

July 9, 2023

The results below are generated from an R script.

```
# Assignment: ASSIGNMENT 5
# Name: Smith, David
# Date: 2023-07-09

## Load the necessary packages for the script and set the default options as necessary.

# ** ggplot2 **
library(ggplot2)
theme_set(theme_minimal())

# ** pastecs **
library(pastecs)
options(scipen=100)
options(digits=2)

# ** moments **
library(moments)

# ** readxl **
library(readxl)

# ** dplyr **
library(dplyr)

# ** purrr **
library(purrr)

# ** scales **
library(scales)

# ** tidyr **
library(tidyr)

# ** stringr **
library(stringr)

## Set the working directory to the root of your DSC 520 directory
setwd("C:\\GitLab-Projects\\Bellevue\\SMITH-DSC520")

## Load the 'data/week-7-housing.xlsx' dataset and get the initial details
housing_df <- read_excel("data/week-7-housing.xlsx")
summary(housing_df)
```

```
##      Sale Date                Sale Price      sale_reason  sale_instrument
##  Min.   :2006-01-03 00:00:00.00  Min.    :    698  Min.    : 0.0  Min.    : 0.0
## 1st Qu.:2008-07-07 00:00:00.00 1st Qu.: 460000  1st Qu.: 1.0  1st Qu.: 3.0
## Median :2011-11-17 00:00:00.00 Median : 593000  Median : 1.0  Median : 3.0
## Mean   :2011-07-28 15:07:32.48 Mean    : 660738  Mean    : 1.6  Mean    : 3.7
## 3rd Qu.:2014-06-05 00:00:00.00 3rd Qu.: 750000  3rd Qu.: 1.0  3rd Qu.: 3.0
## Max.   :2016-12-16 00:00:00.00 Max.    :4400000  Max.    :19.0  Max.    :27.0
## sale_warning      sitetype      addr_full      zip5
## Length:12865      Length:12865      Length:12865      Min.   :98052
## Class :character   Class :character   Class :character   1st Qu.:98052
## Mode  :character   Mode  :character   Mode  :character   Median :98052
##                                     Mean   :98053
##                                     3rd Qu.:98053
##                                     Max.   :98074
##      ctyname      postalctyn      lon      lat      building_grade
## Length:12865      Length:12865      Min.   :-122  Min.   :47  Min.   : 2.0
## Class :character   Class :character   1st Qu.: -122  1st Qu.:48  1st Qu.: 8.0
## Mode  :character   Mode  :character   Median : -122  Median :48  Median : 8.0
##                                     Mean   : -122  Mean   :48  Mean   : 8.2
##                                     3rd Qu.: -122  3rd Qu.:48  3rd Qu.: 9.0
##                                     Max.   : -122  Max.   :48  Max.   :13.0
## square_feet_total_living  bedrooms  bath_full_count bath_half_count bath_3qtr_count
## Min.   : 240              Min.   : 0.0  Min.   : 0.0  Min.   :0.0  Min.   :0.0
## 1st Qu.: 1820              1st Qu.: 3.0  1st Qu.: 1.0  1st Qu.:0.0  1st Qu.:0.0
## Median : 2420              Median : 4.0  Median : 2.0  Median :1.0  Median :0.0
## Mean   : 2540              Mean   : 3.5  Mean   : 1.8  Mean   :0.6  Mean   :0.5
## 3rd Qu.: 3110              3rd Qu.: 4.0  3rd Qu.: 2.0  3rd Qu.:1.0  3rd Qu.:1.0
## Max.   :13540              Max.   :11.0  Max.   :23.0  Max.   :8.0  Max.   :8.0
##      year_built  year_renovated  current_zoning      sq_ft_lot      prop_type
## Min.   :1900      Min.   : 0  Length:12865      Min.   : 785  Length:12865
## 1st Qu.:1979      1st Qu.: 0  Class :character   1st Qu.: 5355  Class :character
## Median :1998      Median : 0  Mode  :character   Median : 7965  Mode  :character
## Mean   :1993      Mean   : 26  Mean   : 22229
## 3rd Qu.:2007      3rd Qu.: 0  3rd Qu.: 12632
## Max.   :2016      Max.   :2016  Max.   :1631322
##      present_use
## Min.   : 0
## 1st Qu.: 2
## Median : 2
## Mean   : 7
## 3rd Qu.: 2
## Max.   :300
```

Identify the baseline distribution information before any modifications.
