R - Practice 01 - v1.1

Sangkon Han(sangkon@pusan.ac.kr)

2023-09-10

Contents

dplyr & tidyr

```
df <- mpg
str(df)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
nrow(df); ncol(df)
```

```
## [1] 234
```

```
## [1] 11
```

Manipulate variables(columns)

select(), rename()

```
df.car.info <- select(df, manufacturer, model, year)
head(df.car.info)
```

```
## # A tibble: 6 x 3
##   manufacturer model  year
##   <chr>        <chr> <int>
## 1 audi         a4     1999
## 2 audi         a4     1999
## 3 audi         a4     2008
## 4 audi         a4     2008
## 5 audi         a4     1999
## 6 audi         a4     1999
```

```
select(df, starts_with(match = "m")) %>% head()
```

```
## # A tibble: 6 x 2
##   manufacturer model
##   <chr>        <chr>
## 1 audi         a4
## 2 audi         a4
## 3 audi         a4
## 4 audi         a4
## 5 audi         a4
## 6 audi         a4
```

```
select(df, contains(match = "r")) %>% head()
```

```
## # A tibble: 6 x 4
##   manufacturer  year trans      drv
```

```
##   <chr>        <int> <chr>      <chr>
## 1 audi          1999 auto(l5)   f
## 2 audi          1999 manual(m5) f
## 3 audi          2008 manual(m6) f
## 4 audi          2008 auto(av)   f
## 5 audi          1999 auto(l5)   f
## 6 audi          1999 manual(m5) f
```

```
select(df, ends_with(match = "y")) %>% head()
```

```
## # A tibble: 6 x 2
##     cty   hwy
##   <int> <int>
## 1    18    29
## 2    21    29
## 3    20    31
## 4    21    30
## 5    16    26
## 6    18    26
```

```
select(df, 1:3) %>% head()
```

```
## # A tibble: 6 x 3
##   manufacturer model displ
##   <chr>        <chr> <dbl>
## 1 audi         a4      1.8
## 2 audi         a4      1.8
## 3 audi         a4      2
## 4 audi         a4      2
## 5 audi         a4      2.8
## 6 audi         a4      2.8
```

```
select(df, c(2,5,7)) %>% head()
```

```
## # A tibble: 6 x 3
##   model   cyl drv
##   <chr> <int> <chr>
## 1 a4        4 f
## 2 a4        4 f
## 3 a4        4 f
## 4 a4        4 f
## 5 a4        6 f
## 6 a4        6 f
```

```
select(df, 9:11) %>% head()
```

```
## # A tibble: 6 x 3
##     hwy fl    class
##   <int> <chr> <chr>
## 1    29 p     compact
## 2    29 p     compact
## 3    31 p     compact
## 4    30 p     compact
## 5    26 p     compact
## 6    26 p     compact
```

```
select(df, (ncol(df)-2):ncol(df)) %>% head()
```

```
## # A tibble: 6 x 3
##     hwy fl    class
##   <int> <chr> <chr>
## 1    29 p     compact
## 2    29 p     compact
## 3    31 p     compact
## 4    30 p     compact
## 5    26 p     compact
## 6    26 p     compact
```

```
df1 <- rename(df, mnfc = manufacturer, mod = model)
head(df1)
```

```
## # A tibble: 6 x 11
##   mnfc  mod   displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr> <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi  a4      1.8  1999     4 auto(15)   f        18    29 p     compact
## 2 audi  a4      1.8  1999     4 manual(m5) f        21    29 p     compact
## 3 audi  a4      2    2008     4 manual(m6) f        20    31 p     compact
## 4 audi  a4      2    2008     4 auto(av)   f        21    30 p     compact
## 5 audi  a4      2.8  1999     6 auto(15)   f        16    26 p     compact
## 6 audi  a4      2.8  1999     6 manual(m5) f        18    26 p     compact
```

```
df1 <- select(df, mnfc = manufacturer, mod = model, everything())
head(df1)
```

```
## # A tibble: 6 x 11
##   mnfc  mod   displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr> <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi  a4      1.8  1999     4 auto(15)   f        18    29 p     compact
## 2 audi  a4      1.8  1999     4 manual(m5) f        21    29 p     compact
## 3 audi  a4      2    2008     4 manual(m6) f        20    31 p     compact
## 4 audi  a4      2    2008     4 auto(av)   f        21    30 p     compact
## 5 audi  a4      2.8  1999     6 auto(15)   f        16    26 p     compact
## 6 audi  a4      2.8  1999     6 manual(m5) f        18    26 p     compact
```

mutate() / transmute()

```
df <- mutate(df, `avg miles per gallon` = (cty + hwy) / 2)
head(df)
```

```
## # A tibble: 6 x 12
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(15)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(15)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 1 more variable: `avg miles per gallon` <dbl>
```

```
df <- mutate(df, car = paste(manufacturer, model, sep = " "),
             `cyl / trans` = paste(cyl, " cylinders", " / ", trans, " transmission", sep = ""))
```

```r
head(df)
```

```
## # A tibble: 6 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```r
df1 <- transmute(df, `avg miles per gallon` = (cty + hwy) / 2)
head(df1)
```

```
## # A tibble: 6 x 1
##   `avg miles per gallon`
##                    <dbl>
## 1                   23.5
## 2                   25
## 3                   25.5
## 4                   25.5
## 5                   21
## 6                   22
```

```r
df2 <- mutate(df, car = paste(manufacturer, model, sep = " "),
              `cyl / trans` = paste(cyl, " cylinders", " / ", trans, " transmission", sep = ""))
head(df2)
```

```
## # A tibble: 6 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```r
df2 <- transmute(df, car = paste(manufacturer, model, sep = " "),
              `cyl / trans` = paste(cyl, " cylinders", " / ", trans, " transmission", sep = ""))
head(df2)
```

```
## # A tibble: 6 x 2
##   car     `cyl / trans`
##   <chr>   <chr>
## 1 audi a4 4 cylinders / auto(l5) transmission
## 2 audi a4 4 cylinders / manual(m5) transmission
## 3 audi a4 4 cylinders / manual(m6) transmission
## 4 audi a4 4 cylinders / auto(av) transmission
## 5 audi a4 6 cylinders / auto(l5) transmission
## 6 audi a4 6 cylinders / manual(m5) transmission
```

