rm(list=ls()) use brush icon

install.packages("dplyr")

install.packages('tidyverse')

library(dplyr)

library(tidyverse)

library(lubridate)

library(stringr)

library(car)

library(corrplot)

library(gmodels)

library(psych)

library(vcd)

library(gridExtra)

library(plotly)

library(RColorBrewer)

library(ggplot2)

library(effects)

library(Deducer)

library(rms) for pseudo-R-Squared

1.Look at the data

2. Remove digitpattern

3. check fr data type and convert – new column (easy to filter with old)

4. missing data omit na – rows, columns.

and odd values. Remove odd values

Start with hypothesis

5. univariate analysis

6. check fr outliers. Outliers at least try to remove or comment it ll have impact on the analysis

Do bivariate plots

Tests

Say d result

load("Session 7 (review).RData")

View(realestate)

glimpse(realestate)

colSums(is.na(realestate))

credit1<-na.omit(credit1)

digitpattern <- "\\$|\\.00|\\,"

gsub(digitpattern, "", cost) # remember to reassign the processed output and convert to the correct data type if needed

bp$ESTIMATED\_COST <- as.numeric(gsub(digitpattern, "", bp$ESTIMATED\_COST))

Tab

up arrow

Whenever you're working with a new dataset, the first thing you should do is look at

| it! What is the format of the data? What are the dimensions? What are the variable

| names? How are the variables stored? Are there missing data? Are there any flaws in

| the data?

Now, store the result of x - 3 in a new variable called y.

> y <- x-3

| Excellent job!

|========================= | 29%

| What is the value of y? Type y to find out.

> y

[1] 9

The easiest way to create a vector is with the c() function, which

| stands for 'concatenate' or 'combine'. To create a vector containing

| the numbers 1.1, 9, and 3.14, type c(1.1, 9, 3.14). Try it now and

| store the result in a variable called z.

c() function is used for creating a vector

?c – Help

Getwd()

Ls() – list all objects – which are also given in the side

List.files()

rm(list=ls()) – removes objects

<https://github.com/swirldev/swirl_courses/blob/master/R_Programming/Sequences_of_Numbers/lesson.yaml>

rep(c(0, 1, 2), times = 10)

seq(5, 10, length=30)

Make sure there is no space between the `>` and `=` symbols.

`>`, `<=`, `==` for exact equality, and `!=`

|  |
| --- |
| A | B (logical 'or' a.k.a. 'union') or whether they are both |
|  | TRUE with A & B (logical 'and' a.k.a. 'intersection'). Lastly, !A is the negation |
|  | of A and is TRUE when A is FALSE and vice versa. |

Sample:

my\_data <- sample(c(y, z), 100)

my\_na <- is.na(my\_data) gives true is value is na. true is 1 so sum

sum(my\_na)

NaN, which stands for 'not a number'.

try x[1:10] to view the first ten elements of x.

y <- x[!is.na(x)] to capture all non-missing values from x.

x[!is.na(x) & x > 0]

x[c(3, 5, 7)] - subset the 3rd, 5th, and 7th elements of x

x[-c(2, 10)] except 2n 10

ch7 – matrices n dataframes:

my\_vector <- 1:20

dim(my\_vector) no. length(my\_vector) 20

dim(my\_vector) <- c(4, 5) 4 rows 5 columns

class(my\_vector) – matrix

same as my\_matrix2 <- matrix(1:20, nrow=4, ncol=5)

adding char - patients <- c("Bill", "Gina", "Kelly", "Sean")

Type cbind(patients, my\_matrix) to add the names of our patients to the matrix of numbers.

This causes everything to be character

So we need data.frame()

matrices can only contain ONE class of data – numeric or character not both

my\_data <- data.frame(patients, my\_matrix)

single object of class `data.frame

# data frame. It is the default class for

| data read into R using functions like read.csv() and read.table()

data types:

something

[1] 12 9 11

Something is a vector and class of it is numeric

class(a>b) logical

b<-"hello" class(b) - "character"

?class # help on a function

help(package="ggplot2") #?doesnt wrk here

a <-5

b<-"hello"

c<-FALSE

> is.logical(a)

[1] FALSE

> is.numeric(a)==FALSE

[1] FALSE

> class(a)!=class(c)

[1] TRUE

a3 <- round(a3,1)

head(a3) gives 10

tail(a3)

new.seq <- round(tail(my.seq,n=4),1) gives last 4 n not 10 . n rounded to one decimal

session 2:

dim(plants)

[1] 5166 10 – rows, columns

names(plants)

column names

Summary()

For categorical variables (called 'factor' variables in R), summary() displays the

| number of times each value (or 'level') occurs in the data.

