bikes

##Load and examine data. Bike usage versus hour of day.

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   1.0.0     v dplyr   0.8.3
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0

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

```
library(readr)
library(dplyr)


bike_data = read.csv("london_merged.csv")

head(bike_data)
```

```
##             timestamp cnt  t1  t2   hum wind_speed weather_code is_holiday
## 1 2015-01-04 00:00:00 182 3.0 2.0  93.0        6.0            3          0
## 2 2015-01-04 01:00:00 138 3.0 2.5  93.0        5.0            1          0
## 3 2015-01-04 02:00:00 134 2.5 2.5  96.5        0.0            1          0
## 4 2015-01-04 03:00:00  72 2.0 2.0 100.0        0.0            1          0
## 5 2015-01-04 04:00:00  47 2.0 0.0  93.0        6.5            1          0
## 6 2015-01-04 05:00:00  46 2.0 2.0  93.0        4.0            1          0
##   is_weekend season
## 1          1      3
## 2          1      3
## 3          1      3
## 4          1      3
## 5          1      3
## 6          1      3
```

```
dim(bike_data)
```

```
## [1] 17414    10
```

```
#add time as a variable
bike_data = bike_data %>%
   mutate(time = as.numeric(substring(timestamp, 12, 13)))

#plot usage vs hour of day
hours = aggregate(bike_data, by=list(Time = bike_data$time), mean)
```

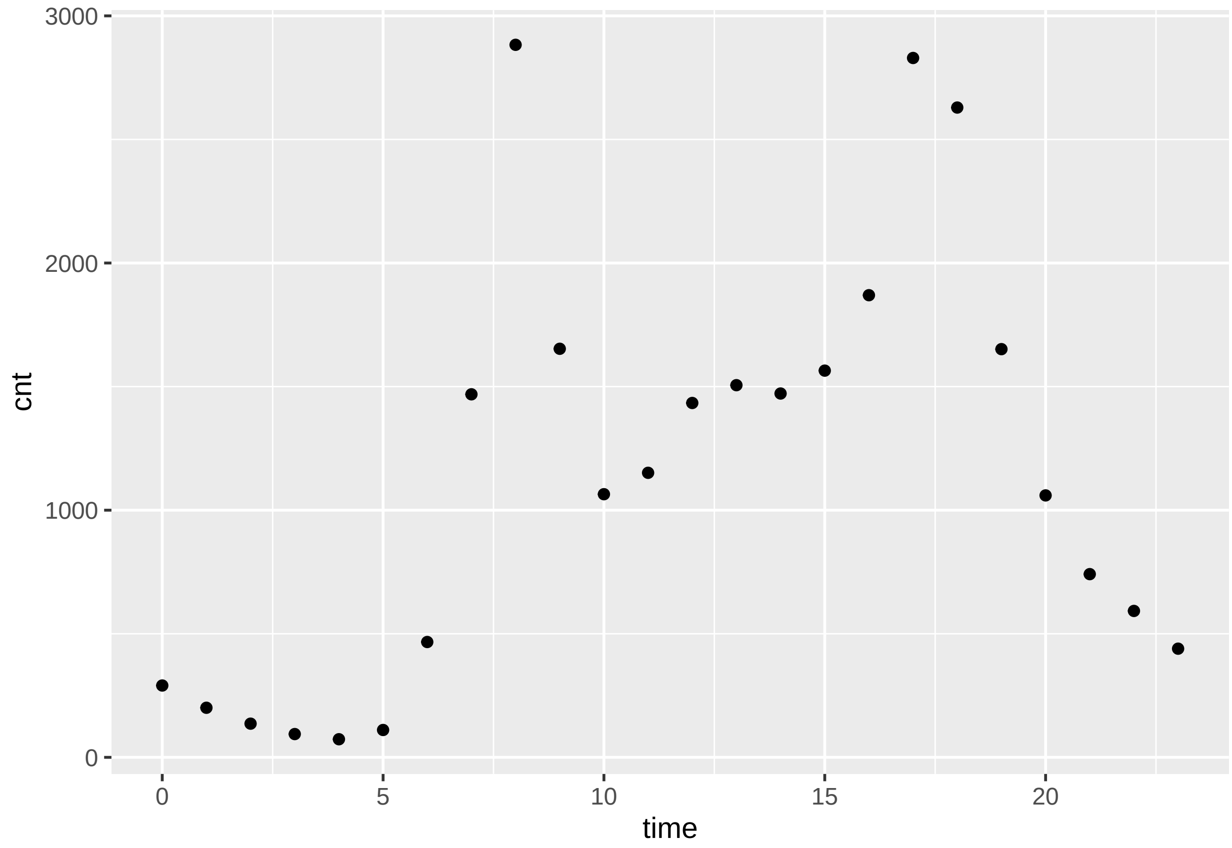
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
head(hours)
```

```
##   Time timestamp       cnt       t1        t2      hum wind_speed
## 1    0        NA 290.60912 11.23964 10.238950 78.13847   13.95407
## 2    1        NA 200.63122 10.97629  9.941759 79.30180   13.67795
## 3    2        NA 136.30374 10.74133  9.662968 80.30374   13.38558
## 4    3        NA  94.24549 10.53190  9.443135 81.07767   13.10264
## 5    4        NA  73.31345 10.35714  9.236477 81.67060   12.95978
## 6    5        NA 110.70735 10.25312  9.146325 82.10610   12.92094
##   weather_code is_holiday is_weekend   season time
## 1     2.693370 0.02209945  0.2845304 1.488950    0
## 2     2.820442 0.02209945  0.2845304 1.490331    1
## 3     3.013870 0.02219140  0.2829404 1.492372    2
## 4     2.886269 0.02219140  0.2843273 1.490985    3
## 5     2.944521 0.02219140  0.2843273 1.490985    4
## 6     2.986130 0.02219140  0.2843273 1.490985    5
```

```
ggplot(hours, aes(x=time, y=cnt)) + geom_point()
```




```r
#plot usage throughout all the days

daily_bike_data = bike_data %>% mutate(day = as.Date(timestamp, format="%Y-%m-%d")) %>%
  group_by(day) %>% # group by the day column
  summarise(day_cnt=sum(cnt))  # calculate the SUM of all cnts that occurred on each day

dim(daily_bike_data)
```

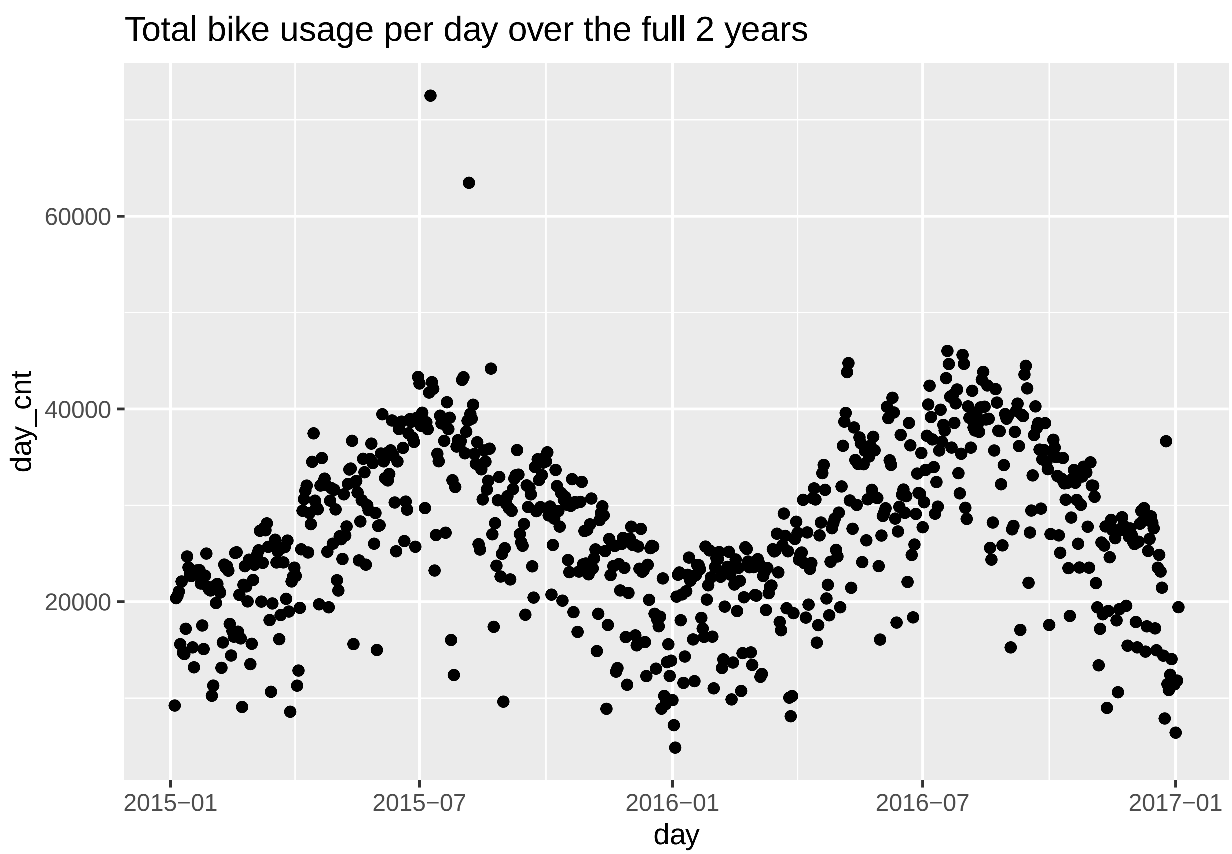
```
## [1] 730   2
```

```r
head(daily_bike_data, 50)
```

```
## # A tibble: 50 x 2
##    day        day_cnt
##    <date>       <int>
##  1 2015-01-04    9234
##  2 2015-01-05   20372
##  3 2015-01-06   20613
##  4 2015-01-07   21064
##  5 2015-01-08   15601
##  6 2015-01-09   22104
##  7 2015-01-10   14709
##  8 2015-01-11   14575
##  9 2015-01-12   17199
## 10 2015-01-13   24697
```

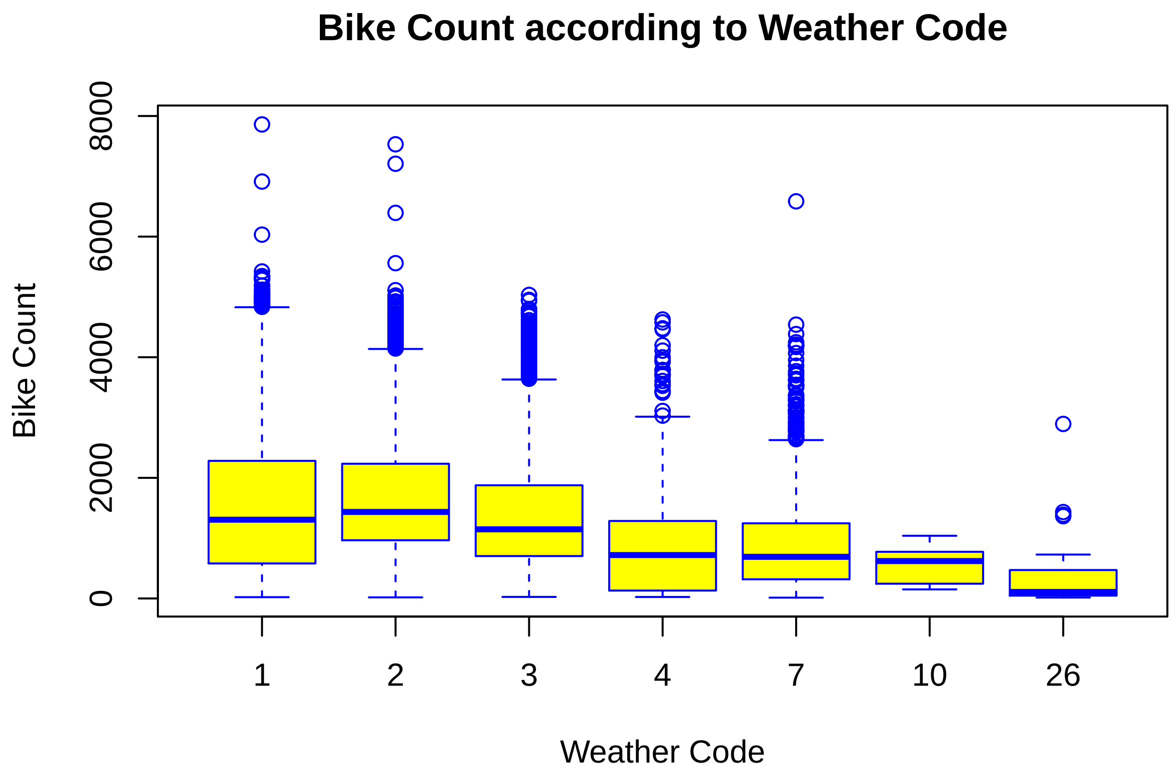
```
## # ... with 40 more rows
```