Manipulate variables(row)

filter(), slice()

```
filter(df, manufacturer == "audi") %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
filter(df, manufacturer == "audi" & year == 1999) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model      displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr>      <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 audi         a4           1.8  1999     4 auto(~ f        18    29 p     comp~
## 2 audi         a4           1.8  1999     4 manua~ f        21    29 p     comp~
## 3 audi         a4           2.8  1999     6 auto(~ f        16    26 p     comp~
## 4 audi         a4           2.8  1999     6 manua~ f        18    26 p     comp~
## 5 audi         a4 quattro   1.8  1999     4 manua~ 4        18    26 p     comp~
## 6 audi         a4 quattro   1.8  1999     4 auto(~ 4        16    25 p     comp~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
df1 <- filter(df, manufacturer == "audi" | manufacturer == "dodge")
head(df1)
```

```
## # A tibble: 6 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
df2 <- filter(df, manufacturer %in% c("audi", "dodge"))
head(df2)
```

```
## # A tibble: 6 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
```

```
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
filter(df, hwy >= 30) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model  displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr>  <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4       2    2008     4 manual(m6) f        20    31 p     comp~
## 2 audi         a4       2    2008     4 auto(av)   f        21    30 p     comp~
## 3 chevrolet    malibu   2.4  2008     4 auto(l4)   f        22    30 r     mids~
## 4 honda        civic    1.6  1999     4 manual(m5) f        28    33 r     subc~
## 5 honda        civic    1.6  1999     4 auto(l4)   f        24    32 r     subc~
## 6 honda        civic    1.6  1999     4 manual(m5) f        25    32 r     subc~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
filter(df, year != 1999) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model      displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr>      <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 audi         a4           2    2008     4 manua~ f        20    31 p     comp~
## 2 audi         a4           2    2008     4 auto(~ f        21    30 p     comp~
## 3 audi         a4           3.1  2008     6 auto(~ f        18    27 p     comp~
## 4 audi         a4 quattro   2    2008     4 manua~ 4        20    28 p     comp~
## 5 audi         a4 quattro   2    2008     4 auto(~ 4        19    27 p     comp~
## 6 audi         a4 quattro   3.1  2008     6 auto(~ 4        17    25 p     comp~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
slice(df, 1:5) %>% head()
```

```
## # A tibble: 5 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
slice(df, 20:30) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model       displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr>       <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 chevrolet    c1500 subu~   5.3  2008     8 auto~ r        11    15 e     suv
## 2 chevrolet    c1500 subu~   5.3  2008     8 auto~ r        14    20 r     suv
## 3 chevrolet    c1500 subu~   5.7  1999     8 auto~ r        13    17 r     suv
## 4 chevrolet    c1500 subu~   6    2008     8 auto~ r        12    17 r     suv
## 5 chevrolet    corvette      5.7  1999     8 manu~ r        16    26 p     2sea~
## 6 chevrolet    corvette      5.7  1999     8 auto~ r        15    23 p     2sea~
```

```
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
slice(df, (nrow(df)-9):nrow(df)) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model      displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr>      <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 volkswagen   new beetle   2    1999     4 auto(~ f        19    26 r     subc~
## 2 volkswagen   new beetle   2.5  2008     5 manua~ f        20    28 r     subc~
## 3 volkswagen   new beetle   2.5  2008     5 auto(~ f        20    29 r     subc~
## 4 volkswagen   passat       1.8  1999     4 manua~ f        21    29 p     mids~
## 5 volkswagen   passat       1.8  1999     4 auto(~ f        18    29 p     mids~
## 6 volkswagen   passat       2    2008     4 auto(~ f        19    28 p     mids~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

arrange

```
# Sort rows by year (ascending order)
arrange(df, year) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model      displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr>      <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 audi         a4           1.8  1999     4 auto(~ f        18    29 p     comp~
## 2 audi         a4           1.8  1999     4 manua~ f        21    29 p     comp~
## 3 audi         a4           2.8  1999     6 auto(~ f        16    26 p     comp~
## 4 audi         a4           2.8  1999     6 manua~ f        18    26 p     comp~
## 5 audi         a4 quattro   1.8  1999     4 manua~ 4        18    26 p     comp~
## 6 audi         a4 quattro   1.8  1999     4 auto(~ 4        16    25 p     comp~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
# Sort rows by year (descending order)
arrange(df, desc(year)) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model      displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr>      <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 audi         a4           2    2008     4 manua~ f        20    31 p     comp~
## 2 audi         a4           2    2008     4 auto(~ f        21    30 p     comp~
## 3 audi         a4           3.1  2008     6 auto(~ f        18    27 p     comp~
## 4 audi         a4 quattro   2    2008     4 manua~ 4        20    28 p     comp~
## 5 audi         a4 quattro   2    2008     4 auto(~ 4        19    27 p     comp~
## 6 audi         a4 quattro   3.1  2008     6 auto(~ 4        17    25 p     comp~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
# Sort rows by year (ascending order), cyl and displ
df.sort <- arrange(df, year, cyl, displ)
head(df.sort)
```

