Stat230 - HW 3

Owen Forman

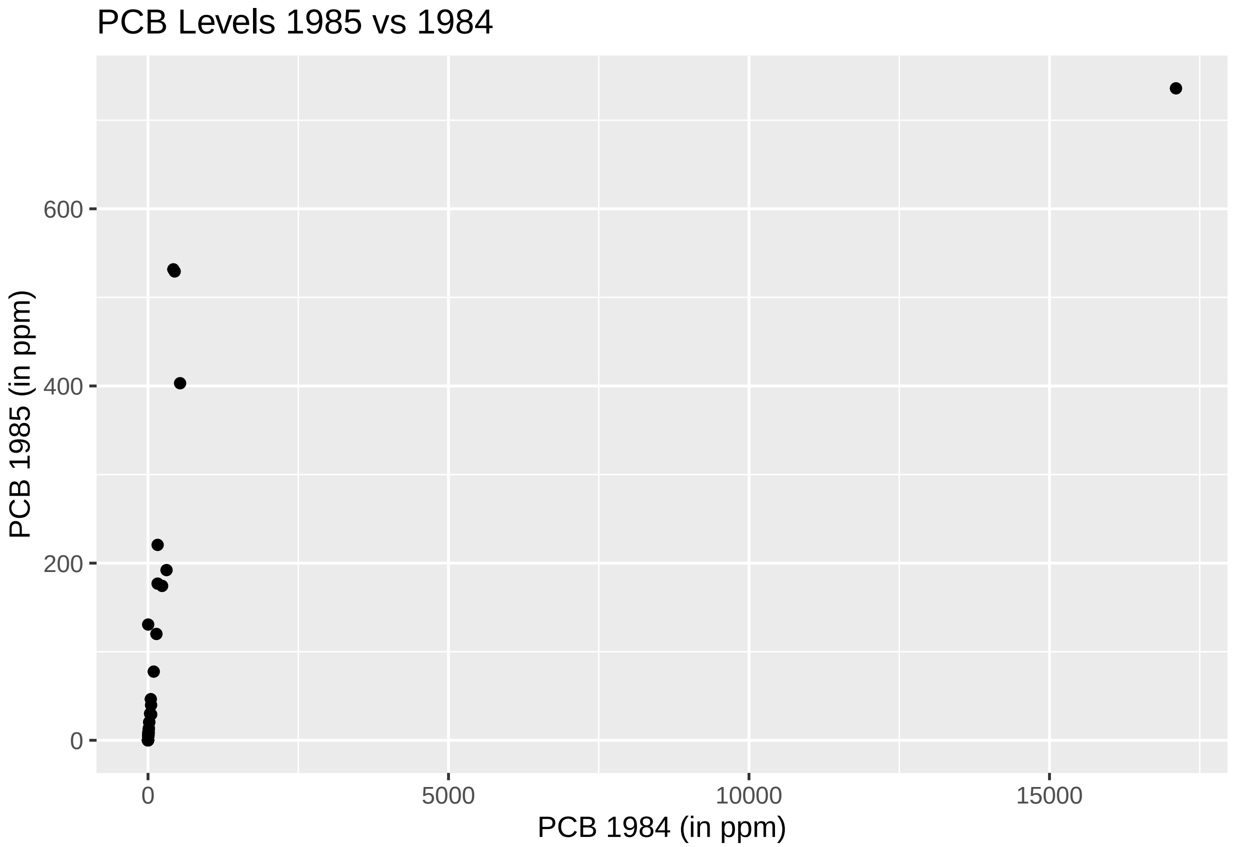
```r
#loads necessary libraries
library(ggplot2)
library(patchwork)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(broom)
```

