A Compendium Of Statistics Questions

Richard White

2017-06-07

Contents

Chapter 1

Purpose

This is a compendium of commonly asked statistical questions.

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., .

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2016) in this sample book, which was built on top of R Markdown and **knitr**

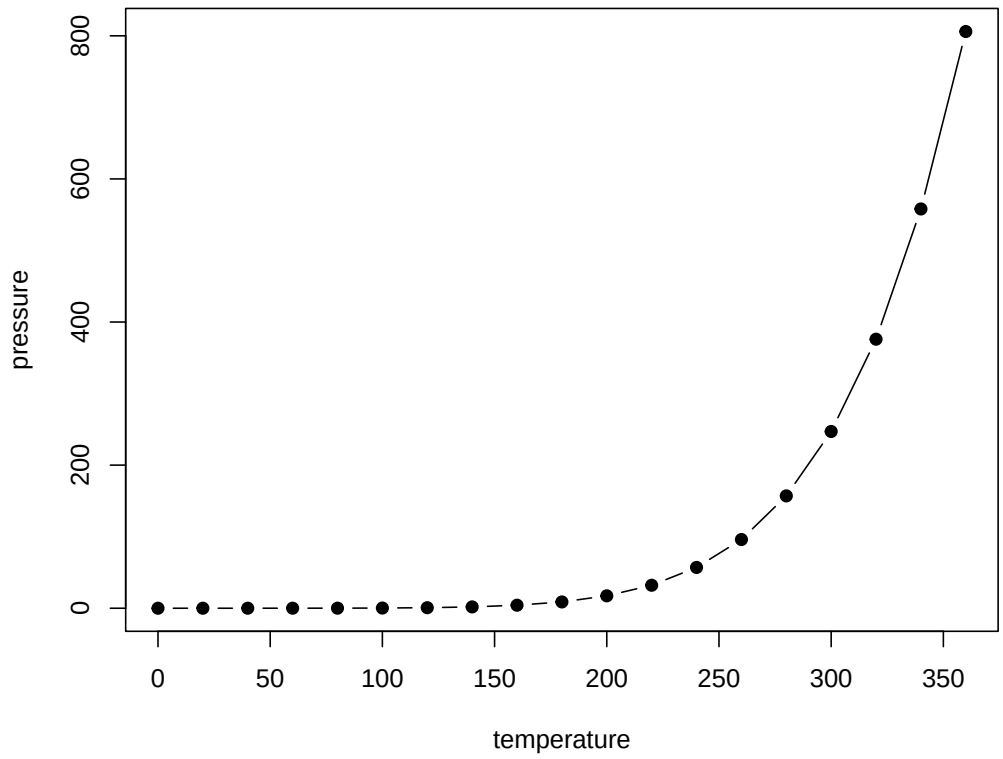

Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Folder structure

Here is a review of existing methods.

Chapter 4

Docker

4.1 What is Docker, and why is it good?

http://blog.kaggle.com/2016/02/05/how-to-get-started-with-data-science-in-containers/

4.2 What is REPL?

Read-eval-print loop. Basically, the user types (reads) stuff into an interactive terminal, the script is evaluated, and results printed. This loops over and over, until the script is finished.

Of course, if you type directly into the interactive terminal, your scripts are lost to eternity. Thus it is better to type your scripts into a text file and have them automatically copied into the interactive terminal. The most well-known example of this is RStudio.

4.3 What is VIM?

"Vim is a highly configurable text editor built to make creating and changing any kind of text very efficient." <www.vim.org>