```
#plot bike usage vs day
ggplot(daily_bike_data, aes(x = day, y = day_cnt)) + geom_point() + labs(title = "Total bike usage per d
```



Usage versus Weather Code

```
no_night <- bike_data %>%
  filter(time >= 4 & time <= 21)
boxplot(cnt~weather_code,
data= no_night,
main="Bike Count according to Weather Code",
xlab="Weather Code",
ylab="Bike Count",
col="yellow",
border= "blue"
)
```



Linear Models of All Rides

```r
library(ggplot2)
library(tidyverse)
library(readr)
library(dplyr)

bike_data <- read.csv("london_merged.csv")
head(bike_data, 24)
```

```
##              timestamp cnt  t1   t2   hum wind_speed weather_code
## 1  2015-01-04 00:00:00 182 3.0  2.0  93.0        6.0            3
## 2  2015-01-04 01:00:00 138 3.0  2.5  93.0        5.0            1
## 3  2015-01-04 02:00:00 134 2.5  2.5  96.5        0.0            1
## 4  2015-01-04 03:00:00  72 2.0  2.0 100.0        0.0            1
## 5  2015-01-04 04:00:00  47 2.0  0.0  93.0        6.5            1
## 6  2015-01-04 05:00:00  46 2.0  2.0  93.0        4.0            1
## 7  2015-01-04 06:00:00  51 1.0 -1.0 100.0        7.0            4
## 8  2015-01-04 07:00:00  75 1.0 -1.0 100.0        7.0            4
## 9  2015-01-04 08:00:00 131 1.5 -1.0  96.5        8.0            4
## 10 2015-01-04 09:00:00 301 2.0 -0.5 100.0        9.0            3
## 11 2015-01-04 10:00:00 528 3.0 -0.5  93.0       12.0            3
## 12 2015-01-04 11:00:00 727 2.0 -1.5 100.0       12.0            3
```

```
## 13 2015-01-04 12:00:00  862 2.0 -1.5  96.5       13.0            4
## 14 2015-01-04 13:00:00  916 3.0 -0.5  87.0       15.0            3
## 15 2015-01-04 14:00:00 1039 2.5  0.0  90.0        8.0            3
## 16 2015-01-04 15:00:00  869 2.0 -1.5  93.0       11.0            3
## 17 2015-01-04 16:00:00  737 3.0  0.0  93.0       12.0            3
## 18 2015-01-04 17:00:00  594 3.0  0.0  93.0       11.0            3
## 19 2015-01-04 18:00:00  522 3.0  1.5  93.0        6.5            3
## 20 2015-01-04 19:00:00  379 3.0  1.0  93.0        7.0            3
## 21 2015-01-04 20:00:00  328 3.0  3.0  93.0        4.0            3
## 22 2015-01-04 21:00:00  221 3.0  2.5  93.0        5.0            4
## 23 2015-01-04 22:00:00  178 3.0  2.0  93.0        6.0            4
## 24 2015-01-04 23:00:00  157 4.0  3.5  87.0        5.0            4
##    is_holiday is_weekend season
## 1           0          1      3
## 2           0          1      3
## 3           0          1      3
## 4           0          1      3
## 5           0          1      3
## 6           0          1      3
## 7           0          1      3
## 8           0          1      3
## 9           0          1      3
## 10          0          1      3
## 11          0          1      3
## 12          0          1      3
## 13          0          1      3
## 14          0          1      3
## 15          0          1      3
## 16          0          1      3
## 17          0          1      3
## 18          0          1      3
## 19          0          1      3
## 20          0          1      3
## 21          0          1      3
## 22          0          1      3
## 23          0          1      3
## 24          0          1      3
```

```r
dim(bike_data)
```

```
## [1] 17414    10
```

```r
bike_data <- bike_data %>%
  mutate(time = as.numeric(substring(timestamp, 12, 13)))

model1 <- lm(cnt ~ hum, data = bike_data)
model2 <- lm(cnt ~ t1, data = bike_data)
model3 <- lm(cnt ~ t1 + hum + is_weekend, data = bike_data)

summary(model1)
```

```
##
## Call:
```

```
## lm(formula = cnt ~ hum, data = bike_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2037.5  -627.5  -267.5   346.8  5990.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3681.2262    37.5471   98.04   <2e-16 ***
## hum          -35.0933     0.5093  -68.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 961.9 on 17412 degrees of freedom
## Multiple R-squared:  0.2143, Adjusted R-squared:  0.2142
## F-statistic:  4748 on 1 and 17412 DF,  p-value: < 2.2e-16
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = cnt ~ t1, data = bike_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1930.4  -680.0  -227.9   426.8  6234.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   199.04      18.57   10.72   <2e-16 ***
## t1             75.72       1.36   55.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 999.8 on 17412 degrees of freedom
## Multiple R-squared:  0.1512, Adjusted R-squared:  0.1511
## F-statistic:  3101 on 1 and 17412 DF,  p-value: < 2.2e-16
```

```r
summary(model3)
```

```
##
## Call:
## lm(formula = cnt ~ t1 + hum + is_weekend, data = bike_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2011.9  -605.7  -246.8   359.7  5931.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2614.019     50.953   51.30   <2e-16 ***
## t1            44.367      1.417   31.30   <2e-16 ***
## hum          -27.178      0.552  -49.24   <2e-16 ***
```

```
## is_weekend  -204.732     15.642  -13.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 931.8 on 17410 degrees of freedom
## Multiple R-squared:  0.2627, Adjusted R-squared:  0.2626
## F-statistic:  2068 on 3 and 17410 DF,  p-value: < 2.2e-16
```

```r
mean(model1$residuals^2)
```

```
## [1] 925103.5
```

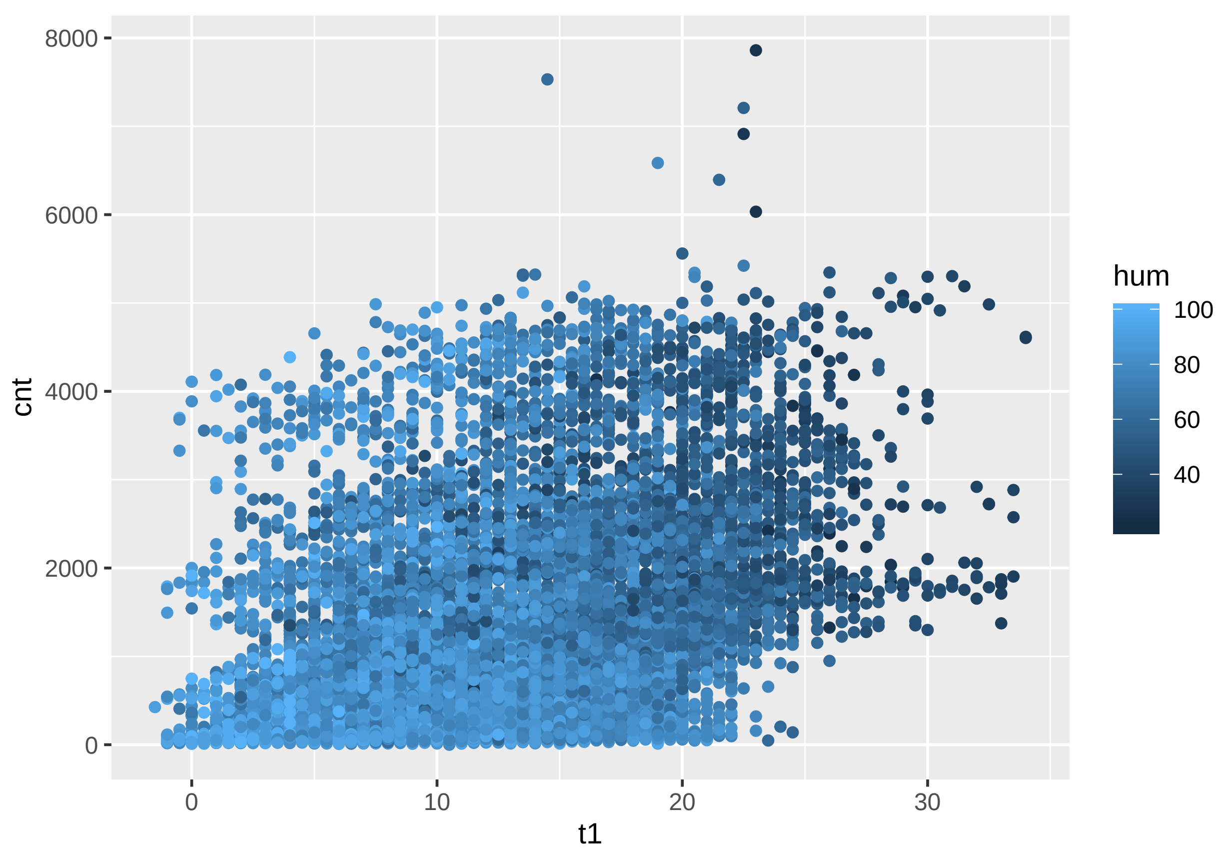
```r
mean(model2$residuals^2)
```

```
## [1] 999412.4
```

```r
mean(model3$residuals^2)
```

```
## [1] 868038.3
```

```r
ggplot(bike_data, aes(y = cnt, x = t1, color = hum)) + geom_point()
```



Linear Models of Weekend Rides

```
weekend_bikes <- bike_data %>%
  filter(is_weekend == 1)%>%
  filter(time >= 4 & time <= 21)

model1_2 <- lm(cnt ~ hum, data = weekend_bikes)
model2_2 <- lm(cnt ~ t1, data = weekend_bikes)
model3_2 <- lm(cnt ~ t1 + hum, data = weekend_bikes)

mean(model1_2$residuals^2)
```

```
## [1] 465708.8
```

```
mean(model2_2$residuals^2)
```

```
## [1] 599416.1
```

```
mean(model3_2$residuals^2)
```

```
## [1] 379425.7
```

```
summary(model1_2)
```

```
##
## Call:
## lm(formula = cnt ~ hum, data = weekend_bikes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1889.24  -478.38   -66.95   433.98  3104.28
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4559.7442    54.4588   83.73   <2e-16 ***
## hum          -47.5977     0.7489  -63.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 682.6 on 3734 degrees of freedom
## Multiple R-squared:  0.5197, Adjusted R-squared:  0.5195
## F-statistic:  4040 on 1 and 3734 DF,  p-value: < 2.2e-16
```

```
summary(model2_2)
```

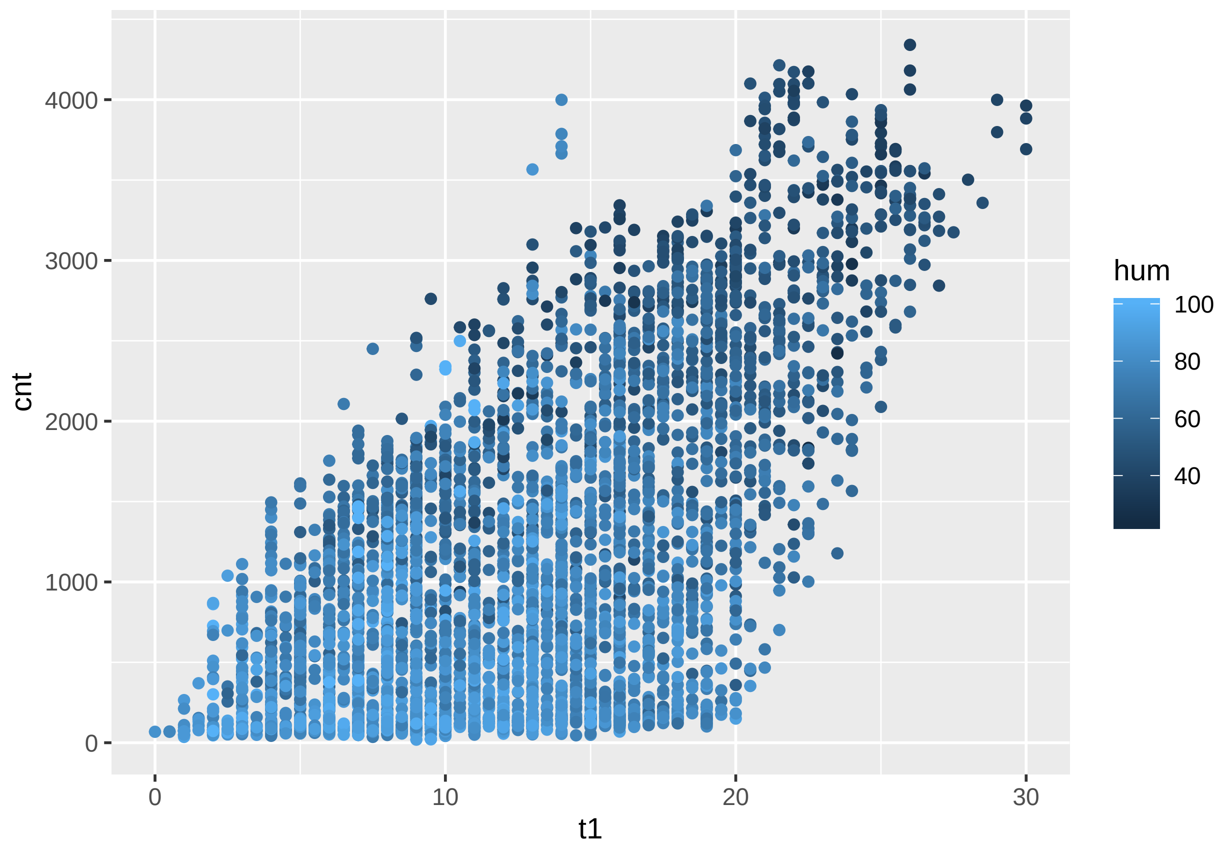
```
##
## Call:
## lm(formula = cnt ~ t1, data = weekend_bikes)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1795.02  -588.22    -9.67   597.69  2700.16 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -208.925     31.428  -6.648 3.41e-11 ***
## t1           107.697      2.243  48.016  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 774.4 on 3734 degrees of freedom
## Multiple R-squared:  0.3817, Adjusted R-squared:  0.3816 
## F-statistic:  2306 on 1 and 3734 DF,  p-value: < 2.2e-16
```

