Wrangling

Data Visualization for Social Good

CorrelAid Switzerland

February 2021

Tidyverse

1. The tidyverse is…

   - A collection of user-friendly **packages** for analyzing **tidy data**
   - An **ecosystem** for analytics and data science with common design principles
   - A **dialect** of the R language

ggplot2
Visualization for exploration and reporting

dplyr
Data organization, transformation, and summaries

stringr
Text analysis using regular expressions

readr
Reading rectangular text files (.txt, .csv)

tibble
Lean version of the data frame

tidyr
dplyr-helper package for data pivoting

purrr
Functional programming

forcats
Handling factors

%>%

1 The **novel pipe operator** from the `magrittr` package makes chaining commands easy.

```
# Numeric vector
score <- c(8, 4, 6, 3, 7, 3)
score
```

```
## [1] 8 4 6 3 7 3
```

```
# Mean: Base-R-style
mean(score)
```

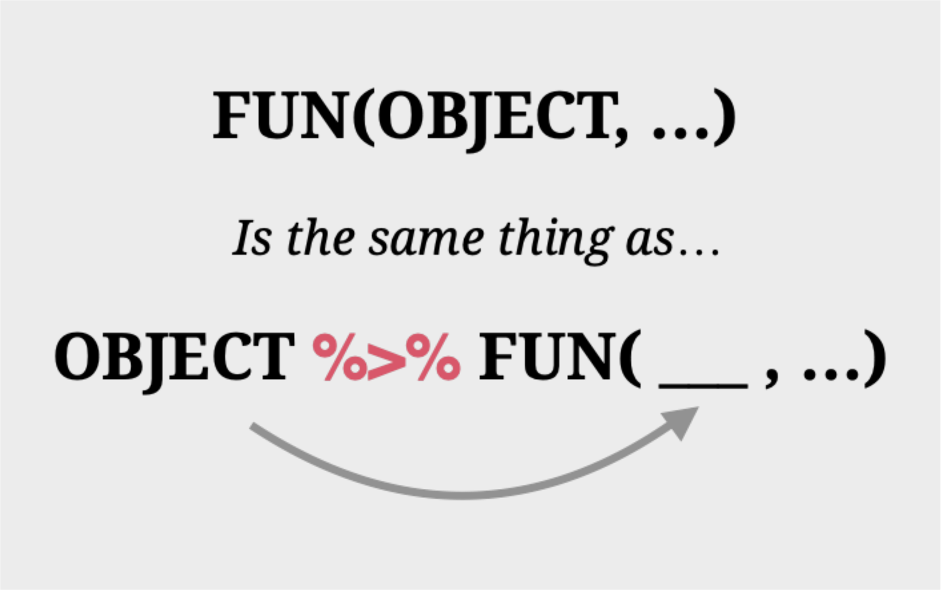
```
## [1] 5.167
```

```
# Mean: dplyr-style
score %>%
  mean()
```

```
## [1] 5.167
```

%>%

1. The **novel pipe operator** from the `magrittr` package makes chaining commands easy.



FUN(OBJECT, ...)