4.4 What is vim-slime?

https://github.com/jpalardy/vim-slime

4.5 Which Docker containers should I use?

https://github.com/rocker-org

https://github.com/Kaggle/docker-python

4.6 Putting it all together

https://github.com/raubreywhite/docker

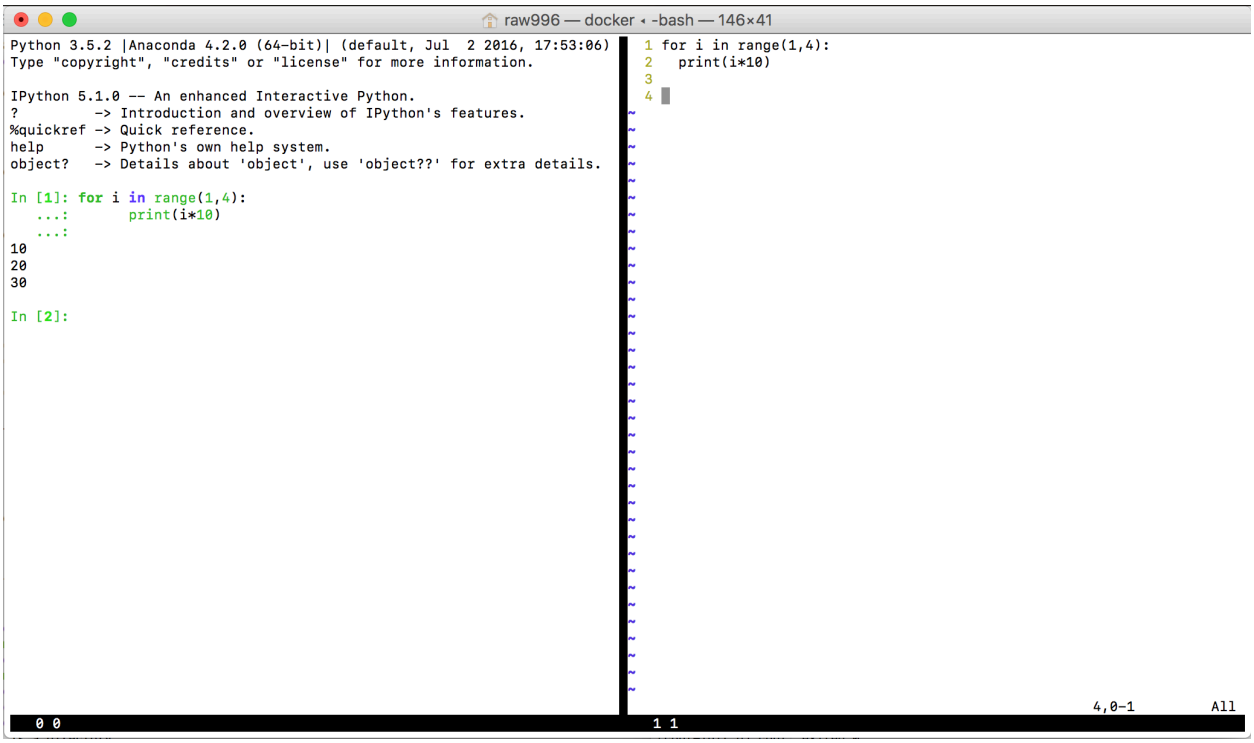

Figure 4.1: img

Chapter 5

Linear Regression vs ANOVA

5.1 Summary

Many ANOVA computations can be performed using linear regression models, with nested/hierarchical problems requiring mixed effects regression models (Gelman, 2005).

5.2 Empirical evidence

First we create some data, shown in Figure 5.1

```
library(ggplot2)

set.seed(4)
x <- rep(0:2,100)
y <- (x+1)/7 + rnorm(length(x))
data <- data.frame(y=y,x=x)

q <- ggplot(data,aes(x=x,y=y,group=x))
q <- q + geom_boxplot()
print(q)
```

Then we establish the linear regression model

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \epsilon_i$$

where $i = 1, \ldots n$ and $\epsilon_i \sim N(0, \sigma^2)$

```
fit <- lm(y ~ factor(x), data)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ factor(x), data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48573 -0.70825 -0.05413  0.67424  2.59549
##
## Coefficients:
```

