CT5102: Programming for Data Analytics

Week 10: Reference Classes (RC) in R

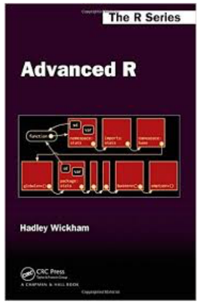
https://github.com/JimDuggan/CT5102

Dr. Jim Duggan,

Information Technology,

School of Engineering & Informatics

Overview (Wickham 2015)

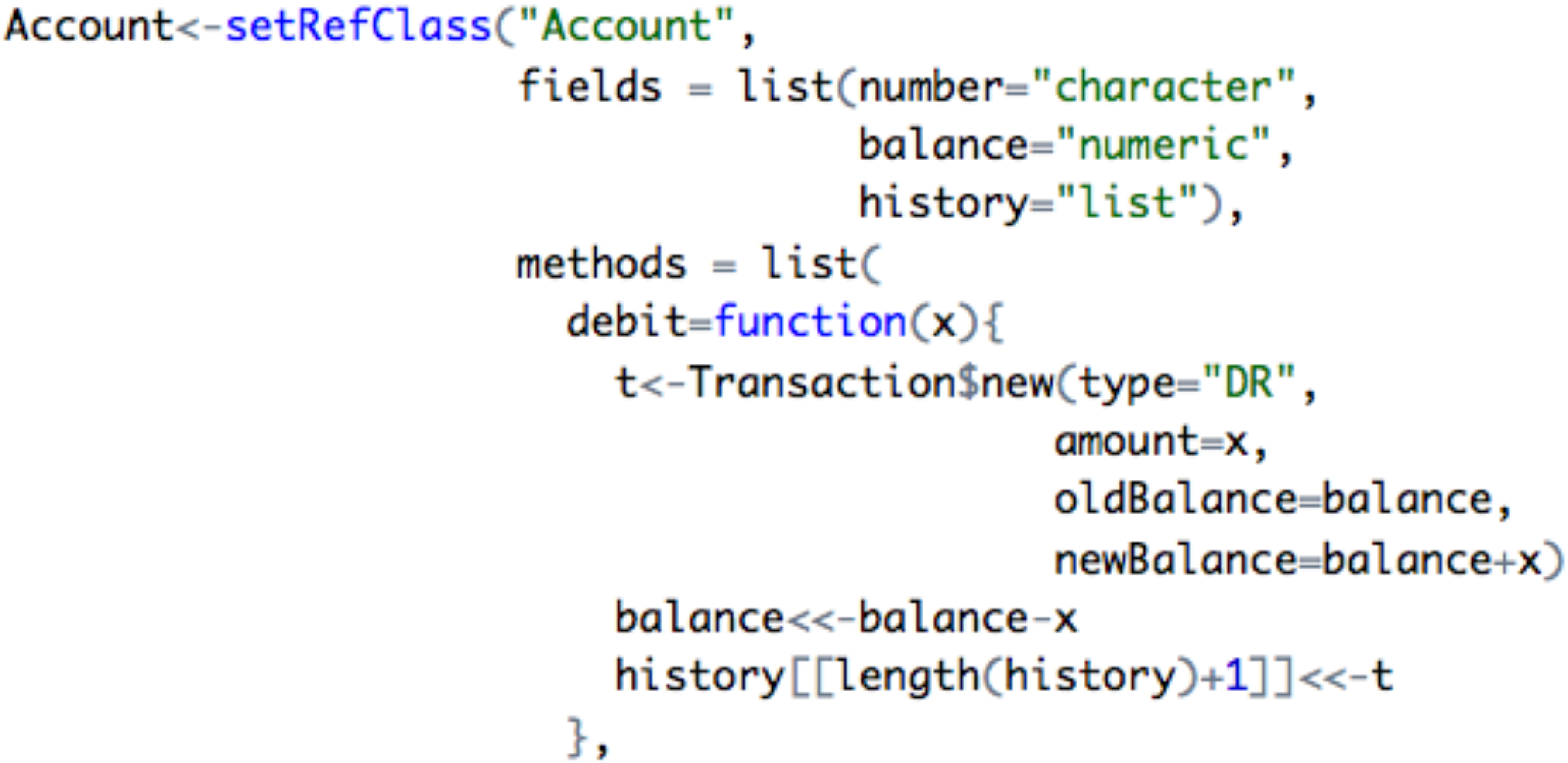


- The newest OO system in R, introduced in version 2.12
- Fundamentally different to S3 and S4
  - RC methods belong to objects, not functions
  - RC objects are mutable: the usual R copy-on-modify semantics do not apply
- Properties make RC objects behave more like objects from other languages (Python, Ruby, Java, C#)
- Best used for describing stateful objects, objects that change over time

