Load Libraries and Data

In [1]:
```
suppressMessages(library(tidyverse))
suppressMessages(library(lubridate))
suppressMessages(library(lattice))
suppressMessages(library(caret))
```

In [2]:
```
#List the Kaggle files.

list.files(path = "../input/ashrae-energy-prediction")
```

'building_metadata.csv' 'sample_submission.csv' 'test.csv' 'train.csv'
'weather_test.csv' 'weather_train.csv'

In [4]:
```
train <- read_csv("../input/ashrae-energy-prediction/train.csv")
dim(train)
```

```
Parsed with column specification:
cols(
  building_id = col_double(),
  meter = col_double(),
  timestamp = col_datetime(format = ""),
  meter_reading = col_double()
)
```

20216100 4

In [5]:
```
building_metadata <- read_csv("../input/ashrae-energy-prediction/building_metadata.csv")
dim(building_metadata)
```

```
Parsed with column specification:
cols(
  site_id = col_double(),
  building_id = col_double(),
  primary_use = col_character(),
  square_feet = col_double(),
  year_built = col_double(),
  floor_count = col_double()
)
```

1449 6

In [6]:
```
weather_train <- read_csv("../input/ashrae-energy-prediction/weather_train.csv")
dim(weather_train)
```

```
Parsed with column specification:
cols(
  site_id = col_double(),
  timestamp = col_datetime(format = ""),
  air_temperature = col_double(),
  cloud_coverage = col_double(),
  dew_temperature = col_double(),
  precip_depth_1_hr = col_double(),
  sea_level_pressure = col_double(),
  wind_direction = col_double(),
  wind_speed = col_double()
)


139773  9
```

Join the Data Sets

In [7]:
```
train_building <- left_join(train, building_metadata)
```

```
Joining, by = "building_id"
```

In [8]:
```
train_building_weather <- left_join(train_building, weather_train)
```

```
Joining, by = c("timestamp", "site_id")
```

In [9]:
```
glimpse(train_building_weather)
summary(train_building_weather)
```

```
Observations: 20,216,100
Variables: 16
$ building_id        <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
13, 14, …
$ meter              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ timestamp          <dttm> 2016-01-01, 2016-01-01, 2016-01-01, 2016
-01-01, 2…
$ meter_reading      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ site_id            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ primary_use        <chr> "Education", "Education", "Education", "E
ducation"…
$ square_feet        <dbl> 7432, 2720, 5376, 23685, 116607, 8000, 27
926, 1210…
$ year_built         <dbl> 2008, 2004, 1991, 2002, 1975, 2000, 1981,
1989, 20…
$ floor_count        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
A, NA, NA…
$ air_temperature    <dbl> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 2
5, 25, 25…
$ cloud_coverage     <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6,…
$ dew_temperature    <dbl> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 2
0, 20, 20…
$ precip_depth_1_hr  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
A, NA, NA…
$ sea_level_pressure <dbl> 1019.7, 1019.7, 1019.7, 1019.7, 1019.7, 1
019.7, 10…
$ wind_direction     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ wind_speed         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
```

```
  building_id         meter          timestamp
 Min.   :   0.0   Min.   :0.0000   Min.   :2016-01-01 00:00:00
 1st Qu.: 393.0   1st Qu.:0.0000   1st Qu.:2016-04-05 21:00:00
 Median : 895.0   Median :0.0000   Median :2016-07-04 17:00:00
 Mean   : 799.3   Mean   :0.6624   Mean   :2016-07-03 22:59:40
 3rd Qu.:1179.0   3rd Qu.:1.0000   3rd Qu.:2016-10-02 22:00:00
 Max.   :1448.0   Max.   :3.0000   Max.   :2016-12-31 23:00:00

 meter_reading         site_id       primary_use         square_feet
 Min.   :       0   Min.   : 0.000   Length:20216100    Min.   :   2
83
 1st Qu.:      18   1st Qu.: 3.000   Class :character   1st Qu.: 325
27
 Median :      79   Median : 9.000   Mode  :character   Median : 727
09
 Mean   :    2117   Mean   : 7.992                      Mean   :1077
83
 3rd Qu.:     268   3rd Qu.:13.000                      3rd Qu.:1391
13
 Max.   :21904700   Max.   :15.000                      Max.   :8750
00

   year_built        floor_count       air_temperature  cloud_covera
ge
 Min.   :1900       Min.   : 1         Min.   :-28.90   Min.   :0
 1st Qu.:1951       1st Qu.: 1         1st Qu.:  8.60   1st Qu.:0
 Median :1969       Median : 3         Median : 16.70   Median :0
 Mean   :1968       Mean   : 4         Mean   : 15.99   Mean   :2
 3rd Qu.:1993       3rd Qu.: 6         3rd Qu.: 24.10   3rd Qu.:4
 Max.   :2017       Max.   :26         Max.   : 47.20   Max.   :9
 NA's   :12127645   NA's   :16709167   NA's   :96658    NA's   :8825
365
 dew_temperature  precip_depth_1_hr sea_level_pressure wind_directio
n
 Min.   :-35.00   Min.   : -1       Min.   : 968.2     Min.   :  0
 1st Qu.:  0.00   1st Qu.:  0       1st Qu.:1011.6     1st Qu.: 70
 Median :  8.90   Median :  0       Median :1016.0     Median :180
 Mean   :  7.75   Mean   :  1       Mean   :1016.1     Mean   :173
 3rd Qu.: 16.10   3rd Qu.:  0       3rd Qu.:1020.5     3rd Qu.:280
 Max.   : 26.10   Max.   :343       Max.   :1045.5     Max.   :360
 NA's   :100140   NA's   :3749023   NA's   :1231669    NA's   :14490
48
   wind_speed
 Min.   : 0.00
 1st Qu.: 2.10
 Median : 3.10
 Mean   : 3.38
 3rd Qu.: 4.60
 Max.   :19.00
 NA's   :143676
```

Clean the Data

In [10]:
```
# Remove floor_count variable because of excessive NA's.

subset <- train_building_weather %>% select(-floor_count)
```

In [11]:
```
# Count buildings by meter type.

subset %>% group_by(meter, building_id) %>% count() %>% group_by(meter
) %>% count()
```

A grouped_df: 4 × 2

meter	n
<dbl>	<int>
0	1413
1	498
2	324
3	145

In [13]:
```
# Remove observations with NA's.

subset <- na.omit(subset)
```

In [14]:
```
# Remove outlier observations for meter type 2.

subset <- subset %>% filter(building_id != 1099 | meter != 2)
```

