_flip()+  
 guides(fill=guide\_legend(ncol=2, title.position = "top"))+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 plot.title = element\_text(size=12),  
 legend.position="bottom",  
 legend.title = element\_text(size=8),  
 legend.justification="left"  
 )  
tempWeightChart2



ggsave("chart\_output/weights2.png")

## Saving 6.5 x 4.5 in image

ggsave("chart\_output/weights2.svg")

## Saving 6.5 x 4.5 in image

##### Illustrating impacts by life cycle stage

Previously the impacts associated with each scenario were given only as net values – the sum of three life cycle stages. It can be interesting to show how those three stages contribute to the net. To do that, we sum impacts by scenario and LCstage:

tempImpactData2 <-  
 impactsInDetail %>%  
 group\_by(scenario, LCstage, impactCategory, impactUnits) %>%  
 summarise(impact=sum(impact)) %>%  
 ungroup() %>%  
 mutate(  
 scenario = factor(scenario, levels = rev(scenarioOrder)),  
 impactLabel =   
 paste(  
 impactCategory,  
 " (",  
 impactUnits,  
 ")",  
 sep=""  
 ),  
 LCstage=  
 factor(  
 LCstage,  
 levels=c("production","endOfLife","endOfLifeTransport")  
 )  
 )

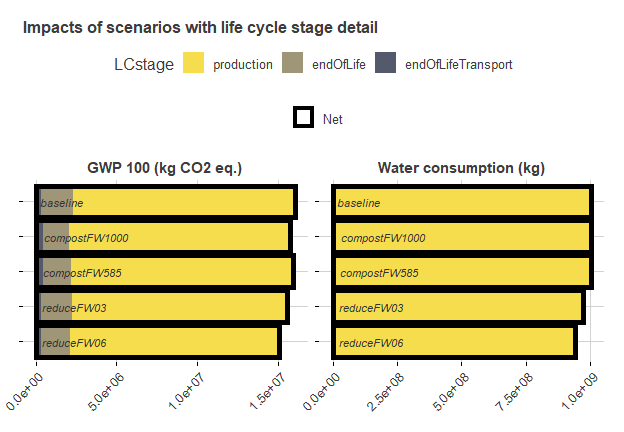
## `summarise()` has grouped output by 'scenario', 'LCstage', 'impactCategory'. You can override using the `.groups` argument.

kable(tempImpactData2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| scenario | LCstage | impactCategory | impactUnits | impact | impactLabel |
| baseline | endOfLife | GWP 100 | kg CO2 eq. | 2002172.33 | GWP 100 (kg CO2 eq.) |
| baseline | endOfLife | Water consumption | kg | -496424.15 | Water consumption (kg) |
| baseline | endOfLifeTransport | GWP 100 | kg CO2 eq. | 321332.89 | GWP 100 (kg CO2 eq.) |
| baseline | endOfLifeTransport | Water consumption | kg | 848544.17 | Water consumption (kg) |
| baseline | production | GWP 100 | kg CO2 eq. | 13694860.74 | GWP 100 (kg CO2 eq.) |
| baseline | production | Water consumption | kg | 1003703883.22 | Water consumption (kg) |
| compostFW1000 | endOfLife | GWP 100 | kg CO2 eq. | 1588107.69 | GWP 100 (kg CO2 eq.) |
| compostFW1000 | endOfLife | Water consumption | kg | -87158.42 | Water consumption (kg) |
| compostFW1000 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 448263.39 | GWP 100 (kg CO2 eq.) |
| compostFW1000 | endOfLifeTransport | Water consumption | kg | 1183729.72 | Water consumption (kg) |
| compostFW1000 | production | GWP 100 | kg CO2 eq. | 13694860.74 | GWP 100 (kg CO2 eq.) |
| compostFW1000 | production | Water consumption | kg | 1003703883.22 | Water consumption (kg) |
| compostFW585 | endOfLife | GWP 100 | kg CO2 eq. | 1759944.52 | GWP 100 (kg CO2 eq.) |
| compostFW585 | endOfLife | Water consumption | kg | -257003.70 | Water consumption (kg) |
| compostFW585 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 457720.51 | GWP 100 (kg CO2 eq.) |
| compostFW585 | endOfLifeTransport | Water consumption | kg | 1208703.14 | Water consumption (kg) |
| compostFW585 | production | GWP 100 | kg CO2 eq. | 13694860.74 | GWP 100 (kg CO2 eq.) |
| compostFW585 | production | Water consumption | kg | 1003703883.22 | Water consumption (kg) |
| reduceFW03 | endOfLife | GWP 100 | kg CO2 eq. | 1923126.59 | GWP 100 (kg CO2 eq.) |
| reduceFW03 | endOfLife | Water consumption | kg | -438750.77 | Water consumption (kg) |
| reduceFW03 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 312003.38 | GWP 100 (kg CO2 eq.) |
| reduceFW03 | endOfLifeTransport | Water consumption | kg | 823907.73 | Water consumption (kg) |
| reduceFW03 | production | GWP 100 | kg CO2 eq. | 13287968.71 | GWP 100 (kg CO2 eq.) |
| reduceFW03 | production | Water consumption | kg | 973307751.77 | Water consumption (kg) |
| reduceFW06 | endOfLife | GWP 100 | kg CO2 eq. | 1844080.85 | GWP 100 (kg CO2 eq.) |
| reduceFW06 | endOfLife | Water consumption | kg | -381077.38 | Water consumption (kg) |
| reduceFW06 | endOfLifeTransport | GWP 100 | kg CO2 eq. | 302673.88 | GWP 100 (kg CO2 eq.) |
| reduceFW06 | endOfLifeTransport | Water consumption | kg | 799271.29 | Water consumption (kg) |
| reduceFW06 | production | GWP 100 | kg CO2 eq. | 12881076.68 | GWP 100 (kg CO2 eq.) |
| reduceFW06 | production | Water consumption | kg | 942911620.32 | Water consumption (kg) |