Str()it tells

| us that the class of plants is 'data.frame' and that it has 5166 observations and 10

| variables. It then gives us the name and class of each variable, as well as a preview

| of its contents.

# We can also convert a dataframe to a tibble, using the function: tibble()

# using data.frame() will convert to a data frame.

Glimpse()

# Examine data in a spreadsheet-like format - View(supplyfile)

Wen u convert to factor do as separate file else u cant filter with less than symbol

supplyfile$ID <- as.character(supplyfile$ID)

supplyfile$Member <- as.factor(supplyfile$Member)

supplyfile$StartDate <- as.character(supplyfile$StartDate) #first set date to character n then convert using

# Use lubridate (loaded with tidyverse) to convert the character into a date

supply1$StartDate <- mdy(supply1$StartDate)

summary(supply1$StartDate)

bp1$issue\_date <- mdy(bp1$ISSUE\_DATE)

summary(bp1$issue\_date)

employees %>% arrange(desc(salarykd), gender)

supplyfile$total\_purchase – creates a new column in the data

sometimes subset n rename - empmale

sometimes overwrite

while recoding new column - supplyfile$MemberFactor[supplyfile$Member==1]<-"Member"

supplyfile$MemberFactor[supplyfile$Member==0]<-"NotMember"

distribution:

hist(supplyfile$total\_purchase)

boxplot(supplyfile$Plumbing)

par(mfrow=c(2,2)) 2 rows 2 columns in plot canvas

hist(supply1$Plumbing)

hist(supply1$Electric)

hist(supply1$HVAC)

hist(supply1$total\_purchase)

boxplot(supplyfile$Plumbing)

boxplot(supplyfile$Electric)

boxplot(supplyfile$HVAC)

boxplot(supplyfile$total\_purchase)

supplymembers <- supplyfile[supplyfile$Member==1,]

target <- supplyfile[supplyfile$total\_purchase<=threshold,]

# Method 3: Using tidyverse

target2 <- supplyfile %>% filter(total\_purchase<=threshold) %>% select(StartDate, Member, total\_purchase) # selects columns

LessAvgMon <- employees %>% filter(months < avgmon) %>% select(gender\_factor, salarykd)

session 3:

colSums(is.na(bp)) # colSums sums up the values from a numeric vector/column

View(colSums(is.na(bp)))

Sapply apply loop function

*#22 rows have Na value in this column*

***forbes1 <- forbes[!is.na(forbes$FiveYrReturn),]***

*#removing those rows*

*a2 <- is.na(forbes1$FiveYrReturn)*

*sum(a2)*

*# verifying no NA are there*

Lapply sapply

convert <- c("ESTIMATED\_COST","AMOUNT\_PAID")

pb[convert] <- sapply(bp[convert],as.numeric)

emp\_factors <- c("department", "gender", "marital", "overtime", "quit", "travel")

emp[emp\_factors] <- lapply(emp[emp\_factors], factor)

<https://www.regular-expressions.info/rlanguage.html>

digitpattern <- "\\$|\\.00|\\,"

gsub(digitpattern, "", cost) # remember to reassign the processed output and convert to the correct data type if needed

bp$ESTIMATED\_COST <- as.numeric(gsub(digitpattern, "", bp$ESTIMATED\_COST))

session 4: if else

bp1$contractor\_city <- gsub("CHGO", "CHICAGO", bp1$contractor\_city)

bp1$city <- ifelse(grepl("CHICAGO", bp1$contractor\_city), "Chicago", "Other")

ifelse(supplyfile$total\_purchase<=threshold, "Target", "NonTarget") %>% table()

Uni: distribution:

Summary prop.table plot

Factor variable – barplot

Numeric variable mean hist boxplot density plots qq plot - Histogram with overlaid density and rug plots

*Bivariate:*

*2factors- table, prop table,barplot*

tab\_permit\_city <- table(bp1$permit, bp1$city)

barplot(prop.table(xtab\_permit\_city), col=1:10, ylim=c(0,1), main="Permits by City", beside=T)

