Example 3.1. Reconsider the pig problem of Example 1.1, but now take into account the fact that the growth rate of the pig is not constant. Assume that the pig is young, so that the growth rate is increasing. When should we sell the pig for maximum profit?

linear growth model

estimate about how fast pig grows

$w(t) = 200 + 5t$

$w(0) = 200$ lbs

$w'(0) \approx 5$ lb/day

$C(t) = 0.45t$

$p(t) = 0.65 - 0.01t$

$R(t) = w(t)p(t)$

$P(t) = R(t) - C(t)$

Exponential growth model.

$w(t) = w_0 e^{ct}$

exponential growth...

$w(0) = 200 = w_0 e^{c\cdot 0} = w_0 = 200$

$w'(0) = 5 = w_0 c e^{c\cdot 0} = 200c = 5$

$c = \frac{5}{200} = \frac{1}{40}$

$C(t) = 0.45t$

$p(t) = 0.65 - 0.01t$

$R(t) = w(t)p(t)$

$P(t) = R(t) - C(t)$

Note exponential growth means the weight gained is proportial to the current weight of the pig.

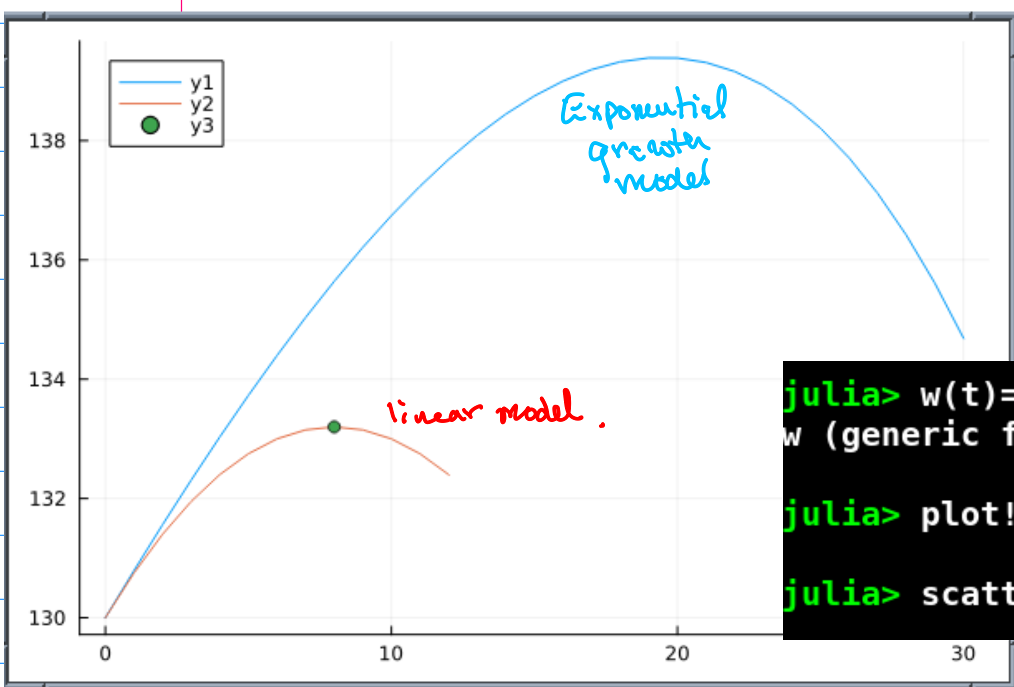
$$\frac{dw}{dt} = cw$$

proportionality units $[c] = \frac{1}{[\text{time}]}$

dimensions of c

```
julia> w(t)=200*exp(c*t)
w (generic function with 1 method)

julia> c=1/40
0.025

julia> C(t)=0.45*t
C (generic function with 1 method)

julia> p(t)=0.65-0.01*t
p (generic function with 1 method)

julia> R(t)=w(t)*p(t)
R (generic function with 1 method)

julia> P(t)=R(t)-C(t)
P (generic function with 1 method)
```

```
julia> using Plots

julia> plot(P,0:1:30)
```



linear model comparison.

```
julia> w(t)=200+5*t
w (generic function with 1 method)

julia> plot!(P,0:1:12)

julia> scatter!((8,P(8)))
```

```
julia> using Symbolics

julia> D(f,x)=expand_derivatives(Differential(x)(f))
D (generic function with 1 method)

julia> @variables t
1-element Vector{Num}:
 t
```

```
julia> dP=D(P(t),t)
-0.45 - 2.0exp(0.025t) + 5.0(0.65 - 0.01t)*exp(0.025t)

julia> expand(dP)
-0.45 + 1.25exp(0.025t) - 0.05t*exp(0.025t)
```

Want to solve $P'(t)=0$ to find the maximum.

$$P'(t) = \underbrace{-0.45 + 1.25\exp(0.025t) - 0.05t\cdot\exp(0.025t)}_{\text{difficult term...}} \approx 0$$

