Ggplot2

Fabrice Rossi

CEREMADE
Université Paris Dauphine

2020

Outline

Introduction

Core principles

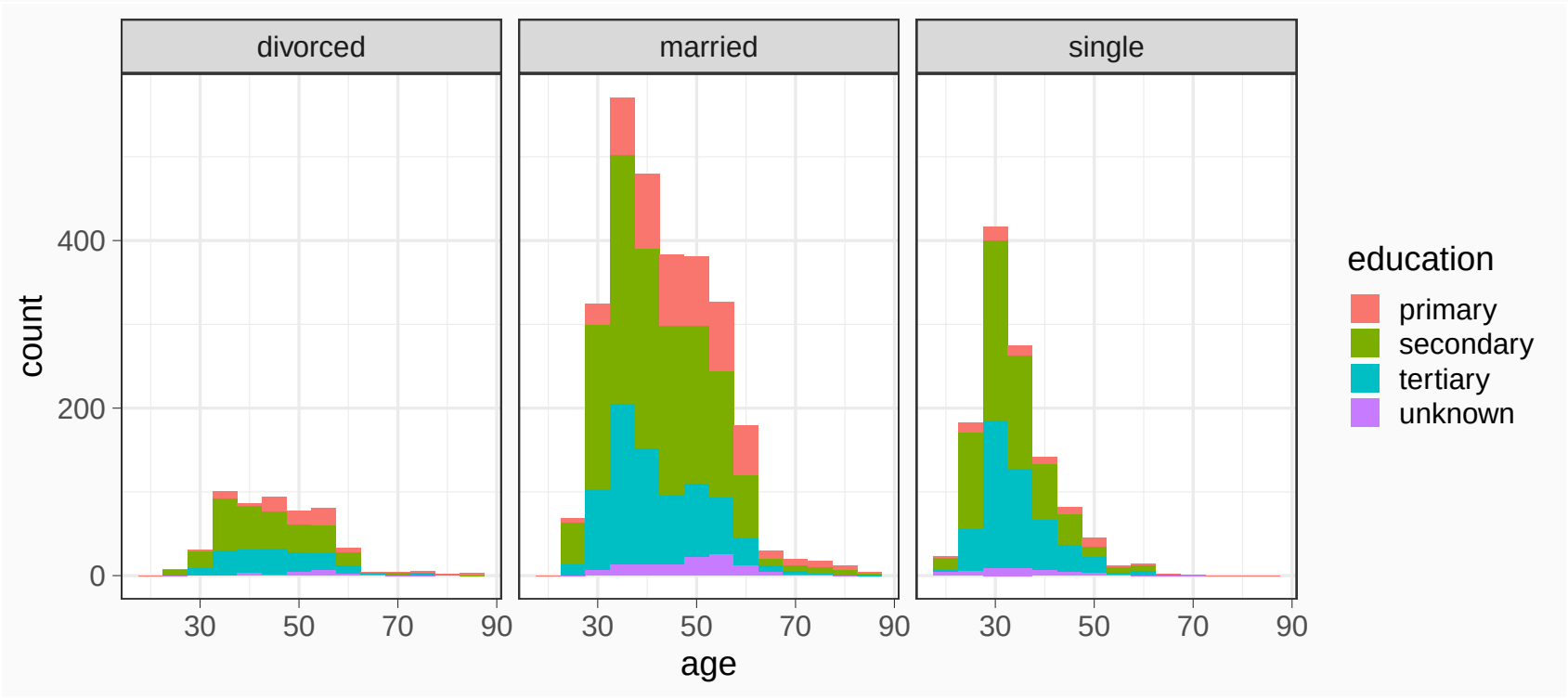
A tour of ggplot2

Extensions and other packages

Ggplot2

https://ggplot2.tidyverse.org/

- ggplot2 is a graphic system for R
- ggplot2 takes a declarative approach to graphics
- current standard for data science in R



Ggplot2

Pros

- explain what you want not how to do it (declarative)
- high quality defaults (e.g. for colors)
- easy conditional analysis
- consistent presentation
- included best practices

Cons

- rather steep learning curve (new logic compared to standard R plot)
- data science oriented (needs a data frame)
- difficult to customize in some circumstances
- no interactivity

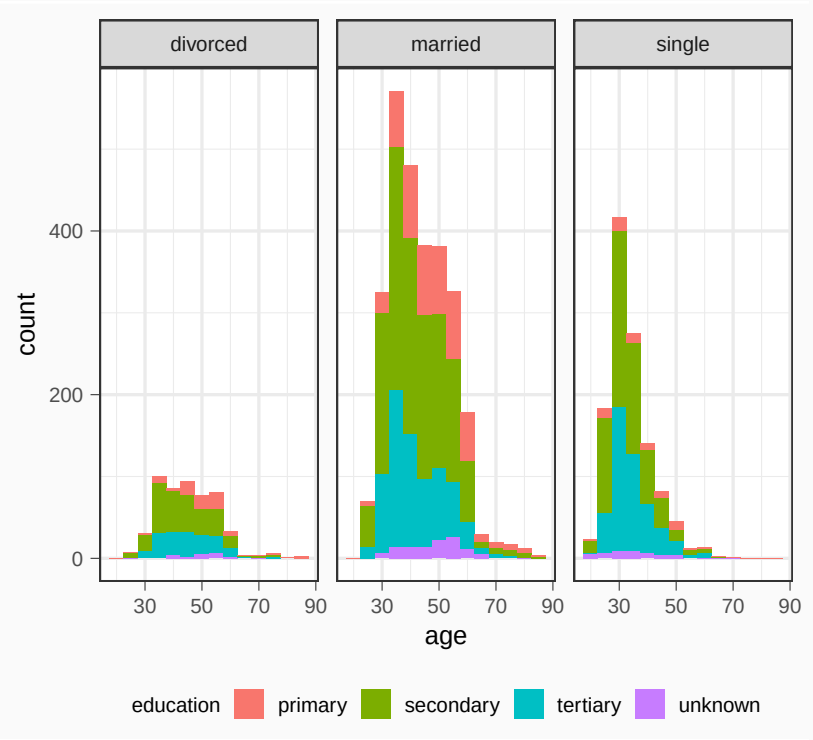
Principles

In ggplot, a plot is composed of

- a data set
- a mapping from some variables to *aesthetics* (graphical primitives)
- layers that compute summaries of the data (*stats*)
- layers that represent with graphical objects the data (*geom*)
- scales that transform values in the data space into values in the *aesthetics*
- a coordinate system
- a faceting specification for conditional analysis
- a *theme* which specifies details such as fonts and colormap

Example

- data: age, education level and marital status
- mappings:
  - x axis: age
  - fill color: education level
- geom: histogram
- facet: conditionally on marital status
- theme: lot of tweaking...



Outline

Data set

Bank marketing data set

- Bank+Marketing
- direct marketing (phone call)
- 17 variables
- target variable: success of the offer
- 3 groups of variables:
  - description of the client (age, education, etc.)
  - banking status (balance, loan, etc.)
  - previous contact with this client

Analysis

- marketing oriented questions
  - success/failure characterization
  - effect of past contacts
- more general questions
  - relationship between variables
  - characterization of the customers

Minimal graphic

Simple graphic in ggplot2

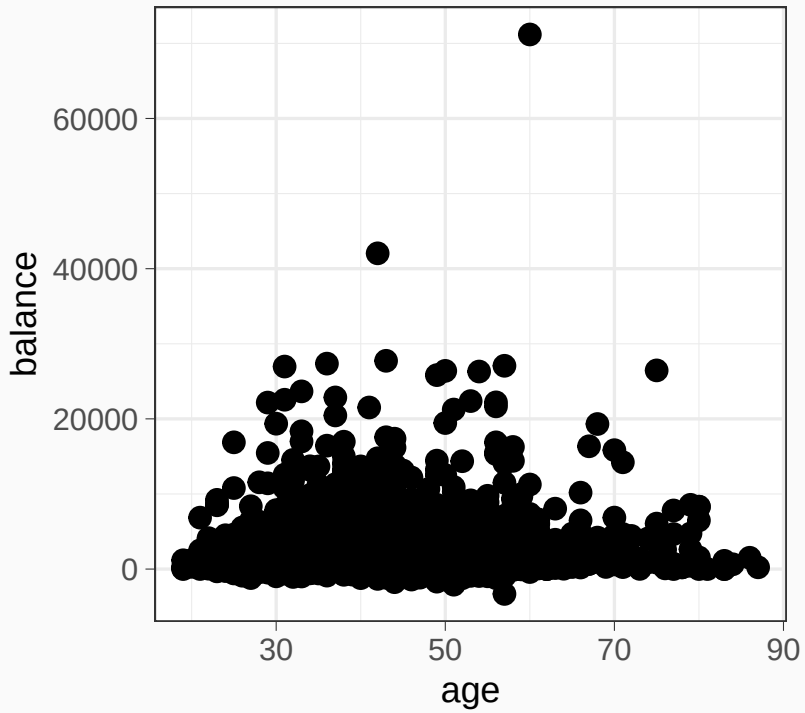
- a data set
- a mapping from the variables to the aesthetics
- a geom layer

Bank example

- data: `bank`
- mapping:
  - `age` to x axis
  - `balance` to y axis
- layer: points (scatter plot)

Example

```
ggplot(bank, aes(x = age,
    y = balance)) + geom_point()
```



General form

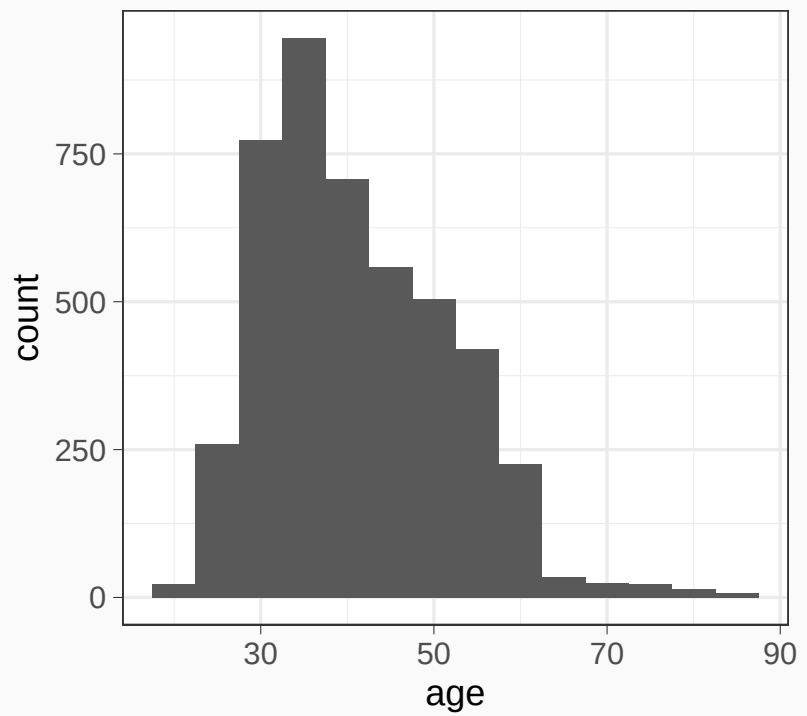
Typical form

```
ggplot(data, aes(mapping)) + layer
```

- (almost) always starts with `ggplot` to set the data and the mappings
- then followed by one or several calls to other ggplot2 functions
- at least one `geom_something` layer to draw someting

Example

```
ggplot(bank, aes(x = age)) +
    geom_histogram(binwidth = 5)
```



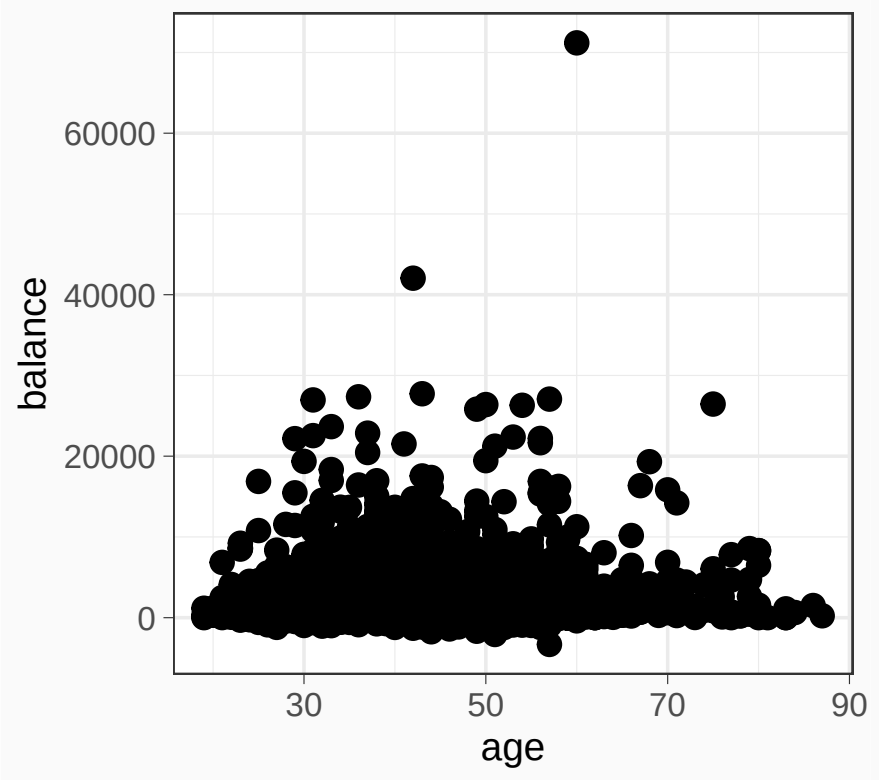
Parameters

Geom layers

- each `geom_something` has specific parameters
- reasonable defaults in many cases
- aesthetics can be specified as parameters

Example

```
ggplot(bank, aes(age,balance)) +
    geom_point()
```



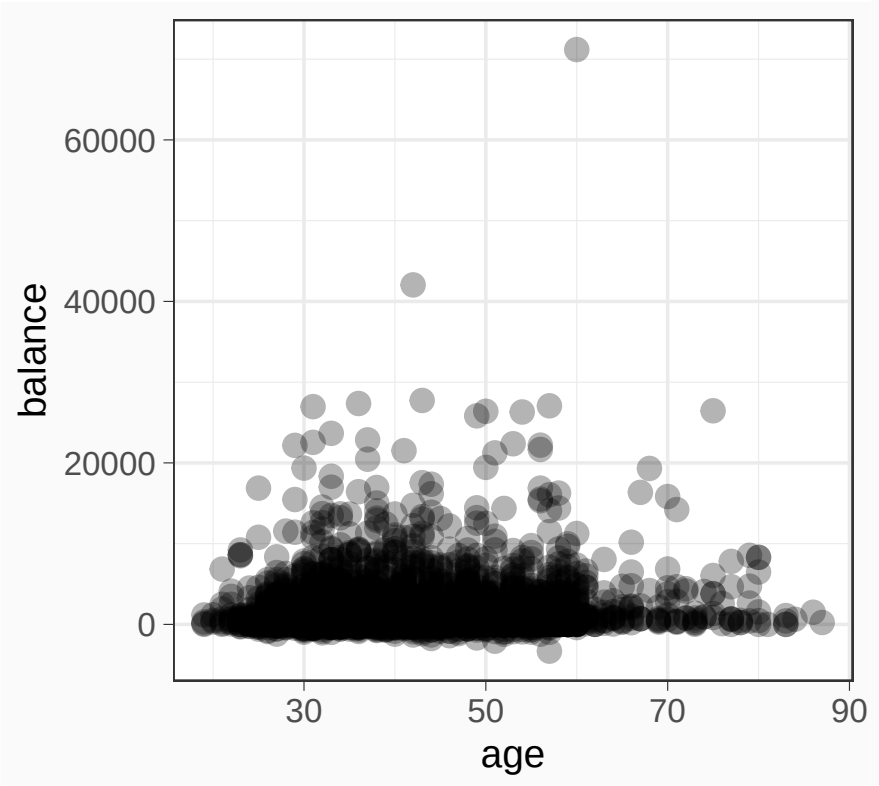
Parameters

Geom layers

- each `geom_something` has specific parameters
- reasonable defaults in many cases
- aesthetics can be specified as parameters

Example

```
ggplot(bank, aes(age,balance)) +
    geom_point(alpha=0.3)
```



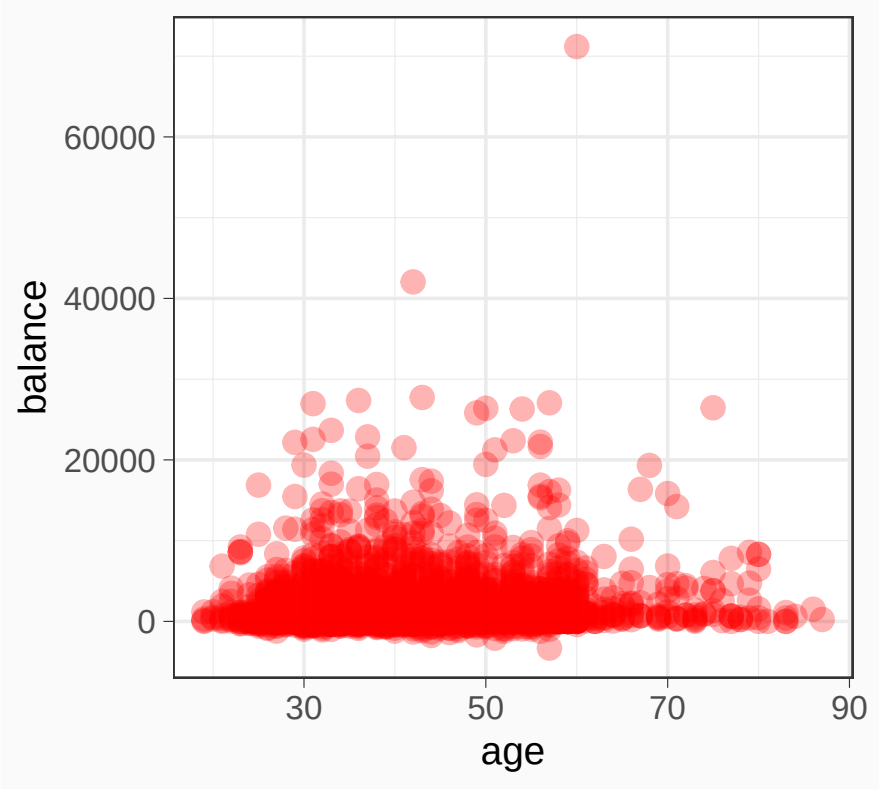
Parameters

Geom layers

- each `geom_something` has specific parameters
- reasonable defaults in many cases
- aesthetics can be specified as parameters

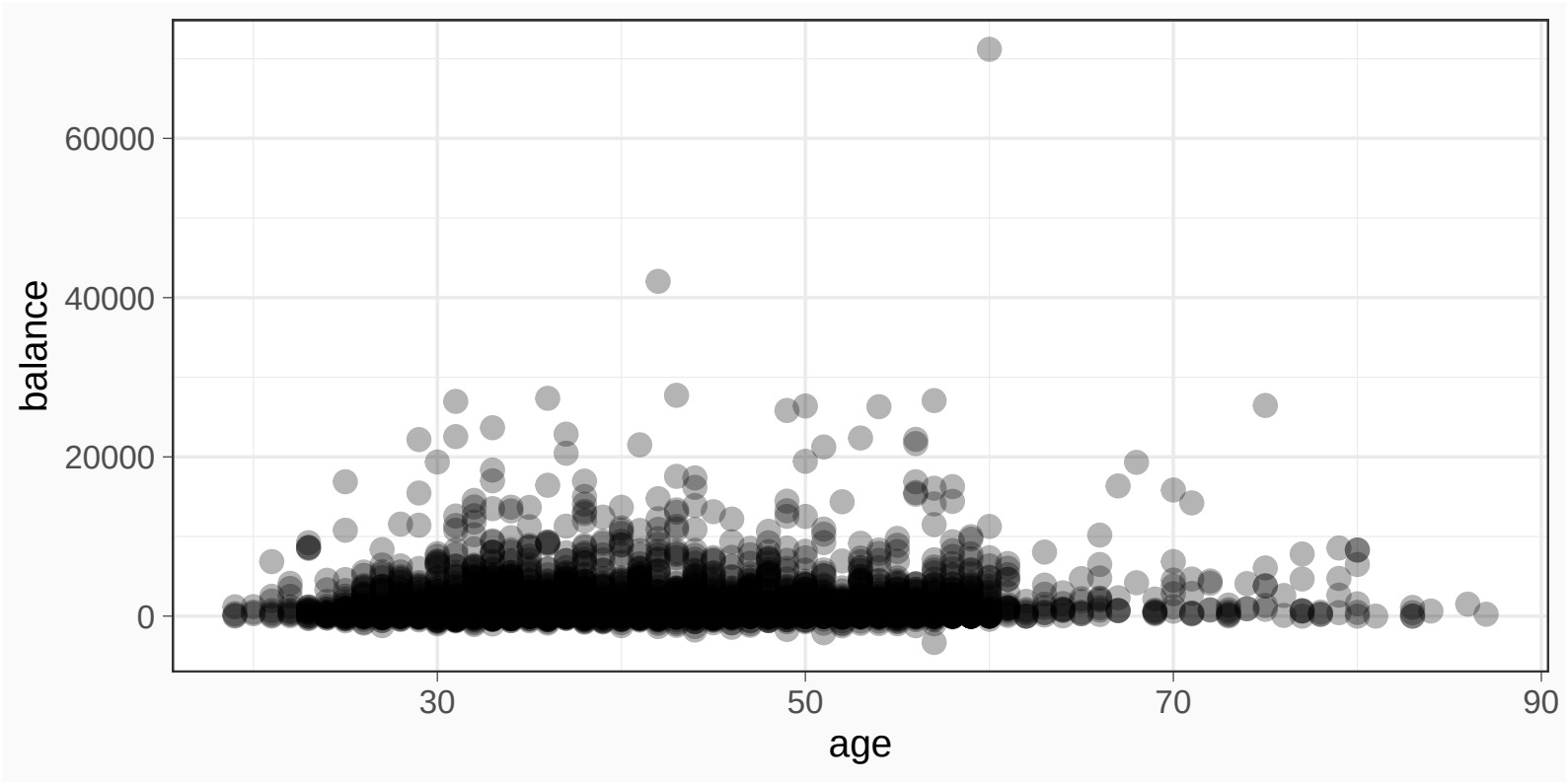
Example

```
ggplot(bank, aes(age,balance)) +
    geom_point(alpha=0.3,color="red")
```



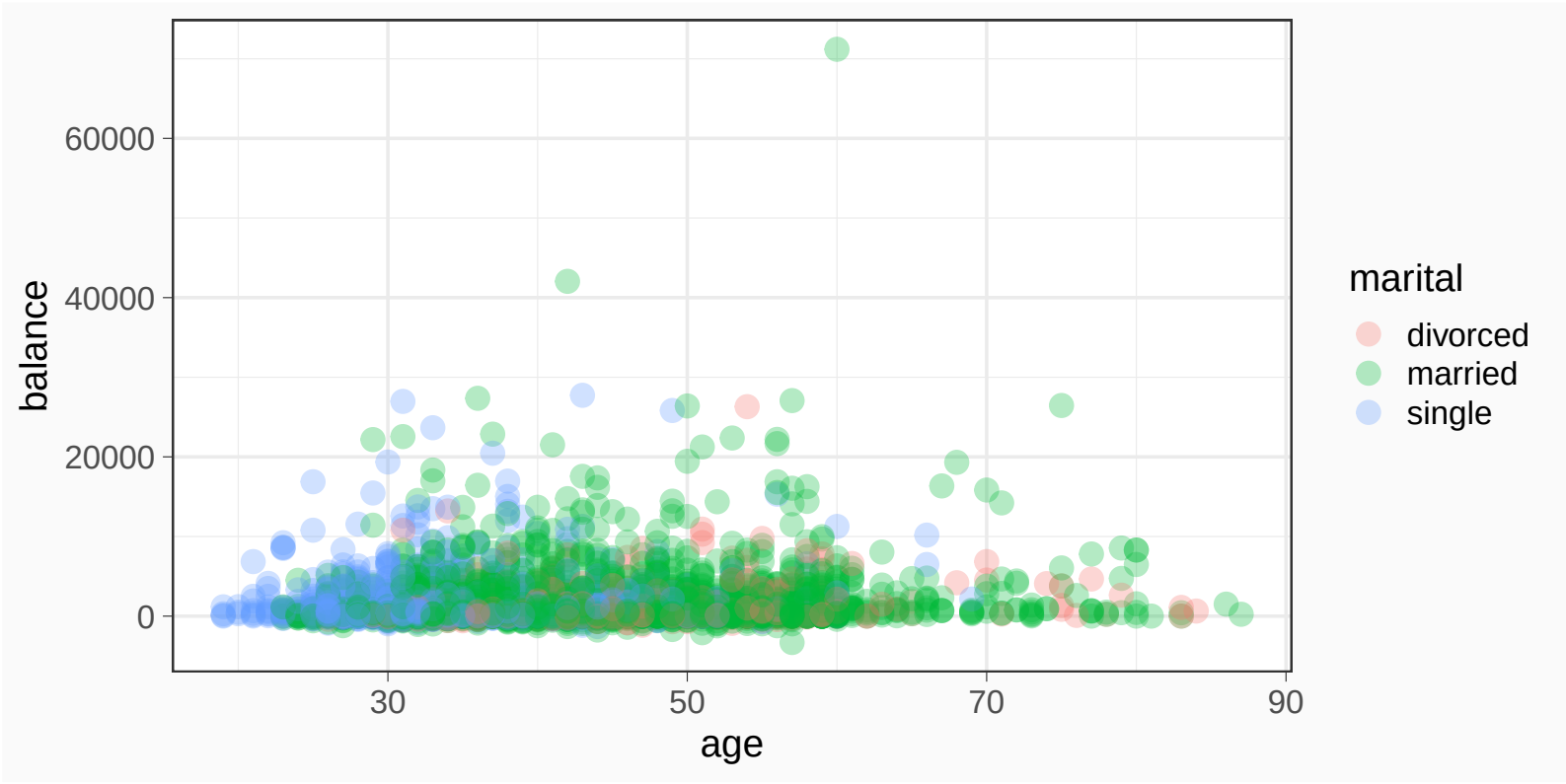
Mapping many variables

```
ggplot(bank, aes(x=age, y=balance)) +
    geom_point(alpha=0.3)
```



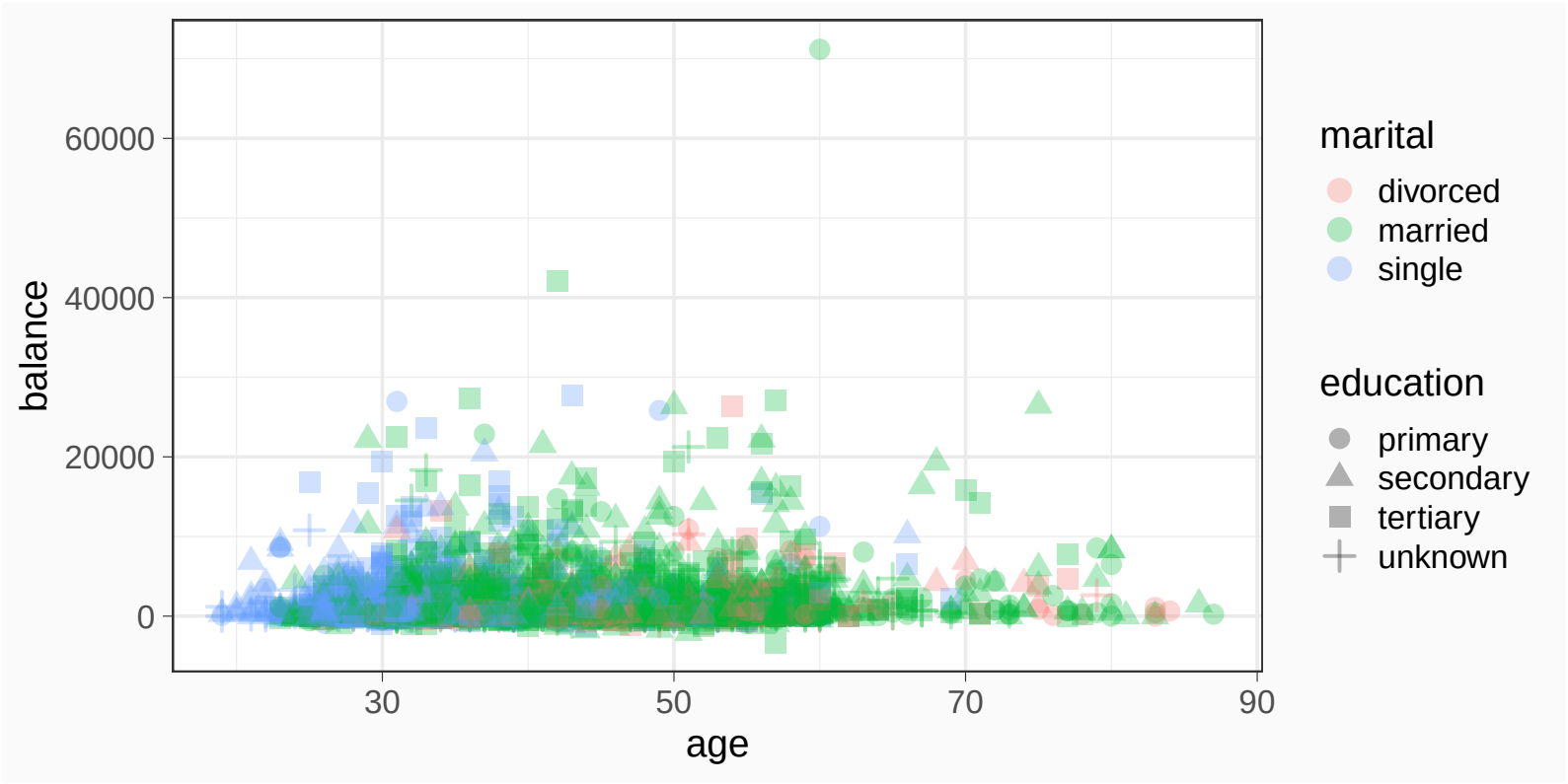
Mapping many variables

```
ggplot(bank, aes(x=age, y=balance, color=marital)) +
    geom_point(alpha=0.3)
```



Mapping many variables

```
ggplot(bank, aes(x=age, y=balance, color=marital, shape=education)) +
    geom_point(alpha=0.3)
```



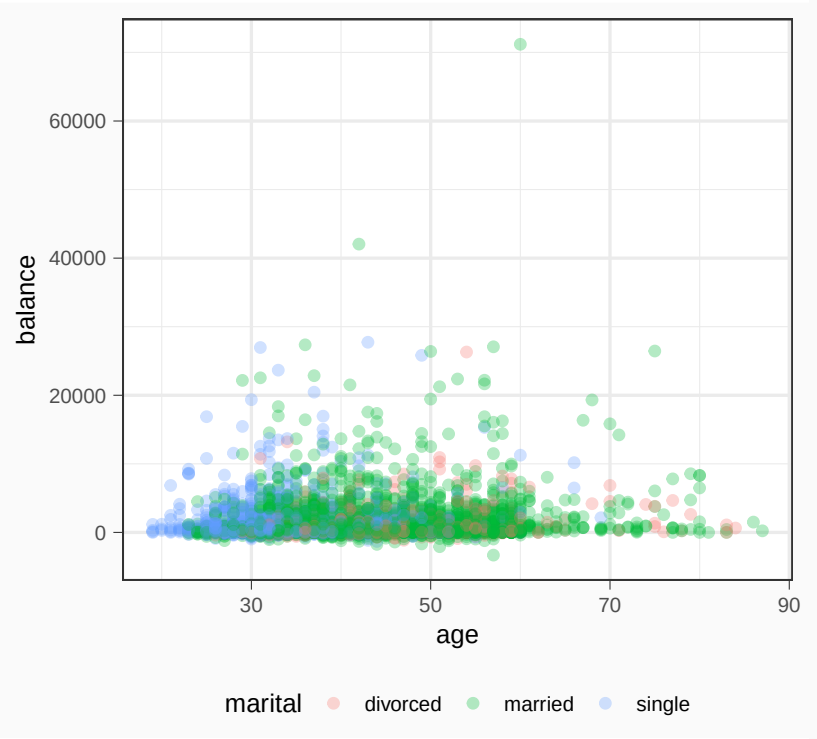
Facetting

Aesthetic limitations

- only a small number of graphical attributes can be used in a given graphic
- encoding quality differs from one attribute to another
- visual clutter

Small multiples

- combine several graphs
- keep each graph simple



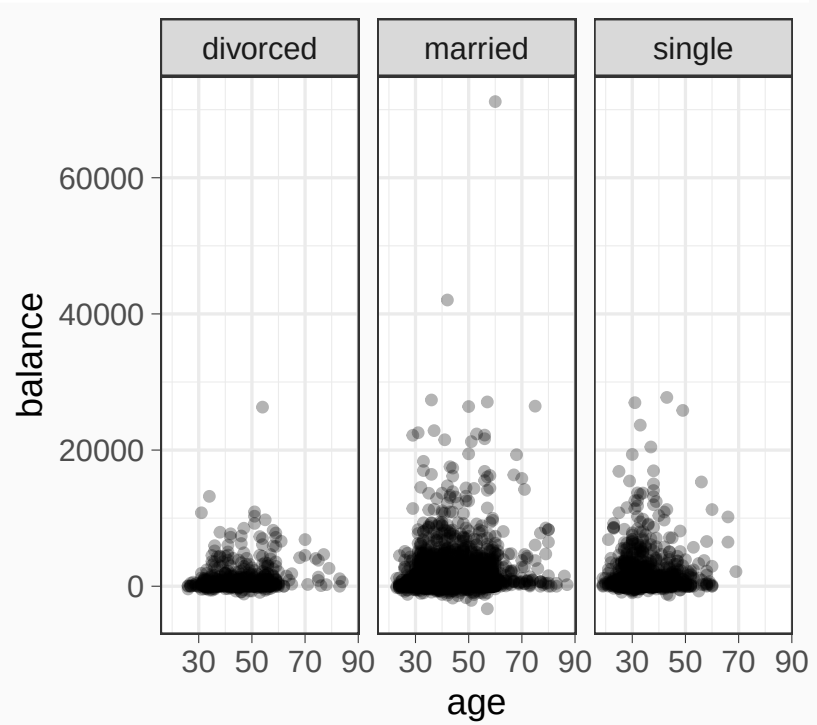
Facetting

Aesthetic limitations

- only a small number of graphical attributes can be used in a given graphic
- encoding quality differs from one attribute to another
- visual clutter

Small multiples

- combine several graphs
- keep each graph simple




```
ggplot(bank, aes(x=age, y=balance)) +
    geom_point(alpha=0.3,size=0.5) +
        facet_wrap(~marital)
```

Facetting

`facet_wrap`

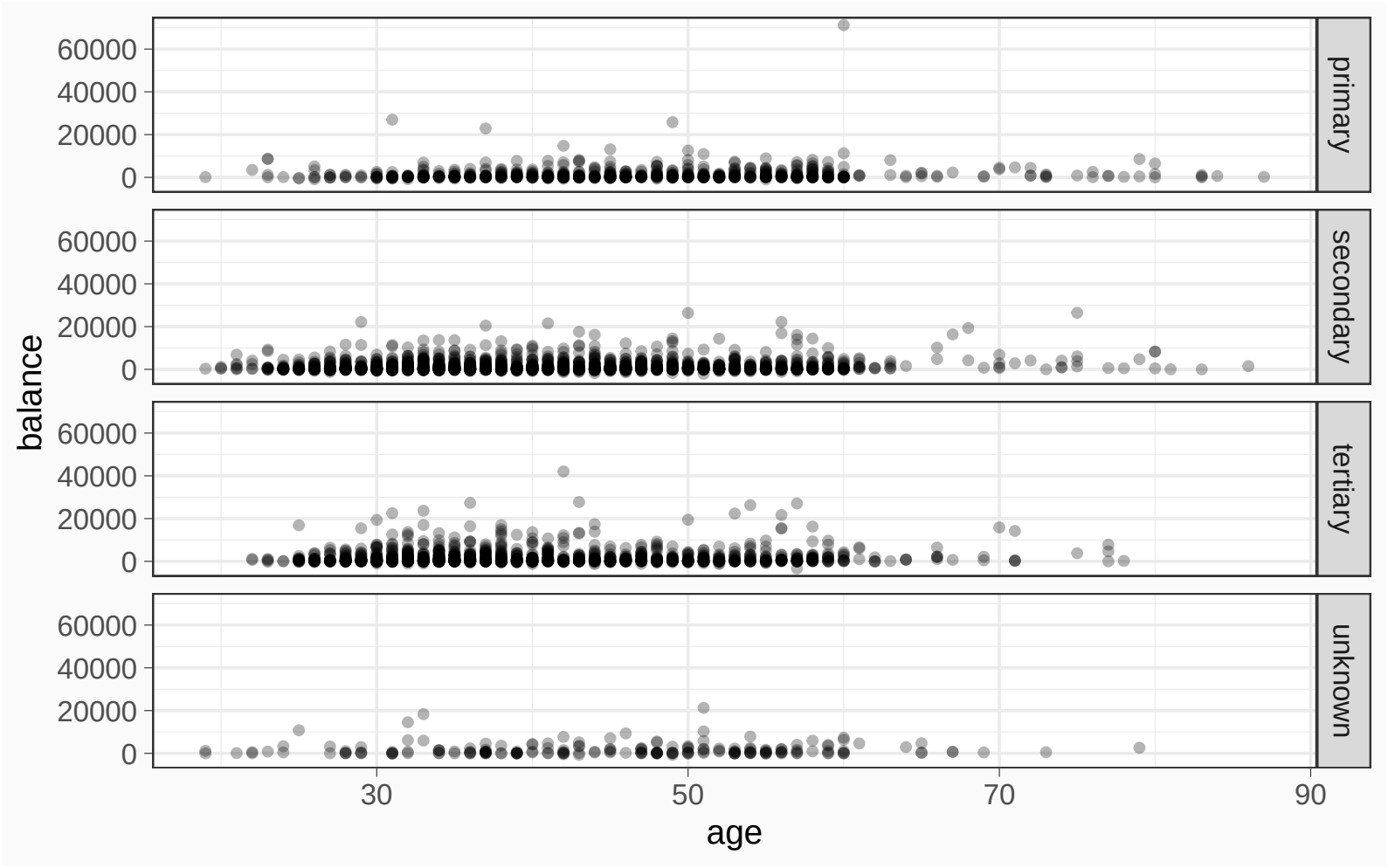
- conditioning on a single variable
- `facet_warp(~variable)`
- one panel per value of `variable`

`facet_grid`

- conditioning on one or two variables
- `facet_grid(.~x)`: horizontal
- `facet_grid(y~.)`: vertical
- `facet_grid(y~x)`: as a grid