# Test – chi sq, cross table

summary(xtab\_permit\_city) # or more directly

chisq.test(bp1$permit, bp1$city) # but there's a better option

CrossTable(bp1$permit, bp1$city, prop.c = F, prop.r = F,

dnn = c('Permit', 'City'), format=c("SPSS"), chisq = T)

2 numeric – plot,abline, lines.smooth

plot(log(bp2$total\_fee+1)~log(bp2$est\_cost+1), col="steelblue", pch=20, cex=0.75)

abline(lm(log(total\_fee+1)~log(est\_cost+1), data=bp2), col="red", lwd=2)

abline – regrn line

# Correlation matrix

cormat <- cor(bp2[,numcols])

round(cormat,2)

corrplot(cormat, method="circle", addCoef.col="grey", type="upper")

# pairs(bp1[,numcols]) # this one is slow. Try it later!

# Test –correlation. Regrn, pairs

# Correlation

cor(bp1$est\_cost, bp1$total\_fee)

cor.test(bp1$est\_cost, bp1$total\_fee) # As expected

Factor and a numeric variable – boxplot(left side numeric), a compaitive kernel density plot

# Explore

aggregate(est\_cost ~ city, data = bp1, FUN="mean", na.rm=T)

aggregate(est\_cost ~ city, data = bp1, FUN="sd", na.rm=T)

# Tidyverse

bp1 %>% group\_by(city) %>% summarize(avg=mean(est\_cost), median=median(est\_cost), sd=sd(est\_cost))

# Plot

boxplot(est\_cost ~ city, data=bp2, main="Comparing Distributions by City",

xlab="City", ylab="Estimated Cost", col=c("orange", "steelblue"))

Using dplyr's group\_by and summarise is a good alternative to the aggregate function. For instance,

library(dplyr)

realestate %>% group\_by(airco) %>% summarise(avg=mean(price), med=median(price), std=sd(price))

# It is possible to plot comparative density plots with ggplot, like so:

realestate %>% ggplot(aes(x=price, fill=airco)) +

geom\_density(lwd=2, alpha=0.4)

# Test –anova, tukeys

est\_cost\_model <- aov(log(est\_cost+1)~city, data=bp2)

summary(est\_cost\_model)

# Tukey pairwise comparisons – if the factor has more than 2 levels tukey is needed. Category – yes, no – 2 levels not needed.. more than tat need to do

TukeyHSD(est\_cost\_model)

# Numeric variable and a factor

boxplot(pay~division, data=baseball,

col=brewer.pal(3, "RdGy"),

main="Distribution of budget\nby divisions")

# A comparative kernel density plot

plot(density(baseball$pay[baseball$division=="Central"]), col="red", lwd=2, ylim=c(0,0.02))

lines(density(baseball$pay[baseball$division=="East"]), col="blue", lwd=2)

lines(density(baseball$pay[baseball$division=="West"]), col="green", lwd=2)

6:

# Plotting three variables (two numeric and a factor)

# Univariate

# numeric

# factor

# Bivariate

# Two numeric

# Two factors - use 3rd option 100% fill

# Numeric and factor

baseball %>% ggplot(aes(x=division, y=pay)) +

geom\_boxplot(aes(fill = division)) +

scale\_fill\_brewer(palette="Set2")

# foll is density kernel plot

baseball %>% ggplot(aes(x=pay, fill=division)) +

scale\_fill\_brewer(palette="Spectral")+

geom\_density(alpha=.5) #alpha capcity of plot

# Three (or more) variables

ggplot(data=cars, aes(x=mileage, y=price,

color=transmission)) +

geom\_point() +

facet\_grid(.~model) #4factors #price vs mileage by tansmission by model #facets always #no much detail

# grid used for two variables, wrap for one

# Statistical plots (e.g., linear and smoothers)

# Arranging ggplots on canvas

#-saving ggplots

Interactive, heatmap

Regrn:

# Now we can model (regress) price given mileage #we seee correlan betn proce yr mileage#including in model

mod1<-lm(price ~ mileage, data=cars)

mod1

options(scipen=99)

summary(mod1) #adj rsq. by knowing mileage we can predict price with 65% accuracy

confint(mod1) #interpret it as fr a unit incease of mile per unit wats the price range

