FA4_EDA

ABLIAN

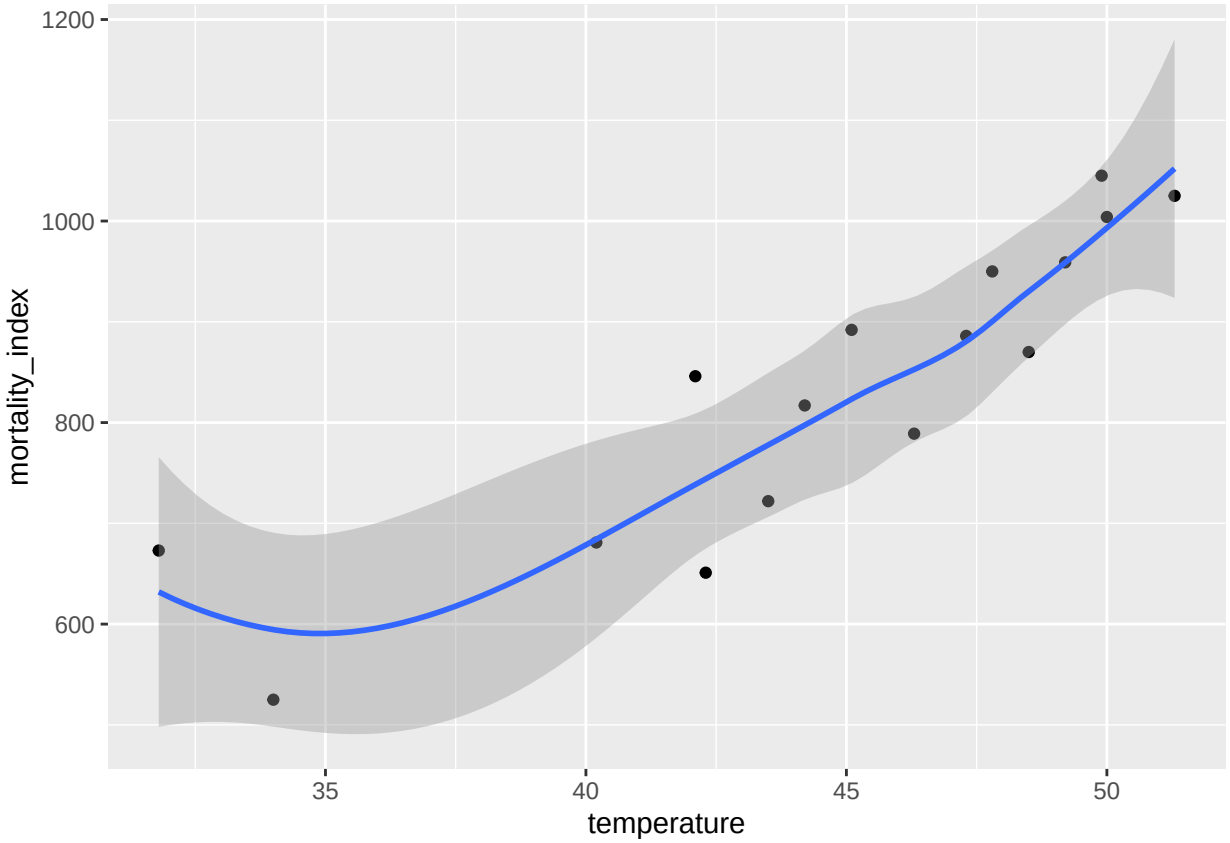
2025-03-05

```
library(ggplot2)
dataset<-read.csv("D:/FEU/3RD YR 2ND SEM/EDA/mortality_by_latitude.csv")


ggplot(dataset, aes(x = temperature, y = mortality_index)) +
  geom_point() +
  geom_smooth(method = "loess")
```

Using the Mortality by Latitude data Download Mortality by Latitude data, make a plot of mortality index against mean average temperature. Is it hollow up or hollow down? Try to identify a transformation of one of the variables that will straighten out the relationship, and make a plot of the residuals to check for any remaining patterns.

```
## `geom_smooth()` using formula = 'y ~ x'
```

