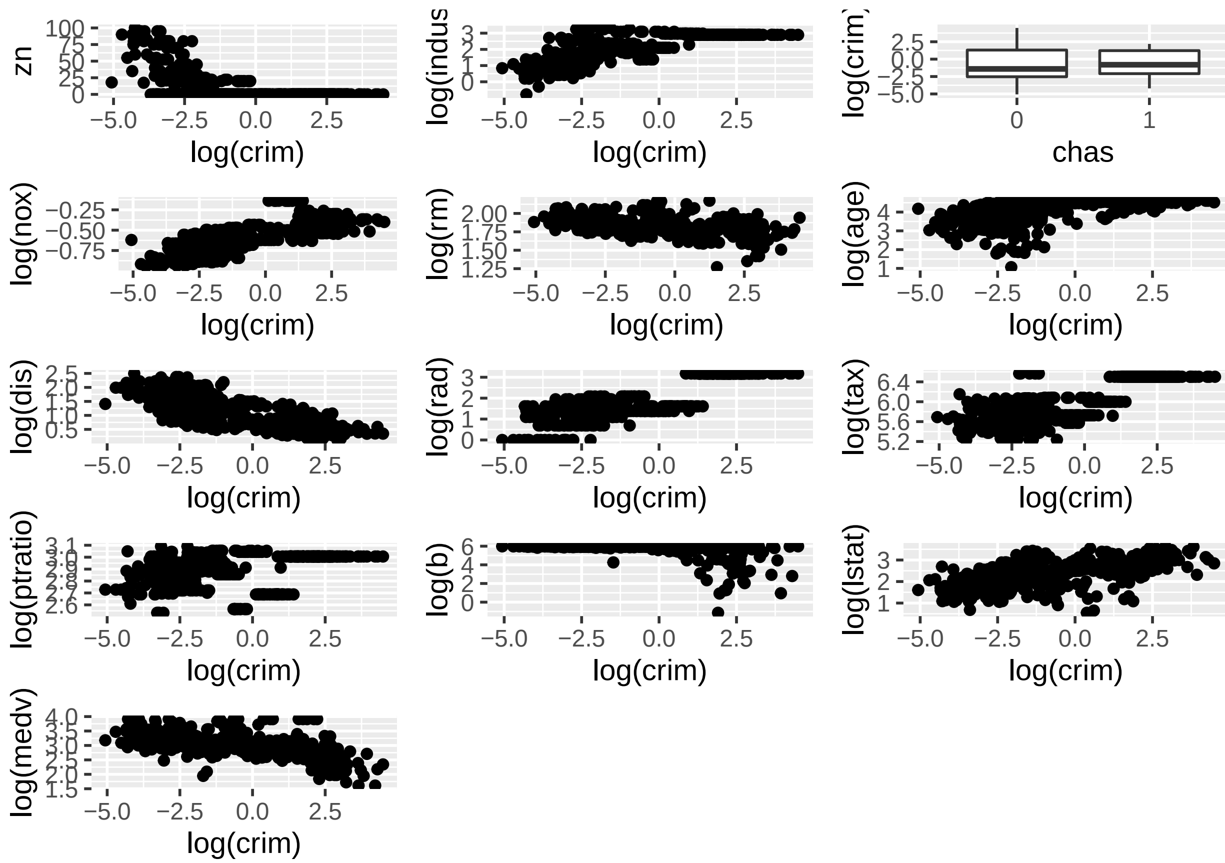
Boston-Housing Data

Fit a model that predicts per capita crime rate by town (crim) using only one predictor variable. Use plots to justify your choice of predictor variable and the appropriateness of any transformations you use

```
library(modelr)
library(readr)
library(dplyr)
library(ggplot2)
library(mlbench)

data(BostonHousing)


data <- BostonHousing %>% as_tibble()

p1 <- ggplot(data) +geom_point(aes(log(crim), zn))
p2 <- ggplot(data) +geom_point(aes(log(crim),log(indus)))
p3 <- ggplot(data) +geom_boxplot(aes(y = log(crim), x = chas))
p4 <- ggplot(data) +geom_point(aes(log(crim), log(nox)))
p5 <- ggplot(data) +geom_point(aes(log(crim), log(rm)))
p6 <- ggplot(data) +geom_point(aes(log(crim),log( age)))
p7 <- ggplot(data) +geom_point(aes(log(crim), log(dis)))
