

Pedalling Forward: The Evolution of Dedicated Cycling Infrastructure in Canadian Cities from 2010 to 2022

Calculating Road Lengths By Infrastructure Type

Richard Wen richard.wen@utoronto.ca

February 27, 2024

Contents

Libraries	1
Helper Function	2
Example Calculations with Vancouver Data	3
Data	3
Preprocessing	4
Step 1: Calculate Install Road Lengths By Infrastructure Type	5
Step 2a: Calculate Yearly Road Lengths By Infrastructure Type for Change 1	6
Step 2b: Calculate Yearly Road Lengths for Replaced Infrastructure Types from Change 1	7
Step 3a: Calculate Yearly Road Lengths By Infrastructure Type for Change 2	8
Step 3b: Calculate Yearly Road Lengths for Replaced Infrastructure Types from Change 2	8
Step 4: Calculate Final Yearly Road Lengths by Infrastructure Type	9
Final Function	10
Test Final Function	12

Libraries

```
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.2      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.2      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(glue)
library(readxl)
```

Helper Function

The following function calculates yearly road lengths by infrastructure type using cumulative sums and filling in missing years and types.

For a given infrastructure type, the total road length for a given year is expressed below:

$$length_{year,type} = f(year,type) = \sum_{i=year_{min}}^{year} l_{i,type} \mid l \geq 0$$

Where:

- $year$ is the given year
- $type$ is the infrastructure type
- $year_{min}$ is the earliest year available in the data
- $l_{i,type}$ is the road length l for previous years i and infrastructure j
- $l_{i,type}$ is set to 0 if there is no data

```
#' Calculate Yearly Road Lengths By Infrastructure Type
#
#' @param df A data.frame with three columns containing the year, type, and road lengths.
#' @param year_col The name (char) or index (int) of the column containing the years.
#' @param type_col The name (char) or index (int) of the column containing the infrastructure type
#' @param len_col The name (char) or index (int) of the column containing the road lengths.
#' @param out_col The name (char) of the column containing the calculated yearly road lengths by type.
#
#' @return A data.frame with three columns containing the year, type, and calculated yearly road length.
#' @export
#
calc_yearly_len <- function(
  df,
  year_col = 1,
  type_col = 2,
  len_col = 3,
  out_col = "len"
) {

  # Convert data types
  df[[year_col]] <- as.integer(df[[year_col]])
  df[[type_col]] <- as.character(df[[type_col]])
  df[[len_col]] <- as.numeric(df[[len_col]])

  # Remove rows with empty type
  out <- df %>% filter(
    !is.na(.data[[type_col]])
  )

  # Add dummy len for each type and year combo
  # Covers cases where type and year combo does not exist
  # E.g. No new PL installs in 2021, hence a record PL in 2021 does not exist
  type_uniq <- unique(out[[type_col]])
  type_n <- length(type_uniq)
  year_uniq <- min(out[[year_col]], na.rm = TRUE):max(out[[year_col]], na.rm = TRUE)
  year_n <- length(year_uniq)
  out <- out %>% add_row(
```

```

    !!year_col := rep(year_uniq, each = type_n),
    !!type_col := rep(type_uniq, year_n),
    !!len_col := rep(0, type_n * year_n)
  )

  # Calc cumsum for each non-empty type ordered by year
  out <- out %>%
    arrange(.data[[year_col]]) %>%
    group_by(.data[[type_col]]) %>%
    mutate(
      !!out_col := cumsum(segment_km)
    )

  # Get the last cumsum for each year and type
  out <- out %>%
    group_by(.data[[year_col]], .data[[type_col]]) %>%
    arrange(desc(row_number())) %>%
    slice(1)

  # Return only the columns spec
  out <- out %>% select(c(
    year_col,
    type_col,
    out_col
  ))
  return(out)
}

```

Example Calculations with Vancouver Data

Calculate road lengths per year adjusted for up to 2 changes, assuming data contains lengths in meters.