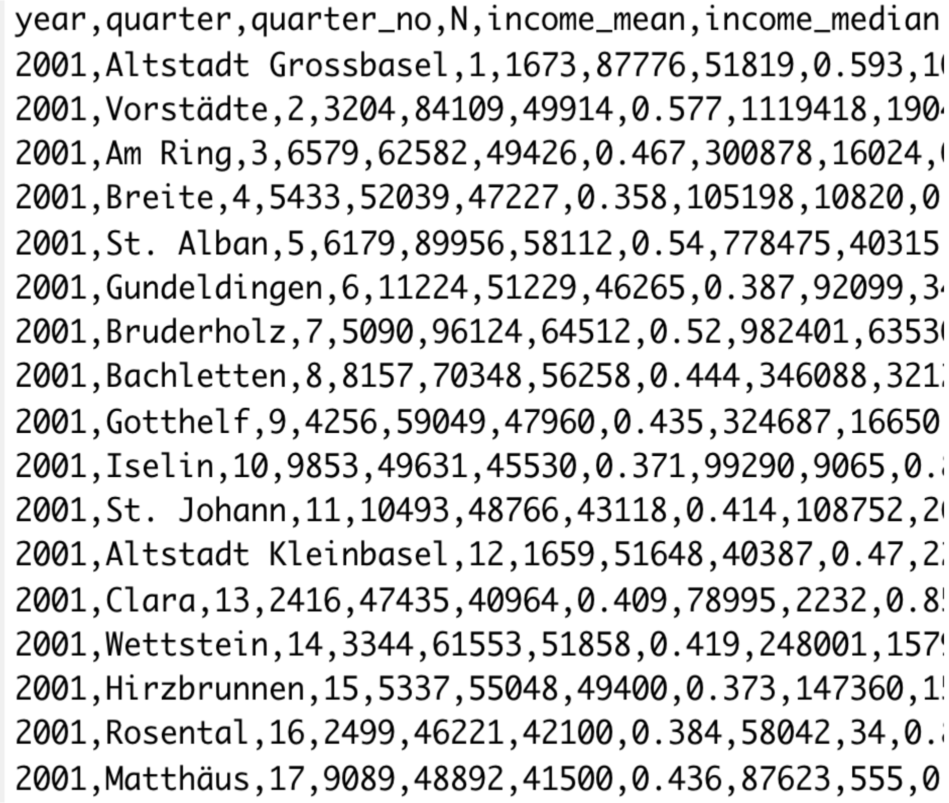
Is the same thing as...

OBJECT %>% FUN(___ , ...)

The **OBJECT** to the left of the pipe %>% becomes the first argument to the **FUN( )** to the right of the pipe

readr

1. Benefits over `read.csv`:
   - Better type inference
   - Avoids `factors`
   - Produces **`tibble`**



```
year,quarter,quarter_no,N,income_mean,income_median
2001,Altstadt Grossbasel,1,1673,87776,51819,0.593,1
2001,Vorstädte,2,3204,84109,49914,0.577,1119418,190
2001,Am Ring,3,6579,62582,49426,0.467,300878,16024,
2001,Breite,4,5433,52039,47227,0.358,105198,10820,0
2001,St. Alban,5,6179,89956,58112,0.54,778475,40315
2001,Gundeldingen,6,11224,51229,46265,0.387,92099,3
2001,Bruderholz,7,5090,96124,64512,0.52,982401,6353
2001,Bachletten,8,8157,70348,56258,0.444,346088,321
2001,Gotthelf,9,4256,59049,47960,0.435,324687,16650
2001,Iselin,10,9853,49631,45530,0.371,99290,9065,0.
2001,St. Johann,11,10493,48766,43118,0.414,108752,2
2001,Altstadt Kleinbasel,12,1659,51648,40387,0.47,2
2001,Clara,13,2416,47435,40964,0.409,78995,2232,0.8
2001,Wettstein,14,3344,61553,51858,0.419,248001,157
2001,Hirzbrunnen,15,5337,55048,49400,0.373,147360,1
2001,Rosental,16,2499,46221,42100,0.384,58042,34,0.
2001,Matthäus,17,9089,48892,41500,0.436,87623,555,0
```

readr

1. Benefits over `read.csv`:
   - Better type inference
   - Avoids factors
   - Produces **`tibble`**

```r
# Read in taxation
basel <- read_csv("1_Data/taxation.csv")

basel
```

```
## # A tibble: 357 x 10
##    year quarter quarter_no     N
##   <dbl> <chr>        <dbl> <dbl>
## 1  2001 Altsta…          1  1673
## 2  2001 Vorstä…          2  3204
## 3  2001 Am Ring          3  6579
## 4  2001 Breite           4  5433
## 5  2001 St. Al…          5  6179
## # … with 352 more rows, and 6 more
## #   variables: income_mean <dbl>,
## #   income_median <dbl>,
## #   income_gini <dbl>,
## #   wealth_mean <dbl>,
## #   wealth_median <dbl>,
## #   wealth_gini <dbl>
```

