HST.953x Workshop 2.08: Linear Regression Exercises

H. David Shea

14 Jul 2021

Contents

(Note: Updated and modified from the hst953-edx github version.)

Description

This document contains the codes and scripts to follow the manuscript and contents for the Section 2.08.

3a: Introduction to Data Analysis Approaches

Import dataset from **PhysioNet** - raw data file downloaded in previous exercise.

```
fnm <- fs::path(base_dir, "exercises/exploratory_data_analysis/aline_full_cohort_data.csv")
dat <- tibble(read.csv(fnm))
rm(fnm)

# Public dataset has NA values for variables required to complete workshop
# Replace NA values with defaults since dataset only intended for teaching
dat <- dat %>%
  mutate(
    gender_num = ifelse(is.na(gender_num), 0L, gender_num),
    sofa_first = ifelse(is.na(sofa_first), 0L, sofa_first)
  )
```

Once it is imported, let's take a look on the variables included in this dataset:

```
names(dat)
#>  [1] "aline_flg"          "icu_los_day"        "hospital_los_day"
#>  [4] "age"                "gender_num"         "weight_first"
#>  [7] "bmi"                "sapsi_first"        "sofa_first"
#> [10] "service_unit"       "service_num"        "day_icu_intime"
```

```
#> [13] "day_icu_intime_num" "hour_icu_intime"    "hosp_exp_flg"
#> [16] "icu_exp_flg"        "day_28_flg"         "mort_day_censored"
#> [19] "censor_flg"         "sepsis_flg"         "chf_flg"
#> [22] "afib_flg"           "renal_flg"          "liver_flg"
#> [25] "copd_flg"           "cad_flg"            "stroke_flg"
#> [28] "mal_flg"            "resp_flg"           "map_1st"
#> [31] "hr_1st"             "temp_1st"           "spo2_1st"
#> [34] "abg_count"          "wbc_first"          "hgb_first"
#> [37] "platelet_first"     "sodium_first"       "potassium_first"
#> [40] "tco2_first"         "chloride_first"     "bun_first"
#> [43] "creatinine_first"   "po2_first"          "pco2_first"
#> [46] "iv_day_1"
```

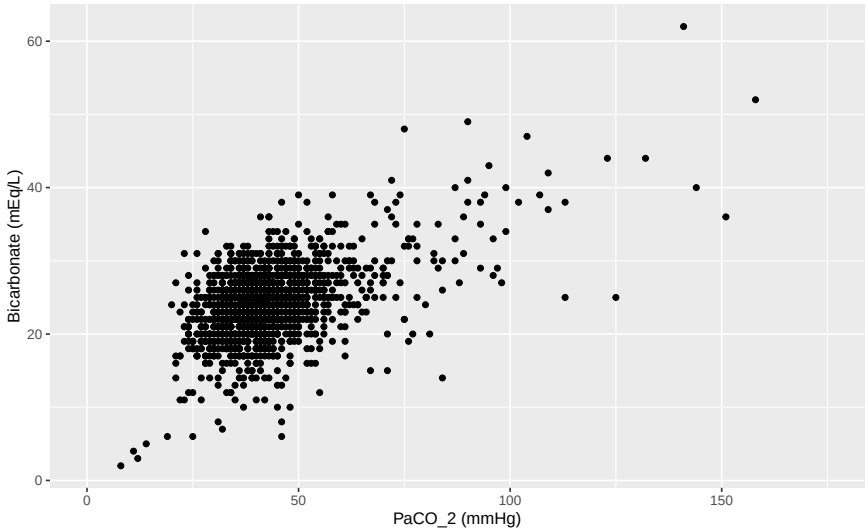
Retrieve the dimension (number of rows and columns) of *dat*:

```
dim(dat)
#> [1] 1776   46
```

3b: Linear Regression

Let's visualize the dataset with a scatter plot:

```
ggplot(dat, aes(pco2_first, tco2_first)) +
        geom_point(na.rm = TRUE) +
        scale_x_continuous(limits = c(0, 175)) +
        labs(x = "PaCO_2 (mmHg)", y = "Bicarbonate (mEq/L)")
```



Let's fit the data to a linear regression model:

```
co2.lm <- lm(tco2_first ~ pco2_first, data = dat)
```

Let's display a summary of this fit:

