



Università degli Studi del Piemonte Orientale

DIPARTIMENTO DI STUDI PER L'ECONOMIA E L'IMPRESA

Corso di Laurea triennale in
ECONOMIA AZIENDALE

PROVA FINALE

**R Commander Graphical User Interface (GUI):
funzionamento e applicazione**

Candidato:

Alessandro Maccario

Matricola 20023776

Relatore:

Prof. Enea Giuseppe Bongiorno

R Commander Graphical User Interface (GUI): funzionamento e applicazione

ALESSANDRO MACCARIO

Indice

| | | |
|----------|---|-----------|
| 1 | INTRODUZIONE | 4 |
| 1.1 | Obiettivo dell'elaborato | 4 |
| 1.2 | Strutturazione dell'elaborato | 4 |
| 2 | RCMDR | 5 |
| 2.1 | R, R Studio, R Commander | 5 |
| 2.1.1 | Che cos'è R | 5 |
| 2.1.2 | Che cos'è R Studio | 7 |
| 2.1.3 | Che cos'è R Commander o Rcmdr | 8 |
| 2.2 | Installazione del pacchetto | 9 |
| 2.3 | Aspetto e funzionalità | 12 |
| 3 | TITANIC DATASET: ANALISI ESPLORATIVA E PRE-DITTIVA | 21 |
| 3.1 | Storia e struttura, obiettivo dell'analisi e metodologie applicabili | 21 |
| 3.2 | Analisi esplorativa del data set (EDA: Exploratory Data Analysis) | 22 |
| 3.3 | Modello di regressione logistico: previsione della sopravvivenza dei passeggeri | 53 |
| 3.4 | Stima dei coefficienti del modello | 58 |
| 3.5 | Applicazione del modello ai dati | 59 |
| 3.6 | Esportazione dei risultati: R Markdown | 87 |

Elenco delle figure

| | | |
|------|---|----|
| 2.1 | Interfaccia di R | 6 |
| 2.2 | Interfaccia di R Studio | 8 |
| 2.3 | Interfaccia di R Commander | 10 |
| 2.4 | Menù a tendina | 11 |
| 2.5 | Rcmdr installazione | 13 |
| 2.6 | Comando library() | 14 |
| 2.7 | Rcmdr GUI | 15 |
| 2.8 | Menù GUI | 16 |
| 2.9 | Toolbar | 18 |
| 2.10 | Script-Markdown | 18 |
| 2.11 | Pannello Output | 19 |
| 2.12 | Pannello Messaggi | 20 |
| 3.1 | Import data da Excel | 23 |
| 3.2 | Finestra di dialogo | 24 |
| 3.3 | Scelta del data set | 25 |
| 3.4 | Caricamento data set | 26 |
| 3.5 | Visualizzazione data set | 27 |
| 3.6 | Summary data set | 29 |
| 3.7 | Risultato Summary | 29 |
| 3.8 | Boxplot box di dialogo | 31 |
| 3.9 | Boxplot tariffa | 32 |
| 3.10 | Boxplot Outliers | 33 |
| 3.11 | Recoding | 35 |
| 3.12 | Box recoding | 36 |
| 3.13 | Risultato recoding | 36 |
| 3.14 | Tavola di contingenza Classe Passeggero | 37 |
| 3.15 | Output Tavola di contingenza | 38 |
| 3.16 | Output Sopravvivenza-Genere | 41 |
| 3.17 | Multi-way table | 42 |
| 3.18 | Multi-way table parametri | 43 |

| | |
|---|----|
| 3.19 Multi-way table Output (1) | 44 |
| 3.20 Multi-way table Output (2) | 45 |
| 3.21 Bar Chart - Classe-Sopravvivenza | 46 |
| 3.22 Graphs - Bar Chart | 47 |
| 3.23 Graphs - Variabili | 48 |
| 3.24 Table of statistics | 49 |
| 3.25 Table of statistics - Box di dialogo | 50 |
| 3.26 Table of statistics - Output | 51 |
| 3.27 Table of statistics - Percentuali di colonna | 52 |
| 3.28 Bar chart - Et -Sopravvivenza | 54 |
| 3.29 Modello lineare e Modello logistico | 57 |
| 3.30 Scheda Set-seed | 60 |
| 3.31 Set-seed box di dialogo | 61 |
| 3.32 Set-seed Output | 62 |
| 3.33 Installazione pacchetto "caret" | 64 |
| 3.34 Richiamo del pacchetto "caret" | 65 |
| 3.35 Frazionamento del data set in train e test set | 67 |
| 3.36 Visualizzazione del train set | 68 |
| 3.37 Titanic training set | 69 |
| 3.38 Generalized linear model | 70 |
| 3.39 Generalized linear model - Box di dialogo | 71 |
| 3.40 GLM scelta delle variabili | 73 |
| 3.41 GLM Output | 74 |
| 3.42 GLM Summary (1) | 75 |
| 3.43 GLM Summary (2) | 76 |
| 3.44 Scheda "Predict using active model" | 78 |
| 3.45 "Predict using active model" box di dialogo | 79 |
| 3.46 predict fitted.logistic.model.1 | 80 |
| 3.47 Pacchetto "arm" | 82 |
| 3.48 Invlogit | 83 |
| 3.49 Titanic predict Output | 85 |
| 3.50 Accuratezza della previsione del modello logistico | 86 |
| 3.51 R Markdown Template | 87 |
| 3.52 Chunks in R Markdown | 88 |
| 3.53 Modifica del testo Markdown | 90 |
| 3.54 Scelta del formato Markdown | 91 |
| 3.55 Markdown output in PDF | 92 |
| 3.56 Markdown output in PDF Commentato | 93 |

Capitolo 1

INTRODUZIONE

1.1 Obiettivo dell'elaborato

L'obiettivo di questo elaborato è di spiegare e mostrare il funzionamento dell'interfaccia grafica (GUI) R Commander del software per l'analisi statistica R. Il suo funzionamento verrà applicato anche all'analisi di un data set (Titanic data set) con l'intenzione di mostrare la potenzialità di tale strumento (in ambiente Windows).

R Commander è stato pensato per avvicinare l'utilizzatore sprovvisto delle adeguate conoscenze informatiche alle analisi statistiche tramite R, attraverso un'interfaccia *point-and-click* simile a quella utilizzata in altri software come, ad esempio, Minitab o Excel, per cui è sufficiente, avendo le conoscenze necessarie per le analisi da approntare sul data set, cliccare su specifici menù a discesa per poi selezionare lo strumento prescelto.

1.2 Strutturazione dell'elaborato

L'elaborato verrà suddiviso in 3 macro-capitoli, come segue:

Capitolo 1: obiettivo dell'elaborato;

Capitolo 2: R, R Studio, R Commander, installazione del pacchetto, aspetto e sue funzionalità;

Capitolo 3: obiettivo dell'analisi esplorativa e metodologie applicabili, EDA, modello di regressione logistico, visualizzazione dei risultati e loro esportazione tramite R Markdown.

Capitolo 2

RCMDR

Prima di spiegare il funzionamento di R Commander, è necessario dare alcune indicazioni di carattere generale sul funzionamento degli strumenti alla base di tale interfaccia.

2.1 R, R Studio, R Commander

2.1.1 Che cos'è R

R è un linguaggio ed un ambiente di lavoro per l'elaborazione statistica e grafica dei dati¹.

Si tratta di un software *open-source* e completamente gratuito, dove con *open-source* si intende che il codice sorgente è disponibile a chiunque sia in grado e abbia intenzione di modificarlo, ad esempio per aggiungerne funzionalità o correggere errori. Nello specifico, R possiede una licenza *GPL* (*General Public License*)²: tale licenza garantisce la completa libertà nel condividere e alterare il sorgente del software.

Tramite l'utilizzo di R è possibile svolgere analisi statistiche *scrivendo codice*, ovvero digitando una serie di comandi per dialogare con il software e indicargli quali azioni compiere. Di conseguenza, una analisi statistica standard prevede la conoscenza del linguaggio corretto per poter operare. Per fare un esempio, se si volesse calcolare la media della variabile *reddito*, si scriverebbe il seguente codice³:

¹Free Software Foundation's GNU project. *r-project*. URL: <https://www.r-project.org/about.html>.

²Richard Stallman. *GNU General Public License*. URL: <https://www.gnu.org/licenses/gpl-3.0.en.html>.

³John Fox. *Using the R commander: A point-and-click interface for R*. CRC Press, 2016.

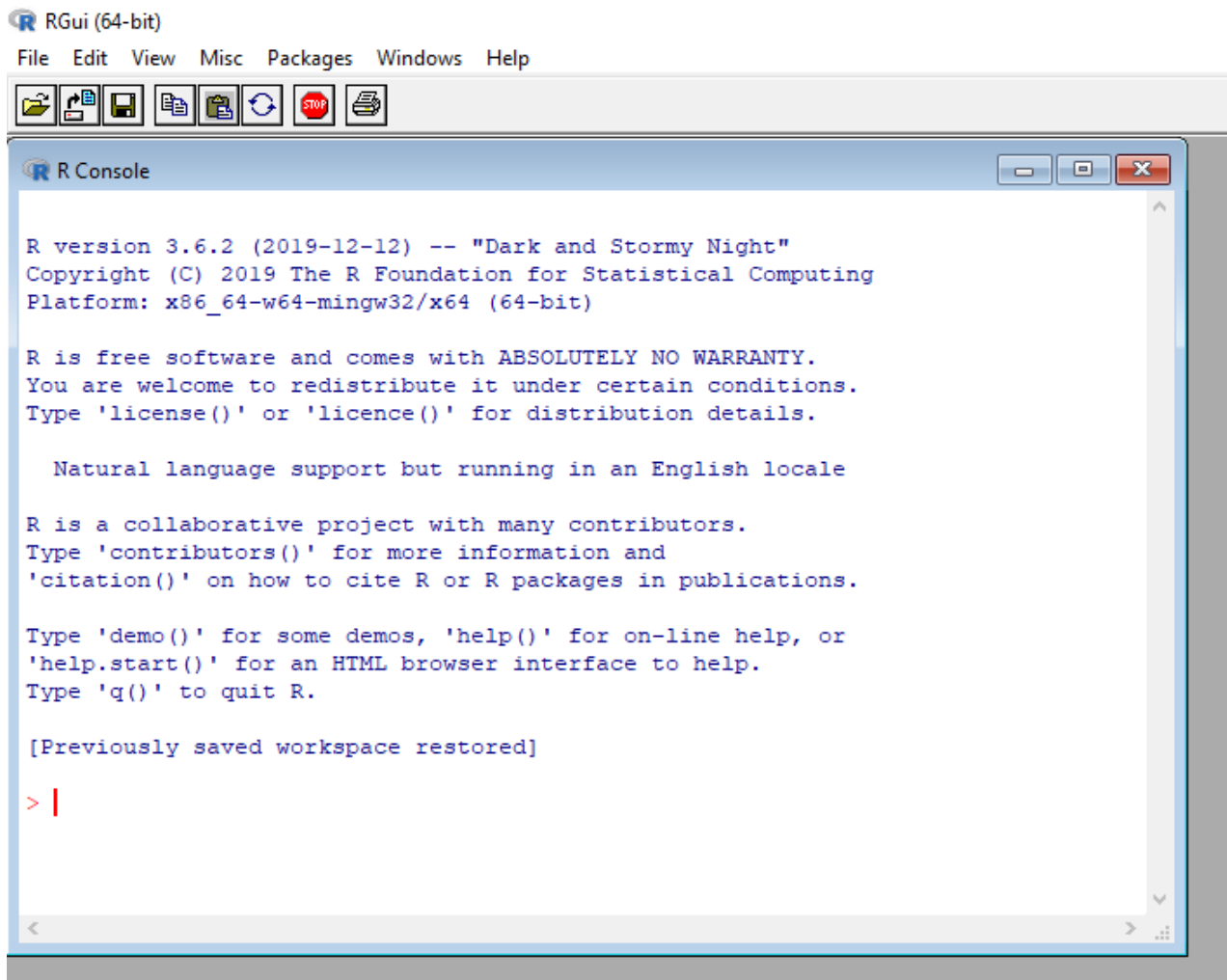


Figura 2.1: Interfaccia di R

```
mean(reddito)
```

Lo spazio di lavoro è minimale e l'interfaccia grafica utilizzata per inserire i comandi si presenta come in Figura 2.1 a pagina 6.

Subito dopo il simbolo `>` si inseriscono i comandi necessari per l'analisi statistica scelta. R può essere usato senza l'ausilio di altri software o altre interfacce grafiche.

2.1.2 Che cos'è R Studio

R Studio è un *IDE* per R, ovvero un *Integrated Development Environment*, un ambiente di sviluppo integrato. Con tale descrizione si intende un'applicazione «che aggrega strumenti di sviluppo comuni in un'unica interfaccia grafica»⁴ per l'utilizzatore. Gli strumenti aggregati sono i seguenti:

- **Editor del codice sorgente:**⁵ si tratta di una finestra nella quale è possibile scrivere il codice sorgente del progetto statistico su cui si sta lavorando con utili funzionalità integrate, quali la marcatura della sintassi che permette di distinguere, ad esempio, tra **funzioni**, **variabili**, **punteggiatura**. Un'altra importante funzionalità è quella del **completamento automatico**, che permette una più rapida scrittura del codice;
- **Interprete:** è un programma per computer che elabora il codice sorgente di un progetto **riga per riga** per poi inviarle al processore. Nel caso sia presente un'errore nell'elaborazione del codice, il processo viene interrotto e viene mostrata la riga nella quale l'errore si è presentato. Questo approccio facilita ovviamente il *debug* istantaneo del codice sorgente;
- **Debugger:** è un software in grado di analizzare e mostrare i **bug**, ovvero gli errori presenti nel codice e di marcarli. L'utente può così più facilmente individuarli e correggerli.

In sintesi, un IDE è uno strumento che semplifica la scrittura del codice al programmatore e l'IDE di R Studio si presenta come in Figura 2.2 a pagina 8.⁶

A differenza di R, R Studio si presenta sotto forma di interfaccia utente e per funzionare necessita dell'installazione primaria di R sulla macchina di lavoro. Se R è il motore, R Studio ne rappresenta i comandi di navigazione. Quest'ultimo mostra diverse informazioni in un'unica finestra di lavoro, facilitando la comprensione delle analisi effettuate e visualizzando interattivamente i risultati sotto differenti punti di vista: quello del codice sorgente, quello grafico e quello delle variabili create e mantenute in memoria.

⁴Red Hat. *Cos'è un'ambiente di sviluppo integrato (IDE)*. URL: <https://www.redhat.com/it/topics/middleware/what-is-ide>.

⁵Hat, *Cos'è un'ambiente di sviluppo integrato (IDE)*.

⁶Sindhu Selvam. *R-Studio*. URL: <https://datascienceplus.com/introduction-to-rstudio>.

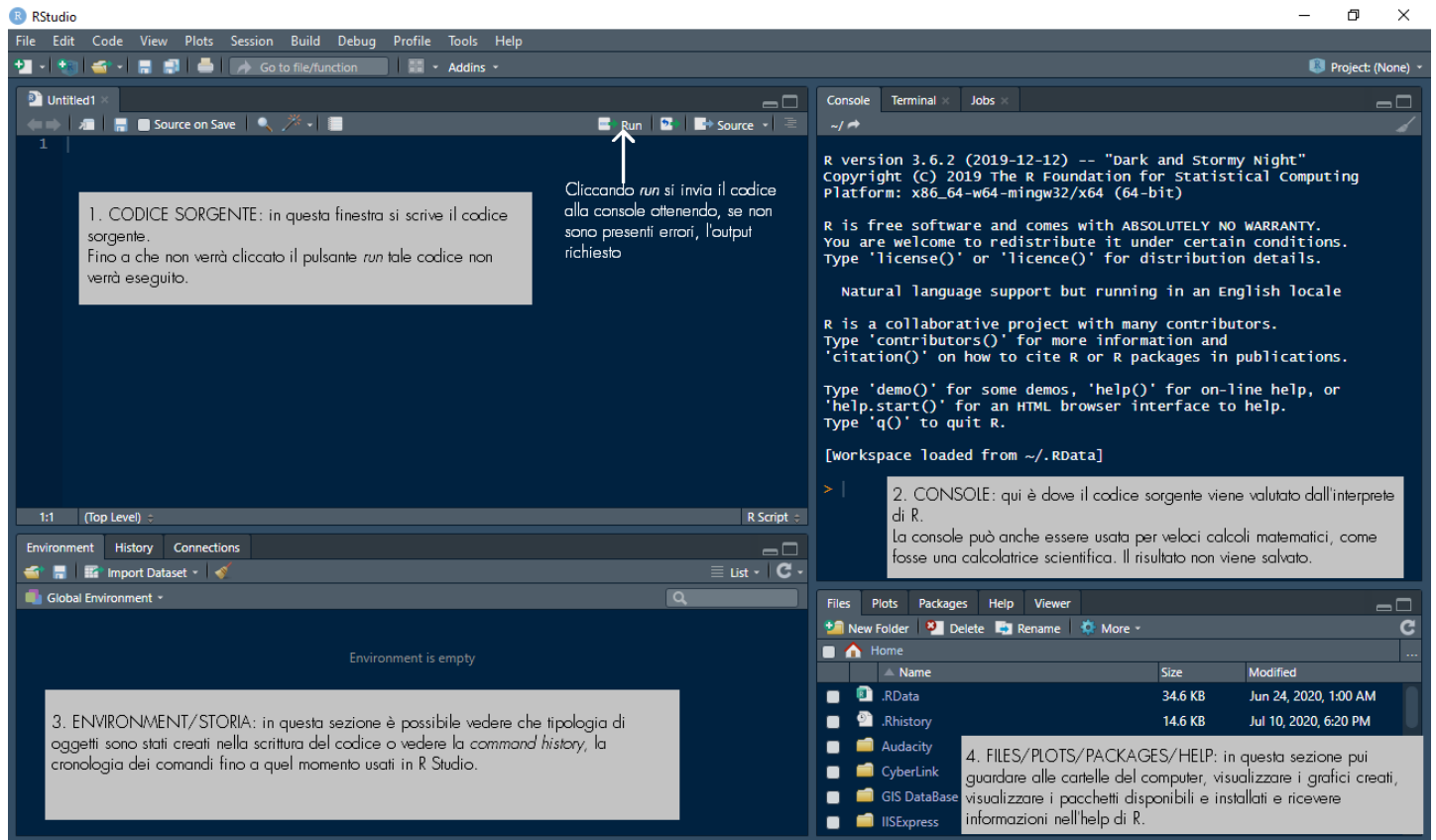


Figura 2.2: Interfaccia di R Studio

2.1.3 Che cos'è R Commander o Rcmdr

R Commander o **Rcmdr** è un'interfaccia grafica (*GUI* o *Graphical User Interface*) per R che presenta la particolarità di poter sfruttare buona parte delle potenzialità di analisi statistica del linguaggio R tramite più semplici menù a tendina. A differenza di R o di R Studio, non è più strettamente necessario conoscere la sintassi del codice sorgente, ma l'interfaccia grafica è basata su un approccio *point-and-click*: tramite menù a discesa e box di dialogo si compiono le analisi che potrebbero svolgersi anche tramite scrittura di codice.

I vantaggi immediatamente percepibili per coloro che non hanno dimestichezza con la programmazione sono:

- **Menù a tendina:** largamente conosciuti e semplici da utilizzare, è sufficiente un click del mouse per operare una semplice analisi o richiamare una funzione;

- **Apprendimento:** è vero che per eseguire una analisi statistica è sufficiente cliccare sul comando corrispondente, ma è altrettanto possibile imparare a scrivere codice utilizzando Rcmdr. Infatti, eseguita l'analisi, oltre all'output statistico, viene fornito il codice sorgente R necessario a generare i risultati: questo permette di visualizzare il *dietro alle quinte* apprendendo quale sia la sintassi del linguaggio;
- **Estensibile:** il pacchetto Rcmdr è in continua espansione. Tramite l'installazione di altri pacchetti è possibile alterare o aumentare i menù stessi e i comandi ivi presenti;
- **Free e Open Source:** come anche R ed R Studio (versione Desktop), Rcmdr è completamente gratuito e *open source*, quindi modificabile ed estendibile da chiunque abbia intenzione di apportare dei cambiamenti o delle migliorie.

In ogni caso, R dovrà essere installato sul proprio computer per poter usufruire della sua interfaccia grafica. Come già sottolineato, si può dire che R sia il motore dentro il quale avvengono le analisi, mentre R Commander è l'intermediario tra il fruitore e il motore stesso, permettendo all'utente di scegliere i comandi appropriati.

R Commander è installabile tramite R (o R Studio) sotto forma di *pacchetto* esterno richiamabile grazie a poche righe di codice che verranno mostrate successivamente.

Per avere un esempio di come R Commander si presenti, si prendano a riferimento la Figura 2.3 a pagina 10 e la Figura 2.4 a pagina 11.

In sintesi, l'utilizzo di Rcmdr semplifica notevolmente il lavoro di analisi dei dati dell'utente inesperto nella scrittura del codice in R grazie al suo utilizzo tramite menù a tendina e un'interfaccia grafica essenziale, ma completa.

2.2 Installazione del pacchetto

Inizialmente pensato per la statistica di base, il pacchetto Rcmdr ha subito una notevole espansione continuativa dal 2002 in poi, nel momento in cui John Fox, principale autore, ne ha iniziato lo sviluppo. L'attuale versione di **Rcmdr** contiene circa 15 000 linee di codice in R e circa 40 *plug-in* per aumentarne le funzionalità.

Una volta aperto R, l'installazione di **Rcmdr** si presenta immediatamente eseguibile tramite il comando:

```
install.packages("Rcmdr")
```

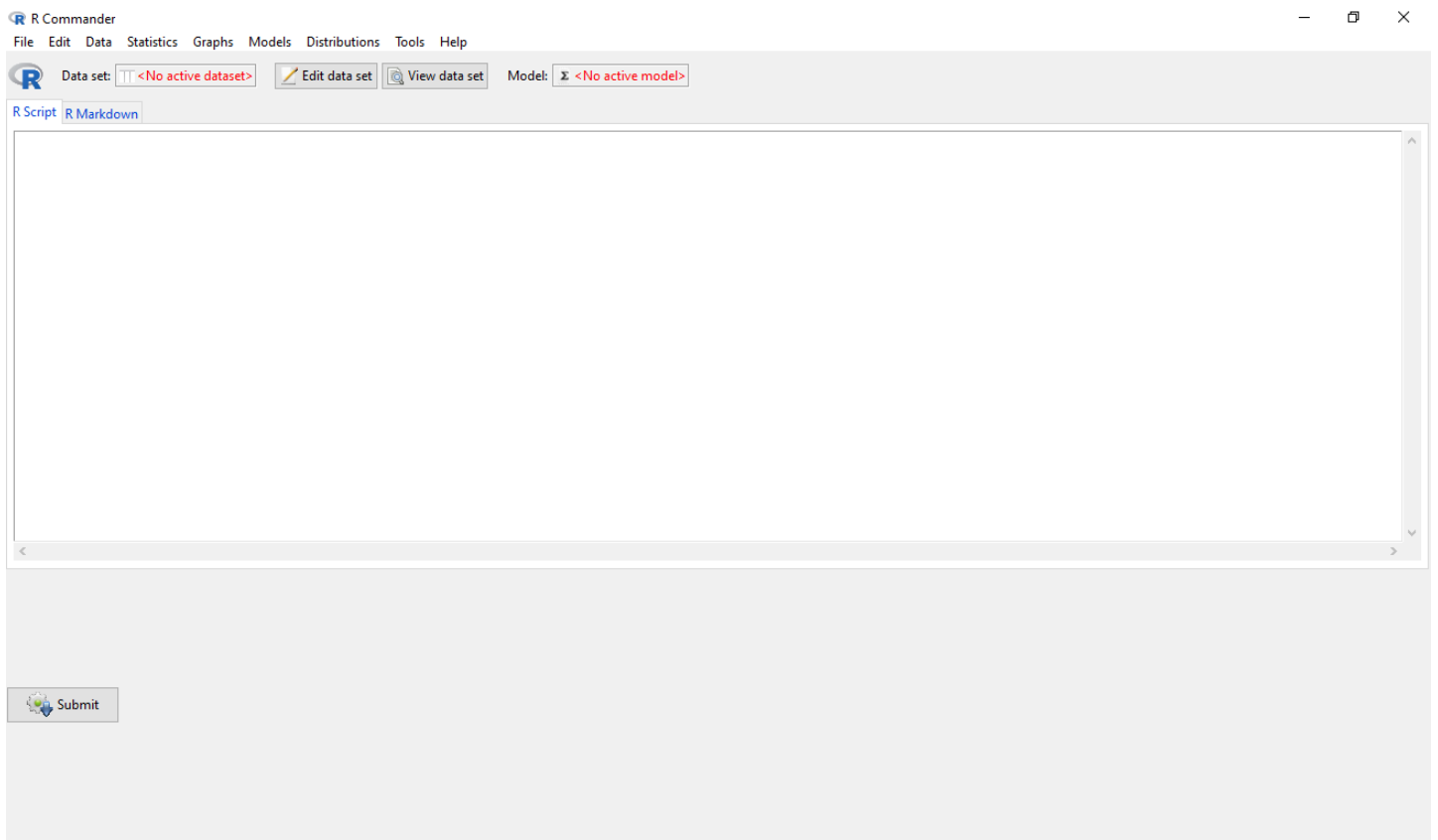


Figura 2.3: Interfaccia di R Commander

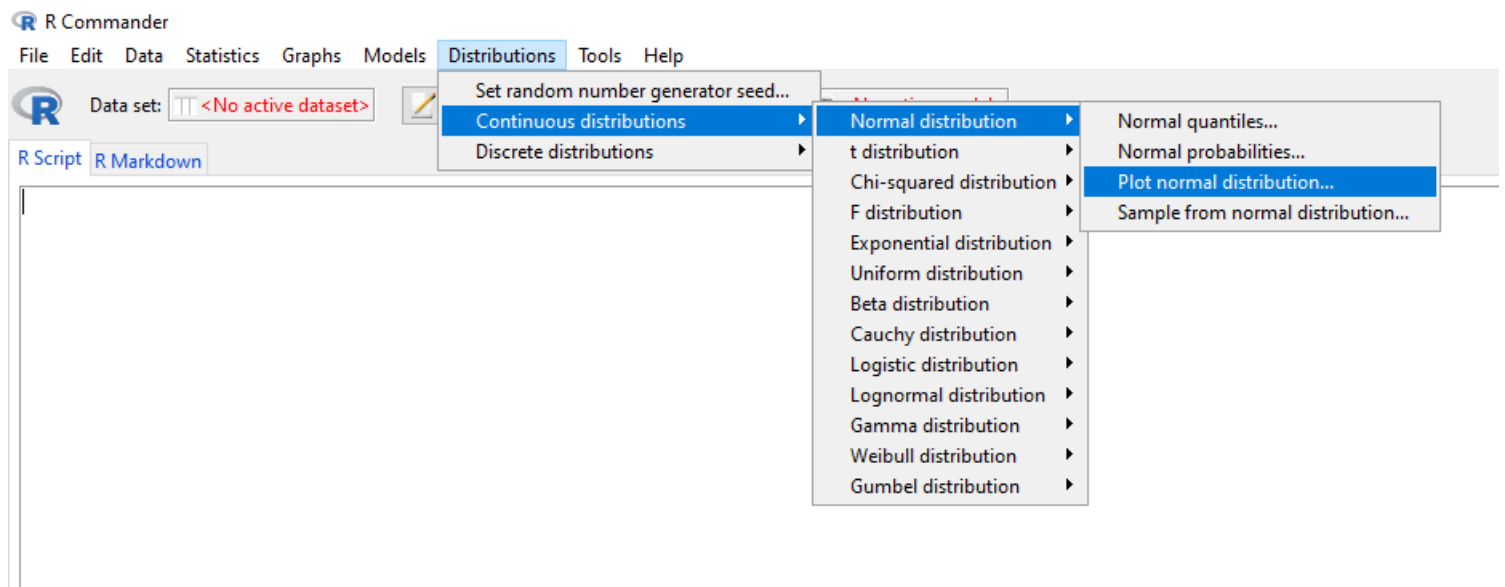


Figura 2.4: Menù a tendina

Si richiede ad R di scaricare ed installare tramite il sito *CRAN* (*Comprehensive R Archive Network*⁷ - sito web che incorpora al suo interno una vasta quantità di pacchetti per aumentare le funzionalità di R) il pacchetto **Rcmdr**.

Visivamente ciò avviene come mostrato in Figura 2.5 a pagina 13. Con il parametro *dependencies = TRUE* si fa sapere ad R di installare anche tutti quei pacchetti aggiuntivi necessari a far funzionare completamente Rcmdr. Non eseguendo tale parametro si potrebbe incorrere nel problema che, mancando alcune estensioni, Rcmdr si presenti mancante di diverse funzionalità.

A questo punto, sarà sufficiente caricare il pacchetto installato tramite il comando:

```
library(Rcmdr)
```

Ovvero, come visualizzato in Figura 2.6 a pagina 14.

L'output che si otterrà sarà l'apertura dell'interfaccia di Rcmdr, pronta per essere utilizzata in tutta la sua praticità. [Figura 2.7, pagina 15]

2.3 Aspetto e funzionalità

La barra dei menù presente in alto alla schermata di R Commander contiene le funzioni utilizzabili tramite la GUI⁸. Di seguito, si analizzano brevemente quali sono le caratteristiche di ogni menù presente. [Figura 2.8, pagina 16]

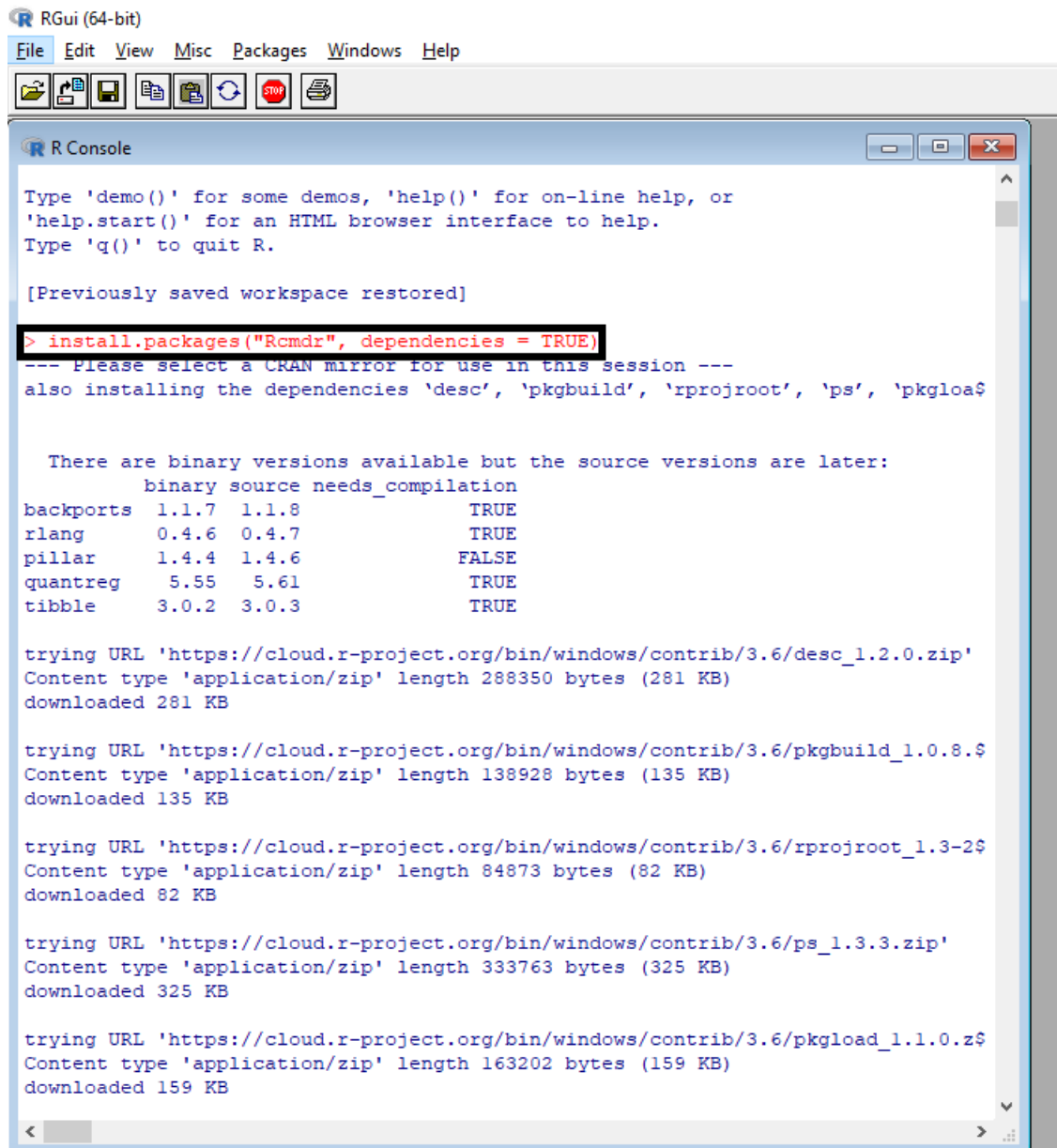
File: il menù *file* contiene elementi per aprire e salvare i vari tipi di file, per modificare la *working directory* di R (ovvero, la cartella di lavoro nella quale si stanno salvando i propri file).

Edit: il menù *edit* contiene elementi per modificare il testo come, ad esempio l'elemento *Copia e Incolla*, ma anche elementi specifici per i documenti in R Markdown (di cui si parlerà successivamente nel corso dell'elaborato).

Data: il menù *data* contiene elementi e sottomenù per importare, esportare e manipolare i dati.

⁷CRAN. URL: <https://cran.r-project.org/index.html>.