```
## # A tibble: 6 x 14
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
```

```
## 1 honda        civic   1.6  1999     4 manual(m5) f        28    33 r     subco~
## 2 honda        civic   1.6  1999     4 auto(l4)   f        24    32 r     subco~
## 3 honda        civic   1.6  1999     4 manual(m5) f        25    32 r     subco~
## 4 honda        civic   1.6  1999     4 manual(m5) f        23    29 p     subco~
## 5 honda        civic   1.6  1999     4 auto(l4)   f        24    32 r     subco~
## 6 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

distinct

```
df.example <- data.frame(id = 1:3, name = c("John", "Max", "Julia"))
df.example <- bind_rows(df.example, slice(df.example, 2)) # create duplicate of 2nd row
df.example <- arrange(df.example, id)
head(df.example)
```

```
##   id  name
## 1  1  John
## 2  2   Max
## 3  2   Max
## 4  3 Julia
```

```
# show table without duplicates
distinct(df.example) %>% head()
```

```
##   id  name
## 1  1  John
## 2  2   Max
## 3  3 Julia
```

```
#  Back to mpg example - lets create a table with duplicates
df.dupl <- select(df, manufacturer, model)
head(df.dupl)
```

```
## # A tibble: 6 x 2
##   manufacturer model
##   <chr>        <chr>
## 1 audi         a4
## 2 audi         a4
## 3 audi         a4
## 4 audi         a4
## 5 audi         a4
## 6 audi         a4
```

```
#   Keep only unique rows without duplicates
df.nodupl <- distinct(df.dupl)
head(df.nodupl)
```

```
## # A tibble: 6 x 2
##   manufacturer model
##   <chr>        <chr>
## 1 audi         a4
## 2 audi         a4 quattro
## 3 audi         a6 quattro
## 4 chevrolet    c1500 suburban 2wd
## 5 chevrolet    corvette
```

```
## 6 chevrolet    k1500 tahoe 4wd
```

Sample rows

```
# sample_n() - Filter n randomly selected rows
set.seed(42)
```

```
# 10 randomly selected rows without replacement
sample_n(df, size = 10, replace = F) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model       displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr>       <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 dodge        dakota pic~   3.7  2008     6 manu~ 4        15    19 r     pick~
## 2 volkswagen   passat        1.8  1999     4 auto~ f        18    29 p     mids~
## 3 dodge        ram 1500 p~   4.7  2008     8 manu~ 4        12    16 r     pick~
## 4 nissan       pathfinder~   4    2008     6 auto~ 4        14    20 p     suv
## 5 dodge        ram 1500 p~   5.9  1999     8 auto~ 4        11    15 r     pick~
## 6 volkswagen   passat        1.8  1999     4 manu~ f        21    29 p     mids~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
# 10 randomly selected rows with replacement
sample_n(df, size = 10, replace = T) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model       displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr>       <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 dodge        caravan 2wd   3.8  2008     6 auto~ f        16    23 r     mini~
## 2 chevrolet    corvette      5.7  1999     8 manu~ r        16    26 p     2sea~
## 3 dodge        ram 1500 p~   5.2  1999     8 auto~ 4        11    15 r     pick~
## 4 honda        civic         1.6  1999     4 manu~ f        28    33 r     subc~
## 5 ford         f150 picku~   5.4  1999     8 auto~ 4        11    15 r     pick~
## 6 subaru       forester a~   2.5  2008     4 auto~ 4        18    23 p     suv
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
# sample_frac() - Filter a fraction of randomly selected rows
# 10% of table rows randomly selected
sample_frac(df, size = 0.1, replace = F) %>% head()
```

```
## # A tibble: 6 x 14
##   manufacturer model       displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr>       <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 hyundai      sonata        2.4  2008     4 auto~ f        21    30 r     mids~
## 2 land rover   range rover   4    1999     8 auto~ 4        11    15 p     suv
## 3 dodge        caravan 2wd   3.3  1999     6 auto~ f        16    22 r     mini~
## 4 ford         f150 picku~   5.4  1999     8 auto~ 4        11    15 r     pick~
## 5 chevrolet    corvette      6.2  2008     8 auto~ r        15    25 p     2sea~
## 6 subaru       forester a~   2.5  2008     4 auto~ 4        20    26 r     suv
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

summarise

```
# Calculate average hwy
summarise(df, `mean hwy` = mean(hwy)) %>% head()
```

```
## # A tibble: 1 x 1
##   `mean hwy`
##        <dbl>
## 1       23.4
```

```
# Count table rows, and count distinct car models
summarise(df, rows = n(), `nr models` = n_distinct(model)) %>% head()
```

```
## # A tibble: 1 x 2
##    rows `nr models`
##   <int>       <int>
## 1   234          38
```

```
# Calculate min / max hwy & cty
summarise(df, `min hwy` = min(hwy),
          `min cty` = min(cty),
          `max hwy` = max(hwy),
          `max cty` = max(cty))
```

```
## # A tibble: 1 x 4
##   `min hwy` `min cty` `max hwy` `max cty`
##       <int>     <int>     <int>     <int>
## 1        12         9        44        35
```

group_by()

```
# Group cars by manufacturer
group_by(df, manufacturer) %>% head()
```

```
## # A tibble: 6 x 14
## # Groups:   manufacturer [1]
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(15)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(15)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 3 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>
```