In [15]:
```
# Assess remaining observations.

glimpse(subset)
summary(subset)
```

```
Observations: 2,879,660
Variables: 15
$ building_id        <dbl> 565, 566, 569, 570, 571, 572, 573, 574, 5
75, 576, …
$ meter              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ timestamp          <dttm> 2016-01-01 01:00:00, 2016-01-01 01:00:00
, 2016-01…
$ meter_reading      <dbl> 8.5000, 0.5210, 243.5000, 79.4880, 16.750
0, 304.95…
$ site_id            <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4,…
$ primary_use        <chr> "Education", "Education", "Education", "E
ducation"…
$ square_feet        <dbl> 15326, 2010, 86091, 193202, 47954, 94175,
23815, 5…
$ year_built         <dbl> 1954, 1957, 1964, 1964, 1980, 1964, 1914,
1905, 19…
$ air_temperature    <dbl> 9.4, 9.4, 9.4, 9.4, 9.4, 9.4, 9.4, 9.4, 9
.4, 9.4, …
$ cloud_coverage     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ dew_temperature    <dbl> -2.2, -2.2, -2.2, -2.2, -2.2, -2.2, -2.2,
-2.2, -2…
$ precip_depth_1_hr  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ sea_level_pressure <dbl> 1021.4, 1021.4, 1021.4, 1021.4, 1021.4, 1
021.4, 10…
$ wind_direction     <dbl> 360, 360, 360, 360, 360, 360, 360, 360, 3
60, 360, …
$ wind_speed         <dbl> 3.1, 3.1, 3.1, 3.1, 3.1, 3.1, 3.1, 3.1, 3
.1, 3.1, …
```

```
  building_id         meter          timestamp
 Min.   :   0.0   Min.   :0.0000   Min.   :2016-01-01 01:00:00
 1st Qu.: 171.0   1st Qu.:0.0000   1st Qu.:2016-04-05 13:00:00
 Median : 235.0   Median :0.0000   Median :2016-07-07 02:00:00
 Mean   : 279.6   Mean   :0.5159   Mean   :2016-07-04 19:09:17
 3rd Qu.: 395.0   3rd Qu.:1.0000   3rd Qu.:2016-10-03 05:00:00
 Max.   :1448.0   Max.   :3.0000   Max.   :2016-12-31 23:00:00
 meter_reading         site_id       primary_use         square_feet
 Min.   :    0.00   Min.   : 0.000   Length:2879660     Min.   :   2
83
 1st Qu.:   19.79   1st Qu.: 2.000   Class :character   1st Qu.: 337
39
 Median :   82.06   Median : 2.000   Mode  :character   Median : 721
02
 Mean   :  265.78   Mean   : 2.088                      Mean   :1101
55
 3rd Qu.:  234.29   3rd Qu.: 3.000                      3rd Qu.:1414
61
 Max.   :22658.40   Max.   :15.000                      Max.   :8503
54
   year_built   air_temperature  cloud_coverage  dew_temperature
 Min.   :1900   Min.   :-17.80   Min.   :0.000   Min.   :-22.800
 1st Qu.:1956   1st Qu.: 14.40   1st Qu.:0.000   1st Qu.:  0.000
 Median :1974   Median : 21.70   Median :2.000   Median :  7.800
 Mean   :1973   Mean   : 21.11   Mean   :2.586   Mean   :  7.433
 3rd Qu.:2002   3rd Qu.: 27.80   3rd Qu.:4.000   3rd Qu.: 14.400
 Max.   :2017   Max.   : 47.20   Max.   :9.000   Max.   : 25.600
 precip_depth_1_hr  sea_level_pressure wind_direction    wind_speed
 Min.   : -1.0000   Min.   : 991.9     Min.   :  0.0   Min.   : 0.00
0
 1st Qu.:  0.0000   1st Qu.:1010.7     1st Qu.: 80.0   1st Qu.: 2.10
0
 Median :  0.0000   Median :1015.3     Median :170.0   Median : 3.10
0
 Mean   :  0.1824   Mean   :1015.3     Mean   :169.5   Mean   : 3.30
2
 3rd Qu.:  0.0000   3rd Qu.:1019.5     3rd Qu.:270.0   3rd Qu.: 4.60
0
 Max.   :221.0000   Max.   :1041.0     Max.   :360.0   Max.   :16.00
0
```

In [16]:
```
# Verify that remaining observations include some of every meter type
in approximately the original proportions.

subset %>% group_by(meter, building_id) %>% count() %>% group_by(meter
) %>% count()
```

A grouped_df: 4 × 2

meter	n
<dbl>	<int>
0	524
1	157
2	66
3	55

In [17]:
```
# Remove site_id identifier variable, which is merely a foreign key. (
Building_id is similar, but is retained for plotting purposes.)

subset <- subset %>% select(-site_id)
```

In [18]:
```
# Create factors from timestamp variable. Keep timestamp for plotting
purposes.

subset <- subset %>%
    mutate(
        week_of_year = week(timestamp),
        day_of_week = wday(timestamp),
        hour_of_day = hour(timestamp))
```

In [19]:
```
# Convert doubles to integers where the values are integer in nature.
(Factor type is not used because it interferes with
# the PCA model building, below.)

subset$building_id  <- as.integer(subset$building_id)
subset$meter        <- as.integer(subset$meter)
subset$week_of_year <- as.integer(subset$week_of_year)
subset$day_of_week  <- as.integer(subset$day_of_week)
subset$hour_of_day  <- as.integer(subset$hour_of_day)
```

In [20]:
```
# Glimpse and summarize.

glimpse(subset)
summary(subset)
```

```
Observations: 2,879,660
Variables: 17
$ building_id        <int> 565, 566, 569, 570, 571, 572, 573, 574, 5
75, 576, …
$ meter              <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ timestamp          <dttm> 2016-01-01 01:00:00, 2016-01-01 01:00:00
, 2016-01…
$ meter_reading      <dbl> 8.5000, 0.5210, 243.5000, 79.4880, 16.750
0, 304.95…
$ primary_use        <chr> "Education", "Education", "Education", "E
ducation"…
$ square_feet        <dbl> 15326, 2010, 86091, 193202, 47954, 94175,
23815, 5…
$ year_built         <dbl> 1954, 1957, 1964, 1964, 1980, 1964, 1914,
1905, 19…
$ air_temperature    <dbl> 9.4, 9.4, 9.4, 9.4, 9.4, 9.4, 9.4, 9.4, 9
.4, 9.4, …
$ cloud_coverage     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ dew_temperature    <dbl> -2.2, -2.2, -2.2, -2.2, -2.2, -2.2, -2.2,
-2.2, -2…
$ precip_depth_1_hr  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,…
$ sea_level_pressure <dbl> 1021.4, 1021.4, 1021.4, 1021.4, 1021.4, 1
021.4, 10…
$ wind_direction     <dbl> 360, 360, 360, 360, 360, 360, 360, 360, 3
60, 360, …
$ wind_speed         <dbl> 3.1, 3.1, 3.1, 3.1, 3.1, 3.1, 3.1, 3.1, 3
.1, 3.1, …
$ week_of_year       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1,…
$ day_of_week        <int> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6,…
$ hour_of_day        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1,…
```

```
  building_id         meter          timestamp
 Min.   :   0.0   Min.   :0.0000   Min.   :2016-01-01 01:00:00
 1st Qu.: 171.0   1st Qu.:0.0000   1st Qu.:2016-04-05 13:00:00
 Median : 235.0   Median :0.0000   Median :2016-07-07 02:00:00
 Mean   : 279.6   Mean   :0.5159   Mean   :2016-07-04 19:09:17
 3rd Qu.: 395.0   3rd Qu.:1.0000   3rd Qu.:2016-10-03 05:00:00
 Max.   :1448.0   Max.   :3.0000   Max.   :2016-12-31 23:00:00
 meter_reading      primary_use         square_feet       year_built
 Min.   :    0.00   Length:2879660     Min.   :   283   Min.   :1900
 1st Qu.:   19.79   Class :character   1st Qu.: 33739   1st Qu.:1956
 Median :   82.06   Mode  :character   Median : 72102   Median :1974
 Mean   :  265.78                      Mean   :110155   Mean   :1973
 3rd Qu.:  234.29                      3rd Qu.:141461   3rd Qu.:2002
 Max.   :22658.40                      Max.   :850354   Max.   :2017
 air_temperature  cloud_coverage  dew_temperature   precip_depth_1_h
r
 Min.   :-17.80   Min.   :0.000   Min.   :-22.800   Min.   : -1.0000
 1st Qu.: 14.40   1st Qu.:0.000   1st Qu.:  0.000   1st Qu.:  0.0000
 Median : 21.70   Median :2.000   Median :  7.800   Median :  0.0000
 Mean   : 21.11   Mean   :2.586   Mean   :  7.433   Mean   :  0.1824
 3rd Qu.: 27.80   3rd Qu.:4.000   3rd Qu.: 14.400   3rd Qu.:  0.0000
 Max.   : 47.20   Max.   :9.000   Max.   : 25.600   Max.   :221.0000
 sea_level_pressure wind_direction    wind_speed      week_of_year
 Min.   : 991.9     Min.   :  0.0   Min.   : 0.000   Min.   : 1.00
 1st Qu.:1010.7     1st Qu.: 80.0   1st Qu.: 2.100   1st Qu.:14.00
 Median :1015.3     Median :170.0   Median : 3.100   Median :27.00
 Mean   :1015.3     Mean   :169.5   Mean   : 3.302   Mean   :27.06
 3rd Qu.:1019.5     3rd Qu.:270.0   3rd Qu.: 4.600   3rd Qu.:40.00
 Max.   :1041.0     Max.   :360.0   Max.   :16.000   Max.   :53.00
  day_of_week     hour_of_day
 Min.   :1.000   Min.   : 0.00
 1st Qu.:2.000   1st Qu.: 5.00
 Median :4.000   Median :11.00
 Mean   :4.007   Mean   :10.84
 3rd Qu.:6.000   3rd Qu.:17.00
 Max.   :7.000   Max.   :23.00
```

