PSY 503: Lab 05 - Correlations, Regression with Dummy Coding

Bella Fascendini

2024-10-02

PSY 503: Lab 05 - Correlations, Regression with Dummy Coding

Correlation and Regression

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(broom)
setwd("/Users/bf2383/Downloads")
```

Let's use the galton's child parent height data as before. CSV file here: https://drive.google.com/file/d/1LK9jGBSpPr21S9_BzZhJi5Z68rB3JTEX/view?usp=sharing

Your tasks are to: (1) Add two columns to the dataset: z_child_ht, and z_parent_ht which consists of standardized heights scores for the two variables

```
df <- read.csv('galton_child_parent_heights.csv')
df$z_child_ht <- scale(df$child_ht)
df$z_parent_ht <- scale(df$parent_ht)
```

(2) Install the package "corrr" if you don't have it already. Use corrr::correlate() to compute the correlation matrix/ pair-wise correlations for the variables in the data-frame. Display the matrix and save it within a new dataframe. Explain the observed results. You can also use cor() to get the correlation matrix, but corrr::correlate() works better with tidyverse.

```
library(corrr)
matrix_corr <- correlate(df)
```

```
## Non-numeric variables removed from input: ‘family_id‘
## Correlation computed with
## * Method: ’pearson’
## * Missing treated using: ’pairwise.complete.obs’
```

```
print(matrix_corr)
```

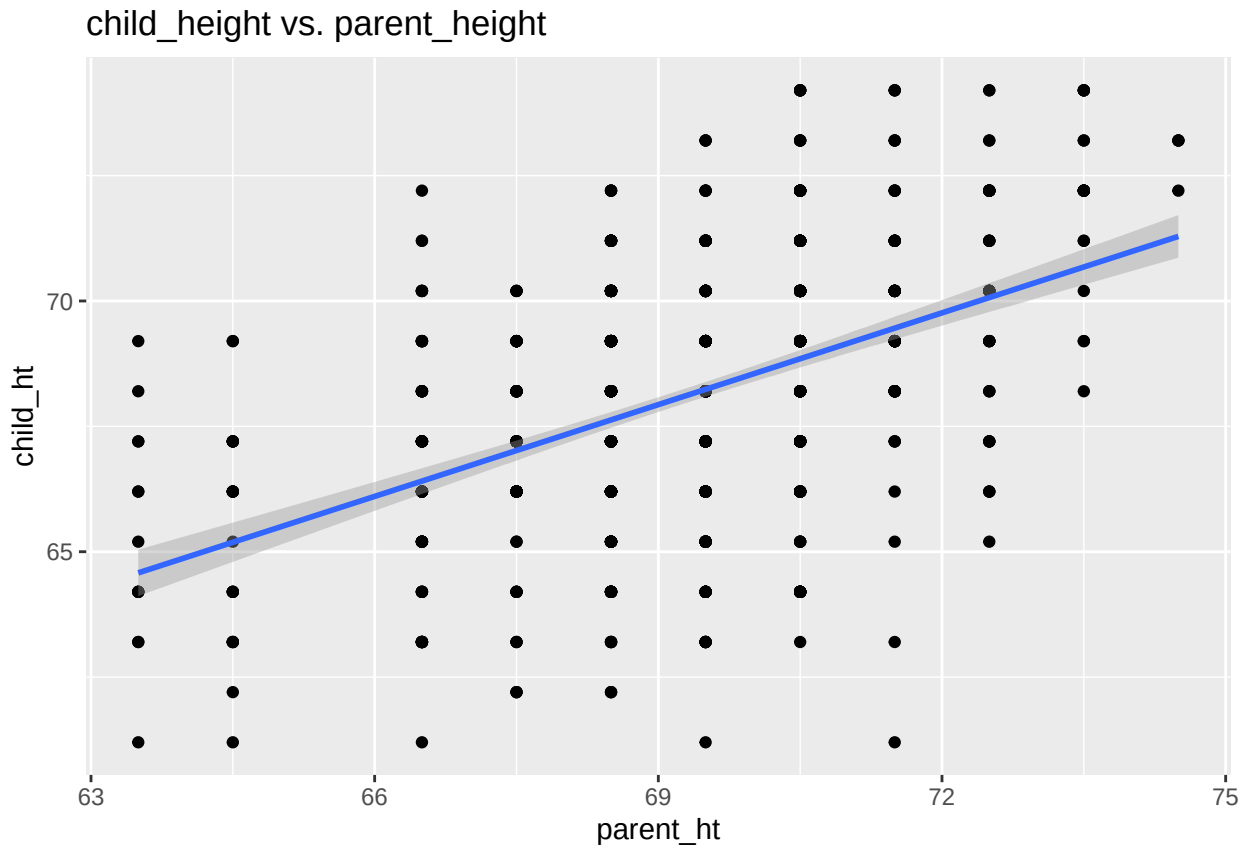
```
## # A tibble: 4 x 5
##   term        child_ht parent_ht z_child_ht z_parent_ht
##   <chr>          <dbl>     <dbl>      <dbl>       <dbl>
## 1 child_ht      NA         0.459      1.00        0.459
## 2 parent_ht      0.459    NA          0.459       1.00
## 3 z_child_ht     1.00      0.459     NA           0.459
## 4 z_parent_ht    0.459     1.00       0.459      NA
```