⁸Fox, *Using the R commander: A point-and-click interface for R*, p. 20.



```

RGui (64-bit)
File Edit View Misc Packages Windows Help

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> install.packages("Rcmdr", dependencies = TRUE)
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies 'desc', 'pkgbuild', 'rprojroot', 'ps', 'pkgload'

There are binary versions available but the source versions are later:
      binary source needs_compilation
backports  1.1.7  1.1.8             TRUE
rlang      0.4.6  0.4.7             TRUE
pillar     1.4.4  1.4.6             FALSE
quantreg   5.55   5.61             TRUE
tibble     3.0.2  3.0.3             TRUE

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.6/desc_1.2.0.zip'
Content type 'application/zip' length 288350 bytes (281 KB)
downloaded 281 KB

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.6/pkgbuild_1.0.8.$
Content type 'application/zip' length 138928 bytes (135 KB)
downloaded 135 KB

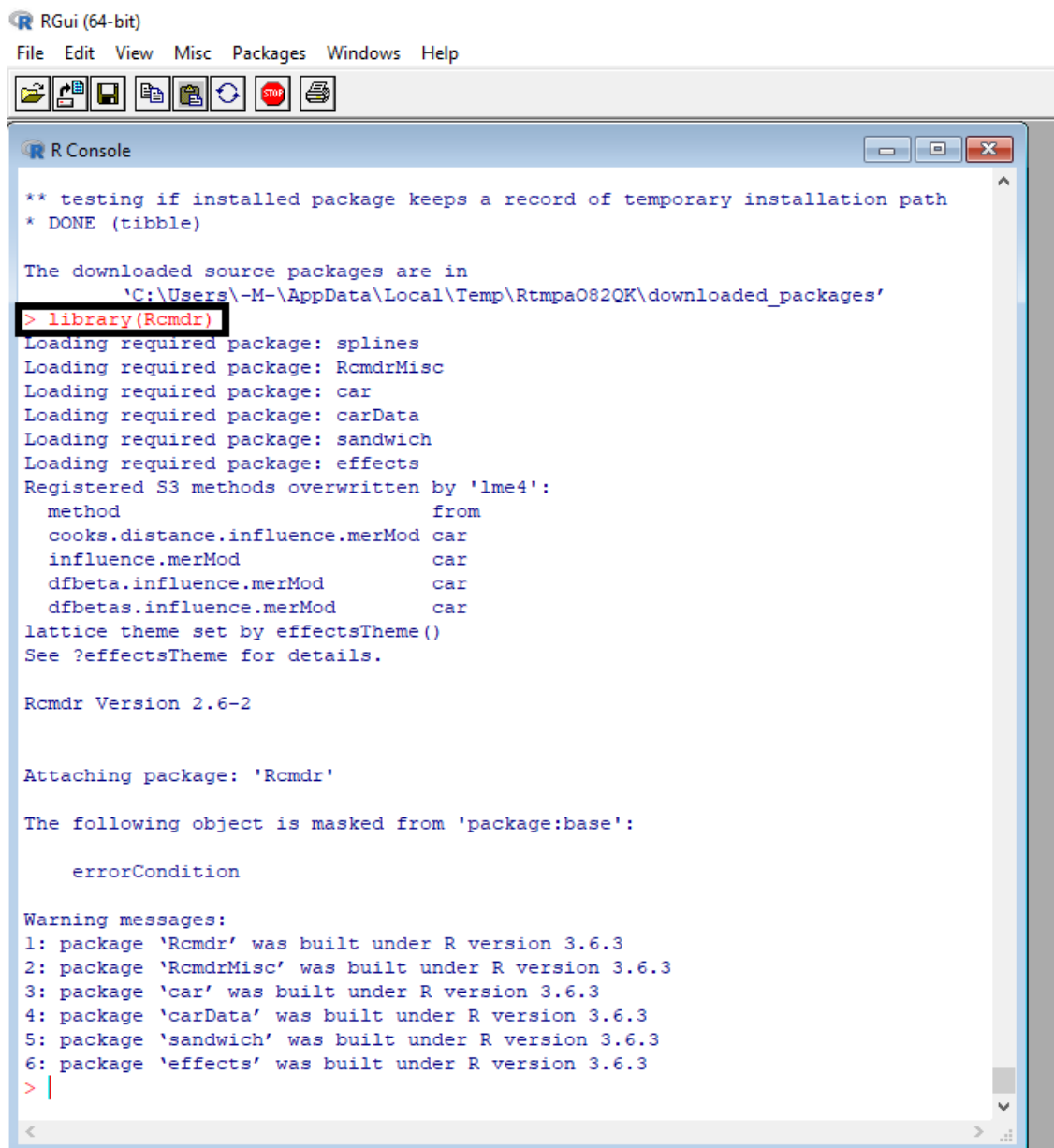
trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.6/rprojroot_1.3-2$
Content type 'application/zip' length 84873 bytes (82 KB)
downloaded 82 KB

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.6/ps_1.3.3.zip'
Content type 'application/zip' length 333763 bytes (325 KB)
downloaded 325 KB

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.6/pkgload_1.1.0.z$
Content type 'application/zip' length 163202 bytes (159 KB)
downloaded 159 KB

```

Figura 2.5: Rcmdr installazione



```

RGui (64-bit)
File Edit View Misc Packages Windows Help

R Console

** testing if installed package keeps a record of temporary installation path
* DONE (tibble)

The downloaded source packages are in
'C:\Users\~M-\AppData\Local\Temp\Rtmpa082QK\downloaded_packages'
> library(Rcmdr)
Loading required package: splines
Loading required package: RcmdrMisc
Loading required package: car
Loading required package: carData
Loading required package: sandwich
Loading required package: effects
Registered S3 methods overwritten by 'lme4':
  method                      from
  cooks.distance.influence.merMod car
  influence.merMod             car
  dfbeta.influence.merMod      car
  dfbetas.influence.merMod     car
lattice theme set by effectsTheme()
See ?effectsTheme for details.

Rcmdr Version 2.6-2

Attaching package: 'Rcmdr'

The following object is masked from 'package:base':

  errorCondition

Warning messages:
1: package 'Rcmdr' was built under R version 3.6.3
2: package 'RcmdrMisc' was built under R version 3.6.3
3: package 'car' was built under R version 3.6.3
4: package 'carData' was built under R version 3.6.3
5: package 'sandwich' was built under R version 3.6.3
6: package 'effects' was built under R version 3.6.3
> |

```

Figura 2.6: Comando library()

Statistics: il menù *statistics* contiene sottomenù per differenti analisi statistiche dei dati, incluse le applicazioni di modelli ai dati.

Graphs: il menù *graphs* contiene elementi e sottomenù per creare classici grafici statistici.

Models: il menù *models* contiene elementi e sottomenù per realizzare diverse operazioni su modelli statistici che sono stati utilizzati per approssimare i dati.

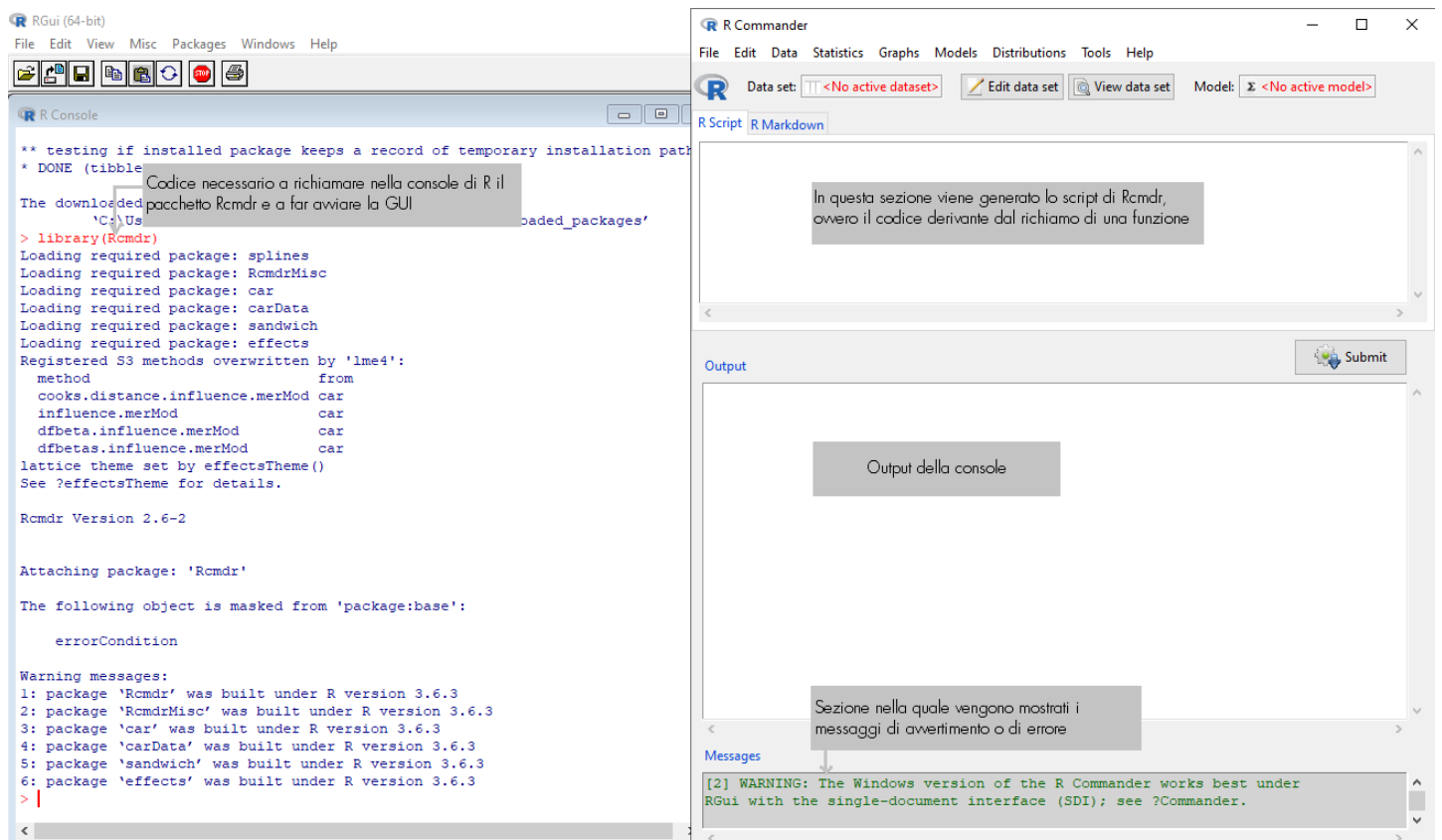


Figura 2.7: Rcmdr GUI

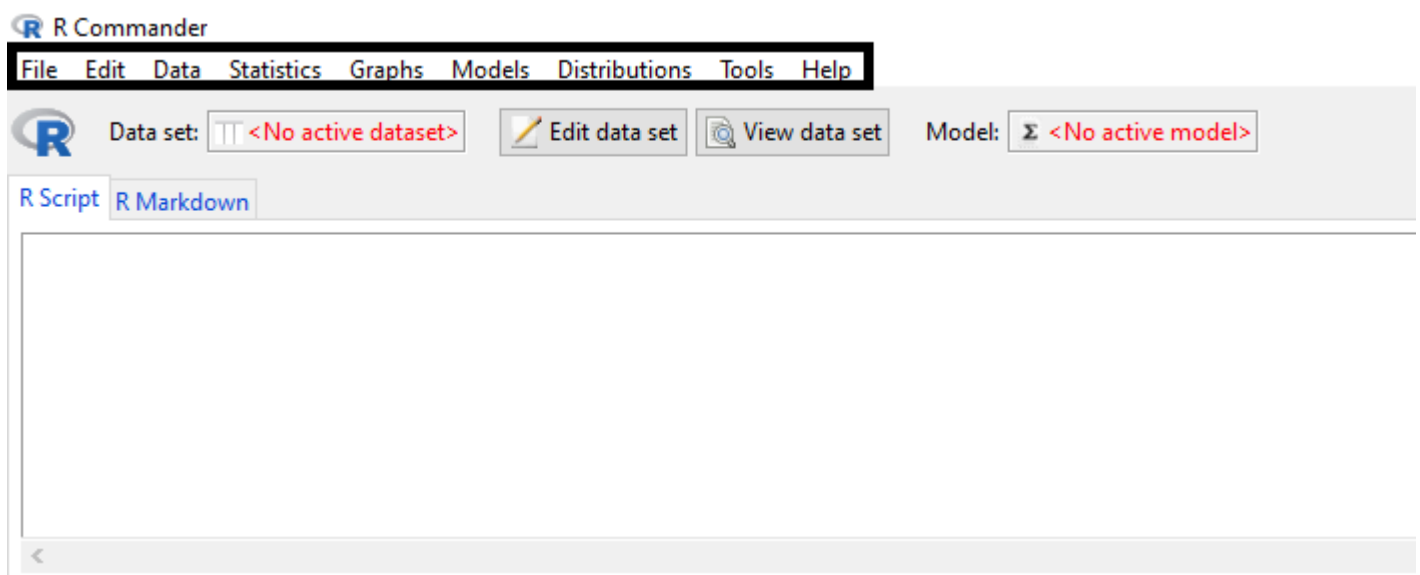


Figura 2.8: Menù GUI

Distributions: il menù *distributions* contiene gli elementi per impostare un seme casuale per le simulazioni, sottomenù per calcolare, ottenere grafici e campionare da una varietà di distribuzioni statistiche.

Tools: il menù *tools* contiene elementi per caricare i pacchetti di R ed R Commander, per impostare e salvare le opzioni di R Commander o per installare software aggiuntivo.

Help: il menù *help* contiene elementi per ottenere informazioni su R Commander ed R, il link ad un breve manuale introduttivo e ai siti di R Commander ed R; informazioni riguardo il data set attivo e collegamenti alle pagine web con istruzioni dettagliate per usare R Markdown nella creazione di rapporti.

Al di sotto dei menù appena presentati, si trova la *toolbar*⁹, ovvero la *barra degli strumenti* nella quale è presente un bottone cliccabile (*Data set*) in cui viene mostrato il data set attualmente attivo, ed un bottone (*Model*) che mostra i modelli statistici attualmente attivi (se Rcmdr viene aperto per la prima volta i bottoni presentano la dicitura *<No active dataset>* o *<No active model>*). [Figura 2.9, pagina 18]

Tali elementi possono essere utilizzati per scegliere fra diversi data sets e differenti modelli rispetto a quelli già attualmente caricati nella memoria di R.

Al di sotto della *toolbar* si hanno due *tabs* definite, rispettivamente, **R Script** e **R Markdown**: durante la sessione di lavoro, ogni comando inserito viene registrato da entrambe le *schede* che sono, allo stesso tempo, modificabili e salvabili. [Figura 2.10, pagina 18]

Inoltre, i comandi salvati nello *Script* di Rcmdr sono manipolabili e, una volta raggiunta una certa dimestichezza nel linguaggio R, è possibile scrivere direttamente il proprio codice ed eseguirlo tramite il pulsante *Submit*, in basso a destra della finestra dello *Script*.

La *tab R Markdown*, registra i comandi imputati durante la sessione di lavoro in un documento dinamico che è possibile modificare per creare un *report* del lavoro svolto. Come sottolineato in precedenza, tale possibilità verrà presentata e spiegata successivamente.

Dopo la mascherina contenente lo Script di R, si trova il pannello dell'*Output*, il quale raccoglie i comandi di R che vengono generati da R Commander insie-

⁹Fox, *Using the R commander: A point-and-click interface for R*, p. 22.

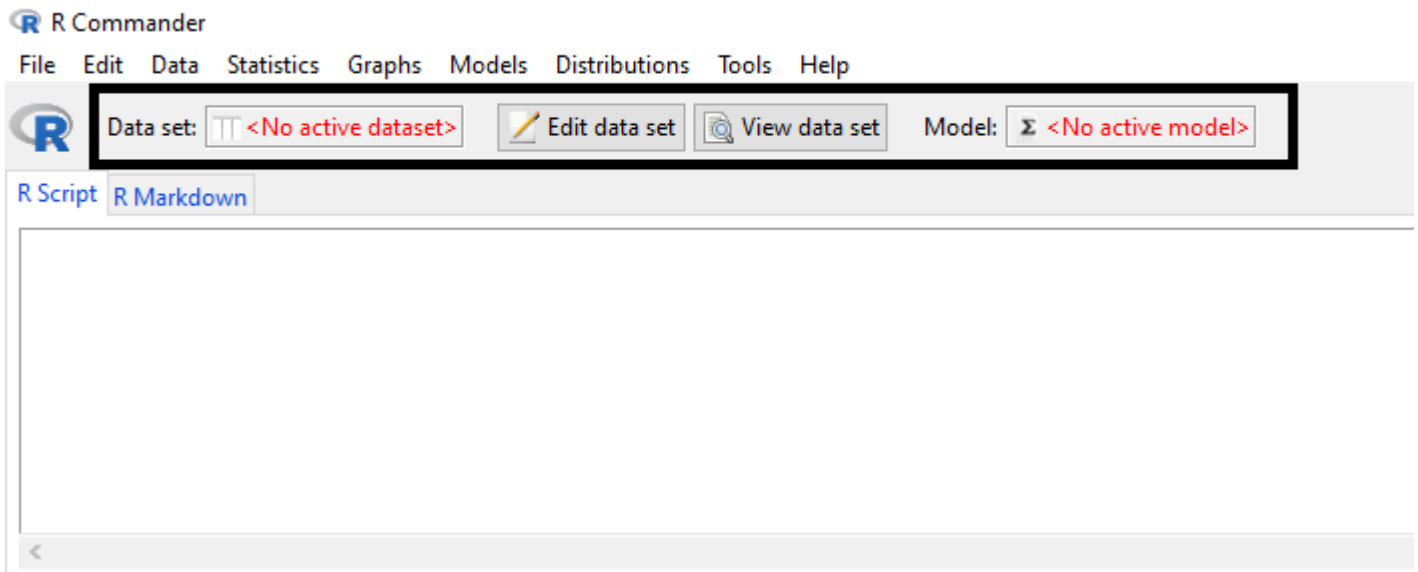


Figura 2.9: Toolbar

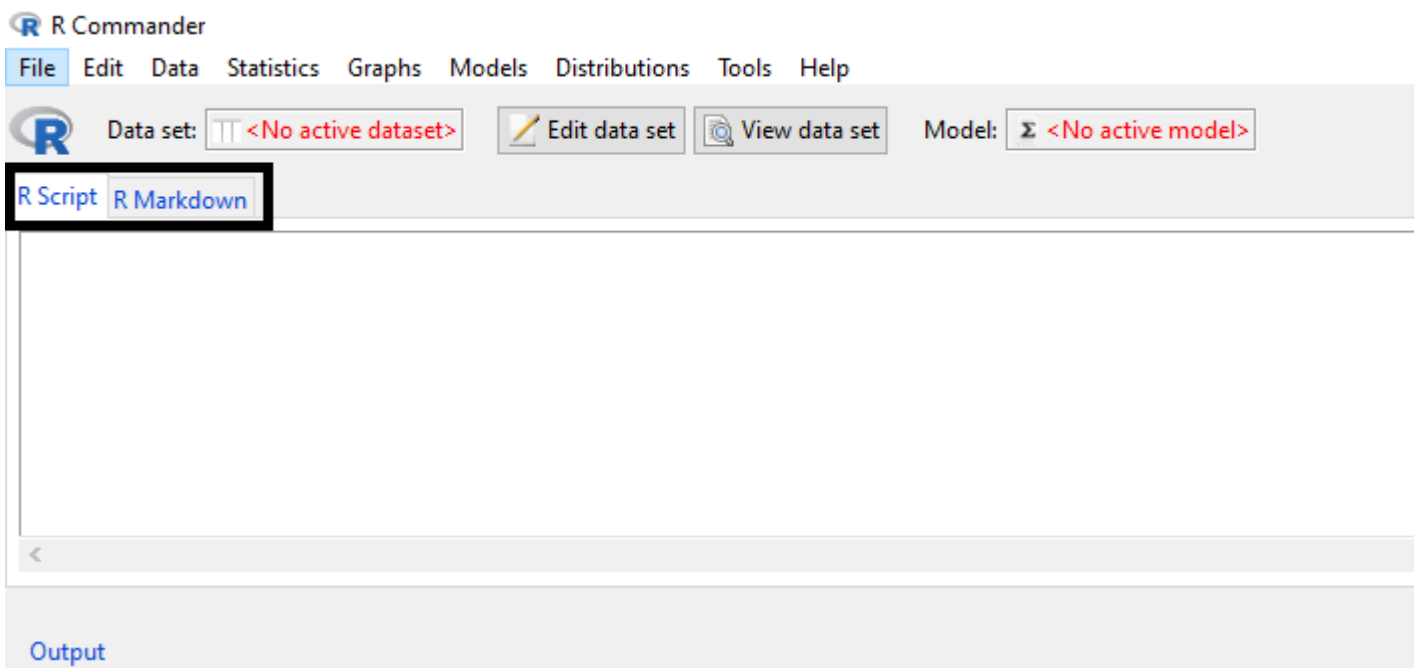


Figura 2.10: Script-Markdown

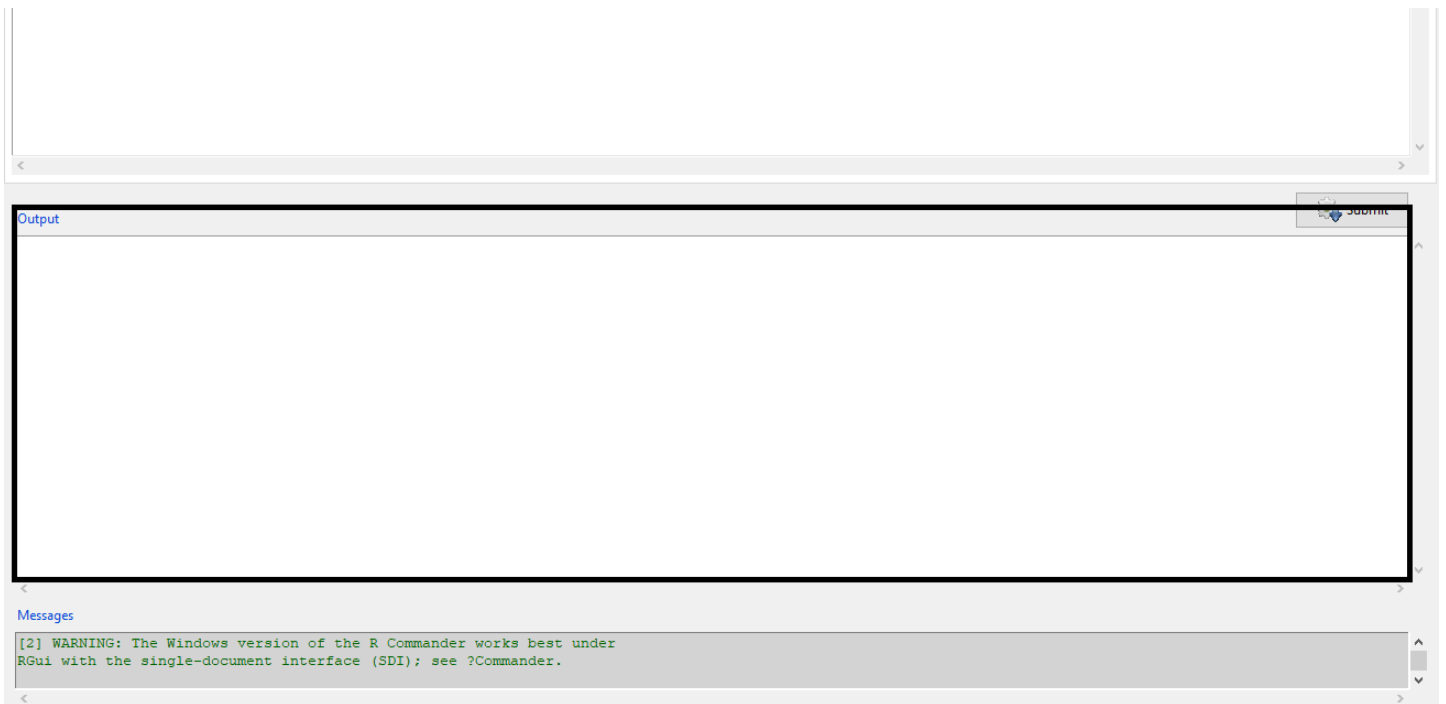


Figura 2.11: Pannello Output

me ai risultati ottenuti dall'Output stesso. Anche in questo caso, è possibile modificare il risultato oppure copiarlo e incollarlo se si ritiene utile farlo. [Figura 2.11, pagina 19]

Infine, al fondo della finestra di R Commander, è presente il pannello *Messages*, il quale registra i messaggi generati da R ed R Commander, numerati e colorati secondo una gerarchia, dai meno importanti (*notes*: in blu scuro) ai più importanti (*warnings*: in verde; *error messages*: in rosso). [Figura 2.12, pagina 20]

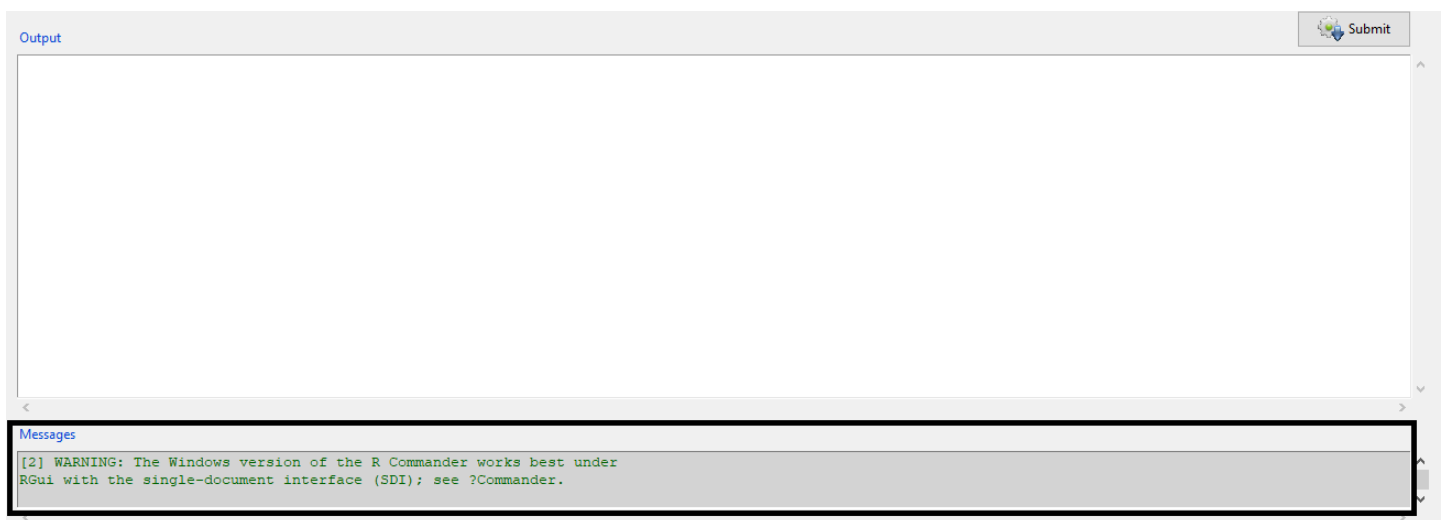


Figura 2.12: Pannello Messaggi

Capitolo 3

TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA

3.1 Storia e struttura, obiettivo dell'analisi e metodologie applicabili

Il 10 aprile 1912, alle 12.15 p.m.¹⁰ il transatlantico *Titanic* salpò da Southampton nel sud dell'Inghilterra, in direzione di New York con 2223¹¹ persone a bordo, di tutte le età e divise in tre classi passeggeri: prima, seconda e terza classe.¹²

Il 14 aprile il transatlantico ricevette tre messaggi di avvertimento da diverse altre imbarcazioni che avevano avvistato o avevano evitato alcuni iceberg sulla stessa rotta del Titanic. Non venne preso nessun provvedimento per evitare l'imminente pericolo.

Alle 11.46 p.m. dello stesso giorno, il Titanic urtò violentemente il ghiaccio. La prua subì i maggiori danni, iniziando ad imbarcare acqua e ad affondare.

Delle 2223 persone a bordo, persero la vita 1517, mentre le rimanenti 706 trovarono la salvezza. Del totale dei passeggeri di prima classe se ne salvò il 60%, il 42% di coloro che erano in seconda classe e solamente il 25% delle persone in terza classe. Infine, si salvò il 24% del totale dell'equipaggio.

¹⁰Committee on Commerce United States Senate. *Titanic Disaster*. 1912. URL: <https://www.senate.gov/artandhistory/history/resources/pdf/TitanicReport.pdf>.

¹¹CNN. *Titanic Fast Facts*. 2020. URL: <https://edition.cnn.com/2013/09/30/us/titanic-fast-facts/index.html>.

¹²Encyclopedia Titanica. *RMS Titanic: An Introduction*. 1996-2020. URL: <https://www.encyclopedia-titanica.org/titanic/>.

Il data set corrispondente consta di 8 variabili ed è stato scaricato gratuitamente dai archivi online dell'Università di Stanford.¹³

Per facilitare il riconoscimento immediato delle variabili presenti, i nomi delle stesse sono stati tradotti in italiano e le variabili ordinate a partire dai nomi dei passeggeri. Sebbene questi ultimi fossero, come precedentemente ricordato 2223, il data set scelto è composto da 887 record in quanto i dati dei restanti passeggeri non furono mai registrati all'atto della partenza.

Per dimostrare le potenzialità e la semplicità di utilizzo di **Rcmdr**, si procederà con l'analisi statistica del data set visualizzandone i risultati. A completamento di una analisi esplorativa dei dati, si applicherà il modello di regressione logistica per operare una previsione sulla sopravvivenza o meno dei passeggeri del Titanic. Di conseguenza, la variabile **Sopravvissuto/a** verrà usata come variabile da spiegare, mentre le restanti saranno considerate come variabili esplicative (o regressori) del fenomeno in esame.

3.2 Analisi esplorativa del data set (EDA: Exploratory Data Analysis)

La finalità del presente capitolo è quella di mostrare le potenzialità di R Commander nell'esplorazione dei dati e nell'applicazione di una metodologia predittiva. L'*EDA* o *Exploratory Data Analysis* si prefigge l'obiettivo di spiegare i dati utilizzando gli strumenti della statistica descrittiva, sia sotto forma di indici di sintesi statistici, sia di appropriati grafici. Senza alcuna ipotesi sul data set, esso viene analizzato con l'intento di identificare le tipologie di appartenenza dei dati (quantitativi o qualitativi), eventuali mancanze o errori in fase di *data entry*, e comprendere possibili relazioni esistenti fra le variabili in studio. In generale, se a priori non si conosce la composizione del data set, solo successivamente si procede con ipotesi e assunzioni su quali possano essere i migliori strumenti per modellizzare i dati.

Una volta aperto R e aver richiamato il pacchetto **Rcmdr** per iniziare l'analisi dei dati, si deve caricare il data set scelto. Uno dei modi per farlo prevede di cliccare nella tab *Data* presente nella barra dei menù, successivamente su *Import data* e infine su *from text file, clipboard or URL...* nel caso in cui si voglia il caricamento di un data set salvato in *.csv*, altrimenti si potrà scegliere l'opzione *From Excel file...*, come mostrato in Figura 3.1 a pagina

¹³Stanford University. *titanic-dataset*. URL: <https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv>.

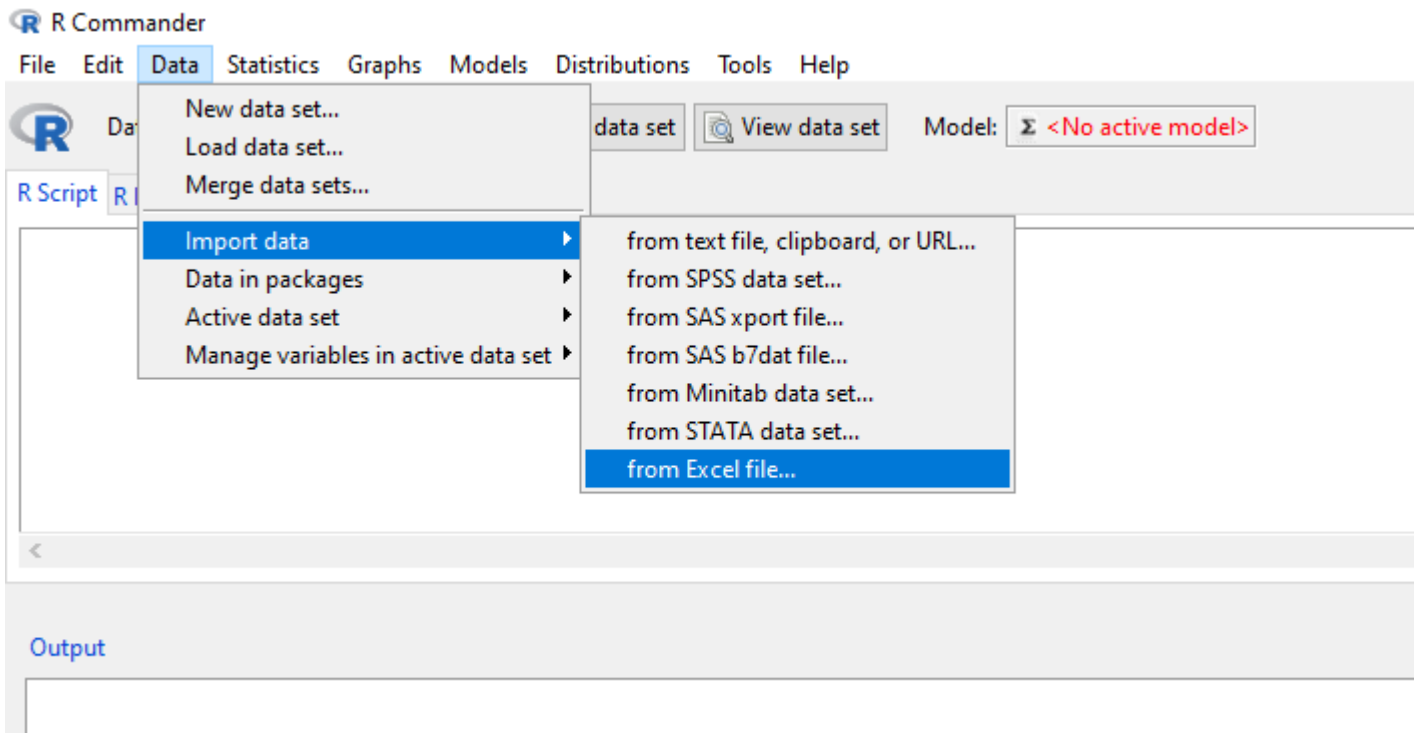


Figura 3.1: Import data da Excel

23. Verrà aperta una finestra di dialogo nella quale sarà possibile impostare alcuni importanti parametri, come mostrato in Figura 3.2 a pagina 24.

È buona prassi modificare fin da subito il nome del data set dal generico *Data set* che compare nel primo riquadro in bianco ad un valore più significativo come, ad esempio, *Titanic*. Poiché il data set racchiude anche i nomi delle variabili, si lascerà invariato il primo *checkbox*; inoltre, nel caso mancassero elementi all'interno delle variabili, potrebbero essere indicati con la sigla *NA* (*Not Available*).

