Dev Template

Group 2

```r
# Fit the OLS model
m.ols <- lm(cnt ~ yr + atemp + holiday + weathersit,
            data = data)
summary(m.ols)
```

```
##
## Call:
## lm(formula = cnt ~ yr + atemp + holiday + weathersit, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3418.7  -612.9    -0.2   732.2  2950.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   489.042    125.858   3.886 0.000111 ***
## yr1          2038.230     74.097  27.507  < 2e-16 ***
## atemp         137.848      4.574  30.136  < 2e-16 ***
## holidayTRUE  -715.501    221.294  -3.233 0.001279 **
## weathersit2  -572.133     79.090  -7.234  1.2e-12 ***
## weathersit3 -2115.108    224.045  -9.441  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 998 on 725 degrees of freedom
## Multiple R-squared:  0.7364, Adjusted R-squared:  0.7346
## F-statistic: 405.1 on 5 and 725 DF,  p-value: < 2.2e-16
```

OLS Model: Paritioning the data by year

```
ols.model1 <- function(data){
  lm(cnt ~ atemp + I(atemp^2) + holiday + weathersit, data)
}
```

The parameter estimates retain their signs if we split the dataset. From an inference POV, this is a good sign. Curiously enough, however, the actual parameter estimates change slightly.

```
ols2011 <- ols.model1(data2011)
summary(ols2011)
```

```
##
## Call:
## lm(formula = cnt ~ atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3004.03  -439.95    52.16   524.30  2178.16
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1337.7637   268.0357  -4.991 9.38e-07 ***
## atemp         341.8815    25.4437  13.437  < 2e-16 ***
## I(atemp^2)     -4.8112     0.5494  -8.758  < 2e-16 ***
## holidayTRUE  -297.3689   232.0310  -1.282    0.201
## weathersit2  -548.9758    82.1682  -6.681 9.06e-11 ***
## weathersit3 -1915.1149   195.1784  -9.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 722.8 on 359 degrees of freedom
## Multiple R-squared:  0.7289, Adjusted R-squared:  0.7252
## F-statistic: 193.1 on 5 and 359 DF,  p-value: < 2.2e-16
```

```
ols2012 <- ols.model1(data2012)
summary(ols2012)
```

```
##
## Call:
## lm(formula = cnt ~ atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3231.0  -618.7     5.0   704.2  3665.1
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2205.3449   457.9463  -4.816 2.16e-06 ***
## atemp         578.4627    41.0607  14.088  < 2e-16 ***
## I(atemp^2)     -9.0043     0.8604 -10.466  < 2e-16 ***
## holidayTRUE  -893.4238   306.6413  -2.914   0.0038 **
## weathersit2  -818.5236   112.3149  -7.288 2.01e-12 ***
## weathersit3 -3127.4767   413.4205  -7.565 3.27e-13 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 995.8 on 360 degrees of freedom
## Multiple R-squared:  0.6943, Adjusted R-squared:   0.69
## F-statistic: 163.5 on 5 and 360 DF,  p-value: < 2.2e-16
```

Accounting for Growth in Bikeshare Users

There is a bit of a problem if we are to use the 2011 data as training data and 2012 as testing data. It assumes the number of users stays constant, which is certainly not true. **This is critical since our response variable is a count**. The number of bikesharing users has significantly grown since 2011 and continues to grow (could we cite something for this?) and treating the dataset as two disjoint datasets might be appopriate. Let's do a hypothesis test to see if $\mu_{2011} - \mu_{2012} = 0$.

```
t.test(data2011$cnt, data2012$cnt)
```

```
##
##  Welch Two Sample t-test
##
## data:  data2011$cnt and data2012$cnt
## t = -18.578, df = 685.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2426.069 -1962.276
## sample estimates:
## mean of x mean of y
##  3405.762  5599.934
```

A possible solution

As such, we need to account for this, since in theory, the true parameter estimate $\hat{\beta}_{atemp}$ should be independent of how many users there are and inturn which year it is. One possible solution is to consider scaling down our response variables in the 2012 data by the following factor $r = \frac{\mu_{2011}}{\mu_{2012}}$. In other words, we want to fit the scaled count variables $\tilde{c_i} = rc_i$. Only this way can we truly and fairly assess the parameter fit $\hat{\beta}$.

```
# Get the ratio of means scaling factor
r <- mean(data2011$cnt)/mean(data2012$cnt)
# Create adjusted count variable in 2012
data2012_ADJ <- data2012 %>% mutate(cnt = r * cnt)
```

