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.

Less technical users may want to spend some time with the interactive web app version of WIC, which does not require programming. This app should be published online in 2021 – check the Waste Impact Calculator github page for status. The app may be useful for more technical users as well, as it illustrates a wide variety of the types of output the WIC framework can create.

### 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

### Outline

This document compares the life cycle environmental impacts associated with 3 management scenarios for a fictional municipal wastestream. To keep printouts short and comprehensible, there are only 2 materials:

* FoodWaste
* Electronics

The management scenarios are:

* “baseline” (representing the “actual, observed” recent mix of recycling and disposal);
* “eliminate\_food\_waste” (food waste is assumed to be zero); and
* “recover\_nearly\_all” (maximize recycling and composting)

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 assorted tabular and graphic results based on impactsInDetail

### 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 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.0  
## v tidyr 1.1.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

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

library(ggthemes) # some themes for plotting  
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  
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.

Here we load the massProfiles table into 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 = "source\_data/mass\_profiles.csv",   
 header = TRUE,   
 stringsAsFactors = FALSE   
 )  
# a formatted printout  
kable(  
 massProfiles,   
 caption="a simple massProfiles table for 3 scenarios and 2 materials"  
)

a simple massProfiles table for 3 scenarios and 2 materials

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| wasteshed | material | disposition | umbDisp | scenario | tons | miles |
| City A | Electronics | landfilling | disposal | baseline | 2.827717e+03 | 180 |
| City A | Electronics | recycling | recovery | baseline | 2.367066e+03 | 180 |
| City A | FoodWaste | anaerobicDigestion | recovery | baseline | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | disposal | baseline | 1.861940e-02 | 180 |
| City A | FoodWaste | combustion | recovery | baseline | 1.451891e+03 | 180 |
| City A | FoodWaste | composting | recovery | baseline | 1.567926e+03 | 180 |
| City A | FoodWaste | landfilling | disposal | baseline | 4.630180e+04 | 180 |
| City A | Electronics | landfilling | disposal | eliminate\_food\_waste | 2.827717e+03 | 180 |
| City A | Electronics | recycling | recovery | eliminate\_food\_waste | 2.367066e+03 | 180 |
| City A | FoodWaste | anaerobicDigestion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | composting | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | landfilling | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | Electronics | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 |
| City A | Electronics | recycling | recovery | recover\_nearly\_all | 5.194783e+03 | 180 |
| City A | FoodWaste | anaerobicDigestion | recovery | recover\_nearly\_all | 4.932163e+04 | 180 |
| City A | FoodWaste | combustion | disposal | recover\_nearly\_all | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | recovery | recover\_nearly\_all | 0.000000e+00 | 180 |
| City A | FoodWaste | composting | recovery | recover\_nearly\_all | 0.000000e+00 | 180 |
| City A | FoodWaste | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 |

Here we list the fields in massProfiles:

str(massProfiles)

## 'data.frame': 21 obs. of 7 variables:  
## $ wasteshed : chr "City A" "City A" "City A" "City A" ...  
## $ material : chr "Electronics" "Electronics" "FoodWaste" "FoodWaste" ...  
## $ disposition: chr "landfilling" "recycling" "anaerobicDigestion" "combustion" ...  
## $ umbDisp : chr "disposal" "recovery" "recovery" "disposal" ...  
## $ scenario : chr "baseline" "baseline" "baseline" "baseline" ...  
## $ tons : num 2.83e+03 2.37e+03 0.00 1.86e-02 1.45e+03 ...  
## $ miles : int 180 180 180 180 180 180 180 180 180 180 ...

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, what kind of management they represent, what life cycle stage they represent, etc.

If desired, we can summarise weight-based waste statistics for each scenario from massProfiles. For example, here is the summed number of tons of waste within each scenario:

massProfiles %>%  
 group\_by(scenario) %>%  
 summarize(tons=sum(tons)) %>%  
 kable()

## `summarise()` ungrouping output (override with `.groups` argument)

|  |  |
| --- | --- |
| scenario | tons |
| baseline | 54516.417 |
| eliminate\_food\_waste | 5194.783 |
| recover\_nearly\_all | 54516.417 |

Here is a similar table with a bit more detail about what happens to all that waste at end of life:

massProfiles %>%  
 group\_by(scenario, material, disposition) %>%  
 summarize(tons=sum(tons)) %>%  
 kable()

