library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ─────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(readr)  
bike <- read\_csv("hour.csv")

## Parsed with column specification:  
## cols(  
## instant = col\_integer(),  
## dteday = col\_date(format = ""),  
## season = col\_integer(),  
## yr = col\_integer(),  
## mnth = col\_integer(),  
## hr = col\_integer(),  
## holiday = col\_integer(),  
## weekday = col\_integer(),  
## workingday = col\_integer(),  
## weathersit = col\_integer(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_integer(),  
## registered = col\_integer(),  
## count = col\_integer()  
## )

View(bike)

bike = bike %>% mutate(season = as\_factor(as.character(season))) %>%  
 mutate(season = fct\_recode(season,  
 "Spring" = "1",  
 "Summer" = "2",  
 "Fall" = "3",  
 "Winter" = "4"))  
  
bike = bike %>% mutate(yr = as\_factor(as.character(yr)))  
bike = bike %>% mutate(mnth = as\_factor(as.character(mnth)))  
bike = bike %>% mutate(hr = as\_factor(as.character(hr)))  
  
bike = bike %>% mutate(holiday = as\_factor(as.character(holiday))) %>%  
 mutate(holiday = fct\_recode(holiday,  
 "NotHoliday" = "0",  
 "Holiday" = "1"))  
  
bike = bike %>% mutate(workingday = as\_factor(as.character(workingday))) %>%  
 mutate(workingday = fct\_recode(workingday,  
 "NotWorkingDay" = "0",  
 "WorkingDay" = "1"))  
  
bike = bike %>% mutate(weathersit = as\_factor(as.character(weathersit))) %>%  
 mutate(weathersit = fct\_recode(weathersit,  
 "NoPrecip" = "1",  
 "Misty" = "2",  
 "LightPrecip" = "3",  
 "HeavyPrecip" = "4"))  
  
bike = bike %>% mutate(weekday = as\_factor(as.character(weekday))) %>%  
 mutate(weekday = fct\_recode(weekday,  
 "Sunday" = "0",  
 "Monday" = "1",  
 "Tuesday" = "2",  
 "Wednesday" = "3",  
 "Thursday" = "4",  
 "Friday" = "5",  
 "Saturday" = "6"))

train.rows = createDataPartition(y = bike$count, p=0.7, list = FALSE)  
train = bike[train.rows,]   
test = bike[-train.rows,]

There are 12,167 rows in the training data set and 5,212 rows in the test data set

model1 = lm(count ~ season, train)  
summary(model1)

##   
## Call:  
## lm(formula = count ~ season, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -235.74 -117.74 -39.16 85.08 767.08   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 112.165 3.218 34.86 <2e-16 \*\*\*  
## seasonSummer 92.810 4.496 20.64 <2e-16 \*\*\*  
## seasonFall 124.580 4.479 27.82 <2e-16 \*\*\*  
## seasonWinter 87.751 4.557 19.25 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 175 on 12163 degrees of freedom  
## Multiple R-squared: 0.06476, Adjusted R-squared: 0.06453   
## F-statistic: 280.8 on 3 and 12163 DF, p-value: < 2.2e-16

The R squared-value is low when predicting count off of season. It does not appear that season and count have a strong correlation when looking at the training data set.

model2 = lm(count ~ mnth, train)  
summary(model2)

##   
## Call:  
## lm(formula = count ~ mnth, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -243.32 -124.37 -37.71 84.02 736.20   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 95.979 5.549 17.295 < 2e-16 \*\*\*  
## mnth2 14.658 7.924 1.850 0.0644 .   
## mnth3 61.065 7.784 7.845 4.70e-15 \*\*\*  
## mnth4 86.710 7.794 11.126 < 2e-16 \*\*\*  
## mnth5 124.729 7.710 16.178 < 2e-16 \*\*\*  
## mnth6 148.343 7.801 19.015 < 2e-16 \*\*\*  
## mnth7 137.474 7.753 17.732 < 2e-16 \*\*\*  
## mnth8 140.002 7.747 18.071 < 2e-16 \*\*\*  
## mnth9 144.822 7.788 18.596 < 2e-16 \*\*\*  
## mnth10 126.229 7.809 16.165 < 2e-16 \*\*\*  
## mnth11 82.077 7.826 10.487 < 2e-16 \*\*\*  
## mnth12 46.393 7.713 6.015 1.85e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 174.1 on 12155 degrees of freedom  
## Multiple R-squared: 0.07562, Adjusted R-squared: 0.07478   
## F-statistic: 90.4 on 11 and 12155 DF, p-value: < 2.2e-16

