```r
#1.)
pmsm_temperature_data <- read.csv("~/Desktop/R/Final Project/pmsm_temperature_data.csv")
pmsm_temperature_data <- na.omit(pmsm_temperature_data)

View(pmsm_temperature_data)

#2.)
pmsm_temperature_data$stator_yoke <- as.factor(pmsm_temperature_data$stator_yoke)

#3.)
summary(pmsm_temperature_data$pm)
sd(pmsm_temperature_data$pm)
```

Minimum: -2.63199

Maximum: 2.91746

Mean: -0.03654

Median: 0.04715

standard deviation: 1.053009

25th percentile: -0.77350

50th percentile: 0.04715

75th percentile: 0.70070

```r
#4.)
summary(pmsm_temperature_data$motor_speed)
sd(pmsm_temperature_data$motor_speed)
```

Minimum: -1.37153

Maximum: 2.02416

Mean: 0.07495

Median: -0.04055

standard deviation: 0.9804036

25th percentile: -0.80668

50th percentile: -0.04055

75th percentile: 0.93933

```r
#5.)
correlation <- cor(pmsm_temperature_data$motor_speed, pmsm_temperature_data$pm)
print(correlation)
```

Correlation Coefficient: 0.4147886

0.4147886 is closer to 0 than 1, although not by much, so it is not a strong correlation.

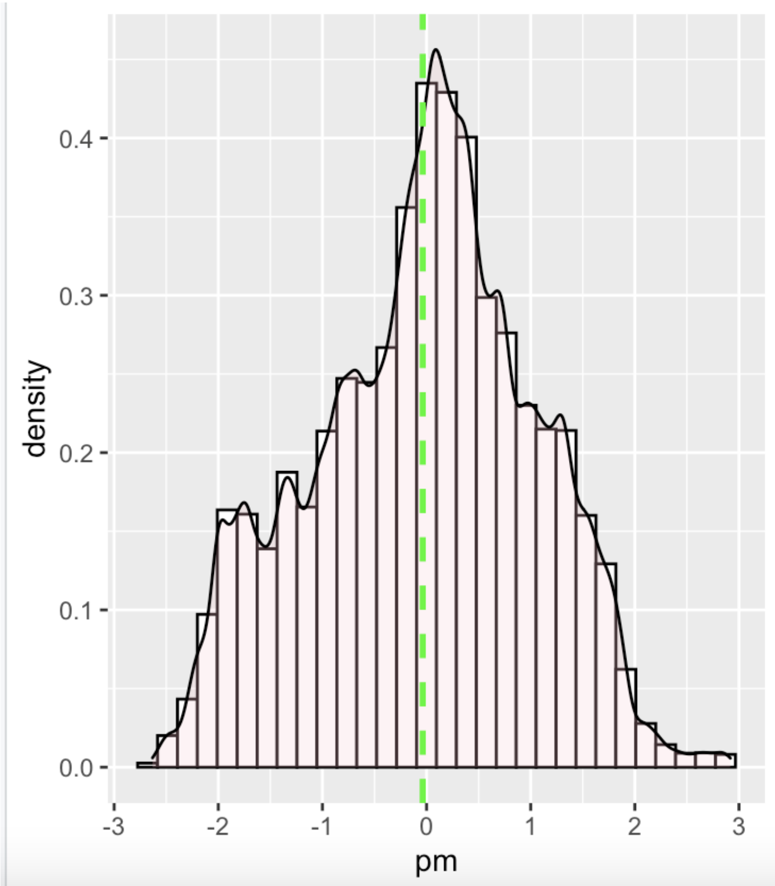
```r
#6.)
yoke_table <- table(pmsm_temperature_data$stator_yoke)

sorted_yoke <- sort(yoke_table, decreasing = TRUE)[1]
mode_yoke <- sorted_yoke[1]
mode_yoke
```

Frequency Table:

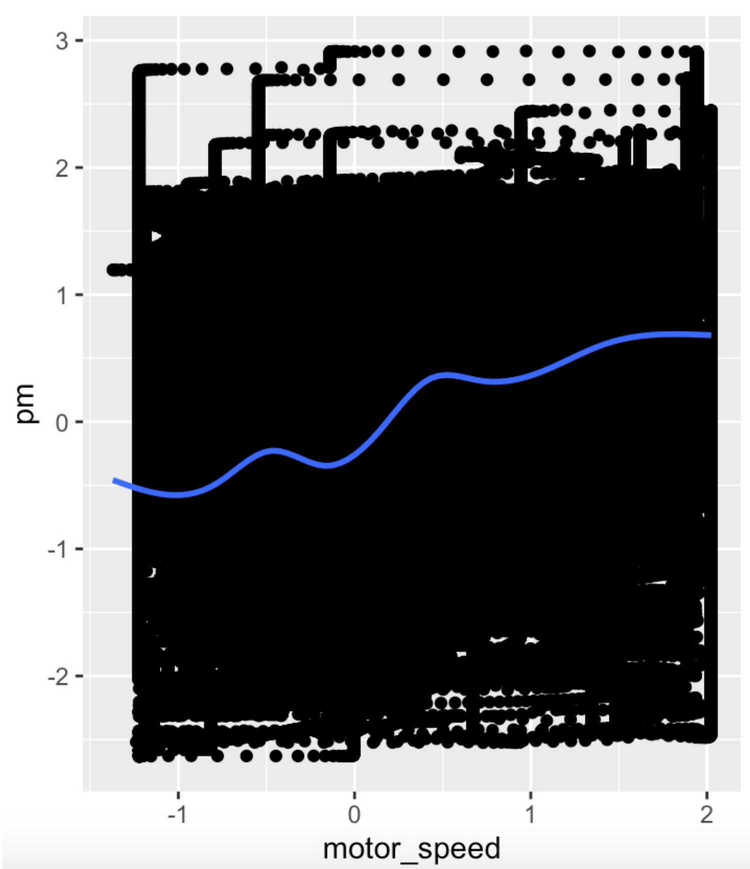
```
Negative Positive
  429137   321584
```

Mode: Negative

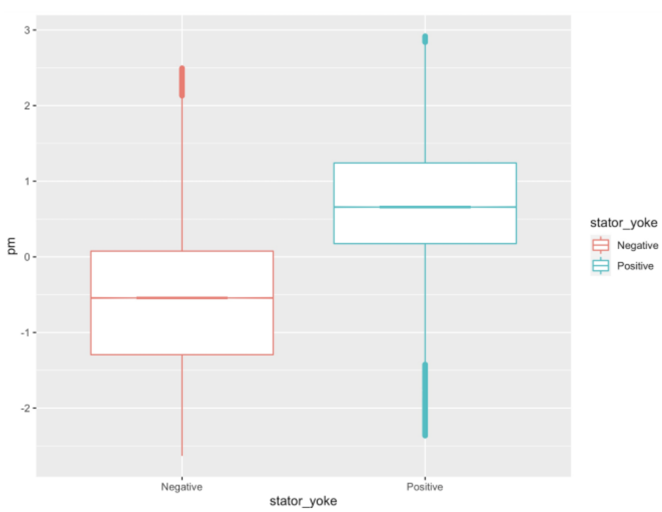
```r
#7.)
library(ggplot2)

ggplot(data = pmsm_temperature_data, aes(x=pm))+
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="pink") + geom_vline(aes(xintercept=mean(pm)),
            color="green", linetype="dashed", size=1)
```




```r
#8.)
ggplot(data = pmsm_temperature_data, aes(x = motor_speed, y=pm))+
  geom_point()+geom_smooth()
```




```
#9.)
ggplot(data=pmsm_temperature_data, aes(x=stator_yoke, y=pm)) +
  geom_boxplot(aes(col= stator_yoke ), notch = TRUE)
ggsave("~/Desktop/R/Final Project/pmyoke.jpg", width = 20, height = 15, units = "cm")
```



```r
#10.)
#a
lm.result1 <- lm(pm ~ ambient + coolant + motor_speed + torque, data= pmsm_temperature_data)
summary(lm.result1)$coefficients
```