2a) **Answer:** The extreme outlier case was Boston Harbor, which had extremely high pcb levels in 1984. Upon removing the outlier though, it still turns out that the data is not yet suited for a SLR model. We can see this visually in the scatter plot "PCB Levels 1985 vs 1984 (No Outlier)" since the trend does not appear to be linear. Additionally, after attempting to fit a SLR model, it becomes even clearer that the model is not appropriate, as the residual plot"Pcb Resid plot (No outliers)" shows clear grouping around x = 0 and residuals become more widespread at larger x's which is a clear patter. Thus an SLR model cannot be used despite removing the outlier.

```r
#reads in data
pcb <- read.csv("https://www.math.carleton.edu/ckelling/data/Pcb.csv")

#creates scatterplot of pcb85 vs pcb84.
ggplot(data = pcb, aes(x = pcb84, y = pcb85)) +
  geom_point() +
  labs(title = "PCB Levels 1985 vs 1984", x = "PCB 1984 (in ppm)", y ="PCB 1985 (in ppm)")
```



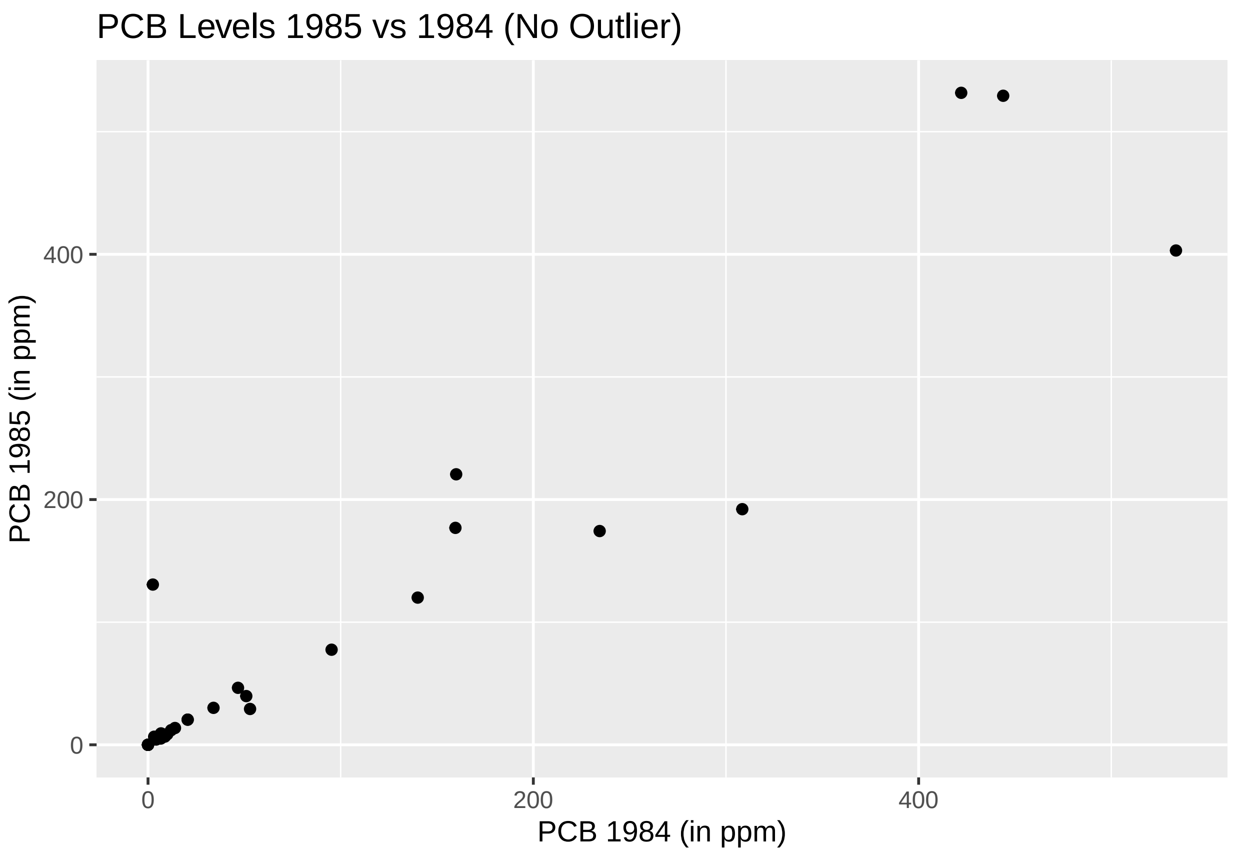
```
#identifies extreme outlier value
which(pcb$pcb84 > 10000)
```

```
## [1] 4
```

