Problem 7: Celestial Neighbourhood

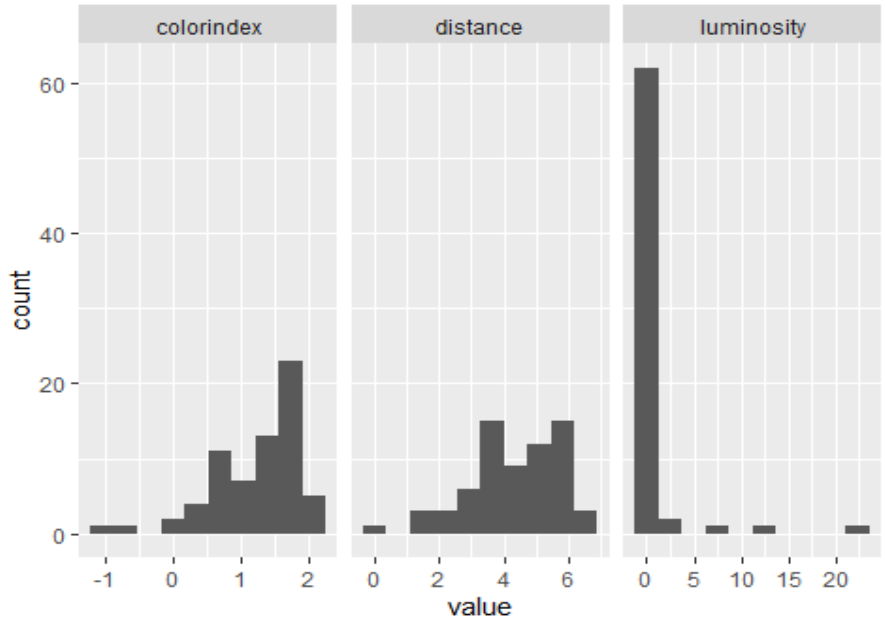
Rakesh Vishwabrahmana

Question 1

Solution:

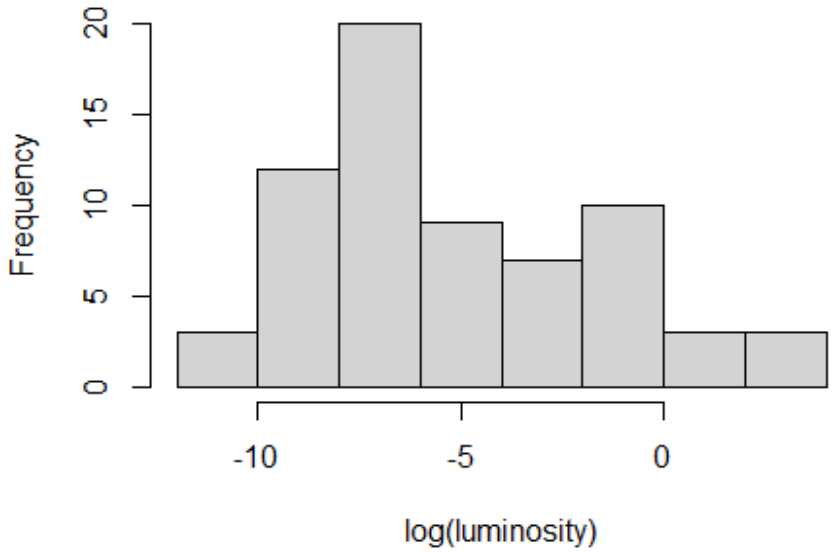
The given chunk below will show a histogram or the distribution of distance, luminosity, colour index.

```
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(moderndive)
library(ggfortify)
star <- read.csv("stars.csv", header = TRUE)
num_star <- star %>%
  select(distance, colorindex, luminosity) %>%
  gather()
num_hist <- ggplot(num_star, aes(value))+
  geom_histogram(bins = 10) +
  facet_wrap(~key, scales = "free_x")
num_hist
```



From the above Figure, we observed that the distribution of **colorindex** and **distance** are approximately normal and slightly left-skewed. But the distribution of luminosity is very much right-skewed. To make the distribution of luminosity is more regular, we use logarithm transform.

```
star <- star %>%
  mutate(log_luminosity = log(luminosity))
hist(star$log_luminosity, xlab = "log(luminosity)",main = "")
```