Now make that into a chart, with life cycle stage impacts in colors, and the (previously calculated) net impact as a black outline.

tempImpactChart2 <-  
 ggplot()+  
 ggtitle("Impacts of scenarios with life cycle stage detail")+  
 theme\_539()+  
 geom\_bar(  
 data = tempImpactData2,  
 aes(x = scenario, y= impact, fill=LCstage),  
 color=NA,  
 # size=2,  
 stat="identity"  
 )+  
 geom\_bar(  
 data=tempImpactData1 %>% mutate(LCstage="Net"),  
 aes(x=scenario, y=impact, color=LCstage),  
 stat="identity",  
 fill=NA,  
 size=2  
 )+  
 geom\_text(  
 data=tempImpactData1 %>% mutate(LCstage="Net"),  
 aes(x=scenario, y=0, label=scenario),  
 stat="identity",  
 color="gray20",  
 size=3,  
 fontface="italic",  
 hjust=-0.1,  
 vjust=0.5  
 )+  
 facet\_wrap(~impactLabel, ncol=2, scales="free")+  
 scale\_color\_manual(values="black")+  
 scale\_fill\_viridis(  
 begin=0.32, end=0.95, option="E", discrete = TRUE,  
 direction = -1  
 )+  
 coord\_flip()+  
 guides(  
 color=guide\_legend(nrow=1, title=NULL),  
 fill=guide\_legend(nrow=1, title.position = "left")  
 )+  
 theme(  
 rect=element\_rect(fill="transparent"),  
 plot.title = element\_text(size=12),  
 legend.position="top",  
 axis.text.x=element\_text(angle=45, hjust=1),  
 axis.text.y=element\_blank(),  
 strip.text = element\_text(size=11, face="bold")  
 )  
tempImpactChart2



ggsave("chart\_output/impacts3.png")

## Saving 6.5 x 4.5 in image

ggsave("chart\_output/impacts3.svg")

## Saving 6.5 x 4.5 in image

The chart above shows that, at least in this analysis, production impacts (the tan color) make by far the biggest contribution to the net impact (black outline). End-of-life credits are visible, in the “recover\_nearly\_all” scenario, but they are smaller than many might guess. End-of-life transport impacts are so small they are not visible, probably obscured by the black “net” line.

##### Heatmap

Another way to compare impact results across scenarios is the “heatmap”, where all impacts are scaled to the baseline value.