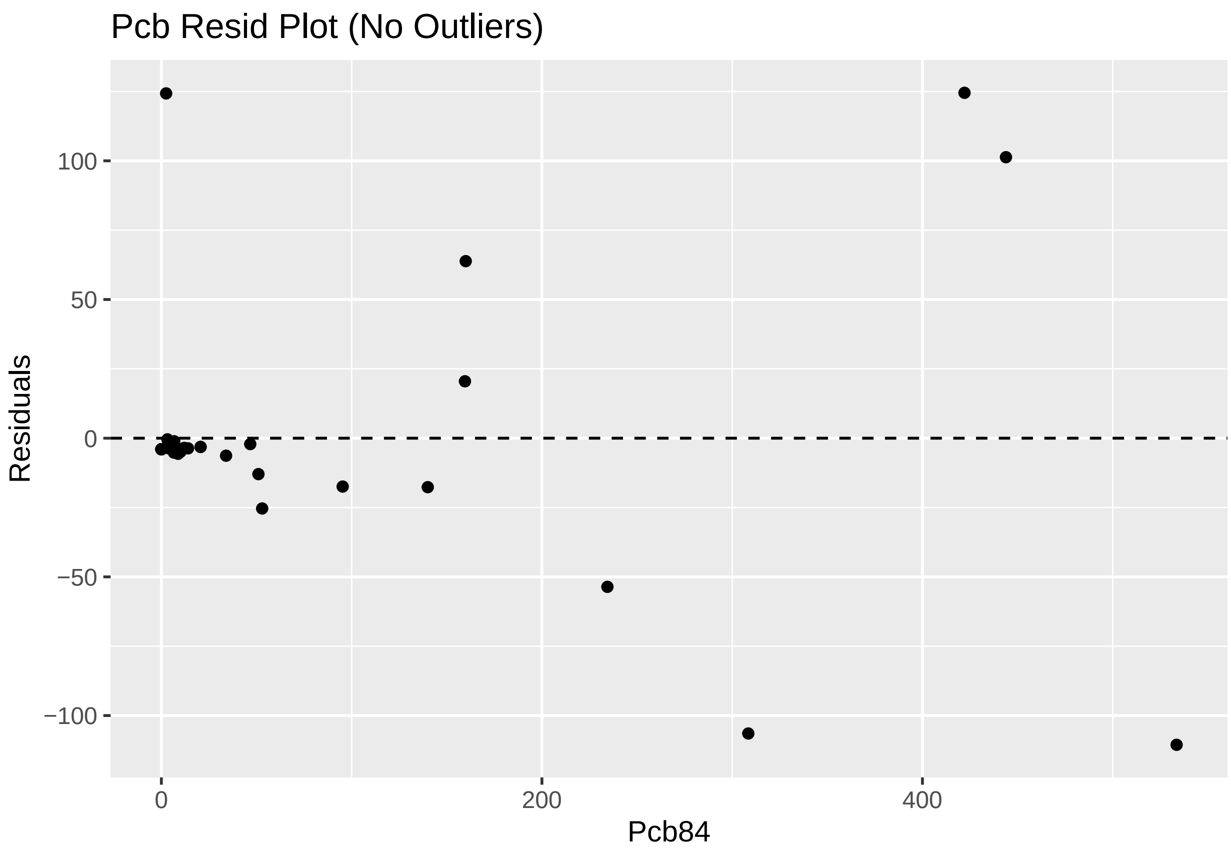
```
#prints row with outlier value
pcb[4,]
```

```
##             Site    pcb84 pcb85
## 4 Boston Harbor 17104.86   736
```

```
#removes outlier value (note that this data point was Boston Harbor)
pcb_new <- pcb[-4,]

#creates new scatterplot of pcb85 vs pcb84 - no outlier included
ggplot(data=pcb_new, aes(x = pcb84, y = pcb85)) +
  geom_point() +
    labs(title = "PCB Levels 1985 vs 1984 (No Outlier)", x = "PCB 1984 (in ppm)", y ="PCB 1985 (in ppm)
```



