Advanced Data Management & Graphics in R/RStudio

Groups (class size : 34)

- **BKA** - Brooke Deering, Kevin Iriawan, Austin Katter
- **CMD** - Chandler Bascue, Max Ntege, Daniel Warner
- **EMS** - Euntaek Jo, Marcella Norris, Seunghyuk Yang
- **JEW** - Jordan Fahey, Eric Hurd, Will Laviano
- **KBS** - Kyler Miller, Bob Morehouse, Shayn Tan
- **KCP** - Katelyn Bennett, Clark Johnson, Peter Kovalyashkin
- **MT** - Margarita Duran Espino, Tanner Hermanson
- **RMM** - Ryan De La Fe, McKenna Hackney, Maia Lance
- **SSM** - Sloan Felton, Steven Milo, Moe Strid
- **TAW** - Taha Mirghorbaninokandeh, Alex Qiu, Wyatt West
- **TI** - Tingjun Chen, Ivan Een
- **TRM** - Taylor Jitngamplang, Ryan Johnson, Meagan Kiefer

Homework 1

- Available on canvas
- Canvas → Assignments → Homework 1
- Due by 18th January 2022, 11:59 pm

Previous Class

- Installation and loading tidyverse packages in R/RStudio
- Setting working directory
- Reading CSV data into R/RStudio
  - Used toyota corolla cars data
- Class and structure of data in R/RStudio
- Data Management
  - Summarize variables
  - Summarize variables by group(s)
  - Filter observation(s)
  - Sorting by variable(s)
  - Creating new variable(s)
  - ……

Today's agenda

- Quick review of basic data operations
- Advanced Data Management & Graphics in R/RStudio
- Advanced Operations
  - Tidying
  - Binding
  - Appending
  - Merging
  - Long ↔ Wide
  - ……
- Graphics
  - Histogram, Bar chart
  - Scatter plot, Boxplot

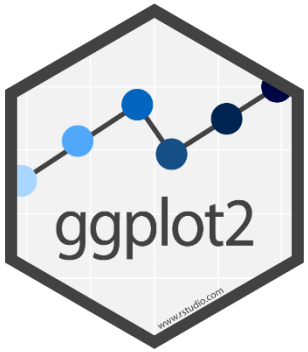
Tidyverse

- Collection of packages for Data Science in R

Data Import

Data Manipulation

Data Tidying



Graphics

Advanced Functions

Data frames

String manipulation

Categorical variables

- Explore https://www.tidyverse.org/ for more details

Data Manipulation

Data Manipulation

- dplyr is grammar of data manipulation
- Functions in this package help you solve the most common data manipulation challenges

Commonly used functions

- **mutate( )** : adds new variables that are functions of existing variables or replaces the existing variables with values of your choice
- **select( )** : picks variables based on their names
- **filter( )** : picks observations based on their values
- **summarise( )** : reduces multiple values down to single summary
- **arrange( )** : change the ordering of observations
- Explore https://dplyr.tidyverse.org/ for more details

Code review from last class

Mandatory steps

- Create a folder name "**d.data_mgmt2**" within the folder "**oba_455_555_ddpm_r/rproject**"
- Download "**data_mgmt2_code.R**" and "**toyota_corolla.csv**" files from canvas
- Place both in location "**oba_455_555_ddpm_r/rproject/d.data_mgmt2**"
- Open RStudio project
- Open "**data_mgmt2_code.R**" file within RStudio

Joining Operations

- Left Join
- Right Join
- Inner Join
- Full Join

Left Join

temp1

id	price
1	13500
2	13750
3	13950
4	14950
5	13750
6	12950
7	16900

temp2

id	age_08_04
3	24
4	26
5	30
6	32
7	27
8	30
9	27
10	23

merge

id	price	age_08_04
1	13500	NA
2	13750	NA
3	13950	24
4	14950	26
5	13750	30
6	12950	32
7	16900	27

```
lj = temp1 %>%
        left_join(temp2, by = "id")
```

Right Join

temp1

id	price
1	13500
2	13750
3	13950
4	14950
5	13750
6	12950
7	16900

temp2

id	age_08_04
3	24
4	26
5	30
6	32
7	27
8	30
9	27
10	23

merge

id	price	age_08_04
3	13950	24
4	14950	26
5	13750	30
6	12950	32
7	16900	27
8	NA	30
9	NA	27
10	NA	23

```
rj = temp1 %>%
        right_join(temp2, by = "id")
```