```r
summary(model3_2)
```

```
## 
## Call:
## lm(formula = cnt ~ t1 + hum, data = weekend_bikes)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1870.98  -415.64   -22.21   401.06  2967.71 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 2984.9368    73.0643   40.85   <2e-16 ***
## t1            59.9766     2.0585   29.14   <2e-16 ***
## hum          -36.2769     0.7798  -46.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 616.2 on 3733 degrees of freedom
## Multiple R-squared:  0.6086, Adjusted R-squared:  0.6084 
## F-statistic:  2903 on 2 and 3733 DF,  p-value: < 2.2e-16
```

```r
ggplot(weekend_bikes, aes(y = cnt, x = t1, color = hum)) + geom_point()
```



Temperature (how it feels)

```
model3_3 <- lm(cnt ~ t2 + hum, data = weekend_bikes)

mean(model3_3$residuals^2)
```

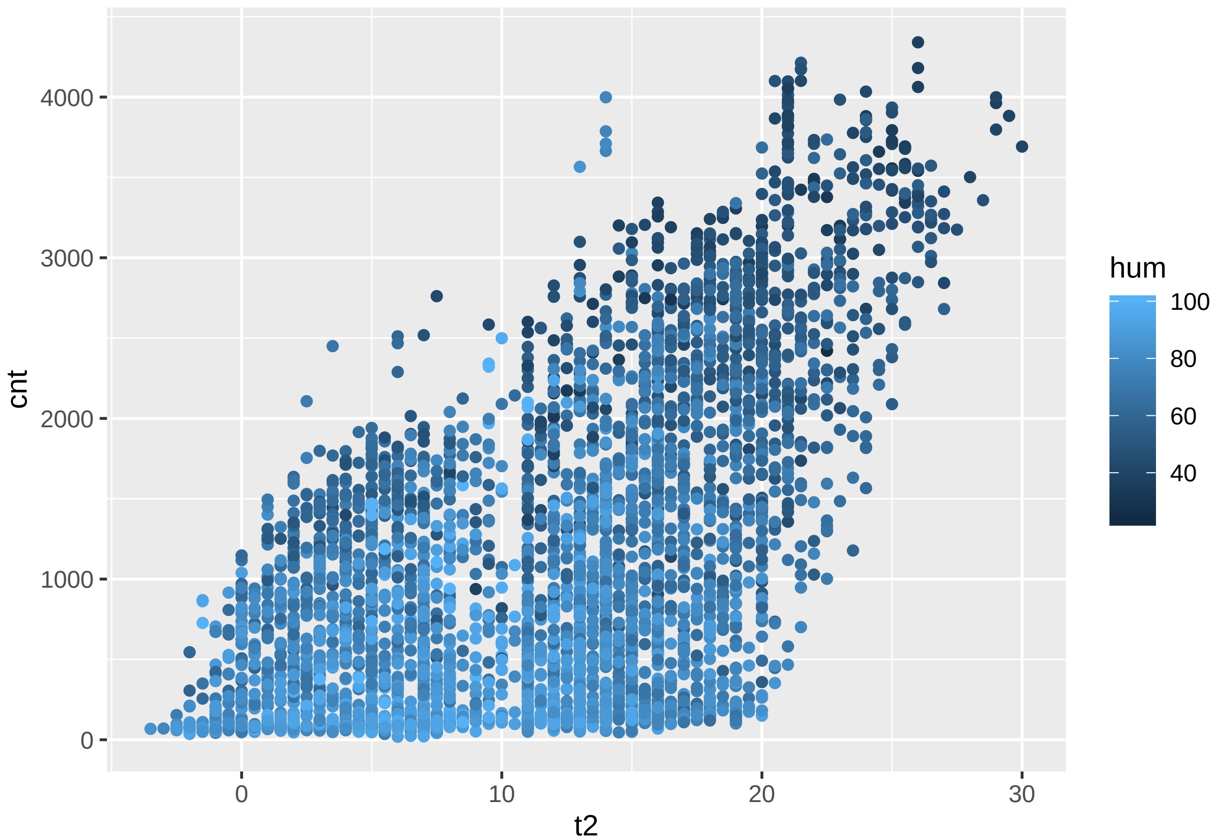
```
## [1] 383941.2
```

```
summary(model3_3)
```

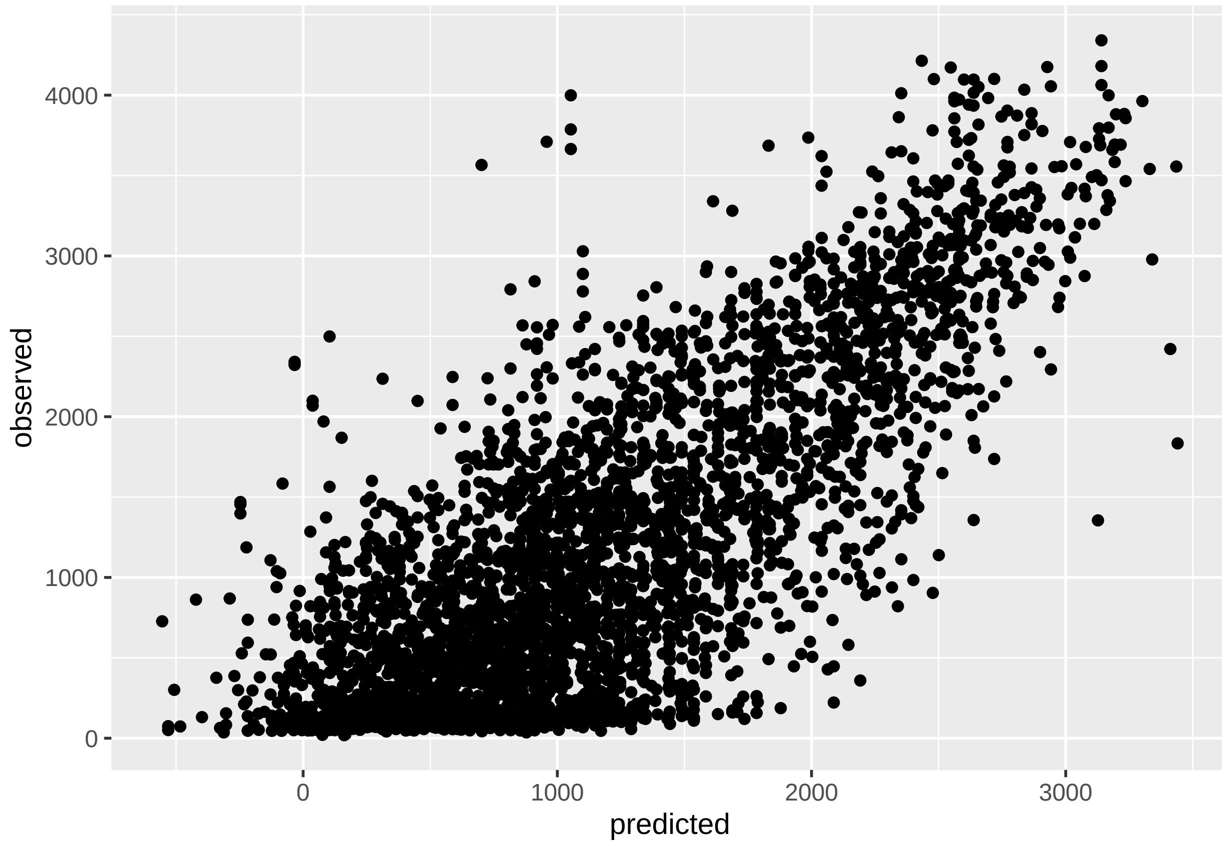
```
##
## Call:
## lm(formula = cnt ~ t2 + hum, data = weekend_bikes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1865.42  -424.37   -16.82   402.71  2945.94
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3314.4227    66.3053   49.99   <2e-16 ***
## t2            47.3597     1.6797   28.20   <2e-16 ***
## hum          -37.9792     0.7608  -49.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 619.9 on 3733 degrees of freedom
## Multiple R-squared:  0.604,  Adjusted R-squared:  0.6038
## F-statistic:  2847 on 2 and 3733 DF,  p-value: < 2.2e-16
```

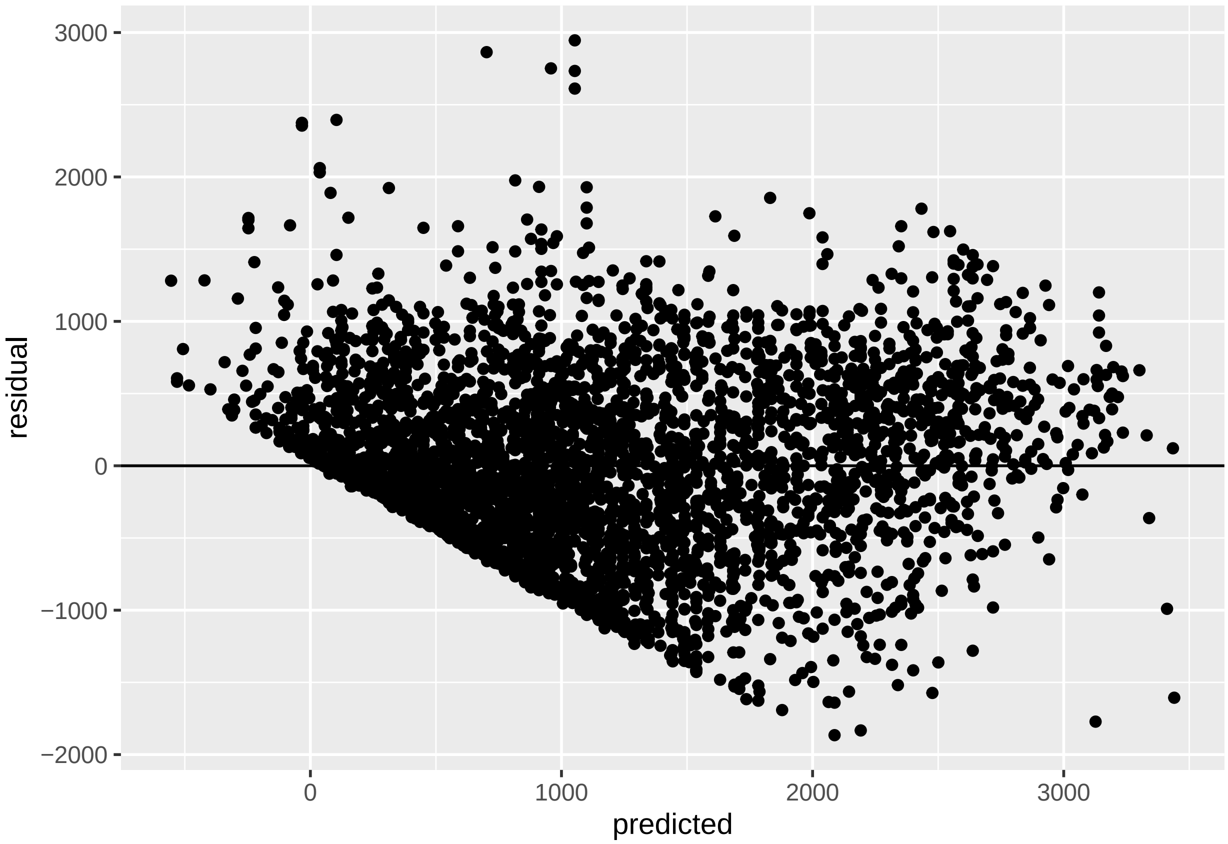
```
ggplot(weekend_bikes, aes(y = cnt, x = t2, color = hum)) + geom_point()
```



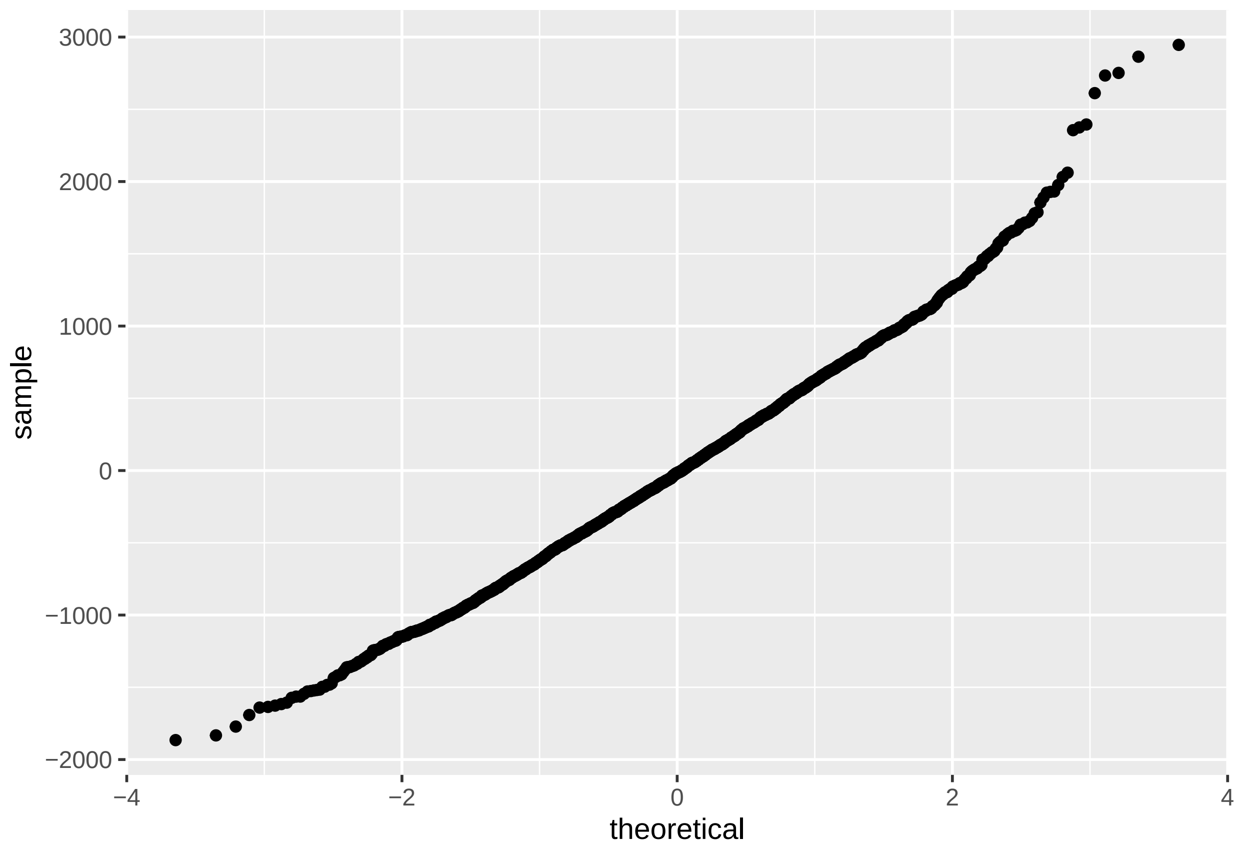