Now, let us fit both models and see how the parameter estimates hold up. The parameter estimates indeed seem much more stable!

```
ols.model1(data2011)
```

```
##
## Call:
## lm(formula = cnt ~ atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Coefficients:
## (Intercept)        atemp   I(atemp^2)  holidayTRUE  weathersit2  weathersit3
##   -1337.764      341.882       -4.811     -297.369     -548.976    -1915.115
```

```
ols.model1(data2012_ADJ)
```

```
##
## Call:
## lm(formula = cnt ~ atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Coefficients:
## (Intercept)        atemp   I(atemp^2)  holidayTRUE  weathersit2  weathersit3
##   -1341.244      351.809       -5.476     -543.361     -497.809    -1902.065
```

Another way we could verify this is by using the adjusted count as a variable, reunifying the dataset, and fitting the same model.

```r
ols.adjusted <- lm(cnt_adj ~ atemp + I(atemp^2) + holiday + weathersit, data = data)
summary(ols.adjusted)
```

```
##
## Call:
## lm(formula = cnt_adj ~ atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2851.97  -434.34    20.12   485.73  2490.05
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1331.4167   194.3750  -6.850 1.58e-11 ***
## atemp         345.3555    17.9451  19.245  < 2e-16 ***
## I(atemp^2)     -5.1099     0.3823 -13.365  < 2e-16 ***
## holidayTRUE  -433.3661   149.9795  -2.890  0.00397 **
## weathersit2  -516.1408    54.0633  -9.547  < 2e-16 ***
## weathersit3 -1879.0269   151.9178 -12.369  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 676.1 on 725 degrees of freedom
## Multiple R-squared:  0.7051, Adjusted R-squared:  0.703
## F-statistic: 346.6 on 5 and 725 DF,  p-value: < 2.2e-16
```

```r
residuals_adj <- data.frame(x = data$instant, y = ols.adjusted$residuals)
# Residual plot by instance
p1 <- ggplot(data = residuals_adj) + geom_point(aes(x, y))
```

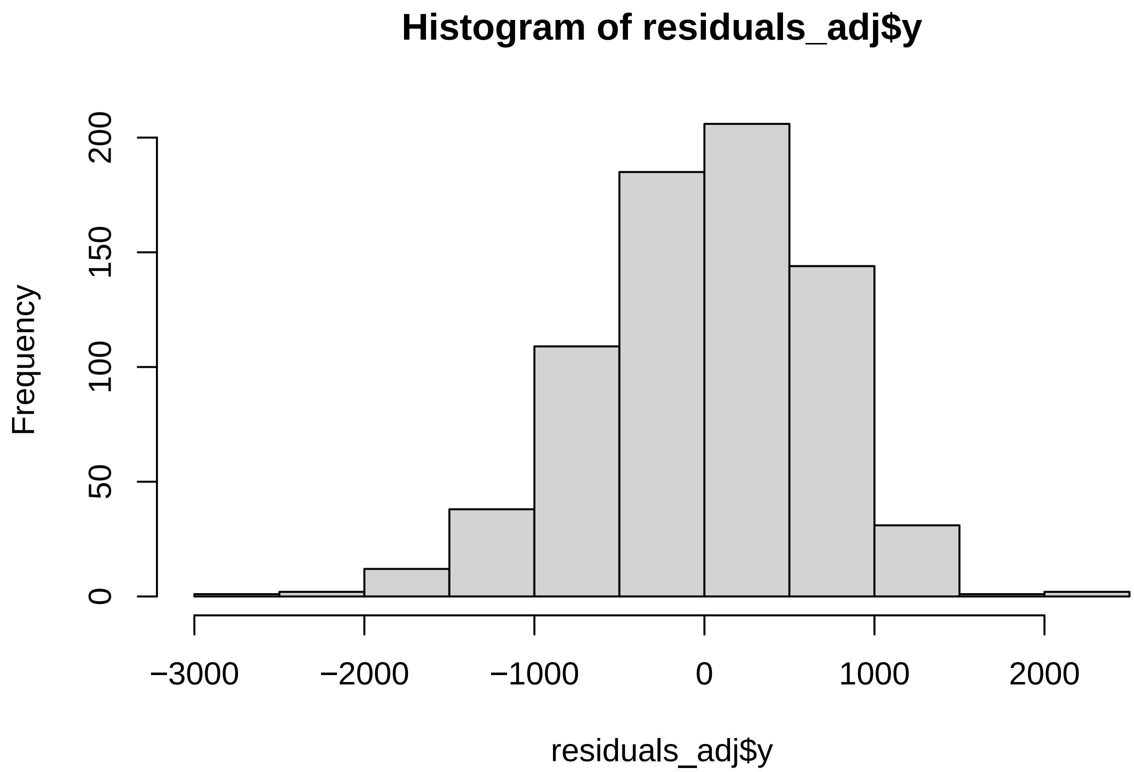
```r
ols.unadjusted <- lm(cnt ~ yr + atemp + I(atemp^2) + holiday + weathersit, data = data)
summary(ols.unadjusted)
```

```
##
## Call:
## lm(formula = cnt ~ yr + atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3630.2  -526.6    12.0   615.7  3291.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2485.6960   259.8010  -9.568  < 2e-16 ***
## yr1          1973.7739    67.2089  29.368  < 2e-16 ***
## atemp         439.4578    24.0369  18.283  < 2e-16 ***
## I(atemp^2)     -6.5205     0.5119 -12.738  < 2e-16 ***
## holidayTRUE  -640.3023   200.2389  -3.198  0.00145 **
## weathersit2  -695.8176    72.1899  -9.639  < 2e-16 ***
## weathersit3 -2343.4495   203.4317 -11.520  < 2e-16 ***
## ---
```