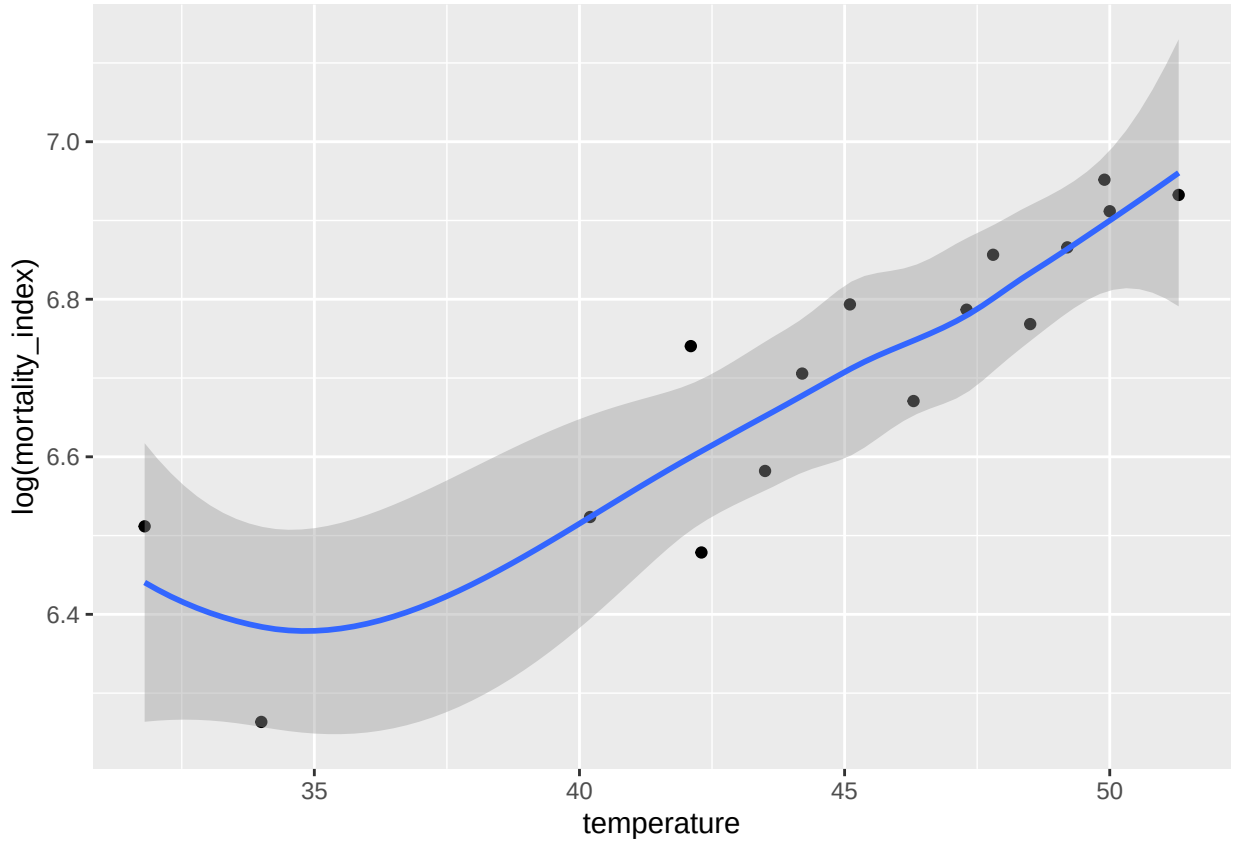


Since it is hollow up, when temperature increases, the mortality index also increases.

```
ggplot(dataset, aes(x = temperature, y = log(mortality_index))) +
  geom_point()+
  geom_smooth(method = "loess")
```