The post-modification validation may or may not be the same.

```
stat.desc(housing_df$square_feet_total_living)
```

```
##      nbr.val      nbr.null      nbr.na      min      max      range
##      12865.00      0.00      0.00      240.00      13540.00      13300.00
##      sum      median      mean      SE.mean  CI.mean.0.95      var
##      32670747.00      2420.00      2539.51      8.73      17.11      979738.81
##      std.dev      coef.var
##      989.82      0.39
```

```

# shapiro.test(housing_df$square_feet_total_living)
kurtosis(housing_df$square_feet_total_living)

## [1] 12

jarque.test(housing_df$square_feet_total_living)

##
## Jarque-Bera Normality Test
##
## data: housing_df$square_feet_total_living
## JB = 45117, p-value <0.0000000000000002
## alternative hypothesis: greater

# Using the dplyr package, use the 6 different operations to analyze/transform the data - *GroupBy,
# *Summarize, *Mutate, *Filter, Select, and Arrange - Remember this isn't just modifying data, you are
# learning about your data also - so play around and start to understand your dataset in more detail

# Use group_by and summarize to get the mean Sale Price by zip code.
dollar_format(prefix = "$", suffix = "", largest_with_cents = 100000000, big.mark = ",", negative_parens

## function (x)
## {
##     dollar(x, accuracy = accuracy, scale = scale, prefix = prefix,
##           suffix = suffix, big.mark = big.mark, decimal.mark = decimal.mark,
##           trim = trim, largest_with_cents = largest_with_cents,
##           negative_parens = negative_parens, ...)
## }
## <bytecode: 0x000002933864d830>
## <environment: 0x0000029338eb97a0>

housing_mean_by_zip5 <- housing_df %>%
  group_by(zip5) %>%
  summarize(Avg_Sale_Price = dollar(mean('Sale Price'))

# Use mutate to replace the NA values in ctyname column with the corresponding postalctyn values.
# The postalctyn contains the correct values, but without any NA occurrences in the column.
cty_na_df <- filter(housing_df, is.na(housing_df$ctyname))
head(cty_na_df)

## # A tibble: 6 x 24
##   'Sale Date'      'Sale Price' sale_reason sale_instrument sale_warning sitetype
##   <dtm>          <dbl>      <dbl>          <dbl> <chr>      <chr>
## 1 2006-01-03 00:00:00      572500          1          3 <NA>      R1
## 2 2006-01-03 00:00:00     184667          1         15 18 51      R1
## 3 2006-01-04 00:00:00    1050000          1          3 <NA>      R1
## 4 2006-01-04 00:00:00     875000          1          3 <NA>      R1
## 5 2006-01-04 00:00:00     660000          1          3 <NA>      R1
## 6 2006-01-04 00:00:00     165000          1          3 <NA>      R1
## # i 18 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>,
## # lon <dbl>, lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## # bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## # year_built <dbl>, year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## # prop_type <chr>, present_use <dbl>

```

```

cty_na_df <- cty_na_df %>% mutate(ctyname = ifelse (is.na(ctyname),postalctyn,ctyname))
head(cty_na_df)

## # A tibble: 6 x 24
##   'Sale Date'      'Sale Price' sale_reason sale_instrument sale_warning sitetype
##   <dtm>           <dbl>         <dbl>         <dbl> <chr>         <chr>
## 1 2006-01-03 00:00:00      572500           1           3 <NA>         R1
## 2 2006-01-03 00:00:00      184667           1          15 18 51        R1
## 3 2006-01-04 00:00:00     1050000           1           3 <NA>         R1
## 4 2006-01-04 00:00:00      875000           1           3 <NA>         R1
## 5 2006-01-04 00:00:00      660000           1           3 <NA>         R1
## 6 2006-01-04 00:00:00      165000           1           3 <NA>         R1
## # i 18 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>,
## # lon <dbl>, lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## # bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## # year_built <dbl>, year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## # prop_type <chr>, present_use <dbl>