Question 2

Solution:

The given chunk will calculate the Spearman correlation coefficient between the three quantitative variables above.

```
dis_lum <- cor.test(star$distance, star$luminosity, method = "spearman")

## Warning in cor.test.default(star$distance, star$luminosity, method =
## "spearman"): Cannot compute exact p-value with ties

dis_lum2 <- cor.test(star$distance, star$log_luminosity, method = "spearman")

## Warning in cor.test.default(star$distance, star$log_luminosity, method =
## "spearman"): Cannot compute exact p-value with ties

dis_col <- cor.test(star$distance, star$colorindex, method = "spearman")

## Warning in cor.test.default(star$distance, star$colorindex, method =
## "spearman"): Cannot compute exact p-value with ties

col_lum <- cor.test(star$colorindex, star$luminosity, method = "spearman")

## Warning in cor.test.default(star$colorindex, star$luminosity, method =
## "spearman"): Cannot compute exact p-value with ties

col_lum2 <- cor.test(star$colorindex, star$log_luminosity, method =
"spearman")

## Warning in cor.test.default(star$colorindex, star$log_luminosity, method =
## "spearman"): Cannot compute exact p-value with ties
```

a) The Spearman correlation coefficient between distance and untransformed luminosity is 0.04 with *p-value* 0.7612 and between distance and transformed luminosity is 0.04 with *p-value* 0.7612.
b) The Spearman correlation coefficient between distance and colour index is -0.09 with *p-value* 0.4728.
c) The Spearman correlation coefficient between colour index and untransformed luminosity is -0.69 with *p-value* 0 and between distance and transformed luminosity is -0.69 with *p-value* 0.

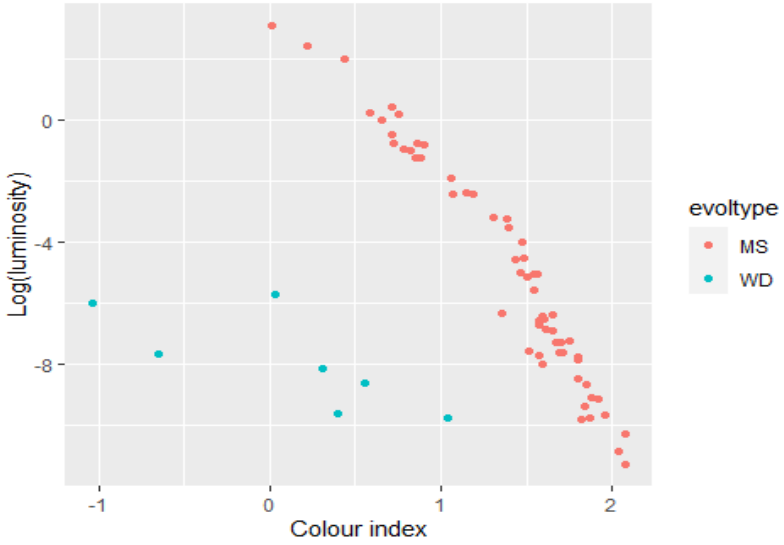
We have observed that the correlation between variables with untransformed / transformed luminosity is exactly the same. It is expected since correlation is invariant with scale or origin change.

Question 3

Solution:

The given chunk below will show a scatter plot of colour index versus transformed luminosity, colour-coded by "EvolType".

```
scatter <- ggplot(star, aes(x=colorindex, y=log_luminosity, color=evoltype))
+
geom_point()+
xlab("Colour index")+ ylab("Log(luminosity)")
scatter
```



Question 4

Solution:

In the given chunk below, we will build a linear regression model transformed luminosity on colour index using only stars with an evoltype of "WD".

```
wd_star <- star %>%
  filter(evoltype == "WD")
mod1 <- lm(log_luminosity ~ colorindex, data = wd_star)
res_mod1 <- get_regression_table(mod1)
#Table 1: Summary of regression parameter for stars with an evoltype of 'WD'
res_mod1
```

```
## # A tibble: 2 x 7
##   term       estimate std_error statistic p_value lower_ci upper_ci
##   <chr>         <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept     -7.77     0.45     -17.2    0        -8.93   -6.61
## 2 colorindex    -1.67     0.672     -2.49   0.055    -3.4     0.052
```

