Polynomial Functions: Non Linear Functions

```
require(ISLR)
```

```
## Loading required package: ISLR
```

```
attach(Wage)
```

Polynomial Regression

1: Single predictor with 4th degree polynomial

```
summary(Wage)
```

```
##       year           age                     maritl           race
##  Min.   :2003   Min.   :18.00   1. Never Married: 648   1. White:2480
##  1st Qu.:2004   1st Qu.:33.75   2. Married      :2074   2. Black: 293
##  Median :2006   Median :42.00   3. Widowed      :  19   3. Asian: 190
##  Mean   :2006   Mean   :42.41   4. Divorced     : 204   4. Other:  37
##  3rd Qu.:2008   3rd Qu.:51.00   5. Separated    :  55
##  Max.   :2009   Max.   :80.00
##
##               education                     region
##  1. < HS Grad      :268   2. Middle Atlantic   :3000
##  2. HS Grad        :971   1. New England       :   0
##  3. Some College   :650   3. East North Central:   0
##  4. College Grad   :685   4. West North Central:   0
##  5. Advanced Degree:426   5. South Atlantic    :   0
##                           6. East South Central:   0
##                           (Other)              :   0
##             jobclass               health      health_ins      logwage
##  1. Industrial :1544   1. <=Good     : 858   1. Yes:2083   Min.   :3.000
##  2. Information:1456   2. >=Very Good:2142   2. No : 917   1st Qu.:4.447
##                                                            Median :4.653
##                                                            Mean   :4.654
##                                                            3rd Qu.:4.857
##                                                            Max.   :5.763
##
##       wage
##  Min.   : 20.09
##  1st Qu.: 85.38
##  Median :104.92
##  Mean   :111.70
##  3rd Qu.:128.68
##  Max.   :318.34
##
```

```
#4th degree polynomial for age input
fit = lm(wage~poly(age,4),data=Wage)
summary(fit)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, 4), data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.707 -24.626  -4.993  15.217 203.693
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     111.7036     0.7287 153.283  < 2e-16 ***
## poly(age, 4)1   447.0679    39.9148  11.201  < 2e-16 ***
## poly(age, 4)2  -478.3158    39.9148 -11.983  < 2e-16 ***
## poly(age, 4)3   125.5217    39.9148   3.145  0.00168 **
## poly(age, 4)4   -77.9112    39.9148  -1.952  0.05104 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.91 on 2995 degrees of freedom
## Multiple R-squared:  0.08626,    Adjusted R-squared:  0.08504
## F-statistic: 70.69 on 4 and 2995 DF,  p-value: < 2.2e-16
```