# Filter rows with duplicate values. I know of the 3 rows with 13,540 square feet, but there
# may be others.
housing_df_dupes <- housing_df %>%
  add_count(square_feet_total_living,zip5) %>%
  filter(n>1) %>%
  distinct(zip5)
nrow(housing_df_dupes)

## [1] 3

# Select, and Arrange - Remember this isn't just modifying data
housing_nmbr_df <- housing_df %>% select_if(is.numeric)
housing_nmbr_df

## # A tibble: 12,865 x 16
##   'Sale Price' sale_reason sale_instrument zip5 lon lat building_grade
##   <dbl>         <dbl>         <dbl> <dbl> <dbl> <dbl> <dbl>
## 1      698000           1           3 98052 -122.  47.7      9
## 2      649990           1           3 98052 -122.  47.7      9
## 3      572500           1           3 98052 -122.  47.7      8
## 4      420000           1           3 98052 -122.  47.6      8
## 5      369900           1           3 98052 -122.  47.7      7
## 6      184667           1          15 98053 -122.  47.7      7
## 7     1050000           1           3 98053 -122.  47.7     10
## 8      875000           1           3 98053 -122.  47.7     10
## 9      660000           1           3 98053 -122.  47.7      9
## 10     650000           1           3 98052 -122.  47.6      8
## # i 12,855 more rows
## # i 9 more variables: square_feet_total_living <dbl>, bedrooms <dbl>,
## # bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## # year_built <dbl>, year_renovated <dbl>, sq_ft_lot <dbl>, present_use <dbl>

# Using the purrr package - perform 2 functions on your dataset. You could use zip_n, keep, discard,
# compact, etc.

housing_pluck <- housing_df %>% pluck(1)
head(housing_pluck)

```

```
## [1] "2006-01-03 UTC" "2006-01-03 UTC" "2006-01-03 UTC" "2006-01-03 UTC" "2006-01-03 UTC"
## [6] "2006-01-03 UTC"

housing_df %>% map_df(~(data.frame(n_distinct = n_distinct(.x), class = class(.x))), .id="variable")

##           variable n_distinct      class
## 1      Sale Date      2933  POSIXct
## 2      Sale Date      2933  POSIXt
## 3      Sale Price     4019  numeric
## 4      sale_reason      17  numeric
## 5      sale_instrument    17  numeric
## 6      sale_warning     184 character
## 7      sitetype         7 character
## 8      addr_full      9737 character
## 9      zip5           4  numeric
## 10     ctynome         3 character
## 11     postalctyn      1 character
## 12     lon           9736  numeric
## 13     lat           9733  numeric
## 14     building_grade    12  numeric
## 15 square_feet_total_living 654  numeric
## 16     bedrooms         12  numeric
## 17     bath_full_count     8  numeric
## 18     bath_half_count     7  numeric
## 19     bath_3qtr_count     7  numeric
## 20     year_built        109  numeric
## 21     year_renovated      40  numeric
## 22     current_zoning     24 character
## 23     sq_ft_lot        6038  numeric
## 24     prop_type         1 character
## 25     present_use        8  numeric

# Using the rbind function to add the updated rows from the cty_na_df data frame.
housing_new = rbind(housing_df, cty_na_df)
nrow(housing_new)

## [1] 18943