Inner Join

temp1

id	price
1	13500
2	13750
3	13950
4	14950
5	13750
6	12950
7	16900

temp2

id	age_08_04
3	24
4	26
5	30
6	32
7	27
8	30
9	27
10	23

merge

id	price	age_08_04
3	13950	24
4	14950	26
5	13750	30
6	12950	32
7	16900	27

```
ij = temp1 %>%
        inner_join(temp2, by = "id")
```

Full Join

temp1

id	price
1	13500
2	13750
3	13950
4	14950
5	13750
6	12950
7	16900

temp2

id	age_08_04
3	24
4	26
5	30
6	32
7	27
8	30
9	27
10	23

merge

id	price	age_08_04
1	13500	NA
2	13750	NA
3	13950	24
4	14950	26
5	13750	30
6	12950	32
7	16900	27
8	NA	30
9	NA	27
10	NA	23

```
fj = temp1 %>%
        full_join(temp2, by = "id")
```

Data Tidying

Data Tidying

- Goal of tidyr package is to create tidy data
- If you ensure that your data is tidy, you'll spend **less time** on **managing data** and **more time** working on your analysis
- Long ↔ Wide

Commonly used functions

- **pivot_longer( ) :** takes multiple columns, and gathers them into key-value pairs: it makes "wide" data longer
- **pivot_wider( ) :** takes two columns (key & value), and spreads into multiple columns: it makes "long" data wider
- Explore https://tidyr.tidyverse.org/ for more details

pivot_longer( )

temp1

company	Y1999	Y2000
A	0.7	2
B	37	80
C	212	213

temp2

company	year	cost
A	Y1999	0.7
B	Y1999	37
C	Y1999	212
A	Y2000	2
B	Y2000	80
C	Y2000	213

```
temp2 = temp1 %>%
        pivot_longer(cols = Y1999:Y2000, names_to = "year", values_to = “cost")
```

pivot_wider( )

temp1

company	year	type	exp
A	1999	cost	0.7
A	1999	revenue	19
A	2000	cost	2
A	2000	revenue	20
B	1999	cost	37
B	1999	revenue	172
B	2000	cost	80
B	2000	revenue	174
C	1999	cost	212
C	1999	revenue	1000
C	2000	cost	213
C	2000	revenue	1000

temp2

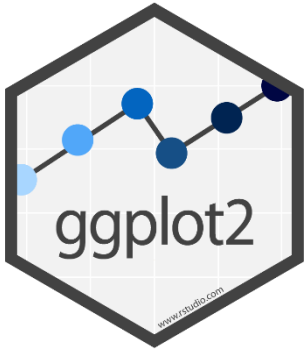
company	year	cost	revenue
A	1999	0.7	19
A	2000	2	20
B	1999	37	172
B	2000	80	174
C	1999	212	1000
C	2000	213	1000

```
temp2 = temp1 %>%
        pivot_wider(names_from = type, values_from = exp)
```

Handling Missing values

- Missing numeric/character data in R is represented by **NA**
- Missing values can lead to incorrect analysis
- Pay keen attention to missing values
- Actions
    - Delete observations
    - Replace with a value
- No correct action
- Depends on data, context, the extent to which it is a problem
- Make conscious action and support why you are doing it