```
modt2_results <- data.frame(observed = weekend_bikes$cnt, predicted = model3_3$fitted.values, residual =

ggplot(modt2_results, aes(y = observed, x = predicted))+ geom_point()
```




```r
ggplot(modt2_results, aes(y = residual, x = predicted))+ geom_point() + geom_hline(yintercept = 0)
```

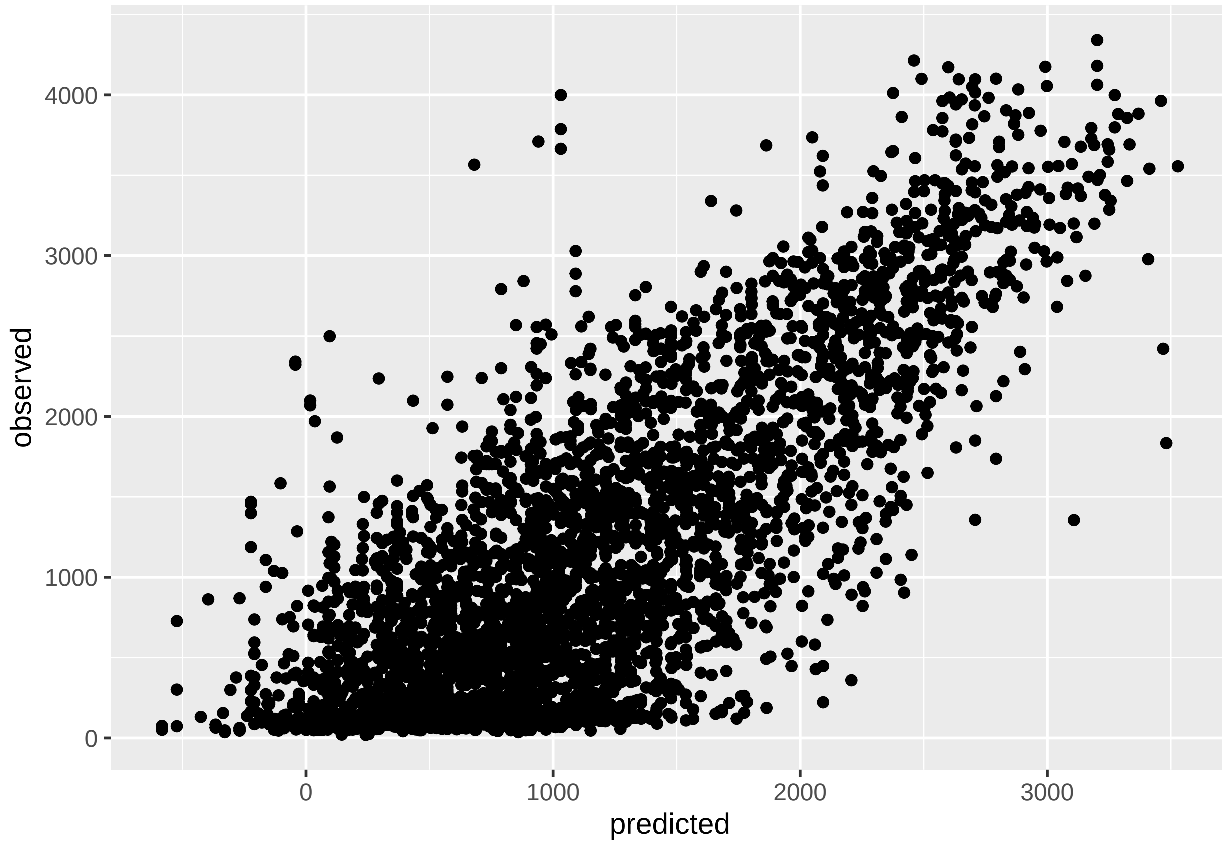



```
ggplot(modt2_results, aes(sample = residual)) +
    geom_qq()
```

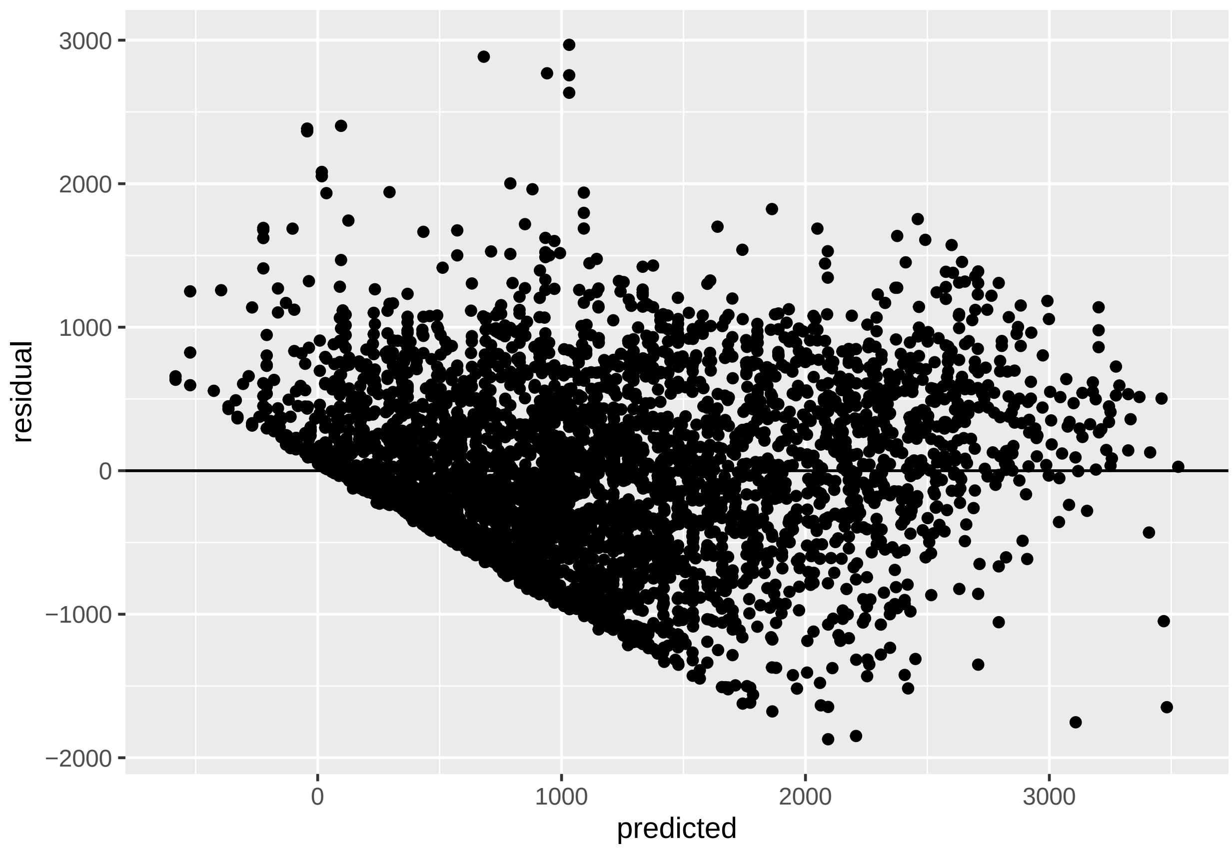


Model Assumptions for Weekend Rides

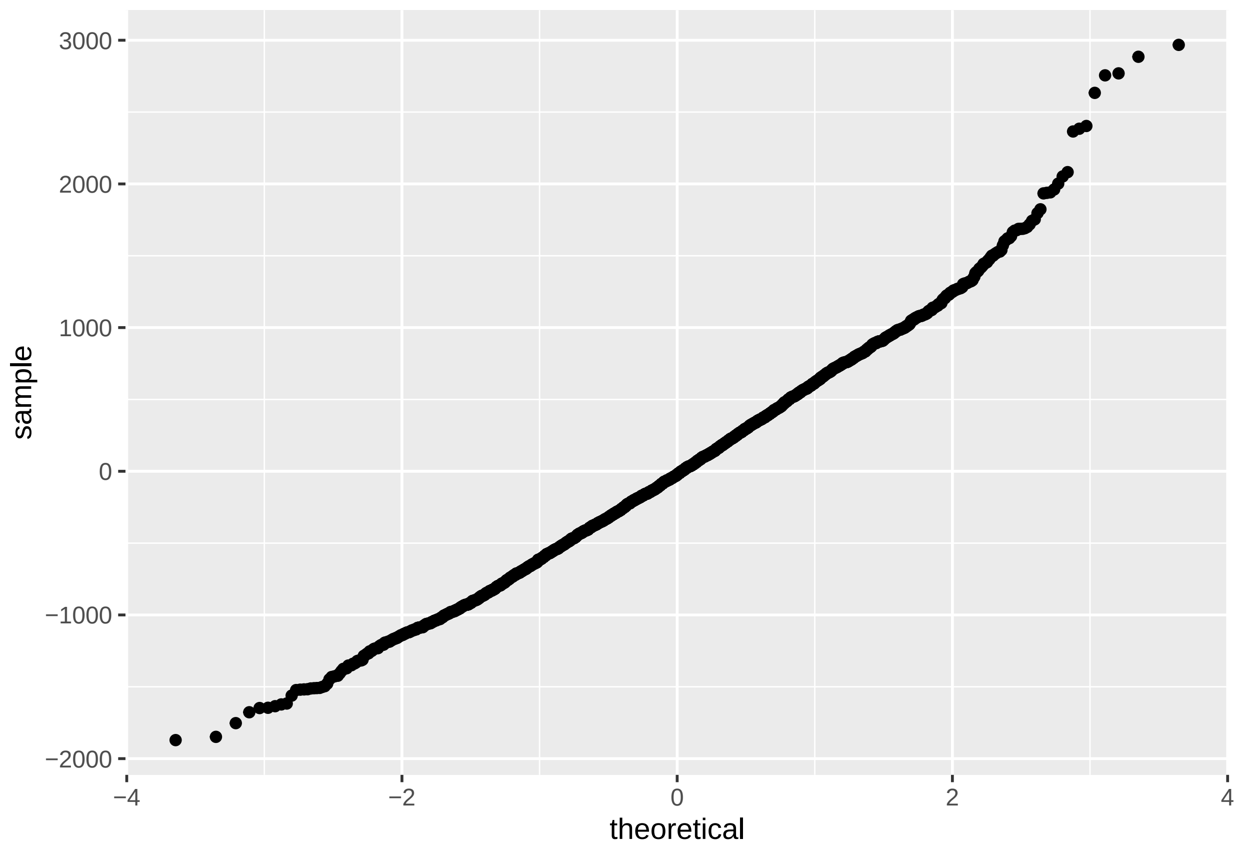
```
mod_results <- data.frame(observed = weekend_bikes$cnt, predicted = model3_2$fitted.values, residual = r
ggplot(mod_results, aes(y = observed, x = predicted))+ geom_point()
```