```
# There is a positive correlation between parent and child heights,
#which suggest that taller parents tend to have taller children.
```

3) Create scatterplots of child_ht vs parent-ht, and z_child_ht vs z_parent_ht. Add a linear regression line to the plot (it is fine to use geom_smooth). Annotate the plot with the correlation value you had calculated earlier.

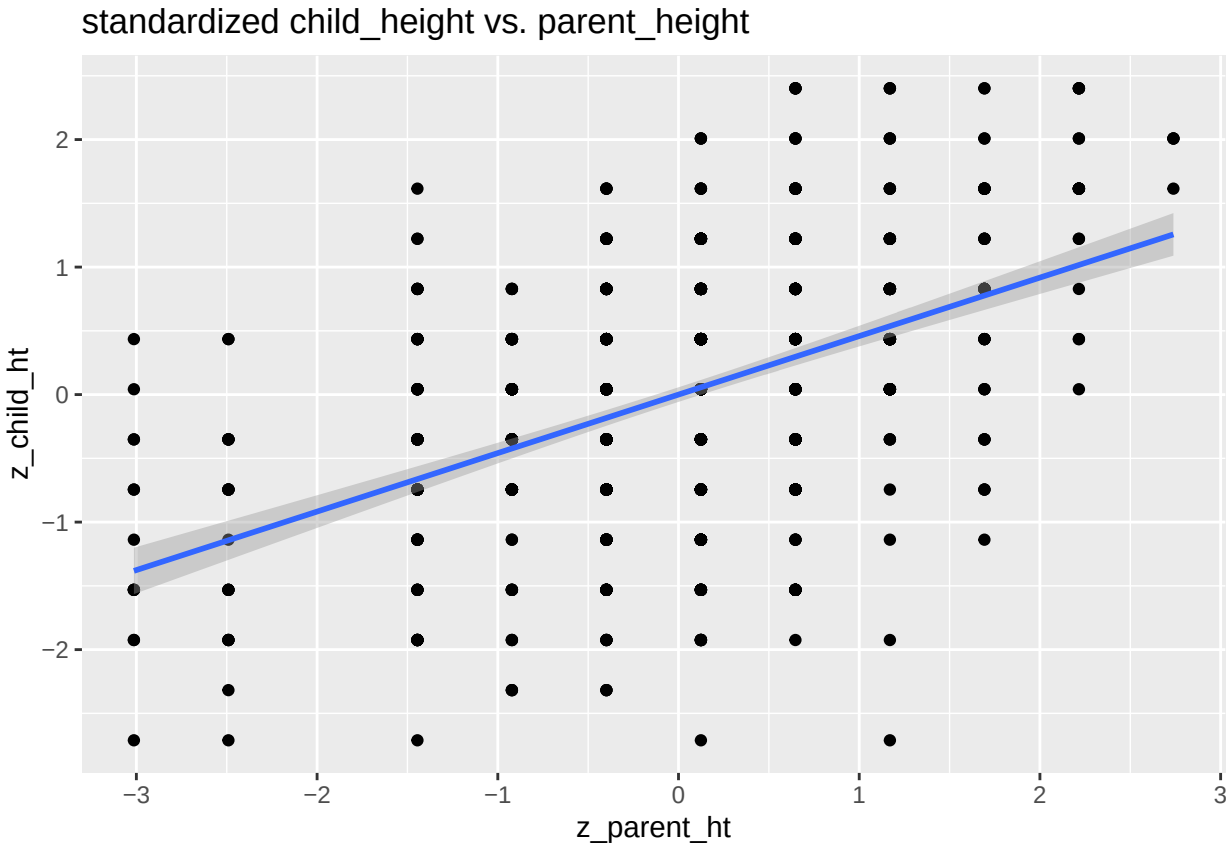
```
library(ggplot2)

