FA4 EDA

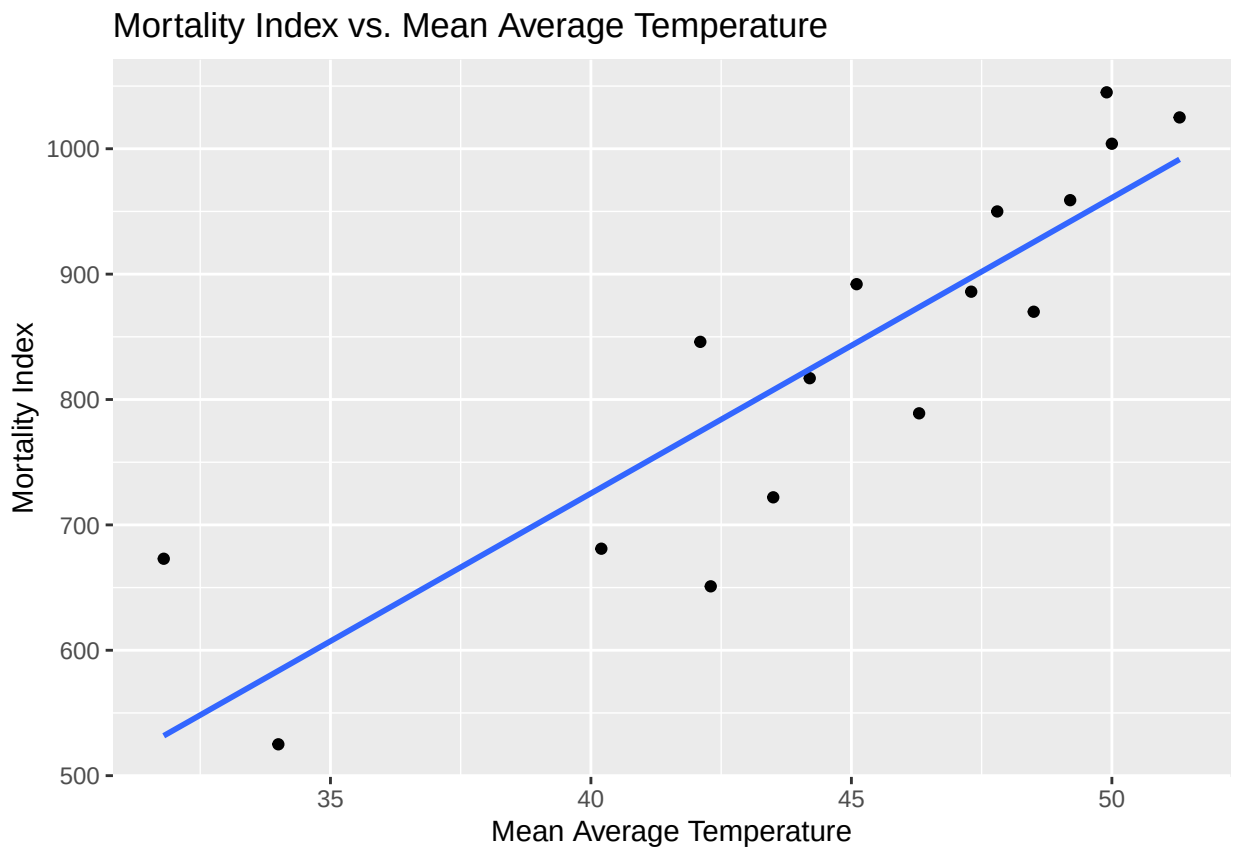
2024-02-28

```
library(ggplot2)
library(gridExtra)
```

```
mortality <- read.csv("mortality_by_latitude.csv")
diamonds <- read.csv("diamonds.csv")
```

```
scatter_plot <- ggplot(mortality, aes(x = temperature, y = mortality_index)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Mean Average Temperature", y = "Mortality Index") +
  ggtitle("Mortality Index vs. Mean Average Temperature")

print(scatter_plot)
```

