Week 11

Business Research Methods

Bhaswar Chakma

04 May 2021

Learning Objectives

- Factor variable
- Data frame:
  - `data.frame()`
  - `tibble()`
  - `tribble()`
- Applying `lm()`

Factor Variables

- to work with categorical variables, variables that have a fixed and known set of possible values.
- to display character vectors in a non-alphabetical order.

```r
# Create a vector of country names using c()
country1 <- c("China", "Bangladesh", "Australia")

# Check type
typeof(country1)
```

```
## [1] "character"
```

```r
# Convert to factor
country2 <- factor(country1)
country2
```

```
## [1] China      Bangladesh Australia
## Levels: Australia Bangladesh China
```

- By default the levels of a factor are arranged alphabetically

```r
# Sort using sort()
sort(country1)
```

```
## [1] "Australia"  "Bangladesh" "China"
```

```r
sort(country2)
```

```
## [1] Australia  Bangladesh China
## Levels: Australia Bangladesh China
```

```r
# Factor with different Levels
country3 <- factor(
  country1,
  levels = c("Bangladesh", "China", "Australia")
)

sort(country3)
```

```
## [1] Bangladesh China      Australia
## Levels: Bangladesh China Australia
```

Example: Plot

```
library(tidyverse)
```

```
?mpg
```

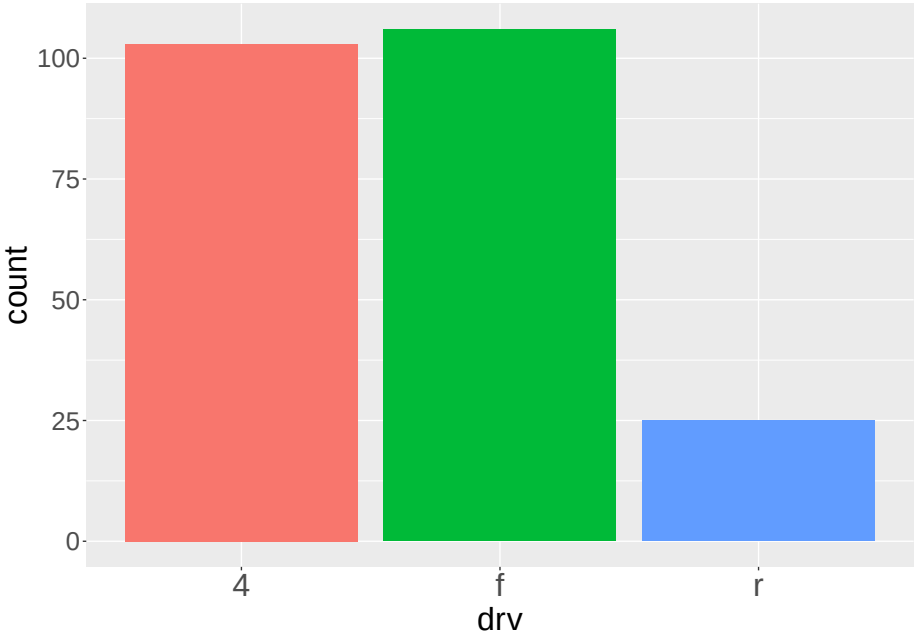
```
table(mpg$drv)
```

```
##
##   4   f   r
## 103 106  25
```

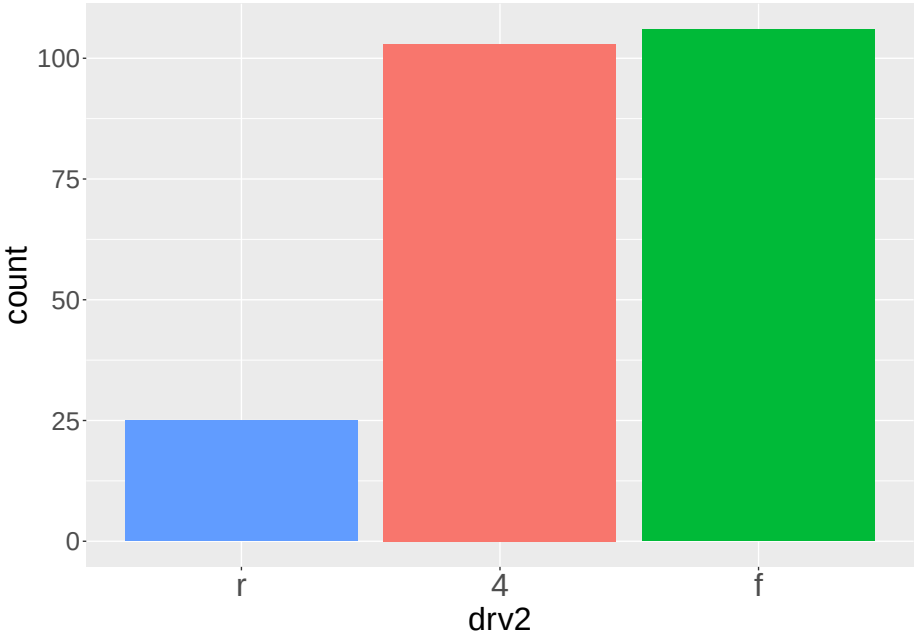
`drv` variable:

- f = front-wheel drive
- r = rear wheel drive
- 4 = 4wd

```
ggplot(data = mpg, aes(x = drv, fill = drv)) +
  geom_bar()
```


count
100
75
50
25
0
4
f
r
drv

```r
mpg %>%
  # Generate a factor variable: drv2
  mutate(
    drv2 = factor(drv, levels = c("r", "4", "f"))
  ) %>%
  ggplot(aes(x = drv2, fill = drv)) +
  geom_bar()
```


100
75
50
25
0
count
r
4
f
drv2

Example: Regression

```
library(MASS)
```

```
# From last class
Boston <- Boston %>%
  mutate(ctax = case_when(
    tax < 250 ~ "low",
    tax > 300 ~ "high",
    TRUE ~ "medium"
  ))
```

```r
# Today
# Generate new variables ctax2 and ctax3 (ctax as factor)
Boston <- Boston %>%
  mutate(ctax2 = factor(ctax),
         ctax3 = factor(
           ctax,
           levels = c("low", "high", "medium"))
         )
```

```r
# From last class
m4 <- lm(medv ~ lstat + ctax, data = Boston)
m4
```

```
##
## Call:
## lm(formula = medv ~ lstat + ctax, data = Boston)
##
## Coefficients:
## (Intercept)        lstat      ctaxlow   ctaxmedium
##     33.3033      -0.9033       2.9465       1.2634
```

```r
# Today! ctax2
m4a <- lm(medv ~ lstat + ctax2, data = Boston)
m4a
```

```
##
## Call:
## lm(formula = medv ~ lstat + ctax2, data = Boston)
##
## Coefficients:
## (Intercept)        lstat     ctax2low  ctax2medium
##     33.3033      -0.9033       2.9465       1.2634
```

```r
# Today! ctax3
m4b <- lm(medv ~ lstat + ctax3, data = Boston)
m4b
```

```
##
## Call:
## lm(formula = medv ~ lstat + ctax3, data = Boston)
##
## Coefficients:
## (Intercept)        lstat    ctax3high  ctax3medium
##     36.2497      -0.9033      -2.9465      -1.6830
```

broom

- `broom::tidy()`
- `broom::augment()`
- `broom::glance()`

```r
# Use the model m4
summary(m4)
```

```
##
## Call:
## lm(formula = medv ~ lstat + ctax, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.549  -3.995  -1.202   1.972  25.305
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.30325    0.68752  48.439  < 2e-16 ***
```

```
broom::tidy(m4)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   33.3      0.688      48.4  2.32e-191
## 2 lstat         -0.903    0.0412    -21.9  3.68e- 75
## 3 ctaxlow        2.95     0.843       3.49 5.16e-  4
## 4 ctaxmedium     1.26     0.731       1.73 8.47e-  2
```

```
broom::augment(m4)
```

```
## # A tibble: 506 x 9
##     medv lstat ctax   .fitted .resid    .hat .sigma     .c
##    <dbl> <dbl> <chr>    <dbl>  <dbl>   <dbl>  <dbl>
##  1  24    4.98 medium   30.1  -6.07  0.0104    6.15 0.002
##  2  21.6  9.14 low      28.0  -6.39  0.0145    6.15 0.004
##  3  34.7  4.03 low      32.6   2.09  0.0157    6.16 0.000
##  4  33.4  2.94 low      33.6  -0.194 0.0162    6.16 0.000
##  5  36.2  5.33 low      31.4   4.76  0.0151    6.15 0.002
##  6  28.7  5.21 low      31.5  -2.84  0.0152    6.15 0.000
##  7  22.9 12.4  high     22.1   0.825 0.00319   6.16 0.000
##  8  27.1 19.2  high     16.0  11.1   0.00395   6.14 0.003
##  9  16.5 29.9  high      6.27 10.2   0.0136    6.14 0.009
```

```
broom::glance(m4)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>
## 1     0.556         0.553  6.15      209. 5.34e-88     3
## # ... with 3 more variables: deviance <dbl>, df.residual
```

Questions?

bhaswar.chakma@ucp.pt