# Currently both the NA and updated ctynome records exist in the dataframe, so we
# have to drop the NA rows now.
housing_drop <- housing_new %>% drop_na(ctynome)
nrow(housing_drop)

## [1] 12865

# Use the cbind function to create a data frame with sale_price and sale_date without the space in the v
sale_price = c(housing_df$'Sale Price')
sale_date = c(housing_df$'Sale Date')
housing_add_cols_df <- cbind(housing_df, sale_price, sale_date)

# Select, and Arrange - Remember this isn't just modifying data
arrange(housing_add_cols_df, desc(sale_price))

##      Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1  2010-03-02   4400000         1             3         35 45         R1
```

##	2	2010-03-02	4400000	1	3	35	45	R1
##	3	2011-11-17	4380542	1	22	11	45	R1
##	4	2011-11-17	4380542	1	22	11	45	R1
##	5	2011-11-17	4380542	1	22	11	45	R1
##	6	2011-11-17	4380542	1	22	11	45	R1
##	7	2011-11-17	4380542	1	22	11	45	R1
##	8	2011-11-17	4380542	1	22	11	45	R1
##	9	2011-11-17	4380542	1	22	11	45	R1
##	10	2011-11-17	4380542	1	22	11	45	R1
##	11	2011-11-17	4380542	1	22	11	45	R1
##	12	2011-11-17	4380542	1	22	11	45	R1
##	13	2011-11-17	4380542	1	22	11	45	R1
##	14	2011-11-17	4380542	1	22	11	45	R1
##	15	2011-11-17	4380542	1	22	11	45	R1
##	16	2011-11-17	4380542	1	22	11	45	R1
##	17	2016-04-27	4311000	1	3	<NA>		R1
##	18	2011-11-17	4140203	1	3	11	45	R1
##	19	2011-11-17	4140203	1	3	11	45	R1
##	20	2011-11-17	4140203	1	3	11	45	R1
##	21	2011-11-17	4140203	1	3	11	45	R1
##	22	2011-11-17	4140203	1	3	11	45	R1
##	23	2011-11-17	4140203	1	3	11	45	R1
##	24	2011-11-17	4140203	1	3	11	45	R1
##	25	2011-11-17	4140203	1	3	11	45	R1
##	26	2011-11-17	4140203	1	3	11	45	R1
##	27	2011-11-17	4140203	1	3	11	45	R1
##	28	2011-11-17	4140203	1	3	11	45	R1
##	29	2011-11-17	4140203	1	3	11	45	R1
##	30	2011-11-17	4140203	1	3	11	45	R1
##	31	2011-11-17	4140203	1	3	11	45	R1
##	32	2011-11-17	4140203	1	3	11	45	R1
##	33	2008-02-26	3995000	1	3	<NA>		R1
##	34	2016-09-20	3950000	1	3	<NA>		R1
##	35	2016-10-05	3850000	1	3	<NA>		R1
##	36	2016-02-05	3750000	1	3	<NA>		R1
##	37	2012-10-02	3462000	1	3	45		R1
##	38	2012-10-02	3462000	1	3	45		R1
##			addr_full	zip5	ctyname	postalctyn	lon lat	building_grade
##	1		12025 154TH PL NE	98052	<NA>	REDMOND	-122 48	11
##	2		12053 154TH PL NE	98052	<NA>	REDMOND	-122 48	6
##	3		17137 NE 120TH ST	98052	REDMOND	REDMOND	-122 48	8
##	4		11818 171ST PL NE	98052	REDMOND	REDMOND	-122 48	8
##	5		17011 NE 118TH WAY	98052	REDMOND	REDMOND	-122 48	8
##	6		16943 NE 118TH WAY	98052	REDMOND	REDMOND	-122 48	8
##	7		16944 NE 118TH WAY	98052	REDMOND	REDMOND	-122 48	8
##	8		16909 NE 120TH ST	98052	REDMOND	REDMOND	-122 48	8
##	9		17128 NE 120TH ST	98052	REDMOND	REDMOND	-122 48	8
##	10		17136 NE 120TH ST	98052	REDMOND	REDMOND	-122 48	8
##	11		11902 171ST PL NE	98052	REDMOND	REDMOND	-122 48	8
##	12		11719 171ST PL NE	98052	REDMOND	REDMOND	