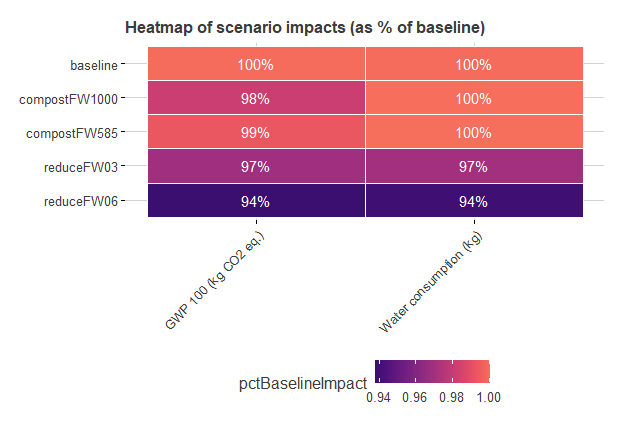
Here is the code to produce a data table which can be drawn as a heatmap:

tempImpactData3 <-  
 tempImpactData1 %>%   
 filter(scenario=="baseline") %>%  
 select(impactLabel, impact) %>%  
 rename(baselineImpact=impact)  
tempImpactData3a <-  
 left\_join(  
 tempImpactData1,  
 tempImpactData3,  
 by= c("impactLabel")  
 ) %>%  
 mutate(  
 pctBaselineImpact=impact/baselineImpact  
 )  
kable(tempImpactData3a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| scenario | impactCategory | impactUnits | impact | impactLabel | baselineImpact | pctBaselineImpact |
| baseline | GWP 100 | kg CO2 eq. | 16018366 | GWP 100 (kg CO2 eq.) | 16018366 | 1.0000000 |
| baseline | Water consumption | kg | 1004056003 | Water consumption (kg) | 1004056003 | 1.0000000 |
| compostFW1000 | GWP 100 | kg CO2 eq. | 15731232 | GWP 100 (kg CO2 eq.) | 16018366 | 0.9820747 |
| compostFW1000 | Water consumption | kg | 1004800455 | Water consumption (kg) | 1004056003 | 1.0007414 |
| compostFW585 | GWP 100 | kg CO2 eq. | 15912526 | GWP 100 (kg CO2 eq.) | 16018366 | 0.9933926 |
| compostFW585 | Water consumption | kg | 1004655583 | Water consumption (kg) | 1004056003 | 1.0005972 |
| reduceFW03 | GWP 100 | kg CO2 eq. | 15523099 | GWP 100 (kg CO2 eq.) | 16018366 | 0.9690813 |
| reduceFW03 | Water consumption | kg | 973692909 | Water consumption (kg) | 1004056003 | 0.9697596 |
| reduceFW06 | GWP 100 | kg CO2 eq. | 15027831 | GWP 100 (kg CO2 eq.) | 16018366 | 0.9381626 |
| reduceFW06 | Water consumption | kg | 943329814 | Water consumption (kg) | 1004056003 | 0.9395191 |

now, to make that into a chart:

tempImpactChart3 <-  
 ggplot()+  
 ggtitle("Heatmap of scenario impacts (as % of baseline)")+  
 theme\_539()+  
 geom\_tile(  
 data=tempImpactData3a,  
 aes(y=scenario, x=impactLabel, fill=pctBaselineImpact),  
 color="white"  
 )+  
 geom\_text(  
 data=tempImpactData3a,  
 aes(  
 y=scenario, x=impactLabel, label=percent(pctBaselineImpact,1)  
 ),  
 color="white"  
 )+  
 scale\_fill\_viridis(begin=0.2, end=0.7, option="A")+  
 theme(  
 plot.title = element\_text(size=12),  
 rect=element\_rect(fill="transparent"),  
 panel.grid = element\_blank(),  
 axis.ticks = element\_line(),  
 axis.text.x = element\_text(hjust=1, angle=45)  
 )  
tempImpactChart3



ggsave("chart\_output/impacts4.png")

## Saving 6.5 x 4.5 in image

ggsave("chart\_output/impacts4.svg")

## Saving 6.5 x 4.5 in image

In this display, the darker colors indicate for the eliminate\_food\_waste scenario show a dramatic effect on water consumption, and a notable effect on energy demand. Shades of orange for the recover\_nearly\_all scenario show much more modest effects, compared to the baseline scenario.

Though this heatmap is very simple – just 6 numbers – this kind of display can be useful for identifying “hotspots” and gradients within grids of dozen or hundreds of numbers.

##### Weight vs impacts within a single scenario