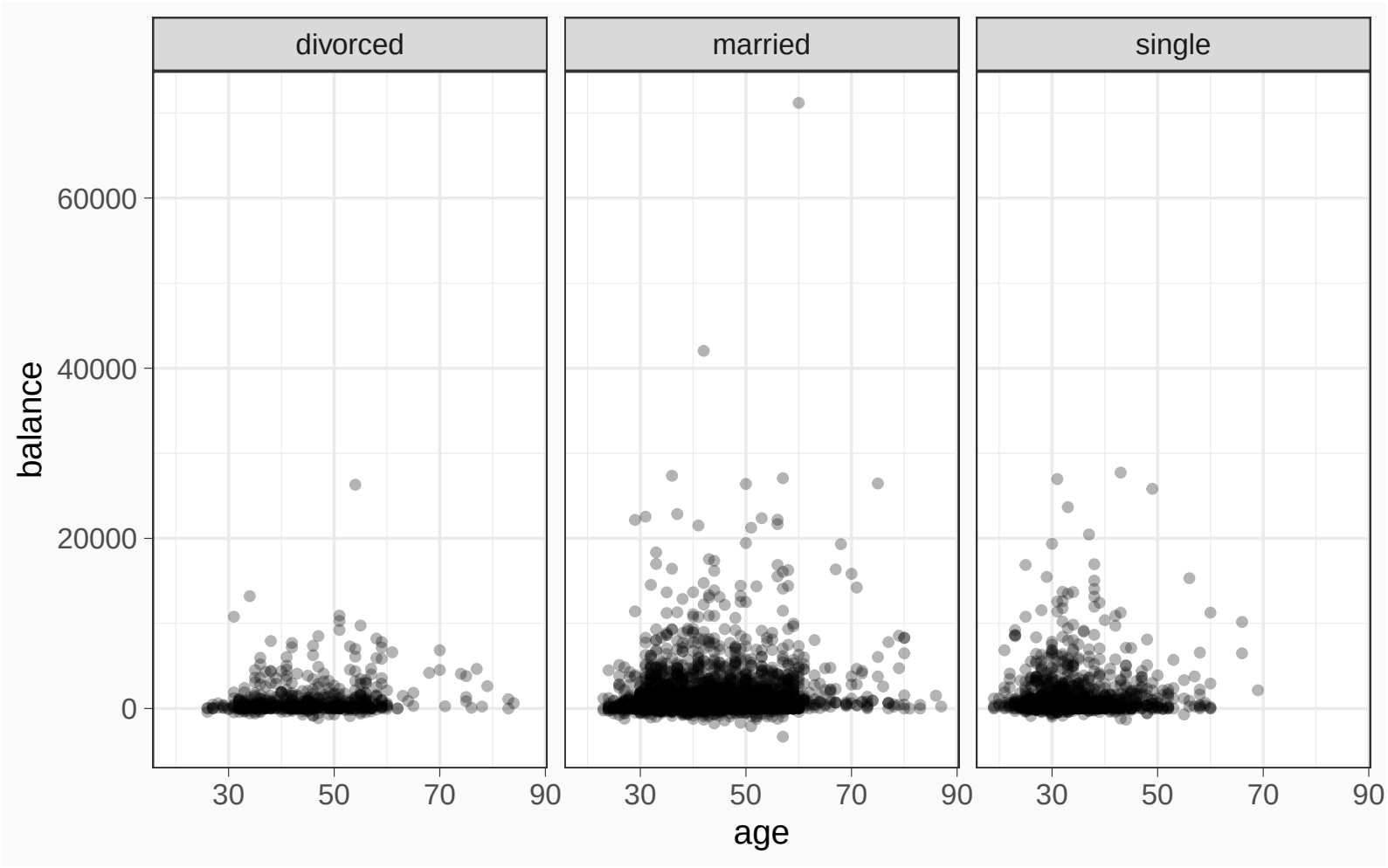
Example

```
ggplot(bank, aes(age, balance)) + geom_point(alpha = 0.3,
    size = 0.5) + facet_grid(education ~ .)
```



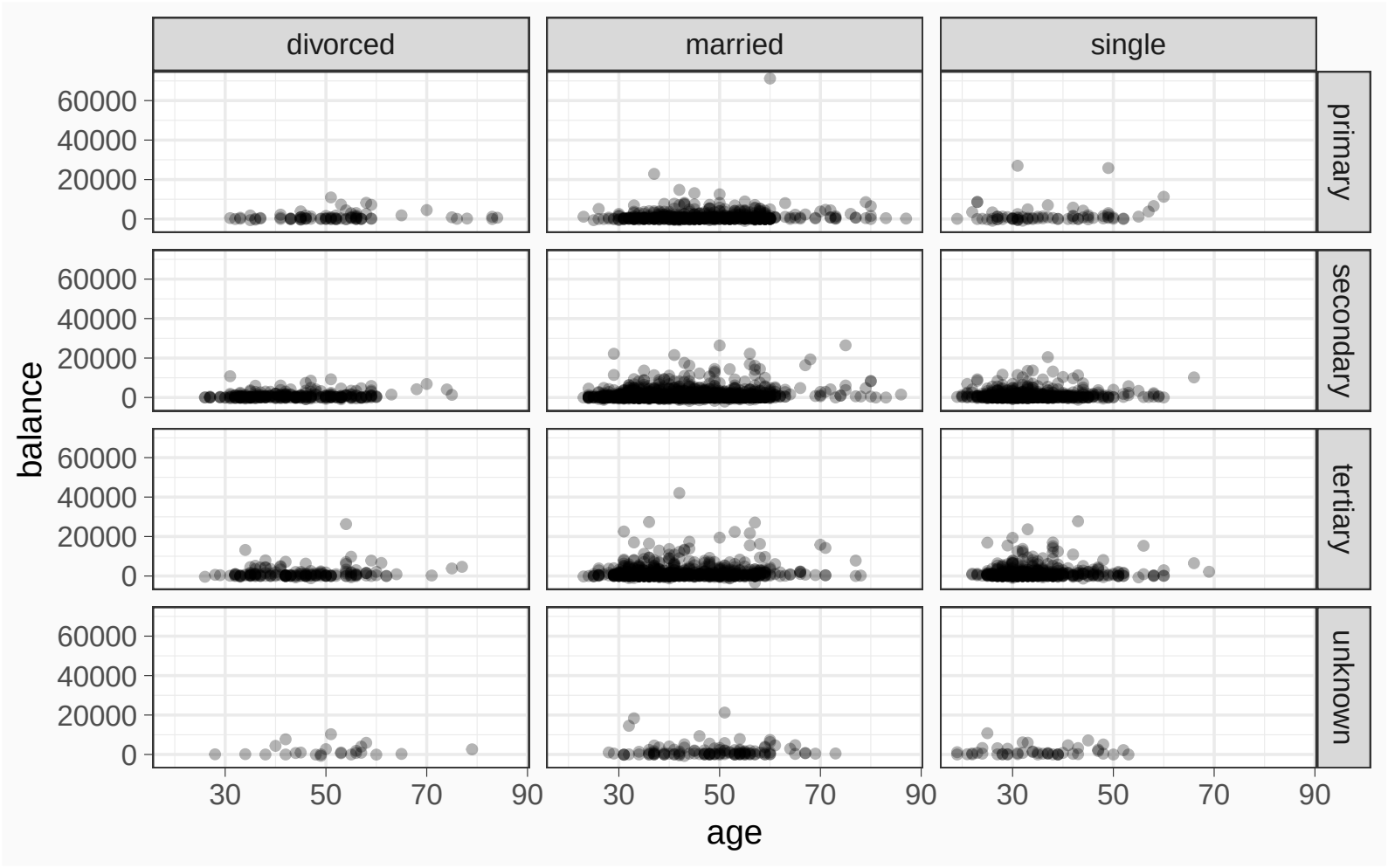
Example

```
ggplot(bank, aes(age, balance)) + geom_point(alpha = 0.3,
    size = 0.5) + facet_grid(~marital)
```



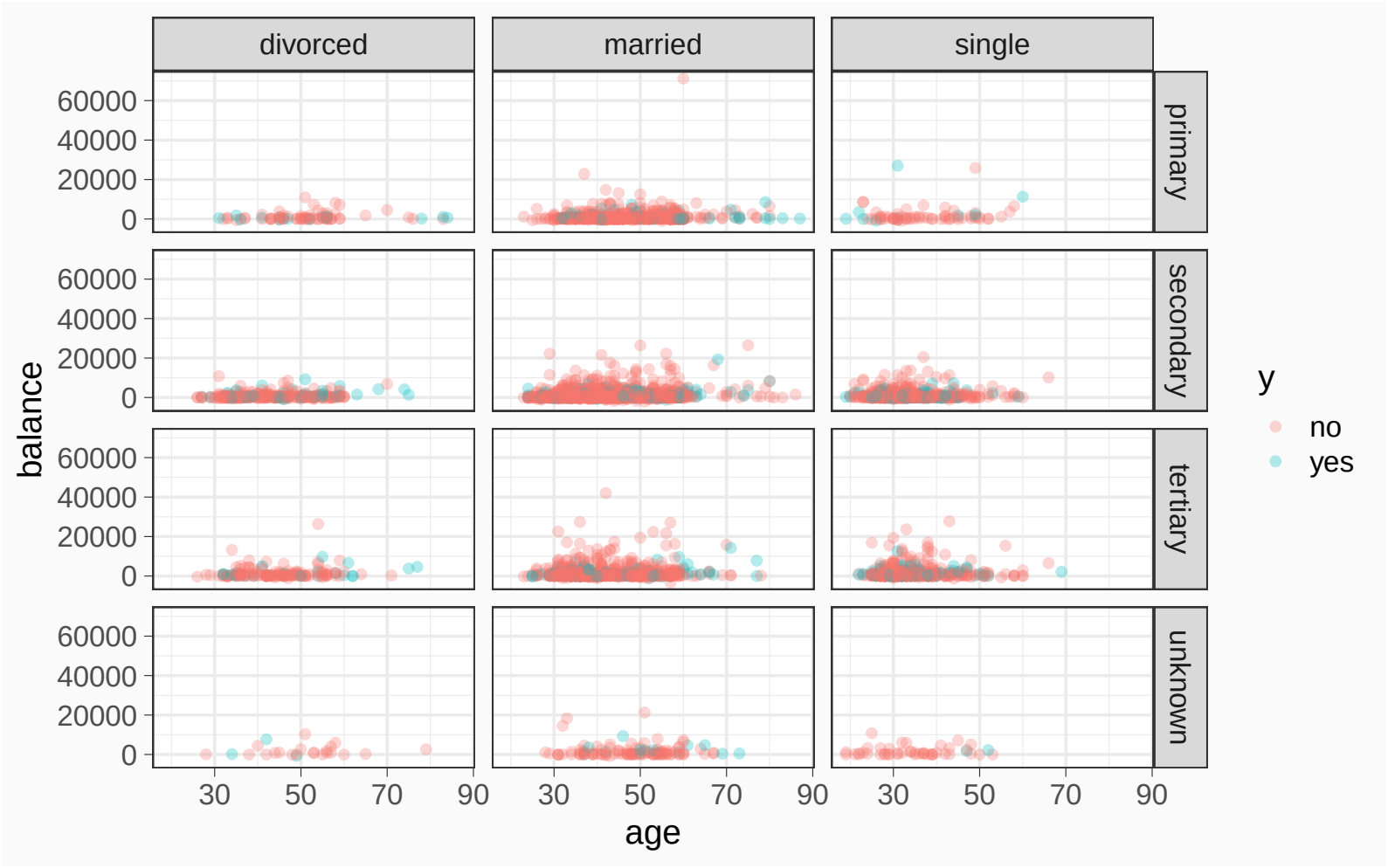
Example

```
ggplot(bank, aes(age, balance)) + geom_point(alpha = 0.3,
    size = 0.5) + facet_grid(education ~ marital)
```



Example

```
ggplot(bank, aes(age, balance, color = y)) + geom_point(alpha = 0.3,
    size = 0.5) + facet_grid(education ~ marital)
```



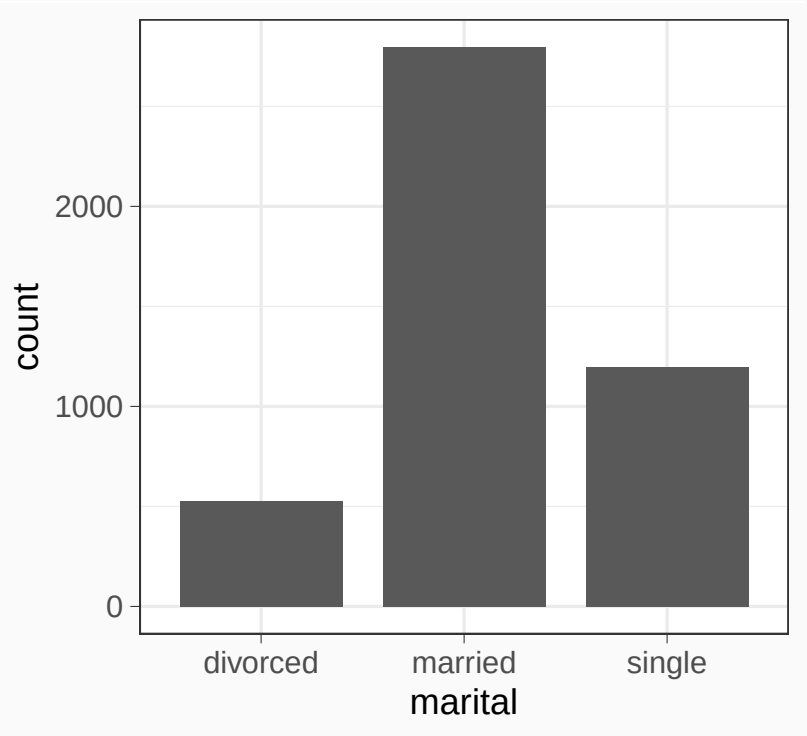
Outline

Bar Graph

Discrete variable distribution

- always use a bar graph (a.k.a. bar plot)
- never a pie chart
- see e.g. Save the Pies of Dessert by Stephen Few

```
ggplot(bank, aes(x = marital)) +
    geom_bar(width = 0.8)
```



`geom_bar`

- `x` aesthetics: the discrete variable
- `fill` aesthetics: color of the bars
- `width` parameter: width of the bars

Bar Graph

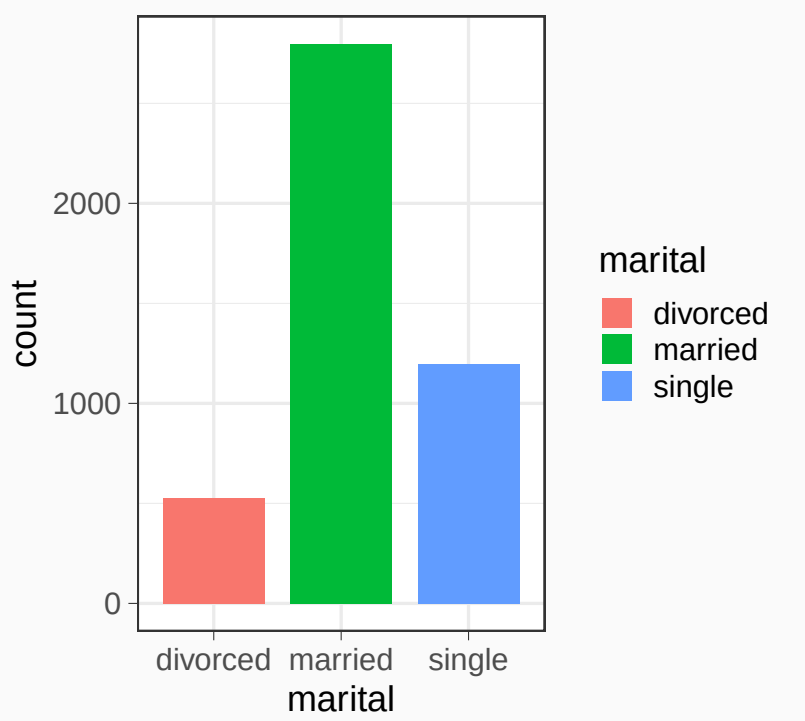
Discrete variable distribution

- always use a bar graph (a.k.a. bar plot)
- never a pie chart
- see e.g. Save the Pies of Dessert by Stephen Few

`geom_bar`

- `x` aesthetics: the discrete variable
- `fill` aesthetics: color of the bars
- `width` parameter: width of the bars

```
ggplot(bank, aes(x = marital,
    fill = marital)) +
    geom_bar(width = 0.8)
```



Bar Graph

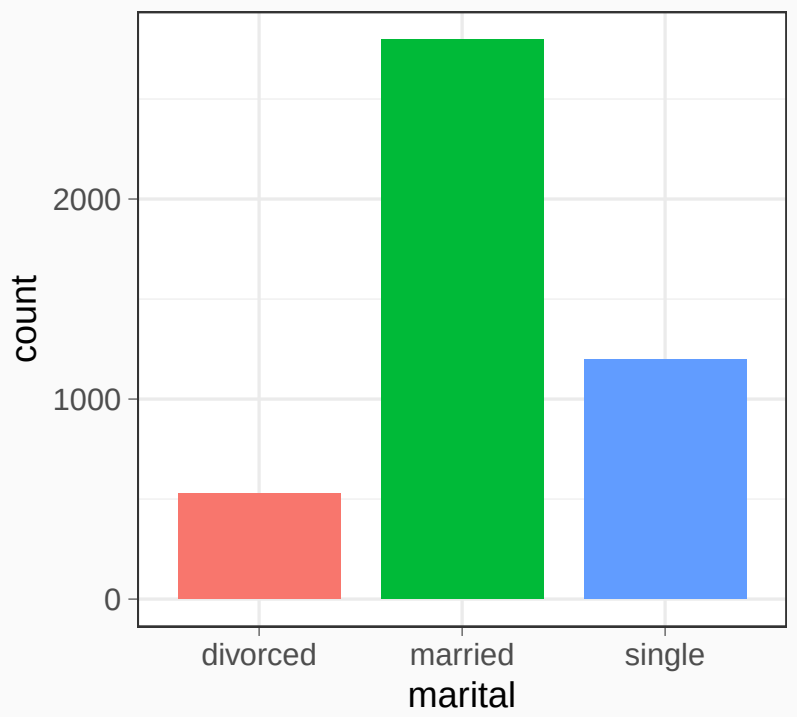
Discrete variable distribution

- always use a bar graph (a.k.a. bar plot)
- never a pie chart
- see e.g. Save the Pies of Dessert by Stephen Few

`geom_bar`

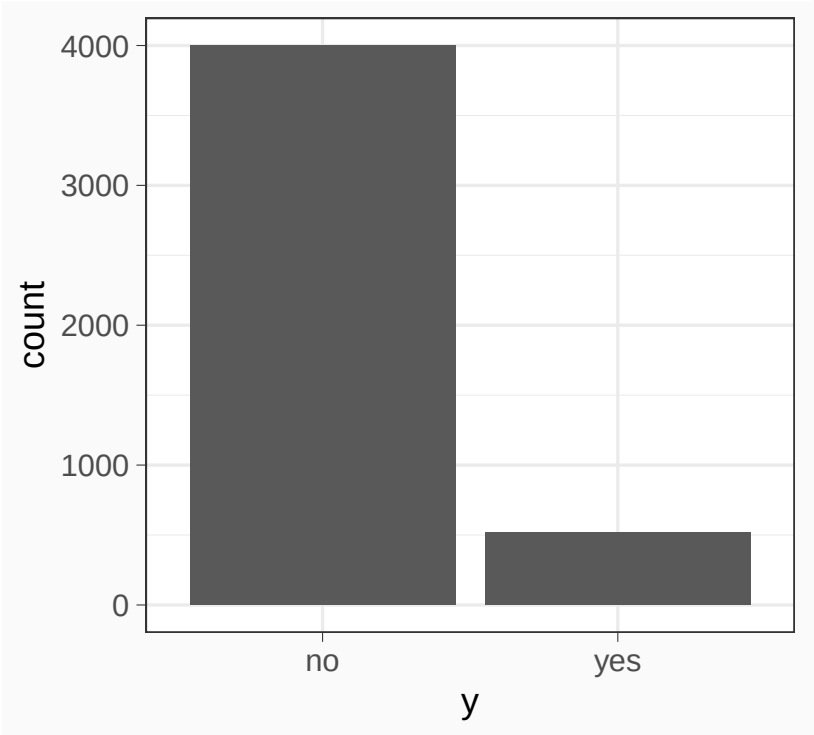
- `x` aesthetics: the discrete variable
- `fill` aesthetics: color of the bars
- `width` parameter: width of the bars

```
ggplot(bank, aes(x = marital,
    fill = marital)) +
    geom_bar(width = 0.8) +
    theme(legend.position = "none")
```

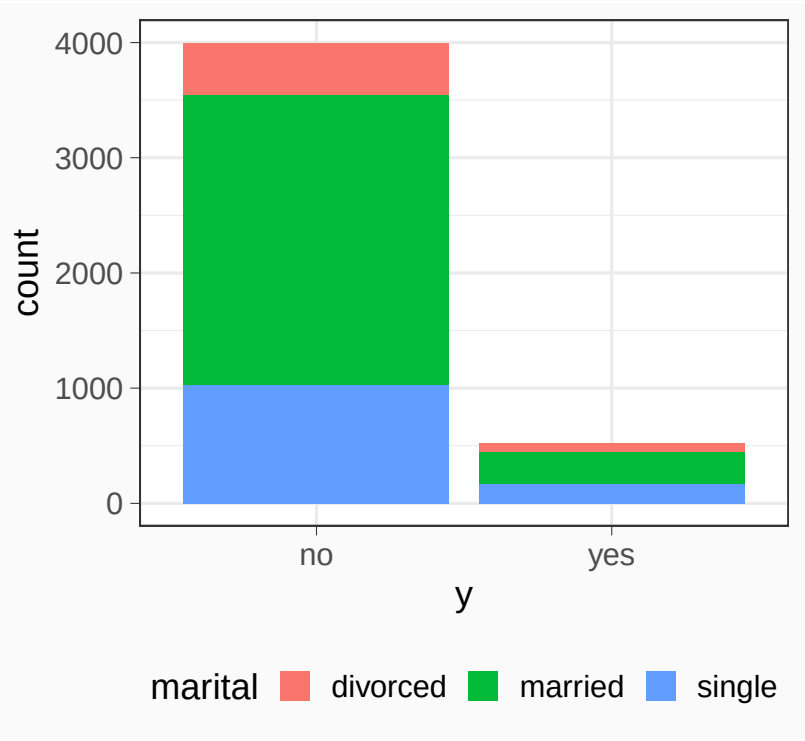


Bank Data

```
ggplot(bank, aes(x = y)) +
    geom_bar()
```

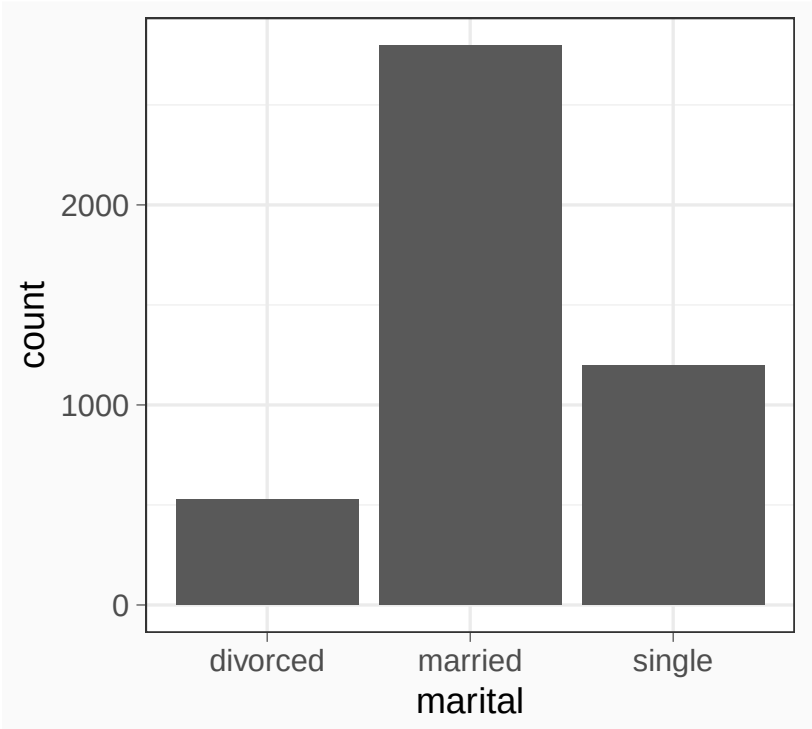



```
ggplot(bank,aes(x=y, fill=marital)) +
    geom_bar() +
    theme(legend.position="bottom")
```

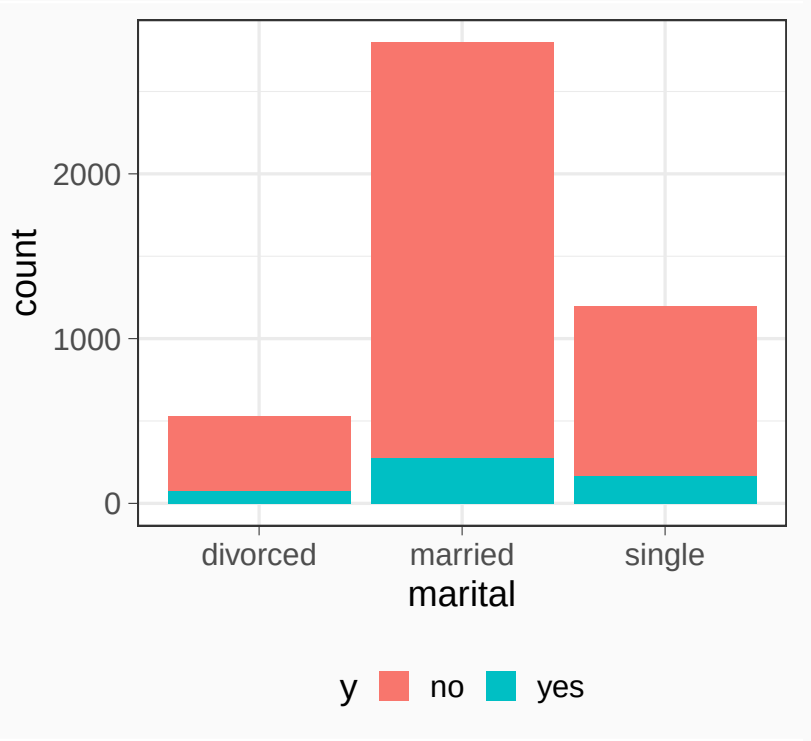


Bank Data

```
ggplot(bank, aes(x = marital)) +
    geom_bar()
```

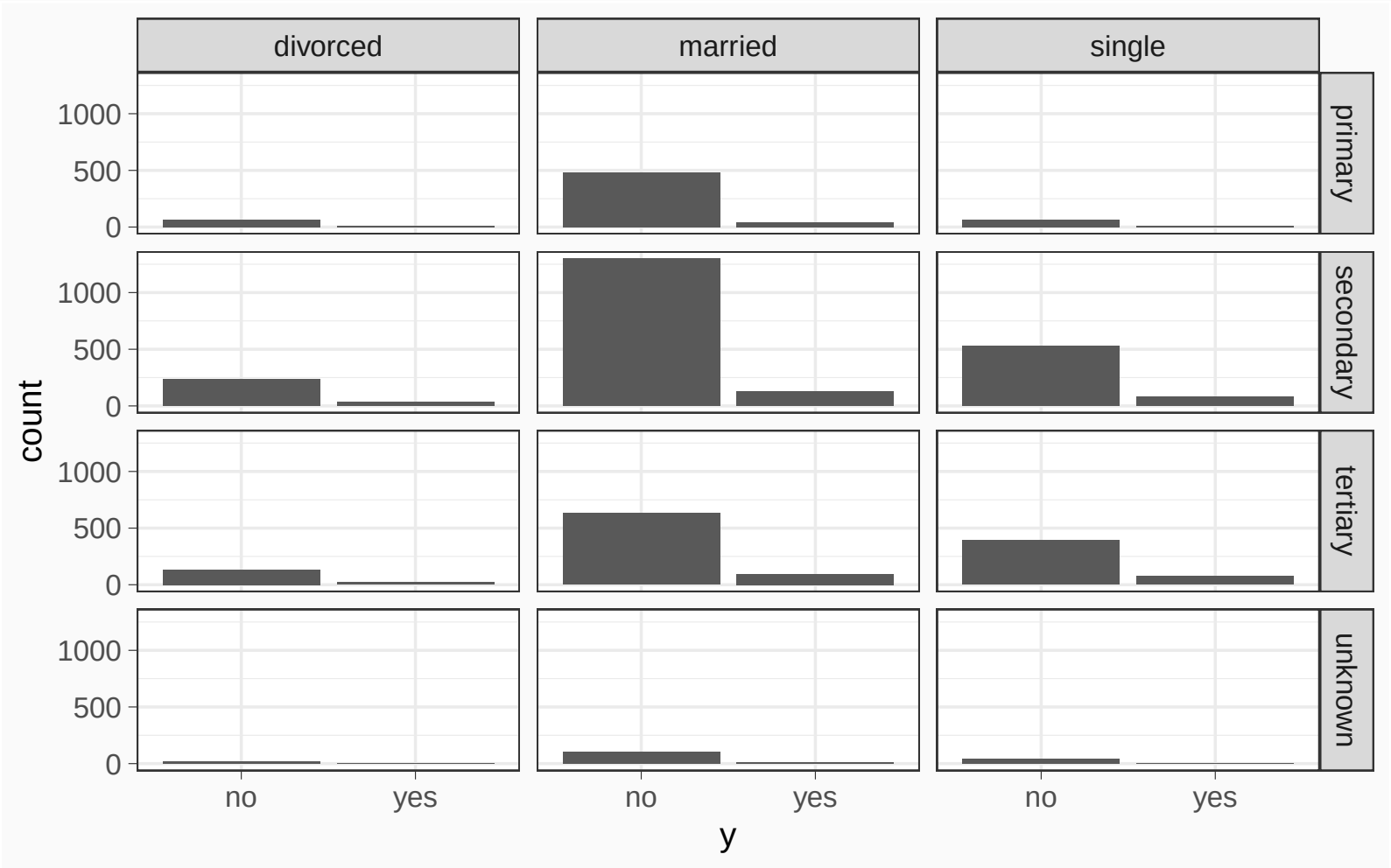



```
ggplot(bank,aes(x=marital, fill=y)) +
    geom_bar() +
    theme(legend.position="bottom")
```



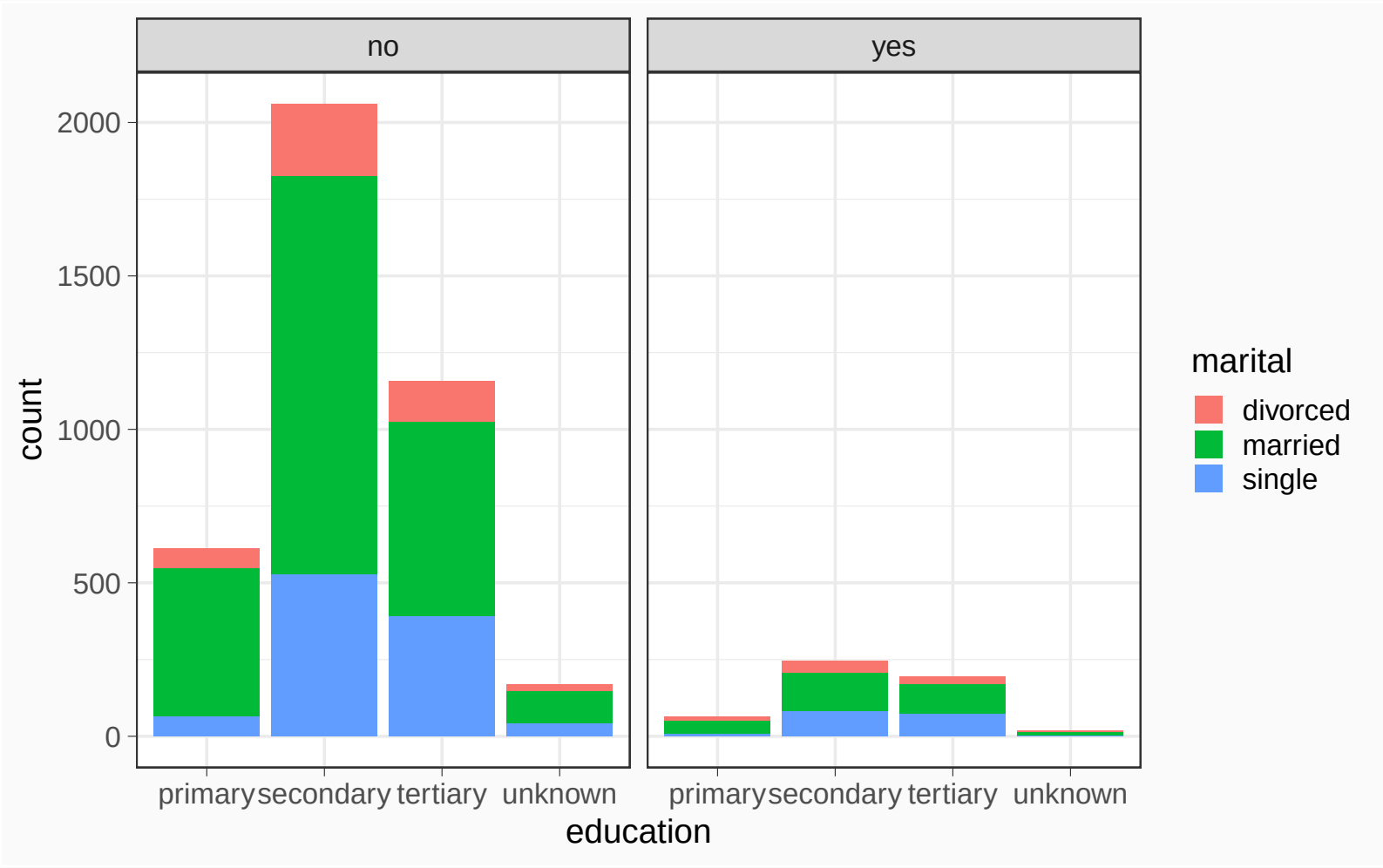
Bank Data

```
ggplot(bank, aes(x = y)) +
    geom_bar() + facet_grid(education ~
    marital)
```



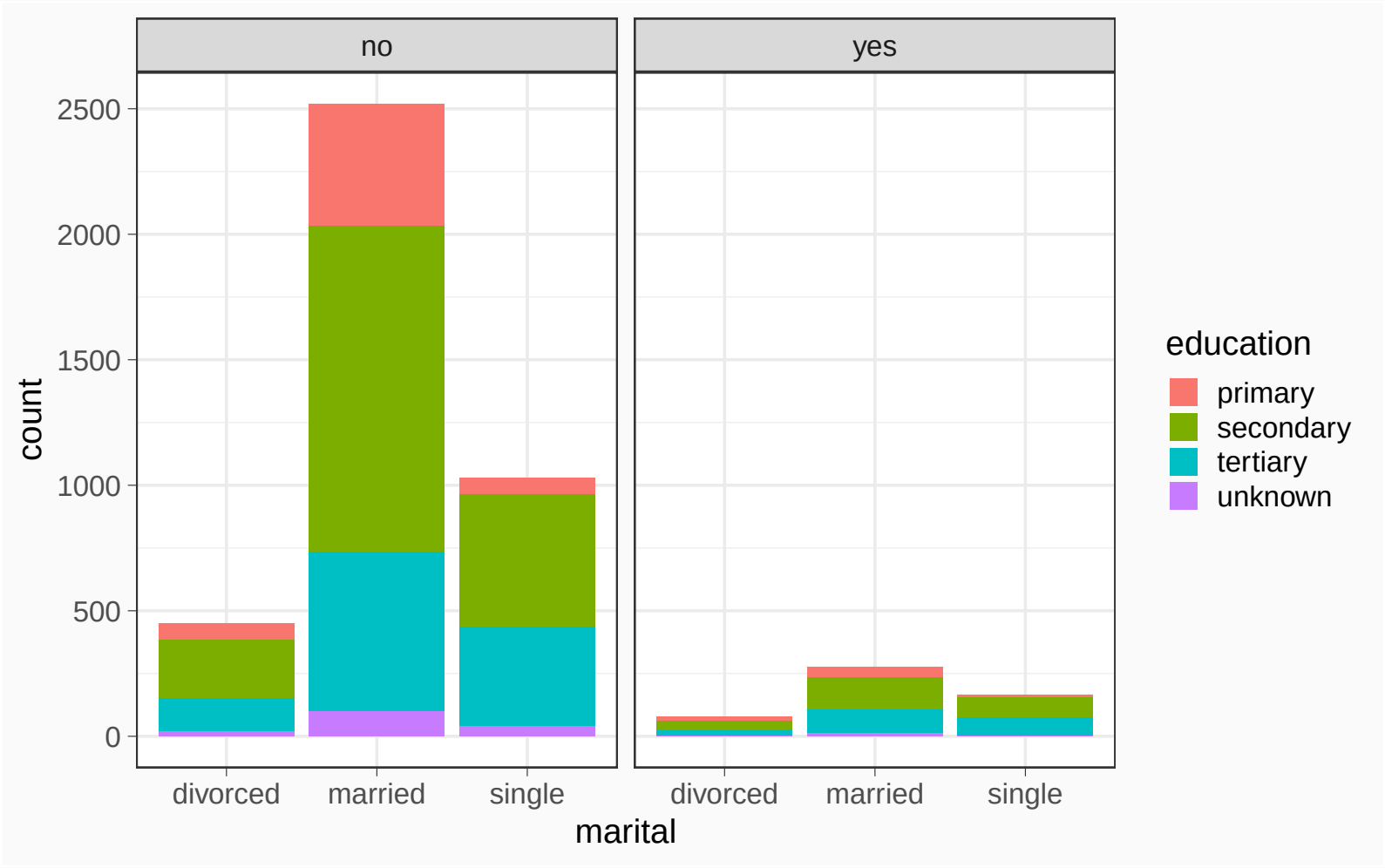
Bank Data

```
ggplot(bank, aes(x = education,
    fill = marital)) +
    geom_bar() + facet_wrap(~y)
```



Bank Data

```
ggplot(bank, aes(x = marital,
    fill = education)) +
    geom_bar() + facet_wrap(~y)
```



Variations on fill

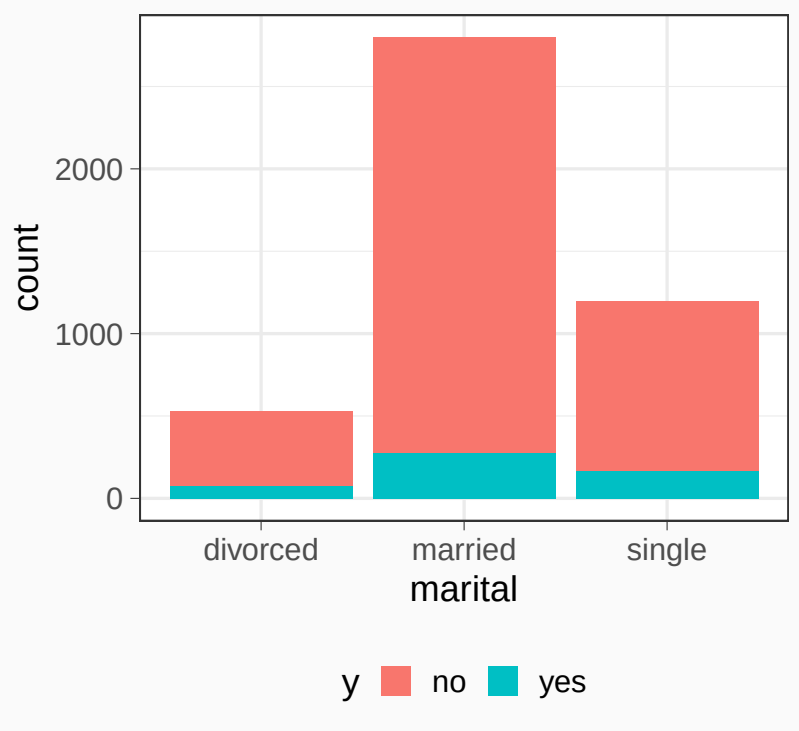
Stacking behavior

- several values for a given `x` induced by `fill`
- stacked by default

Alternatives

- side by side `position="dodge"`
- stacked with normalization `position="fill"`

```
ggplot(bank,aes(x=marital, fill=y)) +
    geom_bar() +
    theme(legend.position="bottom")
```



Variations on fill

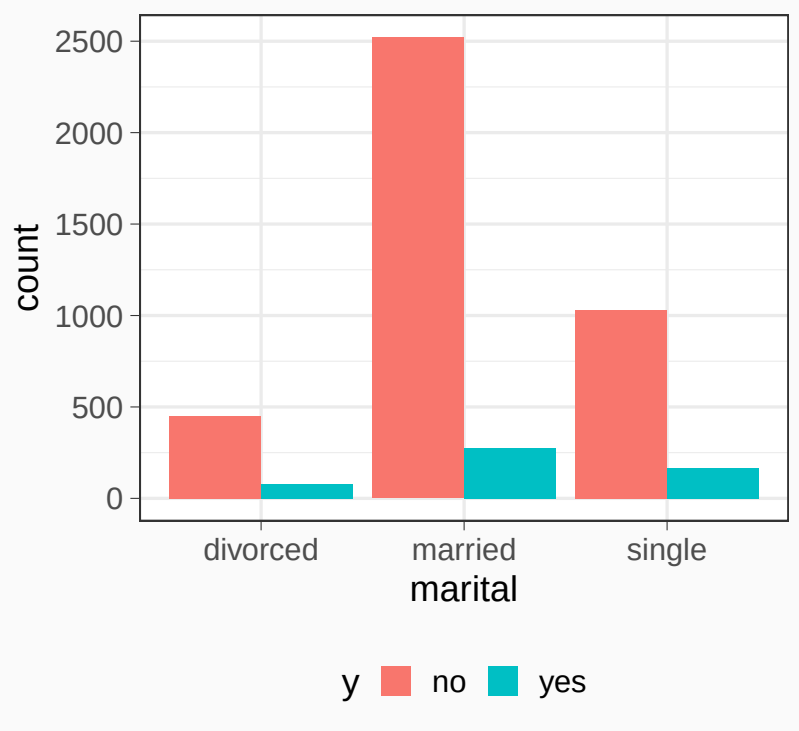
Stacking behavior

- several values for a given `x` induced by `fill`
- stacked by default

Alternatives

- side by side `position="dodge"`
- stacked with normalization `position="fill"`

```
ggplot(bank,aes(x=marital, fill=y)) +
    geom_bar(position="dodge") +
    theme(legend.position="bottom")
```



Variations on fill

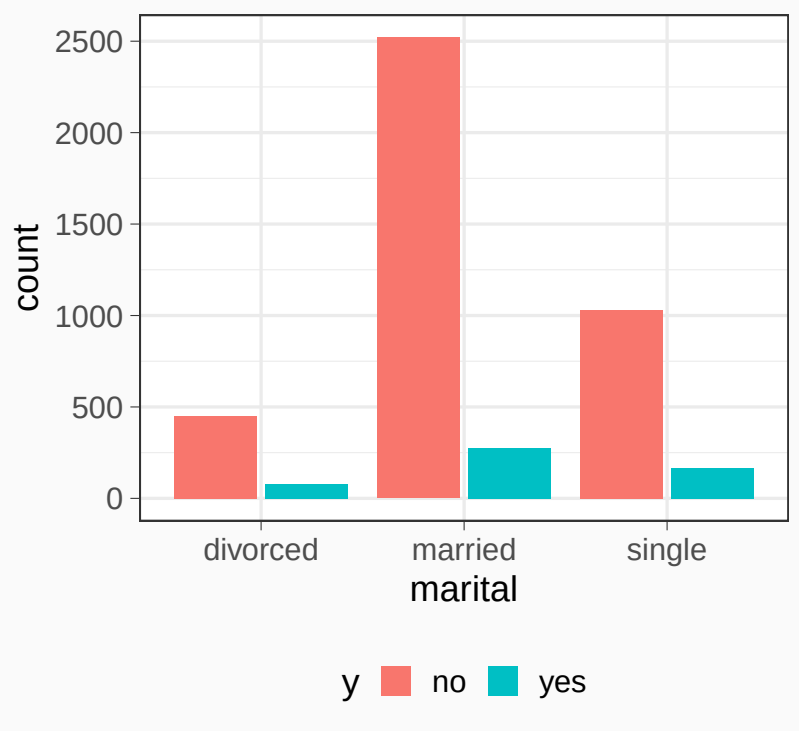
Stacking behavior

- several values for a given `x` induced by `fill`
- stacked by default

Alternatives

- side by side `position="dodge"`
- stacked with normalization `position="fill"`

```
ggplot(bank,aes(x=marital, fill=y)) +
    geom_bar(position="dodge2") +
    theme(legend.position="bottom")
```



Variations on fill

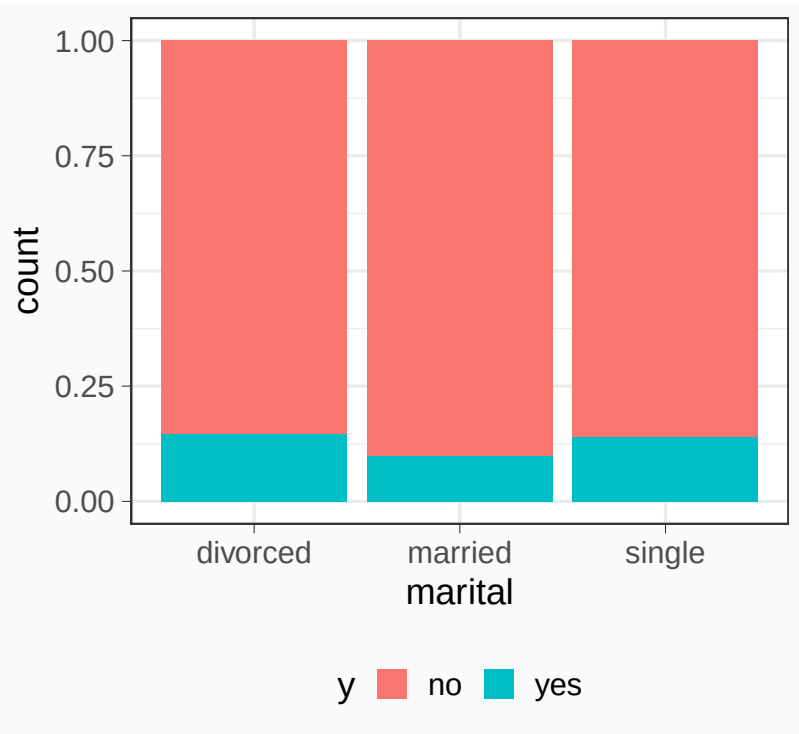
Stacking behavior

- several values for a given `x` induced by `fill`
- stacked by default

Alternatives

- side by side `position="dodge"`
- stacked with normalization `position="fill"`

```
ggplot(bank,aes(x=marital, fill=y)) +
    geom_bar(position="fill") +
    theme(legend.position="bottom")
```



Statistical Transformations

Count?