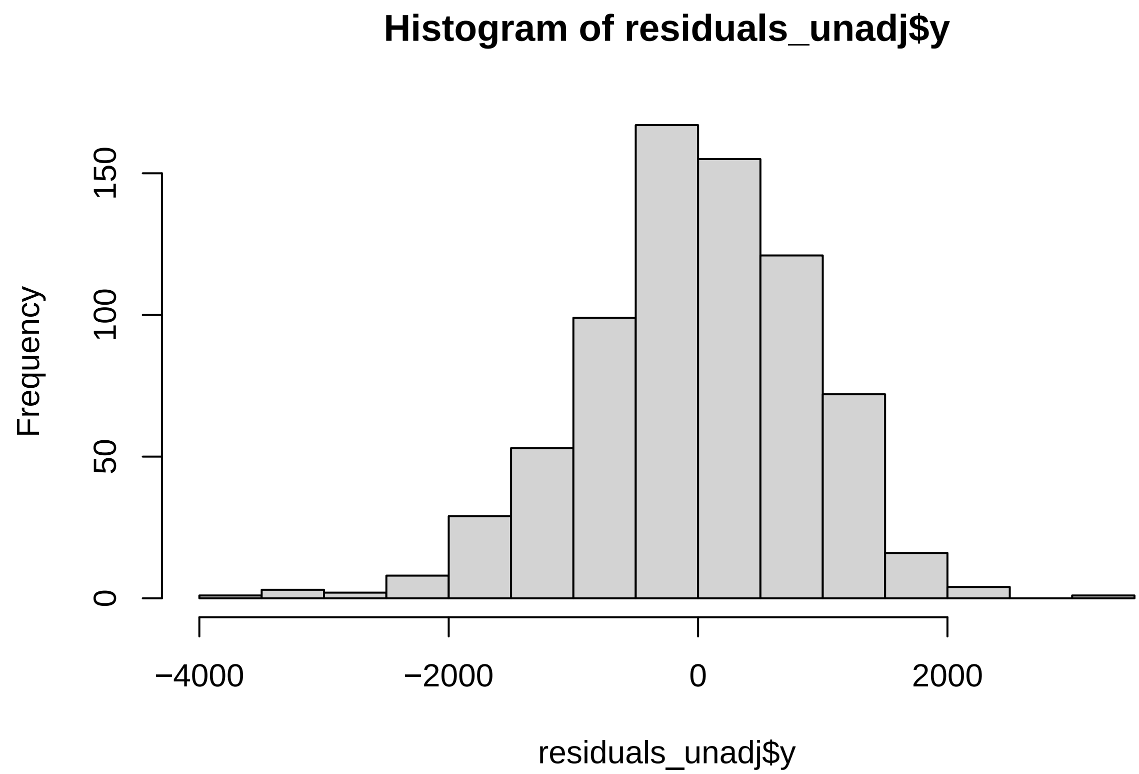
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 902.7 on 724 degrees of freedom
## Multiple R-squared:  0.7847, Adjusted R-squared:  0.7829
## F-statistic: 439.7 on 6 and 724 DF,  p-value: < 2.2e-16
```