# Predict specific values based on a model

# How well did our model perform?

# R-sqrd

lotsize as the regressor

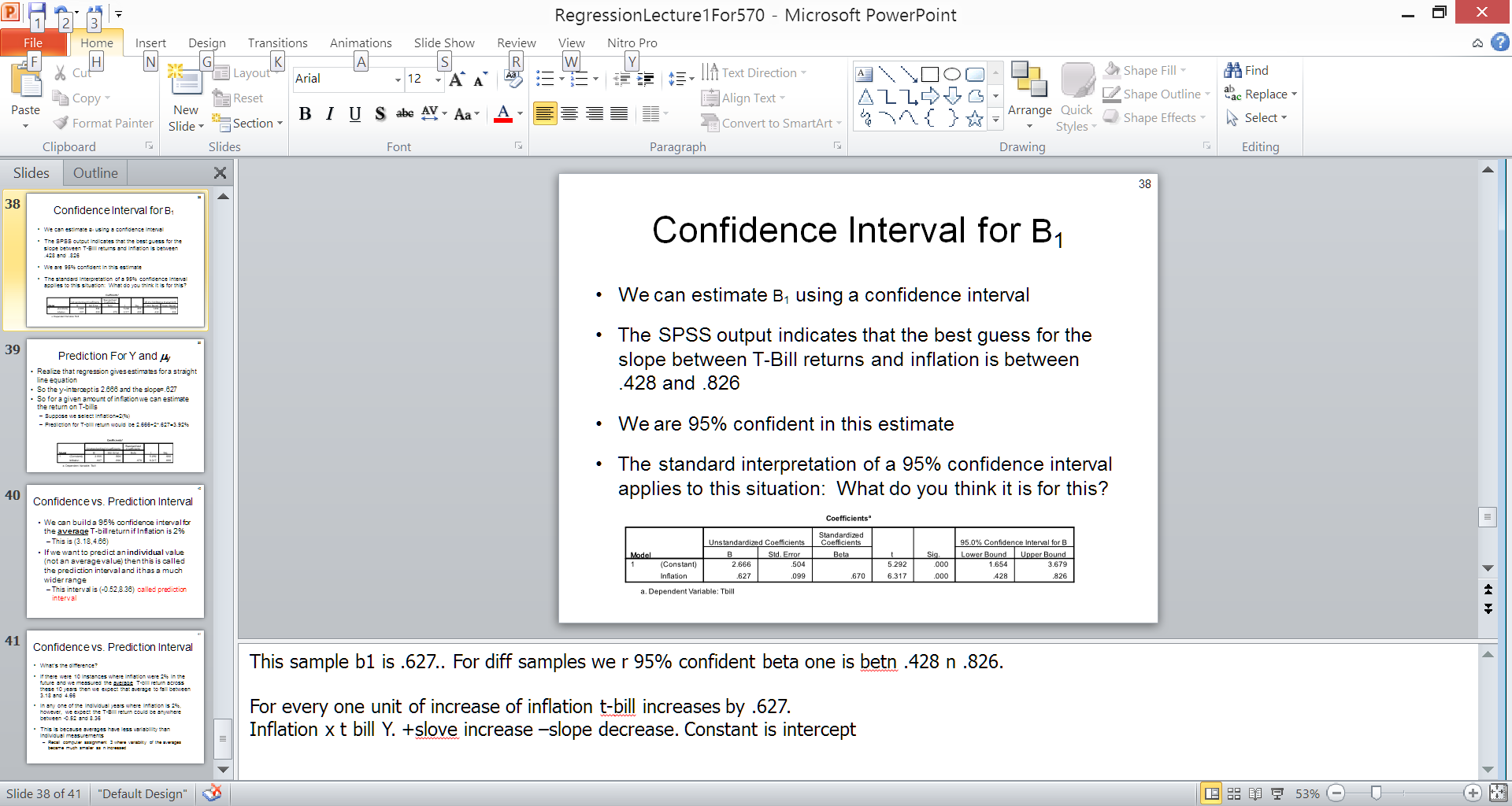
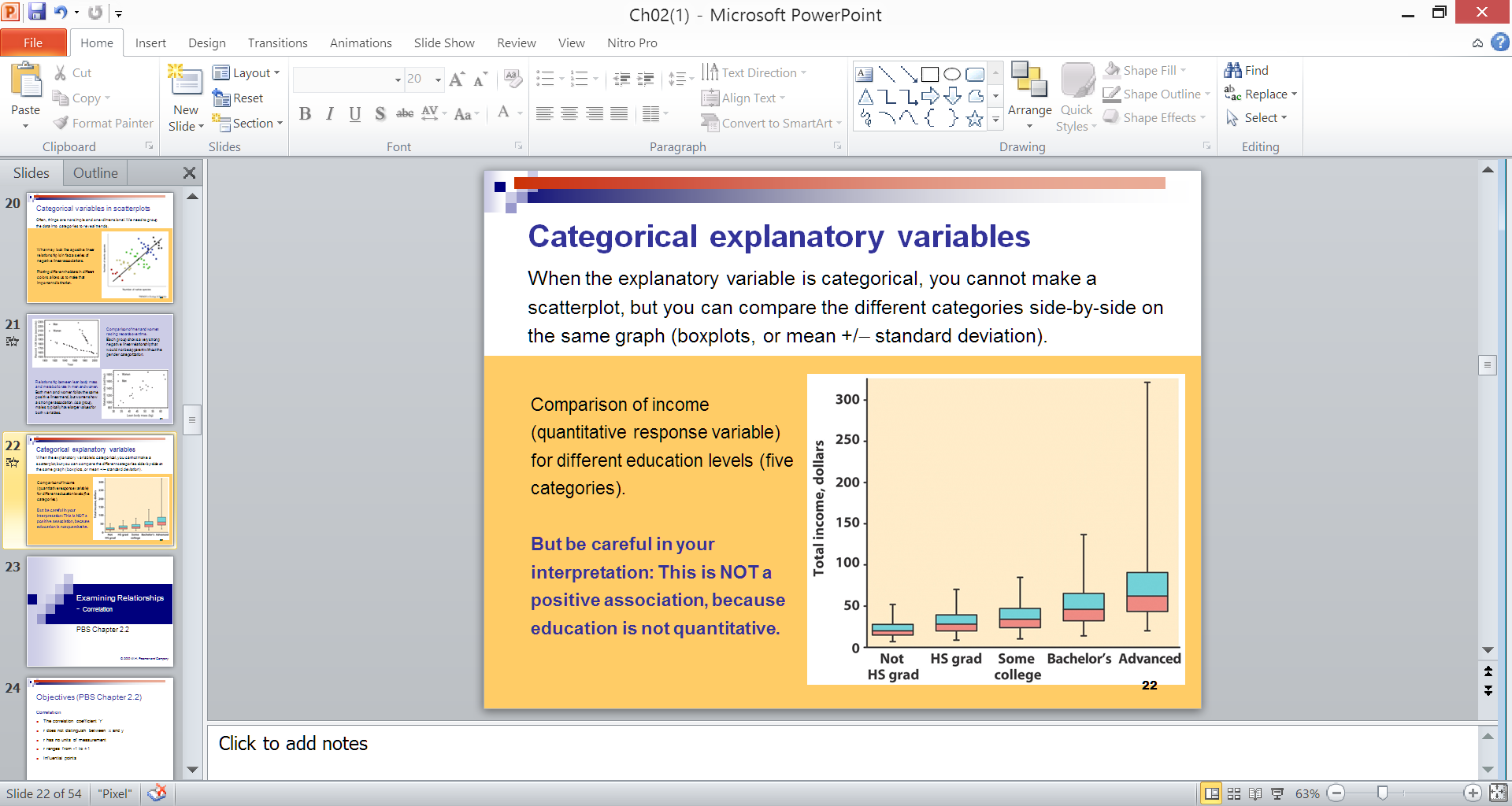
mod2<-lm(price ~ lotsize, data=realestate)