- a bar chart display a `count` variable
- not available in the original data
- implicit transformation

Stat layer

- calculation layers
- with a default geom layer (and vice versa)

Bar chart

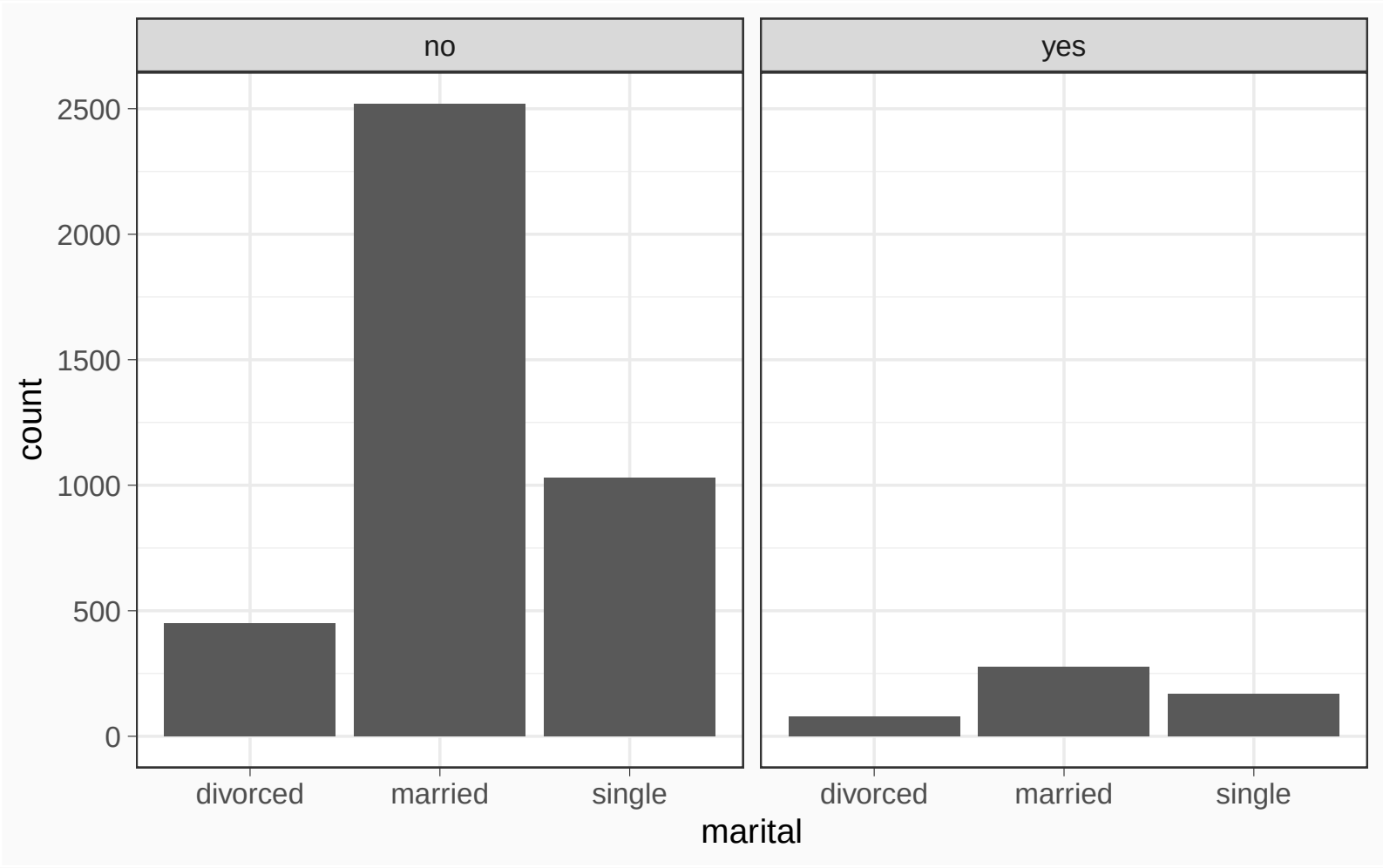
- `geom_bar` and `stat_count`
- `stat_count` computes `count` and `prop`
- `prop` useful in some contexts

Computed variables

- accessible for aesthetics
- `..name..` syntax

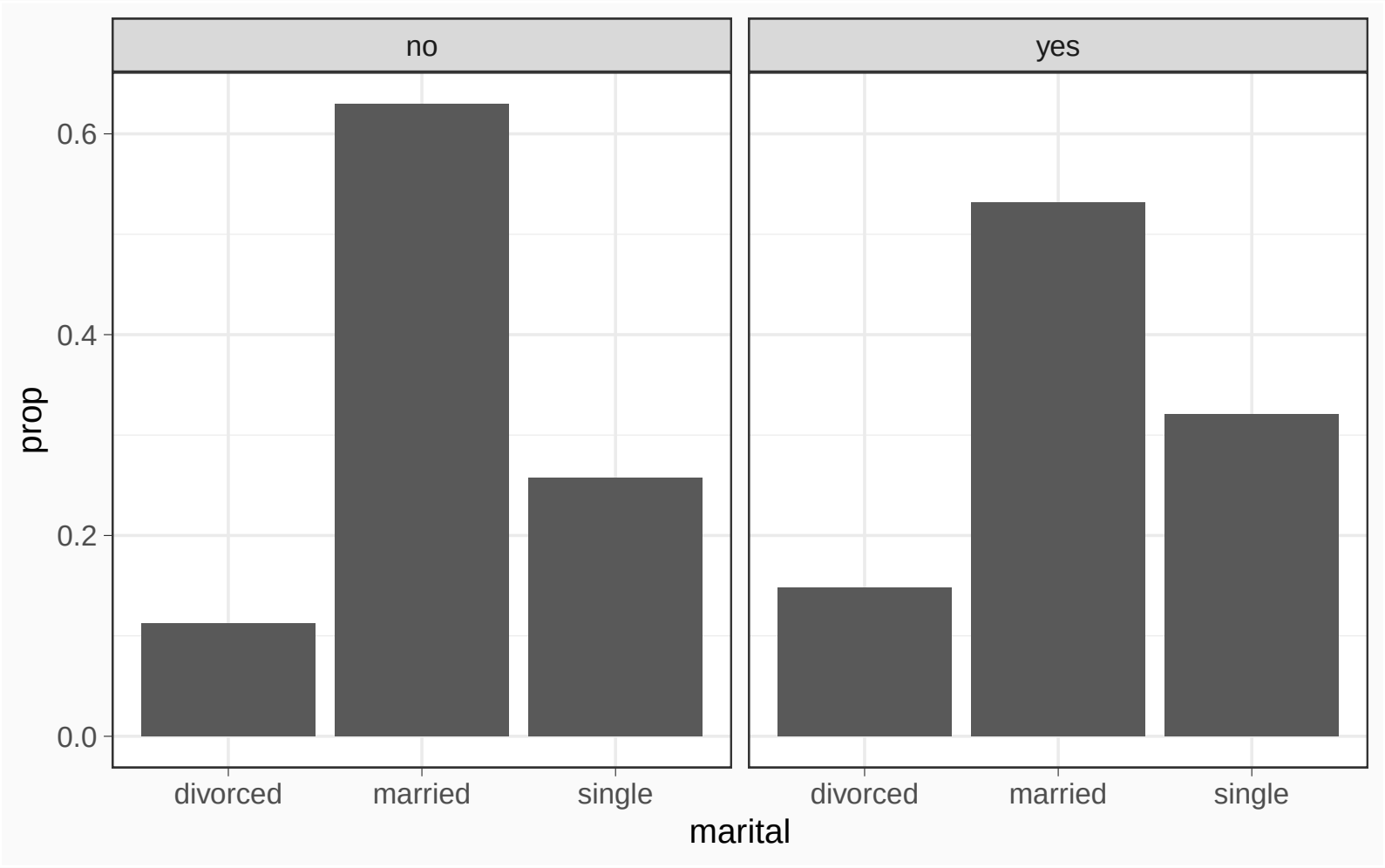
Bank Data

```
ggplot(bank,aes(x=marital)) +
    geom_bar() +
    facet_wrap(~y)
```



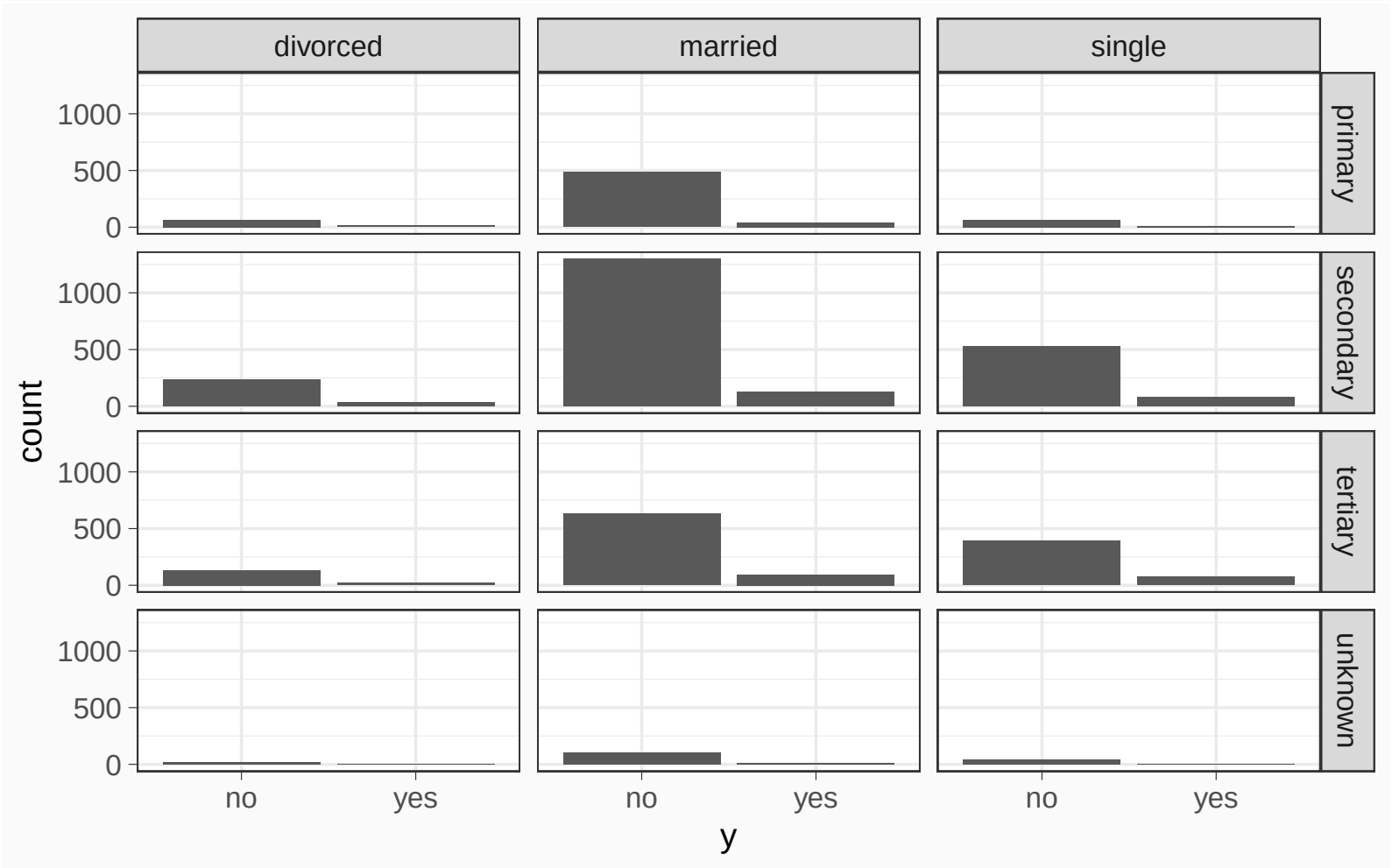
Bank Data

```
ggplot(bank,aes(x=marital)) +
    geom_bar(mapping=aes(y=..prop.., group=1)) +
    facet_wrap(~y)
```



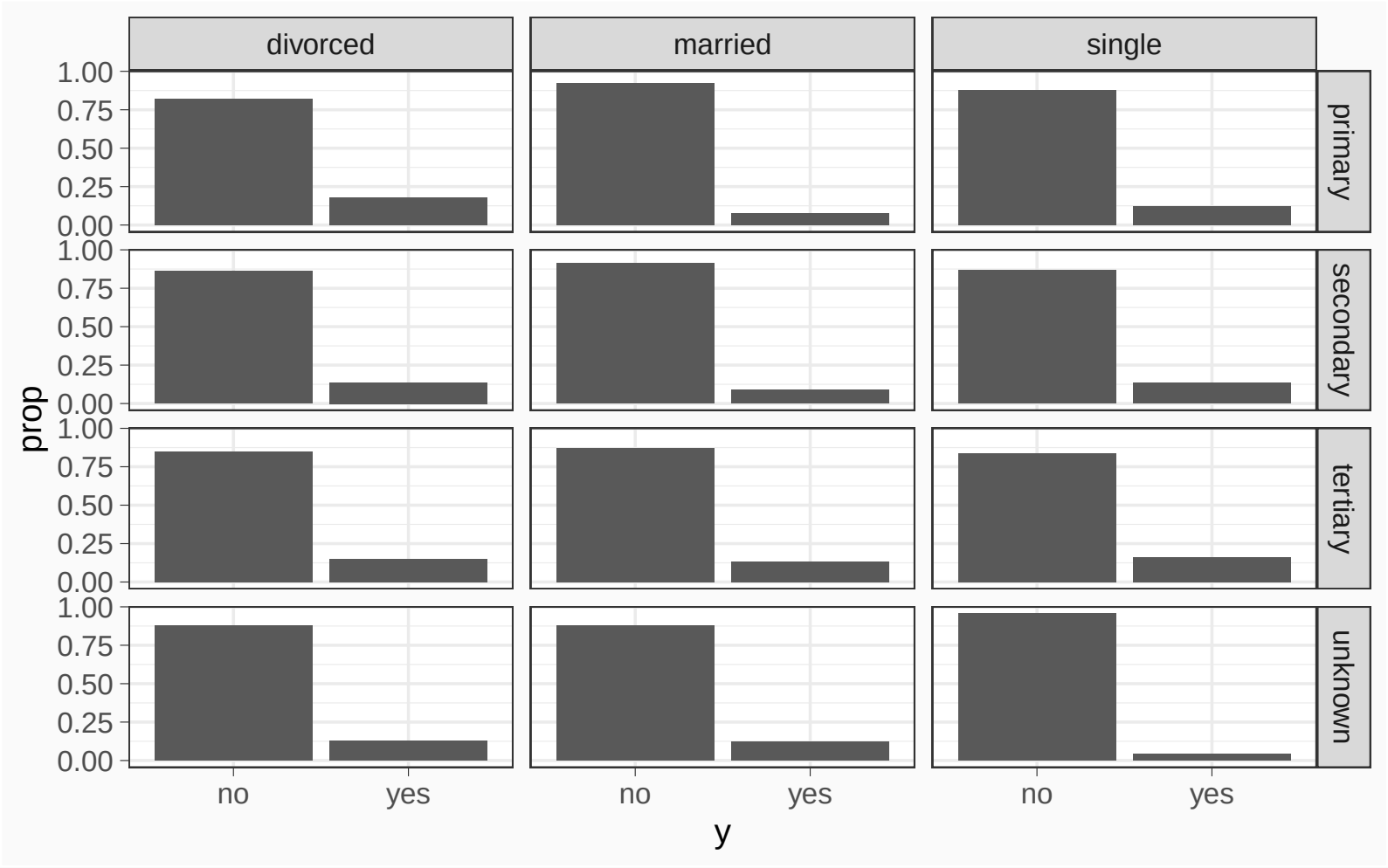
Bank Data

```
ggplot(bank, aes(x = y)) + geom_bar() + facet_grid(education ~
    marital)
```



Bank Data

```
ggplot(bank, aes(x = y)) + geom_bar(mapping = aes(y = ..prop..,
    group = 1)) + facet_grid(education ~ marital)
```



Summarizing continuous variables

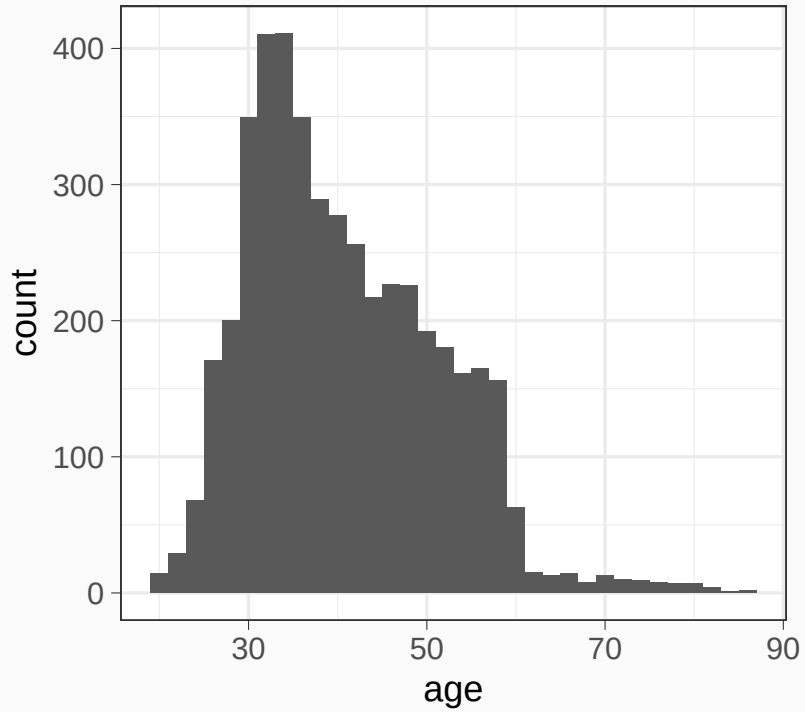
Distribution

- a bar chart represents an estimation of the probability distribution of a discrete variable
- equivalent for continuous variable?

Histogram

- quantify a continuous variable into a discrete one
- display the result in a way vaguely similar to a bar chart

```
ggplot(bank, aes(age)) +
    geom_histogram(binwidth = 2)
```



- `geom_histogram`
- `stat_bin`

Summarizing continuous variables

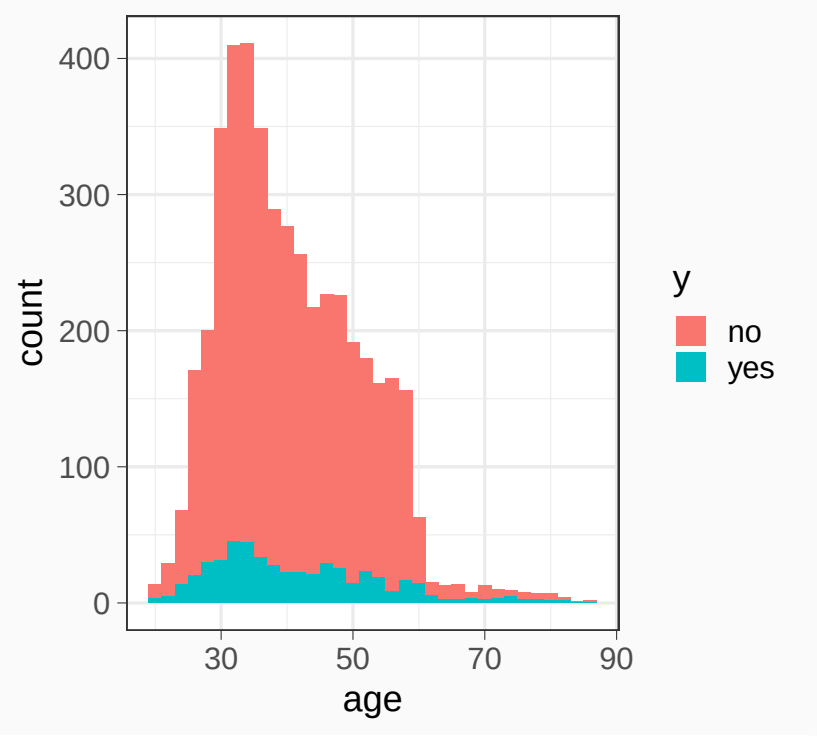
Distribution

- a bar chart represents an estimation of the probability distribution of a discrete variable
- equivalent for continuous variable?

Histogram

- quantify a continuous variable into a discrete one
- display the result in a way vaguely similar to a bar chart

```
ggplot(bank, aes(age, fill = y)) +
    geom_histogram(binwidth = 2)
```



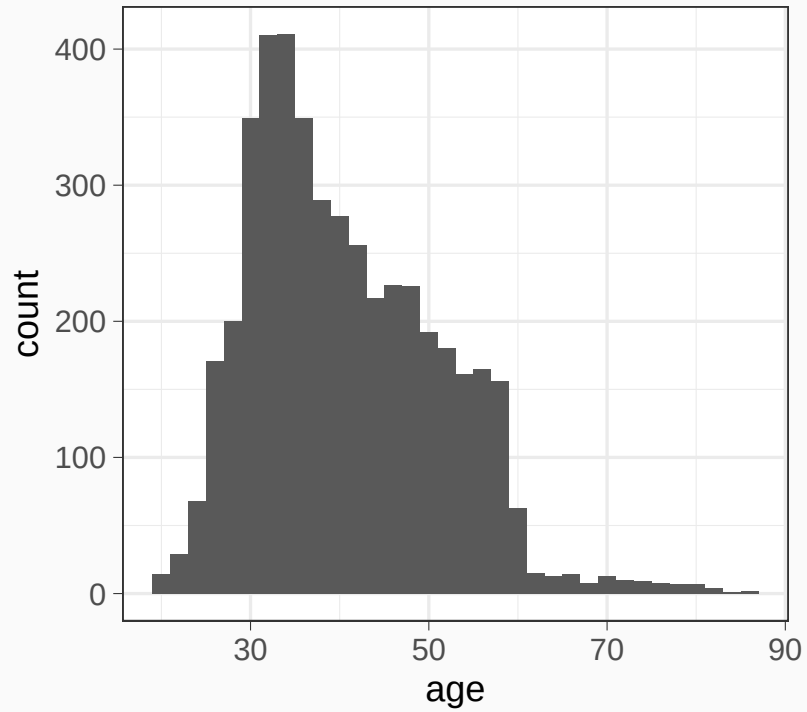
- `geom_histogram`
- `stat_bin`

Summarizing continuous variables

Alternative

- similar binning strategy
- line based representation
- same stat layer, different geom layer

```
ggplot(bank, aes(age)) +
    geom_histogram(binwidth = 2)
```

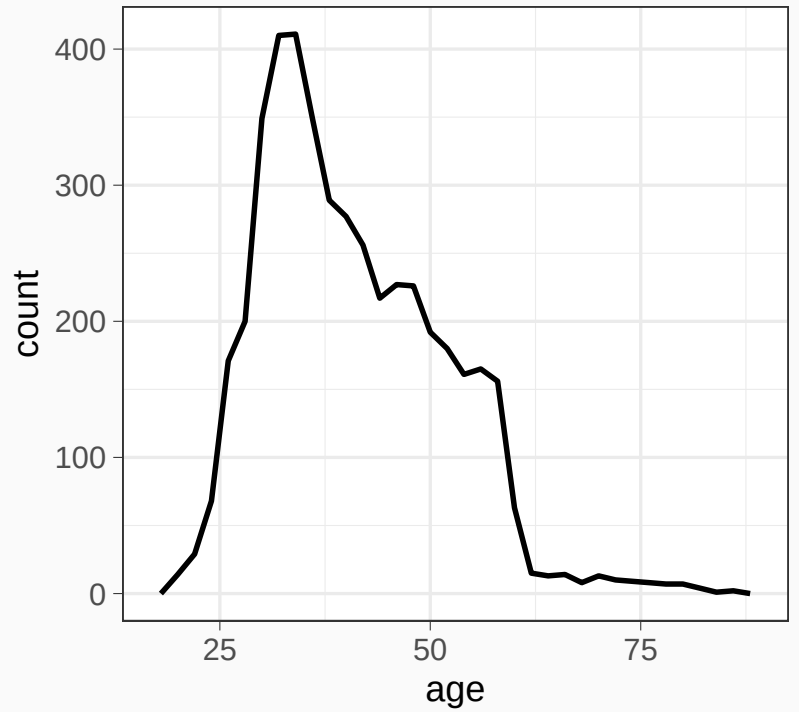


Summarizing continuous variables

Alternative

- similar binning strategy
- line based representation
- same stat layer, different geom layer

```
ggplot(bank, aes(age)) +
    geom_freqpoly(binwidth = 2)
```



Summarizing continuous variables

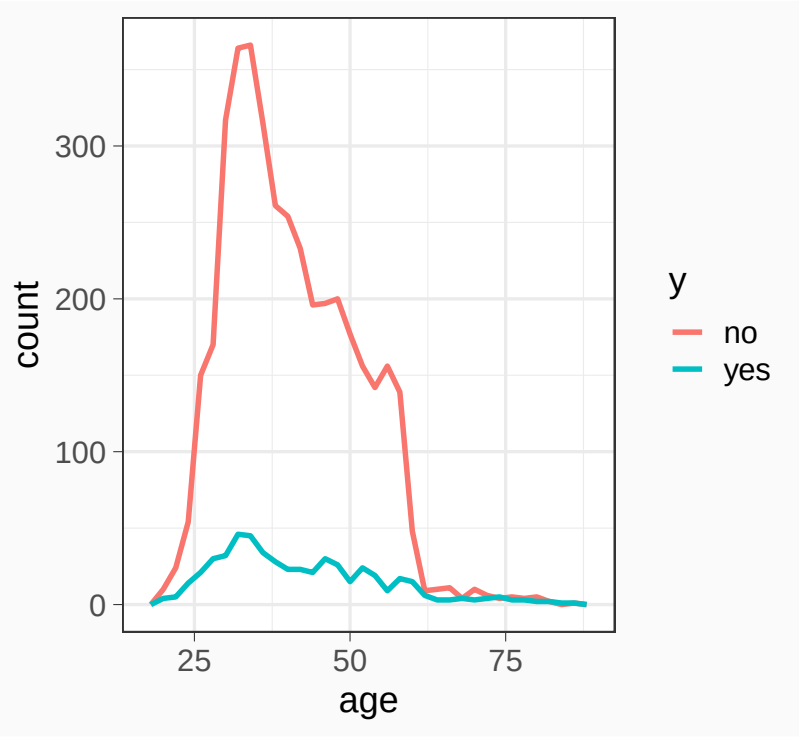
Alternative

- similar binning strategy
- line based representation
- same stat layer, different geom layer

Different conditioning

- no stacking
- multiple curves

```
ggplot(bank, aes(age, color = y)) +
    geom_freqpoly(binwidth = 2)
```



Summarizing continuous variables

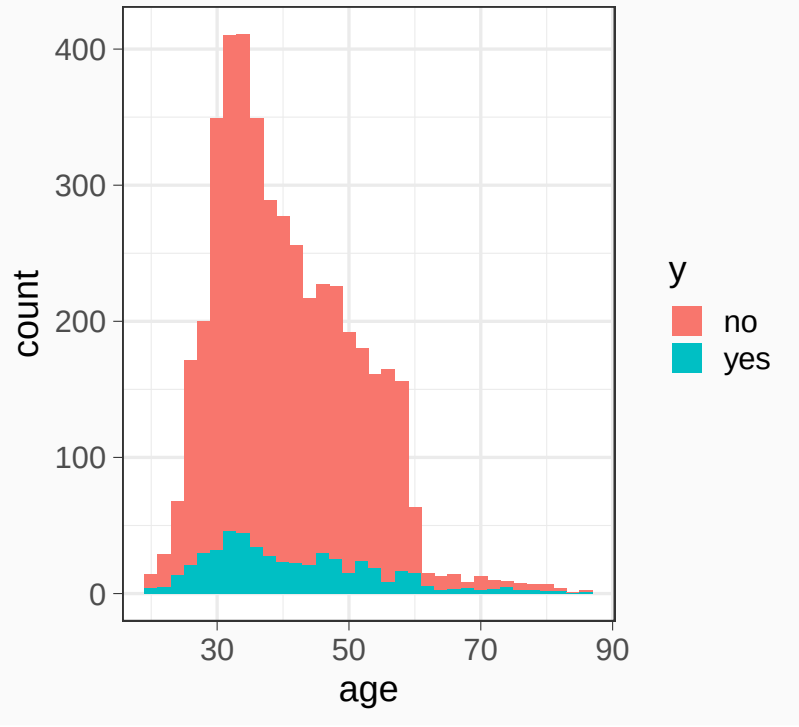
Alternative

- similar binning strategy
- line based representation
- same stat layer, different geom layer

Different conditioning

- no stacking
- multiple curves

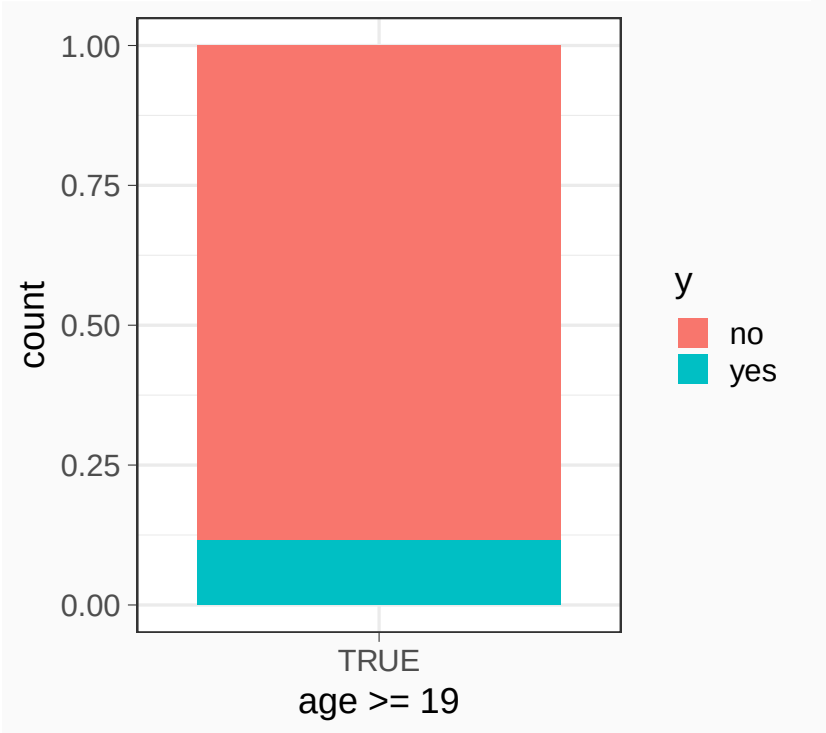
```
ggplot(bank, aes(age, fill = y)) +
    geom_histogram(binwidth = 2)
```



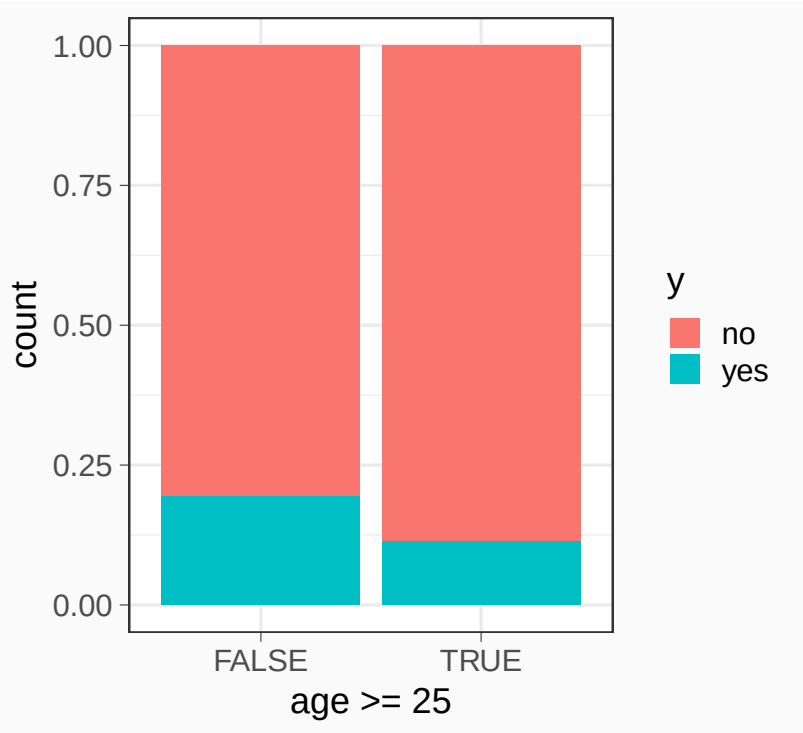
Age dependency

Does the positive answer ratio grow with age?

```
ggplot(bank, aes(x = age >=
    19, fill = y)) +
    geom_bar(position = "fill")
```



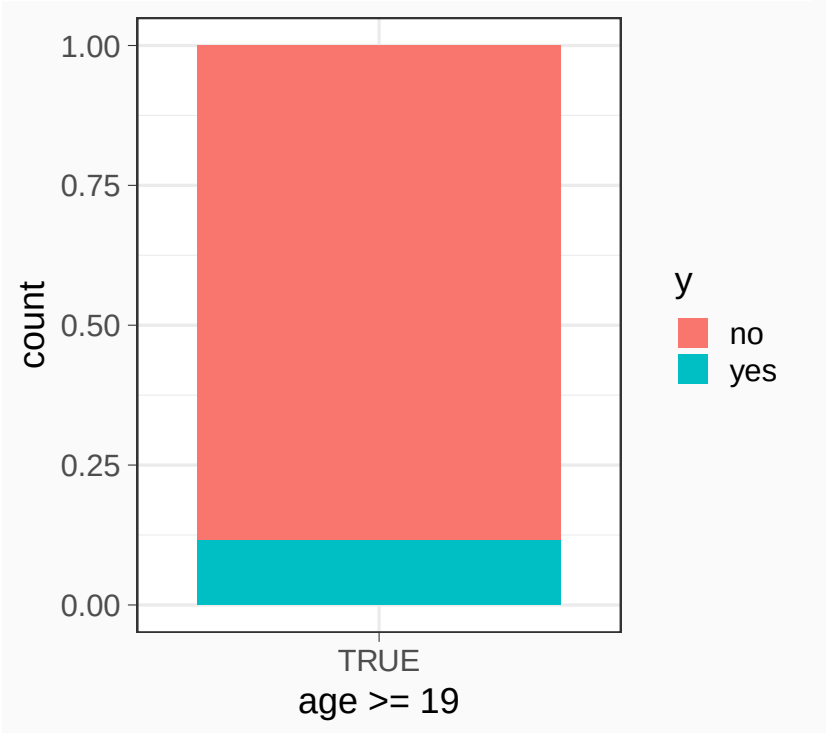

```
ggplot(bank, aes(x = age >=
    25, fill = y)) +
    geom_bar(position = "fill")
```