Graphics



Graphics

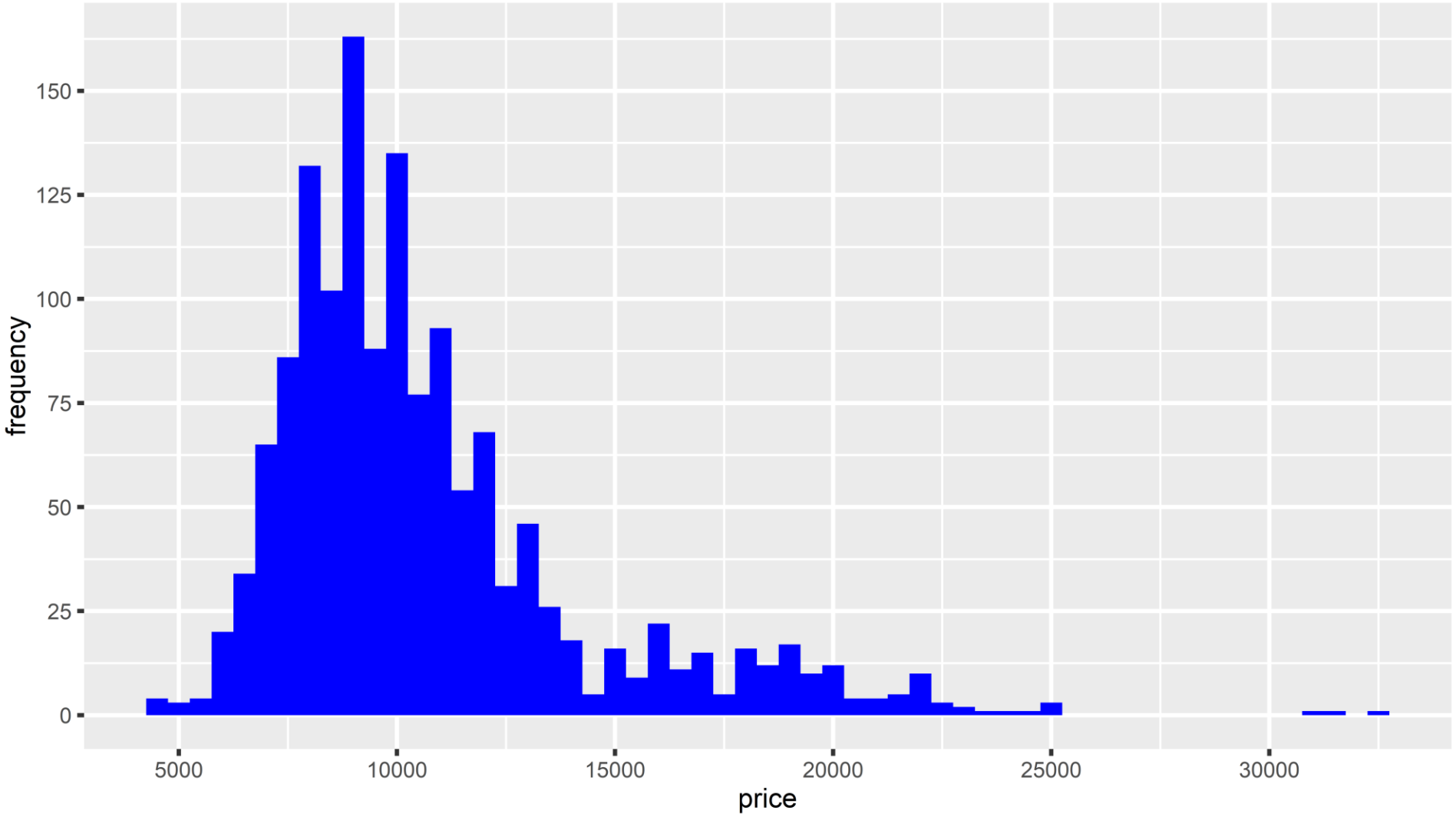
- ggplot2 is a system for declaratively creating graphics
- You provide the data, tell ggplot2
  - How to map variables to aesthetics
  - What graphical primitives to use, and it takes care of the details
- Explore https://ggplot2.tidyverse.org/ for more details