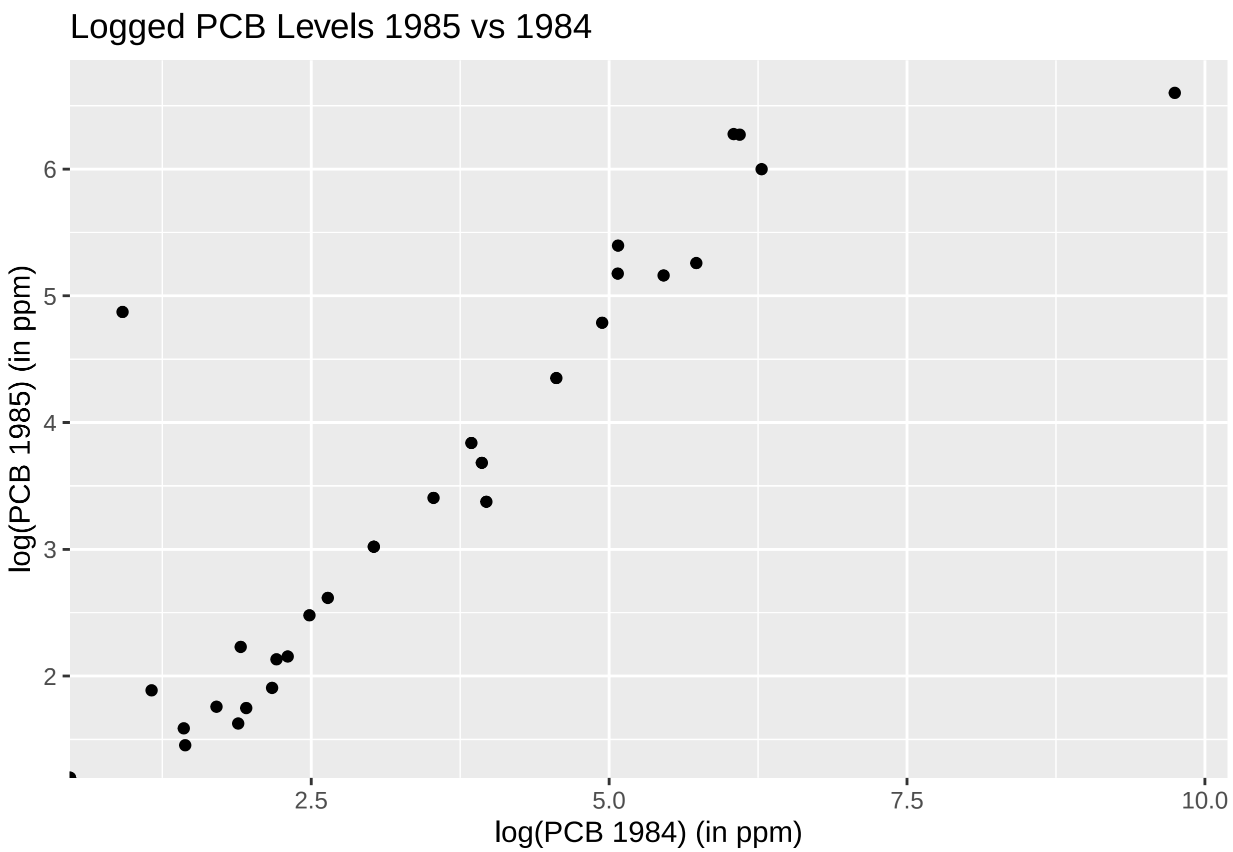
```r
pcb_new %>%
  #creates SLR model on the new no outlier data
  lm(data = ., pcb85 ~ pcb84) %>%

  #augments model
  augment(.) %>%

  #creates and prints resid plot for model
  ggplot(., aes(x = pcb84, y = .resid)) +
    geom_point() +
    geom_hline(yintercept = 0, linetype = "dashed") +
    labs(title = "Pcb Resid Plot (No Outliers)", y = "Residuals", x = "Pcb84")
```