In [21]:
```
# Save a copy of the cleaned data set.

write_csv(subset, "/kaggle/working/subset.csv")
```

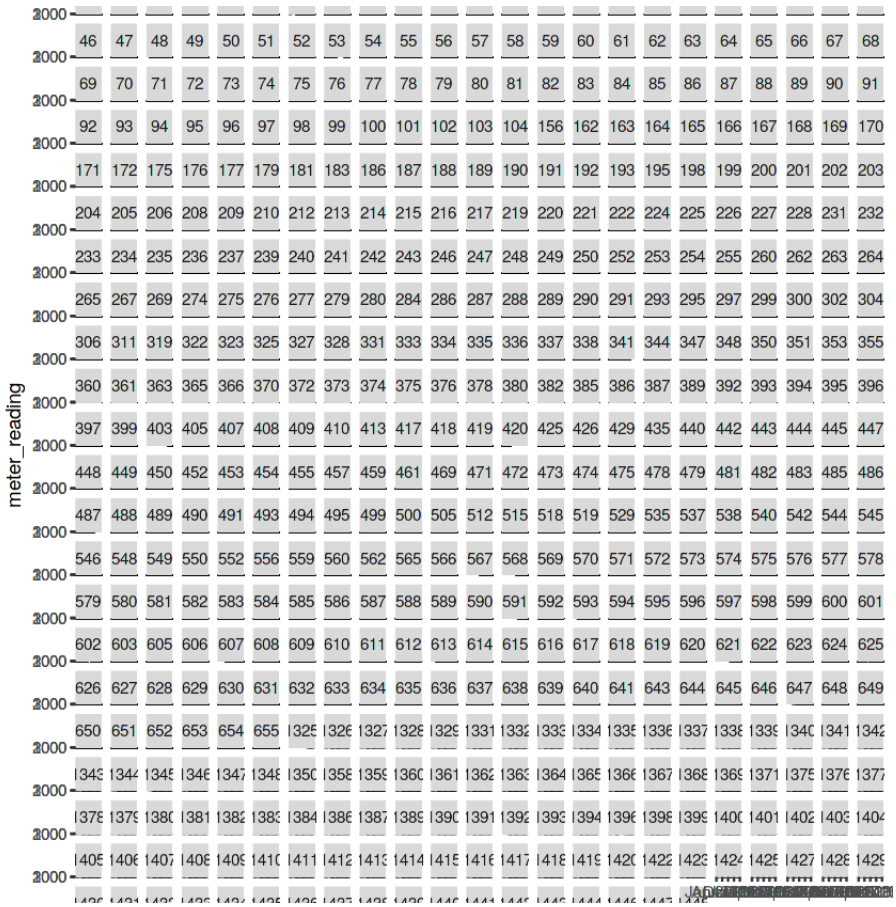
In [ ]:
```
# This read_csv can remain commented-out unlessed desired upon restart
ing the kernel to short-circuit the preceding
# data preparation steps.

# subset <- read_csv("/kaggle/working/subset.csv")
```

Create Subsets by Meter Type for Modeling

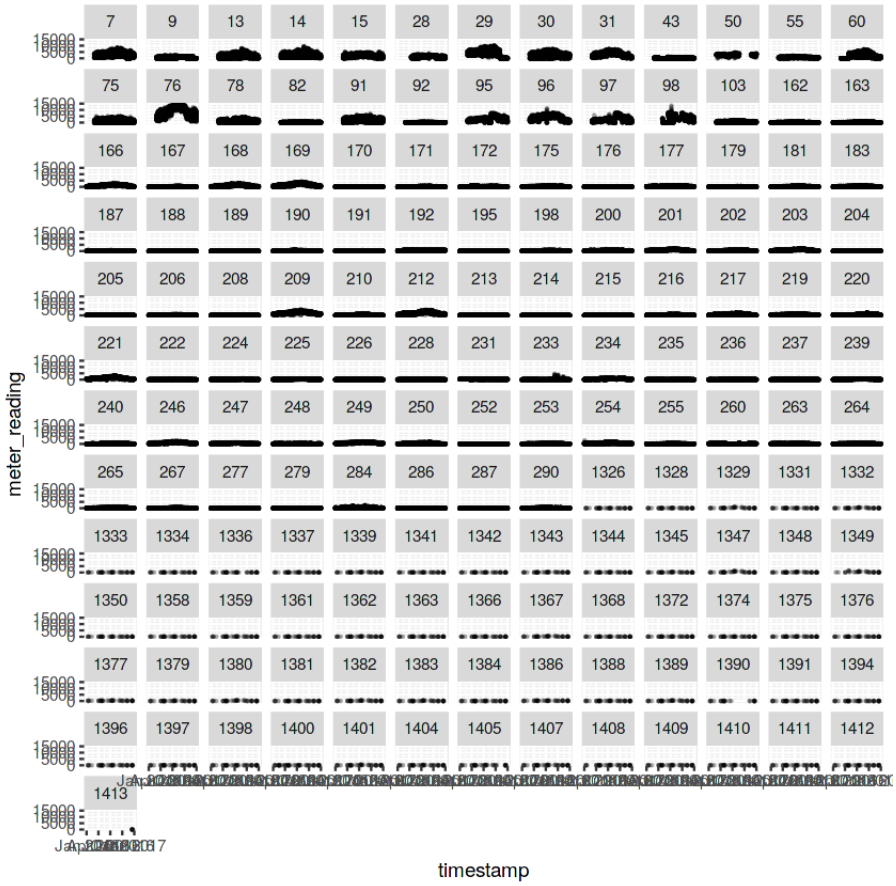
In [22]:
```
# Create subsets for each meter type.

subset_0 <- subset %>% filter(meter == 0)
subset_1 <- subset %>% filter(meter == 1)
subset_2 <- subset %>% filter(meter == 2)
subset_3 <- subset %>% filter(meter == 3)
```

Plot meter_reading vs. timestamp for each building.