```
# Combine summarise() & group_by() - summary statistics for grouped data
# Count number of cars for each manufacturer
summarise(group_by(df, manufacturer), cars = n()) %>% head()
```

```
## # A tibble: 6 x 2
##   manufacturer  cars
##   <chr>        <int>
## 1 audi            18
## 2 chevrolet       19
## 3 dodge           37
```

```
## 4 ford            25
## 5 honda            9
## 6 hyundai         14
```

```
# Calculate mean / min / max hwy for each model
summarise(group_by(df, model),
          `mean hwy` = mean(hwy),
          `min hwy` = min(hwy),
          `max hwy` = max(hwy)) %>% head()
```

```
## # A tibble: 6 x 4
##   model              `mean hwy` `min hwy` `max hwy`
##   <chr>                   <dbl>     <int>     <int>
## 1 4runner 4wd              18.8        17        20
## 2 a4                       28.3        26        31
## 3 a4 quattro               25.8        25        28
## 4 a6 quattro               24          23        25
## 5 altima                   28.7        26        32
## 6 c1500 suburban 2wd       17.8        15        20
```

count()

```
# Count number of table rows
count(df)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   234
```

```
# Count number of cars per model
count(group_by(df, model)) %>% head()
```

```
## # A tibble: 6 x 2
## # Groups:   model [6]
##   model                 n
##   <chr>             <int>
## 1 4runner 4wd           6
## 2 a4                    7
## 3 a4 quattro            8
## 4 a6 quattro            3
## 5 altima                6
## 6 c1500 suburban 2wd    5
```

pipe operator %>%

```
df %>%
  filter(manufacturer == "audi") %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    18
```

```r
df %>%
  filter(manufacturer %in% c("dodge", "chevrolet")) %>%
  select(manufacturer, model, year, class) %>%
  head()
```

```
## # A tibble: 6 x 4
##   manufacturer model              year class
##   <chr>        <chr>             <int> <chr>
## 1 chevrolet    c1500 suburban 2wd  2008 suv
## 2 chevrolet    c1500 suburban 2wd  2008 suv
## 3 chevrolet    c1500 suburban 2wd  2008 suv
## 4 chevrolet    c1500 suburban 2wd  1999 suv
## 5 chevrolet    c1500 suburban 2wd  2008 suv
## 6 chevrolet    corvette            1999 2seater
```

```r
df %>%
  group_by(manufacturer, model, class, trans) %>%
  summarise(`mean hwy` = mean(hwy), cars = n()) %>%
  ungroup() %>%
  filter(`mean hwy` > 30) %>%
  arrange(desc(`mean hwy`)) %>%
  head()
```

```
## # A tibble: 6 x 6
##   manufacturer model      class      trans      `mean hwy`  cars
##   <chr>        <chr>      <chr>      <chr>           <dbl> <int>
## 1 honda        civic      subcompact auto(15)         36       2
## 2 toyota       corolla    compact    manual(m5)       36       2
## 3 toyota       corolla    compact    auto(14)         34       2
## 4 volkswagen   new beetle subcompact manual(m5)       33.7     3
## 5 volkswagen   new beetle subcompact auto(14)         33.5     2
## 6 honda        civic      subcompact auto(14)         32       2
```

pivoting()

```r
table.long <- data.frame(id = 1:6,
                         type = c("a", "b", "a", "c", "c", "a"),
                         count = c(20, 50, 45, 15, 12, 5))
head(table.long)
```

```
##   id type count
## 1  1    a    20
## 2  2    b    50
## 3  3    a    45
## 4  4    c    15
## 5  5    c    12
## 6  6    a     5
```

```r
table.wide <- pivot_wider(table.long,
                          names_from = type,
                          values_from = count)
head(table.wide)
```

```
## # A tibble: 6 x 4
##      id     a     b     c
```

```
##   <int> <dbl> <dbl> <dbl>
## 1     1    20    NA    NA
## 2     2    NA    50    NA
## 3     3    45    NA    NA
## 4     4    NA    NA    15
## 5     5    NA    NA    12
## 6     6     5    NA    NA
```

```
table.long1 <- pivot_longer(table.wide,
                            cols = c("a", "b", "c"),
                            names_to = "type",
                            values_to = "count",
                            values_drop_na = T)
head(table.long1)
```

```
## # A tibble: 6 x 3
##      id type  count
##   <int> <chr> <dbl>
## 1     1 a        20
## 2     2 b        50
## 3     3 a        45
## 4     4 c        15
## 5     5 c        12
## 6     6 a         5
```

```
df.long <- df %>%
  filter(manufacturer %in% c("jeep", "land rover", "hyundai")) %>%
  select(model, trans, hwy) %>%
  group_by(model, trans) %>%
  summarise(`mean hwy` = mean(hwy)) %>%
  ungroup()
head(df.long)
```

```
## # A tibble: 6 x 3
##   model              trans    `mean hwy`
##   <chr>              <chr>         <dbl>
## 1 grand cherokee 4wd auto(14)       18.5
## 2 grand cherokee 4wd auto(15)       17.3
## 3 range rover        auto(14)       15
## 4 range rover        auto(s6)       18
## 5 sonata             auto(14)       27.3
## 6 sonata             auto(15)       28
```

```
df.wide <- df.long %>%
  pivot_wider(names_from = trans,
              values_from = `mean hwy`)
head(df.wide)
```