The model for stars with an evoltype of "WD" indicates not a statistically significant linear relationship between these variables at a 5% level of significant. The resulting *p-value*=0.055 is very much close to the significant threshold 0.05. Since stars with an evoltype of "WD" has small set of observations (7), it might shows significant relation at a 5% level of significant for large sample. However, we can say that the relation is significant at a 10% level of significant.

Question 5

Solution:

In the given chunk below we will build a linear regression model transformed luminosity on colour index using only stars with an evoltype of "MS".

```
ms_star <- star %>%
  filter(evoltype == "MS")
mod2 <- lm(log_luminosity ~ colorindex, data = ms_star)
res_mod2 <- get_regression_table(mod2)
#Table 2: Summary of regression parameter for stars with an evoltype of 'MS'
res_mod2
```

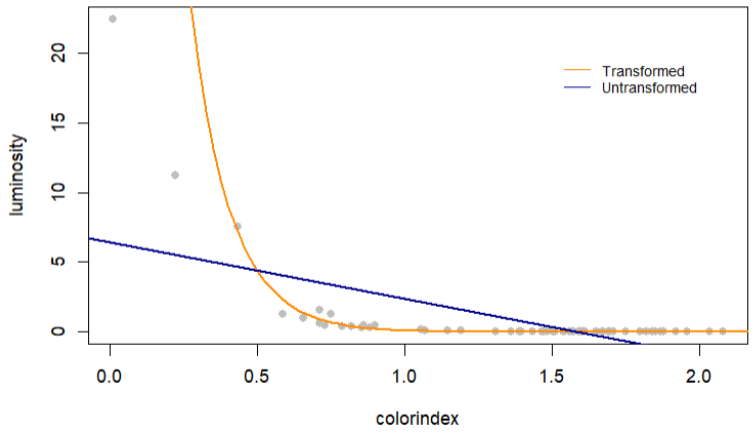
```
## # A tibble: 2 x 7
##   term       estimate std_error statistic p_value lower_ci upper_ci
##   <chr>         <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept      5.15     0.344      15.0       0     4.46     5.84
## 2 colorindex    -7.35     0.236     -31.2       0    -7.82    -6.88
```

The model for stars with an evoltype of "MS" indicate a statistically significant linear relationship between these variables at both 1 and 5% level of significant. The resulting *p-value* is 0.055.

Question 6

Solution:

```
mod0 <- lm(luminosity ~ colorindex, data = ms_star)
plot(luminosity ~ colorindex, data = ms_star, col = "grey",
     pch = 20, cex = 1.5)
curve(exp(mod2$coef[1] + mod2$coef[2] * x),
      from = 0, to = 2.5, add = TRUE, col = "darkorange", lwd = 2)
abline(mod0, col = "darkblue", lwd = 2)
legend(1.5, 20, legend=c("Transformed", "Untransformed"),
       col=c("darkorange", "darkblue"), lty=1, cex=0.8,
       box.lty=0)
```



a) The equation of the line transformed luminosity on colour index is

$$\log(\hat{y}(x)) = \widehat{\beta_0} + \widehat{\beta_0}x = 5.151 + -7.352x$$

b) If we back to the original scale of the data from log-scale, we obtain

$$\hat{y}(x) = \exp(\widehat{\beta_0})\exp(\widehat{\beta_0}x) = \exp(5.151)\exp(-7.352x) = 172.626\ e^{-7.352x}$$