Cliccando sul pulsante **OK**, si aprirà una finestra nella quale ricercare il data set sul proprio computer per poi selezionarlo e cliccare sul pulsante *Open*, come in Figura 3.3 a pagina 25.

L'output di tale azione è visibile in Figura 3.4 a pagina 26.

Nello *Script* viene quindi riportato il codice di R corrispondente al caricamento del data set, mentre nella maschera dell'*Output* vengono raccolti i risultati ottenuti. Infine, nella parte inferiore relativa ai *Messages* si nota già una prima importante informazione sui dati: il data set contiene 887 righe e

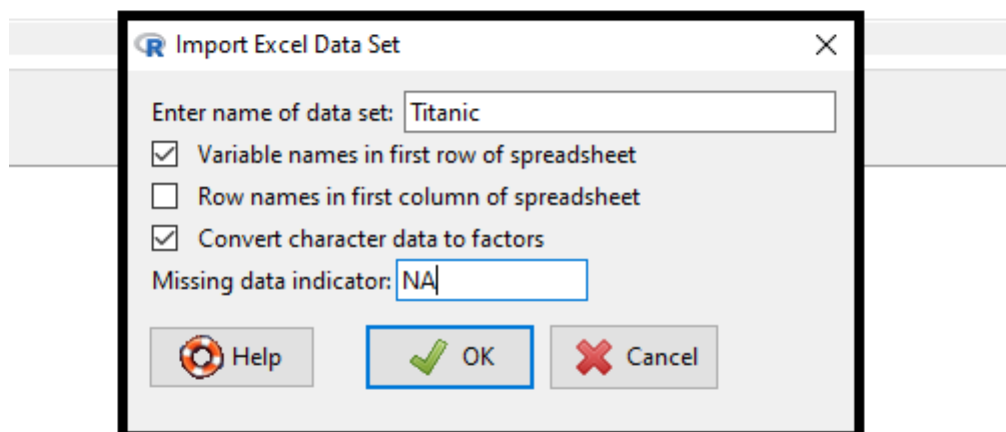


Figura 3.2: Finestra di dialogo

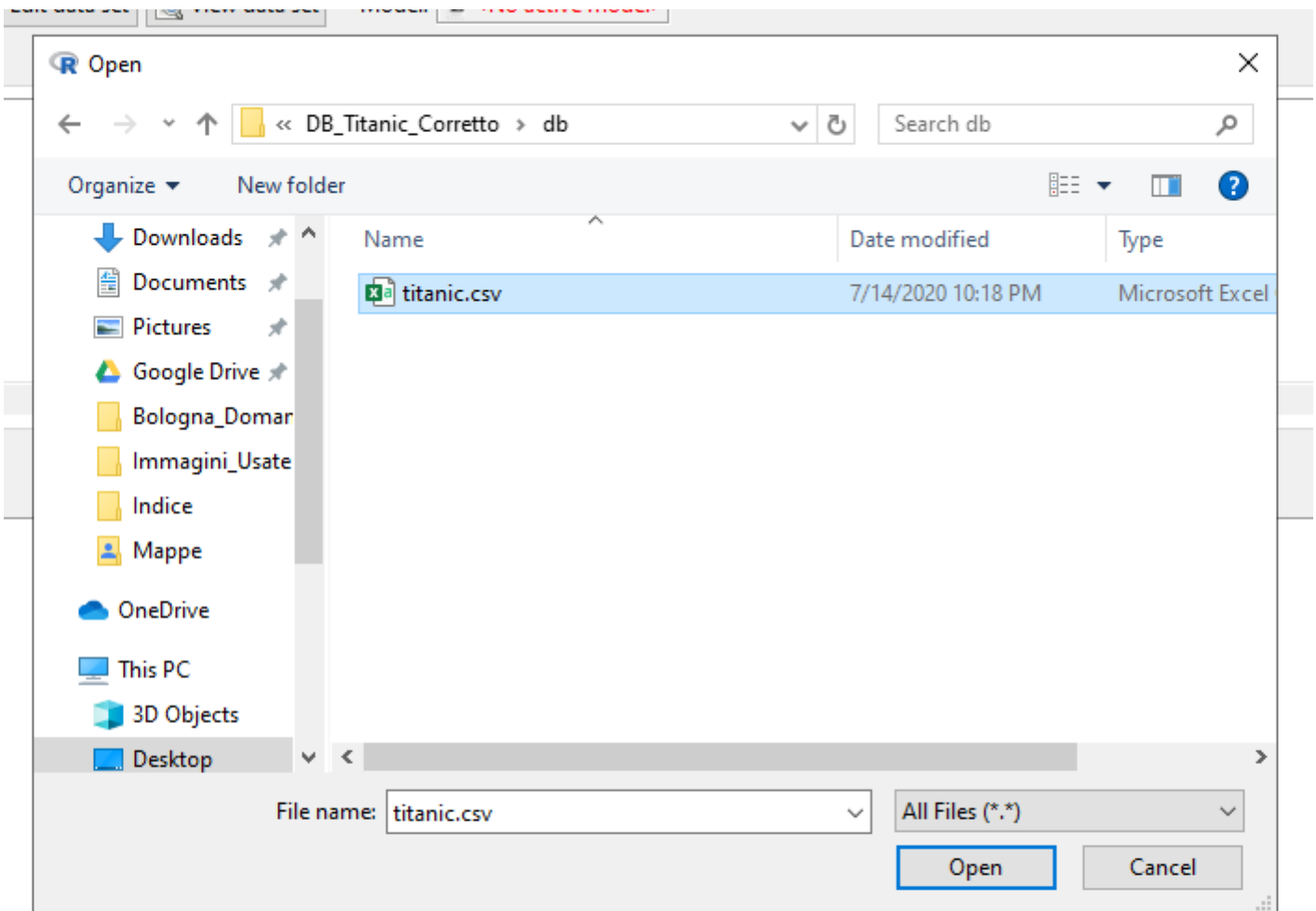


Figura 3.3: Scelta del data set

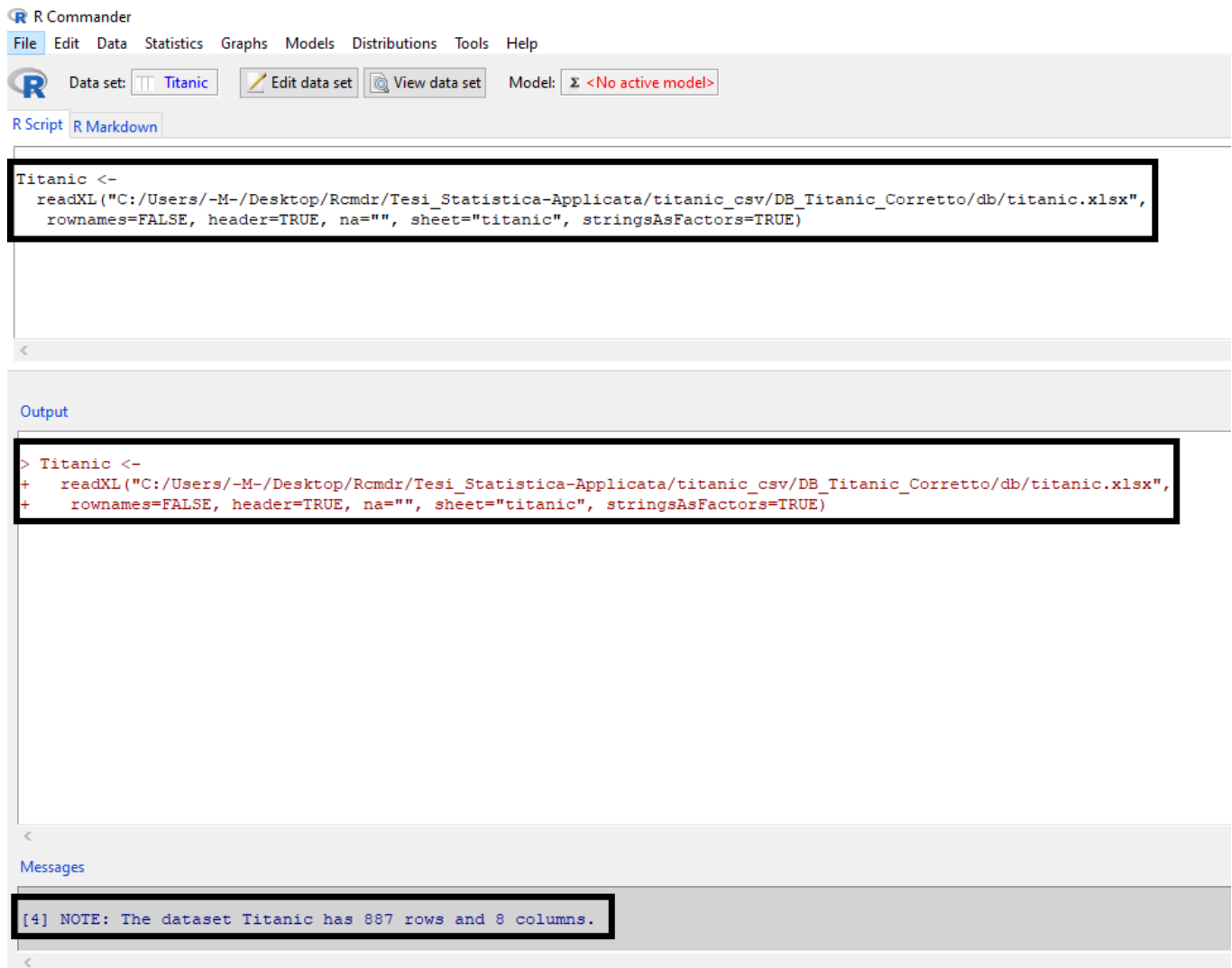
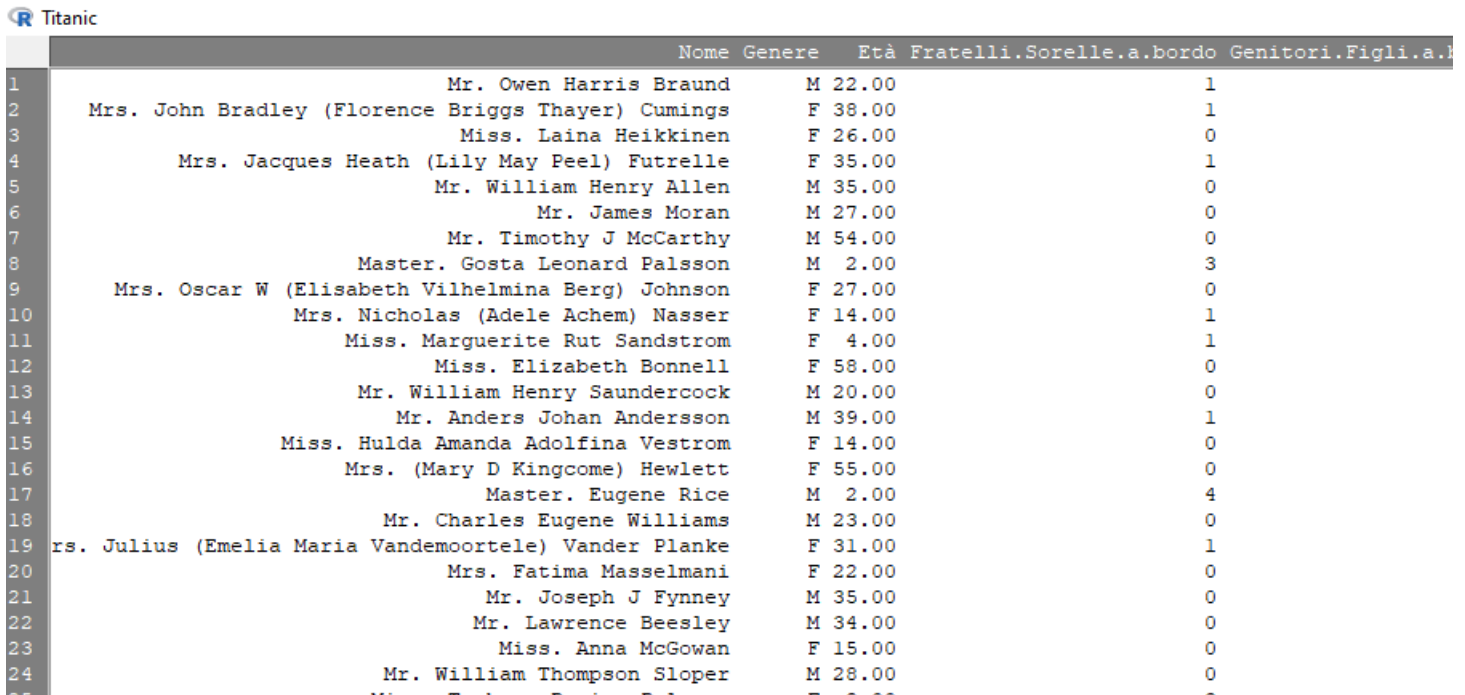


Figura 3.4: Caricamento data set



| | Nome | Genere | Età | Fratelli.Sorelle.a.bordo | Genitori.Figli.a.bordo |
|----|--|--------|-------|--------------------------|------------------------|
| 1 | Mr. Owen Harris Braund | M | 22.00 | 1 | |
| 2 | Mrs. John Bradley (Florence Briggs Thayer) Cumings | F | 38.00 | 1 | |
| 3 | Miss. Laina Heikkinen | F | 26.00 | 0 | |
| 4 | Mrs. Jacques Heath (Lily May Peel) Futrelle | F | 35.00 | 1 | |
| 5 | Mr. William Henry Allen | M | 35.00 | 0 | |
| 6 | Mr. James Moran | M | 27.00 | 0 | |
| 7 | Mr. Timothy J McCarthy | M | 54.00 | 0 | |
| 8 | Master. Gosta Leonard Palsson | M | 2.00 | 3 | |
| 9 | Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johnson | F | 27.00 | 0 | |
| 10 | Mrs. Nicholas (Adele Achem) Nasser | F | 14.00 | 1 | |
| 11 | Miss. Marguerite Rut Sandstrom | F | 4.00 | 1 | |
| 12 | Miss. Elizabeth Bonnell | F | 58.00 | 0 | |
| 13 | Mr. William Henry Saunderson | M | 20.00 | 0 | |
| 14 | Mr. Anders Johan Andersson | M | 39.00 | 1 | |
| 15 | Miss. Hulda Amanda Adolfina Vestrom | F | 14.00 | 0 | |
| 16 | Mrs. (Mary D Kingcome) Hewlett | F | 55.00 | 0 | |
| 17 | Master. Eugene Rice | M | 2.00 | 4 | |
| 18 | Mr. Charles Eugene Williams | M | 23.00 | 0 | |
| 19 | Mrs. Julius (Emelia Maria Vandemoortele) Vander Planke | F | 31.00 | 1 | |
| 20 | Mrs. Fatima Masselmani | F | 22.00 | 0 | |
| 21 | Mr. Joseph J Fynney | M | 35.00 | 0 | |
| 22 | Mr. Lawrence Beesley | M | 34.00 | 0 | |
| 23 | Miss. Anna McGowan | F | 15.00 | 0 | |
| 24 | Mr. William Thompson Sloper | M | 28.00 | 0 | |

Figura 3.5: Visualizzazione data set

8 colonne.

Per visualizzare il data set e controllare quali variabili sono presenti, si clicca sul bottone nella *toolbar* “View data set”, ottenendo quanto presente in Figura 3.5 a pagina 27.

Dalla figura sono visibili 8 variabili in studio che vengono presentate di seguito:

- **Nome:** contenente i nomi di tutti i passeggeri che sono stati registrati prima dell'imbarco;
- **Genere:** carattere qualitativo le cui modalità, misurate in scala nominale, sono “M” ed “F” rispettivamente per “Maschio” e “Femmina”;
- **Età:** carattere quantitativo reale misurato in scala per rapporti, contiene le età dei passeggeri;
- **Fratelli/Sorelle a bordo:** carattere quantitativo discreto misurato in scala per rapporti. Rappresenta la presenza o meno, per ogni passeggero, di fratelli o sorelle (*Siblings*) a bordo del Titanic;

- **Genitori/Figli a bordo:** carattere quantitativo discreto misurato in scala per rapporti. Rappresenta la presenza o meno, per ogni passeggero, di genitori (nel caso dei figli), o di figli (nel caso dei genitori) a bordo del Titanic;
- **Tariffa:** carattere quantitativo reale misurato in scala per rapporti. Rappresenta il costo del biglietto che è stato pagato da ogni singolo passeggero per imbarcarsi;
- **Classe passeggero:** carattere qualitativo ordinale misurato in scala ordinale. Rappresenta la classe (o categoria) nella quale il passeggero si è imbarcato: prima, seconda o terza classe;
- **Sopravvissuto/a:** variabile dicotomica (o binaria), caso particolare di variabile nominale. Le modalità sono rappresentate da 0 ed 1, rispettivamente “Non sopravvissuto” o “Sopravvissuto”.

Nella prima riga sono presenti i nomi di tutte le variabili in studio, mentre nella prima colonna è presente un valore contatore per ogni passeggero.

Volendo comprendere le informazioni principali delle suddette variabili, è sufficiente cliccare la tab *Statistics > Summaries > Active data set* per ottenere i cinque dati di sintesi su ogni singola variabile presente, come in Figura 3.6 a pagina 29.

I risultati sono visualizzati in Figura 3.7 a pagina 29.

Vengono quindi visualizzate tutte le 8 variabili con i 5 valori più importanti, più la media:

- **Min:** il valore minimo presente nella variabile;
- **1st Qu.:** primo quartile di ordine pari a 0.25. Si vuole indicare che almeno il 25% della popolazione presenta un valore del carattere non superiore ad un certo elemento indicato. Ad esempio, se si considera la variabile “Età”, il primo quartile indica che almeno il 25% della popolazione non presenta un’età superiore a 20.25 anni (di conseguenza, il 75% della popolazione presenta un’età superiore a 20.25 anni);
- **Median:** rappresenta la mediana, ovvero un’indice di posizione (come i quartili) che indica il valore centrale della variabile. In generale:
 - Se il numero di dati, ordinati in ordine crescente, è dispari la mediana corrisponde al valore centrale, ovvero al valore che occupa la posizione:

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA 29

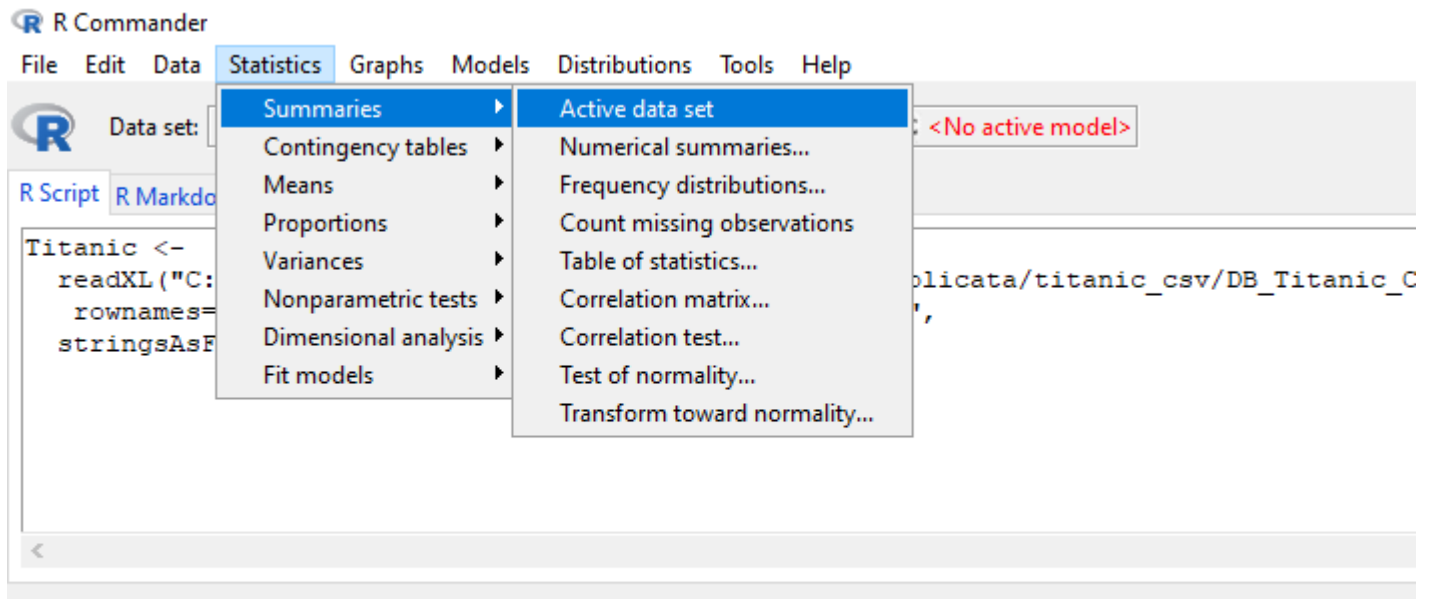


Figura 3.6: Summary data set

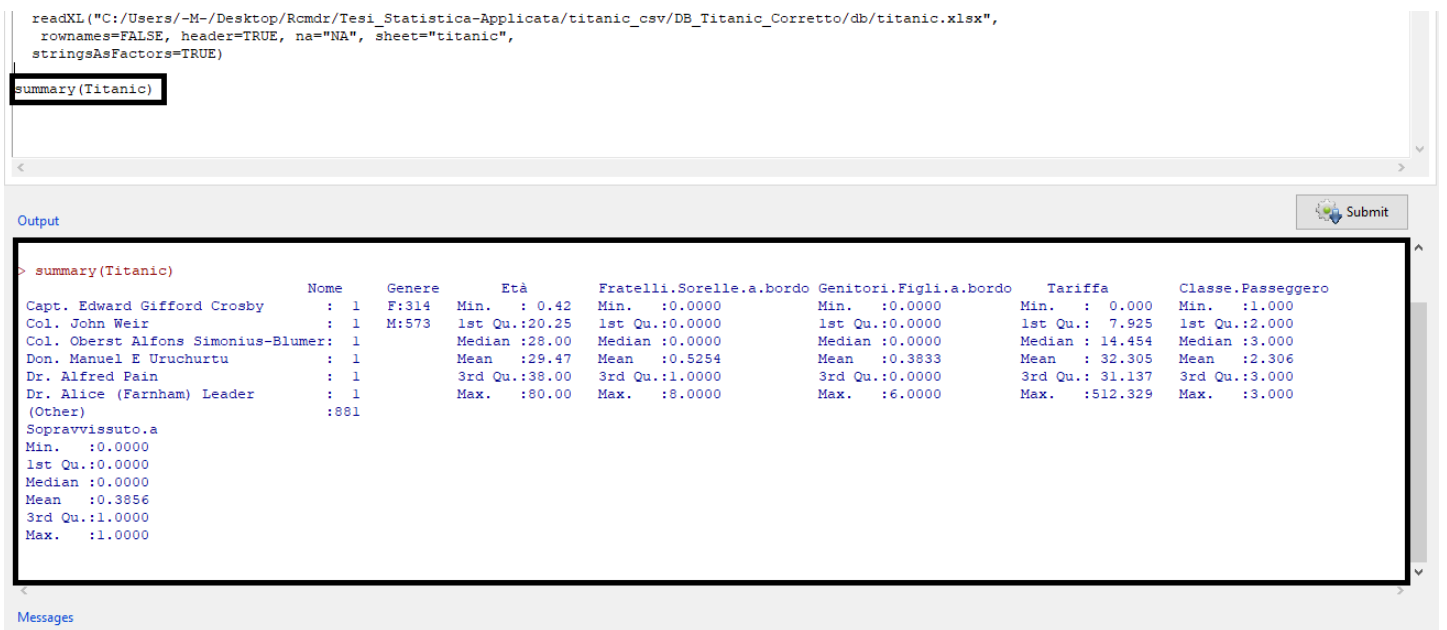


Figura 3.7: Risultato Summary

$$\frac{(n+1)}{2}.$$

- Se il numero di dati, ordinati in ordine crescente, è pari la mediana è calcolata utilizzando i due valori che occupano le posizioni:

$$\frac{n}{2};$$

$$\frac{n}{2} + 1.$$

Generalmente, si sceglie la media aritmetica di questi due valori. Inoltre, la mediana rappresenta anche il secondo quartile di ordine 0.5 che divide la popolazione in due parti con numerosità pari a $\frac{n}{2}$.

- **Mean:** rappresenta la media aritmetica della variabile in studio, generalmente indicata con la seguente formula:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

- **3rd Qu.:** terzo quartile di ordine pari a 0.75;
- **Max:** indica il valore massimo presente nella variabile di riferimento.

È importante ora soffermarsi su alcuni dati del data set in studio:

- Nel data set sono presenti 314 femmine e 573 maschi a bordo del *Titanic*. Sul totale, significa che il 35.40% della popolazione era di genere femminile, mentre il restante 64.60% era di genere maschile;
- L'età presenta un intervallo di variazione compreso fra 0.42 (ad indicare i mesi di vita) e 80.00 anni;
- La variabile *tariffa* ha una media di 32.305 sterline per biglietto ed un valore massimo pari a 512.329 sterline. Poiché il valore massimo è decisamente più grande sia del valore minimo che del terzo quartile come anche della media, è ragionevole pensare che in questa variabile siano presenti *dati anomali*. Essendo la *media* una statistica **non robusta** per valori particolari (ovvero, instabile rispetto a dati atipici), sarà utile anche una rappresentazione tramite un *boxplot* per accertarsi di

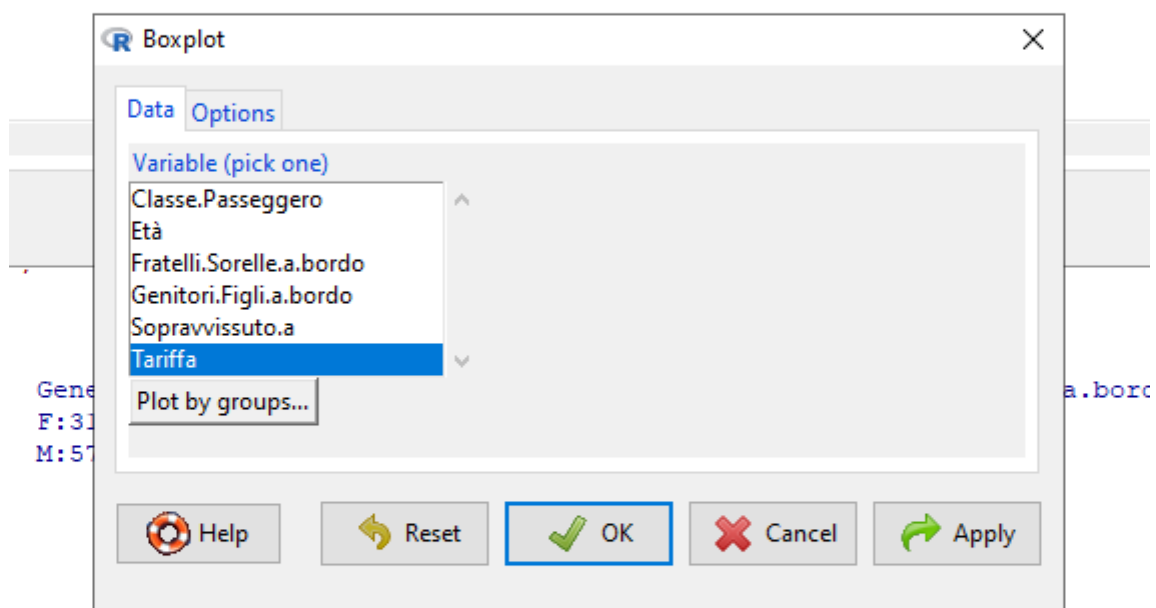


Figura 3.8: Boxplot box di dialogo

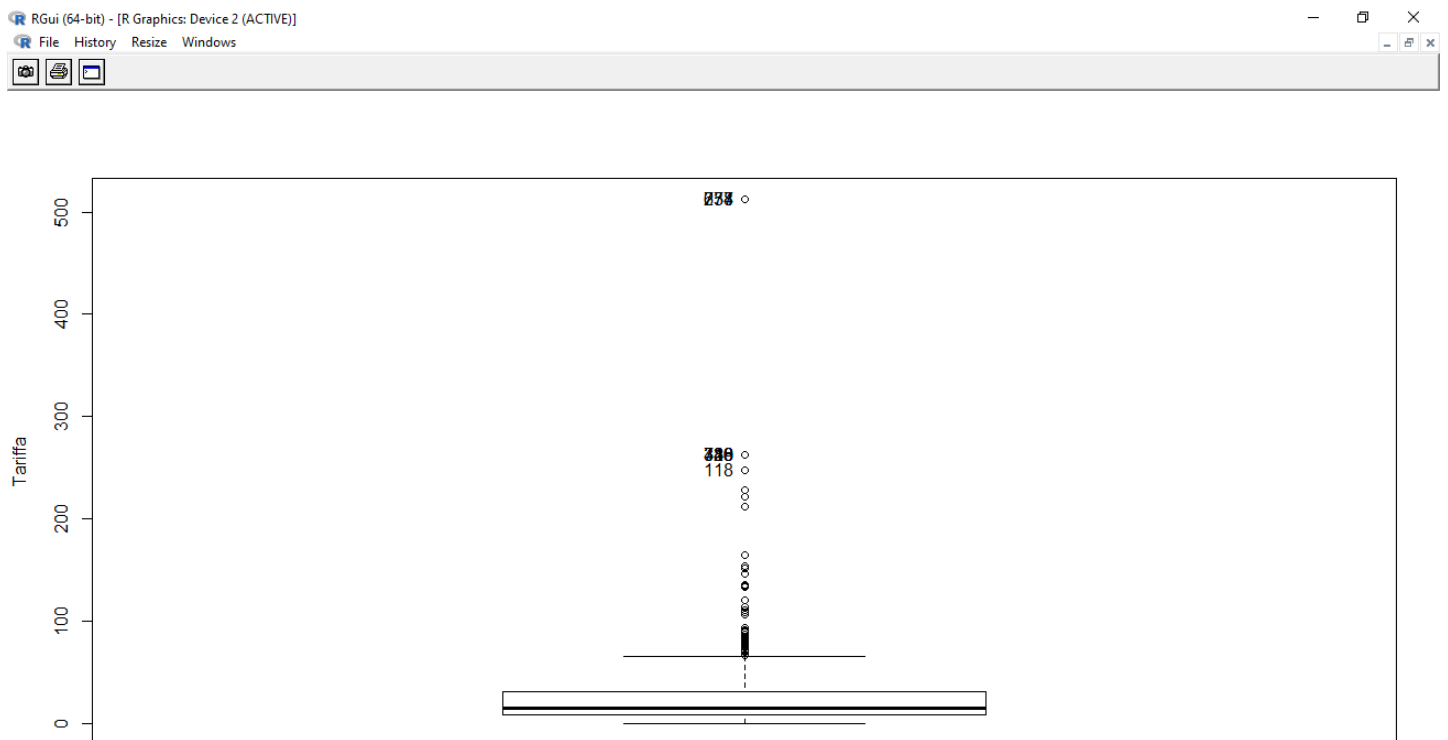


Figura 3.9: Boxplot tariffa

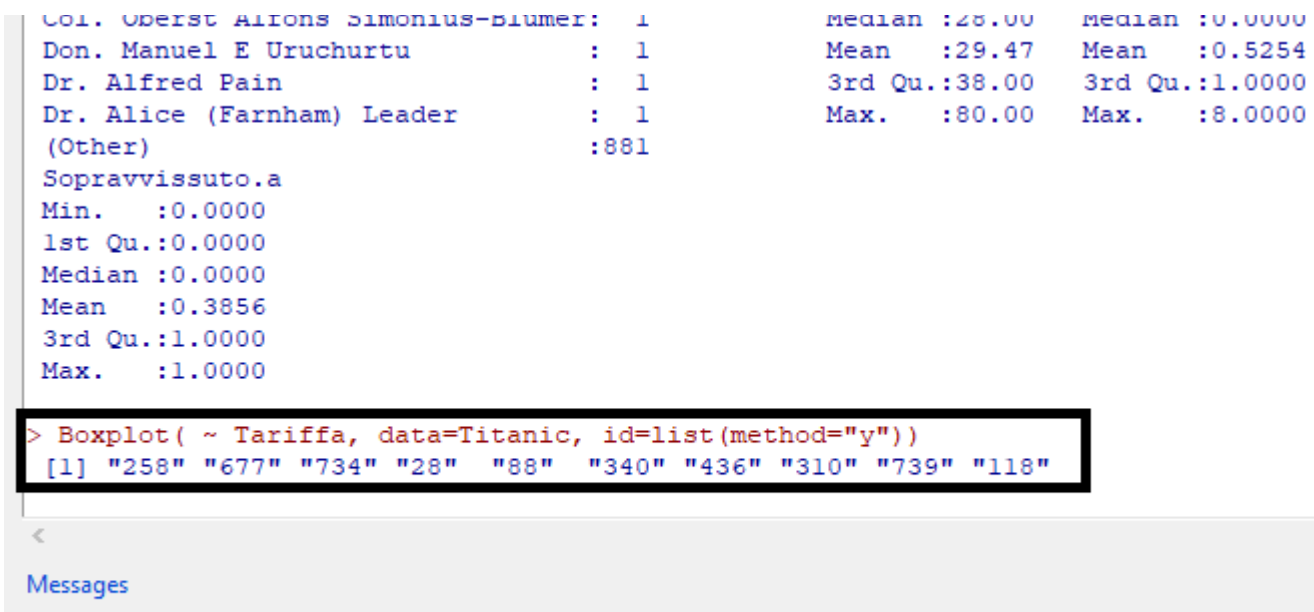


Figura 3.10: Boxplot Outliers

eventuali anomalie presenti. Cliccando sulla tab *Graphs > Boxplot...* e scegliendo dal box di dialogo la voce *Tariffa* come in Figura 3.8 a pagina 31, si ottiene il grafico in Figura 3.9 a pagina 32.

Come è chiaramente visibile, sono presenti non pochi dati anomali che possono distorcere il valore della media “Tariffa”, soprattutto *outlier* di valore alto.

Inoltre, l’Output di R Commander restituisce una informazione aggiuntiva: i valori che sono considerati effettivamente degli *outlier*, relativi al numero del passeggero corrispondente nel data set. [Figura 3.10 a pagina 33].

- Infine, la variabile *Sopravvissuto/a* presenta due sole modalità, 0 e 1: la loro media non ha significato utile ai fini statistici, ma viene comunque visualizzata nel *Summary* fornito da R Commander.