mod2

summary(mod2)

# we can also set confidence interval at 99%

confint(mod2, level=0.99)



So, if R-square is 0.8, it means 80% of the variation in the output variable is explained by the input variables.

#adj rsq. by knowing mileage we can predict price with 65% accuracy

The **p-value** of the test is 1.29410^{-10}, which is less than the significance level alpha = 0.05. We can conclude that wt and mpg are significantly correlated with a correlation coefficient of -0.87 and p-value of 1.29410^{-10} .

R = -negative 80 means strong

Pearson's Chi-squared test

data: housetasks

X-squared = 1944.5, df = 36, p-value < 2.2e-16

In our example, the row and the column variables are statistically significantly associated (*p-value* = 0).

**Correlation test** is used to evaluate the association between two or more variables.

<http://www.sthda.com/english/wiki/one-way-anova-test-in-r>

Df Sum Sq Mean Sq F value Pr(>F)

group 2 3.766 1.8832 4.846 0.0159 \*

Residuals 27 10.492 0.3886

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The output includes the columns *F value* and *Pr(>F)* corresponding to the p-value of the test.

**Interpret the result of one-way ANOVA tests**

As the p-value is less than the significance level 0.05, we can conclude that there are significant differences between the groups highlighted with “\*" in the model summary.

TukeyHSD(res.aov)

Tukey multiple comparisons of means

95% family-wise confidence level

Fit: aov(formula = weight ~ group, data = my\_data)

$group

diff lwr upr p adj

trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711

trt2-ctrl 0.494 -0.1972161 1.1852161 0.1979960

trt2-trt1 0.865 0.1737839 1.5562161 0.0120064

* **diff**: difference between means of the two groups
* **lwr**, **upr**: the lower and the upper end point of the confidence interval at 95% (default)
* **p adj**: p-value after adjustment for the multiple comparisons.

It can be seen from the output, that only the difference between trt2 and trt1 is significant with an adjusted p-value of 0.012.

Univariate analysis:

options(scipen=99)

summary(bp1)

bp1 <- bp1 %>% filter(est\_cost!=9999999999)

bp1$contractor\_city <- gsub("CHGO", "CHICAGO", bp1$contractor\_city)

bp1$city <- ifelse(grepl("CHICAGO", bp1$contractor\_city), "Chicago", "Other")

tab\_city <- table(bp1$city)

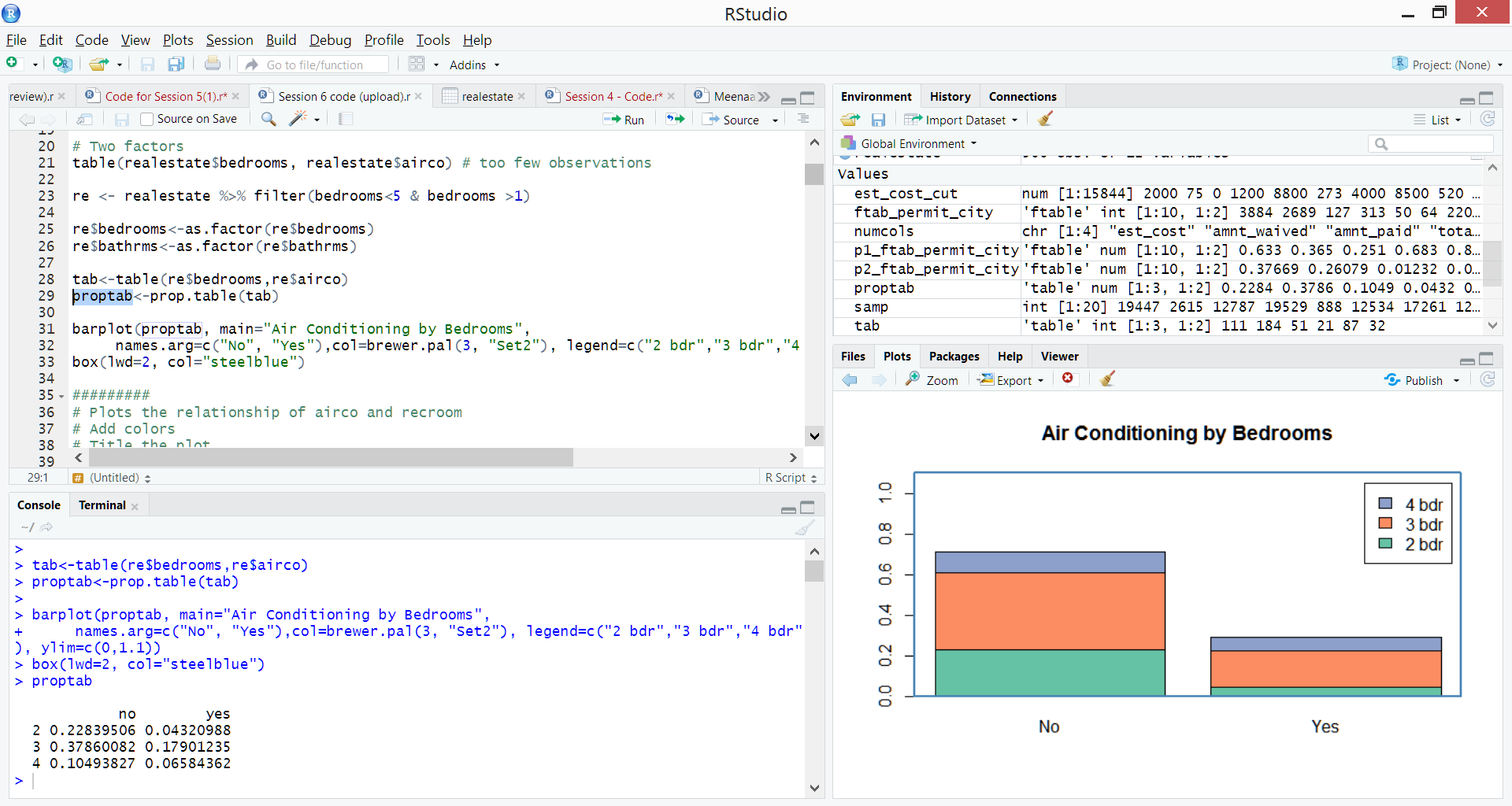
prop.table(tab\_city)