## `summarise()` regrouping output by 'scenario', 'material' (override with `.groups` argument)

|  |  |  |  |
| --- | --- | --- | --- |
| scenario | material | disposition | tons |
| baseline | Electronics | landfilling | 2827.717 |
| baseline | Electronics | recycling | 2367.066 |
| baseline | FoodWaste | anaerobicDigestion | 0.000 |
| baseline | FoodWaste | combustion | 1451.910 |
| baseline | FoodWaste | composting | 1567.926 |
| baseline | FoodWaste | landfilling | 46301.798 |
| eliminate\_food\_waste | Electronics | landfilling | 2827.717 |
| eliminate\_food\_waste | Electronics | recycling | 2367.066 |
| eliminate\_food\_waste | FoodWaste | anaerobicDigestion | 0.000 |
| eliminate\_food\_waste | FoodWaste | combustion | 0.000 |
| eliminate\_food\_waste | FoodWaste | composting | 0.000 |
| eliminate\_food\_waste | FoodWaste | landfilling | 0.000 |
| recover\_nearly\_all | Electronics | landfilling | 0.000 |
| recover\_nearly\_all | Electronics | recycling | 5194.783 |
| recover\_nearly\_all | FoodWaste | anaerobicDigestion | 49321.634 |
| recover\_nearly\_all | FoodWaste | combustion | 0.000 |
| recover\_nearly\_all | FoodWaste | composting | 0.000 |
| recover\_nearly\_all | FoodWaste | landfilling | 0.000 |

This example illustrates an important quality of the Waste Impact Calculator. As you recall from the *Technical Overview*, the tons, dispositions, and miles listed in massProfiles *are the ONLY data* that distinguish solid waste management scenarios from one another. All of the user’s waste management ideas must be expressed in the massProfiles data frame! For example, a waste management scenario that increases recycling compared to baseline should have more tons associated with recycling dispositions, and less tons associated with disposal dispositions.

Here is an example of how massProfile information expresses the difference between solid waste management scenarios, “baseline” and “eliminate\_food\_waste”:

massProfiles %>%  
 filter(  
 scenario=="baseline" | scenario=="eliminate\_food\_waste"  
 ) %>%  
 kable()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| wasteshed | material | disposition | umbDisp | scenario | tons | miles |
| City A | Electronics | landfilling | disposal | baseline | 2.827717e+03 | 180 |
| City A | Electronics | recycling | recovery | baseline | 2.367066e+03 | 180 |
| City A | FoodWaste | anaerobicDigestion | recovery | baseline | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | disposal | baseline | 1.861940e-02 | 180 |
| City A | FoodWaste | combustion | recovery | baseline | 1.451891e+03 | 180 |
| City A | FoodWaste | composting | recovery | baseline | 1.567926e+03 | 180 |
| City A | FoodWaste | landfilling | disposal | baseline | 4.630180e+04 | 180 |
| City A | Electronics | landfilling | disposal | eliminate\_food\_waste | 2.827717e+03 | 180 |
| City A | Electronics | recycling | recovery | eliminate\_food\_waste | 2.367066e+03 | 180 |
| City A | FoodWaste | anaerobicDigestion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | combustion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | composting | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 |
| City A | FoodWaste | landfilling | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 |

The “eliminate\_food\_waste” scenario has far fewer total tons than the other scenarios, because (in this scenario) all food waste has been eliminated– it is neither landfilled nor composted nor combusted. (This is an impressive civic accomplishment! A more realistic goal might have been reducing food waste by 40%, but this example uses the extreme case.)

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 impactFactors table into an R data frame and print it out in a formatted table. The impactFactors table is typically thousands of records long, but for brevity in this example analysis, we filter it to only 2 materials and 2 impact categories.

# loading the impact factor data into an R data frame  
impactFactors <-  
 read.csv(  
 file = "../impactFactors/distributable/impactFactors.csv",   
 header = TRUE,   
 stringsAsFactors = FALSE #  
 ) %>%  
 # for the sake of brevity in this example analysis ,  
 # limiting the impactFactors to two impact categories  
 # and two materials. In regular usage there is no  
 # need to do such filtering -- the impactFactors   
 # data frame may be left complete.  
 filter(  
 (impactCategory== "Energy demand" |   
 impactCategory=="Water consumption") &  
 (material=="Electronics" | material=="FoodWaste")   
 ) %>%  
 # sorting it for easier reading  
 arrange(impactCategory, material, LCstage, disposition)  
# a formatted printout  
kable(impactFactors)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| material | LCstage | disposition | corporateSource | impactCategory | impactUnits | impliedMiles | impactFactor | gabiExportDate | wicImportDate |
| Electronics | endOfLife | landfilling | NA | Energy demand | MJ | 180 | 832.8763 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLife | recycling | NA | Energy demand | MJ | 180 | -13573.2370 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLifeTransport | landfilling | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLifeTransport | recycling | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 |
| Electronics | production | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | anaerobicDigestion | NA | Energy demand | MJ | 180 | -5592.4907 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | combustion | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | combustionNoER | NA | Energy demand | MJ | 180 | 934.0692 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | composting | NA | Energy demand | MJ | 180 | -543.1128 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | landfilling | NA | Energy demand | MJ | 180 | 319.6540 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | anaerobicDigestion | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | combustion | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | combustionNoER | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | composting | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | landfilling | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 |
| FoodWaste | production | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLife | landfilling | NA | Water consumption | kg | 180 | 216.3978 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLife | recycling | NA | Water consumption | kg | 180 | -9763.3786 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLifeTransport | landfilling | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 |
| Electronics | endOfLifeTransport | recycling | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 |
| Electronics | production | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | anaerobicDigestion | NA | Water consumption | kg | 180 | 1230.1352 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | combustion | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | combustionNoER | NA | Water consumption | kg | 180 | 2014.0990 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | composting | NA | Water consumption | kg | 180 | 158.5119 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLife | landfilling | NA | Water consumption | kg | 180 | -250.7538 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | anaerobicDigestion | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | combustion | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | combustionNoER | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | composting | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 |
| FoodWaste | endOfLifeTransport | landfilling | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 |
| FoodWaste | production | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 |