Defining classes and creating bject

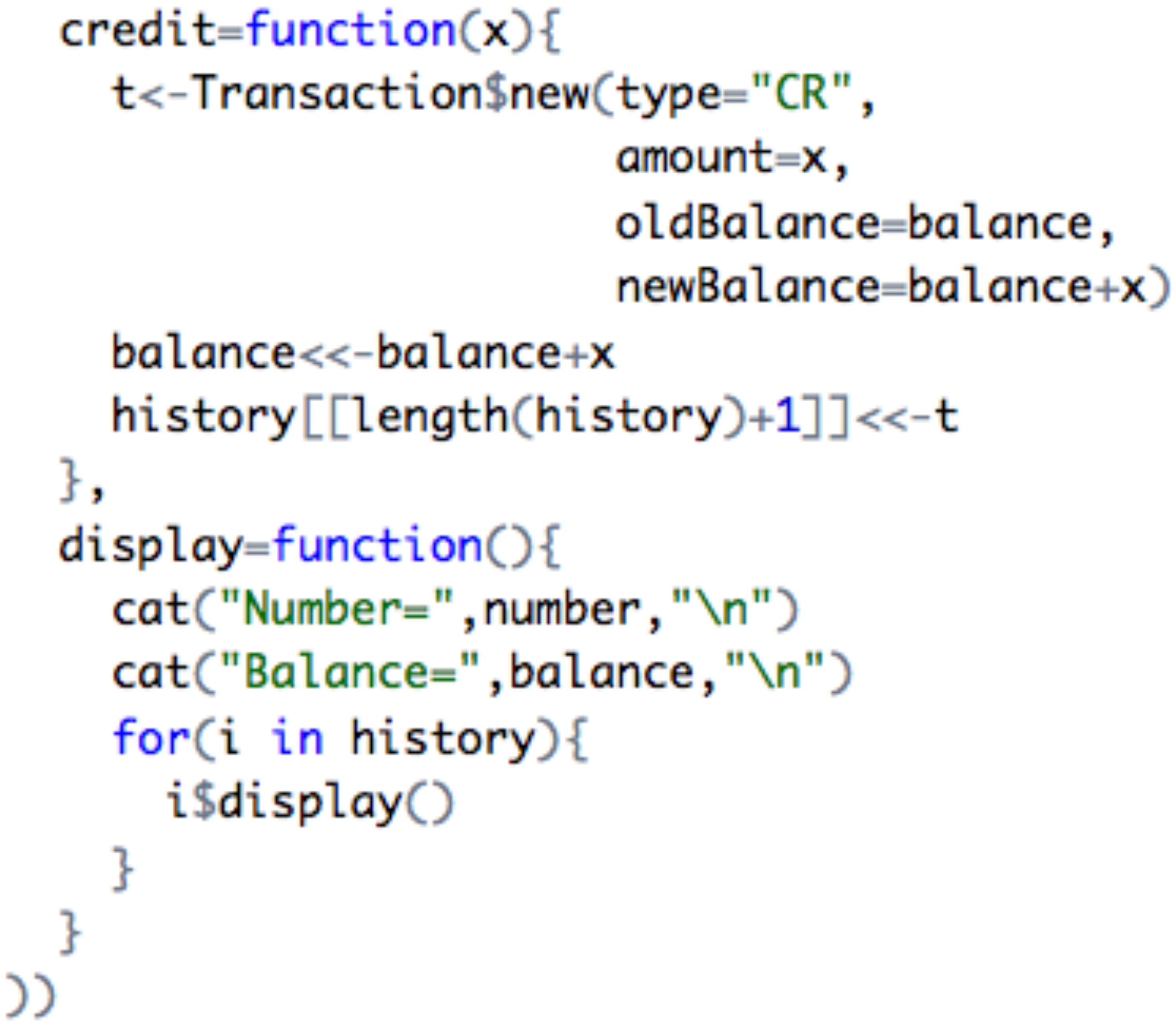
- Similar to creating a new S4 class, but use setRefClass() instead of setClass
- First, and only required argument, is an alphanumeric name
- setRefClass() also accepts
  - List of name-class pairs that define class fields (equivalent to S4 slots)
  - List of functions that change object state
  - State changed using superassignment operator.