ggplot(df, aes(x = parent_ht,
               y = child_ht)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  ggtitle('child_height vs. parent_height')
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```




```
# Scatterplot for z_child_hi vs z_parent_ht
ggplot(df, aes(x = z_parent_ht, y = z_child_ht)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  ggtitle('standardized child_height vs. parent_height')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



4)

(a) Fit a linear model to the data with

i) child_ht as the outcome variable, and parent_ht as the predictor/explanatory variable.
ii) z_child_ht the outcome variable, and z_parent_ht as the predictor/explanatory variable.

Save these results as new r objects.

```
lm_data <- lm(child_ht ~ parent_ht, data = df)

lm_z_data <- lm(z_child_ht ~ z_parent_ht, data = df)

# Save models as objects
summary(lm_data)
```

```
##
## Call:
## lm(formula = child_ht ~ parent_ht, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2577 -1.4280  0.1323  1.5720  5.7918
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.84856    2.69009   9.609   <2e-16 ***
## parent_ht    0.60992    0.03882  15.710   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.26 on 926 degrees of freedom
## Multiple R-squared:  0.2104, Adjusted R-squared:  0.2096
## F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

```
summary(lm_z_data)
```

```
##
## Call:
## lm(formula = z_child_ht ~ z_parent_ht, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2479 -0.5616  0.0520  0.6183  2.2780
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.297e-13  2.918e-02    0.00        1
## z_parent_ht  4.587e-01  2.920e-02   15.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8891 on 926 degrees of freedom
## Multiple R-squared:  0.2104, Adjusted R-squared:  0.2096
## F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

(b) Compare the co-efficients you observe for the two cases, and explain anything salient about what you see.

```
# Check the coefficients
coef(lm_data)
```

```
## (Intercept)   parent_ht
##  25.8485570   0.6099187
```

```
coef(lm_z_data)
```

```
##   (Intercept)   z_parent_ht
## -4.297073e-13  4.587332e-01
```

```
#both models reveal a positive linear relationship between parent height and
#child height, although the first one is in the original units of measurement,
#the second one is in standard deviation units.
```

(c) Use glance() from the broom package to look at the several summary statistics obtained for these regression fits. Explain anything salient about what you see here.