```
residuals_unadj <- data.frame(x = data$instant, y = ols.unadjusted$residuals)
# Residual plot by instance
p2 <- ggplot(data = residuals_unadj) + geom_point(aes(x, y))
```

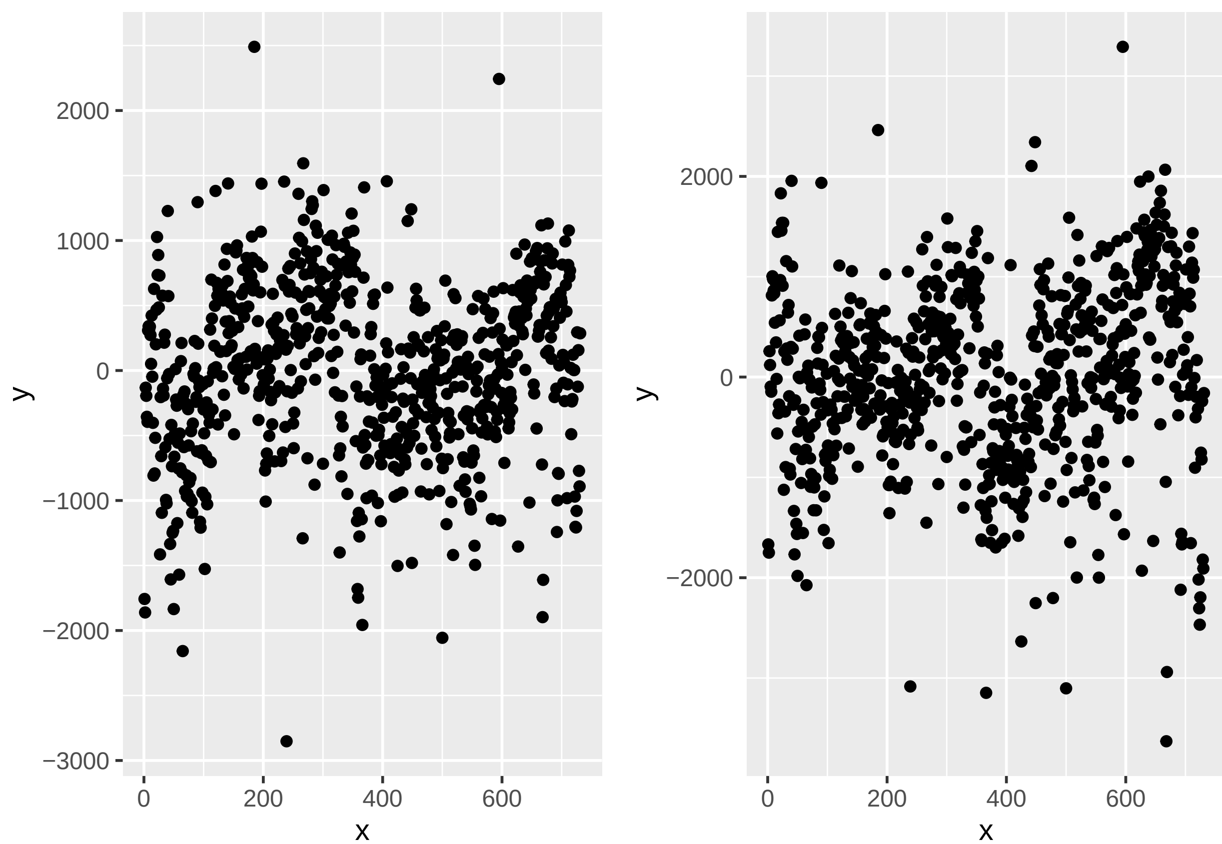
```
hist(residuals_adj$y)
```




```
hist(residuals_unadj$y)
```




```
plot_grid(p1, p2)
```



Model Performance

```
# Train model on 2011 data (don't show it 2012 data)
ols.model <- ols.model1(data = data2011)
summary(ols.model)
```