The fields making up impactFactors can be viewed as:

str(impactFactors)

## 'data.frame': 32 obs. of 10 variables:  
## $ material : chr "Electronics" "Electronics" "Electronics" "Electronics" ...  
## $ LCstage : chr "endOfLife" "endOfLife" "endOfLifeTransport" "endOfLifeTransport" ...  
## $ disposition : chr "landfilling" "recycling" "landfilling" "recycling" ...  
## $ corporateSource: chr NA NA NA NA ...  
## $ impactCategory : chr "Energy demand" "Energy demand" "Energy demand" "Energy demand" ...  
## $ impactUnits : chr "MJ" "MJ" "MJ" "MJ" ...  
## $ impliedMiles : int 180 180 180 180 180 180 180 180 180 180 ...  
## $ impactFactor : num 833 -13573 603 603 468130 ...  
## $ gabiExportDate : chr "2020-11-06" "2020-11-06" "2020-11-06" "2020-11-06" ...  
## $ wicImportDate : chr "2020-12-24" "2020-12-24" "2020-12-24" "2020-12-24" ...

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()` regrouping output by 'material', 'LCstage', 'disposition' (override with `.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 114227.616132393."

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

## [1] "Total tonnage in massProfilesPlus is 228455.232264786."

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

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 3 x 3  
## scenario prodTons eolTons  
## <chr> <dbl> <dbl>  
## 1 baseline 54516. 54516.  
## 2 eliminate\_food\_waste 5195. 5195.  
## 3 recover\_nearly\_all 54516. 54516.