The R squared-value is low when predicting count off of month. It does not appear that month and count have a strong correlation when looking at the training data set.

model3 = lm(count ~ hr, train)  
summary(model3)

##   
## Call:  
## lm(formula = count ~ hr, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -450.49 -61.69 -5.70 50.36 554.11   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 54.115 5.636 9.601 < 2e-16 \*\*\*  
## hr1 -21.979 8.120 -2.707 0.00680 \*\*   
## hr2 -31.953 8.003 -3.993 6.57e-05 \*\*\*  
## hr3 -42.555 8.116 -5.244 1.60e-07 \*\*\*  
## hr4 -47.756 8.056 -5.928 3.15e-09 \*\*\*  
## hr5 -33.947 7.995 -4.246 2.19e-05 \*\*\*  
## hr6 23.570 7.987 2.951 0.00317 \*\*   
## hr7 152.652 7.956 19.187 < 2e-16 \*\*\*  
## hr8 306.308 7.983 38.371 < 2e-16 \*\*\*  
## hr9 163.963 8.015 20.458 < 2e-16 \*\*\*  
## hr10 119.267 8.027 14.859 < 2e-16 \*\*\*  
## hr11 152.582 8.019 19.028 < 2e-16 \*\*\*  
## hr12 197.613 8.043 24.569 < 2e-16 \*\*\*  
## hr13 200.535 7.914 25.338 < 2e-16 \*\*\*  
## hr14 184.351 7.983 23.093 < 2e-16 \*\*\*  
## hr15 194.666 7.979 24.397 < 2e-16 \*\*\*  
## hr16 258.686 7.922 32.655 < 2e-16 \*\*\*  
## hr17 411.376 8.047 51.120 < 2e-16 \*\*\*  
## hr18 368.772 7.979 46.218 < 2e-16 \*\*\*  
## hr19 257.057 7.952 32.326 < 2e-16 \*\*\*  
## hr20 176.544 7.952 22.201 < 2e-16 \*\*\*  
## hr21 115.508 8.043 14.361 < 2e-16 \*\*\*  
## hr22 78.672 7.963 9.879 < 2e-16 \*\*\*  
## hr23 35.398 7.975 4.439 9.14e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 127.8 on 12143 degrees of freedom  
## Multiple R-squared: 0.5024, Adjusted R-squared: 0.5014   
## F-statistic: 533 on 23 and 12143 DF, p-value: < 2.2e-16

The R squared-value is low when predicting count off of hour. Hour and count may be correlated but do not have a strong correlation.

model4 = lm(count ~ holiday, train)  
summary(model4)

##   
## Call:  
## lm(formula = count ~ holiday, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -189.43 -148.43 -47.43 91.57 786.57   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 190.428 1.664 114.452 < 2e-16 \*\*\*  
## holidayHoliday -35.229 9.852 -3.576 0.000351 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 180.9 on 12165 degrees of freedom  
## Multiple R-squared: 0.00105, Adjusted R-squared: 0.0009678   
## F-statistic: 12.79 on 1 and 12165 DF, p-value: 0.0003506

The R squared-value is very small when predicting count off of holiday. It does not appear that holiday and count have a strong correlation when looking at the training data set.

model5 = lm(count ~ weekday, train)  
summary(model5)

##   
## Call:  
## lm(formula = count ~ weekday, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -194.33 -148.33 -47.33 91.37 784.74   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 191.3958 4.3301 44.202 <2e-16 \*\*\*  
## weekdaySunday -13.5301 6.1271 -2.208 0.0272 \*   
## weekdayMonday -7.5145 6.1413 -1.224 0.2211   
## weekdayTuesday 3.9385 6.1521 0.640 0.5221   
## weekdayWednesday 0.8617 6.1254 0.141 0.8881   
## weekdayThursday 1.5003 6.1360 0.245 0.8068   
## weekdayFriday 0.9587 6.1089 0.157 0.8753   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 180.9 on 12160 degrees of freedom  
## Multiple R-squared: 0.00101, Adjusted R-squared: 0.000517   
## F-statistic: 2.049 on 6 and 12160 DF, p-value: 0.05583