c) The equation of the line luminosity on colour index is

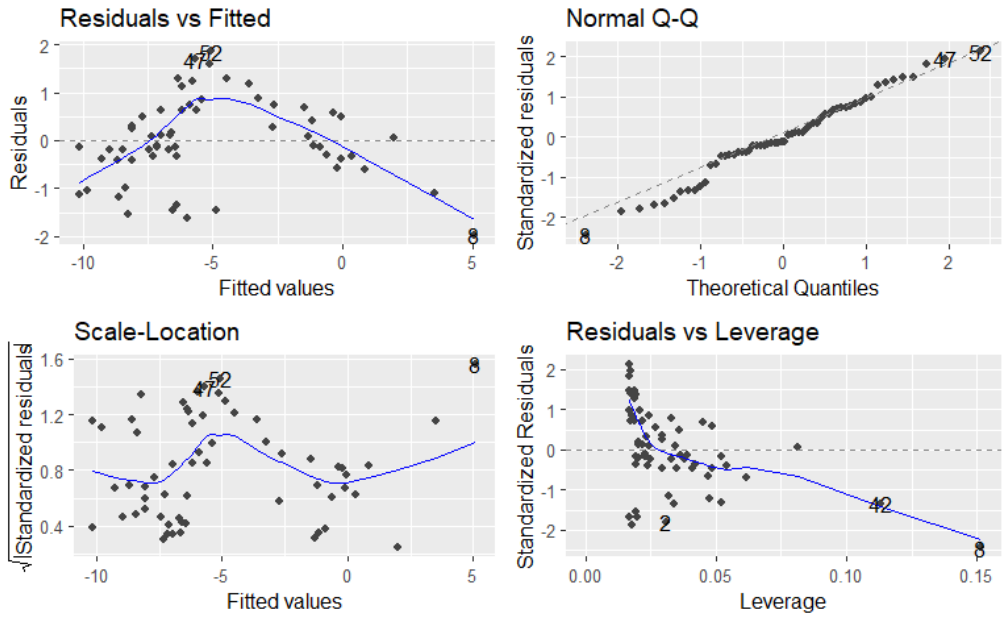
$$\hat{y}(x) = \widehat{\beta_0} + \widehat{\beta_0}x = 6.431 - 4.06x$$

d) Where, $y =$ luminosity and $x =$ colorindex.

Question 7

Solution:

```
autoplot(mod2)
```



a) **Residuals vs. fitted plot**: We observed a distinctive pattern like a parabola in the Residuals vs. fitted plot which implies a non-linear relationship between the variables.

b) **Normal quantile-quantile plot**: We observed in the Q-Q plot there is not a perfect but approximately a straight line which implies that the residuals are normally distributed.

c) **Scale-location plot**: We observed in the Scale-location plot the horizontal line with equally spread points meaning that the residuals are spread equally along the ranges of predictors (homoscedasticity).

d) **Residuals vs. leverage plot**: We observed in the residuals vs. leverage plot all cases are well inside of the Cook's distance lines (a blue line) which implies there is no influential observations.

By perceiving above diagnostic plot we conclude that the statistical assumptions of linear regression satisfied under this model.

Question 8

Solution:

```
SSY <- sum((ms_star$log_luminosity - mean(ms_star$log_luminosity))^2)
SSM <- sum((fitted(mod2) - mean(ms_star$log_luminosity))^2)
SSE <- sum(resid(mod2)^2)
```

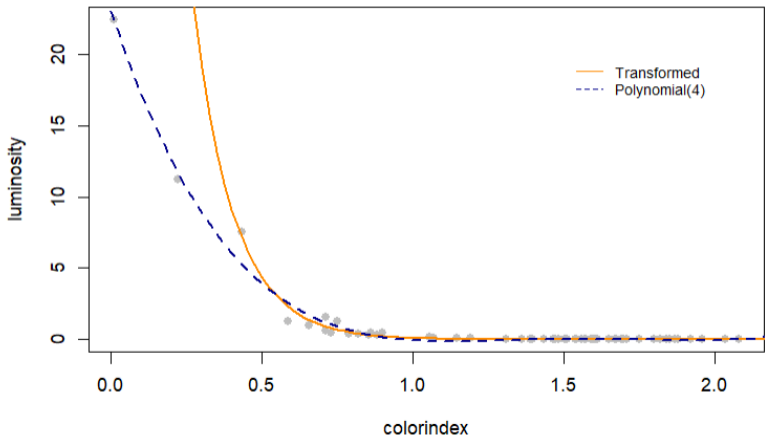
a) The **SSY** for the model is 794.9.
b) The **SSM** for the model is 750.07.
c) The **SSE** for the model is 44.83.