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. Since the resulting file has both tons (from the massProfiles table) and impactFactor scaled to tons (from the impactFactors table), these can 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  
)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| wasteshed | material | disposition | umbDisp | scenario | tons | miles | LCstage | corporateSource | impactCategory | impactUnits | impliedMiles | impactFactor | gabiExportDate | wicImportDate | impact |
| City A | Electronics | landfilling | disposal | baseline | 2.827717e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | 832.8763 | 2020-11-06 | 2020-12-24 | 2.355138e+06 |
| City A | Electronics | recycling | recovery | baseline | 2.367066e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -13573.2370 | 2020-11-06 | 2020-12-24 | -3.212875e+07 |
| City A | Electronics | landfilling | disposal | baseline | 2.827717e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 | 1.706078e+06 |
| City A | Electronics | recycling | recovery | baseline | 2.367066e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 | 1.428149e+06 |
| City A | Electronics | production | production | baseline | 2.827717e+03 | 180 | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 | 1.323740e+09 |
| City A | Electronics | production | production | baseline | 2.367066e+03 | 180 | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 | 1.108095e+09 |
| City A | FoodWaste | anaerobicDigestion | recovery | baseline | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -5592.4907 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | baseline | 1.861940e-02 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 | -1.068965e+01 |
| City A | FoodWaste | combustion | recovery | baseline | 1.451891e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 | -8.335508e+05 |
| City A | FoodWaste | composting | recovery | baseline | 1.567926e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -543.1128 | 2020-11-06 | 2020-12-24 | -8.515606e+05 |
| City A | FoodWaste | landfilling | disposal | baseline | 4.630180e+04 | 180 | endOfLife | NA | Energy demand | MJ | 180 | 319.6540 | 2020-11-06 | 2020-12-24 | 1.480056e+07 |
| City A | FoodWaste | anaerobicDigestion | recovery | baseline | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | baseline | 1.861940e-02 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 1.123625e+01 |
| City A | FoodWaste | combustion | recovery | baseline | 1.451891e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 8.761733e+05 |
| City A | FoodWaste | composting | recovery | baseline | 1.567926e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 9.461970e+05 |
| City A | FoodWaste | landfilling | disposal | baseline | 4.630180e+04 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 2.794177e+07 |
| City A | FoodWaste | production | production | baseline | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | baseline | 1.861940e-02 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 7.269730e+02 |
| City A | FoodWaste | production | production | baseline | 1.451891e+03 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 5.668743e+07 |
| City A | FoodWaste | production | production | baseline | 1.567926e+03 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 6.121789e+07 |
| City A | FoodWaste | production | production | baseline | 4.630180e+04 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 1.807801e+09 |
| City A | Electronics | landfilling | disposal | eliminate\_food\_waste | 2.827717e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | 832.8763 | 2020-11-06 | 2020-12-24 | 2.355138e+06 |
| City A | Electronics | recycling | recovery | eliminate\_food\_waste | 2.367066e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -13573.2370 | 2020-11-06 | 2020-12-24 | -3.212875e+07 |
| City A | Electronics | landfilling | disposal | eliminate\_food\_waste | 2.827717e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 | 1.706078e+06 |
| City A | Electronics | recycling | recovery | eliminate\_food\_waste | 2.367066e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 | 1.428149e+06 |
| City A | Electronics | production | production | eliminate\_food\_waste | 2.827717e+03 | 180 | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 | 1.323740e+09 |
| City A | Electronics | production | production | eliminate\_food\_waste | 2.367066e+03 | 180 | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 | 1.108095e+09 |
| City A | FoodWaste | anaerobicDigestion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -5592.4907 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -543.1128 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | 319.6540 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | anaerobicDigestion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | 832.8763 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | recycling | recovery | recover\_nearly\_all | 5.194783e+03 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -13573.2370 | 2020-11-06 | 2020-12-24 | -7.051002e+07 |
| City A | Electronics | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | recycling | recovery | recover\_nearly\_all | 5.194783e+03 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.3414 | 2020-11-06 | 2020-12-24 | 3.134227e+06 |
| City A | Electronics | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | production | production | recover\_nearly\_all | 5.194783e+03 | 180 | production | NA | Energy demand | MJ | 180 | 468130.3713 | 2020-11-06 | 2020-12-24 | 2.431836e+09 |
| City A | FoodWaste | anaerobicDigestion | recovery | recover\_nearly\_all | 4.932163e+04 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -5592.4907 | 2020-11-06 | 2020-12-24 | -2.758308e+08 |
| City A | FoodWaste | combustion | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -574.1139 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | -543.1128 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Energy demand | MJ | 180 | 319.6540 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | anaerobicDigestion | recovery | recover\_nearly\_all | 4.932163e+04 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 2.976415e+07 |
| City A | FoodWaste | combustion | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Energy demand | MJ | 180 | 603.4704 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 4.932163e+04 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 1.925707e+09 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Energy demand | MJ | 180 | 39043.8618 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | landfilling | disposal | baseline | 2.827717e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | 216.3978 | 2020-11-06 | 2020-12-24 | 6.119116e+05 |
| City A | Electronics | recycling | recovery | baseline | 2.367066e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | -9763.3786 | 2020-11-06 | 2020-12-24 | -2.311056e+07 |
| City A | Electronics | landfilling | disposal | baseline | 2.827717e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 | 3.062285e+05 |
| City A | Electronics | recycling | recovery | baseline | 2.367066e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 | 2.563422e+05 |
| City A | Electronics | production | production | baseline | 2.827717e+03 | 180 | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 | 7.675614e+08 |
| City A | Electronics | production | production | baseline | 2.367066e+03 | 180 | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 | 6.425214e+08 |
| City A | FoodWaste | anaerobicDigestion | recovery | baseline | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | 1230.1352 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | baseline | 1.861940e-02 | 180 | endOfLife | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 | -3.259627e+00 |
| City A | FoodWaste | combustion | recovery | baseline | 1.451891e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 | -2.541771e+05 |
| City A | FoodWaste | composting | recovery | baseline | 1.567926e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | 158.5119 | 2020-11-06 | 2020-12-24 | 2.485349e+05 |
| City A | FoodWaste | landfilling | disposal | baseline | 4.630180e+04 | 180 | endOfLife | NA | Water consumption | kg | 180 | -250.7538 | 2020-11-06 | 2020-12-24 | -1.161035e+07 |
| City A | FoodWaste | anaerobicDigestion | recovery | baseline | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | baseline | 1.861940e-02 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 2.016824e+00 |
| City A | FoodWaste | combustion | recovery | baseline | 1.451891e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 1.572666e+05 |
| City A | FoodWaste | composting | recovery | baseline | 1.567926e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 1.698354e+05 |
| City A | FoodWaste | landfilling | disposal | baseline | 4.630180e+04 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 5.015341e+06 |
| City A | FoodWaste | production | production | baseline | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | baseline | 1.861940e-02 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 2.460685e+03 |
| City A | FoodWaste | production | production | baseline | 1.451891e+03 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 1.918777e+08 |
| City A | FoodWaste | production | production | baseline | 1.567926e+03 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 2.072125e+08 |
| City A | FoodWaste | production | production | baseline | 4.630180e+04 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 6.119111e+09 |
| City A | Electronics | landfilling | disposal | eliminate\_food\_waste | 2.827717e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | 216.3978 | 2020-11-06 | 2020-12-24 | 6.119116e+05 |
| City A | Electronics | recycling | recovery | eliminate\_food\_waste | 2.367066e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | -9763.3786 | 2020-11-06 | 2020-12-24 | -2.311056e+07 |
| City A | Electronics | landfilling | disposal | eliminate\_food\_waste | 2.827717e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 | 3.062285e+05 |
| City A | Electronics | recycling | recovery | eliminate\_food\_waste | 2.367066e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 | 2.563422e+05 |
| City A | Electronics | production | production | eliminate\_food\_waste | 2.827717e+03 | 180 | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 | 7.675614e+08 |
| City A | Electronics | production | production | eliminate\_food\_waste | 2.367066e+03 | 180 | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 | 6.425214e+08 |
| City A | FoodWaste | anaerobicDigestion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | 1230.1352 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | 158.5119 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | -250.7538 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | anaerobicDigestion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | eliminate\_food\_waste | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | eliminate\_food\_waste | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | 216.3978 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | recycling | recovery | recover\_nearly\_all | 5.194783e+03 | 180 | endOfLife | NA | Water consumption | kg | 180 | -9763.3786 | 2020-11-06 | 2020-12-24 | -5.071863e+07 |
| City A | Electronics | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | recycling | recovery | recover\_nearly\_all | 5.194783e+03 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.2953 | 2020-11-06 | 2020-12-24 | 5.625707e+05 |
| City A | Electronics | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | Electronics | production | production | recover\_nearly\_all | 5.194783e+03 | 180 | production | NA | Water consumption | kg | 180 | 271442.1168 | 2020-11-06 | 2020-12-24 | 1.410083e+09 |
| City A | FoodWaste | anaerobicDigestion | recovery | recover\_nearly\_all | 4.932163e+04 | 180 | endOfLife | NA | Water consumption | kg | 180 | 1230.1352 | 2020-11-06 | 2020-12-24 | 6.067228e+07 |
| City A | FoodWaste | combustion | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | -175.0662 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | 158.5119 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLife | NA | Water consumption | kg | 180 | -250.7538 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | anaerobicDigestion | recovery | recover\_nearly\_all | 4.932163e+04 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 5.342445e+06 |
| City A | FoodWaste | combustion | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | combustion | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | composting | recovery | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | landfilling | disposal | recover\_nearly\_all | 0.000000e+00 | 180 | endOfLifeTransport | NA | Water consumption | kg | 180 | 108.3185 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 4.932163e+04 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 6.518204e+09 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |
| City A | FoodWaste | production | production | recover\_nearly\_all | 0.000000e+00 | 180 | production | NA | Water consumption | kg | 180 | 132157.0933 | 2020-11-06 | 2020-12-24 | 0.000000e+00 |

