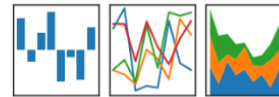




pandas  
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



02.03.2017

# Вычислительные модели с использованием научных библиотек Python Библиотека SymPy и символьные вычисления

# Базовые операции

#1

```
>>> from sympy import *
>>> x, y = symbols('x y')
>>> expr = x + 2*y
>>> expr
x + 2*y
>>> expr + 1
x + 2*y + 1
>>> expr - x
2*y + 1
>>> x*expr
x*(2*y + 1)
```

#2

```
>>> expr = x**y
>>> expr
x**y
>>> expr = expr.subs(y, x**y)
>>> expr
x**(x**y)
>>> expr = expr.subs(y, x**x)
>>> expr
x**(x**(x**x))
```

#3

```
>>> expr = x**4 - 4*x**3 + 4*x**2 - 2*x + 3
>>> replacements = [(x**i, y**i) for i in range(5) if i % 2 == 0]
>>> expr.subs(replacements)
-4*x**3 - 2*x + y**4 + 4*y**2 + 3
```

# Базовые операции

#1

```
>>> import numpy
>>> a = numpy.arange(10)
>>> expr = sin(x)
>>> f = lambdify(x, expr, "numpy")
>>> f(a)
[ 0.  0.84147098 0.90929743 0.14112001 -
 0.7568025 -0.95892427
-0.2794155 0.6569866 0.98935825
 0.41211849]
>>> def mysin(x):
...     return x
>>> f = lambdify(x, expr, {"sin":mysin})
>>> f(0.1)
0.1
```

#2

```
In [1]: from sympy import *
        x, y, z = symbols('x y z')
        init_printing()

In [2]: Integral(sqrt(1/x), x)

Out[2]: 
$$\int \sqrt{\frac{1}{x}} dx$$


>>> print(latex(Integral(sqrt(1/x), x)))
\int \sqrt{\frac{1}{x}}\, dx
```

# Модификация выражений

#1

```
>>> simplify(sin(x)**2 + cos(x)**2)
1
>>> simplify((x**3 + x**2 - x - 1)/(x**2 + 2*x + 1))
x - 1
>>> simplify(gamma(x)/gamma(x - 2))
(x - 2)·(x - 1)
```

#2

```
>>> expand((x + 1)**2)
2
x + 2·x + 1
>>> expand((x + 2)·(x - 3))
2
x - x - 6
```

#3

```
>>> factor(x**2·z + 4·x·y·z + 4·y**2·z)
2
z·(x + 2·y)
```

#4

```
>>> expr = x*y + x - 3 + 2*x**2 - z*x**2 + x**3
>>> collected_expr = collect(expr, x)
>>> collected_expr
3 2
x + x ·(-z + 2) + x·(y + 1) - 3
```

#5

```
>>> expr = 1/x + (3*x/2 - 2)/(x - 4)
>>> cancel(expr)
2
3·x - 2·x - 8
-----
2
2·x - 8·x
```

#6

```
>>> trigsimp(sin(x)**2 + cos(x)**2)
1
>>> trigsimp(sin(x)*tan(x)/sec(x))
2
sin (x)
```



# Производные и интегралы

#1

```
>>> diff(cos(x), x)
-sin(x)
>>> diff(x**4, x, x, x)
24·x
>>> diff(x**4, x, 3)
24·x
>>> expr = exp(x*y*z)
>>> diff(expr, x, y, y, z, z, z, z)
>>> deriv = Derivative(expr, x,
y, y, z, 4)
>>> deriv.doit()
```

#2

```
>>> integrate(cos(x), x)
sin(x)
>>> integrate(exp(-x**2 - y**2),
(x, -oo, oo), (y, -oo, oo))
π
>>> expr = Integral(log(x)**2, x)
>>> expr.doit()
2
x·log(x) - 2·x·log(x) + 2·x
```

# Пределы и ряды

#1

```
>>> limit(sin(x)/x, x, 0)
1
>>> limit(1/x, x, 0, '+')
∞
>>> limit(1/x, x, 0, '-')
-∞
```

#2

```
>>> expr = exp(sin(x))
>>> expr.series(x, 0, 4)
      2
      x
1 + x + —
      2
```

# Уравнения

#1

```
>>> solveset(Eq(x**2, 1), x)
{-1, 1}
>>> solveset(Eq(x**2 - 1, 0), x)
{-1, 1}
>>> solveset(x**2 - 1, x)
{-1, 1}
```

#2

```
>>> solveset(x**2 - x, x)
{0, 1}
>>> solveset(x - x, x, domain=S.Reals)
R
>>> solveset(sin(x) - 1, x, domain=S.Reals)
{
  π
  2·n·π + π | n ∈ Z
  2
}
```

#3

```
>>> solveset(exp(x), x) # No solution exists
∅
>>> solveset(cos(x) - x, x) # Not able to find
solution
{x | x ∈ C ∧ -x + cos(x) = 0}
```

#4

```
>>> linsolve([x + y + z - 1, x + y + 2*z - 3], (x, y, z))
{(-y - 1, y, 2)}
```

#5

```
>>> M = Matrix(((1, 1, 1, 1), (1, 1, 2, 3)))
>>> system = A, b = M[:, :-1], M[:, -1]
>>> linsolve(system, x, y, z)
{(-y - 1, y, 2)}
```

# ОДУ

```
>>> f=symbols('f', cls=Function)
>>> diffeq = Eq(f(x).diff(x, x) - 2*f(x).diff(x)
+ f(x), sin(x))
>>> dsolve(diffeq, f(x))
      x  cos(x)
f(x) = (C1 + C2·x)·e + —————
                        2
>>> dsolve(f(x).diff(x)*(1 - sin(f(x))), f(x))
f(x) + cos(f(x)) = C1
```



# Матричные операции

```
#1 >>> M = Matrix([[1, 3], [-2, 3]])
>>> N = Matrix([[0, 3], [0, 7]])
>>> M + N

$$\begin{bmatrix} 1 & 6 \\ -2 & 10 \end{bmatrix}$$

>>> M*N

$$\begin{bmatrix} 0 & 24 \\ 0 & 15 \end{bmatrix}$$

>>> 3*M

$$\begin{bmatrix} 3 & 9 \\ -6 & 9 \end{bmatrix}$$

>>> M**2

$$\begin{bmatrix} -5 & 12 \\ -8 & 3 \end{bmatrix}$$

>>> M**,-1

$$\begin{bmatrix} 1/3 & -1/3 \\ 2/9 & 1/9 \end{bmatrix}$$

```

```
#2 >>> eye(2)

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

>>> zeros(2, 3)

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

>>> ones(2, 2)

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

>>> diag(1, 2)

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

```

```
#3 >>> M.det()
>>> M.T
```

```
#4 >>> M = Matrix([[3, -2, 4, -2], [5, 3, -3, -2], [5, -2, 2, -2], [5, -2, -3, 3]])
>>> M.eigenvals()
{-2: 1, 3: 1, 5: 2}
>>> M.eigenvects()
```