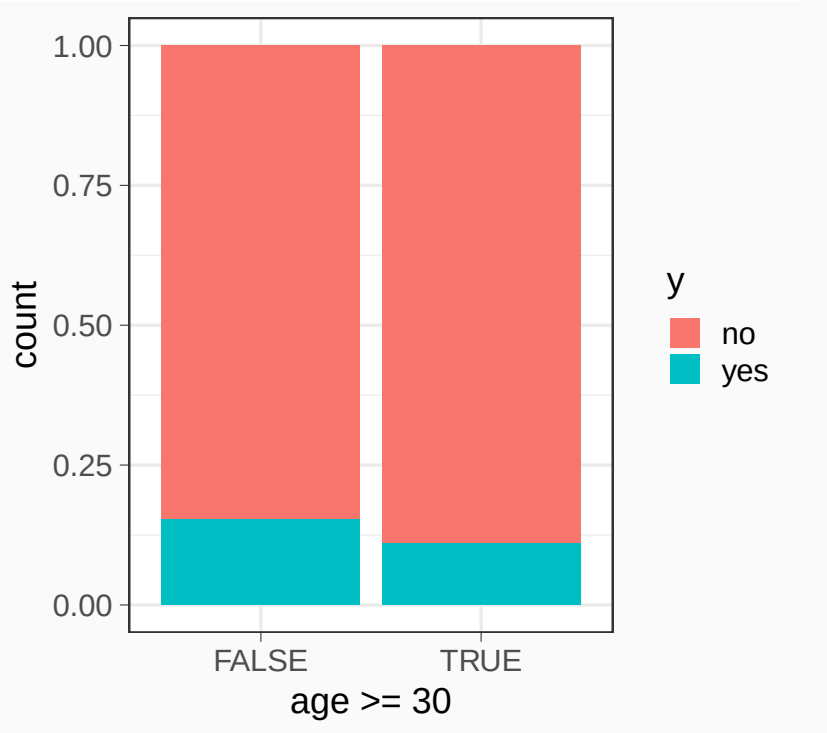
Age dependency

Does the positive answer ratio grow with age?

```
ggplot(bank, aes(x = age >=
    19, fill = y)) +
    geom_bar(position = "fill")
```



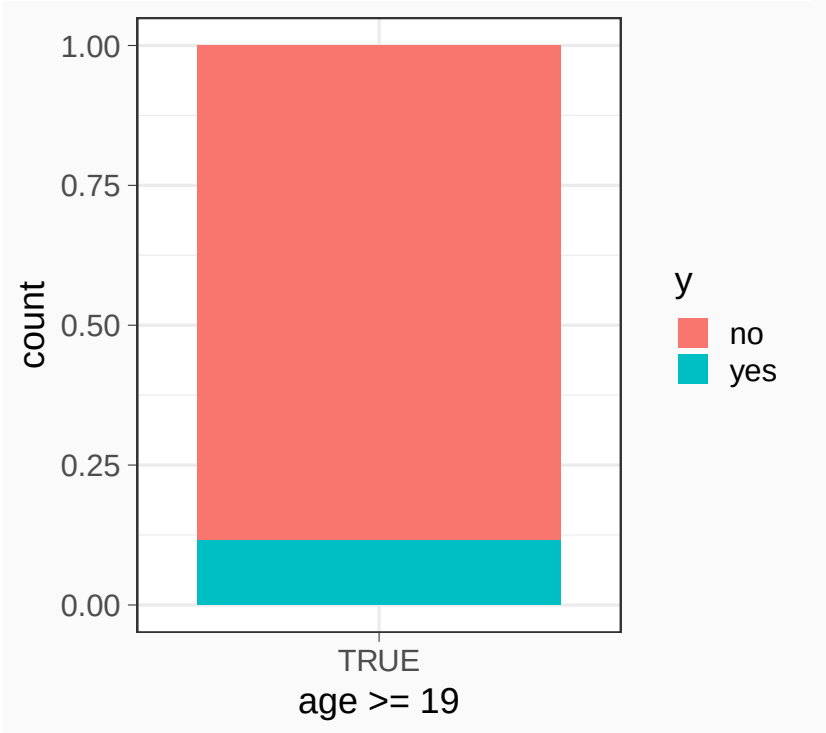

```
ggplot(bank, aes(x = age >=
    30, fill = y)) +
    geom_bar(position = "fill")
```



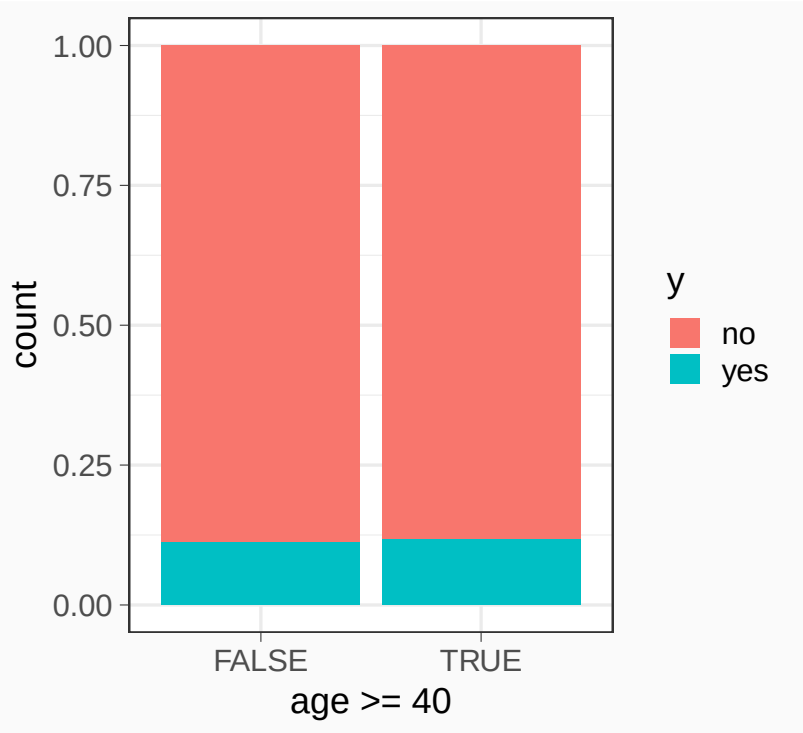
Age dependency

Does the positive answer ratio grow with age?

```
ggplot(bank, aes(x = age >=
    19, fill = y)) +
    geom_bar(position = "fill")
```



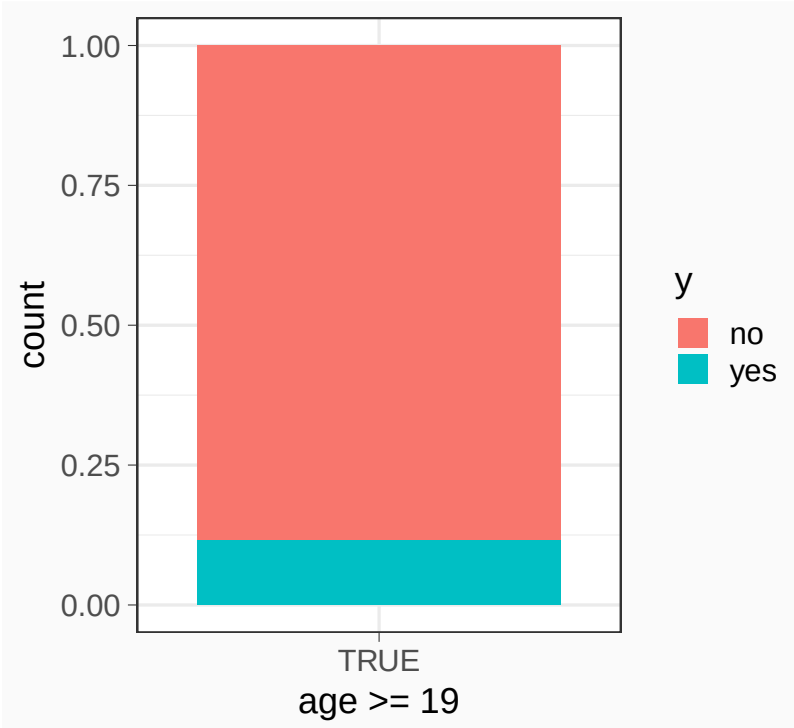

```
ggplot(bank, aes(x = age >=
    40, fill = y)) +
    geom_bar(position = "fill")
```



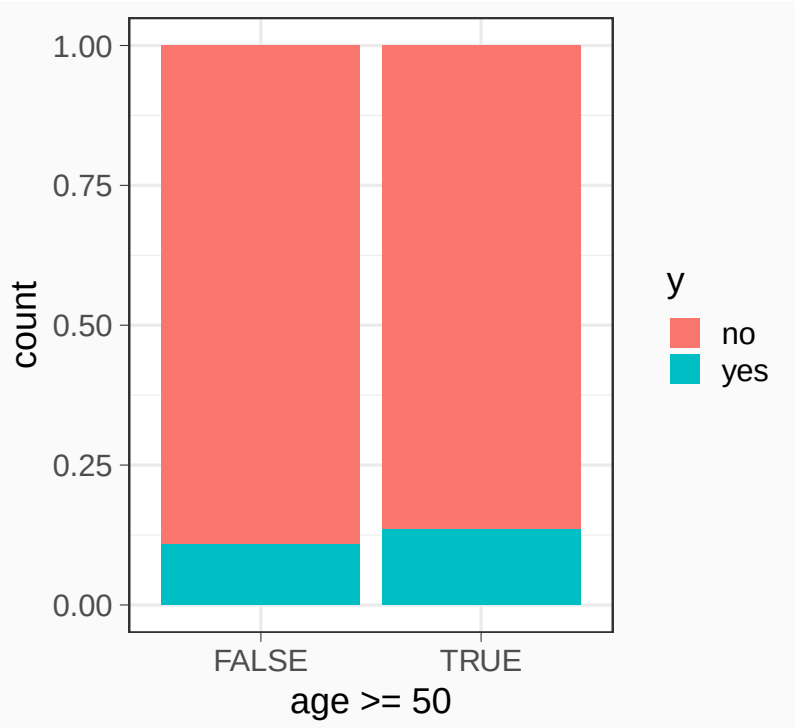
Age dependency

Does the positive answer ratio grow with age?

```
ggplot(bank, aes(x = age >=
    19, fill = y)) +
    geom_bar(position = "fill")
```



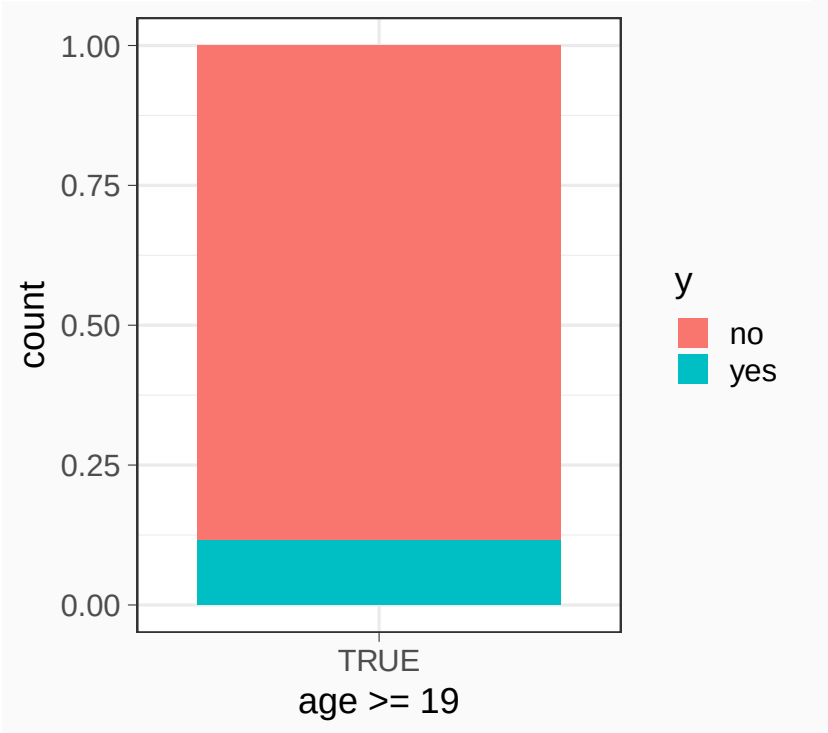

```
ggplot(bank, aes(x = age >=
    50, fill = y)) +
    geom_bar(position = "fill")
```



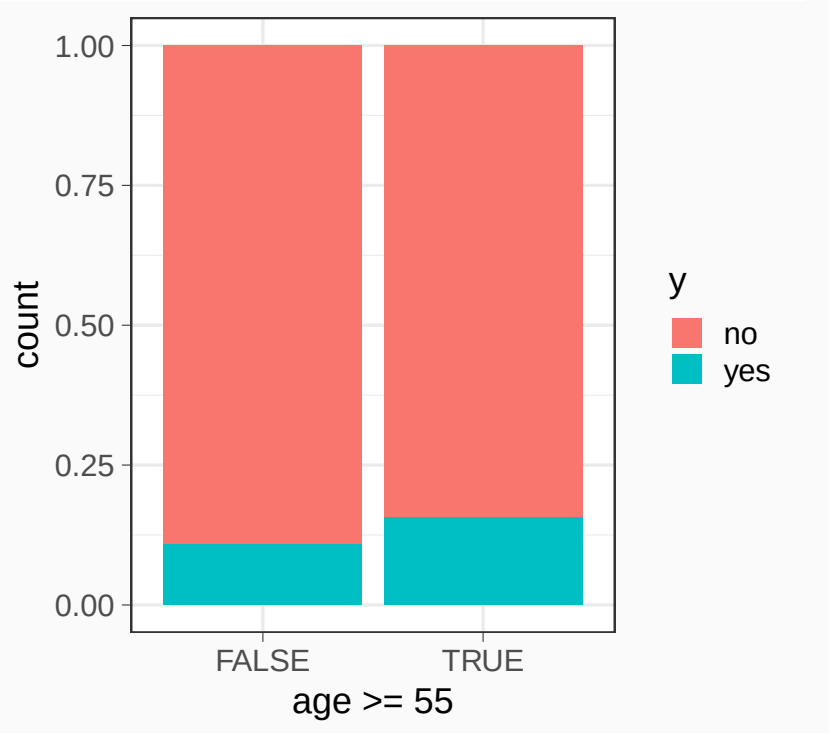
Age dependency

Does the positive answer ratio grow with age?

```
ggplot(bank, aes(x = age >=
    19, fill = y)) +
    geom_bar(position = "fill")
```



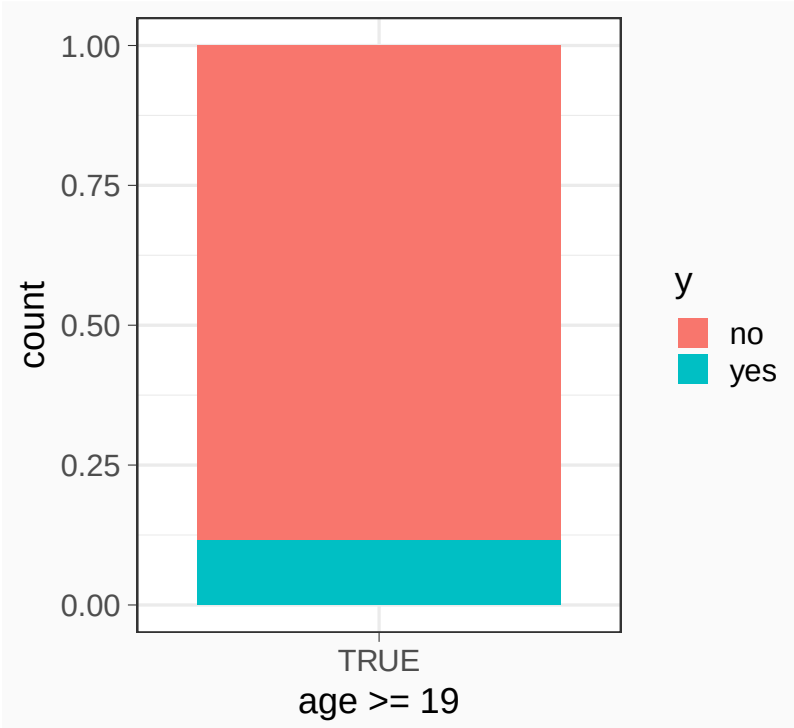

```
ggplot(bank, aes(x = age >=
    55, fill = y)) +
    geom_bar(position = "fill")
```



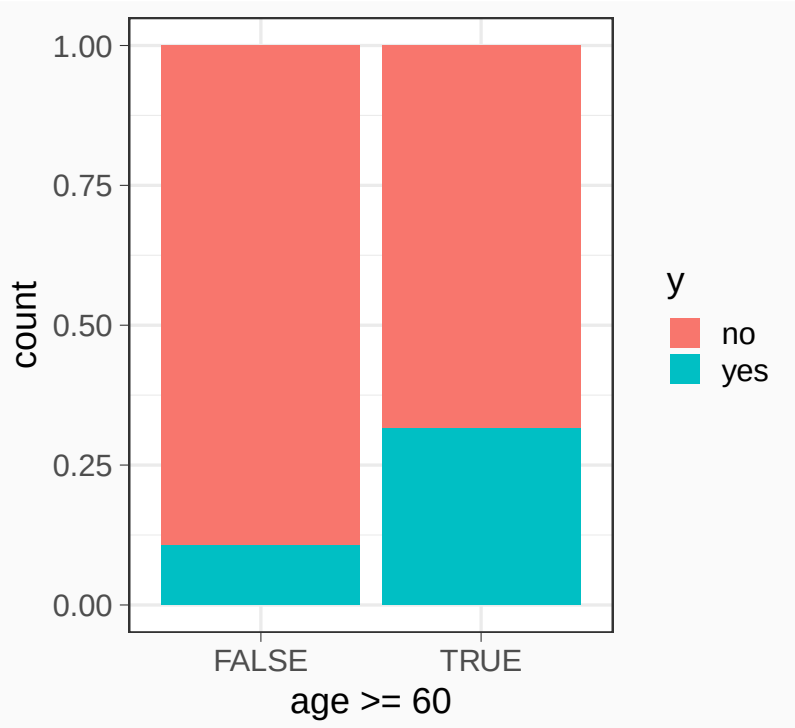
Age dependency

Does the positive answer ratio grow with age?

```
ggplot(bank, aes(x = age >=
    19, fill = y)) +
    geom_bar(position = "fill")
```




```
ggplot(bank, aes(x = age >=
    60, fill = y)) +
    geom_bar(position = "fill")
```



Data Analysis Intermezzo

Age categories

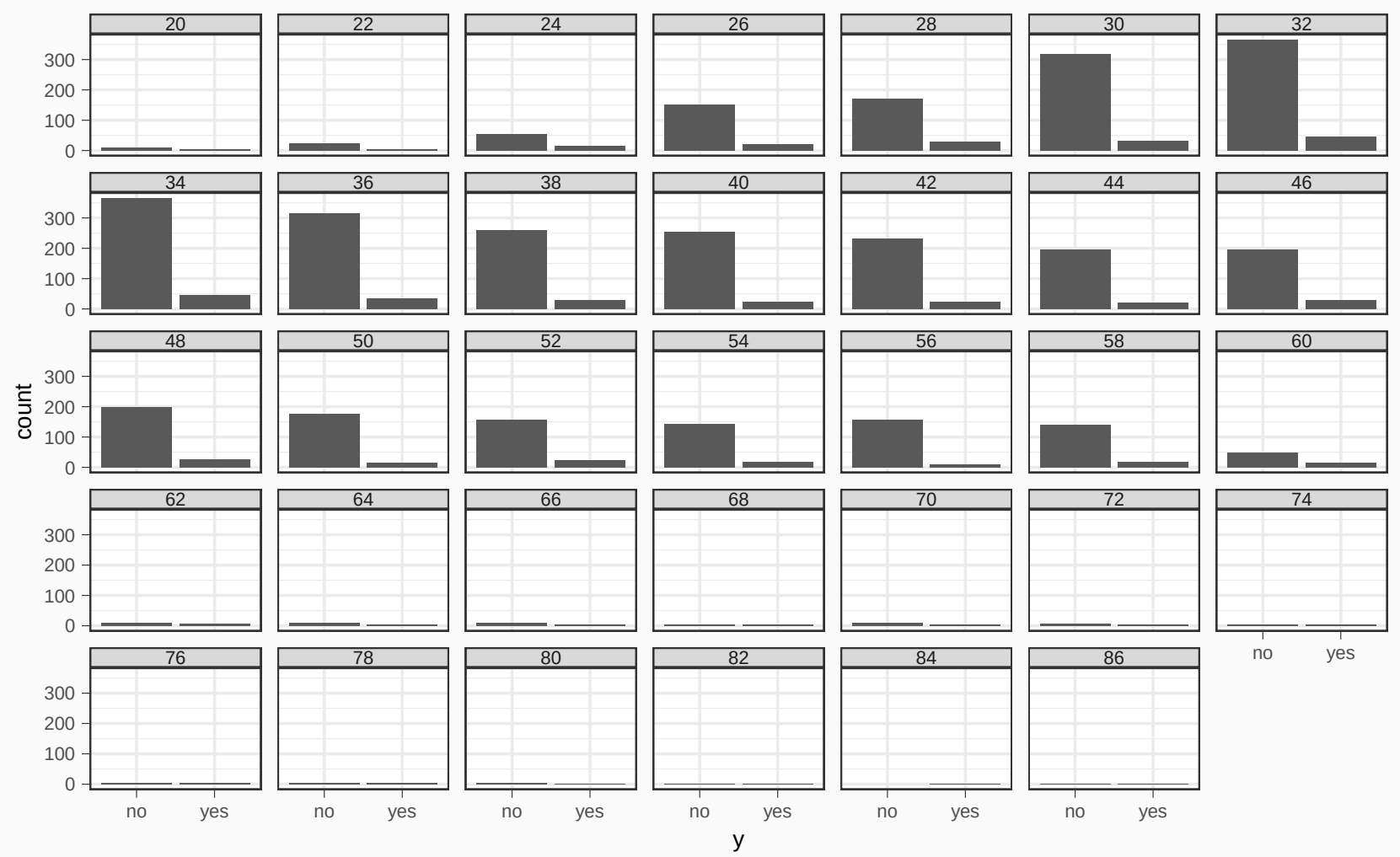
```
bank <- bank %>%
    mutate(agegroup=18+2*cut(age,
           breaks=seq(19,91,by=2),
           labels=FALSE,include.lowest=TRUE))
bank %>% print(n=5,width=35)
## # A tibble: 4,521 x 18
##     age job    marital
##   <dbl> <chr> <chr>
## 1    30 unem~ married
## 2    33 serv~ married
## 3    35 mana~ single
## 4    30 mana~ married
## 5    59 blue~ married
##   education default balance
##   <chr>     <chr>     <dbl>
## 1 primary   no         1787
## 2 secondary no         4789
## 3 tertiary  no         1350
## 4 tertiary  no         1476
## 5 secondary no            0
## # ... with 4,516 more rows,
## #   and 12 more variables:
## #   housing <chr>,
## #   loan <chr>,
## #   contact <chr>, day <dbl>,
## #   month <chr>,
## #   duration <dbl>,
## #   campaign <dbl>,
## #   pdays <dbl>,
## #   previous <dbl>,
## #   poutcome <chr>, y <chr>,
## #   agegroup <dbl>
```

Yes frequencies

```
bankbyage <- bank %>% group_by(agegroup,y) %>%
    summarise(n=n()) %>% group_by(agegroup) %>%
    mutate(freq=n/sum(n)) %>% filter(y=="yes")
bankbyage %>% print(n=6,width=30)
## # A tibble: 34 x 4
## # Groups:   agegroup [34]
##   agegroup y         n   freq
##      <dbl> <chr> <int>  <dbl>
## 1       20 yes       4 0.286
## 2       22 yes       5 0.172
## 3       24 yes      14 0.206
## 4       26 yes      21 0.123
## 5       28 yes      30 0.15
## 6       30 yes      32 0.0917
## # ... with 28 more rows
```

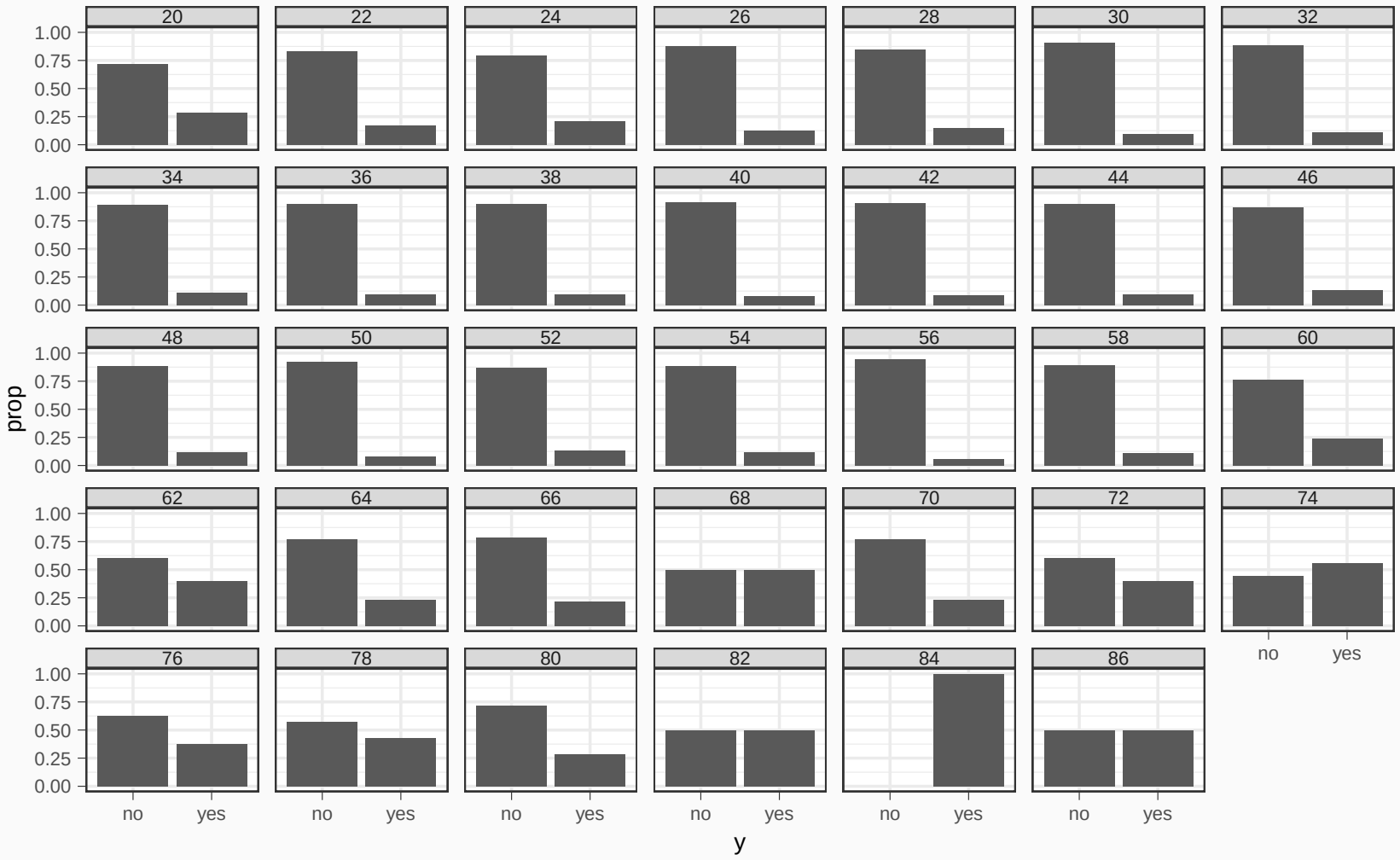
Data Analysis Intermezzo

```
ggplot(bank,aes(y)) + geom_bar() + facet_wrap(~agegroup,nrow=5) +
    theme(text=element_text(size=5),
          strip.text=element_text(size=4,margin=margin(0.01,0.01,0.01,0.01,"cm")))
```



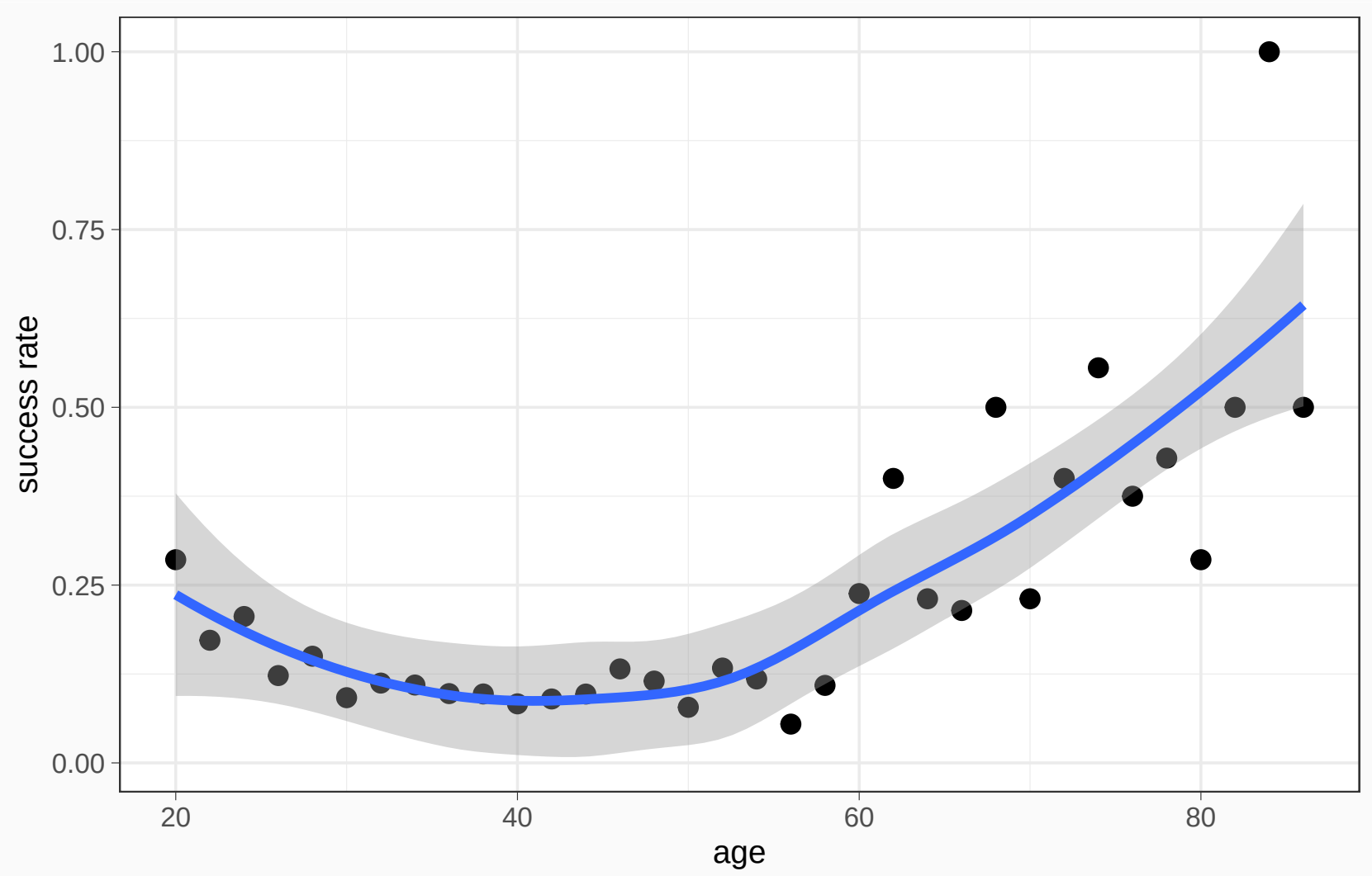
Data Analysis Intermezzo

```
ggplot(bank,aes(y)) + geom_bar(aes(y=..prop..,group=1)) + facet_wrap(~agegroup,nrow=5) +
    theme(text=element_text(size=5),
          strip.text=element_text(size=4,margin=margin(0.01,0.01,0.01,0.01,"cm")))
```



Data Analysis Intermezzo

```
ggplot(bankbyage,aes(x=agegroup,y=freq)) + geom_point() +
    geom_smooth() + labs(x="age",y="success rate")
```



Data Analysis Intermezzo

Typical data analysis sequence

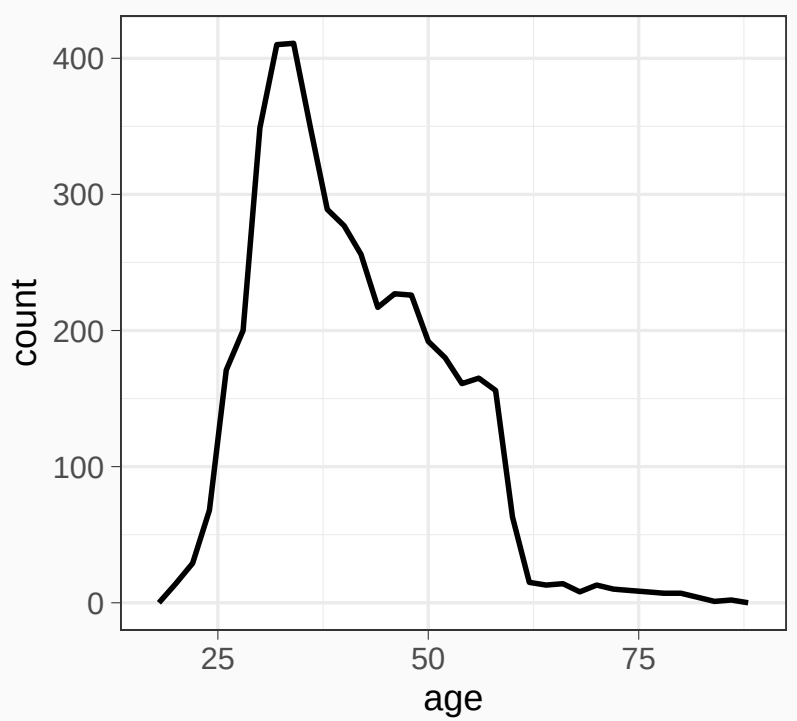
1. display the target variable and another variable
2. spot a potential dependency
3. use other visualizations to confirm/reject the dependency
4. compute new variables (in general aggregated ones)
5. use again visualizations to explore the dependency
6. rinse and repeat

Summarizing continuous variables

Density estimation

- similar goal: probability estimation
- continuous random variable version: density
- smooth estimate (binning replaced by smoothing)
- `geom_density` (with `stat`)

```
ggplot(bank,aes(age)) +
    geom_freqpoly(binwidth = 2)
```

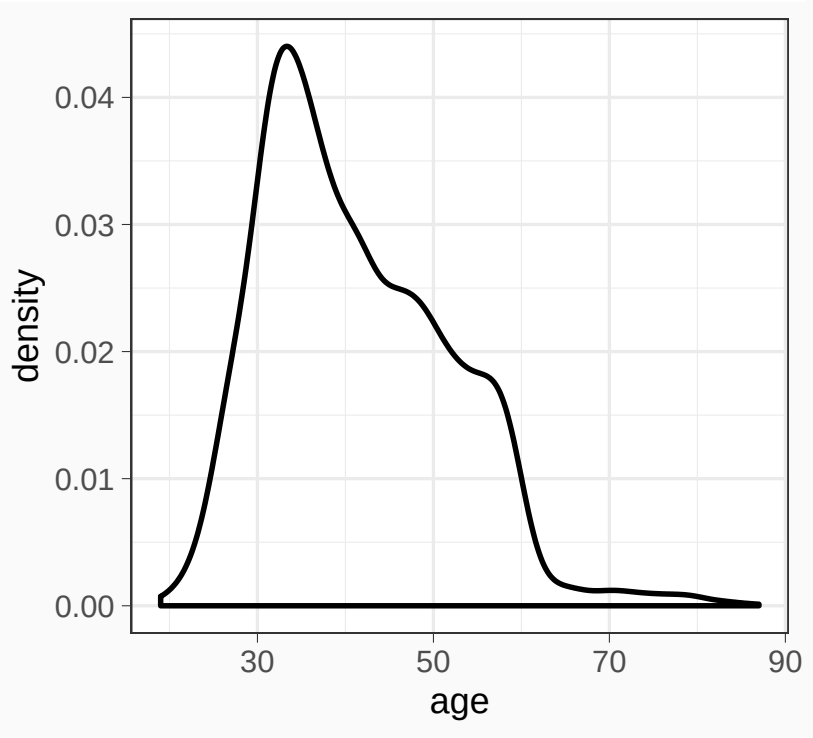


Summarizing continuous variables

Density estimation

- similar goal: probability estimation
- continuous random variable version: density
- smooth estimate (binning replaced by smoothing)
- `geom_density` (with `stat`)

```
ggplot(bank, aes(age)) +
    geom_density()
```

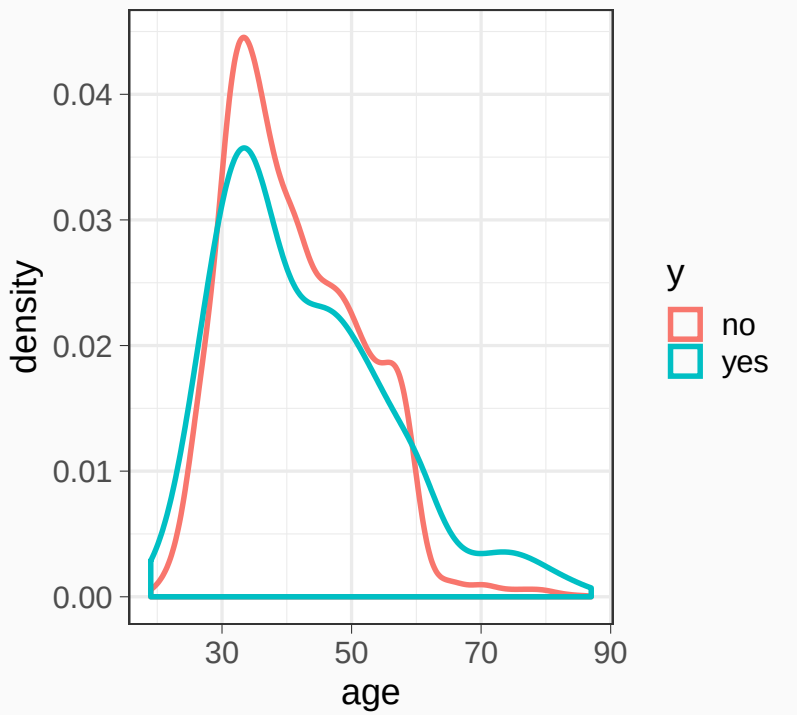


Summarizing continuous variables

Density estimation

- similar goal: probability estimation
- continuous random variable version: density
- smooth estimate (binning replaced by smoothing)
- `geom_density` (with `stat`)

```
ggplot(bank, aes(age, color = y)) +
    geom_density()
```



Different conditioning

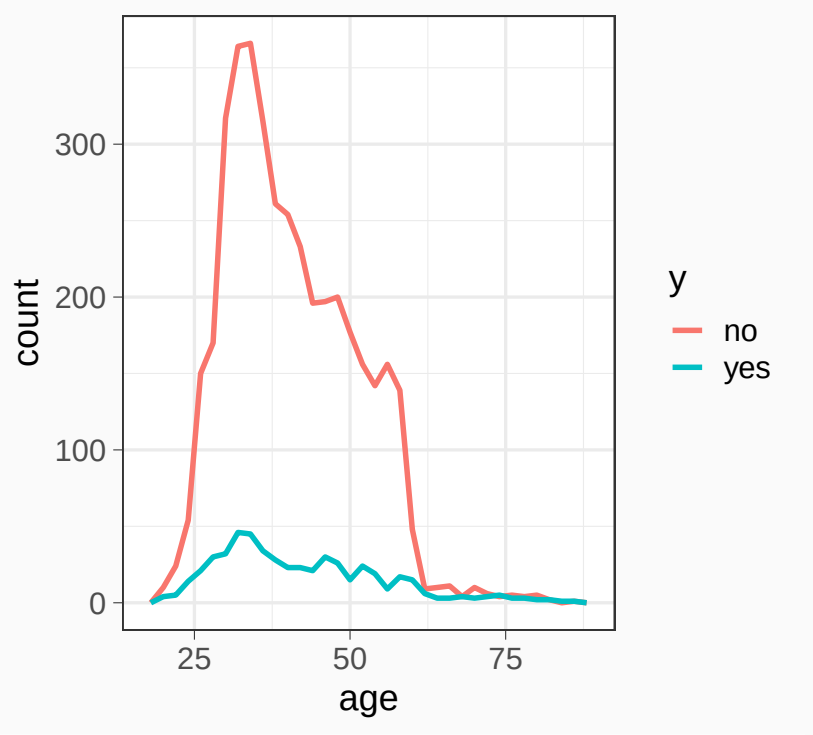
- multiple curves
- curve by curve normalization

Summarizing continuous variables

Density estimation

- similar goal: probability estimation
- continuous random variable version: density
- smooth estimate (binning replaced by smoothing)
- `geom_density` (with `stat`)

```
ggplot(bank, aes(age, color = y)) +
    geom_freqpoly(binwidth = 2)
```