Some things to note about the impactsInDetails table:

* Each line is labeled with the *umbDisp* from massProfiles, so distinctions can be made between recovery and disposal impacts or tonnages if desired.
* There are 3 LCstages (production, end-of-life transport, and end-of-life treatment) represented in impactsInDetail. The properties of the left\_join merge have created the records needed to represent the (hitherto missing) end-of-life transport stage. Since in every impact category, each material & end-of-life disposition has *two* records in impactFactors (one for the LCstage endOfLife and another for endOfLifeTransport), a left\_join has added tonnages related to end-of-life transport.

### Checking the internal consistency of *impactsInDetail*

Before using the impactsInDetail file 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()` regrouping output by 'LCstage' (override with `.groups` argument)

|  |  |  |
| --- | --- | --- |
| LCstage | scenario | tons |
| endOfLife | baseline | 109032.83 |
| endOfLifeTransport | baseline | 109032.83 |
| production | baseline | 109032.83 |
| endOfLife | eliminate\_food\_waste | 10389.57 |
| endOfLifeTransport | eliminate\_food\_waste | 10389.57 |
| production | eliminate\_food\_waste | 10389.57 |
| endOfLife | recover\_nearly\_all | 109032.83 |
| endOfLifeTransport | recover\_nearly\_all | 109032.83 |
| production | recover\_nearly\_all | 109032.83 |

Note that tonnages in the table above are not identical to tonnages listed in 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)

## `summarise()` ungrouping output (override with `.groups` argument)

# 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=="Energy demand"  
 ) %>% # correct set for weight calculations  
 group\_by(scenario, disposition) %>%   
 summarise(tons=sum(tons)) %>%  
 ungroup() %>%  
 filter(tons != 0) %>%  
 mutate(scenario= factor(scenario, levels=rev(scenarioOrder)))

