Linear Models in R, the insulate case study

Mauro Gasparini* Vittorio Zampinetti†

MSML, April 16, 2025

The insulate case study

The insulate database contains data about fuel consumption to heat a certain building. Assume (a big assumption!) that we have two random samples of weeks, before (26 weeks) and after (30 weeks) some insulating work was done on the building: in these days we recorded: * quando: an indicator for before/after * Temp = average weekly external temperature in Celsius * Cons: fuel consuption (in ft^3).

We want to study a linear model explaining Cons as a function of quando and Temp.

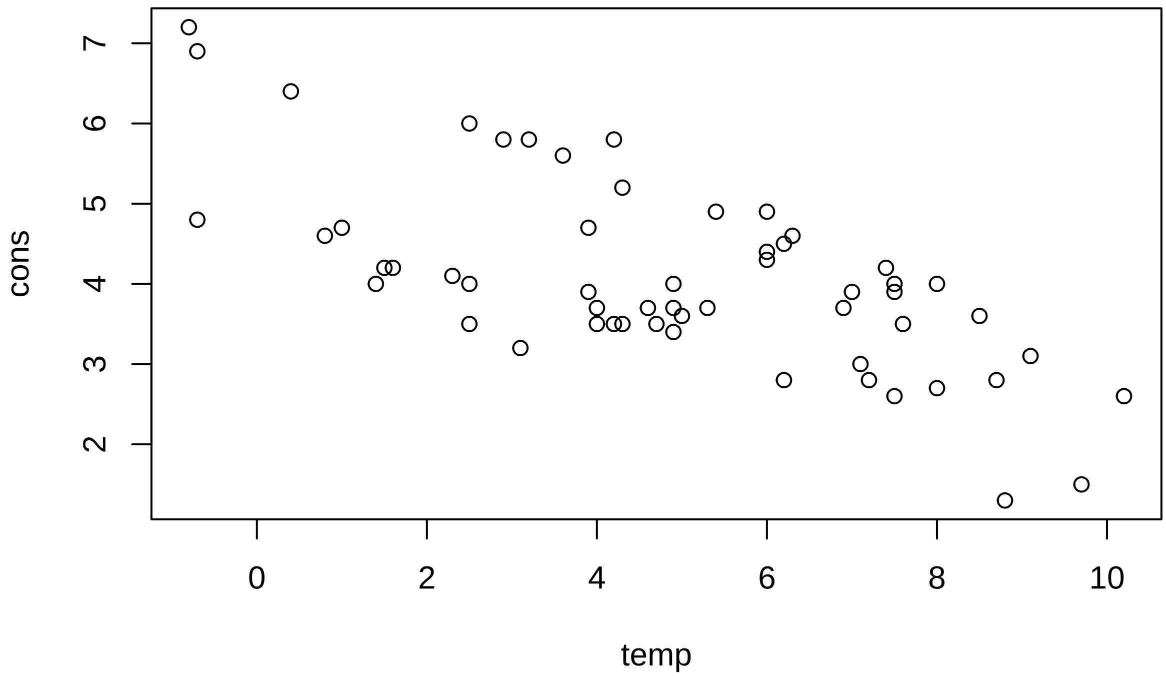
Importing the data from a file and exploring it

```
insulate=read.table("250416 insulate data.txt",
                    col.names=c("quando","temp","cons"))

# take a look at the variable in the database
str(insulate)        # notice quando is a factor with two levels
```

```
## 'data.frame':    56 obs. of  3 variables:
##  $ quando: chr  "prima" "prima" "prima" "prima" ...
##  $ temp  : num  -0.8 -0.7 0.4 2.5 2.9 3.2 3.6 3.9 4.2 4.3 ...
##  $ cons  : num  7.2 6.9 6.4 6 5.8 5.8 5.6 4.7 5.8 5.2 ...
```