Single variable : Numeric

- **Histogram /Frequency**
- Area
- Density
- Dot plot
- QQ

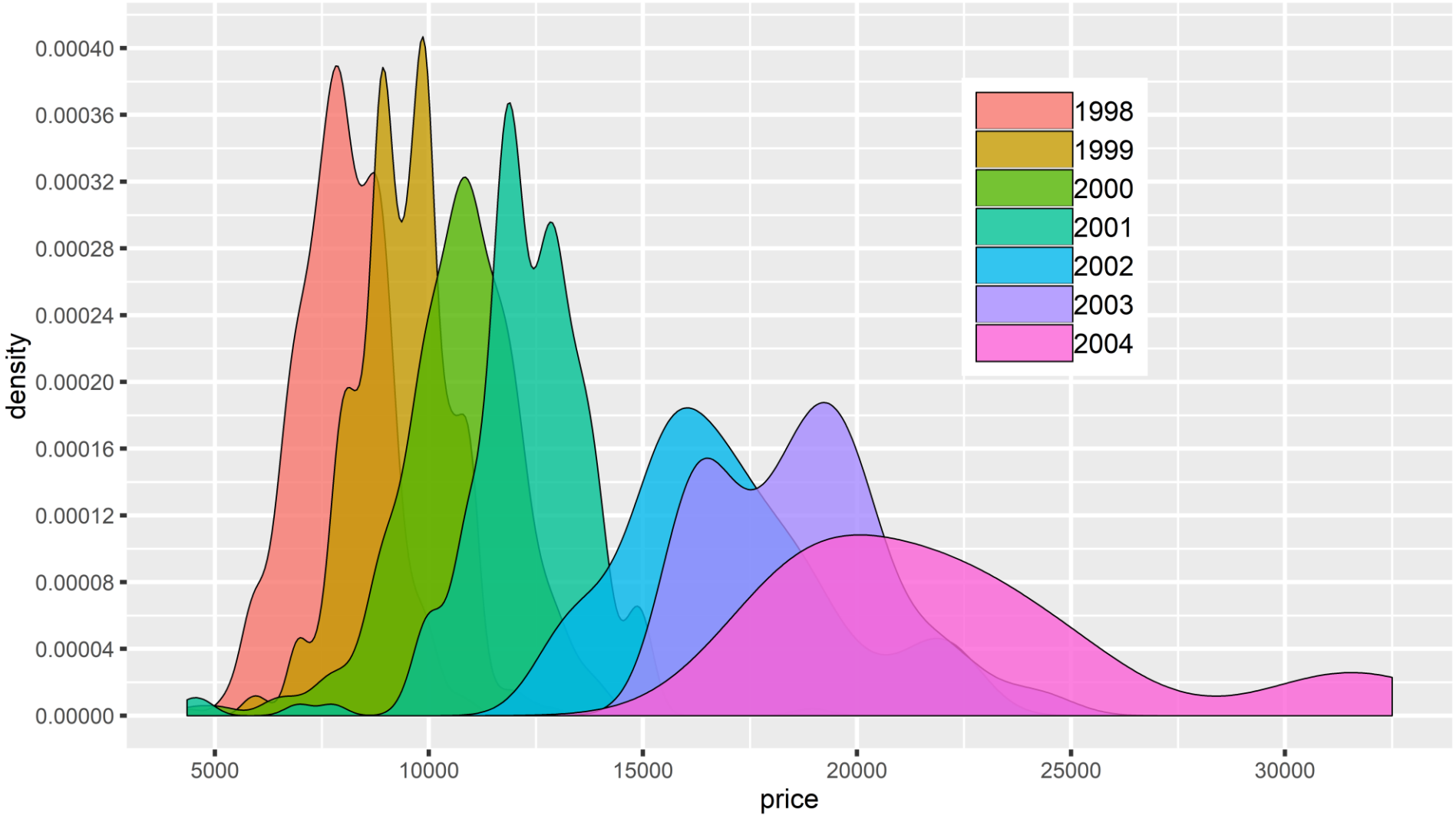
Single variable : Numeric

- Histogram of price



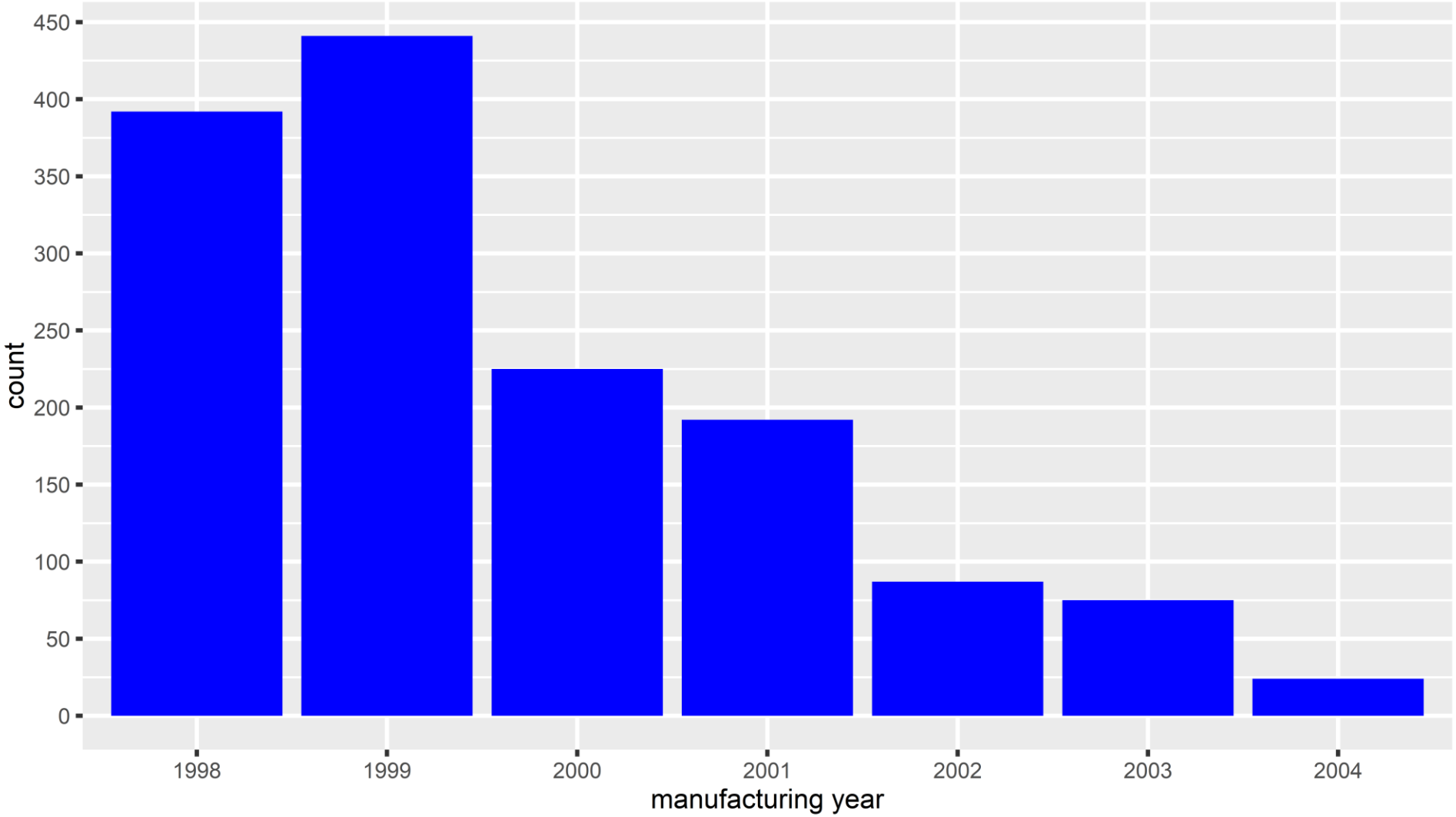
Single variable : Numeric

- Histogram of price by manufacturing year