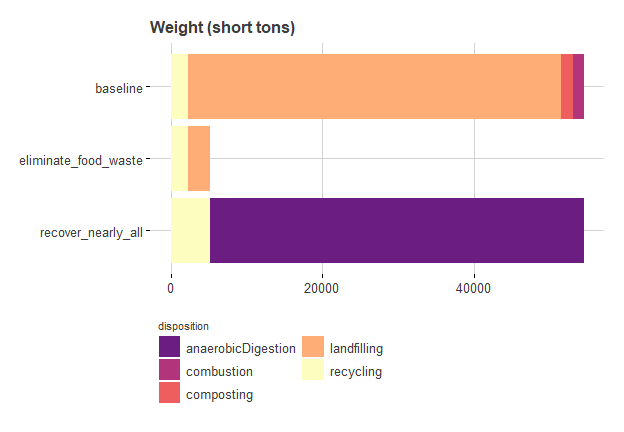
## `summarise()` regrouping output by 'scenario' (override with `.groups` argument)

kable(tempWeightData1)

|  |  |  |
| --- | --- | --- |
| scenario | disposition | tons |
| baseline | combustion | 1451.910 |
| baseline | composting | 1567.926 |
| baseline | landfilling | 49129.515 |
| baseline | recycling | 2367.066 |
| eliminate\_food\_waste | landfilling | 2827.717 |
| eliminate\_food\_waste | recycling | 2367.066 |
| recover\_nearly\_all | anaerobicDigestion | 49321.634 |
| recover\_nearly\_all | recycling | 5194.783 |

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=1, discrete = TRUE, option="A")+  
 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

##### 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 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=""  
 )  
 )

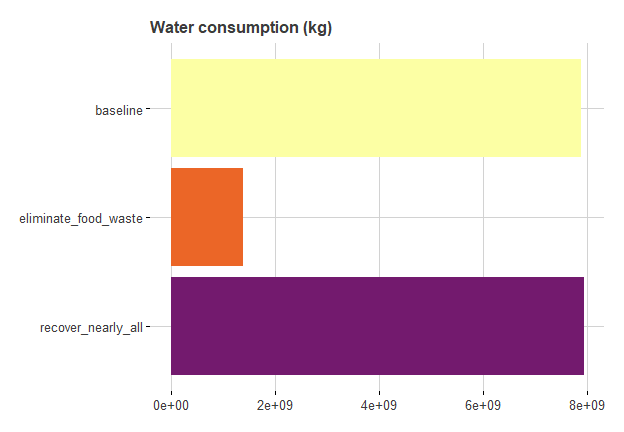
## `summarise()` regrouping output by 'scenario', 'impactCategory' (override with `.groups` argument)

kable(tempImpactData1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| scenario | impactCategory | impactUnits | impact | impactLabel |
| baseline | Energy demand | MJ | 4373782822 | Energy demand (MJ) |
| baseline | Water consumption | kg | 7900076966 | Water consumption (kg) |
| eliminate\_food\_waste | Energy demand | MJ | 2405196177 | Energy demand (MJ) |
| eliminate\_food\_waste | Water consumption | kg | 1388146736 | Water consumption (kg) |
| recover\_nearly\_all | Energy demand | MJ | 4044100203 | Energy demand (MJ) |
| recover\_nearly\_all | Water consumption | kg | 7944145264 | 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")

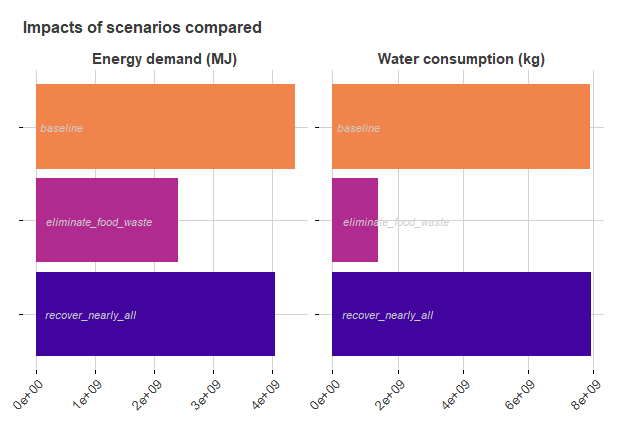
## Saving 6.5 x 4.5 in image

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:

* maximizing recovery (“recover\_nearly\_all”) decreases energy demand somewhat compared to baseline, but does not decrease water consumption.
* eliminating food waste dramatically reduces water consumption, but the decrease in energy demand is somewhat smaller.