Data

Read road segment data along with install and upgrade variables.

```

vanc_raw <- read_excel("../data/raw/vancouver/Vancouver_Bikeways_Complete_AS.xlsm")
vanc_raw

```

```

## # A tibble: 3,666 x 90
##   OID_ object_id bike_rout0 street_na0 bikeway_t0 subtype status street_se0
##   <dbl>   <dbl> <chr>      <chr>      <chr>      <chr> <chr> <chr>
## 1  2812   295162 York      York Ave   Protected~ R      Active Residenti~
## 2  3616   297559 York      York Ave   Protected~ SL      Active Residenti~
## 3   738   296962 Mosaic     Woodland Dr~ Protected~ <NA>     Active Residenti~
## 4  2664   298377 King Edward W King Edwa~ Painted L~ PBP      Active Arterial
## 5  1799   296508 King Edward W King Edwa~ Painted L~ NB      Active Arterial
## 6  1798   296507 King Edward W King Edwa~ Painted L~ NB      Active Arterial
## 7  3251   296465 King Edward W King Edwa~ Painted L~ PBP      Active Arterial
## 8  1586   295842 King Edward W King Edwa~ Painted L~ PBP      Active Arterial
## 9  2432   298376 King Edward W King Edwa~ Painted L~ PBP      Active Arterial
## 10 1209   298375 King Edward W King Edwa~ Painted L~ PBP      Active Arterial
## # i 3,656 more rows
## # i 82 more variables: overall_d0 <chr>, bikeway_d0 <chr>, vehicle_d0 <chr>,

```

```
## # speed_lim0 <dbl>, surface_t0 <chr>, aaa_netwo0 <chr>, aaa_segme0 <chr>,
## # w_n_bound0 <chr>, e_s_bound0 <chr>, snow_remo0 <chr>, segment_l0 <chr>,
## # year_of_c0 <dbl>, construct0 <chr>, upgrade_y0 <dbl>, notes <chr>,
## # ID_DATAENTRY <chr>, ID_ROUTE <chr>, DPR_CHECK_FLAG <chr>, DPR_ENTRY <chr>,
## # DPR_EXCL_FLAG <chr>, DPR_EXCL1318_REASON <chr>, DPR_INCL_1318 <chr>, ...
```

Preprocessing

Clean the raw data.

```
# Preprocess
vanc <- vanc_raw %>%
  rename(
    install_year = "INST_YR",
    install_type = "INST_MIN_TYPE",
    upgrade1_year = "UPGR1_YR",
    upgrade1_type = "UPGR1_MIN_TYPE",
    upgrade2_year = "UPGR2_YR",
    upgrade2_type = "UPGR2_MIN_TYPE",
    segment_km = "segment_l0"
  ) %>%
  mutate( # data types
    install_year = as.integer(install_year),
    install_type = as.character(install_type),
    upgrade1_year = as.integer(upgrade1_year),
    upgrade1_type = as.character(upgrade1_type),
    upgrade2_year = as.integer(upgrade2_year),
    upgrade2_type = as.character(upgrade2_type),
    segment_km = as.numeric(segment_km)
  ) %>%
  mutate_at( # Set values to NA for types
    c("install_type", "upgrade1_type", "upgrade2_type"),
    ~replace(., . %in% c("None", "", "N"), NA)
  ) %>%
  mutate_at( # Set 0 or invalid years to NA
    c("install_year", "upgrade1_year", "upgrade2_year"),
    ~replace(., . <= 0, NA)
  ) %>%
  mutate( # convert lens from meters to km
    segment_km = segment_km / 1000
  ) %>%
  filter( # Remove invalid install year records
    !is.na(install_year)
  )

# Preview
vanc %>% select(
  install_year,
  install_type,
  upgrade1_year,
  upgrade1_type,
  upgrade2_year,
  upgrade2_type,
  segment_km
)
```

```
## # A tibble: 745 x 7
##   install_year install_type upgrade1_year upgrade1_type upgrade2_year
##   <int> <chr>          <int> <chr>          <int>
## 1      2014 PBL              NA <NA>             NA
## 2      2014 PBL              NA <NA>             NA
## 3      1999 <NA>          2021 PBL             NA
## 4      2015 PL              NA <NA>             NA
## 5      2015 PL              NA <NA>             NA
## 6      2015 PL              NA <NA>             NA
## 7      2015 PL              NA <NA>             NA
## 8      2015 PL              NA <NA>             NA
## 9      2015 PL              NA <NA>             NA
## 10     2015 PL              NA <NA>             NA
## # i 735 more rows
## # i 2 more variables: upgrade2_type <chr>, segment_km <dbl>
```