Creating an account class (1)



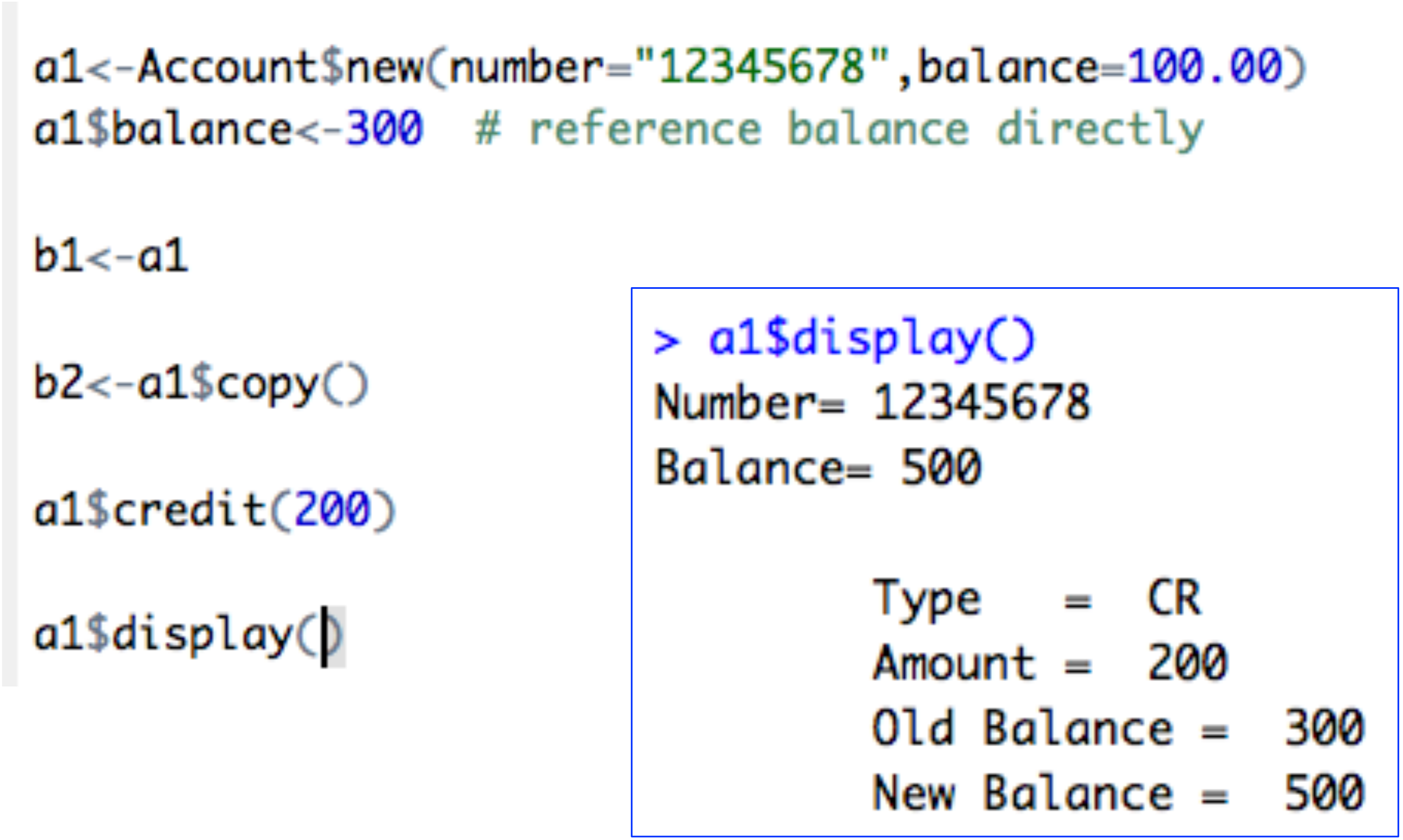
```
Account<-setRefClass("Account",
                      fields = list(number="character",
                                    balance="numeric",
                                    history="list"),
                      methods = list(
                        debit=function(x){
                          t<-Transaction$new(type="DR",
                                             amount=x,
                                             oldBalance=balance,
                                             newBalance=balance+x)
                          balance<<-balance-x
                          history[[length(history)+1]]<<-t
                        },
```