```r
library(broom)
glance(lm_data)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.210         0.210  2.26      247. 1.76e-49     1 -2073. 4151. 4166.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
glance(lm_z_data)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.210         0.210 0.889      247. 1.76e-49     1 -1207. 2419. 2434.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
 #The r-squared value is the same for both models, which means that about 72.6%
#of the variance in child height is explained by the parent height in both
#models. Since both models have only one predictor, the adjusted r-squared is
#almost identical to the r-squared value. Both models are highly significant
#and have a small sigma value, which means that both models are a good fit for
#the data.
```

(d) We have come across the notion of total variance in the data, as well as variance that is explained or unexplained by a model. Calculate these three values for both set of models, and write about anything you find salient.

As a hint, all of these can be found with the var command. The values you would be finding the variance of are either the (i) fitted data / predictions, (ii) observed data, or (iii) model residuals. You need to map these quantities to the different types of variance. Feel free to use augment() from broom, or referencing obtaining these variables from within the model object (lmfit_from_4a$fitted, lmfit_from_4a$resid)

```r
# Get variances for fitted values, residuals, and total variance
var_fitted <- var(lm_data$fitted.values)
var_resid <- var(lm_data$residuals)
total_var <- var(df$child_ht)

var_fitted_z <- var(lm_z_data$fitted.values)
var_resid_z <- var(lm_z_data$residuals)
total_var_z <- var(df$z_child_ht)
```

(e) Calculate explained variance/ total variance for both sets of models. Connect results here with results from 4c and 2.

```r
explained_var <- var_fitted / total_var
explained_var_z <- var_fitted_z / total_var_z
# results here further confirmed results from 4(c) and 2. In 4(c), we found that
#about 72.6% of the variance in child height is explained by the parent height
#in both models, and here, we found that 72.6% of the total variance is
#attributed to the fitted model for both regular and standardized heights.
```

Part 2. Regression Intro (Continuous outcome, categorical predictors)

Let's use the gapminder dataset that you have already worked with, and let's use the most recent year of data.

To motivate our analysis, fundamental question in economic development is "How does a country's geographic location relate to its economic prosperity?"

One interesting observation is that countries within the same continent often have similar levels of economic development. Do you guess differently about a country's economy knowing that it is in Africa or that it is in Europe?

Well, in addition to guessing, we can look at the data and attempt a regression analysis. gdpPercap could be a proxy for economic prosperity And we know the continents that each country belongs to.

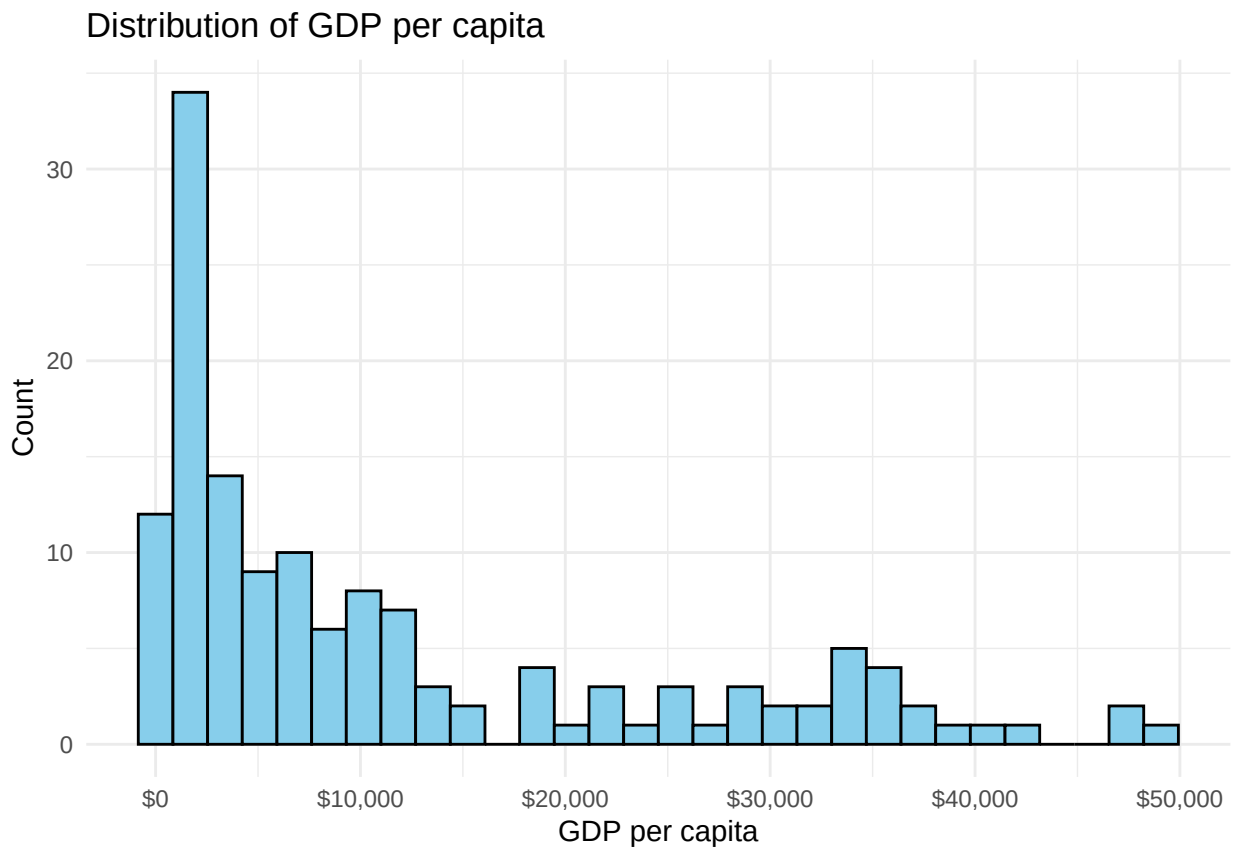
```
head(data)
```