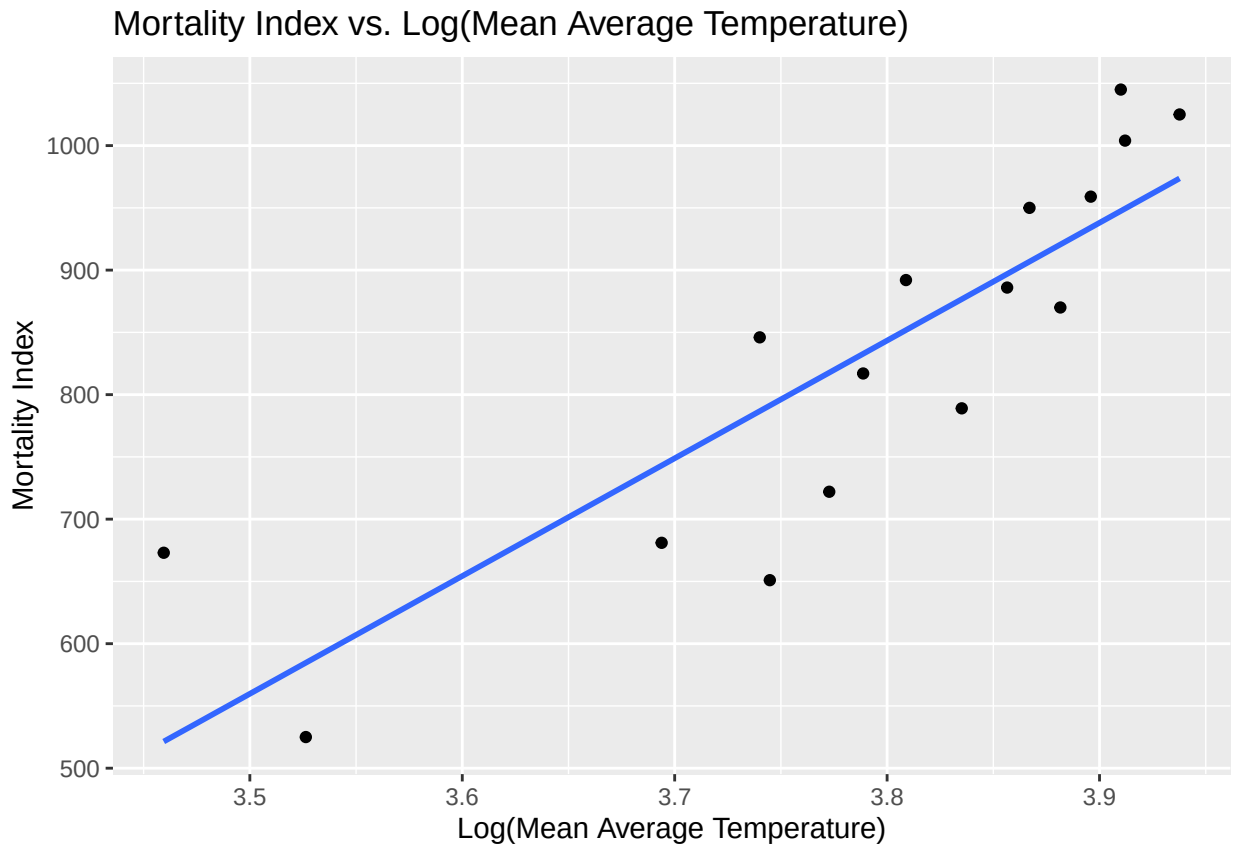
```
## `geom_smooth()` using formula = 'y ~ x'
```