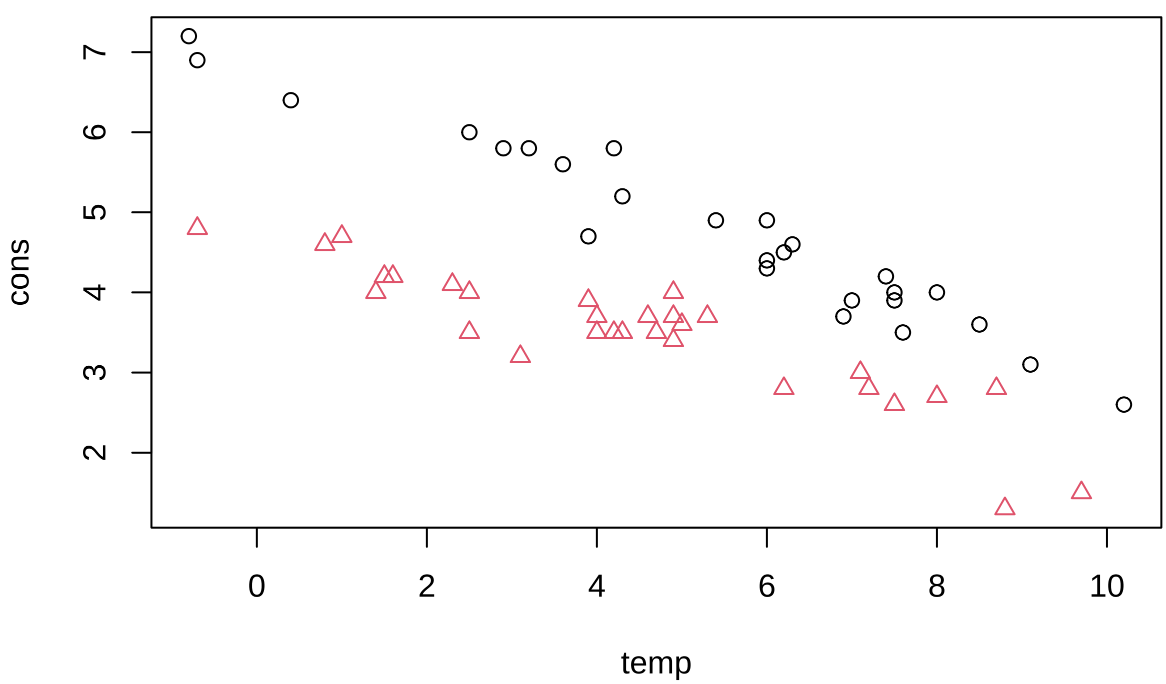
```
# focus on this data frame
attach(insulate)

# Negative relationship between temp and cons
plot(temp,cons)
```

*Politecnico di Torino, mauro.gasparini@polito.it
†Politecnico di Torino, vittorio.zampinetti@polito.it




```r
# Visualizing insulation as well, with some prettyfication
plot(temp,cons,type="n")
points(temp[quando=="prima"],cons[quando=="prima"],pch=1)
points(temp[quando=="dopo"],cons[quando=="dopo"],pch=2,col=2)
title("With and without insulation")
```

With and without insulation



Linear models

Linear model with one quantitative and one binary predictor. Each predictor appear to be necessary when the other is there, since all p-values are close to zero. The all-or-nothing (ANOVA) test is also significant.

```
regr=lm(cons~quando+temp)  ### notice nonalgebraic use of symbol +
summary(regr)
```

```
##
## Call:
## lm(formula = cons ~ quando + temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74236 -0.22291  0.04338  0.24377  0.74314
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.98612    0.10268   48.56   <2e-16 ***
## quandoprima  1.56520    0.09705   16.13   <2e-16 ***
## temp        -0.33670    0.01776  -18.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3574 on 53 degrees of freedom
## Multiple R-squared:  0.9097, Adjusted R-squared:  0.9063
## F-statistic: 267.1 on 2 and 53 DF,  p-value: < 2.2e-16
```