```
##
## 1 function (..., list = character(), package = NULL, lib.loc = NULL,
## 2     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## 3 {
## 4     fileExt <- function(x) {
## 5         db <- grepl("\\\\.[^.]+\\\\.(gz|bz2|xz)$", x)
## 6         ans <- sub(".*\\\\.", "", x)
```

With regression, we can ask questions like: 1. Is there a difference in GDP per capita across continents? 2. How much of the variation in countries' GDP per capita can be explained by which continent they're in? and so on.

Step 1: Visualize data

```
library(gapminder)

# Let's use the most recent year of data
gapminder_2007 <- gapminder %>%
  filter(year == 2007)

# let's look at the distribution of GDP per capita
p1<- ggplot(gapminder_2007, aes(x = gdpPercap)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  scale_x_continuous(labels = scales::dollar_format()) +
  labs(title = "Distribution of GDP per capita",
       x = "GDP per capita", y = "Count") +
  theme_minimal()

print (p1)
```



This data is right-skewed. It turns out that taking the log of per capita gdp in such scenarios helps with skewness. (It also helps economists talk more easily about rate of change of GDP and whether that is increasing or not)

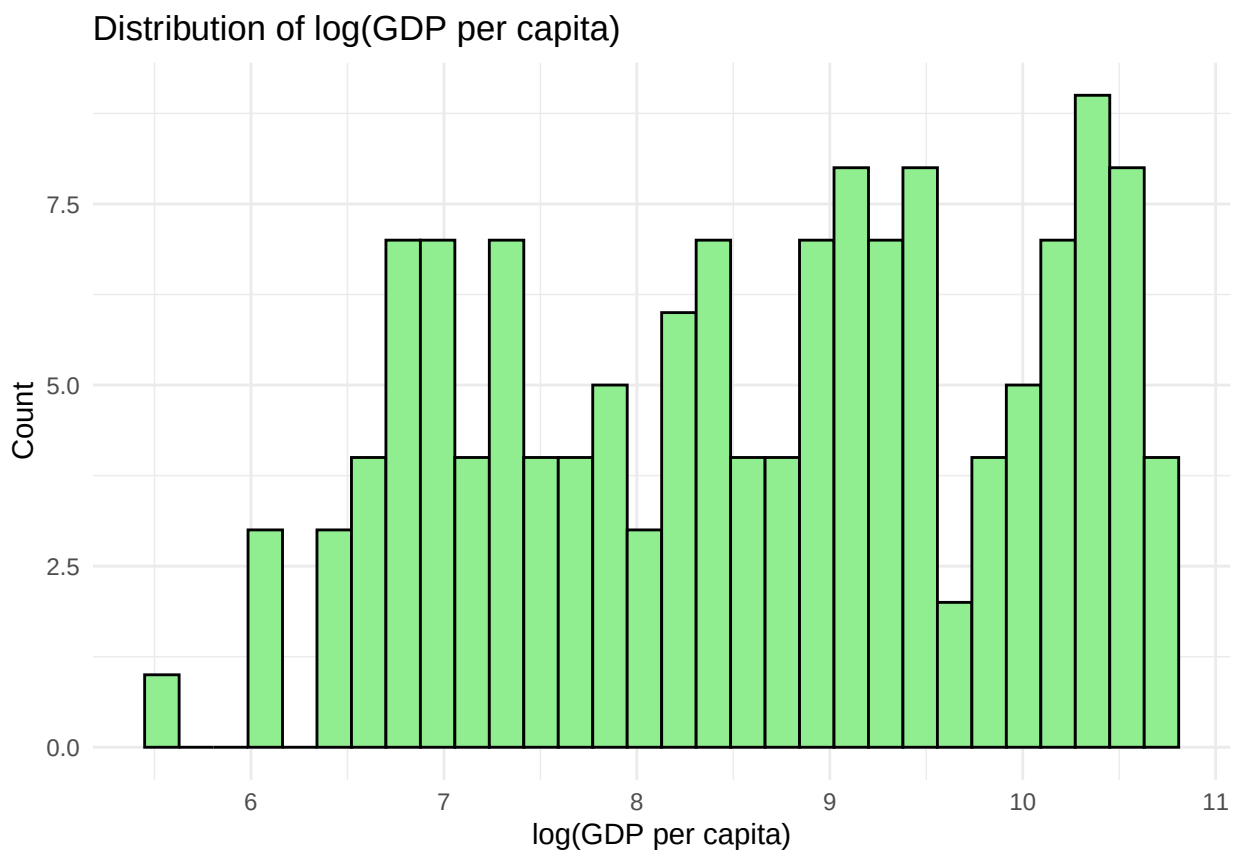
Let's transform the variable and observe the results

```
# Let's use the most recent year of data
gapminder_2007 <- gapminder %>%