Residuals

```
## `geom_smooth()` using formula = 'y ~ x'
```

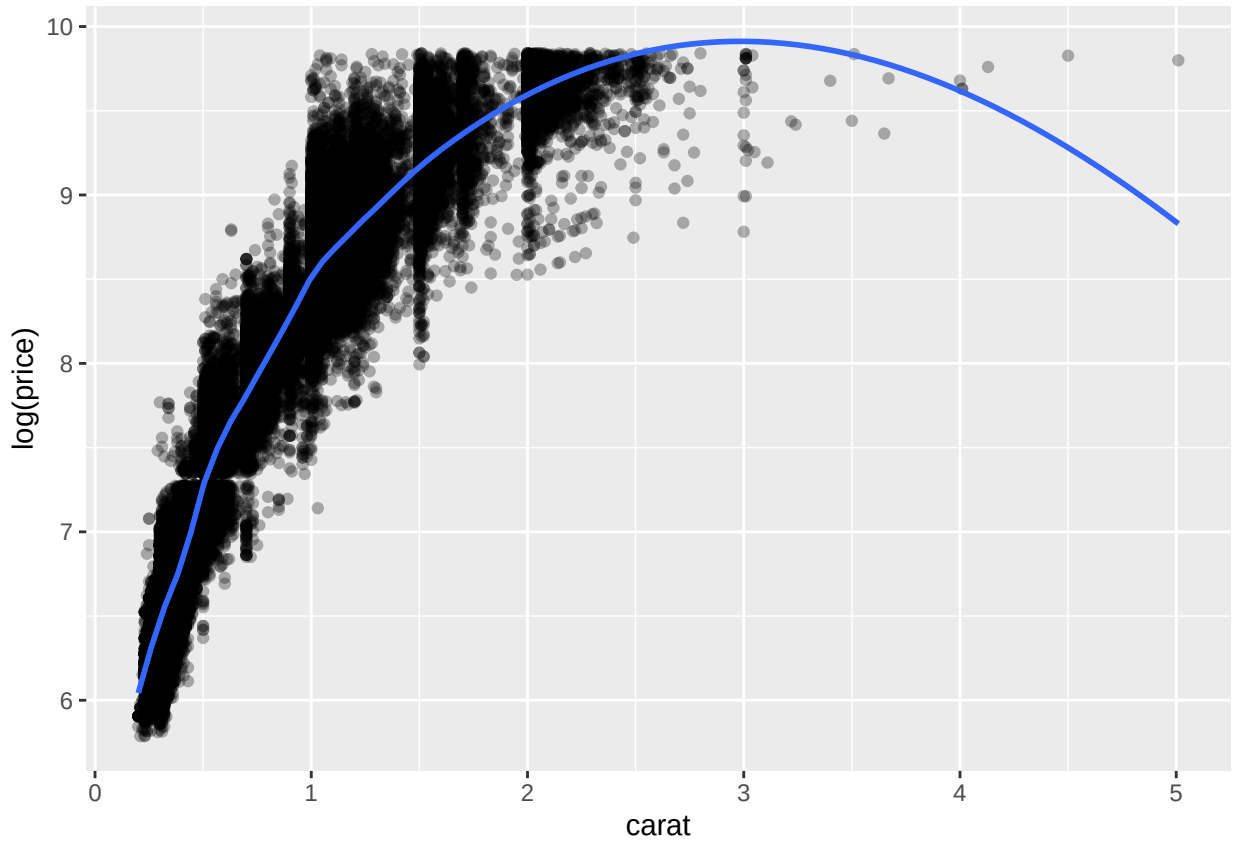



```
data(diamonds)

ggplot(diamonds, aes(x = carat, y = log(price))) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess", span = 0.4, se = FALSE)
```

Using the same subset of the diamonds dataset, make a plot of log price as a function of carat with a loess smoother. Try several values for the span and degree arguments and comment briefly about your choice.

```
## `geom_smooth()` using formula = 'y ~ x'
```



The 0.4 span is the smoothest one and it also covers most of the scattered plot, which could be used for predicting patterns as there are limited outliers when considering the trajectory of the span.

Compare the fit of the loess smoother to the fit of the polynomial + step function regression using a plot of the residuals in the two models. Which one is more faithful to the data?