```
ggplot(mod_results, aes(y = residual, x = predicted))+ geom_point() + geom_hline(yintercept = 0)
```

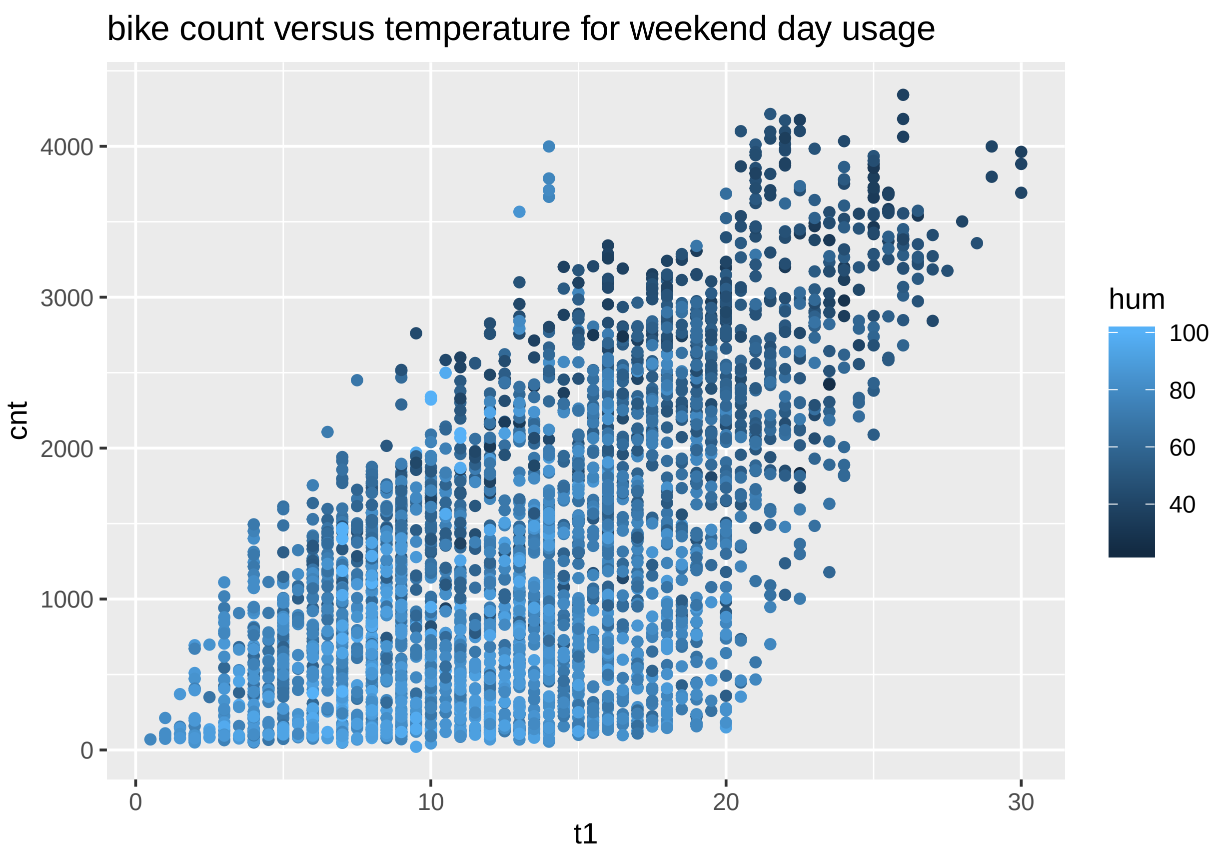



```r
ggplot(mod_results, aes(sample = residual)) +
    geom_qq()
```

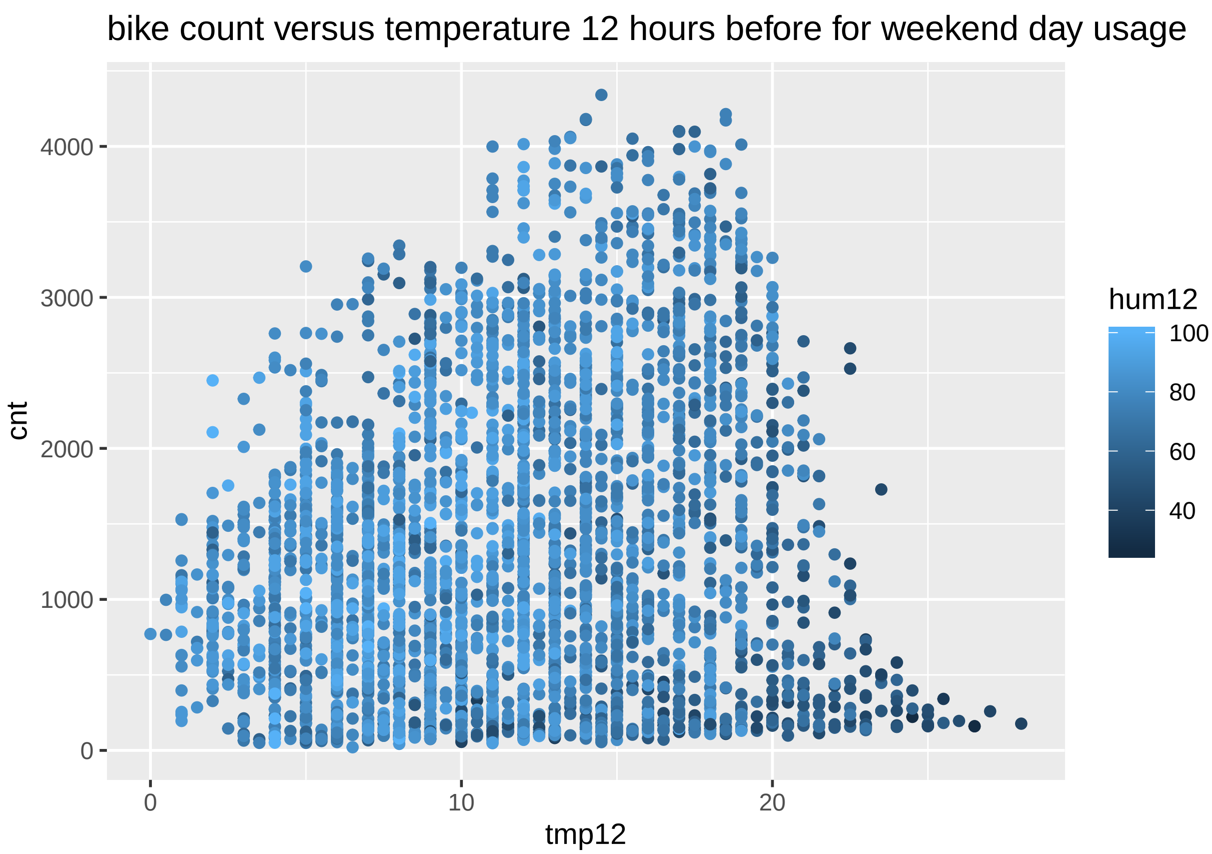


Determining whether current weather or previous weather (12 hours or 24 hours before) is a better predictor of bike usage

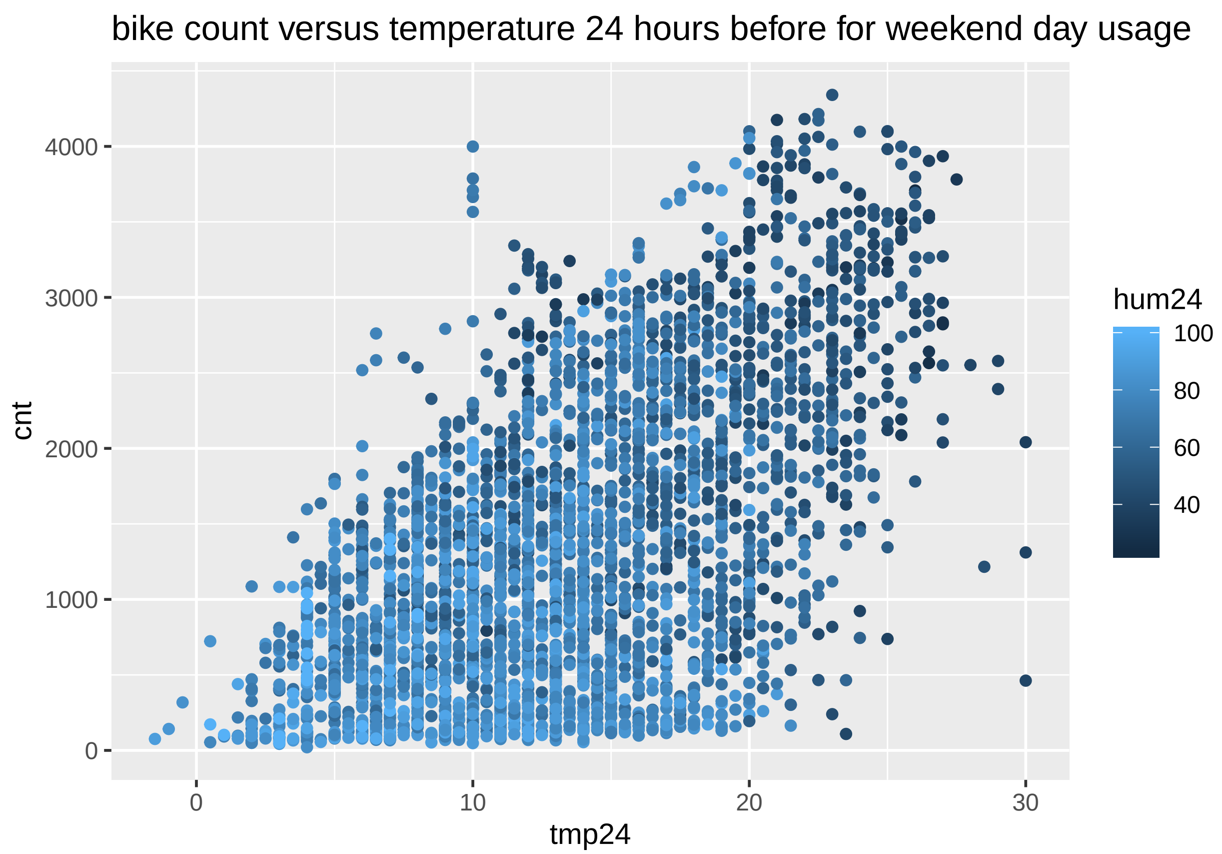
```
#create variables representing temperature 12 and 24 hours before
lagged_bike_data = bike_data %>%
mutate(tmp12 =lag(t1, n = 12)) %>%
mutate(tmp24 =lag(t1, n = 24)) %>%
mutate(hum12 =lag(hum, n = 12)) %>%
mutate(hum24 =lag(hum, n = 24))

#remove rows that don't have data due to being lagged
lagged_bike_data <- lagged_bike_data[-1:-24,]

#filter out times between 8 pm and 6 am and weekdays
dayUsage = lagged_bike_data %>% filter(time >= 6 & time <= 20)
weekendDayUsage <- dayUsage %>% filter(is_weekend == 1)

#after filtering for only weekend
ggplot(weekendDayUsage, aes(y = cnt, x = t1, color = hum)) + geom_point()+ labs(title = "bike count vers
```




```r
ggplot(weekendDayUsage, aes(y = cnt, x = tmp12, color = hum12)) + geom_point()+ labs(title = "bike coun
```




```
ggplot(weekendDayUsage, aes(y = cnt, x = tmp24, color = hum24)) + geom_point()+ labs(title = "bike coun
```