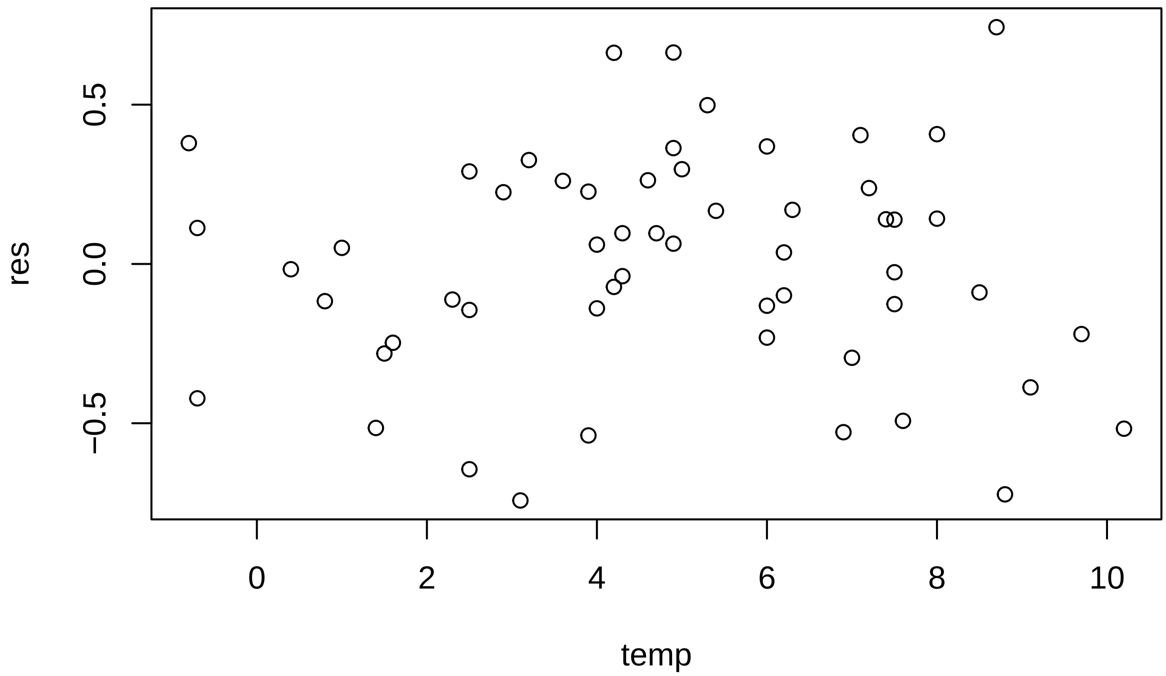
Here are confidence intervals for the regression coefficients

```
confint(regr, level=0.95)
```

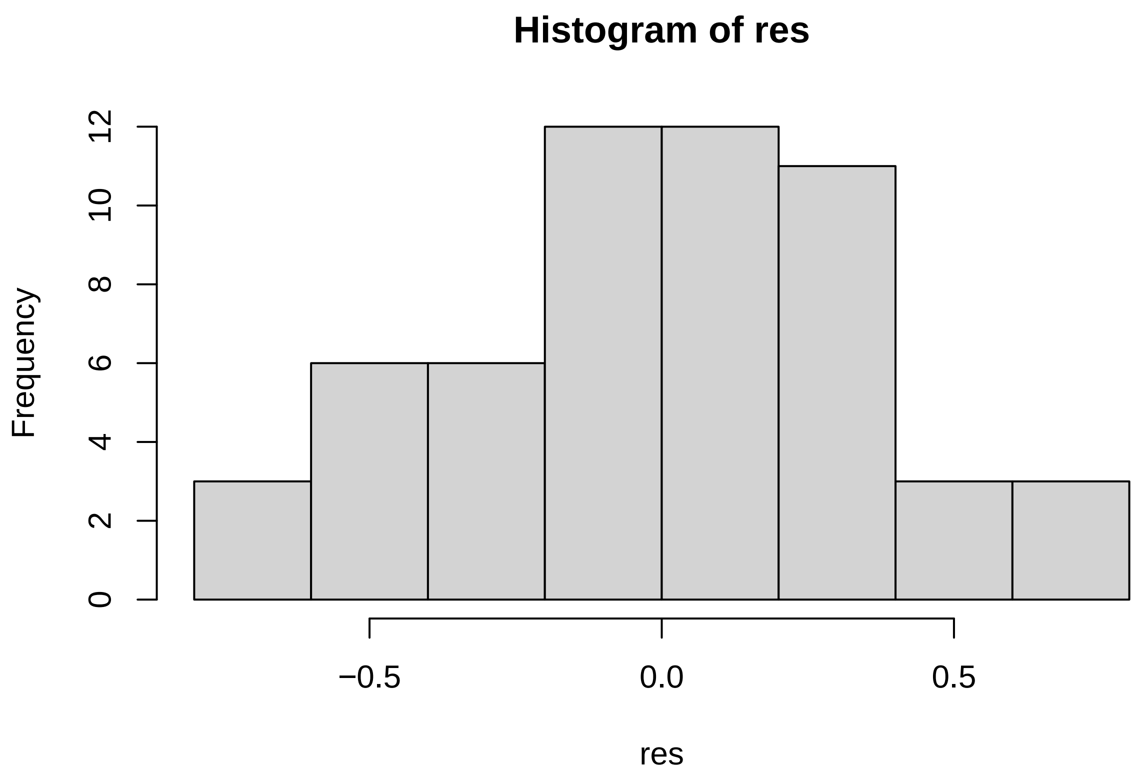
```
##                  2.5 %     97.5 %
## (Intercept)  4.7801676  5.1920806
## quandoprima  1.3705402  1.7598691
## temp        -0.3723252 -0.3010687
```