Pm = -0.003930971 + 0.345750361 * ambient + 0.301152316 * coolant + 0.399119596 * motor_speed + 0.083694444 * torque

```r
#b
lm.result2 <- lm(pm ~ ambient + coolant + u_d + motor_speed + torque + stator_winding, data= pmsm_temperature_data)
summary(lm.result2)$coefficients
```

Pm = -0.006636084 + (0.280087056 * ambient) + (-0.010662764 * coolant) + (-0.270032724 * u_d) + (0.054033292 * motor_speed) + (-0.284207277 * torque) + (0.616822296 * stator_winding)

```r
#c
lm.result3 <- lm(pm ~ ambient + coolant + u_d + u_q + motor_speed + torque + stator_yoke + stator_winding, data= pmsm_temperature_data)
summary(lm.result3)$coefficients
```

Pm = -0.11915072 + (0.28312684 * ambient) + (-0.05159457 * coolant) + (-0.25151648 * u_d) + (-0.06007881 * coolant) + (0.11421214 * motor_speed) + (-0.26052010 * torque) + (0.24344746 * stator_yoke) + (0.53693677 * stator_winding)

```r
#d
summary(lm.result1)$adj.r.squared
summary(lm.result2)$adj.r.squared
summary(lm.result3)$adj.r.squared
```

```
> summary(lm.result1)$adj.r.squared
[1] 0.4541997
> summary(lm.result2)$adj.r.squared
[1] 0.6738776
> summary(lm.result3)$adj.r.squared
[1] 0.6791595
```

I would recommend model 3 because it’s adjusted r squared value is higher than the other two models.

```r
#11.)
library(class)
pmsm_temperature_data$stator_yoke <- as.factor(pmsm_temperature_data$stator_yoke)

predictors <- c("pm", "ambient", "coolant", "u_d", "u_q", "motor_speed", "torque")
data.predictors <- pmsm_temperature_data[predictors]
data.target <- pmsm_temperature_data$stator_yoke

#a
sample.size <- floor(0.85*nrow(pmsm_temperature_data))

train <- data.predictors[1:sample.size, ]
test <- data.predictors[-c(1:sample.size), ]

#b
cl <- data.target[1:sample.size]
knn.test.predict1 <-knn(train[1:3], test[1:3], cl, k=5)

test.label1 <- data.target[-c(1:sample.size)]
table(test.label1, knn.test.predict1)
```

```
             knn.test.predict1
test.label1 Negative Positive
   Negative    46527    12838
   Positive     4603    48641
```

(46527 + 48641) / (46527 + 48641 + 12838 + 4603) = 84.51%

```r
#c
knn.test.predict2 <-knn(train[1:5], test[1:5], cl, k=5)

test.label2 <- data.target[-c(1:sample.size)]
table(test.label2, knn.test.predict2)
```

```
             knn.test.predict2
test.label2 Negative Positive
   Negative    45355    14010
   Positive     4369    48875
```

(45355 + 48875) / (45355 + 48875 + 14010 + 4369) = 83.68%

```r
#d
knn.test.predict3 <-knn(train, test, cl, k=5)

test.label3 <- data.target[-c(1:sample.size)]
table(test.label3, knn.test.predict3)
```

```
            knn.test.predict3
test.label3 Negative Positive
   Negative    44147    15218
   Positive     9917    43327
```

(44147 + 43327) / (44147 + 43327 + 9917 + 15218) = 77.68%

e.)

Based off these tests, model1 (using Pm, Ambien, and Coolant) is the most accurate, making it my recommendation. It also took less time to run than the others.