Step 1: Calculate Install Road Lengths By Infrastructure Type

Calculate cumulative sum of road lengths with installations by infrastructure type ordered by year.

$$length_{year,type}^{install}$$

```
vanc_install <- calc_yearly_len(
  vanc,
  year_col = "install_year",
  type_col = "install_type",
  len_col = "segment_km",
  out_col = "install_len"
) %>% rename(
  year = "install_year",
  type = "install_type"
)
```

```
## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(year_col)
##
##   # Now:
##   data %>% select(all_of(year_col))
##
## See <https://tidysselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(type_col)
##
##   # Now:
##   data %>% select(all_of(type_col))
```

```
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(out_col)
##
##   # Now:
##   data %>% select(all_of(out_col))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
vanc_install
```

```
## # A tibble: 144 x 3
## # Groups:   year, type [144]
##   year type  install_len
##   <int> <chr>      <dbl>
## 1  1986 BUF         0
## 2  1986 LSB      0.0474
## 3  1986 PBL         0
## 4  1986 PL         0
## 5  1987 BUF         0
## 6  1987 LSB      0.0474
## 7  1987 PBL         0
## 8  1987 PL         0
## 9  1988 BUF         0
## 10 1988 LSB      0.0474
## # i 134 more rows
```

Step 2a: Calculate Yearly Road Lengths By Infrastructure Type for Change 1

These road lengths are to be added for the particular years and infrastructure types.

$$length_{year,type}^{upgrade1}$$

```
vanc_upgrade1 <- calc_yearly_len(
  vanc %>% filter( # existing Change 1 with diff type upgrade
    !is.na(upgrade1_year) & install_type != upgrade1_type
  ),
  year_col = "upgrade1_year",
  type_col = "upgrade1_type",
  len_col = "segment_km",
  out_col = "upgrade1_len"
) %>% rename(
  year = "upgrade1_year",
  type = "upgrade1_type"
```

```
)
vanc_upgrade1
```

```
## # A tibble: 36 x 3
## # Groups:   year, type [36]
##   year type upgrade1_len
##   <int> <chr>         <dbl>
## 1  2010 BUF             0
## 2  2010 PBL           3.59
## 3  2010 PL            0.348
## 4  2011 BUF             0
## 5  2011 PBL           3.59
## 6  2011 PL            0.348
## 7  2012 BUF             0
## 8  2012 PBL           3.59
## 9  2012 PL            0.669
## 10 2013 BUF            1.04
## # i 26 more rows
```

Step 2b: Calculate Yearly Road Lengths for Replaced Infrastructure Types from Change 1

These road lengths are to be subtracted for the particular years and infrastructure types.

$$length_{year,type}^{replacement1}$$

```
vanc_upgrade1_repl <- calc_yearly_len(
  vanc %>% filter( # existing upgrade 1 with diff type upgrade
    !is.na(upgrade1_year) & install_type != upgrade1_type
  ),
  year_col = "upgrade1_year",
  type_col = "install_type",
  len_col = "segment_km",
  out_col = "upgrade1_repl_len"
) %>% rename(
  year = "upgrade1_year",
  type = "install_type"
)
vanc_upgrade1_repl
```

```
## # A tibble: 36 x 3
## # Groups:   year, type [36]
##   year type upgrade1_repl_len
##   <int> <chr>         <dbl>
## 1  2010 LSB             0
## 2  2010 PBL           0.348
## 3  2010 PL            3.59
## 4  2011 LSB             0
## 5  2011 PBL           0.348
## 6  2011 PL            3.59
## 7  2012 LSB           0.322
## 8  2012 PBL           0.348
## 9  2012 PL            3.59
```

```
## 10 2013 LSB          0.322
## # i 26 more rows
```

Step 3a: Calculate Yearly Road Lengths By Infrastructure Type for Change 2

These road lengths are to be added for the particular years and infrastructure types.