Now take a look at residuals

```
res=regr$resid
plot(temp,res) # no obvious pattern appears, ok
```



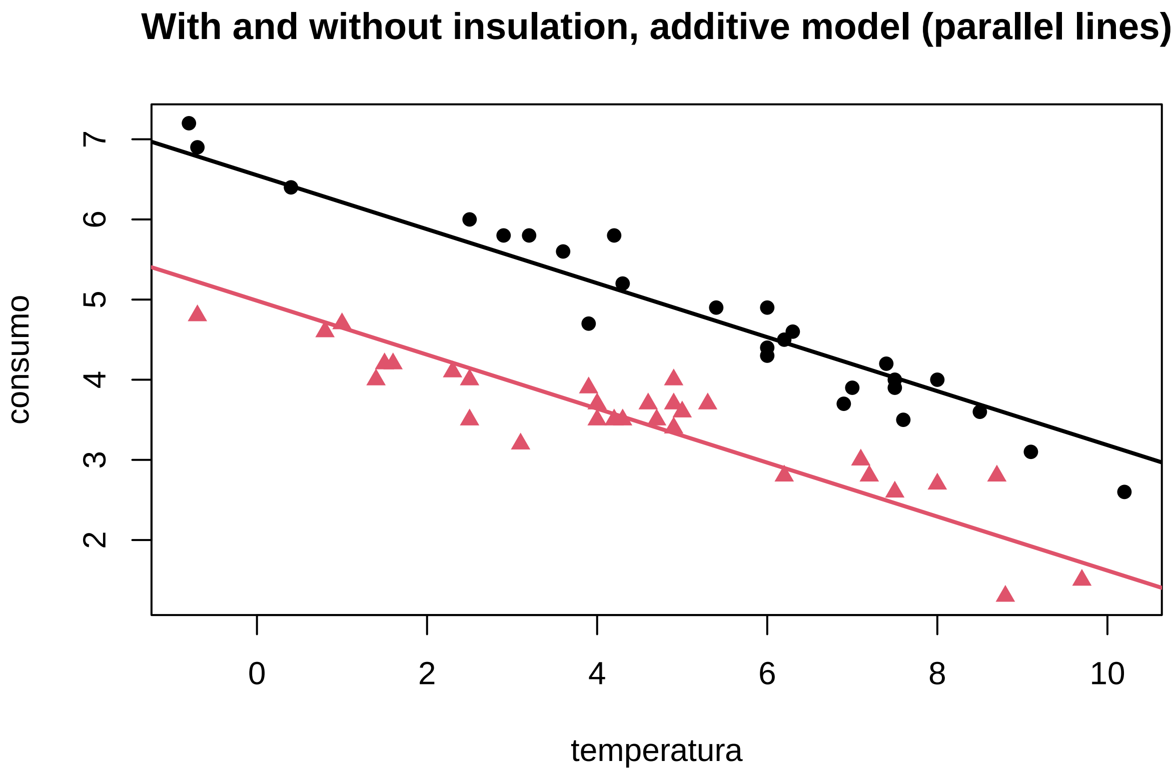

```
hist(res)
```



Redo it all together.

```
plot(temp,cons,type="n", xlab="temperatura", ylab="consumo")
points(temp[quando=="prima"],cons[quando=="prima"],pch=16, col=1)
points(temp[quando=="dopo"], cons[quando=="dopo"], pch=17, col=2)
title("With and without insulation, additive model (parallel lines)")
# Draw the two separate fitted regression lines
```

```
abline(a=regr$coef[1]+regr$coef[2], b=regr$coef[3], lwd=2)
abline(a=regr$coef[1], b=regr$coef[3], col=2, lwd=2)
```



We now fit a model with interaction between the quantitative and the binary predictor, i.e. a non-additive model.

```