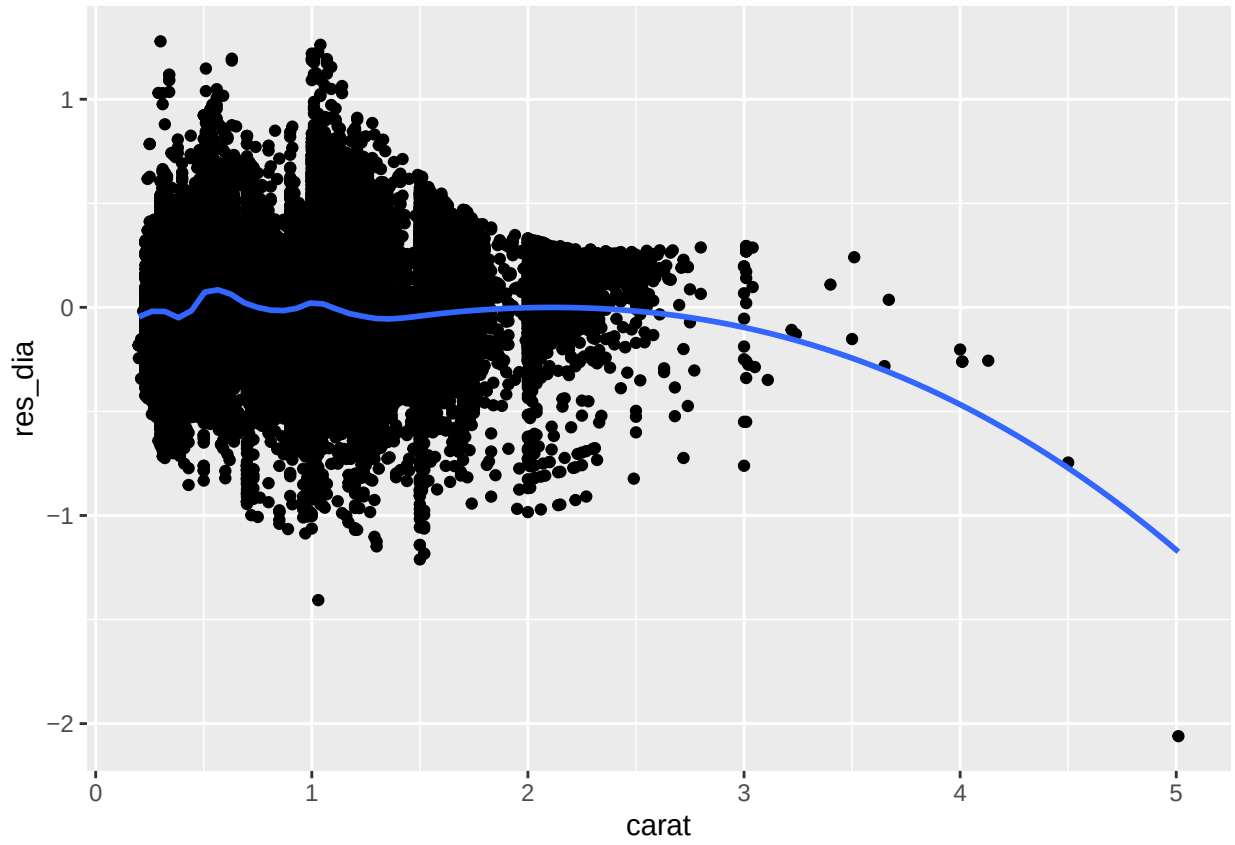
```
model_dia<-lm(log(price)~poly(carat, 3), data=diamonds)
res_dia <- resid(model_dia)
diamonds$res_dia <- res_dia

loess<-ggplot(diamonds, aes(x=carat, y=res_dia)) +
  geom_point()+
  geom_smooth(method="loess", span=0.4, se=FALSE)

loess
```