Single variable : Discrete

- Bar plot of cars sold by manufacturing year

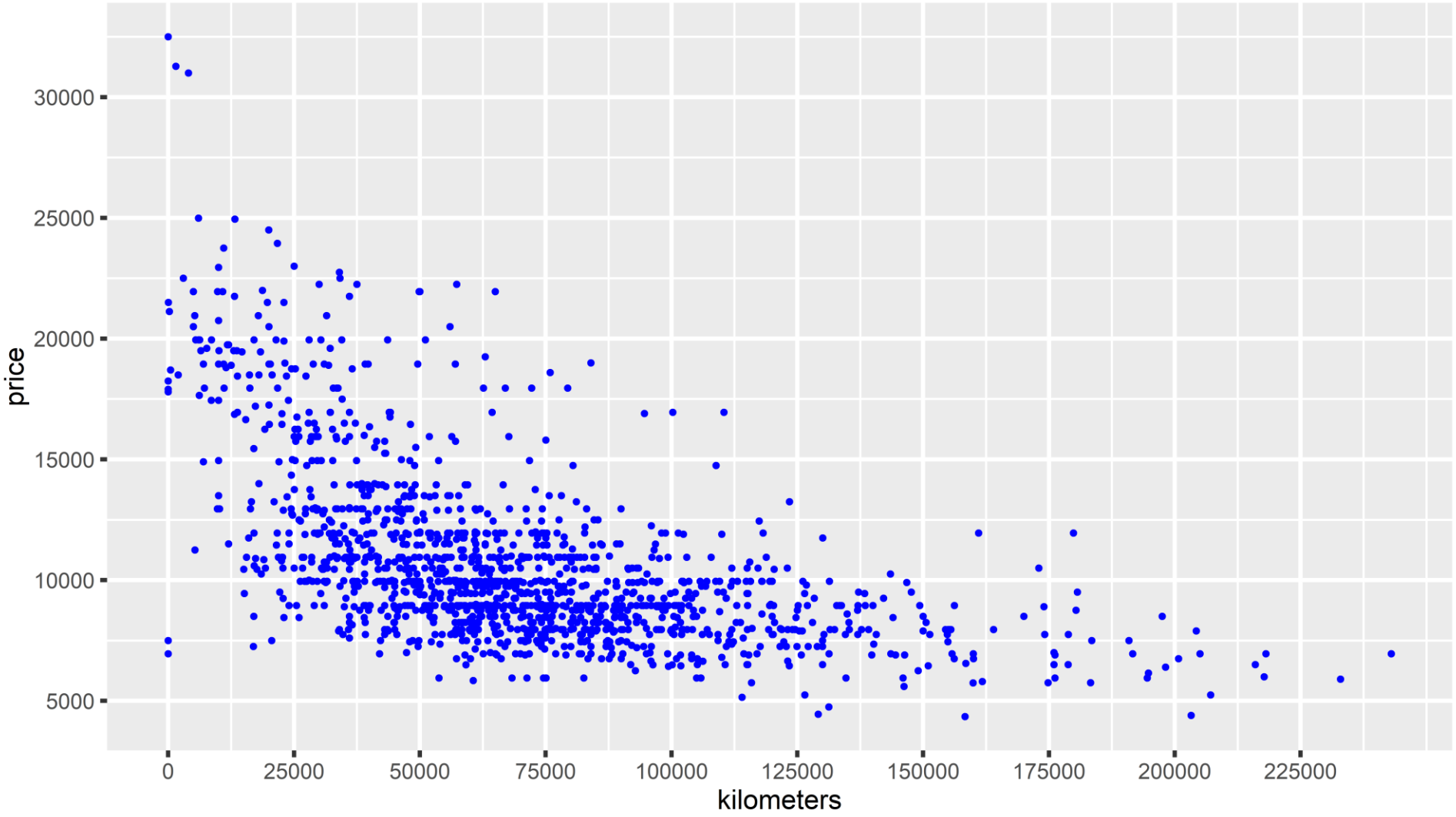


Two variables : Numeric X, Numeric Y

- **Scatter plot**
- Jitter plot
- Point plot
- Quantile plot
- Rug plot

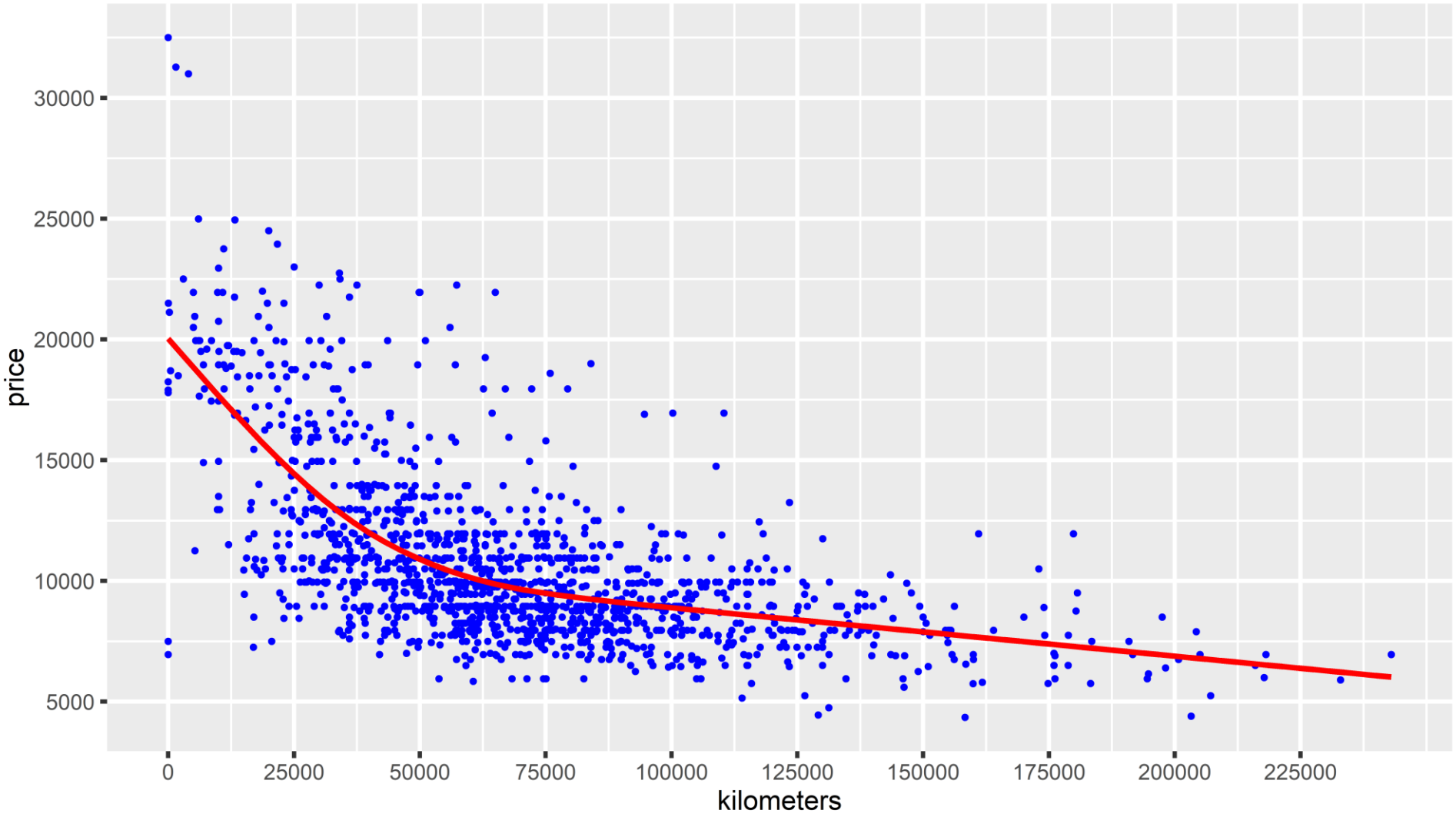