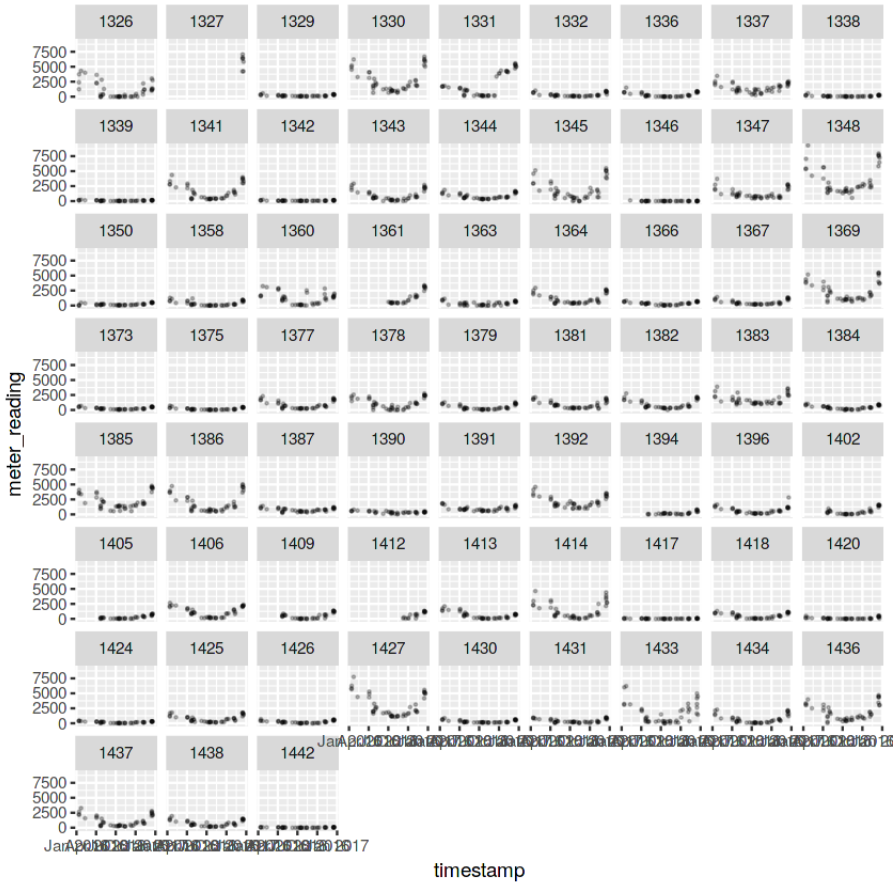
In [23]:
```
ggplot(subset_0, aes(timestamp, meter_reading, group = building_id)) +
  geom_point(size = .25, alpha = .25) +
  facet_wrap(vars(building_id))
```



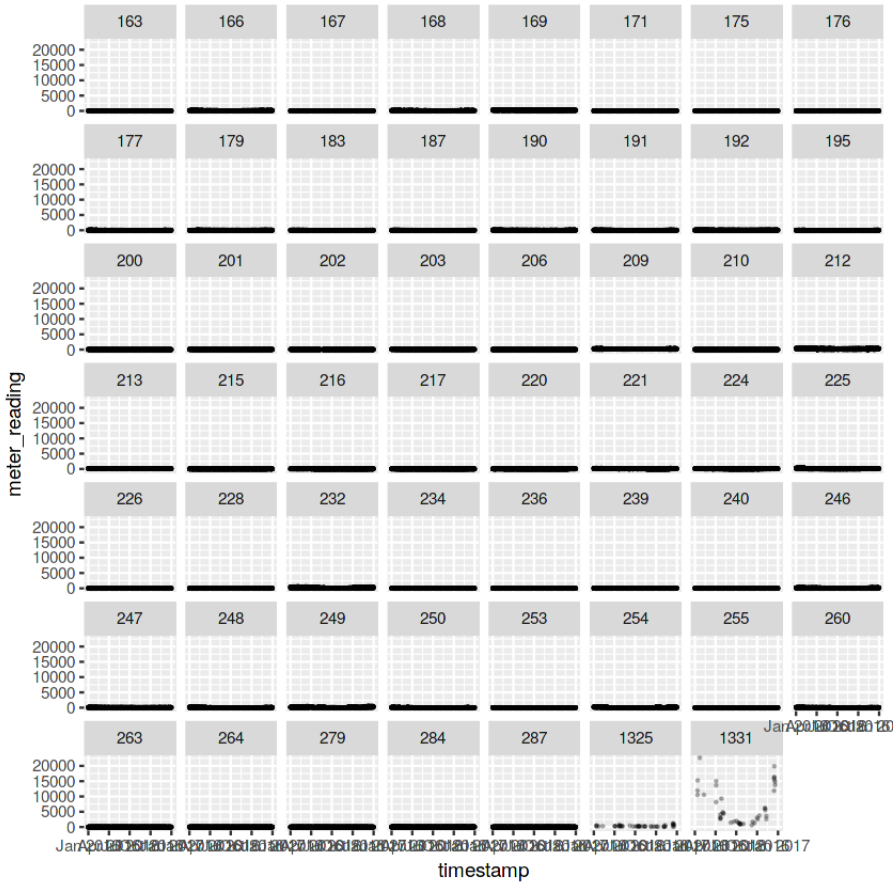
In [24]:
```
ggplot(subset_1, aes(timestamp, meter_reading, group = building_id)) +
  geom_point(size = .25, alpha = .25) +
  facet_wrap(vars(building_id))
```



In [25]:
```
ggplot(subset_2, aes(timestamp, meter_reading, group = building_id)) +
  geom_point(size = .25, alpha = .25) +
  facet_wrap(vars(building_id))
```



In [26]:
```
ggplot(subset_3, aes(timestamp, meter_reading, group = building_id)) +
  geom_point(size = .25, alpha = .25) +
  facet_wrap(vars(building_id))
```



Build Linear Model by Meter Type

In [27]:
```
formula  <- as.formula(
 "meter_reading ~
    primary_use +
    square_feet +
    year_built +
    air_temperature +
    cloud_coverage +
    dew_temperature +
    precip_depth_1_hr +
    sea_level_pressure +
    wind_direction +
    wind_speed +
    week_of_year +
    day_of_week +
    hour_of_day")
formula
```

```
meter_reading ~ primary_use + square_feet + year_built + air_tempera
ture +
    cloud_coverage + dew_temperature + precip_depth_1_hr + sea_level
_pressure +
    wind_direction + wind_speed + week_of_year + day_of_week +
    hour_of_day
```

In [28]:
```
lm_0 <- lm(formula, subset_0)
lm_1 <- lm(formula, subset_1)
lm_2 <- lm(formula, subset_2)
lm_3 <- lm(formula, subset_3)
```

In [29]:
```
# Output adjusted R-squared for each linear model.

summary(lm_0)$adj.r.square
summary(lm_1)$adj.r.square
summary(lm_2)$adj.r.square
summary(lm_3)$adj.r.square
```