Different conditioning

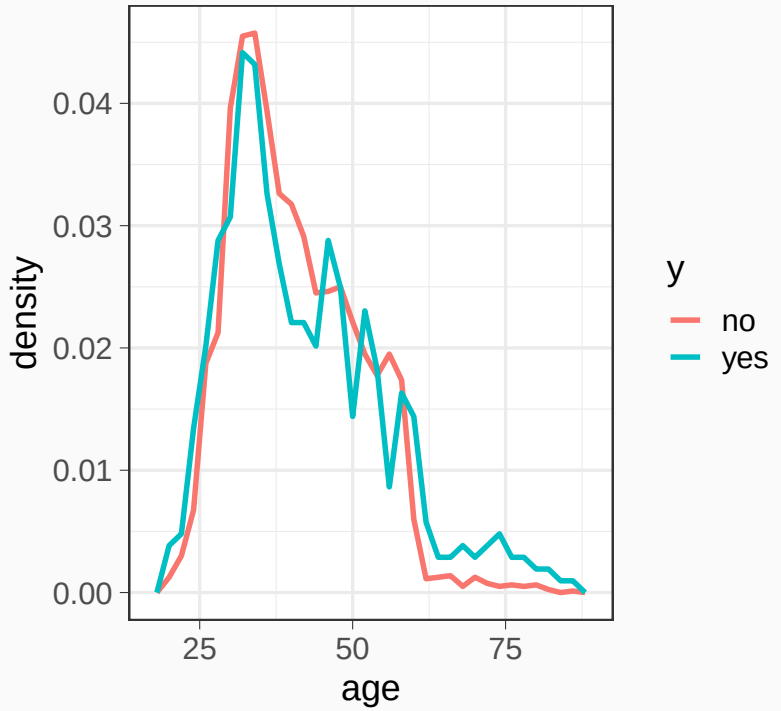
- multiple curves
- curve by curve normalization

Summarizing continuous variables

Density estimation

- similar goal: probability estimation
- continuous random variable version: density
- smooth estimate (binning replaced by smoothing)
- `geom_density` (with `stat`)

```
ggplot(bank, aes(age, color = y)) +
    geom_freqpoly(aes(y=..density..),
                  binwidth = 2)
```



Different conditioning

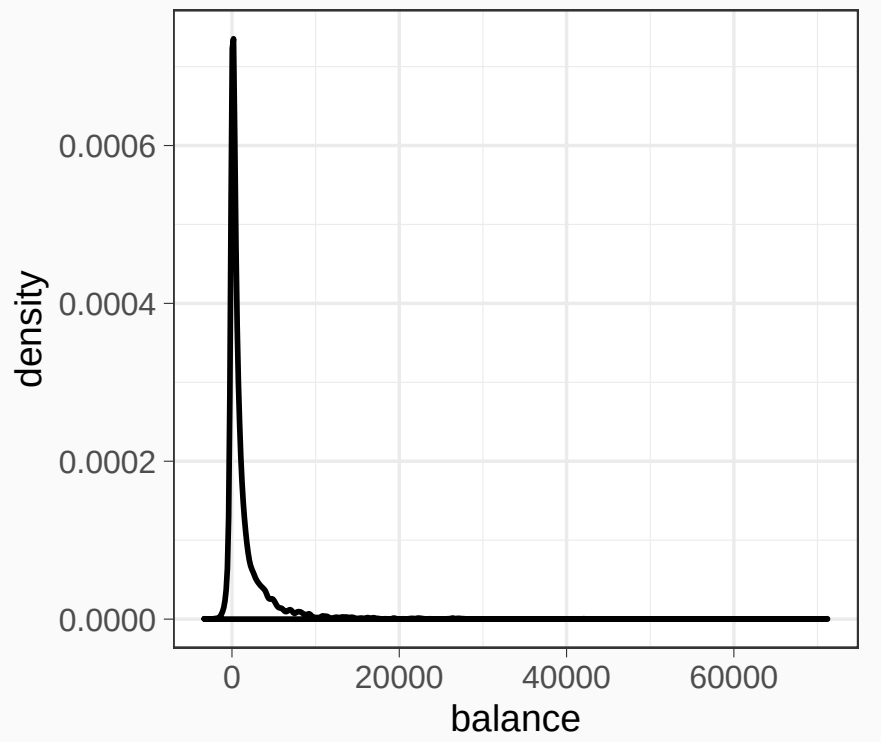
- multiple curves
- curve by curve normalization

Example

Tail effects

- a small number of extreme values
- most graphs are barely readable

```
ggplot(bank,aes(balance)) +
    geom_density()
```

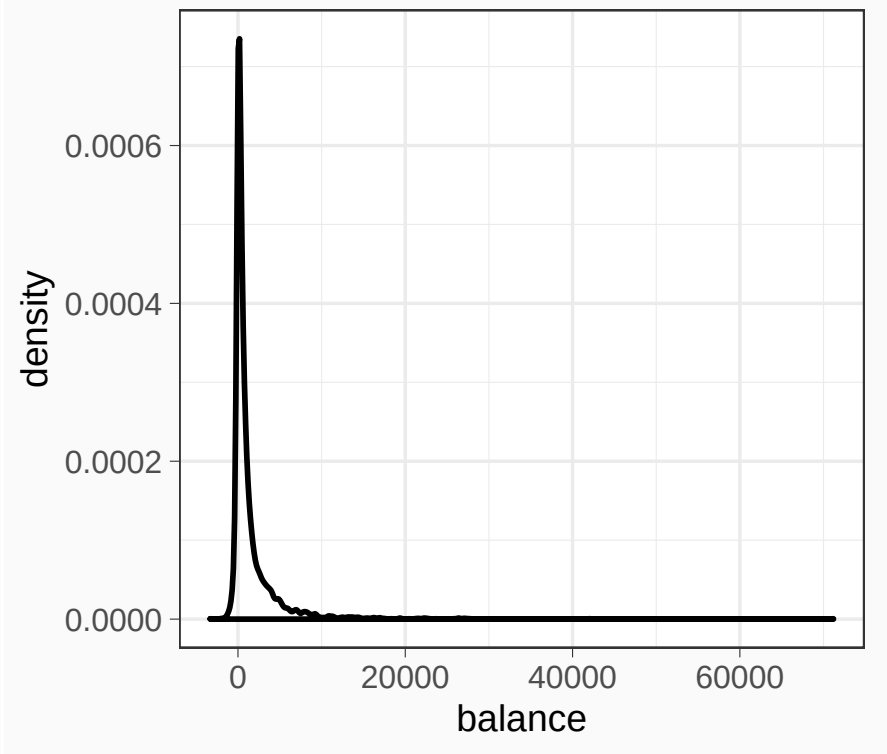


Example

Tail effects

- a small number of extreme values
- most graphs are barely readable

```
ggplot(bank,aes(balance)) +
    geom_density()
```



Filtering

- get quantiles

```
bank %>%summarise(quantile(balance,
                           probs=0.95))
## # A tibble: 1 x
## #   1
##   `quantile(bal~
##             <dbl>
## 1           6102

bank %>%summarise(quantile(balance,
                           probs=0.99))
## # A tibble: 1 x
## #   1
##   `quantile(bal~
##             <dbl>
## 1         14195.
```

Example

Tail effects

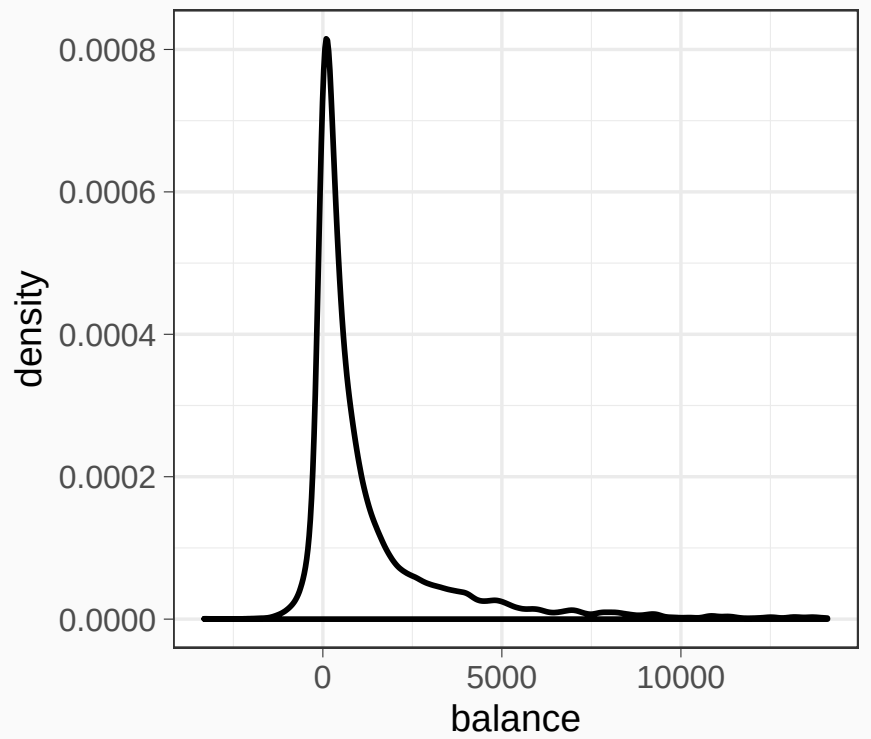
- a small number of extreme values
- most graphs are barely readable

Filtering

- filtering

```
bank2 <- bank %>% filter(balance<=14195)
```

```
ggplot(bank2,aes(balance)) +
    geom_density()
```

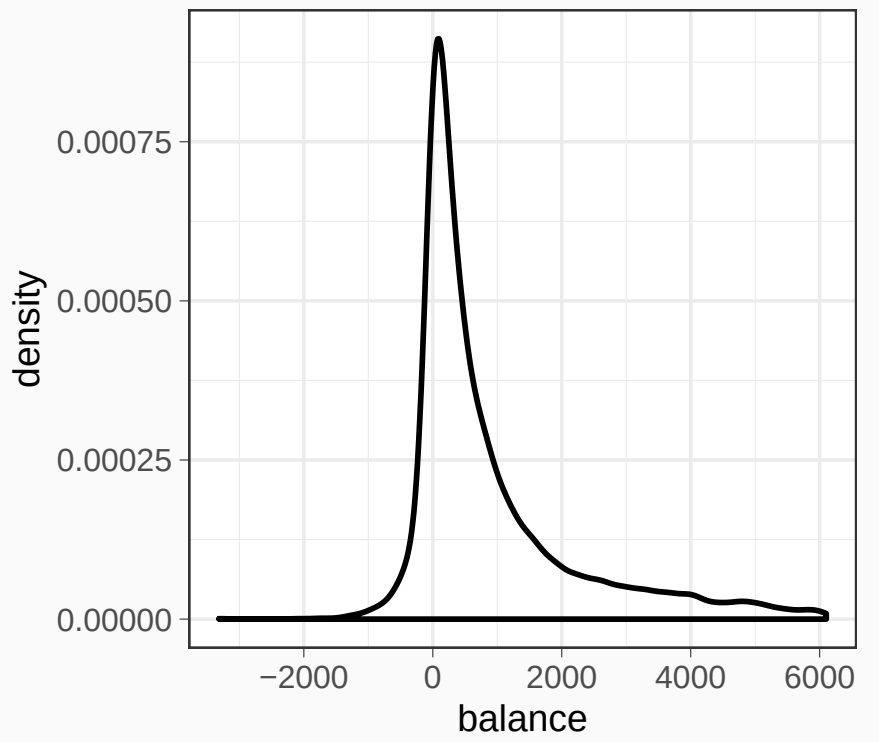


Example

Tail effects

- a small number of extreme values
- most graphs are barely readable

```
ggplot(bank3,aes(balance)) +
    geom_density()
```



Filtering

- filtering

```
bank2 <- bank %>% filter(balance<=14195)
```

- filtering more

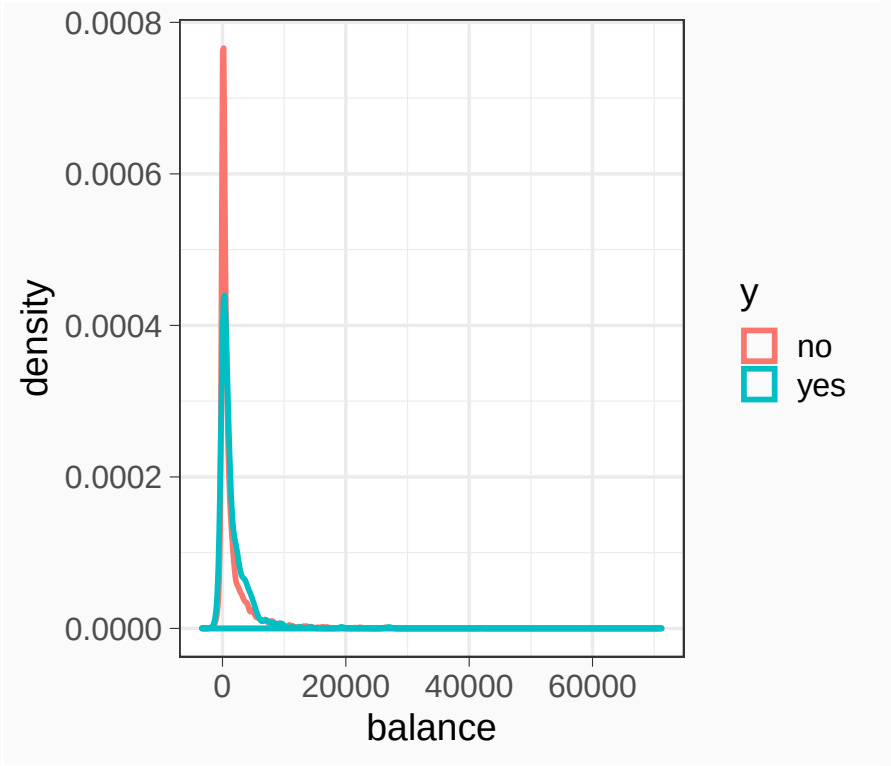
```
bank3 <- bank %>% filter(balance<=6102)
```

Example

Filtering

- extremely important to make graph readable
- usefulness increases with the complexity of the graph
- nonlinear transformations can help also (e.g. logarithm)

```
ggplot(bank,aes(balance,color=y)) +
    geom_density()
```

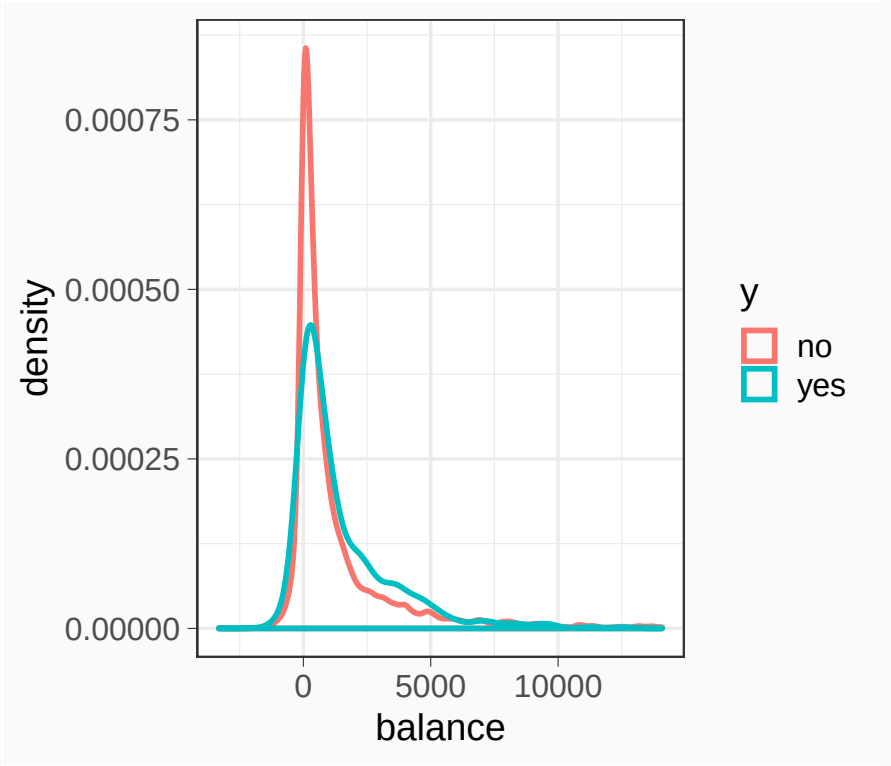


Example

Filtering

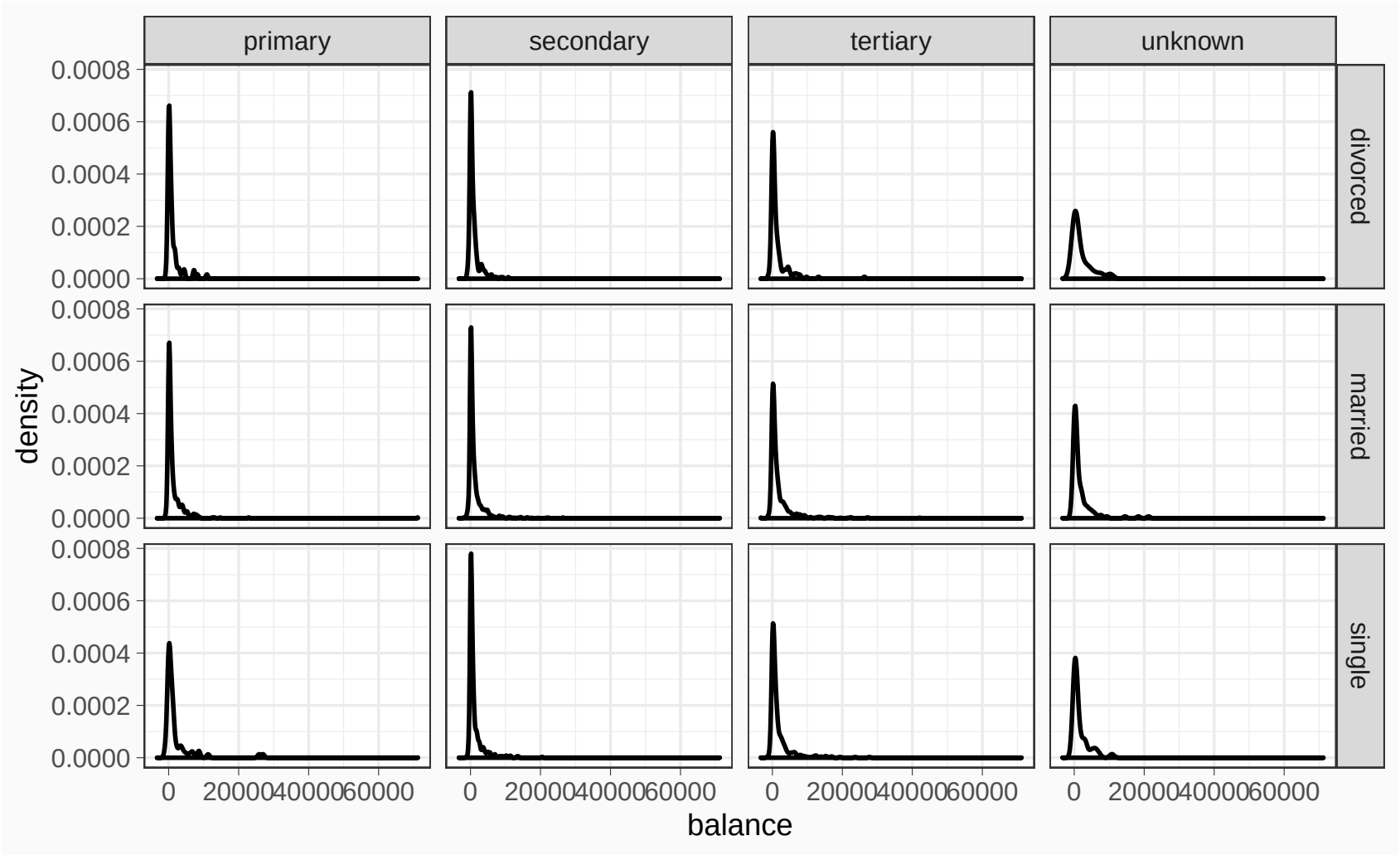
- extremely important to make graph readable
- usefulness increases with the complexity of the graph
- nonlinear transformations can help also (e.g. logarithm)

```
ggplot(bank2,aes(balance,color=y)) +
    geom_density()
```



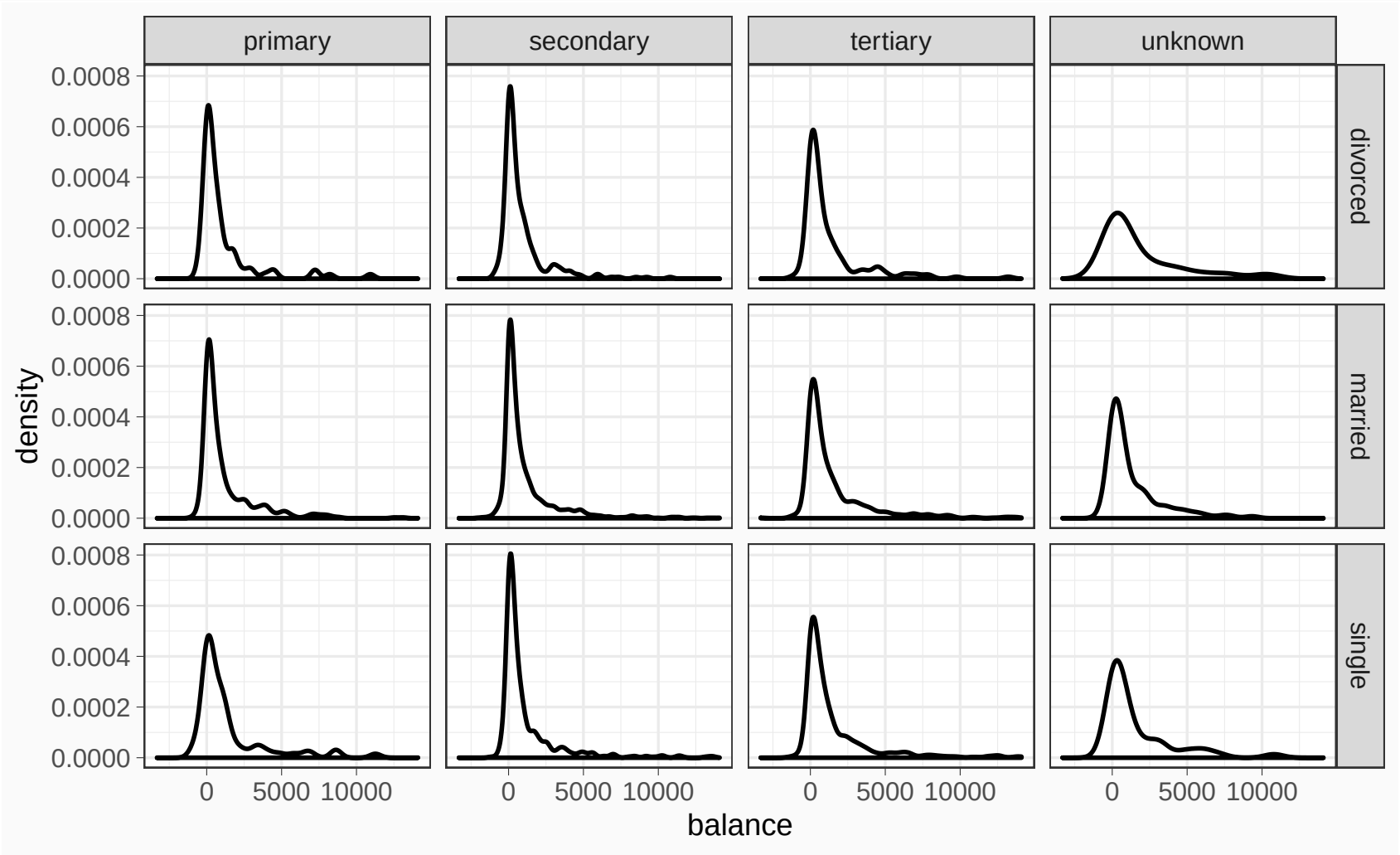
Example

```
bank %>% ggplot(aes(balance)) + geom_density() + facet_grid(marital~education)
```



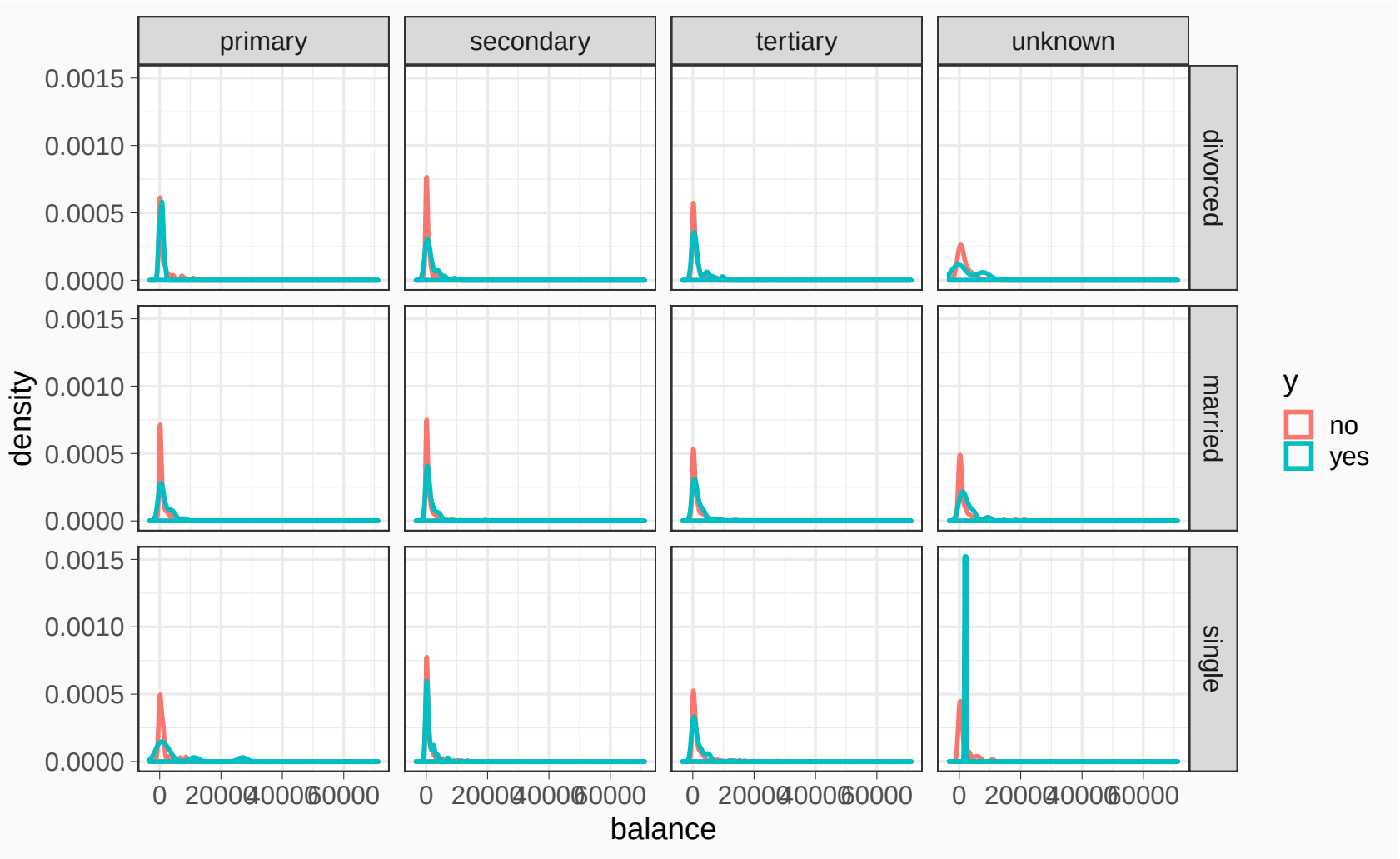
Example

```
bank2 %>% ggplot(aes(balance)) + geom_density() + facet_grid(marital~education)
```



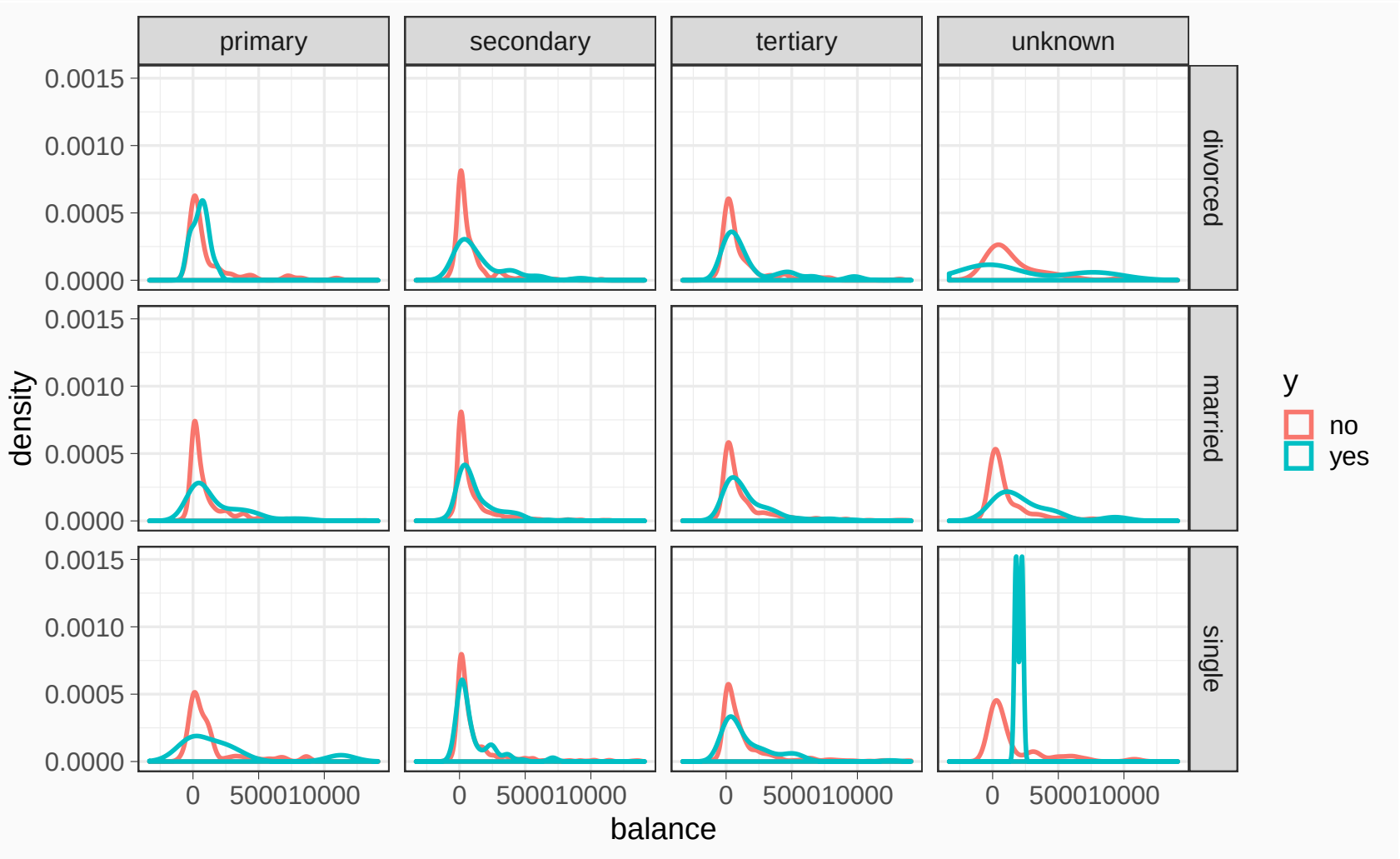
Example

```
bank %>% ggplot(aes(balance,color=y)) + geom_density() + facet_grid(marital~education)
```



Example

```
bank2 %>% ggplot(aes(balance,color=y)) + geom_density() + facet_grid(marital~education)
```



Example

More filtering

```
bank %>% group_by(marital, education) %>% summarise(nb = n()) %>%
    spread(marital, nb)
```

	education	divorced	married	single
1	primary	79	526	73
2	secondary	270	1427	609
3	tertiary	155	727	468
4	unknown	24	117	46

Example

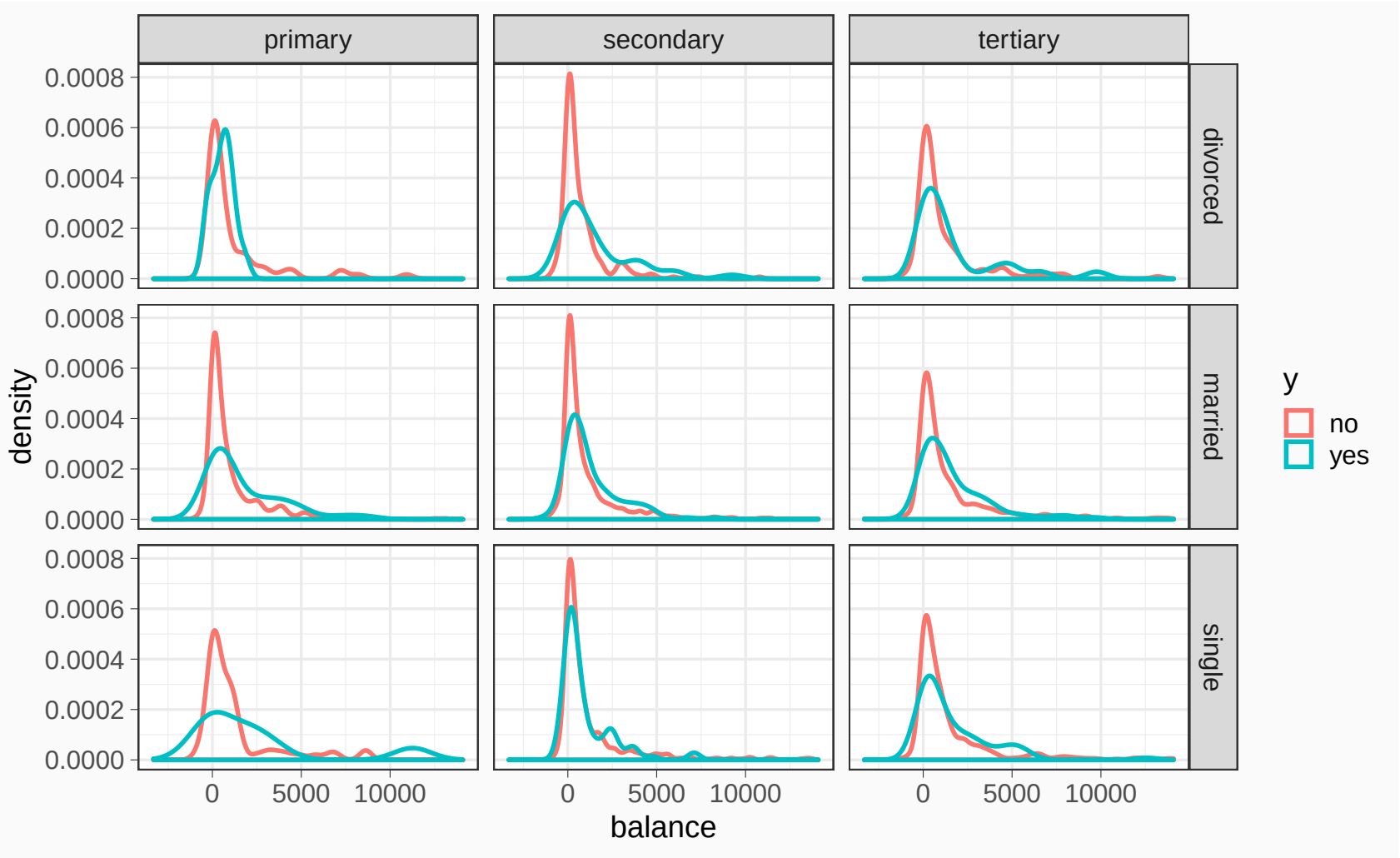
More filtering

```
bank %>% filter(balance <= 14195) %>% group_by(marital, education) %>%
    summarise(nb = n()) %>% spread(marital, nb)
```

	education	divorced	married	single
1	primary	79	523	71
2	secondary	270	1413	608
3	tertiary	154	713	460
4	unknown	24	114	46

Example

```
bank %>% filter(balance<=14195,education!="unknown") %>%
    ggplot(aes(balance,color=y)) + geom_density() + facet_grid(marital~education)
```

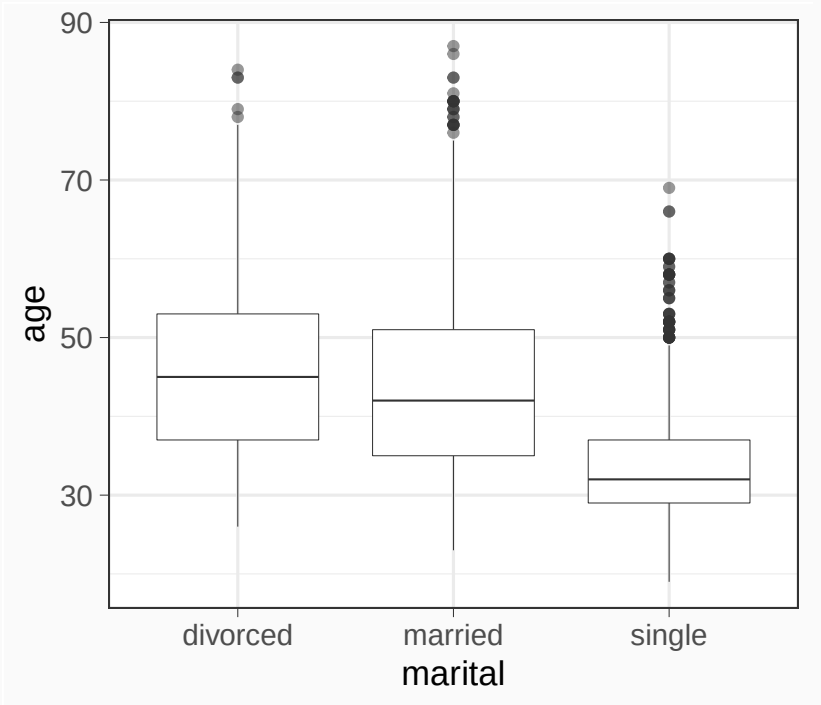


Summarizing continuous variables

Box plots

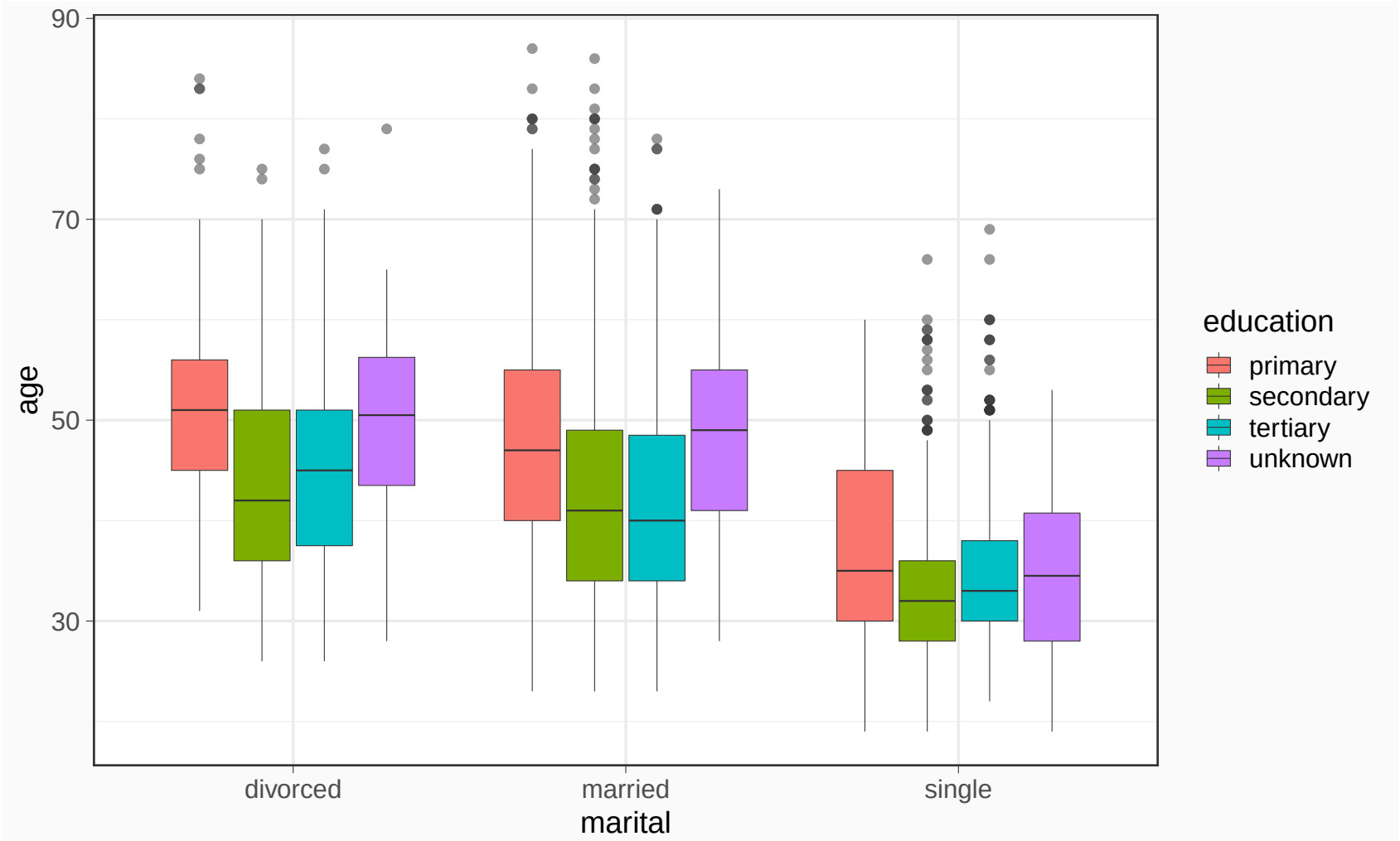
- density estimation and histogram are somewhat complex
- displaying many of them on one graphics is overwhelming
- simpler solution: box plots
- `geom_boxplot` (`stat_boxplot`)

```
ggplot(bank,aes(marital,age)) +
  geom_boxplot(outlier.size=0.5,size=0.1,
               outlier.alpha=0.5)
```