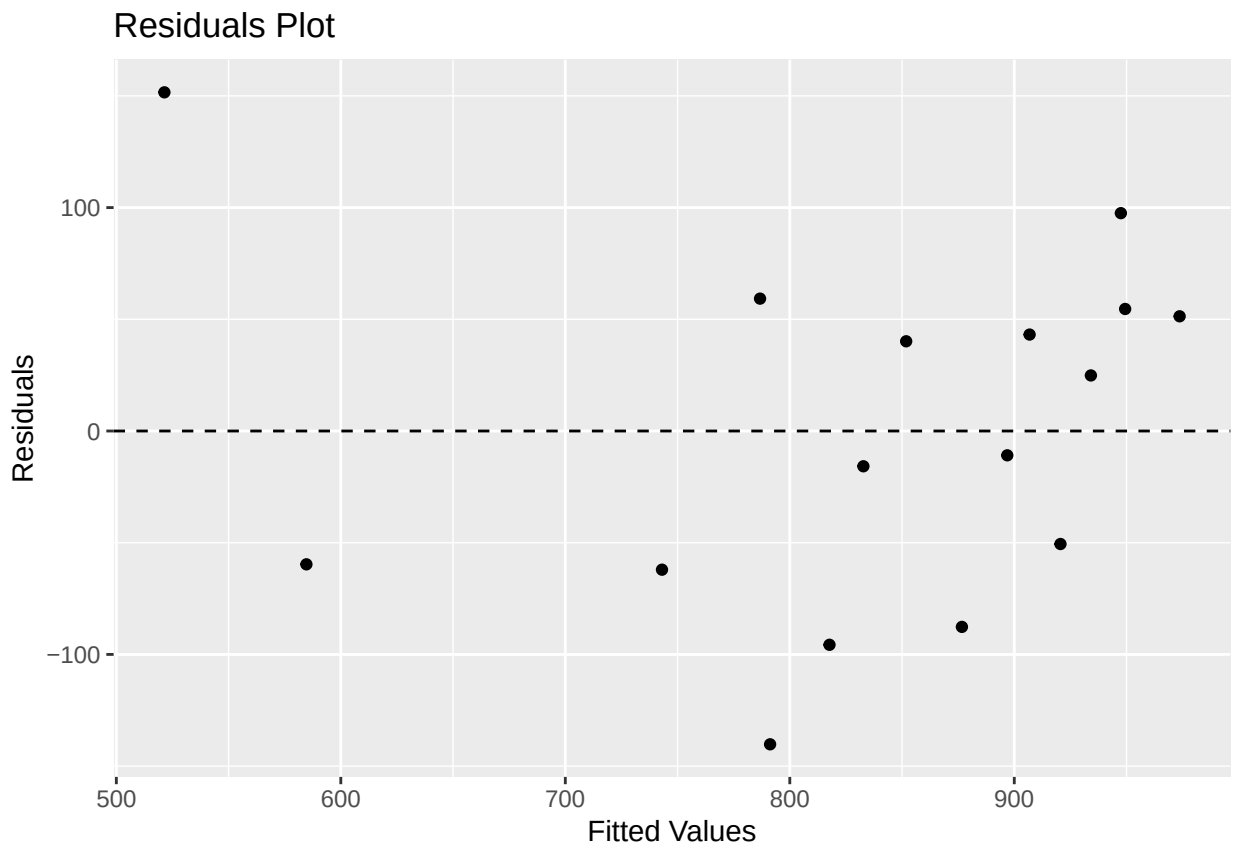
```
###hollowed up because you can see it has a upward trend
```

```
mortality$log_temperature <- log(mortality$temperature)

scatter_plot_transformed <- ggplot(mortality, aes(x = log_temperature, y = mortality_index)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Log(Mean Average Temperature)", y = "Mortality Index") +
  ggtitle("Mortality Index vs. Log(Mean Average Temperature)")

print(scatter_plot_transformed)
```

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




```
residuals_plot <- ggplot(mortality, aes(x = fitted(lm(mortality_index ~ log_temperature)), y = residuals
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted Values", y = "Residuals") +
  ggtitle("Residuals Plot")

print(residuals_plot)
```