0.422747175087908

0.365149327342388

0.35350284368138

0.0770067069705779

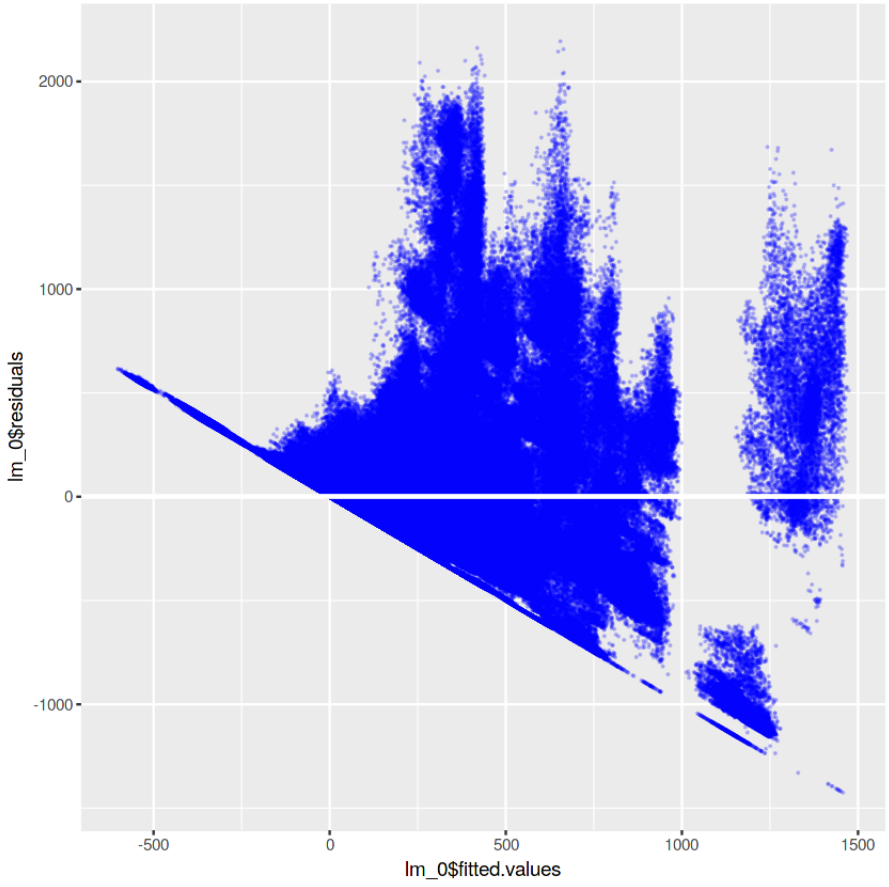
Plot residuals from linear model fits by meter type.

In [30]:
```
fit_0 <- tibble(lm_0$fitted.values, lm_0$residuals)
glimpse(fit_0)
summary(fit_0)
```

```
Observations: 2,023,767
Variables: 2
$ `lm_0$fitted.values` <dbl> -17.32284, -33.08007, 105.73286, 269.05
249, 71.8…
$ `lm_0$residuals`     <dbl> 25.822845, 33.601070, 137.767139, -189.
564485, -…
```

```
 lm_0$fitted.values lm_0$residuals
 Min.   :-601.69    Min.   :-1425.27
 1st Qu.:  75.78    1st Qu.: -108.57
 Median : 159.37    Median :  -30.07
 Mean   : 191.51    Mean   :    0.00
 3rd Qu.: 258.36    3rd Qu.:   57.11
 Max.   :1473.47    Max.   : 2193.41
```

In [31]:
```
ggplot(fit_0, aes(lm_0$fitted.values, lm_0$residuals)) +
  geom_point(size = .1, alpha = .2, color = "blue") +
  geom_hline(yintercept = 0, color = "white", size = 1)
```

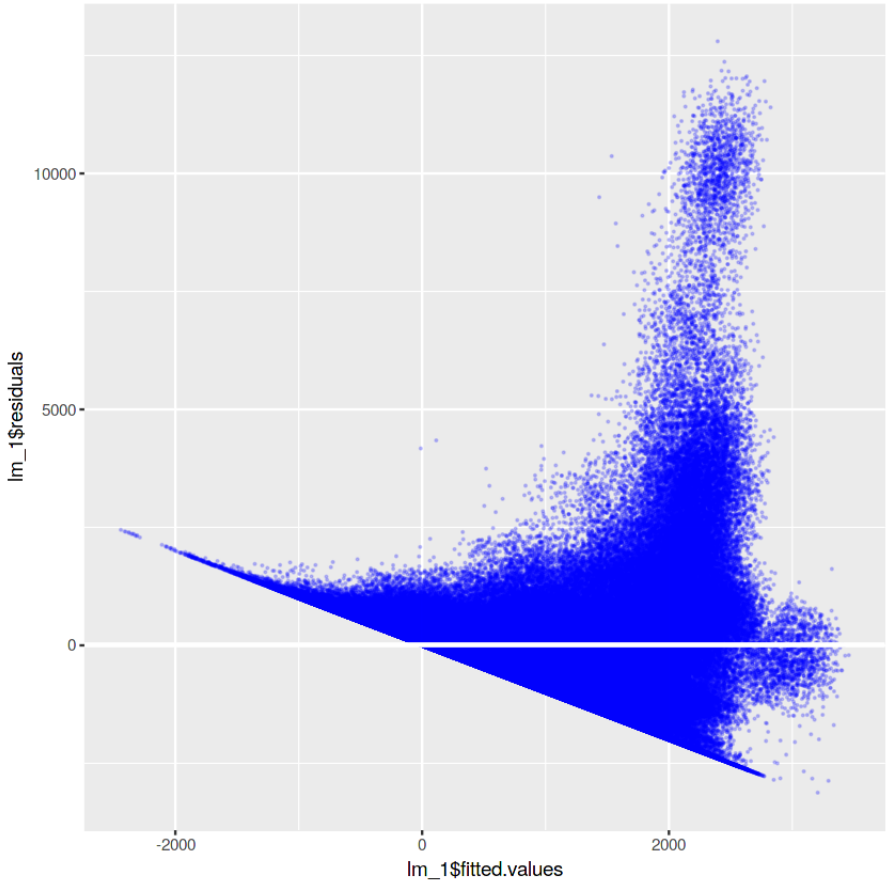


In [32]:
```
fit_1 <- tibble(lm_1$fitted.values, lm_1$residuals)
glimpse(fit_1)
summary(fit_1)
```

```
Observations: 539,611
Variables: 2
$ `lm_1$fitted.values` <dbl> 154.69056, 125.25929, 1415.81701, -84.7
3113, 733…
$ `lm_1$residuals`     <dbl> -154.690558, -125.259287, -1215.568009,
84.73113…
```

```
 lm_1$fitted.values lm_1$residuals
 Min.   :-2442.2    Min.   :-3121.6
 1st Qu.:  116.3    1st Qu.: -515.2
 Median :  573.3    Median : -113.7
 Mean   :  671.1    Mean   :    0.0
 3rd Qu.: 1149.1    3rd Qu.:  308.2
 Max.   : 3458.1    Max.   :12800.8
```

In [33]:
```
ggplot(fit_1, aes(lm_1$fitted.values, lm_1$residuals)) +
  geom_point(size = .1, alpha = .2, color = "blue") +
  geom_hline(yintercept = 0, color = "white", size = 1)
```