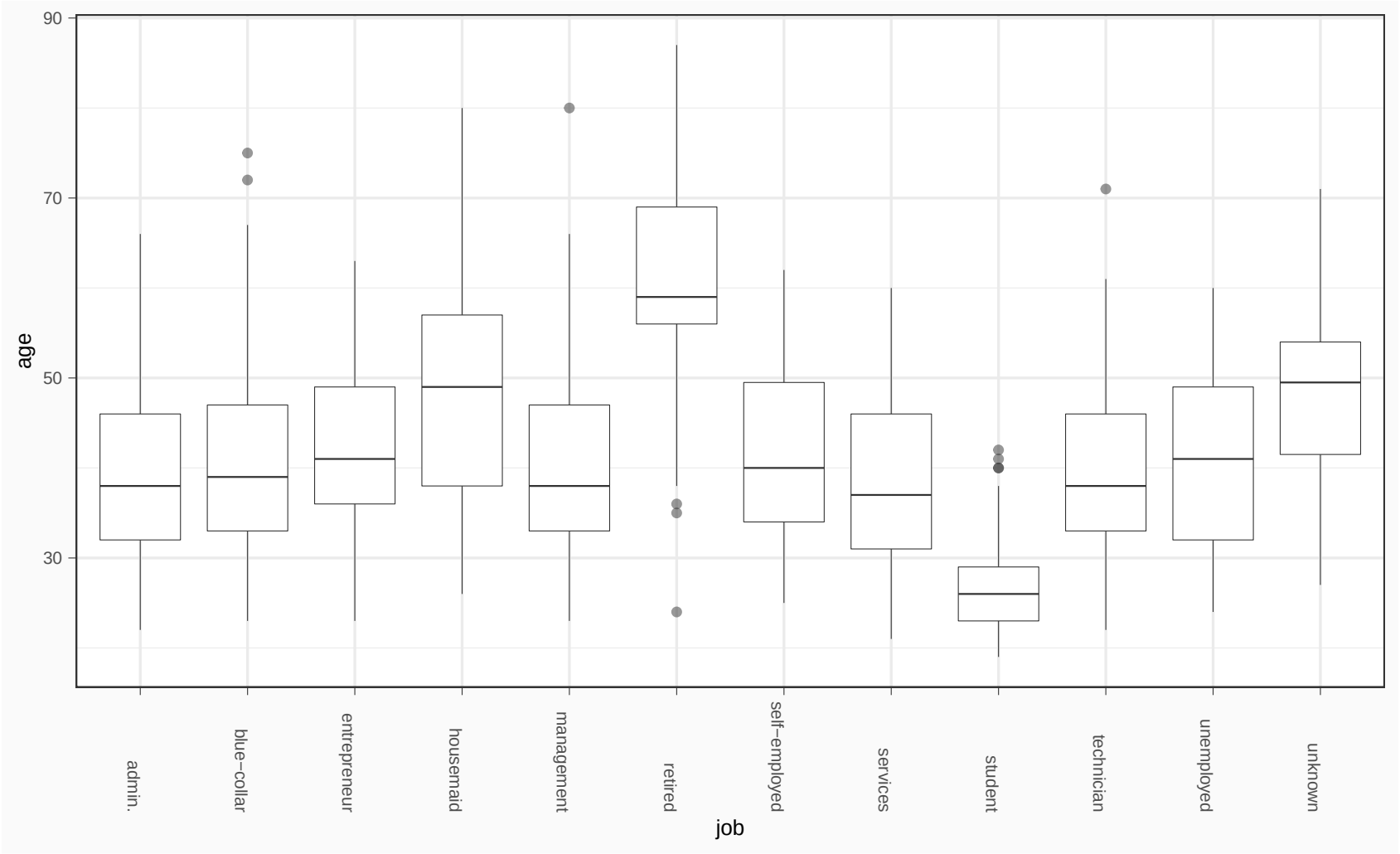
Example

```
ggplot(bank,aes(marital,age,fill=education)) +
    geom_boxplot(outlier.size=0.5,size=0.1,outlier.alpha=0.5)
```



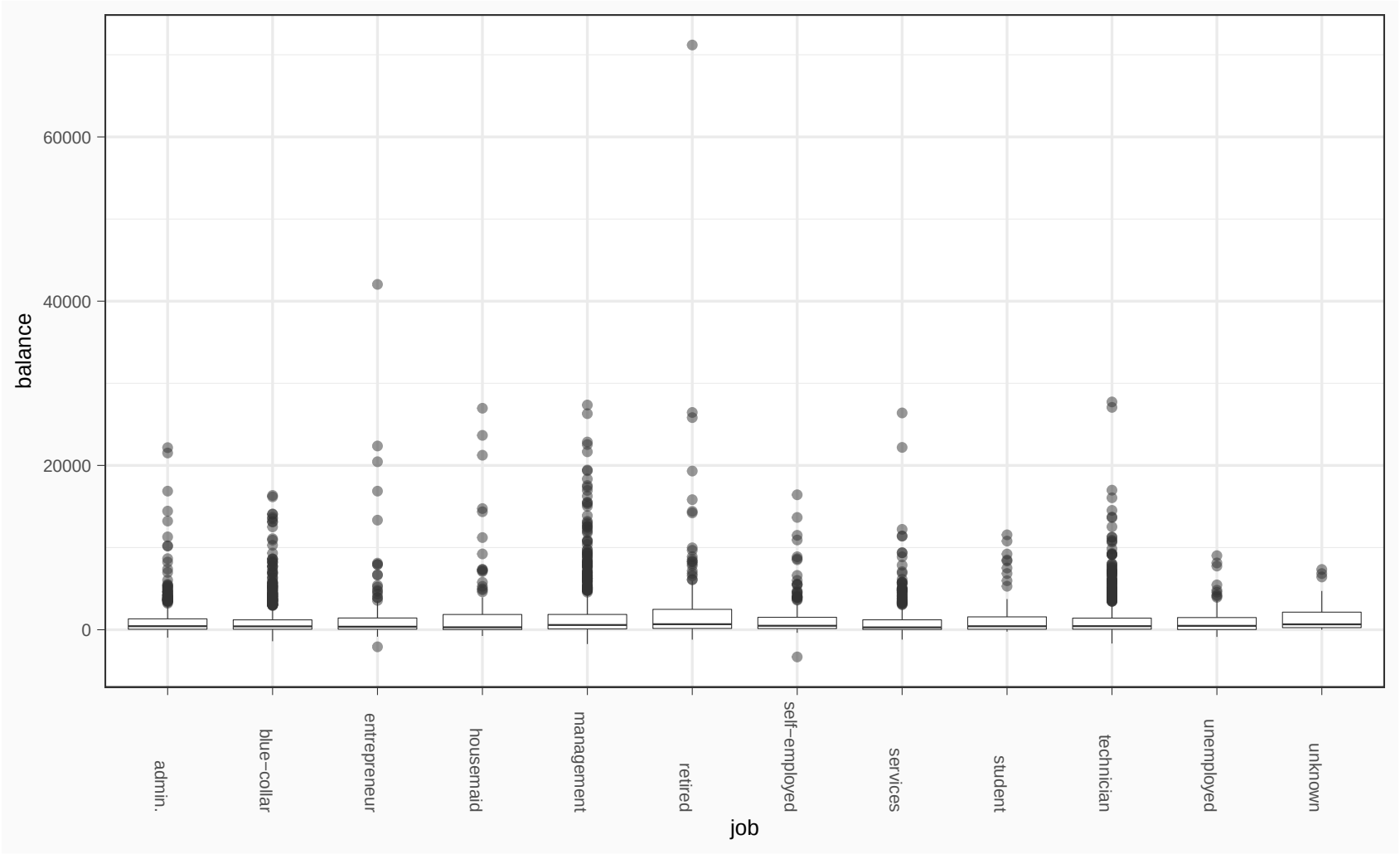
Example

```
ggplot(bank,aes(job,age)) + geom_boxplot(outlier.size=0.5,size=0.1,outlier.alpha=0.5) +
    theme(text=element_text(size=5),axis.text.x = element_text(angle = 270, hjust = 1))
```



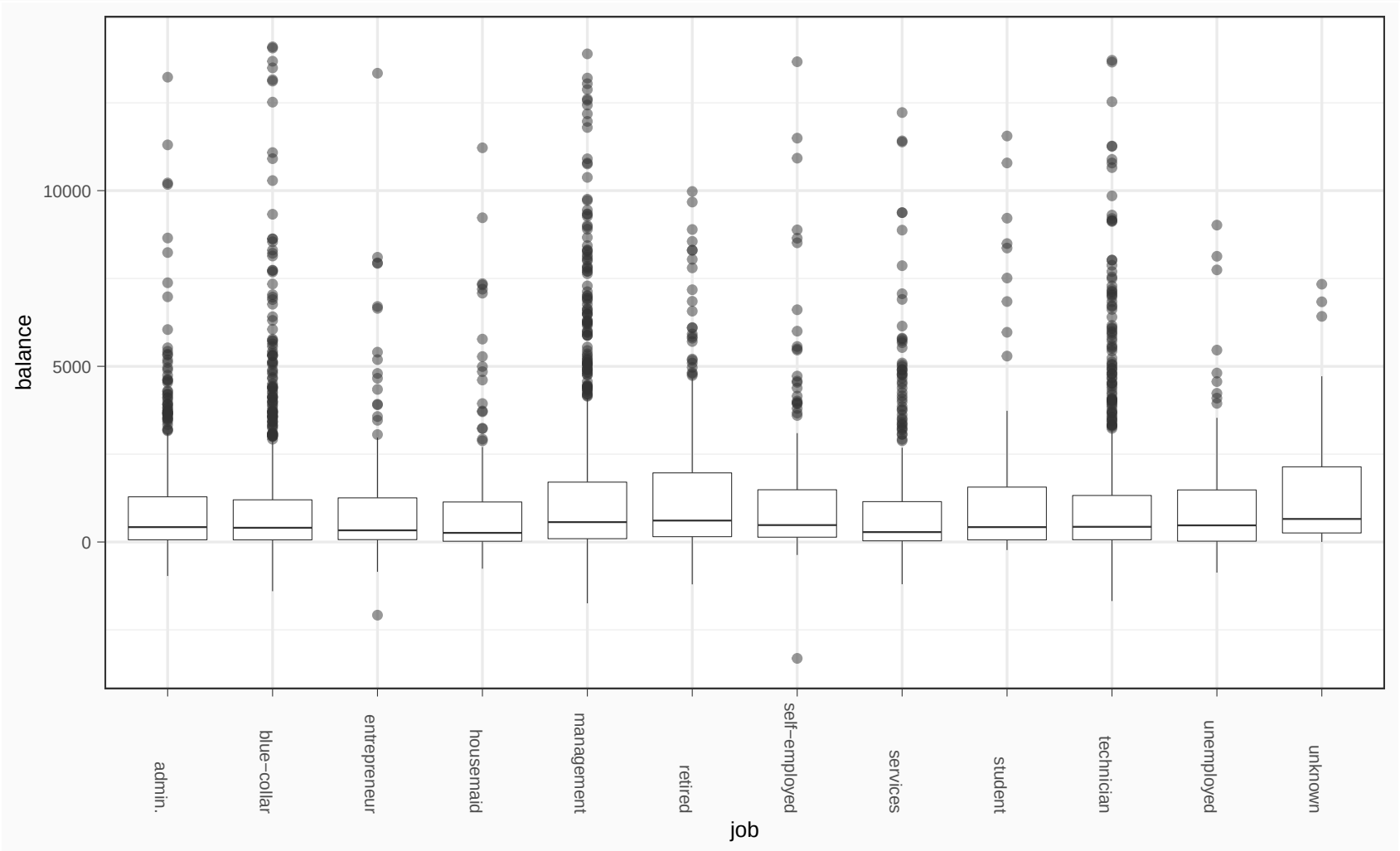
Example

```
ggplot(bank,aes(job,balance)) + geom_boxplot(outlier.size=0.5,size=0.1,outlier.alpha=0.5) +
    theme(text=element_text(size=5),axis.text.x = element_text(angle = 270, hjust = 1))
```



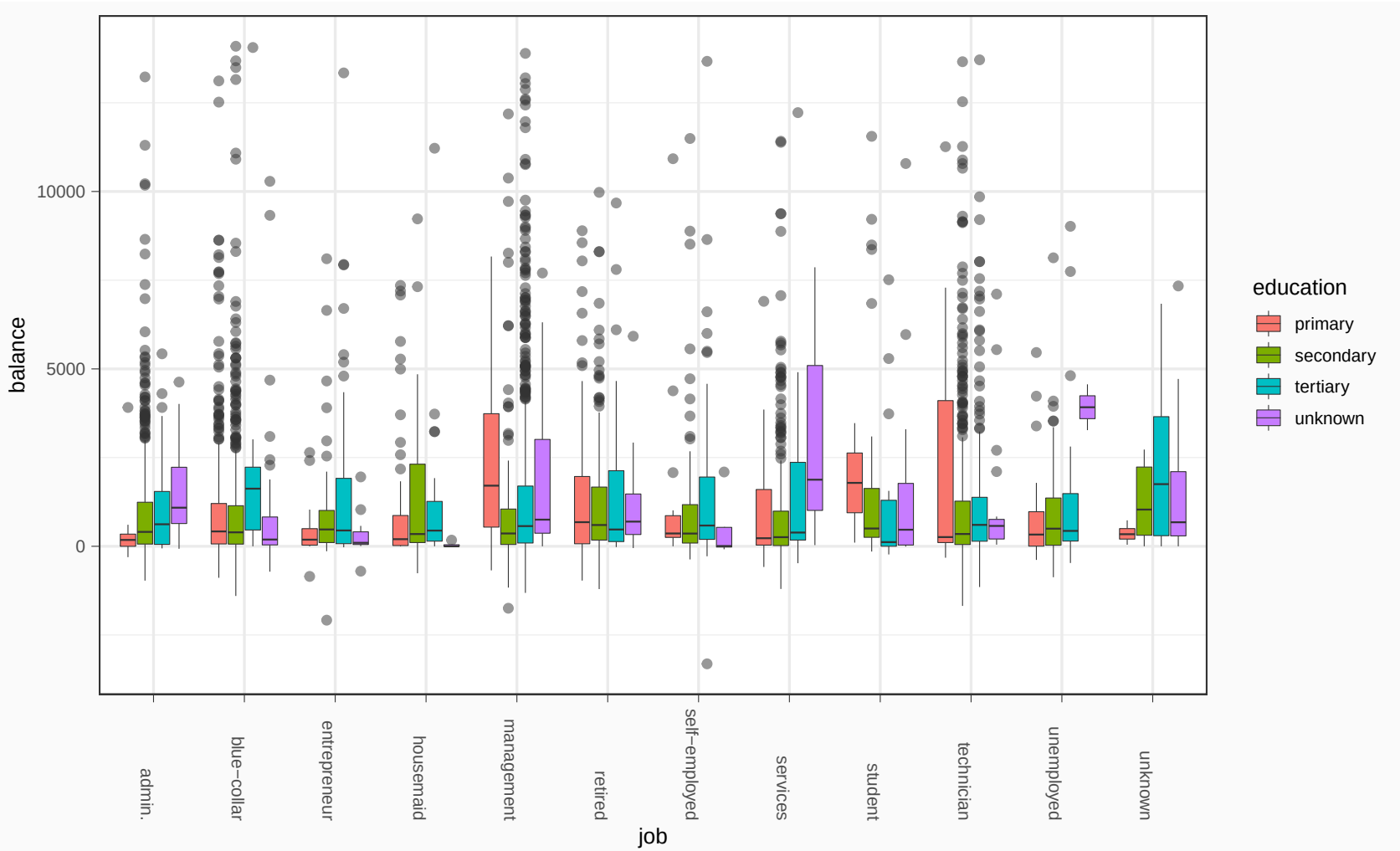
Example

```
bank %>% filter(balance<=14195) %>%
    ggplot(aes(job,balance)) + geom_boxplot(outlier.size=0.5,size=0.1,outlier.alpha=0.5) +
    theme(text=element_text(size=5),axis.text.x = element_text(angle = 270, hjust = 1))
```



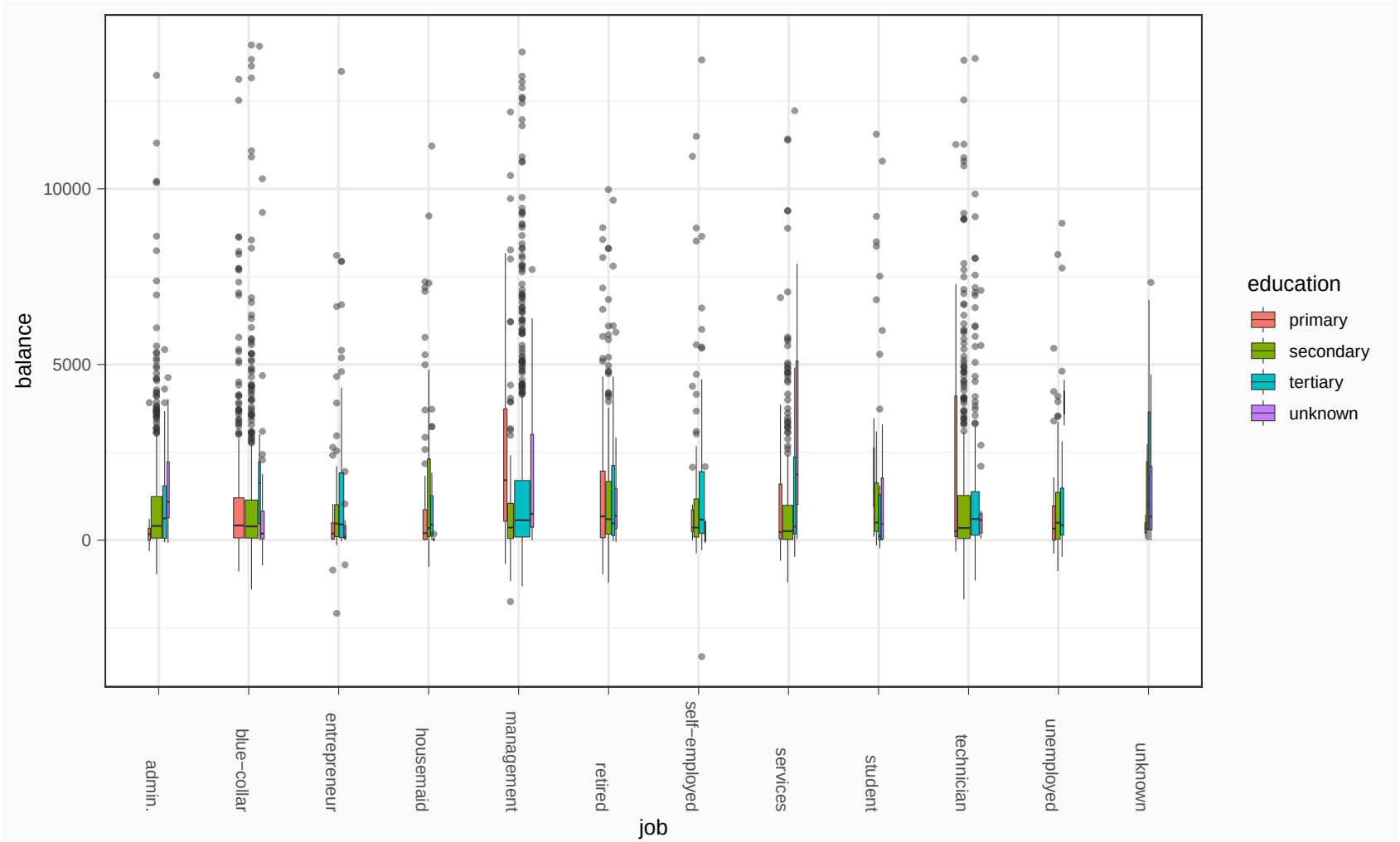
Example

```
bank %>% filter(balance<=14195) %>%
    ggplot(aes(job,balance,fill=education)) +
    geom_boxplot(outlier.size=0.5,size=0.1,outlier.alpha=0.5) +
    theme(text=element_text(size=5),axis.text.x = element_text(angle = 270, hjust = 1))
```



Example

```
bank %>% filter(balance<=14195) %>%
    ggplot(aes(job,balance,fill=education)) +
    geom_boxplot(outlier.size=0.1,size=0.1,outlier.alpha=0.5,varwidth=TRUE) +
    theme(text=element_text(size=5), axis.text.x = element_text(angle = 270, hjust = 1))
```

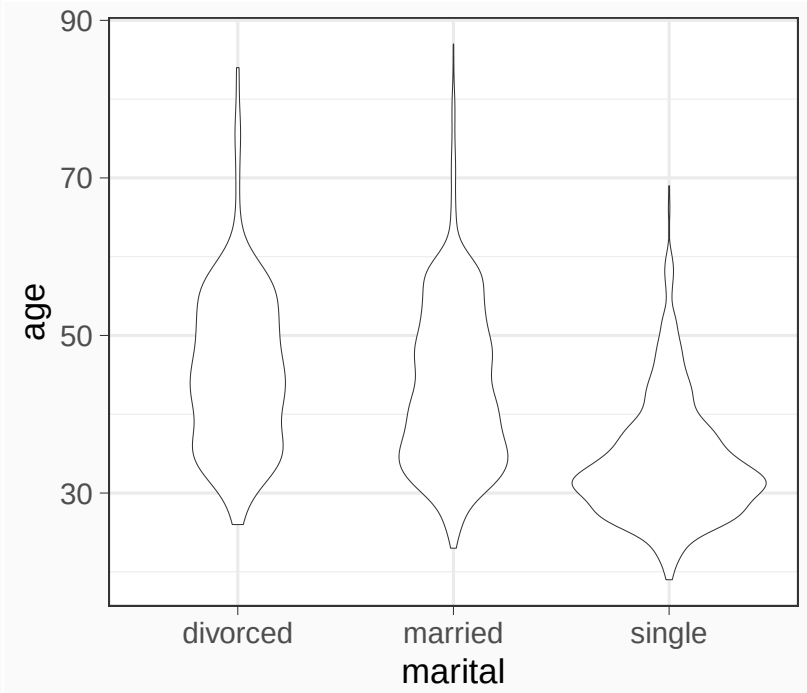


Summarizing continuous variables

Violin plots

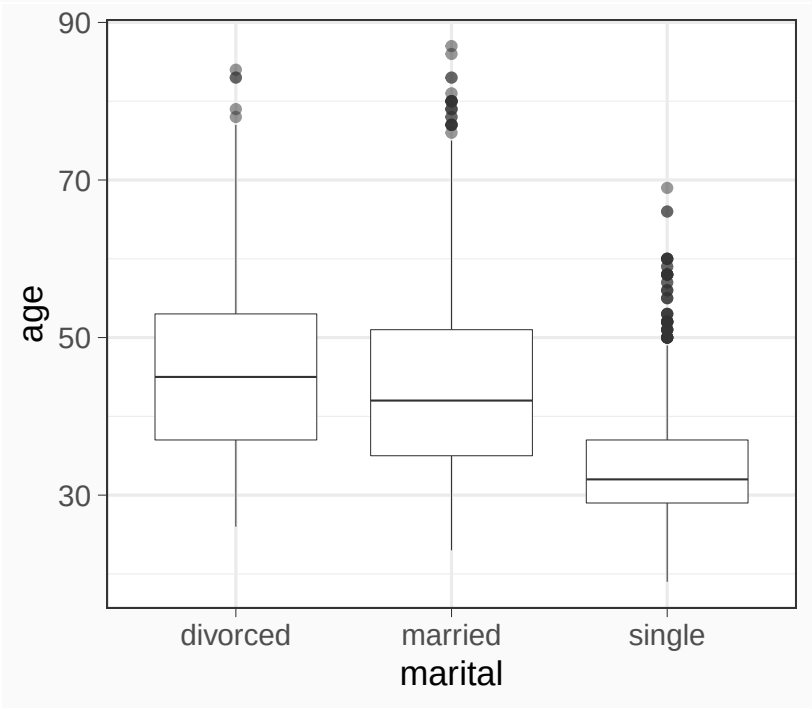
- compromise between box plots and density plots
- conditional approach as box plot
- density visualization as density plot
- `geom_violin` (`stat_ydensity`)

```
ggplot(bank,aes(marital,age)) +
  geom_violin(size=0.1)
```

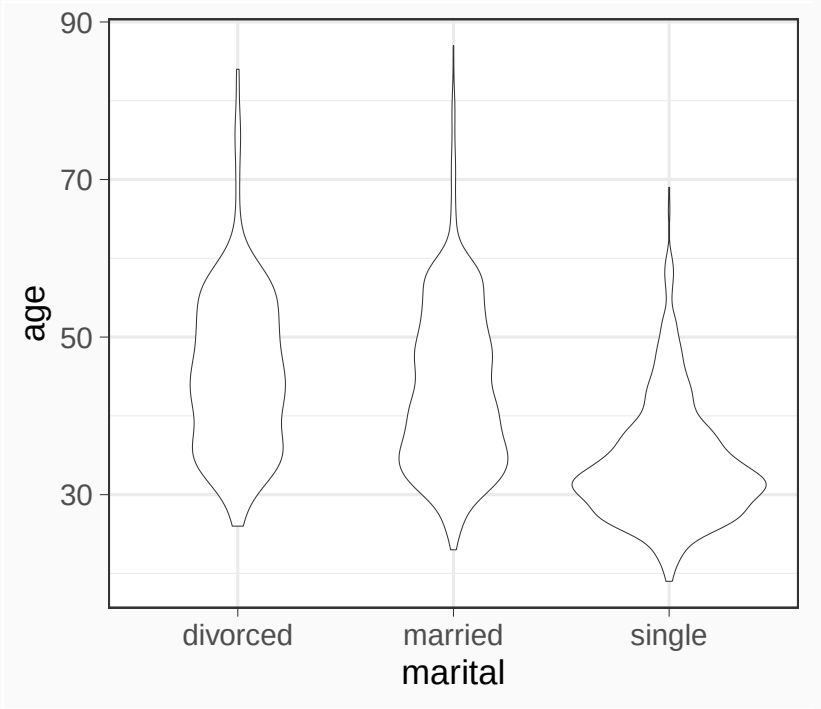


Comparison

```
ggplot(bank,aes(marital,age)) +
  geom_boxplot(outlier.size=0.5,size=0.1,
               outlier.alpha=0.5)
```

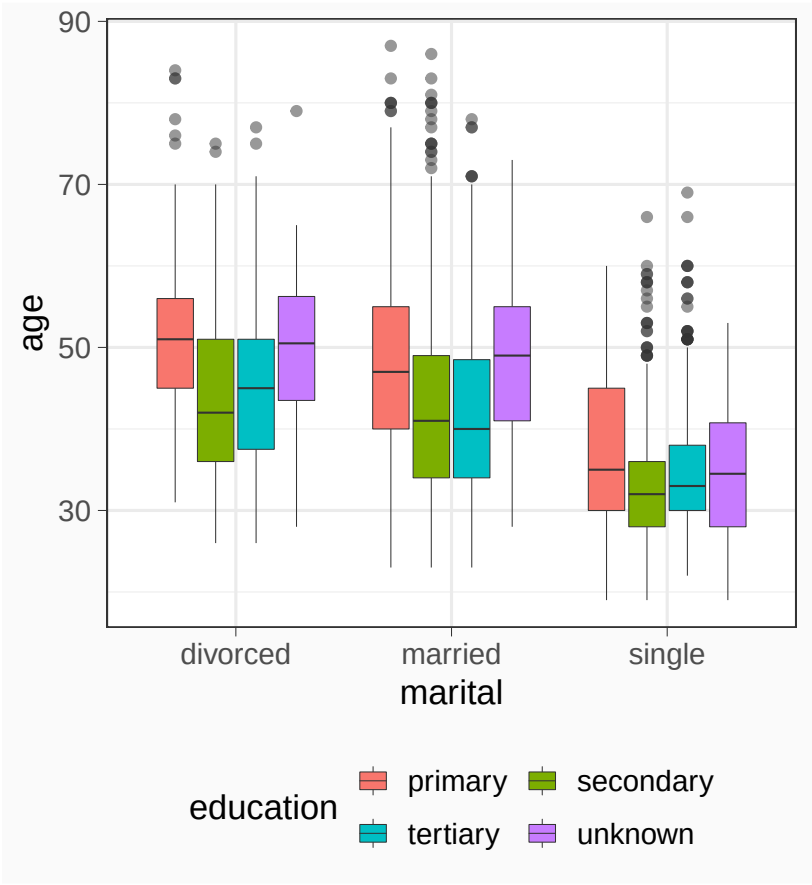



```
ggplot(bank,aes(marital,age)) +
  geom_violin(size=0.1)
```

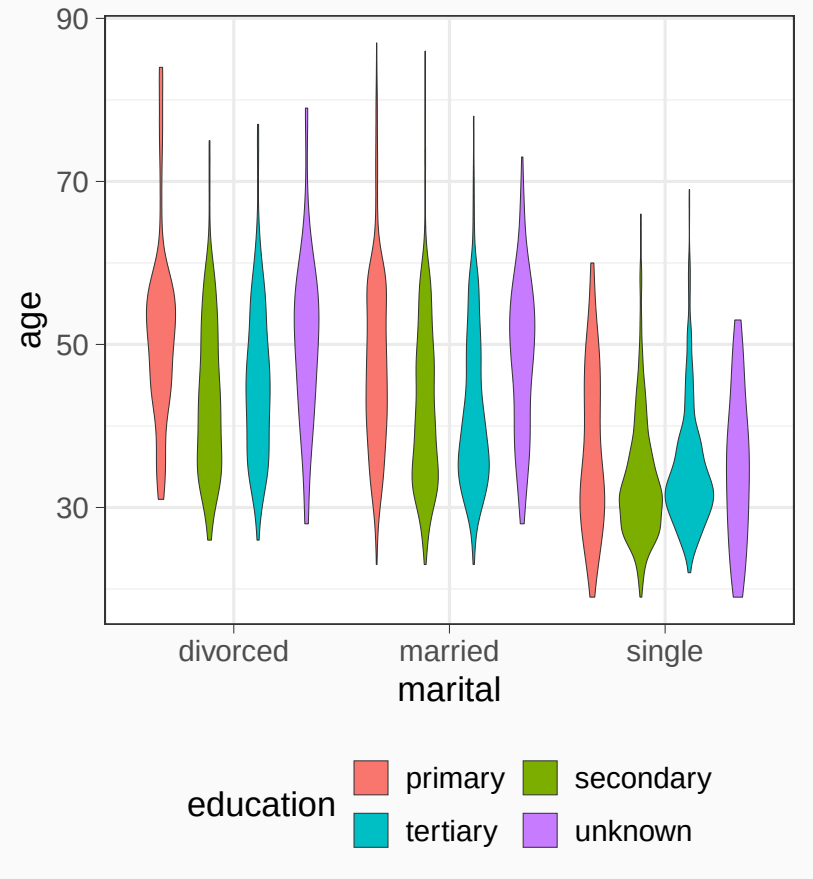


Comparison

```
ggplot(bank,aes(marital,age,fill=education)) +
  geom_boxplot(outlier.size=0.5,size=0.1,
               outlier.alpha=0.5) +
  theme(legend.position="bottom") +
  guides(fill=guide_legend(nrow=2,byrow=TRUE))
```

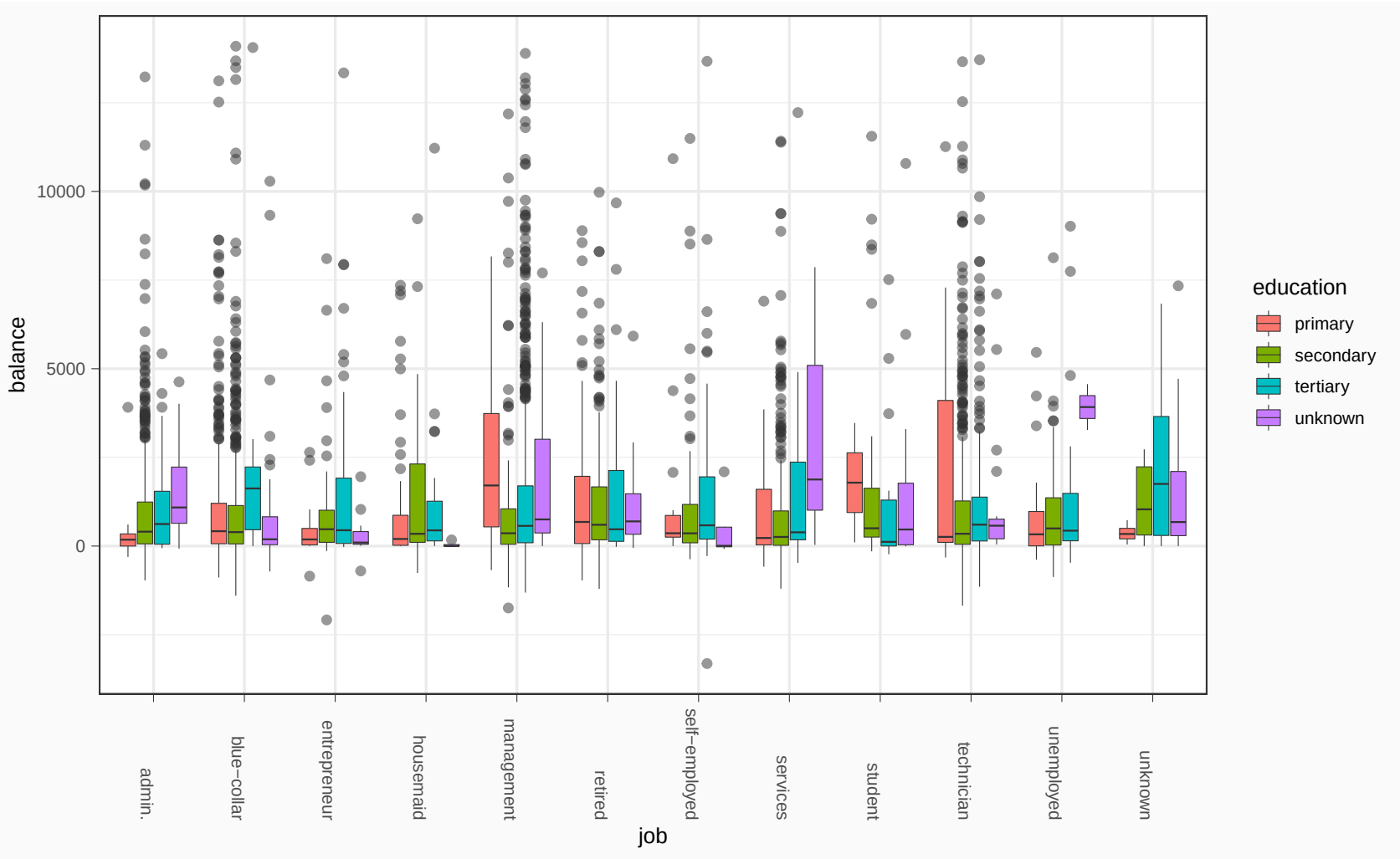



```
ggplot(bank,aes(marital,age,fill=education)) +
  geom_violin(size=0.1) +
  theme(legend.position="bottom") +
  guides(fill=guide_legend(nrow=2,byrow=TRUE))
```



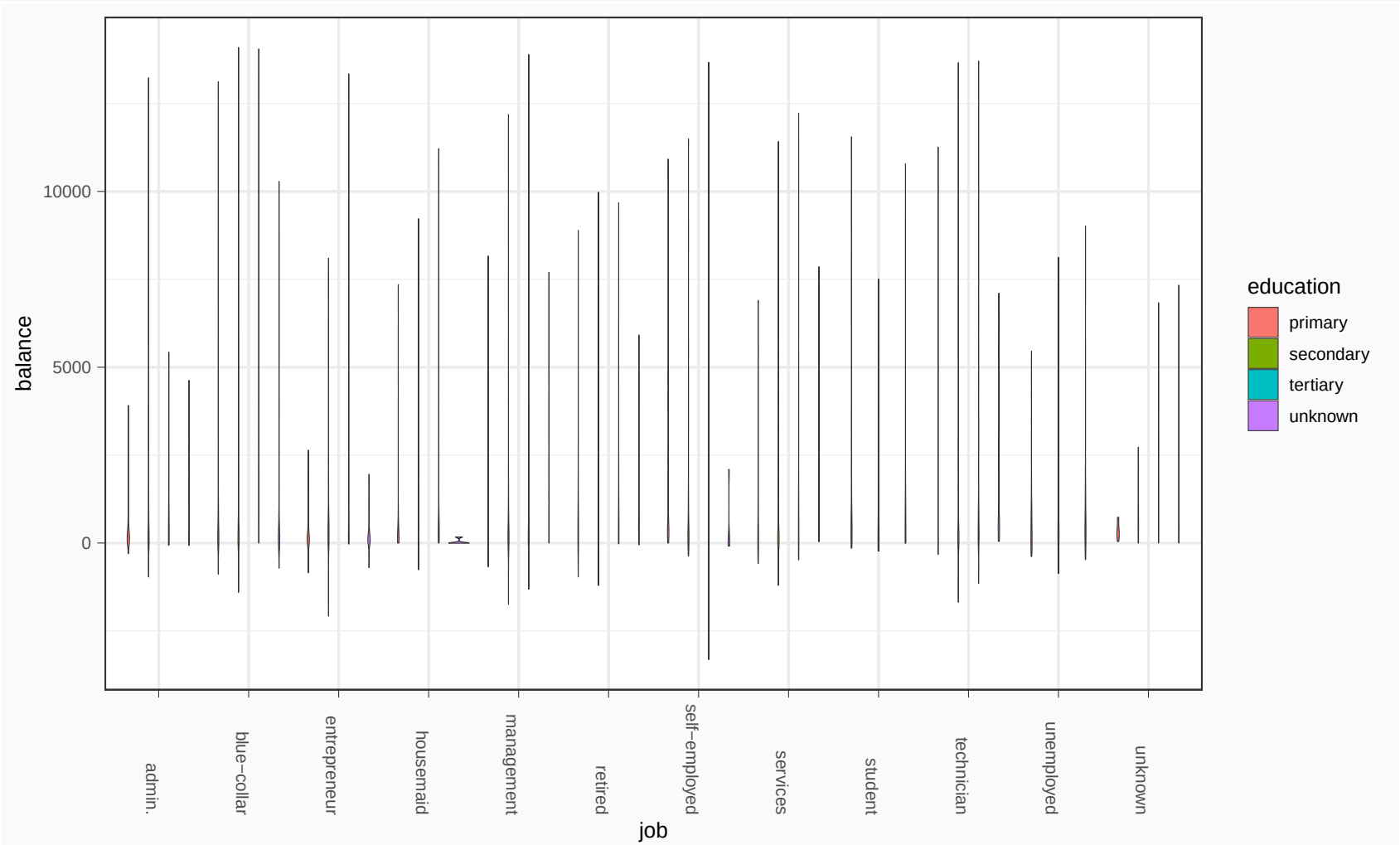
Comparison

```
bank %>% filter(balance<=14195) %>%
    ggplot(aes(job,balance,fill=education)) +
    geom_boxplot(outlier.size=0.5,size=0.1,outlier.alpha=0.5) +
    theme(text=element_text(size=5),axis.text.x = element_text(angle = 270, hjust = 1))
```