barplot(prop.table(tab\_city))

desc<-describe(bp1[numcols])

class(desc)

View(round(desc,2))



# Relationship between a/c and price in the realestate data

#===========

# Step 1 (data type - intuitively, and after inspecting the data, what should the data type(s) be?)

# a/c should be a factor (we consider having/not having a/c in a house, so a two level factor)

# price should be numeric

# Step 2 (clean the data, if needed. inspect data type, NAs and 'odd' values)

summary(realestate) # NAs will be noted; there are no NAs

# Both variables are in their correct type, and do not have any NAs or odd values. No need to filter/change the data.

# Step 3 (hypothesize - intuitively, what should be the relationship?)

# I hypothesize that there should be a relationship between the two variables. More expensive houses are more likely to have a/c.

# Step 4 (examine univariate distributions)

# We can use the summary function in step 2, or

table(realestate$airco) # There are 357 houses without a/c compared to only 143 with a/c in the data.

# For the distribution of price, see Question 3. It shows that the variable is slightly right-skewed.

# Step 5 (examine bivariate relationship)

# Stats

aggregate(price~airco, data=realestate, FUN="mean") # mean of houses with a/c is higher by about 41% than houses without a/c

aggregate(price~airco, data=realestate, FUN="median") # median of houses with a/c is higher by about 41% than houses without a/c

aggregate(price~airco, data=realestate, FUN="sd") # sd of houses with a/c is higher by about 41% than houses without a/c

# There is a noticeable difference in prices between houses with and without a/c. Houses with a/c tend to have a price that is about 41% higher.

# Plot

options(scipen=99)

boxplot(price~airco, data=realestate, main="Comparative boxplot of house prices with/out a/c", col=c("red", "blue"))

# Distribution of prices for both is right-skewed, as indicated by outliers that are over 1.5 IQR in both categories. However there are more outliers for houses without a/c.

# This may be a result of having more cases. It is also interesting to note that the distribution of houses without a/c is more compact (smaller IQR), compared to houses with a/c.

# Step 6 (testing)

mod\_ac <- aov(price~airco, data=realestate)

summary(mod\_ac)

# The ANOVA test shows a strong relationship between price and the air condition variable.

# Step 7 (answer)

# ANSWER: As I hypothesized, all the above suggest that there is a relationship between a/c and house prices.

# Houses with a/c will be, overall, more costly than houses without a/c.

\\\1. backslash wont work - path

forward slas n double backslash ll work

console - output window

everything in green is comment

categorical variable is called as Factor in R

ID collumn not numeric cuz cant take mean distribution etc is string (whch is called as Character in R)

Factor, numeric, Character, Date - data types

vector - unidimensional ; same as array

considering ID a column as vector. cuz column - unidimensional

c function is used for combining concataenating

dates given within double quuotes else ll divide

data frame - 2d set of values - oe fr row n other for column

other data type is array similar to data frame but ll be used only in advanced

diff betn array and data frame is array needs all columns to be ame ata type (all numeric or all char). but data frame accepts mized data types

List - very flexible accepts array data frame array. but very complicated as mixed

common type used is hence data frame

if working in groups its betr to give the code so others can replicate. so during exams after opening in gui copy the line n paste above so he can replicate

we use True or False tstements to filer the data we have

09/13

if columns missmore than 20% data we ll not consider it

\\d - MEANS digit

gonna use long formT as opposed to wide formAT.

cuz wide format diff to build into regrn model. reserah he is not teaching

imp :

gsub grep

del columns

remove rows

POSIXt class high frequency time - mdy-hms

str() - strcture of class