Question 9

Solution:

We consider a parametric 4^{th} order polynomial regression model.

```
mod3 = lm(luminosity ~ poly(colorindex, 4), data = ms_star)
plot(luminosity ~ colorindex, data = ms_star, col = "grey", pch = 20, cex =
1.5)
x_plot = seq(0, 2.5, by = 0.05)
curve(exp(mod2$coef[1] + mod2$coef[2] * x),
      from = 0, to = 2.5, add = TRUE, col = "darkorange", lwd = 2, lty = 1)
lines(x_plot, predict(mod3, newdata = data.frame(colorindex = x_plot)),
      col = "darkblue", lwd = 2, lty = 2)
legend(1.5, 20, legend=c("Transformed", "Polynomial(4)"),
       col=c("darkorange", "darkblue"), lty=1:2, cex=0.8,
       box.lty=0)
```




```
SSM <- sum((fitted(mod3) - mean(ms_star$luminosity))^2)
SSE <- sum(resid(mod3)^2)
```

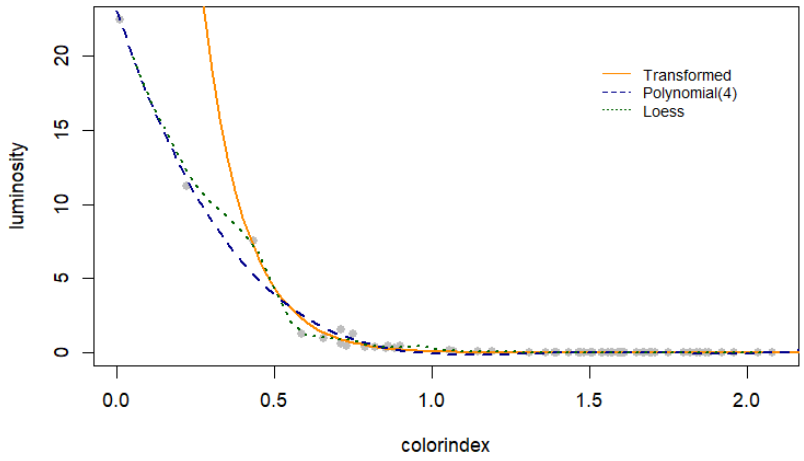
The **SSM** and **SSE** for the polynomial regression model of order 4^{th} are 641.66 and 8.92 respectively. It is clear that the polynomial regression increase **SSM** and decrease **SSE** comparative to the transform luminosity model. Therefore, the polynomial regression improve in the quality of the fit which is also shown in Figure 6.

Question 10

Solution:

We applied a non-parametric approach named Local Regression also called Loess regression that fits multiple regressions in local neighborhood.

```
mod4 <- loess(luminosity ~ colorindex, data = ms_star, span=0.10) # 10%
smoothing span

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1.75

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

plot(luminosity ~ colorindex, data = ms_star, col = "grey", pch = 20, cex =
1.5)
x_plot = seq(0, 2.5, by = 0.05)
curve(exp(mod2$coef[1] + mod2$coef[2] * x),
      from = 0, to = 2.5, add = TRUE, col = "darkorange", lwd = 2, lty = 1)
lines(x_plot, predict(mod3, newdata = data.frame(colorindex = x_plot)),
      col = "darkblue", lwd = 2, lty = 2)
lines(x_plot, predict(mod4, newdata = data.frame(colorindex = x_plot)),
      col = "darkgreen", lwd = 2, lty = 3)
legend(1.5, 20, legend=c("Transformed", "Polynomial(4)","Loess"),
       col=c("darkorange", "darkblue","darkgreen"), lty=1:3, cex=0.8,
       box.lty=0)
```




```
SSM <- sum((fitted(mod4) - mean(ms_star$luminosity))^2)
SSE <- sum(resid(mod4)^2)
```

The **SSM** and **SSE** for the nonparametric Loess regression model are 661.09 and 2.03 respectively.