Two variables : Numeric X, Numeric Y

- Scatter plot between price and kilometers



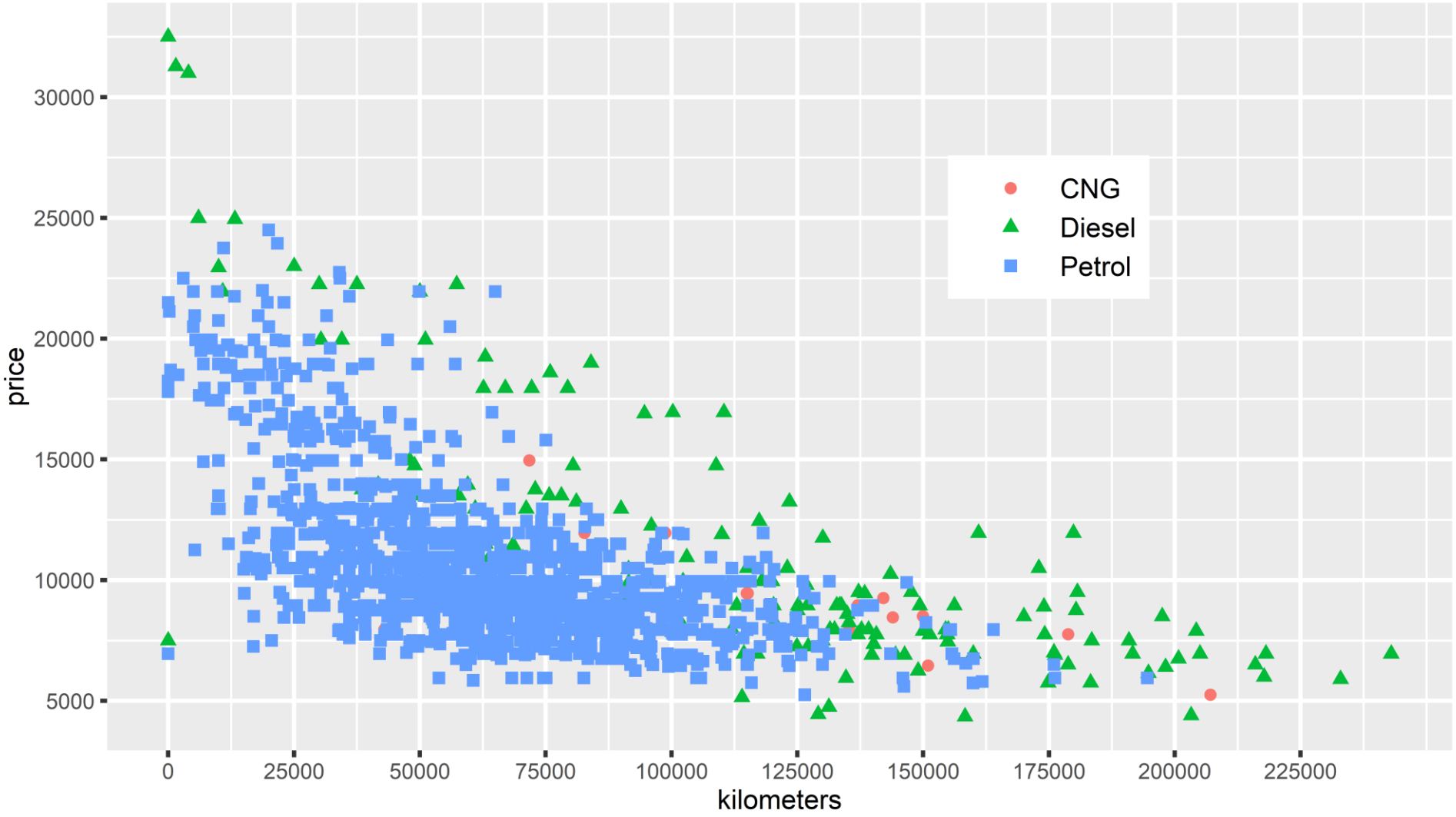
Two variables : Numeric X, Numeric Y

- Scatter plot between price and kilometers with a smooth line



Two variables : Numeric X, Numeric Y