Ciò che si può ancora notare è che non sono presenti *missing values*, ovvero dati mancanti che sarebbero stati rappresentati con il codice *NA*, come spiegato precedentemente. Nel caso fossero presenti, sarebbe visibile una riga al fondo del *Summary* di ogni variabile contenente il testo *NA’s* seguito dal totale dei valori mancanti.

Per comprendere più a fondo il fenomeno in studio, è utile guardare ad una tavola di contingenza o tabella a doppia entrata. Il vantaggio di rappresentare una parte dei dati sotto questa forma è di mettere in luce eventuali relazioni nascoste fra le variabili i cui valori verranno mostrati sotto forma di frequenza assoluta (o relativa).

Volendo approfondire la percentuale di **sopravvivenza** rispetto al *genere* o rispetto alla *tariffa* pagata o, ancora, rispetto alla *classe del passeggero*, si costruisce una tavola di contingenza facendo però attenzione ad un necessario passaggio preliminare: la tavola di contingenza accetta valori suddivisi in categorie e sarà necessario convertire i dati delle rispettive variabili, ad esempio, da numeriche a fattori.

Per farlo, si farà uso della tab *Data > Manage variables in active data set > Recode variables*, come mostrato in Figura 3.11 a pagina 35, da cui si otterrà una finestra di dialogo nella quale dover scegliere quale variabile sarà *ricodificata*, come in Figura 3.12 a pagina 36.

Per cominciare, si è deciso di considerare la variabile “Classe.Passeggero”. Nel box inerente la ricodifica è possibile decidere come avverrà quest’ultima: i valori delle classi sono stati tradotti in fattori racchiundendoli fra apici. Inoltre, nella casella di testo “New variable name or prefix for multiple recodes:” è stato deciso un nuovo nome di variabile: in questo modo verrà creata una *nuova variabile* all’interno del data set originale lasciando inalterata la variabile “Classe.Passeggero”, denominata “Factor.Classe.Passeggero”.

Anche se il vantaggio principale dell’usare un’interfaccia grafica è quello di minimizzare la necessità di scrivere codice cliccando su menù a tendina, Rcmdr non ne elimina del tutto la necessità. Ricodificare le variabili ne è solo un primissimo esempio.¹⁴

Si otterrà quanto riportato in Figura 3.13 a pagina 36.

Sarà a questo punto possibile richiedere ad Rcmdr la creazione di una tavola di contingenza tramite la tab *Statistics > Contingency tables > Two-way table*, ottenendo quanto riportati in Figura 3.14 a pagina 37.

Il risultato può essere visualizzato in Figura 3.15 a pagina 38, nella maschera dell’output di Rcmdr.

La variabile scelta per essere rappresentata nelle colonne è stata la variabile “Factor.Classe.Passeggero” mentre nelle righe, è stata scelta la variabile “Genere”. Sono tre i risultati ottenuti:

- **Frequency table:** tabella delle frequenze. Sono rappresentate le frequenze assolute per classe dei passeggeri e per genere;

¹⁴Fox, *Using the R commander: A point-and-click interface for R*, p. 32.

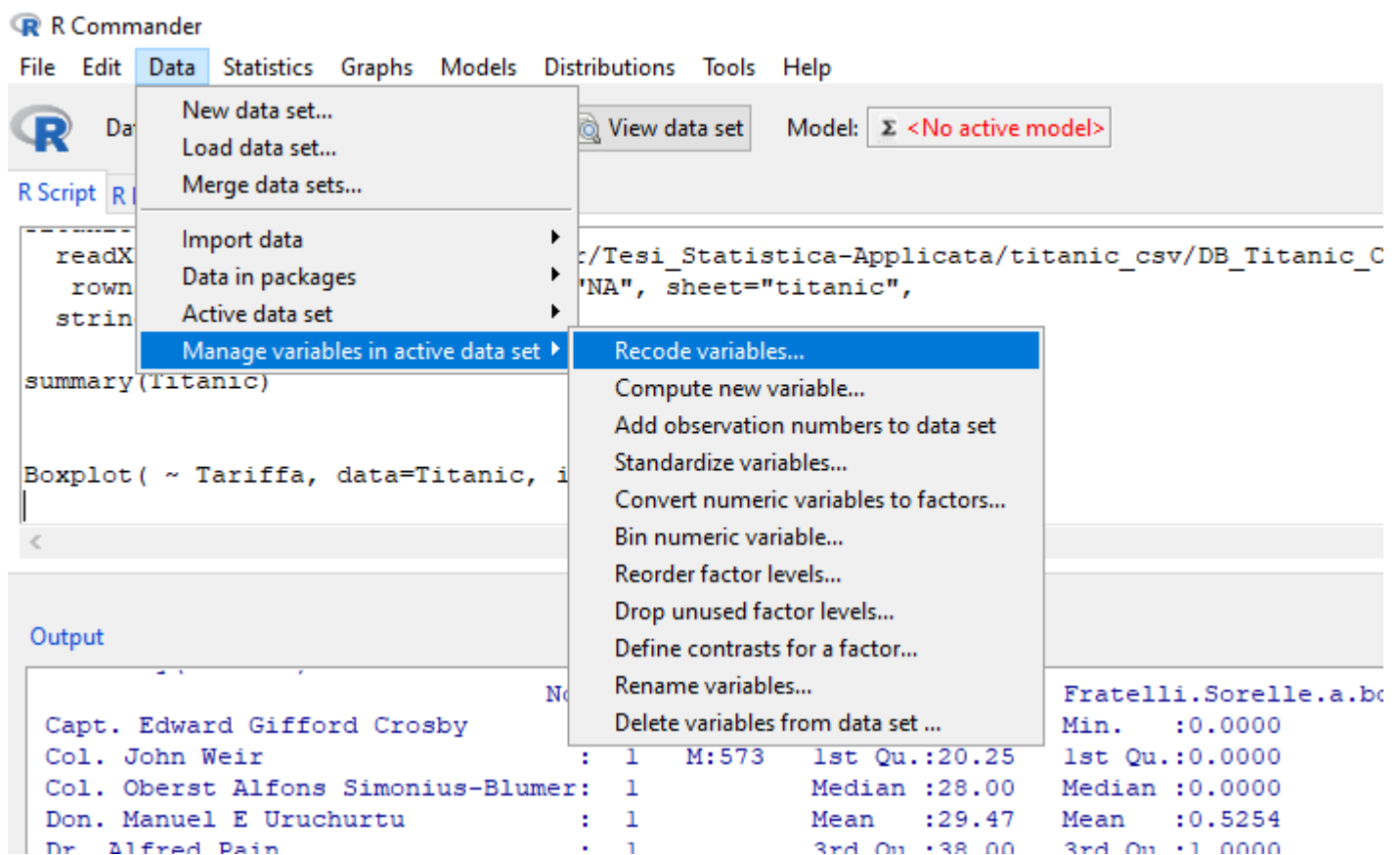


Figura 3.11: Recoding

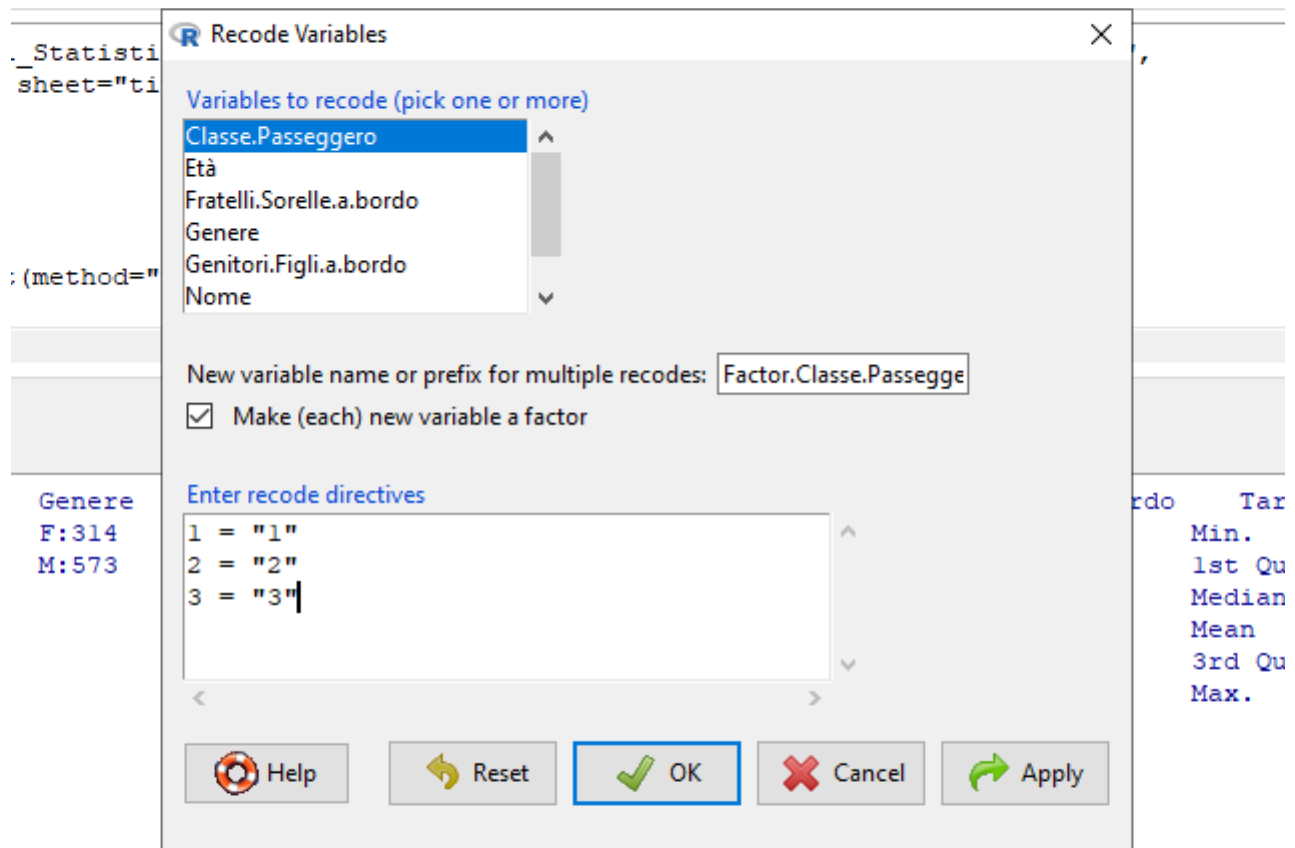


Figura 3.12: Box recoding

```
> Boxplot( ~ Tariffa, data=Titanic, id=list(method="y"))
[1] "258" "677" "734" "28" "88" "340" "436" "310" "739" "118"

> Titanic <- within(Titanic, {
+   Factor.Classe.Passeggero <- Recode(Classe.Passeggero, '1 = "1"; 2 = "2"; 3 = "3";', as.factor=TRUE)
+ })
```

Messages

```
[7] NOTE: The dataset Titanic has 887 rows and 8 columns.
[8] NOTE: The dataset Titanic has 887 rows and 9 columns.
```

Figura 3.13: Risultato recoding

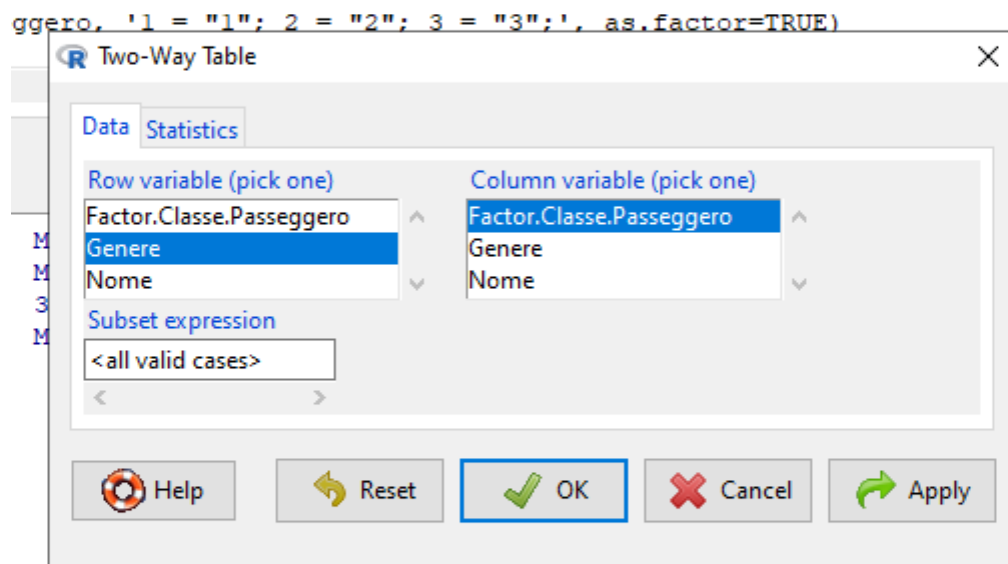


Figura 3.14: Tavola di contingenza Classe Passeggero

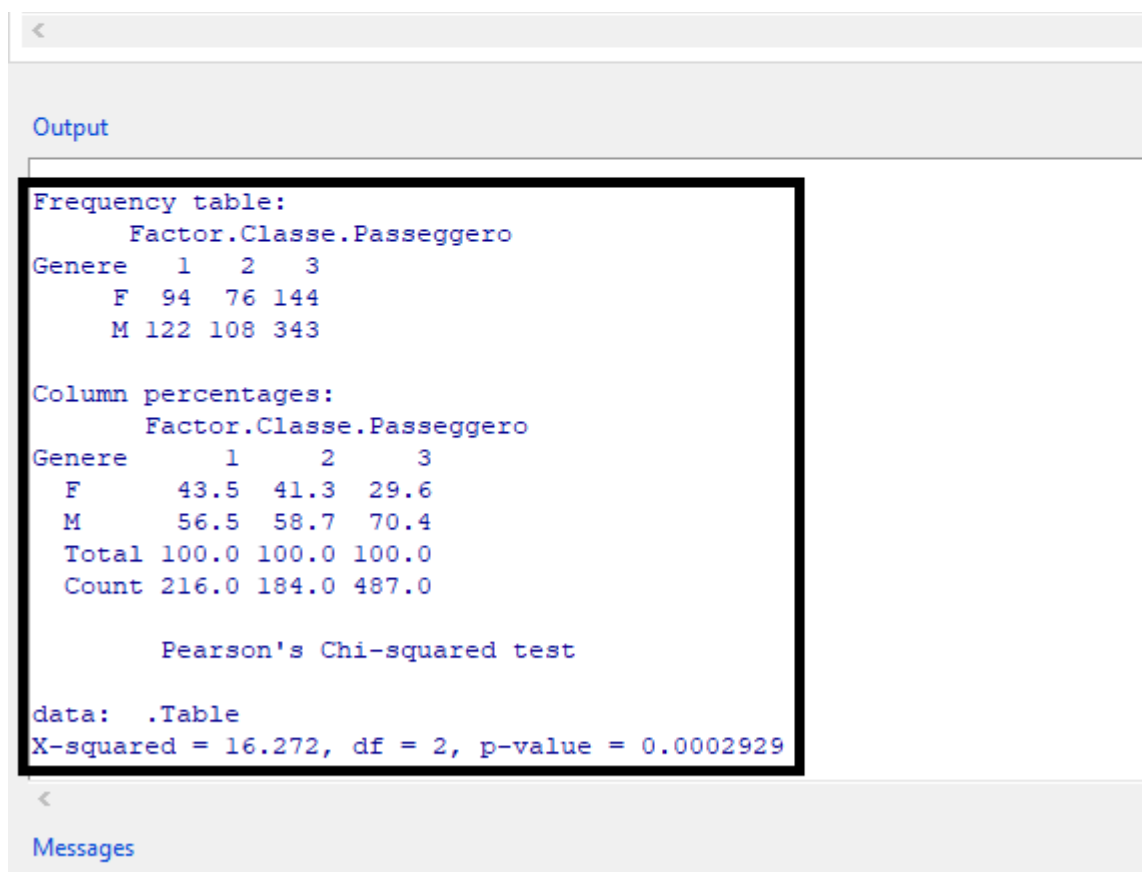


Figura 3.15: Output Tavola di contingenza

- **Column precentages:** in base al campione analizzato, possiamo dire che il 43.5% delle persone appartenenti alla prima classe era di genere femminile mentre il restante 56.5% di genere maschile. Considerando complessivamente il genere femminile, la sua presenza era preponderante nella prima classe, a differenza della terza alla quale vi apparteneva il 70.4% di maschi e solo il 29.6% di passeggeri di genere femminile;
- **Pearson's Chi-squared test:** il test del Chi-quadrato di Pearson è utilizzabile quando i dati sono divisi in gruppi. L'*ipotesi nulla* prevede che le variabili siano *indipendenti*: in questo caso, che non ci sia una relazione fra la classe e il genere dei passeggeri. Per testare tale assunto, il test compara il valore con un modello nel quale i dati sono distribuiti sotto l'ipotesi di indipendenza. Ogni qualvolta i dati osservati non si adattano a tale modello, la probabilità che le variabili siano dipendenti aumenta, permettendo di rifiutare l'ipotesi nulla e di non rifiutare l'*ipotesi alternativa* di dipendenza.

L'output fornisce tre valori tra cui il valore del *Chi-squared test*, il numero di *gradi di libertà* e il p-value. Quest'ultima è la statistica che verrà considerata per trarre le dovute conclusioni. Poiché il suo valore è pressoché equivalente a 0, si può ragionevolmente rigettare l'ipotesi nulla in favore dell'ipotesi alternativa: le variabili sembrano essere dipendenti. Si può concludere che la classe del passeggero dipende dal genere dello stesso.

Infine, volendo precisare la formula per il calcolo del test di Pearson:

$$X^2 = \sum_{i=1}^N \frac{(O_i - E_i)^2}{E_i}.$$

Dove:

- O_i : indica le frequenze osservate;
- E_i : indica le frequenze attese;

Ovvero, si sottraggono le frequenze osservate a quelle attese, le si elevano al quadrato (per non considerare valori negativi) e si divide per le frequenze attese per ottenere un valore normalizzato. Tutto ciò viene calcolato *per ogni i*, ovvero per ogni cella nella tabella di contingenza.

Se nell’immaginario collettivo, quando si parla di disastri marittimi, vale la regola per cui *women and children first*¹⁵, “Prima le donne e i bambini”, si vuole ora comprendere tramite i dati a disposizione se, nel caso in esame, tale sentenza sia corretta.

Si procede quindi con quanto già visto, ovvero ricodifica della variabile “Sopravvissuto.a” e confronto con la variabile “Genere”. Se ne mostreranno, quindi, solamente i risultati, evitando di riportare anche i passaggi intermedi.

Ciò che si ottiene viene mostrato in Figura 3.16 a pagina 41, nella maschera dell’Output di Rcmdr.

Focalizzandosi solamente sulla tavola di contingenza che mostra le frequenze marginali di colonna, si evince che il 68.1% dei passeggeri di genere femminile è sopravvissuto, mentre questo è vero solamente per il 31.9% dei passeggeri di genere maschile.

Inoltre, il p-value pari a 2.2×10^{-16} , equivalente quindi a 0, indica che la sopravvivenza è dipeso, in questo caso, dall’appartenenza al genere femminile o maschile.¹⁶

L’*RMS Titanic* svetta come un raro esempio nel quale poter stabilire un tasso di sopravvivenza femminile più alto in un disastro marittimo, di circa 2.1 volte maggiore rispetto a quello maschile.

Poiché si è anche affermato che il tasso di sopravvivenza fosse maggiore per i bambini, si ricodifica una nuova variabile, dividendo in classi di età i passeggeri: vengono considerati “minorenni” tutti coloro che possedevano un’età al di sotto dei 18 anni, e adulti tutti coloro al di sopra di tale soglia.

La creazione della tabella di contingenza passa attraverso un menù leggermente differente rispetto a quello visto precedentemente.

Infatti, in questo caso, si dovranno prendere in considerazione:

- **Factor.Età**
- **Genere**

¹⁵Mikael Elinder e Oscar Erixson. “Gender, social norms, and survival in maritime disasters”. In: *Proceedings of the National Academy of Sciences* 109.33 (2012).

¹⁶La domanda che ci si può porre è se il “Prima le donne e i bambini” sia o meno una consuetudine dal punto di vista dei disastri avvenuti in mare. Questo argomento è stato studiato e sviluppato in diverse ricerche, in particolare in “Gender, social norms, and survival in maritime disasters” (Elinder e Erixson, “Gender, social norms, and survival in maritime disasters”), nella quale sono stati analizzati dati provenienti da 18 disastri in mare nel periodo compreso tra il 1852 e il 2011. Considerando il destino di oltre 15.000 passeggeri e membri dell’equipaggio di oltre 30 nazionalità, è stato dimostrato che tale adagio non si basa su una consuetudine sociale. Il report evidenzia che il tasso di sopravvivenza dei passeggeri di genere femminile è, in media, circa la metà di quello dei passeggeri di genere maschile.

```
<
Output
Frequency table:
      Factor.Sopravvissuto.a
Genere  No  Si
      F   81 233
      M  464 109

Column percentages:
      Factor.Sopravvissuto.a
Genere      No      Si
      F      14.9  68.1
      M      85.1  31.9
Total 100.0 100.0
Count 545.0 342.0

      Pearson's Chi-squared test

data:  .Table
X-squared = 260.72, df = 1, p-value < 2.2e-16

<
Messages
[3] NOTE: The dataset Titanic has 887 rows and 9 columns.
[4] NOTE: The dataset Titanic has 887 rows and 10 columns.
<
```

Figura 3.16: Output Sopravvivenza-Genere

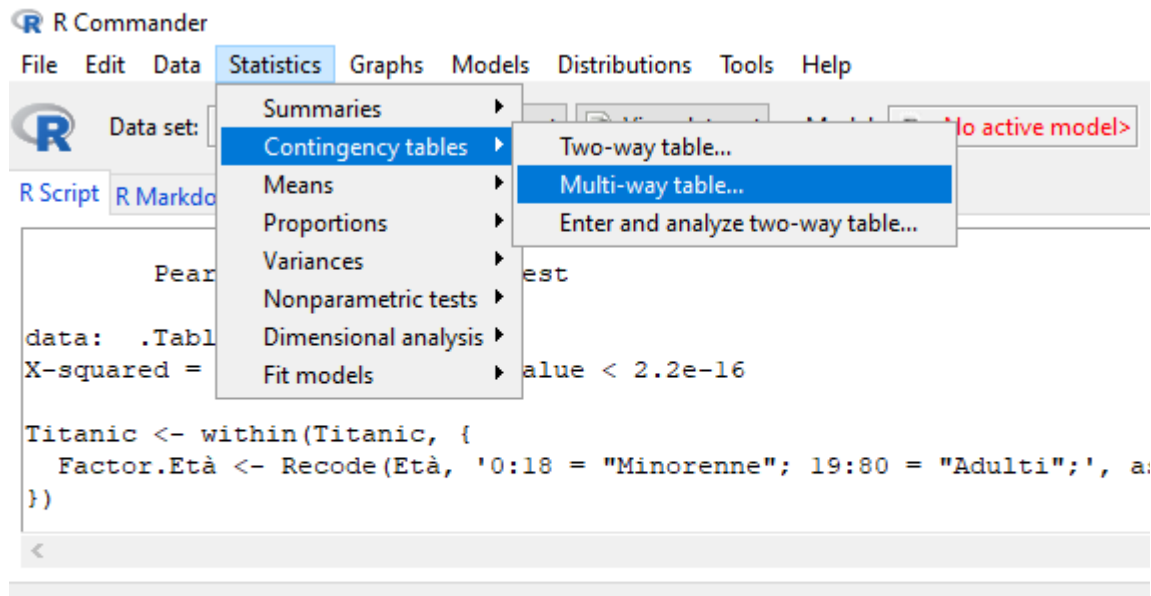


Figura 3.17: Multi-way table

- **Factor.Sopravvissuto.a**

Così facendo però, non si può utilizzare una *two-way table*. Al contrario, si sceglierà una *multi-way table*, dal seguente menù a tendina, come in Figura 3.17 a pagina 42.

I parametri verranno scelti come mostrato in Figura 3.18 a pagina 43.

Come variabile di riga è stato scelto il “Genere”; come variabile di colonna la “Factor.Età” e come *variabile di controllo* la “Factor.Sopravvissuto.a” per evidenziare chi, fra adulti e minorenni e fra genere maschile e femminile sia maggiormente sopravvissuto al disastro.

I risultati sono mostrati in Figura 3.19 a pagina 44 e in Figura 3.20 a pagina 45.

Concentrandosi sulla seconda figura, si guarda alle percentuali di colonna: è immediato notare che fra i sopravvissuti i valori maggiori si ritrovano nel genere femminile sia per gli adulti che per i minorenni e, nel genere maschile, per quanto riguarda i minorenni.

Almeno per il caso del *Titanic*, sembra si possa affermare che la norma non scritta del “Prima le donne e i bambini” sia valsa¹⁷.

¹⁷G. Masarotto e S.M. Iacus. *Laboratorio di statistica con R*. 2003.

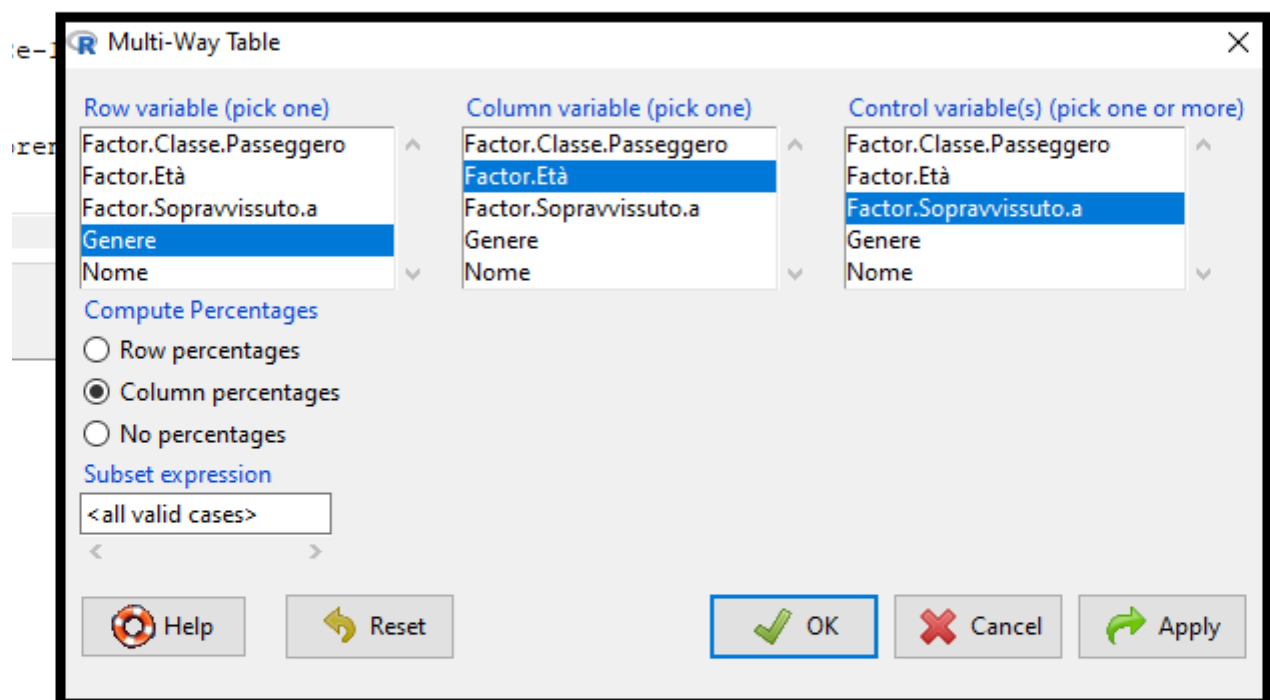


Figura 3.18: Multi-way table parametri

```
<
```

Output

```
+ cat("\nColumn percentages:\n")
+ print(colPercents(.Table))
+ })
```

Frequency table:

```
, , Factor.Sopravvissuto.a = No
```

| | Factor.Età | |
|--------|------------|-----------|
| Genere | Adulti | Minorenne |
| F | 52 | 29 |
| M | 405 | 59 |

```
, , Factor.Sopravvissuto.a = Si
```

| | Factor.Età | |
|--------|------------|-----------|
| Genere | Adulti | Minorenne |
| F | 182 | 51 |
| M | 82 | 27 |

Messages

```
[5] NOTE: The dataset Titanic has 887 rows and 10 columns.
[6] NOTE: The dataset Titanic has 887 rows and 11 columns.
```

```
<
```

Figura 3.19: Multi-way table Output (1)

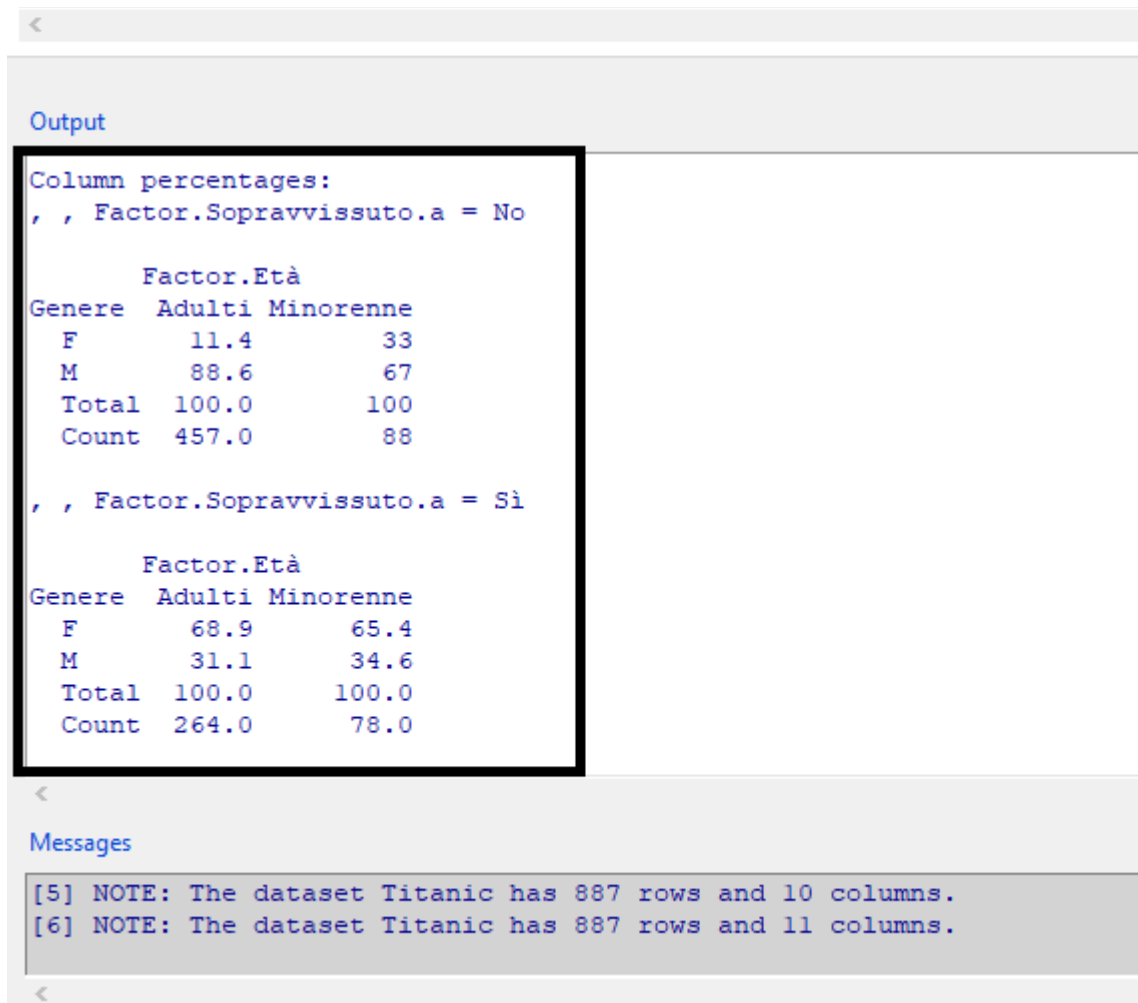


Figura 3.20: Multi-way table Output (2)



Figura 3.21: Bar Chart - Classe-Sopravvivenza

La successiva domanda che ci si pone, è se la classe nella quale il passeggero si è imbarcato è stata determinante per la sua sopravvivenza. Per comprendere se esiste una relazione di questo tipo fra le due variabili, si inizia visualizzando un *Bar Chart* nel quale si inserisce sull'asse delle ascisse la variabile “Classe Passeggero” e sull'asse delle ordinate la “Sopravvivenza %”. [Figura 3.21 a pagina 46]

Per ottenere tale grafico si è cliccato sulla tab *Graphs > Bar graph* come in Figura 3.22 a pagina 47.

Successivamente, si è aperto il box di dialogo: la variabile scelta è “Factor.Classe.Passeggero” e si è anche deciso di cliccare su “Plot by:” e di usare come condizionamento la variabile “Factor.Sopravvissuto.a” come in Figura 3.23 a pagina 48.

Soffermandosi sull'output del grafico a barre, viene visualizzata una forte percentuale di sopravvivenza per i passeggeri (maschi e femmine indistintamente) che avevano acquistato un biglietto in prima classe. La percentuale di sopravvivenza diminuisce nelle successive classi.