2b)

Answer: While we could not justify fitting a SLR model onto the previous graph in 2a), we can justify this model being used in the "Logged PCB Levels 1985 vs 1984" seen below. This is because the data has been transformed using the log function, which has uncovered the linear relationship between pcb84 and pcb85. This can clearly be seen visually, as the trend went from following an exponential line of some kind, to a more controlled linear line. While there are still outliers (which will be assessed in the coming problems), there are only really two or three of note, which may be the result of key information left out of this analysis. Though the reason these points are so extreme cannot be determined, it is safe to say given the low number of outliers, and the extremely strong linear relationship the rest of the data appears to follow, that we can still reasonably use an SLR model despite these outlier points.

```
#creates scatterplot of log(pcb85) vs log(pcb84).
ggplot(data=pcb, aes(x = log(pcb84), y = log(pcb85))) +
  geom_point() +
  labs(title = "Logged PCB Levels 1985 vs 1984", x = "log(PCB 1984) (in ppm)", y ="log(PCB 1985) (in pp
```



Answer: Some of the values = -Inf because the test sites had 0 listed for pcb levels in 1984 and 1985. Since the lim x -> 0 lnx = -Inf it is clear to see that when we too the natural log of these 0 values R spits back -Inf as the answer, since ln(0) is technically undefined.

```
#creates mutated dataset with log(pcb85) and log(pcb84) values
pcb_log <- pcb %>%
  mutate(., pcb84 = log(pcb84), pcb85 = log(pcb85))

#finds which of these values are less than 0, aka -Inf
which(pcb_log$pcb84 < 0)
```

```
## [1] 12 14 16 18 19 21 22 23
```

```
which(pcb_log$pcb85 < 0)
```

```
## [1] 12 14 16 18 19 21 22 23
```

```
#finds which of the original (non-logged) values are = 0
which(pcb$pcb84 == 0)
```

```
## [1] 12 14 16 18 19 21 22 23
```

```
which(pcb$pcb85 == 0)
```

```
## [1] 12 14 16 18 19 21 22 23
```