```r
#build models for weekend
weekendUsageT1 <- lm(cnt ~ t1 + hum, data = weekendDayUsage)
weekendUsage12 <- lm(cnt ~ tmp12 + hum12 , data = weekendDayUsage)
weekendUsage24 <- lm(cnt ~ tmp24 + hum24 , data = weekendDayUsage)

summary(weekendUsageT1) #adjusted R-squared = .4823
```

```
##
## Call:
## lm(formula = cnt ~ t1 + hum, data = weekendDayUsage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1967.42  -402.42    -3.46   396.49  2856.11
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2878.6691    76.7801   37.49   <2e-16 ***
## t1            64.9948     2.1971   29.58   <2e-16 ***
## hum          -34.3598     0.8238  -41.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 600.7 on 3100 degrees of freedom
## Multiple R-squared:  0.6226, Adjusted R-squared:  0.6224
## F-statistic:  2557 on 2 and 3100 DF,  p-value: < 2.2e-16
```

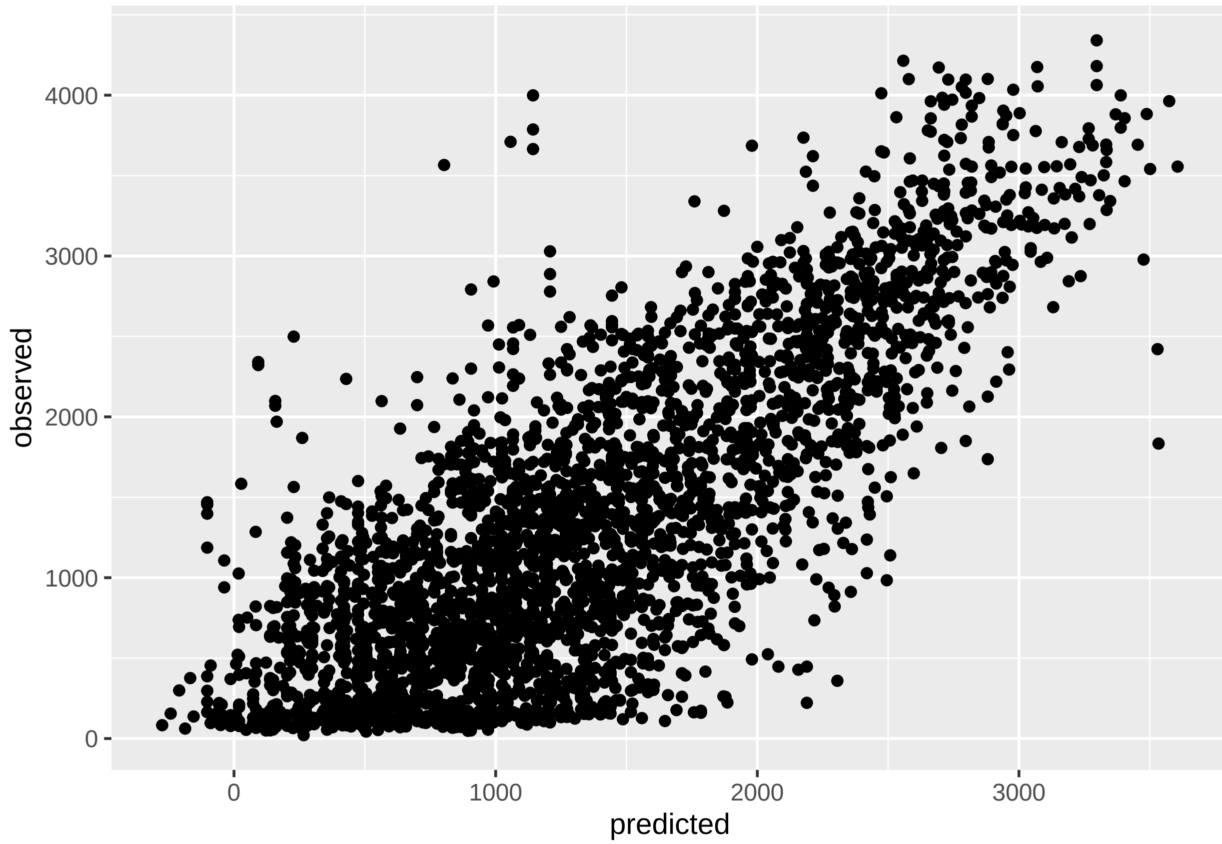
```r
summary(weekendUsage12) #adjusted R-squared = .09851
```

```
##
## Call:
## lm(formula = cnt ~ tmp12 + hum12, data = weekendDayUsage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1868.7  -729.6  -132.2   621.7  2975.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -913.263    141.778  -6.441 1.37e-10 ***
## tmp12         54.816      3.503  15.649  < 2e-16 ***
## hum12         21.203      1.557  13.622  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 932.3 on 3100 degrees of freedom
## Multiple R-squared:  0.09084,    Adjusted R-squared:  0.09025
## F-statistic: 154.9 on 2 and 3100 DF,  p-value: < 2.2e-16
```

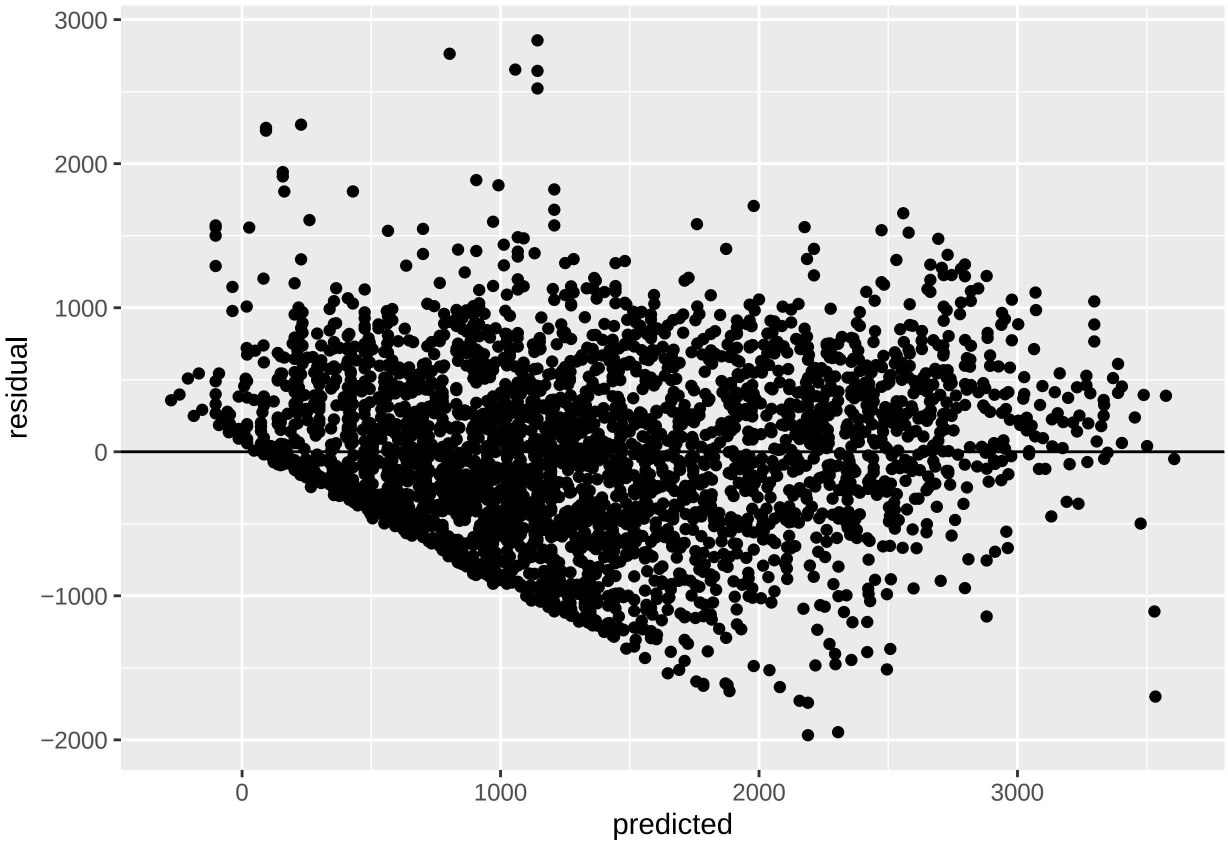
```r
summary(weekendUsage24) #adjusted R-squared = .3198
```

```
##
## Call:
## lm(formula = cnt ~ tmp24 + hum24, data = weekendDayUsage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2717.55  -532.11   -50.94   505.85  2922.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1940.708     95.526   20.32   <2e-16 ***
## tmp24         70.817      2.699   26.24   <2e-16 ***
## hum24        -22.142      1.034  -21.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 744.5 on 3100 degrees of freedom
## Multiple R-squared:  0.4203, Adjusted R-squared:  0.4199
## F-statistic:  1124 on 2 and 3100 DF,  p-value: < 2.2e-16
```

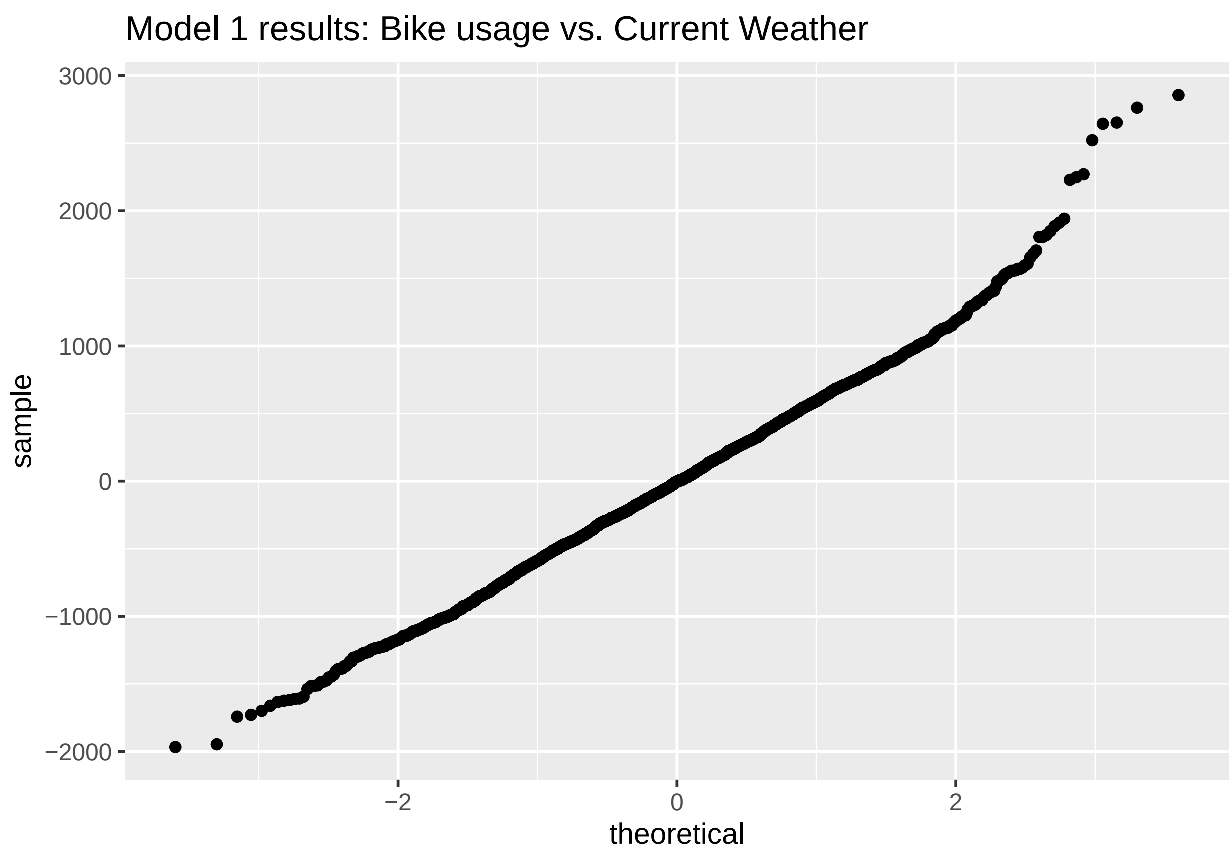
```r
mod_results1 <- data.frame(observed = weekendDayUsage$cnt, predicted = weekendUsageT1$fitted.values, res