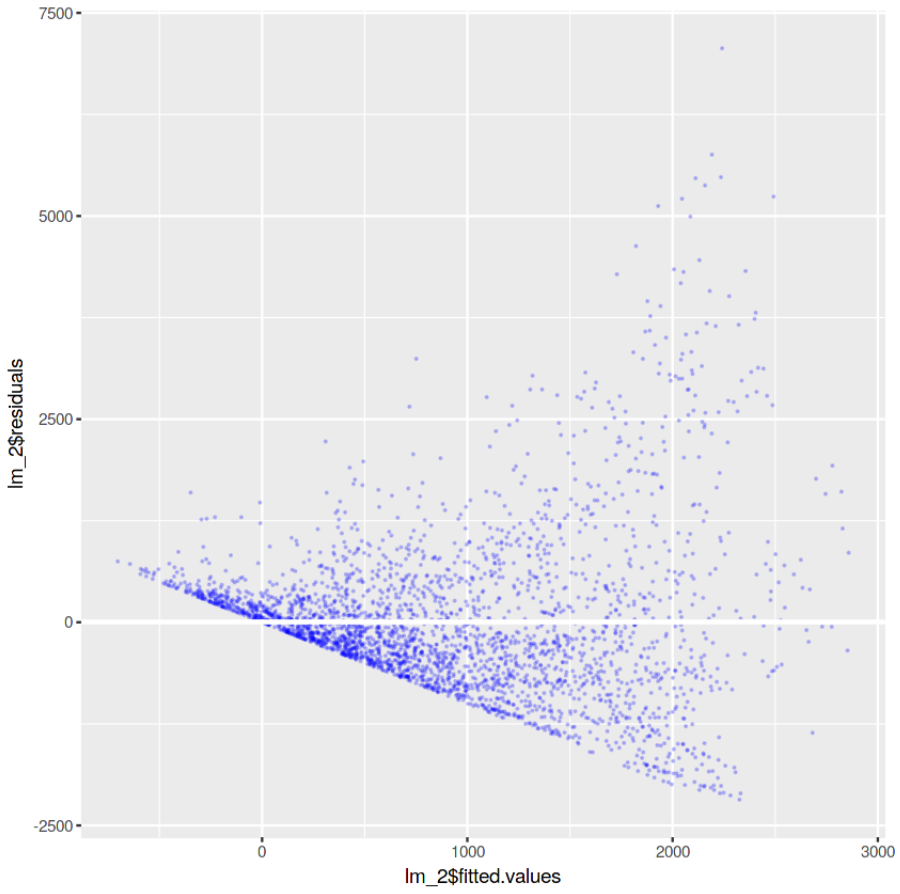


In [34]:
```
fit_2 <- tibble(lm_2$fitted.values, lm_2$residuals)
glimpse(fit_2)
summary(fit_2)
```

```
Observations: 2,970
Variables: 2
$ `lm_2$fitted.values` <dbl> 1227.3808, 896.5069, 2078.2300, 2139.26
46, 1774.…
$ `lm_2$residuals`     <dbl> 1167.2392, -551.6139, 2866.4700, -434.8
246, -116…
```

```
 lm_2$fitted.values lm_2$residuals
 Min.   :-703.2     Min.   :-2182.7
 1st Qu.: 369.2     1st Qu.: -516.4
 Median : 826.5     Median : -117.1
 Mean   : 903.6     Mean   :    0.0
 3rd Qu.:1420.2     3rd Qu.:  312.4
 Max.   :2858.0     Max.   : 7063.8
```

In [35]:
```
ggplot(fit_2, aes(lm_2$fitted.values, lm_2$residuals)) +
  geom_point(size = .1, alpha = .2, color = "blue") +
  geom_hline(yintercept = 0, color = "white", size = 1)
```

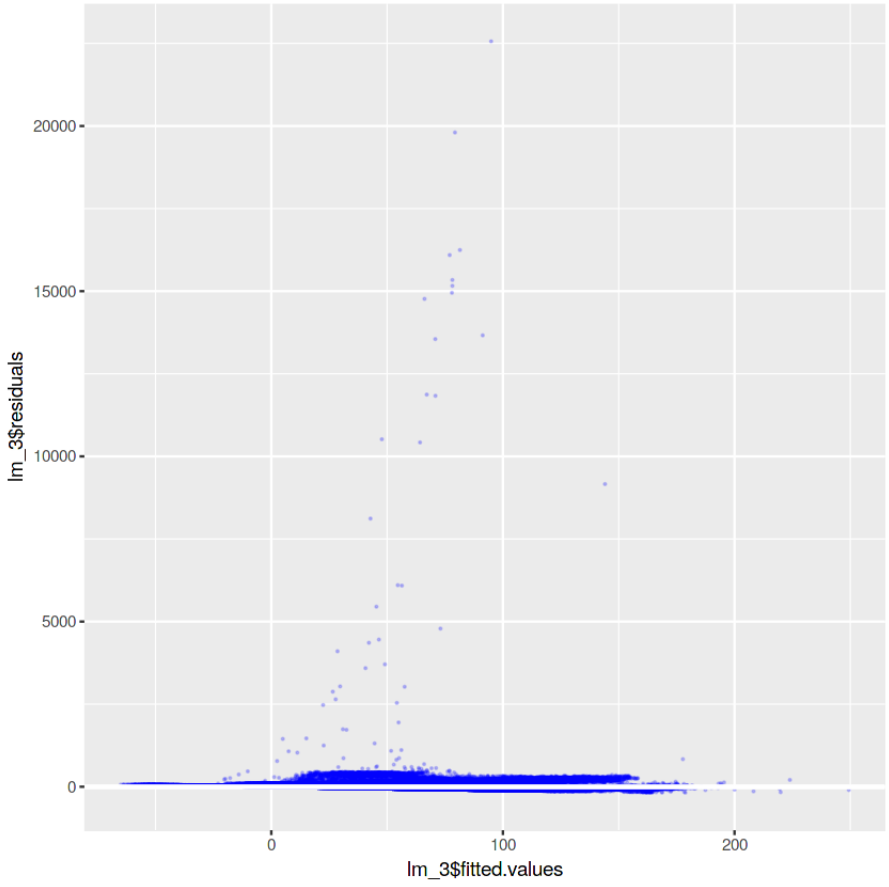


In [36]:
```
fit_3 <- tibble(lm_3$fitted.values, lm_3$residuals)
glimpse(fit_3)
summary(fit_3)
```

```
Observations: 313,312
Variables: 2
$ `lm_3$fitted.values` <dbl> 65.15643, 172.21820, 44.30547, 115.7511
6, 115.97…
$ `lm_3$residuals`     <dbl> -24.126526, -40.336198, -3.275566, 68.8
83840, 18…
```

```
 lm_3$fitted.values lm_3$residuals
 Min.   :-64.98     Min.   : -178.76
 1st Qu.: 17.09     1st Qu.:  -34.83
 Median : 39.60     Median :  -11.32
 Mean   : 41.48     Mean   :    0.00
 3rd Qu.: 63.31     3rd Qu.:   15.78
 Max.   :249.36     Max.   :22563.49
```

In [37]:
```
ggplot(fit_3, aes(lm_3$fitted.values, lm_3$residuals)) +
  geom_point(size = .1, alpha = .2, color = "blue") +
  geom_hline(yintercept = 0, color = "white", size = 1)
```



Principle Component Analysis

In [38]:
```
# Split the data set into training and test sets.

set.seed(0)

training.samples <- subset$meter_reading %>% createDataPartition(p = 0
.8, list = FALSE)
dim(training.samples)

train.data <- subset[training.samples,]
dim(train.data)

test.data  <- subset[-training.samples,]
dim(test.data)
```

2303730 1

2303730 17

575930 17

In [39]:
```
# Create training subsets for each meter type, excluding redundant, id
entifying, and single-valued variables.

train.data.0 <- train.data %>% filter(meter == 0) %>% select(-building
_id, -meter, -timestamp)
dim(train.data.0)

train.data.1 <- train.data %>% filter(meter == 1) %>% select(-building
_id, -meter, -timestamp)
dim(train.data.1)

train.data.2 <- train.data %>% filter(meter == 2) %>% select(-building
_id, -meter, -timestamp)
dim(train.data.2)

train.data.3 <- train.data %>% filter(meter == 3) %>% select(-building
_id, -meter, -timestamp)
dim(train.data.3)
```