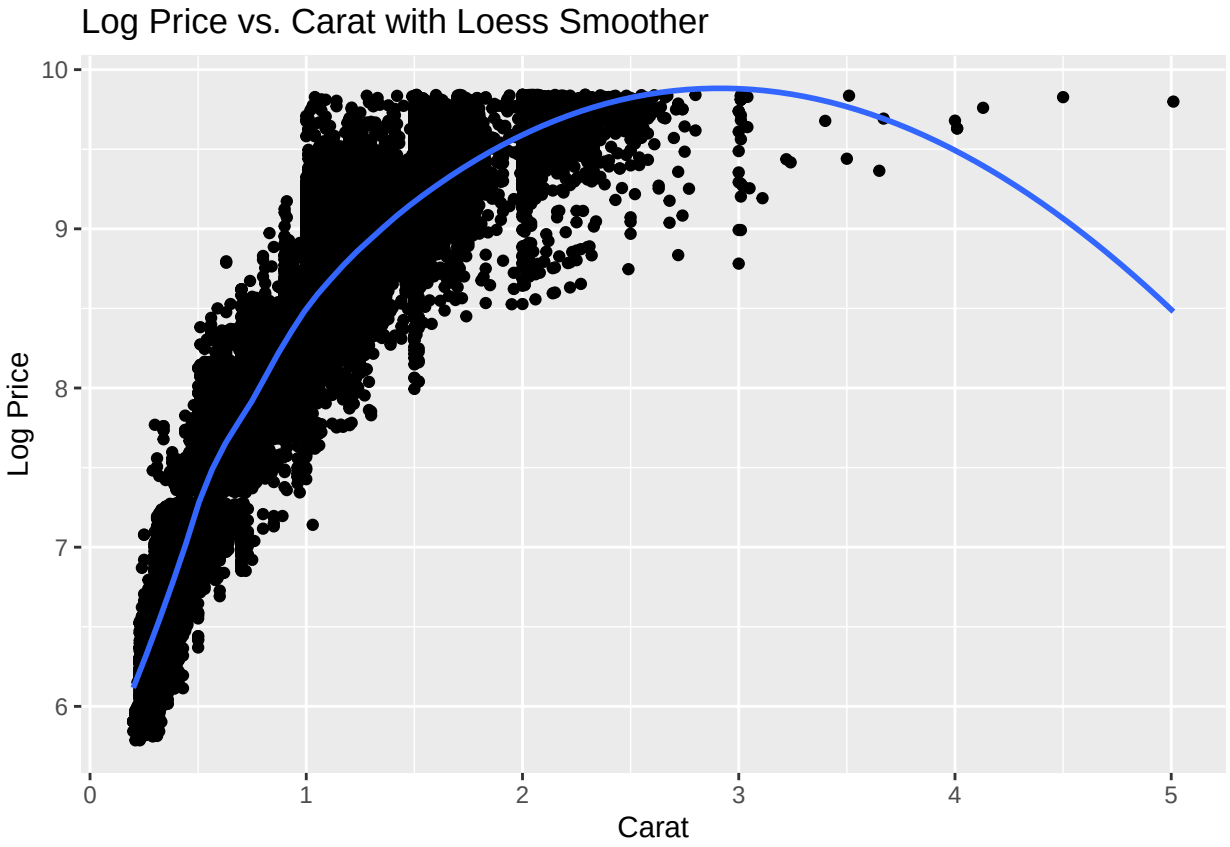



```
log_price_vs_carat_plot <- ggplot(diamonds, aes(x = carat, y = log(price))) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, span = 0.5, degree = 2) +  # You can adjust span and degree
  labs(x = "Carat", y = "Log Price") +
  ggtitle("Log Price vs. Carat with Loess Smoother")
```

```
## Warning in geom_smooth(method = "loess", se = FALSE, span = 0.5, degree = 2):
## Ignoring unknown parameters: ‘degree‘
```

```
print(log_price_vs_carat_plot)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

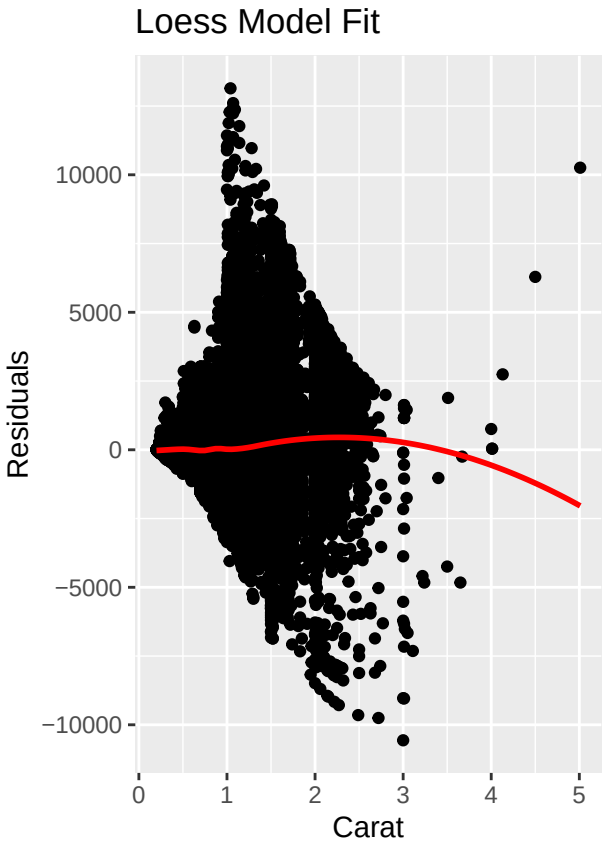


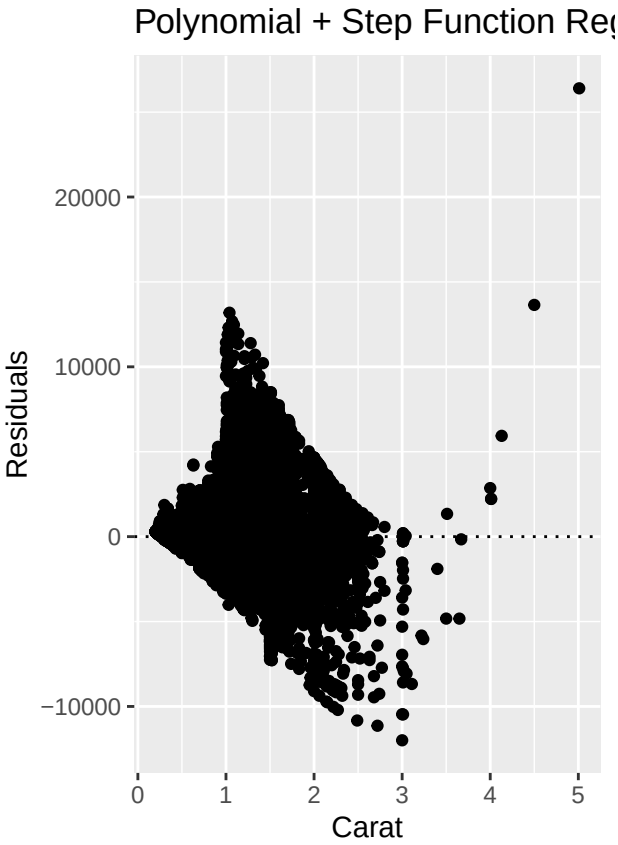

```
###with 0.5 span and degree 2 we can see it provides a balance
###in the relationship of smoothness and capturing the trend in the data.
```

```
loess_model <- loess(price ~ carat, data = diamonds)

poly_step_model <- lm(price ~ poly(carat, 3) + I(carat^3 > 3), data = diamonds)

diamonds$residuals_loess <- resid(loess_model)
diamonds$residuals_poly_step <- residuals(poly_step_model)
```

```
residuals_plot_loess <- ggplot(diamonds, aes(x = carat, y = residuals_loess)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(x = "Carat", y = "Residuals", title = "Loess Model Fit")

residuals_plot_poly_step <- ggplot(diamonds, aes(x = carat, y = residuals_poly_step)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dotted") +
  labs(x = "Carat", y = "Residuals", title = "Polynomial + Step Function Regression Model")

grid.arrange(residuals_plot_loess, residuals_plot_poly_step, nrow = 1)
```

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






```
###Polynomial + step function regression is more faithful.
###The points in the right are much closer to the line compare to the other.
```