```
##
## Call:
## lm(formula = cnt ~ atemp + I(atemp^2) + holiday + weathersit,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3004.03  -439.95    52.16   524.30  2178.16
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1337.7637   268.0357  -4.991 9.38e-07 ***
## atemp         341.8815    25.4437  13.437  < 2e-16 ***
## I(atemp^2)     -4.8112     0.5494  -8.758  < 2e-16 ***
## holidayTRUE  -297.3689   232.0310  -1.282    0.201
## weathersit2  -548.9758    82.1682  -6.681 9.06e-11 ***
## weathersit3 -1915.1149   195.1784  -9.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 722.8 on 359 degrees of freedom
## Multiple R-squared:  0.7289, Adjusted R-squared:  0.7252
## F-statistic: 193.1 on 5 and 359 DF,  p-value: < 2.2e-16
```

```
preds2012 <- predict(ols.model, data2012)
# MSEs
MSE_unadj <- sqrt(mean((preds2012 - data2012$cnt)^2))
MSE_unadj
```

```
## [1] 2290.639
```

```
MSE_adj <- sqrt(mean((preds2012 - data2012$cnt_adj)^2))
MSE_adj
```

```
## [1] 653.3103
```

Now lets compare it to if we used the `yr` variable, how does it compare? In theory, we have the same exact information, so we expect the model performance on the testing data to not be too disparate. Why would changing from a indicator variable scheme to a scaled response variable scheme change performance if both use the same "information"?

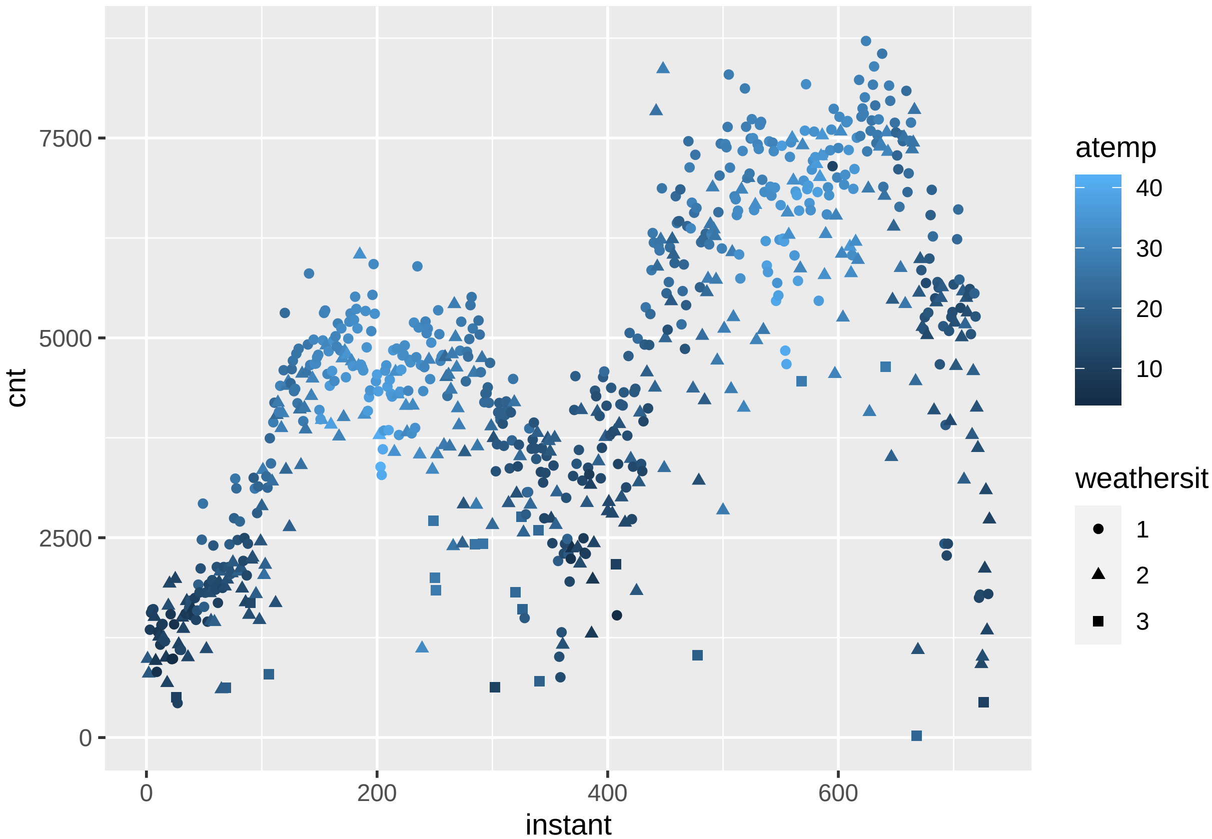
Surprisingly, the model with the scaled response variable scheme did perform better. So, if we account the response variable for expected growth r rather than add a constant $\delta\mu$ to our predictors, this indicates we might get better model performance.