regr2=lm(cons~quando+temp+quando*temp)  ### notice the use of *
summary(regr2)
```

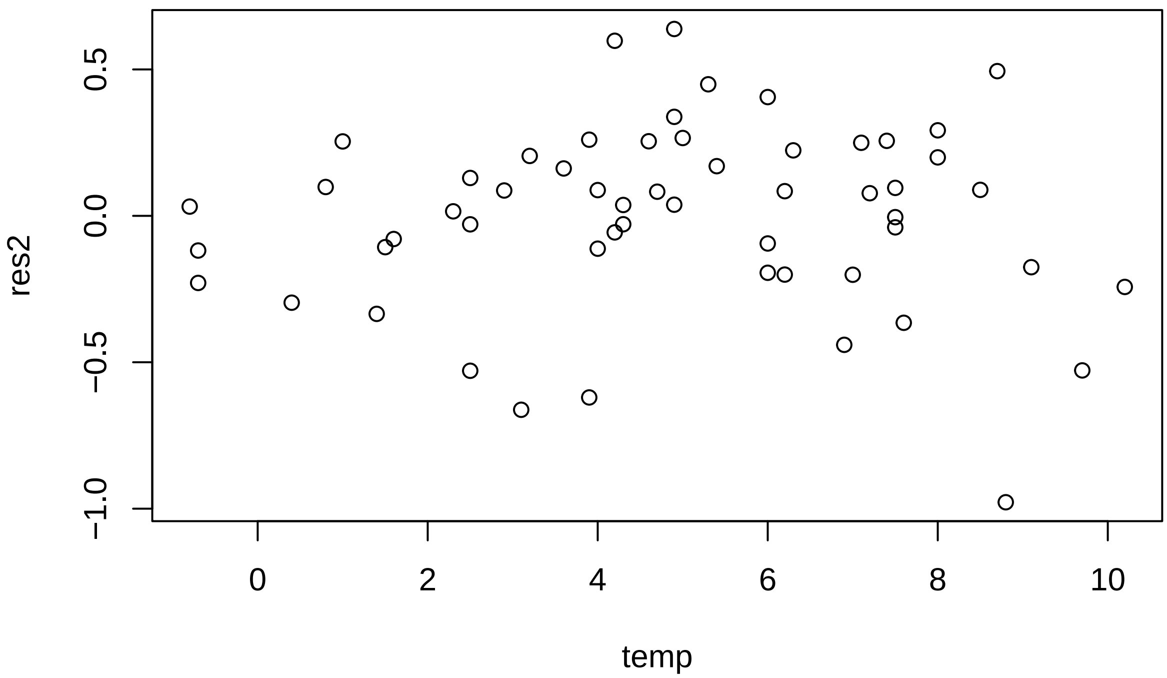
```
##
## Call:
## lm(formula = cons ~ quando + temp + quando * temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97802 -0.18011  0.03757  0.20930  0.63803
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.72385    0.11810  40.000  < 2e-16 ***
## quandoprima       2.12998    0.18009  11.827 2.32e-16 ***
## temp             -0.27793    0.02292 -12.124  < 2e-16 ***
## quandoprima:temp -0.11530    0.03211  -3.591 0.000731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.323 on 52 degrees of freedom
## Multiple R-squared:  0.9277, Adjusted R-squared:  0.9235
## F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16
```

```r
confint(regr2, level=0.95)
```