Comparison

```
bank %>% filter(balance<=14195) %>%
    ggplot(aes(job,balance,fill=education)) +
    geom_violin(size=0.1) +
    theme(text=element_text(size=5),axis.text.x = element_text(angle = 270, hjust = 1))
```



Temporal data

Time series

- many data have a temporal component
- temporal data are easily represented by continuous lines
- must be in an adapted format for ggplot
  - ggplot operates on variables in the data set
  - one variable has to be the time mapped to the x axis
  - any other variable can be time evolving quantity

Example: air quality data set

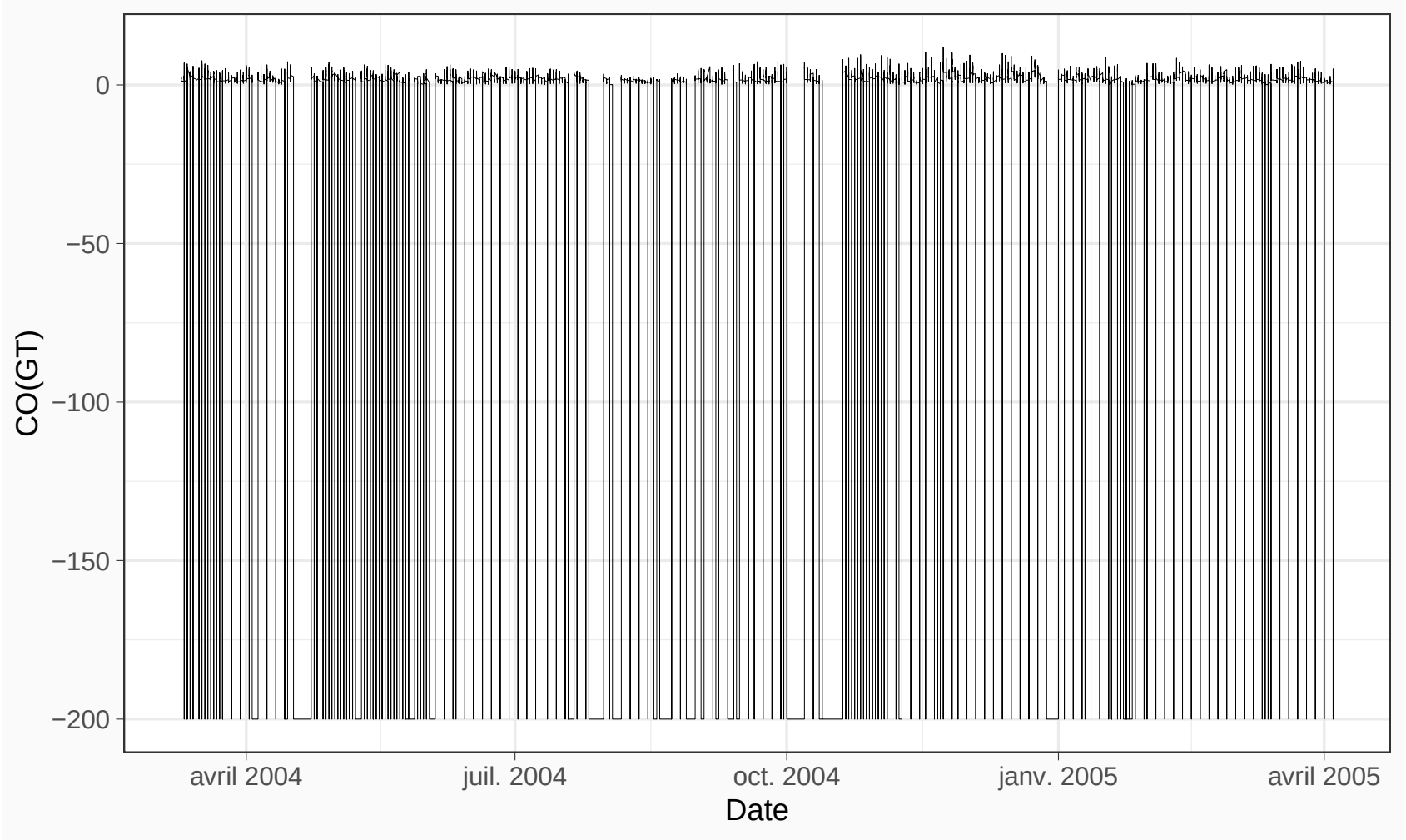
- Air+Quality
- hourly average of reading from chemical sensors
- hourly average of gas concentrations

Example

```
airquality
## # A tibble: 9,357 x 15
##    Date                Time                `CO(GT)` `PT08.S1(CO)` `NMHC(GT)` `C6H6(GT)`
##    <dttm>              <dttm>                 <dbl>         <dbl>      <dbl>      <dbl>
##  1 2004-03-10 00:00:00 1899-12-31 18:00:00      2.6          1360        150       11.9
##  2 2004-03-10 00:00:00 1899-12-31 19:00:00      2            1292.       112        9.40
##  3 2004-03-10 00:00:00 1899-12-31 20:00:00      2.2          1402         88        9.00
##  4 2004-03-10 00:00:00 1899-12-31 21:00:00      2.2          1376.        80        9.23
##  5 2004-03-10 00:00:00 1899-12-31 22:00:00      1.6          1272.        51        6.52
##  6 2004-03-10 00:00:00 1899-12-31 23:00:00      1.2          1197         38        4.74
##  7 2004-03-11 00:00:00 1899-12-31 00:00:00      1.2          1185         31        3.62
##  8 2004-03-11 00:00:00 1899-12-31 01:00:00      1            1136.        31        3.33
##  9 2004-03-11 00:00:00 1899-12-31 02:00:00      0.9          1094         24        2.34
## 10 2004-03-11 00:00:00 1899-12-31 03:00:00      0.6          1010.        19        1.70
## # ... with 9,347 more rows, and 9 more variables: `PT08.S2(NMHC)` <dbl>, `NOx(GT)` <dbl>,
## #   `PT08.S3(NOx)` <dbl>, `NO2(GT)` <dbl>, `PT08.S4(NO2)` <dbl>, `PT08.S5(O3)` <dbl>, T <dbl>,
## #   RH <dbl>, AH <dbl>
```

Example

```
ggplot(airquality, aes(Date, `CO(GT)`)) + geom_line(size = 0.02)
```



Missing data

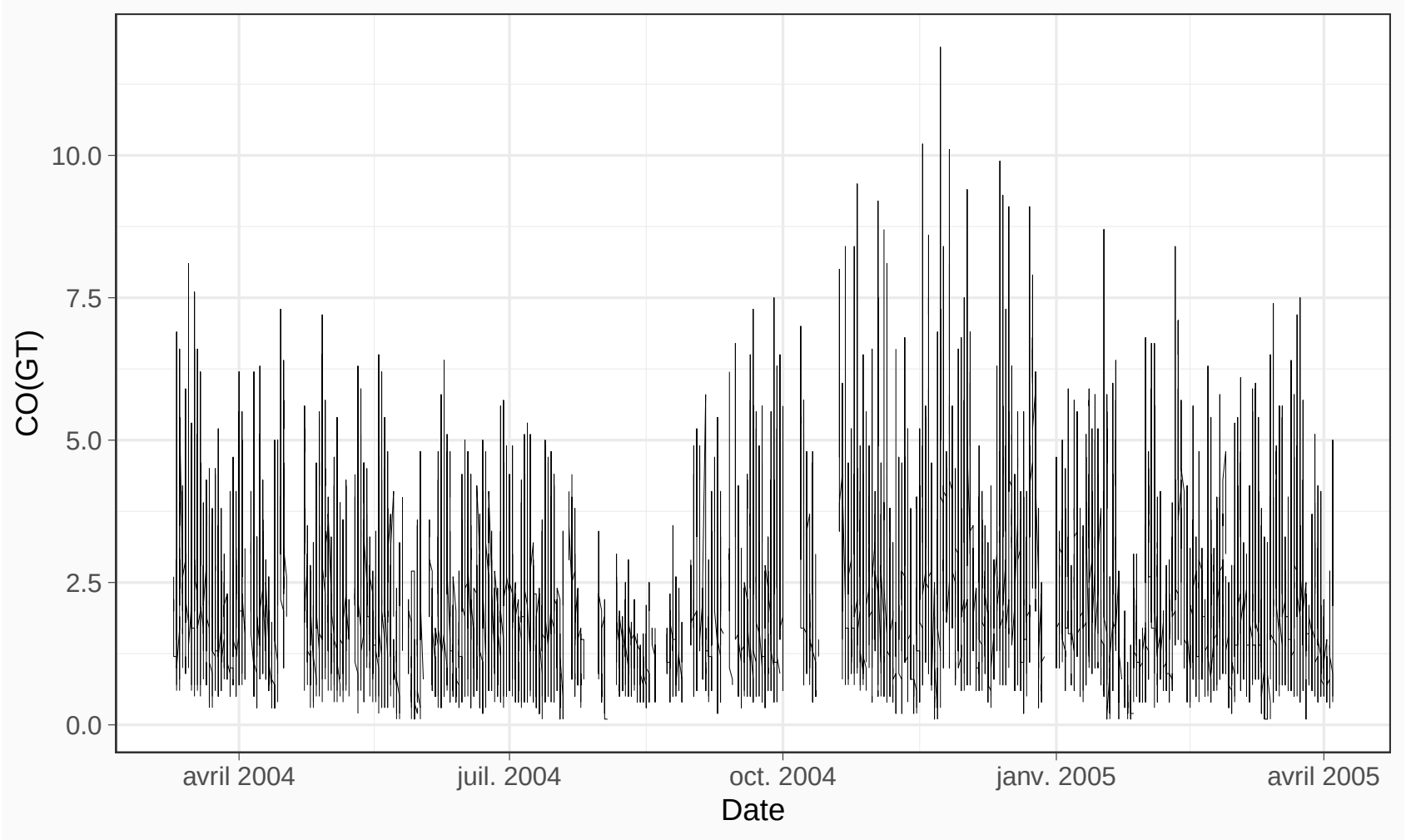
Outliers

- numerous occurrences of -200
- impossible according to the data model
- poor encoding of missing values!

```
airquality <- airquality %>% mutate_all(~replace(., . == -200, NA))
```

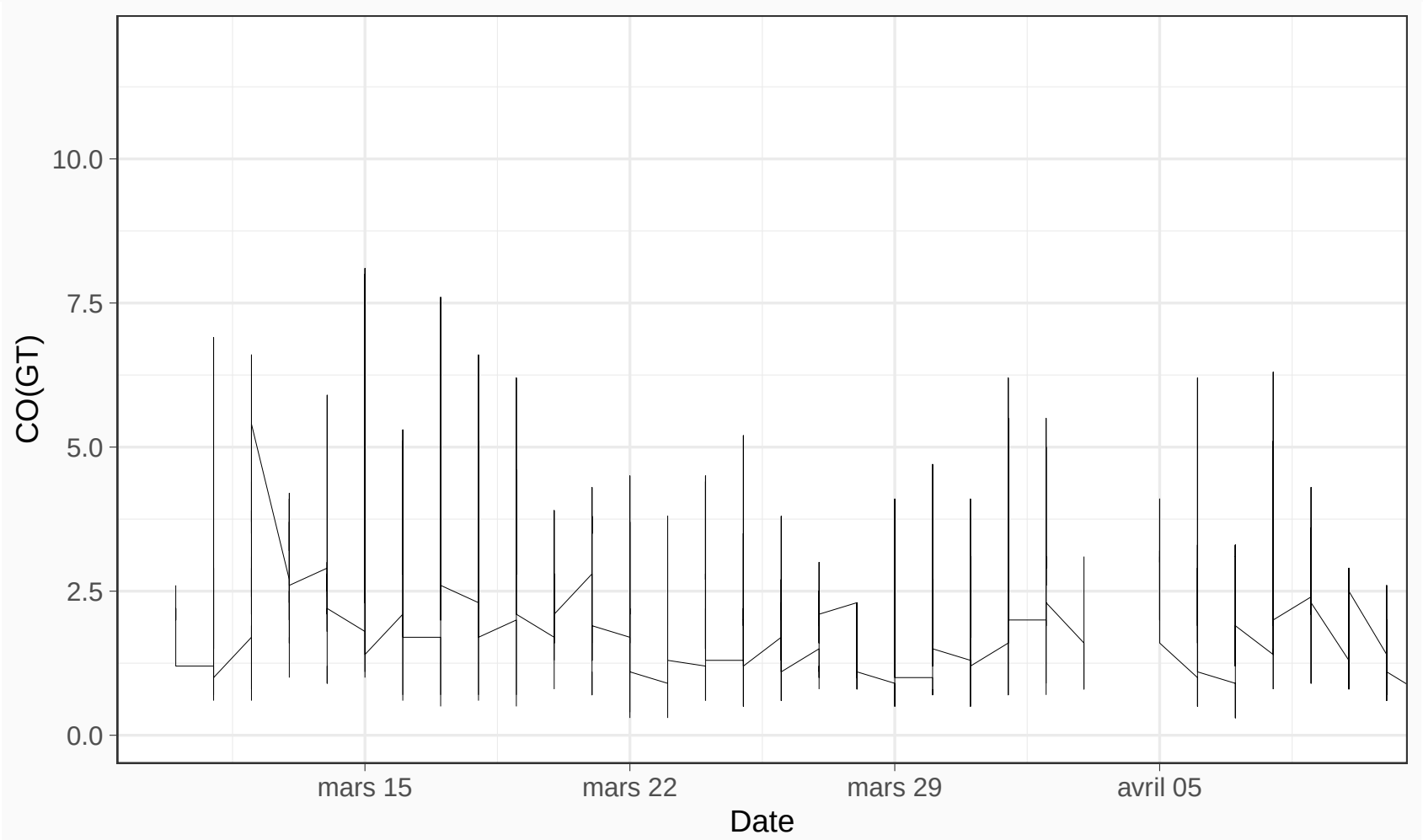
Example

```
ggplot(airquality, aes(Date, `CO(GT)`)) + geom_line(size = 0.02)
```



Zooming

```
ggplot(airquality,aes(Date,`CO(GT)`))+geom_line(size=0.02)+
    coord_cartesian(xlim=as_datetime(c("2004-03-10"," 2004-04-10")))
```



Encoding issues

```
airquality %>% select(Date, Time) %>% print(n = 5, n_extra = 0)
## # A tibble: 9,357 x 2
##   Date                Time
##   <dttm>              <dttm>
## 1 2004-03-10 00:00:00 1899-12-31 18:00:00
## 2 2004-03-10 00:00:00 1899-12-31 19:00:00
## 3 2004-03-10 00:00:00 1899-12-31 20:00:00
## 4 2004-03-10 00:00:00 1899-12-31 21:00:00
## 5 2004-03-10 00:00:00 1899-12-31 22:00:00
## # ... with 9,352 more rows
```

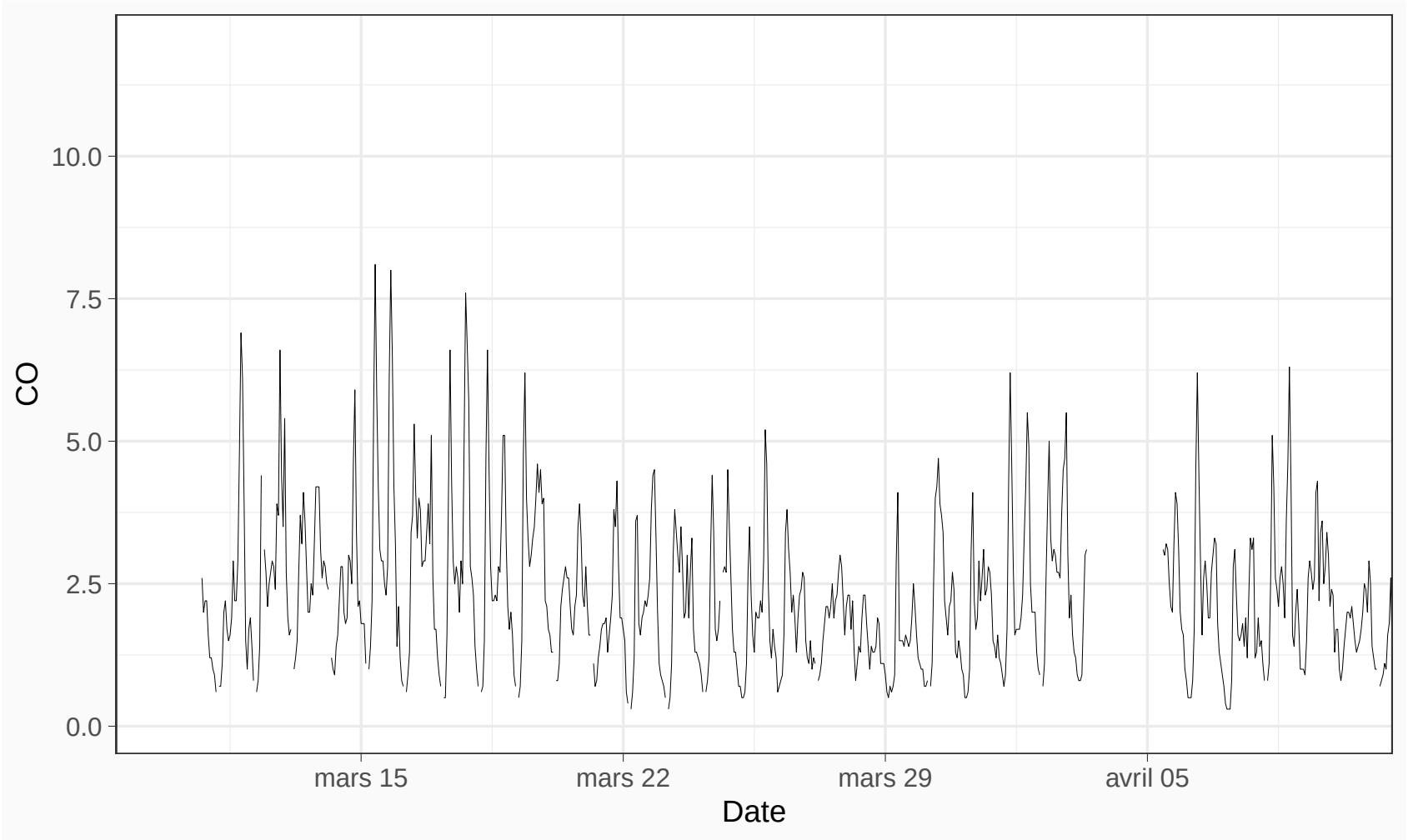
Encoding issues

```
airquality %>% select(Date, Time) %>% print(n = 5, n_extra = 0)
## # A tibble: 9,357 x 2
##   Date                Time
##   <dttm>              <dttm>
## 1 2004-03-10 00:00:00 1899-12-31 18:00:00
## 2 2004-03-10 00:00:00 1899-12-31 19:00:00
## 3 2004-03-10 00:00:00 1899-12-31 20:00:00
## 4 2004-03-10 00:00:00 1899-12-31 21:00:00
## 5 2004-03-10 00:00:00 1899-12-31 22:00:00
## # ... with 9,352 more rows
```

```
airquality <- airquality %>% mutate(fulldate = Date + hour(Time) * 3600)
airquality %>% select(Date, Time, fulldate) %>% print(n = 5, n_extra = 0)
## # A tibble: 9,357 x 3
##   Date                Time                fulldate
##   <dttm>              <dttm>              <dttm>
## 1 2004-03-10 00:00:00 1899-12-31 18:00:00 2004-03-10 18:00:00
## 2 2004-03-10 00:00:00 1899-12-31 19:00:00 2004-03-10 19:00:00
## 3 2004-03-10 00:00:00 1899-12-31 20:00:00 2004-03-10 20:00:00
## 4 2004-03-10 00:00:00 1899-12-31 21:00:00 2004-03-10 21:00:00
## 5 2004-03-10 00:00:00 1899-12-31 22:00:00 2004-03-10 22:00:00
## # ... with 9,352 more rows
```

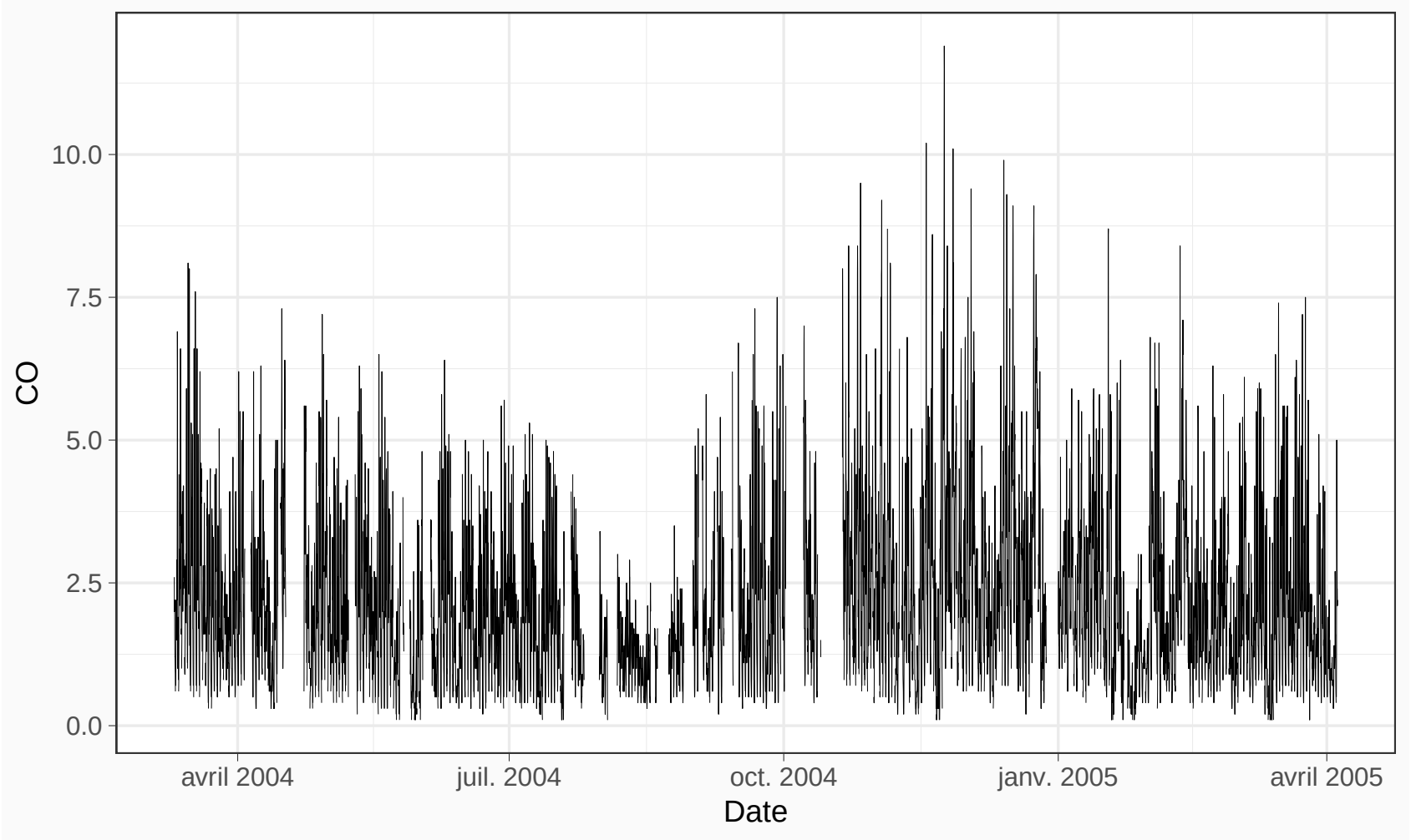
Corrected data

```
ggplot(airquality,aes(fulldate,`CO(GT)`))+geom_line(size=0.02)+
    coord_cartesian(xlim=as_datetime(c("2004-03-10"," 2004-04-10"))) +
        labs(x="Date", y="CO")
```



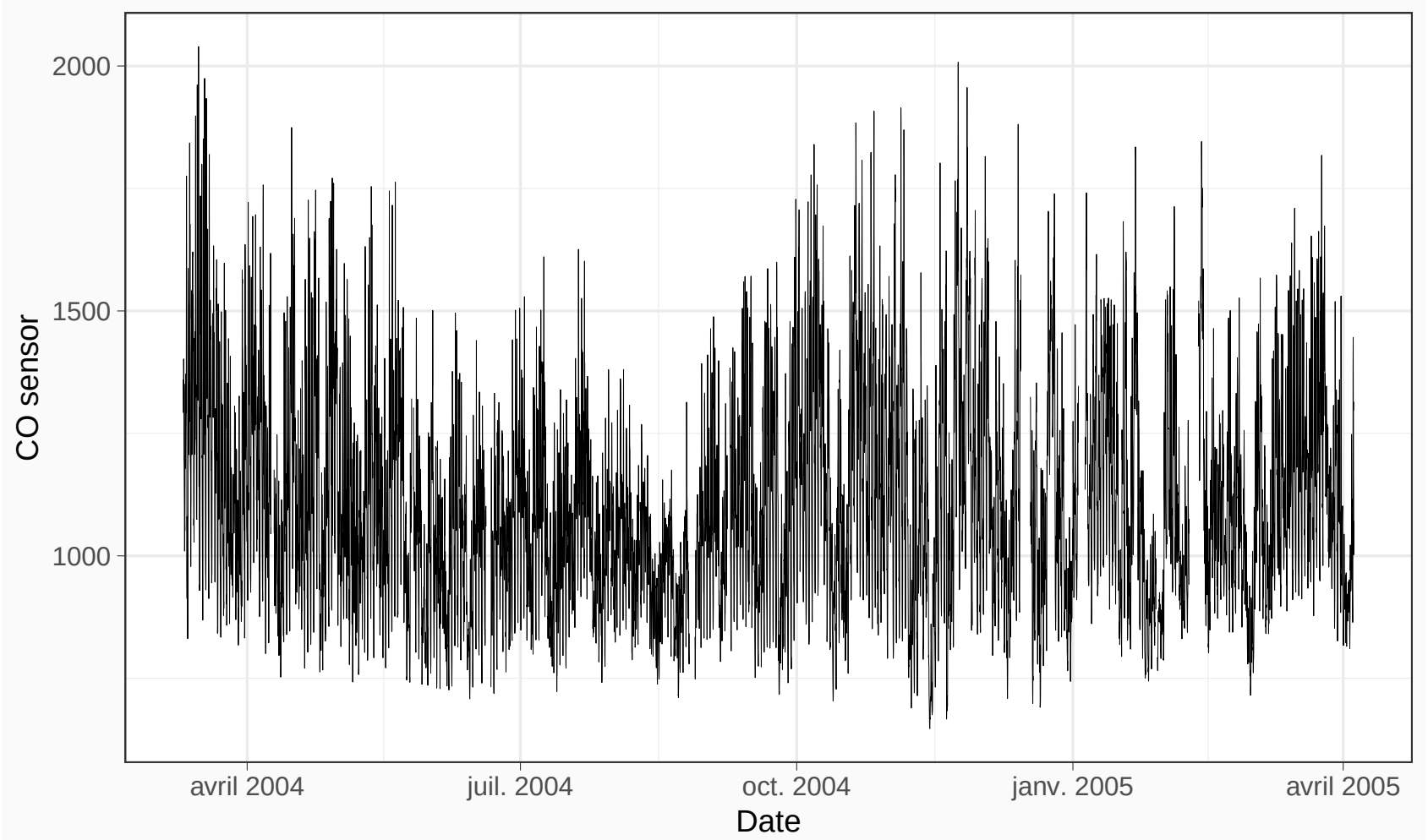
Corrected data

```
ggplot(airquality,aes(fulldate,`CO(GT)`))+geom_line(size=0.02) +
        labs(x="Date", y="CO")
```



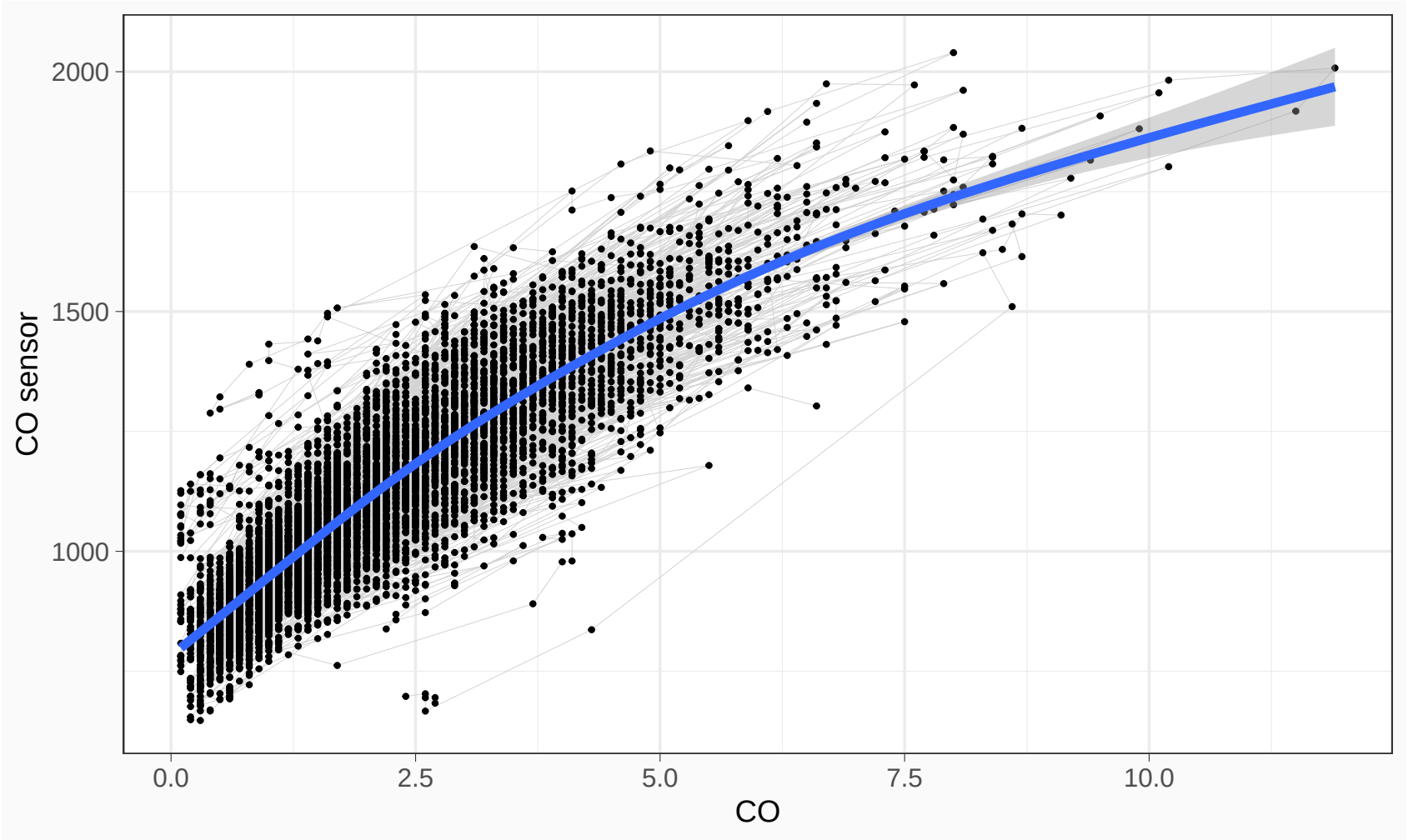
Another series

```
ggplot(airquality,aes(fulldate,`PT08.S1(CO)`))+geom_line(size=0.02) +
        labs(x="Date", y="CO sensor")
```



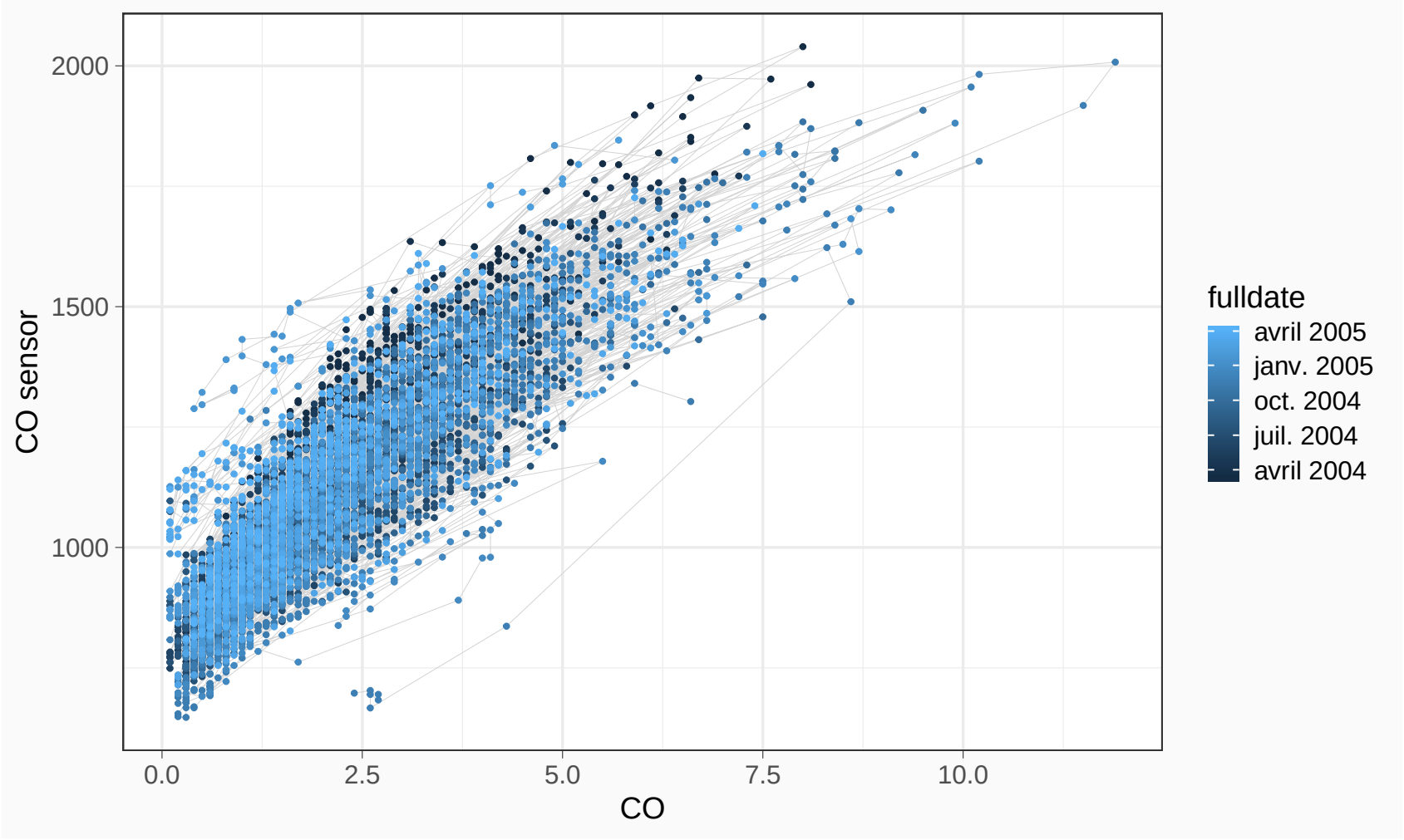
Trajectory

```
ggplot(airquality,aes(`CO(GT)`,`PT08.S1(CO)`))+
    geom_path(size=0.02, color="lightgrey") + geom_point(size=0.1) +
    labs(x="CO", y="CO sensor") + geom_smooth()
```



Trajectory with time

```
ggplot(airquality,aes(`CO(GT)`,`PT08.S1(CO)`)) +
    geom_path(size=0.02, color="lightgrey") +
    geom_point(size=0.1,aes(color=fulldate)) + labs(x="CO", y="CO sensor")
```



Multiple time series

Groups

- `group` aesthetics
- used to group values in a single graphical object (e.g. a line)
- useful to display complex entities on a single graph
- typical use
  - time series!
  - each time series is a group
  - identified by a *variable*

Example

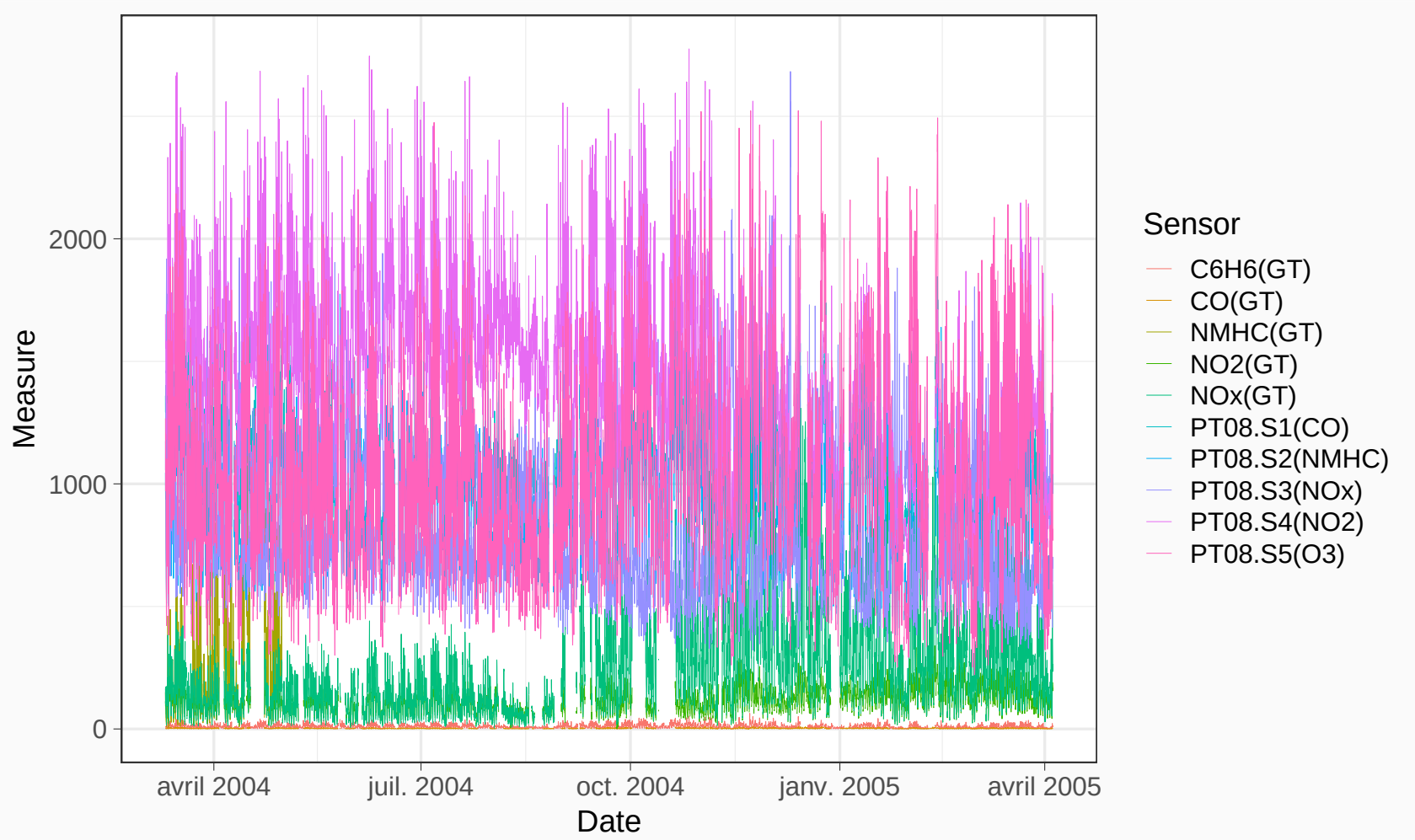
Wide to tall

- air quality date are *wide*
- entities: dates
- gas point of view: a time series per gas
- grouping on gases is impossible: one variable per gas
- turn the data to a *tall* representation: an entity is a date and a gas

```
gases <- airquality %>% select(-Date, -Time, -AH, -RH,
    -T) %>% gather(Sensor, Measure, -fulldate) %>%
    rename(Date = fulldate)
gases %>% print(n = 5)
## # A tibble: 93,570 x 3
##   Date                Sensor Measure
##   <dttm>              <chr>    <dbl>
## 1 2004-03-10 18:00:00 CO(GT)     2.6
## 2 2004-03-10 19:00:00 CO(GT)     2
## 3 2004-03-10 20:00:00 CO(GT)     2.2
## 4 2004-03-10 21:00:00 CO(GT)     2.2
## 5 2004-03-10 22:00:00 CO(GT)     1.6
## # ... with 9.356e+04 more rows
```