$$length_{year,type}^{upgrade2}$$

```
vanc_upgrade2 <- calc_yearly_len(
  vanc %>% filter( # existing upgrade 1 with diff type upgrade
    !is.na(upgrade2_year) & install_type != upgrade2_type
  ),
  year_col = "upgrade2_year",
  type_col = "upgrade2_type",
  len_col = "segment_km",
  out_col = "upgrade2_len"
) %>% rename(
  year = "upgrade2_year",
  type = "upgrade2_type"
)
vanc_upgrade2
```

```
## # A tibble: 9 x 3
## # Groups:   year, type [9]
##   year type upgrade2_len
##   <int> <chr>         <dbl>
## 1  2013 PBL           0.322
## 2  2014 PBL           0.322
## 3  2015 PBL           0.322
## 4  2016 PBL           0.322
## 5  2017 PBL           0.322
## 6  2018 PBL           0.322
## 7  2019 PBL           0.322
## 8  2020 PBL           0.421
## 9  2021 PBL           0.470
```

Step 3b: Calculate Yearly Road Lengths for Replaced Infrastructure Types from Change 2

These road lengths are to be subtracted for the particular years and infrastructure types.

$$length_{year,type}^{replacement2}$$

```
vanc_upgrade2_repl <- calc_yearly_len(
  vanc %>% filter( # existing upgrade 1 with diff type upgrade
    !is.na(upgrade2_year) & install_type != upgrade2_type
  ),
  year_col = "upgrade2_year",
  type_col = "upgrade1_type",
  len_col = "segment_km",
  out_col = "upgrade2_repl_len"
) %>% rename(
```



```

    year = "upgrade2_year",
    type = "upgrade1_type"
  )
vanc_upgrade2_repl

## # A tibble: 18 x 3
## # Groups:   year, type [18]
##   year type upgrade2_repl_len
##   <int> <chr>           <dbl>
## 1  2013 PBL                0
## 2  2013 PL              0.322
## 3  2014 PBL                0
## 4  2014 PL              0.322
## 5  2015 PBL                0
## 6  2015 PL              0.322
## 7  2016 PBL                0
## 8  2016 PL              0.322
## 9  2017 PBL                0
## 10 2017 PL              0.322
## 11 2018 PBL                0
## 12 2018 PL              0.322
## 13 2019 PBL                0
## 14 2019 PL              0.322
## 15 2020 PBL              0.0992
## 16 2020 PL              0.322
## 17 2021 PBL              0.149
## 18 2021 PL              0.322

```

Step 4: Calculate Final Yearly Road Lengths by Infrastructure Type

Join all year road lengths by year and type across installs and upgrades.

Then add road lengths for install and upgrades, and subtract road lengths for replacements from upgrades.

$$length_{year,type}^{install} + length_{year,type}^{upgrade1} + length_{year,type}^{upgrade2} - length_{year,type}^{replacement1} - length_{year,type}^{replacement2}$$

```

vanc_yearly_len <- list(
  vanc_install,
  vanc_upgrade1,
  vanc_upgrade1_repl,
  vanc_upgrade2,
  vanc_upgrade2_repl
) %>%
reduce(
  left_join, by = c("year", "type")
) %>%
ungroup() %>%
mutate(
  across(everything(), ~replace_na(., 0))
) %>%
mutate(
  len = install_len + upgrade1_len + upgrade2_len - upgrade1_repl_len - upgrade2_repl_len
)
vanc_yearly_len

```

```
## # A tibble: 144 x 8
##   year type  install_len upgrade1_len upgrade1_repl_len upgrade2_len
##   <int> <chr>      <dbl>      <dbl>          <dbl>      <dbl>
## 1  1986 BUF         0          0            0          0
## 2  1986 LSB      0.0474         0            0          0
## 3  1986 PBL         0          0            0          0
## 4  1986 PL          0          0            0          0
## 5  1987 BUF         0          0            0          0
## 6  1987 LSB      0.0474         0            0          0
## 7  1987 PBL         0          0            0          0
## 8  1987 PL          0          0            0          0
## 9  1988 BUF         0          0            0          0
## 10 1988 LSB      0.0474         0            0          0
## # i 134 more rows
## # i 2 more variables: upgrade2_repl_len <dbl>, len <dbl>
```