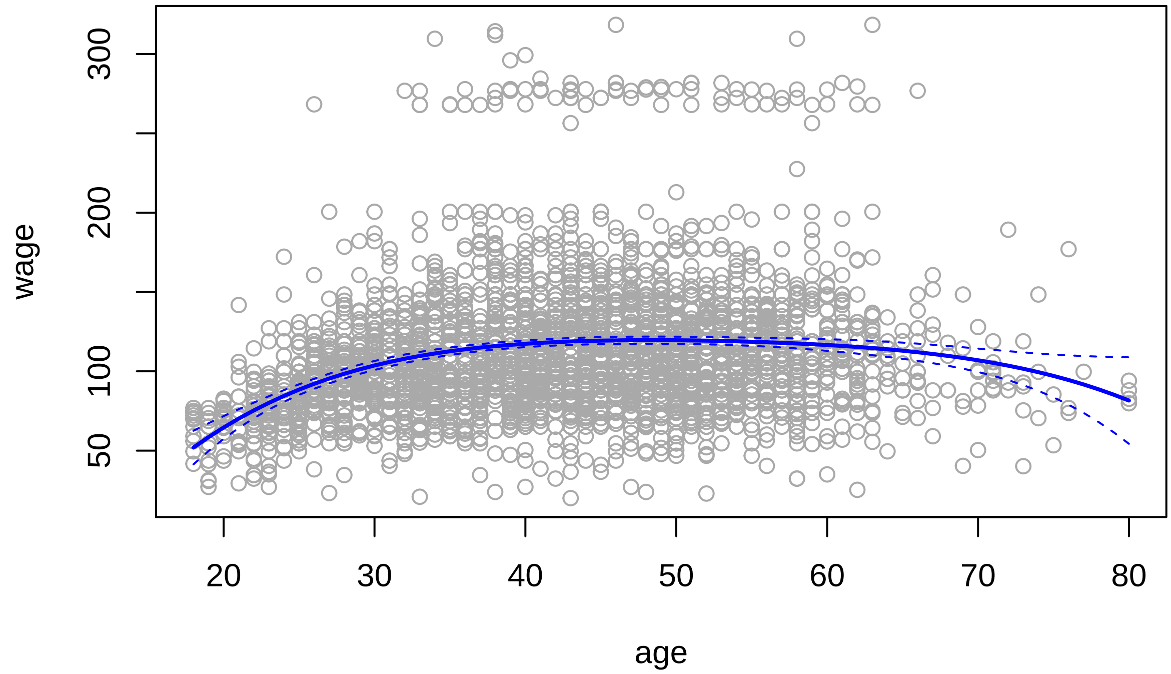
Generating a plot of the function

```
#limits of age parameter
agelims = range(age)
#grid of age parameter max to min
age.grid = seq(from=agelims[1],to=agelims[2])
age.grid
```

```
##  [1] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## [24] 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
## [47] 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
```

Predicting with standard errors

```
preds = predict(fit,newdata =list(age=age.grid),se=TRUE)
se.bands=cbind(preds$fit+2*preds$se,preds$fit-2*preds$se)
plot(age,wage,col="darkgrey")
lines(age.grid,preds$fit,lwd=2,col="blue")
matlines(age.grid,se.bands,col="blue",lty=2)
```



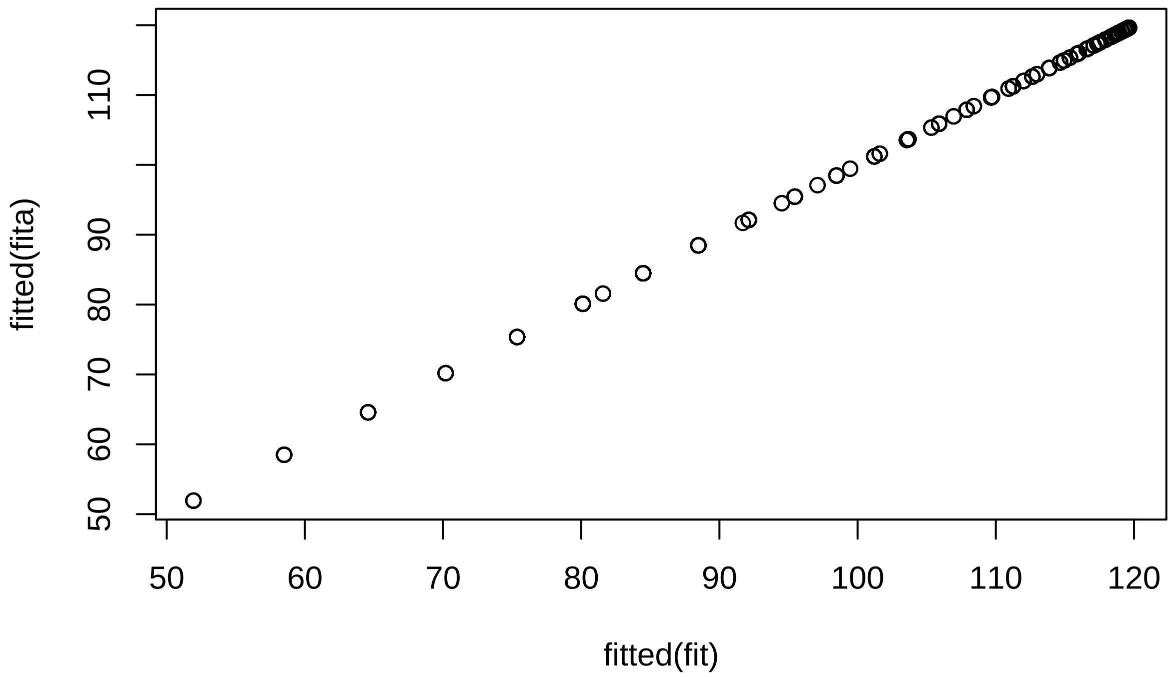
Here we get the two range polynomial bands. Along with fitted polynomial line.