1618581 14

431891 14

2375 14

250883 14

In [40]:
```
# Create test subsets for each meter type, excluding redundant, identi
fying, and single-valued variables.

test.data.0 <- test.data %>% filter(meter == 0) %>% select(-building_i
d, -meter, -timestamp)
dim(test.data.0)

test.data.1 <- test.data %>% filter(meter == 1) %>% select(-building_i
d, -meter, -timestamp)
dim(test.data.1)

test.data.2 <- test.data %>% filter(meter == 2) %>% select(-building_i
d, -meter, -timestamp)
dim(test.data.2)

test.data.3 <- test.data %>% filter(meter == 3) %>% select(-building_i
d, -meter, -timestamp)
dim(test.data.3)
```

```
405186  14
```

```
107720  14
```

```
595  14
```

```
62429  14
```

PCA - Meter Type 0

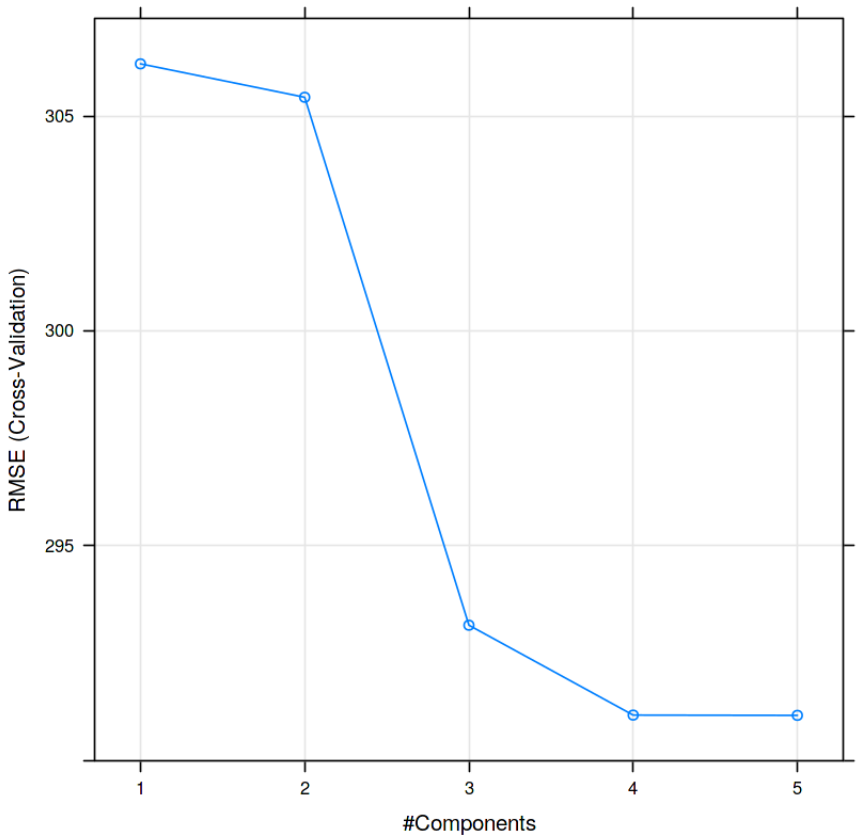
In [25]:
```
# Build models on the training sets.

set.seed(0)

model_0 <- train(
  meter_reading~., data = train.data.0, method = "pcr",
  preProcess = c("center", "scale"),
  trControl = trainControl("cv", number = 5),
  tuneLength = 5)

# Plot model RMSE vs different values of components.
plot(model_0)

# Print the best tuning parameter ncomp that minimizes the cross-valid
ation error, RMSE.
model_0$bestTune
```

A

data.frame:

1 × 1

	ncomp
	<dbl>
5	5



In [26]:
```
# Summarize the final model
summary(model_0$finalModel)
```

```
Data:   X dimension: 1618581 26
        Y dimension: 1618581 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X           7.870   14.079   19.731    24.73    29.16
.outcome    1.653    2.153    9.867    11.15    11.15
```

In [27]:
```
# Make predictions.
predictions <- model_0 %>% predict(test.data.0)

# Model performance metrics
data.frame(
  RMSE = caret::RMSE(predictions, test.data.0$meter_reading),
  Rsquare = caret::R2(predictions, test.data.0$meter_reading))
```

A data.frame: 1 × 2

RMSE	Rsquare
<dbl>	<dbl>
291.7046	0.108999

PCA - Meter Type 1

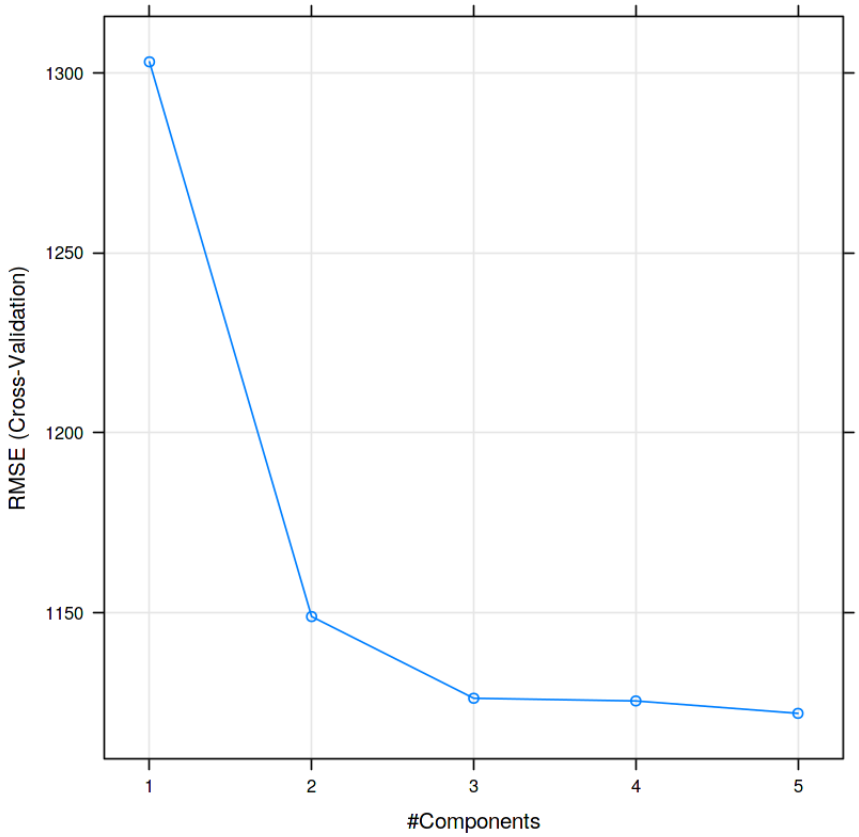
In [22]:
```
# Build models on the training sets.

set.seed(0)

model_1 <- train(
  meter_reading~., data = train.data.1, method = "pcr",
  preProcess = c("center", "scale"),
  trControl = trainControl("cv", number = 5),
  tuneLength = 5)

# Plot model RMSE vs different values of components.
plot(model_1)

# Print the best tuning parameter ncomp that minimizes the cross-valid
ation error, RMSE.
model_1$bestTune
```

A

data.frame:

1 × 1

	ncomp
	<dbl>
5	5



In [23]:
```
# Summarize the final model
summary(model_1$finalModel)
```

```
Data:   X dimension: 431891 22
        Y dimension: 431891 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X          9.4188    17.30    24.50    29.95    35.24
.outcome   0.2316    22.45    25.48    25.58    26.04
```