##### 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()` regrouping output by 'scenario' (override with `.groups` argument)

kable(tempWeightData2)

|  |  |  |  |
| --- | --- | --- | --- |
| scenario | umbDisp | tons | recovTons |
| baseline | disposal | 49129.534 | 0.000 |
| baseline | recovery | 5386.883 | 5386.883 |
| eliminate\_food\_waste | disposal | 2827.717 | 0.000 |
| eliminate\_food\_waste | recovery | 2367.066 | 2367.066 |
| recover\_nearly\_all | recovery | 54516.417 | 54516.417 |

# 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)

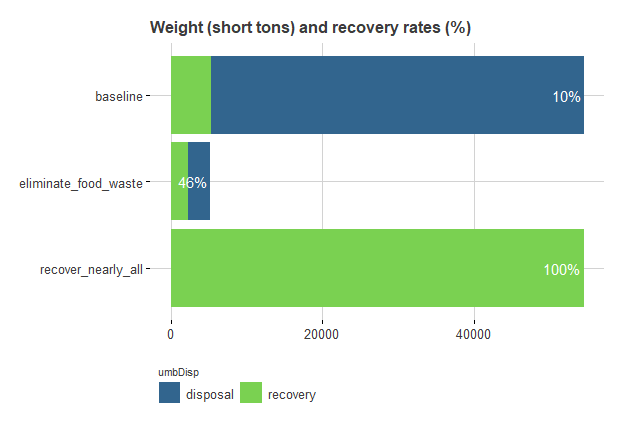
## `summarise()` ungrouping output (override with `.groups` argument)

kable(tempWeightData2a)

|  |  |  |  |
| --- | --- | --- | --- |
| scenario | recovTons | tons | recovRate |
| recover\_nearly\_all | 54516.417 | 54516.417 | 1.0000000 |
| eliminate\_food\_waste | 2367.066 | 5194.783 | 0.4556622 |
| baseline | 5386.883 | 54516.417 | 0.0988121 |

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")  
 )  
 )

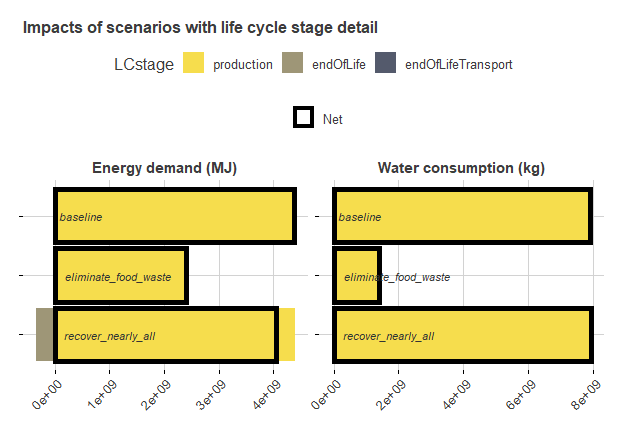
## `summarise()` regrouping output by 'scenario', 'LCstage', 'impactCategory' (override with `.groups` argument)

kable(tempImpactData2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| scenario | LCstage | impactCategory | impactUnits | impact | impactLabel |
| baseline | endOfLife | Energy demand | MJ | -16658176.6 | Energy demand (MJ) |
| baseline | endOfLife | Water consumption | kg | -34114649.0 | Water consumption (kg) |
| baseline | endOfLifeTransport | Energy demand | MJ | 32898375.9 | Energy demand (MJ) |
| baseline | endOfLifeTransport | Water consumption | kg | 5905015.5 | Water consumption (kg) |
| baseline | production | Energy demand | MJ | 4357542622.3 | Energy demand (MJ) |
| baseline | production | Water consumption | kg | 7928286599.9 | Water consumption (kg) |
| eliminate\_food\_waste | endOfLife | Energy demand | MJ | -29773609.7 | Energy demand (MJ) |
| eliminate\_food\_waste | endOfLife | Water consumption | kg | -22498650.0 | Water consumption (kg) |
| eliminate\_food\_waste | endOfLifeTransport | Energy demand | MJ | 3134227.3 | Energy demand (MJ) |
| eliminate\_food\_waste | endOfLifeTransport | Water consumption | kg | 562570.7 | Water consumption (kg) |
| eliminate\_food\_waste | production | Energy demand | MJ | 2431835559.1 | Energy demand (MJ) |
| eliminate\_food\_waste | production | Water consumption | kg | 1410082815.2 | Water consumption (kg) |
| recover\_nearly\_all | endOfLife | Energy demand | MJ | -346340795.2 | Energy demand (MJ) |
| recover\_nearly\_all | endOfLife | Water consumption | kg | 9953648.7 | Water consumption (kg) |
| recover\_nearly\_all | endOfLifeTransport | Energy demand | MJ | 32898375.9 | Energy demand (MJ) |
| recover\_nearly\_all | endOfLifeTransport | Water consumption | kg | 5905015.5 | Water consumption (kg) |
| recover\_nearly\_all | production | Energy demand | MJ | 4357542622.4 | Energy demand (MJ) |
| recover\_nearly\_all | production | Water consumption | kg | 7928286600.0 | 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.