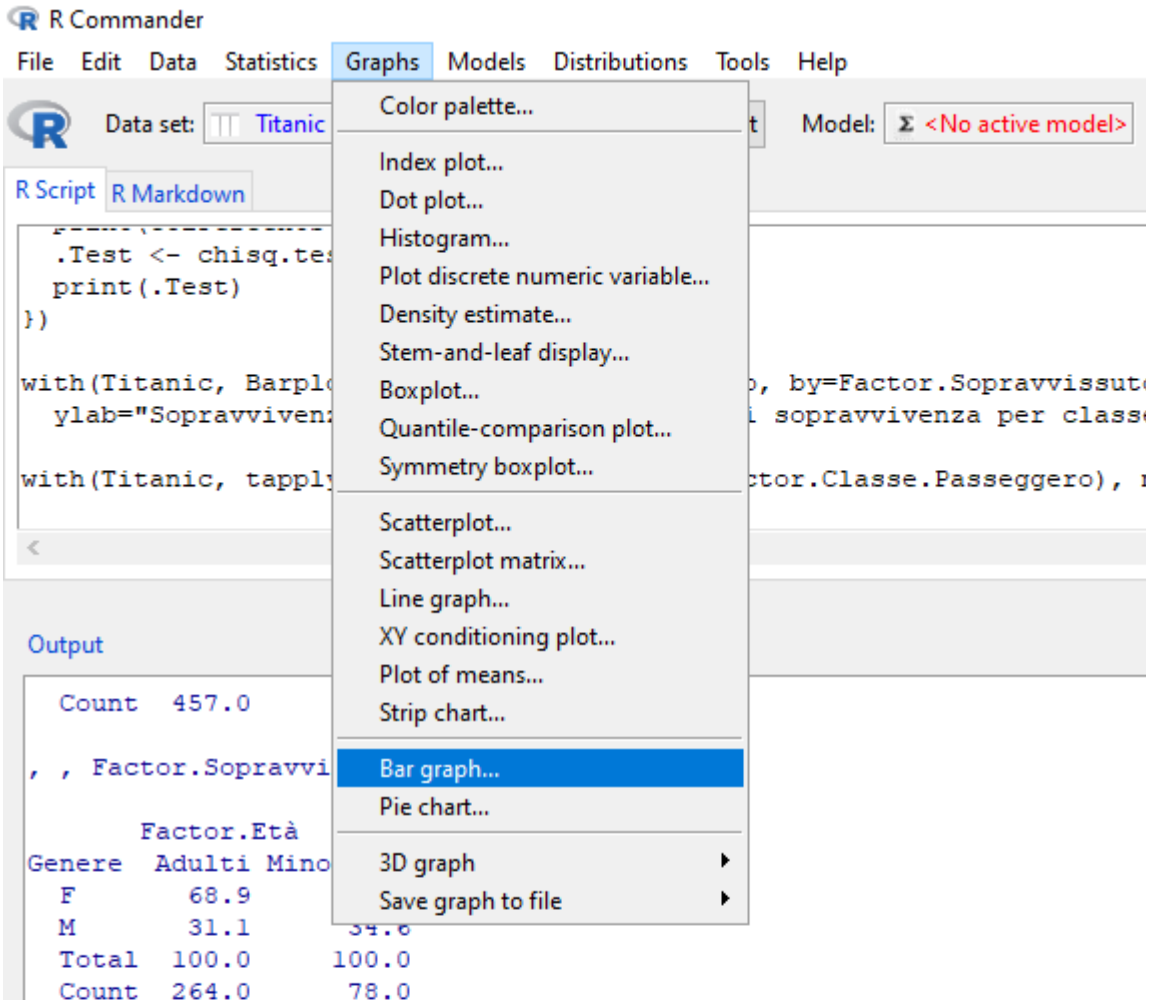


Figura 3.22: Graphs - Bar Chart

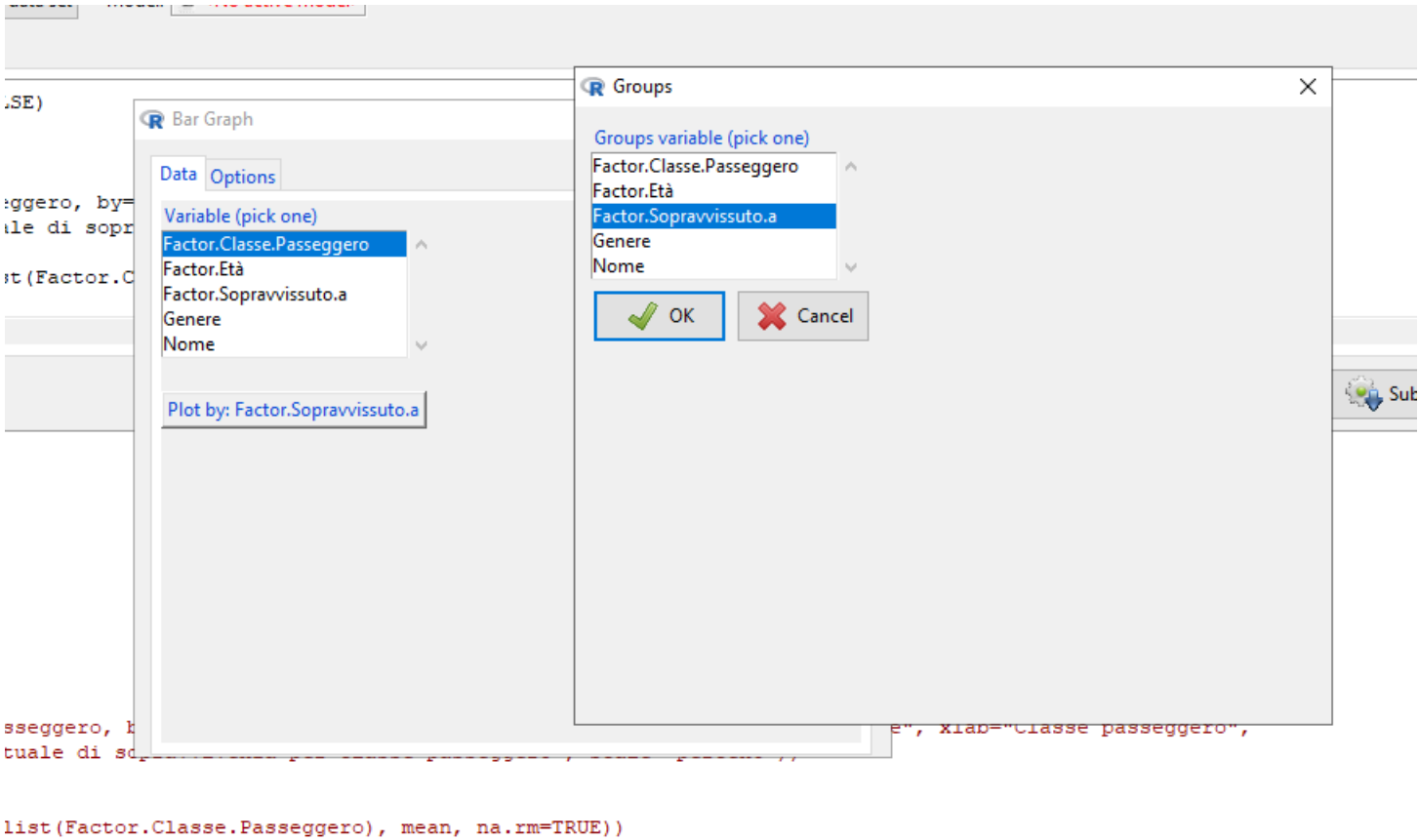


Figura 3.23: Graphs - Variabili

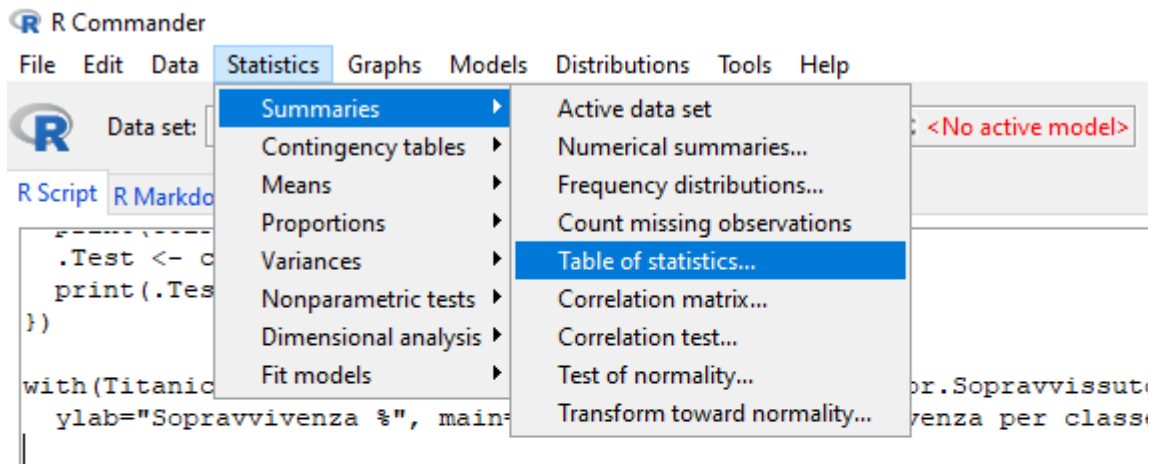


Figura 3.24: Table of statistics

Oltre all’aspetto grafico, tali percentuali di sopravvivenza possono anche essere viste numericamente tramite la tab *Statistics > Summaries > Table of statistics* come in Figura 3.24 a pagina 49.

Il box di dialogo si presenta come in Figura 3.25 a pagina 50.

Le variabili scelte sono state “Factor.Classe.Passeggero” e “Sopravvissuto.a” e la statistica voluta è la media.

Il risultato, visibile in Figura 3.26 a pagina 51

mostra chiaramente che la percentuale di sopravvivenza in prima classe equivaleva a circa il 63%, in seconda classe a circa il 47 % e in terza classe a circa il 24%.

Per avere una certezza maggiore della dipendenza della variabile “Sopravvivenza” rispetto alla “Classe passeggero”, il passo successivo naturale da svolgere è quello del calcolo di un valore numerico che possa indicare tale relazione. Il test sull’indipendenza di Pearson (test del chi-quadro) che viene fornito quando si richiede ad Rcmdr di computare una tabella a doppia entrata, potrebbe rispondere alla domanda. Riportandone solamente i risultati, poiché i procedimenti sono gli stessi già precedentemente presentati, l’output è mostrato in Figura 3.27 a pagina 52.

Il valore del p-value è pari a 2.2×10^{-16} : essendo prossimo a 0, è possibile rigettare l’ipotesi nulla (di indipendenza delle variabili) a favore dell’ipotesi alternativa (di dipendenza). Questo comporta poter concludere l’esistenza di una dipendenza fra la classe del passeggero e la sua sopravvivenza.

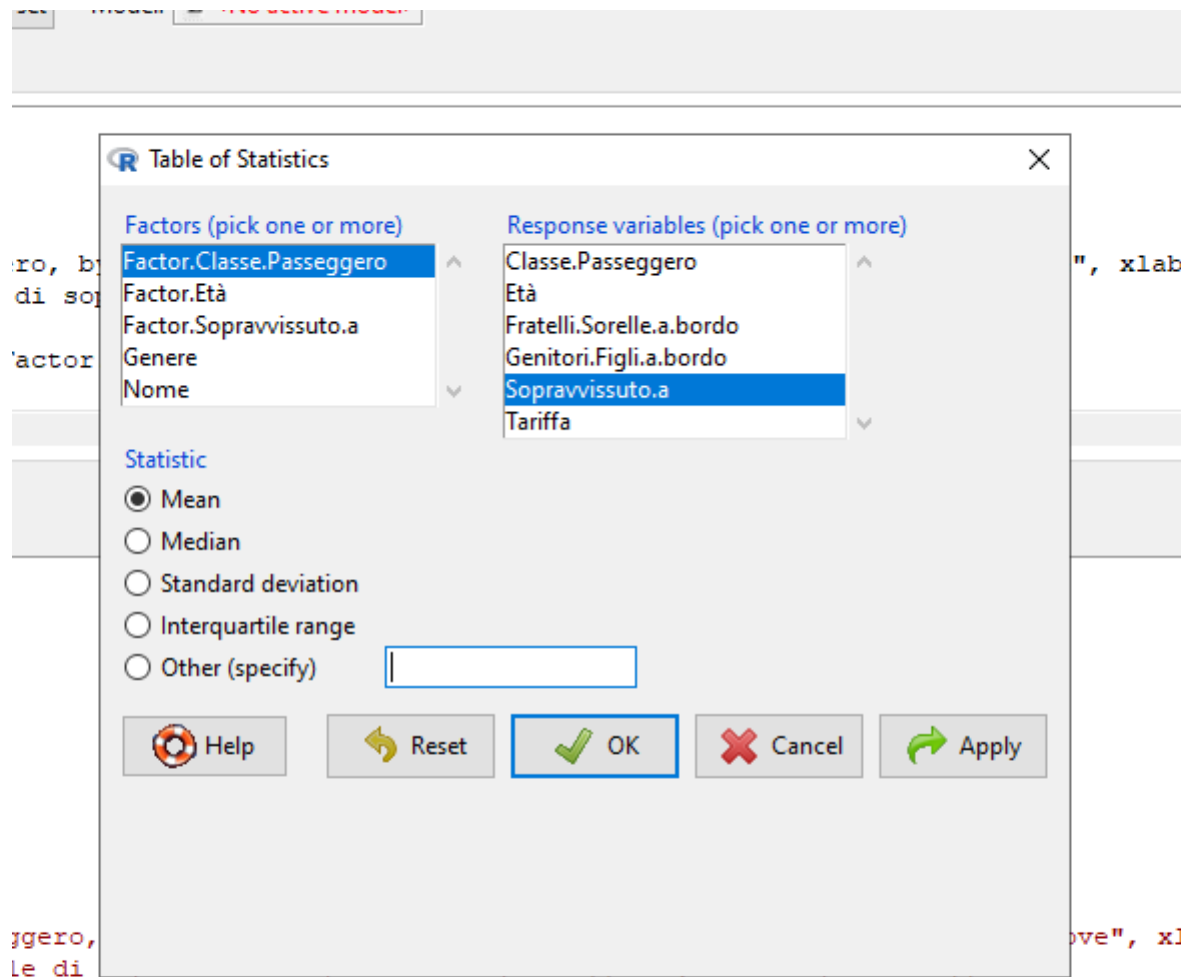


Figura 3.25: Table of statistics - Box di dialogo

```
+ ylab="Sopravvivenza %", main="Percentuale di sopravvivenza per classe passeggero", scale="percent")

> with(Titanic, tapply(Sopravvissuto.a, list(Factor.Classe.Passeggero), mean, na.rm=TRUE))
      1      2      3
0.6296296 0.4728261 0.2443532
```

<

Messages

RGui with the single-document interface (SDI); see ?Commander.
[3] NOTE: The dataset Titanic has 887 rows and 11 columns.

<

Figura 3.26: Table of statistics - Output

```

Output
Frequency table:
              Factor.Sopravvissuto.a
Factor.Classe.Passeggero  No  Sì
1      80 136
2      97  87
3     368 119

Column percentages:
              Factor.Sopravvissuto.a
Factor.Classe.Passeggero  No  Sì
1      14.7 39.8
2      17.8 25.4
3      67.5 34.8
Total 100.0 100.0
Count 545.0 342.0

Pearson's Chi-squared test

data: .Table
X-squared = 101.22, df = 2, p-value < 2.2e-16

Messages
RGui with the single-document interface (SDI); see ?Commander.
[3] NOTE: The dataset Titanic has 887 rows and 11 columns.

```

Figura 3.27: Table of statistics - Percentuali di colonna

Per comprendere, infine, a quanto corrisponda il valore di tale dipendenza, si può calcolare la statistica definita come *V di Cramer*, compresa fra 0 ed 1, calcolata con i dati precedentemente ottenuti tramite il test:

$$V = \sqrt{\frac{X^2}{X_{max}^2}} = \sqrt{\frac{101.22}{887}}.$$

Dove si ricorda che il denominatore è stato ottenuto tramite la seguente formula:

$$X_{max}^2 = N * [\min(k, m) - 1] = 887 * [2 - 1].$$

Il risultato pari a 0.337 809 109 permette di affermare che la dipendenza sia presente fra le variabili anche se non appare così elevata.

Per concludere l'analisi esplorativa del data set, si può dare un'ultima rappresentazione della variabile "Sopravvivenza" relativamente all'età dei passeggeri. Per farlo, si procede con la divisione della variabile "Età" nella rispettiva variabile divisa in classi "Classi.età". Si procede come visto precedentemente per la ricodifica di una variabile da numerica a categorica per poi rappresentarne un bar plot, come in Figura 3.28 a pagina 54.

Dalla lettura del grafico non sembra essere presente una prevalenza di sopravvivenza nelle tre classi centrali di età, mentre una maggiore probabilità di sopravvivenza è visibile nella classe di età dei più giovani (fra gli 0 anni e i 15 anni. Al contrario, per la classe di età fra i 60 anni e gli 80, la percentuale di non sopravvivenza si aggira intorno all'80%).

3.3 Modello di regressione logistico: previsione della sopravvivenza dei passeggeri

Se le analisi fin qui svolte sono servite a comprendere la composizione del data set, in questo paragrafo si vuole applicare una metodologia in grado di prevedere quali passeggeri hanno avuto la maggiore probabilità di sopravvivenza tenuto conto delle variabili già presentate. Mentre in altri modelli di regressione, come ad esempio la regressione lineare, si assume che la variabile risposta Y sia di tipo quantitativo e continua (si vuole modellare, ad esempio, il voto scolastico di alcuni studenti in base al numero di ore di studio), è possibile pensare che tale variabile possa presentarsi sotto forma di dati qualitativi, anche definiti come "Dati categorici"¹⁸. Un approccio alla previsione

¹⁸Gareth James, Daniela Witten et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013, p. 127.

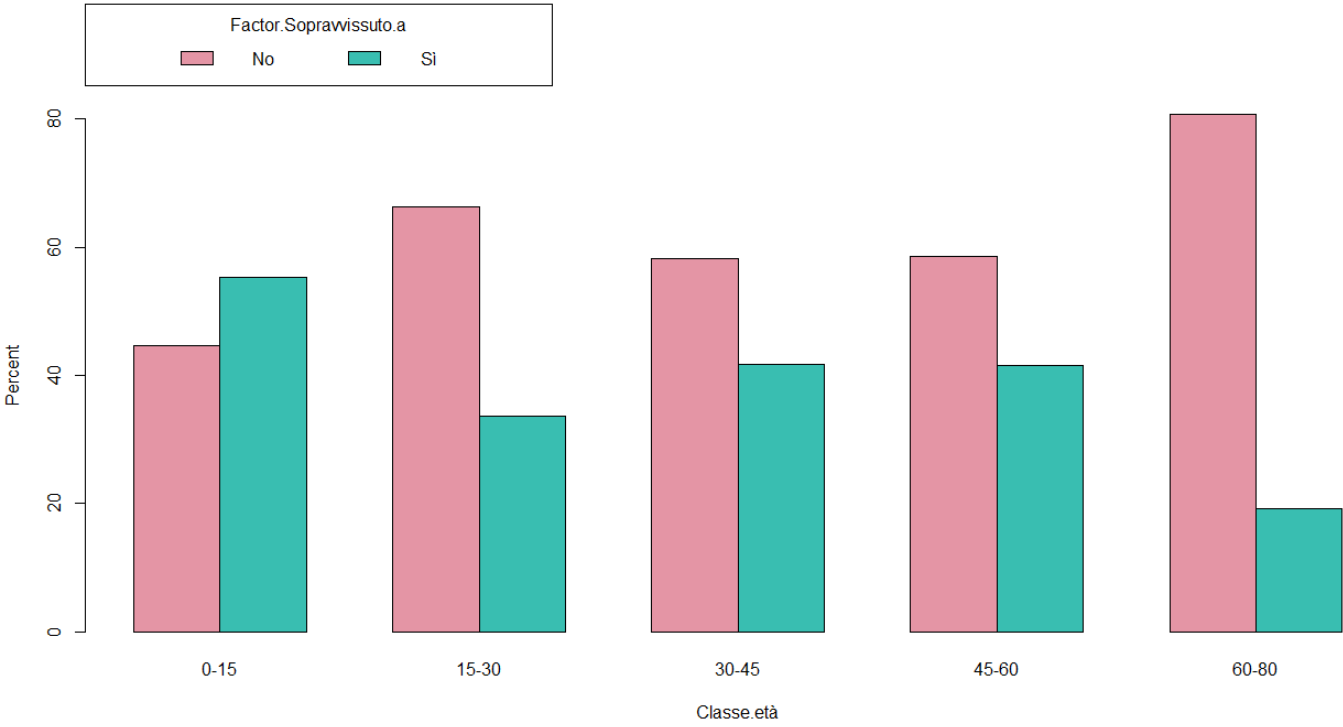


Figura 3.28: Bar chart - Età-Sopravvivenza

di dati qualitativi è definito come *classification*: le osservazioni presenti nel data set vengono classificate in *categorie*. Nella sua versione più semplice, la variabile risposta Y è binaria: “Superato” e “Non superato” per quanto riguarda il risultato di un test, “Default” e “Not default” per il possessore di carta di credito, “Sopravvissuto/a” e “Non sopravvissuto/a” nel caso dei passeggeri della *RMS Titanic*.

Per poter essere utilizzata all’interno della regressione logistica, la variabile “Sopravvissuto.a” è già stata codificata sotto forma di *dummy variable*, ovvero di variabile dicotomica o binaria che indica se un passeggero è o meno sopravvissuto (rispettivamente 1 o 0).

Il valore quindi interessante è p , ovvero la proporzione di passeggeri che è sopravvissuta e che riporta valore 1.

Ma perché non è possibile utilizzare un modello di regressione lineare? La risposta è contenuta nelle assunzioni del modello stesso.

Se la media della distribuzione della variabile Y è definita da:

$$E[Y] = 0 \times q + 1 \times p = p;$$

dove q rappresenta la proporzione dei passeggeri non sopravvissuti pari a $1 - p$.

Mentre la varianza della Y è pari a:

$$Var(Y) = 0^2 \times (1 - p) + 1^2 \times p - p^2 = p(1 - p).$$

Fissato un dato p , la varianza viene a modificarsi in base a tale valore. Si evince quindi che la varianza così definita *non è costante*. Uno degli assunti del modello di regressione lineare prevede la costanza della varianza degli errori, ovvero¹⁹:

$$Var(\varepsilon_j) = \sigma^2.$$

Inoltre, sia Y la variabile risposta con valori 0 e 1. Se volessimo modellare la probabilità del valore 1 con un solo predittore nel modello di regressione lineare, dovremmo scrivere:

$$p = E(Y|x) = \beta_0 + \beta_1 x + \varepsilon.$$

I problemi in cui si incorre utilizzando tale modello sono duplici:

- Da una parte, i valori predetti potrebbero anche essere maggiori di 1 o minori di 0 in quanto non ci sono limiti all’espressione lineare appena vista;

¹⁹Richard Arnold Johnson, Dean W Wichern et al. *Applied multivariate statistical analysis*. Vol. 5. 8. Prentice hall Upper Saddle River, NJ, 2002.

- Dall'altro lato, siccome la varianza di Y dipende dal valore di p , non si può garantire la costanza della varianza degli errori.

Di conseguenza, il modello lineare non è adatto alla previsione di tali quantità dicotomiche.

Si inizia quindi a prendere in considerazione il concetto di *odds*: gli *odds* sono definiti come il rapporto fra il numero di volte in cui l'evento si verifica (p) e il numero di volte in cui l'evento non si verifica ($1 - p$). Ovvero:

$$odds = \frac{p}{1 - p}.$$

Il concetto di ODDS viene usato in ambito medico, biologico ma anche, ad esempio, nel mondo delle scommesse sportive. Infatti, indica chiaramente agli scommettitori la quantità vinta o persa di una loro scommessa: se si prospetta la sconfitta di un pugile, ad esempio, per 3 a 1, significa che una sua vittoria pagherà al giocatore 3 volte la cifra scommessa. Concetto ben diverso da quello di *probabilità*: quest'ultima, infatti, viene indicata come il rapporto tra il numero di volte in cui l'evento si verifica e la somma tra il numero di volte in cui l'evento si verifica e in cui l'evento non si verifica.

Mentre la probabilità è un valore compreso tra 0 ed 1, così può non essere per gli *odds*.

Considerando un altro esempio, se ad 8 persone su 10 non vengono controllati i bagagli²⁰ al momento dell'arrivo in aeroporto, ad esempio, si può dire che $p = 0.8$, ma gli *odds* di non essere controllati sono pari a $\frac{0.8}{0.2} = 4$, ovvero 4 ad 1 di non avere i bagagli controllati. Ma gli *odds* di essere controllati sono pari a $\frac{0.2}{0.8} = \frac{1}{4}$. Ci si trova quindi di fronte una situazione asimmetrica.

D'altra parte, considerando il logaritmo naturale degli *odds* ($\ln(4) = 1.386$ e il $\ln(\frac{1}{4}) = -1.386$) si recupera una situazione di simmetria.

Nella regressione logistica per modellare la probabilità che una variabile dicotomica assuma valore 1, si utilizza la funzione logistica definita come:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

Se quindi gli *odds* sono pari a quanto visto precedentemente, si può dire che, sostituendo:

²⁰Johnson, Wichern et al., *Applied multivariate statistical analysis*.

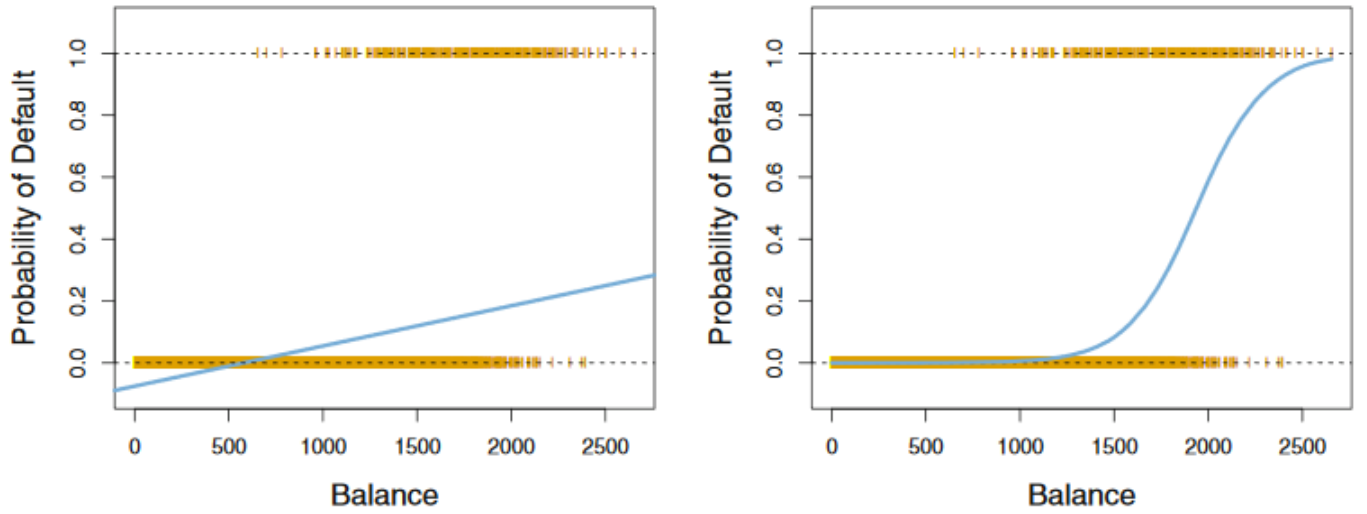


Figura 3.29: Modello lineare e Modello logistico

$$\frac{p(X)}{1 - p(X)} = \dots = \frac{e^{\beta_0 + \beta_1 \cdot X}}{1 + e^{\beta_0 + \beta_1 \cdot X}} \cdot (1 + e^{\beta_0 + \beta_1 \cdot X}) = e^{\beta_0 + \beta_1 \cdot X}.$$

E applicando il logaritmo naturale ad entrambi i lati dell'equazione si otterrà:

$$\ln(odds) = \ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 \cdot X.$$

Un esempio di come si presenta graficamente²¹ il modello logistico è rappresentato in Figura 3.29 a pagina 57.

A sinistra della Figura 3.29 vediamo il modello lineare applicato a valori binari: è chiara la relazione esistente fra la variabile X e la variabile Y . A valori inferiori a 500 per la variabile “Balance”, si ha una probabilità inferiore a 0 di commettere *default*; mentre per valori elevati si potrebbe incorrere in una probabilità superiore ad 1. Concetto che non ha senso dal punto di vista probabilistico.

Al contrario, nel modello logistico a destra, qualsiasi valore della probabilità di default è compresa fra 0 ed 1. A valori positivi di β_1 , un incremento di X equivale ad un incremento della $p(X)$, e viceversa.

²¹James, Witten et al., *An introduction to statistical learning*, p. 131.

3.4 Stima dei coefficienti del modello

Il modello appena presentato contiene un solo regressore X con un solo coefficiente β_1 . Per la stima dei coefficienti, se nel caso del modello lineare si utilizza il metodo *OLS* (*Ordinary Least Squares*), nel modello di regressione logistico si utilizza il metodo del *maximum likelihood*: si cerca di trovare $\hat{\beta}_0$ e $\hat{\beta}_1$ tali che la funzione logistica restituisca un valore di probabilità vicino ad 1 per tutte le osservazioni per cui il fenomeno è avvenuto e un numero vicino a 0 per quelle osservazioni nelle quali il fenomeno non è presente.

Formalmente, si cercano β_0 e β_1 che massimizzano la funzione di verosimiglianza seguente:

$$L(\beta_0, \beta_1) = \prod_{i: y_i=1} p(x_i) \prod_{i': y_{i'}=0} (1 - p(x_{i'})).$$

Il metodo del *maximum likelihood* per la stima dei parametri del modello, ovvero del vettore $\hat{\beta}$, considera tutte le N osservazioni del data set che presentano etichetta di 0 ed 1. Per ogni osservazione etichettata con il valore 1, il metodo cercherà di stimare il vettore $\hat{\beta}$ tale che la $\widehat{p(X)}$ sia il più possibile vicino ad 1; per ogni osservazione etichettata con il valore 0, il metodo cercherà di stimare il vettore $\hat{\beta}$ tale che $\widehat{p(X)}$ sia il più possibile vicino a 0. In quest'ultimo caso, possiamo anche considerare il valore complementare, ovvero $1 - \widehat{p(X)}$ che appare nella formula e che il metodo cercherà di stimare il più possibile prossimo a 1.

I due coefficienti stimati sono scelti, quindi, per massimizzare tale funzione, definita *likelihood function*.

Poiché il caso in esame prevede l'utilizzo di più variabili scelte come regressori, viene considerata la sua espressione multivariata.

In generale, quindi, si può formalizzare come segue:

$$\ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 \cdot X_1 + \dots + \beta_p \cdot X_p;$$

riscrivendo anche:

$$p(X) = \frac{e^{\beta_0 + \beta_1 + \dots + \beta_p \cdot X_p}}{1 + e^{\beta_0 + \beta_1 + \dots + \beta_p \cdot X_p}}.$$

Allo stesso modo che nel caso univariato, anche nel caso multivariato i coefficienti vengono stimati col metodo *maximum likelihood*.

3.5 Applicazione del modello ai dati

Per applicare il modello di regressione logistica al data set è importante prima di tutto soffermarsi sulla divisione dei dati in due sottoinsiemi. Nell'applicazione di un qualsiasi modello predittivo a determinati dati, ciò che si vuole ottenere è *l'allenamento* di quest'ultimo per predire un certo output. Successivamente, si vuole testare tale modello su un nuovo insieme di dati con finalità predittive, generalizzando il suo comportamento a nuovi dati sui quali non è stato allenato.

Tale operazione passa attraverso la divisione del data set originario in due sotto data set chiamati, rispettivamente, *training set* e *testing set*. Con il training set si *allena* il modello sui dati messi a disposizione. Con il test set si *testa* la bontà delle capacità predittive del modello allenato.

La domanda a cui si deve rispondere ora è, quindi, come dividere correttamente i dati a disposizione. Non essendo presente in letteratura una metodologia prevalente di scelta della quantità di osservazioni che deve essere presente in un sub set piuttosto che nell'altro, si è deciso di optare per una soluzione 80/20: l'80% delle osservazioni verrà incluso nel data set di allenamento e il restante 20% in quello del test. Tale scelta viene giustificata dal fatto che maggiore è il numero di osservazioni che ricadono all'interno del data set di allenamento, e maggiori sono le informazioni che il modello ha a disposizione per poter poi tentare di fare previsione su nuovi input.

Prima di procedere alla divisione dei dati, un'operazione preliminare prevede la fissazione di un "seme casuale" necessario alla riproduzione dell'esperimento in corso per ottenere gli stessi risultati in un momento successivo: il comando *set.seed()* ha la funzionalità di generare numeri casuali. Tale espediente è necessario nel momento in cui si afferma, in conclusione, che il modello utilizzato presenta una certa percentuale di accuratezza. Impostando un determinato seme, sarà possibile riprodurre l'intero codice conseguendo gli stessi identici risultati e avallare le proprie asserzioni.

Senza la volontà di essere esaustivi, si può affermare che il calcolatore non è in grado di generare dei *veri* numeri casuali, in quanto il suo compito principale è quello di eseguire operazioni deterministiche: quando gli si richiede di aprire un software una prima ed una seconda volta, il risultato deve essere lo stesso, ovvero l'apertura del programma. Per ovviare al problema dell'impossibilità di generare dei veri numeri casuali, si sono sviluppate procedure che simulano la loro generazione. Tali procedure sono definite come *pseudo-random-number generators*²²: ne esistono di diverse ma, di default,

²²John Fox. *R-Commander*. URL: <https://www.rcommander.com/>, p. 178.

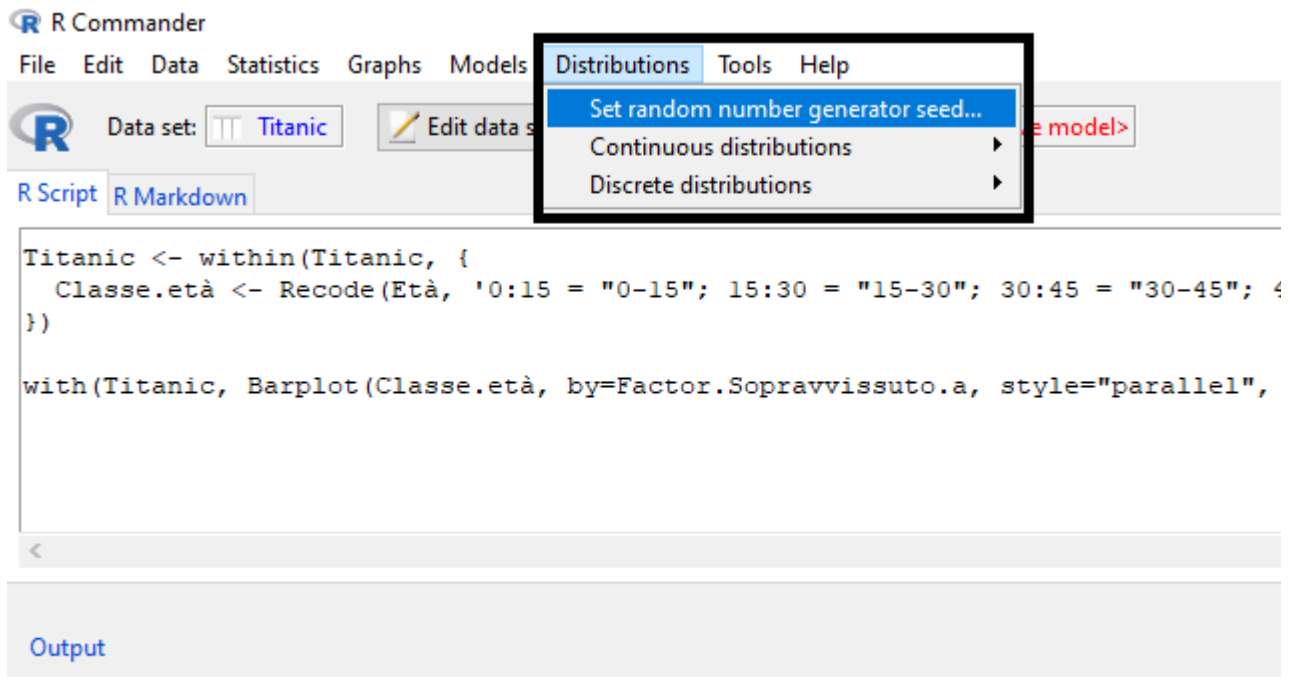


Figura 3.30: Scheda Set-seed

R ed R Commander utilizzano il metodo “Mersenne-Twister”. Si tratta di un algoritmo sviluppato da Makoto Matsumoto e Takuji Nishimura negli anni 1996/1997 per sopperire alle mancanze di altri algoritmi al tempo presenti. Il vantaggio dell’utilizzo di tali numeri *pseudo-casuali* risiede nella possibilità di poter creare una simulazione riproducibile più e più volte ottenendo gli stessi output.