tibble

1. Benefits over `data.frame`:
   - **Better print**: More informative and cleaner
   - More consistent subsetting

```r
# Read in taxation
basel <- read_csv("1_Data/taxation.csv")

basel
```

```
## # A tibble: 357 x 10
##     year quarter quarter_no     N
##    <dbl> <chr>        <dbl> <dbl>
## 1  2001 Altsta…          1  1673
## 2  2001 Vorstä…          2  3204
## 3  2001 Am Ring          3  6579
## 4  2001 Breite           4  5433
## 5  2001 St. Al…          5  6179
## # … with 352 more rows, and 6 more
## #   variables: income_mean <dbl>,
## #   income_median <dbl>,
## #   income_gini <dbl>,
## #   wealth_mean <dbl>,
## #   wealth_median <dbl>,
## #   wealth_gini <dbl>
```

dplyr

1. Benefits over Base R:
   - **No more brackets**
   - **Data masking**
   - Tidy selection
   - Intuitively named functions

Key verbs	**Purpose**
Transformation	
`rename()`	Rename column names
`mutate()`	Create/change columns
Organization	
`arrange()`	Sort
`select()`	Select variables
`slice()`, `filter()`	Select rows
`left_join()`, `inner_join()`, etc.	Join data sets
Aggregation	
`summarize()`	Calculate statistics
`group()`	Summarize group-wise

select()

```
# Select two columns
TIBBLE %>%
  select(VAR1, VAR2)

# Select everything but
TIBBLE %>%
  select(-VAR1)
```

```
basel %>%

  # Select columns
  select(year, quarter, income_mean)
```

```
## # A tibble: 357 x 3
##    year quarter         income_mean
##   <dbl> <chr>                 <dbl>
## 1  2001 Altstadt Gross…       87776
## 2  2001 Vorstädte             84109
## 3  2001 Am Ring               62582
## 4  2001 Breite                52039
## 5  2001 St. Alban             89956
## 6  2001 Gundeldingen          51229
## 7  2001 Bruderholz            96124
## 8  2001 Bachletten            70348
## # … with 349 more rows
```

filter()

```
# Filter using logical comparisons
TIBBLE %>%
  filter(VAR1 == VAL1,
         VAR2 > VAL2,
         VAR3 < VAL3,
         VAR4 == VAL4 | VAR5 < VAL5)
```

```
basel %>%
    select(year, quarter, income_mean) %>%

  # Select rows rows where year is 2017
  filter(year == 2017)
```

```
## # A tibble: 21 x 3
##    year quarter         income_mean
##   <dbl> <chr>                 <dbl>
## 1  2017 Altstadt Gross…       97111
## 2  2017 Vorstädte            103714
## 3  2017 Am Ring               78761
## 4  2017 Breite                56888
## 5  2017 St. Alban            102457
## 6  2017 Gundeldingen          56544
## 7  2017 Bruderholz           105973
## 8  2017 Bachletten            81580
## # … with 13 more rows
```

arrange()

```
# Sort ascending
TIBBLE %>%
  arrange(VAR1, VAR2)

# Sort descending w/ desc()
TIBBLE %>%
  arrange(desc(VAR1), VAR2)
```

```
basel %>%
  select(year, quarter, income_mean) %>%
  filter(year == 2017) %>%

  # Sort by income
  arrange(income_mean)
```

```
## # A tibble: 21 x 3
##    year quarter       income_mean
##   <dbl> <chr>               <dbl>
## 1  2017 Klybeck             41569
## 2  2017 Kleinhüningen       45664
## 3  2017 Clara               50680
## 4  2017 Matthäus            50786
## 5  2017 Iselin              51600
## 6  2017 St. Johann          52890
## 7  2017 Rosental            54543
## 8  2017 Gundeldingen        56544
## # … with 13 more rows
```

summarize()

```
# Create new summary variables
TIBBLE %>%
  summarise(
    NAME1 = SUMMARY_FUN(VAR1),
    NAME2 = SUMMARY_FUN(VAR2)
  )
```

```
basel %>%
  filter(year == 2017) %>%

  # Calculate averages in 2017
  summarize(
    income = mean(income_mean),
    wealth = mean(wealth_mean))
```

```
## # A tibble: 1 x 2
##   income  wealth
##    <dbl>   <dbl>
## 1 72388. 560333.
```