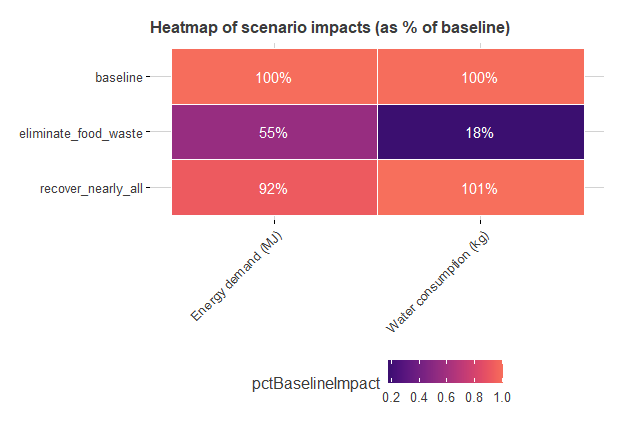
Here is the code to produce a data table which can be drawn as a heatmap:

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 | Energy demand | MJ | 4373782822 | Energy demand (MJ) | 4373782822 | 1.0000000 |
| baseline | Water consumption | kg | 7900076966 | Water consumption (kg) | 7900076966 | 1.0000000 |
| eliminate\_food\_waste | Energy demand | MJ | 2405196177 | Energy demand (MJ) | 4373782822 | 0.5499121 |
| eliminate\_food\_waste | Water consumption | kg | 1388146736 | Water consumption (kg) | 7900076966 | 0.1757131 |
| recover\_nearly\_all | Energy demand | MJ | 4044100203 | Energy demand (MJ) | 4373782822 | 0.9246230 |
| recover\_nearly\_all | Water consumption | kg | 7944145264 | Water consumption (kg) | 7900076966 | 1.0055782 |

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

Most of the results displayed above are high-level summaries. But it’s also helpful to look at a local waste stream in detail, to see how weight and impacts compare across individual materials. Here I set up a data table that can draw a comparison between weights and impacts of individual materials within a single scenario. Both waste and impacts are characterized as a percentage of the relevant waste-shed wide total.

Setting up such comparisons involves some more extended data processing.

tempWeightData4 <-  
 impactsInDetail %>%  
 filter(  
 LCstage == "endOfLife" & impactCategory=="Energy demand"  
 ) %>% # 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()` regrouping output by 'wasteshed', 'scenario', 'material' (override with `.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()` regrouping output by 'wasteshed', 'scenario' (override with `.groups` argument)

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

## `summarise()` regrouping output by 'wasteshed' (override with `.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) %>%  
 filter(impactCategory != "Energy demand")

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")  
 )  
 ) %>%  
 filter(impactCategory != "Energy demand") #removing to save space

## `summarise()` regrouping output by 'wasteshed', 'scenario', 'material', 'LCstage', 'impactCategory' (override with `.groups` argument)

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

## `summarise()` regrouping output by 'wasteshed', 'scenario', 'material', 'impactCategory', 'impactUnits' (override with `.groups` argument)

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

## `summarise()` regrouping output by 'wasteshed', 'scenario', 'impactCategory', 'impactUnits' (override with `.groups` argument)

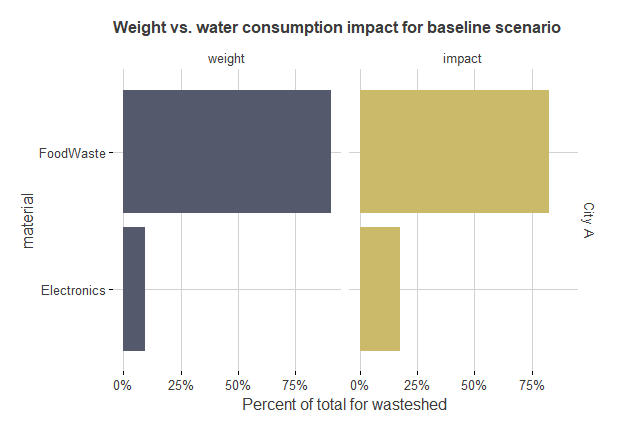
tempImpactData4c <-  
 full\_join(  
 tempImpactData4a,  
 tempImpactData4b,  
 by=c(  
 "wasteshed", "scenario", "impactCategory",  
 "impactUnits", "impactLabel"  
 )  
 ) %>%  
 mutate(pctImpact=impact/allImpact)

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 select only 1 wasteshed and 1 impact category.

weightImpactComparisonData2 <-  
 filter(  
 weightImpactComparisonData1,  
 scenario=="baseline",  
 impactLabel=="Water consumption (kg)"  
 ) %>%  
 mutate(dataType=factor(dataType, levels=c("weight", "impact")))  
ggplot()+  
 ggtitle("Weight vs. water consumption impact for baseline 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(wasteshed~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

This kind of chart often ends up exposing the way weight can be a poor proxy for life cycle impacts. Above, though electronics are a small portion of weight, they are a noticeably large portion of impacts.