#residuals for model 1
ggplot(mod_results1, aes(y = observed, x = predicted))+ geom_point()
```



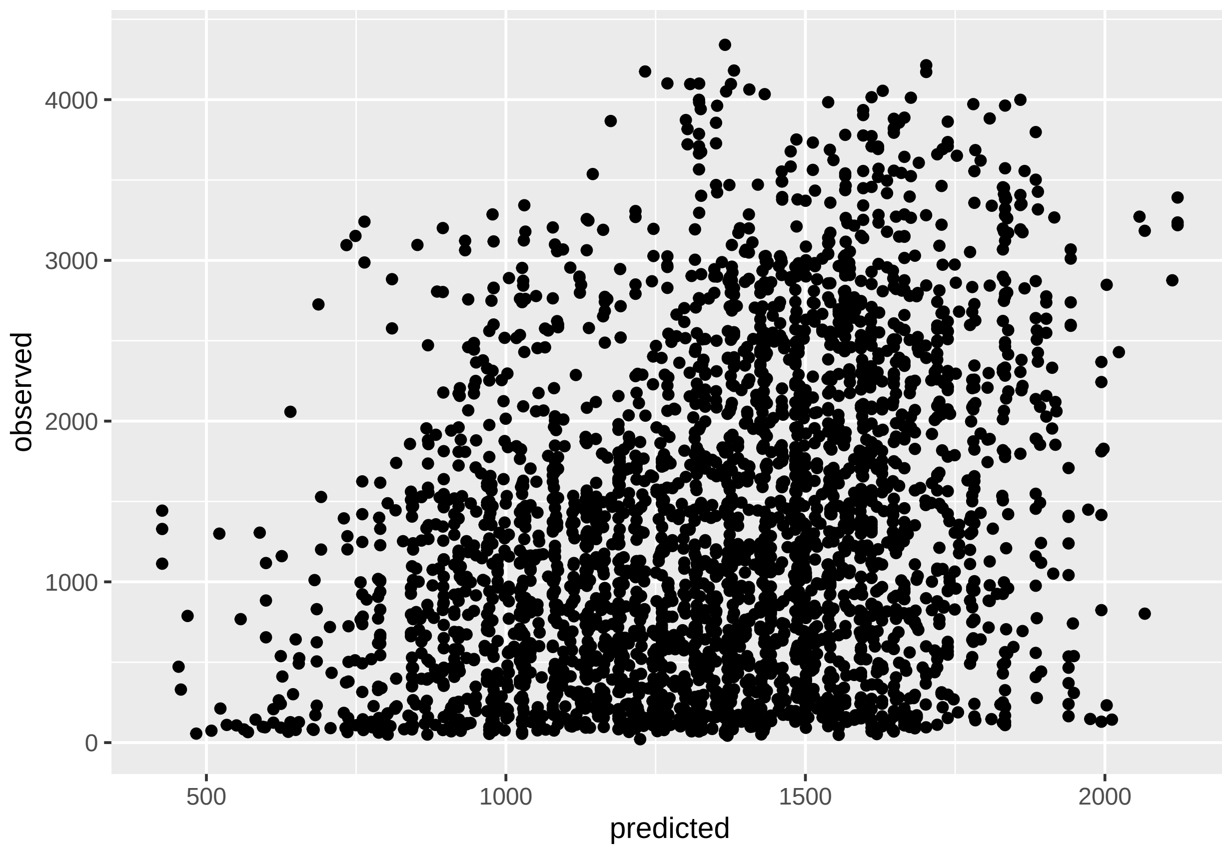

```r
ggplot(mod_results1, aes(y = residual, x = predicted))+ geom_point() + geom_hline(yintercept = 0)
```



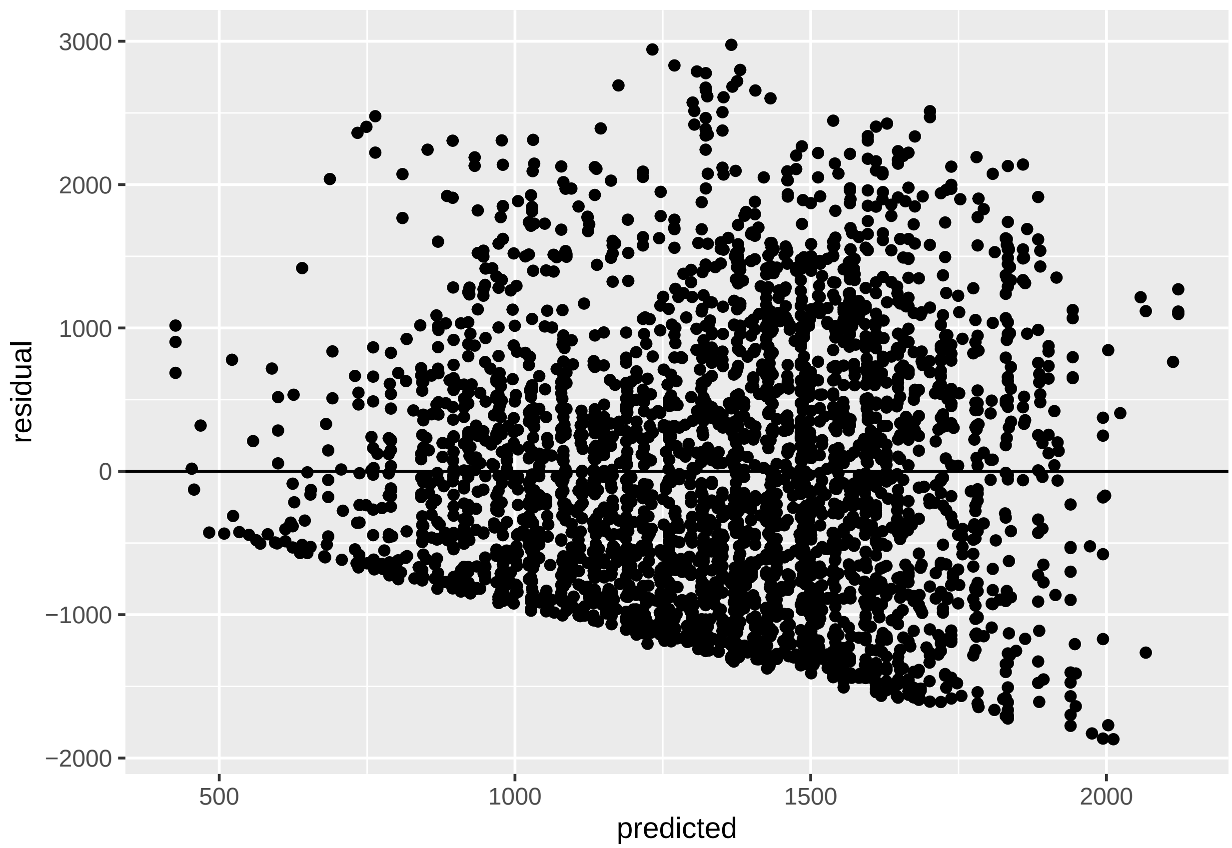

```
ggplot(mod_results1, aes(sample = residual)) +
    geom_qq() + labs(title = "Model 1 results: Bike usage vs. Current Weather")
```



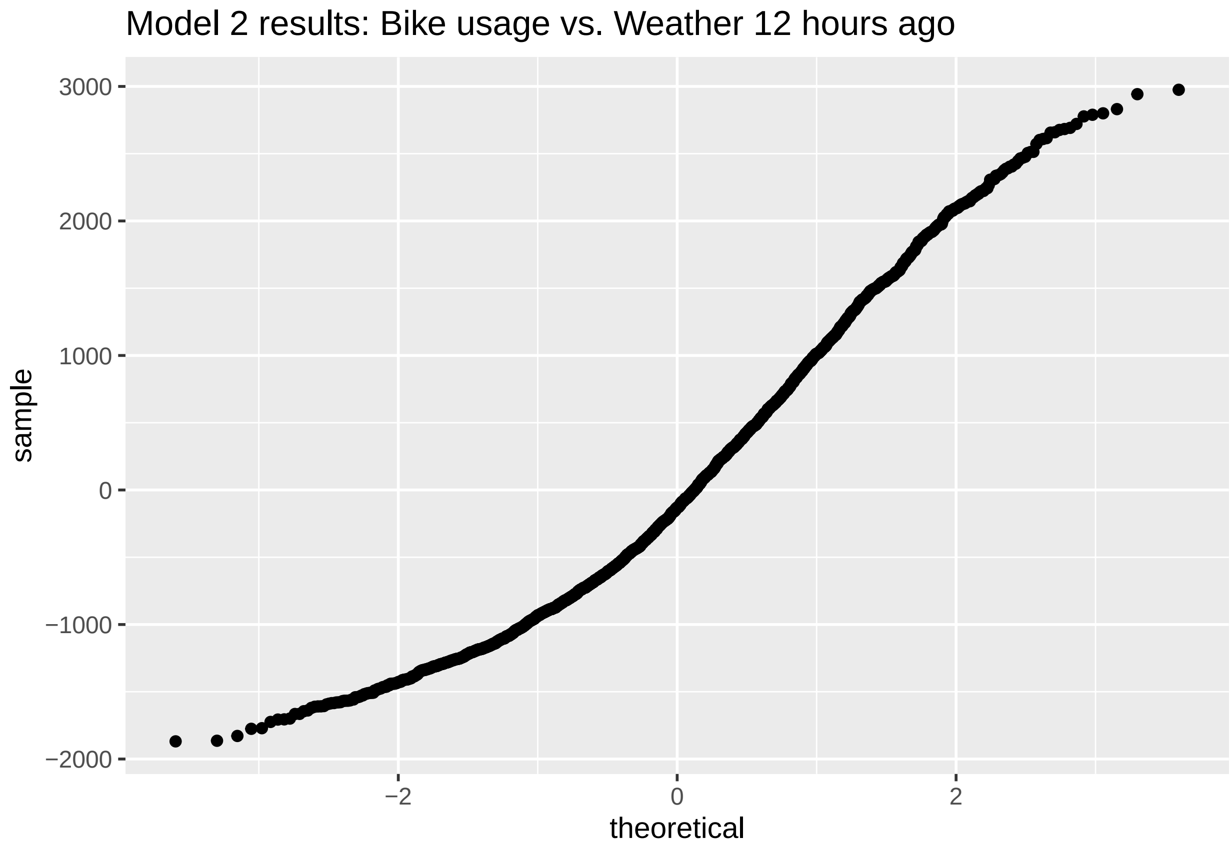

```
mod_results2 <- data.frame(observed = weekendDayUsage$cnt, predicted = weekendUsage12$fitted.values, res

#residuals for model 2
ggplot(mod_results2, aes(y = observed, x = predicted))+ geom_point()
```



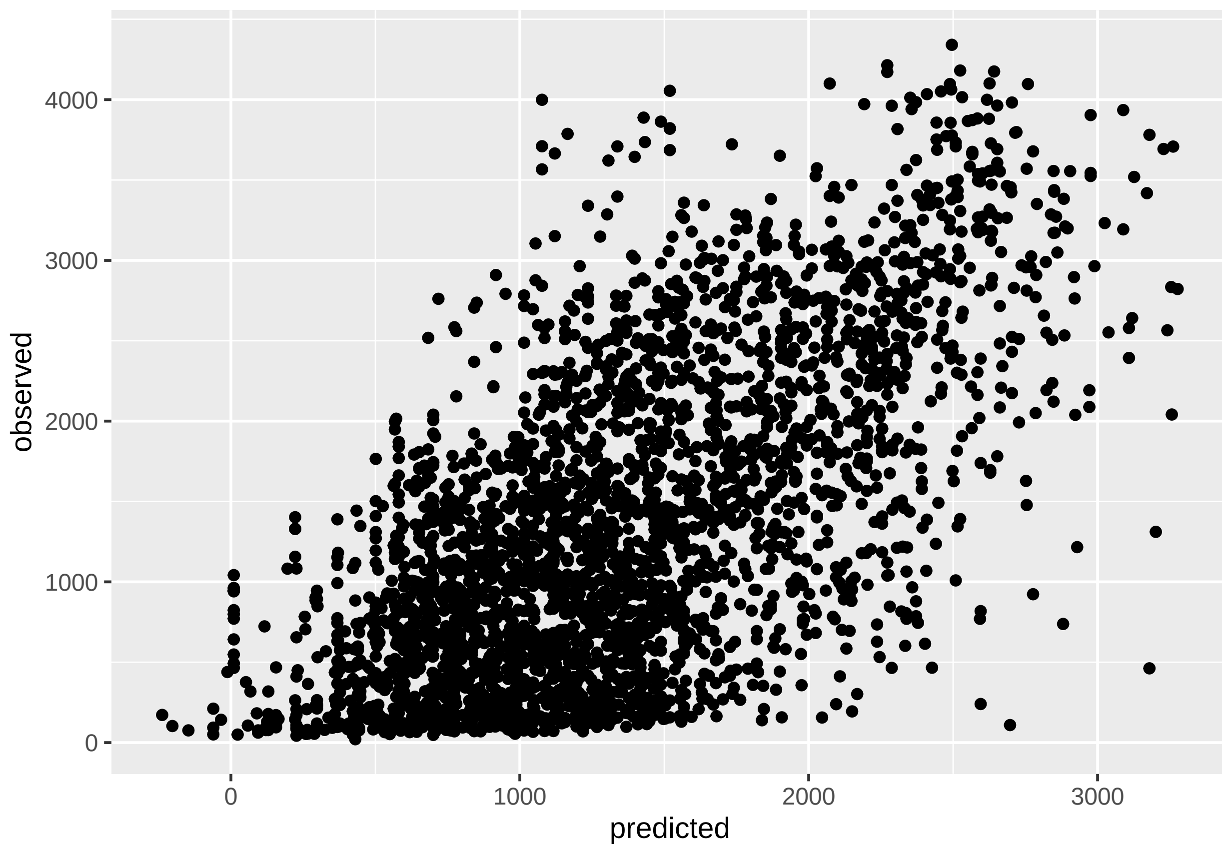