Fitting Polynomials without R

```
# I = identity function used for raising parameters to a power
fita=lm(wage~age+I(age^2)+I(age^3)+I(age^4),data=Wage)
summary(fita)
```

```
##
## Call:
## lm(formula = wage ~ age + I(age^2) + I(age^3) + I(age^4), data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.707 -24.626  -4.993  15.217 203.693
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.842e+02  6.004e+01  -3.067 0.002180 **
## age          2.125e+01  5.887e+00   3.609 0.000312 ***
## I(age^2)    -5.639e-01  2.061e-01  -2.736 0.006261 **
## I(age^3)     6.811e-03  3.066e-03   2.221 0.026398 *
## I(age^4)    -3.204e-05  1.641e-05  -1.952 0.051039 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.91 on 2995 degrees of freedom
## Multiple R-squared:  0.08626,    Adjusted R-squared:  0.08504
## F-statistic: 70.69 on 4 and 2995 DF,  p-value: < 2.2e-16
```

```
plot(fitted(fit),fitted(fita))
```



```
# We can see all the polynomial components differently
summary(fit)
```

```
##
## Call:
## lm(formula = wage ~ poly(age, 4), data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.707 -24.626  -4.993  15.217 203.693
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     111.7036     0.7287 153.283  < 2e-16 ***
## poly(age, 4)1  447.0679    39.9148  11.201  < 2e-16 ***
## poly(age, 4)2 -478.3158    39.9148 -11.983  < 2e-16 ***
## poly(age, 4)3  125.5217    39.9148   3.145  0.00168 **
## poly(age, 4)4  -77.9112    39.9148  -1.952  0.05104 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 39.91 on 2995 degrees of freedom
## Multiple R-squared:  0.08626,    Adjusted R-squared:  0.08504
## F-statistic: 70.69 on 4 and 2995 DF,  p-value: < 2.2e-16
```

If it's not a single predictor or regression then use annova.

Nested sequence of models:

```
# just wage and education
fita=lm(wage~education,data=Wage)
# Wage education and age
fitb=lm(wage~education+age,data=Wage)
# age with degree two
fitc=lm(wage~education+poly(age,2),data=Wage)
fitd=lm(wage~education+poly(age,3),data=Wage)
#using annova with a sequence
anova(fita,fitb,fitc,fitd)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ education
## Model 2: wage ~ education + age
## Model 3: wage ~ education + poly(age, 2)
## Model 4: wage ~ education + poly(age, 3)
##   Res.Df     RSS Df Sum of Sq        F Pr(>F)
## 1   2995 3995721
## 2   2994 3867992  1    127729 102.7378 <2e-16 ***
## 3   2993 3725395  1    142597 114.6969 <2e-16 ***
## 4   2992 3719809  1      5587   4.4936 0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 3 and 4 are not needed.

Fitting other than RSS models to polynomial functions; such as linear regression. Using polynomial model to fit binary response variable, wage 250K+ = 1 or 0