```
## # A tibble: 4 x 6
##   model              `auto(14)` `auto(15)` `auto(s6)` `manual(m5)` `manual(m6)`
##   <chr>                   <dbl>      <dbl>      <dbl>        <dbl>        <dbl>
## 1 grand cherokee 4wd       18.5       17.3         NA           NA           NA
## 2 range rover              15         NA           18           NA           NA
## 3 sonata                   27.3       28           NA           28           NA
## 4 tiburon                  25.7       NA           NA           27           24
```

```r
df.long1 <- df.wide %>%
  pivot_longer(-model,  # exclude column "model" and use all remaining columns!!!
               names_to = "trans",
               values_to = "mean hwy",
               values_drop_na = T)
head(df.long1)
```

```
## # A tibble: 6 x 3
##   model              trans    `mean hwy`
##   <chr>              <chr>         <dbl>
## 1 grand cherokee 4wd auto(l4)       18.5
## 2 grand cherokee 4wd auto(l5)       17.3
## 3 range rover        auto(l4)       15
## 4 range rover        auto(s6)       18
## 5 sonata             auto(l4)       27.3
## 6 sonata             auto(l5)       28
```

separating and uniting

```r
dates <- seq.Date(from = as.Date("2021-01-01"), to = as.Date("2021-12-31"), by = "day") # generate date
table <- data.frame(date = dates)
table %>% head()
```

```
##         date
## 1 2021-01-01
## 2 2021-01-02
## 3 2021-01-03
## 4 2021-01-04
## 5 2021-01-05
## 6 2021-01-06
```

```r
table %>% tail()
```

```
##           date
## 360 2021-12-26
## 361 2021-12-27
## 362 2021-12-28
## 363 2021-12-29
## 364 2021-12-30
## 365 2021-12-31
```

separate()

```r
table.sep <- table %>%
  separate(data = .,
           col = date,
           into = c("year", "month", "dayofmonth"),
           sep = "-") %>%
  mutate(month = as.numeric(month),
         dayofmonth = as.numeric(dayofmonth)) %>%
  arrange(year, month, dayofmonth)
head(table.sep)
```

```
##   year month dayofmonth
## 1 2021     1          1
```

```
## 2 2021     1          2
## 3 2021     1          3
## 4 2021     1          4
## 5 2021     1          5
## 6 2021     1          6
```

```
table.sep_ <- table %>%
  separate(data = .,
           col = date,
           into = c("year", "month", "dayofmonth"),
           sep = "-") %>%
  mutate_at(.tbl = .,                          # which table? - . stands for table in the pipe line!
            .vars = c("month", "dayofmonth"), # which variables are mutated?
            .funs = as.numeric) %>%           # which functions is applied?
  arrange(year, month, dayofmonth)
head(table.sep_)
```

```
##   year month dayofmonth
## 1 2021     1          1
## 2 2021     1          2
## 3 2021     1          3
## 4 2021     1          4
## 5 2021     1          5
## 6 2021     1          6
```

unite()

```
table.unite <- table.sep %>%
  # add leading zeros
  mutate(month = str_pad(month, width = 2, side = "left", pad = "0"), # add leading zeros to month
         dayofmonth = str_pad(dayofmonth, width = 2, side = "left", pad = "0")) %>% # add leading zeros
  unite(data = .,
        col = "date",
        year, month, dayofmonth,
        sep = "-") %>%
  arrange(date)
head(table.unite)
```

```
##         date
## 1 2021-01-01
## 2 2021-01-02
## 3 2021-01-03
## 4 2021-01-04
## 5 2021-01-05
## 6 2021-01-06
```

```
table.unite_ <- table.sep %>%
  # add leading zeros
  mutate_at(.tbl = .,                                # which table? - . stands for table in the pipe line
            .vars = c("month", "dayofmonth"),       # which variables are mutated?
            .funs = str_pad, 2, "left", "0") %>%   # which functions is applied? - function parameters
  unite(data = .,
        col = "date",
        year, month, dayofmonth,
        sep = "-") %>%
```

```
  arrange(date)
head(table.unite_)
```

```
##         date
## 1 2021-01-01
## 2 2021-01-02
## 3 2021-01-03
## 4 2021-01-04
## 5 2021-01-05
## 6 2021-01-06
```

dplyr and tidyr in action

pull() - extract column as vector

```
df %>% pull(hwy) %>% head()
```

```
## [1] 29 29 31 30 26 26
```

```
df %>% pull(hwy) %>% class() %>% head()
```

```
## [1] "integer"
```

```
df %>% select(hwy) %>% class() %>% head()
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

group_by() + mutate()

```
df <- df %>%
  group_by(manufacturer, model) %>%
  mutate(`mean hwy` = mean(hwy)) %>%
  ungroup()
head(df)
```

```
## # A tibble: 6 x 15
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(15)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(15)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 4 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>, `mean hwy` <dbl>
```

case_when() - case when statements

```
df <- df %>%
  mutate(trans_ = str_sub(string = trans,
                          start = 1,
                          end = 1)) %>%  # extract first letter from trans
  mutate(`transmission type` = case_when(trans_ == "a" ~ "automatic",
                                         trans_ == "m" ~ "manual",
```

```
                                        TRUE ~ "NA")) %>%
  select(-trans_)
df %>% count(`transmission type`, trans)  # check car count
```

```
## # A tibble: 10 x 3
##    `transmission type` trans          n
##    <chr>               <chr>      <int>
##  1 automatic           auto(av)       5
##  2 automatic           auto(l3)       2
##  3 automatic           auto(l4)      83
##  4 automatic           auto(l5)      39
##  5 automatic           auto(l6)       6
##  6 automatic           auto(s4)       3
##  7 automatic           auto(s5)       3
##  8 automatic           auto(s6)      16
##  9 manual              manual(m5)    58
## 10 manual              manual(m6)    19
```