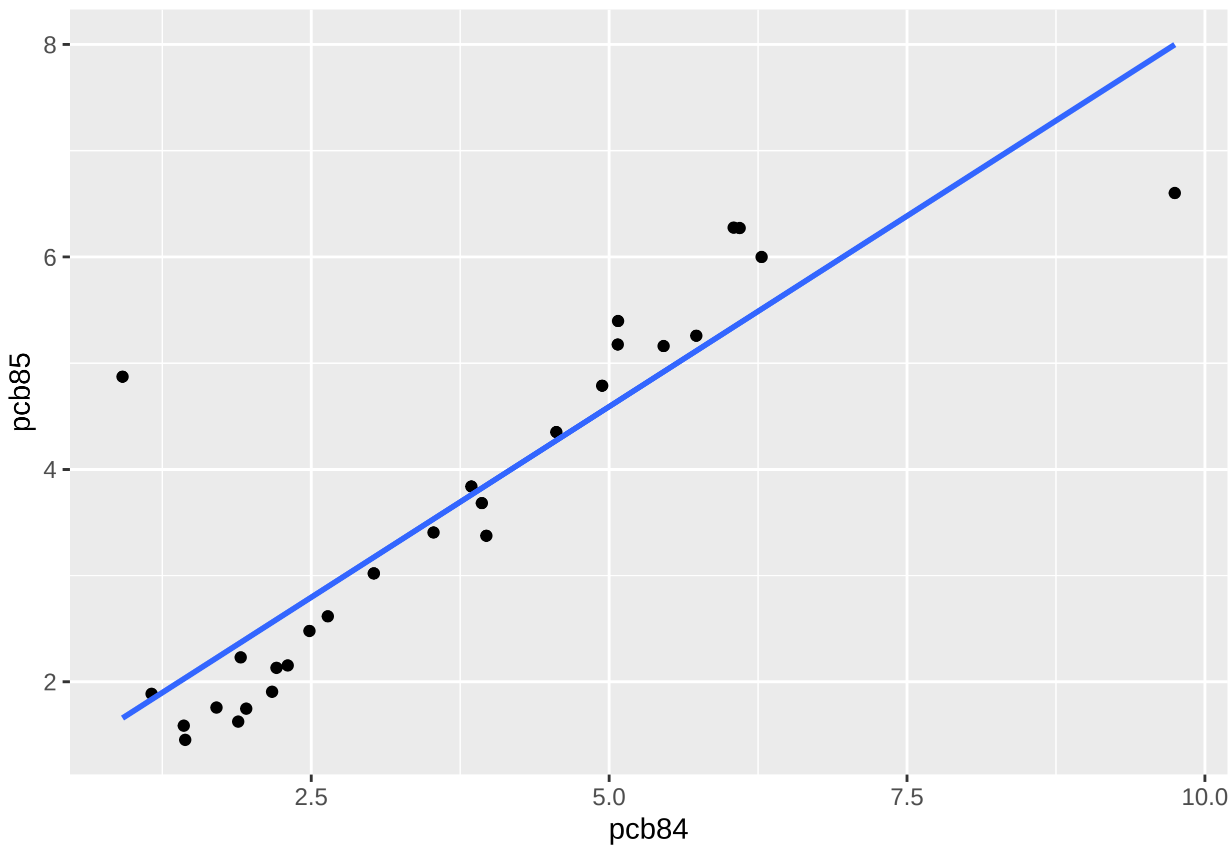
```
#locates which sites had -Inf values and prints them
pcb_log[c(12,14,16,18,19,21,22,23),]
```

```
##                    Site pcb84 pcb85
## 12     Pamilico Sound  -Inf  -Inf
## 14       Sapelo Sound  -Inf  -Inf
## 16          Tampa Bay  -Inf  -Inf
## 18         Mobile Bay  -Inf  -Inf
## 19       Round Island  -Inf  -Inf
## 21      Barataria Bay  -Inf  -Inf
## 22    San Antonio Bay  -Inf  -Inf
## 23 Corpus Christi Bay  -Inf  -Inf
```