The R squared-value is very small when predicting count off of week day. It does not appear that week day and count have a strong correlation when looking at the training data set.

model6 = lm(count ~ temp, train)  
summary(model6)

##   
## Call:  
## lm(formula = count ~ temp, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -291.6 -110.1 -33.0 77.3 744.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.5237 4.1536 -0.126 0.9   
## temp 382.1962 7.7940 49.037 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 165.4 on 12165 degrees of freedom  
## Multiple R-squared: 0.165, Adjusted R-squared: 0.165   
## F-statistic: 2405 on 1 and 12165 DF, p-value: < 2.2e-16

The R squared-value is low when predicting count off of temperature. Temperature and count may be correlated but do not have a strong correlation.

model7 = lm(count ~ weathersit, train)  
summary(model7)

##   
## Call:  
## lm(formula = count ~ weathersit, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -202.72 -141.72 -44.72 88.74 776.76   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 203.718 2.005 101.624 < 2e-16 \*\*\*  
## weathersitMisty -26.457 3.763 -7.031 2.17e-12 \*\*\*  
## weathersitLightPrecip -89.480 5.998 -14.918 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -103.718 126.745 -0.818 0.413   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 179.2 on 12163 degrees of freedom  
## Multiple R-squared: 0.01957, Adjusted R-squared: 0.01933   
## F-statistic: 80.93 on 3 and 12163 DF, p-value: < 2.2e-16

The R squared-value is very small when predicting count off of the weather situation. It does not appear that the weather and count have a strong correlation when looking at the training data set.

predict\_train = predict(model6, newdata = train)  
head(train)

## # A tibble: 6 x 17  
## instant dteday season yr mnth hr holiday weekday workingday  
## <int> <date> <fct> <fct> <fct> <fct> <fct> <fct> <fct>   
## 1 1 2011-01-01 Spring 0 1 0 NotHol… Saturd… NotWorkin…  
## 2 3 2011-01-01 Spring 0 1 2 NotHol… Saturd… NotWorkin…  
## 3 4 2011-01-01 Spring 0 1 3 NotHol… Saturd… NotWorkin…  
## 4 5 2011-01-01 Spring 0 1 4 NotHol… Saturd… NotWorkin…  
## 5 6 2011-01-01 Spring 0 1 5 NotHol… Saturd… NotWorkin…  
## 6 8 2011-01-01 Spring 0 1 7 NotHol… Saturd… NotWorkin…  
## # ... with 8 more variables: weathersit <fct>, temp <dbl>, atemp <dbl>,  
## # hum <dbl>, windspeed <dbl>, casual <int>, registered <int>,  
## # count <int>

The training set prediction for temperature looks fairly accurate when comparing the numbers to the original bike dataset.

predict\_test = predict(model6, newdata = test)  
head(test)

## # A tibble: 6 x 17  
## instant dteday season yr mnth hr holiday weekday workingday  
## <int> <date> <fct> <fct> <fct> <fct> <fct> <fct> <fct>   
## 1 2 2011-01-01 Spring 0 1 1 NotHol… Saturd… NotWorkin…  
## 2 7 2011-01-01 Spring 0 1 6 NotHol… Saturd… NotWorkin…  
## 3 10 2011-01-01 Spring 0 1 9 NotHol… Saturd… NotWorkin…  
## 4 13 2011-01-01 Spring 0 1 12 NotHol… Saturd… NotWorkin…  
## 5 18 2011-01-01 Spring 0 1 17 NotHol… Saturd… NotWorkin…  
## 6 20 2011-01-01 Spring 0 1 19 NotHol… Saturd… NotWorkin…  
## # ... with 8 more variables: weathersit <fct>, temp <dbl>, atemp <dbl>,  
## # hum <dbl>, windspeed <dbl>, casual <int>, registered <int>,  
## # count <int>

The test data set is very similar to the train data set predictions for temperature.

SSE = sum((test$count - predict\_test)^2)  
SST = sum((test$count - mean(test$count))^2)  
1 - SSE/SST

## [1] 0.1610688

The R squared value above is very similar to the model’s performance on the training set so we can assume this is a good model and the model will react similarly with new data.