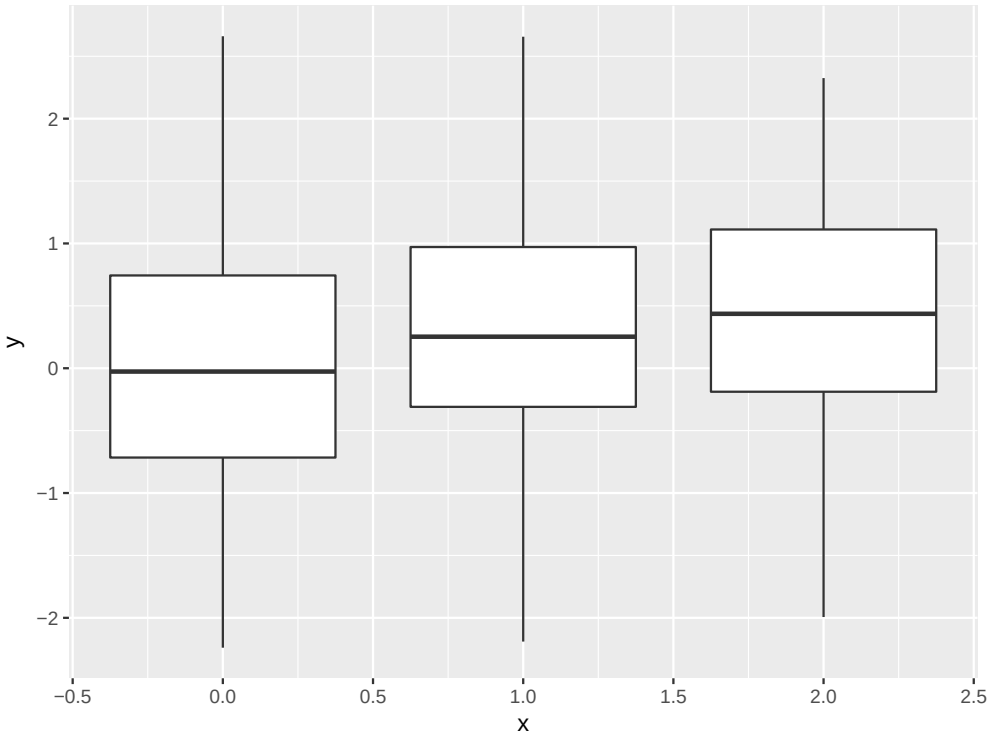

Figure 5.1: Three diffferent groups (x=0, 1, 2)

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06488    0.09662   0.672  0.50239
## factor(x)1   0.23052    0.13664   1.687  0.09264 .
## factor(x)2   0.40818    0.13664   2.987  0.00305 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9662 on 297 degrees of freedom
## Multiple R-squared:  0.02933,    Adjusted R-squared:  0.02279
## F-statistic: 4.487 on 2 and 297 DF,  p-value: 0.01203
```

and we can see that the F-test corresponding to

$$H_0 : \beta_1 = \beta_2 = 0$$

$$H_1 : \text{Not } H_0$$

has a p-value of 0.01203.

We then perform a one-way ANOVA assuming equal variances, with:

$$H_0 : \bar{y}_{\text{x}=0} = \bar{y}_{\text{x}=1} = \bar{y}_{\text{x}=2}$$

$$H_1 : \text{Not } H_0$$

```
oneway.test(y~x,data = data,var.equal = TRUE)
```

```
##
##  One-way analysis of means
```

```
##
## data:  y and x
## F = 4.4869, num df = 2, denom df = 297, p-value = 0.01203
```

and we again see that the p-value is 0.01203.

However, when running a one-way ANOVA assuming unequal variances