-122 48	8
##	13		16955 NE 118TH WAY	98052	REDMOND	REDMOND	-122 48	8
##	14		11703 169TH PL NE	98052	REDMOND	REDMOND	-122 48	9
##	15		16906 NE 118TH WAY	98052	REDMOND	REDMOND	-122 48	8
##	16		17020 NE 118TH WAY	98052	REDMOND	REDMOND	-122 48	8

## 17	17656 NE 116TH ST	98052 REDMOND	REDMOND -122	48	8
## 18	12212 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 19	12238 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 20	12300 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 21	12308 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 22	12351 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 23	12261 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 24	12249 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 25	12237 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 26	12228 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 27	12250 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 28	12393 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 29	12375 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 30	12285 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 31	12273 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 32	12225 164TH CT NE	98052 REDMOND	REDMOND -122	48	8
## 33	13707 160TH AVE NE	98052 <NA>	REDMOND -122	48	13
## 34	2942 W LAKE SAMMAMISH PKWY	NE 98052 REDMOND	REDMOND -122	48	12
## 35	7917 219TH AVE NE	98053 <NA>	REDMOND -122	48	13
## 36	2808 W LAKE SAMMAMISH PKWY	NE 98052 REDMOND	REDMOND -122	48	11
## 37	10814 240TH AVE NE	98053 <NA>	REDMOND -122	48	8
## 38	10818 240TH AVE NE	98053 <NA>	REDMOND -122	48	8
##	square_feet_total_living	bedrooms	bath_full_count	bath_half_count	bath_3qtr_count
## 1	5790	3	2	1	1
## 2	2410	3	1	0	1
## 3	3290	4	2	1	0
## 4	2450	4	2	1	0
## 5	2750	4	2	1	0
## 6	3010	4	2	0	1
## 7	3200	5	2	1	0
## 8	3200	5	2	1	0
## 9	3620	5	3	0	1
## 10	2810	4	2	1	0
## 11	2550	4	2	1	0
## 12	2440	3	2	1	0
## 13	3160	4	2	1	0
## 14	3400	4	3	0	1
## 15	2960	4	3	0	1
## 16	3110	4	2	0	1
## 17	1670	3	1	1	0
## 18	2900	5	3	1	0
## 19	3220	5	3	1	0
## 20	2510	4	3	1	0
## 21	2970	4	3	1	0
## 22	3470	5	3	1	0
## 23	2580	4	2	1	0
## 24	3590	5	2	1	1
## 25	3890	5	2	1	1
## 26	2520	4	3	1	0
## 27	2990	4	3	1	0
## 28	3620	5	3	0	0
## 29	3300	5	3	1	0
## 30	2830	5	3	0	0
## 31	2680	4	2	1	0

## 32		3330	5	2	1	1	
## 33		11810	7	4	1	4	
## 34		7070	5	3	2	1	
## 35		4690	3	2	1	1	
## 36		6600	4	2	1	1	
## 37		2650	4	2	1	0	
## 38		2520	4	2	1	0	
##	year_built	year_renovated	current_zoning	sq_ft_lot	prop_type	present_use	sale_price
## 1	1999	0	A10	657816	R	2	4400000
## 2	1935	0	A10SO	1327090	R	2	4400000
## 3	2012	0	R4	6712	R	2	4380542
## 4	2010	0	R4	4749	R	2	4380542
## 5	2012	0	R4	5816	R	2	4380542
## 6	2012	0	R4	8908	R	2	4380542
## 7	2010	0	R4	4584	R	2	4380542
## 8	2012	0	R4	4681	R	2	4380542
## 9	2012	0	R4	9901	R	2	4380542
## 10	2012	0	R4	13289	R	2	4380542
## 11	