row_number() - add ranks

```
df <- df %>%
  mutate(`car id` = row_number())
head(df)
```

```
## # A tibble: 6 x 17
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 6 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>, `mean hwy` <dbl>, `transmission type` <chr>,
## #   `car id` <int>
```

```
df <- df %>%
  group_by(manufacturer) %>%
  mutate(`car id1` = row_number()) %>%
  ungroup()
head(df)
```

```
## # A tibble: 6 x 18
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 7 more variables: `avg miles per gallon` <dbl>, car <chr>,
## #   `cyl / trans` <chr>, `mean hwy` <dbl>, `transmission type` <chr>,
## #   `car id` <int>, `car id1` <int>
```

Transform table holding flights data

```
df <- hflights
head(df)
```

```
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 5424 2011     1          1         6    1400    1500            AA       428
## 5425 2011     1          2         7    1401    1501            AA       428
## 5426 2011     1          3         1    1352    1502            AA       428
## 5427 2011     1          4         2    1403    1513            AA       428
## 5428 2011     1          5         3    1405    1507            AA       428
## 5429 2011     1          6         4    1359    1503            AA       428
##      TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin Dest Distance
## 5424  N576AA                60      40      -10        0    IAH  DFW      224
## 5425  N557AA                60      45       -9        1    IAH  DFW      224
## 5426  N541AA                70      48       -8       -8    IAH  DFW      224
## 5427  N403AA                70      39        3        3    IAH  DFW      224
## 5428  N492AA                62      44       -3        5    IAH  DFW      224
## 5429  N262AA                64      45       -7       -1    IAH  DFW      224
##      TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 5424      7      13         0                         0
## 5425      6       9         0                         0
## 5426      5      17         0                         0
## 5427      9      22         0                         0
## 5428      9       9         0                         0
## 5429      6      13         0                         0
```

count number of rows/columns, different flights

```
# one flight is represented with!: UniqueCarrier, FlightNum, TailNum, Year, Month, DayofMonth
nrow(df); ncol(df)
```

```
## [1] 227496
```

```
## [1] 21
```

```
df %>%
  count(UniqueCarrier, FlightNum, TailNum, Year, Month, DayofMonth) %>%
  arrange(desc(n)) %>%
  head()
```

```
##   UniqueCarrier FlightNum TailNum Year Month DayofMonth n
## 1            AA       322  N262AA 2011     7          9 1
## 2            AA       322  N435AA 2011     7          2 1
## 3            AA       322  N463AA 2011     6         18 1
## 4            AA       322  N483AA 2011     7         30 1
## 5            AA       322  N491AA 2011     7         16 1
## 6            AA       322  N4XAAA 2011     8         20 1
```

how many columns begin with word "Taxi"?

```
df %>%
  select(starts_with("Taxi")) %>%
  head()
```

```
##      TaxiIn TaxiOut
```

```
## 5424       7       13
## 5425       6        9
## 5426       5       17
## 5427       9       22
## 5428       9        9
## 5429       6       13
```

how many flights were flown less than 1000 miles / greater or equal than 1000 miles

```
df %>%
  mutate(dist1000 = case_when(Distance < 1000  ~ "< 1000 miles",
                              Distance >= 1000 ~ ">= 1000 miles")) %>%
  count(dist1000)
```

```
##        dist1000      n
## 1  < 1000 miles 162107
## 2 >= 1000 miles  65389
```

flights per carrier - sort by top to bottom

```
df %>%
  group_by(UniqueCarrier) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  head()
```

```
## # A tibble: 6 x 2
##   UniqueCarrier     n
##   <chr>         <int>
## 1 XE            73053
## 2 CO            70032
## 3 WN            45343
## 4 OO            16061
## 5 MQ             4648
## 6 US             4082
```

number of cancelled flights per carrier

```
df %>% count(Cancelled) # 1 for cancelled
```

```
##   Cancelled      n
## 1         0 224523
## 2         1   2973
```

```
df %>%
  filter(Cancelled == 1) %>%
  group_by(UniqueCarrier) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  head()
```

```
## # A tibble: 6 x 2
##   UniqueCarrier     n
```

```
##   <chr>         <int>
## 1 XE             1132
## 2 WN              703
## 3 CO              475
## 4 OO              224
## 5 MQ              135
## 6 EV               76
```

percentage of cancelled flights per carrier

```
df %>%
  # count flights break down by cancellation
  group_by(UniqueCarrier, Cancelled) %>%
  count() %>%
  ungroup() %>%
  # calculate total flights
  group_by(UniqueCarrier) %>%
  mutate(`n tot` = sum(n)) %>%
  ungroup() %>%
  # calculate percentages
  mutate(`n percent %` = (n / `n tot`) * 100) %>%
  # keep only cancelled flights - arrange top to bottom
  filter(Cancelled == 1) %>%
  arrange(desc(`n percent %`)) %>%
  head()
```

```
## # A tibble: 6 x 5
##   UniqueCarrier Cancelled     n `n tot` `n percent %`
##   <chr>             <int> <int>   <int>         <dbl>
## 1 EV                    1    76    2204          3.45
## 2 MQ                    1   135    4648          2.90
## 3 B6                    1    18     695          2.59
## 4 AA                    1    60    3244          1.85
## 5 UA                    1    34    2072          1.64
## 6 DL                    1    42    2641          1.59
```