```
# Age condition

fit=glm(I(wage>250) ~ poly(age,3), data=Wage, family=binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = I(wage > 250) ~ poly(age, 3), family = binomial,
##     data = Wage)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2808  -0.2736  -0.2487  -0.1758   3.2868
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.8486     0.1597 -24.100  < 2e-16 ***
## poly(age, 3)1  37.8846    11.4818   3.300 0.000968 ***
## poly(age, 3)2 -29.5129    10.5626  -2.794 0.005205 **
## poly(age, 3)3   9.7966     8.9990   1.089 0.276317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 730.53  on 2999  degrees of freedom
## Residual deviance: 707.92  on 2996  degrees of freedom
## AIC: 715.92
##
## Number of Fisher Scoring iterations: 8
```

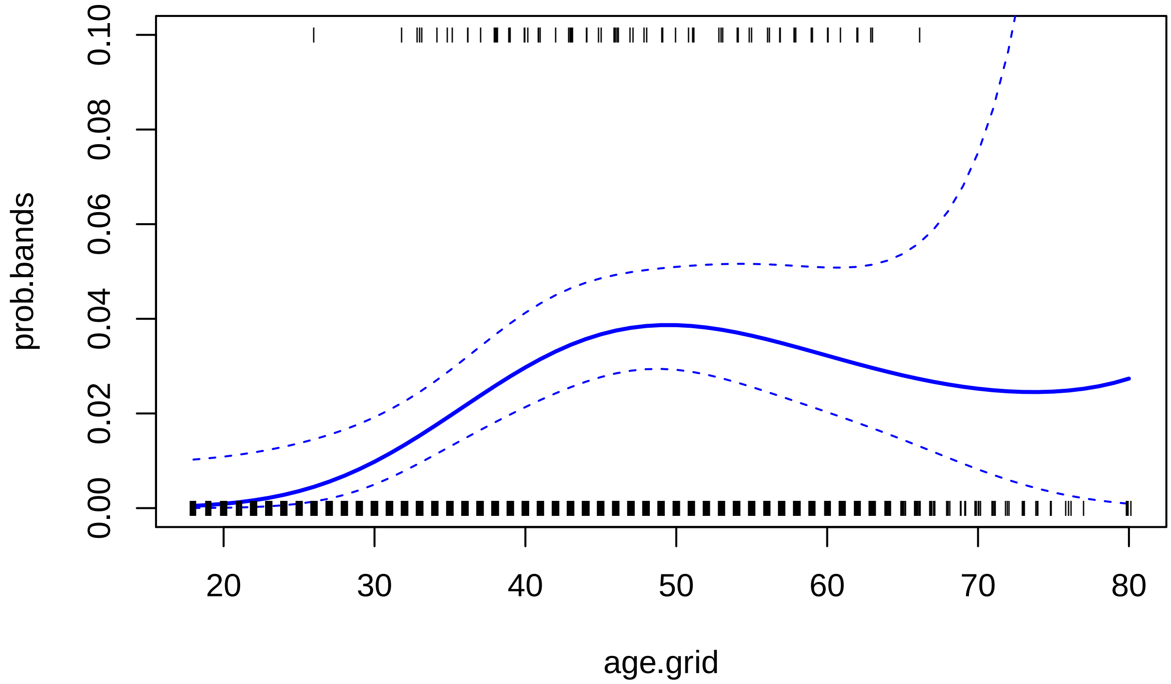
```
preds=predict(fit,list(age=age.grid),se=T)
se.bands=preds$fit + cbind(fit=0,lower=-2*preds$se,upper=2*preds$se)
se.bands[1:5,]
```

```
##         fit      lower     upper
## 1 -7.664756 -10.759826 -4.569686
## 2 -7.324776 -10.106699 -4.542852
## 3 -7.001732  -9.492821 -4.510643
## 4 -6.695229  -8.917158 -4.473300
## 5 -6.404868  -8.378691 -4.431045
```

Predict fits model on a logit scale so to transform to a probability scale we need to apply the inverse logit mapping

$$p = \frac{e^\eta}{1 + e^\eta}.$$

```
prob.bands=exp(se.bands)/(1+exp(se.bands))
matplot(age.grid,prob.bands,col="blue",lwd=c(2,1,1),lty=c(1,2,2),type="l",ylim=c(0,.1))
#find how much data actually occured
points(jitter(age),I(wage>250)/10,pch="|",cex=.5)
```



We get a standard error band where the probabilities all lie between zero and one. We can see zeros (below 250K wage below and above 250K on top). Only around 4% gets more than 250K.