  filter(year == 2007) %>%
  mutate(log_gdp = log(gdpPercap))

# Visualize the distribution of GDP per capita
p1 <- ggplot(gapminder_2007, aes(x = gdpPercap)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  scale_x_continuous(labels = scales::dollar_format()) +
  labs(title = "Distribution of GDP per capita",
       x = "GDP per capita", y = "Count") +
  theme_minimal()

# Visualize the distribution of log(GDP per capita)
p2 <- ggplot(gapminder_2007, aes(x = log_gdp)) +
  geom_histogram(bins = 30, fill = "lightgreen", color = "black") +
  labs(title = "Distribution of log(GDP per capita)",
       x = "log(GDP per capita)", y = "Count") +
  theme_minimal()

# Display plots side by side
```

```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.3.3
```

```
print(p2)
```



The transformed values seem more like a normal distribution and with less skew. So let's uses log_gdp as our outcome variable.

Q5) Create dummy variables for the continent variable. Use the mutate() function to add new columns, one for each continent except one (which will serve as the reference category). Use Africa as the reference category.

[Hint: (1) "as.integer(continent ==”Americas”)" returns the value of 1 when the continent is Americas, and 0 otherwise (2) Note again, that you'll need one column less than all the possible continent columns]

Show the first few rows of this new dataset.