create column date by combining year + month + dayofmonth (remove this columns)

```
df <- df %>%
  # add leading zeros to month and dayofmonth
  mutate_at(.vars = c("Month", "DayofMonth"),
            .funs = str_pad, 2, "left", "0") %>%
  unite(col = "Date", Year, Month, DayofMonth, sep = "-") %>%
  head()
```

check date range

```
df %>%
  summarise(min = min(Date), max = max(Date), n_distinct = n_distinct(Date))
```

```
##          min        max n_distinct
## 1 2011-01-01 2011-01-06          6
```

```r
# count flights per cancellation codes (codes in columns)
# and per carrirs (carriers in rows)
# pivoting required!!!
df %>% count(CancellationCode) # cancellation code "" must have some name other than empty string!
```

```
##   CancellationCode n
## 1                6
```

```r
df %>%
  mutate(CancellationCode = case_when(CancellationCode == "" ~ "0",  # this is not cancelled flight!!!
                                      TRUE ~ CancellationCode)) %>%
  group_by(UniqueCarrier, CancellationCode) %>%
  count() %>%
  ungroup() %>%
  pivot_wider(names_from = CancellationCode,
              values_from = n) %>%
  head()
```

```
## # A tibble: 1 x 2
##   UniqueCarrier   `0`
##   <chr>         <int>
## 1 AA                6
```

Column-wise operations: across()

```r
mpg <- ggplot2::mpg # mpg data
```

summarise() & across()

```r
#  count distinct values in each column
mpg %>%
  summarise(across(.cols = everything(), # which columns:   all columns
                   .fns = n_distinct))   # which function:  count distinct/unique values
```

```
## # A tibble: 1 x 11
##   manufacturer model displ  year   cyl trans   drv   cty   hwy    fl class
##          <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1           15    38    35     2     4    10     3    21    27     5     7
```

```r
mpg %>%
  summarise(across(everything(),
                   n_distinct))
```

```
## # A tibble: 1 x 11
##   manufacturer model displ  year   cyl trans   drv   cty   hwy    fl class
##          <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1           15    38    35     2     4    10     3    21    27     5     7
```

```r
#  calculate mean values for selected columns (list of columns)
mpg %>%
  summarise(across(c(displ, cty, hwy),
                   mean))
```

```
## # A tibble: 1 x 3
##   displ   cty   hwy
##   <dbl> <dbl> <dbl>
```

```
## 1  3.47  16.9  23.4
```

```
#  calculate median value for all numeric columns
mpg %>%
  summarise(across(where(is.numeric),   # "where" clause supports conditions for columns selection!
                   median))
```

```
## # A tibble: 1 x 5
##   displ  year   cyl   cty   hwy
##   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   3.3 2004.     6    17    24
```

```
#  calculate distinct values of character columns
mpg %>%
  summarise(across(where(is.character), n_distinct))
```

```
## # A tibble: 1 x 6
##   manufacturer model trans   drv    fl class
##          <int> <int> <int> <int> <int> <int>
## 1           15    38    10     3     5     7
```

```
#  apply more than one function across multiple columns
#   - calculate mean and median of all numeric columns
mpg %>%
  summarise(across(where(is.numeric),
                   list(avg = ~mean(.x, na.rm = T),     # multiple functions: provided as a list of fun
                        med = ~median(.x, na.rm = T))))
```

```
## # A tibble: 1 x 10
##   displ_avg displ_med year_avg year_med cyl_avg cyl_med cty_avg cty_med hwy_avg
##       <dbl>     <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1      3.47       3.3    2004.    2004.    5.89       6    16.9      17    23.4
## # i 1 more variable: hwy_med <dbl>
```

```
avgmed <- list(avg = ~mean(.x, na.rm = T), med = ~median(.x, na.rm = T))
mpg %>%
  summarise(across(where(is.numeric), avgmed))
```

```
## # A tibble: 1 x 10
##   displ_avg displ_med year_avg year_med cyl_avg cyl_med cty_avg cty_med hwy_avg
##       <dbl>     <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1      3.47       3.3    2004.    2004.    5.89       6    16.9      17    23.4
## # i 1 more variable: hwy_med <dbl>
```

```
#  control names of output columns
mpg %>%
  summarise(across(where(is.numeric), avgmed, .names = "{.fn}:{.col}")) # argument used for column name
```

```
## # A tibble: 1 x 10
##   `avg:displ` `med:displ` `avg:year` `med:year` `avg:cyl` `med:cyl` `avg:cty`
##         <dbl>       <dbl>      <dbl>      <dbl>     <dbl>     <dbl>     <dbl>
## 1        3.47         3.3      2004.      2004.      5.89         6      16.9
## # i 3 more variables: `med:cty` <dbl>, `avg:hwy` <dbl>, `med:hwy` <dbl>
```

```
#  use multiple conditions for column selection
mpg %>%
  summarise(across(where(is.numeric) & ends_with("y"), median))
```

```
## # A tibble: 1 x 2
```

```
##     cty   hwy
##   <dbl> <dbl>
## 1    17    24
```