Per produrre tale risultato in R Commander si fa ricorso alla scheda presente nella barra degli strumenti chiamata *Distributions > Set Random Number Generator Seed*, come mostrato in Figura 3.30 a pagina 60.

Verrà aperto un box di dialogo rappresentato in Figura 3.31 a pagina 61.

Si ha la possibilità di scegliere un qualsiasi valore fra 1 e 100 000: l’importante, se si vuole riprodurre in un momento successivo quanto viene fatto ottenendo gli stessi identici risultati, è di conoscere il valore del seme durante il primo esperimento. Per semplicità, in questo caso, il valore preso a riferimento è il valore 1. Cliccando sul pulsante “OK” si ottiene il seguente risultato in Figura 3.32 a pagina 62 sia nello script di Rcmdr, sia nella maschera dell’Output.

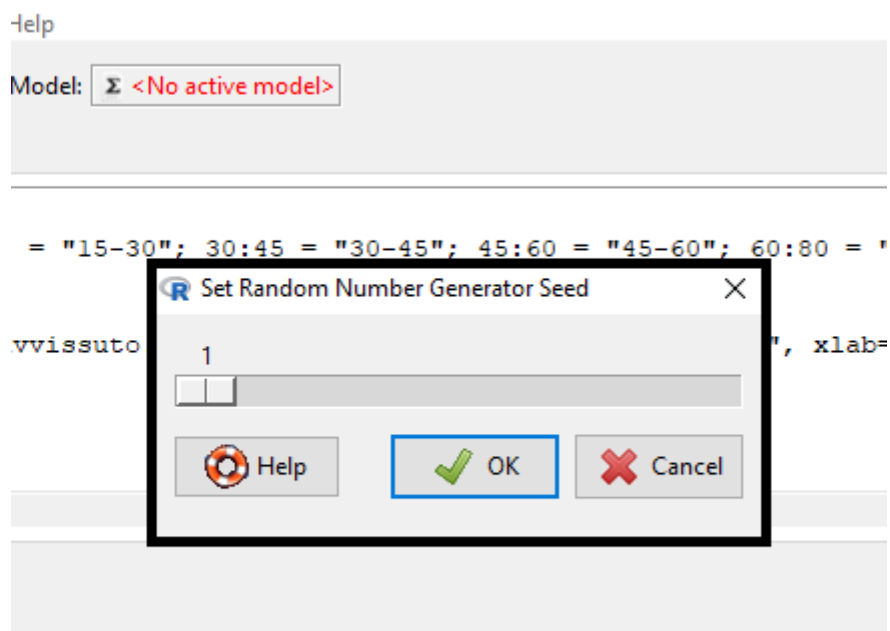


Figura 3.31: Set-seed box di dialogo

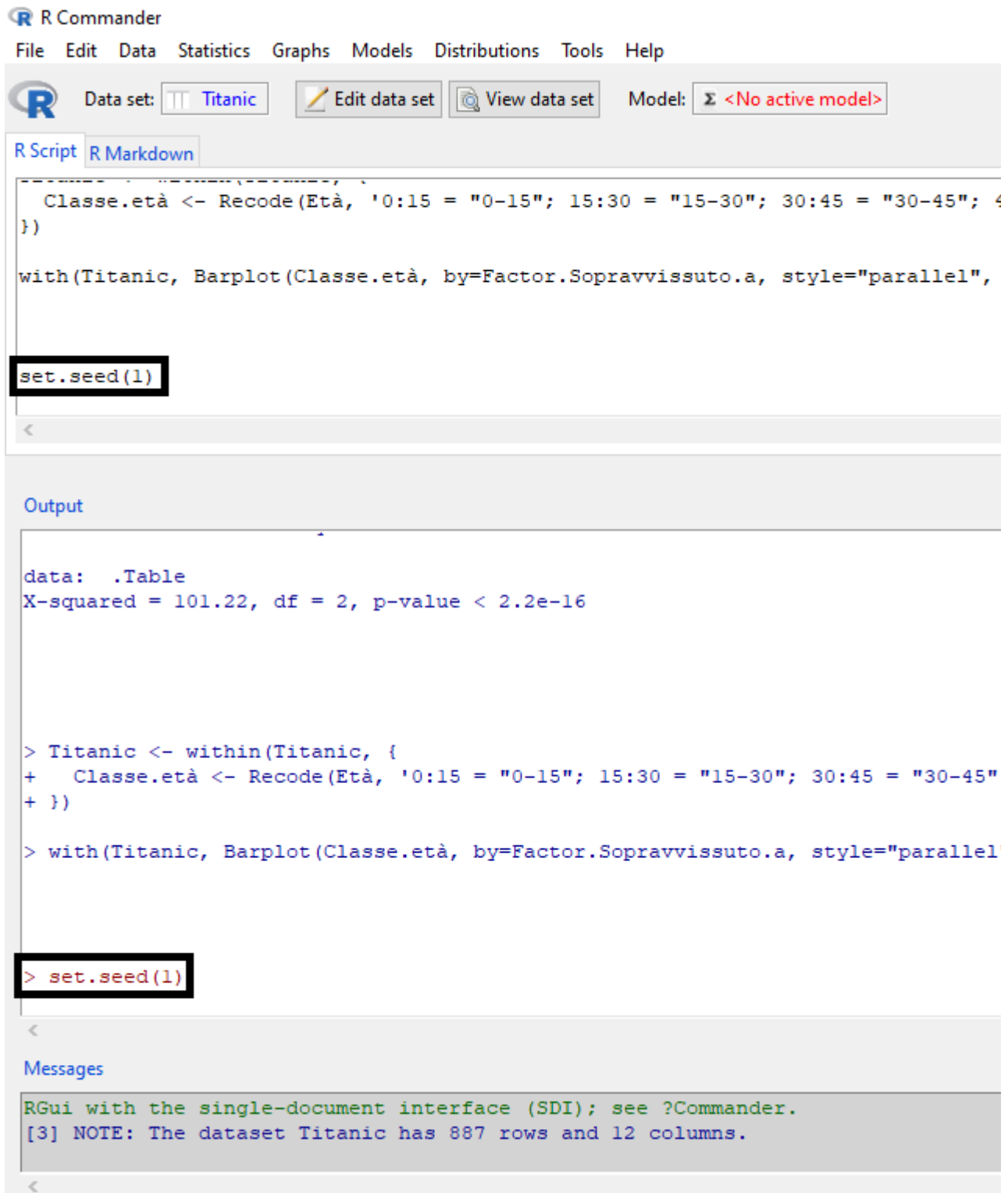


Figura 3.32: Set-seed Output

L'unico passaggio che attualmente non è possibile svolgere interamente tramite le schede presenti in R Commander è la creazione dei due *sub set*, ovvero la divisione dell'intero data set in due sotto insiemi di dati che siano, allo stesso tempo, scelti in maniera casuale. Per farlo, saranno necessarie poche righe di codice che verranno inserite nella finestra dello script di R. Come già ricordato in precedenza, a volte è necessario inserire del codice o scrivere del testo all'interno di R Commander, uscendo dal tracciato principale stabilito nell'introduzione a questo elaborato. Ma, lo si ricorda, il pacchetto stesso è un punto di inizio per imparare ad implementare codice da riga di comando, grazie alla finestra di script che indica il codice sorgente che Rcmdr fornisce in automatico. Questo implica che, dopo aver appreso le conoscenze fondamentali nell'utilizzo del pacchetto, si dovrebbe essere in grado di comprendere e imputare alcune righe di codice all'interno dello script. Si utilizza tale occasione per farlo, immettendo solamente quanto necessario a svolgere il compito di divisione del data set in *train set* e *test set*.

Un metodo per procedere prevede l'installazione di un pacchetto esterno denominato *caret* (*Classification And REgression Training*), il quale contiene le funzioni che velocizzano il lavoro di training del modello per problemi sia di regressione che di classificazione. Poiché in questo elaborato il modello che verrà usato è il modello logistico, trattandosi di un problema di classificazione, tale pacchetto semplifica le azioni da compiere e le righe di codice da scrivere.

Per l'installazione del pacchetto si utilizzerà l'interfaccia grafica di R. In Figura 3.33 a pagina 64 si nota l'inserimento del comando²³:

```
install.packages("caret", dependencies = TRUE)
```

necessario a chiedere ad R di installare, tramite il sito *CRAN* già precedentemente introdotto, quanto necessario al suo utilizzo.

Verrà aperta una finestra in cui sono presenti diversi “Mirror”, ovvero dei server che contengono i dati necessari allo scaricamento del pacchetto. Generalmente viene selezionato in automatico il *0-Cloud*, server consigliato dall'azienda RStudio. Cliccando su “OK” verrà installato il pacchetto e le eventuali dipendenze necessarie (ovvero parti di codice non inserite nel pacchetto principale, ma esterne ad esso, che necessitano di essere installate insieme al pacchetto madre per garantirne il pieno funzionamento). Motivo per cui, nel codice inserito si legge l'argomento *dependencies = TRUE*.

Sarà quindi possibile, a questo punto, richiamare il pacchetto all'interno dell'area dello script di Rcmdr come mostrato in Figura 3.34 a pagina 65.

²³Max Kuhn et al. *A Short Introduction to the caret Package*. URL: <https://cran.r-project.org/web/packages/caret/vignettes/caret.html>.

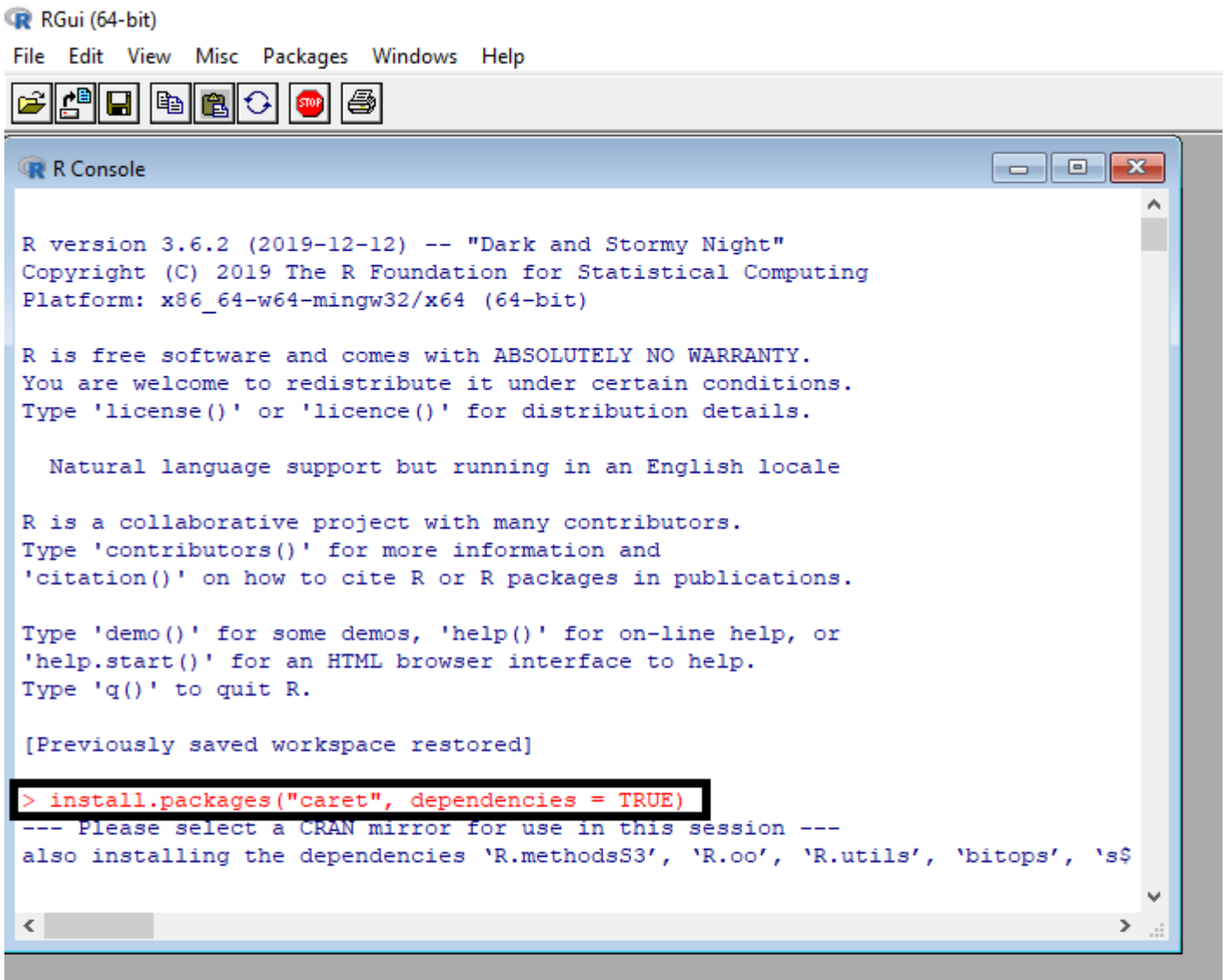


Figura 3.33: Installazione pacchetto “caret”

Poiché all’interno del pacchetto *caret* sono comprese funzioni per il *data splitting*²⁴, ovvero la divisione dei dati proprio con l’obiettivo di creazione di sottoinsiemi, si richiamerà una funzione particolare, *createDataPartition* necessaria a suddividere il data set originario in più data set bilanciati, nei quali

²⁴Max Kuhn et al. *The caret Package*. 2019. URL: <http://topepo.github.io/caret/>.

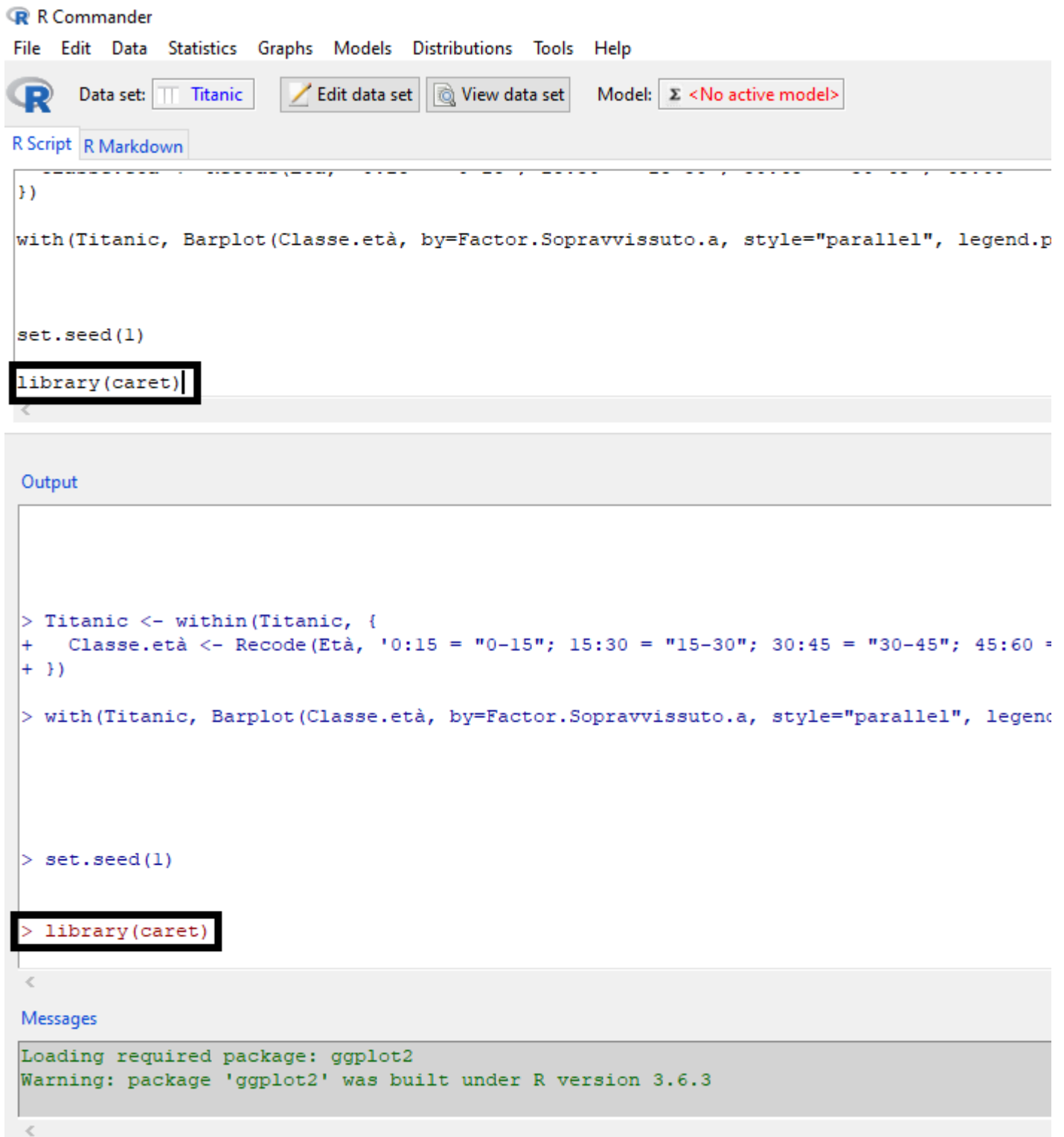


Figura 3.34: Richiamo del pacchetto “caret”

il campionamento avviene in modo casuale all'interno di ciascuna categoria se, ad esempio, l'argomento inserito è di tipo categorico. Tale approccio mantiene un'ordinamento casuale di scelta delle osservazioni rientranti nell'80% del training set e della parte restante, evitando di selezionare manualmente le osservazioni da inserire, correndo il rischio di non considerare alcune categorie di dati. Volendo creare una divisione dell'80% per il test set e del 20% per il training set, nello script di Rcmdr verranno inseriti i seguenti comandi²⁵:

```
train.index <- createDataPartition(
Titanic$Factor.Sopravvissuto.a,
p = 0.8,
list = FALSE,
times = 1)

titanic.train <- Titanic[train.index, ]
titanic.test <- Titanic[-train.index, ]
```

come in Figura 3.35 a pagina 67.

Dove per *list = FALSE* si intende evitare di avere i dati della variabile sotto forma di lista; *times = 1* implica richiedere un'unica divisione del data set; mentre *p = 0.8* serve a specificare la percentuale di osservazioni dell'intero data set che si vogliono considerare. La variabile output sarà un vettore contenente una lista di numeri interi necessari al rimescolamento dei dati in maniera casuale. Con le variabili *titanic.train* e *titanic.test* vengono salvati in esse rispettivamente il primo 80% e il restante 20% (per quest'ultimo è sufficiente inserire un segno negativo di fronte al *train.index* nella variabile *titanic.test* esprimendo il complementare del primo sub set).

Per controllare l'operato ed essere certi che le azioni finora svolte abbiano avuto il loro effetto, è sufficiente inserire un'ulteriore riga di codice, semplicemente per “vedere” come il *titanic.train* è composto. Per farlo, è sufficiente scrivere nello script di R Commander per poi cliccare sul pulsante “Submit” la seguente riga di codice:

```
View(titanic.train)
```

Verrà quindi aperta una finestra di visualizzazione del sub set creato, come mostrato in Figura 3.36 a pagina 68.

Ponendo l'attenzione sulla colonna “Row names” è visibile la mancanza di alcuni elementi che, randomicamente sono stati esclusi per poi essere inseriti nel *test set*.

²⁵Max Kuhn. *Data Splitting*. 2019. URL: <http://topepo.github.io/caret/data-splitting.html#simple-splitting-based-on-the-outcome>.

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA 67

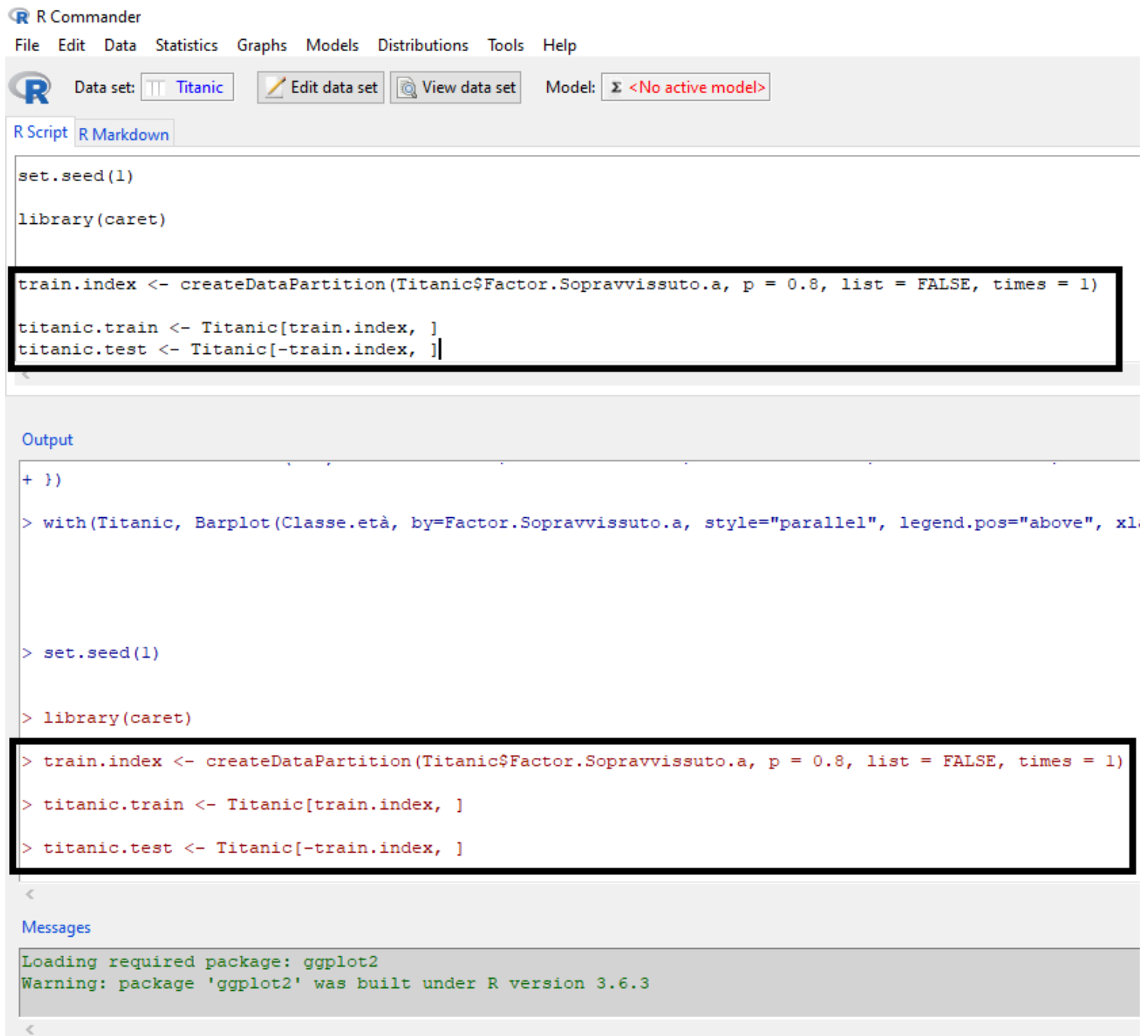


Figura 3.35: Frazionamento del data set in train e test set

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA68

RGui (64-bit) - [Data: titanic.train]

R File

| | row.names | Nome | Gener |
|----|-----------|--|-------|
| 1 | 2 | Mrs. John Bradley (Florence Briggs Thayer) Cumings | F |
| 2 | 3 | Miss. Laina Heikkinen | F |
| 3 | 4 | Mrs. Jacques Heath (Lily May Peel) Futrelle | F |
| 4 | 6 | Mr. James Moran | M |
| 5 | 7 | Mr. Timothy J McCarthy | M |
| 6 | 9 | Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johnson | F |
| 7 | 10 | Mrs. Nicholas (Adele Achem) Nasser | F |
| 8 | 11 | Miss. Marguerite Rut Sandstrom | F |
| 9 | 12 | Miss. Elizabeth Bonnell | F |
| 10 | 13 | Mr. William Henry Saundercock | M |
| 11 | 14 | Mr. Anders Johan Andersson | M |
| 12 | 15 | Miss. Hulda Amanda Adolfina Vestrom | F |
| 13 | 16 | Mrs. (Mary D Kingcome) Hewlett | F |
| 14 | 17 | Master. Eugene Rice | M |
| 15 | 18 | Mr. Charles Eugene Williams | M |
| 16 | 19 | Mrs. Julius (Emelia Maria Vandemoortele) Vander P> | F |
| 17 | 20 | Mrs. Fatima Masselmani | F |
| 18 | 22 | Mr. Lawrence Beesley | M |
| 19 | 23 | Miss. Anna McGowan | F |
| 20 | 24 | Mr. William Thompson Sloper | M |
| 21 | 25 | Miss. Torborg Danira Palsson | F |
| 22 | 27 | Mr. Farred Chehab Emir | M |
| 23 | 28 | Mr. Charles Alexander Fortune | M |
| 24 | 29 | Miss. Ellen O'Dwyer | F |
| 25 | 30 | Mr. Lailo Todoroff | M |
| 26 | 31 | Don. Manuel E Uruchurtu | M |
| 27 | 32 | Mrs. William Augustus (Marie Eugenie) Spencer | F |
| 28 | 34 | Mr. Edward H Wheadon | M |
| 29 | 35 | Mr. Edgar Joseph Meyer | M |
| 30 | 36 | Mr. Alexander Oskar Holverson | M |
| 31 | 37 | Mr. Hanna Mamee | M |
| 32 | 38 | Mr. Ernest Charles Cann | M |

Figura 3.36: Visualizzazione del train set

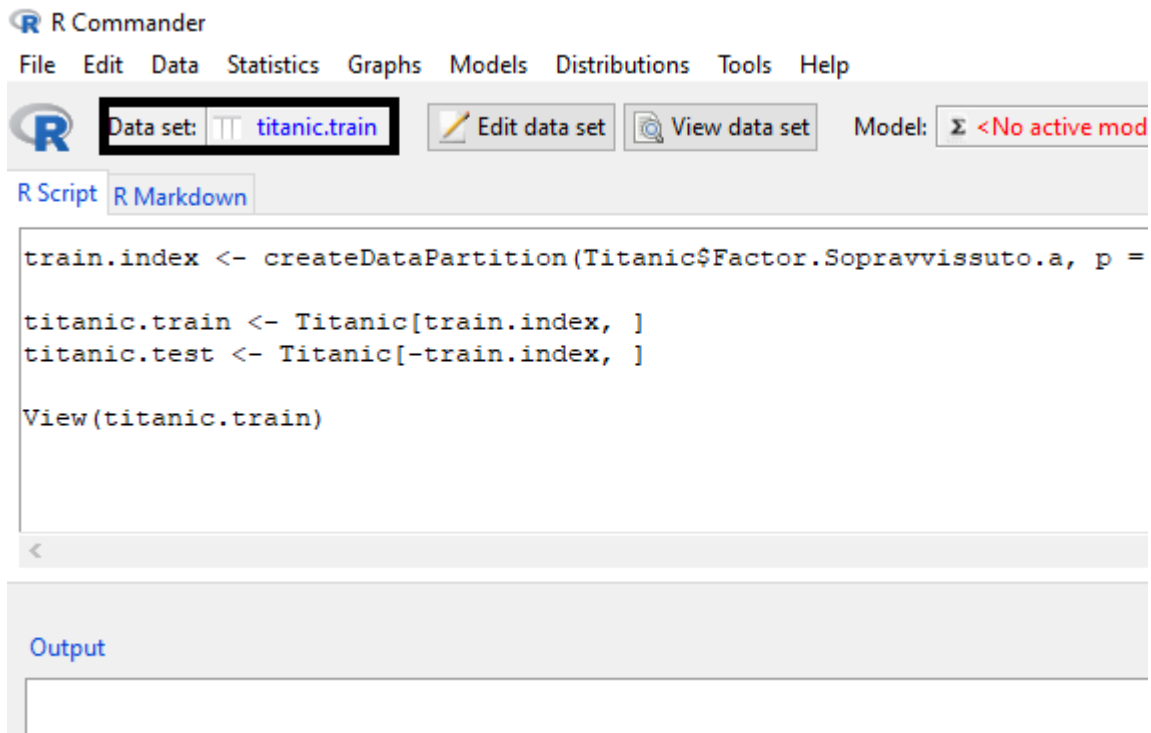


Figura 3.37: Titanic training set

Per procedere all'applicazione della metodologia, si dovrà prima scegliere il data set *titanic.train* che si è creato. Si clicca quindi al di sotto della barra degli strumenti sul pulsante a fianco la scritta *Data set* permettendo l'apertura di una finestra nella quale scegliere il data set corretto. [Figura 3.37 a pagina 69]

Per l'applicazione della regressione logistica, si utilizza il menù *Statistics > Fit models > Generalized linear model* come in Figura 3.38 a pagina 70.

Si otterrà quanto riportato in Figura 3.39 a pagina 71.

Il box di dialogo presenta diverse opzioni da poter modificare: il nome del modello viene generato automaticamente e, nel caso ne fosse creato un altro, verrebbe modificato il numero associato al nome stesso; al centro del box è presente uno spazio nel quale inserire la formula di regressione logistica. Nel campo a sinistra verrà indicata la variabile sulla quale si farà regressione, mentre nel campo a destra i regressori utilizzati. Per completare i campi è possibile inserire i valori a mano, oppure selezionandoli dal box di scelta delle variabili. Cliccando due volte sulla variabile, verrà automaticamente inserita

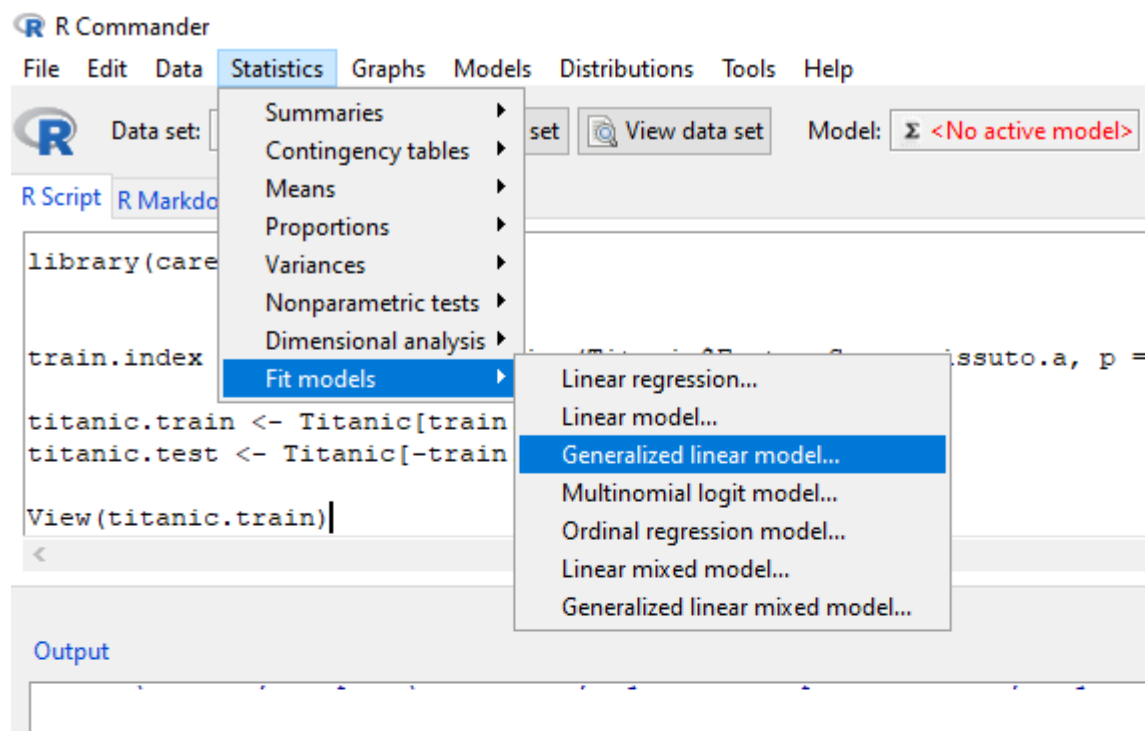


Figura 3.38: Generalized linear model

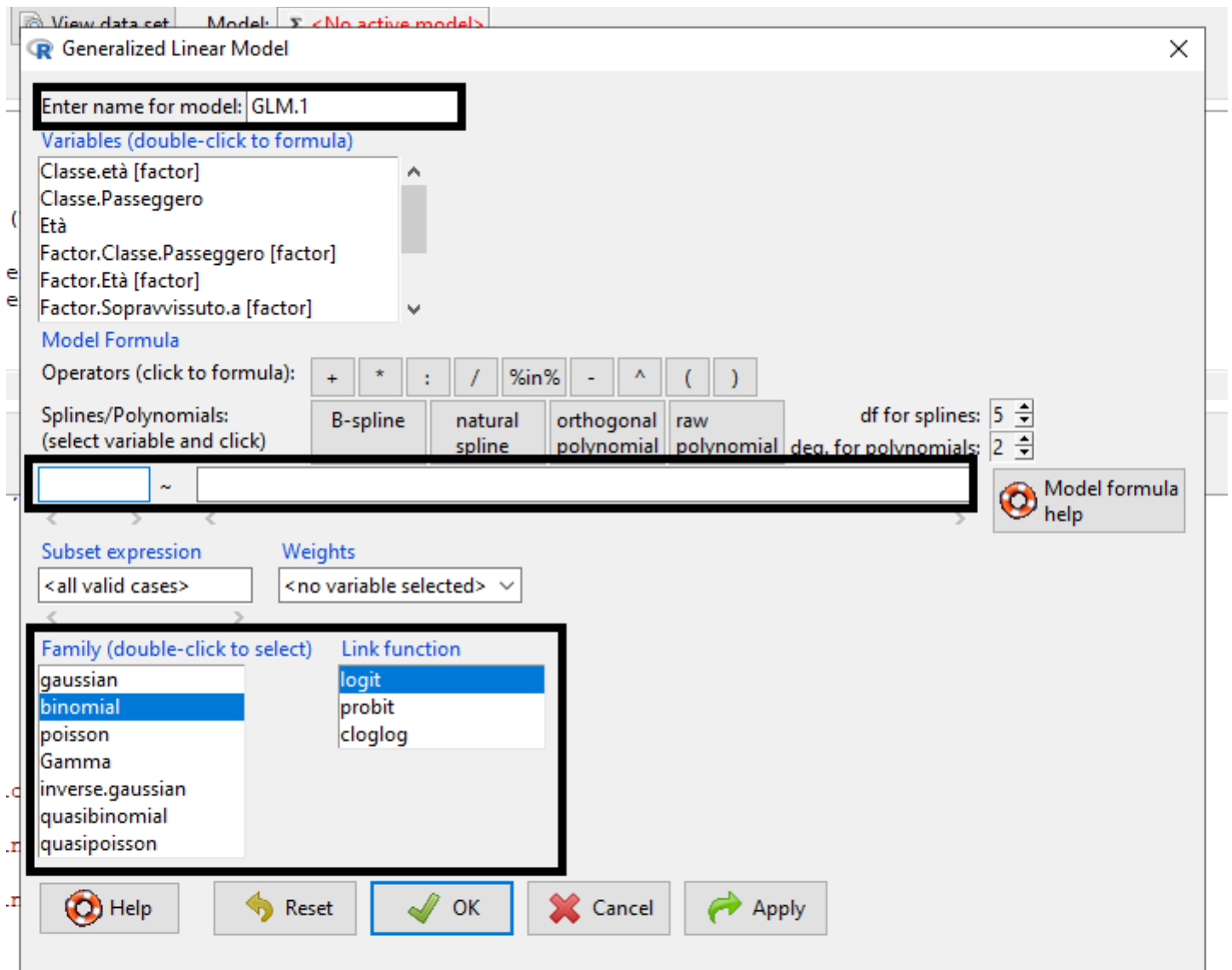


Figura 3.39: Generalized linear model - Box di dialogo

nel campo dei regressori e gli verrà anche aggiunto un segno “+” in quanto R Commander prevede, di default, un modello additivo fra i regressori. L’inserimento di altri operatori è possibile cliccando nella barra degli operatori presenti oppure inserendoli da tastiera. Infine, viene mostrata la famiglia a cui appartiene il modello che, di default, viene preselezionata come “binomial”, indicando che la variabile risposta presenta due fattori, in questo caso il fattore “Sì” e “No” per indicare la sopravvivenza o meno del passeggero. Poiché, come già precedentemente specificato, si userà la funzione logit, essa può essere selezionata nel box delle *link function*.