Final Function

$$length_{year,type}^{install} + length_{year,type}^{change_i} - length_{year,type}^{replacement_i}$$

Where:

- $length_{year,type}^{install}$ are the yearly road lengths for an infrastructure *type* installation
- $length_{year,type}^{change_i}$ are the yearly road lengths for an infrastructure *type* change in order *i*
- $length_{year,type}^{replacement_i}$ are the yearly road lengths for an infrastructure *type* replaced by change in order *i*

```
calc_yearly_adj_len <- function(
  df,
  year_cols = c("install_year", "upgrade1_year", "upgrade2_year"),
  type_cols = c("install_type", "upgrade1_type", "upgrade2_type"),
  len_cols = "segment_km",
  out_cols = c("install_len", "upgrade1_len", "upgrade2_len"),
  out_col = "adj_len",
  repl_suffix = "_replaced"
) {

  # Convert len_col if char
  len_cols <- rep(len_cols, length(year_cols))

  # Check cols same size
  year_cols_n <- length(year_cols)
  type_cols_n <- length(type_cols)
  len_cols_n <- length(len_cols)
  out_cols_n <- length(out_cols)
  if (length(unique(c(year_cols_n, type_cols_n, len_cols_n, out_cols_n))) != 1) {
    stop(glue(
      "The arguments 'year_cols' ({year_cols_n}), 'type_cols' ({type_cols_n}), 'len_cols' ({len_c
    ))
  }

  # Calc yearly lens by infra type per install or change
  out <- list()
  for (i in 1:length(year_cols)) {
```

```

# Get year, type, and len cols
ycol <- year_cols[[i]]
tcol <- type_cols[[i]]
lcol <- len_cols[[i]]
ocol <- out_cols[[i]]

# Calc yearly len for install or change
out <- append(
  out,
  calc_yearly_len(
    df,
    year_col = ycol,
    type_col = tcol,
    len_col = lcol,
    out_col = ocol
  ) %>%
  rename(
    "year" := !!ycol,
    "type" := !!tcol
  ) %>% list
)

# Calc yearly len for replacement
if (i > 1) {
  tcol_repl <- type_cols[[i - 1]]
  lcol_repl <- len_cols[[i - 1]]
  out <- append(
    out,
    calc_yearly_len(
      df %>% filter(.data[[tcol]] != .data[[tcol_repl]]),
      year_col = ycol,
      type_col = tcol_repl,
      len_col = lcol_repl,
      out_col = glue("{ocol}{repl_suffix}")
    ) %>%
    rename(
      "year" := !!ycol,
      "type" := !!tcol_repl
    ) %>% list
  )
}
}

# Calc yearly adj lens by infra type
out <- out %>%
  reduce(
    left_join, by = c("year", "type")
  ) %>%
  ungroup() %>%
  mutate( # set unjoined na values to 0
    across(everything(), ~replace_na(., 0))
  ) %>%
  mutate( # added len by infra types due to install or changes

```

```

    !!out_col := reduce(across(all_of(out_cols)), `+`)
  ) %>%
  mutate( # removed len by infra types due to replacements
    !!out_col := .data[[out_col]] - reduce(across(all_of(paste0(out_cols[2:out_cols_n], repl_su
  )
  return(out)
}

```

Test Final Function

```
calc_yearly_adj_len(vanc)
```

```

## # A tibble: 144 x 8
##   year type  install_len upgrade1_len upgrade1_len_replaced upgrade2_len
##   <int> <chr>      <dbl>      <dbl>          <dbl>          <dbl>
## 1  1986 BUF         0          0              0              0
## 2  1986 LSB      0.0474          0              0              0
## 3  1986 PBL         0          0              0              0
## 4  1986 PL          0          0              0              0
## 5  1987 BUF         0          0              0              0
## 6  1987 LSB      0.0474          0              0              0
## 7  1987 PBL         0          0              0              0
## 8  1987 PL          0          0              0              0
## 9  1988 BUF         0          0              0              0
## 10 1988 LSB      0.0474          0              0              0
## # i 134 more rows
## # i 2 more variables: upgrade2_len_replaced <dbl>, adj_len <dbl>

```