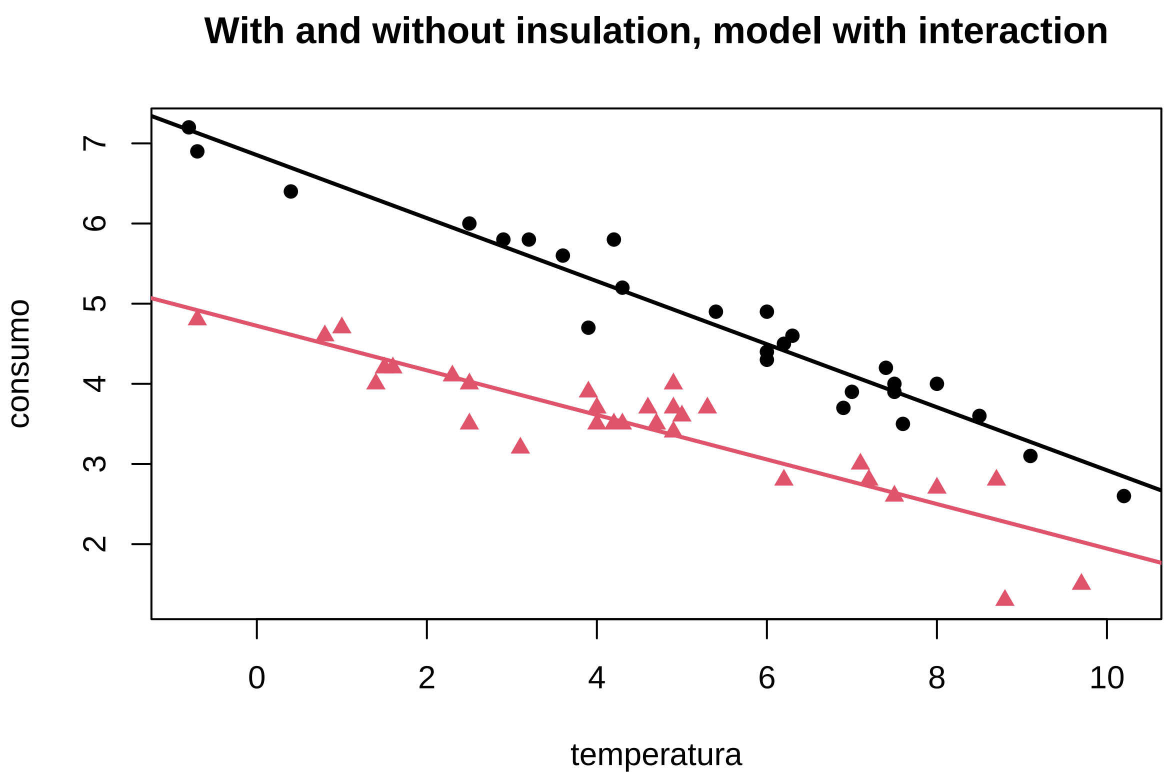
```
##                        2.5 %      97.5 %
## (Intercept)        4.4868714  4.96082799
## quandoprima        1.7685976  2.49135850
## temp              -0.3239359 -0.23193405
## quandoprima:temp -0.1797416 -0.05086618
```

Do a similar analysis for the model with interaction

```r
res2=regr2$resid
plot(temp,res2) # residuals are still ok
```




```r
plot(temp,cons,type="n", xlab="temperatura", ylab="consumo")
points(temp[quando=="prima"],cons[quando=="prima"],pch=16, col=1)
points(temp[quando=="dopo"], cons[quando=="dopo"], pch=17, col=2)
title("With and without insulation, model with interaction")
abline(a=regr2$coef[1]+regr2$coef[2], b=regr2$coef[3]+regr2$coef[4], lwd=2)
abline(a=regr2$coef[1], b=regr2$coef[3], col=2, lwd=2)
```