```
ggplot(mod_results2, aes(y = residual, x = predicted))+ geom_point() + geom_hline(yintercept = 0)
```



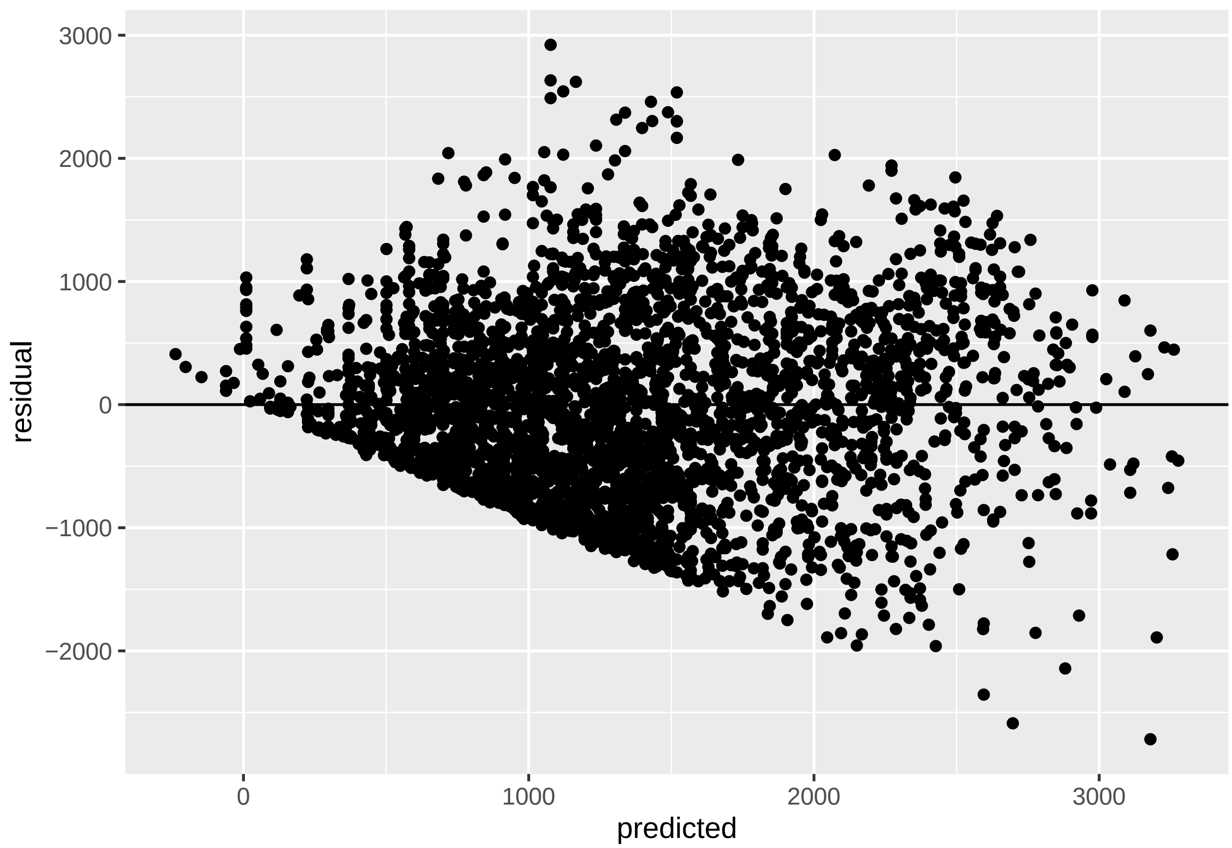

```r
ggplot(mod_results2, aes(sample = residual)) +
    geom_qq()+ labs(title = "Model 2 results: Bike usage vs. Weather 12 hours ago")
```



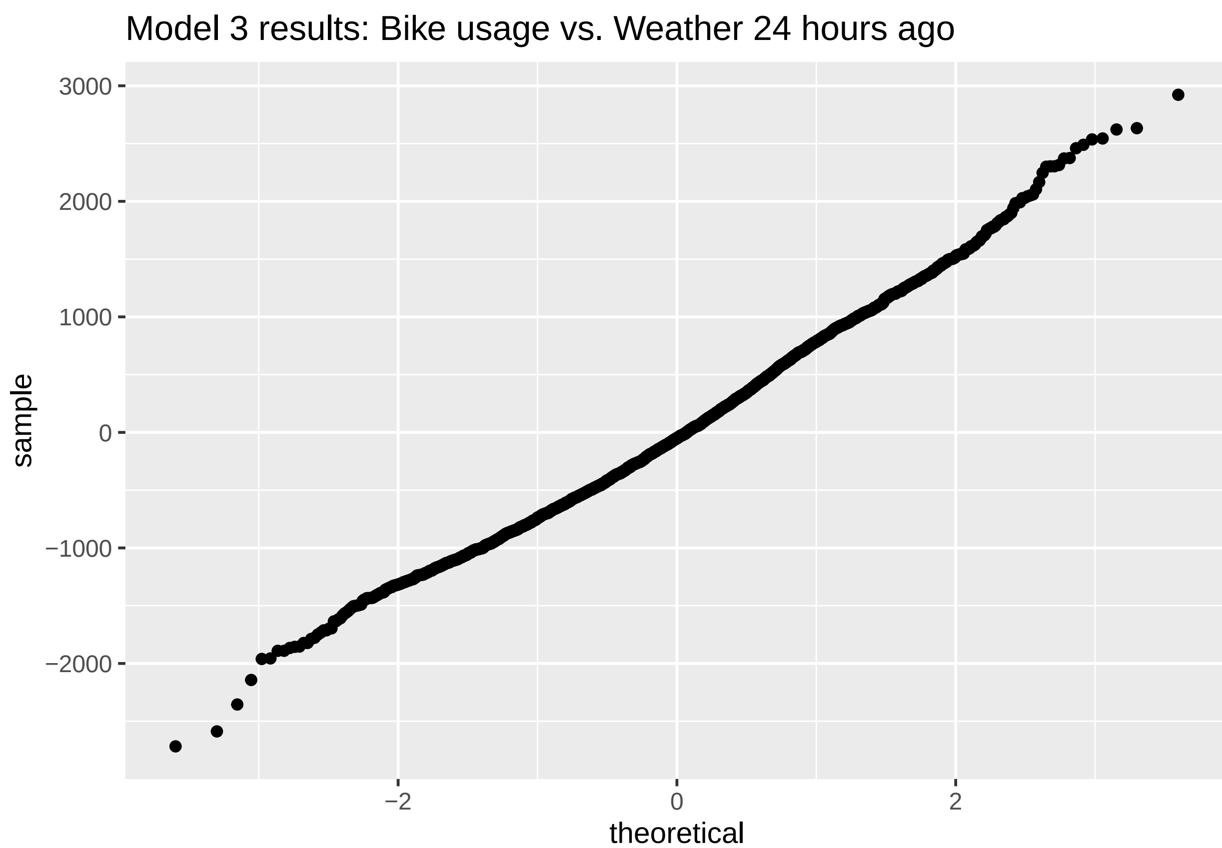

```
mod_results3 <- data.frame(observed = weekendDayUsage$cnt, predicted = weekendUsage24$fitted.values, res

#residuals for model 3
ggplot(mod_results3, aes(y = observed, x = predicted))+ geom_point()
```




```
ggplot(mod_results3, aes(y = residual, x = predicted))+ geom_point() + geom_hline(yintercept = 0)
```

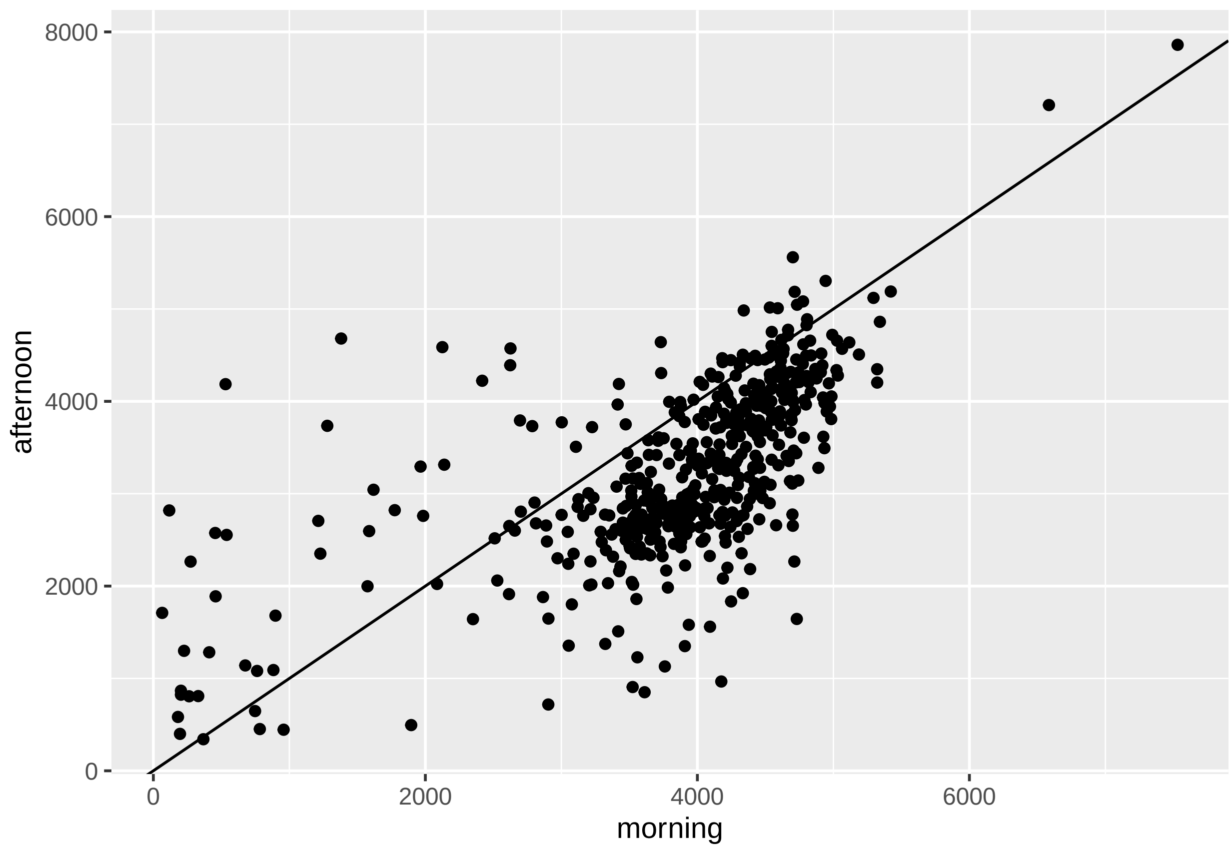



```r
ggplot(mod_results3, aes(sample = residual)) +
    geom_qq()+ labs(title = "Model 3 results: Bike usage vs. Weather 24 hours ago")
```



Correlation of Morning Communte with Afternoon Commute

```
weekdays <- bike_data %>%
  filter(is_weekend == 0)

morning_commutes <- weekdays %>%
  filter(grepl("08:00:00", timestamp))

afternoon_commutes <- weekdays %>%
  filter(grepl("17:00:00", timestamp))
afternoon_commutes <- afternoon_commutes[-c(519),] #one extra value

commutes <- data.frame(morning = morning_commutes$cnt, afternoon = afternoon_commutes$cnt)
ggplot(commutes, aes(y= afternoon, x = morning)) + geom_point() + geom_abline()
```




```r
model_commutes <- lm(afternoon ~ morning , data = commutes)
summary(model_commutes)
```

```
##
## Call:
## lm(formula = afternoon ~ morning, data = commutes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2460.29  -488.08   -30.31   440.55  3027.35
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 828.16312  129.37548   6.401 3.47e-10 ***
## morning       0.62240    0.03239  19.214  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 745.3 on 516 degrees of freedom
## Multiple R-squared:  0.4171, Adjusted R-squared:  0.4159
## F-statistic: 369.2 on 1 and 516 DF,  p-value: < 2.2e-16
```

```r
#Adjusted R-squared: 0.4159
```