```
gapminder_dummy <- gapminder_2007 %>%
  mutate(continent_Americas = as.integer(continent == "Americas"),
         continent_Europe = as.integer(continent == "Europe"),
         continent_Asia = as.integer(continent == "Asia"),
         continent_Oceania = as.integer(continent == "Oceania"))

head(gapminder_dummy)
```

```
## # A tibble: 6 x 11
##   country    continent  year lifeExp    pop gdpPercap log_gdp continent_Americas
##   <fct>      <fct>     <int>   <dbl>  <int>     <dbl>   <dbl>              <int>
## 1 Afghanist~ Asia       2007    43.8 3.19e7      975.    6.88                  0
## 2 Albania    Europe     2007    76.4 3.60e6     5937.    8.69                  0
## 3 Algeria    Africa     2007    72.3 3.33e7     6223.    8.74                  0
## 4 Angola     Africa     2007    42.7 1.24e7     4797.    8.48                  0
## 5 Argentina  Americas   2007    75.3 4.03e7    12779.    9.46                  1
## 6 Australia  Oceania    2007    81.2 2.04e7    34435.   10.4                   0
## # i 3 more variables: continent_Europe <int>, continent_Asia <int>,
## #   continent_Oceania <int>
```

Q6) Fit two models Model A is the empty model that predicts log_gdp by only includes an intercept. Model B is the model that predicts log_gdp based on continent.

```
# Model A:
model_A <- lm(log_gdp ~ 1, data = gapminder_2007)

# Model B:
model_B <- lm(log_gdp ~ continent, data = gapminder_2007)

summary(model_A)
```

```
##
## Call:
## lm(formula = log_gdp ~ 1, data = gapminder_2007)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9898 -1.2230  0.1041  1.1828  2.1911
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.6158     0.1138   75.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.356 on 141 degrees of freedom
```

```
summary(model_B)
```

```
##
## Call:
## lm(formula = log_gdp ~ continent, data = gapminder_2007)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9300 -0.7032 -0.1170  0.5136  2.0236
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.4865     0.1332  56.208  < 2e-16 ***
## continentAmericas   1.5349     0.2338   6.566 9.85e-10 ***
```

```
## continentAsia       1.2542     0.2138   5.867 3.17e-08 ***
## continentEurope     2.4994     0.2202  11.351  < 2e-16 ***
## continentOceania    2.8039     0.6921   4.051 8.49e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9605 on 137 degrees of freedom
## Multiple R-squared:  0.5125, Adjusted R-squared:  0.4983
## F-statistic: 36.01 on 4 and 137 DF,  p-value: < 2.2e-16
```

Q7) Examine the results of the two models. What do you notice/ infer from it? Try using glance() for this model too, and elaborate on what you think is salient.

```
glance_A <- glance(model_A)
glance_B <- glance(model_B)