Several F tests

```
nullo <- lm(cons~1)
solotemp <- lm(cons~temp)
additivo <- lm(cons~temp+quando)
interattivo <- lm(cons~temp*quando)

# null vs interattivo
anova(nullo, interattivo)
```

```
## Analysis of Variance Table
##
## Model 1: cons ~ 1
## Model 2: cons ~ temp * quando
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     55 75.014
## 2     52  5.425  3    69.589 222.33 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(interattivo) # same as all-or-nothing test in summary(interattivo)
```

```
##
## Call:
## lm(formula = cons ~ temp * quando)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97802 -0.18011  0.03757  0.20930  0.63803
```

```
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.72385    0.11810  40.000  < 2e-16 ***
## temp             -0.27793    0.02292 -12.124  < 2e-16 ***
## quandoprima       2.12998    0.18009  11.827 2.32e-16 ***
## temp:quandoprima -0.11530    0.03211  -3.591 0.000731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.323 on 52 degrees of freedom
## Multiple R-squared:  0.9277, Adjusted R-squared:  0.9235
## F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16
```

```
# additivo vs interattivo
anova(additivo, interattivo)
```

```
## Analysis of Variance Table
##
## Model 1: cons ~ temp + quando
## Model 2: cons ~ temp * quando
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     53 6.7704
## 2     52 5.4252  1    1.3451 12.893 0.0007307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(interattivo)# same as interaction test in default view
```

```
##
## Call:
## lm(formula = cons ~ temp * quando)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97802 -0.18011  0.03757  0.20930  0.63803
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.72385    0.11810  40.000  < 2e-16 ***
## temp             -0.27793    0.02292 -12.124  < 2e-16 ***
## quandoprima       2.12998    0.18009  11.827 2.32e-16 ***
## temp:quandoprima -0.11530    0.03211  -3.591 0.000731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.323 on 52 degrees of freedom
## Multiple R-squared:  0.9277, Adjusted R-squared:  0.9235
## F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16
```

```
# solotemp vs interattivo
anova(solotemp, interattivo) # nested models
```

```
## Analysis of Variance Table
##
## Model 1: cons ~ temp
```

```
## Model 2: cons ~ temp * quando
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     54 39.995
## 2     52  5.425  2     34.57 165.67 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## detach the database when you do not use it any more
detach(insulate)
```

Confidence and Prediction intervals

```
predict.lm(additivo,
           newdata=data.frame(quando=c("prima","dopo"),temp=rep(3.2,2)),
           interval="confidence",
       level=.99)
```

```
##        fit      lwr      upr
## 1 5.473898 5.260625 5.687172
## 2 3.908694 3.724325 4.093063
```

```
predict.lm(additivo,
           newdata=data.frame(quando=c("prima","dopo"),temp=rep(3.2,2)),
           interval="prediction",
       level=.99)
```

```
##        fit      lwr      upr
## 1 5.473898 4.495432 6.452365
## 2 3.908694 2.936118 4.881269
```