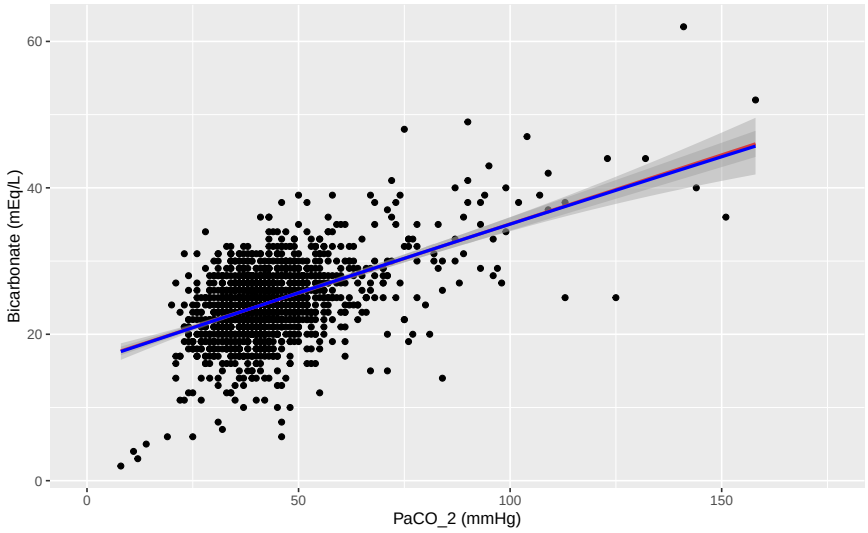
```
summary(co2.lm)
#>
#> Call:
#> lm(formula = tco2_first ~ pco2_first, data = dat)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -18.8852  -2.5080   0.1891   2.8077  19.2005
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 16.210859   0.359676   45.07   <2e-16 ***
#> pco2_first   0.188572   0.007886   23.91   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.395 on 1588 degrees of freedom
#>   (186 observations deleted due to missingness)
#> Multiple R-squared:  0.2647, Adjusted R-squared:  0.2643
#> F-statistic: 571.8 on 1 and 1588 DF,  p-value: < 2.2e-16
```

Fitting a quadratic model:

```
co2.quad.lm <- lm(tco2_first ~ pco2_first + I(pco2_first^2), data = dat)
summary(co2.quad.lm)$coef
#>                       Estimate   Std. Error    t value     Pr(>|t|)
#> (Intercept)     16.0916260327 0.7713394026 20.8619266 1.309513e-85
#> pco2_first       0.1930281243 0.0266927962  7.2314689 7.401248e-13
#> I(pco2_first^2) -0.0000356873 0.0002042135 -0.1747548 8.612946e-01
```

Let's best fit lines to the scatter plots using the **abline** function

```
ggplot(dat, aes(pco2_first, tco2_first)) +
        geom_point(na.rm = TRUE) +
        scale_x_continuous(limits = c(0, 175)) +
        labs(x = "PaCO_2 (mmHg)", y = "Bicarbonate (mEq/L)") +
        geom_smooth(formula = y ~ x, method = "lm", color = "red", na.rm = TRUE) +
        geom_smooth(formula = y ~ x + I(x^2), method = "lm", color = "blue", na.rm = TRUE)
```



The red (linear term only) and blue (linear and quadratic terms) fits are nearly identical. This corresponds with the relatively small coefficient estimate for the I(pco2_first^2) term. The p-value for this coefficient is about 0.86, and at the 0.05 significance level we would likely conclude that a quadratic term is not necessary in our model to fit the data, as the linear term only model fits the data nearly as well.

Statistical Interactions and Testing Nested Models

Check what type of variable is assuming RStudio for **gender_num**.

```
class(dat$gender_num)
#> [1] "integer"
```

Set all (0,1) variables to the correct `factor` class:

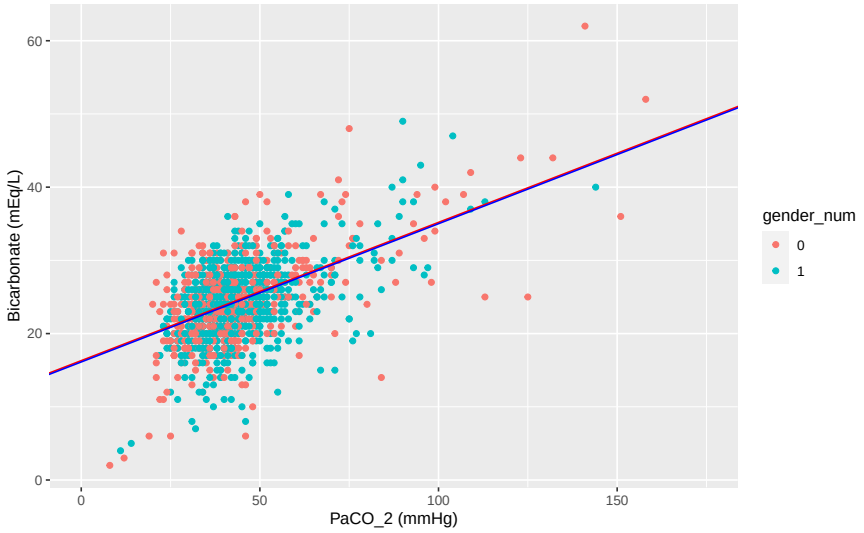
```
is01_factor_column <- function(x) {
  v <- unique(x)
  ((length(v) == 1) & (v[1] %in% c(0,1))) | ((length(v) == 2) & (v[1] %in% c(0,1)) & (v[2] %in% c(0,1)))
}

dat <- dat %>%
  mutate(across(where(is01_factor_column), as.factor))
```

Fit again once it is encoded correctly **gender_num** and retrieve summary:

```
co2.gender.lm <- lm(tco2_first ~ pco2_first + gender_num, data = dat)
summary(co2.gender.lm)$coef
#>                Estimate  Std. Error    t value      Pr(>|t|)
#> (Intercept) 16.3043942 0.377712532 43.1661457 6.337240e-270
#> pco2_first   0.1888542 0.007894741 23.9215128 3.015777e-108
#> gender_num1 -0.1816540 0.223738366 -0.8119036  4.169687e-01
```

Plot fitting lines and compare:

```
ggplot(dat, aes(pco2_first, tco2_first, color = gender_num)) +
        geom_point(na.rm = TRUE) +
        scale_x_continuous(limits = c(0, 175)) +
        labs(x = "PaCO_2 (mmHg)", y = "Bicarbonate (mEq/L)") +
        geom_abline(intercept = coef(co2.gender.lm)[1],
                    slope = coef(co2.gender.lm)[2],
                    color = "red") +
        geom_abline(intercept = coef(co2.gender.lm)[1]+coef(co2.gender.lm)[3],
                    slope = coef(co2.gender.lm)[2],
                    color = "blue")
```