```
oneway.test(y~x,data = data,var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  y and x
## F = 4.5345, num df = 2.00, denom df = 197.56, p-value = 0.01187
```

We see that the p-value is 0.01187, which is not the same as the linear regression.

5.3 Statistical proof

The following proof was taken from (Hardy, 2012)

Suppose your data set consists of a set (x_i, y_i) for $i = 1, \ldots, n$ and you want to look at the dependence of y on x.

Suppose you find the values $\hat{\beta}_0$ and $\hat{\beta}_1$ of β_0 and β_1 that minimize the residual sum of squares

$$\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2.$$

Then you take $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ to be the predicted y-value for any (not necessarily already observed) x-value. That's linear regression.

Now consider decomposing the total sum of squares

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 \text{ where } \bar{y} = \frac{y_1 + \cdots + y_n}{n}$$

with $n - 1$ degrees of freedom, into "explained" and "unexplained" parts:

$$\underbrace{\sum_{i=1}^{n} ((\hat{\beta}_0 + \hat{\beta}_1 x_i) - \bar{y})^2}_{\text{explained}} + \underbrace{\sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2}_{\text{unexplained}}.$$

with 1 and $n - 2$ degrees of freedom, respectively. That's analysis of variance, and one then considers things like F-statistics

$$F = \frac{\sum_{i=1}^{n} ((\hat{\beta}_0 + \hat{\beta}_1 x_i) - \bar{y})^2 / 1}{\sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2 / (n - 2)}.$$

This F-statistic tests the null hypothesis $\beta_1 = 0$, which is the same as the traditional ANOVA:

$$F = \frac{\text{Between SS/df}}{\text{Within SS/df}}.$$

One often first encounters the term "analysis of variance" when the predictor is categorical, so that you're fitting the model

$$y = \beta_0 + \beta_i$$

where i identifies which category is the value of the predictor. If there are k categories, you'd get $k-1$ degrees of freedom in the numerator in the F-statistic, and usually $n-k$ degrees of freedom in the denominator. But the distinction between regression and analysis of variance is still the same for this kind of model.

Chapter 6

Outcomes

6.1 Length of stay

If your outcome is length of stay in a hospital, you should consider using a generalized linear model with Poisson or negative binomial distribution or Gaussian with log link (Austin et al., 2002).

Chapter 7

Multiple imputation

7.1 Longitudinal data

If the longitudinal measurements were taken at roughly ordered intervals (e.g. “1 month checkup”, “5 month checkup”), then try to reshape the data to wide format (one row per person) and then perform multiple imputations (Allison, 2002, UCLA (2017)).

Chapter 8

Regressions using survey data

8.1 Literature summary

- Endogenous sampling can be thought of as cases where the regression error term is related to the sampling criteria (Friedman, 2013, Solon et al. (2013), Fuller (2009))
- In the presence of endogenous sampling, unweighted estimates may be biased, but will be corrected by weighting by the inverse probability of selection (Friedman, 2013, Solon et al. (2013), Fuller (2009))
- In the presence of endogenous sampling, if the sampling probability varies across certain strata and those strata indicators are included in the estimating equation, then the probability of selection should no longer be related to the error term. Subsequently, weighting is not necessary (Friedman, 2013, Solon et al. (2013), Fuller (2009))
- In the case of a linear regression model that correctly specifies the conditional mean, the sampling would be exogenous if the sampling probabilities are independent of the error term in the regression equation. This would be the case, for example, if the sampling probabilities vary only on the basis of explanatory variables (Solon et al., 2013)
- More generally, the issue is whether the sampling is independent of the dependent variable conditional on the explanatory variables (Solon et al., 2013)
- Weighting does not correct for confounding. Adjusting for confounders is still necessary.
- Weighting is useful when the regression model that accounts for sampling probabilities does not make any sense (e.g. in the case-control scenario).

8.2 Setup

```
FormatNicely <- function(x,dp=3){
  formatC(x,digits=dp,format="f")
}

ConvertFitToResults <- function(fit,dataName,analysisName){
  r <- data.frame(coef(summary(fit)))[-1,-c(3:4),drop=F]
  r[,1] <- sprintf("%s [%s]",
                   FormatNicely(r[,1],dp=3),
                   FormatNicely(r[,2],dp=3))
  r <- r[,1,drop=F]
  r <- as.data.frame(t(r))
  r$data <- dataName
  r$analysis <- analysisName
```

```
  return(r)
}
```

8.3 Case-control studies

In this case study we will create a dataset `popData` that has a 1:1 linear relationship between the continuous exposure `x` and and the continuous outcome `y`. We also dichotomise `y` into a binary outcome `yBinary`.

We then create a dataset `casecontrolData` that oversamples cases 5x higher than is found in the normal population `popData`. We will then run linear regressions in the original population dataset and the case-control dataset and see how the effect estimates are affected by oversampling cases.