In [24]:
```
# Make predictions.
predictions <- model_1 %>% predict(test.data.1)

# Model performance metrics
data.frame(
  RMSE = caret::RMSE(predictions, test.data.1$meter_reading),
  Rsquare = caret::R2(predictions, test.data.1$meter_reading))
```

A data.frame: 1 × 2

RMSE	Rsquare
<dbl>	<dbl>
1095.613	0.2671053

PCA - Meter Type 2

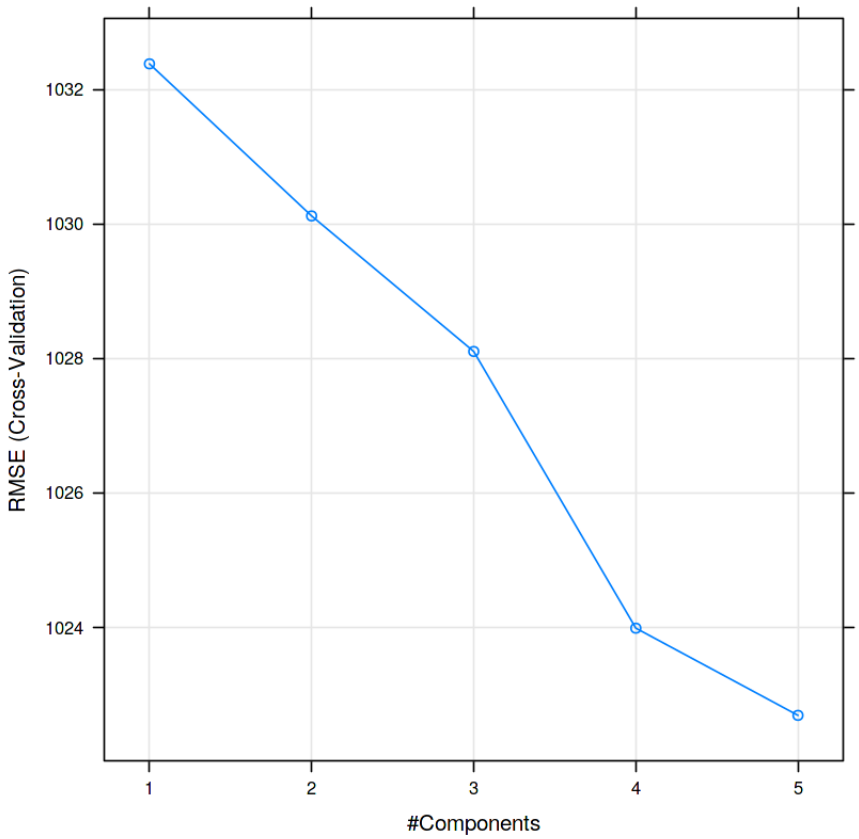
In [18]:
```
# Build models on the training sets.

set.seed(0)

model_2 <- train(
  meter_reading~., data = train.data.2, method = "pcr",
  preProcess = c("center", "scale"),
  trControl = trainControl("cv", number = 5),
  tuneLength = 5)

# Plot model RMSE vs different values of components.
plot(model_2)

# Print the best tuning parameter ncomp that minimizes the cross-valid
ation error, RMSE.
model_2$bestTune
```

A

data.frame:

1 × 1

	ncomp
	<dbl>
5	5



In [20]:
```
# Summarize the final model
summary(model_2$finalModel)
```

```
Data:   X dimension: 2375 19
        Y dimension: 2375 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X           14.00    24.78    33.20    41.21    47.82
.outcome    22.94    23.38    23.63    24.49    24.75
```

In [21]:
```
# Make predictions.
predictions <- model_2 %>% predict(test.data.2)

# Model performance metrics
data.frame(
  RMSE = caret::RMSE(predictions, test.data.2$meter_reading),
  Rsquare = caret::R2(predictions, test.data.2$meter_reading))
```

A data.frame: 1 × 2

RMSE	Rsquare
<dbl>	<dbl>
1015.44	0.2452684

PCA - Meter Type 3

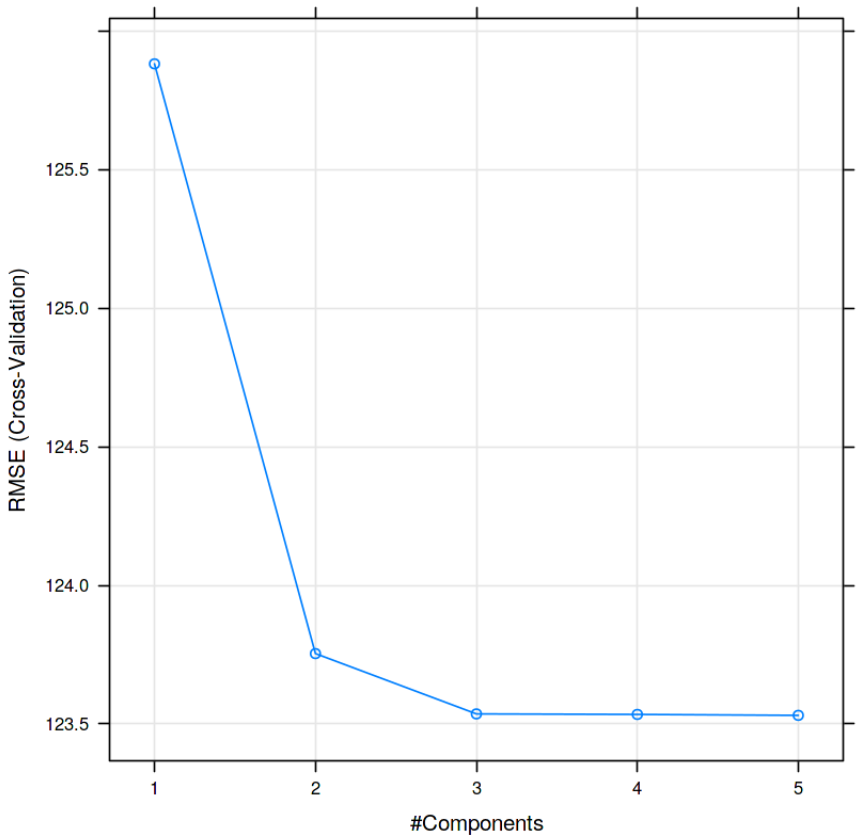
In [15]:
```
# Build models on the training sets.

set.seed(0)

model_3 <- train(
  meter_reading~., data = train.data.3, method = "pcr",
  preProcess = c("center", "scale"),
  trControl = trainControl("cv", number = 5),
  tuneLength = 5)

# Plot model RMSE vs different values of components.
plot(model_3)

# Print the best tuning parameter ncomp that minimizes the cross-valid
ation error, RMSE.
model_3$bestTune
```

A

data.frame:

1 × 1

	ncomp
	<dbl>
5	5



In [16]:
```
# Summarize the final model
summary(model_3$finalModel)
```

```
Data:   X dimension: 250883 17
        Y dimension: 250883 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X          14.231   24.234   33.089   40.449   47.390
.outcome    1.106    4.393    4.727    4.729    4.735
```

In [17]:
```
# Make predictions.
predictions <- model_3 %>% predict(test.data.3)

# Model performance metrics
data.frame(
  RMSE = caret::RMSE(predictions, test.data.3$meter_reading),
  Rsquare = caret::R2(predictions, test.data.3$meter_reading))
```

A data.frame: 1 × 2

RMSE	Rsquare
<dbl>	<dbl>
130.7654	0.04233668

Future work

Try prcomp: e.g., prcomp(subset[,c(1:7,10,11)], center = TRUE,scale. = TRUE)