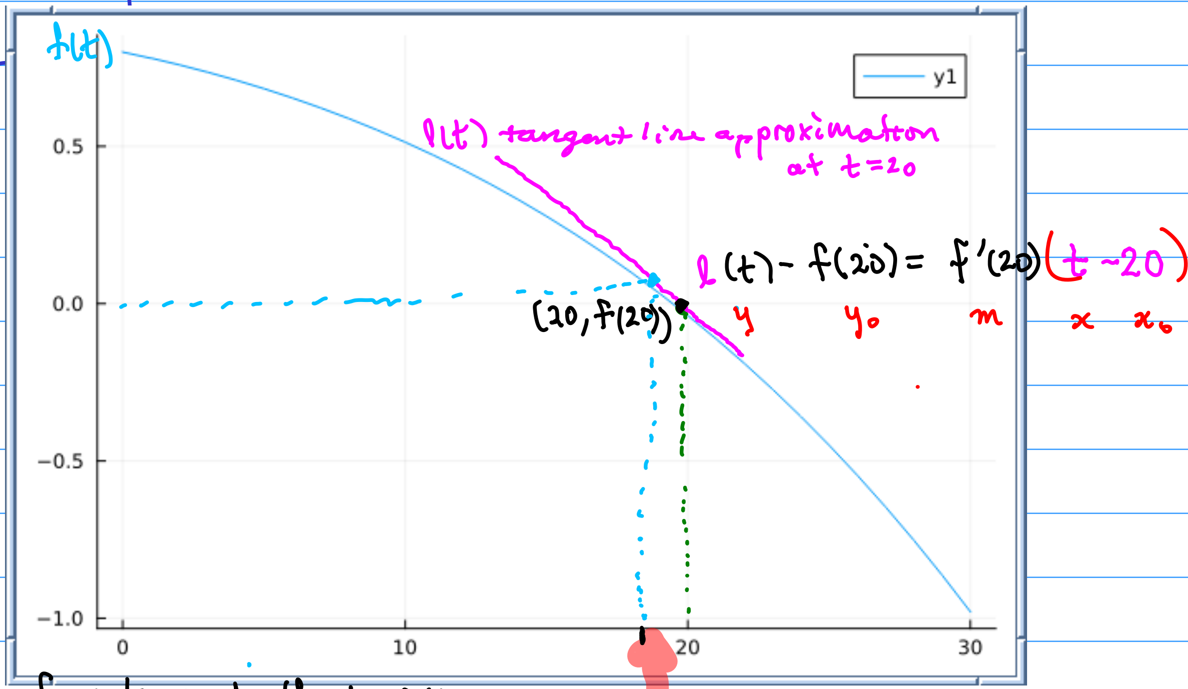
Use successive numerical approximations to solve for t.

Newton's method.

```
julia> "fdP(t)="*string(expand(dP))
"fdP(t)=-0.45 + 1.25exp(0.025t) - 0.05t*exp(0.025t)"

julia> dPs="fdP(t)="*string(expand(dP))
"fdP(t)=-0.45 + 1.25exp(0.025t) - 0.05t*exp(0.025t)"
```

$f(t)$

```
julia> eval(Meta.parse(dPs))
fdP (generic function with 1 m

julia> plot(fdP,0:1:30)
```



Solve for t such that $l(t)=0$

$$0 = l(t) = f'(20)(t-20) + f(20)$$

$$\frac{-f(20)}{f'(20)} = (t-20) \qquad t = 20 - \frac{f(20)}{f'(20)} = 19.475684642955773$$

```
julia> d2P=expand(D(fdP(t),t))
-0.018750000000000003exp(0.025t) - 0.0012500000000000002t*exp(0.025t)
```

```
julia> d2Ps="fd2P(t)=-0.01875*exp(0.025t) - 0.00125*t*exp(0.025t)"
"fd2P(t)=-0.01875*exp(0.025t) - 0.00125*t*exp(0.025t)"

julia> eval(Meta.parse(d2Ps))
fd2P (generic function with 1 method)
```

$$t = 20 - \frac{f(20)}{f'(20)}$$

$$g(x) = x - \frac{f(x)}{f'(x)}, \qquad t = g(20).$$

```
julia> g(x)=x-fdP(x)/fd2P(x)
g (generic function with 1 method)

julia> g(t)
t + (0.45 - 1.25exp(0.025t) + 0.05t*exp(0.025t)) / (-0.01875exp(0.025t) - 0.0012
5t*exp(0.025t))
```

```
julia> g(20)
19.475684642955773
```

Use successive numerical approximations to solve for t.

```
julia> g(g(20))
19.46816097315497

julia> g(g(g(20)))
19.46815944416128

julia> g(g(g(g(20))))
19.46815944416122

julia> g(g(g(g(g(20)))))
19.46815944416122
```

sequence of approximations converged.

```
julia> plot(P,0:1:30)

julia> scatter!((topt,P(topt)))
```