I valori scelti sono mostrati in Figura 3.40 a pagina 73.

Cliccando su “OK” si ottiene il risultato mostrato in Figura 3.41 a pagina 74.

Sono presentate diverse informazioni di seguito analizzate:

- La prima informazione che possiamo avere dalla schermata di R Commander si trova poco al di sotto della barra degli strumenti: a fianco alla scritta *Model*, in blu, possiamo leggere il modello appena creato e attualmente attivo (dove prima era presente la scritta *<No active model>*);
- Nello script di Rcmdr è visibile il codice sorgente necessario alla creazione del modello. Si ha la funzione **glm** che fa regredire la variabile *factor.Sopravvissuto.a* rispetto ai regressori a destra della formula; sono inoltre già espressi anche il riepilogo della funzione tramite il comando *Summary(logistic.model.1)* e gli *odds ratios*;
- Infine, nella finestra dell’Output si possono visualizzare i risultati che, per motivi di spazio, vengono presentati in due figure differenti (Figura 3.42 a pagina 75 e Figura 3.43 a pagina 76)

Dai risultati nella sezione dell’Output si traggono alcune prime conclusioni:

- Tutte le variabili scelte sono statisticamente significative, ovvero il loro p-value è prossimo allo 0 permettendo il rifiuto dell’ipotesi nulla. Tale affermazione sembra non essere totalmente vera riguardo alla variabile “Genitori.Figli.a.bordo”;
- La variabile che ha un maggiore effetto sulla sopravvivenza del passeggero sembra essere quella relativa al genere, nello specifico maschile, e alla classe del passeggero. Riguardo quest’ultima variabile, essendo il segno del coefficiente negativo, si può affermare che all’aumentare del

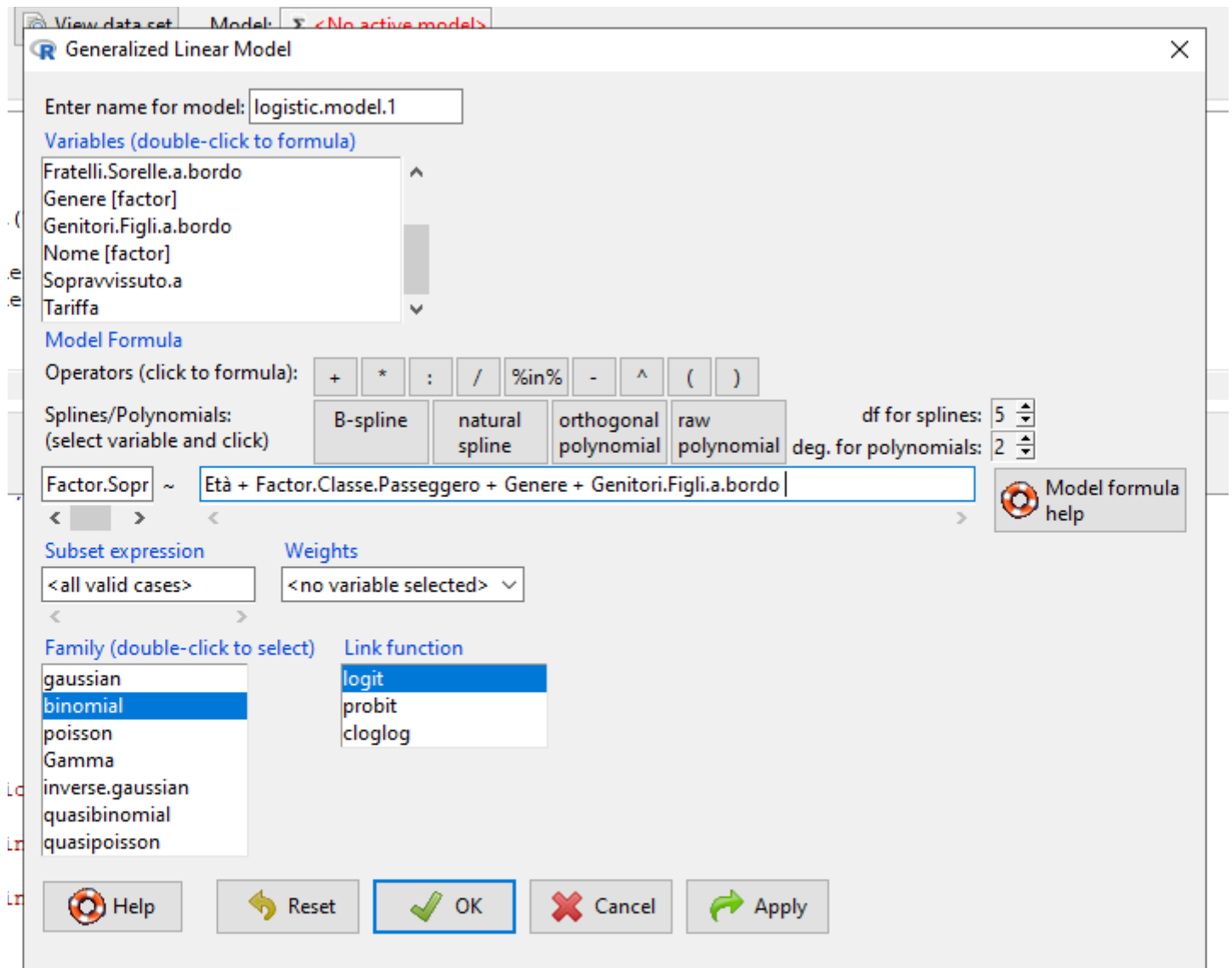


Figura 3.40: GLM scelta delle variabili

valore della classe, da 1 a 3, la probabilità di sopravvivenza decresca. Per la variabile “Genere M” il valore negativo di fronte al coefficiente mostra che all’aumentare del valore (cioè passando da 0 ad 1) si riduce notevolmente la probabilità di sopravvivenza (durante la creazione del

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA74

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: Edit data set View data set Model:

R Script R Markdown

```
View(titanic.train)
```

```
logistic.model.1 <- glm(Factor.Sopravvissuto.a ~ Età + Factor.Classe.Passeggero + Genere + Genitori.Figli.a.bordo, family=binomial(logit),
  data=titanic.train)
summary(logistic.model.1)
exp(coef(logistic.model.1)) # Exponentiated coefficients ("odds ratios")
```

Output

```
> summary(logistic.model.1)
```

Call:

```
glm(formula = Factor.Sopravvissuto.a ~ Età + Factor.Classe.Passeggero +
  Genere + Genitori.Figli.a.bordo, family = binomial(logit),
  data = titanic.train)
```

Deviance Residuals:

| Min | 1Q | Median | 3Q | Max |
|---------|---------|---------|--------|--------|
| -2.6359 | -0.5905 | -0.4354 | 0.6611 | 2.4371 |

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) |
|-------------------------------|-----------|------------|---------|-----------------|
| (Intercept) | 4.189286 | 0.452012 | 9.268 | < 2e-16 *** |
| Età | -0.041154 | 0.008303 | -4.956 | 0.000000719 *** |
| Factor.Classe.Passeggero[T.2] | -1.481225 | 0.303646 | -4.878 | 0.000001071 *** |
| Factor.Classe.Passeggero[T.3] | -2.581335 | 0.289579 | -8.914 | < 2e-16 *** |
| Genere[T.M] | -2.672944 | 0.218478 | -12.234 | < 2e-16 *** |
| Genitori.Figli.a.bordo | -0.332289 | 0.133619 | -2.487 | 0.0129 * |

Messages

```
[15] WARNING: There is only one model in memory.
[16] NOTE: The dataset titanic.train has 710 rows and 12 columns.
```

Figura 3.41: GLM Output

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA75

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: Model:

R Script R Markdown

```
View(titanic.train)
```

```
logistic.model.1 <- glm(Factor.Sopravvissuto.a ~ Età + Factor.Classe.Passeggero + Genere + Genitori.Figli.a.bordo, family=binomial(logit),
  data=titanic.train)
summary(logistic.model.1)
exp(coef(logistic.model.1)) # Exponentiated coefficients ("odds ratios")
```

Output

```
> summary(logistic.model.1)
```

Call:

```
glm(formula = Factor.Sopravvissuto.a ~ Età + Factor.Classe.Passeggero +
  Genere + Genitori.Figli.a.bordo, family = binomial(logit),
  data = titanic.train)
```

Deviance Residuals:

| Min | 1Q | Median | 3Q | Max |
|---------|---------|---------|--------|--------|
| -2.6359 | -0.5905 | -0.4354 | 0.6611 | 2.4371 |

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) |
|-------------------------------|-----------|------------|---------|-----------------|
| (Intercept) | 4.189286 | 0.452012 | 9.268 | < 2e-16 *** |
| Età | -0.041154 | 0.008303 | -4.956 | 0.000000719 *** |
| Factor.Classe.Passeggero[T.2] | -1.481225 | 0.303646 | -4.878 | 0.000001071 *** |
| Factor.Classe.Passeggero[T.3] | -2.581335 | 0.289579 | -8.914 | < 2e-16 *** |
| Genere[T.M] | -2.672944 | 0.218478 | -12.234 | < 2e-16 *** |
| Genitori.Figli.a.bordo | -0.332289 | 0.133619 | -2.487 | 0.0129 * |

Messages

```
[15] WARNING: There is only one model in memory.
[16] NOTE: The dataset titanic.train has 710 rows and 12 columns.
```

Figura 3.42: GLM Summary (1)

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA76

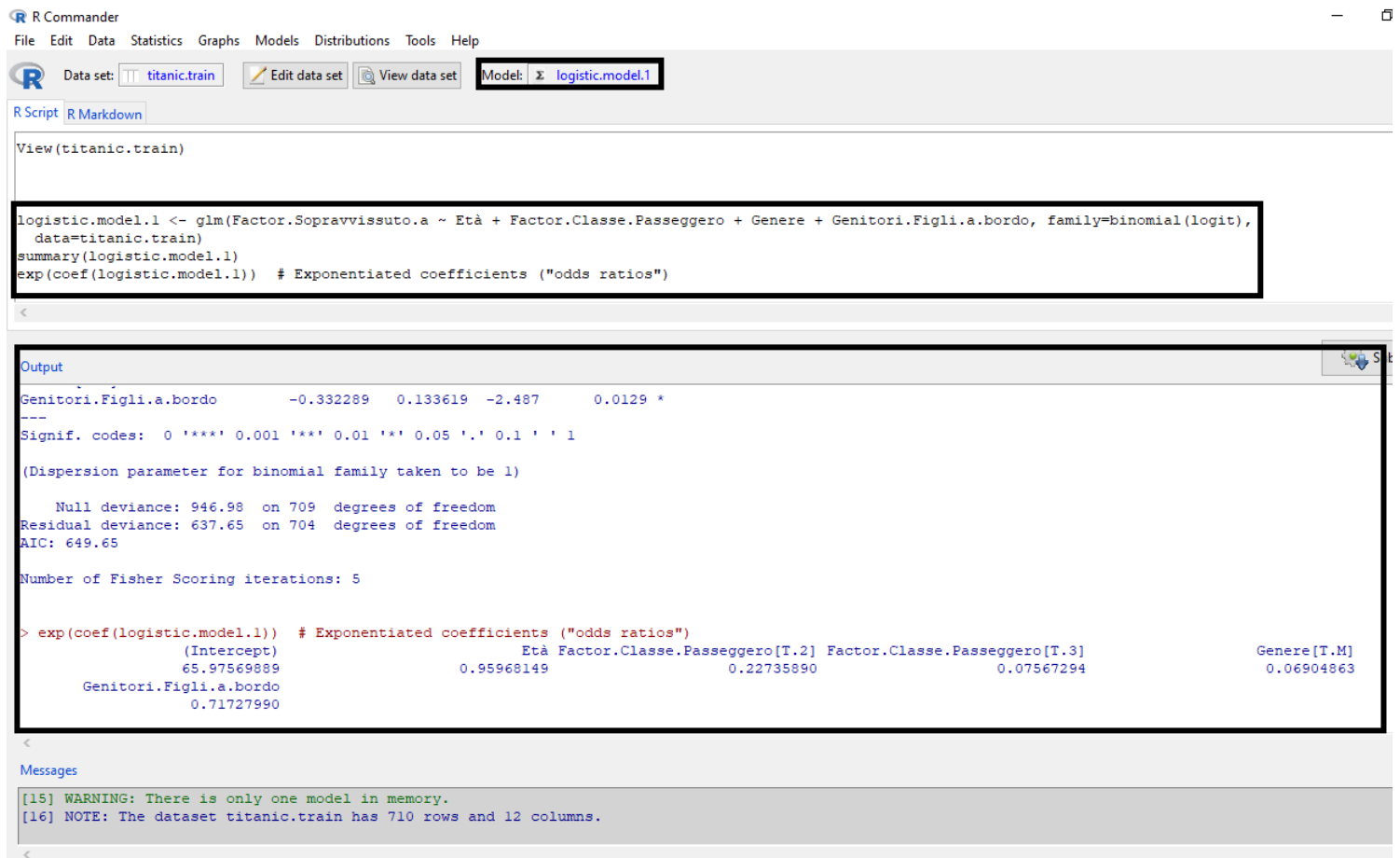


Figura 3.43: GLM Summary (2)

modello, R ha convertito automaticamente il valore “M” in 0 e il valore “F” in 1);

- L’età non sembra essere così importante dal punto di vista della sopravvivenza, anche se si può affermare che maggiore è l’età e minori sono le probabilità di sopravvivenza dato il maggiore impatto del coefficiente negativo;
- Infine, la variabile “Genitori.Figli.a.bordo” è considerata statisticamente significativa anche se il suo p-value non è certamente uguale a 0. La sua significatività potrebbe essere interpretata nel seguente modo: maggiore è il numero di figli a bordo del transatlantico e minore è la probabilità che i genitori cerchino la salvezza lasciando al loro destino una parte della propria famiglia.

Nella Figura 3.43 a pagina 76 l’attenzione si sofferma sulla riga definita dalla funzione `exp(cef(logistic.model.1))` che viene corredata dal commento *Exponentiated coefficients (“odds ratios”)*. Come si è visto, la funzione logit viene applicata agli odds. Con l’applicazione della funzione inversa, ovvero la funzione esponenziale, ciò che si ottengono sono gli “odds ratios”. Si consideri, ad esempio, la variabile genere che presenta un valore di 0.069 048 63. Ciò significa che dato un passeggero di genere femminile, per uno di genere maschile il tasso di sopravvivenza è di circa 14.482 546 58 volte più basso. Considerando invece la variabile età è chiaro come questo ragionamento non sia vero: il tasso di sopravvivenza per una persona che ha un’età superiore ad un’altra è pari solo ad 1.042 012 387 volte in meno.

Una volta applicato il modello logistico al training set, si vuole procedere con l’applicazione al test set.

Per farlo è necessaria l’installazione di un pacchetto che ha il compito di aumentare le funzionalità di R Commander.

Aprendo la finestra di R, si digita quanto segue nella console:

```
install.packages("RcmdrPlugin.UCA")
```

Il pacchetto *RcmdrPlugin.UCA*²⁶ permette l’estensione di alcune funzionalità di Rcmdr tra cui test di casualità, test sulla varianza e, in particolare, funzioni di previsione su modelli attivi.

Richiamando, sempre tramite R, “RcmdrPlugin.UCA” tramite il comando:

²⁶Manuel Munoz-Marquez. *Package ‘RcmdrPlugin.UCA’*. 2018. URL: <https://cran.r-project.org/web/packages/RcmdrPlugin.UCA/RcmdrPlugin.UCA.pdf>.

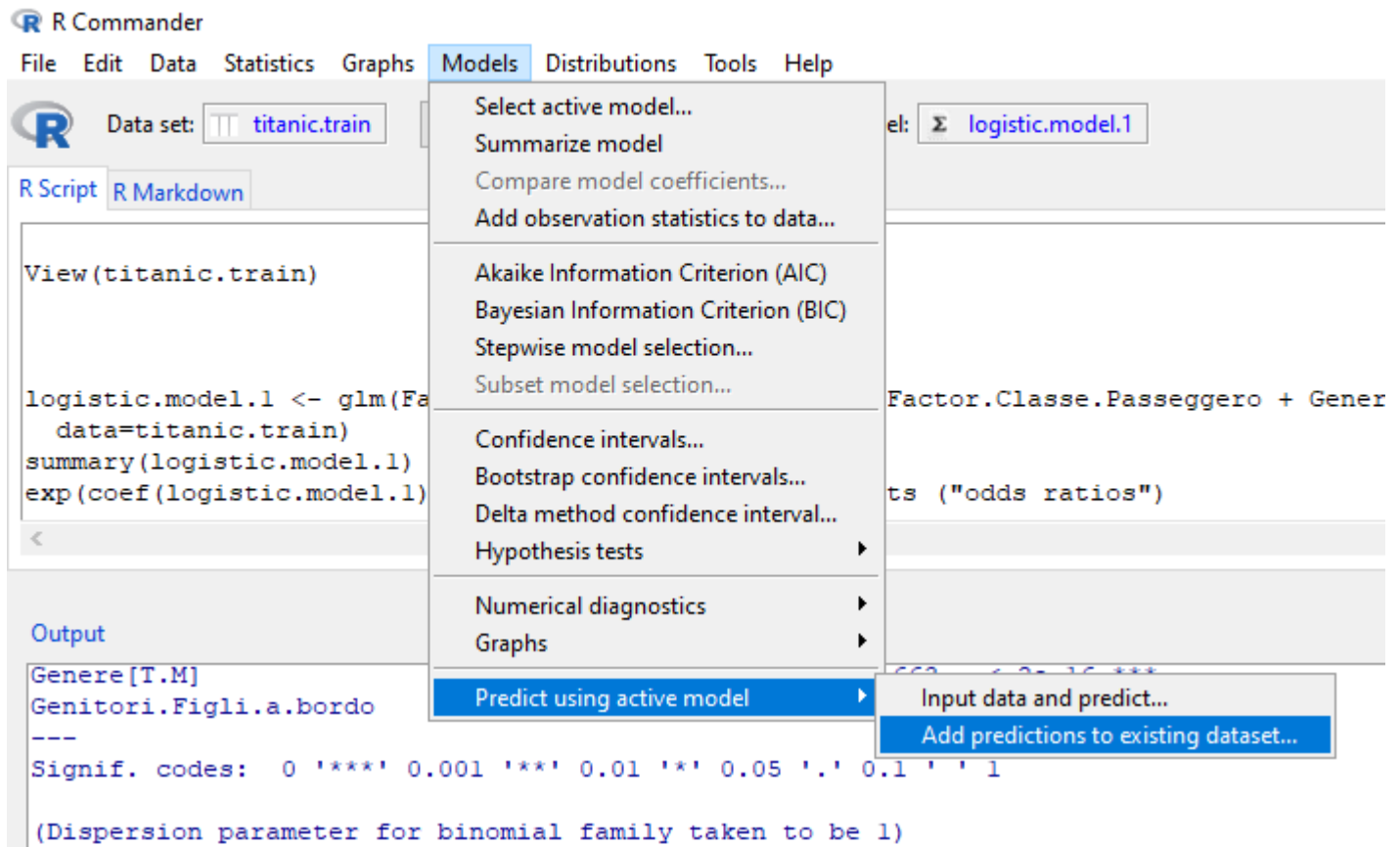


Figura 3.44: Scheda “Predict using active model”

```
library("RcmdrPlugin.UCA")
```

si otterrà l’apertura della finestra di Rcmdr per come è stata vista finora.

Sarà quindi possibile cliccare su *Models > Predict using active model > Add predictions to existing data set* per poter applicare la funzione **predict** al data set di propria scelta, come in Figura 3.44 a pagina 78).

Cliccandovi sopra, si ottiene quanto visibile in Figura 3.45 a pagina 79.

Si seleziona quindi il data set su cui applicare la funzione *predict*, ovvero il *titanic.test* e poi si clicca su “OK”, ottenendo quanto mostrato in Figura 3.46 a pagina 80.

Dove la funzione “Predict” ha i seguenti argomenti:

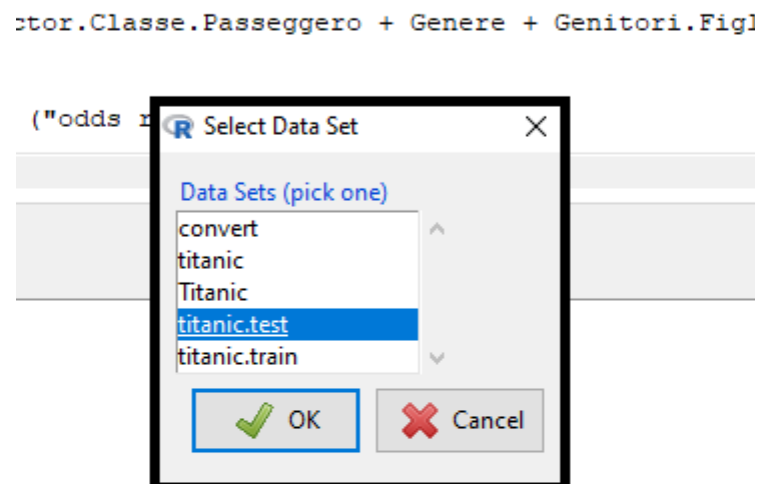


Figura 3.45: “Predict using active model” box di dialogo

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA80

The screenshot shows the R Commander interface. At the top, the menu bar includes File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, and Help. Below the menu bar, the 'Data set:' dropdown is set to 'titanic.train', and the 'Model:' dropdown is set to 'logistic.model.1'. The 'R Script' tab is active, displaying the following R code:

```
logistic.model.1 <- glm(Factor.Sopravvissuto.a ~ Età + Factor.Classe.Passeggero + Genitori.Figli.a.bordo,
  data=titanic.train)
summary(logistic.model.1)
exp(coef(logistic.model.1)) # Exponentiated coefficients ("odds ratios")

titanic.test$fitted.logistic.model.1 <- predict(logistic.model.1, titanic.test)
```

The 'Output' pane shows the results of the model fit:

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

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 946.98  on 709  degrees of freedom
Residual deviance: 637.65  on 704  degrees of freedom
AIC: 649.65

Number of Fisher Scoring iterations: 5

> exp(coef(logistic.model.1)) # Exponentiated coefficients ("odds ratios")
              (Intercept)              Età Factor.Classe.Passeggero[
              65.97569889              0.95968149              0.2273
Genitori.Figli.a.bordo
              0.71727990
```

The 'Messages' pane shows the following notes:

```
[10] NOTE: Output saved to C:/Users/-M-/Desktop/Rcmdr/Tesi_Statistica-Applicata/RComm
[11] NOTE: R workspace saved to C:/Users/-M-/Desktop/Rcmdr/Tesi_Statistica-Applicata/
```

Figura 3.46: predict fitted.logistic.model.1

- Un oggetto sul quale è stato applicato un modello di regressione. In questo caso, si indica il *logistic.model.1*;
- Un dataframe opzionale il quale Rcmdr esaminerà per ritrovare le variabili con le quali predire, in questo caso il *titanic.test*. Nell'applicare la funzione, verranno prese le variabili che sono contenute nel *logistic.model.1*.

Il risultato ottenuto è un vettore \vec{X} contenente valori del campo dei reali (positivi e negativi) che corrispondono a quanto è stato precedentemente definito essere la funzione logaritmo degli *odds*.

Per poter ottenere valori compresi tra 0 ed 1, ovvero le probabilità di appartenere al gruppo dei sopravvissuti oppure dei non sopravvissuti, sarà quindi necessario richiamare la funzione inversa del *logit(x)*, ovvero *invlogit(x)*. Tale funzione necessita di un pacchetto già disponibile all'interno di Rcmdr tramite la scheda *Tools > Load package(s)*, da ricercare sotto il nome di **arm**. Così facendo, in automatico, verranno richiamati quei pacchetti necessari ad Rcmdr per performare quanto richiesto. [Figura 3.47 a pagina 82]

Sarà quindi possibile scrivere direttamente nello script di Rcmdr la funzione:

```
prob.value <-
invlogit(titanic.test$fitted.logistic.model.1)
```

Richiamandola e cliccando sul pulsante “Submit” si otterrà quanto mostrato in Figura 3.48 a pagina 83.

Come si può vedere, si tratta di valori tutti compresi tra 0 ed 1, come vogliono gli assiomi di positività e di certezza contenuti nella definizione assiomatica della probabilità del matematico Andrej Nikolaevič Kolmogorov.

Considerando il modello di regressione logistica binario utilizzato è necessario categorizzare tali valori di probabilità, appartenenti al campo dei valori reali continui, nelle categorie già espresse pari a 0 ed 1. Ipotizzando di considerare che tutte le probabilità espresse che siano maggiori del limite di 0.5 equivalgano ad una maggiore probabilità di sopravvivenza e viceversa, è possibile applicare una brevissima istruzione condizionale definita come **ifelse** affinché i valori previsti possano essere convertiti in una delle due categorie.

Nella finestra dello script si scrive il seguente codice:

```
titanic.predict.output <-
ifelse(titanic.test$fitted.logistic.model.1 > 0.5, 1, 0)
```

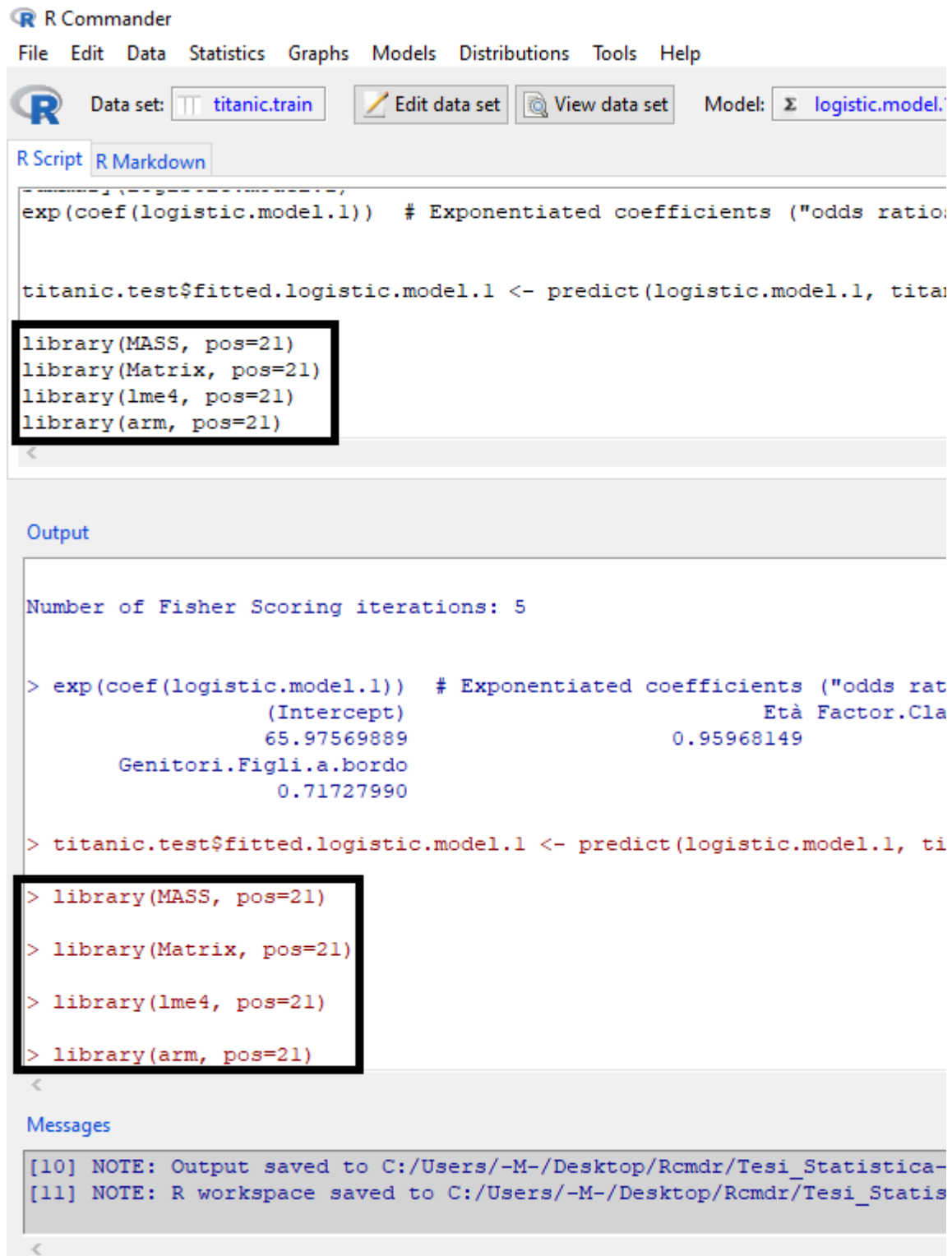


Figura 3.47: Pacchetto “arm”

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA83

```
library(lme4, pos=21)
library(arm, pos=21)
prob.value <- invlogit(titanic.test$fitted.logistic.model.l)
prob.value
```

Output

```
> library(arm, pos=21)
> prob.value <- invlogit(titanic.test$fitted.logistic.model.l)
> prob.value
[1] 0.12234843 0.07548212 0.18548949 0.19698159 0.16557210 0.70415917 0.69551463 0.15142684 0.40691484 0.09640035 0.08456220 0.11799760 0.33992709
[14] 0.59001682 0.88168099 0.11378142 0.09462277 0.12234843 0.20357256 0.09462277 0.06730536 0.39702226 0.23896825 0.26148704 0.81974190 0.27017615
[27] 0.33119620 0.65027764 0.03886390 0.74305493 0.10910118 0.86779632 0.68541242 0.41688348 0.07265993 0.91517444 0.67780908 0.05033103 0.76734058
[40] 0.08456220 0.10574201 0.78034480 0.23156450 0.23156450 0.12234843 0.48812638 0.27836722 0.09115549 0.14115119 0.52053882 0.93985132 0.13551729
[53] 0.67780908 0.59001682 0.66562530 0.83154194 0.15533655 0.95768133 0.65027764 0.56996963 0.92134950 0.96089424 0.81358059 0.81311628 0.96077915
[66] 0.24653278 0.84817636 0.55199176 0.05767897 0.43701817 0.11767966 0.35979366 0.84813610 0.14115119 0.13623550 0.64815929 0.95931815 0.28670907
[79] 0.29519864 0.72642144 0.09462277 0.82574318 0.56214582 0.95019513 0.67337762 0.22432251 0.09462277 0.66533233 0.86303999 0.65709368 0.12561850
[92] 0.07840467 0.12234843 0.20357256 0.10910118 0.18428598 0.92385124 0.67780908 0.52130210 0.68472527 0.78731741 0.77320943 0.06112723 0.06476746
[105] 0.59226503 0.63945260 0.87282136 0.34832660 0.12683662 0.90482272 0.24653278 0.11378142 0.08456220 0.50870018 0.15142684 0.12234843 0.04387558
[118] 0.05767897 0.51898040 0.15996434 0.74188532 0.63133591 0.44716826 0.12683662 0.82574318 0.08456220 0.13623550 0.12683662 0.10191315 0.55985496
[131] 0.14115119 0.05916696 0.12202040 0.70185324 0.45736257 0.38721317 0.09462277 0.83158572 0.27017615 0.64086167 0.28670907 0.11378142 0.81358059
[144] 0.67577434 0.41951970 0.09820769 0.33056026 0.11799760 0.05995691 0.15142684 0.04563478 0.09462277 0.13146480 0.97039245 0.04031216 0.40691484
[157] 0.13146480 0.51977544 0.43701817 0.20357256 0.47785019 0.17179751 0.84774175 0.10191315 0.41791705 0.90482272 0.13146480 0.13146480 0.01613693
[170] 0.71203297 0.90149102 0.11799760 0.10574201 0.53948406 0.82525759 0.79362661 0.96793388
```

Figura 3.48: Invlogit

La struttura di tale istruzione è equivalente a `ifelse(test, yes, no)`: ciò significa che, se il test scelto è vero, l'output sarà "yes", altrimenti "no". Nel caso in questione, se il valore continuo della probabilità è maggiore del valore 0.5 tale elemento verrà convertita nel valore 1, e viceversa. Volendo controllarne l'output, si può usare la funzione `head` applicata a `titanic.predict.output`, ottenendo quanto mostrato in Figura 3.49 a pagina 85.

La variabile `titanic.predict.output` conterrà tanti valori 0 o 1 quante sono le osservazioni contenute nel *test set*.

Il passaggio ulteriore che bisogna effettuare per comprendere la validità del modello logistico utilizzato è quello di confrontare i risultati ottenuti con gli attuali valori presenti nella variabile "Factor.Sopravvissuto.a" del train set. Ovvero, in breve, controllare l'**accuratezza** della previsione.

Saranno necessarie alcune righe di codice per ottenere tale risultato:

```
errore.classificazione <- mean(
titanic.predict.output != titanic.test$Sopravvissuto.a)

print(paste("Accuratezza previsione: ",
1 - errore.classificazione))
```

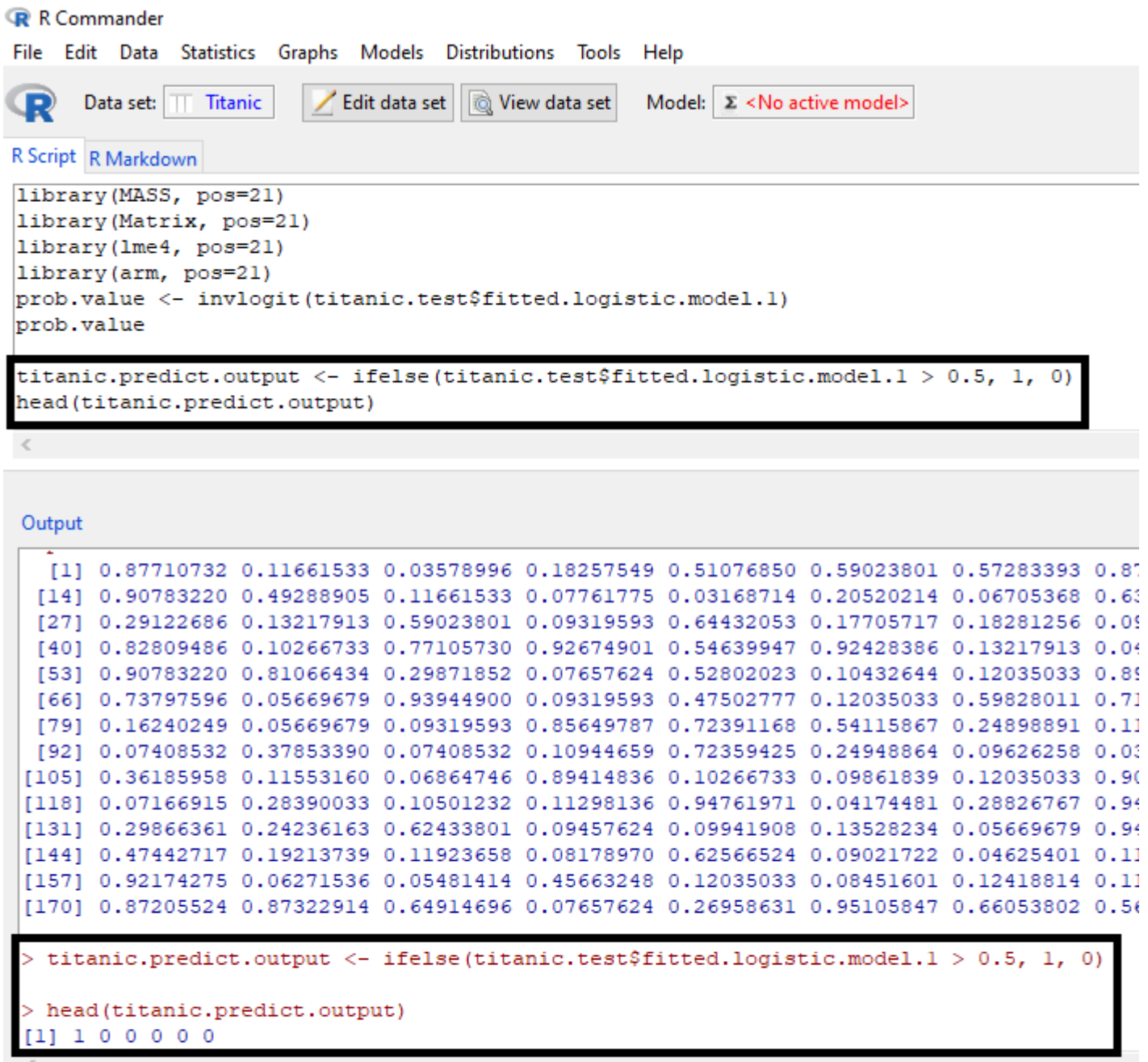
Dove:

- La variabile "errore.classificazione" contiene la media dei valori che sono stati predetti e che sono diversi dagli effettivi;
- E la seconda riga di comando *print* incolla e stampa a schermo i due elementi inseriti come argomento. Se si ha a disposizione l'errore di classificazione, l'accuratezza sarà calcolata come complementare dell'errore.

L'output è visibile in Figura 3.50 a pagina 86.

L'accuratezza della previsione si attesta intorno all'83.62%. Questo significa che se si avesse a disposizione un'ulteriore lista di passeggeri della *RMS Titanic*, senza la conoscenza preventiva dell'esito della variabile "Sopravvivenza", si sarebbe in grado di classificare con tale precisione i sopravvissuti da coloro che non lo sono stati.

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA 85



The screenshot shows the R Commander interface. The 'Data set' dropdown is set to 'Titanic'. The 'Model' dropdown shows '<No active model>'. The 'R Script' tab is active, displaying the following R code:

```
library(MASS, pos=21)
library(Matrix, pos=21)
library(lme4, pos=21)
library(arm, pos=21)
prob.value <- invlogit(titanic.test$fitted.logistic.model.1)
prob.value

titanic.predict.output <- ifelse(titanic.test$fitted.logistic.model.1 > 0.5, 1, 0)
head(titanic.predict.output)
```

The 'Output' pane shows the result of the code execution, displaying a vector of predicted values for the Titanic dataset. The output is a 1x170 matrix of 0s and 1s, representing the predicted survival status for each individual in the dataset. The first few rows of the output are:

```
[1] 0.87710732 0.11661533 0.03578996 0.18257549 0.51076850 0.59023801 0.57283393 0.87
[14] 0.90783220 0.49288905 0.11661533 0.07761775 0.03168714 0.20520214 0.06705368 0.63
[27] 0.29122686 0.13217913 0.59023801 0.09319593 0.64432053 0.17705717 0.18281256 0.09
[40] 0.82809486 0.10266733 0.77105730 0.92674901 0.54639947 0.92428386 0.13217913 0.04
[53] 0.90783220 0.81066434 0.29871852 0.07657624 0.52802023 0.10432644 0.12035033 0.89
[66] 0.73797596 0.05669679 0.93944900 0.09319593 0.47502777 0.12035033 0.59828011 0.71
[79] 0.16240249 0.05669679 0.09319593 0.85649787 0.72391168 0.54115867 0.24898891 0.11
[92] 0.07408532 0.37853390 0.07408532 0.10944659 0.72359425 0.24948864 0.09626258 0.03
[105] 0.36185958 0.11553160 0.06864746 0.89414836 0.10266733 0.09861839 0.12035033 0.90
[118] 0.07166915 0.28390033 0.10501232 0.11298136 0.94761971 0.04174481 0.28826767 0.94
[131] 0.29866361 0.24236163 0.62433801 0.09457624 0.09941908 0.13528234 0.05669679 0.94
[144] 0.47442717 0.19213739 0.11923658 0.08178970 0.62566524 0.09021722 0.04625401 0.11
[157] 0.92174275 0.06271536 0.05481414 0.45663248 0.12035033 0.08451601 0.12418814 0.11
[170] 0.87205524 0.87322914 0.64914696 0.07657624 0.26958631 0.95105847 0.66053802 0.56
```

The output is displayed in a table format with 170 rows and 1 column. The first row of the output is highlighted in red. The output is a vector of predicted values for the Titanic dataset, where 1 indicates survival and 0 indicates non-survival.

```
> titanic.predict.output <- ifelse(titanic.test$fitted.logistic.model.1 > 0.5, 1, 0)
> head(titanic.predict.output)
[1] 1 0 0 0 0 0
```

Figura 3.49: Titanic predict Output

CAPITOLO 3. TITANIC DATASET: ANALISI ESPLORATIVA E PREDITTIVA 86

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: **Titanic** Edit data set View data set Model: **<No active model>**

R Script R Markdown

```
library(arm, pos=21)
prob.value <- invlogit(titanic.test$fitted.logistic.model.1)
prob.value

titanic.predict.output <- ifelse(titanic.test$fitted.logistic.model.1 > 0.5, 1, 0)
head(titanic.predict.output)

errore.classificazione <- mean(titanic.predict.output != titanic.test$Sopravvissuto.a)
print(paste("Accuratezza previsione: ", 1 - errore.classificazione))
```

Output

```
[66] 0.73797596 0.05669679 0.93944900 0.09319593 0.47502777 0.12035033 0.59828011 0.7163849
[79] 0.16240249 0.05669679 0.09319593 0.85649787 0.72391168 0.54115867 0.24898891 0.1166153
[92] 0.07408532 0.37853390 0.07408532 0.10944659 0.72359425 0.24948864 0.09626258 0.0363830
[105] 0.36185958 0.11553160 0.06864746 0.89414836 0.10266733 0.09861839 0.12035033 0.9047958
[118] 0.07166915 0.28390033 0.10501232 0.11298136 0.94761971 0.04174481 0.28826767 0.9420311
[131] 0.29866361 0.24236163 0.62433801 0.09457624 0.09941908 0.13528234 0.05669679 0.9420311
[144] 0.47442717 0.19213739 0.11923658 0.08178970 0.62566524 0.09021722 0.04625401 0.1166153
[157] 0.92174275 0.06271536 0.05481414 0.45663248 0.12035033 0.08451601 0.12418814 0.1166153
[170] 0.87205524 0.87322914 0.64914696 0.07657624 0.26958631 0.95105847 0.66053802 0.5634689

> titanic.predict.output <- ifelse(titanic.test$fitted.logistic.model.1 > 0.5, 1, 0)
> head(titanic.predict.output)
[1] 1 0 0 0 0 0

> errore.classificazione <- mean(titanic.predict.output != titanic.test$Sopravvissuto.a)
> print(paste("Accuratezza previsione: ", 1 - errore.classificazione))
[1] "Accuratezza previsione: 0.836158192090396"
```

Messages

logit

Figura 3.50: Accuratezza della previsione del modello logistico

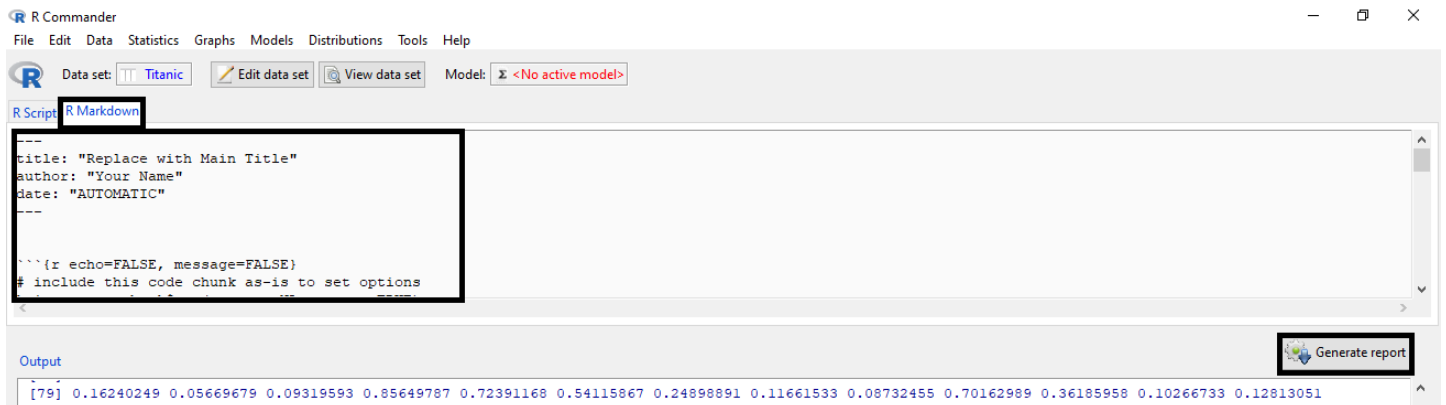


Figura 3.51: R Markdown Template

3.6 Esportazione dei risultati: R Markdown

Ciò che segue non vuole essere un'esposizione completa delle potenzialità di tale strumento, ma solamente una introduzione a quanto può essere prodotto come risultato della propria elaborazione, rimanendo all'interno della filosofia *point-and-click* fin qui seguita.

Markdown è un *linguaggio di markup* per la formattazione di *plain-text* ovvero testo puro, non formattato (per fare un esempio, i file con estensione **.txt** sono file in *plain-text*)²⁷. L'utilizzo di tale linguaggio si discosta da quello dei software definiti *WYSIWYG* (*What You See Is What You Get*): all'interno di programmi come Microsoft Word per la formattazione del testo è sufficiente cliccare specifici pulsanti ottenendo visivamente e nell'immediato il risultato voluto.

All'interno del linguaggio Markdown si aggiungono al testo dei marcatori per denotare quale parola o frase deve essere visualizzata con una formattazione differente. Per indicare ad esempio un titolo, sarà sufficiente aggiungere il simbolo **#** seguito dal testo.

R Commander presenta all'interno dell'interfaccia una scheda denominata "R Markdown" che ha lo scopo di raccogliere e trasformare in linguaggio Markdown tutto il codice prodotto nello script di R Commander. [Figura 3.51 a pagina 87]

Il vantaggio di avere tale strumento integrato in R Commander è quello di poter esportare il lavoro svolto in differenti formati (siti web, documenti, libri,

²⁷Matt Cone. *Getting Started: What is Markdown?* URL: <https://www.markdownguide.org/getting-started/>.

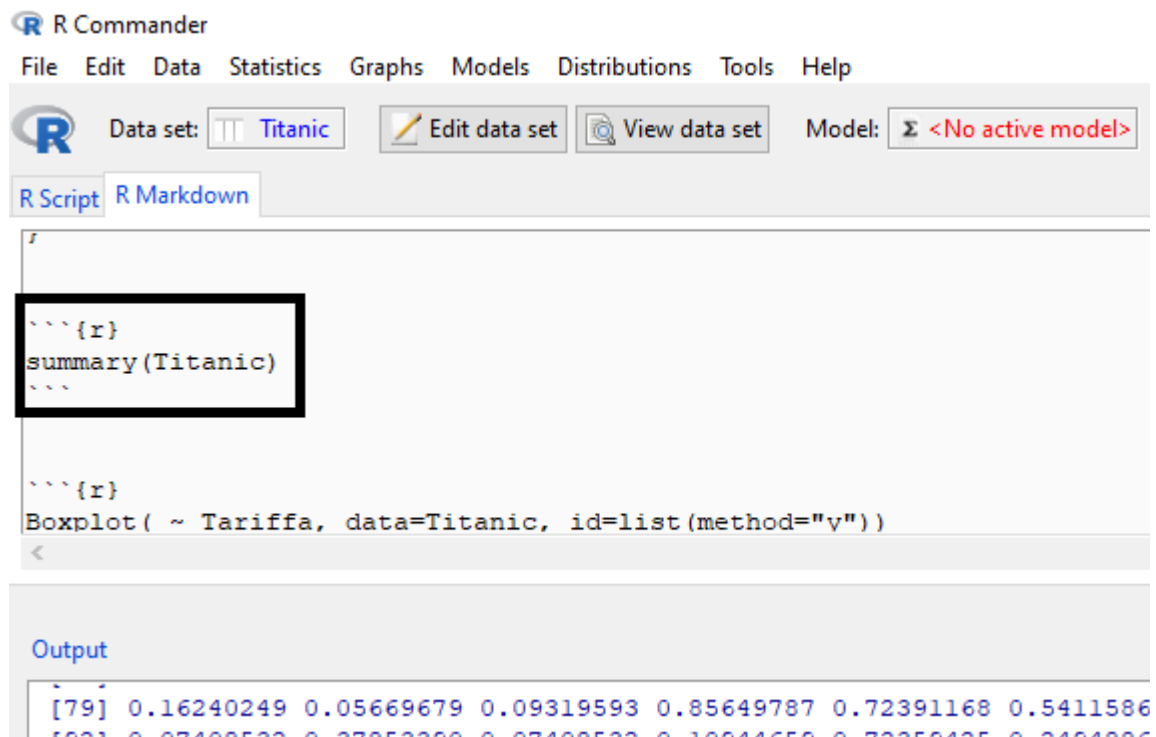


Figura 3.52: Chunks in R Markdown

presentazioni (simili a PowerPoint) e note) ottenendo un output visibilmente professionale e pronto per essere condiviso, senza ulteriori modifiche.

Inoltre, a differenza di Microsoft Word che non permette il copia e incolla del testo in un'altro software testuale mantenendo intatta la formattazione, un testo in Markdown può essere copiato e visualizzato mantenendo inalterato l'aspetto di quanto scritto.

All'interno del testo prodotto da R Markdown sono presenti i *chunks* di codice, ovvero "pezzi" di codice riconoscibili perché inseriti all'interno di delimitatori (tre apostrofi), come in Figura 3.52 a pagina 88.

Una volta completato il lavoro di analisi dei dati, sarà sufficiente modificare l'intestazione della finestra di R Markdown nella quale viene richiesto il titolo della ricerca, l'autore dell'elaborato e la data. Se nella data viene lasciato quanto già presente di default, ovvero "AUTOMATIC", verrà automaticamente espressa la data nella quale viene creato lo scritto compresi l'ora, i minuti e i secondi. Per modificare più comodamente il testo Markdown,

piuttosto che usare la piccola finestra che si ha a disposizione nell'interfaccia, si può cliccare su *Edit > Edit R Markdown document* per ottenere quanto si vede in Figura 3.53 a pagina 90.

Una volta completata la modifica e cliccato sul pulsante “OK”, si potrà procedere cliccando su “Generate report” a destra dell'interfaccia, ottenendo il box di dialogo nel quale scegliere la tipologia di formato di output, come in Figura 3.54 a pagina 91.

Volendo ottenere, ad esempio, un PDF dell'elaborato, lo si seleziona e si clicca su “OK”. R si occuperà di tradurre tutti i *chunks* di codice e di formare il documento finale. L'output di esempio è mostrato in Figura 3.55 a pagina 92. Allo stesso modo, sarà possibile ottenere un prodotto in formato HTML oppure word.

Quanto appena visto rientra all'interno della filosofia del *Literate Programming*, ovvero la scrittura di codice sorgente con lo scopo di spiegare al lettore cosa si vuol far fare al computer, piuttosto che scrivere codice solamente per far comprendere alla macchina quale sia il suo compito. Tale approccio implica l'inserimento di commenti al codice, ma anche un output che possa inglobare sia codice che commenti in un testo che sia facilmente leggibile e comprensibile da chiunque sia in grado di capire il linguaggio statistico ed informatico sottostante. In tale ottica si ottengono studi o ricerche che possano essere riproducibili in quanto si include al loro interno anche il codice scritto e testato, quindi funzionante.

Nel PDF mostrato era presente solamente il codice sorgente. Inserendovi anche i commenti, si può ottenere quanto visibile in Figura 3.56 a pagina 93.

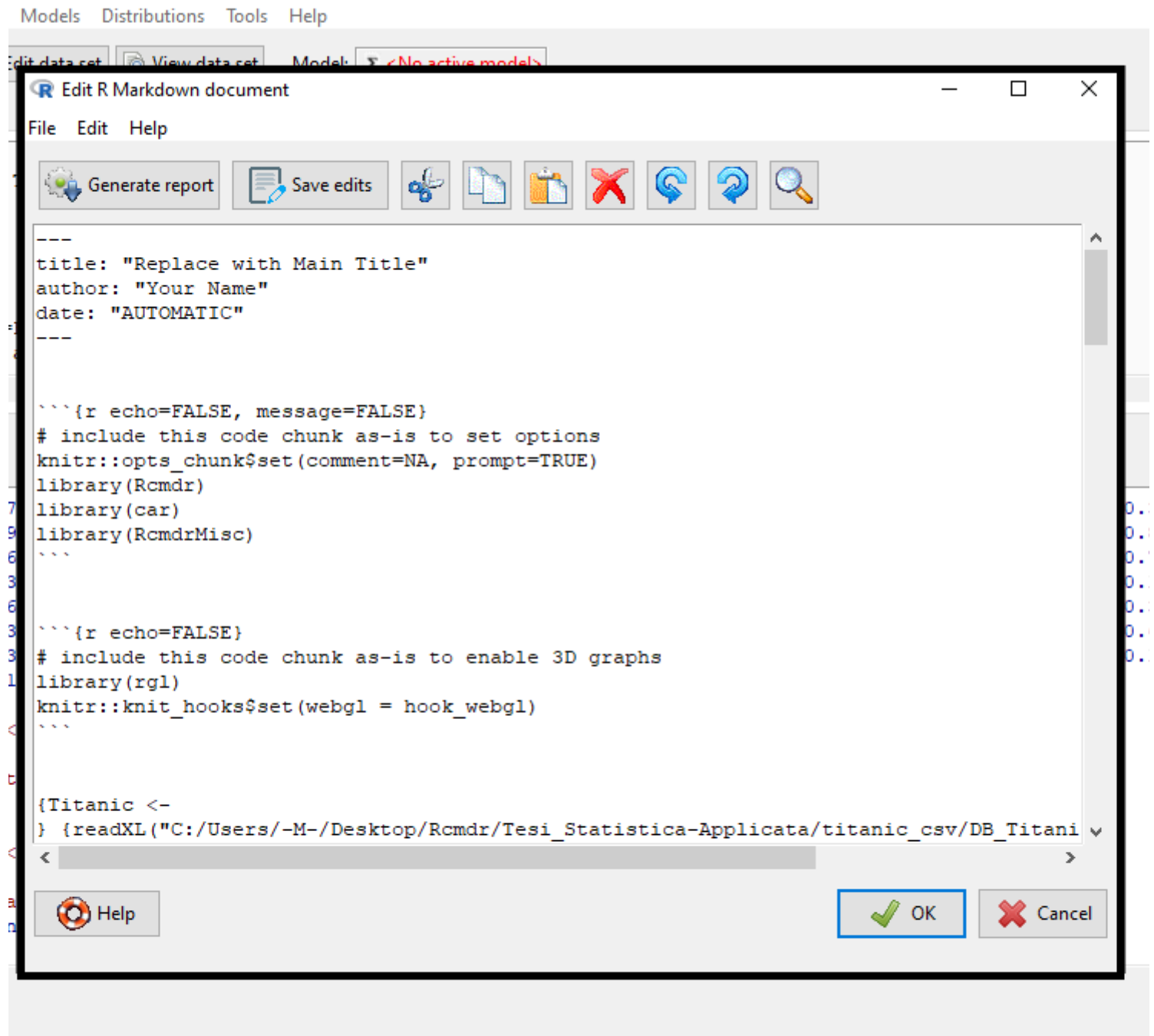


Figura 3.53: Modifica del testo Markdown

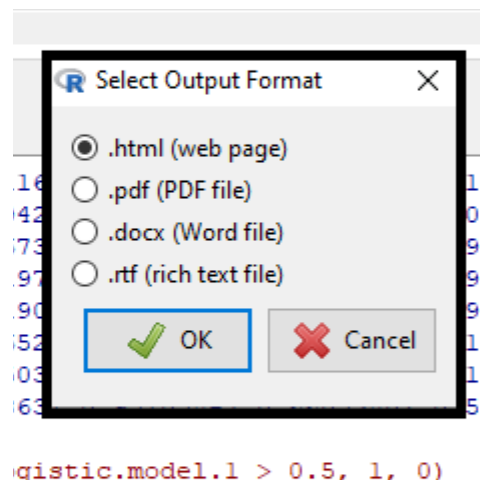


Figura 3.54: Scelta del formato Markdown

R Commander Graphical User Interface (GUI): funzionamento ed applicazione.

Alessandro Maccario

02/08/2020

```
## Warning: package 'Rcmdr' was built under R version 3.6.3
## Warning: package 'RcmdrMisc' was built under R version 3.6.3
## Warning: package 'car' was built under R version 3.6.3
## Warning: package 'sandwich' was built under R version 3.6.3
## Warning: package 'effects' was built under R version 3.6.3
> Titanic <-
+ readXL(paste("C:/Users/~M-/Desktop/Rcmdr",
+ "/Tesi_Statistica-Applicata/titanic_csv",
+ "/DB_Titanic_Corretto/db/titanic.xlsx", sep = ""),
+ rownames=FALSE, header=TRUE, na="NA",
+ sheet="titanic", stringsAsFactors=TRUE
+ )
> summary(Titanic)
```

| | Nome | Genere | Eta |
|------------------------------------|------------------------|-----------------|---------------|
| Capt. Edward Gifford Crosby | : 1 | F:314 | Min. : 0.42 |
| Col. John Weir | : 1 | M:573 | 1st Qu.:20.25 |
| Col. Oberst Alfons Simonius-Blumer | : 1 | | Median :28.00 |
| Don. Manuel E Uruchurtu | : 1 | | Mean :29.47 |
| Dr. Alfred Pain | : 1 | | 3rd Qu.:38.00 |
| Dr. Alice (Farnham) Leader | : 1 | | Max. :80.00 |
| (Other) | :881 | | |
| Fratelli.Sorelle.a.bordo | Genitori.Figli.a.bordo | | Tariffa |
| Min. :0.0000 | Min. :0.0000 | Min. : 0.000 | |
| 1st Qu.:0.0000 | 1st Qu.:0.0000 | 1st Qu.: 7.925 | |
| Median :0.0000 | Median :0.0000 | Median : 14.454 | |
| Mean :0.5254 | Mean :0.3833 | Mean : 32.305 | |
| 3rd Qu.:1.0000 | 3rd Qu.:0.0000 | 3rd Qu.: 31.137 | |
| Max. :8.0000 | Max. :6.0000 | Max. :512.329 | |
| Classe.Passeggero | Sopravvissuto.a | | |
| Min. :1.000 | Min. :0.0000 | | |
| 1st Qu.:2.000 | 1st Qu.:0.0000 | | |
| Median :3.000 | Median :0.0000 | | |
| Mean :2.306 | Mean :0.3856 | | |
| 3rd Qu.:3.000 | 3rd Qu.:1.0000 | | |
| Max. :3.000 | Max. :1.0000 | | |

Figura 3.55: Markdown output in PDF

R Commander Graphical User Interface (GUI): funzionamento ed applicazione.

Alessandro Maccario

02/08/2020

```
## Warning: package 'Rcmdr' was built under R version 3.6.3
## Warning: package 'RcmdrMisc' was built under R version 3.6.3
## Warning: package 'car' was built under R version 3.6.3
## Warning: package 'sandwich' was built under R version 3.6.3
## Warning: package 'effects' was built under R version 3.6.3
```

Caricamento del data set: la prima riga del foglio Excel contiene le variabili in studio:

```
> Titanic <-
+ readXL(paste("C:/Users/~M-/Desktop/Rcmdr",
+ "/Tesi_Statistica-Applicata/titanic_csv",
+ "/DB_Titanic_Corretto/db/titanic.xlsx", sep = ""),
+ rownames=FALSE, header=TRUE, na="NA",
+ sheet="titanic", stringsAsFactors=TRUE
+ )
```

Si richiama la funzione summary() per ottenere un riassunto dei valori principali di ogni variabile:

```
> summary(Titanic)
```

| | Nome | Genere | Eta |
|------------------------------------|------------------------|-----------------|---------------|
| Capt. Edward Gifford Crosby | : 1 | F:314 | Min. : 0.42 |
| Col. John Weir | : 1 | M:573 | 1st Qu.:20.25 |
| Col. Oberst Alfons Simonius-Blumer | : 1 | | Median :28.00 |
| Don. Manuel E Uruchurtu | : 1 | | Mean :29.47 |
| Dr. Alfred Pain | : 1 | | 3rd Qu.:38.00 |
| Dr. Alice (Farnham) Leader | : 1 | | Max. :80.00 |
| (Other) | :881 | | |
| Fratelli.Sorelle.a.bordo | Genitori.Figli.a.bordo | Tariffa | |
| Min. :0.0000 | Min. :0.0000 | Min. : 0.000 | |
| 1st Qu.:0.0000 | 1st Qu.:0.0000 | 1st Qu.: 7.925 | |
| Median :0.0000 | Median :0.0000 | Median : 14.454 | |
| Mean :0.5254 | Mean :0.3833 | Mean : 32.305 | |
| 3rd Qu.:1.0000 | 3rd Qu.:0.0000 | 3rd Qu.: 31.137 | |
| Max. :8.0000 | Max. :6.0000 | Max. :512.329 | |
| Classe.Passeggero | Sopravvissuto.a | | |
| Min. :1.000 | Min. :0.0000 | | |
| 1st Qu.:2.000 | 1st Qu.:0.0000 | | |
| Median :3.000 | Median :0.0000 | | |
| Mean :2.306 | Mean :0.3856 | | |

Figura 3.56: Markdown output in PDF Commentato

CONCLUSIONI

Quanto svolto in tale elaborato ha voluto dimostrare la semplicità d'uso di un pacchetto esterno del software per l'analisi statistica R. L'interfaccia grafica, seppur scarna e meno elaborata rispetto all'*IDE* RStudio, rimane *user-friendly* e di comprensione pressoché immediata nel suo utilizzo.

Un'intera analisi statistica può essere svolta facilmente anche se presenta alcuni limiti in termini di *data visualization* (di rappresentazione visuale) o di metodologie statistiche che possono essere applicate ai dati. Da rimarcare che, dopo essersi ambientati e aver compreso il codice prodotto in automatico nella finestra della script, l'utente dovrebbe essere in grado di poter inserire del proprio codice per accrescere i risultati ottenibili.

Infine, in termini di esportazione dei risultati, R Commander integra anche lo strumento R Markdown, linguaggio di *mark-up* tanto potente quanto flessibile per realizzare una versione condivisibile del proprio elaborato e che rientra nella filosofia del *Literate programming*, filosofia introdotta e sostenuta da Donald Ervin Knuth, matematico, informatico e professore emerito della *Stanford University*, che in un suo articolo pubblicato in *The Computer Journal* nel 1992 scrisse: «Let us change our traditional attitude to the construction of programs: instead of imagining that our main task is to instruct a *computer* what to do, let us concentrate rather on explaining to *human beings* what we want a computer to do.»²⁸

²⁸Donald Ervin Knuth. “Literate programming”. In: *The Computer Journal* 27.2 (1984), pp. 97–111.

Bibliografia e Sitografia

- CNN. *Titanic Fast Facts*. 2020. URL: <https://edition.cnn.com/2013/09/30/us/titanic-fast-facts/index.html>.
- Commerce United States Senate, Committee on. *Titanic Disaster*. 1912. URL: <https://www.senate.gov/artandhistory/history/resources/pdf/TitanicReport.pdf>.
- Cone, Matt. *Getting Started: What is Markdown?* URL: <https://www.markdownguide.org/getting-started/>.
- CRAN. URL: <https://cran.r-project.org/index.html>.
- Elinder, Mikael e Oscar Erixson. “Gender, social norms, and survival in maritime disasters”. In: *Proceedings of the National Academy of Sciences* 109.33 (2012).
- Fox, John. *R-Commander*. URL: <https://www.rcommander.com/>.
- *Using the R commander: A point-and-click interface for R*. CRC Press, 2016.
- Hat, Red. *Cos’è un’ambiente di sviluppo integrato (IDE)*. URL: <https://www.redhat.com/it/topics/middleware/what-is-ide>.
- James, Gareth, Daniela Witten et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.
- Johnson, Richard Arnold, Dean W Wichern et al. *Applied multivariate statistical analysis*. Vol. 5. 8. Prentice hall Upper Saddle River, NJ, 2002.
- Knuth, Donald Ervin. “Literate programming”. In: *The Computer Journal* 27.2 (1984), pp. 97–111.
- Kuhn, Max. *Data Splitting*. 2019. URL: <http://topepo.github.io/caret/data-splitting.html#simple-splitting-based-on-the-outcome>.
- Kuhn, Max et al. *A Short Introduction to the caret Package*. URL: <https://cran.r-project.org/web/packages/caret/vignettes/caret.html>.
- *The caret Package*. 2019. URL: <http://topepo.github.io/caret/>.
- Masarotto, G. e S.M. Iacus. *Laboratorio di statistica con R*. 2003.
- Munoz-Marquez, Manuel. *Package ‘RcmdrPlugin.UCA’*. 2018. URL: <https://cran.r-project.org/web/packages/RcmdrPlugin.UCA/RcmdrPlugin.UCA.pdf>.

- project, Free Software Foundation's GNU. *r-project*. URL: <https://www.r-project.org/about.html>.
- Selvam, Sindhu. *R-Studio*. URL: <https://datascienceplus.com/introduction-to-rstudio>.
- Stallman, Richard. *GNU General Public License*. URL: <https://www.gnu.org/licenses/gpl-3.0.en.html>.
- Titanica, Encyclopedia. *RMS Titanic: An Introduction*. 1996-2020. URL: <https://www.encyclopedia-titanica.org/titanic/>.
- University, Standford. *titanic-dataset*. URL: <https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv>.