2010	0	R4	4368	R	2	4380542
## 12	2011	0	R4	4244	R	2	4380542
## 13	2011	0	R4	5778	R	2	4380542
## 14	2012	0	R4	6740	R	2	4380542
## 15	2012	0	R4	4451	R	2	4380542
## 16	2012	0	R4	5310	R	2	4380542
## 17	1964	0	R4	425145	R	2	4311000
## 18	2012	0	R4	4647	R	2	4140203
## 19	2012	0	R4	4080	R	2	4140203
## 20	2013	0	R4	5032	R	2	4140203
## 21	2013	0	R4	4383	R	2	4140203
## 22	2012	0	R4	5326	R	2	4140203
## 23	2011	0	R4	5913	R	2	4140203
## 24	2012	0	R4	6254	R	2	4140203
## 25	2012	0	R4	5441	R	2	4140203
## 26	2012	0	R4	4442	R	2	4140203
## 27	2013	0	R4	4234	R	2	4140203
## 28	2013	0	R4	6168	R	2	4140203
## 29	2012	0	R4	4676	R	2	4140203
## 30	2012	0	R4	6695	R	2	4140203
## 31	2012	0	R4	7271	R	2	4140203
## 32	2012	0	R4	4596	R	2	4140203
## 33	2000	0	RA2.5SO	139392	R	2	3995000
## 34	1974	2003	R4	29494	R	2	3950000
## 35	2009	0	RA5	77418	R	2	3850000
## 36	2001	0	R4	41217	R	2	3750000
## 37	2013	0	URPSO	4644	R	2	3462000
## 38	2013	0	URPSO	4887	R	2	3462000
##	sale_date						
## 1	2010-03-02						
## 2	2010-03-02						
## 3	2011-11-17						
## 4	2011-11-17						
## 5	2011-11-17						
## 6	2011-11-17						
## 7	2011-11-17						


```
## 8 2011-11-17
## 9 2011-11-17
## 10 2011-11-17
## 11 2011-11-17
## 12 2011-11-17
## 13 2011-11-17
## 14 2011-11-17
## 15 2011-11-17
## 16 2011-11-17
## 17 2016-04-27
## 18 2011-11-17
## 19 2011-11-17
## 20 2011-11-17
## 21 2011-11-17
## 22 2011-11-17
## 23 2011-11-17
## 24 2011-11-17
## 25 2011-11-17
## 26 2011-11-17
## 27 2011-11-17
## 28 2011-11-17
## 29 2011-11-17
## 30 2011-11-17
## 31 2011-11-17
## 32 2011-11-17
## 33 2008-02-26
## 34 2016-09-20
## 35 2016-10-05
## 36 2016-02-05
## 37 2012-10-02
## 38 2012-10-02
## [ reached 'max' / getOption("max.print") -- omitted 12827 rows ]

# Split a string, then concatenate the results back together
addr_fullsplit <- str_split(string=housing_df$addr_full,pattern = " ")
addr_fullconc <- paste(addr_fullsplit,separator=" ")
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8 LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
```

```
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] stringr_1.5.0  tidyr_1.3.0    scales_1.2.1  purrr_1.0.1   dplyr_1.1.2
## [6] readxl_1.4.2   moments_0.14.1 pastecs_1.3.21 ggplot2_3.4.2
##
## loaded via a namespace (and not attached):
## [1] vctrs_0.6.3      cli_3.6.1       knitr_1.43      rlang_1.1.1     xfun_0.39
## [6] stringi_1.7.12   highr_0.10      generics_0.1.3  glue_1.6.2      colorspace_2.1-0
## [11] tinytex_0.45     fansi_1.0.4     cellranger_1.1.0 grid_4.3.1      munsell_0.5.0
## [16] evaluate_0.21    tibble_3.2.1    lifecycle_1.0.3 compiler_4.3.1  pkgconfig_2.0.3
## [21] rstudioapi_0.14  R6_2.5.1        tidyselect_1.2.0 utf8_1.2.3      pillar_1.9.0
## [26] magrittr_2.0.3   tools_4.3.1     withr_2.5.0     gtable_0.3.3    boot_1.3-28.1

Sys.time()

## [1] "2023-07-09 21:05:10 EDT"
```