```
library(data.table)

# Creating population dataset
x <- runif(100000)
popData <- data.table(x)
popData[,y:=x+rnorm(.N)]

# Binary outcome
popData[,yBinary:=0]
popData[y>1,yBinary:=1]
popData[,oversampled:=yBinary]

popData[,inclusionProb := 5]
popData[oversampled==0, inclusionProb:=1]

# Case control dataset
cases <- popData[yBinary==1]
controls <- popData[yBinary==0]

# Cases are sampled 5x higher than normal
casecontrolData <- rbind(cases,cases,cases,cases,cases,controls)
casecontrolData[,id:=1:.N]
```

```
# Full population data set, unweighted regression
res <- vector("list",10)
fit <- glm(y~x,data=popData)
res[[1]] <- ConvertFitToResults(fit,dataName="Pop",analysisName="Unweighted")
res[[1]]$correlation <- FormatNicely(cor(resid(fit),fit$data$oversampled), dp=2)

# Biased dataset (case control with 5x), unweighted regression
fit <- glm(y~x,data=casecontrolData)
res[[2]] <- ConvertFitToResults(fit,dataName="Sample",analysisName="Unweighted")
res[[2]]$correlation <- FormatNicely(cor(resid(fit),fit$data$oversampled), dp=2)

# Biased dataset (case control with 5x), weighted regression
des <- survey::svydesign(id=~id,prob=~inclusionProb,data=casecontrolData)
fit <- survey::svyglm(y~x, design=des)
res[[3]] <- ConvertFitToResults(fit,dataName="Sample",analysisName="Weighted")
res[[3]]$correlation <- FormatNicely(cor(resid(fit),fit$data$oversampled), dp=2)

res <- rbindlist(res,fill=T)
```

Table 8.1: Effects of weights on linear regression coefficient estimates. (*Correlation between residuals and sampling probability).

Data	Analysis	coef(x) [sd(coef(x))]	Correlation*
Pop	Unweighted	0.989 [0.011]	0.74
Sample	Unweighted	0.910 [0.007]	0.77
Sample	Weighted	0.989 [0.009]	0.71

```
setcolorder(res, c("data", "analysis", "x", "correlation"))
setnames(res, c("Data", "Analysis", "coef(x) [sd(coef(x))]", "Correlation*"))

knitr::kable(
  res, booktabs = TRUE,
  caption = 'Effects of weights on linear regression coefficient estimates. (*Correlation between residu
)
```

We can see here that in the presence of endogenous sampling, the sampling probability is highly correlated with the regression error term. By weighting the data by the inverse probability of selection we obtain unbiased estimates of `coef(x)`.

8.4 Oversampling a population with a higher level of the outcome

```
library(data.table)

# Creating population dataset
x <- runif(100000)
popData <- data.table(x)
popData[,poor:=0]
popData[1:10000,poor:=1]
popData[,bmi:=22+1*x+5*poor+rnorm(.N)*2]

# Oversampled poor dataset
poor <- popData[poor==1]
notpoor <- popData[poor==0]

# Poor people are sampled 5x higher than not-poor
oversampledData <- rbind(poor,poor,poor,poor,poor,notpoor)
oversampledData[,id:=1:.N]

# Probability of inclusion
oversampledData[,inclusionProb := 5]
oversampledData[poor==0, inclusionProb:=1]
```

```
# Full population data set, unweighted regression
res <- vector("list",10)
res[[1]] <- ConvertFitToResults(fit <- glm(bmi~x,data=popData),
                                dataName="Pop",analysisName="Unweighted")
res[[1]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), unweighted regression
```

Table 8.2: Effects of weights on linear regression coefficient estimates. (*Correlation between residuals and sampling probability).

Data	Analysis	coef(x) [sd(coef(x))]	coef(poor) [sd(coef(poor))]	Correlation*
Pop	Unweighted	1.060 [0.027]	NA	0.60
Sample	Unweighted	1.108 [0.029]	NA	0.77
Sample	Weighted	1.060 [0.023]	NA	0.58
Pop	Unweighted+Strata	1.032 [0.022]	5.012 [0.021]	0.00
Sample	Unweighted+Strata	1.036 [0.018]	5.012 [0.011]	0.00
Sample	Weighted+Strata	1.032 [0.021]	5.012 [0.011]	0.00