Cool plots of confidence and prediction bands

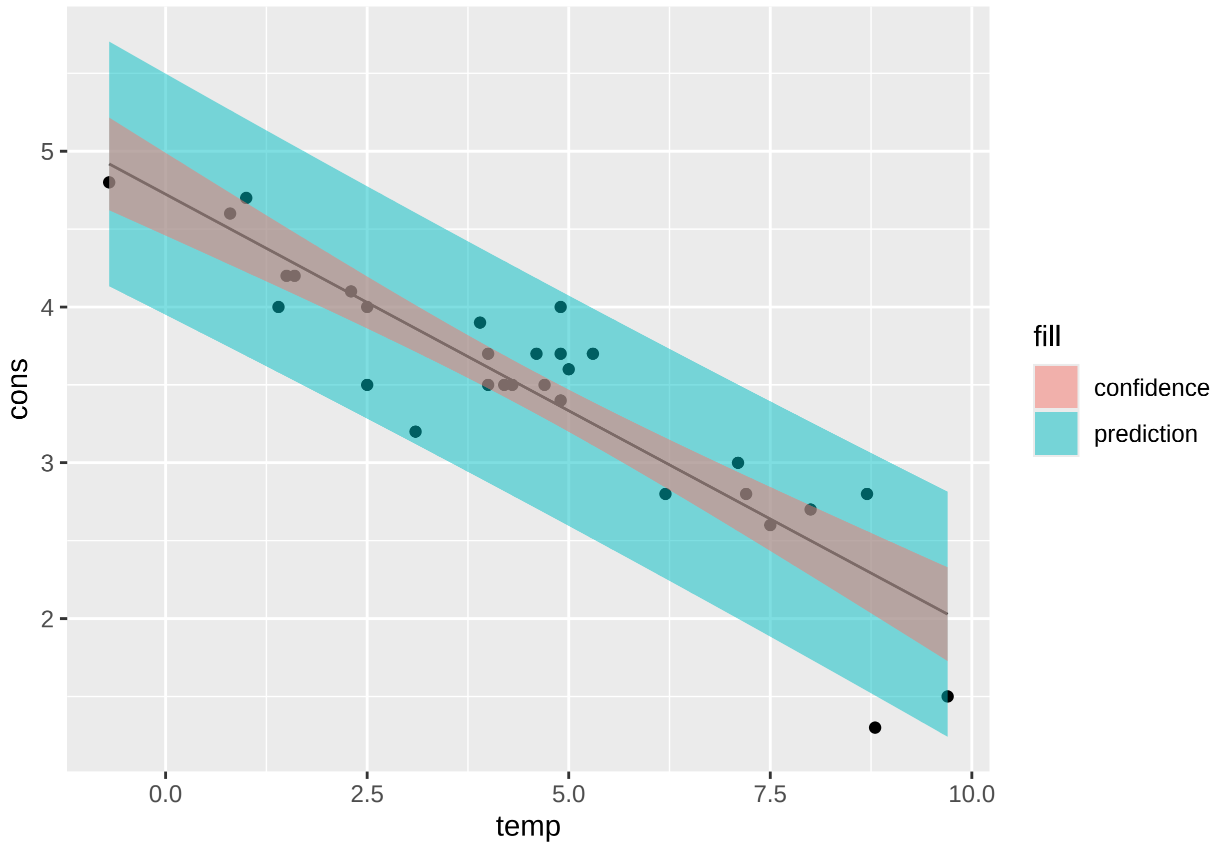
More modern R coding style: using tibbles instead of data frames, using library dplyr for data manipulation and using ggplot for cool plots.

```
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
```

```
library(ggplot2)
# build x grid
dopo_insulate <-
  insulate %>%
  dplyr::filter(quando == "dopo") %>%
  dplyr::select(-quando)
```

```r
# fit a simple model
dopo_simple_lm <- lm(cons ~ temp, data = dopo_insulate)

# build a vector of new data spanning the whole temp support
new_x <- tibble(temp = seq(min(dopo_insulate$temp), max(dopo_insulate$temp), by = 0.05))

# find confidence and prediction intervals for all of the range elements
# and change the column names to make them distinguishable
new_pred <- predict(dopo_simple_lm, newdata = new_x, interval = "prediction") %>%
  as_tibble() %>%
  rename_with(~ paste(.x, "pred", sep = "_")) #
new_conf <- predict(dopo_simple_lm, newdata = new_x, interval = "confidence") %>%
  as_tibble() %>%
  rename_with(~ paste(.x, "conf", sep = "_"))

# join the two interval details
new_data <- bind_cols(new_x, new_pred, new_conf)

ggplot() +
  geom_point(aes(temp, cons), data = dopo_insulate) + # scatter plot
  geom_line(aes(temp, fit_conf), data = new_data) + # regression line
  geom_ribbon(aes(temp, ymin = lwr_pred, ymax = upr_pred, fill = "prediction"),
    data = new_data, alpha = .5
  ) + # pred intervals
  geom_ribbon(aes(temp, ymin = lwr_conf, ymax = upr_conf, fill = "confidence"),
    data = new_data, alpha = .5
  ) # conf intervals
```



cons
5
4
3
2
0.0
2.5
5.0
7.5
10.0
temp
fill
confidence
prediction