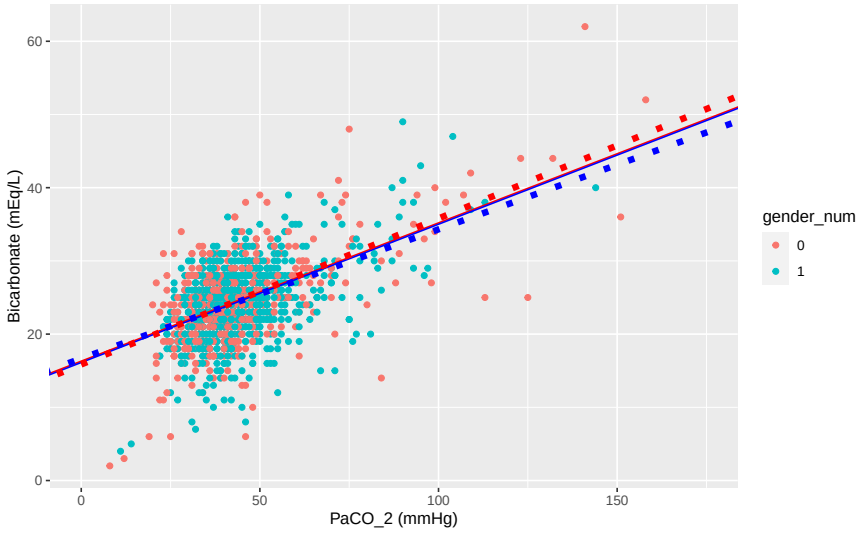


Modeling taking into consideration variables' interactions:

```
co2.gender.inteaction.lm <- lm(tco2_first ~ pco2_first * gender_num, data = dat)
summary(co2.gender.inteaction.lm)$coef
#>                              Estimate Std. Error   t value      Pr(>|t|)
#> (Intercept)             15.85443226 0.48869107 32.442648 1.591490e-177
#> pco2_first               0.19939518 0.01072876 18.585105  6.559901e-70
#> gender_num1              0.81437833 0.72225677  1.127547  2.596819e-01
#> pco2_first:gender_num1 -0.02297002 0.01583758 -1.450348  1.471591e-01
```

```
ggplot(dat, aes(pco2_first, tco2_first, color = gender_num)) +
        geom_point(na.rm = TRUE) +
        scale_x_continuous(limits = c(0, 175)) +
        labs(x = "PaCO_2 (mmHg)", y = "Bicarbonate (mEq/L)") +
        geom_abline(intercept = coef(co2.gender.lm)[1],
                    slope = coef(co2.gender.lm)[2],
                    color = "red") +
        geom_abline(intercept = coef(co2.gender.lm)[1]+coef(co2.gender.lm)[3],
                    slope = coef(co2.gender.lm)[2],
                    color = "blue") +
        geom_abline(intercept = coef(co2.gender.inteaction.lm)[1],
                    slope = coef(co2.gender.inteaction.lm)[2],
                    color = "red", lty=3, lwd = 2) +
```

```
        geom_abline(intercept = coef(co2.gender.inteaction.lm)[1]+coef(co2.gender.inteaction.lm)[3],
                    slope = coef(co2.gender.inteaction.lm)[2]+coef(co2.gender.inteaction.lm)[4],
                    color = "blue", lty=3, lwd = 2)
```



Perform anova analysis:

```
anova(co2.lm, co2.gender.inteaction.lm)
#> Analysis of Variance Table
#>
#> Model 1: tco2_first ~ pco2_first
#> Model 2: tco2_first ~ pco2_first * gender_num
#>   Res.Df   RSS Df Sum of Sq      F Pr(>F)
#> 1   1588 30674
#> 2   1586 30621  2    53.349 1.3816 0.2515
```

We will highlight the reported F-test p-value *Pr(>F)*, which in this case is 0.2515. In nested models, the null hypothesis is that all coefficients in the larger model but not in the smaller model are zero. In the case we are testing, our null hypotheses are $B2$ and $B3 = 0$. Since the p-value exceeds the typically used significance level ($a = 0.05$), we would not reject the null hypothesis and say the smaller model likely explains the data just as well as the larger model.

Before presenting the results, some discussion of how you got the results should be done. It is a good idea to report the following: whether you transformed the outcome or any covariates in any way (e.g., by taking the logarithm), what covariates you considered, and how you chose the covariates which were in the model you reported.

In our example above, we did not transform the outcome (TCO2); we considered PCO2 both as a linear and quadratic term; and we considered gender on its own and as an interaction term with PCO2. We first evaluated whether a quadratic term should be included in the model by using a t-test, after which we considered a model with gender and a gender-PCO2 interaction, and performed model selection with an F-test. Our final model involved only a linear PCO2 term and an intercept.

This model showed that TCO2 increassed 0.19 (SE: 0.008, $p < 0.0001$) units per unit increase of PCO2. The Multiple R-squared for he model was 0.2647.

When reporting your results, it's a good idea to report three aspects for each covariate.

Confidence and Prediction Intervals

Get confidence intervals:

```
confint(co2.lm)
#>                  2.5 %     97.5 %
#> (Intercept) 15.5053693 16.9163494
#> pco2_first   0.1731033  0.2040403
```

Predict the outcome over the range of covariate values we observed determined by the min and max functions:

```
grid.pred <- tibble(pco2_first = seq.int(from = min(dat$pco2_first, na.rm = T),
                                         to = max(dat$pco2_first, na.rm = T)))

preds <- predict(co2.lm, newdata = grid.pred, interval = "prediction")

grid.pred$lwr <- preds[,2]
grid.pred$upr <- preds[,3]
```

```
ggplot(dat, aes(pco2_first, tco2_first)) +
        geom_point(na.rm = TRUE) +
        scale_x_continuous(limits = c(0, 175)) +
        labs(x = "PaCO_2 (mmHg)", y = "Bicarbonate (mEq/L)") +
        geom_smooth(formula = y ~ x, method = "lm", color = "red", lwd = 2, na.rm = TRUE) +
        geom_line(aes(x = pco2_first, y = lwr), data = grid.pred, lty = 3) +
        geom_line(aes(x = pco2_first, y = upr), data = grid.pred, lty = 3)
```