```
res[[2]] <- ConvertFitToResults(fit <- glm(bmi~x,data=oversampledData),
                                dataName="Sample",analysisName="Unweighted")
res[[2]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), weighted regression
des <- survey::svydesign(id=~id,prob=~inclusionProb,data=oversampledData)
res[[3]] <- ConvertFitToResults(fit <- survey::svyglm(bmi~x, design=des),
                                dataName="Sample",analysisName="Weighted")
res[[3]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Full population data set, unweighted regression + strata indicator
res[[4]] <- ConvertFitToResults(fit <- glm(bmi~x+poor,data=popData),
                                dataName="Pop",analysisName="Unweighted+Strata")
res[[4]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), unweighted regression + strata indicator
res[[5]] <- ConvertFitToResults(fit <- glm(bmi~x+poor,data=oversampledData),
                                dataName="Sample",analysisName="Unweighted+Strata")
res[[5]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), weighted regression + strata indicator
des <- survey::svydesign(id=~id,prob=~inclusionProb,data=oversampledData)
res[[6]] <- ConvertFitToResults(fit <- survey::svyglm(bmi~x+poor, design=des),
                                dataName="Sample",analysisName="Weighted+Strata")
res[[6]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

res <- rbindlist(res,fill=T)
setcolorder(res,c("data", "analysis", "x", "poor", "correlation"))
setnames(res,c("Data", "Analysis", "coef(x) [sd(coef(x))]", "coef(poor) [sd(coef(poor))]", "Correlation*

knitr::kable(
  res, booktabs = TRUE,
  caption = 'Effects of weights on linear regression coefficient estimates. (*Correlation between residu
)
```

We can see here that in the first three models the sampling probability is highly correlated with the regression error term. Models 1 and 3 provide unbiased estimates (through exogenous sampling and weighting, respectively.) In models 4 through to 6, we included an explanatory variable `poor` to account for the varying sampling probabilities. Subsequently, the sampling probability is no longer correlated with the regression error term, and we obtain unbiased estimates.

8.5 Oversampling a population with a higher level of the outcome and exposure

```
library(data.table)

# Creating population dataset
x <- runif(100000)
popData <- data.table(x)
popData[,poor:=0]
popData[1:10000,poor:=1]
popData[,x:=x+2*poor]
popData[,bmi:=22+1*x+5*poor+rnorm(.N)*2]

# Oversampled poor dataset
poor <- popData[poor==1]
notpoor <- popData[poor==0]

# Poor people are sampled 5x higher than not-poor
oversampledData <- rbind(poor,poor,poor,poor,poor,notpoor)
oversampledData[,id:=1:.N]

# Probability of inclusion
oversampledData[,inclusionProb := 5]
oversampledData[poor==0, inclusionProb:=1]
```

```
# Full population data set, unweighted regression
res <- vector("list",10)
res[[1]] <- ConvertFitToResults(fit <- glm(bmi~x,data=popData),
                                dataName="Pop",analysisName="Unweighted")
res[[1]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), unweighted regression
res[[2]] <- ConvertFitToResults(fit <- glm(bmi~x,data=oversampledData),
                                dataName="Sample",analysisName="Unweighted")
res[[2]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), weighted regression
des <- survey::svydesign(id=~id,prob=~inclusionProb,data=oversampledData)
res[[3]] <- ConvertFitToResults(fit <- survey::svyglm(bmi~x, design=des),
                                dataName="Sample",analysisName="Weighted")
res[[3]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Full population data set, unweighted regression + strata indicator
res[[4]] <- ConvertFitToResults(fit <- glm(bmi~x+poor,data=popData),
                                dataName="Pop",analysisName="Unweighted+Strata")
res[[4]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), unweighted regression + strata indicator
res[[5]] <- ConvertFitToResults(fit <- glm(bmi~x+poor,data=oversampledData),
                                dataName="Sample",analysisName="Unweighted+Strata")
res[[5]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