```r
#filters -Inf values out
pcb_log_filtered <- pcb_log %>%
  filter(., pcb84 > 0, pcb85 >0)
```

```r
#creates scatterplot of log(pcb85) vs log(pcb84) filtered to have no -Inf values.
ggplot(data=pcb_log_filtered, aes(x = pcb84, y = pcb85)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

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

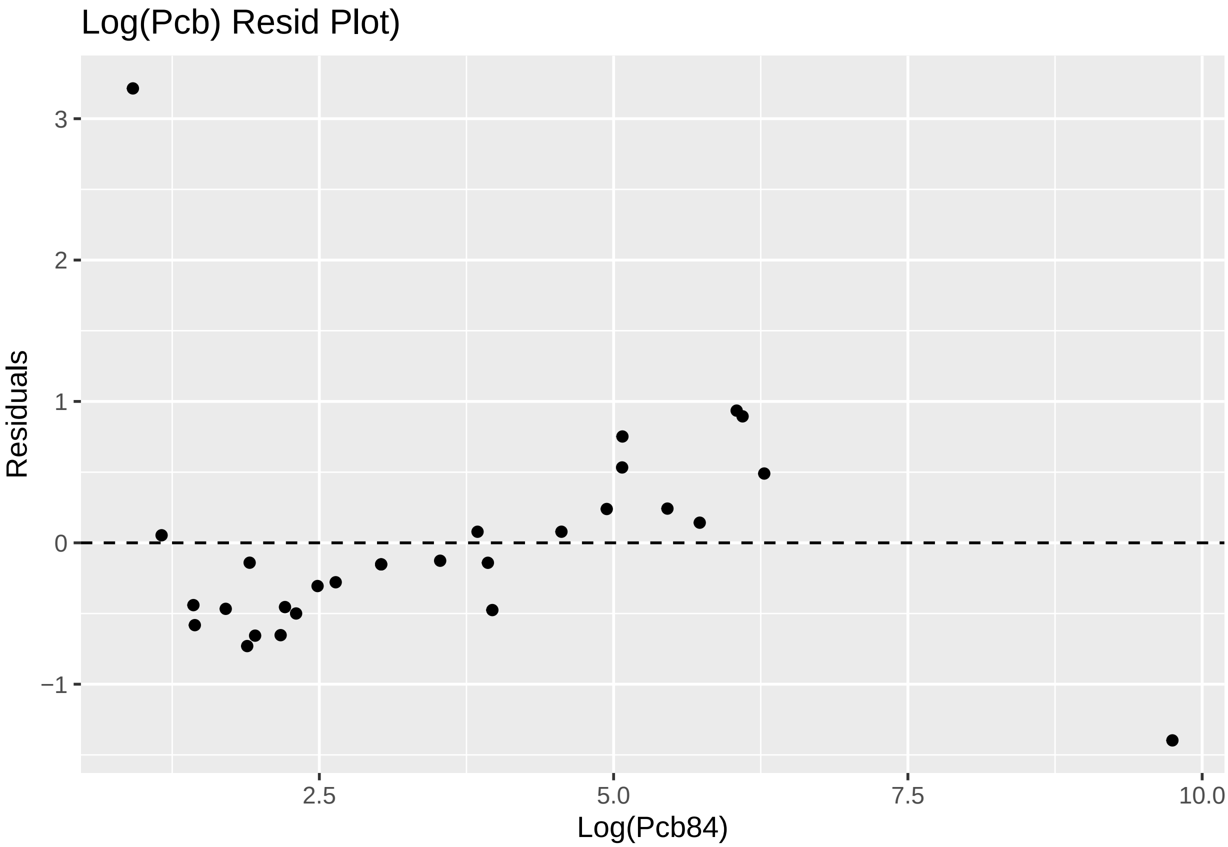