p8 <- ggplot(data) +geom_point(aes(x =log(crim), y = log(rad)))
p9 <- ggplot(data) +geom_point(aes(log(crim), log(tax)))
p10 <- ggplot(data) +geom_point(aes(log(crim), log(ptratio)))
p11 <- ggplot(data) +geom_point(aes(log(crim), log(b)))
p12 <- ggplot(data) +geom_point(aes(log(crim), log(lstat)))
p13 <- ggplot(data) +geom_point(aes(log(crim), log(medv)))

gridExtra::grid.arrange(p1,p2, p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13, ncol =3)
```




```
#Dis variable seems to have strong relationship with crim
fit1 <- lm(log(crim) ~ log(dis), data)

rmse(fit1, data)
```

```
## [1] 1.443389
```

```
#Parameters of fitted models

coef(fit1)
```

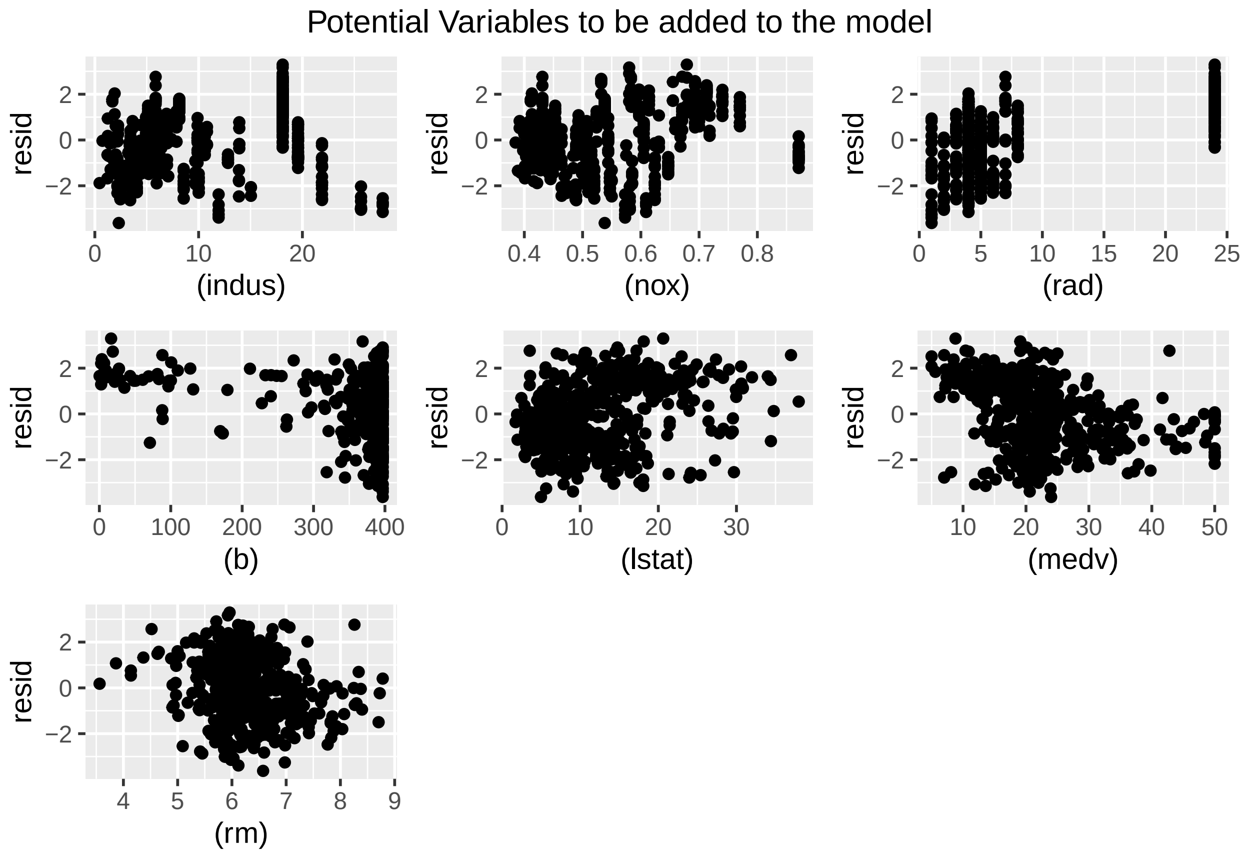
```
## (Intercept)    log(dis)
##    2.761124   -2.981030
```