```
preds2012 <- predict(ols.unadjusted, data2012)
MSE <- sqrt(mean((preds2012 - data2012$cnt)^2))
MSE
```

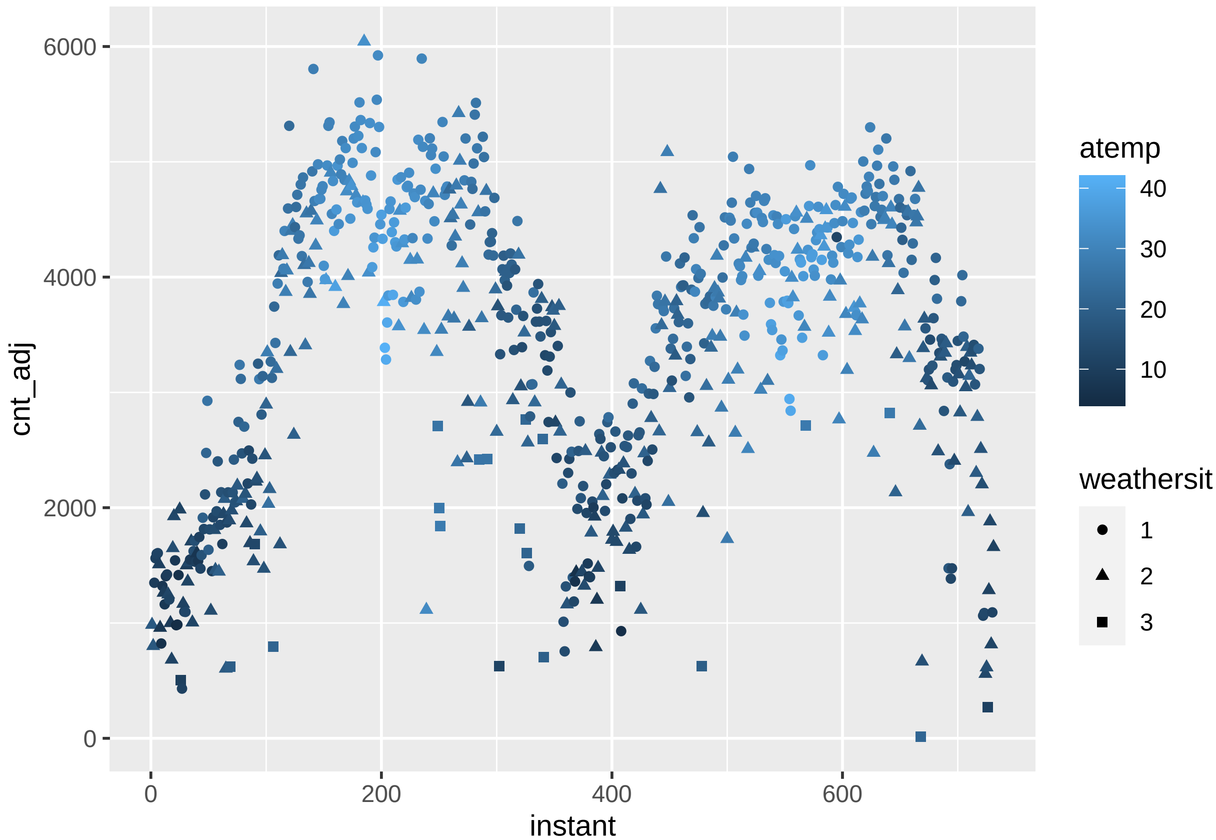
```
## [1] 1022.831
```

Plot showing the scaled and unscaled data series

```
rbind(data2011, data2012) %>%
  ggplot() +
  geom_point(aes(x = instant, y = cnt, color = atemp, shape = weathersit))
```




```
data %>%
  ggplot() +
  geom_point(aes(x = instant, y = cnt_adj, color = atemp, shape = weathersit))
```



atemp
40
30
20
10
weathersit
1
2
3
6000
4000
2000
0
cnt_adj
0
200
400
600
instant