The results below are generated from an R script.

```
#setwd(’~/Dsc520’)
library(readr)
library(readxl)
library(dplyr)
library(utils)
library(purrr)
library(tidyverse)

#source_Url <- "http://content.bellevue.edu/cst/dsc/520/id/resources/10-week-housing-data/week-6-housing
#download.file(url=source_Url, destfile = ’data/exercisedata.xlsx’, method=’curl’)
housing <- read_excel(’data/exercisedata.xlsx’)

## Error:  ‘path‘ does not exist:  ’data/exercisedata.xlsx’

#head(housing)
#str(housing)
#View(housing)

#dplyr package
#using summarise()
Total_sales <- housing %>% summarise(total_sales = sum(‘Sale Price‘))
Total_sales

## # A tibble: 1 x 1
##   total_sales
##         <dbl>
## 1  8500391149

#using groupby()
Total_sales_zip <- housing %>% group_by(zip5) %>% summarise(Total_sales =sum(‘Sale Price‘))
#View(Total_sales_zip)

#using select
Housing_sales_zip_bedroom_sqft <- housing %>%
  select(‘Sale Price‘,zip5, bedrooms, square_feet_total_living)
#View(Housing_sales_zip_bedroom_sqft)

#using mutate()
new_housing_byK <- Housing_sales_zip_bedroom_sqft %>%
  mutate(price_in_K = ‘Sale Price‘ / 1000)
#View(new_housing_byK)

#using filter()
housing_byK_expensive <- new_housing_byK %>% filter(price_in_K > 300)
#View(housing_byK_expensive)

#using arrange()
housing_sorted_98052 <- Housing_sales_zip_bedroom_sqft %>%
  filter(zip5 == 98052) %>%
  filter(bedrooms!= 0) %>%
  arrange(desc(‘Sale Price‘))
View(housing_sorted_98052)
```

```
housing_sorted_98053 <- Housing_sales_zip_bedroom_sqft %>%
  filter(zip5 == 98053) %>%
  filter(bedrooms!= 0) %>%
  arrange(desc(‘Sale Price‘))
View(housing_sorted_98053)

#Purrr package function
#using keep()
Top_sale <- Total_sales_zip %>%
  map(sample, 3) %>%
  keep(function(x) mean(x) > 1000000)
Top_sale

## $Total_sales
## [1] 4839145476   69462700     645000

Top_sale2 <- Total_sales_zip %>%
  map(sample, 3) %>%
  discard(function(x) mean(x) < 1000000)
Top_sale2

## $Total_sales
## [1] 3591137973   69462700     645000

#create subset
Bighouse_98053_4bed <- Housing_sales_zip_bedroom_sqft %>%
  filter(zip5 == 98053) %>%
  filter(bedrooms == 4 & square_feet_total_living > 7000) %>%
  arrange(desc(‘Sale Price‘))
#View(Bighouse_98053_4bed)

Bighouse_98053_3bed <- Housing_sales_zip_bedroom_sqft %>%
  filter(zip5 == 98053) %>%
  filter(bedrooms == 3 & square_feet_total_living > 7000) %>%
  arrange(desc(‘Sale Price‘))
#View(Bighouse_98053_3bed)

#using cbind(0)
Sale_price_98053_4bed <- Bighouse_98053_4bed[1]
Sqft_98053_4bed <- Bighouse_98053_4bed[4]
Bighouse_98053 <- cbind(Sale_price_98053_4bed, Sqft_98053_4bed)
#View(Bighouse_98053)

#using rbind()
Copy_98053 <- Bighouse_98053
#View(Copy_98053)
duplicate <- rbind(Bighouse_98053, Copy_98053)
#View(duplicate)

#split strings and concatenate it back
sentence <- "Four score and seven years ago our fathers brought forth on this continent"

c <- unlist(strsplit(sentence," "))
print(c)
```

```
##  [1] "Four"      "score"     "and"       "seven"     "years"     "ago"       "our"
##  [8] "fathers"   "brought"   "forth"     "on"        "this"      "continent"

new_string = paste(c, collapse = ’ ’)
print(new_string, quote = FALSE)

## [1] Four score and seven years ago our fathers brought forth on this continent

#split string in a column and combine them back
#create a new data frame
short_housing <- housing %>%
  select(‘Sale Price‘,zip5, bedrooms, square_feet_total_living, addr_full)

#split address string in addr_full into 2 new columns, delete addr_full column
short_housing[c(’st_number’, ’address’)] <- str_split_fixed(short_housing$addr_full, ’ ’, 2)
new_housing <- short_housing %>% select(-c(’addr_full’))

#paste it back
new_housing$full <- paste(new_housing$st_number, new_housing$address, sep=" ")
#View(new_housing)

#remove the 2 new columns created before
back_housing <- new_housing %>% select(-c(’address’,’st_number’))
#View(back_housing)
```

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/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0   tidyr_1.3.0     tibble_3.2.1
##  [6] ggplot2_3.4.2   tidyverse_2.0.0 purrr_1.0.1     dplyr_1.1.2     readxl_1.4.2
## [11] readr_2.1.4
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.3     compiler_4.3.1   highr_0.10       tidyselect_1.2.0 scales_1.2.1
##  [6] R6_2.5.1         generics_0.1.3   knitr_1.43       munsell_0.5.0    pillar_1.9.0
## [11] tzdb_0.4.0       rlang_1.1.1      utf8_1.2.3       stringi_1.7.12   xfun_0.39
## [16] timechange_0.2.0 cli_3.6.1        withr_2.5.0      magrittr_2.0.3   grid_4.3.1
## [21] rstudioapi_0.14  hms_1.1.3        lifecycle_1.0.3  vctrs_0.6.2      evaluate_0.21
## [26] glue_1.6.2       cellranger_1.1.0 fansi_1.0.4      colorspace_2.1-0 tools_4.3.1
## [31] pkgconfig_2.0.3
```

```r
Sys.time()
```

```
## [1] "2023-07-09 11:01:38 CDT"
```