# Biased dataset (poor oversampled 5x), weighted regression + strata indicator
```

Table 8.3: Effects of weights on linear regression coefficient estimates. (*Correlation between residuals and sampling probability).

Data	Analysis	coef(x) [sd(coef(x))]	coef(poor) [sd(coef(poor))]	Correlation*
Pop	Unweighted	3.026 [0.010]	NA	0.13
Sample	Unweighted	3.286 [0.006]	NA	0.09
Sample	Weighted	3.026 [0.006]	NA	0.13
Pop	Unweighted+Strata	1.017 [0.022]	4.951 [0.048]	0.00
Sample	Unweighted+Strata	1.033 [0.018]	4.919 [0.039]	0.00
Sample	Weighted+Strata	1.017 [0.021]	4.951 [0.043]	0.00

```
des <- survey::svydesign(id=~id,prob=~inclusionProb,data=oversampledData)
res[[6]] <- ConvertFitToResults(fit <- survey::svyglm(bmi~x+poor, design=des),
                                dataName="Sample",analysisName="Weighted+Strata")
res[[6]]$correlation <- FormatNicely(cor(resid(fit),fit$data$poor), dp=2)

res <- rbindlist(res,fill=T)
setcolorder(res,c("data", "analysis", "x", "poor", "correlation"))
setnames(res,c("Data", "Analysis", "coef(x) [sd(coef(x))]", "coef(poor) [sd(coef(poor))]", "Correlation

knitr::kable(
  res, booktabs = TRUE,
  caption = 'Effects of weights on linear regression coefficient estimates. (*Correlation between residu
)
```

We can see here that in the first three models the sampling probability is highly correlated with the regression error term. Models 1 and 3 provide estimates unbiased due to the endogenous sampling (through exogenous sampling and weighting, respectively), however, these estimates are still biased due to confounding. In models 4 through to 6, we included an explanatory variable `poor` to account for the varying sampling probabilities. Subsequently, the sampling probability is no longer correlated with the regression error term, and we obtain unbiased estimates.

Chapter 9

Matching

9.1 In case control studies

The aim of matching is to find controls with similar observable characteristics to the cases. This reduces bias due to confounding (Rubin, 1973).

These studies can be analysed using either conditional logistic regression or mixed effects logistic regression (with random intercepts for each matched stratum).

9.2 In non-case control studies

The aim of matching is to find non-exposed observations with similar observable characteristics to the exposed observations. This reduces bias due to confounding (Rubin, 1973).

These studies can be analysed using mixed effects regression (with random intercepts for each matched stratum).

Bibliography

Allison, P. D. (2002). *Missing data*. Number no. 07-136 in Sage university papers. Quantitative applications in the social sciences. Sage Publications, Thousand Oaks, Calif.

Austin, P. C., Rothwell, D. M., and Tu, J. V. (2002). A Comparison of Statistical Modeling Strategies for Analyzing Length of Stay after CABG Surgery. *Health Services and Outcomes Research Methodology*, 3(2):107–133.

Friedman, J. (2013). Tools of the trade: when to use those sample weights.

Fuller, W. A. (2009). *Sampling Statistics*. John Wiley & Sons, Inc., Hoboken, NJ, USA. DOI: 10.1002/9780470523551.

Gelman, A. (2005). Analysis of variance? Why it is more important than ever. *The Annals of Statistics*, 33(1):1–53.

Hardy, M. (2012). Difference between regression analysis and analysis of variance? - Cross Validated.

Rubin, D. B. (1973). Matching to Remove Bias in Observational Studies. *Biometrics*, 29(1):159–183.

Solon, G., Haider, S., and Wooldridge, J. (2013). What Are We Weighting For? Technical Report w18859, National Bureau of Economic Research, Cambridge, MA. DOI: 10.3386/w18859.

UCLA, S. C. G. (2017). How can I perform multiple imputation on longitudinal data using ICE?

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.3.8.