- Scatter plot between price and kilometers by fuel_type

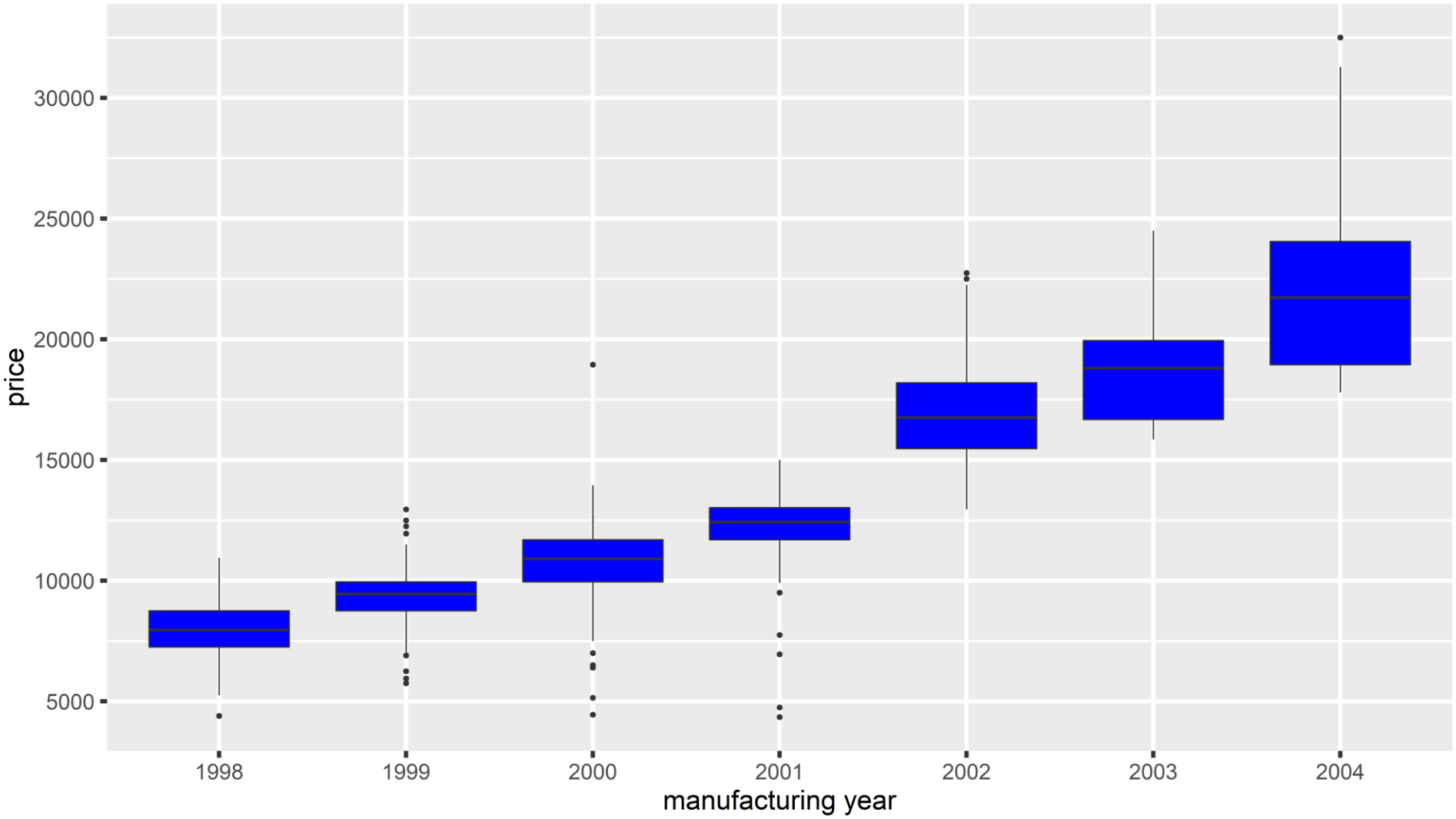


Two variables : Discrete X, Numeric Y

- **Box plot**
- Dot plot
- Violin plot

Two variables : Character X, Numeric Y

- Box plot between manufacturing year and price



More Graphics in ggplot2

- Bivariate distributions
- Visualizing error
- Zooming
- Maps
- Faceting
- Contours, Raster, Tile, Heat, Bubble etc.

Next Class

- *k*-Nearest Neighbor (*k*-NN) as Classification
- Application of *k*-NN in R/RStudio and Inference

Thank You