Plot the residuals of the fitted model

```
#PLOTTING RESIDUALS FOR ALL THE VARIABLES
a1 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (zn), y = resid))+
  geom_point()

a2 <- data %>%
  add_residuals(fit1) %>%
```

```
  ggplot(aes(x = (indus), y = resid))+
  geom_point()

a3 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (chas), y = resid))+
  geom_boxplot()

a4 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (nox), y = resid))+
  geom_point()

a5 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = age, y = resid))+
  geom_point()


a6 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (rad), y = resid))+
  geom_point()

a7 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (tax), y = resid))+
  geom_point()

a8 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (ptratio), y = resid))+
  geom_point()

a9 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (b), y = resid))+
  geom_point()

a10 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (lstat), y = resid))+
  geom_point()


a11 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (medv), y = resid))+
  geom_point()

a12 <- data %>%
  add_residuals(fit1) %>%
  ggplot(aes(x = (rm), y = resid))+
```

```
  geom_point()


gridExtra::grid.arrange(a2, a4, a6,  a9, a10, a11,a12, ncol = 3, top = grid::textGrob("Potential Variab
```



Zn - No pattern Found indus - weak negative relation chas - No Pattern Found nox - weak positive relation age - No Pattern Found rad - positive relation tax - No pattern found ptratio - No pattern found b - non-linear relation lstat - weak positive medv -weak negative relation rm - non linear relation

Fit a new model for predicting per capita crime rate by town, adding or removing variables based on the residual plots

```
#fitting variables Of observed patterns
fit2 <- lm(log(crim) ~ log(dis) + log(indus) + log(nox) + log(rad)+ log(b)+ log(lstat)
               + log(medv) + log(rm), data)



rmse(fit2, data)
```