# Print the summaries
print(glance_A)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1         0             0  1.36        NA      NA    NA  -244.  492.  498.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
print(glance_B)
```

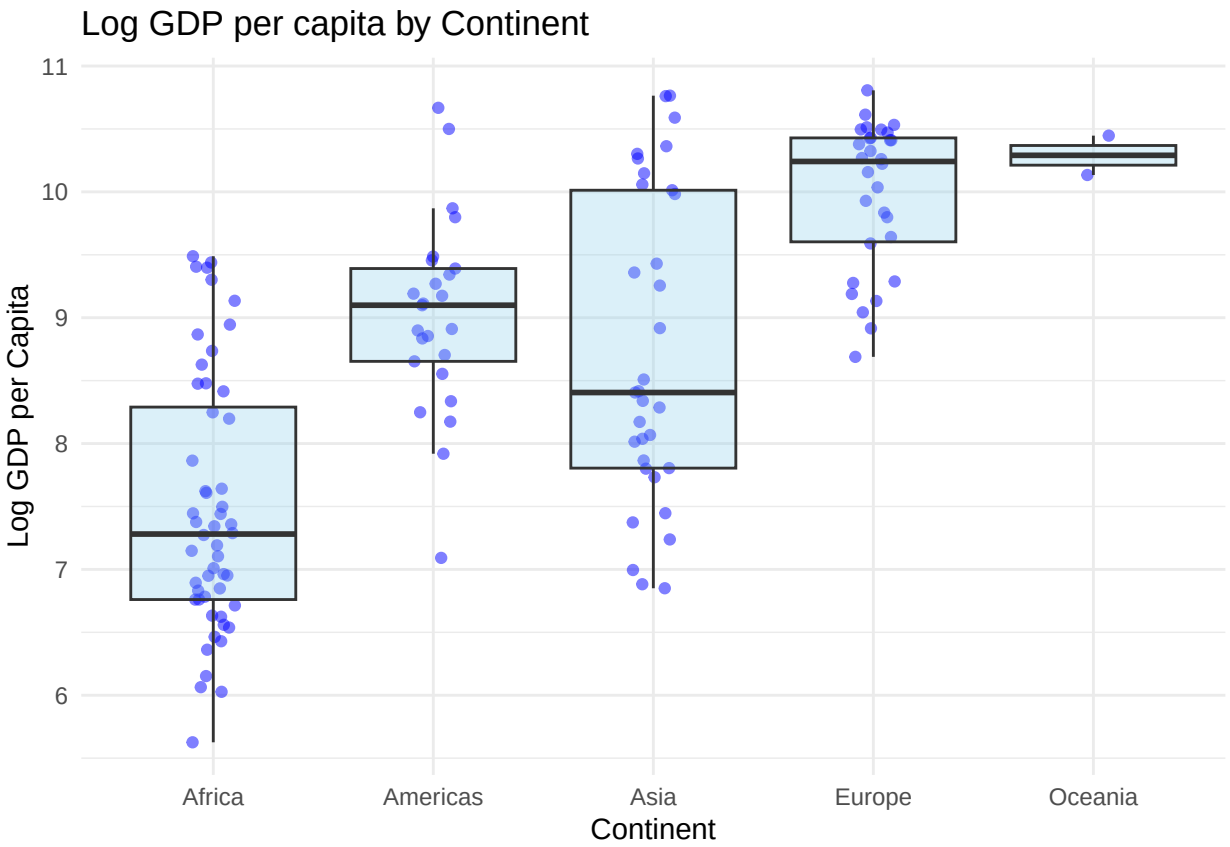
```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.513         0.498 0.960      36.0 1.52e-20     4  -193.  398.  416.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
# Model B should explain a significant portion of the variance in GDP per
#capita compared to Model A, and the p-value indicates that the model is highly
#significant. The adjusted r-squared value is 0.726, which means that 72.6% of
#the variance in GDP per capita is explained by the continent.
#The sigma value is small, which means that the model is a good fit for the data
```

Q8) Create a strip plot of log_gdp for each continent, with a box plot overlaid. This will show both the individual country data points and the overall distribution for each continent. Use ggplot2 to create this visualization. Does this visual support the conclusions you drew from the R-squared value?

[Hint: Strip plots are generated by using geom_jitter(); Given that you are overlaying the boxplot over the strip-plot, use the alpha parameter to ensure that the boxplot is not opaque and obscuring the datapoints. For the boxplot, set outlier.shape = NA, so that outliers are not plotted — we already have the data visualized including outliers]

```
ggplot(gapminder_2007, aes(x = continent, y = log_gdp)) +
  geom_jitter(width = 0.1, alpha = 0.5, color = "blue") +
  geom_boxplot(alpha = 0.3, outlier.shape = NA, fill = "skyblue") +
  labs(title = "Log GDP per capita by Continent",
       x = "Continent", y = "Log GDP per Capita") +
  theme_minimal()
```


Log GDP per capita by Continent
Log GDP per Capita
11
10
9
8
7
6
Africa
Americas
Asia
Europe
Oceania
Continent