summarise() ~ group_by() & across()

```
#  calculate sum of all numeric columns break down by car model
mpg %>%
  group_by(model) %>%
  summarise(across(where(is.numeric),
                   sum)) %>%
  ungroup() %>%
  head()
```

```
## # A tibble: 6 x 6
##   model              displ  year   cyl   cty   hwy
##   <chr>              <dbl> <int> <int> <int> <int>
## 1 4runner 4wd         20.9 12012    34    91   113
## 2 a4                  16.3 14020    34   132   198
## 3 a4 quattro          19.4 16028    40   137   206
## 4 a6 quattro          10.1  6015    20    48    72
## 5 altima              16.8 12030    28   124   172
## 6 c1500 suburban 2wd  27.6 10031    40    64    89
```

```
# calculate mean value for selected columns break down by car manufacturer & model
mpg %>%
  group_by(manufacturer, model) %>%
  summarise(across(c(displ, cty, hwy), mean)) %>%
  ungroup() %>%
  head()
```

```
## # A tibble: 6 x 5
##   manufacturer model              displ   cty   hwy
##   <chr>        <chr>              <dbl> <dbl> <dbl>
## 1 audi         a4                  2.33  18.9  28.3
## 2 audi         a4 quattro          2.42  17.1  25.8
## 3 audi         a6 quattro          3.37  16    24
## 4 chevrolet    c1500 suburban 2wd  5.52  12.8  17.8
## 5 chevrolet    corvette            6.16  15.4  24.8
## 6 chevrolet    k1500 tahoe 4wd     5.7   12.5  16.2
```

mutate() & across()

```
#  round up (ceiling) all numeric columns
mpg %>%
  mutate(across(where(is.numeric), ~ceiling(.x))) %>%
  head()
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <dbl> <dbl> <chr>      <chr> <dbl> <dbl> <chr> <chr>
## 1 audi         a4        2  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4        2  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4        2  2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4        2  2008     4 auto(av)   f        21    30 p     compa~
```

```
## 5 audi         a4        3  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4        3  1999     6 manual(m5) f        18    26 p     compa~
```

```
#  convert all character columns to upper case
mpg %>%
  mutate(across(where(is.character),~str_to_upper(.x))) %>%
  head()
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 AUDI         A4      1.8  1999     4 AUTO(L5)   F        18    29 P     COMPA~
## 2 AUDI         A4      1.8  1999     4 MANUAL(M5) F        21    29 P     COMPA~
## 3 AUDI         A4      2    2008     4 MANUAL(M6) F        20    31 P     COMPA~
## 4 AUDI         A4      2    2008     4 AUTO(AV)   F        21    30 P     COMPA~
## 5 AUDI         A4      2.8  1999     6 AUTO(L5)   F        16    26 P     COMPA~
## 6 AUDI         A4      2.8  1999     6 MANUAL(M5) F        18    26 P     COMPA~
```

mutate() ~ group_by() & across()

```
#  calculate mean value for all numeric columns break down by car manufacturer
#   - aggregate mean value of numeric columns for each manufacturer
#   - keep all the rows!
mpg %>%
  group_by(manufacturer) %>%
  mutate(across(where(is.numeric) & -year, # column "year" is removed from calculation!
                ~mean(.x, na.rm = T),
                .names = "{.col}_avg_manufacturer")) %>%
  ungroup() %>%
  head()
```

```
## # A tibble: 6 x 15
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
## # i 4 more variables: displ_avg_manufacturer <dbl>, cyl_avg_manufacturer <dbl>,
## #   cty_avg_manufacturer <dbl>, hwy_avg_manufacturer <dbl>
```

```
# if_any() / if_all() with filter()
# if_any() : keeps the rows where the predicate is true for at least one selected column
# if_all() : keeps the rows where the predicate is true for all selected columns
starwars <- dplyr::starwars # star wars data set
?starwars
```

```
#  filter rows where at least one column doesn't have NA value
starwars %>%
  filter(if_any(.cols = everything(), .fns = ~ !is.na(.x))) %>%
  head()
```

```
## # A tibble: 6 x 14
##   name      height  mass hair_color skin_color eye_color birth_year sex   gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
```

```
## 1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
## 2 C-3PO        167    75 <NA>       gold       yellow         112   none  mascu~
## 3 R2-D2         96    32 <NA>       white, bl~ red             33   none  mascu~
## 4 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
## 5 Leia Org~    150    49 brown      light      brown           19   fema~ femin~
## 6 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
## # i 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
#  filter rows where all columns don't have NA value
starwars %>%
  filter(if_all(.cols = everything(), .fns = ~ !is.na(.x))) %>%
  head()
```

```
## # A tibble: 6 x 14
##   name      height  mass hair_color skin_color eye_color birth_year sex   gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
## 1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
## 2 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
## 3 Leia Org~    150    49 brown      light      brown           19   fema~ femin~
## 4 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
## 5 Beru Whi~    165    75 brown      light      blue            47   fema~ femin~
## 6 Biggs Da~    183    84 black      light      brown           24   male  mascu~
## # i 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
#  filter rows where column "cty" or "hwy" have values greater than 20
mpg %>%
  filter(if_any(c(cty, hwy), ~ . > 20)) %>%  # condition written as function
  head()
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(15)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(15)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

```
#  filter rows where column "cty" and "hwy" have values greater than 20
mpg %>%
  filter(if_all(c(cty, hwy), ~ . > 20)) %>%
  head()
```

```
## # A tibble: 6 x 11
##   manufacturer model  displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr>  <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4       1.8  1999     4 manual(m5) f        21    29 p     comp~
## 2 audi         a4       2    2008     4 auto(av)   f        21    30 p     comp~
## 3 chevrolet    malibu   2.4  2008     4 auto(14)   f        22    30 r     mids~
## 4 honda        civic    1.6  1999     4 manual(m5) f        28    33 r     subc~
## 5 honda        civic    1.6  1999     4 auto(14)   f        24    32 r     subc~
## 6 honda        civic    1.6  1999     4 manual(m5) f        25    32 r     subc~
```