```r
  labs(title = "Filtered logged PCB Levels 1985 vs 1984", x = "log(PCB 1984) (in ppm)", y ="log(PCB 198
```

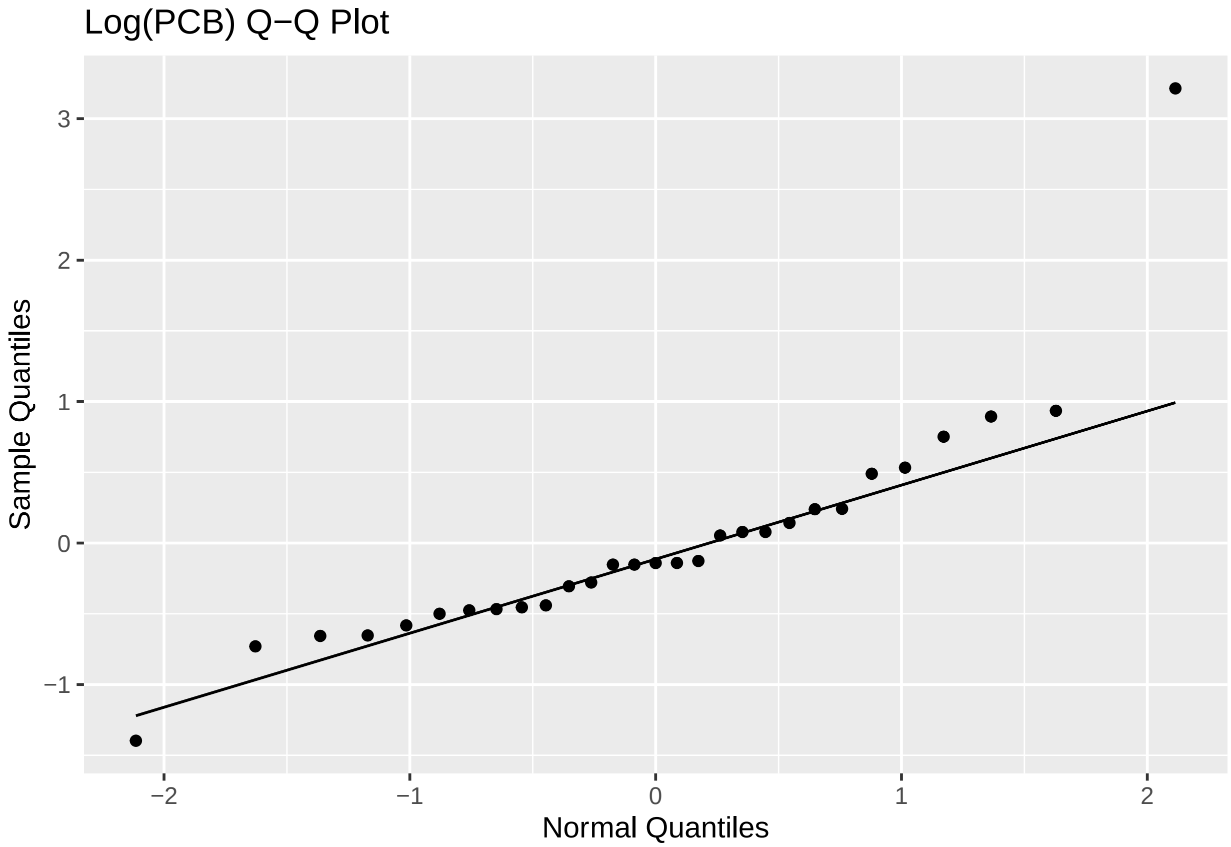
```
## $x
## [1] "log(PCB 1984) (in ppm)"
##
## $y
## [1] "log(PCB 1985) (in ppm)"
##
```

```
## $title
## [1] "Filtered logged PCB Levels 1985 vs 1984"
##
## attr(,"class")
## [1] "labels"
```

```
pcb_log_filtered_fit <- lm(data = pcb_log_filtered, pcb85 ~ pcb84)

pcb_log_filtered_aug <- augment(pcb_log_filtered_fit)

ggplot(pcb_log_filtered_aug, aes(x = pcb84, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Log(Pcb) Resid Plot)", y = "Residuals", x = "Log(Pcb84)")
```



```
ggplot(pcb_log_filtered_aug, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line() +
  labs(title = "Log(PCB) Q-Q Plot", y = "Sample Quantiles", x = "Normal Quantiles")
```



```
#finds which entries have resid > 2 (upper outlier) and resid < -1 (lower outlier)
which(pcb_log_filtered_aug$.resid > 2 | pcb_log_filtered_aug$.resid < -1)
```

```
## [1]  4 10
```

```
#prints out the entries (note we found which ones they were in the augmented dataset, and then locate t
pcb_log_filtered[c(4,10),]
```

```
##              Site    pcb84    pcb85
## 4  Boston Harbor 9.7471179 6.601230
## 10  Delaware Bay 0.9162907 4.872675
```