group_by()

```
# Create grouped summary variables
TIBBLE %>%
  group_by(GRUPPEN_VAR) %>%
  summarise(
    NAME1 = SUMMARY_FUN(VAR1),
    NAME2 = SUMMARY_FUN(VAR2)
  )
```

```
basel %>%

  # Calculate averages for all years
  group_by(year) %>%
  summarize(
    income = mean(income_mean),
    wealth = mean(wealth_mean))
```

```
## # A tibble: 17 x 3
##    year income  wealth
##   <dbl>  <dbl>   <dbl>
## 1  2001 63027. 347770.
## 2  2002 63555. 367401.
## 3  2003 63083. 373278.
## 4  2004 62298. 353968.
## 5  2005 63133. 441864.
## 6  2006 64148. 465242.
## 7  2007 66594  435270.
## 8  2008 66463. 401131.
## # … with 9 more rows
```

group_by()

```
# Create grouped summary variables
TIBBLE %>%
  group_by(GRUPPEN_VAR) %>%
  summarise(
    NAME1 = SUMMARY_FUN(VAR1),
    NAME2 = SUMMARY_FUN(VAR2)
  )
```

```
basel %>%

  # Calculate averages for all years
  group_by(year) %>%
  summarize(
    income = mean(income_mean),
    wealth = mean(wealth_mean)) %>%
  arrange(income)
```

```
## # A tibble: 17 x 3
##    year income  wealth
##   <dbl>  <dbl>   <dbl>
## 1  2004 62298. 353968.
## 2  2001 63027. 347770.
## 3  2003 63083. 373278.
## 4  2005 63133. 441864.
## 5  2002 63555. 367401.
## 6  2006 64148. 465242.
## 7  2011 66050. 398102.
## 8  2008 66463. 401131.
## # … with 9 more rows
```

*_join()

```
# Join two tibbles
TIBBLE1 %>%
  left_join(TIBBLE2,
            by = c("KEY1" = "KEY2"))
```

```
basel %>%
  group_by(year) %>%
  summarize(
    income = mean(income_mean),
    wealth = mean(wealth_mean)) %>%

  # join back to basel
  right_join(basel)
```

```
## # A tibble: 357 x 12
##     year income wealth quarter
##    <dbl>  <dbl>  <dbl> <chr>
## 1   2001 63027. 3.48e5 Altsta…
## 2   2001 63027. 3.48e5 Vorstä…
## 3   2001 63027. 3.48e5 Am Ring
## 4   2001 63027. 3.48e5 Breite
## 5   2001 63027. 3.48e5 St. Al…
## 6   2001 63027. 3.48e5 Gundel…
## 7   2001 63027. 3.48e5 Bruder…
## 8   2001 63027. 3.48e5 Bachle…
## # … with 349 more rows, and 8 more
## #   variables: quarter_no <dbl>,
## #   N <dbl>, income_mean <dbl>,
## #   income_median <dbl>,
## #   income_gini <dbl>,
## #   wealth_mean <dbl>, …
```

tidyr

1. Benefits over Base R:
   - Did not exist before.

adapted from tidyexplain

pivot_longer()

```
# wide to long
TIBBLE %>%
  pivot_longer(cols = VARS,
               names_to = NAME1,
               values_to = NAME2)
```

```
# wide to long
basel %>%
  select(year, quarter,
         income_mean, wealth_mean) %>%
  pivot_longer(c(income_mean, wealth_mean))
```

```
## # A tibble: 714 x 4
##    year quarter      name     value
##   <dbl> <chr>        <chr>    <dbl>
## 1  2001 Altstadt Gr… income… 8.78e4
## 2  2001 Altstadt Gr… wealth… 1.01e6
## 3  2001 Vorstädte    income… 8.41e4
## 4  2001 Vorstädte    wealth… 1.12e6
## 5  2001 Am Ring      income… 6.26e4
## 6  2001 Am Ring      wealth… 3.01e5
## 7  2001 Breite       income… 5.20e4
## 8  2001 Breite       wealth… 1.05e5
## # … with 706 more rows
```

Practical