Creating an account class (2)



```
  credit=function(x){
    t<-Transaction$new(type="CR",
                       amount=x,
                       oldBalance=balance,
                       newBalance=balance+x)
    balance<<-balance+x
    history[[length(history)+1]]<<-t
  },
  display=function(){
    cat("Number=",number,"\n")
    cat("Balance=",balance,"\n")
    for(i in history){
      i$display()
    }
  }
))
```

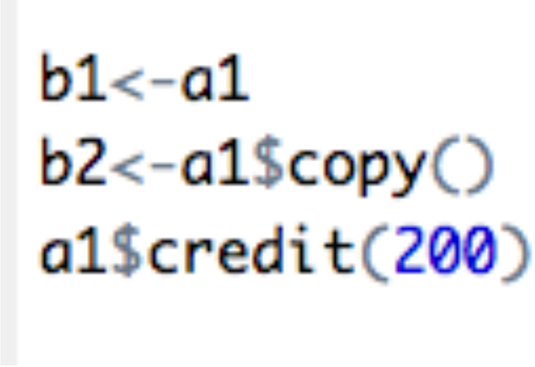
Instantiating objects



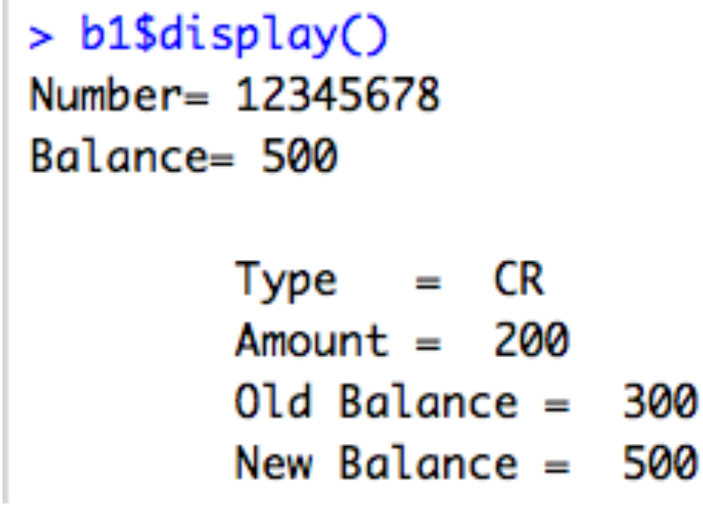
```
a1<-Account$new(number="12345678",balance=100.00)
a1$balance<-300  # reference balance directly

b1<-a1

b2<-a1$copy()

a1$credit(200)

a1$display()
```

```
> a1$display()
Number= 12345678
Balance= 500

        Type     =   CR
        Amount  =   200
        Old Balance  =   300
        New Balance  =   500
```

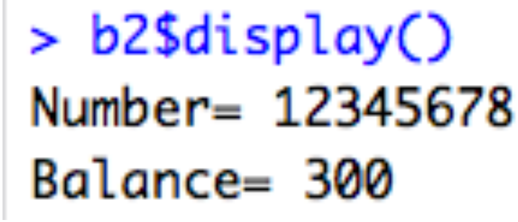
Copy objects



```
b1<-a1
b2<-a1$copy()
a1$credit(200)
```



```
> b1$display()
Number= 12345678
Balance= 500

        Type    =  CR
        Amount =  200
        Old Balance =  300
        New Balance =  500
```



```
> b2$display()
Number= 12345678
Balance= 300
```

Other features of RC…

- All reference classes inherit from envRefClass.
- This provides useful methods such as copy(), callSuper(), field(), export() and show()
- Method dispatch is simple in RC, as methods are associated with classes, and not functions.

Challenge 10.1

- Write an RC Class for a Stock Keeping Unit (SKU), with attributes: id, description, onHand, totalReceived, totalShipped
- Methods include
  - receive(amt)
  - ship(amt)
  - display()