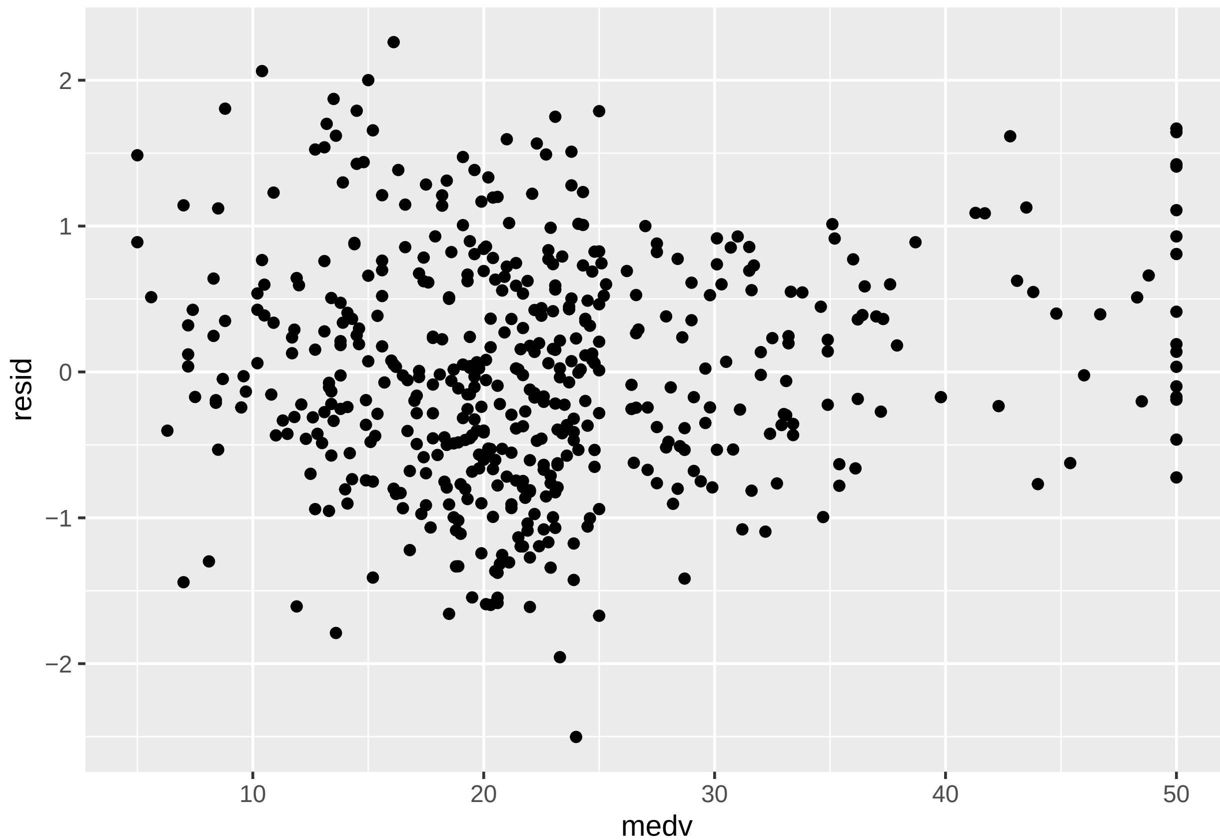
```
## [1] 0.7935773
```

```
#Trying to adjust model by removing non-linear patterned variables and making transformations

fit3 <- lm(log(crim) ~ (dis) + log(indus) + log(nox) + (rad)+ log(lstat)
               + log(medv) , data )
```

```r
rmse(fit3, data)
```

```
## [1] 0.7854267
```

```r
#New model seems to work better

#plotting against residuals for checking scatterness
data %>%
  add_residuals(fit3) %>%
 ggplot(aes(x = medv, y = resid))+
  geom_point()
```




```r
#All the predictors seems to be showing good relation with the response variable

summary(fit3)
```

```
##
## Call:
## lm(formula = log(crim) ~ (dis) + log(indus) + log(nox) + (rad) +
##     log(lstat) + log(medv), data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.50242 -0.53418 -0.03344  0.54717  2.26126
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.23775    0.79858  -0.298   0.7660
## dis         -0.07274    0.03080  -2.362   0.0186 *
## log(indus)   0.36340    0.07788   4.666 3.94e-06 ***
## log(nox)     2.96335    0.35727   8.294 1.02e-15 ***
## rad          0.13280    0.00541  24.545  < 2e-16 ***
## log(lstat)   0.10268    0.11260   0.912   0.3623
## log(medv)   -0.24898    0.16040  -1.552   0.1212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7909 on 499 degrees of freedom
## Multiple R-squared:  0.8678, Adjusted R-squared:  0.8662
## F-statistic: 545.8 on 6 and 499 DF,  p-value: < 2.2e-16
```

New model performs better than previous with lower rmse and high R- squared value. Scattered pattern is found when plotted against residuals. summary shows almost all the variables are significant with relatively lower p-values.

cross-validation for a linear model (fit using lm) and returns the average root-mean-square-error across all folds.

```
f1 <- function(a, dataset, k){

  set.seed(1)
  data_cv <- crossv_kfold(dataset,k)
  data_cv <- data_cv %>%
    mutate(fit = map(train, ~lm(formula = a, data = .)),
           #pred = map2(test, ~add_predictions(fit, data = .)),
           rmse_test = map2_dbl(fit, test, ~rmse(.x,.y)))
  print(mean(data_cv$rmse_test))

}

f1(arr_delay ~ month, flights, 3)
```

```
## [1] 44.62699
```

Using 5-fold cross-validation…

```
#model fitted in HW5
f1(log(crim) ~ (dis) + log(indus) + log(nox) + (rad)+ log(lstat) + log(medv), BostonHousing, 5)
```

```
## [1] 0.7959519
```

```
#Trying new models
f1(log(crim) ~ (dis) + log(indus) + log(nox) + (rad)+ log(lstat) + log(medv), BostonHousing, 3)
```

```
## [1] 0.7985524
```

```r
f1(log(crim) ~  (rad)+ log(lstat) + log(medv), BostonHousing, 5)
```

```
## [1] 1.019802
```