Loess

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
stepwise<-step(model_dia, direction="both")
```

Step

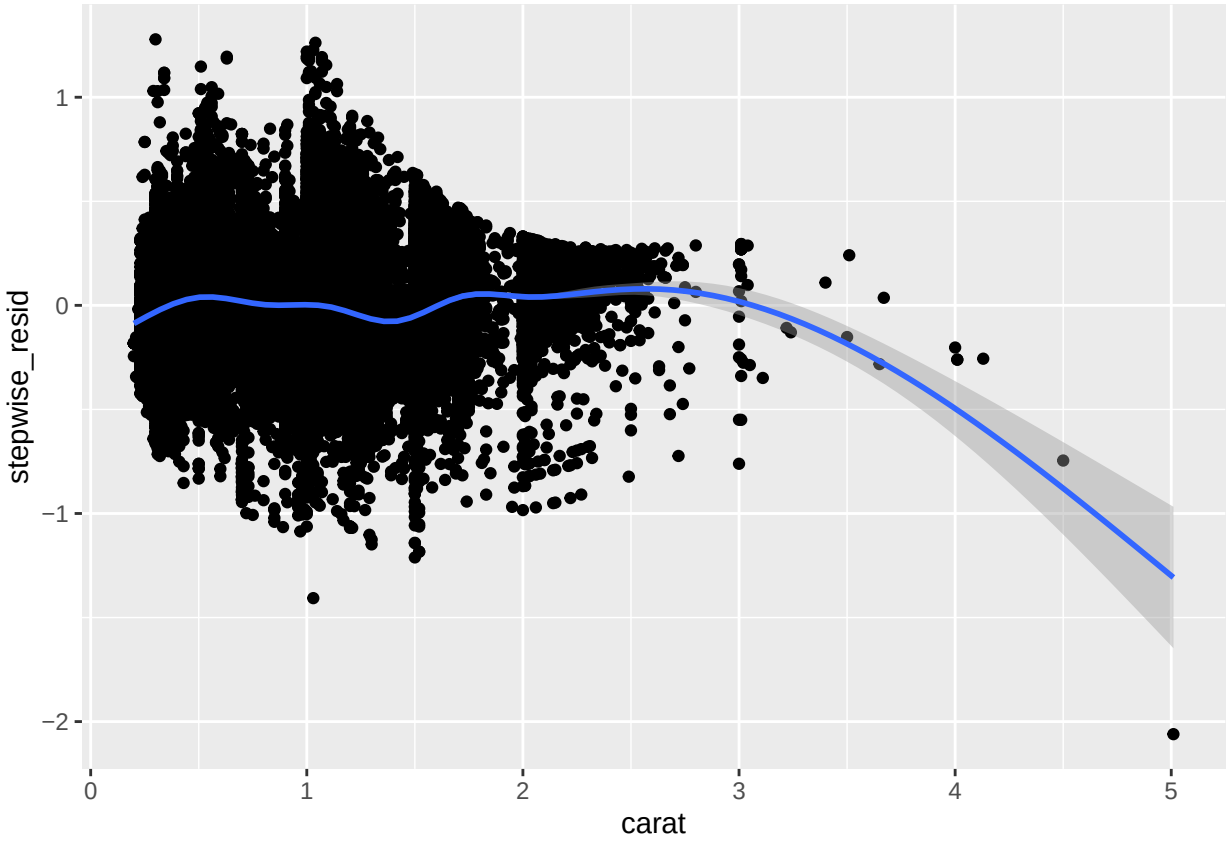
```
## Start:  AIC=-146210.1
## log(price) ~ poly(carat, 3)
##
##                  Df Sum of Sq   RSS     AIC
## <none>                         3586 -146210
## - poly(carat, 3)  3     51945 55531    1570
```

```
diamonds$stepwise_resid<-resid(stepwise)

stepwise<-ggplot(diamonds, aes(x=carat, y=stepwise_resid))+
  geom_point()+
  geom_smooth()

stepwise
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Comparison: The stepwise graph includes zero residual, showing instances where the model's prediction matches the actual values whereas loess didn't omit zero residuals which suggests that it is not fully accurate compared to the stepwise.