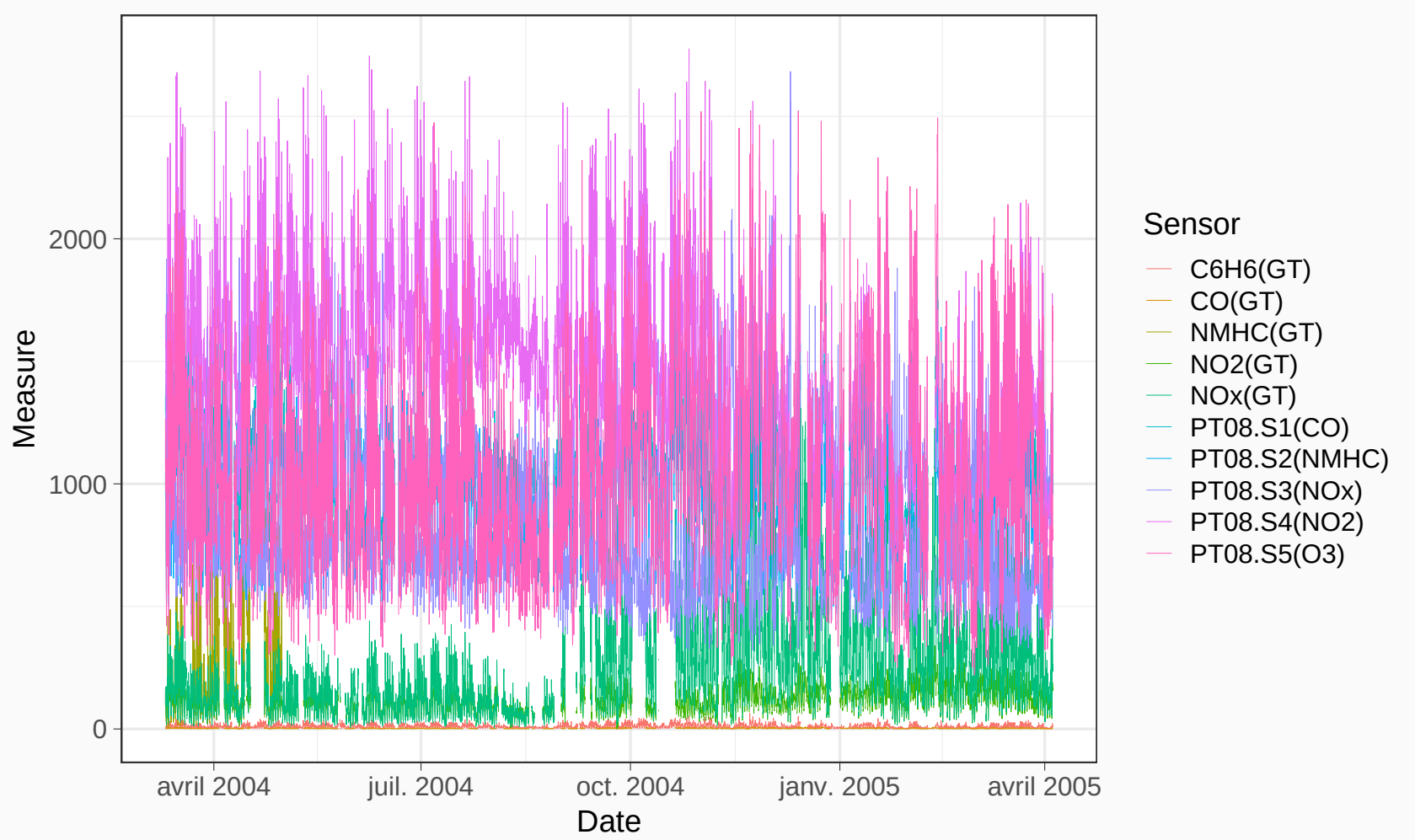
Multiple series

```
ggplot(gases, aes(Date, Measure, group = Sensor, color = Sensor)) +
    geom_line(size = 0.02)
```



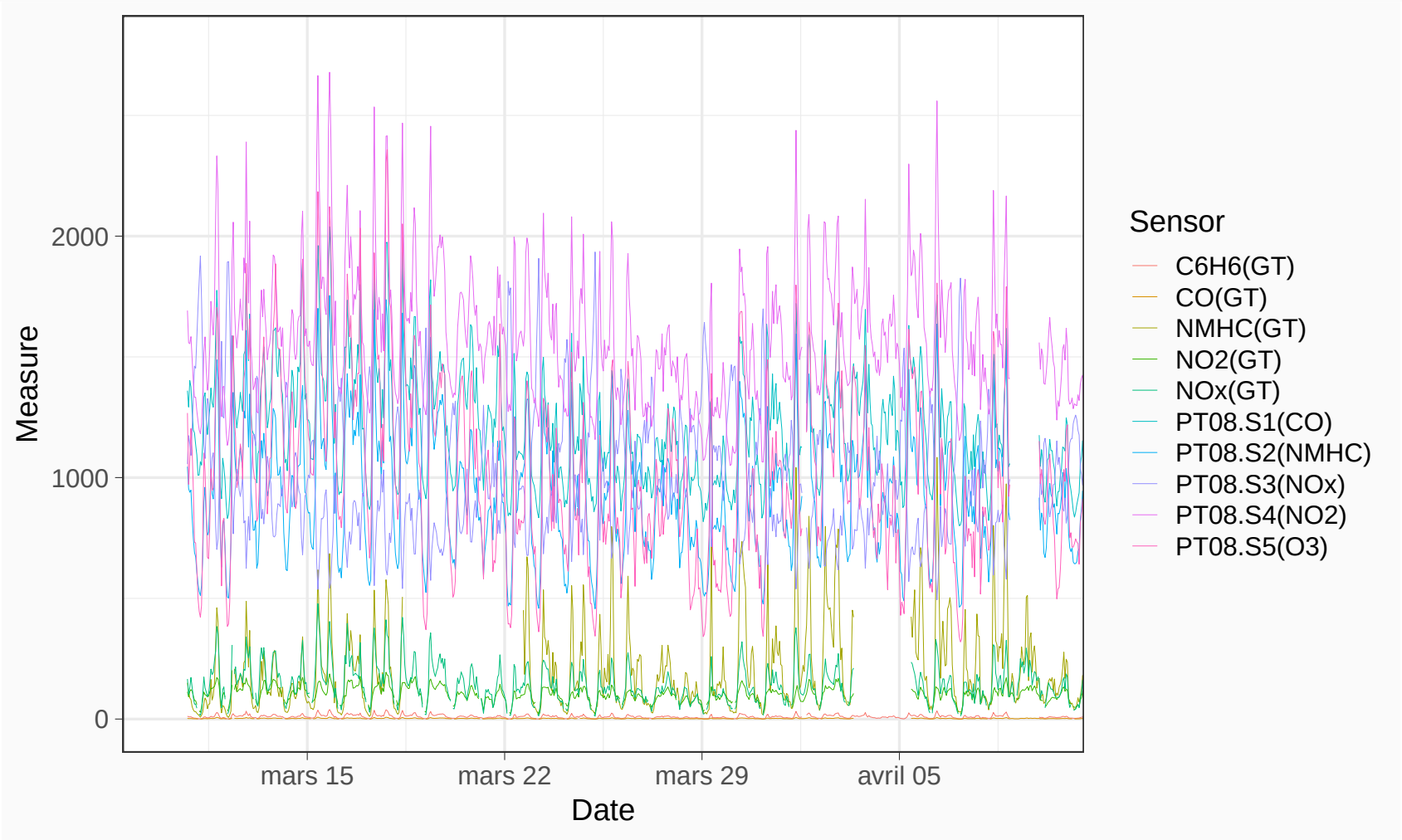
Multiple series

```
ggplot(gases, aes(Date, Measure, group = Sensor, color = Sensor)) +
    geom_line(size = 0.02)
```



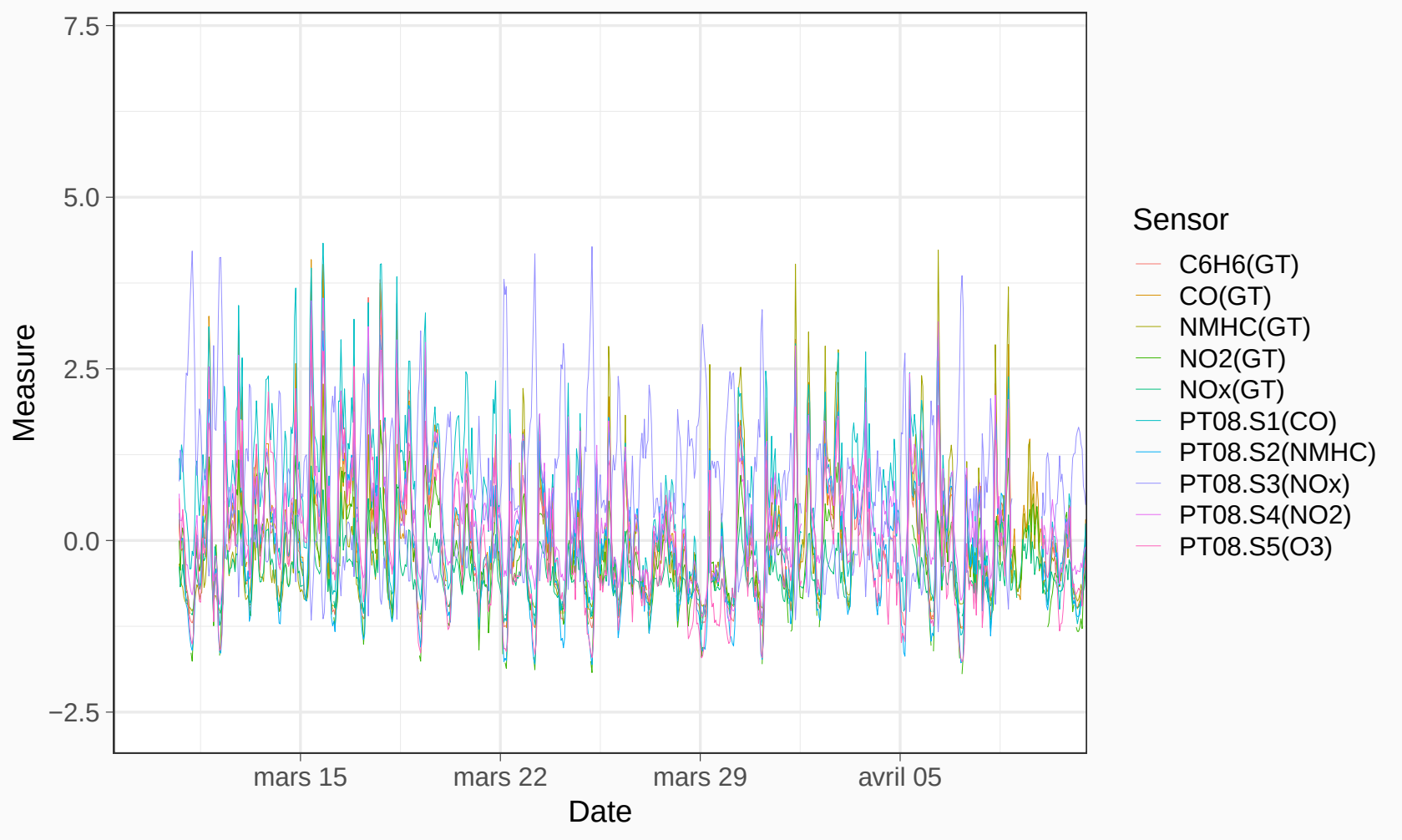
Multiple series

```
ggplot(gases, aes(Date, Measure, group = Sensor, color = Sensor)) +
    geom_line(size = 0.02) + coord_cartesian(xlim = as_datetime(c("2004-03-10",
    " 2004-04-10")))
```



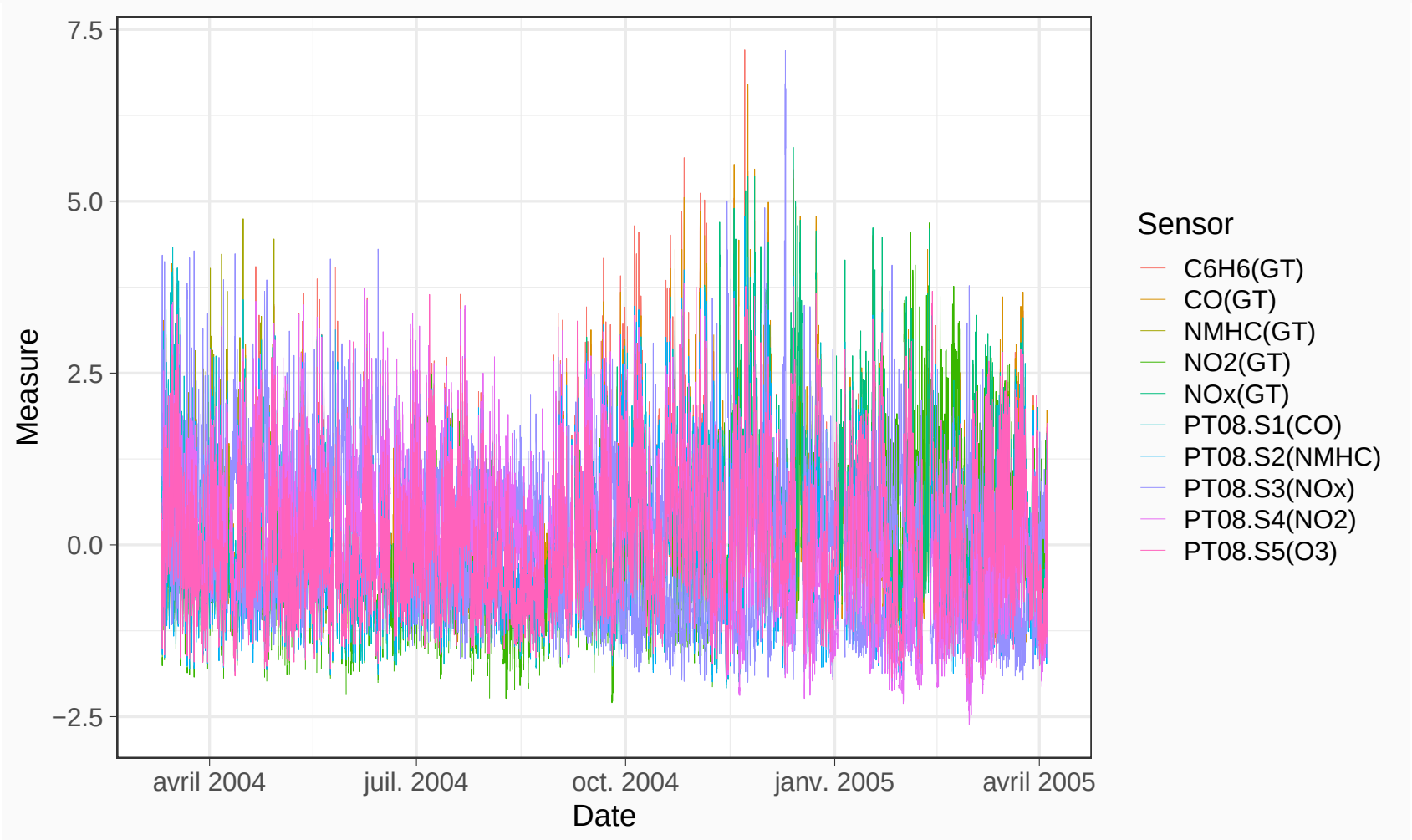
Multiple series

```
ggplot(gases.n, aes(Date, Measure, group = Sensor,
    color = Sensor)) + geom_line(size = 0.02) + coord_cartesian(xlim = as_datetime(c("2004-03-10",
    " 2004-04-10")))
```



Multiple series

```
ggplot(gases.n, aes(Date, Measure, group = Sensor,
    color = Sensor)) + geom_line(size = 0.02)
```



Example

Weekly sales data

- Sales_Transactions_Dataset_Weekly
- 811 products
- sales per week for each product

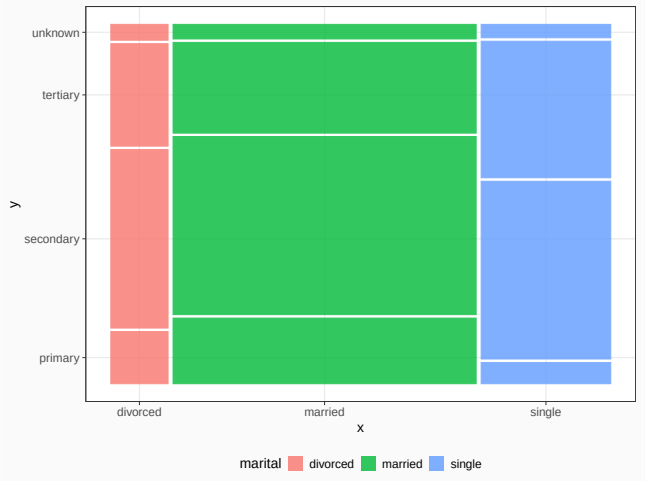
```
prodperweek %>% print(10, n_extra = 2)
## # A tibble: 811 x 53
##    Product_Code    W0    W1    W2    W3    W4    W5    W6    W7    W8    W9
##    <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 P1              11    12    10     8    13    12    14    21     6    14
##  2 P2               7     6     3     2     7     1     6     3     3     3
##  3 P3               7    11     8     9    10     8     7    13    12     6
##  4 P4              12     8    13     5     9     6     9    13    13    11
##  5 P5               8     5    13    11     6     7     9    14     9     9
##  6 P6               3     3     2     7     6     3     8     6     6     3
##  7 P7               4     8     3     7     8     7     2     3    10     3
##  8 P8               8     6    10     9     6     8     7     5    10    10
##  9 P9              14     9    10     7    11    15    12     7    13    12
## 10 P10             22    19    19    29    20    16    26    20    24    20
## # ... with 801 more rows, and 42 more variables: W10 <dbl>, W11 <dbl>, ...
```

Outline

Mosaic plots

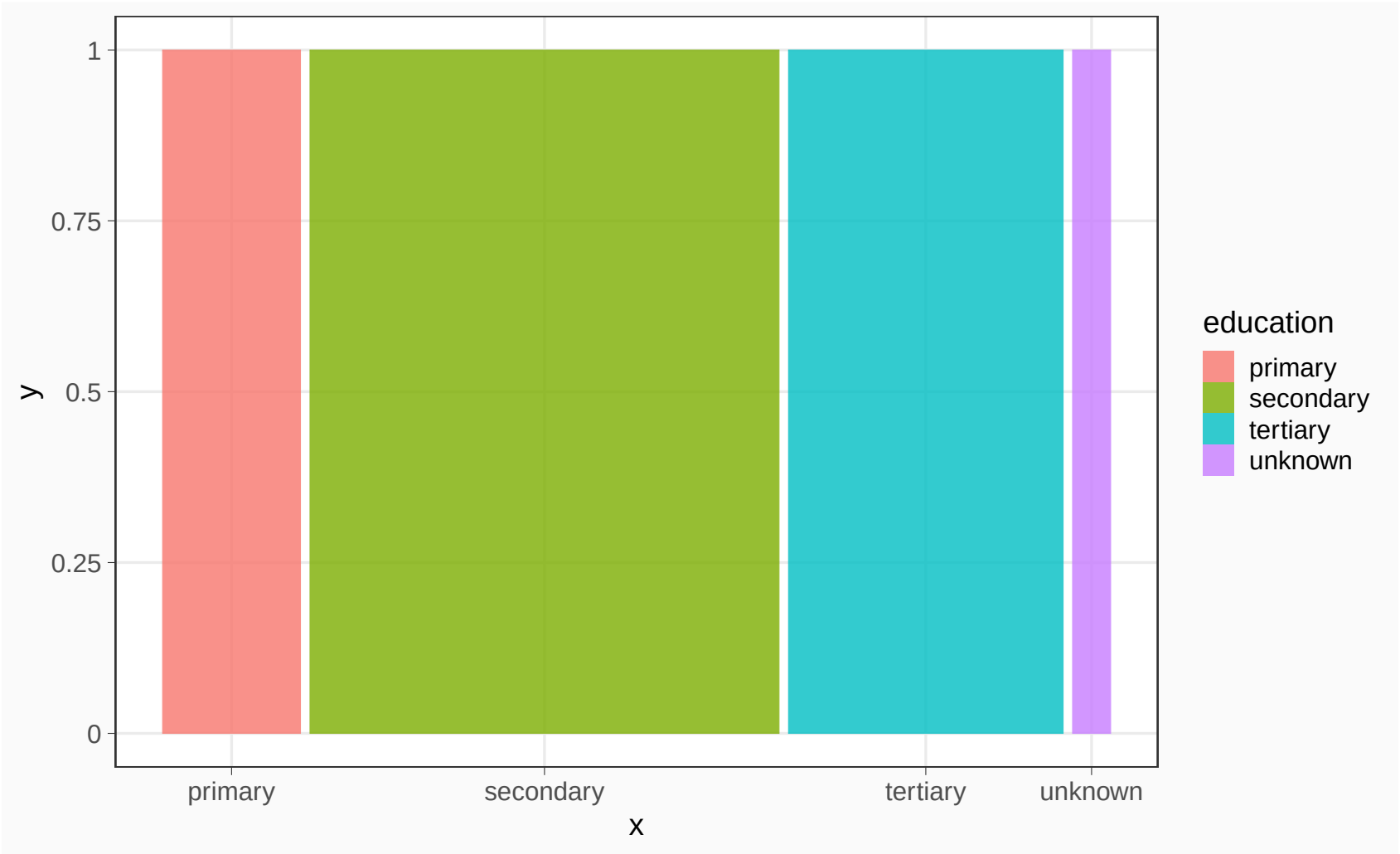
Principles

- Mosaic plots display dependencies between categorical variables
- Recursive splitting and conditional analysis:
  - horizontal widths are proportional to the first variable category frequencies
  - vertical heights are proportional to the second variable category conditional frequencies
- can be used with more than 2 variables (less readable)
- ggmosaic package for ggplot



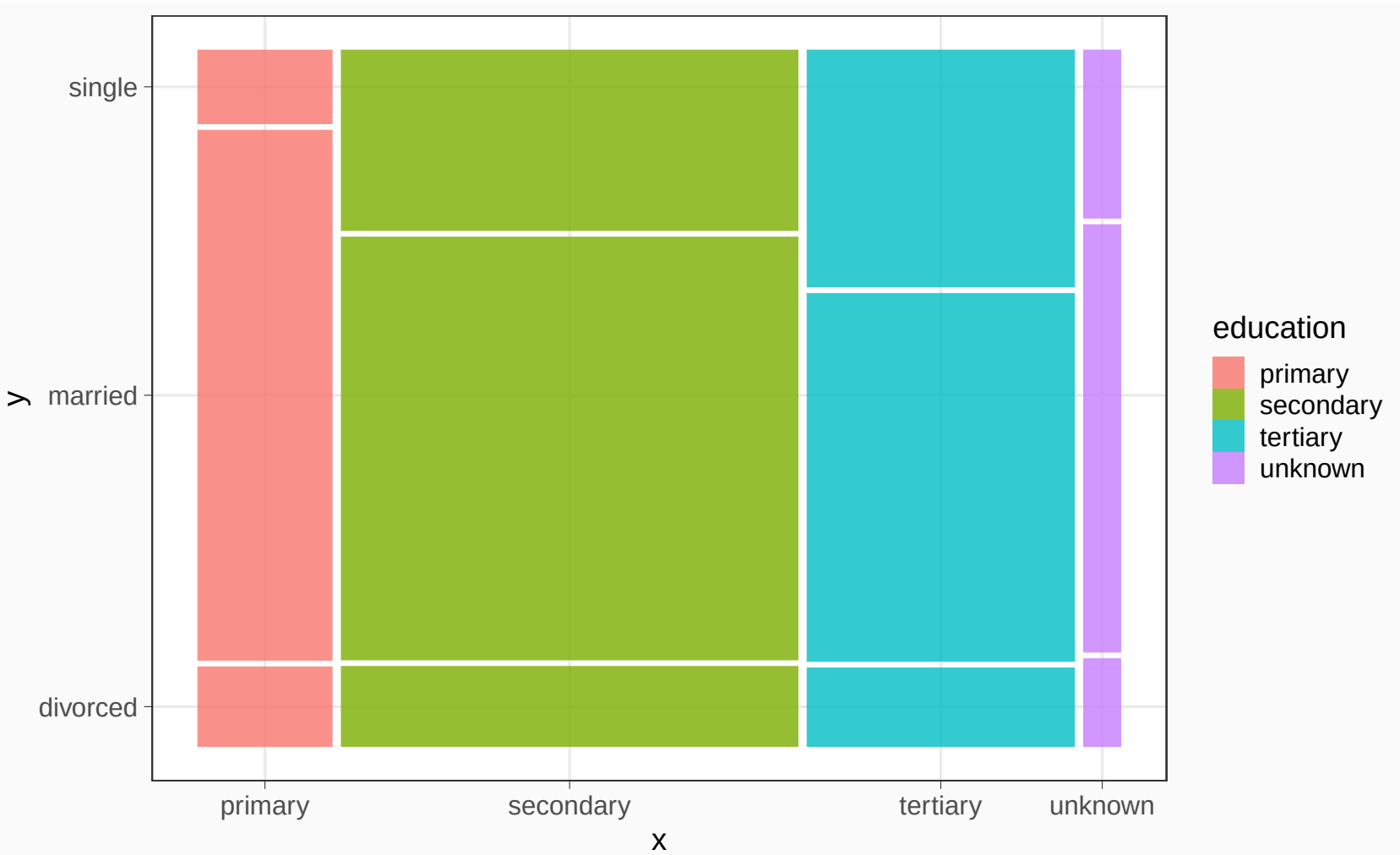
Mosaic plots

```
ggplot(bank) + geom_mosaic(mapping=aes(x=product(education),fill=education))
```



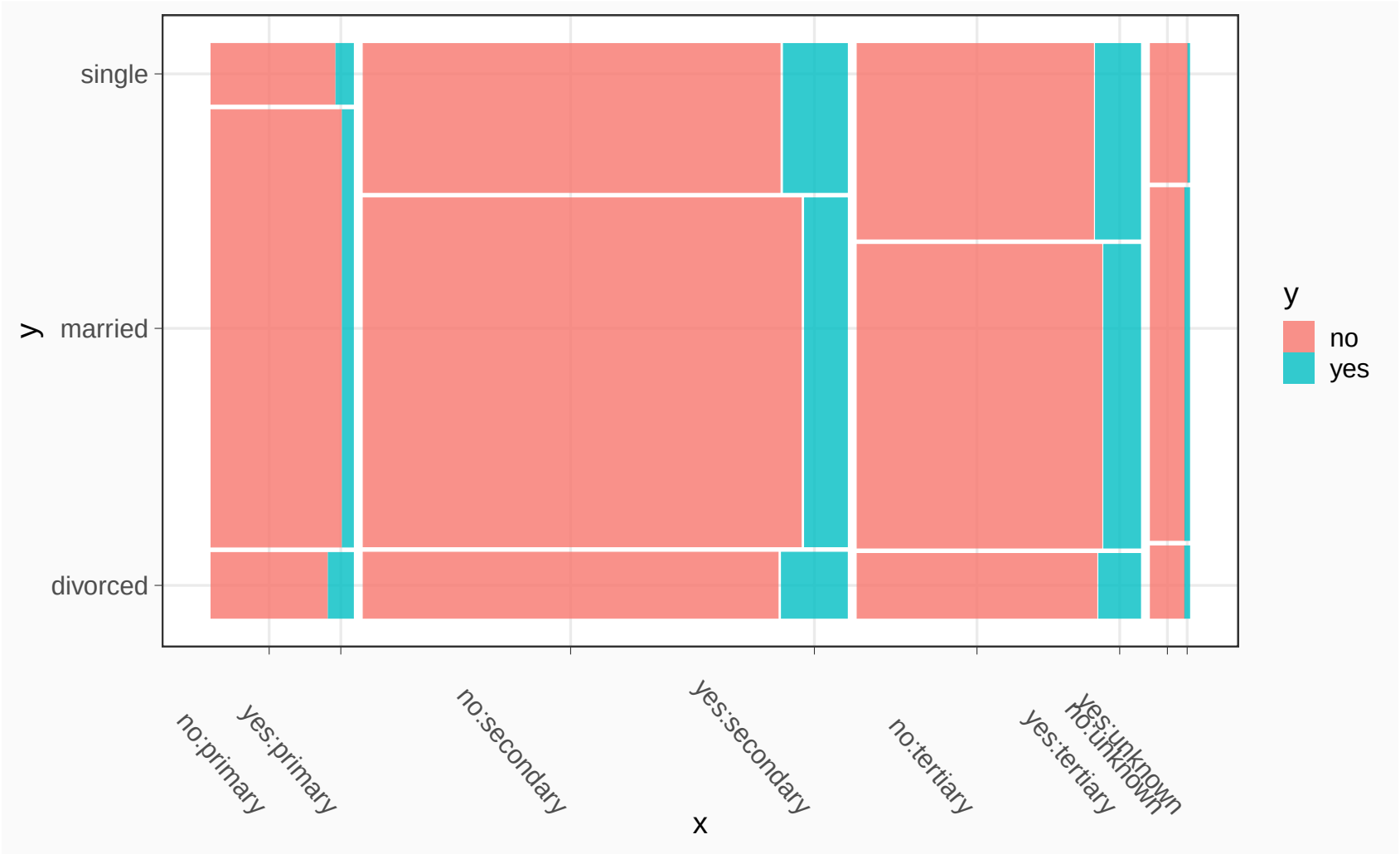
Mosaic plots

```
ggplot(bank) + geom_mosaic(mapping=aes(x=product(marital,education),fill=education))
```



Mosaic plots

```
ggplot(bank) + geom_mosaic(mapping=aes(x=product(y,marital,education),fill=y)) +
    theme(axis.text.x = element_text(angle = -50, hjust = 1))
```



Alternative package

Limitations

- ggmosaic provides only illustrative plots
- it misses diagnostic plots

Categorical data analysis

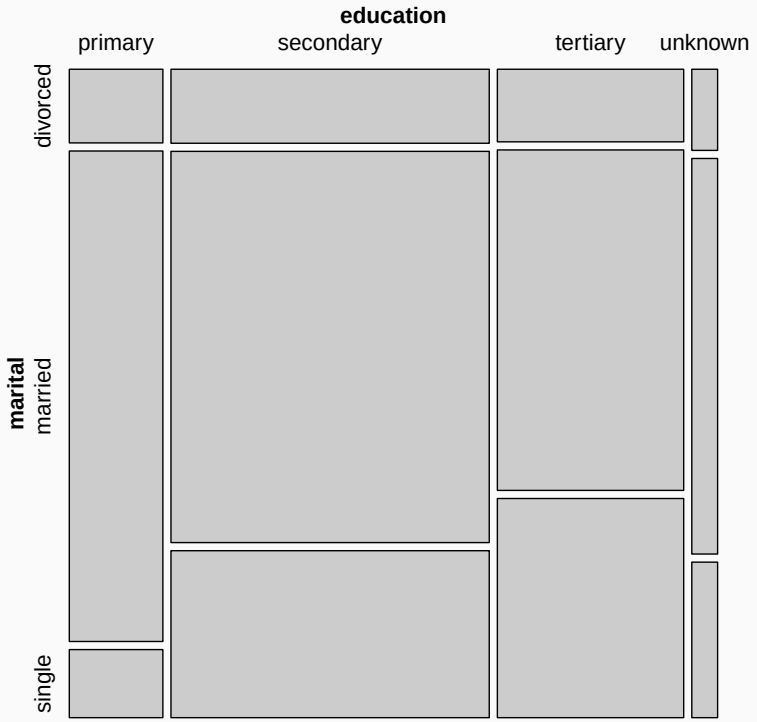
- specific methods
- specific visualization techniques
- vcd and vcdExtra packages

Example

With vcd

```
structable(marital~education,data=bank)
##           marital divorced married single
## education
## primary                  79     526     73
## secondary               270    1427    609
## tertiary                155     727    468
## unknown                  24     117     46

mosaic(structable(marital~education,
                  data=bank),
       direction=c("v"))
```



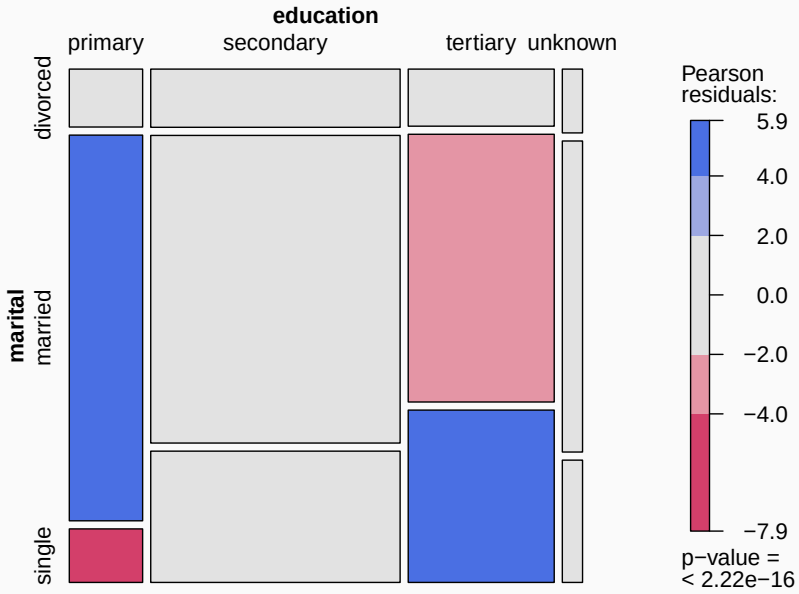
Example

With vcd

```
mosaic(structable(marital~education,
                  data=bank),
       direction=c("v"),gp=shading_hcl)
```

Shading

- diagnostic plot
- Pearson residuals of a chi-squared test
- divergent colormap:
  - red: less than expected under independence
  - blue: more than expected under independence



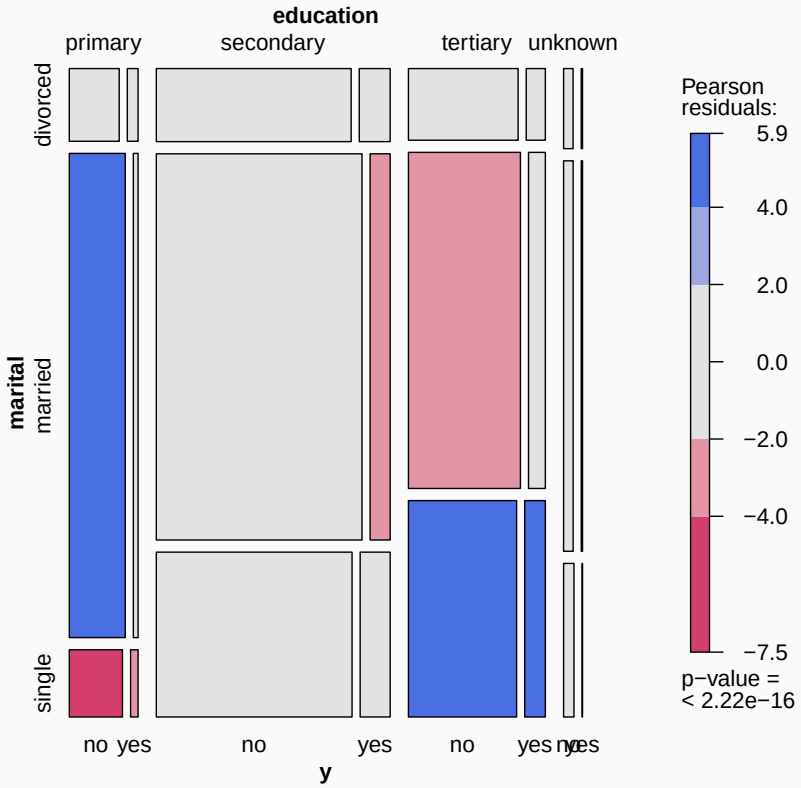
Example

With vcd

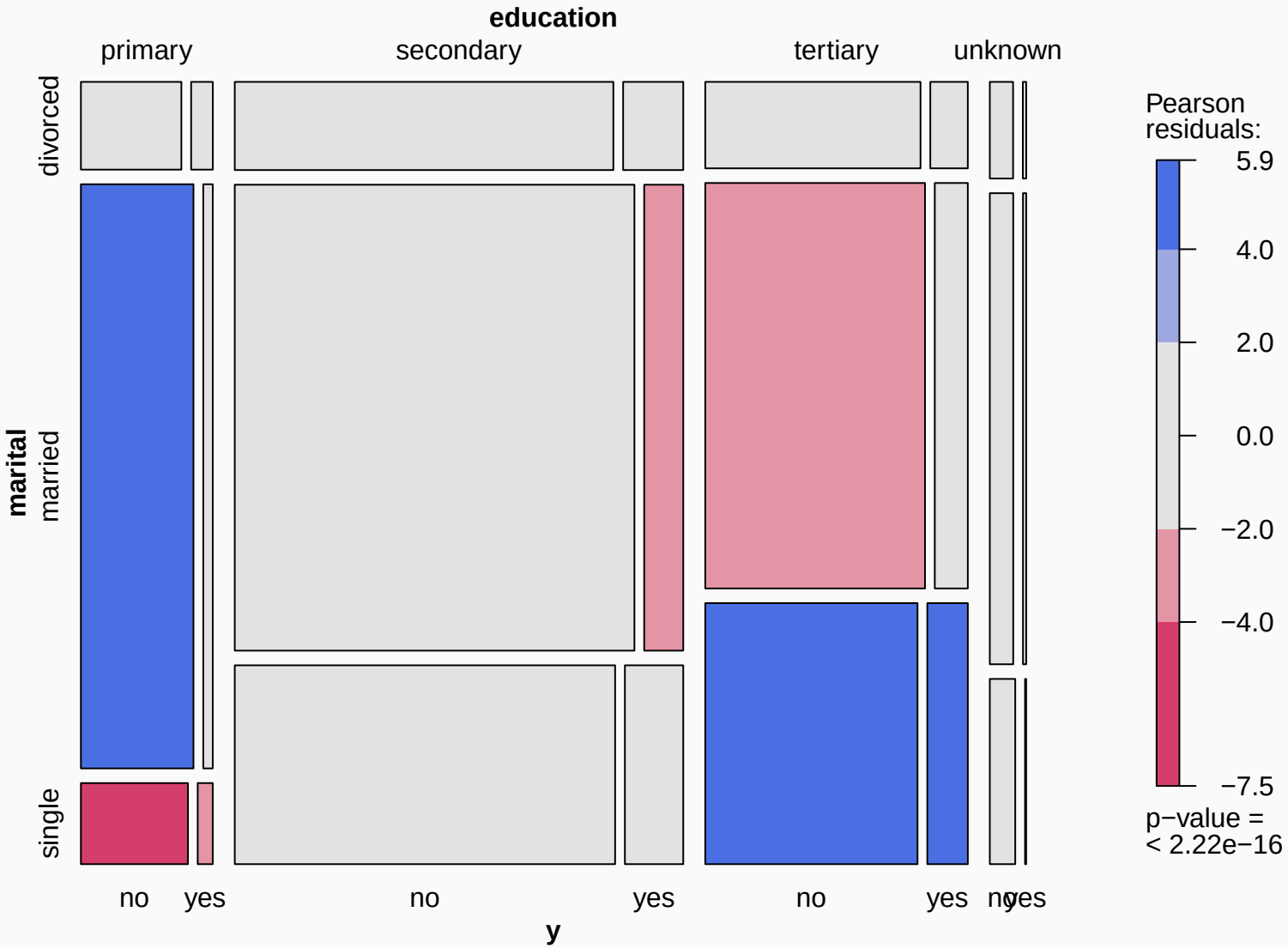
```
mosaic(structable(y~education+marital,
                  data=bank),
       direction=c("v"),gp=shading_hcl)
```

Shading

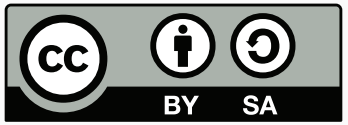
- diagnostic plot
- Pearson residuals of a chi-squared test
- divergent colormap:
  - red: less than expected under independence
  - blue: more than expected under independence



Example


education
primary
secondary
tertiary
unknown
divorced
married
single
marital
no
yes
no
yes
no
yes
no
yes
y
Pearson residuals:
5.9
4.0
2.0
0.0
−2.0
−4.0
−7.5
p−value = < 2.22e−16

Licence



Version

Last git commit: 2020-01-27
By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)
Git hash: 3aeca5ec2f6c6884d5584abc31bc2c55fa38022c

Changelog

- October 2019: initial version